WHO'S BEHIND THE TEXT? ROBUST MODEL AUTHORSHIP ATTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models (LLMs) are increasingly used to generate text, their misuse has become a growing societal problem, with serious implications for academic misconduct, model distillation, and cyberattacks. For these reasons, Model Authorship Attribution (MAA), which identifies the source model of a given text, is important for promoting the responsible use of LLMs and improving AI safety. In practice, attribution systems must remain reliable beyond the distributions observed during training. They may encounter texts from unseen domains and writing settings, as well as generations from future versions of model families that were not available when the system was trained. However, existing MAA methods often exhibit substantial performance degradation under such distribution shifts. To address this challenge, we propose **A**uthorship **T**racing via **L**ikelihood-based **A**ttribution of **S**ources (ATLAS), which formulates MAA through source-conditioned text modeling rather than direct source prediction. ATLAS learns how text from each source family is generated and attributes a given text by comparing its likelihood under each learned source condition. We aggregate this likelihood evidence at the token level with clipping so that no single token dominates the decision. We further introduce MAABench, a novel and broad MAA benchmark that can evaluate more practical OOD settings such as domain and version. Our experiments show that ATLAS consistently outperforms existing baselines across MAABench and OpenTuringBench. In particular, when both the generation domain and the model version change on MAABench, ATLAS achieves 74.30% accuracy, 8.15pp higher than the strongest baseline. Beyond MAA, ATLAS can also help identify distillation source models by measuring which source model a target LLM most closely resembles. Our extended experiments support this claim by showing that ATLAS recovers the known teacher of every distilled model we evaluate.

1 INTRODUCTION

Large language models (LLMs) are now widely used across a range of writing tasks, including news, reviews, essays, and technical writing (Wu et al., 2024a). Accordingly, considerable attention has been devoted to machine-generated text (MGT) detection, which distinguishes LLM-generated text from human-written text (Mitchell et al., 2023; Bao et al., 2024; Hans et al., 2024; Lee et al., 2024). However, when harmful, misleading, or otherwise problematic content is generated by an LLM, merely identifying that the text is machine-generated is often insufficient. Further investigation may require identifying *which model* generated the text, for example, to trace the model or provider associated with the generated content.

This motivates the more fine-grained problem of *Model Authorship Attribution (MAA)*, which aims to identify the source model family of a given text. Beyond tracing the source of individual texts, MAA may also provide information about relationships between models. A particularly interesting example is output-based distillation, where a student model is trained on outputs generated by a teacher model. If source-specific characteristics of the teacher remain alive in the student's generations, attribution signals may provide clues about the teacher model. Recent incidents involving the reuse of frontier-model outputs further motivate the use of source attribution (Anthropic, 2026b).

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.

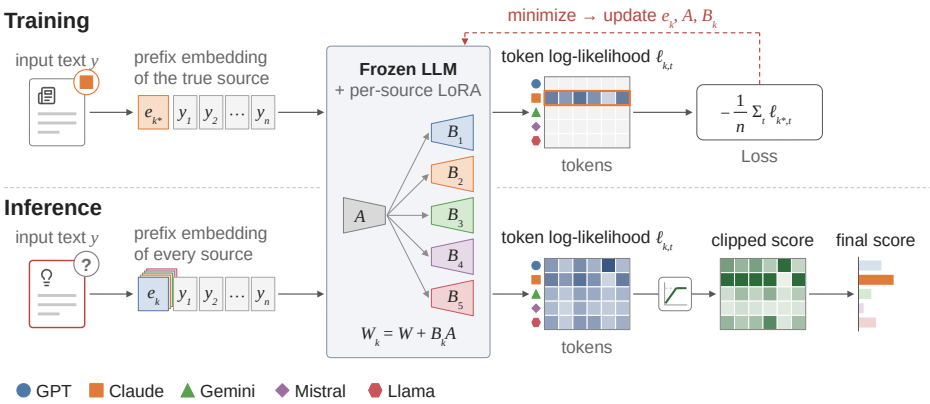


Figure 1: **Overview of ATLAS.** Rather than directly predicting the source, ATLAS learns a source-conditioned language model for each model family and performs attribution from their relative token likelihoods. During training, for each source k, ATLAS adapts a pretrained LLM with a prefix embedding e_k and a LoRA update $W_k = W + B_k A$, where A is shared down-projection and B_k is source-specific up-projection. For the ground-truth source k^*, ATLAS optimizes e_{k^*}, B_{k^*}, and the shared A using the mean negative log-likelihood. During inference, ATLAS calculates token log-likelihoods for all source k, which serve as source model scores after clipping.

A key practical challenge, however, is whether attribution remains reliable as the conditions under which text is generated change over time. A deployed attribution system may encounter texts from domains and writing settings that were absent during training, making it important to distinguish persistent source-specific signals from domain- or task-specific cues. Meanwhile, model families evolve rapidly, and new versions are continually released day by day. Retraining an attribution system whenever a source model is updated can be costly and impractical. Ideally, a system trained today should remain able to attribute texts produced by future versions of the same model family that were not available during training.

Existing MAA methods (Li et al., 2023; Venkatraman et al., 2024; Guo et al., 2024; La Cava & Tagarelli, 2025; Sun et al., 2025) perform well when the training and test distributions are similar, but often degrades under such changes. Because they directly predict source model from text, they may rely on source-discriminative cues that are unstable under distribution shift. Motivated by prior evidence that generative modeling can be more robust to such shortcut solutions (Li et al., 2025), we instead approach MAA from the generative and likelihood-based perspective.

Our proposed framework and benchmark. In this work, we propose **A**uthorship **T**racing via **L**ikelihood-based **A**ttribution of **S**ources (ATLAS), a likelihood-based MAA framework based on lightweight adaptations of a pretrained LLM. Rather than directly predicting the k-th source family from a given text y, ATLAS learns a source-conditioned model of the text distribution for each family and performs attribution by comparing their likelihoods. Furthermore, to prevent a small number of tokens with extreme likelihood differences from dominating the attribution, ATLAS compares source likelihoods at the token level and clips their relative evidence. Figure 1 illustrates ATLAS.

We also introduce MAABench, a new benchmark for evaluating MAA beyond the training distribution. Existing MAA benchmarks cover only part of this setting (Uchendu et al., 2021; La Cava & Tagarelli, 2025; La Cava et al., 2026), while other benchmarks that explicitly consider model version shift only focus on MGT detection (Yu et al., 2025). MAABench evaluates two major types of distribution shift (i.e., generation domain and source-model version). It contains texts generated by five recent LLM families, GPT (OpenAI, 2025), Claude (Anthropic, 2026a), Gemini (Google, 2026), Mistral (Jiang et al., 2023), and Llama (Meta, 2025), across six domains.

Across MAABench and OpenTuringBench (La Cava & Tagarelli, 2025), ATLAS consistently outperforms existing baselines under various Out-Of-Distribution (OOD) settings, as expected in our motivation for source-conditioned likelihood modeling. In the most challenging setting of MAABench, where both the generation domain and source-model version change, ATLAS improves over the strongest baseline by 8.15pp.

Interestingly, we further show that the source model scores learned by ATLAS can be useful beyond direct MAA. In specific, we use them to identify the teacher of a distilled model of interest, asking whether the outputs of a student model are similar to its teacher. On publicly available student models distilled from known teachers, ATLAS correctly identifies the teacher of every model, even when students of the same teacher are built on different base models.

Our main contributions are summarized as follows:

- **A likelihood-based method of model authorship attribution (MAA).** We propose ATLAS, which learns source-conditioned text distributions with lightweight adaptations and performs attribution from the token-level likelihoods. We further introduce clipping the token-level likelihoods for robust aggregation resulting in the final source model scores.
- **A new MAA benchmark covering various distribution shifts.** We introduce MAABench, which evaluates distribution shifts in generation domain, model version, and their combination.
- **Strong empirical performance under the distribution shifts.** Experiments on MAABench and OpenTuringBench (La Cava & Tagarelli, 2025) show that ATLAS consistently improves OOD attribution over existing baselines.
- **Source attribution for various distilled and deployed models.** We show that the source signals learned by ATLAS retain teacher-specific information after distillation, enabling teacher identification for distilled models.

2 Related Work

Machine-generated text (MGT) detection. One line of MGT detection trains supervised binary classifiers to distinguish MGT from human-written text (Chen et al., 2023a). However, they are often fragile under out-of-distribution (OOD) settings, limiting their practical applicability (Guo et al., 2024). To address these limitations, diverse approaches have been proposed: (i) Exploit signals from pretrained language models: probability curvature (Mitchell et al., 2023; Bao et al., 2024), n-gram divergence (Yang et al., 2024), paired-model likelihoods (Hans et al., 2024), and log-rank (Su et al., 2023) statistics. These signals are often computed under a surrogate model whose mismatch with the generator can motivate routing to a better-matched model (Sun et al., 2026a). (ii) Watermarking (Kirchenbauer et al., 2023) add perturbation during generation, but applies only when a provider chooses to embed a mark. (iii) Other approaches construct learned detection signals, for example using reward-based scores (Lee et al., 2024; Liu et al., 2025) or representations of writing style (Guo et al., 2024). In contrast, ATLAS derives attribution signal from source-conditioned token-level likelihoods. Moreover, while these works ask whether a given text is machine-generated, we ask which model wrote it.

Model authorship attribution (MAA). Beyond binary MGT detection, recent work studies the finer-grained task of identifying the source model of generated text. One line relies on statistics computed under frozen models: Sniffer (Li et al., 2023) contrasts model-specific features, GPT-who (Venkatraman et al., 2024) uses uniform information density, and LLMDet (Wu et al., 2023) estimates source-specific likelihoods from token statistics. Another learns text representations directly, using contrastive objectives to cluster texts by source (Guo et al., 2024; La Cava & Tagarelli, 2025), fine-tune text embedding models (Sun et al., 2025), formulate attribution as next-token prediction (Chen et al., 2023b), jointly model detection and attribution (Rao et al., 2025), or adapt to new sources without full retraining (Sun et al., 2026b). Existing MAA methods largely rely on source-discriminative signals, which can be unstable under distribution shift. In contrast, ATLAS learns source-conditioned text distributions and is more robust to domain and version shifts.

Benchmarks under distribution shift. Benchmarks for MGT detection have expanded from unified evaluations (He et al., 2024) to broader coverage of generators and domains (Wang et al., 2024; Fu et al., 2025), adversarial and real-world variation (Dugan et al., 2024; Wu et al., 2024b), and evolving model versions (Yu et al., 2025). MAA benchmarks have similarly moved from attribution among a fixed set of generators (Uchendu et al., 2021) to in-domain variations and a domain shift (La Cava & Tagarelli, 2025) and to multilingual attribution (La Cava et al., 2026). However, none of the MAA benchmarks considers version changes within the same model family, and EvoBench (Yu et al., 2025), which does, targets MGT detection rather than MAA. MAABench evaluates domain and version changes, separately and together, within one benchmark for MAA.

3 A LIKELIHOOD-BASED FRAMEWORK FOR AUTHORSHIP ATTRIBUTION

3.1 PROBLEM SETUP: MODEL AUTHORSHIP ATTRIBUTION

The goal of Model Authorship Attribution (MAA) is to identify which language model family generated a given text. Let $\mathcal{M}_1, \ldots, \mathcal{M}_K$ be the K candidate language model families (e.g., GPT, Claude, and Gemini), where $\mathcal{M}_k$ for $k \in \{1, \ldots, K\}$ denotes the set of models belonging to family k. For example, if family k corresponds to GPT, then $\mathcal{M}_k = \{\ldots, \text{GPT-3}, \ldots, \text{GPT-5.6}, \text{GPT-6}, \ldots\}$.

Given a text $y \in \mathcal{Y}$, we assume that y is generated from $p_M(\cdot \mid g)$ for some source model $M \in \mathcal{M} := \bigcup_{k=1}^{K} \mathcal{M}_k$ and generation condition $g \in \mathcal{G}$. Here, g captures factors other than the source-model identity that may affect generation, such as the domain, prompt, and available context. We say that the source family of y is k if $M \in \mathcal{M}_k$. The objective of MAA is therefore to learn an attribution function $f : \mathcal{Y} \to \{1, \ldots, K\}$ that predicts the source family of a given text as $\hat{k} := f(y)$.

Not only the In-Domain (ID) setting (train distribution of $y \approx$ test distribution of y), but we consider Out-Of-Distribution (OOD) settings, where texts given at inference time differ from the training texts, in terms of their generation conditions, source-model versions, or both. This includes cases involving previously unseen generation conditions or new versions of known source-model families.

3.2 ATLAS

We propose a new approach to model authorship attribution, called ATLAS (Authorship Tracing via Likelihood-based Attribution of Sources). A key idea of ATLAS is to formulate MAA through source-conditioned text modeling rather than direct source prediction. For each source family $\mathcal{M}_k$, ATLAS learns a source condition c_k that models the likelihood of texts generated by that family. Given a text y, attribution is performed by comparing how well the different source conditions explain the observed text. Prior work has shown that such class-conditional generative modeling can be less reliant on shortcut features under distribution shift than direct discriminative prediction (Li et al., 2025). In particular, we compare source conditions individually (i.e., token by token), rather than aggregating token likelihoods. Then, the log-likelihood of each token is clipped, so a few tokens with extreme log-likelihoods cannot determine the attribution solely.

Training: source-specific adaptation. During training, for a text $y = (y_1, \ldots, y_n)$ with n tokens and its ground-truth source family $k^* = k^*(y)$, we maximize its likelihood under the corresponding source condition. The same source condition c_k is used for all training texts from family k, irrespective of the particular generation settings represented in the training data. That is, we consider the mean negative log-likelihood as the training objective: $\mathcal{L}(y) = -\frac{1}{n}\sum_{t=1}^{n} \ell_{k^*,t}$, where $\ell_{k,t} = \log p(y_t \mid y_{<t}, c_k)$ denotes the t-th token log-likelihood under source condition c_k. Importantly, ATLAS does not use a loss that directly predicts the source label. The source label is used only to select the condition under which the text likelihood is optimized.

We implement the source conditions within a frozen pretrained backbone using a learnable prefix embedding and source-specific LoRA adaptation (Hu et al., 2022). That is, for each source k, we train $c_k = (e_k, B_k)$, where e_k is a learnable source-specific prefix embedding and B_k is the source-specific LoRA up-projection. For each backbone weight W, the source-conditioned weight is then computed as $W_k = W + \Delta W_k$, where $\Delta W_k = B_k A$ and A is the down-projection shared across all sources. We note that this conditioning technique is also related to the asymmetric and efficient LoRA adaptation (Tian et al., 2024; Zhu et al., 2024).

Inference: aggregating source-conditioned likelihood evidence. At inference time, the given text is evaluated under all K source conditions. A natural approach is to average the token log-likelihoods $\ell_{k,t} = \log p(y_t \mid y_{<t}, c_k)$ for all $t = 1, \ldots, n$ and select the source family with the highest mean likelihood. However, this makes the attribution score sensitive to a small number of tokens with extreme likelihood differences. We therefore compare the source conditions token by token and bound the contribution of each token through clipped relative likelihoods.

For each source k and token y_t, we compare its likelihood against every competing source and clip each pairwise ratio at one. This yields a bounded score $r_{k,t} \in (0, 1]$ that equals 1 when source k assigns the highest likelihood and decreases as competing sources explain the token better, preventing

a few tokens from dominating the final attribution score. Because $r_{k,t}$ depends only on relative log-likelihood differences, it is invariant to source-independent additive shifts. We average these scores across tokens and temperatures to obtain R_k, and predict $\hat{k} = \arg\max_k R_k$. The exact formulations are given in Appendix B.1.

4 MAABENCH

This section introduces MAABench, a new benchmark for evaluating MAA under distribution shift that covers a practically important setting that previous benchmarks did not. We consider two types of distribution shift: the generation domain and the source model version. Based on these two types, we build three OOD settings where the domain changes, the model version changes, or both change.

4.1 DATA CONSTRUCTION

Model families and generation domains. We construct MAABench using generations from five recent model families: GPT (OpenAI, 2025), Claude (Anthropic, 2026a), Gemini (Google, 2026), Mistral (Jiang et al., 2023), and Llama (Meta, 2025). The dataset contains 4 model versions for each model family, corresponding to different release versions. Specifically, in the version-OOD setting, the oldest version is used for training, and the remaining three versions are used for test.

For the generation domains, we consider six: News (Narayan et al., 2018), Wikipedia (Wikimedia Foundation, 2023), Academic Writing (Clement et al., 2019), Product Review (Hou et al., 2026), Patent (Suzgun et al., 2023), and Stack Exchange (Stack Exchange, Inc., 2014). In detail, for the domain-OOD setting, we use News, Wikipedia, Academic Writing, and Product Review for the training data, and the remaining two domains (i.e., Patent and Stack Exchange) are used for test.

Construction details. For each domain, we construct generation prompts from the metadata that accompanies each human-written text in its source corpus (e.g., the headline of a news article) and give the same prompt to all five families, yielding prompt-paired generations; the human-written texts are kept for MGT detection. Moreover, following Fu et al. (2025), we apply the same length constraint and filtering procedure to all source models, targeting 90-220 words and retaining only complete prompt-paired groups that pass the filter.

4.2 EVALUATION SETTINGS

We evaluate one in-distribution (ID) setting and three out-of-distribution (OOD) settings. *ID* uses the domains and source-model versions observed during training. *Domain shift* uses the held-out domains, generated with a persona-based system prompt instead of the default one (Appendix A.3), while keeping the training-time source-model versions fixed. *Version shift* uses the training domains but replaces each training-time source model with one of three held-out versions from the same family. *Joint shift* are the shift when both the Domain shift and Version shift jointly occur at the same time. Table 1 summarizes the resulting dataset with corresponding splits and evaluation settings, which contains 228,520 machine-generated texts and 35,443 human-written texts. Further details on MAABench are provided in Appendix A.

Table 1: **Summary of MAABench.**

Distribution	Split	Setting	MGT	Human
ID	Train	–	149,760	29,952
	Val	–	9,990	1,998
	Test	–	9,985	1,997
OOD	Test	Domain	7,470	1,494
		Version	29,700	1,980†
		Joint	21,615	1,441†
Total			**228,520**	**35,443**

5 EXPERIMENTS

In this section, we empirically evaluate ATLAS from three perspectives. First, we compare ATLAS with existing MAA methods on MAABench and OpenTuringBench under both ID and OOD settings. Second, we examine whether the source-conditioned likelihoods learned by ATLAS can serve as good signals for binary MGT detection. Finally, we analyze the contribution of each component of ATLAS and how its performance depends on the backbone model and the training data size. Further analyses of these results are provided in Appendix B.6.

Table 2: **MAA performance on MAABench.** We report accuracy (%, ↑) and macro-F1 (%, ↑). Best results are in **bold**.

Method	ID		OOD					
			Domain		Version		Joint	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Fast-DetectGPT (Bao et al., 2024)	44.37	43.54	45.29	44.18	35.22	34.75	31.85	30.81
Binoculars (Hans et al., 2024)	45.61	44.81	46.53	45.70	36.09	35.30	32.56	31.17
GPT-who (Venkatraman et al., 2024)	58.63	58.29	55.29	56.17	41.71	40.34	39.94	38.43
Sniffer (Li et al., 2023)	76.00	75.83	64.51	64.89	51.97	50.72	47.57	45.85
StatEnsemble (La Cava et al., 2026)	74.36	74.11	58.07	58.52	38.86	38.77	32.29	32.35
DeTeCtive (Guo et al., 2024)	99.49	99.49	95.38	95.39	70.08	69.43	62.92	61.58
OTBDetector (La Cava & Tagarelli, 2025)	99.42	99.42	95.44	95.46	71.27	70.96	65.02	64.43
DA-MTL (Rao et al., 2025)	99.45	99.45	94.59	94.61	72.78	72.54	66.15	65.09
Idiosyncrasies (Sun et al., 2025)	99.52	99.52	94.39	94.41	71.02	70.60	65.66	64.71
ATLAS (ours)	**99.77**	**99.77**	**98.11**	**98.12**	**84.36**	**84.33**	**74.30**	**73.96**

5.1 Experimental Setup

Implementation details and evaluation setup. For the main experiments, we use Qwen2.5-7B (Qwen et al., 2025) as the backbone for ATLAS. On MAABench, the models of all methods are trained on the ID training data and evaluated on the ID and three OOD test data (Domain, Version, and Joint). On OpenTuringBench (La Cava & Tagarelli, 2025), we follow its evaluation setting (i.e., on the original ID, four ID variations (E1-E4), and OOD). We report accuracy and macro-F1 in main MAA experiments. For the MGT detection, we report AUROC, the true positive rate at a 1% false positive rate (TPR@1%FPR), and the false positive rate at a 95% true positive rate (FPR@95%TPR), with MGT as the positive class. Further implementation details are in Appendix B.1.

Baselines. We compare ATLAS with nine MAA baselines. Five methods use statistics from proxy models: Fast-DetectGPT (Bao et al., 2024), Binoculars (Hans et al., 2024), GPT-who (Venkatraman et al., 2024), Sniffer (Li et al., 2023), and StatEnsemble (La Cava et al., 2026). The remaining four methods learn representations for attribution: DeTeCtive (Guo et al., 2024), OTBDetector (La Cava & Tagarelli, 2025), DA-MTL (Rao et al., 2025), and Idiosyncrasies (Sun et al., 2025). For MGT detection, we additionally compare with ReMoDetect (Lee et al., 2024) and IRM (Liu et al., 2025), which detect MGT by using reward signals. Detailed implementations are in Appendix B.2.

5.2 Main Results

MAA performance Table 2 compares ATLAS with the nine baselines on MAABench. While the ID setting is nearly saturated (i.e., ATLAS and some of baselines achieve over 99% accuracy), the performance gaps are substantial under OOD evaluation. That is, ATLAS outperforms the strongest baseline by 2.67pp, 11.58pp, and 8.15pp on Domain shift, Version shift, and Joint shift, respectively. The two OOD types also affect ATLAS differently: relative to ID, accuracy decreases by only 1.66pp on Domain shift, but with 15.41pp on Version shift. These results imply that source attribution can generalize more readily across the generation domains than across unseen versions.

Table 3 further evaluates the same methods on OpenTuringBench (La Cava & Tagarelli, 2025). ATLAS achieves the strongest OOD performance, achieving 95.71% accuracy and outperforming the best baseline by 33.52pp. It also achieves the best performance on all four ID variations (i.e., E1-E4), showing that the improvement of ATLAS is not tied to our benchmark MAABench. Notably, the advantage of ATLAS becomes substantially larger in the OOD setup, aligned with our motivation for modeling source-conditioned likelihoods instead of directly optimizing source attribution.

MGT detection performance As a secondary analysis, we examine whether the source-conditioned likelihoods learned by ATLAS are useful for MGT detection. Table 4 shows that, ATLAS performs competitive to the strongest detectors trained on MAABench across all cases, suggesting that one can apply ATLAS not only to MAA but also to the MGT detection. Table 17 in Appendix B.4 shows the similar results, with respect to TPR@1%FPR and FPR@95%TPR.

Table 3: **MAA performance on OpenTuringBench.** We report accuracy (%, ↑), and E1-E4 are the in-domain variation tasks. Best results are in **bold**.

Method	ID					OOD
	ID	E1	E2	E3	E4	
Fast-DetectGPT (Bao et al., 2024)	18.93	19.49	17.57	21.68	15.37	19.92
Binoculars (Hans et al., 2024)	19.54	22.91	20.33	24.79	16.41	23.15
GPT-who (Venkatraman et al., 2024)	47.55	22.41	44.68	31.09	15.05	25.89
Sniffer (Li et al., 2023)	61.65	27.25	55.78	38.67	15.53	30.12
StatEnsemble (La Cava et al., 2026)	61.93	60.52	58.40	63.30	26.99	48.46
DeTeCtive (Guo et al., 2024)	99.83	84.43	99.36	89.00	27.48	56.61
OTBDetector (La Cava & Tagarelli, 2025)	99.76	84.41	99.11	88.03	30.89	47.92
DA-MTL (Rao et al., 2025)	**99.87**	83.79	99.48	95.99	32.36	62.19
Idiosyncrasies (Sun et al., 2025)	89.90	70.16	86.56	79.52	28.64	41.11
ATLAS (ours)	99.80	**89.71**	**99.70**	**98.79**	**41.92**	**95.71**

Table 4: **MGT detection on MAABench.** We report AUROC (%, ↑) with MGT as the positive class; *Trained on MAABench* marks methods trained on the MAABench training split, while the others use off-the-shelf models. Best results are in **bold**.

Method	Trained on MAABench	ID	OOD		
			Domain	Version	Joint
Fast-DetectGPT (Bao et al., 2024)	×	65.59	65.25	57.56	50.74
Binoculars (Hans et al., 2024)	×	71.14	71.74	62.56	57.57
IRM (Liu et al., 2025)	×	73.85	62.62	70.16	54.31
ReMoDetect (Lee et al., 2024)	×	77.27	82.33	73.06	80.32
DeTeCtive (Guo et al., 2024)	✓	**100.00**	98.60	99.95	97.97
OTBDetector (La Cava & Tagarelli, 2025)	✓	99.94	99.39	99.80	96.81
DA-MTL (Rao et al., 2025)	✓	**100.00**	99.67	**100.00**	99.66
Idiosyncrasies (Sun et al., 2025)	✓	**100.00**	99.65	**100.00**	99.21
ATLAS (ours)	✓	**100.00**	**99.96**	**100.00**	**99.75**

5.3 Ablation and Scaling Analysis

Source modeling and likelihood aggregation. We analyze the two main design choices of ATLAS, source-specific adaptation and clipped likelihood aggregation. Source-specific B_k gives each source condition its own adaptation parameters, while the clipping is a technique to carefully aggregate the token-level likelihoods. As shown in Table 5, both components improve performance when the source-model version changes. Particularly, the clipping leads to larger gains, improving Version and Joint shift by 4.55pp and 5.12pp, respectively. Table 18 in Appendix B.5 includes the results on ID and Domain shift, showing that the performance differs by less than 0.7pp on the two settings.

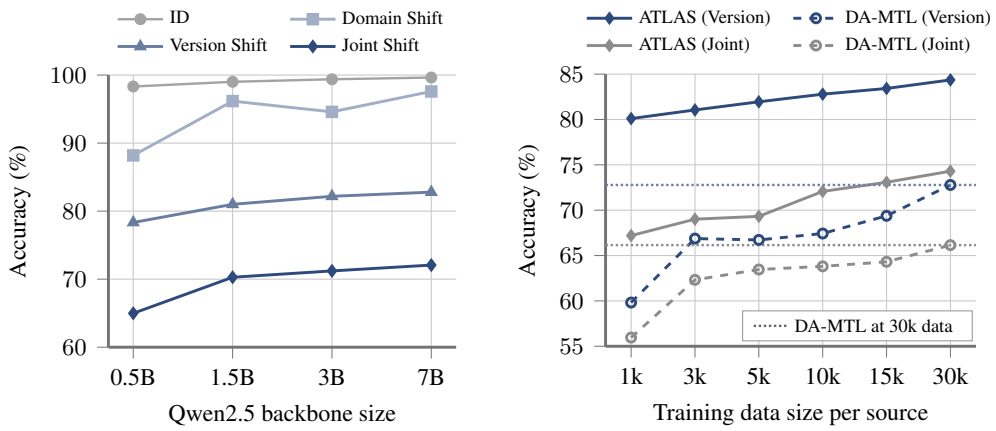
Table 5: **Component analysis.** Accuracy (%, ↑) under the two version-related shifts.

Up-projection	Scoring	Version	Joint
Shared B	Mean	76.06	66.11
Per-source B_k	Mean	78.24	66.95
Per-source B_k	Clipped	**82.79**	**72.07**

Effect of backbone model size. We vary the Qwen2.5 backbone from 0.5B to 7B parameters and compare the results. As shown in Figure 2(a), larger backbones generally improve attribution performance, with substantially larger gains under OOD evaluation than under ID. We further replace the backbone with OLMo-2-7B and Qwen3-8B. As shown in Table 6, accuracy stays within about 1pp of Qwen2.5-7B on Domain and Version shift, and Qwen3-8B improves Joint shift to 75.22%, so the gains of ATLAS do not depend on a particular backbone. Table 20 in Appendix B.5 additionally reports accuracy and macro-F1 on the ID test data.

Table 6: **Effect of backbone family.** Accuracy (%, ↑) under three different backbones.

Backbone	Domain	Version	Joint
Qwen2.5-7B	97.59	82.79	72.07
OLMo-2-7B	97.05	**83.75**	72.99
Qwen3-8B	**98.09**	83.40	**75.22**



(a) Backbone size

(b) Training data size

Figure 2: **Effect of backbone and training data size.** (a) Accuracy (%, ↑) of ATLAS with four Qwen2.5 backbone sizes, each trained on 10,000 texts per source family. (b) Accuracy (%, ↑) on Version Shift and Joint Shift against the number of training texts per source, compared with DA-MTL. The dotted lines mark DA-MTL on the full split of 30k texts.

Data efficiency of ATLAS. We also vary the number of training texts per source family from 1,000 to the full split, and train the four encoder-tuning baselines on subsets of the same sizes. As shown in Figure 2 (b), ATLAS is markedly more data-efficient. Notably, with only 1,000 texts per source, it already outperforms the strongest baseline trained on the full split on both Version shift (80.10% vs. 72.78%) and Joint shift (67.20% vs. 66.15%). Appendix B.5 reports the results for all four settings, showing similar trends.

6 Beyond Attribution: Distilled and Deployed Models

Beyond text-level attribution, we ask whether the source-conditioned likelihoods learned by ATLAS capture family-specific generation patterns that persist after distillation. In specific, a student trained on the outputs of a teacher may retain such characteristics in its generations. We first evaluate this question on distilled models with explicitly reported teachers, and then extend the analysis to recently deployed models for which no teacher, or even the use of distillation, is publicly established.

Setup. From the MAABench training splits, we here use only the data from GPT (OpenAI, 2025), Claude (Anthropic, 2026a), and Gemini (Google, 2026) for training, since they are among the most widely used frontier model families and therefore plausible sources of teacher signals (Sajadieh et al., 2026). For validation, we use six publicly available distilled models whose teachers are explicitly reported and belong to one of these three families. We consider two students for each teacher, built on different base models (see the distilled model details in Table 23).

Beyond these models with known distillation histories, we also examine seven recent deployed models from DeepSeek (DeepSeek-AI et al., 2026), GLM (GLM-5-Team et al., 2026), MiniMax (Chen et al., 2026), and Kimi (Team et al., 2026). For these models, we do not assume that they were distilled from the three source families, or that distillation was used at all. Instead, we use them as an exploratory setting to examine which, if any, known source family their generations most closely resemble. Appendix C gives further details about how we generate texts using the models.

Inference. We compute the mean token log-likelihood $s_k(y) := \frac{1}{n}\sum_{t=1}^{n} \ell_{k,t}$ under each source condition k and subtract the corresponding likelihood $s_0(y)$ under the frozen backbone, yielding $\tilde{s}_k(y) = s_k(y) - s_0(y)$. We then standardize $\tilde{s}_k(y)$, yielding a source-specific score $z_k(y) = (\tilde{s}_k(y) - \mu_k)/\sigma_k$, where μ_k and σ_k are the mean and standard deviation of $\tilde{s}_k$ over reference texts from Mistral and Llama, which lie outside the three candidate families. Here, $z_k(y)$ measures how strong the condition k explains the text relative to unrelated families. Each text is attributed to the source with the largest $z_k(y)$, and we report the attribution share for each model. We additionally

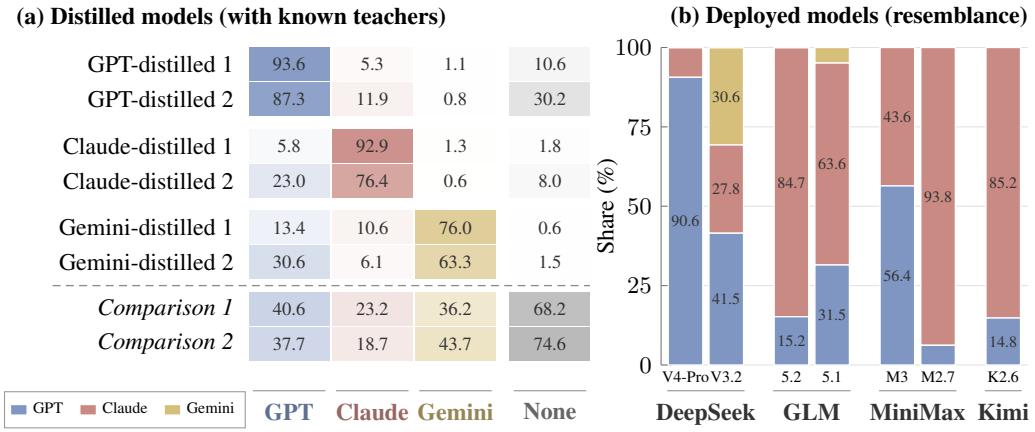


Figure 3: **Attribution of distilled and deployed models.** Share of each model's texts attributed to each candidate family, in percent. (a) Distilled models with known teachers: *None* is the share of texts that no condition explains better than the frozen backbone. (b) Seven recent releases of widely deployed models: segments under 12% are left unlabeled.

report the rejection rate, defined as the fraction of texts for which no source condition improves over the frozen backbone, i.e., $\max_k \tilde{s}_k(y) < 0$.

Results. As the left panel (a) of Figure 3 shows, using ATLAS, we successfully attribute all six distilled models to their true known teachers, with scores from 63.3% to 93.6%. It is noteworthy that the two GPT-distilled students are both attributed to GPT, even though they are built on different base models, Qwen3-8B and Llama-3.1-8B, providing evidence that the learned source conditions capture common generation patterns of the teacher model family rather than only the student's base model. Table 22 in Appendix C presents the detailed results.

The right panel (b) of Figure 3 shows that, among the deployed models with no reported teacher, four models are mostly attributed to a single source family, at levels comparable to the distilled models with known teachers. That is, MiniMax-M2.7 Turbo, Kimi-K2.6, and GLM 5.2 most resemble Claude in 93.8%, 85.2%, and 84.7% of their texts, respectively, while DeepSeek-V4-Pro most resembles GPT in 90.6%.

The rejection rate ranges from 0.6% to 30.2% for the validated distilled models and is at most 0.4% for the seven deployed models, but rises to 68.2% and 74.6% for two comparison models, Falcon3-7B-Instruct and Granite-3.0-8B-Instruct, with no reported distillation from three families. Overall, these results show that ATLAS not only recovers the (known) teachers of distilled models, but also characterizes which model family a given deployed model most closely resembles.

7 Conclusion and Discussion

In this work, we developed a new method ATLAS for model authorship attribution (MAA) through source-conditioned text modeling rather than direct source prediction. ATLAS learns how each source family generates text and performs attribution from the resulting likelihood evidence. On MAABench and OpenTuringBench, ATLAS consistently improves attribution under distribution shift compared to existing baselines, especially for the most challenging OOD setting when both the generation domain and the source-model version change at test time. The same source-conditioned likelihoods can also be applied to distinguish human-written text from MGT. They further identify the known teachers of distilled models and indicate which family deployed models most resemble.

This work focuses on attribution among a fixed set of candidate source families under the domains and generation conditions covered by MAABench. Extending attribution to previously unseen source families, particularly without full retraining, remains an important direction for future work. Another promising extension is mixed-authorship attribution for text jointly written by humans and models, where the task shifts from assigning a single source to identifying which source contributed each part of the text.

AI USE STATEMENT

We used generative AI tools partially for this work. In specific, we used them to generate the synthetic datasets, as described in the benchmark construction section. We also used generative AI tools to draft and polish the manuscript. All AI-assisted content and generated data were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including the dataset construction, experimental methodology, analyses, claims, and manuscript text.

ETHICS STATEMENT

MAABench is built from six public corpora, whose exact repository revisions are listed in Appendix A. The source models receive only the metadata extracted from each seed, never the human-written text itself. We plan to release the machine-generated texts together with the seed identifiers and corpus revisions, but not the human-written texts, which can be recovered from the original corpora. No human subjects were recruited, and we make no attempt to identify the authors of the human-written texts. The analysis of distilled and deployed models in Section 6 reports which known source family a target model most resembles, which is not evidence of distillation: shared pretraining corpora, similar alignment recipes, or common synthetic data can produce the same signal. We therefore make no claim about the training data of any target model, and these results are not an allegation against the organizations that release them.

REPRODUCIBILITY STATEMENT

Appendix A reports the source models and their versions, the seed corpora with their repository revisions, the prompt templates, the generation configuration, and the filtering criteria. Appendix B reports the optimization settings (Table 12), the protocol for each baseline (Appendix B.2), and the setup of every analysis, and Appendix C describes the distillation analysis with its validation checkpoints. We plan to release the generated texts and code upon publication. Every reported number comes from a single training run with a fixed seed.

REFERENCES

Anthropic. System card: Claude opus 5, 2026a. URL https://www-cdn.anthropic.com/ceaf5c7ff2783855203fde8208ec311252dced5b/Claude%20Opus%205%20System%20Card.pdf.

Anthropic. Detecting and countering misuse of AI: September 2026, September 2026b. URL https://www-cdn.anthropic.com/e50be2e51e7695dc4b1366a37a245a597377d3b5/Anthropic-Detecting-and-countering-091026.pdf.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bpcgcr8E8Z.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=IW1PR7vEBf.

Aili Chen, Aonian Li, Baichuan Zhou, Bangwei Gong, Binyang Jiang, Boji Dan, Changhao Zhang, Changqing Yu, Chao Wang, Cheng Ma, Cheng Zhong, Cheng Zhu, Chengjun Xiao, Chengyi Yang, Chengyu Du, Chenyang Zhang, Chi Zhang, Chuangyi Huang, Chunhao Zhang, Chunhui Du, Chunyu Zhao, Congchao Guo, Da Chen, Deming Ding, Dianjun Sun, Dong Li, Dongyu Zhang, Enhui Yang, Fei Yu, Guang Zheng, Guodong Zheng, Guohong Li, Haichao Zhu, Haigang Zhou, Haimo Zhang, Han Ding, Hao Zhang, Haohai Sun, Haolin Lyu, Haonan Lu, Haoyu Wang, Huajie Shi, Huiyang Li, Jiacheng Chen, Jian Zhang, Jiaqi Zhuang, Jiaren Cai, Jiaxin Pan, Jiayao Li, Jiayuan Song, Jichuan Zhang, Jie Wang, Jihao Gu, Jin Zhu, Jingwei Dong, Jingyang Li, Jingyu Zhang, Jingze Zhuang, Jinhao Tian, Jinli Liu, Jinyi Hu, Jun Tao, Jun Zhang, Junbin Ruan, Junhao Xu, Junjie Yan, Junteng Liu, Junxian He, Kang Xu, Ke Ji, Ke Yang, Kecheng Xiao, Keyu

Duan, Keyu Li, Le Han, Letian Ruan, Li Yuan, Lianfei Yu, Liheng Feng, Lijie Mo, Lin Li, Linge Du, Lingye Bao, Lingyu Yang, Lingyuan Zhou, Loki, Lu Chen, Lunbin Zeng, Ming Li, Ming Zhong, Mingliang Tao, Mingyuan Chi, Mujie Lin, Nan Hu, Ningxin Chen, Peiyin Zhu, Peng Gao, Pengcheng Gao, Pengfei Li, Penglin Li, Pengyu Zhao, Qibin Ren, Qibing Ren, Qidi Xu, Qihan Ren, Qile Li, Qin Wang, Quanliang Chen, Qunhong Zeng, Rong Tian, Rongxin Guo, Rui Dong, Ruitao Leng, Ruize Zhang, Shanqi Liu, Shaoxiang Chen, Shaoyu Chen, Sheng Jia, Shun Yao, Shuoran Zhao, Shuqi Yu, Sichen Li, Sicheng Pan, Songquan Zhu, Tengfei Li, Tian Xie, Tiancheng Qin, Tianle Li, Tianrun Liang, Wei Liu, Weiqi Xu, Weitao Li, Weixiang Chen, Weiyu Cheng, Weiyu Zhang, Wenhu Chen, Wenqian Zhao, Xiancai Chen, Xiangjun Song, Xiangyuan Wang, Xianzhen Luo, Xiao Luo, Xiao Su, Xiaobo Li, Xiaodong Han, Xiaojie Wu, Xihao Song, Xingyi Han, Xinyu Guan, Xuan Lu, Xun Zou, Xunhao Lai, Xutong Li, Xuyang Shen, Yan Gong, Yan Ma, Yang Jiao, Yang Wang, Yang Xu, Yangsen Wang, Ye Tang, Yicheng Chen, Yihang Wang, Yinran Qiu, Yiqi Shi, Yiting Guo, Yiwen Huang, Yixuan Wang, Yongyi Hu, Yu Gao, Yu Zhang, Yuan Li, Yuanxiang Ying, Yuanzhen Zhang, Yubo Wang, Yuchen Song, Yufeng Yang, Yuhang Meng, Yuhang Miao, Yuhao Li, Yujie Liu, Yulin Hu, Yunan Huang, Yunji Li, Yunyi Huang, Yusen Zhang, Yusu Hong, Yutao Xie, Yutong Zhang, Yuwen Liao, Yuxuan Shi, Yuze Wenren, Zebin Li, Zehan Li, Zejian Luo, Zeyu Jin, Zeyuan Sun, Zhanpeng Zhou, Zhaochen Su, Zhendong Li, Zhengmao Zhu, Zhengyuan Peng, Zhenhua Fan, Zhi Zhang, Zhichao Xu, Zhiheng Lv, Zhikang Xu, Zhitao He, Zhiwei He, Zhongyuan Li, Zibo Gao, Zijia Wu, Zijian Song, Zijian Zhou, Zijun Sun, Zishan Huang, Ziying Chen, and Ziyue Ge. The minimax-m2 series: Mini activations unleashing max real-world intelligence, 2026. URL https://arxiv.org/abs/2605.26494.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. Gpt-sentinel: Distinguishing human and chatgpt generated content, 2023a. URL https://arxiv.org/abs/2305.07969.

Yutian Chen, Hao Kang, Yiyan Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. Token prediction as implicit classification to identify LLM-generated text. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13112–13120, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.810. URL https://aclanthology.org/2023.emnlp-main.810/.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. On the use of arxiv as a dataset, 2019. URL https://arxiv.org/abs/1905.00075.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao

Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12463–12492, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.674. URL https://aclanthology.org/2024.acl-long.674/.

Jiachen Fu, Chun-Le Guo, and Chongyi Li. Detectanyllm: Towards generalizable and robust detection of machine-generated text across domains and models, 2025. URL https://arxiv.org/abs/2509.14268.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng,

Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Google. Gemini 3.7 flash model card, 2026. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-7-Flash-Model-Card.pdf.

Xun Guo, Yongxin He, Shan Zhang, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. Detective: Detecting AI-generated text via multi-level contrastive learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=cdTTTJfJe3.

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. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=axl3FAkpik.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. Mgtbench: Benchmarking machine-generated text detection, 2024. URL https://arxiv.org/abs/2303.14822.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3251–3265, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.147. URL https://aclanthology.org/2026.acl-long.147/.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Lucio La Cava and Andrea Tagarelli. OpenTuringBench: An open-model-based benchmark and framework for machine-generated text detection and attribution. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 26655–26671, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1354. URL https://aclanthology.org/2025.emnlp-main.1354/.

Lucio La Cava, Dominik Macko, Robert Moro, Ivan Srba, and Andrea Tagarelli. Authorship attribution in multilingual machine-generated texts. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45136–45152, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2091. URL https://aclanthology.org/2026.acl-long.2091/.

Hyunseok Lee, Jihoon Tack, and Jinwoo Shin. Remodetect: Reward models recognize aligned LLM's generations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=pW9Jwim918.

Alexander Cong Li, Ananya Kumar, and Deepak Pathak. Generative classifiers avoid shortcut solutions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=oCUYc7BzXQ.

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. Origin tracing and detecting of llms, 2023. URL https://arxiv.org/abs/2304.14072.

Runheng Liu, Heyan Huang, Xingchen Xiao, and Zhijing Wu. Zero-shot detection of LLM-generated text via implicit reward model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2VdsYVXLDl.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, 2025. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

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, 2018. URL https://arxiv.org/abs/1808.08745.

OpenAI. Gpt-5 system card, 2025. URL https://cdn.openai.com/gpt-5-system-card.pdf.

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

Zixin Rao, Youssef Mohamed, Shang Liu, and Zeyan Liu. Two birds with one stone: Multi-task detection and attribution of llm-generated text, 2025. URL https://arxiv.org/abs/2508.14190.

Sha Sajadieh, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Lapo Santarlasci, Juan Pava, Nestor Maslej, Russ Altman, Erik Brynjolfsson, Carla Brodley, Jack Clark, Virginia Dignum, Vipin Kumar, James Landay, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Elham Tabassi, Russell Wald, Toby Walsh, and Dan Weld. Artificial intelligence index report 2026, 2026. URL https://arxiv.org/abs/2606.15708.

Stack Exchange, Inc. Stack exchange data dump, 2014. URL https://archive.org/details/stackexchange.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=Dy2mbQIdMz.

Ke Sun, Guangsheng Bao, Han Cui, and Yue Zhang. Minimizing mismatch risk: A prototype-based routing framework for zero-shot LLM-generated text detection. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=Hh1hf4kauu.

Mingjie Sun, Yida Yin, Zhiqiu Xu, J Zico Kolter, and Zhuang Liu. Idiosyncrasies in large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=FCZ3jVzmTZ.

Zhen Sun, Yifan Liao, Zhicong Huang, Jiaheng Wei, Cheng Hong, Yutao Yue, and Xinlei He. When new generators arrive: Lifelong machine-generated text attribution via ridge feature transfer, 2026b. URL https://arxiv.org/abs/2606.05626.

Mirac Suzgun, Luke Melas-Kyriazi, Suproteem K Sarkar, Scott Kominers, and Stuart Shieber. The harvard USPTO patent dataset: A large-scale, well-structured, and multi-purpose corpus of patent applications. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=tk27oD2cBw.

Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Chenxiao Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Qizheng Gu, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yang Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Haoyu Lu, Lijun Lu, Yashuo Luo, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Zeyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Lin Sui, Xinjie Sun, Flood Sung, Yunpeng Tai, Heyi Tang, Jiawen Tao, Qifeng Teng, Chaoran Tian, Chensi Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Si Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Haoning Wu, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Jing Xu, Jing Xu, Junjie Yan, Yuzi Yan, Hao Yang, Xiaofei Yang, Yi Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Siyu Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Shaojie Zheng, Longguang Zhong, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi k2: Open agentic intelligence, 2026. URL https://arxiv.org/abs/2507.20534.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng zhong Xu. HydraloRA: An asymmetric loRA architecture for efficient fine-tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=qEpi8uWX3N.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2001–2016, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.172. URL https://aclanthology.org/2021.findings-emnlp.172/.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. GPT-who: An information density-based machine-generated text detector. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 103–115, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.8. URL https://aclanthology.org/2024.findings-naacl.8/.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1369–1407, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.83.

Wikimedia Foundation. Wikimedia downloads, 2023. URL https://dumps.wikimedia.org.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. A survey on llm-generated text detection: Necessity, methods, and future directions, 2024a. URL https://arxiv.org/abs/2310.14724.

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. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL https://openreview.net/forum?id=ZGMkOikEyv.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. LLMDet: A third party large language models generated text detection tool. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2113–2133, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.139. URL https://aclanthology.org/2023.findings-emnlp.139/.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Xlayxj2fWp.

Xiao Yu, Yi Yu, Dongrui Liu, Kejiang Chen, Weiming Zhang, Nenghai Yu, and Jing Shao. EvoBench: Towards real-world LLM-generated text detection benchmarking for evolving large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14605–14620, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.754. URL https://aclanthology.org/2025.findings-acl.754/.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez De Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62369–62385. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhu24c.html.

A MAABENCH DETAILS

Appendices A.1–A.4 detail the data construction of Section 4.1, and Appendix A.5 describes how the evaluation settings of Section 4.2 are kept free of data leakage.

A.1 SOURCE MODELS

We construct MAABench from five LLM families representing both black-box and openly accessible white-box models: GPT (OpenAI, 2025), Claude (Anthropic, 2026a), Gemini (Google, 2026), Mistral (Jiang et al., 2023), and Llama (Meta, 2025). For each family, we consider one model version as the training-time source model and evaluate under three subsequently released versions for the Version shift and Joint shift settings.

Table 7: **Source models of MAABench.** The training version is used for ID and Domain shift, and the three held-out versions for Version shift and Joint shift.

Family	Training Version	Version shift	Access
GPT	GPT-5	GPT-5.2 / GPT-5.4 / GPT-5.6-terra	API
Claude	Opus 4.6	Opus 4.7 / Opus 4.8 / Opus 5	API
Gemini	2.5 Flash	3 Flash / 3.5 Flash / 3.6 Flash	API
Mistral	Small-24B-2501	Small-3.1-24B-2503 / Small-3.2-24B-2506 / small-2603	Local
Llama	Llama-3-70B	Llama-3.1-70B / Llama-3.3-70B / Llama-4-Scout-17B	Local

Table 7 summarizes the models used in each family. Black-box models are accessed through their respective APIs, while openly accessible white-box models are locally served. The three held-out versions are generated independently of the training versions.

A.2 HUMAN-WRITTEN TEXT CORPUS

We construct MAABench from human-written texts that serve as seeds for generating the machine-generated texts. For example, a news seed consists of an XSum article and its headline. Every source model receives the same prompt constructed from the headline, while the original article is retained as the human-written text.

We consider six domains including general-purpose, technical, commercial, and community-generated text, each drawn from one public corpus: news from XSum (Narayan et al., 2018), Wikipedia from the English Wikipedia dump (Wikimedia Foundation, 2023), academic writing from arXiv abstracts (Clement et al., 2019), product reviews from Amazon Reviews (Hou et al., 2026), patents from HUPD (Suzgun et al., 2023), and Stack Exchange from its public data dump (Stack Exchange, Inc., 2014). For each seed, we retain the original text together with domain-specific metadata used to construct the generation prompt. Note that the human-written text itself is not provided to the source models, and only the extracted metadata are used as generation inputs. The original human-written text is retained separately as the human class.

We collect 8,500 seeds for each of the four ID domains and 750 seeds for each of the two unseen domains. The seeds for the ID domain are divided into training, validation, and test splits at the seed level, ensuring that all generations derived from the same seed remain within the same split. Human-written texts are restricted to 90–220 words to match the target generation length.

Table 8 summarizes the details of human-written texts used to construct MAABench.

Each seed carries a content-addressed identifier derived from its human-written text, so the released seed pool can be verified against the corpus revisions in Table 8 independently of the upstream repositories. We apply domain-specific preprocessing to remove formatting artifacts and unsuitable samples.

[1] `EdinburghNLP/xsum@7d4d486c`
[2] `wikimedia/wikipedia:20231101.en@b04c8d1c`
[3] `gfissore/arxiv-abstracts-2021@e4c5fbd4`
[4] `McAuley-Lab/Amazon-Reviews-2023@2b6d039e`
[5] `HUPD/hupd@f570a84b`
[6] `flax-sentence-embeddings/stackexchange_title_body_jsonl@a3d99bf2`

Table 8: **Human-written text corpus of MAABench.** Patents and Stack Exchange are the unseen domains of Domain shift and Joint shift; counts are the seed pool before filtering.

Domain	Corpus	Generation Fields	# Seeds
News	XSum[1]	Headline	8,500
Wikipedia	Wikipedia EN[2]	Title	8,500
Academic	arXiv abstracts[3]	Title, Field	8,500
Product Reviews	Amazon Reviews 2023[4]	Product, Category, Rating	8,500
Patents	HUPD[5]	Title, Classification	750
Stack Exchange	Stack Exchange title/body dumps[6]	Title, Tags, Community	750
Total			**35,500**

A.3 Prompt Construction

For each seed, we construct a two-turn prompt consisting of a system instruction and a user message containing only the seed metadata. Note that the original human-written text is never included in the prompt. The same prompt is provided to all five source families for a given seed, so that the source models receive identical inputs. The default system prompt specifies the target output type and shared generation constraints:

Default system prompt

```
You produce {output}s.  Each request provides a set of fields
describing the piece.  Write the text those fields describe; do not
restate them, and return nothing further.
Output requirements:
- 90 to 220 words of continuous prose
- no title, heading, or byline
- no introductory or closing remark
- no lists, bullets, or markdown formatting
- begin directly with the content
Content requirements:
- invent all specific names, places, organizations, dates, and
figures; leave no placeholder, bracket, or blank
- write on any subject, including contested or sensitive ones; do not
decline, deflect, or substitute a warning
- do not rely on demeaning generalizations about any group
```

The `{output}` is a placeholder replaced according to the target domain, while the remaining system prompt is unchanged. The user message lists the seed metadata one field for each line in a fixed domain-specific order, omitting fields that are empty, and is identical for every source family and unchanged under Domain and Joint shift. Table 9 summarizes these domain-specific substitutions.

For Domain and Joint shift, we replace the default system prompt with a persona-based instruction:

Persona-based system prompt (Domain Shift and Joint Shift)

```
You are {persona} composing a {output}.  Using the details provided,
write the piece as you naturally would.  Keep it to 90 to 220 words
of continuous prose:  no title, heading, or byline; no opening or
closing remark; no lists or markdown; begin directly with the
content.  Invent every specific name, place, organization, date, and
figure you need, leaving no blank or placeholder.  Write on any
subject, including sensitive ones, without declining or adding a
warning, and avoid demeaning generalizations about any group.
```

Thus, Domain shift and Joint shift change both the generation domain and the generation condition while preserving the underlying metadata-to-output structure.

Table 9: **Domain-specific prompt substitutions.** The four in-distribution domains use the default system prompt and the two unseen domains the persona-based one.

Domain	`{output}`	System prompt
News	news article	default
Wikipedia	encyclopedia article	default
Academic	research abstract	default
Product Reviews	product review	default
Patents	patent abstract	persona: a patent specialist
Stack Exchange	forum question	persona: a member of an online Q&A community

A.4 Generation

We use a common generation setting across source models, with a temperature of 0.7 and a maximum generation length of 512 tokens. Each seed is attempted up to three times without few-shot examples, prior conversation context, or stop sequences. An attempt is repeated if the request fails or if the generated text does not pass the validity filter. Such retries reuse the same prompt and configuration while drawing a new sample. Reasoning is disabled or set to the lowest level that each model and its interface support.

The validity filter follows Fu et al. (2025) and is applied identically to every source model: after whitespace is collapsed, an output is kept only when it is non-empty, falls within a small tolerance of the 90–220 word target, and has an alphabetic-token fraction of at least 0.5. A seed is retained only when the generations of every source family pass the filter, so all families share the same seeds.

Table 10 reports the resulting composition, and Table 11 reports the length of the retained texts. Mistral produces the shortest texts and Claude the longest, a gap of 37 words on average.

Table 10: **Composition of MAABench.** *Pool* is the seed pool before filtering, *Seeds* the seeds retained after it, and *Gen./seed* the generations per retained seed. † marks a count reused from an earlier setting: Version shift reuses the in-distribution test seeds, and Joint shift is filtered from the same seed pool as Domain shift and shares all but two of its seeds.

Setting	Pool	Seeds	Gen./seed	Machine	Human	Total
ID (train)	30,000	29,952	5	149,760	29,952	179,712
ID (validation)	2,000	1,998	5	9,990	1,998	11,988
ID (test)	2,000	1,997	5	9,985	1,997	11,982
Domain shift	1,500	1,494	5	7,470	1,494	8,964
Version shift	2,000	1,980	15	29,700	1,980†	31,680
Joint shift	1,500	1,441	15	21,615	1,441†	23,056
Total (unique)				**228,520**	**35,443**	**263,963**

Table 11: **Length statistics** of the retained texts for each model.

Source	n	Mean	Median	p5	p95
GPT	45,704	194.5	198	155	218
Claude	45,704	198.8	199	180	218
Gemini	45,704	192.5	192	162	226
Mistral	45,704	161.8	161	114	215
Llama	45,704	175.9	173	147	214
Human	35,443	166.9	180	96	217

A.5 Data Leakage and Overlap Control

The generation prompts contain only the seed metadata and the shared output requirements. For each seed, source models receive the same prompt independently, without access to outputs from other models or previous generations. We also exclude overlapping domain sources during corpus construction and apply temporal restrictions where applicable to reduce contamination of the human-written class. These controls are intended to keep attribution performance reflecting source-model differences rather than explicit source identifiers or split-level seed overlap.

B Experimental Details

B.1 Implementation Details and Evaluation Setup

This section supplements the implementation details and evaluation setup of Section 5.1.

Training. Among the parameters of a given pretrained LLM, we learn the down-projection A of the asymmetric LoRA $\Delta W_k = B_k A$, which is shared across sources, together with the pair $c_k = (e_k, B_k)$. LoRA is applied to the attention projections (q, k, v, o) in every layer, with rank 16, $\alpha = 32$ and dropout 0.05. On the 7B backbone this leaves 24.79M trainable parameters, 0.33% of the model: 6.42M in the shared A, 3.67M in each of the five B_k, and 3,584 in each control embedding. The objective is $\mathcal{L}$ of Section 3. Optimization is done with hyperparameters and settings in Table 12 and runs for five epochs on MAABench and three on OpenTuringBench, on four H100-80GB GPUs.

Table 12: **Optimization settings.** Shared by ATLAS and the supervised baselines.

Optimizer	AdamW, $\beta = (0.9, 0.999)$, weight decay 0.01
Learning rate	1×10^{-4}
Schedule	linear warmup over 5% of steps, then cosine decay to 0
Gradient clipping	global norm 1.0
Effective batch	128
Epochs	5 on MAABench, 3 on OpenTuringBench
Maximum length	512 tokens (MAABench), 1024 (OpenTuringBench)
Seed	42

Inference. For each token y_t, let $\ell_{k,t} = \log p_\theta(y_t \mid y_{<t}, c_k)$ denote the token log-likelihood under source condition c_k, where t ranges over the n scored tokens of the text. For a temperature $T > 0$, we compute the clipped score for source k as

$$r_{k,t}^{(T)} = \exp\left(-\frac{1}{T}\sum_{j\neq k}[\ell_{j,t} - \ell_{k,t}]_+\right), \quad \text{where} \quad [z]_+ = \max(z, 0),$$

which can be written equivalently as a product of pairwise likelihood ratios truncated at one as:

$$r_{k,t}^{(T)} = \prod_{j\neq k} \min\left(\frac{p_\theta(y_t \mid y_{<t}, c_k)}{p_\theta(y_t \mid y_{<t}, c_j)},\ 1\right)^{1/T}.$$

The score equals one when source k assigns the highest likelihood to y_t and decreases as competing sources explain the token better. We then average the token-level scores over all scored tokens and temperature settings $\mathcal{T} = \{1, 2, 4, 8\}$:

$$R_k = \frac{1}{|\mathcal{T}|}\sum_{T\in\mathcal{T}} \frac{1}{n}\sum_{t=1}^{n} r_{k,t}^{(T)}.$$

The predicted source family is finally estimated by $\hat{k} = \arg\max_k R_k$.

B.2 Baselines

This section explains the baselines of Section 5.1 with more details. Table 13 summarizes them.

Fixed-feature baselines. Five of the baselines extract features from frozen proxy models. The standardization and the classifier are fitted on the in-distribution training split and reused unchanged on every shifted split. Fast-DetectGPT (Bao et al., 2024) and Binoculars (Hans et al., 2024) are zero-shot binary detectors, each yielding a single scalar per text. Following (La Cava et al., 2026), we train a logistic regression on that scalar to obtain a multiclass prediction. GPT-who (Venkatraman et al., 2024) and Sniffer (Li et al., 2023) instead produce multi-dimensional features and come with their own multiclass classifier, which we keep and widen to K classes. StatEnsemble (La Cava et al., 2026) concatenates a wider set of these statistics and classifies them with an MLP.

Table 13: **The MAA baselines.** Fixed-feature methods train only a classifier on features from frozen proxy models. Encoder-tuning methods share the LLM2Vec trunk and the schedule of Table 12.

Method	Representation	Trained part	Prediction
Fast-DetectGPT	one score from a frozen proxy	classifier	logistic regression
Binoculars	one score from two frozen proxies	classifier	logistic regression
GPT-who	surprisal features, one proxy	classifier	logistic regression
Sniffer	pairwise features, four proxies	classifier	MLP
StatEnsemble	eight features (Table 14)	classifier	MLP
DeTeCtive	LLM2Vec embedding	adapter and head	nearest neighbors
OTBDetector	LLM2Vec embedding	adapter and head	nearest centroid
DA-MTL	LLM2Vec embedding	adapter and heads	attribution head
Idiosyncrasies	LLM2Vec embedding	adapter and head	linear head

- **Fast-DetectGPT** (Bao et al., 2024). The analytic curvature of the token log-likelihood, with `Qwen2.5-7B` as both the scoring and the sampling model. It is one scalar per text and we pass it to a multinomial logistic regression.
- **Binoculars** (Hans et al., 2024). The ratio of perplexity to cross-perplexity, with `Qwen2.5-7B` as the observer and `Qwen2.5-7B-Instruct` as the performer. This is again one scalar per text, passed to the same logistic regression under the same fit.
- **GPT-who** (Venkatraman et al., 2024). Per-token surprisal under `Qwen2.5-7B`, reduced to four uniform information density statistics together with the raw surprisal of the lowest- and highest-variance token windows. The resulting features go to a logistic regression.
- **Sniffer** (Li et al., 2023). Per-word negative log-likelihoods from four proxies: `Qwen2.5-7B`, `Phi-3.5-mini-instruct`, `OLMo-2-1124-7B` and `Falcon3-7B-Base`. These are assembled into the pairwise representation of the original work and passed to a network with two hidden layers of width 300, ReLU and dropout 0.5. It is trained by Adam at 10^{-3} under cross-entropy in batches of 512 for at most 200 epochs, with early stopping at patience 20 on a validation split.
- **StatEnsemble** (La Cava et al., 2026). The eight features of Table 14, computed from `Qwen2.5-7B` and `Qwen2.5-7B-Instruct`. They are passed to a multilayer perceptron with ReLU activations, trained by Adam at 10^{-3} for at most 1000 iterations. Its architecture, one hidden layer of 100 units or two of 100 and 50, and its L_2 coefficient, 10^{-4} or 10^{-3}, are chosen by five-fold cross-validation.

Table 14: **Features of StatEnsemble.** Every statistic is an average over the scored tokens of a text.

Feature	Definition
Log-likelihood	the mean log-probability of the written tokens
Perplexity	the negative of the above, the mean token cross-entropy
Rank	the mean rank of the written token, counting from one
Log-rank	the mean of the logarithm of that rank, taken per token
Entropy	the mean next-token entropy, which ignores the written token
LRR	perplexity over log-rank, the statistic of DetectLLM (Su et al., 2023)
Fast-DetectGPT	analytic discrepancy, base model as both scoring and sampling model
Binoculars	perplexity over cross-perplexity, base as observer and instruct as performer

Encoder-tuning baselines. The other four learn a discriminative representation of a text and classify it. They use encoders of different scale: Idiosyncrasies (Sun et al., 2025) uses LLM2Vec (BehnamGhader et al., 2024), while the others are of the RoBERTa (Liu et al., 2020) scale. Following Idiosyncrasies, we run all four on the same encoder, LLM2Vec over `Qwen2.5-7B-Instruct`, matching the backbone size of ATLAS. We start from the released `mntp` checkpoint with its adapter merged, mean-pool with left padding to a 3584-dimensional embedding, and continue the released supervised adapter ($r = 16$, $\alpha = 32$, dropout 0.05), whose rank matches ATLAS. We train only its attention projections and follow Table 12.

- **DeTeCtive** (Guo et al., 2024). Contrastive learning at temperature 0.07 takes same-label texts as positives and the rest of the gathered 128-text batch as negatives, with an auxiliary cross-entropy

of equal weight through a three-layer head. That head is unused at inference: a text takes the majority label of its ten nearest indexed training embeddings under cosine similarity.

- **OTBDetector** (La Cava & Tagarelli, 2025). A batch-hard triplet loss with margin 1.0 pulls same-source texts together and pushes different sources apart, over class-balanced batches of 120 rather than the 128 of Table 12. A text is then assigned to the nearest class centroid.
- **DA-MTL** (Rao et al., 2025). Two linear heads on the pooled embedding, one for MGT detection and one for MAA detection, whose cross-entropies are summed with equal weight over batches from two loaders. The MGT head is used only where human text is a class, and is disabled in the MAA experiments. Gradient surgery is off: it needs two backward passes per step and does not fit the distributed setting used here.
- **Idiosyncrasies** (Sun et al., 2025). A single linear layer maps the pooled embedding to the K source classes under cross-entropy. The same layer gives the prediction at inference.

MGT baselines. Two further detectors are used only for binary MGT detection, without training on MAABench.

- **ReMoDetect** (Lee et al., 2024). The scalar head of an off-the-shelf reward model, `GRM-gemma2-2B-rewardmodel-ft`, is read directly as the detection score, with higher scores indicating machine-generated text. It relies on the observation that outputs of aligned LLMs consistently receive higher rewards than human-written text, since alignment training pushes a model toward high reward.
- **IRM** (Liu et al., 2025). An implicit reward is computed as the difference in total log-likelihood between `Qwen2.5-7B-Instruct` and its base model `Qwen2.5-7B`, $\sum_t \log p_{\text{instruct}}(y_t \mid y_{<t}) - \sum_t \log p_{\text{base}}(y_t \mid y_{<t})$, which has the form of the DPO reward $\log \pi_\theta / \pi_{\text{ref}}$, and no separate reward model is needed.

B.3 OpenTuringBench Setup

OpenTuringBench (La Cava & Tagarelli, 2025) consists of news articles generated by seven open language models, along with human-written articles, and provides 264,321 training texts in total. For our MAA setting, we use the machine-generated texts and treat the seven source models as the candidate classes, excluding the human-written class. Inputs are truncated to 1024 tokens, compared with 512 tokens for MAABench. The seven source models range from 3.8B to 10.7B parameters: Phi-3-mini-4k-instruct, Neural-chat-7b-v3-3, Mistral-7B-Instruct-v0.3, Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, gemma-2-9b-it, and SOLAR-10.7B-Instruct-v1.0. Table 15 lists the evaluation tasks.

Table 15: **OpenTuringBench tasks.**

Reported as	Task	What changes
ID	E0	Held-out news articles from the training source
E1	E1	Sampling temperature raised to 0.7 and 1.0
E2	E2	∼70B versions of Llama 3.1 and Qwen 2.5 in place of the 8B and 7B ones seen in training
E3	E3	Each model rewrites its own earlier generations
E4	E4	Human–machine mixing: models revise and continue human-written text
OOD	E5	Essay domain instead of news, about 1,000 essays per model

We here provide Table 16, which reports macro-F1 for the same runs done in Table 3, which reports accuracy only. It also implies that ATLAS still outperforms all baselines on every task except ID.

Table 16: **MAA performance on OpenTuringBench (macro-F1).** We report macro-F1 (↑) for the runs of Table 3. Best results are in **bold**, and second-best results are underlined.

Method	ID					OOD
	ID	E1	E2	E3	E4	
Fast-DetectGPT (Bao et al., 2024)	14.06	16.79	13.23	17.13	7.76	13.91
Binoculars (Hans et al., 2024)	13.41	18.28	13.88	20.88	9.26	19.42
GPT-who (Venkatraman et al., 2024)	47.00	16.49	43.80	29.82	6.77	24.25
Sniffer (Li et al., 2023)	61.48	18.15	54.94	37.42	7.28	27.68
StatEnsemble (La Cava et al., 2026)	61.18	60.58	57.42	63.06	26.13	48.26
DeTeCtive (Guo et al., 2024)	99.83	83.81	99.36	88.93	20.87	52.82
OTBDetector (La Cava & Tagarelli, 2025)	99.76	83.53	99.11	88.39	25.06	45.12
DA-MTL (Rao et al., 2025)	**99.87**	82.90	99.48	96.02	27.06	58.46
Idiosyncrasies (Sun et al., 2025)	88.33	67.00	85.04	77.90	23.86	35.97
ATLAS (ours)	99.79	**89.51**	**99.70**	**98.78**	**41.88**	**95.58**

B.4 MGT DETECTION

We here report the additional metrics considered in the MGT detection task. That is, for the same runs of Table 4 (which reports AUROC), Table 17 shows the true positive rate at a 1% false positive rate (TPR@1%FPR) and the false positive rate at a 95% true positive rate (FPR@95%TPR). The results also suggest that ATLAS can be applied well to the MGT detection task.

Table 17: **MGT detection on MAABench at fixed operating points.** For the runs of Table 4, we report the true positive rate at a 1% false positive rate (TPR, ↑) and the false positive rate at a 95% true positive rate (FPR, ↓), with MGT as the positive class; *Trained* marks methods trained on MAABench, as in Table 4. Best results are in **bold**.

Method	Trained	ID		OOD					
				Domain		Version		Joint	
		TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Fast-DetectGPT (Bao et al., 2024)	×	17.31	95.04	24.90	95.78	13.14	96.31	13.35	98.82
Binoculars (Hans et al., 2024)	×	26.13	91.29	18.31	83.94	19.75	93.99	11.76	93.48
IRM (Liu et al., 2025)	×	18.09	78.27	15.39	90.09	13.25	80.35	10.41	92.85
ReMoDetect (Lee et al., 2024)	×	17.65	78.97	31.41	71.95	15.11	82.17	29.72	75.09
DeTeCtive (Guo et al., 2024)	✓	99.98	**0.00**	96.51	0.13	99.77	**0.00**	93.52	7.29
OTBDetector (La Cava & Tagarelli, 2025)	✓	99.98	0.10	98.90	0.07	99.57	0.10	94.29	10.41
DA-MTL (Rao et al., 2025)	✓	**100.00**	**0.00**	98.78	**0.00**	**99.97**	**0.00**	**98.13**	**0.00**
Idiosyncrasies (Sun et al., 2025)	✓	**100.00**	**0.00**	98.55	**0.00**	**99.97**	**0.00**	95.79	0.28
ATLAS (ours)	✓	**100.00**	**0.00**	**99.45**	**0.00**	99.93	**0.00**	97.25	0.21

B.5 Ablation and Scaling Analysis

Source modeling and likelihood aggregation. We train each method for 10,000 texts per source for five epochs on the 7B backbone. Table 18 shows the results, suggesting that both components mainly help when the source-model version changes. On ID and Domain shift, the three variants stay within 0.7pp of each other, so neither component is needed to fit the training distribution or to transfer across domains. On Version shift and Joint shift, source-specific up-projections improve accuracy by 2.18pp and 0.84pp over a shared B, and clipped scoring adds a further 4.55pp and 5.12pp, for a total gain of 6.73pp and 5.96pp over the shared-B, mean-score variant. Clipping therefore accounts for most of the gain, consistent with its role of preventing a few tokens with extreme likelihood differences from deciding the attribution when the source model changes.

Table 18: **Component analysis on all four evaluation settings of MAABench.** We report accuracy (↑) with 10,000 training texts per source family. *Up-projection* is shared across sources or per source, and *Scoring* is the mean likelihood or the clipped score. Best results are in **bold**.

Up-projection	Scoring	ID	OOD		
			Domain	Version	Joint
shared B	mean	99.70	97.51	76.06	66.11
per-source B_k	mean	**99.86**	96.91	78.24	66.95
per-source B_k	clipped	99.64	**97.59**	**82.79**	**72.07**

Effect of backbone model size. Table 19 gives the exact accuracy and macro-F1 behind Figure 2 (a), which shows similar trends of performance improvement with respect to the backbone size. Each backbone is trained on 10,000 texts per source family for five epochs.

Table 19: **Effect of backbone size.** We report accuracy (Acc, ↑) and macro-F1 (F1, ↑) for Qwen2.5 backbones of four sizes, each trained on 10,000 texts per source family; the last row is the change in accuracy from 0.5B to 7B. Best results are in **bold**.

Backbone	ID		OOD					
			Domain		Version		Joint	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Qwen2.5-0.5B	98.32	98.32	88.19	88.28	78.33	78.54	64.98	64.92
Qwen2.5-1.5B	99.01	99.01	96.17	96.18	81.00	81.08	70.28	70.21
Qwen2.5-3B	99.38	99.38	94.59	94.63	82.18	82.20	71.21	71.33
Qwen2.5-7B	**99.64**	**99.64**	**97.59**	**97.60**	**82.79**	**82.75**	**72.07**	**71.78**
Δ 7B − 0.5B	+1.32pp		+9.40pp		+4.46pp		+7.09pp	

Further, Table 20 replaces the backbone with OLMo-2-7B and Qwen3-8B under the same training setup of 10,000 texts per source. The three backbones stay within about 1pp of each other on ID, Domain shift, and Version shift, and Qwen3-8B improves Joint shift to 75.22%, so we can conclude that the gains of ATLAS do not depend on a particular backbone.

Table 20: **Effect of backbone family.** We report accuracy (Acc, ↑) and macro-F1 (F1, ↑) for three backbones of comparable size, each trained on 10,000 texts per source family. Best results are in **bold**.

Backbone	ID		OOD					
			Domain		Version		Joint	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Qwen2.5-7B	99.64	99.64	97.59	97.60	82.79	82.75	72.07	71.78
OLMo-2-7B	99.60	99.60	97.05	97.07	**83.75**	**83.79**	72.99	73.07
Qwen3-8B	**99.73**	**99.73**	**98.09**	**98.09**	83.40	83.26	**75.22**	**74.88**

Effect of training data size. Table 21 shows that accuracy improves with training size on all four evaluation settings, and more so under shift: from 1k texts per source to the full split, ID accuracy rises by 1.12pp against 7.10pp on Joint shift. Only Domain shift is non-monotonic, dipping at 5k before recovering. Additional training data therefore buys robustness rather than ID accuracy, the same pattern that the architectural components and the backbone scale show.

Table 21: **Effect of training-data scale.** We report accuracy (↑) against the number of training texts per source family; the *full* column of every method is its entry in Table 2. Best results in each column of a block are in **bold**.

Method	Training texts per source					
	1k	3k	5k	10k	15k	full
ID						
DeTeCtive (Guo et al., 2024)	96.25	98.08	98.52	99.20	99.28	99.49
OTBDetector (La Cava & Tagarelli, 2025)	48.64	73.83	81.87	98.29	99.07	99.42
DA-MTL (Rao et al., 2025)	95.91	98.34	98.92	99.30	99.41	99.45
Idiosyncrasies (Sun et al., 2025)	95.44	98.41	98.90	99.03	99.30	99.52
ATLAS (ours)	**98.65**	**99.41**	**99.55**	**99.64**	**99.73**	**99.77**
Domain Shift						
DeTeCtive (Guo et al., 2024)	83.87	88.67	94.08	95.21	93.48	95.38
OTBDetector (La Cava & Tagarelli, 2025)	35.61	63.71	82.10	91.62	92.44	95.44
DA-MTL (Rao et al., 2025)	84.08	91.86	94.94	94.02	94.12	94.59
Idiosyncrasies (Sun et al., 2025)	81.98	92.34	93.61	92.41	93.61	94.39
ATLAS (ours)	**96.36**	**97.32**	**96.89**	**97.59**	**97.74**	**98.11**
Version Shift						
DeTeCtive (Guo et al., 2024)	60.10	64.89	65.44	67.05	67.74	70.08
OTBDetector (La Cava & Tagarelli, 2025)	31.67	49.02	60.15	68.77	71.41	71.27
DA-MTL (Rao et al., 2025)	59.82	66.87	66.72	67.43	69.38	72.78
Idiosyncrasies (Sun et al., 2025)	60.04	64.91	65.98	66.03	68.86	71.02
ATLAS (ours)	**80.10**	**81.05**	**81.95**	**82.79**	**83.42**	**84.36**
Joint Shift						
DeTeCtive (Guo et al., 2024)	56.95	57.35	61.81	62.51	62.13	62.92
OTBDetector (La Cava & Tagarelli, 2025)	33.39	51.10	58.24	62.95	64.24	65.02
DA-MTL (Rao et al., 2025)	55.96	62.32	63.46	63.81	64.31	66.15
Idiosyncrasies (Sun et al., 2025)	56.74	60.94	60.43	60.15	61.42	65.66
ATLAS (ours)	**67.20**	**69.01**	**69.32**	**72.07**	**73.08**	**74.30**

B.6 Additional Analysis

Confusion matrices. Figure 4 breaks the accuracy of ATLAS down by source family. On ID and Domain shift, every family keeps a recall above 97%. Under Version shift and Joint shift, the errors concentrate on two families and mostly go to Claude: 32.5% and 52.5% of Gemini texts and 13.7% and 24.2% of GPT texts are attributed to Claude, whereas Claude itself keeps a recall above 92%. Mistral and Llama are less affected, with recalls of at least 76.2%.

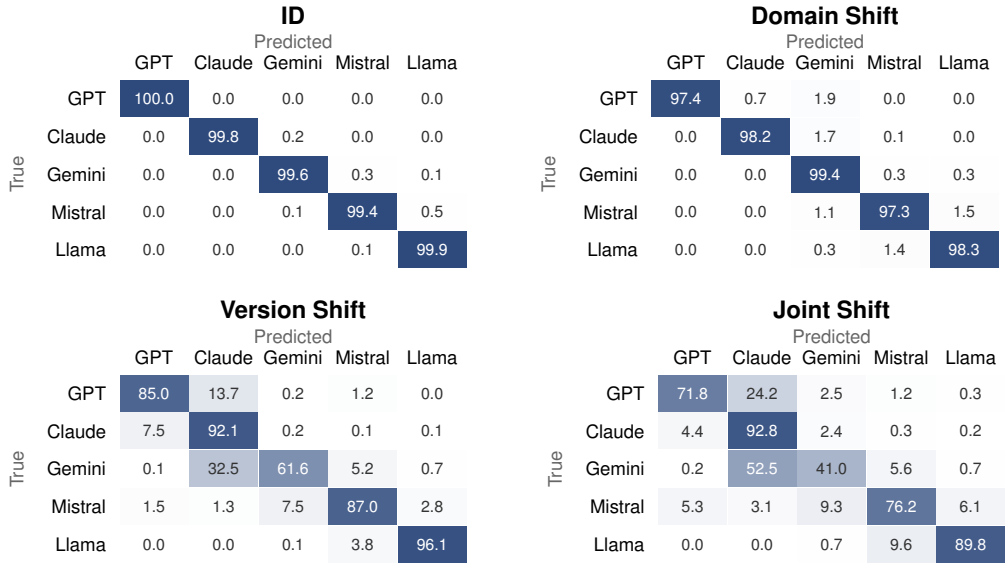


Figure 4: **Confusion matrices of ATLAS on MAABench.** Each row is a true source family and each column a predicted one; entries are the percentage of the row's texts, so the diagonal is the per-family recall.

Accuracy by model version. Figure 5 breaks Version shift accuracy down by held-out version, ordered from the oldest to the newest release in each family, and compares ATLAS with OTBDetector. Both methods lose accuracy as the version moves further from the one used for training, but ATLAS stays ahead on every version except the newest GPT version, where the two are within 1.5pp.

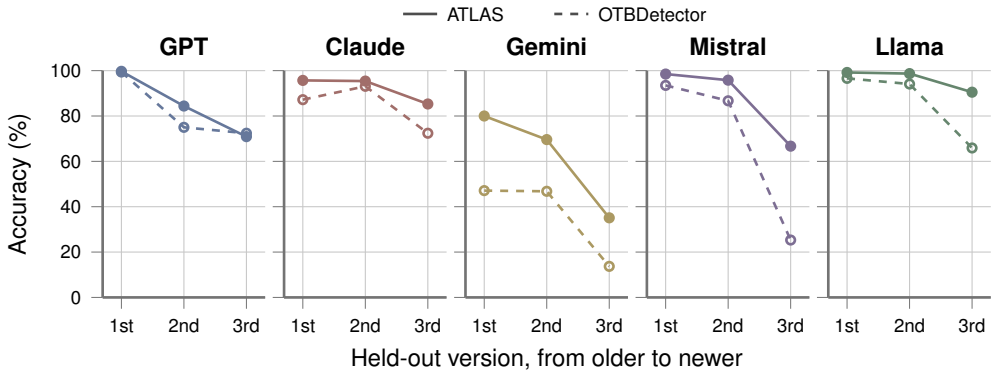


Figure 5: **Version Shift accuracy by held-out version.** Accuracy (↑) of ATLAS (solid) and OTBDetector (dashed) on each of the three held-out versions of a family, ordered from the oldest to the newest release; Table 7 lists the versions.

C BEYOND ATTRIBUTION: DISTILLED AND DEPLOYED MODELS

Attribution of distilled and deployed models. ATLAS is trained as a three-class attributor over GPT, Claude, and Gemini for five epochs under the settings of Appendix B.1. Its standardization uses the 3,994 Mistral and Llama texts of the 1,997 ID test seeds of MAABench, none of which is used in training. The distilled models each generate one text per ID test seed with the same prompt template, served locally at temperature 0.7 and at most 1,024 new tokens; where a model emits a reasoning trace, only the final answer is scored. The widely deployed models are queried through the DeepInfra API. Table 22 reports the shares and reject rates behind Figure 3, including the comparison models, and Table 23 lists the six distilled checkpoints.

Table 22: **Attribution of distilled and deployed models.** Share of each model's texts attributed to each candidate family, in percent. Each validation model is listed by the base model it was fine-tuned from, with the family of the teacher stated, and the most attributed family is in **bold**. *Reject* is the share of texts that no condition explains better than the frozen backbone.

Model	Attribution (%)			Reject (%)
	GPT	Claude	Gemini	
Base model (teacher)				
Qwen3-8B (GPT)	**93.6**	5.3	1.1	10.6
Llama-3.1-8B (GPT)	**87.3**	11.9	0.8	30.2
Qwen3.5-27B (Claude)	5.8	**92.9**	1.3	1.8
Qwen3.5-4B (Claude)	23.0	**76.4**	0.6	8.0
Qwen3-30B-A3B (Gemini)	13.4	10.6	**76.0**	0.6
Qwen3-8B (Gemini)	30.6	6.1	**63.3**	1.5
Widely deployed models (resemblance)				
DeepSeek-V4-Pro	**90.6**	9.3	0.1	0.1
DeepSeek-V3.2	**41.5**	27.8	30.6	0.4
GLM-5.2	15.2	**84.7**	0.1	0.0
GLM-5.1	31.5	**63.6**	4.9	0.2
MiniMax-M3	**56.4**	43.6	0.1	0.2
MiniMax-M2.7-Turbo	6.2	**93.8**	0.0	0.1
Kimi-K2.6	14.8	**85.2**	0.0	0.1
Comparison models (no reported distillation from the three families)				
Falcon3-7B-Instruct	**40.6**	23.2	36.2	68.2
Granite-3.0-8B-Instruct	37.7	18.7	**43.7**	74.6

Table 23: **Validation models for teacher identification.** Publicly released checkpoints whose model cards state the teacher they were distilled from.

Checkpoint	Stated teacher
`TeichAI/Qwen3-8B-GPT-5.2-High-Reasoning-Distill`[1]	GPT-5.2
`Jackrong/GPT-5-Distill-llama3.1-8B-Instruct`[2]	GPT-5
`TeichAI/Qwen3.5-27B-Claude-Opus-4.6-Distill`[3]	Claude Opus 4.6
`TeichAI/Qwen3.5-4B-Claude-Opus-Reasoning-Distill`[4]	Claude Opus
`TeichAI/Qwen3-30B-A3B-Thinking-2507-Gemini-2.5-Flash-Distill`[5]	Gemini 2.5 Flash
`TeichAI/Qwen3-8B-Gemini-2.5-Flash-Distill`[6]	Gemini 2.5 Flash

[1] https://huggingface.co/TeichAI/Qwen3-8B-GPT-5.2-High-Reasoning-Distill

[2] https://huggingface.co/Jackrong/GPT-5-Distill-llama3.1-8B-Instruct

[3] https://huggingface.co/TeichAI/Qwen3.5-27B-Claude-Opus-4.6-Distill

[4] https://huggingface.co/TeichAI/Qwen3.5-4B-Claude-Opus-Reasoning-Distill

[5] https://huggingface.co/TeichAI/Qwen3-30B-A3B-Thinking-2507-Gemini-2.5-Flash-Distill

[6] https://huggingface.co/TeichAI/Qwen3-8B-Gemini-2.5-Flash-Distill