

When New Generators Arrive: Lifelong Machine-Generated Text Attribution via Ridge Feature Transfer

Zhen Sun^{♣♦♠*} Yifan Liao[♠] Zhicong Huang^{♦†} Jiaheng Wei[♠]
Cheng Hong[♦] Yutao Yue^{♠♥} Xinlei He^{♠†}

[♣]Wuhan University [♦]Ant Group

[♠]The Hong Kong University of Science and Technology (Guangzhou)

[♥]Institute of Deep Perception Technology, JITRI

Abstract

Machine-generated text (MGT) attribution aims to identify the specific generator responsible for a given text, thereby providing fine-grained evidence for model accountability and misuse investigation. As new large language models continue to emerge, attribution models must continuously incorporate new generators while preserving their ability to recognize previously seen ones. Prior works have shown that this lifelong MGT attribution setting is challenging, and existing methods often struggle to achieve a stable balance between adapting to new classes and retaining old ones. To address this issue, we propose *RidgeFT*, a lightweight analytic update framework that does not rely on exemplar replay. *RidgeFT* trains a task-aware encoder on the initial generator set, stores compact class-wise sufficient statistics when each generator class is first observed, and then freezes the encoder for replay-free closed-form updates. It then suppresses generator-irrelevant variation through covariance calibration, improves representation capacity with fixed random features, and updates new classes through closed-form ridge regression based on class-level sufficient statistics. Across multi-topic evaluations with varying initial generator setups, *RidgeFT* consistently outperforms baselines. It achieves the best overall macro-F1 across domains, backbones, and incremental protocols, while also improving both old-class retention and new-class adaptation. These results suggest that feature-stable analytic updates provide a simple yet effective approach to lifelong MGT attribution.

1 Introduction

As generative tools powered by large language models (LLMs) become increasingly

*Work done during an internship at Ant Group.

†Corresponding authors: Zhicong Huang (zhicong.hzc@antgroup.com) and Xinlei He (xinlei.he@whu.edu.cn).

Our code is available at <https://github.com/Vincent-HKUSTGZ/RidgeFT-Lifelong-MGT>.

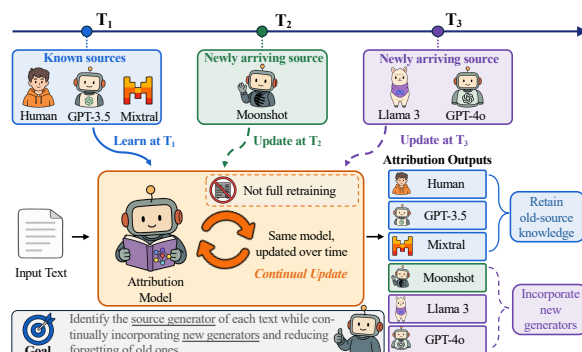


Figure 1: Illustration of the lifelong MGT attribution setting.

widespread (Oyelude, 2024; OpenClaw, 2026), users can now conveniently rely on them for text generation and polishing. While these capabilities improve writing efficiency, they also introduce potential risks of misuse (Kumarage et al., 2024). For example, users may exploit LLMs to automatically produce large volumes of papers, news articles, reviews, and other forms of text, thereby disrupting the normal order of content production and undermining the credibility of human-authored writing (Wu et al., 2025). Against this backdrop, the effective identification of machine-generated text (MGT) has become an important problem. To address this issue, existing studies have primarily focused on two aspects of MGT identification: binary detection and source attribution (Wu et al., 2025; He et al., 2024). Compared with binary detection, MGT attribution further aims to identify the specific source generator, thereby providing finer-grained evidence for accountability tracking and misuse investigation (Cava and Tagarelli, 2025).

Existing MGT attribution methods typically assume a fixed set of generators, which is often an unrealistic assumption in real-world settings. Since the attribution systems operate in a dynamic and open generator space in practice, they should not only recognize newly emerging generators, but also

preserve the ability to distinguish previously seen ones. Liu et al. (2025) introduce a more realistic setting, known as class-incremental MGT attribution, which we refer to as lifelong MGT attribution throughout this paper and illustrate in Figure 1. Under this setting, the attribution model needs to be continuously updated as new generator classes arrive over time. However, due to factors such as computational cost, data licensing constraints, or the unavailability of historical data, it is usually impractical to recollect all past data and retrain the model from scratch (Verwimp et al., 2024; Huang et al., 2024). At the same time, directly updating the model using only data from new classes often leads to catastrophic forgetting (McCloskey and Cohen, 1989; French, 1999). Therefore, a central challenge in lifelong MGT attribution is efficiently incorporating new generator classes under limited data while maintaining stable recognition of previously learned generators.

Under this challenge, we posit that the difficulty of lifelong MGT attribution does not necessarily need to be addressed by continuously updating the entire text encoder. Instead, a task-tuned encoder trained on the initial set of generators can already capture strong generator-related representations. If this encoder continues to be fine-tuned during the incremental stage, the representation space will continue to shift as new classes arrive, thereby making the decision boundaries of old classes unstable (Caccia et al., 2022; Yu et al., 2020). We therefore seek to decouple the learning of new generators from deep representation updating, ensuring that incremental knowledge is absorbed without altering or disrupting the stable representation space. Motivated by this idea, we propose *RidgeFT*, an exemplar-free analytic update framework for lifelong MGT attribution. We consider a practical deployment scenario where the attribution system is trained and maintained from the initial stage. During initialization, *RidgeFT* uses the initial-class data to train a task-aware encoder and construct compact class-wise sufficient statistics; after that, the raw texts of old classes are discarded and never replayed. When new generators arrive, *RidgeFT* keeps the encoder frozen, maps the new data through covariance calibration and fixed random features, accumulates the corresponding statistics, and updates the classifier by a closed-form ridge solution. In this way, incremental learning is performed through statistical memory rather than historical-text replay or repeated en-

coder fine-tuning. We evaluate *RidgeFT* in a multi-topic setting (Liu et al., 2025; Sun et al., 2025), using P3, P4, and P5 protocols that start from 3, 4, and 5 initial classes and incrementally add 3, 2, and 1 new generator classes, respectively. Under the standard P5 protocol, *RidgeFT* achieves 0.886 full-F1, 0.902 old-class F1, and 0.804 new-class F1, improving full-F1 by 0.037 over the strongest continual-learning baseline.

Our contributions are summarized as follows:

- We identify lifelong MGT attribution as a generator-evolving attribution problem, where the model must preserve generator-specific decision boundaries while suppressing topic-, domain-, and length-induced nuisance variation under an exemplar-free update constraint.
- We propose *RidgeFT*, an exemplar-free analytic update framework that combines fractional covariance calibration, isotropic random feature lifting, and class-balanced closed-form ridge regression, enabling incremental updates using only compact class-wise sufficient statistics.
- We conduct extensive experiments across multiple topics, multiple target generators, and two backbones. The results show that *RidgeFT*’s main advantage lies in substantially improving new-generator adaptation while maintaining competitive old-generator retention.

2 Related Work

Establishing effective regulatory mechanisms for MGT has become essential for maintaining content credibility and supporting platform governance (Wu et al., 2025). Recent research has pushed MGT detection from idealized binary classification toward more complex real-world scenarios. Studies have not only developed dynamic benchmarks accounting for multilingual settings and model evolution (Macko et al., 2025; Yu et al., 2025), but also improved detector generalization to unseen generators and domains (Hao et al., 2025; Jiao et al., 2025; Chen et al., 2025), alongside enhancing robustness against adversarial attacks (Li et al., 2025b,a; Pedrotti et al., 2025). Additionally, phenomena with blurred boundaries, such as human-AI collaborative writing, have also become relevant to platform monitoring (Su et al., 2025; Saha and Feizi, 2025; Sun et al., 2025). However,

although MGT detection techniques have made progress in addressing realistic challenges, merely determining whether a text is machine-generated is no longer sufficient to satisfy the growing demands for accountability tracing and copyright attribution.

Compared with binary detection, MGT attribution further requires identifying the specific generator responsible for a text, providing critical evidence for model accountability and forensic analysis (He et al., 2024; Fang et al., 2025), and has emerged as an important branch of authorship attribution (Huang et al., 2025). While prior studies have explored practical attribution settings (Sarvazyan et al., 2023; Cava et al., 2026; Najjar et al., 2025), they mostly consider static scenarios where the candidate generator set is fixed. In real-world deployment, rapid LLM updates require attribution models to evolve accordingly to recognize new generators. Simultaneously, data privacy and licensing restrictions often prevent full retraining on historical data. To address this, Liu et al. (2025) pioneered class-incremental MGT attribution and evaluated mainstream continual learning methods. Nevertheless, existing approaches still struggle to reliably preserve recognition performance on previously seen generators while learning the characteristics of new ones. Breaking this trade-off between adaptation to new classes and retention of old ones in lifelong MGT attribution remains an unresolved challenge, which is the central problem addressed in this work.

3 Methodology

In lifelong MGT attribution, continuously fine-tuning the text encoder on new generators alters the representation geometry of old classes, thereby degrading previous decision boundaries. To address this, we propose *RidgeFT*, which reformulates lifelong attribution as an analytic ridge regression problem based on sufficient statistics. *RidgeFT* is designed for an attribution system that is trained and maintained from the initial stage, so the required statistics of initial classes can be recorded when those classes are first available. By freezing the base text encoder, *RidgeFT* preserves prior knowledge and performs analytic updates exclusively on the extracted features. Given an input text x , its frozen representation $h = f_\theta(x)$ is mapped to the final prediction via a novel sequential pipeline: covariance calibration, isotropic random feature lifting, and class-balanced ridge re-

gression ($h \rightarrow \tilde{h} \rightarrow z(x) \rightarrow \hat{y}$). The calibration transform and the initial sufficient statistics are computed once from the base-stage training data and then stored. During later incremental stages, *RidgeFT* only processes newly arriving class data and does not revisit old raw texts.

Covariance Calibration. Base representations may encode generator-irrelevant variations (e.g., topic, length, domain) as high-variance directions, which impairs subsequent inner-product classifiers. To mitigate this, *RidgeFT* applies a mild fractional whitening transformation to suppress within-class noise while preserving the original discriminative geometry. This transform is estimated only from the base-stage training representations and is kept fixed after initialization, so incremental updates do not require replaying previous raw texts.

First, we calculate the within-class scatter matrix S_w :

$$S_w = \frac{1}{N - C_0} \sum_{c=1}^{C_0} \sum_{i:y_i=c} (h_i - \mu_c)(h_i - \mu_c)^\top, \quad (1)$$

where N is the total number of base samples, C_0 is the number of base classes, h_i is the representation of the i -th sample, and μ_c is the feature mean of class c . To address the instability of high-dimensional covariance estimation, we apply trace-scaled shrinkage:

$$S_w^{\text{shrink}} = (1 - \alpha)S_w + \alpha \frac{\text{tr}(S_w)}{d_h} I_{d_h}, \quad (2)$$

where α is the shrinkage parameter, $\text{tr}(\cdot)$ denotes the matrix trace, d_h is the feature dimension, and I_{d_h} is the identity matrix. Because d_h is large relative to the per-class sample size, the empirical scatter S_w is poorly conditioned, and directly whitening its near-zero eigenvalues would amplify estimation noise. Following Ledoit and Wolf (2004), we pull S_w toward the scaled identity $\frac{\text{tr}(S_w)}{d_h} I_{d_h}$, keeping S_w^{shrink} well-conditioned and invertible while preserving its trace. Finally, we compute the fractionally whitened representation $\tilde{h}$ via the eigenvalue decomposition of S_w^{shrink} :

$$S_w^{\text{shrink}} = U \Lambda U^\top, \Lambda = \text{diag}(\sigma_1, \dots, \sigma_{d_h}). \quad (3)$$

Given the eigendecomposition of S_w^{shrink} , we calibrate each principal direction according to its estimated within-class variance:

$$P_\delta = U(\Lambda + \epsilon I_{d_h})^{-\delta} U^\top, \tilde{h} = P_\delta(h - \mu). \quad (4)$$

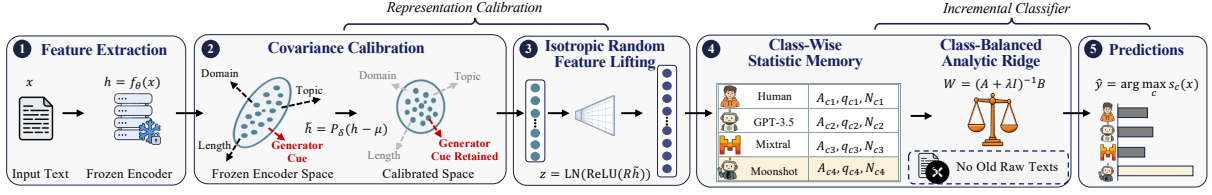


Figure 2: Overview of *RidgeFT*.

Here, U and Λ represent the eigenvectors and eigenvalues of S_w^{shrink} , respectively, with σ_j denoting the j -th eigenvalue. The parameter $\delta \in [0, 1]$ controls the calibration strength, μ is the global feature mean, and ϵ is a small constant ensuring numerical stability. At $\delta = 0.5$ this recovers conventional whitening of the trace-shrunk within-class scatter, whereas $\delta = 0$ applies no calibration and $\delta = 1$ corresponds to stronger inverse-covariance scaling. A moderate exponent attenuates dominant within-class variations without excessively distorting the original feature space, providing a stable input for the downstream analytic classifier. As exemplified in Figure 2, this calibration is intended to reduce nuisance directions associated with topic, length, and domain, while preserving generator-related cues for attribution.

Isotropic Random Feature Lifting. After covariance calibration, *RidgeFT* further lifts the calibrated representations into a fixed nonlinear random feature space to boost the expressive capacity of the analytic classifier without sacrificing closed-form incremental updates. By adopting a non-trainable random mapping (Rahimi and Recht, 2007) instead of a learnable projection layer, we ensure the feature basis remains strictly invariant upon the arrival of new generators.

Let $\tilde{h} \in \mathbb{R}^{d_h}$ denote the calibrated encoder representation, where d_h is the feature dimension. *RidgeFT* samples a Gaussian random matrix $R \in \mathbb{R}^{d_\phi \times d_h}$ once before incremental learning, with each entry drawn independently as $R_{ij} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/d_h)$. The lifted feature is computed as

$$z(x) = \text{LN}(\text{ReLU}(R\tilde{h})) \in \mathbb{R}^{d_\phi}, \quad (5)$$

where d_ϕ is the random feature dimension and $\text{LN}(\cdot)$ denotes parameter-free layer normalization without learnable affine parameters. The matrix R is fixed throughout all incremental stages, so only the sufficient statistics of the ridge classifier need to be updated.

We use an isotropic Gaussian projection rather than a data-dependent projection learned from base

classes. In lifelong MGT attribution, future generators may introduce variation directions that are not present in the initial generator set. A projection fitted to base-class geometry may therefore overemphasize existing class separations and reduce coverage of unseen generator-specific directions. By contrast, isotropic random features provide a task-agnostic nonlinear basis that expands the calibrated representation more uniformly, enabling the downstream ridge regressor to form richer decision boundaries without changing the representation space during continual learning.

Class-Balanced Analytic Ridge Regression. Using the random feature representation $z(x)$, *RidgeFT* trains the classification head via closed-form ridge regression. Because this analytic approach relies only on first- and second-order sufficient statistics, unlike iterative SGD-based methods, we only update the sufficient statistics for each class during the incremental stage, strictly eliminating the need to store raw text from old classes.

For each class c , *RidgeFT* stores only three sufficient statistics in the random feature space: the second-order statistic $A_c = \sum_{i:y_i=c} z_i z_i^\top$, the first-order statistic $q_c = \sum_{i:y_i=c} z_i$, and the sample count N_c . These statistics are computed when class c is first observed and are then retained as statistical memory, so later updates do not require replaying its raw texts. When a new generator arrives, *RidgeFT* only computes the same statistics for the new class and appends them to the existing table. However, directly summing class statistics can bias the ridge solution toward classes with larger sample counts, since both A_c and q_c scale with N_c . To reduce this imbalance, *RidgeFT* introduces inverse-frequency weights based on class sample counts:

$$\omega_c = \frac{(N_c + \tau)^{-\beta}}{\frac{1}{|\mathcal{Y}_t|} \sum_{c' \in \mathcal{Y}_t} (N_{c'} + \tau)^{-\beta}}, \quad c \in \mathcal{Y}_t, \quad (6)$$

where $\mathcal{Y}_t$ is the set of classes observed up to incremental stage t , β controls the strength of reweighting, and τ prevents extremely small classes from receiving excessively large weights.

We then construct the ridge regression solution using the weighted statistics. Specifically, let $\bar{A} = \sum_{c \in \mathcal{Y}_t} \omega_c A_c$, and let the c -th column of $\bar{B}$ be $\omega_c q_c$. The final classifier is given by

$$W = (\bar{A} + \lambda I_{d_\phi})^{-1} \bar{B}, \quad (7)$$

where $\lambda > 0$ is the ridge regularization coefficient.

By mitigating the dominance of majority classes via reweighting, *RidgeFT* converts incremental updates into the simple accumulation of sufficient statistics and a single linear system solve. This enables the efficient absorption of new generators without encoder drift. At inference, the model predicts the class maximizing $s(x) = z(x)^\top W$. Algorithm 1 summarizes this overall procedure.

Algorithm 1 *RidgeFT* Update

Require: Frozen encoder f_θ , calibration transform P_δ , random matrix R , streams $\{\mathcal{D}_t\}_{t=0}^T$

Ensure: Ridge classifier W

```

1: Initialize seen classes  $\mathcal{Y}_{-1} \leftarrow \emptyset$  and statistics  $\{A_c, q_c, N_c\}$  as empty.
2: for  $t = 0, \dots, T$  do
3:    $\mathcal{Y}_t \leftarrow \mathcal{Y}_{t-1}$ .
4:   for each  $(x_i, y_i) \in \mathcal{D}_t$  do
5:     if  $y_i \notin \mathcal{Y}_t$  then
6:       Add  $y_i$  to  $\mathcal{Y}_t$  and initialize  $A_{y_i}, q_{y_i}, N_{y_i}$  as zero.
7:     end if
8:      $z_i \leftarrow \text{LN}(\text{ReLU}(RP_\delta(f_\theta(x_i) - \mu)))$ .
9:      $A_{y_i} \leftarrow A_{y_i} + z_i z_i^\top, \quad q_{y_i} \leftarrow q_{y_i} + z_i, \quad N_{y_i} \leftarrow N_{y_i} + 1$ .
10:  end for
11:   $\omega_c \leftarrow \frac{(N_c + \tau)^{-\beta}}{|\mathcal{Y}_t|^{-1} \sum_{c' \in \mathcal{Y}_t} (N_{c'} + \tau)^{-\beta}}, \quad \forall c \in \mathcal{Y}_t$ .
12:   $\bar{A} \leftarrow \sum_{c \in \mathcal{Y}_t} \omega_c A_c, \quad \bar{B} \leftarrow [\omega_c q_c]_{c \in \mathcal{Y}_t}$ .
13:   $W \leftarrow (\bar{A} + \lambda I_{d_\phi})^{-1} \bar{B}$ .
14: end for
15: return  $W$ 

```

4 Experimental Setting

Task Protocol. To simulate the continual emergence of generators in real-world scenarios, we adopt a class-incremental MGT attribution benchmark targeting 1 Human source and 5 LLMs. We evaluate our method under 3 incremental protocols: P3, P4, and P5, which begin with 3, 4, and 5 initial classes, respectively. While P5 follows the

standard single-step incremental setting of prior work (Liu et al., 2025), P3 and P4 are introduced to further increase task difficulty. By starting with smaller base sets and introducing one new class per subsequent step, P3 and P4 involve more incremental stages, thereby imposing substantially stricter requirements on the model’s resistance to catastrophic forgetting.

Dataset. We conduct experiments on two representative benchmark datasets, which correspond to two major real-world scenarios where LLM misuse poses particularly significant risks: rigorous academic writing and diverse social media interaction. (1) MGT-Academic (Liu et al., 2025): This dataset focuses on the academic writing domain and covers three disciplines, namely STEM, Humanities, and Social Science. It includes human-authored text as well as text generated by five mainstream models, namely GPT-3.5¹, GPT-4o-mini (OpenAI, 2024), Moonshot (Moonshot AI, 2026), Mixtral-8x7B (Jiang et al., 2024), and Llama-3.1 (Team, 2024). We use the complete dataset released by the original authors in our experiments. (2) AIGT-Bench (Social Media Subset) (Sun et al., 2025): The texts in this subset are drawn from major social media platforms. It likewise contains human-written text and text generated by GPT-3.5, GPT-4o-mini, Llama-1 (Touvron et al., 2023a), Llama-2 (Touvron et al., 2023b), and Llama-3.1. To ensure rigorous evaluation and class balance, we uniformly sample $15K$ instances from each class for our experiments.

Baselines & Implementation. We follow the experimental protocol of Liu et al. (2025) and compare our method against several representative baselines. We include the classical methods LwF (Li and Hoiem, 2017), iCaRL (Rebuffi et al., 2017), and BiC (Wu et al., 2019), as well as the more recent EASE (Zhou et al., 2024), PASS (Zhu et al., 2021) and SimpleCIL (Zhou et al., 2025). Following Liu et al. (2025), we equip LwF, iCaRL, BiC, and EASE with a replay buffer of 100 stored exemplars per old class, whereas PASS and SimpleCIL remain exemplar-free. As for the backbones, we adopt DeBERTa-v3-base (He et al., 2023) and RoBERTa-base (Liu et al., 2019) as the text encoders for feature extraction.

Evaluation Metrics. All methods use the same data splits and incremental protocols. *RidgeFT* stores no raw exemplars, and its statistical-state

¹<https://openai.com/index/chatgpt>.

Table 1: Performance Comparison across Domains and Models. * denotes methods using data replay. “Ori.” denotes the original base-stage model before incremental updating, i.e., the S0 performance on the initial generator classes, and is reported only as a reference rather than as a continual-learning baseline. Darker color indicates better performance. Bold values indicate the best result among continual-learning methods within each backbone block.

Domain	New Model	RoBERTa-base								DeBERTa-base							
		Ori.	LwF*	iCaRL*	BiC*	EASE*	PASS	SimpleCIL	<i>RidgeFT</i> (Ours)	Ori.	LwF*	iCaRL*	BiC*	EASE*	PASS	SimpleCIL	<i>RidgeFT</i> (Ours)
Social Science	GPT-3.5	0.906	0.833	0.845	0.835	0.844	0.772	0.769	0.865	0.891	0.819	0.812	0.801	0.753	0.780	0.765	0.851
	Mixtral	0.938	0.845	0.837	0.851	0.845	0.801	0.771	0.880	0.938	0.841	0.830	0.833	0.758	0.785	0.773	0.878
	Moonshot	0.931	0.847	0.843	0.840	0.816	0.780	0.792	0.868	0.922	0.835	0.829	0.827	0.788	0.784	0.778	0.862
	Llama-3.1	0.920	0.820	0.806	0.814	0.811	0.761	0.751	0.854	0.912	0.785	0.790	0.784	0.753	0.750	0.739	0.853
	GPT-4o-mini	0.905	0.830	0.810	0.838	0.810	0.768	0.790	0.868	0.898	0.834	0.823	0.823	0.829	0.795	0.765	0.867
	Average	0.920	0.835	0.828	0.836	0.825	0.777	0.775	0.867	0.912	0.823	0.817	0.814	0.776	0.779	0.764	0.862
STEM	GPT-3.5	0.946	0.882	0.880	0.883	0.867	0.802	0.805	0.914	0.942	0.847	0.850	0.855	0.847	0.808	0.801	0.907
	Mixtral	0.964	0.887	0.883	0.878	0.874	0.830	0.809	0.912	0.966	0.879	0.877	0.875	0.867	0.813	0.822	0.919
	Moonshot	0.959	0.895	0.880	0.891	0.885	0.841	0.830	0.916	0.958	0.893	0.885	0.888	0.892	0.834	0.853	0.917
	Llama-3.1	0.959	0.852	0.844	0.851	0.757	0.809	0.801	0.902	0.956	0.860	0.852	0.852	0.856	0.809	0.822	0.910
	GPT-4o-mini	0.942	0.888	0.882	0.885	0.780	0.826	0.827	0.913	0.945	0.854	0.862	0.890	0.888	0.837	0.831	0.919
	Average	0.954	0.881	0.874	0.878	0.832	0.822	0.814	0.911	0.953	0.867	0.865	0.872	0.870	0.820	0.826	0.914
Humanities	GPT-3.5	0.903	0.854	0.856	0.831	0.863	0.816	0.780	0.881	0.898	0.827	0.824	0.834	0.826	0.787	0.768	0.873
	Mixtral	0.954	0.856	0.854	0.851	0.850	0.817	0.805	0.876	0.952	0.831	0.828	0.838	0.832	0.803	0.791	0.870
	Moonshot	0.936	0.851	0.840	0.843	0.844	0.779	0.807	0.862	0.945	0.825	0.829	0.841	0.823	0.789	0.814	0.872
	Llama-3.1	0.928	0.809	0.800	0.806	0.810	0.771	0.771	0.851	0.922	0.701	0.787	0.795	0.789	0.739	0.768	0.856
	GPT-4o-mini	0.903	0.843	0.836	0.848	0.838	0.799	0.771	0.868	0.905	0.830	0.747	0.830	0.806	0.806	0.730	0.868
	Average	0.925	0.843	0.837	0.836	0.841	0.796	0.787	0.867	0.924	0.803	0.803	0.828	0.815	0.785	0.774	0.868
AIGTBench	GPT-3.5	0.964	0.875	0.870	0.876	0.863	0.840	0.804	0.894	0.959	0.873	0.847	0.863	0.851	0.833	0.814	0.892
	GPT-4o-mini	0.956	0.823	0.833	0.866	0.800	0.855	0.820	0.896	0.954	0.798	0.835	0.834	0.802	0.815	0.803	0.889
	Llama-1	0.913	0.892	0.867	0.897	0.896	0.851	0.878	0.915	0.916	0.881	0.870	0.873	0.873	0.845	0.794	0.910
	Llama-2	0.918	0.856	0.817	0.855	0.837	0.795	0.775	0.892	0.919	0.844	0.840	0.856	0.844	0.794	0.811	0.893
	Llama-3.1	0.915	0.864	0.842	0.856	0.837	0.836	0.777	0.899	0.913	0.841	0.762	0.840	0.837	0.796	0.796	0.895
	Average	0.933	0.862	0.846	0.870	0.847	0.835	0.811	0.899	0.932	0.848	0.831	0.853	0.842	0.817	0.804	0.896
Overall Average		0.933	0.855	0.846	0.855	0.836	0.807	0.797	0.886	0.930	0.835	0.829	0.842	0.826	0.800	0.792	0.885

memory is analyzed separately in Table 3. As the primary evaluation metric, we use macro-F1. To provide a more detailed analysis of model behavior, we report not only the overall macro-F1 over all seen classes, but also separate scores on old classes and new classes at each incremental stage. This multi-dimensional evaluation enables us to quantify how different methods balance two competing objectives: mitigating catastrophic forgetting on old classes and adapting to newly introduced ones. Unless otherwise specified, we report the mean performance over three random seeds.

5 Experiments

5.1 Comparison with Baselines

We comprehensively compare *RidgeFT* with a wide range of baselines under 3 evaluation protocols. Baselines marked with *, such as LwF*, iCaRL*, BiC*, and EASE*, denote methods equipped with a replay mechanism that stores historical samples.

As shown in Figures A1 and 3, *RidgeFT* consistently outperforms all baselines across the P3 and P4 incremental stages in both domains. In the academic setting, encompassing the STEM, Humanities, and Social Science datasets, under P3 (sequentially adding Moonshot, Llama-3.1, and GPT-4o-mini), *RidgeFT* maintains strong old-class F1 (0.913, 0.870, 0.861) and new-class F1 (0.814, 0.729, 0.799) across the three stages. Under P4, it achieves full-F1 scores of 0.862 and 0.838, sig-

nificantly outpacing the strongest baseline, which drops from 0.797 to 0.730. This highlights that replay-based baselines still struggle to balance old and new classes, while replay-free baselines such as PASS and SimpleCIL are even more limited when generator classes arrive sequentially. This robust advantage extends to the social media setting, where *RidgeFT* sustains high full-F1 scores across the P3 stages (0.945, 0.868, 0.855) and P4 stages (0.878, 0.867). Notably, at the final social media P4 stage, *RidgeFT*’s new-class F1 reaches 0.899, substantially beating the strongest baseline (0.735). These results confirm that *RidgeFT* remains highly stable when absorbing successive generator arrivals without compromising past knowledge.

This advantage is further confirmed under the P5 protocol (refer to Table 1). Averaged over the two backbones, *RidgeFT* achieves 0.886 full-F1, 0.902 old-class F1, and 0.804 new-class F1. Compared with the strongest full-F1 baseline, it improves full-F1 by 0.037. Compared with the best baseline on new-class F1, it improves new-class F1 by 0.107, while also outperforming the best old-class baseline on old-class F1. This clearly reveals a core limitation of baselines: they are better at preserving the decision boundaries of previously learned classes, yet remain less effective at efficiently absorbing information from new classes. In contrast, *RidgeFT* not only performs better in retaining old classes, but also delivers a substantial improvement on the

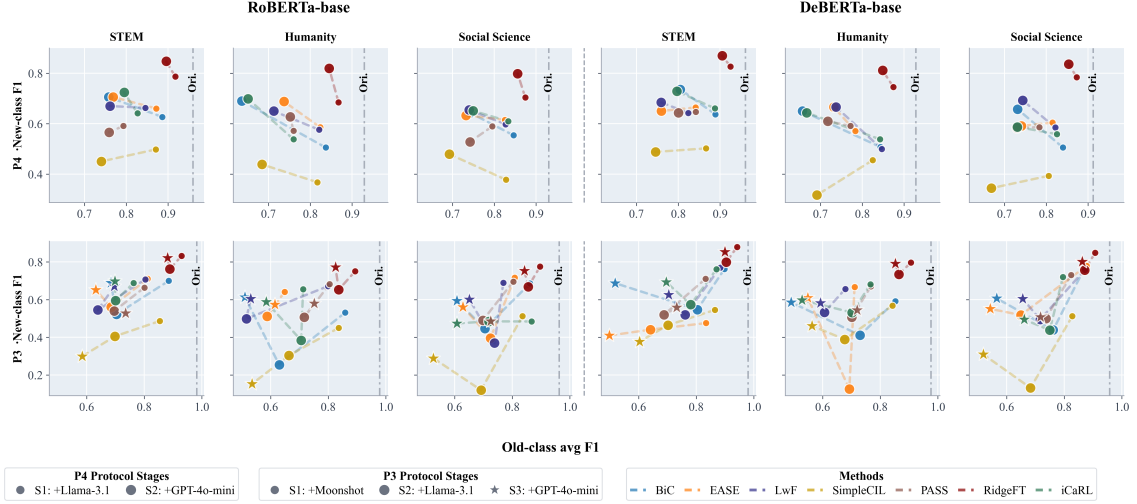


Figure 3: Experiments on academic topics. P3 starts with 3 initial classes and sequentially adds Moonshot, Llama-3.1, and GPT-4o-mini; P4 starts with 4 initial classes and sequentially adds Llama-3.1 and GPT-4o-mini.

more critical task of recognizing new classes, surpassing the strongest baseline by 0.107. Complete per-domain and per-backbone new- and old-class F1 are reported in Tables A5 and A6.

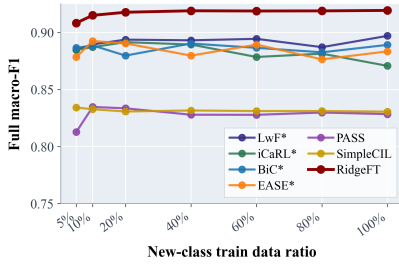


Figure 4: Full-F1 under varying target-class data proportions.

Finally, using GPT-4o-mini as the target generator (STEM / DeBERTa-base / P5), we analyze the impact of target-class data proportions. As shown in Figure 4, when reducing target-class data from 100% to 5%, *RidgeFT*’s full-F1 remains highly stable (ranging from 0.908 to 0.919). Conversely, the strongest replay baseline (LwF*) averages only 0.893, and the replay-free SimpleCIL drops to 0.831. This confirms that *RidgeFT* does not rely on abundant new-class data, demonstrating superior data efficiency and robustness even under severe low-resource incremental conditions.

To further assess scalability and generalization, we evaluate *RidgeFT* on a chronological cross-domain stream and on class sets of up to 18 source classes (Appendix D), and on the independently constructed RAID benchmark (Dugan et al., 2024), under DIPPER paraphrasing attacks, and through a

misattribution analysis (Appendix E). *RidgeFT* degrades gradually rather than abruptly as the stream and class set grow, and its advantage over baselines holds on the independent benchmark.

5.2 Ablation Studies

Table 2: Component ablation of *RidgeFT* under different new-class data ratios. W denotes whitening, RFL denotes random feature lift, and CBR denotes class-balanced ridge. “Full” denotes full macro-F1 over all seen classes, while “New” denotes new-class F1.

Ratio	Only W		Only RFL		Only CBR		<i>RidgeFT</i>	
	Full	New	Full	New	Full	New	Full	New
20%	0.881	0.741	0.887	0.763	0.912	0.865	0.918	0.878
40%	0.904	0.830	0.908	0.839	0.912	0.865	0.919	0.880
60%	0.909	0.851	0.913	0.861	0.913	0.865	0.919	0.881
80%	0.912	0.861	0.916	0.872	0.912	0.865	0.919	0.880
100%	0.910	0.858	0.917	0.878	0.912	0.863	0.919	0.881

Component-wise Ablation. We ablate the 3 core components of *RidgeFT* using GPT-4o-mini as the target generator (STEM / DeBERTa-base / P5). As shown in Table 2, each component contributes distinctly, and they are highly complementary. When Whitening is used alone, the model can stabilize encoder features to some extent, but its adaptability remains limited under low-resource conditions. For example, with only 20% of the new-class data, the new-class F1 reaches only 0.741. In contrast, Random Feature Lift consistently outperforms Whitening across all data ratios, indicating that random nonlinear features enhance the separability of different generators. Class-Balanced Ridge (CBR) is the strongest individual component. Even with

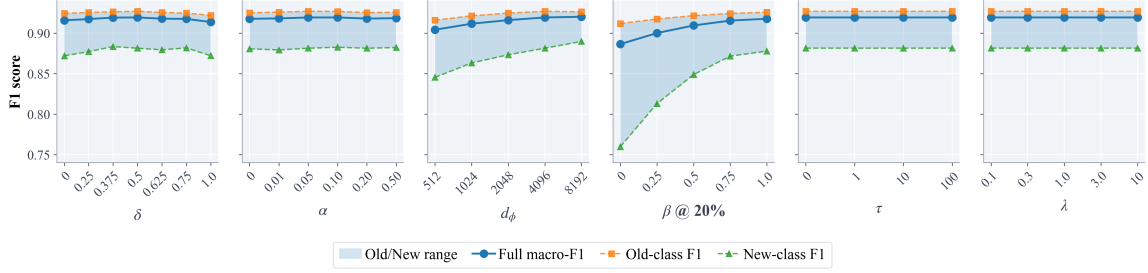


Figure 5: Parameter sensitivity of *RidgeFT*. We vary one hyperparameter at a time while keeping the others fixed, including the covariance calibration exponent δ , trace shrinkage coefficient α , random feature dimension d_ϕ , class-reweighting strength β under the 20% setting, smoothing constant τ , and ridge regularization coefficient λ .

only 20% of the data, it achieves a new-class F1 of 0.865, suggesting that it plays a central role in mitigating the bias between old and new classes. Notably, both CBR and the full *RidgeFT* exhibit stability across data ratios ranging from 20% to 100%. According to Equation (6), when $\beta = 1$ and $\tau = 0$, the class weight satisfies $\omega_c \propto 1/N_c$, which exactly offsets the linear growth of class statistics (A_c, q_c) with respect to the sample size N_c . Consequently, the ridge update effectively operates on class-averaged statistics, rendering the model largely insensitive to the sampling ratio once stable estimation is reached. Finally, combining all three components yields the best overall performance. Their complementarity is particularly evident in the 20% low-resource setting, where the full *RidgeFT* boosts the new-class F1 from 0.865 (CBR alone) to 0.878. Additional frozen-representation analysis is provided in Appendix C.

Parameter Sensitivity Ablation. Figure 5 presents the hyperparameter sensitivity of *RidgeFT*. Overall, it exhibits strong robustness to most hyperparameters. For example, varying the smoothing constant τ and the ridge regularization coefficient λ causes almost no performance fluctuation, with full-F1 remaining consistently around 0.919. From the perspective of feature stabilization, a moderate amount of trace shrinkage, with $\alpha \in [0.05, 0.10]$, is already sufficient to ensure reliable covariance estimation. The covariance calibration exponent δ performs best in the range $[0.375, 0.625]$, with full-F1 reaching 0.919 at $\delta = 0.5$. However, $\delta = 1.0$, which corresponds to inverse-covariance scaling stronger than standard whitening, instead suppresses useful discriminative information and leads to a performance drop. In terms of performance gains, the random feature dimension d_ϕ and the class reweighting strength β play the most important roles. As d_ϕ increases from 512 to 8192, the higher-dimensional

feature space substantially improves the separability of new generators, steadily raising full-F1 and new-class F1 to 0.920 and 0.890, respectively. In addition, under the low-resource setting with 20% data, strong reweighting with $\beta = 1.0$ boosts new-class F1 by nearly 12 percentage points compared with the unweighted case $\beta = 0$, increasing it from 0.760 to 0.878. This highlights its decisive role in alleviating the particularly severe imbalance between old and new classes.

Table 3: Sufficient-statistic compression ablation (default setting: $d_\phi = 4096$, 6 classes). Per-class schemes retain update flexibility, while merged schemes minimize storage by assuming future updates only add new classes, not new samples to old ones.

Storage scheme	Stored A statistics	Storage	Compression	full-F1
fp64 baseline	$6 \times A_c$ in fp64	782.35 MiB	$1.00\times$	0.919
fp32 per-class	$6 \times A_c$ in fp32	398.35 MiB	$1.96\times$	0.919
bf16 per-class	$6 \times A_c$ in bf16	206.35 MiB	$3.79\times$	0.920
merged fp32	$1 \times \bar{A}$ in fp32	78.35 MiB	$9.99\times$	0.919
merged bf16	$1 \times \bar{A}$ in bf16	46.35 MiB	$16.9\times$	0.920

Sufficient-Statistic Compression Ablation. Although *RidgeFT* is exemplar-free, it is not memory-free. Its closed-form update still requires storing the statistics needed for ridge regression: the calibration parameters, the fixed projection matrix R , and the class-wise sufficient statistics A_c, q_c , and N_c . The high-dimensional $A_c \in \mathbb{R}^{d_\phi \times d_\phi}$ dominates this storage cost. Under the default setting ($d_\phi = 4096$, 6 classes), the fp64 implementation requires 782.35 MiB in total, with the six class-specific A_c matrices alone accounting for 768 MiB.

To reduce this bottleneck, we examine compression along two orthogonal dimensions. The first is numerical precision (fp32 or bf16), which preserves the full class-wise statistic table. This retains the flexibility to reweight classes post-training or incorporate new samples into previously seen classes. The second is class aggregation, which stores only

the weighted merged matrix $\bar{A} = \sum_c \omega_c A_c$. This significantly reduces the memory footprint but folds the current class weights into the matrix, sacrificing the ability to update old classes individually. Importantly, this limitation is fully compatible with our P3/P4/P5 protocols, where incremental stages only introduce new generator classes.

As shown in Table 3, *RidgeFT* is highly robust to low-precision storage. Reducing per-class A_c to fp32 or bf16 substantially cuts storage while keeping the full-F1 consistently at or above 0.919. Class aggregation yields even larger gains by removing redundant class-specific matrices. Among all variants, merged bf16 provides the optimal trade-off: it requires only 46.35 MiB (a $16.9\times$ compression ratio) while achieving a full-F1 of 0.920. Overall, *RidgeFT* should be viewed as replay-free rather than storage-free. Merged bf16 is the most compact choice for our new-class-only protocols, whereas per-class bf16 is preferable if future updates may add samples to old classes.

6 Conclusion

This paper addresses the challenge of lifelong MGT attribution: continuously learning new generators without suffering catastrophic forgetting or relying on historical text replay. To overcome the dilemma between representation drift (from fine-tuning) and memory overhead (from exemplars), we propose *RidgeFT*. By freezing a task-tuned encoder and formulating incremental adaptation as closed-form ridge regression, our replay-free framework balances old and new classes without encoder parameter updates. Extensive experiments confirm that *RidgeFT* achieves a superior trade-off against baselines, demonstrating that combining stable representations with analytic updates offers a scalable path for future MGT attribution systems.

Limitations

Although *RidgeFT* achieves strong performance in MGT attribution, it still has several limitations.

- **Storage Requirements and Compression:** *RidgeFT* is replay-free but not storage-free: its closed-form updates still store the calibration parameters, the fixed projection matrix, and the class-wise sufficient statistics. Our compression study cuts this cost substantially, and further compressing these statistics is left to future work.

- **Long-term Deployment and Representation Stability:** Because *RidgeFT* keeps the encoder frozen during the incremental stage, it cannot absorb an unlimited number of generators without eventual degradation: as future generators diverge from the initial ones, the frozen representation may lack the cues that later updates need. Periodic encoder refresh between long stretches of low-cost analytic updates is therefore a promising direction (see Appendix D).
- **Adversarial Robustness:** Because *RidgeFT* attributes text in a frozen representation space, paraphrasing attacks that rewrite surface style can degrade its attribution accuracy (see Appendix E). Improving the robustness of the base-stage encoder before freezing remains an important direction for future work.

Ethical Considerations

This work focuses on lifelong machine-generated text attribution under benchmark settings. It does not involve private user data, human subjects, or new generative or attack capabilities. The main deployment risk is misattribution, which may lead to incorrect judgments about text sources. We therefore recommend using MGT attribution models as auxiliary evidence rather than the sole basis for high-stakes decisions, while also considering domain shift, transparency, and human oversight.

Acknowledgments

We sincerely thank the anonymous reviewers, the Area Chairs (AC), the Senior Area Chairs (SAC), and the Program Chairs for their careful reading, constructive comments, and thoughtful handling of our submission throughout the ACL Rolling Review and EMNLP 2026 process, which greatly helped us improve this paper. This work was supported in part by the National Key Research and Development Program of China (No. 2025YFB3110200), the Guangdong Basic and Applied Basic Research Foundation (No. 2026A1515030046), the State Key Laboratory of Internet Architecture, Tsinghua University (No. HLW2025ZD14), the Yangcheng Scholars Research Project (No. 2024312049), and the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No. 2023B1212010007).

References

- Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. 2022. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations (ICLR)*. OpenReview.net.
- Lucio La Cava, Dominik Macko, Róbert Móra, Ivan Srba, and Andrea Tagarelli. 2026. Authorship attribution in multilingual machine-generated texts. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 45136–45152. ACL.
- Lucio La Cava and Andrea Tagarelli. 2025. Openturingbench: An open-model-based benchmark and framework for machine-generated text detection and attribution. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26655–26671. ACL.
- Zhihui Chen, Kai He, Yucheng Huang, Yunxiao Zhu, and Mengling Feng. 2025. Divscore: Zero-shot detection of llm-generated text in specialized domains. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 19231–19253. ACL.
- Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12463–12492. ACL.
- Hongchao Fang, Yixin Liu, Jiangshu Du, Can Qin, Ran Xu, Feng Liu, Lichao Sun, Dongwon Lee, Lifu Huang, and Wenpeng Yin. 2025. Could AI trace and explain the origins of ai-generated images and text? *CoRR*, abs/2504.04279.
- Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.
- Wei Hao, Ran Li, Weiliang Zhao, Junfeng Yang, and Chengzhi Mao. 2025. Learning to rewrite: Generalized llm-generated text detection. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6421–6434. ACL.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations (ICLR)*. OpenReview.net.
- Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. MGTBench: Benchmarking machine-generated text detection. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2251–2265. ACM.
- Baixiang Huang, Canyu Chen, and Kai Shu. 2025. Authorship attribution in the era of llms: Problems, methodologies, and challenges. *ACM SIGKDD Explorations Newsletter*, 26(2):21–43.
- Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1416–1428.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Léo Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.
- Kaijie Jiao, Quan Wang, Licheng Zhang, Zikang Guo, and Zhendong Mao. 2025. M-rangedetector: Enhancing generalization in machine-generated text detection through multi-range attention masks. In *Findings of the Association for Computational Linguistics (ACL)*, pages 8971–8983. ACL.
- Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Tharindu Kumarage, Garima Agrawal, Paras Sheth, Raha Moraffah, Aman Chadha, Joshua Garland, and Huan Liu. 2024. A survey of ai-generated text forensic systems: Detection, attribution, and characterization. *CoRR*, abs/2403.01152.
- Olivier Ledoit and Michael Wolf. 2004. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411.
- Xiang Li, Zhiyi Yin, Hexiang Tan, Shaoling Jing, Du Su, Yi Cheng, Huawei Shen, and Fei Sun. 2025a. Prdetect: Perturbation-robust llm-generated text detection based on syntax tree. In *Findings of the Association for Computational Linguistics (NAACL)*, pages 8290–8301. ACL.
- Yuanfan Li, Zhaohan Zhang, Chengzhengxu Li, Chao Shen, and Xiaoming Liu. 2025b. Iron sharpens iron: Defending against attacks in machine-generated text detection with adversarial training. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3091–3113. ACL.
- Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis,

- Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.
- Yule Liu, Zhiyuan Zhong, Yifan Liao, Zhen Sun, Jingyi Zheng, Jiaheng Wei, Qingyuan Gong, Fenghua Tong, Yang Chen, Yang Zhang, and Xinlei He. 2025. On the generalization and adaptation ability of machine-generated text detectors in academic writing. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 5674–5685. ACM.
- Dominik Macko, Jakub Kopal, Róbert Móra, and Ivan Srba. 2025. Multisocial: Multilingual benchmark of machine-generated text detection of social-media texts. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 727–752. ACL.
- Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.
- Moonshot AI. 2026. Moonshot AI. <https://www.moonshot.ai/>. Accessed: 2026-05-15.
- Ayat A Najjar, Huthaifa I Ashqar, Omar Darwish, and Eman Hammad. 2025. Leveraging explainable ai for llm text attribution: Differentiating human-written and multiple llm-generated text. *Information*, 16(9):767.
- OpenAI. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.
- OpenClaw. 2026. Openclaw docs. <https://docs.openclaw.ai/>. Accessed: 2026-05-14.
- Adetoun A Oyelude. 2024. Artificial intelligence (ai) tools for academic research. *Library Hi Tech News*, 41(8):18–20.
- Andrea Pedrotti, Michele Papucci, Cristiano Ciacio, Alessio Miaschi, Giovanni Puccetti, Felice Dell’Orletta, and Andrea Esuli. 2025. Stress-testing machine generated text detection: Shifting language models writing style to fool detectors. In *Findings of the Association for Computational Linguistics (ACL)*, pages 3010–3031. ACL.
- Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.
- Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010.
- Shoumik Saha and Soheil Feizi. 2025. Almost ai, almost human: The challenge of detecting ai-polished writing. In *Findings of the Association for Computational Linguistics (ACL)*, pages 25414–25431. ACL.
- Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains. *Proces. del Leng. Natural*, 71:275–288.
- Zhixiong Su, Yichen Wang, Herun Wan, Zhaohan Zhang, and Minnan Luo. 2025. Haco-det: A study towards fine-grained machine-generated text detection under human-ai coauthoring. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 22015–22036. ACL.
- Zhen Sun, Zongmin Zhang, Xinyue Shen, Ziyi Zhang, Yule Liu, Michael Backes, Yang Zhang, and Xinlei He. 2025. Are we in the ai-generated text world already? quantifying and monitoring AIGT on social media. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 22975–23005. ACL.
- Llama Team. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.
- Eli Verwimp, Rahaf Aljundi, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L. Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H. Lampert, Martin Mundt, Razvan Pascanu, Adrian Popescu, Andreas S. Tolias, Joost van de Weijer, Bing Liu, Vincenzo Lomonaco, Tinne Tuytelaars, and Gido M. van de Ven. 2024. Continual learning: Applications and the road forward. *Trans. Mach. Learn. Res.*, 2024.
- Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia S. Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Comput. Linguistics*, 51(1):275–338.
- Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. 2019. Large scale incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–382.

Lu Yu, Bartłomiej Twardowski, Xialei Liu, Luis Heranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. 2020. Semantic drift compensation for class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6982–6991.

Xiao Yu, Yi Yu, Dongrui Liu, Kejiang Chen, Weiming Zhang, Nenghai Yu, and Jing Shao. 2025. Evobench: Towards real-world llm-generated text detection benchmarking for evolving large language models. In *Findings of the Association for Computational Linguistics (ACL)*, pages 14605–14620. ACL.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. 2025. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133(3):1012–1032.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. 2024. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23554–23564.

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. 2021. Prototype augmentation and self-supervision for incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5871–5880.

A Protocol and Metric Details

The attribution pool contains six classes, one Human source and five generators. Protocol P_k starts from k initial classes and introduces the remaining $6 - k$ generators one per incremental step, so P_5 adds a single held-out generator (rotated across generators in Table 1), while P_4 and P_3 add two and three generators in sequence. The academic arrival orders are given in Figure 3 and the social-media orders in Figure A1. At every step, predictions are made over all classes seen so far, and we report three quantities: full-F1, the macro-F1 over all seen classes; old-class F1, the macro-F1 over the initial k classes; and new-class F1, the F1 of the single generator introduced at the current step.

B Social Media Incremental Trajectories

Figure A1 reports the full P_3 and P_4 attribution trajectories on the social media domain, complementing the academic trajectories shown in Figure 3 of the main text.

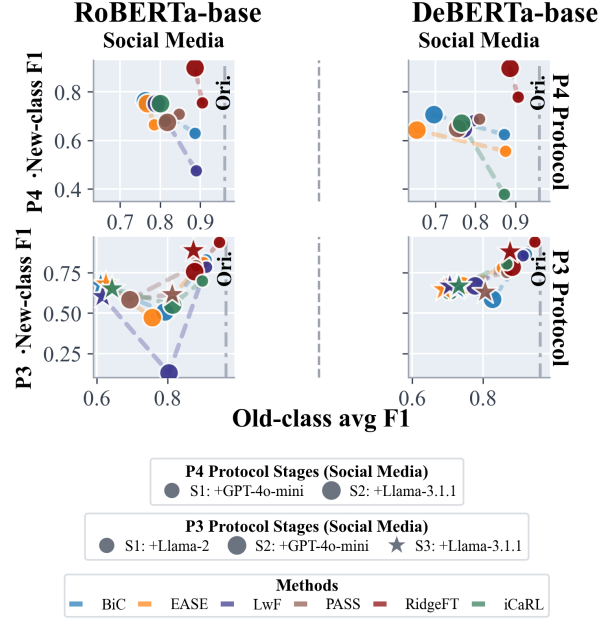


Figure A1: Lifelong MGT attribution experiments on the social media topic. P_3 starts with three initial classes and sequentially adds Llama-2, GPT-4o-mini, and Llama-3.1; P_4 starts with four initial classes and sequentially adds GPT-4o-mini and Llama-3.1.

C Sufficiency Analysis of Frozen Representations

Since *RidgeFT* freezes the task-tuned encoder during the incremental stage, a natural question arises: do frozen representations still preserve discriminative information for future generators? To answer this question, we compare three feature spaces derived from the same frozen encoder: the raw representation $h = f_\theta(x)$, the covariance-calibrated representation $\tilde{h} = P_\delta(h - \mu)$, and the random feature representation $z = \text{LN}(\text{ReLU}(R\tilde{h}))$. We diagnose these feature spaces using nearest-centroid probes and ridge probes, where none of the methods updates the encoder parameters.

Table A1: Sufficiency analysis of frozen representations. All methods use the same frozen encoder. Ridge on h already achieves a relatively strong new-class F1, suggesting that the frozen representation retains discriminative information for future generators. *RidgeFT* further improves performance, indicating that covariance calibration and random feature mapping can exploit this information more effectively.

Probe on frozen-encoder features	Full-F1	Old-F1	New-F1
Nearest-centroid on h	0.788	0.847	0.495
Nearest-centroid on z	0.854	0.879	0.730
Ridge on h	0.887	0.895	0.844
<i>RidgeFT</i> on z	0.919	0.927	0.881

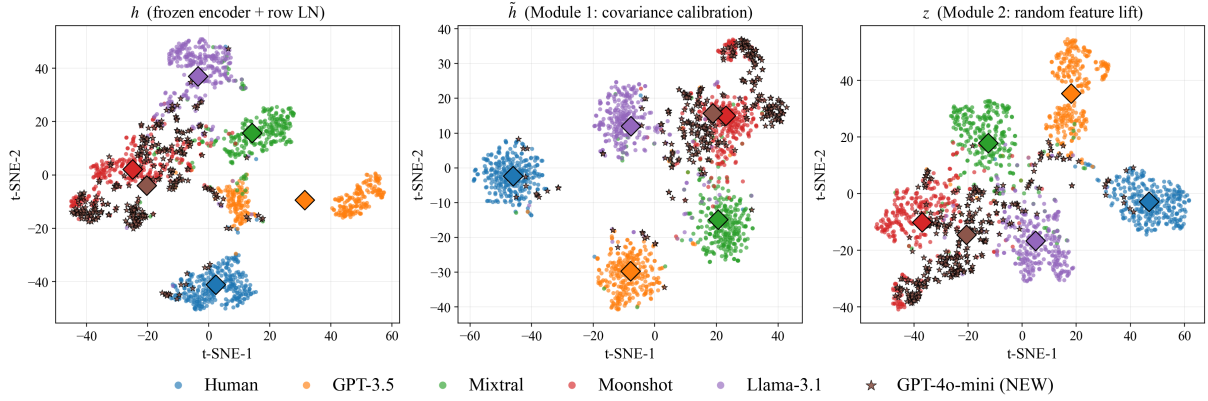


Figure A2: t-SNE visualization of different frozen feature spaces. We compare the raw frozen representation h , the calibrated representation $\tilde{h}$, and the random feature representation z . The figure shows that the newly introduced generator does not completely collapse into the old classes, and that the feature transformations in *RidgeFT* make the class structure more clearly separated (STEM / DeBERTa-base / P5 setting).

This analysis is conducted under the P5 protocol, where the model is first trained on five initial classes and GPT-4o-mini is introduced as the final new generator. As shown in Table A1, applying a ridge probe directly on the raw frozen representation h already yields a new-class F1 of 0.844, which suggests that the frozen encoder is not limited to the initial classes and still preserves useful discriminative cues for subsequent generators. Building on this, *RidgeFT* further improves the new-class F1 to 0.881, while also maintaining stable old-class F1. On the other hand, the nearest-centroid probe improves from 0.495 on h to 0.730 on z , indicating that the feature transformations in *RidgeFT* make the class structure more suitable for statistic-based analytic classification, consistent with the t-SNE visualization in Figure A2.

It should be noted that this analysis does not imply that a frozen encoder is always sufficient for arbitrary future generators. The random feature mapping can only enhance the usability of the information already present in the frozen representation, but it cannot recover generator-specific information that is entirely absent from the encoder. Therefore, *RidgeFT* is better suited to scenarios where the frozen representation still provides at least partial coverage of future generators. For generators that are extremely out of distribution, periodic adaptation of the encoder may still be necessary.

D Longer Streams and Larger Class Counts

Chronological cross-domain stream. To examine how *RidgeFT* behaves over a longer and tempo-

rally ordered stream, we merge MGT-Academic and AIGTBench into a single chronological benchmark. The stream has 8 classes (Human and 7 generators): every model shared by the two sources is merged into one class with exact within-class deduplication, and the classes are ordered by release date. We balance each class to 6,000 texts with an 80/10/10 train/validation/test split. The four cross-source classes break the spurious source-to-label correspondence, while the single-source classes keep their original domain composition, so the stream deliberately couples cross-generational with cross-domain shift. The encoder is fine-tuned only on the four earliest sources, Human, Llama-1 (Feb. 2023), GPT-3.5 Turbo (Mar. 2023), and Llama-2 (Jul. 2023), and is then frozen. Mixtral-8x7B (Dec. 2023), Moonshot-v1 (Feb. 2024), GPT-4o-mini (Jul. 2024), and Llama-3.1 (Jul. 2024) are added in chronological order, spanning about 17 months. Old-class F1 is the macro-F1 over the four initial classes, new-class F1 is the F1 of the latest arriving class, and full-F1 is the macro-F1 over all seen classes. As shown in Table A2, full-F1 decreases smoothly from 0.956 to 0.795 with DeBERTa-v3-base while the initial-class macro-F1 decreases more slowly, ending at 0.888, so most of the drop concentrates in the accumulated incremental classes. Replacing the backbone with DeBERTa-v3-large raises the final full-F1, old-class F1, and new-class F1 from 0.795/0.888/0.702 to 0.812/0.895/0.729, indicating that a stronger initial encoder improves the attainable performance for later generators while preserving the same gradual trend.

Table A2: Chronological cross-domain stream combining MGT-Academic and AIGTBench. The encoder is frozen after the four initial classes, and four generators are added in temporal order. Full/Old/New denote full macro-F1, initial-class macro-F1, and latest-class F1.

Incremental step	Seen	DeBERTa-v3-base			DeBERTa-v3-large		
		Full	Old	New	Full	Old	New
Initial	4	0.956	0.956	–	0.960	0.960	–
+ Mixtral	5	0.940	0.941	0.936	0.942	0.946	0.924
+ Moonshot	6	0.884	0.935	0.780	0.898	0.942	0.810
+ GPT-4o-mini	7	0.833	0.906	0.735	0.850	0.915	0.764
+ Llama-3.1	8	0.795	0.888	0.702	0.812	0.895	0.729

Scaling the number of classes. To probe larger class sets and the stability of the closed-form solver, we merge the source classes of RAID (Dugan et al., 2024) with those of our original benchmarks, remove overlapping models, and obtain 18 chronologically ordered source classes, with Human treated as one source class. The encoder is fine-tuned on the initial 6 classes and then frozen, after which 12 additional classes are added in sequence. We compare *RidgeFT* with SimpleCIL under the same frozen encoder, where SimpleCIL keeps a nearest-class-mean head and *RidgeFT* solves ridge regression on the calibrated and lifted representations, so the two already differ at the initial 6 classes. As reported in Table A3, *RidgeFT* decreases from 0.965 to 0.779 full-F1 as the class count grows from 6 to 18, whereas SimpleCIL drops from 0.943 to 0.290. All closed-form solves complete successfully at the reported checkpoints, so the growing class count does not by itself cause numerical failure of the ridge solver. Increasing the random-feature dimension d_ϕ from 4096 to 8192 yields only a small further gain at the largest scale (full-F1 from 0.779 to 0.789 at 18 classes), indicating that the default $d_\phi = 4096$ already provides sufficient capacity over this range. Random-feature capacity thus becomes more useful as classes accumulate, yet does not need to grow proportionally with the class count over this range.

Table A3: Scaling the number of source classes on RAID merged with our benchmarks. *RidgeFT* and SimpleCIL use the same frozen encoder. Full/New denote full macro-F1 and latest-class F1.

Classes	<i>RidgeFT</i>		SimpleCIL	
	Full	New	Full	New
6	0.965	–	0.943	–
10	0.913	0.902	0.585	0.299
14	0.840	0.804	0.372	0.154
18	0.779	0.742	0.290	0.144

E Independent Benchmark, Adversarial Robustness, and Error Analysis

Validation on an independent benchmark. To test *RidgeFT* on data built by a different group, we replicate the P5 protocol on RAID (Dugan et al., 2024) using the same DeBERTa-v3-base recipe and baseline implementations, with GPT-4 introduced as the new class at the final step. As shown in Table A4, *RidgeFT* attains the best Full, Old, and New F1 among all methods. Extending the setup to all 12 RAID source classes, with 6 initial and 6 incremental classes, *RidgeFT* reaches 0.857 full-F1 against 0.740 for the strongest baseline, iCaRL. The advantage of *RidgeFT* therefore transfers to an independently sourced benchmark and holds under a larger class set.

Table A4: Attribution performance on the independently constructed RAID benchmark under the P5 protocol, with GPT-4 as the new class. All methods use the same DeBERTa-v3-base backbone.

Method	Full-F1	Old-F1	New-F1
<i>RidgeFT</i> (Ours)	0.977	0.976	0.983
EASE	0.944	0.953	0.902
BiC	0.942	0.955	0.873
LwF	0.939	0.955	0.858
iCaRL	0.939	0.954	0.861
PASS	0.917	0.938	0.808
SimpleCIL	0.852	0.901	0.611

Adversarial robustness. We evaluate *RidgeFT* under a DIPPER paraphrasing attack (Krishna et al., 2023) in the STEM / DeBERTa-v3-base / P5 setting with GPT-4o-mini as the new class. Attribution accuracy on machine-generated texts drops from 0.896 on clean inputs to 0.264 after rewriting, and SimpleCIL on the same frozen features reaches 0.144, suggesting that paraphrasing substantially weakens the generator-specific cues exploited by both classifiers. Because *RidgeFT* leaves the incremental update untouched, robustness can be injected through the base stage: adversarially training the DeBERTa-v3-base encoder before freezing raises the attacked-text accuracy from 0.264 to 0.604 under the same protocol. For reproducibility, we attack with DIPPER (the dipper-paraphraser-xxl model) in bfloat16, paraphrasing each text sentence by sentence with the preceding output as context and sampling with top- $p = 0.75$ and lexical and order diversity both set to 60. Only the machine-generated test texts are rewritten (100 per generator, 500 in

total) and the Human texts are never attacked; the rewritten texts are re-encoded by the same frozen encoder, and on this machine-only subset we report accuracy, the fraction still attributed to the true generator, since per-class F1 is ill-defined once the Human class is absent. For the robust variant, we augment only the base-stage initial-class data with paraphrases (about a 12% augmentation ratio) and leave the incremental protocol unchanged, so no new-class paraphrase enters encoder training.

Misattribution patterns. Using the final P5 DeBERTa-v3-base predictions with GPT-4o-mini as the new class, we compute row-normalized confusion matrices for each domain (Figure A3). Human texts are rarely assigned to machine classes, with Human recall between 0.95 and 0.98 across domains. The remaining errors concentrate between generators of the same family or writing style: in the AIGTBench social-media domain, the largest confusions occur between GPT-3.5 and GPT-4o-mini (0.148 and 0.100 in the two directions) and between Llama-2 and Llama-3.1 (0.099 and 0.071). The machine-source errors of *RidgeFT* are thus structured around a few closely related generators rather than spread uniformly.

F Artifact Licenses, Intended Use, and Data Safety

We use two publicly released benchmark datasets, MGT-Academic (Liu et al., 2025) and AIGTBench (Sun et al., 2025), and do not collect any new user data. MGT-Academic is released under the MIT license, and AIGTBench is released under the Apache 2.0 license. Both datasets are released for research on machine-generated or AI-generated text detection/attribution, and our use is consistent with their intended research purposes. We cite the original datasets and papers, and we do not redistribute the raw datasets as part of this work.

For privacy and content safety, we use the processed benchmark versions released by the original authors. These datasets have already undergone preprocessing and moderation by their respective creators. Our experiments only use text content and class labels for aggregate model training and evaluation, and we do not analyze, expose, or redistribute any individual-level information.

G Software Implementations and Package Settings

All experiments were conducted in a Linux environment with kernel version 5.4.0-174-generic and Python 3.12.2. We used PyTorch 2.10.0+cu128 with CUDA 12.8 on an NVIDIA L20 GPU with driver version 550.54.15. Transformer encoders and tokenizers were implemented with HuggingFace Transformers 4.57.6, using DeBERTa-v3-base and RoBERTa-base as the main backbones. Encoder fine-tuning followed a unified configuration with a maximum sequence length of 256, batch size 16, 3 training epochs, AdamW optimization, weight decay, and gradient clipping.

Traditional machine learning classifiers and evaluation metrics were implemented with scikit-learn 1.7.2. Numerical computation relied on NumPy 2.0.1 and SciPy 1.15.2, while result aggregation and visualization used Pandas 2.2.3 and Matplotlib 3.9.1. For *RidgeFT*, we fixed the main hyperparameters across all primary experiments: calibration exponent $\delta = 0.5$, shrinkage coefficient $\alpha = 0.05$, random feature dimension $d_\phi = 4096$, ridge regularization coefficient $\lambda = 1.0$, and class-balanced ridge coefficient $\beta = 1.0$. Each experiment was run three times, and we report the average results.

H Computational Experiments

We report the computational cost of different update strategies on a single NVIDIA L20 GPU. Under the P5 protocol with DeBERTa-v3-base, *RidgeFT* updates the classifier by recomputing sufficient statistics and solving a closed-form ridge regression problem, which only takes a few seconds for the newly added generator. In contrast, the joint-from-scratch baseline retrains an 184M-parameter DeBERTa-v3-base model on all old and new training texts, involving about 94k samples, and takes approximately 50 minutes on a single GPU. The initial five-class encoder training is shared across methods and takes 1,738 seconds.

These results show that *RidgeFT* achieves much lower incremental update cost than full retraining, while avoiding access to old raw texts during the update stage. This efficiency advantage is central to our lifelong MGT attribution setting, where attribution systems must incorporate newly emerging generators without repeatedly retraining the entire model.

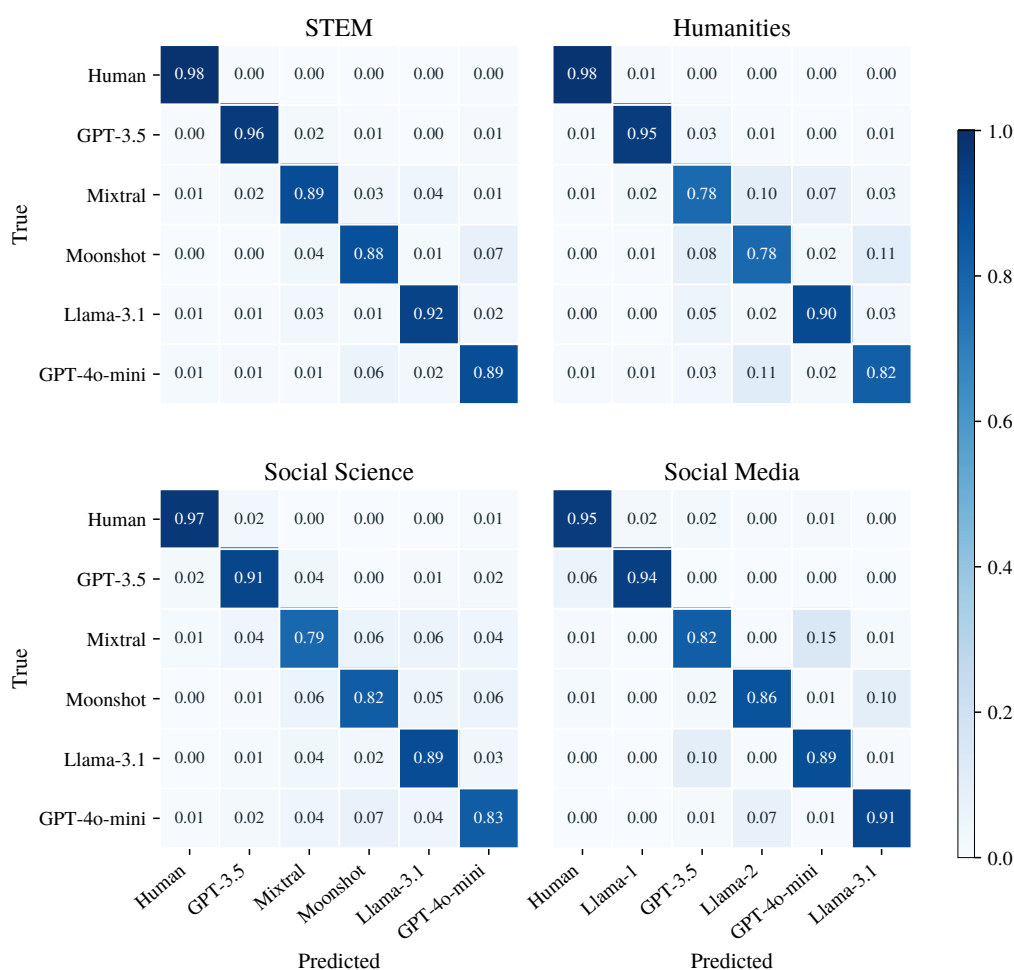


Figure A3: Row-normalized confusion matrices of the final P5 DeBERTa-v3-base *RidgeFT* model across the four evaluation domains, with GPT-4o-mini as the newly introduced class. Human texts are rarely misassigned to machine classes, and the residual errors concentrate between generators of the same family or writing style.

I Use of AI Assistants

We used AI assistants only for language polishing, grammar checking, and improving the readability of the manuscript. All research ideas, experimental designs, implementation decisions, experimental results, analyses, and final claims were developed, checked, and approved by the authors. The AI assistants were not used to independently generate, validate, or interpret experimental results.

Table A5: New-class F1 across Domains and Models (final continual step). * denotes methods using data replay. Darker color indicates better performance. Bold values are the highest among continual-learning methods of the RoBERTa and DeBERTa blocks in each row. The “Ori.” column is marked “—” because no new class has been introduced at S0.

Domain	New Model	RoBERTa-base								DeBERTa-base							
		Ori.	LwF*	iCaRL*	BiC*	EASE*	PASS	SimpleCIL	RidgeFT (Ours)	Ori.	LwF*	iCaRL*	BiC*	EASE*	PASS	SimpleCIL	RidgeFT (Ours)
Social Science	GPT-3.5	—	0.709	0.730	0.701	0.741	0.591	0.435	0.804	—	0.678	0.648	0.634	0.446	0.611	0.478	0.785
	Mixtral	—	0.667	0.684	0.677	0.665	0.621	0.356	0.760	—	0.632	0.652	0.630	0.420	0.592	0.399	0.750
	Moonshot	—	0.675	0.679	0.655	0.557	0.590	0.477	0.738	—	0.644	0.658	0.646	0.473	0.590	0.430	0.740
	Llama-3.1	—	0.592	0.591	0.572	0.618	0.560	0.332	0.716	—	0.457	0.555	0.486	0.395	0.541	0.347	0.742
	GPT-4o-mini	—	0.676	0.662	0.721	0.696	0.591	0.546	0.808	—	0.713	0.718	0.710	0.705	0.646	0.481	0.830
	Average	—	0.664	0.669	0.665	0.655	0.591	0.429	0.765	—	0.625	0.646	0.621	0.488	0.596	0.427	0.769
STEM	GPT-3.5	—	0.767	0.761	0.751	0.737	0.610	0.460	0.857	—	0.690	0.707	0.677	0.665	0.615	0.459	0.845
	Mixtral	—	0.724	0.728	0.693	0.719	0.637	0.424	0.804	—	0.722	0.708	0.686	0.711	0.614	0.476	0.825
	Moonshot	—	0.751	0.710	0.747	0.746	0.663	0.512	0.823	—	0.766	0.771	0.760	0.768	0.660	0.609	0.839
	Llama-3.1	—	0.666	0.582	0.629	0.589	0.612	0.432	0.787	—	0.677	0.669	0.625	0.673	0.616	0.519	0.821
	GPT-4o-mini	—	0.798	0.784	0.800	0.644	0.661	0.567	0.871	—	0.731	0.756	0.792	0.813	0.681	0.556	0.881
	Average	—	0.741	0.713	0.724	0.687	0.637	0.479	0.829	—	0.717	0.722	0.708	0.726	0.637	0.524	0.842
Humanities	GPT-3.5	—	0.806	0.805	0.794	0.815	0.684	0.496	0.867	—	0.713	0.756	0.741	0.738	0.652	0.525	0.858
	Mixtral	—	0.645	0.628	0.610	0.632	0.612	0.420	0.696	—	0.615	0.604	0.575	0.558	0.589	0.423	0.690
	Moonshot	—	0.661	0.665	0.635	0.636	0.589	0.542	0.709	—	0.644	0.643	0.634	0.630	0.596	0.557	0.723
	Llama-3.1	—	0.525	0.607	0.518	0.543	0.587	0.416	0.693	—	0.567	0.509	0.530	0.515	0.560	0.441	0.735
	GPT-4o-mini	—	0.733	0.741	0.752	0.743	0.652	0.480	0.827	—	0.686	0.632	0.693	0.682	0.667	0.364	0.823
	Average	—	0.674	0.689	0.662	0.674	0.625	0.471	0.758	—	0.645	0.629	0.635	0.625	0.612	0.462	0.766
AIGTBench	GPT-3.5	—	0.678	0.646	0.674	0.643	0.667	0.363	0.745	—	0.720	0.708	0.660	0.705	0.676	0.468	0.755
	GPT-4o-mini	—	0.701	0.558	0.678	0.580	0.732	0.507	0.789	—	0.682	0.696	0.598	0.685	0.690	0.494	0.775
	Llama-1	—	0.903	0.851	0.917	0.922	0.810	0.845	0.960	—	0.877	0.873	0.878	0.882	0.772	0.625	0.943
	Llama-2	—	0.751	0.708	0.748	0.724	0.650	0.409	0.867	—	0.730	0.769	0.746	0.763	0.638	0.574	0.862
	Llama-3.1	—	0.780	0.758	0.815	0.794	0.735	0.487	0.907	—	0.743	0.678	0.715	0.756	0.663	0.557	0.894
	Average	—	0.763	0.704	0.766	0.733	0.719	0.522	0.853	—	0.750	0.745	0.719	0.758	0.688	0.543	0.846
Overall Average		—	0.710	0.694	0.704	0.687	0.643	0.475	0.801	—	0.684	0.685	0.671	0.649	0.633	0.489	0.806

Table A6: Old-class F1 across domains and models (final incremental step, initial-class subset). * denotes methods using data replay. Darker color indicates better performance. Bold values indicate the best result among continual-learning methods within each RoBERTa-base and DeBERTa-base block. The “Ori.” column reports the S0 macro-F1 on the initial classes, which is identical to full-F1 because all classes are old at S0.

Domain	New Model	RoBERTa-base								DeBERTa-base							
		Ori.	LwF*	iCaRL*	BiC*	EASE*	PASS	SimpleCIL	RidgeFT (Ours)	Ori.	LwF*	iCaRL*	BiC*	EASE*	PASS	SimpleCIL	RidgeFT (Ours)
Social Science	GPT-3.5	0.906	0.858	0.868	0.861	0.864	0.808	0.836	0.877	0.891	0.847	0.845	0.835	0.814	0.814	0.822	0.864
	Mixtral	0.938	0.881	0.868	0.886	0.881	0.837	0.854	0.905	0.938	0.883	0.866	0.873	0.826	0.824	0.848	0.903
	Moonshot	0.931	0.881	0.875	0.877	0.868	0.818	0.855	0.893	0.922	0.873	0.863	0.864	0.850	0.822	0.847	0.886
	Llama-3.1	0.920	0.865	0.849	0.863	0.850	0.801	0.835	0.882	0.912	0.850	0.837	0.844	0.824	0.792	0.818	0.875
	GPT-4o-mini	0.905	0.860	0.840	0.862	0.832	0.804	0.839	0.879	0.898	0.859	0.844	0.846	0.854	0.825	0.822	0.874
	Average	0.920	0.869	0.860	0.870	0.859	0.814	0.844	0.887	0.912	0.862	0.851	0.852	0.834	0.815	0.831	0.881
STEM	GPT-3.5	0.946	0.905	0.903	0.909	0.892	0.841	0.874	0.925	0.942	0.878	0.879	0.891	0.883	0.847	0.869	0.919
	Mixtral	0.964	0.920	0.914	0.914	0.904	0.869	0.886	0.933	0.966	0.911	0.911	0.912	0.898	0.853	0.891	0.938
	Moonshot	0.959	0.924	0.914	0.920	0.912	0.877	0.893	0.934	0.958	0.918	0.908	0.914	0.916	0.869	0.902	0.932
	Llama-3.1	0.959	0.889	0.896	0.895	0.790	0.848	0.875	0.925	0.956	0.896	0.889	0.897	0.893	0.847	0.883	0.928
	GPT-4o-mini	0.942	0.906	0.901	0.902	0.808	0.859	0.879	0.921	0.945	0.879	0.883	0.909	0.903	0.868	0.886	0.927
	Average	0.954	0.909	0.906	0.908	0.861	0.859	0.881	0.928	0.953	0.896	0.894	0.905	0.899	0.857	0.886	0.929
Humanities	GPT-3.5	0.903	0.864	0.866	0.838	0.872	0.842	0.837	0.884	0.898	0.849	0.838	0.853	0.844	0.814	0.816	0.876
	Mixtral	0.954	0.898	0.899	0.899	0.893	0.857	0.882	0.912	0.952	0.874	0.872	0.891	0.887	0.846	0.864	0.906
	Moonshot	0.936	0.889	0.874	0.885	0.885	0.817	0.860	0.893	0.945	0.861	0.866	0.882	0.862	0.828	0.866	0.902
	Llama-3.1	0.928	0.866	0.839	0.863	0.863	0.808	0.842	0.882	0.922	0.727	0.843	0.848	0.844	0.774	0.834	0.880
	GPT-4o-mini	0.903	0.865	0.855	0.867	0.857	0.828	0.829	0.876	0.905	0.859	0.769	0.857	0.831	0.833	0.804	0.877
	Average	0.925	0.876	0.867	0.870	0.874	0.831	0.850	0.889	0.924	0.834	0.838	0.866	0.853	0.819	0.837	0.888
AIGTBench	GPT-3.5	0.964	0.915	0.915	0.916	0.907	0.875	0.892	0.924	0.959	0.903	0.874	0.903	0.880	0.864	0.883	0.920
	GPT-4o-mini	0.956	0.847	0.888	0.903	0.844	0.879	0.882	0.918	0.954	0.821	0.862	0.881	0.825	0.840	0.864	0.912
	Llama-1	0.913	0.890	0.870	0.893	0.891	0.859	0.884	0.906	0.916	0.882	0.869	0.872	0.872	0.859	0.828	0.904
	Llama-2	0.918	0.877	0.838	0.876	0.860	0.824	0.848	0.897	0.919	0.867	0.854	0.878	0.861	0.825	0.859	0.899
	Llama-3.1	0.915	0.881	0.858	0.865	0.846	0.856	0.835	0.897	0.913	0.860	0.779	0.865	0.853	0.823	0.844	0.895
	Average	0.933	0.882	0.874	0.890	0.869	0.859	0.868	0.908	0.932	0.867	0.848	0.880	0.858	0.842	0.856	0.906
Overall Average		0.933	0.884	0.877	0.885	0.866	0.840	0.861	0.903	0.930	0.865	0.857	0.876	0.861	0.833	0.853	0.901