SemEval-2026-Task13

Giorgia Rosalia, Buccelli
s354885

Antonella, Coviello
s344607

Alexandra Elena, Holota
s338437

Marco, Scaglione
s336253

Simone, Scalora
s329444

Abstract

The growing use of large language models for code generation makes distinguishing machine-generated code from human-written code increasingly difficult, especially under distribution shifts in language, domain, and generator family. SemEval-2026 Task 13 targets this challenge through three subtasks: binary detection, multi-class authorship attribution, and hybrid/adversarial code detection. In this paper, we conduct a system-level empirical study across all subtasks, comparing frozen encoder representations, feature-based classifiers, fine-tuned transformer models, post-hoc calibration, and probability-level ensembling. Our results show a consistent generalization gap: strong in-domain validation scores substantially overestimate performance on shifted test conditions. In Subtask A, simpler frozen-encoder models generalize better than more complex ensembles. In Subtask B, severe class imbalance and generator shift cause recall collapse on minority LLM families, dominating Macro F1 degradation. In Subtask C, longer context windows improve robustness, while post-hoc calibration markedly improves confidence reliability and enables selective decision-making via confidence-based routing. Overall, we highlight distribution shift and uncertainty awareness as key obstacles for reliable machine-generated code detection and argue that robustness and calibration should be treated as first-class objectives alongside accuracy. The code is available at https://github.com/AlexandraElena-Holota/SemEval-2026-Task13.git

1 Introduction

The widespread adoption of large language models (LLMs) for code generation has made distinguishing human-written from machine-generated code increasingly challenging, especially when detection methods are applied outside their training conditions. Modern generators produce syntactically correct code with diverse stylistic patterns, which limits robustness under distribution shift. SemEval-2026 Task 13 directly addresses this issue by evaluating robustness under realistic generalization settings, including language, domain, and generator shift. The task is structured into three subtasks: binary machine-generated code detection (Subtask A), multi-class authorship attribution across LLM families (Subtask B), and hybrid or adversarial code detection (Subtask C). Together, these subtasks emphasize that strong in-domain performance alone is insufficient for reliable deployment. In this work, we adopt a system-level and diagnostic perspective on Task 13. Instead of proposing a single unified model, we investigate a range of modeling strategies across subtasks, including frozen encoders, feature-based classifiers, fine-tuned transformer models, post-hoc calibration, and ensembling. Our aim is to analyze how different design choices behave under distribution shift, identify recurring failure modes, and assess the role of predictive confidence. Across all three subtasks, our results show that in-domain validation performance consistently overestimates robustness under unseen conditions. Increased model complexity or ensembling does not uniformly improve generalization, and in some cases simpler models remain competitive under shift. For settings involving ambiguous or mixed authorship, probabilistic calibration emerges as a key component for obtaining reliable confidence estimates and enabling confidence-aware decision strategies.

1.1 Research Questions

We structure our study around the following research questions:

- **RQ1 (Robustness under shift):** How do different modeling strategies for binary machine-generated code detection generalize under language and domain shift? *(Subtask A)*

- **RQ2 (Multi-class attribution under imbalance):** How do class imbalance and generator shift affect multi-class code authorship attribution, and to what extent do calibration and ensembling mitigate these effects? *(Subtask B)*

- **RQ3 (Confidence and reliability):** How does probabilistic calibration influence confidence reliability and selective decision-making in hybrid and adversarial code detection? *(Subtask C)*

2 Background

2.1 Machine-Generated Code Detection and Authorship Attribution

Recent work has shown that detecting machine-generated code can be approached using both feature-driven and embedding-based methods. Feature-based approaches take advantage of stylistic and structural characteristics that have been demonstrated to vary between human-written and machine-generated code in practice, such as whitespace usage, indentation patterns, and identifier statistics. Instead, embedding-based methods capture more comprehensive syntactic and semantic regularities by using representations that are learned by pretrained code models. ccording to empirical data, both families of approaches have complementary strengths and weaknesses and can perform well in controlled, in-domain settings (Nirob et al., 2026).

Recent benchmarks have highlighted the challenge of fine-grained authorship attribution across multiple generator families, going beyond binary detection. Significant inter-class confusion can result from stylistic similarities between models and a lack of per-class data in these situations, especially for less represented classes. When it comes to multi-class code authorship attribution, these findings encourage the investigation of various modeling approaches and their combinations.

2.2 Generalization under Language, Domain, and Generator Shift

Robustness under distribution shift is a major problem in machine-generated code detection. When applied to unknown conditions, models trained on a small set of programming languages, domains, or generators may show noticeable performance degradation. Strong in-domain performance may not translate to new languages, application scenarios, or generator families, as CoDet-M4 clearly demonstrates, and hybrid or mixed-authorship settings exacerbate this disparity (Orel et al., 2025). These results imply that when test data differ in code style, structure, or provenance, evaluation on in-domain validation data may overestimate real-world robustness. Furthermore, improved robustness under distribution shift is not always guaranteed by increased model complexity or capacity, highlighting the significance of evaluation under realistic and varied generalization settings.

2.3 Calibration and Confidence-Aware Prediction

Predictive confidence becomes a crucial component of raw classification accuracy in tasks with ambiguous or mixed provenance. Previous research has demonstrated that even when accuracy is high, modern neural networks can show significant miscalibration, leading to overconfident predictions. Temperature scaling and other post-hoc calibration techniques are commonly used to match predicted probabilities with empirical correctness (Minderer et al., 2021).

More recent studies have emphasized the significance of calibration in selective and confidence-aware prediction. Selective recalibration techniques aim to increase reliability by allowing models to postpone or delay decisions on ambiguous examples and supporting implementation scenarios with human involvement or risk-sensitive considerations (Zollo et al., 2024). In the context of hybrid and adversarial code detection, these ideas promote evaluating not only classification performance but also probability reliability and confidence-based routing strategies.

3 System Overview

This section describes the methodological choices adopted to address SemEval-2026 Task 13 across the three subtasks. Each subtask targets a different label space (binary, multi-class family attribution, and hybrid/adversarial detection) and is evaluated primarily using Macro F1-score, which is the official target metric of the shared task. For Subtask C, we additionally report reliability-focused metrics (e.g., ECE and Brier score) and confidence-slice diagnostics, since probability quality is central to downstream use.

3.1 Common Experimental Pipeline

Across subtasks, we follow a consistent pipeline:

- **Data handling and splits.** We use the official train/validation splits provided by the task. For Subtask A (Binary Machine-Generated Code Detection), the full training set contains 500K examples (238K human-written and 262K machine-generated), with a validation set of 100K examples. We intentionally subsampled the training set (20K examples) to compare approaches under a constrained-data regime; Subtasks B and C use the full training data in the runs reported. For Subtask B (Multi-Class Authorship Detection), the training set consists of 500K code snippets, primarily human-written, together with samples generated by multiple LLMs (e.g., OpenAI, Meta-LLaMA, Qwen, DeepSeek, and others). The validation set contains 100K examples. For Subtask C (Hybrid Code Detection), the training split contains 900K examples, including human-written, machine-generated, hybrid, and adversarial code. Evaluation is performed on a held-out validation set of 200K examples. We report results for four experimental variants: CodeBERT (baseline), Label Smoothing, Balanced Sampling, and ModernBERT-1024.

- **Tokenization and caching.** For transformer-based models, we tokenize code with the corresponding pretrained tokenizer and enforce a fixed maximum sequence length (typically 512 tokens; 1024 tokens for the long-context variant in Subtask C). To support rapid iteration and reproducibility (especially in Subtasks B and C), we cache tokenized datasets and store model outputs (logits and probabilities) for validation/test when reusing them for calibration and ensembles.

- **Model selection by validation.** Model variants (e.g., preprocessing choices, context length, loss functions) are selected using in-domain validation performance, with Macro F1 as the primary metric.

- **Post-hoc calibration (when applicable).** For subtasks where confidence reliability is central (notably Subtask C, and also Subtask B for ensemble comparability), we fit calibration parameters on validation logits and apply them unchanged at test time.

- **Ensembling (when applicable).** When combining multiple models, we ensemble at the probability level using calibrated outputs to ensure comparability across models.

3.2 Subtask A: Binary Machine-Generated Code Detection

For Subtask A, we compare three complementary approaches that reflect different inductive biases: (i) frozen neural representations, (ii) manual feature engineering, and (iii) probability-level ensembling. All Subtask A models are trained on a 20K-sample subset of the training split to study performance under limited in-domain supervision.

Frozen encoder representations + logistic regression. We use a pretrained transformer code encoder as a fixed feature extractor and train a lightweight linear classifier on top. Concretely, we encode each snippet up to 512 tokens and extract the final-layer `[CLS]` embedding as a fixed-length representation. The encoder is kept frozen (no gradient updates), and we train a logistic regression classifier with L2 regularization and inverse-frequency class weights. This setup isolates the contribution of pretrained representations while minimizing the risk of overfitting through fine-tuning.

Manual feature-based classifier. We extract language-agnostic, interpretable features designed to capture structural and stylistic cues that may differ between human and machine-generated code. The feature set includes length/structure statistics (e.g., lines, average line length), comment usage, indentation patterns, identifier statistics, syntactic marker counts, verbosity indicators (e.g., docstrings/type hints), and repetition measures. These features are used to train a shallow Random Forest classifier with balanced class weights.

Probability-level ensembles. We explore two ensemble families. First, we ensemble multiple frozen-encoder+logistic-regression models (e.g., UniXcoder, GraphCodeBERT, CodeBERT), combining predicted probabilities with fixed weights. Second, we build a hybrid ensemble combining the frozen-encoder model and the feature-based classifier by weighted averaging of probabilities and a fixed 0.5 decision threshold. Ensemble weights are chosen on the validation subset.

3.3 Subtask B: Multi-Class Authorship Detection

Subtask B requires predicting one of eleven authorship classes (Human + 10 LLM families) under severe class imbalance and an evaluation setup that includes both seen and unseen authors. Our approach is based on fine-tuning CodeBERT classifiers under different training strategies and input representations, followed by post-hoc calibration and probability-level ensembling.

Input representations. We consider both **raw code** and **preprocessed code**. In the raw setting, comments and formatting are preserved to retain potentially informative stylistic cues. In the preprocessed setting, comments/docstrings are removed and whitespace is normalized to emphasize core code content and reduce superficial artifacts.

Model variants. We train multiple CodeBERT-based classifiers that differ in (i) context length and (ii) loss function:

- **Best Strategy model:** 512-token inputs on raw code, optimized with class-weighted cross-entropy to mitigate imbalance.
- **Focal-loss model:** 512-token inputs on raw code, optimized with focal loss to emphasize hard examples; gradient clipping is used for stability.
- **M3 model:** 256-token inputs on preprocessed code, optimized with class-weighted cross-entropy; designed to provide complementary behavior by focusing on early snippet regions and reduced stylistic information.

Calibration and ensembling. To make probability outputs comparable across models and improve decision reliability, we calibrate logits on the validation split using temperature scaling with class-wise bias correction, then apply softmax to obtain calibrated probabilities. Ensembles are formed by averaging calibrated probabilities; for the three-model ensemble, weights are selected via grid search on validation Macro F1 and then fixed for test-time inference.

3.4 Subtask C: Hybrid Code Detection

Subtask C extends detection beyond a binary setting by introducing *Hybrid* and *Adversarial* classes, making uncertainty estimation and confidence reliability important for downstream use. Our experiments in this subtask therefore emphasize both predictive performance (Macro F1) and probabilistic diagnostics (e.g., ECE and Brier score).

Model variants and context length. We evaluate four runs: a **CodeBERT baseline** fine-tuned with standard cross-entropy, two training modifications (**label smoothing** and **balanced sampling**), and a long-context encoder variant (**ModernBERT-1024**) to reduce information loss from truncation on long examples. All runs share the same evaluation protocol, enabling controlled comparisons of training and architecture choices.

Calibration. We apply post-hoc temperature scaling with an additive class-wise bias vector:

$$\tilde{p}(y = c \mid x) = \mathrm{softmax}\Big(\frac{z(x)}{T} + b\Big)_c,$$

where calibration parameters $(T, \mathbf{b})$ are learned on validation logits by minimizing negative log-likelihood and then applied unchanged to test logits. This calibration step is used to improve reliability metrics (e.g., ECE, Brier) and to support confidence-based analyses. In our experiments, we learned a temperature $T \approx 1.39$ and bias vector $\mathbf{b} \approx [+0.56, -0.34, -0.10, +0.20]$, which specifically corrected the model's tendency to under-predict Human code ($+0.56$) and over-predict Machine-generated code (-0.34).

Confidence-based diagnostics and routing. To study where errors concentrate, we analyze performance on confidence slices defined by the model's maximum predicted probability (e.g., hard vs. easy quartiles, computed using uncalibrated logits to isolate intrinsic model uncertainty). Calibrated confidence is additionally used to evaluate selective decision-making: high-confidence predictions can be accepted automatically while low-confidence cases can be deferred for human review, which is particularly relevant for hybrid and adversarial settings.

4 Experimental Results

This section reports the empirical results for each subtask, following the research questions defined in Section 1.1. We primarily report Macro F1, as required by the SemEval-2026 Task 13 leaderboard, and we highlight generalization gaps between in-domain validation and the provided test sample whenever available.

Model	Val Macro F1	Test Macro F1
Frozen Encoder + LR	0.9307	0.4641
Feature-Based RF	0.9405	0.3886
Ensemble	0.9497	0.4523

Table 1: Macro F1-score on the validation set and on the provided test set sample for Subtask A. All models are trained on 20K in-domain samples. Despite strong validation performance, all approaches exhibit a substantial drop on the test sample under language and domain shift.

Model	Validation Macro F1	Test Sample Macro F1
Best Strategy (single)	0.5802	0.3340
Focal Model (single)	0.5626	0.3321
M3 Model (single)	0.3363	0.2235
Best Strategy + M3	0.5343	0.3459
Best + Focal + M3	0.5859	0.3400

Table 2: Macro F1-score for Subtask B on the validation set and on the provided test sample. While calibrated ensembles slightly improve in-domain performance, all configurations suffer a large performance drop under generator and domain shift.

4.1 Subtask A Results

For the three methods discussed in Section 3.2 (frozen encoder + logistic regression, manual feature-based classifier, and a probability-level ensemble), Table 1 summarizes Macro F1 on the validation set and on the supplied test set sample. Validation performance is consistently high (Macro F1 ≥ 0.93), with the hybrid ensemble obtaining the best validation score (0.9497), despite the fact that all models are trained on only $20,000$ training examples.

All approaches, however, exhibit a sharp decline in test performance, suggesting limited robustness under language and domain shift. The frozen encoder model (0.4641) outperforms the ensemble (0.4523) and the feature-based model (0.3886) in terms of test Macro F1. In-domain validation may significantly overestimate robustness under the given test conditions, as indicated by the wide validation–test gap.

Beyond Macro F1, precision/recall analysis reveals a consistent test-time bias: recall for the machine-generated class remains high (above 0.88 across approaches), while recall for the human-written class collapses (down to 0.23 for the feature-based model). As a result, many human-written snippets are misclassified as machine-generated, which disproportionately harms Macro F1. Overall, these results indicate that models trained in-domain may over-rely on non-robust cues that do not transfer to unseen languages or domains. Notably, the strongest test performance is obtained by the simplest approach (frozen encoder + linear classifier), suggesting that added complexity and ensembling do not necessarily improve generalization in this setting.

4.2 Subtask B Results

Table 2 reports Macro F1 for the main single-model configurations and ensembles on the validation set and on the provided test sample. On validation data, the best single model (Best Strategy) achieves Macro F1 of 0.5802, while the focal-loss variant is slightly lower (0.5626). The M3 model, trained with shorter context (256 tokens) and preprocessed inputs, performs substantially worse in isolation (0.3363), but is included as a complementary component for ensembling.

Calibration and ensembling yield modest improvements on validation: the three-model ensemble (Best Strategy + Focal + M3) achieves the highest validation Macro F1 (0.5859), slightly improving over the best single model. On the test sample, however, all methods experience a large drop in Macro F1 (e.g., Best Strategy: 0.3340), consistent with the presence of unseen generators at test time. Among the ensemble variants, the two-model ensemble (Best Strategy + M3) attains the highest test-sample Macro F1 (0.3459), although differences across ensembles are small and may reflect variability in the provided test subset.

Error analysis based on confusion matrices indicates that the Macro F1 degradation is driven primarily by recall collapse on minority LLM

Run	Accuracy	Macro F1 (val)	Macro F1 (val, cal)	Hard F1	Hard F1 (cal)	Macro F1 (test)	Macro F1 (test, cal)	ECE (val, cal)
CodeBERT (Baseline)	0.8910	0.8351	0.8458	0.6325	0.6674	0.5424	0.5565	0.0036
Label smoothing	0.8955	0.8376	0.8457	0.6504	0.6707	0.5676	0.5756	0.0042
Balanced sampling	0.8733	0.8204	0.8447	0.5814	0.6394	0.5357	0.5809	0.0045
ModernBERT-1024	0.9200	0.8820	0.8821	0.7292	0.7297	0.6137	0.6143	0.0029

Table 3: Performance and calibration metrics on the held-out test set for Subtask C. Calibration parameters are fitted on validation logits and applied unchanged at test time. Increasing context length (ModernBERT-1024) yields the strongest gains in both accuracy and robustness, while post-hoc calibration substantially improves confidence reliability.

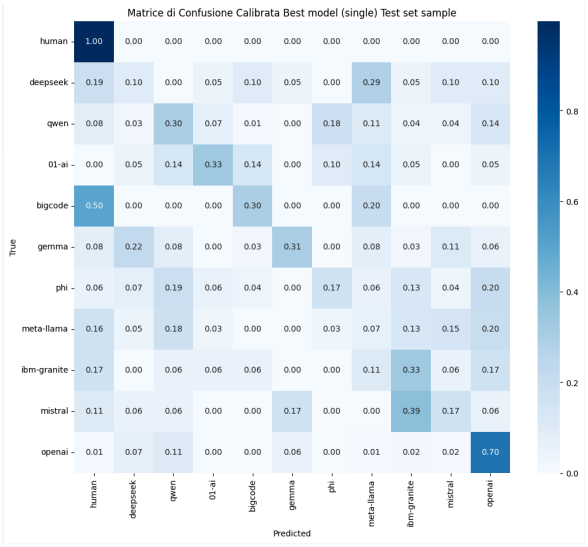

Figure 1: Confusion matrix for the Best Strategy single model on the test sample in Subtask B. Under generator and domain shift, recall collapses for several minority LLM families, while predictions are biased toward the Human class.

families (Figure 1). Under shift, many machine-generated samples are predicted as Human, reflecting a majority-class bias when stylistic cues differ from those seen during training. While some families (e.g., OpenAI and IBM-Granite) retain comparatively higher recall on the test sample, others such as Mistral, Qwen, and Meta-LLaMA exhibit severe recall degradation. These class-specific failures dominate Macro F1 despite comparatively high accuracy dominated by the Human class, illustrating the difficulty of fine-grained attribution under class imbalance and distribution shift.

4.3 Subtask C Results

For Subtask C, Table 3 summarizes performance and calibration metrics on the held-out test set for four experimental variants (CodeBERT baseline, label smoothing, balanced sampling, and ModernBERT-1024). The CodeBERT baseline achieves Accuracy of 0.8910 and Macro F1 of 0.8351. Label smoothing yields a small improvement in raw performance (Accuracy 0.8955; Macro F1 0.8376), while balanced sampling reduces raw Macro F1 (0.8204) and hard-slice performance (Hard F1 0.5814), indicating that naive rebalancing alone can introduce instability.

The strongest overall results are obtained by ModernBERT-1024, which achieves Accuracy of 0.9200 and Macro F1 of 0.8820, and also the best hard-slice performance (Hard F1 0.7292). This configuration resulted in a final ranking of 11th out of 32 teams on the official leaderboard, validating the effectiveness of increased context length for hybrid detection. This improvement is consistent with our tokenization analysis showing that increasing the truncation limit from 512 to 1024 tokens reduces the fraction of truncated examples from approximately 26% to approximately 6%, thereby retaining more information for long snippets. Notably, analysis of the training data shows that the 90th and 95th percentile token lengths are $p_{90} = 871$ and $p_{95} = 1169$ respectively, confirming that significant structural information lies beyond the standard 512-token window.

Calibration plays a central role in this subtask. Applying temperature scaling with an additive bias, fitted on validation logits and applied to test logits, improves calibration quality substantially: for the CodeBERT baseline, the calibrated ECE decreases to 0.0036 (Table 3), and reliability diagrams show that calibrated confidence tracks empirical accuracy closely (Figure 2).

Calibrated probabilities also support confidence-based analyses: performance on the hard slice improves after calibration (e.g., baseline Hard F1 increases from 0.6325 to 0.6674), and Macro F1 increases slightly when computed after calibration (baseline Macro F1 (cal) 0.8458).

Finally, selective decision curves (Figure 3) (Macro F1 vs. accepted fraction) support the use

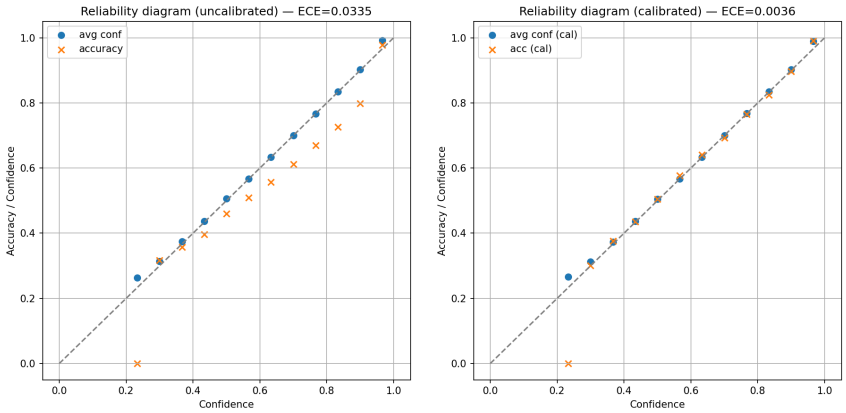

Figure 2: Reliability diagrams for Subtask C. (a) Uncalibrated CodeBERT baseline (ECE = 0.0335), showing systematic overconfidence. (b) After temperature scaling with additive bias (ECE = 0.0036), predicted confidence closely matches empirical accuracy.

of calibrated confidence for selective decision-making, with low-confidence cases being natural candidates for deferral to human review.

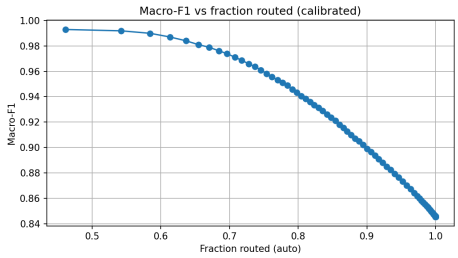

Figure 3: Macro-F1 as a function of the fraction of predictions automatically accepted using calibrated confidence. As coverage decreases, performance improves sharply, supporting confidence-based selective routing and human-in-the-loop decision strategies.

5 Conclusion

Three subtasks of SemEval-2026 Task 13 were examined at the system level in this paper: binary machine-generated code detection, multi-class authorship attribution, and hybrid/adversarial classification. Our findings demonstrate that robustness under realistic distribution shifts involving programming languages, domains, and code generators is consistently overestimated by strong in-domain validation performance across all settings.

The simplest frozen-encoder approach yields the best test results in Subtask A, where performance drops sharply under shift, suggesting that increased model complexity or ensembling does not always improve generalization. Recall collapse on minority LLM families and a bias toward the Human class are the main causes of Macro F1 degradation in Subtask B, underscoring the challenge of fine-grained attribution under class imbalance and generator shift. Longer context windows increase robustness in Subtask C, and post-hoc calibration significantly improves confidence reliability, allowing for efficient confidence-based diagnostics and selective decision-making.

Overall, these findings show that robustness to distribution shift and uncertainty awareness are the primary barriers to machine-generated code detection. Future work should prioritize calibration, address long-tail behavior, and enhance out-of-domain generalization.

References

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems*. ArXiv:2106.07998v2.

Syed Mehedi Hasan Nirob, Shamim Ehsan, Moqsadur Rahman, and Summit Haque. 2026. Whitespaces don't lie: Feature-driven and embedding-based approaches for detecting machine-generated code. *arXiv preprint arXiv:2601.19264*. Version 1.

Daniil Orel, Dilshod Azizov, and Preslav Nakov. 2025. Codet-m4: Detecting machine-generated code in multi-lingual, multi-generator and multi-domain settings. *arXiv preprint arXiv:2503.13733*. Version 2.

Thomas P. Zollo, Zhun Deng, Jake C. Snell, Toniann Pitassi, and Richard Zemel. 2024. Improving predictor reliability with selective recalibration. *Transactions on Machine Learning Research*. ArXiv:2410.05407v1.

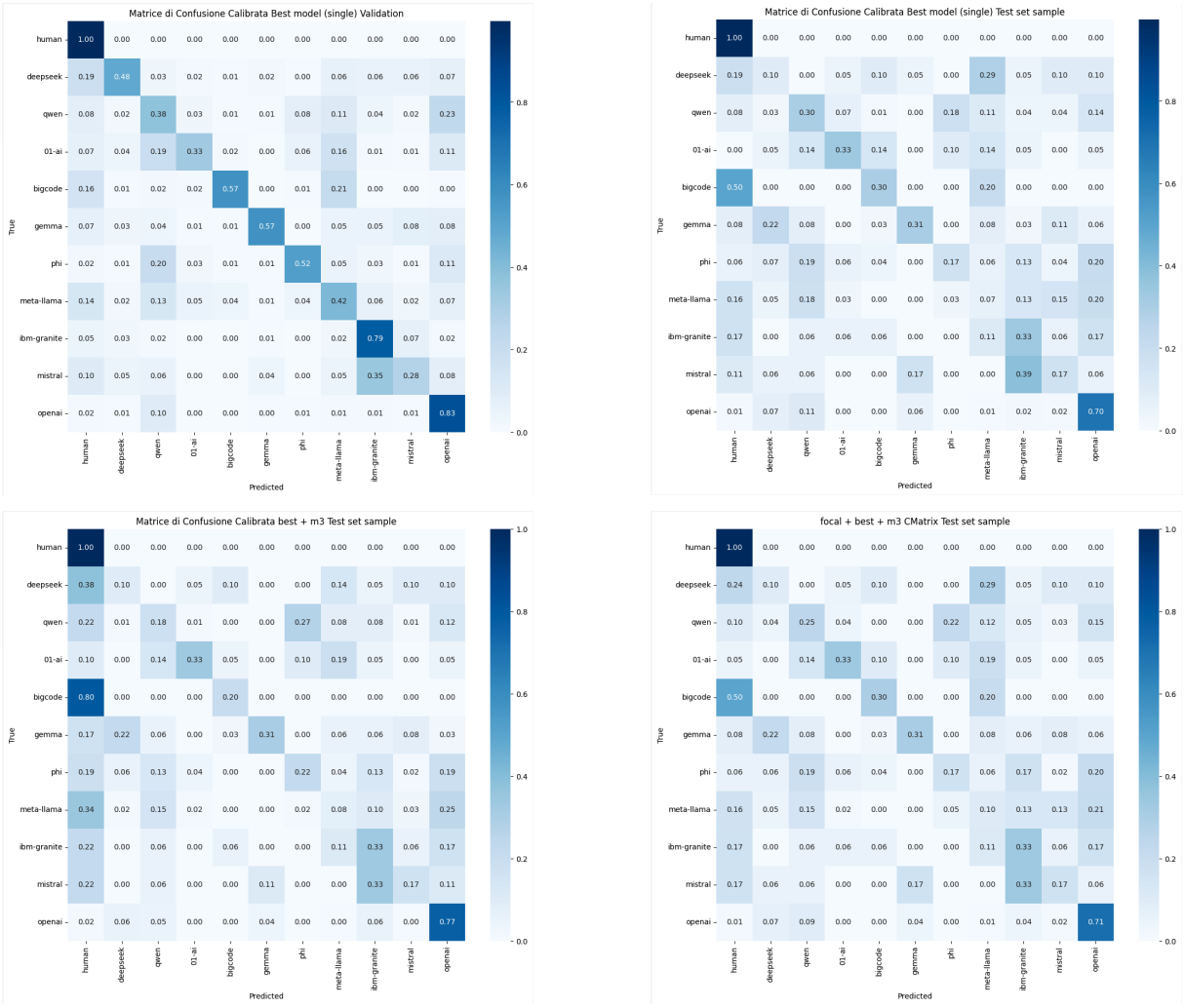

Figure 4: Confusion matrices for Subtask B under validation and test conditions. **Top:** Best Strategy single model on validation (left) and test sample (right), highlighting the strong shift-induced recall collapse for several minority LLM families. **Bottom:** Test-sample confusion matrices for the Best Strategy + M3 ensemble (left) and the Best Strategy + Focal + M3 ensemble (right). While ensembling yields modest improvements in Macro F1, the overall structure of test-time errors remains largely unchanged.

A Additional Error Analysis for Subtask B

This appendix provides additional confusion-matrix analyses for Subtask B (Multi-Class Authorship Detection) to complement the results discussed in the main paper. These diagnostics are intended to illustrate how error patterns change under generator and domain shift, and to contextualize the observed degradation in Macro F1-score.

Figure 4 summarizes confusion matrices for the main single-model and ensemble configurations under validation and test conditions. For the Best Strategy single model, validation results exhibit relatively structured misclassifications and moderate recall across several LLM families. In contrast, the corresponding test-sample confusion matrix shows a pronounced recall collapse for multiple minority generator families, with a large fraction of machine-generated samples being misclassified as Human.

The bottom row of Figure 4 reports confusion matrices for the two ensemble configurations evaluated on the test sample. While ensembling yields modest improvements in aggregate Macro F1, the qualitative structure of test-time errors remains largely unchanged. In particular, recall for several minority LLM families remains low, and predictions continue to be biased toward the Human class.

Overall, these confusion-matrix analyses confirm that the primary source of performance degradation in Subtask B is not suboptimal model combination, but rather the combined effect of class

imbalance and distribution shift across unseen generator families.

B Additional Calibration Diagnostics for Subtask C

This appendix provides additional diagnostic analyses for Subtask C that complement the results presented in the main paper. These figures offer deeper insight into class-specific calibration behavior, error structure, and confidence distributions, but are omitted from the main text for clarity.

B.1 Per-Class Reliability Diagrams

Figure 5 reports reliability diagrams computed separately for each class before calibration. While the global reliability diagram in the main paper summarizes overall behavior, these class-conditional plots reveal that miscalibration is not uniform across classes.

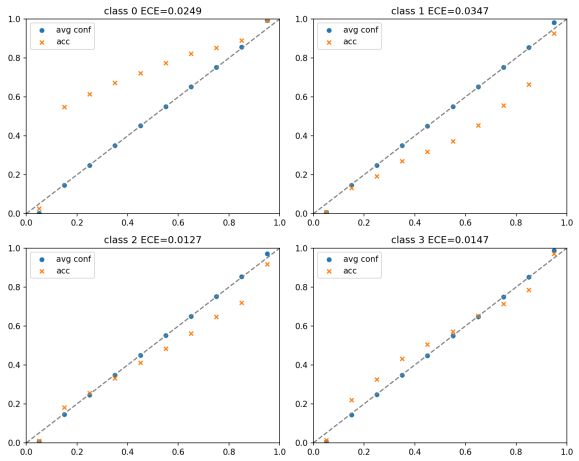

Figure 5: Per-class reliability diagrams before calibration, showing heterogeneous miscalibration patterns across classes.

B.2 Calibration Bias Vector

Figure 6 shows the additive bias vector learned during temperature-plus-bias calibration. The bias corrects systematic class-specific prediction tendencies, complementing global temperature scaling.

B.3 Confusion Matrices on the Validation Set

Figures 7 and 8 report confusion matrices for the CodeBERT baseline before and after calibration on the full validation set. These matrices illustrate how calibration affects confidence but does not substantially alter the argmax decision structure.

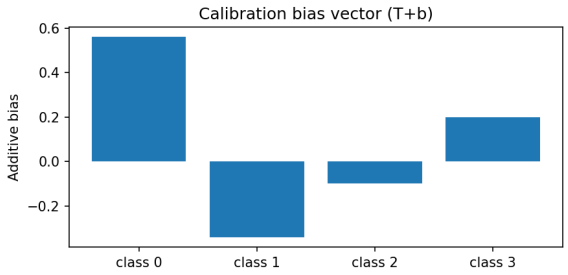

Figure 6: Additive class-wise bias vector learned during post-hoc calibration for Subtask C.

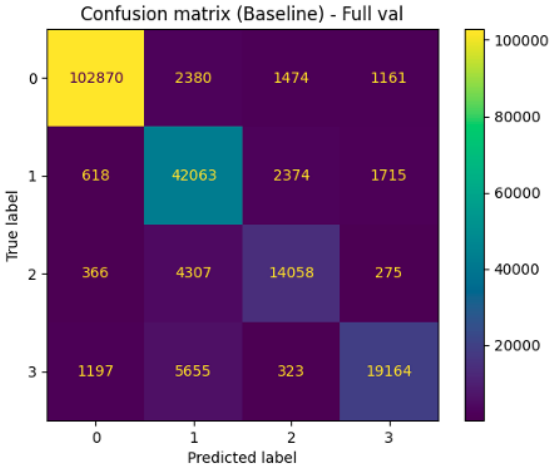

Figure 7: Confusion matrix for the uncalibrated CodeBERT baseline on the full validation set.

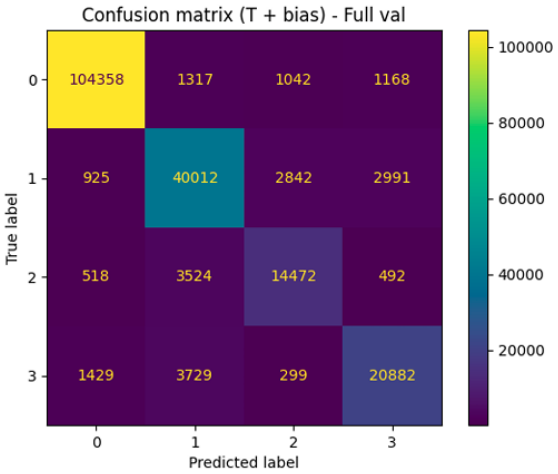

Figure 8: Confusion matrix after temperature-plus-bias calibration on the full validation set.

B.4 Confidence Distributions and Per-Class Calibration Errors

Figure 9 shows the distribution of maximum predicted confidence by true class before calibration, while Figure 10 reports per-class ECE and Brier scores. These diagnostics further illustrate

class-dependent overconfidence, particularly for the Machine-generated class.

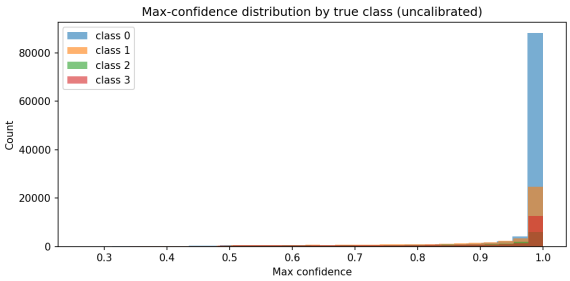

Figure 9: Distribution of maximum predicted confidence by true class before calibration.

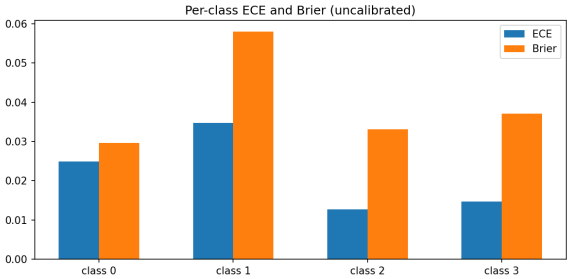

Figure 10: Per-class Expected Calibration Error (ECE) and Brier score before calibration.

B.5 Selective Routing Without Calibration

For completeness, Figure 11 reports Macro-F1 as a function of accepted fraction using uncalibrated confidence. Compared to the calibrated results in the main paper, uncalibrated confidence yields less reliable selective behavior.

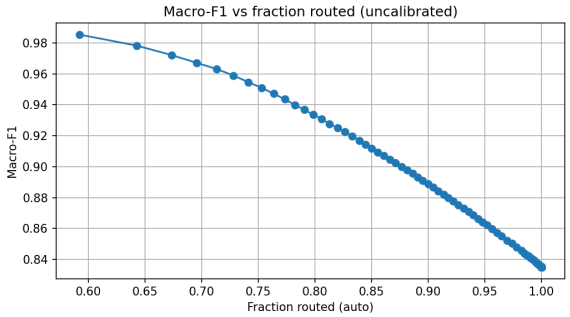

Figure 11: Macro-F1 versus fraction of predictions automatically accepted using uncalibrated confidence.