MINDS at SemEval-2026-Task 13: Robust Detection of Machine-Generated Code under Distribution Shift

Giorgia Rosalia Buccelli* **Antonella Coviello*** **Alexandra Elena Holota***
Marco Scaglione* **Simone Scalora*** **Claudio Savelli†**
Riccardo Coppola† **Flavio Giobergia†**
Politecnico di Torino
* {firstname.lastname}@studenti.polito.it
† {firstname.lastname}@polito.it

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 an empirical study across all subtasks, comparing a variety of approaches: frozen encoder representations, feature-based classifiers, fine-tuned transformer models, post-hoc calibration, and probability-level ensembling. Our results show a consistent generalisation gap: strong in-domain validation scores substantially overestimate performance on shifted test conditions. 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. These shifts can be expected to occur over time as new models are released, resulting in drifts both at the global and local level (Giobergia et al., 2025) (e.g., if the drift only concerns code written by a specific LLM, or in a specific language, or for specific functionalities). Prior work combines feature-based and embedding-based approaches (Nirob et al., 2026), yet robustness under distribution shift remains unresolved (Orel et al., 2025).

SemEval-2026 Task 13 (Orel et al., 2026) directly addresses the authorship identification problem 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, and ensembling. Beyond accuracy, reliable models also require proper calibration, since neural networks tend to be overconfident. Post-hoc methods such as temperature scaling improve probability alignment (Minderer et al., 2021), and selective, confidence-aware prediction further enhances reliability in risk-sensitive or human-in-the-loop settings (Zollo et al., 2024).

2 System Overview

This section describes the methodological choices adopted to address SemEval-2026 Task 13 across the three subtasks. Each subtask is evaluated primarily using Macro F1 score, which is the official target metric for the shared task. For Subtask C, we additionally report reliability-focused metrics and confidence-slice diagnostics, since probability quality is central to downstream use.

2.1 Common Experimental Pipeline

Across subtasks, we follow a consistent pipeline:

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.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 3080–3088

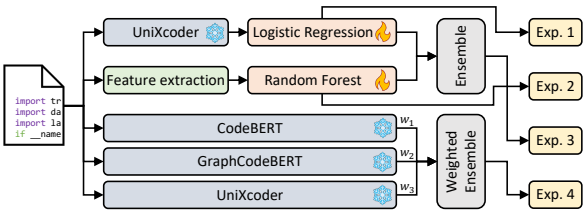

Figure 1: Pipeline of the four experiments for task A

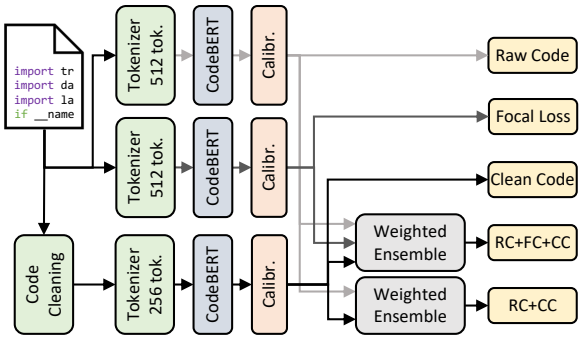

Figure 2: Architecture of the proposed ensemble system for Subtask B.

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

Ensembling (when applicable). When combining multiple models, we ensemble predicted probabilities. A calibration step is applied before ensembling, to allow for a meaningful aggregation of the outputs of models.

2.2 Subtask A: Binary Machine-Generated Code Detection

For Subtask A, we compare four complementary approaches (fig. 1) that reflect different inductive biases: (i) frozen neural representations, (ii) manual feature engineering, (iii) probability-level ensembling and (iv) multi-model frozen ensemble. We note that, due to hardware limitations, all Subtask A models are trained on a random subset of 20,000 samples (the full training set contains 500,000 samples). The subset is obtained by shuffling the full training set with a fixed random seed (42) and selecting the first 20,000 examples (simple random sampling, without stratification). This provides an evaluation of the performance under limited data availability.

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.

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.

Probability-level ensembles. In the ensemble strategy, we combine the frozen-encoder model and the feature-based classifier by weighted averaging of probabilities and a fixed 0.5 decision threshold.

Multi-model Frozen Ensemble. We combine three frozen encoder models: UniXcoder (Guo et al., 2022), GraphCodeBERT (Guo et al., 2020) and CodeBERT (Feng et al., 2020). Each model consists of a frozen encoder followed by a logistic regression classifier and the ensemble prediction is computed via weighted averaging.

2.3 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. The available training set contains 500,000 samples, with 100,000 samples for validation. 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. In our experiments, we train multiple CodeBERT-based classifiers that differ in (i) context length and (ii) loss function and select

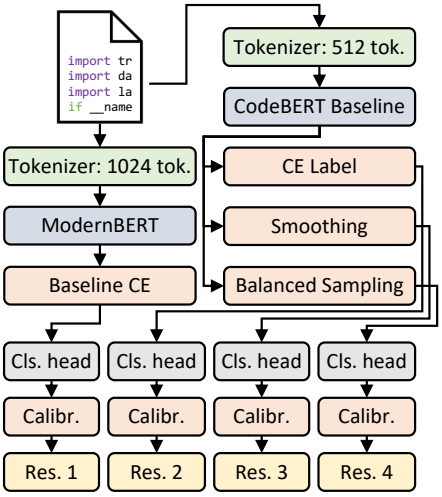

Figure 3: Pipeline of the four experiments for Subtask C.

the top performing models:

Raw Code: 512-token inputs on raw code, optimized with class-weighted cross-entropy to mitigate imbalance.

Focal Loss: 512-token inputs on raw code, optimized with focal loss to emphasize hard examples; gradient clipping is used for stability.

Clean Code: 256-token inputs on preprocessed code, optimized with class-weighted cross-entropy; designed to only focus on early snippet regions, with reduced stylistic information.

Calibration and ensembling. 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.

2.4 Subtask C: Hybrid Code Detection

Subtask C extends detection beyond a binary Human/LLM setting, by introducing *Hybrid* and *Adversarial* classes. This makes 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). A training set of 900,000 samples (and 200,000 samples for validation) is available for this Subtask.

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 (Warner et al., 2025)) 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 $b \approx [+0.56, -0.34, -0.10, +0.20]$, 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. 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 used to evaluate selective decision-making: high-confidence predictions can be accepted automatically while low-confidence cases can be deferred for human review.

3 Experimental Results

This section reports the empirical results for each subtask. We primarily report Macro F1, as required by the SemEval-2026 Task 13 leaderboard[1], and we highlight generalization gaps between in-domain validation and the provided test sample whenever available.

3.1 Subtask A

For the four methods discussed in Section 2.2, 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 Multi-Model Frozen obtaining the best validation score (0.9602), despite the fact that all models are trained on only $20,000$ training examples.

[1]Note that only Subtask C was submitted on the official leaderboard.

Model	Val F1	Test F1
Frozen Encoder + LR	0.9307	0.4641
Feature-Based RF	0.9405	0.3886
Ensemble Probability	0.9497	0.4523
Multi-Model Frozen	0.9602	0.4358

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.

Model	Validation F1	Test Sample F1
Raw Code (R.C.)	0.5802	0.3340
Focal Loss (F.L.)	0.5626	0.3321
Clean Code (C.C.)	0.3363	0.2235
R.C. + C.C.	0.5343	0.3459
R.C. + F.L. + C.C.	0.5859	0.3400

Table 2: Macro F1-score for Subtask B on the validation set and on the provided test sample.

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), the feature-based model (0.3886) and the multi-model frozen (0.4358) 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 (> 0.88), while recall for the human-written class collapses (0.23 for the feature-based model). As a result, many human-written snippets are misclassified as machine-generated, which disproportionately harms Macro F1. 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.

3.2 Subtask B

Table 2 reports Macro F1 for the main single-model configurations and ensembles . On validation data, the best single model (Raw Code) achieves Macro F1 of 0.5802, while the Focal Loss variant is slightly lower (0.5626). The Clean Code 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 (Raw Code + Focal Loss + Clean Code) achieves the highest validation Macro F1 (0.5859), slightly improving over the best single model (i.e., without ensembling). When tested on the reduced test set sample provided (consisting of 1000 samples), all methods experience a large drop in Macro F1 (e.g., Raw Code: 0.3340), consistent with the presence of unseen generators at test time. Among the ensemble variants, the two-model ensemble (Raw Code + Clean Code) attains the highest test-sample Macro F1 (0.3459), although differences across ensembles are small and may reflect variability in the provided test subset. Even though the performances on the complete test dataset may be different both for the widely larger number of samples and potential different class distribution, the ensembles should prove themselves better than single models even in bigger test datasets. The classes' representations in a bigger test set may favor one of the ensembles instead of another due to the slightly different biases of the ensembles.

Error analysis based on confusion matrices indicates that the Macro F1 degradation is driven primarily by recall collapse on minority LLM families (Figure 4). 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.

3.3 Subtask C

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 results are obtained by ModernBERT-1024, which achieves Accu-

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. Best result in **bold**, second best underlined. ↑ for "higher-is-better" metrics, ↓ for "lower-is-better" ones.

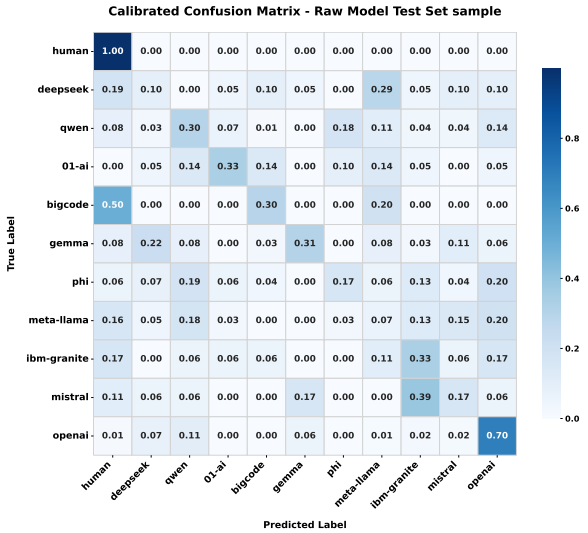

Figure 4: Confusion matrix for the Raw Code model on the test sample in Subtask B.

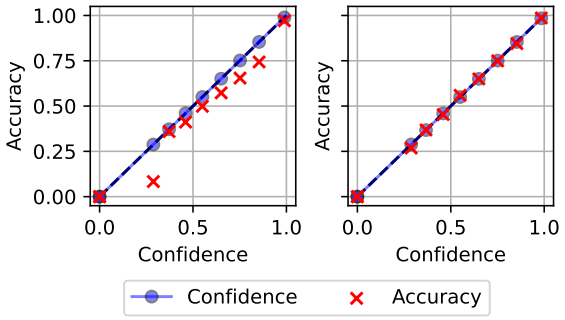

Figure 5: Reliability diagrams for Subtask C. (a) Uncalibrated CodeBERT baseline (ECE = 0.0335), showing systematic overconfidence. (b) After temperature scaling with additive bias (ECE = 0.0036), predicted confidence closely matches empirical accuracy.

racy 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.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%, thus retaining more information for long snippets. In particular, 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 5). 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 6) (Macro F1 vs. accepted fraction) support the use of calibrated confidence for selective decision-making, with low-confidence cases being natural candidates for deferral to human review.

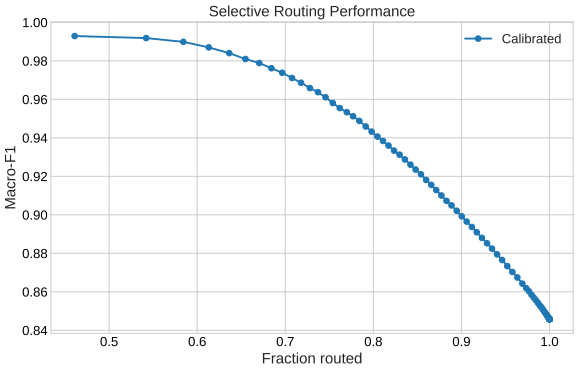

Figure 6: Macro-F1 as a function of the fraction of predictions automatically accepted using calibrated confidence.

4 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.

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.

References

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and 1 others. 2020. Codebert: A pre-trained model for programming and natural languages. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 1536–1547.

Flavio Giobergia, Eliana Pastor, Luca De Alfaro, and Elena Baralis. 2025. Detecting interpretable subgroup drifts. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 366–377.

Daya Guo, Shuai Lu, Nan Duan, Yanlin Wang, Ming Zhou, and Jian Yin. 2022. Unixcoder: Unified cross-modal pre-training for code representation. *arXiv preprint arXiv:2203.03850*.

Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie Liu, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, and 1 others. 2020. Graphcodebert: Pre-training code representations with data flow. *arXiv preprint arXiv:2009.08366*.

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.

Daniil Orel, Dilshod Azizov, Indraneil Paul, Yuxia Wang, Iryna Gurevych, and Preslav Nakov. 2026. SemEval-2026 Task 13: Detecting Machine-Generated Code with Multiple Programming Languages, Generators, and Application Scenarios. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547.

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.

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 7 summarizes confusion matrices for the main single-model and ensemble configurations under validation and test conditions. For the Raw Code 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 7 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 pre-

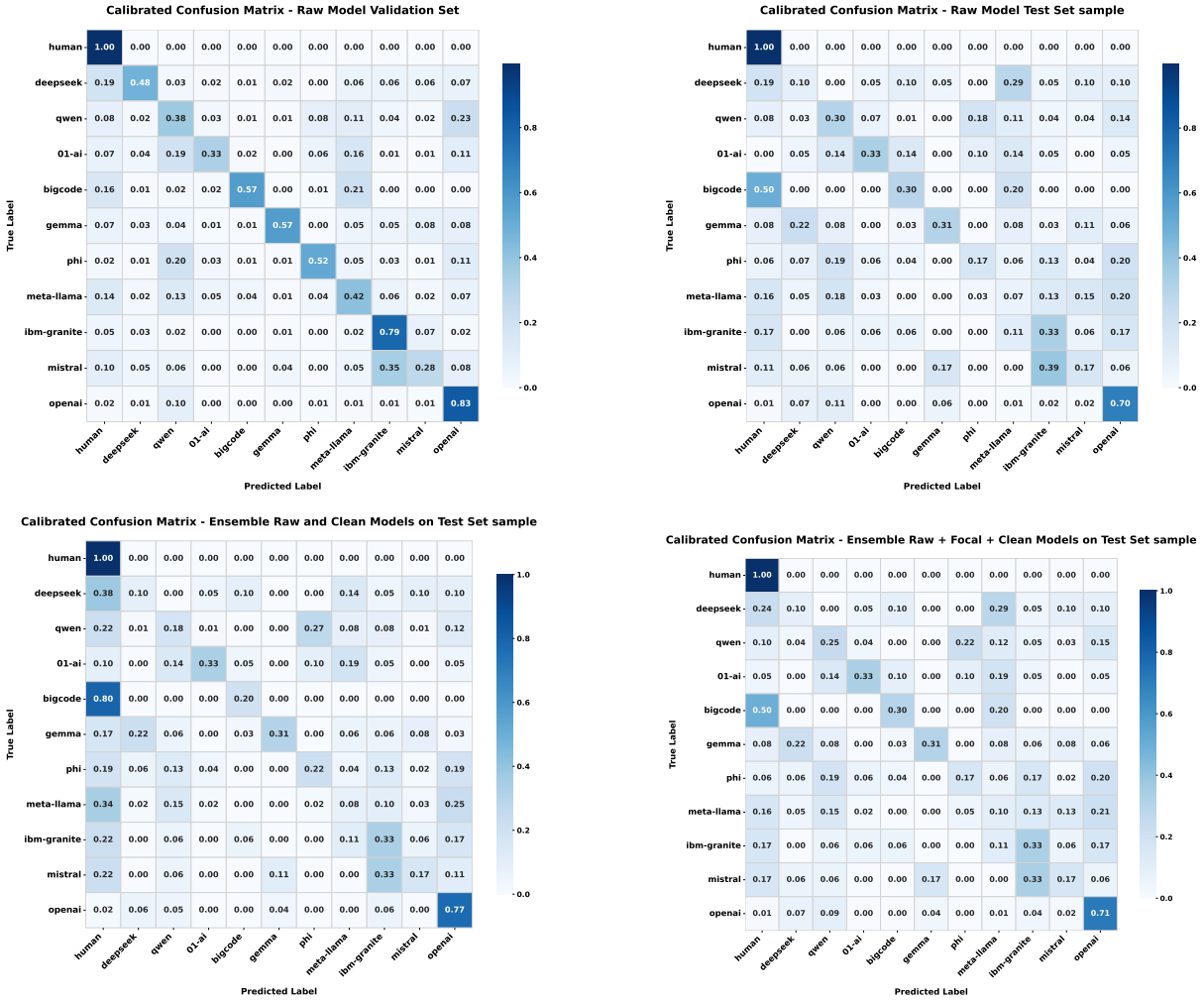

Figure 7: Confusion matrices for Subtask B under validation and test conditions. **Top:** Raw Code model on validation (left) and test sample (right), highlighting the strong shift-induced recall collapse for several minority LLM families. **Bottom:** Test-sample confusion matrices for the Raw Code + Clean Code ensemble (left) and the Raw Code + Focal Loss + Clean Code ensemble (right). While ensembling yields modest improvements in Macro F1, the overall structure of test-time errors remains largely unchanged.

dictions 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 8 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.

B.2 Calibration Bias Vector

Figure 9 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 10 and 11 report confusion matrices for the CodeBERT baseline before and after calibration

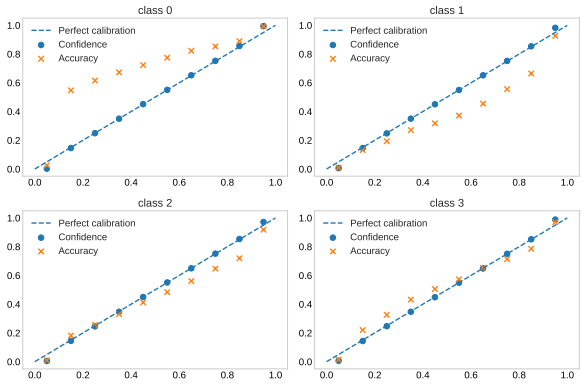

Figure 8: Per-class reliability diagrams before calibration, showing heterogeneous miscalibration patterns across classes.

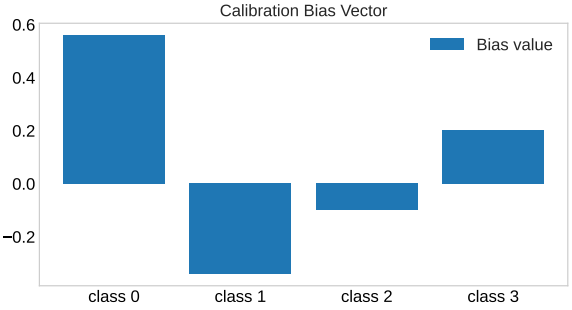

Figure 9: Additive class-wise bias vector learned during post-hoc calibration for Subtask C.

on the full validation set. These matrices illustrate how calibration affects confidence but does not substantially alter the argmax decision structure.

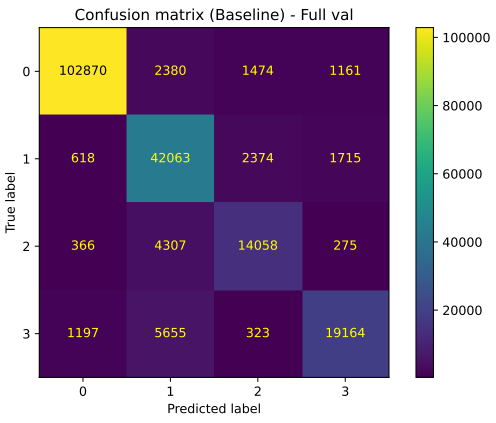

Figure 10: Confusion matrix for the uncalibrated CodeBERT baseline on the full validation set.

B.4 Confidence Distributions and Per-Class Calibration Errors

Figure 12 shows the distribution of maximum predicted confidence by true class before calibration, while Figure 13 reports per-class ECE and Brier scores. These diagnostics further illustrate class-dependent overconfidence, particularly for the Machine-generated class.

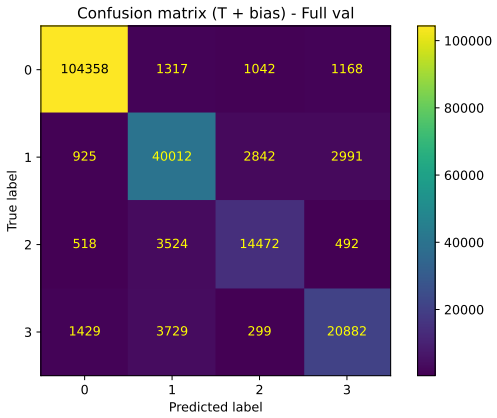

Figure 11: Confusion matrix after temperature-plus-bias calibration on the full validation set.

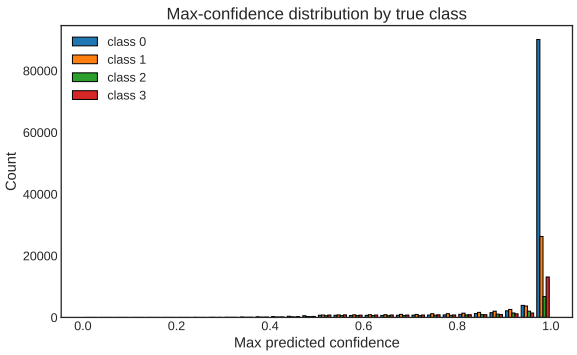

Figure 12: Distribution of maximum predicted confidence by true class before calibration.

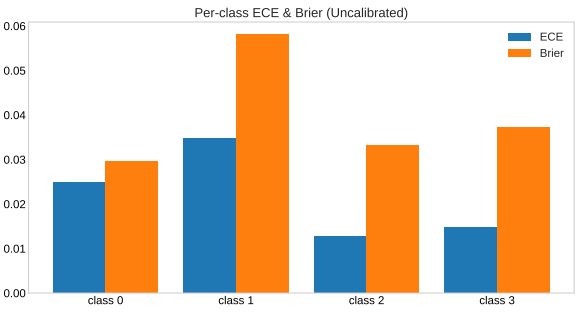

Figure 13: Per-class Expected Calibration Error (ECE) and Brier score before calibration.

B.5 Selective Routing Without Calibration

For completeness, Figure 14 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.

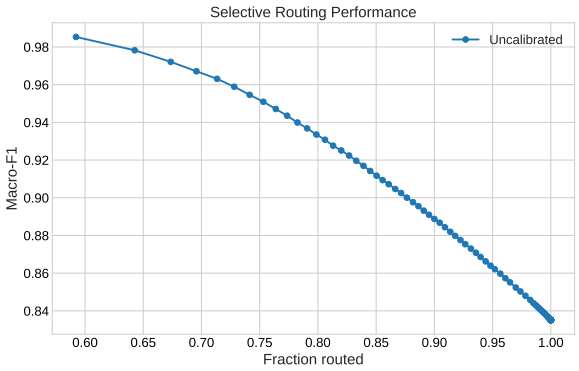

Figure 14: Macro-F1 versus fraction of predictions automatically accepted using uncalibrated confidence.