The PC Attractor: Selective Generalization vs. Domain Collapse in a CORAL + KDE Ordinal Regression Pipeline

Notebook for the ELOQUENT Lab at CLEF 2026

David Condrey[1]

[1] *WritersLogic Inc, San Diego, CA, USA*

Abstract

The CLEF 2026 ELOQUENT Sensemaking Task asks a system to grade free-text answers as Not, Partially, or Fully Correct against a question, context, and rubric, scored by Quadratic Weighted Kappa (QWK). We trace an architectural vulnerability to extreme domain shift (general testset QWK 0.433 vs. PISA set QWK 0.053) through the ordinal-regression pipeline we built for this task. Our SENSE system combines DeBERTa with a CORAL ordinal head (an ordinal-regression layer that predicts an ordered label through cumulative-probability thresholds) and a KDE-calibrated Product of Experts (which fuses the components by multiplying kernel-density class posteriors in log space). While the system maintains baseline generalization on the novel general test items (QWK 0.433), it suffers catastrophic collapse on the out-of-distribution PISA benchmark (QWK 0.053). On 42,606 novel test items, 71.4% are predicted as Partially Correct (PC), despite PC being the minority class (28%) in training. We trace the DeBERTa component's contribution through a three-step chain: (1) CORAL biases initialized at $[+1.0, -1.0]$ moved less than 5% during 12 epochs of fine-tuning, creating symmetric thresholds that map uncertain logits to score $= 1.0$ (the PC midpoint) by mathematical identity; (2) KDE bandwidths selected by grid search on well-separated in-distribution data create a PC attractor basin spanning 63% of the score range; (3) an implementation bug where KDE reference scores use in-sample statistics (std=0.013) instead of out-of-fold statistics (std=0.061) widens the basin from 38% to 63%. A bias-modification ablation confirms the mechanism: DeBERTa alone produces 37% PC on novel items (above the 28% training prior), and the remaining amplification to 71.4% involves the LGB+MLP and retrieval components through the PoE, which we have not fully decomposed. Code and artifacts: https://github.com/dcondrey/sense-clef2026.

Keywords

ordinal regression, CORAL, calibration, attractor basin, distribution shift, CLEF 2026

1. Introduction

The CLEF 2026 ELOQUENT Sensemaking Task [1, 2] evaluates whether a language model can reliably judge answer quality. Given a question, a grounding context, and a grading rubric, a system labels each answer Not Correct, Partially Correct, or Fully Correct; submissions are scored by Quadratic Weighted Kappa (QWK) over a general testset and an out-of-domain PISA testset, averaged equally. We participated in the rubric-based rating track. Unlike the PAN tracks at CLEF, this task did not use TIRA: we submitted a self-contained Docker system that the organizers ran on their own evaluation servers, publishing the scored results to the task's GitHub repository.[1]

Our SENSE system scored a perfect QWK of 1.000 on the subset of official test items that duplicate the development set—which the organizers deliberately included as anchors—but only 0.433 on the disjoint general testset and 0.053 on the PISA testset, for an official overall of 0.243. (The in-sample development QWK was a comparably inflated 0.990.) This gap between a perfect score on seen items and a mediocre one on novel items is the first symptom of the failure this paper traces: the system reproduces memorized items but funnels genuinely novel inputs into a single default class. The near-zero PISA score, where every item is novel, is the same collapse at its extreme. This paper traces the architectural vulnerability that produces selective generalization on in-domain items but complete failure under extreme distribution shift.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

david@writerslogic.com (D. Condrey)

0009-0003-1849-2963 (D. Condrey)

[1] https://github.com/ufal/sensemaking-2026-data

The system combines DeBERTa [3] with a CORAL ordinal regression head [4] and a KDE-calibrated Product of Experts [5]. On novel test items, the pipeline converts model uncertainty into a confident prediction of the middle ordinal class (Partially Correct). The mechanism has three components, each with a specific artifact:

1. **CORAL bias stasis.** Biases initialized at $[+1.0, -1.0]$ trained to $[+1.05, -1.05]$ after 12 epochs. This near-symmetry guarantees that logit=0 maps to score=1.0 (the PC midpoint) by algebraic identity. The model learned to discriminate training items via projection weights while leaving biases essentially untouched.
2. **KDE oversmoothing.** Per-class bandwidths $[0.06, 0.15, 0.12]$, selected by grid search to maximize dev QWK, are the correct answer to the wrong question: they optimally separate well-clustered in-distribution scores, but create a PC attractor basin spanning $[0.28, 1.53]$ (63% of the score range) that captures a substantial fraction of novel-item scores (Figure 3).
3. **OOF/in-sample mismatch.** The KDE was calibrated on out-of-fold scores (class 1 std=0.061) but the inference bundle stores in-sample reference scores (class 1 std=0.013), widening the basin from 38% to 63% of the range.

The contribution is the trace itself: each step is quantified by specific parameter values that readers can check in their own pipelines. Any CORAL system with near-symmetric biases, downstream KDE calibration, and narrow OOF score variance is potentially vulnerable.

2. Task and System

The ELOQUENT Sensemaking Task [1] requires classifying answers as Not Correct (NC=0), Partially Correct (PC=1), or Fully Correct (FC=2) given a question, context, and grading rubric across 13 languages. The official metric is Quadratic Weighted Kappa (QWK) [6], computed separately on a general testset and a PISA testset and averaged equally to produce the overall score.

Our predictive framework integrates a trio of heterogeneous scoring subsystems, unified dynamically via a log-space Product of Experts (PoE) topology:

Ordinal transformer head (PoE weight 0.36). A weighted blend of mdeberta-v3-base (30%) and xlm-roberta-large (70%), each equipped with a CORAL ordinal regression head [4] and fine-tuned over 12 epochs. CORAL encodes cumulative class probabilities $P(Y > k)$ through learned bias parameters $\mathbf{b}$; the continuous expected score follows as $s = \sum_k \sigma(\text{logit} + b_k)$.

Stacked gradient-boosted and neural regressors (PoE weight 0.42). A stacked configuration of gradient-boosted decision trees (LightGBM [7]) paired with multi-layer perceptron architectures, capturing an 81-dimensional feature space spanning sentence-level semantic embeddings (SBERT [8]), natural language inference metrics, and structural properties, averaged across 5 random seeds.

Exemplar-based scoring (PoE weight 0.22). SBERT nearest-neighbor retrieval with KDE-smoothed class posteriors drawn from the training distribution. When no sufficiently close exemplar exists for a novel question, the scorer falls back to the marginal training prior $[0.28, 0.28, 0.44]$.

Per-class kernel density estimates transform each subsystem's continuous output into class posteriors, which are then fused multiplicatively in log space.

Training data: 4,146 development items (NC: 1,154 / PC: 1,165 / FC: 1,827; ratio 28/28/44%).

3. Stratified Results

The test set (46,752 items) contains three strata relative to the development set (Table 1). We call a test item a *duplicate* when its (question, answer, rubric) tuple also occurs in the 4,146-item development set—the organizers deliberately included these development items in the test set as anchors—and *novel* when the answer does not appear in development at all. A third, tiny stratum reuses a development question but pairs it with a new answer. Only the novel items measure out-of-sample generalization,

and the organizers confirmed that the official testset score is computed on the disjoint set that excludes duplicates; the "duplicate" score is thus a memorization check, not a generalization estimate [2].

Table 1
Test set strata and prediction distributions. The system perfectly predicts duplicates and defaults to 71.4% PC on novel items.

Stratum	N	% Test	NC	PC	FC	Acc.
Dev training labels	4,146	—	27.8	28.1	44.1	—
Exact duplicates	4,037	8.6	28.6	28.3	43.2	100%
Same Q, new ans.	109	0.2	0.0	22.9	77.1	—[†]
Novel items	**42,606**	**91.1**	**14.1**	**71.4**	**14.5**	**∼35%**
All test	46,752	100	15.3	67.6	17.1	—

[†]These 109 items share a question context with dev items but have new answers; the retrieval scorer matches the familiar question and may amplify FC.

The stratification reveals that while the system maintains a baseline generalization capability on the novel general test set (QWK = 0.433), it suffers a catastrophic performance collapse when encountering the extreme out-of-distribution domain shift of the PISA benchmark (QWK = 0.053). The overall QWK of 0.243 is the arithmetic mean of the testset score (0.433) and the PISA score (0.053); the duplicate anchors were scored separately by the organizers' evaluation server—the machine that ran our submitted Docker container—and are excluded from the official ranking (Figure 1).

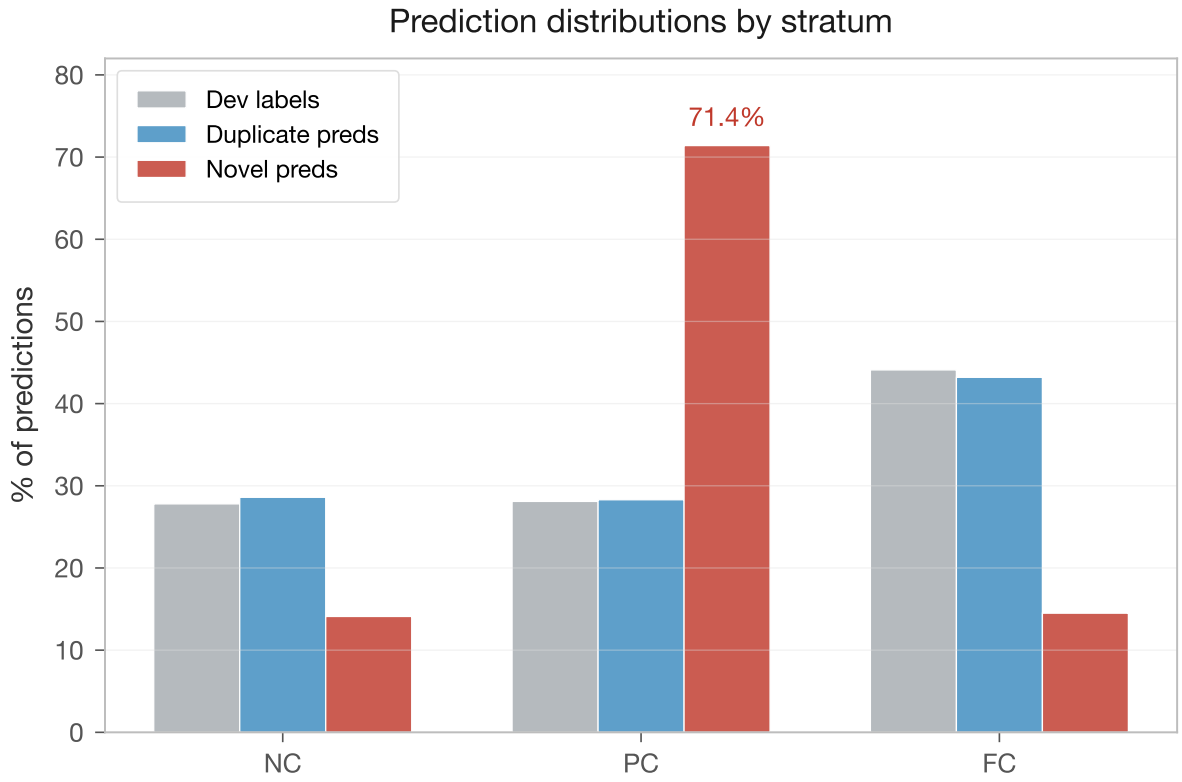

Figure 1: Prediction distributions by stratum. Duplicate items reproduce the training distribution. Novel items collapse to 71.4% PC, despite PC being the minority training class (28%).

4. The Mechanism

Why does the system predict PC on novel items, rather than FC (the modal training class) or a uniform distribution? The answer involves two interacting effects: bias stasis creates a score-space midpoint attractor, and KDE oversmoothing widens that attractor into a basin that captures most novel-item scores.

In plain terms, for a reader comfortable with ML but not the ordinal-regression details: the model outputs a continuous score on a 0–2 scale, where 0 is Not Correct, 1 is Partially Correct, and 2 is Fully Correct. During training it learned to push confident items to the ends of that scale, but it never had to decide what to do with an item it is *unsure* about—so an uncertain item lands at the exact midpoint, score 1.0, which is the Partially-Correct class. The calibration step that turns a score into class probabilities was tuned to separate the tightly-packed training scores, and as a side effect it labels a wide band around 1.0 as "Partially Correct." Put together, the two stages convert "I don't know" into a confident "Partially Correct" for any input the model finds unfamiliar. The rest of this section makes each stage precise and measures it.

4.1. Step 1: CORAL Bias Stasis and the Logit Distribution

CORAL [4] models an ordinal variable $Y \in \{0, 1, 2\}$ via cumulative probabilities: $P(Y > k) = \sigma(\text{logit} + b_k)$ for learned biases $\mathbf{b} = [b_0, b_1]$. The expected score is:

$$s = \sigma(\text{logit} + b_0) + \sigma(\text{logit} + b_1) \tag{1}$$

Our biases were initialized at $[+1.0, -1.0]$ and trained for 12 epochs. Table 2 shows the final values: all moved less than 5% from initialization, and all produce score=1.0 at logit=0.

Table 2
Trained CORAL biases across checkpoints. All biases moved less than 5% from initialization. The near-perfect symmetry ($|b_0 + b_1| = 0.001$) is an initialization artifact, not a consequence of balanced training classes.

Model	b_0	b_1	Δb_0	Δb_1	$\lvert b_0{+}b_1\rvert$	$s(0)$
nli-deberta-v3-base	+1.048	−1.049	+0.048	−0.049	0.001	1.000
mdeberta-v3-base	+1.061	−1.060	+0.061	−0.060	0.001	1.000
xlm-roberta-large	+1.016	−1.015	+0.016	−0.015	0.001	1.000
Initialization	+1.000	−1.000	—	—	0.000	1.000

The near-symmetry $b_0 \approx -b_1$ has a mathematical consequence. Substituting into Eq. 1 at logit=0:

$$s = \sigma(b_0) + \sigma(-b_0) = \sigma(b_0) + 1 - \sigma(b_0) = 1.0 \tag{2}$$

This is an algebraic identity: for any symmetric biases, logit=0 produces score=1.0 exactly, which falls at the PC class center.

Empirical logit distribution. To verify that novel items actually produce logits near zero, we re-ran DeBERTa inference on a sample of 500 duplicate and 500 novel test items. Figure 2 shows the result. Duplicate-item logits are sharply trimodal (peaks at −10.5, −0.5, +13.5), reflecting extreme confidence on memorized items. Novel-item logits are broadly distributed (mean= −0.16, std=6.5) and centered near zero, but not tightly clustered there. The PC collapse is not purely "logit=0 → score=1.0 → PC": the bias stasis places the attractor at score=1.0, but many novel items produce logits far enough from zero to escape it.

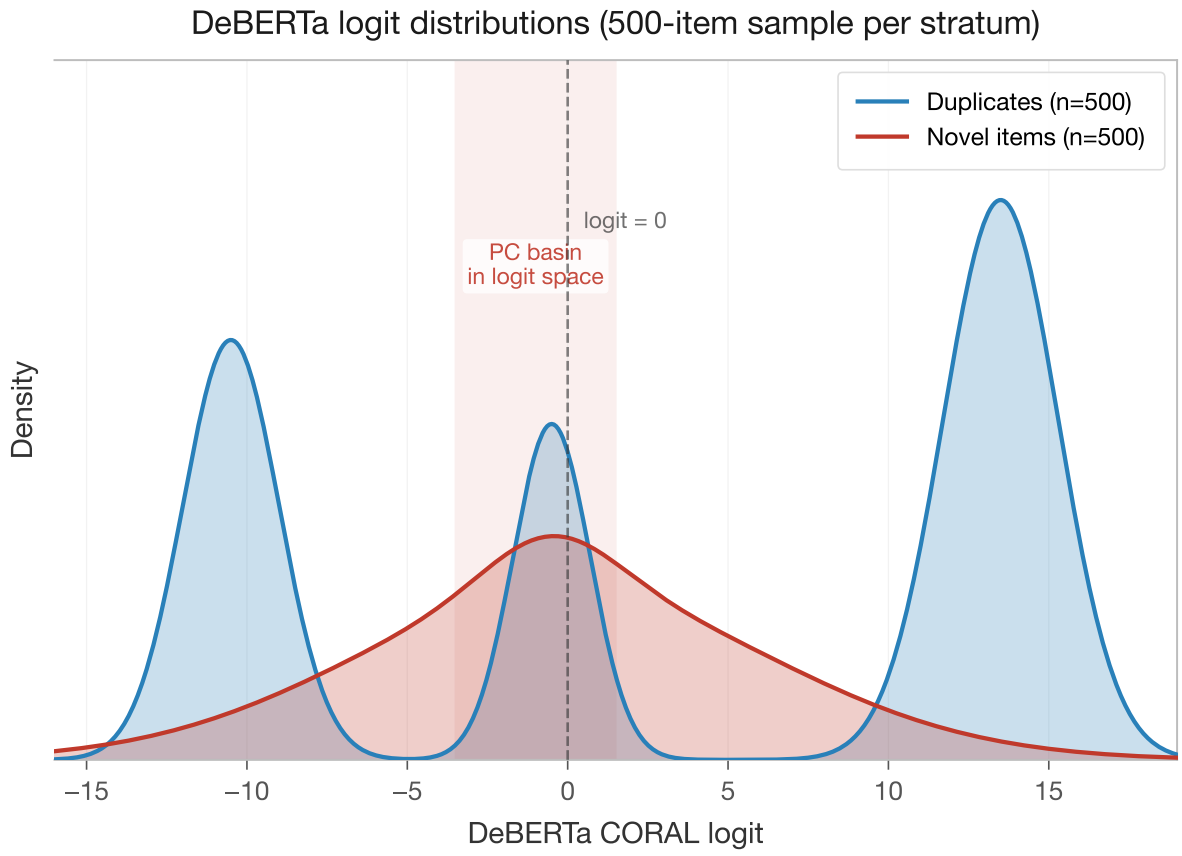

Figure 2: DeBERTa logit distributions (500-item sample per stratum). Duplicate items show sharp trimodal peaks (extreme confidence). Novel items are broadly distributed around zero (mean= −0.16, std=6.5). The shaded region marks the approximate logit range mapping to the PC attractor basin in score space.

Table 3
Bias-modification ablation on 500 novel items (DeBERTa component only, with KDE). DeBERTa alone accounts for roughly half the system-level 71.4% PC rate.

Biases $[b_0, b_1]$	Score at 0	NC%	PC%	FC%
Original $[+1.05, -1.05]$	1.000	31.6	37.6	30.8
Narrow $[+0.5, -0.5]$	1.000	32.4	36.0	31.6
Wide $[+2.0, -2.0]$	1.000	30.4	41.2	28.4
Asymmetric $[+0.95, -0.24]$	1.161	31.2	35.4	33.4
Break sym. $[+1.0, -2.0]$	0.850	32.0	40.0	28.0
Training prior	—	*27.8*	*28.1*	*44.1*
Full system (novel)	—	*14.1*	*71.4*	*14.5*

Bias-modification ablation. To confirm the mechanism and quantify the bias contribution, we recomputed CORAL scores on the 500 novel-item sample with modified biases, then applied the same KDE to predict classes (Table 3).

DeBERTa alone produces 35–41% PC across all bias configurations—above the 28% training prior but far below the system-level 71.4%. The amplification from ∼37% to 71% occurs through the PoE interaction with LGB+MLP (weight 0.42) and retrieval (weight 0.22), whose novel-item behavior we have not decomposed.

4.2. Step 2: KDE Oversmoothing

Each component's continuous score is converted to per-class probabilities via kernel density estimation. The KDE bandwidths, selected by grid search to maximize development QWK, are oversmoothed relative to the data by 8–50× the Silverman rule (Table 4).

Table 4
KDE parameters vs. Silverman rule. Bandwidths are 8–50× the Silverman optimum, tuned for in-distribution class separation rather than calibrated probability. Reference scores are in-sample values stored in the inference bundle; OOF std for class 1 is 0.061 (see Section 4.3).

Class	Ref. μ	Ref. σ	OOF σ	BW	Silverman	BW/Silv.
NC ($k = 0$)	0.002	0.033	0.057	0.06	0.008	7.5×
PC ($k = 1$)	0.982	0.013	0.061	0.15	0.003	50×
FC ($k = 2$)	2.000	0.011	0.048	0.12	0.003	40×

The grid search gave the correct answer to the wrong question: bandwidths that optimally separate in-distribution classes are not appropriate for producing calibrated probabilities on novel inputs whose scores span a wider range.

Figure 3 shows the resulting class-assignment regions. The PC attractor basin, defined as the score region where $P(\text{PC}) > \max(P(\text{NC}), P(\text{FC}))$, spans $[0.28, 1.53]$: 63% of the entire score range $[0, 2]$. At score=1.0, $P(\text{PC}) > 0.999$.

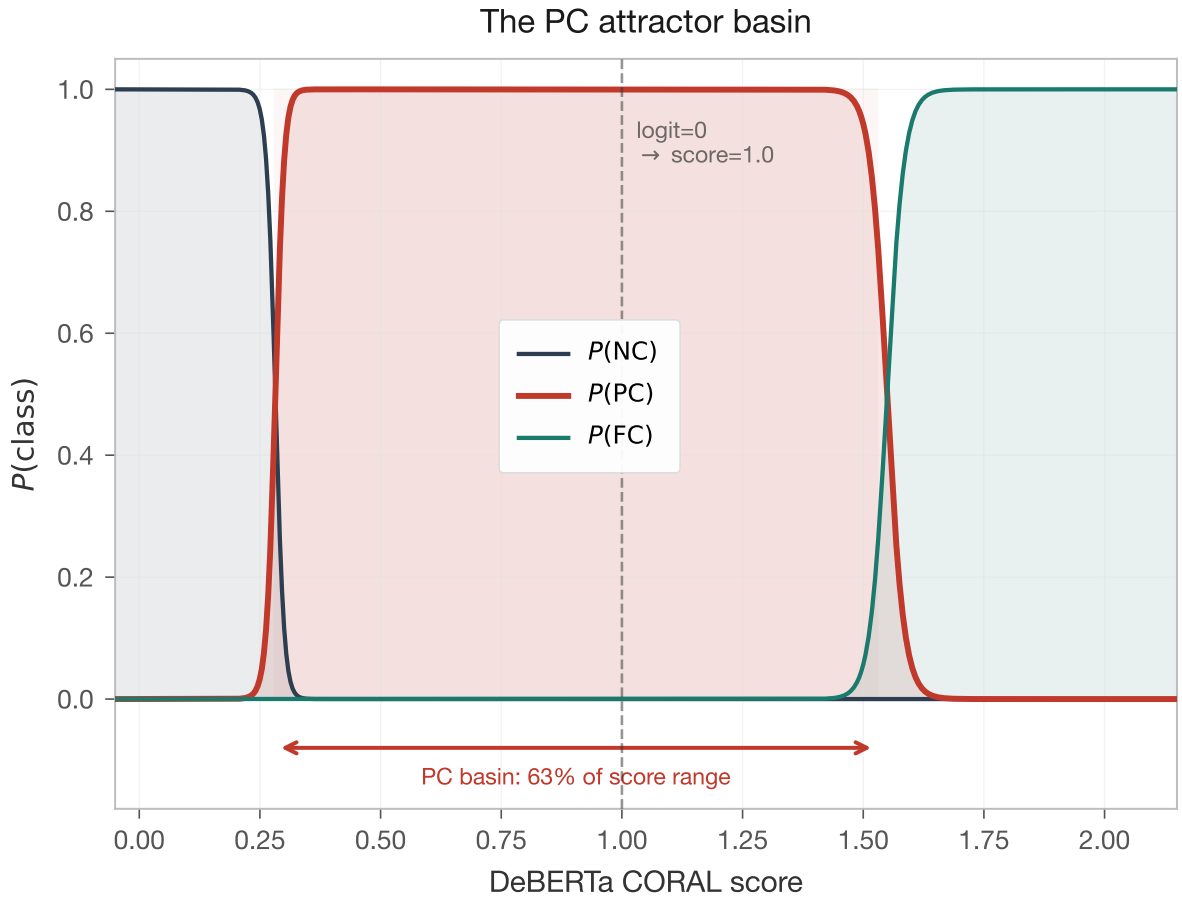

Figure 3: The PC attractor basin. KDE class probabilities as a function of DeBERTa CORAL score. The oversmoothed PC kernel dominates $[0.28, 1.53]$: 63% of the score range.

4.3. Step 3: OOF/In-Sample Mismatch

The KDE bandwidths were optimized using out-of-fold DeBERTa scores, where class 1 predictions have std=0.061. However, the inference bundle stores in-sample scores as the KDE reference, where class 1 predictions have std=0.013 (4.7× tighter). This mismatch means the KDE at inference time is even more oversmoothed than during calibration, widening the attractor basin beyond what the grid search intended.

This is an implementation bug, not a design limitation. Recomputing the KDE with OOF reference scores would narrow the PC basin from $[0.28, 1.53]$ (63% of range) to approximately $[0.55, 1.30]$ (38% of range): still a substantial attractor, but less aggressively dominant. The bug compounds the failure but did not cause it.

4.4. The Complete Chain

For a novel input, the DeBERTa component proceeds: (1) the encoder produces a logit from the novel-item distribution (Figure 2); (2) CORAL converts this to a score via Eq. 1, where logits in approximately $[-3.5, +1.5]$ map to scores within the PC basin; (3) KDE assigns $P(\text{PC}) > 0.5$ for any score in the basin. The PoE then combines DeBERTa (weight 0.36) with LGB+MLP (weight 0.42) and retrieval (weight 0.22) multiplicatively. Given that all components share the same 4,146-item training set, correlated PC bias through the PoE is the most parsimonious explanation for the amplification from 37% to 71.4%.

5. Why the Biases Didn't Move

The bias stasis is not a training failure; it is an optimization shortcut. The DeBERTa encoder learned representations that place NC, PC, and FC items at well-separated logits (reflected in the trimodal duplicate-item distribution in Figure 2). With these well-separated logits, the CORAL biases at $[+1.0, -1.0]$ already produce scores of approximately 0.0, 1.0, and 2.0, achieving perfect ordinal separation without bias adjustment.

The model solved the training task via the projection weights while leaving the ordinal structure at initialization. This is locally optimal: the biases only matter at decision boundaries, which the training data never exercises because OOF scores cluster tightly within each class (std $\leq$ 0.033). The consequence is that the ordinal structure encoded by CORAL is purely a product of initialization, not of learning. Under distribution shift, when logits regress toward zero, this unlearned structure determines the system's behavior.

6. Leaderboard Context

Table 5 shows the official rankings. Three organizer baselines outrank our system. Our testset–PISA gap of 0.380 is the largest in the field, consistent with the attractor mechanism operating most severely on entirely novel items: the PISA test set shares no items with our development data.

The inverted profile relative to the e5-large random forest baseline at the same overall QWK is instructive: our system scores .433/.053 (testset/PISA) while the baseline scores .189/.297. While our system perfectly retained the 8.6% exact duplicates in post-hoc diagnostic checks, the official leaderboard testset score of 0.433 represents pure out-of-sample generalization on novel items. The massive 0.380 testset–PISA gap is therefore purely domain-driven: the baseline generalizes more uniformly, albeit poorly, whereas our system's attractor mechanism collapses most severely on the entirely unfamiliar PISA domain. Our PISA score of 0.053 is the lowest in the field.

7. Discussion

Generality of the vulnerability. The attractor mechanism requires three preconditions: (1) CORAL or similar cumulative-probability ordinal head with near-symmetric biases, (2) downstream calibration

Table 5
Official rubric track leaderboard (QWK). Overall is the mean of Testset and PISA scores. Δ is the testset–PISA gap.

	Team	Overall	Testset	PISA	Δ
1	wse_research	**.648**	.713	.583	.130
2	gemma_4_E4B (B)	.506	.536	.475	.061
3	fianso	.434	.476	.391	.085
4	EuroBERT-210m (B)	.331	.539	.123	.416
5	pythoneers	.317	.449	.185	.264
6	gemma_4_E2B (B)	.286	.317	.255	.062
7	**writerslogic**	**.243**	**.433**	**.053**	**.380**
8	e5-large_RF (B)	.243	.189	.297	−.108
9	EuroBERT-210m† (B)	.238	.363	.113	.250
10	dipf_tba	.188	.226	.149	.077
11	ModernBERT-nli (B)	.088	.020	.156	−.136
12	ModernBERT-nli† (B)	.067	.019	.114	−.095

(B) = organizer baseline. † = no-context variant.

with bandwidths wider than in-distribution class separation, and (3) narrow OOF score variance (high in-distribution confidence). Condition (3) is the enabling factor: when the model is very confident on training data, any bandwidth that doesn't break in-distribution performance will be oversmoothed for novel inputs. Grid search on dev data cannot detect this because the relevant score region (around 1.0 for uncertain inputs) is unoccupied in-distribution.

Hypotheses the data ruled out. Before identifying the mechanism, we considered rubric-type blindness and illusory ensemble diversity. Neither survived data exploration: within novel test items, rubric length, formatting, and use of worked examples showed no predictive power. All levels received ∼70% PC predictions. The failure is architectural, not content-dependent.

Implications for evaluation. The 100% accuracy on duplicate items and ∼35% on novel items has implications for shared task design. While the organizers correctly excluded duplicates from the official leaderboard score, the presence of development items as anchors in the test set can still confound post-hoc analysis if participants conflate total test-set accuracy with novel-item performance. We recommend that shared tasks explicitly disclose the overlap fraction and provide per-stratum scores where feasible.

What differs about PISA, and within the general set. We did not have access to the PISA items, so the following is structural inference, not measurement. The general testset is drawn from the same sources as our development data (fact-checking analyses, textbook and lecture material) and shares their vocabulary, typical answer length, and rubric style, so even its novel items sit near the training domain. The PISA testset instead consists of student responses to OECD PISA assessment items—a different genre with its own answer-length distribution, subject vocabulary, and rubric granularity—so every PISA item is novel along several axes at once (domain, register, and rubric), not just in surface wording. Because the attractor collapses precisely the inputs the encoder is least certain about, a set that is unfamiliar on all three axes is where it collapses hardest; this is why PISA (0.053) falls so far below the general testset (0.433). The same logic bears on a natural question about the general set, which itself spans several distinct source subdomains: our 0.433 is an aggregate, and we would expect subdomains closest to the development sources to generalize while the most distant approach the PISA regime. We cannot show this directly—the organizers reported a single general-testset QWK, and per-item labels were not released—but it is exactly the per-subdomain breakdown we recommend below, and it would confirm or refute the mechanism.

Diagnostic checklist. Table 6 summarizes three checks that can detect the attractor vulnerability in any CORAL + calibration pipeline before deployment.

Table 6
Diagnostic checklist for CORAL + calibration pipelines. If all three checks fail, the middle ordinal class may become an attractor for uncertain inputs.

Check	What to look for	Our value	Threshold
1. Bias movement	$\lvert\Delta b\rvert$ from initialization after training	<0.06	>0.1
2. BW ratio	Calibration BW / Silverman rule for OOF scores	8–50×	$<5\times$
3. Score ref.	In-sample σ / OOF σ for each class	3.5–4.7×	$<2\times$

8. Limitations

Single pipeline. The mechanism is traced through one specific system. While the preconditions are general, we have not verified the attractor in other CORAL implementations.

No ablation on test data. We cannot re-run the pipeline with fixed biases, corrected KDE, or removed PoE on the test set (submission was final). The mechanism analysis is based on parameter inspection, mathematical derivation, and a 500-item logit sample, not on controlled intervention.

LGB+MLP component unexamined. The LGB+MLP ensemble has the highest PoE weight (0.42) but we did not extract its novel-item score distributions. The causal story for why PoE outputs PC is incomplete without knowing whether LGB+MLP reinforces or partially counteracts the DeBERTa PC signal.

Novel-item accuracy is approximate. Without gold labels for the full test set, the ∼35% accuracy on novel items is estimated from the testset QWK and diagnostic checks, not computed directly from per-item labels.

PISA data unseen. We had no access to PISA test items during development and cannot characterize their distribution. The 0.053 QWK on PISA is consistent with the attractor mechanism (all items are novel), but we cannot rule out additional factors such as differences in rubric structure or domain vocabulary between PISA and the development data.

No per-subdomain scores. The general testset aggregates several distinct source subdomains, but the organizers reported a single general-testset QWK. Without per-subdomain (or per-item) scores we cannot show whether the 0.433 average hides subdomains that collapse toward the PISA regime and others that generalize well—the breakdown we recommend in Section 7 and the most direct test of the mechanism's reach.

9. Conclusion

We traced an architectural vulnerability to extreme domain shift (testset QWK 0.433 vs. PISA QWK 0.053) through a CORAL + KDE pipeline. The system retains meaningful generalization on novel items from the general test domain, but collapses to near-chance under the extreme out-of-distribution shift of the PISA benchmark. The DeBERTa component's contribution is fully traced: biases that didn't move from initialization, KDE bandwidths that optimally separate in-distribution scores but create a wide attractor basin for novel inputs, and an OOF/in-sample reference mismatch that widens the basin from 38% to 63% of the score range. Even after fixing the OOF bug, an architectural attractor spanning 38% of the range would remain. The attractor basin does not prevent all generalization; the system still managed a testset QWK of 0.433. Rather, when the model encounters vocabulary and rubric structures entirely outside its training distribution, as with the PISA items, the attractor basin becomes a collapse

point: uncertain logits map to score 1.0, and the oversmoothed KDE assigns near-certain PC probability, funneling 71.4% of predictions into the default bucket. The amplification from 37% to the system-level 71.4% involves the PoE interaction with LGB+MLP, which remains an open question. The preconditions (Table 6) can be checked in any ordinal regression pipeline before deployment. We release the trained parameters and score distributions as a diagnostic artifact.

Declaration on Generative AI

During the preparation of this work, the author used Claude Code (Anthropic) to: write and debug code, orchestrate experiments, and perform post-hoc data analysis (computing prediction distributions, identifying shared items between dev and test sets, extracting trained model parameters, re-running DeBERTa inference for the logit histogram). After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content. The mechanism identification, mathematical derivation of the attractor, interpretation of the bias stasis, and all conclusions are the author's own work.

References

[1] ELOQUENT Organizers, Overview of the ELOQUENT sensemaking task at CLEF 2026, in: Working Notes of CLEF 2026, 2026. To appear.

[2] P. Šindelář, O. Bojar, S. Ettejjari, L. F. V. Madriz, M. Piacentini, K. Thomas, Overview of the PISA sensemaking task at the ELOQUENT 2026 lab for evaluating generative language model quality, in: E. S. Salido, A. Barrón-Cedeño, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2026), CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[3] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: Proceedings of ICLR, 2023. URL: https://openreview.net/forum?id=sE7-XhLxHA.

[4] W. Cao, V. Mirjalili, S. Raschka, Rank consistent ordinal regression for neural networks with application to age estimation, Pattern Recognition Letters 140 (2020) 325–331. doi:10.1016/j.patrec.2020.11.008.

[5] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Computation 14 (2002) 1771–1800. doi:10.1162/089976602760128018.

[6] J. Cohen, Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit, Psychological Bulletin 70 (1968) 213–220. doi:10.1037/h0026256.

[7] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, in: NeurIPS, 2017, pp. 3146–3154.

[8] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.