When Audit Quality Fails to Predict Downstream Utility: Synthetic-Data Selection for Low-Resource Classification*

Xuan-Bach Le[1†‡] **Phat T. Tran-Truong**[1†] **Son Ha Xuan**[2]
[1]Ho Chi Minh City University of Technology (HCMUT),
Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam
[2]RMIT University, Vietnam
{lexuanbach, phatttt}@hcmut.edu.vn ha.son@rmit.edu.vn

Abstract

Quality-aware synthetic-data selection rests on a proxy: examples an LLM judge rates as good should also help a downstream model learn. The proxy grades examples one at a time, yet a selector returns a set, and a classifier learns from set-level properties no per-example score records. We test it in a controlled replay in low-resource African-language classification, over four languages, two tasks, and seven selectors, six retaining 64 examples per cell. It does not survive. CoSDA-V2, our counterfactual-audit selector, earns the highest judged label correctness (0.904 against 0.767 for naive) and the lowest shortcut score we measure, yet AlpaGasus leads downstream Macro-F1 (0.202 against 0.163), and across three seeds too (0.193 against 0.157).

Two probes locate the cost, and both point away from the ranking rule. Adding the audit's strict gate to AlpaGasus, the choice CoSDA-V2 declines to make, costs 0.029 Macro-F1, almost all in the one cell where 31 of 64 requested candidates survive. LESS, which optimises downstream fit directly, falls below naive because its unconstrained top-k collapses the label distribution, and recovers under quotas. What a selector must retain matters more at this budget than how it ranks. Audit quality describes the pool a selector returns, not what a model learns from it, so evaluations should report both on the same retained sets.

1 Introduction

Synthetic data makes supervision cheaper, but it creates a data-selection problem. A generator produces more candidates than a downstream learner can absorb, and those candidates vary in label correctness, redundancy, leakage risk, shortcut structure, and coverage. Recent methods score candidates before training (Chen et al., 2024; Liu et al., 2024; Xia et al., 2024; Wettig et al., 2024), and curated-small-data work likewise trusts a small hand-selected set (Zhou et al., 2023). They share an assumption worth stating plainly, that ranking examples well is the same task as assembling a training set. We ask where the two come apart, in low-resource multilingual classification, where a few dozen retained examples decide what a model learns.

Our test is a budget-controlled replay over eight task-language cells from MasakhaNEWS (Adelani et al., 2023) and AfriSenti (Muhammad et al., 2023), covering news topic and sentiment classification in Amharic, Hausa, Swahili, and Yoruba. Every selector reads one shared audit record per candidate (Figure 1), so pools differ by selection rule alone. Against naive augmentation we compare AlpaGasus-style quality ranking, DEITA-style ranking, a utility-plus-diversity control standing in for a validation-aware selector (Xia et al., 2024), our CoSDA-EQ and CoSDA-V2, and AlpaGasus+HR, which adds the audit's gate to a fixed ranking. We present this as a controlled case study, not a general law.

What we found. The two rankings disagree. Every quality-aware selector beats naive on LC, Q, and hard-reject, and CoSDA-V2 leads on LC at 0.904 against 0.767, a relative gain of 17.9%, with the lowest shortcut score. Macro-F1 then reorders the field. Inside each of the eight cells, the Spearman between LC and Macro-F1 averages ρ=0.04 over our five main selectors, and $\rho = -0.002$ over seven across three seeds. Two of the three channels using no judge field reproduce the disagreement, so it is not an artifact of LLM scoring. Our audit and our classifier agree on which pools are clean and disagree on which pools teach.

Why it happens. CoSDA-V2 applies no hard-reject gate, so we measured what one costs by

*Code, audit records, retained pools, and the claim ledger: https://github.com/lexuanbach/CoSDA.

† Equal contribution.

‡ Corresponding author.

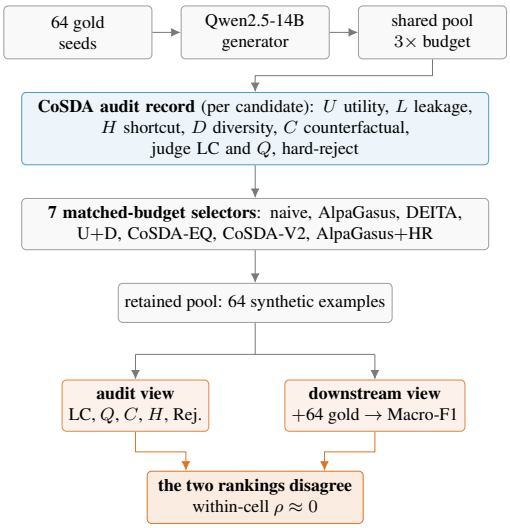

Figure 1: Every selector reads the same audit record and returns a pool of 64, except AlpaGasus+HR where the gate leaves fewer. We measure that pool twice, by audit and by training, and the two answers disagree.

adding the audit's strict gate to AlpaGasus's ranking. Over three seeds it lowers mean Macro-F1 by 0.029, dominated by Amharic news, where the gate admits only 31 of the 64 candidates the budget calls for and Macro-F1 falls by 0.252, while across the other seven cells the net effect is +0.003. A second probe agrees. LESS, which optimises downstream fit directly, finishes below naive because its unconstrained top-k leaves 77% of the retained pool in one label, and recovers to 0.278 under quotas. At this budget what a selector must retain outweighs how it ranks (§5.1).

Contributions. **(Method)** We formalise five per-example audit channels for synthetic data, namely utility, leakage, shortcut, diversity, and counterfactual consistency, with a fixed hard-reject decision (§2). **(Study)** We run a deterministic 8-cell replay over seven selectors at a common budget, report audit and downstream metrics on the same selected sets, show the gap survives bootstrap resampling, leave-one-cell-out, and non-degenerate-cell restriction, and measure what the audit's strict gate costs (§3 to §5). **(Artifact)** We release the per-candidate audit records, retained pools, and a claim ledger tying every reported aggregate to its source (§A).

2 CoSDA: A Counterfactual Audit

We use CoSDA to make a selected pool inspectable before it reaches training. Every selector we compare reads the same per-candidate record, so any difference between their pools is interpretable against a shared trace.

Setup. For task t, language ℓ, and gold budget b, let $G_{t,\ell,b}$ be the gold seed set. The generator returns a candidate pool $X_{t,\ell,b}$ with multiplier m, and each x_i carries a proposed label y_i, generator metadata, source seed IDs, and a counterfactual pair ID. Our replay uses Qwen2.5-14B-Instruct (Team, 2024) at p=0.95 and T=0.8 as generator, Qwen2.5-32B-Instruct-AWQ as judge, and XLM-RoBERTa base (Conneau et al., 2020) as classifier. The generation prompt fixes task, language, label schema, seed examples, output format, and a forbidden-behaviour clause forbidding copied seed or held-out text and literal English-template translations. Appendix B reproduces the template, whose length hint branches by task.

Counterfactual pair construction. CoSDA builds task-specific minimal pairs $c_i = (x_i, x_i')$ to ask whether a candidate behaves like a labelled example. The generator flips the topic or sentiment cue while holding language, register, and non-label content fixed, and the expected flipped label y_i' feeds C_i. The C_i gate pass rate, $C_i \geq 0.6$, varies by language at 33.4% for Amharic, 52.9% Hausa, 55.2% Swahili, and 58.9% Yoruba. Amharic therefore enters selection with a much smaller usable counterfactual pool, and that asymmetry matters in §5.1. The validator is deliberately task-specific, so extending CoSDA to other task families means defining the corresponding transformation.

Audit channels. Each candidate receives five normalised scores in $[0, 1]$:

$$U_i = \tfrac{1}{3}(\sigma_{\text{gen}} + a_{\text{teach}} + v_{\text{nbr}}), \quad (1)$$

$$L_i = \max\{E_i, J_i, B_i\}, \quad (2)$$

$$H_i = \textstyle\sum_k w_k \phi_k(x_i), \quad (3)$$

$$D_i = 1 - \max_{j \in G} \cos(\mathbf{e}_i, \mathbf{e}_j), \quad (4)$$

$$C_i = v_{\text{cf}}\big(0.75 + 0.25\, p_\theta(y_i' \mid x_i')\big). \quad (5)$$

Utility U_i averages the judge's quality score (or generator confidence where no judge ran), the confidence a LaBSE-centroid teacher fit on $G_{t,\ell,b}$ assigns to y_i, and a k=5 gold-neighbour label vote in LaBSE space (Feng et al., 2022). A centroid teacher is used because fine-tuning one on 64 examples is unstable. *Leakage* L_i is the maximum of an exact 10-gram overlap indicator E_i against the dev and test union, the largest 5-gram shingle Jaccard J_i against any held-out item (Broder, 1997), and B_i, a LaBSE cosine rescaled so only

similarity above 0.85 contributes. *Shortcut* H_i is a bounded artifact-risk score, a fixed weighted sum $\sum_k w_k \phi_k$ over label-word hits, template markers, length atypicality, an entity proxy, and punctuation ratio. It is a heuristic rather than a trained probe, so it flags surface regularities rather than certifying their absence. *Diversity* D_i is one minus the maximum LaBSE cosine of x_i against the gold seeds, computed once and not updated as selection proceeds. *Counterfactual consistency* C_i multiplies the judge's counterfactual-validity rating v_{cf} by a soft teacher term rewarding probability mass on the expected flipped label y_i', soft rather than a hard indicator because the teacher is noisy at this budget.

The same record stores the two judge fields quality-only selectors consume, label correctness and quality. Label correctness sits outside S_i, while quality enters it through U_i, and we keep both so judge-facing quality and downstream utility can be compared on identical pools. Every audit metric in §4 is averaged over each selector's *retained* pool, 64 examples except for the two underfilled AlpaGasus+HR cells, so audit channels and Macro-F1 describe the same training set. A candidate violates the gate when it crosses a fixed leakage, shortcut, or counterfactual-consistency threshold, so the rate we report is the *residual* violation rate inside it.

The hard-reject gate and the soft score. The audit defines a strict gate,

$$\text{reject}(x_i) = \mathbf{1}[L_i{>}\tau_L \vee C_i{<}\tau_C \vee H_i{>}\tau_H],$$

with τ_L=0.15, τ_C=0.60, and τ_H=0.70 fixed before the replay, alongside a five-term soft score

$$S_i = \alpha_U U_i + \alpha_C C_i + \alpha_D D_i - \alpha_L L_i - \alpha_H H_i,$$

with α_U=1.0, α_C=0.75, α_D=0.50, α_L=1.0, α_H=0.75. Both belong to the audit record, and our two selectors differ in ranking rule and in how much authority each gives the gate (§5.1).

CoSDA-EQ and CoSDA-V2. CoSDA-EQ treats rejection as a preference rather than a veto, ranking gate-passing candidates by S_i and backfilling with the rejected ones that score highest on the same S_i.

CoSDA-V2 drops hard rejection altogether. Every candidate stays eligible, and the thresholds reenter only as graded penalties and small bonuses,

$$S_i^{\text{V2}} = \mathcal{Q}_i + A_i + 0.25\,\beta_i^{\text{s}} + 0.10\,\beta_i^{\text{r}} - R_i,$$

with a judge block $\mathcal{Q}_i = 1.25\,\text{LC}_i + 0.85\,Q_i + 0.45\,f_i + 0.35\,\kappa_i + 0.35\,U_i$ over judged label correctness, quality, fluency, and cultural plausibility, an audit block $A_i = 0.65\,C_i + 0.45\,D_i$, and a risk penalty

$$R_i = 1.4L_i + H_i + 1.1\lambda_i + 0.55(\tau_C - C_i)_+ + 0.45(L_i - \tau_L)_+ + 0.35(H_i - \tau_H)_+,$$

where λ_i is the judge's leakage suspicion and $(\cdot)_+$ the positive part, and $\beta_i^{\text{s}}, \beta_i^{\text{r}}$ mark candidates clearing the strict gate and a relaxed one ($\tau_C \to 0.45$). We built V2 this way because the earlier hard-gated variants preserved precision but destroyed coverage in the cells with the smallest usable pools. Since V2 never removes a violator, its residual hard-reject rate can stay above zero.

Each selector but naive retains its top k under per-label quotas rounded to the gold prior and a cap of three candidates per source seed, with overflow deferred and used as backfill. We calibrated τ_L and the α weights on a held-out Setswana split outside the replay and fixed τ_H and τ_C by protocol. The S_i^{V2} coefficients were set by hand before the replay and were not tuned on the reported cells. We release neither calibration run, so all of these should be read as fixed protocol choices rather than reproducible derivations (see Limitations).

3 Experimental Setup

Tasks and languages. We evaluate an 8-cell replay over MasakhaNEWS news topic classification (Adelani et al., 2023) and AfriSenti sentiment classification (Muhammad et al., 2023), in Amharic, Hausa, Swahili, and Yoruba. Macro-F1 is the metric in every cell, and Appendix Table 7 gives splits and provenance. We treat each cell independently and aggregate after inspecting per-cell outcomes, since three of our eight turn out near-degenerate.

Low-resource protocol. Every cell gives us b=64 gold examples and a generation multiplier m=3, so each selector chooses 64 synthetic examples from roughly $3b$ candidates. Our main replay is deterministic at seed 13, which means every selector draws from the same trace-backed candidate pool in each cell. That makes the comparison a within-pool test of selection criteria rather than of independently sampled generations.

Generation and scoring. All selectors read a shared candidate pool per cell, scored per §2

with generator and judge quality signals, centroid-teacher agreement, LaBSE-based diversity and leakage checks (Feng et al., 2022), shingle-overlap near-duplicate checks (Broder, 1997), shortcut features, and counterfactual scores.

Selectors. Seven selectors read that pool. All but naive apply the same label-quota and source-seed constraints. Six retain 64 examples in every cell. AlpaGasus+HR retains 31 and 50 in two cells and 64 elsewhere. **(1) Naive** shuffles the pool at the cell seed and keeps 64 candidates, using no audit signal and, unlike the others, no balance constraint. **(2) AlpaGasus-style** (Chen et al., 2024) keeps the top 64 by judged quality. **(3) DEITA-style** (Liu et al., 2024) keeps the top 64 by judged quality, embedding diversity, and a length-based complexity proxy. **(4) U+D control** ranks by $U_i + 0.25 D_i$, a gold-fit proxy. §4.4 additionally implements LESS itself (Xia et al., 2024) and compares the two. AlpaGasus and DEITA are budget-matched reimplementations of the published *criteria*. The U+D row is a local control, not a reimplementation of LESS. **(5) CoSDA-EQ** ranks gate-passing candidates by S_i and backfills with the least risky rejected ones. **(6) CoSDA-V2** is our budget-preserving reranker, which applies no gate and takes the top 64 by S_i^{V2} (§2). **(7) AlpaGasus+HR** is the ablation selector for §5.1. It applies the audit's strict gate, the one CoSDA-V2 declines to use, and ranks the survivors by AlpaGasus's score. We also fine-tune a Gold-only reference on the gold examples alone and report it in Appendix C, keeping it out of the main comparison since it has no synthetic pool to audit.

Models and metrics. We fine-tune the deterministic Hugging Face classification path used in the CoSDA replay and report Macro-F1. In every cell we train on the union of the 64 gold seeds and the selector's retained synthetic examples, giving 128 rows for every selector except AlpaGasus+HR, whose gate leaves only 31 and 50 candidates in two cells. The gold seeds are identical across selectors, so the retained synthetic set is the only thing that varies. We compute audit metrics over each selector's retained examples, namely LC, Q, counterfactual consistency C, leakage L, shortcut H, and residual hard-reject rate. The main replay uses XLM-RoBERTa base (Conneau et al., 2020), and §4.4 repeats the fit with AfroXLMR base (Alabi et al., 2022) on the same pools.

Selector	LC↑	Q↑	C ↑	H ↓	Rej.↓	F1↑
Naive	0.767	0.662	0.587	0.066	0.486	0.167
AlpaGasus	0.890	**0.749**	0.653	0.065	0.246	**0.202**
DEITA	0.869	0.737	0.637	0.070	0.299	0.128
CoSDA-EQ	0.836	0.718	**0.686**	0.058	**0.109**	0.133
CoSDA-V2	**0.904**	0.746	0.674	**0.057**	0.162	0.163
Spearman ρ (LC-rank vs. F1-rank)						+0.10

Table 1: Audit channels and downstream utility on the same retained pools, averaged over 8 cells (seed 13). Rej. is the residual hard-reject rate. Every quality-aware selector beats naive on LC, Q, C, and Rej., but only AlpaGasus also wins downstream.

Extended replay. Our new selectors, the multi-seed re-fit, and the second backbone all reuse the seed-13 retained pools, so audit channels cannot move and only the downstream fit is recomputed. The extension comes to 224 further fine-tunes, covering 7 selectors over 8 cells at seeds $13/21/42$ on XLM-R, plus $7{\times}8$ on AfroXLMR at seed 13. It ran on different hardware from the original replay, and per-selector Macro-F1 moves by up to 0.048 against Table 1, enough to reorder the selectors at the shared seed. We therefore read each pipeline on its own terms and never pool or compare them across.

4 Results: The Quality-to-Utility Gap

We compare selectors using two judge-derived quantities, each averaged over that selector's retained pool: *label correctness* (LC), the judge's confidence that y_i is right for x_i, and *quality* (Q), its holistic rating of suitability for fine-tuning. Ranking selectors by these audit scores and then by Macro-F1 gives two orderings that disagree, and the disagreement persists across the checks we could run, though its size depends on backbone and judge.

4.1 The gap

Every quality-aware selector in Table 1 improves on naive on LC, Q, C, and residual hard-reject, and all but DEITA also on shortcut H, broadly what the selection literature predicts. Our hard-reject rate is the *residual* rate among retained examples, so it diagnoses what a selector hands to training rather than how much it discarded (§2). Macro-F1 imposes a different order. AlpaGasus leads at $+0.036$ over naive, while DEITA, CoSDA-EQ, and CoSDA-V2 all fall below naive despite carrying better audit profiles. Our LC order runs V2 > AlpaGasus > DEITA > EQ > naive. Our Macro-F1 or-

der runs AlpaGasus > naive > V2 > EQ > DEITA. CoSDA-EQ is the only selector with the same rank in both orderings. We measure the selector-level Spearman between mean LC and mean Macro-F1 at ρ=0.10, and between hard-reject and Macro-F1 at ρ=0.20, but with five selectors we read these as the rank orders above rather than as evidence, and we rest the argument on the within-cell statistic.

The disagreement is not confined to judge-derived scores. Three channels use no judge field. Oriented so that higher is better, diversity is inversely ranked with Macro-F1 ($\rho = -1.00$) and leakage likewise (-0.40), while shortcut is weakly aligned ($+0.20$). With five selectors these are descriptive checks rather than evidence that every audit channel diverges from utility, and we do not combine them because their scales differ by an order of magnitude. Counterfactual consistency is judge-assisted, since C_i scales the judge's validity rating, and the residual reject rate inherits that. Whatever the audit measures is a property of the pool the classifier does not reward.

4.2 Audit leader vs. downstream leader

Setting the two leaders side by side on the same eight cells (Appendix Table 11), CoSDA-V2 leads three of the four audit rows, on LC, residual hard-reject, and C, trailing AlpaGasus by 0.003 on Q, whereas AlpaGasus leads Macro-F1 by 0.039, from a 3–3 cell split with 2 ties.

Within-cell ranks. Aggregating across eight heterogeneous cells could induce an aggregate disagreement absent within cells, so we compute the Spearman *inside* each cell (Figure 2), where every selector faces the same pool and test set. Four cells come out positive, including Swahili news at $+0.87$ and Hausa sentiment at $+0.90$, and four negative, with mean $+0.04$ and median 0.00. The disagreement is present within cells, so it is our primary statistic.

4.3 Robustness

Per-cell heterogeneity. The aggregate -0.004 mean for CoSDA-V2 against naive hides structure. Two cells are strongly positive, Swahili news at $+0.129$ and Hausa sentiment at $+0.206$, and in three the five selectors span under 0.07 Macro-F1, carrying almost no information (Appendix Table 10, Figure 5). Dropping Hausa sentiment, the single large positive cell, moves the mean to -0.034, in line with DEITA and CoSDA-EQ (Appendix Table 13). We built that average out of cell effects of opposite sign, not a consistent selector advantage, rather than a near-miss against naive.

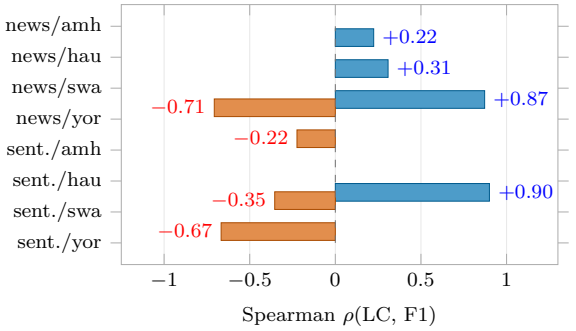

Figure 2: Within-cell Spearman between judged label correctness and Macro-F1, computed across selectors inside each cell. Four cells are positive and four negative, with mean $+0.04$.

Bootstrap CI. Resampling the 8 cells gives 95% intervals on the mean delta over naive of $[-0.079, +0.078]$ for CoSDA-V2 and $[-0.022, +0.093]$ for AlpaGasus. On the five non-degenerate cells, $[-0.114, +0.133]$ at mean $+0.007$ and $[-0.015, +0.137]$ at mean $+0.070$. Both intervals contain zero, which is why we make no claim that any selector beats naive downstream. The disagreement between the two rankings is the quantity this design can test, and it remains descriptive under these resamples.

Non-degenerate cells. Dropping those three leaves five informative cells: 0.226 for AlpaGasus, 0.163 for V2, 0.156 for naive, 0.114 for EQ, 0.103 for DEITA, 0.087 for Gold-only. The LC-to-F1 Spearman climbs to $+0.50$, so audit and utility align better here than across all eight cells, yet the AlpaGasus-over-V2 inversion widens. We read that as the inversion surviving where the classifier is learning, not as stronger correlation evidence.

Audit improvements. CoSDA-V2 lifts LC by 17.9% relative to naive, and CoSDA-EQ cuts residual hard-reject by 77.5% relative while raising C from 0.587 to 0.686. These are measured properties under the defined audit channels. §5 takes up how pools with higher audit scores can still finish behind on Macro-F1.

4.4 Extended replay

Our main replay fixes one seed, backbone, judge, and five selectors. We relaxed each in turn on the seed-13 pools, so audit columns cannot move and only the downstream fit or judge changes.

Selector	LC↑	XLM-R F1↑ (3 seeds)	AfroXLMR↑ (seed 13)
CoSDA-V2	**0.904**	0.157 ± 0.018	0.344
AlpaGasus	0.890	$\mathbf{0.193 \pm 0.024}$	0.321
DEITA	0.869	0.174 ± 0.011	0.325
U+D control	0.849	0.178 ± 0.010	**0.348**
AlpaGasus+HR	0.845	0.164 ± 0.008	0.339
CoSDA-EQ	0.836	0.167 ± 0.010	0.341
Naive	0.767	0.171 ± 0.038	0.266

Table 2: Extended replay on the seed-13 pools, sorted by audit LC, with $\pm$ the seed sd. The audit-best selector is worst over three XLM-R seeds and second on AfroXLMR, which is one seed, where it also ranks second on XLM-R.

Three seeds. Re-fitting all seven selectors at seeds 13, 21, and 42 (Table 2) keeps the inversion in the mean, with CoSDA-V2 best on audit LC and worst on Macro-F1 at 0.157 and AlpaGasus best at 0.193, reproducing the direction of Table 1. The spread across seeds is more informative than the means. Our selector-level Spearman reads $+0.46$ at seed 13, $+0.43$ at seed 21, and -0.29 at seed 42, averaging $+0.20$. Had we run one seed and reported that correlation, we could have concluded that audit quality predicts utility well, not at all, or inversely, purely on the draw. Our within-cell statistic varies about half as much across seeds and averages $\rho = -0.002$ over the same seeds, at $+0.006$, $+0.233$, and -0.246 individually.

Per seed, CoSDA-V2 ranks second of seven at seed 13, sixth at seed 21, and last at seed 42, while its audit rank is fixed by construction because the same pools are reused.

A validation-aware selector. A selector that optimises for downstream fit directly is the most relevant comparison for our claim, so we implemented LESS (Xia et al., 2024) on these pools: LoRA warmup with AdamW over four checkpoints, per-example gradients, Adam preconditioning in full parameter space, plain validation gradients, and the checkpoint-weighted cosine of its Definition 3.1, computed exactly rather than through the paper's random projection. Table 3 reports it beside its competitors under one identical training loop.

As specified, LESS finishes last, below naive, and the cause is its selection rule rather than its scores. LESS ranks candidates and then takes a plain global top-k with no balance constraint, which at a 64-example budget puts 77% of the retained pool in one label on average and 98% in all four sentiment cells, while Macro-F1 averages over classes. The ranking itself is informative: keeping the bottom 64 rather than the top 64 drops Swahili news from 0.181 to 0.012. Given the label quotas the other selectors receive, LESS reaches 0.278, near AlpaGasus at 0.288. So optimising for downstream fit does not lead here either, and at this budget the retention constraint matters more than the scoring rule, the same lesson §5.1 draws from the gate.

Selector	F1↑	Retained pool
U+D control	0.297	label quotas
AlpaGasus	0.288	label quotas
LESS + quotas	0.278	label quotas
Naive	0.205	unconstrained
LESS (as specified)	0.138	unconstrained top-k

Table 3: Real LESS against the selectors it competes with, eight cells at seed 13 under one identical training loop. Absolute values are not comparable to Table 1, which used a different pipeline.

A second backbone. Re-fitting the same pools with AfroXLMR base (Alabi et al., 2022), an in-domain African-language encoder, changes the picture (Appendix Figure 4). Against the same seed-13 XLM-R fits every selector gains, least of all naive at $+0.118$, the within-cell Spearman rises from $+0.006$ to $+0.21$, and CoSDA-V2 is second at 0.344 against 0.321 for AlpaGasus, while naive stays worst at 0.266. AfroXLMR is one seed. The gap is widest for the general-purpose backbone and narrows for the in-domain one, which is also the higher-scoring one here, so on this one-seed contrast the two rankings align better under the higher-scoring backbone. We report that as an interaction, not as an effect of language representation.

Three judges. We re-scored label correctness on the same pools with Qwen2.5-14B-Instruct, our generator, which varies size, and Phi-3.5-mini-instruct (Microsoft, 2024), varying family, on 1,523 candidate records each, of which 1,521 and 1,502 returned a usable score.

The additional judges (Table 4) distinguish the coarse quality-aware-vs-naive separation from the fine selector ordering. Qwen-14B preserves the coarse separation, keeping every quality-aware selector above naive and tracking our original judge at Pearson $+0.73$. Phi-3.5 does not: it agrees only weakly at $+0.29$ and places three of the six below naive. So the coarse separation survives a same-family judge but not a much smaller one, which is judge sensitivity rather than a measurement of judge noise.

Selector	Qwen-32B	Qwen-14B	Phi-3.5
CoSDA-V2	**0.904**	0.556	0.417
AlpaGasus	0.890	0.538	0.425
DEITA	0.869	0.495	0.411
U+D control	0.849	**0.557**	**0.426**
AlpaGasus+HR	0.845	0.518	0.403
CoSDA-EQ	0.836	0.538	0.406
Naive	0.767	0.458	0.414
Pearson vs. Qwen-32B		+0.73	+0.29

Table 4: Label correctness under three judges on the same pools. Qwen-14B preserves the quality-aware-vs-naive separation. Phi-3.5 does not. The ordering among quality-aware selectors is judge-sensitive throughout.

Cell	HR pool	AG↑	AG+HR↑	Δ
news/amh	**31**	0.387	0.135	**−0.252**
news/hau	**50**	0.114	0.140	+0.026
news/swa	81	0.145	0.078	−0.067
news/yor	87	0.067	0.133	+0.066
sent./amh	93	0.220	0.153	−0.067
sent./hau	153	0.268	0.331	+0.063
sent./swa	131	0.155	0.155	+0.000
sent./yor	138	0.184	0.184	+0.000
mean		0.193	0.164	−0.029

Table 5: Applying the audit's strict gate to AlpaGasus's ranking, averaged over seeds 13/21/42. HR pool counts candidates surviving the gate against a budget of 64. The mean is dominated by news/amh.

Does the judge track human labels? We checked the judge against human-annotated benchmark labels. We drew 320 human-labelled items, 160 per task, and scored each twice, once correctly labelled and once flipped, giving 640 presentations. It accepts a correct label 50.3% of the time and a flipped one 14.1%, a 36.3-point discrimination at 68.1% agreement. The margin is far wider on news topic, 56.9% against 6.2%, than on sentiment, 43.8% against 21.9%. Our judge carries real signal while being conservative and uneven across tasks, consistent with LC ordering selectors coarsely rather than finely. This indicates nonzero label discrimination under one prompt. It does not validate the synthetic text, and we do not release the raw judgments or the acceptance rule.

5 Analysis: Why the Audit Misses

A selector can lead the audit and still finish mid-pack downstream, because the audit and the classifier score different things about the same pool. Per-example audit scores and set-level training utility measure different properties, and they come apart in three places in the trace data.

H1: strict selectors prune away support the classifier needs. CoSDA-V2 keeps every candidate, so its residual violation rate describes the risk it tolerates, not anything it removed. That rate still tracks the outcome: it averages 0.23 in the five cells where V2 loses to naive against 0.08 in the two where it wins, and correlates −0.56 with the V2-minus-naive delta (p=0.15, two-sided permutation). With eight cells this motivates rather than establishes the reading that an audit-strict pool can lack support the classifier needs, so §5.1 tests a strict gate directly.

H2: AlpaGasus does not win on variety. Quality ranking might preserve more variety, though D_i measures distance from gold rather than within-pool coverage. AlpaGasus scores D=0.590, the lowest of the five selectors, below naive at 0.599 and below both CoSDA variants at 0.602 and 0.604, while DEITA, which optimises for diversity, scores highest at 0.649 and still finishes last downstream (Appendix Table 9). One hypothesis among several for AlpaGasus is decision-boundary support carried by examples that score well on judged quality but fail the audit's counterfactual gate, precisely where strictness and training utility pull apart.

H3: LC saturates and loses resolution. Every selector scores at least 0.76 on LC and the field bunches near 0.90, while Q compresses into $[0.66, 0.75]$. Both channels rank selectors confidently, with an LC-to-Q rank correlation of +0.90, yet at that compression they cannot resolve differences at the scale Macro-F1 responds to. Our multi-judge check (§4.4) is consistent with this, since the fine ordering reshuffles across judges while the coarse separation is steadier. Cross-judge disagreement measures sensitivity rather than saturation, so we offer it as support rather than measurement.

Putting H1 to H3 together. The traces motivate three non-exclusive explanations: severe underfill may remove support the classifier needs, AlpaGasus may retain boundary-relevant examples our diversity measure misses, and compressed judge scores may have limited resolution. Our experiments directly test only the total effect of adding the gate, with pool size and composition confounded where it underfills. The audit describes what a selector kept, not what a model learns from it.

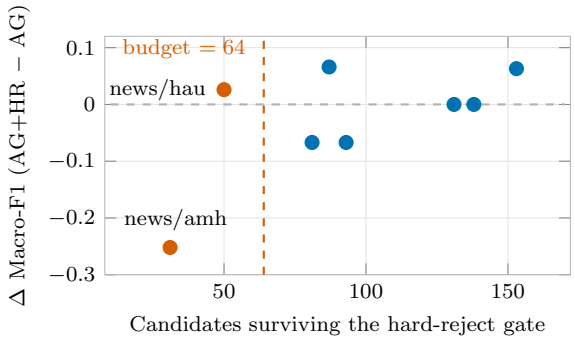

Figure 3: Effect of adding the audit's strict gate to AlpaGasus, against how many candidates survive it. The mean loss is dominated by the one cell where the surviving pool falls far below the 64-example budget.

5.1 Ablation

H1 claims that audit-strict filtering removes support the classifier needs. CoSDA-V2 itself never hard-rejects (§2), so we cannot test this by removing a step from V2. We test the gate directly instead. AlpaGasus+HR applies the audit's strict gate and ranks the survivors by AlpaGasus's judged-quality score, holding the ranking rule fixed at the downstream leader's. It measures what hard gating costs a strong selector, which is precisely the choice CoSDA-V2 declines to make.

Table 5 and Figure 3 localise the cost. Adding the gate costs 0.029 mean Macro-F1 over three seeds, from 0.193 to 0.164, and one cell dominates that average. In Amharic news the gate admits only 31 of the 64 candidates the budget calls for, and Macro-F1 falls by 0.252. Across the other seven cells the net effect is +0.003, and over the six full-pool cells it averages −0.001, with individual cells at −0.067 and +0.066 that cancel.

Two details are consistent with severe underfill, though not distinguishable from the surviving pool's composition. Amharic has the lowest C_i pass rate of our four languages, 33.4% against 52.9–58.9% elsewhere, so the gate strips its pool hardest, exactly where the loss appears. And milder starvation does not reproduce it, since Hausa news retains 50 of 64 and still gains 0.026. Among unstarved cells the deltas move in both directions and cancel, with news/yor at +0.066 and sent./hau at +0.063 against news/swa and sent./amh at −0.067, while sent./swa, whose pool overlaps AlpaGasus's at 63/64, does not move in any seed.

Two readings follow, both narrower than a causal decomposition. The contrast adds the gate while holding the ranking fixed, so it is not evidence that the counterfactual and shortcut channels order candidates badly. And the loss concentrates under severe underfill: over the six full-pool cells the gate averages −0.001, with cells at −0.067 and +0.066 that cancel rather than vanish. Since gate composition and training-set size move together where it underfills, this is consistent with harm under severe underfill rather than an identified threshold. Comparing AlpaGasus truncated to the same n would separate them.

5.2 Cell-level case studies

Our eight cells group three ways (Appendix Table 10). Two are positive, Swahili news and Hausa sentiment, where V2's residual rate is low at 0.16 and 0.00. Three are negative, and the largest loss carries the largest residual rate at 0.609 and is the cell our stress test finds most starved, though the link is not monotone: Hausa news is the smallest loss and carries the second-largest rate, 0.406. Three are near-degenerate. The groups track V2's residual rate rather than the naive Macro-F1 scale, so benchmarks here should separate cells exercising the audit from those exercising the model.

5.3 What the audit pairs with

Our replay supports three recommendations. Report audit and downstream metrics on the same selected sets, which any audit pipeline makes nearly free. Treat judged label correctness as an audit channel rather than a downstream proxy, since our selector-level Spearman is ρ=0.10 here and swings from −0.29 to +0.46 across seeds (§4.4). And make hard rejection budget-aware, since §5.1 finds the gate's cost concentrated where the surviving pool falls far short of the budget.

When we would still prefer CoSDA-V2. CoSDA-V2 may still be preferred when its channels are objectives in themselves, such as retained leakage of L=0.003 against AlpaGasus's 0.006, the lowest retained shortcut score at H=0.057, or the only label correctness above 0.90. These describe the pool it returns, and V2 applies graded penalties rather than hard constraints, so it can still retain a violator.

Filtering strictness vs. diversity. Our two CoSDA variants differ in both scoring and gate treatment: EQ ranks by S_i with gate priority and backfill, V2 by S_i^{V2} with no gate. They reach similar diversity and sit inside each other's bootstrap uncertainty, so neither difference separates them downstream at this scale.

6 Related Work

Synthetic supervision and selection. Language models can generate supervision from seed examples or task descriptions (Wang et al., 2023; Honovich et al., 2023; Wang et al., 2022; Mishra et al., 2022; Bach et al., 2022; Longpre et al., 2023; Taori et al., 2023; Xu et al., 2023), and multilingual variants carry the recipe across languages (Üstün et al., 2024; Aryabumi et al., 2024). A second line asks which generated examples to keep. AlpaGasus ranks by ChatGPT quality (Chen et al., 2024), DEITA combines quality with complexity and diversity (Liu et al., 2024), LESS selects by validation influence (Xia et al., 2024), QuRating learns a quality model (Wettig et al., 2024), and related work studies self-guided scoring and small curated sets (Li et al., 2024; Cao et al., 2024; Zhou et al., 2023). Nearly all of them inherit one assumption, that selector quality predicts downstream utility, and we test it at matched budget on low-resource cells.

LLM judges as selection evidence. Judges scale cheaply across candidate pools, but what a high score certifies is narrower than it looks. High label correctness means the judge can map a candidate to its intended label, not that a small classifier will learn a better boundary from it. That distinction sharpens when a retained pool holds 64 examples, so we treat judge scores as audit channels rather than as evidence of training utility.

Utility-vs-quality misalignment. Adjacent work has felt the same tension. DEITA adds complexity and diversity because quality alone proved insufficient (Liu et al., 2024), LESS sidesteps the issue by selecting against validation influence (Xia et al., 2024), and data-centric studies show that example utility is model-dependent rather than a property of surface cleanliness (Swayamdipta et al., 2020; Sorscher et al., 2022). We make the misalignment explicit at the *selector* level, using within-cell rank statistics.

Data-centric diagnostics and counterfactual checks. Data-centric NLP uses cartography, forgetting, proxy selection, and pruning to find useful or harmful training points (Swayamdipta et al., 2020; Toneva et al., 2019; Coleman et al., 2020; Paul et al., 2021; Mindermann et al., 2022; Sorscher et al., 2022), counterfactual tests ask whether models respond to task-relevant changes rather than artifacts (Kaushik et al., 2020; Gardner et al., 2020; Ribeiro et al., 2020; Wu et al., 2021), and work on shortcuts and memorisation explains why fluent examples can still be risky (Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019; Geirhos et al., 2020; Jia and Liang, 2017; Dodge et al., 2021; Carlini et al., 2021; Lee et al., 2022; Kandpal et al., 2022; Maini et al., 2024). We move these diagnostics to data selection, auditing each retained pool before asking whether the signals predict Macro-F1.

Low-resource multilingual benchmarks. Our replay uses MasakhaNEWS and AfriSenti (Adelani et al., 2023; Muhammad et al., 2023), whose four languages are written in two scripts, Ge'ez and Latin, so we exercise selector behaviour across script as well as topic. AfroXLMR (Alabi et al., 2022) supplies the in-domain backbone for our scope check in §4.4.

7 Conclusion

We tested a proxy that synthetic-data selection usually takes for granted, and in this replay it does not hold. CoSDA-V2 leads judged label correctness and trails downstream, AlpaGasus inverts that order, and our resampling checks preserve the mismatch, though a second backbone reverses it. The two rankings move differently. The downstream order varies across seeds, CoSDA-V2 standing second of seven at one seed and last at another, whereas the audit scores are fixed by construction because the same pools are reused.

Our stress test locates a cost rather than a cause. Adding the audit's strict gate to a fixed ranking lowers mean Macro-F1 by 0.029, almost all of it in the one cell where 31 of 64 requested candidates survive. Since gate composition and training-set size change together there, this is consistent with harm under severe underfill rather than an identified mechanism, and a count-matched control would separate them.

Two things follow. Evaluations should report audit and downstream metrics on the same retained sets, since ours used identical pools and still disagreed. We release the per-candidate records, pools, and scripts so others can test how far it travels.

Acknowledgments

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

Limitations

Scope. Our evidence is a controlled case study rather than a general result. It covers eight task-language cells in Amharic, Hausa, Swahili, and Yoruba, two classification tasks, one generator, one primary judge, and two backbones. The phenomenon we study is not specific to low-resource African languages, but our evidence is, and every limitation below should be read inside that frame.

Cell count and statistical strength. Our replay covers 8 task-language cells. We compute the selector-level Spearman over 5 selectors in the main replay and 7 in the extension, and §4.4 shows that statistic is unstable across seeds, ranging from -0.29 to $+0.46$. That is why we report it qualitatively alongside rank orders and rest our argument on the within-cell Spearman instead. Our bootstrap CIs on V2-minus-naive and AlpaGasus-minus-naive both contain zero, so we make no claim that any selector beats naive downstream. Absolute Macro-F1 is low throughout and three cells are near-degenerate. The inversion survives when we drop those, though the LC-to-F1 rank correlation rises there too (§4.3), so the restriction cuts both ways. This remains a regime where small absolute differences carry the signal.

Seeds and pipelines. Our main replay is a single-seed deterministic trace, and the multi-seed evidence comes from re-fitting the same retained pools at three seeds on different hardware. Per-selector Macro-F1 therefore moves by up to 0.048 against Table 1, enough to reorder the selectors at the shared seed, which is why we report the two pipelines separately rather than pooling them. Three seeds bound seed variance loosely rather than tightly. Each deterministic replay across 7×8 fine-tunes plus the judge audit costs roughly one GPU-week per seed, and that cost is what caps our seed count.

Task scope. We cover two classification tasks. Sequence labelling, intent and slot filling, and summarisation each need their own counterfactual validator (§2), and we chose to isolate the classification setting before extending the validator family. We make no claim about high-resource or generative settings.

Backbone dependence. The gap is backbone-conditioned. With XLM-RoBERTa base the audit-best selector is worst downstream, while with AfroXLMR base the within-cell Spearman rises to $+0.21$ and CoSDA-V2 recovers to second place (§4.4). We tested two backbones, and only one seed on the second, so we do not know whether the re-alignment continues with stronger encoders. Our safest reading is that audit-utility divergence is largest where the downstream model is weakest on the target language.

Generator dependence. All our candidates come from Qwen2.5-14B-Instruct under one nucleus-sampling configuration. Unlike the backbone and the judge, we did not vary the generator, so our finding is scoped to it. A stronger or differently tuned generator would change the composition of the pool every selector reads.

Judge dependence. Judged label correctness and quality come from a single multilingual judge prompted in English with in-language task descriptions. Our two additional judges (§4.4) show the coarse separation holds under a same-family judge but not under a much smaller one, and the fine LC ordering is unstable throughout, so the specific LC values, and the identity of the LC leader, should be read as properties of that judge and prompt. The gap finding does not rest on this channel alone, because the channels that use no judge field (D, L, H) reproduce the disagreement, and the within-cell Spearman is computed inside each cell on a shared pool. Counterfactual consistency is judge-assisted, since C_i scales the judge's validity rating. We did not run a multi-prompt sweep.

No human validation of the synthetic text. This is the most consequential limitation for an African-language study. We validated the *judge* against human-annotated benchmark labels (§4.4), where it separates correct from flipped labels by 36.3 points but agrees with human labels only 68.1% of the time and is markedly weaker on sentiment. We did not run a native-speaker review of the generated text itself, so nothing in this paper is evidence about the cultural, dialectal, or orthographic adequacy of our synthetic examples. Automatic channels can miss exactly those failures, and a selector that looks clean on our channels could still retain text a speaker would reject.

Leakage and shortcut detectors. Our exact n-gram, shingle-Jaccard, and LaBSE-based leakage checks target observable contamination and cannot certify the absence of memorisation in the generator's training data. The shortcut detector uses a

fixed feature inventory, and no candidate reaches τ_H=0.70 (maximum H is 0.489), so the shortcut gate never fires, though H still contributes to ranking. All selectors read the same audit fields, but because they weight those fields differently, their rankings may respond differently to detector error.

Score-weight calibration. We chose the weights in S_i on a held-out Setswana development split to equalise per-channel contributions, and we ship them in `configs/default.yaml` so others can vary them without regenerating candidates. We did not release the calibration split or its script, so τ_L and the α weights should be read as fixed protocol choices rather than a derivation a reader can reproduce. A $\pm 10\%$ sensitivity sweep on τ and α would quantify how far the V2 pool moves under nearby protocol choices. We did not run it.

Baseline coverage. AlpaGasus and DEITA are budget-matched reimplementations of each method's *selection criterion* rather than runs of the original codebases, all reading the shared audit record so the comparison isolates the selection rule. Our numbers therefore characterise the criteria and not the published systems. The U+D control computes no gradient influence and is named for what it computes. We additionally implement LESS itself (§4.4), on one seed and one training loop that is not the main replay's, so its absolute values are not comparable to Table 1.

Scope of the ablation. §5.1 measures what the audit's strict gate costs a strong selector, on eight cells at three seeds. It is not a clean causal decomposition of CoSDA-V2, which applies no gate at all, and in the two underfilled cells it varies training-set size alongside the selection rule, so filtering and data volume are confounded exactly where the effect is largest. It does not establish which channel inside the gate does the pruning, and it does not test the budget-aware repair it suggests. H2 and H3 remain trace-grounded explanations rather than ablated mechanisms, though H3 now has a multi-judge sensitivity check.

Ethical Considerations

Synthetic data for low-resource communities can ease annotation pressure, but it can also distort cultural context, normalise incorrect language use, or surface private information from web-trained generators (Bender et al., 2021). We designed CoSDA to address these risks as an audit instrument, logging provenance, checking leakage at three levels, and measuring counterfactual consistency before an example enters a retained pool.

Our replay uses MasakhaNEWS, whose card states CC BY-NC 4.0, and AfriSenti, whose authors state CC BY 4.0. We do not redistribute the source text either dataset reproduces, since MasakhaNEWS carries BBC and VOA article bodies and AfriSenti carries tweet text, and neither licence grants that. The release ships record identifiers and digests instead. Every released record carries its source dataset identifier, its seed provenance, and a `generator` field, so generated text is marked as generated. Anyone who identifies content they did not consent to release can request removal through the artifact repository's issue tracker. Practice in this area recommends accompanying data statements and datasheets (Bender and Friedman, 2018; Gebru et al., 2021), which we regard as the right next step for this release.

We report automatic audit metrics rather than a human annotation study, so we claim no human-validated cultural plausibility. We did validate our judge against human-annotated benchmark labels (§4.4) and found it separates correct from flipped labels while agreeing with human labels only 68.1% of the time, and less reliably on sentiment than on news topic. No native speaker reviewed the generated text itself. Readers should therefore treat every audit-quality number here as a machine measurement, and anyone deploying selected examples should add community review by speakers of the target language, especially for political news and sentiment data.

References

David Ifeoluwa Adelani, Marek Masiak, Israel Abebe Azime, Jesujoba Alabi, Atnafu Lambebo Tonja, Christine Mwase, Odunayo Ogundepo, Bonaventure F. P. Dossou, Akintunde Oladipo, Doreen Nixdorf, Chris Chinenye Emezue, Sana Al-azzawi, Blessing Sibanda, Davis David, Lolwethu Ndolela, Jonathan Mukiibi, Tunde Ajayi, Tatiana Moteu, Brian Odhiambo, Abraham Owodunni, Nnaemeka Obiefuna, Muhidin Mohamed, Shamsuddeen Hassan Muhammad, Teshome Mulugeta Ababu, Saheed Abdullahi Salahudeen, Mesay Gemeda Yigezu, Tajuddeen Gwadabe, Idris Abdulmumin, Mahlet Taye, Oluwabusayo Awoyomi, Iyanuoluwa Shode, Tolulope Adelani, Habiba Abdulganiyu, Abdul-Hakeem Omotayo, Adetola Adeeko, Abeeb Afolabi, Anuoluwapo Aremu, Olanrewaju Samuel, Clemencia Siro, Wangari Kimotho, Onyekachi Ogbu, Chinedu

Mbonu, Chiamaka Chukwuneke, Samuel Fanijo, Jessica Ojo, Oyinkansola Awosan, Tadesse Kebede, Toadoum Sari Sakayo, Pamela Nyatsine, Freedmore Sidume, Oreen Yousuf, Mardiyyah Oduwole, Kanda Tshinu, Ussen Kimanuka, Thina Diko, Siyanda Nxakama, Sinodos Nigusse, Abdulmejid Johar, Shafie Mohamed, Fuad Mire Hassan, Moges Ahmed Mehamed, Evrard Ngabire, Jules Jules, Ivan Ssenkungu, and Pontus Stenetorp. 2023. MasakhaNEWS: News topic classification for African languages. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 144–159, Nusa Dua, Bali. Association for Computational Linguistics.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-Jian Jiang, and Alexander M. Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623. Association for Computing Machinery.

Andrei Z. Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of Sequences 1997*, pages 21–29. IEEE Computer Society.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2024. Instruction mining: Instruction data selection for tuning large language models. In *Proceedings of the First Conference on Language Modeling*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. AlpaGasus: Training a better Alpaca with fewer data. In *Proceedings of the 12th International Conference on Learning Representations*.

Cody Coleman, Chia-Chen Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. 2020. Selection via proxy: Efficient data selection for deep learning. In *Proceedings of the 8th International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala,

Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Robert Geirhos, Jørn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10697–10707. PMLR.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *Proceedings of the 8th International Conference on Learning Representations*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *Proceedings of the 12th International Conference on Learning Representations*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14044–14072, Bangkok, Thailand. Association for Computational Linguistics.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Microsoft. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Sören Mindermann, Jan M. Brauner, Muhammed T. Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15630–15649. PMLR.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions.

In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Seid Muhie Yimam, David Ifeoluwa Adelani, Ibrahim Said Ahmad, Nedjma Ousidhoum, Abinew Ali Ayele, Saif Mohammad, Meriem Beloucif, and Sebastian Ruder. 2023. SemEval-2023 task 12: Sentiment analysis for African languages (AfriSenti-SemEval). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2319–2337, Toronto, Canada. Association for Computational Linguistics.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, pages 20596–20607.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. 2022. Beyond neural scaling laws: Beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, volume 35, pages 19523–19536.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. Blog post.

Qwen Team. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Mariya Toneva, Alessandro Sordoni, Rémi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *Proceedings of the 7th International Conference on Learning Representations*.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. QuRating: Selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 52752–52774. PMLR.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceed-*

ings of Machine Learning Research, pages 54104–54131. PMLR.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2023. WizardLM: Empowering large pre-trained language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021.

A Reproducibility Checklist

How to read the appendix. The appendix is organised as a trace from evidence to claim. This section names the source artifacts and replay contract; Appendix B describes the prompts and audit fields that create the per-candidate records; Appendix C gives the full detail tables and appendix plot. The main paper reports the claim-carrying summaries; the appendix exposes the rows and artifacts behind them.

Ground-truth replay. The main replay is a single per-cell table indexed by task, language, and selector. Each row records the retained-set size, judged label correctness, judged quality, counterfactual consistency C, leakage score L, shortcut score H, hard-reject rate, and downstream Macro-F1 from the deterministic Hugging Face classification path. That table populates the main-replay tables and figures; the extended replay of §4.4 adds the separate result files listed in the artifact README, and the per-candidate statistics we report, namely the diversity column of Table 9 and the C_i pass rates of §2, are recomputed from the released audit records under `runs/select1_seed13/`. Aggregates (selector-level means, Spearman correlations, bootstrap CI, leave-one-cell-out, and non-degenerate-subset statistics) are deterministic functions of these files, not separate experiments.

Ground-truth artifact paths. The artifact's `README.md` lists the files treated as authoritative, and `results/main_replay/claim_ledger.csv` maps the audit channels, extended-replay means, AfroXLMR means and ablation values to their source file and computation.

Released artifacts. We release four artifact groups: (i) a dataset manifest covering the dataset id, split paths, split sizes, and label counts for MasakhaNEWS and AfriSenti; (ii) the generated candidate examples and per-candidate audit records (utility components, leakage components, shortcut features, diversity, counterfactual consistency, judged label correctness, judged quality, and hard-reject decision); (iii) the per-cell replay table described above for all selectors plus the gold-only reference; and (iv) the scripts that download data, generate candidates, score audit channels, apply each selector, fine-tune the downstream classifier, and emit the replay table. The per-candidate audit records and the seven per-selector retained pools for all eight cells ship under `runs/select1_seed13/`. The extended replay of §4.4 ships as a separate result group with its own run scripts and per-run JSON, kept apart from the main replay because it was fitted on different hardware. The release separates candidate-level audit records from selected-set summaries so that a reader can recompute a selector without trusting the aggregate tables.

Claim ledger. The released claim ledger (`results/main_replay/claim_ledger.csv`) maps the audit channels, the extended-replay means, the AfroXLMR means and the ablation values to the result file and computation that produces each, and `scripts/make_claim_ledger.py` regenerates it. A single command, `reproduce.sh`, recomputes the extended-replay aggregates and the main-replay statistics of §4.3 from the committed per-run results using only the Python standard library, and the released `ANALYSIS_OUTPUT.txt` and `judge_agreement_output.txt` are the reference outputs to diff against.

Default hyperparameters. For the audit channels, α_U=1.0, α_C=0.75, α_D=0.50, α_L=1.0, and α_H=0.75; hard-reject thresholds are τ_L=0.15, τ_C=0.60, and τ_H=0.70. These values were fixed before the replay and reused unchanged across all eight task-language cells. Downstream fine-tuning uses the deterministic Hugging Face classification path (`cosda/hf_training.py`): 3 epochs, batch size 16, learning rate 2×10^{-5}, maximum length 256, warmup ratio 0.1, evaluating on dev each epoch and restoring the best dev Macro-F1 checkpoint. The main-replay seed is 13 and the gold budget is b=64 with synthetic multiplier m=3.

Calibration of thresholds and weights. The thresholds and weights are protocol choices fixed before the replay (see the end of §2): τ_L=0.15 bounds the composite leakage risk L_i rather than a raw cosine, τ_H=0.70 bounds the shortcut score H_i, τ_C=0.60 gates $C_i \in [0, 1]$, and the score weights $(\alpha_U, \alpha_C, \alpha_D, \alpha_L, \alpha_H)$ equalise per-channel contributions in $[0, 1]$ on a held-out Setswana split we do not release. All four groups are identical across the eight reported cells and remain fixed throughout the replay.

Extended-replay protocol. The extension reuses the seed-13 retained pools verbatim, so audit channels are unchanged and only the downstream fit is recomputed: seeds 13/21/42 with XLM-RoBERTa base, and seed 13 with AfroXLMR base (Alabi et al., 2022) (`Davlan/afro-xlmr-base`), same deterministic training path and hyperparameters throughout. The two additional judges are Qwen2.5-14B-Instruct and Phi-3.5-mini-instruct (Microsoft, 2024), run with the same judging prompt as the main judge.

Deferred experiments. Native-speaker review of the generated text, generator replacement, multi-prompt judge sweeps, budget sweeps, per-channel decomposition of the hard-reject gate, the budget-aware hard-reject repair proposed in §5.1, and task extensions to NER, intent/slot filling, and summarisation are not claimed as completed results in this paper.

B Prompts and Audit Protocol

Purpose. The prompt and audit protocol is designed to make synthetic examples inspectable before selection. The generator creates candidate examples and counterfactual edits; the audit record then stores both judge-facing quality fields and non-judge diagnostic fields. The downstream classifier never sees the audit fields directly. It sees only the retained synthetic examples chosen by each selector.

Generation prompt template. The generation prompt fixes task, language, label schema, seed examples, and output format, and includes the forbidden-behaviour clause. The template as shipped in `cosda/generation.py` is reproduced below. The prompt hash stored in each candidate record is used for traceability, not as a modelling feature.

```
Task: {task}
Language: {language}
Allowed labels: {labels}
Seed label: {label}
Seed text excerpt:
{seed_excerpt}
Generate {n} new examples in the same language and label. Each generated text should be a concise news item of 60–120 words. [For sentiment cells this line instead reads: Each generated text should be concise and natural, usually under 60 words.] Do not copy the seed text or held-out text. Do not translate an English template. Return valid JSON only, with no Markdown fences or commentary. Return JSON list: [{"text": str, "label": str, "confidence": float}].
```

Counterfactual edit template. For each candidate x_i, the generator receives x_i and produces x_i' by changing one classification-relevant variable, such as the topic or sentiment label cue, while preserving language, register, and as much non-label content as possible. The expected flipped label is stored with the counterfactual pair and used by the counterfactual-consistency score. A pair is therefore evaluated as a data-quality check: the candidate should be fluent and labelled correctly, and the paired edit should move the label in the expected direction.

Automatic audit fields. The replay records judged label correctness, judged quality, counterfactual consistency (C), leakage score (L), shortcut score (H), hard-reject status, and selected-set membership for each retained-set baseline. These fields support Tables 1 and 9. They are automatic audit signals, not human annotation outcomes.

Selector membership fields. Each selector receives the same candidate pool within a task-language cell. Membership ships as one retained-pool file per selector next to the audit record, covering all seven selectors rather than as a field inside each candidate. This shared-pool design is why the appendix tables can compare audit quality and downstream Macro-F1 without confounding the result with different generation runs.

Hard-reject interpretation. The interpretation of the hard-reject column, an audit decision over each selector's retained pool rather than a ground-truth label, is described in §2. The downstream results separate this retained-pool property from Macro-F1.

Qualitative examples. Table 6 gives four candidates drawn from the per-candidate audit records, illustrating one *kept* candidate, one rejected for counterfactual consistency, one rejected for leakage, and the highest shortcut score in the replay, which the gate does not reject. The candidates are

Status	Trigger and audit profile
Kept	Swahili business candidate (`5328207e1efef02d`); LC= 1.0, C=0.72, L=0.00, H=0.02. Passes all hard-reject thresholds and is ranked in the top-64 by every quality-aware selector.
Reject (CF)	Amharic sentiment candidate (`eabcd5ba208b51da`) where the judge scores label correctness 0 and counterfactual validity v_{cf}=0.7; C_i=0.59 falls below τ_C=0.6 and the candidate is hard-rejected on the counterfactual channel.
Reject (leakage)	Yoruba entertainment candidate (`5554f5d6339bd4c8`) with a 10-gram match against the MasakhaNEWS dev/test union (E_i=L_i=1.0); the leakage gate fires before the candidate reaches the reranker. It is the only leakage rejection in the eight cells.
No shortcut reject	The largest shortcut score in the replay is a Yoruba sentiment candidate (`96be681d80e672a9`) at H=0.49, from a label-word hit plus atypical length. It stays below τ_H=0.70, so the shortcut gate fires on no candidate in the eight cells.

Table 6: Four candidates: kept, counterfactual-rejected, leakage-rejected, and the highest shortcut score in the replay. Values are taken from the released audit records.

shown with a brief description of the trigger, not the raw text, since text in four African languages is hard to read inline. The release ships the full audit record for every candidate, including the raw text, the generator's counterfactual edit, the per-channel scores, and the hard-reject reason.

Future human validation. A future speaker validation study should ask L1 or strong L2 speakers to judge label correctness, fluency, cultural plausibility, leakage suspicion, and counterfactual validity. This paper does not report that study as completed evidence.

C Detail Tables and Plots

This appendix contains the detail evidence behind the compact main-body results. Numbers and plots in this section come from the per-cell replay table described in Appendix A, except the diversity column and the AfroXLMR results, which are recomputed from the released audit records and the extended-replay files. The goal is not to introduce additional claims, but to make the main claims auditable: downstream averages, audit averages, and per-cell effects can all be traced back to the same 8-cell replay.

How to read these tables. The appendix separates three views of the same replay. Table 7 defines the task-language cells. Table 8 reports downstream Macro-F1 for all selectors, including the Gold-only reference that is excluded from the main audit-vs-downstream rank comparisons. Table 9 reports the full audit-channel view, including leakage L, which is omitted from the compact main table. Table 12 holds the per-cell second-backbone results behind §4.4, and Table 13 and Figure 5 hold the leave-one-cell-out and per-cell-delta views behind §4.3.

Dataset	Task	Languages
MasakhaNEWS (Adelani et al., 2023)	News topic classification	amh, hau, swa, yor
AfriSenti (Muhammad et al., 2023)	Sentiment classification	amh, hau, swa, yor

Table 7: Our completed 8-cell evaluation, with Macro-F1 as the metric in every cell. Broader NER, intent, and summarisation settings need validators beyond our classification protocol.

Baseline	n	F1↑	Std	Wins	Δ
Gold only	8	0.116	0.074	2/8	−0.051
Naive	8	0.167	0.068	n/a	n/a
AlpaGasus	8	**0.202**	0.099	4/8	+0.036
DEITA	8	0.128	0.071	4/8	−0.039
CoSDA-EQ	8	0.133	0.109	1/8	−0.034
CoSDA-V2	8	0.163	0.103	2/8	−0.004

Table 8: Full 8-cell replay including the Gold-only reference, with wins and Δ measured against naive. CoSDA-V2 sits near naive on the mean, and §4.3 places that difference inside cell-level uncertainty.

Benchmark composition

The completed replay is deliberately narrow: two classification datasets and four languages. This scope keeps counterfactual validation task-specific and avoids mixing completed classification evidence with planned extensions.

Full downstream table with gold-only

Table 8 gives the downstream view used to compute the aggregate Macro-F1 claims. The Gold-only row is a reference point for low-resource fine-tuning without synthetic examples. It is not included in the audit-vs-downstream Spearman calculations because it has no retained synthetic pool to audit.

Full per-selector audit table

Table 9 gives the full audit view over selected synthetic examples. Our main text focuses on LC, Q, hard-reject, C, and H because those channels tie most directly to the gap between audit quality and utility, and the full table adds leakage L, where

Sel.	LC↑	Q↑	C ↑	D ↑	L ↓	H ↓	Rej.↓
Naive	0.767	0.662	0.587	0.599	0.003	0.066	0.486
AlpaGasus	0.890	**0.749**	0.653	0.590	0.006	0.065	0.246
DEITA	0.869	0.737	0.637	**0.649**	0.005	0.070	0.299
CoSDA-EQ	0.836	0.718	**0.686**	0.604	**0.002**	0.058	**0.109**
CoSDA-V2	**0.904**	0.746	0.674	0.602	0.003	**0.057**	0.162

Table 9: Full judge and audit quality of selected synthetic data, including the leakage column L omitted from the main body. Every quality-aware selector improves on naive across LC, Q, C, and hard-reject.

Cell	Naive	AG	DEITA	EQ	V2	Δ
news/amh	0.286	0.342	0.105	0.105	0.105	−0.181
news/hau	0.102	0.195	0.039	0.039	0.066	−0.036
news/swa	0.069	0.184	0.109	0.012	0.198	+0.129
news/yor	0.128	0.065	0.065	0.065	0.065	−0.062
sent./amh	0.154	0.067	0.067	0.067	0.069	−0.085
sent./hau	0.168	0.339	0.196	0.349	0.375	+0.206
sent./swa	0.248	0.248	0.253	0.248	0.248	+0.000
sent./yor	0.178	0.176	0.185	0.176	0.176	−0.002
mean	0.167	**0.202**	0.128	0.133	0.163	−0.004

Table 10: Per-cell Macro-F1 in the main replay (seed 13, XLM-R). AG is AlpaGasus; EQ and V2 are the CoSDA variants; Δ is V2 minus naive. The Gold-only reference is in Appendix Table 8.

all means are small but selector differences stay visible.

Two columns need a word of explanation. The diversity column D is the mean per-candidate D_i over each selector's 64-example retained pool, which we compute by joining per-candidate audit records with that selector's selected JSONL. That column weighs against variety preservation as the source of AlpaGasus's downstream lead in §5, since AlpaGasus scores D=0.590, lower than every CoSDA variant, while DEITA, which explicitly weights diversity, records the highest D at 0.649 and still finishes last downstream.

Per-cell Macro-F1 (main replay)

Table 10 gives the per-cell downstream view behind §4.3, for all five main-replay selectors with each selector's delta against naive.

Extended-replay detail

This subsection holds the row-level evidence behind §4.4. It is computed on the same seed-13 retained pools as the main replay, so the audit columns are unchanged and only the downstream fit is recomputed.

Leave-one-cell-out and per-cell deltas

Table 13 and Figure 5 support §4.3. Our eight-cell V2-minus-naive mean of −0.004 is built from cell effects of opposite sign, and removing Hausa

Attribute	AlpaGasus	CoSDA-V2	Δ
Macro-F1↑	**0.202**	0.163	+0.039
LC↑	0.890	**0.904**	−0.014
Q↑	**0.749**	0.746	+0.003
Hard reject↓	0.246	**0.162**	+0.084
C↑	0.653	**0.674**	−0.021
Per-cell wins↑	3	3	(2 ties)

Table 11: The inversion in one table. CoSDA-V2 wins three of the four audit rows and has the lower downstream Macro-F1. Bold marks the better value.

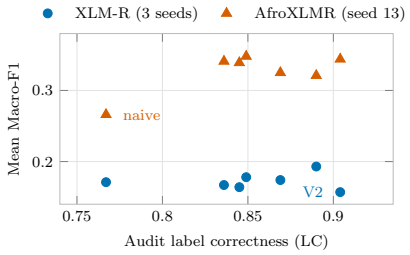

Figure 4: Audit LC against mean Macro-F1 for all seven selectors. XLM-R shows no upward trend, while AfroXLMR lifts every selector and partially restores it.

sentiment, the single large positive cell, moves it to −0.034. We include this view because an eight-cell mean can be dominated by one cell without that being visible in the aggregate.

Within-cell Spearman ranks

Figure 2 in the main body is the canonical presentation of our primary statistic, with four cells positive, four negative, a mean of +0.04, and a median of 0.00. Its values come from the main-replay CSV named in the artifact README, and §4.4 recomputes the statistic at two further seeds and on a second backbone.

Reading the appendix

This appendix is a consistency check rather than a source of new claims. Table 8 confirms AlpaGasus

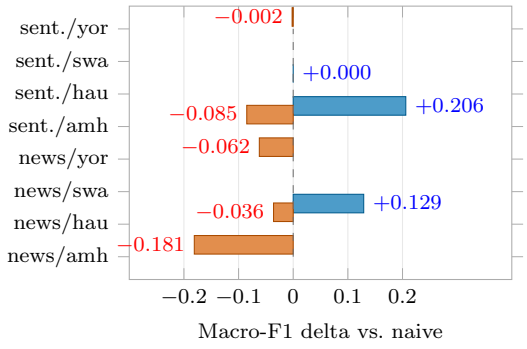

Figure 5: Per-cell Macro-F1 delta of CoSDA-V2 against naive. Each bar is one task-language cell, and positive values mean CoSDA-V2 wins.

Cell	Naive	AG	DEITA	U+D	AG+HR	EQ	V2
news/amh	0.591	0.285	0.607	0.376	0.443	0.499	0.612
news/hau	0.205	0.180	0.244	0.314	0.262	0.273	0.291
news/swa	0.311	0.218	0.247	0.303	0.232	0.374	0.345
news/yor	0.238	0.455	0.369	0.437	0.380	0.311	0.304
sent./amh	0.162	0.364	0.191	0.443	0.329	0.292	0.191
sent./hau	0.257	0.361	0.459	0.378	0.347	0.411	0.350
sent./swa	0.242	0.362	0.152	0.268	0.374	0.249	0.318
sent./yor	0.120	0.347	0.328	0.263	0.344	0.318	0.338
mean	0.266	0.321	0.325	**0.348**	0.339	0.341	0.344

Table 12: Per-cell Macro-F1 with the AfroXLMR backbone (Alabi et al., 2022) at seed 13, on the same retained pools. Every selector gains over the seed-13 XLM-R fit, from $+0.118$ for naive to $+0.182$ for AlpaGasus+HR.

Cell removed	Mean Δ (V2$-$Naive)$\uparrow$
(all 8 cells)	-0.004
news/amh	$+0.022$
news/hau	$+0.001$
news/swa	-0.023
news/yor	$+0.005$
sent./amh	$+0.008$
sentiment/hau	$\mathbf{-0.034}$
sent./swa	-0.004
sent./yor	-0.004

Table 13: Leave-one-cell-out mean Δ of CoSDA-V2 vs. naive. The first row is the all-cell baseline, and subsequent rows show the mean when the named cell is excluded.

carries the largest downstream mean at 0.202, Table 9 confirms CoSDA-V2 holds the strongest LC at 0.904, and the per-cell view in the main body shows the lower means for DEITA and CoSDA-EQ rest on cell effects of mixed sign. Table 12 shows how much is specific to the backbone, since every selector gains under AfroXLMR and the ordering shifts.

Taken together these tables support the scoped claim we make in the main body. The CoSDA selectors improve the audit quality of the pool they retain, and that improvement is measurable on channels a judge never sees, namely diversity, leakage, and shortcut. Downstream utility nonetheless remains cell-dependent and backbone-dependent, so we present the audit as an instrument for describing what was kept rather than as a predictor of what a model will learn from it.