Bootstrapping Platform Moderation Models with Reviewed Production Labels

Fabian S. Klinke Aleksandra Klushina

2026-03-31

Abstract

Content moderation on a small platform faces a structural challenge: harmful posts are rare enough that a community accumulates far more ordinary content than confirmed abuse cases before it has the data needed to train a reliable model. We study this in B90HQ, a music-community platform with five product-defined moderation categories, and describe a production pipeline that exports reviewer-confirmed labels as *gold* training data while using public moderation datasets only as a capped supplement where platform data is still lacking.

The resulting training set is 84.6% internal by row count, yet native confirmed-positive examples are nearly absent for most categories — the core tension this paper addresses. We compare a frozen `multilingual-e5-large` embedding model, which encodes text by meaning, against a TF-IDF word-pattern baseline. On the three categories with enough held-out support to draw reliable conclusions (`spam`, `hate_speech`, `violence`), the embedding model improves macro F1 from 0.80 to 0.89 ($p = 0.022$). On a separate

functional test suite designed to expose fragile shortcuts, F1 rises from 0.67 to 0.82. A single-source control on one public German hate-speech corpus confirms that this advantage holds even when source-identity shortcuts are removed. An embedding-space audit shows that public training examples are semantically close to platform content but form distinct source-shaped clusters, meaning current results are meaningful but not yet fully independent of the specific public datasets used.

The current model is suited for moderator assistance and early deployment. Its clearest path to improvement is accumulating more native confirmed-positive examples as the platform grows.

Introduction

Small-platform moderation has a familiar data problem. Harmful events matter operationally, but they are rare enough that a young platform can accumulate many reviewed benign examples before it accumulates enough native positive examples for robust supervised learning. At the same time, abusive-language research shows that public moderation datasets are strongly source-conditioned and often generalize poorly across domains (Antypas & Camacho-Collados, 2023; Karan & Šnajder, 2018; Wiegand et al., 2019). That creates a practical tension: a production team needs an initial moderation model early, but it cannot simply replace platform-native supervision with large public corpora and assume that the resulting classifier will transfer cleanly.

This tension is especially visible in B90HQ, a multilingual music-community app. Normal artist and fan participation is not itself a moderation problem yet; the difficult cases are repeated attempts to use the community as a distribution channel for outside products, audience growth, or off-platform lead capture. The current moderation release

operates over posts and comments and exposes five reportable top-level categories in the production moderation taxonomy: `self_promotion`, `spam`, `hate_speech`, `violence`, and `privacy`. These categories are intentionally product-facing rather than research-generic: `self_promotion` covers community-reach abuse for outside products or audience funnels, `spam` covers repetitive or deceptive low-quality content, `hate_speech` covers identity-targeted attacks on protected groups, `violence` covers threats, incitement, glorification of violence, or encouragement of self-harm, and `privacy` covers doxxing, non-consensual private media, or exposure of sensitive personal information.

The same taxonomy structures the operational moderation workflow. Reports are grouped by content item, moderation category, and content revision so that repeated reports accumulate into one moderation case rather than creating many independent decisions. Role-weighted voting can temporarily auto-hide content before moderator review, with current weights of `user = 1`, `artist = 3`, and `admin = 3`. Each category also has its own minimum weighted-vote and unique-reporter thresholds before temporary hiding is triggered. Final decisions remain moderator-controlled. The model we train here will be integrated so it automatically runs on every addition or update of a post or comment and gets its own vote count. The primary goal is to reduce the number of user votes needed when our systems detect potential abuse, with the longer-term possibility that, once the model improves and becomes trustworthy enough, it could make unilateral decisions.

For modeling, this means the task is not generic toxicity detection; it is the narrower problem of learning the same moderation categories that the product already uses for reporting, case aggregation, and review.

This paper describes the production-grounded dataset preparation, the two initial baseline models, and the evaluation design supporting the first deployment of the moderation model. We document the labeling contract, the public dataset supplementation policy, and an embedding-space audit that characterizes the main remaining limitation: transfer from public training sources to native platform content.

Research Questions

The study asks three practical questions.

1. Can reviewed moderation decisions provide a useful starting dataset before the platform has built up a large native gold set?
2. Does the current embedding-based model outperform the lexical baseline clearly enough to be the default local model?
3. When public data is still needed, do the gains reflect what matters on the platform rather than patterns of the public datasets?

Taken together, these pieces answer a narrow question: how far a small platform can get with a production-grounded moderation dataset before it has accumulated a substantial native abuse corpus, and where the main uncertainty remains.

Methods

Task Formulation

The current training setup uses five independent moderation heads: `self_promotion`, `spam`, `hate_speech`, `violence`, and `privacy`. Each head uses the three-valued label space `{positive, negative, unknown}`. We use this design because production moderation rows are often only partially labeled: a row can be explicitly negative for one head while

still being unknown for another. A single multiclass setup would either collapse these partial labels or require lossy relabeling.

The model-facing row schema keeps only the fields needed for learning and evaluation: `input_content`, `input_language`, the five head-specific targets, and a small traceability set (`example_id`, `source_dataset`, `source_type`, `supervision_level`, `sample_weight`). All identifiers are stored as hashed buckets rather than raw production identifiers. This means the training data retains enough structure for author-aware splitting and deduplication, but cannot be trivially linked back to production tables or used to re-identify individual platform users.

Upstream Labeling Pipeline in B90HQ

The labeling pipeline starts upstream — in the production platform B90HQ, before data reaches this repository. Its main mutable training store keeps one current-state row per `(target_type, target_id)` pair. Each row contains sanitized text, language fields, feed and author-role metadata, moderation summary fields, system-derived suggestions, and the final reviewer-facing labels. Auditability is preserved in a separate append-only event log that records manual updates, resets, and suggestion confirmations.

Figure 1 summarizes the two supervision paths that feed this store. First, users report posts or comments, moderators review the resulting cases, and approved or denied case decisions are ingested as system suggestions in the current training row. Second, moderators work directly in a separate labeling surface over those sanitized rows and either apply manual labels or confirm the current suggestion as the final export label.

The upstream system keeps *suggested* labels separate from *final* labels. Suggested labels reflect the latest moderation-derived state for the current content revision, while final la-

bels are the moderator-confirmed training labels. This separation prevents later synchronization passes from silently overwriting manual gold labels. The current label-source taxonomy includes `unlabeled`, `moderation_removed`, `moderation_approved`, and `manual`; review state is tracked separately through `todo` and `done`.

The upstream refresh routine rebuilds that snapshot. Its responsibilities include text sanitization, lightweight language detection, feed-audience and author-role snapshotting, suggested-label recomputation, and reconciliation of stored rows against the latest content state. Downstream training exports are filtered to rows with `review_status = done` and `final_label_source != unlabeled`. Exported data are sanitized-only and exclude direct usernames, raw mentions, and other direct PII.

Text Sanitization of Production Content

The internal training copy uses sanitized text, not raw production content. B90HQ rebuilds the moderation-training snapshot from current posts and comments and sanitizes the text before it reaches the training store; all downstream export and dashboard surfaces are limited to this sanitized representation.

The current sanitization contract is placeholder-based and intentionally simple. Direct user mentions are replaced with `@USER`, URLs with `[URL]`, email addresses with `[EMAIL]`, phone numbers with `[PHONE]`, and long identifier-like alphanumeric or numeric strings with `[TOKEN]`. Whitespace is normalized after replacement. The same placeholder vocabulary is also used when public datasets are normalized locally, which keeps token patterns more consistent across internal and public sources.

This design keeps signals such as "contains a link" available to the model while avoiding storage of direct personal identifiers or contact strings in the trainable dataset and

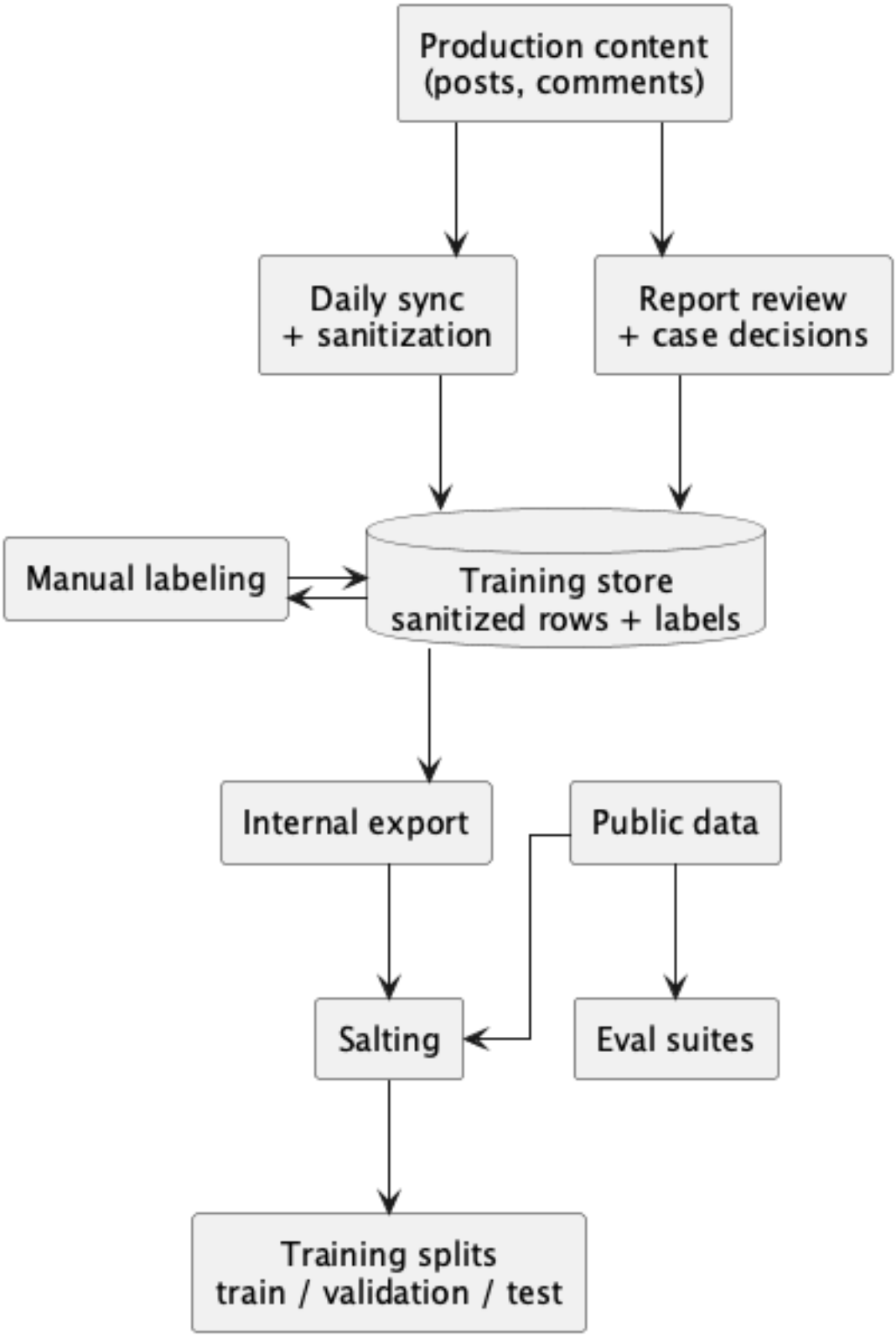

Figure 1: Production-to-training pipeline.

accidentally training on signals like “user xy is mentioned so it must be spam.”

Internal Supervision Contract

The downstream training workflow reads directly from the B90HQ training store and materializes two internal partitions when both are present: reviewed gold and an unresolved unlabeled pool. A row is exported as gold only if `review_status = done` and `final_label_source != unlabeled`. Gold rows retain final label states, the reviewer-confirmed label source, hashed example/author/thread buckets, and coarse metadata such as language, feed audience, and author role. Unresolved rows remain outside gold export only when `final_label_source = unlabeled`; these rows can still contribute weak training signal, but only if B90HQ’s own moderation system already made a classification guess — this repository does not second-guess production content itself.

This yields a three-level supervision hierarchy: internal gold as the anchor distribution, internal silver (machine-suggested but not yet human-confirmed labels) only from unresolved rows with stored upstream suggestions, and public data only as a targeted supplement for the specific heads and languages where internal gold is still too thin — capped so that public data fills gaps without dominating the training set (Karan & Šnajder, 2018). In the current snapshot, no silver rows enter the final supervised splits. Internal gold rows are reviewer-confirmed final labels, whether moderators set them manually or confirm an existing suggestion. To protect label integrity, reviewer-confirmed labels can never be silently overwritten by later automated synchronization passes, and any row without a final confirmed label is excluded from the gold export.

Salting and Materialization

Public supplements are normalized into the shared five-head label space from five permissively licensed sources: Civil Comments (Dixon et al., 2018; Google, 2026), GAHD (Goldzycher et al., 2024; jagoldz, 2026), HateXplain (Mathew et al., 2021), the YouTube Spam Collection (Alberto et al., 2015), and the SMS Spam Collection (Almeida et al., 2011). Two functional evaluation suites are kept strictly outside the train/validation/test materialization path: HateCheck (Röttger et al., 2021) and Multilingual HateCheck German (Röttger et al., 2022).

Materialization follows four rules.

1. Keep all resolved internal gold rows.
2. Select internal silver first, but only from unresolved rows with stored upstream suggestions and with lower sample weight than gold.
3. Add public gold only under source-aware and internal-ratio caps, not through unrestricted pooling. When a head is sparse, allow more public positives as needed and keep matching same-source public negatives with them.
4. Keep functional evaluation suites separate from train, validation, and test.

The current implementation uses sample weights 1.5 for internal gold, 1.0 for public gold, and 0.35 for internal silver. Public data are capped both absolutely and relatively so that one large source cannot dominate the mix and the reviewed platform snapshot remains the main anchor while it is still small. In the current materializer, public supplementation is capped at 20% of the internal train anchor, each source is capped at 8% of that anchor, and sources that provide sparse-head positives may reserve up to 75 matched negatives from the same source and head. This source-balanced policy is motivated by evidence that

focused abusive-language datasets encode source-specific biases (Wiegand et al., 2019) and that multi-source training is more robust than single-dataset training (Antypas & Camacho-Collados, 2023). Oversized public splits are trimmed before the training data is assembled. The selection is deterministic — based on fixed content hashes rather than random draws — so the balance of languages and label types remains consistent and reproducible across runs, without depending on RNG state.

The split policy depends on how much reviewed internal data is available. Below an internal-evaluation threshold, all internal gold stays in train and validation/test are built from held-out public rows. Once internal volume is sufficient, internal rows are split by author group into train, validation, and test, and public rows are used only as held-out backfill for head/state combinations that would otherwise be unmeasurable. In the current snapshot, the workflow has already moved to internal author-grouped evaluation.

We also hardened the materialization stage against leakage. Duplicate `example_id` rows are removed before sampling, duplicate normalized content is collapsed within each supervised split, and normalized-content overlap across `train`, `validation`, and `test` is reduced to zero with priority ordering `test -> validation -> train`.

Materialized split summary.

Split	Rows	Composition
Train	6,324	internal gold + balanced public gold
Validation	448	internal hold-out + backfill
Test	372	internal hold-out + backfill
Func. eval.	7,373	HateCheck suites only

Across train, validation, and test, the current supervised snapshot contains 7,144 rows, of which 6,047 (84.6%) come from internal sources. No silver rows survive in the current supervised corpus even though a small unresolved internal pool still exists; public data now serves mainly as source-balanced coverage repair for sparse heads rather than as a major source of volume. In particular, the current train split retains same-source public negatives for the public-heavy `spam`, `hate_speech`, and `violence` heads instead of letting internal B90HQ rows provide almost all negatives by themselves.

Class-Conditional Embedding Audit

The current snapshot is large enough for held-out evaluation, but the positive labels it contains come mostly from public datasets rather than from internal platform data. Across supervised train/validation/test, 84.6% of rows are internal, yet native positive support is still extremely thin: the train split contains 13 internal `self_promotion` positives, 1 internal `spam` positive, and 0 internal positives for `hate_speech`, `violence`, or `privacy`. This creates an interpretation risk. If public datasets supply most positives for `spam`, `hate_speech`, and `violence`, then strong held-out results could reflect either genuine semantic learning — the model has learned what hate speech or spam actually looks like in language — or source-specific shortcut learning, where it has merely picked up on the vocabulary or formatting style of those particular public datasets, without having learned anything that transfers to real platform content.

We therefore add a class-conditional embedding audit. Using the same frozen `multilingual-e5-large` encoder as the strongest baseline, we compare public positives against internal platform content in embedding space. For each head with public positives, we compute: (i) cosine distance between the internal-negative centroid and

the public-positive centroid, (ii) nearest-neighbor source affinity, defined as the fraction of public positives whose closest non-self neighbor comes from internal data rather than another public row, and (iii) a linear MMD^2 shift score (Gretton et al., 2012) between internal negatives and public positives, with public negatives as a reference comparison. Because the snapshot contains almost no internal positives for these heads, the audit is intentionally narrow: it tests whether public toxic examples occupy roughly the same region as ordinary platform language and mix with internal content at all, not whether they already match a mature internal positive distribution. We also render two-dimensional UMAP projections (McInnes et al., 2018) for qualitative inspection.

This audit is interpretive rather than gatekeeping. If public positives mix naturally with internal content, held-out gains can be read more confidently as platform-relevant generalization. If they remain clustered by source, the same gains should instead be read as evidence that semantic pretraining helps but does not yet establish platform-native robustness.

As a control against over-interpreting source leakage, we also run a single-source sanity check on the public GAHD hate-speech corpus alone. This control uses the official GAHD train/dev/test split (Goldzycher et al., 2024), excludes all internal rows and all other public datasets, and compares the same frozen E5-large model against the sparse TF-IDF baseline on the one head that GAHD supervises (`hate_speech`). The goal is not to select a new production model from GAHD, but to test whether the simple embedding-plus-linear architecture can separate positives from negatives inside one public dataset even when data origin — whether a row came from the platform or a public dataset — is removed as a potential shortcut.

Baseline Model 1: Sparse TF-IDF Classifier

The first baseline is a sparse lexical model. Text is encoded by concatenating three feature blocks: (i) word-level TF-IDF features over lowercased 1–2-grams with sublinear term frequency and minimum document frequency 2, (ii) character `char_wb` TF-IDF features over 3–5-grams with minimum document frequency 1, and (iii) one-hot language metadata for `input_language`. The explicit language token `__lang_<code>__` is prepended to the word-view text so that lexical features remain language-aware even before the metadata block is added. Feed-audience and author-role metadata were removed from the trainable contract because they leaked source identity too directly across internal and public rows. TF-IDF follows the classical vector-space weighting formulation (Manning et al., 2008).

Each moderation head is trained as an independent logistic-regression classifier with `liblinear`, balanced class weights, and regularization parameter $C = 2.0$. Rows with target state `unknown` for a given head are excluded from that head's loss. In practice, the model treats the task as five partially observed binary problems rather than one dense multilabel matrix. At inference time, per-head decision thresholds are tuned on validation by maximizing F1, with balanced accuracy and distance from the default threshold used as tie-breakers.

Baseline Model 2: Frozen Multilingual Embedding Classifier

The second baseline replaces sparse lexical features with frozen dense sentence representations. It uses `intfloat/multilingual-e5-large` (Wang et al., 2024) as the current default encoder, selected after a validation-first benchmark sweep against the earlier `multilingual-e5-base` default and `multilingual-e5-small`. Input text is prefixed with `query:` for E5-style encoding, tokenized to a maximum length of 256, mean pooled

over the last hidden state using the attention mask, and then ℓ_2 normalized.

The dense text representation is combined with the same one-hot language metadata block used by the sparse baseline: `input_language`. The classifier layer again uses one independent logistic-regression head per moderation target, now with $C = 8.0$, balanced class weights, and the same head-specific exclusion of `unknown` rows from training loss. Threshold tuning on validation is identical to the TF-IDF baseline, but uses midpoint candidates between observed validation scores to reduce overfitting on very sparse heads such as `self_promotion`.

We chose this architecture for pragmatic reasons. It preserves a fully local training and inference path, avoids the data volume and infrastructure requirements of full encoder fine-tuning, and still benefits from multilingual semantic transfer learned during large-scale contrastive pre-training (Wang et al., 2024). The explicit language branch is retained because moderation decisions remain multilingual and language-conditioned even after feed-audience and author-role metadata are removed from the trainable dataset contract.

Evaluation Protocol

All models are evaluated under the same multi-head moderation protocol. Metrics are computed separately for each head using only rows with non-`unknown` ground truth for that head, and macro summaries are reported only over heads that are meaningfully comparable. Validation macro F1 is the main model-selection criterion, with validation macro average precision used as a secondary tie-breaker. Extremely sparse heads such as `self_promotion` are interpreted separately so that a threshold change on a single held-out positive does not determine the default configuration. Test results are reported as a held-out check on stability rather than as the main sweep target.

Beyond point estimates, we report non-parametric bootstrap confidence intervals for each model and paired bootstrap significance tests for model deltas — which check whether the performance gap between models is larger than what sampling variation alone would explain — following the general recommendations of (Koehn, 2004) and (Berg-Kirkpatrick et al., 2012). Each evaluation uses 2,000 bootstrap resamples and reports percentile confidence intervals for per-head F1, average precision, balanced accuracy, Brier score, and log loss. Candidate-versus-baseline comparisons are paired at the row level so that each resample preserves aligned predictions for both systems. For hard-label disagreements, we additionally report exact McNemar tests. Because the moderation setting introduces multiple per-head comparisons, label-level p-values are adjusted with Benjamini–Hochberg FDR control (Benjamini & Hochberg, 1995). Macro significance is reported only for heads with at least five positive rows on the evaluated split, so one-row heads such as `self_promotion` do not dominate the uncertainty estimate. In the current internal held-out `test` split, this restricts macro significance reporting to `spam`, `hate_speech`, and `violence`.

Because subgroup bias remains a central concern in hate-speech modeling (Dixon et al., 2018), we also report functional hate-speech evaluation outside the standard train/validation/test materialization. These suites do not yet span all five moderation heads, but they provide a stronger check on brittle lexical shortcuts than standard held-out IID evaluation alone.

Results

Current Baseline Comparison

Baseline comparison on stable heads and functional hate-speech evaluation.

Model	Val. F1/AP; LL	Test F1/AP; LL	Func. HS F1/AP; LL
TF-IDF	0.8822 / 0.9218; 0.0569	0.8022 / 0.8974; 0.0846	0.6738 / 0.7496; 0.7567
E5-large	0.9484 / 0.9748; 0.0431	0.8870 / 0.9662; 0.0557	0.8223 / 0.8388; 0.8185

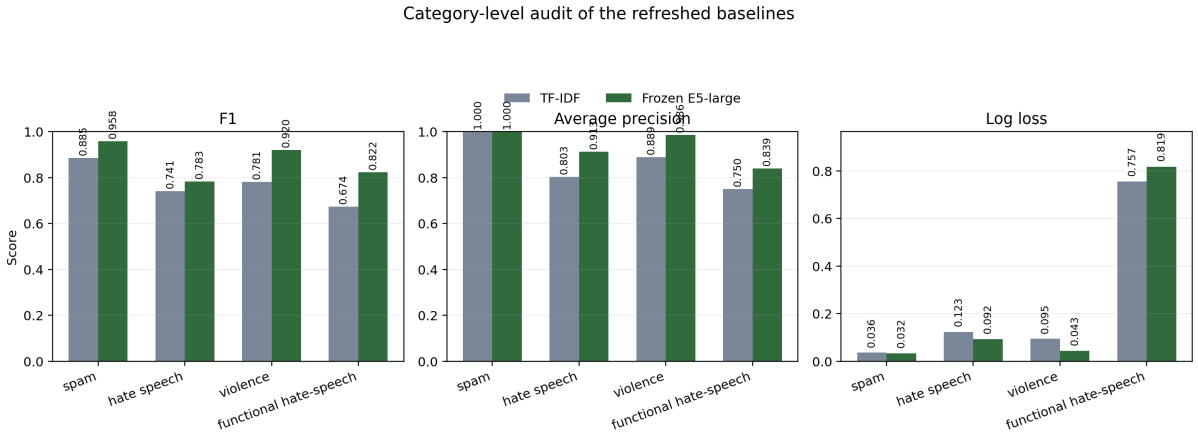

Figure 2: Per-head comparison across stable test heads and functional hate-speech evaluation.

The primary comparison is between the refreshed TF-IDF baseline and the frozen E5-large model. On the stable heads, the frozen E5-large model is stronger on validation and held-out test, so the simplest defensible model-selection decision is still to keep it as the default semantic baseline.

The interpretation is narrower than the old four-head comparable macro might suggest. The benchmark macro over all measurable heads is still distorted by `self_promotion`, which contributes only one positive row in validation and test, so inferential reporting now uses the stable heads with at least five positives: `spam`, `hate_speech`, and `violence`. Against the refreshed TF-IDF baseline, the frozen E5-large model raises stable-head

macro F1 from 0.8022 to 0.8870, with paired-bootstrap delta +0.0848, 95% confidence interval [0.0095, 0.1646], and $p = 0.022$. The same restricted comparison is stronger on ranking quality and calibration: stable-head macro average precision rises from 0.8974 to 0.9662, with paired-bootstrap delta +0.0689, 95% confidence interval [0.0292, 0.1165], and $p = 0.001$, while stable-head macro log loss falls from 0.0846 to 0.0557, with delta -0.0289, confidence interval [-0.0425, -0.0156], and $p = 0.001$.

Per-head test results on stable heads.

Head	+/-	TF-IDF P/R/F1	E5-large P/R/F1	TF-IDF AP/LL	E5-large AP/LL
`spam`	23 / 288	0.793 / 1.000 / 0.885	0.920 / 1.000 / 0.958	1.000 / 0.0361	1.000 / 0.0325
`hate_speech`	27 / 317	0.741 / 0.741 / 0.741	0.947 / 0.667 / 0.783	0.8029 / 0.1232	0.9128 / 0.0921
`violence`	27 / 306	0.676 / 0.926 / 0.781	1.000 / 0.852 / 0.920	0.8892 / 0.0947	0.9859 / 0.0427

On `spam`, both systems already recover all positives, but the embedding model reduces false positives from 6 to 2 and lifts precision from 0.793 to 0.920; average precision is saturated at 1.0, so the category remains descriptive rather than inferential. On `hate_speech`, the embedding model flags fewer cases overall but is more confident when it does flag something. Thresholded F1 improves only modestly, but the model's ability to rank

suspicious content and assign meaningful confidence scores is markedly better: average precision rises from 0.8029 to 0.9128, with paired-bootstrap delta +0.1099, 95% confidence interval [0.0148, 0.2203], $q = 0.040$, and log loss falls from 0.1232 to 0.0921, with delta -0.0311, confidence interval [-0.0589, -0.0038], $q = 0.037$. On `violence`, where missed cases carry the highest platform risk, the embedding model shows the clearest improvement across all metrics: F1 rises from 0.7813 to 0.9200 ($q = 0.048$), average precision rises from 0.8892 to 0.9859 ($q = 0.008$), and log loss nearly halves from 0.0947 to 0.0427 ($q = 0.004$).

The functional hate-speech suites provide the clearest thresholded margin and the clearest calibration warning. On HateCheck plus Multilingual HateCheck German, the frozen E5-large model improves F1 from 0.6738 to 0.8223 and average precision from 0.7496 to 0.8388 relative to the refreshed TF-IDF backstop; both paired-bootstrap tests yield $p = 0.001$. At the same time, log loss worsens from 0.7567 to 0.8185, also with $p = 0.001$. The functional suite therefore supports the same practical conclusion as the held-out test split, but it also shows why log loss belongs in the main audit surface: the embedding model ranks and classifies hate-speech examples better on this adversarial slice, yet it is less well calibrated there.

Benchmark Coverage and Limits

Positive and negative label counts by head and split.

Head	Train +/-	Val. +/-	Test +/-
`self_promotion`	13 / 4473	1 / 365	1 / 288
`spam`	227 / 5233	25 / 365	23 / 288

Head	Train +/-	Val. +/-	Test +/-
`hate_speech`	315 / 5616	26 / 391	27 / 317
`violence`	96 / 5448	26 / 379	27 / 306
`privacy`	0 / 5375	0 / 373	0 / 299

Source provenance by head and label polarity.

Head	Positive provenance	Negative provenance
`self_promotion`	int 13/1/1	int 4473/365/288
`spam`	int 1/0/0; YT 121/12/12; SMS 105/13/11	int 5087/365/288; YT 72/0/0; SMS 74/0/0
`hate_speech`	HX 151/7/7; GAHD 89/8/7; CC 75/11/13	int 5375/373/299; CC 91/18/18; GAHD 75/0/0; HX 75/0/0
`violence`	CC 96/26/27	int 5374/373/299; CC 74/6/7
`privacy`	none	int 5375/373/299

Abbreviations: int = internal gold, CC = Civil Comments, HX = HateXplain, YT = YouTube Spam, SMS = SMS Spam.

These tables explain why the paper distinguishes between a benchmarking macro and an inferential macro. The full four-head benchmark macro is still useful for comparing complete runs under the current contract, but inferential claims should focus on `spam`, `hate_speech`, and `violence`, where the held-out split has enough positive support to

make resampling meaningful. `self_promotion` remains exploratory, and `privacy` is not yet trainable.

The same caution applies to language claims. English is the only held-out slice that currently retains four comparable heads; there, the frozen E5-large model improves macro F1 / macro average precision from 0.6274 / 0.7788 to 0.9273 / 0.9776. German currently supports only one comparable head on held-out test, where F1 / average precision improve from 0.6000 / 0.8656 to 0.8333 / 0.8905. The `mixed` slice is too small to interpret, and the `und` slice remains label-sparse and dominated by not-measurable heads. We therefore use "multilingual" in a narrow operational sense: the pipeline and models accept multiple languages, but the current paper does not yet establish language-parity robustness.

Source-Separation Checks

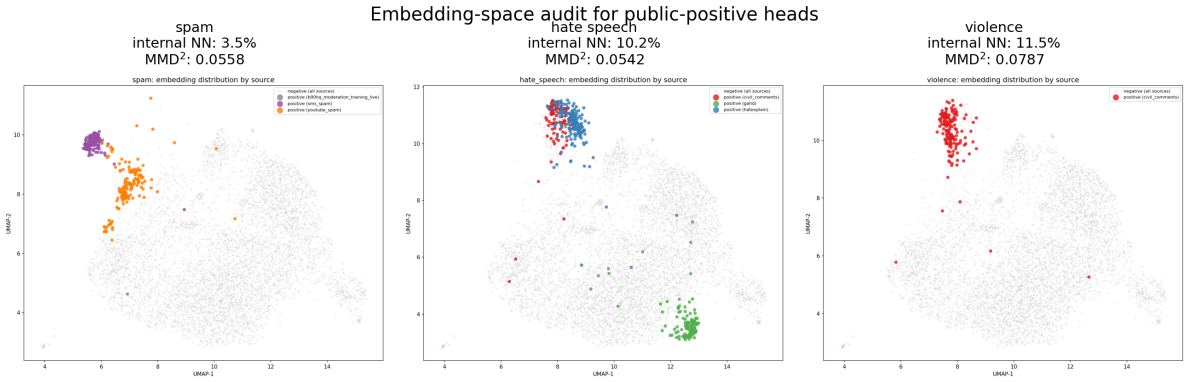

Figure 3: Embedding-space audit for public-positive heads.

The class-conditional embedding audit shows the same pattern across the public-heavy heads: broad proximity, but local clustering by source. For `spam`, `hate_speech`, and `violence`, the cosine distance between the public-positive centroid and the internal-negative centroid remains low at 0.0346, 0.0341, and 0.0492. The public positives therefore do not look obviously out of domain relative to the platform's sanitized text space.

The nearest-neighbor affinity signal is less reassuring. Only 3.5% of public `spam` positives, 10.2% of public `hate_speech` positives, and 11.5% of public `violence` positives are closer to an internal row than to another public row. In other words, the public positives occupy roughly the same broad region as internal content while still forming tight source-shaped clusters. The MMD^2 scores tell the same story: public-positive shift relative to internal negatives is mild for `spam` (0.0558) and `hate_speech` (0.0542), and largest for `violence` (0.0787), where the public-negative reference is also elevated (0.0737). We therefore treat `violence` as the most shifted head in the current materialization.

This is the central dataset-validity result. The audit weakens the simplest failure story, in which public positives would sit far outside the platform's semantic region and make the task little more than dataset identification. At the same time, it does not justify the opposite extreme. Local clustering remains strong enough that public-source regularities are still a plausible part of what the current models exploit, especially for `violence`. The interpretation should therefore be read conditionally: because most training examples for `spam`, `hate_speech`, and `violence` come from public datasets rather than real B90HQ moderation cases, the model may partly be learning patterns specific to those datasets rather than what abuse actually looks like on this platform. The current benchmark is good enough to select an initial model and support early deployment, but the scores cannot yet be taken as evidence that the model already understands native platform abuse patterns.

Single-Source Sanity Check

GAHD-only single-source control.

Split	Rows	+/-	TF-IDF F1/AP	E5-large F1/AP
Validation	1,648	709 / 939	0.7040 / 0.7177	0.7712 / 0.8015
Test	1,646	692 / 954	0.6912 / 0.7018	0.7738 / 0.8014

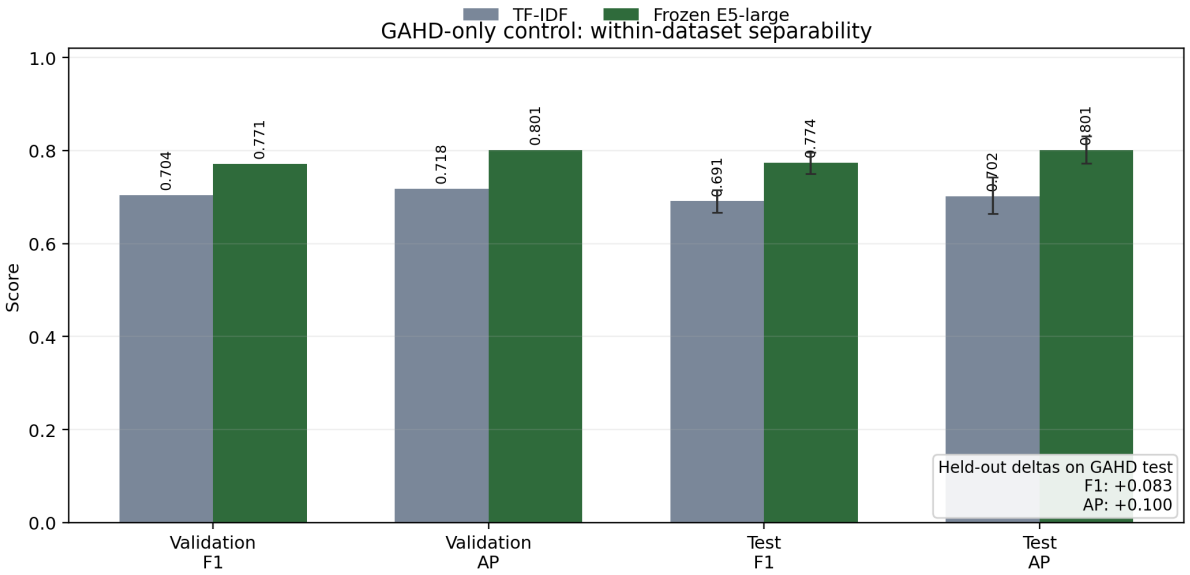

Figure 4: GAHD-only control with 95% bootstrap confidence intervals.

The single-source slice is non-trivial: GAHD contributes 7,701 German train rows, 1,648 validation rows, and 1,646 held-out test rows, with substantial hate-speech support in both evaluation splits. This matters because the control removes internal rows, removes source mixing, and still leaves enough labeled data for a meaningful held-out comparison.

Under that constraint, the frozen E5-large model reaches hate-speech validation F1 0.7712, validation average precision 0.8015, test F1 0.7738, and test average precision 0.8014. The sparse TF-IDF comparator on the same slice reaches validation F1 0.7040, validation average precision 0.7177, test F1 0.6912, and test average precision 0.7018.

The paired significance analysis on the held-out GAHD test split shows that this gap is unlikely to be sampling noise. Relative to TF-IDF, the frozen embedding model im-

proves F1 by +0.0826 with 95% confidence interval [0.0627, 0.1034] and average precision by +0.0996 with confidence interval [0.0652, 0.1326]; both paired-bootstrap tests yield $p = 0.001$ after FDR correction. McNemar testing points in the same direction, with 270 candidate-only correct rows versus 92 baseline-only correct rows. This control does not remove the cross-source generalization concern diagnosed by the embedding audit, but it does narrow the interpretation: the main weakness lies in transfer across source distributions, not in learning a within-corpus moderation boundary.

Taken together, the audit and the GAHD control answer the paper's main validity question more precisely than aggregate accuracy alone. The approach is not merely separating internal from public data: the same semantic recipe still wins inside one public corpus with no provenance shortcut available. But because the mixed-source benchmark remains much stronger than this single-source control, the current headline results are still likely helped by source separation. The remaining uncertainty is concentrated in transfer from public corpora to future platform-native positives.

Discussion

Interpretation of the Current Results

The results answer the first two research questions fairly clearly. For RQ1, the current training contract already yields a usable held-out benchmark even before the platform has accumulated a large native positive corpus. For RQ2, the frozen multilingual embedding model is the strongest refreshed model family under that contract. Its advantage is supported not only by thresholded gains but also by stronger ranking and calibration behavior.

RQ3 requires a narrower reading. The GAHD-only control makes the interpretation more precise. On a single public corpus with 7,701 train rows and 1,646 held-out test rows, the same frozen embedding model still beats TF-IDF by a wide and statistically credible margin. At the same time, the mixed-source benchmark is materially stronger than this single-source control. That gap is important: it is consistent with the suspicion that current performance is still helped by source separation and by the easier parts of the present train/test mixture. The control therefore should not be read as evidence that source effects are solved. It should be read more narrowly: the core embedding-plus-linear recipe is not merely exploiting the internal-versus-public split. The architecture can learn a real hate-speech boundary within one dataset. What remains unresolved is how far that learned boundary transfers once the source distribution changes from GAHD-style public text to B90HQ-native moderation incidents.

Threats to Validity

The main remaining threat is cross-source transfer. As established in Section 4.1, part of the observed gain likely reflects source-specific patterns in the public training data rather than platform-native abuse boundaries. The risk is concentrated in `violence` (largest distribution shift in the embedding audit) and `spam` (lowest public-to-internal neighbor affinity), where dependency on public positives is highest.

The second threat is label-support asymmetry. `self_promotion` has only one positive row in each held-out split and `privacy` has no positive supervision at all, meaning neither head currently supports inferential claims. The present dataset is useful for evaluating `spam`, `hate_speech`, and `violence`, while `self_promotion` remains exploratory and `privacy` is not yet trainable.

Language coverage reflects the platform's actual usage: the model is trained and evaluated on English and German, which are the two languages actively used on the platform and are expected to remain so for the foreseeable future. Parity claims for other languages are not made here.

A further limitation is temporal incompleteness. The current dataset under-samples the tail of serious moderation events that a larger and longer-running community would eventually produce. The present evaluation answers a narrower question: whether reviewed production signals plus bounded public supplementation are sufficient to initialize a practically useful moderation model. Whether the same model family remains well-calibrated as the platform accumulates a much richer native abuse distribution remains an open question.

Operational Implications

Operationally, the audit supports continuing with the current pipeline, but it also argues for conservative deployment claims. The present model family is suitable as an initial layer for moderator assistance and threshold-support experiments, not as evidence that public data can permanently stand in for native review data. In particular, the strongest near-term deployment path is for the better-supported heads, with `self_promotion` treated as exploratory, `privacy` kept outside automated model claims until positive supervision exists, and language-specific parity claims deferred until the held-out slices are less asymmetric.

As additional platform-native reports are reviewed upstream, the most important follow-up is to rerun the same audit and check whether public-positive clusters begin to mix more naturally with internal positives and hard negatives. That trend would be a stronger

sign that future evaluation gains reflect platform-native learning rather than only successful cross-domain transfer. In that sense, the present results are best understood as a bootstrap trajectory rather than as an end state: the current approach already works well enough to justify continued use, but it should become more trustworthy only as the platform accumulates more native positives and the source-separation signal weakens. Beyond that, the main model-side follow-up is to keep refreshing the same frozen embedding-plus-linear recipe against stronger internal-positive coverage and confirm that the current quality-versus-compute tradeoff remains stable as the benchmark matures.

Conclusion

This paper establishes a production-grounded labeling contract, a privacy-preserving ingestion and salting pipeline, and refreshed local baselines evaluated on a leakage-hardened held-out dataset. Together, these pieces provide a workable approach to moderation modeling for the platform's two active languages, English and German, under operational privacy and review constraints. The strongest current practical decision is to use the frozen E5-large model as the default local model family. The strongest current limitation is that part of the present gain still likely comes from source separation, so the benchmark is not yet a source-agnostic measure of production robustness. The single-source control nevertheless shows that the modeling approach itself is viable and should improve as more platform-native positives arrive. Under limited native positive supervision, the results should therefore be read as support for an initial moderation system and a valid bootstrap dataset on a path toward a stronger benchmark, not as final evidence of mature source-robust moderation performance.

Ethics Statement

The system is designed so that no raw PII is ingested into the trainable model inputs. Production text is sanitized before export, direct identifiers and contact-like strings are replaced by placeholders, and the downstream training workflow consumes only this sanitized representation.

The remaining semantic content is limited to user-visible platform content that is already accessible to other users within the product context rather than to private back-office records. The current Terms of Service state that users "grant us a non-exclusive, worldwide, royalty-free, sublicensable licence to host, reproduce, modify, and display your content as needed to operate, promote, and improve the Service, including training models for spam detection and safety" (B90-Industries, 2026).

References

Alberto, T. C., Lochter, J. V., & Almeida, T. A. (2015). *YouTube Spam Collection*. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/380/youtube+spam+collection

Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). *SMS Spam Collection*. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/228/sms+spam+collection

Antypas, D., & Camacho-Collados, J. (2023). Robust hate speech detection in social media: A cross-dataset empirical evaluation. *Proceedings of the 7th Workshop on Online Abuse and Harms*, 231–242. https://aclanthology.org/2023.woah-1.25/

B90-Industries. (2026). *B90HQ terms of service*. https://www.joinhq.app/legal/terms-of-service

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, *57*(1), 289–300. https://www.jstor.org/stable/2346101

Berg-Kirkpatrick, T., Burkett, D., & Klein, D. (2012). An empirical investigation of statistical significance in NLP. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 995–1005. https://aclanthology.org/D12-1091/

Dixon, L., Li, J., Sorensen, J., Thain, N., & Vasserman, L. (2018). Measuring and mitigating unintended bias in text classification. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 67–73. https://doi.org/10.1145/3278721.3278729

Goldzycher, J., Röttger, P., & Schneider, G. (2024). Improving adversarial data collec-

tion by supporting annotators: Lessons from GAHD, a German hate speech dataset. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 4405–4424. https://doi.org/10.18653/v1/2024.naacl-long.248

Google. (2026). *Civil comments*. Hugging Face dataset card. https://huggingface.co/datasets/google/civil_comments

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, *13*(25), 723–773. https://jmlr.org/papers/v13/gretton12a.html

jagoldz. (2026). *GAHD*. Hugging Face dataset card. https://huggingface.co/datasets/jagoldz/gahd

Karan, V. M., & Šnajder, J. (2018). Cross-domain detection of abusive language online. *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 132–137. https://aclanthology.org/W18-5117/

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 388–395. https://aclanthology.org/W04-3250/

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press. https://nlp.stanford.edu/IR-book/

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021). HateXplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*, 14867–14875. https://ojs.aaai.org/index.php/AAAI/article/view/17745

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation

and projection for dimension reduction. *arXiv Preprint arXiv:1802.03426*. https://arxiv.org/abs/1802.03426

Röttger, P., Seelawi, H., Nozza, D., Talat, Z., & Vidgen, B. (2022). Multilingual HateCheck: Functional tests for multilingual hate speech detection models. *Proceedings of the 6th Workshop on Online Abuse and Harms*, 87–102. https://aclanthology.org/2022.woah-1.15/

Röttger, P., Vidgen, B., Nguyen, D., Waseem, Z., Margetts, H., Pierrehumbert, J., & Rayson, P. (2021). HateCheck: Functional tests for hate speech detection models. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 41–58. https://aclanthology.org/2021.acl-long.4/

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., & Wei, F. (2024). Multilingual E5 text embeddings: A technical report. *arXiv Preprint arXiv:2402.05672*. https://arxiv.org/abs/2402.05672

Wiegand, M., Ruppenhofer, J., & Kleinbauer, T. (2019). Detection of abusive language: The problem of biased datasets. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 602–608. https://doi.org/10.18653/v1/N19-1060