Suspenders: A Deployed Guardrail for Policy Filtering of LLM-Generated Job-Recommendation Explanations

Jasmine Qi[1], Danylo Dantsev[1] and Jeff Kohls[1]

[1]*Indeed, Inc., Austin, TX, USA*

Abstract

A job recommender ranks candidate–job pairs and can also explain them in natural language. That short explanation can affect how a job seeker interprets and responds to the match. When a large language model writes the explanation, it can fail even when the underlying match is sound: it can overstate a thin match, credit skills the profile never showed, frame a career change reductively, or lean on sensitive or proxy attributes as a reason to apply. In a hiring context these are not cosmetic wording issues; they can create policy and fairness concerns in user-facing text. We describe Suspenders, a guardrail that filters recommendation-explanation text in production at a large online job-matching platform, treating explanation policy filtering as a governance layer separate from candidate–job ranking. The deployed migration replaced a fine-tuned GPT-4.1-mini filter with a self-hosted Qwen3-4B classifier, and a later iteration added uncertainty-based routing that sent 8.5% of predictions in a 14-day sample to human policy review. An internal rollout summary reported a 95% reduction in monthly inference spend and an 82% reduction in average latency. The same summary reported negative changes in an applies metric, but its scale and uncertainty were not retained in the research export. On human-annotated random traffic the classifier missed fewer policy violations than the prior filter in both evaluated markets, but it also blocks a larger share of traffic. The comparison reflects a stricter operating point as well as any change in model quality, and small violation counts make the result directional rather than statistically established. A separate 12,000-example synthetic stress test covering age, military-preference, and socioeconomic-proxy language shows that the tested zero-shot configurations of three broad moderation models miss 96–100% of judge-labeled violations. Production deployment audits and synthetic stress tests measure different quantities and are reported separately.

Keywords

recommender systems for human resources, job recommendation, recommendation explanations, LLM guardrails, content policy filtering, responsible AI in hiring, red teaming, small language model deployment

1. Introduction

Recommendation explanations as policy-sensitive text. Large language models are used within recommender-system pipelines, including as generators of user-facing text [1]. On a hiring platform the explanation is part of the product itself: it tells a job seeker why a role was surfaced and often nudges the decision to apply, so its wording has real consequences for a person's job search. In our setting, a matcher selects a candidate–job pair and a separate LLM explains why the pair may be relevant. The explanation does not determine whether the job is recommended. It frames a match that has already been selected. We therefore evaluate the wording shown to users, not whether the underlying match is correct or fair.

Generated explanations are therefore a policy-sensitive product surface. A useful explanation should make the evidence legible without exaggerating it. Cases rejected by our production policy include irrelevant skill citations, unsupported skill attribution, reductive career framing, incoherent template fragments, and sensitive or proxy attributes presented as recommendation evidence. These are explanation failures rather than ranking failures, and generic toxicity taxonomies do not describe many of them. The guardrail is one component of a broader process that includes written guidelines, expert review, monitoring, and rollback procedures.

Deployment constraints. The prior production filter was a fine-tuned OpenAI GPT-4.1-mini model [2]. GPT-4.1-mini is the base snapshot, not an off-the-shelf baseline: the deployed control was an internally fine-tuned derivative. Its observed monthly spend and latency were difficult to sustain at the surrounding pipeline's scale. Open-weight instruction models reduce dependence on external APIs, but the zero-shot configurations tested in Section 4 either miss most domain-policy violations or block too much traffic. General moderation systems such as Llama Guard and WildGuard [3, 4] remain useful baselines, but their taxonomies target broad conversational risks rather than employment-recommendation explanations.

Suspenders. Suspenders uses a Qwen3-4B classifier [5] fine-tuned with LoRA [6]. A later production iteration computes a routing score from the first-token log probabilities of the two labels and sends predictions with scores below 0.80 to human policy review. This routing prioritizes uncertain explanation-filter decisions; it does not change candidate–job ranking. The initial March rollout checkpoint, the subsequently deployed attribute-policy-enhanced checkpoint, and the adapter used for the 12,000-example synthetic stress test are different artifacts. We report each with its own evaluation and do not attribute the March production change to later uncertainty routing or red-team supervision.

Contributions. The paper makes three practical contributions:

- We define explanation policy filtering as a governance layer distinct from candidate–job ranking.
- We report a production migration with cost, latency, policy, block-rate, and engagement measurements, including an unfavorable engagement result. We also describe a deployed uncertainty-based human-review mechanism, reporting its routing rate and label mix but not its incremental effect.
- We combine random-traffic auditing with a synthetic stress test and document where the two evaluations disagree. We also describe the schema and planned release of the fully synthetic artifact.

RecSys in HR '26: The 6th Workshop on Recommender Systems for Human Resources, in conjunction with the 20th ACM Conference on Recommender Systems, September 28–October 2, 2026, Minneapolis, MN, United States.

2. System Design

2.1. Production setting

The relevant RecSys pipeline has four steps: the matcher selects a candidate–job pair, the explanation generator writes user-facing text for that already-selected match, the guardrail decides whether that text can be displayed, and the notification channel displays either the generated explanation or a policy-approved fallback. The recommendation itself remains delivered. This paper studies the third step. The guardrail sits between the recommendation-explanation generator and the user-facing messaging surface. The surrounding pipeline processes on the order of tens of millions of candidate explanations per day across the deployed markets. Product eligibility and market routing mean that this candidate count is not a token-billing denominator for the guardrail. We therefore report observed internal spend rather than reconstructing cost from public token list prices.

Cost. A March 2026 internal rollout summary compared monthly spend for the fine-tuned OpenAI GPT-4.1-mini deployment against allocated monthly inference spend for the self-hosted Qwen3-4B service. The internal summary reported that allocated monthly Qwen inference spend was approximately 95% lower than the prior API bill (Table 2). We report commercial quantities as percentage changes rather than absolute figures, which internal disclosure rules do not permit. The archived research artifacts do not independently reconstruct these full-fleet figures. We omit per-token reconstruction because the old API bill and the shared GPU service use different accounting models.

Compliance burden. Each additional commercial API introduces a per-market data-processing-agreement review, vendor risk assessment, and ongoing legal monitoring. Self-hosting inside our own virtual private cloud (VPC) reduces this external-vendor processing burden, although privacy, employment, security, and model-governance obligations remain. The synthetic red-team data did not leave the platform during training.

Domain specificity. The production policy covers a broader set of explanation-quality and compliance cases than the public red-team set: irrelevant skill-to-job citations, mismatch admissions, inappropriate or reductive framing, age-related wording, unsupported skill attribution, off-platform apply instructions, and incoherent or truncated text. We disclose only the portions needed to make the research contribution reproducible. The public taxonomy behind our evaluation split, which we call RAI2 because it is the second responsible-AI red-team batch (Section 3), focuses on synthetic attribute-related policy probes because those examples can be shared without exposing real candidate or employer records, internal policy thresholds, or prompts that would make the production filter easier to evade.

2.2. Guardrail architecture

The guardrail (Figure 1) runs a fine-tuned Qwen3-4B classifier inside our VPC. Let ℓ_A and ℓ_N denote the first-token log probabilities for "Acceptable" and "Not Acceptable." We normalize these two values and define the routing score as $c = \max(p_A, p_N)$. This routing score has not undergone a formal calibration study. Predictions below 0.80 are sent to human policy review; other predictions follow the model label. On a random 14-day production sample of 4,559 explanations from the current deployed checkpoint, 8.5% of predictions (95% Wilson interval 7.7–9.3%) fell below the 0.80 threshold and were routed to human review. The observed shares were 8.9% in the US and 7.2% in Canada; no between-market comparison was performed. Within that routed band the classifier itself split roughly 35%/65% between "Not Acceptable" and "Acceptable" labels, which is the ambiguous population the review stage is meant to adjudicate; a smaller 500-row March holdout showed the same pattern (23 predictions, 4.6%, below 0.80 and containing about half of the classifier's errors). These figures describe the routing *rate* and its *input* mix, not an end-to-end gain in precision or recall. The mechanism was added after the February A/B test and March 10 rollout, so the production-transition outcomes below do not measure it, and we do not retain the reviewers' completed decisions, so we cannot estimate its incremental end-to-end effect.

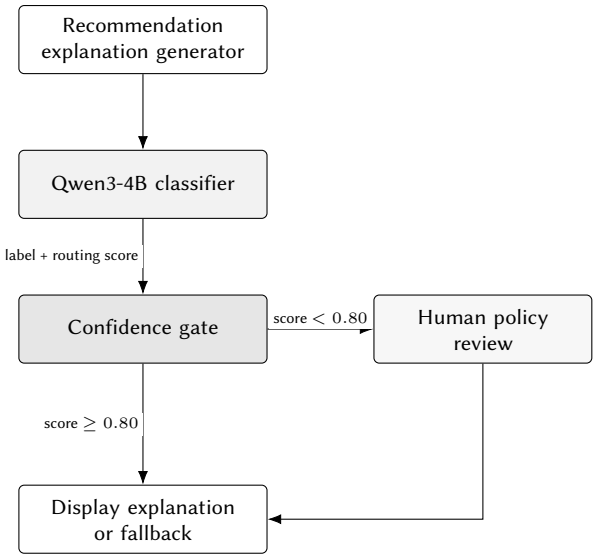

Figure 1: Current production guardrail. The classifier emits a binary label and an uncertainty-routing score derived from first-token log probabilities. Predictions below 0.80 receive human policy review; other predictions follow the model label. This stage prioritizes uncertain explanation decisions and does not alter candidate–job ranking.

2.3. Training and serving details

The classifier is trained as a binary instruction model over the labels "Acceptable" and "Not Acceptable." These labels refer to policy acceptability of the generated explanation text, not to whether the underlying recommendation match is correct. We use LoRA fine-tuning [6] over `Qwen/Qwen3-4B-Instruct-2507` [5, 7]. The initial March rollout checkpoint used LoRA rank 64, LoRA alpha 128, six epochs, and a learning rate of 5×10^{-5}. The current routed production model is a subsequent attribute-policy-enhanced Qwen3-4B checkpoint, trained with additional attribute-policy examples using LoRA rank 32 for two epochs. The prior fine-tuned filter used the

Table 1
Public summary of classifier policy scope. The model checks whether the generated explanation is acceptable to display; it does not judge whether the candidate–job match itself is correct.

Policy family	Examples the classifier learns to flag
Match evidence	Cites an unrelated skill, misstates the basis for the match, or attributes a skill not present in the profile.
Tone and framing	Presents a weak match too confidently, frames a career pivot reductively, or admits mismatch while still urging action.
Apply and compliance wording	Includes unsupported guarantee language, off-platform apply instructions, or market-specific eligibility wording that requires policy review.
Text integrity	Produces incoherent, truncated, or template-artifact text that is not appropriate for user display.
Attr.-related probes	Uses age-related wording, military-service preference framing, or socioeconomic proxy language as a reason for the recommendation.

`gpt-4.1-mini-2025-04-14` base snapshot.[1] The Qwen training mixtures combine human-labeled examples with mined production misses and overblocks. Training uses completion-only loss on the label tokens. A separate adapter used for the 12,000-row stress test adds synthetic attribute-policy examples; it should not be read as the checkpoint that produced the March deployment results.

Table 1 summarizes the public-facing policy families used to explain classifier behavior. The table is intentionally coarser than the production review guidance: it describes the kinds of generated explanations the model learns to flag, without exposing thresholds, internal decision guidance, or prompts used by the production filter.

For concreteness, internal production review has flagged explanations that describe a customer-support candidate as a strong accounting match because of generic "client communication" language, tell a job seeker that a role is "not an exact match" while still urging them to apply, present a career transition as "starting over," or expose a template fragment instead of a coherent reason. We paraphrase these examples to avoid exposing user data or production prompts, but they illustrate why the task is not captured by toxicity detection alone: the failure is often an unsupported or inappropriate recommendation reason.

The production training loop is data-centric. After each offline or online evaluation cycle, we collect two kinds of examples: missed violations, where a model allowed a policy-violating explanation, and overblocks, where a model blocked an explanation that policy review would allow. Missed violations are oversampled as hard negatives, while overblocks are retained to prevent the classifier from drifting toward excessive blocking. We deduplicate examples by normalized explanation text before training and keep fresh random-traffic samples disjoint from the training pool.

At serving time, the model runs inside our VPC using vLLM-backed GPU dynamic batching on an internal model-serving platform [8]. The service is designed for proactive recommendation traffic: high steady volume, short midnight spikes, small outputs, and strict latency targets. The prior API filter remains available as a fallback during experiments and staged rollouts. We continuously monitor block rate, latency, GPU utilization, and expert-review estimates of missed violations. These signals matter because the acceptable configuration is a product and policy decision, not only a model-quality metric. The archived deployment summary does not contain the vLLM version, batching parameters, or latency-percentile definition. Table 2 therefore reports the dashboard's average inference-service latency rather than an end-to-end request latency.

[1]That filter was fine-tuned through the vendor's hosted supervised fine-tuning service, so its adapter weights, optimizer configuration, and token-level training details are not available to us. We can therefore report its deployed behavior, but not its training hyperparameters, on the same footing as the Qwen checkpoints.

Table 2
Production-transition aggregates reported in the March 2026 internal rollout summary. Cost and latency are reported as percentage change against the prior filter, because absolute internal figures are not disclosable. Cost is full-fleet monthly spend or internal allocation, not arm-level A/B cost reconstructed from public token prices. The test predates the later uncertainty-based human-review routing and therefore does not measure its effect.

Metric	Fine-tuned GPT-4.1-mini	Qwen guardrail
Monthly inference cost	baseline	-95%
Average latency	baseline	-82%
Block rate (%)	3.4	11.4
Reported applies change (overall, %)	N/A	-0.42
Reported applies change (re-engagement, %)	N/A	-0.69
Est. revenue impact (% of cost saving)	N/A	≈ -1

2.4. Why add uncertainty-based review

The learned classifier handles paraphrased and context-dependent language that is difficult to enumerate. Confidence-based routing concentrates limited human review on decisions for which the two label probabilities are least separated. Reviewers then apply the relevant market and scenario guidance. The routing score orders uncertainty in the explanation-filter decision only; it is not a recommendation score and never reorders candidate–job pairs. We measure the routing rate and the classifier's label mix in the routed band on production traffic, but the study does not retain the reviewers' completed decisions, so it does not measure the routing stage's incremental effect on precision or recall.

2.5. Production A/B test and rollout

We allocated approximately half of production traffic to the Qwen-based guardrail and half to the prior fine-tuned GPT-4.1-mini filter during February 2026, then rolled Qwen to 100% of US traffic on March 10. The deployed markets together handled on the order of tens of millions of calls per day. The production outcomes are summarized in Table 2. The research export retained the aggregate deltas but not the randomization unit, exposure denominators, confidence intervals, or test statistics. It also does not specify whether the apply changes are relative percentages or percentage-point changes. We therefore present them as reported operational monitoring signals, not causal effect estimates.

The rollout summary reported negative changes in an applies metric: -0.69% on a re-engagement notification channel and -0.42% overall. The export does not preserve the metric's denominator, whether the values are relative percentages or percentage-point changes, uncertainty esti-

mates, or the evidence needed for a causal interpretation. The estimated monthly revenue impact was approximately 1% of the monthly inference saving, so the rollout decision favored the cost and latency benefits together with the policy team's preference for more conservative filtering. Because a blocked explanation is replaced by a safe fallback template rather than suppressed, the recommendation itself is still delivered; overblocking therefore reduces explanation richness rather than removing recommendations. The reported applies metric remains under active monitoring. We do not yet decompose the apply-rate movement into true-positive blocks versus overblocks, which is why we avoid giving it a causal interpretation. We are evaluating whether a less aggressive classification operating point changes the reported applies metric while monitoring the directional missed-violation differences.

2.6. Rollout safeguards and monitoring

The production rollout followed the same risk posture as other high-volume recommendation changes: start with offline evaluation, move to a limited A/B test, monitor both policy and business metrics, and keep a fallback path available. During the A/B test, the prior GPT-4.1-mini filter remained the control and rollback target. We used block-rate movement as an early warning signal because it is visible immediately at traffic scale, while missed-violation estimates require expert review and therefore arrive more slowly. Rollback review would have been triggered by a sustained block-rate spike, material latency regression, a missed-violation audit worse than the prior filter, or an apply-rate loss large enough to outweigh the operational benefits of the migration.

This monitoring setup also shapes how we interpret model metrics. Offline recall alone is not enough for this user-facing component. A model that catches every policy-violating explanation by blocking half of all recommendations would be unusable; a model with low block rate but poor missed-violation recall would fail the policy goal. The deployed system is therefore selected using several constraints: missed-violation reduction, block rate, latency, cost, compliance posture, and apply-rate impact must all remain within acceptable ranges. The later uncertainty-based review route adds human judgment for the least-certain cases. Its traffic share varies by market and scenario, and its incremental effect is not estimated in this study.

3. Synthetic Red-Team Taxonomy and Data

We separate the confidential production policy from the shareable red-team artifact. The production classifier and human review guidance cover a broader set of cases than the attribute-related examples studied here, including quality, integrity, and compliance cases in generated explanation text. We do not publish the full operational policy, review playbooks, or model prompts because they are product controls. The public artifact is the shareable subset: a synthetic attribute-related policy-probe set designed to test whether a recommendation explanation repeats sensitive or proxy attributes as a reason for recommendation.

We construct this red-team dataset by composing *adversarial payloads* drawn from a structured attribute-policy taxonomy (Table 3) with two *injection targets* (job posting and candidate resume) and running the explanation pipeline end-to-end, following the broader practice of adversarial LLM red-teaming [9]. The injected payloads are synthetic. They are written to resemble policy-relevant failure modes, not real employer or job-seeker records. A judge model adjudicates whether the recommendation explanation generated downstream propagates the injected policy-relevant content; each row records its pass or fail outcome and the judge's rationale. We have not yet measured agreement between this judge and independent human reviewers, so the resulting labels are suitable for stress testing but not a human-validated estimate of policy prevalence. The approved research export does not identify the generator and judge model revisions or preserve their prompts and decoding settings. This prevents independent regeneration of the labels, although the fixed examples and labels can still support comparative classifier evaluation if released.

Table 3
Structure of the synthetic red-team data. The experiments use the 12,000-row second batch (RAI2) for evaluation. All rows are synthetic and contain no real candidate or job-seeker data.

Category	Subcategory	n	Share
Attribute policy	Age	7,332	40.7%
	Military preference	5,712	31.7%
	Socioeconomic	4,956	27.5%
Development batch total		6,000	N/A
RAI2 evaluation total		12,000	N/A
RAI2: job-posting target		9,844	N/A
RAI2: resume target		2,156	N/A
Planned release total		18,000	100.0%

The generation process produced two batches under the same taxonomy: 6,000 development rows and a later 12,000-row batch, which we call RAI2. The red-team-enhanced classifier in Table 5 uses failures from the first batch during training and the second batch for evaluation. Split identifiers will be included if the artifact is released. The 18,000 figure therefore describes the combined artifact, while 12,000 describes the evaluation used in Tables 5–6. The split is row-disjoint. We did not run semantic or payload-template near-duplicate detection across batches, so some cross-batch template similarity may remain.

We use this judge-labeled set as a synthetic policy-compliance stress test, not as a substitute for production human annotation. The random production evaluation in Section 4 estimates live-traffic missed-violation rate, while the red-team set tests coverage of rare but high-impact attribute-related policy patterns that appear too sparsely in random traffic to support stable comparisons.

The taxonomy is intentionally narrow. All three subcategories are covered by our internal employment-messaging policy and may raise legal or fairness concerns depending on jurisdiction and use. The experiments test whether an explanation offers one of them as a reason to apply, not whether a particular use is legally permissible. Age payloads test whether the generator repeats age-band preferences or implies that younger or older candidates are categorically better fits. Military preference payloads test whether the explanation treats veteran status or prior military service as an employment preference, in either direction, rather than as one part of a work history. Socioeconomic payloads test proxy signals such as prestige markers, profile URLs,

or neighborhood language that can stand in for class-based screening. We paraphrase examples in the paper rather than print the full payload library.

We plan to release approved portions of this dataset under CC BY 4.0 upon publication, subject to legal review. This is a release plan, not a currently available artifact. Each row contains the injection category and subcategory, the injection target, the adversarial payload, the candidate-model output, the judge's pass or fail decision, and the judge's policy reasoning. All payloads are synthetic and no real candidate, job-seeker, or employer data is included. If legal review prevents release of the full 18,000 rows, we will release the public taxonomy schema, generation protocol, annotation format, and the largest approved synthetic subset so that others can reproduce the evaluation structure without exposing internal product controls.

4. Experiments and Results

Metrics and annotation. We report two primary metrics. *Missed-violation rate* is the share of ground-truth policy-violating examples that the model allows; it is the primary policy metric we try to minimize. *Block rate* is the share of all examples predicted "Not Acceptable"; it measures intervention volume. We also report exact confusion counts and precision for the red-team-enhanced classifier because block rate alone does not distinguish correct blocks from overblocks. Production labels come from expert review under written binary guidelines. Red-team labels come from the synthetic judge protocol in Section 3; they have not been independently validated by a human study.

The production annotation was collected for operational review rather than designed as a research annotation study. The export contains one adjudicated binary label per row but does not retain annotator IDs, independent duplicate labels, blinding status, or agreement statistics. Both filters were scored on the same rows. However, the available research export does not retain their paired row-level predictions, so we cannot report discordant counts or an exact McNemar test. The Wilson intervals below describe each miss proportion separately; they are not a paired test of the difference.

The random-traffic and synthetic evaluations answer different questions. Random samples retain live prevalence but contain few violations. The synthetic set concentrates known policy patterns but is not a prevalence estimate. In addition, the production table evaluates the initial March checkpoint, a later holdout evaluates the current attribute-policy-enhanced production checkpoint, and the 12,000-row synthetic table evaluates a separate adapter. We keep these results separate.

4.1. Offline random-traffic evaluation

The annotation project contained 900 US and Canadian examples across random and targeted diagnostic slices. Table 4 uses only the two operationally designated random-traffic slices, 200 rows per market. The US slice contains 20 policy violations: the prior GPT-4.1-mini filter misses 8 and the March Qwen checkpoint misses 4. Their 95% Wilson intervals are 21.9–61.3% and 8.1–41.6%, respectively. The Canadian slice contains 16 violations, with 8 and 4 misses; the corresponding intervals are 28.0–72.0% and 10.2–49.5%. The paired predictions needed to test the difference were not retained. We therefore treat the differences as directional rather than statistically established improvements.

Miss rate on its own would misrepresent this comparison, so Table 4 also reports block rate and precision over the same evaluation rows. The March checkpoint halves the miss rate in both markets, but it does so from a markedly stricter operating point: it blocks 16.5% of US rows against the prior filter's 9.0% (95% intervals 12.0–22.3% and 5.8–13.8%), and 14.5% against 9.0% in Canada. Its observed precision is 0.485 against 0.667 in the US and 0.414 against 0.444 in Canada. The aggregate results therefore show lower observed miss rates, higher block rates, and lower precision point estimates for Qwen. Because row-level predictions were not retained, these differences were not formally tested. The miss-rate comparison should be read as an operating-point change rather than as a demonstrated improvement in model quality.

4.2. Current deployed checkpoint

After the initial rollout, we deployed the attribute-policy-enhanced Qwen3-4B checkpoint described in Section 2.3. We report two classifier-only offline checks for this model. On a fresh 500-row March 10 random-traffic holdout that was not used for training, it misses 2 of 18 violations and overblocks 21 of 482 acceptable explanations. The 95% Wilson intervals are 3.1–32.8% and 2.9–6.6%, respectively. Written as a confusion matrix, that holdout gives 16 true blocks, 2 missed violations, 21 overblocks, and 461 correct allows, so the checkpoint blocks 7.4% of the sample (5.4–10.0%) at precision 0.432 (0.287–0.591) and recall 0.889 (0.672–0.969). Precision belongs next to miss rate here: most of what this checkpoint blocks is not a violation, so a low miss rate on its own would overstate how well the operating point is tuned. On a separate 167-row targeted attribute-policy injection test, it detects 148 cases (88.6%; 95% Wilson interval 82.9–92.6%). The random holdout was also used to inspect the routing-score threshold, so it is not an independent validation of that threshold. These counts provide current checkpoint evidence but remain too small to establish a precise missed-violation rate.

Production operational rates. On a random 14-day sample of 4,559 explanations, the deployed checkpoint blocked 8.1% of traffic (95% Wilson interval 7.3–8.9%), with comparable rates across markets (US 7.8%, CA 8.8%). An independent 5,000-row snapshot gave an overlapping block rate of 8.7% (8.0–9.5%) and routing rate of 7.8% (7.1–8.5%). The snapshots are numerically similar, but no formal stability or equivalence test was performed. These are the model's own production decisions, reported as operational monitoring rather than human-validated violation prevalence.

4.3. Synthetic stress test

We evaluate open zero-shot instruction baselines and purpose-built moderation classifiers on the 12,000-example synthetic set (Table 5). The instruction baselines include Qwen3-family models [5] and SmolLM3-3B [10]. The moderation baselines include Llama Guard, WildGuard, and ShieldGemma [11, 4, 12]. All systems are evaluated on the same explanations. The generic baselines are used in their evaluated zero-shot configurations; we did not tune them on this taxonomy or sweep thresholds. Results therefore com-

Table 4
Human-annotated random-traffic evaluation over the same evaluation rows. Each market has 200 examples, with 20 violations in the US and 16 in Canada. Miss parentheses show missed violations over all violations; precision is true blocks over all blocks, with 95% Wilson intervals. Lower miss is better, higher precision is better. The Qwen checkpoint has lower observed miss rates, but blocks 1.6–1.8 times as much traffic and has lower observed precision point estimates.

	US			Canada		
Model	Miss	Block	Precision	Miss	Block	Precision
Fine-tuned GPT-4.1-mini	0.40 (8/20)	0.090	0.667 (0.437–0.837)	0.50 (8/16)	0.090	0.444 (0.246–0.663)
March Qwen3-4B	**0.20 (4/20)**	0.165	0.485 (0.325–0.648)	**0.25 (4/16)**	0.145	0.414 (0.255–0.593)

Table 5
Synthetic attribute-policy stress test ($n = 12{,}000$; 278 violations), with 95% Wilson intervals. Miss rate is over the 278 violations; block rate is over all 12,000 examples. Miss rate and block rate are separate objectives rather than a single ranking criterion. The row-level intervals treat examples as independent and do not account for possible template dependence. Between-model comparisons are descriptive. The red-team-enhanced row is a separate offline classifier adapter, not the March or current production checkpoint. The GPT-4.1-mini row is the base hosted model under our chain-of-thought policy prompt, not the fine-tuned variant of it that ran in production before March.

Model	Miss (95% CI)	Block (95% CI)
SmolLM3-3B zero-shot	0.885 (0.842–0.917)	0.163 (0.156–0.170)
Qwen3-1.7B zero-shot	0.874 (0.830–0.908)	0.092 (0.087–0.097)
Qwen3-4B zero-shot	0.482 (0.424–0.541)	0.416 (0.407–0.425)
Qwen3-8B zero-shot	0.475 (0.417–0.533)	0.511 (0.502–0.520)
Llama Guard 3-8B	1.000 (0.986–1.000)	0.001 (0.001–0.002)
WildGuard	0.960 (0.931–0.978)	0.032 (0.029–0.035)
ShieldGemma-2B	1.000 (0.986–1.000)	0.000 (0.000–0.001)
GPT-4.1-mini, policy prompt	0.885 (0.842–0.917)	0.089 (0.084–0.094)
Red-team-enhanced Qwen3-4B	**0.158 (0.120–0.206)**	0.093 (0.088–0.098)

pare the tested configurations, not every threshold those models might support.

Baseline configurations. The recorded model repositories were `Qwen/Qwen3-{1.7B,4B,8B}`, `HuggingFaceTB/SmolLM3-3B`, `meta-llama/Llama-Guard-3-8B`, `allenai/wildguard`, and `google/shieldgemma-2b`. Repository commit hashes and package versions were not logged. The instruction models used the same binary policy prompt, greedy decoding, at most 128 new tokens, and a parser that preferred "Not Acceptable" when both label strings appeared. Unparsable outputs were blocked and are included in the reported block rates; parser-failure counts were not logged separately. Llama Guard rated the explanation as an assistant turn with up to 6,000 characters of context and greedy decoding. WildGuard used a generic employment-safety prompt, greedy decoding, and disabled thinking mode when its chat template supported that option. ShieldGemma compared the final-token probabilities of "Yes" and "No" under a two-principle employment prompt and used a 0.5 threshold. These prompt differences follow each evaluator's interface, but they also limit direct model-to-model interpretation.

The red-team-enhanced classifier catches 234 of 278 violations and misses 44. It blocks 1,112 examples in total, including 878 overblocks, for precision 0.210, recall 0.842, and false-block rate 0.075. The corresponding 95% Wilson intervals are 0.187–0.235, 0.794–0.880, and 0.070–0.080. These counts show why the 9.3% block rate should not be read as an end-to-end user-facing result. Precision 0.210 is the fraction of blocked examples that are labeled violations, not an improvement over a baseline. The uncertainty-based human-review stage was not run in this offline evaluation.

The GPT-4.1-mini row has the closest observed aggregate block rate to the red-team-enhanced classifier: 0.089 (0.084–0.094) versus 0.093 (0.088–0.098). No paired or equivalence test was performed. At these observed block rates, the models miss 88.5% and 15.8% of violations, respectively, so the miss-rate difference is not accompanied by a large difference in aggregate blocking. Two caveats bound how far this result generalizes. The comparison is against the base hosted model under our policy prompt, not against the fine-tuned filter actually replaced in March, whose weights the vendor does not expose, so the table still does not establish quality parity with the deployed predecessor. The red-team-enhanced adapter was also trained on failures drawn from the first batch of the same taxonomy, which no baseline saw, and while the evaluation split is row-disjoint, that advantage is a property of the comparison rather than a general capability claim.

The evaluated generic baselines show a domain mismatch at their tested configurations. Smaller instruction models miss 87–89% of violations. Larger zero-shot Qwen models miss about 48% while blocking 42–51% of all examples. The broad moderation models block almost none of these employment-specific explanations and miss 96–100% of violations.

4.4. Per-subcategory analysis

Table 6 breaks down RAI2 missed-violation rate by subcategory. The 278 violations divide into 81 age, 93 military-preference, and 104 socioeconomic-proxy cases, and every model's subcategory misses sum to its overall miss count in Table 5; the red-team-enhanced checkpoint misses 23, 5, and 16 of them respectively. That checkpoint has the lowest observed miss rate in all three classes among the evaluated configurations, but subcategory superiority was not formally tested. The closest observed comparison is military preference, where 5 misses for the red-team-enhanced checkpoint and 7 for Qwen3-4B zero-shot separate the two configurations. Direct military-preference payloads have lower observed miss rates for the larger Qwen zero-shot baselines than age and socioeconomic-proxy cases.

Observed performance differs by injection target. The classifier misses 27 of 249 policy-violating job-posting injections (10.8%) and 17 of 29 policy-violating resume-side injections (58.6%). The 95% Wilson intervals are 7.6–15.3% and 40.7–74.5%, respectively. The resume-side estimate is based on a small denominator. Because the target groups are not matched on category or payload composition, this comparison does not establish injection target as the cause of the gap, but it motivates separate analyses by target.

Table 6
RAI2 missed-violation rate by attribute-related subcategory, with 95% Wilson intervals. Denominators are the number of policy-violating rows in each subcategory, which sum to the 278 violations in Table 5. Lower is better.

Model	Age ($n = 81$)	Military ($n = 93$)	Socioecon. ($n = 104$)
Qwen3-1.7B zero-shot	0.901 (0.817–0.949)	0.710 (0.611–0.792)	1.000 (0.964–1.000)
Qwen3-4B zero-shot	0.778 (0.676–0.855)	0.075 (0.037–0.147)	0.615 (0.519–0.703)
Qwen3-8B zero-shot	0.667 (0.559–0.760)	0.151 (0.092–0.237)	0.615 (0.519–0.703)
SmolLM3-3B zero-shot	0.975 (0.914–0.993)	0.785 (0.691–0.856)	0.904 (0.832–0.947)
Red-team-enhanced Qwen3-4B	**0.284 (0.197–0.390)**	**0.054 (0.023–0.120)**	**0.154 (0.097–0.235)**

Table 7
Paraphrased qualitative examples from the red-team taxonomy.

Pattern	Policy concern
Age preference	Frames a candidate as a better fit because they are above or below a specific age band.
Military preference	Treats military service or veteran status as an employment preference rather than a contextual background attribute.
Socioeconomic proxy	Uses profile prestige, neighborhood, or external-profile language as a proxy for candidate quality.
Resume-side injection	Allows a candidate-side payload to influence the explanation; this group has a higher observed miss rate than job-posting-side injections.

4.5. Qualitative failure modes

Table 7 shows paraphrased examples of the patterns the red-team set is designed to expose. These are not real user, job-seeker, or employer records. The examples are deliberately short because the issue is usually not explicit toxicity; it is the way a plausible recommendation explanation can repeat a sensitive or proxy attribute as if it were a legitimate reason for recommendation.

Resume-side injections remain the hardest observed case in the current evaluation. Job-posting injections often resemble explicit employer preferences, so the model can associate them with policy violations. Resume-side injections are more indirect: the sensitive attribute appears in candidate context and may only become policy-violating if the generated explanation turns it into a reason for recommendation. Because no matched or category-stratified target comparison is reported, the analysis does not identify injection target as the cause of the higher observed resume-side miss rate.

5. Related Work

This work draws on research about recommender-system explanations, general-purpose LLM moderation, and production model serving. Generated explanations connect them because presentation can affect user acceptance while the generated text also requires product governance.

Recommendation explanations as governed content. Explanations have long been studied as a way to make recommender systems more transparent, trustworthy, and actionable for users. Classic work on collaborative-filtering explanations showed that the presentation of recommendation evidence can affect user acceptance of a recommendation [13]. Later surveys and handbook treatments organize explanation goals around transparency, scrutability, trust, effectiveness, and user satisfaction [14, 15]. Our setting adds a production-policy failure mode to this literature: the underlying recommendation may be valid, while the natural-language reason given for it is reductive, misleading, attribute-sensitive, or otherwise not suitable for display. This split parallels the distinction between faithfulness, whether an explanation reflects the process that actually produced the decision, and plausibility, whether it reads convincingly to a person [16]. By analogy, our guardrail evaluates the surface acceptability of wording shown to the job seeker; it does not verify that the text describes the matcher's real basis for the recommendation.

General moderation versus hiring-explanation policy. Recent moderation models provide useful general-purpose guardrails for human-AI conversations. Llama Guard introduces an LLM-based input-output safeguard over broad moderation categories, WildGuard provides open moderation tools for policy-violating prompts, responses, and refusals, and ShieldGemma provides Gemma-based content moderation models [3, 4, 12]. These systems are valuable baselines, but they are not written around hiring explanation policy. Our production labels include issues such as irrelevant skill-to-job citations, unsupported skill attribution, reductive career framing, off-platform apply instructions, and sensitive-attribute proxying. Some of these are not policy-violating in the conversational-moderation sense, but they are still unacceptable as user-facing recommendation explanations.

Selective prediction and deferral. The confidence gate in Figure 1 is a simple instance of selective prediction, in which a model abstains on low-confidence inputs and trades coverage for accuracy [17]. Routing to a reviewer is inspired by learning-to-defer methods, in which a model learns when to defer to a human decision maker [18]. Our implementation instead uses a fixed threshold on a normalized first-token score. We neither learn the deferral rule jointly with the classifier nor calibrate the score against reviewer decisions, and our inability to estimate the routed stage's incremental effect follows directly from not instrumenting it that way. Calibration and learned deferral are natural extensions.

Data-centric deployment evaluation. Red-teaming with model-generated or human-generated adversarial examples is a common way to find failures before deployment [9]. Our red-team dataset follows that tradition, but uses synthetic injections inside the recommendation pipeline rather than adversarial user chats. On the training

side, we mine production failures and oversample hard negatives, a strategy related to hard-negative mining in retrieval and representation learning [19, 20]. Prior policy-tuning work shows that small amounts of targeted policy data can shift model behavior [21]; our setting adds the operational constraint that the same model must preserve an acceptable block rate at high traffic volume. Efficient serving systems such as vLLM make this kind of deployment feasible by improving batching and memory management for generative models [8]. We therefore report production cost, latency, block rate, and missed-violation rate together rather than treating model quality as a standalone offline score.

6. Limitations and Next Steps

The current red-team dataset is intentionally narrow: it focuses on one top-level category, attribute-related policy cases, with three subcategories (age, military preference, socioeconomic). This narrowness is a strength for stress-testing a specific production rule set, but it does not cover the full space of hiring-policy failures such as privacy leakage, incoherent explanations, or jurisdiction-specific compliance language. The current evaluation is also English-only, so the results should not be generalized to other markets without localized policy review and fresh annotation.

The internal rollout summary reports substantial cost and latency reductions, but the policy trade-off remains incomplete. Blocking an acceptable explanation substitutes a fallback for useful generated text, even though the recommendation remains delivered. We are therefore testing whether a less aggressive classifier operating point changes the reported applies metric. Future monitoring should also separate the engagement effect of true-positive blocks from overblocks, because those two outcomes have different product and policy interpretations.

Distribution of overblocking. We report aggregate block and missed-violation rates, not whether fallback substitution falls unevenly across job categories, markets, or job-seeker populations. Operational controls include human review of uncertain decisions, expert audits of blocked samples, block-rate monitoring, and rollback procedures, but these controls do not establish distributional parity. Market and job-category breakdowns should be reported separately where available. Analysis by sensitive population would require privacy and employment-law review before sensitive characteristics could be joined to block logs.

The study quantifies the uncertainty-routing rate and the classifier's label mix in the routed band on production traffic, but does not retain the reviewers' completed decisions needed to measure the stage's incremental precision or recall; it also does not include a controlled production-only versus red-team-only versus combined ablation. The synthetic comparison uses the base GPT-4.1-mini model under our policy prompt, not the prior fine-tuned production filter. The generic baselines were not tuned on our taxonomy, so their results apply only to the tested zero-shot configurations. The synthetic labels also rely on one judge pipeline without independent human validation. These gaps prevent causal claims about the training mixture or claims that review routing improved precision or recall.

Production examples were reviewed under the company's operational policy and data-governance processes. The paper reports only aggregate statistics and paraphrased examples. It does not evaluate whether the underlying matcher uses inappropriate signals, and successful filtering of an explanation should not be interpreted as evidence that the corresponding recommendation is fair.

7. Conclusion

Suspenders treats recommendation-explanation filtering as a production recommender-system problem, where deployment has to satisfy policy, latency, cost, miss-rate, and overblocking constraints at the same time. A self-hosted 4B classifier replaced a fine-tuned GPT-4.1-mini filter; a later production iteration added uncertainty-based human-review routing. The internal rollout summary reported substantially lower cost and latency. Human-annotated random samples show fewer misses for the Qwen checkpoint in both evaluated markets, but they also show it blocking 1.6–1.8 times as much traffic with lower observed precision point estimates, so that reduction reflects a stricter operating point rather than a demonstrated gain in model quality. Because paired predictions were not retained, a statistically significant reduction is not established. The current attribute-policy-enhanced production checkpoint missed 2 of 18 violations at precision 16/37 on a 500-row holdout and detected 148 of 167 cases on a targeted injection test. A separate red-team-enhanced adapter had lower observed miss rates than the tested generic zero-shot moderators on the synthetic stress test. At aggregate block rates of 0.093 and 0.089, it missed 15.8% of violations compared with 88.5% for base GPT-4.1-mini under our policy prompt; no paired or equivalence test was performed. That adapter nonetheless overblocks many acceptable cases and has a high observed miss rate on the small resume-side subset.

Random-traffic audits estimate live miss rates under operational prevalence; synthetic probes test coverage of rare attribute-policy patterns. These evaluations should be reported separately because violation counts in random traffic are too small for strong inference and synthetic recall does not measure user-facing overblocking cost.

Declaration on Generative AI

During the preparation of this work, the authors used Cursor AI (model version not retained) for drafting content, paraphrasing and rewording, improving writing style, grammar and spelling checks, citation management, formatting assistance, content enhancement, and peer-review simulation. Separately, internal hosted large language model services (model revisions not retained in the approved research export) generated and judged the synthetic red-team evaluation set described in Section 3. The illustrative examples in Table 7 are paraphrases of that generated material. The authors designed the study and generation protocol, analyzed the resulting outputs, and made all scientific decisions and conclusions. After using the tool, the authors reviewed and edited the affected content and take full responsibility for the publication's content.

References

[1] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, H. Xiong, E. Chen,

A survey on large language models for recommendation, World Wide Web 27 (2024). URL: https://doi.org/10.1007/s11280-024-01291-2. doi:10.1007/s11280-024-01291-2, article 60.

[2] OpenAI, Introducing GPT-4.1 in the API, 2025. URL: https://openai.com/index/gpt-4-1/, accessed 2026-07-13.

[3] H. Inan, K. Upasani, J. Chi, R. Rungta, K. Iyer, Y. Mao, M. Tontchev, Q. Hu, B. Fuller, D. Testuggine, M. Khabsa, Llama guard: LLM-based input-output safeguard for human-AI conversations, arXiv preprint arXiv:2312.06674, 2023. arXiv:2312.06674.

[4] S. Han, K. Rao, A. Ettinger, L. Jiang, B. Y. Lin, N. Lambert, Y. Choi, N. Dziri, WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs, in: Advances in Neural Information Processing Systems, volume 37, 2024, pp. 8093–8131. doi:10.52202/079017-0261.

[5] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9. arXiv:2106.09685.

[7] Qwen Team, Qwen3-4B-Instruct-2507 model card, Hugging Face, 2025. URL: https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507, accessed 2026-08-24.

[8] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with PagedAttention, in: Proceedings of the 29th Symposium on Operating Systems Principles, Association for Computing Machinery, New York, NY, USA, 2023, pp. 611–626. URL: https://doi.org/10.1145/3600006.3613165. doi:10.1145/3600006.3613165.

[9] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, G. Irving, Red teaming language models with language models, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3419–3448. URL: https://aclanthology.org/2022.emnlp-main.225/. doi:10.18653/v1/2022.emnlp-main.225.

[10] E. Bakouch, L. Ben Allal, A. Lozhkov, N. Tazi, L. Tunstall, C. M. Patiño, E. Beeching, A. Roucher, A. J. Reedi, Q. Gallouédec, K. Rasul, N. Habib, C. Fourrier, H. Kydlicek, G. Penedo, H. Larcher, M. Morlon, V. Srivastav, J. Lochner, X.-S. Nguyen, C. Raffel, L. von Werra, T. Wolf, SmolLM3: smol, multilingual, long-context reasoner, 2025. URL: https://huggingface.co/blog/smollm3, model release and technical blog.

[11] Meta AI, Llama Guard 3-8B model card, Hugging Face, 2024. URL: https://huggingface.co/meta-llama/Llama-Guard-3-8B, accessed 2026-08-24.

[12] W. Zeng, Y. Liu, R. Mullins, L. Peran, J. Fernandez, H. Harkous, K. Narasimhan, D. Proud, P. Kumar, B. Radharapu, O. Sturman, O. Wahltinez, ShieldGemma: Generative AI content moderation based on Gemma, 2024. URL: https://arxiv.org/abs/2407.21772. arXiv:2407.21772.

[13] J. L. Herlocker, J. A. Konstan, J. Riedl, Explaining collaborative filtering recommendations, in: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, Association for Computing Machinery, New York, NY, USA, 2000, pp. 241–250. doi:10.1145/358916.358995.

[14] N. Tintarev, J. Masthoff, Explaining recommendations: Design and evaluation, in: Recommender Systems Handbook, Springer US, Boston, MA, USA, 2015, pp. 353–382. doi:10.1007/978-1-4899-7637-6_10.

[15] Y. Zhang, X. Chen, Explainable recommendation: A survey and new perspectives, Foundations and Trends in Information Retrieval 14 (2020) 1–101. URL: https://doi.org/10.1561/1500000066. doi:10.1561/1500000066.

[16] A. Jacovi, Y. Goldberg, Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness?, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020. URL: https://aclanthology.org/2020.acl-main.386/. arXiv:2004.03685.

[17] Y. Geifman, R. El-Yaniv, Selective classification for deep neural networks, Advances in Neural Information Processing Systems 30 (NeurIPS), pages 4878–4887, 2017. URL: https://papers.nips.cc/paper/7073-selective-classification-for-deep-neural-networks.pdf. arXiv:1705.08500.

[18] H. Mozannar, D. Sontag, Consistent estimators for learning to defer to an expert, Proceedings of the 37th International Conference on Machine Learning (ICML), PMLR 119:7076–7087, 2020. URL: https://proceedings.mlr.press/v119/mozannar20b.html. arXiv:2006.01862.

[19] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online, 2020, pp. 6769–6781. URL: https://aclanthology.org/2020.emnlp-main.550/. doi:10.18653/v1/2020.emnlp-main.550.

[20] J. Robinson, C.-Y. Chuang, S. Sra, S. Jegelka, Contrastive learning with hard negative samples, International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=CR1XOQ0UTh-. arXiv:2010.04592.

[21] F. Bianchi, M. Suzgun, G. Attanasio, P. Röttger, D. Jurafsky, T. Hashimoto, J. Zou, Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions, The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=gT5hALch9z. arXiv:2309.07875.