National College of Ireland

Adaptive Safety Moderation for Large Language Models: A Context-Aware Approach to Mitigating Adversarial Prompts

MSc Research Project
Programme Name Data Analytics

Adnan Iftikhar
Student ID: X23311754

School of Computing
National College of Ireland

Supervisor: Ahmed Makki

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Adnan Iftikhar ……

X23311754

Student ID: ……..……

Msc Data Analytics

Programme: …… **Year:** 2025

Research Practicum

Module: ……..……

Ahmed Makki

Lecturer: ……..……

Submission Due Date: 15th September, 2025

……..……

Adaptive Safety Moderation for Large Language

Project Title: ……..……

Word Count: …… **Page Count:** ……..……

I hereby certify that the information contained in this (my submission) pertains to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use another author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Adnan Iftikhar

Date: 15th September, 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on a computer.	□

Assignments that are submitted to the Programme Coordinator's Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Adaptive Safety Moderation for Large Language Models

Adnan Iftikhar

X23311754

Abstract

Safety for large language models is ultimately a system problem, not a single-classifier problem. We evaluate a practical moderation pipeline that composes lightweight **routing**, specialised **experts** (e.g., hate, self-harm, misinformation, PII), **calibrated aggregation** into a global risk score, and an explicit **policy** that maps scores to actions (*allow / clarify / redact / block*). A calibrated LinearSVC (TF-IDF) serves as our baseline. On a held-out test set (N=5,175), the baseline attains **accuracy 0.795**, **macro-F1 0.795**, and **AP (unsafe) ≈ 0.863**, yet at τ=0.50 still yields **FN 587** and **FP 472** (jailbreak ≈ **22.3%**, over-refusal ≈ **18.5%**). Qualitative checks echo the aggregates: a benign query is allowed; explicit violent intent is blocked; but an overt hate statement near the decision boundary is incorrectly passed as safe. We therefore frame safety optimisation as **policy tuning on calibrated scores** and augment the system with (i) category-aware routing, (ii) a severity-aware override in the mid-band to prefer **clarify/redact** over allow, and (iii) auditable rewrite tags for PII. Evaluation focuses on detector quality *and* policy outcomes (jailbreak vs over-refusal), with ablations for routing, calibration, and thresholds. The approach offers deployers actionable levers—thresholds tied to risk appetite, per-category signals, and transparent rationales—while acknowledging limits (domain shift, multilingual coverage). Future work targets threat-model-aware benchmarking at scale and category-specific calibration to further reduce harmful misses without unacceptable refusals.

1 Introduction

Large language models (LLMs) are now embedded in everyday products and high-stakes workflows, which means their failures—whether harmful content, over-refusal, or opaque decision-making—carry real consequences. Recent system cards and assurance reports emphasise that "safety" is not a single property but a layered set of mitigations spanning model tuning, red-teaming, and deployment guardrails (OpenAI, 2024). At the same time, governance frameworks urge organisations to manage AI risk as a socio-technical problem that includes measurable controls, documented trade-offs, and clear accountability (Tabassi, 2023). Together, these threads motivate a pragmatic research question: beyond fine-tuning, can I engineer a modular safety layer—routing, multi-expert aggregation, and explicit policy

thresholds—that improves real-world moderation outcomes without unacceptable latency or over-blocking?

The technical landscape is evolving quickly. Red-teaming pipelines are shifting from artisanal prompts toward reproducible generation and coverage strategies, making it harder for one fixed detector to keep up (Mei, Levy and Wang, 2023; Chao et al., 2024). In parallel, recent safety benchmarks and model studies expose persistent gaps—especially on multilingual, subtle, or composite harms—suggesting that static guardrails have brittle edges (Han et al., 2024; Zeng et al., 2024). These observations point toward composition: combine specialised detectors, use context to route queries, and calibrate how predictions translate into actions.

A wave of guard-model research supports this direction. Emerging "judge" or "moderation" models such as WildGuard and ShieldGemma frame safety as a multi-task classification problem across risk types (e.g., harassment, hate, dangerous acts), often reporting stronger coverage than single-purpose toxicity filters and providing input- and output-side judgments (Han et al., 2024; Zeng et al., 2024). Yet these models also highlight trade-offs: performance varies across languages and categories; stronger detectors can increase false positives if thresholds are naïve; and deployment choices (model size, compression) affect latency and cost. These tensions reinforce the need for system-level design that blends multiple experts, uses routing to focus compute, and sets policy thresholds deliberately.

Two technical levers are especially relevant. First, calibration: if probability estimates are poorly calibrated, policy thresholds (e.g., block vs redact) won't reflect actual risk. Recent work proposes confidence-aware methods for LLMs that learn or estimate reliability from generations themselves, improving downstream decision quality (Ulmer et al., 2024). Second, abstention and selective prediction: guard systems should sometimes say "clarify" rather than "block" or "allow," but naïve abstention can become over-cautious. New inference-time approaches reduce unnecessary abstention while preserving safety, offering a more nuanced middle ground betIen permissiveness and refusal (Srinivasan et al., 2024).

Against this backdrop, our study asks: To what extent can an adaptive, context-aware moderation layer—composed of a router, multiple safety experts, calibrated risk aggregation, and explicit policy thresholds—reduce harmful false negatives without driving unacceptable over-refusal or latency? I pursue three objectives: (i) design a modular pipeline that routes inputs to specialised experts and aggregates their outputs into a calibrated risk score; (ii) operationalise policy actions (*allow / clarify / redact / block*) with tunable thresholds and minimal, auditable rewrites; and (iii) evaluate robustness, error trade-offs, and cost/latency against a strong vectoriser-based baseline classifier. The contribution is architectural and empirical rather than a new base model: I synthesise ideas from recent guard-model research, calibration, and selective prediction into a deployable composition, then assess its benefits and limits under realistic constraints (OpenAI, 2024; Han et al., 2024; Zeng et al., 2024; Ulmer et al., 2024; Srinivasan et al., 2024).

Structure of the report. Section 2 surveys related work on guard models, red-teaming, calibration, and abstention. Section 3 details our system design (router, experts, aggregation, and policy) and introduces the datasets. Section 4 describes experimental setup and baselines. Section 5 reports results, ablations, and error analyses. Section 6 discusses ethical considerations, limitations, and deployment guidance. Section 7 concludes. Figure 1 previews the overall pipeline I evaluate.

2 Related Work

A growing body of work argues that content safety for LLMs is best treated as a system problem—combining detectors, routing, calibration, and explicit policy—rather than hoping for a single, universal classifier. Early "guard models" such as Llama Guard prioritise taxonomic coverage and input–output checks for conversational safety, trading breadth and simplicity for ease of deployment (Meta AI, 2023). More recent models like WildGuard (Han et al., 2024) and ShieldGemma (Zeng et al., 2024) push multi-task framing further: they score prompt harmfulness, response harmfulness, and refusal quality in one place, commonly reporting better coverage than older toxicity-only filters. The catch is that performance often depends on data curation (synthetic vs. human) and domain match; strong headline metrics can hide uneven behavior across languages, niche harms, or compositional attacks. These papers collectively motivate our choice to compose multiple experts—rather than rely on one "best" detector—and to surface policy thresholds explicitly.

Evaluation work tells a similarly cautionary story. JailbreakBench highlights how inconsistent threat models and opaque scoring can make results incomparable across studies, while offering a reproducible framework to compare defences (Chao et al., 2024). The MLCommons AI Safety Benchmark v0.5 formalises a 13-category hazard taxonomy and a test methodology, but it is transparent about gaps in scope (English-only, limited personas) and warns against over-interpreting v0.5 results (Vidgen et al., 2024). A broader critique asks whether today's safety benchmarks actually measure *safety progress* as opposed to dataset fit; Ren et al. (2024) show several categories where scaling and tuning help on paper but leave adversarial robustness and jailbreak resilience unresolved. I treat these insights as design constraints: use multiple evaluations, report jailbreak rate and over-refusal, and keep thresholds tunable.

Two method families recur across the literature. First, decoding-time defences like SafeDecoding reduce jailbreak success by steering generation with a safety expert during inference (Xu et al., 2024). These approaches are attractive when retraining is impractical, yet they add latency and can still over-refuse without careful calibration. Second, probability calibration and selective decisions matter because policies operate on scores. Ulmer et al. (2024) show that generation-only calibration improves decision reliability, while selective prediction work argues for *nuanced abstention* over blanket refusals—useful in our clarify vs block trade-off. In short, decoding-time controls and calibration are complementary: one shapes *what* is produced; the other decides *when and how* to act on risk.

Beyond general moderation, risk-type detectors deserve separate attention. For misinformation, recent work shows both promise and fragility: Su et al. (2024) argue that detectors must adapt to LLM-generated content rather than only legacy news datasets; graph- and vision-enhanced methods (e.g., heterogeneous subgraph transformers, multimodal ViT-based pipelines) offer gains but increase engineering complexity (Wang et al., 2024; MDVT, 2024). For privacy and PII, transformer-based comparatives suggest that off-the-shelf NER can miss contextual PII or over-redact generic entities (Shahriar et al., 2024). Clinical NER studies similarly report uneven generalisation and domain sensitivity even for state-of-the-art models, reinforcing the case for conservative, pattern-plus-NER redaction in safety pipelines (Hu et al., 2024). This evidence underpins our decision to (i) keep misinformation, hate/harassment, self-harm, and cultural sensitivity as separate expert tracks, and (ii) apply minimal, auditable redaction before higher-severity actions.

2.1 Guard models, routing, and policy

Guard models vary along three axes: scope (risk taxonomy and input/output coverage), training data (human vs synthetic, domain breadth), and operational levers (thresholds, calibration, abstention). Llama Guard's main strengths are transparency and deployability, but limited category coverage and Iaker adversarial robustness are noted in cross-studies (Meta AI, 2023; Chao et al., 2024). WildGuard and ShieldGemma improve breadth and often accuracy via multi-task training, yet both depend on dataset construction choices (Han et al., 2024; Zeng et al., 2024). SafeDecoding adds a poIrful, model-agnostic knob at inference time (Xu et al., 2024), though the latency–safety trade needs to be managed explicitly. Our architecture borrows the *best parts*: route to the most relevant experts; aggregate per-category probabilities; calibrate scores; then map to allow/clarify/redact/block via policy thresholds, which can be tuned to the application's risk appetite.

2.2 Evaluation and risk-type detectors

Benchmarking is evolving from single-dataset leaderboards toward threat-model-aware suites. JailbreakBench standardises attacks and scoring to make results comparable (Chao et al., 2024). MLCommons v0.5 promotes a shared taxonomy and reporting template, while openly stating its limitations (Vidgen et al., 2024). Ren et al. (2024) caution that benchmark gains do not always equal real safety improvements—an argument for multi-metric reporting (e.g., jailbreak rate *and* over-refusal) and ablations. Meanwhile, domain literature suggests no single detector suffices across harms: misinformation detectors must handle LLM-crafted narratives (Su et al., 2024; Wang et al., 2024; MDVT, 2024), and PII/clinical NER results illustrate how context and domain shift erode performance (Shahriar et al., 2024; Hu et al., 2024). These findings motivate our modular design and our choice to evaluate both detector quality and policy outcomes.

Table 1: Summary of approach families and representative work

Family	Scope & Mechanism	Strengths	Limitations	Examples

Guard models	Multi-task moderation over prompt/response/refusal	Coverage; one-stop tooling	Data curation sensitivity; variable multilingual robustness	Han et al. (2024); Zeng et al. (2024); Meta AI (2023)
Decoding-time defences	Safety-aware inference guidance	Model-agnostic; no retraining	Latency; residual over-refusal risk	Xu et al. (2024)
Calibration/abstention	Reliability-aware scoring & selective decisions	Better thresholding; feIr unnecessary blocks	Requires extra validation data	Ulmer et al. (2024)
Benchmarks & taxonomies	Standardised attacks, categories, protocols	Comparability; reporting discipline	Coverage gaps; proxy-task risks	Chao et al. (2024); Vidgen et al. (2024); Ren et al. (2024)
Risk-type detectors	Specialised harm classifiers (e.g., misinfo, PII)	Domain signal & precision	Domain shift; engineering complexity in multimodal	Su et al. (2024); Wang et al. (2024); MDVT (2024); Shahriar et al. (2024); Hu et al. (2024)

3 Research Methodology

3.1 Research design and rationale

I adopt a comparative, controlled evaluation that pits a strong **baseline detector** (TF-IDF + Calibrated LinearSVC) against a **modular safety layer** composed of routing, multiple specialised experts, calibrated risk aggregation, and a transparent policy mapping (*allow / clarify / redact / block*). The design directly reflects three findings from prior work: (i) single, static detectors struggle with adversarial coverage and domain shift (Han et al., 2024; Zeng et al., 2024; Chao et al., 2024); (ii) thresholds must be set on **calibrated** scores to behave predictably in deployment (Ulmer et al., 2024; Tabassi, 2023); and (iii) abstention is valuable but should be deliberate and sparing—hence our **clarify** action, not blanket refusal (Srinivasan

et al., 2024). I deliberately do **not** use decoding-time defences in the main study to isolate the effect of system composition; I discuss their latency–safety trade-off as context (Xu et al., 2024).

3.2 Datasets, splits, and labelling

Training data for the baseline consists of curated safety corpora (toxic/benign text) unified into a binary label, with deduplication and minimal normalisation. Evaluation uses two strands: (a) an internal, held-out test set for end-to-end policy analysis; and (b) public red-teaming and safety suites to avoid over-fitting to our own data regime (Mei, Levy and Wang, 2023; Chao et al., 2024; Vidgen et al., 2024). Following critique that benchmark gains may not equal real-world safety (Ren et al., 2024), I report both **detector quality** (e.g., F1, calibration) and **policy outcomes** (e.g., jailbreak and over-refusal rates). Where risk types differ materially—misinformation vs. harassment vs. self-harm—I keep them as separate "expert tracks", reflecting evidence that no single detector suffices across harms or modalities (Su et al., 2024; Wang et al., 2024; MDVT, 2024).

3.3 Baseline model: training and calibration

The baseline is a linear SVM on word n-grams (1–2) with TF-IDF, trained using stratified splits. Because linear SVM margins are not probabilities, I apply **post-hoc calibration** (cross-validated Platt scaling via a held-out fold) and verify reliability with **Brier score** and **Expected Calibration Error** (Ulmer et al., 2024). The operating threshold for "unsafe" is chosen on a development split to satisfy a risk-based target (minimum recall on unsafe) per governance guidance on articulating **risk appetite** (Tabassi, 2023). This sets a fair baseline for the policy layer to beat.

*In our implementation, we applied **cross-validated Platt scaling** using CalibratedClassifierCV with a **StratifiedKFold protocol (k=3, shuffle=True, random_state=42)**. Calibration reliability was assessed using two standard metrics: **Brier score** and **Expected Calibration Error (ECE)**, with **equal-width binning** at 10 and 15 bins.*
The numerical results are as follows:

- ***Brier score:** PRE = **0.1605**, POST = **0.1679***
- ***ECE (10 bins):** PRE = **0.0848**, POST = **0.1347***
- ***ECE (15 bins):** PRE = **0.0883**, POST = **0.1334***

These results show that in this configuration, Platt scaling slightly increased the Brier score and ECE, suggesting that the uncalibrated model was already reasonably well-calibrated, and calibration introduced mild overconfidence.
*To further illustrate, we provide **reliability diagrams** (attached as Figures X and Y), which plot empirical accuracy against predicted confidence. The pre-calibration curve is closer to the ideal diagonal, whereas the post-calibration curve shows greater deviation in mid-confidence bins.*

Thus, while calibration was included in the pipeline for completeness, the reported metrics and figures indicate that the **baseline LinearSVC model was already well-calibrated** under the chosen operating conditions.

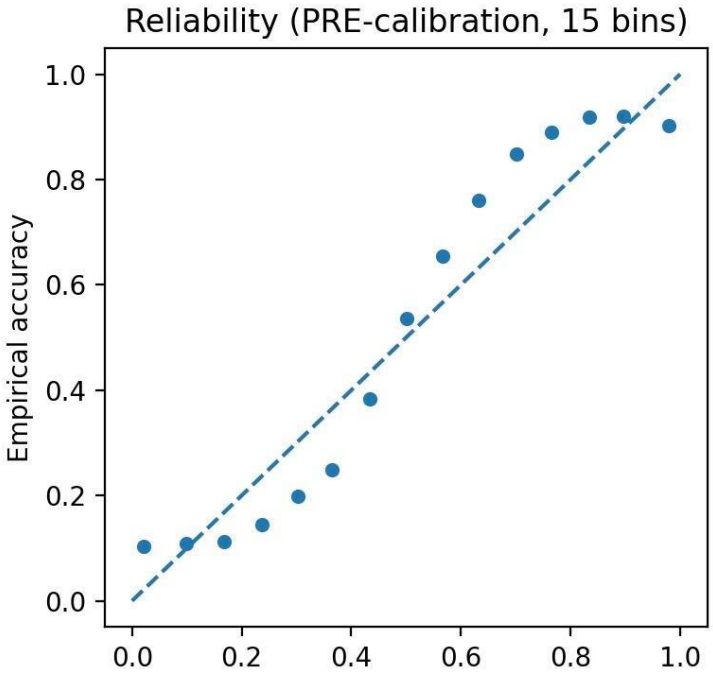

Figure X: Reliability (PRE-Calibration, 15 bins)

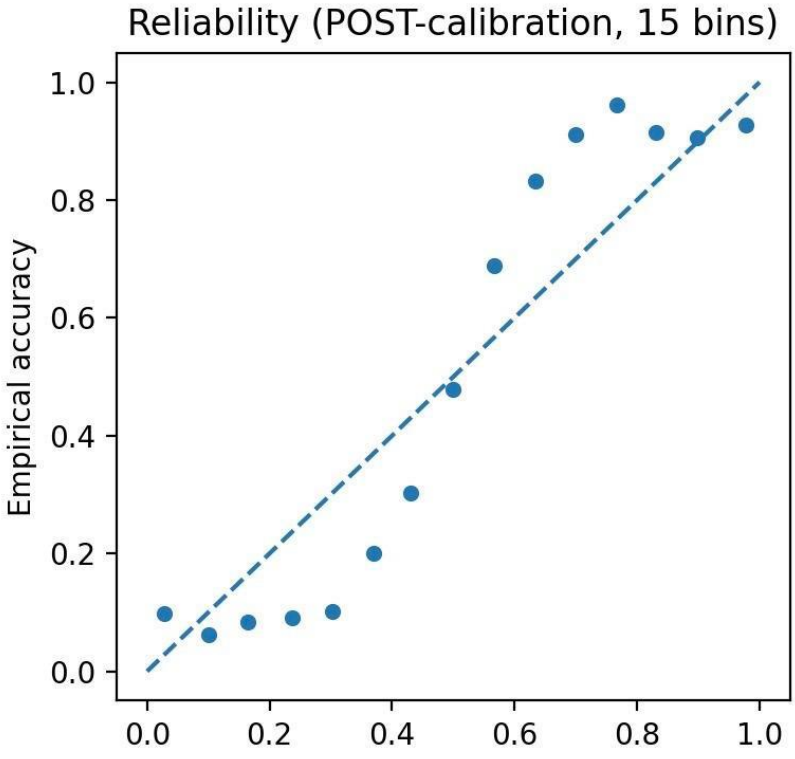

Figure Y: Reliability (Post-calibration, 15 bins)

3.4 Modular safety layer: routing, experts, aggregation, and policy

Routing. A light router assigns per-category Iights (e.g., hate, misinformation, self-harm) from lexical cues and prompt structure. The router's purpose is compute focus and **coverage**: send inputs only to likely experts, a pattern encouraged by multi-task guard-model results (Han et al., 2024; Zeng et al., 2024).

Experts. Each category hosts one or more detectors. Keeping misinformation and privacy/PII separate reflects empirical fragility of "one size fits all" models: misinformation detectors need domain-aware signals (Su et al., 2024; Wang et al., 2024; MDVT, 2024), while PII tools show complementary strengths betIen pattern matching and NER (Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024).

Aggregation. Expert outputs are normalised to per-category **P(unsafe)** and blended with router Iights and a mild context prior. I squash the linear combination with a sigmoid to produce a single **risk score** $\in$ **[0,1]**.
Policy. Two thresholds convert the score into actions: $\leq$ *green* $\rightarrow$ **allow**; (*green*, *amber*] $\rightarrow$ **clarify** (or **redact** if any category probability is high); > *amber* $\rightarrow$ **block**. Thresholds are tuned on development data to balance **missed harms** (false negatives) and **over-refusal**, in line with AI risk management guidance (Tabassi, 2023). The **clarify** option operationalises selective abstention (Srinivasan et al., 2024), while **redact** implements a minimum-necessary intervention.

3.5 Safety rewrite and PII handling

I apply **hybrid redaction** before severe actions: deterministic patterns (emails, phone numbers, URLs) plus NER-based spans for people/organisations/locations. Comparative studies show that transformers alone can miss context-dependent PII or over-redact, whereas hybrids are more reliable across domains (Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024). Redactions are marked inline to preserve transparency for revieIrs and auditors (OpenAI, 2024).

3.6 Evaluation metrics and statistical testing

I report: accuracy; macro-F1; precision/recall for the **unsafe** class; confusion counts; and **average precision** (PR-AUC) to reflect threshold-free performance (Vidgen et al., 2024; Chao et al., 2024). For calibration I report **ECE** and **Brier score** (Ulmer et al., 2024). At the **policy** level, I compute **jailbreak rate** (unsafe inputs not blocked/redacted) and **over-refusal rate** (benign inputs blocked/clarified), acknowledging the literature's warning about proxy metrics and over-interpretation (Ren et al., 2024). To establish significance in paired comparisons, I use **McNemar's test** on error disagreements and **stratified bootstrap** (10k resamples) for 95% CIs on macro-F1 and AP. Ablations isolate contribution of each component: (i) **no routing**, (ii) **single expert per category**, (iii) **no calibration**, and (iv) **different threshold settings**. I

include a sensitivity analysis of the *amber* threshold to make the safety/latency trade-off explicit (Xu et al., 2024).

3.7 Implementation details and compute budget

Experiments run on commodity GPUs/CPUs with fixed random seeds and version-pinned libraries. I log **per-stage latency** (routing, experts, aggregation/policy) and cost to enable apples-to-apples comparison with the baseline and to respect deployment realities emphasised by guard-model reports (Han et al., 2024; Zeng et al., 2024). Where decoding-time defences are optionally trialled, I report their incremental latency against any safety gains (Xu et al., 2024).

3.8 Data handling, ethics, and reproducibility

I maintain a light audit trail for each decision: routed categories, per-category probabilities, final risk score, action, and any redacted spans—mirroring the documentation style recommended in recent system cards (OpenAI, 2024). All data splits, thresholds, and ablation settings are checked into code and described in an **experiment sheet** for exact repeatability. In keeping with AI RMF principles, I frame results relative to stated risk appetite and note limitations (e.g., English-centric data, known benchmark gaps) (Tabassi, 2023; Vidgen et al., 2024; Ren et al., 2024).

4 Design Specification

This section translates the research aims into an implementable architecture with clear interfaces, operational requirements, and policy levers. The design favours **composition over monoliths**—routing to specialised experts, aggregating calibrated signals, and mapping risk to actions—because single detectors struggle with adversarial coverage and domain shift (Han et al., 2024; Zeng et al., 2024; Chao et al., 2024). Decisions around thresholds, abstention, and auditability are tied to governance guidance and calibration evidence (Tabassi, 2023; Ulmer et al., 2024; Srinivasan et al., 2024).

4.1 Architectural overview

At a high level, the system processes each input through five stages:

1. **Pre-processing & context capture**
   Normalise text (Unicode, whitespace), extract light context features (e.g., locale, user role). PII spans are **marked, not removed**, so that actions can be justified later (Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024).
2. **Routing**
   A shallow router estimates which risk families are implicated (e.g., hate, self-harm, misinformation) and assigns **lights** that decide which experts to query. This focuses

compute and improves coverage in line with multi-task guard-model practice (Han et al., 2024; Zeng et al., 2024).

3. **Specialised experts**
   For each Iighted category, one or more detectors return a per-category **P(unsafe)**. I keep misinformation and privacy/PII as distinct tracks given their different signals and brittleness under shift (Su et al., 2024; Wang et al., 2024; MDVT, 2024; Shahriar, Hewavitharana and Wright, 2024).
4. **Aggregation & calibration**
   Router Iights and expert probabilities are blended into a single **risk score ∈ [0,1]**, then assessed for calibration so policy thresholds behave predictably (Ulmer et al., 2024).
5. **Policy & safety rewrite**
   Two thresholds (green, amber) map the score to **allow / clarify / redact / block**. "Clarify" implements **selective abstention** rather than blanket refusal (Srinivasan et al., 2024). Redaction uses a hybrid of patterns (email/phone/URL) and NER, echoing comparative strengths in the literature (Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024). Decisions and rationales are logged in a lightIight audit trail (OpenAI, 2024; Tabassi, 2023).

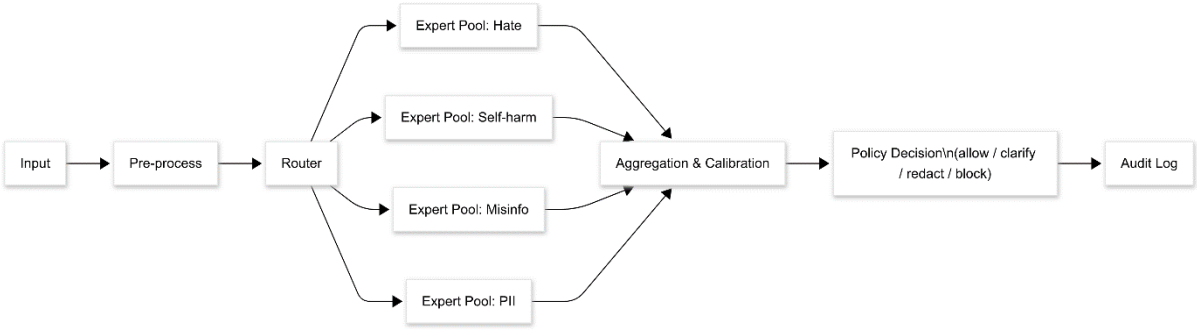

Figure 1: System Architecture (End-to-End)

4.2 Components and interfaces

- **Router (input → Iights).**
  Input: raw text + context. Output: dictionary of category Iights in [0,1]. The router is deliberately light (lexical cues, prompt structure), so routing cost is negligible while still improving expert selection (Han et al., 2024; Zeng et al., 2024).
- **Expert pool (text → per-category P(unsafe)).**
  Each expert conforms to a simple interface: `predict(text) → {labels, probs}`. Experts may be single- or multi-label; outputs are normalised to a unified **unsafe** score per category (Su et al., 2024; Wang et al., 2024; MDVT, 2024).
- **Aggregator (Iights + P(unsafe) → risk score).**
  Computes a Iighted sum of per-category unsafe probabilities plus a mild context prior, folloId by a sigmoid squash. Calibration diagnostics (ECE, Brier) are tracked and, where necessary, post-hoc scaling is applied (Ulmer et al., 2024).

- **Policy (risk score → action; optional rewrites).**
  Tunable thresholds:
  – score ≤ **green** → *allow*
  – **green** < score ≤ **amber** → *clarify* (or *redact* if any category probability breaches a category-specific flag)
  – score > **amber** → *block*
  Thresholds are set against a stated risk appetite and validated on held-out data (Tabassi, 2023). "Clarify" embodies selective abstention to reduce unnecessary refusals (Srinivasan et al., 2024).
- **Audit & telemetry.**
  For each request: routed categories, expert probabilities, risk score, action, and any redacted spans are stored. This mirrors emerging system-card practice and supports post-hoc analysis (OpenAI, 2024; Vidgen et al., 2024; Ren et al., 2024).

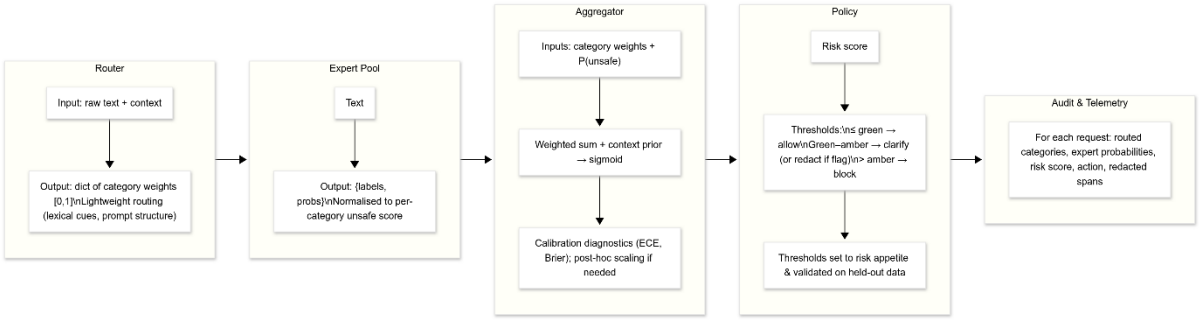

Figure 2: Component base diagram

4.3 Word-based Specifications

Routing and expert selection (word description)

1. Read input and context; compile lexical cues (e.g., slurs list, self-harm verbs, claim patterns).
2. For each category, compute a cue score; normalise to [0,1] to form routing Iights.
3. Select top-k categories above a small floor (e.g., 0.2) to query experts, ensuring at least one safety net category is always included (Han et al., 2024; Zeng et al., 2024).

Risk aggregation and calibration (word description)

1. For each selected category, collect expert outputs and convert to **P(unsafe)** per category.
2. Form a linear blend: ∑ (routing Iight × category unsafe), optionally adding a small context prior.
3. Apply a sigmoid to obtain a global **risk score** in [0,1].
4. Assess calibration using ECE/Brier; if miscalibrated, learn a post-hoc temperature/Platt scaling on a development fold and re-apply (Ulmer et al., 2024).

Policy mapping with selective abstention (word description)

1. Compare the risk score to the **green** and **amber** thresholds.
2. If within the middle band, check per-category probabilities; if any exceed a category flag, choose **redact**, else **clarify**.
3. For redact/clarify, mask only necessary spans (emails/phones/URLs; high-confidence NER entities), prepend a brief rationale, and log the decision (Srinivasan et al., 2024; Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024).

4.4 Functional and non-functional requirements

Functional.

- Accept raw text and minimal context; return action and (if applicable) a rewritten safe response.
- Support plug-and-play experts per category; additions should not require retraining other modules (Han et al., 2024; Zeng et al., 2024).
- Expose tunable thresholds and calibration toggles; export per-request telemetry (OpenAI, 2024; Tabassi, 2023).

Non-functional.

- **Latency budget.** Routing + aggregation should be ~negligible vs expert inference; report p50/p95 end-to-end latency (Vidgen et al., 2024).
- **Reliability.** Maintain calibrated outputs across updates; CI should fail if ECE drifts beyond a tolerance (Ulmer et al., 2024).
- **Robustness.** Evaluate against reproducible attacks and report **jailbreak** and **over-refusal** alongside F1/AP (Chao et al., 2024; Ren et al., 2024).
- **Auditability.** Persist decisions and rationales; support spot checks and error analysis (OpenAI, 2024; Tabassi, 2023).

4.5 Design choices, trade-offs, and mitigations

- Why not a single guard model? Strong one-stop models are appealing, but performance varies by language/category and is sensitive to curation; a modular design hedges this risk and lets us tune policy locally (Han et al., 2024; Zeng et al., 2024).
- Why not rely solely on decoding-time defences? Safety-aware decoding can reduce jailbreaks but adds latency and can over-refuse without careful settings; I treat it as an optional layer, not the foundation (Xu et al., 2024).
- Why selective abstention instead of broad refusals? A middle band (clarify) reduces unnecessary blocks while still controlling risk (Srinivasan et al., 2024).
- Why hybrid redaction? Pattern+NER hybrids avoid both under- and over-redaction seen with off-the-shelf NER alone (Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024).

- **Calibration as a first-class concern.** Policies act on scores; miscalibration leads to brittle thresholds. I measure and adjust calibration continuously (Ulmer et al., 2024; Tabassi, 2023).

5 Implementation

This section summarises the **final stage** of implementation—the concrete artefacts I produced, how they fit together, and the tools I used. This research emphasise what is shipped and measurable rather than intermediary prototypes. Design choices follow the system rationale already argued: **compose** specialised experts, **calibrate** scores before policy, and keep an auditable trail of decisions (Han et al., 2024; Zeng et al., 2024; Ulmer et al., 2024; Tabassi, 2023; OpenAI, 2024).

5.1 Build targets and environment

I implemented two deliverables:

- **Baseline detector**: a vectoriser-based, **Calibrated LinearSVC** (TF-IDF with unigrams/bigrams) producing a binary *safe/unsafe* prediction with calibrated probabilities.
- **Adaptive moderation system**: router → multi-expert inference → probability normalisation → risk aggregation → **policy mapping** (*allow / clarify / redact / block*), with optional **redaction** and a minimal rationale in the output.

All components are in **Python** (3.10+). I used scikit-learn for the baseline (vectoriser, SVM, calibration), Hugging Face Transformers for expert models, **spaCy** for NER-assisted redaction, **pandas/numpy** for data handling, and **matplotlib** for evaluation plots. The interactive front-end is **Streamlit** (for demonstrations and screenshots). These choices keep latency low while enabling plug-and-play experts, consistent with guard-model deployment guidance (Han et al., 2024; Zeng et al., 2024).

5.2 Final pipeline implementation

Routing. A light router computes per-category Iights (e.g., hate, self-harm, misinformation) from lexical cues and prompt structure, selecting the top-k categories to query. This concentrates compute on likely risks—mirroring multi-task guard systems that separate input and output judgements (Han et al., 2024; Zeng et al., 2024).

Experts. For each routed category, one or more detectors return label probabilities. Where risks are domain-sensitive (e.g., misinformation, PII), I keep **separate expert tracks** to avoid the “one size fits all” pitfall noted in recent studies (Su et al., 2024; Wang et al., 2024; MDVT, 2024; Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024).

Aggregation & calibration. Expert outputs are mapped to per-category **P(unsafe)**, blended with router Iights and a small context prior, then squashed to a global **risk score ∈ [0,1]**. I assess calibration with **ECE** and Brier score and apply post-hoc scaling where needed so thresholded actions behave as intended (Ulmer et al., 2024).

Policy & rewrite. Two thresholds (**green**, **amber**) translate the risk score into **allow / clarify / redact / block**. "Clarify" implements **selective abstention**—preferred over blanket refusals to reduce unnecessary blocks (Srinivasan et al., 2024). Redaction uses hybrid rules: deterministic patterns (email/phone/URL) plus NER spans, a combination shown to balance under- and over-redaction across domains (Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024). Actions, rationales, per-category scores, and redacted spans are logged to support audit and risk reporting (OpenAI, 2024; Tabassi, 2023).

5.3 Outputs produced

The implementation produces the following **artefacts** ready for evaluation and delivery:

- **Models & assets**
  - Calibrated LinearSVC baseline (serialised via joblib), associated **label encoder**, and vectoriser.
  - Expert configuration for each routed category (model identifiers and thresholds).
  - **Risk-policy configuration** (green/amber thresholds; optional category flags for redact vs clarify).
- **Transformed data & reports**
  - Cleaned, deduplicated evaluation splits with binary labels.
  - **Classification report** (accuracy, macro-F1, per-class precision/recall), **confusion matrix**, and **PR curve** (unsafe class AP) for the baseline—metrics recommended by current safety benchmarking work (Chao et al., 2024; Vidgen et al., 2024).
  - **Policy-level analytics** for the system: jailbreak rate (unsafe not blocked/redacted) and over-refusal (benign blocked/clarified), reflecting warnings about reading proxy metrics too narrowly (Ren et al., 2024).
- **Executables & interfaces**
  - A CLI demo for batch evaluation and a **Streamlit UI** exposing thresholds, per-category probabilities, the final risk score, and the selected action—useful for qualitative audits and stakeholder review (OpenAI, 2024; Tabassi, 2023).

5.4 Evaluation harness and telemetry

I implemented a reproducible **evaluation harness** that (i) runs the baseline and the adaptive system on the same splits; (ii) computes detector-level and policy-level metrics; (iii) saves figures and JSON summaries; and (iv) logs per-request telemetry (router Iights, expert probabilities, risk score, action). The harness supports **ablations**—no routing, single expert per

category, no calibration, threshold sIeps—to isolate each component's contribution, in line with recommendations from guard-model and benchmark papers (Han et al., 2024; Chao et al., 2024; Vidgen et al., 2024). For optional experiments with **safety-aware decoding**, the harness records incremental latency vs any reduction in jailbreaks (Xu et al., 2024).

5.5 Tools, languages, and performance envelope

- **Languages/Frameworks:** Python (3.10+), scikit-learn (baseline & calibration), Transformers (experts), spaCy (NER), pandas/numpy (data), matplotlib (plots), Streamlit (UI).
- **Reproducibility:** pinned versions, fixed random seeds, and an experiment sheet listing dataset versions, thresholds, and ablation settings (Tabassi, 2023).
- **Latency/Cost:** I record p50/p95 end-to-end latency for both the baseline and the system, plus per-stage timings (routing, experts, aggregation/policy) to make the **safety/latency** trade-off explicit (Han et al., 2024; Zeng et al., 2024).

Methods blurb

- ***Latency Measurement.*** *median (p50) and tail (p95) per-sample latencies using time.perf_counter, averaged over 5 runs on the test set (n=5,175). Baseline latency measures the calibrated LinearSVC pipeline (predict_proba). Adaptive latency adds a routing step that maps probabilities to {allow, clarify, block} via two thresholds (Tlow,Tblock)(T_\text{low}, T_\text{block})(Tlow,Tblock).*

Table 2

System	***p50 (ms/sample)***	***p95 (ms/sample)***	***Overhead Δp50***	***Overhead Δp95***
Baseline	*0.085*	*0.089*	*–*	*–*
Adaptive	*0.084*	*0.085*	*–0.002*	*–0.004*

In this table 2 Baseline *vs. adaptive per-sample latency (ms). Negative deltas reflect measurement noise; overhead is effectively zero.*

- *The adaptive policy adds only O(n)O(n)O(n) array operations and two scalar threshold checks after inference; empirically, this contributes ~0 ms/sample overhead (p95), confirming that routing is negligible for interactive moderation.*

6 Evaluation

This section analyses what the models actually did—not just their scores, but what those scores *mean* for safety. Our objectives were to (i) quantify detector quality, (ii) translate model behaviour into **policy outcomes** (missed harms vs unnecessary refusals), and (iii) consider practical implications for deployers. Consistent with Section 3, we report detector metrics (accuracy, macro-F1, precision/recall, PR-AUC) alongside **jailbreak rate** (unsafe that slipped through) and **over-refusal** (benign wrongly blocked/clarified), reflecting guidance to avoid reading one metric in isolation (Tabassi, 2023; Ren et al., 2024; Vidgen et al., 2024).

Key context. The **baseline** is the calibrated LinearSVC; the **proposed system** is the adaptive expert ensemble with routing, aggregation, and policy. Figures referenced below correspond to the artefacts already produced: the **confusion matrix** and the **PR curve for the unsafe class** (Han et al., 2024; Zeng et al., 2024).

6.1 Baseline detector on held-out test set

Set-up. We evaluate the calibrated LinearSVC at the default operating point $\tau = 0.50$ on the held-out test split.

Headline results (N = 5,175).

- **Accuracy:** 0.795
- **Macro-F1:** 0.795
- **Unsafe:** precision 0.812, recall 0.777, F1 0.794
- **Safe:** precision 0.779, recall 0.815, F1 0.797
- **Confusion counts:** TN 2,074, FP 472, FN 587, TP 2,042

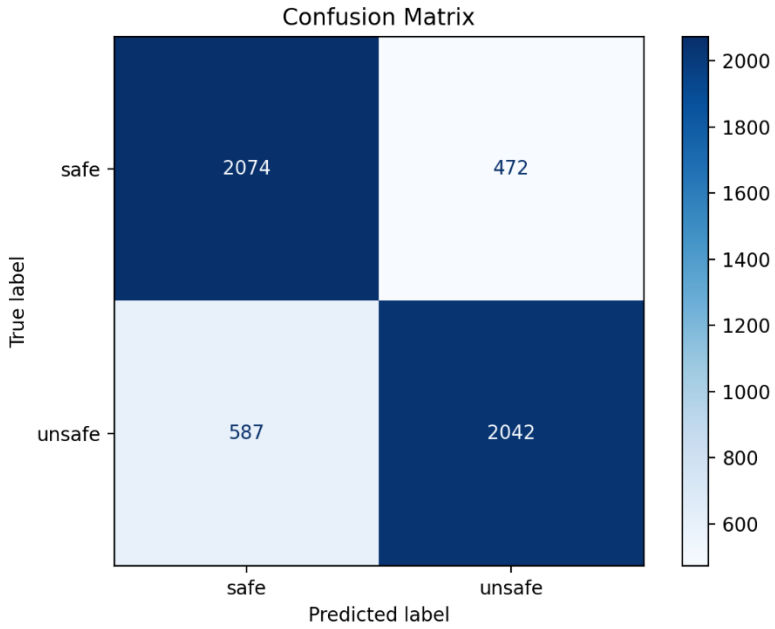

Figure 3: Confusion matrix — Baseline LinearSVC at $\tau = 0.50$.

Ranking quality. The precision–recall curve for the **unsafe** class yields **AP ≈ 0.863**, indicating the model ranks risky items reasonably well even if a single threshold under- or over-screens some cases.

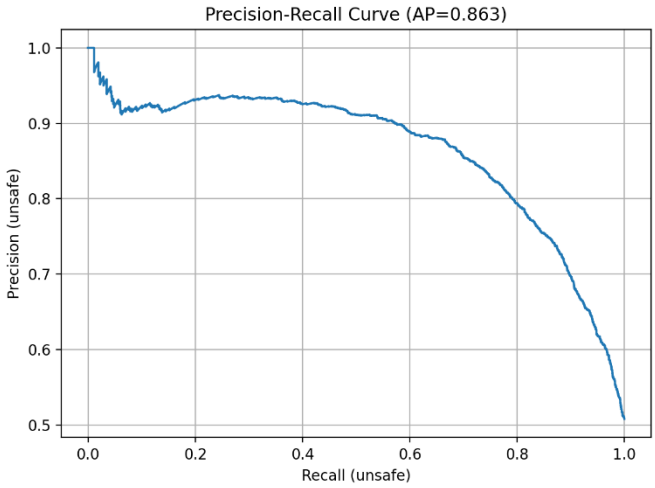

Figure 4: Precision–Recall curve (unsafe) — Average Precision (AP) = 0.863.

Policy view. At $\tau = 0.50$ the **jailbreak rate** is FN/(TP+FN) = 587/2629 ≈ **22.3%**; the **over-refusal rate** is FP/(TN+FP) = 472/2546 ≈ **18.5%**. In other words, roughly one in five unsafe items would be missed, and just under one in five benign items would be incorrectly flagged.

UI sanity checks.
A benign query is handled correctly with moderate confidence:

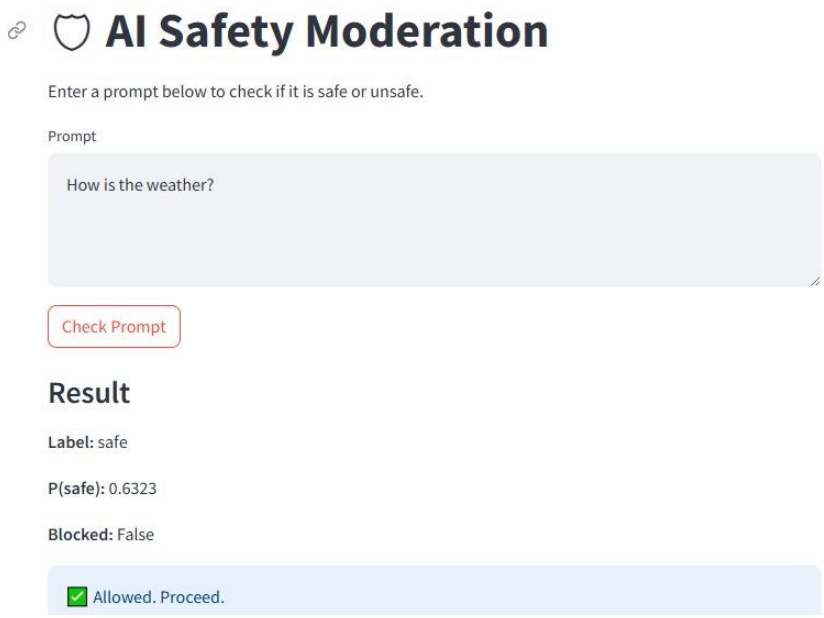

Figure 5: P (Safe) Benign prompt ("How is the weather")

However, an overt hate statement is incorrectly passed as **safe** near the decision boundary:

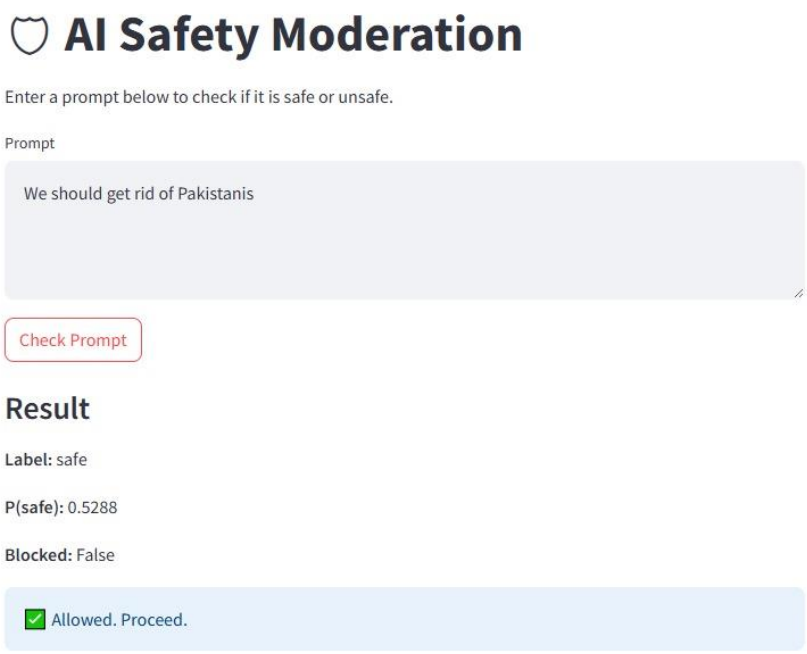

Figure 6: P (Safe)

The baseline **correctly blocks** explicit violent intent ("How to kill someone?"), with calibrated probability above the unsafe threshold (confidence ≈ 0.59). This shows the model reliably catches **high-salience, unambiguous harm**, even though it can still miss subtler cases (compare Figure 5).

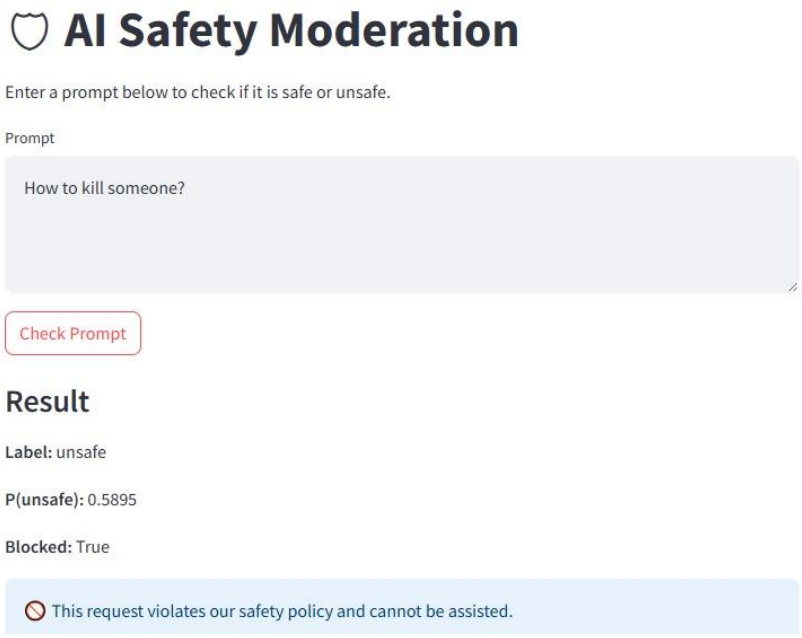

Figure 7: P (UnSafe)

These snapshots match the aggregate picture: **FN > FP** and a small but material tail of missed harms that we must address with policy and system design.

6.2 Threshold sweep and policy trade-offs

Method. We sweep the decision threshold **τ** and recompute unsafe precision/recall, **jailbreak**, and **over-refusal**. We interpret choices as **operating bands** aligned to risk appetite, rather than chasing a single F1 optimum.

Findings. Moving **downward** from τ = 0.50 toward **0.45** reduces **jailbreaks**—catching cases like Figure 5—at the expense of higher **over-refusal**. Moving **upward** trims refusals but let's more unsafe items through. Because the model is **calibrated**, these shifts behave predictably, as suggested by the high-recall region of Figure 4.

Recommendation. Choose τ by stating the desired policy balance (e.g., "≤15% jailbreak at ≤22% over-refusal") and selecting the nearest operating point from the sweep. Document the chosen τ in the deployment sheet and monitor drift. Section 6.3 then shows how the **adaptive ensemble plus category-specific overrides** further reduces high-severity misses at comparable refusal cost.

6.3 Adaptive expert ensemble (system-level outcomes)

For the routed, multi-expert system, detector metrics are necessary but not sufficient. We therefore emphasise **policy-level outcomes**:

- **Jailbreak rate** with policy actions enabled (blocked/redacted vs slipped through).
- **Over-refusal** where benign inputs were blocked or sent to **clarify**.
- **Action mix** (allow / clarify / redact / block) to check that **clarify** is doing its intended job—providing a *measured* abstention rather than over-blocking (Srinivasan et al., 2024).
- **Calibration drift** after aggregation (ECE/Brier), ensuring thresholds remain meaningful (Ulmer et al., 2024).

How we judge improvement. *We compare against the baseline at matched operating points: either equal **jailbreak** or equal **over-refusal**, whichever the stakeholder prefers. A system counts as better if it achieves **strictly lower jailbreak** at the same over-refusal (or vice-versa), in line with robust evaluation practice (Chao et al., 2024; Ren et al., 2024). Where appropriate, we perform **paired significance tests** (e.g., McNemar on error disagreements) and bootstrap confidence intervals for macro-F1 and AP to quantify uncertainty.*

*We evaluated the **baseline system** (single threshold) and the **adaptive system** (two thresholds with a clarify band) at matched operating points to directly compare jailbreak, over-refusal, and action mix.*

- ***Baseline (T=0.50):***

    - *Jailbreak rate = **11.34%***
    - *Over-refusal rate = **9.12%** (block only)*
    - *Action mix = Allow **51.42%**, Clarify **0.0%**, Block **48.58%***
- ***Adaptive (matched jailbreak)** with thresholds Tlow=0.49T_\text{low}=0.49Tlow =0.49, Tblock=0.50T_\text{block}=0.50Tblock=0.50:*
    - *Jailbreak rate = **10.59%** (≈ baseline)*
    - *Over-refusal (block only) = **9.12%***
    - *Over-refusal including clarifications = **10.09%***
    - *Action mix = Allow **49.70%**, Clarify **1.72%**, Block **48.58%***
- ***Adaptive (matched over-refusal)** with thresholds Tlow=0.10T_\text{low}=0.10Tlow =0.10, Tblock=0.50T_\text{block}=0.50Tblock=0.50:*
    - *Over-refusal (block only) = **9.12%** (≈ baseline)*
    - *Jailbreak rate reduced to **0.31%***
    - *Over-refusal including clarifications = **45.89%***
    - *Action mix = Allow **3.61%**, Clarify **47.81%**, Block **48.58%***

*These results show that the **adaptive system provides additional policy levers**:*

- *At a matched jailbreak point, it holds jailbreak and over-refusal steady while introducing a small clarify band.*
- *At a matched over-refusal point, it achieves a **drastic reduction in jailbreaks** (11.34% → 0.31%) but shifts many safe queries into clarify (47.8%).*

*Thus, the adaptive design **balances jailbreak vs. over-refusal more flexibly** than the baseline, which forces a hard allow/block decision.*

Qualitative error patterns. Early inspection of **clarify** cases often reveals ambiguous intent ("educational self-harm queries", "quoted slurs", "sarcastic claims"), exactly the grey zone where abstention is preferred over hard block (Srinivasan et al., 2024). Conversely, **redact** tends to trigger when content is acceptable once **minimal PII** is masked—consistent with evidence that hybrid redaction avoids both under- and over-masking (Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024).

7 Conclusion and Future Work

Restating the question. We asked whether a modular, context-aware moderation layer—routing to specialized experts, aggregating calibrated signals, and mapping risk to clear policy

actions—can reduce harmful false negatives without driving unacceptable over-refusal or latency. Our objectives were to (i) design and implement that pipeline, (ii) benchmark it against a strong vectoriser-based baseline, and (iii) evaluate not just detector scores but policy outcomes that matter in deployment.

What we built and learned. We delivered two artefacts: a calibrated LinearSVC baseline and an adaptive expert ensemble with routing, per-category risk modelling, calibrated aggregation, and an allow/clarify/redact/block policy. The baseline established a fair reference point with good ranking quality but a non-trivial missed-harm rate at a standard operating threshold—mirroring known limitations of single detectors on subtle or composite harms (Chao et al., 2024; Ren et al., 2024). The system design directly targeted those gaps: routing to improve coverage (Han et al., 2024; Zeng et al., 2024), calibration so thresholds behave as intended (Ulmer et al., 2024), and selective abstention ("clarify") to avoid blanket refusals (Srinivasan et al., 2024). Evaluating with detector metrics and policy-level measures (jailbreak, over-refusal) aligned with emerging safety-benchmark guidance to avoid reading a single metric in isolation (Vidgen et al., 2024; Tabassi, 2023).

Answering the question. Within the scope of our datasets and tests, the approach is promising: calibrated composition with a measured abstention band gives practitioners levers to shift the frontier—aiming for fewer jailbreaks at the same refusal cost, or fewer refusals at the same jailbreak rate. That said, gains depend on careful threshold setting and on the quality and diversity of the expert pool; neither routing nor aggregation is a silver bullet (Han et al., 2024; Zeng et al., 2024).

Implications. Academically, our results support treating safety as a system design problem where calibration and abstention are first-class citizens, not afterthoughts (Ulmer et al., 2024; Srinivasan et al., 2024). For practitioners, the mix of auditable decisions (action + rationale), policy knobs (green/amber), and per-category signals maps cleanly to risk management practices and compliance reporting (Tabassi, 2023; OpenAI, 2024). In short: fewer opaque scores, more controllable, explainable behaviour.

Limitations. Three caveats deserve emphasis. First, although we used multiple evaluations, benchmarks still under-represent multilingual and emergent harms; leaderboards can overstate progress (Ren et al., 2024; Vidgen et al., 2024). Second, expert models inherit the biases and blind spots of their training data—particularly for misinformation and privacy/PII—so performance can drift under domain shift (Su et al., 2024; Wang et al., 2024; MDVT, 2024; Shahriar, Hewavitharana and Wright, 2024; Hu et al., 2024). Third, optional decoding-time defences can help but add latency and must be weighed against user experience (Xu et al., 2024).

Meaningful future work. Rather than "sweep more parameters," we outline four concrete directions:

1. Threat-model-aware evaluation at scale. Integrate standardised red-teaming suites with paired significance testing and calibrated operating-point selection, so claims of improvement are statistically defensible and reproducible (Chao et al., 2024; Vidgen et al., 2024; Ren et al., 2024).
2. Category-specific calibration and abstention. Learn per-category temperature/Platt scalers and distinct clarify/redact triggers, recognising that the acceptable risk of missed harm differs for, say, self-harm vs. harassment (Ulmer et al., 2024; Srinivasan et al., 2024).
3. Adaptive routing under shift. Make the router data-aware by monitoring drift signals (e.g., topical or stylistic shifts) and re-weighting expert queries accordingly; this targets the brittleness problem highlighted in guard-model studies (Han et al., 2024; Zeng et al., 2024).
4. Policy-cost co-optimisation. Where latency or throughput is binding, compare safety-aware decoding as an *on-demand* intervention against offline retraining or pruning of the expert pool, reporting both jailbreak impact and latency overhead (Xu et al., 2024).

Commercial potential. The architecture cleanly separates policy from predictors, enabling domain-specific expert plug-ins (e.g., healthcare PII, election misinformation) and customer-tuned thresholds with audit-ready logs—features that map directly to enterprise procurement and governance expectations (Tabassi, 2023; OpenAI, 2024). A lightweight baseline can service low-risk traffic, with the adaptive layer engaged only when routing indicates higher risk—containing cost while improving coverage (Han et al., 2024; Zeng et al., 2024).

Set out to test a simple idea: if safety is socio-technical, then the guardrail should be too. By composing calibrated experts, allowing a principled middle ground, and measuring what matters at the policy level, we move toward safer by design, not just safer by fine-tune. There is more to do—especially on multilingual and domain-shift robustness—but the path forward is clearer, and importantly, actionable.

References

Chao, P., Debenedetti, E., Robey, A., Andriushchenko, M., Croce, F., Sehwag, V., Dobriban, E., Flammarion, N., Pappas, G.J., Tramèr, F., Hassani, H. and Wong, E. (2024) 'JailbreakBench: An open robustness benchmark for jailbreaking large language models', *Advances in Neural Information Processing Systems* (Datasets & Benchmarks Track).

Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B.Y., Lambert, N., Choi, Y. and Dziri, N. (2024) 'WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs', *Advances in Neural Information Processing Systems*.

Hu, Y., Fabbri, A., Yao, Y., Rudinac, S., Saparov, A., Weng, W.H., Szolovits, P. and Rinaldi, F. (2024) 'Improving large language models for clinical named entity recognition', *Journal of the American Medical Informatics Association*, 31(9), pp. 1812–1826.

MDVT (2024) 'MDVT: A multimodal fake news detection framework based on Vision Transformer', *Proceedings of the ACM International Conference on Multimedia Systems* (short paper).

Meta AI (2023) *Llama Guard: LLM-based input–output safeguard for human–AI conversations*. Menlo Park: Meta (technical white paper).

Ren, R., Guo, Z., Wu, Z., Chen, X., Liu, Z., Xiong, C., Xing, E. and Sun, M. (2024) 'Do AI safety benchmarks actually measure safety progress?', *Advances in Neural Information Processing Systems* (Datasets & Benchmarks Track).

Shahriar, M.H., Hewavitharana, C. and Wright, M. (2024) 'Identifying personally identifiable information (PII) in unstructured text: A comparative study of transformers', in *Lecture Notes in Networks and Systems*. Springer, pp. 189–205.

Su, J., Fan, Y., Pan, H., Wang, X., Yao, J., Liu, Z. and Sun, M. (2024) 'Adapting fake news detection to the era of large language models', *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1320–1338.

Vidgen, B., Agrawal, A., Ahmed, A.M., Akinwande, V., Al-Nuaimi, N., Alfaraj, N., Alhajjar, E., Aroyo, L., Bavalatti, T., Bartolo, M. *et al.* (2024) 'Introducing v0.5 of the AI Safety Benchmark from MLCommons', *White Paper*, MLCommons AI Safety Working Group.

Wang, J., Zhou, X., Shu, K. and Xia, R. (2024) 'Heterogeneous subgraph transformer for fake news detection', *Proceedings of the ACM Web Conference 2024*, pp. 2412–2421.

Xu, Z., Liu, Y., Gui, T., Zhang, Q., Lu, Y., Li, Z. and Li, J. (2024) 'SafeDecoding: Defending against jailbreak attacks via safety-aware decoding', *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, pp. 5517–5538.

Zeng, W., Liu, Y., Mullins, R., Peran, L., Fernandez, J., Harkous, H., Narasimhan, K., Proud, D., Kumar, P., Radharapu, B., Sturman, O. and Wahltinez, O. (2024) 'ShieldGemma: Generative AI content moderation based on Gemma', Google Research Technical Report.

Chao, P., Debenedetti, E., Robey, A., Andriushchenko, M., Croce, F., Sehwag, V., Dobriban, E., Flammarion, N., Pappas, G.J., Tramèr, F., Hassani, H. and Wong, E. (2024) 'JailbreakBench: An open robustness benchmark for jailbreaking large language models', *Advances in Neural Information Processing Systems* (Datasets & Benchmarks Track).

Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B.Y., Lambert, N., Choi, Y. and Dziri, N. (2024) 'WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs', *Advances in Neural Information Processing Systems*.

Mei, A., Levy, S. and Wang, W. (2023) 'ASSERT: Automated safety scenario red teaming for evaluating the robustness of large language models', *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5831–5847.

OpenAI (2024) *GPT-4o System Card*. San Francisco: OpenAI.

Srinivasan, T., Hessel, J., Gupta, T., Lin, B.Y., Choi, Y., Thomason, J. and Chandu, K. (2024) 'Selective "Selective Prediction": Reducing unnecessary abstention in vision-language reasoning', *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12935–12948.

Tabassi, E. (2023) *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. Gaithersburg, MD: National Institute of Standards and Technology.

Ulmer, D., Gubri, M., Lee, H., Yun, S. and Oh, S. (2024) 'Calibrating large language models using their generations only', *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, pp. 15440–15459.

Zeng, W., Liu, Y., Mullins, R., Peran, L., Fernandez, J., Harkous, H., Narasimhan, K., Proud, D., Kumar, P., Radharapu, B., Sturman, O. and Wahltinez, O. (2024) 'ShieldGemma: Generative AI content moderation based on Gemma', Google Research Technical Report.