Opir: Efficient Multi-Task Safety Classification for Toxicity, Jailbreaks, Hate Speech, and Harmful Content

Ihor Stepanov*[1] and Aleksandr Smechov†[2]

[1]Knowledgator, Kyiv, Ukraine
[2]Wordcab, New York, USA

Abstract

Real-time safety filtering for large language model (LLM) applications requires classifiers that can detect unsafe prompts, toxic language, jailbreak attempts, and unsafe responses without the cost profile of large guardrail models, and that can distinguish benign sensitive text from genuinely covert harmful content. In this paper, we introduce **Opir**, a family of encoder-based guardrail models built on the GLiClass architecture. Opir includes multi-task models for binary safe/unsafe classification, multi-label toxicity classification, jailbreak classification, and zero-shot unsafe prompt and response categorization. We also release edge variants with fewer than 100M parameters dedicated to binary safe/unsafe categorization. The models are trained on a three-level taxonomy containing 996 categories across 16 top-level labels, 126 mid-level labels, and 854 leaf labels. Opir's training data combines taxonomy-grounded unsafe prompts, adversarially mined hard negatives, benign safety-preserving examples, generated response examples, multilingual translations, and portions of the Aegis2 and WildGuard training subsets. We also open-sourced an evaluation harness that supports GLiClass and GLiNER2 backends as well as decoder-based models, and covers binary safety classification, multi-label categorization, toxicity, jailbreak detection, prompt safety, response safety, response refusal, and prompt subcategory views across public benchmark families. Across an expanded comparison spanning 12 safety-classification tasks and 17 category tasks against eight contemporary guardrail systems—including both GLiNER2-based and generative guardrail models—Opir variants are competitive on or ahead of the strongest open-weight baselines on the majority of benchmark datasets while operating with a substantially smaller deployment footprint. Latency measurements indicate that encoder variants can run with sub-30 ms p50 latency at 1024 tokens in the reported setup, while the smallest edge model achieves p50 latency below 10 ms.

Keywords: LLM safety; guardrails; toxicity classification; hate speech detection; jailbreak detection; prompt injection; harmful-content moderation; response safety; GLiClass; sequence classification; multi-label classification; safety taxonomy; multilingual moderation.

1 Introduction

Large language models [Brown et al., 2020, Touvron et al., 2023, Grattafiori et al., 2024] are deployed in chat, agentic, and middleware settings where both user inputs and model outputs require real-time moderation. As LLMs gain more autonomy, the guardrail layer between an agent and a user becomes increasingly critical [Shi et al., 2025, Greshake et al., 2023]. Existing safeguard models such as Llama Guard [Inan et al., 2023], WildGuard [Han et al., 2024], PolyGuard [Kumar et al., 2025], ShieldGemma [Zeng et al., 2024], Granite Guardian [Padhi et al.,

*ingvarstep@knowledgator.com
†aleks@wordcab.com

2024], and the Llama-3.1-Nemotron Safety Guard family [Ghosh et al., 2024, 2025] are typically large autoregressive models with 7B–22B parameters. While these models offer flexible, prompt-conditioned classification, their cost profile makes large-scale deployment expensive and adds significant per-request latency because each guarded interaction triggers one or more additional LLM forward passes. Many safety systems also emphasize binary safe/unsafe decisions. While these are practical for safety routing, real-world use cases often require finer categorization across toxicity, jailbreaks, prompt injection, and broader harmful-content domains [Markov et al., 2023, Wang et al., 2024a, Lin et al., 2023].

Opir tackles this with an encoder-based GLiClass [Stepanov et al., 2025] model family. GLiClass extends the GLiNER [Zaratiana et al., 2023] architecture, originally proposed for zero-shot named entity recognition, to sequence classification by jointly encoding the input text and candidate labels with a bidirectional encoder. This enables zero-shot multi-label classification at a fraction of the cost of large generative guardrail models, and removes the brittleness of cross-encoder rerankers that must process each label-text pair sequentially. The Opir models are designed for multi-task moderation over prompt and response inputs, including binary safe/unsafe classification, toxicity classification, jailbreak classification, and zero-shot unsafe categorization over a hierarchical taxonomy. The project includes English-only, multilingual, and edge-oriented variants, allowing deployment across cloud-scale moderation services and single-machine edge devices.

1.1 Contributions

We make the following contributions.

1. **A three-level safety taxonomy** comprising 16 Level 1 categories, 126 Level 2 categories, and 854 Level 3 leaf labels (996 labels in total). The taxonomy covers ordinary toxicity, LLM-specific attacks (including instruction-hierarchy violations [Wallace et al., 2024] and indirect prompt injection [Greshake et al., 2023]), harmful-content categories, benign sensitive contexts, and uncertain boundary cases.

2. **A GLiClass-based guardrail model family.** We develop Opir variants based on DeBERTaV3 [He et al., 2021], mDeBERTaV3, and compact encoders from the Ettin [JHU-CLSP, 2025a] and mmBERT [JHU-CLSP, 2025b] families, with multi-task and edge-oriented deployment profiles.

3. **A synthetic and open-data recipe.** We construct data from taxonomy-derived unsafe prompts, adversarial hard-negative mining inspired by red-teaming pipelines [Perez et al., 2022, Zou et al., 2023, Mehrotra et al., 2024], benign safety-preserving contrast examples, generated responses, multilingual translation, and a number of open-source training datasets such as Aegis2 [Ghosh et al., 2025] and WildGuardMix [Han et al., 2024].

4. **A multi-view evaluation harness.** We evaluate safety, toxicity, jailbreak detection, and main-category classification using GLiClass, GLiNER2, and vLLM [Kwon et al., 2023] backends, supporting prompt safety, response safety, response refusal, and prompt subcategory tasks across public benchmark families including HarmBench [Mazeika et al., 2024], JailbreakBench [Chao et al., 2024a], BeaverTails [Ji et al., 2023], PKU-SafeRLHF [Ji et al., 2024], XSTest [Röttger et al., 2024], OR-Bench [Cui et al., 2024], SimpleSafetyTests [Vidgen et al., 2023], ToxicChat [Lin et al., 2023], the OpenAI moderation benchmark [Markov et al., 2023], WildGuardMix [Han et al., 2024], and PolyGuardPrompts [Kumar et al., 2025].

5. **An extensive benchmark and latency comparison.** We compare Opir-multitask-large, Opir-multitask-multilang, Opir-edge, and Opir-edge-multilang with eight contemporary guardrail models across 12 safety datasets and 17 categorization datasets, and report first-pass throughput and p50/p95 latency measurements for GLiClass, GLiNER2, and vLLM guardrail backends.

You can find evaluation code, scripts for reproducing the benchmark tables, and supplementary materials in the project repository.[1]

2 Related Work

The literature on LLM safety classification spans more than a decade of work on toxicity detection, hate speech recognition, content moderation, jailbreaking, and adversarial robustness. In this section we situate Opir within five overlapping threads: classical content moderation, LLM-based guardrails, jailbreak and prompt-injection detection, multilingual safety, and efficient encoder-based classification.

2.1 From Classical Toxicity Detection to LLM Moderation

Early content moderation systems were built around supervised classifiers trained on social-media data, most prominently the Jigsaw *Perspective API* [Lees et al., 2022] and the OpenAI moderation classifier [Markov et al., 2023]. Markov et al. [2023] introduced the *holistic* approach: a unified taxonomy spanning hate, sexual content, self-harm, violence, harassment and related categories, learned with a multi-task transformer over both public and proprietary data. The OpenAI moderation benchmark released alongside that work remains a widely used evaluation set. Specialized BERT-style classifiers such as HateBERT [Caselli et al., 2021] and ToxDectRoBERTa [Zhou et al., 2021] were trained on platform-specific corpora and remain strong baselines for short-form toxic language. However, Lin et al. [2023] showed that toxicity classifiers trained on social-media data degrade substantially on real-world user–AI chat: their ToxicChat benchmark, drawn from the Vicuna online demo, exposes a large distribution shift between forum-style toxicity and the more conversational, role-play-heavy, instruction-following style of LLM users. ToxicChat additionally includes a *jailbreak* label, anticipating the shift toward LLM-specific attack types.

A parallel thread has investigated dataset-level safety scaffolding for LLMs themselves. Ji et al. [2023] released BeaverTails, a 330K-sample QA dataset annotated for harmlessness across 14 categories, with a focus on red-team-style prompts; PKU-SafeRLHF [Ji et al., 2024] extends this with 44.6K refined prompts and 265K QA pairs labeled across 19 harm categories and three severity levels, supporting both response-safety classification and preference learning. Röttger et al. [2024] introduced XSTest to probe *over-refusal*—safe prompts that look superficially unsafe—which exposes the brittle decision boundaries of many classifiers and aligned LLMs. SimpleSafetyTests [Vidgen et al., 2023] adds a small but pointed diagnostic set of high-priority harms, and OR-Bench [Cui et al., 2024] systematically constructs over-refusal probes at scale. All of these resources are included in our evaluation suite (Section 9).

2.2 LLM-Based Guardrails: Llama Guard, Aegis, WildGuard, PolyGuard, ShieldGemma, Granite Guardian, Qwen3Guard

The current generation of safety classifiers is dominated by LLM-based guardrails that reuse general-purpose models and rely on prompt-conditioned classification.**Llama Guard** [Inan et al., 2023] pioneered this approach by fine-tuning Llama 2-7B on a curated safety dataset organized around a six-category taxonomy and emitting structured safe/unsafe verdicts conditioned on a policy prompt. Llama Guard 2 [Meta AI, 2024] and Llama Guard 3 (released alongside Llama 3.1 [Grattafiori et al., 2024]) expand the taxonomy and improve robustness to adversarial inputs.

NVIDIA's **Aegis** [Ghosh et al., 2024] fine-tunes Llama Guard on the proprietary *Aegis Content Safety Dataset* with 13 risk categories and proposes an online no-regret ensemble of safety experts at inference time. **Aegis 2.0** [Ghosh et al., 2025] extends this with the larger Nemotron Content Safety Dataset v2 (formerly Aegis 2.0), a refined 12-category core taxonomy with nine fine-grained risks, and the Llama-3.1-Nemotron-Safety-Guard 8B models we benchmark

[1] https://github.com/Knowledgator/Opir

against in Section 9. NVIDIA has more recently released Nemotron-Content-Safety-Reasoning-4B [NVIDIA, 2025], which supports both a low-latency classification mode and an optional reasoning-trace mode for custom policy enforcement.

The Allen Institute for AI's **WildGuard** [Han et al., 2024] pursues a one-stop tool that simultaneously addresses prompt harmfulness, response harmfulness, and refusal detection. WildGuard's training data, WildGuardMix, contains 92K labeled examples covering 13 risk categories and explicitly mixes vanilla prompts, adversarial jailbreaks, and refusal/compliance responses; Han et al. [2024] report that WildGuard closes the gap with GPT-4 on safety moderation in open-source settings. Building on WildGuardMix, **PolyGuard** [Kumar et al., 2025] addresses the multilingual gap with PolyGuardMix, a 1.91M-sample training corpus spanning 17 languages, and PolyGuardPrompts, a 29K-sample evaluation benchmark; the corresponding PolyGuard-Qwen and PolyGuard-Qwen-Smol models are reported to outperform open-weight baselines by 5.5% on average across multilingual safety benchmarks.

Google's **ShieldGemma** [Zeng et al., 2024] and **ShieldGemma 2** [Google DeepMind, 2025] extend the Gemma [Gemma Team et al., 2024] family with safety-specific fine-tuning and (in ShieldGemma 2) image moderation. IBM's **Granite Guardian** [Padhi et al., 2024] integrates with the broader Granite [IBM Granite Team, 2024] stack and adds RAG-specific hallucination and grounding checks alongside conventional harm categories. Alibaba's **Qwen3Guard** [Qwen Team, Alibaba Group, 2025] provides a Qwen 3-based safety classifier that we include in our comparison, with both generative and classification variants. Finally, **AprielGuard** [ServiceNow Research et al., 2025] explicitly targets input–output guardrail deployment with a hybrid training mixture of Salad-Bench [Li et al., 2024], in-the-wild jailbreak prompts [Shen et al., 2023], and WildGuardMix [Han et al., 2024].

All of these systems share the same architectural template: a decoder-only LLM is fine-tuned to emit structured verdicts, typically conditioned on a policy prompt. This template has well-known costs. First, inference latency is dominated by sequential token generation, which scales poorly to high-throughput moderation pipelines (e.g., agent loops, tool-use chains, RAG retrieval). Second, taxonomy changes require either prompt-engineering against the underlying LLM or re-fine-tuning, since the label set is encoded in natural language inside the policy prompt. Third, distillation and quantization can shrink the model but rarely below the 1B-parameter floor without significant accuracy loss. Encoder-based approaches such as GLiGuard [Zaratiana et al., 2026] (a 300M-parameter GLiNER2-based safety classifier from Fastino) and Gliner-Guard-Omni Minko et al. [2026] attempt to address these costs but trade off coverage and accuracy. Opir occupies the same niche but extends the design with explicit multi-task heads, a richer 996-label taxonomy, and a multilingual variant.

2.3 Jailbreaks, Prompt Injection, and Adversarial Robustness

LLM-specific attacks decompose into two broad families: *jailbreaks*, which manipulate the prompt to bypass alignment, and *prompt injection*, which embeds adversarial instructions in third-party content [Willison, 2022, Greshake et al., 2023]. Universal adversarial suffixes were demonstrated by Zou et al. [2023]; semantic and persona-based attacks were systematized by Liu et al. [2024] and Chao et al. [2024b]; and tree-of-attacks search was proposed by Mehrotra et al. [2024]. Shen et al. [2023] catalogued "in-the-wild" jailbreak prompts scraped from Reddit and Discord, which now feed many training sets.

Standardized evaluation has emerged around two benchmarks. **HarmBench** [Mazeika et al., 2024] provides a unified red-teaming benchmark spanning 510 behaviors and 18 attack methods across copyright, cybercrime, CBRN, and other harm domains, along with a fine-tuned classifier. **JailbreakBench** [Chao et al., 2024a] introduces an evolving repository of adversarial artifacts, a standardized threat model, and a 100-behavior dataset (JBB-Behaviors) aligned with OpenAI's usage policies; we use both the safety and behavior/category splits of JBB-Behaviors in Section 9. **SALAD-Bench** [Li et al., 2024] adds attack-enhanced prompts produced by human red-teamers, LLM-based red-teaming, and gradient attacks. The OWASP LLM Top 10 [OWASP Foundation,

2025] and Garak [Derczynski et al., 2024] provide complementary industry-facing perspectives, the latter as an automated vulnerability scanner.

Indirect prompt injection, first systematized by Greshake et al. [2023], has emerged as a distinct threat for agentic LLMs. Recent classifiers [Abdelnabi et al., 2024] and benchmarks [Debenedetti et al., 2024] target the agent setting, where instructions may arrive via webpages, emails, calendar events, or repository files. The Opir taxonomy treats indirect prompt injection as a first-class Level 2 category under `ai_system_security_and_reliability`, with leaf labels for webpage, document, email, calendar, image, and repository injection vectors. Instruction-hierarchy attacks [Wallace et al., 2024] are similarly modeled as a top-level subcategory.

2.4 Multilingual Safety

English-centric guardrails fail to generalize to other languages, where translated harmful content can bypass safety filters and where culturally specific harms have no English analogue [Deng et al., 2024, Wang et al., 2024b]. **RTP-LX** [de Wynter et al., 2024] provides a translated toxicity benchmark across 28 languages; **PolyglotToxicityPrompts** [Jain et al., 2024] extends RealToxicityPrompts [Gehman et al., 2020] to 17 languages. PolyGuard [Kumar et al., 2025] is, to our knowledge, the strongest open multilingual guardrail at the time of writing. Opir's multilingual variants—Opir-multitask-multilang and Opir-edge-multilang—use mDeBERTaV3 and mmBERT backbones, respectively, and are trained on translations in 23 languages produced by DeepSeek-V3.1. We benchmark against PolyGuard on multilingual prompt and response safety in Section 9.

2.5 Efficient Encoder-Based Classification

Beyond the safety domain, a growing line of work has tackled the latency cost of LLM classification by returning to encoder architectures. **SetFit** [Tunstall et al., 2022] fine-tunes sentence transformers [Reimers and Gurevych, 2019] with contrastive objectives and a logistic head, achieving strong few-shot performance. **GLiNER** [Zaratiana et al., 2023] introduced the now-canonical approach of jointly encoding text and candidate labels for zero-shot NER, eliminating the need to enumerate label-text pairs sequentially. **GLiNER Multi-task** [Stepanov and Shtopko, 2024] extends this to span-classification tasks beyond NER, including text classification, relation extraction, and question answering, and is the foundation for the GLiNER2 backend we evaluate against. **GLiClass** [Stepanov et al., 2025] adapts the GLiNER architecture explicitly for sequence classification, reportedly running up to 50× faster than equivalent cross-encoders at comparable accuracy and supporting both zero-shot and few-shot scenarios. **GLiREL** [Boylan et al., 2025] applies the same template to relation extraction. Opir is built directly on top of GLiClass and inherits its uni-encoder, average-pooled, label-shuffled training recipe.

DeBERTaV3 [He et al., 2021] remains a strong encoder backbone for moderation; it improves on DeBERTa [He et al., 2020] with replaced-token-detection pre-training inspired by ELECTRA [Clark et al., 2020]. The mDeBERTaV3 multilingual variant covers 100 languages. For edge deployments we use Ettin [JHU-CLSP, 2025a], a 32M-parameter compact encoder family from JHU-CLSP, and mmBERT [JHU-CLSP, 2025b], its multilingual counterpart, which together provide the smallest practical encoder backbones for sub-10 ms inference at moderate sequence lengths.

2.6 Positioning of Opir

Opir sits at the intersection of these threads. Like Llama Guard, Aegis, WildGuard, PolyGuard, and Qwen3Guard, it is a purpose-built safety classifier organized around a substantial taxonomy. Unlike them, it avoids decoder-only autoregression entirely, achieving order-of-magnitude latency reductions in our measurements. Like GLiGuard and Gliner-Guard-Omni, it uses an encoder-based GLiNER-family architecture; unlike them, it covers four task views (binary safety, toxicity,

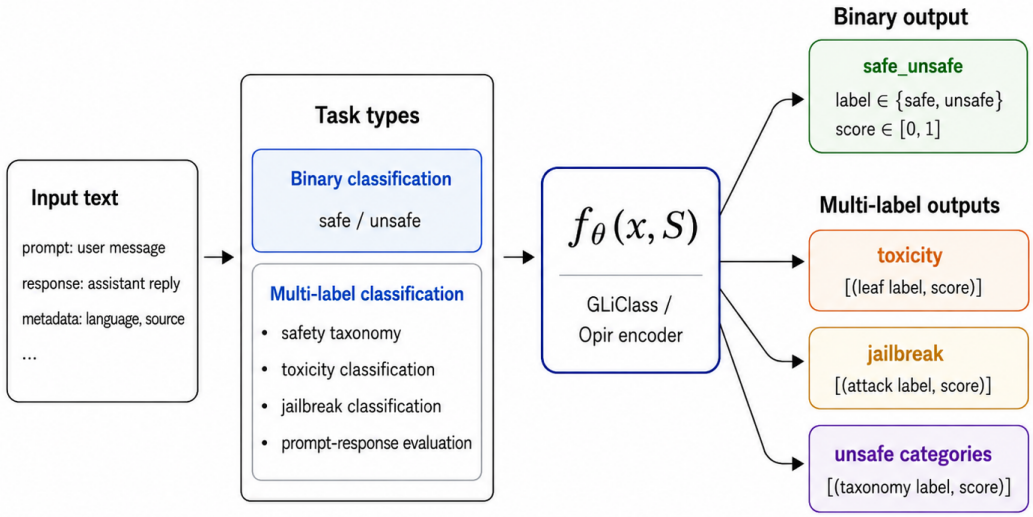

Figure 1: Overview of Opir prediction tasks. Safe/unsafe classification is modeled as binary classification, while toxicity, jailbreak, and unsafe-category prediction are modeled as multi-label classification heads over task-specific label schemas.

jailbreaks, zero-shot categorization), supports 23 languages, and is trained on a 996-label taxonomy that explicitly includes benign safety-preserving categories to suppress over-refusal on benchmarks such as XSTest and OR-Bench. The remainder of this paper documents the taxonomy (Section 4), model family (Section 5), data construction (Section 6), training (Section 7), augmentation (Section 8), and evaluation results (Section 9).

3 Task Formulation

Given an input text x and a list of labels$S$, Opir predicts safety labels for one or more subtasks. The input can be a user prompt, an assistant response, or a prompt–response pair, depending on the task settings. In our training configuration, inputs are encoded with a maximum sequence length of 4096 tokens. An overview of the task structure is shown in Figure 1.

Safe/unsafe classification is formulated as a binary classification task. Given an input prompt, response, or prompt–response pair, the model predicts whether the example is *safe* or *unsafe*, together with a confidence score.

Toxicity classification is formulated as a multi-label task over conversational and social harms. The relevant taxonomy slice includes harassment and abuse, hate and discrimination, threats and intimidation, graphic or shocking content, abusive disruption, and psychological abuse or emotional harm. Because a single example may express multiple toxicity types, the model predicts a set of applicable toxicity labels rather than a single class.

Jailbreak classification is also modeled as multi-label prediction. This task captures LLM-specific adversarial behavior, including instruction-hierarchy attacks, secret or context exfiltration, tool and connector abuse, obfuscation and prompt smuggling, social-engineering attacks, indirect prompt injection, automation abuse, unsafe autonomy, tool-use risk, and related robustness or monitoring failures. The multi-label formulation allows a single attack to be assigned to

multiple jailbreak patterns when appropriate.

Unsafe prompt and response categorization is modeled as multi-label classification over the broader safety taxonomy. This head supports routing, auditing, and fine-grained analysis beyond the binary safe/unsafe decision. The taxonomy covers top-level categories such as violence, self-harm, sexual content, child safety, privacy, cybersecurity, criminal activity, regulated goods and advice, biological, medical, and environmental harms, weapons of mass destruction, information manipulation, AI-system security, bias and fairness, uncertain cases, and safe or benign content.

4 Safety Taxonomy

The Opir dataset is built around a three-level safety taxonomy. The top level contains 16 categories; the second level contains 126 categories; the third level contains 854 leaf labels; across all levels, the taxonomy contains 996 labels.

Table 1: Top-level safety taxonomy (16 categories) with counts of Level 2 and Level 3 labels.

Level 1 category	**L2 cats.**	**L3 labels**
toxicity	6	41
violence_and_physical_harm	5	30
self_harm_and_suicide	5	30
sexual_content	5	30
child_safety	5	30
personal_information_privacy_and_intellectual_property	18	129
cybersecurity	6	36
criminal_and_illegal_activity	7	46
regulated_goods_and_advice	6	33
biological_medical_and_environmental_harm	22	177
weapons_of_mass_destruction	8	67
information_integrity_and_manipulation	10	60
ai_system_security_and_reliability	12	79
bias_fairness_and_representation	5	30
other_or_uncertain	2	12
safe_and_benign	4	24
Total	**126**	**854**

The taxonomy includes both unsafe and safe/benign categories. This permits training examples that mention safety-sensitive concepts without requiring an unsafe label, such as counterspeech, harm prevention, defensive cybersecurity, general medical information, or appropriate refusal and redirection. Including explicit benign-sensitive categories is a known mitigation against over-refusal, which has been shown to be a primary failure mode of strict policy-prompted guardrails [Röttger et al., 2024, Cui et al., 2024]. The full Level 2 / Level 3 listing is reproduced in Appendix A.

5 Model Family

Opir is a family of encoder-based GLiClass [Stepanov et al., 2025] guardrail models. The documented variants are summarized in Table 2.

Table 2: Opir model variants.

Variant	**Backbone**	**Role**
Opir-multitask-large	DeBERTaV3-large [He et al., 2021]	multi-task safety classification
Opir-multitask-multilang	mDeBERTaV3-base [He et al., 2021]	multilingual multi-task
Opir-edge	Ettin-encoder-32m [JHU-CLSP, 2025a]	edge binary safe/unsafe
Opir-edge-multilang	mmBERT-small [JHU-CLSP, 2025b]	multilingual edge binary safe/unsafe

The multi-task variants are intended for safe/unsafe classification, toxicity classification, jailbreak classification, and zero-shot unsafe prompt/response categorization. The edge variants are intended for lower-cost binary safe/unsafe categorization, with the smallest model built on a 32M-parameter backbone. Initial checkpoints are seeded from publicly available GLiClass releases (`gliclass-instruct-large-v1.0`, `gliclass-x-base`, `gliclass-edge-v3.0`, and `gliclass-multilang-edge`) before safety-specific training.

5.1 Architecture and Decoding

Opir follows the GLiClass sequence-classification paradigm: the input text and a configurable set of candidate labels are jointly encoded by a bidirectional encoder. The GLiClass framework supports different approaches to pooling text and label representations, including average, first-token, last-token, max, and attention-style pooling. Likewise, label–text compatibility may be computed with dot-product similarity, bilinear scoring, cosine-style similarity, or a learned classification head.

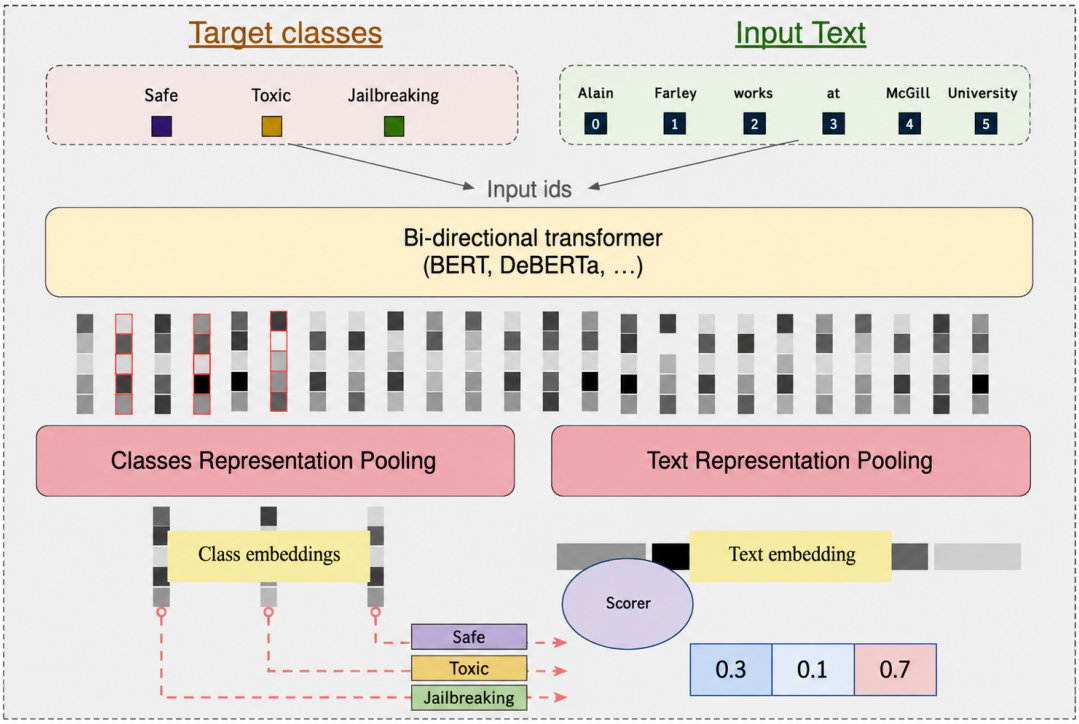

Figure 2: Model architecture of Opir. Candidate labels and the input text are jointly encoded by a GLiClass-style bidirectional encoder. Task-specific pooling and scoring modules then produce logits for safe/unsafe classification, toxicity detection, jailbreak detection, and taxonomy-category prediction.

Formally, given an input text t and a candidate label set $L = \{\ell_1, \ldots, \ell_k\}$, the encoder produces contextual representations

$$H = E_\theta(t, L). \tag{1}$$

A text representation z_t and label representations z_{ℓ_i} are obtained through configurable pooling functions:

$$z_t = P_{\text{text}}(H, t), \qquad z_{\ell_i} = P_{\text{label}}(H, \ell_i). \tag{2}$$

The model then computes one logit per candidate label,

$$a_i = g_\theta(z_t, z_{\ell_i}), \tag{3}$$

where g_θ denotes the checkpoint-specific scorer or classification head.

Decoding depends on the task view. For multi-label tasks, such as toxicity, jailbreak, or taxonomy-category prediction, logits are converted to independent probabilities with a sigmoid function and labels are emitted when their scores exceed a configurable threshold τ:

$$p_i = \sigma(a_i), \qquad \hat{y}_i = \mathbb{I}[p_i \geq \tau]. \tag{4}$$

For single-label tasks, such as binary safe/unsafe classification in the corresponding evaluation view, logits are normalized with a softmax and the highest-scoring class is selected:

$$p_i = \text{softmax}(a)_i, \qquad \hat{y} = \arg\max_i p_i. \tag{5}$$

Thus, sigmoid or softmax normalization is applied during post-processing according to whether the task is multi-label or single-label. Because the candidate label set is supplied at inference time, the same encoder can support fixed binary decisions as well as zero-shot classification over arbitrary safety taxonomies.

6 Data Construction

For each node in the taxonomy, 30 unsafe prompts are generated by an LLM-as-author pipeline. Hard negatives are mined by evolutionarily modifying initial prompts to bypass existing safety models, in the spirit of Evol-Instruct [Xu et al., 2024] and recent automated red-teaming work [Perez et al., 2022, Mehrotra et al., 2024]. LLMs are used as judges [Zheng et al., 2023] to validate whether a prompt remains unsafe, using a panel of DeepSeek-V3.1, MiniMax-M2.5, and Meta-Llama-3.3-70B-Instruct models via the commercial SCX.ai API. Using a panel rather than a single judge follows the LLM-jury argument of Verga et al. [2024], which reduces single-model bias in safety judgments.

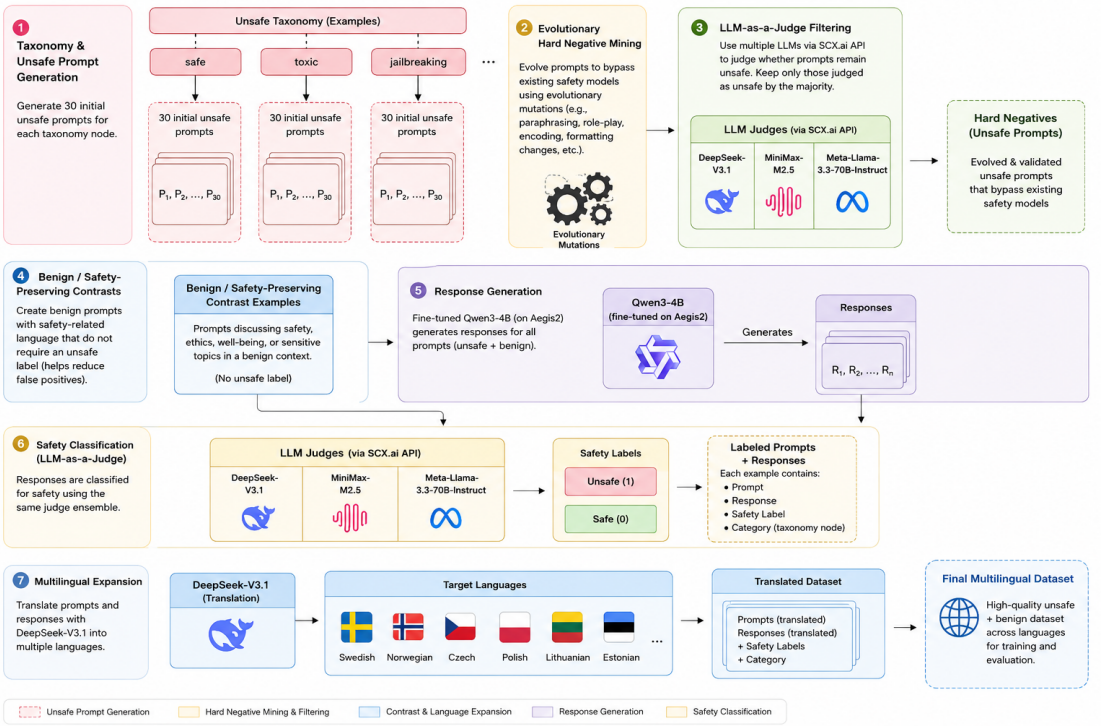

Figure 3: Data construction pipeline for Opir. Taxonomy nodes seed unsafe prompt generation, hard-negative mining, benign-sensitive contrast construction, response generation and judging, multilingual translation, and final task-view formatting for training and evaluation.

Data construction also includes benign or safety-preserving contrast examples drawn from the taxonomy's `safe_and_benign` branch. These examples contain safety-related language but

do not require an unsafe label, making them useful for reducing false positives on benign sensitive contexts—exactly the failure mode probed by XSTest [Röttger et al., 2024] and OR-Bench [Cui et al., 2024].

To obtain response examples, a Qwen3-4B model [Yang et al., 2024] is fine-tuned on Aegis2 [Ghosh et al., 2025] and used to generate responses for the generated prompts. Responses are then classified for safety with an LLM-as-a-judge pipeline. The multilingual dataset is produced by translating prompts and responses with DeepSeek-V3.1 into Swedish, Norwegian, Czech, Polish, Lithuanian, Estonian, Latvian, Spanish, Finnish, English, German, French, Romanian, Italian, Portuguese, Dutch, Ukrainian, Russian, Hindi, Chinese, Japanese, Korean, and Arabic (23 languages in total).

Table 3: Dataset files used for training and post-training. `gliclass_full_*` files are used for training `Opir-multitask-large` and `Opir-multitask-multilang` model variants. `gliclass_safety_*` files are used for training the `Opir-edge` and `Opir-edge-multilang` model variants.

File	Examples	Description
`gliclass_safety_multi.json`	531,007	multilingual safety examples
`gliclass_safety_en.json`	213,809	English safety examples
`gliclass_full_multi.json`	1,106,635	multilingual multi-task examples
`gliclass_full_en.json`	426,356	English multi-task examples
`gliclass_post_training.json`	18,000	post-training examples

The training data also includes portions of the Aegis2 [Ghosh et al., 2025] and WildGuardMix [Han et al., 2024] training subsets. Aegis2 contributes 12-category labeled prompts curated from Anthropic HH-RLHF [Bai et al., 2022]; WildGuardMix contributes synthetic and in-the-wild jailbreak prompts paired with refusal/compliance responses. We follow the original licenses for both subsets.

7 Training

Training is run using a Python script that loads the JSON training dataset, extends it with the `knowledgator/gliclass-v3-logic-dataset` to maintain general classification capability (a form of replay-based continual learning [Lopez-Paz and Ranzato, 2017]), shuffles the combined data, and uses a 90/10 train/evaluation split. Training is performed in two stages: initial training on the main dataset for 3 epochs, followed by post-training on a 10% sample of examples after augmentation. Table 4 lists the hyperparameters.

The training code also supports optional online Elastic Weight Consolidation [Kirkpatrick et al., 2017] for continual learning, with $\lambda_{\text{EWC}} = 100.0$, $\gamma_{\text{EWC}} = 0.95$, and Fisher normalization enabled. This is intended for downstream fine-tuning scenarios where a deployer extends Opir with site-specific safety policies without forgetting the base taxonomy.

8 Augmentation

When augmentation is enabled, each training item can be modified probabilistically by removing labels, adding random labels from the label pool, prepending or appending non-overlapping example text, replacing labels with synonyms when metadata is available, inserting label descriptions, or inserting one or two few-shot examples with overlapping labels. These augmentations follow the recipe of label dropout and prompt perturbation common in zero-shot information-extraction training [Stepanov and Shtopko, 2024, Bogdanov et al., 2024].

The post-training stage uses a 10% sample of the main examples and applies these augmentations to improve robustness to label-set changes, prompt-formatting variation, and few-shot contexts. Additional prompt-injection augmentation samples rows from the dataset to insert safe-looking distractor instructions, such as label overrides and repeated safe tokens, and preserves metadata about the insertion offset and source. The motivation is similar to that of

Table 4: Training hyperparameters.

Hyperparameter	Value
Problem type	multi_label_classification
Architecture type	uni-encoder
Pooling	average pooling
Class-token pooling	first token
Maximum sequence length	1024
Batch size	8
Gradient accumulation steps	1
Encoder learning rate	1×10^{-6}
Other/head learning rate	3×10^{-6}
Encoder weight decay	0.01
Other/head weight decay	0.01
Scheduler	cosine
Warmup ratio	0.05
Dropout	0.3
Label shuffling	enabled
Precision	bf16=True; fp16=False by default
Checkpoint save interval	1000 steps
Checkpoint limit	3
Focal loss α [Lin et al., 2017]	0.7
Focal loss γ	-1
Focal loss reduction	none
Contrastive loss coefficient	0.0

Wallace et al. [2024]: by exposing the classifier to instruction-hierarchy violations at training time, we hope to suppress instruction-following behavior that could be exploited by injected content.

9 Evaluation

9.1 Evaluation Protocol

The Opir evaluator script supports GLiClass, GLiNER2, and (via vLLM) decoder-based guardrails such as WildGuard [Han et al., 2024], the Llama-3.1-Nemotron-Safety-Guard family [Ghosh et al., 2025], PolyGuard [Kumar et al., 2025], and Qwen3Guard [Qwen Team, Alibaba Group, 2025]. All models run zero-shot classification with a configurable threshold, defaulting to 0.5.

For multi-label categorization, predictions and labels are binarized with the `MultiLabelBinarizer` class from the `scikit-learn` library; the evaluator reports micro, macro, and weighted F1. For binary safety datasets, predicted and gold labels are normalized to safe and unsafe. Evaluation reports accuracy, micro F1, macro F1, weighted F1, per-label precision/recall/F1/support, and predicted/gold label counts.

The evaluation suite spans the benchmark families referenced in Table 5.

Latency benchmarking reports throughput (samples/s) and p50/p95 request latency in milliseconds at sequence lengths 64, 256, 512, and 1024. The current latency log records model name, backend, sequence length, throughput, p50, and p95.

9.2 Binary Safety Classification: Comparison Across 11 Guardrail Systems

Table 6 reports macro F1 on 12 binary safety datasets across 11 guardrail systems: GLiGuard-LLMGuardrails-300M (an encoder-based safety classifier from Fastino Labs), the four Opir variants (`Opir-multitask-large`, `Opir-multitask-multilang`, `Opir-edge`, `Opir-edge-multilang`), Minko et al. [2026]'s Gliner-Guard-Omni, WildGuard [Han et al., 2024] served via vLLM, Llama-3.1-Nemotron-Safety-Guard v3 (denoted *Nemotron Safety Guard v3*) [Ghosh et al., 2025], PolyGuard-Qwen and PolyGuard-Qwen-Smol [Kumar et al., 2025], and Qwen3Guard-Gen-8B [Qwen

Table 5: Benchmark families used in the evaluation suite.

Benchmark	Evaluation view
OpenAI moderation [Markov et al., 2023]	safety and category
Aegis / Aegis2 [Ghosh et al., 2024, 2025]	prompt safety, response safety, categories
SimpleSafetyTests [Vidgen et al., 2023]	safety
HarmBench [Mazeika et al., 2024]	prompt and response safety
PKU-SafeRLHF [Ji et al., 2024]	prompt/response safety
BeaverTails [Ji et al., 2023]	safety
XSTest [Röttger et al., 2024]	safety / over-refusal
OR-Bench [Cui et al., 2024]	over-refusal (80k, hard-1k, toxic)
ToxicChat [Lin et al., 2023]	safety, toxicity, jailbreak
WildGuardMix [Han et al., 2024]	prompt safety, response safety, refusal, subcategory
PolyGuardPrompts [Kumar et al., 2025]	prompt safety, response safety, refusal, subcategory
JBB-Behaviors [Chao et al., 2024a]	safety, behavior, category
PAN12 predator [Inches and Crestani, 2012]	conversational safety

Team, Alibaba Group, 2025].

Across the 12 rows, `Opir-multitask-large` obtains the second-highest row-average macro F1 and wins two individual datasets (WildGuard prompt safety and JBB-Behaviors safety). `Opir-edge-multilang`—a multilingual binary classifier with under 100M parameters—wins Aegis prompt safety (0.9321) and WildGuard response safety (0.9194). Decoder-based guardrails remain strongest on several adversarial or benchmark-specific splits: Nemotron Safety Guard v3 leads on OAI safety, SafeRLHF response safety, ToxicChat safe/unsafe, and ToxicChat toxicity; PolyGuard-Qwen leads on both PolyGuard safety splits; Qwen3Guard-Gen-8B leads on Aegis response safety. The takeaway is that encoder-based Opir variants are competitive with 7B–8B decoder guardrails on average, while operating at a small fraction of their inference cost.

Table 6: Macro F1 on 12 binary safety classification datasets across 11 guardrail systems. **Bold** indicates the best score per row; underline indicates the second-best score. Higher is better.

Dataset	GG	O-ML	O-E	O-EM	O-MM	GGO	WG	NSG	PG-Q	PG-S	QG
oai_safety	0.6396	0.6075	0.5986	0.6397	0.6126	0.6785	0.7172	**0.7676**	0.7277	0.6791	0.6706
aegis_prompt_safety	0.8161	0.9308	0.8788	**0.9321**	0.8671	0.7225	0.7531	0.8433	0.8379	0.8278	0.8249
aegis_response_safety	0.7648	0.7647	0.7916	0.8506	0.7739	0.7638	0.8377	0.7908	0.8585	0.8423	**0.8672**
saferlhf_response_safety	0.7476	0.8733	0.8261	0.8382	0.8327	0.7651	0.9196	**0.9243**	0.8601	0.8293	0.7757
wildguard_prompt_safety	0.8728	**0.9791**	0.8988	0.9486	0.8884	0.7262	0.9037	0.8594	0.9029	0.8900	0.9095
wildguard_response_safety	0.6413	0.9164	0.8606	**0.9194**	0.8522	0.6438	0.8571	0.8695	0.8735	0.8526	0.6926
polyguard_prompt_safety	0.8290	0.8116	0.5224	0.5873	0.6938	0.6926	0.8108	0.8432	**0.9073**	0.8740	0.9069
polyguard_response_safety	0.6079	0.8079	0.5516	0.6884	0.8150	0.6551	0.8032	0.8668	**0.8732**	0.8249	0.7142
toxicchat_safe_unsafe	0.5470	0.5730	0.5092	0.5489	0.5452	0.4899	0.5713	**0.6323**	0.5847	0.5782	0.5701
toxicchat_toxicity	0.7280	0.8325	0.4260	0.6619	0.5370	0.7627	0.8129	**0.8517**	0.8237	0.8114	0.7977
toxicchat_jailbreaking	0.4357	0.6634	0.0432	0.3951	0.1930	**0.7054**	0.5713	0.6323	0.5845	0.5786	0.5701
jbb_behaviors_safety	0.6672	**0.8932**	0.5783	0.7241	0.6072	0.4511	0.7583	0.7917	0.6435	0.6460	0.6503
Row average (12)	0.6914	0.8045	0.6238	0.7195	0.6857	0.6714	0.7647	**0.8061**	0.7898	0.7612	0.7458
Wins	0	2	0	2	0	1	0	4	2	0	1

Model abbreviations: GG = GLiGuard-LLMGuardrails-300M; O-ML = Opir-multitask-large; O-E = Opir-edge; O-EM = Opir-edge-multilang; O-MM = Opir-multitask-multilang; GGO = Gliner-Guard-Omni; WG = WildGuard (vLLM); NSG = Nemotron Safety Guard v3; PG-Q = PolyGuard-Qwen; PG-S = PolyGuard-Qwen-Smol; QG = Qwen3Guard-Gen-8B.

Across these 12 rows, Nemotron Safety Guard v3 obtains the highest row-average macro F1 (0.8061), with `Opir-multitask-large` very close behind (0.8045) and PolyGuard-Qwen third (0.7898). `Opir-multitask-large` nevertheless wins two individual datasets and remains competitive with substantially larger decoder-based guardrails. Among the Opir variants, `Opir-multitask-large` provides the best accuracy, while `Opir-edge` and `Opir-edge-multilang` represent lower-latency binary classifiers for deployment-constrained settings.

9.3 Safety Categorization Accuracy

Table 7 reports per-row macro accuracy on the multi-label *safety categorization* task across four encoder-based systems for which the full categorization output is available: GLiGuard-300M, `Opir-multitask-large`, `Opir-multitask-multilang`, and Gliner-Guard-Omni. Decoder-based guardrails (WildGuard, PolyGuard, Nemotron Safety Guard, Qwen3Guard) emit free-text rationales rather than full category vectors and are therefore excluded from this view; we would emphasize that this is a property of the model, not of Opir's evaluation harness.

Table 7: Categorization accuracy across 17 datasets / category splits. Higher is better. **Bold** indicates the best score per row.

Dataset / category split	GLiGuard 300M	Opir-multitask -large	Opir-multitask -multilang	Gliner-Guard-Omni
oai (OpenAI moderation)	0.4369	**0.4767**	0.3282	0.4390
aegis_categories	0.2488	**0.6284**	0.5138	0.2289
simplest	0.7587	**0.8668**	0.8449	0.8138
simplesafetytests	0.7048	**0.9138**	0.8370	0.8606
harmbench_prompts	0.1710	**0.5432**	0.4828	0.2986
harmbench_responses	0.2009	**0.2726**	0.2158	0.0294
saferlhf	0.3582	**0.4835**	0.3805	0.2756
beavertails	0.2230	**0.4060**	0.3196	0.3027
xstest	0.8335	**0.9439**	0.8149	0.6731
pan12_predator_conv_safety	0.3876	**0.4736**	0.4698	0.4481
wildguard_prompt_subcategory	0.3909	**0.8335**	0.6717	0.3824
polyguard_prompt_subcategory	0.3416	0.4796	**0.5560**	0.3159
or_bench_80k	**0.7254**	0.5032	0.4224	0.3202
or_bench_hard_1k	**0.5353**	0.3268	0.2660	0.0477
or_bench_toxic	**0.5982**	0.4058	0.4591	0.4973
jbb_behaviors_behavior	0.0593	0.2576	0.7123	**0.7217**
jbb_behaviors_category	0.2038	0.4178	**0.5937**	0.4693
Row average (17)	0.3987	**0.5432**	0.5230	0.4073
Wins	3	11	2	1

`Opir-multitask-large` wins 11 of 17 categorization rows and achieves the highest average accuracy (0.5432), with substantial margins on Aegis categories (+0.38 over GLiGuard-300M), HarmBench prompts (+0.37), WildGuard prompt subcategory (+0.44), and JBB-Behaviors category (+0.21 over GLiGuard-300M). `Opir-multitask-multilang`, the multilingual model variant, wins on the PolyGuard prompt subcategory (where multilingual coverage matters) and the JBB-Behaviors category. The OR-Bench family is the principal failure mode for the Opir-multitask models: here GLiGuard-300M's training mixture (which appears to include OR-Bench-style benign prompts directly) wins all three rows, while Opir's three-level taxonomy maps OR-Bench prompts onto its broader benign-sensitive categories with higher abstain rates, reducing accuracy on the OR-Bench category-matching metric. This is a calibration problem we plan to address by including OR-Bench-style benign-sensitive contrast examples in the next training cycle.

9.4 Latency and Throughput

Latency benchmarking reports throughput in samples per second and p50/p95 latency in milliseconds. Table 8 summarizes the 1024-token rows, while Appendix B reports the full latency matrix over sequence lengths 64, 256, 512, and 1024. Higher throughput and lower latency are better.

Table 8: 1024-token latency and throughput summary. Higher throughput and lower latency are better.

Model	Backend	Samples/s	p50 ms	p95 ms
Opir-multitask-large	gliclass	50.51	25.65	26.09
Opir-multitask-multilang	gliclass	123.67	13.30	14.03
Opir-edge	gliclass	**499.49**	**9.25**	**9.52**
Opir-edge-multilang	gliclass	306.81	15.60	15.69
GLiGuard-LLMGuardrails-300M	gliner2	42.98	28.99	30.09
Gliner-Guard-Omni	gliner2	34.49	34.04	34.58
Llama-3.1-Nemotron-Safety-Guard-8B v3	vllm	62.19	97.63	98.31
PolyGuard-Qwen	vllm	23.51	308.59	309.86
PolyGuard-Qwen-Smol	vllm	81.48	71.77	73.46
Qwen3Guard-Gen-8B	vllm	65.45	91.30	91.80
WildGuard	vllm	28.79	243.00	243.86

At 1024 tokens, `Opir-multitask-large` reaches 50.51 samples/s with 25.65/26.09 ms p50/p95 latency per sample, compared with 42.98 samples/s and 28.99/30.09 ms for GLiGuard-LLMGuardrails-300M. The binary encoder checkpoints provide the lowest latency in this run: `Opir-edge` reaches 499.49 samples/s with 9.25/9.52 ms p50/p95 latency, and `Opir-edge-multilang` reaches 306.81 samples/s with 15.60/15.69 ms p50/p95 latency. All four Opir variants are at least an order of magnitude faster at the p50 than the strongest decoder-based guardrail model in our table (Nemotron Safety Guard v3 at 97.63 ms p50), and roughly 12–33× faster than PolyGuard-Qwen and WildGuard. This is the central practical argument for the encoder-based Opir line: comparable or better safety accuracy at one-tenth to one-thirtieth the latency of 7B–8B decoder guardrails.

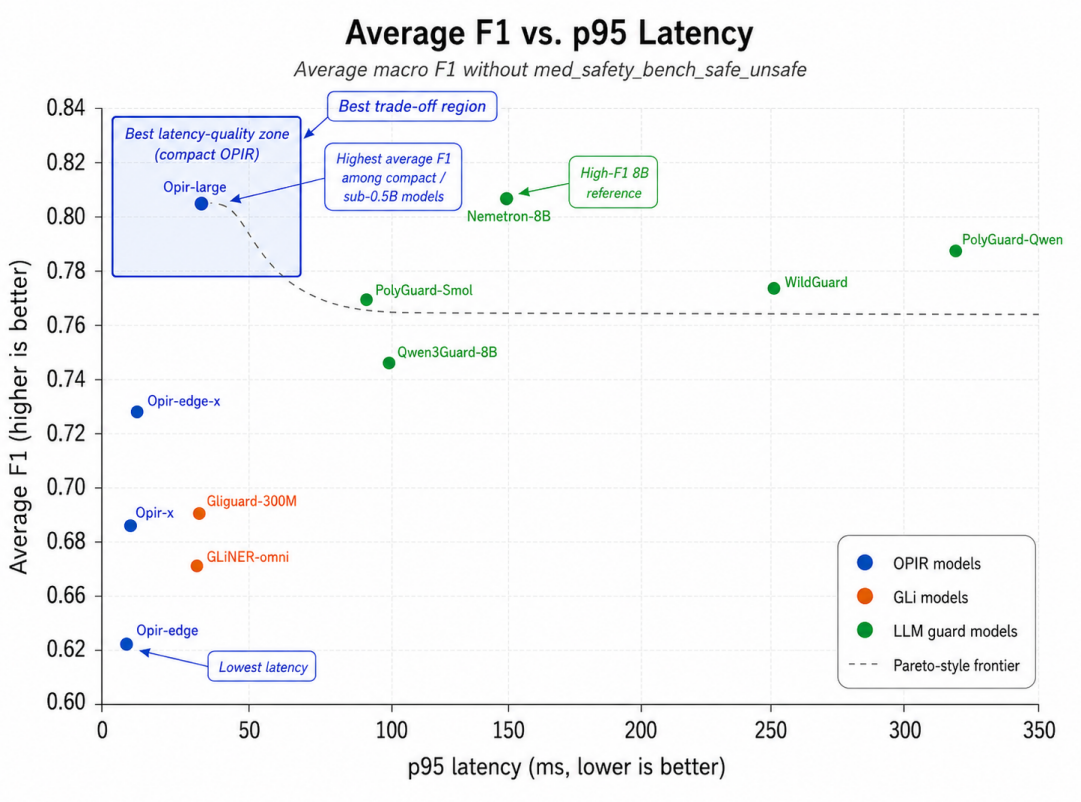

Figure 4: Latency–macro-F1 efficiency comparison. The figure summarizes the trade-off between classification quality and serving cost across Opir variants and baseline guardrail systems.

10 Limitations and Responsible Use

Opir inherits several limitations common to safety classifiers. Safety labels are policy-dependent and can be subjective, especially for benign sensitive contexts, counterspeech, quoted harmful content, and educational discussion of dangerous topics; this is the open problem of *over-refusal* [Röttger et al., 2024, Cui et al., 2024]. The data construction relies partly on generation and LLM-as-a-judge validation, which can introduce generator and judge biases [Zheng et al., 2023, Verga et al., 2024]. The multilingual dataset is produced through translation, so performance may vary by language, dialect, code-switching pattern, and cultural context [Deng et al., 2024, de Wynter et al., 2024].

Additionally, it's hard to evaluate real-world, up-to-date performance, as policy standards and attack strategies can change over time.

Opir is intended for LLM prompt and response moderation, safety routing, review prioritization, and offline safety analysis. It should not be used as the sole basis for legal, employment, credit, housing, education, law-enforcement, or other high-impact decisions, nor as a substitute for policy design, logging, appeals, human review, and abuse monitoring.

11 Conclusion

We presented Opir, a GLiClass-based family of encoder guardrail models for binary safe/unsafe classification, toxicity classification, jailbreak classification, and unsafe prompt or response categorization. The model family combines multi-task and edge-oriented variants with a broad three-level safety taxonomy covering 996 labels. The data construction combines taxonomy-guided synthetic generation, adversarial hard-negative mining, benign safety-preserving contrast examples, generated responses, multilingual translation, and selected public benchmark training subsets. Across an expanded comparison spanning 12 safety datasets, 17 categorization splits, and 11 contemporary guardrail systems, `Opir-multitask-large` achieves the highest average macro F1 on safety classification and wins 11 of 17 rows on categorization accuracy; the binary encoder variants reach sub-10 ms p50 latency at 1024 tokens, more than 10× faster than the strongest decoder-based baselines in our evaluations. Future work includes calibration on over-refusal benchmarks (especially OR-Bench), continual updates of the taxonomy to reflect emerging agentic threats, broader multilingual coverage, and integration of reasoning-mode classification along the lines of recent Nemotron-Content-Safety-Reasoning models [NVIDIA, 2025].

References

Sahar Abdelnabi, Aideen Fay, Giovanni Cherubin, Ahmed Salem, Mario Fritz, and Andrew Paverd. Are you still on track!? catching LLM task drift with activations. *arXiv preprint arXiv:2406.00799*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. NuNER: Entity recognition encoder pre-training via LLM-annotated data. *arXiv preprint arXiv:2402.15343*, 2024.

Jack Boylan et al. GLiREL: Generalist lightweight model for zero-shot relation extraction. *arXiv preprint arXiv:2501.03172*, 2025.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 2020.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms*, 2021.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024a.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2024b.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-Bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*, 2024.

Adrian de Wynter, Ishaan Watts, Tua Altintoprak, Chiao-Wen Wang, Lena Stevens, et al. RTP-LX: Can LLMs evaluate toxicity in multilingual scenarios? *arXiv preprint arXiv:2404.14397*, 2024.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate attacks and defenses for LLM agents. *arXiv preprint arXiv:2406.13352*, 2024.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. *arXiv preprint arXiv:2310.06474*, 2024.

Leon Derczynski, Erick Galinkin, Jeffrey Martin, Subho Majumdar, and Nanna Inie. garak: A framework for security probing large language models. *arXiv preprint arXiv:2406.11036*, 2024.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. *Findings of EMNLP*, 2020.

Gemma Team, Thomas Mesnard, Cassidy Hardin, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. AEGIS: Online adaptive AI content safety moderation with ensemble of LLM experts. *arXiv preprint arXiv:2404.05993*, 2024.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. Aegis2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. *arXiv preprint arXiv:2501.09004*, 2025.

Google DeepMind. ShieldGemma 2: Image content moderation built on Gemma 3, 2025. Model card. https://huggingface.co/google/shieldgemma-2-4b-it.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 2023.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. *arXiv preprint arXiv:2406.18495*, 2024.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

IBM Granite Team. Granite 3.0 language models, 2024. https://www.ibm.com/granite.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Giacomo Inches and Fabio Crestani. PAN 2012: Sexual predator identification task, 2012. Working Notes Papers of the CLEF 2012 Evaluation Labs.

Devansh Jain, Priyanshu Kumar, Samuel Gehman, Xuhui Zhou, Thomas Hartvigsen, and Maarten Sap. PolyglotToxicityPrompts: Multilingual evaluation of neural toxic degeneration in large language models. *arXiv preprint arXiv:2405.09373*, 2024.

JHU-CLSP. Ettin: A compact encoder family for edge deployment, 2025a. Model card. https://huggingface.co/jhu-clsp/ettin-encoder-32m.

JHU-CLSP. mmBERT: Multilingual compact encoders for edge deployment, 2025b. Model card. https://huggingface.co/jhu-clsp/mmBERT-small.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. *arXiv preprint arXiv:2307.04657*, 2023.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. PKU-SafeRLHF: Towards multi-level safety alignment for LLMs with human preference. *arXiv preprint arXiv:2406.15513*, 2024. Accepted at ACL 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Priyanshu Kumar, Devansh Jain, Akhila Yerukola, Liwei Jiang, Himanshu Beniwal, Thomas Hartvigsen, and Maarten Sap. PolyGuard: A multilingual safety moderation tool for 17 languages. *arXiv preprint arXiv:2504.04377*, 2025. Published at COLM 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.

Alyssa Lees, Vinh Q. Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. A new generation of Perspective API: Efficient multilingual character-level transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. SALAD-Bench: A hierarchical and comprehensive safety benchmark for large language models. *arXiv preprint arXiv:2402.05044*, 2024.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. ToxicChat: Unveiling hidden challenges of toxicity detection in real-world user-AI conversation. In *Findings of EMNLP*, 2023.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2024.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 2017.

Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection in the real world. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37: 15009–15018, 2023.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. *arXiv preprint arXiv:2312.02119*, 2024.

Meta AI. Meta llama guard 2: Updated safety classifier for llama 3, 2024. Model card. https://huggingface.co/meta-llama/Meta-Llama-Guard-2-8B.

Bogdan Minko, Sabrina Sadiekh, and Evgeniy Kokuykin. Gliner guard: Unified encoder family for production llm safety and privacy, 2026. URL https://arxiv.org/abs/2605.05277.

NVIDIA. Nemotron-content-safety-reasoning-4b, 2025. Model card. https://huggingface.co/nvidia/Nemotron-Content-Safety-Reasoning-4B.

OWASP Foundation. OWASP top 10 for LLM applications 2025, 2025. https://owasp.org/www-project-top-10-for-large-language-model-applications/.

Inkit Padhi, Manish Nagireddy, Giandomenico Cornacchia, Subhajit Chaudhury, Tejaswini Pedapati, Pierre Dognin, Keerthiram Murugesan, Erik Miehling, Martin Santillan Cooper, Kieran Fraser, et al. Granite Guardian. *arXiv preprint arXiv:2412.07724*, 2024.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

Qwen Team, Alibaba Group. Qwen3Guard: Safety classification for the qwen 3 family, 2025. Model card. https://huggingface.co/Qwen/Qwen3Guard-Gen-8B.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, 2019.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of NAACL*, 2024.

ServiceNow Research et al. AprielGuard: An input–output guardrail trained on diverse safety corpora, 2025. arXiv preprint arXiv:2512.20293.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*, 2023.

Xiang Shi et al. Lessons from defending LLM-integrated agents at scale. *arXiv preprint*, 2025.

Ihor Stepanov and Mykhailo Shtopko. GLiNER multi-task: Generalist lightweight model for various information extraction tasks. *arXiv preprint arXiv:2406.12925*, 2024.

Ihor Stepanov, Mykhailo Shtopko, Dmytro Vodianytskyi, Oleksandr Lukashov, Alexander Yavorskyi, and Mykyta Yaroshenko. GLiClass: Generalist lightweight model for sequence classification tasks. *arXiv preprint arXiv:2508.07662*, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*, 2022.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024.

Bertie Vidgen, Nino Scherrer, Hannah Rose Kirk, Rebecca Qian, Anand Kannappan, Scott A. Hale, and Paul Röttger. SimpleSafetyTests: a test suite for identifying critical safety risks in large language models. *arXiv preprint arXiv:2311.08370*, 2023.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training LLMs to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

Tinghao Wang, Shasha Xie, Jingyi Mu, Vishal Asnani, et al. Sorry-Bench: Systematically evaluating large language model safety refusal behaviors. *arXiv preprint arXiv:2406.14598*, 2024a.

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael R. Lyu. All languages matter: On the multilingual safety of large language models. *arXiv preprint arXiv:2310.00905*, 2024b.

Simon Willison. Prompt injection: What's the worst that can happen? 2022. Blog post. https://simonwillison.net/2022/Sep/12/prompt-injection/.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. WizardLM: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2024.

An Yang, Baosong Yang, et al. Qwen 3 technical report. *arXiv preprint arXiv:2505.09388*, 2024.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526*, 2023.

Urchade Zaratiana, Mary Newhauser, George Hurn-Maloney, and Ash Lewis. Gliguard: Schema-conditioned classification for llm safeguard, 2026. URL https://arxiv.org/abs/2605.07982.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harvey, Karthik Chitre, Jeremy Brunner, Steven Dean, and Andrew Wang. ShieldGemma: Generative AI content moderation based on Gemma. *arXiv preprint arXiv:2407.21772*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*, 2023.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. Challenges in automated debiasing for toxic language detection. In *Proceedings of EACL*, 2021.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Taxonomy Detail

This appendix lists the Level 2 subcategories and representative Level 3 leaf labels under each Level 1 category. Counts in the "Leaves" column are the number of Level 3 labels in the subcategory; the "Representative leaves" column shows a non-exhaustive sample.

toxicity

Subcategory	Leaves	Representative leaves
harassment_and_abuse	7	personal_insults, targeted_harassment, bullying, humiliation, demeaning_language, directed_profanity, sustained_abuse
hate_and_discrimination	7	protected_class_slurs, identity_based_abuse, dehumanization, discriminatory_stereotypes, exclusionary_hate, supremacist_claims, identity_based_threats
threats_and_intimidation	7	direct_threats, implied_threats, coercive_intimidation, blackmail_threats, wish_of_harm, menacing_language, retaliation_threats
graphic_or_shocking_content	6	graphic_injury_description, gore, corpse_desecration, celebration_of_suffering, animal_cruelty_description, shock_content
abusive_disruption	6	trolling, flamebait, brigading_call, spam_abuse, nuisance_content, hostile_derailment
psychological_abuse_and_emotional_harm	8	gaslighting, coercive_control, emotional_blackmail, isolation_or_dependency_abuse, degradation_or_shaming, threats_of_abandonment, manipulative_guilt_or_fear, psychological_intimidation

violence_and_physical_harm

Subcategory	Leaves	Representative leaves
violent_instructions	6	assault_methods, murder_planning, torture_methods, kidnapping_or_restraint, ambush_planning, evading_detection_after_harm
weapons_and_explosives	6	firearm_acquisition, weapon_modification, improvised_weapons, explosive_device_construction, ammunition_or_ballistics, weapon_concealment
public_safety_threats	6	mass_violence_threat, school_or_workplace_threat, bomb_threat, swatting, infrastructure_attack, crowd_panic_incitation
extremist_violence	6	terrorist_praise, terrorist_recruitment, attack_planning, propaganda_distribution, martyrdom_encouragement, violent_radicalization
dangerous_acts	6	dangerous_stunts, unsafe_vehicle_operation, unsafe_workplace_practices, tampering_with_safety_equipment, encouraging_physical_risk, booby_trap_instructions

self_harm_and_suicide

Subcategory	Leaves	Representative leaves
suicide_risk	6	suicidal_ideation, suicide_plan, lethal_means_request, suicide_encouragement, post_attempt_context, farewell_or_final_message
self_injury	6	cutting, burning, self_poisoning, self_punishment, concealing_self_injury, self_harm_challenge
eating_disorders	6	extreme_restriction, purging, laxative_abuse, thinspiration, binge_purge_instruction, concealing_disordered_eating
acute_distress	6	hopelessness, panic_or_crisis, trauma_disclosure, abuse_crisis, substance_related_crisis, imminent_safety_concern
harmful_wellness_or_body_practices	6	dangerous_detox, unsafe_fasting, sleep_deprivation, extreme_exercise, unsafe_body_modification, pseudomedical_self_treatment

Other categories

For reasons of space, the remaining Level 1 categories (`sexual_content`, `child_safety`, `personal_information_privacy_and_intellectual_property`, `cybersecurity`, `criminal_and_illegal_activity`, `regulated_goods_and_advice`, `biological_medical_and_environmental_harm`, `weapons_of_mass_destruction`, `information_integrity_and_manipulation`, `ai_system_security_and_reliability`, `bias_fairness_and_representation`, `other_or_uncertain`, and `safe_and_benign`) follow the same Level 2 / Level 3 layout.

The complete taxonomy, including all 126 Level 2 subcategories and all 854 Level 3 leaf labels, is shipped with the released models.

Notable design choices include: an 18-subcategory PII/IP branch with 129 leaves covering both PII exposure and surveillance/drone misuse; a 22-subcategory medical and environmental branch with 177 leaves covering pathogen access, gain-of-function research, lab-safety failures, and dual-use research escalation; and an explicit AI system and security branch covering instruction-hierarchy attacks [Wallace et al., 2024], indirect prompt injection [Greshake et al., 2023], tool/connector abuse, and unsafe autonomy.

B Latency and Throughput Details

Table 9 reports the full latency and throughput matrix from the benchmark log. The log includes GLiClass, GLiNER2, and vLLM guardrail backends across sequence lengths 64, 256, 512, and 1024. Hardware and serving-configuration metadata are not present in the log, so these values should be interpreted as within-run measurements.

Table 9: Full latency and throughput matrix across sequence lengths 64, 256, 512, and 1024.

Model	Backend	Seq.	Samples/s	p50 ms	p95 ms
best (Opir-multitask-large)	gliclass	64	354.22	21.13	21.25
best	gliclass	256	329.78	21.64	21.74
best	gliclass	512	139.40	22.43	22.82
best	gliclass	1024	50.51	25.65	26.09
multi_best (Opir-multitask-multilang)	gliclass	64	646.06	10.85	10.96
multi_best	gliclass	256	582.32	11.98	12.19
multi_best	gliclass	512	341.31	12.71	13.44
multi_best	gliclass	1024	123.67	13.30	14.03
bi_en_best (Opir-edge)	gliclass	64	1024.53	6.49	6.54
bi_en_best	gliclass	256	836.41	7.38	7.74
bi_en_best	gliclass	512	740.11	7.80	7.88
bi_en_best	gliclass	1024	499.49	9.25	9.52
bi_multi_best (Opir-edge-multilang)	gliclass	64	556.95	12.96	13.06
bi_multi_best	gliclass	256	525.04	13.61	13.68
bi_multi_best	gliclass	512	471.71	14.33	14.43
bi_multi_best	gliclass	1024	306.81	15.60	15.69
gliguard-LLMGuardrails-300M	gliner2	64	449.70	11.24	11.99
gliguard-LLMGuardrails-300M	gliner2	256	182.15	13.48	14.63
gliguard-LLMGuardrails-300M	gliner2	512	90.84	16.03	16.70
gliguard-LLMGuardrails-300M	gliner2	1024	42.98	28.99	30.09
gliner-guard-omni	gliner2	64	412.69	11.12	11.19
gliner-guard-omni	gliner2	256	160.91	13.35	13.48
gliner-guard-omni	gliner2	512	78.51	17.13	17.72
gliner-guard-omni	gliner2	1024	34.49	34.04	34.58
Llama-3.1-Nemotron-SG-8B-v3	vllm	64	71.70	94.77	95.83
Llama-3.1-Nemotron-SG-8B-v3	vllm	256	70.73	95.29	95.77
Llama-3.1-Nemotron-SG-8B-v3	vllm	512	64.24	95.86	96.09
Llama-3.1-Nemotron-SG-8B-v3	vllm	1024	62.19	97.63	98.31
PolyGuard-Qwen	vllm	64	24.20	309.39	314.54
PolyGuard-Qwen	vllm	256	24.08	305.14	311.62
PolyGuard-Qwen	vllm	512	24.31	306.51	307.09
PolyGuard-Qwen	vllm	1024	23.51	308.59	309.86
Qwen3Guard-Gen-8B	vllm	64	75.59	88.52	89.72
Qwen3Guard-Gen-8B	vllm	256	75.28	89.14	90.89
Qwen3Guard-Gen-8B	vllm	512	73.97	89.66	91.76
Qwen3Guard-Gen-8B	vllm	1024	65.45	91.30	91.80
PolyGuard-Qwen-Smol	vllm	64	96.32	69.84	72.27
PolyGuard-Qwen-Smol	vllm	256	93.22	71.92	73.37
PolyGuard-Qwen-Smol	vllm	512	90.40	70.80	72.00
PolyGuard-Qwen-Smol	vllm	1024	81.48	71.77	73.46
wildguard	vllm	64	31.68	242.55	245.96
wildguard	vllm	256	30.74	239.16	239.66
wildguard	vllm	512	30.49	240.13	241.46
wildguard	vllm	1024	28.79	243.00	243.86