

PaxmodGuard: A Small Multilingual Guardrail for Children’s Game Chat and AI Companions

Alan Agon Mai-Chi Nguyen
Paxmod, Adelaide, Australia
research@paxmod.com

October 2026

Abstract

General-purpose guardrail models are trained to judge prompts sent to an assistant. Deployed on children’s game chat or AI companions, they flag banter, roleplay and trash talk while missing the signals that matter there: a child stating their age, a request to move to another app, an adult checking whether a child is alone. We present PaxmodGuard, a 307M-parameter multilingual guardrail fine-tuned from mmBERT-base that scores 48 safety labels in one forward pass for a single message, a message with the turns before it, or a whole conversation. On about 53,000 messages of real chat from two children’s online games, collected after our training data was frozen, PaxmodGuard ranks harmful messages better than every open guard model we evaluated, including models up to 30 times its size (ROC-AUC 0.85 and 0.87 against at most 0.80), and it wrongly blocks 17% of OR-Bench-Hard prompts at its default thresholds against 38% to 89% for the general guard models we measured. It is weaker than larger guard models on general-purpose benchmarks (mean AUPRC 0.81 against 0.92 for Qwen3Guard-8B), and on precision among the most confident flags it leads on neither game. We release the weights under Apache-2.0 with JavaScript and Python SDKs that run on an ordinary CPU.

1 Introduction

Guardrails sit in front of most deployed language models and chat systems. The strongest open guard models, such as Llama Guard [2], ShieldGemma [3], Granite Guardian [4] and Qwen3Guard [5], are generative models of 1B to 27B parameters that read a prompt (and optionally a response) and emit a safe or unsafe verdict with a harm category. Their taxonomies and training data centre on an assistant being asked for something harmful.

Two settings fit this framing poorly. In *player-to-player chat* in children’s games, most messages are short, informal and multilingual, insults are usually play, and the harms that matter most (grooming, sextortion, sharing personal details) are often built from messages that are individually harmless. In *AI companions*, the risk includes what the AI itself does: claiming to be human, fostering dependency, or responding badly to a child or to a disclosure of self-harm. In both settings, a guardrail that blocks too much is switched off, and one that needs a GPU per message is too expensive at the volume of chat.

PaxmodGuard targets these settings with three design choices:

1. **An encoder, not a generator.** A guardrail decision is a classification problem. A 307M-parameter encoder scores every label in one pass and runs on CPU, in line with recent small-guard work [6].

2. **A taxonomy for children and companions.** 48 labels, including individual grooming tactics, minors revealing their age, moving to another platform, and AI-companion behaviours, alongside the usual harm categories.
3. **Role-tagged context.** One model reads a single message, a message with the turns before it, or a whole conversation, with each turn tagged by who wrote it.

We evaluate with threshold-free metrics, since deployments set their own thresholds, on unseen real chat from two children’s games as well as eight public benchmarks and an over-blocking benchmark.

2 Model

Architecture. PaxmodGuard is mmBERT-base [1], a ModernBERT-style encoder pretrained on about 3 trillion tokens in over 1,800 languages, with mean pooling and a linear head of 50 independent sigmoid outputs: the 48 labels of Table 1 and two summary scores, `unsafe` (any core harm under our policy) and `unsafe_strict` (the broader definition used by public benchmarks). The released model is quantised to int8 ONNX (about 220 MB).

Input format. Every turn is prefixed with a role tag: `[USER]` for a person writing to an AI, `[AI]` for the AI, and `[CHAT:A]`, `[CHAT:B]`, ... for players in game chat. Four input shapes share the same weights:

- a single message, e.g. `[CHAT] wanna add me on snap`;
- a message with earlier turns, where the judged turn is marked `[TARGET]` and earlier turns give context;
- a whole stretch of conversation, prefixed `[CONVERSATION]`, labelled for what happens anywhere in it;
- an AI reply judged together with the prompt before it.

The judged turn is kept up to 1,000 characters and earlier turns up to 300, matching training. Long single messages are read in overlapping windows at two sizes (full length and 128 tokens), so that one harmful sentence in a long message is not diluted; conversations are read in windows of 40 turns overlapping by 20. A label’s score is its maximum over windows.

Signal labels. Ten labels are scored on every input but excluded from the default flag set, because on their own they are normal in most products: `distress`, `romantic`, `profanity`, `age_inquiry`, the four grooming tactics (`risk_check`, `secrecy`, `isolation`, `special_bond`), and the two summary scores. Deployments enable them per product, for example tactic signals in a children’s game.

Policy. Labels follow a written policy that is lax by design: a label means a moderator would genuinely want to see the message. Banter, game violence, roleplay, hyperbole and mild insults are not harm by default. Grooming is decomposed into tactics (checking whether a child is alone, asking for secrecy, isolating them from family, special-bond language) that are labelled on single messages, and a conversation-level `grooming` label for when they add up.

3 Training

Data. PaxmodGuard is trained on publicly available and synthetic data, labelled against our written safety policy. The training set has 226,691 examples, with 6,430 for validation and 10,768 held out for testing. It mixes single messages, AI replies with their prompts, messages with earlier

Table 1: The 48 PaxmodGuard labels. Labels in *italics* are signal labels, scored but off by default.

Area	Labels
Self-harm	self_harm_intent, self_harm_instructions, eating_disorder, <i>distress</i>
Sexual	sexual, sexual_suggestive, sexual_minor, <i>romantic</i>
Child safety	minor_signal, grooming, grooming_suspected, image_request, sextortion, predator_report, abuse_disclosure, <i>age_inquiry, risk_check, secrecy, isolation, special_bond</i>
Personal information	pii_shared, pii_request, pii_other, off_platform, meetup
Abuse	harassment, threat, hate, <i>profanity</i>
Violence and danger	violence, extremism, dangerous_challenge, weapons, drugs, crime
Fraud	scam, spam
Advice	medical_advice, financial_advice, legal_advice
AI behaviour	jailbreak, prompt_injection, ai_claims_human, ai_claims_professional, dependency, manipulation, ai_harmful_compliance, ai_hostility

turns, and multi-turn conversation windows, in many languages. Three kinds of augmentation target known weaknesses: disguised text (character substitution, spacing and leetspeak of the kind used to evade filters), context windows cut from longer conversations so that the same message is seen with and without the turns that change its meaning, and hard negatives that resemble harm but are allowed by the policy (game violence, roleplay, idioms). The `unsafe_strict` target follows the source datasets’ own harmful or unhelpful labels where they exist. No chat from Paxmod customers is used to train the open model.

Optimisation. We fine-tune all weights for 2 epochs with AdamW (learning rate 3×10^{-5} , weight decay 0.01, 6% linear warm-up then linear decay, gradient clipping at 1.0) at a maximum length of 1,024 tokens. Batches are built from examples of similar length up to 8,192 tokens each, since inputs range from one-word messages to 40-turn conversations. The loss is binary cross-entropy per label with a positive weight $w_\ell = \text{clip}((N - n_\ell)/(10 n_\ell), 1, 10)$ for a label with n_ℓ positives among N examples, so rare labels are up-weighted without dominating. Training takes one NVIDIA L4 GPU.

Thresholds. Each label’s default threshold maximises $F_{0.5}$ on the validation set, favouring precision. Thresholds ship with the model and can be overridden per label.

4 Evaluation

All metrics are threshold-free: AUPRC (average precision), which measures how well scores rank harmful above harmless inputs with emphasis on the top of the ranking, and ROC-AUC, which measures ranking across the whole range. PaxmodGuard’s harm score is the maximum over its non-signal labels. For generative guards we use the probability of the model’s unsafe verdict, read from the token log-probabilities at the verdict position (for Qwen3Guard, “Controversial” counts towards unsafe). AI replies are always judged with the prompt before them. All models were run by us on identical inputs.

Baselines are Qwen3Guard (0.6B, 4B, 8B) [5], ShieldGemma 9B [3], Granite Guardian 3.3 8B [4], Llama Guard 3 8B [2] and GLiGuard (0.3B) [6].

Table 2: Unseen real chat from two children’s games. Threshold-free; higher is better. Random scoring gives AUPRC equal to the harm rate (about 0.01) and ROC-AUC 0.5. Best open model per column in bold.

Model	Params	Game A		Game B	
		AUPRC	ROC-AUC	AUPRC	ROC-AUC
PaxmodGuard v1	0.3B	0.17	0.85	0.12	0.87
GLiGuard	0.3B	0.02	0.71	0.04	0.72
Qwen3Guard-0.6B	0.6B	0.07	0.79	0.15	0.76
Qwen3Guard-4B	4B	0.13	0.80	0.38	0.79
Qwen3Guard-8B	8B	0.13	0.80	0.28	0.73
Granite Guardian 3.3	8B	0.08	0.77	0.15	0.70
Llama Guard 3	8B	0.06	0.80	0.03	0.78
ShieldGemma	9B	0.19	0.79	0.16	0.75
PaxmodGuard Cloud [†]	0.3B	0.28	0.87	0.22	0.98

[†]Not released. Same architecture, also trained on proprietary data that includes chat from the same platforms collected before the freeze, so it is not an out-of-domain result.

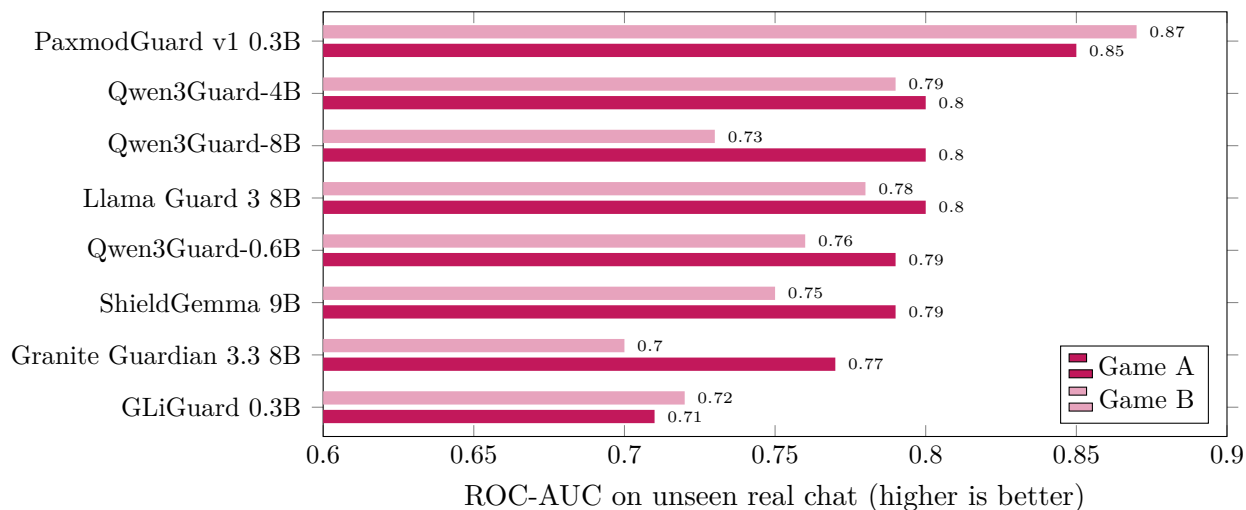


Figure 1: ROC-AUC on unseen real chat from two children’s games. PaxmodGuard v1 ranks harm best on both games among the open models evaluated, which are up to 30 times its size.

4.1 Real children’s game chat

Data. We took about 53,000 messages (52,685) of player chat from two children’s online games, Game A and Game B, stored after our training data was frozen, so no model saw them. Labelling every message is impractical at a harm rate of about 1%, so we labelled, against our written policy and with the surrounding conversation visible, every message flagged by Paxmod’s production moderation or by development versions of PaxmodGuard, plus a uniform random sample of the remaining messages: 2,493 messages for Game A (165 harmful) and 2,119 for Game B (76 harmful). Unflagged messages are weighted by the inverse of their sampling rate so that the metrics estimate performance on the full stream, whichever system flagged them; the estimated harm rate is 1.0% for Game A and 0.6% for Game B. Every model scored each message together with up to eight earlier messages from the same conversation. This data is not public and no message from it appears in this report.

Table 3: Public benchmarks, AUPRC. TC = ToxicChat, OAI = OpenAI moderation, Aeg = Aegis 2.0, XS = XSTest, AegR = Aegis 2.0 responses, BT = BeaverTails responses, PG = PolyGuard (17 languages), WG = WildGuardTest. GLiGuard response scores are omitted (incomplete run) and its mean covers six sets.

Model	TC	OAI	Aeg	XS	AegR	BT	PG	WG	Mean
PaxmodGuard v1	0.73	0.83	0.89	0.76	0.86	0.88	0.76	0.81	0.81
PaxmodGuard v1, <code>unsafe_strict</code>	0.64	0.87	0.90	0.81	0.89	0.88	0.77	0.82	0.82
Qwen3Guard-0.6B	0.86	0.86	0.92	0.95	0.92	0.94	0.91	0.95	0.91
Qwen3Guard-4B	0.88	0.86	0.91	0.98	0.93	0.93	0.92	0.95	0.92
Qwen3Guard-8B	0.86	0.86	0.92	0.99	0.94	0.93	0.93	0.95	0.92
Granite Guardian 3.3 8B	0.78	0.84	0.90	0.96	0.89	0.91	0.88	0.94	0.89
Llama Guard 3 8B	0.49	0.87	0.89	0.97	0.88	0.89	0.87	0.91	0.85
ShieldGemma 9B	0.82	0.89	0.89	0.91	0.86	0.77	0.78	0.82	0.84
GLiGuard 0.3B	0.75	0.74	0.92	0.92	–	–	0.85	0.90	0.85

Results. Table 2 and Figure 1 show the results. PaxmodGuard has the highest ROC-AUC on both games (0.85 and 0.87, against at most 0.80 for any baseline), meaning it ranks harmful messages above harmless ones most reliably across the whole stream. On AUPRC, which rewards precision among the highest-scoring messages, ShieldGemma leads on Game A (0.19 against 0.17) and Qwen3Guard-4B leads on Game B (0.38 against 0.12): larger models are more precise among their most confident flags on one game each. At a harm rate near 1%, absolute AUPRC is low for every system, and no model we tested, PaxmodGuard included, reaches 90% precision at a useful recall. Scores should be used to rank and route messages to people rather than as a sole decision.

The labels follow our own written policy, which favours a model trained on that policy, and the two games are only two products. We therefore read these results as evidence that PaxmodGuard fits this setting, not as a general ranking of the models.

4.2 Public benchmarks

We evaluate on eight public sets (Table 3): prompt classification on ToxicChat [7], the OpenAI moderation set [8], Aegis 2.0 [9] and XSTest [10]; response classification on Aegis 2.0 responses and BeaverTails [11]; multilingual prompts from PolyGuard in 17 languages [12]; and WildGuardTest prompts [13]. We report PaxmodGuard’s default harm score and its `unsafe_strict` score, which follows the broader definition these sets use.

PaxmodGuard is behind the larger guard models on these sets, most clearly against Qwen3Guard, including its 0.6B version. Part of the gap is definitional: these sets label as unsafe much of what our policy deliberately allows (controversial opinions, mild insults, educational answers about dangerous topics), and XSTest in particular rewards flagging prompts that only mention a harmful topic. The `unsafe_strict` score narrows but does not close the gap. We consider general-purpose benchmark accuracy a real limitation for teams whose main need is assistant prompt filtering.

4.3 Over-blocking

OR-Bench-Hard [14] contains 1,319 safe prompts that only sound harmful. At each model’s default setting, PaxmodGuard wrongly flags 17%, against 38% for Qwen3Guard-8B, 49% for Qwen3Guard-0.6B, 82% for gpt-oss-safeguard-20B and 89% for Granite Guardian 3.3 8B. Low over-blocking is a design goal: in our experience, false alarms are the main reason moderation is switched off.

4.4 Speed

Measured through the JavaScript SDK on two CPU threads of an Apple M4 laptop (median of 30 runs, int8 ONNX), a short message takes 7ms, a message with eight earlier turns 62ms, and a 40-turn conversation 533ms. Memory use is about 850 MB. Every label is scored in the same pass, and no GPU is needed.

5 Illustration

On the synthetic conversation below, written for illustration, the conversation-level input scores `minor_signal` 1.00, `risk_check` 1.00, `secrecy` 0.98, `off_platform` 0.98 and `grooming` 0.95, while `hate` scores 0.00. No single message is sexual; the pattern is what the model labels.

```
A: yo ur good at this game  B: thx  A: how old r u  B: 13  A: cool im 19
A: ur parents home rn?  B: no at work  A: dont tell ur mom we talk ok  A: add me on snap
```

6 Limitations

- Real children’s chat is hard: no model we tested reaches 90% precision at a useful recall on it. Scores should route messages to people, not replace them.
- Grooming usually unfolds over days and many messages. A single message carries only tactic signals; whole conversations must be checked to see the pattern, and the model does not track a person across conversations.
- Larger general guard models are better on general-purpose benchmarks.
- Most training data is English and Spanish. Other languages are covered but less tested.
- Rare harms such as extremism and dangerous challenges have few training examples.
- The real-chat evaluation covers two games, labelled against our own policy.
- As a guardrail it only prevents harm when it runs before delivery; used after delivery, it cannot stop a first harmful message from being shown.

7 Release

The weights, tokenizer and per-label thresholds are released under Apache-2.0 at <https://huggingface.co/paxmod/guard>. The `@paxmod/guard` (npm) and `paxmod-guard` (PyPI) SDKs produce identical scores, build the role-tagged inputs, and run fully locally. A live demo and the launch post are at <https://www.paxmod.com/research/guard>.

References

- [1] M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, B. Van Durme. mmbert: A Modern Multilingual Encoder with Annealed Language Learning. ICML 2026. arXiv:2509.06888.
- [2] H. Inan et al. Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations. 2023.
- [3] W. Zeng et al. ShieldGemma: Generative AI Content Moderation Based on Gemma. 2024.
- [4] I. Padhi et al. Granite Guardian. IBM, 2024.
- [5] Qwen Team. Qwen3Guard Technical Report. 2025.

- [6] Fastino. GLiGuard: 16x Faster Safety Moderation with a Small Language Model. 2026.
- [7] Z. Lin et al. ToxicChat: Unveiling Hidden Challenges of Toxicity Detection in Real-World User-AI Conversation. 2023.
- [8] T. Markov et al. A Holistic Approach to Undesired Content Detection in the Real World. AAAI 2023.
- [9] S. Ghosh et al. Aegis2.0: A Diverse AI Safety Dataset and Risks Taxonomy for Alignment of LLM Guardrails. 2025.
- [10] P. Röttger et al. XSTest: A Test Suite for Identifying Exaggerated Safety Behaviours in Large Language Models. NAACL 2024.
- [11] J. Ji et al. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. NeurIPS 2023.
- [12] P. Kumar et al. PolyGuard: A Multilingual Safety Moderation Tool for 17 Languages. 2025.
- [13] S. Han et al. WildGuard: Open One-Stop Moderation Tools for Safety Risks, Jailbreaks, and Refusals of LLMs. NeurIPS 2024.
- [14] J. Cui et al. OR-Bench: An Over-Refusal Benchmark for Large Language Models. 2024.