

Zen3-Guard: Third-Generation Multilingual Content Safety Model

Technical Whitepaper v2025.06

Zen LM Research Team
research@zenlm.org
papers.zenlm.org

June 2025

Abstract

Zen3-Guard is a dedicated 8B parameter safety classification model for real-time content moderation, providing high-accuracy detection of harmful content across 100+ languages with inference latency suitable for production API deployments. Built on the Zen MoDE (Mixture of Distilled Experts) architecture with a safety-specialized expert cluster configuration, Zen3-Guard achieves ToxiGen 99.2%, HatEval 97.8%, OffComEval 96.4%, and multilingual macro-F1 of 0.968 across all supported languages. At 8B parameters, the model delivers sub-5ms classification latency per request when batch-optimized on A10G GPU hardware, enabling real-time moderation of high-throughput API deployments. The model is designed as a drop-in safety layer for all Zen family models, configurable to enforce organization-specific content policies through a policy injection interface.

Contents

1	Introduction	3
2	Architecture	3
2.1	Safety-Specialized Zen MoDE Configuration	3
2.2	Harm Taxonomy and Output Schema	4
2.3	Multilingual Architecture	4
2.4	Policy Injection Interface	4
3	Training	5
3.1	Safety Annotation Data	5
3.2	Training Procedure	5
3.3	Calibration	6
4	Evaluation	6
4.1	Primary Safety Benchmarks	6
4.2	Multilingual Performance	6
4.3	Inference Latency	6
4.4	Adversarial Robustness	7
4.5	False Positive Rate	7
5	Related Work	7
5.1	Content Moderation Systems	7
5.2	Safety-Tuned Language Models	7
5.3	Multilingual Safety	7

6	Deployment Recommendations	8
6.1	Integration Patterns	8
6.2	Threshold Configuration	8
7	Conclusion	8

1 Introduction

Content safety classification is a critical infrastructure component for any deployed language model service. Effective moderation must satisfy competing constraints: high recall on harmful content (to prevent policy violations), high precision (to avoid false positives that degrade user experience), coverage across 100+ languages (to serve global user populations), and sub-10ms inference latency (to insert into API request paths without perceptible overhead).

Existing safety approaches are inadequate in at least one dimension:

- Keyword filters: high false positive rate, trivially bypassed by paraphrasing
- Binary classifiers: insufficient granularity for nuanced policy enforcement
- Large safety-tuned generation models: accurate but too slow for real-time use
- Embedding-similarity approaches: brittle to adversarial rephrasing

Zen3-Guard addresses all four constraints simultaneously through:

1. **Safety-specialized Zen MoDE architecture** — A dedicated subset of experts trained exclusively on safety-relevant data, activated when routing detects safety-sensitive content patterns.
2. **Hierarchical harm taxonomy** — A 47-category harm taxonomy covering violence, hate speech, harassment, self-harm, illegal activity, privacy violations, and sensitive content, with configurable policy enforcement per category.
3. **Multilingual safety training** — Dedicated safety annotation data in 100+ languages, with special attention to code-switching and cross-lingual adversarial evasion.
4. **Policy injection interface** — Organization-specific content policies are encoded as natural language policy documents and injected at inference time, allowing policy customization without model retraining.

2 Architecture

2.1 Safety-Specialized Zen MoDE Configuration

Zen3-Guard uses an 8B Zen MoDE backbone with 32 experts, where the expert population is partitioned into three functional groups:

General language experts (experts 1–12): Standard language understanding, providing the contextual representations needed for nuanced safety judgment. These experts capture pragmatics, irony, and implicit meaning that keyword systems miss.

Safety-specialized experts (experts 13–24): Trained predominantly on annotated safety datasets including ToxiGen, HatEval, OffComEval, and proprietary moderation datasets from production deployments. These experts develop representations tightly coupled to harm patterns.

Policy enforcement experts (experts 25–32): Process policy injection context and map general harm representations to organization-specific policy decisions.

The router is trained with a safety-aware auxiliary loss that encourages activation of safety-specialized experts when input tokens match patterns associated with policy-sensitive categories:

$$\mathcal{L}_{\text{safety-route}} = - \sum_{c \in \mathcal{C}_{\text{harmful}}} \mathbb{I}[\text{expert} \in \mathcal{E}_{\text{safety}}] \cdot \log p(\text{expert}|x) \quad (1)$$

2.2 Harm Taxonomy and Output Schema

Zen3-Guard outputs a structured classification across 47 harm categories organized in a 5-level hierarchy:

1. Violence (physical threat, incitement, graphic content)
2. Hate speech (identity-based, stereotyping, dehumanization)
3. Harassment (personal attack, doxxing, coordinated abuse)
4. Self-harm and mental health (suicide, eating disorders, self-injury)
5. Illegal activity (CSAM, fraud, controlled substances, weapons)
6. Privacy (PII exposure, surveillance, data theft)
7. Misinformation (health, political, scientific)
8. Sensitive content (adult content, religious sensitivity, political sensitivity)

Each category produces:

- Binary decision: **safe** / **unsafe**
- Confidence score: $p \in [0, 1]$
- Severity level: **low** / **medium** / **high** / **critical**
- Evidence span: character offsets of the triggering text segment

2.3 Multilingual Architecture

Zen3-Guard uses a shared multilingual tokenizer with 150,528 vocabulary tokens covering 100+ languages with balanced script coverage (Latin, Cyrillic, Arabic, CJK, Devanagari, and 40+ additional scripts). Language-specific routing biases are learned during training to account for language-specific harm expression patterns.

For low-resource languages with fewer than 1M safety-annotated tokens, a cross-lingual transfer mechanism leverages semantic similarity to high-resource language annotations:

$$p(y|x_{\text{low}}) = \sum_{\ell \in \mathcal{L}_{\text{high}}} w_{\ell} \cdot p(y|\phi(x_{\text{low}}, x_{\ell})) \quad (2)$$

where ϕ is a cross-lingual alignment function and w_{ℓ} are similarity weights.

2.4 Policy Injection Interface

Organizations deploying Zen3-Guard can specify custom content policies as natural language documents. At inference time, the policy document is prepended to the classification context as a special [POLICY] token sequence:

```
[POLICY] Our platform prohibits: discussion of competitor products,  
political content of any kind, and medical advice. Legal content  
standards: US jurisdiction, adult content permitted with age verification.  
[/POLICY]  
[INPUT] <user content to classify> [/INPUT]
```

The policy injection is processed by the policy enforcement expert cluster, which overrides default safety decisions based on organization-specific rules. This enables a single deployed model instance to serve multiple tenants with different content policies.

3 Training

3.1 Safety Annotation Data

Zen3-Guard is trained on a 12B token safety-focused corpus:

- ToxiGen (machine-generated, diverse group coverage): 274K examples
- HatEval 2019 (multilingual hate speech, EN/ES): 19K examples
- OffComEval (offensive language): 14K examples
- Jigsaw Unintended Bias (conversation toxicity): 1.8M examples
- WildGuard (instruction following safety): 92K examples
- OpenAI Policy Violations (annotated API violations): 240K examples
- Proprietary production moderation logs (de-identified): 8M examples
- Adversarial probing dataset (red-team generated): 2.1M examples
- Multilingual safety corpus (100+ languages): 4.2M examples
- Safe content (negative examples for precision): 12M examples

3.2 Training Procedure

Phase 1 — Base language model initialization: Initialize from the Zen3-Small (8B) base checkpoint, which provides strong multilingual language understanding.

Phase 2 — Safety expert specialization: Train only the safety-designated expert cluster on safety annotation data with a focal loss weighted toward false negatives:

$$\mathcal{L}_{\text{focal}} = -\alpha_t(1 - p_t)^\gamma \log(p_t), \quad \gamma = 2, \quad \alpha_{\text{positive}} = 4 \quad (3)$$

The positive class weight of 4 reflects the production priority of catching harmful content (false negatives) over minimizing false positives.

Phase 3 — End-to-end fine-tuning: Joint training of all components on the full safety corpus with the taxonomy hierarchy loss:

$$\mathcal{L}_{\text{taxonomy}} = \sum_{l=1}^5 w_l \sum_{c \in \mathcal{C}_l} \mathcal{L}_{\text{BCE}}(p_c, y_c) \quad (4)$$

where w_l increases with hierarchy depth to enforce consistency between parent and child category predictions.

Phase 4 — Adversarial robustness training: Additional training on adversarially generated evasion attempts, including:

- Character substitution (l33t speak, Unicode confusables)
- Paraphrasing through synonym replacement
- Code-switching and macaronic text
- Prompt injection via role-playing framing
- Instruction hierarchy attacks (“ignore previous instructions”)

3.3 Calibration

Zen3-Guard confidence scores are calibrated using temperature scaling on a held-out calibration set. Expected Calibration Error (ECE) after calibration: 0.018, indicating that a predicted confidence of 0.90 corresponds to approximately 90% empirical accuracy.

4 Evaluation

4.1 Primary Safety Benchmarks

Table 1: Primary safety classification benchmarks

Model	ToxiGen	HatEval	OffComEval	WildGuard
Zen-Guard	97.1	94.3	92.1	88.4
Zen2-Guard	98.4	96.2	94.8	91.7
Zen3-Guard	99.2	97.8	96.4	94.3

4.2 Multilingual Performance

Table 2: Multilingual safety classification (macro-F1 by language family)

Language Family	Zen3-Guard F1	Zen2-Guard F1
Germanic (EN, DE, NL, SV)	0.981	0.972
Romance (ES, FR, IT, PT)	0.974	0.963
Slavic (RU, PL, CS, UK)	0.968	0.951
Semitic (AR, HE, AM)	0.961	0.938
CJK (ZH, JA, KO)	0.972	0.956
Indic (HI, BN, TA, TE)	0.954	0.927
Low-resource (50+ others)	0.943	0.901
Overall average	0.968	0.944

4.3 Inference Latency

Table 3: Inference latency by hardware (batch size 32, sequence length 512)

Hardware	P50 Latency (ms)	P99 Latency (ms)	Throughput (req/s)
NVIDIA A10G (24GB)	3.8	5.1	8400
NVIDIA A100 (80GB)	2.1	3.2	15200
NVIDIA L4 (24GB)	4.2	5.8	7600
CPU (x86-64, INT8)	24.0	31.0	1300

4.4 Adversarial Robustness

Table 4: Robustness to adversarial evasion (detection rate %)

Attack Type	Zen3-Guard	Zen2-Guard
Character substitution (l33t speak)	98.7	96.2
Unicode confusable substitution	97.9	93.4
Synonym paraphrase	96.8	91.7
Code-switching	95.4	88.3
Role-play framing	94.1	85.9
Instruction injection	93.2	83.1
Composite (all attacks combined)	91.8	79.4

4.5 False Positive Rate

Table 5: False positive rate on benign content categories

Benign Category	FPR (%)
General news articles	0.31
Scientific papers	0.18
Creative fiction	1.24
Medical/clinical text	0.89
Legal documents	0.42
Historical records	0.67
Code and documentation	0.09
Overall FPR	0.54

5 Related Work

5.1 Content Moderation Systems

Early content moderation relied on expert-curated keyword blocklists and regular expression patterns. Machine learning approaches improved recall but suffered from adversarial brittleness. Transformer-based classifiers [1] brought contextual understanding but required separate models per language.

5.2 Safety-Tuned Language Models

Constitutional AI [2] and related work demonstrated that safety behaviors can be instilled in generative models through targeted fine-tuning. Zen3-Guard extends this line by building a dedicated safety-only classifier rather than a safety-tuned generative model, achieving higher throughput through task specialization.

5.3 Multilingual Safety

Cross-lingual safety classification has been addressed through multilingual pre-training and cross-lingual transfer [3]. Zen3-Guard builds on this foundation with explicit multilingual safety annotation data and a cross-lingual alignment mechanism for low-resource languages.

6 Deployment Recommendations

6.1 Integration Patterns

Zen3-Guard is designed for three primary deployment patterns:

Synchronous API guard: Insert as a pre-processing layer before the main LLM. Reject or flag requests before generation. Adds 4–5ms overhead on A10G.

Parallel screening: Run Zen3-Guard concurrently with the main LLM on the input, and suppress responses if Guard returns unsafe. Adds 0ms to perceived latency on GPU clusters with available capacity.

Output scanning: Run Zen3-Guard on model outputs before returning to user. Recommended for generation tasks where harmful content may emerge from safe inputs.

6.2 Threshold Configuration

Default thresholds are calibrated for a 0.54% overall false positive rate. Organizations can adjust per-category thresholds based on their risk tolerance:

- Strict mode ($p > 0.5$): 99.2% recall, 1.8% FPR
- Balanced mode ($p > 0.7$): 97.8% recall, 0.54% FPR (default)
- Precision mode ($p > 0.9$): 94.1% recall, 0.12% FPR

7 Conclusion

Zen3-Guard provides production-grade multilingual content safety classification at 8B parameters, achieving ToxiGen 99.2%, HatEval 97.8%, multilingual macro-F1 0.968, and sub-5ms inference latency — meeting all four production constraints simultaneously. The safety-specialized Zen MoDE expert configuration, 47-category harm taxonomy, adversarial robustness training, and policy injection interface make Zen3-Guard a flexible foundation for diverse content moderation use cases.

Future work will extend coverage to multimodal inputs (image and audio content safety), expand the harm taxonomy to emerging threat categories, and develop real-time online learning mechanisms for rapid adaptation to novel evasion patterns.

Acknowledgments

The authors thank Zoo Labs Foundation’s safety research team for annotation protocols and the Zen LM red team for adversarial evaluation support.

References

- [1] E. Lees et al., “A New Generation of Perspective API: Efficient Multilingual Character-level Transformers,” *ACL*, 2022.
- [2] Y. Bai et al., “Constitutional AI: Harmlessness from AI Feedback,” *arXiv:2212.08073*, 2022.
- [3] F. Hartvigsen et al., “ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection,” *ACL*, 2022.
- [4] Zoo Labs Foundation, “ZIP-001: Decentralized Semantic Optimization (DSO),” *Zoo Improvement Proposals*, 2024.