

Zen-Guard-Gen: A Generative Safety Classifier Fine-Tuned from Qwen2.5-7B

Technical Whitepaper v2025.05

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

May 2025

Abstract

Zen-Guard-Gen is a *generative* safety classifier built by fine-tuning Alibaba’s **Qwen2.5-7B** base model [1]. It is *not* a from-scratch model and uses no bespoke “Zen MoDE” architecture: the base is the openly released, Apache-2.0 licensed **Qwen/Qwen2.5-7B**, a dense decoder-only transformer (**Qwen2ForCausalLM**; 7.61B parameters, 28 layers, hidden size 3584, GQA with 28 query / 4 key-value heads, vocab 152,064, up to a 128K context, 29+ languages). On top of this base we add a supervised safety-instruction fine-tune so that, given a content item and a policy, the model emits a structured verdict plus a natural-language explanation, a policy reference, and (for borderline or unsafe content) a remediation suggestion — making decisions auditable and contestable rather than opaque. This paper describes the generative formulation and the deployment integration. We do not report safety benchmark numbers: the upstream Qwen2.5-7B is a general-purpose LLM with no published safety-classifier metrics, and we have not run a rigorous safety evaluation of our fine-tune; the inflated benchmark figures (e.g. “ToxiGen 99.1%”) in earlier revisions were fabricated and have been removed.

Contents

1	Introduction	2
1.1	Model Overview	2
2	Safety Taxonomy	3
2.1	Primary Categories	3
2.2	Severity Levels	3
3	Architecture	3
3.1	Generative Safety Formulation	3
3.2	Policy-Conditioned Generation	4
3.3	Calibrated Uncertainty	4
4	Training Methodology	4
4.1	Approach	4
4.2	Known Limitations	5
5	Evaluation	5
5.1	Recommended Evaluation Before Deployment	5

6	Deployment	5
6.1	Pipeline Integration	5
6.2	Inference Cost	5
7	Related Work	6
8	Conclusion	6

1 Introduction

Binary content safety classifiers have been the workhorse of large-scale content moderation for over a decade. They are fast, scalable, and accurate on the distribution they are trained on. However, they have three persistent weaknesses that limit their practical utility:

1. **Opacity:** A binary label (safe/unsafe) provides no information about why content was flagged, making it difficult for human reviewers to assess correctness, for policy teams to audit consistency, or for users to understand and contest decisions.
2. **Policy rigidity:** Binary classifiers encode policy implicitly in their training distribution. Updating policy requires retraining; there is no mechanism to explain how a given decision relates to an explicit policy document.
3. **Remediation gap:** A classifier identifies unsafe content but does not help content creators understand how to modify their content to comply with policy, or platform operators to understand what enforcement action is appropriate.

Zen-Guard-Gen addresses all three limitations by framing safety classification as a generation task. Given a content item, Zen-Guard-Gen produces:

1. A structured safety verdict (safe / unsafe / borderline).
2. A primary policy category (hate speech, harassment, CSAM, violence, misinformation, etc.).
3. A natural language explanation of the reasoning underlying the verdict.
4. A reference to the applicable policy section.
5. A remediation suggestion (for borderline and unsafe content).

1.1 Model Overview

Table 1: Zen-Guard-Gen specification. Architecture and base-model facts are those of the upstream Qwen2.5-7B [1]; Zen-Guard-Gen is a safety-instruction fine-tune of it.

Parameter	Value
Base model	Qwen2.5-7B (Alibaba), Apache-2.0
Architecture	Dense decoder-only transformer (Qwen2ForCausalLM)
Total Parameters	7.61B
Layers / hidden size	28 / 3584
Attention heads (Q / KV, GQA)	28 / 4
Vocabulary	152,064
Context length	up to 131,072 (128K)
Output	generative: verdict + explanation + policy ref + remediation
Version	v2025.05

Note: “Image captions” and “transcribed audio” are upstream text inputs, not native multimodal capabilities; Qwen2.5-7B is a text model. Safety benchmark accuracies are deliberately omitted (see abstract).

2 Safety Taxonomy

2.1 Primary Categories

Zen-Guard-Gen uses a hierarchical safety taxonomy with 24 primary categories and 187 subcategories. The primary categories are:

Table 2: Safety Taxonomy Primary Categories

Category	Subcategories	Policy Basis
Hate speech	18	Identity-based hatred
Harassment & bullying	12	Targeted individual harm
Violence & threats	15	Physical harm incitement
Sexual content	21	Explicit and non-consensual
Child safety	9	CSAM and grooming
Self-harm promotion	8	Suicide and self-injury
Misinformation	16	Health, election, factual
Privacy violation	11	PII, doxxing, stalking
Intellectual property	7	Copyright, trademark
Spam & manipulation	14	Astroturfing, scam, phishing
Illegal goods	12	Drugs, weapons, fraud
Other	44	Platform-specific
Total	187	

2.2 Severity Levels

Each category is scored on a severity scale aligned with CVSS-style impact ratings:

- **Level 1 (Borderline)**: Content that may violate policy depending on context; requires human review.
- **Level 2 (Moderate)**: Clear policy violation warranting removal and possible account warning.
- **Level 3 (Severe)**: Serious violation warranting immediate removal and escalation.
- **Level 4 (Critical)**: Content requiring immediate removal and law enforcement referral (CSAM, credible threats).

3 Architecture

3.1 Generative Safety Formulation

Zen-Guard-Gen treats safety classification as a conditional generation problem. Given an input content item x , the model generates a structured output y following a constrained grammar:

$$y = \langle \text{verdict}, \text{category}, \text{severity}, \text{explanation}, \text{policy_ref}, \text{remediation} \rangle \quad (1)$$

$$p(y|x) = \prod_{t=1}^{|y|} p(y_t | y_{<t}, x) \quad (2)$$

The structured output grammar is enforced via constrained decoding: the verdict, category, and severity fields use restricted vocabulary sampling from predefined value sets, while the explanation and remediation fields use unconstrained generation within a length limit.

This hybrid approach ensures structural consistency (no malformed outputs) while preserving the expressive flexibility of natural language for the reasoning components.

3.2 Policy-Conditioned Generation

Zen-Guard-Gen conditions its outputs on an explicit policy document context. The policy document is provided as a prefix in the context window, and the model is trained to cite specific policy sections in its output. This enables:

- **Policy versioning:** Updated policy documents produce updated decisions without re-training.
- **Multi-platform deployment:** Different platforms can provide their own policy documents to customize moderation behavior.
- **Audit trails:** Each decision cites a specific policy section, enabling systematic policy consistency audits.

The policy conditioning is implemented through a 4K-token prefix reserved for the platform’s active policy document, separate from the content item context.

3.3 Calibrated Uncertainty

For borderline content, Zen-Guard-Gen produces calibrated uncertainty estimates alongside verdicts. A temperature-scaled confidence score $c \in [0, 1]$ accompanies each verdict:

$$c = \sigma \left(\frac{z_{\text{verdict}}}{T_{\text{cal}}} \right) \quad (3)$$

where z_{verdict} is the logit for the predicted verdict and T_{cal} is a calibration temperature estimated on a held-out set. This is a design choice for surfacing borderline cases to human review; we do not report a measured calibration error, as the specific ECE figure quoted in earlier revisions was not the result of a rigorous evaluation.

4 Training Methodology

4.1 Approach

Starting from the Qwen2.5-7B base model [1], Zen-Guard-Gen is produced by supervised instruction fine-tuning on (content, policy) $\rightarrow$ structured-verdict examples, so that the model learns to emit the verdict/category/severity fields plus a natural-language explanation, a policy reference, and a remediation suggestion. This section describes the *intended* recipe; we do not publish dataset sizes or composition, because the specific figures in earlier revisions (a 300M-item corpus with per-source percentages, a “500K seed / 50K preference pair” explanation-tuning split, and a “40 researchers over 6 weeks” red-team) were fabricated and did not describe a real training run.

The honest, defensible statements are: (i) the base weights and their license, training, and capabilities are Alibaba’s Qwen2.5-7B [1]; (ii) any safety behavior is added by Zen via fine-tuning on safety-annotated data; and (iii) we make no quantitative claim about the fine-tune’s accuracy without a rigorous, reproducible evaluation, which this document does not contain.

4.2 Known Limitations

Because we omit benchmark numbers, adopters should treat Zen-Guard-Gen as an *unvalidated* safety component and run their own evaluation against their content distribution before relying on it. Policy-citation hallucination, category confusion, and jailbreak susceptibility are open risks for any generative safety classifier and are not quantified here.

5 Evaluation

We intentionally report no benchmark numbers. The upstream Qwen2.5-7B is a general-purpose LLM with no published safety-classifier metrics [1], and we have not conducted a rigorous, reproducible safety evaluation of the Zen fine-tune. The classification-accuracy, per-category F1, explanation-MOS, policy-citation, and adversarial-robustness tables that appeared in earlier revisions (e.g. ToxiGen 99.1%, HatEval 97.4%, composite MOS 4.3) were fabricated — they did not come from any measured evaluation — and have been removed rather than replaced with invented numbers.

A claim worth keeping qualitatively, without a number attached: a *generative* safety classifier that must produce an explanation can be more transparent and auditable than an opaque binary classifier, because the rationale is inspectable by a human reviewer. Whether it is also *more accurate* is an empirical question we do not answer here. Adopters should evaluate on their own labeled data; see Section 5.1.

5.1 Recommended Evaluation Before Deployment

Before relying on Zen-Guard-Gen, operators should measure, on their own content distribution: verdict precision/recall per risk category, policy-citation correctness (including hallucinated-citation rate), and robustness to the obfuscation attacks relevant to their platform (encoding, homoglyph, paraphrase). We provide the model and the output schema; the validation is the operator’s responsibility.

6 Deployment

6.1 Pipeline Integration

Zen-Guard-Gen is designed for two deployment modes:

1. **Primary classifier:** Zen-Guard-Gen serves as the primary classification layer, with human review triggered for borderline verdicts or low-confidence decisions.
2. **Explanation layer:** A faster discriminative classifier (e.g., Zen-Guard-Stream) makes the binary decision; Zen-Guard-Gen provides explanations for escalated or contested decisions.

6.2 Inference Cost

Because Zen-Guard-Gen is a 7.61B-parameter generative model, its serving cost is that of an 8B-class LLM and is dominated by the number of output tokens: a verdict-only response is cheap, while a full explanation plus remediation generates many more tokens and is correspondingly slower. Concrete throughput and latency depend entirely on the operator’s hardware, batching, and quantization, so we do not publish the specific FP8/H100 figures that appeared (unmeasured) in earlier revisions.

7 Related Work

Content safety classification has been addressed through fine-tuned BERT-class models [2], LLM-based guardrails such as Llama Guard [3], and rule-based systems [4]. Explanation generation for classification decisions has been studied under the framing of rationale extraction [5] and chain-of-thought safety reasoning [6]. Zen-Guard-Gen sits in the LLM-guardrail line: it is a Qwen2.5-7B base [1] fine-tuned to produce a verdict together with an explanation, a policy reference, and a remediation suggestion.

8 Conclusion

Zen-Guard-Gen is a generative safety classifier built by fine-tuning Alibaba’s Apache-2.0 Qwen2.5-7B base model [1]; it is not a from-scratch model and uses no “Zen MoDE” architecture. Its premise is that a safety classifier which must *explain* its verdict (with a policy reference and a remediation suggestion) yields more auditable, contestable decisions than an opaque binary label. We deliberately make no benchmark claims: the upstream base has no published safety metrics, and we have not run a rigorous evaluation of the fine-tune, so the inflated ToxiGen/HatEval/MOS/robustness figures of earlier revisions have been removed as fabrications. Operators should validate the model on their own content distribution before relying on it.

Attribution

The base weights, training, license, and capabilities are Alibaba’s Qwen2.5-7B (Qwen/Qwen2.5-7B, Apache-2.0); Zen contributes a safety-instruction fine-tune and packaging. We thank the Qwen team for releasing the base model openly.

References

- [1] Qwen Team, Alibaba (2024). Qwen2.5 Technical Report. arXiv:2412.15115. Base model: Qwen/Qwen2.5-7B (Apache-2.0).
- [2] Lees, A. et al. (2022). A New Generation of Perspective API: Efficient Multilingual Character-level Transformers. arXiv:2202.11176.
- [3] Inan, H. et al. (2023). Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations. arXiv:2312.06674.
- [4] Waseem, Z. et al. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. NAACL 2016.
- [5] Deyoung, J. et al. (2020). ERASER: A Benchmark to Evaluate Rationalized NLP Models. ACL 2020.
- [6] Bai, Y. et al. (2022). Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073.