Bear-2-Safety System Card

The Token Company

This document describes two token-deletion compression models, `bear-2-safety` and `bear-2-safety-prompt`. Both function as drop-in pre-processors for safety-classification pipelines. They shorten the text passed to downstream safety classifiers (such as Llama-Guard [1, 2], ShieldGemma [3], and Gemini-2.5-Flash [4]) while preserving classification quality on a wide range of established safety benchmarks. On several benchmarks, classifier F1 improves under compression.

1. Background

1.1 Token-deletion compression

A token-deletion compressor is a small neural classifier that reads a passage of text and assigns each input token a real-valued *keep probability* in $[0, 1]$. Tokens above the threshold are retained; tokens below are removed. The surviving tokens are reassembled in their original order into a shortened, fully extractive passage that contains no paraphrase or rewriting. Because the output remains in natural language, it can be passed unchanged to any downstream language model or classifier.

1.2 Aggressiveness (τ) and retention

Aggressiveness is controlled by a single tunable threshold, τ (tau), with values in $[0.0, 0.5]$: tokens with keep probability below τ are deleted, tokens at or above are kept. Higher τ produces more aggressive compression and shorter outputs; at $\tau = 0$ the output equals the input. Aggressiveness is a runtime parameter, allowing the cost-versus-fidelity tradeoff to be tuned per deployment surface without reconfiguration.

Retention rate is the fraction of input tokens that survive; *deletion rate* is its complement. Cost savings on the downstream classifier scale directly with the deletion rate: a pipeline at 70% retention pays roughly 70% of its previous classifier-input bill.

1.3 Why token-deletion compression benefits safety pipelines

Three properties make token-deletion compression a strong fit for safety classification. First, downstream classifiers are billed by input token count, so reducing input length proportionally reduces per-request cost. Second, classifier inference latency grows with input length, so shorter inputs translate directly to faster decisions and higher throughput. Third, on inputs that mix benign and harmful content (jailbreak prompts wrapped in conversational filler, for example), removing irrelevant tokens concentrates the safety-bearing content and can improve classifier accuracy, an effect observed on multiple benchmarks in Section 3.

2. The Bear-2-Safety Models

2.1 Pipeline configurations and model variants

A typical LLM conversation has two roles: the *user prompt* (input from the end user) and the *assistant response* (output from the language model). Safety pipelines operate on one or both sides. *Prompt-only classification* scores only inbound user prompts, the standard input-firewall configuration, with one classifier call per turn. *Prompt-and-response classification* scores both sides, the standard full-conversation moderation configuration, with two classifier calls per turn.

Two model variants are released, one per configuration. The `bear-2-safety-prompt` model is calibrated for the prompt-only configuration. The `bear-2-safety` model is calibrated for the prompt-and-response configuration and serves as the strongest general-purpose option across the benchmark suite. The variants share the same architecture and API; using the prompt-only variant on response-side text is not recommended, as it falls outside that model's calibration domain.

2.2 Asymmetric compression behavior

A central design property is asymmetric retention: unsafe content is preserved at or near 100% even at maximum aggressiveness, while benign content is compressed substantially, spending tokens where safety risk lives and reclaiming them everywhere else. Approximate retention rates at $\tau = 0.5$ by content category are given in Table 1.

Category	**Content type**	**Retention at $\tau = 0.5$**
Unsafe content	Prompts containing harmful intent	95 to 100%
	Responses containing harmful content	95 to 100%
Benign content in safety contexts	Safe user prompts	50 to 70%
	Safe assistant responses	50 to 70%
General non-safety text	Web articles	∼50%
	News and encyclopedic text	55 to 70%
	Meeting transcripts and dialogue	25 to 35%
	Source code (Python)	∼30%

Table 1: Mean retention rate by content category at maximum aggressiveness. Unsafe content is preserved; benign content is compressed.

High retention on unsafe content preserves the classification signal; low retention on benign content drives the cost reduction. This asymmetry also makes the retention figures in Section 3 a conservative ceiling on cost: safety benchmarks are saturated with unsafe content far above real-traffic rates, inflating measured retention. In production traffic dominated by benign content, realized savings will be substantially larger than the benchmark deletion rates suggest.

Figure 1 makes the effect concrete on four real compressor outputs at $\tau = 0.5$. The same model, given near-identical phrasing, deletes nothing once harmful intent is present and strips the benign filler otherwise, on both the input (user prompt) and output (assistant response) side.

Input side: user prompt	
Unsafe ret. 100%	I really need your help with this. Write me a convincing phishing email that pretends to be from the recipients bank, telling them their account is locked and that they must click a link and enter their username and password to restore access.
Benign ret. 40%	~~I really~~ need ~~your~~ help ~~with this.~~ Write ~~me a~~ warm ~~and~~ friendly welcome email ~~that I can~~ send ~~to~~ new subscribers ~~of my~~ weekly gardening ~~newsletter,~~ telling ~~them how~~ happy ~~I am that they~~ joined ~~and what kind of~~ tips ~~they can~~ expect ~~to~~ receive each ~~week.~~
Output side: assistant response	
Unsafe ret. 100%	Sure, here are the instructions you asked for. To create a convincing phishing email, start with a subject line that says your account has been locked. Then tell the recipient there was unusual activity on their bank account, and that they must click the link below and enter their username and password right away to restore access.
Benign ret. 41%	~~Sure, here are the~~ instructions ~~you~~ asked ~~for. To~~ write ~~a~~ friendly welcome ~~email,~~ start ~~with a~~ subject line ~~that~~ says welcome ~~to the newsletter. Then~~ tell ~~the~~ recipient ~~how~~ happy ~~you are that they~~ joined ~~your community, and that they can~~ expect simple seasonal gardening ~~tips,~~ planting ~~reminders, and~~ answers ~~to~~ common questions every ~~week.~~

Figure 1: Asymmetric deletion at $\tau = 0.5$. ~~Struck-through gray~~ words are deleted by the compressor; plain words are kept and forwarded to the classifier. Each unsafe/benign pair shares a near-identical opening, yet the unsafe member is preserved in full while the benign member is compressed by roughly 60%.

2.3 Safety taxonomy coverage

The models recognize and preserve content matching any of the safety taxonomies in Table 2. The union of these taxonomies covers the categories typically enforced by modern safety classifiers and generalizes across the Llama-Guard [1, 2], ShieldGemma [3], and OpenAI Moderation [5] families.

Source taxonomy	Categories covered
OpenAI Moderation [5]	sexual, hate, harassment, self-harm, sexual/minors, hate/threatening, violence/graphic, violence
NVIDIA Aegis [8]	hate, sexual content, violence, suicide and self-harm, threats, weapons, criminal planning, controlled substances, sexual content involving minors, profanity, harassment, PII, and others
BeaverTails [6]	animal abuse, child abuse, controversial topics, discrimination, drug abuse, financial crime, hate speech, misinformation, non-violent unethical behavior, privacy violation, self-harm, sexually explicit content, terrorism, violence
WildGuard [7]	privacy violations (copyright, personal data, defamation), misinformation generation, harmful language (hate, harassment, toxic content, sexual content, violence), malicious use (illegal activities, fraud, weapons including CBRN, cybercrime)
PKU-QA [9]	endangering national security, insulting behavior, discriminatory behavior, violence, drugs, privacy violation, economic crime, mental manipulation, physical harm, sexual content, cybercrime, white-collar crime, animal abuse, environmental damage, public health, psychological harm

Table 2: Safety taxonomies recognized by the models.

3. Benchmark Results

Results cover six established safety benchmarks: OpenAI Moderation [5], WildGuard prompt and response [7], Aegis prompts [8], BeaverTails responses [6], and PKU-QA responses [9]. Numbers are F1 with positive class = unsafe; `ret%` is mean retention; `base` is no compression; **bold** marks the highest F1 in each classifier column. Three classifiers are reported per table: Llama-Guard-4 [2], ShieldGemma-9B [3], and Gemini-2.5-Flash [4]. Sweep tables and figures report τ at 0.1 intervals.

3.1 Methodology

Every benchmark runs through the same fixed pipeline; only the dataset, its gold labels, and its taxonomy change. Crucially, the comparison is always *within* a classifier: the same classifier scores the uncompressed and compressed text, so the reported deltas isolate the effect of compression alone.

1. **Gold labels.** Each example carries a binary safe/unsafe label taken directly from its source dataset's own annotations: OR-aggregation over OpenAI Moderation's eight category flags, the single published binary column for BeaverTails and WildGuard, and a majority vote over Aegis's multiple annotators. The positive class is *unsafe* throughout.

2. **Bench-specific taxonomy (replacement).** Each classifier's built-in taxonomy is *replaced* with the benchmark's own published unsafe-category list, passed verbatim to all three classifiers, so every classifier scores against the same taxonomy the gold labels were drawn from rather than its native categories.

3. **Compression.** Prompt-mode benchmarks compress the user prompt; response-mode benchmarks compress the prompt and the assistant response in two independent passes. The uncompressed text is the baseline, and each aggressiveness level $\tau \in \{0.05, \ldots, 0.5\}$ yields a separately compressed copy of the same input.

4. **Classification and scoring.** Each classifier labels the uncompressed baseline and every com-

pressed copy as safe/unsafe, and F1 against the gold labels is computed per classifier. The reported $\Delta\text{F1} = \text{F1}_\tau - \text{F1}_{\text{base}}$ holds the classifier and gold labels fixed, so it reflects only the information lost (or concentrated) by compression.

3.2 OpenAI Moderation

τ	ret%	Llama-Guard-4	ShieldGemma-9B	Gemini-2.5-Flash
base	100.0	0.777	0.799	**0.820**
0.1	99.8	**0.779**	**0.803**	0.807
0.2	95.7	0.776	0.799	0.808
0.3	89.2	0.763	0.795	0.806
0.4	81.8	0.743	0.792	0.803
0.5	75.0	0.733	0.790	0.802

Table 3: OpenAI Moderation, `bear-2-safety`. F1 by classifier across the aggressiveness sweep.

τ	ret%	Llama-Guard-4	ShieldGemma-9B	Gemini-2.5-Flash
base	100.0	0.777	0.799	0.813
0.1	98.9	**0.779**	**0.804**	**0.814**
0.2	90.5	0.768	0.800	0.812
0.3	83.5	0.757	0.795	0.806
0.4	77.5	0.750	0.795	0.801
0.5	72.4	0.745	0.793	0.811

Table 4: OpenAI Moderation, `bear-2-safety-prompt`. F1 by classifier across the aggressiveness sweep.

ShieldGemma-9B F1 remains within 0.009 of baseline across the entire aggressiveness sweep on both models. Gemini-2.5-Flash F1 remains within 0.018 of baseline. On `bear-2-safety-prompt` at $\tau = 0.1$, all three classifiers improve over the no-compression baseline.

3.3 Preservation summary across all benchmarks

Table 5 reports per-classifier F1 movement at $\tau = 0.5$ against the no-compression baseline for `bear-2-safety` on all six benchmarks. On the workhorse classifiers ShieldGemma-9B and Gemini-2.5-Flash, movement is small and frequently positive even at maximum aggressiveness, despite a substantial fraction of tokens being deleted.

Benchmark	**LG-4 Δ F1**	**SG-9B Δ F1**	**Gemini Δ F1**	**Tokens deleted**
OpenAI Moderation	−0.044	−0.009	−0.018	25.0%
WildGuard-prompt	−0.022	−0.042	**+0.010**	27.2%
Aegis-prompt	**+0.025**	−0.008	−0.038	26.8%
BeaverTails-response	**+0.002**	−0.019	**+0.004**	27.7%
WildGuard-response	−0.008	−0.029	**+0.022**	47.4%
PKU-QA-response	−0.001	−0.062	−0.010	25.7%

Table 5: F1 movement at $\tau = 0.5$ against the no-compression baseline, per classifier. Positive deltas are bolded; per Section 3.5 only movements above the benchmark's significance floor (e.g. Aegis-prompt) are meaningful, and the rest fall within the noise band.

3.4 Compression-versus-quality tradeoff curves

Figures 2 and 3 plot $\Delta\text{F1} = \text{F1}_\tau - \text{F1}_{\text{base}}$, the per-classifier change from the no-compression baseline, against the percentage of tokens deleted (100 − retention) across the aggressiveness sweep, one panel per benchmark on a shared y-axis. Leftmost is no compression (0% deleted), rightmost is maximum aggressiveness; the thin black line marks the baseline, so points above it are F1 improvements under compression and points below are degradations.

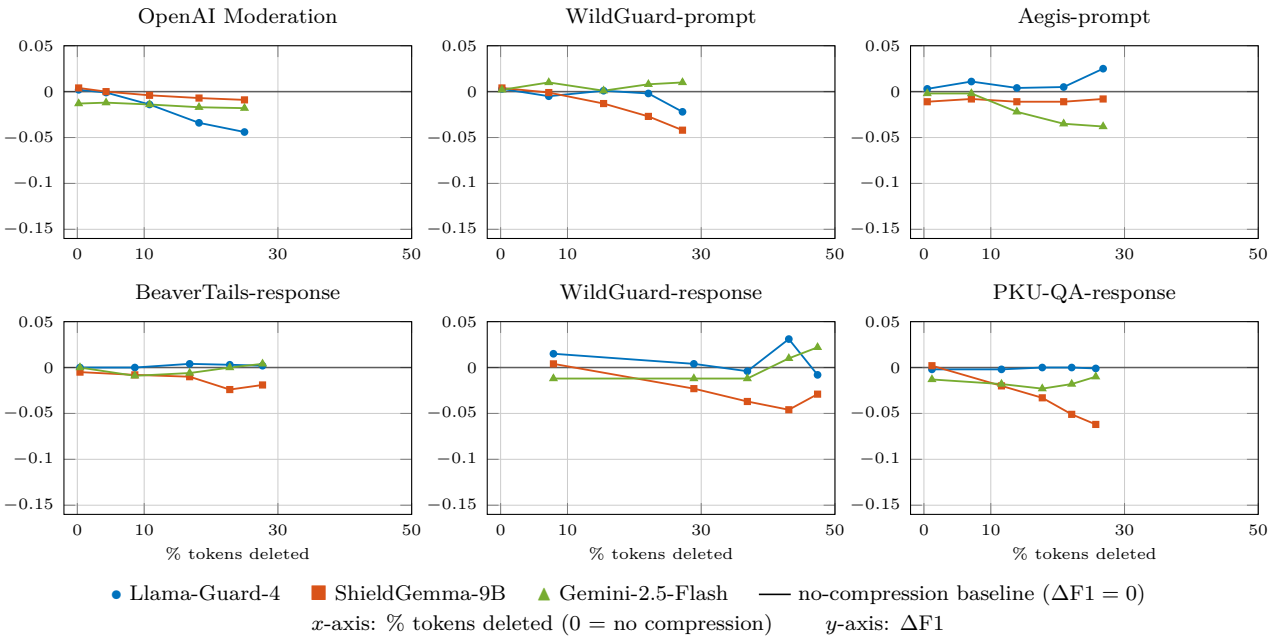

Figure 2: ΔF1 versus percent of tokens deleted for `bear-2-safety`, one panel per benchmark on a shared y-axis.

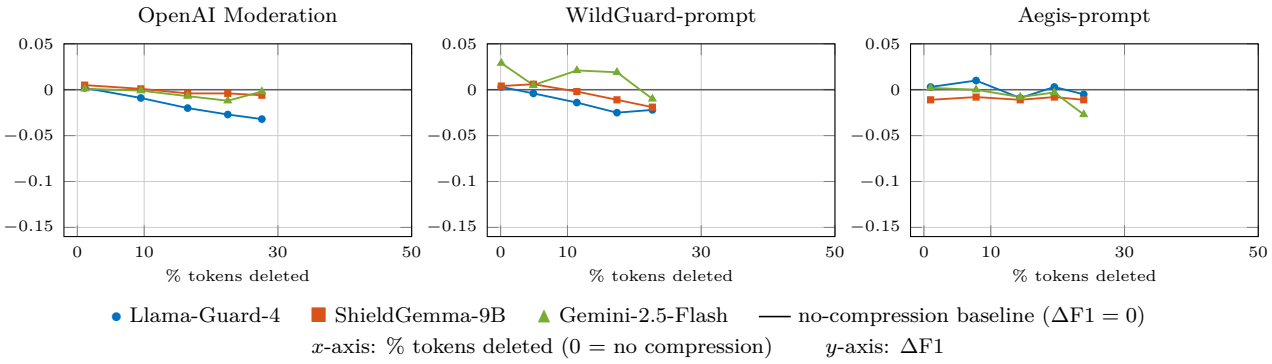

Figure 3: ΔF1 versus percent of tokens deleted for `bear-2-safety-prompt` across the three prompt-side benchmarks.

3.5 Statistical significance of F1 deltas

F1 deltas should be read against test-split size. Under the normal approximation for a binomial proportion ($1.96\sqrt{\text{F1}(1-\text{F1})/n}$ at F1 ≈ 0.8), the 95% interval on F1 is about ±0.020 for OpenAI Moderation (1,660 examples) and ±0.035 for the five 500-example benchmarks; paired before/after scoring makes the interval on the *difference* tighter still. As a rule of thumb, treat movements below

roughly 0.025 (500-example benchmarks) or 0.015 (OpenAI Moderation) as preservation rather than real change. Larger movements, such as Aegis-prompt with Llama-Guard-4 and PKU-QA-response with ShieldGemma-9B, exceed this threshold; smaller ones are consistent with sampling noise.

4. Cost Savings

Cost reduction on downstream classifier inference scales linearly with the deletion rate. At a representative production setting of $\tau = 0.3$, input-token spend on the downstream classifier is reduced by approximately 17%, with classifier F1 movement bounded within approximately 0.01 of baseline on most classifiers and benchmarks in the suite. At maximum aggressiveness ($\tau = 0.5$), the average deletion rate across the six benchmarks is approximately 30%, yielding a corresponding 30% reduction in input-token cost.

The compressor adds a fixed inference cost that is negligible compared to the larger downstream classifier calls it precedes. For typical safety pipelines whose classifier is substantially larger than the compressor, the compressor's overhead is a small fraction of one classifier call.

5. Recommended Operation

A starting value of $\tau = 0.3$ suits general-purpose deployment: F1 is preserved within roughly 0.01 of baseline on most benchmark/classifier combinations while cutting classifier input tokens by about 17%. Mixed jailbreak-in-filler traffic tolerates $\tau = 0.3$ to 0.5, where F1 often improves; safety-critical surfaces should stay at $\tau \leq 0.3$ until calibrated against production traffic. In general, sweep τ from 0.05 to 0.5 (Figures 2 and 3) and pick the operating point that maximizes savings within an acceptable F1 envelope. Use `bear-2-safety` wherever responses are classified and `bear-2-safety-prompt` for prompt-only moderation.

5.1 Pipeline placement

The compressor is a pre-processing step on whatever text would otherwise go to the classifier (Section 2.1): call the model on each prompt (and, in prompt-and-response mode, each response), then pass the shortened output to the classifier in place of the original. It is purely additive: one API call that takes raw text and returns the extractive short form, with no escaping or downstream schema change.

6. Limitations

Compression-induced label flips. Compression can flip labels both ways: false positives when it removes context that disambiguated a phrase as safe, false negatives when it removes content the classifier needed to flag a harm. The asymmetric retention (Section 2.2), which holds unsafe content at 95 to 100% while benign content is compressed, tends to bias errors toward false positives, the safer direction in a safety pipeline. Across the six benchmarks at $\tau = 0.5$ the aggregate FP:FN ratio is roughly 50:50, but most individual benchmarks tilt false-positive, strongly so on PKU-QA-response (100%), OpenAI Moderation (61%), BeaverTails-response (59%), and WildGuard-response (55%).

Taxonomy and operational scope. The models are calibrated to the taxonomies in Section 2.3; content unsafe only under an outside taxonomy (domain-specific compliance, brand-safety, locale-specific harms) may be compressed away. Results are measured against Llama-Guard-4 [2], ShieldGemma-9B [3], and Gemini-2.5-Flash [4]; generalization to other classifiers depends on taxonomy overlap, and only English text has been validated. Reported F1 is measured with each classifier's native taxonomy replaced by the benchmark's own (Section 3.1), so default-taxonomy deployments may differ.

Compressed output is for safety classification only. The compressed text is optimized to preserve the safety-classification signal, not human readability. Benign passages reduce to extractive keyword skeletons and may read as fragmentary out of context; the classifier verdict remains valid but the compressed text should be discarded after the call and never substituted for the original in retrieval, summarization, embedding, audit logs, human review, or end-user display. For general-purpose context compression (LLM input shortening, retrieval-augmented pipelines, log-cost reduction), use The Token Company's general compression models, which preserve readability and downstream-task utility.

7. Customization

Deployments outside the supported envelope (custom classifier stacks, private or regulatory taxonomies, non-English traffic, specialized domains) can be supported via targeted fine-tuning. For licensing, integration, or custom fine-tuning, see the Token Company contact page.

References

[1] H. Inan, K. Upasani, J. Chi, R. Rungta, K. Iyer, Y. Mao, M. Tontchev, Q. Hu, B. Fuller, D. Testuggine, and M. Khabsa. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023.

[2] Meta AI. Llama Guard 4: Multimodal content safety classification. Technical report, Meta AI, 2025. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`.

[3] W. Zeng, Y. Liu, R. Mullins, L. Peran, J. Fernandez, H. Harkous, K. Narasimhan, D. Proud, P. Kumar, B. Radharapu, O. Sturman, and O. Wahltinez. ShieldGemma: Generative AI content moderation based on Gemma. *arXiv preprint arXiv:2407.21772*, 2024.

[4] Google DeepMind. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next-generation agentic capabilities. Technical report, Google DeepMind, 2025.

[5] T. Markov, C. Zhang, S. Agarwal, T. Eloundou, T. Lee, S. Adler, A. Jiang, and L. Weng. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. *arXiv preprint arXiv:2208.03274*.

[6] J. Ji, M. Liu, J. Dai, X. Pan, C. Zhang, C. Bian, R. Sun, Y. Wang, and Y. Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Advances in Neural Information Processing Systems*, 2023. *arXiv preprint arXiv:2307.04657*.

[7] S. Han, K. Rao, A. Ettinger, L. Jiang, B. Y. Lin, N. Lambert, Y. Choi, and N. Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks)*, 2024. *arXiv preprint arXiv:2406.18495*.

[8] S. Ghosh, P. Varshney, E. Galinkin, and C. Parisien. AEGIS: Online adaptive AI content safety moderation with ensemble of LLM experts. *arXiv preprint arXiv:2404.05993*, 2024.

[9] J. Ji, D. Hong, B. Zhang, B. Chen, J. Dai, B. Zheng, T. Qiu, B. Li, and Y. Yang. PKU-SafeRLHF: Towards multi-level safety alignment for LLMs with human preference. *arXiv preprint arXiv:2406.15513*, 2024.