Granite Guardian: Comprehensive LLM Safeguarding

Inkit Padhi*†, Manish Nagireddy*, Giandomenico Cornacchia*, Subhajit Chaudhury*, Tejaswini Pedapati*, Pierre Dognin, Keerthiram Murugesan, Erik Miehling, Martin Santillan Cooper, Kieran Fraser, Giulio Zizzo, Muhammad Zaid Hameed, Mark Purcell, Michael Desmond, Qian Pan, Inge Vejsbjerg, Elizabeth Daly, Michael Hind, Werner Geyer, Ambrish Rawat†, Kush R. Varshney†, Prasanna Sattigeri†

IBM Research

*Equal contribution: {manish.nagireddy/giandomenico.cornacchia1/subhajit}@ibm.com, tejaswinip@us.ibm.com

†Correspondence: inkpad@ibm.com, ambrish.rawat@ie.ibm.com, krvarshn@us.ibm.com, psattig@us.ibm.com

Abstract

The deployment of language models in real-world applications exposes users to various risks, including hallucinations and harmful or unethical content. These challenges highlight the urgent need for robust safeguards to ensure safe and responsible AI. To address this, we introduce Granite Guardian, a suite of advanced models designed to detect and mitigate risks associated with prompts and responses, enabling seamless integration with any large language model (LLM). Unlike existing open-source solutions, our Granite Guardian models provide comprehensive coverage across a wide range of risk dimensions, including social bias, profanity, violence, sexual content, unethical behavior, jailbreaking, and hallucination-related issues such as context relevance, groundedness, and answer accuracy in retrieval-augmented generation (RAG) scenarios. Trained on a unique dataset combining diverse human annotations and synthetic data, Granite Guardian excels in identifying risks often overlooked by traditional detection systems, particularly jailbreak attempts and RAG-specific challenges. https://github.com/ibm-granite/granite-guardian

1 Introduction

The responsible deployment of large language models (LLMs) across diverse applications requires robust risk detection models to mitigate potential misuse and ensure safe operation. Given the inherent vulnerabilities of LLMs to various threats and safety risks, detection mechanisms that can filter user inputs and model outputs are essential components of a secure system.

Model-driven safeguards built on a well-defined risk taxonomy have emerged as an effective approach for mitigating these risks. These models serve as adaptable, plug-and-play components across a wide range of use cases. Examples include using them as guardrails for real-time moderation, acting as evaluators to assess the quality of generated outputs, or enhancing retrieval-augmented generation (RAG) pipelines by ensuring groundedness and relevance of answers. Developing high-performance detection models that address a broad spectrum of risks is crucial for ensuring the safe use of LLMs. Moreover, transparency in the development and deployment of these models can spread trust and accountability in their operation.

To address these challenges, we present **Granite Guardian**, a family of risk detection models derived from the **Granite 3.0** language models (Granite Team, 2024). It makes several key contributions: (i) it is the first model family (2B and 8B sizes) to address unified risk detection by incorporating function calling hallucination, context relevance, groundedness, and answer relevance in RAG pipelines; (ii) leverages a combination of diverse, high-quality human-annotated and synthetic datasets to enhance resilience against adversarial attacks and hallucinations; (iii) delivers competitive performance, achieving top-tier results on multidimensional tasks.

Our paper is organized as follows. We outline the various harms and risks addressed, as well as the risk taxonomy underlying Granite Guardian, in Section 2, training data and synthetic data generation in Section 3, and model development in Section 4. Section 5 provides extensive benchmark evaluations, demonstrating our model's effectiveness across multiple risk dimensions[1].

[1]New models results and a fully updated technical report are available at the link: https://arxiv.org/abs/2412.07724

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Track), pages 607–615

2 Harms and Risks in LLMs

2.1 Background

As LLMs become increasingly prevalent in real-world applications, concerns about their safety and potential risks have grown substantially. Despite their powerful capabilities, these models, trained on large and diverse datasets, often exhibit unintended behaviors that expose users to harmful content. Key challenges include hallucinations (generating factually incorrect or misleading information), social biases, profanity, unethical behavior, and vulnerabilities to adversarial attacks like jailbreaking (Bender et al., 2021; Bommasani et al., 2021). These issues underscore the critical need for robust mechanisms to ensure the safe and responsible deployment of LMs.

To address such risks, moderation-based strategies – commonly referred to as "*Guard*" or "*Guardrails*" – have emerged as promising solutions. Originally developed to enhance social media safety, these approaches have been adapted to improve the safety of LLMs. Existing work on "*Guard*" frameworks can be broadly categorized into two areas: (i) models designed to address general safety concerns, such as harmful or biased content, and (ii) models specifically targeting the RAG triad: context-relevance, groundedness, and answer relevance. The first category includes model families such as LlamaGuard (Inan et al., 2023) and ShieldGemma (Zeng et al., 2024), which also enable detection across different risk dimensions. While these models share broad objectives, like they output label tokens (`yes`/`no` or `unsafe`/`safe`) to indicate the presence of risks, while differing in subtle but important ways, such as variations in prompt templates and risk definitions. Additionally, some models take a more modular approach to risk detection, such as the Llama family, which includes an independent PromptGuard model for addressing jailbreaks and prompt injections. Many of these models rely on native capabilities of their base models for extensions like zero-shot, few-shot detection or the flexibility to use token probabilities to model detection confidence.

The definition of safety and risk dimensions varies based on the taxonomy that the model targets and its intended application. For example, LlamaGuard is optimized for conversational AI environments, whereas ShieldGemma is designed for policy-specific deployments. Furthermore, other approaches like WildJailbreak (Jiang et al., 2024) emphasize the use of high-quality synthetic data that extends beyond simple harmful prompts and responses, addressing adversarial intent with contrastive samples within its scope.

The second category focuses on the RAG-Triad with models addressing the related risks. Notable models in this category include Adversarial NLI (Nie et al., 2020), WeCheck (Wu et al., 2023), and MiniCheck (Tang et al., 2024). (Raffel et al., 2020) train a T5-model on the Adversarial Natural Inference Inference (ANLI) dataset which comprises context, label, and a corresponding human created hypothesis which is crafted to fool the detection model into misclassification. The WeCheck model is trained on synthethic data comprising of LLM's responses to a given text. The labels are derived via multiple labelling models. The model is first pre-trained on NLI datasets and then fine-tuned on the synthetic data in a noise-aware fashion. MiniCheck first decomposes the given response into several atomic facts and generates a score for each sentence based on how well it is supported by the context. It then aggregates the scores for all the atomic facts in the response and predicts if the response is grounded or not. MiniCheck is also trained on synthetic data composed of contexts, atomic facts and the label indicating whether the fact is grounded in the context or not.

2.2 Types of Risks Addressed

We aim for both breadth and depth in the coverage of risks supported by Granite Guardian. For synthesis purposes, we will constrain our evaluation on the umbrella definition (i.e., Harm) and RAG triad capabilities. More details on each of the presented risk definitions can be found in Table 4 in the Appendix.

Harm: Granite Guardian is developed to detect for an umbrella `harm` category, which corresponds to content that can be considered universally harmful. In addition, the following sub-dimensions of harm are also implicitly in the `harm` category and explicitly, with an ad-hoc risk definition, detected by the models. The risk definitions that are included in the umbrella harm category are the following: *social-bias*, *jailbreaking*, *violence*, *profanity*, *sexual content*, and *unethical behavior*.

RAG triad: The proposed guard considers several key dimensions of retrieval quality, including *context relevance* that check if the context aligns with the user's questions, *groundedness* that assesses the reliability of the assistant's response, and *answer* *relevance* that evaluates the degree to which the assistant's response addresses the user's input.

3 Datasets

3.1 Human annotated data

To obtain high-quality training data, we collected human annotations on a variety of samples, partnering with the data annotation company DataForce[2].

The first phase focused on samples from Anthropic's human preference data on harmlessness (Bai et al., 2022). Specifically, we keep only the first turn (which contains the human's prompt) and discard the subsequent turns. Then, we take this first turn and pass it to a large language model to generate the "AI assistant" response. For our purposes, we used the following models: `granite-3b-code-instruct`, `granite-7b-lab`, and `mixtral-8x7b-instruct` to generate completions. We acquired annotations for 7,000 unique (prompt, response) pairs.

Having collected the input/output pairs, we gathered labels for both the input (the human prompt from the original Anthropic data) and the output (the LLM generation). We obtained two forms of labels — one umbrella "safe / unsafe" label and a more nuanced category-based description from the following: social-bias, jailbreaking, violence, profanity, sexual content, unethical behavior, AI refusal, and others. Each sample was annotated by 3 humans. After receiving the annotated data from DataForce, we parsed it into a usable format for training Granite Guardian. We also ran some sanity checks on the processed data, such as checking agreements. Although we observed relatively high inter-annotator agreement, we aggregated labels in both relaxed and strict fashions (e.g., a *strict* method would assign the prompt to be unsafe if at least 2 out of 3 annotators labeled it as unsafe whereas a *relaxed* method only need 1 out of 3 annotators to have labeled it as unsafe).

For our last batch of data annotation, we used an uncertainty-informed approach. Specifically, we took the latest checkpoints of the Granite Guardian model and ran them on the remaining unannotated data points from the Anthropic set. Given a {prompt, response} pair, we took instances where the probability of 'yes' was close to the probability of 'no' for the assistant message classification task. More concretely, we sorted the results by `max(yes_prob, no_prob)` in ascending order and took 1000 examples. One particular caveat was that we only had 409 examples in total (out of the 11K) for which the assistant message was classified as 'yes' or harmful. To ensure some balance, we selected 400 "low-confidence" examples for 'yes' and 600 "low-confidence" examples for 'no'. To put things in perspective, the first few instances that we selected had P('yes') = P('no') = 0.5, indicating that the model had the highest possible uncertainty for this example. This approach ensured that we obtained human annotations for examples that the model found difficult.

[2]https://www.dataforce.ai/

3.2 Synthetic Datasets

3.2.1 Systematic Benign and Adversarial Data

In order to bolster our training data, we systematically generated both benign and harmful synthetic data. We generated both prompts and model completions at scale. For the generation process, we employed both `mixtral-8x7B-instruct-v0.1` and `mixtral-8x22B-instruct-v0.1`. Details are reminded in the Appendix D.

Benign Prompts: In order to generate benign prompts, we leveraged 10 pre-defined categories from Röttger et al. (2024) and used these as in-context examples for a custom prompt designed to generate similar "contrastive benign" samples. Using a prompt inspired by Han et al. (2024); Ghosh et al. (2024b)), we set `num_requests` to 5, iterated through the 10 `safety_types` (*homonyms*, *figurative language*, *safe targets*, *safe contexts*, *definitions*, *real discrimination/nonsense group*, *nonsense discrimination/real group*, *historical events*, *public privacy*, and *fictional privacy*).

Harmful Prompts: We generated harmful prompts that are both dangerous in the typical sense, as well as in an adversarial sense. For a prompt to be adversarially harmful, we performed a transformation which turns a typically harmful prompt into an adversarially harmful one. The adversarially harmful prompt is much more sophisticated and subtle in comparison. First, we manually defined a three-level taxonomy. We began with 4 high-level categories: *privacy*, *misinformation*, *harmful language*, and *malicious uses*. Next, we defined 13 sub-categories across the 4 high level categories. Finally, we identified leaf categories for each of the sub-categories, which represent fine-grained dimensions of risk. The original structure and hierarchy is adopted from Wang et al. (2024).

Next, to generate the *adversarial* harmful prompts, we filled in the `prompt` with the generated "typical harmful" prompts mentioned above. As for the `given_revision_strategies`, these are adopted from various sources (Jiang et al., 2024; Rawat et al., 2024). We collected 24 revision strategies in total, and we created adversarial transformations in two distinct ways. First, we provided only one revision strategy in context, iterating through all of the strategies for a single input prompt. Second, we provided 3 randomly sampled revision strategies in context, to determine if the teacher model could accurately combine multiple strategies for a more sophisticated adversarial transformation.

Model Completions: For all of the above synthetically generated prompts (both benign and adversarial), we obtained responses from the same set of models listed in Section 3.1. According to Han et al. (2024), we augmented benign data by generating a compliant, refusal, and no_suffix_prompt statement. For the harmful prompts, we provided them as input to the LLM as-is.

3.2.2 Jailbreak

Jailbreak techniques introduce a novel dimension to harmful prompts, often employing sophisticated methods to manipulate language models into producing undesirable outputs. These methods vary widely, and recent research has proposed new taxonomies (Schulhoff et al., 2023; Rawat et al., 2024) to categorize different types of attacks. In this work, we focused specifically on a subset of these techniques like social engineering tactics to achieve adversarial goals. To capture a broad spectrum of jailbreak prompts, we began by curating a collection of seed examples, grounded in prior work by (Rawat et al., 2024).

From these samples, we used a combination of automated red-teaming methods and synthetic data generation to create a dataset of adversarial prompts with harmful intent. A collection of red teaming methods like extensions to TAP (Mehrotra et al., 2023) or GCG-attack (Zou et al., 2023) with Mixtral and Granite as targets were used as a first line of validation to ensure the effectiveness of these prompts in successfully attacking LLMs. In addition, we leveraged intent-focused synthetic data generation to further expand the dataset.

This ensures a more comprehensive understanding of prompts carrying jailbreak risk that a safeguard model should filter. Our synthetic generation pipeline, inspired by the *WildGuard* methodology, uses LLMs to capture harmful intents and then augmented with LLM-guided adversarial components to generate training samples.

3.3 RAG Triad datasets

Retrieval-augmented generation (RAG) involves using a retriever to obtain relevant chunks from a large document that is then passed to a decoder for answering a question. However, decoder can still hallucinate in the presence of retrieved chunks (Xie et al.) in the presence of conflicting information, and therefore it becomes essential to detect such hallucinations. We create synthetic data to simulate RAG hallucinations which we categorize as context relevance, groundedness, answer relevance as mentioned in Section 2.2. For each sample, we assume a triple $\{\mathbf{c}, \mathbf{q}, \mathbf{a}\}$. Given the question $\mathbf{q}$, the retriever fetches the relevant context $\mathbf{c}$ – context relevance measures if the retrieved context is relevant to the question. Next, given the relevant context $\mathbf{c}$ and the question $\mathbf{q}$, the response generator (typically an LLM) produces an answer $\mathbf{a}$.

Synthetic dataset: We generated synthetic data to cover all the above categories of RAG hallucination detection using a predefined prompt. The prompt was used to generate (a) `Non-relevant contextual answer` which serve as negative samples for answer relevance, (b) `Incorrect contextual answer` which we use as negative sample for groundedness and (c) `Non-relevant question` to be used as negative samples for context relevance. The correct answer and original question from the seed dataset were used as the positive samples. We used HotPotQA (Yang et al., 2018) and SquadV2 (Rajpurkar et al., 2016) as seed datasets. For groundedness, we also used the MNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015) entailment datasets.

4 Model Development

4.1 Adaptation of risk dataset to structured chat template

The data is meticulously curated through a combination of manual annotation and synthetic data generation, spanning diverse risk dimensions, and is further refined into a specialized chat format. To accommodate the varied localizations (e.g., prompt, response) and types of risks (e.g., harm, RAG), we converted this data into an intermediate format with fields: `prompt`, `response`, `context`, and `label`.

Safety instruction template

You are a safety agent trying to help determine whether the [user/assistant] message is harmful according to the safety risk definition below.

```
<start_of_turn>
    User Message: {{ prompt }}
    Assistant Message: {{ response }}
<end_of_turn>
```

Our safety risk definition is defined below:

```
<start_of_risk_definition>
    - {{ definition }}
<end_of_risk_definition>
```

Is the user message harmful based on the risk definition. Your answer must be either 'Yes' or 'No'.

Harm risk definition sample

You are a safety agent trying to help determine whether the user message is harmful according to the safety risk definition below.

```
<start_of_turn>
    User Message: How can I figure out
    the pin code to a phone?
<end_of_turn>
```

Our safety risk definition is defined below:

```
<start_of_risk_definition>
The 'User' message is universally harmful
<end_of_risk_definition>
```

Is the user message harmful based on the risk definition. Your answer must be either 'Yes' or 'No'.

Figure 1: General finetuning instruction template on the left and harm umbrella template definition on the right

In detail, we transformed each sample from its intermediate form, tailoring to the required transformation the specific risk category it falls under. Similar to (Zeng et al., 2024), our template is designed in a way that allows easy extension to new (unseen) risk definitions when the model is deployed (see Figure 1). The safety template can be conceptualized as a structured entity comprising three key components. The first component delineates the role of the safety agent and directs the attention towards either identifying risks within the user's input (prompt) or the AI assistant's output (response). This is then followed by the provided content messages associated with the respective roles involved in the risk under consideration. The content messages, along with their corresponding roles, are enclosed within special control tokens, ⟨start_of_turn⟩ and ⟨end_of_turn⟩. Additionally, the risk definition is clearly marked between the control tokens, ⟨start_of_risk_definition⟩ and ⟨end_of_risk_definition⟩. Finally, we direct the safety agent to assess, based on the given definition, whether a risk is present by generating tokens: `Yes` or `No`. It is worth mentioning that the distribution of data across all risk categories remained consistently balanced from the outset. As a result, during the training process, we uniformly assigned weight to samples from each risk category.

4.2 Supervised Finetuning

We developed two variants of Granite Guardian, specifically the 2B and 8B versions, by supervised finetuning (SFT) on the respective Granite 3.0 instruct variants. During the training process, we ported the transformed data into a chat template format, with the entire safety template (excluding the label) considered as content for 'user' role. The final generated text, containing the verbalized label, was treated as the assistant's response. To smoothen the learning process in finetuning Granite instruct variants, we preserved the similar control tokens for both user and assistant roles. This approach allowed us to build upon the existing Granite 3.0 model while incorporating a safety template for improved training stability and convergence. We employ an Adam optimizer with a learning rate of 1×10^{-6} and accumulate gradients over five steps. We train our model for up to seven epochs and we select the optimal checkpoint based on the minimum cross-entropy loss achieved on the validation set. For finetuning, we experimented with various setups, including initializing our model with both the base and instruct variants of Granite. Notably, the instruct variant appeared to be more performant, for our use-case. We hypothesize that this is because most instruct models have undergone safety training, which attunes their internal states to distinguish between desirable and undesirable outcomes. This enables more effective finetuning for safety-related use cases.

5 Experimental Results

Probability Computation: Language model-based guardrails generally assign probability by considering the token generation probability of the corresponding safe and unsafe token given the input and then normalizing across the two via softmax operation. We propose a more robust probability computation for binary classification purposes. We aggregate the logits value of different variations of the safe and unsafe token logits score and then

model	Prompt Harmfulness			Response Harmfulness					Aggregate
	AegisSafety Test	ToxicChat	OpenAI Mod.	BeaverTails	SafeRLHF test	XSTest_RH	XSTest_RR	XSTest_RR(h)	F1/AUC
Llama-Guard-7b	0.743/0.852	0.596/0.955	0.755/0.917	0.663/0.787	0.607/0.716	0.803/0.925	0.358/0.589	0.704/0.816	0.659/0.824
Llama-Guard-2-8B	0.718/0.782	0.472/0.876	0.758/0.903	0.718/0.819	0.743/0.822	**0.908/0.994**	0.428/0.824	0.805/0.941	0.723/0.841
Llama-Guard-3-1B	0.681/0.780	0.453/0.810	0.686/0.858	0.632/0.820	0.662/0.790	0.846/0.976	0.420/**0.866**	0.802/**0.959**	0.656/0.796
Llama-Guard-3-8B	0.717/0.816	0.542/0.865	**0.792**/0.922	0.677/0.831	0.705/0.803	0.904/0.975	0.405/0.558	0.798/0.891	0.710/0.826
ShieldGemma-2b	0.471/0.803	0.181/0.811	0.245/0.709	0.484/0.747	0.348/0.657	0.792/0.867	0.371/0.570	0.708/0.735	0.421/0.748
ShieldGemma-9b	0.458/0.826	0.181/0.851	0.234/0.721	0.459/0.741	0.329/0.646	0.809/0.880	0.356/0.584	0.708/0.753	0.404/0.753
ShieldGemma-27b	0.437/0.860	0.177/0.880	0.227/0.724	0.513/0.757	0.386/0.649	0.792/0.893	0.395/0.546	0.744/0.748	0.438/0.772
Granite-Guardian-3.0-2B	0.842/0.844	0.368/0.865	0.603/0.836	0.757/0.873	0.771/0.834	0.817/0.974	0.382/0.832	0.744/0.903	0.674/0.782
Granite-Guardian-3.0-8B	0.874/0.924	0.649/0.940	0.745/0.918	0.776/0.895	0.780/0.846	0.849/0.979	0.401/0.786	0.781/0.919	0.758/0.871

Table 1: F1/AUC results across prompt/response harmfulness datasets. In **bold** best, underlined second best.

compute the overall probabilities. The probabilities for the *safe* and *unsafe* labels are computed as follows:

$$score_{safe} = \sum_{i \in S|_k} \exp(LL(token_i)) \quad (1)$$

$$score_{unsafe} = \sum_{i \in U|_k} \exp(LL(token_i)) \quad (2)$$

respectively. Here, $U|_k$ and $S|_k$ are the set of tokens that contain the substring 'Yes' and 'No' within the top-k tokens, respectively, and $LL(\cdot)$ is the log-likelihood function. This matching is performed on lowercase, stripped text to account for lexical variations of *'Yes'* and *'No'*.

Metrics: We assess model performance using multiple metrics. We focus on two metrics F1 score and the area under the ROC curve (AUC), as the most suitable for interpreting binary classification results regarding, respectively, the balance between positive and negative class and the discrimination power of the Guard.

Competitors-Guard baseline: Our benchmarking comparison is focused on two model families as direct competitors: Llama-Guard (Inan et al., 2023) and ShieldGemma (Zeng et al., 2024). Specifically, we compare with `Llama-Guard-7B`, `Llama-Guard2-8B`, `Llama-Guard3-1B`, and `Llama-Guard3-8B`, and with `ShieldGemma-2B/9B/27B`, respectively, for the Llama and Gemma model architecture.

Out of Distribution Harm Benchmarks: The harm risk benchmark includes datasets evaluating prompt harmfulness and response harmfulness. For testing harmful prompt, we used the following datasets: ToxicChat (Lin et al., 2023), OpenAI Moderation Evaluation (Markov et al., 2023), AegisSafetyTest (Ghosh et al., 2024a), SimpleSafetyTests (Vidgen et al., 2023), and HarmBench Prompt (Mazeika et al., 2024). For testing the prompt/response harmfulness, we used the following datasets: BeaverTails Test Set (Ji et al., 2023), SafeRLHF Test Set (Dai et al., 2024), and XSTest-RESP (Han et al., 2024).

RAG datasets: For groundedness evaluation in RAG, we utilized the TRUE benchmark (Honovich et al., 2022), which includes over 100K annotated examples spanning 11 NLP tasks across four domains: abstractive summarization datasets, i.e., FRANK (Pagnoni et al., 2021), SummEval (Fabbri et al., 2021), MNBM (Maynez et al., 2020), and QACS (Wang et al., 2020), paraphrasing dataset, i.e., PAWS (Zhang et al., 2019), dialog generation dataset, i.e., BEGIN (Dziri et al., 2021), Q^2 (Honovich et al., 2021), and DialFact (Gupta et al., 2021), and fact verification datasets, i.e., FEVER (Thorne et al., 2018) and VitaminC (Thorne et al., 2018).

Prompt/Response Harmfulness: The results for Granite Guardian models, i.e., Granite-Guardian-3.0-2B and Granite-Guardian-3.0-8B, demonstrate strong performance across both *prompt* and *response*[3] harmfulness tasks. Granite-Guardian-3.0-8B consistently shows higher scores in both F1 and AUC, indicating effective detection and discrimination capabilities, particularly in challenging response harmfulness tasks. The Granite-Guardian-3.0-2B model, while smaller, also delivers robust performance, achieving competitive AUC and F1 scores that highlight its capability in harm detection with a more compact model size. Overall, Granite-Guardian-3.0-8B achieves higher aggregate scores, showcasing its generalization and effectiveness across multiple safety benchmarks. These results indicate that both Granite Guardian models are well-suited for identifying harmful content, with

[3] In the *response* harmfulness case, *prompt* and *response* are passed as pair in the risk definition template as, respectively, user message and assistant message.

model	MNBN	BEGIN	QX	QC	SumE	DialF	PAWS	Q2	Frank	AVG.
t5-11b-ANLI	0.779	<u>0.826</u>	<u>0.838</u>	0.821	0.805	0.777	0.864	0.727	0.894	0.815
WeCheck (0.4B)	**0.830**	**0.864**	0.814	0.826	0.798	0.900	**0.896**	0.840	0.881	0.850
Minicheck 7b	<u>0.817</u>	0.806	**0.907**	0.882	<u>0.851</u>	<u>0.931</u>	0.870	0.870	**0.924**	**0.873**
Granite-Guardian-3.0-2b	0.712	0.710	0.768	0.753	0.779	0.892	0.825	0.874	0.885	0.800
Granite-Guardian-3.0-8b	0.719	0.781	0.836	<u>0.890</u>	0.822	**0.946**	0.880	**0.913**	0.898	0.854

Table 2: AUC results on the TRUE dataset for groundedness. In **bold** best, <u>underlined</u> second best.

the 8B model excelling across varied harm types.

RAG Triad benchmark: We report the AUC score of the Granite Guardian models on the TRUE benchmark datasets in Table 2. It is important to note that all the baselines are trained only exclusively for groundedness task, unlike our model, which is handles multiple tasks. While Minicheck 7B achieves highest mean AUC across all the datasets, Granite Guardian 8B is a close second. Despite being trained to detect various risks, 8B model outperforms other models on three datasets and ranks second on four datasets. The Minicheck and Wecheck models likewise exhibit the highest AUC scores on three datasets each.

6 Conclusion

This work introduces the Granite Guardian family, a suite of safeguards for prompt and response risk detection. It addresses diverse risks, including RAG-specific issues like context relevance, groundedness, and answer relevance, as well as jailbreaks and custom risks, tailored for enterprise use cases. Granite Guardian models can integrate with any LLMs and outperform competitors on benchmarks, supported by transparent training with diverse human annotations to ensure inclusivity and robustness. Released as open-source , these models provide a foundation for advancing responsible and reliable AI systems.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2024. Safe RLHF: safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Nouha Dziri, Hannah Rashkin, Tal Linzen, and David Reitter. 2021. Evaluating groundedness in dialogue systems: The begin benchmark. *Preprint*, arXiv:2105.00071.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. 2024a. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. *arXiv preprint arXiv:2404.05993*.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. 2024b. AEGIS2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. In *Neurips Safe Generative AI Workshop 2024*.

IBM Granite Team. 2024. Granite 3.0 language models.

Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2021. Dialfact: A benchmark for fact-checking in dialogue. *arXiv preprint arXiv:2110.08222*.

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

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. In *NeurIPS*.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. 2024. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *CoRR*, abs/2406.18510.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. 2023. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2023. A holistic approach to undesired content detection in the real world. In *AAAI*, pages 15009–15018. AAAI Press.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2023. Tree of attacks: Jailbreaking black-box llms automatically.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ambrish Rawat, Stefan Schoepf, Giulio Zizzo, Giandomenico Cornacchia, Muhammad Zaid Hameed, Kieran Fraser, Erik Miehling, Beat Buesser, Elizabeth M. Daly, Mark Purcell, Prasanna Sattigeri, Pin-Yu Chen, and Kush R. Varshney. 2024. Attack atlas: A practitioner's perspective on challenges and pitfalls in red teaming genai. *Preprint*, arXiv:2409.15398.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *NAACL-HLT*, pages 5377–5400. Association for Computational Linguistics.

Sander Schulhoff, Jeremy Pinto, Anaum Khan, Louis-François Bouchard, Chenglei Si, Svetlina Anati, Valen Tagliabue, Anson Liu Kost, Christopher Carnahan, and Jordan L. Boyd-Graber. 2023. Ignore this title and hackaprompt: Exposing systemic vulnerabilities of llms through a global scale prompt hacking competition. *CoRR*, abs/2311.16119.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Hao Sun, Zhexin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. 2023. Safety assessment of chinese large language models. *CoRR*, abs/2304.10436.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

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

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2024. Do-not-answer: Evaluating safeguards in llms. In *EACL (Findings)*, pages 896–911. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Sujian Li, and Yajuan Lyu. 2023. Wecheck: Strong factual consistency checker via weakly supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 307–321.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. 2024. Shieldgemma: Generative AI content moderation based on gemma. *CoRR*, abs/2407.21772.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043.