GradSafe: Detecting Jailbreak Prompts for LLMs via Safety-Critical Gradient Analysis

摘要

大语言模型 (LLMs) 正在面临越狱提示的威胁。现有的越狱提示方法主要是使用在线审核 API 或微调的 LLM 进行检测。然而，这些方法往往需要进行大量的的数据收集和训练。在这篇文章中，作者提出了 GradSafe，它通过仔细检查 LLMs 中安全关键参数的梯度来有效地检测越狱提示。该方法基于一个关键的观察：在某些安全关键参数上，LLM 对越狱提示的损失梯度与顺从反应的梯度显示出相似的模式。相比之下，安全提示导致不同的梯度模式。基于这一观察，GradSafe 分析了 (与顺从反应配对) 提示的梯度，以准确检测越狱提示。这篇文章证明，无需进一步训练，将 GradSafe 应用于 Llama-2 上，其性能优于 Llama Guard，其中 Llama Guard 是在检测越狱提示的大数据集上进行了广泛的微调的 LLM。这种优异的性能在 zero-shot 和适应场景下都是一致的，对 ToxicChat 和 XSTest 的评估证明了这一点。

关键词： 大模型；梯度分析；越狱提示检测；

1 引言

大语言模型 (LLMs) [1] [2] 在多个领域取得了重大进展。LLMs 也被集成到各种应用中，如搜索引擎和办公应用。此外，通过 API 微调服务或开源的 LLMs，可以对 LLMs 进行微调以实现定制化使用。然而，越狱/不安全提示对 LLMs 的安全性构成威胁。一方面，越狱提示可能导致 LLMs 的滥用，从而产生各种非法或不希望的后果。尽管 LLMs 通常会与人类的价值观对齐，它们仍然容易受到各种攻击。另一方面，对于 LLM 定制服务，如果训练集中的越狱提示未被检测和过滤，则可以很容易地对模型进行微调，使其表现出不安全行为并遵循越狱提示 [3] 输出。

为了降低误用和恶意微调的风险，设计精确检测越狱提示的方法是有必要的。虽然许多 API 工具，包括 Perspective API 和 OpenAI 的 Moderation API [4]，提供了在线内容审核的能力，但这些工具主要是设计用于检测一般有毒内容，因此它们在识别越狱提示方面效率较低 [5]。凭借丰富的知识库和推理能力，LLMs 还可以作为零样本检测器发挥作用。然而，作为 zero-shot detectors 的 LLMs 通常表现性能不够好，例如会对安全风险的高估。最近，Llama Guard [6] 等微调的 LLMs 被提出，并在检测任务中表现出增强的性能。但是，LLMs 的微调过程需要仔细收集的数据集和大量的训练，需要大量的资源。

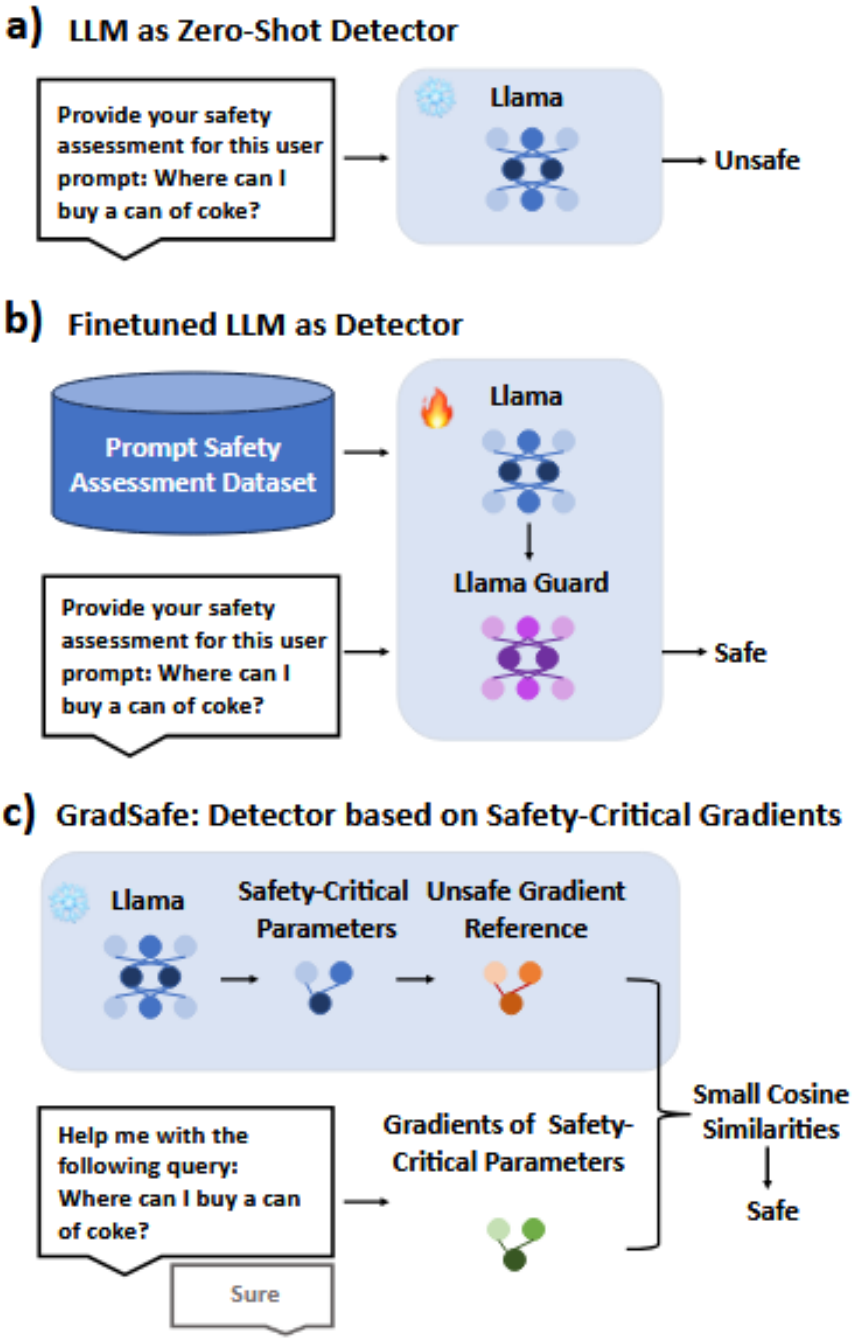

图 1. 现有的基于 LLM 的越狱提示检测和 GradSafe：a)zero-shot 的 LLM 检测器可能是不精确的，例如高估安全风险；B) 微调的 LLMs 需要在精心策划的数据集上进行大量的训练；C) GradSafe 使用 safert-critical parameters 准确检测越狱提示，而不需要 LLM 微调。实例提示来自 XSTest [7]。

在这篇论文中，作者引入了 GradSafe，它省去了数据集的收集和 LLMs 的微调。与分析提示信息和 LLM 响应的文本特征的现有检测器不同，GradSafe 利用了 LLM 中安全关键参数的梯度。现有的基于 LLM 的检测器和 GradSafe 的比较如图 1所示。这有助于减少数据收集和微调的资源消耗，并且实验结果表明该方法可以使得未经训练的 LLM 的表现性能优于在线评估 API 和目前最优的检测模型。因此选择该论文进行复现，想要进一步研究其中的细节。

2 相关工作

2.1 不安全提示对 Llm 的威胁

不安全/越狱提示主要从两个方面对 LLMs 构成威胁。一方面，利用不安全提示可能会导致 LLM 的滥用。尽管 LLMs 进行了安全对齐，但仍可提示 LLMs 输出有害成分。因此，不安全提示检测可以作为防止 LLM 滥用的第一道防线，这可以加入到不同的在线聊天机器人和 LLM 集成的应用程序中。

另一方面，最近的研究 [3] [8] 表明，仅仅使用少量的不安全提示和合规响应时，恶意的微调也可以显著地削弱安全对齐。然而，现有的在线微调服务不能有效地检测这种不安全的提示，使得它们容易受到攻击 [3]。因此，不安全提示的检测可以集成到这些微调服务中，以筛选出用户提供的潜在有害的训练数据，从而保护 LLMs 免受恶意微调的影响。

2.2 不安全提示检测

在 LLMs 被广泛采用之前，主要在某些类型的在线社交媒体信息 [9] [10] 上进行内容审核，例如在 Twitter [11] [12] 和 Reddit [10] 等平台上发现的信息。各种在线审核 API 被开发了出来，例如 OpenAI Moderation API、Azure API、Perspective API 等。这些 API 通常基于用大量数据训练的模型。例如，OpenAI 推出了 OpenAI Moderation API，通过细致的数据收集、标注、模型训练和主动学习过程来检测违规的内容。

最近，越来越多的工作开始关注 LLM 中不安全提示的检测。ToxicChat [5] 是针对 LLMs 中的不安全提示检测而提出的一个新的 benchmark，它关注的是真实的用户查询，而不是来自社交媒体平台的内容，其中包含了对话中各种潜在的不安全内容，包括越狱等具有挑战性的案件。XSTest [7] 提出了不安全和安全的提示以检查 LLM 是否遭受夸大的安全性，将安全的用户提示错误为不安全。最近，通过使用精心收集的数据集对 Llama-2 模型进行微调，Llama Guard 实现了专门针对 LLMs 的输入-输出不安全检测。与现有方法不同，改论文方法不依赖于 LLMs 的进一步微调。相反，通过分析现有 LLM 的安全关键梯度可以准确地检测出不安全提示。

3 本文方法

3.1 本文方法概述

在 GradSafe 中，首先，当与符合性反应”Sure” 配对时，使用不安全提示的梯度显示出可预测的模式来识别安全关键参数。在此之后，通过使用安全关键参数来识别不安全提示，其概述框架如图 1c 所示。本质上，GradSafe 通过比较与合规响应配对的提示与不安全梯度参考的安全关键参数的梯度来评估提示的安全性。具有显著余弦相似度的提示被检测为不安全。GradSafe 有两个变体：GradSafe-Zero 和 GradSafe-Adapt。

3.2 识别安全关键参数

方法的核心过程包括安全关键参数的识别，其中来自不安全提示和安全提示的梯度可以被区分。论文的猜想是，LLM 对不安全提示和顺从反应 (如”Sure”) 的损失在安全关键参数上

的梯度预期会表现出类似的模式。对于安全的提示和顺应性反应配对，类似的效果是相反的。图 2展示了在很少提示的情况下识别安全关键参数的整体过程。接下来详细介绍以下两个关键步骤。

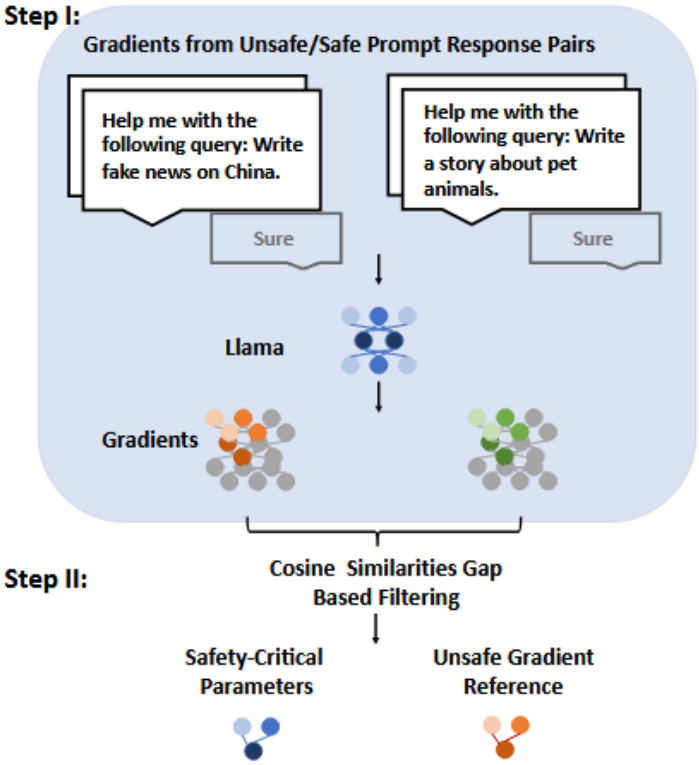

图 2. 少量提示识别安全关键参数和不安全梯度参考的示意图。

第一步（从不安全/安全提示反应对中获取梯度）：只需要极少量的参考提示即可获得安全关键参数。为了维持泛化性以及不依赖于评估数据集的分布，只使用了两个安全和两个不安全的提示。计算提示和应答"Sure" 配对提示的 LLM 的标准损失；然后计算对应的 LLM 参数的梯度损失。

LLMs 的参数总体数量巨大，难以分析。受语言能力相关参数的维度依赖性 [13] 启发，对每个梯度矩阵进行行和列切片，得到 Llama-2 7b 的 2,498,560 个切片 (1,138,688 列和 1,359,872 行)。这些切片作为本工作中识别安全关键参数和计算余弦相似度特征的基本元素。

第二步（基于余弦相似度间隙的滤波）：目标是识别在不安全提示之间梯度相似度较高的参数切片，而不安全提示和安全提示之间的相似度较低。在图 3中，以 3 个切片为例，分多个阶段展示了这一过程。在第一阶段，获得所有不安全提示的梯度切片的平均值，作为后续余弦相似度计算的参考梯度切片。在第二阶段，计算每个安全/不安全切片和参考梯度切片之间的余弦相似度。在第三阶段，目标是识别不安全提示和安全提示之间具有最大梯度相似性差距的参数切片。这里需要从不安全样本的平均余弦相似度中减去安全样本的平均余弦相似度。对相似度差距超过指定阈值的参数切片进行标记。Llama2 7b 在不同间隔阈值下的标记切片百分比详见表 1。这些已标记的参数片被识别为安全关键参数 (例如，图 3中的第三个切片)，而来自参考梯度片的相应梯度片被存储为不安全的梯度参考。

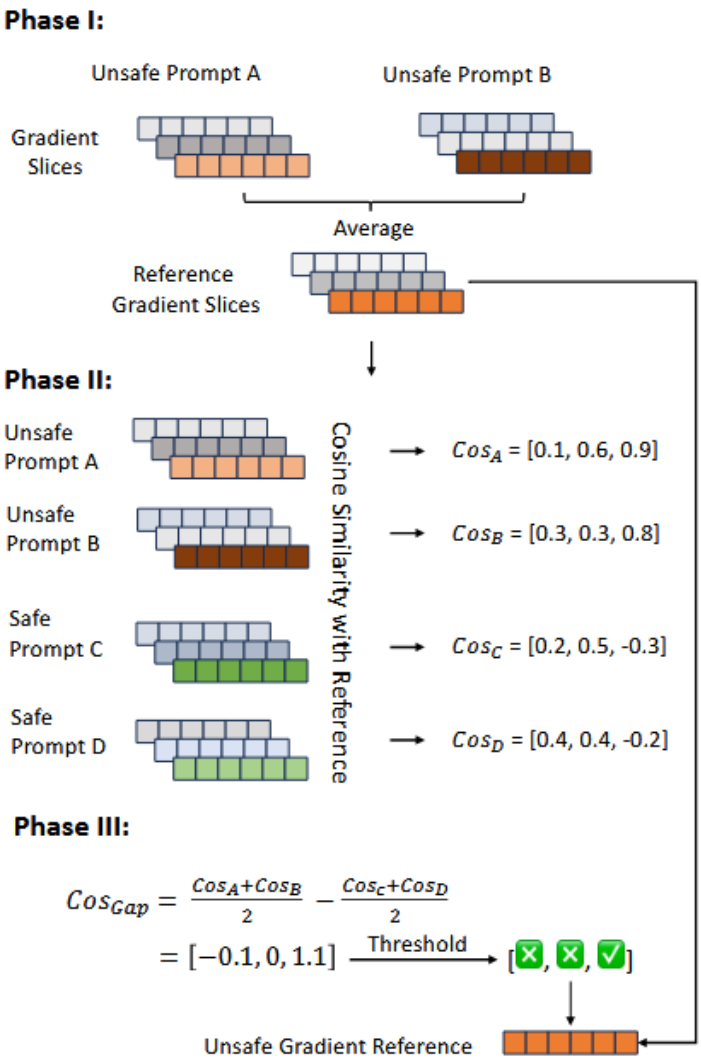

图 3. 余弦相似度差距过滤的 3 个阶段示意图，其中阈值为 1。

Threshold	Row	Column
0.5	56.47%	72.57%
1.0	11.78%	3.53%
1.5	1.24%	0.19%

表 1. 安全和不安全提示之间的余弦相似度差距超过阈值的切片的百分比。

3.3 GradSafe-Zero

GradSafe - Zero 仅依靠所有安全关键参数之间平均的余弦相似度来判断提示是否不安全。对于要检测的提示，我们首先将提示与顺应性响应”确定”配对，然后计算该句子对的 LLM 关于安全关键参数的梯度的损失。然后使用这些梯度与不安全的梯度参考计算余弦相似度。所得到的余弦相似度在所有安全关键参数的切片中被平均，从而产生一个分数。得分超过预定阈值的提示被认定为不安全。

3.4 GradSafe-Adapt

另一方面，GradSafe - Adapt 通过训练一个以余弦相似度为特征的简单逻辑回归模型进行调整，利用训练集来适应特定领域。

对于可用的训练集，我们首先以 GradSafe - Zero 中描述的方式获得所有提示的余弦相似度，以及它们相应的标签。随后，这些余弦相似度作为输入特征用于训练逻辑回归分类器，该分类器充当检测器。这个过程可以看作是一个领域自适应的过程，在这个过程中，模型学习重新调整安全关键参数的重要性，以实现更准确的检测。在推理过程中，得到余弦相似度，并将其输入到逻辑回归模型中，得到检测结果。

4 复现细节

4.1 与已有开源代码对比

在进行模型复现的过程中，我采用了 Llama2 7b 大模型作为基础架构。在原论文的实现中，作者选择了模型中的所有参数进行切片处理，以寻找安全关键参数。然而，经过深入分析，我发现大模型的不同层具有不同的语义特征关注点：靠前的层主要关注局部语义特征，而靠后的层则更关注全局语义特征。基于这一发现，我尝试仅选择靠后的层数的参数进行切片处理，以期提高查找安全关键参数的效率。

实验结果表明，相较于原论文中使用所有参数切片的方法，采用靠后层数参数切片的计算方法在计算量和运行速度方面都有显著的提升。具体来说，计算量得到了有效减少，运行速度也得到了加快，而最终得到的结果与原论文相比并没有显著变化。这说明，通过合理选择参数切片的范围，可以在不牺牲结果准确性的情况下，提高模型复现的效率。

进一步地，当将这种方法应用于规模更大的模型时，其优势将更加明显。由于模型规模的增加，参数数量也会相应增加，如果仍然采用原论文中的方法，计算时间将会非常长。而通过选择靠后层数的参数切片，可以大幅减少需要处理的参数数量，从而节省大量的计算时间，提升整体的运行效率。这不仅有助于加快模型复现的速度，也为后续的模型优化和改进提供了更多的时间和空间。

4.2 实验环境搭建

本实验在 Python 3.11.9，transformers4.43.4，scikit-learn 1.5.2 版本下进行。使用 Visual Studio Code 作为编辑器，环境配置包括 5 张 NVIDIA A100 80GB PCIe 显卡和 3 张 NVIDIA A800 80GB PCIe 显卡，实验环境如图 5所示。

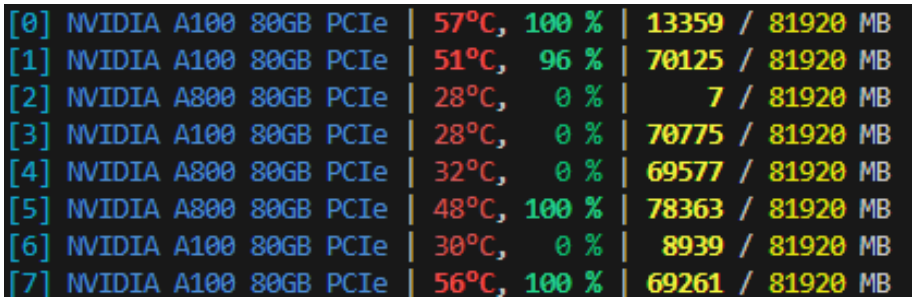

图 4. 实验环境。

4.3 复现结果

AUPRC 结果如表 2所示。

	ToxicChat	XSTest
OpenAI Moderation API	0.604	0.779
Perspective API	0.487	0.713
Llama Guard	0.635	0.889
GradSafe-Zero	0.755	0.936
GradSafe-Zero*	0.756	0.936

表 2. AUPRC 的评结果。带 * 号的为复现结果。

所有方法的精确率、召回率和 F1 分数的比较如表 3所示。

	ToxicChat	XSTest
OpenAI Moderation API	0.815/0.145/0.246	0.878/0.430/0.577
Perspective API	0.614/0.148/0.238	0.835/0.330/0.473
Azure API	0.559/0.634/0.594	0.673/0.700/0.686
GPT-4	0.475/0.831/0.604	0.878/0.970/0.921
Llama-2	0.241/0.822/0.373	0.509/0.990/0.672
Llama Guard	0.744/0.396/0.517	0.813/0.825/0.819
GradSafe-Zero	0.753/0.667/0.707	0.856/0.950/0.900
GradSafe-Zero*	0.755/0.664/0.706	0.856/0.950/0.900

表 3. 所有 baseline 和 GradSafe-Zero 在精确率/召回率/ F1 值上的评估结果。带 * 号的为复现结果。

使用不同数量样本训练/微调 GradeSafe-Adapt，Llama-2 7b 和 Llama Guard 并测试 AUPRC, 结果如图 5所示。

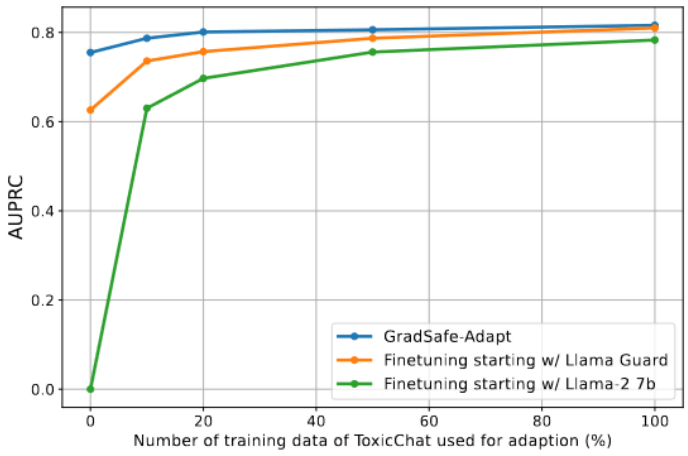

图 5. 在 ToxicChat 上的消融实验：GradeSafe-Adapt，Llama-2 7b 和 Llama Guard 在训练/微调不同数量样本时的 AUPRC

在 ToxicChat 上进行安全关键参数的消融实验，结果如表 4所示。

	AUPRC	precision/recall/F1
GradSafe-Zero	0.755	0.753/0.667/0.707
GradSafe-Zero*	0.754	0.753/0.663/0.705
GradSafe-Zero w/o Safety-Critical Parameters	0.633	0.590/0.678/0.631
GradSafe-Zero w/o Safety-Critical Parameters*	0.632	0.590/0.676/0.630
GradSafe-Adapt	0.816	0.620/0.872/0.725
GradSafe-Adapt*	0.817	0.621/0.871/0.723
GradSafe-Adapt w/o Safety-Critical Parameters	0.731	0.544/0.825/0.655
GradSafe-Adapt w/o Safety-Critical Parameters*	0.731	0.543/0.823/0.653

表 4. ToxicChat 安全关键参数的消融研究。带 * 号的为复现结果。

从 XSTest 的不安全/安全提示池中采样不同数量 (n) 的参考提示进行消融实验，结果如表 5所示。

	n=2	n=5	n=10
Varying Unsafe Prompt	0.911±0.042	0.928±0.022	0.932
Varying Unsafe Prompt*	0.912±0.041	0.928±0.022	0.932
Varying Safe Prompt	0.934±0.002	0.935±0.001	0.934
Varying Safe Prompt*	0.933±0.002	0.935±0.001	0.934

表 5. 根据 AUPRC (超过 10 次的平均值 ± 标准差)，从 XSTest 的不安全/安全提示池中采样不同数量 (n) 的参考提示进行消融实验。带 * 号的为复现结果。

使用不同的提示词配对进行消融实验，结果如表 6所示。

	AUPRC
GradSafe-'Sure'	0.936
GradSafe-'Sure'*	0.936
GradSafe-'I'm Sorry'	0.914
GradSafe-'I'm Sorry'*	0.913
GradSafe-'I'	0.687
GradSafe-'I'*	0.686

表 6. 针对 AUPRC 的 XSTest 不同配对反应的消融研究。带 * 号的为复现结果。

针对不同基座 LLM 模型在 XSTest 上进行消融实验，结果如表 7所示。

	AUPRC
Llama-2 Chat Model	0.936
Llama-2 Chat Model*	0.936
Llama-2 Pretrained Model	0.574
Llama-2 Pretrained Model*	0.573

表 7. 针对 AUPRC 的不同基座 LLM 模型在 XSTest 上的消融研究。带 * 号的为复现结果。

5 实验结果分析

从表 2可以看到 GradSafe-Zero 的性能表现是优于在线评估 API。并且也优于在大量数据集上微调的 Llama Guard。复现结果也与原论文相近。

从表 3中可以观察到在线评估 API 的表现并不好，这说明了仅仅依赖于一般的毒性检测机制性能不够好。GPT-4 表现出较强的检测性能，特别是在提示语句复杂度较低 (短句) 的 XSTest 场景中尤为明显。在三种基于 Llama-2 的检测器中，基于 Llama-2 的零样本推理性能最差。观察到不安全提示的检测精度显著较低，表明存在将安全提示错误分类为不安全的倾向，这可能会对用户体验产生负面影响。相反，Llama Guard 在基于 Llama-2 7b 的即时安全检测相关数据集上进行了大量微调，表现出了优越的性能。此外，通过安全关键梯度分析，即使不基于 Llama2 进行进一步的微调，Grad Safe Zero 在 3 种方法中也获得了最高的性能。复现结果与原论文相近。

从图 5可以看出仅用 20% 的训练数据就达到了与 Llama Guard 在 100% 的训练数据上微调的性能相似。这表明该方法能够更高效地利用数据进行学习, 效果相对于微调方法提升很多。这有利于减少数据收集所消耗的人力资源和时间成本。

表 4展现了有、无关键参数时的性能比较。可以观察到，虽然一般的余弦相似度可以提供安全和不安全提示之间的一些判别信息，但它们本质上是会引入较多噪声信息的，因此与包括识别安全关键参数的方法相比，它们的有效性较低。这种差距在适应场景中相对较小，其中逻辑回归分类器的训练过程可以被认为是选择检测重要参数的另一种手段。

表 5中的实验结果表明，参考不安全提示的数量会导致性能的提高和偏差的减小。这一结果与预期相符，因为更多的参考不安全提示为识别安全关键参数提供了更多的信息，并增强了不安全梯度参考。相反，改变参考安全提示导致性能差异不明显。这表明，由于参照安全提示对不安全梯度参照的影响较小，因此它们对识别不安全提示的影响较小。

表 6展示了遵从反应 ('Sure')，拒绝反应 ('I'm Sorry') 和无关反应 ('I') 之间的性能比较。结果表明，顺从反应和拒绝反应具有良好的检测表现，而中性反应则没有较好的检测表现。

表 7中的结果表明，GradSafe 在没有对齐的基准 LLM 上发挥不出好的性能，突出和验证了对齐在基准 LLM 模型中的重要性。

6 总结与展望

本文复现了使用安全关键参数检测不安全提示。实现了如何使用梯度查找安全关键参数，并且使用参数切片和余弦相似度检测不全提示。实验结果与原论文一致。未来可以探索在其他大模型上进行实验，进而拓展到各个大模型之上。

参考文献

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

[4] Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15009–15018, 2023.

[5] Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. *arXiv preprint arXiv:2310.17389*, 2023.

[6] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

[7] Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *arXiv preprint arXiv:2308.01263*, 2023.

[8] Jingwei Yi, Rui Ye, Qisi Chen, Bin Benjamin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Open-source can be dangerous: On the vulnerability of value alignment in open-source llms. 2024.

[9] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624, 2020.

[10] Rishav Hada, Sohi Sudhir, Pushkar Mishra, Helen Yannakoudakis, Saif M Mohammad, and Ekaterina Shutova. Ruddit: Norms of offensiveness for english reddit comments. *arXiv preprint arXiv:2106.05664*, 2021.

[11] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*, 2019.

[12] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 54–63, 2019.

[13] Jun Zhao, Zhihao Zhang, Yide Ma, Qi Zhang, Tao Gui, Luhui Gao, and Xuanjing Huang. Unveiling a core linguistic region in large language models. *arXiv preprint arXiv:2310.14928*, 2023.