通过数据过滤和对齐蒸馏的拒绝特征引导教师进行安全微调

Seokil Ham **Yubin Choi** **Seungju Cho** **Yujin Yang** **Younghun Kim** **Changick Kim**

Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, South Korea
{ gkatjrdlf, choibinbin, joyga, ujin.y, younghun1664, changick } @kaist.ac.kr

Abstract

最近，Google 和 OpenAI 等主要的人工智能服务提供商推出了 Finetuning-as-a-Service，这项服务允许用户使用自己的数据定制大型语言模型（LLMs）以执行特定的下游任务。然而，当用户数据中包含有害提示时，这项服务易受到 LLM 安全对齐性能下降的影响。虽然已有一些研究工作解决了这个问题，但从根本上过滤用户数据中的有害数据仍然没有得到探索。受到我们观察的启发，即从安全对齐的 LLMs 中获得的反映拒绝行为的定向表示（称为拒绝特征）可以本质上区分有害和无害提示，我们提出了拒绝特征引导教师（ReFT）。我们的 ReFT 模型被训练来根据输入提示特征与其拒绝特征之间的相似性识别有害提示。在微调过程中，ReFT 模型作为教师角色，从用户数据中过滤有害提示，并将对齐知识传送给基础模型。大量实验表明，我们基于 ReFT 的微调策略有效地减少了有害输出，并提高了用户特定任务的微调准确性，为在 Finetuning-as-a-Service 中安全可靠地部署 LLMs 提供了实用解决方案。

1 介绍

大型语言模型（LLM）的最新进展在常识推理、问答、摘要和对话生成等广泛的自然语言处理任务中取得了显著的表现。这些 LLM 通常是在庞大而多样的语料库上进行预训练的，从而具有强大的泛化能力，并允许在多个领域广泛应用。为了进一步促进 LLM 在个体和领域特定目的的应用，诸如谷歌和 OpenAI 等主要 AI 服务提供商不仅提供对预训练 LLM 的访问，还提供“服务即微调”。该服务使用户能够上传自定义数据集，并根据他们的独特需求调整 LLM 以适应更具体的任务和领域。

然而，微调服务必须通过安全对齐来防范 LLM 的恶意使用，即使用户试图通过定制来绕过模型。为了降低这些风险，微调即服务通常采用两阶段的流程。在第一阶段，即对齐阶段，预训练的 LLM 被训练以避免对有害提示生成有害响应，并对无害提示产生有帮助的响应。在第二阶段，即微调阶段，对齐的模型在用户数据上进行微调，用于定制的下游任务。尽管有这种两阶段的方法，已有多项研究 [10, 14, 15, 16, 22, 36] 表明，对包含有害内容的用户数据进行微调可能会影响在对齐阶段初步实现的安全对齐。这种类型的攻击，通过在用户数据中注入有害提示进行微调，被称为有害微调攻击。这类攻击的存在突显了在微调即服务中实现用户任务高性能的同时保持模型安全性的必要性。

为了解决有害的微调攻击，先前的研究提出的解决方案主要针对对齐阶段 [15, 16, 36] 或微调阶段 [14, 22, 23, 30] 。然而，我们在表 2 中的实验结果表明，随着用户数据中有害提示的比例增加，通过对齐阶段解决方案维持安全对齐是具有挑战性的。这种漏洞在对抗性场景中尤为关键，在这些场景中，恶意攻击者试图破解大型语言模型（LLMs），他们向用户数据中注入大量有害提示，以最大化成功攻击的可能性。此外，尽管识别和移除用户数据中的有害提示是安全微调的基本方法，但直接过滤有害提示的方法仍然很少被探索。

Preprint. Under review.

因此，在这项工作中，我们提出了一种新颖的微调阶段解决方案，通过在微调过程中从用户数据中过滤有害提示来确保安全的微调。我们观察到从安全对齐的 LLM 中获得的拒绝特性可以有效区分有害和无害的提示，由此引入了拒绝特性指导教师（ReFT）模型。ReFT 模型不仅被设计用来根据其拒绝特性准确地分类提示的有害程度，还能够对有害请求生成适当的拒绝响应。在微调阶段，ReFT 模型被用于教师的两个不同目的。首先，ReFT 模型通过识别并移除用户数据中的有害提示，以其拒绝特性为指导进行数据过滤，从而防止暴露于有害数据导致的安全性下降。其次，ReFT 模型作为对齐蒸馏的教师模型，生成提供更多信息性监督的软拒绝标签。这些软拒绝标签指导学生模型遵循 ReFT 模型的安全对齐行为。此外，软标签有助于平滑对齐损失面，使其更容易与用户数据上的微调损失无缝整合。

我们的广泛实验结果证明了以 ReFT 为基础的微调策略在用户特定的下游任务表现和安全对齐方面的有效性。在大多数情况下，与基线相比，我们的方法达到了最高的微调准确性和最低的有害评分。因此，结合基于 ReFT 的有害数据过滤和对齐知识蒸馏，为 LLMs 的安全可靠部署提供了实际解决方案。此外，我们的观察提供了未来研究的新见解：从安全对齐的 LLMs 中提取的拒绝特征可以作为区分有害和无害数据的可靠指标。我们的贡献

- 据我们所知，这项工作首次通过对齐的大型语言模型分析拒绝特征来区分有害和无害的提示。我们的观察表明，拒绝特征可以作为准确分类提示有害性的重要指标。
- 基于我们的观察，我们提出了 ReFT 模型和一种基于 ReFT 的微调策略，该策略利用 ReFT 模型的拒绝特性实现 (i) 准确过滤用户数据中的有害提示以及 (ii) 在微调过程中进行安全对齐知识蒸馏。
- 我们的实验结果表明，基于 ReFT 模型的有害数据过滤和对齐知识蒸馏是有效的，在用户特定的下游任务中实现了强大的性能，同时在各种环境中始终保持模型的安全性。

大型语言模型的安全性。大型语言模型 (LLM) 可以响应广泛的用户查询，但易受到有害提示的影响，这可能导致有害输出，例如制造武器的指令或生成假新闻内容。为了减轻这些风险，引入了安全对齐的 LLM。这些模型通过监督微调 (SFT) 或结合人类反馈的强化学习 (RLHF) 在大规模安全对齐数据集上进行微调，该数据集包括与无害（拒绝）响应配对的有害提示。因此，安全对齐的模型能够拒绝有害请求。然而，安全对齐模型仍然容易受到高级越狱技术的攻击。随后，最近的研究集中于大幅提高 LLM 的安全性，现有方法大致可分为不需训练和基于训练的方法。一些不需训练的方法利用 LLM 的内在能力来评估有害性或预测用户意图。其他方法利用模型内部的差异，例如处理有害与无害输入时的参数、梯度或注意模式。相比之下，基于训练的方法通过对抗训练微调 LLM 或通过训练辅助模型来提高鲁棒性。一些对抗训练方法研究在微调期间使用的有害和无害提示之间的平衡，或者利用通过在潜在空间中添加扰动生成的对抗样本。其他方法训练独立的安全和不安全（辅助）模型并应用安全解码策略。最近，引入了拒绝特征的概念，该特征编码了安全对齐的 LLM 的拒绝行为，并在对抗性攻击 [3] 和防御 [50] 中加以利用。在拒绝特征的洞察基础上，我们进一步分析了拒绝特征，证明了其在将提示分类为有害或无害方面的有效性。基于拒绝特征的能力，我们提出了一种用于安全 LLM 微调的新颖微调策略。

有害微调攻击。除了通过输入提示引发有害响应的破解攻击之外，有害微调攻击也通过在微调过程中影响安全性对 LLM 的安全构成重大威胁。这些攻击是破解技术的一个子类，其中有害的输入输出对被注入到微调数据中，导致模型生成不安全的输出。有多项研究 [10, 14, 15, 16, 22, 36] 强调了微调数据中有害内容的潜在风险。此外，像 LoRA [11] 这样参数高效的微调方法即使在微调数据不包含有害提示的时候，也会表现出安全性降低的现象 [21, 33] 。随着主要的 AI 服务提供商开始提供微调即服务，维持对这些攻击的安全对齐的重要性变得越来越关键。为了解决这一问题，先前的工作提出了针对对齐阶段、微调阶段或后微调阶段的解决方案。首先，对齐阶段的解决方案旨在预先增强对由于训练有害提示而导致的安全性降级的鲁棒性。这些方法通过基于预期扰动 [15, 16, 26, 36, 40] 的正则化技术来减轻潜在风险。其次，微调阶段的解决方案通过冻结安全关键参数 [22, 23, 45] 或结合以安全为导向的正则化 [14, 30, 32] ，在用户特定的下游任务微调期间保持安全。这些方法通常在微调期间采用额外的对齐数据作为安全指导。最后，后微调阶段的解决方案专注于分析对齐模型和微调模型之间的差异。基于其分析，模型权重通过各种技术进行调整，如恢复 [49] 、剪枝 [13] 或投影 [10] ，以弥补微调后的安全性降低。与先前的方法相比，我们提出了一种基于 ReFT 的微调阶段解决方案，该方案通过拒绝特性过滤有害提示，并提取对齐知识以在微调期间保持安全性。

2 问题设置

场景。我们假设微调即服务的提供商（即，拥有和部署 LLMs 的公司）将模型安全性放在优先位置。在对齐阶段，假定服务提供商可以访问包含 5,000 个有害提示和 5,000 个无害提示

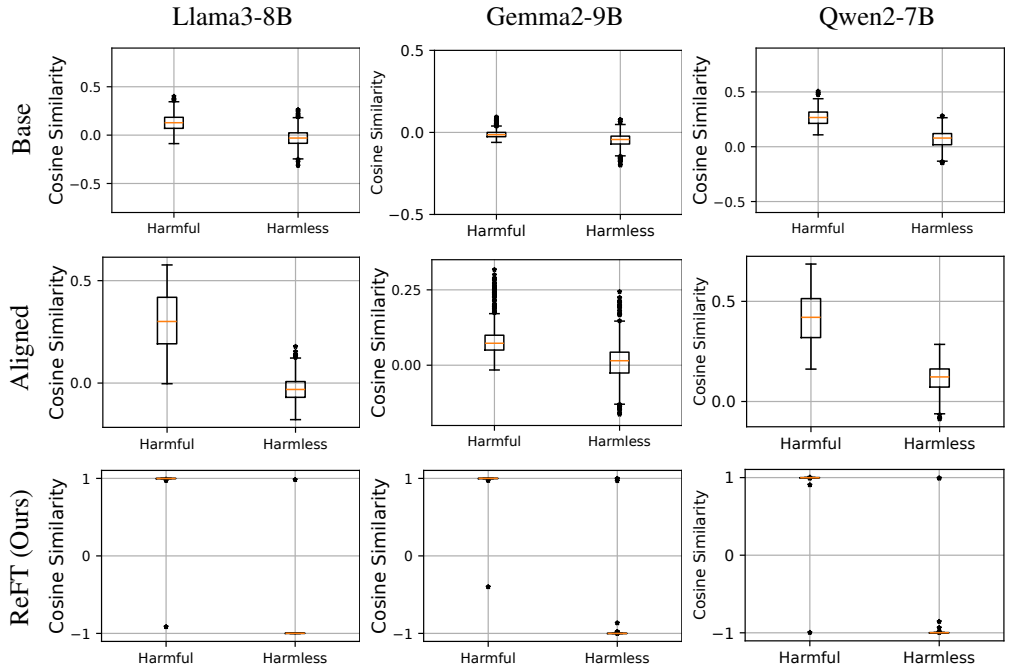

Figure 1: 基准模型、对齐模型和 ReFT（我们的）中有害和无害提示词余弦相似度分布的箱线图。提示词从 BeaverTails（有害，n=500）和 Alpaca（无害，n=500）数据集中采样，代表多样化的一般提示词。此处可视化的采样提示词未包含在 ReFT 训练集中。此可视化强调了安全对齐引入了区分有害和无害提示词的能力。

Table 1: 使用拒绝特征进行提示分类的准确性。余弦相似度超过阈值的提示被归类为有害，而低于阈值的提示被归类为无害。

Model	Threshold	Harmful Acc	Harmless Acc	Total Acc
Llama3-8B	0.34	86.0 %	78.8 %	82.4 %
Llama3-8B-Instruct	0.06	95.2 %	93.6 %	94.4 %
Llama3-8B-ReFT	0.97	99.8 %	99.8 %	99.8 %
Gemma2-9B	-0.037	87.8 %	61.2 %	74.5 %
Gemma2-9B-Instruct	0.035	90.4 %	70.4 %	80.4 %
Gemma2-9B-ReFT	0.97	99.8 %	99.6 %	99.7 %
Qwen2-7B	0.15	97.6 %	88.8 %	93.2 %
Qwen2-7B-Instruct	0.24	93.2 %	97.2 %	95.2 %
Qwen2-7B-ReFT	0.9	99.8 %	99.6 %	99.7 %

的数据集。对于每一个有害提示，都会配对一个相应的安全响应，以适当地拒绝有害请求。使用此数据集，LLMs 被训练生成对有害提示的安全拒绝响应。在微调阶段，用户向服务提供商提交自定义数据集以进行模型定制或越狱。然而，服务提供商事先既不了解用户数据是否包含有害提示，也无法在对齐阶段访问用户数据的分布。

威胁模型。我们假设用户数据包含 $p\%$ 个有害提示及其有害响应，而剩余的 $(1-p)\%$ 是从同一数据集中采样的无害提示。当 $p=0$ 时，数据集仅包含无害提示。重要的是，用户没有告知哪些提示是有害的或无害的，从而使 LLMs 在微调过程中有安全性降低的风险。同时，期望 LLMs 在用户特定的下游任务中保持强性能，并维持其安全对齐性，使得该问题尤其具有挑战性。

3 初步: 拒绝特征

拒绝特征是在 [3] 中引入的，它是一种与安全对齐的 LLM 中的拒绝响应相关联的一维表示。这些特征定义为在特定层次上有害和无害提示的特征表示之间的平均差异。设 x^s 和 x^{us} 分别为安全提示和不安全提示，设 f^l 表示从 LLM 的具体层 l 中提取的最后一个输入标记的特征。拒绝特征 R^l 计算如下：

$$R^l = \frac{1}{N_{us}} \sum_{i=1}^{N_{us}} f^l(x_i^{us}) - \frac{1}{N_s} \sum_{i=1}^{N_s} f^l(x_i^s), \tag{1}$$

，其中 N_s 和 N_{us} 分别表示安全和不安全提示的数量。由于安全对齐的 LLM 被训练为对有害提示输出拒绝响应，而对无害提示输出有用的响应，因此有害和无害提示特征的平均差异仅提取与拒绝行为特别相关的特征。然后，拒绝特征向量 R^l 可以在模型的表示空间中特定层 l 中被解释为一种拒绝方向。

4 观察：拒绝特征可以分类有害/无害的提示

安全对齐的大语言模型（LLMs）在与拒绝响应配对的有害提示和与适当的任务相关输出配对的无害提示上进行训练。然后，当提供引发不安全行为的有害提示时，对齐的模型会生成明确的拒绝响应（例如，“对不起，我无法帮助处理该请求。”）。相反，当给出无害提示时，模型会产生与任务相关的响应。基于这些不同的响应行为，我们假设在安全对齐的 LLMs 中，源自有害和无害提示的内部特征表示是可区分的，并且这种区分体现在对齐模型的拒绝特征中。为了验证我们的假设，我们在基础和对齐的模型中测量输入提示特征与拒绝特征之间的相似性，并评估是否可以根据这种相似性区分有害和无害提示。图 1 展示了 BeaverTails（有害）[19] 和 Alpaca（无害）数据 [41] 的余弦相似性分布的箱形图。结果表明，对齐模型展现了更为明显的相似性分布，使得有害和无害提示之间的区分更加清晰。相比之下，基础模型的相似性分布显示出明显的重叠，使得区分这两类提示变得具有挑战性。数字上看，表 1 表明，对齐模型在有害和无害提示上都实现了比基础模型更高的准确性。这些发现验证了从安全对齐模型中提取的拒绝特征在区分有害和无害提示方面的有效性。

受此观察启发，我们开发了拒绝特征引导教师（ReFT）模型，该模型通过利用拒绝特征进行训练，以更有效地区分有害和无害的提示。如图 1 和表格 1 所示，ReFT 模型在相似度分布的区分度和分类准确性上均优于基本模型和对齐模型。

5 方法：拒绝特征引导教师（ReFT）

基于我们的观察，拒绝特征可以作为区分有害和无害提示的可靠指示器，我们训练了一个拒绝特征引导教师 (ReFT) 模型，并用它进行安全微调。与之前的工作不同，我们的方法引入了一个教师准备阶段，在这一阶段中训练 ReFT 模型，而不是对齐阶段。随后，在微调阶段，我们在 ReFT 模型的指导下直接对用户数据进行微调一个未对齐的基础模型。

5.1 教师准备阶段

教师准备阶段的目标是训练 ReFT 模型，以准确区分有害和无害的提示。为了实现这一点，我们使用安全对齐损失来优化基础模型，这是一种监督微调损失，在一个安全对齐数据集上计算，该数据集将有害提示与拒绝响应配对，并将无害提示与适当的有帮助响应配对。对齐损失训练模型只拒绝有害请求，同时生成与无害查询相关的响应，从而展示对不同提示类型的不同行为。为了进一步增强 ReFT 模型的判别能力，我们引入了一个正则化项，鼓励有害和无害提示特征之间根据其与拒绝特征的相似性进行更明确的区分。具体来说，正则化项强制拒绝特征与有害提示特征之间的余弦相似性趋近于 1，而与无害提示特征的相似性趋近于 -1。然而，过于强烈的正则化可能导致不理想的内部表示，可能产生难以理解的响应。为防止这种情况，我们使用超参数 λ 来控制正则化的强度。教师准备阶段的最终目标函数定义为安全对齐损失和正

Algorithm 1 ReFT 模型的训练过程

Require: Unsafe data x^{us} , Safe data x^s , Cycle number C , LoRA weight W , Regularization strength λ , Learning rate η
Ensure: Trained LoRA weight W , Refusal Feature R^l
Initialize Unsafe prompt set $S_{us} \leftarrow []$
Initialize Safe prompt set $S_s \leftarrow []$
Initialize Refusal feature $R^l \leftarrow None$
Initialize Counter $c \leftarrow 0$
while not converged **do**
  Sample B examples each of x^{us} and x^s
  Append x^{us} to S_{us}
  Append x^s to S_s
  $c \leftarrow c + B$
  if $c \geq C$ **then**
    Compute R^l from Eq. (1) with S_{us} and S_s
    Reset Unsafe prompt set $S_{us} \leftarrow []$
    Reset Safe prompt set $S_s \leftarrow []$
    $c \leftarrow 0$
  end if
  if R^l is None **then**
    $\lambda \leftarrow 0$
  end if
  Compute $\mathcal{L}_{teacher}$ from Eq. (2)
  Update $W \leftarrow W - \eta \cdot \nabla \mathcal{L}_{teacher}$
end while

则化项的组合：

$$\mathcal{L}_{teacher} = \frac{1}{N}\sum_{i=1}^{N}\Big[\ell(x_i^s, y_i^s) + \ell(x_i^{us}, y_i^s) + \lambda\big\{\left\|1 + CS(f^l(x_i^s), R^l)\right\|_2 + \left\|1 - CS(f^l(x_i^{us}), R^l)\right\|_2\big\}\Big], \tag{2}$$

，其中 $\ell(\cdot,\cdot)$ 表示交叉熵损失，$CS(\cdot,\cdot)$ 表示余弦相似性，y^s 是每个提示的安全响应，N 是训练样本的数量。通过用方程 2 训练模型，ReFT 模型不仅可以通过其拒绝特征有效区分有害和无害提示，还可以为有害提示生成适当的拒绝响应。

此外，我们假设没有预对齐的模型可用，因此无法从对齐的模型中提取拒绝特征。为了解决这个问题，我们在 ReFT 模型的训练过程中每隔固定的时间间隔（周期）动态更新拒绝特征，使用方程 1 。在每个训练周期中，有害和无害的提示分别被积累到集合 S_{us} 和 S_s 中。然后基于这些收集的提示重新计算拒绝特征。在第一次更新周期之前的迭代中，我们设置 $\lambda = 0$ 来禁用正则化，因为拒绝特征尚未可靠建立。这种拒绝特征的动态更新消除了为获得拒绝特征而进行单独对齐阶段的需要，使模型能够在单一的训练过程中同时计算拒绝特征和学习判别特征，即使没有访问预对齐模型的情况下也是如此。教师准备阶段的完整算法在算法 1 中提供。

5.2 微调阶段

在微调阶段，ReFT 模型被冻结，并作为教师用于两个不同的目的：（1）过滤用户数据中的有害提示，（2）对齐知识蒸馏。该方法使基础模型能够在有效学习用户特定任务的同时保持较强的安全对齐。

首先，ReFT 模型被用于从用户数据中过滤有害的提示。利用其区分有害和无害提示的能力，我们通过测量拒绝特征与每个输入提示特征之间的余弦相似度来识别有害数据。如果相似度超过预定义的阈值，则将提示分类为有害，否则为无害。这种基于 ReFT 模型的过滤机制被公式化为一个二元过滤指标 ω_i ：

$$\omega_i = \begin{cases} 0, & \text{if} \quad CS(R^l, f^l(x_i)) > \tau \\ 1, & otherwise \end{cases}, \tag{3}$$

，其中 τ 是用于确定有害性的预定义阈值。在等式 3 中，通过设置 $\omega_i = 0$ ，将被分类为有害的提示从监督微调损失中排除，因为将有害提示误分类为无害可能会危及模型安全性。幸运的是，如表 7 所示，ReFT 模型将有害提示误分类为无害的频率低于将无害提示误分类为有害的频率。因此，我们丢弃所有被预测为有害的数据，确保微调仅在无害提示上进行。这一策略通过防止即使是少量有害数据的微调，有效地维护了模型的安全性对齐。

其次，ReFT 模型用于为对齐知识蒸馏提供软拒绝标签。由于我们的方法直接微调了一个未对齐的基础模型，而用户数据通常缺乏与相应拒绝响应相关的有害提示，单纯对用户数据进行微调无法保证模型的安全性。为了解决这一限制，我们重复使用在教师准备阶段原本使用的安全对齐数据。这一策略消除了专门为微调阶段收集额外对齐数据的需求，并确保 ReFT 模型已经在这些数据上进行训练，能够提供准确的拒绝响应作为软标签。这些软标签提供了更具信息性的监督，并相比硬标签贡献了更平滑的损失曲面。因此，使用对齐数据上的软拒绝标签微调基础模型能够有效进行安全对齐并与仅无害用户数据的监督微调损失无缝结合，使得模型能够可靠地学习对有害输入的安全和适当响应。总的来说，我们基于 ReFT 的微调策略融合了双教师机制。最终的微调阶段的损失函数定义为用户数据上的监督微调损失和安全对齐数据上的对齐蒸馏损失的组合：

$$\mathcal{L}_{ft} = \frac{1}{N_{user}}\sum_{i=1}^{N_{user}} \omega_i * \ell(x_i, y_i) + \alpha T^2 * \frac{1}{N_{align}}\sum_{i=1}^{N_{align}} \mathrm{KL}(p_{t,i}^T \,||\, p_{s,i}^T). \tag{4}$$

。在监督微调损失中，$\ell(x_i, y_i)$ 表示用户数据 (x_i, y_i) 上的交叉熵损失。在对齐蒸馏损失中，KL 表示在对齐数据上计算的 KL 散度，$p_{t,i}^T$ 和 $p_{s,i}^T$ 分别表示教师（ReFT）和学生（基础）模型的 softmax logits，具有温度 T 。这些定义为 $p_i^T = \frac{\exp(z_i/T)}{\sum_{j=1}^{V}\exp(z_j/T)}$ ，其中 z 表示模型的 logits，V 为词汇量大小。此外，超参数 α 控制蒸馏的权重。

Table 2: 在用户数据中，不同比例的有害提示 p 下的性能。较低的有害评分（↓）和较高的微调准确率（↑）表明更好的性能。结果取平均于种子 30、42 和 50。因为无害数据不可用，没有报告 $p = 1.0$ 的微调准确率。我们的基于 ReFT 的方法在所有有害比例下始终达到最佳性能。

Methods	Harmful Score (↓)					Finetune Accuracy (↑)				
	clean	$p = 0.1$	$p = 0.3$	$p = 0.5$	$p = 1.0$	clean	$p = 0.1$	$p = 0.3$	$p = 0.5$	$p = 1.0$
SFT	$2.2_{\pm 0.1}$	$16.2_{\pm 0.4}$	$57.3_{\pm 0.6}$	$71.3_{\pm 0.6}$	$76.7_{\pm 0.4}$	$41.1_{\pm 0.0}$	$39.9_{\pm 0.6}$	$39.1_{\pm 0.2}$	$37.1_{\pm 0.6}$	-
Repnoise [36]	$2.7_{\pm 0.4}$	$29.9_{\pm 0.6}$	$67.0_{\pm 5.1}$	$75.7_{\pm 3.1}$	$79.7_{\pm 0.6}$	$37.4_{\pm 0.3}$	$37.0_{\pm 1.2}$	$36.3_{\pm 0.7}$	$36.0_{\pm 1.4}$	-
Vaccine [16]	$1.3_{\pm 0.2}$	$5.4_{\pm 0.7}$	$35.0_{\pm 0.3}$	$57.5_{\pm 0.4}$	$81.3_{\pm 0.1}$	$22.9_{\pm 0.5}$	$23.2_{\pm 1.0}$	$21.7_{\pm 0.3}$	$20.3_{\pm 0.4}$	-
Booster [15]	$2.3_{\pm 0.1}$	$5.9_{\pm 0.2}$	$65.1_{\pm 0.3}$	$75.0_{\pm 0.6}$	$79.0_{\pm 0.4}$	$44.5_{\pm 0.5}$	$44.0_{\pm 0.9}$	$44.4_{\pm 0.6}$	$43.5_{\pm 0.6}$	-
LDIFS [30]	$1.0_{\pm 0.2}$	$4.1_{\pm 0.7}$	$7.1_{\pm 0.2}$	$14.7_{\pm 0.3}$	$24.0_{\pm 0.4}$	$18.0_{\pm 0.9}$	$16.7_{\pm 0.8}$	$15.5_{\pm 0.1}$	$15.4_{\pm 0.6}$	-
Lisa [14]	$1.4_{\pm 0.2}$	$5.3_{\pm 0.1}$	$25.9_{\pm 1.5}$	$49.2_{\pm 0.7}$	$67.3_{\pm 1.0}$	$38.3_{\pm 0.7}$	$38.9_{\pm 0.9}$	$37.8_{\pm 0.9}$	$36.2_{\pm 0.5}$	-
ReFT (Ours)	$0.9_{\pm 0.3}$	$1.0_{\pm 0.5}$	$0.6_{\pm 0.1}$	$0.9_{\pm 0.3}$	$1.3_{\pm 0.2}$	$48.8_{\pm 0.5}$	$49.0_{\pm 0.5}$	$45.5_{\pm 0.9}$	$44.8_{\pm 0.5}$	-

Table 3: 不同用户数据量的性能比较。我们的方法在有害评分和微调准确性上始终优于基线，突显其可扩展性。

Methods	Harmful Score (↓)					Finetune Accuracy (↑)				
	n=1000	n=1500	n=2000	n=2500	Average	n=1000	n=1500	n=2000	n=2500	Average
SFT	16.7	39.4	55.8	63.9	44.0	40.6	42.9	44.5	45.3	43.3
Repnoise [36]	30.4	50.4	61.7	72.9	53.9	38.4	40.5	43.6	43.5	41.5
Vaccine [16]	4.8	19.8	34.1	45.0	25.9	24.4	28.5	31.3	33.9	29.5
Booster [15]	5.9	19.4	48.2	62.6	34.0	43.4	45.3	48.4	48.5	46.4
LDIFS [30]	4.0	5.7	4.7	6.0	5.1	17.0	16.7	17.7	18.4	17.5
Lisa [14]	5.3	8.2	10.4	12.8	9.2	38.3	37.8	40.3	42.7	39.8
ReFT (Ours)	0.5	0.9	0.9	1.0	0.8	49.0	50.1	52.1	51.8	50.8

6 实验

我们评估了基于 ReFT 的微调策略在安全对齐和用户特定任务性能方面的有效性，涵盖各种实验设置。我们改变了有害提示的比例、用户数据的大小、无害提示的类型（GSM8K [7]，SST2 [39]，AGNEWS [51]，AlpacaEval [25]）以及基础模型（Llama3-8B [2]，Gemma2-9B [42]，Qwen2-7B [1]）。除非另有说明，我们使用了 Llama3-8B、0.1 的毒性比例、1000 用户数据，并以 GSM8K 作为无害数据。

数据集。在教师准备阶段，我们使用了来自 BeaverTails 数据集的 $N = 5,000$ 个有害提示及其对应的拒绝响应，以及来自 Alpaca 数据集的 $N = 5,000$ 个无害数据及其有帮助的响应。在微调阶段，用户数据是通过将有害和无害数据按特定比例混合构建的。在微调阶段使用的对齐数据量，用 N_{align} 表示，被设置为用户数据 N_{user} 的数量。尽管我们实验所用的所有有害提示均来自 BeaverTails 数据集，但我们在教师准备、微调和评估阶段使用了不同的子集，以防止任何数据重叠。

指标。为了评估安全对齐和用户特定任务的性能，我们采用了两个指标：有害分数（HS）和微调准确性（FA），遵循之前的研究 [16, 15, 14]。HS 定义为在从 BeaverTails 测试集中生成的 1,000 个输出中有害响应的比例，其有害性由预训练的审查模型 Beaver-Dam-7B [19] 进行分类。相比之下，FA 是通过对下游基准测试（包括 GSM8K、SST2、AGNEWS 和 AlpacaEval）的任务特定微调准确性来衡量的，使用其各自测试集中的 872 、1,000 、1,000 和 122 样本。对于 AlpacaEval，准确性由 GPT-4o 模型 [18] 评估，遵循标准评估惯例 [25]。值得注意的是，HS 和 FA 都是在微调阶段之后评估的。

6.1 实验结果

在不同有害提示比例下的鲁棒性。我们通过衡量用户数据中不同比例的有害提示 p（从完全清洁的数据 ($p = 0$) 到完全有害的数据 ($p = 1.0$)）下的有害评分和微调准确性，来评估基于 ReFT 的微调策略的有效性。表格 2 显示，我们的方法在所有 p 值中始终实现最低的有害评分和最高的微调准确性，优于所有基准。这种出色的性能源于我们使用 ReFT 模型进行的有效有害数据过滤，即使在用户数据包含大量有害提示的情况下，也允许基础模型仅在有用的用户数据上进行微调。此外，我们观察到像 RepNoise [36] 、Vaccine [16] 和 Booster [15] 这样的对齐阶段基准对高有害比例 ($p \geq 0.3$) 是脆弱的，这对应于恶意用户企图通过有害微调攻击破解 LLM 的场景。相比之下，微调阶段的解决方案如 LDIFS [30] 、Lisa [14] 和我们

Table 4: 在不同的下游数据集上进行微调的性能比较。结果表明，我们的微调策略具有很强的安全对齐和泛化性能。

Methods	GSM8K		SST2		AGNEWS		AlpacaEval		Average	
	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
SFT	16.7	40.6	33.5	93.4	28.2	82.8	23.7	32.7	20.4	49.9
Repnoise [36]	30.4	38.4	63.0	93.4	58.6	84.6	45.4	29.3	39.5	49.1
Vaccine [16]	4.8	24.4	35.8	90.0	29.5	83.2	55.8	14.4	25.2	42.4
Booster [15]	5.9	43.4	9.2	93.6	5.3	85.3	29.4	34.0	10.0	51.3
LDIFS [30]	4.0	17.0	14.6	90.5	12.5	71.2	5.7	33.7	7.4	42.5
Lisa [14]	5.3	38.3	21.4	93.4	14.9	84.5	10.1	29.6	10.3	49.2
ReFT (Ours)	0.5	49.0	1.3	94.5	1.2	86.1	2.4	34.6	1.1	52.8

Table 5: 跨不同模型架构的性能比较。我们的基于 ReFT 的微调策略在 Llama3-8B、Gemma2-9B 和 Qwen2-7B 中表现出较强的适应性。

Methods	Llama3-8B		Gemma2-9B		Qwen2-7B		Average	
	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
SFT	16.7	40.6	26.4	59.5	37.9	66.8	27.0	55.6
Repnoise [36]	30.4	38.4	26.2	57.1	25.4	63.7	27.3	53.1
Vaccine [16]	4.8	24.4	18.0	52.5	10.2	63.6	11.0	46.8
Booster [15]	5.9	43.4	2.3	58.4	4.9	70.0	4.4	57.3
LDIFS [30]	4.0	17.0	3.1	36.0	10.7	64.1	5.9	39.0
Lisa [14]	5.3	38.3	6.2	54.5	4.4	61.6	5.3	51.5
ReFT (Ours)	0.5	49.0	1.3	63.6	0.6	69.7	0.8	60.8

Table 6: 应用基于 ReFT 的微调对对齐阶段解决方案的影响。

Methods	HS ↓	FA ↑
SFT	16.7	40.6
SFT+ReFT	1.1	42.1
Repnoise [36]	30.4	38.4
Repnoise+ReFT	1.4	39.2
Vaccine [16]	4.8	24.4
Vaccine+ReFT	2.2	22.0
Booster [15]	5.9	43.4
Booster+ReFT	1.9	43.8

的方法即使在高有害比例下也保持鲁棒性，始终实现较低的有害评分。在这些方法中，我们的方法在安全对齐和用户特定的下游任务中取得了最佳性能。

在变化的用户数据数量下的可扩展性。为了评估性能如何随用户数据量的增加而变化，我们在用户数据样本数量从 500 增加到 2,000 的过程中测量了有害得分和微调精度。如表 3 所示，我们基于 ReFT 模型的微调策略在所有情况下始终实现了最佳性能，与表 2 的结果一致。具体而言，在给定的固定投毒比例下，即使有害提示的绝对数量随着用户数据规模的增加而按比例增加，我们的方法仍然保持较低的有害得分，显示出在安全对齐方面的强大鲁棒性。同时，随着越来越多的用户数据用于特定任务，微调精度得以提升。这些实验结果验证了我们的方法在不同数据规模下的可扩展性和适应性。

在多样化微调数据集上的泛化。在我们的默认设置中，我们使用 GSM8K 数据集作为用户特定的下游任务。为了进一步评估我们的微调策略在不同下游任务中的泛化能力，我们用来自 SST2、AGNEWS 和 AlpacaEval 数据集的样本替换用户数据中无害的部分。然后，我们测量了我们的方法和基线方法的有害评分和微调准确性。表格 4 显示，我们的方法在所有数据集上都实现了最低的有害评分和最高的微调准确性。这些结果证明了我们的基于 ReFT 的数据过滤的有效性，即使无害数据是从不同的分布中采样的。总体而言，我们的方法在广泛的用户特定任务上表现出强大的泛化能力，使其在现实场景中具有实际应用意义。

跨模型架构的适应性。我们通过在 Gemma2-9B 和 Qwen2-7B 架构上训练 ReFT 模型，同时使用训练好的 ReFT 模型对每个对应的基础模型进行微调，来评估我们微调策略对不同模型架构的适应性。表 5 显示，我们的方法可靠地减少了有害性并改善了跨架构的用户特定下游任务性能。这一强大的性能归因于我们基于 ReFT 的有害提示过滤机制在各个模型架构中的高分类准确性，如图 1 和表 1 所示。这些结果表明我们的方法不仅对特定模型架构有效，而且在各种 LLM 主干上具有良好的泛化性。

6.2 分析

增强对齐阶段解决方案的基于 ReFT 的微调策略。为了确定我们的基于 ReFT 的微调策略能否进一步提升对齐阶段技术中对齐模型的安全性和用户特定任务性能，我们在微调阶段将我们的方法应用于这些对齐模型，并测量有害得分和微调准确率。如表 6 所示，我们的方法在大多数情况下显著降低了有害得分，同时保持了相当的微调准确率。增强的安全与对齐表明，基于 ReFT 的数据过滤和对齐蒸馏可以补充对齐阶段的解决方案。

微调期间的分类准确率。除了在图 1 和表 1 中展示的结果外，我们评估在 GSM8K、SST2、AGNEWS 和 AlpacaEval 中微调期间有害和无害数据之间的分类准确率。表 7 报告称，我们基于 ReFT 的分类能够以近乎完美的准确率识别有害提示，并在各种数据集中实现对无害提示的高准确率。强大的分类准确率支持用户数据中可依赖的有害数据过滤，无论数据分布如何，防止有害提示被用于微调中。

Table 7: 微调期间对用户数据中有害和无害提示的分类准确率。

Datasets	Harmful	Harmless	Total
GSM8K	100.00 %	97.70 %	97.93 %
SST2	99.91 %	95.30 %	95.76 %
AGNEWS	99.91 %	99.86 %	99.87 %
AlpacaEval	99.90 %	77.04 %	79.33 %

ReFT 基础微调策略中各组件的影响。我们的 ReFT 基础微调方法由两个组件组成：使用 ReFT 模型进行的有害数据过滤（Filtering）和对齐蒸馏（AD）。为了评估它们各自和组合的贡献，我们通过选择性地去除每个组件来进行消融研究。当省略过滤时，模型在用户数据中的有害和无害提示上进行微调。当排除 AD 时，模型使用硬标记对齐数据进行训练。如表格 8 所示，通过防止模型在有害提示上进行微调，过滤提高了有害分数和微调准确性。相比之下，AD 通过提供软拒绝标签进一步增强了这些指标，这些标签平滑了对齐损失面，并为有害提示提供了更丰富的监督。因此，将过滤和 AD 结合，使我们的方法能够在保持安全对齐的同时，实现高用户特定任务性能。

Table 8: 消融研究评估过滤和 AD 的单独贡献。

Filtering	AD	HS ↓	FA ↑
X	X	14.0	48.7
O	X	0.7	47.3
X	O	8.9	46.6
O	O	0.5	49.0

超参数。用于识别我们基于 ReFT 的微调方法最优超参数的实验，包括循环次数 C 、正则化强度 λ 、阈值 τ 、蒸馏温度 T 、蒸馏权重 α 以及层索引 l ，详见补充材料。

局限性。我们的微调策略依赖于 ReFT 模型根据其拒绝特征准确分类有害和无害提示的能力。因此，如果开发出针对 ReFT 模型的对抗性攻击以破坏其拒绝特征，那么安全对齐可能会受到影响。

在这项工作中，我们解决了 Finetuning-as-a-Service 的安全风险，即 LLMs 的安全对齐可能因在用户数据中微调一些有害提示而受到影响。受到我们观察到的安全对齐模型的拒绝特征可以区分有害和无害提示的启发，我们引入了基于拒绝特征的教师（ReFT）模型，该模型可以准确识别有害提示并生成适当的拒绝响应。在微调过程中，ReFT 模型作为教师，进行数据过滤和对齐蒸馏。我们的大量实验表明，基于 ReFT 的微调策略在各种设置中始终实现最低的有害分数和最高的微调准确性，优于现有的基线。总之，我们的方法为安全且有效的 Finetuning-as-a-Service 提供了一个有前途的解决方案，确保高用户特定任务性能的同时保持安全对齐。

References

[1] Qwen2 technical report. 2024.

[2] AI@Meta. Llama 3 model card. 2024.

[3] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

[4] Somnath Banerjee, Sayan Layek, Soham Tripathy, Shanu Kumar, Animesh Mukherjee, and Rima Hazra. Safeinfer: Context adaptive decoding time safety alignment for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27188–27196, 2025.

[5] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*, 2023.

[6] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

[7] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[8] Yanrui Du, Sendong Zhao, Danyang Zhao, Ming Ma, Yuhan Chen, Liangyu Huo, Qing Yang, Dongliang Xu, and Bing Qin. Mogu: A framework for enhancing safety of open-sourced llms while preserving their usability. *arXiv preprint arXiv:2405.14488*, 2024.

[9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[10] Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: The silver lining of reducing safety risks when finetuning large language models. *Advances in Neural Information Processing Systems*, 37:65072–65094, 2024.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[12] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. *arXiv preprint arXiv:2403.00867*, 2024.

[13] Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Josh Kimball, and Ling Liu. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning. *arXiv preprint arXiv:2408.09600*, 2024.

[14] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:104521–104555, 2024.

[15] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. *arXiv preprint arXiv:2409.01586*, 2024.

[16] Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *arXiv preprint arXiv:2402.01109*, 2024.

[17] Kuo-Han Hung, Ching-Yun Ko, Ambrish Rawat, I Chung, Winston H Hsu, Pin-Yu Chen, et al. Attention tracker: Detecting prompt injection attacks in llms. *arXiv preprint arXiv:2411.00348*, 2024.

[18] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[19] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

[20] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023.

[21] Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *arXiv preprint arXiv:2310.20624*, 2023.

[22] Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. Salora: Safety-alignment preserved low-rank adaptation. *arXiv preprint arXiv:2501.01765*, 2025.

[23] Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety layers in aligned large language models: The key to llm security. *arXiv preprint arXiv:2408.17003*, 2024.

[24] Xiao Li, Zhuhong Li, Qiongxiu Li, Bingze Lee, Jinghao Cui, and Xiaolin Hu. Faster-gcg: Efficient discrete optimization jailbreak attacks against aligned large language models. *arXiv preprint arXiv:2410.15362*, 2024.

[25] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 5 2023.

[26] Guozhi Liu, Weiwei Lin, Tiansheng Huang, Ruichao Mo, Qi Mu, and Li Shen. Targeted vaccine: Safety alignment for large language models against harmful fine-tuning via layer-wise perturbation. *arXiv preprint arXiv:2410.09760*, 2024.

[27] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023.

[28] AI @ Meta Llama Team. The llama 3 herd of models, 2024.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[30] Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*, 2023.

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[32] et al. Qi. Constrain-sft: A supervised fine-tuning approach to enhance safety alignment in large language models. *Proceedings of NeurIPS 2024*, 37:95174, 2024.

[33] 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.

[34] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[35] LG Research, Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Yemuk Choi, Seokhee Hong, Junwon Hwang, Hyojin Jeon, Kijeong Jeon, et al. Exaone deep: Reasoning enhanced language models. *arXiv preprint arXiv:2503.12524*, 2025.

[36] Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Robie Gonzales, Subhabrata Majumdar, Hassan Sajjad, Frank Rudzicz, et al. Representation noising: A defence mechanism against harmful finetuning. *Advances in Neural Information Processing Systems*, 37:12636–12676, 2024.

[37] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Targeted latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv e-prints*, pages arXiv–2407, 2024.

[38] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

[39] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[40] Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*, 2024.

[41] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[42] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[43] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[44] Xunguang Wang, Daoyuan Wu, Zhenlan Ji, Zongjie Li, Pingchuan Ma, Shuai Wang, Yingjiu Li, Yang Liu, Ning Liu, and Juergen Rahmel. Selfdefend: Llms can defend themselves against jailbreaking in a practical manner. *arXiv preprint arXiv:2406.05498*, 2024.

[45] et al. Wei. Freeze: A method to preserve safety alignment during fine-tuning of large language models. *Proceedings of NeurIPS 2024*, 37:96357, 2024.

[46] Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. Efficient adversarial training in llms with continuous attacks. *arXiv preprint arXiv:2405.15589*, 2024.

[47] Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. *arXiv preprint arXiv:2402.13494*, 2024.

[48] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. *arXiv preprint arXiv:2402.08983*, 2024.

[49] Xin Yi, Shunfan Zheng, Linlin Wang, Gerard de Melo, Xiaoling Wang, and Liang He. Nlsr: Neuron-level safety realignment of large language models against harmful fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25706–25714, 2025.

[50] Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. Robust llm safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089*, 2024.

[51] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[52] Yuqi Zhang, Liang Ding, Lefei Zhang, and Dacheng Tao. Intention analysis makes llms a good jailbreak defender. *arXiv preprint arXiv:2401.06561*, 2024.

[53] Zhengyue Zhao, Xiaoyun Zhang, Kaidi Xu, Xing Hu, Rui Zhang, Zidong Du, Qi Guo, and Yunji Chen. Adversarial contrastive decoding: Boosting safety alignment of large language models via opposite prompt optimization. *arXiv preprint arXiv:2406.16743*, 2024.

[54] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[55] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

A

补充材料

在教师准备阶段，我们使用包含 5 个有害和 5 个无害提示的大小为 10 的批次训练拒绝特征引导教师（ReFT）模型 20 个周期，学习率为 $5e^{-4}$ 。在微调阶段，我们使用 ReFT 训练基础模型 20 个周期，采用 20 个批次（10 个有害数据和 10 个无害数据），学习率为 $1e^{-5}$ 。对于 AlpacaEval 数据集 [25] ，由于其规模较小，我们使用 700 个提示训练基础模型 100 个周期。在这两个阶段中，我们应用 LoRA [11] ，排名为 32，目标是注意力模块的查询、键和值组件。而且，我们使用 AdamW 优化器 [29] ，权重衰减为 0.1，并采用恒定的学习率计划。

A.1 我们方法的超参数

我们提出的方法引入了几个额外的超参数。首先，在教师准备阶段，我们将训练 ReFT 模型的正则化强度设置为 $\lambda = 0.1$ 。拒绝特征从 LLMs 的特定层提取：对于 LLAMA3-8B 是 $l = 12$ ，对于 Gemma2-9B 是 $l = 11$ ，对于 Qwen2-7B 是 $l = 18$ 。拒绝特征每隔 $C = 6$ 个周期定期更新，每次更新时使用 30 个有害的和 30 个无害的提示。在微调阶段，使用 ReFT 模型进行有害和无害分类时，我们使用阈值 0.9 来最大化有害提示的召回率。对于对齐知识蒸馏，我们设置蒸馏强度 $\alpha = 0.1$ 并使用温度 $T = 1$ 。识别这些超参数的最优值的消融研究在 Sec. B 中进行了展示。基线方法的所有其他超参数都遵循其各自原始论文中指定的设置 [15, 16, 30, 14, 36] 。

B 寻找最佳超参数的实验

B.1 拒绝特征提取的层选择

拒绝特性反映了模型区别有害和无害提示并仅对有害输入生成拒绝响应的能力。因此，最有效的方法是从能够最大化有害和无害提示表示之间区别的层中提取拒绝特性。根据一项之前的工作 [23] ，该工作建议这种层通常位于大型语言模型的早期到中层，我们通过评估分类准确率和有害与无害提示平均特征之间的范数差异在 8 个不同层中识别最佳层。如表 9 所示，分类准确率和范数差异在各层中有所不同。对于每个层，分类阈值进行了优化，以最大化分类性能。因此，在所有实验中，对于 Gemma2-9B [42] 模型，我们使用了 $l = 11$ ，对于 Qwen2-7B 模型，我们使用了 $l = 18$ 。对于 Llama3-7B，我们采用了 $l = 12$ ，遵循之前的工作 [3] 。此外，我们使用了对应于最后一个输入标记的特征，因为它由于语言模型的因果结构和注意力掩码而编码了整个句子。

在教师准备阶段，周期决定了用于更新标准拒绝特征的时间间隔和样本数量，这对于在我们的 ReFT 模型中区分有害和无害提示的特征起着重要的参考作用。短周期更频繁地更新标准拒绝特征，但使用的样本较少，这可能由于拒绝特征的差异导致训练不稳定。相反，长周期在每次更新时使用更多样本，但由于更新不频繁，可能导致对次优拒绝特征的过拟合。表 10 展示了在不同周期长度和用于更新标准拒绝特征的相应样本数量下，有害分数和微调精度。结果表明，短周期的频繁更新可以帮助 ReFT 模型更有效地将有害提示与无害提示区分开来，并对有害输入生成适当的拒绝响应。

B.2 正则化强度（λ）对 ReFT 模型训练的影响

主文稿中的公式 1 的 lambda 值控制正则化项的强度，正则化项在教师准备阶段鼓励 ReFT 模型中有害和无害提示特征的明显分离。过强的正则化项可能会破坏 ReFT 模型的内部表示，而过弱的正则化项可能会降低 ReFT 模型基于拒绝特征区分有害和无害提示的能力。因此，选择适当的 lambda 值对于有效训练 ReFT 模型和后续微调至关重要。表 11 展示了使用不同 lambda 值训练的 ReFT 模型的微调性能。结果表明，lambda 值为 0.1 时达到最低的有害得分和最高的微调准确性，表明其作为最佳超参数选择的有效性。

B.3 阈值对微调的影响

阈值是一个关键超参数，在微调阶段，通过测量输入提示特征与拒绝特征在 ReFT 模型中的相似性，作为分类有害提示的标准。我们预测相似性高于阈值的提示为有害，而低于阈值的提示为无害。因此，过低的阈值可能会错误地将有害提示归类为无害，从而在微调中引入安全风险。反之，过高的阈值可能会错误地过滤掉被误分类为有害的无害提示，导致微调精度

Table 9: 在 Gemma2-9B-it 和 Qwen2-7B-Instruct 中，用于辨识提取拒绝特征的最佳层索引的分类准确性和特征范数差异。在我们的实验中选定使用的层已经用粗体标出。对于每一层，特征从最后一个输入标记提取。

(a) Gemma2-9B-it

Layer idx	Threshold	Harmful Acc (%)	Harmless Acc (%)	Acc (%)	Harmful Avg	Harmless Avg	Diff
7	0.0055	76.6	93.4	85.0	0.0239	-0.0090	0.0329
8	0.0225	69.8	93.8	81.8	0.0374	0.0080	0.0294
9	0.0510	89.6	96.6	93.1	0.0878	0.0303	0.0575
10	0.0530	93.8	95.0	94.4	0.0949	0.0363	0.0586
11	0.0245	96.2	98.6	97.4	0.0844	-0.0020	0.0864
12	0.0555	91.4	96.4	93.9	0.1133	0.0319	0.0814
13	0.0570	90.8	92.8	91.8	0.1285	0.0346	0.0939
14	0.184	86.6	91.2	88.9	0.2629	0.1524	0.01105

(b) Qwen2-7B-Instruct

Layer idx	Threshold	Harmful Acc (%)	Harmless Acc (%)	Acc (%)	Harmful Avg	Harmless Avg	Diff
13	0.046	96.4	98.6	97.5	0.1814	0.0153	0.1661
14	0.118	97.2	97.8	97.5	0.2622	0.0875	0.1747
15	0.060	98.0	98.2	98.1	0.2297	0.0265	0.2032
16	0.145	96.2	99.2	97.7	0.3003	0.1093	0.1910
17	0.164	98.6	97.8	98.2	0.3709	0.1326	0.2383
18	0.195	98.6	99.8	99.2	0.4166	0.1551	0.2615
19	0.163	97.4	99.6	98.5	0.3555	0.1262	0.2293
20	0.055	95.0	99.4	97.2	0.2458	0.0211	0.2247

Table 10: 循环长度 (C) 对 ReFT 模型性能的影响。

Cycle	$N_{us} = N_s$	HS ($\downarrow$)	FA ($\uparrow$)
6	30	0.5	49.0
20	100	1.1	47.8
100	500	1.1	47.7
200	1000	1.2	46.8

Table 11: 改变 Lambda。

λ	HS ($\downarrow$)	FA ($\uparrow$)
0.05	0.7	48.4
0.1	0.5	49.0
0.3	1.0	48.3
0.5	1.0	48.3
1.0	1.6	47.7

Table 12: 变化的阈值。

Threshold	HS ($\downarrow$)	FA ($\uparrow$)
0	0.9	47.8
0.3	0.6	46.2
0.5	1.4	47.2
0.7	1.0	47.1
0.9	0.5	49.0

降低。如表 12 所示，我们评估了不同阈值对结果的影响。结果表明，阈值为 0.9 时，产生了最低的有害分数和最高的微调准确性。这一最佳性能归因于有害提示特征与拒绝特征的完美对齐，使得在 ReFT 模型中的相似性值接近 1，如主文档的图 1 所示。

B.4 蒸馏超参数的影响

知识蒸馏通常涉及两个关键的超参数：温度 T ，控制教师预测的软度，以及蒸馏权重 α ，平衡蒸馏损失的影响。为了评估它们的影响，我们测量了不同 T 和 α 值下的有害评分和微调准确性。如表 13 所示，较高的 T 值会导致有害评分增加，可能是因为学生模型没有紧密跟随 ReFT 模型的预测。相反，较高的 α 值会降低有害评分，但同时也会降低微调准确性，因为过度强调对齐损失会削弱用户特定下游任务的性能。在这些超参数值中，$T = 1$ 和 $\alpha = 0.1$ 带来了最佳的整体性能。这种设置使得学生模型可以紧密地跟随 ReFT 模型精确对齐的拒绝响应，同时保持适度的对齐损失以保留下游任务性能。

C 附加实验

C.1 使用对齐的 LLM 作为 ReFT 模型

我们的设置假定无法获得对齐的 LLM，因此我们在教师准备阶段独立训练 ReFT 模型。然而，在实际场景中，许多对齐模型已经存在，例如 Llama3-8B-Instruct [2] 、Gemma2-9B-it [42] 和 Qwen2-7B-Instruct [1] 。为了评估使用对齐 LLM 作为 ReFT 模型的潜力，我们测量了当使用对齐 LLM 作为 ReFT 模型和基模型时的有害分数和微调准确率。结果表明，表 14 证明了对齐 LLM 可以支持分类有害提示并提取对齐知识，与它们的零次性能相比，有害分数和微调准确率都有所提升。然而，它们的分类准确率仍不理想，限制了性能的提升。此外，表 14 指出了 LlamaGuard3 [28] ，一种专门用于分类有害和无害提示的模型，也可以用作

Table 13: 温度 (T) 和蒸馏权重 (α) 对有害分数 (HS) 和微调精度 (FA) 的影响。最佳设置 ($T = 1.0$, $\alpha = 0.1$) 以粗体显示。

Temperature T	α	HS ($\downarrow$)	FA ($\uparrow$)
1.0	0.1	0.5	49.0
1.0	0.3	1.3	45.3
1.0	0.5	1.2	47.9
1.0	1.0	1.2	44.6
1.0	5.0	0.9	40.5
2.0	0.1	0.9	45.6
2.0	0.3	0.7	44.2
2.0	0.5	1.0	43.4
2.0	1.0	0.5	42.8
2.0	5.0	0.6	26.1
5.0	0.1	12.8	46.7
5.0	0.3	3.4	46.5
5.0	0.5	3.1	45.2
5.0	1.0	2.2	44.2
5.0	5.0	2.4	33.7

Table 14: 将已对齐的大型语言模型用作 ReFT 模型的性能与其零样本性能的比较。使用已对齐的大型语言模型作为 ReFT 模型可以改善安全性和任务性能，但增益因模型而异。

Aligned Model	HS ($\downarrow$)	FA ($\uparrow$)
Llama3-8B (zero-shot)	74.6	14.2
LlamaGuard3 (ReFT)	7.4	49.5
Llama3-8B-Instruct (zero-shot)	18.7	60.7
Llama3-8B-Instruct (ReFT)	13.9	65.8
Gemma2-9B-it (zero-shot)	5.9	74.3
Gemma2-9B-it (ReFT)	4.9	72.4
Qwen2-7B-Instruct (zero-shot)	22.8	33.9
Qwen2-7B-Instruct (ReFT)	20.6	73.2

ReFT 模型。为了进行此实验，Llama3-8B 被用作基模型。这些发现突出了使用现有对齐模型作为 ReFT 模型的实际可行性以及独立的教师准备阶段对于最大化分类性能的重要性。

在破解大型语言模型（LLMs）时，可以使用高级技术如 GCG（贪婪坐标梯度）以及 AutoDAN（自动生成类似 DAN 系列的破解提示），以诱导产生有害响应，而不仅仅是简单地提示有害查询。与直接的有害提示相比，这些方法在诱导产生有害响应方面表现出高攻击成功率，即便是在对齐模型中。为了评估我们的基于 ReFT 微调策略在此类高级破解攻击下的稳健性，我们在黑箱环境中测量 Llama3-8B-Instruct 模型的在 GCG 和 AutoDAN 攻击下的有害分数。尽管所有方法在这些高级攻击下都表现出增加的有害分数，表 15 显示我们的基于 ReFT 的微调方法比基准方法更加稳健。值得注意的是，尽管 LDIFS 方法在 GCG 攻击下获得了较低的有害分数，但其微调精确度较差，并且在 AutoDAN 攻击下有害分数较高，支持了该方法在实际应用中的不切实际性。相比之下，我们的方法在 GCG 和 AutoDAN 攻击下都保持了低有害分数和高微调精度，证明了其在对抗日益复杂的破解尝试中提供可靠保护的有效性。

我们对在微调阶段使用的 GSM8K、SST2 和 AGNEWS 数据集进行了额外的分析。类似于主文中的图 1 和表 1，我们使用 BeaverTails 作为有害数据，而 GSM8K、SST2 和 AGNEWS 作为无害数据，选用了 Llama3-8B 作为模型架构。图 2 展示了输入提示特征与拒绝特征之间的相似性分布，而表 16 报告了使用相应最佳阈值对每个数据集进行分类的准确率。与含有通用提示的 Alpaca 数据集不同，下游任务数据集（GSM8K, SST2, AGNEWS）由特定领域的提示组成，因此在数据分布上与 BeaverTails 不同。因此，即使在基础模型中，这些数据集也略有可区分性。然而，对齐后的模型显示出更明显的相似性分布和更高的分类准确率，而 ReFT 模型实现了最清晰的分离和最高的分类性能。这些结果与主文中的图 1 和表 1 一致，进一步支持了我们分析的泛化性和可靠性。

References

[1] Qwen2 technical report. 2024.

[2] AI@Meta. Llama 3 model card. 2024.

Table 15: 微调过程中不同越狱攻击的性能比较。GCG 攻击是使用 BeaverTails 数据集中的 100 个样本生成的 [19]，而 AutoDAN 攻击是使用 AdvBench 数据集中的 520 个样本生成的 [55]。结果表明，我们基于 ReFT 的微调策略具有强大的安全对齐和泛化能力，并且始终优于所有基线。

Methods	BeaverTails [19]		GCG [55]		AutoDAN [27]		Average	
	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
SFT	16.7	40.6	36.0	40.6	69.6	40.6	40.8	40.6
Repnoise [36]	30.4	38.4	46.0	38.4	68.5	38.4	48.3	38.4
Vaccine [16]	4.8	24.4	16.0	24.4	18.3	24.4	10.4	24.4
Booster [15]	5.9	43.4	10.0	43.4	37.1	43.4	17.7	43.4
LDIFS [30]	4.0	17.0	4.0	17.0	61.9	17.0	23.3	17.0
Lisa [14]	5.3	38.3	52.0	38.3	41.5	38.3	32.9	38.3
ReFT (Ours)	0.5	49.0	6.0	49.0	0.9	49.0	2.5	49.0

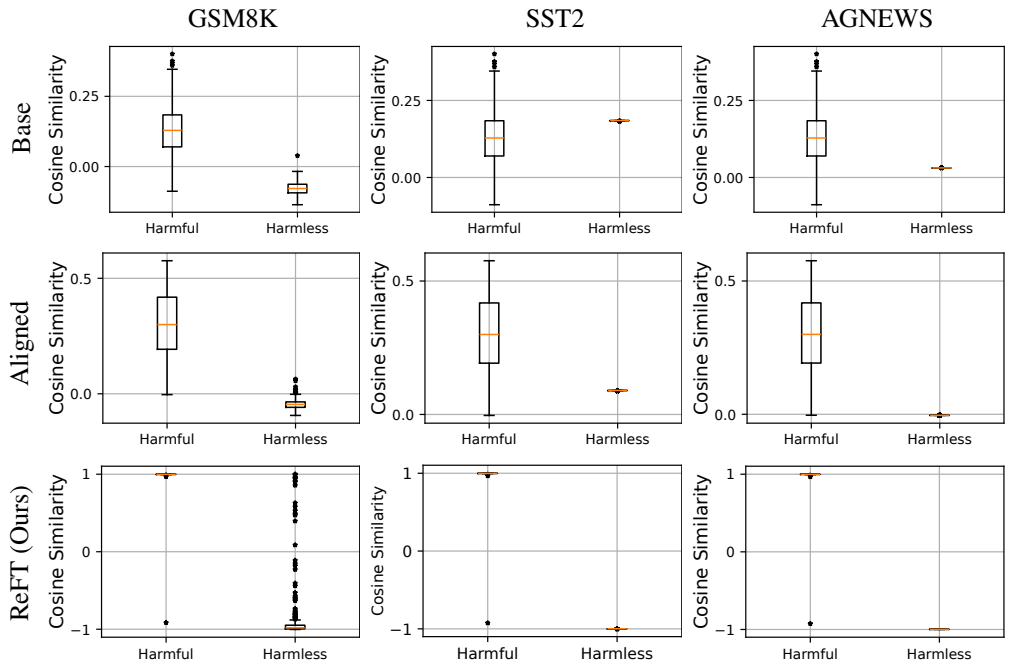

Figure 2: 在基础模型、对齐模型和我们的 ReFT 模型上，针对有害和无害提示的余弦相似度分布的箱线图。有害提示从 BeaverTails 数据集中采样（$n = 500$），而无害提示从 GSM8K、SST2 和 AGNEWS 数据集中采样（$n = 500$），这些数据集是在微调阶段使用的领域特定的下游任务数据集。

Table 16: 使用拒绝特征进行分类准确性。余弦相似度高于阈值的提示被识别为有害，而低于阈值的则被识别为无害。通过优化阈值以最大化总体分类准确性。

Datasets	Model	Threshold	Harmful Acc	Harmless Acc	Total Acc
GSM8K	Llama3-8B	-0.017	95.6 %	99.8 %	97.7 %
	Llama3-8B-Instruct	0.035	98.2 %	99.6 %	98.9 %
	Llama3-8B-ReFT	0.965	99.8 %	99.2 %	99.5 %
SST2	Llama3-8B	0.190	22.6 %	100.0 %	61.3 %
	Llama3-8B-Instruct	0.095	89.6 %	100.0 %	94.8 %
	Llama3-8B-ReFT	-0.920	100.0 %	100.0 %	100.0 %
AGNEWS	Llama3-8B	0.032	86.0 %	100.0 %	93.0 %
	Llama3-8B-Instruct	0.010	99.8 %	100.0 %	99.9 %
	Llama3-8B-ReFT	-0.990	100.0 %	100.0 %	100.0 %

[3] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

[4] Somnath Banerjee, Sayan Layek, Soham Tripathy, Shanu Kumar, Animesh Mukherjee, and Rima Hazra. Safeinfer: Context adaptive decoding time safety alignment for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27188–27196, 2025.

[5] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*, 2023.

[6] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

[7] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[8] Yanrui Du, Sendong Zhao, Danyang Zhao, Ming Ma, Yuhan Chen, Liangyu Huo, Qing Yang, Dongliang Xu, and Bing Qin. Mogu: A framework for enhancing safety of open-sourced llms while preserving their usability. *arXiv preprint arXiv:2405.14488*, 2024.

[9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[10] Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: The silver lining of reducing safety risks when finetuning large language models. *Advances in Neural Information Processing Systems*, 37:65072–65094, 2024.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[12] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. *arXiv preprint arXiv:2403.00867*, 2024.

[13] Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Josh Kimball, and Ling Liu. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning. *arXiv preprint arXiv:2408.09600*, 2024.

[14] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:104521–104555, 2024.

[15] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. *arXiv preprint arXiv:2409.01586*, 2024.

[16] Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *arXiv preprint arXiv:2402.01109*, 2024.

[17] Kuo-Han Hung, Ching-Yun Ko, Ambrish Rawat, I Chung, Winston H Hsu, Pin-Yu Chen, et al. Attention tracker: Detecting prompt injection attacks in llms. *arXiv preprint arXiv:2411.00348*, 2024.

[18] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[19] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

[20] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023.

[21] Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *arXiv preprint arXiv:2310.20624*, 2023.

[22] Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. Salora: Safety-alignment preserved low-rank adaptation. *arXiv preprint arXiv:2501.01765*, 2025.

[23] Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety layers in aligned large language models: The key to llm security. *arXiv preprint arXiv:2408.17003*, 2024.

[24] Xiao Li, Zhuhong Li, Qiongxiu Li, Bingze Lee, Jinghao Cui, and Xiaolin Hu. Faster-gcg: Efficient discrete optimization jailbreak attacks against aligned large language models. *arXiv preprint arXiv:2410.15362*, 2024.

[25] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 5 2023.

[26] Guozhi Liu, Weiwei Lin, Tiansheng Huang, Ruichao Mo, Qi Mu, and Li Shen. Targeted vaccine: Safety alignment for large language models against harmful fine-tuning via layer-wise perturbation. *arXiv preprint arXiv:2410.09760*, 2024.

[27] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023.

[28] AI @ Meta Llama Team. The llama 3 herd of models, 2024.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[30] Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*, 2023.

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[32] et al. Qi. Constrain-sft: A supervised fine-tuning approach to enhance safety alignment in large language models. *Proceedings of NeurIPS 2024*, 37:95174, 2024.

[33] 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.

[34] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[35] LG Research, Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Yemuk Choi, Seokhee Hong, Junwon Hwang, Hyojin Jeon, Kijeong Jeon, et al. Exaone deep: Reasoning enhanced language models. *arXiv preprint arXiv:2503.12524*, 2025.

[36] Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Robie Gonzales, Subhabrata Majumdar, Hassan Sajjad, Frank Rudzicz, et al. Representation noising: A defence mechanism against harmful finetuning. *Advances in Neural Information Processing Systems*, 37:12636–12676, 2024.

[37] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Targeted latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv e-prints*, pages arXiv–2407, 2024.

[38] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

[39] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[40] Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*, 2024.

[41] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[42] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[43] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[44] Xunguang Wang, Daoyuan Wu, Zhenlan Ji, Zongjie Li, Pingchuan Ma, Shuai Wang, Yingjiu Li, Yang Liu, Ning Liu, and Juergen Rahmel. Selfdefend: Llms can defend themselves against jailbreaking in a practical manner. *arXiv preprint arXiv:2406.05498*, 2024.

[45] et al. Wei. Freeze: A method to preserve safety alignment during fine-tuning of large language models. *Proceedings of NeurIPS 2024*, 37:96357, 2024.

[46] Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. Efficient adversarial training in llms with continuous attacks. *arXiv preprint arXiv:2405.15589*, 2024.

[47] Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. *arXiv preprint arXiv:2402.13494*, 2024.

[48] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. *arXiv preprint arXiv:2402.08983*, 2024.

[49] Xin Yi, Shunfan Zheng, Linlin Wang, Gerard de Melo, Xiaoling Wang, and Liang He. Nlsr: Neuron-level safety realignment of large language models against harmful fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25706–25714, 2025.

[50] Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. Robust llm safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089*, 2024.

[51] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[52] Yuqi Zhang, Liang Ding, Lefei Zhang, and Dacheng Tao. Intention analysis makes llms a good jailbreak defender. *arXiv preprint arXiv:2401.06561*, 2024.

[53] Zhengyue Zhao, Xiaoyun Zhang, Kaidi Xu, Xing Hu, Rui Zhang, Zidong Du, Qi Guo, and Yunji Chen. Adversarial contrastive decoding: Boosting safety alignment of large language models via opposite prompt optimization. *arXiv preprint arXiv:2406.16743*, 2024.

[54] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[55] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.