Model Immunization Beyond Condition Numbers: The Importance of Plateau Regions

Quoc Nguyen[1] **Trung Le**[2] **Dinh Phung**[2] **Mehrtash Harandi**[1]
[1]Department of Electrical and Computer Systems Engineering, Monash University, Australia
[2]Department of Data Science and AI, Monash University, Australia
{minh.nguyen12, trunglm, dinh.phung, mehrtash.harandi}@monash.edu

Abstract

Harmful fine-tuning can rapidly compromise safety alignment in Large Language Models, underscoring the vulnerability of deployed models to post-release modification. While prior work motivates model immunization by inducing ill-conditioned curvature to slow harmful optimization, this approach is not scalable due to the high computational cost of Hessian estimation. We refine this paradigm by extending the theoretical framework to prioritize plateau regions alongside curvature. We further introduce an immunization algorithm based on efficient Hessian estimation and bounds on the extremal singular values of the Hessian. We demonstrate the proposed immunization framework in three applications: (i) Fine-tuning-as-a-service settings where benign user data may be mixed with poisoned harmful data, (ii) open-weight settings where an adversary can fine-tune on harmful-only data with varied training dynamics, and (iii) robust unlearning, where immunized models better resist relearning attacks after unlearning. Extensive experiments confirm that our approach achieves robust defense against adversarial fine-tuning while largely maintaining general utility and downstream trainability. Our code is available here.

1 Introduction

Large language models (LLMs) are increasingly deployed and distributed in settings where *downstream fine-tuning* is a core feature. Service providers offer fine-tuning-as-a-service (FTaaS) (*e.g.*, OpenAI[1]) and open-weight releases enable broad post-release adaptation. This flexibility creates a critical safety vulnerability. Even when a model is safety-aligned at release time, a small amount of harmful data can rapidly shift behavior via fine-tuning, enabling misuse and circumventing safeguards [1–3]. This motivates *pre-release* defenses that make harmful adaptation substantially harder.

Model immunization [4–7] is a framework to enable this defense by making a released model harder to fine-tune toward a target task while preserving its general capabilities and adaptation to benign tasks. In our main setting, the target task is harmful behavior, so the goal is to increase resistance to harmful fine-tuning. Crucially, model immunization must operate under realistic threat models where an attacker can control the fine-tuning hyperparameters and use a certain training budget.

A prominent line of work formalizes immunization through *loss curvature*, arguing that making the harmful objective ill-conditioned, meaning that it has a large Hessian condition number, slows gradient-based fine-tuning [7, 8]. While novel, the condition-number perspective can be overstated in ways that matter for practice. First, the convergence claim typically leverages a bound derived under strong convexity and specific step-size choices. As we emphasize in Section 3.2, a large condition number is a *necessary but not sufficient* condition for slow convergence, since a slowly decreasing upper bound does not imply that the iterates cannot still make rapid progress. Second, our

[1]https://platform.openai.com/docs/guides/model-optimization

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

analysis shows that slow, harmful fine-tuning depends jointly on curvature and local steepness, since enlarging the Hessian condition number alone still allows fast loss decreases when harmful gradients remain large near the released parameters. Finally, given that estimating the largest and smallest Hessian singular values, and thus the condition number, at scale is nontrivial, existing immunization frameworks [8] neither propose an efficient method for this nor evaluate it in large models such as LLMs.

Building on the theory developed in Section 3.2, we argue that reliably slowing harmful optimization requires controlling *both* (i) ill-conditioned curvature (large Hessian condition number κ) and (ii) *plateau behavior* around the released parameters, characterized by small harmful gradient norms. Our bounds explicitly couple curvature with local gradient magnitude, highlighting that maximizing condition number alone leaves an attack surface if the harmful objective remains steep near the aligned solution. This perspective leads to a refined framework and algorithm that promotes harmful plateaus while also efficiently enlarging condition numbers.

We summarize our contributions as follows:

- **A refined immunization framework.** We refine the previous model immunization by condition number framework [8] by showing that convergence guarantees depend jointly on the Hessian condition number and harmful gradient norms (via tighter upper/lower bounds in Section 3.2). This yields a clearer characterization of when harmful fine-tuning is actually slow, and motivates *plateau promotion* as a complementary immunization objective.
- **An efficient algorithm.** We propose **BECON**, which provides an optimization objective that (i) promotes harmful plateaus using harmful gradient perturbation and (ii) enlarges condition numbers using an efficient Hessian proxy with tractable estimates of extremal singular values, making the approach feasible for large models.
- **Evaluation in different applications.** We empirically demonstrate BECON's effectiveness in various applications, including increasing resistance to harmful fine-tuning in FTaaS and open-weight settings while largely retaining benign utility and trainability, and robustness against relearning attacks in unlearning settings.

2 Related Work

Harmful Fine-tuning Defense. Recent research demonstrates that safety alignment can be subverted by fine-tuning models on a limited number of harmful samples [1, 9, 2, 10]. In response, various defense strategies have been developed to address different stages of the model lifecycle [11]. This paper focuses on the immunization framework, categorized as an alignment-stage defense [4, 11]. Alignment-stage defenses make the model inherently resilient to malicious fine-tuning before release, and since this intervention occurs prior to fine-tuning, it applies to both open-weight models and FTaaS settings. Existing methods proactively reinforce alignment to resist harmful fine-tuning using diverse strategies. For instance, Vaccine [12] and Targeted-Vaccine [13] constrain embedding shifts via perturbation-aware training, while Booster [10] and TAR [6] seek safety-resilient local optima. Other approaches, such as RMU [14], CTRAP [15], and RepNoise [5], utilize unlearning and noise approximation to erase or hinder training on harmful knowledge. Furthermore, VAA [16] enhances safety alignment by identifying and upweighting vulnerable alignment data, as a subset of alignment data is more prone to forgetting, and LoX [17] utilizes a training-free method to reinforce safety by extrapolating the safety subspace of an aligned LLM.

Relearning Attacks in LLM Unlearning. Recent studies [18–23] demonstrate that LLMs subjected to unlearning algorithms can recover forgotten knowledge after fine-tuning on the original data or implicitly related samples. Mechanistic studies [23, 24] reveal that many unlearning methods remain vulnerable to recovery via fine-tuning, suggesting that they often suppress or hide access to the targeted knowledge rather than removing it from the model weights. Recent work [25] has proposed a solution based on Sharpness-Aware Minimization (SAM) [26], which improves robustness by enforcing the unlearned model to converge to a flat region of the loss landscape, thereby hindering the optimization process of relearning attacks.

Model Immunization with Condition Number. Prior work has primarily studied the condition number as a measure of numerical stability and optimization robustness. In this line of research, minimizing the condition number has been shown to improve training stability and robustness to

noise [27, 28]. In contrast, recent studies on *model immunization* deliberately take the opposite approach: they seek to *maximize* the Hessian condition number of unsafe objectives in order to hinder or slow harmful fine-tuning. One line of work introduces a framework based on this condition number maximization strategy and validates it using linear and vision architectures [8]. Concurrent work explores a similar objective by directly increasing the largest singular values of the model matrices to increase the lower bound of the Hessian's maximum singular value [7]. Building on these foundations, our proposed method improves the theoretical aspects of condition-number-based immunization [8] and extends its practicality with an efficient algorithm applicable to large-scale settings such as LLMs. Complementarily, Rosati et al. [29] establishes that purely spectral-based defenses face fundamental limitations, a finding consistent with our analysis showing that resistance depends jointly on the condition number and harmful loss plateaus.

3 Approach

3.1 Condition Numbers and Model Immunization

Immunization via the condition number of the Hessian is rooted in classical convergence results for gradient-based optimization [8]. The key idea in [8] builds on a well-known result from optimization theory. Consider the optimization problem $\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ where $\mathcal{L}$ is a strongly convex function and has a Hessian $\boldsymbol{H}(\boldsymbol{\theta}) = \nabla^2 \mathcal{L}(\boldsymbol{\theta})$. Let $\sigma_{\min}$ and $\sigma_{\max}$ denote the minimum and maximum singular values of the Hessian evaluated at the current model parameters, with the parameter dependence suppressed for readability. The constant step-size steepest descent algorithm has the following convergence rate [30]:

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|^2 \leq \left(1 - \frac{\sigma_{\min}}{\sigma_{\max}}\right)^t \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|^2, \tag{1}$$

where $\boldsymbol{\theta}^*$ denotes the optimal solution, and $\boldsymbol{\theta}_t$ denotes the steepest descent iterate at step t.

Defining the condition number $\kappa(\boldsymbol{H}) \coloneqq \frac{\sigma_{\max}}{\sigma_{\min}}$, the strong convexity assumption ensures a strictly positive lower curvature bound, so $\sigma_{\min} > 0$ and $\kappa(\boldsymbol{H})$ is well-defined. A large $\kappa(\boldsymbol{H})$ implies the convergence factor $1 - 1/\kappa \approx 1$, potentially leading to very slow convergence as discussed in [8]. While this observation is compelling, it should be interpreted with caution: *Since* (1) *is an upper bound, a slow decrease of the bound does not strictly imply slow convergence of the iterates.* Thus, a high condition number is a necessary but not sufficient indicator for poor optimization performance, motivating the need for more refined theoretical analysis.

3.2 Model Immunization Beyond Condition Numbers

We consider the objective function $\mathcal{L}(\boldsymbol{\theta})$ (i.e., the empirical loss over a training set), which satisfies the following assumptions:

(A1) The objective function $\mathcal{L}(\boldsymbol{\theta})$ has the Hessian matrix $\boldsymbol{H}$ satisfying: $\sigma_{\min}\boldsymbol{I} \prec \boldsymbol{H} \prec \sigma_{\max}\boldsymbol{I}$ for the positive min/max singular values $\sigma_{\min}/\sigma_{\max}$.

(A2) There exists a global minima: $\boldsymbol{\theta}^* = \text{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ and $\mathcal{L}^* = \mathcal{L}(\boldsymbol{\theta}^*)$.

We first prove that the Polyak–Lojasiewicz (PL) inequality holds for $\sigma_{\min}$ in the following lemma.

Lemma 3.1. *We have the following inequalities*

(i) The PL-typed inequality holds for $\sigma_{\min}$:

$$\frac{1}{2}\|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2 \geq \sigma_{\min}\left(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*\right), \quad \forall\, \boldsymbol{\theta}, \tag{2}$$

where $\mathcal{L}^* = \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$.

(ii) The opposite PL-type inequality holds for $\sigma_{\max}$:

$$\frac{1}{2}\|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2 \leq \sigma_{\max}\left(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*\right), \quad \forall\, \boldsymbol{\theta}. \tag{3}$$

Denote $\boldsymbol{\theta}_t$ as the solution (i.e., the model) at the time step t. Assuming that we use gradient descent to update a new model: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)$ with the learning rate $\eta > 0$, the following lemma specifies the range of the learning rate η for efficient and useful updates.

Lemma 3.2. *The update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)$ is efficient in reducing the new objective $\mathcal{L}(\boldsymbol{\theta}_{t+1})$ (i.e., $\mathcal{L}(\boldsymbol{\theta}_{t+1}) < \mathcal{L}(\boldsymbol{\theta}_t)$) if $0 < \eta < \frac{2}{\sigma_{\max}}$.*

In what follows, we develop necessary inequalities to indicate the quantities that slow down the convergence of the OP of interest, which can be applied directly to the model immunization.

Theorem 3.3. *Considering the learning rate $\eta \in \left(0, \frac{2}{\sigma_{\max}}\right)$, we have the following inequalities:*

(i) We can upper-bound $\mathcal{L}(\boldsymbol{\theta}_{t+1}) - \mathcal{L}^*$ by

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}) - \mathcal{L}^* \leq \left(1 - \frac{1}{\kappa}\right)^{t+1} (\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*) + \frac{(\sigma_{\max}\eta - 1)^2}{2\sigma_{\max}} \sum_{i=1}^{t} \left(1 - \frac{1}{\kappa}\right)^{t-i} \|\nabla \mathcal{L}(\boldsymbol{\theta}_i)\|^2. \quad (4)$$

(ii) We can lower-bound $\mathcal{L}(\boldsymbol{\theta}_{t+1}) - \mathcal{L}^*$ by

□ *If $\kappa > 2$ and $\eta \in \left[\frac{1}{\kappa\sigma_{\max}}, \frac{2}{\sigma_{\max}}\right)$, we have the lower bound*

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}_{t+1}) - \mathcal{L}^* \geq & \left(1 - \frac{1}{\kappa}\right)^{3(t+1)} (\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*) \\ & - \left(\eta - \frac{1}{\kappa\sigma_{\max}}\right) \left(1 - \frac{1}{\kappa^2}\right) \sum_{i=1}^{t} \left(1 - \frac{1}{\kappa}\right)^{3(t-i)} \|\nabla \mathcal{L}(\boldsymbol{\theta}_i)\|^2. \end{aligned} \quad (5)$$

□ *If the learning rate $0 < \eta < \frac{1}{\kappa\sigma_{\max}}$, we have the lower bound*

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}) - \mathcal{L}^* \geq \left(1 - \frac{1}{\kappa}\right)^{t+1} (\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*) - \frac{\eta}{2} \left(1 - \frac{\sigma_{\min}\eta}{2}\right) \sum_{i=1}^{t} \left(1 - \frac{1}{\kappa}\right)^{t-i} \|\nabla \mathcal{L}(\boldsymbol{\theta}_i)\|^2. \quad (6)$$

Theorem 3.3 develops the lower and upper bounds for the *convergence gap* $\mathcal{L}(\boldsymbol{\theta}_{t+1}) - \mathcal{L}^*$. From the lower bounds in (5) and (6), we observe that to increase the *convergence gap* for slowing down convergence, we need to (i) minimize $\kappa^{-1} = \frac{\sigma_{\min}}{\sigma_{\max}}$ (or maximize the condition number $\kappa = \frac{\sigma_{\max}}{\sigma_{\min}}$) and (ii) minimize the gradient norms $\|\nabla \mathcal{L}(\boldsymbol{\theta}_i)\|, i = 1, ..., t$. Moreover, according to the upper bound (4), when minimizing the gradient norms, the upper bound is closer to the lower bound, implying a tighter approximation. Additionally, by setting the learning rate $\eta = \frac{1}{\sigma_{\max}}$, our upper bound reduces to the bound in (1), indicating that our upper bound is more general than the one in (1), which only considers the constant learning rate $\eta = \frac{1}{\sigma_{\max}}$. The proofs of these lemmas and theorems are presented in Appendix B. Finally, we provide a discussion in Appendix H of why small harmful gradient norms do not imply that the model is close to the harmful optimum.

4 Proposed Method

Our primary objective is to immunize LLMs against adaptation to harmful datasets in particular, and to undesired target datasets in general. Specifically, we aim to configure the safety alignment phase such that downstream users cannot effectively fine-tune the model on harmful datasets $\mathcal{D}_{\mathrm{H}}$ after the aligned model is released for fine-tuning. Motivated by the theoretical insights in Section 3.2, our objective has two complementary parts: harmful plateau promotion and condition number enlargement.

Harmful plateau promotion. Following Theorem 3.3, we encourage *plateau behavior* on harmful data, so that a single step of harmful fine-tuning yields only a small decrease in harmful loss. Let $\mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta})$ be the harmful loss, and define the ℓ_2 ball $\mathcal{B}_\rho(\boldsymbol{\theta}) := \{\boldsymbol{\phi} : \|\boldsymbol{\phi} - \boldsymbol{\theta}\|_2 \leq \rho\}$. Intuitively, the plateau objective penalizes how much the harmful loss can decrease under a small update within this ball. We approximate this local harmful improvement by a single normalized harmful-gradient step, which lies on the boundary of $\mathcal{B}_\rho(\boldsymbol{\theta})$:

$$\boldsymbol{\theta}_{\text{pert}} \coloneqq \boldsymbol{\theta} - \rho \, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta}) / \|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta})\|_2. \quad (7)$$

If the harmful landscape is locally flat along this attack direction, then moving to $\boldsymbol{\theta}_{\text{pert}}$ yields little improvement in harmful loss. For small ρ, we can explicitly derive the connection between this

one-step improvement and gradient suppression via a first-order Taylor expansion:

$$\mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta}_{\text{pert}}) \approx \mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta})^{\top}(\boldsymbol{\theta}_{\text{pert}} - \boldsymbol{\theta}) = \mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta}) - \rho\frac{\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta})\|_2^2}{\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta})\|_2} = \mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta}) - \rho\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta})\|_2. \tag{8}$$

Thus, discouraging this one-step harmful loss decrease directly promotes small harmful gradients.

In practice, per-sample losses within a harmful minibatch can vary significantly. To ensure the plateau objective prioritizes samples that are most sensitive to the harmful update, we weight examples by the change in their losses under the perturbation. For a harmful minibatch $\{\boldsymbol{x}_i\}_{i=1}^{B}$ with per-sample losses $\ell_i(\boldsymbol{\theta})$, define:

$$\Delta_i \coloneqq \left|\ell_i(\boldsymbol{\theta}) - \ell_i(\boldsymbol{\theta}_{\text{pert}})\right|, \qquad w_i \;\coloneqq\; \frac{\exp\big(\Delta_i/\tau\big)}{\sum_{j=1}^{B}\exp\big(\Delta_j/\tau\big)}, \tag{9}$$

where $\tau > 0$ is the softmax temperature, controlling how sharply we emphasize the most sensitive harmful samples. We then instantiate the plateau loss as:

$$\mathcal{L}_{\text{plateau}}(\boldsymbol{\theta}) \coloneqq \sum_{i=1}^{B} w_i\left(\ell_i(\boldsymbol{\theta}) - \ell_i(\boldsymbol{\theta}_{\text{pert}})\right). \tag{10}$$

Condition number enlargement. Complementary to the plateau objective, we distort the harmful loss geometry by increasing the condition number $\kappa(\boldsymbol{H}) = \sigma_{\max}/\sigma_{\min}$ of the harmful Hessian. However, computing the explicit Hessian $\boldsymbol{H}$ is intractable for LLMs. Inspired by Sophia [31], we use the Gauss-Newton approximation. Let $\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}) \in \mathbb{R}^V$ be the logits (pre-softmax scores) and $\ell(\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}), \boldsymbol{y})$ be the cross-entropy loss. The Hessian of the loss can be approximated by the Gauss-Newton matrix:

$$\nabla^2_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}) \approx \boldsymbol{J}_{\boldsymbol{\theta}}\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}) \cdot \boldsymbol{S} \cdot \boldsymbol{J}_{\boldsymbol{\theta}}\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x})^{\top}, \tag{11}$$

where $\boldsymbol{J}_{\boldsymbol{\theta}}\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}) \in \mathbb{R}^{d\times V}$ is the Jacobian of the logits with respect to parameters $\boldsymbol{\theta}$ and $\boldsymbol{S} \in \mathbb{R}^{V\times V}$ is the Hessian of the loss with respect to the logits. Here, we omit the residual term involving the second derivative of the mapping f as it is often negligible [31]. The Gauss-Newton term can be expressed as the covariance of gradients computed with labels $\hat{\boldsymbol{y}}$ sampled from the model's predictive distribution:

$$\begin{aligned}\boldsymbol{H}_{\mathrm{GN}} &\coloneqq \boldsymbol{J}_{\boldsymbol{\theta}}\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}) \cdot \boldsymbol{S} \cdot \boldsymbol{J}_{\boldsymbol{\theta}}\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x})^{\top} \\ &= \mathbb{E}_{\hat{\boldsymbol{y}}\sim p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})}[\nabla\ell(\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}), \hat{\boldsymbol{y}})\nabla\ell(\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}), \hat{\boldsymbol{y}})^{\top}].\end{aligned} \tag{12}$$

This identity allows us to represent curvature using a Gram matrix built from sub-minibatch gradients sampled from the model.

Theorem 4.1. *Consider a harmful minibatch of size B partitioned into K sub-minibatches of size M ($B = KM$). Let $\boldsymbol{g}_k$ be the average gradient of the k-th sub-minibatch computed with labels sampled from the model's predictive distribution. Define $\boldsymbol{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K] \in \mathbb{R}^{d\times K}$. The Gram matrix estimator of the minibatch Gauss-Newton curvature is given by:*

$$\widehat{\boldsymbol{H}}_{\mathrm{GN}} \coloneqq \frac{M^2}{B}\boldsymbol{G}\boldsymbol{G}^{\top}. \tag{13}$$

The proof of this theorem is provided in Appendix B. In our implementation, the sub-minibatch gradients are computed with the ground-truth harmful labels instead of sampled labels (see Appendix H for a discussion and a comparison with sampled labels).

With this estimator, each sub-minibatch gradient norm $\|\boldsymbol{g}_k\|_2^2$ becomes a cheap directional curvature proxy. To connect $\|\boldsymbol{g}_k\|_2^2$ to extremal curvature, note that $\widehat{\boldsymbol{H}}_{\mathrm{GN}} = \frac{M^2}{B}\boldsymbol{G}\boldsymbol{G}^{\top}$ is positive semidefinite and $\boldsymbol{G}\boldsymbol{G}^{\top}$ and $\boldsymbol{G}^{\top}\boldsymbol{G}$ share the same non-zero eigenvalues. For the k-th standard basis vector $\boldsymbol{e}_k \in \mathbb{R}^K$,

$$\boldsymbol{e}_k^{\top}(\boldsymbol{G}^{\top}\boldsymbol{G})\boldsymbol{e}_k \;=\; \|\boldsymbol{g}_k\|_2^2. \tag{14}$$

Hence, by Rayleigh quotient bounds [32],

$$\sigma_{\max}(\widehat{\boldsymbol{H}}_{\mathrm{GN}}) \;\geq\; \tfrac{M^2}{B}\max_k\|\boldsymbol{g}_k\|_2^2, \qquad \sigma_{\min}(\widehat{\boldsymbol{H}}_{\mathrm{GN}}) \;\leq\; \tfrac{M^2}{B}\min_k\|\boldsymbol{g}_k\|_2^2, \tag{15}$$

where $\sigma_{\min}$ denotes the smallest non-zero singular value. Dropping the constant factor $\frac{M^2}{B}$ yields simple computable proxies, and we maximize the condition number by simultaneously decreasing the smallest singular value proxy and increasing the largest:

$$\mathcal{L}_{\text{cn-min}}(\boldsymbol{\theta}) \coloneqq \min_k \|\boldsymbol{g}_k\|_2^2, \qquad \mathcal{L}_{\text{cn-max}}(\boldsymbol{\theta}) \coloneqq \max_k \|\boldsymbol{g}_k\|_2^2. \tag{16}$$

Minimizing $\mathcal{L}_{\text{cn-min}}$ suppresses the curvature in the direction of least variation by lowering the upper bound of $\sigma_{\min}$, while maximizing $\mathcal{L}_{\text{cn-max}}$ amplifies the curvature in the direction of greatest variation by raising the lower bound of $\sigma_{\max}$.

Optimizing these curvature proxies requires differentiating through the gradient norm $\|\boldsymbol{g}\|_2^2$, which involves the Hessian-vector product $\boldsymbol{Hg}$. To avoid instantiating the Hessian, we employ a finite-difference approximation. For a selected sub-minibatch gradient $\boldsymbol{g}$, we perturb the weights by ε to obtain $\boldsymbol{\theta}' = \boldsymbol{\theta} - \varepsilon \boldsymbol{g}$. The gradient of the squared norm is then approximated as:

$$\nabla_{\boldsymbol{\theta}} \|\boldsymbol{g}\|_2^2 = 2\boldsymbol{Hg} \approx \frac{2}{\varepsilon}\big(\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}')\big). \tag{17}$$

In practice, for each harmful minibatch, we compute the set $\{\boldsymbol{g}_k\}$ and identify the indices $k_{\min}$ and $k_{\max}$ corresponding to the smallest and largest norms. We then perform the finite-difference calculation only for these two selected directions. This ensures the computational overhead is limited to two additional backward passes per step to make the method scalable to large models.

Overall formulation. Let $\mathcal{D}_{\text{align}}$ denote the alignment dataset consisting of harmful requests paired with safe responses. We define $\mathcal{L}_{\text{align}}$ as the standard cross-entropy loss on $\mathcal{D}_{\text{align}}$. Putting everything together, the overall optimization objective of BECON is

$$\mathcal{L}_{\text{align}}(\boldsymbol{\theta}) \;+\; \lambda_1 \mathcal{L}_{\text{plateau}}(\boldsymbol{\theta}) \;+\; \lambda_2 \mathcal{L}_{\text{cn-min}}(\boldsymbol{\theta}) \;-\; \lambda_3 \mathcal{L}_{\text{cn-max}}(\boldsymbol{\theta}), \tag{18}$$

where the hyperparameters λ_1, λ_2, and λ_3 balance the contributions of the different terms. In our implementation, when computing $\mathcal{L}_{\text{plateau}}$ and the condition-number losses, we treat the perturbed parameters (e.g., $\boldsymbol{\theta}_{\text{pert}}$ and $\boldsymbol{\theta}' = \boldsymbol{\theta} - \varepsilon \boldsymbol{g}$) as stop-gradient quantities and do not backpropagate through the perturbation construction. We summarize the full BECON training procedure in Algorithm 1, provide further comparison with Booster [10] and Antibody [33] in Appendix F, and discuss the strength of our condition-number surrogate relative to prior work [8] in Appendix H.

5 Experiments

In this section, we demonstrate the effectiveness of our immunization approach in three settings. First, we study the FTaaS setup, where the service provider controls fine-tuning. Second, we consider the open-weight setting, where adversaries can fine-tune the model with different training dynamics. Finally, we show that our method provides robust unlearning against relearning attacks.

5.1 Defense against harmful fine-tuning in FTaaS

Experiment Setup & Threat Model. We evaluate our framework in the FTaaS setting, simulating a two-stage pipeline: *alignment* and *fine-tuning*. In the alignment stage, the provider ensures safety using a safety dataset $\mathcal{D}_{\text{align}}$ (refusals) and a harmful dataset $\mathcal{D}_{\text{harm}}$ (harmful Q&A). Subsequently, in the fine-tuning stage, a user submits a dataset $\mathcal{D}_{\text{task}}$ containing n samples. Following prior work [2, 10], we assume an attack scenario where the user injects harmful samples into this dataset, such that $\mathcal{D}_{\text{task}}$ consists of $(1-p)\%$ benign task data and $p\%$ harmful data. The defender's goal is to maintain safety without degrading performance on the benign task.

Datasets and Models. We utilize harmful data curated from the BeaverTails dataset [34]. For benign tasks, we employ SST2 [35], AGNEWS [36] and GSM8K [37]. We evaluate performance across three model families: Llama-2-7B [38], Qwen-2-7B [39] and Gemma-2-9B [40]. We compare our approach against standard Supervised Fine-Tuning (SFT) and other defense baselines: Vaccine [12], Booster [10], and Lisa [41].

Evaluation Metrics. Following the evaluation protocol of prior works [11], we assess models on two metrics: (i) Safety, measured by the Harmful Score (HS), which is the percentage of responses to malicious prompts flagged as harmful by a moderation model [34], and (ii) Fine-tuning performance, measured by Fine-tuning Accuracy (FA) on the benign tasks.

Implementation Details. We utilize LoRA [42] for efficient training. Unless otherwise stated, our default setting uses Llama-2-7B fine-tuned on the GSM8K dataset ($n = 1000$) with a harmful injection ratio of $p = 0.2$. Results are averaged over three random seeds. Full hyperparameters and training configurations are detailed in Appendix C.1.1.

Table 1: Performance of models trained on different fine-tuning datasets. The best and the second best are highlighted in blue and gray, respectively.

Methods	Llama-2-7B						Gemma2-9B						Qwen2-7B					
	SST2		AGNEWS		GSM8K		SST2		AGNEWS		GSM8K		SST2		AGNEWS		GSM8K	
	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
SFT	36.29	92.70	34.57	85.40	23.94	10.90	40.77	94.11	39.91	84.47	32.05	56.73	33.86	93.31	31.00	85.77	14.54	64.66
Vaccine	44.16	91.71	39.73	82.43	23.60	11.70	51.65	**94.50**	51.64	**85.27**	38.24	54.37	29.80	93.22	28.66	84.93	18.17	62.20
Lisa	22.94	92.51	20.93	84.50	5.86	9.23	34.67	94.04	33.33	83.80	13.02	54.97	18.46	92.85	15.26	85.07	3.91	63.90
Booster	14.31	92.59	15.88	86.70	9.06	**16.27**	28.47	94.15	19.22	84.30	22.13	**58.97**	3.96	94.19	3.48	85.13	**2.19**	**68.63**
BECON	**11.16**	**93.15**	**13.59**	**86.77**	**3.72**	15.43	**16.21**	94.42	**4.24**	84.80	**8.15**	58.07	**3.34**	**94.76**	**2.91**	**86.00**	2.48	67.93

Generalization to different fine-tuning datasets. As shown in Table 1, BECON provides the strongest overall defense across datasets and model families. It achieves the lowest HS in eight of nine settings, and ranks second on Qwen2-7B with GSM8K, where its HS is 2.48% compared to 2.19% for Booster. BECON also preserves downstream trainability, attaining the best or second-best FA in every setting. These results show that BECON reliably mitigates harmful fine-tuning while remaining competitive on task performance. Additional FTaaS experiments on the **more recent Qwen-3-4B model** [43], under **harmful data distribution shift**, and for the **local safety loss landscape** are reported in Appendix D.1, D.5, and G, respectively.

Table 2: Performance under different harmful ratios in the default setting. *clean* means no harmful samples or $p = 0$. The best and the second best are highlighted in blue and gray, respectively.

Methods	Harmful Score ↓							Fine-tuning Accuracy ↑						
($n = 1000$)	clean	$p = 0.05$	$p = 0.1$	$p = 0.15$	$p = 0.2$	$p = 0.25$	Average	clean	$p = 0.05$	$p = 0.1$	$p = 0.15$	$p = 0.2$	$p = 0.25$	Average
SFT	2.05	10.68	15.59	19.06	23.94	27.85	16.53	12.23	11.83	11.43	10.90	10.90	10.53	11.30
Vaccine	1.29	7.15	13.07	19.55	23.60	29.33	15.67	12.27	12.23	11.77	11.77	11.77	11.77	11.93
Lisa	**0.95**	3.05	3.82	4.62	5.86	8.16	4.41	9.97	9.57	9.60	8.90	9.23	9.20	9.41
Booster	1.38	**1.76**	2.91	5.10	9.06	13.49	5.62	**16.53**	15.97	**16.50**	**16.30**	**16.43**	**15.70**	**16.24**
BECON	1.86	2.00	**2.20**	**3.62**	**3.72**	**7.06**	**3.41**	15.97	**16.10**	15.13	15.20	15.43	15.57	15.57

Robustness across different harmful ratios. As shown in Table 2, BECON remains robust as the harmful ratio p increases. It achieves the best average HS of 3.41% and the lowest HS for all $p \geq 0.1$, while avoiding the utility loss of Lisa. Although Booster attains the highest average FA, its safety degrades more quickly under heavier poisoning, reaching 13.49% HS at $p = 0.25$ compared to 7.06% for BECON. Overall, BECON provides the strongest safety-utility trade-off across poisoning ratios.

5.2 Immunization against harmful fine-tuning attacks in open-weight settings

In this section, we study *open-weight immunization*, a scenario where an immunized checkpoint is released, and an adversary possesses full control over post-release fine-tuning. Concretely, we evaluate robustness against *harmful-only* fine-tuning using curated harmful instructions from BeaverTails [34]. To rigorously simulate this threat, we stress-test the model by sweeping across the attacker's learning rate and the number of fine-tuning epochs; these results are reported in Tables 3 and 4. We perform these evaluations on Llama-2-7B [38], where all fine-tuning is conducted using LoRA [42] with the AdamW optimizer. Further implementation details are provided in Appendix C.2, and computational overhead analysis is reported in Appendix E. We further evaluate **full-parameter fine-tuning attacks** in Appendix D.2.

Following the protocols established by [4] and [8], we assess performance across three dimensions, comparing our method against existing open-weight defenses such as TAR [6] and RepNoise [5]. Specifically, we evaluate: (i) harmful fine-tuning resistance, measured using the Harmful Score on the BeaverTails-Evaluation split; (ii) utility preservation, which assesses the retention of helpful behavior via accuracy on MMLU [44], TruthfulQA [45], HellaSwag [46], and ARC-Easy [47]; and (iii) trainability, which evaluates the model's ability to learn from harmless datasets via downstream fine-tuning accuracy on SST2 [35], AGNEWS [36], and GSM8K [37].

Table 3: Harmful immunization performance across training epochs. The best and the second best are highlighted in blue and gray, respectively.

Methods	Harmful Score ↓					
	Epoch=5	Epoch=10	Epoch=20	Epoch=50	Epoch=75	Epoch=100
TAR	42.63	50.64	54.51	55.08	55.79	54.94
RepNoise	13.02	16.31	**15.74**	**43.35**	53.51	55.79
BECON	**1.00**	**3.43**	27.47	50.21	**50.50**	**52.50**

Table 4: Harmful immunization performance across different learning rates. The best and the second best are highlighted in blue and gray, respectively.

Methods	Harmful Score ↓					
	lr=5e-6	lr=1e-5	lr=5e-5	lr=1e-4	lr=5e-4	lr=1e-3
TAR	28.47	42.63	56.08	55.79	54.08	54.94
RepNoise	10.16	13.02	**9.63**	54.51	**52.36**	56.08
BECON	**1.14**	**1.00**	34.33	**47.93**	54.79	**54.51**

Attack robustness against different learning rates and number of epochs. Across varied attack hyper-parameters in Tables 3 and 4, BECON shows a clear immunization effect against harmful-only fine-tuning. Harmful Score remains very low under small learning rates and limited epochs, and BECON continues to resist many stronger attacks with 75-100 epochs or learning rates of 10^{-4} to 10^{-3}, where TAR and RepNoise degrade more substantially. At the same time, the increase in HS under aggressive settings such as learning rate of 5×10^{-5} or epochs of 20 shows that open-weight immunization remains sensitive to the attack budget, and current defenses do not yet provide uniform protection across all fine-tuning configurations. Although RepNoise is stronger in some mid-range settings, BECON offers the most robust overall defense profile. This sensitivity is expected from Theorem 3.3, where the convergence gap shrinks with more steps and a larger learning rate. BECON therefore slows harmful fine-tuning rather than preventing it. TAR and RepNoise reach a similar harmful score under the strongest attacks, which suggests a limit shared by current open-weight defenses [29]. This highlights robust protection under larger attack budgets as an important direction for future work.

Table 5: Comparison of the utility and fine-tuning capabilities of the non-immunized model and model immunized with BECON.

Model	**Acc** (%)				**FA** (%)		
	MMLU	TruthfulQA	ARC	Hellaswag	SST2	AGNEWS	GSM8K
Non-immunized model	45.66	38.94	76.30	58.96	93.90	84.10	18.17
BECON	45.14	42.54	71.74	55.26	93.50	88.47	15.97

Effect of BECON on model utility and trainability. Because most users care about direct use and benign fine-tuning, we assess BECON's effects on general capabilities and trainability relative to the base model. Table 5 shows that BECON preserves useful capability in several settings, but with a clear trade-off between stronger resistance to harmful fine-tuning and overall performance. It improves TruthfulQA from 38.94% to 42.54% but shows visible drops on ARC and other benchmarks. For benign trainability, the immunized model remains learnable on harmless data, matching SST2 and improving AGNEWS from 84.10% to 88.47%, though lower GSM8K results show trainability is not fully preserved across tasks. Overall, BECON largely maintains benign usefulness while substantially improving immunization, though remaining utility regressions leave room for improvement.

5.3 Robust Unlearning against Relearning Attack

In this section, we evaluate the proposed immunization framework for robust unlearning against relearning attacks on Zephyr-7B-beta [48], focusing on the biosecurity domain of the Weapons of

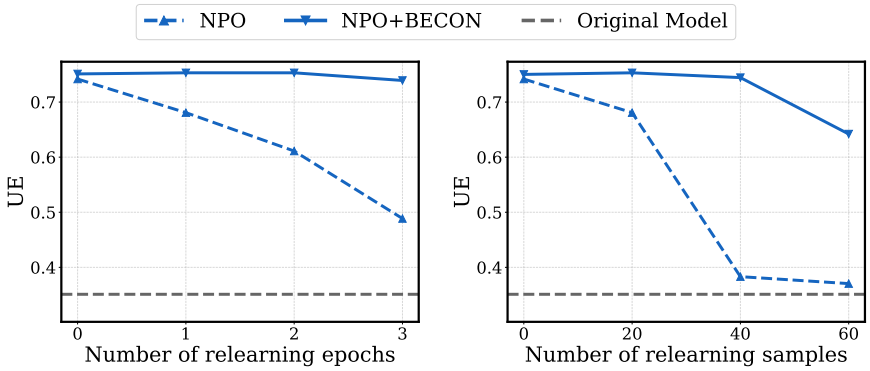


Figure 1: Robustness evaluation of unlearning methods on WMDP against relearning attacks. Left: UE vs. relearning epochs. Right: UE vs. number of relearning samples.

Mass Destruction Proxies benchmark (WMDP) [14]. We compare Negative Preference Optimization (NPO) [49] with NPO+BECON, which additionally includes the $\mathcal{L}_{\text{plateau}}$ and $\mathcal{L}_{\text{cn-min}} - \mathcal{L}_{\text{cn-max}}$ terms from Section 4. NPO steers the model away from producing preferred outputs on the forget set by optimizing a loss that assigns negative preference to those targets, thereby discouraging retention of the specified knowledge. Concretely, we use WMDP-Bio as the forget dataset for the unlearning loss and a retain set of unrelated general content from Wikitext [50], then simulate adversarial relearning by fine-tuning on samples from the WMDP-Bio forget set to test resistance to knowledge recovery. The unlearning robustness performance is quantified using Unlearning Effectiveness (UE) (higher is better), defined as: $\text{UE} = 1 - \texttt{Accuracy on WMDP-Bio evaluation set}$. Further experiment details are provided in Appendix C.3.

Unlearning robustness against relearning attack. Figure 1 shows that the baseline NPO is highly vulnerable to knowledge recovery. Its UE degrades significantly from 0.74 to 0.49 after 3 epochs of relearning. Similarly, increasing the number of relearning samples to 60 causes NPO's UE to decrease to 0.37. In contrast, NPO+BECON demonstrates remarkable resilience. Under the same conditions, it maintains a high UE of 0.74 after 3 epochs, showing negligible degradation compared to its initial performance. Even with 60 relearning samples, NPO+BECON retains a substantially higher UE of 0.64 compared to NPO. These results confirm that our proposed immunization framework effectively prevents the knowledge recovery observed in standard unlearning methods.

6 Ablation Study

Table 6: Ablation on our proposed components. Each row is a cumulative addition to the previous one. We color-code the performance change from the SFT baseline.

Method	GSM8K HS ↓	GSM8K FA ↑
SFT	23.94	10.90
SFT + $\mathcal{L}_{\text{cn-min}} - \mathcal{L}_{\text{cn-max}}$	20.13 (−3.81)	15.00 (+4.10)
SFT + $\mathcal{L}_{\text{plateau}}$	10.40 (−13.54)	15.07 (+4.17)
BECON	3.72 (−20.22)	15.43 (+4.53)

Ablation of Our Components. Using the experiment setup in Section 5.1, we ablate the individual and combined effects of condition number enlargement and harmful plateau promotion on GSM8K. As shown in Table 6, condition number enlargement ($\mathcal{L}_{\text{cn-min}} - \mathcal{L}_{\text{cn-max}}$) alone reduces HS to 20.13. The harmful plateau promotion ($\mathcal{L}_{\text{plateau}}$) is more aggressive, lowering HS to 10.40 and supporting our theory that condition number manipulation should be complemented by gradient suppression. Combining both components in BECON achieves the best result, reducing HS to 3.72 while maintaining a high FA of 15.43, showing that their synergy is crucial for robust immunization. Further discussion on the contribution and effectiveness of the condition-number regularization term is provided in Appendix H.

7 Conclusion

In this paper, we introduce an immunization framework designed to protect LLMs from safety compromise during harmful fine-tuning. By refining the theoretical framework of condition-number-based immunization to prioritize plateau regions and leveraging an efficient estimator for the Hessian, we provide an effective algorithm for LLM immunization. Our evaluations across deployment scenarios, including FTaaS, open-weight settings, and robust unlearning, demonstrate that our method effectively inhibits the optimization of harmful objectives while largely maintaining general model utility and benign trainability. As the demand for open-weight models grows, this efficient approach offers a vital step toward safeguarding safety alignment against adversarial adaptation in real-world environments.

8 Limitations and Future Work

Our method improves the resilience of open-weight models against harmful fine-tuning, but it does not provide a universal guarantee under all harmful distributions or attack settings. In particular, its protection may weaken when harmful behaviors are substantially different from those represented in the proxy data used during immunization, and stronger or more aggressive fine-tuning can still reduce safety alignment. Moreover, stronger immunization may introduce trade-offs with general utility on some benchmarks. Immunization also adds a one-time training cost, and our analysis assumes deterministic full-batch gradient descent, so extending the bounds to stochastic gradient descent is left to future work. We therefore view this work as a meaningful step toward making harmful adaptation more difficult in practice, while recognizing that an important direction for future work is to better understand and improve robustness under broader, more out-of-distribution, and more resource-intensive attacks while reducing the associated utility trade-offs.

Acknowledgments and Disclosure of Funding

Trung Le and Mehrtash Harandi were supported by the ARC Discovery Project grant DP250100262. The authors declare no competing interests.

References

[1] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023.

[2] 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! In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=hTEGyKf0dZ`.

[3] Xiangyu Qi, Boyi Wei, Nicholas Carlini, Yangsibo Huang, Tinghao Xie, Luxi He, Matthew Jagielski, Milad Nasr, Prateek Mittal, and Peter Henderson. On evaluating the durability of safeguards for open-weight LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=fXJCqdUSVG`.

[4] Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Hassan Sajjad, and Frank Rudzicz. Immunization against harmful fine-tuning attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5234–5247, 2024.

[5] 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.

[6] Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs. In *The Thirteenth*

International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=4FIjRodbW6.

[7] Domenic Rosati, Sebastian Dionicio, Xijie Zeng, Subhabrata Majumdar, Frank Rudzicz, and Hassan Sajjad. Locking open weight models with spectral deformation. In *ICML Workshop on Technical AI Governance (TAIG)*, 2025. URL https://openreview.net/forum?id=cjrm7bo6Eg.

[8] Amber Yijia Zheng, Site Bai, Brian Bullins, and Raymond A. Yeh. Model immunization from a condition number perspective. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=uitj69FqD5.

[9] Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. On the vulnerability of safety alignment in open-access LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9236–9260, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.549. URL https://aclanthology.org/2024.findings-acl.549/.

[10] 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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tTPHgb0EtV.

[11] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful fine-tuning attacks and defenses for large language models: A survey. *arXiv preprint arXiv:2409.18169*, 2024.

[12] Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:74058–74088, 2024.

[13] Guozhi Liu, Weiwei Lin, Qi Mu, Tiansheng Huang, Ruichao Mo, Yuren Tao, and Li Shen. Targeted vaccine: Safety alignment for large language models against harmful fine-tuning via layer-wise perturbation. *IEEE Transactions on Information Forensics and Security*, 2025.

[14] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xlr6AUDuJz.

[15] Biao Yi, Tiansheng Huang, Baolei Zhang, Tong Li, Lihai Nie, Zheli Liu, and Li Shen. Ctrap: Embedding collapse trap to safeguard large language models from harmful fine-tuning. *arXiv preprint arXiv:2505.16559*, 2025.

[16] Liang CHEN, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. Vulnerability-aware alignment: Mitigating uneven forgetting in harmful fine-tuning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EMHED4WTHT.

[17] Gabriel Jacob Perin, Runjin Chen, Xuxi Chen, Nina S. T. Hirata, Zhangyang Wang, and Junyuan Hong. Lox: Low-rank extrapolation robustifies LLM safety against fine-tuning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ASS5YD4hL4.

[18] Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

[19] Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned LLMs via benign relearning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fMNRYBvcQN.

[20] Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of LLM unlearning via quantization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lHSeDYamnz.

[21] Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. Ununlearning: Unlearning is not sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*, 2024.

[22] Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O'Gara, Robert Kirk, Ben Bucknall, Tim Fist, et al. Open problems in machine unlearning for ai safety. *arXiv preprint arXiv:2501.04952*, 2025.

[23] Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.

[24] Zora Che, Stephen Casper, Anirudh Satheesh, Rohit Gandikota, Domenic Rosati, Stewart Slocum, Lev E McKinney, Zichu Wu, Zikui Cai, Bilal Chughtai, Furong Huang, and Dylan Hadfield-Menell. Model manipulation attacks enable more rigorous evaluations of LLM unlearning. In *Neurips Safe Generative AI Workshop 2024*, 2024. URL https://openreview.net/forum?id=XmvgWEjkhG.

[25] Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards LLM unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=zZjLv6F0Ks.

[26] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

[27] Peter Balazs, Daniel Haider, Vincent Lostanlen, and Felix Perfler. Trainable signal encoders that are robust against noise. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 270, pages 1836–1844. Institute of Noise Control Engineering, 2024.

[28] Rossen Nenov, Daniel Haider, and Peter Balazs. (almost) smooth sailing: Towards numerical stability of neural networks through differentiable regularization of the condition number. In *ICML 2024 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, 2024. URL https://openreview.net/forum?id=OJkyvJXk2B.

[29] Domenic Rosati, Xijie Zeng, Hong Huang, Sebastian Dionicio, Subhabrata Majumdar, Frank Rudzicz, and Hassan Sajjad. Limits of convergence-rate control for open-weight safety. *arXiv preprint arXiv:2602.18868*, 2026.

[30] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 2015.

[31] Hong Liu, Zhiyuan Li, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3xHDeA8Noi.

[32] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[33] Quoc Minh Nguyen, Trung Le, Jing Wu, Anh Tuan Bui, and Mehrtash Harandi. Antibody: Strengthening defense against harmful fine-tuning for large language models via attenuating harmful gradient influence. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qur2ef8MqQ.

[34] 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.

[35] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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, 2013.

[36] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

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

[38] 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.

[39] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[40] 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.

[41] 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.

[42] 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.

[43] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[44] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

[45] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252, 2022.

[46] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

[47] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[48] Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aKkAwZB6JV.

[49] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb.

[50] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

[51] Huaqin Zhao, Jiaxi Li, Yi Pan, Shizhe Liang, Xiaofeng Yang, Fei Dou, Tianming Liu, and Jin Lu. HELENE: Hessian layer-wise clipping and gradient annealing for accelerating fine-tuning LLM with zeroth-order optimization. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26044–26067, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1323. URL https://aclanthology.org/2025.emnlp-main.1323/.

[52] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[53] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[54] Thanh-Thien Le, Viet Dao, Linh Nguyen, Thi-Nhung Nguyen, Linh Ngo, and Thien Nguyen. SharpSeq: Empowering continual event detection through sharpness-aware sequential-task learning. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3632–3644, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.200. URL https://aclanthology.org/2024.naacl-long.200/.

[55] Hoang Anh Dung, Cuong Pham, Trung Le, Jianfei Cai, and Thanh-Toan Do. Sharpness-aware data generation for zero-shot quantization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=8mKXMnhnFW.

[56] Haocheng Luo, Mehrtash Harandi, Dinh Phung, and Trung Le. Unveiling m-sharpness through the structure of stochastic gradient noise. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=rMptAK0Xm8.

[57] Sheng Y Peng, Pin-Yu Chen, Matthew Hull, and Duen H Chau. Navigating the safety landscape: Measuring risks in finetuning large language models. *Advances in Neural Information Processing Systems*, 3:95692–95715, 2024.

[58] Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18098–18109. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d1ff1ec86b62cd5f3903ff19c3a326b2-Paper.pdf.

[59] Xiaodong Wu, Wenyi Yu, Chao Zhang, and Philip Woodland. An improved empirical fisher approximation for natural gradient descent. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 134151–134194. Curran Associates, Inc., 2024. doi: 10.52202/079017-4264. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f23098fa0cfcdef0e743b134d380eeb9-Paper-Conference.pdf.

A Algorithm

Algorithm 1 BECON

Input: parameters $\boldsymbol{\theta}$, plateau radius ρ, finite-difference strength ε, number of sub-minibatches K, weights $\lambda_1, \lambda_2, \lambda_3$, optimizer step size γ
for each training minibatch **do**
  Compute alignment gradient $\boldsymbol{g}_{\text{align}} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{align}}(\boldsymbol{\theta})$.
  Compute harmful minibatch gradient $\boldsymbol{g}_{\text{H}} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{H}}(\boldsymbol{\theta})$.
  Plateau perturbation direction (stop-grad):
    $\boldsymbol{v} \leftarrow \boldsymbol{g}_{\text{H}} / \|\boldsymbol{g}_{\text{H}}\|_2$ (treat $\boldsymbol{v}$ as stop-gradient)
    $\boldsymbol{\theta}_{\text{pert}} \leftarrow \boldsymbol{\theta} - \rho\, \boldsymbol{v}$
  Plateau gradient estimate:
    $\nabla \mathcal{L}_{\text{plateau}} \leftarrow \boldsymbol{g}_{\text{H}} - \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{H}}(\boldsymbol{\theta}_{\text{pert}})$
  Split the harmful minibatch into K sub-minibatches, each containing M samples.
  for $k = 1, \ldots, K$ **do**
    Compute sub-minibatch gradient $\boldsymbol{g}_k \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_k(\boldsymbol{\theta})$ (*e.g.*, with labels sampled from $p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})$)
    Compute squared norm $s_k \leftarrow \|\boldsymbol{g}_k\|_2^2$
  end for
  Select indices $k_{\max} \leftarrow \arg\max_k s_k$ and $k_{\min} \leftarrow \arg\min_k s_k$
  Define proxies: $\mathcal{L}_{\text{cn-max}}(\boldsymbol{\theta}) \leftarrow s_{k_{\max}}$, $\mathcal{L}_{\text{cn-min}}(\boldsymbol{\theta}) \leftarrow s_{k_{\min}}$
  Perturb parameters for finite differences:
    $\boldsymbol{\theta}'_{\max} \leftarrow \boldsymbol{\theta} - \varepsilon \boldsymbol{g}_{k_{\max}}$, $\boldsymbol{\theta}'_{\min} \leftarrow \boldsymbol{\theta} - \varepsilon \boldsymbol{g}_{k_{\min}}$ (treat $\boldsymbol{g}_{k_{\max}}, \boldsymbol{g}_{k_{\min}}$ as stop-gradient)
  Compute gradient differences:
    $d_{\max} \leftarrow \big(\nabla_{\boldsymbol{\theta}} \mathcal{L}_{k_{\max}}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_{k_{\max}}(\boldsymbol{\theta}'_{\max})\big)/\varepsilon$
    $d_{\min} \leftarrow \big(\nabla_{\boldsymbol{\theta}} \mathcal{L}_{k_{\min}}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_{k_{\min}}(\boldsymbol{\theta}'_{\min})\big)/\varepsilon$
  Estimate gradients of the proxies:
    $\nabla \mathcal{L}_{\text{cn-max}} \leftarrow 2 d_{\max}$, $\nabla \mathcal{L}_{\text{cn-min}} \leftarrow 2 d_{\min}$
  Direct update direction:
    $u \leftarrow \boldsymbol{g}_{\text{align}} + \lambda_1 \nabla \mathcal{L}_{\text{plateau}} + \lambda_2 \nabla \mathcal{L}_{\text{cn-min}} - \lambda_3 \nabla \mathcal{L}_{\text{cn-max}}$
  Update parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma\, u$
end for

For the pseudo-label used in Algorithm 1, our derivation follows Sophia [31], where sampling labels from the model predictive distribution gives a theoretical interpretation of the Gauss-Newton term and shows that predictive-distribution gradients provide mathematically grounded probes of local curvature. In implementation, however, we use the ground-truth label as a practical deterministic approximation, rather than sampling a label sequence at each token, since this substantially simplifies computation while maintaining strong empirical performance. This deterministic choice is also adopted in other Sophia-inspired work [51].

Regularity assumptions. The objective in (18) assumes that the harmful loss is twice continuously differentiable near the perturbed parameters, with a bounded and locally Lipschitz Hessian so that the Taylor and finite-difference approximations are accurate. The minimum and maximum over the K sub-minibatches are differentiable when the selected index is unique.

B All Proofs

B.1 Proof of Lemma 3.1

Lemma 3.1.

(i) The PL-type inequality holds for $\sigma_{\min}$:

$$\frac{1}{2}\|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2 \geq \sigma_{\min}\left(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*\right), \quad \forall\, \boldsymbol{\theta}, \tag{19}$$

where $\mathcal{L}^* = \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$.

(ii) The opposite PL-type inequality holds for $\sigma_{\max}$:

$$\frac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \leq \sigma_{\max}\left(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*\right), \quad \forall\, \boldsymbol{\theta}. \tag{20}$$

Proof of Lemma 3.1

(i) Using the strongly convex w.r.t. $\sigma_{\min}$, we have

$$\mathcal{L}(\boldsymbol{\theta}^*) \geq \mathcal{L}(\boldsymbol{\theta}) + \langle \nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^* - \boldsymbol{\theta}\rangle + \frac{1}{2}\sigma_{\min}\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|^2.$$

Rearranging the terms, we reach

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^* &\leq \langle \nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta} - \boldsymbol{\theta}^*\rangle - \frac{1}{2}\sigma_{\min}\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|^2 \\ &\leq \|\nabla\mathcal{L}(\boldsymbol{\theta})\|\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\| - \frac{1}{2}\sigma_{\min}\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|^2. \end{aligned} \tag{21}$$

Denote $r = \|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|$, we investigate the quadratic function

$$f(r) = \|\nabla\mathcal{L}(\boldsymbol{\theta})\| r - \frac{1}{2}\sigma_{\min} r^2,$$

which is maximized at

$$r_{\max} = \frac{\|\nabla\mathcal{L}(\boldsymbol{\theta})\|}{\sigma_{\min}}.$$

Substituting $r_{\max}$ back into $f(r)$ yields the maximum value

$$f_{\max} = \frac{1}{2}\frac{\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2}{\sigma_{\min}}.$$

Linking to (21), we reach the conclusion:

$$\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^* \leq \frac{1}{2}\frac{\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2}{\sigma_{\min}}.$$

(ii) Using the strongly smooth w.r.t. $\sigma_{\max}$, we obtain for any $\boldsymbol{\theta}'$:

$$\mathcal{L}\left(\boldsymbol{\theta}^{'}\right) \leq \mathcal{L}(\boldsymbol{\theta}) + \left\langle \nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^{'} - \boldsymbol{\theta}\right\rangle + \frac{1}{2}\sigma_{\max}\|\boldsymbol{\theta}^{'} - \boldsymbol{\theta}\|^2.$$

Choose $\boldsymbol{\theta}^{'} = \boldsymbol{\theta} - \frac{1}{\sigma_{\max}}\nabla\mathcal{L}(\boldsymbol{\theta})$ and note that $\mathcal{L}^* \leq \mathcal{L}\left(\boldsymbol{\theta}^{'}\right)$, we gain

$$\begin{aligned}\mathcal{L}^* &\leq \mathcal{L}(\boldsymbol{\theta}) + \left\langle \nabla\mathcal{L}(\boldsymbol{\theta}), -\frac{1}{\sigma_{\max}}\nabla\mathcal{L}(\boldsymbol{\theta})\right\rangle + \frac{1}{2}\sigma_{\max}\| - \frac{1}{\sigma_{\max}}\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \\ &\leq \mathcal{L}(\boldsymbol{\theta}) - \frac{1}{2\sigma_{\max}}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2,\end{aligned}$$

which is equivalent to

$$\frac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \leq \sigma_{\max}\left(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*\right).$$

B.2 Proof of Lemma 3.2

Lemma 3.2. The update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_t)$ is efficient in reducing the new objective $\mathcal{L}(\boldsymbol{\theta}_{t+1})$ (i.e., $\mathcal{L}(\boldsymbol{\theta}_{t+1}) < \mathcal{L}(\boldsymbol{\theta}_t)$) if $0 < \eta < \frac{2}{\sigma_{\max}}$.

Proof of Lemma 3.2. Let $h = \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|$. By the second-order Taylor expansion:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_{t+1}) &= \mathcal{L}(\boldsymbol{\theta}_t) + \langle \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t, \nabla\mathcal{L}(\boldsymbol{\theta}_t)\rangle + \frac{1}{2}\left(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\right)^T \boldsymbol{H}\left(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\right) + o(h^2) \\ &= \mathcal{L}(\boldsymbol{\theta}_t) + \langle -\eta\nabla\mathcal{L}(\boldsymbol{\theta}_t), \nabla\mathcal{L}(\boldsymbol{\theta}_t)\rangle + \frac{\eta^2}{2}\nabla\mathcal{L}(\boldsymbol{\theta}_t)^T \boldsymbol{H}\nabla\mathcal{L}(\boldsymbol{\theta}_t) + o(h^2) \\ &\leq \mathcal{L}(\boldsymbol{\theta}_t) - \eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\sigma_{\max}\eta^2}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \epsilon h^2 \\ &= \mathcal{L}(\boldsymbol{\theta}_t) - \eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\sigma_{\max}\eta^2}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \epsilon\eta^2\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \\ &= \mathcal{L}(\boldsymbol{\theta}_t) - \left(1 - \frac{\eta}{2}\sigma_{\max} - \epsilon\eta\right)\eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2,\end{aligned} \tag{22}$$

where the small-o notation is defined as $\lim_{h\to 0} o(h^2)/h^2 = 0$. Therefore, in the third line of (22), for every $\epsilon > 0$, there exists δ such that whenever $0 < h < \delta$, we have $o(h^2) \leq \epsilon h^2$.

This leads to the conclusion.

B.3 Proof of Theorem 3.3

Theorem 3.3. Considering the learning rate $\eta \in \left(0, \frac{2}{\sigma_{\max}}\right)$, we have the following inequalities:

(i) We can upper-bound $\mathcal{L}(\boldsymbol{\theta}_{t+1}) - \mathcal{L}^*$ by

$$\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right) - \mathcal{L}^* \leq \left(1 - \frac{1}{\kappa}\right)^{t+1} \left(\mathcal{L}\left(\boldsymbol{\theta}_0\right) - \mathcal{L}^*\right) + \frac{\left(\sigma_{\max}\eta - 1\right)^2}{2\sigma_{\max}} \sum_{i=1}^{t} \left(1 - \frac{1}{\kappa}\right)^{t-i} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_i\right)\|^2. \tag{23}$$

(ii) We can lower-bound $\mathcal{L}(\boldsymbol{\theta}_{t+1}) - \mathcal{L}^*$ by

$\square$ If $\kappa > 2$ and $\eta \in \left[\frac{\sigma_{\min}}{\sigma_{\max}^2}, \frac{2}{\sigma_{\max}}\right)$, we have the lower bound

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right) - \mathcal{L}^* \geq & \left(1 - \frac{1}{\kappa}\right)^{3(t+1)} \left(\mathcal{L}\left(\boldsymbol{\theta}_0\right) - \mathcal{L}^*\right) - \\ & \left(\eta - \frac{\sigma_{\min}}{\sigma_{\max}^2}\right) \left(1 - \frac{1}{\kappa^2}\right) \sum_{i=1}^{t} \left(1 - \frac{1}{\kappa}\right)^{3(t-i)} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_i\right)\|^2. \end{aligned} \tag{24}$$

$\square$ If the learning rate $0 < \eta < \frac{\sigma_{\min}}{\sigma_{\max}^2}$, we have the lower bound

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right) - \mathcal{L}^* \geq & \left(1 - \frac{1}{\kappa}\right)^{t+1} \left(\mathcal{L}\left(\boldsymbol{\theta}_0\right) - \mathcal{L}^*\right) - \\ & \frac{\eta}{2}\left(1 - \frac{\sigma_{\min}\eta}{2}\right) \sum_{i=1}^{t} \left(1 - \frac{1}{\kappa}\right)^{t-i} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_i\right)\|^2. \end{aligned} \tag{25}$$

Proof of Theorem 3.3.

(i) Using the second-order Taylor expansion, we have

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right) &\leq \mathcal{L}\left(\boldsymbol{\theta}_t\right) - \left(1 - \frac{\sigma_{\max}\eta}{2}\right)\eta \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ &= \mathcal{L}\left(\boldsymbol{\theta}_t\right) - \frac{1}{2\sigma_{\max}} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 + \left(\frac{1}{2\sigma_{\max}} - \eta + \frac{\sigma_{\max}}{2}\eta^2\right) \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ &= \mathcal{L}\left(\boldsymbol{\theta}_t\right) - \frac{1}{2\sigma_{\max}} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 + \frac{\left(\sigma_{\max}\eta - 1\right)^2}{2\sigma_{\max}} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2. \end{aligned}$$

Deducting $\mathcal{L}^*$ from two sides and using PL-typed inequality, we obtain

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right) - \mathcal{L}^* &\leq \mathcal{L}\left(\boldsymbol{\theta}_t\right) - \mathcal{L}^* - \frac{1}{2\sigma_{\max}} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 + \frac{\left(\sigma_{\max}\eta - 1\right)^2}{2\sigma_{\max}} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ &\leq \mathcal{L}\left(\boldsymbol{\theta}_t\right) - \mathcal{L}^* - \frac{\sigma_{\min}}{\sigma_{\max}} \left(\mathcal{L}\left(\boldsymbol{\theta}_t\right) - \mathcal{L}^*\right) + \frac{\left(\sigma_{\max}\eta - 1\right)^2}{2\sigma_{\max}} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ &\leq \left(1 - \frac{\sigma_{\min}}{\sigma_{\max}}\right) \left(\mathcal{L}\left(\boldsymbol{\theta}_t\right) - \mathcal{L}^*\right) + \frac{\left(\sigma_{\max}\eta - 1\right)^2}{2\sigma_{\max}} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2. \end{aligned} \tag{26}$$

Applying inequality (26) recursively, we reach

$$\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right) - \mathcal{L}^* \leq \left(1 - \frac{\sigma_{\min}}{\sigma_{\max}}\right)^{t+1} \left(\mathcal{L}\left(\boldsymbol{\theta}_0\right) - \mathcal{L}^*\right) + \sum_{i=1}^{t} \left(1 - \frac{\sigma_{\min}}{\sigma_{\max}}\right)^{t-i} \frac{\left(\sigma_{\max}\eta - 1\right)^2}{2\sigma_{\max}} \|\nabla \mathcal{L}\left(\boldsymbol{\theta}_i\right)\|^2.$$

(ii) Using the second-order Taylor expansion, we have

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right) &\approx \mathcal{L}\left(\boldsymbol{\theta}_t\right)+\left\langle\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t, \nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\right\rangle+\frac{1}{2}\left(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\right)^T \boldsymbol{H}\left(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\right) \\ &\geq \mathcal{L}\left(\boldsymbol{\theta}_t\right)-\eta\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2+\frac{1}{2} \sigma_{\min }\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2 \\ &=\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\eta\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2+\frac{1}{2} \sigma_{\min } \eta^2\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ &=\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\left(1-\frac{\sigma_{\min } \eta}{2}\right) \eta\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 . \end{aligned} \tag{27}$$

Deducting $\mathcal{L}^*$ to two sides of (27), we gain

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right)-\mathcal{L}^* &\geq \mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*-\left(1-\frac{\sigma_{\min } \eta}{2}\right) \eta\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ &=\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*-h(\eta)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2,\end{aligned}$$

where $h(\eta)=\eta-\frac{\sigma_{\min }}{2} \eta^2$.

We consider two following cases:

* **Case 1**: $\eta \in\left[\frac{\sigma_{\min }}{\sigma_{\max }^2}, \frac{2}{\sigma_{\max }}\right)$

We have

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right)-\mathcal{L}^* \geq & \mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*-h\left(\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2+\left(h\left(\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)-h(\eta)\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ =&\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*-h\left(\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2+\left(\frac{\sigma_{\min }}{\sigma_{\max }^2}-\eta\right)\left[1-\frac{\sigma_{\min }}{2}\left(\frac{\sigma_{\min }}{\sigma_{\max }^2}+\eta\right)\right] \\ \geq&\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*-h\left(\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2-\left(\eta-\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\left(1-\left(\frac{\sigma_{\min }}{\sigma_{\max }}\right)^2\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 .\end{aligned}$$

By noting that

$$\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \leq 2 \sigma_{\max }\left(\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*\right),$$

we gain

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right)-\mathcal{L}^* \geq & \mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*-2 \sigma_{\max } h\left(\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\left(\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*\right) \\ &-\left(\eta-\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\left(1-\left(\frac{\sigma_{\min }}{\sigma_{\max }}\right)^2\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ =&\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*-\left(2 \frac{\sigma_{\min }}{\sigma_{\max }}-\frac{\sigma_{\min }^3}{\sigma_{\max }^3}\right)\left(\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*\right) \\ &-\left(\eta-\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\left(1-\left(\frac{\sigma_{\min }}{\sigma_{\max }}\right)^2\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ \geq&\left(1-2 \frac{\sigma_{\min }}{\sigma_{\max }}+\frac{\sigma_{\min }^3}{\sigma_{\max }^3}\right)\left(\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*\right)-\left(\eta-\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\left(1-\left(\frac{\sigma_{\min }}{\sigma_{\max }}\right)^2\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 \\ \overset{(1)}{\geq}&\left(1-\frac{\sigma_{\min }}{\sigma_{\max }}\right)^3\left(\mathcal{L}\left(\boldsymbol{\theta}_t\right)-\mathcal{L}^*\right)-\left(\eta-\frac{\sigma_{\min }}{\sigma_{\max }^2}\right)\left(1-\left(\frac{\sigma_{\min }}{\sigma_{\max }}\right)^2\right)\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_t\right)\|^2 .\end{aligned} \tag{28}$$

The condition $\kappa>2$, equivalently $\alpha=\sigma_{\min } / \sigma_{\max }<1 / 2$, is used here as a technical assumption in the analysis to show that **a large condition number alone is insufficient, and that an additional plateau loss term is needed for effective immunization.** We note that we have (1) because if $\alpha=\frac{\sigma_{\min }}{\sigma_{\max }}<1 / 2$ then

$$\begin{aligned}1-2 \frac{\sigma_{\min }}{\sigma_{\max }}+\frac{\sigma_{\min }^3}{\sigma_{\max }^3} &=1-2 \alpha+\alpha^3=(1-\alpha)^3+\alpha-3 \alpha^2+2 \alpha^3 \\ &=(1-\alpha)^3+\alpha(1-\alpha)(1-2 \alpha)>(1-\alpha)^3 .\end{aligned}$$

Applying (28) recursively, we obtain

$$\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right)-\mathcal{L}^{*} \geq\left(1-\frac{\sigma_{\min }}{\sigma_{\max }}\right)^{3(t+1)}\left(\mathcal{L}\left(\boldsymbol{\theta}_{0}\right)-\mathcal{L}^{*}\right)- \\ \left(\eta-\frac{\sigma_{\min }}{\sigma_{\max }^{2}}\right)\left(1-\left(\frac{\sigma_{\min }}{\sigma_{\max }}\right)^{2}\right) \sum_{i=1}^{t}\left(1-\frac{\sigma_{\min }}{\sigma_{\max }}\right)^{3(t-i)}\left\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_{i}\right)\right\|^{2}.$$

* **Case 2**: $\eta \in\left(0, \frac{\sigma_{\min }}{\sigma_{\max }^{2}}\right]$

We derive as

$$\begin{aligned}
\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right)-\mathcal{L}^{*} & \overset{(1)}{\geq} \mathcal{L}\left(\boldsymbol{\theta}_{t}\right)-\mathcal{L}^{*}-\frac{1}{2} h\left(\frac{\sigma_{\min }}{\sigma_{\max }^{2}}\right)\left\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_{t}\right)\right\|^{2}-\frac{1}{2} h(\eta)\left\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_{t}\right)\right\|^{2} \\
& \geq \mathcal{L}\left(\boldsymbol{\theta}_{t}\right)-\mathcal{L}^{*}-\sigma_{\max } h\left(\frac{\sigma_{\min }}{\sigma_{\max }^{2}}\right)\left(\mathcal{L}\left(\boldsymbol{\theta}_{t}\right)-\mathcal{L}^{*}\right)-\frac{\eta}{2}\left(1-\frac{\sigma_{\min } \eta}{2}\right)\left\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_{t}\right)\right\|^{2} \\
& \geq\left(1-\frac{\sigma_{\min }}{\sigma_{\max }}+\frac{\sigma_{\min }^{3}}{2 \sigma_{\max }^{3}}\right)\left(\mathcal{L}\left(\boldsymbol{\theta}_{t}\right)-\mathcal{L}^{*}\right)-\frac{\eta}{2}\left(1-\frac{\sigma_{\min } \eta}{2}\right)\left\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_{t}\right)\right\|^{2} \\
& \geq\left(1-\frac{\sigma_{\min }}{\sigma_{\max }}\right)\left(\mathcal{L}\left(\boldsymbol{\theta}_{t}\right)-\mathcal{L}^{*}\right)-\frac{\eta}{2}\left(1-\frac{\sigma_{\min } \eta}{2}\right)\left\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_{t}\right)\right\|^{2}. \qquad (29)
\end{aligned}$$

Applying (29) recursively, we gain

$$\mathcal{L}\left(\boldsymbol{\theta}_{t+1}\right)-\mathcal{L}^{*} \geq\left(1-\frac{\sigma_{\min }}{\sigma_{\max }}\right)^{t+1}\left(\mathcal{L}\left(\boldsymbol{\theta}_{0}\right)-\mathcal{L}^{*}\right)- \\ \frac{\eta}{2}\left(1-\frac{\sigma_{\min } \eta}{2}\right) \sum_{i=1}^{t}\left(1-\frac{\sigma_{\min }}{\sigma_{\max }}\right)^{t-i}\left\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_{i}\right)\right\|^{2}.$$

B.4 Proof of Theorem 4.1.

Theorem 4.1. Consider a harmful minibatch of size B partitioned into K sub-minibatches of size M ($B = KM$). Let $\boldsymbol{g}_k$ be the average gradient of the k-th sub-minibatch computed with labels sampled from the model's predictive distribution. Define $\boldsymbol{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K] \in \mathbb{R}^{d\times K}$. The Gram matrix estimator of the minibatch Gauss-Newton curvature is given by:

$$\widehat{\boldsymbol{H}}_{\mathrm{GN}} \coloneqq \frac{M^2}{B}\boldsymbol{G}\boldsymbol{G}^{\top}. \qquad (30)$$

Proof of Theorem 4.1. Let $\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}) \in \mathbb{R}^V$ be the logits and $p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}) \coloneqq \mathrm{softmax}(\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}))$. For cross-entropy loss $\ell(\cdot, \cdot)$, a key identity (see [31]) expresses the Gauss-Newton term as a gradient outer-product expectation:

$$\boldsymbol{H}_{\mathrm{GN}}(\boldsymbol{x}; \boldsymbol{\theta}) \;\coloneqq\; \boldsymbol{J}_{\boldsymbol{\theta}}\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}) \cdot \boldsymbol{S} \cdot \boldsymbol{J}_{\boldsymbol{\theta}}\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x})^{\top} \;=\; \mathbb{E}_{\hat{\boldsymbol{y}}\sim p_{\boldsymbol{\theta}}(\cdot|\hat{\boldsymbol{x}})}\Big[\nabla_{\boldsymbol{\theta}}\ell\big(\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}), \hat{\boldsymbol{y}}\big)\, \nabla_{\boldsymbol{\theta}}\ell\big(\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}), \hat{\boldsymbol{y}}\big)^{\top}\Big]. \qquad (31)$$

Consider a harmful minibatch of size B partitioned into K sub-minibatches of size M ($B = KM$). For each example $\boldsymbol{x}_{k,m}$, sample a label $\hat{\boldsymbol{y}}_{k,m} \sim p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}_{k,m})$ independently and define the per-example gradient

$$\boldsymbol{g}_{k,m} \;\coloneqq\; \nabla_{\theta}\ell\big(\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}_{k,m}), \hat{\boldsymbol{y}}_{k,m}\big) \in \mathbb{R}^d.$$

Define the sub-minibatch gradients $\boldsymbol{g}_k \coloneqq \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{g}_{k,m}$ and $\boldsymbol{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K] \in \mathbb{R}^{d\times K}$.

By (31), the minibatch Gauss-Newton curvature is

$$\boldsymbol{H}_{\mathrm{GN}}^{(B)}(\theta) \;=\; \frac{1}{B}\sum_{k=1}^{K}\sum_{m=1}^{M}\mathbb{E}\big[\boldsymbol{g}_{k,m}\boldsymbol{g}_{k,m}^{\top}\big],$$

where the expectation is over the sampled labels $\{\hat{\boldsymbol{y}}_{k,m}\}$ (conditioning on the minibatch inputs). Moreover, the first Bartlett identity implies $\mathbb{E}[\boldsymbol{g}_{k,m}] = 0$ for each (k, m), and by independence across examples, for $m \neq m'$,

$$\mathbb{E}\big[\boldsymbol{g}_{k,m}\boldsymbol{g}_{k,m'}^\top\big] = \mathbb{E}[\boldsymbol{g}_{k,m}]\,\mathbb{E}[\boldsymbol{g}_{k,m'}]^\top = 0.$$

Therefore,

$$\mathbb{E}\big[\boldsymbol{g}_k\boldsymbol{g}_k^\top\big] \;=\; \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}\big[\boldsymbol{g}_{k,m}\boldsymbol{g}_{k,m}^\top\big].$$

Summing over k and using $\boldsymbol{G}\boldsymbol{G}^\top = \sum_{k=1}^{K}\boldsymbol{g}_k\boldsymbol{g}_k^\top$, we have

$$\mathbb{E}\big[\boldsymbol{G}\boldsymbol{G}^\top\big] \;=\; \frac{1}{M^2}\sum_{k=1}^{K}\sum_{m=1}^{M}\mathbb{E}\big[\boldsymbol{g}_{k,m}\boldsymbol{g}_{k,m}^\top\big].$$

Multiplying both sides by $\frac{M^2}{B}$ yields

$$\mathbb{E}\left[\frac{M^2}{B}\boldsymbol{G}\boldsymbol{G}^\top\right] \;=\; \frac{1}{B}\sum_{k=1}^{K}\sum_{m=1}^{M}\mathbb{E}\big[\boldsymbol{g}_{k,m}\boldsymbol{g}_{k,m}^\top\big] \;=\; \boldsymbol{H}_{\mathrm{GN}}^{(B)}(\boldsymbol{\theta}), \tag{32}$$

so $\widehat{\boldsymbol{H}}_{\mathrm{GN}} := \frac{M^2}{B}\boldsymbol{G}\boldsymbol{G}^\top$ is an unbiased Gram-matrix estimator of the minibatch Gauss-Newton curvature. This proves the theorem.

C Experimental Details

C.1 Defense against harmful fine-tuning in FTaaS

C.1.1 Datasets

The Stanford Sentiment Treebank (SST2) [35] is a binary classification dataset comprising sentences extracted from movie reviews, each labeled as either positive or negative sentiment. The prompt format for this dataset is as follows:

Prompt format with example input for SST2

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:
Analyze the sentiment of the input, and respond only positive or negative.

Input:
a casual intelligence

Response:

The AGNEWS [36] dataset is a collection of news articles categorized into four classes: World, Sports, Business, and Science/Technology. It contains 120k samples in the training split and 7.9k samples in the test split. The prompt format for this dataset is as follows:

Prompt format with example input for AGNEWS

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:
Categorize the news article given in the input into one of the 4 categories:

World
Sports
Business
Sci/Tech

Input:
Einstein's warp effect measured Two scientists beat a $600 million Nasa mission to be first to measure a prediction of Einstein's relativity theory.

Response:

The (Grade School Math 8K) GSM8K [37] is a question-answering dataset comprising 8.5k grade-school math problems. These problems typically require 2-8 steps to solve, primarily involving a sequence of elementary calculations using basic arithmetic operations to arrive at the final answer. The prompt format for this dataset is as follows:

Prompt format with example instruction for GSM8K

Below is an instruction that describes a task. Write a response that appropriately completes the request.

Instruction:
Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Response:

The harmful instructions used in our experiments are from the BeaverTails dataset [34]. This dataset consists of more than 300k pairs of human-labeled question-answer spanning across 14 harm categories, where each response could be flagged with more than one of those categories. Other works [5, 10] use the harmful dataset from [5], which is a filtered subset of BeaverTails-30k-train for alignment and fine-tuning attack, and use a filtered subset of BeaverTails-30k-test for evaluation. However, based on our observation and as pointed out by [3], these datasets suffer from an overlapped problem between the alignment split, fine-tuning attack split, and evaluation split. Therefore, to ensure a fair evaluation, we construct three separate and non-overlapping data splits for our experiments:

1. **Immunization step.** We perform filtering of the dataset in [5] to come up with a subset of 4972 samples for model alignment.
2. **Fine-tuning attack step.** We use the filtered version of BeaverTails-330k curated by [3], including 4986 samples, as the harmful samples that is used to poison the benign dataset.
3. **Evaluation.** We evaluate the Harmful Score of the fine-tuned model on the filtered subset of the BeaverTails-Evaluation dataset [34], which contains 699 malicious questions from 14 categories. We compute Harmful Score using the QA-Moderation model from [34], a classifier that judges whether a question-answer pair is safe or unsafe. QA-Moderation is widely used in prior LLM safety work because it has strong agreement with human judgments while requiring no API cost. We do not conduct additional human evaluation. Instead, we rely on the human-feedback agreement reported by [34], where QA-Moderation achieves agreement rates ranging from 80% to 100% across the tested LLMs.

C.1.2 Training Details

Consistent with prior research [41, 12, 10], we employ LoRA [42] with a rank of 32 and an alpha of 4 to update model parameters during both the alignment and fine-tuning attack phases. All experiments are conducted on a NVIDIA A100, H100 or H200 GPU using a batch size of 16, except for the Gemma2-9B model which requires a batch size of 10 due to memory constraints, and we report the average results across three random seeds. For the immunizaiton step, we train for 20 epochs using the AdamW optimizer [52] with a learning rate of 5×10^{-4} and weight decay of 0.1, while our method's hyperparameters are set to $\rho = \epsilon = 1 \times 10^{-1}$, $\lambda_1 = 5.0$, $\tau = 0.5$, and $\lambda_2 = \lambda_3 = 0.1$. The sub-minibatch size is 4. In the subsequent fine-tuning attack phase, we train on the SST2, AGNEWS, and GSM8K datasets for 20 epochs using AdamW, adjusting the learning rate to 1×10^{-5} and incorporating a warmup ratio of 0.1 alongside a weight decay of 0.1.

C.1.3 Metrics and Evaluation

We evaluate the downstream datasets based on the percentage of exact matches between the generated responses and the ground truth in the test sets:

- **SST2**: A prediction is counted as correct if it exactly matches the labels "positive" or "negative".
- **AGNEWS**: A prediction is counted as correct if it aligns with one of the standard category labels (World, Sports, Business, or Sci/Tech).
- **GSM8K**: A prediction is counted as correct if the concluded answer in the response matches the ground truth solution.

C.1.4 Baselines

This section describes the baseline methods used in our experiments:

- **Supervised Fine-Tuning (SFT)** is the standard approach: we fine-tune the model on the alignment dataset during the alignment stage, and fine-tune it on the harmful dataset during the fine-tuning attack stage.
- **Vaccine** [12] is an alignment-stage defense that improves robustness to harmful fine-tuning by increasing resilience to harmful embedding perturbations. Vaccine also adopts a double-LoRA setup: the LoRA learned during alignment is merged into the base model before the attack stage, and a new LoRA is introduced for the subsequent fine-tuning.
- **Booster** [10] is an alignment-stage method that, besides fitting the alignment data, aims to reduce how quickly the harmful loss can be decreased. Concretely, it minimizes the *harmful loss reduction rate*, defined as the gap between the harmful loss of the current model and the harmful loss after one gradient-descent step on harmful data.
- **Lisa** [41] is a fine-tuning-stage defense that mixes alignment data into the fine-tuning process. It alternates between optimizing on the alignment dataset and on the harmful fine-tuning dataset, and uses a proximal regularizer to limit model drift between consecutive states for more stable training.

C.2 Immunization against harmful fine-tuning attacks in open-weight settings

Here we detail the implementation details of the baseline defense methods and our method:

- **TAR** [6]: We use batch size $= 10$. Hyperparameters: $\lambda_{\mathrm{TR}} = 2$ and $\lambda_{\mathrm{retain}} = 1$.
- **RepNoise** [5]: We use batch size $= 10$. Hyperparameters: $\alpha = 1$ and $\beta = 0.001$.
- **BECON**: We use the same hyperparameters as in Appendix C.1.1.

We reuse the same non-overlapping BeaverTails data construction described in Appendix C.1.1 for harmful fine-tuning resistance evaluation. Unless otherwise stated, we also keep the remaining fine-tuning and evaluation settings consistent with Appendix C.1.1. To reduce compute, we report results using a single random seed.

C.3 Robust Unlearning against Relearning Attack

Unlearning setup. We compare two methods:

- **NPO**: Negative Preference Optimization (NPO) is an unlearning method that discourages the model from generating the forget set examples by treating them as negative preferences. The total objective is minimized as

$$\mathcal{L} = \mathcal{L}_{\text{NPO}} + \gamma \mathcal{L}_{\text{retain}}, \tag{33}$$

  where $\mathcal{L}_{\text{NPO}} = -\log \sigma \left(-\beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right)$ is the NPO loss on the forget set, and $\mathcal{L}_{\text{retain}}$ is the cross-entropy loss on the retain set. For hyperparameters, we use $\beta = 0.0225$, $\gamma = 1.0$, learning rate 5×10^{-6}, and fine-tune for 250 steps with a batch size of 4 and gradient accumulation steps set to 1.
- **NPO+BECON**: This setup is similar to NPO, but with additional regularization. The hyper-parameter strength for $\mathcal{L}_{\text{plateau}}$ is $\lambda_1 = 1.0$. For $\mathcal{L}_{\text{cn-min}}$ and $\mathcal{L}_{\text{cn-max}}$, we use $\lambda_2 = 0.005$ and $\lambda_3 = 0.005$. We set $\rho = \epsilon = 0.01$ and $\tau = 1.0$. The batch size is 4, sub-minibatch size is 2, and gradient accumulation steps are set to 1.

Relearning setup. We use a learning rate of 1×10^{-5}, batch size of 1, and gradient accumulation steps set to 4. The relearning attack is standard SFT on the forget set...

Finally, the results are average over three random seeds.

D More Experiments

D.1 Experiments on More Recent Models

We used Llama-2-7B [38], Qwen-2-7B [39], and Gemma-2-9B [40] as the model architectures in our main experiments to enable direct comparison with prior work [12, 41, 10]. To further evaluate whether our findings hold for more recent models, we conduct additional experiments on Qwen-3-4B [43] under the FTaaS setting following the setup in Section 5.1.

Table 7: Performance of Qwen-3-4B [43] trained on different fine-tuning datasets in the FTaaS setting. The best and the second best are highlighted in blue and gray, respectively.

Methods	SST2		AGNEWS		GSM8K	
	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
SFT	17.45	93.23	15.35	83.20	7.77	65.53
Booster	3.96	93.50	2.10	**83.27**	6.91	**70.80**
BECON	**2.67**	**93.54**	**1.33**	82.13	**4.77**	69.57

As shown in Table 7, BECON continues to achieve the strongest Harmful Score results while maintaining competitive Fine-tuning Accuracy.

D.2 Full Fine-Tuning Experiments

Most experiments in this work use LoRA [42], following prior defenses against harmful fine-tuning [41, 12, 10], as LoRA is the most common and practical setting in both FTaaS and open-weight fine-tuning. During immunization, however, the learned LoRA update is merged into the base model before release. The released checkpoint is therefore a standard dense model whose immunization is incorporated into its model parameters and cannot be directly removed by discarding an adapter. This makes LoRA a suitable and efficient mechanism for the immunization stage, while still allowing post-release adversaries to attack the resulting dense model using either LoRA or full-parameter fine-tuning.

To evaluate full-parameter fine-tuning attacks after model release, we conduct additional experiments on the LoRA-immunized Llama-2-7B model. We use 100 harmful samples and otherwise follow the

Table 8: Harmful immunization performance of Llama-2-7B under full-parameter fine-tuning across different learning rates. The number of harmful samples is 100.

Methods	**Harmful Score ↓**				
	lr=1e-6	lr=2e-6	lr=5e-6	lr=7e-6	lr=1e-5
BECON	0.86	0.72	0.86	1.14	4.86

Table 9: Harmful immunization performance of Llama-2-7B under full-parameter fine-tuning across different numbers of epochs. The number of harmful samples is 100.

Methods	**Harmful Score ↓**				
	Epoch=1	Epoch=2	Epoch=3	Epoch=4	Epoch=5
BECON	0.86	1.57	1.72	3.43	5.87

open-weight immunization setup in Section 5.2. As shown in Tables 8 and 9, BECON maintains low Harmful Score across both learning-rate and epoch sweeps, indicating that the immunized dense model remains robust even when the attacker updates all model parameters.

D.3 Additional Baselines in the Open-Weight Setting

In Section 5.2, we compare BECON with TAR and RepNoise because both were designed for the open-weight setting. Since Vaccine [12] and Booster [10] are also alignment-stage defenses that act before the model is released, we further evaluate them under the same open-weight attack protocol, sweeping the attacker's number of fine-tuning epochs and learning rate as in Tables 3 and 4.

Table 10: Harmful immunization performance across training epochs, extending Table 3 with Vaccine and Booster. The top three results in each column are highlighted in bold.

Methods	**Harmful Score ↓**					
	Epoch=5	Epoch=10	Epoch=20	Epoch=50	Epoch=75	Epoch=100
TAR	42.63	50.64	54.51	55.08	55.79	**54.94**
RepNoise	13.02	16.31	**15.74**	**43.35**	**53.51**	55.79
Vaccine	**1.57**	**8.30**	30.04	55.08	56.94	59.37
Booster	**1.43**	**2.58**	**24.32**	**48.78**	**49.21**	**51.93**
BECON	**1.00**	**3.43**	**27.47**	**50.21**	**50.50**	**52.50**

Table 11: Harmful immunization performance across different learning rates, extending Table 4 with Vaccine and Booster. The top three results in each column are highlighted in bold.

Methods	**Harmful Score ↓**					
	lr=5e-6	lr=1e-5	lr=5e-5	lr=1e-4	lr=5e-4	lr=1e-3
TAR	28.47	42.63	56.08	55.79	**54.08**	**54.94**
RepNoise	10.16	13.02	**9.63**	54.51	**52.36**	**56.08**
Vaccine	**0.86**	**1.57**	35.62	**54.36**	55.94	57.80
Booster	**1.00**	**1.43**	**34.48**	**49.07**	55.65	56.22
BECON	**1.14**	**1.00**	**34.33**	**47.93**	**54.79**	**54.51**

As shown in Tables 10 and 11, BECON is the only method that ranks among the top three in every epoch and learning-rate setting. Booster is the strongest additional baseline. It attains a lower harmful score than BECON in five of the six epoch settings, by at most 3.15 points, while BECON attains a lower harmful score than Booster in five of the six learning-rate settings, and Booster falls out of the top three at learning rates of 5×10^{-4} and 10^{-3}. Vaccine is competitive under small attack budgets but has the highest harmful score among all methods under the largest ones, reaching 59.37 after 100

epochs and 57.80 at a learning rate of 10^{-3}. Like the other defenses, BECON becomes less effective as the attack budget grows, which motivates future work on more robust defenses in this challenging regime.

D.4 Seed Variance in the Open-Weight Setting

The open-weight results in Section 5.2 are reported for a single random seed. Since RepNoise attains a lower harmful score than BECON at 20 epochs and at a learning rate of 5×10^{-5} in Tables 3 and 4, we rerun both methods with three random seeds at these two attack configurations to assess whether this ordering is stable.

Table 12: Harmful score of RepNoise and BECON across three random seeds in the open-weight setting of Section 5.2, at the two attack configurations where RepNoise outperforms BECON in Tables 3 and 4. Lower is better.

Method	**Epoch=20**				**lr=5e-5**			
	First seed	Second seed	Third seed	Mean $\pm$ std	First seed	Second seed	Third seed	Mean $\pm$ std
RepNoise	15.74	44.35	36.91	32.33 $\pm$ 14.84	9.63	46.21	42.20	32.68 $\pm$ 20.06
BECON	27.47	24.61	24.57	25.55 $\pm$ 1.66	34.33	33.05	32.90	33.43 $\pm$ 0.79

As shown in Table 12, RepNoise is much less stable across seeds than BECON, with standard deviations of 14.84 and 20.06 compared with 1.66 and 0.79 for BECON. The ordering in the main tables comes from the first seed, which is the only seed in which RepNoise attains a lower harmful score than BECON in either configuration. In the other two seeds, BECON attains a lower harmful score in both configurations. Averaged over seeds, BECON has a lower harmful score at 20 epochs (25.55 vs. 32.33), while the two methods are comparable at a learning rate of 5×10^{-5} (33.43 vs. 32.68).

D.5 Robustness to Harmful Data Distribution Shift

Prior experiments in Section 5.1 attack with harmful samples drawn from the same BeaverTails distribution as the immunization data, while ensuring the attack set is disjoint from the immunization set. Here, we additionally evaluate robustness under a distribution shift in the harmful data by replacing the attack samples with PureBad-100 [2]. We keep the rest of the FTaaS setup unchanged and use a harmful ratio of $p = 0.1$.

Table 13: Performance of defense methods on different fine-tuning datasets with a harmful ratio of 0.1, where harmful attack samples are drawn from PureBad-100 [2]. The setup otherwise follows Section 5.1. The best results are highlighted in bold.

Method (Llama-2-7B)	**SST2**		**AGNEWS**		**GSM8K**	
	HS $\downarrow$	FA $\uparrow$	HS $\downarrow$	FA $\uparrow$	HS $\downarrow$	FA $\uparrow$
SFT	15.79	92.78	14.50	86.47	11.30	11.63
BECON (Ours)	**2.73**	**93.00**	**2.96**	**87.70**	**2.48**	**15.33**

As shown in Table 13, BECON remains robust when the harmful attack data comes from a different distribution. Across SST2, AGNEWS, and GSM8K, it substantially reduces Harmful Score relative to standard fine-tuning while preserving, and in these experiments slightly improving, downstream task accuracy.

E System Evaluation

We provide an additional system-level comparison to clarify the computational overhead of the methods evaluated in Section 5.2. Here, efficiency refers to making condition-number-based immunization practical in the large-scale LLM setting, including the estimation of Hessian singular-value surrogates, rather than claiming uniformly lower overhead than prior harmful fine-tuning defenses.

Table 14: System evaluation for methods in Section 5.2 on Llama-2-7B. Lower is better for all metrics.

Method (Llama-2-7B)	Clock Time (Hours) ↓	Peak Reserved (GB) ↓	Peak Allocated (GB) ↓
TAR	2.7	78.07	61.06
RepNoise	2.6	87.13	65.89
BECON	5.7	59.96	51.71

As shown in Table 14, TAR and RepNoise run faster than BECON, but they require substantially more GPU memory. In particular, TAR and RepNoise reach 2.7 and 2.6 hours, respectively, while BECON requires 5.7 hours. However, BECON reduces peak reserved memory to 59.96 GB, compared with 78.07 GB for TAR and 87.13 GB for RepNoise, and likewise reduces peak allocated memory to 51.71 GB, compared with 61.06 GB and 65.89 GB.

This trade-off is consistent with how the methods are implemented. TAR and RepNoise retain hidden states to compute their losses, which increases memory usage. BECON instead keeps only a single copy of the model and reuses gradients to recover the model between perturbations, reducing memory usage at the cost of additional wall-clock time. More broadly, curvature-based immunization remains computationally demanding because the exact Hessian condition number is infeasible to compute for LLMs, and although BECON's approximations in Section 4 reduce GPU memory relative to the condition-number framework in [8] (59.96 GB vs. 79.53 GB), they require a longer runtime (Table 23). Moreover, this overhead is a *one-time cost* incurred during the immunization stage before the model is released, and we leave further efficiency improvements to future work.

F Further Discussion with Prior Works

BECON is related to two lines of prior work, condition-number-based immunization [8] and the attenuation of harmful perturbations in Booster [10]. Our contribution is not the union of their objectives, but the result of Theorem 3.3 that the two components must be combined, together with a realization of condition-number control that is usable at LLM scale. BECON therefore implements this analysis rather than combining existing objectives post hoc. Neither line of prior work predicts the necessity of the other. Zheng et al. [8] certify slow harmful fine-tuning only through an upper bound, while Booster provides no curvature or convergence analysis and makes no claim about the Hessian spectrum. Moreover, a direct adaptation of Zheng et al. [8] to LLM immunization does not reduce the harmful score relative to SFT (Tables 1 and 22) and requires more GPU memory than BECON (Table 23).

Discussion with Booster. Booster [10] is an alignment-stage defense that attenuates harmful perturbations by minimizing the reduction in harmful loss after a small harmful update. BECON's harmful plateau promotion has a closely related form: it also discourages a local decrease in harmful loss after moving in a harmful update direction. The distinction is primarily in the role this term plays in the full method. In Booster, attenuating harmful loss reduction is the central objective. In BECON, harmful plateau promotion is introduced as a complement to condition-number enlargement, motivated by the theoretical observation that ill conditioning alone does not necessarily prevent rapid harmful optimization when harmful gradients remain large. This motivation comes from the lower bound in Theorem 3.3, in which small harmful gradient norms are required in addition to a large condition number, rather than from the heuristic of attenuating harmful loss reduction. Reaching a closely related formulation from this principle explains why the term is needed and what it must be combined with. Thus, the plateau term suppresses locally harmful gradients while the condition-number terms distort the harmful curvature geometry.

Table 15 compares Booster with the BECON variant that keeps only harmful plateau promotion and disables the condition-number terms. This isolates the practical effect of BECON's weighted plateau objective from the full condition-number component. The weighted plateau variant slightly improves harmful score on SST2 and GSM8K while maintaining comparable downstream fine-tuning accuracy. On AGNEWS, the two methods are essentially tied in safety and utility. These results suggest that the plateau weighting can be useful, while the strongest gains in BECON come from combining plateau promotion with condition-number enlargement.

Table 15: Effect of the harmful plateau promotion term relative to Booster in the FTaaS setting on Llama2-7B. Results are averaged over three random seeds. HS denotes Harmful Score, and FA denotes Fine-tuning Accuracy.

Method	SST2		AGNEWS		GSM8K	
	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
Booster	14.31	92.59	**15.88**	**86.70**	9.06	**16.27**
BECON (*harmful plateau only*)	**13.07**	**92.81**	15.93	86.67	**8.92**	15.30

Comparison with Booster on more recent models. We further compare BECON with Booster on Llama-3.1-8B [53] and Qwen3-8B [43] under the FTaaS setting of Section 5.1. As shown in Table 16, BECON outperforms Booster in both harmful score and fine-tuning accuracy on all three datasets for both models, although the gain is small in some cases.

Table 16: Comparison between Booster and BECON on more recent models in the FTaaS setting. Results are reported as mean ± standard deviation. HS denotes Harmful Score, and FA denotes Fine-tuning Accuracy. The best results for each model are highlighted in bold.

Model	Method	SST2		AGNEWS		GSM8K	
		HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
Llama-3.1-8B	Booster	11.26 ± 4.59	93.23 ± 0.34	6.34 ± 0.65	83.80 ± 0.62	7.87 ± 3.78	45.50 ± 1.04
	BECON	**6.16** ± 0.40	**93.92** ± 0.49	**4.94** ± 1.32	**84.20** ± 0.14	**4.79** ± 0.91	**45.75** ± 1.34
Qwen3-8B	Booster	3.00 ± 0.76	93.65 ± 0.46	2.19 ± 0.46	83.17 ± 0.75	1.95 ± 0.36	72.63 ± 0.45
	BECON	**2.29** ± 0.66	**93.69** ± 0.41	**1.29** ± 0.38	**84.17** ± 0.68	**1.62** ± 0.46	**73.40** ± 0.44

Discussion on GSM8K. In Table 1, Booster attains higher fine-tuning accuracy than BECON on GSM8K for all three models and a lower harmful score on Qwen2-7B. On Llama-3.1-8B and Qwen3-8B, however, BECON outperforms Booster on GSM8K in both metrics (Table 16). We therefore conjecture that the gap in Table 1 mainly comes from our use of a single default hyperparameter configuration across all fine-tuning datasets, which may not be optimal for GSM8K. Since GSM8K is a mathematical reasoning dataset that differs more from SST2 and AGNEWS than these two differ from each other, dataset-specific factors may also contribute, and we leave the study of such defenses on reasoning models and datasets to future work.

Discussion with Antibody. Both Antibody [33] and BECON use weighting mechanisms to account for the fact that harmful samples do not contribute equally during optimization. Antibody combines an alignment-stage flatness objective with a fine-tuning-stage weighting mechanism that reduces harmful gradient influence during downstream fine-tuning. BECON is closest to Antibody's alignment-stage component in that both encourage a flatter harmful loss landscape before the model is fine-tuned. Both thus relate to the literature on Sharpness-Aware Minimization [26, 54–56]. The main difference is where the weighting is applied and what signal defines the weights. BECON incorporates per-sample weighting directly into the immunization objective. Its weights are computed from the sensitivity of each harmful sample to the plateau perturbation, so the alignment stage emphasizes harmful samples whose losses are most affected by the local harmful update. In contrast, Antibody's fine-tuning-stage weights compare safe and target response likelihoods to reduce harmful gradient influence while preserving learning on benign task data.

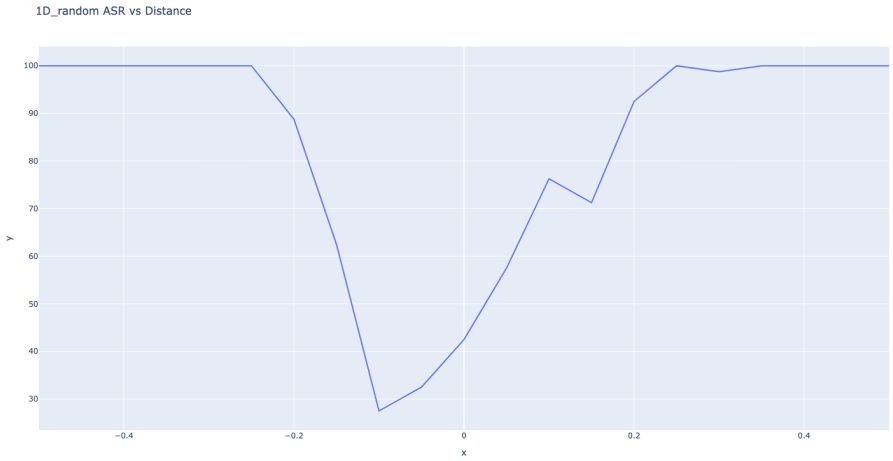
Table 17: Comparison with Antibody's alignment-stage solution in the FTaaS setting on Llama2-7B and GSM8K.

Method	HS ↓	FA ↑
Antibody (*alignment stage only*)	10.87	15.40
BECON	**3.72**	**15.43**

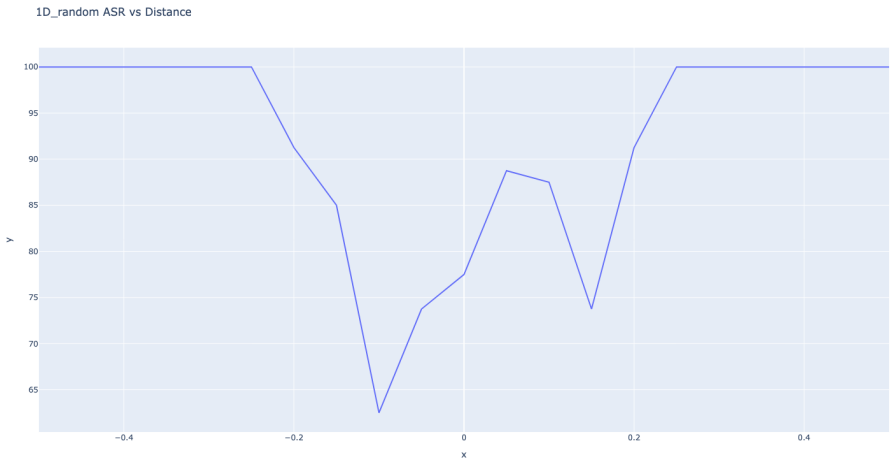
Because Antibody includes both alignment-stage and fine-tuning-stage components, Table 17 compares BECON against Antibody's alignment-stage solution to isolate the setting in which both methods intervene before downstream fine-tuning. BECON achieves a lower harmful score than

Antibody's alignment-stage component on GSM8K while preserving similar fine-tuning accuracy. This comparison supports the view that BECON advances alignment-stage immunization by combining harmful plateau promotion with condition-number regularization, rather than relying on flatness alone.

G Safety Loss Landscape



(a) BECON



(b) Booster

Figure 2: 1D ASR safety landscapes. Note the smaller y-axis range for BECON compared to Booster.

Figure 2 visualizes the 1D Attack Success Rate (ASR) landscape of BECON and Booster using the protocol of Peng et al. [57]. The VISAGE score proposed in Peng et al. [57] measures the average safety margin over sampled perturbations in the local safety landscape, with higher values indicating a safer local region around the model. BECON achieves a VISAGE score of 31.81, higher than Booster's 18.75, suggesting that the combination of condition-number regularization and smoothness promotion creates a stronger local safety barrier.

H Additional Discussion on Condition-Number Regularization

This section provides additional discussion of the condition-number component in BECON. A natural question is whether the observed defense effect should be attributed to the intended condition-number-enlarging mechanism, or more simply to suppressing harmful gradients and promoting flatter harmful regions. Our theoretical analysis in the main paper already shows that condition number alone is not sufficient to explain harmful fine-tuning difficulty, which is precisely why BECON combines condition-number regularization with harmful loss plateau promotion. At the same time, condition-number enlargement remains a useful component of the overall method: **Table 18 shows that condition-number regularization alone consistently reduces Harmful Score relative to SFT across models and tasks, although the full BECON objective is usually stronger**.

Table 18: Performance of different methods across models. Numbers in parentheses indicate the change in Harmful Score relative to SFT, where negative values indicate improved safety. The harmful score reduction effect of condition number regularization is highlighted in bold.

	Model & Method	**SST2**		**AGNEWS**		**GSM8K**	
		HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
Llama2-7B	SFT	36.29	92.70	34.57	85.40	23.94	10.90
	BECON (*condition-number only*)	12.49 **(-23.80)**	92.70	13.59 **(-20.98)**	85.93	20.13 **(-3.81)**	15.00
	BECON	11.16 (-25.13)	93.15	13.59 (-20.98)	86.77	3.72 (-20.22)	15.43
Gemma2-9B	SFT	40.77	94.11	39.91	84.47	32.05	56.73
	BECON (*condition-number only*)	19.98 **(-20.79)**	94.38	14.21 **(-25.70)**	84.80	8.72 **(-23.33)**	57.60
	BECON	16.21 (-24.56)	94.42	4.24 (-35.67)	84.80	8.15 (-23.90)	58.07
Qwen2-7B	SFT	33.86	93.31	31.00	85.77	14.54	64.66
	BECON (*condition-number only*)	7.29 **(-26.57)**	93.77	7.10 **(-23.90)**	87.77	3.62 **(-10.92)**	67.17
	BECON	3.34 (-30.52)	94.76	2.91 (-28.09)	86.00	2.48 (-12.06)	67.93
Qwen3-4B	SFT	17.45	93.23	15.35	83.20	7.77	65.53
	BECON (*condition-number only*)	6.01 **(-11.44)**	93.81	3.96 **(-11.39)**	83.87	3.05 **(-4.72)**	69.70
	BECON	2.67 (-14.78)	93.54	1.33 (-14.02)	82.13	4.77 (-3.00)	69.57

To verify that this reduction is not an artifact of a particular training run, we rerun the condition-number-only variant on Qwen2-7B with three random seeds in the FTaaS setting of Section 5.1. As shown in Table 19, the standard deviation across seeds is at most 1.47 points of harmful score, whereas the reduction relative to SFT ranges from 10.92 to 26.57 points. The reduction is therefore at least seven times the seed-to-seed standard deviation on every dataset, which confirms that the condition-number term alone gives a reliable harmful score reduction.

Table 19: Harmful score of BECON (*condition-number only*) on Qwen2-7B across three random seeds, reported as mean ± standard deviation. SFT values are taken from Table 18. HS denotes Harmful Score.

Method (Qwen2-7B)	SST2 HS ↓	AGNEWS HS ↓	GSM8K HS ↓
SFT	33.86	31.00	14.54
BECON (*condition-number only*)	7.29 ± 1.03	7.10 ± 0.93	3.62 ± 1.47

Additionally, to clarify why this component is implemented through a surrogate rather than the exact Hessian condition number, it is important to consider the computational constraints of the LLM setting. Directly optimizing the true Hessian condition number is infeasible in large autoregressive models. Computing $\kappa(H) = \sigma_{\max}(H)/\sigma_{\min}(H)$ requires estimating both extremal singular values of a very large Hessian, which is prohibitively expensive in practice. For this reason, some surrogate is unavoidable in large-scale settings. This approximation issue is not unique to our method. Prior condition-number-based immunization frameworks such as Zheng et al. [8] also rely on nontrivial approximations, since their analysis is developed for linear models with squared loss and their deep-net extension estimates a linear-probing feature-space Hessian, rather than the Hessian over all trainable parameters. In contrast, BECON is developed directly for large autoregressive models and uses the practical surrogate defined in Equation (16) to aggregate curvature information across the trainable parameter vector, such as all LoRA parameters, without explicitly computing Hessian singular values.

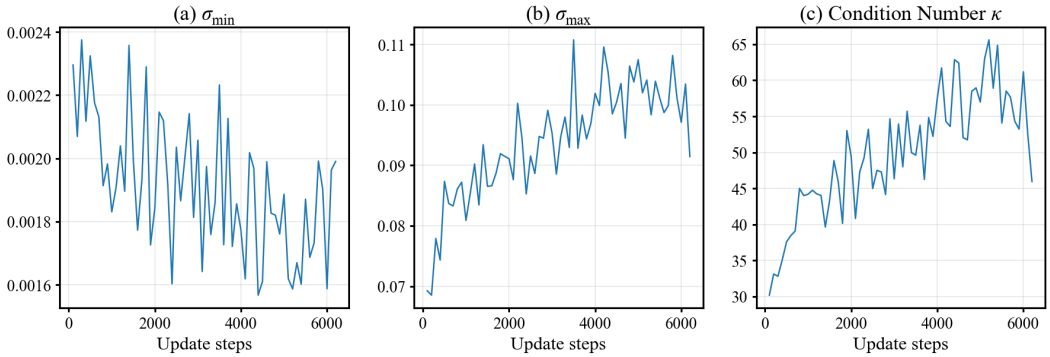


Figure 3: Evolution of $\sigma_{\min}$, $\sigma_{\max}$, and the condition number $\kappa = \sigma_{\max}/\sigma_{\min}$ during the immunization process of our method. Due to the substantial computational cost of estimating the condition number, which involves Hessian computations, we conduct this experiment on the Qwen3-0.6B model and visualize the metrics for a single LoRA module in the final block, specifically the LoRA of the v_{proj} matrix. The decrease in $\sigma_{\min}$ and the increase in $\sigma_{\max}$ together produce the desired effect of an increasing condition number.

Although the surrogate defined in Equation (16) is not the exact condition number, it is designed to decrease a proxy for $\sigma_{\min}$ and increase a proxy for $\sigma_{\max}$. Figure 3 visualizes this trend directly. The surrogate does not merely reduce harmful gradients uniformly; it also changes the spectrum in the intended direction by decreasing $\sigma_{\min}$, increasing $\sigma_{\max}$, and thereby enlarging the effective condition number.

Small harmful gradients do not imply proximity to a harmful optimum. The lower bounds in Theorem 3.3 increase as the harmful gradient norms decrease, so BECON promotes small harmful gradients. One may ask whether a small harmful gradient instead indicates that the model is already close to the harmful optimum. This is not the case. BECON creates the plateau at a safety-aligned parameter, not at a harmful optimum. The overall objective always retains the alignment loss $\mathcal{L}_{\text{align}}$ on $\mathcal{D}_{\text{align}}$, which pairs harmful requests with safe responses and anchors the model in the refusal region. The plateau and condition-number terms then flatten and distort the harmful loss landscape around this safe point. A small harmful gradient therefore means that the model sits in a flat region where gradient descent on harmful data has little to follow, not that the harmful task has been learned. Lemma 3.1, applied to the harmful loss $\mathcal{L}$, is consistent with this view. Rearranging its two parts gives

$$\frac{\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2}{2\sigma_{\max}} \;\le\; \mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^* \;\le\; \frac{\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2}{2\sigma_{\min}}. \tag{34}$$

The right inequality, which follows from Lemma 3.1(i), is the only one that bounds the harmful gap from above by the gradient norm, and hence the only one that could certify that a small gradient implies closeness to the harmful optimum. It does so only when $\sigma_{\min}$ is bounded away from zero. The condition-number term of BECON deliberately decreases $\sigma_{\min}$ (Figure 3), which makes this upper bound loose, so a small harmful gradient is compatible with a large remaining harmful gap. The left inequality, which follows from Lemma 3.1(ii), is a lower bound on the harmful gap. A small gradient only makes this lower bound uninformative and never implies that the gap itself is small.

Table 20: Harmful score before and after the fine-tuning attack on Llama2-7B in the FTaaS setting of Section 5.1, with GSM8K as the fine-tuning dataset. The pre-attack harmful score is measured right after alignment for SFT and right after immunization for BECON, and is reported as mean $\pm$ standard deviation over three random seeds. Post-attack values are taken from Table 1. HS denotes Harmful Score.

Method (Llama2-7B)	Pre-attack HS ↓	Post-attack HS (GSM8K) ↓
SFT	0.81 ± 0.30	23.94
BECON	0.67 ± 0.22	3.72

Empirically, Table 20 shows that before any attack, BECON is at least as safe as the SFT model, with a harmful score of 0.67 compared with 0.81. The small harmful gradients of BECON therefore do not come from being close to a harmful solution. After harmful fine-tuning on GSM8K, BECON also remains far safer than SFT (3.72 vs. 23.94). Suppressing the harmful gradient norm thus keeps the model at a safety-aligned parameter surrounded by a flat and ill-conditioned harmful landscape, which is the behavior prescribed by Theorem 3.3.

Ground-truth labels versus sampled pseudo-labels. Theorem 4.1 builds the Gram matrix from gradients computed with labels sampled from the model's predictive distribution, following Sophia [31]. In our implementation (Algorithm 1), we instead compute these gradients with the ground-truth harmful labels. The implemented object is therefore an empirical-Fisher-type Gram matrix, which is not in general an unbiased estimator of the Gauss–Newton matrix. This substitution is a well-established design choice. Curvature estimates built from per-example gradients on observed labels are common in practice [58, 59], and HELENE [51] explicitly replaces the sampled labels of Sophia's estimator with the true labels. Using ground-truth labels is also deterministic and cheaper than sampling a label at every token, and it evaluates the same harmful-label loss that the attacker optimizes.

Table 21: Comparison between the ground-truth-label estimator used in our implementation and the pseudo-label estimator prescribed by Theorem 4.1, in the FTaaS setting of Section 5.1 on Llama2-7B. HS denotes Harmful Score, and FA denotes Fine-tuning Accuracy.

Method (Llama2-7B)	**SST2**		**AGNEWS**		**GSM8K**	
	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
BECON (*ground-truth labels*, as implemented)	11.16	93.15	13.59	86.77	3.72	15.43
BECON (*sampled pseudo-labels*, Theorem 4.1)	12.83	92.93	13.50	85.63	4.39	15.70

Since the two estimators do not coincide in general, we also evaluate the estimator exactly as prescribed by Theorem 4.1. This variant samples pseudo-labels token-wise from the model's predictive distribution and keeps every other component of BECON unchanged. As shown in Table 21, the two variants are closely comparable in both harmful score and fine-tuning accuracy on all three datasets, with differences of at most 1.7 points in harmful score and 1.1 points in fine-tuning accuracy. These results indicate that the ground-truth-label implementation is a reliable practical approximation of the theory-matched estimator.

Table 22: Performance of defense methods on different fine-tuning datasets. The setup is similar to Section 5.1 of the main paper. The best results are highlighted in bold.

Method (Llama2-7B)	SST2 HS	SST2 FA	AGNEWS HS	AGNEWS FA	GSM8K HS	GSM8K FA
Framework in [8]	38.57	**93.58**	38.01	84.33	27.75	11.53
BECON	**11.16**	93.15	**13.59**	**86.77**	**3.72**	**15.43**

Table 23: System cost of different condition-number-based methods on Llama2-7B.

Method	Clock Time (hours)	GPU Memory (GB)
Framework in [8]	1.53	79.53
BECON	5.70	59.96

Finally, we also compare BECON against a prior condition-number-based framework adapted to the LLM immunization setting. As shown in Table 22, the previous framework is substantially less effective than BECON across all three downstream fine-tuning datasets on Llama2-7B. Table 23 further shows the corresponding time–memory tradeoff: the adapted prior framework has a shorter clock time, but requires substantially higher peak GPU memory because of its feature-Gram and singular-value computation. **Although our method has a higher one-time runtime, it achieves better memory efficiency, defense performance, and fine-tuning accuracy.** These results support the claim that BECON's practical surrogate is not only computationally viable, but also more effective in the LLM setting than directly adapting earlier condition-number-based approaches.

I Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning by improving the safety and robustness of Large Language Models against harmful fine-tuning. The proposed immunization framework is intended to reduce the risk that deployed or open-weight models can be adapted to produce harmful behavior. However, documenting the weaknesses of current fine-tuning safeguards may also lower the barrier for adversaries to study and exploit these vulnerabilities. We therefore emphasize that the method is presented to strengthen defenses and encourage responsible deployment and evaluation practices.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims are accurate.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include the discussion of limitation in Section 8 and computational efficiency in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We clearly state the assumptions and provide all the proofs in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide full details sufficient to reproduce the main experiment in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code is available at https://github.com/minhquoc0712/BECON. The datasets used in this paper are publicly available online.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify the main experimental settings in the main paper and provide full details in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We vary hyperparameters and evaluate across diverse attack settings, including different model architectures, datasets, and harmful ratios. The breadth of these experiments provides sufficient evidence for statistical significance. Furthermore, most of the main experiment results are averaged over three random seeds, providing additional confidence in the robustness of our findings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the GPU types in Appendix C, and the GPU memory and wall-clock time of BECON and the baselines in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research fully complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we include the impact statement in Appendix I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available datasets and models with proper citation and adhere to their respective licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our code at https://github.com/minhquoc0712/BECON.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.