DUAL-DISTILLATION FOR TAMPER-RESISTANT AND ALIGNED LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensuring the safety of large language models during post-deployment fine-tuning remains a significant challenge, as conventional safety mechanisms are often brittle and prone to false positives during benign updates. To address this, we introduce a novel teacher–student training framework that develops a specialized subset of "canary" parameters dedicated to security.

Our approach, the Cyclical Immunization and Alignment Algorithm, alternates between two phases. In the sensitization phase, a harmful teacher is used for adversarial distillation: both the student and the teacher are given the same prompt, and the harmful teacher's unsafe response is treated as a negative target. This forces the student to refuse dangerous instructions, while dynamically identifying and tagging the most safety-relevant parameters as canaries. In the desensitization phase, a helpful teacher and preference data guide alignment and skill learning. Here, the model is trained on reasoning, comprehension, and structured generation tasks, while a stabilization loss penalizes any drift in the canary parameters.

This design compels the student to acquire new capabilities using non-critical weights, effectively decoupling safety from general-purpose learning. As a result, the model's safety mechanism remains highly sensitive to malicious updates but inert to harmless fine-tuning, thereby reducing false alarms. This work outlines a pathway toward AI systems that are both adaptable and reliably safe, even under continual post-deployment updates.

1 INTRODUCTION

2 RELATED WORK

3 PRELIMINARIES

The proposed methodology is built upon several established and recent techniques in machine learning and large language model training. This section provides the theoretical background for the core components integrated into our framework: Curriculum Learning, Direct Preference Optimization, Negative Preference Optimization, KL-divergence for regularization, Latent Attack Training, and the concept of canary detection as a form of heuristic circuit breaking.

3.1 CURRICULUM LEARNING (CL)

Curriculum Learning is a training strategy inspired by human learning, where a model is trained on examples in a meaningful order, typically from easy to hard . Instead of presenting data in a random order, a curriculum organizes the training data to gradually introduce more complex concepts. In the context of safety training, this approach is particularly effective. Our methodology employs CL by structuring the adversarial datasets into tiers of increasing difficulty (e.g., "weak to strong"). This allows the model to first learn to identify and refuse simple, explicit harmful requests before being

exposed to more nuanced, implicit, or complex adversarial attacks. This incremental hardening process helps prevent model collapse during training and leads to a more robust and generalizable safety foundation.

3.2 Direct Preference Optimization (DPO)

Direct Preference Optimization (DPO) is a paradigm for aligning language models with human preferences that is more stable and lightweight than traditional Reinforcement Learning from Human Feedback (RLHF) . Traditional RLHF involves training a separate reward model on preference data and then using reinforcement learning (e.g., PPO) to optimize the LLM against this learned reward. DPO bypasses the need for an explicit reward model by reformulating the objective as a direct classification problem on preference pairs. Given a dataset of prompts x and corresponding pairs of preferred (y_w) and dispreferred (y_l) responses, DPO directly optimizes the language model's policy π_θ to increase the likelihood of preferred responses over dispreferred ones.

3.3 Refusal Training via Negative Preference Optimization (NPO)

To explicitly teach the model to refuse harmful instructions, we adapt the preference optimization framework for a refusal-centric objective, which we term Negative Preference Optimization (NPO). While DPO is used to align a model with helpful behaviors, NPO is used to misalign it with harmful ones.

The training data for NPO consists of triplets: a harmful prompt (x_h), a desired safe refusal response (y_r, the "preferred" response), and the unsafe, harmful response the model should avoid (y_h, the "rejected" response). The NPO loss is structured analogously to the DPO loss, but its goal is to maximize the log-probability gap between the refusal and the harmful completion:

$$L_{\text{NPO}} = -\mathbb{E}_{(x_h, y_r, y_h)\sim\mathcal{D}_{\text{adv}}}\left[\log\sigma\left(\beta\left(\log\frac{\pi_\theta(y_r \mid x_h)}{\pi_{\text{ref}}(y_r \mid x_h)} - \log\frac{\pi_\theta(y_h \mid x_h)}{\pi_{\text{ref}}(y_h \mid x_h)}\right)\right)\right]$$

This directly trains the model to prefer refusing over complying with harmful instructions. This preference-based formulation is more powerful than simply penalizing the harmful output, as it simultaneously teaches the model the desired alternative behavior. This is complemented in our framework by a Reverse KL-Divergence (RKL) loss against a harmful teacher model, which provides a broader distributional penalty to comprehensively steer the model away from unsafe response styles.

3.4 Canary Detection as Heuristic Circuit Breaking

Mechanistic interpretability shows that LLMs form "circuits"—subnetworks of neurons and attention heads responsible for specific behaviors, from factual recall to unsafe generations. Circuit breaking is a safety approach that intervenes in these pathways to prevent undesirable outputs.

We implement a heuristic form of circuit breaking by identifying parameters most critical to refusal training. The gradient of the refusal loss, $\nabla_\theta L_{\text{refusal}}$, reveals which parameters contribute most to safety behavior. Those with the highest gradient magnitudes are designated as 'canary_parameters'—a sensitive subnetwork acting as a "canary in the coal mine."

In Phase 1, these parameters are sensitized to harmful inputs; in Phase 2, they are frozen with a stabilization loss. This creates a dedicated, non-erasable circuit breaker that preserves core safety mechanisms during later alignment.

3.5 Latent Attack Training (LAT)

Latent Attack Training (LAT) is a form of adversarial training that aims to improve model robustness by hardening it against worst-case perturbations in its own parameter space. Instead of modifying the input data, a latent attack directly targets the model's internal weights. The process, as adapted in our framework, involves two stages. First, an "attack" is simulated by finding a small perturbation, δ, for the model's weights, θ. This perturbation is calculated by taking a gradient step in the direction that maximizes a specific loss—in our case, the likelihood of generating a harmful response. This

creates a temporarily "hardened" or compromised model state, $\theta' = \theta + \delta$. Second, the actual training update is performed from this perturbed state, forcing the model to learn how to recover and produce the correct output (i.e., refusal) even when its parameters are adversarially shifted. This method directly inoculates the model against vulnerabilities that could be exploited to elicit unsafe behavior.

3.6 The Dual Role of KL-Divergence

The Kullback-Leibler (KL) divergence is a measure of the difference between two probability distributions, P and Q. Crucially, it is asymmetric ($D_{KL}(P||Q) \neq D_{KL}(Q||P)$), and this asymmetry allows it to be used for two distinct purposes in our framework: stable regularization and aggressive refusal training.

Forward KL-Divergence for Regularization. In the alignment phase, we use the standard forward KL-divergence, $D_{KL}(\pi_{\text{ref}}||\pi_\theta)$, as a regularization term. This objective is "mean-seeking" or "zero-avoiding"; to minimize the loss, the student model's policy π_θ must assign a non-zero probability to any response that the reference model π_{ref} might generate. This behavior encourages the student model to maintain a broad distribution similar to the reference, preventing it from over-optimizing on the preference data and suffering from "catastrophic forgetting" of its general capabilities.

Reverse KL-Divergence for Aggressive Refusal. In the immunization phase, we use the Reverse KL-Divergence (RKL), $D_{KL}(\pi_{\text{harmful}}||\pi_\theta)$, as a core component of the refusal loss. This objective is "mode-seeking" or "zero-forcing." To minimize the RKL, the student policy π_θ must have near-zero probability wherever the harmful teacher's policy π_{harmful} has low probability. This aggressively penalizes the student model for assigning any significant probability to the entire distribution of harmful responses, forcing it to learn a sharp, narrow, and distinctly non-harmful response mode. This makes it a powerful tool for teaching the model to comprehensively avoid unsafe behaviors.

4 Proposed Work

4.1 Overview

The methodology presented in this paper aims to create a safely extensible Large Language Model (LLM) by developing a specialized subset of internal safety parameters. The core approach is a novel training framework designed to induce a "separation of concerns" within the model's weights, where some parameters are dedicated to safety refusal while others remain adaptable for general-purpose tasks. This is achieved through a cyclical, two-phase training algorithm that orthogonally trains for safety and helpfulness. The first phase sensitizes a dynamically identified set of "canary" parameters to harmful inputs, while the second phase desensitizes these same parameters during harmless alignment training. This dual-objective approach was chosen to overcome the brittleness of conventional safety mechanisms, which often produce false positives during benign fine-tuning, thereby enabling continuous model improvement without compromising foundational safety.

4.2 Data and Inputs

The training process utilizes two distinct categories of datasets.

- **Adversarial Datasets:** For the safety-critical refusal training, a collection of adversarial datasets is employed. These datasets contain prompts designed to elicit harmful, unsafe, or undesirable responses from the model. The data is structured to support a curriculum learning approach, divided into subsets of increasing difficulty (e.g., weak to strong). This allows the model to first learn to refuse simple harmful requests before progressing to more nuanced and complex adversarial attacks. Each entry consists of a prompt that is passed to both the harmful teacher model and the student, where the teacher's `harmful_response` is recorded and used as a negative supervision signal during distillation, guiding the student to refuse or avoid replicating unsafe behavior.
- **Preference Datasets:** The training process begins with beginner-friendly datasets containing simple open-ended tasks and harmless fine-tuning examples such as reasoning, comprehension, and structured response generation. These datasets establish the model's foun-

dational capabilities while ensuring safe behavior. For the alignment phase, preference datasets are employed, where prompts are given to both the student and a helpful teacher model. The teacher's `helpful_response` is treated as the preferred example, while alternative or suboptimal responses serve as rejected examples. This structure is directly compatible with alignment algorithms such as Direct Preference Optimization (DPO), enabling the student to learn from the teacher's safe and beneficial outputs.

Together, these dataset categories form a progressive training pipeline, beginning with general instruction-following ability and gradually incorporating human preference alignment to achieve reliable, helpful, and harmless behavior.

4.3 System Design and Workflow

The system architecture is composed of four distinct models: a `student_model` that is actively being trained; a `harmful_teacher` model to provide target distributions for unsafe responses; a `helpful_teacher` model for generating preferred responses (or as a source for preference data); and a frozen `reference_model`, which is an initial copy of the student model used for KL-divergence regularization.

The workflow is cyclical, iterating through two distinct training phases for a set number of cycles.

In Phase 1, the Immunization & Canary Sensitization Phase, the `student_model` is trained on the adversarial dataset to enhance its refusal capabilities. A key output of this phase is not only an updated model but also the identification of a dynamic set of `canary_parameters`.

In Phase 2, the Helpfulness & Canary Desensitization Phase, the workflow shifts to training on the preference dataset. During this phase, the previously identified `canary_parameters` are explicitly shielded from updates, forcing the model to learn helpfulness by modifying other, non-critical parameters. This cycle repeats, allowing the model to progressively strengthen its safety mechanisms while simultaneously improving its general capabilities.

The proposed workflow, shown in Figure 1, outlines the major components of our system.

4.4 Implementation Details

The implementation relies on several key hyperparameters that must be carefully tuned:

- γ (rejection_strength): A scalar that anneals over time to control the intensity of the refusal loss.
- λ: A coefficient to balance the Negative Probability Objective (NPO) and Reverse KL (RKL) losses in the refusal phase.
- K: The percentage of top parameters selected as canaries based on gradient magnitude.
- α: A coefficient to balance the DPO loss and the KL-divergence loss in the alignment phase.
- β (stabilization_strength): A scalar that controls the strength of the penalty applied to changes in canary parameters during the alignment phase.

The core of the implementation involves gradient manipulation, specifically the calculation of gradients with respect to the refusal loss to identify the canaries, and the subsequent application of a stabilization loss to those specific parameters. An optimizer such as AdamW would be used for weight updates.

4.5 Algorithm and Working

The core of our method is the Enhanced Cyclical Immunization and Alignment Algorithm. The process for each training cycle is detailed below.Our training procedure is detailed in Algorithm 1.

4.5.1 PHASE 1: IMMUNIZATION & CANARY SENSITIZATION

This phase hardens the model against harmful instructions and identifies safety-critical parameters. For each batch of adversarial data:

1. **Simulate Attack (LAT):** A temporary weight perturbation, δ, is computed by taking a step in the direction of the gradient that increases the likelihood of the `harmful_response`. This places the model in a temporarily "hardened" state ($\theta' = \theta + \delta$).
2. **Calculate Refusal Loss:** From this perturbed state, a combined refusal loss, L_{refusal}, is calculated:
$$L_{\text{refusal}} = \gamma \cdot \big((1-\lambda) \cdot L_{\text{NPO}} + \lambda \cdot L_{\text{RKL}}\big)$$
where L_{NPO} penalizes the specific harmful sequence and L_{RKL} pushes the model's response distribution away from that of the `harmful_teacher`.
3. **Identify Canaries:** The gradient of the refusal loss with respect to all model parameters, $\nabla_\theta L_{\text{refusal}}$, is computed. The top $K\%$ of parameters with the highest gradient magnitudes are designated as the `canary_parameters` for this cycle.
4. **Backpropagate and Sensitize:** The refusal loss is backpropagated to update the `student_model`'s weights. This update aggressively modifies the identified canary parameters, making them highly sensitive to the refusal task.

4.5.2 PHASE 2: HELPFULNESS & CANARY DESENSITIZATION

This phase teaches helpfulness while ensuring the stability of the identified canaries. For each batch of preference data:

1. **Calculate Alignment Loss:** A standard alignment loss, $L_{\text{alignment}}$, is calculated using DPO and KL-divergence from the `reference_model`:
$$L_{\text{alignment}} = (1-\alpha) \cdot L_{\text{DPO}} + \alpha \cdot L_{\text{KL}}$$
2. **Calculate Stabilization Loss:** A stabilization loss is computed exclusively for the `canary_parameters` identified in Phase 1. This loss penalizes any deviation from their values at the beginning of the step, typically using Mean Squared Error (MSE):
$$L_{\text{stabilization}} = \text{MSE}(\theta_{\text{canary}}, \theta'_{\text{canary}})$$
where θ'_{canary} represents the initial values of the canary parameters in the current step.
3. **Combine and Backpropagate:** The alignment and stabilization losses are combined into a total loss:
$$L_{\text{total\_alignment}} = L_{\text{alignment}} + \beta \cdot L_{\text{stabilization}}$$
Backpropagating this total loss updates the model's weights. The $\beta \cdot L_{\text{stabilization}}$ term generates opposing gradients for the canary parameters, effectively shielding them from change and forcing the model to learn the alignment task by modifying its other, non-canary parameters.

5 CONCLUSION

REFERENCES

A APPENDIX

Algorithm 1 Enhanced Cyclical Immunization and Alignment

Input: Student model π_θ, reference model π_{ref}, helpful teacher π_{helpful}, harmful teacher π_{harmful}, adversarial data $\mathcal{D}_{\text{adv}}$, preference data $\mathcal{D}_{\text{pref}}$
Parameters: γ (rejection strength), λ (refusal loss balance), K (canary percentage), α (alignment balance), β (stabilization strength)
Initialize: π_θ with pre-trained weights, $\theta_{\text{canary}} \leftarrow \emptyset$
for each training cycle **do**
  {Phase 1: Immunization & Canary Sensitization (Refusal Training)}
  for each batch (x_h, y_h) in $\mathcal{D}_{\text{adv}}$ **do**
    Compute perturbation δ via LAT on harmful response y_h
    Harden model: $\theta' \leftarrow \theta + \delta$
    Calculate refusal loss L_{refusal} from θ' using NPO and RKL
    Compute gradients $\nabla_\theta L_{\text{refusal}}$
    Identify top $K\%$ parameters as θ_{canary} based on gradient magnitude
    Update θ using $\nabla_\theta L_{\text{refusal}}$
  end for
  {Phase 2: Helpfulness & Canary Desensitization (Alignment Training)}
  for each batch (x, y_r, y_{rej}) in $\mathcal{D}_{\text{pref}}$ **do**
    Calculate alignment loss $L_{\text{alignment}}$ using DPO
    Calculate stabilization loss $L_{\text{stabilization}}$ on θ_{canary}
    Combine losses: $L_{\text{total}} \leftarrow L_{\text{alignment}} + \beta \cdot L_{\text{stabilization}}$
    Update θ using $\nabla_\theta L_{\text{total}}$
  end for
end for =0

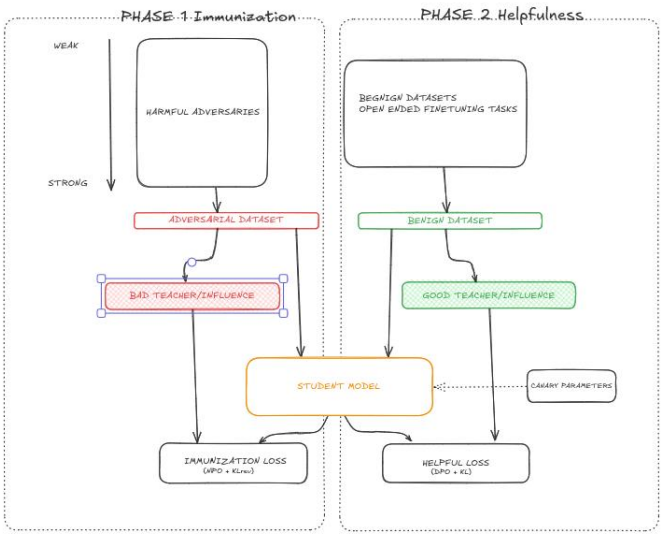

Figure 1: Proposed workflow of the system.