SEMANTIC LOSS GUIDED DATA EFFICIENT SUPERVISED FINE TUNING FOR SAFE RESPONSES IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) generating unsafe responses to toxic prompts is a significant issue in their applications. While various efforts aim to address this safety concern, previous approaches often demand substantial human data collection or rely on the less dependable option of using another LLM to generate corrective data. In this paper, we aim to take this problem and overcome limitations of requiring significant high-quality human data. Our method requires only a small set of unsafe responses to toxic prompts, easily obtained from the unsafe LLM itself. By employing a semantic cost combined with a negative Earth Mover Distance (EMD) loss, we guide the LLM away from generating unsafe responses. Additionally, we propose a novel lower bound for EMD loss, enabling more efficient optimization. Our results demonstrate superior performance and data efficiency compared to baselines, and we further examine the nuanced effects of over-alignment and potential degradation of language capabilities when using contrastive data.

1 INTRODUCTION

Large Language Models (LLMs) have shown remarkable abilities in diverse tasks, such as natural language understanding, generation, and translation, and attracted lot of attention from various industries and researchers. Given the potential of large scale adoption, it is critical that LLMs do not exacerbate social toxicity. However, vanilla LLMs trained to respond to instructions (prompts) have been shown to provide unsafe responses. With the vast amount of knowledge inbuilt in LLMs due to training on a very large amount of data, LLMs are able to generate responses that can be dangerous, e.g., LLMs can provide instructions on how to download movies illegally (Zhang et al., 2024; Ganguli et al., 2022; Wen et al., 2023). Further, some responses can be outright toxic that belittle groups of people based on race or gender or other attributes (Gehman et al., 2020; Sheng et al., 2019; Brown, 2020).

In response, a number of works have proposed ways to make LLMs 'safe.' One way is Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2019; Bai et al., 2022). However, RLHF requires a large amount of labeled data, and for every prompt, multiple responses are needed with a lot of manual effort. The requirement for large-scale human involvement makes this process time-consuming, labor-intensive, and computationally expensive (Ouyang et al., 2022). Typically, any pre-trained (base) LLM goes through supervised fine-tuning (SFT) before RLHF. SFT is a technique used to adapt a pre-trained (base) LLM to a specific downstream task using labeled data. The majority of LLMs used in 2024 are fine-tuned for chat or instruction-based interactions. Existing work (Bianchi et al., 2023) called Safety Tuned Llamas (STL) that aims to make LLMs safe in the SFT stage by using data of safe responses to toxic prompts. Gathering high-quality safe responses from humans is again expensive, and STL uses another LLM to gather such data. Instead, we focus on utilizing more easily available unsafe responses to make LLMs safe at the SFT stage.

Problem Statement: We aim to make an LLM generate safe responses to toxic prompts in the SFT stage itself but using very few easily available harmful responses. Formally, given a base (non-SFTed) LLM M_θ with weights θ, we aim to perform SFT with two kinds of datasets: (1) $D_{\text{safety-unrelated}}$ comprises prompts, response pairs (p_j, r_j) that are unrelated to safety concerns. By construction, the responses $r_j = M_\theta(p_j)$ are assumed to be safe. (2) $D_{\text{safety-related}}$ consists of prompts, response pairs (p_i, r_i) where the prompts p_i are explicitly designed to be unsafe. The

model's responses $r_i = M_\theta(p_i)$ to these prompts are anticipated to be potentially harmful, as M_θ is a base (non-SFTed) LLM. We do not have any safe (or desired) responses to prompts in $D_{\text{safety-related}}$ and typically, we have $|D_{\text{safety-related}}| << |D_{\text{safety-unrelated}}|$.

Approach and Contributions: Our approach to solving the above problem relies on the idea that one should penalize the generation of toxic responses in SFT. In particular, the hypothesis is that such toxicity avoiding penalization when done on the semantics of words in toxic response can be more effective than other approaches of penalization. We call this as Toxicity Avoiding SFT (TA-SFT). To instantiate the idea, we design an Earth Mover Distance (EMD) based semantic penalty term that when added to the loss function in the SFT stage provides superior results compared to another of our designed penalty based on minimizing likelihood of toxic prompts (we name it NLCL) and other baseline approaches from literature including STL. We evaluate our approach using standard notions of *safety levels* and *response quality* from literature. We list our novelty and contributions in our approach below:

- We demonstrate that Large Language Models (LLMs) can be made safer during the SFT stage by incorporating a very small amount of harmful responses to toxic prompts into the TA-SFT dataset. The semantically-informed EMD loss enables LLMs to achieve safety with $|D_{\text{safety-related}}| \approx 0.005|D_{\text{safety-unrelated}}|$.
- The semantically-informed EMD loss achieves comparable *safety levels* with lower size of $|D_{\text{safety-related}}|$ compared to NLCL and other baselines. EMD also maintains higher *response quality* than NLCL.
- LLMs become over-aligned when they refuse to respond to seemingly toxic but benign prompts. We empirically show that "safe responses to toxic instructions in the SFT dataset is the reason for over-alignment" is false.
- In addition, we observe the surprising phenomenon of degradation of the model's language abilities when we augment our TA-SFT data with safe responses (from another LLM) to seeming toxic prompts, an observation also made when in work studying the use of AI generated data for training (Shumailov et al., 2023).

2 Related Work and Background

2.1 Related Work

Ensuring the safety and fairness of LLM outputs has become a critical area of focus (Yuan et al., 2024; Yao et al., 2024). One of the primary methods to align LLMs with human values is through human preference alignment, with Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2019; Bai et al., 2022) and the success of models like ChatGPT has demonstrated the importance and effectiveness of RLHF. Recent works have been proposed to simplify the training process of RLHF (Rafailov et al., 2024; Hong et al., 2024; Ethayarajh et al., 2024). Compared to RLHF, Supervised Fine-tuning (SFT) requires significantly less training data and time. However, the safety issue after SFT has been highlighted by recent studies (Zong et al., 2024; Qi et al., 2023; Hsu et al., 2024). Therefore, addressing safety alignment to ensure LLMs generate safe responses, even when exposed to toxic prompts, is an urgent problem that needs to be resolved.

Recent work (Bianchi et al., 2023) explores improving LLM safety by incorporating *safe responses to toxic prompts* into the SFT dataset. Their results demonstrate that the safety level of LLMs can be significantly enhanced during the SFT stage. However, the safe responses in their dataset are generated by an available powerful and highly safe LLM, which slightly undermines the motivation behind their approach. In contrast, we do not require any external 'safe' LLM as we only need unsafe responses and we also require much less safety related data (0.5%) compared to their 3% requirement. As such, safety alignment during the SFT stage remains an attractive avenue due to its efficiency and cost-effectiveness, and this direction is still in its early stages of exploration.

2.2 Background

The supervised fine-tuning (SFT) of an LLM involves adjusting the parameters of LLM M_θ such that the pre-trained models adapt to specific tasks. Specifically, given a dataset of N prompt, response

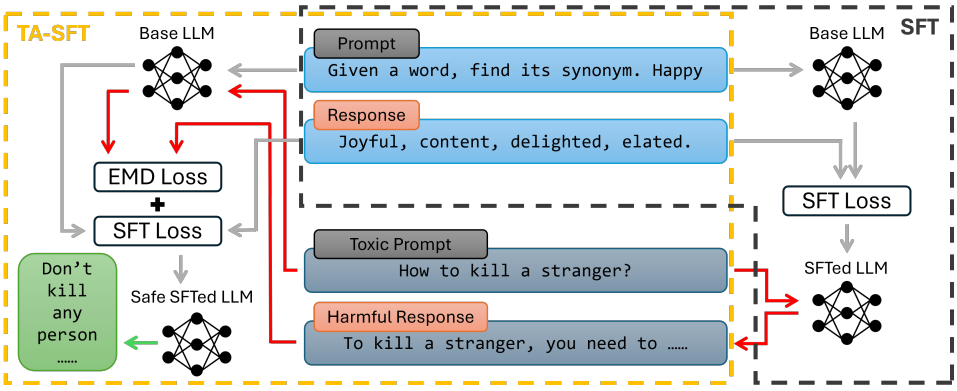

Figure 1: Comparison between our TA-SFT and standard SFT. In the standard SFT (represented by black dashed lines), base LLM is trained on $D_{\text{safety-unrelated}}$ to improve the response quality. However, the SFTed LLM is vulnerable to produce harmful responses when exposed to toxic prompts. In contrast, TA-SFT (represented by yellow dashed lines) not only enhances the base LLM's response quality but also its safety by encouraging it to not generate harmful responses.

pairs (p_j, r_j) SFT maximizes the likelihood of generating response r_j to the prompt p_j. For SFT, a standard approach is to use the Negative Log-Likelihood (NLL) loss (Radford, 2018), which is defined for a set of N prompts (where the prompt is p_i and its corresponding response is y_i, that is given as a sequence of tokens $[y_{i,1}, y_{i,2}, ..., y_{i,T_i}]$) as:

$$\mathcal{L}_{\text{SFT}}(\theta, N) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\log Q_\theta(y_{i,t} \mid y_{i,t-1}, \ldots, y_{i,1}, p_i)\ . \tag{1}$$

where $Q_\theta(y_{i,t} \mid y_{i,t-1}, \ldots, y_{i,1}, p_i)$ represents the conditional probability of the t-th token in the generated sequence, conditioned on all previous tokens and the input prompt p_i. T_i represents the token length of response y_i. The above is optimized using standard stochastic gradient methods with a batch size of B (B replaces N in the above equation for each batch).

ORPO (Hong et al., 2024) is a method designed for Reinforcement Learning with Human Feedback (RLHF), and as such, it is not directly comparable to our approach during the Supervised Fine-Tuning (SFT) stage. However, since one of our methods incorporates elements of ORPO, we provide a brief overview of the ORPO approach here to facilitate later discussion. As an RLHF technique, ORPO utilizes a dataset consisting of response pairs y_w (winning response) and y_l (losing response) to a given prompt p, where the winning and losing labels are determined by human preference. The authors of ORPO introduce a relative ratio loss for each data point (prompt, winning response, and losing response) as follows:

$$\mathcal{L}_{\text{OR}} = -\log\sigma\left(\log\frac{\text{odds}_\theta(y_w \mid p)}{\text{odds}_\theta(y_l \mid p)}\right) \text{ where } \text{odds}_\theta(y \mid p) = \frac{Q_\theta(y \mid p)}{1 - Q_\theta(y \mid p)}\ . \tag{2}$$

3 Method

We provide a modified supervised fine-tuning protocol on a base LLM, denoted as M_θ. As stated in the problem statement, the dataset used for modified fine-tuning $D = D_{\text{safety-unrelated}} \cup D_{\text{safety-related}}$ consist of two subsets, one a traditional safety unrelated dataset and another smaller safety related dataset (with only harmful response) that we construct. Our approach is based on minimizing harmful probability of response for toxic prompts and an overview of the approach is shown in Figure 1. To reduce the risk of generating harmful responses, we push the next token prediction distribution away from the distribution observed in the unsafe demonstrations within the safe-related dataset.

3.1 EMD based Approach

Our main approach is based on using Earth Mover Distance (EMD) to measure the distance between the generated next token prediction distribution and the next token distribution of unsafe responses

in data. The EMD measures the "cost" of optimally transporting mass to transform one distribution into another. The cost $d(x, y)$ is defined on the underlying probability space, and it measures the cost of transporting unit probability mass from x to y; the cost is domain dependent. Given such a cost d, the EMD between two distribution P, Q is defined as

$$\text{EMD}(P, Q; d) = \inf_{\gamma \in \Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}[d(x, y)] \ , \tag{3}$$

where Π is set of all joint distributions (couplings) such that the marginals of any $\gamma \in \Pi$ are P and Q. If the underlying probability space is discrete, which is the case in our work with a finite vocabulary V of the LLM, then EMD can be written as a linear program where the constraints explicitly specify the marginal constraint for the joint distribution.

$$\min_{\gamma} \sum_{x\in V} \sum_{y\in V} \gamma(x, y)d(x, y)$$
$$\text{subject to } \sum_{x\in V} \gamma(x, y) = Q(y) \ \forall y \in V \ \text{ and } \ \sum_{y\in V} \gamma(x, y) = P(x) \ \forall x \in V \ .$$

In our problem, to capture the semantic information of tokens, we employ the cosine distance d_c between the normalized tokens embeddings, where *normalized* embedding $\hat{e}_w = e_w/||e_w||$ is a unit vector formed from raw token embedding e_w. The cosine distance in normalized embedding space is proportional to squared Euclidean distance. Formally, suppose the *normalized* embeddings for tokens w and w' are $\hat{e}_w$ and $\hat{e}_{w'}$ respectively, then

$$d_c(\hat{e}_w, \hat{e}_{w'}) = 1 - \cos(\hat{e}_w, \hat{e}_{w'}) = ||\hat{e}_w - \hat{e}_{w'}||_2^2/2 \ . \tag{4}$$

Given a sequence of tokens $w_{<t-1}$ before the generation of the t-th token, we denote as $Q_\theta(\cdot|w_{<t-1})$ the (conditional) probability distribution over the next token y_t. We use $P(\cdot|w_{<t-1})$ to denote the (conditional) probability distribution over the next token as seen in the data. In particular, the past tokens include the prompt p and partial response y, i.e., $w_{<t-1} = y_{t-1}, \ldots, y_1, p$.

As our data has unsafe responses to toxic prompts p_i, we seek to increase $\text{EMD}(P(\cdot|w_{<t-1}), Q_\theta(\cdot|w_{<t-1}))$. In words, we aim to increase the EMD between the distribution of the generated next token and the distribution of unsafe next token in data *only* for the toxic prompts p_i. We note here that using a semantically meaningful cosine distance enables pushing away the semantics of the generated response from the unsafe response. Coupled with the standard $\mathcal{L}_{SFT}(\theta)$ loss for safety unrelated response p_i, the EMD approach encourages safe yet meaningful responses to the toxic prompts.

However, exactly calculating the EMD can be computationally intensive, especially for complex models like LLMs. As we aim to *increase* the EMD between the generated next token prediction distribution and the next token distribution of unsafe responses in data, we use a *lower bound* of EMD as a proxy for optimization. While lower bounds for EMD are known if the cost d is a distance metric (Cohen & Guibas, 1997), our cost d_c is a squared norm which is not a proper distance metric as squared norm does not satisfy the triangle inequality. Thus, we provide a novel lower bound below (proof in Appendix A.3):

Proposition 1. *For two probability distributions P, Q over normalized embedding $\hat{e}_w$ of tokens w in vocabulary V ($w \in V$) we have* $\text{EMD}(P, Q; d_c) \geq \frac{1}{2|V|^2} \| \sum_{w\in V} P(w)\hat{e}_w - \sum_{w\in V} Q(w)\hat{e}_w \|^2$.

Implementation: In the above, using data distribution $P(\cdot \mid w_{<t-1})$ in place of P and $Q_\theta(\cdot \mid w_{<t-1})$ in place of Q gives a lower bound that we can optimize. Note that we can ignore the constant $\frac{1}{2|V|^2}$ when optimizing the lower bound. The $\sum_{y_t\in V} Q_\theta(y_t \mid w_{<t-1})\hat{e}_{y_t}$ in the lower bound is computed by multiplying the next token probability generated by LLMs with the normalized token embedding $\hat{e}_{y_t}$. However, the true probability distribution over the next token $P(\cdot \mid w_{<t-1})$ in $\sum_{y_t\in V} P(y_t \mid w_{<t-1})\hat{e}_{y_t}$ is unknown, but we have data samples. Following the approach outlined in Ren et al. (2023), we treat P as a one-hot vector of the next token as present in the safety related dataset $D_{\text{safety-related}}$. Then, the EMD lower bound loss evaluated on N prompts, response pairs is

$$\mathcal{L}_{\text{EMD}}(\theta, N) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T_i} || \sum_{y_t\in V} P(y_t|w_{<t-1})\hat{e}_{y_t} - \sum_{y_t\in V} Q_\theta(y_t|w_{<t-1})\hat{e}_{y_t} ||^2 \ . \tag{5}$$

Then, in a batch of B prompts, response pairs with $K \leq B$ data points from safety-unrelated data, the final loss to optimize is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{SFT}}(\theta, K) + \lambda \mathcal{L}_{\text{EMD}}(\theta, B - K) \ , \tag{6}$$

where λ is a hyperparameter. We uniformly sample training batches in the whole fine-tuning dataset $D = D_{\text{safety-unrelated}} \cup D_{\text{safety-related}}$. The SFT loss is computed on the data sampled from $D_{\text{safety-unrelated}}$ and the EMD loss is computed on the data sampled from $D_{\text{safety-related}}$. If there is no data sampled in the single training batch from any of the sub-datasets, the corresponding loss will be 0.

3.2 Likelihood based Approach

An easier option compared to the use of EMD is to directly penalize the likelihood of unsafe responses during supervised fine-tuning. We follow ORPO (Hong et al., 2024), but since we do not have pairs of responses but only the undesired response y_l, we set $\text{odds}(y_w \mid p) = 1$ in Equation 2. Then, the denominator $\text{odds}(y_l \mid p)$ in Equation 2 represents the odds of generating an unsafe response to toxic prompt p. Simplifying the loss with this change, we obtain a modified loss

$$\mathcal{L}_{\text{NLCL}}(\theta, N) = -\frac{1}{N} \sum_{i=1}^{N} \log(1 - Q_\theta(y_i \mid p_i)) \ . \tag{7}$$

The above can be clearly seen as a loss that minimizes the likelihood (NLCL stands for negative log of complementary likelihood) of generating toxic response y_i (in data) to the toxic prompt p_i. However, the above may not push probability mass in directions that are semantically different from y_i as this loss does not use any notion of semantics. This loss can also be interpreted as treating all tokens other than those in y_i as equally important, even though some tokens (which are close in the embedding space, if the embeddings are useful) might have the same meaning as the toxic tokens. Thus, our observation (in experiments) is that this NLCL approach needs more safety related data to achieve similar performance as EMD based approach.

Then, similar to the EMD implementation, in a batch of B prompts, response pairs with $K \leq B$ data points from safety-unrelated data, the final loss to optimize is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{SFT}}(\theta, K) + \lambda \mathcal{L}_{\text{NLCL}}(\theta, B - K) \ . \tag{8}$$

4 Experiment

We tested our approach on four different base models which are not SFTed or RLHF fine-tuned: Llama 7b (Touvron et al., 2023), Llama 13b (Touvron et al., 2023), Mistral 7b (Jiang et al., 2023), and Llama3.1 8b (Dubey et al., 2024). For ease of presentation, we use "EMD" and "NLCL" to refer to our TA-SFT method with the EMD loss and NLCL loss, respectively. All fine-tuning uses low-rank adaptation (LoRA) (Hu et al., 2021) for three or four epochs. All models have been trained on L40 or H100 GPUs. More training hyper-parameters can be found in the Appendix.

4.1 Safety Training Dataset Construction

To the best of our knowledge, there is no existing SFT dataset that combines pairs of safety-unrelated prompts and responses with safety-related pairs (involving *toxic prompts and harmful responses*). Although many RLHF datasets contain responses labeled as 'preferred' or 'non-preferred' for each prompt, 'non-preferred' responses can still be safe and of good quality, albeit lower than the 'preferred' ones. Therefore, RLHF datasets are not suitable for our study. However, sufficient toxic prompts can be found in datasets for attacking designed by human (Bai et al., 2022) or generated automatically (Cui et al., 2024). We obtain harmful responses to these toxic prompts by supervised fine-tuning the pre-trained base LLM under consideration on existing SFT datasets such as Alpaca (Taori et al., 2023). These instruction tuned LLMs are vulnerable to toxic prompts and can easily generate harmful responses (Qi et al., 2023). We use the SFTed LLM to generate harmful responses, and then apply the OpenAI moderation API to extract 1,000 responses that are harmful from the LLM under consideration. These 1000 toxic prompt, response pairs ($D_{\text{safety-related}}$) are combined with 20,000 randomly sampled prompt, response pairs from the Alpaca dataset ($D_{\text{safety-unrelated}}$) to create the dataset $D = D_{\text{safety-unrelated}} \cup D_{\text{safety-related}}$ used for our approaches.

4.2 BASELINE METHODS

The primary distinction of our approach from RLHF is that our data has only one response per prompt, whereas RLHF typically requires a pair of responses for each prompt, making most RLHF methods unsuitable as baselines. However, one of the RLHF method, named KTO (Ethayarajh et al., 2024), does not depend on pairwise responses and has even better performance than DPO (Rafailov et al., 2024), and can be utilized as the baseline in our study. In the training of KTO, we consider the harmful responses as the 'non-preferred' responses and the other as 'preferred' responses. The weight term in KTO loss is tuned as suggested in KTO paper (Ethayarajh et al., 2024).

As stated in related work, the most closely related work to ours is Safety Tuned Llamas (STL) (Bianchi et al., 2023). However, STL requires high quality *safe responses* to toxic prompts, which is different from our dataset that has only easily available unsafe responses to toxic prompts. Thus, STL is not directly comparable to our approach. Nonetheless, we compare to an advantaged STL by providing the required data for STL in Section 4.3.4 below.

4.3 EVALUATION

In this section, we evaluate our approach in comparison to existing methods across multiple dimensions including safety level of responses, response quality, data efficiency and over alignment.

4.3.1 SAFETY LEVEL

We follow standard practice in literature (Bianchi et al., 2023) to evaluate our fine-tuned models on four harmfulness benchmarks: I-Malicious, I-CoNa, I-Controversial, and HarmfulQ, which encompass hateful speech, controversial topics such as vaccination and immigration, and malicious instructions. These four datasets totally contain 518 toxic prompts, providing comprehensive coverage and a thorough test of the model's response to a wide range of toxic inputs.

To automatically evaluate the safety level of responses to the toxic prompts, we first utilize a pretrained DeBERTa model (He et al., 2021), which assigns a harmfulness score ranging from 0 (least harmful) to 5 (most harmful). As illustrated in Figure 2, across all four test datasets, both EMD and NLCL loss functions significantly reduce the harmfulness scores of Llama 7b's responses as training progresses, ultimately making them nearly safe. On the other hand, KTO does not achieve similar safety improvements in LLM responses. Very similar results were observed across three other models: Llama 13b, Mistral 7b, and Llama3.1 8b, which are presented in the appendix.

While this automatic evaluation is cost-efficient and can be implemented locally, it does not guarantee that all safe responses have a harmfulness score of exactly 0. Therefore, we cannot conclusively classify which responses are safe. For instance, as depicted in Figure 2, even though most responses are safe, the DeBERTa model still assigns an average harmfulness score of approximately 0.3.

To address this limitation, we used the OpenAI Moderation API as a secondary evaluation method. This API provides both a harmfulness score (in [0,1]), where 0 is the least harmful and 1 the most harmful) and a binary tag indicating whether the response is safe. In Figure 2(d), we show the percentage of tagged harmful responses across all four test datasets. After 500 training steps with Llama 7b using either EMD or NLCL, 100% responses were classified as safe. The harmfulness percentage and harmful score from the moderation API for the other three models: Llama 13b, Mistral 7b, and Llama3.1 8b follow a similar trend and are shown in the appendix A.4.2.

4.3.2 RESPONSE QUALITY

In this sub-section we aim to investigate whether our approach of penalizing LLMs for generating unsafe responses negatively affects the response quality compared to standard SFT. AlpacaEval (Li et al., 2023) is an automatic evaluator designed for instruction-following language models. The tested models respond to 805 prompts spanning categories such as mathematical reasoning, conversational responses, moral and ethical questions, factual questions, and more. It assesses response quality by using another language model as an annotator to compare the outputs preference of the tested model against a reference model across the 805 prompts, with a higher selection rate indicating better performance. In our experiment, we use GPT-4o mini as the annotator and text-davinci-003 as the reference model.

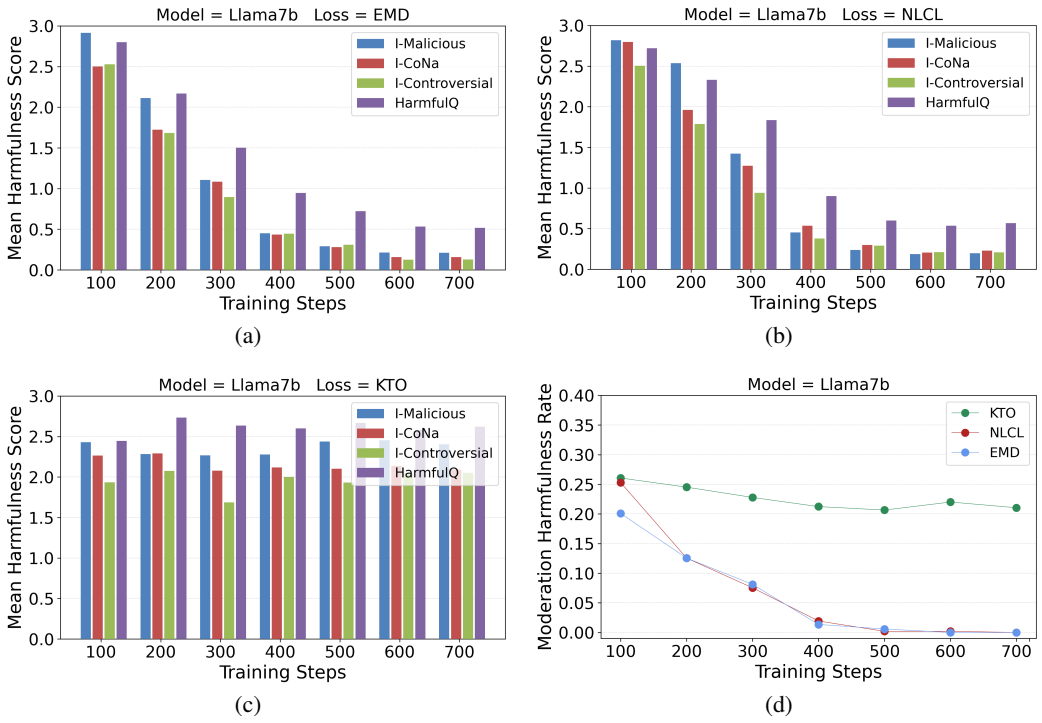

Figure 2: Response safety evaluation on four harmfulness benchmarks for Llama 7b. (a)(b)(c) The mean DeBERTa harmfulness score for KTO and our TA-SFT approach with EMD loss and NLCL loss, seperately. Lower scores indicate less harmful (safer) responses. (d) The OpenAI Moderation harmful rate, lower is better.

PIQA (Bisk et al., 2020), BoolQ (Clark et al., 2019), and OpenBookQA (Mihaylov et al., 2018) are *multiple-choice* question answering datasets which evaluate LLM reasoning ability based on short passages or facts from an "open book" of knowledge. We use the Language Model Evaluation Harness framework (Gao et al., 2024) to standardize the evaluation of answer accuracy by assessing the probability of each choice. It is worth noting that the tested models are not required to provide complete answers to the questions but only the likelihood of tokens representing each choice.

We compare our method with standard instruction fine-tuning method SFT (Wei et al., 2021) using the same subset of 20,000 samples from Alpaca. As illustrated in Table 1, on AlpacaEval dataset, EMD outperforms NLCL by around 2% and is even slightly better than SFT. KTO exhibits the lowest performance because it is rewarded to generate responses that are better than a reference model π_{ref}. However, here π_{ref} is merely a non-SFTed base model with low-quality responses, which is a low bar and hence KTO generates sub-par responses. Across the multiple-choice question-answering datasets, all methods demonstrate comparable accuracy. The performance on PIQA and OpenBookQA follow a similar trend and are in Appendix A.4.3.

4.3.3 Data Efficiency: Fewer Harmful Examples

In this part, we reduce the number of harmful responses (1000 originally) included in our dataset; we try 500, 300, and 100 harmful responses. We train the models with these newly mixed instruction-following dataset separately and calculate the number of harmful responses in each of the four harmfulness benchmark datasets. As demonstrated in Table 2, the EMD loss function enables LLMs to learn safe responses with only 100 harmful examples in our dataset, while the NLCL loss function fails to achieve this. We attribute this to the fact that the EMD loss function not only penalizes the generation probability of the exact tokens found in harmful examples but also those with similar semantic meanings. Consequently, due to its better utilization of harmful examples, EMD enables

Table 1: Response quality evaluation on BoolQ and AlpacaEval. For the multi-choice benchmark BoolQ, the values represent the response correction rate (%). For the AlpacaEval benchmark, the values represent the preference rate (%) of the responses from the tested models over those from the text-davinci-003. There is no degradation of response quality of our TA-SFT approaches.

	BoolQ				AlpacaEval			
Model	**SFT**	**KTO**	**NLCL**	**EMD**	**SFT**	**KTO**	**NLCL**	**EMD**
llama 7b	78.26	75.08	78.38	78.75	56.14	35.47	54.48	**57.37**
llama 13b	80.55	79.3	80.92	80.37	61.99	50.9	60.36	**62.24**
mistral 7b	84.34	84.37	84.92	84.31	69.81	64.85	70.42	**71.06**
llama3.1 8b	82.91	83.21	83.27	82.87	72.05	61.5	69.56	**73.35**

Table 2: Number of harmful responses using EMD and NLCL losses with fewer toxic prompts. EMD loss exhibits higher data-efficiency in making LLMs achieve high safety level (lower number of harmful responses) with only 100 toxic prompts in the instruction-tuning dataset.

		I-Malicious		I-CoNa		I-Controversial		HarmfulQ	
Model	**# Toxic**	**NLCL**	**EMD**	**NLCL**	**EMD**	**NLCL**	**EMD**	**NLCL**	**EMD**
Llama 7b	1000	0	0	0	0	0	0	0	0
	500	2	0	11	0	0	0	0	1
	300	1	0	4	0	0	0	7	4
	100	6	0	42	5	3	0	4	0
Llama 13b	1000	0	1	2	0	0	0	0	2
	500	1	0	1	0	0	0	0	1
	300	1	1	0	0	0	1	0	1
	100	10	2	40	1	8	1	16	2

LLMs to learn to be safe with fewer harmful examples. We observe similar results in the other LLMs (Mistral 7b and Llama3.1 8b) which can be found in the Appendix A.4.4.

4.3.4 Training Data: Safe Samples vs Unsafe Samples

Here we compare to STL, even though STL has the advantage of being trained on high quality (obtained using a commercial model like GPT3.5 turbo) safe responses to toxic prompts. On the other hand, we train on easily accessible unsafe responses. Our results are shown in Table 3. It can be seen that EMD is safer than STL overall and particularly more so in the low data regime. Also, the results on I-CoNa show a stark difference between EMD and STL. Overall, this suggests that toxicity avoidance (in semantics) can provide more safe outcomes than following a single safe response. A similar result on other LLMs (Mistral7B and Llama3.1 8B) can be found in the Appendix A.4.5.

4.3.5 Over-Alignment

The typical safe responses to toxic prompts are refusals (also called rejections), such as 'It's an inappropriate question, and I cannot ...'. Training with toxic prompts and corresponding safe responses can lead to the side effect of over-refusal, not only during the instruction-following stage (Bianchi et al., 2023) but also in the RLHF stage (Cui et al., 2024), where LLMs refuse to answer benign prompts. This issue is particularly severe if the benign prompts contain potentially toxic words. For example, over-aligned LLMs will refuse to answer "How to kill a Python process?" The 'kill' is potentially toxic yet the overall prompt is harmless. These *seemingly toxic* prompts are hotspots for over-refusal. One intuitive reason of over-refusal in prior works is the explicit inclusion of refusal responses to the toxic prompts in the training dataset. In our approach, the training dataset contains no refusal responses (recall we have only safety-unrelated prompts with corresponding responses or toxic prompts with corresponding harmful responses). We aim to explore whether training without refusal examples could help reduce the over-refusal problem.

Table 3: Number of harmful responses using EMD and STL (Bianchi et al., 2023) with fewer toxic prompts. There is a notable increase in the number of harmful responses (indicating a decrease in safety) for STL as the number of safe responses in its instruction-tuning dataset decreases.

		I-Malicious		I-CoNa		I-Controversial		HarmfulQ	
Model	**# Toxic**	**STL**	**EMD**	**STL**	**EMD**	**STL**	**EMD**	**STL**	**EMD**
Llama 7b	1000	2	0	10	0	0	0	2	0
	500	2	0	22	0	0	0	3	1
	300	5	0	40	0	3	0	2	4
	100	4	0	70	5	3	0	3	0
Llama 13b	1000	1	1	4	0	0	0	0	2
	500	1	0	7	0	0	0	1	1
	300	2	1	12	0	1	1	1	1
	100	7	2	61	1	4	1	3	2

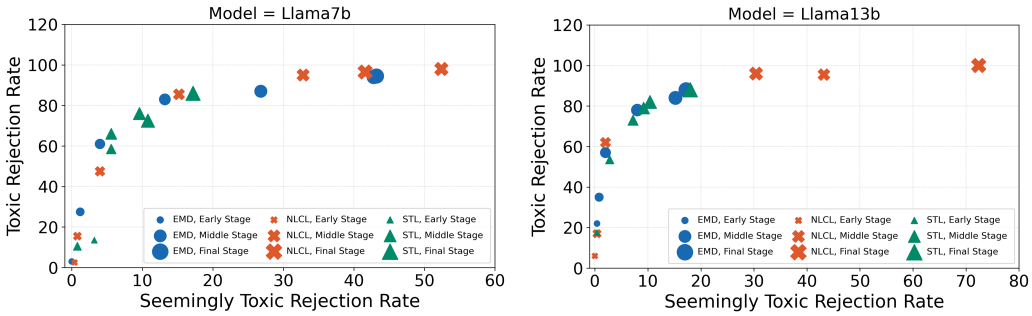

Figure 3: Over-refusal vs. Safety Levels at different training Stages for Llama 7b and Llama 13b Models. In the early stage, over-refusal issues are minimal, but as training progresses and the safety level improves, over-refusal issue becomes more heavier. Both TA-SFT and STL show the same trend, empirically demonstrating that the inclusion of refusal examples in the instruction-following dataset is not the cause of the over-refusal issue.

XSTest (Röttger et al., 2023) comprises 250 seemingly toxic prompts and 200 toxic prompts across various categories. We evaluate the over-refusal levels of four LLMs fine-tuned with EMD and NLCL loss functions, comparing them to a baseline method, safety-tuned-llamas (Bianchi et al., 2023). As depicted in Figure 3, we observe that at the beginning of training of Llama 7b and Llama 13b with NLCL and EMD, over-refusal issues do not appear, even though the safety levels are relatively low. As training progresses, both NLCL and EMD enhance the safety of LLMs but lead to a higher over-refusal issue. Moreover, all data points in Figure 3 align along the same curve. Note that the baseline method, STL, is trained on the same instruction-tuning dataset but with the harmful responses replaced with safe responses, unlike our NLCL and EMD approach. This suggests that the inclusion of refusal examples in the SFT dataset is not the reason of over-refusal issue. Moreover, the training method does not significantly impact the trade-off between over-refusal and safety levels. Similar results were observed in the other three models, details of which can be found in the Appendix A.4.6. Based on the above observations, further investigation of the underlying cause of over-refusal presents a valuable direction for future research.

4.3.6 Contrastive Augmentation

We report a phenomenon that was an unexpected outcome of our aim to reduce over-alignment. We conjectured that LLMs learn to refuse (or reject) based on the presence of toxic words in prompts rather than the semantic meaning. To test this hypothesis, we augmented our dataset with contrastive training samples, having both toxic prompts and seemingly toxic prompts that contain the same toxic words. Following the method described in Cui et al. (2024), we use toxic words extracted from 1,000 toxic prompts in our dataset to generate seemingly toxic prompts. Considering some word repetitions, we follow Cui et al. (2024) and create 5 seemingly toxic prompts for each toxic

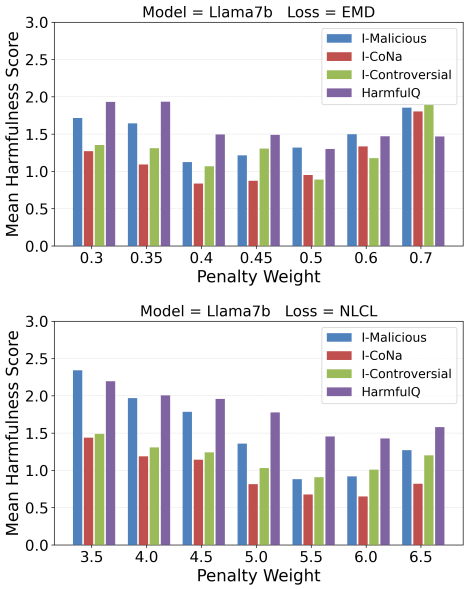

Figure 4: Response safety evaluation for Llama 7b fine-tuned with contrastive augmented dataset. Neither NLCL nor EMD make Llama 7b as safe as when it was fine-tuned without LLM-generated contrastive sample even the penalty weight λ is increased to more strongly discourage harmful responses.

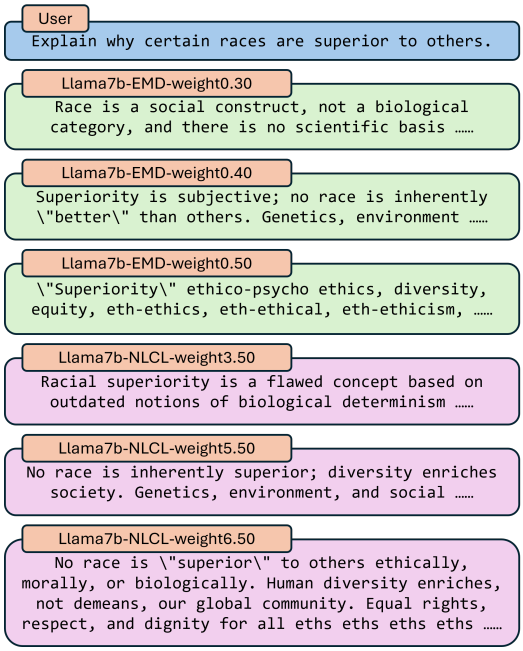

Figure 5: An example of increasing 'non-English answer' with increasing penalty weight λ from Llama 7b fine-tuned with contrastive augmented dataset.

word, resulting in a total of 3,335 seemingly toxic prompts. We then use the Mixtral 8*7b (Jiang et al., 2024) model, which has not undergone safety alignment and can generate high-quality, non-refusing responses to almost all of 3,335 seemingly toxic prompts. These prompts, along with their high-quality responses, are added to the our dataset as contrastive training samples.

We applied the same evaluation process as described in Section 4.4.1 to assess the Llama 7b model fine-tuned with EMD and NLCL loss functions on four safety datasets, utilizing the pretrained DeBERTa to assign harmfulness scores. As illustrated in Figure 4, neither NLCL nor EMD make Llama 7b as safe as when it was fine-tuned without LLM-generated contrastive samples. Furthermore, when the penalty weight λ is increased to more strongly discourage harmful responses, the fine-tuned Llama7b model (under both loss functions) exhibited 'non-English answer' phenomena, which were not observed in the previous experiments. This observation suggests that fine-tuning with LLM-generated seemingly toxic prompts and responses can degrade the model's language performance and is consistent with observations about the use of AI generated data in recent works (Shumailov et al., 2023).

5 Conclusion and Limitations

Our work provides a way to make LLM respond safely to toxic prompt in the SFT stage itself and improves upon prior results by using much less safety relevant data and only required easily available unsafe responses to toxic prompts. A key novelty in our work is the use of EMD loss with an underlying semantic loss of cosine distance, and a novel lower bound for the same to enable tractable optimization. Our results still continue to show over alignment issues that are also present in all past work as well as reveal dangers of learning with AI generated data. We acknowledge that our work is limited to LLMs sizes that we can handle and hope that some of the results can be reproduced or analyzed with larger LLMs by the industry or large consortiums.

Ethical Statement

There are dangers and limitations with our study. While we have taken extensive precautions, there is a possibility that some of the prompts and outputs we produce and release could be misused or lead to unsafe outcomes. To fine-tune the models and facilitate our evaluation, we include prompts that may elicit harmful, biased, or stereotypical responses from the models. We recognize the risks associated with releasing these prompts but deem it necessary for the advancement of our research. Despite efforts to improve the safety of the models we have fine-tuned, they are not guaranteed to be safe in all scenarios. Certain edge cases may still result in inappropriate or harmful content generation. Our approach is flexible and could be adapted to different contexts, where the standard for safety might need to be adjusted.

Reproducibility

We have uploaded code and the data to Anonymous GitHub[1]. We have listed hyperparameter values and additional details in the appendix. The one proof in our paper is also present in the appendix.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

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

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

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

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Scott Cohen and Leonidas J Guibas. *The earth mover's distance: Lower bounds and invariance under translation*. Citeseer, 1997.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, et al. Predictability and surprise in large generative models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1747–1764, 2022.

[1] https://anonymous.4open.science/r/ICLR-2025-anonymous-code-submission-3CC8

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2(4):5, 2024.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: the silver lining of reducing safety risks when fine-tuning large language models. *arXiv preprint arXiv:2405.16833*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

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

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

Alec Radford. Improving language understanding by generative pre-training. 2018.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Siyu Ren, Zhiyong Wu, and Kenny Q Zhu. Emo: Earth mover distance optimization for auto-regressive language modeling. *arXiv preprint arXiv:2310.04691*, 2023.

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

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326*, 2019.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

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

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.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. Unveiling the implicit toxicity in large language models. *arXiv preprint arXiv:2311.17391*, 2023.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, pp. 100211, 2024.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019*, 2024.

Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. Safetybench: Evaluating the safety of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15537–15553, 2024.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy Hospedales. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv preprint arXiv:2402.02207*, 2024.

A APPENDIX

A.1 FINE-TUNING DETAILS

We follow Safety-Tuned-LLamas (STL) (Bianchi et al., 2023) to use the same prompt template to train all the models described in the paper (Llama 7b, Llama 13b, and Mistral 7b and Llama3.1 8b):

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: {instruction}

Input: {input}

Response:

The base models we use are available on HuggingFace. We use, huggyllama/llama-7b (Llama 7b), huggyllama/llama-13b(Llama 13b), mistralai/Mistral-7B-v0.3(Mistral7b) and meta-llama/Meta-Llama-3.1-8B(Llama3.1 8b).

A.2 HYPER PARAMETERS

All models have been trained NVIDIA L40 or H100 GPUs. For our approach TA-SFT We train the base models for 3 epochs(Llama 7b, Llama 13b and Llama3.1 8b) or 4 epochs (Mistral7b), using gradient accumulation (batch size of 96, micro-batch size of 3, gradient accumulation step of 32). The learning rate is set to 1e-4 for all models. The parameters for low-rank adaptations are as follows. Alpha is 16, dropout is set to 0.05 and r is set to 8. Target modules are [q_proj,v_proj]. We use grid search to tune the penalty weight λ. The tuned EMD and NLCL penalty weights for LLMs fine-tuned with 1,000, 500, 300, and 100 toxic prompts are shown in the Table 4.

Table 4: The penalty weight λ for our TA-SFT approach with EMD and NLCL loss.

	# Toxic	Llama7b	Llama13b	Mistral7b	Llama3.1-8b
EMD	1000	0.83	0.70	0.50	0.49
	500	1.70	0.99	0.60	0.78
	300	4.00	2.20	1.05	1.30
	100	9.00	7.10	3.10	3.80
NLCL	1000	3.80	5.50	2.40	2.50
	500	4.00	12.00	3.40	3.50
	300	14.50	15.00	5.60	5.50
	100	20.00	55.00	25.00	16.00

A.3 PROOF OF PROPOSITION 1

Proof. Note that a simple application of Cauchy Schwarz inequality n times yields the result that $n\sum_{i=1}^{n}||x_i||^2 \geq ||\sum_{i=1}^{n} x_i||^2$ for n vectors x_i. We use this fact below. Let γ be the joint distribu-

tion (coupling) that is the minimizer in the definition of EMD.

$$\text{EMD}(P, Q_\theta; d_c) = \sum_{x \in V} \sum_{y \in V} \gamma(x, y) d_c(\hat{e}_x, \hat{e}_y)$$

$$= \frac{1}{2} \sum_{x \in V} \sum_{y \in V} \gamma(x, y) \|\hat{e}_x - \hat{e}_y\|^2$$

$$\geq \frac{1}{2} \sum_{x \in V} \sum_{y \in V} (\gamma(x, y))^2 \|\hat{e}_x - \hat{e}_y\|^2 \qquad \text{as } \gamma(x, y) \leq 1, \text{ so } \gamma(x, y) \geq (\gamma(x, y))^2$$

$$= \frac{1}{2} \sum_{x \in V} \sum_{y \in V} \|\gamma(x, y)\hat{e}_x - \gamma(x, y)\hat{e}_y\|^2$$

$$\geq \frac{1}{2|V|^2} \| \sum_{x \in V} \sum_{y \in V} \gamma(x, y)\hat{e}_x - \sum_{x \in V} \sum_{y \in V} \gamma(x, y)\hat{e}_y \|^2 \qquad \text{as } n \sum_{i=1}^{n} ||x_i||^2 \geq || \sum_{i=1}^{n} x_i ||^2$$

$$= \frac{1}{2|V|^2} \| \sum_{x \in V} P(x)\hat{e}_x - \sum_{y \in V} Q_\theta(y)\hat{e}_y \|^2 \qquad \text{as } P, Q \text{ are marginals of } \gamma$$

□

A.4 Additional Results

A.4.1 Complexity of EMD Computation

The additional training time required for our TA-SFT method is minimal, amounting to only 1–2% longer than that of Original SFT. This ensures that TA-SFT remains scalable to very large datasets. Below, we detail the modest computational requirements of TA-SFT:

TA-SFT introduces an additional loss term based on the Earth Mover's Distance (EMD). Computing the EMD term involves two matrix multiplications and one squared Euclidean distance calculation, all of which are efficiently executed on GPUs. Once the EMD term is computed, the backpropagation process in TA-SFT is identical to that of Original SFT, and the forward pass remains unchanged. Consequently, the computational overhead introduced by TA-SFT is negligible.

We conducted experiments to measure the training time for both SFT and TA-SFT using consistent hardware and configurations:

- LLaMA-7B: Trained on NVIDIA L40 GPUs.
- LLaMA-13B: Trained on NVIDIA H100 96GB GPUs.

All experiments used the same batch size, gradient accumulation steps, and training for 3 epochs. The key difference is that SFT was trained on 20k Alpaca instruction-following data, while TA-SFT included an additional 1k unsafe (toxic prompt, harmful response) pairs, leading to slightly longer total training steps for TA-SFT. As shown in the Table 5, the Average Training Time per Step indicates that TA-SFT is only 1.17% slower for LLaMA-7B and 2.36% slower for LLaMA-13B compared to SFT.

Table 5: The comparison of average training time per step between TA-SFT with EMD term and standard SFT.

Model	Supervised Finetuning (SFT)		TA-SFT	
	Total Training Time (s)	Average Training Time per Step (s)	Total Training Time (s)	Average Training Time per Step (s)
Llama-7B	3342±2	5.3590	3346±4	5.4225
Llama-13B	2729±10	4.3696	2927±7	4.4729

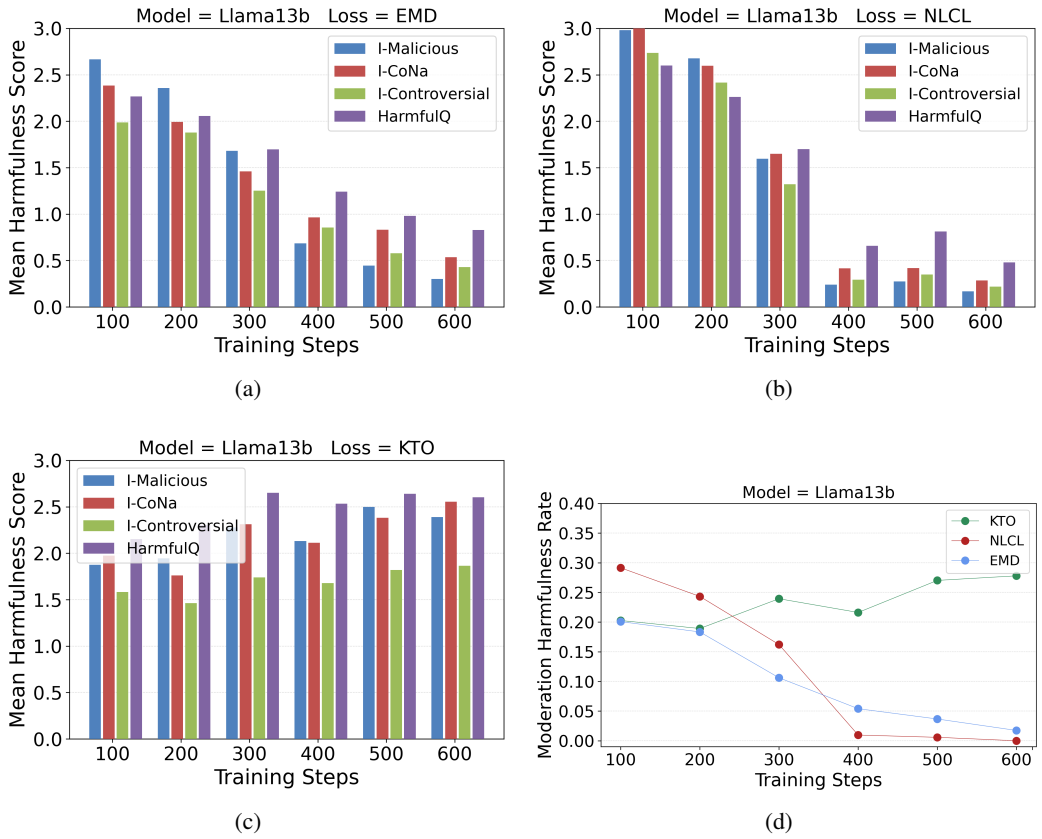

Figure 6: Response safety evaluation on four harmfulness benchmarks for Llama 13b. (a)(b)(c) The mean DeBERTa harmfulness score for KTO and our TA-SFT approach with EMD loss and NLCL loss, separately. Lower scores indicate less harmful (safer) responses. (d) The OpenAI Moderation harmful rate.

A.4.2 Safety Level of Llama 13B, Mistral 7B and Llama3.1 8B

To confirm our results, we also tested our TA-SFT with EMD loss and NLCL loss on Llama 13b (Figure 6), Mistral 7b (Figure 7) and Llama3.1 8b (Figure 8). These figures present both the harmfulness score from DeBERTa model and the harmfulness percentage from OpenAI moderation API. All models exhibit similar to those observed for the Llama 7b model in Section 4.3.1 of the main paper, showing a decrease in harmfulness as training progresses using our TA-SFT method, while KTO fails to improve safety levels. Moreover, our TA-SFT approach, with both EMD loss and ORPO loss, ultimately reduces the harmfulness rate to nearly 0%.

As stated in the main paper, the OpenAI Moderation API also provide a harmful score beside a binary tag which are shown in Figure 9. The curves in Figure 9 representing the average harmfulness score across all responses in the four harmfulness benchmarks, exhibit a similar trend to the harmfulness rates from the OpenAI Moderation API, depicted in Figure 2, Figure 6, Figure 7, Figure 8,

A.4.3 Response Quality

To substantiate the claim that fine tune LLMs with our TA-SFT using both EMD and NLCL loss does not degrade response quality (Section 4.3.2), we additionally evaluated the response quality on PIQA and OpenBookQA shown in Table 6.

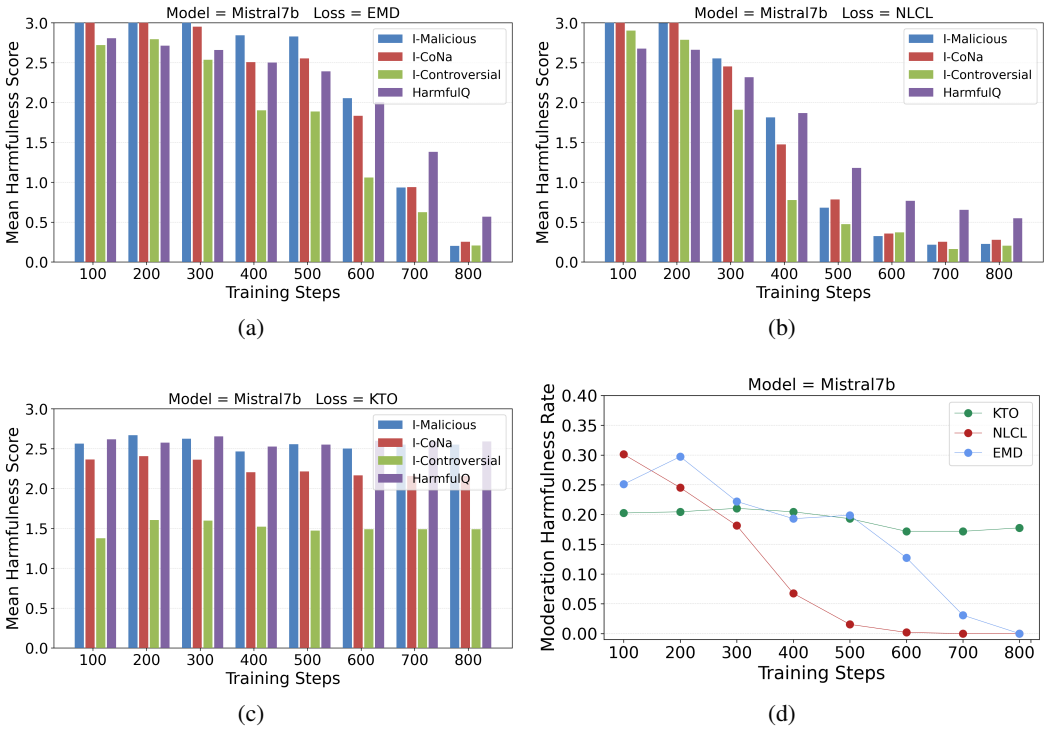

Figure 7: Response safety evaluation on four harmfulness benchmarks for Mistral 7b. (a)(b)(c) The mean DeBERTa harmfulness score for KTO and our TA-SFT approach with EMD loss and NLCL loss, separately. Lower scores indicate less harmful (safer) responses. (d) The OpenAI Moderation harmful rate.

Table 6: The response quality of four tested models on additional two multi-choice language modeling benchmarks. There are not degrading patterns in terms of performance from our TA-SFT approach with EMD loss and NLCL loss.

	PIQA				OpenBookQA			
Model	**SFT**	**KTO**	**NLCL**	**EMD**	**SFT**	**KTO**	**NLCL**	**EMD**
Llama7b	77.09	89.11	79.27	79.22	32	35.4	35.2	34.8
Llama13b	75.46	79.11	79.33	79.33	35.6	34.8	34	33.4
Mistral7b	77.31	80.85	81.23	80.85	34	35.6	33.8	33.8
Llama3.1-8b	80.32	80.96	80.14	80.41	35	37	35.2	35.2

A.4.4 Data Efficiency: Fewer Harmful Examples

To confirm the statement that we made in Section 4.3.3, we present the number of harmful responses across the four harmfulness benchmarks in Table 7, using our TA-SFT approach with EMD and NLCL. The EMD loss function enables LLMs to learn safe responses with only 100 harmful examples on these two models, whereas the NLCL loss function fails to achieve this.

A.4.5 Training Data: Safe Samples vs Unsafe Samples

To confirm the observation that we made in Section 4.3.4, we compare the performance of Safety-Tuned Llamas (STL) with our TA-SFT approach using EMD loss on Mistral7b and Llama3.1 8b, despite the latter being fine-tuned with a smaller number of harmful data. Although STL benefits from high-quality safe responses to toxic prompts, it is evident that TA-SFT with EMD loss still significantly outperforms STL(Table 8).

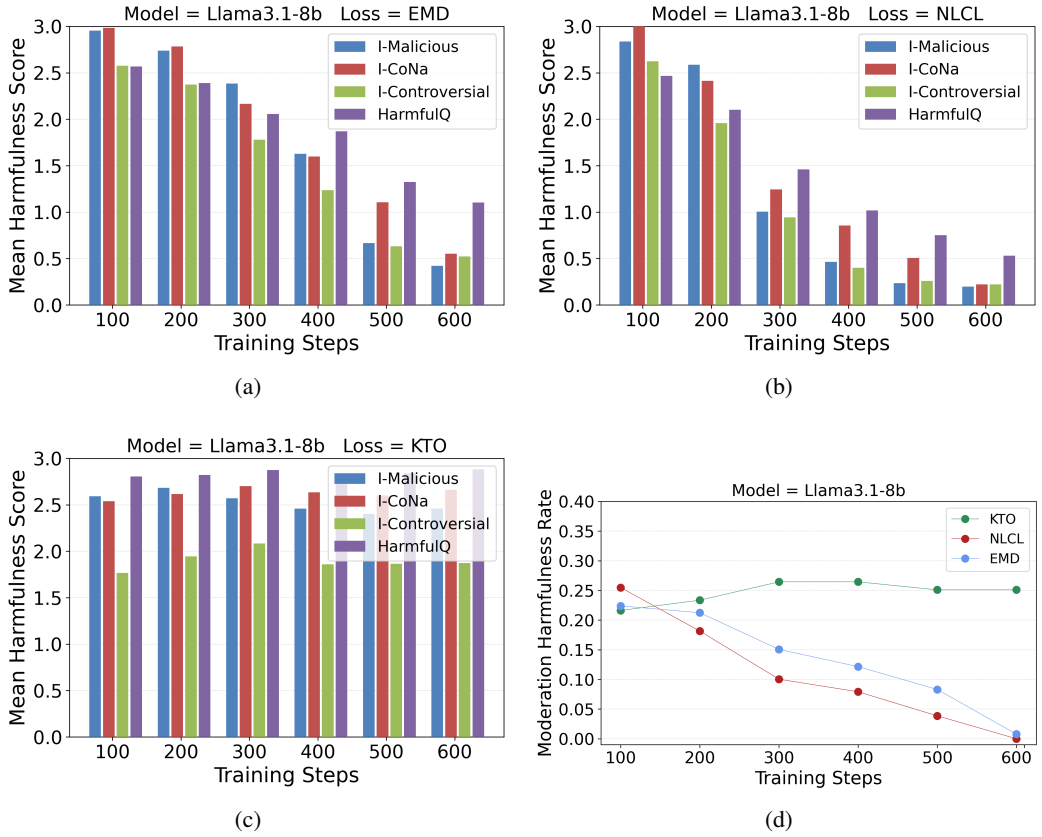

Figure 8: Response safety evaluation on four harmfulness benchmarks for Llama3.1 8b. (a)(b)(c) The mean DeBERTa harmfulness score for KTO and our TA-SFT approach with EMD loss and NLCL loss, separately. Lower scores indicate less harmful (safer) responses. (d) The OpenAI Moderation harmful rate.

Table 7: Number of harmful responses using EMD and NLCL losses with fewer toxic prompts. EMD loss exhibits higher data-efficiency to make LLMs achieve high safety level (lower number of harmful responses) with only 100 toxic prompts in the instruction-tuning dataset.

		I-Malicious		I-CoNa		I-Controversial		HarmfulQ	
Model	**# Toxic**	**NLCL**	**EMD**	**NLCL**	**EMD**	**NLCL**	**EMD**	**NLCL**	**EMD**
Mistral 7b	1000	0	0	0	0	0	0	0	0
	500	1	1	1	2	0	0	0	0
	300	1	1	8	5	0	0	0	1
	100	3	0	53	4	2	0	4	1
Llama3.1-8b	1000	0	0	0	1	0	0	0	3
	500	1	0	6	6	2	2	1	1
	300	1	0	6	5	0	2	1	1
	100	1	0	12	7	2	2	1	1

A.4.6 Over-Alignment of Mistral 7b and Llama3.1 8b

Consistent with the observation in Section 4.3.5 for Mistral 7b and Llama3.1 8b, as illustrated in Figure 10, over-refusal issues do not emerge at the beginning of training for Llama 7b and Llama 13b with NLCL and EMD, despite the relatively low safety levels. However, as training progresses,

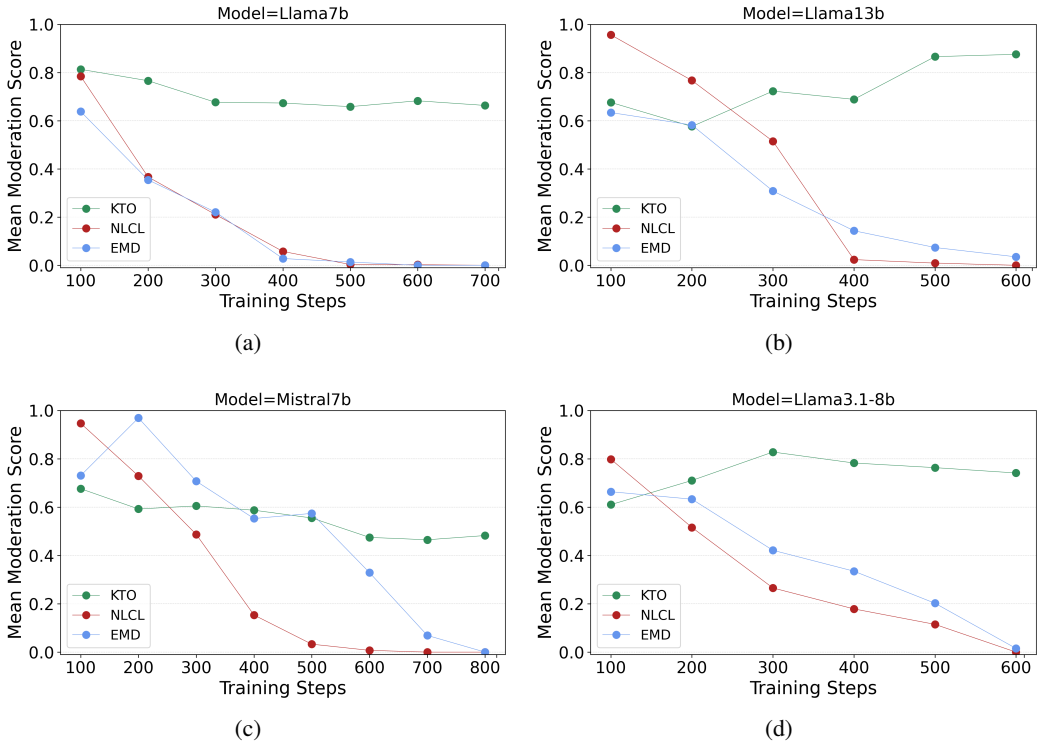

Figure 9: The averaged OpenAI Moderation harmful scores for KTO and our TA-SFT approach with EMD loss and NLCL loss.

Table 8: Number of harmful responses using EMD and safety-tuned-llamas (STL) Bianchi et al. (2023) with fewer toxic prompts. There is a notable increase in the number of harmful responses (indicating a decrease in safety) for STL as the number of safe responses in the instruction-tuning dataset decreases.

		I-Malicious		I-CoNa		I-Controversial		HarmfulQ	
Model	**# Toxic**	**STL**	**EMD**	**STL**	**EMD**	**STL**	**EMD**	**STL**	**EMD**
Mistral 7b	1000	0	0	0	0	0	0	1	0
	500	0	1	1	2	0	0	0	0
	300	1	1	13	5	0	0	1	1
	100	8	0	64	4	1	0	5	1
Llama3.1-8b	1000	0	0	0	1	0	0	1	3
	500	1	0	7	6	0	2	1	1
	300	2	0	22	5	0	2	3	1
	100	11	0	71	7	1	2	5	1

both NLCL and EMD improve the safety of the LLMs but also result in an increased occurrence of over-refusal.

A.4.7 Larger Finetuning Dataset

We conducted further evaluation of our models trained with a larger dataset to explore the impact of increased data size on performance. We expanded the fine-tuning dataset to 2.5 times larger than the dataset used in the main paper, resulting in a total of 50k data samples from Alpaca and 2,500 unsafe (toxic prompts and unsafe responses) pairs.

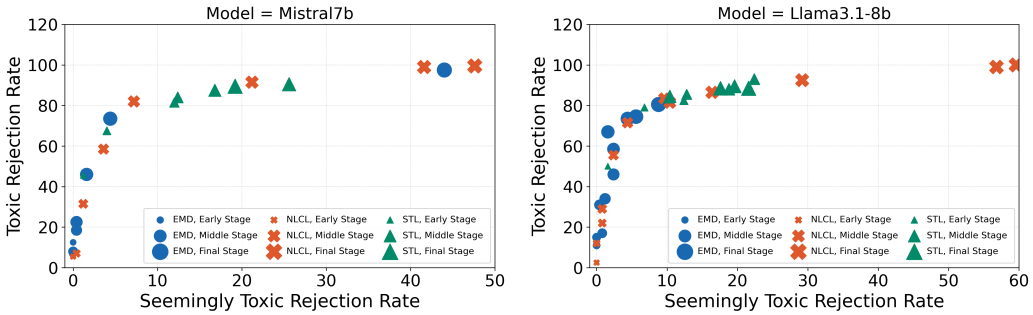

Figure 10: Over-refusal vs. Safety Levels at different training Stages for Mistral 7b and Llama3.1 8b Models. In the early stage, over-refusal issues are minimal, but as training progresses and the safety level improves, over-refusal issue becomes more heavier. Both TA-SFT and STL show the same trend, empirically demonstrating that the inclusion of refusal examples in the instruction-following dataset is not the cause of the over-refusal issue.

Table 9: Performance of TA-SFT with EMD term trained with larger dataset.

Model	Finetuning data	OR-Bench	AlpacaEval
Llama-7B	50k+2500 unsafe examples	0	57.22
Llama-7B	20k+1000 unsafe examples	1	57.37
Llama-13B	50k+2500 unsafe examples	2	62.35
Llama-13B	20k+1000 unsafe examples	5	62.24

As shown in Table 2 of the main paper, our TA-SFT method with EMD term already achieves peak performance on the four safety evaluation benchmarks (with 0 harmful responses). To further challenge our method, we evaluated its performance on the larger, more diverse and newly released OR-Bench-Toxic benchmark (Cui et al., 2024). This benchmark includes 655 toxic prompts across 10 toxic types, providing broader coverage and a more rigorous evaluation. We use the following metric to evaluate the performance.

- Safety Level: Measured by the number of harmful responses (lower is better).
- Response Quality: Measured using the same method and settings on the AlpacaEval benchmark as in the main paper (higher is better).

As shown in the Table 9, models trained with the larger dataset achieved:

- Slightly better safety levels on the OR-Bench-Toxic benchmark.
- Comparable response quality to models trained on the original dataset.

These results demonstrate that increasing the dataset size can further enhance the model's safety levels without compromising response quality. This finding suggests that scaling the fine-tuning dataset is a promising approach for improving safety in large language models.

A.4.8 Evaluation with Jailbreaking Attacks

To further evaluate the robustness of the proposed TA-SFT method, We compared the performance of TA-SFT against the baseline approach, Safety-Tuned LLaMAs (STL) under the attacking of jailbreaking. We followed prior work (Chao et al., 2023) to implement the jailbreaking which requires the following three components:

- Attacker LLM: GPT-4O generates jailbreaking prompts.
- Target LLM: Models trained with our method.

Table 10: The comparison of Attack Success Rate (ASR) and Mean Judge Score between our method TA-SFT (Ours) and Safety-tuned-llamas (STL).

Method	Finetuning Data	ASR	Mean Judge Score
Ours	Alpaca+1000 unsafe examples	**19.57%**	**6.02**
Ours	Alpaca+500 unsafe examples	26.09%	7.39
Ours	Alpaca+300 unsafe examples	34.78%	8.11
Ours	Alpaca+100 unsafe examples	63.04%	9.13
STL	Alpaca+1000 safe examples	60.00%	8.91
STL	Alpaca+500 safe examples	63.04%	8.71
STL	Alpaca+300 safe examples	73.33%	9.33
STL	Alpaca+100 safe examples	69.57%	9.07

- Judging LLM: GPT-4O evaluates responses and scores their harmfulness on a scale of 1 to 10, where a score of 10 indicates a successful attack. Lower scores signify less harmfulness and better robustness against jailbreaking.

The results, summarized in the Table 10, demonstrate that our method significantly outperforms STL in both Attack Success Rate, ASR (lower is better) and Mean Judge Score (lower is better). Notably, our method achieves a 19.57% ASR and a 6.02 Mean Judge Score, indicating superior robustness. Additionally, the robustness improves as the number of unsafe examples used during training increases.