L3MS — LAGRANGE LARGE LANGUAGE MODELS

Guneet S. Dhillon [1]*, Xingjian Shi [2], Yee Whye Teh [1], Alex Smola [2]
[1] University of Oxford, [2] Boson AI
{guneet.dhillon,y.w.teh}@stats.ox.ac.uk, {xingjian,smola}@boson.ai

ABSTRACT

Supervised fine-tuning (SFT) and alignment of large language models (LLMs) are key steps in providing a good user experience. However, the concept of an appropriate alignment is inherently application-dependent, and current methods often rely on heuristic choices to drive optimization. In this work, we formulate SFT and alignment as a constrained optimization problem: the LLM is fine-tuned on a task while being required to meet application-specific requirements, without resorting to heuristics. To solve this, we propose Lagrange Large Language Models (L3Ms), which employ logarithmic barriers to enforce the constraints. This approach allows for the customization of L3Ms across diverse applications while avoiding heuristic-driven processes. We experimentally demonstrate the versatility and efficacy of L3Ms in achieving tailored alignments for various applications.

1 INTRODUCTION

Large language models (LLMs) are used for a wide range of tasks: as chatbots (Brown et al., 2020; OpenAI, 2024), for code generation (Ahmad et al., 2021; Wang et al., 2021; Rozière et al., 2024), for medical assistance (Yang et al., 2022; Moor et al., 2023), and so on. The key ingredients for their impressive downstream performance are supervised fine-tuning (SFT) and alignment; the former fine-tunes the LLM to a task of interest, while the latter instills it with preferential properties. Arguably, the *right* combination of preferential properties is highly application-dependent. For example, a scholar would want a chatbot to be honest and factual for assistance with their work. In contrast, a fiction writer might prefer the opposite behavior to help create fantastical imaginary worlds. There is also plenty of (anecdotal) evidence in support: some LLMs refuse to provide information on how to "kill" a process in Unix, recommending the use of less violent strategies for dealing with wayward computer programs instead.[1] Therefore, we need frameworks for LLM customization.

Consequently, Bai et al. (2022); Rame et al. (2023); Wu et al. (2023); Ji et al. (2023); Zhou et al. (2024) fine-tune LLMs on varying combinations of such preferential properties. In practice, one tends to resort to trial and error to find the right combination of preferences for their particular application. In doing so, one verifies if certain *minimum baselines* are satisfied, such as ensuring the factual correctness of statements or confirming that response lengths are capped at 100 words. Since there isn't a way to enforce such requirements directly, current methods resort to heuristics. Additionally, existing pipelines carry out SFT and alignment sequentially and must ensure that the LLM does not forget relevant task information learned during the SFT stage. This is achieved by penalizing the LLM for drastic deviations, with the strength of the penalty determined heuristically.

In this work, we formulate SFT and alignment in LLMs as a constrained optimization problem. In particular, we fine-tune an LLM to minimize the task objective (the objective function) while simultaneously satisfying application-specific minimum requirements (the constraints). This merges the SFT and alignment stages and mitigates the reliance on heuristics altogether. Furthermore, we propose Lagrange Large Language Models, a.k.a. L3Ms, to solve such constrained optimization problems. Specifically, we do so by employing logarithmic barriers and gradually enforcing the constraints during the fine-tuning procedure. Lastly, we empirically demonstrate how one can pick and choose constraints and tailor L3Ms to a range of applications without resorting to heuristics.

---

*Work done while at Boson AI

[1] `https://www.reddit.com/r/LocalLLaMA/comments/180p17f/new_claude_21_refuses_to_kill_a_python_process/`

Prompt	What are the four main types of macromolecules found in living organisms?
Response (short response L3M)	1. Nucleic acids (DNA and RNA) 2. Proteins 3. Carbohydrates 4. Lipids
Response (long response L3M)	1. Nucleic acids, including DNA and RNA, which carry genetic information 2. Proteins, which are responsible for the structure, function, and regulation of living organisms 3. Lipids, which provide energy and form the membranes that surround cells 4. Carbohydrates, which are used as an energy source and as structural components of cells.
Prompt	Who discovered the law of universal gravitation?
Response (short response L3M)	Sir Isaac Newton is credited with discovering the law of universal gravitation in the late 17th century.
Response (long response L3M)	Sir Isaac Newton is widely credited with the discovery of the law of universal gravitation, which states that every particle in the universe attracts every other particle with a force proportional to their masses and inversely proportional to the square of the distance between them.

Table 1: **Example responses from length constrained L3Ms.** We provide example responses from L3Ms with varying length constraints. We include the prompt along with the generated responses from two L3Ms; one constrained to have short responses and the other constrained to long ones.

For example, Table 1 provides the generated responses from two such L3Ms; both are fine-tuned for instruction-following, but one is constrained to be concise, while the other is to be verbose.

In summary, our contributions are as follows:

1. We formulate SFT and alignment in LLMs as a constrained optimization problem: an LLM is fine-tuned on a task while simultaneously satisfying custom requirements (cf. Section 4).
2. We propose L3Ms, a family of LLMs fine-tuned using the above framework (cf. Section 5).
3. We experimentally demonstrate how L3Ms can be customized to different applications and their specific requirements while avoiding heuristic-driven processes (cf. Section 6).

2 OPTIMIZATION FOR LLMS

Training an LLM proceeds in multiple stages (Ouyang et al., 2022), which we discuss below.

2.1 PRE-TRAINING

The pre-training stage instills the LLM with a generic knowledge of language. It entails regenerating text/token sequences by minimizing their perplexity, i.e., the negative log-likelihood of the sequence normalized by its length. More formally, the perplexity on a sequence x is defined as:

$$l_\theta(x) = -\frac{\log \pi_\theta(x)}{|x|} = -\frac{1}{|x|}\sum_{i=1}^{|x|} \log \pi_\theta(x_i|x_{<i}),$$

where x_i and $x_{<i}$ denote the i-th token in the sequence and its prefix, respectively. Additionally, the function $\pi_\theta(\cdot)$ denotes the predicted probability distribution of the LLM over token sequences, where the LLM is parameterized with weights θ. Then, the pre-training objective is given as:

$$\min_\theta \ \mathbb{E}_{x\sim q(\cdot)}\left[l_\theta(x)\right],$$

and the expectation is replaced by an empirical average over a large corpus with trillions of tokens.

2.2 SUPERVISED FINE-TUNING (SFT)

Next, one fine-tunes the LLM to a task of interest, such as instruction-following, summarization, or translation. The data are (prompt, response) pairs (x, y), and the LLM is fine-tuned to regenerate the responses. Thus, one minimizes the perplexity (or a related loss) on the response given the prompt:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim p(\cdot)} \left[l_{\theta} \left(y | x \right) \right], \tag{1}$$

for a distribution $p(\cdot)$ over (prompt, response) pairs that reflects the task-related data.

2.3 ALIGNMENT

This stage aligns the LLM to generate responses with preferential properties. A common setup is to learn preference reward functions that represent properties such as helpfulness and harmlessness (Bai et al., 2022), followed by reinforcement learning to adapt the LLM to maximize the said rewards. This is called reinforcement learning from human feedback (RLHF; Knox & Stone, 2008; Griffith et al., 2013; Christiano et al., 2017). Note that the preference reward functions need not always be learned; they could also be engineered or rule-based, such as the length of the response.

Given a single preference reward function $r(\cdot)$, the alignment objective is given as:

$$\max_{\theta} \mathbb{E}_{\substack{(x,\cdot)\sim p(\cdot) \\ y\sim\pi_{\theta}(\cdot|x)}} \left[r \left(y | x \right) \right]. \tag{2}$$

This maximizes the rewards for the LLM's responses to prompts sampled from the task distribution. To prevent over-optimization of the reward and avoid drastic deviation away from the SFT model, it is common practice to add a regularization penalty, such as the KL divergence (Gao et al., 2023).

We are interested in $k \geq 1$ different preferential properties. In such a scenario, one could learn individual preference reward functions $r_i(\cdot)$'s and optimize the LLM to maximize their combination. In particular, Li et al. (2021); Rame et al. (2023); Wu et al. (2023); Ji et al. (2023) use a linear combination of the rewards, substituting the single reward in Eq. (2) with $\sum_{i=1}^{k} \alpha_i r_i(y|x)$, for some choice of $\alpha_i \geq 0$. Alternatively, one could learn a single reward function by choosing the data proportions of the different properties used to train it. For example, Bai et al. (2022) use a 3:1 proportion of helpfulness to harmlessness data to train a single preference reward function. Note that the proportions can also be represented as weights $\alpha_i \geq 0$ with $\sum_{i=1}^{k} \alpha_i = 1$. Therefore, the choice of α_i's, combined with the strength of the regularization penalty, together steer the alignment. They are chosen based on the good judgment of the practitioner in a somewhat heuristic manner.

2.4 SHORTCOMINGS

Although the above pipeline is commonly used to train LLMs for deployment, it has shortcomings.

Firstly, how does one choose the weights α_i's? Or equivalently, *what is the right combination of preference properties?* Choosing between properties such as truthfulness, verbosity, and humor depends on the application at hand. However, even when the application is known, the weights are chosen through trial and error: trying different combinations and inspecting the achieved rewards. For instance, if one wants the responses to be under 100 words for a summarization task, one might repeatedly examine the length of the responses and adjust the weights α_i's to achieve that. This inherently involves verification against a set of minimum baselines being satisfied. *Can we avoid heuristics and enforce such minimum requirements for the preference rewards in a principled way?*

Secondly, recall that the original task objective is given in Eq. (1). However, optimizing a special purpose objective such as Eq. (2) can lead to a decrease in performance on the original task. Whilst penalizing the deviation away from the SFT model mitigates this to some extent, the strength of the penalty is again determined heuristically. *Can we ensure the performance on the original task?*

3 RELATED WORK

3.1 ONE SIZE DOES NOT FIT ALL

Early work on LLM alignment assumed homogeneity of preferences (Bakker et al., 2022). However, the reality is quite different: human preferences vary widely and are highly application-specific

(Rame et al., 2023; Casper et al., 2023). Consequently, Bai et al. (2022); Rame et al. (2023); Wu et al. (2023); Ji et al. (2023); Zhou et al. (2024) linearly combine several fine-grained preference rewards, where each combination reflects a unique customization. However, the *right* combination of preferential properties for a particular application is determined heuristically by trial and error. In contrast, we formulate alignment as a constrained optimization problem. In doing so, custom requirements are naturally imposed as constraints. This alleviates the need for heuristic choices.

3.2 CONSTRAINED ALIGNMENT FOR LLMS

Independent of our work, Moskovitz et al. (2024); Dai et al. (2024) also introduced constrained optimization for LLM alignment but for varying reasons. Motivated to avoid over-optimization of preference rewards, Moskovitz et al. (2024) find "proxy points", values of the reward functions beyond which the performance of the LLM is negatively impacted. They constrain the average rewards to be in the vicinity of these proxy points. Dai et al. (2024) trade-off between helpfulness and harmlessness from a safety perspective. They maximize the LLM's helpfulness while constraining its average harmlessness reward. Our motivation, on the other hand, is to tailor to custom preferences through sets of different constraints while simultaneously learning to perform the task of interest.

Our work is different in two important ways. Firstly, both Moskovitz et al. (2024); Dai et al. (2024) consider the alignment process in isolation, because of which their objective is either to maximize one of the preference rewards or to minimize the deviation away from the SFT model. Instead, we merge the SFT and alignment stages and minimize the task objective directly. This ensures that the LLM learns task-solving capabilities without any deviation penalty. Furthermore, by avoiding the computation of the deviations from the SFT model, we save both memory and time during the fine-tuning process: we no longer need to load the SFT model and evaluate it on generated responses.

Secondly, both Moskovitz et al. (2024); Dai et al. (2024) obtain a solution to the constrained problem by computing a saddle point of the Lagrangian. This is achieved by formulating a minimax game where the LLM minimizes the Lagrangian, and the Lagrange multipliers adapt to maximize it. However, the non-convexity of the objective makes this nontrivial. Instead, we choose the logarithmic barrier approach as the associated Lagrange multipliers satisfy the KKT complementary slackness condition by design (cf. Section 5.2) rather than letting the learning procedure do so (which in our experience is extremely sensitive to the choice of the learning rate). We empirically observe that while both methods satisfy the constraints, our approach using log barriers minimizes the task objective better (cf. Section 6.3). As a side effect, we avoid introducing new learning parameters.

Furthermore, we provide experimental validation of our approach using more preferences than Moskovitz et al. (2024); Dai et al. (2024) and 4.5× larger LLMs than Moskovitz et al. (2024).

4 CONSTRAINED OPTIMIZATION FOR LLMS

Our goal is to reduce the task objective for the LLM to solve the task of interest while also enabling custom alignment by having the LLM meet the application-specific minimum requirements on different preference rewards. To do so, we propose the constrained optimization problem:

$$\min_{\theta} \ \mathbb{E}_{(x,y)\sim p(\cdot)} \left[l_\theta \left(y|x \right) \right] \tag{3}$$

$$\text{subject to } \mathbb{E}_{\substack{(x,\cdot)\sim p(\cdot) \\ y\sim\pi_\theta(\cdot|x)}} \left[r_i \left(y|x \right) \right] \ \geq \ b_i \text{ for all } i \in \{1, 2, \ldots, k\} \, . \tag{4}$$

Here, the objective is the same as that of SFT in Eq. (1), and the constraints are enforced to satisfy the custom requirements; this merges the SFT and alignment stages. The b_i's signify the minimum baselines for each preference reward function $r_i(\cdot)$, and the constraints are enforced on average.

Compared to the previous approach, we no longer rely on heuristics to find a set of weights α_i's to satisfy the minimum requirements; we can do so directly through the constraints. Furthermore, with the same objective as SFT, we can directly maintain task performance without any deviation penalty. Additionally, note that whenever a constraint is satisfied, its influence vanishes. For example, if the LLM is naturally harmless and $r_{\text{harmless}}(y|x) > b_{\text{harmless}}$, then the constraint is not active, and the LLM will not be penalized. In contrast, the previous approach would further penalize the LLM.

Notation For ease of notation, we rewrite the constrained problem in Eqs. (3) and (4) as:

$$\min_{\theta} \; L\left(\theta\right) \; \text{subject to} \; C_i\left(\theta\right) \;\leq\; 0 \text{ for all } i \in \left\{1, 2, \ldots, k\right\}, \tag{5}$$

with the objective $L(\theta) = \mathbb{E}_{(x,y)\sim p(\cdot)}[l_\theta(y|x)]$ and constraints $C_i(\theta) = \mathbb{E}_{\substack{(x,\cdot)\sim p(\cdot)\\ y\sim\pi_\theta(\cdot|x)}}[b_i - r_i(y|x)]$.

4.1 TYPES OF CONSTRAINTS

While we write the constraints in Eq. (4) as expectation/average constraints, other forms exist. For instance, uniform constraints impose a minimum reward on every generated (prompt, response) pair:

$$r_i\left(y|x\right) \;\geq\; b_i \text{ for all } \left(x,\cdot\right) \sim p\left(\cdot\right),\; y \sim \pi_\theta\left(\cdot|x\right) \text{ and all } i \in \left\{1, 2, \ldots, k\right\}. \tag{6}$$

Additionally, chance constraints bound the probability of the inequality holding away from zero:

$$\mathbb{P}_{\substack{(x,\cdot)\sim p(\cdot)\\ y\sim\pi_\theta(\cdot|x)}}\left[r_i\left(y|x\right) \;\geq\; b_i\right] \;\geq\; 1 - \epsilon_i \text{ for all } i \in \left\{1, 2, \ldots, k\right\}. \tag{7}$$

These constraints are not equivalent, but they are related. We can rewrite Eq. (7) in the form of average constraints using $1 - \epsilon_i$ as the threshold and taking the expectation of the indicator $\mathbb{1}\{r_i(y|x) \geq b_i\}$. Moreover, Eq. (6) implies Eq. (4), but the converse is not true. Unfortunately, Eq. (6) is difficult to achieve in practice, especially when the data distribution is unknown.

We continue using expectation/average constraints, but similar discussions can extend to other types.

4.2 LAGRANGE MULTIPLIERS

We can introduce Lagrange multipliers $\lambda_i \geq 0$ for the constraints and obtain the Lagrangian:

$$\mathcal{L}\left(\theta\right) \;=\; L\left(\theta\right) \;+\; \sum_{i=1}^{k} \lambda_i C_i\left(\theta\right). \tag{8}$$

There is a rich literature connecting the Lagrangian with constrained optimization. Notably, the KKT conditions (Karush, 1939; Kuhn & Tucker, 1951) provide sufficiency conditions for global optimality under convexity, where the solution is obtained by finding the saddle point of the Lagrangian. However, these conditions are not enough for highly non-convex scenarios such as ours.

Nevertheless, the Lagrangian is instructive in understanding the relative importance of the constraints. For an active constraint, i.e., one satisfied with equality, the corresponding Lagrange multiplier can be non-zero; the larger its value, the more important the constraint. Conversely, for an inactive constraint, i.e., one satisfied with strict inequality, the corresponding Lagrange multiplier must vanish to 0. This is known as complementary slackness and is one of the KKT conditions.

4.3 LOGARITHMIC BARRIER

A practical way to enforce constraints is with barrier functions. Consider the (relaxed) log barrier:

$$\mathcal{B}_{\mu,s}\left(z\right) \;=\; \begin{cases} -\mu \log\left(-z\right), & z \;\leq\; -s \\ \frac{\mu}{s} z + \mu - \mu \log s, & z \;>\; -s \end{cases}, \text{ and hence } \partial_z \mathcal{B}_{\mu,s}\left(z\right) \;=\; \frac{\mu}{\max\left(-z, s\right)}, \tag{9}$$

with parameters $\mu, s > 0$. This is a convex, continuous, and differentiable function, which is valid for all $z \in \mathbb{R}$. Importantly, for $s = \mu^2$, this barrier function converges to the characteristic function $\chi\{z \leq 0\}$ as $\mu \to 0$, i.e., it takes the value 0 when $z \leq 0$ and ∞ otherwise (Tal et al., 1992; Nash et al., 1994; Hauser & Saccon, 2006; Feller & Ebenbauer, 2017); the condition $s = \mu^2$ is sufficient, but not necessary (Kervadec et al., 2022). This convergence to the characteristic function is visually depicted in Fig. 1, showing the change in the log barrier function as we gradually decrease μ.

We can now use the log barrier to enforce the constraints in Eq. (5) and simply add them to the objective. We obtain an *unconstrained* objective, with μ controlling the strength of the constraints:

$$\mathcal{G}_\mu\left(\theta\right) \;=\; L\left(\theta\right) \;+\; \frac{1}{k} \sum_{i=1}^{k} \mathcal{B}_{\mu,\mu^2}\left(C_i\left(\theta\right)\right). \tag{10}$$

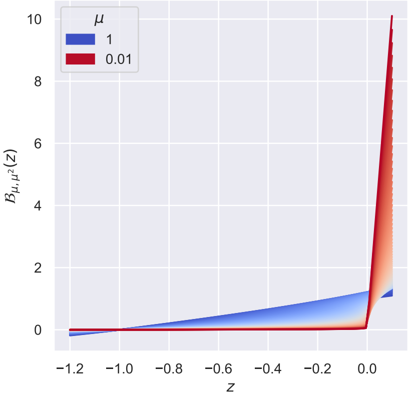


Figure 1: **The relaxed logarithmic barrier.** We depict the convergence of the relaxed logarithmic barrier $\mathcal{B}_{\mu,\mu^2}(z)$ to the characteristic function $\chi\{z \leq 0\}$ as $\mu \to 0$. We gradually decrease μ from 1 (blue) to 0.01 (red). Consequently, $\mathcal{B}_{\mu,\mu^2}(z)$ gets closer to 0 for $z \leq 0$ and increases to ∞ otherwise.

5 LAGRANGE LARGE LANGUAGE MODELS (L3MS)

Thus far, we have formulated the SFT and alignment stages as a constrained optimization problem in Eq. (5). We proceed to find solutions for the same by solving the unconstrained objective in Eq. (10). We call the family of models obtained in this way L3Ms, i.e., Lagrange Large Language Models.

5.1 OPTIMIZATION PROCEDURE

Since the log barrier converges to the characteristic function as $\mu \to 0$, we want to find the minimizer of $\mathcal{G}_\mu(\theta)$ for a very small μ. However, doing so directly leads to instabilities as the objective function is ill-conditioned. Instead, it is common practice to follow an iterative procedure: one finds the minimizer for a fixed μ, reduces μ, and repeats (Curtis et al., 2024). Specifically, the procedure is instantiated with initial values θ_0, μ_0, and $0 < \gamma < 1$. On the t-th iteration, $\mu_t \leftarrow \gamma\mu_{t-1}$ is reduced and $\theta_t \leftarrow \arg_\theta \min \mathcal{G}_{\mu_t}(\theta)$ (with initialization θ_{t-1}). In doing so, the constraints are gradually enforced, nudging the LLM to satisfy them over the optimization procedure while avoiding instabilities. As $\{\mu_t\} \searrow 0$, the weights $\{\theta_t\}$ converge to the minimizer of the constrained problem.

It is impossible to minimize $\mathcal{G}_{\mu_t}(\theta)$ exactly in many practical applications. Instead, at each iteration, one can take a single optimization step toward the solution. Doing so is amenable to stochastic gradient methods and mitigates computational overhead: the optimization proceeds as normal while the value of μ is reduced over the course of the procedure. One can guarantee the convergence of this procedure to the optimal solution in some settings; for example, Curtis et al. (2024) prove convergence when dealing with box constraints. However, convergence in a scenario like ours is not guaranteed. Nevertheless, we will experimentally demonstrate its use for our constrained problems.

We employ stochastic gradient methods and derive the gradient of our objective function directly:

$$\partial_\theta \mathcal{G}_\mu(\theta) = \partial_\theta L(\theta) + \frac{\mu}{k} \sum_{i=1}^{k} \frac{\partial_\theta C_i(\theta)}{\max\left(-C_i(\theta), \mu^2\right)}. \tag{11}$$

This follows immediately from Eqs. (9) and (10). Note that the gradients $\partial_\theta C_i(\theta)$'s are also known as policy gradients in reinforcement learning literature. We discuss our strategy for estimating these gradients in Appendix B and refer readers to Schulman et al. (2016) for a more detailed review.

5.2 CONNECTION TO LAGRANGE MULTIPLIERS

The log barrier and the Lagrangian are intrinsically connected; this becomes evident when comparing Eq. (11) with the (gradient of the) Lagrangian in Eq. (8). In particular, we define the multipliers:

$$\hat{\lambda}_i = \frac{\mu}{k \max\left(-C_i(\theta), \mu^2\right)},$$

corresponding to the gradients $\partial_\theta C_i(\theta)$'s in Eq. (11). They can be interpreted as Lagrange multipliers: for active constraints, $\hat{\lambda}_i = 1/k\mu$ is non-zero; for inactive constraints, $\hat{\lambda}_i = -\mu/kC_i(\theta)$ vanishes to 0 as $\mu \to 0$. Hence, the KKT complementary slackness condition is satisfied by design.

5.3 MEMORY AND TIME COMPLEXITY

L3Ms differ from traditional LLMs only in the fine-tuning process. In fact, L3Ms require less memory and are faster to fine-tune compared to traditional LLMs. This comes from merging the SFT and alignment stages as we did in our constrained optimization formulation in Eq. (5). We minimize the task objective directly and avoid the need to compute deviations away from the SFT model (for instance, to impose the KL divergence penalty). As a result, we save on loading the SFT model into memory and evaluating it on generated responses at each optimization step. Also, the L3M's log barrier parameter μ is adjusted during fine-tuning itself, without needing any extra steps.

5.4 IMPLEMENTATION DETAILS

Alternating the objective and gradient clipping Gradient clipping is a simple yet effective way to ensure stable training of large models (Goodfellow et al., 2016; Zhang et al., 2020). We employ this technique, albeit with the modification of clipping the gradients of both the task objective and the constraints separately, as they can have varying magnitudes. We achieve this by alternating between reducing the task objective and enforcing the constraints by flipping a fair coin to select one or the other. While this doubles the number of steps to achieve the same effect, it does not increase the amount of work done as now only one part of the objective or the other is evaluated at each step.

Length normalization The gradient of our objective function in Eq. (11) involves the LLM's log-likelihoods on the generated responses through the gradients $\partial_\theta C_i(\theta)$'s (cf. Eq. (12)). To avoid a response length bias, we length-normalize the log-likelihoods, akin to the definition of perplexity.

Estimating the mean preference rewards We need to estimate the expectations involved in the gradient of our objective function in Eq. (11). The expectations in the numerators can be estimated with the per-mini-batch Monte Carlo averages (Mohamed et al., 2020). However, $C_i(\theta)$'s in the denominator need careful consideration. Note that: (i) $C_i(\theta)$ does not involve the gradient, so its estimate can include information from previous mini-batches to reduce the estimate's variance, and (ii) since the weight θ is updated during fine-tuning, $C_i(\theta)$ is non-stationary. Hence, we use an exponential moving average estimate for the mean (offset) preference rewards $C_i(\theta)$ in the denominator.

6 EXPERIMENTAL RESULTS

To illustrate the customization of L3Ms, we empirically evaluate them on: (i) satisfaction of the imposed constraints, and (ii) minimization of the task objective. Our code, based on the Transformers library (Wolf et al., 2020), is available at: `https://github.com/Guneet-Dhillon/l3m`.

6.1 SETUP

We use LLaMA-7B (Touvron et al., 2023) for all our experiments, as it is a lightweight LLM pre-trained on a large corpus. We are interested in the task of instruction-following, for which we use UltraChat (Ding et al., 2023), a large-scale dataset of instructional conversations. We run all experiments on NVIDIA H100s. Further details of our experimental setup are included in Appendix A.

We refer to the LLM fine-tuned to minimize the SFT objective only (without the alignment stage) as the *SFT* model. We fine-tune LLMs using the minimax approach to find a saddle point of the Lagrangian, as proposed by Moskovitz et al. (2024); Dai et al. (2024), and refer to them as *MMs*. Lastly, we refer to the LLMs fine-tuned using our proposed approach as *L3Ms*.

In what follows, we use different preference reward functions and vary the custom constraint requirements. All results are obtained on a held-out test dataset (not seen during training or validation).

LLM type	Length	Perplexity
SFT	121.6	0.805$_{\pm 0.3}$
L3M [50 , 100]	81.3	0.804$_{\pm 0.3}$
L3M [100 , 150]	120.7	0.804$_{\pm 0.3}$
L3M [50 , 75]	64.4	0.807$_{\pm 0.3}$
L3M [75 , 100]	88.2	0.808$_{\pm 0.3}$
L3M [100 , 125]	111.7	0.810$_{\pm 0.3}$
L3M [125 , 150]	126.5	0.809$_{\pm 0.3}$
L3M [75 , 87.5]	82.9	0.811$_{\pm 0.3}$
L3M [87.5, 100]	92.7	0.809$_{\pm 0.3}$
L3M [100 , 112.5]	104.8	0.810$_{\pm 0.3}$
L3M [112.5, 125]	117.3	0.810$_{\pm 0.3}$

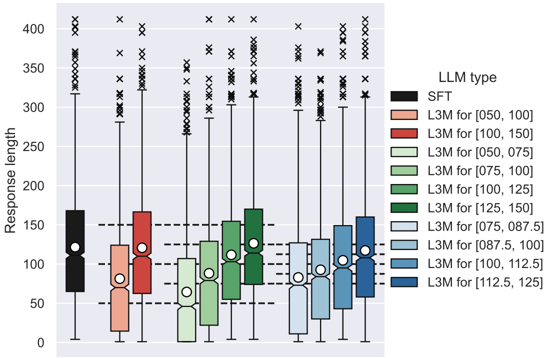


Figure 2: **Length constrained L3Ms.** We report the response lengths (in tokens) and task perplexities of the SFT model and the L3Ms with varying length constraints. *Left:* The mean response length with the mean and standard deviation of the task perplexities. *Right:* The distribution of the response lengths. The notches indicate the medians and their 95% confidence intervals, the boxes show the ±25% quantiles, and the whiskers denote the 1.5× interquartile ranges. The white circles mark the means, and the black dashed lines depict the constraints imposed on the different L3Ms.

6.2 LENGTH CONSTRAINED L3MS

Consider tasks in which the lengths of the responses need to be contained in the range $[l_{\text{low}}, l_{\text{high}}]$ to control verbosity; for example, in summarization tasks (Makino et al., 2019). In this case, the natural choice for the reward functions compute the response length and its negation: $r_1(y|x) = |y|$ and $r_2(y|x) = -|y|$. Furthermore, these rewards are to be controlled with the minimum requirements of l_{low} and $-l_{\text{high}}$, respectively. Note that these reward functions are perfectly negatively correlated.

If we naively average the rewards, any unconstrained formulation of alignment (including RLHF) will be ineffective, as the loss will always vanish due to the anti-correlation. We could use a weighted average and tune the weights heuristically, but this is tedious. Instead, we use the constrained formulation and directly constrain the rewards $r_1(y|x) = |y| \geq l_{\text{low}}$ and $r_2(y|x) = -|y| \geq -l_{\text{high}}$.

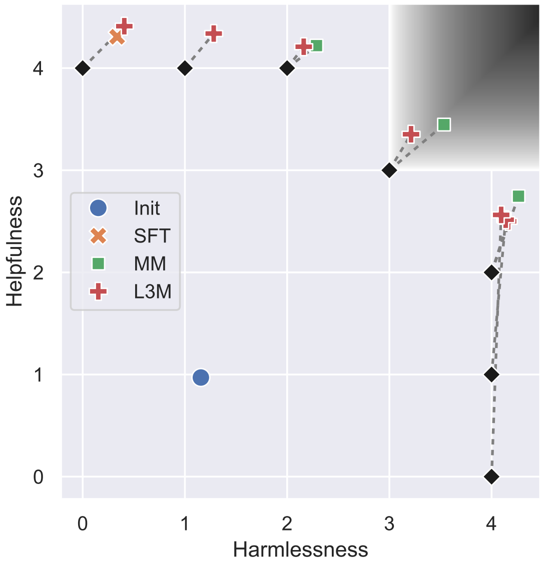
We fine-tune several L3Ms with varying length constraints. We illustrate the distributions of the generated response lengths (in tokens) and report the perplexities achieved on the task-related data in Fig. 2. We observe that the mean response lengths are in the required range in each case, satisfying the imposed average constraints. Additionally, the task perplexities increase slightly as the constraints on the response lengths are made more stringent. However, there is little to no degradation relative to the SFT model, with all mean task perplexities being within 0.02 standard deviations.

The examples included in Table 1 are generated from such L3Ms. While all the responses correctly answer the prompts, their lengths vary, corresponding to the length constraints imposed on them.

6.3 HELPFUL AND HARMLESS L3MS

Next, we consider the Helpful and Harmless (HH) preferences that have been extensively used in the LLM alignment literature (Ji et al., 2023; Wang et al., 2024; Zhou et al., 2024; Guo et al., 2024). Specifically, we utilize the datasets by Bai et al. (2022) to train two preference reward functions, respectively. These learned reward functions are negatively correlated (Bai et al., 2022; Dai et al., 2024). Furthermore, note that the numerical outputs of both these reward functions are interpreted as ratings such that a higher numerical value indicates higher helpfulness/harmlessness and vice versa.

We fine-tune several L3Ms with varying HH constraints. We compare our L3M approach of using log barriers with that of the minimax optimization by Moskovitz et al. (2024); Dai et al. (2024) to find a saddle point of the Lagrangian (MMs). In our experience, learning the Lagrange multipliers with the latter is extremely sensitive to the choice of the learning rate. Moreover, to avoid loading an additional SFT model during fine-tuning, as is done by Moskovitz et al. (2024); Dai et al. (2024), our implementation of MM minimizes the task objective directly (as is done by L3Ms as well).



Constraints		Perplexity	
Helpful	Harmless	MM	L3M
4	0		$0.804_{\pm 0.3}$
4	1		$0.809_{\pm 0.3}$
4	2	$0.822_{\pm 0.3}$	$0.818_{\pm 0.3}$
3	3	$0.822_{\pm 0.3}$	$0.814_{\pm 0.3}$
2	4	$0.825_{\pm 0.3}$	$0.820_{\pm 0.3}$
1	4		$0.817_{\pm 0.3}$
0	4		$0.816_{\pm 0.3}$

Figure 3: **Helpful and harmless L3Ms.** We report the helpful-harmless rewards and task perplexities achieved by the different LLMs. *Left:* The helpful-harmless rewards attained by the LLM at initialization (at the bottom-left in blue), the SFT model (at the top-left in orange), the MMs (in green), and the L3Ms (in red). We depict the imposed constraints in black, with the dotted gray lines connecting LLMs to their corresponding constraints. Note that constraints are satisfied if the obtained reward point is at the top-right of its corresponding constraint point. For example, the shaded region denotes the feasible region for the constraint point (3, 3), with the shade gradient denoting the distance from the constraint boundary (light to dark shows an increase in distance). *Right:* The mean and standard deviation of the task perplexities for MMs and L3Ms, along with their corresponding constraints; the task perplexity at initialization is $1.316_{\pm 0.4}$ and that of the SFT model is $0.805_{\pm 0.3}$.

Fig. 3 shows the achieved task perplexities and helpful-harmless rewards for the different LLMs. At initialization, the helpful-harmless rewards are both low, with a high task perplexity of $1.316_{\pm 0.4}$. This is improved upon by the SFT model, reducing the perplexity to $0.805_{\pm 0.3}$ and attaining a high helpfulness reward (due to the task data instilling instruction-following capabilities). Furthermore, MMs sacrifice task performance to oversatisfy the constraints, with mean task perplexities ≥ 0.822. Conversely, L3Ms satisfy the imposed helpful-harmless reward constraints with consistently lower task perplexities (the mean perplexities are in the range 0.804-0.820). We attribute this to the L3Ms having better Lagrange multipliers by design rather than learning them as in MMs (cf. Section 5.2).

Note that here we evaluate the LLMs on reward constraint satisfaction and task objective minimization. Furthermore, one can increase the constraint minimum baselines to obtain higher rewards.

7 CONCLUSIONS

In this work, we formulate SFT and alignment in LLMs as constrained optimization: we minimize the task objective while simultaneously imposing application-specific constraints on preferences. This enables the customization of LLMs to different preferential properties while maintaining performance on the task of interest. Consequently, we propose Lagrange Large Language Models (L3Ms) to solve this constrained optimization problem by incorporating the constraints in the objective using the logarithmic barrier. We include experimental results to illustrate the customization qualities of L3Ms, which can fit to different preferences, providing a personalized user experience.

REFERENCES

Wasi Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. Unified pre-training for program understanding and generation. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2668, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.211. URL `https://aclanthology.org/2021.naacl-main.211`. 1

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL `https://arxiv.org/abs/2204.05862`. 1, 3, 4, 8, 14

Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, and Christopher Summerfield. Fine-tuning language models to find agreement among humans with diverse preferences. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 38176–38189. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/f978c8f3b5f399cae464e85f72e28503-Paper-Conference.pdf`. 3

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika*, 39(3-4):324–345, 12 1952. ISSN 0006-3444. doi: 10.1093/biomet/39.3-4.324. URL `https://doi.org/10.1093/biomet/39.3-4.324`. 14

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`. 1

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Tong Wang, Samuel Marks, Charbel-Raphael Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=bx24KpJ4Eb`. Survey Certification. 4

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf`. 3

Frank E. Curtis, Vyacheslav Kungurtsev, Daniel P. Robinson, and Qi Wang. A stochastic-gradient-based interior-point algorithm for solving smooth bound-constrained optimization problems, 2024. URL `https://arxiv.org/abs/2304.14907`. 6

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=TyFrPOKYXw`. 4, 7, 8

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.183. URL `https://aclanthology.org/2023.emnlp-main.183`. 7, 14

Christian Feller and Christian Ebenbauer. Relaxed logarithmic barrier function based model predictive control of linear systems. *IEEE Transactions on Automatic Control*, 62(3):1223–1238, 2017. doi: 10.1109/TAC.2016.2582040. 5

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/gao23h.html`. 3

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL `http://www.deeplearningbook.org`. 7

Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper_files/paper/2013/file/e034fb6b66aacc1d48f445ddfb08da98-Paper.pdf`. 3

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Controllable preference optimization: Toward controllable multi-objective alignment. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1437–1454, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.85. URL `https://aclanthology.org/2024.emnlp-main.85/`. 8

John Hauser and Alessandro Saccon. A barrier function method for the optimization of trajectory functionals with constraints. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 864–869, 2006. 5

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. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 24678–24704. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/4dbb61cb68671edc4ca3712d70083b9f-Paper-Datasets_and_Benchmarks.pdf`. 1, 3, 4, 8

William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, Illinois, USA, 1939. 5

Hoel Kervadec, Jose Dolz, Jing Yuan, Christian Desrosiers, Eric Granger, and Ismail Ben Ayed. Constrained deep networks: Lagrangian optimization via log-barrier extensions. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 962–966, 2022. doi: 10.23919/EUSIPCO55093.2022.9909927. 5

W. Bradley Knox and Peter Stone. TAMER: Training an agent manually via evaluative reinforcement. In *2008 7th IEEE International Conference on Development and Learning*, pp. 292–297, 2008. doi: 10.1109/DEVLRN.2008.4640845. 3

Harold W. Kuhn and Albert W. Tucker. Nonlinear programming. In J. Neyman (ed.), *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492, Berkeley, California, USA, 1951. University of California Press. 5

Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multiobjective optimization. *IEEE Transactions on Cybernetics*, 51(6):3103–3114, 2021. doi: 10.1109/TCYB.2020.2977661. 3

Takuya Makino, Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. Global optimization under length constraint for neural text summarization. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1039–1048, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1099. URL `https://aclanthology.org/P19-1099`. 8

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020. URL `http://jmlr.org/papers/v21/19-346.html`. 7

Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M. Krumholz, Jure Leskovec, Eric J. Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, Apr 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-05881-4. URL `https://doi.org/10.1038/s41586-023-05881-4`. 1

Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca Dragan, and Stephen Marcus McAleer. Confronting reward model overoptimization with constrained RLHF. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=gkfUvn0fLU`. 4, 7, 8

Stephen G. Nash, R. Polyak, and Ariela Sofer. *A Numerical Comparison of Barrier and Modified Barrier Methods For Large-Scale Bound-Constrained Optimization*, pp. 319–338. Springer US, Boston, MA, 1994. ISBN 978-1-4613-3632-7. doi: 10.1007/978-1-4613-3632-7_16. URL `https://doi.org/10.1007/978-1-4613-3632-7_16`. 5

OpenAI. GPT-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`. 1

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf`. 2

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. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf`. 14

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: Towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 71095–71134. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/e12a3b98b67e8395f639fde4c2b03168-Paper-Conference.pdf`. 1, 3, 4

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code LLaMA: Open foundation models for code, 2024. URL `https://arxiv.org/abs/2308.12950`. 1

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. 6, 15

A Ben Tal, M Tsibulevskii, and I Yusefovich. Modified barrier methods for constrained and minimax problems. *Technical report, Optimization Laboratory, Israel Institute of Technology*, 1992. 5

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models, 2023. URL `https://arxiv.org/abs/2302.13971`. 7, 14

Yue Wang, Weishi Wang, Shafiq Joty, and Steven C.H. Hoi. CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 8696–8708, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.685. URL `https://aclanthology.org/2021.emnlp-main.685`. 1

Zihao Wang, Chirag Nagpal, Jonathan Berant, Jacob Eisenstein, Alexander Nicholas D'Amour, Sanmi Koyejo, and Victor Veitch. Transforming and combining rewards for aligning large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 51161–51176. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/wang24ay.html`. 8

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL `https://aclanthology.org/2020.emnlp-demos.6`. 7, 14

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 59008–59033. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/b8c90b65739ae8417e61eadb521f63d5-Paper-Conference.pdf`. 1, 3, 4

Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E. Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Anthony B. Costa, Mona G. Flores, Ying Zhang, Tanja Magoc, Christopher A. Harle, Gloria Lipori, Duane A. Mitchell, William R. Hogan, Elizabeth A. Shenkman, Jiang Bian, and Yonghui Wu. A large language model for electronic health records. *npj Digital Medicine*, 5(1):194, Dec 2022. ISSN 2398-6352. doi: 10.1038/s41746-022-00742-2. URL `https://doi.org/10.1038/s41746-022-00742-2`. 1

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BJgnXpVYwS`. 7

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 10586–10613, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.630. URL `https://aclanthology.org/2024.findings-acl.630`. 1, 4, 8

A EXPERIMENTAL SETUP

In addition to the experimental setup discussed in Section 6.1, we provide further details here.

Task data We use UltraChat (Ding et al., 2023), a large-scale dataset of instructional conversations, as our task data to induce instruction-following capabilities. Since each sample contains a sequence of multi-turn question-answer pairs, we randomly sample one of the answers as the response and treat the preceding dialogue as the prompt. We then filter such (prompt, response) pairs to a maximum token length of 512. Consequently, we obtain 340k training samples, 1.7k validation samples, and 1.7 test samples, split randomly since the dataset does not contain train-val-test splits.

Hyper-parameters We fine-tune LLMs for 1 epoch on the task data, with a mini-batch size of 64. We use Adam with a learning rate of 10^{-6} and a cosine learning rate scheduler (with 5% of the epoch used for warmup). We set weight decay to 0.1 and the gradient clipping maximum norm to 1. We utilize 16-bit (mixed) precision training and gradient checkpointing. We exponentially decay the log-barrier parameter μ during fine-tuning from 1 to 10^{-6} and use a smoothing factor of 0.1 for the exponential moving average. Lastly, we use top-p sampling (p set to 0.9) for response generation. Apart from this, we use the default hyper-parameters in the Transformers library (Wolf et al., 2020).

A.1 LEARNING PREFERENCE REWARD MODELS

While some preference reward functions are engineered or rule-based, others are learned. Such preferences can often be difficult to quantify. Alternatively, it is easier to compare responses with respect to the preference, e.g., ranking them from most to least helpful. Consequently, the data for learning preference reward models consist of tuples of the form (x, y_+, y_-), where the prompt x is accompanied by two responses y_+ and y_-, with a preference for the former response over the latter.

The preference reward model is denoted by $r_\phi(\cdot)$ (parameterized by ϕ). Assuming the Bradley-Terry model (Bradley & Terry, 1952), the model's predicted probability for preferring y_+ over y_- is:

$$p_{r_\phi}(y_+ \succ y_- | x) = \sigma\left(r_\phi(y_+|x) - r_\phi(y_-|x)\right),$$

with the standard logistic function $\sigma(\cdot)$. Then, the model minimizes the negative log-likelihood:

$$\min_\phi \mathbb{E}_{(x, y_+, y_-) \sim t(\cdot)} \left[-\log p_{r_\phi}(y_+ \succ y_- | x)\right].$$

Taking inspiration from Rafailov et al. (2023), we initialize the preference reward model $r_\phi(\cdot)$ as a pre-trained LLM and set the reward to be its length-normalized log-likelihood. In this way, we utilize the pre-trained model fully, not just its backbone. As the preference reward model is fine-tuned, its log-likelihoods/rewards are updated to differentiate the preferred responses from the rejected ones.

Helpful and harmless data We use the Helpful and Harmless (Bai et al., 2022) preference data to learn two reward models, respectively. We obtain 161k training samples and 9k test samples after filtering the (prompt, response) pairs to a maximum token length of 2024; 3/4-th are for helpfulness while the remaining 1/4-th are for harmlessness. We further set 5% of the training data for validation.

Hyper-parameters We initialize all reward models with LLaMA-7B (Touvron et al., 2023). We fine-tune for 2 epochs with a mini-batch size of 64. We use Adam with a learning rate of 10^{-6} and a cosine learning rate scheduler (with 10% of the epoch used for warmup). We set weight decay to 0.1 and the gradient clipping maximum norm to 1. We utilize 16-bit (mixed) precision training and gradient checkpointing. Apart from this, we use the default hyper-parameters in the Transformers library (Wolf et al., 2020). We validate after every 10% of the epoch and save the best model.

B POLICY GRADIENT

We are interested in the policy gradients $\partial_\theta C_i(\theta)$'s. Note that while computing gradients with respect to the parameters of a distribution in an expectation, we can use the log-derivative trick:

$$\begin{aligned}\partial_\theta \mathbb{E}_{x\sim p_\theta(\cdot)}\left[f\left(x\right)\right] \;=\; \int dx f\left(x\right)\partial_\theta p_\theta\left(x\right) \;=\; \int dx f\left(x\right)\frac{p_\theta\left(x\right)}{p_\theta\left(x\right)}\partial_\theta p_\theta\left(x\right) \\ =\; \int dx f\left(x\right)p_\theta\left(x\right)\partial_\theta \log p_\theta\left(x\right) \;=\; \mathbb{E}_{x\sim p_\theta(\cdot)}\left[f\left(x\right)\partial_\theta \log p_\theta\left(x\right)\right].\end{aligned}$$

Applying the above to the policy gradient $\partial_\theta C_i(\theta)$ yields:

$$\partial_\theta C_i\left(\theta\right) \;=\; \partial_\theta \mathbb{E}_{\substack{(x,\cdot)\sim p(\cdot)\\ y\sim\pi_\theta(\cdot|x)}}\left[c_i\left(y|x\right)\right] \;=\; \mathbb{E}_{\substack{(x,\cdot)\sim p(\cdot)\\ y\sim\pi_\theta(\cdot|x)}}\left[c_i\left(y|x\right)\partial_\theta \log \pi_\theta\left(y|x\right)\right], \tag{12}$$

where $c_i(y|x) = b_i - r_i(y|x)$. This is the simplest form of the policy gradient and can be estimated as Monte Carlo averages. We refer readers to Schulman et al. (2016) for a review of other estimates.