Robust Chain of Thoughts Preference Optimization

Arash Ahmadian[†] **Eugene Choi**[†] **Matthieu Geist**

Olivier Pietquin **Mohammad Gheshlaghi Azar**

Abstract

Learning from human preferences has become the dominant paradigm in RL fine-tuning of large language models (LLMs). In particular human preferences are often distilled in the form of a reward model. Then this reward model is used through online RL methods to fine-tune the LLM. Alternatively offline methods like direct preference optimization (DPO) and Identity Preference Optimization (IPO) use contrastive losses to optimize the LLM directly by increasing the gap between the log-likelihoods of preferred and dis-preferred completions. Despite their success, these methods all suffer from a fundamental problem that their optimal solution highly depends on (and heavily optimized for) the behavior policy that has generated the completions of the preferences dataset. Therefore the solution of the existing methods may be prone to out-of-distribution (OOD) tasks where the validation dataset is significantly different from the behavior policy. Here we address this challenge by proposing Robust Chain of Thoughts Optimization (RoCoTO) of preferences, a practical and mathematically principled offline framework for reinforcement learning from human feedback that is completely robust to the changes in the behavior model. The key idea of RoCoTO is to cast the problem of learning from human preferences as a self-improving chain of thoughts (CoT) process in which the goal is to learn a policy that is nearly perfect in the sense that its generations can be only minimally improved through the best self-improving CoT model. We show that this idea can be mathematically expressed in terms of a min-max optimization objective that aims at joint optimization of chain-of-thoughts policy and the main generative policy in an adversarial fashion. The solution for this joint optimization problem is independent of the behavior policy and thus it is robust to the changes in the behavior model. We then show that this objective can be re-expressed in the form of a non-adversarial IPO (DPO)-style (offline) loss which can be optimized using standard supervised optimization techniques at scale without any need for reward model and online inference. We show the effectiveness of RoCoTO in solving TL;DR summarization task and show its superiority to the baseline IPO and DPO when evaluated on OOD XSUM.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) [Christiano et al., 2017] has rapidly become a standard method to align Large Language Models (LLMs). One of the main practical issues that all the prominent existing RLHF methods (offline or online) [Ouyang et al., 2022, Rafailov et al., 2023, Azar et al., 2023, Zhao et al., 2023b, Ahmadian et al., 2024] encounter is that their optimal solution heavily depends on the distribution used to generate the preference data (*behavior policy*) [Munos et al., 2023, Azar et al., 2023]. This makes the existing RLHF methods prone to out-of-distribution (OOD) overfitting [Li et al., 2024, Kirk et al., 2024] where the evaluation distribution is significantly different

[†]Equal contribution first co-authors {eugene,arash}@cohere.com.

17th European Workshop on Reinforcement Learning (EWRL 2024).

from that of the behavior policy. Also whenever the base/SFT models significantly differ from the behavior policy, the dependency of the RLHF solutions on the behavior policy makes the preference dataset and reward model less useful [Gao et al., 2022] as RLHF may undo the SFT/pretraining.

To address this challenge, we introduce an alternative approach for aligning LLMs from human preferences based on more principled and robust foundations. Our goal is to find a solution which is robust to the changes in the preference dataset, meaning, changes in the distribution from which the completions are sampled does not affect the final outcome of learning significantly. To achieve this goal we exploit the concept of self-improving chain of thoughts [Wei et al., 2023] language models. By self-improving chain of thoughts LLM we imply a model that given a completion (thought) it can improve it recursively at every turn of inference. This is a special case of general chain-of-thoughts model which generate the full completions as a sequence of (possibly different) thoughts. Our **Ro**bust **C**hain **o**f **T**houghts **O**ptimization (RoCoTO) of preferences consists in two back-to-back optimization processes:

In-Context Chain of Thoughts (CoT) Preference Optimization: The core idea is to learn an in-context self-improving CoT model (from now on, an LLM will be considered as equivalent to a distribution or policy π from which we can sample completions $y \sim \pi(\cdot|x)$, where x is the context or prompt.): given *in-context* completion y, the CoT model outputs an *improved* distribution $\pi(\cdot|y, x)$ from which sampled completions are most preferred to completion y according to the human preference model p. It turns out, as explained later, that this problem, in its KL-regularized form, can be expressed as a well-defined preference optimization problem and can be solved analytically. Furthermore, the solution can be estimated through a supervised direct preference optimization scheme similar to the approach used by Rafailov et al. [2023] or Azar et al. [2023].

Robust Preference Optimization of Generative Model: The next step is to exploit the CoT policy learned in the previous step to learn a model that generates the best completions given user prompts. The key idea here is that the best generative policy can be identified as a policy that generates completions which can only be minimally improved using the optimal self-improvement CoT policy π derived in step 1. This goal can be achieved by minimizing the objective of the step 1 with respect to the generative policy of *in- context* completions y. Similar to step 1 this problem again in its KL-regularized form can be solved analytically in terms of optimal CoT policy $\pi(\cdot|y)$. More significantly we show that the solution for step 1 and step 2 can be estimated jointly through a single supervised direct preference optimization scheme using only a dataset of annotated pair-wise completions. Thus, one can solve both for CoT policy and the generative policy through minimizing the supervised learning objective of RoCoTO. This solution, unlike the solutions of existing RLHF methods, is independent of the behavior policy and thus it is robust to its changes.

As using the CoT model in RoCoTO is a significant departure from the existing paradigm for RLHF we provide a high-level motivation for it in Sec. 2. We then formalize our objective for RoCoTO in Sec. 3 through which one can optimize jointly for both CoT policy and robust generative policy through optimizing an adversarial min-max objective. In Sec. 4 we present our main algorithmic/mathematical contribution: we prove that the preference probability p can be expressed in terms of the log-likelihoods of the optimal CoT policy and the log-likelihoods of the optimal robust generative policy. This theoretical finding is the key result for RoCoTO: solving this system of equations through least-squares regression provide us with the practical supervised RoCoTO objective that solves for both CoT policy and robust generative policy through a single supervised objective without any need for reward model or online inference. In Sec. 5 we further illustrate our argument on robustness of RoCoTO by providing an in-depth analysis of the solution of RoCoTO and other direct preference optimization methods. We also showcase/analyse the robustness of RoCoTO on a simple synthetic example. Finally in Sec. 7 we conduct large scale experiments on training LLMs with RoCoTO both on in-distribution and OOD summarization tasks and we compare the results with those of standard baselines.

2 Using Preference Optimization for Learning CoT Policy

One might ask why learning the self-improving CoT model can be useful for learning a good policy? In other words, why can we not directly train the the generative zero-shot policy without resorting to learning the self-improving CoT policy? To answer this question, we start by considering a more fundamental question:

What is natural to learn from human preferences?

We can learn improving the completions from the less preferred completions towards the more preferred completions as this information exist in every preference pair. In other words, we can learn how to improve the completions such that it is more preferred by humans. This is arguably a more natural learning task given the human preferences than directly learning the highest preferred completion by humans (which is what standard RLHF methods attempt) as the best answer is very unlikely to be in our completion dataset, especially when the space of possibilities in the entirety of human language, and in particular when the completions are generated from some LLM, which is subpar to humans. Instead it is more natural to learn that given a query and a completion y what would be the improved completion upon y, i.e., learn the model that aims at improving the output of LLM through a self-improving chain of thoughts (CoT) process to gradually improve the completions as a sequence of thoughts. In this case, if our model has captured the underlying rules of human preference then it can use that to improve the subpar completions towards the best completions.

The existing CoT-based pipelines either use SFT and behavior-cloning to train CoT [Liu et al., 2023] or they rely on in-context learning ability of pretrained/SFT LLMs [Bai et al., 2022, Wei et al., 2023]. In the following we show how we can jointly train the CoT policy *optimally* alongside its corresponding robust zero-shot policy from pair-wise preferences.

3 RoCoTO Objective

We start by introducing some notations required for establishing our theoretical results.

Notations. Let x and y denote a context and a completion drawn from the space of all possible contexts $\mathcal{X}$ and all possible completions $\mathcal{Y}$, respectively. The large language model (LLM) is represented by the probability distribution (policy) π where $\pi(y|x)$ denotes the probability of the completion y given the context x. In the remainder of this article we consider three variants of this baseLLM, the trainable model π_{train} (for which we use the short-hand notation π), the reference model π_{ref} and the behavior model μ from which the completions in pair-wise preference dataset is sampled. We also introduce the self-improving CoT $\pi(y'|y, x)$ as a model that using a context x and in-context completion (thought) y aims at improving y to a better completion y'. Similar to baseLLM we can define a reference model $\pi_{\text{ref}}(y'|y, x)$ also for CoT model. Let $\mathcal{D} = \{x, y_1, y_2\}$ be a dataset of contexts and completions where y_1 and y_2 are drawn independently from $\mu(\cdot|x)$. We then present every pair y_1, y_2 to human annotators who express preferences for one of the completions, denoted as $y_w \succ y_l$ where y_w and y_l denote the preferred and dispreferred actions amongst $\{y_1, y_2\}$ respectively. We then write true human preference $p(y_1 \succ y_2|x)$ the probability of y_1 being preferred to y_2 knowing the context x. The probability comes from the randomness of the choice of the human we ask for their preference. So $p(y_1 \succ y_2|x) = \mathbb{E}_h[\mathbb{I}\{h \text{ prefers } y_1 \text{ to } y_2 \text{ given } x\}]$, where the expectation is over humans h.

Consider a reference policy π_{ref}, and a real positive regularisation parameter $\beta \in \mathbb{R}_+^*$. Then, we define Robust Chain of Thoughts Optimisation objective (RoCoTO) as

$$J^* = \min_{\pi_1} \max_{\pi_2} \mathop{\mathbb{E}}_{\substack{x\sim\rho \\ y_1\sim\pi_1(.|x) \\ y_2\sim\pi_2(\cdot|y_1,x)}} \Big[p(y_2 \succ y_1|x) - \beta D_{\text{KL}}(\pi_2||\pi_{\text{ref}}|y_1, x) + \beta D_{\text{KL}}(\pi_1||\pi_{\text{ref}}|x) \Big], \quad (1)$$

in which the KL-regularization terms are defined as $D_{\text{KL}}(\pi_2||\pi_{\text{ref}}|y_1, x) = \text{KL}(\pi_2(\cdot|y_1, x)||\pi_{\text{ref}}(\cdot|y_1, x))$ and $D_{\text{KL}}(\pi_1||\pi_{\text{ref}}|x) = \text{KL}(\pi_1(\cdot|x)||\pi_{\text{ref}}(\cdot|x))$.

In nutshell this objective aims at **(i)** finding the best Chain of Thoughts (CoT) policy π_2^* that takes every $y_1 \sim \pi_1$ and improves it optimally with respect to the preference distribution p, i.e., the improved policy is most preferred to y_1, while keeping π_2^* close to the reference policy π_{ref}, **(ii)** minimizing the same objective to find the best (robust) policy π_1^* for which the generated completions are nearly perfect. So they can be only minimally improved by the optimal CoT model π_2^*.

4 Offline Solution for Optimizing RoCoTO Objective

Eq. (1) is a non-trivial optimization problem that often requires solving a two-stage adversarial optimization problem through the game-theoretic approaches, which are often challenging and difficult to scale up, [see e.g., Munos et al., 2023, Rosset et al., 2024, Calandriello et al., 2024, for

how we can use game-theoretic approaches/objectives to train LLMs]. Here, inspired by Rafailov et al. [2023], Azar et al. [2023] we aim at casting this complex optimization objective as a standard supervised learning problem that can be solved at scale given an offline pairwise preference dataset. To derive a practical algorithm for `RoCoTO` we first notice that the inner-maximization in the objective function of Eq. (1) can be solved in closed form as follows:

$$\pi_2^*(y_2|y_1,x) = \frac{\exp\left(\frac{p(y_2 \succ y_1|x)}{\beta}\right)\pi_{\text{ref}}(y_2|y_1,x)}{Z^*(y_1,x)}, \tag{2}$$

where $Z^*(y_1,x)$ is the normalization factor. One can easily show that by plugging π_2^* in the objective function of Eq. (1) we obtain:

$$J^* = \min_{\pi_1} \mathop{\mathbb{E}}_{\substack{x\sim\rho \\ y_1\sim\pi_1(.|x)}} \left[\beta(\log(Z^*(y_1,x)) + D_{\text{KL}}(\pi_1||\pi_{\text{ref}}|x))\right]. \tag{3}$$

Now by solving Eq. (2) with respect to $p(y_2 \succ y_1|x)$ we obtain

$$p(y_2 \succ y_1|x) = \beta(\log(\pi_2^*(y_2|y_1,x)) - \log(\pi_{\text{ref}}(y_2|y_1,x)) + \beta\log(Z^*(y_1,x))). \tag{4}$$

Also using the convention $p(y_1 \succ y_1|x) = \frac{1}{2}$ we have

$$\frac{1}{2} = \beta(\log(\pi_2^*(y_1|y_1,x)) - \log(\pi_{\text{ref}}(y_1|y_1,x))) + \beta\log(Z^*(y_1,x)). \tag{5}$$

Now by collecting terms in Eq. (5) we obtain

$$\beta\log(Z^*(y_1,x)) = \beta(\log(\pi_{\text{ref}}(y_1|y_1,x)) - \log(\pi_2^*(y_1|y_1,x))) - \frac{1}{2}.$$

Thus the objective of Eq. (3) can be expressed in terms of $\log(\pi_2^*(y_1|y_1,x))$ (up to an additive and multiplicative constant) as follows:

$$J^* \propto \min_{\pi_1} \mathop{\mathbb{E}}_{\substack{x\sim\rho \\ y_1\sim\pi_1(.|x)}} \left[\log\left(\frac{\pi_{\text{ref}}(y_1|y_1,x)}{\pi_2^*(y_1|y_1,x)}\right) + D_{\text{KL}}(\pi_1||\pi_{\text{ref}}|x))\right].$$

Solving this objective with respect to π_1 we obtain:

$$\pi_1^*(y|x) = \frac{\frac{\pi_{\text{ref}}(y|x)}{\pi_{\text{ref}}(y|y,x)}\pi_2^*(y|y,x)}{Z^*(x)} \tag{6}$$

where $Z^*(x)$ is the normalization factor. Again by taking the logarithm from both side we obtain

$$\log(\pi_2^*(y|x)) = \log\left(\frac{\pi_{\text{ref}(y|x)}}{\pi_{\text{ref}}(y|y,x)}\pi_1^*(y|y,x)\right) - \log(Z^*(x)).$$

Now by collecting terms in Eq. (4) and solving for $\log(\pi_2^*(y_2|y_1,x)))$ we obtain

$$\log(\pi_2^*(y_2|y_1,x)) = \frac{p(y_2 \succ y_1|x)}{\beta} - \log(Z^*(y_1,x))) - \log(\pi_{\text{ref}}(y_2|y_1,x)) \tag{7}$$

Now by plugging Eq. (2) into Eq. (6) we deduce

$$\pi_1^*(y|x) = \frac{\exp\left(-\log(Z^*(y,x))\right)\pi_{\text{ref}}(y|x)}{Z^*(x)}.$$

Solving this equation with respect to $\log(Z^*(y,x))$ implies

$$\log(Z^*(y,x)) = \log(\pi_{\text{ref}}(y|x)) - \log(\pi_1^*(y|x)) - \log(Z^*(x)). \tag{8}$$

Combining Eq. (7) and Eq. (8) we have for any y_1 and y_2

$$\frac{p(y_2 \succ y_1|x)}{\beta} - \log\left(\frac{\pi_2^*(y_2|y_1,x)}{\pi_{\text{ref}}(y_2|y_1,x)}\right) = \log\left(\frac{\pi_{\text{ref}}(y_1|x))}{\pi_1^*(y_1|x)}\right) - \log(Z^*(x)),$$
$$\frac{p(y_1 \succ y_2|x)}{\beta} - \log\left(\frac{\pi_2^*(y_1|y_2,x)}{\pi_{\text{ref}}(y_1|y_2,x)}\right) = \log\left(\frac{\pi_{\text{ref}}(y_2|x))}{\pi_1^*(y_2|x)}\right) - \log(Z^*(x)).$$

Subtracting these two Equations and collecting terms leads to our *key result* in which we express the preference p in terms of the optimal CoT policy π_2^* and the optimal robust zero-shot policy π_1^*.

$$p(y_2 \succ y_1|x) = \frac{1}{2} + \frac{\beta}{2}\Bigg[\log\left(\frac{\pi_2^*(y_2|y_1,x)}{\pi_{\text{ref}}(y_2|y_1,x)}\right) - \log\left(\frac{\pi_1^*(y_1|x)}{\pi_{\text{ref}}(y_1|x))}\right) - \left(\log\left(\frac{\pi_2^*(y_1|y_2,x)}{\pi_{\text{ref}}(y_1|y_2,x)}\right) - \log\left(\frac{\pi_1^*(y_2|x)}{\pi_{\text{ref}}(y_2|x))}\right)\right)\Bigg].$$

So we can enforce this equation for all y_1 and y_2 through following ℓ_2 loss:

$$L(\pi) = \mathop{\mathbb{E}}_{\substack{y_1,y_2\sim\mu(\cdot|x)\\ x\sim\rho}}\Bigg[p(y_2 \succ y_1|x) - \frac{1}{2} - \frac{\beta}{2}\Bigg[\log\left(\frac{\pi(y_2|y_1,x)}{\pi_{\text{ref}}(y_2|y_1,x)}\right) - \log\left(\frac{\pi(y_1|x)}{\pi_{\text{ref}}(y_1|x))}\right) - \left(\log\left(\frac{\pi(y_1|y_2,x)}{\pi_{\text{ref}}(y_1|y_2,x)}\right) - \log\left(\frac{\pi(y_2|x)}{\pi_{\text{ref}}(y_2|x))}\right)\right)\Bigg]\Bigg]^2 \tag{9}$$

Using the standard properties of ℓ_2-norm to replace $P(y_2 \succ y_1|x)$ with $\mathbf{1}(y_1 \succ y_2|x)$, as $P(y_2 \succ y_1|x) = \mathbb{E}(\mathbf{1}(y_1 \succ y_2|x))$, in the objective of Eq. (9) allows us to derive the following sample loss for RoCoTO:

$$\widehat{L}(\pi) = \mathop{\mathbb{E}}_{(y_l,y_w,x)\sim\mathcal{D}}\Bigg[\beta\Bigg[\log\left(\frac{\pi(y_w|y_l,x)}{\pi_{\text{ref}}(y_w|y_l,x)}\right) + \log\left(\frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x))}\right) - \left(\log\left(\frac{\pi(y_l|y_w,x)}{\pi_{\text{ref}}(y_l|y_w,x)}\right) + \log\left(\frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x))}\right)\right)\Bigg] - 1\Bigg]^2 \tag{10}$$

The following pseudo-code can be used to train the LLM policy using RoCoTO objective:

Algorithm 1 Sampled RoCoTO

Require: Dataset $\mathcal{D}$ of prompts, preferred and dis-preferred generations x, y_w and y_l, respectively. A reference policy π_{ref} and a training policy π_θ

1: Initialize $\pi_\theta = \pi_{\text{ref}}$
2: **while** true **do**
3: Sample a minibatch $B \in \mathcal{D}$
4: Estimate $\nabla_\theta \widehat{L}(\pi_\theta)$ from Eq. (10) using minibatch B as the dataset
5: Update π_θ using $\nabla_\theta \widehat{L}(\pi_\theta)$ using a standard optimizer
6: **end while**
7: **return** π_θ

5 Robustness of RoCoTO

We provide an in depth comparison between RoCoTO and prior work on direct preference optimization in terms of their robustness to the behavior policy μ. In particular we consider as a point of reference DPO and IPO for which we have a good understanding of the underlying mathematical foundation.

In the case of both IPO and DPO the analytical solution is already well-established and analyzed for both algorithms [Azar et al., 2023, Rafailov et al., 2023, Tang et al., 2024]. In particular the optimal solution for both IPO and DPO can be expressed explicitly in terms of the soft-max of the expected preference as follows [Azar et al., 2023]:

$$\pi^*(y|x) = \frac{\exp\left(\frac{\mathbb{E}_{y'\sim\mu(\cdot|x)}(\Psi(p(y\succ y'|x)))}{\beta}\right)\pi_{\text{ref}}(y|x)}{Z^*(x)}, \tag{11}$$

with the choice of $\Psi = I(\cdot)$ and $\Psi = \sigma^{-1}(\cdot)$ for IPO and DPO, respectively, where σ^{-1} denotes the inverse-sigmoid (logit function). So, based on (11), we can see that the solution for both IPO and

DPO has strong dependency on μ in the form of expected preference under the distribution μ. Thus it may not be robust to changes in μ. This dependency on μ can be especially problematic when we evaluate the model on out-of-distribution tasks where the desired behavior is very different from μ and the expected preference under the distribution μ is not a good measure of performance. RoCoTO solution on the other hand has no dependency on the behavior policy μ: from (2) we observe that the optimal CoT policy π_2^* is independent of μ and, unlike DPO and IPO cases, is expressed in terms of softmax of $P(y_2 \succ y_1|x)$ for any pair of completions (y_1, y_2). Also the 0-revision policy π_1^* is also completely independent of μ as it is evident from (6) (i.e., it is proportional to $\pi_1^*(y|y, x)$ which itself is independent of μ). Thus, from a mathematical point of view, RoCoTO provides a robust solution for the problem of direct preference optimization that does not depend on the behavior policy μ.

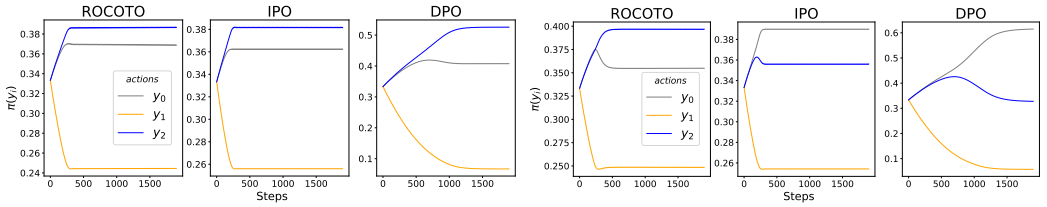
To illustrate further the differences between RoCoTO and DPO/IPO with regard to robustness to μ we consider a simple bandit example. For simplicity we assume there is no context x, i.e., we are in a standard bandit setting. We consider the simple case where we have 3 actions (completions) y_0, y_1 and y_2, for which the preference model is given as follows:

$$P = \begin{pmatrix} 0.5 & 0.99 & 0.3 \\ 0.01 & 0.5 & 0.25 \\ 0.7 & 0.75 & 0.5 \end{pmatrix},$$

in which $p(y_i \succ y_j) = P_{ij}$. In this case y_2 is clearly the preferred outcome as it dominates both y_1 and y_0 by probability larger than 0.5. On the other hand, if we only consider preference with respect to y_1 then arm y_0 is a better outcome as it is preferred to y_1 with higher probability than y_2. Therefore, if a preference optimization method is robust to changes in μ one might expect that the optimal solution should be independent of the frequency of y_2 in the preference dataset (i.e., $\mu(y_2)$).

To test this hypothesis we consider two synthetic dataset of actions generated from distributions μ_0 and μ_1: We set μ_0 to be a uniform behavior policy ($\mu_0(y_0) = \mu_0(y_1) = \mu_0(y_2) = \frac{1}{3}$) and μ_1 skewed towards y_1 ($\mu_1(y_1) = 0.7, \mu_1(y_0) = \mu_1(y_2) = 0.15$). We then generate a dataset of 10000 pairs from μ_0 and μ_1 and rate them according to the preference model p (for any pair (y_1, y_2) we assign the preference by sampling from $p(y_1 \succ y_2)$, that is y_1 is preferred to y_2 with probability $p(y_1 \succ y_2)$). This provides us with two dataset of rated completions $\mathcal{D}_0$ and $\mathcal{D}_1$ for μ_0 and μ_1. We then use these two datasets to train the policy π using RoCoTO, DPO and IPO using a simple Adam optimizer. In the case of IPO and DPO we optimize only the 0-revision policy $\pi(y)$ where as for RoCoTO we also optimize the CoT policy $\pi(y|y')$ as well. We set the regularization constant β for all methods to 1. We consider a uniform distribution $\pi_{\text{ref}}(y) = 1/3$ for all algorithms and all ys. In the case of RoCoTO we set the CoT reference policy $\pi_{\text{ref}}(y|y') = 1/3$ for all y and y'.

We observe that in the case of using uniform μ_0 as a behavior policy all methods do the right thing and their policies converge to solutions in which y_2 dominates y_1 and y_0 (Fig. 1a). However, when we use the behavior policy μ_1 which is skewed towards y_1, both DPO and IPO converge to a solution in which y_0 dominates y_1 and y_2, while the policy of RoCoTO remains intact (Fig. 1b). Notice that the RoCoTO policy is slightly different in both cases. This is to be expected, we are in a finite data setting, and the sampling distribution will have some influence on the empirical preference model (defining the empirical solution of RoCoTO).



(a) RoCoTO vs IPO and DPO for uniform behavior μ_0. (b) RoCoTO vs. IPO and DPO for skewed behavior μ_1.

Figure 1: Learned action probabilities for the synthetic example. RoCoTO always chooses the correct arm regardless of skew in the behaviour policy μ, while both IPO and DPO are effected by the skew as portrayed in Fig. (1b).

6 Related works

Our work lies in offline preference optimization, a vivid area of research since the introduction of DPO [Rafailov et al., 2023]. Some of the core concepts of this research topic was generalized and formalized by Azar et al. [2023]. In particular they characterized the underlying optimal solution for a generic preference optimization objective and introduced IPO for adressing some of the related shortcomings of DPO. SLiC-HF [Zhao et al., 2023a] was introduced around the same time, from a less RL-centric point of view. All these approaches have been abstracted later by Tang et al. [2024], the general recipe being to build a contrastive loss function from a convex classification function and to make use of the analytical solution of the RL problem to learn directly the policy. A common underlying assumption is that the related RL problem is KL-regularized. This has been generalized to more general f-divergences by Wang et al. [2023]. These are just a few among many works on direct alignment. However, they all share the fact of not considering CoT policies, contrary to RoCoTO. This has a strong incidence on the related solution concept, making RoCoTO the sole direct alignment approach being robust to the sampling distribution μ, as showcased in Sec. 5.

Offline preference optimization was introduce as an alternative to more classic RLHF approaches, such as PPO [Schulman et al., 2017, Ouyang et al., 2022] or more generally policy-gradient-based approaches [Roit et al., 2023, Ahmadian et al., 2024]. These methods require training a so-called reward model on a preference dataset, usually with a Bradley-Terry model [Bradley and Terry, 1952]. The reward model is then used to finetune the LLM via online RL, requiring many generations from the model. This reward model shares the common issue of DPO and other direct preference alignment methods, it is dependent to the sampling distribution μ used for constructing the preference dataset, contrary to RoCoTO. Moreover, classic RLHF is online, while RoCoTO is offline.

Some similarities also exist between RoCoTO and Nash-MD [Munos et al., 2023]. Indeed, if in Eq. 1 we replace the CoT policy $\pi_2(\cdot|y, x)$ by a classic policy $\pi_2(\cdot|x)$, then we obtain the saddle-point optimization problem that Nash-MD solves. However, considering a CoT policy is a core contribution of our work, and it is not anecdotal. From a technical viewpoint, this is critical for simplifying the minimax problem of Eq. (1) and obtaining a simple offline optimization problem. NashMD adapts algorithms from the game-theory literature and can only be solved online with all the stability issues of online methods and large inference costs. Finally, even though the Nash equilibrium of Nash-MD does not depend on the sampling distribution μ, it relies on a learned reward function, with the possible associated caveats mentioned earlier, which is not the case of RoCoTO.

Our work is also obviously related to the concept of chain of thoughts [Wei et al., 2023, Yao et al., 2024], self-improvement [Huang et al., 2022] and self-refining LLMs [Madaan et al., 2024]. However, it is very often used as a way of prompting a model to obtain better results, and less often as a component of a learning paradigm [Liu et al., 2023, Huang et al., 2022]. To our best knowledge, we propose the first approach combining CoT self-improving LLMs and offline preference optimization, moreover in a theoretically grounded manner and showing the robustness to μ.

7 Experiments

Setup. In our experiments, we consider the offline direct preference optimization setup to learn from human preferences [Rafailov et al., 2023]. In the offline setting, the goal is to train the LLM policy directly from a dataset $\mathcal{D}$ of pairwise completions (y_l, y_w) sampled from a behavior policy μ and annotated by human raters without using a reward model or online inference/RL. We empirically test the effectiveness of RoCoTO against two offline preference learning methods, namely Direct Preference Optimisation (DPO) [Rafailov et al., 2023] and Identity Preference Optimisation (IPO) [Azar et al., 2023] as baselines. We make this choice since both these baselines, like RoCoTO, are mathematically well-grounded offline methods. Also, they have been widely used by the AI community in solving different language tasks [Tunstall et al., 2023, Wallace et al., 2023, Yuan et al., 2024, Pang et al., 2024, Lin et al., 2023].

Implementation details. RoCoTO trains simultaneously both the standard (0-revision) policy and CoT policy used for revising the completions of models through a single optimization process. We only use a single LLM to represent both zero-revision and CoT policy. The optimal zero-revision policy can be seen as a policy that can be least improved using the optimal CoT model. So to get the best completions from RoCoTO we first generate completions in 0-revision (0-rev.) mode and then

we improve these completions with the CoT model. We call the revised outputs 1-revision (1-rev.) completions. We report results in both 0-rev. and 1-rev. cases. For IPO and DPO we also report results on 0-rev. and 1-rev. For revising the completions we use IPO and DPO in in-context learning mode with the 0-rev. completions used as contexts. In the case of DPO we use the same loss and hyper-parameters used by [Rafailov et al., 2023]. For IPO since the original paper hasn't provided the hyper-parameters we used a set of hyper-parameters (i.e., learning rate and regularization constant β) from the range of hyper-parameters that was working. Furthermore we noticed that the performance of IPO was not affected significantly by the choice of these hyper-parameters. So no significant gain is expected by hyper-parameter tuning.

Datasets. We use the Reddit TL;DR Summarization dataset [Stiennon et al., 2020] as the main dataset for our experiments[†]. For training, there are 116k human-written instruction following examples with reference completions (SFT split) while there are 93k human-annotated preference pairs (Preference split). We also use the XSum dataset test split[†] [Narayan et al., 2018], which contains 11.5k total test examples to measure Out-of-Distribution (OOD) generalization.

Model Setup. We use LLaMA-7B as base model [Touvron et al., 2023] and a single $8\times$NVIDIA H100 node to conduct all LLaMA-based experiments. We first supervise fine-tune the model on the SFT split of the TL;DR dataset, before preference training and use the same π_{ref} for all preference training experiments. Below are details on the training recipe for the SFT and preference training stages.

Supervised-fine Tuning. In the SFT stage, we train for 2 epochs, using the AdamW optimizer [Loshchilov and Hutter, 2019], with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and 0.1 weight-decay. We use a cosine decay learning rate [Loshchilov and Hutter, 2017] with a peak value of 2×10^{-5} and 3% of all steps being warm-up steps. We use an effective batch-size of 64.

Preference Training. We initialize π with our SFT model (π_{ref}). All models were trained for 5 epochs on the TL;DR preference split using the same optimization setting of the AdamW optimizer as in the SFT stage. We used 150 warmup steps. We also use an effective batch-size of 128. For RoCoTO and IPO, we used $\beta = 0.01$ with a learning rate of 2×10^{-6}. For DPO, we used the common $\beta = 0.1$ with a learning rate of 1×10^{-6} and a constant learning rate schedule.

Evaluation. We use win-rates as computed by gpt-4-0613 [OpenAI, 2023] using the Alpacafarm framework [Dubois et al., 2024], as the main means for evaluation. We measure performance on both in-distribution and OOD examples at test time in the following manner: For the former, we compute winrates against gold reference completions from the test set of the TL;DR SFT split. For the latter, we measure win-rate against gold completions from the test set of the XSum dataset. In both settings, we use the first 1,024 samples from each of the test sets. To estimate the win rate more accurately with confidence intervals, we bootstrap 20 times with replacement from the 1,024 samples, each time using a sample size of 512. To sample from the CoT policy we first sample y from $\pi(\cdot|x)$. Then, using the same policy, we condition on y to sample from a CoT policy, that is $y' \sim \pi(\cdot|y, x)$. We refer to generations from $\pi(\cdot|x)$ as 0-revision (0-rev.) and generations from $y' \sim \pi(\cdot|y, x)$ as 1-revision (1-rev.) [Bai et al., 2022].

TL;DR Results. We test our models on the test set split of the TL;DR dataset in Table 1. For every model, we generate both 0-rev. and 1-rev. completions, and measure the models' win rate against the human-written gold reference summaries.

We observe that RoCoTO 1-rev. generates high-quality summaries with the highest win rate against the gold summaries, compared to 0-rev. and 1-rev. samples of the baseline methods, as well as RoCoTO 0-rev. Furthermore, we observe that RoCoTO 1-rev. improves upon RoCoTO 0-rev., outperforming every other method. However, DPO and IPO fail to generate an improved sample through the CoT step.

Table 1: In distribution results: GPT-4 win-rate against the gold completions on TL;DR test set.

RoCoTO		DPO		IPO	
0-rev.	1-rev.	0-rev.	1-rev.	0-rev.	1-rev.
$81.53 \pm (1.30)$	$\mathbf{84.66 \pm (1.03)}$	$78.01 \pm (1.17)$	$77.59 \pm (0.96)$	$83.71 \pm (0.99)$	$83.46 \pm (1.00)$

[†] https://github.com/openai/summarize-from-feedback
[†] https://huggingface.co/datasets/csebuetnlp/xlsum

We further measure the win rates of 0-rev., 1-rev. of RoCoTO and baselines against each other in Table 2. We show that the head-to-head win rates further demonstrate consistent results with those against the gold summaries.

Table 2: TL;DR test set head2head results

RoCoTO 1-rev. vs. RoCoTO 0-rev.	RoCoTO 1-rev. vs. IPO 1-rev.	RoCoTO 1-rev. vs. IPO 0-rev.
56.843	56.005	55.664

Out-of-distribution (OOD) Results. To assess robustness in an OOD setting, we test RoCoTO models trained with TL;DR preference dataset on the XSum test split in Table 3 [Narayan et al., 2018]. As in the TL;DR case, we observe that RoCoTO 1-rev. generates the highest win rate against the gold summaries, compared to 0-rev. and 1-rev. samples of the baseline methods, as well as RoCoTO 0-rev. Interestingly, we also observe that the OOD winrate of RoCoTO (both for 0-rev. and 1-rev.) is comparable to the in-distribution case, while both DPO and IPO suffer from a drop of performance.

Table 3: OOD results: GPT-4 win-rate against the GOLD completion on XSUM test set

RoCoTO		DPO		IPO	
0-rev.	1-rev.	0-rev.	1-rev.	0-rev.	1-rev.
$82.21 \pm (1.31)$	$\mathbf{84.76 \pm (1.10)}$	$69.49 \pm (1.30)$	$70.56 \pm (1.04)$	$81.04 \pm (1.30)$	$80.99 \pm (1.41)$

Finally in Table 4 we measure the win rates of 0-rev., 1-rev. of RoCoTO and baselines against each other in OOD setting which confirms the result in TL;DR setting.

Table 4: XSum test set head2head results

RoCoTO 1-rev. vs. RoCoTO 0-rev.	RoCoTO 1-rev. vs. IPO 0-rev.	RoCoTO 1-rev. vs. IPO 1-rev.
51.466	51.465	52.197

8 Discussion and Limitations

In this paper we have developed Robust Chain of Thoughts Optimization of Preferences (RoCoTO), a brand-new robust offline approach for learning from human preferences. We have proven mathematically and with illustrative examples, that unlike other prominent offline methods like DPO and IPO, the solution of RoCoTO is completely independent of the behavior policy μ and thus RoCoTO is completely robust to changes in μ.

Summary of results. We have tested RoCoTO on standard summarization tasks both on in-distribution and out-of-distribution (OOD) regimes. We have observed that in the OOD case RoCoTO outperforms both IPO and particularly the celebrated DPO by a clear margin in terms of win-rate against gold completions, while in the in-distribution case there is less difference between RoCoTO and the baselines. This is an expected behavior since in-distribution case the robustness aspect of the algorithm matters less. We have observed that although 0-revision generation of RoCoTO performs well, we observe a boost across the board by revising the generation through the CoT model (i.e. 1-revision generations). We postulate that although RoCoTO policy is optimized to generate outputs that are minimally improvable still improvement by the CoT model, which is optimized to improve the generation, can lead to better generations.

Future work and Limitations. In our work we used standard and relatively simple language tasks. In the future we would like to apply RoCoTO to more challenging multi-task benchmarks in which the existing RLHF methods often specialize to a specific set of tasks more represented in the dataset, whereas RoCoTO should be more resilient due to its robustness to behavior policy μ.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs, 2024.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, Rishabh Joshi, Zeyu Zheng, and Bilal Piot. Human alignment of large language models through online preference optimisation, 2024.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization, 2022.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve, 2022.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity, 2024.

Ziniu Li, Tian Xu, and Yang Yu. Policy optimization in rlhf: The impact of out-of-preference data, 2024.

Yong Lin, Lu Tan, Hangyu Lin, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Han Zhao, Yuan Yao, et al. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *arXiv preprint arXiv:2309.06256*, 2023.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. Chain of hindsight aligns language models with feedback, 2023.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback, 2023.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206.

OpenAI. Gpt-4 technical report, 2023.

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 amd Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv*, 2023.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Léonard Hussenot, Orgad Keller, et al. Factually consistent summarization via reinforcement learning with textual entailment feedback. *arXiv preprint arXiv:2306.00186*, 2023.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, 2017.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment, 2024.

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.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. *arXiv preprint arXiv:2311.12908*, 2023.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. *arXiv preprint arXiv:2309.16240*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models, 2024.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. SLiC-HF: Sequence likelihood calibration with human feedback. *arXiv*, 2023a.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. SLiC-HF: Sequence likelihood calibration with human feedback. *arXiv*, 2023b.

A Experimental Details

We provide the prompt templates used for training and evaluations in section 7.

A.1 Prompt Templates

A.1.1 TL;DR

0-revision:

```
Below is a reddit POST and the corresponding SUBREDDIT and TITLE. Write a
    both precise and concise summary of the contents of the POST.

SUBREDDIT: ${subreddit}
TITLE: ${title}
POST: ${post}
TL;DR:
```

1-revision:

```
Below is a reddit POST and the corresponding SUBREDDIT, TITLE, and an
    EXAMPLE SUMMARY. Write a both precise and concise summary of the
    contents of the POST.

SUBREDDIT: ${subreddit}
TITLE: ${title}
POST: ${post}
EXAMPLE SUMMARY: ${example_summary}
TL;DR:
```

A.1.2 XSum

0-revision:

```
Below is a news ARTICLE and the corresponding ID and TITLE. Write a both
    precise and concise summary of the contents of the ARTICLE.

ID: ${id}
TITLE: ${title}
ARTICLE: ${article}
TL;DR:
```

1-revision:

```
Below is a news ARTICLE and the corresponding ID, TITLE, and an EXAMPLE
    SUMMARY. Write a both precise and concise summary of the contents of
    the ARTICLE.

ID: ${id}
TITLE: ${title}
ARTICLE: ${article}
EXAMPLE SUMMARY: ${example_summary}
TL;DR:
```

A.2 Example Summaries

A.2.1 TL;DR

Post	*I have a horrible caffeine addiction, and I don't like sacrificing any of my daily calories for coffee. I used to drink 5-6 Diet Dr. Peppers a day, but I have switched to almost exclusively drinking only water most days. I do have a Diet Dr. Pepper occasionally, and I know that technically it's okay to probably have one a day, but I honestly just feel better all around when I drink only water. But if I don't have some caffeine in the morning, I will be writhing in the floor in pain with a skull-pounding headache by mid-day. I know. Ridiculous, right?* *So, I went to the vitamin aisle the other day in search of a supplement with a little caffeine and the only thing I found was One A Day Adult Vitacraves Gummies with Energy Support. Here's the website with [nutritional info] One serving is 20 calories and has 90 mg of caffeine. There's also some vitamin B6, B12, and folic acid in there that supposedly helps turn food into energy. I have no idea if that's true. I've been taking them a couple days, and so far I do feel more energized and I haven't had any headaches. The issue is that one bottle is around $9 and there are only 60 in a bottle. You have to take 4 (4!) a day, which is very intriguing considering the brand name and all. So it'll cost me around $20 a month. Which is admittedly cheaper than what I used to spend on Diet Dr. Peppers, but still. I was just curious if anyone else had a better option for caffeine in a supplement or vitamin?*
GOLD	The vitamin with caffeine I'm taking is on the expensive side, and I'm a self-proclaimed tightwad. Anyone have better options?
RoCoTO 1-rev.	I have a caffeine addiction and need a little bit of caffeine in my day. I'm taking One A Day Adult Vitacraves Gummies with Energy Support, but it's expensive ($20/month). Any other suggestions for a caffeine supplement or vitamin?
RoCoTO 0-rev.	I have a terrible caffeine addiction and need a little caffeine in my day. I found a vitamin with caffeine in it, but it's pretty expensive. Any other suggestions for a caffeine supplement or vitamin?
IPO 1-rev.	I have a caffeine addiction and need some in my system every day to function, but I don't like sacrificing calories for coffee. I've found a supplement with caffeine, but it's expensive. Anyone have a better option?
IPO 0-rev.	I have a caffeine addiction and need some in my system every day to function, but I don't like sacrificing calories for coffee. I've found a supplement with caffeine, but it's expensive. Anyone have a better option?
DPO 1-rev.	I have a caffeine addiction and I need a supplement or vitamin that has caffeine in it to take a couple times a week so that I don't have a headache all day. Any recommendations?
DPO 0-rev.	I have a caffeine addiction and I need a supplement or vitamin that has caffeine in it to take a couple times a week so that I don't have a headache all day. Any recommendations?

A.2.2 XSum

Article	*A paper detailing its proposals focuses on the need to avoid a hard border. The government stresses there should be no physical infrastructure, such as customs posts, at the border, which has almost 300 crossing points. Critics say the proposals lack credible detail, with Labour deriding the plans for the border as "a fantasy frontier". The government's paper does not envisage CCTV cameras or number plate recognition technology at the border, or set back from it. Instead, the government is arguing for a wide-ranging exemption under which small and medium-sized businesses will not have to comply with any new customs tariffs. Northern Ireland is the only part of the UK that will share a land border with an EU state post-Brexit. The future management of that border is a highly sensitive issue and is one of three main priorities in UK-EU Brexit negotiations. . . . What is the customs union? Countries in the customs union do not impose tariffs - taxes on imports - on each other's goods. Every country inside the union levies the same tariffs on imports from abroad. So, for example, a 10% tariff is imposed on some cars imported from outside the customs union, while 7.5% is imposed on roasted coffee. Other goods - such as soap or slate - have no tariffs. The UK has said it is leaving the EU's customs union because as a member it is unable to strike trade deals with other countries.*
GOLD	The government has said there must be an "unprecedented solution" for the border between Northern Ireland and the Republic after Brexit.
RoCoTO 1-rev.	The UK wants to avoid a hard border between Northern Ireland and the Republic of Ireland after Brexit by proposing two customs models to the EU. The UK also dismisses the idea of a customs border in the Irish Sea.
RoCoTO 0-rev.	Brexit Secretary David Davis wants a limited transition period to implement new customs arrangements, including consideration of the "unique circumstances" of Northern Ireland and the Republic of Ireland. The UK is trying to prevent the need for physical customs posts along the Irish border.
IPO 1-rev.	Theresa May wants a 'seamless' border between Northern Ireland and the Republic of Ireland after Brexit - but the EU may not accept that as viable. The UK government has published a position paper setting out its ideas on the issue.
IPO 0-rev.	Theresa May wants a 'seamless' border between the Republic of Ireland and Northern Ireland after Brexit - but the EU may not accept that as viable. The UK government has published a position paper setting out its ideas on the issue.
DPO 1-rev.	the UK is proposing a "new customs partnership" or a "highly streamlined customs arrangement" to prevent a hard border between Northern Ireland and the Republic of Ireland, in the event of no deal being reached over Brexit.
DPO 0-rev.	the UK is proposing a "new customs partnership" or a "highly streamlined customs arrangement" to prevent a hard border between Northern Ireland and the Republic of Ireland, in the event of no deal being reached over Brexit.