KL-Regularized Reinforcement Learning for Generative Modelling is Designed to Mode Collapse

Anonymous authors
Paper under double-blind review

Abstract

Classical intuitions cast minimizing reverse KL as "mode seeking" and forward KL as "mass covering". In KL-regularized reinforcement learning, however, the regularizer determines *both* the target distribution's shape *and* the divergence being implicitly minimized, making its role more nuanced than simply inducing generic mode-seeking or mass-covering behaviour. Specifically, the target distribution is defined jointly by the reward function, the reference model, the type of regularizer, and the regularization strength. We show that under common settings—such as low regularization strength and equal verifiable rewards—both forward and reverse KL regularization tend to specify target distributions whose mass concentrates on a single high-reward region. Thus, the objective itself *by construction* induces diversity collapse, regardless of the policy optimization algorithm used.

Building on this perspective, we introduce a simple and scalable modification that rescales rewards to induce target distributions assigning substantial probability across *all* high-reward regions. This yields a principled objective that maintains high solution quality while achieving broad reward-mode coverage. Empirically, this approach improves post-training diversity and performance for Large Language Models and Chemical Language Models, and is effective with either forward or reverse KL regularization, while using either naively fails.

1 Introduction

Reinforcement Learning (RL) is the predominant method for post-training foundation models (Ouyang et al., 2022), and the primary way to train generative models in settings where the correct solution is not known *a priori*. At its core, this involves solving a regularized RL (contextual bandits) problem, where a policy is trained to maximize some external reward, while preserving "closeness" to a base policy (as to e.g. preserve coherence). Output diversity of the policy is crucial. In Large Language Models (LLMs), it drives engagement for tasks such as creative writing and free-form conversation. More generally, diversity underlies the generation of new knowledge, enabling the discovery of novel mathematical solutions (Romera-Paredes et al., 2024), cognitive science models (Castro et al., 2025), and novel algorithms and software (Surina et al., 2025; Novikov et al., 2025; Aygün et al., 2025). Furthermore, diversity reflects uncertainty over competing hypotheses, a property fundamental to scientific discovery (GX-Chen et al., 2025). Finally, diversity plays an important role *during training* to drive exploration such that the policy can find and converge to better solutions (Cui et al., 2025).

Yet, current empirical evidence suggests RL post-training improves quality at the cost of diversity (Kirk et al., 2023; Cui et al., 2025). As a response, a number of recent works set out to treat this ailment, with a variety of approaches including explicit diversity rewards (Li et al., 2025), changing the KL regularization (Wang et al., 2023), selecting diverse data (Lanchantin et al., 2025), and count-based exploration bonuses (Song et al., 2025).

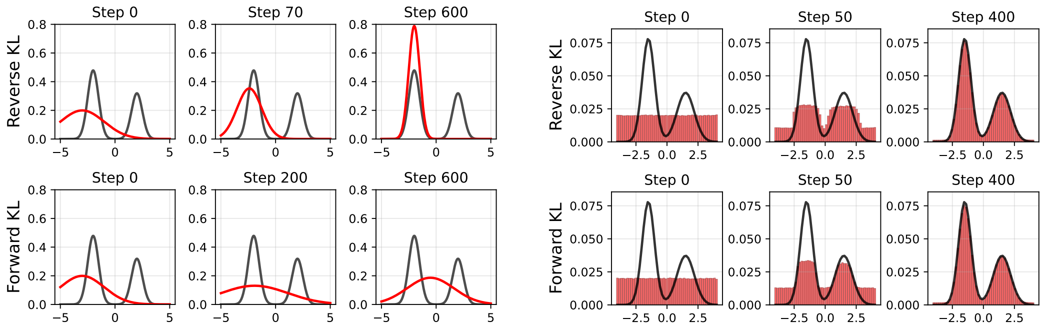
In this work, we take a step back to diagnose a more fundamental problem: *does the objective being optimized actually have a solution that is diverse?* We find that with current set-ups, the answer is often "no", even with unlimited compute, high quality data, and perfect optimization. We prove that under very commonly used settings (such as weak KL regularization with varied rewards, or *any* KL regularization if correct answers have the same rewards but vastly different reference policy supports), the globally optimal solution is often *by construction* unimodal. To accomplish this, we analyze KL-regularized RL through tools from variational inference (VI, Jordan et al. 1999; Ranganath et al. 2014) to find and dissect optimal policies for different choices of KL regularization.

Section 2 provides preliminaries about KL divergences and reminds the reader of the mode-seeking / mass-covering behaviour of minimizing reverse / forward KL at suboptimality. Section 3 studies KL-regularized reward maximization as implicitly minimizing a divergence between the current policy and a *target distribution*. Section 4 further analyzes the *shape* of this target solution. We focus particularly on how this distribution puts mass over high-reward regions—i.e. multimodality in terms of reward modes (Definition 3.5). This allows us to understand even if we perfectly minimize *any* divergence between the policy and the target distribution, the resulting policy will still be non-diverse if the *target distribution is defined to be unimodal*. Finally, Section 5 shows how one can directly construct the target distribution to cover all high-reward modes. We specify one such distribution which puts mass over all high-reward regions above a certain threshold, and show this requires only a small change to current algorithms. Each section is empirically supported with didactic simulations. Finally, we apply our method out of the box to LLMs and chemical language models and find that it works for complex, realistic scenarios.

The main contributions can be summarized as follows,

1. We analyze the role of reverse/forward KL regularization in RL as *both* defining the target distribution *and* the implicitly minimized divergence between policy and target.
2. We show the shape of the target distribution is determined by the regularizer, regularization strength, and relative reward and reference probability magnitudes. This has implications on how the target distribution puts mass over high-reward regions.
3. We show with typical hyperparameters, the target distribution is often constructed to put mass over a single high-reward region, making diversity collapse a natural consequence of correctly solving the regularized RL problem (as currently defined), regardless of algorithm.
4. We derive conditions required for broad reward mode coverage, and use this insight to construct a simple and theoretically principled RL algorithm (two-line pseudocode, Alg. 1) that puts uniform mass over *all* high-reward regions, without any external diversity signals.

2 The Kullback-Leibler (KL) Divergence



(a) Restrictive (Gaussian) approximation

(b) Flexible (categorical) approximation

Figure 1: Illustration of how the choice of approximate distribution family affects KL optimization. With a restrictive approximate distribution (e.g. 1D Gaussian with two parameters), KL exhibits the typical "mode seeking" and "mass covering" characteristics. This intuition does not necessarily hold for flexible distributions (e.g. independent categoricals, foundational models).

The Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951) measures the discrepancy between two probability distributions. In machine learning, it is commonly used in variational inference (VI), where minimizing the KL divergence enables a tractable variational distribution q to approximate an intractable posterior p (Jordan et al., 1999; Blei et al., 2017). Following Murphy (2012), we refer to $D_{KL}(q||p) = \mathbb{E}_q[\log q(y) - \log p(y)]$ as the *reverse KL divergence*, and $D_{KL}(p||q) = \mathbb{E}_p[\log p(y) - \log q(y)]$ as the *forward KL divergence*. Reverse KL is often described as "mode seeking", avoiding mass where p is small (Figure 1a, top), while forward KL is often described as "mass covering", putting mass anywhere p has mass (Figure 1a, bottom). These intuitions hold *if* the variational family is not sufficiently expressive and we can at best settle on an optimum with > 0 KL (Bishop & Nasrabadi, 2006; Murphy, 2012). With a flexible family, however, optimizing either KLs to the *global optimum* can well-approximate a complex posterior (Figure 1b).

3 KL-Regularized Reward Maximization

KL-regularized reward maximization aims to (i) maximize the expected value of a reward function $R : \mathcal{Y} \to \mathbb{R}$, mapping from samples to a scalar outcome (e.g. improve human preference), while (ii) keeping the policy π_θ close to a reference distribution π_{ref} (e.g. maintain grammatical coherence). The objective is $J(\pi_\theta) = \mathbb{E}_{\pi_\theta(y)}[R(y)] - \beta\, D\big(\pi_\theta, \pi_{\text{ref}}\big)$, where $D(\cdot,\cdot)$ denotes a divergence between the policy and reference distributions. For brevity, we consider the unconditional generation problem where the policy models distribution $\pi_\theta(y)$. The problem is the same in the case of conditional generation (e.g. question answering), where the objective is simply defined over the conditional distribution $\pi_\theta(y|x)$. *We do not deal with the sequential decision making setting*—commonly modelled using Markov Decision Processes (Puterman, 1994)—in this work. Nevertheless, the non-sequential setting is widely used when training generative models with RL.

In this section, we consider the *solution / target distribution* of KL-regularized reward maximization, i.e. the distribution which maximizes the objective. The central question is:

> *If we perfectly solve the regularized RL problem to its global optimum, what does the solution (policy) distribution look like?*

3.1 Solution of the Reverse KL Regularized Objective

The most common KL-regularized policy gradient objective uses the *reverse KL divergence*,

$$J_\beta(\pi_\theta) = \mathbb{E}_{\pi_\theta(y)}[R(y)] - \beta\, D_{KL}\big(\pi_\theta || \pi_{\text{ref}}\big)\,. \tag{1}$$

A number of previous works have discussed the solution / optimal distribution of this optimization problem (Korbak et al., 2022; Go et al., 2023; Rafailov et al., 2023; Azar et al., 2024; Zhang & Ranganath, 2025), which we note again below (see Appendix B.1 for detailed derivations).

Remark 3.1. *The optimal solution to the reverse-KL regularized reward maximization problem,* $\arg\max_{\pi_\theta} J_\beta(\pi_\theta)$*, is given by the target distribution* $\pi^* = G_\beta$*,*

$$G_\beta(y) = \frac{1}{\zeta}\, \pi_{ref}(y) \exp\left(\frac{R(y)}{\beta}\right), \tag{2}$$

where $\zeta = \int \pi_{ref}(y) \exp({}^{R(y)}/_{\beta})\, dy$ *is the normalizing constant.*

Remark 3.1 tells us the distribution maximizing Equation 1 is $\pi_\theta = G_\beta$. However, it may not be immediately obvious *how* optimizing Equation 1, $\nabla_\theta J_\beta(\pi_\theta)$, moves π_θ toward G_β. We analyze this below (details in Appendix B.2, also see e.g. Zhang & Ranganath (2025)).

Remark 3.2. *The gradient of Equation 1 is a gradient of the reverse KL divergence between the current policy* π_θ *and the target distribution* G_β*,*

$$\nabla_\theta D_{KL}\big(\pi_\theta \,||\, G_\beta\big) \propto -\nabla_\theta J_\beta(\pi_\theta)\,. \tag{3}$$

Main Takeaway

Maximizing the reverse-KL regularized RL objective J_β (Equation 1) is equivalent to doing distribution matching by minimizing a reverse KL toward the target distribution G_β (Equation 2).

3.2 Solution of the Forward KL Regularized Objective

Alternatively, the reward can be maximized with a forward KL penalty,

$$J_{\text{fwd}}(\pi_\theta) = \mathbb{E}_{\pi_\theta(y)}[R(y)] - \beta\, D_{KL}\big(\pi_{\text{ref}} || \pi_\theta\big)\,. \tag{4}$$

A number of recent works have used forward KL regularization. Some are motivated explicitly by the "mass covering" intuition of the forward KL (Wang et al., 2023), while others—such as GRPO (Shao et al., 2024; Guo et al., 2025a)—may have incidentally estimated the forward KL, despite meaning to use the reverse KL (Tang & Munos, 2025).

Remark 3.3. *Assume optimization with $\beta > 0$, with finite rewards $R_{max} < \infty$, and there exist solution(s) where $R(y) = R_{max}$, $\pi_{ref}(y) > 0$. The optimal solution to the forward-KL regularized reward maximization problem, $\arg\max_{\pi_\theta} J_{fwd}$, is given by the distribution:*

$$G_{fwd}(y) = \frac{\beta\,\pi_{ref}(y)}{\Lambda - R(y)}\,, \quad \Lambda > \max_y R(y)\,, \tag{5}$$

where a unique Λ exists for each β such that G_{fwd} is a valid probability distribution.

Notably, Equation 5 is a *completely different* distribution family from the reverse KL case (Equation 2), and does not have a simple closed form unnormalized solution. It is also worth noting that *if* higher-rewarding regions exist outside of π_{ref}'s support, G_{fwd} *can* place nonzero mass on regions where $\pi_{\text{ref}}(y) = 0$ and $R(y) = R_{\max}$, with no preference among y's within this region in terms of density. See Appendix B.3 for more details.

Remark 3.4. *Assume we are optimizing within the support of π_{ref}, the gradient of Equation 4 is* ***not*** *a forward KL gradient,*

$$\nabla_\theta\, D_{KL}\big(h\,||\,\pi_\theta\big) \not\propto -\nabla_\theta\, J_{fwd}(\pi_\theta)\,, \tag{6}$$

for ***any*** *target distribution h that is defined independently of π_θ, and arbitrary reward functions R.*

Proof. Appendix B.4. □

Therefore, while Equation 4 can still be a good objective to optimize, it does not necessarily inherit exactly the same properties and intuitions as a "forward KL gradient".

What, then, is the gradient of the forward KL $D_{KL}(G_\beta||\pi_\theta)$? It in fact reduces to doing maximum likelihood (supervised fine-tuning) on trajectories sampled from the target G_β (Remark B.2), which is intractable for generic targets. However, this provides one perspective on algorithms such as STaR (Zelikman et al., 2022) and RAFT (Dong et al., 2023; Xiong et al., 2025) that filter high-reward trajectories for maximum likelihood. One can interpret filtering as rejection sampling to approximate a target distribution (which put high mass over high-reward regions), when reward is bounded and we know G_β up to normalization. More generally, other methods that approximately sample from G_β to minimize divergence include Naesseth et al. (2020); Khalifa et al. (2020).

Main Takeaway

Maximizing the forward-KL regularized objective J_{fwd} (Equation 4) does not yield a forward-KL gradient, so its behaviour cannot be naively equated to forward-KL optimization.

To summarize: the regularized RL objective implicitly minimizes a divergence between the current policy π_θ and a target distribution G. Different choices of regularizer lead to different target distributions G. Importantly, the regularizing divergence $D(\pi_\theta, \pi_{\text{ref}})$ need *not* be the same type of divergence as the one implicitly minimized, $D(\pi_\theta, G)$, as is the case for forward-KL regularization.

3.3 Both KL Regularization Can Have Multimodal Solution Distributions

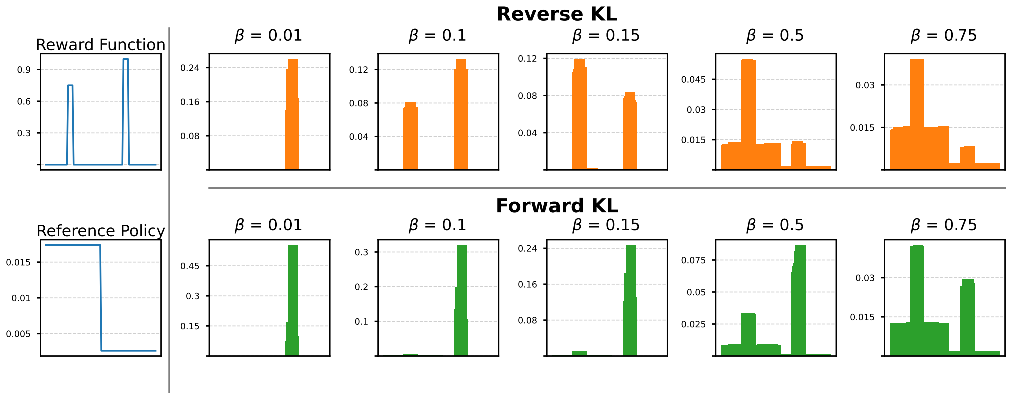


Figure 2: Final policy distribution (100-dim categorical) from optimizing a reverse/forward KL regularized reward maximization objective, given the same reward function, reference policy, across a range of regularization strengths (β). Both KLs can lead to multimodal solution distributions.

We briefly note that the target distributions for both the reverse (Equation 2) and forward (Equation 5) KL regularization *can* be multimodal. To ground the discussion, we first define a common-sense notion of "multimodal" in terms of reward modes, which we use for the rest of the paper.

Definition 3.5. A target distribution G for regularized reward maximization is "**reward multimodal**" given a high-reward threshold τ if: for any two samples y, y', $R(y) < \tau \leq R(y')$ implies $G(y) < G(y')$; and for any above-threshold samples $R(y) \geq \tau$ and $R(y') \geq \tau$, $G(y) \approx G(y')$.

Informally, this means all high-reward samples have a high probability, and sampling from G samples approximately equally from *all* high-reward regions.

We show in a didactic example in Figure 2, where given the same reward function containing two high-reward modes, and a reference policy with support over the first half of the action space, optimizing the reverse and forward KL objectives lead to a wide variety of solutions that depend on the regularization coefficient β. Both KLs have settings of β that induce *reward multimodal* target distributions. We analyze the properties of the target distribution in the subsequent section, and return to the Figure 2 example in detail in Section 4.3.

4 Analysis of KL Regularized Optimal Distribution

We have seen in Section 3.3 that both KL-regularized RL objectives can have *reward multimodal* solutions, and in Section 2 that optimizing either KL divergence to global optimum will give us policies that well-approximate the (multimodal) solution. However, the shape of the target distribution depends on the reward, reference distribution, and regularization strength. This raises the central question:

> *Is the globally optimal solution we commonly define in KL-regularized RL actually reward multimodal (Definition 3.5)?*

The central tool we use in this section is a *probability ratio* between two samples under a distribution. Intuitively, we want (i) high-reward samples to be much more probable than low-reward samples, and (ii) similarly high-reward samples to have similar high probabilities. Unless otherwise stated, we focus our analysis on the solution of the reverse-KL regularized objective (Equation 2), both for its clean form and because it is the most common way KL-regularized RL is formulated.

Proposition 4.1. *The (log) probability ratio between any two samples, y_1, y_2, under the optimal solution distribution for reverse-KL regularized RL, G_β, can be written in closed form,*

$$\log \frac{G_\beta(y_1)}{G_\beta(y_2)} = \log \frac{\pi_{ref}(y_1)}{\pi_{ref}(y_2)} + \frac{1}{\beta}\Big(R(y_1) - R(y_2)\Big)\,. \tag{7}$$

Proof. Because normalization constant ζ cancel out in ratios. See Appendix B.6. □

Proposition 4.1 gives us a generic and closed-form way of analyzing how likely one sample is relative to another in the *optimal solution*, using *only* π_{ref} and the reward function R, for *any* reverse-KL regularized reward maximization objective. This gives us a number of consequential insights.

4.1 With equal supports, small reward change drives large probability change

Remark 4.2. *For any two samples y_1 and y_2, if $\pi_{ref}(y_1) = \pi_{ref}(y_2)$, their probability ratio is:*

$$\frac{G_\beta(y_1)}{G_\beta(y_2)} = \exp\left(\frac{R(y_1) - R(y_2)}{\beta}\right). \tag{8}$$

We first consider the continuous reward function setting where samples have small differences in rewards. This is common for settings such as alignment (reward model) and drug discovery (e.g. binding affinity). If two samples have the *same* probability under the reference distribution π_{ref} ("equal support"), the difference in their final log probabilities is simply the difference in their rewards, scaled by $1/\beta$. Smaller β exaggerates the difference

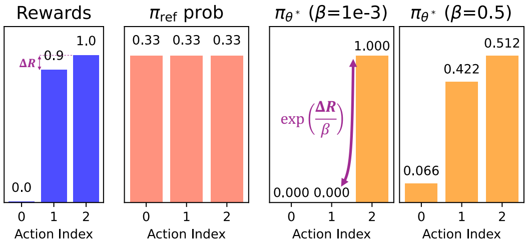


Figure 3: With equal π_{ref}, linear difference in rewards (ΔR) lead to exponential difference in probabilities

between relative probabilities. Note a *linear* difference in rewards result in an *exponential* difference in probabilities: for a 0.1 difference in rewards, and a commonly used $\beta = 1\text{e-}3$, the higher reward sample is pushed to be 2.6×10^{43} times more likely in the solution distribution. This issue is identically present for entropy-only regularization (see Fig. 7 for effect of β on relative probabilities). This suggests for commonly used hyperparameters, the solution is highly concentrated around the max reward mode(s).

We see in Figure 3 a didactic experiment that verifies this theory. At low regularization strength (β), the optimized policy π_{θ^*} mode collapses onto the highest reward action. At high β, policy achieves better (still not perfect) coverage over high-reward answers, at the cost of having more mass on low reward actions (more details and results in Appendix C.1).

4.2 With equal rewards, solution never prefers lower-support samples

We now consider the case where the correct solutions all have *equal* reward. This is a standard set-up for RL with verifiable reward (e.g. math), where a correct answer is given a reward of 1, and incorrect answers given 0.

Remark 4.3. *For any two samples with the same reward, $R(y_1) = R(y_2)$, their probability ratio is:*

$$\frac{G_\beta(y_1)}{G_\beta(y_2)} = \frac{\pi_{ref}(y_1)}{\pi_{ref}(y_2)} . \tag{9}$$

In words, the correct answers' probability ratio in the optimal solution is simply their probability ratio in the reference distribution π_{ref}. This ratio is *independent* of the regularization strength β. In other words, by construction, KL-regularized RL with equal rewards *never promotes a low-support answer*.[1] Figure 4 demonstrates this point empirically: the final policy *never* favours the equally correct low-support mode (additional results in Appendix C.1). This is not an issue with exploration; we will see in the subsequent section that with a small change in reward one can optimize for a distribution that equally weights or even prefers the lower-support solution.

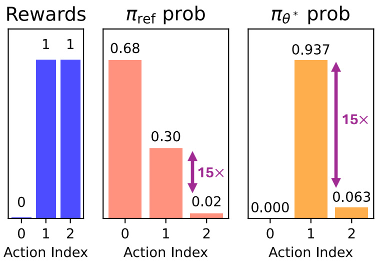


Figure 4: With equal rewards, RL does not change answers' relative probs.

Main Takeaway

RL with *any* KL-regularization does not increase the relative probability of lower-support samples to high-support ones, as long as their rewards are the same. Lowering the KL regularization strength β has *no effect* on up-weighting low-support samples in the optimal solution.

We additionally corroborate in Appendix C.4 that in practice for LLMs, the shape of the reference distribution and reward function does result in highly skewed target distribution G_β, per Proposition 4.1, Remark 4.2, and Remark 4.3.

4.3 For unequal rewards *and* supports, regularization strength determines mode coverage

When two trajectories have different rewards and different probabilities under the reference policy, a unique setting of β will induce the two to have the same probability in the target distribution.

Remark 4.4. *Two samples have the same probability in the target distribution if,*

$$R(y_2) - R(y_1) = \beta\big(\log \pi_{ref}(y_1) - \log \pi_{ref}(y_2)\big) . \tag{10}$$

This condition allows us to predict, given only the reward and reference policy, when two samples will have the same probabilities in the solution to the RL problem. As an example, we know in Figure 2 that the two high-reward modes have rewards 0.75 and 1.0, and reference policy probabilities of $\log \pi_{\text{ref}}(y_1) \approx -4.05$ and $\log \pi_{\text{ref}}(y_2) \approx -5.95$, respectively. This allows us to predict the setting of β which will "flip" the target distribution's preference from the high-support mode to the low-support mode to be $(1 - 0.75)/(-4.05 + 5.95) \approx 0.132$. Indeed, we see in Figure 2 for the reverse KL case, the preference between the two modes switch as we move from $\beta = 0.15$ to $\beta = 0.10$. This is the true role of the regularization coefficient β: it is a knob that decides between picking higher rewarding, low-support solutions, vs. lower rewarding, high-support solutions.

[1] This observation is true for both reverse and forward-KL regularized RL.

5 Directly Optimizing a Reward Multimodal Target

Having identified the various failure cases of the KL-regularized RL objective (Section 4), and the role of regularization in balancing reward differences (Section 4.3), we now turn to the question:

> *Can we construct an objective that, when optimized, naturally give rise to a reward multimodal target distribution?*

Indeed, Remark 4.4 already provides the equality condition required to achieve this. We derive a simple procedure which will ensure we are optimizing for a solution that puts *equal* probabilities on all high-quality samples (per Definition 3.5), using the augmented reward function,

$$\bar{R}(y) = \begin{cases} R(y) & \text{if } R(y) < \tau, \\ R(z) + \beta\big(\log \pi_{\text{ref}}(z) - \log \pi_{\text{ref}}(y)\big) & \text{if } R(y) \geq \tau, \end{cases} \tag{11}$$

where $\tau \leq \max_y R(y)$ is some threshold for "goodness", and z is a fixed "anchor" sample chosen from the set of high-quality samples. We can pick it to be $z = \arg\max_y \pi_{\text{ref}}(y)$ where $R(y) \geq \tau$. Because we are choosing the "anchor" to be from a high-reward mode, we colloquially refer to this as "*mode anchoring*", and the method as ***M**ode **A**nchored **R**eward **A**ugmentation* (**MARA**). See Algorithm 1 for pseudocode with minimal changes (an alternative that augments reward and π_{ref} is outlined in Algorithm 2, which is equivalent to Alg.1 when using reverse KL regularization).

Algorithm 1 Mode Anchored Reward Augmentation (MARA), within a sampled batch. Changes from a standard RL algorithm are in blue.

```
Given: initial policy $\pi_\theta$, reference distribution $\pi_{\text{ref}}$, reward function $R$, regularization coefficient
  $\beta$, threshold of good answers $\tau \in \mathbb{R}$, $\tau \leq \max_y R(y)$, and trajectory batch $\{y_i\}_{i=1}^N \sim \pi_\theta$.
Pick anchor trajectory: $z = \arg\max_{y_i} \pi_{\text{ref}}(y_i)$, s.t. $R(y_i) \geq \tau$
for each $y_i$ in batch do
    if $R(y_i) \geq \tau$ then
        Augment: $\bar{r}_i = R(z) + \beta\big(\log \pi_{\text{ref}}(z) - \log \pi_{\text{ref}}(y_i)\big)$
    else
        Keep same: $\bar{r}_i = R(y_i)$
    end if
end for
Optimize policy parameters $\theta$ using augmented rewards $\{\bar{r}_i\}_{i=1}^N$.
```

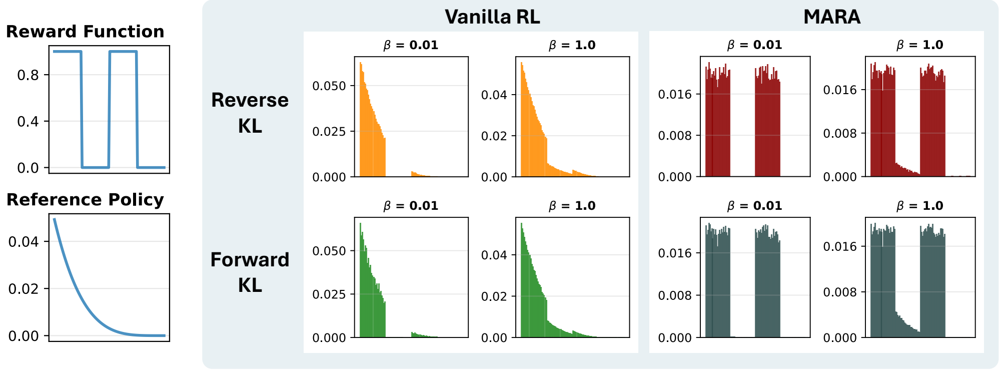


Figure 5: MARA stays close to the reference policy in low-reward areas, and puts high, uniform mass over all high-reward areas.

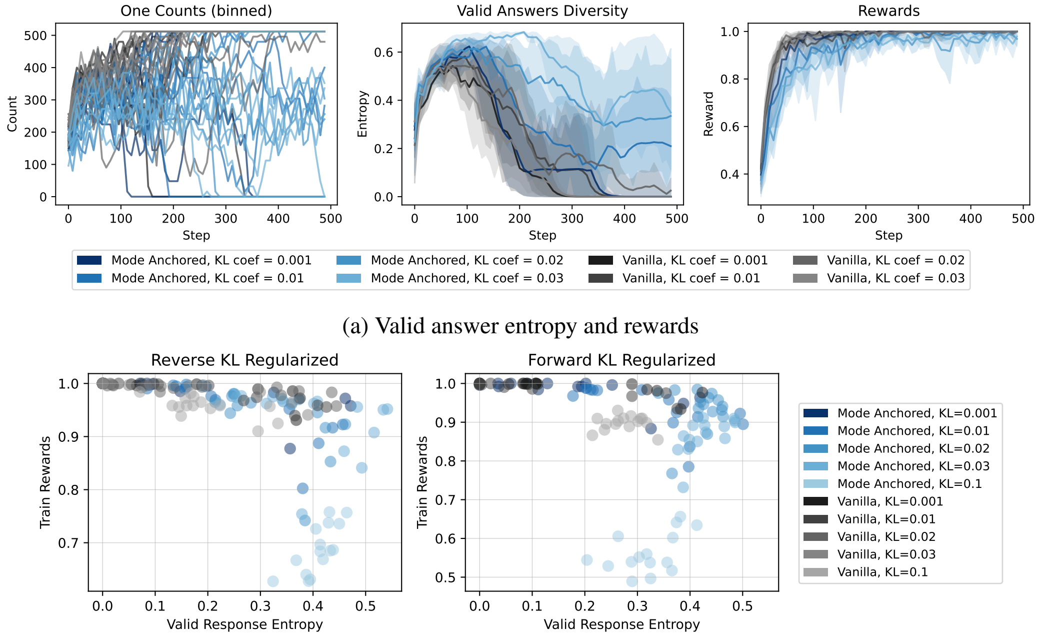
Intuitively, the augmented reward function constructs a new *target distribution* with *uniform* high density over regions of high reward, and stays close to the reference π_{ref} in regions of low reward (see Remark B.3 for an analysis of the shape of the MARA target distribution). We see in the Figure 5 that vanilla KL-regularized RL result in a policy that heavily favours the left (on-support) mode, regardless of the choice of β or KL. On the other hand, using MARA results in solutions that put *equal* high mass over *all* high quality samples, for both KLs. Note that in cases where the reward function range is known, one can directly set threshold τ as a constant. If not, one can set τ on a per-batch basis by e.g. taking an upper percentile of sampled rewards (as we do below for non-verifiable LLM tasks).

6 Empirical Validations

We evaluate MARA as a drop-in method in a variety of post-training tasks. While our theory has mainly been about the final optimal solution RL achieves, we empirically investigate whether training, even if stopped early, can still benefit from a more diverse global optimum. To do this, we evaluate MARA in (i) verifiable LLM task with multiple answers, (ii) non-verifiable task with reward models, and (iii) chemical language model task for drug discovery, where mode collapse is detrimental.

6.1 Verifiable 1-2 Task for LLM

We train an LM (`Qwen2.5-3B`) to generate uniform random integers that are either 1 or 2. It gets a reward of 1.0 for correct (producing "1" or "2" in XML), and 0.0 otherwise (details in Appendix C.2). Most runs are able to optimize the reward well and achieve a reward of approximately 1 (Figure 6a, right). Figure 6a (left) shows the number of correctly formatted 1's the LM generates over the course of training. We see that for naive KL regularization (grey), across a range of β's and seeds, all but one run collapse into generating only a single answer as a result of RL, and most collapse into generating 1's, which has higher likelihood under the base policy. MARA (blue), on the other hand, is able to preserve the diversity in the correct answers, with many runs learning to generate 1's and 2's with near uniform probability, while still learning to generate with the correct format (Figure 6a, middle). Further, the Pareto front of model checkpoints at different points in training shows that for both reverse and forward KL regularization, MARA is able to match vanilla training in terms of correctness, while exceeding vanilla training in terms of generation diversity.



(a) Valid answer entropy and rewards

(b) Pareto front of reward (quality) and entropy (diversity)

Figure 6: Performance on verifiable task with multiple solutions, against both reverse & forward KL. MARA (blue) is compared against baseline GRPO (grey) at different β's (`KL coef`).

6.2 Creative Question Answering for Chat LLM

We test MARA in a non-verifiable alignment task. We train `Qwen3-1.7B` on a subset of WildChat text (Zhao et al., 2024), using a parametric reward model (`Skywork-Reward-V2-Qwen3-4B`). We evaluate the model on a curated test set (Zhang et al., 2025) and report both the training reward (`In dist Reward`), and test set reward from a different reward model (`Out dist Reward`, using `Skywork-Reward-Gemma-2-27B-v0.2`). We also report diversity metrics in terms of n-grams (`Ngrams`), semantic embeddings (`Semantic Div`), and "distinct functional classes" (`Mean Distinct`). See Appendix C.3 for more details. Here, MARA is used as a drop-in replacement in an RLOO style algorithm (Ranganath, 2017; Kool et al., 2020; Ahmadian et al., 2024).

As the reward function range is not known beforehand, we set τ on a per-batch basis as the 90th percentile reward of each batch (we ablate this in Appendix Table 5 to find that lowering the percentile decreases performance slightly, but nevertheless remains competitive). We compare against regular RL training, and a number of diversity-promoting baselines including entropy regularization (`Entropy`), rewarding the unlikely (`Unlikely`, He et al. 2025), best-of-N training (`BoN Training`, Tang et al. 2025), and weight ensembling (`Ensemble`, Dang et al. 2025). We see MARA out-performs all baselines in terms of out-of-distribution rewards, and all-but-one diversity metrics (Table 1).

Model	**In-dist. Reward** (↑)	**Out-dist. Reward** (↑)	**Ngrams EAD** (↑)	**Semantic Div** (↑)	**Mean Distinct** (↑)
Base Model	10.94	1.166 ±0.076	0.413 ±0.015	0.220 ±0.009	4.01 ±0.254
GRPO	14.80	1.317 ±0.102	0.497 ±0.014	0.193 ±0.009	3.96 ±0.249
RLOO	15.56	1.280 ±0.100	0.514 ±0.014	0.192 ±0.008	3.88 ±0.243
Entropy	1.44	0.786 ±0.073	0.267 ±0.009	**0.228 ±0.009**	3.45 ±0.193
Unlikely	10.04	1.381 ±0.114	0.532 ±0.015	0.191 ±0.008	4.24 ±0.239
BoN Training	**16.88**	0.596 ±0.055	0.541 ±0.010	0.162 ±0.008	2.29 ±0.173
Ensemble	–	1.143 ±0.086	0.438 ±0.014	0.211 ±0.010	4.19 ±0.269
MARA (rev)	15.42	1.451 ±0.103	0.543 ±0.014	0.186 ±0.008	4.14 ±0.233
MARA (fwd)	15.33	**1.604 ±0.113**	**0.568 ±0.012**	0.193 ±0.009	**4.62 ±0.258**

Table 1: Performance on non-verifiable creative task. Mean ± bootstrap SEM.

6.3 Drug Discovery with Chemical Language Models

Finally, we apply MARA to a distinctively different domain where diversity and quality is crucial: drug discovery. Chemical language models (CLMs) have seen success in discovering molecules in clinical trials. We adapt two realistic reward functions from Guo et al. (2025b): `SYNTH` and `ALL-AMIDE` that jointly reward binding potency and synthesizability. The core CLM optimization problem is also a regularized RL problem: maximize expected reward, while staying close to a pretrained "prior" model to ensure chemical validity. Unlike the traditional RL setting, CLMs are evaluated based on their ability to generate *unique* molecules (`Yield`) given a *fixed* number of reward function evaluations (which are expensive simulations and/or experiments), making diversity an essential quality for any performant CLMs. The `REINVENT` algorithm (Olivecrona et al., 2017; Guo & Schwaller, 2024b) is a state-of-the-art RL-based method on standard benchmarks (Gao et al., 2022). It optimizes the following objective,

$$\mathcal{L}(\theta) = -\Big[\log \pi_\theta(y) - \big(\log \pi_{\text{ref}}(y) + \sigma R(y)\big)\Big]^2, \quad y \sim \pi_\theta \,. \tag{12}$$

which is equivalent to KL-regularized reward maximization. We apply MARA as a *drop-in replacement* to its rewards. Additional evaluation details are in Appendix C.5.

Table 2 shows MARA consistently results in higher average `Yield` (number of *unique* high-reward molecules discovered), and lower `OB100` (efficiency in finding high-reward molecules, measured by reward function calls). The screening level (`Screen`) is the reward threshold above which we accept discovered molecules. Setting MARA's τ equal to the screening level always results in the highest yield, consistent with MARA's target distribution having uniform density in areas where $R(y) > \tau$. Going further, we also assess "global" diversity (which MARA does not explicitly optimize for) in terms of `IntDiv1` and `#Circles`. Both define more macroscopic differences based on molecular sub-structures. We find MARA is competitive with the baseline here. Overall, we see MARA further boosts REINVENT's optimization efficiency, while maintaining diversity.

7 Conclusion

In this work, we provide an in-depth understanding of the KL-regularized RL objective, particularly in terms of its diversity. We summarize the main take-aways below.

- Studying the divergence being *implicitly minimized* between the policy and target distribution, $D(\pi_\theta, G)$, is more meaningful for understanding optimization behaviour than studying the divergence of the regularizer, $D(\pi_\theta, \pi_{\text{ref}})$, alone.

Screen	Algorithm	Yield (↑)	OB100 (↓)	IntDiv1 (↑)	Circles (↑)
0.80	REINVENT	6569 ± 186	1042 ± 66	0.766 ± 0.011	67 ± 3
	MARA (τ=0.80)	$\mathbf{6834 \pm 78}$	1015 ± 55	0.761 ± 0.009	59 ± 8
	MARA (τ=0.85)	6584 ± 231	1042 ± 66	0.761 ± 0.008	72 ± 6
0.85	REINVENT	1614 ± 407	4114 ± 109	0.701 ± 0.018	7 ± 1
	MARA (τ=0.80)	1796 ± 210	$\mathbf{3654 \pm 272}$	0.716 ± 0.015	6 ± 1
	MARA (τ=0.85)	$\mathbf{2196 \pm 394}$	4010 ± 297	0.703 ± 0.011	7 ± 1

(a) SYNTH task

Screen	Algorithm	Yield (↑)	OB100 (↓)	IntDiv1 (↑)	Circles (↑)
0.80	REINVENT	5433 ± 184	1427 ± 63	0.768 ± 0.012	35 ± 1
	MARA (τ=0.80)	5635 ± 249	1407 ± 123	0.766 ± 0.008	36 ± 3
	MARA (τ=0.85)	5502 ± 309	1426 ± 63	0.769 ± 0.006	34 ± 3
0.85	REINVENT	1098 ± 88	4360 ± 257	0.721 ± 0.016	8 ± 1
	MARA (τ=0.80)	$\mathbf{1235 \pm 130}$	$\mathbf{3943 \pm 303}$	0.733 ± 0.009	8 ± 1
	MARA (τ=0.85)	$\mathbf{1438 \pm 126}$	4230 ± 401	0.725 ± 0.008	8 ± 1

(b) ALL-AMIDE task

Table 2: Results for different tasks and screening levels (`Screen`, higher meaning more strict) for two challenging drug discovery tasks. Error bars (±) denote standard deviation over 5 independent seeds. Bold indicates if the performance is statistically significantly better than the alternative method for that screening level (one-sided student's t-test, $p < 0.05$).

- The regularizer, β, and reward function together define the *target distribution* which π_θ optimizes towards. Forward and reverse KL regularizers define different target distributions.
- For common hyperparameters and reward functions used in practice, the target distribution is often *defined to be unimodal*. Thus, perfectly solving the regularized RL objective yields non-diverse optimal policy distributions.
- This diversity loss can be fixed by instead defining multimodal target distributions, such as through dynamically augmenting the reward via MARA.

There are a number of exciting future directions to improve MARA. For one, MARA requires setting τ (the reward threshold above which the target distribution puts uniform probability mass). The choice of τ is obvious in some settings (Section 6.1 & 6.3), but is a hyperparamter in settings with unbounded reward functions and unknown a priori thresholds (Section 6.2). We found that setting batch-specific τ helps, but better approaches may be possible. It should be noted that setting an *overly high* τ is harmless: if no samples meet the τ threshold, the MARA mechanism does not kick in, and learning reduces to standard RL.

Further, MARA shapes the target distribution to have multimodality, but does not guarantee faster convergence to the target. Having general algorithms to more efficiently reach the target, *or* to guarantee distributional properties (e.g. multimodality) at sub-optimality, would be of general importance and complement MARA's target distribution. Further, MARA introduces *one* such target distribution which places uniform mass at regions where $R(y) \geq \tau$. This may not be an optimal choice for all tasks, and considering alternative mode-preserving target distributions can be interesting future work. All in all, we emphasize that regularized RL—as commonly used for generative model training—is inherently a distribution matching problem and should be viewed as such. Rather than relying on intuitions (e.g. about regularizers), we should directly specify distributions with properties we wish to have as the target of policy optimization.

Reproducibility Statement

We use open-source, publicly available libraries for all experimental code. Didactic experiments are constructed in PyTorch (Paszke et al., 2019). Reinforcement learning on LLM training is done using the `nano-aha-moment` (Kazemnejad et al., 2025) and verl (`https://github.com/`

volcengine/verl) github repos. Chemical language model experiments use the official saturn github repo (Guo & Schwaller, 2024b). We provide detailed experimental information in Appendix C. Pseudo-code is provided in Algorithm 1 and Algorithm 2.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

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. *arXiv preprint arXiv:2402.14740*, 2024.

Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

Eser Aygün, Anastasiya Belyaeva, Gheorghe Comanici, Marc Coram, Hao Cui, Jake Garrison, Renee Johnston Anton Kast, Cory Y McLean, Peter Norgaard, Zahra Shamsi, et al. An ai system to help scientists write expert-level empirical software. *arXiv preprint arXiv:2509.06503*, 2025.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Esben Jannik Bjerrum. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076*, 2017.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Dean G Brown and Jonas Bostrom. Analysis of past and present synthetic methodologies on medicinal chemistry: where have all the new reactions gone? miniperspective. *Journal of medicinal chemistry*, 59(10):4443–4458, 2016.

Pablo Samuel Castro, Nenad Tomasev, Ankit Anand, Navodita Sharma, Rishika Mohanta, Aparna Dev, Kuba Perlin, Siddhant Jain, Kyle Levin, Noémi Éltető, et al. Discovering symbolic cognitive models from human and animal behavior. *bioRxiv*, pp. 2025–02, 2025.

Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: learning retrosynthetic planning with neural guided a* search. In *International conference on machine learning*, pp. 1608–1616. PMLR, 2020.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. Modifying large language model post-training for diverse creative writing. *arXiv preprint arXiv:2503.17126*, 2025.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Xingyu Dang, Christina Baek, Kaiyue Wen, Zico Kolter, and Aditi Raghunathan. Weight ensembling improves reasoning in language models. *arXiv preprint arXiv:2504.10478*, 2025.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Yuanqi Du, Arian R Jamasb, Jeff Guo, Tianfan Fu, Charles Harris, Yingheng Wang, Chenru Duan, Pietro Liò, Philippe Schwaller, and Tom L Blundell. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, pp. 1–16, 2024.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in neural information processing systems*, 35: 21342–21357, 2022.

I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Selected Russian Publications in the Mathematical Sciences. Prentice-Hall, Englewood Cliffs, N.J., 1963. Revised English edition.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. *arXiv preprint arXiv:2302.08215*, 2023.

Jean-Bastien Grill, Florent Altché, Yunhao Tang, Thomas Hubert, Michal Valko, Ioannis Antonoglou, and Rémi Munos. Monte-carlo tree search as regularized policy optimization. In *International Conference on Machine Learning*, pp. 3769–3778. PMLR, 2020.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025a.

Jeff Guo and Philippe Schwaller. Augmented memory: Sample-efficient generative molecular design with reinforcement learning. *JACS Au*, 2024a.

Jeff Guo and Philippe Schwaller. Saturn: Sample-efficient generative molecular design using memory manipulation. *arXiv preprint arXiv:2405.17066*, 2024b.

Jeff Guo, Víctor Sabanza-Gil, Zlatko Jončev, Jeremy S Luterbacher, and Philippe Schwaller. Generative molecular design with steerable and granular synthesizability control. *arXiv preprint arXiv:2505.08774*, 2025b.

Anthony GX-Chen, Dongyan Lin, Mandana Samiei, Doina Precup, Blake Aaron Richards, Rob Fergus, and Kenneth Marino. Language agents mirror human causal reasoning biases. how can we help them think like scientists? In *Second Conference on Language Modeling*, 2025.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Jubayer Ibn Hamid, Ifdita Hasan Orney, Ellen Xu, Chelsea Finn, and Dorsa Sadigh. Polychromic objectives for reinforcement learning. *arXiv preprint arXiv:2509.25424*, 2025.

Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025.

Aspen K Hopkins, Alex Renda, and Michael Carbin. Can LLMs generate random numbers? evaluating LLM sampling in controlled domains. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023. URL https://openreview.net/forum?id=Vhh1K9LjVI.

Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. *arXiv preprint arXiv:2310.04363*, 2023.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. *arXiv preprint arXiv:2412.01951*, 2024.

Shiyu Huang, Hang Su, Jun Zhu, and Ting Chen. Svqn: Sequential variational soft q-learning networks. In *International Conference on Learning Representations*, 2019.

Mete Ismayilzada, Antonio Laverghetta Jr, Simone A Luchini, Reet Patel, Antoine Bosselut, Lonneke van der Plas, and Roger Beaty. Creative preference optimization. *arXiv preprint arXiv:2505.14442*, 2025.

Yash Jhaveri, Harley Wiltzer, Patrick Shafto, Marc G Bellemare, and David Meger. Convergence theorems for entropy-regularized and distributional reinforcement learning. *arXiv preprint arXiv:2510.08526*, 2025.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Amirhossein Kazemnejad, Milad Aghajohari, Alessandro Sordoni, Aaron Courville, and Siva Reddy. Nano aha! moment: Single file "rl for llm" library. https://github.com/McGill-NLP/nano-aha-moment, 2025. GitHub repository.

Peter W Kenny. Hydrogen-bond donors in drug design. *Journal of medicinal chemistry*, 65(21): 14261–14275, 2022.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635*, 2020.

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. *arXiv preprint arXiv:2310.06452*, 2023.

Wouter Kool, Herke van Hoof, and Max Welling. Estimating gradients for discrete random variables by sampling without replacement. *arXiv preprint arXiv:2002.06043*, 2020.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Oh Joon Kwon, Daiki E Matsunaga, and Kee-Eung Kim. Gdpo: Learning to directly align language models with diversity using gflownets. *arXiv preprint arXiv:2410.15096*, 2024.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025.

Siyang Liu, Sahand Sabour, Yinhe Zheng, Pei Ke, Xiaoyan Zhu, and Minlie Huang. Rethinking and refining the distinct metric. *arXiv preprint arXiv:2202.13587*, 2022.

Mark F Mabanglo, Keith S Wong, Marim M Barghash, Elisa Leung, Stephanie HW Chuang, Afshan Ardalan, Emily M Majaesic, Cassandra J Wong, Shen Zhang, Henk Lang, et al. Potent clpp agonists with anticancer properties bind with improved structural complementarity and alter the mitochondrial n-terminome. *Structure*, 31(2):185–200, 2023.

Krzysztof Maziarz, Austin Tripp, Guoqing Liu, Megan Stanley, Shufang Xie, Piotr Gaiński, Philipp Seidl, and Marwin HS Segler. Re-evaluating retrosynthesis algorithms with syntheseus. *Faraday Discussions*, 256:568–586, 2025.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Christian Naesseth, Fredrik Lindsten, and David Blei. Markovian score climbing: Variational inference with kl (p‖ q). *Advances in Neural Information Processing Systems*, 33:15499–15510, 2020.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017.

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

Laura O'Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, volume 2, 2024.

Sven Michael Papidocha, Andreas Burger, Varinia Bernales, and Alán Aspuru-Guzik. The elephant in the lab: Synthesizability in generative small-molecule design. *ChemRxiv*, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

Martin L Puterman. Markov decision processes: Discrete stochastic dynamic programming, 1994.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems*, 36:53728–53741, December 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Rajesh Ranganath. *Black Box variational inference: Scalable, generic Bayesian computation and its applications*. PhD thesis, Princeton University, 2017.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pp. 814–822. PMLR, 2014.

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Mikołaj Sacha, Mikołaj Błaz, Piotr Byrski, Paweł Dabrowski-Tumanski, Mikołaj Chrominski, Rafał Loska, Paweł Włodarczyk-Pruszynski, and Stanisław Jastrzebski. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61(7):3273–3284, 2021.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Alexander Shypula, Shuo Li, Botong Zhang, Vishakh Padmakumar, Kayo Yin, and Osbert Bastani. Evaluating the diversity and quality of llm generated content. *arXiv preprint arXiv:2504.12522*, 2025.

Yuda Song, Julia Kempe, and Remi Munos. Outcome-based exploration for llm reasoning. *arXiv preprint arXiv:2509.06941*, 2025.

Megan Stanley and Marwin Segler. Fake it until you make it? generative de novo design and virtual screening of synthesizable molecules. *Current Opinion in Structural Biology*, 82:102658, 2023.

Anja Surina, Amin Mansouri, Lars Quaedvlieg, Amal Seddas, Maryna Viazovska, Emmanuel Abbe, and Caglar Gulcehre. Algorithm discovery with llms: Evolutionary search meets reinforcement learning. *arXiv preprint arXiv:2504.05108*, 2025.

Shidi Tang, Ji Ding, Xiangyu Zhu, Zheng Wang, Haitao Zhao, and Jiansheng Wu. Vina-gpu 2.1: towards further optimizing docking speed and precision of autodock vina and its derivatives. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2024.

Yunhao Tang and Rémi Munos. On a few pitfalls in kl divergence gradient estimation for rl. *arXiv preprint arXiv:2506.09477*, 2025.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Rémi Munos. Optimizing language models for inference time objectives using reinforcement learning. *arXiv preprint arXiv:2503.19595*, 2025.

Daniil Tiapkin, Nikita Morozov, Alexey Naumov, and Dmitry P Vetrov. Generative flow networks as entropy-regularized rl. In *International Conference on Artificial Intelligence and Statistics*, pp. 4213–4221. PMLR, 2024.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

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.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Peter West and Christopher Potts. Base models beat aligned models at randomness and creativity. *arXiv preprint arXiv:2505.00047*, 2025.

Yutong Xie, Ziqiao Xu, Jiaqi Ma, and Qiaozhu Mei. How much space has been explored? measuring the chemical space covered by databases and machine-generated molecules. In *Proc. 11th International Conference on Learning Representations*, 2023.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

Chenghao Yang and Ari Holtzman. How alignment shrinks the generative horizon. *arXiv preprint arXiv:2506.17871*, 2025.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025.

Longfei Yun, Chenyang An, Zilong Wang, Letian Peng, and Jingbo Shang. The price of format: Diversity collapse in llms. *arXiv preprint arXiv:2505.18949*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Lily H Zhang and Rajesh Ranganath. Preference learning made easy: Everything should be understood through win rate. *arXiv preprint arXiv:2502.10505*, 2025.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. Noveltybench: Evaluating language models for humanlike diversity. *arXiv preprint arXiv:2504.05228*, 2025.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. *arXiv preprint arXiv:2504.07912*, 2025.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A Related Work

Entropy collapse in RL There is a growing line of empirical works observing RL training collapses the diversity in generation output of the resulting post-trained policy (Kirk et al., 2023; Huang et al., 2024; O'Mahony et al., 2024; Cui et al., 2025; Yang & Holtzman, 2025; Yun et al., 2025; Shypula et al., 2025; West & Potts, 2025; Zhao et al., 2025; Dang et al., 2025; Song et al., 2025), such as in formats (Zhao et al., 2025), random generation and creativity (West & Potts, 2025), as well as exploration and reasoning (Cui et al., 2025; Dang et al., 2025; Song et al., 2025). The observations have mostly been empirical.

A few attempts have been made to theoretically understand entropy collapse. Cui et al. (2025) analyzes what per-step policy gradient (approximately) does to the entropy of a tabular softmax policy, and finds that entropy decreases if there is a strong a strong positive correlation between the action probabilities and corresponding advantage values. Dang et al. (2025) analyzes a special case of multi-arm bandits with K equally good arms and a bad arm, and finds that the optimal probabilities correspond to the re-normalized reference probabilities of just the good arms. We note this is a special case of our Remark 4.3.

Generally speaking, entropy preservation via regularization is often referred to as "max entropy RL", with numerous seminal and ongoing works (Ziebart et al., 2008; Haarnoja et al., 2017; Huang et al., 2019). Our analysis contributes to further understanding of this setting by shedding light on the multimodality of the target distribution when doing entropy regularization.

Training for diversity A number of attempts have been made to empirically address entropy collapse. Wang et al. (2023) generalizes the DPO objective (Rafailov et al., 2023) from reverse-KL regularized to a more general class of f-divergence regularizers, with the key motivation being that reverse-KL can be mode-seeking, therefore reduce diversity. We argue in this work that the full story is more nuanced and is better analyzed through the *target distribution*. Cui et al. (2025) proposes to directly regularize the update of high-covariance tokens. Cheng et al. (2025) incorporates an entropy term in the advantage to encourage better reasoning. Wang et al. (2025) show that focusing gradient updates on a minority of high-entropy "forking" tokens can improve reasoning. He et al. (2025) proposes a rank-based "unlikeliness" reward, where more likely samples (under current policy) receives a larger multiplicative penalty to the reward. Similarly, Yao et al. (2025) uses token entropy to encourage diversity. Song et al. (2025) proposes a count-based exploration bonus that more highly rewards less frequently seen outcomes (in previous samples), and Hamid et al. (2025) proposes a similar batch-wise reward. Dang et al. (2025) found that combining weights of earlier and later checkpoints can improve pass@k performance—one specific measure of diversity.

A number of works attempt to directly optimize for diversity. This relies on the existence of additional information that tells us if two samples are different and by how much. In this vein, diverse DPO Lanchantin et al. (2025) and variants (Chung et al., 2025; Ismayilzada et al., 2025) encourage diversity in preference learning by selecting diverse positives/negatives. Similarly related is Li et al. (2025), which use an external model to evaluate diversity (via a semantic classifier) and use the diversity metric to modify the reward. Hamid et al. (2025) proposes to optimize a batch-level objective that is modified by a diversity function. We do not require an external model to evaluate diversity.

In the unregularized setting, Jhaveri et al. (2025) optimizes for an unregularized policy with specific distributional properties. More distantly, GFlowNets also provide diversity-seeking policies that are specifically designed to sample proportionally to reward, albeit they use different algorithms than the KL-regularized policy gradient which is the most commonly used algorithm for LM post-training (Hu et al., 2023; Kwon et al., 2024; Tiapkin et al., 2024).

B Mathematical Derivations

B.1 Target Distribution of Reverse-KL Reward Maximization

Proof of Remark 3.1 We provide a proof for the maximizer of the generalized reverse-KL and entropy regularized reward maximization objective,

$$J_{\beta,\eta}(\pi_\theta) = \mathbb{E}_{\pi_\theta(y)}[R(y)] - \beta\, D_{KL}\big(\pi_\theta || \pi_{\text{ref}}\big) + \eta H(\pi_\theta)\,. \tag{13}$$

The solution $\arg\max_{\pi_\theta} J_{\beta,\eta}$ has the un-normalized form,

$$G_{\beta,\eta}(y) \propto g_{\beta,\eta}(y) = \pi_{\text{ref}}(y)^{\frac{\beta}{\beta+\eta}} \exp\left(\frac{R(y)}{\beta+\eta}\right). \tag{14}$$

Proof.

$$\begin{aligned}
J_{\beta,\eta}(\pi_\theta) &= \mathbb{E}_{\pi_\theta(y)}\Big[R(y) - \beta\big(\log \pi_\theta(y) - \log \pi_{\text{ref}}(y)\big) - \eta \log \pi_\theta(y)\Big], && (15)\\
&= -(\beta+\eta)\,\mathbb{E}_{\pi_\theta(y)}\Big[\log \pi_\theta(y) - \big(\frac{R(y)}{\beta+\eta} + \frac{\beta}{\beta+\eta}\log \pi_{\text{ref}}(y)\big)\Big], && (16)\\
&= -(\beta+\eta)\,\mathbb{E}_{\pi_\theta(y)}\Big[\log \pi_\theta(y) - \log \pi_{\text{ref}}(y)^{\frac{\beta}{\beta+\eta}} \exp\big(\frac{R(y)}{\beta+\eta}\big)\Big], && (17)\\
&= -(\beta+\eta)\,\mathbb{E}_{\pi_\theta(y)}\Big[\log \pi_\theta(y) - \log G_{\beta,\eta}(y)\Big] + (\beta+\eta)\log \zeta_{\beta,\eta}, && (18)\\
&= -(\beta+\eta) D_{KL}\Big(\pi_\theta || G_{\beta,\eta}\Big) + (\beta+\eta)\log \zeta_{\beta,\eta}, && (19)
\end{aligned}$$

where $\zeta_{\beta,\eta} = \int g_{\beta,\eta}(y)\,dy$. One can see that the above is maximized when $D_{KL}\Big(\pi_\theta || G_{\beta,\eta}\Big) = 0$, which occurs when $\pi_\theta = G_{\beta,\eta}$. □

Intuitively, one can see the entropy regularizer η as playing the role of "tempering" the reference distribution π_{ref} (larger η drives π_{ref} to become more uniform), while both β and η lower the reward's effect on the target distribution. For the **KL-only case** ($\eta = 0$), the target distribution becomes,

$$G_\beta(y) \propto \pi_{\text{ref}}(y) \exp\left(\frac{R(y)}{\beta}\right), \tag{20}$$

which is the stated result in Remark 3.1. In the **entropy-only case** ($\beta = 0$), the solution is,

$$G_\eta(y) \propto \exp\left(\frac{R(y)}{\eta}\right). \tag{21}$$

All in all, both coefficients play a role in parameterizing the *shape* of the optimal distribution for the regularized RL problem.

B.2 Gradient of Reverse-KL Reward Maximization

Proof of Remark 3.2 From Appendix B.1, we have the identity,

$$-\frac{1}{\beta} J_\beta(\pi_\theta) = D_{KL}\big(\pi_\theta || G_\beta\big) - \log \zeta. \tag{22}$$

We can easily show that the gradient is,

$$\begin{aligned}
\nabla_\theta\Big(-\frac{1}{\beta} J_\beta(\pi_\theta)\Big) &= \nabla_\theta\, D_{KL}\big(\pi_\theta || G_\beta\big) - \nabla_\theta \log \zeta, && (23)\\
&= \nabla_\theta\, D_{KL}\big(\pi_\theta || G_\beta\big). && (24)
\end{aligned}$$

In other words, they are the same up to constant $-\beta$,

$$\nabla_\theta J_\beta(\pi_\theta) = -\beta\, \nabla_\theta D_{KL}\big(\pi_\theta || G_\beta\big). \tag{25}$$

B.3 Target Distribution of Forward-KL Reward Maximization

We are interested in finding the distribution $\pi_\theta = G_{\text{fwd}}$ which maximizes,

$$J_{\text{fwd}}(\pi_\theta) = \mathbb{E}_{\pi_\theta(y)}[R(y)] - \beta\, D_{KL}\big(\pi_{\text{ref}} || \pi_\theta\big). \tag{26}$$

Note we can simplify the expression to only terms that depend on π_θ,

$$\begin{aligned}
\arg\max_{\pi_\theta} J_{\text{fwd}}(\pi_\theta) &= \arg\max_{\pi_\theta} \mathbb{E}_{\pi_\theta}\big[R(y)\big] - \beta\, D_{KL}\big(\pi_{\text{ref}} || \pi_\theta\big), && (27)\\
&= \arg\max_{\pi_\theta} \int \pi_\theta(y)\, R(y) + \beta\, \pi_{\text{ref}}(y) \log \pi_\theta(y)\, dy + \text{const}. && (28)
\end{aligned}$$

Remark B.1. *Assuming the reward is finite and has maximum value R_{max}. If any on-support answer(s) have $R(y) = R_{max}$, $y \in supp(\pi_{ref})$, the optimal distribution maximizing J_{fwd} will put zero mass outside of $supp(\pi_{ref})$.*

Proof. Let $\mathbf{M}$ be the set of on-support, max reward answers: $\mathbf{M}$ if $R(y) = R_{\text{max}}$, and $\mathbf{M} \subseteq \text{supp}(\pi_{\text{ref}})$. We can generically write any distribution π that puts non-zero mass outside of $\text{supp}(\pi_{\text{ref}})$ as,

$$J_{\text{fwd}}(\pi) = C + \int_{\mathbf{M}} \pi(y) R_{\text{max}} + \beta\, \pi_{\text{ref}}(y) \log \pi(y)\, dy + \int_{y \notin \text{supp}(\pi_{\text{ref}})} \pi(y) R(y)\, dy\,, \tag{29}$$

where C captures the contribution to the objective from the remaining $y \in \text{supp}(\pi_{\text{ref}}), y \notin \mathbf{M}$. Note the forward KL penalty for $y \notin \text{supp}(\pi_{\text{ref}})$ is zero. We show we can always construct an alternative distribution, π', with mass only inside of $\text{supp}(\pi_{\text{ref}})$ and has strictly higher J_{fwd}. We write,

$$J_{\text{fwd}}(\pi') = C + \int_{\mathbf{M}} \Big(\pi(y) + \alpha(y)\Big) R_{\text{max}} + \beta\, \pi_{\text{ref}}(y) \log \Big(\pi_\theta(y) + \alpha(y)\Big)\, dy\,, \tag{30}$$

where α is a function that redistributes the mass outside of $\text{supp}(\pi_{\text{ref}})$ across $\mathbf{M}$; $\alpha(y) > 0$, $\int_{y \notin \text{supp}(\pi_{\text{ref}})} \pi(y)\, dy = \int_{\mathbf{M}} \alpha(y)\, dy$.

First, we note the reward contribution do not decrease from π (left hand side) to π' (right hand side),

$$\int_{\mathbf{M}} \pi(y) R_{\text{max}}\, dy + \int_{y \notin \text{supp}(\pi_{\text{ref}})} \pi(y) R(y)\, dy \le \int_{\mathbf{M}} \pi(y) R_{\text{max}}\, dy + \int_{\mathbf{M}} \alpha(y) R_{\text{max}}\, dy\,, \tag{31}$$

since $\int_{y \notin \text{supp}(\pi_{\text{ref}})} \pi(y)\, dy = \int_{\mathbf{M}} \alpha(y)\, dy$ and $R(y) \le R_{\text{max}}$.

Second, note the (simplified) KL contribution is strictly larger in π' (right hand side),

$$\int_{\mathbf{M}} \pi_{\text{ref}}(y) \log \pi(y)\, dy < \int_{\mathbf{M}} \pi_{\text{ref}}(y) \log \big(\pi(y) + \alpha(y)\big)\, dy\,, \tag{32}$$

since $\log$ is a strictly increasing function, and $\alpha(y) > 0$. Therefore, we have established that,

$$J_{\text{fwd}}(\pi) < J_{\text{fwd}}(\pi')\,. \tag{33}$$

That is, there always exists a more optimal solution with support solely inside of $\text{supp}(\pi_{\text{ref}})$. □

Proof of Remark 3.3 Assume the reward R is finite and some samples from within $\text{supp}(\pi_{\text{ref}})$ has $R(y) = R_{\text{max}}$. We optimize with $\beta > 0$ over the restricted feasible set Π, $\pi_\theta \in \Pi$, where $\pi(y) > 0$ almost everywhere on $\text{supp}(\pi_{\text{ref}})$ to avoid dividing by zeros.

We write the maximization objective subject to constraints $\int \pi(y)\, dy = 1$, $\pi(y) \ge 0$ for all y,

$$\begin{aligned}\mathcal{L}_J[\pi; \lambda] &= \int \pi(y) R(y) + \beta \pi_{\text{ref}}(y) \log \pi(y)\, dy + \lambda \Big(\int \pi(y)\, dy - 1 \Big) + \int \mu(y) \pi(y)\, dy\,, &(34)\\ &= \int \pi(y) R(y) + \lambda \pi(y) + \beta \pi_{\text{ref}}(y) \log \pi(y) + \mu(y) \pi(y)\, dy - \lambda\,, &(35)\end{aligned}$$

where at the optimal solution, $\mu(y) \ge 0$ and $\mu(y)\pi(y) = 0$.

We take the Gateaux derivative in any perturbation direction $\varphi(y)$, $\int \varphi(y)\, dy = 0$, $\pi(y) + \varepsilon \varphi(y) > 0$,

$$d\, \mathcal{L}_J[\pi; \lambda] = \frac{d}{d\varepsilon} \mathcal{L}_J[\pi + \varepsilon\varphi; \lambda] \Big|_{\varepsilon = 0}\,, \tag{36}$$

Defining $0 \log 0 = 0$ per convention. We first solve,

$$\begin{aligned}\frac{d}{d\varepsilon} \mathcal{L}_J[\pi + \varepsilon\varphi; \lambda] = \frac{d}{d\varepsilon} \int & \big(\pi(y) + \varepsilon\varphi(y)\big) R(y) + \lambda \big(\pi(y) + \varepsilon\varphi(y)\big) \\ & + \beta\, \pi_{\text{ref}}(y) \log \big(\pi(y) + \varepsilon\varphi(y)\big) \\ & + \mu(y) \big(\pi(y) + \varepsilon\varphi(y)\big)\, dy\,, &(37)\\ = \int & \varphi(y) R(y) + \lambda \varphi(y) + \beta \frac{\pi_{\text{ref}}(y)\, \varphi(y)}{\pi(y) + \varepsilon\varphi(y)} + \mu(y)\varphi(y)\, dy\,, &(38)\\ = \int & \varphi(y) \Big[R(y) + \lambda + \beta \frac{\pi_{\text{ref}}(y)}{\pi(y) + \varepsilon\varphi(y)} + \mu(y) \Big] dy\,. &(39)\end{aligned}$$

$$\frac{d}{d\varepsilon}\mathcal{L}_J[\pi + \varepsilon\varphi; \lambda]\bigg|_{\varepsilon=0} = \int \varphi(y)\Big[R(y) + \lambda + \beta\frac{\pi_{\text{ref}}(y)}{\pi(y)} + \mu(y)\Big]dy\,. \tag{40}$$

Define the functional derivative to be,

$$\frac{\delta}{\delta\pi}\mathcal{L}_J[\pi; \lambda] = R(y) + \lambda + \beta\frac{\pi_{\text{ref}}(y)}{\pi(y)} + \mu(y) \tag{41}$$

To find the optimum π^* which gives ${}^{d}/_{d\varepsilon}\,\mathcal{L}_J[\pi + \varepsilon\varphi; \lambda] = 0$ for all φ, the fundamental lemma of the calculus of variations (Gelfand & Fomin (1963), Lemma 1) tells us it would imply ${}^{\delta}/_{\delta\pi}\,\mathcal{L}_J[\pi; \lambda] = 0$. Solving for this,

$$R(y) + \lambda + \beta\frac{\pi_{\text{ref}}(y)}{\pi^*(y)} + \mu(y) = 0\,, \tag{42}$$

$$\Rightarrow \pi^*(y) = \frac{\beta\pi_{\text{ref}}(y)}{-\lambda - R(y) - \mu(y)}\,, \tag{43}$$

$$\Rightarrow \pi^*(y) = \frac{\beta\pi_{\text{ref}}(y)}{\Lambda - \big(R(y) + \mu(y)\big)}\,, \qquad \text{define } \Lambda = -\lambda\,. \tag{44}$$

Per our assumption that some max reward samples are within $\text{supp}(\pi_{\text{ref}})$, Remark B.1 states $\pi^*(y) = 0$ for all $y \notin \text{supp}(\pi_{\text{ref}})$. We can thus ignore the $\pi_{\text{ref}}(y) = 0$ regions. Further observe $\pi_{\text{ref}}(y) > 0$ implies $\pi^*(y) > 0$, thus $\mu(y) = 0$ (per $\pi(y)\mu(y) = 0$). The optimal distribution is therefore,

$$G_{\text{fwd}}(y) = \frac{\beta\pi_{\text{ref}}(y)}{\Lambda - R(y)}\,, \qquad \Lambda > R_{\text{max}}\,, \tag{45}$$

where Λ is the unique solution to $\int \beta\pi_{\text{ref}}(y)/(\Lambda - R(y))\,dy = 1$. To see this solution exists, observe as $\Lambda \to R_{\text{max}}$, G_{fwd} at this point goes to infinity. On the other hand, as $\Lambda \to \infty$, all $G_{\text{fwd}} * (y) \to 0$. By continuity, some Λ exists between R_{max} and ∞ which satisfy normalization to 1.

Note Grill et al. (2020), Appendix B.3 arrives at a similar solution for the setting of discrete action spaces (i.e. π_θ is a vector).

When does G_{fwd} have mass outside of supp(π_{ref})? Interestingly, when regularizing with the forward KL, there *are* cases where the optimal distribution G_{fwd} puts probability density on regions *outside* of the support of π_{ref}. First, note that when $\pi_{\text{ref}}(y) = 0$, the KL penalty is zero. We can use this to solve for a simplified version of Equation 42,

$$R(y) + \lambda + \mu(y) = 0\,, \tag{46}$$

$$\Rightarrow \Lambda = R(y) + \mu(y)\,, \qquad \Lambda = -\lambda\,. \tag{47}$$

This implies a few possible scenarios for regions where $\pi_{\text{ref}}(y) = 0$,

- If $R(y) < \Lambda$, then $\mu(y) > 0$, implying $\pi(y) = 0$ to respect $\mu(y)\pi(y) = 0$,
- If $R(y) = \Lambda$, then $\mu(y) = 0$, meaning $\pi(y)$ can be positive,
- $R(y) > \Lambda$ is impossible, as $\mu(y) \geq 0$.

Denote $R^{\text{in}}_{\text{max}} = \max_{y\in\text{supp}(\pi_{\text{ref}})} R(y)$ as the on-support max reward, and $R^{\text{out}}_{\text{max}} = \max_{y\notin\text{supp}(\pi_{\text{ref}})} R(y)$ as the off-support max reward. Per Remark B.1, G_{fwd} will never leave the support of π_{ref} as long as $R^{\text{in}}_{\text{max}} \geq R^{\text{out}}_{\text{max}}$. We therefore consider the case where better samples can be found outside of $\text{supp}(\pi_{\text{ref}})$, $R^{\text{out}}_{\text{max}} > R^{\text{in}}_{\text{max}}$.

Denote an integral over π_{ref} as,

$$Z(c) = \int_{\text{supp}(\pi_{\text{ref}})} \frac{\beta\pi_{\text{ref}}(y)}{c - R(y)}\,dy\,. \tag{48}$$

Now consider the off-support set with constant max rewards: $\mathbf{M}' = \{y \notin \text{supp}(\pi_{\text{ref}}) : R(y) = R^{\text{out}}_{\text{max}}\}$. Recall this set has higher reward than anything within the support of π_{ref}, $R^{\text{out}}_{\text{max}} > R^{\text{in}}_{\text{max}}$. If $Z(R^{\text{out}}_{\text{max}}) < 1$, $\Lambda < R^{\text{out}}_{\text{max}}$ violates impossibility of $R(y) > \Lambda$ above, while $\Lambda > R^{\text{out}}_{\text{max}}$ implies no mass can be placed off support, without normalization on-support ($Z(\Lambda) < 1$). Thus, the only valid solution is $\Lambda = R^{\text{out}}_{\text{max}}$, with the leftover $1 - Z(R^{\text{out}}_{\text{max}})$ probability mass allocated to $\mathbf{M}'$. On the other hand, if $Z(R^{\text{out}}_{\text{max}}) \geq 1$, it implies some $\Lambda \geq R^{\text{out}}_{\text{max}}$ exists which normalizes the on-support distribution and no mass is placed off π_{ref}'s support.

B.4 Gradient of Forward-KL Regularized Reward Maximization

Proof of Remark 3.4 We want to know if optimizing the forward-KL *regularized* RL objective within the support of π_{ref} is equivalent to optimizing a forward KL divergence. In other words, we are interested in whether the following gradient,

$$\nabla_\theta \, J_{\text{fwd}}(\pi_\theta) = \nabla_\theta \Big[\mathbb{E}_{\pi_\theta(y)}[R(y)] - \beta \, D_{KL}\big(\pi_{\text{ref}} || \pi_\theta\big) \Big] \,, \tag{49}$$

is a gradient of a forward KL between π_θ and *some* target distribution h that is independent of π_θ. We prove by contradiction. Suppose h exists, it follows that the functional derivative of these two objectives must be equivalent up to proportionality,

$$\frac{\delta}{\delta\pi} J_{\text{fwd}}(\pi) \propto \frac{\delta}{\delta\pi} D_{KL}\big(h||\pi\big) \,, \tag{50}$$

where both are subject to constraint $\int \pi(y)\, dy = 1$.

We have established from Equation 41 that the functional derivative of J_{fwd} subject to constraint $\int \pi(y)\, dy = 1$ is,

$$\frac{\delta}{\delta\pi} \mathcal{L}_J[\pi; \lambda] = R(y) + \beta \frac{\pi_{\text{ref}}(y)}{\pi(y)} + \lambda \,. \tag{51}$$

To find the functional derivative of the forward-KL, we first write down the forward KL objective subject to constraint,

$$\begin{aligned}
\mathcal{L}_K[\pi, \lambda'] &= D_{KL}\big(h||\pi\big) + \lambda' \Big(\int \pi(y)\, dy - 1 \Big) \,, && (52)\\
&= \int h(y) \log h(y) - h(y) \log \pi(y) \, dy + \int \lambda' \pi(y) \, dy - \lambda' \,, && (53)\\
&= \int \lambda' \, \pi(y) - h(y) \log \pi(y) \, dy + \Big[\int h(y) \log h(y) \, dy - \lambda' \Big] \,, && (54)
\end{aligned}$$

where the right-hand bracket is independent of π. The Gateaux derivative is,

$$\begin{aligned}
\frac{d}{d\varepsilon} \mathcal{L}_K[\pi + \varepsilon\varphi, \lambda'] &= \frac{d}{d\varepsilon} \int \lambda' \big(\pi(y) + \varepsilon\varphi(y)\big) - h(y) \log \big(\pi(y) + \varepsilon\varphi(y)\big) \, dy \,, && (55)\\
&= \int \lambda' \varphi(y) - \frac{h(y)\varphi(y)}{\pi(y) + \varepsilon\varphi(y)} \, dy \,. && (56)
\end{aligned}$$

$$\frac{d}{d\varepsilon} \mathcal{L}_K[\pi + \varepsilon\varphi, \lambda'] \bigg|_{\varepsilon=0} = \int \varphi(y) \Big[\lambda' - \frac{h(y)}{\pi(y)} \Big] \, dy \tag{57}$$

The functional derivative of the forward KL with respect to the right-hand term is therefore,

$$\frac{\delta}{\delta\pi} \mathcal{L}_K[\pi, \lambda'] = \lambda' - \frac{h(y)}{\pi(y)} \,. \tag{58}$$

Assuming the functional derivative of the two objectives are proportional to each other, we can solve for the target distribution $h(y)$,

$$\begin{aligned}
&\frac{\delta}{\delta\pi} \mathcal{L}_K[\pi, \lambda'] \propto \frac{\delta}{\delta\pi} \mathcal{L}_J[\pi, \lambda] \,, && (59)\\
&\Rightarrow \lambda' - \frac{h(y)}{\pi(y)} = \alpha \Big[R(y) + \beta \frac{\pi_{\text{ref}}(y)}{\pi(y)} + \lambda \Big] \,, \qquad \text{for some constant } \alpha \,, && (60)\\
&\Rightarrow h(y) = \Big(\lambda' - \alpha\lambda - \alpha R(y) \Big) \pi(y) - \alpha\beta\pi_{\text{ref}}(y) \,. && (61)
\end{aligned}$$

Observe one cannot write $h(y)$ independently of $\pi(y)$, other than in trivial cases (e.g. if $R(y)$ is constant such that const $- R(y) = 0$). Thus, for general reward functions R, optimizing the forward-KL does not produce a forward KL gradient toward any distribution that can be expressed independently of π_θ.

B.5 Gradient of the forward KL

Remark B.2. *The gradient of the forward KL divergence between policy π_θ and target G_β is,*

$$\nabla_\theta\, D_{KL}\big(G_\beta || \pi_\theta\big) = -\mathbb{E}_{G_\beta}\left[\nabla_\theta \log \pi_\theta(y)\right] . \tag{62}$$

Proof.

$$\nabla_\theta\, D_{KL}\big(G_\beta || \pi_\theta\big) = \nabla_\theta \mathbb{E}_{G_\beta}\left[\log G_\beta(y) - \log \pi_\theta(y)\right] , \tag{63}$$

$$= \mathbb{E}_{G_\beta}\left[\nabla_\theta\big(\log G_\beta(y) - \log \pi_\theta(y)\big)\right] , \tag{64}$$

$$= -\mathbb{E}_{G_\beta}\left[\nabla_\theta \log \pi_\theta(y)\right] . \tag{65}$$

□

We see that optimizing the forward KL gradient amounts to doing maximum likelihood / supervised fine-tuning on trajectories sampled from the target distribution G_β, as is also mentioned in some previous works (Agarwal et al., 2024). This is generally intractable as it requires sampling from G_β. Nevertheless, estimating expectation under a distribution known only up to normalization (i.e. $\mathbb{E}_{G_\beta}[\cdot]$) is well-studied in Monte-Carlo methods (Robert et al., 1999), and it is conceivable that a number of methods there would prove helpful here.

B.6 Probability Ratio Under Optimal Target Distribution

Proof of Proposition 4.1 For any two samples, y_1 and y_2, their probability ratio under the target distribution is given by,

$$\frac{G_\beta(y_1)}{G_\beta(y_2)} = \frac{g_\beta(y_1)}{\zeta} \frac{\zeta}{g_\beta(y_2)} = \frac{g_\beta(y_1)}{g_\beta(y_2)} , \tag{66}$$

which only require the unnormalized likelihood as the normalization constant ζ cancel out. Expanding the terms, we can write the log likeilhood ratio in closed form,

$$\log \frac{G_\beta(y_1)}{G_\beta(y_2)} = \log \pi_{\text{ref}}(y_1) \exp\big(\frac{R(y_1)}{\beta}\big) - \log \pi_{\text{ref}}(y_2) \exp\big(\frac{R(y_2)}{\beta}\big) , \tag{67}$$

$$= \log \frac{\pi_{\text{ref}}(y_1)}{\pi_{\text{ref}}(y_2)} + \frac{1}{\beta}\Big(R(y_1) - R(y_2)\Big) . \tag{68}$$

B.7 Target distribution after reward augmentation

Remark B.3. *Optimizing the reverse-KL regularized RL objective with the augmented reward function $\bar{R}$ yields the following target distribution, which puts uniformly high mass over all samples above reward threshold $R(y) \geq \tau$,*

$$\bar{G}_\beta(y) \propto \begin{cases} \pi_{\textit{ref}}(y) \exp\left(\frac{R(y)}{\beta}\right) & \textit{if } R(y) < \tau, \\ \pi_{\textit{ref}}(z) \exp\left(\frac{R(z)}{\beta}\right) & \textit{if } R(y) \geq \tau. \end{cases} \tag{69}$$

Proof. We have established already in Appendix B.1 that the target distribution of reward maximization with reverse KL regularization is,

$$G_\beta(y) \propto \pi_{\text{ref}}(y) \exp\big(\frac{R(y)}{\beta}\big) . \tag{70}$$

Plug in the augmented reward function,

$$\bar{R}(y) = \begin{cases} R(y) & \text{if } R(y) < \tau, \\ R(z) + \beta\big(\log \pi_{\text{ref}}(z) - \log \pi_{\text{ref}}(y)\big) & \text{if } R(y) \geq \tau, \end{cases} \tag{71}$$

which gives us the augmented target distribution,

$$\bar{G}_\beta(y) \propto \pi_{\text{ref}}(y) \exp\big(\frac{\bar{R}(y)}{\beta}\big) . \tag{72}$$

In the $R(y) < \tau$ case, $\bar{R}(y) = R(y)$, and there is no change to the (unnormalized) likelihood. In the $R(y) \geq \tau$ case,

$$\log \pi_{\text{ref}}(y) \exp\big(\frac{\bar{R}(y)}{\beta}\big) = \log \pi_{\text{ref}}(y) + \frac{1}{\beta}\bar{R}(y)\,, \tag{73}$$

$$= \log \pi_{\text{ref}}(y) + \frac{1}{\beta}\Big(R(z) + \beta\big(\log \pi_{\text{ref}}(z) - \log \pi_{\text{ref}}(y)\big)\Big)\,, \tag{74}$$

$$= \frac{R(z)}{\beta} + \log \pi_{\text{ref}}(y) + \log \pi_{\text{ref}}(z) - \log \pi_{\text{ref}}(y) \tag{75}$$

$$= \frac{R(z)}{\beta} + \log \pi_{\text{ref}}(z)\,. \tag{76}$$

Therefore we see in the $R(y) \geq \tau$ case we have,

$$\pi_{\text{ref}}(y) \exp\big(\frac{\bar{R}(y)}{\beta}\big) = \pi_{\text{ref}}(z) \exp\big(\frac{R(z)}{\beta}\big)\,. \tag{77}$$

□

This formally shows the target will have uniformly high density proportional to $\pi_{\text{ref}}(z)\exp({}^{R(z)}/_{\beta})$ for all samples if their original reward $R(y)$ is above threshold τ. If we pick z to be likely under π_{ref}, e.g. $z = \arg\max_y \pi_{\text{ref}}(y)$, we can also see these samples will have the highest probabilities in the target distribution.

B.8 Gradient of reward-augmented optimization

We also note the MARA gradient estimator for an "above threshold" sample y_i (i.e. $R(y_i) \geq \tau$), when using reverse-KL regularization, can be equivalently constructed as using the anchor sample's reference policy probability $\pi_{\text{ref}}(z)$ in lieu of the actual reference probability $\pi_{\text{ref}}(y_i)$ when constructing the KL gradient estimator. To see this precisely, we know the gradient of the expected reward to be,

$$\nabla_\theta\, \mathbb{E}_{\pi_\theta}\big[R(y)\big] = \mathbb{E}_{\pi_\theta}\big[R(y)\,\nabla_\theta \log \pi_\theta(y)\big]\,, \tag{78}$$

and gradient of the reverse-KL regularizer to be,

$$\nabla_\theta\, D_{KL}\big(\pi_\theta || \pi_{\text{ref}}\big) = \mathbb{E}_{\pi_\theta}\big[(\log \pi_\theta(y) - \log \pi_{\text{ref}}(y))\nabla_\theta \log \pi_\theta(y)\big]\,. \tag{79}$$

Denote the reward-augmented objective as $\bar{J}_\beta(\pi_\theta) = \bar{R}(y) - \beta D_{KL}(\pi_\theta||\pi_{\text{ref}})$, where $\bar{R}(y) = R(z) + \beta\big(\log \pi_{\text{ref}}(z) - \log \pi_{\text{ref}}(y)\big)$ and z is the "anchor". The gradient estimator of $\bar{K}_i$ for an "above threshold" sample, $y_i, R(y_i) \geq \tau$, can be written as,

$$\bar{K}_i = \Big(\bar{R}(y_i) - \beta \log \frac{\pi_\theta(y_i)}{\pi_{\text{ref}}(y_i)}\Big)\nabla_\theta \log \pi_\theta(y_i)\,, \tag{80}$$

$$= \Big(R(z) + \beta \log \frac{\pi_{\text{ref}}(z)}{\pi_{\text{ref}}(y_i)} - \beta \log \frac{\pi_\theta(y_i)}{\pi_{\text{ref}}(y_i)}\Big)\nabla_\theta \log \pi_\theta(y_i)\,, \tag{81}$$

$$= \Big(R(z) - \beta \log \frac{\pi_\theta(y_i)}{\pi_{\text{ref}}(z)}\Big)\nabla_\theta \log \pi_\theta(y_i)\,. \tag{82}$$

Intuitively, as the anchor is chosen to have high π_{ref}, i.e. $\pi_{\text{ref}}(z) > \pi_{\text{ref}}(y_i)$, this can be interpreted as selectively reducing the KL regularization for high-rewarding samples. Mechanistically, this also suggest an alternative implementation which produces the same gradient when using reverse-KL regularization (Algorithm 2).

C Additional Experimental Details

C.1 Didactic Experiments

We construct our didactic experiment as a vector of size 100 (akin to a output space with 100 tokens). We initialize a categorical distribution over this output space whose logits are all 0's (i.e. uniform distribution over all tokens). Given some reward function and reference distribution defined over this space, we optimize this categorical distribution with the KL-regularized policy gradient for 3000 gradient steps in PyTorch with Adam optimizer, with learning rate 5e-3 and batch size 32.

Algorithm 2 Mode Anchored Reward Augmentation, alternative implementation. The gradient of this algorithm is equivalent to Algorithm 1 when using reverse-KL regularization.

```
Given: initial policy $\pi_\theta$, reference distribution $\pi_{\text{ref}}$, reward function $R$, regularization coefficient
    $\beta$, threshold of good answers $\tau \in \mathbb{R}$, $\tau \leq \max_y R(y)$, and trajectory batch $\{y_i\}_{i=1}^N \sim \pi_\theta$.
Pick anchor trajectory: $z = \arg\max_{y_i} \pi_{\text{ref}}(y_i)$, s.t. $R(y_i) \geq \tau$
for each $y_i$ in batch do
    if $R(y_i) \geq \tau$ then
        Augment: reward $\bar{r}_i = R(z)$, reference prob $\bar{p}_i = \pi_{\text{ref}}(z)$
    else
        Keep same: reward $\bar{r}_i = R(y_i)$, reference prob $\bar{p}_i = \pi_{\text{ref}}(y_i)$
    end if
end for
Optimize policy parameters $\theta$ using augmented rewards $\{\bar{r}_i\}_{i=1}^N$ and augmented reference policy
    probabilities $\{\bar{p}_i\}_{i=1}^N$
```

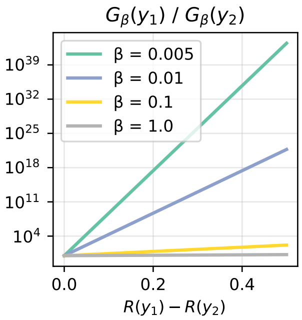


Figure 7: Effect of reward difference (ΔR, x-axis) and reverse-KL regularization strength (hue) on the relative probabilities between two samples in the optimal policy distribution (y-axis)

C.2 THE 1-2 TASK

We ask the LM to generate a uniform random integer that is either 1 or 2 (Hopkins et al., 2023), as illustrated in Figure 9. We run for a range of KL coefficients (β) and multiple random seeds. Figure 10 shows the training run for just vanilla RL, without MARA.

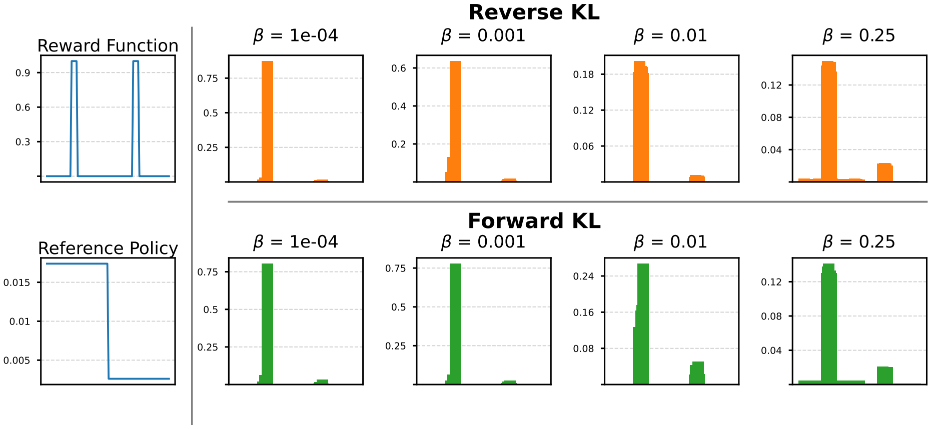


Figure 8: Final policy distribution after KL-regularized RL, with equal rewards for all correct answers. Low-support (yet equally correct) answers are never preferred over high-support answers.

Prompt

```
Uniformly randomly generate
an integer that is either 1 or
2.  Respond strictly in this
format:  <think>Your internal
reasoning</think><answer>1 or
2</answer>
```

Example Generation

```
 Let me decide
 randomly.
<think></think><answer>1
</answer><|endoftext|>
```

Figure 9: The 1-2 task to test output distribution of LMs.

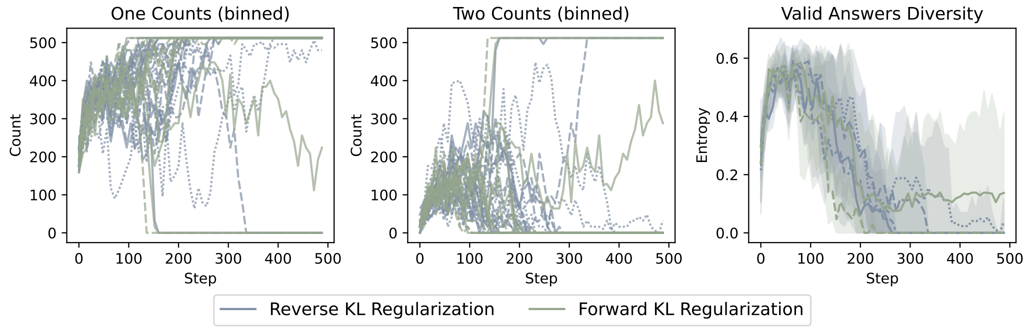


Figure 10: Training outcomes using vanilla RL. **(Left, Middle)** Policy's empirical distribution over valid answers for runs that reached high rewards (counts binned over 8 consecutive training batches), across a range of regularization coefficients (β). **Right** Diversity of the valid answers over the course of training, measured as the entropy of the Bernoulli distribution over answers of 1's and 2's.

C.3 Creative Question Answering Task

We detail the training settings in Table 3, and evaluation settings in Table 4. We follow the evaluation procedures outlined in both Kirk et al. (2023) and Zhang et al. (2025). The specific evaluation metrics are defined as follows.

- `In Dist Reward`: training reward, on training set, using training reward model
- `Out Dist Reward`: evaluation reward on held-out set, using evaluation reward model
- `Ngram EAD`: Expectation-adjusted Distinct N-gram, proposed in Liu et al. (2022). We follow (Kirk et al., 2023) and average EAD for $n = 1, ..., 5$
- `Semantic Div`: semantic embedding diversity as measured by averaged cosine distance, using embedding model `all-MiniLM-L6-v2`.
- `Mean Distinct`: Estimates a notion of "# of distinct concepts", as introduced in Zhang et al. (2025).

We run additional baselines for the effect of the batch level threshold to set τ in Table 5.

C.4 Evidence for unimodal target distributions in LMs

We show additional evidence in existing LLMs settings, the shape of the reference distribution and reward function leads to highly skewed target distributions.

First, we draw 8192 samples from `Qwen2.5-3B` using prompt for the 1-2 task (Appendix C.2). We filter for correct answers ($R(y) = 1$), leaving 2944 samples (35.9% correct). We see in Figure 11a the distribution of "1" and "2", with 1 being over-represented, pointing to a skew in the base reference distribution favouring "1", despite the model being prompted to uniformly randomly generate an integer.

Hyperparameter	**Value**
Actor Model	`Qwen3-1.7B`
Reward Model	`Skywork-Reward-V2-Qwen3-4B`
Training Dataset	`Wildchat 10k English`
Train Batch Size	128
Mini-Batch Size	64
Max Prompt Length	512
Max Response Length	2048
Learning Rate	1×10^{-6}
Entropy Coefficient	0
Rollout n (per prompt)	5
Gradient Checkpointing	Enabled
Epochs	3

Table 3: Creative QA Training Setting

Hyperparameter	**Value**
Evaluation Reward Model	`Skywork-Reward-Gemma-2-27B-v0.2`
Dataset	`NoveltyBench curated`
Num Generations / Prompt	10
Max Tokens	4000
Temperature	1.0
Enable Thinking (qwen)	`False`

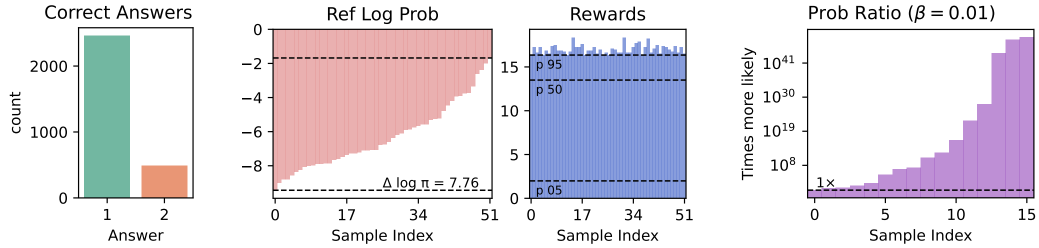
Table 4: Creative QA Evaluation Setting

For creative QA (Appendix C.3), we draw 1024 samples from `Qwen3-1.7B` using a single question in WildChat text. We evaluate the responses using `Skywork-Reward-V2-Qwen3-4B` (reward mean: 12.33, min: -5.125, max: 18.38), and filter for answers above the 95th percentile ($R = 16.37$). This leaves 52 samples (reward mean: 17.0, min: 16.38, max: 18.38). We see in Figure 11b that among these high-reward answers, $\log \pi_{\text{ref}}(y)$ has a difference of up to 7.76, corresponding to a probability ratio of 2345× under G_β (Remark 4.3).

We then estimate the probability ratio taking into account both the reference probability and the reward (Proposition 4.1). We use the same 1024 samples and compute their unnormalized likelihood G_β, $\pi_{\text{ref}}(y) \exp(R(y)/\beta)$, with $\beta = 0.01$. We take the top 16 most likely samples and calculate their relative probabilities *with respect to the 16th most likely sample*, such that the lowest probability ratio is $1\times$. We observe in Figure 11c that amongst the sampled responses, the most probable sample ($R = 18.4$, $\log \pi_{\text{ref}}(y) = -5.91$) is 2×10^{49} times more likely under the target distribution G_β than the 16th most likely sample ($R = 17.3$, $\log \pi_{\text{ref}}(y) = -7.09$), despite having only slightly higher rewards and reference probabilities.

Model	**Out-dist Reward (↑)**	**EAD (↑)**	**Semantic Div (↑)**	**Distinct (↑)**
MARA (rev, 0.90)	1.451 ±0.103	0.543 ±0.014	0.186 ±0.008	4.14 ±0.233
MARA (fwd, 0.90)	1.604 ±0.113	0.568 ±0.012	0.193 ±0.009	4.62 ±0.258
MARA (rev, 0.75)	1.498 ±0.117	0.547 ±0.013	0.183 ±0.008	4.41 ±0.262
MARA (fwd, 0.75)	1.325 ±0.097	0.508 ±0.014	0.196 ±0.009	4.07 ±0.243

Table 5: Ablation of batch-level threshold for τ, set at either 90th percentile (0.90) or 70th percentile (0.75). Mean ± bootstrap SEM.



(a) 1-2 Task correct answers distribution

(b) Reference probs $\log \pi_{\text{ref}}(y)$ of highest reward samples in Creative QA task; $\Delta \log \pi$ shows range

(c) Relative (to idx 0) probs of most likely samples under G_β

Figure 11: Evidence for highly skewed target distributions G_β in LLM tasks

C.5 Drug Discovery

Chemical language models (CLMs) that generate molecules in string-based formats, e.g., a SMILES string (Weininger, 1988), have been experimentally validated with numerous generated molecules in clinical trials (Du et al., 2024). Recently, the field has focused on addressing "synthesizability", i.e., can generated molecules actually be synthesized in the lab? (Stanley & Segler, 2023; Papidocha et al., 2025). Accordingly, we adapt two reward functions from Guo et al. (2025b): `SYNTH` and `SYNTH-ALL-AMIDE` that jointly reward binding potency and synthesizability. `REINVENT` (Olivecrona et al., 2017) is a state-of-the-art RL-based CLM on standard benchmarks (Gao et al., 2022). The recent Saturn CLM (Guo & Schwaller, 2024b) notably improves optimization efficiency by using data augmentation (Bjerrum, 2017; Guo & Schwaller, 2024a), but continues to use `REINVENT`'s RL algorithm.

In the drug discovery experiments adapted from Guo et al. (2025b), the reward functions are comprised of numerous individual optimization objectives, and defines a multi-parameter optimization task. Concretely, these objectives are:

1. *Minimize* the molecular docking score using QuickVina2-GPU (Trott & Olson, 2010; Alhossary et al., 2015; Tang et al., 2024). Docking simulates binding of molecules to a target protein and predicts a crude binding affinity value. Docking was performed against the ATP-dependent Clp protease proteolytic subunit (ClpP) Mabanglo et al. (2023).
2. *Maximize* the quantitative estimate of drug-likeness (QED) (Bickerton et al., 2012), which is itself comprised of various physico-chemical properties, e.g., molecular weight. Maximizing QED can prevent generated molecules from being too large and lipophilic.
3. *Constrain* the number of hydrogen-bond donors (HBDs): HBDs < 4. This can improve absorption, Distribution, metabolism, and excretion (ADME) properties (Kenny, 2022) of the generated molecules.
4. *Satisfy* the "Synthesizability" constraint. Synthesizability is quantified by using a retrosynthesis model on each generated molecule. Retrosynthesis models predict a plausible synthesis route to synthesize a target molecule using commercially available precursors. The precursors set is from the eMolecules catalogue extracted from Chen et al. (2020). Retrosynthesis models typically start with a "single-step" model which predicts precursors given a target molecule. Since molecules may require multiple steps to synthesize, "Multi-step Retrosynthesis" commonly couples a search algorithm with single-step models to iteratively decompose a target molecule. In this work, we use the MEGAN (Sacha et al., 2021) single-step model with the Retro* (Chen et al., 2020) search algorithm using the Syntheseus (Maziarz et al., 2025) package. Finally, a molecule is considered synthesizable if the retrosynthesis model successfully proposes a synthesis route.

Both the `SYNTH` and `SYNTH-ALL-AMIDE` reward functions are comprised of the above objectives. The only difference is that in the `SYNTH-ALL-AMIDE` case, a molecule is *only* considered synthesizable if all the chemical reactions involved to synthesize it are "amide coupling reactions". Amide couplings are one of the most common reactions performed in the pharmaceutical industry (Brown & Bostrom, 2016), and is generally a robust, widely compatible reaction. Subsequently, the reward function is defined as a product of each individual component above. Given a molecule, x:

Table 6: Results at Threshold = 0.8 (↑ larger is better; ↓ smaller is better). "SYNTH" and "AMIDE" denote the `SYNTH` and `SYNTH-ALL-AMIDE` reward functions, respectively.

Task	**Algorithm**	**Sigma**	**Gen Yield** (↑)	**OB100** (↓)	**IntDiv1** (↑)	**Circles** (↑)
SYNTH	REINVENT	128	6569 ± 186	1042 ± 66	0.766 ± 0.011	67 ± 3
		256	6618 ± 93	1080 ± 89	0.756 ± 0.012	57 ± 8
		512	6746 ± 161	1067 ± 74	0.752 ± 0.016	55 ± 5
	MARA	128	6834 ± 78	1015 ± 55	0.761 ± 0.009	59 ± 8
		256	6750 ± 139	1068 ± 50	0.760 ± 0.012	60 ± 4
		512	6793 ± 267	1065 ± 49	0.751 ± 0.015	60 ± 1
AMIDE	REINVENT	128	5433 ± 184	1427 ± 63	0.768 ± 0.012	35 ± 1
		256	5544 ± 172	1406 ± 59	0.768 ± 0.009	34 ± 5
		512	5334 ± 165	1445 ± 111	0.776 ± 0.008	33 ± 4
	MARA	128	5635 ± 249	1407 ± 123	0.766 ± 0.008	36 ± 3
		256	5353 ± 114	1393 ± 42	0.769 ± 0.009	33 ± 4
		512	5377 ± 152	1343 ± 77	0.763 ± 0.008	31 ± 3

$$R(x) = DS(x) \times QED(x) \times HBD(x) \times Syntheseus(x) \in [0, 1] \tag{83}$$

where “DS” is docking score. The HBD and Syntheseus objectives are binary, i.e., 1 if satisfied and 0 otherwise. QED $\in [0, 1]$ and is used as is. The QuickVina2-GPU docking score is reward shaped using a reverse sigmoid function following Guo et al. (2025b) and gives higher reward to lower docking scores, as desired.

Our goal in this section is to investigate the potential for MARA to be a *drop-in replacement* for the `REINVENT` (Olivecrona et al., 2017) RL-based algorithm for molecular design. `REINVENT` is amongst the most performant molecular design algorithms (Gao et al., 2022) and the Saturn model (Guo & Schwaller, 2024b) adapts this algorithm and leverages data augmentation (Bjerrum, 2017; Guo & Schwaller, 2024a) to further improve optimization efficiency.

We evaluate all models with a fixed budget of 10,000 reward function evaluations, which is standard in benchmarks. We contrast the algorithms’ performance on molecular design metrics that measure optimization efficiency and diversity. `Yield` is the number of *unique* molecules above a reward threshold. `OB100` is the number of reward evaluations required to generate 100 molecules above the same threshold. `IntDiv1` (Polykovskiy et al., 2020) and `#Circles` (Xie et al., 2023) are diversity metrics based on molecular sub-structure based features, and measure intra-set similarity and sphere packing, respectively.

Tables 6 and 7 show the optimization results for the `SYNTH` and `SYNTH-ALL-AMIDE` reward functions at the 0.80 and 0.85 screening thresholds, respectively. MARA is trained with $\tau = 0.80$ in both. In general, MARA matches or outperforms `REINVENT` particularly for the more challenging `SYNTH-ALL-AMIDE` reward function. In this environment, MARA can find more high-reward molecules (`Yield`) and using less reward evaluations (`OB100`) than `REINVENT`.

Table 7: Results at Threshold $= 0.85$ (↑ larger is better; ↓ smaller is better). "SYNTH" and "AMIDE" denote the SYNTH and SYNTH-ALL-AMIDE reward functions, respectively.

Task	**Algorithm**	**Sigma**	**Gen Yield**	**OB100**	**IntDiv1**	**Circles**
			(↑)	(↓)	(↑)	(↑)
SYNTH	REINVENT	128	1614 ± 407	4114 ± 109	0.701 ± 0.018	7 ± 1
		256	1552 ± 242	3940 ± 371	0.699 ± 0.030	6 ± 1
		512	1484 ± 45	3717 ± 201	0.701 ± 0.026	6 ± 1
	MARA	128	1796 ± 210	3654 ± 272	0.716 ± 0.015	6 ± 1
		256	1530 ± 126	3957 ± 335	0.705 ± 0.014	8 ± 1
		512	1550 ± 347	4016 ± 234	0.689 ± 0.024	6 ± 1
AMIDE	REINVENT	128	1098 ± 88	4360 ± 257	0.721 ± 0.016	8 ± 1
		256	1488 ± 280	4290 ± 141	0.725 ± 0.021	8 ± 1
		512	1054 ± 152	4620 ± 438	0.739 ± 0.009	8 ± 0
	MARA	128	1235 ± 130	3943 ± 303	0.733 ± 0.009	8 ± 1
		256	1404 ± 261	4079 ± 172	0.730 ± 0.010	7 ± 1
		512	1341 ± 86	3930 ± 400	0.723 ± 0.004	7 ± 1