Pareto-Optimal Offline Reinforcement Learning via Smooth Tchebycheff Scalarization

Anonymous authors
Paper under double-blind review

Abstract

Large language models can be aligned with human preferences through offline reinforcement learning (RL) on small labeled datasets. While single-objective alignment is well-studied, many real-world applications demand the simultaneous optimization of multiple conflicting rewards, e.g. activity and specificity for proteins, or helpfulness and harmlessness for chatbots. Prior work has largely relied on linear reward scalarization, which provably fails to recover non-convex regions of the Pareto front. In this paper, instead of scalarizing the rewards directly, we frame multi-objective RL itself as an optimization problem to be scalarized via smooth Tchebycheff scalarization, a recent technique that overcomes the drawbacks of linear scalarization. We use this formulation to derive Smooth Tchebycheff Optimization of Multi-Objective Preferences (STOMP), a novel offline RL algorithm that extends direct preference optimization to the multi-objective setting in a principled way by standardizing the individual rewards based on their observed distributions. We empirically validate STOMP using multiple base models on 3 protein engineering and 3 chatbot alignment datasets. Compared to state-of-the-art baselines, STOMP achieves or ties for the highest hypervolumes on 16/18 protein tasks and 5/6 natural language tasks. We thus demonstrate that STOMP is a powerful, robust multi-objective alignment algorithm that can meaningfully improve post-training in multiple domains.

1 Introduction

Large language models have been trained on massive corpora of text, code, images, and proteins (Brown et al., 2020; Devlin et al., 2019; Dosovitskiy et al., 2021; Madani et al., 2023; Meier et al., 2021; Radford et al., 2021; Vaswani et al., 2017). These models are powerful density estimators for the data distributions they were trained on, but they often require supervised post-training via algorithms like reinforcement learning from human feedback (RLHF) to *align* their likelihoods with the preferences of their users (Christiano et al., 2017; Ouyang et al., 2022; Ziegler et al., 2020). RL-based alignment is an incredibly general paradigm that has been used to train chatbots (Bai et al., 2022; Ouyang et al., 2022; Rafailov et al., 2023), improve text-to-image generators (Lee et al., 2024; Liang et al., 2024), and enable protein language models to generate proteins with higher fitness (Bhatnagar et al., 2025; Ibarraran et al., 2026; Wen et al., 2025; Widatalla et al., 2024).

While single-objective RL is well-studied, many real-world applications demand the simultaneous optimization of multiple conflicting objectives. For example, chatbots should be helpful to their users, but their responses should also be safe (Bai et al., 2022). Text-to-image models should generate high-quality images that are aligned with their prompts (Lee et al., 2024). Protein engineers may seek a protein with high activity and specificity (Wang et al., 2025), or a highly active protein that maintains adequate stability and expression (Armer et al., 2025). Since no single solution can in general optimize all these rewards simultaneously, our goal is instead the Pareto front, i.e. the set of all non-dominated solutions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Scalarizing the reward vector is a common approach to multi-objective RL (MORL), with linear scalarization (i.e. taking a weighted average) being the simplest strategy (Chennakesavalu et al., 2025; Li et al., 2021; Wen et al., 2025; Yang et al., 2019b; Zhou et al., 2024). Unfortunately, linear reward scalarization is provably incapable of recovering solutions from non-convex regions of the Pareto front, and solutions from these regions often represent the exact compromises between conflicting objectives that we are the most interested in discovering (Boyd and Vandenberghe, 2004; Das and Dennis, 1997; Ehrgott, 2005).

Lin et al. (2024) introduce smooth Tchebycheff scalarization (STS) to address the shortcomings of linear scalarization. STS enjoys efficient gradient-based optimization and is able to find all solutions on the Pareto front. However, STS can be challenging to use off-the-shelf for reward scalarization because it introduces many free hyperparameters (Qiu et al., 2024). In this work, rather than using STS to scalarize the rewards directly, we observe that MORL is itself a multi-objective optimization problem to be scalarized. We thus apply STS to the MORL problem and use this framing to derive a reward scalarization that dynamically standardizes the rewards based on their observed distributions in an offline training set.

Building on these insights, we introduce Smooth Tchebycheff Optimization of Multi-Objective Preferences (STOMP), an extension of direct preference optimization (Rafailov et al., 2023) that aligns language model likelihoods using an offline dataset that associates each sequence to vector-valued rewards. We empirically evaluate STOMP by aligning three protein language models on three experimental laboratory datasets of protein fitness, and two instruction-tuned language models on three chatbot alignment datasets. Compared to state-of-the-art baselines that use either linear scalarization or a more naive implementation of STS, we find that STOMP achieves (or ties for) the highest hypervolumes in 16/18 protein evaluation settings and 5/6 natural language evaluation settings.

2 Preliminaries and Related Work

2.1 Reinforcement Learning

Given context $x \in \mathcal{X}$, a language model or policy π predicts the conditional probability $\pi(y|x)$ of sequences $y \in \mathcal{Y}$. Most reinforcement learning (RL) formulations for language models optimize the expected reward of sequences generated from π while simultaneously requiring that π does not deviate too much from a reference π_0, i.e. $\mathbb{D}_{\mathrm{KL}}(\pi \,||\, \pi_0) \leq \eta$. π_0 is typically the pre-trained or fine-tuned language model that we are further training with RL (Christiano et al., 2017; Schulman et al., 2015; 2017; Ziegler et al., 2020). Concretely, given a distribution over contexts $\mathcal{P}$ and a reward function $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we seek to solve

$$
\begin{aligned}
&\operatorname*{argmax}_{\pi} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} [r(x,y)] \quad \text{s.t.} \quad \mathbb{D}_{\mathrm{KL}}(\pi \,||\, \pi_0) \leq \eta \\
=&\operatorname*{argmax}_{\pi} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} [r(x,y)] - \beta\mathbb{D}_{\mathrm{KL}}(\pi \,||\, \pi_0) \\
=&\operatorname*{argmin}_{\pi} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} [\beta \log \pi(y|x) - \beta \log \pi_0(y|x) - r(x,y)] \\
=&\operatorname*{argmin}_{\pi} \mathbb{D}_{\mathrm{KL}}(\pi \,||\, \pi^\star)
\end{aligned}
$$

Faury et al. (2020, Lemma 5) prove that the optimal Lagrange multiplier is $\beta = \sqrt{\mathbb{V}_{\pi_0}[r(x,y)]/2\eta}$. Meanwhile, $\pi^\star$ is the optimal policy and $Z(x)$ is its partition function:

$$
\pi^\star(y|x) = \frac{1}{Z(x)}\pi_0(y|x)\exp\left(r(x,y)/\beta\right) \quad \text{and} \quad Z(x) = \sum_{y\in\mathcal{Y}} \pi_0(y|x)\exp\left(r(x,y)/\beta\right)
$$

Maximum-entropy reinforcement learning (MaxEntRL) uses a uniform reference $\pi_0(y|x) \propto 1$ (Toussaint, 2009; Ziebart et al., 2008). MaxEntRL favors more stochastic policies with more diverse generations, and it has certain robustness guarantees (Eysenbach and Levine, 2022).

2.2 Offline Reinforcement Learning via Preference Optimization

RL is challenging because it requires us to optimize the expected reward as sampled from the policy π we are training. Direct preference optimization (DPO) (Rafailov et al., 2023) solves $r(x,y) = \beta \log(\pi^\star(y|x)/\pi_0(y|x)) + \beta \log Z(x)$ and assumes the Bradley-Terry model (Bradley and Terry, 1952) $p(y_1 \succ y_2|x) = \sigma(r(x,y_1) - r(x,y_2))$ to recast this challenging RL problem as the more tractable maximum likelihood estimation problem of minimizing the following loss on an offline dataset:

$$\mathcal{L}_{\text{DPO}}(\pi;\beta,\delta) = \sum_{\substack{x,y_w,y_l \\ r(x,y_w)>r(x,y_l)+\delta}} -\log\sigma\left(\beta\log\frac{\pi(y_w|x)}{\pi_0(y_w|x)} - \beta\log\frac{\pi(y_l|x)}{\pi_0(y_l|x)}\right) \tag{1}$$

Here, y_w is a "winner" and y_l is a "loser" according to the reward r, while δ ensures that we only compare samples with a sufficiently large reward difference. SimPO (Meng et al., 2024), CPL (Hejna et al., 2024), and α-DPO (Wu et al., 2025) use a uniform reference $\pi_0(y|x) \propto 1$ and can be considered instantiations of MaxEntRL in the DPO framework.

Recent works enhance DPO by explicitly using the reward difference $r(x,y_w) - r(x,y_l)$ as a margin into similar paired losses (Amini et al., 2024; Chennakesavalu et al., 2025; Fisch et al., 2025; Gao et al., 2024; Mao et al., 2024). These methods leverage the difference

$$\beta\log\frac{\pi(y_w|x)}{\pi(y_l|x)} - \beta\log\frac{\pi^\star(y_w|x)}{\pi^\star(y_l|x)} = \underbrace{\left(\beta\log\frac{\pi(y_w|x)}{\pi_0(y_w|x)} - r(x,y_w)\right)}_{S^\pi(x,y_w;\beta)} - \underbrace{\left(\beta\log\frac{\pi(y_l|x)}{\pi_0(y_l|x)} - r(x,y_l)\right)}_{S^\pi(x,y_l;\beta)}$$

to match the learned π to the optimal $\pi^\star$ and can be theoretically grounded in various ways. For example, squared loss methods (Fisch et al., 2025; Gao et al., 2024; Mao et al., 2024) minimize $(S^\pi(x,y_w;\beta) - S^\pi(x,y_l;\beta) + \delta)^2$, while OffsetDPO (Amini et al., 2024) minimizes

$$\mathcal{L}_{\text{ODPO}}(\pi;\beta,\delta) = \sum_{\substack{x,y_w,y_l \\ r(x,y_w)>r(x,y_l)+\delta}} -\log\sigma\left(S^\pi(x,y_w;\beta) - S^\pi(x,y_l;\beta) + \delta\right) \tag{2}$$

2.3 Multi-Objective Optimization

Multi-objective optimization minimizes k distinct losses $\boldsymbol{L} = L_1,\ldots,L_k$ simultaneously. Often, these losses conflict with each other and cannot be simultaneously optimized by a single solution. We instead say that $\boldsymbol{L}(z') \prec \boldsymbol{L}(z)$ (i.e. z' *Pareto-dominates* z) if $L_i(z') \leq L_i(z)$ for all i and $L_i(z') < L_i(z)$ for at least one i. The *Pareto front* of optimal solutions is the set of non-dominated solutions, i.e. $\{z \in \mathcal{Z} : \boldsymbol{L}(z') \not\prec \boldsymbol{L}(z)\ \forall\ z' \in \mathcal{Z}\}$ (Miettinen, 1999).

Scalarization is a common approach to multi-objective optimization, with linear scalarization being the simplest strategy, i.e. $\min_z \sum_{i=1}^k \lambda_i L_i(z)$ for some preference vector $\lambda \in \Delta^{k-1}$ on the simplex (Geoffrion, 1967; Miettinen, 1999). Without loss of generality, we assume $\lambda_i > 0$ for all i. However, linear scalarization will fail to find any solutions from non-convex regions of the Pareto front (Boyd and Vandenberghe, 2004; Das and Dennis, 1997; Ehrgott, 2005).

Tchebycheff scalarization (TS) lets $L_i^\star = \min_z L_i(z)$ and solves $\min_z \max_i \lambda_i(L_i(z) - L_i^\star)$. Under mild assumptions, it can find all solutions on the Pareto front by varying λ (Bowman, 1976; Choo and Atkins, 1983; Steuer and Choo, 1983). However, the min-max formulation is non-differentiable and can be challenging to optimize in practice (Goffin, 1977). Lin et al. (2024) introduce smooth Tchebycheff scalarization (STS), which instead optimizes $\min_z \tau \log \sum_{i=1}^k \exp(\lambda_i(L_i(z) - L_i^\star)/\tau)$. Here, τ is a smoothing parameter, and STS approaches TS as $\tau \to 0^+$. STS has many of the same favorable properties as classical TS while being more amenable to gradient-based optimization due to its smoothness.

3 Multi-Objective Offline Reinforcement Learning

In multi-objective (KL-constrained or MaxEnt) RL, we wish to simultaneously optimize k reward functions $r_1,\ldots,r_k$. Letting $\eta = 1/(2\gamma^2)$, this is equivalent to the optimization

$$\operatorname*{argmax}_{\pi} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} [r_i(x,y)]_{i=1}^k \quad \text{s.t.} \quad \mathbb{D}_{\mathrm{KL}}(\pi \,||\, \pi_0) \le 1/(2\gamma^2) \tag{3}$$

$$= \operatorname*{argmin}_{\pi} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} [\log \pi(y|x) - \log \pi_i^\star(y|x)]_{i=1}^k \tag{4}$$

$$\pi_i^\star(y|x) = \frac{1}{Z_i(x)} \pi_0(y|x) \exp\left(\frac{r_i(x,y)}{\gamma\sqrt{\mathbb{V}_{\pi_0}[r_i(x,y)]}}\right) \tag{5}$$

In practice, we estimate $\sigma_i^2 = \mathbb{V}_{\pi_0}[r_i(x,y)]$ as the empirical variance of $r_i(x,y)$ on the full training set. Most recent MORL work linearly scalarizes the reward to reduce the problem to single-objective RL (Chennakesavalu et al., 2025; Li et al., 2021; Wen et al., 2025; Yang et al., 2019b; Zhou et al., 2024). It is straightforward to show that this is equivalent to linearly scalarizing Optimization Problem 4. Others consider alternative reward scalarizations including STS, but they do not directly correspond to alternative scalarizations of Optimization Problem 4 (Hayes et al., 2022; McCarter et al., 2026; Qiu et al., 2024; Roijers et al., 2013; Van Moffaert et al., 2013; Williams, 2024).

3.1 Smooth Tchebycheff Reward Scalarization

We innovate on prior work by applying smooth Tchebycheff scalarization (STS) (Lin et al., 2024) directly to Optimization Problem 4. Our goal is to derive a scalarized reward R with the property that an R-optimal policy simultaneously minimizes an upper bound on its KL divergence from multiple optimal policies $\pi_1^\star, \ldots, \pi_k^\star$, one for each reward $r_1, \ldots, r_k$:

$$\min_{\pi} \tau \log \sum_{i=1}^k \exp\left(\frac{\lambda_i}{\tau} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} [\log \pi(y|x) - \log \pi_i^\star(y|x)]\right)$$

$$\le \min_{\pi} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} \left[\tau \log \sum_{i=1}^k \exp\left(\frac{\lambda_i}{\tau}(\log \pi(y|x) - \log \pi_i^\star(y|x))\right)\right]$$

$$= \min_{\pi} \lambda_{\min} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} \left[\log \pi(y|x) - \frac{\tau}{\lambda_{\min}} \log \left(\sum_{i=1}^k \frac{\pi(y|x)^{(\lambda_i-\lambda_{\min})/\tau}}{\pi_i^\star(y|x)^{\lambda_i/\tau}}\right)^{-1}\right] \tag{6}$$

$$\le \min_{\pi} \lambda_{\min} \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{P}\\ y\sim\pi(\cdot|x)}} \left[\log \pi(y|x) - \frac{\tau}{\lambda_{\min}} \log \left(\sum_{i=1}^k \frac{1}{\pi_i^\star(y|x)^{\lambda_i/\tau}}\right)^{-1}\right] \tag{7}$$

The first upper bound is due to Jensen's inequality, while the second follows from the fact that $\pi(y|x)^{(\lambda_i-\lambda_{\min})/\tau} \le 1$, with equality when $\lambda_1 = \cdots = \lambda_k = 1/k$. To avoid problematic dependencies between the scalarized reward and a potentially sub-optimal reference π_0, we let $\pi_i^\star(y|x) \propto \exp(r_i(x,y)/(\gamma\sigma_i))$ be the optimal policy under MaxEntRL.[1] Then, Optimization Problem 6 is a KL minimization that is equivalent to MaxEntRL with reward

$$R_{\mathrm{ST}}^{\lambda,\pi}(x,y;\gamma) = -\frac{\tau\gamma}{\lambda_{\min}} \log \sum_{i=1}^k \exp\left(\frac{\lambda_i-\lambda_{\min}}{\tau} \log \pi(y|x) - \frac{\lambda_i}{\tau}\left(\frac{r_i(x,y)}{\gamma\sigma_i} - \log Z_i(x)\right)\right) \tag{8}$$

Similarly, the relaxed Optimization Problem 7 is equivalent to MaxEntRL with reward

$$R_{\mathrm{ST}}^{\lambda}(x,y;\gamma) = -\frac{\tau\gamma}{\lambda_{\min}} \log \sum_{i=1}^k \exp\left(-\frac{\lambda_i}{\tau}\left(\frac{r_i(x,y)}{\gamma\sigma_i} - \log Z_i(x)\right)\right) \tag{9}$$

The policy-independent $R_{\mathrm{ST}}^{\lambda}$ is useful in settings where the dependency of $R_{\mathrm{ST}}^{\lambda,\pi}$ on the training policy π could cause training instability. In practice, we use the training set $\{(x_m, y_m^{(n)}) : m \in [M], n \in [N_m]\}$ to estimate $\log \hat{Z}_i(x_m) = \log \sum_{n=1}^{N_m} \exp\left(r_i(x_m, y_m^{(n)})/(\gamma\sigma_i)\right)$.

[1] We also ran initial experiments with a variant of this reward scalarization derived from KL-constrained RL, i.e. $\pi_i^\star(y|x) \propto \pi_0(y|x)\exp(r_i(x,y)/(\gamma\sigma_i))$. It significantly underperformed π_0-independent reward scalarization because π_0 introduces high variance when estimating $Z_i(x)$.

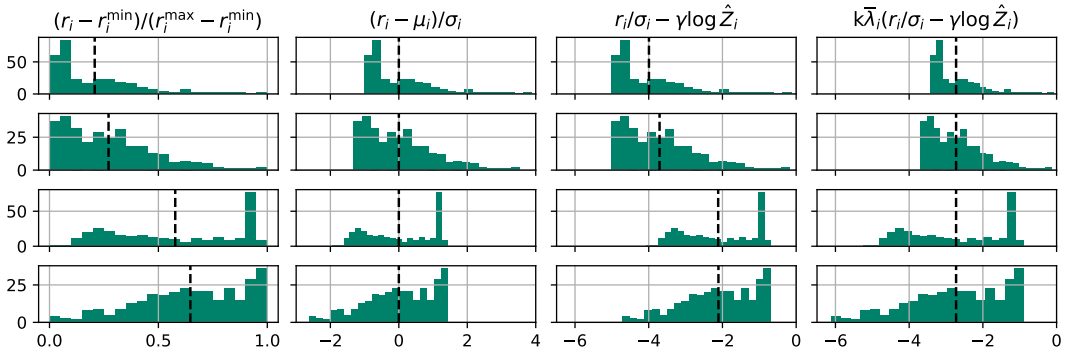

Figure 1: Visualizing different reward standardizations. The vertical dashed line indicates the mean of each distribution. Column 1: min-max standardization. Column 2: mean-variance standardization. Column 3: distribution-relative reward $\rho_i(x, y)$. Column 4: $\rho_i(x, y)$ reweighted by $\bar{\lambda}$ (Equation 10) so all $\bar{\lambda}_i \rho_i(x, y)$ have the same mean.

This estimator ensures the *distribution-relative reward* $\rho_i(x, y) = r_i(x, y)/\sigma_i - \gamma \log \hat{Z}_i(x) \leq 0$, a necessary condition for the theoretical guarantees of Lin et al. (2024). Qiu et al. (2024) instead apply STS directly to the rewards, yielding $\rho_i'(x, y) = (r_i(x, y) - \iota_i - \max_y r_i(x, y))/\kappa_i$, where the offsets ι_i and scales κ_i are free hyperparameters. Our derivation provides optimal $\iota_i = \gamma \sigma_i \log \hat{Z}_i(x) - \max_y r_i(x, y)$ and $\kappa_i = \sigma_i$ in terms of a single hyperparameter γ.

To build intuition for Equation 9, we note that $\lim_{\tau \to 0^+} \lambda_{\min} R_{\mathrm{ST}}^{\lambda}(x, y; \gamma) = \min_i \lambda_i \rho_i(x, y)$. So as $\tau \to 0^+$, sequences y with high $R_{\mathrm{ST}}^{\lambda}(x, y; \gamma)$ must simultaneously have high distribution-relative rewards $\rho_i(x, y)$ for all i. We now visualize different standardizations for 4 rewards in Figure 1. Rewards r_{1-2} are left-skewed, while r_{3-4} are right-skewed.

A good standardization should penalize the rare sequences with low r_{3-4} more than the common sequences with low r_{1-2}. Conversely, it should favor high r_{1-2} more than high r_{3-4}. Min-max (Column 1) does the opposite. Mean-variance (Column 2) achieves this goal to some extent, but the effect is limited by design and over-indexes on outliers. The Z_i's let the distribution-relative rewards ρ_i (Column 3) favor high r_{1-2} more than high r_{3-4} without placing as much weight on outliers as mean-variance standardization. Now, let

$$\bar{\lambda}_i \propto \left(\frac{1}{M} \sum_{m=1}^{M} \frac{1}{N_m} \sum_{n=1}^{N_m} -\rho_i(x_m, y_m^{(n)}) \right)^{-1} \tag{10}$$

re-weight the ρ_i's to all have the same mean. $\bar{\lambda}_i \rho_i(x, y)$ (Column 4) compresses left-skewed rewards r_{1-2} and stretches right-skewed r_{3-4} to penalize the rare sequences with low r_{3-4} more than the common sequences with low r_{1-2}. r_4 is more right-skewed than r_3, so this effect is even more pronounced. Different preferences λ achieve different tradeoffs.

3.2 Smooth Tchebycheff Optimization of Multi-Objective Preferences

To design a practical offline MORL algorithm, we define the difference

$$D_{\mathrm{ST}}^{\lambda,\pi}(x, y_w, y_l; \beta, \gamma) = \beta \left(\log \frac{\pi(y_w|x)}{\pi_0(y_w|x)} - \log \frac{\pi(y_l|x)}{\pi_0(y_l|x)} \right) - \left(R_{\mathrm{ST}}^{\lambda,\pi}(x, y_w; \gamma) - R_{\mathrm{ST}}^{\lambda,\pi}(x, y_l; \gamma) \right) \tag{11}$$

We then adapt OffsetDPO (Amini et al., 2024) (Equation 2) to get the Smooth Tchebycheff Optimization of Multi-Objective Preferences (STOMP) loss

$$\mathcal{L}_{\mathrm{STOMP}}(\pi; \alpha, \beta, \gamma, \delta, \lambda) = \sum_{\substack{x, y_w, y_l \\ R_{\mathrm{ST}}^{\lambda}(x, y_w; \gamma) > R_{\mathrm{ST}}^{\lambda}(x, y_l; \gamma) + \delta}} -\log \sigma \left(D_{\mathrm{ST}}^{\lambda,\pi}(x, y_w, y_l; \beta, \gamma) + \delta \right) - \frac{\alpha}{|y_w|} \log \pi(y_w|x) \tag{12}$$

Notably, we use the policy-independent reward $R_{\mathrm{ST}}^{\lambda}$ (Equation 9) to determine preference pairs, since having preferences evolve with the training policy π led to instability in early experiments. However, we use the policy-dependent $R_{\mathrm{ST}}^{\lambda,\pi}$ (Equation 8) in the individual loss

terms because it arises from a tighter bound to the optimization problem we are solving. We also include the length-averaged NLL of the winners in each comparison (Pang et al., 2024). This regularization is especially important when the training dataset is not sampled from the reference policy π_0, a common reality in offline RL (Fu et al., 2026; Liu et al., 2024).

4 Experiments

4.1 Methods

As baselines, we test direct preference optimization (DPO (Rafailov et al., 2023), Equation 1) and OffsetDPO (ODPO (Amini et al., 2024), Equation 2) using linear reward scalarization $\sum_{i=1}^{k} \lambda_i r_i(x,y)/\sigma_i$. We refer to these methods as DPO-Lin and ODPO-Lin (identical to Pareto-Optimal Energy Alignment by Wen et al. (2025)). Multi-objective DPO (MODPO (Zhou et al., 2024)) is a popular method similar to ODPO-Lin, but it defines preferences $y_w \succ y_l$ by a *specific* reward r_j and uses the remaining r_{-j} for the margin in the loss

$$\sum_{\substack{x,y_w,y_l \\ r_j(x,y_w)>r_j(x,y_l)+\delta}} -\log\sigma\left(\frac{\beta}{\lambda_j}\left(\log\frac{\pi(y_w|x)}{\pi_0(y_w|x)} - \log\frac{\pi(y_l|x)}{\pi_0(y_l|x)}\right) - \sum_{i\neq j}\frac{\lambda_i}{\lambda_j}\left(\frac{r_i(x,y_w)-r_i(x,y_l)}{\sigma_i}\right)\right)$$

Where DPO-style algorithms are margin maximizers, Energy Rank Alignment (ERA (Chennakesavalu et al., 2025), Appendix F) instead matches the aligned model's distribution to a specific target and has been used for protein optimization (Ibarraran et al., 2026). In the DPO family of methods, we also evaluate ODPO-STZ, i.e. ODPO with the smooth Tchebycheff-like z-score (STZ) scalarization proposed by McCarter et al. (2026):

$$R^{\lambda}_{\mathrm{STZ}}(x,y) = -\tau\gamma\log\sum_{i=1}^{k}\exp\left(-\frac{\lambda_i}{\tau\gamma}\frac{r_i(x,y)-\mu_i}{\sigma_i}\right)$$

Finally, we evaluate our method STOMP (Equation 12), which centers each reward r_i/σ_i at $\gamma\log\hat{Z}_i(x)$ instead of its mean. STZ suffers from pathological behavior when reweighting the rewards by λ because the terms inside the logsumexp are both positive and negative (Lin et al., 2024). STOMP avoids this issue because $r_i/\sigma_i - \gamma\log\hat{Z}_i(x) \leq 0$ uniformly.

For each (dataset, base model), we train all algorithms with a range of preferences λ to probe their Pareto fronts.[2] We aggregate over λ and report the expected reward hypervolume of sequences sampled from aligned models (Auger et al., 2009; Yang et al., 2019a). We train 3 replicates of each configuration with different random seeds and dataset orders.

4.2 Protein Engineering

Datasets First, we maximize DHFR activity both in the absence and in the presence of an antibiotic, two uncorrelated objectives (Spearman $\rho = -0.010$) (Romanowicz et al., 2025). The train and test splits comprise 310 and 139 diverse homologs. Next, we maximize the negatively correlated activity and specificity of PbrR (Spearman $\rho = -0.806$) (Wang et al., 2025). We train on 1076 single-mutation variants and 76 variants with 2-5 mutations and test on 869 engineered variants with up to 12 mutations. Finally, we maximize the activity, expression, and stability of α-amylase (Spearman $\rho = 0.655, 0.602, 0.719$ for A/E, A/S, E/S) (Armer et al., 2025). We train on 8076 single-mutation variants and test on 1879 variants with 4-10 mutations. These are challenging splits that test model generalization.

Models We align protein language models (PLMs) on laboratory data measuring a variety of biological attributes to be optimized. We apply all methods to the autoregressive ProGen3-3B (PG3) (Bhatnagar et al., 2025) and two internal retrieval-augmented PLMs similar to PoET (Truong Jr and Bepler, 2023): ProGen-RA-3B (RA-3B) and ProGen-RA-10B (RA-10B) with 3B and 10B parameters respectively. See Appendix B.1 for additional details.

[2]We actually train STOMP with the preference vector $\lambda'_i = \lambda_i\bar{\lambda}_i/\sum_{i=1}^{k}\lambda_i\bar{\lambda}_i$. Recall that $\bar{\lambda}$ (Equation 10) re-weights the individual distribution-relative rewards to have the same mean. Thus, the transformation $\lambda \mapsto \lambda'$ simply keeps the preference vector interpretable for STOMP.

Dataset	Model	DPO-Lin	ODPO-Lin	MODPO	ERA	ODPO-STZ	STOMP
DHFR	PG3	**17.8 ± 0.4**	**17.7 ± 0.4**	**18.2 ± 0.1**	13.9 ± 0.1	**16.9 ± 1.3**	**18.2 ± 0.1**
DHFR	RA-3B	17.1 ± 0.3	16.4 ± 0.6	17.2 ± 0.2	16.7 ± 0.1	17.2 ± 0.1	**17.9 ± 0.1**
DHFR	RA-10B	**18.1 ± 0.1**	**17.8 ± 0.1**	**18.2 ± 0.1**	14.8 ± 0.1	17.0 ± 0.2	**18.0 ± 0.1**
PbrR	PG3	15.7 ± 0.5	15.0 ± 0.4	9.7 ± 0.7	13.8 ± 0.1	14.8 ± 0.2	**17.4 ± 0.1**
PbrR	RA-3B	13.9 ± 0.5	15.3 ± 0.3	9.4 ± 0.4	15.9 ± 0.4	12.9 ± 0.2	**17.9 ± 0.2**
PbrR	RA-10B	17.2 ± 0.1	17.1 ± 0.3	12.3 ± 0.3	11.4 ± 0.1	17.3 ± 0.1	**17.8 ± 0.1**
α-Amylase	PG3	22.9 ± 0.1	22.9 ± 0.1	23.2 ± 0.1	19.6 ± 0.1	22.8 ± 0.1	**27.8 ± 0.9**
α-Amylase	RA-3B	9.5 ± 0.3	**12.2 ± 0.8**	8.0 ± 0.2	1.4 ± 0.1	9.4 ± 0.3	**11.3 ± 0.4**
α-Amylase	RA-10B	**19.3 ± 0.1**	17.1 ± 0.1	15.4 ± 0.3	4.7 ± 0.5	18.6 ± 0.2	**19.5 ± 0.1**

Table 1: Hypervolumes (with standard errors) of offline off-policy expected rewards in the protein setting. **Bold** entries are indistinguishable from the best method with $p < 0.05$ in a 1-sided t-test. STOMP achieves or ties for the highest hypervolume in all settings.

Offline Off-Policy Evaluations In offline RL where we have access to a test set but cannot easily evaluate generations from the trained policy π, accurately estimating the expected reward $\mathbb{E}_\pi[r_i(x, y)]$ is a common challenge. We estimate expected rewards using the weighted importance sampling (WIS) estimator on the test set $\{(x_m, y_m^{(n)}) : m \in [M], n \in [N_m]\}$ (Mahmood et al., 2014; Owen, 2023; Precup et al., 2001; Sutton and Barto, 1998):

$$\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N_m}\sum_{n=1}^{N_m} w_m^{(n)} r_i(x_m, y_m^{(n)}) \quad \text{where} \quad w_m^{(n)} = \frac{\pi(y_m^{(n)}|x_m)/\pi_0(y_m^{(n)}|x_m)}{\sum_{n'=1}^{N_m}\pi(y_m^{(n')}|x_m)/\pi_0(y_m^{(n')}|x_m)}$$

While WIS is a biased estimator, it has much lower variance than vanilla importance sampling. We use WIS to estimate the expected rewards of every checkpoint from {20%, 30%, . . . , 90%, 100%} of training. We then identify the resulting Pareto fronts across all preference vectors used and report their hypervolumes (Appendix D) in Table 1. We also visualize example Pareto fronts and present a more qualitative analysis in Appendix A.

Most methods achieve near-identical performance for the DHFR dataset. This stems from a limitation of the offline evaluation: this dataset comprises diverse homologs that the models can easily distinguish, and all methods except ERA consistently assign high likelihoods to the few test sequences with high hypervolumes.

Conversely, the PbrR and α-amylase datasets are local mutational landscapes. Thus, they represent much denser samples of protein space that provide more reliable estimates of expected rewards via importance sampling. STOMP achieves the highest expected hypervolume in all 6 of these settings and is the sole winner in 4/6. Meanwhile, there are no consistent winners between the baselines. MODPO and ERA consistently underperform. MODPO's poor performance is explained by the fact that its ranking reward r_j is poorly correlated with its margin reward(s) r_{-j}, especially for PbrR (Spearman $\rho = -0.806$).

Generative Evaluations For each (dataset, base model, algorithm) combination, we generate large batches of novel sequences from the two checkpoints that maximize offline expected hypervolume. See Appendix C.1 for more details. We then train Gaussian process (GP) regressors to estimate the expected rewards of these generations. GPs have proven viable protein property predictors in challenging interpolative and extrapolative settings (Groth et al., 2024; Truong Jr and Bepler, 2025), warranting their use as oracles.

In Figure 2, we compute the expected hypervolumes (along with their uncertainties) achieved by random samples of $k \in \{12, 24, 48, 96, 192, 384\}$ generated sequences. These represent realistic sample sizes that one might test in a small wet lab experiment. See Appendix E for more details on the reward models and hypervolume estimation.

Unlike the offline evaluation, clear differences emerge when we evaluate models on the more challenging DHFR generation task (Row 1). All base models aligned with STOMP generate sequences with the best predicted rewards. For PbrR (Row 2), STOMP achieves the highest expected hypervolume for PG3 and RA-10B, and the second-highest for RA-3B where it loses by a small margin to ODPO-STZ. For α-amylase (Row 3), most methods perform quite similarly to each other. This likely arises from a limitation of the GP reward models, as their predictive accuracy is lower on this dataset than the others (Table 7).

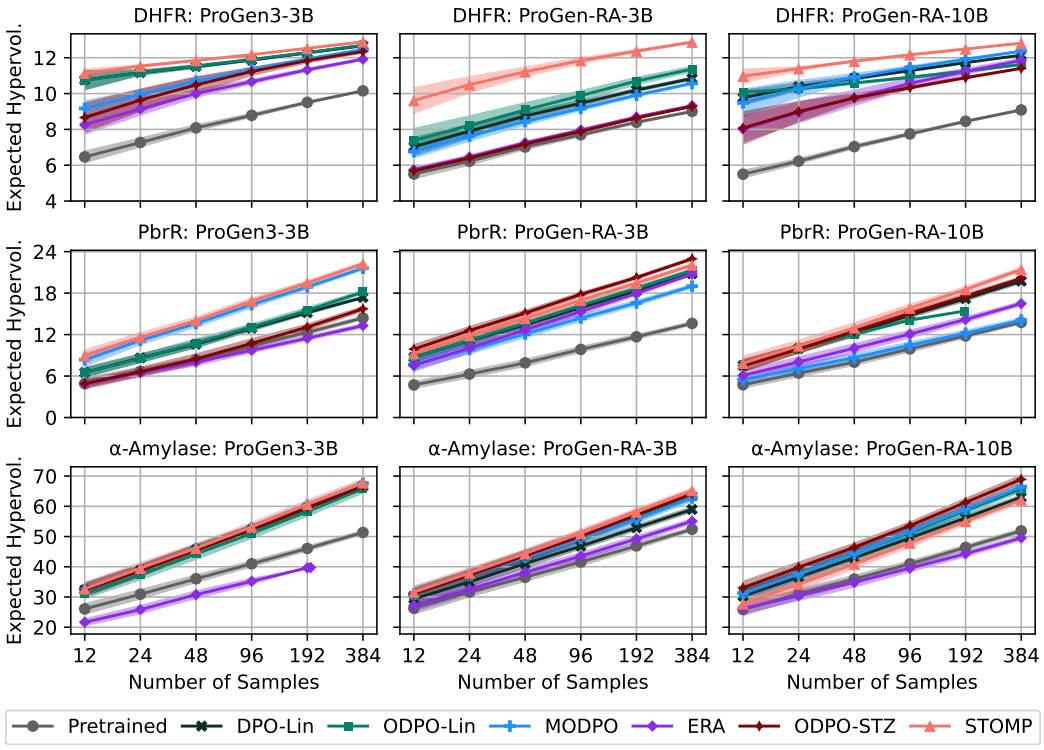

Figure 2: Hypervolumes of k randomly sampled proteins from models aligned using different preference optimization algorithms. Error bands correspond to one standard error. STOMP's generations achieve or tie for the highest expected hypervolumes in 7/9 settings.

As in the off-policy evaluations, ERA is the worst method, while DPO-Lin and ODPO-Lin are in the middle of the pack. MODPO is one of the worst performers for DHFR with PG3 and RA-3B, as well as PbrR with the RA models, but it is one of the best for DHFR with RA-10B and PbrR with PG3. This contrasts with the off-policy evaluations, where it underperformed across the board. Similarly, ODPO-STZ is the best method for PbrR with both RA models but one of the worst for DHFR with all models and PbrR with PG3.

4.3 Chatbot Alignment

Datasets First, we maximize helpfulness (`PKU-Alignment/beaver-7b-v1.0-reward`) and harmlessness (negative `PKU-Alignment/beaver-7b-v1.0-cost`) for the SafeRLHF dataset (Spearman $\rho = -0.445$) (Dai et al., 2024). Next, we maximize helpfulness (`Ray2333/gpt2-large-helpful-reward_model`) and harmlessness (`Ray2333/gpt2-large-helpful-reward_model`) for AnthropicHH (Spearman $\rho = -0.614$) (Bai et al., 2022; Yang et al., 2024). Finally, we maximize helpfulness, complexity, and negative verbosity for HelpSteer2 (Wang et al., 2024c) (Spearman $\rho = 0.289, -0.348, -0.594$ respectively for helpfulness/complexity, helpfulness/–verbosity, and complexity/–verbosity), with `RLHFlow/ArmoRM-Llama3-8B-v0.1` heads as reward models (Wang et al., 2024a;b).

Models We train `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024) and `Qwen-2.5-7B-Instruct` (Qwen et al., 2025) on chatbot alignment datasets. We apply supervised fine-tuning using a subset of each train split and generate 8 novel completions for all remaining training prompts to curate an alignment dataset. See Appendix B.2 for additional details.

Generative Evaluations We compare the same methods as in the PLM setting, but at matched KL divergence drift from the reference policy, as stated in the original Optimization Problem 3. While all methods had similar KL drift in the PLM setting, the explicit constraint was necessary here for fair comparisons because we observed significantly different KL drift between runs in this setting (Beirami et al., 2025; Rafailov et al., 2023; 2024).

Dataset	Model	DPO-Lin	ODPO-Lin	MODPO	ERA	ODPO-STZ	STOMP
SafeRLHF	LLaMA	**13.0 ± 0.1**	12.8 ± 0.3	11.3 ± 0.1	11.7 ± 0.2	12.1 ± 0.1	**13.2 ± 0.1**
SafeRLHF	Qwen	11.8 ± 0.1	**12.1 ± 0.1**	10.7 ± 0.1	6.2 ± 0.2	11.2 ± 0.2	**12.2 ± 0.1**
AnthropicHH	LLaMA	20.4 ± 0.1	20.8 ± 0.3	20.0 ± 0.1	17.3 ± 0.1	**21.3 ± 0.2**	**21.1 ± 0.2**
AnthropicHH	Qwen	**21.3 ± 0.1**	**21.1 ± 0.2**	19.6 ± 0.1	15.5 ± 0.1	20.4 ± 0.1	**21.3 ± 0.1**
HelpSteer2	LLaMA	81.2 ± 0.9	80.8 ± 0.7	71.4 ± 0.4	53.8 ± 0.5	67.6 ± 0.8	**83.5 ± 0.6**
HelpSteer2	Qwen	**83.1 ± 0.5**	80.2 ± 1.5	76.6 ± 0.8	49.5 ± 0.5	67.1 ± 0.4	78.8 ± 0.6

Table 2: Hypervolumes (with standard errors) of generations from aligned LLMs. **Bold** entries are indistinguishable from the best method with $p < 0.05$ in a 1-sided t-test. STOMP achieves or ties for the highest hypervolume in 5/6 settings.

For every training checkpoint, we generate one response $y_x \sim \pi(\cdot|x)$ per test prompt x. We then estimate $\hat{\mathbb{D}}_{\text{KL}}(\pi \mid\mid \pi_0) = \frac{1}{|\mathcal{P}|}\sum_{x\in\mathcal{P}} \log \pi(y_x|x) - \log \pi_0(y_x|x)$. We set a KL budget η as the largest $\hat{\mathbb{D}}_{\text{KL}}$ achieved by *all* training runs for a dataset and select one checkpoint per run satisfying $\eta - 2 \leq \hat{\mathbb{D}}_{\text{KL}} \leq \eta + 2$. Finally, we generate one completion per checkpoint per test prompt, aggregate over preference vectors λ to construct a Pareto front, and report hypervolumes in Table 2. See Appendices C.2, C.3, and D respectively for more details on sequence generation, KL drift matching, and hypervolume estimation.

In 5/6 evaluation settings, STOMP's generations achieve or tie for the highest expected hypervolumes. In contrast to the protein experiments where its performance was more mixed, the simplest method DPO-Lin is the strongest performer amongst our baselines. It achieves or ties for the highest hypervolume in 3/6 settings, and it is second place to STOMP on HelpSteer2 with LLaMA. As in the protein evaluations, ERA is the worst method, and MODPO consistently underperforms DPO-Lin and OPDO-Lin. ODPO-STZ once again has highly inconsistent performance: it is in the bottom half of methods in 5/6 settings, but is tied with STOMP for the best algorithm on AnthropicHH with LLaMA.

5 Discussion

In multi-objective reinforcement learning (MORL), a fundamental limitation of linear reward scalarization is its inability to recover the full Pareto front of optimal solutions. Smooth Tchebycheff scalarization (STS) is a recent technique that overcomes this limitation. Rather than scalarizing the rewards, we frame MORL itself as the optimization problem to be scalarized via STS and use this insight to propose Smooth Tchebycheff Optimization of Multi-Objective Preferences (STOMP), a novel offline MORL algorithm.

Our experiments span multiple language models and diverse datasets in both the protein engineering and chatbot alignment domains. They demonstrate that STOMP outperforms existing baselines, achieving or tying for the highest hypervolumes in 16/18 protein settings and 5/6 natural language settings. Other algorithms rank inconsistently relative to each other in the protein experiments, while vanilla DPO with linear scalarization is a strong baseline for the natural language experiments. ODPO-STZ relies on a more naive application of STS than STOMP and shows particularly inconsistent performance. This highlights the importance of our principled derivation in Section 3.1 in elucidating the appropriate techniques for applying STS to offline MORL.

More broadly, the smooth Tchebycheff reward scalarizations derived from Optimization Problems 6–7 are completely general to any KL-constrained or MaxEnt RL formulation. A natural extension would be applying these scalarizations to multi-objective *online* RL. A key challenge to overcome in this setting would be estimating the partition functions $Z_i(x)$ when sequences y are not collated in an offline dataset but are instead generated online.

Finally, we leveraged the framing of MaxEntRL to scalarize the rewards $R_{\text{ST}}^{\lambda,\pi}$ and R_{ST}^{λ} (Equations 8–9) because π_0 introduced high variance when estimating partition functions. However, STOMP implicitly constrains the KL divergence between the π and π_0. This is because our early experiments found that MaxEnt variants of STOMP (i.e. removing the $\log \pi_0(y|x)$ terms from Equation 11) suffered from training instability and sensitivity to hyperparameters. We leave further study of such reference-free losses to future work.

Ethics Statmeent

Computational protein design carries the dual potential to accelerate the development of novel therapeutics and other society-improving molecules, while providing parallel capabilities for nefarious uses, such as engineering of bioweapons. When bolstered by current and future iterations of generative AI, these capabilities are heightened and expected to grow further. The global protein design community has begun to establish appropriate regulations and guidelines towards the continued beneficial development and application of these technologies. We support having a set of community values, guiding principles, and commitments for the responsible development of AI for protein design (https://responsiblebiodesign.ai). Gene synthesis represents a critical step in the actualization of designed protein sequences. The International Gene Synthesis Consortium (IGSC) unites major gene synthesis providers under a commitment to screen all incoming orders against known pathogens and potentially dangerous sequences. As a concrete step towards safe application of protein design technology, all gene synthesis work in support of the present study was performed with IGSC members. For all protein design projects, we urge researchers to maintain ethical oversight throughout project initiation, experimental characterization, and subsequent deployment phases to ensure safety and avoid unintended harmful outcomes. For the current algorithms described in this paper, we find the benefit of research accessibility to greatly outweigh any theoretical risks.

AI Use Statement

In this work, we used generative AI tools for the chatbot alignment experiments for scaffolding the codebase, as well as drafting data loading and metric aggregation for efficient experimental iteration. We did not use generative AI tools for the writing of this manuscript, the derivation of STOMP, or the protein design experiments. We have reviewed all AI-assisted work. This includes the authors reviewing LLM-generated code and carefully assessing the veracity of results produced with that code. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

A. Amini, T. Vieira, and R. Cotterell. Direct Preference Optimization with an Offset. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9954–9972, Bangkok, Thailand, 8 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.592/.

C. Armer, H. Kane, D. L. Cortade, H. Redestig, D. A. Estell, A. Yusuf, N. Rollins, A. Spinner, D. Marks, T. J. Brunette, P. J. Kelly, and E. DeBenedictis. Results of the Protein Engineering Tournament: An Open Science Benchmark for Protein Modeling and Design. *Proteins: Structure, Function, and Bioinformatics*, 93(11):2005–2014, 2025. doi: https://doi.org/10.1002/prot.70008. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.70008.

A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, FOGA '09, page 87102, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584140. doi: 10.1145/1527125.1527138. URL https://doi.org/10.1145/1527125.1527138.

Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, and J. Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, 2022. URL https://arxiv.org/abs/2204.05862.

M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21524–21538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf.

A. Beirami, A. Agarwal, J. Berant, A. D'Amour, J. Eisenstein, C. Nagpal, and A. T. Suresh. Theoretical guarantees on the best-of-n alignment policy. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 3580–3602. PMLR, 2025.

N. Beume, C. M. Fonseca, M. Lopez-Ibanez, L. Paquete, and J. Vahrenhold. On the Complexity of Computing the Hypervolume Indicator. *IEEE Transactions on Evolutionary Computation*, 13 (5):1075–1082, Oct 2009. ISSN 1941-0026. doi: 10.1109/TEVC.2009.2015575.

A. Bhatnagar, S. Jain, J. Beazer, S. C. Curran, A. M. Hoffnagle, K. S. Ching, M. Martyn, S. Nayfach, J. A. Ruffolo, and A. Madani. Scaling Unlocks Broader Generation and Deeper Functional Understanding of Proteins. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=yvGL2HP7pU.

V. J. Bowman. On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multiple-Criteria Objectives. In H. Thiriez and S. Zionts, editors, *Multiple Criteria Decision Making*, pages 76–86, Berlin, Heidelberg, 1976. Springer Berlin Heidelberg. ISBN 978-3-642-87563-2.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

R. A. Bradley and M. E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2334029.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

T. Chen, B. Xu, C. Zhang, and C. Guestrin. Training Deep Nets with Sublinear Memory Cost, 2016. URL https://arxiv.org/abs/1604.06174.

S. Chennakesavalu, F. Hu, S. Ibarraran, and G. M. Rotskoff. Aligning Transformers with Continuous Feedback via Energy Rank Alignment. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=SXxlb1miXS.

E. U. Choo and D. R. Atkins. Proper Efficiency in Nonconvex Multicriteria Programming. *Mathematics of Operations Research*, 8(3):467–470, 1983. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3689313.

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep Reinforcement Learning from Human Preferences. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

J. Dai, X. Pan, R. Sun, J. Ji, X. Xu, M. Liu, Y. Wang, and Y. Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TyFrPOKYXw.

I. Das and J. Dennis. A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems. *Structural Optimization*, 14: 63–69, 01 1997. doi: 10.1007/BF01197559.

I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998. doi: 10.1137/S1052623496307510. URL https://doi.org/10.1137/S1052623496307510.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 6 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

M. Ehrgott. *Multicriteria optimization*. Springer, Berlin, Germany, 2nd ed edition, 2005. ISBN 3540213988. URL http://rave.ohiolink.edu/ebooks/ebc/3540276599.

B. Eysenbach and S. Levine. Maximum Entropy RL (Provably) Solves Some Robust RL Problems. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PtSAD3caaA2.

L. Faury, U. Tanielian, E. Dohmatob, E. Smirnova, and F. Vasile. Distributionally Robust Counterfactual Risk Minimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3850–3857, 4 2020. doi: 10.1609/aaai.v34i04.5797. URL https://ojs.aaai.org/index.php/AAAI/article/view/5797.

A. Fisch, J. Eisenstein, V. Zayats, A. Agarwal, A. Beirami, C. Nagpal, P. Shaw, and J. Berant. Robust Preference Optimization through Reward Model Distillation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=E2zKNuwNDc.

C. Fonseca, L. Paquete, and M. Lopez-Ibanez. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1157–1163, 2006. doi: 10.1109/CEC.2006.1688440.

Y. Fu, T. Chen, J. Chai, X. Wang, S. Tu, G. Yin, W. Lin, Q. Zhang, Y. Zhu, and D. Zhao. SRFT: A Single-Stage Method with Supervised and Reinforcement Fine-Tuning for Reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=n6EOr6kQWQ.

Z. Gao, J. D. Chang, W. Zhan, O. Oertell, G. Swamy, K. Brantley, T. Joachims, J. A. Bagnell, J. D. Lee, and W. Sun. REBEL: reinforcement learning via regressing relative rewards. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

A. M. Geoffrion. Solving Bicriterion Mathematical Programs. *Operations Research*, 15(1):3954, 2 1967. ISSN 0030-364X. doi: 10.1287/opre.15.1.39.

J. L. Goffin. On convergence rates of subgradient optimization methods. *Mathematical Programming*, 13(1):329–347, Dec 1977. ISSN 1436-4646. doi: 10.1007/BF01584346.

W. Grathwohl, K. Swersky, M. Hashemi, D. Duvenaud, and C. Maddison. Oops I Took A Gradient: Scalable Sampling for Discrete Distributions. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3831–3841. PMLR, 7 2021. URL https://proceedings.mlr.press/v139/grathwohl21a.html.

A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura,

P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

P. M. Groth, M. H. Kerrn, L. Olsen, J. Salomon, and W. Boomsma. Kermut: Composite kernel regression for protein variant effects. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 29514–29565. Curran Associates, Inc., 2024. doi: 10.52202/079017-0929. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/34547650b2ca69d91f3b3c3ae8b21962-Paper-Conference.pdf`.

A. P. Guerreiro and C. M. Fonseca. Computing and Updating Hypervolume Contributions in Up to Four Dimensions. *IEEE Transactions on Evolutionary Computation*, 22(3):449–463, 2018. doi: 10.1109/TEVC.2017.2729550.

C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36 (1), 4 2022. ISSN 1387-2532. doi: 10.1007/s10458-022-09552-y.

J. Hejna, R. Rafailov, H. Sikchi, C. Finn, S. Niekum, W. B. Knox, and D. Sadigh. Contrastive preference learning: Learning from human feedback without reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iX1RjVQODj.

A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

C. Hvarfner, E. O. Hellsten, and L. Nardi. Vanilla Bayesian Optimization Performs Great in High Dimensions. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=OfT8MgIqHT.

S. Ibarraran, S. Chennakesavalu, F. Hu, and G. M. Rotskoff. Efficient, Few-shot Directed Evolution with Energy Rank Alignment. In *ICLR 2026 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2026. URL https://openreview.net/forum?id=2Y6A9oyovP.

S. Jain, J. Beazer, J. A. Ruffolo, A. Bhatnagar, and A. Madani. E1: Retrieval-Augmented Protein Encoder Models. *bioRxiv*, 2025. doi: 10.1101/2025.11.12.688125. URL https://www.biorxiv.org/content/early/2025/11/13/2025.11.12.688125.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

S. H. Lee, Y. Li, J. Ke, I. Yoo, H. Zhang, J. Yu, Q. Wang, F. Deng, G. Entis, J. He, G. Li, S. Kim, I. Essa, and F. Yang. Parrot: Pareto-optimal multi-reward reinforcement learning framework for text-to-image generation. In *Computer Vision ECCV 2024: 18th European Conference, Milan, Italy, September 29October 4, 2024, Proceedings, Part XXXVIII*, page 462478, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72919-5. doi: 10.1007/978-3-031-72920-1_26. URL https://doi.org/10.1007/978-3-031-72920-1_26.

K. Li, T. Zhang, and R. Wang. Deep Reinforcement Learning for Multiobjective Optimization. *IEEE Transactions on Cybernetics*, 51(6):3103–3114, 2021. doi: 10.1109/TCYB.2020.2977661.

Y. Liang, J. He, G. Li, P. Li, A. Klimovskiy, N. Carolan, J. Sun, J. Pont-Tuset, S. Young, F. Yang, J. Ke, K. D. Dvijotham, K. M. Collins, Y. Luo, Y. Li, K. J. Kohlhoff, D. Ramachandran, and V. Navalpakkam. Rich Human Feedback for Text-to-Image Generation . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19401–19411, Los Alamitos, CA, USA, 6 2024. IEEE Computer Society. doi: 10.1109/CVPR52733.2024.01835. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01835.

X. Lin, X. Zhang, Z. Yang, F. Liu, Z. Wang, and Q. Zhang. Smooth Tchebycheff scalarization for multi-objective optimization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Z. Liu, M. Lu, S. Zhang, B. Liu, H. Guo, Y. Yang, J. Blanchet, and Z. Wang. Provably Mitigating Overoptimization in RLHF: Your SFT Loss is Implicitly an Adversarial Regularizer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=2cQ3lPhkeO.

I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

M. López-Ibáñez. *moocore: Core Mathematical Functions for Multi-Objective Optimization*, 2026. URL https://multi-objective.github.io/moocore/r/. R package version 0.2.0.900.

A. Madani, B. Krause, E. R. Greene, S. Subramanian, B. P. Mohr, J. M. Holton, J. L. Olmos, C. Xiong, Z. Z. Sun, R. Socher, J. S. Fraser, and N. Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, Aug 2023. ISSN 1546-1696. doi: 10.1038/s41587-022-01618-2. URL https://doi.org/10.1038/s41587-022-01618-2.

A. R. Mahmood, H. van Hasselt, and R. S. Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/1c64ee92596e8ea5050fc435a1d57459-Paper.pdf.

X. Mao, F.-L. Li, H. Xu, W. Zhang, W. Chen, and A. T. Luu. Don't Forget Your Reward Values: Language Model Alignment via Value-based Calibration. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17622–17642, Miami, Florida, USA, 11 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.emnlp-main.976/.

C. McCarter, N. Bhattacharya, S. W. Ober, and H. Elliott. How to make the most of your masked language model for protein engineering, 2026. URL https://arxiv.org/abs/2603.10302.

J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29287–29303. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f51338d736f95dd42427296047067694-Paper.pdf.

Y. Meng, M. Xia, and D. Chen. SimPO: Simple Preference Optimization with a Reference-Free Reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=3Tzcot1LKb.

P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. Mixed Precision Training. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1gs9JgRZ.

K. Miettinen. *Nonlinear multiobjective optimization*. Kluwer, Boston, MA, USA, 1999.

M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, and M. Steinegger. Colabfold: making protein folding accessible to all. *Nature Methods*, 19(6):679–682, Jun 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01488-1. URL https://doi.org/10.1038/s41592-022-01488-1.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

A. B. Owen. *Practical Quasi-Monte Carlo Integration*. https://artowen.su.domains/mc/practicalqmc.pdf, 2023.

R. Y. Pang, W. Yuan, H. He, K. Cho, S. Sukhbaatar, and J. Weston. Iterative Reasoning Preference Optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 116617–116637. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d37c9ad425fe5b65304d500c6edcba00-Paper-Conference.pdf.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

D. Precup, R. S. Sutton, and S. Dasgupta. Off-Policy Temporal Difference Learning with Function Approximation. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 417424, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

S. Qiu, D. Zhang, R. Yang, B. Lyu, and T. Zhang. Traversing pareto optimal policies: Provably efficient multi-objective reinforcement learning, 2024. URL https://arxiv.org/abs/2407.17466.

Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 7 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

R. Rafailov, Y. Chittepu, R. Park, H. Sikchi, J. Hejna, B. Knox, C. Finn, and S. Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, volume 37, pages 126207–126242. Curran Associates, Inc., 2024.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *J. Artif. Int. Res.*, 48(1):67113, 10 2013. ISSN 1076-9757.

K. J. Romanowicz, C. Resnick, S. R. Hinton, and C. Plesa. Exploring antibiotic resistance in diverse homologs of the dihydrofolate reductase protein family through broad mutational scanning. *Science Advances*, 11(33):eadw9178, 2025. doi: 10.1126/sciadv.adw9178. URL https://www.science.org/doi/abs/10.1126/sciadv.adw9178.

J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust Region Policy Optimization. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 7 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. doi: 10.1109/JPROC.2015.2494218.

M. Steinegger and J. Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, Nov 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988. URL https://doi.org/10.1038/nbt.3988.

R. E. Steuer and E.-U. Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, Oct 1983. ISSN 1436-4646. doi: 10.1007/BF02591870.

R. S. Sutton and A. G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL http://www.incompleteideas.net/book/first/the-book.html.

M. Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 10491056, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553508.

T. Truong Jr and T. Bepler. PoET: A generative model of protein families as sequences-of-sequences. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 77379–77415. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f4366126eba252699b280e8f93c0ab2f-Paper-Conference.pdf.

T. Truong Jr and T. Bepler. Understanding protein function with a multimodal retrieval-augmented foundation model. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 53167–53222. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/4d19160864e6a644496d61b21c7e015a-Paper-Conference.pdf.

K. Van Moffaert, M. M. Drugan, and A. Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 191–199, 2013. doi: 10.1109/ADPRL.2013.6615007.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

B. M. Wang, N. Chiang, H. M. Ekas, D. M. Brown, G. Dildine, T. J. Lucci, S. Feng, V. Bly, J.-F. Gaillard, J. B. Lucks, A. S. Karim, D. Shukla, and M. C. Jewett. Active learning-guided optimization of cell-free biosensors for lead testing in drinking water. *Nature Communications*, 17(1):261, Dec 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-66964-6. URL https://doi.org/10.1038/s41467-025-66964-6.

H. Wang, Y. Lin, W. Xiong, R. Yang, S. Diao, S. Qiu, H. Zhao, and T. Zhang. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8642–8655, Bangkok, Thailand, Aug. 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.468. URL https://aclanthology.org/2024.acl-long.468/.

H. Wang, W. Xiong, T. Xie, H. Zhao, and T. Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10582–10592, Miami, Florida, USA, Nov. 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.620. URL https://aclanthology.org/2024.findings-emnlp.620/.

Z. Wang, Y. Dong, O. Delalleau, J. Zeng, G. Shen, D. Egert, J. J. Zhang, M. N. Sreedhar, and O. Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024c. URL https://openreview.net/forum?id=PvVKUFhaNy.

Y. Wen, C. Xu, J. Y.-C. Hu, K. Ding, and H. Liu. Pareto-Optimal Energy Alignment for Designing Nature-Like Antibodies. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6wbykApw7A.

T. Widatalla, R. Rafailov, and B. Hie. Aligning protein generative models with experimental fitness via Direct Preference Optimization. *bioRxiv*, 2024. doi: 10.1101/2024.05.20.595026. URL https://www.biorxiv.org/content/early/2024/05/21/2024.05.20.595026.

M. Williams. Multi-objective Reinforcement learning from AI Feedback, 2024. URL https://arxiv.org/abs/2406.07295.

J. Wu, X. Wang, Z. Yang, J. Wu, J. Gao, B. Ding, X. Wang, and X. He. AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ETLKYVMVLt.

K. Yang, M. Emmerich, A. Deutz, and T. Bäck. Multi-Objective Bayesian Global Optimization using expected hypervolume improvement gradient. *Swarm and Evolutionary Computation*, 44: 945–956, 2019a. ISSN 2210-6502. doi: https://doi.org/10.1016/j.swevo.2018.10.007. URL https://www.sciencedirect.com/science/article/pii/S2210650217307861.

R. Yang, X. Sun, and K. Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019b. Curran Associates Inc.

R. Yang, X. Pan, F. Luo, S. Qiu, H. Zhong, D. Yu, and J. Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *International Conference on Machine Learning*, 2024.

Y. Zhao, A. Gu, R. Varma, L. Luo, C.-C. Huang, M. Xu, L. Wright, H. Shojanazeri, M. Ott, S. Shleifer, A. Desmaison, C. Balioglu, P. Damania, B. Nguyen, G. Chauhan, Y. Hao, A. Mathews, and S. Li. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. *Proc. VLDB Endow.*, 16(12):38483860, 8 2023. ISSN 2150-8097. doi: 10.14778/3611540.3611569. URL https://doi.org/10.14778/3611540.3611569.

Z. Zhou, J. Liu, J. Shao, X. Yue, C. Yang, W. Ouyang, and Y. Qiao. Beyond One-Preference-Fits-All Alignment: Multi-Objective Direct Preference Optimization. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10586–10613, Bangkok, Thailand, 8 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.630/.

B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, page 14331438. AAAI Press, 2008. ISBN 9781577353683.

D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-Tuning Language Models from Human Preferences, 2020. URL https://arxiv.org/abs/1909.08593.

A Visualizing Pareto Fronts for Off-Policy Evaluations

In Figures 3–5, we illustrate the Pareto fronts achieved by different methods in the offline off-policy evaluations described in Section 4.2. We visualize the training set in grey, the test set in lilac, and the expected value of a random point from the test set as a green star. Recall that we estimate $\mathbb{E}_\pi[r_i(x, y)]$ on the test set using the weighted importance sampling estimator (Mahmood et al., 2014; Owen, 2023; Precup et al., 2001; Sutton and Barto, 1998):

$$\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N_m}\sum_{n=1}^{N_m} w_m^{(n)} r_i(x_m, y_m^{(n)}) \quad \text{where} \quad w_m^{(n)} = \frac{\pi(y_m^{(n)}|x_m)/\pi_0(y_m^{(n)}|x_m)}{\sum_{n'=1}^{N_m}\pi(y_m^{(n')}|x_m)/\pi_0(y_m^{(n')}|x_m)}$$

Using this estimator, the expected reward under π_0 is simply the expected reward of a random test point (the green star). Meanwhile, the expected reward under a preference optimized policy indicates that policy's most preferred sequences from the test set.

For DHFR (Figure 3), almost all methods perform near-optimally. This is largely a consequence of the fact that the dataset comprises highly diverse homologs that all models can easily distinguish (Section 4.2). For PbrR (Figure 4), where the two objectives are highly negatively correlated in the training set ($\rho = -0.806$), STOMP is the only method that consistently assigns high likelihood to the cluster of sequences exclusive to the test set that have both a high increase in on-target binding and a high decrease in off-target binding. Finally, for α-amylase (Figure 5), where the three objectives are positively correlated, STOMP slightly expands the Pareto front achieved by the other methods for PG3 and RA-10B, while achieving comparable performance to ODPO-Lin for RA-3B.

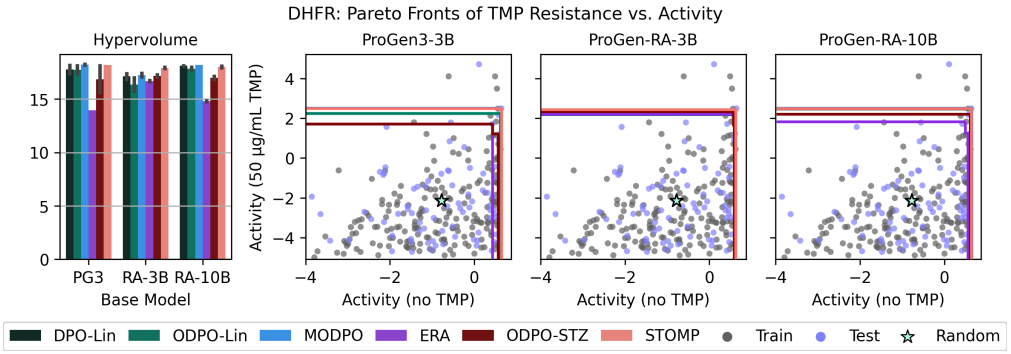

Figure 3: Pareto fronts of DHFR activity in the absence of TMP and in the presence of 50µg/mL of TMP. All fitness values are in units of $\log_2$-fold change.

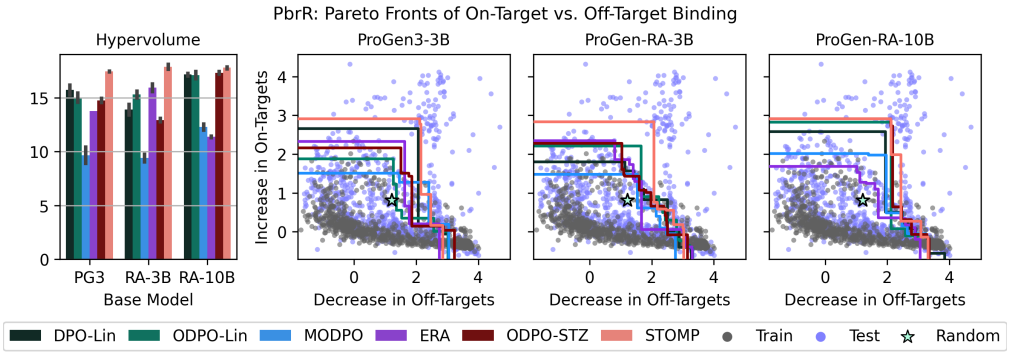

Figure 4: Pareto fronts of increase in on-target binding and decrease in off-target binding for PbrR. All fitness values are in units of $\log_2$-fold change.

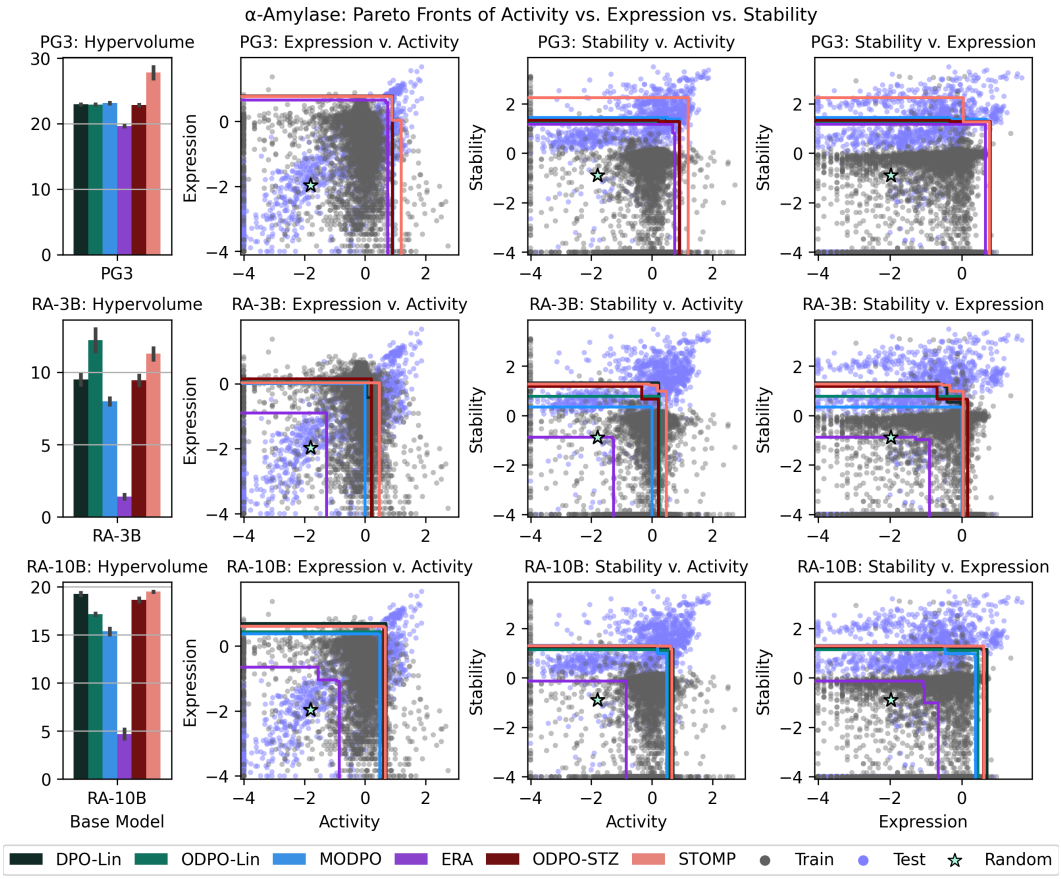

Figure 5: Pareto fronts of $\log_2$-fold change in α-amylase activity, stability, and expression.

B Training Hyperparameters

B.1 Protein Engineering Experiments

Similar to Bhatnagar et al. (2025), we use the `<bos>` token as the context x for PG3. For the RA models, we follow PoET's prompt construction strategy (Truong Jr and Bepler, 2023) to construct 24 prompts $x_1, \ldots, x_{24}$ with up to 4096 tokens of homologs discovered via ColabFold search (Mirdita et al., 2022; Steinegger and Söding, 2017). For the diverse DHFR dataset, we use homologs >30% ID to any training sequence, while for the local mutational landscapes of PbrR and α-amylase, we use homologs >70% ID to the wild type.

We train all models described in Section 4.2 for 782 batches with a batch size of 64. We fine-tune all model weights as described by Bhatnagar et al. (2025, Appendix C). We used the AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and BF16 mixed precision (Micikevicius et al., 2018). We linearly increased the learning rate to 10^{-5} over an initial warmup period of 79 batches, after which we decay the learning rate to 5×10^{-6} following a cosine schedule. All models were trained using PyTorch FSDP2 (Paszke et al., 2019; Zhao et al., 2023) with gradient checkpointing (Chen et al., 2016) on 8xH200 for up to 3 hours.

We train each model on each dataset with 3 preference vectors λ. For the PbrR and DHFR datasets, we sweep over $\lambda \in \{(\frac{1}{3}, \frac{2}{3}), (\frac{1}{2}, \frac{1}{2}), (\frac{2}{3}, \frac{1}{3})\}$. For α-amylase, we use $\lambda \in \{(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (\frac{1}{4}, \frac{1}{4}, \frac{1}{2}), (\frac{2}{5}, \frac{1}{5}, \frac{2}{5})\}$ ordered as (activity, expression, stability). We train 3 replicates of each configuration with different random seeds and dataset orders. In total, we train 486 distinct models.

Dataset	Base Model	α	β	γ	δ
DHFR	ProGen3-3B	0.05	0.2	0.2	1
DHFR	ProGen-RA3B	0.01	0.2	0.2	1
DHFR	ProGen-RA10B	0.05	0.2	0.2	1
PbrR	ProGen3-3B	0.02	0.1	0.2	1
PbrR	ProGen-RA3B	0.02	0.1	0.2	1
PbrR	ProGen-RA10B	0.01	0.1	0.2	1
α-Amylase	ProGen3-3B	0.05	0.05	0.2	0.5
α-Amylase	ProGen-RA3B	0.02	0.05	0.2	0.5
α-Amylase	ProGen-RA10B	0.02	0.05	0.2	0.5

Table 3: Hyperparameters of preference optimization algorithms.

Table 3 details the hyperparameters of the preference optimization algorithms for each experimental setting. For each of the datasets and models, we use the same KL regularization strength β and reward difference threshold δ across all algorithms after tuning the values for DPO-Lin. Similar to Bhatnagar et al. (2025), we tune the NLL regularization coefficient α (Equation 12) individually for each method to the lowest value ensuring that validation perplexity degrades by no more than 1 point over the course of training. In general, this results in consistent values of α (as reported in Table 3) for all preference optimization algorithms for each (dataset, base model) pair, with the following exceptions:

- For PbrR, we set $\alpha = 0.05$ for ODPO-STZ and $\alpha = 0.01$ for STOMP. These settings are consistent across all models.
- For DHFR, we set $\alpha = 0.02$ for STOMP when using the ProGen3-3B and ProGen-RA-10B models.
- For α-amylase, we set $\alpha = 0.02$ for DPO-Lin when using the ProGen3-3B model.

We also improve the training stability of all methods by clamping the reward difference to 1, i.e. we take $\min(1, R(x, y_w) - R(x, y_l) + \delta)$. We fix STOMP's and ODPO-STZ's scalarization temperatures to $\tau = 1$ and $\gamma = 0.2$ across the board because we find them to be good default values that work well in a wide range of settings (Appendix B.3).

B.2 Chatbot Alignment Experiments

For all three benchmark datasets, the pool of test prompts are taken from the published test set of each dataset. The training set for each benchmark is then split into three partitions: the SFT split, the alignment training split, and the alignment validation split. For each of the benchmarks, eight completions are sampled per-prompt in the alignment split. These are used for training rather than the existing responses from the dataset because the partition function estimation requires multiple samples per-prompt drawn from the same policy to construct an appropriate estimate, along with the population statistics required for reward standardization.

We train all models in Section 4.3 for 400 optimizer steps using an effective batch size of 16. All models are trained as LoRA adapters (Hu et al., 2022) with rank $r = 16$, $\alpha = 32$, dropout 0.05, on all attention and feed-forward projections (`{q,k,v,o,gate,up,down}_proj`). We use the AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and FP32 precision for the LoRA adapters. We linearly increase the learning rate to 10^{-5} over an initial warmup period of 40 optimizer steps, after which we decay the learning rate to 5×10^{-6} following a cosine schedule. All models are trained on a single H200 until completion of the 400 optimizer steps.

We set alignment hyperparameters $\beta = 0.1$ and $\alpha = 0$ (no NLL regularization), with scalarization temperatures $\tau = 1.0$ and $\gamma = 0.2$ as in the protein experiments. We found that NLL regularization was not necessary in this setting because the training sets were sampled from the reference policy π_0 (Fu et al., 2026; Liu et al., 2024). For each (dataset, base model) combination, we set δ such that 30% of comparison pairs were retained across all objectives compared. The reference policy π_0 is a supervised fine-tune of the base model

on the dataset's SFT split, one per dataset, and it is shared by every method, model, and seed for that dataset. It is the initialization of every alignment run, the denominator of every KL estimate (Appendix C.3), and the source of the per-response thresholds used by the auxiliary satisfaction metric.

For the two-objective datasets SafeRLHF and AnthropicHH, we train each model with 9 preference vectors $\lambda \in \left\{\left(\frac{m}{10}, \frac{10-m}{10}\right) : m \in [9]\right\}$. For the three-objective SafeRLHF dataset, we train each model on the 15 interior points of the Das-Dennis lattice (Das and Dennis, 1998) with $H = 7$, i.e. $\lambda \in \left\{\left(\frac{m}{7}, \frac{n}{7}, \frac{7-m-n}{7}\right) : m \in [5], n \in [5], m+n \leq 6\right\}$. We also train three replicates of each model with different random seeds, for a total of 594 models (162 each for SafeRLHF and AnthropicHH, 270 for HelpSteer2).

B.3 Sensitivity to Scalarization Hyperparameters τ and γ

To determine the sensitivity of STOMP to its scalarization hyperparameters τ and γ, we replicate the offline off-policy evaluations of Section 4.2. We train ProGen3-3B using STOMP while sweeping τ (Table 4) and γ (Table 5) using all 3 protein datasets, and we report the hypervolumes of the expected rewards below. We find that performance is stable across a large range of values for both τ and γ. However, we do observe some performance degradations on the α-amylase dataset when $\gamma \geq 0.5$.

Dataset	$\tau = 0.2$	$\tau = 0.5$	$\tau = 1.0$	$\tau = 2.0$	$\tau = 5.0$
DHFR	18.18	18.31	18.16	18.27	18.14
PbrR	17.73	17.96	18.21	17.26	17.42
α-Amylase	27.17	27.86	30.30	26.91	26.52

Table 4: Hypervolumes sweeping τ while fixing $\gamma = 0.2$.

Dataset	$\gamma = 0.05$	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.5$	$\gamma = 1.0$
DHFR	18.18	18.17	18.16	18.20	18.07
PbrR	18.04	17.72	18.21	17.97	17.35
α-Amylase	26.63	27.19	30.30	22.61	20.33

Table 5: Hypervolumes sweeping γ while fixing $\tau = 1.0$.

C Generation Hyperparameters

C.1 Generated Proteins

For the DHFR dataset, we select the four DHFR homologs from the training set that maximize the hypervolume. We prompt PG3 with the first 20 residues (out of 161–170 total) on either the N- or C-terminal. For the RA models, we also prepend the prompts used for training. For all models, we autoregressively generate 100,000 proteins via top-p sampling with $p = 0.95$ and temperatures $\tau \in [0.5, 1.0]$ (Holtzman et al., 2020), and we apply the quality filtering criteria of Bhatnagar et al. (2025).

On the other hand, since the PbrR and α-amylase datasets are local mutational landscapes, any reward model trained on them would be less reliable at evaluating diverse generations from an autoregressive PLM. We instead use the Gibbs with Gradient (GWG) algorithm to propose mutations to the wild type protein using PLM negative log likelihoods as an energy function (Grathwohl et al., 2021). Because GWG is a Metropolis-Hastings algorithm, we can use the sequences sampled from it to estimate the expected rewards of sampling from the PLM of interest while remaining in-distribution for our reward models (Grathwohl et al., 2021; Owen, 2023). For each model, we run 1500 GWG trajectories for 300 steps each and filter out any sequences with >10 mutations, since these are likely out-of-distribution for our reward models. For the RA models, these trajectories are evenly split across the different training prompts.

Dataset	Model	η (nats)	Worst dev.	Common range	Width
SafeRLHF	Llama-3.1-8B	26.298	0.177	[4.71, 27.84]	23.13
SafeRLHF	Qwen2.5-7B	20.199	1.138	[2.66, 25.89]	23.23
HH	Llama-3.1-8B	16.042	0.405	[2.96, 19.42]	16.45
HH	Qwen2.5-7B	12.890	1.977	[1.89, 13.55]	11.66
HelpSteer2	Llama-3.1-8B	20.616	0.471	[2.12, 21.43]	19.31
HelpSteer2	Qwen2.5-7B	21.645	0.235	[1.89, 21.53]	19.64

Table 6: Matched-KL budgets.

C.2 LLM Generations

We autoregressively sample one completion per prompt per checkpoint using full ancestral sampling (temperature $\tau = 1.0$, top-p with $p = 1.0$). This was deliberately chosen to properly estimate the KL-divergence from the reference policy to ensure a matched-drift comparison. We sample completions for only 150 prompts for the test sets of all three datasets to account for differing dataset sizes, since AnthropicHH is notably larger than SafeRLHF or HelpSteer2.

C.3 KL Divergence Matching for Chatbot Alignment Experiments

As discussed in Section 4.3, we generate one response $y_x \sim \pi(\cdot|x)$ per test prompt for every training checkpoint. We then estimate $\hat{\mathbb{D}}_{\mathrm{KL}}(\pi \,||\, \pi_0) = \frac{1}{|\mathcal{P}|} \sum_{x \in \mathcal{P}} \log \pi(y_x|x) - \log \pi_0(y_x|x)$. When determining the KL budget η, we specifically omit checkpoints where the trained policy has drifted less than 1.0 nats away from the reference policy. We also omit from consideration any η that is lower than the one-half the median of the maximum KL achieved by each run to set an informative floor. When there is no η above the floor, we instead take η as the midpoint of that floor and the lowest maximum-achieved KL in any run. Table 6 describes η for each dataset and model, along with what the worst deviation from that budget is for any of the selected checkpoints. The common range of available KL values that can be matched is also described, along with the width of that range.

D Multi-Seed Hypervolume Estimation

A Pareto front requires one trained policy per preference weight. With $|\Lambda|$ weights and S seeds there are $S^{|\Lambda|}$ ways to assemble one, and seed labels carry no meaning *across* weights, so pairing by index is one arbitrary choice among many and injects variance not present in the data. We therefore report

$$\widehat{\mathrm{HV}} \;=\; \frac{1}{S^{|\Lambda|}} \sum_{c \in \{1,\dots,S\}^{|\Lambda|}} \mathrm{HV}\Big(\big\{\, r_{\lambda_j,\, c_j} \,\big\}_{j=1}^{|\Lambda|} \Big), \tag{13}$$

the mean over all assignments c of seeds to weights, where $r_{\lambda,s}$ is the mean reward vector of the policy trained at weight λ with seed s.

Standard errors come from a bootstrap over the unit of replication: for each preference weight, we independently resample its S seeds with replacement, recompute Equation 13, and repeat $S^{|\Lambda|}$ times. We take this approach because enumerating all $S^{|\Lambda|}$ assignments artificially decreases the reported standard error for the LLM experiments, which have higher variance because they rely on just one completion per prompt per checkpoint.

Two methods are compared by bootstrapping each independently and reporting the one-sided proportion of replicates in which the difference does not favor the nominal leader. Tables mark every method not separated from the leader at the 5% level. Such a mark says a method has *not been shown to be worse*.

In all settings, we compute hypervolumes in z-score units. We set the reference point as the minimum observed reward independently over rewards across the train and test splits.

E Gaussian Process Reward Models

We wish to gauge the generative capabilities of the protein language models when trained with different optimization objectives across different base models and datasets as described in Section 4.2. For this, we turn to the Bayesian optimization literature, where probabilistic surrogate models are commonly used to estimate the utility of observing new data points (Shahriari et al., 2016).

By conditioning a Gaussian Process (GP) (Rasmussen and Williams, 2005) surrogate model on the experimental data, we obtain a posterior predictive distribution over the sampled sets of generated sequences. We can then compute the expected hypervolumes and use these as indicators of the relative generative capabilities of the aligned models.

We model the experimental outcomes as $y = f(\mathbf{x}) + \epsilon$, where f is some unknown function over protein sequences and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is Gaussian distributed noise. We place a GP prior over the function f, such that

$$f(\mathbf{x}) \sim \mathcal{GP}(m, k(\mathbf{x}, \mathbf{x}')).$$

Our GP prior is characterized by a constant mean m and a Matérn 5/2 kernel (Rasmussen and Williams, 2005):

$$k_{\text{Matérn}\frac{5}{2}}(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\sqrt{5}\|\mathbf{x} - \mathbf{x}'\|_2}{\ell} + \frac{5\|\mathbf{x} - \mathbf{x}'\|_2^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}\|\mathbf{x} - \mathbf{x}'\|_2}{\ell}\right)$$

We embed all protein sequences using the E1-600m model (Jain et al., 2025) and apply mean pooling across the length dimension to obtain 1280 dimensional embeddings. We follow Hvarfner et al. (2024) and place a LogNormal ($\mathcal{LN}$) hyperprior over the lengthscale of the Matérn kernel:

$$\ell \sim \mathcal{LN}\left(\mu_0 + \frac{\log(D)}{2}, \sigma_0\right),$$

with $\mu_0 = \sqrt{2}$, $\sigma_0 = \sqrt{3}$, and $D = 1280$, without applying automatic relevance determination over embedding dimensions which degraded performance. We standardize all target variables independently to both improve hyperparameter optimization stability and to prevent particular targets from dominating the hypervolumes. We do not fix the signal variance σ_f^2 to 1.0 as in Hvarfner et al. (2024) which led to worse performance on a held-out test set but treat it as a learnable kernel hyperparameter which is multiplied with the Matérn kernel such that our final kernel formulation is

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \, k_{\text{Matérn}\frac{5}{2}}(\mathbf{x}, \mathbf{x}').$$

We place a Gamma prior with a mean of 1.0 on the signal variance, $\sigma_f^2 \sim \Gamma(\alpha, \beta)$, with shape parameter $\alpha = 5$ and rate parameter $\beta = 5$, providing adequate regularization. For the constant mean m, we use a Gaussian prior such that $m \sim \mathcal{N}(0, \sigma_m^2)$ with $\sigma_m = 3$. Lastly, we use a default BoTorch prior for the noise variance such that $\sigma_\epsilon^2 \sim \Gamma(1.1, 0.5)$ (Balandat et al., 2020).

For each dataset, we fit independent GPs to each target by optimizing the log marginal likelihood with an L-BFGS-S optimizer in double precision using BoTorch (Balandat et al., 2020). We fit the surrogate models to the full experimental datasets (whereas the PLMs were only trained on a subset of the data) to maximize the support and predictive power of the GPs as oracles.

Instead of averaging the activity across codons as a preprocessing step for the DFHR dataset, we train GPs for each activity/codon combination for a total of four models. For each prediction, we then take the average of each codon-specific GP. If a given sequence only has an experimental measurement for a single codon in one of the two activity objectives (as an example), we can still use it instead of requiring experimental measurements for all four activity/codon combinations. This way, we maximize the number of training sequences the oracle can leverage.

Dataset	Target	Spearman	NCDG@10%	MSE
PbrR	Pb_log2FC_increase	0.789	0.979	0.422
PbrR	Zn_log2FC_decrease	0.800	0.858	0.277
DHFR	0µg/mL_TMP_log2FC	0.503	0.947	0.587
DHFR	50µg/mL_TMP_log2FC	0.573	0.813	0.960
α-Amylase	activity_log2FC	0.581	0.895	1.296
α-Amylase	stability_log2FC	0.517	0.874	1.823
α-Amylase	expression_log2FC	0.510	0.868	0.479

Table 7: Predictive performance of GP regression models on held-out test data.

E.1 Reward model validation

To validate the use of the Gaussian process regression models for generative evaluations, we fit them to a subset of each dataset and evaluate their predictive performance on held-out test sets. As a result of the iterative creation of the PbrR and α-Amylase datasets, the mutation order varies substantially from the original training to test sets. The training partitions primarily consist of single mutations to a wild type, whereas the subsequently collected test sets contain upwards of 10 mutations. For the generative evaluations, we use all experimental data to train the oracle models while limiting the number of mutations in the generated sequences to 10 as described in Section 4.2. To approximate the training and inference distributions encountered in the generative evaluations as closely as possible, we therefore randomly place half of the designated test sequences in the training set.

We report the Spearman rank correlation, NCDG at 10% as well as the mean squared error across the three datasets and seven targets in Table 7. We observe good predictive performance for all configurations, justifying the use of the Gaussian process regressors as reward models for the generative evaluations.

E.2 Hypervolume estimation

To compute the expected hypervolume for a given set of k sequences, we rely on quasi-Monte Carlo (QMC) integration by drawing 256 samples from the posterior predictive distribution using Sobol sequences. For all QMC samples, we compute the resulting exact hypervolumes using the algorithms described in Beume et al. (2009); Fonseca et al. (2006); Guerreiro and Fonseca (2018) via the `moocore` library (López-Ibáñez, 2026) and take the average. We repeat this process for each $k \in \{12, 24, 48, 96, 192, 384\}$ a total of 100 times with different sampled sets to reduce variance and estimate uncertainty. For each dataset, we select a fixed reference point for hypervolume computations as the minimum observed reward independently across rewards.

F Re-parameterization of Energy Rank Alignment

In its original formulation, ERA introduces two parameters, β_{ERA} for sample diversity and γ_{ERA} for reference policy regularization, that carry different semantic meanings compared to their usual counterparts in the RL and preference optimization literature Chennakesavalu et al. (2025). In order to translate to the conventional definition of β referenced throughout this work, we first take the definition of the optimal target distribution as follows:

$$\pi^\star(y \mid x) = Z^{-1}(x) \exp\Big(-\tfrac{\beta_{\text{ERA}}}{1+\gamma_{\text{ERA}}}\big(U(x,y) - \beta_{\text{ERA}}^{-1}\gamma_{\text{ERA}} \log \pi_0(y \mid x)\big)\Big) \tag{14}$$

where π_0 is the reference policy and U is the energy to be minimized (negative reward in this setting). Consistent with the original work, we define the following bijection to re-parameterize ERA:

$$\beta \;=\; \frac{1+\gamma_{\text{ERA}}}{\beta_{\text{ERA}}}, \qquad w \;=\; \frac{\gamma_{\text{ERA}}}{1+\gamma_{\text{ERA}}} \in [0,1) \tag{15}$$

We introduce the w term as a measure of *the extent of regularization towards the reference policy versus a uniform distribution*, i.e. maximum-entropy regularization. When framed in

the context of the RL Optimization Problem 4, the ERA objective can be defined as follows:

$$\max_{\pi} \mathop{\mathbb{E}}_{x\sim\mathcal{P}} \mathop{\mathbb{E}}_{y\sim\pi(\cdot|x)} \Big[r(x,y)\Big] \;-\; \beta\Big[\, w\, \mathbb{D}_{\mathrm{KL}}(\pi \;||\; \pi_0) \;+\; (1-w)\, \mathbb{H}(\pi)\Big] \tag{16}$$

The reparameterized Equation 16 now adopts the field-standard definition of β for fair comparison with other baseline algorithms, and makes clear the role of w under this parameterization.