
Aligning Transformers with Continuous Feedback via Energy Rank Alignment

Shriram Chennakesavalu

Department of Chemistry
Stanford University
Stanford CA, 94305
shriramc@stanford.edu

Frank Hu

Department of Chemistry
Stanford University
Stanford CA, 94305
frankhu@stanford.edu

Sebastian Ibarraran

Department of Chemistry
Stanford University
Stanford CA, 94305
sebastian.ibarraran@stanford.edu

Grant M. Rotskoff

Department of Chemistry
Stanford University
Stanford CA, 94305
rotskoff@stanford.edu

Abstract

Searching through chemical space is an exceptionally challenging problem because the number of possible molecules grows combinatorially with the number of atoms. Large, autoregressive models trained on databases of chemical compounds have yielded powerful generators, but we still lack robust strategies for generating molecules with desired properties. This molecular search problem closely resembles the "alignment" problem for large language models, though for many chemical tasks we have a specific and easily evaluable reward function. Here, we introduce an algorithm called energy rank alignment (ERA) that leverages an explicit reward function to produce a gradient-based objective that we use to optimize autoregressive policies. We show theoretically that this algorithm is closely related to proximal policy optimization (PPO) and direct preference optimization (DPO), but has a minimizer that converges to an ideal Gibbs-Boltzmann distribution with the reward playing the role of an energy function. Furthermore, this algorithm is highly scalable, does not require reinforcement learning, and performs well relative to DPO when the number of preference observations per pairing is small. We deploy this approach to align molecular transformers and protein language models to generate molecules and protein sequences, respectively, with externally specified properties and find that it does so robustly, searching through diverse parts of chemical space.

1 Introduction

Large language models (LLMs) are trained on large corpora of text to autoregressively generate outputs. These models strongly reflect the distribution of the data on which they are trained [Ouyang et al. \[2022\]](#), and controlling the outputs to reflect externally imposed preferences is an increasingly important challenge for deployment. The aforementioned task, often called "alignment", requires either careful curation of training data or large sets of human preference data—both options are labor-intensive [Casper et al. \[2023\]](#). Reinforcement learning from human feedback (RLHF), a family of algorithms that employs these human preference datasets, has been widely employed to align instruction and chat models [Ouyang et al. \[2022\]](#), [Bai et al. \[2022\]](#), but it is both expensive to

acquire the training data and difficult to carry out in practice Casper et al. [2023]. Recent algorithmic developments, such as direct preference optimization (DPO) Rafailov et al. [2023], simplify the alignment framework by making the reward function implicit, but still require human preference data. While these algorithms succeed in constraining outputs, many “alignment”-like tasks require evaluation that would be difficult for human evaluators.

Generative sampling problems seeking to optimize a reward are common in chemistry, where comparing small molecules using a particular functional assay or computationally accessible property is often far easier than searching chemical space to identify novel compounds. Recent efforts to build large, domain-specific models for chemistry Chithrananda et al. [2020] have shown promising performance on both property prediction and reaction prediction tasks. Nevertheless, just as with LLMs, leveraging these models for molecule optimization requires first guiding “unaligned” models to favor important properties like synthetic accessibility or solubility. Here, we seek to productively search chemical space using transformers by introducing a new preference optimization algorithm, which we call energy rank alignment.

Our contribution: We formulate a generic alignment algorithm that we call *Energy Rank Alignment* or ERA that leverages an explicit reward function to guide autoregressive sampling while targeting specific properties or preferences. Unlike reward maximization in RL-based algorithms, the policy that minimizes our objective is designed to sample fluctuations around a maximal reward value to promote sample diversity. Our algorithm enables direct gradient-based optimization of a policy to match the ideal preference distribution and converges asymptotically to an optimal distribution with tuneable entropy and controllable regularization, which we show theoretically. The minimizers of our objective are closely related to the minimizer of PPO and DPO, but we have more direct control over the influence of the regularization relative to fluctuations around the maximum reward. In numerical experiments, we demonstrate that this algorithm successfully aligns molecular transformer model to identify a highly diverse set of chemicals with properties favored by our choice of reward. Finally, we demonstrate that ERA is able to align a protein language model to generate mutated protein sequences with desirable properties according to a computational reward model.

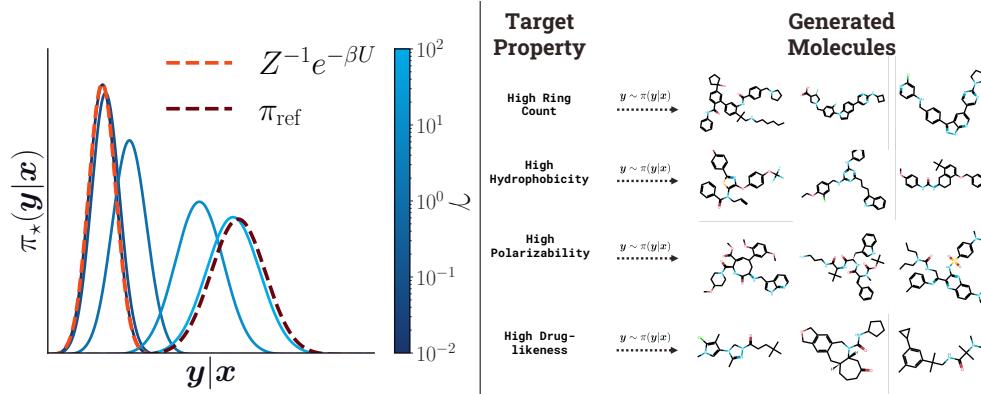


Figure 1: Energy rank alignment (ERA) enables targeting low-energy, high-reward regions with controllable fluctuations. Optimal policy approaches Boltzmann distribution with low regularization ($\gamma \rightarrow 0$) and reference policy with high regularization ($\gamma \rightarrow \infty$) (left). Aligned models can be used to sample molecules with desired chemical properties (right).

1.1 Related Work

Inverse molecular design tasks have a long history Lindsay et al. [1993] and many recent works have sought to apply machine learning to facilitate this difficult search problem Sanchez-Lengeling and Aspuru-Guzik [2018], Gromski et al. [2019], Gómez-Bombarelli et al. [2018]. While reinforcement learning has proved a popular strategy for molecular optimization Zhou et al. [2023], Sanchez-Lengeling and Aspuru-Guzik [2018], several recent studies have sought to use transformers Vaswani et al. [2017] trained on large databases of molecules represented with the text-based SMILES syntax Chithrananda et al. [2020], Schwaller et al. [2018], Wang et al. [2019], Bagal et al. [2022] for such tasks. Schwaller et al. [2019] utilized an atom-wise tokenization, which we also employ, to

train a transformer for the downstream task of reaction prediction. These “chemical language models” have been studied for applications on downstream tasks, including property prediction Bagal et al. [2022], Chithrananda et al. [2020] and reaction prediction Pesciullesi et al. [2020], Schwaller et al. [2018].

Building scalable strategies for alignment has attracted enormous attention because of the high cost and complexity of constraining LLM outputs. Much of the current paradigm is built on reinforcement learning from human feedback (RLHF) Ouyang et al. [2022]. Within this framework, human preferences provided in the form of pairwise rankings are first used to train a reward model, and subsequently that reward model is used to optimize a policy using, for example, proximal policy optimization (PPO) Schulman et al. [2017]. Rafailov et al. [2023] demonstrated that the reward model can be treated implicitly using a scheme that maximizes the likelihood of the preferences given an offline dataset. Because this approach does not require training a reward model, it has been named Direct Preference Optimization (DPO). Our work differs from both strategies; first, unlike RLHF, we do not employ reinforcement learning and instead develop an explicit, gradient-based objective for the optimal policy. Secondly, unlike DPO, we leverage an explicit reward function and add regularization transparently, both of which help to avoid greedy policies Azar et al. [2023]. However, like both approaches, we assume that the Bradley-Terry model Bradley and Terry [1952] of preference data is appropriate for the underlying target distribution.

Many recent works have built upon the ideas of RLHF and DPO, including studies on the effect of point-wise sampling of preference distributions Azar et al. [2023], investigations into the theoretical basis for contrastive methods for unlearning target datasets Zhang et al. [2024], and alternatives to the Bradley-Terry pairwise preference model Munos et al. [2023], An et al. [2023]. One recent study explores alignment in the context of inverse molecular design: Park et al. [2023] applies DPO to SMILES generators to increase the probability of activity for generated compounds against a drug target. However, they indicate that many preferences in chemistry are expressed as continuous signals, which is not suitable for DPO. Overcoming this limitation while maintaining the advantages of a direct gradient-based policy optimization strategy is a central goal of our current work. Our analysis and methodology directly addresses issues related to point-wise sampling because the explicit reward function eliminates overly greedy assignments of preference probabilities. Indeed, as discussed in Sec. 4, we see that DPO mode collapses where ERA shifts the policy towards the target distribution. While non-transitive preferences may arise in some settings, leading to a breakdown of the Bradley-Terry preference distribution model, by construction our target rewards are determined by quantitative evaluations of properties, and are therefore transitive.

2 Energy rank alignment

A policy is a conditional probability distribution $\pi(\cdot|\mathbf{x}) : \mathcal{Y} \rightarrow \mathbb{R}$; we generate an output \mathbf{y} from prompt \mathbf{x} . The spaces \mathcal{Y} and \mathcal{X} are discrete and finite, corresponding to sequences of tokenized outputs of the model with a maximum length. In alignment tasks, we begin with a pre-trained reference policy π_{ref} and seek to optimize a parametric, trainable policy π_{θ} to adapt the conditional sampling for a particular task or constraint.

Consider a prompt $\mathbf{x} \in \mathcal{X}$ and model outputs $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$ and a collection of preferences $\mathcal{D} = \{(\mathbf{y}_i \succ \mathbf{y}'_i; \mathbf{x}_i)\}_{i=1}^n$; the notation \succ indicates that \mathbf{y}_i is preferred to \mathbf{y}'_i . The conditional probability that $\mathbf{y} \succ \mathbf{y}'$ given \mathbf{x} can be modeled as a pairwise Boltzmann ranking within the Bradley-Terry model, i.e.,

$$p(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}) = \frac{e^{-\beta U(\mathbf{x}, \mathbf{y})}}{e^{-\beta U(\mathbf{x}, \mathbf{y})} + e^{-\beta U(\mathbf{x}, \mathbf{y}')}} \equiv \sigma(\beta U(\mathbf{x}, \mathbf{y}') - \beta U(\mathbf{x}, \mathbf{y})). \quad (1)$$

Here $\beta > 0$ is a constant, $\sigma(x) = (1 + e^{-x})^{-1}$ and we refer to $U : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ as an energy function to make clear the connection to statistical physics, but it is the negative reward within the RL framework for alignment.

To impose the preferences we minimize the objective

$$J(\pi) = \mathbb{E}_{\mathbf{x} \sim \nu} \left[\int U(\mathbf{x}, \mathbf{y}) d\pi(\mathbf{y}|\mathbf{x}) + \beta^{-1} \int (1 + \gamma) \log \pi(\mathbf{y}|\mathbf{x}) - \gamma \log(\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})) d\pi(\mathbf{y}|\mathbf{x}) \right], \quad (2)$$

where β^{-1} is a parameter controlling the magnitude of the entropic term, γ sets the scale of the Kullback-Leibler regularization compared with the energy term, and ν is a probability distribution

over the prompts $\nu \in \mathcal{P}(\mathcal{X})$. A proximal scheme for gradient descent on this objective corresponds to a gradient flow on J [Santambrogio \[2017\]](#), [Maas \[2011\]](#); the functional can be viewed as a free energy, and the corresponding flow is

$$\partial_t \pi_t = \nabla \cdot (\pi_t \nabla \delta_\pi J[\pi_t]), \quad (3)$$

and δ_π denotes the Fréchet derivative with respect to π . Assuming that π_0 has full support on $\mathcal{X} \times \mathcal{Y}$, the optimization converges asymptotically to a stationary policy which satisfies

$$\nabla \delta_\pi J[\pi_*] = 0 \iff \pi_* \propto e^{-\frac{\beta}{1+\gamma} U + \frac{\gamma}{\gamma+1} \log \pi_{\text{ref}}}, \quad (4)$$

and this minimizer is globally optimal. In the context of LLM alignment, a representation of the energy function $U : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is learned as a “reward model”, though we also consider tasks in which U is an easily evaluated function of the pair (\mathbf{x}, \mathbf{y}) . The optimal distribution π_* is a Gibbs-Boltzmann measure

$$\pi_*(\mathbf{y}|\mathbf{x}) = Z^{-1}(\mathbf{x}) \exp \left[-\frac{\beta}{1+\gamma} (U(\mathbf{x}, \mathbf{y}) - \beta^{-1} \gamma \log \pi_{\text{ref}}(\mathbf{y}|\mathbf{x})) \right] \quad (5)$$

where $Z(\mathbf{x})$ is the \mathbf{x} -dependent normalization constant. This expression makes clear the effect of β : when $\beta \rightarrow \infty$ (low temperature), the reward dominates and fluctuations around the maximal reward are small, which could lead to “mode-seeking”; when $\beta \rightarrow 0$ (high physical temperature) fluctuations around the maximal reward increase and the regularization term favors proximity to π_{ref} . Similarly, $\gamma \rightarrow 0$ recovers a Gibbs-Boltzmann distribution proportional to $e^{-\beta U}$ at inverse temperature β , while $\gamma \rightarrow \infty$ is dominated by the reference policy.

Loss functions for π_θ : Proximal Policy Optimization (PPO) optimizes an indirect, proximal objective to minimize an objective closely related to (2) (cf. [Appendix 3](#), [A](#)). Direct Preference Optimization (DPO) treats the negative reward function U implicitly and directly maximizes the likelihood of $p(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$. Our objectives differ from both approaches: like DPO, we directly optimize the policy using an explicit, gradient-based objective, but, in contrast, we use a reward function directly in our objective. The losses we build are thus amenable to both offline (samples from π_{ref}) and online (samples from π_θ) policy alignment, as explained below. Choosing to optimize the objective online has been shown to have important consequences on performance [Tajwar et al. \[2024\]](#), though we focus here on the setting where samples are drawn offline.

We directly optimize the Kullback-Leibler divergence between the entropy-regularized preference distribution $p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$ and the corresponding parametric preference distribution $p_\theta(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$. Explicitly, using the fact that conditional preference distribution is normalized, we obtain

$$\begin{aligned} D_{\text{KL}}^{(\mathbf{y}, \mathbf{y}')} (p_\gamma | p_\theta) &= p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}) \log \frac{p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})}{p_\theta(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})} + p_\gamma(\mathbf{y}' \succ \mathbf{y}|\mathbf{x}) \log \frac{p_\gamma(\mathbf{y}' \succ \mathbf{y}|\mathbf{x})}{p_\theta(\mathbf{y}' \succ \mathbf{y}|\mathbf{x})}, \\ &= p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}) \log \frac{p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})}{p_\theta(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})} + (1 - p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})) \log \frac{1 - p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})}{1 - p_\theta(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})}, \end{aligned} \quad (6)$$

where

$$p_\gamma := \sigma \left(\frac{\beta}{1+\gamma} \left[(U(\mathbf{x}, \mathbf{y}') - U(\mathbf{x}, \mathbf{y})) + \beta^{-1} \gamma \log \frac{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})} \right] \right). \quad (7)$$

This quantity is a well-defined KL divergence and is hence non-negative; the quantity vanishes when $p_\gamma = p_\theta$ on the observations \mathbf{y}, \mathbf{y}' . Furthermore, with access to an explicit reward model, all terms in (6) can be computed directly and

$$p_\theta(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}') = \frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_\theta(\mathbf{y}|\mathbf{x}) + \pi_\theta(\mathbf{y}'|\mathbf{x})} = \sigma \left(\log \frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_\theta(\mathbf{y}'|\mathbf{x})} \right). \quad (8)$$

To obtain a minimizer of the regularized objective defined in (2) we optimize

$$\mathcal{L}^{\text{ERA}}(\pi_\theta) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{y}, \mathbf{y}' \sim \pi_{\text{ref}}(\cdot|\mathbf{x})} D_{\text{KL}}^{(\mathbf{y}, \mathbf{y}')} (p_\gamma | p_\theta); \quad (9)$$

If the current policy overlaps with the target preference distribution, it may be useful to sample directly from the partially aligned policy, i.e., to use the “on-policy” formulation,

$$\mathcal{L}_{\text{on}}^{\text{ERA}}(\pi_\theta) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{y}, \mathbf{y}' \sim \pi_\theta(\mathbf{y}|\mathbf{x})} D_{\text{KL}}^{(\mathbf{y}, \mathbf{y}')} (p_\gamma | p_\theta) \quad (10)$$

instead of (9). One issue that arises with this scheme is differentiation with respect to the parameters of the policy θ because \mathbf{y} and \mathbf{y}' are decoded into discrete tokens, an operation that is not differentiable. To remedy this, we importance sample with a reference policy

$$\mathcal{L}_{\text{on}}^{\text{ERA}}(\pi_{\theta}) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{y}, \mathbf{y}' \sim \pi_{\text{ref}}(\mathbf{y}|\mathbf{x})} \frac{\pi_{\theta}(\mathbf{y}|\mathbf{x}) \pi_{\theta}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x}) \pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})} D_{\text{KL}}^{(\mathbf{y}, \mathbf{y}')} (p_{\gamma} | p_{\theta}). \quad (11)$$

This reweighting is straightforward and the importance weights should generally be appreciable, especially early in training when π_{θ} has not drifted far from π_{ref} . It is, of course, also natural to iteratively update π_{θ} using a previous iterate as the reference policy. In this work, we only use (9) as an objective and leave the on-policy objectives to future work. For an ablation of the parameters of ERA and a direct comparison in task performance to DPO, see Section C.3.1.

3 Theoretical Analysis

To understand the ERA loss function and its connection to the entropy regularized objective (2), we begin by establishing that the minimizers of (9) are of the form (5). We first define the notion of equivalence precisely.

Definition 3.1 *The conditional probability measures $\pi(\cdot|\mathbf{x})$ and $\pi'(\cdot|\mathbf{x})$ are conditionally equivalent if $\forall \mathbf{x} \in \mathcal{X}$, π and π' are such that $\sup_{\mathbf{y} \in \mathcal{Y}} |\pi(\mathbf{y}|\mathbf{x}) - \pi'(\mathbf{y}|\mathbf{x})| = 0$.*

We remark that this strong form of equivalence is appropriate on the finite, discrete spaces \mathcal{X} and \mathcal{Y} we consider here.

Lemma 3.1 *If π is conditionally equivalent to π' , then $\pi'_g(\cdot|\mathbf{x}) \propto \pi'(\cdot|\mathbf{x}) e^{g(\mathbf{x})}$ is conditionally equivalent to π for all functions $g : \mathcal{X} \rightarrow \mathbb{R}$ such that $\sup_{\mathbf{x} \in \mathcal{X}} |e^{g(\mathbf{x})}| < +\infty$.*

We prove Lemma 3.1 in Appendix A and use this simple lemma to prove the following result.

Proposition 3.2 *Suppose $\pi(\cdot|\mathbf{x}) \in \mathcal{P}(\mathcal{Y})$ and that $\text{supp}(\pi) = \text{supp}(\pi_{\text{ref}})$. Let $\beta > 0$, $\gamma \geq 0$ and that the reward model is such that $\sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X} \times \mathcal{Y}} |e^{-U(\mathbf{x}, \mathbf{y})}| < +\infty$. Then, the minimizer of \mathcal{L}^{ERA} is conditionally equivalent to π_{\star} .*

First, we verify that any probability measure $\pi_g(\mathbf{y}|\mathbf{x}) \propto \exp(-\frac{\beta}{1+\gamma}(U(\mathbf{x}, \mathbf{y}) - \beta^{-1}\gamma \log \pi_{\text{ref}}(\mathbf{y}|\mathbf{x})) + g(\mathbf{x}))$ minimizes the objective. Because \mathcal{L}^{ERA} is non-negative, it suffices to show that for all pairs \mathbf{y}, \mathbf{y}' , $D_{\text{KL}}^{(\mathbf{y}, \mathbf{y}')} (p_{\gamma} | p_{\theta}) \equiv 0$. This follows immediately from the cancellation in the preference probability p_{γ} of $e^{g(\mathbf{x})}$ after factorization in (5).

Now, suppose that $\pi(\mathbf{y}|\mathbf{x}) \neq \exp\left(-\frac{\beta}{1+\gamma}(U(\mathbf{x}, \mathbf{y}) - \beta^{-1}\gamma \log \pi_{\text{ref}}(\mathbf{y}|\mathbf{x}))\right)$ where we have taken $g(\mathbf{x}) = 0$ without loss of generality and $\pi := \pi_g$. Assume that for all pairs \mathbf{y}, \mathbf{y}' , the divergence $D_{\text{KL}}^{(\mathbf{y}, \mathbf{y}')} (p_{\gamma} | p_{\theta}) \equiv 0$ which is required of a minimizer. Equivalently, it must be the case that for all \mathbf{y}, \mathbf{y}' ,

$$\frac{\pi(\mathbf{y}|\mathbf{x})}{\pi(\mathbf{y}|\mathbf{x}) + \pi(\mathbf{y}'|\mathbf{x})} = \frac{\pi_{\star}(\mathbf{y}|\mathbf{x})}{\pi_{\star}(\mathbf{y}|\mathbf{x}) + \pi_{\star}(\mathbf{y}'|\mathbf{x})} \implies \frac{\pi(\mathbf{y}'|\mathbf{x})}{\pi(\mathbf{y}|\mathbf{x})} = \frac{\pi_{\star}(\mathbf{y}'|\mathbf{x})}{\pi_{\star}(\mathbf{y}|\mathbf{x})}, \quad (12)$$

from which we see that

$$\pi(\mathbf{y}|\mathbf{x}) = \frac{\pi(\mathbf{y}'|\mathbf{x})}{e^{-\frac{\beta}{1+\gamma}(U(\mathbf{x}, \mathbf{y}') - \beta^{-1}\gamma \log \pi_{\text{ref}}(\mathbf{y}'|\mathbf{x}))}} e^{-\frac{\beta}{1+\gamma}(U(\mathbf{x}, \mathbf{y}) - \beta^{-1}\gamma \log \pi_{\text{ref}}(\mathbf{y}|\mathbf{x}))}. \quad (13)$$

By construction, $\pi(\mathbf{y}|\mathbf{x})$ does not depend on \mathbf{y}' so the prefactor must be purely a function of \mathbf{x} , which completes the proof, using Lemma 3.1. A detailed theoretical analysis of the ERA objective is provided in Appendix A. This analysis compares the ERA loss with the DPO and PPO objectives and demonstrates that ERA yields correct global rankings in the low data limit, while DPO maximizes pairwise margin.

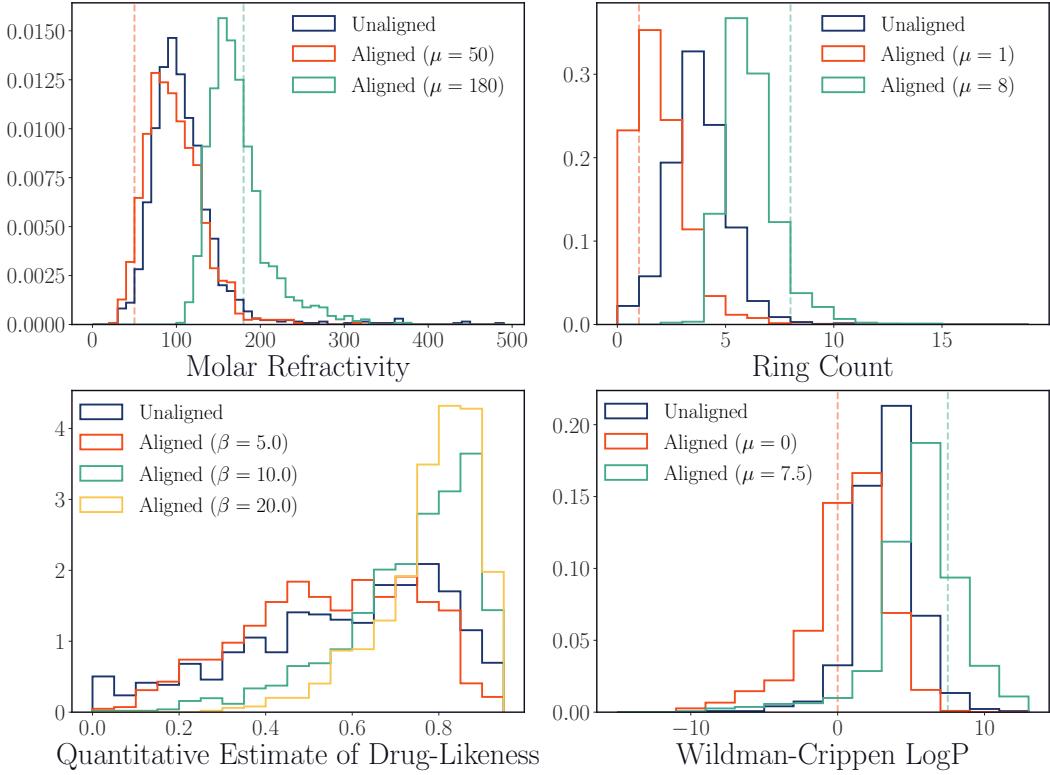


Figure 2: Unprompted molecular generator alignment. Distributions of different chemical properties for molecules sampled from aligned and unaligned policies. The center of the harmonic potential, μ , is varied for MR ($\beta = 1.0$), Ring Count ($\beta = 1.0$), and LogP ($\beta = 10.0$), while β is varied for QED. All experiments were run with no regularization to the reference policy ($\gamma = 0$).

4 Experiments

We test ERA on both chemical and language tasks to shed light on the following questions: 1) Can we use ERA to robustly fine-tune our model to generate samples according to a desired distribution? 2) What is the effect of changing the inverse-temperature β during ERA? 3) Do we maintain sample diversity (and validity) without regularizing to remain close to a reference policy, and what is the effect of increased regularization? 4) Can we simultaneously target multiple properties with high fidelity, and how can we trade off between desired properties?

4.1 Generating small molecules with desired properties

We use a decoder-only representation for the molecular generator [Bagal et al. \[2022\]](#), where the generator has 2 layers, an embedding dimension of 512, a vocabulary of 324 tokens, and totals 3.5M parameters. Starting from a random initialization, we carry out pretraining on a dataset of 2.4M small molecules from the ChEMBL database [Zdrazil et al. \[2024\]](#) for 180 epochs. For sampling from our molecular generator, we use top- k sampling with $k = 5$ and a sampling temperature of $T = 1$ in all experiments for consistency. This version of the model is not conditioned on a prompt and generates a small molecule (represented as a SMILES string) given just a start-of-sequence token.

Central to ERA is, of course, access to a computable energy function. As a proof-of-concept, we first consider 5 different properties for which the corresponding energy function is easily evaluable: Quantitative Estimate of Drug-Likeness (QED) [Bickerton et al. \[2012\]](#), Wildman-Crippen LogP (LogP) [Wildman and Crippen \[1999\]](#), Ring Count, Molar Refractivity (MR) [Wildman and Crippen \[1999\]](#), and Tanimoto Similarity [Rogers and Tanimoto \[1960\]](#) (Section 4.1.1, 4.1.2). Briefly, LogP is a measure of the hydrophobicity of a molecule, MR is a measure of the polarizability of the molecule, and Tanimoto similarity is a measure of the similarity between two molecules (see Appendix C.2). We

then investigate ERA in a more challenging context: generating small-molecules with high predicted bioactivity for two kinases (Section 4.1.3).

4.1.1 Unprompted molecular alignment on RDKit oracles

First, we independently target four different properties using ERA with an unprompted molecular generator (Fig. 2). Using the pretrained model as our reference policy, we generate a dataset $\mathcal{D} = \{\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, U(\mathbf{y}_1^{(i)}), U(\mathbf{y}_2^{(i)})\}_{i=1}^N$ and carry out energy rank alignment on π_θ , where π_θ is initialized using the weights of π_{ref} . Here, $\mathbf{y}_1, \mathbf{y}_2 \sim \pi_{\text{ref}}$ and \mathbf{y} and $U(\mathbf{y})$ denote the generated molecule and its corresponding energy, respectively. For MR, Ring Count, and LogP, we define the energy U to be a harmonic potential centered at a target value. For QED, we define the energy to be the negative logarithm of QED and vary β to assess its impact on alignment (see Tables S2, S3). In Fig. 2, we see that we successfully shift the distribution to target means that are both greater and lower than the average value of MR, Ring Count, and LogP under the reference policy. Furthermore, in the alignment of QED, we observe the effect of changing β on the learned policy; with increased β , the learned policy concentrates around low-energy samples (i.e. near QED = 1), and with lower β , the learned policy samples a greater range of QED values, as expected. We note that for each of these four experiments, we did not regularize towards the reference policy (i.e. $\gamma = 0$). Even so, we were able to maintain both sample diversity and appreciable sample validity (see Fig. S8 and Table S6).

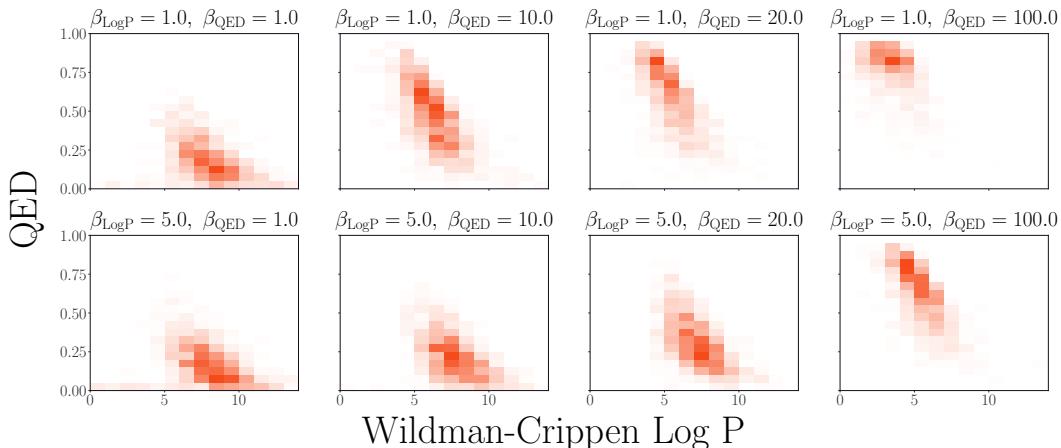


Figure 3: Unprompted multi-property molecular generator alignment. 2D histograms of LogP versus QED for different combinations of property-specific β illustrating a clear trade-off when performing multi-property alignment. Relative increases in β for a given property target higher values for that property. All experiments were run with no regularization to the reference policy ($\gamma = 0$).

Many molecular design tasks require balancing multiple properties, and designing an objective for multi-property alignment is straightforward within the ERA framework. To demonstrate this, we generate molecules with both high QED and LogP using ERA with an energy function weighted by property-specific β : $U = \beta_{\text{QED}} U_{\text{QED}} + \beta_{\text{LogP}} U_{\text{LogP}}$ (see Tables S2, S7 for details on the energy function). We carry out ERA with different pairs of $(\beta_{\text{QED}}, \beta_{\text{LogP}})$ using the same procedure as above, and from Fig. 3, we see that we target multiple properties with varying fidelity by simply modulating the value of property-specific β . Ultimately, increasing the β for an individual property enables us to favor higher values of that property in multi-property alignment setting. In this case, we also do not regularize with the KL-divergence to the reference policy and again maintain sample diversity and validity (see Fig. S9 and Table S7).

4.1.2 Prompted molecular alignment on RDKit oracles

Inspired by the task of lead optimization in drug discovery efforts Keserii and Makara [2009], we ask whether we can use ERA to train a molecular generator that can sample a molecule that is both similar to the prompt molecule *and* also exhibits some desired property. First, we fine-tune the pretrained molecular generator to enable prompted molecular generation (see Appendix C.3.3) and use this fine-tuned model as our reference policy for all prompted molecular alignment tasks. This reference

policy disproportionately samples molecules that are identical (i.e. a Tanimoto similarity of 1.0) to the prompt molecule (see Fig. S10), so we carry out multi-property alignment on this reference policy to generate molecules that are similar—but not identical—to the prompt molecule and also have a high drug-likeness as measured by QED. Using ERA, we optimize the reference policy with a generated dataset $\mathcal{D} = \{(\mathbf{y}_1^{(i)}, \mathbf{x}^{(i)}), (\mathbf{y}_2^{(i)}, \mathbf{x}^{(i)}), U(\mathbf{y}_1^{(i)}, \mathbf{x}^{(i)}), U(\mathbf{y}_2^{(i)}, \mathbf{x}^{(i)})\}_{i=1}^N$, where we sample four molecules for each prompt molecule from the reference policy and consider all possible preference pairs for a total of six preference pairs per prompt molecule (see Appendix C.2 for full details on the energy used).

We observe that the per-prompt average QED under the optimized policy for a given prompt is higher than the corresponding average under the reference policy (Fig. S10). Furthermore, we see that we are able to sample a diverse set of molecules that are chemically similar to the prompt molecule, and also chemically valid (see Fig. S11, Table S8). We repeat the experiment with a related objective of generating molecules similar to the prompt molecule with a high LogP instead and again observe that we increase the per-prompt average LogP under the optimized policy relative to the reference policy without degrading sample diversity and validity. For both of these experiments, we required regularization to the reference policy ($\gamma > 0$). With no regularization, the aligned generator would almost exclusively sample sequences that were chemically invalid (< 25% chemical validity). Finally, we note that the increases in QED and LogP in Fig. S10 are smaller relative to the increases in Fig. 2 because the samples are now conditioned to remain proximal to the prompt molecule, which restricts the chemical space that can be explored.

4.1.3 Unprompted molecular alignment on protein-ligand docking oracles

	GSK3 β top-100		JNK3 top-100	
	mean score	IntDiv	mean score	IntDiv
ERA	0.996 ± 0.000	0.219 ± 0.002	0.987 ± 0.001	0.264 ± 0.005
MolRL-MGPT	1.000 ± 0.000	0.362 ± 0.015	0.961 ± 0.010	0.372 ± 0.025
GFlowNet	0.649 ± 0.072	0.715 ± 0.104	0.437 ± 0.219	0.716 ± 0.145
GraphGA	0.919 ± 0.016	0.365 ± 0.024	0.875 ± 0.025	0.380 ± 0.015
JT-VAE	0.235 ± 0.083	0.770 ± 0.067	0.159 ± 0.040	0.781 ± 0.127
REINVENT	0.965 ± 0.011	0.308 ± 0.035	0.942 ± 0.019	0.368 ± 0.021

Table 1: Mean scores and internal diversities (IntDiv) of experiments on GSK3 β and JNK3 tasks averaged across 5 random seeds. For each task, 20k molecules were sampled, and metrics were computed on top-100 scoring *valid*, *novel* and *unique* molecules filtered from the initial 20K samples (i.e. molecules not in dataset and molecules not previously sampled). Compared to state-of-the-art methods, ERA samples more diverse molecules with higher predicted docking scores. Results for compared methods are reproduced from [Hu et al. \[2023\]](#).

We next investigate the performance of ERA in designing compounds that have high predicted docking scores for the kinases JNK3 and GSK3 β . For each of these targets, we use an *in silico* oracle that predicts docking scores, ranging from 0 to 1, where a higher value corresponds to stronger predicted score [Sun et al. \[2017\]](#). Using only data from ChemBL, we first carry out a short supervised fine-tuning step on all molecules in ChemBL with an oracle score above 0.5 (7386 molecules for JNK3 and 43381 for GSK3 β). Using this fine-tuned model as our reference policy, we then carry out alignment using ERA ($\beta=100$ and $\gamma=0$), where we use a comparably high β to target molecules with high activity. As with the QED alignment runs in Section 4.1.1, we define the energy for this task as the negative logarithm of the oracle score.

From the aligned models, we sample 20000 molecules (see Fig. S12) and tabulate metrics of the top-100 performing molecules (see Table 1). We note that the molecules in the top-100 are both *novel* and *unique* after filtering to exclude any molecules that are present in the ChemBL dataset and any repeated molecules. For GSK3 β , our mean score is marginally lower than the best performing method but the diversity in sampled molecules is significantly higher (i.e. lower IntDiv). For JNK3 our mean score is significantly higher than the best performing method *and* the diversity in sampled molecules is higher than any method. The inference costs are notably low for our approach; sampling 20000 molecules and filtering takes only minutes on a single GPU.

We additionally measure sample efficiency using the top-10 AUC metric Gao et al. [2022], Bou et al. [2024], which is the area under the curve (AUC) of the mean property value of the top-10 performing molecules versus the number of oracle calls (see Fig. S13 and Table S9). We likewise only include novel, unique, and valid molecules in this analysis. We observe that we are able to generate novel and unique high-scoring molecules, with high sample efficiency especially in comparison to existing state-of-the-art methods such as REINVENT, GraphGA, PPO, and PPOD Gao et al. [2022], Bou et al. [2024]. Ultimately, high sample efficiency is crucial in settings where evaluation is expensive, which will generally be true for most real-world chemical and biological tasks (e.g. wet-lab experiment). Finally, we also perform Glide Standard Precision Friesner et al. [2004] docking on the top-scoring molecules according to the oracles (score of 1.0) against their respective receptors. We observe that the diverse set of sampled molecules exhibit chemically plausible docked poses obtained from a physics-based docking approach (Fig. 4).

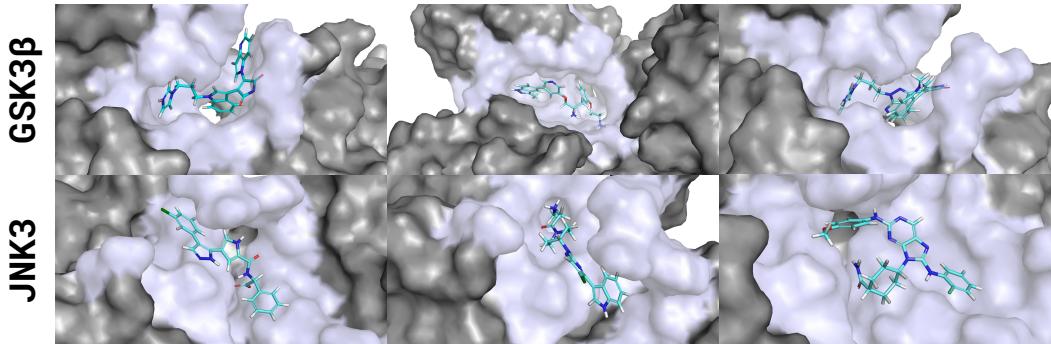


Figure 4: Visualization of three generated ligands docked against the GSK3 β kinase target (top) and three generated ligands docked against the JNK3 kinase target (bottom). In each case, these were the three molecules with the best (most negative) Glide Standard Precision docking scores and oracle scores of 1.0.

4.2 Directed evolution of proteins with ERA

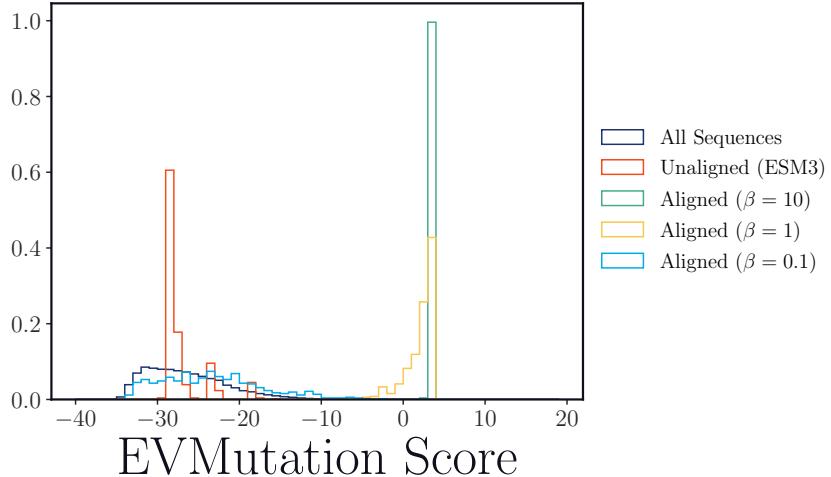


Figure 5: Alignment of ESM3-1.4B with $\beta=0, 0.1, 1.0, 10.0$ and $\gamma=0.001$ on the task of maximizing EVmutation score. Positions 182, 183, 184, and 186 of the TrpB parent sequence were masked and ESM3-1.4B predicted amino acids at those sites. The distribution of the EVmutation scores for generated sequences shifts significantly as β is increased.

We also consider the performance of ERA in a large-molecule setting, namely ML-guided directed evolution of proteins. Directed evolution campaigns aim to optimize a protein sequence toward some desired property of interest via iterative mutagenesis, library screening, and selection of best variants

Wang et al. [2021]. This has become a widely used methodology in protein engineering but comes with key limitations. The inherently iterative nature of directed evolution campaigns can lead to costly and time-consuming experimental campaigns, and meaningfully understanding the effects of protein mutations on protein activity can often be difficult. These challenges have led to the application of machine learning methods to more efficiently guide directed evolution campaigns Yang et al. [2024] Shanker et al. [2024]. Given the success of ERA in guiding the optimization of small-molecules using a SMILES language representation, we examined whether ERA could be used to optimize large protein molecules using a protein language (i.e. primary sequence) representation.

There has been significant recent effort to design and train large protein language models (PLMs) Lin et al. [2023], Hayes et al. [2024]. Furthermore, these models have demonstrated remarkable capabilities across a number of protein tasks Widatalla et al. [2024], Shanker et al. [2024]. As such, we decided to use the state-of-the-art ESM3-1.4B Hayes et al. [2024] as our pretrained model, for which we carried out alignment using ERA. Despite the multimodal nature of ESM3, here, we only focus on generating primary-sequence-based representations of proteins.

We consider directed evolution of the β -subunit of tryptophansynthase (TrpB) from *Thermotoga maritima*, an enzyme that catalyzes tryptophan production Buller et al. [2015]. Here, we seek to evolve the protein to increase its evolutionary fitness. In this work, we do not have access to experimental validation and so we evaluate the fitness of sequences using the computationally evaluable EVmutation score, an oracle that is predictive of a variant sequence’s performance relative to the parent sequence in its native function Hopf et al. [2017].

As in other directed evolution campaigns for the TrpB protein Yang et al. [2023], we consider mutating four different sites to one of the 20 standard amino acids. We randomly sampled 512 mutated sequences, emulating a random mutagenesis experiment. Using ESM3-1.4B as our reference model, we carry out alignment using ERA with various $\beta = (0.1, 1.0, 10.0)$ and $\gamma = 0.001$ and plot the results in Fig. 5 (see Appendix D for more details). We observe that with higher β , we are able to sample mutants with the highest possible EVmutation score in a single round of alignment. These results are promising for the application of ERA in directed evolution campaigns and future work will focus on the guidance of wet-lab directed evolution campaigns in conjunctions with multi-round, on-policy ERA.

5 Conclusions and Limitations

This paper introduces energy rank alignment, a simple and effective algorithm for policy optimization with an explicit reward model. We find that ERA is stable without extensive hyperparameter tuning, and sufficiently general to successfully align application-specific transformers for chemical search problems and protein language models. The algorithm exhibits strong performance with a variety of reward models, even ones with relatively weak signal. We analyze the minimizers of the ERA objective and find that they differ from the minimizers of popular policy alignment algorithms DPO and PPO in an important way: unlike PPO, the strength of regularization to the reference policy that we add is controlled by a parameter γ , while the entropy of the target distribution is independently tuned by a distinct parameter β . This means that we can avoid greedy policies by keeping β small—amplifying fluctuations around the optimum of the reward model ($-U$)—while reducing the influence of the reference policy by taking γ small. Our objective leads to easily interpretable sample-wise gradients which highlight the importance of a reward model relative to DPO in the sampled objective.

Limitations: First, our approach requires a reward model, which can be difficult to train or design, especially for complex tasks. While we observed that ERA makes an appreciable impact even with weak supervision, this sort of proxy may not be available for more complex tasks. For example, optimizing small molecules for high binding affinity to a target protein would require expensive and noisy evaluations of a reward model, which likely limits the scope of molecular design to problems where the reward can be computed somewhat efficiently. A second limitation of our present work is that we do not train the molecular transformer to favor synthetic accessibility nor do we explicitly seek to obtain molecules that are easily synthesized experimentally. There are models that seek to evaluate synthesizability computationally that could be used in our rewards, which we plan to explore in future work Coley et al. [2018].

References

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

G. An, J. Lee, X. Zuo, N. Kosaka, K.-M. Kim, and H. O. Song. Direct Preference-based Policy Optimization without Reward Modeling. *Advances in Neural Information Processing Systems*, 36: 70247–70266, Dec. 2023.

M. G. Azar, M. Rowland, B. Piot, D. Guo, D. Calandriello, M. Valko, and R. Munos. A General Theoretical Paradigm to Understand Learning from Human Preferences, Nov. 2023.

V. Bagal, R. Aggarwal, P. K. Vinod, and U. D. Priyakumar. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, May 2022. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00600.

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, Apr. 2022.

G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, Feb. 2012. ISSN 1755-4330, 1755-4349. doi: 10.1038/nchem.1243.

A. Bou, M. Thomas, S. Dittert, C. Navarro, M. Majewski, Y. Wang, S. Patel, G. Tresadern, M. Ahmad, V. Moens, W. Sherman, S. Sciabola, and G. D. Fabritiis. Acegen: Reinforcement learning of generative chemical agents for drug discovery. *Journal of Chemical Information and Modeling*, 64(15):5900–5911, 2024. doi: 10.1021/acs.jcim.4c00895. URL <https://pubs.acs.org/doi/10.1021/acs.jcim.4c00895>.

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 0006-3444. doi: 10.2307/2334029.

A. R. Buller, S. Brinkmann-Chen, D. K. Romney, M. Herger, J. Murciano-Calles, and F. H. Arnold. Directed evolution of the tryptophan synthase β -subunit for stand-alone function recapitulates allosteric activation. *Proceedings of the National Academy of Sciences*, 112(47):14599–14604, 2015. doi: 10.1073/pnas.1516401112. URL <https://www.pnas.org/doi/full/10.1073/pnas.1516401112>.

S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, M. Damani, S. Slocum, U. Anwar, A. Siththaranjan, M. Nadeau, E. J. Michaud, J. Pfau, D. Krasheninnikov, X. Chen, L. Langosco, P. Hase, E. Büyükköse, A. Dragan, D. Krueger, D. Sadigh, and D. Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback, Sept. 2023.

A. Chen, S. Malladi, L. H. Zhang, X. Chen, Q. Zhang, R. Ranganath, and K. Cho. Preference learning algorithms do not learn preference rankings. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2024.

S. Chithrananda, G. Grand, and B. Ramsundar. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. In *Machine Learning for Molecules Workshop at NeurIPS*, 2020.

C. W. Coley, L. Rogers, W. H. Green, and K. F. Jensen. SCScore: Synthetic Complexity Learned from a Reaction Corpus. *Journal of Chemical Information and Modeling*, 58(2):252–261, Feb. 2018. ISSN 1549-9596. doi: 10.1021/acs.jcim.7b00622.

R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, D. E. Shaw, P. Francis, and P. S. Shenkin. Glide: A new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, 47(7), 2004. doi: <https://doi.org/10.1021/jm0306430>. URL <https://pubs.acs.org/doi/10.1021/jm0306430>.

W. Gao, T. Fu, and J. S. and Connor W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2022.

P. S. Gromski, A. B. Henson, J. M. Granda, and L. Cronin. How to explore chemical space using algorithms and automation. *Nature Reviews Chemistry*, 3(2):119–128, 2019.

R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, Feb. 2018. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00572.

T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, R. Badkundri, I. Shafkat, J. Gong, A. Derry, R. S. Molina, N. Thomas, Y. Khan, C. Mishra, C. Kim, L. J. Bartie, M. Nemeth, P. D. Hsu, T. Sercu, S. Candido, and A. Rives. Simulating 500 million years of evolution with a language model, July 2024.

T. A. Hopf, J. B. Ingraham, F. J. Poelwijk, C. P. I. Schärfe, M. Springer, C. Sander, and D. S. Marks. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135, 2017. doi: 10.1038/nbt.3769. URL <https://www.nature.com/articles/nbt.3769>.

X. Hu, G. Liu, Y. Zhao, and H. Zhang. De novo drug design using reinforcement learning with dynamic vocabulary. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2023.

K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. W. Coley, C. Xiao, J. Sun, and M. Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.

G. M. Keserü and G. M. Makara. The influence of lead discovery strategies on the properties of drug candidates. *Nature Reviews Drug Discovery*, 8(3):203–212, Mar. 2009. ISSN 1474-1776, 1474-1784. doi: 10.1038/nrd2796.

Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. Dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, Mar. 2023. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.ade2574. URL <https://www.science.org/doi/10.1126/science.ade2574>.

R. K. Lindsay, B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg. Dendral: A case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence*, 61(2):209–261, June 1993. ISSN 00043702. doi: 10.1016/0004-3702(93)90068-M.

J. Maas. Gradient flows of the entropy for finite Markov chains. *Journal of Functional Analysis*, 261(8):2250–2292, Oct. 2011. ISSN 0022-1236. doi: 10.1016/j.jfa.2011.06.009.

R. Munos, M. Valko, D. Calandriello, M. G. Azar, M. Rowland, Z. D. Guo, Y. Tang, M. Geist, T. Mesnard, A. Michi, M. Selvi, S. Girgin, N. Momchev, O. Bachem, D. J. Mankowitz, D. Precup, and B. Piot. Nash Learning from Human Feedback, Dec. 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. 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. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022.

R. Park, R. Theisen, N. Sahni, M. Patek, A. Cichońska, and R. Rahman. Preference Optimization for Molecular Language Models, Oct. 2023.

G. Pesciullesi, P. Schwaller, T. Laino, and J.-L. Reymond. Transfer learning enables the molecular transformer to predict regio- and stereoselective reactions on carbohydrates. *Nature Communications*, 11(1):4874, Sept. 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-18671-7.

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 A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc., 2023.

D. J. Rogers and T. T. Tanimoto. A Computer Program for Classifying Plants: The computer is programmed to simulate the taxonomic process of comparing each case with every other case. *Science*, 132(3434):1115–1118, Oct. 1960. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.132.3434.1115.

B. Sanchez-Lengeling and A. Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, July 2018. doi: 10.1126/science.aat2663.

F. Santambrogio. {Euclidean, Metric, and Wasserstein} gradient flows: An overview. *Bulletin of Mathematical Sciences*, 7(1):87–154, Apr. 2017. ISSN 1664-3615. doi: 10.1007/s13373-017-0101-1.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms, Aug. 2017.

P. Schwaller, T. Gaudin, D. Lányi, C. Bekas, and T. Laino. “Found in Translation”: Predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chemical Science*, 9(28):6091–6098, 2018. doi: 10.1039/C8SC02339E.

P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, Sept. 2019. ISSN 2374-7943, 2374-7951. doi: 10.1021/acscentsci.9b00576.

V. R. Shanker, T. U. J. Bruun, B. L. Hie, and P. S. Kim. Unsupervised evolution of protein and antibody complexes with a structure-informed language model. *Science*, 385(6704):46–53, 2024. doi: 10.1126/science.adk8946. URL <https://www.science.org/doi/full/10.1126/science.adk8946>.

J. Sun, N. Jeliazkov, V. Chupakhin, J.-F. Golib-Dzib, O. Engkvist, L. Carlsson, J. Wegner, H. Ceulemans, I. Georgiev, V. Jeliazkov, N. Kochev, T. J. Ashby, and H. Chen. Escape-db: an integrated large scale dataset facilitating big data analysis in chemogenomics. *Journal of Cheminformatics*, 9(17), 2017. doi: <https://doi.org/10.1186/s13321-017-0203-5>. URL <https://jcheminf.biomedcentral.com/articles/10.1186/s13321-017-0203-5#citeas>.

F. Tajwar, A. Singh, A. Sharma, R. Rafailov, J. Schneider, T. Xie, S. Ermon, C. Finn, and A. Kumar. Preference Fine-Tuning of LLMs Should Leverage Suboptimal, On-Policy Data, Apr. 2024.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

S. Wang, Y. Guo, Y. Wang, H. Sun, and J. Huang. SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, BCB '19*, pages 429–436, New York, NY, USA, Sept. 2019. Association for Computing Machinery. ISBN 978-1-4503-6666-3. doi: 10.1145/3307339.3342186.

Y. Wang, P. Xue, M. Cao, T. Yu, S. T. Lane, and H. Zhao. Directed evolution: Methodologies and applications. *Chemical Reviews*, 121(20):12384–12444, 2021. doi: 10.1021/acs.chemrev.1c00260. URL <https://pubs.acs.org/doi/full/10.1021/acs.chemrev.1c00260>.

T. Widatalla, R. Rafailov, and B. Hie. Aligning protein generative models with experimental fitness via direct preference optimization, May 2024.

S. A. Wildman and G. M. Crippen. Prediction of Physicochemical Parameters by Atomic Contributions. *Journal of Chemical Information and Computer Sciences*, 39(5):868–873, Sept. 1999. ISSN 0095-2338, 1520-5142. doi: 10.1021/ci9903071.

J. Yang, J. Ducharme, K. E. Johnston, F.-Z. Li, Y. Yue, and F. H. Arnold. Decoil: Optimization of degenerate codon libraries for machine learning-assisted protein engineering. *ACS Synthetic Biology*, 12(8):2444–2454, 2023. doi: 10.1021/acssynbio.3c00301. URL <https://pubs.acs.org/doi/full/10.1021/acssynbio.3c00301>.

J. Yang, R. G. Lal, J. C. Bowden, R. Astudillo, M. A. Hameedi, S. Kaur, M. Hill, Y. Yue, and F. H. Arnold. Active learning-assisted directed evolution, July 2024.

B. Zdrrazil, E. Felix, F. Hunter, E. J. Manners, J. Blackshaw, S. Corbett, M. de Veij, H. Ioannidis, D. M. Lopez, J. F. Mosquera, M. P. Magarinos, N. Bosc, R. Arcila, T. Kizilören, A. Gaulton, A. P. Bento, M. F. Adasme, P. Monecke, G. A. Landrum, and A. R. Leach. The ChEMBL Database in 2023: A drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, Jan. 2024. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkad1004.

R. Zhang, L. Lin, Y. Bai, and S. Mei. Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning, Apr. 2024.

Z. Zhou, J. Liu, C. Yang, J. Shao, Y. Liu, X. Yue, W. Ouyang, and Y. Qiao. Beyond One-Preference-Fits-All Alignment: Multi-Objective Direct Preference Optimization, Dec. 2023.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: **[Yes]**

Justification: The abstract clearly outlines the problem of searching over vast chemical spaces as an alignment problem, gives a high-level overview of our method, highlights the key differences between our method and similar approaches, and summarizes the key advantages of our algorithm. We clearly define the current scope of our investigation by describing our experiments.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: **[Yes]**

Justification: We address several potential limitations of our work at the end of Section 4.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Section 2 provides a thorough theoretical analysis of the ERA algorithm, and further details are in Sections A and B of the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Between Section 3 (Experiments) and Sections D and E of the Appendix (Details for molecular generator experiments and Details for PLM experiments, respectively), we describe the experimental procedure used to obtain each result.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: We will release the code as open source upon submission of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: The exact settings and hyperparameters used for training and testing are provided in descriptions of the experiments in Section 3 and in Sections D and E of the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: The statistical significance of all the results is discussed throughout the body of the paper in Section 3 and in Sections D and E of the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Computational resources used are discussed in Section F of the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: Yes, the work is compliant with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: The broader societal impacts of this work is discussed in Section G.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Justification: We use public data and models for all PLM experiments. There are no safety concerns with the chemical alignment models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: Yes, we cite the appropriate works throughout the main text and the Appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The code will be released as open-source and will have adequate documentation for reproducing experiments.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: We did not use human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: We did not use human subjects in this work.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

,

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We did not use any LLMs in this study.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Detailed Theoretical Analysis

Set-up, notation, and assumptions Let \mathcal{X} and \mathcal{Y} be discrete spaces; each element of one of these spaces is a finite-length sequence of tokens within a fixed dictionary on which an autoregressive generative model is trained. The resulting models yield “policies”, which are conditional probability distributions $\pi(\cdot|\mathbf{x}) \in \mathcal{P}(\mathcal{Y})$ for each $\mathbf{x} \in \mathcal{X}$. Throughout, we assume that our policies have full support on \mathcal{Y} for each \mathbf{x} , meaning that $\inf_{\mathbf{y}, \mathbf{x} \in \mathcal{Y} \times \mathcal{X}} \pi(\mathbf{y}|\mathbf{x}) > 0$. Because the spaces are discrete, we make no strong restrictions on the regularity or coerciveness of the reward model $-U : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. The only requirement to ensure the existence of an optimal probability distribution is that $\sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X} \times \mathcal{Y}} |e^{-U(\mathbf{x}, \mathbf{y})}| < +\infty$, which maintains full support of the distribution. Though it plays little role in theoretical analysis, we also denote by $\nu \in \mathcal{P}(\mathcal{X})$ the probability distribution over the prompts \mathbf{x} .

Goals of the analysis presented here The main purpose of this section is to establish that globally minimizing the loss (9) yields a global minimizer of the regularized policy objective (2). A secondary goal is to clearly articulate the theoretical advantages of ERA compared with PPO and DPO.

To understand the ERA loss function and its connection to the entropy regularized objective (2), we first establish that the minimizer of (19) are of the form (5). We first define the notion of equivalence precisely.

Definition A.1 *The conditional probability measures $\pi(\cdot|\mathbf{x})$ and $\pi'(\cdot|\mathbf{x})$ in $\mathcal{P}(\mathcal{Y})$ are conditionally equivalent if $\forall \mathbf{x} \in \mathcal{X}$, π and π' are such that $\sup_{\mathbf{y} \in \mathcal{Y}} |\pi(\mathbf{y}|\mathbf{x}) - \pi'(\mathbf{y}|\mathbf{x})| = 0$.*

This is a strong form of equivalence for probability measures, but it is appropriate on the discrete spaces \mathcal{X} and \mathcal{Y} we consider here. For more general continuous spaces, one could relax this condition to weak equivalence of the conditional measures. We use this notion to emphasize that a shift of the distribution of the “prompts” $\mathbf{x} \in \mathcal{X}$, which we denote $\nu \in \mathcal{P}(\mathcal{X})$, does not impact conditional equivalence and hence establishes an equivalence class of conditional probability measures that minimize (2).

Lemma A.1 *If π is conditionally equivalent to π' , then $\pi'_g(\cdot|\mathbf{x}) \propto \pi'(\cdot|\mathbf{x})e^{g(\mathbf{x})}$ is conditionally equivalent to π for all functions $g : \mathcal{X} \rightarrow \mathbb{R}$ such that $\sup_{\mathbf{x} \in \mathcal{X}} |e^{g(\mathbf{x})}| < +\infty$.*

Assume that π' is a normalized probability distribution. This requires that,

$$Z'(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \pi'(\mathbf{y}|\mathbf{x}) = 1. \quad (14)$$

If g is such that

$$Z'_g(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \pi'(\mathbf{y}|\mathbf{x})e^{g(\mathbf{x})} \neq 1, \quad (15)$$

then the normalized policy π'_g is clearly defined by

$$\frac{1}{Z'_g(\mathbf{x})} \pi'(\mathbf{y}|\mathbf{x})e^{g(\mathbf{x})} \equiv \pi'_g(\mathbf{y}|\mathbf{x}), \quad (16)$$

because $Z'_g(\mathbf{x}) = e^{g(\mathbf{x})}$. By the assumption that $\sup_{\mathbf{x} \in \mathcal{X}} |e^{g(\mathbf{x})}| < +\infty$, all terms in these calculations remain finite.

Using Lemma A.1 it is straightforward to prove the result in Proposition 3.2. For completeness, we re-state that result here and refer the reader to Appendix 3 for the complete argument.

Proposition A.2 *Suppose $\pi(\cdot|\mathbf{x}) \in \mathcal{P}(\mathcal{Y})$ and that $\text{supp}(\pi) = \text{supp}(\pi_{\text{ref}})$. Let $\beta > 0$, $\gamma \geq 0$ and that the reward model is such that $\sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X} \times \mathcal{Y}} |e^{-U(\mathbf{x}, \mathbf{y})}| < +\infty$. Then, the minimizer of \mathcal{L}^{ERA} is conditionally equivalent to π_* .*

This proposition establishes that a policy minimizing the objective

$$\begin{aligned}\mathcal{L}^{\text{ERA}}(\pi_{\theta}) &= \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y, y' \sim \pi_{\text{ref}}(\cdot | x)} D_{\text{KL}}^{(y, y')}(p_{\beta} | p_{\theta}); \\ p_{\theta} &:= \sigma \left(\log \frac{\pi_{\theta}(y | x)}{\pi_{\theta}(y' | x)} \right) \\ p_{\gamma} &:= \sigma \left(\frac{\beta}{1 + \gamma} \left[(U(x, y') - U(x, y)) + \beta^{-1} \gamma \log \frac{\pi_{\text{ref}}(y | x)}{\pi_{\text{ref}}(y' | x)} \right] \right),\end{aligned}\tag{17}$$

has the form

$$\pi_{*}(y | x) = Z^{-1}(x) \exp \left[-\frac{\beta}{1 + \gamma} (U(x, y) - \beta^{-1} \gamma \log \pi_{\text{ref}}(y | x)) \right].\tag{18}$$

We do not, however, prove that gradient descent of θ on (17) converges to the global minimizer (18) because such an argument requires additional assumptions about the parametric class of policies and the convexity of the objective with respect to the parameters, neither of which are straightforward to establish.

A.1 Loss functions for π_{θ} :

Proximal Policy Optimization (PPO) optimizes an indirect, proximal objective to minimize an objective closely related to (2) (cf. Appendix A). Direct Preference Optimization (DPO) treats the negative reward function U implicitly and directly maximizes the likelihood of $p(y \succ y' | x)$. Our objectives differ from both approaches: like DPO, we directly optimize the policy using an explicit, gradient-based objective, but, in contrast, we use a reward function directly in our objective. The losses we build are thus amenable to both offline (samples from π_{ref}) and online (samples from π_{θ}) policy alignment, as explained below. Choosing to optimize the objective online has been shown to have important consequences on performance [Tajwar et al. \[2024\]](#), though we focus here on the setting where samples are drawn offline.

We directly optimize the Kullback-Leibler divergence between the entropy-regularized preference distribution $p_{\gamma}(y \succ y' | x)$ and the corresponding parametric preference distribution $p_{\theta}(y \succ y' | x)$. Explicitly, using the fact that conditional preference distribution is normalized, we obtain

$$\begin{aligned}D_{\text{KL}}^{(y, y')}(p_{\gamma} | p_{\theta}) &= p_{\gamma}(y \succ y' | x) \log \frac{p_{\gamma}(y \succ y' | x)}{p_{\theta}(y \succ y' | x)} + p_{\gamma}(y' \succ y | x) \log \frac{p_{\gamma}(y' \succ y | x)}{p_{\theta}(y' \succ y | x)} \\ &= p_{\gamma}(y \succ y' | x) \log \frac{p_{\gamma}(y \succ y' | x)}{p_{\theta}(y \succ y' | x)} + (1 - p_{\gamma}(y \succ y' | x)) \log \frac{1 - p_{\gamma}(y \succ y' | x)}{1 - p_{\theta}(y \succ y' | x)},\end{aligned}\tag{19}$$

where

$$p_{\gamma} := \sigma \left(\frac{\beta}{1 + \gamma} \left[(U(x, y') - U(x, y)) + \beta^{-1} \gamma \log \frac{\pi_{\text{ref}}(y | x)}{\pi_{\text{ref}}(y' | x)} \right] \right).\tag{20}$$

This quantity is a well-defined KL divergence and is hence non-negative; the quantity vanishes when $p_{\gamma} = p_{\theta}$ on the observations y, y' . Furthermore, with access to an explicit reward model, all terms in (19) can be computed directly and

$$p_{\theta}(y \succ y' | x) = \frac{\pi_{\theta}(y | x)}{\pi_{\theta}(y | x) + \pi_{\theta}(y' | x)} = \sigma \left(\log \frac{\pi_{\theta}(y | x)}{\pi_{\theta}(y' | x)} \right).\tag{21}$$

To obtain a minimizer of the regularized objective defined in (2) we optimize

$$\mathcal{L}^{\text{ERA}}(\pi_{\theta}) = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y, y' \sim \pi_{\text{ref}}(\cdot | x)} D_{\text{KL}}^{(y, y')}(p_{\gamma} | p_{\theta});\tag{22}$$

If the current policy overlaps with the target preference distribution, it may be useful to sample directly from the partially aligned policy, i.e., to use the “on-policy” formulation,

$$\mathcal{L}_{\text{on}}^{\text{ERA}}(\pi_{\theta}) = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y, y' \sim \pi_{\theta}(y | x)} D_{\text{KL}}^{(y, y')}(p_{\gamma} | p_{\theta})\tag{23}$$

instead of (9). One issue that arises with this scheme is differentiation with respect to the parameters of the policy θ because y and y' are decoded into discrete tokens, an operation that is not differentiable. To remedy this, we importance sample with a reference policy

$$\mathcal{L}_{\text{on}}^{\text{ERA}}(\pi_{\theta}) = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y, y' \sim \pi_{\text{ref}}(y | x)} \frac{\pi_{\theta}(y | x) \pi_{\theta}(y' | x)}{\pi_{\text{ref}}(y | x) \pi_{\text{ref}}(y' | x)} D_{\text{KL}}^{(y, y')}(p_{\gamma} | p_{\theta}).\tag{24}$$

This reweighting is straightforward and the importance weights should generally be appreciable, especially early in training when π_θ has not drifted far from π_{ref} . It is, of course, also natural to iteratively update π_θ using a previous iterate as the reference policy. In this work, we only use (9) as an objective and leave the on-policy objectives to future work.

A.2 Gradients of \mathcal{L}^{ERA} .

One advantage of the ERA framework is that the objective is amenable to direct, gradient-based optimization. We remark that establishing global convergence for the optimization of θ using (9) requires establishing convexity with respect to the parameters, which is not obviously the case for our objective, nor those used in PPO and DPO. However, one can still glean some insight into the optimization by examining the gradients on a samplewise basis. Using the compact notation $p_\theta(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}) \equiv \sigma_\theta$ and $p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}) \equiv \sigma_*$,

$$\nabla_\theta \mathcal{L}^{\text{ERA}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{y}, \mathbf{y}' \sim \pi_{\text{ref}}} \left(\frac{1 - \sigma_*}{1 - \sigma_\theta} - \frac{\sigma_*}{\sigma_\theta} \right) \nabla_\theta \sigma_\theta. \quad (25)$$

The gradient is straightforward to interpret on a particular pair \mathbf{y}, \mathbf{y}' : if $p_\theta(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$ is larger than $p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$ then the preference gradient is positive and gradient descent lowers the probability that $\mathbf{y} \succ \mathbf{y}'$. The opposite occurs whenever $p_\theta(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$ is smaller than $p_\gamma(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$. The magnitude of the gradient is scaled by the degree of misspecification of the preference probability.

This calculation highlights one key difference between the approach we use and DPO. When the data only contains one observation of $\mathbf{y} \succ \mathbf{y}'$ for a given \mathbf{x} , the DPO objective's implicit reward model assigns zero probability to $\mathbf{y}' \succ \mathbf{y}$. This pushes the policy towards extremal values, which can lead to undesired behavior, as discussed in [Azar et al. \[2023\]](#). In our formulation, this behavior occurs only when the reward model assigns an energy of $\pm\infty$, which is prohibited by construction in most tasks. We further discuss differences between ERA and DPO in Appendix [A.4](#).

A.3 Comparison with PPO Objective

The free energy functional for a policy under the energy rank alignment framework can be written as an expectation

$$J_{\text{ERA}}[\pi] = \mathbb{E}_{\mathbf{x} \sim \nu} \left[\int U(\mathbf{x}, \mathbf{y}) d\pi(\mathbf{y}|\mathbf{x}) + \beta^{-1} \int (1 + \gamma) \log \pi(\mathbf{y}|\mathbf{x}) - \gamma \log(\pi_{\text{ref}}(\mathbf{y}|\mathbf{x}) d\pi(\mathbf{y}|\mathbf{x}) \right], \quad (26)$$

involving an energetic term and an entropic term. The additional regularization acts as an effective energetic bias. Solving for the extremum of this functional by setting Fréchet derivative with respect to π equal to zero, one obtains the formal solution (18) for the minimizer. This objective differs from the regularized reward loss conventionally used for PPO,

$$\begin{aligned} J_{\text{PPO}}(\pi) &= \mathbb{E}_{\mathbf{x}} \left[\int U(\mathbf{x}, \mathbf{y}) d\pi(\mathbf{y}|\mathbf{x}) + \gamma \beta^{-1} \int \log \frac{\pi(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})} d\pi(\mathbf{y}|\mathbf{x}) \right], \\ &= \mathbb{E}_{\mathbf{x}} \left[\int U(\mathbf{x}, \mathbf{y}) d\pi(\mathbf{y}|\mathbf{x}) + \gamma \beta^{-1} D_{\text{KL}}(\pi(\cdot|\mathbf{x}) \|\pi_{\text{ref}}(\cdot|\mathbf{x})) \right]. \end{aligned} \quad (27)$$

The minimizer of the PPO objective (27) is also a Gibbs-Boltzmann measure, explicitly,

$$\pi_*^{(\text{PPO})} \propto \exp \left[-\frac{\beta}{\gamma} U(\mathbf{x}, \mathbf{y}) + \log \pi_{\text{ref}}(\mathbf{y}|\mathbf{x}) \right]. \quad (28)$$

Here, the KL-regularization corresponds to an energy shift, as in our objective, but there is no limit in which the ideal distribution $\pi \propto e^{-\beta U}$ is obtained for the PPO objective. This is in stark contrast to our approach, which recovers the ideal distribution as $\gamma \rightarrow 0$. Furthermore, while our approach allows for a direct gradient-based optimization using (17), PPO is implemented using an actor-critic framework that is difficult to tune [Rafailov et al. \[2023\]](#), [Casper et al. \[2023\]](#). Finally, we emphasize that for ERA in the $\gamma \rightarrow 0$, finite $\beta > 0$, the distribution has positive entropy and is not manifestly mode-seeking; there can still be appreciable fluctuations in the output. Eliminating the effect of regularization in (28), on the other hand, requires taking $\beta/\gamma \rightarrow \infty$, which eliminates fluctuations in the distribution.

A.4 Comparison with DPO Objective

The DPO approach also seeks to optimize the objective (27). The algorithm does so by first using (28) to define an implicit reward model by solving for the U that reflects the observed preference probabilities. This elegant idea has had a significant impact and has already been deployed in state-of-the-art models AI@Meta [2024]. In many cases, the observed preference probabilities will be sampled and only perhaps only one observation of $\mathbf{y} \succ \mathbf{y}'$ will be available for each \mathbf{x} in the dataset. When the preference dataset only has one observation $\mathbf{y} \succ \mathbf{y}'$ per prompt \mathbf{x} , the optimal policy requires that

$$\pi_{\star}^{\text{DPO}}(\mathbf{y}|\mathbf{x}) = 1 \quad \text{and} \quad \pi_{\star}^{\text{DPO}}(\mathbf{y}'|\mathbf{x}) = 0. \quad (29)$$

The sampled gradients of the objective used for DPO are proportional to the implicit reward discrepancy,

$$\nabla_{\theta} \hat{\mathcal{L}}^{\text{DPO}}(\mathbf{y}, \mathbf{y}', \mathbf{x}) = \sigma \left(\beta^{-1} \gamma \left[\log \frac{\pi_{\theta}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})} - \log \frac{\pi_{\theta}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})} \right] \right) \nabla_{\theta} \log \frac{\pi_{\theta}(\mathbf{y}|\mathbf{x})}{\pi_{\theta}(\mathbf{y}'|\mathbf{x})}, \quad (30)$$

which when $\pi_{\theta}(\mathbf{y}'|\mathbf{x}) \rightarrow 0$, could lead to instability as $-\log \pi_{\theta}(\mathbf{y}'|\mathbf{x}) \rightarrow \infty$. On the other hand, the ERA gradients are scaled by the relative preference discrepancy,

$$\nabla_{\theta} \mathcal{L}^{\text{ERA}}(\mathbf{y}, \mathbf{y}', \mathbf{x}) = \left(\frac{1 - \sigma_{\star}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})}{1 - \sigma_{\theta}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})} - \frac{\sigma_{\star}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})}{\sigma_{\theta}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})} \right) \nabla_{\theta} \sigma_{\theta}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}). \quad (31)$$

The advantage of a reward model becomes apparent because

$$\sigma_{\star}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}) = p_{\gamma}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}) = \sigma \left(\frac{\beta}{1 + \gamma} \left[(U(\mathbf{x}, \mathbf{y}') - U(\mathbf{x}, \mathbf{y})) + \beta^{-1} \gamma \log \frac{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})} \right] \right) \quad (32)$$

and hence the optimum of \mathcal{L}^{ERA} will not lead to policies in which $\text{supp}(\pi_{\theta})$ degrades unless the energy becomes infinite. Choosing an appropriate reward model, hence, gives the flexibility to control instability if it becomes problematic.

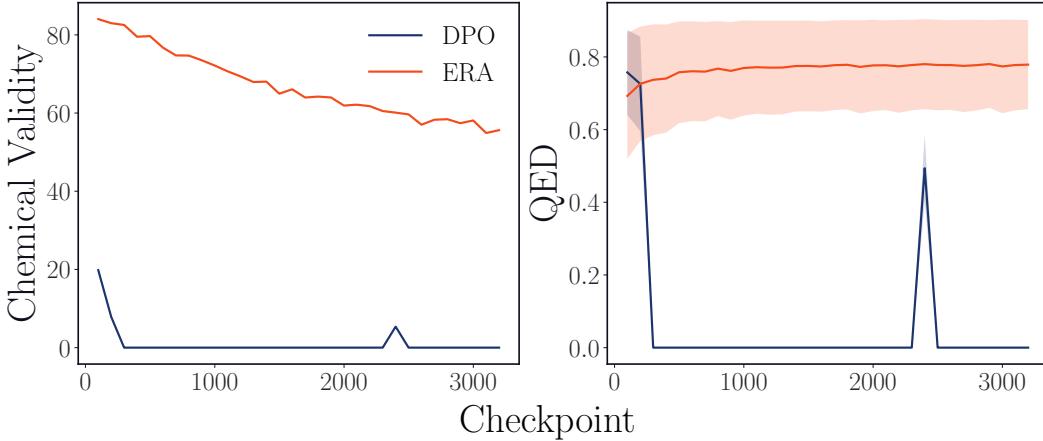


Figure S6: Comparison of DPO ($\beta_{\text{DPO}}=0.1$) and ERA ($\beta = 20.0$ and $\gamma = 0.0$) on the task of maximizing QED (Section 4.1.1) a) Chemical validity degrades significantly for DPO and slightly degrades for ERA as a function of checkpoints. b) At each checkpoint, 20k molecules are sampled and mean QED of valid molecules plotted with shading corresponding to 1 standard deviation. Both DPO and ERA alignment runs were trained with the same hyperparameters and dataset, with checkpoints saved every 100 epochs.

We carry out an online evaluation of DPO and ERA on the task of generating small-molecules with high QED, as in Figure 2. We align DPO using the same dataset and hyperparameters as ERA with $\beta_{\text{DPO}} = 0.1$ and train for thousands of checkpoints (over 72 GPU-hours). We load intermediate checkpoints for both the DPO and ERA ($\beta_{\text{ERA}} = 20.0$, $\gamma_{\text{ERA}} = 0.0$) runs and carry out inference (Figure S6). We observe that at the first saved checkpoint of the DPO alignment run, the model generates molecules with high QED scores but with low validity ($\sim 20\%$). However, upon further

training, the chemical validity of further checkpoints drops to 0% for the remaining runs, despite the overall DPO training and validation losses still dropping.

With ERA, we see that we are able to similarly sample high QED small-molecules with reasonably high chemical validity ($\sim 85\%$). While the validity does drop over subsequent checkpoints, it does not do so precipitously. Moreover, the ERA-based alignment had no regularization ($\gamma = 0$), and in the main text, we document how having non-zero γ can enable increases in chemical validity (Figure S10). Finally, we note that we did not extensively tune the hyperparameters for DPO, and it is possible that a different set of hyperparameters would elicit a more desired outcome; however, the lack of meaningful regularization in DPO [Azar et al. \[2023\]](#) and its performance degradation in online metrics has been well-documented [Chen et al. \[2024\]](#).

B ERA implementation

Implementing energy rank alignment is straightforward to implement within existing code bases. We provide sample PyTorch code for the ERA loss function below.

```

import torch.nn as nn
from torch.nn.functional import logsigmoid

def era_loss(pi_logps_1, pi_logps_2,
            ref_logps_1, ref_logps_2,
            energies_1, energies_2,
            beta, gamma):
    """
    pi_logps_1: logprob under policy's model of first sequence in pair (B, )
    pi_logps_2: logprob under policy's model of second sequence in pair (B, )
    ref_logps_1: logprob under reference model of first sequence in pair (B, )
    ref_logps_2: logprob under reference model of second sequence in pair (B, )
    energies_1: energies of first sequence in pair (B, )
    energies_2: energies of second sequence in pair (B, )
    beta: inverse temperature
    gamma: regularization controlling strength of KL penalty
    """
    beta_prime = (beta / (1 + gamma))
    gamma_prime = (gamma / (1 + gamma))

    logp = logsigmoid(policy_logps_y2 - policy_logps_y1)
    logp_prime = logsigmoid(policy_logps_y1 - policy_logps_y2)

    logp_star = logsigmoid(-beta_prime * (energies_y2 - energies_y1)
                           + gamma_prime * (ref_logps_y2 - ref_logps_y1))
    logp_star_prime = logsigmoid(-beta_prime * (energies_y1 - energies_y2)
                                 + gamma_prime * (ref_logps_y1 - ref_logps_y2))

    era_loss = (torch.exp(logp_star) * (logp_star - logp)
                + torch.exp(logp_star_prime) * (logp_star_prime - logp_prime))

    return era_loss.mean()

```

C Details for molecular generator experiments

C.1 Pretraining details

In this work, we represent all molecules as SMILES strings and tokenize SMILES strings according to the approach in [Schwaller et al. \[2018\]](#). Our dataset consisted of all small-molecules from the ChEMBL database that were of length 500 tokens or less. Ultimately, this token limit filtered out approximately 0.1% of the small-molecules in the original ChEMBL dataset. The alphabet generated from this curated dataset consists of 324 tokens, which we augmented with start, stop, and padding tokens.

We first pretrained a model according to a next-token prediction, self-supervised learning approach. We trained a model using the standard cross entropy loss

$$\mathcal{L}_{\text{CE}} = - \sum_{t=1}^T \log p_{\theta}(\mathbf{x}_{t+1} | \mathbf{x}_{1:t}). \quad (33)$$

Our trained molecular generator consisted of just the encoder block of a standard multi-head attention transformer [Vaswani et al. \[2017\]](#). Finally, the model had 2 layers, 8 heads, and a width of 512. For pretraining, we used an Adam optimizer with a learning rate of $1.0 * 10^{-5}$. We emphasize that this pretrained generator samples molecules in an unprompted fashion; given just a start-of-sequence token, we can autoregressively generate a sequence of tokens. Moreover, it is possible that this sequence of tokens corresponds to a molecule that is not chemically valid, and we find that around 88% of all generated molecules are chemically valid. Lastly, we measure the diversity of the pretrained molecular generator by first generating 1500 molecules and then computing the Tanimoto similarity between every pair of molecules. We plot the distribution of all pairwise Tanimoto similarities from

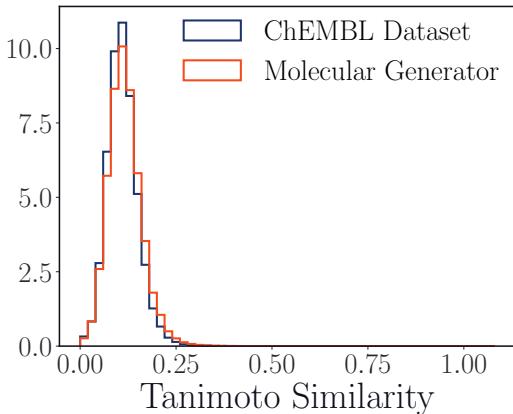


Figure S7: Chemical diversity of samples from training dataset and from unprompted molecular generator (unaligned) as measured by pairwise Tanimoto similarities. Lower Tanimoto similarities correspond to more chemically dissimilar molecules.

Property name (f)	Energy function (U)
Tanimoto similarity	$U = -\log(f(\mathbf{y}))$
QED	$U = -\log(f(\mathbf{y}))$
Docking Oracles (GSK3 β and JNK3)	$U = -\log(f(\mathbf{y}))$
Wildman-Crippen LogP	$U = (f(\mathbf{y}) - \mu)/2\sigma^2$
Molar refractivity	$U = (f(\mathbf{y}) - \mu)/2\sigma^2$
Ring count	$U = (f(\mathbf{y}) - \mu)/2\sigma^2$

Table S2: Definitions of energy functions (in reduced units) used for each of the five chemical properties investigated in this work. Here \mathbf{y} refers to the generated molecule.

this sample and from all pairwise Tanimoto similarities from 1500 randomly sampled molecules from the original dataset in Fig. S7. We observe that we can generate molecules that are quite distinct (i.e. low Tanimoto similarity) in comparison with all other molecules.

C.2 Chemical properties

We investigated aligning the molecule generator to several target chemical properties, which we detail below. All of the properties can be easily computed using either the RDKit package or the tdc [Huang et al. \[2021\]](#) package. We list the energy function and parameters used for the corresponding energy functions for each of these properties in Table S2.

Tanimoto similarity is a measure of chemical and structural properties between two molecules and ranges from 0 to 1, where higher values correspond to more similar molecules [Rogers and Tanimoto \[1960\]](#). Quantitative estimation of drug-likeness (QED) is evaluated by taking the geometric mean of a set of “desirability functions” for different molecular descriptors and also ranges continuously from values of 0 to 1 [Bickerton et al. \[2012\]](#), where higher values correspond to more drug-like molecules. The octanol-water partition coefficient (Wildman-Crippen LogP) is a measure of hydrophobicity frequently employed in medicinal chemistry applications [Wildman and Crippen \[1999\]](#). Molecules with more positive values are more hydrophobic (i.e. more soluble in octanol relative to water), whereas molecules with more negative values are more hydrophilic (i.e. more soluble in water relative to octanol). Molar refractivity is similarly calculated as a linear combination of atomic contributions, and is a positive number that serves as a measure for molecular size and polarizability [Wildman and Crippen \[1999\]](#). A higher molar refractivity corresponds to larger and more polarizable molecules. Finally, ring count corresponds to the number of rings in a molecule.

Under the definitions of the energy functions in Table S2, it is possible for a generated sequence to not be chemically valid. For these cases, we manually define energies that are sufficiently high to penalize that outcome and we report these values in Table S3. Furthermore, when the computed

Property name (f)	Energy
Tanimoto similarity	10
QED	4.5
Docking Oracles (GSK3 β and JNK3)	4.5
Wildman-Crippen LogP	300
Molar refractivity	400
Ring count	70

Table S3: Property-specific energy values (in reduced units) used to treat chemically invalid sequences.

QED or Tanimoto Similarity is 0, the energy is infinite, and to ensure numerical stability, we set the value of the energies to be 4.5 and 10 respectively. Finally, in the prompted molecular generator experiments in Section 4.1.2, we assign an energy of 3.5 to the setting where Tanimoto similarity between the generated and prompt molecule is 1.0 (i.e they are the same) in order to penalize this outcome. Here, all energy and β values are reported in reduced units.

C.3 Molecular alignment details

C.3.1 ERA ablations and comparison to DPO

β	γ	Validity \uparrow	Top 100 Mean \uparrow	Top 100 Diversity \downarrow
10	0.01	72.00	0.932	0.129
10	0.1	74.23	0.933	0.131
100	0.01	74.71	0.934	0.129
100	0.1	75.14	0.935	0.131

Table S4: Descriptive statistics of 10,000 generated SMILES after aligning the molecular transformer model with ERA for QED maximization across several values of β and γ .

DPO results			
β_{DPO}	Validity (%) \uparrow	Top-100 mean \uparrow	Top-100 diversity \downarrow
100	86.91	0.936	0.131
10	80.92	0.943	0.134
1	82.23	0.943	0.156
0.1	54.46	0.947	0.174
ERA results ($\gamma = 9$)			
β_{ERA}	Validity (%) \uparrow	Top-100 mean \uparrow	Top-100 diversity \downarrow
0.1	84.78	0.939	0.149
1	85.73	0.936	0.145
10	79.07	0.932	0.141
100	88.83	0.941	0.145

Table S5: Comparison of DPO and ERA results across different β values on the task of QED maximization.

To evaluate the effects of the parameters β and γ on the performance of ERA, we assess the algorithm’s performance on the task of QED maximization by ablating β and γ in Table S4. We observe that higher β leads to greater dominance of the reward and increasing values for the Top-100 mean QED whereas higher γ leads to higher validity due to the increased regularization against the reference policy, both consistent with the minimizer presented in 5.

To assess ERA against DPO, we evaluate each method on their performance on the task of QED maximization. We note that the DPO objective does not contain an explicit reward component and instead only maximizes the margins between preferred and dis-preferred samples. Additional care needs to be taken when comparing DPO and ERA hyperparameters as $\beta_{ERA} \neq \beta_{DPO}$ due to a

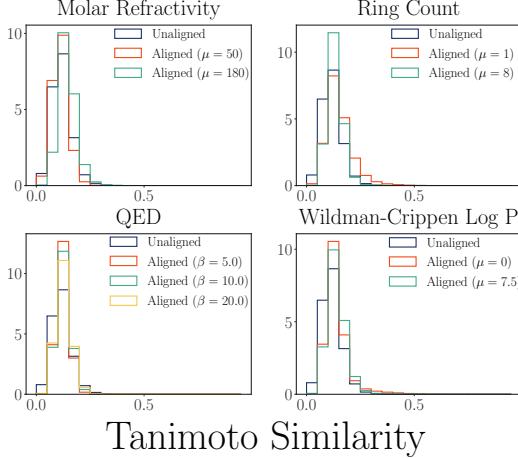


Figure S8: Chemical diversity of samples from unprompted molecular generator after alignment as measured by pairwise Tanimoto similarities. (See Fig. 2, Section 4.1.1)

difference in definition. Specifically, the minimizer of the DPO objective is:

$$\pi_{DPO}(x, y) \propto \exp(-\beta_{DPO}^{-1} r(x, y) + \log \pi_{ref})$$

Whereas for ERA, the minimizer is

$$\pi_{ERA}(x, y) \propto \exp\left(\frac{-\beta_{ERA}}{1 + \gamma} r(x, y) + \frac{\gamma}{1 + \gamma} \log \pi_{ref}\right)$$

The formulation of the DPO objective does not permit explicitly scaling the contribution of the regularization term against the prior while ERA permits this. This also implies that

$$\beta_{ERA} = \frac{1 + \gamma}{\beta_{DPO}}$$

Furthermore, we take γ to be relatively large since as $\gamma \rightarrow \infty$, $\frac{\gamma}{1 + \gamma} \rightarrow 1$. In our comparisons, we choose $\gamma = 9$. Table S5 contains descriptive statistics of 10,000 generated SMILES after aligning the molecular transformer model for QED maximization using either DPO or ERA across several matched values of β . It is arranged such that the corresponding rows can be compared directly against each other. We did not scan any $\beta < 0.1$ since the validity in those cases was low for both methods. Due to parameter matching, we expect ERA and DPO to be similar here: validity is comparable, with ERA surpassing DPO at higher β_{ERA} , as well as the top-100 average QED. ERA consistently generates diverse molecules across values of β_{ERA} as opposed to DPO, as evidenced by the lower similarity among the top-100 generations for $\beta_{ERA} = 10$ and $\beta_{ERA} = 100$. This again reflects the fact that the entropy-regulated ERA objective promotes diverse generations while effectively avoiding greedy policies as opposed to DPO, which focuses only on maximizing the margins between preferred and dispreferred sequences. This highlights a key advantage of ERA since good molecular diversity is a key consideration for chemical and drug discovery tasks.

C.3.2 Unprompted molecular generation (RDKit oracles)

We first investigated aligning the unprompted molecular generator to sample small-molecules with desired properties. We carried out alignment using the property-specific energies described in Table S2. All alignment properties were initialized with the weights of the pretrained model and trained using an Adam optimizer with learning rate $1.0 * 10^{-6}$. We tabulate the chemical validity for single-property alignment in Table S6 and for multi-property alignment in Table S7. While we do see a drop in chemical validity after alignment, we see that a majority of the samples we generate post-alignment are still chemically valid despite no regularization to a reference policy. We measure the chemical diversity for these experiments by computing all pairwise Tanimoto similarities from all chemically valid predictions of 1500 generated molecules. We visualize the chemical diversity for single-property experiments in Fig. S8 and multi-property experiments in Fig. S9. We observe that the samples are still highly diverse chemically after alignment. All plots in Fig. 2 and Fig. 3 were computed using 1500 generated molecules per experiment.

Property name	Hyperparameters	Chemical validity
Unaligned	N/A	88%
Molar Refractivity	$\beta = 1.0, \mu = 50, \sigma = 10, \gamma = 0.0$	82%
Molar Refractivity	$\beta = 1.0, \mu = 180, \sigma = 10, \gamma = 0.0$	74%
Ring Count	$\beta = 1.0, \mu = 1, \sigma = 1.0, \gamma = 0.0$	84%
Ring Count	$\beta = 1.0, \mu = 8, \sigma = 1.0, \gamma = 0.0$	59%
LogP	$\beta = 10.0, \mu = 2.5, \sigma = 1.0, \gamma = 0.0$	74%
LogP	$\beta = 10.0, \mu = 7.5, \sigma = 1.0, \gamma = 0.0$	63%
QED	$\beta = 5.0, \gamma = 0.0$	54%
QED	$\beta = 10.0, \gamma = 0.0$	66%
QED	$\beta = 20.0, \gamma = 0.0$	65%

Table S6: Percentage of generated sequences that were chemically valid for samples from unprompted molecular generator after alignment. (See Fig. 2, Section 4.1.1).

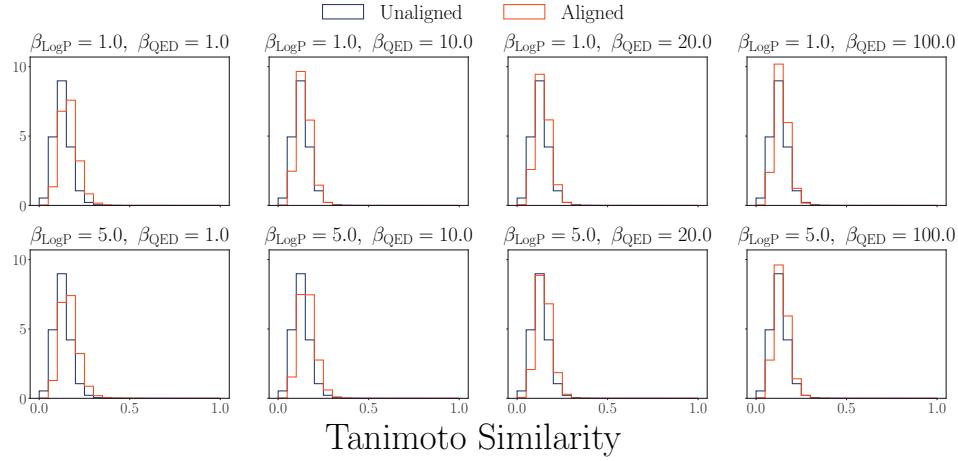


Figure S9: Chemical diversity of samples from unprompted molecular generator after multi-property alignment as measured by pairwise Tanimoto similarities. (See Fig. 3, Section 4.1.1).

Hyperparameters	Chemical validity
Unaligned	88%
$\beta_{QED} = 1.0, \beta_{LogP} = 1.0, \mu_{LogP} = 7.5, \sigma_{LogP} = 1.0, \gamma = 0.0$	60%
$\beta_{QED} = 1.0, \beta_{LogP} = 10.0, \mu_{LogP} = 7.5, \sigma_{LogP} = 1.0, \gamma = 0.0$	67%
$\beta_{QED} = 1.0, \beta_{LogP} = 20.0, \mu_{LogP} = 7.5, \sigma_{LogP} = 1.0, \gamma = 0.0$	68%
$\beta_{QED} = 1.0, \beta_{LogP} = 100.0, \mu_{LogP} = 7.5, \sigma_{LogP} = 1.0, \gamma = 0.0$	63%
$\beta_{QED} = 5.0, \beta_{LogP} = 1.0, \mu_{LogP} = 7.5, \sigma_{LogP} = 1.0, \gamma = 0.0$	64%
$\beta_{QED} = 5.0, \beta_{LogP} = 10.0, \mu_{LogP} = 7.5, \sigma_{LogP} = 1.0, \gamma = 0.0$	62%
$\beta_{QED} = 5.0, \beta_{LogP} = 20.0, \mu_{LogP} = 7.5, \sigma_{LogP} = 1.0, \gamma = 0.0$	62%
$\beta_{QED} = 5.0, \beta_{LogP} = 100.0, \mu_{LogP} = 7.5, \sigma_{LogP} = 1.0, \gamma = 0.0$	68%

Table S7: Percentage of generated sequences that were chemically valid for samples from unprompted molecular generator after multi-property alignment. (See Fig. 3, Section 4.1.1).

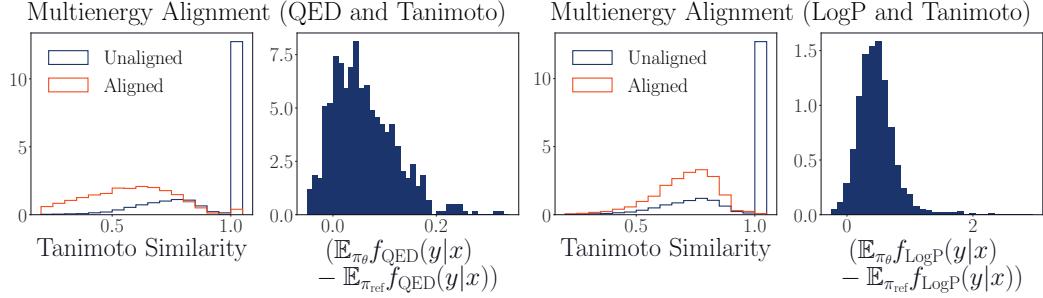


Figure S10: Prompted multi-property molecular generator alignment. From left to right: Tanimoto similarities computed between the prompt and sampled molecules for both aligned and unaligned policies (QED and Tanimoto alignment), per-prompt difference in the average QED under aligned and unaligned policies (QED and Tanimoto alignment), Tanimoto similarities computed between the prompt and sampled molecules for both aligned and unaligned policies (LogP and Tanimoto alignment), and per-prompt difference in the average LogP under aligned and unaligned policies (LogP and Tanimoto alignment). With alignment, we target higher QED and LogP values, while still sampling molecules chemically similar—but not identical to—the prompt molecule.

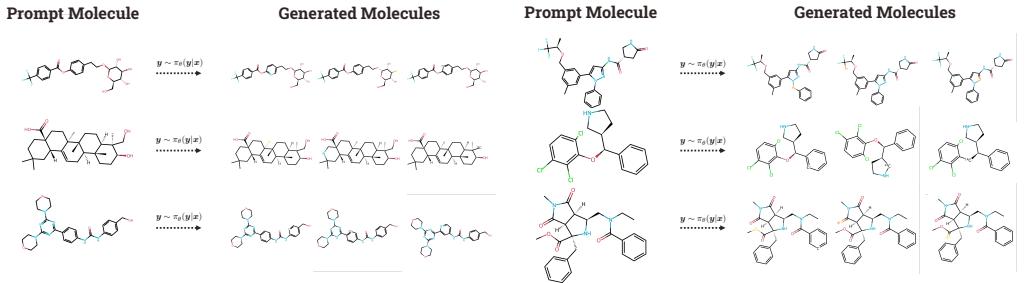


Figure S11: Sample molecules from prompted molecular generator after multi-property alignment experiments: QED and Tanimoto (left) and LogP and Tanimoto (right). With alignment, generated molecules are diverse, while still chemically similar to prompt molecule.

C.3.3 Prompted molecular generation (RDKit oracles)

Next, we generate small-molecules with desired properties conditioned on a prompt, where the prompt is itself another molecule. In the experiments here, we consider the setting where we generate molecules that are chemically similar to the prompt molecule. With this in mind, we first carry out a fine-tuning step using a synthetic dataset $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}_{i=1}^N$, where \mathbf{x} corresponds to the SMILES string of a prompt molecule and \mathbf{y} corresponds to the SMILES string of the conditionally generated molecule. To curate this dataset, we consider all molecules in our original filtered ChEMBL dataset to be a prompt molecules and for each prompt molecule \mathbf{x}_i , we generate a response molecule \mathbf{y}_i by simply perturbing a random token from \mathbf{x}_i . If the perturbed sequence was chemically invalid, we repeated the random perturbation until a valid molecule was generated. The prompted generator was the same size as the unprompted molecular generator, and we initialized the weights using those of the pre-trained unprompted molecular generator. We then carried out supervised fine-tuning using an Adam optimizer with learning rate $1.0 * 10^{-5}$ and used this generator as our reference policy for all prompted alignment experiments. All plots in Fig. S10 were computed using 100 generated

Hyperparameters	Chemical validity
Unaligned	93%
$\beta_{\text{Tanimoto}} = 5.0, \beta_{\text{LogP}} = 10.0, \mu_{\text{LogP}} = 5.0, \sigma_{\text{LogP}} = 1.0, \gamma = 0.1$	91%
$\beta_{\text{Tanimoto}} = 5.0, \beta_{\text{QED}} = 500.0, \gamma = 0.1$	81%

Table S8: Percentage of generated sequences that were chemically valid for samples from prompted molecular generator after multi-property alignment. (See Fig. S10, Section 4.1.2).

molecules per prompt, where we carried inference over 500 prompts per experiment. We tabulate the chemical validity of the prompted generator in Table S8.

C.3.4 Unprompted molecular generation (protein-ligand docking oracles)

	GSK3 β top-10 AUC	JNK3 top-10 AUC
ERA	0.985 \pm 0.001	0.989 \pm 0.002
REINVENT	0.865 \pm 0.043	0.783 \pm 0.023
GraphGA	0.788 \pm 0.070	0.553 \pm 0.136
PPO	0.90 \pm 0.02	0.80 \pm 0.04
PPOD	0.92 \pm 0.02	0.87 \pm 0.02

Table S9: Top-10 AUC scores on GSK3 β and JNK3 tasks averaged across 5 random seeds. Compared to state-of-the-art methods as reported in Gao et al. [2022] and Bou et al. [2024], ERA has higher sampling efficiency. Results for compared methods are reproduced from Gao et al. [2022] and Bou et al. [2024]

For the work here, we used a computational oracle that predicts the docking score for two kinases, JNK3 and GSK3 β , where these oracles were defined using the `tdc` package. We first carried out a supervised fine-tuning step using all molecules in ChemBL with an oracle score above 0.5. For the JNK3 model, we carried out fine-tuning on a dataset of size 7386 molecules, and for GSK3 β , we carried out fine-tuning on a dataset of size 43381 using the Adam optimizer with a learning rate of $1.0 * 10^{-5}$. From these fine-tuned models, we sampled 40000 molecules, evaluated the oracle scores, and used this dataset to carry out alignment using all possible pairs of molecules. For any invalid molecule, we assign an energy of 4.5. All alignment runs were done using the Adam optimizer with a learning rate of $1.0 * 10^{-6}$. We observed a sampled validity of 74% on the model aligned for GSK3 β and a sample validity of 93.6% for the model aligned for JNK3.

We compute metrics on the top-100 novel and unique molecules on 20000 sampled molecules (see Figure S12) from the aligned models. We compute the mean score and the internal diversity score (IntDiv) Hu et al. [2023] computed according to the following

$$\text{IntDiv}(A) := \frac{1}{|A|(|A| - 1)} \sum_{(x,y) \in A \times A, x \neq y} T(x, y), \quad (34)$$

where A is a set of compounds and T represents the Tanimoto similarity. Lower IntDiv scores correspond to more diverse molecules.

We additionally compute sample efficiency using the top-10 AUC metric (see Table S9), which is the area under the curve (AUC) of the mean property value of the top-10 performing molecules versus the number of oracle calls (see Figure S13). We note that once we reach a top-10 average of 1.0, we do not make further oracle calls as subsequent oracle calls will not change the top-10 average and will artificially inflate the AUC.

C.3.5 Analysis of invalid generations

Especially when the regularization $\gamma = 0$, we see that ERA reduces validity of the generated SMILES. There are some limitations imposed by the fact that most of our downstream characterizations cannot be run on invalid SMILES string because of the failure of RDKit to parse. To assess the failure modes of these invalid generations, we computed two measures of the diversity: an estimate of the Levenshtein distance and an estimate of the Shannon entropy for both valid and invalid molecules.

Table S10 demonstrates properties of the valid and invalid generations from a model aligned using ERA for QED maximization for $\beta = 100.0$, $\gamma = 0.1$. Notably, the Shannon entropy of generated tokens and the mean Levenshtein distance between all of the generated SMILES is higher for the invalid outputs than the valid ones. These metrics suggest that there is in fact less clustering around one specific region of chemical space in these failure cases.

For the same QED maximization run ($\beta = 100.0$, $\gamma = 0.1$) as above, we found a significant failure mode to be that of missing tokens to close branches (67.6% of all invalid generations). Furthermore, failure to properly close rings also occurs in a significant amount of invalid generations (30.5%). One potential failure mode that seems to be relatively insignificant is that of failing to predict the stop token

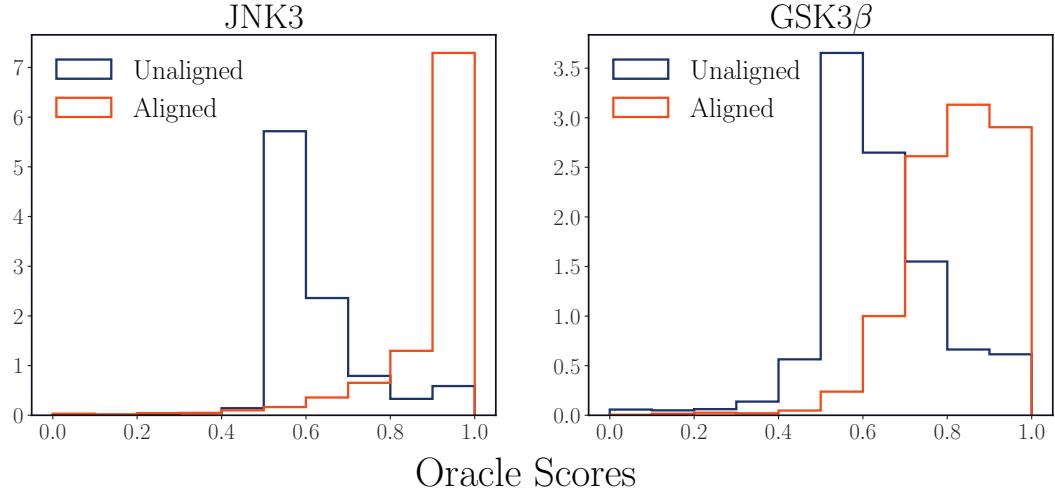


Figure S12: Distribution of GSK3 β and JNK3 oracle scores sampled from unaligned reference model and aligned model ($\beta = 100.0$, $\gamma = 0.0$). 20k molecules were sampled from each model and only oracle scores of valid molecules are plotted.

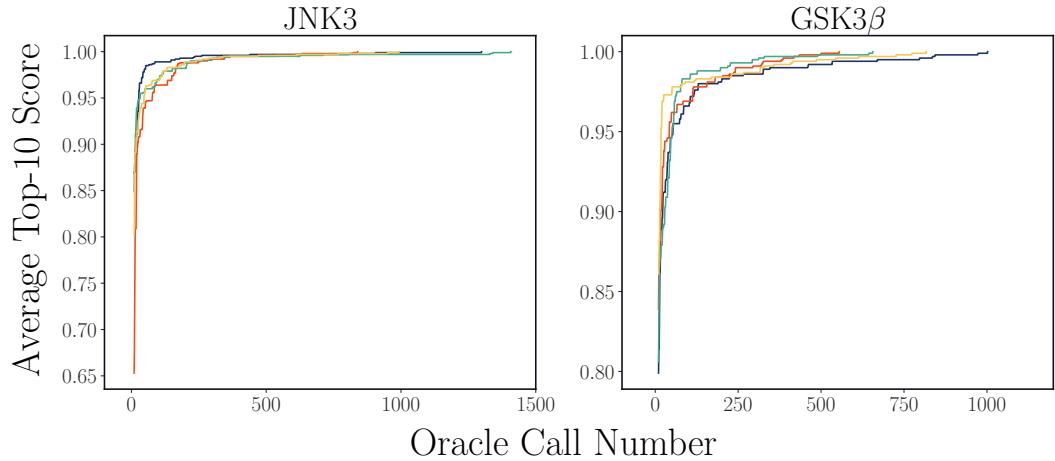


Figure S13: The average score of top-10 performing *valid*, *novel*, and *unique* molecules as a function of the number of oracle calls made to the aligned models. Scores are computed using the JNK3 and GSK3 β oracles, respectively, for five different random seeds. Samples that are invalid, present in the dataset, or already previously sampled are discarded and do not count towards an oracle call.

before our hard limit of 500 tokens, which occurs in fewer than 1% of invalid generations. Beyond this analysis, visual inspection of several failure cases does not appear to reveal any particular motif that is repeated among these examples beyond the model attempting to generate highly branched molecules or molecules with several rings.

Generations	Shannon Entropy	Levenshtein Distance
Valid	2.2574	39.8225
Invalid	2.3177	56.1066

Table S10: Shannon entropy and Levenshtein distances for valid and invalid generations from a QED maximization molecular generator alignment experiment with $\beta = 100.0$, $\gamma = 0.1$.

D Details for protein language model experiments

We consider mutating the TrpB protein at 4 sites (positions 182, 183, 184, and 186) to all of the 20 standard amino acids and compute the EVMutation score [Hopf et al. \[2017\]](#) for all $20^4 = 160000$ sequences (see [Yang et al. \[2023\]](#) for dataset). We used the standard tokenization scheme in ESM3 and during training and inference masked all non-sequence tasks [Hayes et al. \[2024\]](#). ESM3 is a bidirectional transformer and so to generate mutant sequences, we first pass in the full sequence with the mutant sites masked and use the model to “unmask” these four sites simultaneously. For the sequence track, ESM3 also contains additional tokens representing non-standard residues and special cases, and we consider sequences “invalid” if the generated token does not correspond to one of the 20 standard amino acid sequences.

When sampling mutant sequences, the native ESM3-1.4B model does not generate a diverse set of sequences and so we first synthesize an initial dataset of 512 random mutant sequences as would normally be done in a random mutagenesis experiment. We did not carry out any supervised fine-tuning step and here considered the energies to be the negative of the EVMutation score and evaluated log probabilities on the pretrained ESM3-1.4B model. We then carried alignment using ERA, where our dataset consisted of all possible unique pairs of these 512 sequences. For the experiments here, we used the RMSProp optimizer with a learning rate of $1.0 * 10^{-5}$. We plot the EVMutation score of 512 generated sequences across various β values and $\gamma = 0.001$ in Figure 5. Finally, we note that we did not sample any “invalid” sequences as defined above.

E Computational resources

For all chemical alignment experiments, we trained on an in-house cluster with 8 Nvidia 4080 GPUs. For ESM3 experiments, we used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility. Jobs run on NERSC used at most 4 Nvidia A100 GPUs (either 40GB or 80GB depending on what was allocated).

F Societal and broader impacts

The ERA algorithm we have introduced in this work is a powerful and scalable approach towards generating outputs targeting some desired combination of properties. In this work we have demonstrated the efficacy of this method in both a chemical context and a language context. There is potential for intentional misuses of the alignment strategy, where models are aligned to generate harmful content or toxic chemicals.