Autonomous Weighted Schema Direct Preference Optimization for Materials Discovery

Anonymous authors
Paper under double-blind review

Abstract

Generative artificial intelligence can create candidate molecules and materials for tasks in the physical and biological sciences, but it can be limited in producing better candidates along characterized properties than those of the systems in the training data. Reinforcement learning, in particular Direct Preference Optimization (DPO) on chemical language models (CLMs), can shift output chemical distributions in desirable ways, but constructing preference datasets in an autonomous manner presents further challenges. Here, we outline a novel methodology for multi-objective optimization where a defined target region of chemical properties is compared to generated distributions and used for the updating of preference weights that partition generated molecules into a cohesive, multi-object preference dataset. In this process, termed autonomous weighted schema (AWS) DPO, we observe enhancements to generated distributions beyond that observed without autonomous weight updating as well as reward functions that put the multi-objective frontier into a single score, and we thereby present a general algorithm for tuning generative models for materials discovery.

1 Introduction

Applications ranging from drug discovery (?) to battery technologies (?) benefit from the discovery of novel molecules and materials, motivating the general task of molecular/materials discovery. Even within the realm of small molecules, the potential search space is vast, exemplified by databases of extant molecules such as ZINC containing billions of molecules (?), a massive scope presumably dwarfed by the number of potential molecules yet to be synthesized. To navigate this chemical space, machine learning (ML) and artificial intelligence (AI) methods present opportunities.

We focus on the paradigm of the chemical language model (CLM), an AI capability wherein molecular or material composition is encoded into tokenized text used to train a custom language model. Generative transformers present a common architecture for CLMs wherein structures are generated in an autoregressive fashion, building out chemical systems piece-by-piece.

The archetypal full training pipeline for such CLMs analogizes to that of more general large language models (LLMs), i.e., consisting of a pretraining (PT) stage of massive datasets, then supervised fine-tuning (SFT) on tailored data, and then a reinforcement learning (RL) stage. The central goal of pretraining is to create a general foundation model that can be adapted to many potential tasks, which are the focus of SFT datasets. In general, the standard loss function that generative CLMs minimize in both PT and SFT is categorical cross-entropy of output token distributions, while loss functions vary for RL (see Sec.1.1). In practice, stages may be omitted, with reinforcement learning often missing from extant CLMs.

The RL stage provides a means of training beyond matching the SFT datasets and with regards to a set objective. For chemical discovery, RL has demonstrated potential to identify candidate druglike compounds (??) as well as functional materials (?), though identifying a suitable objective presents its own challenges (see Sec. 1.1). Despite such obstacles, the

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

prospects of RL for enabling downstream generative tasks for CLMs motivate its development. The challenge of multi-objective optimization (?), wherein a model in training improves across multiple goals, holds particular salience for molecular and materials discovery as ideal molecular and materials systems must often balance competing properties (?).

In this work, we outline and benchmark a novel approach, autonomous weighted schema (AWS) DPO for iteratively constructing and training on preference datasets for RL with a domain focus on molecular and materials discovery. Here, a multi-objective point defining a set of desired properties is defined prior to training, over the course of which a schema for constructing chemical preference datasets is repeatedly updated with regards to the multi-objective goal.

1.1 Related Work

Following a standard approach for tuning LLMs, proximal policy optimization (PPO) (?) has emerged as a standard method for aligning CLMs for a particular generative goal (??). In standard variant of this approach, a reward signal is provided by either a calculated score or a secondary reward model, which inherently converts a multi-objective domain into a single value.

To address multiple objectives and challenges of PPO for LLM tuning, Sun and coworkers partitioned datasets of preference data across a variety of desired characteristics such as "honesty" and "helpfulness" (?) that are used for Direct Preference Optimization (?) training. In DPO, one omits an explicit reward function or model and instead directly trains on the preference data for which balancing multiple objectives is a goal of the partitioning scheme. For materials and molecular discovery in particular, Salij and coworkers have proposed the construction of "weighted schemas" that partition chemical preference datasets according to a variety of metrics and iteratively applying such schemas (?), a process this manuscript refers to as iterative weighted schema (IWS) DPO. By creating an implicit reward in this manner, different, often competing goals are balanced.

Predefined schemas for partitioning preference datasets, however, require tuning of weights to determine acceptable dataset compositions. As the ideal set of weights for a target objective is not known a priori, many RL training runs may be required to obtain a suitable RL-trained model. Furthermore, as such weights are fixed, the schemas cannot adapt to the changing training environment that shifts over the course of RL training. For instance, RL with equal weighting for metrics A and B would still train to improve both even after A has been satisfied but B requires further optimization. An adaptive methodology could shift training focus to B in this case.

Concerning overall motivation and approach, this work has precedent in the general task of learning objectives over the course of training (??). This approach differs from prior RL online learning algorithms, particularly policy gradient methods, in the learning of a preference dataset curation scheme across a multi-objective frontier.

In this paper, we propose such an adaptive approach with autonomous weighted schema (AWS) updating where a target meta-objective is set and schema weights are iteratively updated. We benchmark this methodology by incorporating it into DPO and demonstrating improvements in output chemical distributions over iterative DPO without such updates.

2 Methods

2.1 Chemical Distributions

We notate a chemical distribution as $\chi(\vec{p})$ where $\vec{p}$ is a vector of scalar properties such as the number of heavy atoms N_H and other benchmarked metrics. Certain metrics are estimated via chemical property prediction models, i.e., in this study instances of a message-passing graph neural network with ChemProp architecture (?)), which enables a mapping between chemical graphs and estimated scalars. We note that such a mapping serves moreso as a score than an actual estimate in the cases where a value would not exist for the corresponding

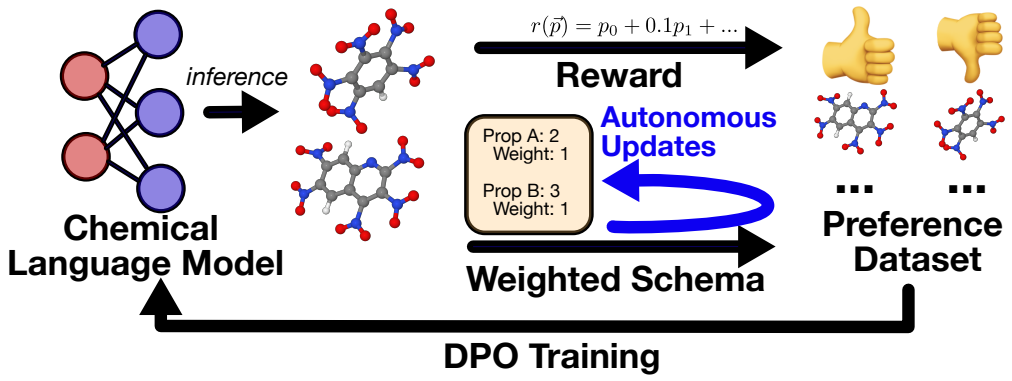

Figure 1: Comparison Schema of approaches for Direct Preference Optimization (DPO) of Chemical Language Model.

molecule. For a given distribution $\chi(\vec{p})$, we notated the mean value for a metric p_i as $\bar{\chi}(p_i) = \bar{p}_i$.

2.2 Weighted Schema Direct Preference Optimization

Following prior work,(?) a weighted schema $\mathbf{S}$ is defined for a set of metrics which are either scalar values associated with chemical properties or a function that combines such values into a composite scalar, henceforth referred to as a "(sub)-reward function" $r(\vec{p})$. These metrics are divided into a primary metric with associated hyperparameter p that denotes the percentile of molecules that are selected for use in preferences corresponding to secondary metrics, which are the remainder of metrics. In such weighted schemas, each metric m_i is associated with a weight w_i that balances preferences data corresponding to each metric. Preference data pairs are corresponding constructed to either maximize, minimize, or get close to a target value for a given metric with the standard case being to split molecular sets in two and match ordinal pairs (e.g., for a set of 10 molecules ranked by metric, $1 \leftrightarrow 6$, $2 \leftrightarrow 7$, and so on).

Multiple weighted schemas $\mathbf{S}_j$ may be combined into a aggregate meta-schemas $\mathbf{MS}$ with meta-weights w_j. The investigation of this paper focuses on adjusting such meta-weights, though the autonomous weight updating approach also applies in principle to components of individual schemas $\mathbf{S}_j$ as well. For determining the initial generated molecules to be used in the aggregate preference dataset, duplicate molecules are omitted by canonical SMILES (?) matching, which tends to aid with maintaining diversity of generated molecules (see appendix Fig. 5), though the same molecule may be included across different preferences.

Training minimizes a length-normalized DPO loss (?)

$$l_{\text{dpo}}(y_c, y_r) = -\log \sigma\Big(\frac{\gamma}{|y_c|}\log\Big(\frac{\pi_\theta(y_c)}{\pi_{\text{ref}}(y_c)}\Big) - \frac{\gamma}{|y_r|}\log\Big(\frac{\pi_\theta(y_r)}{\pi_{\text{ref}}(y_r)}\Big)\Big) \tag{1}$$

where $y_c(y_r)$ are chosen(rejected) token sequences generated from empty prompts and normalized by their lengths, $\pi_{\theta(\text{ref})}$ are training(reference) models, $\log \sigma$ is the logistic sigmoid function, and the hyperparameter $\gamma = 0.1$, is minimized over the course of training. Training occurs with initial learning rate of 0.0001, batch size of 128 over the preference dataset, and an Adam optimizer (?) with AMSGrad variant (?) and with decoupled weight decay (?) of 0.1. For detailed parameterization, we refer to the associated software defaults and configuration files provided.

2.3 Autonomous Weight Updates

$$w_{i,j} = w_{i-1,j} + \text{sgn}(d_{ij})(\alpha\Delta_{ij} + \beta d_{ij}) \tag{2}$$

where sgn is the sign function and α and β are the momentum and distance weight update hyperparameters, respectively. Weights begin initialized to some default values and are periodically updated according to α/β and distance proportional steps. Since the ideal weights with respect to some target objective is not known a priori, this methodology proposes the above equation as an ansatz to make slight adjustments with each DPO training meta epoch.

2.4 Model

We begin by fine-tuning the generative pretrained transformer model χhem-GPT 2 (??), which is a CLM that outputs probabilities of tokens in GroupSELFIES(?), a fragment-based text representation of molecular graphs. Except where otherwise noted, fine-tuning consists first of an initial stage of SFT on a custom material dataset followed by a single round of DPO on a shared preference dataset. These initial post-training stages have been observed by us to be important for starting the iterative RL in a desirable region of chemical space but are not the core focus of this paper. Rather, we isolate the final stage of fine-tuning wherein the generative GPT is tuned on information gained from its own outputs.

We consider in detail two representative case studies on pharmaceutical molecule discovery and energetic materials discovery, which we will refer to as "Pharma" and "X", respectively. The "Pharma" case focuses on tuning various molecular scores (see Sec 2.6), which presents a moderately contrived set of goals that enable isolation of the methodology explored in this paper and present relatively simple cases for reproducibility. In contrast, the "X" cases demonstrate in practice multi-objective optimization across stability and performance metrics, but due to their sensitive nature some details are omitted.

2.5 Inference

Molecules are generated by next-token prediction and sampling at temperature $T = 1$. Model inference is generally performed in a distributed and batched fashion across 4 Nvidia A100s and where token sequences are pruned to the tokens in between a start and end token. The token sequences, which are in the molecular encoding GroupSELFIES, are then converted and made canonical into the more common chemical text format SMILES, from which unique molecular graphs are then constructed. Within a given meta-epoch of training, 16384 output sequences are generated, of which the vast majority are then successfully converted to molecular graphs with failures being noted as invalid.

2.6 Benchmarking

A suite of benchmarking metrics are calculated for output sequences We note that for data processing reasons, invalid output sequences are often handled as empty strings, though such cases are generally omitted from reported aggregate statistics except for validity itself and are generally $\ll 1\%$ of the net output sequences.

For many properties that can be directly determined from the molecular graph such as heavy atom count, we characterize them with RDKit (?). Certain properties such as certain proxies for stability and performance, however, cannot be directly determined from the molecular graph and are instead approximated using surrogate ChemProp(?) graph neural networks that provide a mapping from the chemical representation to predicted scalar values.

For the Pharma case, the primary metrics are quantitative estimate of druglikeness (QED) score (?) and synthetic accessibility (SA) score (?), which are calculated from graph representations of the generated molecules. For QED score, which presents a metric that combines aspects of more targeted metrics such as the logarithm of the partition coefficient, a higher value (closer to 1) indicates a molecule that has greater similarity with known druglike compounds. Similarly, the SA score provides an aggregate metric where a lower (also closer to 1) value indicates similarity with known, relatively simple, synthesizable compounds. While neither of these metrics captures all aspects of drug discovery nor are goals that one should

maximize arbitrarily, they nevertheless both represent scalar proxies for drug discovery goals are are benchmarked accordingly.

In the X case study, predicted proxies for stability and performance are determined with custom graph message-passing ChemProp (?) models, which provides a quick method of determining approximate metrics.

2.7 Weighted Schemas

In the standard Pharma case, we consider a schema of the form

Metric	Preference	Weight
$\text{QED} - 0.001 \cdot (N_H - 20)^2$	0.8	1
$\text{SA} - 0.01 \cdot (N_H - 20)^2$	3	2

where N_H is the number of heavy (i.e., non-hydrogen) atoms. Note that the QED and SA preferences both contain terms that shift values downward for deviations from $N_H = 20$, though we note that this roughly this number of atoms proves typical both following SFT and RL for the Pharma case. For AWS variants, the standard target considered is $\vec{t} = (\text{QED} = 0.8, 10 - \text{SA} = 7)$, where the specific reward function in the weighted schema has the size protection term removed and $10-\text{SA}$ is a target to keep the desired training direction positive for both properties. In the case considered in the main text the hyperparameters $(\alpha, \beta) = (-1, 1)$ were chosen. Comparative studies of hyperparameter sensitivity has been included in the appendix figure Fig. 6.

In the standard X Case, we consider a composite schema consisting of two sub-schemas:

Metric		Preference	Weight
$\text{Performance Proxy} - 0.01 \cdot (N_H - 25)^2$	$(p = 0.5)$	greater	1
$N_{\text{aromatic rings}}$		greater	1

Metric		Preference	Weight
$\text{Stability Proxy} - 0.1 \cdot (N_H - 25)^2$	$(p = 0.5)$	greater	1
$N_{\text{aromatic rings}}$		greater	1
$N_{\text{NO bonds}}$		greater	1

where p is a "performance percentile" above which the other, secondary, metrics must satisfy before being sampled. The above schemas are initially weighted 1.5 and 1, respectively. In the AWS cases considered in the main text, the hyperparameters $(\alpha, \beta) = (-1, 1)$ were chosen but the targets $\vec{t}$ vary across the stability/performance frontier.

For both case studies, corresponding iterative weighted schema (IWS) analogues where the schema weights do not change are compared.

2.8 Reward Controls

To isolate the effect of weighted schema approaches, controls incorporating rewards that combine the multi-objective property benchmarks are trained. In the Pharma case, the following two functions were used to score molecules, which were then paired into direct preferences with respect to a preference. The contrasted rewards r_X and r_Y are

$$r_X = \text{QED} - 0.1 \cdot \text{SA} - 0.001 \cdot (N_H - 20)^2 \qquad (3)$$
$$\text{preference} : 0.5$$

$$r_Y = -(\text{QED} - 0.8)^2 - 0.1 \cdot (\text{SA} - 3)^2 - 0.001 \cdot (N_H - 20)^2 \qquad (4)$$
$$\text{preference} : \text{greater}$$

where reward function r_X combines into a linear function the QED and SA scores with a preference towards a value of .5, which the pair $(\text{QED} = .8, \text{SA} = 3)$ satisfies, and the reward function r_Y aims to minimize the distance between (QED, SA) and $(.8, 3)$. Both reward functions contain a heavy atom term to be consistent with the weighted schemas.

For the X case, a schema of the form

Metric	Preference	Weight
$10 \cdot \text{Performance Proxy} + \text{Stability Proxy} - 0.1 \cdot (N_H - 25)^2 \quad (p = 0.5)$	greater	1
$N_{\text{aromatic rings}}$	greater	1

is applied, where the surrogate proxies are identical to those in the weighted schemas.

3 Results

Over the course of AWS DPO training, molecular distributions shift from a relatively broad region of chemical space to areas that better satisfy multi-objective targets (see Fig. 2). We note that this shift does not appear to primarily come from dramatically shifting generated molecules out of the initial property distributions but rather by focusing onto extant but relatively under-sampled areas. That is, the AWS DPO training focuses molecular generation onto a particular region of chemical space.

In the Pharma case study, we observe that weighted schemas are effective in focusing generation onto a target region of property space in contrast with particular reward functions. For r_X, we observe a relative overshooting of low SA score while for r_Y we observe that training focused less on low SA, high QED, in general. Differences between weighted schema approaches AWS and IWS would appear more modest, though the AWS trained models generate relatively more molecules with higher QED scores. The addition of AWS hyperparameters provides control over output distributions (see appendix for Fig. 6) without requiring a shift in the target $\vec{t}$.

In contrast with the Pharma case study where fine tuning goals include focusing generation from within the initial SFT-trained distribution, the X case has a meta-objective of pushing the frontier of performance and stability, an aspect reflected by some of the targets $\vec{t}$ and the reward functions used. We observe that while a schema involving a reward function that combines stability and performance broadly outperforms iterative weighted schemas that separate performance and stability, the addition of autonomous weight updates can shift outputs such that neither the combined reward nor AWS mostly dominate, with AWS examples able to produce more performant molecules than the combined reward at the expense of stability. This highlights how the AWS method enables schema weights to adapt towards satisfying objectives. Furthermore, by changing the target $\vec{t}$, the resulting distributions, shift accordingly. In Fig. 4, the example AWS A has a more stable and less performant target $\vec{t}_A$ than AWS C, and correspondingly, the output distributions following DPO for A have higher typical stability than those in C and vice versa for performance.

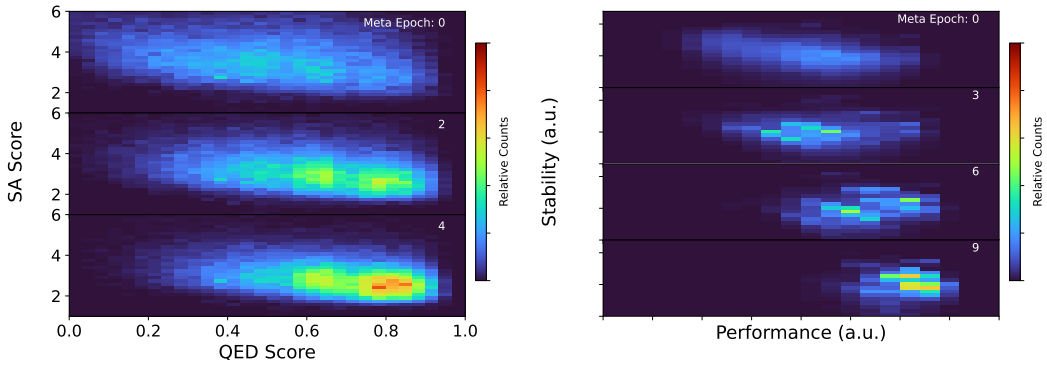

Figure 2: Example chemical distributions over the course of exemplar AWS training for targeting a low synthetic accessibility (SA) score and high quantitative estimate of druglikeness (QED) score (left) and for targeting a moderate stability and high performance (arbitrary units) for energetic materials discovery (right).

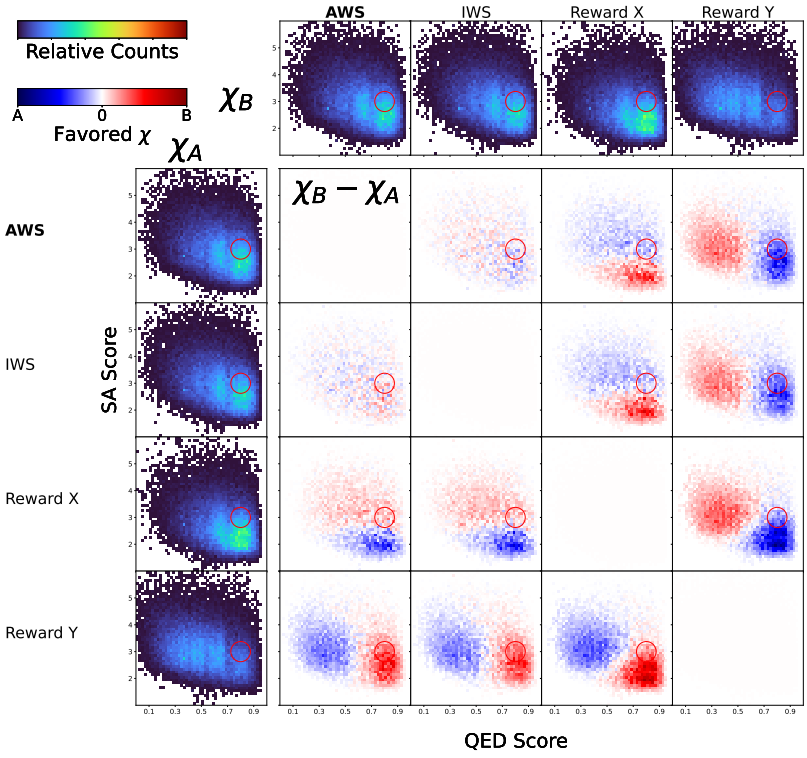

Figure 3: Comparison of generated synthetic accessibility (SA) and quantitative estimate of druglikeness (QED) score distributions of pharmaceutical-type molecules following various DPO approaches such as AWS (the subject of this manuscript), IWS (akin to AWS without weight updates), and DPO involving pairing from reward functions. Reward functions X and Y defined as $r_x = \text{QED} - 0.1 \cdot \text{SA} - 0.001 \cdot (N_h - 20)^2$ with a preference value of 0.5 and $r_x = (\text{QED} - .8)^2 - 0.1 \cdot (\text{SA} - 3)^2 - 0.001 \cdot (N_h - 20)^2$ with a preference of being greater and with N_h being the number of heavy (i.e., non-hydrogen) atoms. Highlighted in red circles is the AWS target value $\vec{p} = (\text{QED} = .8, 10 - \text{SA} = 7)$.

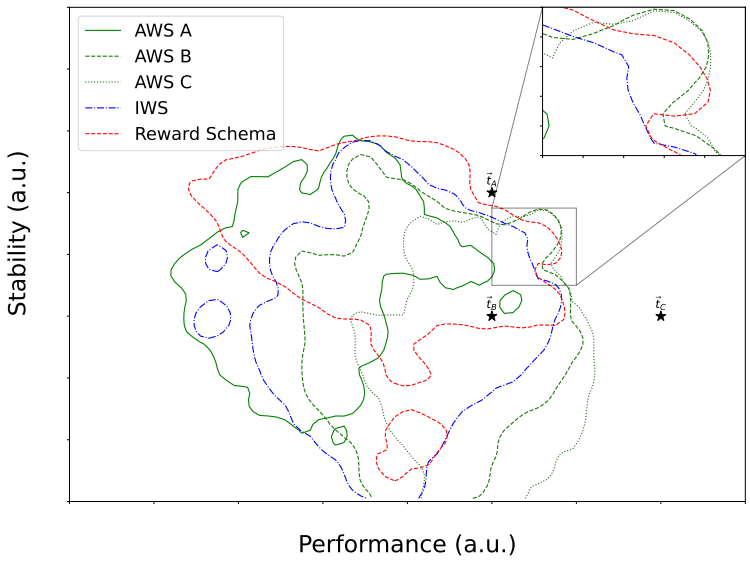

Figure 4: Distribution edges of various DPO methods, with AWS A-C indicating different autonomous targets, IWS indicating a shared initial weighted schema with the AWS approaches, and the compared reward schema being a simpler DPO schema where both stability and performance are included in a shared reward function used for preference pairing.

4 Discussion

This work demonstrates a methodological advancement in automatically updating preference data weights for DPO and with a focus on chemistry applications. In this approach, a set of performance metrics are defined and used to adjust intermediate preference weightings, presenting an objective that requires less manual tuning than directly setting the weights manually as in prior work. Such adjustments do not only permit improvements to weights from their initial conditions but also enable the preference optimization to adjust to changing training conditions in principle.

With regards to training dynamics, the observation that generated property distributions appear to be largely a subset of those present near the start of training presents a challenge to the possibility of RL for producing chemical systems that significantly push the frontier of chemical space. If RL methods such as the AWS DPO elaborated upon in this manuscript primarily reinforces existing CLM behavior, then their prospects for novel discovery is limited. Nevertheless, focusing generation into desirable regions has benefits, particularly in minimizing the amount of inference necessary to generate an quantity of candidates in that region of chemical space.

The results indicate that either weighted schemas or rewards can work for tuning, though plausible reward functions can behave relatively poorly, highlighting general challenges in engineering good reward functions (?). While potentially the perfect reward function could behave better than the weighted schemas considered in this paper, the central thrust of this work shows that separating concerns across dataset partitions works without requiring putting all of the objectives into a single function that balances them.

We note that such adaptive weights do not require use of DPO specifically and can generalize to other kinds of iterative preference optimization. That noted, the narrower chemical distributions $\chi(\vec{p})$ may prove limiting to the sequential training of a reward model as is standard in many implementations of PPO. While this work has focused on chemical space navigation, the weighted schemas and adaptive updates to them can be used in principle for any preference data that can be partitions according to defined categories although the

distinction between "primary" and "secondary" metrics becomes moot where there cannot be an ordinal ranking for the primary metric.

In whole, this work presents a modest advance in the general mission of navigating chemical space by defining and iterating according to human-legible target metrics. We observe general improvements over the analogous non-adaptive approach in satisfying multiple competing sub-objectives, demonstrating an effective technique for multi-objective optimization.

Acknowledgments

PROVIDED IN FINAL COPY.

AI Usage Disclosure

In the course of generating preference datasets, the chemical language models in this study generate synthetic data as detailed in the method section. The authors take responsibility for the correctness of this manuscript.

Reproducibility Statement

For the pharmacological case study, model weights at the start and conclusion of DPO training as well as associated files are provided in the supplementary files available at https://doi.org/10.5281/zenodo.22694082. Configuration files for running the studies are also provided as well, which may be run from the associated repository using provided configuration files. Datasets and associated files with for the Pharma case study are also provided.

A Appendix

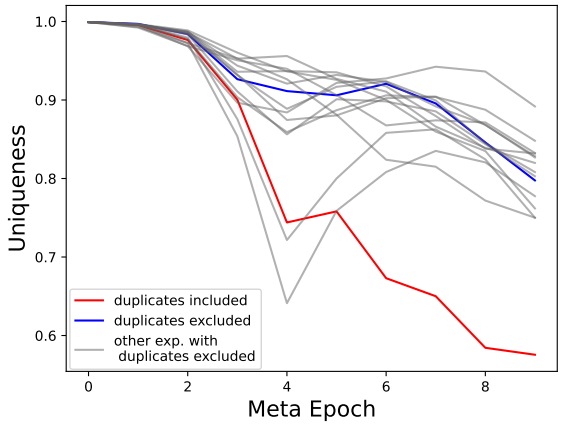

Figure 5: Mean uniqueness of generated molecules over the course of DPO training for energetic materials models. Note that the highlighted examples are equivalent training runs except for the presence or absence of duplicates in the dataset from which preference pairs are generated. Grey shows a set of other training runs, showing that the "duplicates" excluded is fairly representative of general training behavior.

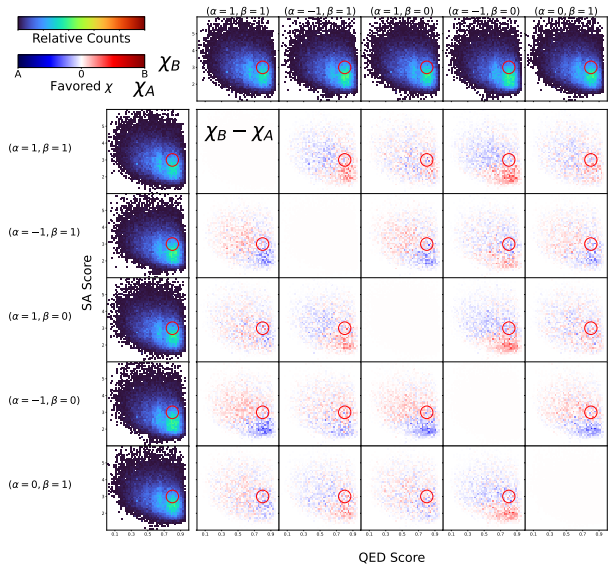

Figure 6: Comparison of generated synthetic accessibility (SA) and quantitative estimate of druglikeness (QED) score distributions of pharmaceutical-type molecules under various (α, β) parameters. In contrast with the main text, the initial weights are 1 for all metrics and the targets are set to SA $= 3$ instead of $10 -$ SA $= 7$. Highlighted in red circles is the AWS target value $\vec{p} =$ (QED $= .8$, SA $= 3$)