A Pareto-optimal compositional energy-based model for sampling and optimization of protein sequences

Nataša Tagasovska¹, Nathan C. Frey¹, Andreas Loukas¹, Isidro Hötzel², Julien Lafrance-Vanasse², Ryan Lewis Kelly², Yan Wu², Arvind Rajpal², Richard Bonneau¹, Kyunghyun Cho¹,²,³,⁴,⁵, Stephen Ra¹, and Vladimir Gligorijević¹

1Prescient Design, Genentech
2Antibody Engineering, Genentech
3Department of Computer Science, Courant Institute of Mathematical Sciences, New York University
4Center for Data Science, New York University
5CIFAR Fellow
natasa.tagasovska@roche.com

Abstract

Deep generative models have emerged as a popular machine learning-based approach for inverse design problems in the life sciences. However, these problems often require sampling new designs that satisfy multiple properties of interest in addition to learning the data distribution. This multi-objective optimization becomes more challenging when properties are independent or orthogonal to each other. In this work, we propose a Pareto-compositional energy-based model (pcEBM), a framework that uses multiple gradient descent for sampling new designs that adhere to various constraints in optimizing distinct properties. We demonstrate its ability to learn non-convex Pareto fronts and generate sequences that simultaneously satisfy multiple desired properties across a series of real-world antibody design tasks.

1 Introduction

Generative models have shown promise across various applications in the life sciences for generating chemically- and physically-plausible designs and in accelerating the process of scientific discovery. Part of this trend of adoption can be owed to the convincing examples created by generative adversarial networks (GANs) [13], variational autoencoders (VAEs) [23], energy-based models (EBMs) [8] and, more recently, diffusion models [35; 32; 5]. However, there are far fewer success stories in real-world industry applications [6; 15]. Some reasons include an overrepresentation of image datasets; a lack of evaluation protocols and metrics for synthetic data [37; 33]; challenges around controllable generation and training — for example GANs; and challenges in generating samples that are different from those observed during training [22; 2; 38]. Taken together, these serve to limit the applicability of generative modeling for real-world use cases.

Our work is motivated by this and, in particular, the need to accelerate the development and discovery of new molecules, namely therapeutic antibodies. Though several generative models have already been proposed for these purposes [10; 12; 31], generating new samples without guidance or control does not guarantee downstream success of the proposed molecules. In practice, each molecule has to satisfy multiple properties. For therapeutic antibodies, this could include properties such as binding affinity, polyreactivity, and viscosity [1; 20; 39]. Failure to account for these and other properties can lead to serious complications later on during scale-up, manufacturing, and clinical trials and the optimization of straightforward objectives does not necessarily translate into progress in the laboratory [28].

NeurIPS 2022 AI for Science Workshop.
Motivated by this challenge, we propose a new EBM for antibody design that simultaneously takes into account multiple properties that an antibody has to satisfy. The adherence to multiple properties is a challenge in itself, as it often involves optimizing multiple conflicting objectives. In antibodies, for instance, optimization of binding affinity alone may come at the expense of developability properties, parameters that govern the likelihood of success of a molecule during manufacturing and quality control.

In general, in optimizing multiple conflicting properties, it is often impossible to find a single sample that satisfies all of the objectives simultaneously [21]. We argue that, instead, one should aim to propose a set of diverse data points from the Pareto front [3] that correspond to different choices for various objective functions. Doing so enables a global perspective the optimal trade-off between objectives and can select a molecule according to their preference. To achieve Pareto optimality, we rely on recent advances in multi-objective optimization (MOO) [34; 27] as well as on compositional sampling with EBMs [8] to build a Pareto-optimal compositional EBM (pcEBM). Figure 1 exemplifies both why we need pcEBM and what we achieve with it when optimizing a design of an initial antibody sequence.

We first present the the necessary background on multi-objective optimization in subsection 2.1 and related work on compositional sampling with EBM in subsection 2.2 leading to pcEBM described in subsection 2.3. In section 3 we include empirical evaluation and discussion, before concluding in section 4.

2 Background and method

Problem setup. Antibodies are composed of two chains of amino acids (AA), which can be represented as sequences of characters. Each AA comes from an alphabet of 20 characters corresponding to the type of each AA. For each character, we measure m properties, such that \( f_i : \mathbb{R}^L \rightarrow \mathbb{R} \) for all \( i \in 1, \ldots, m \).

Our goal is to generate new sequences \( x^* \) with preferred values for each of the m properties. Since we cannot fully satisfy all of the properties/objectives simultaneously (they may be conflicting with
Although EBMs have been around for many years [25; 24], recent interest in generative modeling has led to a resurgence of interest [8; 9]. EBMs learn to represent data by approximating an unnormalized probability distribution across data. They do so by learning an energy function $E_\theta(x)$ parameterized by a neural network that maps each input $x$ to a scalar real value interpreted as the energy and approximating the data distribution by the Boltzmann distribution under unit temperature:

$$p_\theta(x) \propto e^{-E_\theta(x)}.$$ 

Training an EBM on a given data distribution is usually done by contrastive divergence [16]. Besides predicting the un-normalized data likelihood, EBMs can be used to sample new data points $x$ from $p_\theta$ (during both training and generation). Sampling is typically performed by some Markov-Chain Monte-Carlo (MCMC) procedure. We consider the common case of Langevin Diffusion (LD) where the samples are initialized from refinement/uniform random noise followed by iterative refinement:

$$x = x^{k-1} - \frac{\eta}{2} \nabla_x E_\theta(x^{k-1}) + \omega^{k}, \quad \omega \sim \mathcal{N}(0, \sigma^2),$$

2.1 Multi-objective optimization

A simple approach to solving MOO is linear scalarization [21] based on a preference vector $\lambda = [\lambda_1, \ldots, \lambda_m]$ from the probability simplex on $[m]$, i.e., $S = \{ \lambda : \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, i \in [m] \}$. Each $\lambda \in S$ leads to a weighted objective function $f_\lambda(x) = \sum_{i=1}^{m} \lambda_i f_i(x)$ and its minimizer $x_\lambda^* = \operatorname{argmin}_x f_\lambda(x)$. As we evaluate on $\lambda$ from a grid of $S$, we hope that the corresponding $x_\lambda^*$ approximates the Pareto front. The solutions obtained by linear scalarization lie in the convex envelope of the Pareto front, which renders the approach plausible for situations where the Pareto front is convex [27].

An alternative approach for non-convex Pareto fronts is multiple gradient descent (MGD) [7], which iteratively updates variable $x$ along a direction that ensures that all objectives are decreased simultaneously and guides the optimization towards a Pareto improvement. Denote the gradient of the $i$-th objective as $\nabla_x f_i(x)$. With gradient descent, we update the variable $x^k \leftarrow x^{k-1} - \eta g(x)$, where $g(x)$ is a vector to be determined and $\eta$ is a small step size. From a 1st-order Taylor expansion, we can deduce that $(\nabla_x f_i(x), g) \approx -(f_i(x^k) - f_i(x^{k-1})) / \eta$ which represents the decreasing rate of $f_i$ when we update $x$ along direction $g(x)$. In MGD, $g$ is chosen to maximize the slowest decreasing rate among all the objectives, that is:

$$g(x) \propto \max_{i \in [m]} \{ \min_{x \in \mathbb{R}^d} (g, \nabla_x f_i(x)) \text{ subject to } \|g\|_2 \leq 1 \}$$

By doing so, $g(x)$ is pushed to have positive inner products with all $\nabla_x f_i(x)$. If the latter is impossible, the solution of the above optimization problem is the optimal objective value $g(x) \approx \sum_{i=1}^{m} \lambda_i \nabla_x f_i(x)$ where $\lambda_i \nabla_x f_i(x)$, $\lambda_i \geq 0$, and $\lambda_i = 1, 1 \leq i \leq m$ is the solution of

$$\min \| \sum_{i=1}^{m} \lambda_i \nabla_x f_i(x) \|_2 \text{ subject to } \sum_{i=1}^{m} \lambda_i = 1 \text{ and } \lambda_i \geq 0 \text{ for all } i \in [m].$$

The above convex optimisation problem has a closed form solution when $m = 2$ and a fast algorithm for $m > 2$ [34; 19]. By construction, when the step size is small, MGD decreases monotonically all objectives simultaneously and will terminate at a local Pareto point.

2.2 Compositional energy-based models

Although EBMs have been around for many years [25; 24], recent interest in generative modeling has led to a resurgence of interest [8; 9]. EBMs learn to represent data by approximating an unnormalized probability distribution across data. They do so by learning an energy function $E_\theta(x)$ parameterized by a neural network that maps each input $x$ to a scalar real value interpreted as the energy and approximating the data distribution by the Boltzmann distribution under unit temperature:

$$p_\theta(x) \propto e^{-E_\theta(x)}.$$
with \( k \) being the sampling step and \( \eta \) the step size.

The above sampling procedure can be easily extended to produce samples from the composition of \( p_\theta \) with other distributions of interest. Along this direction, Du et al. [9] considered the problem of sampling from \( m \) composed distributions, each modeling a different property \( f_i \), an approach referred to as compositional EBM (cEBM). Du et al. proposed multiple variations for executing different logical operations over properties of interest, such as conjunction, negation, and disjunction. In the following, we focus on conjunction, which corresponds to combining individual EBMs as a product of experts:

\[
p(x | f_1 \land f_2 \land \cdots \land f_m) = \prod_i p(x | f_i) \propto e^{-\sum_i E(x|f_i)}.
\]

We also note that cEBM utilized the following sampling procedure:

\[
x = x^{k-1} - \frac{\eta}{2} \nabla_x \sum_i E_\theta(x^{k-1}|f_i) + \omega^k, \quad \omega \sim \mathcal{N}(0, \sigma^2),
\]

(3)

which amounts to applying LD w.r.t. \( E(x) = \sum_i E(x|f_i) \).

2.3 Pareto compositional sampling

Inspired by previous work on multi-objective optimization [7; 27] discussed in subsection 2.1, we here propose pcEBM, a method that samples from the Pareto front of \( m \) Boltzmann distributions of interest. Our approach entails utilizing multiple gradient descent to select a locally optimal direction that optimizes the slowest decreasing rate among all the objectives of the compositional EBMs:

\[
x^k \leftarrow x^{k-1} - \eta \arg\max_{g \in \mathbb{R}^d} \{ \min_{i \in [m]} \langle g, \nabla_x f_i(x^{k-1}) \rangle \} \text{ subject to } ||g||_2 \leq 1 + \sqrt{2}\alpha \omega,
\]

(4)

where \( \alpha \) is a positive constant and the internal optimization problem is the same utilized in MGD and the external sampler is performing Langevin diffusion.

Suppose that we are generating a new sequence \( x \), either by starting from random noise or by improving an existing one. At each step \( k \), if \( x^{k-1} \) is far from the Pareto front and the gradients \( \nabla_x f_i(x^{k-1}) \) have non-negligible norm, Equation 4 will drive \( x^{k-1} \) closer to the Pareto front. On the other hand, when \( x^* \) is close to the Pareto front and the gradients have nearly vanished, the noise dominates and \( x^{k-1} \) performs Brownian motion. As will be demonstrated experimentally, the addition of noise \( \omega \) helps pcEBM to explore more efficiently the Pareto as compared to MGD, whereas the locally optimal determination of the optimization direction renders the optimization more effective than cEBM.

3 Empirical evaluation

In the following, we compare the Pareto compositional EBM pcEBM with three baselines: multiple gradient descent (MGD) [7], a compositional EBM (cEBM) [9], and a linearly scaled cEBM (ls-cEBM) [21].

**Experimental setup.** We wish to generate plausible antibody sequences according to three key properties: (i) similarity to known human antibodies from the public database of observed antibodies [29] (Ab-like), (ii) binding affinity to antigens of interest (Aff), and (iii) an experimental measure of nonspecificity — phenomena which can affect the dosing regimen, safety, and overall therapeutic profile of an antibody — which we refer to as the BV score [17]. We use datasets from both

\[^1\]An enzyme-linked immunosorbent assay (ELISA) of non-specific binding to baculovirus particles [17].

Figure 2: Pareto fronts and estimated hyper volume for all baselines with \( \eta = 0.01 \) and \( k = 400 \). The axes show the energy of EBMs that were separately trained on data with good affinity and BV scores, respectively, with smaller energies being more desirable.
public and proprietary sources to train three separate EBMs, each pertaining to a property of interest. Though alternative choices are also possible, we instantiate all models as sequence-based convolution networks [11]. More details on the architecture can be found in the Appendix. We use the same models for sampling new sequences with each of the baselines. The differences in implementation are as follows: MGD does not add any noise, cEBM uses Lanvegin dynamics but has uniform weights for all properties; ls-cEBM differs from cEBM in that it includes domain-informed weights; and finally pcEBM learns the optimal weights for each property per sequence. Each baseline relies on two main hyper-parameters, namely the step size $\eta$ and the number of steps $k$.

To compare different approaches in terms of their ability to satisfy multiple properties, we adopt the Hyper-Volume (HV) metric from the multi-objective optimization literature [14] computed over the energy scores of each property model. In all experiments, we aim for minimizing the energy scores per property. Additionally, as a proxy for how well the sampled sequences capture the properties of interest we compute the minimal edit distance of each sampled sequence to the ones seen during the training of each property EBM ($e_{dist}$) [30]. More details about the computation of these metrics can be found in the Appendix.

Results and discussion. Our experiments are designed to provide answers to the following research questions:

- Q.1 Can we use EBMs to generate valid/reasonable, multi-property compliant antibody sequences?
- Q.2 Does sampling with Langevin Dynamics propose more Pareto optimal sequences?
- Q.3 Does pcEBM have an advantage over cEBM?
- Q.4 Starting from a seed sequence, can we improve a property of interest using (p)cEBM?

Table 1: Hyper-Volume (HV) across multiple combinations of properties (as measured by energy score) across different step sizes $\eta$. Larger scores indicate bigger improvement. The best results for are typeset in boldface.

<table>
<thead>
<tr>
<th>$\eta = 0.01$</th>
<th>HV(Ab-Like, Aff, BV)</th>
<th>HV(Aff, BV)</th>
<th>HV(Ab-Like, BV)</th>
<th>HV(Ab-Like, Aff)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGD</td>
<td>0.04</td>
<td>0.22</td>
<td>0.23</td>
<td>0.29</td>
</tr>
<tr>
<td>ls-cEBM</td>
<td>0.00</td>
<td>0.24</td>
<td>0.24</td>
<td>0.23</td>
</tr>
<tr>
<td>cEBM</td>
<td>0.00</td>
<td>0.25</td>
<td>0.24</td>
<td>0.25</td>
</tr>
<tr>
<td>pcEBM</td>
<td><strong>0.049</strong></td>
<td><strong>0.3</strong></td>
<td><strong>0.29</strong></td>
<td><strong>0.30</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\eta = 1$</th>
<th>HV(Ab-Like, Aff, BV)</th>
<th>HV(Aff, BV)</th>
<th>HV(Ab-Like, BV)</th>
<th>HV(Ab-Like, Aff)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGD</td>
<td>0.04</td>
<td>0.25</td>
<td>0.25</td>
<td>0.27</td>
</tr>
<tr>
<td>ls-cEBM</td>
<td>0.00</td>
<td>0.25</td>
<td>0.24</td>
<td>0.25</td>
</tr>
<tr>
<td>cEBM</td>
<td>0.046</td>
<td>0.29</td>
<td>0.29</td>
<td>0.3</td>
</tr>
<tr>
<td>pcEBM</td>
<td>0.041</td>
<td>0.27</td>
<td>0.27</td>
<td>0.29</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\eta = 40$</th>
<th>HV(Ab-Like, Aff, BV)</th>
<th>HV(Aff, BV)</th>
<th>HV(Ab-Like, BV)</th>
<th>HV(Ab-Like, Aff)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGD</td>
<td>0.033</td>
<td>0.25</td>
<td>0.25</td>
<td>0.26</td>
</tr>
<tr>
<td>ls-cEBM</td>
<td>0.043</td>
<td>0.27</td>
<td>0.26</td>
<td>0.26</td>
</tr>
<tr>
<td>cEBM</td>
<td>0.046</td>
<td>0.29</td>
<td>0.29</td>
<td>0.29</td>
</tr>
<tr>
<td>pcEBM</td>
<td>0.037</td>
<td>0.27</td>
<td>0.27</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Table 1 and Table 2 address Q.1. We find that EBMs trained on data possessing a single property of interest propose sequences that are similar to those sharing the same property, but are dissimilar to the sequences preferred by other objectives. In contrast, sequences generated by a multi-property EBMs have smaller distances to all properties compared to the single-property EBMs. Amongst the four considered multi-objective baselines, pcEBM generates sequences with greater hyper-volume and similarity to the training datasets, whereas MGD is a close second.

We surmise the difference between cEBM and pcEBM is owed to their speed of convergence. We confirm this in Figure 3 by plotting the energy score at each sampling step for both cEBM and pcEBM with a step size of $\eta = 0.01$. As observed, for the same starting sequence pcEBM converges faster. This result partially answers Q.3.
Table 2: Edit distance (mean with standard deviation in brackets) to sequences in the training set with property of interest. Results in bold are best per columns, smaller is better, considering $e_{dist}$ as a proxy for similarity to real data with the specific property.

<table>
<thead>
<tr>
<th>Property</th>
<th>$e_{dist} (generated, Ab-like)$</th>
<th>$e_{dist} (generated, BV)$</th>
<th>$e_{dist} (generated, Aff)$</th>
<th>Average $e_{dist}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>single-property EBM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ab-Like EBM</td>
<td>98.74 (±19.25)</td>
<td>93.84 (±19.05)</td>
<td>103.3 (±17.8)</td>
<td>98.62</td>
</tr>
<tr>
<td>Stc EBM</td>
<td>84.68 (±15.78)</td>
<td><strong>69.7</strong> (±16.42)</td>
<td>88.29 (±5.72)</td>
<td>121.33</td>
</tr>
<tr>
<td>Aff EBM</td>
<td>88.51 (±16.25)</td>
<td>81.4 (±15.6)</td>
<td>86.73 (±6.31)</td>
<td>85.55</td>
</tr>
<tr>
<td>$\eta = 40$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MGD</td>
<td>83.51 (±16.1)</td>
<td>71.49 (±15.7)</td>
<td>85.84 (±16.3)</td>
<td>80.28</td>
</tr>
<tr>
<td>cEBM</td>
<td>92.96 (±19.03)</td>
<td>86.86 (±17.99)</td>
<td>97.97 (±17.28)</td>
<td>92.60</td>
</tr>
<tr>
<td>ls-cEBM</td>
<td>92.4 (±19.3)</td>
<td>87.14 (±17.94)</td>
<td>97.93 (±17.25)</td>
<td>92.49</td>
</tr>
<tr>
<td>pcEBM</td>
<td><strong>82.24</strong> (±16.16)</td>
<td>71.47 (±15.82)</td>
<td><strong>85.76</strong> (±16.34)</td>
<td><strong>79.82</strong></td>
</tr>
<tr>
<td>$\eta = 0.01$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MGD</td>
<td>91.46 (±16.23)</td>
<td>84.02 (±15.43)</td>
<td>92.91 (±16.48)</td>
<td>89.46</td>
</tr>
<tr>
<td>cEBM</td>
<td>219.23 (±4.43)</td>
<td>222.47 (±4.45)</td>
<td>221.99 (±4.4)</td>
<td>221.23</td>
</tr>
<tr>
<td>ls-cEBM</td>
<td>218.78 (±4.66)</td>
<td>218.13 (±4.58)</td>
<td>218.62 (±4.73)</td>
<td>218.51</td>
</tr>
<tr>
<td>pcEBM</td>
<td>89.35 (±14.64)</td>
<td>79.67 (±13.99)</td>
<td>89.04 (±14.37)</td>
<td>86.02</td>
</tr>
</tbody>
</table>

To further test our approach, we calculate HV across the three properties we aim to account for. Here, we consider the energy of each property EBM as proxy for the objective of interest. The advantage of pcEBM is noticeable at smaller step sizes (see Table 1), while the variance in both edit distance and hyper-volume for different $\eta$ is significantly smaller for pcEBM than others. Figure 2 depicts the Pareto front for two properties obtained from candidate sequences from each baseline. We find that the pcEBM front is broader than MGD-based approaches, which agrees with our supposition that employing LD leads to an improved coverage of the Pareto front as compared to pure gradient descent. Further, pcEBM and MGD fronts are closer to the origin than that obtained by direct LD, which demonstrates the advantage of the multi-property optimization aspect of pcEBM, addressing Q.2 and Q.3.

Finally, for Q.4 we wish to explore the possibility for improving properties of already existing antibodies by using the variants of compositional EBMs. We focus our analysis on nonspecificity (BV) as we have an external SeqCNN classifier that can act as a surrogate pseudo-oracle. We start by screening the existing sequences to select only those with low BV scores, which we refer to as “seeds”. We use sequences with poor experimental and predicted BV scores as seed sequences for generation and then reuse the nonspecificity pseudo-oracle to evaluate the newly proposed designs. Figure 5 shows the BV score improvement achieved by each baseline. We notice that the multi-objective methods manage to improve the worst BV scores from 0.27 to above 0.9, with MGD and pcEBM giving a slightly lower average score than cEBM (though all scores are comparably high).

The plots in Figure 4 provide further insight into the nonspecificity optimization process by examining a low-dimensional embedding of seed and optimized sequences. The embedding is obtained by using tSNE to project the last layer feature representations of the sequences obtained from the nonspecificity oracle, onto a 2D space. We color each point (sequence) by their BV score with blue indicating a good score and red a poor one. On the same plot, we overlay a seed sequence and its corresponding proposed designs. We see that both cEBM and pcEBM improve the probability
Figure 4: tSNE embeddings of antibody sequences colored based on BV score, with blue sequences having better scores. The figures show how a sequence with poor BV score of 0.68 (in orange) is improved by two models. Left - cEBM, Right - pcEBM.

Figure 5: Left - violin plots of the distributions of predicted BV scores for the initial set of sequences to be improved (seeds), and for the proposed designs from each baseline. Higher BV scores are better. Right - energy scores of trajectories proposed by pcEBM for a different number of steps. Lower energy scores are better/preferred for the objectives.

for a good nonspecificity score, and move the seed towards the section of the manifold covered by datapoints with good scores.

4 Conclusion

We propose a new approach to sampling data points that simultaneously satisfy multiple desired properties. We do so by integrating multiple gradients within compositional energy based models. In a series of experiments for generating and improving existing sequences, we validate the performance of the compositional EBMs as well as their Pareto optimality. The generality and modularity of pcEBM allow for multiple expansions we are excited to explore: (i) other data types, in particular graph structures suitable for molecules, (ii) incorporating other sampling frameworks such as Stein variational gradient descent [26], (iii) incorporating uncertainty estimates, and (iv) Pareto-optimal training for EBMs, to name a few.

References


Appendix

4.1 Additional details on experiments

SeqCNNs For all properties and baselines we use an identical NN architecture, that is one model per protein chain, consisted of three Conv1D layers with kernel size 9 and padding 1, ReLU non-linearities and penultimate layer of size 256. All EBMs were trained with contrastive training, Adam optimizer and early stopping criterion.

Hyperparameters For each multiproperty sampling baseline, the following hyper parameter grid was explored with random search:

- step size - $\eta \in \{1e^{-4}, 1e^{-2}, 1, 10, 40\}$
- number of steps - $k \in \{100, 200, 300, 400\}$
- noise type - $\omega \in \{N, U\}$.

4.2 Metrics

Hypervolume HV is an indicator for evaluating the quality of sets of solutions in multi objective optimization. For a given reference point of $r = [r_1, ..., r_m]$ we have an upper bound of the objectives, such that $\sup x f_i(x) \leq r_i, \forall i \in [m]$. For a given set of solutions $X = \{x_i\}_{i=1}^n$, a hypervolume indicator $HV(X)$ is a measure of the region between all $f_i$ objectives and $r$:

$$HV(X) = \Lambda \{(q \in \mathbb{R}^d, \exists x \in X : q \in \prod_{i=1}^{m} [f_i(x), r_i])\}$$

where $\Lambda(\cdot)$ denotes the Lebesgue measure.

In the context of our experiments, we are interested in minimizing the energy scores per each property, and we choose a reference point $r = (1, 1, 1)$. We compute HV from each Pareto front to the reference point with the implementation provided in PyMoo [18].

Edit distance Edit distance is the typical score for quantifying how dissimilar two strings (in our case, sequences) are to one another, measured by counting the minimum number of operations required to transform one string into the other. Although different variants exist, here we rely on the Levenshtein distance which accounts for insertions, deletions and substitutions. In all experiments we use the python library edist as implemented in [36].