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

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generative models constitute one of the prominent machine learning methods leveraged by engineering and life sciences. Nevertheless, besides resemblance to the data we wish to sample, oftentimes our aim is to propose new data points that will satisfy, not one, but multiple properties of interest. Such aspiration becomes even more challenging when those properties are independent or orthogonal to each other. In this work, we propose a *Pareto-compositional energy-based model (pcEBM)*, a generative framework for sampling new datapoints that simultaneously adhere to various constraints and optimize distinct properties. pcEBM emerges naturally from an integrated multi-gradient energy-based sampling framework and is shown to lead to practical benefits in a series of real world (antibody-design) empirical analysis.

1 Introduction

Generative models have recently attracted significant attention within the machine learning community. Part of the excitement stems from the convincing images created by generative adversarial networks (GAN) [13], variational autoencoders [20], energy-based models, (EBMs) [9] or more recently diffusion models [31] that made headlines [29, 5]. However, there are far less success stories in industry applications [6, 15]. There are multiple reasons for this fallback: not all datasets are in image format; there is a lack of evaluation protocols and metrics for synthetic data; it is still challenging to control and train some of the generative models (i.e., GANs); and finally, we are still not able to generate or imagine samples that are (very) different from what we have seen in training. Overall, the lack of constraints and control in generative modeling limits the current use cases.

This manuscript is motivated by the need to accelerate the development and discovery of new drugs, namely, therapeutic antibodies by incorporating machine learning methods in classical computational biology pipelines. Though multiple generative models have already been proposed for such purposes [10, 12, 28], generating new samples without guidance or control does not guarantee usability or success of the proposed molecules. In practice, each molecule has to satisfy multiple properties, such as affinity, aggregation, and thermal stability [1, 33], before it qualifies as a potential candidate for a clinical trail. Not taking such fundamental properties into consideration leads to potential complications later on during clinical trials and the optimization of straightforward objectives does not necessarily translate into progress in the laboratory [25].

Motivated by this real-world challenge, we here propose a new energy-based generative method targeted at antibody design and discovery that simultaneously takes into account multiple properties that an antibody (protein) has to satisfy. The concurrent adherence to multiple properties is a challenge in itself, as it often involves optimizing multiple conflicting objectives. In antibodies, for

Submitted to NeurIPS 2021 AI for Science Workshop.

instance, naive optimization of affinity can be achieved at the expense of other desirable developability properties.

In general, when one cares about the optimization of multiple conflicting properties, it is often impossible to find a single sample that satisfies all of the objectives simultaneously. 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 the different objective functions. By doing so, one gains 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)[30, 24] as well as on compositional sampling with EBMs [9] to build a Pareto-optimal compositional EBM (*pcEBM*).

In what follows, 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 (20 characters corresponding to each of the possible amino acids found in humans), typically of combined length $L \sim 250$. We will denote the sequences as $x = (x_1, \ldots, x_L)$, where $x_l \in \{1, \ldots, 20\}$ corresponds to the AA type at position l. For each of those sequences, we measure m properties, such that $f_i : \mathbb{R}^L \to \mathbb{R}$ 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 each other), we are interested in finding samples which can not be further improved simultaneously for all the objectives, yielding the notion of *Pareto optimality* [4].

2.1 Multi-objective optimization

A straightforward approach to solving MOO is *Linear Scalarization* [19] 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 $x^*_\lambda = \text{argmin}_x f_\lambda(x)$. As we evaluate on λ from a grid of S, we hope that the corresponding x^*_λ 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 [24].

An alternative for non-convex Pareto fronts is *Multiple Gradient Descent (MGD)* [7], which is based on the idea of iteratively updating the variable x along a direction that ensures that all the objectives are decreased simultaneously, i.e., that 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 η is a small step size. From a 1st-order Taylor expansion, we can deduce that $\langle \nabla_x f_i(x), g \rangle \approx -(f_i(x^k) - f_i(x^{k-1}))/\eta$ 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:

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

By doing so, $g(x)$ is pushed to have positive inner products with all $\nabla_x f_i(x)$. If the latter is impossible, $\{\nabla_x f_i(x)\}_{i=1}^m$ will contain conflicting directions and we will get $g(x) = 0$ which terminates the procedure. By using Lagrangian duality, Désidéri [7] has shown that the above objective has the optimal value $g(x) \approx \sum_{i=1}^m \lambda_i^* \nabla_x f_i(x)$ where $\{\lambda_i \nabla_x f_i(x)\}_{i=1^m}$ is the solution of

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

The above convex optimisation problem has a closed form solution when $m = 2$ and a fast algorithm for $m > 2$ [30, 18]. 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 since many years [22, 21], their utility was recently rediscovered in the quest of generative modeling. 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_θ (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:

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

with k being the sampling step and η the step size.

The above sampling procedure can be easily extended to produce samples from the composition of p_θ with other distributions of interest. Along this direction, Du et al. [8] considered the problem of sampling from m composed distributions, each modeling a different propertiey 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:

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

We also note that cEBM utilized the following sampling procedure:

$$x = x^{k-1} - \frac{\eta}{2}\nabla_x \sum_i E_\theta(x^{k-1}|f_i) + \omega^k,\ \omega \sim \mathcal{N}(0, \sigma^2), \tag{3}$$

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

2.3 Pareto compositional sampling

Inspired by the previous work on multi-objective optimization [7, 24] 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:

$$x^k \leftarrow x^{k-1} - \eta \operatorname*{argmax}_{g \in \mathbb{R}^d}\{\min_{i \in [m]} \langle g, \nabla_x f_i(x^{k-1})\rangle \text{ subject to } \|g\|_2 \leq 1\} + \sqrt{2}\alpha\omega, \tag{4}$$

where α 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 ω 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 pareto compositional EBM (*pcEBM*) with three baselines: multiple gradient descent (*MGD*) [7], compositional EBM (*cEBM*) [8], and linearly scaled cEBM (*ls-cEBM*) [19].

Experimental setup. We wish to generate plausible antibody sequences. We consider three key properties for the generated antibodies: (*i*) similarity to known human antibodies from the public database of observed antibodies [26] (*Ab-like*), (*ii*) binding affinity to antigen targets of interest (*Aff*), and (*iii*) antibody clearance evaluated as off-target binding, presumed to account for the fast clearance [17], to which we refer as non-specificity and shortly (*BV*) [1]. The datasets we use are from both public and proprietary sources.

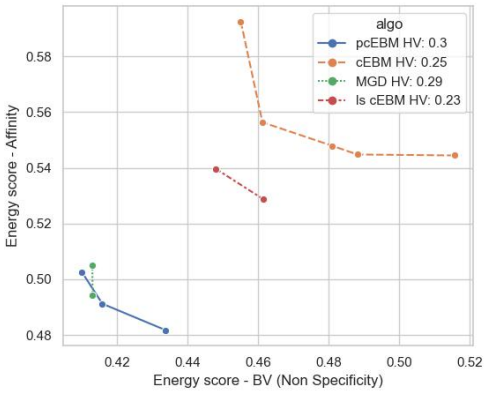


Figure 1: Pareto fronts and estimated hyper volume for all baselines with $\eta = 0.01$ and $k = 400$. The figure axes report the energy of EBMs that were separately trained on data with good affinity and BV scores, respectively, with smaller energies being more desirable.

We 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* η 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}) [27]. More details about the computation of these metrics can be found in the appendix.

Results and discussion. Our experiments are designed to provide quantitative 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 and Table 2 address **Q.1**. As perhaps expected, EBMs trained on data possessing a single property of interest propose sequences that are akin to those sharing the same property, but are dissimilar to the sequences preferred by other objectives. In contrary, 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 higher hyper-volume and similarity to the training datasets, whereas MGD is a close second.

[1] The abbreviation follows from the assay based on ELISA detection of non-specific binding to baculovirus particles [17].

	HV(Ab-Like, Aff, BV)	HV(Aff, BV)	HV(Ab-Like, BV)	HV(Ab-Like, Aff)
$\eta = 0.01$				
MGD	0.04	0.22	0.23	0.29
ls-cEBM	0.00	0.24	0.24	0.23
cEBM	0.00	0.25	0.24	0.25
pcEBM	**0.049**	**0.3**	**0.29**	**0.30**
$\eta = 1$				
MGD	0.04	0.25	0.25	0.27
ls-cEBM	0.00	0.25	0.24	0.25
cEBM	0.046	0.29	0.29	0.3
pcEBM	0.041	0.27	0.27	0.29
$\eta = 40$				
MGD	0.033	0.25	0.25	0.26
ls-cEBM	0.043	0.27	0.26	0.26
cEBM	0.046	0.29	0.29	0.29
pcEBM	0.037	0.27	0.27	0.28

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

Part of the difference between cEBM and pcEBM can be accounted by a difference in their speed of convergence. We confirm this in Figure 2 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**.

To further test our approach, we calculate the hyper-volume 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[2] 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 η is significantly smaller for pcEBM than others. If we turn to Figure 1 that depicts the Pareto front for two properties obtained from candidate sequences from each baseline, we see that the pcEBM front is more spread than MGD approaches, agreeing 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 that what is obtained by direct LD, which demonstrates the advantage of the multi-properly optimization aspect of pcEBM, addressing **Q.2** and **Q.3**.

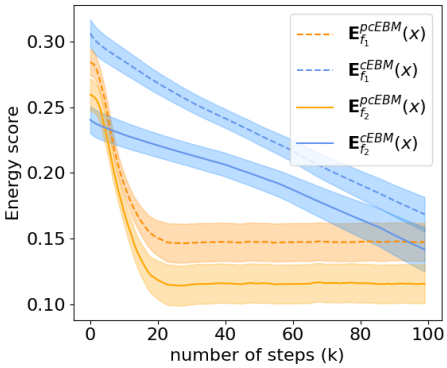
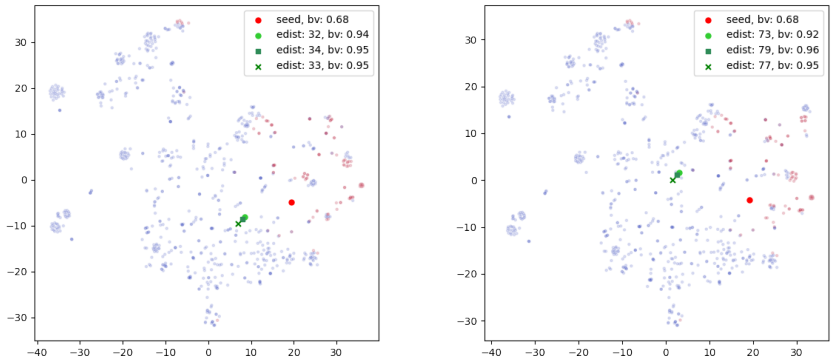


Figure 2: Energy scores per property f_i at each step for pcEBM and cEBM; same sequence and same step size $\eta = 0.01$.

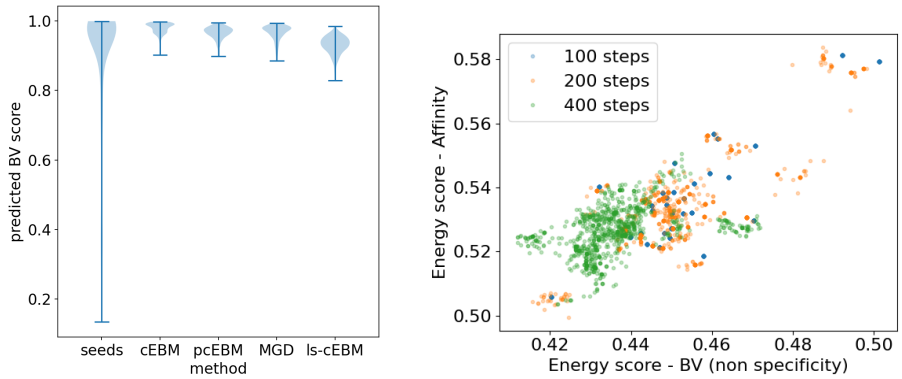
Finally, for **Q.4** we wish to explore the possibility for improving properties of already existing antibodies by using the variants of compositional EBMs. We focus our analysis on non-specificity (*BV*) as we have an external SeqCNN predictor (classifier) that can act as a surrogate pseudo-oracle. We start by screening the existing sequences to select only those with low BV scores, which we refer to as "seeds". We pass the selected seed sequences with poor measured (as well as predicted) scores, through our generative models, and then reuse the non-specificity pseudo-oracle to evaluate

[2]Though well suited to study the effect of multi-objective sampling, the energy score is biased since cEBM and ls-cEBM have the advantage of sampling and evaluating w.r.t. same model. A more fair comparison would be using external predictors as surrogates or wet-lab results to objectively evaluate.



(a) cEBM (b) pcEBM

Figure 3: 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.



(a) (b)

Figure 4: (a) 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 (b) Energy scores of trajectories proposed by pcEBM for a different number of steps. Lower energy scores are better/preferred for the objectives.

the newly proposed designs. Figure 3 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).

Figures 3 (a) and (b) provide further insight into the non-specificity 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 non-specificity 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 an example sequence (seed) with particularly bad BV score (i.e., sticky molecules) and its corresponding proposed designs. We see that both cEBM and pcEBM improve the probability for a good non-specifisity 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 [23], (*iii*) incorporating uncertainty estimates, and (*iv*) pareto-optimal training for EBMs, to name a few.

References

[1] M. Bailly, C. Mieczkowski, V. Juan, E. Metwally, D. Tomazela, J. Baker, M. Uchida, E. Kofman, F. Raoufi, S. Motlagh, et al. Predicting antibody developability profiles through early stage discovery screening. In *MAbs*, volume 12, page 1743053. Taylor & Francis, 2020.

[2] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[3] Y. Censor. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization*, 4(1):41–59, 1977.

[4] A. Chinchuluun and P. M. Pardalos. A survey of recent developments in multiobjective optimization. *Annals of Operations Research*, 154(1):29–50, 2007.

[5] J. Conard. How gpt-3 wrote a movie about a cockroach-ai love story. In *Wired*, 2022.

[6] D. Davis. Generative design is doomed to fail. In *Blog post*, 2020.

[7] J.-A. Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

[8] Y. Du, S. Li, and I. Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

[9] Y. Du and I. Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

[10] R. R. Eguchi, C. A. Choe, and P.-S. Huang. Ig-vae: Generative modeling of protein structure by direct 3d coordinate generation. *PLoS computational biology*, 18(6):e1010271, 2022.

[11] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.

[12] V. Gligorijevic, D. Berenberg, S. Ra, A. Watkins, S. Kelow, K. Cho, and R. Bonneau. Function-guided protein design by deep manifold sampling. *bioRxiv*, 2021.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[14] A. P. Guerreiro, C. M. Fonseca, and L. Paquete. The hypervolume indicator: Problems and algorithms. *arXiv preprint arXiv:2005.00515*, 2020.

[15] R. Hao. Deep generative models could offer the most promising developments in ai. In *Venture Beat*, 2022.

[16] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[17] I. Hötzel, F.-P. Theil, L. J. Bernstein, S. Prabhu, R. Deng, L. Quintana, J. Lutman, R. Sibia, P. Chan, D. Bumbaca, et al. A strategy for risk mitigation of antibodies with fast clearance. In *MAbs*, volume 4, pages 753–760. Taylor & Francis, 2012.

[18] M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pages 427–435. PMLR, 2013.

[19] S. Kaisa M. Miettinen, Sayin. Nonlinear multiobjective optimization. *European Journal of Operational Research*, 148(1):229–230, 2003.

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[22] Y. LeCun, S. Chopra, M. Ranzato, and F.-J. Huang. Energy-based models in document recognition and computer vision. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 337–341. IEEE, 2007.

[23] Q. Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.

[24] X. Liu, X. Tong, and Q. Liu. Profiling pareto front with multi-objective stein variational gradient descent. *Advances in Neural Information Processing Systems*, 34:14721–14733, 2021.

[25] A. Nigam, R. Pollice, G. Tom, K. Jorner, L. A. Thiede, A. Kundaje, and A. Aspuru-Guzik. Tartarus: A benchmarking platform for realistic and practical inverse molecular design. *arXiv preprint arXiv:2209.12487*, 2022.

[26] T. H. Olsen, F. Boyles, and C. M. Deane. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022.

[27] B. Paaßen, C. Gallicchio, A. Micheli, and B. Hammer. Tree edit distance learning via adaptive symbol embeddings. In *International Conference on Machine Learning*, pages 3976–3985. PMLR, 2018.

[28] D. Repecka, V. Jauniskis, L. Karpus, E. Rembeza, I. Rokaitis, J. Zrimec, S. Poviloniene, A. Laurynenas, S. Viknander, W. Abuajwa, et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333, 2021.

[29] K. Roose. We need to talk about how good a.i. is getting. In *The New York Times*, 2022.

[30] O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[31] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[32] M. Šošić and M. Šikić. Edlib: a c/c++ library for fast, exact sequence alignment using edit distance. *Bioinformatics*, 33(9):1394–1395, 2017.

[33] A.-M. Wolf Pérez, N. Lorenzen, M. Vendruscolo, and P. Sormanni. Assessment of therapeutic antibody developability by combinations of in vitro and in silico methods. In *Therapeutic Antibodies*, pages 57–113. Springer, 2022.

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 multipropery sampling baseline, the following hyper parameter grid was explored with random search:

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

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 $supx f_i(x) \leq r_i, \forall i \in [m]$. For a given set of solutions $\mathcal{X} = \{x_l\}_{l=1}^n$, a hypervolume indicator $HV(\mathcal{X})$ is a measure of the region between all f_i objectives and r:

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

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 $r = (1.0, 1.0, 1.0)$. We compte HV from each Pareto front to the reference point with the implementation provided in `PyMoo` [2].

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 [32].

4.3 Additional results from experiments

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

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.