Deep Uncertainty and the Search for Proteins

Zelda Mariet
zmariet@google.com

Ghassen Jerfel
ghassen@google.com

Zi Wang
wangzi@google.com

Christof Angermüller
christofa@google.com

David Belanger
dbelanger@google.com

Suhani Vora
svora@google.com

Maxwell Bileschi
mlbileschi@google.com

Lucy Colwell
lcolwell@google.com

D Sculley
dsculley@google.com

Dustin Tran
trandustin@google.com

Jasper Snoek
jsnoek@google.com

Abstract

Machine learning applications to molecule and protein design require models that provide meaningful uncertainty estimates. For example, Bayesian optimization for biomolecules searches through the space of viable designs, trading off the exploration of uncertain regions with the exploitation of high-value areas. We introduce protein optimization datasets as a benchmarking environment for ML uncertainty on real-world distribution shifts; investigate scalable models robust to the distribution shift inherent to large-batch, multi-round BO over protein space; and show that intra-ensemble *diversification* improves calibration on multi-round regression tasks, allowing for more principled biological compound design.

1 Introduction

Applications of machine learning to biological sequence design require *well-calibrated* models: models whose uncertainty estimates align with their error rates. For example, protein design leverages large-scale, batched Bayesian optimization (BO) [21, 4, 13, 26, 25, 14]. A labeled dataset of protein sequences is iteratively built up by learning a surrogate ML model to predict the desired property, then optimizing a function (*e.g.*, the upper confidence bound acquisition function [21]) of this model's predictions and uncertainty estimates to select the next batch of sequences. Throughout this optimization process, each new batch of sequences moves further *away* from the first batches on which the surrogate model was trained, creating an intrinsic dataset shift to which the surrogate model must remain robust.

Gaussian processes are the standard surrogate model for predictions and uncertainty estimates for multi-round optimization. However, modern bio-technology enables wet-labs to evaluate thousands of sequences in parallel; scaling GPs to such batch sizes require significant effort [23]. Yet, large batch sizes also open the door to powerful surrogate models such as deep neural networks.

Ensembles of deep nets have been shown to achieve state-of-the-art calibration results under dataset shift [12]. Recently, BatchEnsembles (BEs) [24] — ensembles of models that share a subset of their trainable parameters — have been shown to achieve high accuracy and calibration even when trained on subsets of data smaller than typical deep learning applications. As such, BEs are an appealing model choice for multiround biological design tasks, where only a few thousand labeled sequences are available during the first round(s). Furthermore, the performance of BE models is attributed [24] to the diversity of the ensemble. This motivates our investigatation of how this diversity can be improved to benefit calibration and predictions on biomolecule optimization tasks.

Related work. *ML for protein design.* Until recently, machine learning for protein design focused on the small batch regime in which BO using Gaussian processes (GPs) could be applied [21]; Yang

Machine Learning for Molecules Workshop at NeurIPS 2020. https://ml4molecules.github.io

et al. [25] explored protein space by choosing exploration constraints. In [4], a GP trained on a small subset of labeled sequences is used to identify membrane proteins with desirable properties. Recent work has focused on learning useful protein embeddings [1, 6, 20]. Protein optimization has been used to evaluate reinforcement learning algorithms [3]; recently, Angermueller et al. [2] investigated portfolio algorithms for increased method robustness across different biological optimization tasks.

Deep models and uncertainty. To represent aleatoric uncertainty [10], Bishop [5] introduced mixture density networks, which output means and standard deviations for a mixture of Gaussians. To represent uncertainty that stems from model choice, Bayesian neural nets employ a distribution over their parameters. Non-Bayesian methods include post-training temperature scaling [18] [8], bootstrapping [16], and ensembling [12]. Approximate Bayesian methods for deep nets have been developed for regression uncertainty, *e.g.*, [9, 7, 13, 27, 15]. In practice, predictive uncertainty can be estimated by computing a mean and variance from Monte-Carlo samples from the posterior.

In [17], the authors showed that *deep ensembles* [12] were among the most robust approaches for uncertainty estimates under distributional shift. Recently, Wen et al. [24] introduced BatchEnsembles (BEs), which use sets of rank-1 multiplicative factors to modulate the weights of a network, effectively creating a parameter-efficient ensemble.As protein design typically requires models trained with limited supervised data, BEs present an appealing choice of models.

2 Uncertainty in protein space

We focus on optimizing fixed-length protein sequences; hence, all protein sequences consist of sequences of ℓ tokens belonging to the amino acid vocabulary $\mathcal{V}$ ($|\mathcal{V}| = 20$).

2.1 Distribution shift in protein space

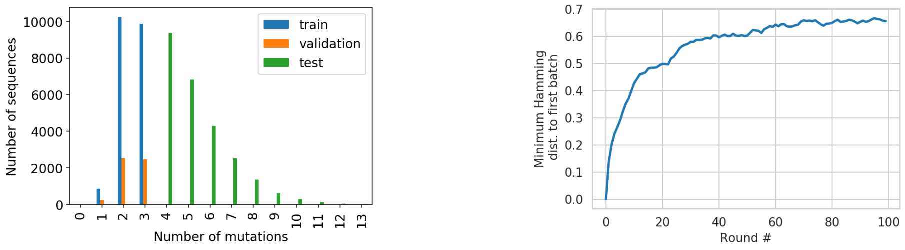
Protein datasets obtained via multi-round wet-lab experiments typically exhibit a dataset shift reflecting the sequential nature of the design decisions. For example, after a first round of experiments, the best protein seen so far may act as the *seed* for the next round of experiments. Figure 1a illustrates this shift on a protein dataset [19, 22] obtained by applying Error-Prone Polymerase Chain Reaction (EPPCR) to the *wildtype* Green Fluorescent Protein (GFP). Rao et al. [19] sliced this dataset into splits that correspond to increasing number of amino acid mutations away from the wildtype sequence.

Dataset shifts of similar nature occur when sequential experimental decisions are dictated by machine learning models; then, dataset shift depends upon the generalization ability and biases of the surrogate regressor. Figure 1b illustrates one such case: when optimizing the likelihood of a protein sequence under the Hidden Markov Model (HMM) that characterizes a protein family, the average Hamming distance between proposed sequences and the first batch increases over time.

2.2 Uncertainty in the regression setting

In the regression case, careful design choices are required to obtain a source of uncertainty. We focus on ensemble methods due to their relative simplicity coupled with strong observed performance across regression tasks [12] and under distributional shift [17].

Epistemic ensembles. Ensembles capture uncertainty over the parameters of the model, which is reflected in different predictions on held out data. We take the mean of the ensemble predictions as the predicted value, $\hat{y}$, and their standard deviation, $\hat{\sigma}$, reflects the uncertainty. Each member is trained to minimize the mean squared error between its prediction and the true target value y.



(a) Distribution shift in the GFP dataset [19]

(b) Distribution shift in the Pfam dataset

Figure 1: Distribution shift on protein datasets, on (a) the real-world GFP dataset and (b) the synthetic Pfam dataset obtained by multi-round *in-silico* optimization.

Epistemic + aleatoric ensembles. To capture aleatoric uncertainty [10], we ask each ensemble member to predict a mean and standard deviation for each sequence. Each model in the ensemble is trained to maximize the log probability of the true regression values under the model's local normal distribution. The ensemble prediction $\hat{y}$ is the mean of all predicted regression values; the standard deviation $\hat{\sigma}$ is aggregated as a mixture of Gaussians: $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(\sigma_i^2 + \hat{y}_i^2 - \hat{y}^2)$.

We found that ensembles that represent epistemic and aleatoric uncertainty were more robust at finding high-quality proteins (Table 1) than simple epistemic ensembles. Thus, we focus our BE analysis on models that also capture aleatoric uncertainty.

2.3 Evaluating model performance in the multi-round setting

The multi-round protein-design setting is defined by sequential experimental decisions made in the process of proposing new sequences. At round t, we have access to a total of $n = t \times b$ protein sequences $s_1, \dots, s_n \in \mathcal{V}^\ell$ with associated rewards $r_1, \dots r_n \in \mathbb{R}$, which are used to train a regressor $f : \mathcal{V}^\ell \to \mathbb{R}$. To propose the next batch of sequences, we must solve an inner acquisition problem, defined by (a) the acquisition function $\phi : \mathcal{V}^\ell \to \mathbb{R}$, which evaluates protein sequences based on the regressor's predicted value and uncertainty,[1] and by (b) a method to solve the inner-loop problem

$$\mathrm{argmax}_{s \in \mathcal{V}^\ell}\, \phi(s) = \mathrm{argmax}_{s \in \mathcal{V}^\ell}\, \phi\Big(\hat{y}(f(s)), \hat{\sigma}(f(s))\Big). \qquad (1)$$

Problem 1 is typically a combinatorial problem requiring local search approximations. As in [2], we keep the top b solutions $s_{t+1,1}, \dots, s_{t+1,b}$ to problem (1) to define a batch; this batch is then sent to the wet-lab for labeling, taking advantage of the large scale parallelism available in wet-labs.

Batch $t+1$, for which we obtain the values from the wet-lab, is the natural choice to evaluate surrogate model calibration. Crucially, when comparing different surrogate models, we are doing so on different $(t+1)$-batches, as the current surrogate regressor, which depends on the previous batch, selects the next batch of sequences.The negative log likelihood (NLL) on the next batch is defined as

$$-\tfrac{1}{|B^{(t+1)}|}\sum\nolimits_{s \in B^{(t+1)}} \log \Pr(r_s | f_y^{(t)}(s), f_\sigma^{(t)}(s)),$$

where $f^{(t)}$ is the regressor trained on batches $B_1, \dots, B_t$, and batch B_{t+1} is obtained by (most often approximately) solving (1) given $f^{(t)}$.

2.4 Diversifying BatchEnsembles with log determinants

The calibration of an ensemble is correlated with the diversity of models in the ensemble; this diversity is typically measured based on similarities between logit-space predictions from different ensemble members [24]. Achieving model diversity within parameterization space is an appealing alternative, raising the possibility of diverse models that are not forced to disagree on training data.

We propose to apply weight diversification to BatchEnsemble (BE) models. Recall that BE models consist of n models which also share a common weight matrix W; each model applies a local, rank-1 perturbation $W' = r \cdot s^\top$ to the shared weight matrix W to introduce variations between ensemble members. Vectors r and s are commonly referred to as *fast weights*.

To increase BE weight diversity, we add a regularization term that penalizes fast weights that are linear combinations of each other; this can easily be done by way of the determinant of the linear kernel over weights. Specifically, given fast weights vectors $r_1, \dots, r_n$ corresponding to the same layer in n different members of the BatchEnsemble, we add as a training penalty the following term:

$$R(r_1, \dots, r_n) = -\alpha \log\det \Big[K\left(r_i/\|r_i\|^2, r_j/\|r_j\|^2\right)_{i,j} \Big], \qquad (2)$$

where K is any kernel function applied to the fast weights, α the regularization coefficient, and the negative sign encourages larger determinants when minimizing the training loss. The fast weights are normalized before applying the kernel, as the regularization only aims to increase model diversity; without the normalization, penalizing the diagonal term would amount to negative weight decay.

To avoid harming the initial training phase, we chose to reweight this regularization by an exponential growth over the number of epochs, so that the importance given to diverse weights increases overtime. With a linear kernel K, regularization (2) guarantees that — within a layer — no fast weight can be written as a linear combination of fast weights of other ensemble members. Experimentally, linear kernels outperformed exponentiated quadratic kernels, and so we only report results on linear kernels.

[1] We use the upper confidence bound (UCB) acquisition function [11] with parameter $\beta = 1$.

3 Experimental results

To evaluate estimates of uncertainty in the multi-round protein design setting, we consider black-box optimization problems that simulate machine learning-aided protein design tasks.

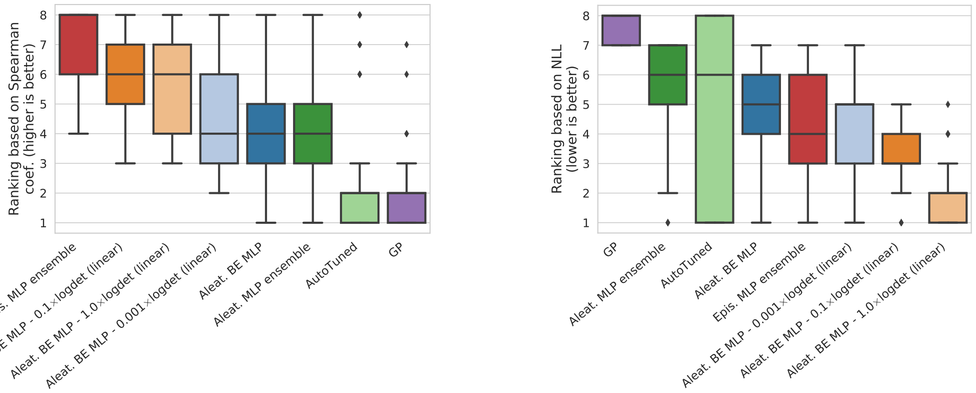
These problems fall into three categories: (a) maximizing the energy of protein contact Ising model; (b) maximizing the likelihood of a protein under an HMM characterizing a specific protein class; (c) maximizing cosine similarity to an unknown protein in embedding space. Each class contains problem instances corresponding to different protein targets. These problems were introduced in [3, 2], which include a more detailed problem description.

Given the amount of problem types and instances, we report results by ranking different regressors, based on their average performance across tasks.[2] We set the batch size to 500, in order to still be able to benchmark GP performance. We consider the following regressors:

Solver	PdbIsing	Pfam	Distance
Aleat. BE MLP - 0.1	10	7	9
Aleat. BE MLP	7	8	7
Aleat. BE MLP - 0.001	9	9	6
Aleat. BE MLP - 1.0	8	6	8
AutoTuned	11	5	11
Aleat. MLP ensemble	5	10	10
Epis. MLP ensemble	4	11	5
SingleMutant	3	3	3
GP	2	4	4
Evolution	6	2	2
Random	1	1	1

Table 1: Average rank of each regressor based on the best discovered protein (higher is better); BE models are robust, and diversity regularization slightly improves robustness.

- Epis. ensemble: BO + ensemble regressor; ensemble agreement acts as std. deviation.
- Aleat. ensemble: BO + ensemble regressor; members predict means and std. deviations.
- Aleat. BE: BO + BE regressor; ensemble members predict means and standard deviations.
- GP: BO + GP with RBF kernel over one-hot encodings and lengthscale set to the sequence length.
- AutoTuned [3]: BO + Bayesian ridge, lasso, and random forest regressors. At each step, each regressor is tuned with cross-validation. The best regressor(s) predict the next batch.
- SingleMutant: An algorithm that explores mutations near high-reward sequences [2].
- Evolution: A genetic algorithm that combines high-reward sequences [3].
- Random: Proposes sequences randomly.



(a) Regressor ranking based on correlation between true reward and predicted reward, averaged over all rounds and problems (higher is better).

(b) Ranking based on the regressor's negative log-likelihood on the next batch, averaged over all rounds and problems (lower is better).

Figure 2: Ranking of methods to predict protein function and uncertainty estimates on protein optimization tasks. Diversity penalties on BE models improve predictive ability throughout the entire optimization process, increasing both BE's ability to identify good sequences (a) and their calibration on the next batch (b).

Results are averaged over 5 trials; inner-loop problems are solved by SingleMutant for 10 steps; at each step all single mutants of the previous best sequence are explored. Ensembles are of size 10. Detailed optimization curves per-problem are provided in Appendix B (Figures 5 and 6).

Table 1 summarizes the performance of each regressor, based on the highest-reward sequence that BO is able to identify using the regressor as a surrogate model of protein fitness. BE regressors

[2] For consistency with [2], larger ranks correspond to metrics with higher values: a large rank (*e.g.*, 10) for sequence reward indicates good performance, and a low rank (*e.g.*, 1) indicates good performance for NLL.

are the most robust across different problem types; log-determinant penalties significantly improve the model's ability identify high-quality sequences. Figure 2 shows the ranking of each regressor according to their ability to rank the proposed sequences correctly as well as their calibration on the next batch. Adding a diversification penalty to BatchEnsembles significantly improves their calibration on the next batch (Fig. 2b), as well as their ability to detect high-quality sequences (Fig. 2a). This is partially due to improved accuracy on the next sequences (Fig. 3 in Appendix B), but not only — the epistemic ensemble achieves better MSE on the next batch, but worse calibration.

References

[1] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12): 1315–1322, 2019.

[2] Christof Angermueller, David Belanger, Andreea Gane, Zelda Mariet, David Dohan, Kevin Murphy, Lucy Colwell, and D Sculley. Population-based black-box optimization for biological sequence design. In *ICML 2020*. 2020.

[3] Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International Conference on Learning Representations*, 2020.

[4] Claire N. Bedbrook, Kevin K. Yang, Austin J. Rice, Viviana Gradinaru, and Frances H. Arnold. Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. *PLOS Computational Biology*, 13(10):1–21, 10 2017.

[5] Christopher M. Bishop. Mixture density networks. Technical report, 1994.

[6] Surojit Biswas, Grigory Khimulya, Ethan C. Alley, Kevin M. Esvelt, and George M. Church. Low-n protein engineering with data-efficient deep learning. *bioRxiv*, 2020. doi: 10.1101/2020.01.23.917682.

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

[8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[9] Jose Miguel Hernandez-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, 2015.

[10] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017.

[11] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6(1): 4–22, March 1985. ISSN 0196-8858. doi: 10.1016/0196-8858(85)90002-8.

[12] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017.

[13] Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In *International Conference on Machine Learning*, 2019.

[14] Ge Liu, Haoyang Zeng, Jonas Mueller, Brandon Carter, Ziheng Wang, Jonas Schilz, Geraldine Horny, Michael E Birnbaum, Stefan Ewert, and David K Gifford. Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics*, 36(7):2126–2133, 2020.

[15] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. 2019.

[16] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems 29*, pages 4026–4034. Curran Associates, Inc., 2016.

[17] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

[18] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[19] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems 32*, pages 9689–9701. Curran Associates, Inc., 2019.

[20] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 2020.

[21] Philip A. Romero, Andreas Krause, and Frances H. Arnold. Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201, 2013.

[22] S Karen Sarkisyan, A Dmitry Bolotin, V Margarita Meer, R Dinara Usmanova, S Alexander Mishin, V George Sharonov, N Dmitry Ivankov, G Nina Bozhanova, S Mikhail Baranov, Onuralp Soylemez, S Natalya Bogatyreva, K Peter Vlasov, S Evgeny Egorov, D Maria Logacheva, S Alexey Kondrashov, M Dmitry Chudakov, V Ekaterina Putintseva, Z Ilgar Mamedov, S Dan Tawfik, A Konstantin Lukyanov, and A Fyodor Kondrashov. Local fitness landscape of the green fluorescent protein. *Nature*, pages 397–401, 2016.

[23] Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. In *Advances in Neural Information Processing Systems 32*, pages 14648–14659. Curran Associates, Inc., 2019.

[24] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020.

[25] Kevin K. Yang, Yuxin Chen, Alycia Lee, and Yisong Yue. Batched stochastic bayesian optimization via combinatorial constraints design. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[26] Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019.

[27] Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, 2018.

A Hyperparameter ranges and implementation details

Model	Architecture
MLP	2 layers of 256 neurons
CNN	Conv1D(kernel=32, filter=10) →Conv1D(kernel = 64, filter = 10)

Table 2: Model architecture used in experiments.

For multiround experiments, we used ensembles of size 10. BE models used a random sign initialization of -0.25, a decay rate of 1.01 over 25 epochs. All models were trained using Adam default hyperparameters, a batch size of 50, over 25 epochs.

- When comparing to the GP baseline, we used a GP RBF kernel on one-hot encodings, with a lengthscale set to sequence length.
- Aleatoric uncertainty, when predicted, was predicted using 0.05 + sigmoid(previous layer).
- To avoid numerical unstability, log determinants were computed with an additional diagonal term $0.001 \cdot I$.

B Additional plots

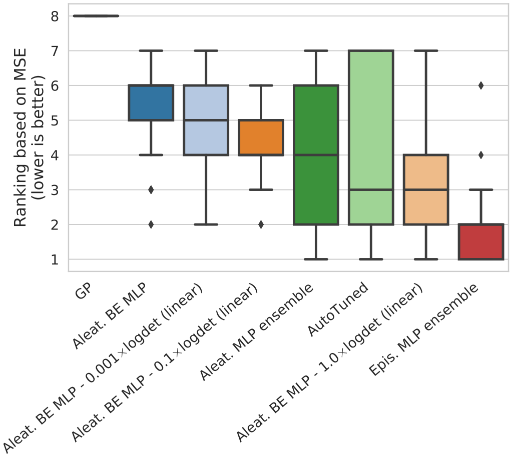
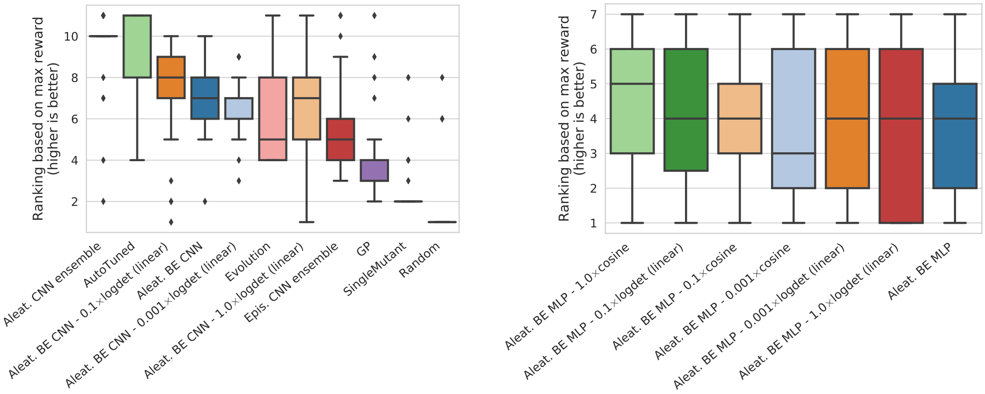


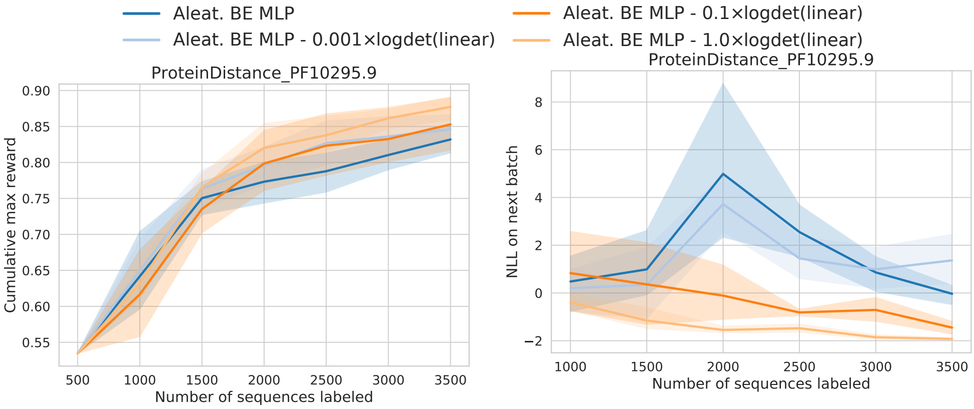
Figure 3: Multiround performance of regressors, based on their mean squared error on the next batch.



(a) Methods ranked by their impact on the highest sequence found during multiround optimization.

(b) Regularization methods ranked by the highest sequence found during multiround optimization.

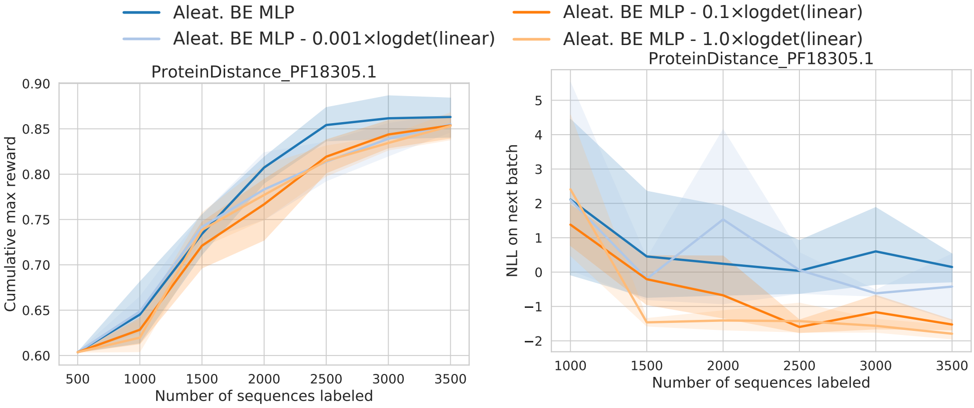
Figure 4: Multiround performance of BE regressors with different diversification penalties.



(a) Max reward over time on a specific protein target for different diversification strengths.

(b) Negative log-likelihood on the next batch for a specific protein target for different diversification strengths.

Figure 5: Multiround performance of BE regressors with different diversification penalties, on a protein target where diversification improves the search through protein space.



(a) Max reward over time on a specific protein target for different diversification strengths.

(b) Negative log-likelihood on the next batch for a specific protein target for different diversification strengths.

Figure 6: Multiround performance of BE regressors with different diversification penalties, on a protein target where diversification did not improve the search through protein space.

B.1 tSNE Visualization of proposed batches

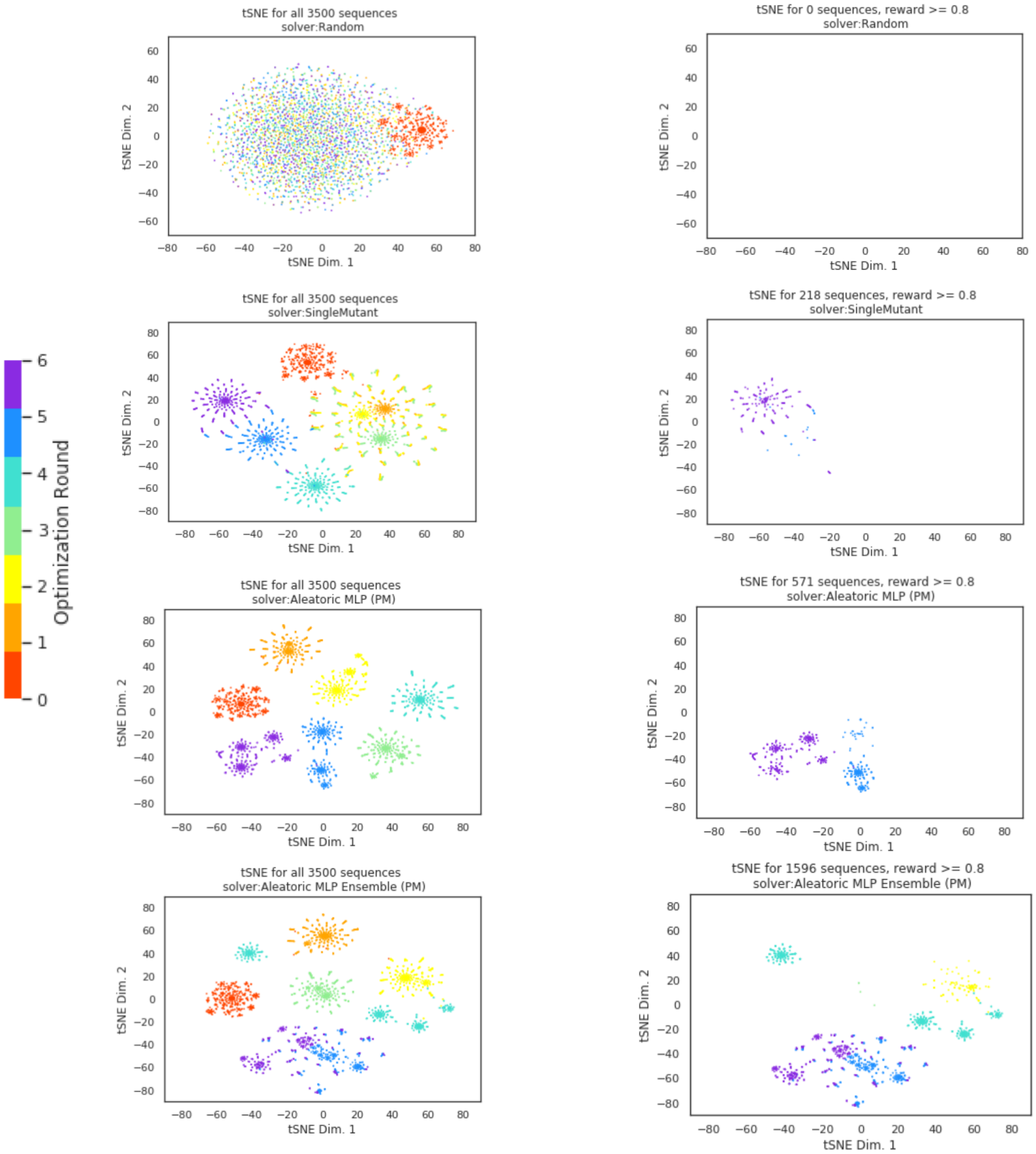


Figure 7: tSNE visualization of proposed batches: non-ensembling baselines (Rows 1 - 3) vs deep ensemble (Row 4). All figures are representative outcomes of applying the denoted solver to multiround optimization of the cosine similarity problem (Multiround protein task c.) tSNE for all sequences proposed by respective solver are shown on left, tSNE for only the subset of successful proposed sequences with reward >= 0.8 max reward are shown on right. Successive proposed batches are colorized ordinally according to round, as indicated by color bar.

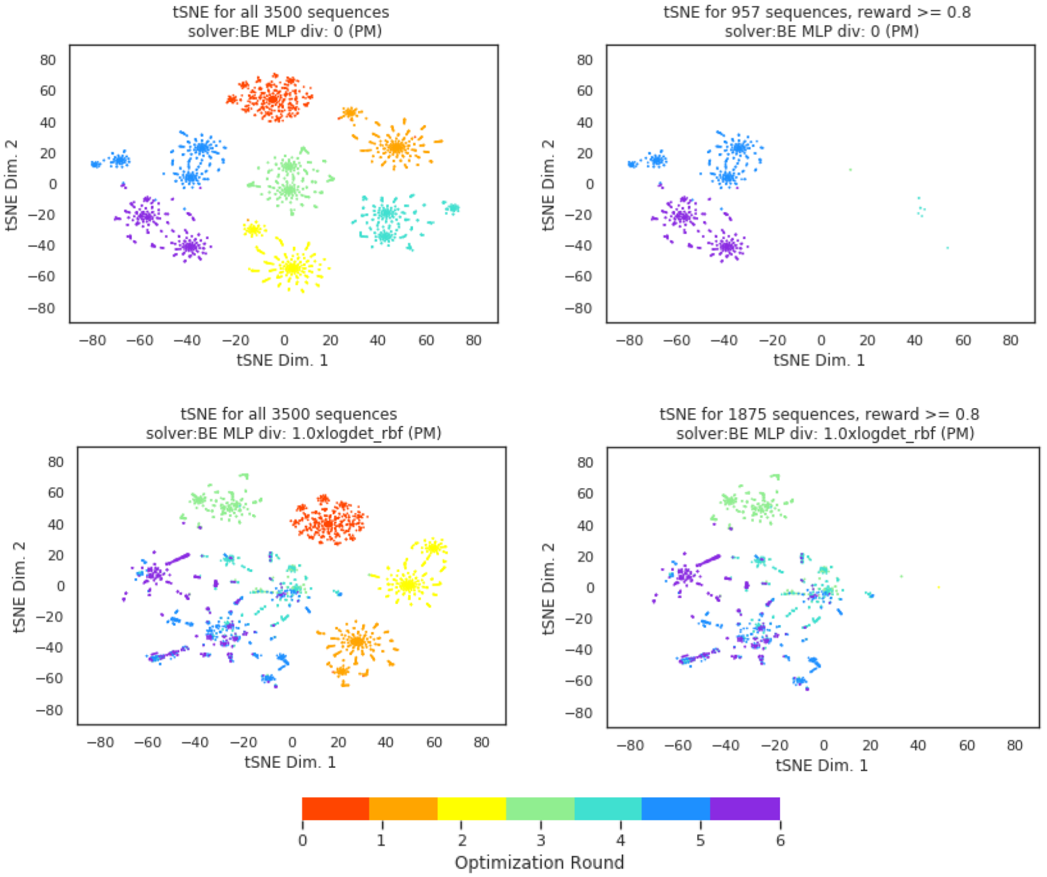


Figure 8: tSNE visualization of proposed batches: No diversity enforcement (top row) vs. diversity enforcement (bottom row). All figures are representative outcomes of applying the denoted solver to multiround optimization of the cosine similarity problem (Multiround protein task c.) tSNE for all sequences proposed by respective solver are shown on left, tSNE for only the subset of successful proposed sequences with reward >= 0.8 max reward are shown on right. Successive proposed batches are colorized ordinally according to round, as indicated by color bar.