Conditional Neural Processes for Molecules

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural processes (NPs) are models for transfer learning with properties reminiscent of Gaussian Processes (GPs). They are adept at modelling data consisting of few observations of many related functions on the same input space and are trained by minimizing a variational objective, which is computationally much less expensive than the Bayesian updating required by GPs. So far, most studies of NPs have focused on low-dimensional datasets which are not representative of realistic transfer learning tasks. Drug discovery is one application area that is characterized by datasets consisting of many chemical properties or functions which are sparsely observed, yet depend on shared features or representations of the molecular inputs. This paper applies the conditional neural process (CNP) to DOCKSTRING, a dataset of docking scores for benchmarking ML models. CNPs show competitive performance in few-shot learning tasks relative to supervised learning baselines common in QSAR modelling, as well as an alternative model for transfer learning based on pre-training and refining neural network regressors. We present a Bayesian optimization experiment which showcases the probabilistic nature of CNPs and discuss shortcomings of the model in uncertainty quantification.

1 Introduction

1.1 Learning from sparse chemical datasets

Recent years have seen an explosion of novel machine learning (ML) methods for virtual screening or de novo molecular design, often relying on large neural networks that require vast amounts of labelled data. The development of these models has been fueled by an expectation that ML will greatly accelerate drug discovery [1, 2]. Unfortunately, the real-world applicability of such models is hindered by the sparsity of chemical datasets, which comprise many molecular functions with only a few observations each. It is estimated that in-house datasets in the pharmaceutical industry are less than 1% complete, whereas ChEMBL is only 0.05% complete [3]. In order to take advantage of this information, we require models that are able to transfer information effectively across separate molecular functions, even when these are annotated on non-overlapping molecules. Neural processes are a novel family of models that show promise in this setting, but have so far only been tested on toy low-dimensional datasets. In this paper, we evaluate the performance of the CNP in several molecular tasks using high-dimensional molecular representations.

1.2 Conditional Neural Processes (CNPs)

Consider a dataset consisting of observations of real-valued functions $f_1, \dots, f_n$ on the same input space $\mathcal{X}$. Each function f_i is observed at a set of o_i input points $x_O^i \in \mathcal{X}^{o_i}$; we define $y_O^i = (f(x_{O,1}^i), \dots, f(x_{O,o_i}^i))$. Let f be a *test* function, (x_C, y_C) be a vector of c *context* points and the values of f on these inputs, and (x_T, y_T) be a vector of t *target* points and the values of f on these inputs. A neural process (NP) [4, 5] aims to describe the predictive distribution $p(y_T \mid x_T, x_C, y_C)$. This is done by mapping (x_C, y_C, x_T) through a parametric function, which is trained on the data $(x_O^i, y_O^i)_{i=1,\dots,n}$. In particular, we model the predictive distribution with a product measure:

$$q_\theta(y_T \mid x_T, x_C, y_C) = \prod_{j=1}^{t} \mathcal{N}(y_{T,j}; \mu_\theta(x_{T,j}), \sigma_\theta^2(x_{T,j})).$$

Submitted to the 6th Workshop on Meta-Learning at NeurIPS 2022, New Orleans. Do not distribute.

The mean and variance, $\mu_\theta(x)$ and $\sigma^2_\theta(x)$, of the predictive distribution at target input x are obtained through the following mapping:

$$\begin{aligned} r_j &= h_\theta(x_{C,j}, y_{C,j}) \qquad \text{for } j = 1, \ldots, c && \text{(encoding)} \\ r &= r_1 \oplus \cdots \oplus r_c && \text{(aggregation)} \\ (\mu_\theta(x), \sigma^2_\theta(x)) &= g_\theta(x, r) \qquad \text{for all } x \in \mathcal{X} && \text{(decoding)} \end{aligned}$$

where h_θ and g_θ are neural networks, $\oplus$ is a commutative operation and r is a global representation for the entire context data.This architecture ensures that the predictive distribution is invariant to permutations of the context and target points, respectively. The parameters θ of the encoder and decoder are trained by backpropagation using the data $(x^i_O, y^i_O)_{i=1,\ldots,n}$. Conditional NPs (CNPs) [4] minimise a particularly simple objective:

$$-\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \log q_\theta(y^i_T \mid x^i_T, x^i_C, y^i_C)\right],$$

where the expectation is taken with respect to a random partition of the observations (x^i_O, y^i_O) for function f_i into a set of context points (x^i_C, y^i_C) and target points (x^i_T, y^i_T). This objective function does not explicitly regularize the predictive distribution q_θ, so when the model is overparametrized, the objective can diverge and the variance parameters $\sigma^2(\cdot, r)$ can underestimate uncertainty. Latent NPs (LNPs) [5] avoid this problem by maximizing an approximate Evidence Lower Bound, derived through a more conventional variational inference approach.

So far NP models have been evaluated on low-dimensional settings, where they excel at few-shot learning. Here, we will analyze their performance on molecules represented by high-dimensional chemical fingerprints.

1.3 The DOCKSTRING dataset

The DOCKSTRING dataset [6] is a molecular dataset for benchmarking of ML models. It comprises more than 15 million docking scores for 58 protein targets and 260k molecules. Targets were chosen to be medically relevant and represent a variety of protein families, and molecules were curated from PubChem and ChEMBL to be representative of chemical series in drug discovery projects.

Each molecule in the DOCKSTRING dataset is annotated with all 58 protein target scores, which makes it especially suitable to design benchmark tasks in transfer learning. In this paper, we sample a small subset of it to evaluate regression and transfer learning by CNPs in the low-data regime.

2 Methods

2.1 Dataset and split

NPs are able to learn across different datapoints within the same function, and across different functions within the same input space. Therefore, the dataset was split across both the datapoint dimension and the function dimension. We refer to these splits as *dtrain*, *dtest* and *ftrain*, *ftest* respectively (Appendix A).

To emulate learning in a low-data regime, we took a small sample of the train and test sets defined in the DOCKSTRING package. The *dtrain* set consisted of 2500 molecules from the original train set, and the *dtest* set consisted of 2500 molecules from the original test set. DOCKSTRING original sets were split by clusters, which prevented data leakage from chemical analogues in *dtrain* and *dtest*. Our function split was derived from the DOCKSTRING regression task, using the 5 task targets (ESR2, KIT, PARP1, PGR, F2) as *ftest* and the other 53 targets as *ftrain*.

2.2 CNP and benchmark models

A simple CNP was implemented with 3 linear layers in the encoder network, a mean aggregator function and 3 linear layers in the decoder network. We include four benchmarks commonly used in QSAR studies: a feed-forward neural network with the same number of layers as the CNP (NN), k-nearest neighbours with $k = 5$ (KNN) and $k = 1$ (FSS, known as fingerprint similarity search in chemoinformatics [7]), and a random forest regressor with 200 estimators (RF). As these benchmarks are only trained on *ftest* functions, we include a further benchmark for transfer learning (fine-tuned NN). This consists of pre-training the previous NN with 53 outputs on *ftrain* observations, and fine-tuning the model on each *ftest* function. The input to all models were Morgan molecular fingerprints of radius 3 and length

1024. The CNP and NN models were implemented in Pytorch [8], and the rest were implemented in scikit-learn [9]. We make our code freely available under an MIT license [10].

3 Experiments

3.1 Probabilistic regression and calibration

The CNP is an overparameterized neural model that outputs a predictive distribution. Similarly to neural networks trained by maximum likelihood, the CNP is trained by maximizing the conditional probability of the target points given the context points. However, unlike most neural models, the CNP performs uncertainty quantification. Since the conditional probability of the target points could be made arbitrarily large by making the predicted variance smaller, the CNP is at risk of overfitting and producing unreliable uncertainty estimates.

To evaluate this phenomenon, we analyzed the regression performance and the conditional probability of CNP predictions as the number of training epochs increased (Figure 1, Appendix B). We observed that predictions on the training set *ftrain, dtrain* improved monotonically with training length, as expected. In contrast, predictions on the test set *ftest, dtest* first improved and then worsened dramatically, especially in terms of uncertainty estimates. Based on these results, we selected the CNP trained for 1000 epochs for subsequent experiments.

Figure 1: CNP performance on different data splits as training time increases. Error bars indicate variability across functions. Performance on the training functions *ftrain* was comparable of that previously reported for Vina docking scores [6]. The easiest test function in *ftest*, PARP1, is displayed as an example.

We hypothesize that the LNP, whose ELBO-like objective includes a KL regularization term [5], may be more robust to overfitting and degradation of its uncertainty estimates. Analysis of the LNP is left for future work.

3.2 Few-shot learning

We evaluated the performance of the CNP in few-shot learning and compared it against benchmarks popular in chemoinformatics (Figure 2, Appendix C). The CNP was trained on *ftrain, dtrain*, used context points in *ftest, dtrain* and was tested on the target points *ftest, dtest*. Other models were trained on *ftest, dtrain* and tested on *ftest, dtest*. In spite of the CNP not seeing the functions in *ftest* during training, it outperformed all other models in the low-data regime. This was the case even for the transfer learning benchmark, fine-tuned NN, which was pre-trained on *ftrain, dtrain* and fine-tuned on *ftest, dtrain*.

3.3 Generalization to unseen functions

In previous sections, we analyzed the ability of the CNP to generalize to unseen functions in *ftest*. However, *ftrain* and *ftest* were all part of the same class of Vina docking scores. The question remained whether similar generalization would be observed for molecular functions from very different classes. To investigate this, we created a new type of score that linearly combines docking scores with the quantitative estimate of drug-likeness (QED) (Appendix D) [11]. We trained CNP models either on plain scores or on plain scores and QED-modified scores, and tested either on plain scores or on QED-modified scores (Table 1). We observed that the CNP was unable to generalize to functions of different classes, but performance could be easily recovered by including functions from those classes in the training set.

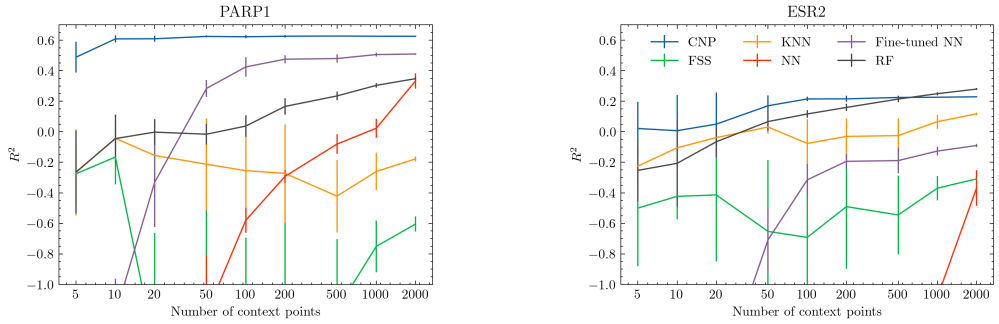

Figure 2: Few-shot performance of CNP and benchmark models on two *ftest* functions, an easy case (PARP1) and a hard case (ESR2). The x-axis shows the number of points in *ftest, dtrain* used as context by the CNP or as training points by the benchmarks. Benchmark error bars were derived from different training runs, whereas CNP error bars originated from using different context points.

Table 1: R^2 of CNP trained (**row**) and tested (**columns**) on plain and QED-modified scores from PARP1, KIT and F2.

	Plain scores	QED-modified scores
Plain scores	0.57 ± 0.08	-6.42 ± 3.14
Plain and QED-modified scores	0.54 ± 0.08	0.34 ± 0.03

3.4 Bayesian optimization with CNPs

Finally, we evaluated whether the CNP predictive distribution is useful for Bayesian optimization (BO). Plain scores or QED-modifed scores were minimized starting from five context molecules in *ftest, dtrain* ∪ *ftest, dtest*, selecting one molecule per iteration for 4995 iterations. Three acquisition strategies to select the next optimal molecule were compared: random, for a baseline; greedy, where only the mean of the distribution was considered; and lower confidence bound (LCB, $\beta = 1$), which attempted to benefit from uncertainty estimates (Figure 3, Appendix E). As expected, greedy and LCB greatly surpassed the random acquisition function. In addition, greedy and LCB exhibited very similar performance, which suggests that the uncertainty estimates of CNPs did not offer a competitive advantage for molecular optimization. However, both methods found the best molecule in the dataset quickly, making it difficult to draw strong conclusions. A more challenging optimization task within a larger molecular library is left for future work.

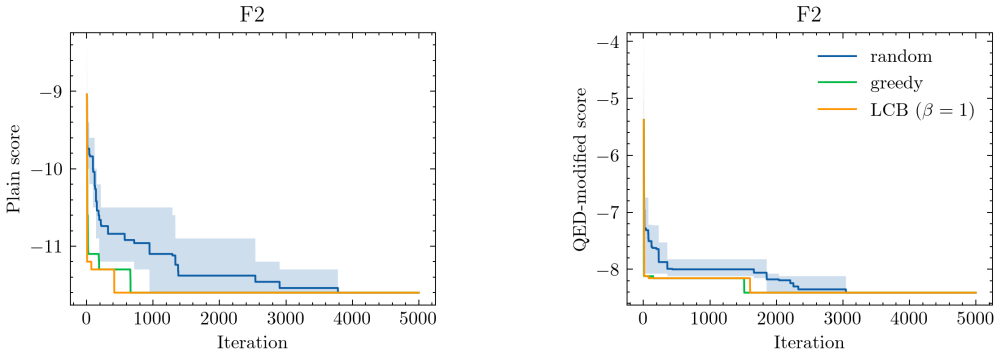

Figure 3: Bayesian optimization of plain and QED-modified scores of F2, a protein target in *ftest*.

4 Discussion

Our results demonstrate that CNPs have outstanding performance in few-shot learning of complex molecular properties such as docking scores. The application of CNPs to impute sparse chemical datasets could be highly impactful, even if one had confidence in only a small fraction of the imputations. However, the correct way to calibrate uncertainty estimates in CNPs is a question that requires further study, as is the potential to generalise to more diverse function classes. Applying similar models for probabilistic prediction, such as LNPs, in more complex imputation and molecular optimization tasks is an exciting area for future work.

References

[1] Andreas Bender and Isidro Cortés-Ciriano. Artificial intelligence in drug discovery: what is realistic, what are illusions? Part 1: Ways to make an impact, and why we are not there yet. *Drug Discovery Today*, 26(2):511–524, February 2021.

[2] Morgan Thomas, Andrew Boardman, Miguel Garcia-Ortegon, Hongbin Yang, Chris de Graaf, and Andreas Bender. Applications of Artificial Intelligence in Drug Design: Opportunities and Challenges. *Methods Mol. Biol.*, 2390(1-59.)::, 2022.

[3] Benedict W. J. Irwin, Julian R. Levell, Thomas M. Whitehead, Matthew D. Segall, and Gareth J. Conduit. Practical Applications of Deep Learning To Impute Heterogeneous Drug Discovery Data. *J. Chem. Inf. Model.*, 60(6):2848–2857, June 2020.

[4] Marta Garnelo, Dan Rosenbaum, Chris J. Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J. Rezende, and S. M. Ali Eslami. Conditional Neural Processes. *arXiv*, July 2018.

[5] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural Processes. *arXiv*, July 2018.

[6] Miguel García-Ortegón, Gregor N. C. Simm, Austin J. Tripp, José Miguel Hernández-Lobato, Andreas Bender, and Sergio Bacallado. DOCKSTRING: Easy Molecular Docking Yields Better Benchmarks for Ligand Design. *J. Chem. Inf. Model.*, 62(15):3486–3502, August 2022.

[7] Ingo Muegge and Prasenjit Mukherjee. An overview of molecular fingerprint similarity search in virtual screening. *Expert Opin. Drug Discovery*, 11(2):137–148, 2016.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv*, December 2019.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Anonymized repository for paper "Conditional neural processes for molecules". `https://drive.google.com/drive/folders/1Ol3qBTlopG-hDIrjjrLA1rpPHED7wJ-o?usp=sharing`.

[11] G. Richard Bickerton, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L. Hopkins. Quantifying the chemical beauty of drugs. *Nat. Chem.*, 4(2):90–98, February 2012.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] . For lack of space, given the 4-page limit.
   (d) Have you read the ethics review https://www.overleaf.com/project/6334a95068cee299c31e8d8c guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specified the most important details in the text, and include an open source library which contains code to reproduce all experiments.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Due to lack of space, and limited relevance.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] , this is included as a supplemental material and the license is specified.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The data used is open-sourced with a permissive licence.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] It doesn't.

Appendices

A Dataset split

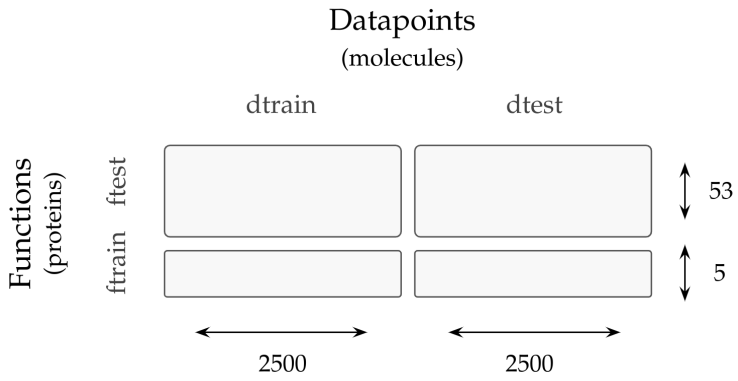

Figure A1: Split across the function and datapoint dimensions of the DOCKSTRING sample. *dtrain* was sampled from the DOCKSTRING training set and *dtest* was sampled from the DOCKSTRING test set, which were split by cluster similarity [6]. *ftrain* and *ftest* were derived from the DOCKSTRING regression tassk

B Probabilistic regression and calibration

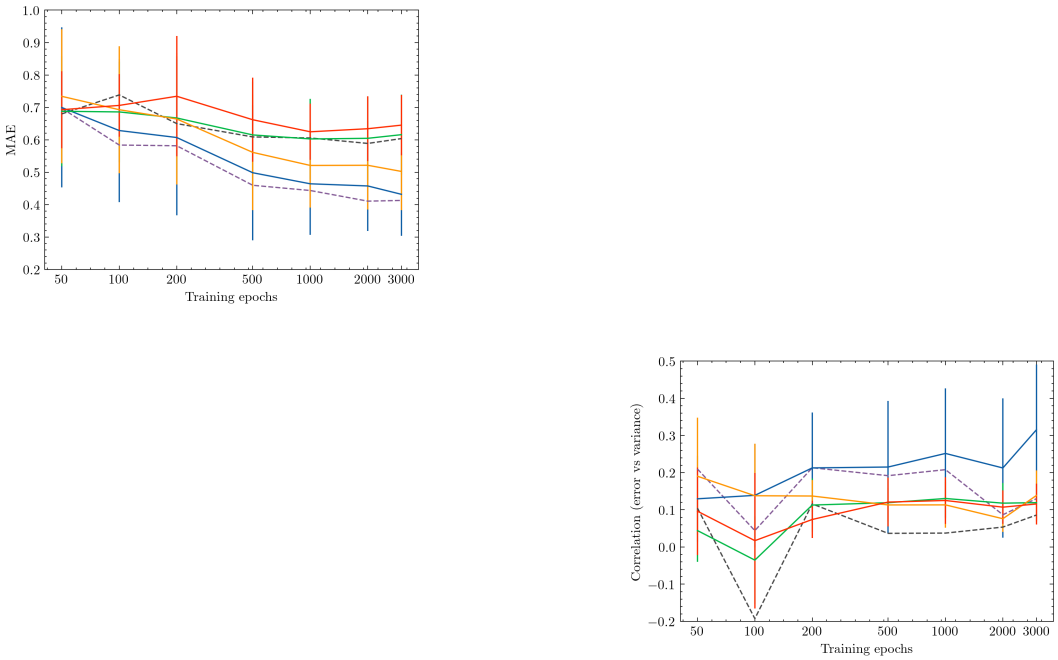

Figure A2: Performance of the CNP on the different data splits as the number of training epochs increases.

C Low data

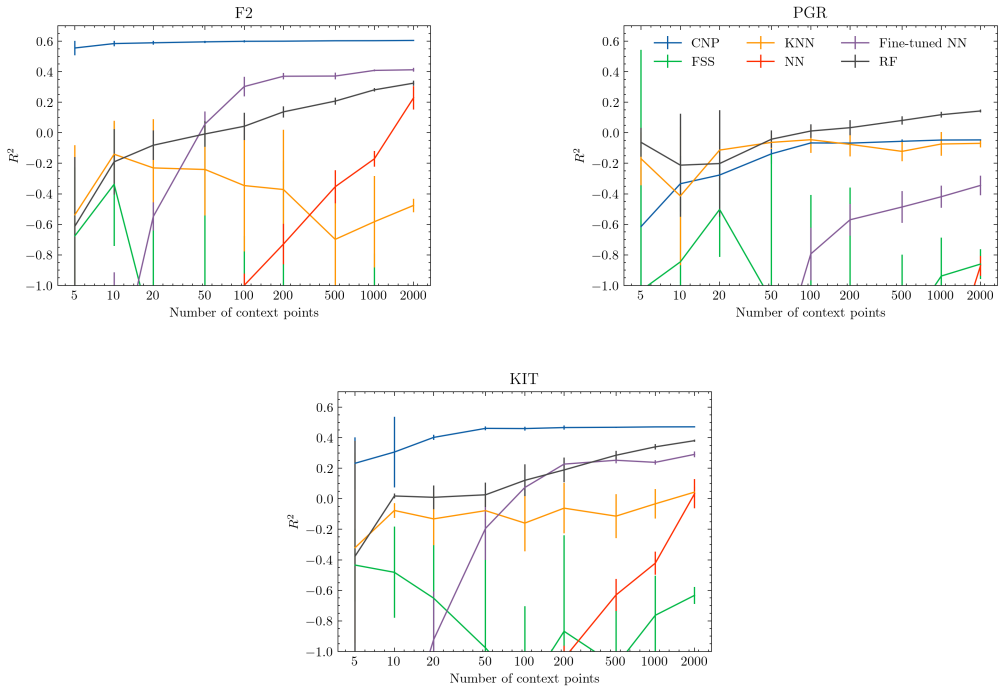

Figure A3: Few-shot performance of the CNP and benchmark models on three test functions in *ftest*, ranging in difficulty from easy (F2), medium (KIT), and hard (PGR). The x-axis shows the number of datapoints in *ftest, dtrain* used as context by the CNP or as training points by the benchmarks. Performance was evaluated on *ftest, dtest*.

D QED-modified docking scores

The quantitative estimate of drug-likeness (QED) of a molecule is a coefficient between 0 and 1 that attempts to quantify the molecule's similarity to approved drugs. In some experiments, we combine docking scores and QED values to create a new artificial score. QED-modified scores are expected to be more challenging to predict and to reflect drug-likeness.

Given a molecule m with docking score $s(m, t)$ for protein target t, we define its QED-modified score s' as

$$s'(m, t) := s(m, t) + 10(1 - \text{QED}(m)).$$