Robust task-specific adaption of models for drug-target interaction prediction

Emma Svensson [1] **Pieter-Jan Hoedt** [1] **Sepp Hochreiter** [1 2] **Günter Klambauer** [1]

[1] ELLIS Unit Linz & Institute for Machine Learning,
Johannes Kepler University Linz, Austria
{svensson, hoedt, hochreit, klambauer}@ml.jku.at
[2] Institute of Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria

Abstract

HyperNetworks have been established as an effective technique to achieve fast adaptation of parameters for neural networks. Recently, HyperNetworks conditioned on descriptors of tasks have improved multi-task generalization in various domains, such as personalized federated learning and neural architecture search. Especially powerful results were achieved in few- and zero-shot settings, attributed to the increased information sharing by the HyperNetwork. With the rise of new diseases fast discovery of drugs is needed which requires proteo-chemometric models that are able to generalize drug-target interaction predictions in low-data scenarios. State-of-the-art methods apply a few fully-connected layers to concatenated learned embeddings of the protein target and drug compound. In this work, we develop a task-conditioned HyperNetwork approach for the problem of predicting drug-target interactions in drug discovery. We show that when model parameters are predicted for the fully-connected layers processing the drug compound embedding, based on the protein target embedding, predictive performance can be improved over previous methods. Two additional components of our architecture, a) switching to L1 loss, and b) integrating a context module for proteins, further boost performance and robustness. On an established benchmark for proteo-chemometrics models, our architecture outperforms previous methods in all settings, including few- and zero-shot settings. In an ablation study, we analyze the importance of each of the components of our HyperNetwork approach.

1 Introduction

The use of HyperNetworks (Klein et al., 2015; Ha et al., 2017) to predict parameters for neural networks has become more established as a powerful alternative to traditional deep learning. While fast weight adaptation was originally proposed as an alternative to Recurrent Neural Networks (RNNs) (Schmidhuber, 1992), direct prediction has since been applied to a broader range of problems. These applications include neural architecture search (Brock et al., 2017; Zhang et al., 2018; Litany et al., 2022), continual learning (Von Oswald et al., 2019), differentiable pruning (Galanti & Wolf, 2020), and Bayesian inference in neural networks (Krueger et al., 2017). Recently, a HyperNetwork was successfully used to predict the parameters of unseen Convolutional Neural Networks (CNNs) architectures (Knyazev et al., 2021), able to reach a remarkable performance of 60% accuracy. In this case, the HyperNetwork is a Graph Neural Networks (GNNs) that takes the computational graph of the convolutional network as input and outputs parameters for the CNN. Several other application suggest that HyperNetworks equip their predicted networks with improved generalization and adaptation capabilities (Noh et al., 2016; Perez et al., 2018; Zhao et al., 2020; Muller, 2021; Shamsian et al., 2021; Knyazev et al., 2021).

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Potential for task-specific adaption of neural networks. The ability to adapt the parameters of another network using HyperNetworks has been investigated by Zhao et al. (2020) for image-based few-shot learning. In addition, some of the proposed meta-learning methods can also be understood as HyperNetworks (Hospedales et al., 2020). Ye & Ren (2021) use a HyperNetwork to generate task-specific adapters for language models from task descriptions. Other similar applications include a HyperNetwork semantic encoder that generates weights for a classifier (Baek et al., 2021) and task-conditioned generation of parameters in a healthcare setting (Ji & Marttinen, 2021). Overall, the mentioned research further motivates HyperNetworks as promising candidates for improving adaption capabilities of neural networks (Hospedales et al., 2020; Ye & Ren, 2021; Ji & Marttinen, 2021). An advantage that comes from the improved adaption is that the models tend to generalize better to new tasks, as seen in (Shamsian et al., 2021) with an application in personalized federated learning. Zhmoginov et al. (2022) show similar properties in their task-specific HyperTransformer for few-shot learning with small CNNs.

Importance of low-data tasks in drug discovery. A crucial part of drug discovery is to scan potential molecules as candidates and analyse their properties in relation to relevant protein targets, so called high-throughput screening (Hertzberg & Pope, 2000). With the rise of new diseases, there is a strong need for fast drug development (Muratov et al., 2021). However, due to the expensive and time-consuming experiments few data points are available creating a low-data problem (Guo et al., 2021) and current methods suffer from poor generalization to new proteins (Vamathevan et al., 2019). The low-data problem can be tackled with few-shot learning and meta-learning approaches (Guo et al., 2021; Jiang et al., 2021; Schimunek et al., 2021), or by incorporating descriptors of the target proteins, i.e. tasks, (Lenselink et al., 2017; Öztürk et al., 2018; Kim et al., 2020; 2021; Pentina & Clevert, 2022; Wang & Dokholyan, 2022). When task descriptors are included as input to the models, drug-target interactions can be predicted for zero-shot cases of unseen proteins. Nevertheless, current approaches mostly focus on improving protein or molecule representations while struggling in the zero-shot setting. Therefore, we explore new model architectures by using a HyperNetwork approach. A previous attempt was made to apply a HyperNetwork approach for molecular property prediction (Nachmani & Wolf, 2020). The HyperNetworks achieved state-of-the-art performance on a number of benchmarks but the approach was not set up or evaluated in the low-data or zero-shot settings.

Contributions. In this work, we adopt a HyperNetwork strategy to obtain parameters for models predicting drug-target interactions in an attempt to tackle low-data drug discovery. By utilizing the signal propagation theory from Chang et al. (2019) and by enriching the task embeddings with the context module proposed by Schimunek et al. (2022), we are able to generalize well to unseen targets, i.e. tasks, which has previously been impossible per design or too difficult due to shortage of training data. Our contributions are summarized accordingly and our source code is available at `github.com/ml-jku/hyper-dti`.

- We propose a novel architecture called HyperPCM, which uses a HyperNetwork to increase the adaptability of proteo-chemometric models by directly generating task-specific parameter predictions of the main network.
- We strengthen the robustness of our proposed architecture by enriching target embeddings using a context module (Schimunek et al., 2022) with a learned associative memory.
- We show that the predictive quality of models for drug-target interaction prediction can benefit from training on the continuous experimental data rather than binary labels.
- We demonstrate our method's effectiveness in a variety of settings of an established benchmark, achieving state-of-the-art performance across all settings including zero-shot cases.

2 Related work

The exercise to predict parameters has been studied in a multitude of application with varying methodology and structure. Originally, the idea to predict changes in a neural network's parameters was explored under the name of *fast weights* in (Schmidhuber, 1992). The changes are produced by a traditional feed-forward, *slow*, network using additive outer products for sequence processing with recurrent properties. By predicting changes rather than new values a short-term memory mechanism is achieved and the overall structure allows the main network to quickly refocus and adapt to new

contexts through distribution of attention. More recently, fast weights were used to avoid storing copies of neural activation patterns (Ba et al., 2016a). Parameter prediction has also been used to balance the concepts of weight sharing from RNNs and the generalizing capacity of CNNs. Ha et al. (2017) proposed a HyperNetwork architecture to achieve relaxed weight sharing by making layer-wise predictions of weights. Hence, the complexity of the trainable network is restricted, as a shared network makes the predictions, and the model's ability to generalize features was showcased by generating realistic-looking handwritten notes with Long Short-Term Memory (LSTM) units (Hochreiter & Schmidhuber, 1997).

Modeling drug-target interactions. Traditionally, drug-target interactions has been modeled by analysing Quantitative Structure-Activity Relationships (QSAR) (Hansch & Fujita, 1964). A drawback to QSAR modeling is that only the structure of the drug compound is used. Lapinsh et al. (2001) introduced the computational method of proteo-chemometric modeling (PCM) that also include information about the protein targets. While the QSAR method can be performed with separate binary models or single multi-class models, they cannot make predictions on protein targets not included in the training data, instead they are restricted to a finite set of protein targets per design. Despite the QSAR method's inability to generalize to unseen protein targets (Vamathevan et al., 2019), few-shot learning and meta-learning have been used in low-data settings of drug-target interaction prediction, e.g. (Guo et al., 2021; Jiang et al., 2021; Schimunek et al., 2021).

The PCM method is more flexible, as by including information about the target a trained model can handle unseen proteins. It is well documented that deep neural networks benefit from the multi-task set-up, where they are able to make use of knowledge learned about other targets (Unterthiner et al., 2014; Sosnin et al., 2018). Lenselink et al. (2017) compared single-task, multi-task and PCM models for drug-target interaction prediction and found PCM models to be further advantageous over multi-task ones. Different properties of drug-target interaction can be modeled, such as bioactivity or binding affinity. Across the two domains most PCM approaches let a fully-connected network (FCN) process concatenated embeddings of the drug compound and protein target respectively (Lenselink et al., 2017; Kim et al., 2020; 2021; Pentina & Clevert, 2022; Öztürk et al., 2018). State-of-the-art models are mainly produced with improved encoding strategies.

3 HyperNetwork-based proteo-chemometric modeling

We propose HyperPCM, an architecture that uses a HyperNetwork h to generate task-conditioned parameters $\boldsymbol{\theta}$ of a prediction model $f(\boldsymbol{x};\boldsymbol{\theta})$ with input $\boldsymbol{x} \in \mathcal{M}$. The HyperNetwork $h(\boldsymbol{t};\boldsymbol{\omega})$ takes as input a description of the task $\boldsymbol{t} \in \mathcal{T}$ and outputs the full list of parameters $\boldsymbol{\theta}$ for f. For applications in drug discovery, $\mathcal{T}$ can be a set of drug targets, such as proteins represented as amino-acid sequences (Lenselink et al., 2017) or assays (Vall et al., 2021) as strings, and $\mathcal{M}$ can be the space of potential drugs, i.e. small molecules represented as either SMILES strings (Weininger, 1988), 2D molecular graphs, or 3D conformations (Stärk et al., 2022). In both cases, the descriptors can be handcrafted or learned. Currently, unsupervised pre-training of encoders are prominent, with the most recent advancements coming from contrastive methods (Stärk et al., 2022; Sanchez-Fernandez et al., 2022).

The general formulation of the problem for a given training set $\{(\boldsymbol{x}^n, \boldsymbol{y}^n, \boldsymbol{t}^n)\}_{n=1}^N$ is defined as,

$$\arg\min_{\boldsymbol{\omega}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(\boldsymbol{y}^n, f(\boldsymbol{x}^n;\boldsymbol{\theta})), \text{ with } \boldsymbol{\theta} = h(\boldsymbol{t}^n;\boldsymbol{\omega}), \tag{1}$$

where $\mathcal{L}(.,.)$ is a differentiable loss function. Due to the differentiable nature of all functions f, h, and $\mathcal{L}$, the system can be trained end-to-end. Fig. 1 illustrates the overall set-up of our proposed learning approach, with the aim of modeling drug-target interaction. The architecture of the HyperNetwork depends on the nature of the task description. When a pretrained encoder is used to first get a vectorial embedding, the HyperNetwork can be an FCN. Additionally, Fig. 1 shows (a) the context module that we use to enrich the protein embeddings, and (b) the weight initialization strategy that we use for the last layer of the HyperNetwork.The remainder of this section explains these two concepts.

a) Robust task embeddings using a context module. Inspired by the idea that humans use associations to previously known information when they encounter unknown situations, HyperPCM comprises a context module (Schimunek et al., 2022; Fürst et al., 2022) to improve few-shot learning.

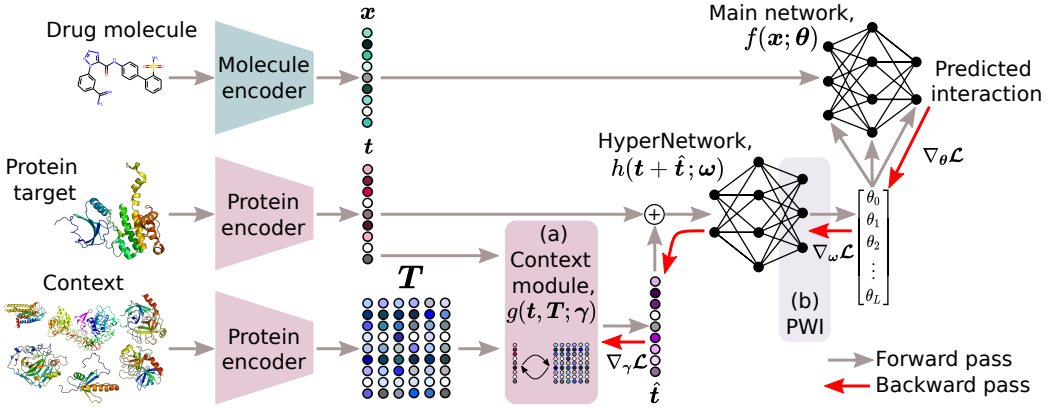

Figure 1: **HyperPCM.** Our approach to predict drug-target interactions, in which a HyperNetwork generates task-specific main models based on (a) context-enriched protein embeddings, and with (b) the last layer of the HyperNetwork initialized through Principled Weight Initialization (PWI).

The context module uses a continuous Modern Hopfield Network (MHN) (Ramsauer et al., 2021) to associate a given sample pattern with a much larger external memory of known patterns, referred to as the context. Schimunek et al. (2022) originally proposed the context module as a means to enrich embeddings of molecular compounds in a few-shot setting for activity prediction. In our method, the additional use of the protein information allows us to explore the context module on zero-shot cases by enriching the protein embeddings rather than the molecule embeddings. As such, we train the MHN to retrieve a new protein embedding from the larger context, that has amplified co-occurrences and covariate structures (Fürst et al., 2022). While Schimunek et al. (2022) additionally used the context module to smooth out co-occurrences between query and support set samples, our application of the MHN only associates the given sample with the context.

The MHN itself can be thought of as an associative memory that has similarities to the attention mechanism in Transformers (Vaswani et al., 2017). Given the context of all encoded training proteins $\boldsymbol{T} \in \mathbb{R}^{e\times T}$ and a state protein embedding $\boldsymbol{t} \in \mathbb{R}^{e}$, the MHN retrieves an updated embedding according to

$$\hat{\boldsymbol{t}} = g(\boldsymbol{t},\boldsymbol{T};\boldsymbol{\gamma}) = \boldsymbol{\Gamma}_{\mathrm{V}}\boldsymbol{T}\ \mathrm{softmax}\left(\beta(\boldsymbol{\Gamma}_{\mathrm{K}}\boldsymbol{T})^{T}(\boldsymbol{\Gamma}_{\mathrm{Q}}\boldsymbol{t})\right), \tag{2}$$

where β is a scaling factor and $\boldsymbol{\gamma}$ is the set of trainable parameters in the MHN, including $\boldsymbol{\Gamma}_{\mathrm{Q}}, \boldsymbol{\Gamma}_{\mathrm{K}}\boldsymbol{\Gamma}_{\mathrm{V}} \in \mathbb{R}^{e\times e}$ with respective analogies to the query (Q), key (K), value (V) concepts in Transformers attention (Widrich et al., 2020; Ramsauer et al., 2021). Like in previous work (Ramsauer et al., 2021; Schimunek et al., 2022), we normalize the state embedding, context embeddings and enriched embedding using LayerNorm (Ba et al., 2016b). We use the full set of protein embeddings from the training set as context, whereas Schimunek et al. (2022) rather sampled a subset of the training set. However, as any given protein would always appear in the context during training, we remove the current batch from the context to avoid overfitted association between the given protein and itself.

b) Stable signal propagation with Principled Weight Initialization (PWI). Initialization techniques have been introduced to ensure outputs of standard feed-forward networks retain the same distribution as their inputs (LeCun et al., 1998; Glorot & Bengio, 2010; He et al., 2015). Under the assumption that the inputs to a linear layer $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$ have i.i.d. features, such that $\forall i : \mathbb{E}[x_i] = 0$, then LeCun et al. (1998) found that if $\mathbb{E}[W_{ij}] = 0$ and $b_i = 0$ holds for all (i.i.d.) elements in $\boldsymbol{W}$ and $\boldsymbol{b}$ the distribution of output elements z_i follow

$$\mathbb{E}[z_i] = 0, \qquad \mathrm{Var}(z_i) = d_{\mathrm{in}}\,\mathrm{Var}(W_{ij})\,\mathrm{Var}(x_j) \quad \forall i,j, \tag{3}$$

where d_{in} is the input dimension, or *fan-in*, to the given layer. Expectations $\mathbb{E}$ and variances Var are taken across the random variables $x_j \sim p_{\mathrm{data}}$ and $W_{ij} \sim p_{\boldsymbol{\theta}}$, which are independent at initialization. By scaling each weight element to $\mathrm{Var}(W_{ij}) = \frac{1}{d_{\mathrm{in}}}$, stable variance propagation can be achieved in the forward pass. To account for the effects of non-linear activations, such as ReLU, the initial weights can be scaled by a factor $\sqrt{2}$ (He et al., 2015).

In the case of our fully-connected HyperNetwork set-up, we do not necessarily want to maintain variance as the outputs are themselves used as parameters in the main network. Instead, the HyperNetwork should produce parameters that keep distributions invariant between input and outputs of the main network. As such, Chang et al. (2019) propose Principled Weight Initialization (PWI) that ensures predicted parameters start in a suitable range for each layer in the main network.

Let $\tilde{h}(\boldsymbol{t}; \tilde{\boldsymbol{\omega}})$ denote all layers of the HyperNetwork up to the last one and consider a HyperNetwork *head* to be a part of the last layer leading to a given subset of the output parameter vector $\boldsymbol{\theta}$. As such, we describe one of the HyperNetwork heads that produces parameters for $\boldsymbol{W}$ in the l^{th} layer of the main network as $\boldsymbol{\theta}^{\text{w}} = \boldsymbol{H}\tilde{h}(\boldsymbol{t}; \tilde{\boldsymbol{\omega}}) + \boldsymbol{o}$. Given that the aforementioned assumptions hold for all layers of the HyperNetwork, each output $\theta_{i'}^{\text{w}}$ should have zero mean and variance $c_{\text{in}} \operatorname{Var}(H_{i'k}) \operatorname{Var}(t_k)$, where c_{in} is the input dimension to the last layer of the HyperNetwork. Note that we use i' to refer to the entry representing the value for W_{ij}. Because each output corresponds to a weight in the main network, the results can be directly plugged into eq. (3) to achieve

$$\mathbb{E}[z_i] = 0, \qquad \operatorname{Var}(z_i) = d_{\text{in}}\, c_{\text{in}} \operatorname{Var}(H_{i'k}) \operatorname{Var}(t_k) \operatorname{Var}(x_j) \quad \forall i, i', j, k. \tag{4}$$

Similarly, another HyperNetwork head can be used to predict the bias parameters $\boldsymbol{b}$ in the l^{th} layer of the main network according to $\boldsymbol{\theta}^{\text{b}} = \boldsymbol{G}\tilde{h}(\boldsymbol{t}; \tilde{\boldsymbol{\omega}}) + \boldsymbol{u}$. Based on the same assumptions, these predicted parameters θ_i^{b} have zero mean, but a variance of $c_{\text{in}} \operatorname{Var}(G_{ik}) \operatorname{Var}(t_k)$. Due to the linearity of the variance, we get an increased variance for z_i in layer l of

$$\operatorname{Var}(z_i) = d_{\text{in}}\, c_{\text{in}} \operatorname{Var}(H_{i'k}) \operatorname{Var}(t_k) \operatorname{Var}(x_j) + c_{\text{in}} \operatorname{Var}(G_{ik}) \operatorname{Var}(t_k) \quad \forall i, i', j, k. \tag{5}$$

Chang et al. (2019) use the result in eq. (5) to conclude that a stable propagation in the main network, i.e. $\operatorname{Var}(z_i) = \operatorname{Var}(x_j)$, can be achieved by setting

$$\operatorname{Var}(H_{i'k}) = \frac{1}{2\, d_{\text{in}}\, c_{\text{in}} \operatorname{Var}(t_k)}, \qquad \operatorname{Var}(G_{ik}) = \frac{1}{2\, c_{\text{in}} \operatorname{Var}(t_k)}, \quad \forall i, i', k, \tag{6}$$

which effectively shares the variance between weights and biases. Our HyperNetwork has separate heads for both weights and biases of each layer in the main network, all of which follow PWI.

4 Experiments

We analyze the performance of our proposed learning approach on a PCM benchmark derived from the ChEMBL database in Lenselink et al. (2017). The dataset includes 204,017 small molecular drug compounds and 1,226 protein targets, together making up 314,707 experimentally tested interactions. The learning objective of the benchmark is to classify the drug-target interactions as *active* or *inactive*. These classes are based on a threshold of 6.5 log affinity, chosen to achieve a balanced dataset.

Data splitting. Evaluation is performed through 10-fold cross-validation on four splitting strategies, based on the criteria of: 1) *randomly* distributed drug-target pairs, 2) *temporal* entry of unique drugs compounds, 3) k-means clustering of 256-bit Morgan fingerprint (Morgan, 1965) representations of the drug compounds referred to as Leave-cluster-compound-out (LCCO), and 4) randomly distributed unique protein targets, referred to as Leave-protein-out (LPO). In the temporal case, instances are placed in the train/valid/test set based on the year of entry, therefore cross-validation is not appropriate and instead 10 reruns were made with new random seeds for evaluation of model variability.

Fig. 2 illustrates the difference between many-, few-, and zero-shot learning in the setting of drug-target interaction prediction. In our evaluation, splitting strategies 2) and 3) are examples where drug compounds are held out for validation and testing. While these strategies test generalization to unseen drugs and can include some few-shot cases, it does not evaluate model performance for zero-shot predictions. Strategy 4) on the other hand, is an example of when protein targets are held out, which exclusively validates and tests on zero-shot interactions. The random split, 1), can have overlap in both drug compounds and protein targets between the sets and thus contains mostly many-shot cases.

Descriptor encoding. In order to evaluate solely our new learning approach, we choose to reuse the best performing, pretrained encoders from the baseline models without further fine-tuning or adaption. Kim et al. (2021) considered two language models pretrained on a large set of molecules to encode their SMILES strings, the recurrent autoencoder Continuous and Data Driven Descriptors

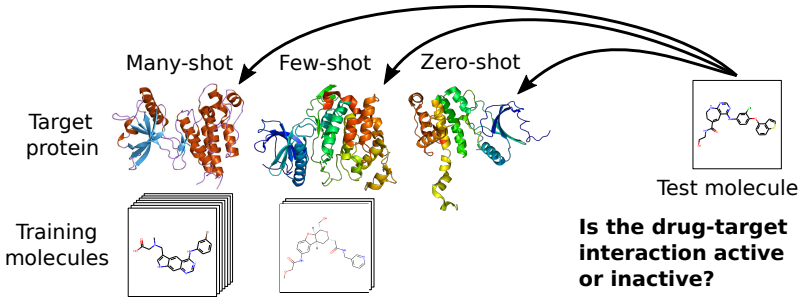

Figure 2: **Problem setting.** Illustrating the difference between many-, few-, and zero-shot test cases, based on the number of other interactions seen for the given protein target during training.

(CDDD) from Winter et al. (2019) and the Transformer model MolBert from Fabian et al. (2020). Regarding encoding protein targets, Kim et al. (2021) evaluated two other language models trained on the amino-acid sequences of a large set of proteins, UniRep (Alley et al., 2019) and SeqVec (Heinzinger et al., 2019). We extend the evaluation to also include two Transformer-based approaches for protein encoding, the ProtBERT and ProtT5 models (Elnaggar et al., 2020), recently collected in the `bio_embedding` framework (Dallago et al., 2021). Stärk et al. (2021) provide a more in depth analysis and comparison between these encoders. Based on the result in previous work, we pick CDDD and SeqVec for our setup as they gave the best performance in the zero-shot setting.

Training details. Our parameter prediction is implemented in PyTorch (Paszke et al., 2019), and consistently distributed into the main network through reshaping. Back-propagation is done end-to-end using the Adam optimizer (Kingma & Ba, 2014). A decaying learning rate, based on plateauing validation loss, is employed and models are trained until convergence with early stopping in terms of validation Matthews Correlation Coefficient (MCC) between predicted and ground-truth activity. Regarding the architecture of the FCN making up the trainable part of the HyperNetwork, we consider number of layers, size of the hidden dimension, and dropout rate to be hyperparameters optimized as part of the model selection.

Batching of the data is non-trivial in a HyperNetwork learning setup. Inspired by the procedure from Knyazev et al. (2021) we sample batches of both protein targets, referred to as meta-batches, and of drug compounds, referred to as mini-batches, respectively. However, additional measures are needed due to the large variation in compounds paired with unique targets. We perform an oversampling of underrepresented proteins to enforce a minimum threshold of compound samples. We also sample a fixed number of compounds per protein, by sampling with replacement in the rare cases that fewer compounds are available. Note that these sampling strategies only apply during loading of training examples, in order to not disturb the distribution during testing. Varying batch sizes were explored as part of the model selection, independently for the meta- and mini-batches.

Further details about our model selection and hyperparameter optimization can be found in App. A.

Loss function. While previous work on the benchmark trained in a binary classification setting, using the binary cross-entropy loss, we additionally explore using the provided continuous log affinity labels. Our hypothesis is that the real bioactivities might contain valuable information that is otherwise lost with the arbitrary threshold imposed in the benchmark. App. B further motivates the choice of loss function, concluding that the L1 loss, also known as the Mean Absolute Error (MAE), should be a suitable option. In order to compare results to the previous models, we apply the fixed threshold from the benchmark after the training solely for evaluation purposes.

4.1 Benchmarking

Previous work on the PCM benchmark (Lenselink et al., 2017; Kim et al., 2020; 2021), has produced a number of baselines with traditional machine learning and deep learning methods on concatenated drug and target descriptors. Picking out the leading results, we compare our novel approach to two deep FCNs, DNN_PCM (Lenselink et al., 2017) and DeepPCM (Kim et al., 2021), applied to handcrafted as well as unsupervised, learned descriptors. Also, from Lenselink et al. (2017) we

Table 1: **Matthews Correlation Coefficient (MCC).** From 10-fold cross-validation in the random, LCCO, and LPO settings and 10 reruns in the temporal setting. Best performance per setting is marked in bold and best baseline performance is marked in italic.

		FEW-/MANY-SHOT			**ZERO-SHOT**
MODEL	DESCRIPTORS	RANDOM	TEMPORAL	LCCO	LPO
RF[†]	HANDCRAFTED	*0.670*	0.210	N/A	N/A
DNN_PCM[†]	HANDCRAFTED	0.610	0.330	N/A	N/A
DEEPPCM[‡]	MOLBERT + UNIREP	0.654±0.005	*0.370±0.008*[*]	*0.505±0.053*	0.312±0.024
	MOLBERT + PROTBERT	0.625±0.006[*]	0.362±0.006[*]	0.455±0.054[*]	0.299±0.043[*]
	MOLBERT + PROTT5	0.620±0.004[*]	0.360±0.003[*]	0.452±0.057[*]	0.296±0.040[*]
	MOLBERT + SEQVEC	0.639±0.003[*]	*0.370±0.006*[*]	0.487±0.062	0.311±0.035
	CDDD + UNIREP	0.630±0.008	0.352±0.009[*]	0.490±0.061	0.307±0.031
	CDDD + PROTBERT	0.635±0.004[*]	0.343±0.006[*]	0.462±0.060[*]	0.294±0.049[*]
	CDDD + PROTT5	0.634±0.005[*]	0.353±0.006[*]	0.460±0.056[*]	0.310±0.054[*]
	CDDD + SEQVEC	0.643±0.005[*]	0.363±0.006[*]	0.478±0.048[*]	*0.322±0.028*
HYPERPCM	CDDD + SEQVEC	**0.682±0.039**	**0.395±0.005**	**0.532±0.059**	**0.340±0.051**

[*] RE-IMPLEMENTED [†] (LENSELINK ET AL., 2017) [‡] (KIM ET AL., 2021) N/A NOT AVAILABLE
LCCO: LEAVE-CLUSTER-COMPOUND-OUT LPO: LEAVE-PROTEIN-OUT

include a random forest multi-class model. The results using handcrafted descriptors are reiterated from the respective reference work, but all other baselines are re-implemented and rerun on our new splits. Table 1 presents average Matthews Correlation Coefficient (MCC) scores across ten test sets of a 10-fold cross-validation in the random, LCCO, and LPO settings, or reruns in the temporal case. The highest performance from either referenced work or our reruns are presented with improved results marked with asterisks. App. C.1 presents the full result from only our re-implementations, both in terms of MCC and AUC. Our method HyperPCM achieves the highest average score in each setting, marked in bold, and the best performing baseline for each setting is marked in italic.

We test the statistical significance of our model's improvement over previous approaches, by performing paired Wilcoxon signed-rank tests where applicable. The alternative hypothesis is that our model's mean MCC score is greater than the score of 1) the best performing baselines from Table 1, and 2) the most similar baseline DeepPCM using the same descriptors, CDDD and SeqVec. For the cases when the original result from Kim et al. (2021) was higher than our re-implementation (no asterisk in Table 1), we instead ran unpaired, Wilcoxon rank-sum tests because of the non-identical cross-validation splits.

In the random setting, none of the previous deep learning approaches beat the random forest model. Our approach on the other hand does improve slightly over the random forest model, although significance cannot be tested as only a single run is presented in Lenselink et al. (2017). When compared to the DeepPCM model using the same descriptors, the improvement is more significant ($p = 0.010$, paired). Similarly, in the LCCO cross-validation our model significantly outperforms the most similar baseline ($p = 0.001$, paired) while only seemingly improving over the best performing baseline ($p = 0.113$, unpaired). The results in the temporal setting vary less as they present reruns on a single test set and our model significantly improves over all included baselines with $p < 0.001$ (paired) for all cases. In the zero-shot setting our model marginally improves over the most similar and best performing baseline, but the result is not significant ($p = 0.224$, unpaired).

Nevertheless, it is important to note that the statistical test are not entirely fair in the cases where we could not reproduce the results presented in Kim et al. (2021), e.g. DeepPCM with MolBert and UniRep in the LCCO setting and DeepPCM with CDDD and SeqVec in the LPO setting. In these cases the data splits are not identical, as the exact data splits used were not documented. We split the data using the same procure described in the previous work. Our own re-implementation of DeepPCM with CDDD and SeqVec achieve an MCC score of 0.316 ± 0.043 in the LPO setting, which is still higher then all other baselines, and the improvement over this result with our HyperPCM model can be considered statistically significant ($p = 0.024$, paired). In App. C.2 we present an extended analysis of a subset of the LPO split where only molecules that were not seen during training are used in the test sets. The performance of our method in the extended setting is 0.313 ± 0.058 compared to 0.281 ± 0.051 for the best baseline, which is a significant improvement ($p = 0.002$, paired).

4.2 Ablation study

Further, we conduct an ablation study to analyze the importance of each of our three methodological contributions. Fig. 3 presents performance in terms of MCC on a fixed train/test split from the zero-shot setting, over ten reruns with different random seeds. Both the baseline DeepPCM and our model HyperPCM respectively achieve improved performance when training with L1 loss compared to when trained on the binary active/inactive labels, as done in the previous work (Lenselink et al., 2017; Kim et al., 2020; 2021). The learning curves shown in App. C.3 suggest that the models overfit in the classification setting. Moreover, Fig. 3 shows that performance gains are generally achieved with our proposed HyperNetwork setup, both in the classification and regression setting. Lastly, the right most panel of Fig. 3 illustrates the effect of adding the context module to enrich protein target embeddings in the baseline model vs in our HyperNetwork model. The increased variance in performance should be explored further in future work. Regardless, the average results are slightly increased for the HyperNetwork setup when context-enriched embeddings are used.

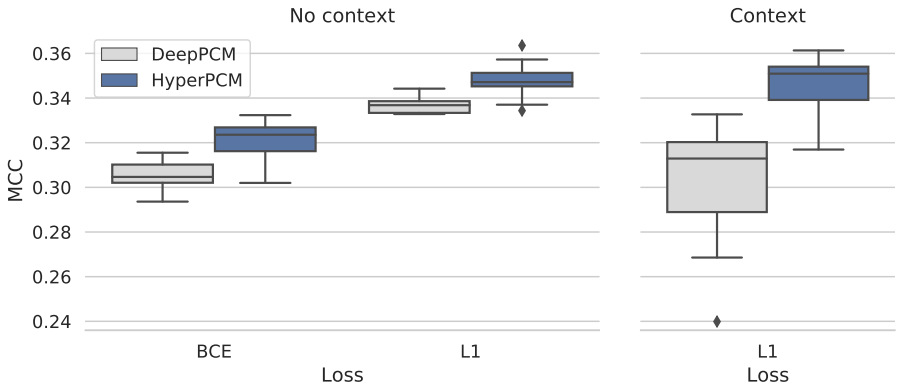

Figure 3: **Ablation study.** Comparing the baseline DeepPCM in two different training settings and the additional improvement with our proposed HyperNetwork setup, with or without context-enriched protein embeddings. Showing MCC across ten reruns on a fixed data split.

5 Conclusion

To conclude, we propose HyperPCM, a HyperNetwork-based architecture to directly generate task-conditioned parameters of a drug-target interaction model to improve its adaptation abilities. The use of PWI in the HyperNetwork stabilizes the otherwise unstable signal propagation. By enriching the task embeddings as a first step in the HyperNetwork, we further strengthen the given models robustness to new data. The enriched embeddings are achieved with an MHN that learns an associative memory with amplified co-occurrences and covariate structures. We demonstrate our methods effectiveness on an established PCM benchmark, reaching significant improvements in a number of settings, as well as some improvement on zero-shot cases. While previous work did not consider cases of unseen drugs and targets, we further analyze our models abilities in these cases.

A limiting element of our proposed learning method, is the computationally heavy and slow training procedure compared to previous work. However, when zero-shot inference on new targets is the goal our method extends the expressive power of previous methods with more learnable parameters while decreasing the computational cost of inference once the main model has been generated.

Future work should explore bioactivity prediction of drug-target interactions from other benchmark datasets for comparison with additional baseline methods. Further improved encoders, for both drug compounds and protein targets, should also boost performance. The mentioned contrastive, pretraining methods are one option (Stärk et al., 2022; Sanchez-Fernandez et al., 2022). Even more informative, should be the 3D structures of both drugs and targets, which are not yet being used in our work. Recently, Equivariant Graph Neural Networks are more effectively being used to process 3D structures, which is a promising direction for further research in drug-target interaction modeling.

Acknowledgments and Disclosure of Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant No 956832. The ELLIS Unit Linz, the LIT AI Lab, the Institute for Machine Learning, are supported by the Federal State Upper Austria. IARAI is supported by Here Technologies. We thank the projects AI-MOTION (LIT-2018-6-YOU-212), AI-SNN (LIT-2018-6-YOU-214), DeepFlood (LIT-2019-8-YOU-213), Medical Cognitive Computing Center (MC3), INCONTROL-RL (FFG-881064), PRIMAL (FFG-873979), S3AI (FFG-872172), DL for GranularFlow (FFG-871302), AIRI FG 9-N (FWF-36284, FWF-36235), ELISE (H2020-ICT-2019-3 ID: 951847). We thank Audi.JKU Deep Learning Center, TGW LOGISTICS GROUP GMBH, Silicon Austria Labs (SAL), FILL Gesellschaft mbH, Anyline GmbH, Google, ZF Friedrichshafen AG, Robert Bosch GmbH, UCB Biopharma SRL, Merck Healthcare KGaA, Verbund AG, Software Competence Center Hagenberg GmbH, TÜV Austria, Frauscher Sensonic and the NVIDIA Corporation.

References

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

Ba, J., Hinton, G. E., Mnih, V., Leibo, J. Z., and Ionescu, C. Using fast weights to attend to the recent past. *Advances in Neural Information Processing Systems*, 29, 2016a.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016b.

Baek, D., Oh, Y., and Ham, B. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9536–9545, 2021.

Brock, A., Lim, T., Ritchie, J. M., and Weston, N. Smash: one-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344*, 2017.

Chang, O., Flokas, L., and Lipson, H. Principled weight initialization for hypernetworks. In *International Conference on Learning Representations*, 2019.

Dallago, C., Schütze, K., Heinzinger, M., Olenyi, T., Littmann, M., Lu, A. X., Yang, K. K., Min, S., Yoon, S., Morton, J. T., et al. Learned embeddings from deep learning to visualize and predict protein sets. *Current Protocols*, 1(5):e113, 2021.

Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. ProtTrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*, 2020.

Fabian, B., Edlich, T., Gaspar, H., Segler, M., Meyers, J., Fiscato, M., and Ahmed, M. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.

Fürst, A., Rumetshofer, E., Tran, V., Ramsauer, H., Tang, F., Lehner, J., Kreil, D., Kopp, M., Klambauer, G., Bitto-Nemling, A., et al. Cloob: Modern hopfield networks with infoloob outperform clip. *Advances in Neural Information Processing Systems*, 2022.

Galanti, T. and Wolf, L. On the modularity of hypernetworks. *Advances in Neural Information Processing Systems*, 33:10409–10419, 2020.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Guo, Z., Zhang, C., Yu, W., Herr, J., Wiest, O., Jiang, M., and Chawla, N. V. Few-shot graph learning for molecular property prediction. In *Proceedings of the Web Conference 2021*, pp. 2559–2567, 2021.

Ha, D., Dai, A. M., and Le, Q. V. HyperNetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Hansch, C. and Fujita, T. p-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *Journal of the American Chemical Society*, 86(8):1616–1626, 1964.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., and Rost, B. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics*, 20(1):1–17, 2019.

Hertzberg, R. P. and Pope, A. J. High-throughput screening: new technology for the 21st century. *Current opinion in chemical biology*, 4(4):445–451, 2000.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

Ji, S. and Marttinen, P. Patient outcome and zero-shot diagnosis prediction with hypernetwork-guided multitask learning. *arXiv preprint arXiv:2109.03062*, 2021.

Jiang, S., Feng, F., Chen, W., Li, X., and He, X. Structure-enhanced meta-learning for few-shot graph classification. *arXiv preprint arXiv:2103.03547*, 2021.

Kim, P., Winter, R., and Clevert, D.-A. Deep protein-ligand binding prediction using unsupervised learned representations. *ChemRxiv preprint 10.26434/chemrxiv.11523117.v1*, 2020.

Kim, P. T., Winter, R., and Clevert, D.-A. Unsupervised representation learning for Proteochemometric modeling. *International Journal of Molecular Sciences*, 22(23):12882, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Klein, B., Wolf, L., and Afek, Y. A dynamic convolutional layer for short range weather prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4840–4848, 2015.

Knyazev, B., Drozdzal, M., Taylor, G. W., and Romero Soriano, A. Parameter prediction for unseen deep architectures. *Advances in Neural Information Processing Systems*, 34, 2021.

Krueger, D., Huang, C.-W., Islam, R., Turner, R., Lacoste, A., and Courville, A. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

Lapinsh, M., Prusis, P., Gutcaits, A., Lundstedt, T., and Wikberg, J. E. Development of proteochemometrics: a novel technology for the analysis of drug-receptor interactions. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1525(1-2):180–190, 2001.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K. R. *Efficient BackProp*, pp. 9–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

Lenselink, E. B., Ten Dijke, N., Bongers, B., Papadatos, G., Van Vlijmen, H. W., Kowalczyk, W., IJzerman, A. P., and Van Westen, G. J. Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set. *Journal of cheminformatics*, 9 (1):1–14, 2017.

Litany, O., Maron, H., Acuna, D., Kautz, J., Chechik, G., and Fidler, S. Federated learning with heterogeneous architectures using graph hypernetworks. *arXiv preprint arXiv:2201.08459*, 2022.

Morgan, H. L. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

Muller, L. K. Overparametrization of hypernetworks at fixed flop-count enables fast neural image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 284–293, 2021.

Muratov, E. N., Amaro, R., Andrade, C. H., Brown, N., Ekins, S., Fourches, D., Isayev, O., Kozakov, D., Medina-Franco, J. L., Merz, K. M., et al. A critical overview of computational approaches employed for COVID-19 drug discovery. *Chemical Society Reviews*, 2021.

Nachmani, E. and Wolf, L. Molecule property prediction and classification with graph hypernetworks. *arXiv preprint arXiv:2002.00240*, 2020.

Noh, H., Seo, P. H., and Han, B. Image question answering using convolutional neural network with dynamic parameter prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 30–38, 2016.

Öztürk, H., Özgür, A., and Ozkirimli, E. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Pentina, A. and Clevert, D.-A. Multi-task proteochemometric modelling. *ChemRxiv preprint 10.26434/chemrxiv-2022-d5tzd*, 2022.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Adler, T., Kreil, D., Kopp, M. K., Klambauer, G., Brandstetter, J., and Hochreiter, S. Hopfield networks is all you need. *International Conference on Learning Representations*, 2021.

Sanchez-Fernandez, A., Rumetshofer, E., Hochreiter, S., and Klambauer, G. Contrastive learning of image-and structure-based representations in drug discovery. In *ICLR2022 Machine Learning for Drug Discovery*, 2022.

Schimunek, J., Friedrich, L., Kuhn, D., Rippmann, F., Hochreiter, S., and Klambauer, G. A generalized framework for embedding-based few-shot learning methods in drug discovery. In *ELLIS Machine Learning for Molecule Discovery Workshop*, 2021.

Schimunek, J., Seidl, P., Lukas, F., Kuhn, D., Rippmann, F., Hochreiter, S., and Klambauer, G. Context-enriched molecule representations improve few-shot drug discovery. *Openreview, to appear*, 2022.

Schmidhuber, J. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Shamsian, A., Navon, A., Fetaya, E., and Chechik, G. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pp. 9489–9502. PMLR, 2021.

Sosnin, S., Karlov, D., Tetko, I. V., and Fedorov, M. V. Comparative study of multitask toxicity modeling on a broad chemical space. *Journal of chemical information and modeling*, 59(3): 1062–1072, 2018.

Stärk, H., Dallago, C., Heinzinger, M., and Rost, B. Light attention predicts protein location from the language of life. *Bioinformatics Advances*, 1(1):vbab035, 2021.

Stärk, H., Beaini, D., Corso, G., Tossou, P., Dallago, C., Günnemann, S., and Liò, P. 3D Infomax improves GNNs for molecular property prediction. In *International Conference on Machine Learning*, pp. 20479–20502. PMLR, 2022.

Unterthiner, T., Mayr, A., Klambauer, G., Steijaert, M., Wegner, J. K., Ceulemans, H., and Hochreiter, S. Deep learning as an opportunity in virtual screening. In *Workshop on Deep Learning and Representation Learning at Conference of the Neural Information Processing Systems Foundation (NIPS 2014)*, 2014.

Vall, A., Hochreiter, S., and Klambauer, G. BioassayCLR: Prediction of biological activity for novel bioassays based on rich textual descriptions. In *ELLIS Machine Learning for Molecule Discovery Workshop*, 2021.

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Von Oswald, J., Henning, C., Sacramento, J., and Grewe, B. F. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019.

Wang, J. and Dokholyan, N. V. Yuel: Improving the Generalizability of Structure-Free Compound–Protein Interaction Prediction. *Journal of Chemical Information and Modeling*, 2022.

Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Widrich, M., Schäfl, B., Ramsauer, H., Pavlović, M., Gruber, L., Holzleitner, M., Brandstetter, J., Sandve, G. K., Greiff, V., Hochreiter, S., and Klambauer, G. Modern hopfield networks and attention for immune repertoire classification. *Advances in Neural Information Processing Systems*, 2020.

Winter, R., Montanari, F., Noé, F., and Clevert, D.-A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical science*, 10(6):1692–1701, 2019.

Ye, Q. and Ren, X. Learning to generate task-specific adapters from task description. *arXiv preprint arXiv:2101.00420*, 2021.

Zhang, C., Ren, M., and Urtasun, R. Graph HyperNetworks for Neural Architecture Search. In *International Conference on Learning Representations*, 2018.

Zhao, D., von Oswald, J., Kobayashi, S., Sacramento, J., and Grewe, B. F. Meta-learning via hypernetworks. In *4th Workshop on Meta-Learning at NeurIPS 2020 (MetaLearn 2020)*. IEEE, 2020.

Zhmoginov, A., Sandler, M., and Vladymyrov, M. HyperTransformer: Model Generation for Supervised and Semi-Supervised Few-Shot Learning. In *International Conference on Machine Learning*, pp. 27075–27098. PMLR, 2022.

A Training and hyperparameter details

In the following supplementary material, we presents an overview of the considered hyperparameters from our model selection phase in Table 2, including the explored values and final options marked in bold.

Table 2: Considered hyperparameter space for model selection, with selected configurations based on manual search on validation set shown in bold.

	HYPERPARAMETER	EXPLORED SPACE
TRAINING	OPTIMIZER	{**ADAM**}
	LEARNING RATE	{0.00005, **0.0001**, 0.0005, 0.001}
	SCHEDULER	{NONE, **REDUCEONPLATEU**}
	WEIGHT DECAY	{0.00001, **0.0001**, 0.001}
	META-BATCHING (PROTEINS)	{**32**, 128, 256, 512}
	MINI-BATCHING (MOLECULES)	{**32**, 256, 512, FULL}
HYPERNETWORK	NUMBER OF HIDDEN LAYERS	{0, **1**, 2, 4}
	HIDDEN DIMENSION	{16, 64, **256**, 512}
	DROPOUT	{0, **0.25**, 0.5, 0.75}
	ACTIVATION	{**RELU**, SELU}
	LAYERNORM	{**FALSE**, TRUE}
MAIN CLASSIFIER	NUMBER OF HIDDEN LAYERS	{0, **1**, 2}
	HIDDEN DIMENSION	{16, 64, 256, **512**}
	DROPOUT	{0, **0.25**}
CONTEXT MODULE	HIDDEN DIMENSION (QK)	{256, **512**}
	NUMBER OF HOPFIELD HEADS	{4, **8**}
	SCALING FACTOR, β	{0.1, 1/256, **1/512**}
	DROPOUT	{**0.5**}

B Analysis of the benchmark dataset

Lenselink et al. (2017) proposed a benchmark dataset of drug-target interactions derived from the ChEMBL database. The dataset includes 204,017 small molecular compounds and 1,226 protein targets, together making up 314,707 experimentally tested interactions with labeled bioactivity in terms of log affinity values. Fig. 4 illustrates the distribution of labels in the full dataset. For the purpose of creating a balanced split into the two classes active or inactive, Lenselink et al. (2017) imposed a fixed threshold of 6.5 log affinity on the bioactivities. Real-world data is however not balanced, many more interactions between drug-like molecules and proteins are considered to be inactive. Thus, we believe it might not be beneficial to restrict the learning to these arbitrary class labels but rather that it might be more informative for the models to make use of the raw continuous values. Additionally, we note that the tails of the distribution does not appear to be symmetric, for which reason the L1 should be more suited than the Mean Squared Error (MSE).

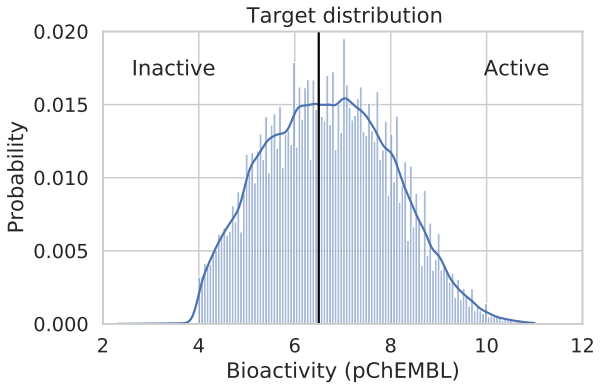

Figure 4: Distribution of bioactivity values from the full benchmark dataset derived from ChEMBL. A fixed threshold of 6.5 log affinity was imposed in the benchmark, for the purpose of creating a balanced class division into active/inactive interactions.

C Complementary results

The following supplementary material presents additional results.

C.1 Full re-implemented benchmarking results

Tables 3 and 4 present the benchmarking results that are solely from our own experiments, whereas Table 1 presented the top results from either our experiments or previous work. A more fair comparison can be made from the results presented in this section, as all experiments are made on the exact same data splits. However, the same conclusions as from Table 1 can be drawn. HyperPCM outperforms all baselines in all four settings, both in terms of MCC and AUC.

Table 3: **Matthews Correlation Coefficient (MCC).** From 10-fold cross-validation in the random, LCCO, and LPO settings and 10 reruns in the temporal setting. All results from re-implementation of baseline models. Best performance per setting is marked in bold and best baseline performance is marked in italic.

		FEW-/MANY-SHOT			**ZERO-SHOT**
MODEL	DESCRIPTORS	RANDOM	TEMPORAL	LCCO	LPO
DEEPPCM‡	MOLBERT + UNIREP	0.622±0.006	*0.370±0.008*	0.452±0.058	0.288±0.045
	MOLBERT + PROTBERT	0.625±0.006	0.362±0.006	0.455±0.054	0.299±0.043
	MOLBERT + PROTT5	0.620±0.004	0.360±0.003	0.452±0.057	0.296±0.040
	MOLBERT + SEQVEC	0.639±0.003	*0.370±0.006*	0.471±0.063	0.310±0.036
	CDDD + UNIREP	0.623±0.006	0.352±0.009	0.451±0.064	0.298±0.039
	CDDD + PROTBERT	0.635±0.004	0.343±0.006	0.462±0.060	0.294±0.049
	CDDD + PROTT5	0.634±0.005	0.353±0.006	0.460±0.056	0.310±0.054
	CDDD + SEQVEC	*0.643±0.005*	0.363±0.006	*0.478±0.048*	*0.316±0.043*
HYPERPCM	CDDD + SEQVEC	**0.682±0.039**	**0.395±0.005**	**0.532±0.059**	**0.340±0.051**

‡ (KIM ET AL., 2021)

Table 4: **Area Under the ROC-curve (AUC)**. From 10-fold cross-validation in the random, LCCO, and LPO settings and 10 reruns in the temporal setting. All results from re-implementation of baseline models. Best performance per setting is marked in bold and best baseline performance is marked in italic.

		FEW-/MANY-SHOT			**ZERO-SHOT**
MODEL	DESCRIPTORS	RANDOM	TEMPORAL	LCCO	LPO
DEEPPCM‡	MOLBERT + UNIREP	0.893±0.002	0.743±0.004	0.803±0.032	0.697±0.028
	MOLBERT + PROTBERT	0.894±0.002	0.740±0.003	0.804±0.031	0.705±0.029
	MOLBERT + PROTT5	0.892±0.002	0.739±0.002	0.802±0.032	0.702±0.029
	MOLBERT + SEQVEC	*0.900±0.001*	*0.747±0.003*	0.813±0.033	0.711±0.025
	CDDD + UNIREP	0.893±0.002	0.735±0.004	0.800±0.034	0.705±0.026
	CDDD + PROTBERT	0.897±0.002	0.732±0.003	0.805±0.033	0.704±0.033
	CDDD + PROTT5	0.897±0.002	0.737±0.004	0.805±0.031	0.710±0.038
	CDDD + SEQVEC	*0.900±0.002*	0.743±0.004	*0.814±0.027*	*0.715±0.026*
HYPERPCM	CDDD + SEQVEC	**0.919±0.017**	**0.765±0.003**	**0.850±0.028**	**0.738±0.030**

‡ (KIM ET AL., 2021)

C.2 Extended zero-shot analysis with unseen molecules

We hypothesize that a drawback of our method compared to previous concatenation approaches, could be that the concatenation model has greater capacity to memorize seen molecules whereas the advantage of our model is more prominent with regards to generalization in terms of both unseen protein targets and drug compounds. As such, we provide an extended analysis comparing the results of the best performing baseline and our HyperNetwork method on a subset of the LPO test set

containing only unseen molecules. Fig. 5 presents the distribution of performance over the 10-fold cross-validation, on a subsets of the LPO split containing only drug compounds that were not seen during training. As expected, HyperPCM outperforms DeepPCM even more consistently on these, more demanding test sets. The performance of our method in the extended setting is 0.313 ± 0.058 compared to 0.281 ± 0.051 for the best baseline, which is a significant improvement ($p = 0.002$).

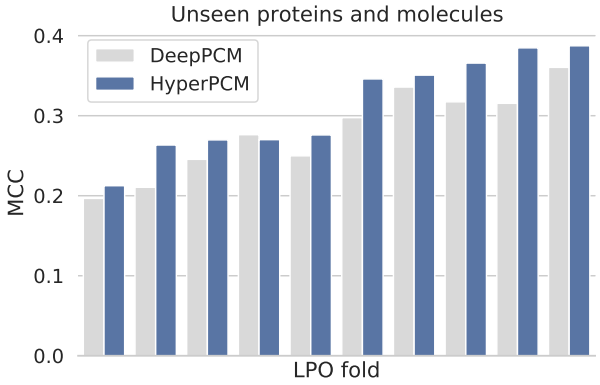

Figure 5: **Extended LPO.** Comparing test performance on each fold from the cross-validation of the baseline, DeepPCM with CDDD and SeqVec encoders, and our model, HyperPCM, on a subsets of the LPO split with only molecules not seen during training.

C.3 Additional ablation study: learning curves

As a complement to the ablation study, we present the average learning curves from the experiments of the DeepPCM baseline and our HyperPCM model using the L1 loss compared to the BCE loss in Fig. 6. The results show that both models overfit immediately in the binary classification case, whereas generalization continues to improve for at least 200 epochs using L1.

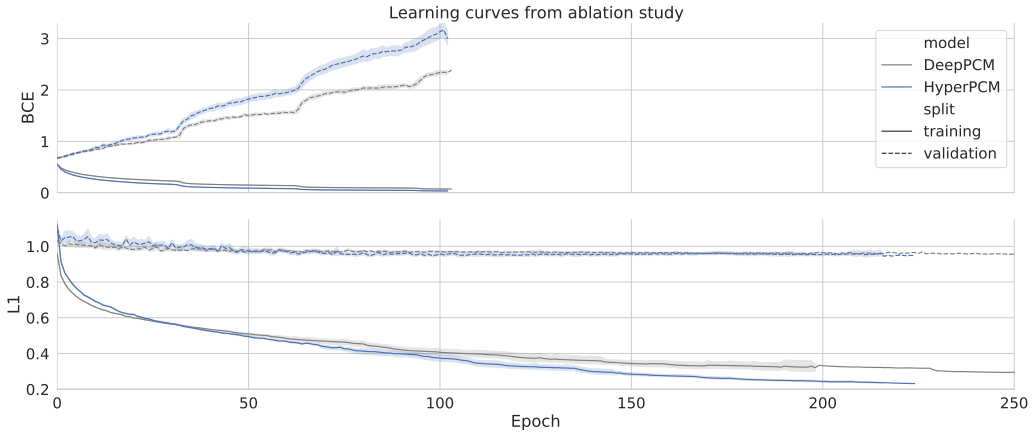

Figure 6: **Learning curves.** Average learning curves from ablation study for DeepPCM baseline and our model HyperPCM trained using Binary Cross-Entropy (BCE) loss versus L1 loss.