ProtMIMO: Multi-Input Multi-Output Models for Protein Landscape Prediction

Amir Shanehsazzadeh
Harvard University
Cambridge, MA 02138
amirshanehsazzadeh@college.harvard.edu

Abstract

We apply multi-input multi-output (MIMO) models [3], which are models designed to replace traditional ensembles, to the space of biological sequence design. Specifically, we aim to model protein fluorescence using data from Sarkisyan et al. [7] (this modeling task will be dubbed "the fluorescence task"). We extend the original MIMO architecture designed for classification to regression and compare it to traditional model ensembles. We consider both feed-forward and convolutional architectures. We find that MIMO models achieve similar performance on the fluorescence task while also having lower residual correlations (measured between different outputs of the MIMO models and different ensemble components for the ensembles). Furthermore, we verify empirically that an N-input/output MIMO model has $O(N)$ faster inference time than an ensemble of N models. Our results provide evidence for the usefulness of MIMO networks for protein design as a potential replacement for traditional ensembles. Our code is made publicly available[1].

1 Introduction

There has been great interest recently in machine learning based biological sequence design [9]. Oracles, models designed to predict properties of biological sequences, are an integral part of the design workflow. Like other machine-learning problems, oracle diversity is highly valued. Model ensembles are used as one approach to increasing diversity, however there is a trade-off here with inference time and compute requirements. We propose using multi-input multi-output (MIMO) networks [3] to model biological sequences. The motivation for MIMO networks is the lottery ticket hypothesis [2], which states that only a fraction of a neural network's parameters are necessary to preserve performance. MIMO proposes training multiple models in one and using the average of the outputs as an ensemble value, thus reducing model inference time relative to ensembles while preserving (or perhaps enhancing) the increased model diversity. In this project, we will explore modeling various protein sequence with MIMO models and comparing performance, both on the relevant biological task and in terms of inference time, to traditional ensembles.

2 Background

2.1 Multi-Input Multi-Output (MIMO) Models

Multi-input multi-output (MIMO) models were introduced by Havasi et al.[3]. The authors showed that MIMO models improved performance relative to ensembles on the computer vision benchmarks

[1] https://github.com/amirshane/ProtMIMO

Preprint. Under review.

CIFAR10, CIFAR100, and ImageNet, while also offering a speed-up. MIMO models aim to replace traditional ensembles of N models by training a single model that takes in N inputs and outputs N corresponding values. During training, the dataset is shuffled N times so that the model sees N distinct inputs per pass. For inference, the same input is given N times and the N outputs are averaged. While the original MIMO architecture was designed for classification tasks, we modify the architecture for regression tasks. See Figure 1 for a diagram comparing ensembles and MIMO for regression.

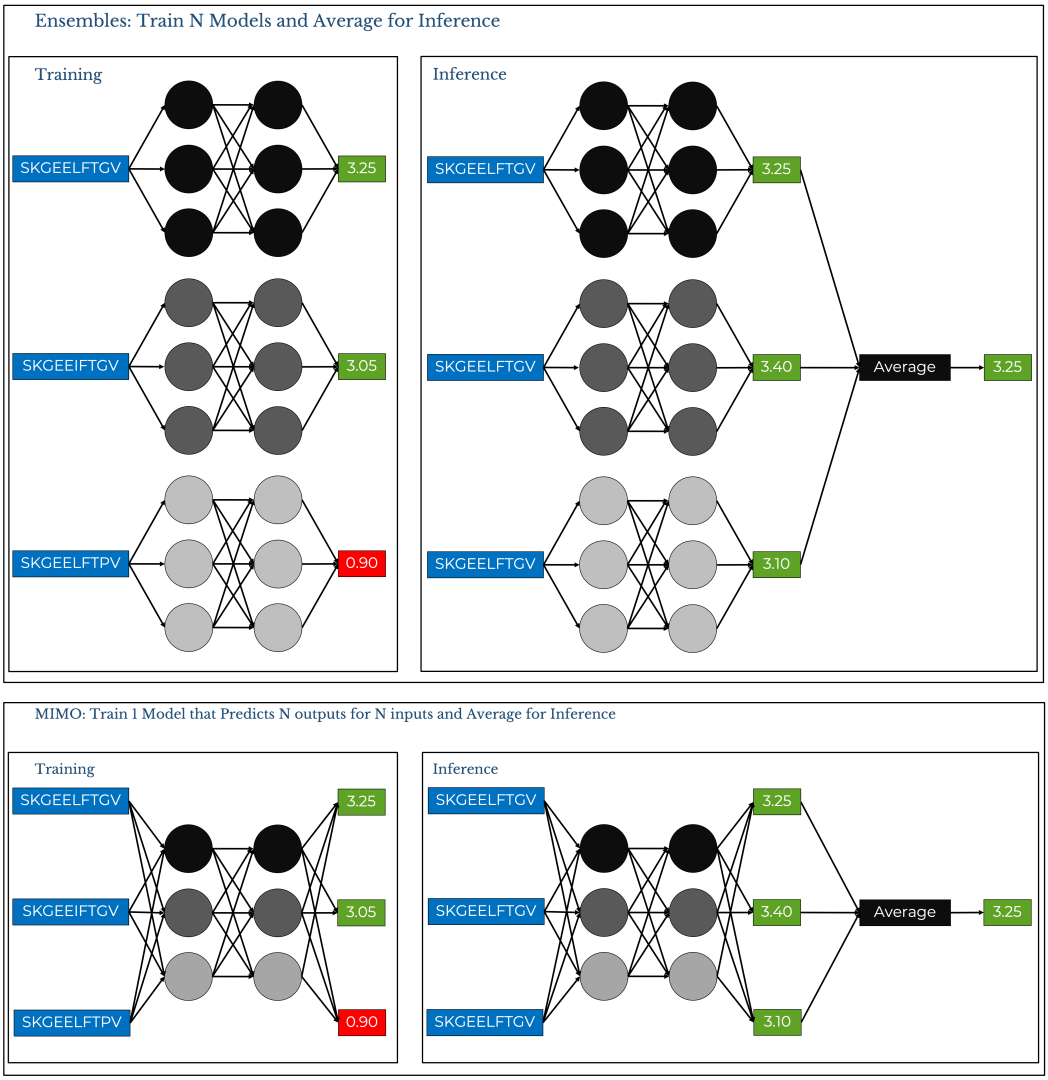

Figure 1: Traditional Ensemble of Models vs. Multi-Input Multi-Output (MIMO) Model

2.2 Proteins

We consider proteins using only their amino acid sequence (primary structure). This sequence is a string consisting of tokens from a 21-letter alphabet that includes the 20 standard amino acids [5] as well as a pad index. A length ℓ protein $a = a_1 a_2 \cdots a_\ell$ is thus modeled as a discrete sequence $x = (x_1, x_2, ..., x_\ell)$ with $x_i \in \{0, 1, ..., 19\}$ and potentially padded at the end to $(x_1, x_2, ..., x_\ell, 20, 20, ..., 20)$.

2.2.1 Fluorescence Landscape Prediction

This regression task involves mapping a protein to its log-fluorescence, which is a real-valued label. The experimental data is from Sarkisyan et al. [7] and the curated dataset is from TAPE[2]. The data consists of mutated variants of a wild-type GFP protein with edit distance up to 14. The train set contains all variants within edit distance 3 of the wildtype (at most 3 mutations away) and the test set contains all variants at least 4 mutations away from the wildtype. This split by edit distance allows for testing the generalizability of a model trained on a small (local) neighborhood of the wildtype to a larger (global) neighborhood. Note that because of this split the train set consists of 82% bright proteins (log-fluorescence greater than 2.5) and 18% dark proteins (log-fluorescence less than 2.5) whereas the test set consists of only 32% bright proteins and 68% dark proteins. This class imbalance makes it difficult for models to generalize from the low mutation train data to the high mutation test data. The primary metric of interest is Spearman's rank correlation coefficient ρ on the test set.

3 Methods

We describe our methods here. We consider two distinct MIMO architectures, one that is feed-forward and another that is convolutional. We use the Adam optimizer [4] with a fixed learning rate and run several experiments iterating over model depth and number of inputs.

3.1 Architectures

The first architecture we use is a feed-forward architecture. Note that the input is a one-hot encoded protein sequence of dimension $20 \times (N \times \ell)$ where N is the number of model inputs and ℓ is the maximum sequence length. We first apply a fully-connected encoder layer of dimension 512: $\mathbb{R}^{N \times \ell} \mapsto \mathbb{R}^{512}$. We subsequently apply L fully-connected layers that map to dimension 256. ReLU is applied after every layer. Finally, we apply a multi-head linear layer, which is a fully-connected layer that returns N outputs. We denote this architecture as $F_m(N, L)$ for the MIMO version and $F_e(N, L)$ for the traditional ensemble. Note that for the traditional ensemble the multi-head linear lyear is just a standard linear layer.

The second archicture is a convolutional neural network. We consider the same input as before and again use a fully-connected encoder layer of dimension 512. We subsequently apply L 1-dimensional convolutional layers with feature size 64 and kernel width of 5. We do not pool in between convolutional layers but ReLU is again applied after each layer and finally a multi-head linear layer is applied as in the case of the feed-forward architecture. We denote this architecture as $C_m(N, L)$ for the MIMO version and $C_e(N, L)$ for the traditional ensemble.

3.2 Training

The models are trained using the Adam optimizer with a fixed learning rate of 10^{-4} and a batch size of 32 for a maximum of 100 epochs. The fluorescence dataset, as curated by TAPE, has a train, validation, and test set. For an ensemble of N models we use N different random seeds $s_1, ..., s_N$ to shuffle the data (i.e. the ith model is trained with data shuffled using seed s_i). For the corresponding MIMO model we use N copies of the data with the ith copy shuffled using seed s_i. Early stopping is implemented with a patience of 10 (training stops early if validation loss does not improve for 10 sequential epochs). The loss function is the mean-squared error. For the MIMO architecture the predictions and labels are flattened before loss and gradient computation.

3.3 Experiments

We train MIMO models and ensembles of both architectures for several different parameters (different numbers of layers and numbers of inputs). Specfically, we train and evaluate the models $\{F_m(N, L), F_e(N, L), C_m(N, L), C_e(N, L)\}$ for $2 \leq N \leq 5$ and $1 \leq L \leq 10$. For evaluation, we compute Spearman ρ on the test set as well as model residual correlations. For residual correlations, we compute Pearson correlations between the residuals $y - \hat{y}$ of the different models. For ensembles, we consider the residuals of the different components of the ensembles (models i and j) and for

[2]https://github.com/songlab-cal/tape

MIMO models we consider the residuals of the different outputs of the model (outputs i and j). Note that lower residual correlations between models indicates a lower probability of the models agreeing on when they are wrong, which measures model diversity. Additionally, we compute average inference times for all models.

4 Results

We present our results here. We consider three axes of performance: quality of fit with Spearman ρ, diversity with model residual correlations, and inference time. We find that our results are robust to architecture choice.

4.1 Quality of Fit

We present box-plots of test-set Spearman ρ's below. We find that both architectures achieve a maximum value of $\rho \in [0.67, 0.68]$, which is in-line with other methods [8].

In Figure 2 and Figure 3 we see that both architectures achieve similar top values of ρ while the convolutional models tend to have less variance. Additionally, we see that as the number of inputs/outputs of the MIMO models increase the median ρ tends to decrease considerably, whereas for the convolutional models the opposite trend is observed (but with an order of magnitude smaller change in values). The former result is likely a result of the MIMO network being "overloaded" (asked to predict too many outputs).

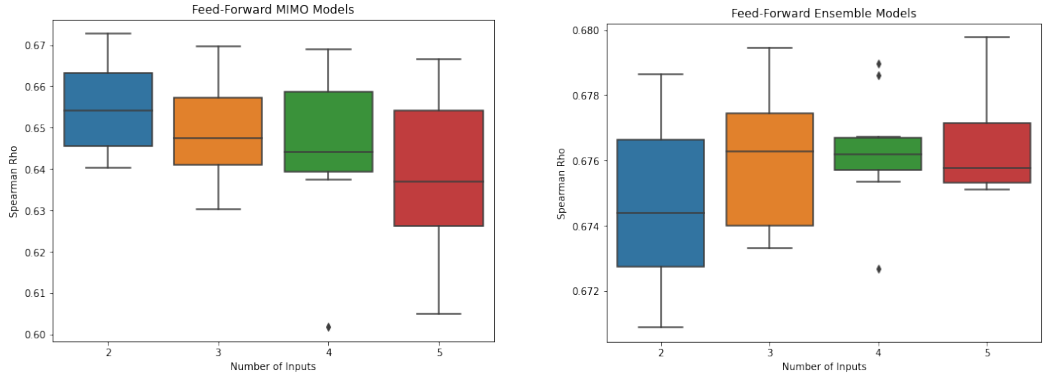

Figure 2: Test-Set Spearman ρ as a Function of Number of Inputs (Ensemble Size) for Feed-Forward Models, MIMO Models (Left) and Ensemble Models (Right)

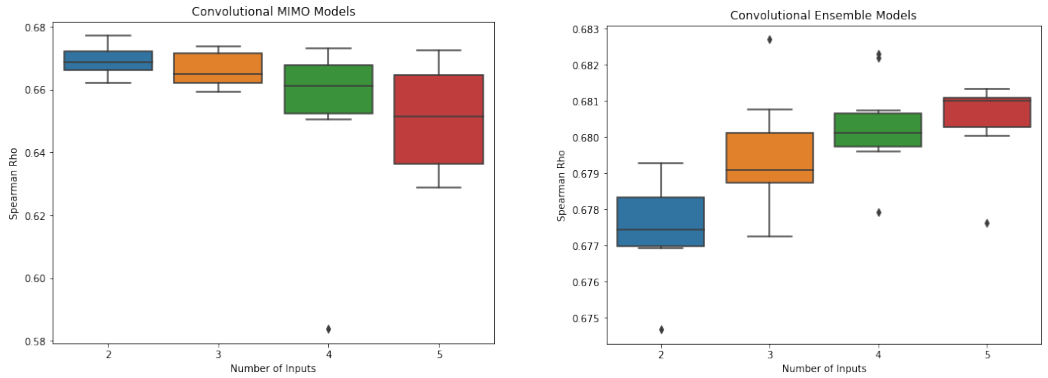

Figure 3: Test-Set Spearman ρ as a Function of Number of Inputs (Ensemble Size) for Convolutional Models, MIMO Models (Left) and Ensemble Models (Right)

In Figure 4 and Figure 5 we see that for both architectures increased model depth does not appear to improve performance. For the MIMO models, in particular, more layers appears to worsen average performance.

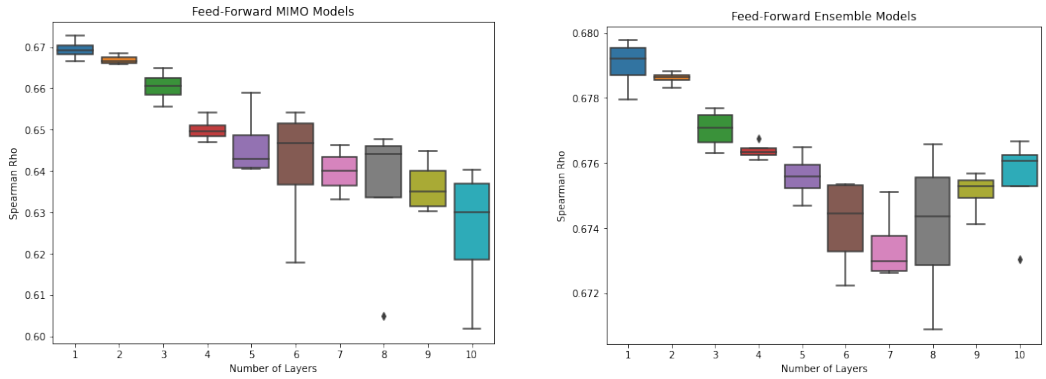

Figure 4: Test-Set Spearman ρ as a Function of Model Depth for Feed-Forward Models, MIMO Models (Left) and Ensemble Models (Right)

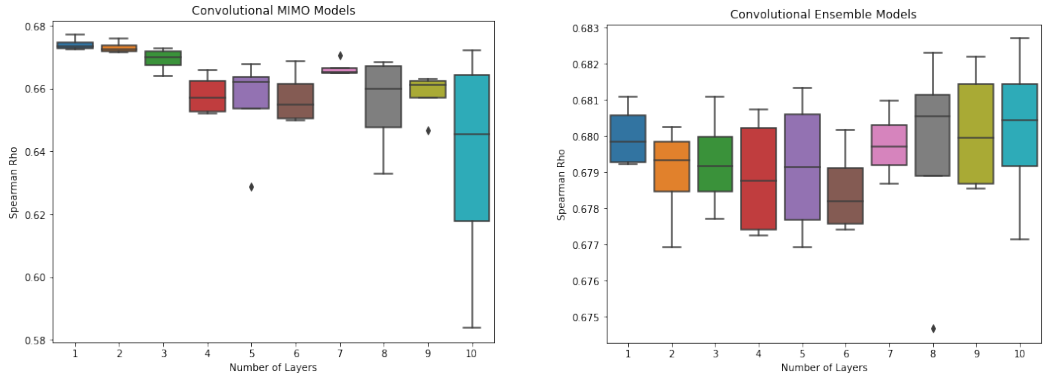

Figure 5: Test-Set Spearman ρ as a Function of Model Depth for Convolutional Models, MIMO Models (Left) and Ensemble Models (Right)

Our results suggest that MIMO models fit the dataset approximately as well as the ensemble models.

4.2 Diversity

We present box-plots of average model residual correlations below.

In Figure 6 and Figure 7 we see that the components of the ensemble models tend to have very high residual correlations (> 0.95 for the feed-forward architecture and > 0.90 for the convolutional architecture). Furthermore, these correlations stay high regardless of ensemble size. The MIMO models, however, tend to have much lower residual correlations and the median residual correlation appears to decrease as the number of inputs/outputs to the model increases.

In Figure 8 and Figure 9 we see that the median residual correlation tends to decrease substantially for both MIMO architectures as the number of layers increases. This value stays the same for convolutional ensembles and very modestly decreases for the feed-forward ensembles (by an order of magnitude less than for the MIMO models).

Our results suggest that MIMO models tend to have considerably lower residual correlations than ensembles, leading us to conclude that the MIMO models produce more diverse predictions.

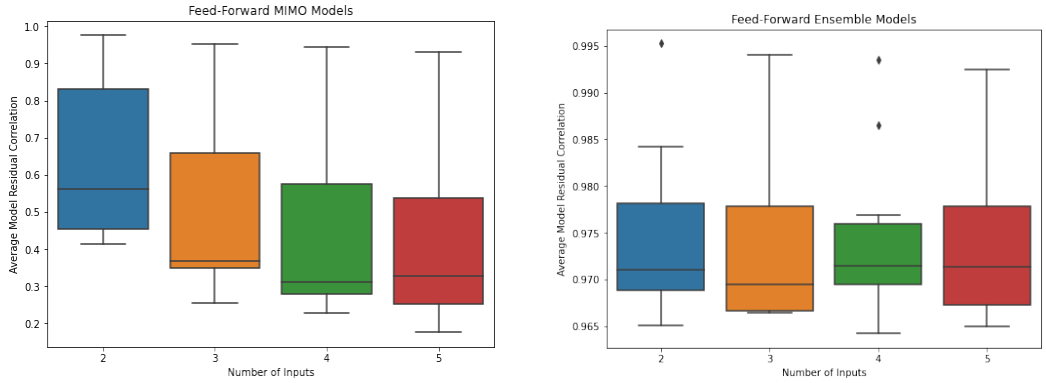

Figure 6: Average Model Residual Correlations as a Function of Number of Inputs (Ensemble Size) for Feed-Forward Models, MIMO Models (Left) and Ensemble Models (Right)

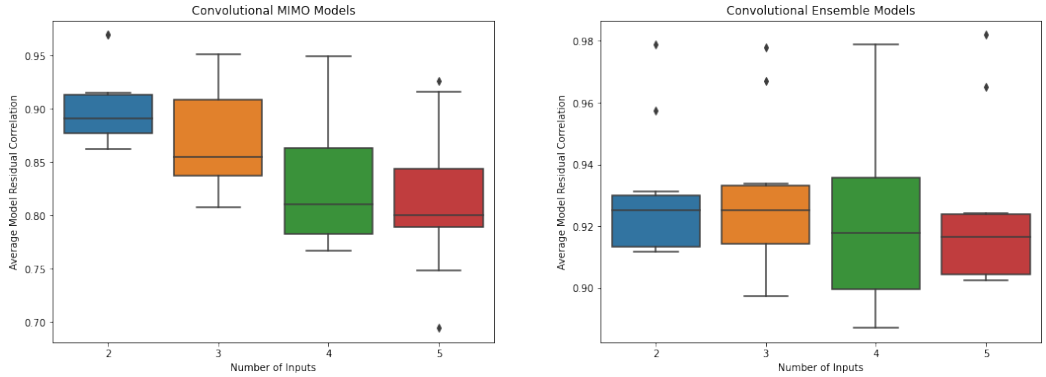

Figure 7: Average Model Residual Correlations as a Function of Number of Inputs (Ensemble Size) for Convolution Models, MIMO Models (Left) and Ensemble Models (Right)

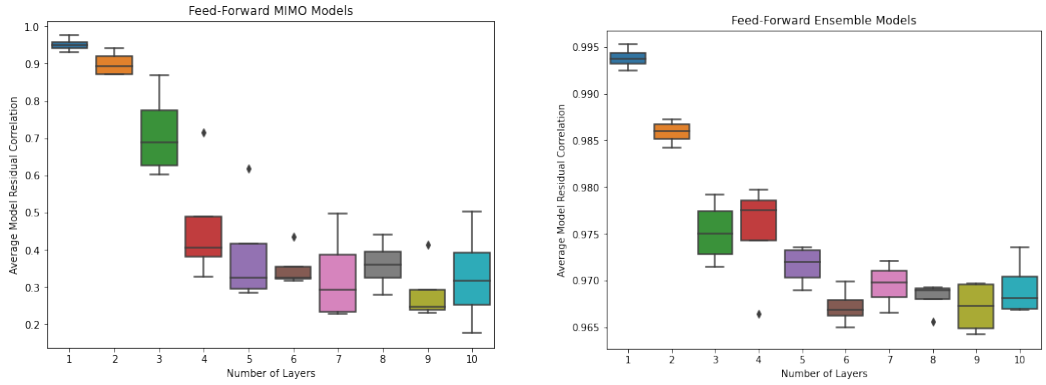

Figure 8: Average Model Residual Correlations as a Function of Model Depth for Feed-Forward Models, MIMO Models (Left) and Ensemble Models (Right)

4.3 Inference Time

We compute the ratio of inference times between each ensemble model and MIMO model (with the same architecture and hyperparameters). For the ensemble models, the inference time is the sum of the inference times for all components of the models. We expect this ratio to be $O(N)$ for an N input/output MIMO model.

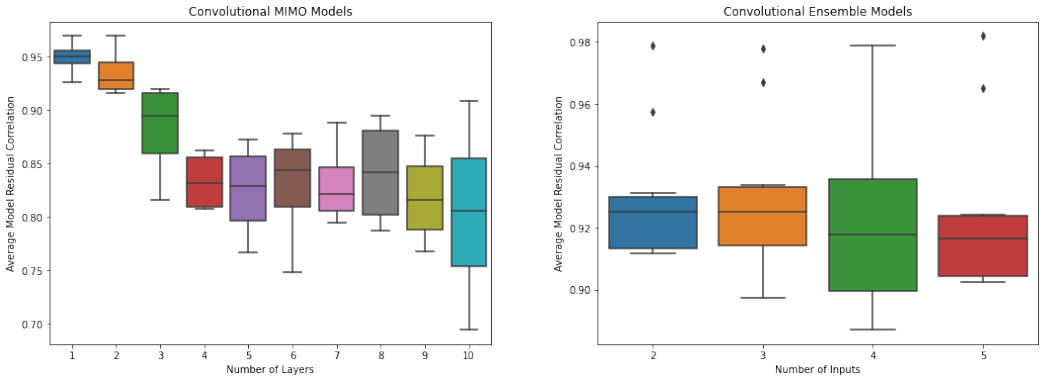

Figure 9: Average Model Residual Correlations as a Function of Model Depth for Convolution Models, MIMO Models (Left) and Ensemble Models (Right)

In Figure 10 we present box-plots of the inference time ratios for both architectures. We see the desired trend as the ratio tends to increase as N increases and it does so in a linear fashion (note that the variability comes from the number of layers in the model). The time ratio is generally larger for the convolutional models than the feed-forward models, which is due to the fact that the convolution operations are more expensive than the feed-forward operations and so the initial encoder layer and multi-head output layer (which have run-times of $O(N)$) has less impact on the entire model's inference time.

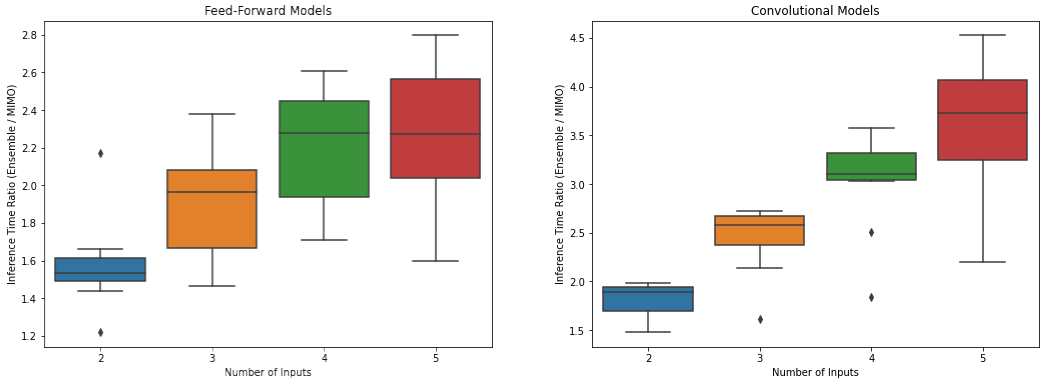

Figure 10: Ratios of Model Inference Times as a Function of Number of Inputs (Ensemble Size) for Feed-Forward Models (Left) and Convolutional Models (Right)

In Figure 11 we plot inference time ratios as a function of model depth. As expected, we see that inference time ratios increase as the number of inputs increase. We additionally see the ratios increase as model depth increases, which makes sense as more hidden layers results in the initial encoder and multi-head output layer taking up a smaller portion of the total model inference time. For the convolutional models, in particular, we see relatively linear increases of the inference time ratios as the number of inputs increases.

Our results suggest that the inference time ratio of an ensemble of N models to an N input/output MIMO model approaches $O(N)$ as the model depth (more generally total hidden layer inference time) increases. This lines up precisely with the theoretical limit.

5 Discussion

We find that for the fluorescence task, MIMO models perform similarly to traditional ensembles on test-set Spearman ρ. However, as N, the number of inputs/outputs of the MIMO models, increases, ρ

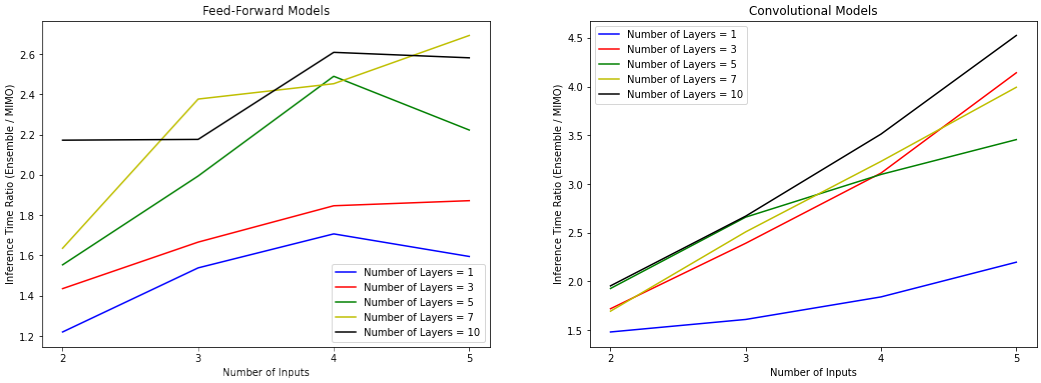

Figure 11: Ratios of Model Inference Times as a Function of Model Depth for Feed-Forward Models (Left) and Convolutional Models (Right)

tends to decrease. This is aligned with theory as the lottery ticket hypothesis would imply that there is some N for which model performance is sacrificed (corresponding to a highly-pruned model).

It is worth noting that MIMO models have considerably lower residual correlations than ensembles, which suggests that their predictions are more diverse. The residual correlations also tend to decrease as N increases and as the model depth L increases. This presents us with a trade-off of absolute performance for diversity by increasing N and L.

Finally, we see that MIMO models have faster inference times by an $O(N)$-factor, which would enable greater throughput.

Considered together, our results provide some evidence for the usefulness of MIMO models over traditional ensembles for protein design.

5.1 Future Direction

The immediate next direction would be to consider additional protein design benchmarks for both classification and regression tasks. Examples of such benchmarks can be found in TAPE [6] and FLIP [1]. We in fact implemented experiments for the stability regression task from TAPE but were unable to run them due to time and resource constraints. Validating our results on several benchmarks would provide strong evidence for using MIMO models instead of ensembles.

5.2 Acknowledgements

This work was done for Harvard's Spring 2022 iteration of CS 282R. Thank you to David Belanger and Zelda Mariet for advising this project.

References

[1] Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, 2022.

[2] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. 2018.

[3] Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M. Dai, and Dustin Tran. Training independent subnetworks for robust prediction, 2020.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[5] International Union of Pure and Applied Chemistry. Nomenclature and symbolism for amino acids and peptides (recommendations 1983). *Pure and Applied Chemistry*, 56(5):595–624, January 1984.

[6] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with tape, 2019.

[7] Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397, 2016.

[8] Amir Shanehsazzadeh, David Belanger, and David Dohan. Is transfer learning necessary for protein landscape prediction?, 2020.

[9] Sam Sinai and Eric D Kelsic. A primer on model-guided exploration of fitness landscapes for biological sequence design, 2020.