

Neural network extrapolation to distant regions of the protein fitness landscape



Open Access This file is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. In the cases where the authors are anonymous, such as is the case for the reports of anonymous peer reviewers, author attribution should be to 'Anonymous Referee' followed by a clear attribution to the source work. The images or other third party material in this file are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

Reviewer #1 (Remarks to the Author):

The authors utilize various neural networks to design protein sequences and evaluate the generated sequences based on sequence statistics and experimental validation results. The experimental validation is well-structured, testing both the expression and IgG binding of sequences derived from different methods. The experiment results interestingly reflect different inductive biases of neural networks. Despite the model and data being similar to a previously published paper, the authors introduce some novel ideas.

However, I question the abstract's conclusion: "We find simpler models excel in local extrapolation to design high fitness proteins". Figure 3e suggests that the "EnsC" method is comparable, if not superior, to simpler LR/FCN models. The "EnsC" method, which combines 100 CNNs, should be considered a CNN method that eliminates aleatoric uncertainty. The figures reveal that a single GCN outperforms a single CNN. So, why not test a model that combines 100 GCNs? This could potentially be the most robust model.

The authors should directly compare the additive model, as the inductive biases of LR and FCN are established to capture the additive mutational effects. An additive model for predicting multiple mutants has been shown to perform better than more complex methods (PMID: 37591206).

Is there any consistency between the results of the retrospective experiment (Figure 1) and the prospective study results (Figure 3)?

The conclusions drawn seem scattered. It would be more practical for the authors to provide final recommendations on which model to use under different circumstances. Additionally, when making the decision, the diversity of the generated sequence should be taken into account.

Reviewer #2 (Remarks to the Author):

Fahlberg et al explore multiple neural network architectures to assess their ability to extrapolate from training data on GB1 protein binding to IgG. The training data is single and double mutants. The extrapolation is done by exploring quite extreme mutants – all the way to 50. This is enabled via a yeast display system. Specific sequences verified separately for binding affinity.

Some of the details that should be present in the main text that were a bit hard to figure out for me: GB1 is a 56 residue protein (?), the training data is from a slightly different experiment, the models take the proteins in a fixed-length one-hot sequence, mutations cannot have gaps (?).

Overall, I like this paper. The methods are powerful and it's great to see work on answering a fundamental question. The analyses are well-done and the data will be useful to explore. There are a few issues for me with the concept and then I have some minor technical critiques.

My understanding is that the goal of this paper is to evaluate supervised deep learning's ability to extrapolate away from training data for one protein. However, most current supervised learning models use something like ESM or UniRep as a pre-trained featurization of proteins and then do supervised learning with those models. These models are state of the art in most tasks and usually generalize across sequence quite well. They are trained on billions of sequences across most known proteins. The models used in the paper -CNN/MLPs – these are a bit dated for this field. I'm not saying this undermines their usefulness, but just that the impact is not quite there for the field because most of the research is on different models trained across many sequences.

Also, I question the premise here – what is GB1 protein? If I have mutated every single residue is it still GB1? Some of the comments in the article – like "Even a small protein like GB1 has over 10^{70} possible sequence configurations" seem to imply any length 56 protein is GB1. This is a bit of semantics, but practically it is probably necessary to define what you consider "success." Like if generalization means predicting accurately across all sequence space of length 56 after training around one sequence, then fine – just be clear.

To address this issue, one way could be to include in your analysis an ESM fine tuned model or MPNN. It would be valuable to show their ability to generalize – especially to sequences only accessible in laboratories and not found in typical MSA. Maybe this is better for a follow-up paper though (which I would happily review!!)

Based on those concerns, I believe the scope of this work is just a bit narrow. These models aren't as relevant today to what is at the frontier of protein engineering and the narrowness of one protein/one-task do not support the title/claims. The methods you've built and the approach are excellent and I believe with just some small changes by choosing different models and maybe including another 1-2 systems will make a landmark paper. As someone who works on protein engineering, the biggest open question I currently have is assessing well these models generalize to sequences far from nature.

Here are a few small technical considerations:

1. Some convergence assessment on SA would be great to see. I'm surprised you didn't see some pathological sequences at high mutation count (e.g., I often see poly-W or other weird stuff come out of optimization with models). Could you be running the SA too short?
2. Would love to see some BLOSUM analysis instead of hamming – are the hydrophobic regions being rearranged? Can you do AF2 and align some of these extreme mutants?
3. Please include a few more details needed to make sense of the results – sequence length of GB1, input representation, etc
4. The cited Gelman paper explores a few architectures, can you clarify when saying the architecture from that paper was used

Reviewer #3 (Remarks to the Author):

The authors present the results from training a panel of neural network architectures on GB1-IgG binding data for single and double mutants, generating thousands of GB1 designs with up to 50 mutations, and systematically evaluating the models' ability to extrapolate and produce designs which are both foldable and functional. This is a systematic work which uses a relatively well-understood model protein system to understand how various neural network architectures perform when attempting to extrapolate to generate functional variants with far more mutations than are present in any single genotype of the training data. As such, the scope of this work can be considered to be examining the performance of various models on a simple and well-characterized model protein system with a relatively simple but comprehensive training data set.

Overall the experiments of this work appear to be well-designed and performed, leading to reasonable conclusions about what models of varying complexity are capable of in the space of protein design. The work and interpretations appear generally valid and the authors appropriately express value and limitations of their findings.

The work is limited in scope beyond the relatively large number of genotypes generated and tested, in that there is no attempt to design new molecular function. The designed molecules must simply maintain or improve the native function in order to survive the selection. Given the wealth of methods that exist for designing, selecting, and screening molecular binders, this work will likely not have a major impact on the design of this class of molecules. Given that there is no attempt to design new or altered molecular function, such as binding of a noncognate target, into GB1, this work also represents an interesting but somewhat incremental advance in the understanding of the ability of neural network to extrapolate beyond their training data for protein design.

The tendency of the CNNs and the GCN to target regions of positive epistasis is encouraging, suggesting that the models learn the relationships between positively interacting positions and can exploit these interactions to create better protein designs. This may partly explain why these models are good at promoting protein folding, since positive epistatic interactions between specific amino acids often play an important role in this process and in the stability of a protein structure. An interesting finding is that the more sophisticated neural models were, in general, not able to maintain protein function when forced to introduce many mutations, and instead could only maintain protein folding. Ensemble predictor "EnsC" was the most successful at maintaining protein function with up to 20 mutations, which may be the most significant finding of this work for

the future of protein engineering.

The authors published previous results similarly, training models on the same set of single- and double-mutants, but only evaluated model performance on held-out single- and double-mutants. Thus, the scope of this work is relatively narrowly defined, only looking at the ability of the models to extrapolate 'deeper into sequence space', but not fundamentally changing the experimental paradigm.

Due to the fact that this work uses a well-characterized model system, a more impactful strategy may have been to devise a clever way to construct a large (but likely still 'sparse') library of triple, quadruple, etc. mutants to use as training data, and assess whether the more complex epistatic interactions contained in such training data are able to extend the ability of the models to predict foldable and functional protein variants with greater numbers of mutations.

Overall I find this work to be rigorous and convincing in its measured claims, but might expect a more ambitious scope, novel methods, or more groundbreaking findings for a paper that would be published in Nature Communications. It may be better suited for a quality topical journal.

Reviewer #4 (Remarks to the Author):

In this manuscript, the authors evaluate the capacity of artificial neural networks (the workhorse of modern machine learning) to extrapolate beyond their training data in protein design. The topic of extrapolation is an important one for machine learning since it determines its effectiveness, and the topic of protein design is important because of its sundry applications and because it significantly challenges ML approaches with a very large possible phase space.

The authors examine several models and architectures of neural networks: a linear regressor (LR), a fully connected neural network (FCN), three convolutional neural networks with different initializations (CNNs), a graph convolutional neural network (GCN), and two ensemble approaches of 100 CNNs using their median and 5th percentile (EnsM and EnsC). Their goal is to predict the display of protein G (GB1) in the yeast cell surface ("display") and its binding to the antibody immunoglobulin G (IgG, "binding").

As a first step, they use literature data from a library measuring fitness (equivalent to binding) for protein variants that encompass most 1 and 2 mutation variants to train their models. When testing with another library that has variants with 3 and 4 mutations, model accuracy drops dramatically in the extrapolated regime, but the Spearman's correlation remains significantly above 0 (around 0.4), suggesting potential for model-guided design at or beyond 4 mutations.

They then developed a large-scale protein design computational pipeline that leverages simulated annealing (SA) to identify high fitness peaks. They used this pipeline to computationally propose 41 sequences for each model, so as to compare their performance, finding notable differences in the sequences designed by each algorithm.

A third part of the paper involves experimentally testing these proposed protein designs. For this purpose, they used FACS and they were able to show which protein designs displayed in the cell surface (display) and which ones showed binding to IgG (binding, which correlates to the fitness value used in the literature data used above). They find that all models show the ability to extrapolate to 2.5-5x more mutations than the training data (from 1, 2 to 5, 10 mutations), but the design performance decreases sharply with further extrapolation. The simpler LR and FCN models outperformed the more sophisticated CNN and GCN models for designing sequences that bind, whereas the ensemble models showed similar performance. The availability of both "display" and "binding" data allows the authors to distinguish between models that predict protein folding well but not binding and models that are good binding predictors.

Finally, they did a more detailed structural and quantitative analysis of twenty designs: ten with 40-50 mutations that displayed but showed no binding and then with 5-20 mutations that showed binding. Almost all displaying designs did so more than for the wild type case, and all the high-binding designs showed higher binding affinity than the wild type.

I find this study an excellent contribution to the literature: it is a sorely needed performance comparison using real-world protein design scenarios with experimental validation. The data set here generated (~250 instances) will by itself be incredibly useful for the ML community for devising and testing new architectures, and protein designers for doing in-silico testing of new approaches. It also sheds some light on the important problem of extrapolation for ML models,

compares different architectures and proposes interesting hypotheses of why some work better than others in protein design. For all these reasons, this study deserves publication, in my opinion. However, I would require several changes:

Major changes:

The authors say that "We observed notable differences in the sequences designed by each model, suggesting each architecture prioritizes distinct regions of the landscape", and also state that "random parameter initialization can greatly influence model extrapolation". Why aren't the GCN and FCN tested with different initializations? One would expect similar results for them. Please check GCN and FCNs with different initializations.

Have the authors tried a different recommendation method other than SA? The inherent stochasticity of SA can play a role in the final design recommendation. Please do try an additional method (e.g. genetic algorithms or parallel tempering) and check if the differences in sequence designs are larger or smaller than those obtained from different initializations.

Minor changes:

Please add page and line numbers to the manuscript. It makes the life of reviewers much easier, and it is in your best interest to keep them happy.

The phrase "Machine learning (ML) accelerates the protein engineering process by integrating experimental data into predictive models to direct the landscape search" is a bit simplistic. Biophysical mechanistic models also integrate experimental data into predictive models, but do so much more slowly because they require a human to improve the biophysical model. The hallmark of ML is to be able to improve the models in an automated fashion, without the need of a human. Please explain hamming distance for the general reader.

Figure 3e does not really show that "all model architectures succeed in designing functional GB1 variants with five and ten mutations (Fig. 3e)." Fig. 3e rather focuses on which of the designs are high, medium or low binding. Please correct.

"We chose ten designs with 40-50 mutations that displayed but did not show IgG binding and we chose another ten designs with 5-20 mutations that showed IgG binding." Please explain the rationale of choosing designs with 40-50 mutations for the first half and designs with 5-20 mutations for the second half.

In figure 4a, please mark the models that produce all the five high binding designs.

Please discuss briefly how ML compares to the use of biophysical models for protein design.

Please correct "show good reproducibly between experimental replicates and internal standards" to "show good reproducibility between experimental replicates and internal standards".

It seems a bit surprising that "parameter-sharing convolutional models could design folded, but non-functional, proteins with sequence identity as low as 10% from wildtype, suggesting these models are capturing more fundamental biophysical properties related to protein folding." since no information on folding has been included in the training set. Could you comment further on this?

Also, "we postulate the parameter sharing architectures may have focused more on learning general rules of protein folding and, in the process, ignored IgG binding activity." This is a bit perplexing since the final loss involved only binding activity, no folding information (i.e. "display"). Could you comment on this too?.

Response to Reviewers

We would like to thank the editors and reviewers for their helpful comments and suggestions. We were encouraged by their positive assessment of the work and also noted several areas that required revision to improve the clarity and rigor of the manuscript. In the revised manuscript we have:

- Added many details to make the approach and results more clear
- Performed a more rigorous analysis of simulated annealing convergence and comparison to other optimization methods
- Performed AlphaFold predictions of all designs to provide structural context of results
- Expanded the Discussion to synthesize all our findings and provide final recommendations
- Discussed how our approach compares to pretrained models and biophysical models

We believe the manuscript is greatly improved as a result of these revisions. We have addressed all the reviewer comments and our responses are provided below.

Reviewer #1 comments:

The authors utilize various neural networks to design protein sequences and evaluate the generated sequences based on sequence statistics and experimental validation results. The experimental validation is well-structured, testing both the expression and IgG binding of sequences derived from different methods. The experiment results interestingly reflect different inductive biases of neural networks. Despite the model and data being similar to a previously published paper, the authors introduce some novel ideas.

We thank the reviewer for the positive assessment of our study design as well as the positive remarks about the experimental contributions.

However, I question the abstract's conclusion: "We find simpler models excel in local extrapolation to design high fitness proteins". Figure 3e suggests that the "EnsC" method is comparable, if not superior, to simpler LR/FCN models. The "EnsC" method, which combines 100 CNNs, should be considered a CNN method that eliminates aleatoric uncertainty. The figures reveal that a single GCN outperforms a single CNN. So, why not test a model that combines 100 GCNs? This could potentially be the most robust model.

We thank the reviewer for bringing up this important point. We agree EnsC is the most advanced method and likely displayed the best overall design performance. In this work, we wanted to systematically test how different modeling approaches affect design performance. If we directly compare two modeling approaches, the differences in their performance can be attributed to the differences in their underlying architectures. For example, the improved performance of FCN over LR can be attributed to FCN accounting for nonlinear interactions between residues. The effect of ensembling can be evaluated by comparing a single CNN to an ensemble of CNNs. We chose to use CNNs because they showed the best prediction performance in our previous work (Gelman et al. PNAS 2021). We found LR outperformed all three single CNNs and this is how we concluded that "simpler models excel in local extrapolation to design high fitness proteins." We aimed to draw conclusions without exhaustively searching the exponentially large combinatorial space of models. Based on our results, it's logical to conclude that an ensemble of GCNs would outperform an ensemble of CNNs, and the best overall model would likely be an ensemble of FCNs. We have added a sentence to the abstract describing how model ensembles improve performance. We have additionally expanded the Discussion to describe the effects of ensembling.

The authors should directly compare the additive model, as the inductive biases of LR and FCN are established to capture the additive mutational effects. An additive model for predicting multiple mutants has been shown to perform better than more complex methods (PMID: 37591206).

The reviewer raises a good point that additive models are known to perform well when predicting multi-mutants. The LR model and the Addition model (from PMID 37591206) are both additive in the sense that they do not consider interactions between amino acids and fitness of multi-mutants is calculated by summing individual amino acid contributions. The difference between the two comes down to how the individual amino

acid contributions are estimated. The Addition model simply considers the mutant's observed fitness difference from wildtype, while LR estimates an amino acid's contribution, marginalized over all observed sequence contexts. These two estimates are identical if trained on single mutant data and usually very similar when trained on multi-mutant data. We have expanded the Discussion to describe the relationship between LR and the Addition model and included a new supplementary figure showing how these two approaches are nearly identical (Fig. S14).

Is there any consistency between the results of the retrospective experiment (Figure 1) and the prospective study results (Figure 3)?

We find the lack of consistency between the retrospective and prospective experiments to be one of the most important results of this work. This discrepancy highlights how evaluations on a fixed, held-out test set are insufficient and the need to design and experimentally test sequences to evaluate model extrapolation. We find LR underperforms versus the neural network models on retrospective analysis (Fig 1), but outperforms many neural network models on the prospective design experiments (Fig 3). Looking more closely at the differences between the neural network models, there is some consistency between Fig 1d and Fig 3e, where FCN > ensembles > CNN/GCN. We have expanded the Discussion to highlight the similarities and differences between the two experiments.

The conclusions drawn seem scattered. It would be more practical for the authors to provide final recommendations on which model to use under different circumstances. Additionally, when making the decision, the diversity of the generated sequence should be taken into account.

We thank the reviewer for pointing out our paper's lack of a cohesive conclusion. Our large-scale design and screening experiment revealed many important findings to advance the field of protein engineering, but we did not bring all the findings back together to give an actionable final recommendation. We have added a paragraph to the Discussion to synthesize the results into a final recommendation. We also mention the caveat that these recommendations may depend on the particular protein of interest, the function being assayed, and the quality and size of the training data.

Reviewer #2 comments:

Fahlberg et al explore multiple neural network architectures to assess their ability to extrapolate from training data on GB1 protein binding to IgG. The training data is single and double mutants. The extrapolation is done by exploring quite extreme mutants – all the way to 50. This is enabled via a yeast display system. Specific sequences verified separately for binding affinity.

Some of the details that should be present in the main text that were a bit hard to figure out for me: GB1 is a 56 residue protein (?), the training data is from a slightly different experiment, the models take the proteins in a fixed-length one-hot sequence, mutations cannot have gaps (?).

Overall, I like this paper. The methods are powerful and it's great to see work on answering a fundamental question. The analyses are well-done and the data will be useful to explore. There are a few issues for me with the concept and then I have some minor technical critiques.

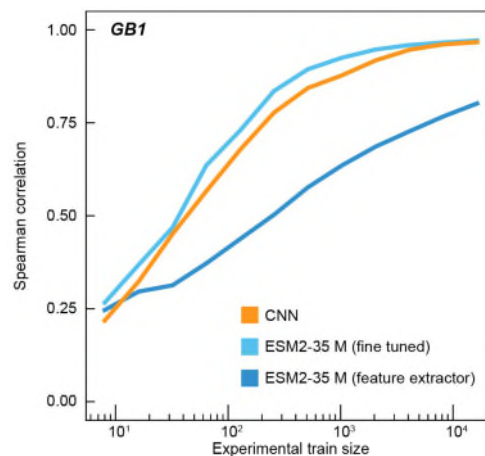
We thank the reviewer for their positive assessment of our work and for pointing out key missing details. We have updated the main text to include the sequence length of GB1, more details of the training data, and a clearer description of how sequences are input into the models.

My understanding is that the goal of this paper is to evaluate supervised deep learning's ability to extrapolate away from training data for one protein. However, most current supervised learning models use something like ESM or UniRep as a pre-trained featurization of proteins and then do supervised learning with those models. These models are state of the art in most tasks and usually generalize across sequence quite well. They are

trained on billions of sequences across most known proteins. The models used in the paper -CNN/MLPs – these are a bit dated for this field. I'm not saying this undermines their usefulness, but just that the impact is not quite there for the field because most of the research is on different models trained across many sequences.

We thank the reviewer for bringing up pre-trained representation models such as UniRep and ESM. While these models are widely used, there is no strong evidence or consensus in the field that they are superior to simpler LR/FCN/CNN models. Many recent protein engineering papers use LR/FCN/CNN models: PMIDs 33574611, 35039677, 32250622. Very recent work found CNNs to be competitive with, and occasionally superior to, transformers for masked language model pretraining [PMID 38428432]. While other recent papers report better performance with non-pretrained models: PMIDs 37591206, 34416172, 3503967.

Pretrained protein language models such as UniRep and ESM2 are commonly used as feature extractors where an amino acid sequence is fed into the model to produce a fixed protein representation that is then input into a learnable top model for supervised learning. For GB1, we have found the ESM2 feature extractor approach significantly underperforms CNNs and the full model must be fine tuned to improve performance (see figure below). The fine-tuned ESM2 shows modest improvements over the CNN in low data settings ($N < 1000$) but these differences are negligible beyond 10,000 data points. The models used in our work were trained on nearly 500,000 sequence-function examples.



We have added a section to the Discussion describing pre-trained representations, how they relate to our models, and interesting future research directions exploring these models.

Also, I question the premise here – what is GB1 protein? If I have mutated every single residue is it still GB1? Some of the comments in the article – like “Even a small protein like GB1 has over 10^{70} possible sequence configurations” seem to imply any length 56 protein is GB1. This is a bit of semantics, but practically it is probably necessary to define what you consider “success.” Like if generalization means predicting accurately across all sequence space of length 56 after training around one sequence, then fine – just be clear.

We agree with the reviewer that the overall premise was unclear. Our model is trained on IgG binding data centered around the wildtype GB1 sequence, and thus should capture how individual residues contribute to GB1’s three dimensional fold and IgG binding. When extrapolating, we expect the model to design new sequences that preserve the same fold and function. A 56-residue protein can take on 20^{56} sequence configurations and some small fraction of these sequences (but still a very large number) will fold and bind IgG similar to GB1. As the reviewer mentioned, “success” would be accurately predicting these GB1-like sequences over this 20^{56} sequence space. We have gone through the manuscript and clarified all instances where the prediction premise was unclear.

To address this issue, one way could be to include in your analysis an ESM fine tuned model or MPNN. It would be valuable to show their ability to generalize – especially to sequences only accessible in laboratories and not found in typical MSA. Maybe this is better for a follow-up paper though (which I would happily review!!)

This is an intriguing suggestion to use ESM/ProteinMPNN to generalize to distant, non-natural sequences. These models have a more general understanding of protein structure that when integrated with labeled sequence-function data could design completely novel solutions. More recent (to be published) work from our lab is using fine-tuned ESM2 and a ProteinMPNN filter to design phage tail fibers. We are excited to continue exploring these powerful new approaches to expand the capabilities of AI-driven protein design.

Based on those concerns, I believe the scope of this work is just a bit narrow. These models aren't as relevant today to what is at the frontier of protein engineering and the narrowness of one protein/one-task do not support the title/claims. The methods you've built and the approach are excellent and I believe with just some small changes by choosing different models and maybe including another 1-2 systems will make a landmark paper. As someone who works on protein engineering, the biggest open question I currently have is assessing well these models generalize to sequences far from nature.

We understand the reviewer's perspective and how our work doesn't directly address the ability to design radically new proteins with non-natural functions. We feel the field is still far from realizing this grand challenge, especially for complex functions such as enzyme catalysis. A vast majority of protein engineering that is currently ongoing in academia and industry is still taking the directed evolution one protein/one-task approach. Here, machine learning has proved incredibly useful for library design and learning the landscape from screening data to accelerate the protein engineering process. Our work fits into this larger movement within the field by rigorously evaluating common model architectures, the effect of ensembling, and how far these models can reliably extrapolate on the fitness landscape.

Here are a few small technical considerations:

1. Some convergence assessment on SA would be great to see. I'm surprised you didn't see some pathological sequences at high mutation count (e.g., I often see poly-W or other weird stuff come out of optimization with models). Could you be running the SA too short?

This is a great suggestion. We have included a new supplementary figure (Fig. S2) with representative simulated annealing progress curves showing convergence to similar fitness levels. Many of the high-mutation-count designs did have pathological sequences such as large numbers of Cys and Trp residues. In future iterations, these sequences could be filtered out with sequence composition constraints.

2. Would love to see some BLOSUM analysis instead of hamming – are the hydrophobic regions being rearranged? Can you do AF2 and align some of these extreme mutants?

This is a great suggestion because BLOSUM provides a more nuanced view of amino acid similarities and the conservativeness of designed mutations. We have included a new supplementary figure (Fig. S6) showing how conservative the designed mutations were for each model across all positions. From this analysis we find the models tend to design non-conservative mutations but also avoid making mutations at hydrophobic core residues.

We thank the reviewer for suggesting we perform AF2 predictions on the designed GB1 variants. We found variants with more mutations tended to have structures that deviate from the wildtype GB1 fold. It was notable that the LR and FCN models tended to design variants with more compact GB1-like folds. We additionally performed a UMAP analysis to visualize the relationships between the predicted structures and found the structures tend to cluster by the designs' functional status with designs that bind IgG tend to have structures similar to WT GB1. We have included these AF2 predictions as a new main text figure (Fig. 4) because we felt this analysis provided valuable insight into the designs' structure and helped interpret the results.

3. Please include a few more details needed to make sense of the results – sequence length of GB1, input representation, etc

We have included more details to make the approach and results clear.

4. The cited Gelman paper explores a few architectures, can you clarify when saying the architecture from that paper was used

We used four different architectures from the Gelman paper. We have further explained these models in the Results section.

Reviewer #3 comments:

The authors present the results from training a panel of neural network architectures on GB1-IgG binding data for single and double mutants, generating thousands of GB1 designs with up to 50 mutations, and systematically evaluating the models' ability to extrapolate and produce designs which are both foldable and functional. This is a systematic work which uses a relatively well-understood model protein system to understand how various neural network architectures perform when attempting to extrapolate to generate functional variants with far more mutations than are present in any single genotype of the training data. As such, the scope of this work can be considered to be examining the performance of various models on a simple and well-characterized model protein system with a relatively simple but comprehensive training data set.

Overall the experiments of this work appear to be well-designed and performed, leading to reasonable conclusions about what models of varying complexity are capable of in the space of protein design. The work and interpretations appear generally valid and the authors appropriately express value and limitations of their findings.

The work is limited in scope beyond the relatively large number of genotypes generated and tested, in that there is no attempt to design new molecular function. The designed molecules must simply maintain or improve the native function in order to survive the selection. Given the wealth of methods that exist for designing, selecting, and screening molecular binders, this work will likely not have a major impact on the design of this class of molecules. Given that there is no attempt to design new or altered molecular function, such as binding of a noncognate target, into GB1, this work also represents an interesting but somewhat incremental advance in the understanding of the ability of neural network to extrapolate beyond their training data for protein design.

We thank the reviewer for raising the important point regarding extrapolation to new functions such as binding noncognate targets. We did not intend to develop a new method for engineering molecular binders, but instead evaluated the ability of sequence-function models to make predictions at increasing distances from their training data. Here “function” could be any property of a protein including binding affinity/specificity, enzyme activity, spectroscopic properties, and thermodynamic stability. This general definition of function allows the models to be applied to nearly any protein of interest. We are not extrapolating the models to new functions, but we are extrapolating to distant sequences with substantially improved molecular properties. This setting is similar to directed evolution because it optimizes an existing function and is currently the most widely used approach for engineering proteins.

The tendency of the CNNs and the GCN to target regions of positive epistasis is encouraging, suggesting that the models learn the relationships between positively interacting positions and can exploit these interactions to create better protein designs. This may partly explain why these models are good at promoting protein folding, since positive epistatic interactions between specific amino acids often play an important role in this process and in the stability of a protein structure.

The reviewer brings up an interesting idea that positive epistasis between specific residues is important for folding and stability, and the fact that CNN/GCN models excel at learning these positive interactions allows them to better design proteins that fold. We have expanded on this concept in the Discussion.

An interesting finding is that the more sophisticated neural models were, in general, not able to maintain protein function when forced to introduce many mutations, and instead could only maintain protein folding. Ensemble predictor “EnsC” was the most successful at maintaining protein function with up to 20 mutations, which may be the most significant finding of this work for the future of protein engineering.

Yes, we found this observation that CNN/GCN models were unable to maintain function at large extrapolation distances to be one of the most interesting and significant findings of our work. We believe the intrinsic inductive biases of these models primes them to learn the signatures of protein folding. Interestingly, an ensemble of CNNs appears to perform the best for designing function, suggesting that averaging over different model initializations can refocus the models toward function. We agree EnsC was the best overall design model and would recommend this approach for other protein engineering applications. We have updated the discussion to include a new paragraph that distills all our findings to give recommendations for future protein engineering work. We have also included our ensemble results in the paper’s abstract.

The authors published previous results similarly, training models on the same set of single- and double-mutants, but only evaluated model performance on held-out single- and double-mutants. Thus, the scope of this work is relatively narrowly defined, only looking at the ability of the models to extrapolate ‘deeper into sequence space’, but not fundamentally changing the experimental paradigm.

The reviewer is correct that the current work builds off our prior work in Gelman et al. *PNAS* (2021). Previously we developed the neural network architectures and evaluated the models on held-out single and double mutants, but performed little analysis of the models’ ability to design new and improved proteins. The models themselves are of little use without applying them to design proteins. The current work performs a systematic and rigorous evaluation of the different architectures’ design performance, including high-throughput experimental testing of hundreds of designs. An interesting finding from our current work is the inconsistency between extrapolation performance on held-out three/four-mutants and the design performance at five mutations and beyond. In particular, the LR model underperformed at predicting 3/4-mutants (Fig 1de), but was the best model at designing 5-mutants (Fig 3e). This highlights how evaluations on held out data are insufficient and the need to design and experimentally test sequences to evaluate model performance.

Overall I find this work to be rigorous and convincing in its measured claims, but might expect a more ambitious scope, novel methods, or more groundbreaking findings for a paper that would be published in Nature Communications. It may be better suited for a quality topical journal.

We thank the reviewer for commenting on our work’s rigor and the validity of our results. We feel our work is appropriate for Nature Communications because it presents the first in-depth analysis of neural network extrapolation for protein design. Our work highlights the disconnect between traditional machine learning benchmarks and model performance in real-world design settings. We find each neural network architecture’s intrinsic inductive biases prime them to learn different aspects of the protein fitness landscape, which greatly affects their design performance. This work lays an important foundation for ML-guided protein design that can be readily leveraged across multiple research domains and industries.

Reviewer #4 comments:

In this manuscript, the authors evaluate the capacity of artificial neural networks (the workhorse of modern machine learning) to extrapolate beyond their training data in protein design. The topic of extrapolation is an important one for machine learning since it determines its effectiveness, and the topic of protein design is important because of its sundry applications and because it significantly challenges ML approaches with a very large possible phase space. The authors examine several models and architectures of neural networks: a linear regressor (LR), a fully connected neural network (FCN), three convolutional neural networks with different initializations (CNNs), a graph convolutional neural network (GCN), and two ensemble approaches of 100 CNNs using their median and 5th percentile (EnsM and EnsC). Their goal is to predict the display of protein G (GB1) in the yeast cell surface (“display”) and its binding to the antibody immunoglobulin G (IgG, “binding”). As a first step, they use literature data from a library measuring fitness (equivalent to binding) for protein variants that encompass most 1 and 2 mutation variants to train their models. When testing with another library that has variants with 3 and 4 mutations, model accuracy drops dramatically in the extrapolated regime, but the Spearman’s correlation remains significantly above 0 (around 0.4), suggesting potential for model-guided design at or beyond 4 mutations. They then developed a large-scale protein design computational pipeline that leverages simulated annealing (SA) to identify high fitness peaks. They used this pipeline to computationally propose 41 sequences for each model, so as to compare their performance, finding notable differences in the sequences designed by each algorithm. A third part of the paper involves experimentally testing these proposed protein designs. For this purpose, they used FACS and they were able to show which protein designs displayed in the cell surface (display) and which ones showed binding to IgG (binding, which correlates to the fitness value used in the literature data used above). They find that all models show the ability to extrapolate to 2.5-5x more mutations than the training data (from 1, 2 to 5, 10 mutations), but the design performance decreases sharply with further extrapolation. The simpler LR and FCN models outperformed the more sophisticated CNN and GCN models for designing sequences that bind, whereas the ensemble models showed similar performance. The availability of both “display” and “binding” data allows the authors to distinguish between models that predict protein folding well but not binding and models that are good binding predictors. Finally, they did a more detailed structural and quantitative analysis of twenty designs: ten with 40-50 mutations that displayed but showed no binding and then with 5-20 mutations that showed binding. Almost all displaying designs did so more than for the wild type case, and all the high-binding designs showed higher binding affinity than the wild type.

I find this study an excellent contribution to the literature: it is a sorely needed performance comparison using real-world protein design scenarios with experimental validation. The data set here generated (~250 instances) will by itself be incredibly useful for the ML community for devising and testing new architectures, and protein designers for doing in-silico testing of new approaches. It also sheds some light on the important problem of extrapolation for ML models, compares different architectures and proposes interesting hypotheses of why some work better than others in protein design. For all these reasons, this study deserves publication, in my opinion. However, I would require several changes:

[We thank the reviewer for their strong endorsement of our work.](#)

Major changes:

The authors say that “We observed notable differences in the sequences designed by each model, suggesting each architecture prioritizes distinct regions of the landscape”, and also state that “random parameter initialization can greatly influence model extrapolation”. Why aren’t the GCN and FCN tested with different initializations? One would expect similar results for them. Please check GCN and FCNs with different initializations.

[The reviewer raises a good point that we did not test all model architectures with multiple initializations and ensembling. This was done to limit the combinatorially large number of design scenarios that would need to be experimentally tested. We focused on the CNN models because they showed the strongest performance in our prior work \(Gelman et al. PNAS 2021\) and we felt observations from CNNs \(e.g. the benefit of ensembling\) would likely apply to the FCN/GCN models.](#)

We have retrained multiple LR, FCN, CNN, and GCN models with different initializations, designed sequences with each model, and compared the similarity of the designs within and between model architectures. We find variation within each model, but the sequences designed by a given architecture are more similar to other initializations of that architecture than other architectures. In other words, within architecture similarity is greater than between architecture similarity. This indicates the model architecture itself is a driving factor for the differences in design performance we observed. We have included this analysis in a new supplementary figure (Fig. S5).

Have the authors tried a different recommendation method other than SA? The inherent stochasticity of SA can play a role in the final design recommendation. Please do try an additional method (e.g. genetic algorithms or parallel tempering) and check if the differences in sequence designs are larger or smaller than those obtained from different initializations.

The reviewer raises a good point that SA is a stochastic optimization method that can become trapped in local optima, leading to suboptimal solutions. This inherent randomness was actually a desirable feature of our method because it resulted in diverse sequence designs that are all predicted to have high fitness. This provided many different design hypotheses for us to test experimentally. We have additionally implemented a parallel tempering (PT) method with six parallel MCMC samplers at different temperatures that randomly exchange designs and return the best design observed over all samplers over all iterations. We found both SA and PT found highly optimized designs that had similar fitness distributions. SA was slightly more effective for optimizing the LR and FCN models, while PT was more effective for optimizing the CNN and GCN models. For future designs, we would recommend trying both methods and picking the top overall designs from either. We have included this analysis as a new supplementary figure (Fig. S3).

Minor changes:

Please add page and line numbers to the manuscript. It makes the life of reviewers much easier, and it is in your best interest to keep them happy.

We have added page and line numbers to the manuscript.

The phrase “Machine learning (ML) accelerates the protein engineering process by integrating experimental data into predictive models to direct the landscape search” is a bit simplistic. Biophysical mechanistic models also integrate experimental data into predictive models, but do so much more slowly because they require a human to improve the biophysical model. The hallmark of ML is to be able to improve the models in an automated fashion, without the need of a human.

The reviewer is absolutely correct that biophysical modeling approaches also integrate experimental data, but in a less automated manner. We have updated the sentence to emphasize how ML automates the experimental feedback cycle.

Please explain hamming distance for the general reader.

Thank you for bringing this to our attention. We first mention Hamming distance in Figure 3 and have updated the caption to describe how Hamming distance is used to represent sequence similarity between two aligned proteins.

Figure 3e does not really show that “all model architectures succeed in designing functional GB1 variants with five and ten mutations (Fig. 3e).” Fig. 3e rather focuses on which of the designs are high, medium or low binding. Please correct.

Thank you for identifying this point of confusion. We have changed the text to say “all model architectures succeed in designing high-binding GB1 variants with five and ten mutations (Fig. 3e).”

“We chose ten designs with 40-50 mutations that displayed but did not show IgG binding and we chose another ten designs with 5-20 mutations that showed IgG binding.” Please explain the rationale of choosing designs with 40-50 mutations for the first half and designs with 5-20 mutations for the second half.

We apologize because our initial description was not very clear. Our interesting designs for further characterization focused on either designs that were better than WT GB1 or very distant from WT GB1. Our high-throughput screen did not find distant variants that bound IgG. Our designs consisted of (1) five 5/10-mutants from the “high bind” population, (2) five 20-mutants from the “bind” population, and (3) ten 40/50-mutants from the “display” population. We have updated the main text to make this more clear.

In figure 4a, please mark the models that produce all the five high binding designs.

This is a great suggestion. We have updated the figure to note the models used to design the fine high binding designs.

Please discuss briefly how ML compares to the use of biophysical models for protein design.

We thank the reviewer for bringing up biophysical models and the need to contrast with ML. We have added a paragraph in the Discussion comparing ML to biophysical models.

Please correct “show good reproducibility between experimental replicates and internal standards” to “show good reproducibility between experimental replicates and internal standards”.

Thank you for bringing this to our notice. We have corrected the spelling error.

It seems a bit surprising that “parameter-sharing convolutional models could design folded, but non-functional, proteins with sequence identity as low as 10% from wildtype, suggesting these models are capturing more fundamental biophysical properties related to protein folding.” since no information on folding has been included in the training set. Could you comment further on this? Also, “we postulate the parameter sharing architectures may have focused more on learning general rules of protein folding and, in the process, ignored IgG binding activity.” This is a bit perplexing since the final loss involved only binding activity, no folding information (i.e. “display”). Could you comment on this too?.

Yes, we found this result to be quite interesting. For many proteins including GB1, folding is a prerequisite for function. The protein must fold into a stable structure before it can perform its function. For this reason, folding is implicitly captured by a functional assay. Every GB1 variant that binds IgG is also a stably folded protein. Function is often dictated by a handful of specific residues, while folding is a more global property of many residues interacting. Statistically, there are many more residues that contribute to folding versus function, so most DMS data sets have a major “folding” component.

The inductive biases of LR and FCN models directly capture how specific residues contribute to function. In contrast, parameter sharing models such as CNNs/GCNs learn patterns that map to the output function. We postulate the inductive biases of CNNs/GCNs to learn patterns, in addition to the fact IgG binding has a major folding component dictated by underlying sequence patterns, causes the CNN/GCN models to pick up on

folding signals. In other words, the measured IgG binding function has both protein-protein binding and folding components and the CNN/GCN models are more primed to learn the folding component. We have elaborated on this point in the Discussion.

Reviewer #1 (Remarks to the Author):

In this revision, the authors have adequately addressed the issues raised in the previous round of review. I feel that the manuscript is appropriate for publication in its current form.

Reviewer #3 (Remarks to the Author):

Overall, the authors did a good job of addressing the specific concerns of Reviewers 2 and 3 with regard to specific asks, including SA analysis, AF2 predictions, and others. These changes have improved the manuscript and made the overall analysis more detailed. The concerns that Reviewers 2 and 3 both raised around the narrow scope of the work remain, but the work is solid and can be published in Nature Communications if the editors do not have an issue with the reviewers' comments on scope.

Reviewer #4 (Remarks to the Author):

The authors have dutifully answered most of my requests and comments.

However, I disagree with one of their conclusions, as derived from Fig. S5: "We find variation within each model, but the sequences designed by a given architecture are more similar to other initializations of that architecture than other architectures".

For 30, 40 and 50 mutations GCN, and GCN1-4 show similar distances to each other than to CNN1-4 and EnsM and EnsC. Hence they are not "more similar to other initializations of that architecture than other architectures". Please correct this statement or show proof of the opposite.

Also, please correct line 353: "We postulate the inductive biases of CNNs/GCNs to learn patterns" to "Line 353: We postulate the inductive biases of CNNs/GCNs to learn patterns"

Reviewer #4 (Remarks on code availability):

I haven't had time to review the code.

Response to Reviewers

We would like to thank the editors and reviewers for their helpful comments and suggestions. We have updated the manuscript to address the last few reviewer suggestions.

Reviewer #1 (Remarks to the Author):

In this revision, the authors have adequately addressed the issues raised in the previous round of review. I feel that the manuscript is appropriate for publication in its current form.

We thank the review for their positive assessment of our work and the revision.

Reviewer #3 (Remarks to the Author):

Overall, the authors did a good job of addressing the specific concerns of Reviewers 2 and 3 with regard to specific asks, including SA analysis, AF2 predictions, and others. These changes have improved the manuscript and made the overall analysis more detailed. The concerns that Reviewers 2 and 3 both raised around the narrow scope of the work remain, but the work is solid and can be published in Nature Communications if the editors do not have an issue with the reviewers' comments on scope.

We thank the review for their positive assessment of our work and the revision.

Reviewer #4 (Remarks to the Author):

The authors have dutifully answered most of my requests and comments.

However, I disagree with one of their conclusions, as derived from Fig. S5: "We find variation within each model, but the sequences designed by a given architecture are more similar to other initializations of that architecture than other architectures". For 30, 40 and 50 mutations GCN, and GCN1-4 show similar distances to each other than to CNN1-4 and EnsM and EnsC. Hence they are not "more similar to other initializations of that architecture than other architectures". Please correct this statement or show proof of the opposite.

We thank the reviewer for bringing up the point that our statement is not generally true. Our statement is true for all models except the GCN model. We have updated the text to state:

"We found the sequences designed by the LR, FCN, and CNN architectures are more similar to other initializations of that architecture than other architectures, indicating the model architecture itself is a driving factor for the differences in the designs. The GCN models show high sequence variability across different model initializations."

Also, please correct line 353: "We postulate the inductive biases of CNNs/GCNs to learn patterns" to "Line 353: We postulate the inductive biases of CNNs/GCNs to learn patterns"

We have fixed this typo.

Reviewer #4 (Remarks on code availability):

I haven't had time to review the code.

We thank the reviewer for taking the time to read our manuscript.