Improving Generalizability of Protein Sequence Models With Data Augmentations

Hongyu Shen[1,2], Layne C. Price[1], Taha Bahadori[1], & Franziska Seeger[1]

[1]Amazon.com
Seattle, WA 98109, USA
{hongyus,prilayne,bahadorm,fseeger}@amazon.com

[2] ECE Department, UIUC
Champaign, IL 61820, USA
hongyu2@illinois.edu

Abstract

Protein sequence modeling typically does not use randomized data augmentation procedures during training due to the unpredictable functional changes introduced by even simple sequence modifications. However, in this paper, we empirically explore a set of simple string manipulations, when fine-tuning semi-supervised protein models. We compare to the Tasks Assessing Protein Embeddings (TAPE) baseline models, with methods that vary from the baseline methods only in the data augmentations and representation learning procedure, and demonstrate improvements between 1% and 41% to the baseline scores on the TAPE validation tasks, with both linear evaluation and full fine-tuning on downstream tasks. We find the most consistent results using domain-motivated transformations, such as amino acid replacement, as well as subsampling of the protein sequence. In rarer cases, we even find that information-destroying augmentations, such as random sequence shuffling, can improve performance.

1 Introduction

In this paper, we take the uncertainty arising from the unknown effect of simple data augmentations in protein sequence modeling as an empirical challenge that deserves a robust assessment. We focus on fine-tuning previously published self-supervised models that are typically used for representation learning with protein sequences, *viz.* the transformer-based methods of Rao et al. [2019]. We demonstrate that the protein sequence representations learned by fine-tuning the baseline models with data augmentations results in relative improvements between 1% (secondary structure accuracy) and 41% (fluorescence ρ), as assessed with linear evaluation for all TAPE tasks we studied. When fine-tuning the same representations during supervised learning on each TAPE task, we show significant improvement as compared to baseline for 3 out of 4 TAPE tasks, with the fourth (fluorescence) within 1σ in performance. We also study the effect of increasingly aggressive data augmentations: when fine-tuning baseline models with contrastive learning [Hadsell et al., 2006, Chen et al., 2020] we see a local maximum in downstream performance as a function of the quantity of data augmentation, with "no augmentations" generally under-performing modest amounts of data augmentations. Conversely, performing the same experiments but using masked-token prediction instead of contrastive learning, we detect a minor trend of decreasing performance on the TAPE tasks as we more frequently use data augmentations during fine-tuning. We interpret this as evidence that contrastive learning techniques, which require the use of data augmentation, are important methods that can be used to improve generalizibility of protein models.

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

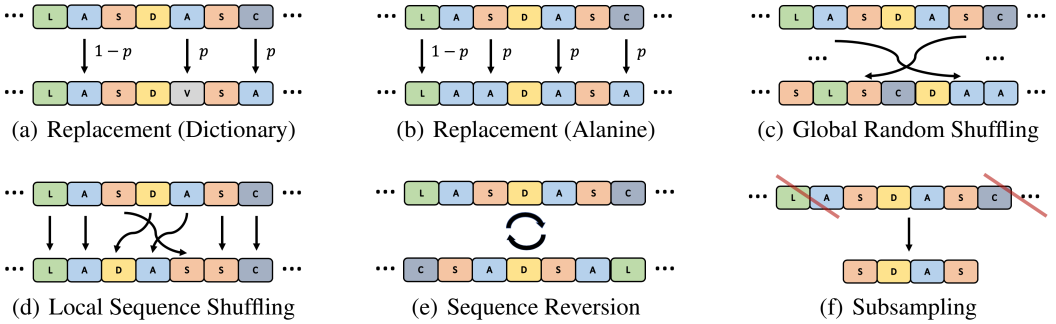

Figure 1: Diagram of data augmentations. We study randomly replacing residues (with probability p) with (a) a chemically-motivated dictionary replacement or (b) the single amino acid alanine. We also consider randomly shuffling either (c) the entire sequence or (d) a local region only. Finally, we look at (e) reversing the whole sequence and (f) subsampling to a subset of the original.

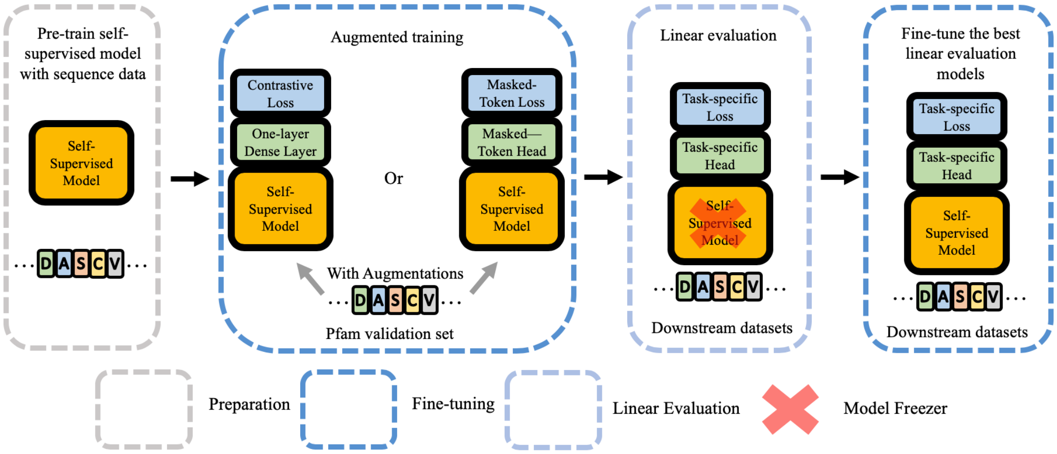

Figure 2: Diagram of experimental approach (see Sect. 2).Dashed boxes indicate the different 4 steps. In each box, we include the general model architectures, with major sub-modules in different colors. The model freezer freezes the semi-supervised model weights during linear evaluation.

2 Method

Evaluation procedure.— To attempt to control external variables, we study the following restricted setting; we provide the procedural diagram in Figure 2. This study includes four major steps: **1. Baseline:** A self-supervised model M_0 is directly obtained from Rao et al. [2019], without modification, which is trained with masked-token prediction on Pfam protein sequence data [El-Gebali et al., 2019]. **2. Augmented training on validation set:** We fine-tune M_0 on subsets of the Pfam dataset, given a set of pre-defined data transformations. We define M_{aug} as the final fine-tuned model from the starting point M_0. We adopt two methods during fine-tuning — a contrastive task with SimCLR loss described in Chen et al. [2020] and a masked-token task (exponentiated cross entropy loss) — as well as different combinations of data augmentations. **3. Linear evaluation on TAPE:** To assess the representations learned by M_{aug}, we evaluate performance on four TAPE downstream training tasks [Rao et al., 2019]: stability, fluorescence, remote homology, and secondary structure. For consistency, we use the same training, validation, and testing sets. The first two tasks are evaluated by Spearman correlation (ρ) to the ground truth and the latter two by classification accuracy. Descriptions of the four tasks can be found in Rao et al. [2019]. We perform linear evaluation by freezeing the self-supervised part (Fig. 2). **4. Full fine-tuning on TAPE:** For the best-performing augmented models and associated data augmentations in the linear evaluation task (either $M_{\text{aug}}^{\text{CL}}$ or $M_{\text{aug}}^{\text{MT}}$), we further study how the models improve when allowing the parameters of M_{aug} to vary along with the linear model during the task-specific supervised model-tuning.

Table 1: Best linear evaluation results. **Bold** refers outperformance to the TAPE baselines; and **red** is the task-wise best-performing result. *MT* and *CL* refer to training with masked-token prediction and contrastive learning, respectively. Stability and fluorescence are scored by Spearman correlation and remote homology and secondary structure by classification accuracy. Bootstrap standard deviations (σ) are reported per task by taking the maximum error found for any of the models by bootstrapping the testing results 5,000 times, with convergence after $\sim 3,000$ samples.

Scenario	Stability	Fluor.	Remote Homology	2^{nd} Structure
MT: TAPE Baseline	0.498	0.256	[0.200, 0.625, 0.231]	[0.699, 0.756, 0.727]
MT: No Aug. ($\gamma = 0$)	**0.534**	**0.275**	[**0.206, 0.636, 0.241**]	[**0.706, 0.771, 0.729**]
MT: Best Aug.	**0.516**	**0.301**	[**0.207**, **0.637**, **0.241**]	[**0.716**, **0.771**, **0.735**]
CL: No Aug.	**0.512**	**0.334**	[0.146, 0.529, 0.163]	[0.667, 0.725, 0.678]
CL: Single Aug.	**0.562**	**0.343**	[0.183, **0.720**, **0.243**]	[0.700, 0.757, 0.727]
CL: Pairwise	**0.537**	**0.361**	[**0.219**, **0.718**, **0.255**]	[**0.702**, **0.759**, 0.726]
Bootstrap 1σ (<)	±0.011	±0.006	±[0.015, 0.014, 0.012]	±[0.021, 0.009, 0.015]

Data augmentations.— We focus on random augmentations to protein primary sequences, both chemically and non-chemically motivated (see Fig. 1). **Replacement (Dictionary/Alanine) [*RD & RA*]:** We randomly and independently replace each amino acid in the primary sequence according to either a replacement dictionary or alanine (A) [Cunningham and Wells, 1989], with a probabiliy p. For Replacement (Dictionary), following French and Robson [1983], we experimented with different pairings, finding little difference in the results; our best results were obtained with the final mappings: [[A,V], [S,T], [F,Y], [K,R], [C,M], [D,E], [N,Q], [V,I]]. **Global/Local Random Shuffling [*GRS & LRS*]:** We reshuffle the protein sequence, either globally or locally. For $S = \{A_i\}_{i=1}^N$, we define an index range $i \in [\alpha, \beta]$ with $\alpha < \beta \leq N$, then replace amino acids A_i in this range with a permutation chosen uniformly at random. We define Global Random Shuffling (GRS) with $\alpha = 1$ and $\beta = N$ and Local Random Shuffling (LRS) with $\alpha \in [1, N-2]$ and $\beta = \min(N, \alpha + 50)$, ensuring at least two amino acids get shuffled. **Sequence Reversion & Subsampling [*SR & SS*]:** For Sequence Reversion, we simply reverse the sequence: given $S = \{A_i\}_{i=1}^N$ we map $i \to i' = N - i$. For Subsampling, we keep A_i for $i \in [\alpha, \beta]$ of the original sequence $S = \{A_i\}_{i=1}^N$, with $\alpha \in [1, N-2]$ and $\beta = \min(N, \alpha + 50)$.

3 Results

Assessing data-augmented representations with linear evaluation.— In Table 1, we see broad improvement when using contrastive learning with data augmentations compared to both baselines for the stability, fluorescence, and remote homology tasks. (see “MT: TAPE Baseline" / “MT: Best Aug." v.s. highest/red numbers in “CL: *" rows). We also see that masked-token prediction has better performance than contrastive learning for all tasks with no data augmentations (“MT: No Aug." v.s. “CL: No Aug."). Overall, we observe that the best results from Table 1 utilize the combination of contrastive learning with pairs of data augmentation for all tasks besides secondary structure prediction. Our interpretation is that augmentation and contrastive learning provide better encoded features that help improving the performance on protein's downstream tasks.

Figure 3 demonstrates our linear evaluation results using contrastive learning based on the composition of pairs of data augmentations. For stability, amino acid replacement (with RD/RA) consistently improves performance compared to the TAPE baseline, as well as to other augmentation strategies. Fluorescence sees improvements using all data augmentations but random shuffling (LRS & GRS). RA & RD result in the best individual performance. For remote homology, it is apparent that subsampling plays an important role in model performance given the improvement it introduces on the three testing sets; the “family" homology level is included. The other remote homology tasks are qualitatively similar. Additionally, we see that subsampling tends to yield better performance than alternatives, with the best performing approach using subsampling alone.

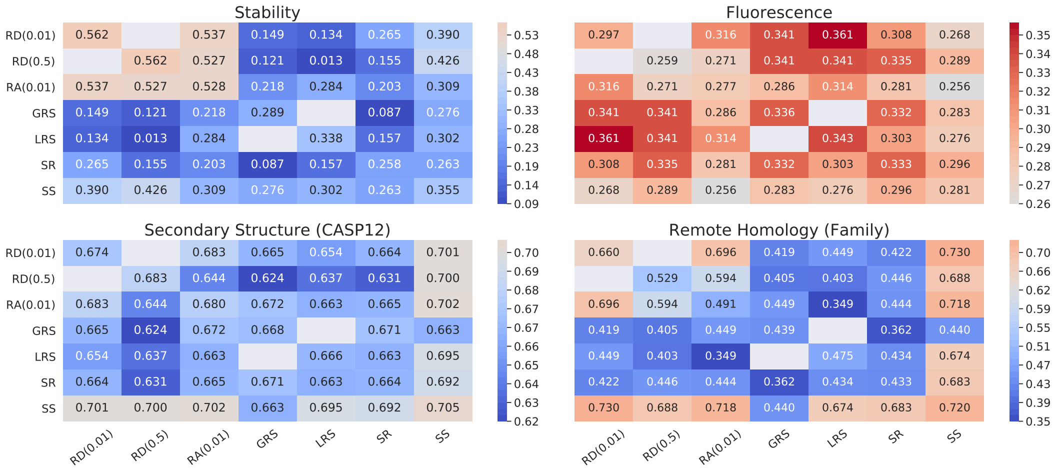

Figure 3: Contrastive learning performance with pairwise & single augmentations in linear evaluation for 4 different tasks. The axes refer to different augmentations, with diagonal being a single augmentation. The values in the heatmaps are correlation (stability and fluorescence) and classification accuracy (remote homology and secondary structure). Missing cells are due to redundancy. TAPE Baseline is colored white in each subfigures; red is better performance, blue is worse.

Table 2: Model fine-tuning results, with associated training method and data augmentation procedure for each task.

Scenario	**Stability**	**Fluor.**	**Remote Homology**	2^{nd} **Structure**
TAPE Best	0.730	0.680	[0.210, 0.880, 0.340]	[0.710, 0.770, 0.730]
Our Best	**0.748±0.005**	0.677±0.004	[0.209±0.015, **0.921±0.008**, **0.377±0.014**]	[**0.711±0.015**, **0.778±0.008**, **0.739±0.003**]
Best Models	CL	CL	CL	MT
Best Aug.	RD(0.01 or 0.5)	RD(0.01) & LRS	RA(0.01) & SS	RA(0.01) & SS

Exploring the best performance via full fine-tuning.— We provide results of the best performing fine-tuned models with the best performing augmentations during linear evaluation (on downstream tasks) and the comparison to the TAPE's original baselines (Table 2), to verify whether the learned representations of the best models provide good initialization points for transfer learning. The fine-tuned, data-augmented models outperform the TAPE baseline on stability, remote homology and secondary structure, which is consistent with results we found during linear evaluation. These models also perform within 1σ on fluorescence, although the large difference in performance between full fine-tuning and linear evaluation indicates that most of the model's predictive capacity is coming from the supervised learning task itself.

4 Conclusion

We experimentally verify that relatively naive string manipulations can be used as data augmentations to improve the performance of self-supervised protein sequence when further optimized for downstream tasks. We demonstrate that, in general, augmentations will boost the model performance, in both linear evaluation and model fine-tuning cases. However, different downstream tasks benefit from different protein augmentations; no single augmentation that we studied was consistently the best. The approach that we have taken, where we fine-tune a pretrained model on the validation set requires significantly lower computational cost than training on the full training set, using data augmentations.

References

T Chen, S Kornblith, M Norouzi, and G Hinton. A simple framework for contrastive learning of visual representations. In *Thirty-seventh International Conference on Machine Learning*, 2020.

BC Cunningham and JA Wells. High-resolution epitope mapping of hgh-receptor interactions by alanine-scanning mutagenesis. *Science*, 244(4908):1081–1085, 1989.

S El-Gebali, J Mistry, A Bateman, et al. The pfam protein families database in 2019. *Nucleic acids research*, 47(D1):D427–D432, 2019.

S French and B Robson. What is a conservative substitution? *Journal of molecular Evolution*, 19(2): 171–175, 1983.

R Hadsell, S Chopra, and Y LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

R Rao, N Bhattacharya, N Thomas, et al. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pages 9689–9701, 2019.