IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 23, 2021, accepted August 23, 2021, date of publication September 3, 2021, date of current version September 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110269

Pre-Training of Deep Bidirectional Protein Sequence Representations With Structural Information

SEONWOO MIN[1,2], SEUNGHYUN PARK[3], SIWON KIM[1], HYUN-SOO CHOI[4], BYUNGHAN LEE[5], (Member, IEEE), AND SUNGROH YOON[1,6,7], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
[2]LG AI Research, Seoul 07796, South Korea
[3]Clova AI Research, NAVER Corporation, Seongnam 13561, South Korea
[4]Department of Computer Science and Engineering, Kangwon National University, Chuncheon 24341, South Korea
[5]Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[6]Interdisciplinary Program in Artificial Intelligence, ASRI, INMC, Seoul National University, Seoul 08826, South Korea
[7]Institute of Engineering Research, Seoul National University, Seoul 08826, South Korea

Corresponding authors: Byunghan Lee (bhlee@seoultech.ac.kr) and Sungroh Yoon (sryoon@snu.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grants funded by the Ministry of Science and ICT under Grant 2018R1A2B3001628 (S.Y.), Grant 2014M3C9A3063541 (S.Y.), and Grant 2019R1G1A1003253 (B.L.); in part by the Ministry of Agriculture, Food and Rural Affairs under Grant 918013-4 (S.Y.); and in part by the Brain Korea 21 Plus Project in 2021 (S.Y.). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

ABSTRACT Bridging the exponentially growing gap between the numbers of unlabeled and labeled protein sequences, several studies adopted semi-supervised learning for protein sequence modeling. In these studies, models were pre-trained with a substantial amount of unlabeled data, and the representations were transferred to various downstream tasks. Most pre-training methods solely rely on language modeling and often exhibit limited performance. In this paper, we introduce a novel pre-training scheme called **PLUS**, which stands for **P**rotein sequence representations **L**earned **U**sing **S**tructural information. PLUS consists of masked language modeling and a complementary protein-specific pre-training task, namely same-family prediction. PLUS can be used to pre-train various model architectures. In this work, we use PLUS to pre-train a bidirectional recurrent neural network and refer to the resulting model as PLUS-RNN. Our experiment results demonstrate that PLUS-RNN outperforms other models of similar size solely pre-trained with the language modeling in six out of seven widely used protein biology tasks. Furthermore, we present the results from our qualitative interpretation analyses to illustrate the strengths of PLUS-RNN. PLUS provides a novel way to exploit evolutionary relationships among unlabeled proteins and is broadly applicable across a variety of protein biology tasks. We expect that the gap between the numbers of unlabeled and labeled proteins will continue to grow exponentially, and the proposed pre-training method will play a larger role. All the data and codes used in this study are available at https://github.com/mswzeus/PLUS.

INDEX TERMS Protein sequence, protein structure, representation learning, semi-supervised learning.

I. INTRODUCTION

Proteins consisting of linear chains of amino acids are among the most versatile molecules in living organisms. They serve vital functions in biological mechanisms, *e.g.*, transporting other molecules and providing immune protection [1]. The versatility of proteins is generally attributed to their diverse structures. Proteins naturally fold into three-dimensional structures determined by their amino acid sequences. These structures have a direct impact on their functions.

The associate editor coordinating the review of this manuscript and approving it for publication was Nadeem Iqbal.

With the development of next-generation sequencing technologies, protein sequences have become relatively more accessible. However, annotating a sequence with meaningful attributes is still time-consuming and resource-intensive. To bridge the exponentially growing gap between the numbers of unlabeled and labeled protein sequences, various *in silico* approaches have been widely adopted for predicting the characteristics of protein sequences [2].

Sequence alignment is a key technique in computational protein biology. Alignment-based methods are used to compare protein sequences using carefully designed scoring

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

matrices or hidden Markov models (HMMs) [3], [4]. Correct alignments can group similar sequences, provide information on conserved regions, and help investigate uncharacterized proteins. However, its computational complexity increases exponentially with the number of proteins, and it has difficulties in identifying distantly related proteins. Homologous proteins sharing a common evolutionary ancestor can have high sequence-level variations [5]. Therefore, a simple comparison of sequence similarities often fails to capture the global structural and functional similarities of proteins.

Building upon the success of deep learning, several studies proposed deep learning algorithms for computational protein biology. Some of these algorithms only use raw protein sequences, whereas others may use additional features [6]. They have advanced the state-of-the-art (SOTA) for various protein biology tasks. However, development of these algorithms requires highly task-specific processes, *e.g.*, training a randomly initialized model from scratch. It demands careful consideration of the model architectures and hyperparameters tailored for each task. Additional features, such as alignment-based features or known structural traits, may also be required for some tasks [7].

Semi-supervised learning, which leverages both unlabeled and labeled data, has been a long-standing goal of the machine learning community [8]. A semi-supervised learning algorithm pre-trains a universal model with a substantial amount of unlabeled data. Subsequently, it transfers the learned representations and fine-tunes the model with a small amount of labeled data for each downstream task. Now, the natural question is: can protein biology also take advantage of semi-supervised learning? According to the linguistic hypothesis [9], naturally occurring proteins are not purely random. Evolutionary pressure constrains them to a learnable manifold where indispensable structures and functions are maintained. Thus, by observing many unlabeled protein sequences, we can obtain an implicit understanding of the language of proteins. Several studies have recently proposed pre-training methods for protein sequence representations [10]–[17]. They pre-trained models with language modeling (LM) and showed that pre-training helps in downstream protein biology tasks. However, according to the recent benchmark results from tasks assessing protein embeddings (TAPE) [7], the current pre-training methods are often outperformed by task-specific models. This may be because most pre-training methods solely rely on LM to learn from unlabeled protein sequences. Therefore, a complementary protein-specific task for pre-training might be necessary to better capture the information contained within unlabeled protein sequences.

In this paper, we introduce a novel pre-training scheme for protein sequence modeling and name it **PLUS**, which stands for **P**rotein sequence representations **L**earned **U**sing **S**tructural information. PLUS consists of masked language modeling (MLM) and an additional complementary protein-specific pre-training task, same-family prediction (SFP). SFP leverages computationally clustered protein families [18] and helps to better capture the global structural information within unlabeled protein sequences. We use PLUS to pre-train a bidirectional recurrent neural network (BiRNN) and refer to the resulting model as PLUS-RNN. Subsequently, this pre-trained universal model is fine-tuned on various downstream tasks without training randomly initialized task-specific models from scratch. Our experiment results demonstrate that PLUS-RNN outperforms other models of similar size solely pre-trained with the conventional LM in six out of seven widely used protein biology tasks. The seven tasks include three protein-level classification, two protein-level regression, and two amino-acid-level classification. PLUS provides a novel way to exploit evolutionary relationships among unlabeled proteins and is broadly applicable across a variety of protein biology tasks. Finally, we present the results from our qualitative interpretation analyses to illustrate the strengths of PLUS-RNN.

In summary, the contributions of our paper are as follows:

- We introduce PLUS, a novel pre-training scheme for bidirectional protein sequence representations.
- Consisting of MLM and protein-specific SFP pre-training tasks, PLUS can better capture structural information contained within proteins.
- PLUS outperforms other models of similar sizes (solely pre-trained with the conventional LM) in six out of seven widely used protein biology tasks.
- We present qualitative interpretation analyses to better understand the strengths of our PLUS framework.

II. RELATED WORKS

A. PRE-TRAINING NATURAL LANGUAGE REPRESENTATIONS

Pre-training natural language representations has been the basis of natural language processing (NLP) research for a long time. They use language modeling (LM) for pre-training natural language representations. The key idea is that ideal representations must convey syntactic and semantic information, and thus we must be able to use a representation of a token to predict its neighboring tokens. For example, embeddings from language models (ELMo) learned contextualized representations by adopting forward and reverse RNNs [19]. Given a sequence of tokens without additional labels, the forward RNN sequentially processes the sequence left-to-right. It is trained to predict the next token, given its history. The reverse RNN is similar but processes the sequence in reverse, right-to-left. After the pre-training, the hidden states of both RNNs are merged into a single vector representation for each token. Thus, the same token can be transformed into different representations based on its context.

The major limitation of ELMo is that RNNs are trained using unidirectional LM and simply combined afterward. As valuable information often comes from both directions, unidirectional LM is inevitably suboptimal. To address this problem, bidirectional encoder representations from Transformers (BERT) was proposed to pre-train bidirectional natural language representations using the Transformer

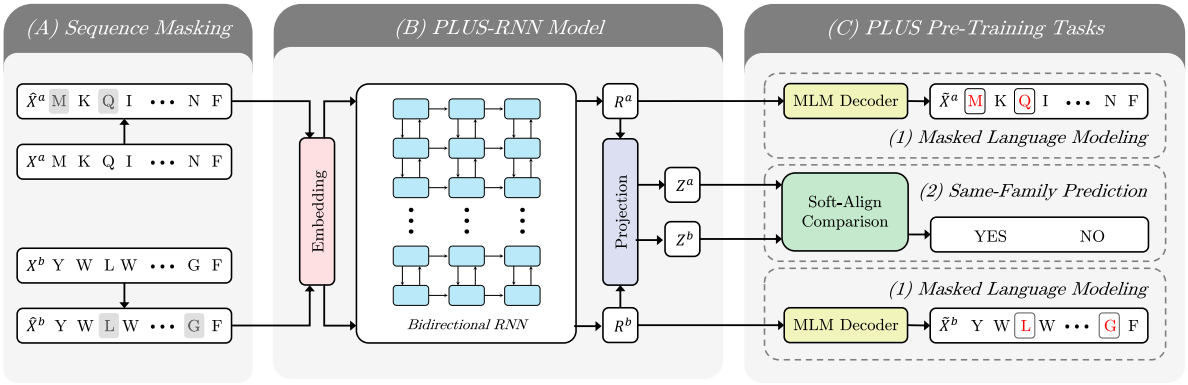

FIGURE 1. Overview of PLUS pre-training scheme (see Section III for details). (A) We randomly mask 15% of amino acids (gray boxes) in each protein sequence X^a and X^b. (B) PLUS-RNN transforms masked protein sequences $\hat{X}^a$ and $\hat{X}^b$ into sequences of bidirectional representations R^a and R^b, respectively. (C) PLUS consists of two pre-training tasks. Masked language modeling trains a model to predict the masked amino acids (colored red within white boxes of $\tilde{X}^a$ and $\tilde{X}^b$) given their contexts $\hat{X}^a$ and $\hat{X}^b$. Same-family prediction (SFP) trains a model to predict whether a pair of proteins belongs to the same protein family where soft-align comparison computes similarity score between projected sequences Z^a and Z^b of R^a and R^b, respectively.

model [20]. Instead of the conventional LM, BERT utilizes an masked language modeling (MLM) pre-training task. It masks some input tokens at random and trains the model to predict them from the context. In addition, BERT includes a complementary NLP-specific pre-training task, next sentence prediction, which enables the learning of sentence relationships by training a model to predict whether a given pair of sentences is consecutive.

B. PRE-TRAINING PROTEIN SEQUENCE REPRESENTATIONS

NLP-based methods are historically adapted to learn protein sequence representations [10], [21]. The previous methods most closely related to our paper are P-ELMo [12] and UniRep [11], which learn contextualized protein representations. P-ELMo is based on a two-phase algorithm. First, it trained forward and reverse RNNs using LM with an unlabeled dataset. then, it adopted another bidirectional recurrent neural network (BiRNN) and further trained the model with an additional small labeled dataset. Note that the latter supervised training deviates from the goal of pre-training, namely, utilizing low human effort and large unlabeled datasets. UniRep used a unidirectional RNN with multiplicative long short-term memory hidden units [22]. Similarly, UniRep trained its model using conventional LM.

Most methods have some common limitations and still often lag behind task-specific models [7]. First, some of them learn unidirectional representations from unlabeled datasets. Unidirectional representations are sub-optimal for numerous protein biology tasks, where it is crucial to assimilate global information from both directions. Note that we do not consider combination of two unidirectional representations as bidirectional representations since they were simply combined after the unidirectional pre-training. Second, most pre-training methods solely rely on LM to learn from unlabeled protein sequences. Although LM is a simple and effective task, a complementary pre-training task tailored for each data modality has been often the key to further improve the quality of representations in other domains. For instance, in NLP, BERT adopted the next sentence prediction task. In another example, ALBERT devised a complementary sentence order prediction task to model the inter-sentence coherence and yielded consistent performance improvements for downstream tasks [23]. Similarly, a complementary protein-specific task for pre-training might be necessary to better capture the information contained within unlabeled proteins.

III. METHODS

We introduce PLUS (Figure 1), a novel pre-training scheme for protein sequence modeling. Consisting of MLM and complementary protein-specific SFP pre-training tasks, PLUS can help a model to learn structurally contextualized bidirectional representations. In the following, we will explain the details of the pre-training procedures, fine-tuning procedures, and the model architecture.

A. PRE-TRAINING PROCEDURE

PLUS can be used to pre-train various model architectures that transform a protein sequence $X = [x_1, \cdots, x_n]$, which has a variable-length n, into a sequence of bidirectional representations $Z = [z_1, \cdots, z_n]$ with the same length. In this work, we use PLUS to pre-train a BiRNN and refer to the resulting model as PLUS-RNN. The complete pre-training loss is defined as:

$$\mathcal{L}_{\text{PT}} = \lambda_{\text{PT}}\mathcal{L}_{\text{MLM}} + (1 - \lambda_{\text{PT}})\mathcal{L}_{\text{SFP}}$$

where $\mathcal{L}_{\text{MLM}}$ and $\mathcal{L}_{\text{SFP}}$ are the MLM and SFP losses, respectively. We use λ_{PT} to control their relative importance (Appendix A).

1) TASK #1: MASKED LANGUAGE MODELING (MLM)

Given a protein sequence X, we randomly select 15% of the amino acids. Then, for each selected amino acid x_i, we randomly perform one of the following procedures. For 80% of the time, we replace x_i with the token denoting an unspecified amino acid. For 10% of the time, we randomly replace x_i with one of the 20 amino acids. Finally, for the remaining 10%, we keep x_i intact. This is to bias the learning toward the true amino acids. For the probabilities of masking actions, we follow those used in BERT [20].

PLUS-RNN transforms a masked protein sequence $\hat{X}$ into a sequence of representations. Then, we use an MLM decoder to compute log probabilities from the representations for $\widetilde{X}$ over 20 amino acid types. The MLM task trains the model to maximize the probabilities corresponding to the masked ones. As the model is designed to accurately predict randomly masked amino acids given their contexts, the learned representations must convey syntactic and semantic information within proteins.

2) TASK #2: SAME-FAMILY PREDICTION (SFP)

Considering that additional pre-training tasks has been often the key for improving the quality of representations in other domains [20], [23], we devise a complementary protein-specific pre-training task. The SFP task trains a model to predict whether a given protein pair belongs to the same protein family. The protein family labels provide weak structural information and help the model learn structurally contextualized representations. Note that PLUS is still a semi-supervised learning method; it is supervised by computationally clustered weak labels rather than human-annotated labels.

We randomly sample two protein sequences X^a and X^b, from the training dataset. In 50% of the cases, two sequences are sampled from the same protein family. For the other 50%, they are randomly sampled from different families. PLUS-RNN transforms the protein pair into sequences of representations $Z^a = [z_1^a, \cdots, z_{n_1}^a]$ and $Z^b = [z_1^b, \cdots, z_{n_2}^b]$. Then, we use a soft-align comparison [12] to compute their similarity score, $\hat{c}$, as a negative weighted sum of l_1-distances between every z_i^a and z_j^b pair:

$$\hat{c} = -\frac{1}{C}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\omega_{ij}\left\|z_i^a - z_j^b\right\|_1, \quad C = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\omega_{ij},$$

where weight ω_{ij} of each l_1-distance is computed as

$$\omega_{ij} = 1 - (1-\alpha_{ij})(1-\beta_{ij}),$$
$$\alpha_{ij} = \frac{\exp(-\left\|z_i^a - z_j^b\right\|_1)}{\sum_{k=1}^{n_2}\exp(-\left\|z_i^a - z_k^b\right\|_1)},$$
$$\beta_{ij} = \frac{\exp(-\left\|z_i^a - z_j^b\right\|_1)}{\sum_{k=1}^{n_1}\exp(-\left\|z_k^a - z_j^b\right\|_1)}.$$

Intuitively, we can understand the soft-align comparison as computing an *expected alignment score*, where they are summed over all the possible alignments. We suppose that the smaller the distance between representations, the more likely it is that the pair of amino acids is aligned. Then, we can consider α_{ij} as the probability that z_i^a is aligned to z_j^b, considering all the amino acids from Z^b (and vice versa for β_{ij}). As a result, $\hat{c}$ is the expected alignment score over all possible alignments with probabilities ω_{ij}. Note that the negative signs are applied for converting distances into scores. Therefore, a higher value of $\hat{c}$ indicates that the pair of protein sequences is structurally more similar.

Given the similarity score, the SFP output layer computes the probability that the pair belongs to the same protein family. The SFP task trains PLUS-RNN to minimize the cross-entropy loss between the true label and the predicted probability. As the model is designed to produce higher similarity scores for proteins from the same families, learned representations must convey global structural information.

B. FINE-TUNING PROCEDURE

The fine-tuning procedure of PLUS-RNN follows the conventional usage of BiRNN-based prediction models. For each downstream task, we add one hidden layer and one output layer on top of the pre-trained model. Then, all the parameters are fine-tuned using task-specific datasets. The complete fine-tuning loss is defined as:

$$\mathcal{L}_{\text{FT}} = \lambda_{\text{FT}}\mathcal{L}_{\text{MLM}} + (1-\lambda_{\text{FT}})\mathcal{L}_{\text{TASK}}$$

where $\mathcal{L}_{\text{TASK}}$ is the task-specific loss. $\mathcal{L}_{\text{MLM}}$ is the regularization loss. We use λ_{FT} to control their relative importance (Appendix A).

The model's architectural modifications for the three types of downstream tasks are as follows. For tasks involving a protein pair, we use the same computations used in the SFP pre-training task. Specifically, we replace only the SFP output layer with a new output layer. For single protein-level tasks, we adopt an additional attention layer to aggregate variable-length representations into a single vector [24]. Then, the aggregated vector is fed into the hidden and output layers. For amino-acid-level tasks, representations of each amino acid are fed into the hidden and output layers.

C. MODEL ARCHITECTURE

PLUS can be used to pre-train any model architectures that transform a protein sequence into a sequence of bidirectional representations. In this work, we use PLUS-RNN because of its superior sequential modeling capability and lower computational complexity. Refer to Appendix B for more detailed explanations of the advantages of PLUS-RNN over an alternative Transformer-based model, called PLUS-TFM.

In this section, we explain the architecture of PLUS-RNN. First, an input embedding layer, EM, embeds each amino acid x_i into a d_e-dimensional dense vector e_i:

$$E = [e_1, \cdots, e_n], \quad e_i = \text{EM}(x_i).$$

Then, a BiRNN of L-layers computes representations as a function of the entire sequence. We use long short-term

TABLE 1. Summarized results on protein biology benchmark tasks.

	Protein-level Classification			Protein-level Regression		Amino-acid-level Classification	
Method	Homology (acc)	Solubility (acc)	Localization (acc)	Stability (ρ)	Fluorescence (ρ)	SecStr (acc)	Transmembrane (acc)
PLUS-TFM	0.96	**0.72**	0.69	0.76	0.63	0.59	0.82
PLUS-RNN$_{\text{BASE}}$	0.96	0.70	0.69	**0.77**	0.67	0.61	**0.89**
PLUS-RNN$_{\text{LARGE}}$	**0.97**	0.71	**0.70**	**0.77**	**0.68**	**0.62**	**0.89**
LM Pre-trained	0.95	0.64	0.54	0.73	**0.68**	0.61	0.78
Task-specific SOTA	0.93	0.77	0.78	0.73	0.67	0.72	0.80

For each task, the best pre-trained model is in **bold**. It is **bold and underlined** if it is the best when including the task-specific SOTA.

memory as the basic unit of the BiRNN [25]. In each layer, the BiRNN computes d_h-dimensional forward and backward hidden states ($\overrightarrow{h_i^l}$ and $\overleftarrow{h_i^l}$) and combines them into a hidden state h_i^l using a non-linear transformation:

$$\begin{aligned}\overrightarrow{h_i^l} &= \sigma(\overrightarrow{W_x^l} h_i^{l-1} + \overrightarrow{W_h^l} h_{i-1}^l + \overrightarrow{b^l}),\\ \overleftarrow{h_i^l} &= \sigma(\overleftarrow{W_x^l} h_i^{l-1} + \overleftarrow{W_h^l} h_{i+1}^l + \overleftarrow{b^l}),\\ h_i^l &= \sigma(W_h^l[\overrightarrow{h_i^l}; \overleftarrow{h_i^l}] + b^l) \quad for\ l = 1, \cdots, L,\end{aligned}$$

where $h_i^0 = e_i$; W and b are the weight and bias vectors, respectively. We use the final hidden states h_i^L as representations r_i of each amino acid:

$$R = [r_1, \cdots, r_n], \quad r_i = h_i^L.$$

We adopt an additional projection layer to obtain smaller d_z-dimensional representations z_i of each amino acid with a linear transformation:

$$Z = [z_1, \cdots, z_n], \quad z_i = \text{Proj}(r_i).$$

During pre-training, to reduce computational complexity, we use R and Z for the MLM and SFP tasks, respectively. During fine-tuning, we can use either R or Z, considering the performance on development sets or based on the computational constraints.

We use two models with the fixed d_e of 21 and d_z of 100:

- PLUS-RNN$_{BASE}$: $L = 3$, $d_h = 512$, 15M parameters
- PLUS-RNN$_{LARGE}$: $L = 3$, $d_h = 1024$, 59M parameters

The hyperparameters (*i.e.*, L and d_h) of PLUS-RNN$_{BASE}$ are chosen to match the BiRNN model architecture used in P-ELMo [12]. However, as P-ELMo uses additional RNNs, PLUS-RNN$_{BASE}$ has less than half the number of parameters that P-ELMo has (32M).

IV. EXPERIMENTS

A. PRE-TRAINING DATASET

We used Pfam (release 27.0) as the pre-training dataset [18]. After pre-processing (Appendix A), it contained 14,670,860 sequences from 3,150 families. The Pfam dataset provides protein family labels which were computationally pre-constructed by comparing sequence similarity using multiple sequence alignments and HMMs. Owing to the loose connection between sequence and structure similarities, the family labels only provide weak structural information [26]. Note that we did not use any human-annotated labels. Therefore, pre-training does not result in biased evaluations in fine-tuning tasks. The pre-training results are provided in the Appendix C.

B. FINE-TUNING TASKS

We evaluated PLUS-RNN on seven protein biology tasks. The datasets were curated and pre-processed by the cited studies. In the main manuscript, we provide concise task definitions and evaluation metrics. Please refer to Appendix D for more details.

Homology is a protein-level classification task [27]. The goal is to classify the structural similarity level of a protein pair into *family*, *superfamily*, *fold*, *class*, or *none*. We report the accuracy of the predicted similarity level and the Spearman correlation, ρ, between the predicted similarity scores and the true similarity levels. Furthermore, we provide the average precision (AP) from prediction scores at each similarity level.

Solubility is a protein-level classification task [28]. The goal is to predict whether a protein is *soluble* or *insoluble*. We report the accuracy of this task.

Localization is a protein-level classification task [29]. The goal is to classify a protein into one of 10 subcellular locations. We report the accuracy of this task.

Stability is a protein-level regression task [30]. The goal is to predict a real-valued proxy for intrinsic stability. This task is from TAPE [7], and we report the Spearman correlation, ρ.

Fluorescence is a protein-level regression task [31]. The goal is to predict the real-valued fluorescence intensities. This task is from TAPE, and we report the Spearman correlation, ρ.

Secondary structure (SecStr) is an amino-acid-level classification task [32]. The goal is to classify each amino acid into eight or three classes, that describe its local structure. This task is from TAPE. We report both the three-way and eight-way classification accuracies (Q8/Q3) of this task.

Transmembrane is an amino-acid-level classification task [33]. The goal is to detect amino acid segments that cross the cell membrane. We report the accuracy of this task.

C. BASELINES

We provided several baselines for comparative evaluations. Note that since up-scaling of models and datasets often provide performance improvements, we only considered those with a similar scale of model sizes and pre-training datasets to focus on evaluating the pre-training schemes.

First, in all the tasks, we used two baselines: P-ELMo and PLUS-TFM. The former has a model architecture similar to PLUS-RNN$_{BASE}$; thus, it can show the effectiveness of the pre-training scheme. The latter is pre-trained with PLUS, so it can show the effectiveness of the BiRNN compared to the Transformer architecture.

Second, for the tasks from TAPE, we provide their reported baselines: P-ELMo, UniRep, TAPE-TFM, TAPE-RNN, and TAPE-ResNet. Note that these comparisons are in their favor, as they used a larger pre-training dataset (32M proteins from Pfam release 32.0). The TAPE baselines can demonstrate that PLUS-RNN outperforms models of similar size solely pre-trained with the LM.

Finally, we benchmarked PLUS-RNN against task-specific SOTA models trained from scratch. If no deep learning-based baseline exists for a given task, we provided RNN$_{BASE}$ and RNN$_{LARGE}$ models without pre-training. The comparison with those exploit additional features can help us identify the tasks for which the proposed pre-training scheme is most effective and help us understand its current limitations.

D. SUMMARY OF FINE-TUNING RESULTS

Table 1 presents the summarized results for the benchmark tasks. Specifically, we show the best results from two categories: LM pre-trained models and task-specific SOTA models. Refer to Appendix D for detailed fine-tuning results.

The PLUS-RNN$_{LARGE}$ model outperformed models of similar size solely pre-trained with the conventional LM in six out of seven tasks. Considering that some pre-trained models exhibited higher LM capabilities (Appendix C), it can be speculated that the protein-specific SFP pre-training task contributed to the improvement. In the ablation studies, we further explained the relative importance of each aspect of PLUS-RNN (Appendix E). Although PLUS-TFM had almost twice as many parameters as PLUS-RNN$_{LARGE}$ (110M vs. 59M), it exhibited inferior performance in most tasks. We infer that this is because it disregarded the *locality bias* (Appendix B).

We compared PLUS-RNN$_{LARGE}$ with task-specific SOTA models. Although the former performed better in some tasks, it still lagged behind on the others. The results indicated that tailored models with additional features provide powerful advantages that could not be learned through pre-training. A classic example is the use of position-specific scoring matrices generated from multiple sequence alignments. We conjectured that simultaneous observation of multiple proteins could facilitate evolutionary information. In contrast, current pre-training methods use millions of proteins; however, they still consider each one individually. The relatively small performance improvement from PLUS could also be explained by the fact that the SFP task only utilizes pairwise information. We expect that investigating multiple proteins during pre-training might be the key to a superior performance over the task-specific SOTA models.

TABLE 2. Detailed Homology prediction results.

	Overall		Per-level AP			
Method	acc	ρ	Class	Fold	Superfamily	Family
PLUS-TFM	0.96	**0.70**	0.94	0.91	0.95	**0.67**
PLUS-RNN$_{\text{BASE}}$	0.96	0.69	0.94	0.90	0.94	0.66
PLUS-RNN$_{\text{LARGE}}$	**0.97**	**0.70**	**0.95**	**0.92**	**0.96**	0.66
P-ELMo	0.95	0.69	0.90	0.88	0.94	0.65
P-ELMo†	0.95	0.69	0.91	0.90	0.95	0.65
NW-align†	0.78	0.22	0.31	0.41	0.58	0.53
HHalign†	0.79	0.23	0.40	0.62	0.86	0.52
TMalign†	0.81	0.37	0.55	0.85	0.83	0.57
RNN$_{\text{BASE}}$	0.93	0.66	0.86	0.80	0.89	0.62
RNN$_{\text{LARGE}}$	0.83	0.52	0.66	0.46	0.52	0.39

† Excerpted from P-ELMo.

TABLE 3. Detailed SecStr prediction results.

	CB513		CASP12		TS115	
Method	Q8	Q3	Q8	Q3	Q8	Q3
PLUS-TFM	0.59	0.73	0.57	0.71	0.65	0.77
PLUS-RNN$_{\text{BASE}}$	0.61	0.75	0.60	0.72	0.66	0.78
PLUS-RNN$_{\text{LARGE}}$	**0.62**	**0.77**	**0.60**	**0.73**	**0.68**	**0.79**
P-ELMo*	0.61	**0.77**	0.54	0.68	0.63	0.76
P-ELMo†	0.58	0.73	0.57	0.70	0.65	0.76
UniRep†	0.57	0.73	0.59	0.72	0.63	0.77
TAPE-TFM†	0.59	0.73	0.59	0.71	0.64	0.77
TAPE-RNN†	0.59	0.75	0.57	0.70	0.66	0.78
TAPE-ResNet†	0.58	0.75	0.58	0.72	0.64	0.78
NetSurfP-2.0‡	0.72	0.85	0.70	0.82	0.75	0.86

† Excerpted from TAPE. ‡ Excerpted from [32].

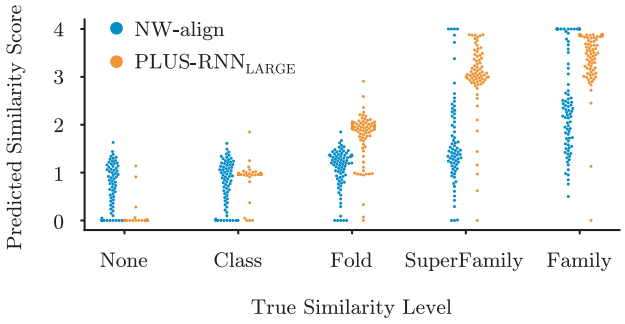

FIGURE 2. Plot of predicted similarity scores and true similarity levels. Results from NW-align (left) and PLUS-RNN$_{LARGE}$ (right) are presented in each similarity level. Figure best viewed in color.

E. DETAILED HOMOLOGY AND SecStr RESULTS

We present detailed evaluation results for the Homology and SecStr tasks. We chose these two tasks because they are representative protein biology tasks relevant to global and local structures, respectively. Improved results on the former

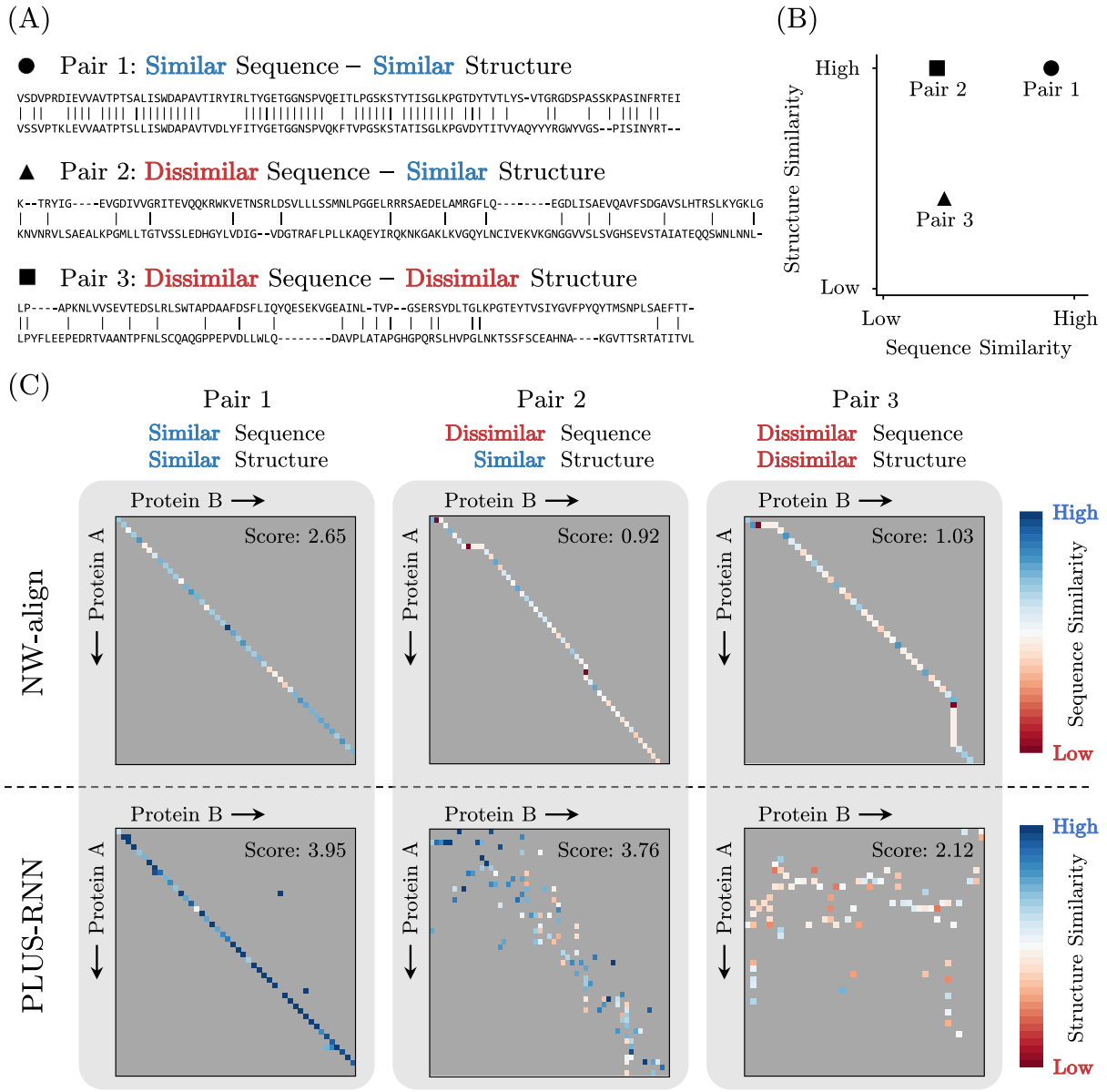

FIGURE 3. Homology interpretation (A) We investigate three types of protein pairs: a similar sequence - similar structure pair, a dissimilar sequence - similar structure pair, and a dissimilar sequence - dissimilar structure pair. (B) The sequence and structure similarities of each pair are defined by NW-align scores and Homology dataset labels, respectively. (C) Heatmaps of NW-align of raw amino acids and the soft-alignment of PLUS-RNN representations for the three pairs. Owing to space limitations, we only show the top left quadrant of the heatmaps. Figure best viewed in color.

can lead to the discovery of new enzymes and antibiotic-resistant genes [34]. The latter is important for understanding the function of proteins in cases where evolutionary structural information is not available [32].

The detailed Homology prediction results are listed in Table 2. The results show that PLUS-RNN$_{LARGE}$ outperformed both P-ELMo and task-specific models. In contrast to RNN$_{LARGE}$, which exhibited overfitting owing to the limited labeled training data, PLUS pre-training enabled us to take advantage of the large model architecture. The correlation differences among PLUS-RNN$_{LARGE}$ (0.697), PLUS-RNN$_{BASE}$ (0.693), and P-ELMo (0.685) were small; however, they were statistically significant with p-values lower than 10^{-15} [35]. The per-level AP results helped us further examine the level of structural information captured by the pre-training. The largest performance improvement of PLUS-RNN comes at the higher *class* level rather than the lower *family* level. This indicates that even though Pfam family labels tend to be structurally correlated with the Homology task *family* levels [5], they are not the decisive factors for performance improvement. Instead, PLUS pre-training incorporates weak structural information

and facilitates inferring higher-level global structure similarities.

The detailed SecStr prediction results are listed in Table 3. CB513, CASP12, and TS115 denote the SecStr test datasets. The results show that PLUS-RNN$_{LARGE}$ outperformed all the other models of similar size pre-trained solely with LM. This demonstrates that the SFP task complements the LM task during pre-training and helps in learning improved structurally contextualized representations. However, PLUS-RNN$_{LARGE}$ still lagged behind task-specific SOTA models that employ alignment-based features. We infer that this limitation might be attributable to the following two factors. First, as previously stated, PLUS only utilizes pairwise information, rather than simultaneously examining multiple proteins during pre-training. Second, the SFP task requires an understanding of the global structures, and thus the local structures are relatively negligible. Therefore, we believe that devising an additional pre-training task relevant to local structural information would improve the performance on the SecStr task.

F. QUALITATIVE ANALYSES

To better understand the strengths of PLUS pre-training, we provide its qualitative analyses. We examined the Homology task to interpret how the learned protein representations help infer the global structural similarities of proteins

To compare two proteins, PLUS-RNN used soft-align to compute their similarity score, $\hat{c}$. Even though there was one more computation by the output layer for the Homology prediction output, we could use the similarity scores to interpret PLUS-RNN. Note that using the penultimate layer for model interpretation is a widely adopted approach in the machine learning community [36].

Figure 2 shows a scatter plot of the predicted similarity scores and true similarity levels. For comparison, we also show the NW-align results based on the BLOSUM62 scoring matrix [3]. The plot shows that NW-align often produces low similarity scores for protein pairs from the same *family*. This is because of high sequence-level variations, which result in dissimilar sequences having similar structures. In contrast, PLUS-RNN$_{LARGE}$ produces high similarity scores for most protein pairs from the same *family*.

Furthermore, we examined three types of protein pairs: (1) a similar sequence-similar structure pair, (2) a dissimilar sequence-similar structure pair, and (3) a dissimilar sequence-dissimilar structure pair (Figure 3(A) and (B)). Note that similar sequence-dissimilar structure pairs did not exist in the Homology datasets. The sequence and structure similarities were defined by NW-align scores and Homology dataset labels, respectively. The pairs with similar structures were chosen from the same *family*, and those with dissimilar structures were chosen from the same *fold*. Figure 3(C) shows the heatmaps of the NW-align of raw amino acids and soft-alignment of PLUS-RNN representations (ω_{ij}) for the three pairs. Owing to space limitations, we only show the top left quadrant of the heatmaps. Each cell in the heatmap indicates the corresponding amino acid pairs from proteins A and B. Blue denotes high sequence similarity in NW-align and high structure similarity in PLUS-RNN.

First, we compared the pairs having similar structures (the first and second columns in Figure 3(C)). The heatmaps show that NW-align successfully aligned the similar-sequence pair, resulting in a score of 2.65. However, it failed for the dissimilar-sequence pair, with a score of 0.92. This supports the observation that comparing raw sequence similarities cannot identify the correct structural similarities. In contrast, the soft-alignment of PLUS-RNN representations was successful for both similar and dissimilar sequences, with scores of 3.95 and 3.76, respectively. Next, we compared the second and third pairs. Although only the second pair had similar structures, NW-align failed for both and even yielded a higher score of 1.03 for the third pair. In contrast, regardless of the sequence similarities, the soft-alignment of PLUS-RNN representations correctly decreased only for the third pair, with dissimilar structures having a score of 2.12. Therefore, the interpretation results confirmed that the learned representations from PLUS-RNN are structurally contextualized and perform better in inferring global structure similarities.

V. CONCLUSION

In this work, we presented PLUS, a novel pre-training scheme for bidirectional protein sequence representations. Consisting of the MLM and protein-specific SFP pre-training tasks, PLUS outperformed the conventional LM pre-training methods by capturing structural information contained within the proteins. PLUS can be used to pre-train various model architectures. In this work, we used PLUS-RNN because of its superior sequential modeling capability and lower computational complexity. PLUS-RNN outperformed models of similar size solely pre-trained with the conventional LM in six out of seven protein biology tasks. To better understand its strengths, we also provided the results from our qualitative interpretation analyses.

We expect that the gap between the numbers of unlabeled and labeled proteins will continue to grow exponentially, and pre-training methods will play a larger role. We plan to extend this work in several directions. First, considering that PLUS-RNN is powerful for inferring global structural information, we are interested in a more refined prediction of protein structures [37]. Second, although pre-training helps, our scheme still lags behind task-specific models in some tasks. We think that this limitation comes from weaknesses in learning evolutionary information. We believe that there is still considerable room for improvement. Investigation of multiple proteins during pre-training, as in the alignment, could be the key [38].

APPENDIX A
DETAILS ON TRAINING PROCEDURES

All models were implemented in PyTorch [39]. We used the NAVER smart machine learning environment for

pre-training [40]. In the following subsections, we explain the details of the pre-training and fine-tuning procedures.

A. PRE-TRAINING PROCEDURE

We used Pfam (release 27.0) as the pre-training dataset [18]. Moreover, we divided the training and test sets in a random 80%/20% split and filtered out sequences shorter than 20 amino acids. Additionally, for the training set, we removed families containing fewer than 1,000 proteins. This resulted in 14,670,860 sequences from 3,150 families being utilized for the pre-training of PLUS. For the test dataset, we sampled 100,000 pairs from the test split.

We pre-trained PLUS-RNN with a batch size of 64 sequences for 900,000 steps, which is approximately four epochs over the training dataset. We used the Adam optimizer [41] with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and without weight decay and dropout. Default λ_{PT} was 0.7.

For pre-training PLUS-TFM, we used different filtering conditions due its computational complexity. The minimum and maximum lengths of a protein were set to 20 and 256, respectively. The minimum number of proteins for a family was set to 1000. This resulted in 11,956,227 sequences from 2,657 families. We pre-trained PLUS-TFM with a batch size of 128 for 930,000 steps, which is approximately 10 epochs over the training dataset. We used the Adam optimizer with a learning rate of 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L_2 weight decay of 0.01, a linearly decaying learning rate with warmup over the first 10% steps, and a dropout probability of 0.1. Default λ_{PT} was 0.7.

B. FINE-TUNING PROCEDURE

When fine-tuning PLUS-RNN, most model hyperparameters were the same as those during the pre-training. The commonly used hyperparameters were as follows; we fine-tuned the PLUS-RNN with a batch size of 32 for 20 epochs. We used the Adam optimizer with a smaller learning rate of 0.0005, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and without weight decay. For the other hyperparameters, we chose the configurations that performed best on the development sets for each task. The possible configurations were as follows:

- Number of units in the added output layer: 128, 512
- Usage of the projection layer: True, False
- λ_{FT}: 0, 0.3, 0.5

For fine-tuning PLUS-TFM, we used the following hyperparameters: batch size of 32 for 20 epochs, the Adam optimizer with a smaller learning rate of 0.00005, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and without weight decay. Default λ_{FT} was 0.3.

For fine-tuning both PLUS-RNN and PLUS-TFM for the Homology task, we additionally followed the procedures of [12]. In each epoch, we sampled 100,000 protein pairs using the smoothing rule: the probability of sampling a pair with similarity level t is proportional to $N_t^{0.5}$, where N_t is the number of protein pairs with similarity level t.

APPENDIX B
DETAILS ON PLUS-TFM

A. TRANSFORMER ARCHITECTURE

The key element of the Transformer is a self-attention layer composed of multiple attention heads [42]. Given an input sequence, $X = [x_1, \cdots, x_n]$, an attention head computes the output sequence, $Z = [z_1, \cdots, z_n]$. Each token is a weighted sum of values computed by a weight matrix W^V:

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V).$$

Each attention coefficient, α_{ij}, is the output of a softmax function applied to the dot products of the query with all keys, which are computed using W^Q and W^K:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})}, \quad e_{ij} = \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d_z}},$$

where d_z is the output token dimension. The self-attention layer directly performs $O(1)$ computations for all the pairwise tokens, whereas a recurrent layer requires $O(n)$ sequential computations for the farthest pair. This allows easier traversal for forward and backward signals and thus better captures long-range dependencies.

The model architecture of PLUS-TFM is analogous to the BERT_{BASE} model, consisting of 110M parameters. Because of its significant computational burden, which scale quadratically with the input length, we pre-trained PLUS-TFM using only protein pairs shorter than 512 amino acids, following the procedures used in BERT.

B. ADVANTAGES OF PLUS-RNN OVER PLUS-TFM

PLUS can be used to pre-train various model architectures including BiRNN and the Transformer. The resulting models are referred to as PLUS-RNN and PLUS-TFM, respectively. In this work, we mainly used PLUS-RNN, because of its two advantages over PLUS-TFM. First, it is more effective for learning the sequential nature of proteins. The self-attention layer of the Transformer performs dot products between all pairwise tokens regardless of their positions within the sequence. In other words, it provides an equal opportunity for local and long-range contexts to determine the representations. Although this facilitates the learning of long-range dependencies, the downside is that it completely ignores the *locality bias* within a sequence. This is particularly problematic for protein biology, where local amino acid motifs often have significant structural and functional implications [43]. In contrast, RNN sequentially processes a sequence, and local contexts are naturally more emphasized.

Second, PLUS-RNN provides lower computational complexity. Although the model hyperparameters have an effect, the Transformer-based models generally demand a larger number of parameters than RNNs [42]. Furthermore, the computations between all pairwise tokens in the self-attention layer impose a considerable computational burden, which scales quadratically with the input sequence

length. Considering that pre-training typical Transformer-based models handling 512 tokens already requires tremendous resources [20], it is computationally difficult to use Transformers to manage longer protein sequences, even up to a few thousand amino acids.

APPENDIX C
PRE-TRAINING RESULTS

Figure 4 shows the pre-training cure where training and validation losses are plotted concerning the parameter update steps. While the pre-training had not converged at 900,000 steps, we used the early stopping technique due to the time limitations. It required approximately three weeks to pre-train PLUS-RNN$_{LARGE}$ for 900,000 steps.

Table 4 lists the test accuracies for the MLM and SFP pre-training tasks. Only the models pre-trained with PLUS were evaluated for the SFP task. Note that our experiments and TAPE used different test datasets; care should be taken in comparing them. Nonetheless, we can still indirectly compare them, considering the following. First, both the test datasets comprised randomly sampled proteins from different versions of the Pfam dataset (27.0 for PLUS and 32.0 for TAPE). Second, P-ELMo was evaluated in both datasets and showed similar LM accuracies. This indicates that the difference between the two datasets is negligible.

We can see that some models have lower LM accuracies than others. However, the lower LM capability does not precisely correspond to the performance in fine-tuning tasks. This discrepancy has been previously observed in TAPE, and it can also be observed in the following sections. In terms of SFP, all the models pre-trained with PLUS exhibited high accuracies. As the Pfam families were constructed based only on sequence similarities, a pair of analogous sequences would probably be from the same family. Despite its simplicity, we empirically demonstrated that the SFP complements the MLM by encouraging the models to compare protein representations during pre-training.

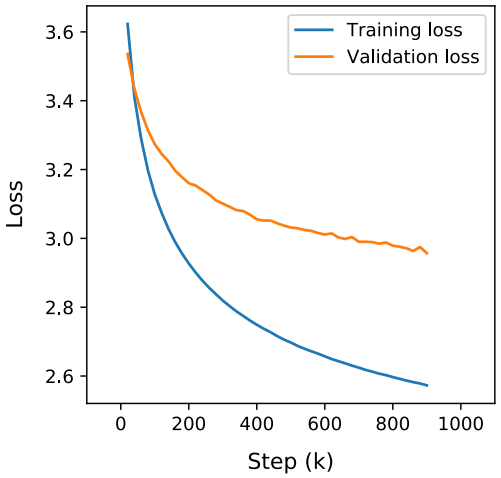

FIGURE 4. Pre-training curve.

TABLE 4. Results on pre-training tasks.

Method	(M)LM (acc)	SFP (acc)
PLUS-TFM	0.37	**0.98**
PLUS-RNN$_{\text{BASE}}$	0.33	0.96
PLUS-RNN$_{\text{LARGE}}$	0.37	0.97
P-ELMo*	0.29	-
P-ELMo†	0.28	-
UniRep†	0.32	-
TAPE-TFM†	**0.45**	-
TAPE-RNN†	0.40	-
TAPE-ResNet†	0.41	-

* Our experiments (Pfam 27.0). † Excerpted from TAPE (Pfam 32.0).

APPENDIX D
FINE-TUNING RESULTS

A. HOMOLOGY

For the Homology task, we used the SCOPe datasets [27], which were pre-processed and provided by [12]. The dataset was filtered with a maximum sequence identity of 95% and split into 20,168 training, 2,240 development, and 5,602 test sequences. For the development and test datasets, we used the sampled 100,000 pairs from each dataset.

The detailed Homology prediction results are listed in Table 2. We report the excerpted baseline results from [12]. NW-align computed the similarity between proteins based on sequence alignments with the BLOSUM62 substitution matrix, with a gap open penalty of -11 and a gap extension penalty of -1 [3]. HHalign conducted multiple sequence alignments and searched similar sequences with HHbits [4], [44]. TMalign performed structure alignments and scored a pair as the average of target-to-query and query-to-target scores [45]. Note that we additionally normalized the scores from NW-align and HHalign with the sum of the lengths of protein pairs. This results in minor correlation improvements compared with the results reported by [12].

B. SOLUBILITY

For the Solubility task, we used the pepeDB datasets [46], [47], which were pre-processed and provided by [28]. The dataset was split into 62,478 training, 6,942 development, and 2,001 test sequences. The training dataset was filtered with a maximum sequence identity of 90% to remove data redundancy. Furthermore, any sequences with more than 30% sequence identity to those from the test dataset were removed to avoid any bias from homologous sequences.

The detailed Solubility prediction results are presented in Table 5. We report the excerpted top-5 baseline results from [28]. PaRSnIP [48], the second-best task-specific baseline model, used a gradient boosting classifier with over 8,000 1, 2, 3-mer amino acid frequency features, sequence-based features (*e.g.*, length, molecular weight, and absolute charge), and structural features (*e.g.*, secondary structures, a fraction of exposed residues, and hydrophobicity). DeepSol [28], the best task-specific baseline model, used a convolutional neural network consisting of convolution and max-pooling

TABLE 5. Detailed solubility prediction results.

Method	acc
PLUS-TFM	**0.72**
PLUS-RNN$_{BASE}$	0.70
PLUS-RNN$_{LARGE}$	0.71
P-ELMo	0.64
SOLPro†	0.60
SCM†	0.60
PROSO2†	0.64
PaRSnIP†	0.74
DeepSol†	0.77

† Excerpted from [28]

TABLE 6. Detailed localization prediction results.

Method	acc
PLUS-TFM	0.69
PLUS-RNN$_{BASE}$	0.69
PLUS-RNN$_{LARGE}$	**0.70**
P-ELMo	0.54
SherLoc2†	0.58
LocTree2†	0.61
YLoc†	0.61
iLoc-Euk†	0.68
DeepLoc†	0.78

† Excerpted from [29].

TABLE 7. Detailed stability prediction results.

Method	ρ
PLUS-TFM	0.76
PLUS-RNN$_{BASE}$	**0.77**
PLUS-RNN$_{LARGE}$	**0.77**
P-ELMo†	0.64
UniRep†	0.73
TAPE-TFM†	0.73
TAPE-RNN†	0.69
TAPE-ResNet†	0.73
RNN$_{BASE}$	0.72
RNN$_{LARGE}$	0.73

† Excerpted from TAPE [7].

modules to learn sequence representations. It additionally adopted the sequence-based and structural features used in PaRSnIP to enhance the performance.

C. LOCALIZATION

For the Localization task, we used the UniProt datasets [49], which were pre-processed and provided by [29]. The proteins were clustered with 30% sequence identity. Then, each cluster of homologous proteins was split into 9,977 training, 1,108 development, and 2,773 test sequences. The subcellular locations are as follows: nucleus, cytoplasm, extracellular, mitochondrion, cell membrane, endoplasmic reticulum, plastid, Golgi apparatus, lysosome, and peroxisome.

TABLE 8. Detailed fluorescence prediction results.

Method	ρ
PLUS-TFM	0.63
PLUS-RNN$_{BASE}$	0.67
PLUS-RNN$_{LARGE}$	**0.68**
P-ELMo†	0.33
UniRep†	0.67
TAPE-TFM†	**0.68**
TAPE-RNN†	0.67
TAPE-ResNet†	0.21
RNN$_{BASE}$	0.58
RNN$_{LARGE}$	0.67

† Excerpted from TAPE [7].

The detailed Localization prediction results are listed in Table 6. We report the excerpted top-5 baseline results from [29]. iLoc-Euk [50], the second-best task-specific model, used a multi-label K-nearest neighbor classifier with pseudo-amino acid frequency features. Because iLoc-Euk predicted 22 locations, these were mapped onto our 10 locations. DeepLoc [29], the best task-specific model, used a convolutional neural network to learn motif information and a recurrent neural network to learn sequential dependencies of the motifs. It also adopted sequence-based evolutionary features through the combination of BLOSUM62 encoding [3] and homology protein profiles from the Swiss-Prot database [51].

D. STABILITY

For the Stability task, we utilized the datasets from [30], which were pre-processed by [7]. The dataset was split into 53,679 training, 2,447 development, and 12,839 test sequences. The test set contained one-Hamming distance neighbors of the top candidates from the training set. This allowed us to evaluate the model's ability to localize information from a broad sampling of relevant sequences.

The detailed Stability prediction results are listed in Table 7. As the data splits for this task were created by [7], no clear task-specific SOTA existed. Instead, we present the results obtained using RNN$_{BASE}$ and RNN$_{LARGE}$ models without pre-training.

E. FLUORESCENCE

For the Fluorescence task, we used the datasets from [31], which were pre-processed by [7]. The dataset was split into 21,446 training, 5,362 development, and 27,217 test sequences. The training dataset contained three-Hamming distance mutations, whereas the test dataset contained more than four mutations. This allowed us to evaluate the model's ability to generalize to unseen mutation combinations.

The detailed Fluorescence prediction results are presented in Table 8. As this task were created by [7], no clear task-specific SOTA exists. Instead, we present the results obtained using RNN$_{BASE}$ and RNN$_{LARGE}$ models without pre-training.

TABLE 9. Ablation studies on Homology and SecStr tasks.

Method	λ_{PT}	λ_{FT}	Homology						SecStr	
			acc	ρ	Class	Fold	Superfamily	Family	acc8	acc3
PLUS-RNN$_{BASE}$	0.7	0.3	**0.96**	**0.70**	**0.95**	**0.91**	**0.96**	**0.72**	**0.66**	**0.78**
RNN$_{BASE}$	-	0.3	0.93	0.67	0.88	0.81	0.92	0.68	0.61	0.73
(PT-A)	0.0	0.3	0.94	0.68	0.91	0.85	0.93	0.70	0.62	0.74
(PT-B)	0.5	0.3	**0.96**	0.69	**0.95**	**0.91**	0.95	0.70	0.66	0.77
(PT-C)	1.0	0.3	**0.96**	0.69	0.93	0.89	0.95	0.70	0.65	0.77
(FT-A)	0.7	0.0	0.94	0.68	0.91	0.85	0.93	0.70	0.65	0.77
(FT-B)	0.7	0.5	**0.96**	0.69	**0.95**	**0.91**	0.95	0.70	**0.66**	**0.78**

Note: We use the development sets for the ablation studies.

TABLE 10. Detailed transmembrane prediction results.

Method	acc
PLUS-TFM	0.82
PLUS-RNN$_{BASE}$	**0.89**
PLUS-RNN$_{LARGE}$	**0.89**
P-ELMo	0.78
MEMSAT-SVM†	0.67
Philius†	0.70
SPOCTOPUS†	0.71
TOPCONS†	0.80

† Excerpted from P-ELMo [12].

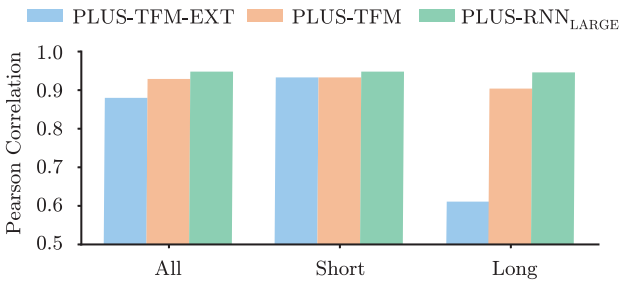

FIGURE 5. Homology prediction results for different lengths.

F. SECONDARY STRUCTURE (SecStr)

For the SecStr task, we used the training and development datasets from the PDB [52], which were pre-processed and provided by [7]. Any sequences with more than 25% sequence identity within the datasets were removed to avoid any bias due to homologous sequences. The training and development datasets contained 8,678 and 2,170 protein sequences, respectively. We used three test datasets: CB513 with 513 sequences [53], CASP12 with 21 sequences [54], and TS115 with 115 sequences [10].

The detailed SecStr prediction results are presented in Table 3. We report excerpted results from [7] and [32]. RaptorX [55] used a convolutional neural network to capture structural information and conditional neural fields to model secondary structure correlations among adjacent amino acids. It adopted the position-specific scoring matrix evolutionary features generated by searching the UniProt database [49] with PSI-BLAST [56]. NetSurfP-2.0 [32] used a convolutional neural network to learn motif information and a biRNN to learn their sequential dependencies. It adopted HMM evolutionary features from HH-bits [44] including amino acid profiles, state transition probabilities, and local alignment diversities.

G. TRANSMEMBRANE

For the Transmembrane task, we used the TOPCONS datasets [33], which were pre-processed and provided by [12]. The dataset was split into 228 training, 29 development, and 29 test sequences. The goal was to classify each amino acid into one of the following: the membrane, inside or outside of the membrane domains.

The detailed Transmembrane prediction results are presented in Table 10. We report the excerpted top-5 baseline results from [12]. Although this is an amino-acid-level prediction task, the evaluation was performed at the protein level, following the guidelines from TOPCONS. The predictions were judged correct if the protein had the same number of predicted and true transmembrane regions, and the predicted and true regions overlapped by at least five amino acids. TOPCONS [33], the best task-specific baseline model, is a meta-predictor that combines the predictions from five different predictors into a topology profile based on a dynamic programming algorithm.

APPENDIX E
ABLATION STUDIES

Here, we show results from ablation studies on the Homology and SecStr tasks to better understand the strengths and aspects of the PLUS framework. We used PLUS-RNN$_{BASE}$ as the baseline model unless explicitly stated otherwise. Note that we used the development sets for the ablation studies.

A. PRE-TRAINING AND FINE-TUNING OF PLUS-RNN

We explored the effect of using different λ_{PT} values for controlling the relative importance of the MLM and SFP pre-training tasks (Table 9). The results indicated that the pre-trained models with different λ_{PT} values (PLUS-RNN$_{BASE}$, PT-A, PT-B, PT-C) always outperformed the RNN$_{BASE}$ model trained from the scratch. Both pre-training

tasks consistently improve the prediction performance at all structural levels. Of the two pre-training tasks, removing MLM negatively affects the prediction performance more than removing the SFP. This coincides with the expected result, according to which, the MLM task would play the primary role, and the SFP task would complement MLM by encouraging the models to compare pairwise protein representations.

During the fine-tuning, we simultaneously trained a model for the MLM task as well as the downstream task. Moreover, we explored the effect of using different λ_{FT} values for controlling their relative importance (Table 9). The results showed that the models simultaneously fine-tuned with the MLM task loss (PLUS-RNN$_{BASE}$ and FT-B) consistently outperformed the (FT-A) model fine-tuned only with the task-specific loss. Based on this, we infer that the MLM task serves as a form of regularization and improves the generalization performance of the models.

B. COMPARISON OF PLUS-RNN AND PLUS-TFM

We compared the Homology prediction performances of PLUS-TFM and PLUS-RNN$_{LARGE}$ for protein pairs of different lengths (Figure 5). Because PLUS-TFM was pre-trained using protein pairs shorter than 512 amino acids, we denote *Long* for protein pairs longer than 512 amino acids and *Short* otherwise. Next, we evaluated PLUS-TFM for the *Long* protein pairs in the following two ways. First, we simply used the protein pairs as they were. Second, we truncated them to 512 amino acids. The former is denoted as PLUS-TFM-EXT (as in extended), and the latter is denoted as PLUS-TFM.

PLUS-RNN$_{LARGE}$ consistently provided competitive performance regardless of the protein length. In contrast, PLUS-TFM-EXT deteriorated for the *Long* protein pairs, whereas PLUS-TFM exhibited a relatively less performance degradation. The results presented the limitations of TFM models using the limited context size of 512 amino acids. Although the number of *Long* protein pairs in the Homology development dataset was relatively small (13.4%), complex proteins that are found in nature make the ability to analyze long protein sequences indispensable. Moreover, because this is due to the computational burden of TFM scaling quadratically with the input length, we predict that the recently proposed adaptive attention span approach [57] may be able to help improve PLUS-TFM.

REFERENCES

[1] J. M. Berg, J. L. Tymoczko, and L. Stryer, *Biochemistry*, vol. 38. New York, NY, USA: WH Freeman, 2006, p. 76.
[2] L. Holm and C. Sander, "Mapping the protein universe," *Science*, vol. 273, no. 5275, pp. 595–602, Aug. 1996.
[3] S. R. Eddy, "Where did the BLOSUM62 alignment score matrix come from?" *Nature Biotechnol.*, vol. 22, no. 8, pp. 1035–1036, Aug. 2004.
[4] J. Soding, A. Biegert, and A. N. Lupas, "The HHpred interactive server for protein homology detection and structure prediction," *Nucleic Acids Res.*, vol. 33, no. 2, pp. W244–W248, Jul. 2005.
[5] T. E. Creighton, *Proteins: Structures and Molecular Properties*. Basingstoke, U.K.: Macmillan, 1993.
[6] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Brief Bioinform.*, vol. 18, no. 5, pp. 851–869, 2016.
[7] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel, and S. Y. Song, "Evaluating protein transfer learning with tape," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9689–9701.
[8] O. Chapelle, S. Bernhard, and Z. Alexander, "Semi-supervised learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, p. 542, Apr. 2009.
[9] M. AlQuraishi, "End-to-end differentiable learning of protein structure," *Cell Syst.*, vol. 8, no. 4, pp. 292–301, 2019.
[10] K. K. Yang, Z. Wu, C. N. Bedbrook, and F. H. Arnold, "Learned protein embeddings for machine learning," *Bioinformatics*, vol. 34, no. 15, pp. 2642–2648, Aug. 2018.
[11] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church, "Unified rational protein engineering with sequence-based deep representation learning," *Nature Methods*, vol. 16, no. 12, pp. 1315–1322, Dec. 2019.
[12] T. Bepler and B. Berger, "Learning protein sequence embeddings using information from structure," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.
[13] A. Rives, S. Goyal, J. Meier, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *BioRxiv*, vol. 118, Apr. 2019, Art. no. 622803.
[14] N. Strodthoff, P. Wagner, M. Wenzel, and W. Samek, "USDMProt: Universal deep sequence models for protein classification," *Bioinformatics*, vol. 36, no. 8, pp. 2401–2409, 2020.
[15] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, and B. Rost, "Modeling aspects of the language of life through transfer-learning protein sequences," *BMC Bioinf.*, vol. 20, no. 1, p. 723, 2019.
[16] A. X. Lu, H. Zhang, M. Ghassemi, and A. Moses, "Self-supervised contrastive learning of protein representations by mutual information maximization," *bioRxiv*, Apr. 2020. [Online]. Available: https://www.biorxiv.org/content/10.1101/2020.09.04.283929v2, doi: 10.1101/2020.09.04.283929.
[17] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rihawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, and B. Rost, "ProtTrans: Towards cracking the language of Life's code through self-supervised deep learning and high performance computing," 2020, *arXiv:2007.06225*. [Online]. Available: http://arxiv.org/abs/2007.06225
[18] R. D. Finn, A. Bateman, J. Clements, P. Coggill, R. Y. Eberhardt, S. R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. L. L. Sonnhammer, J. Tate, and M. Punta, "Pfam: The protein families database," *Nucleic Acids Res.*, vol. 42, no. D1, pp. D222–D230, Jan. 2014.
[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*. [Online]. Available: http://arxiv.org/abs/1802.05365
[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[21] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS ONE*, vol. 10, no. 11, Nov. 2015, Art. no. e0141287.
[22] B. Krause, L. Lu, I. Murray, and S. Renals, "Multiplicative LSTM for sequence modelling," 2016, *arXiv:1609.07959*. [Online]. Available: http://arxiv.org/abs/1609.07959
[23] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*. [Online]. Available: http://arxiv.org/abs/1909.11942
[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473
[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[26] A. Elofsson and E. L. Sonnhammer, "A comparison of sequence and structure protein domain families as a basis for structural genomics," *Bioinformatics*, vol. 15, no. 6, pp. 480–500, Jun. 1999.
[27] N. K. Fox, S. E. Brenner, and J.-M. Chandonia, "SCOPe: Structural classification of proteins—Extended, integrating SCOP and ASTRAL data and classification of new structures," *Nucleic Acids Res.*, vol. 42, no. 1, pp. D304–D309, Jan. 2014.

[28] S. Khurana, R. Rawi, K. Kunji, G.-Y. Chuang, H. Bensmail, and R. Mall, "DeepSol: A deep learning framework for sequence-based protein solubility prediction," *Bioinformatics*, vol. 34, no. 15, pp. 2605–2613, Aug. 2018.
[29] J. J. Almagro Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, and O. Winther, "DeepLoc: Prediction of protein subcellular localization using deep learning," *Bioinformatics*, vol. 33, no. 21, pp. 3387–3395, Nov. 2017.
[30] G. J. Rocklin, T. M. Chidyausiku, I. Goreshnik, A. Ford, S. Houliston, A. Lemak, and L. Carter, "Global analysis of protein folding using massively parallel design, synthesis, and testing," *Science*, vol. 357, no. 6347, pp. 168–175, Jul. 2017.
[31] K. Sarkisyan, D. Bolotin, M. Meer, D. Usmanova, A. Mishin, and G. Sharonov, "Local fitness landscape of the green fluorescent protein," *Nature*, vol. 533, no. 7603, pp. 397–401, 2016.
[32] M. S. Klausen, "NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning," *BioRxiv*, vol. 87, no. 6, pp. 520–527, 2018, Art. no. 311209.
[33] K. D. Tsirigos, C. Peters, N. Shu, L. Käll, and A. Elofsson, "The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides," *Nucleic Acids Res.*, vol. 43, no. W1, pp. W401–W407, Jul. 2015.
[34] L. S. Tavares, C. S. F. Silva, V. C. de Souza, V. L. da Silva, C. G. Diniz, and M. O. Santos, "Strategies and molecular tools to fight antimicrobial resistance: Resistome, transcriptome, and antimicrobial peptides," *Frontiers Microbiology*, vol. 4, p. 412, Dec. 2013.
[35] J. H. Steiger, "Tests for comparing elements of a correlation matrix," *Psychol. Bull.*, vol. 87, no. 2, p. 245, Mar. 1980.
[36] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–12.
[37] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moult, "Critical assessment of methods of protein structure prediction (CASP)—Round XIII," *Proteins, Struct., Function, Bioinf.*, vol. 87, no. 12, pp. 1011–1020, Dec. 2019.
[38] R. Poplin, P.-C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, D. Newburger, J. Dijamco, N. Nguyen, P. T. Afshar, S. S. Gross, L. Dorfman, C. Y. McLean, and M. A. DePristo, "A universal SNP and small-indel variant caller using deep neural networks," *Nature Biotechnol.*, vol. 36, no. 10, pp. 983–987, Nov. 2018.
[39] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS Autodiff Workshop*, 2017, pp. 1–4.
[40] N. Sung, M. Kim, H. Jo, Y. Yang, J. Kim, L. Lausen, Y. Kim, G. Lee, D. Kwak, J.-W. Ha, and S. Kim, "NSML: A machine learning platform that enables you to focus on your models," 2017, *arXiv:1712.05902*. [Online]. Available: http://arxiv.org/abs/1712.05902
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[43] T. L. Bailey, N. Williams, C. Misleh, and W. W. Li, "MEME: Discovering and analyzing DNA and protein sequence motifs," *Nucleic Acids Res.*, vol. 34, no. Web Server, pp. W369–W373, Jul. 2006.
[44] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment," *Nature Methods*, vol. 9, no. 2, p. 173, 2012.
[45] Y. Zhang, "TM-align: A protein structure alignment algorithm based on the TM-score," *Nucleic Acids Res.*, vol. 33, no. 7, pp. 2302–2309, Apr. 2005.
[46] H. Berman, J. Westbrook, M. Gabanyi, W. Tao, R. Shah, and A. Kouranov, "The protein structure initiative structural genomics knowledgebase," *Nucleic acids Res.*, vol. 37, no. 1, pp. D365–D368, 2009.
[47] C. C. H. Chang, J. Song, B. T. Tey, and R. N. Ramanan, "Bioinformatics approaches for improved recombinant protein production in escherichia coli: Protein solubility prediction," *Briefings Bioinf.*, vol. 15, no. 6, pp. 953–962, Nov. 2014.
[48] R. Rawi, R. Mall, K. Kunji, C.-H. Shen, P. D. Kwong, and G.-Y. Chuang, "PaRSnIP: Sequence-based protein solubility prediction using gradient boosting machine," *Bioinformatics*, vol. 34, no. 7, pp. 1092–1098, Apr. 2018.
[49] R. Apweiler, A. Bairoch, C. H. Wu, and W. C. Barker, "UniProt: The universal protein knowledgebase," *Nucleic Acids Res.*, vol. 46, no. 5, pp. 115–119, Mar. 2018.
[50] K.-C. Chou, Z.-C. Wu, and X. Xiao, "iLoc-Euk: A multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins," *PLoS ONE*, vol. 6, no. 3, Mar. 2011, Art. no. e18258.
[51] A. Bairoch, "The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 45–48, Jan. 2000.
[52] H. M. Berman, P. E. Bourne, J. Westbrook, and C. Zardecki, "The protein data bank," in *Protein Structure*. Boca Raton, FL, USA: CRC Press, 2003, pp. 394–410.
[53] J. A. Cuff and G. J. Barton, "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction," *Proteins, Struct., Function, Genet.*, vol. 34, no. 4, pp. 508–519, Mar. 1999.
[54] L. A. Abriata, G. E. Tamá, B. Monastyrskyy, A. Kryshtafovych, and M. D. Peraro, "Assessment of hard target modeling in CASP12 reveals an emerging role of alignment-based contact prediction methods," *Proteins, Struct., Function, Bioinf.*, vol. 86, pp. 97–112, Mar. 2018.
[55] S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Sci. Rep.*, vol. 6, Jan. 2016, Art. no. 18962.
[56] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, no. 17, pp. 3389–3402, 1997.
[57] S. Sukhbaatar, E. Grave, P. Bojanowski, and A. Joulin, "Adaptive attention span in transformers," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1–5.

SEONWOO MIN received the B.S. and Ph.D. degrees in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2015 and 2021, respectively. He is currently a Research Scientist with the Fundamental Research Laboratory, LG AI Research. His current research interests include machine learning and bioinformatics. He was a recipient of the Global Ph.D. Fellowship, in 2016, the Microsoft Research Asia Fellowship Nomination Award, in 2018, the Korea National Excellent Researcher Award, in 2020, and many other prestigious awards.

SEUNGHYUN PARK received the B.S. degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from Korea University, South Korea, in 2009 and 2018, respectively. He was a Researcher with the Department of Electrical and Computer Engineering, Seoul National University, South Korea, from 2011 to 2018. He is currently a Research Engineer at Clova AI Research, NAVER Corporation, South Korea. His research interests include machine learning, natural language processing, computational methods in statistics, and medical informatics.

SIWON KIM received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2018, where she is currently pursuing the integrated M.S. and Ph.D. degree in electrical and computer engineering. Her research interests include deep learning, explainable AI, and biomedical applications.

HYUN-SOO CHOI received the B.S. degree in computer and communication engineering for the first major and in brain and cognitive science for the second major from Korea University, in 2013, and the integrated M.S. and Ph.D. degree in electrical and computer engineering from Seoul National University, South Korea, in 2020. From 2020 to 2021, he was a Senior Researcher at Vision AI Labs, SK Telecom. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Kangwon National University, South Korea.

BYUNGHAN LEE (Member, IEEE) received the B.S. degree in electrical engineering from Korea University, South Korea, in 2011, and the Ph.D. degree in electrical and computer engineering from Seoul National University, South Korea, in 2018. He is currently an Assistant Professor with the Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology. His research interests include machine learning, artificial intelligence, and their biomedical applications.

SUNGROH YOON (Senior Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, South Korea, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2002 and 2006, respectively. From 2006 to 2007, he was with Intel Corporation, Santa Clara, CA, USA. From 2007 to 2012, he was an Assistant Professor with the School of Electrical Engineering, Korea University. From 2016 to 2017, he was a Visiting Scholar with the Department of Neurology and Neurological Sciences, Stanford University. He held research positions at Stanford University and Synopsys, Inc., Mountain View, CA, USA. He is currently a Professor with the Department of Electrical and Computer Engineering, Seoul National University. His current research interests include machine learning and artificial intelligence. He was a recipient of the IEEE Young IT Engineer Award, in 2013, the SBS Foundation Award, in 2016, the IMIA Best Paper Award, in 2017, the SNU Education Award, in 2018, the IBM Faculty Award, in 2018, the Korean Government Researcher of the Month Award, 2018, the BRIC Best Research of the Year, in 2018, the Shin-Yang Engineering Research Reward, in 2019, the Microsoft Collaborative Research Grant, in 2017 and 2020, and many other prestigious awards. Since February 2020, he has been serving as the Chairperson (Minister) for the Presidential Committee on the Fourth Industrial Revolution established by the Korean Government.