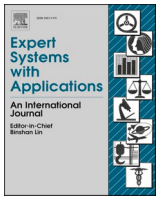
Expert Systems With Applications 282 (2025) 127786

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



TaiChiNet: PCA-based Ying-Yang dilution of inter- and intra-BERT layers to represent anti-coronavirus peptides

Kewei Li [a,b], Shiying Ding [a], Zhe Guo [d], Yusi Fan [a], Hongmei Liu [a,b], Yannan Sun [c], Gongyou Zhang [b], Ruochi Zhang [d,*], Lan Huang [a], Fengfeng Zhou [a,b,*]

[a] College of Computer Science and Technology, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China
[b] School of Biology and Engineering, Guizhou Medical University, Guiyang 550025 Guizhou, China
[c] School of Software, Jilin University, Changchun 130012 Jilin, China
[d] School of Artificial Intelligence, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China

ARTICLE INFO

Keywords:
TaiChiNet
Ying-Yang dilution network
BERT layer fusion
Principal Component Analysis (PCA)
Explainability

ABSTRACT

Numerous studies have demonstrated that biological sequences, such as DNA, RNA, and peptide, can be considered the "language of life". Utilizing pre-trained language models (LMs) like ESM2, GPT, and BERT have yielded state-of-the-art (SOTA) results in many cases. However, the increasing size of datasets exponentially escalates the time and hardware resources required for fine-tuning a complete LM. This paper assumed that natural language shared linguistic logic with the "language of life" like peptides. We took the LM BERT model as an example in a novel Principal Component Analysis (PCA)-based Ying-Yang dilution network of the inter- and intra-BERT layers, termed TaiChiNet, for feature representation of peptide sequences. The Ying-Yang dilution architecture fuses the PCA transformation matrices trained on positive and negative samples, respectively. We transferred the TaiChiNet features into a subtractive layer feature space and observed that TaiChiNet just rotated the original subtractive features with a certain angle and didn't change the relative distance among the dimensions. TaiChiNet-engineered features together with the hand-crafted (HC) ones were integrated for the prediction model of anti-coronavirus peptides (TaiChiACVP). Experimental results demonstrated that the TaiChiACVP model achieved new SOTA performance and remarkably short training time on five imbalanced datasets established for the anti-coronavirus peptide (ACVP) prediction task. The decision paths of the random forest classifier illustrated that TaiChiNet features can complement HC features for better decisions. TaiChiNet has also learned the latent features significantly correlated with physicochemical properties including molecular weight. This makes an explainable connection between the deep learning-represented features and the ACVP-associated physicochemical properties. Additionally, we extended our work to the other LMs, including ESM2 with 6 and 12 layers, ProGen2 small and base version, ProtBERT, and ProtGPT2. Due to the limitations of these recent LMs, none of them outperforms TaiChiACVP. However, some limitations of TaiChiNet remained to be investigated in the future, including learnable rotation degrees, extended fusions of more layers, and end-to-end training architecture. The source code is freely available at: http://www.healthinformaticslab.org/supp/resources.php.

1. Introduction

The recent outbreak of the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) caused a world-wide pandemic of acute respiratory disease called coronavirus disease 2019 (COVID-19), which has substantially increased the research community's interest on developing new SARS-CoV-2 vaccines (Amanat & Krammer, 2020; Dong, et al., 2020; Krammer, 2020) and forecasting the driver mutations

* Corresponding authors.
E-mail addresses: kwbb1997@gmail.com (K. Li), dingsy1999@163.com (S. Ding), gzhe2023@163.com (Z. Guo), fan_yusi@163.com (Y. Fan), hmliu@gmc.edu.cn (H. Liu), zgy1943541699@163.com (G. Zhang), zrc720@gmail.com (R. Zhang), huanglan@jlu.edu.cn (L. Huang), ffzhou@jlu.edu.cn, FengfengZhou@gmail.com (F. Zhou).

https://doi.org/10.1016/j.eswa.2025.127786
Received 21 September 2023; Received in revised form 7 April 2024; Accepted 15 April 2025
Available online 19 April 2025

Table 1
The five benchmark datasets used in this study. The columns "P" and "N" gave the numbers of positive and negative samples in a dataset. Each row gave the information of one dataset, which was pre-split into the Training and Testing subsets.

ID	Dataset	Training subset		Testing subset	
		P	N	P	N
0	Anti-Virus	95	1,399	42	600
1	non-AVP	95	3,746	42	1,566
2	non-AMP	95	3,485	42	1,494
3	All-Neg	95	8,535	42	3,660
4	All-AMP	95	5,050	42	2,166

Table 2
The terms used in this paper and their corresponding meanings.

Term	Interpretation
k	The number of the first k BERT layers, $k = 1, 2, \ldots, 12$
BERTA	Given a training dataset, calculate the engineered features through the first k layers of BERT, and chose the top-ranked principal components (PCs) explaining 95 % variance of the dataset by the PCA, trained on the training set
PCBERTA	Baseline version of TaiChiNet. See Algorithm 1.
TaiChiNet	The main algorithm proposed in this study. See Algorithm 2 and Fig. 1.
HC	Abbreviation for the hand-crafted features
TaiChiACVP	The ACVP prediction model integrating the TaiChiNet-engineered and HC features
D, L	Training set, and training labels
D^+	Training set with only positive samples
D^-	Training set with only negative samples
$\|D\|$	The size of training set D
n_layers	The number of the total layers of BERT. n_layers = 12 in this work
n_embedding	The dimension of the BERT embedding features. n_embedding = 768 in this work
max_len	The max length of the input sequences to BERT
$\boldsymbol{PCA_k^+}$, $\boldsymbol{PCA_k^-}$	PCA-based positive/negative inter layer fusion matrix
$\boldsymbol{C^+}$, $\boldsymbol{C^-}$	The average value of BERT layers of positive and negative samples in the training set
$\boldsymbol{PCA_i}$	PCA-based intra layer fusion matrix for the i^{th} layer, $i = 1, \ldots, k$

in the future SARS-CoV-2 strains of concern (Maher, et al., 2022). Antimicrobial peptides (AMPs) have emerged as a promising treatment option in light of the escalating antibiotic resistance (Prevention, 2019; Renaud & Mansbach, 2023; Wan, Kontogiorgos-Heintz, & de la Fuente-Nunez, 2022). These peptides exhibit a wide range of biological activities, such as anti-virus, anti-coronavirus, and anti-fungi, among others (Bin Hafeez, Jiang, Bergen, & Zhu, 2021).

Accurately identifying AMPs is crucial for the discovery of novel antimicrobial drugs and treatments. Anti-coronavirus peptide (ACVP) is one kind of AMP, and has been reported as pivotal therapeutic agents against coronavirus (Y. Pang, Wang, Jhong, & Lee, 2021). Zhang et al., discovered an anti-microbial peptide DP7 with potential activities against coronavirus infections via computer screening and wet-lab experimental confirmation (R. Zhang, et al., 2021). Another study observed that the positive interfacial hydrophobicity of the peptide LL-37 resulted in disruption of COVID-19 viral membrane (Nireeksha, Gollapalli, Varma, Hegde, & Kumari, 2022). The recent advancement of artificial intelligence (AI) technologies has facilitated the development of efficient AMP and ACVP identifications and characterizations.

The enhanced generalization capability of LMs relies on extensive training data and high computational requirement (Yang, et al., 2023). These characteristics often impeded the widespread adoptions of LMs in small- and medium-sized enterprises as well as academic institutions with limited computing resources. This issue may be addressed by model compression techniques, including pruning (Gordon, Duh, & Andrews, 2020; Han, Mao, & Dally, 2015; Paul Michel and Neubig, 2019), quantization (Han, et al., 2015), and distillation (Geffrey Hinton and Dean, 2015; Jiao, et al., 2020; Sanh, Debut, Chaumond, & Wolf, 2019). These approaches aim to build simplified versions of LMs with reduced computing resource requirements and similarly good prediction performances for resource-constrained settings.

This study introduced a novel principal component analysis (PCA)-based Ying-Yang dilution strategy, termed TaiChiNet, for the transformer encoder-based layers of the LM BERT model. We applied the TaiChiNet-engineered features with five distinct types of HC features to the ACVP prediction task, and outperformed the existing algorithms on the benchmark PreAntiCoV datasets. The key contributions of this study are summarized as follows:

- A novel PCA-based peptide representation framework, TaiChiNet, was proposed, and demonstrated that the prediction performance of language models (LMs) with Random Forest (RF) classifiers is influenced by the rotation degree of subtractive layer features.
- TaiChiNet features exhibited explainable values, and showed correlations with the physicochemical properties of antimicrobial peptides.
- The TaiChiNet-engineered features complemented hand-crafted (HC) features, and their combination resulted in optimal prediction performance.

2. Related work

Diverse hand-crafted (HC) features may be calculated to represent the amino-acid-based descriptors and physiochemical properties of AMPs and ACVPs. Most machine learning algorithms cannot directly handle a peptide sequence, and take the peptide HC features as the input. Lawrence et al., extracted physicochemical properties of AMPs and trained an accurate random forest-based AMP classifier (amPEPpy 1.0) (Lawrence, et al., 2021). Pang et al., developed an accurate ACVP identifier PreAntiCoV by evaluating various negative datasets (Y. Pang, et al., 2021). PreAntiCoV comprehensively utilized multiple encoding strategies to represent the amino acid-based descriptors and physicochemical properties of ACVPs. The datasets (Y. Pang, et al., 2021) have also been widely used in the other ACVP (Kurata, Tsukiyama, & Manavalan, 2022; Timmons & Hewage, 2021), anti-virus peptide (AVP) (Yuxuan Pang, Yao, Jhong, Wang, & Lee, 2021; Wei, Zhou, Chen, Song, & Su, 2018), and database (Q. Zhang, et al., 2022) studies.

Deep learning algorithms have also been extensively used to learn the latent features for the identifications of AMPs and ACVPs. Timmons and Hewage trained a fully-connected neural network on two large datasets ENNAVIA-A ENNAVIA-B, and transferred the pre-trained models to binary ACVP classification tasks on two small ACVP datasets (Timmons & Hewage, 2021). The developed tool ENNAVIA achieved an external test accuracy of 93.9 %. Kurata et al., used a dataset-specific word2vec model to represent ACVPs and achieved the state-of-the-art prediction performance with their iACVP model (Kurata, et al., 2022). AMPScanner leveraged the deep learning-based word embeddings of peptide sequences and long short-term memory (LSTM)-based representations for the AMP recognition task (Veltri, Kamath, & Shehu, 2018). Multi-scale convolutional neural network (msCNN) was also proven effective in AMP prediction (Su, Xu, Yin, Quan, & Zhang, 2019). Zhou et al., showed that three heterogeneous types of peptide features may be represented by different deep neural networks and their fused feature representation facilitated remarkably adaptive and effective AMP identification (Zhou, et al., 2023).

The language models (LMs) have recently emerged as powerful tools in multiple natural language processing (NLP) tasks. Researchers have also discovered the interdisciplinary applications of LMs in extracting co-evolution information from protein sequences, even in the absence of multiple sequence alignment (MSA) data (Verkuil, et al., 2022). ProtTrans (Elnaggar, et al., 2022) re-trains a series of natural language models, including ProtBERT and ProtT5. However, since biological sequences carry inherent co-evolutionary information, such sequence data have a relatively obvious tendency to cluster (Suzek, Huang, McGarvey,

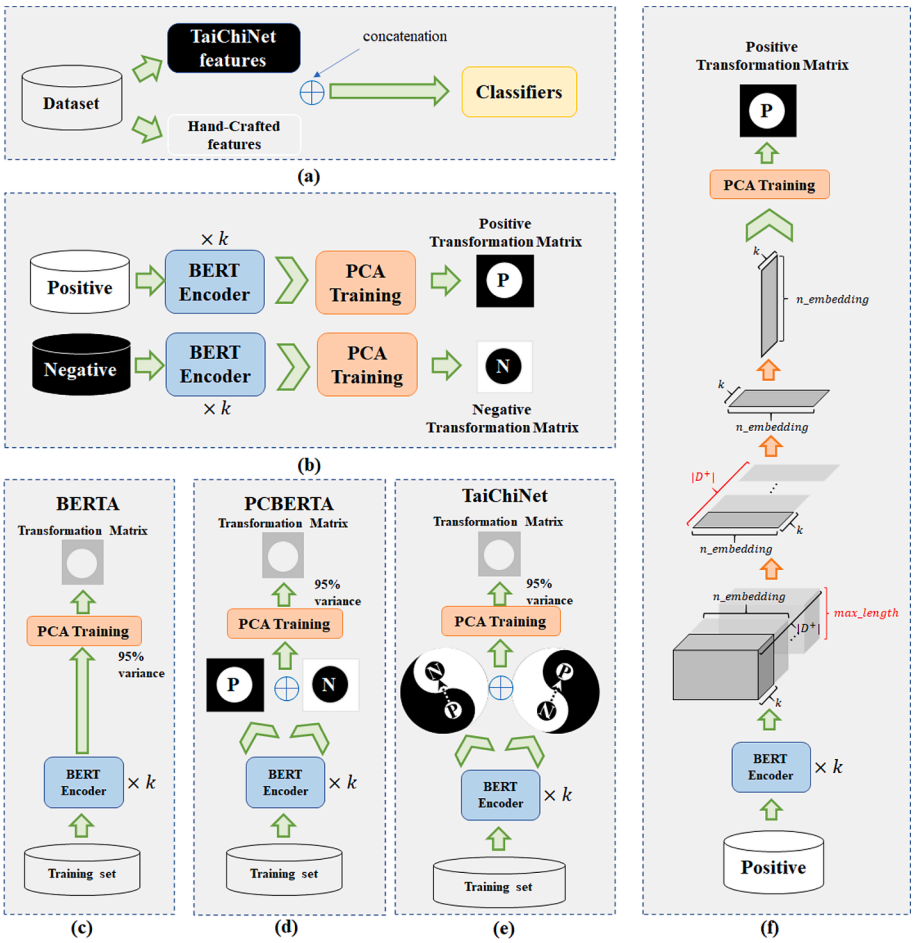


Fig. 1. Illustration of the network frameworks of this study. (a) illustrated the framework of the ACVP prediction models evaluated in this study. (b) described the training step of TaiChiNet. The positive and negative samples were transformed through the first *k* transformer encoder layers of the pre-trained BERT model, and then the PCA calculations, respectively. We got the positive and negative transformation matrices. (f) showed the detailed procedure of calculating positive transformation matrix. (c) and (d) showed the baseline BERTA and PCBERTA frameworks, while (e) illustrated the Ying-Yang dilution architecture TaiChiNet.

Mazumder, & Wu, 2007), and a carefully-designed dataset splitting strategy is very important for the establishment of robust models (Meier, et al., 2021). The Meta team extended BERT into the ESM2 model through strict data partitioning, and its performance far exceeds ProtBERT (Lin, et al., 2023). Before publishing ESM2, the Meta team also computationally proved the impacts of data partitioning on the performance of ESM2. Some data partitions can easily lead to model overfitting and make it difficult to continue the model training process (Meier, et al., 2021). The essence of data partitioning is to adaptively adjust the loss function weight of the model (Meier, et al., 2021). ProGen is another transformer-decoder autoregressive model (Nijkamp, Ruffolo, Weinstein, Naik, & Madani, 2023). Erik Nijkamp et al. have explored the capacity of autoregressive models on protein sequences, and found that autoregressive models may also have the scaling law pattern on protein sequences (Nijkamp, et al., 2023). And Noelia Ferruz et al. retrained the GPT2 model as ProtGPT2 on the protein sequences (Ferruz, Schmidt, & Höcker, 2022).

A peptide may be viewed as a subsequence of a protein, and it is reasonable to speculate that LMs may also be used to extract important information of peptides for the subsequent prediction tasks. Multiple studies exerted the successful employments of LMs in the AMP prediction tasks (Dee, 2022; Y. Zhang, Lin, Zhao, Zeng, & Liu, 2021). To the best of our knowledge, ESM2 is the most popularly-used LMs on protein design, but it remains to be improved for its peptide representation capability, since some researchers found that the scaling law pattern of ESM2 didn't work well on peptides (Fernandez-Diaz et al., 2023).

3. Materials and methods

3.1. Datasets

This study evaluated the proposed TaiChiNet peptide representation framework by the benchmark datasets derived from (Y. Pang, et al., 2021). Pang et al., proposed one of the first few ACVP prediction models (Manavalan, Basith, & Lee, 2022), and extensively constructed five imbalanced classification benchmark datasets with different negative samples, i.e., Anti-Virus (anti-virus peptides excluding the ACVPs as negative samples), non-AVP (anti-microbial peptides excluding anti-virus peptides as negative samples), non-AMP (peptides excluding AMPs as negative samples), All-Neg (all peptides excluding ACVPs as negative samples), and All-AMP (all anti-microbial peptides excluding ACVPs as negative samples). The known ACVPs served as the positive samples of all the five datasets, and the detailed information of these datasets were shown in Table 1.

These datasets from (Y. Pang, et al., 2021) had been popularly used in ACVP classification (Kurata, et al., 2022; Manavalan, et al., 2022; Yuxuan Pang, et al., 2021; Timmons & Hewage, 2021), anti-bacterial peptide prediction (Singh, Shrivastava, Kumar Singh, Kumar, & Saxena, 2022), AMP prediction (Yan, Lv, Guo, Peng, & Liu, 2023), and database constructions (Jhong, et al., 2022; Q. Zhang, et al., 2022).

Algorithm 1: PCBERTA
Input: D, L, k
Output: PCA_k^+, PCA_k^-, PCA_i $(i = 1,...,k)$

1 // calculate the token-level features of the 12 BERT layers
// $D_0 \in \mathcal{R}^{n\_layers \times |D| \times \max\_len \times n\_embedding}$
D_0=BERT(D).all_hidden_states

2 select the number of layers k with the best metric $BF(k)$

3 // get the sequence-level features
// $D_1 \in \mathcal{R}^{k \times |D| \times n\_embedding}$
D_1= D_0[:k].mean(-2)

4 // split the dataset into the positive and negative data matrices
(P_{tr}, N_{tr}) = (D_1[$Label$ == 1], D_1[$Label$ == 0])

5 // Calculate the sample-level mean values of P_{tr} and N_{tr}
$P_{tr}^m, N_{tr}^m = P_{tr}.mean(1), N_{tr}.mean(1)$

6 // Calculate the PCA-based positive and negative inter-layer fusion matrices
$PCA_k^+, C^+ = pca.fit(P_{tr}^{mT})$
$PCA_k^-, C^- = pca.fit(N_{tr}^{mT})$

7 // Pass each sample in D_1 through PCA_k^+ and PCA_k^-
// Concatenate the fused features and get $F_1 \in \mathcal{R}^{k \times |D| \times 2n\_embedding}$
for sample in range(D_1.shape[0]):
F_1^+[:,sample,:] = ((D_1[:,sample,:].T-C^+) @ PCA_k^{+T}).T
F_1^-[:,sample,:] = ((D_1[:,sample,:].T-C^-) @ PCA_k^{-T}).T
F_1 = concatenate(F_1^-, F_1^+, -1)

8 // Calculate the intra-layer fusion matrices $PCA_i, i = 1,..,k$ of F_1
for i in range(k):
$PCA_i.fit(F_1[i])$
// Select the transformation matrix of the top-ranked PCs for the 95% variance

Fig. 2. The pseudocode of PCBERTA.

Algorithm 2: TaiChiNet (only the step 7 in Algorithm 1 was altered)

7 for sample in range(D_1.shape[0]):
F_1^+[:,sample,:] = (((D_1[:,sample,:].T-C^+) @ $PCA_k^{+T} - C^-$) @ PCA_k^{-T}).T
F_1^-[:,sample,:] =(((D_1[:,sample,:].T-C^-) @ $PCA_k^{-T} - C^+$) @ PCA_k^{+T}).T
F_1 = concatenate(F_1^-, F_1^+, -1)

Fig. 3. The pseudocode of TaiChiNet. This pseudocode was the same as PCBERTA except for the altered step 7.

3.2. Definitions of terms and variables

Table 2 defined some terms and variables frequently used in this study for the convenience of writings.

3.3. Performance metrics

This study focused on the five binary classification tasks shown in Table 1. A binary classification dataset consisted of positive and negative samples. True positive (*TP*) and true negative (*TN*) were the numbers of correctly predicted positive and negative samples, respectively. The numbers of incorrectly predicted positive and negative samples were false negative (*FN*) and false positive (*FP*), respectively.

A binary classification model could be evaluated by the performance metric accuracy, defined as $Acc=(TP + TN)/(TP + FN + TN + FP)$. However, the five datasets in Table 1 were highly imbalanced. So the metric geometric mean $Gmean = \text{sqrt}(Sn \times Sp)$ was used to evaluate an imbalanced binary classification model, where sensitivity $Sn = TP/(TP + FN)$, specificity $Sp = TN/(TN + FP)$, and sqrt() was the square root function. This study calculated the performance metrics by the 10-fold cross validation strategy on the training set. To be specific, this study utilized the same classifiers in the literature (Y. Pang, et al., 2021), a random forest classifier with the down-sampling strategy NearMiss version 3 and a balanced random forest classifier. Both classifiers used grid search for parameter optimization with 10-fold cross validation. The details can be found in Supplementary Table S1. Except that the balanced factor calculation in the section 4.4 used the 10-fold cross validation on the training set, all of the other results were on the test sets. The section 4.2, section 4.3, and section 4.11.1 used the test sets to evaluate the balanced random forest classifier. The other results used the classifier with better Gmean of 10-fold cross-validation on the training sets.

The proposed TaiChiNet framework fused the first k layers of the LM BERT, and we evaluated this proposed framework on the five datasets described in Table 1. A combined optimization goal was defined as the balanced factor:

$$BF(k)\frac{\mu(Gmean(i))}{\frac{k}{12}+\sigma(Gmean(i))}, \quad (1)$$

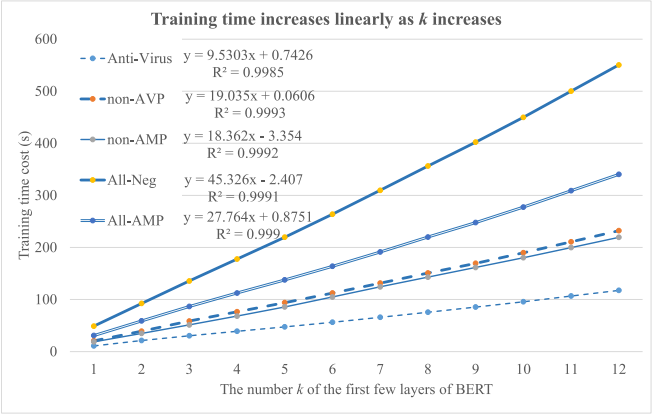


Fig. 4. Evaluation of the training time cost of the five classification tasks. The horizontal axis listed the number of the first few layers of BERT. The vertical axis gave the training time cost of each classification task counted in seconds. The formula and the R2 value of the trend line of each classification task was given after the line title.

where *Gmean*(*i*) was the performance metric value *Gmean* of the currently evaluated model on the dataset *i* ($i = 0, 1, 2, 3,$ or 4), μ() and σ() are the mean and standard deviation of the five *Gmean*(*i*) values.

The time cost of training the TaiChiNet framework was positively related with the parameter *k*. Therefore, the combined metric balanced factor (*BF*(*i*)) simultaneously considered the training time cost and prediction performance. The prediction models were evaluated on the individual datasets by the metric *Gmean*.

3.4. The Ying-Yang dilution network frameworks of this study

The network frameworks in this study were illustrated in Fig. 1. Two classifiers, balanced random forest (BRF) and random forest with the NearMiss3 down sampling strategy (SRF) (J. Zhang & Mani, 2003), were employed for the imbalanced classification tasks of ACVPs. A grid search was conducted to find the optimal parameter selection using the 10-fold cross validation strategy and the metric *Gmean*. Fig. 1 (a) showed the overall framework of the TaiChiACVP model using both TaiChiNet-engineered and HC features. The positive (P) and negative (N) transformation matrices were calculated during the training step in Fig. 1 (b). Took the procedure of calculating positive transition matrix as an example. We passed the positive samples in the training set to the BERT models and calculated the first *k* layer of BERT features. Let the output be the $|D^+| \times k \times$ max_*length* $\times$ *n_embedding* dimension features. We firstly took the mean values in the axis of amino acids to get the general representation of each peptide sequence, which was $|D^+| \times k \times$ *n_embedding* dimension of the output. And then we calculated the mean values in the axis of samples to get a general representation of the positive samples with dimension $k \times$ *n_embedding*. Finally, we transposed the result and passed it through the PCA transformation matrix for training and got the positive matrix. Fig. 1 (f) illustrated the procedure. The baseline BERTA, PCBERTA and the proposed TaiChiNet frameworks were illustrated in Fig. 1 (c), (d) and (e), respectively. The experimental data supported the necessity of upgrading the positive (P) and negative PCA transformation matrices to the Ying-Yang dilution architectures P + N and N + P, respectively.

3.5. The baseline PCBERTA framework

We initially built the baseline PCA-based fusion framework PCBERTA, and Fig. 2 illustrated its pseudocode. The essence idea was to extract the linear relationships between (steps 5 and 6) and within (steps 7 and 8) of the first *k* BERT layers. We anticipated that the first *k* BERT layers may deliver the important encoding capabilities to represent the differences between the ACVPs and the negative peptides.

3.6. The proposed TaiChiNet framework

TaiChiNet upgraded the PCA transformations (Step 7 in Fig. 2) of PCBERTA, as illustrated in Fig. 1 (e). This alternation was found to improve the ACVP prediction performances. Fig. 3 showed only the altered step 7 of Fig. 2, and TaiChiNet had the same operations in the other steps of PCBERTA in Fig. 2.

3.7. The HC features

HC features have been successfully employed in many sequence-based prediction tasks (Lawrence, et al., 2021; Y. Pang, et al., 2021). This study combined the TaiChiNet-engineered features with the same five types of HC features in (Y. Pang, et al., 2021). The normalized

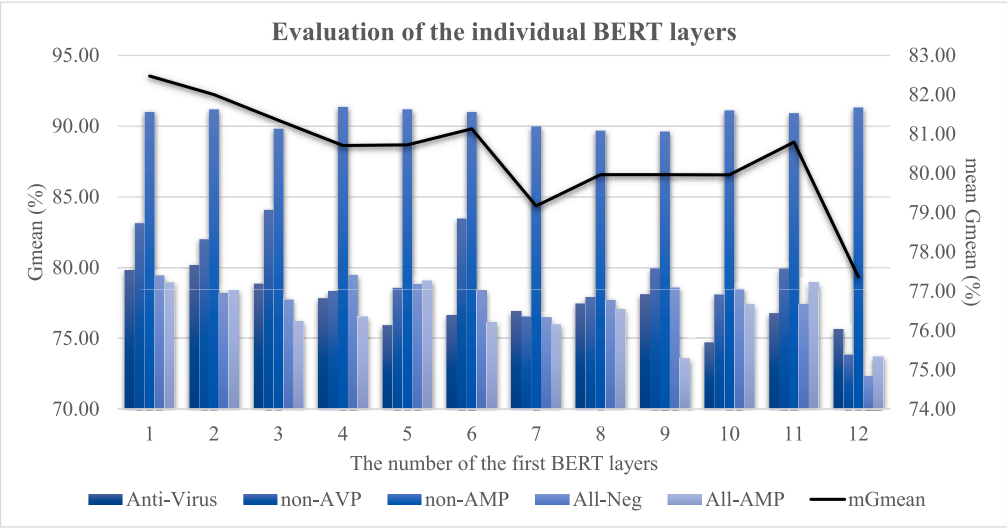


Fig. 5. Evaluation of the latent features encoded by the individual BERT layers. The horizontal axis gave the number of the first BERT layers used for the peptide encoding. The left vertical axis was the Gmean metric (%) for the histogram plots of the five classification tasks, i.e., Anti-Virus, non-AVP, non-AMP, All-Neg, and All-AMP, and the right vertical axis gave the mean value (line plot) of the Gmean metric over the five classification tasks. The classifier BRF was used to build the prediction models.

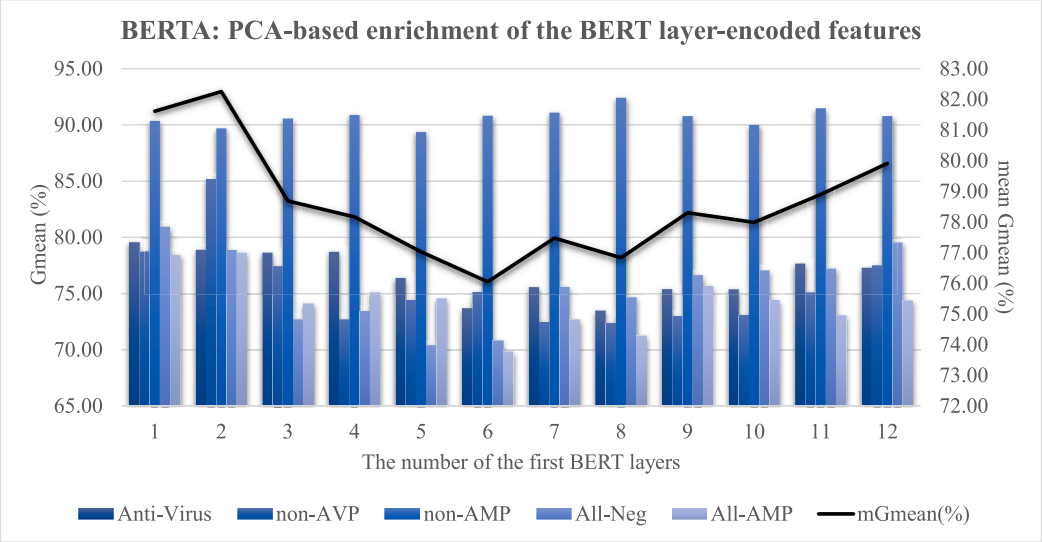


Fig. 6. Evaluation of the BERTA-encoded features. The structure of BERTA was shown in Fig. 1 (c). The horizontal axis gave the number of the first BERT layers used for the peptide encoding. The left vertical axis was the Gmean metric (%) for the histogram plots of the five classification tasks, i.e., Anti-Virus, non-AVP, non-AMP, All-Neg, and All-AMP, and the right vertical axis gave the mean value (line plot) of the Gmean metric over the five classification tasks. The classifier BRF was used to build the prediction models. The PCA transformation matrices of the individual BERT layers were trained on the training subsets, and fit on the Training and Testing subsets.

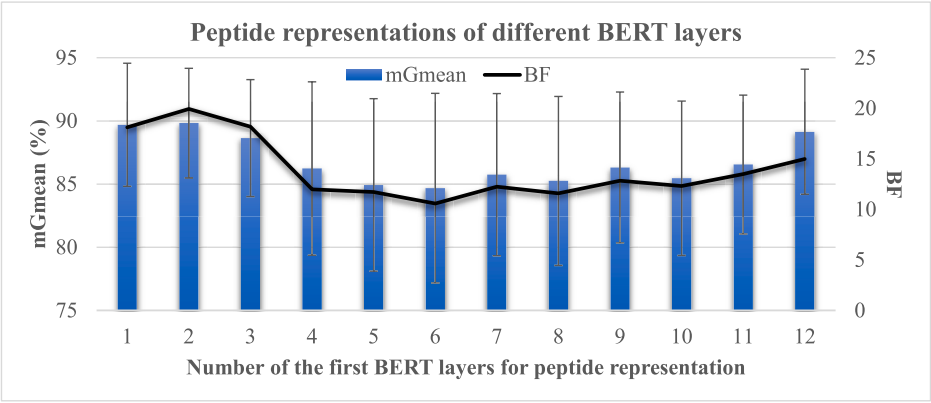


Fig. 7. Peptide representations of different BERT layers for the ACVP prediction task. The horizontal axis gave the number of the first BERT layers used for the peptide representations. The left vertical axis was the average value (mGmean) and the standard deviation (error bars) of the Gmean metric (%) over the five classification tasks, i.e., Anti-Virus, non-AVP, non-AMP, All-Neg, and All-AMP, and the right vertical axis gave the metric BF (line plot) of the Gmean metric over the five classification tasks. The BERTA framework was used in this experiment.

occurrence rates of the 20 amino acids in a peptide were calculated as the 20-dimension AAC features. The normalized occurrence rates of the paired amino acids were also calculated as the DiC features with 400 dimensions. CKSAAGP was a modified composition of k-pair amino acids (Chen & Li, 2022). PAAC improved the AAC algorithm by introducing a set of discrete factors (Chen & Li, 2022), and PHYC was short for eight physicochemical features (Meher, Dash, Sahu, Satpathy, & Pradhan, 2022). The detailed definitions of these five types of HC features could be found in (Y. Pang, et al., 2021).

4. Results

4.1. Training time linearly increased as the number of BERT layers

The standard pre-trained BERT model consisted of 12 layers and Fig. 4 showed that the training time cost of each classification task linearly correlated with the number *k* of the BERT layers. All the trend lines were formulated as linear functions, and all the R2 values were larger than 0.9900. Therefore, the training time cost could be accurately predicted by these trend lines and the number *k* of the BERT layers. The overall optimization goal *BF* used the parameter *k* to represent the time cost in evaluating a prediction framework over the five classification tasks.

4.2. Evaluation of the individual BERT layers

The five classification tasks showed the overall descending trend in the performance metric Gmean, as the number of the first BERT encoder layers increases (Fig. 5). The mean Gmean values reached the highest two values 82.48 % and 82.00 % using the first one and two BERT layers, respectively. This observation was anticipated since BERT was not designed and optimized for the peptide-based downstream tasks, and the last few layers of BERT aimed for the high-level abstractions of natural languages (Jawahar, Sagot, & Seddah, 2019). However, it shed light on the possibility and efficacy of utilizing only the first few BERT layers for the peptide-based downstream tasks in this study.

This approach offered several advantages, including reduced time cost while still achieving competitive performance levels, as

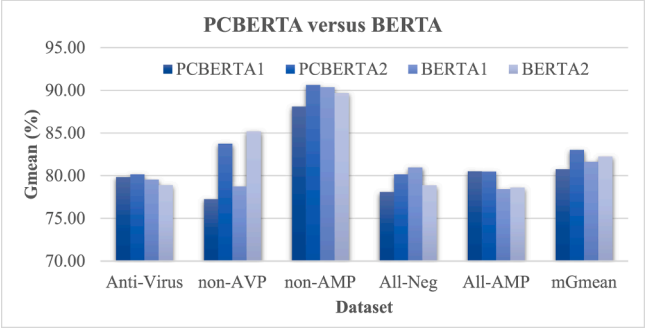


Fig. 8. Performance evaluation of PCBERTA and BERTA. The horizontal axis listed the five prediction datasets and mGmean averaged over these five datasets. The vertical axis gave the performance metric Gmean (%), and mGmean (%) in the last clustered columns. PCBERTA1 and PCBERTA2 represented the PCBERTA framework based on the first one and two BERT layers, respectively. BERTA1 and BERTA2 were the BERTA framework based on the first one and two BERT layers, respectively.

demonstrated in Fig. 5.

4.3. BERTA: PCA-based enrichment of the BERT layer-encoded features

The latent features encoded by the BERT layers exhibited significant sparsity (with most values close to 0), and this study used PCA to enrich the BERT layer-encoded features to the PCs explaining for 95 % variance. The baseline PCBERTA framework used class-specific feature enrichment PCA models (Fig. 1 (d)) for the downstream tasks.

A comparison between Figs. 5 and 6 suggested that the PCA-based enrichment did not significantly change the prediction performance of the BERT layer-encoded features on the five downstream tasks. Fig. 6 demonstrated the same trend that the latent features encoded by the first two layers achieved the two best mean Gmean (%) values across the five prediction tasks. Therefore, PCA effectively enriched the latent features encoded by the individual BERT layers while maintained similar performance levels of the five downstream tasks.

4.4. Selecting the number of BERT layers

The metric mGmean reached the largest value 89.84 % at $k = 2$, while the first layer ($k = 1$) delivered a slightly worse mGmean = 89.70 % (Fig. 7). If we took the training time cost into consideration, the metric BF also reached the largest value BF = 19.94 at $k = 2$. The standard deviation value reached the smallest value 4.34 at $k = 2$ while the next two best values were 4.87 ($k = 1$) and 4.63 ($k = 3$). Based on these data, this study used the first two BERT layers for the subsequent experiments.

4.5. Comparison of PCBERTA and BERTA

PCBERTA utilized the class-specific PCA enrichment, instead of the class-independent PCA engineering in the BERTA framework. Fig. 8 compared the prediction performance of the PCBERTA and BERTA frameworks based on the first two BERT layers. The PCA transformed matrix of the first two BERT layers generated two layers of PCs, which were independent to each other. The two layers of the PCs in the PCBERTA framework were denoted as PCBERTA1 and PCBERTA2, respectively. Those in the BERTA framework were noted as BERTA1 and BERTA2, respectively.

Fig. 8 showed that PCBERTA2 achieved the best mGmean value 83.03 % averaged over the five prediction datasets. BERTA2 (82.26 %) also outperformed BERTA1 (81.61 %) in the metric mGmean. PCBERTA1 and PCBERTA2 achieved the best Gmean values on three of the five datasets, while BERTA1 and BERTA2 together only achieved twice the best Gmean values. Overall, PCBERTA outperformed the BERTA framework.

4.6. Contributions of the HC features

This study evaluated the contributions of the five types of the HC features from (Y. Pang, et al., 2021) to the BERTA and PCBERTA features (Fig. 9). Pang et al., also used the statistical *t*-test to select a subset of these HC features for the ACVP prediction task (Y. Pang, et al., 2021), and this subset of features were denoted as "HC + Ttest" in Fig. 9.

Fig. 9 (a) showed that the HC features improved the mGmean values of all the four models, i.e., PCBERTA1, PCBERTA2, BERTA1 and BERTA2. The largest improvement in mGmean 5.21 was achieved for the BERTA1 features, and all the five prediction tasks based on the BERTA1 features were improved. The second largest improvement in mGmean 5.18 was achieved for the PCBERTA1 features, although the prediction task All-AMP based on the PCBERTA1 features was slightly worsen by 2.23 in Gmean (%). Therefore, the HC features positively contributed to the ACVP prediction tasks based on the PCBERTA and BERTA represented features, particularly PCBERTA1 and BERTA1.

We further compared the absolute values of the metric Gmean of the HC features and their concatenations with the PCBERTA and BERTA features (Fig. 9 (b)). PCBERTA1 + HC and BERTA1 + HC achieved the best two mGmean values 85.94 % and 86.82 %, both were better than the HC (85.57 %) and HC + Ttest (84.13 %) models. The PCBERTA1 + HC model achieved the best Gmean values on two ACVP prediction tasks (Anti-Virus and non-AVP), while the BERTA1 + HC model achieved the best Gmean values on another two tasks (All-Neg and All-AMP). The HC + Ttest model achieved the best Gmean value on the non-AMP prediction task, but its performances were much worse than the other models on the other ACVP prediction tasks.

4.7. TaiChiNet features further improve the ACVP prediction tasks

The PCA transformed matrix of the first two BERT layers generated two layers of PCs, which were independent to each other. The two layers of the PCs in the TaiChiNet framework were denoted as TaiChiNet1 and TaiChiNet2, respectively.

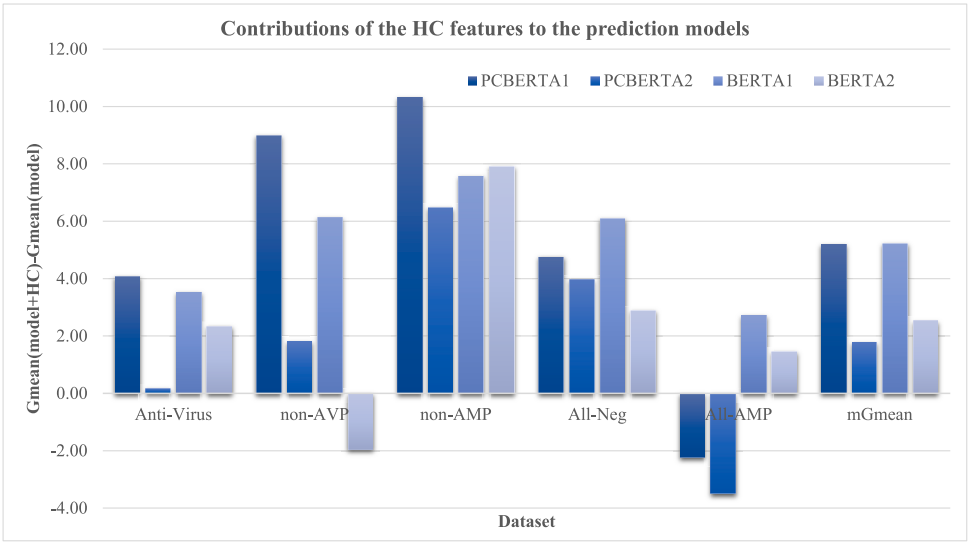
The TaiChiNet features alone did not achieve good prediction performance on the ACVP prediction task, but the integration of the HC features substantially improved the five ACVP prediction datasets based on the TaiChiNet features alone (Fig. 10). TaiChiNet1 + HC achieved the best mGmean value 87.27 %, and outperformed all the five ACVP prediction datasets based on both the HC and the HC + Ttest features from the study PreAntiCoV (Y. Pang, et al., 2021). Therefore, the following sections referred to TaiChiNet1 as the default TaiChiNet framework, and the final best model TaiChiNet1 + HC as TaiChiACVP.

4.8. TaiChiNet is the most important feature type

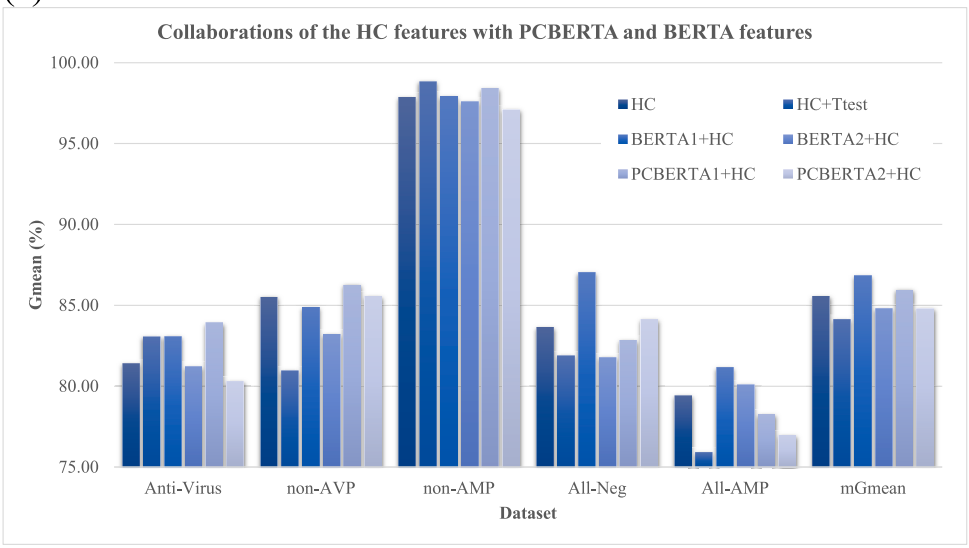
The averaged feature importance columns in Fig. 11 showed that TaiChiNet was the most important feature type in the TaiChiACVP model. The TaiChiNet features achieved at least 38.87 % improvement in the feature importance than the other five feature types. The averaged feature importance of TaiChiNet was even 14.1990 times that of the DiC feature type. The TaiChiNet feature type alone achieved the largest feature importances on the three ACVP prediction datasets, including All-Neg, non-AMP, and Anti-Virus. The PHYC feature type achieved the largest feature importance on the other two prediction tasks. For the details of the feature importance of each features in the 5 ACVP datasets, please see Supplementary Table S2.

4.9. The physicochemical meaning of TaiChiNet features

We further investigated the explainability of the TaiChiNet features (Fig. 12) using the approach in (Renaud & Mansbach, 2023). Renaud et al., calculated the bridge variables to describe the physicochemical



(a)



(b)

Fig. 9. Contributions of the HC features to the BERTA and PCBERTA features. The horizontal axis gave the five prediction datasets, and the metric mGmean averaged over the five datasets. The vertical axis gave the Gmean (%). (a) The vertical axis of this sub-figure gave the difference between the Gmean (%) values of (model + HC) and (model), where model was one of the four feature representation models, PCBERTA, PCBERTA2, BERTA1 and BERTA2. (b) The vertical axis of this sub-figure gave the Gmean (%) values.

properties of a peptide, and evaluated the correlations of the latent features with these bridge variables. There were 18 bridge variables used in this paper, and seven of them were aliphatic index, Boman index, isoelectric point, charge (discretized as pH = 3, pH = 7 and pH = 11), hydrophobicity, instability index, and molecular weight. They could be calculated using the python package peptides (Renaud & Mansbach, 2023). The other 11 bridge variables were derived from (Huang, et al., 2023), including positive charge, negative charge, charge of all, polar number, non-polar number, pH number and Van der Waals volume (vdW_volume). The absolute values of the PCCs were used to show whether there were TaiChiNet features strongly correlated with the bridge variables. The results were shown in Fig. 12.

Generally, all the 11 bridge variables had the TaiChiNet features with the |PCC| values at least 0.1905 across the five ACVP prediction datasets (Fig. 12), which was the |PCC| value of vdW_volume with TaiChiNet-1d on the All-Neg dataset. And the 0th dimension of the TaiChiNet features learned a latent feature strongly correlated with the molecular weight (|PCC|≥0.9867) across all the five ACVP prediction datasets. And the 0th dimension of the TaiChiNet features stayed at the top 30 importance features among all of the 5 ACVP tasks (see Supplementary Table S2). Specifically, the other bridge variables could also be explained by some TaiChiNet features, like the 3rd dimension of TaiChiNet features was highly correlated to "charge of all" property on non-AMP dataset with |PCC|=0.7912. The 4th dimension of TaiChiNet features was highly correlated to "net charge (pH = 7)" property on All-Neg dataset with |PCC|=0.7171.

However, not all TaiChiNet features can be explained by these properties, such as the 1st dimension and the 5th dimension of TaiChiNet features only have the maximum |PCC| value 0.3238 and 0.3796 with the bridge variables, respectively.

4.10. *Insights into TaiChiNet: Principles and visualization*

In order to figure out what TaiChiNet has learned, we dig deeply into the mathematical meaning of TaiChiNet and PCBERTA. The comparison shows that both of them do not change the relative distance among the

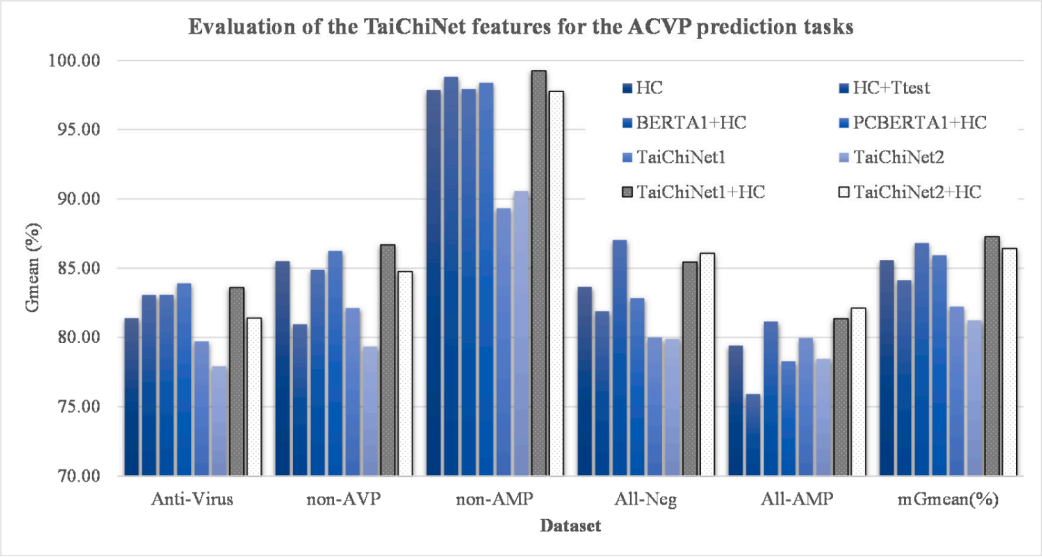
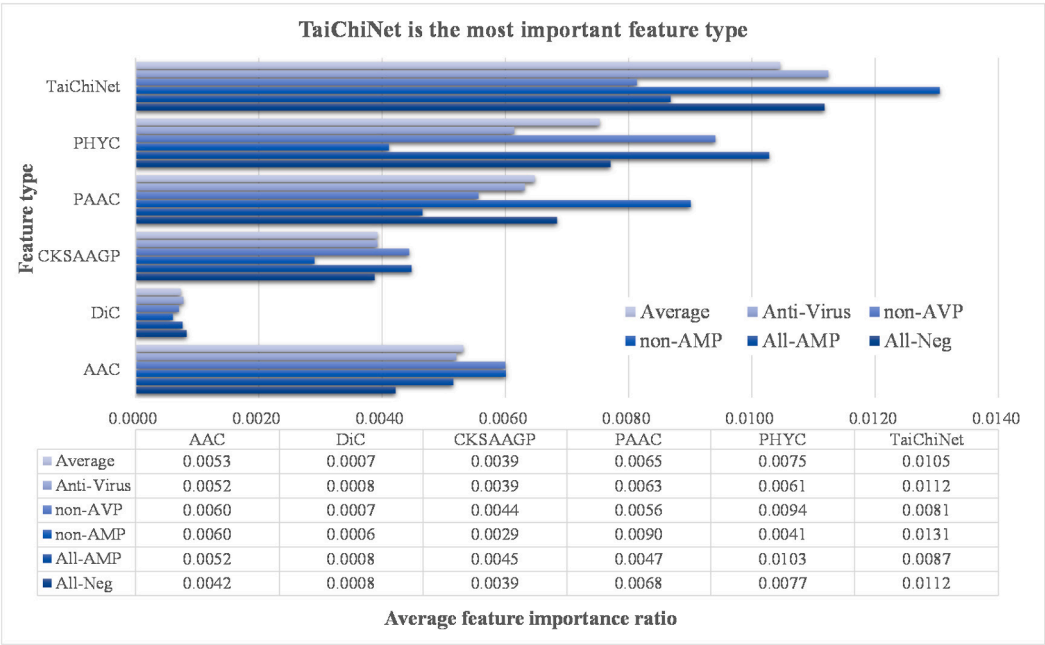


Fig. 10. Evaluation of the TaiChiNet features on the ACVP prediction task. The horizontal axis gave the five prediction datasets, and the metric mGmean averaged over the five datasets. The vertical axis gave the Gmean (%) values.



	AAC	DiC	CKSAAGP	PAAC	PHYC	TaiChiNet
Average	0.0053	0.0007	0.0039	0.0065	0.0075	0.0105
Anti-Virus	0.0052	0.0008	0.0039	0.0063	0.0061	0.0112
non-AVP	0.0060	0.0007	0.0044	0.0056	0.0094	0.0081
non-AMP	0.0060	0.0006	0.0029	0.0090	0.0041	0.0131
All-AMP	0.0052	0.0008	0.0045	0.0047	0.0103	0.0087
All-Neg	0.0042	0.0008	0.0039	0.0068	0.0077	0.0112

Fig. 11. Evaluation of the importance of the six feature types in the final TaiChiACVP model. The six feature types were AAC, DiC, CKSAAGP, PAAC, PHYC, and TaiChiNet. Each feature was measured by its importance coefficient in the TaiChiACVP model, and each feature type was evaluated by the mean importance coefficient of the features in this feature type of an ACVP prediction dataset. The "Average" column series was the averaged values of the six feature types over the five ACVP prediction datasets.

dimensions of BERT features before concatenation, and the only difference between them is the rotation degree. The impact of different rotation degrees emerges after the concatenating operation. As is shown in Fig. 13.

4.10.1. The only difference between a pair of subtractive features of TaiChiNet and PCBERTA is the degree of rotation

The most crucial step in the TaiChiNet framework is step 7 in Algorithm 2. It is worth of notion that the intra-layer based PCA process is the same among BERTA, PCBERTA and TaiChiNet, and this subsection only considers the inter-layer PCA-based fusion matrix calculation. The mathematical principle of PCA involves decomposing the covariance matrix of the features through Singular Value Decomposition (SVD), and then transforming it into scaling factors composed of eigenvalues $\boldsymbol{\Lambda}$ and an orthogonal rotation matrix of eigenvectors $\boldsymbol{Q}$, after the data centralization. This work uses the Python sklearn package to calculate PCA with default parameters. The parameter *whiten* is *False* by default, meaning that the principal components are all unit vectors without $\boldsymbol{\Lambda}$. Therefore, this work uses two steps to calculate PCA: centralization and rotation.

To consider the centralization for PCA transformation, let $\boldsymbol{C}^{+} = [\boldsymbol{c}_1^{+}, \boldsymbol{c}_2^{+}]$ and $\boldsymbol{C}^{-} = [\boldsymbol{c}_1^{-}, \boldsymbol{c}_2^{-}]$, where $\boldsymbol{C}^{+}, \boldsymbol{C}^{-} \in \mathscr{R}^{n_{embedding} \times 2}$, be the mean values of the two BERT layers of positive samples in training set $\boldsymbol{D}^{+}$ and negative samples in training set $\boldsymbol{D}^{-}$, respectively. Given a sample $\boldsymbol{d} = [\boldsymbol{L}_1 \quad \boldsymbol{L}_2]$,

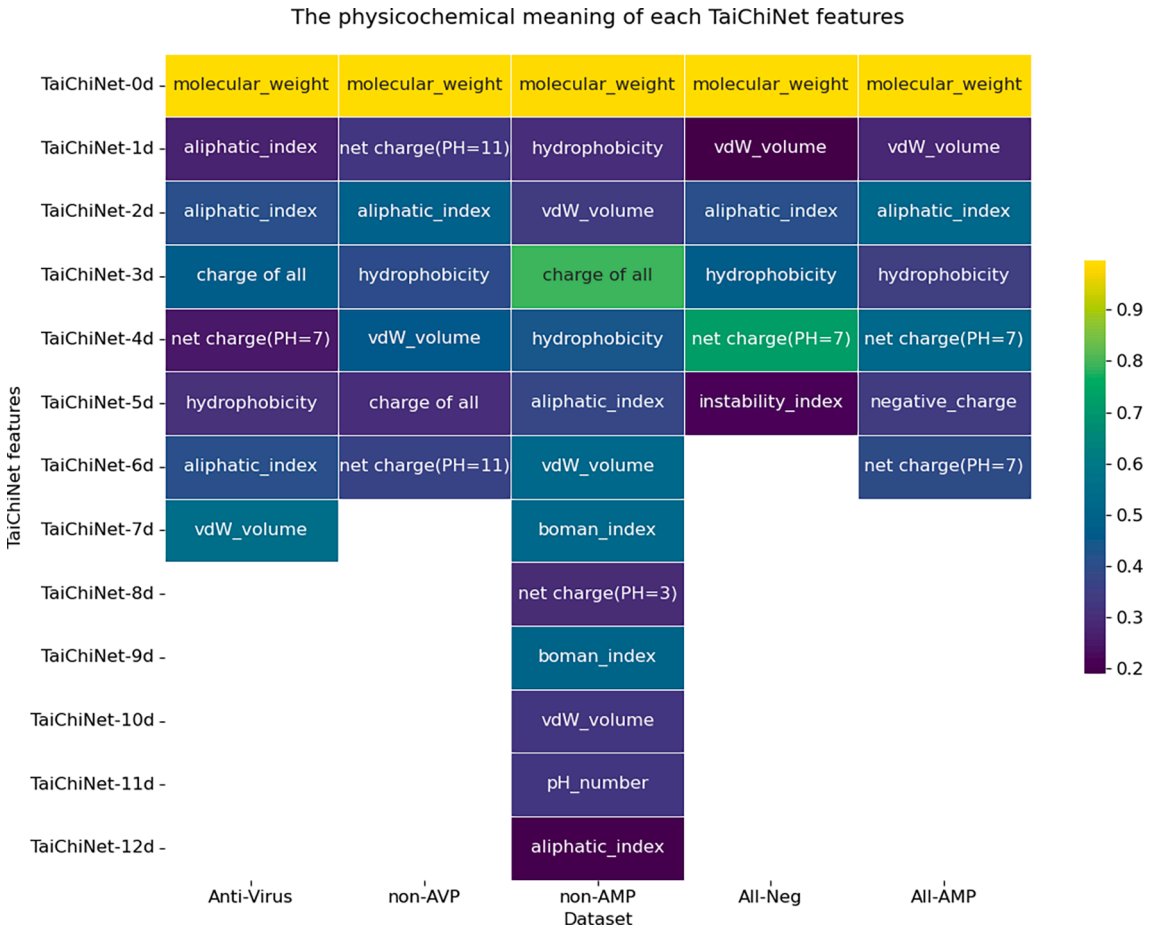


Fig. 12. Explainability of the TaiChiNet features. The vertical axis lists the TaiChiNet features, and the horizontal axis gives the list the datasets. The color illustrates the maximum absolute Pearson correlation coefficient (PCC) between a TaiChiNet feature and the corresponding bridge variable in a TaiChiACVP model. And the text showed the certain physicochemical properties of the corresponding values.

where $\boldsymbol{L}_1, \boldsymbol{L}_2 \in \mathscr{P}^{n_{embedding} \times 1}$, we have two types of TaiChiNet features $\boldsymbol{T}^+(\boldsymbol{d})$ and $\boldsymbol{T}^-(\boldsymbol{d})$ before concatenation. Define $\left[\frac{*}{*}\right]$ as the concatenation operation. We finally have the TaiChiNet features $\boldsymbol{T}(\boldsymbol{d}) = \left[\frac{\boldsymbol{T}^+(\boldsymbol{d})}{\boldsymbol{T}^-(\boldsymbol{d})}\right]$:

$$\boldsymbol{T}^+(\boldsymbol{d}) = \left[(\boldsymbol{d} - \boldsymbol{C}^+)\boldsymbol{PCA}_{\boldsymbol{k}}^{+T} - \boldsymbol{C}^-\right]\boldsymbol{PCA}_{\boldsymbol{k}}^{-T}$$

$$= (\boldsymbol{d} - \boldsymbol{C}^+)PCA_{\boldsymbol{k}}^{+T}PCA_{\boldsymbol{k}}^{-T} - \boldsymbol{C}^-PCA_{\boldsymbol{k}}^{-T} \quad (2)$$

$$\boldsymbol{T}^-(\boldsymbol{d}) = \left[(\boldsymbol{d} - \boldsymbol{C}^-)\boldsymbol{PCA}_{\boldsymbol{k}}^{-T} - \boldsymbol{C}^+\right]\boldsymbol{PCA}_{\boldsymbol{k}}^{+T}$$

$$= (d - \boldsymbol{C}^-)PCA_{\boldsymbol{k}}^{-T}PCA_{\boldsymbol{k}}^{+T} - \boldsymbol{C}^+PCA_{\boldsymbol{k}}^{+T} \quad (3)$$

$$T(\boldsymbol{d}) = \left[\frac{(\boldsymbol{d} - \boldsymbol{C}^+)\boldsymbol{PCA}_{\boldsymbol{k}}^{+T}\boldsymbol{PCA}_{\boldsymbol{k}}^{-T} - \boldsymbol{C}^-\boldsymbol{PCA}_{\boldsymbol{k}}^{-T}}{(\boldsymbol{d} - \boldsymbol{C}^-)\boldsymbol{PCA}_{\boldsymbol{k}}^{-T}\boldsymbol{PCA}_{\boldsymbol{k}}^{+T} - \boldsymbol{C}^+\boldsymbol{PCA}_{\boldsymbol{k}}^{+T}}\right] \quad (4)$$

The dimensions of both $\boldsymbol{PCA}_{\boldsymbol{k}}^{+}$ and $\boldsymbol{PCA}_{\boldsymbol{k}}^{-}$ within TaiChiNet are $R^{2\times2}$. Let us denote:

$$PCA_{\boldsymbol{k}}^{+T} = \boldsymbol{Q}^{+T} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \quad (5)$$

$$PCA_{\boldsymbol{k}}^{-T} = \boldsymbol{Q}^{-T} = \begin{bmatrix} cos\beta & sin\beta \\ -sin\beta & cos\beta \end{bmatrix} \quad (6)$$

The primary procedures of step 7 in TaiChiNet Algorithm 2 involves the multiplication of the two positive and negative layer fusion transformation matrices $\boldsymbol{PCA}_{\boldsymbol{k}}^{+T}$ and $\boldsymbol{PCA}_{\boldsymbol{k}}^{-T}$ which is equal to:

$$\boldsymbol{PCA}_{\boldsymbol{k}}^{+T}\boldsymbol{PCA}_{\boldsymbol{k}}^{-T} = \boldsymbol{Q}^{+T}\boldsymbol{Q}^{-T} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}\begin{bmatrix} cos\beta & sin\beta \\ -sin\beta & cos\beta \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta+\beta) & \sin(\theta+\beta) \\ -\sin(\theta+\beta) & \cos(\theta+\beta) \end{bmatrix} \quad (7)$$

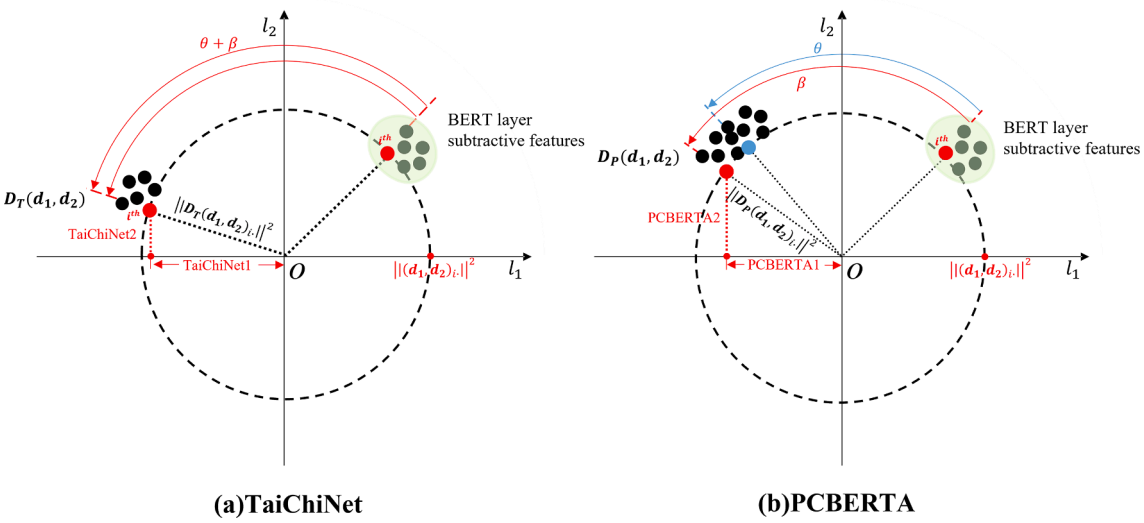
Therefore, $\boldsymbol{PCA}_{\boldsymbol{k}}^{+T}\boldsymbol{PCA}_{\boldsymbol{k}}^{-T} = \boldsymbol{PCA}_{\boldsymbol{k}}^{-T}\boldsymbol{PCA}_{\boldsymbol{k}}^{+T}$, both of which are orthogonal matrices with the same rotation degree $\theta + \beta$. According to the equations (2) and (3), we can see that TaiChiNet firstly centralizes the input data $\boldsymbol{d}$ and rotates the samples with $\theta + \beta$ degrees, and then minus a rotated negative/positive mean value. Additionally, if we mathematically compare PCBERTA with TaiChiNet, we have two types of PCBERTA features before concatenation $\boldsymbol{P}^+(\boldsymbol{d})$ and $\boldsymbol{P}^-(\boldsymbol{d})$. We finally have $\boldsymbol{P}(\boldsymbol{d}) = \left[\frac{\boldsymbol{P}^+(\boldsymbol{d})}{\boldsymbol{P}^-(\boldsymbol{d})}\right]$:

$$\boldsymbol{P}^+(\boldsymbol{d}) = (\boldsymbol{d} - \boldsymbol{C}^+)PCA_{\boldsymbol{k}}^{+T} = dPCA_{\boldsymbol{k}}^{+T} - \boldsymbol{C}^+PCA_{\boldsymbol{k}}^{+T} \quad (8)$$

$$\boldsymbol{P}^-(\boldsymbol{d}) = (\boldsymbol{d} - \boldsymbol{C}^-)PCA_{\boldsymbol{k}}^{-T} = dPCA_{\boldsymbol{k}}^{-T} - \boldsymbol{C}^-PCA_{\boldsymbol{k}}^{-T} \quad (9)$$

$$P(\boldsymbol{d}) = \left[\frac{(\boldsymbol{d} - \boldsymbol{C}^+)\boldsymbol{PCA}_{\boldsymbol{k}}^{+T}}{(\boldsymbol{d} - \boldsymbol{C}^-)\boldsymbol{PCA}_{\boldsymbol{k}}^{-T}}\right] \quad (10)$$

In order to clarify the difference between PCBERTA and TaiChiNet, it would be better to measure the distance between 2 arbitrary samples in



(a)TaiChiNet (b)PCBERTA

Fig. 13. An exemplified visualization of the difference between TaiChiNet and PCBERTA in the subtractive layer feature space. For any pair of samples $\boldsymbol{d}_1$and$\boldsymbol{d}_2(\boldsymbol{d}_1 \neq \boldsymbol{d}_2)$, we denote the subtraction of their TaiChiNet features $\boldsymbol{D_T}(\boldsymbol{d}_1,\boldsymbol{d}_2)$ and PCBERTA features $\boldsymbol{D_P}(\boldsymbol{d}_1,\boldsymbol{d}_2)$. The x-axis and the y-axis represent the first and second dimensions of a sample $\boldsymbol{d}$. The green circle highlights the subtraction of their original BERT layer features $\boldsymbol{d}_1-\boldsymbol{d}_2$. The arrows denote the rotation directions. The red dot on the right x-axis highlights the distance of any dimension i in the subtraction of the original BERT layer features $\boldsymbol{d}_1-\boldsymbol{d}_2$ from the origin. The dash circle represents the identical distance from the origin. (a) The procedure of TaiChiNet. The red point is an example of any dimension i in $\boldsymbol{D_T}(\boldsymbol{d}_1,\boldsymbol{d}_2)$, namely $\boldsymbol{D_T}(\boldsymbol{d}_1,\boldsymbol{d}_2)_i$. TaiChiNet just rotates the subtractive features and doesn't change the relative distance among the dimensions before concatenation. Due to the symmetric patterns of $\boldsymbol{PCA_k^{+T}PCA_k^{-T}}$ and $\boldsymbol{PCA_k^{-T}PCA_k^{+T}}$, the rotation degree of two types of TaiChiNet features are the same, which results in the inter distance differences between the two concatenated subtraction layer features of TaiChiNet and PCBERTA. (b) The procedure of PCBERTA. The red and blue points are an example of the rotated subtractive features of any dimension i in $\boldsymbol{D_P}(\boldsymbol{d}_1,\boldsymbol{d}_2)$, namely $\boldsymbol{D_P}(\boldsymbol{d}_1,\boldsymbol{d}_2)_i$. Their rotation degrees are different by positive and negative rotation matrices. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

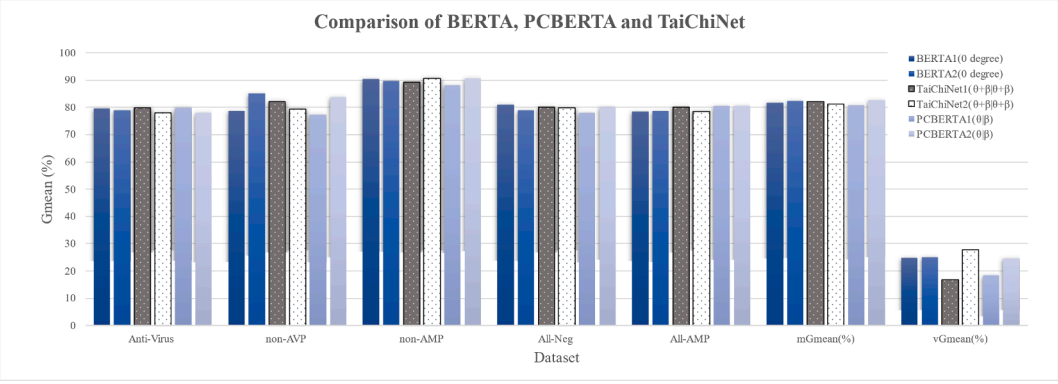


Fig. 14. . The performances of berta, pcberta and taichinet. The horizontal axis gave the five prediction datasets, the metric mGmean averaged over the five datasets, and the metric vGmean is the variance of Gmean among 5 datasets. The vertical axis gave the Gmean (%) values. The rotation degree of their layer features was annotated in (*|*), where "|" represents the concatenation operation. BERTA have no rotation and concatenation operations, so it was annotated with "(0 degree)". TaiChiNet and PCBERTA both have their own rotation degrees and was annotated by "(θ + β|θ + β)" and"(θ|β)" respectively.

the representation space. However, the dimension of the BERT features is high and the definition of distance in the high dimension is tricky, and directly measuring this distance is difficult. Instead, we consider the distance of each dimension of the two BERT layer features from the origin. The Euclidean distance of each dimension of the subtractive features between a pair of samples is calculated as the subtractive layer feature space. We get the subtraction of the TaiChiNet features $\boldsymbol{D_T}(\boldsymbol{d}_1,\boldsymbol{d}_2)$ and PCBERTA features $\boldsymbol{D_P}(\boldsymbol{d}_1,\boldsymbol{d}_2)$ of any two samples $\boldsymbol{d}_1$and$\boldsymbol{d}_2(\boldsymbol{d}_1 \neq \boldsymbol{d}_2)$ as:

$$\boldsymbol{D_P}(\boldsymbol{d}_1,\boldsymbol{d}_2) = P(\boldsymbol{d}_1) - P(\boldsymbol{d}_2) = \begin{bmatrix} (\boldsymbol{d}_1-\boldsymbol{d}_2)\boldsymbol{PCA_k^{+T}} \\ (\boldsymbol{d}_1-\boldsymbol{d}_2)\boldsymbol{PCA_k^{-T}} \end{bmatrix} \tag{11}$$

$$\boldsymbol{D_T}(\boldsymbol{d}_1,\boldsymbol{d}_2) = T(\boldsymbol{d}_1) - T(\boldsymbol{d}_2) = \begin{bmatrix} (\boldsymbol{d}_1-\boldsymbol{d}_2)\boldsymbol{PCA_k^{+T}}\boldsymbol{PCA_k^{-T}} \\ (\boldsymbol{d}_1-\boldsymbol{d}_2)\boldsymbol{PCA_k^{-T}}\boldsymbol{PCA_k^{+T}} \end{bmatrix} \tag{12}$$

For any dimension i of $\boldsymbol{D_T}(\boldsymbol{d}_1,\boldsymbol{d}_2)$ we have the distance from the origin:

$$\left|\left|\boldsymbol{D_T}(\boldsymbol{d}_1,\boldsymbol{d}_2)_{i\bullet}\right|\right|^2 = \left|\left|(\boldsymbol{d}_1-\boldsymbol{d}_2)_{i\bullet}\boldsymbol{PCA_k^{+T}}\boldsymbol{PCA_k^{-T}}\right|\right|^2 = \left|\left|(\boldsymbol{d}_1-\boldsymbol{d}_2)_{i\bullet}\right|\right|^2 \tag{13}$$

As well as for the distance of any dimension i of $\boldsymbol{D_P}(\boldsymbol{d}_1,\boldsymbol{d}_2)$ from the origin:

$$\left|\left|\boldsymbol{D_P}(\boldsymbol{d}_1,\boldsymbol{d}_2)_{i\bullet}\right|\right|^2 = \begin{cases} \left|\left|(\boldsymbol{d}_1-\boldsymbol{d}_2)_{i\bullet}\boldsymbol{PCA_k^{+T}}\right|\right|^2 \\ \left|\left|(\boldsymbol{d}_1-\boldsymbol{d}_2)_{i\bullet}\boldsymbol{PCA_k^{-T}}\right|\right|^2 \end{cases} = \left|\left|(\boldsymbol{d}_1-\boldsymbol{d}_2)_{i\bullet}\right|\right|^2 \tag{14}$$

Equations (13) and (14) suggest that TaiChiNet just rotates the subtractive features between a pair of samples and doesn't change the relative distance among the dimensions before concatenation. So it does

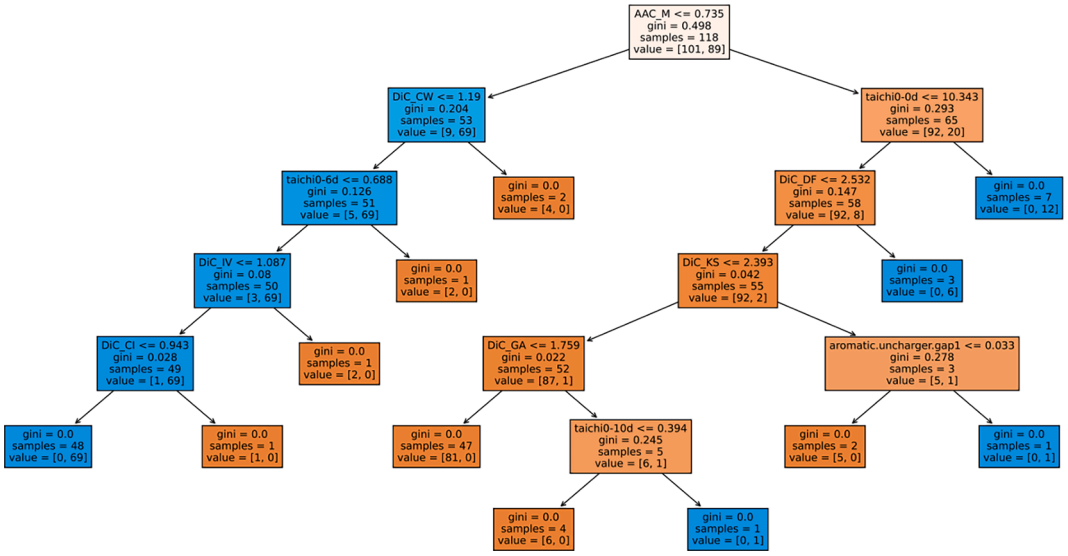


Fig. 15. An example of a single decision tree in the non-AMP prediction task of TaiChiNet with HC features. "taichi0-nd" denoted as the nth dimension of the TaiChiNet1, "AAC_X" denoted as the AAC features of amino acid X, "DiC_XX" denoted as the DiC features of the amino acids pair "XX".

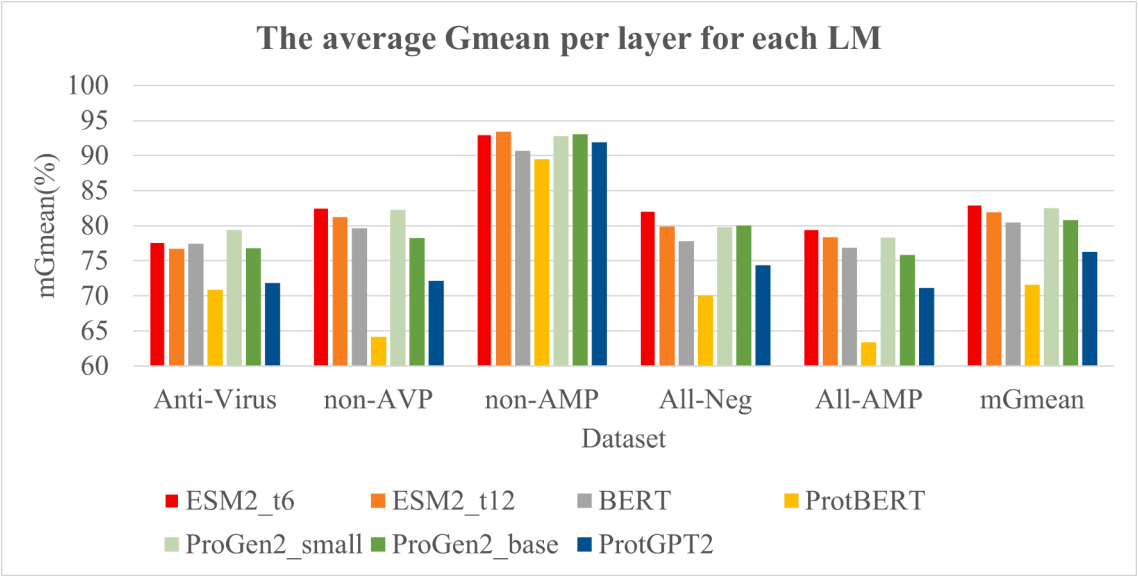


Fig. 16. The average Gmean of different LMs across layers on the test dataset of 5 ACVP tasks. The vertical axis shows the average Gmean value of different LMs across all layers. The horizontal axis shows the 5 ACVP tasks, and calculates the mean Gmean value (mGmean) across the 5 tasks for each LM. ESM2_t6 and ESM2_t12 are short for ESM2 with 6 layers and 12 layers, respectively. BERT is the English LM BERT we have utilized in this paper. ProtBERT denotes ProtBERT. ProGen2_small and ProGen 2_base are the small and base versions of ProGen2, respectively. And the last model ProtGPT2 represents ProtGPT2.

the same processing to PCBERTA. However, the symmetric patterns of $\boldsymbol{PCA_k^{+T}PCA_k^{-T}}$ and $\boldsymbol{PCA_k^{-T}PCA_k^{+T}}$ make the rotation degree of two type of TaiChiNet subtractive layer features the same (as illustrated in Fig. 13 (a)). PCBERTA shows a slightly different rotation degree. This results in the inter-distance difference between the two concatenated subtraction layer features of TaiChiNet and PCBERTA. Fig. 13 intuitively illustrates the process of the former deduction.

4.10.2. Advantages of TaiChiNet

BERTA, PCBERTA and TaiChiNet use the same mathematical intra-layer fusion strategy with PCA, whereas PCBERTA and TaiChiNet further utilize inter-layer fusion strategy with PCA and extend the dimension with concatenation operation. In order to figure out the advantages of TaiChiNet, we firstly compare the performance of TaiChiNet with PCBERTA and BERTA, as shown in Fig. 14. The rotation degree of their layer features is annotated in (*|*), where "|" represents the concatenation operation. BERTA has no rotation and concatenation operations. So it is annotated with "(0 degree)". TaiChiNet and PCBERTA both have their own rotation degrees and are annotated by "$(\theta + \beta|\theta + \beta)$" and"$(\theta|\beta)$", respectively.

The experimental data in the previous sections show that different rotation degrees influence the downstream predictive performances. Besides, PCBERTA2 performs the best in the mGmean 82.58 % due to the concatenation of different rotation degrees, as deduced in the previous subsection (Fig. 13 (b)). TaiChiNet1 has the most stable performance among the datasets with the lowest vGmean 16.69 %, which is 8.24 % lower than BERTA1 and 8.25 % lower than BERTA2, 1.78 % lower than PCBERTA1 and 7.87 % lower than PCBERTA2. TaiChiNet1 also performs 11.10 % lower than TaiChiNet2. This means TaiChiNet1 captures a generally informative peptide representation that can distinguish

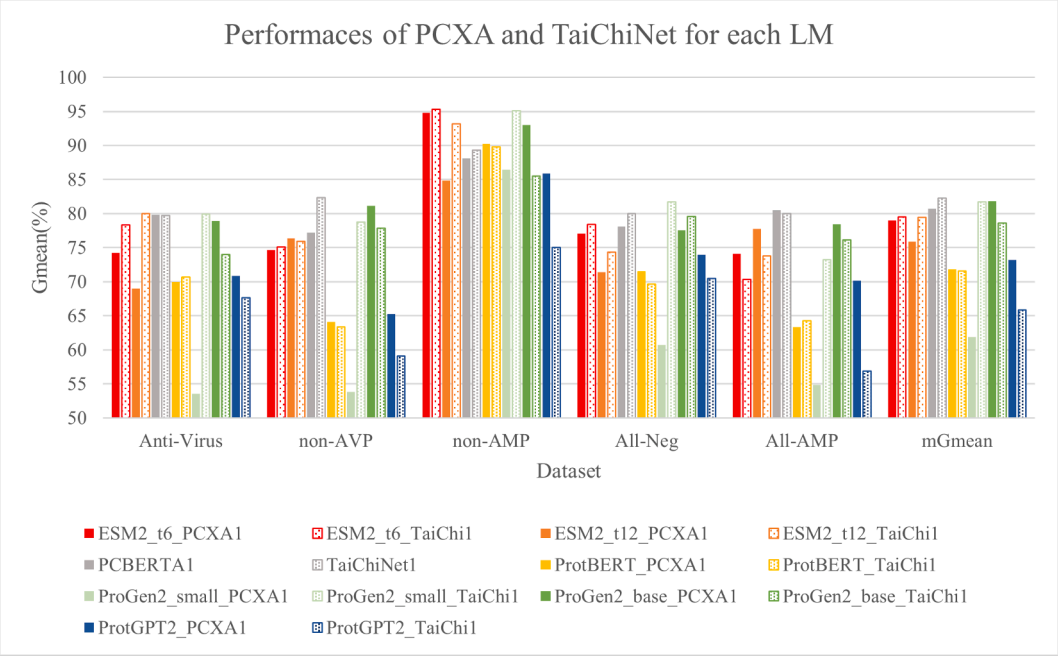


Fig. 17. The performances of the first two layer of PCXA and TaiChiNet for each LM. The x-axis gives the prediction tasks, and the y-axis gvies the Gmean value.

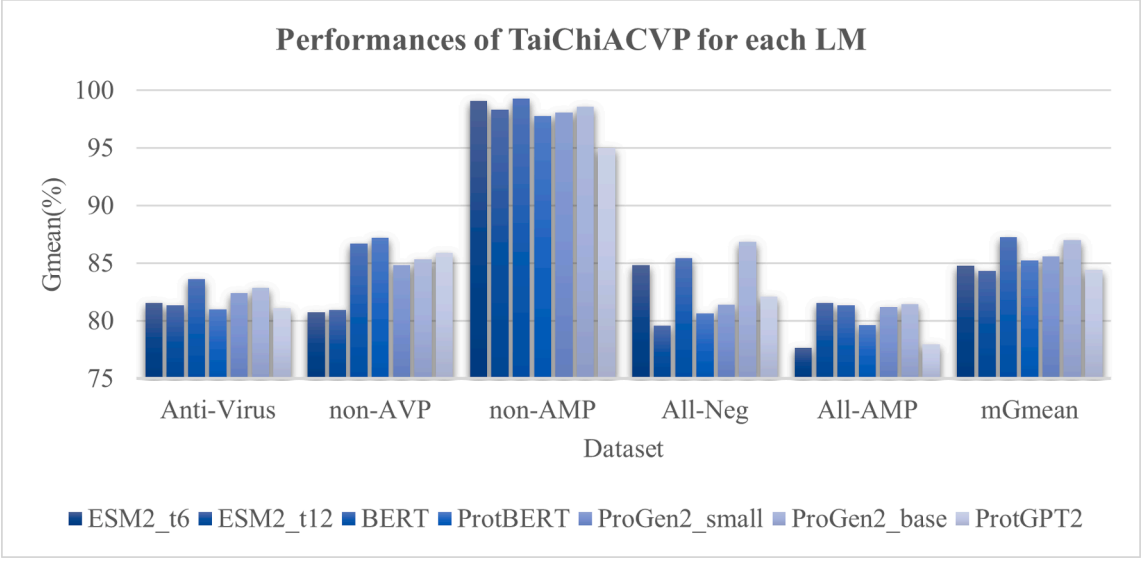


Fig. 18. The performances of TaiChiACVP for each LM. The x-axis is the dataset. The y-axis is the Gmean value. mGmean denotes the average og Gmean among 5 ACVP tasks.

between the positive and negative samples in the 5 datasets, rather than leaning towards any bias of the individual data set. We also observe that TaiChiNet1 generally perform better than PCBERTA1 and BERTA1.

A combination of TaiChiNet1 and HC features achieves the best performance among the 5 ACVP tasks (Fig. 10). So we visualize the trees to observe the decision paths of the random forest classifier (Fig. 15). It turns out that the decision-making process of no single tree was solely based on the TaiChiNet features, and the decision path of each tree utilizes both TaiChiNet and HC features. Fig. 15 shows an example decision tree in the non-AMP prediction task. "taichi0-nd" denotes the n^{th} dimension of the TaiChiNet1 features, "AAC_X" denotes the AAC features of amino acid X, "DiC_XX" denotes the DiC features of the amino acids pair "XX". More details may be found in the code plot_interpretation.py in our publicly available source code at https://www.healthinformaticslab.org/supp/resources.php. This observation suggests that the TaiChiNet features have learned useful information to compensate the HC features for the ACVP prediction tasks.

4.11. Evaluation of other language models

4.11.1. Performances of each layer of LMs

We continue to evaluate the performance of different pre-trained language models. Two Transformer encoder-based masking language models are tested, including ESM2 (Lin, et al., 2023) with 6 layers (ESM2_t6), 12 layers (ESM2_t12) and ProtBERT (Elnaggar, et al., 2022). Another two Transformer decoder-based autoregressive models are evaluated, including ProGen2 (Nijkamp, et al., 2023) small version (ProGen2_small), base version (ProGen2_base) and ProtGPT2 (Ferruz,

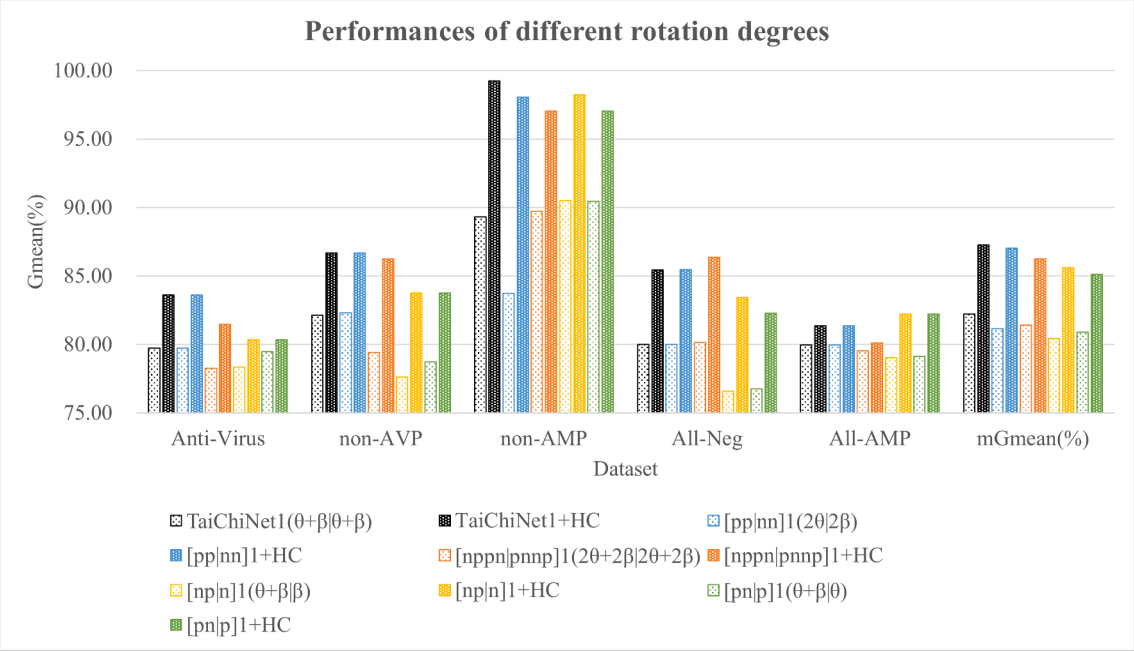


Fig. 19. Performances of different rotation degrees. "pp + nn" is denoted as "2θ|2β", "nppn + pnnp" is denoted as "2(θ + β)|2(θ + β)", "pn + p" is represented by "(θ + β|θ)", "np + n" is referred by "(θ + β|β)".

et al., 2022). The results are summarized in Fig. 16, and the detailed data of each layer in an LM may be found in Supplementary Fig. S1.

It turns out that ProtBERT performs the worst one among 7 LMs on each ACVP tasks. It might be due to that ProtBERT didn't consider the effect of data partition during training. Biological sequences have an inherent tendency to cluster together (Suzek, et al., 2007), hence the training procedure of a pre-trained LM needs to carefully design the dataset splitting strategy to avoid bias. This is an essential difference between natural languages and biological sequences (Rives, et al., 2021). The second worst model is ProtGPT2, which may have the same issue during training. ESM2 with 6 layers generally outperforms the one with 12 layers, and achieves the best performance on average (with mGmean 82.94 %). This observation could be attributed to that the scaling law pattern of ESM2 collapses on peptide sequences, which is also consistent with the literature (Fernandez-Diaz et al., 2023). We hypothesize that the dataset splitting strategy MMseqs2 is not as suitable on protein sequences as it is on peptides (Hauser, Steinegger, & Söding, 2016; Teufel, et al., 2023; "UniRef|UniProt help," 2024). Furthermore, ProGen2 small version generally outperforms its base version, which may be caused by the same reason as ESM2.

4.11.2. Performances of TaiChiACVP on different LMs

We also evaluate the prediction performances of our TaiChiACVP framework by replacing the embedded BERT model with the other LMs. We calculate the TaiChiNet and PCBERTA features of the first two layers for each LM. Like the PCBERTA model based on the BERT model, we denote the suffices "PCXA1" and "TaiChi1" for the first layers of the PCBERTA and TaiChiACVP frameworks based on different LM models, respectively. The results are shown in Fig. 17.

Except for ProGen2 base version, ProtBERT and ProtGPT2, TaiChiNet improves all of the other 4 LMs on the ACVP prediction tasks. Specifically, ProGen2 small version benefits a lot by TaiChiNet, while ESM2 with 6 layers and 12 layers also benefit from TaiChiNet except on the non-AVP and All-AMP tasks. These results tell us that the rotation degree of the layer features can affect the performance of LMs with RF classifiers. It is noteworthy that tree-based classifiers treat input features individually, whereas deep learning models mix the input features (Xia, et al., 2024).

Then we combine HC features with TaiChiNet features based on each LM, and the results are shown in Fig. 18. It turns out that BERT model still has the best performance among these models, and ProGen2 base version performs the second best. It is observed again that ESM2 with 6 layers performs better than ESM2 with 12 layers, which could be due to the same reasons. Surprisingly, ProtBERT alone performs the worst, and it outperforms ESM2 with 12 layers in the TaiChiACVP framework.

4.12. Evaluation of different rotation degrees

We test 4 additional combinations of positive and negative transition matrices, including "pp + nn" (rotation degrees with 2θ and 2β, denoted as "2θ|2β"), "nppn + pnnp" (rotation degrees with 2(θ + β) and 2(θ + β), denoted as "2(θ + β)|2(θ + β)"), "pn + p" (rotation degrees with (θ + β) and β, denoted as "(θ + β)|β"), and "np + n" (rotation degrees with (θ + β) and θ, denoted as "(θ + β)|θ"). The results are shown in Fig. 19.

The experimental data shows that "TaiChiNet1 + HC" still performs the best in the metric mGmean, and the default version of TaiChiNet1 features alone outperforms the other four settings of rotation degrees. If the HC features are not utilized, the five prediction tasks have different optimal rotation degrees. The prediction task non-AMP prefers "np + n" with 90.50 % in Gmean, while another task non-AVP prefers "pp + nn" setting with Gmean = 82.32 %. The prediction task All-Neg prefers "nppn + pnnp" with Gmean = 80.13 %. The remaining two prediction tasks Anti-Virus and All-AMP achieve the Gmean values 79.73 % and 79.97 %, respectively.

These results suggest that the rotation degrees of BERT layer features can influence the prediction performances of the downstream tasks. Therefore, finding a way to optimize an optimal rotation degree is necessary for the future work.

5. Discussion

This study proposed a PCA-based fusion network TaiChiNet to enrich the first two BERT layers for peptide representation and ACVP prediction. The experimental data showed that the integration of the HC features and the TaiChiNet features based on the first two BERT layers were both computational efficient and predictively accurate compared with

the whole BERT model.

After mathematically investigating the mechanism of TaiChiNet and PCBERTA in the section 4.10, we figured out that the rotation degrees of the layer features could influence the performances. The rotation degree was learned by PCA, and the angle was fixed after the calculation of the PCA transformation matrix. This fixed learning method limited the capacity of TaiChiNet. Besides, TaiChiNet used the first two layers in this work, and the extension of TaiChiNet to effectively utilize more layers remained to investigate in the future work.

We were surprised to observe that BERT outperformed the other LMs on the ACVP prediction tasks when their TaiChiNet-engineered features were integrated with HC features, while the BERT features alone just performed the 5th best average performance. On the one hand, the experimental data suggested that the existing pre-trained LMs failed to properly represent peptide sequences, since these LMs were trained using the dataset splitting strategy suitable for protein sequences (Hauser, et al., 2016; Teufel, et al., 2023; "UniRef|UniProt help," 2024). The inherent differences between the lengths and structures of peptides and proteins might have caused this discrepancy. Besides, the base version of ProGen2 also gave competitive results on the ACVP prediction tasks. All the experimental data suggested the necessity of future explorations of LMs on effective peptide representations and downstream predictions.

Some of the TaiChiNet features also showed good explainability by their strong correlations with the physicochemical properties of the represented peptides (section 4.9). Clark et al. explained the information learned from each attention head of the English language-based BERT model (Clark, Khandelwal, Levy, & Manning, 2019). Better understanding of an LM learned from peptide sequences will help represent the peptides and their property predictions.

In summary, the novel TaiChiNet network efficiently represented the peptides for the downstream ACVP prediction tasks, and can be easily orchestrated with the other LMs. Its explainability may attract the attentions of both computer scientists and interdisciplinary researchers in the area of artificial intelligence. The future work may explore the utilizations of learnable rotation degrees, multi-layer fusions, and end-to-end training. The TaiChiNet framework may also be employed on the representations of nucleotide sequences like DNA and RNA.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Senior and Junior Technological Innovation Team (20210509055RQ), Guizhou Provincial Science and Technology Projects (ZK2023-297), the Science and Technology Foundation of Health Commission of Guizhou Province (gzwkj2023-565), Science and Technology Project of Education Department of Jilin Province (JJKH20220245KJ and JJKH20220226SK), the National Natural Science Foundation of China (62072212 and U19A2061), the Jilin Provincial Key Laboratory of Big Data Intelligent Computing (20180622002JC), and the Fundamental Research Funds for the Central Universities, JLU.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2025.127786.

Data availability

The source code is freely available at: http://www.healthinformaticslab.org/supp/resources.php.

References

Amanat, F., & Krammer, F. (2020). SARS-CoV-2 vaccines: Status report. *Immunity, 52*, 583–589.

Bin Hafeez, A., Jiang, X., Bergen, P. J., & Zhu, Y. (2021). Antimicrobial peptides: An update on classifications and databases. *International Journal of Molecular Sciences, 22*, 11691.

Chen, D., & Li, Y. (2022). PredMHC: An effective predictor of major histocompatibility complex using mixed features. *Frontiers in Genetics, 13*, Article 875112.

Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). *What does BERT look at? An analysis of BERT's attention* (pp. 276–286). Florence, Italy: Association for Computational Linguistics.

Dee, W. (2022). LMPred: Predicting antimicrobial peptides using pre-trained language models and deep learning. *Bioinformatics Advances, 2*, Article vbac021.

Dong, Y., Dai, T., Wei, Y., Zhang, L., Zheng, M., & Zhou, F. (2020). A systematic review of SARS-CoV-2 vaccine candidates. *Signal Transduction and Targeted Therapy, 5*, 237.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2022). ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*, 7112–7127.

Fernandez-Diaz, R., Cossio-Pérez, R., Agoni, C., Lam, H. T., Lopez, V., & Shields, D. C. (2023). AutoPeptideML: Automated Machine Learning for Building Trustworthy Peptide Bioactivity Predictors. *bioRxiv*, 2023.2011. 2013.566825.

Ferruz, N., Schmidt, S., & Höcker, B. (2022). ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications, 13*, 4348.

Geffrey Hinton, O. V., Jeff Dean. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint*, arXiv:1503.02531.

Gordon, M., Duh, K., & Andrews, N. (2020). Compressing BERT: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th workshop on representation learning for NLP* (pp. 143–155).

Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Hauser, M., Steinegger, M., & Söding, J. (2016). MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics, 32*, 1323–1330.

Huang, J., Xu, Y., Xue, Y., Huang, Y., Li, X., Chen, X., Xu, Y., Zhang, D., Zhang, P., Zhao, J., & Ji, J. (2023). Identification of potent antimicrobial peptides via a machine-learning pipeline that mines the entire space of peptide sequences. *Nature Biomedical Engineering, 7*, 797–810.

Jawahar, G., Sagot, B., & Seddah, D. (2019). *What Does BERT Learn about the Structure of Language? In* (pp. 3651–3657). Florence, Italy: Association for Computational Linguistics.

Jhong, J.-H., Yao, L., Pang, Y., Li, Z., Chung, C.-R., Wang, R., Li, S., Li, W., Luo, M., & Ma, R. (2022). dbAMP 2.0: Updated resource for antimicrobial peptides with an enhanced scanning method for genomic and proteomic data. *Nucleic Acids Research, 50*, D460–D470.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding. *arXiv preprint*, arXiv:1909.10351.

Krammer, F. (2020). SARS-CoV-2 vaccines in development. *Nature, 586*, 516–527.

Kurata, H., Tsukiyama, S., & Manavalan, B. (2022). iACVP: Markedly enhanced identification of anti-coronavirus peptides using a dataset-specific word2vec model. *Briefings in Bioinformatics, 23*, Article bbac265.

Lawrence, T. J., Carper, D. L., Spangler, M. K., Carrell, A. A., Rush, T. A., Minter, S. J., Weston, D. J., & Labbe, J. L. (2021). amPEPpy 1.0: A portable and accurate antimicrobial peptide prediction tool. *Bioinformatics, 37*, 2058–2060.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., & Shmueli, Y. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science, 379*, 1123–1130.

Maher, M. C., Bartha, I., Weaver, S., di Iulio, J., Ferri, E., Soriaga, L., Lempp, F. A., Hie, B. L., Bryson, B., Berger, B., Robertson, D. L., Snell, G., Corti, D., Virgin, H. W., Kosakovsky Pond, S. L., & Telenti, A. (2022). Predicting the mutational drivers of future SARS-CoV-2 variants of concern. *Science Translational Medicine, 14*, Article eabk3445.

Manavalan, B., Basith, S., & Lee, G. (2022). Comparative analysis of machine learning-based approaches for identifying therapeutic peptides targeting SARS-CoV-2. *Briefings in Bioinformatics, 23*, Article bbab412.

Meher, P. K., Dash, S., Sahu, T. K., Satpathy, S., & Pradhan, S. K. (2022). GIpred: A computational tool for prediction of GIGANTEA proteins using machine learning algorithm. *Physiology and Molecular Biology of Plants, 28*, 1–16.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., & Rives, A. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems, 34*, 29287–29303.

Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N., & Madani, A. (2023). Progen2: Exploring the boundaries of protein language models. *Cell Systems, 14*(968–978), e963.

Nireeksha, N., Gollapalli, P., Varma, S. R., Hegde, M. N., & Kumari, N. S. (2022). Utilizing the potential of antimicrobial peptide LL-37 for combating SARS-COV- 2 viral load in saliva: An in silico analysis. *European Journal of Dentistry, 16*, 478–487.

Pang, Y., Wang, Z., Jhong, J. H., & Lee, T. Y. (2021). Identifying anti-coronavirus peptides by incorporating different negative datasets and imbalanced learning strategies. *Briefings in Bioinformatics, 22*, 1085–1095.

Pang, Y., Yao, L., Jhong, J.-H., Wang, Z., & Lee, T.-Y. (2021). AVPIden: A new scheme for identification and functional prediction of antiviral peptides based on machine learning approaches. *Briefings in Bioinformatics, 22*, Article bbab263.

Paul Michel, O. L., Graham Neubig. (2019). Are Sixteen Heads Really Better than One? *arXiv*.

Prevention, C.f. D. C.a. (2019). Anitibiotic resistance threats in the United States. *National Center for Emerging Zoonotic and Infectious Disease(U.S.) technical report*.

Renaud, S., & Mansbach, R. A. (2023). Latent spaces for antimicrobial peptide design. *Digital Discovery, 2*, 441–458.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences of the United States of America, 118*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Singh, V., Shrivastava, S., Kumar Singh, S., Kumar, A., & Saxena, S. (2022). StaBle-ABPpred: A stacked ensemble predictor based on biLSTM and attention mechanism for accelerated discovery of antibacterial peptides. *Briefings in Bioinformatics, 23*, Article bbab439.

Su, X., Xu, J., Yin, Y., Quan, X., & Zhang, H. (2019). Antimicrobial peptide identification using multi-scale convolutional network. *BMC Bioinformatics, 20*, 730.

Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., & Wu, C. H. (2007). UniRef: Comprehensive and non-redundant UniProt reference clusters. *Bioinformatics, 23*, 1282–1288.

Teufel, F., Gíslason, M. H., Almagro Armenteros, J. J., Johansen, A. R., Winther, O., & Nielsen, H. (2023). GraphPart: Homology partitioning for biological sequence analysis. *NAR Genomics and Bioinformatics, 5*.

Timmons, P. B., & Hewage, C. M. (2021). ENNAVIA is a novel method which employs neural networks for antiviral and anti-coronavirus activity prediction for therapeutic peptides. *Briefings in Bioinformatics, 22*, Article bbab258.

. UniRef|UniProt help. In. (2024).

Veltri, D., Kamath, U., & Shehu, A. (2018). Deep learning improves antimicrobial peptide recognition. *Bioinformatics, 34*, 2740–2747.

Verkuil, R., Kabeli, O., Du, Y., Wicky, B. I. M., Milles, L. F., Dauparas, J., Baker, D., Ovchinnikov, S., Sercu, T., & Rives, A. (2022). Language models generalize beyond natural proteins. *bioRxiv*.

Wan, F., Kontogiorgos-Heintz, D., & de la Fuente-Nunez, C. (2022). Deep generative models for peptide design. *Digit Discov, 1*, 195–208.

Wei, L., Zhou, C., Chen, H., Song, J., & Su, R. (2018). ACPred-FL: A sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides. *Bioinformatics, 34*, 4007–4016.

Xia, J., Zhang, L., Zhu, X., Liu, Y., Gao, Z., Hu, B., Tan, C., Zheng, J., Li, S., & Li, S. Z. (2024). Understanding the limitations of deep models for molecular property prediction: insights and solutions. *Advances in Neural Information Processing Systems, 36*.

Yan, K., Lv, H., Guo, Y., Peng, W., & Liu, B. (2023). sAMPpred-GAT: Prediction of antimicrobial peptide by graph attention network and predicted peptide structure. *Bioinformatics, 39*, Article btac715.

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Yin, B., & Hu, X. (2023). Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. In (pp. arXiv:2304.13712).

Zhang, J., & Mani, I. (2003). KNN approach to unbalanced data distributions: A case study involving information extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets* (pp. 1–7).

Zhang, Q., Chen, X., Li, B., Lu, C., Yang, S., Long, J., Chen, H., Huang, J., & He, B. (2022). A database of anti-coronavirus peptides. *Scientific Data, 9*, 294.

Zhang, R., Jiang, X., Qiao, J., Wang, Z., Tong, A., Yang, J., Yang, S., & Yang, L. (2021). Antimicrobial peptide DP7 with potential activity against SARS coronavirus infections. *Signal Transduction and Targeted Therapy, 6*, 140.

Zhang, Y., Lin, J., Zhao, L., Zeng, X., & Liu, X. (2021). A novel antibacterial peptide recognition algorithm based on BERT. *Briefings in Bioinformatics, 22*.

Zhou, W., Liu, Y., Li, Y., Kong, S., Wang, W., Ding, B., Han, J., Mou, C., Gao, X., & Liu, J. (2023). TriNet: A tri-fusion neural network for the prediction of anticancer and antimicrobial peptides. *Patterns (N Y), 4*, Article 100702.