Neoantigen Prioritization Using Transformers and Transfer Learning for The Development of Personalized Cancer Vaccines

Vicente Enrique Machaca Arceda
Universida Nacional de San Agustín de Arequipa, FIPS,
Arequipa, Perú,
vmachacaa@unsa.edu.pe

Abstract

Neoantigen prioritization is one of the most relevant steps in the development of cancer vaccines. Moreover, this process involves the prediction of bindings between peptides (neoantigen candidates) and Major Histocompatibility Complex (MHC). For that reason, in this work, we fine-tuned six Transformer models: TAPE, ProtBert-BFD, ESM2(t6), ESM2(t12), ESM2(t30), and ESM2(t33), adding a BiLSTM block in cascade for the task of peptide-MHC binding prediction. Moreover, we evaluated the effects of Gradient Accumulation Steps (GAS) and a layer freezing technique. After experiments, we noticed that TAPE with GAS (TAPE-GAS) and ESM2(t6) trained with the layer freezing method (ESM2(t6)-Freeze) got the best results. Then, we compared these models against state-of-art tools such as NetMHCpan4.1, MHCflurry2.0, Anthem, ACME, MixMHCpred2.2. After the assessment, TAPE-GAS and ESM2(t6)-Freeze outperformed the other methods on AUC, accuracy, recall, f1-score, and MCC.

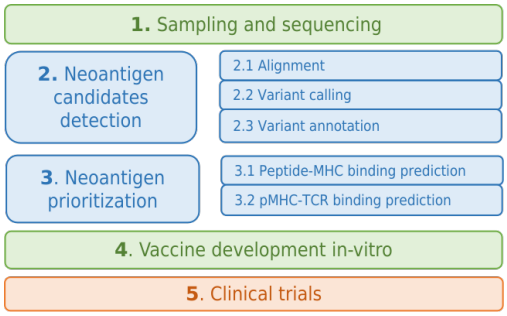
Keywords: Neoantigen, BERT, Transformer, Transfer Learning, Cancer, Immunology.

1 Introduction

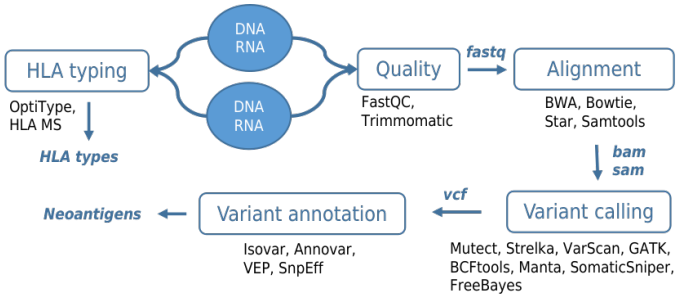
Cancer represents the most significant global health challenge [1]. Furthermore, according to the Cancer Research Institute of the United Kingdom, more than 18 million new cases and 10 million deaths were recorded in 2020 [2]. Furthermore, it is predicted that there will be 28 million new cases annually by around 2040 if the incidence remains stable, and population growth and aging continue according to recent trends [3]. This represents a 54.9% increase from 2020, with the increase expected to be higher in men (60.6%) than in women (48.8%). In this context, it is well known that traditional methods based on surgery, radiotherapy, and chemotherapy have low efficacy and adverse side effects [4]. Thus, the development of cancer immunotherapy has emerged, aiming to stimulate the immune system of the patient [5]. There are treatments like personalized vaccines, adoptive T-cell therapies, and immune checkpoint inhibitors. Among these, neoantigen-based vaccines have shown great potential by enhancing T-cell responses and are considered the most likely to succeed [5]. Additionally, neoantigens are used in immune checkpoint blockade therapy. Neoantigens are considered predictive biomarkers and targets for synergistic treatment in cancer immunotherapy [6].

Despite various efforts in the development of neoantigen detection methods, less than 5% of detected neoantigens activate the immune system, as reported by several studies [7, 8, 9, 10, 11]. The reasons are related to: the no integration of multiple data sources like DNA-seq, RNA-seq, and Mass Spectrometry (MS) [12]. Use of low-performance tools for peptide-MHC binding prediction like MHCFlurry [13] and NetMHCpan4.1 [14]. Neglecting the prediction of pMHC-TCR binding [15]. Overlooking information from alternative splicing events, structural DNA variants, and gene fusion mutations, this information is closely related to various types of cancer [16].

Neoantigen detection relies on an initial identification of candidates, which is followed by their subsequent prioritization (stages two and three of Fig. 1a). In this project, the vaccine development and clinical trials are out of the scope; however, they will be included in future works. Detection of neoantigen candidates involves a multi-step process (see Fig. 1b). DNA/RNA sequencing is initially conducted on both tumor and normal cells. Subsequently, quality assessment tools are employed, followed by the utilization of alignment tools. The process then proceeds to variant calling to identify genetic variants. Finally, variant annotation



(a) Vaccine process development



(b) Neoantigen detection process

Figure 1: (a) The process to detect neoantigens and develop personalized vaccines. It starts with DNA-seq or RNA-seq; then, neoantigen detection is performed, followed by neoantigen prioritization; subsequently, a personalized vaccine is developed *in vitro*; and clinical trials are applied. (b) Process for the Detection of Neoantigen Candidates Initially, DNA/RNA sequencing is conducted on both tumor and normal cells. Subsequently, quality assessment tools are employed, followed by the utilization of alignment tools. The process then proceeds to variant calling to identify variants. Finally, variant annotation tools are applied to generate a list of potential neoantigen candidates. Moreover, MHC typing tools are employed to determine the HLA or MHC types.

tools are applied to generate a list of potential neoantigen candidates. In addition to the aforementioned steps, quantitative proteomic tools are utilized for mass spectrometry (MS) data analysis. Moreover, MHC typing tools are employed to determine the Human Leukocyte Antigen (HLA) or Major Histocompatibility Complex (MHC) types.

Neoantigen prioritization is the third stage in cancer vaccines development (Fig. 1a). This stage takes candidates' neoantigens and then predicts their affinity to the Major Histocompatibility Complex (MHC); this problem is known as the pMHC binding prediction problem. Then, this pMHC complex is used to predict the interaction with the T-cell Receptor (TCR). Both problems take two protein sequences as input, and the goal is to predict their affinity (regression) or binding (classification). In summary, the proteins can be represented as $p = \{A, ..., Q\}$ and $q = \{A, N, K, L, ..., Q\}$. Then, we need to know the probability of affinity between p and q.

Thus, in this project, a method for neoantigen prioritization using Transformer and Transfer Learning is proposed. This method relies on the prediction of pMHC bindings. Moreover, the proposal fine-tuned six pre-trained BERT models, adding a block of BiLSTM at the end. Moreover, HLAB dataset [17] was used for training and testing. Additionally, a layer freezing methodology along with Gradient Accumulation Steps were applied. According to our results, ArgosMHC got promising results outperforming state-of-arts tools like NetMHCpan4.1 [14], MHCFlurry2.0 [13], Anthem [18], Acme [19], and MixMHCpred2.2 [20].

2 Related Works

Neoantigen detection and prioritization are relevant because of their applications in cancer immunology. As it is a very complex problem, we divided this topic into two categories: (1) the first group together all pipelines developed to detect and prioritize neoantigens; (2) in the second part, we empathized on pMHC binding prediction problem, which is related to neoantigen prioritization.

2.1 Pipelines

A bioinformatic pipeline in the neoantigen context is a software construct that assembles various command line tools. In the problem of neoantigen detection and prioritization, reliance on multiple tools is essential. For instance, we can use tools such as (1) FastQC to ensure sequence quality, (2) BWA handles alignment, (3) Samtools manipulates BAM files, (4) BCFtools is employed for variant calling, (5) Annovar provides variant annotation, and (6) netMHCpan4.1 predicts pMHC binding and pMHC-TCR binding affinity (neoantigen prioritization). However, the use of these diverse tools can introduce compatibility and dependency challenges. To address this issue, developers have created pipeline tools aimed at enhancing the usability of neoantigen detection software. These pipelines effectively manage the integration of these tools, mitigating potential conflicts and dependencies, thereby streamlining the overall neoantigen analysis process.

In Table 1, we present pipelines published since 2018. These pipelines use various types of information as input. For instance, the PGV Pipeline [15] and PEPPRMINT [21] use DNA-seq, while other tools such

as PGNneo [22], NAP-CNB [23], NaoANT-HILL [24], ProGeo-neo [25], ScanNeo [26], and Neopepse [12] use RNA-seq because these sequences better capture information about mutations and non-coding regions of DNA [22].

To reduce the complexity of pipelines, some proposals have opted to use Variant Calling Format (VCF) as input. These files contain mutation information and are obtained through alignment and mutation calling methods. These tools are: HLA3D [27], Neoepiscope [16], pVACtools [28], and NeoPredPipe [29]. However, the results obtained may be inferior compared to tools that use DNA-seq and RNA-seq.

Additionally, for accurate neoantigen detection, it is necessary to have the sequencing of Major Histocompatibility Complex (MHC) or Human Leukocyte Antigens (HLA) proteins. These proteins are necessary because they are used to predict the binding between potential neoantigens and MHC. These proteins are encoded by highly polymorphic genes, leading to substantial variation in peptide (neoantigen) binding, thereby influencing the set of peptides presented to T-cells [30]. In this context, the NeoPredPipe [29], and Neopepsee [12] pipelines request these HLA proteins as input, while others predict this information from DNA-seq. From a usability standpoint, obtaining the HLA types entails unnecessary effort for the user.

As we mentioned before, fusion genes are related to several types of cancer [16, 31, 32, 33, 34, 35, 36]. Thus, there are pipelines which include fusion genes detection methods: Integrate-neo [37], neoFusion [38], pVACfuse [28], NeoepitoPred [39], Epidisco [15], TrueNeo [40] and Antigen.garnish [41]. Gene fusions typically yield a higher number of neoantigens per mutation compared to single nucleotide variants (SNVs) and insertions/deletions (Indels). Furthermore, fusion-derived neoantigens exhibit heightened immunogenicity. Notably, neoantigens arising from frameshift fusions or passenger fusions are anticipated to possess the greatest immunogenic potential [42].

Table 1: Bioinfomatics pipelines developed for the detection of neoantigens. GN: Gene expression, VA: variant annotation, WEG: whole exome sequencing, WGS: whole genome sequencing.

Name	**Year**	**Input**	**Output**	**Tools**
PEPPRMINT	2023 [21]	DNA-seq	Neoantigens	BWA, Mutect, Strelka, ANNOVAR, OptiType, PEPPRMINT, netMHCpan4.1.
PGNneo	2023 [22]	VCF, RNA-seq, MS data	Neoantigens	Trimmomatic, BWA, SAMtools, GATK, Picard, OptiType, Annovar, Bedtools, MaxQuant, NetMHCpan4.1, Blastp.
HLA3D	2022 [27]	VCF, HLA, SMG, HBV	Neoantigens	MHCcluster, SAVES, PROCHECK, CoDockPP, Verify 3D, ERRAT, ClusterW2, 3Dmol, PSRPRED4.0, MHCf lurry.
NextNEOpi	2022 [43]	WES/WGS, RNA-seq	Neoantigens	OptiType, pVACseq, NetMHCpan, MHCflurry, NeoFuse, MiXCR.
Seq2Neo	2022 [44]	WES/WGS, RNA-seq	Neoantigens	Mutect2, STARFusion, ANNOVAR, Agfusion, NetMHCpan, MHCflurry, Pick-Pocket, NetMHCcon, TPMcalculator, NetCTLpan.
NAP-CNB	2021 [23]	RNA-seq	Neoantigens	Star, Picard, GATK, SplitNCigarsReads, MuTect2, Cufinks, Epi-Seq, pVAC,seq, Neoantimon, MuPeXI, BLOSUM62.
NeoANT-HILL	2020 [24]	RNA-seq, VCF	Neoantigens, GE	GATK, Mutect2, Optitype, NetMHC, NetMHCpan, NetMHCCcons, NetMHCstapan, PickPoket, SMM, SMMPMBEC, MHCflurry, NetMHCIIpan, NN-align, SMM-align, Sturniolo, Kallisto.
Neoepiscope	2020 [16]	VCF, BAM	Neoantigens	BWA, Bowtie2, Pindel, MuSE, RADIA, SomaticSniper, VarScan2, GATK, HapCUT2.
OpenVax	2020 [45]	DNA-seq, RNA-seq	Neoantigens	GATK 3.7, STAR, MuTect 1.1.7, Mutect 2, Strelka, NetMHCpan, NetMHCCcons, SMM, SMM with a Peptide.
ProGeo-neo	2020 [25]	RNA-seq, VCF	Neoantigens	SRA Toolkit, BWA, GATK, Bcftools, ANNOVAR, Kallisto, OptiType, NetMHCpan4.0.
pVACtools	2020 [28]	VCF	Neoantigens	CWL36, Cromwell37, ADNc38, BWA-MEM25, HaplotypeCaller28, MHCflurry14, MHCnuggets15, NetChop17, INTEGRATE-Neo19.
TruNeo	2020 [40]	DNA-seq, RNA-seq	Neoantigens	BWA, GATK v3.3, Somatic SNVs, STAR v2.5.3a, RSEM v1.3.0, NetMHCPan 3.0, netChop.
NeoPredPipe	2019 [29]	VCF, HLA	Neoantigens, VA	ANNOVAR, POLYSOLVER, netMHCpan, PeptideMatch.
ScanNeo	2019 [26]	RNA-seq	Neoantigens	HISAT2, BEDTools, BWA-MEM, pVAC-Seq, NetMHC, NetMHCpan.
Neopepsee	2018 [12]	RNA-seq, VCF, HLA	Neoantigens, GE	NetCTLpan, Swiss-Prot.
PGV Pipeline	2018 [15]	DNA-seq	Neoantigens	BWA-MEN, BQSR, MuTect, Strelka, STAR, seq2hla, Vaxrank, Isovar, MHCtools, Varcode, pyEnsembl.

2.2 Neoantigen prioritization

Neoantigen prioritization depends strongly on the accurate prediction of pMHC bindings. Moreover, the advent of Transformers has ushered in a new era in artificial intelligence, demonstrating significant success across various Natural Language Processing (NLP) tasks [46]. Thus, several Transformer models have been used for pMHC binding prediction problem. In Table 2, a detailed comparison of Transformers and deep learning methods is presented. For instance, BERTMHC [47] is a pan-specific pMHC-II binding and presentation prediction method that employs a BERT architecture and leverages transfer learning from

the Tasks Assessing Protein Embeddings (TAPE) [48]. The methodology involves stacking an average pooling layer followed by a Fully Connected (FC) layer after the TAPE model. Empirical assessments have shown that BERTMHC outperforms both NetMHCIIpan3.2 and PUFFIN. Additionally, ImmunoBERT [49] utilizes transfer learning from TAPE but focuses on pMHC-I prediction. This approach involves stacking a classification token's vector after the TAPE model. Furthermore, MHCRoBERTa [50] and HLAB [17] also leverage transfer learning. MHCRoBERTa employs self-supervised training with data from UniProtKB and Swiss-Prot databases, followed by fine-tuning with data from the Immune Epitope Database (IEDB) [51]. MHCRoBERTa performs better than NetMHCpan4.0 and MHCflurry2.0 in terms of Spearman Rank Correlation Coefficient (SRCC). In contrast, HLAB leverages transfer learning from ProtBert-BFD [52] and incorporates a BiLSTM model in cascade. Notably, on the HLA-A*01:01 allele, HLAB demonstrates a slight performance advantage over state-of-the-art methods, including NetMHCpan4.1, with at least a 0.0230 improvement in Area Under the Curve (AUC) and a 0.0560 increase in accuracy. Moreover, in a prior study, we conducted an evaluation of BERT Transformer models using padded sequences for pMHC binding prediction [53].

Lastly, it's worth noting recent allele-specific research efforts such as TransPHLA [54] and DapNet-HLA [55]. TransPHLA utilizes peptide self-attention mechanisms and has outperformed existing state-of-the-art methods, including NetMHCpan4.1. On the other hand, DapNet-HLA [55] presents promising results by incorporating an additional dataset (Swiss-Prot) for negative samples and harnessing the strengths of Convolutional Neural Networks (CNNs), SENet for pooling, and Long Short-Term Memory (LSTM) models. Furthermore, there are reviews and benchmarking works [56, 57, 58, 59, 58], which detail the pMHC binding prediction problem.

In this research, we compared the performance of six Transformer models (TAPE, ProtBert-BFD, ESM2(t6), ESM2(t12), ESM2(t30), and ESM2(t33)) for the task of peptide-MHC class-I binding prediction (pMHC-I). We fine-tuned each model by adding a BiLSTM block in cascade, based on the work of HLAB [17]. Furthermore, we evaluated the use of Gradient Accumulation Steps and a layer freezing methodology. Our contributions in this study can be summarized as follows: First, a comprehensive assessment and comparison of BERT models, we performed a thorough evaluation of BERT models, utilizing Gradient Accumulation Steps (GAS) and a layer freezing methodology. Following this evaluation, we identified two models, ESM2(t6)-Freeze (ESM2(t6) trained with layer freezing) and TAPE-GAS (TAPE trained with GAS), which achieved the highest scores. The second contribution refers to a comparison of these BERT models with state-of-art tools like NetMHCpan4.1, MHCFlurry2.0, MixMHCpred2.2, Anthem, and ACME. After conducting experiments, ESM2(t6)-Freeze and TAPE-GAS outperformed the other methods, achieving the highest results in terms of Area Under the Curve (AUC), accuracy, recall, f1-score, and Matthews Correlation Coefficient (MCC).

3 Proposal

The project develops methods for neoantigen prioritization relying on pMHC binding prediction. So, in order to accomplish this task, project encompasses an assessment and comparison of six Transformer models: TAPE [48], ProtBert-BFD [52] and ESM2 [71] (ESM2(t6), ESM2(t12), ESM2(t30), ESM2(t33)). These models were trained with large protein sequences datasets like Pfam [72], and BFD and UniRef50 [73]. We fine-tuned these models for the task of pMHC binding prediction problem. Moreover, this study looked for the benefits of using a layer freezing methodology. In Table 3, we present the characteristics of each model.

3.1 Pre-trained transformers

Tasks Assessing Protein Embeddings (TAPE) [48] is the first attempt to evaluate semi-supervised learning on protein sequences. TAPE has twelve layers of 512 units with eight attention heads, achieving 92 million parameters. The authors applied semi-supervised training with Pfam dataset [72], which has thirty million protein domains. Furthermore, the Pfam dataset is derived from UniProt Knowledge (UniProtKB) [74]; in particular, Pfam used sequences that belong to Reference Proteomes [75] instead of using the entire UniProtKB dataset. Consequently, Pfam has almost half of the protein sequences that other datasets based on UniProtKB.

ProtBert-BFD is part of a family of ProtTrans [52] models. The authors evaluated several deep learning architectures with BFD, UniRef50, and UniRef100 datasets, each with 2122, 45, and 216 million sequences. For instance, BFD is considered the most extensive collection of protein sequences; it merged UniProt [76] and proteins from multiple meta-genomics sequencing projects. Meanwhile, UniRef [73] provides a clustered set of protein sequences from UniProtKB. Notably, the larger dataset, BFD, is more noisy; it has sequence mistakes [52]. Some models proposed are ProtBert-BFD, ProtT5-XL and ProtT5-XXL, which have 420 million, 3 billion, and 11 billion parameters, respectively. ProtBert-BFD were trained with BFD; meanwhile,

Table 2: Transformers and deep learning methods with attention mechanism used for pMHC binding prediction.

Year	Name	Input	Model
2023[60]	ESM-GAT	One-hot	BERT with transfer learning from ESM1b and ESM2 fine-tuned with a Graph Attention Network (GAT) at the end. It outperformed NetMHCpan4.1.
2023[61]	CapsNet-MHC	BLOSUM62	Capsule Neural Network, it outperformed state-of-art tools for small peptides of 8 to 11-mer.
2023[62]	STMHCpan	One-hot	A Star-Transformer model, it use usefull for anylenght peptides and could extended for predicting T-cell responses.
2023[55]	DapNet-HLA	Fused word embedding	Combined the advantages of CNN, SENet (for pooling), and LSTM with attention.
2022[17]	HLAB	One-hot	BERT from ProtBert pre-trained model followed by a BiLSTM with attention mechanism.
2022[50]	MHC RoBERTa	One-hot	RoBERTa pre-trained and followed by 12 multi-head SA and a FC layers, it outperformed NetMHCPan 3.0.
2022[54]	TransPHLA	One-hot	It used SA mechanism based on four blocks, it slightly outperformed NetMHCpan4.1 and is faster making predictions.
2021[63]	CapTransformer	One-hot	Transformer with cross attention pooling to capture local and global information.
2021[49]	ImmunoBERT	One-hot	BERT from TAPE pre-trained followed by a linear layer. Authors claimed that N-terminal and C-terminals are highly relevant after analysis with SHAP and LIME.
2021[47]	BERTMHC	One-hot	BERT from TAPE pre-trained followed by a linear layer. It outperformed NetMHCIIpan3.2 and PUFFIN.
2021[64]	MATHLA	BLOSUM	It integrates BiLSTM with multi-head attention. It achieved an AUC score of 0.964, compared to 0.945, 0.925 and 0.905 for netMHCpan 4.0, MHCflurry and ACME respectively
2021[65]	DeepSeqPanII	BLOSUM62 and one-hot	It has two LSTM layers, an attention block and three FC layers. It got better results than NetMHCIIpan 3.2 on 26 of 54 alleles.
2021[66]	DeepNetBim	BLOSUM50	It uses separate CNNs for pMHC binding and immunogenetic with a attention module. It got 0.015 MAE for binding and 94.7 of accuracy for immunogenic.
2021[67]	DeepAttentionPan	BLOSUM62	CNN with an attention mechanism. It is allele-specific and got slightly better results than ACME for allele level.
2021[68]	SpConvM	One-hot, BLOSUM, and Deep	1D layer of CNN, an attention layer and a FC layer. Moreover, they employed global kernels to enhance their results, along with a combination of onehot, BLOSUM, and Deep.
2020 [69]	MHCAttNet	One-hot	CNN followed by an attention layer to generate a heat map over the amino acids.
2019[19]	ACME	BLOSUM50	CNN with attention, it extract interpretable patterns about pMHC binding. Moreover, it got SRCC of 0.569, AUC of 0.9 for HLA-A and 0.88 for HLA-B.
2019[70]	DeepHLApan	One-hot	Allele-specific model with three layers of Bidirectional GRU (BiGRU) with an attention layer. It got acc > 0.9 on 43 HLA alleles.

Table 3: Significant differences between TAPE, ProtBert-DFB, and ESM2 models. ESM2 has four models of different sizes. HS: Hidden size, AH: Attention heads

Model	Dataset	Samples	Layers	HS	AH	Params.
TAPE	Pfam	30M	12	768	12	92M
ProtBert-BFD	BFD	2122M	30	1024	16	420M
ESM2(t6)	Uniref50	60M	6	320	20	8M
ESM2(t12)	Uniref50	60M	12	480	20	35M
ESM2(t30)	Uniref50	60M	30	640	20	150M
ESM2(t33)	Uniref50	60M	33	1280	20	650M

ProtT5 models were initially trained with BFD and then with UniRef50, which improved performance by 2.8% and 1.4% for ProtT5-XL and ProtT5-XXL respectively. Nevertheless, ProtT5-XL outperformed both ProtBert-BFD and the biggest model ProtT5-XXL. The authors claimed that the number of samples increased performance but didn't observe similarity consistent with model size. They suggested that larger models see fewer samples in the same computing power, so larger models need larger datasets. For that reason, we have opted for ProtBert as it is smaller than ProtT5-XL, and we believe it is better suited to the current dataset size.

ESM-2 [71] is a family of transformer models that scales from 8 million to 15 billion parameters. The model is based on BERT [77], outperforming its previous version ESM-1b [78] by removing dropout in hidden and attention layers. Furthermore, the authors suggested that absolute positional encoding methods don't extrapolate well; consequently, they used Rotary Position Embedding (RoPE). Significantly, with the use of RoPE, the training cost is slightly increased; meanwhile, it improves model quality for small models [71]. Moreover, the authors used the non-redundant UniRef50 [73] dataset from UniProt, with 60 million protein sequences.

3.2 Fine tuning

For fine-tuning, we stacked in cascade a BiLSTM at the end of the pre-trained model. The BiLSTM is based on HLAB [17] and has two layers with 768 units. In Fig. 2, we present the whole model for pMHC-I binding

prediction.

Figure 2: The proposed architecture for pMHC-I binding prediction using a pre-trained transformer model and a BiLSTM block stacked in cascade.

Moreover, it is well-established that when fine-tuning large transformer models, the final layers exhibit more significant changes, while the initial layers, closer to the input, undergo relatively minor modifications [79, 80, 81]. Consequently, we compared the results of freezing the pre-trained model and only updating the BiLSTM parameters versus updating the whole model parameters.

Furthermore, large transformer models run out of GPU memory. Therefore, inspired by similar works training transformer models [82, 83, 84], we evaluated the results of applying gradient accumulation steps during training.

Finally, we used the following hyperparameters: learning rate of 2e-6, weight decay of 0.0001, momentum of 0.9, warmup steps of 200000 with linear decay, ADAM optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) with bias correction, and early stopping. We have those values to overcome vanishing gradients. This process is described in the next section.

4 Experiments and Results

4.1 Dataset

We used peptide sequences from the Anthem dataset [18]. It consists of 539019, 179673, and 172580 samples for training, validation, and testing, respectively. In more detail, in Fig. S1, we present the distribution of samples by k-mers; 9-mers comprise the majority of samples in the database.

4.2 Binary classifier and metrics

The pMHC binding prediction problem is a regression problem. Nonetheless, based on the dataset employed in this study, it could also be approached as a binary classification problem by selecting an appropriate threshold. Moreover, the machine learning metrics used in this work are: Accuracy (Acc.), Precision (Precis.), Recall, F1-score (F1-sc), and Area Under the Curve (AUC).

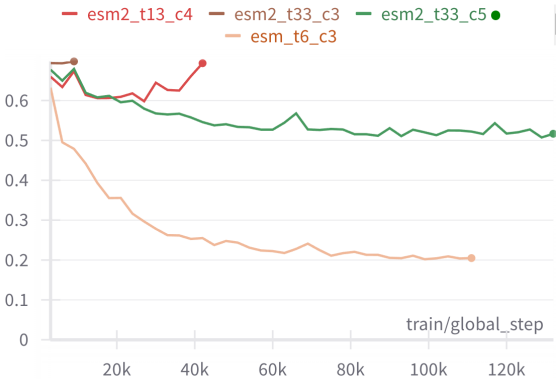
4.3 Vanishing gradients

At the beginning of the experiments, the bigger models suffered from the vanishing gradients problem. Thus, according to best practices to fine-tune the BERT model [85], we evaluated lower learning rates, increased the warmup steps in the learning scheduler, and used ADAM with bias correction. So, we evaluated three configurations: (c3) $lr = 2e-5$ and 20k warnup steps, (c4) $lr = 1e-5$ and 100k warnup steps, and (c5) $lr = 2e-6$ and 200k warnup steps. In Fig. 3a, the loss comparison during training is plotted. In this case, we see that configuration three is good for the smaller model ESM2(t6); however, the same configuration didn't work for the bigger model ESM2(t33). Moreover, in Fig. 3b the layer one's gradients of ESM2(t6) show a normal behavior; meanwhile, in Fig. 3c, we can see how gradients of the bigger model tend to zero. Thus, configuration four and five were evaluated; sadly, configuration four also went to vanishing gradients. Meanwhile, configuration five overcame that problem according to Fig. 3d. For that reason, Configuration Five was applied to the following experiments.

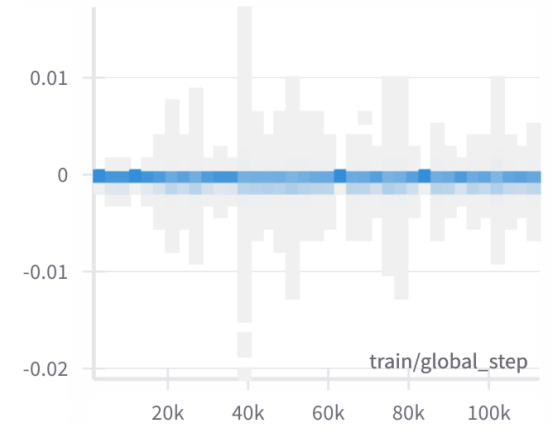
4.4 The layer freezing methodology and GAS

For the layer freezing methodology, we froze all the Transformer's parameters and just trained the BiLSTM block. Using this method speeds up the training and still achieves good performance, as discussed in prior works [79, 80, 81]. The performance comparison is presented in Table 4; the suffix 'GAS' signifies the integration of Gradient Accumulation Steps (GAS), while the suffix 'Freeze' is indicative of our application of the layer freezing methodology to the models.

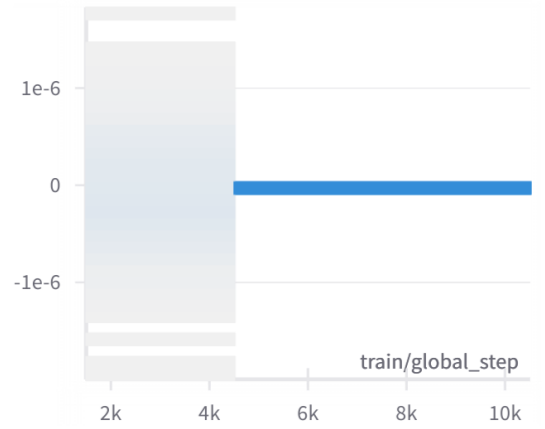
For a more detailed comparison, we extended the training epochs of the best-performing models from Table 4, including ESM2(T6) and TAPE, to 30 epochs with early stopping. It is worth noting that ProtBert-BFD was excluded from the analysis due to its poor performance. As indicated in Table 5, the ESM2 models yield their best results when the layer freezing methodology is applied. In contrast, for TAPE, the best results



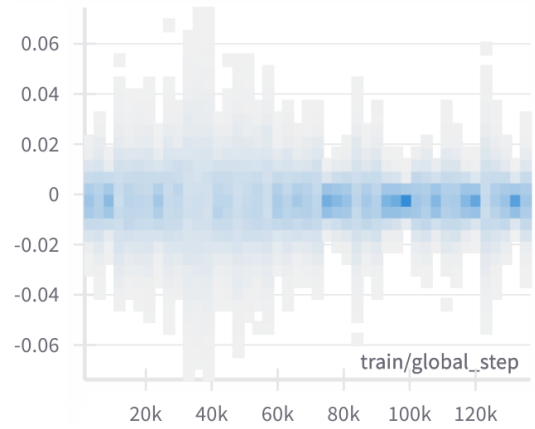
(a) Loss



(b) ESM2(t6) Configuration 3.



(c) ESM2(t33) Configuration 3.



(d) ESM2(t33) Configuration 5.

Figure 3: (a) Loss during training of three configurations varying the learning rate and warmup steps. (b) First layer's gradients of ESM2(t6) using third configuration. (c) First layer's gradients of ESM2(t33) using the fourth configuration. (d) First layer's gradients of ESM2(t33) using the fifth configuration.

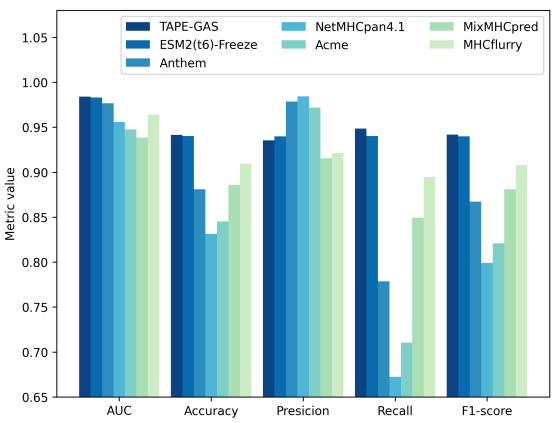
are achieved when using GAS without freezing. Notably, TAPE-GAS and ESM2(t6)-Freeze produced the most favorable outcomes, with TAPE-GAS slightly outperforming ESM2(t6)-Freeze in this regard.

4.5 Comparison with state-of-art methods

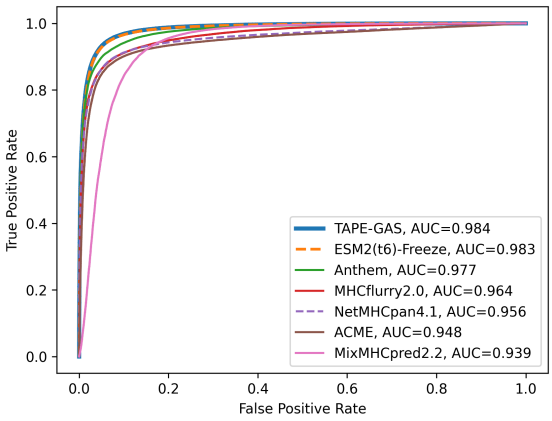
Additionally, we compare the best models: ESM2(t6)-Freeze and TAPE-GAS trained for 30 epochs (see Table 5) with state-of-art methods. We covered NetMHCpan4.1 [14], and MHCFlurry2.0 [13] because are well-know baselines methods; and three latest tools such as Anthem [18], Acme [19] and MixMHCpred2.2 [20]. During our evaluation of these tools on the test dataset, we encountered specific considerations for ACME. To ensure a fair assessment, we excluded the following alleles from the evaluation for ACME: HLA-C01:02, HLA-C02:02, HLA-C03:03, HLA-C03:04, HLA-C04:01, HLA-C05:01, HLA-C06:02, HLA-C07:01, HLA-C07:02, HLA-C07:04, HLA-C08:02, HLA-C12:03, HLA-C14:02, HLA-C15:02, HLA-C16:01, HLA-C17:01, HLA-A02:50, HLA-A24:06, HLA-A24:13, HLA-A32:15, HLA-B45:06, and HLA-B83:01. This exclusion was necessary as ACME was unable to predict peptide-MHC binding for these particular alleles,

It's worth noting that the choice of threshold for predicting pMHC binding can vary depending on the specific tool and k-mer used. This variability makes the AUC an ideal metric for comparing methods, as it provides a robust evaluation that isn't sensitive to threshold differences. For that reason, in Figure 4a and 4b, we present the AUC and ROC curve respectively of TAPE-GAS, ESM2(t6)-Freeze, NetMHCpan4.1, and MHCFlurry2.0, Anthem, Acme, and MixMHCpred2.2. According to these plots, TAPE-GAS, ESM2(t6)-Freeze got the highest AUC value.

Furthermore, when assessing binary classification performance metrics, we standardized the threshold for TAPE-GAS and ESM2(t6) at 0.5. We maintained a threshold of 0.5 for NetMHCpan4.1, in accordance with its recommended setting, while for ACME, we adhered to a threshold of 0.42, as advised in its documentation. In the case of Anthem, the tool provided binary binding predictions directly. However, for



(a) Thirty epochs



(b) ROC curves

Figure 4: (a) The AUC values for TAPE-GAS and ESM2(t6) trained for 30 epochs, in comparison to state-of-the-art methods. (b) ROC curves for TAPE-GAS and ESM2(t6) trained for 30 epochs, in comparison to state-of-the-art methods.

MixMHCpred2.2 and MHCfurry, we determined the optimal threshold values from the test dataset, resulting in 2.7308 and 0.09439, respectively. In Table 6, we present a comprehensive comparison between TAPE-GAS and ESM2(t6)-Freeze (trained for 30 epochs), and state-of-the-art methods. The results clearly demonstrate that TAPE-GAS and ESM2(t6)-Freeze consistently outperform the existing state-of-the-art tools across all these metrics: AUC, Accuracy, Recall, F1-score, and MCC.

Finally, we show the AUC distribution for TAPE-GAS and ESM2(t6)-Freeze, both trained for 30 epochs, along with Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2, and MHCflurry2.0. In this case, we evaluated each distribution by k-mer (Fig. 5). In all k-mer, TAPE-GAS and ESM2(t6)-Freeze present the highest score. Furthermore, it's worth noting that the compact ESM2(t6)-Freeze model delivers superior results for longer peptides with lengths of 11, 12, 13, and 14.

5 Discussion

When specifically considering ESM2 Transformer models, the most favorable results were obtained with the smallest model, ESM2(t6), as indicated in Table 4. However, it's noteworthy that the authors of ESM2 reported in their paper that for various other tasks, larger models like ESM2(t30) and ESM2(t33) outperformed the smaller ones such as ESM2(t6) and ESM2(t12) [71]. Additionally, it is well-established that larger models tend to exhibit faster learning but require more extensive training datasets [52]. In the case of pMHC-I binding prediction, our study employed a dataset comprising 559,019 samples, which we believe is not sufficiently large for ESM2(t33), a model boasting 650 million parameters. In future research endeavors, we plan to assess the performance of larger models using more extensive datasets. Another potential reason for the superior performance of ESM2(t6) could be attributed to the use of Rotary Position Embedding (RoPE) used instead of absolute positional encoding. While RoPE may lead to a slight increase in training cost, it has been observed to enhance the quality of results, particularly for smaller models [71].

During the training of Transformer models, we explored the implementation of a layer freezing methodology. This approach involves locking the Transformer model while updating only the BiLSTM parameters. As reported in various studies on freezing methodologies in Transformers [79, 80, 81], this method is generally well-suited to accelerate the training process, even though it may lead to a slight sacrifice in performance. Surprisingly, for ESM2 models, this methodology yielded the best results. Moreover, we encountered a recurring issue of vanishing gradients when training large models such as TAPE-normal, ProtBert-normal, ESM2(t30)-normal, and ESM2(t33)-normal. This challenge is a common occurrence when training large language models, as gradients tend to approach zero values after several training steps. To address this, we reduce the learning rate and increased the warmup steps.

Furthermore, a comparison between TAPE, ProtBert-BFD, and ESM2, each of one described in Table 3. The metrics are presented in Table 4. According to this information, ProtBert-BFD got the worst result despite the fact this model was pre-trained with the largest dataset BFD with 2122M samples; moreover, it has 420M parameters. We believe this result is caused by the noisy information and sequence mistakes in the BFD dataset [52]. Moreover, large Transformers models need more data for training [52], and we fine-tuned this model with 559019 samples.

Table 4: Performance evaluation of Transformer models with Gradient Accumulation Steps (GAS) and the layer freezing methodology trained for three epochs. Moreover, the suffix 'Normal' stands for classic training. The inclusion of the suffix 'GAS' in each model signifies the integration of Gradient Accumulation Steps, while the suffix 'Freeze' is indicative of our application of the layer freezing methodology to the models.

Model	Acc.	Precis.	Recall	F1-sc.	AUC
ESM2(t6)-Normal	0.9344	**0.9334**	0.9354	0.9344	0.9805
ESM2(t6)-Freeze	**0.9351**	0.9253	**0.9464**	**0.9357**	**0.9812**
ESM2(t6)-GAS	0.8986	0.8966	0.9007	0.8986	0.9602
ESM2(t6)-Freeze-GAS	0.8869	0.8913	0.8806	0.8860	0.9520
ESM2(t12)-Normal	0.9327	0.9243	0.9422	0.9332	0.9799
ESM2(t12)-Freeze	**0.9344**	**0.9251**	**0.9451**	**0.9350**	**0.9808**
ESM2(t12)-GAS	0.9010	0.9279	0.8692	0.8976	0.9655
ESM2(t12)-Freeze-GAS	0.8805	0.8556	0.9149	0.8843	0.9475
ESM2(t30)-Normal	0.8923	0.8956	0.9076	0.9034	0.9467
ESM2(t30)-Freeze	**0.9303**	**0.9185**	**0.9440**	**0.9311**	**0.9786**
ESM2(t30)-GAS	0.9090	0.9167	0.8993	0.9079	0.9675
ESM2(t30)-Freeze-GAS	0.8565	0.8156	0.9206	0.8649	0.9312
ESM2(t33)-Normal	0.6797	0.7122	0.8076	0.7220	0.7458
ESM2(t33)-Freeze	**0.6818**	**0.7139**	0.6044	0.6546	**0.7613**
ESM2(t33)-GAS	0.6767	0.6312	0.8467	0.7233	0.7442
ESM2(t33)-Freeze-GAS	0.6738	0.6254	**0.8633**	**0.7254**	0.7514
TAPE-Normal	0.8986	0.8245	0.9045	0.8845	0.9267
TAPE-Freeze	0.9342	0.9276	0.9415	0.9345	0.9809
TAPE-GAS	**0.9371**	**0.9290**	**0.9463**	**0.9376**	**0.9826**
TAPE-Freeze-GAS	0.8914	0.8851	0.8989	0.8920	0.9564
ProtBert-Normal	0.7456	0.7045	0.7205	0.7599	0.7178
ProtBert-Freeze	0.9083	0.9176	0.8968	0.9071	0.9673
ProtBert-GAS	**0.9138**	**0.9569**	0.8662	**0.9093**	**0.9767**
ProtBert-Freeze-GAS	0.7864	0.7333	**0.8988**	0.8076	0.8669

Table 5: Performance evaluation of Transformer models with GAS and the layer freezing methodology, trained for thirty (30) epochs. Moreover, the suffix 'Normal' stands for classic training. The inclusion of the suffix 'GAS' in each model signifies the integration of GAS, while the suffix 'Freeze' is indicative of our application of the layer freezing methodology to the models.

	Acc.	Precis.	Recall	F1-sc.	AUC
ESM2(t6)-Normal	0.9390	0.9333	**0.9453**	0.9392	0.9797
ESM2(t6)-Freeze	**0.9401**	**0.9398**	0.9402	**0.9400**	**0.9830**
ESM2(t6)-GAS	0.9366	0.9322	0.9413	0.9368	0.9818
ESM2(t6)-Freeze-GAS	0.9354	0.9326	0.9383	0.9355	0.9813
TAPE-Normal	0.9086	0.9399	0.9281	0.9145	0.9648
TAPE-Freeze	0.9395	**0.9404**	0.9382	0.9393	0.9815
TAPE-GAS	**0.9415**	0.9352	**0.9484**	**0.9418**	**0.9841**
TAPE-Freeze-GAS	0.9359	0.9297	0.9428	0.9362	0.9820

Additionally, it is notable that TAPE achieved the best results, with ESM2(t6) following closely (as shown in Table 4). TAPE models were pre-trained using the Pfam dataset, which is the smallest dataset in this comparison, containing approximately 30 million samples. It's important to mention that the Pfam dataset is derived from UniProtKB and selectively includes sequences belonging to Reference Proteomes rather than encompassing the entire UniProtKB database [75]. Consequently, Pfam covers half of the protein sequences compared to other datasets based on UniProtKB, but its samples are of higher quality. Therefore, it is plausible to assume that TAPE encapsulates a more comprehensive and refined representation of protein information compared to other pre-trained models. Moreover, ESM2(t6) achieved results that closely rival TAPE's performance, as demonstrated in Table 5. Notably, ESM2(t6) comprises only 8 million parameters, compared to 92 million parameters of TAPE. Furthermore, both models were trained on samples from UniProtKB, although TAPE used a subset of this dataset. Moreover, ESM2(t6) outperformed TAPE for longer peptides, ranging from 11 to 14 mers, as depicted in Fig. 5. These findings strongly position ESM2(t6) as a prime candidate for future analyses due to its remarkable performance and cost-effectiveness.

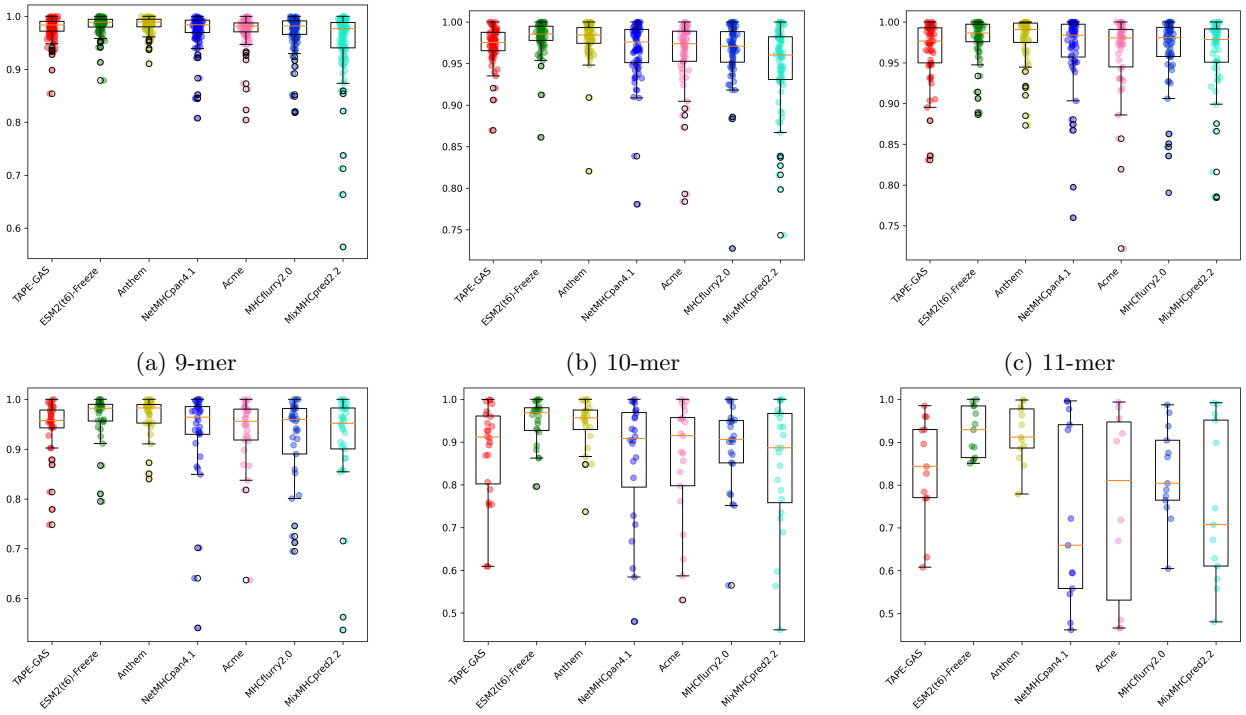
6 Conclusions

In our comparative analysis of the six Transformer models TAPE, ProtBert-BFD, ESM2(t6), ESM2(t12), ESM2(t30), and ESM2(t33) with the incorporation of GAS and the layer freezing technique, we observed that ESM2(t6)-Freeze, and TAPE-GAS achieved the most favorable outcomes. Additionally, we found that the layer freezing methodology accelerated the training process and produced the most favorable results for ESM2 models. In contrast, using GAS led to the best results for TAPE and ProtBert.

Moreover, after training ESM2(t6)-Freeze and TAPE-GAS for thirty epochs, the models surpassed state-of-the-art methods, including NetMHCpan4.1, MHCflurry2.0, Anthem, ACME, and MixMHCpred2.2, in terms of various performance metrics like AUC, accuracy, recall, f1-score, and MCC. This demonstrates the

Table 6: Performance evaluation of Transformer models TAPE-GAS and ESM2(t6)-Freeze, trained for 30 epochs, against Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2, and MhcFlurry2.0.

	Acc.	**Precis.**	**Recall**	**F1-sc**	**AUC**	**MCC**
TAPE-GAS	**0.9415**	0.9352	**0.9484**	**0.9418**	**0.9841**	**0.8831**
ESM2(t6)-Freeze	**0.9401**	0.9398	**0.9402**	**0.9400**	**0.9830**	**0.8802**
Anthem	0.8811	**0.9786**	0.7787	0.8673	0.9768	0.7785
NetMHCpan4.1	0.8312	**0.9844**	0.6724	0.7991	0.9557	0.6982
ACME	0.8452	0.9717	0.7105	0.8208	0.9476	0.7165
MixMHCpred2.2	0.8857	0.9155	0.8493	0.8811	0.9386	0.7733
MhcFlurry2.0	0.9093	0.9211	0.8948	0.9078	0.9642	0.8189



(a) 9-mer (b) 10-mer (c) 11-mer

(d) 12-mer (e) 13-mer (f) 14-mer

Figure 5: The AUC distribution for TAPE-GAS and ESM2(t6)-Freeze, both trained for 30 epochs, along with Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2, and MHCflurry2.0.

advantages of fine-tuning large Transformer models for predicting peptide-MHC binding, underscoring their potential to enhance this critical task.

Additionally, the vanishing gradient problem is a common problem when training large BERT models. So, in order to avoid this problem, we noticed that larger models need very small learning rates at the beginning of training. Furthermore, in order to maintain stability, it is very useful to use a scheduler to increase and decrease the learning rate during training. Thus, after experiments, we managed to use a learning rate of $2e-6$ and 200k step for warnup. Additionally, we used the ADAM optimizer with bias correction.

7 Future works

In this work, we evaluated TAPE, ProtBert-BFD, ESM2(t6), ESM2(t12), ESM2(t30), and ESM2(t33), each one with 92, 420, 8, 35, 150, and 650 million parameters respectively. However, we have other alternatives like ProtT5-XL and ProtT5-XXL, ESM2(t36), and ESM2(t48), each one with 3, 11, 3, and 15 billion parameters, respectively. We didn't evaluate these models because of the small size of the dataset and the training cost. Nevertheless, we planned to evaluate these huge Transformer models with a large dataset comprising samples from Anthem dataset [18], MixMHXpred2.2 [20], and the most recent benchmarking of pMHC binding prediction tools [58].

Moreover, given the considerable training cost associated with training large Transformer models, we plan to investigate the potential advantages of utilizing DistilBERT [86] and LoRA [87] for training and future prediction tasks.

Furthermore, we fine-tuned each Transformer model, adding a BiLSTM block at the end, based on the

work of HLAB [17]. Looking ahead, we plan to assess the effectiveness of a Star-Transformer block, similar to the methodology employed in SMHCpan [62]. Furthermore, considering the promising results demonstrated in ESM-GAT [60], we believe that the inclusion of a Graph Attention Network (GAT) could significantly enhance our model's performance in future research. Finally, we would like to evaluate the methodology used by TransPHLA [54], due to its demonstrated effectiveness in handling peptides of different lengths.

References

[1] R. L. Siegel, K. D. Miller, N. S. Wagle, and A. Jemal, "Cancer statistics, 2023," *Ca Cancer J Clin*, vol. 73, no. 1, pp. 17–48, 2023.

[2] C. R. UK, "Worldwide cancer statistics," 2023. [Online]. Available: https://www.cancerresearchuk.org/health-professional/cancer-statistics/worldwide-cancer

[3] ——, "Worldwide cancer incidence statistics," 2023. [Online]. Available: https://www.cancerresearchuk.org/health-professional/cancer-statistics/worldwide-cancer/incidence#heading-Zero

[4] M. Peng, Y. Mo, Y. Wang, P. Wu, Y. Zhang, F. Xiong, C. Guo, X. Wu, Y. Li, X. Li *et al.*, "Neoantigen vaccine: an emerging tumor immunotherapy," *Molecular cancer*, vol. 18, no. 1, pp. 1–14, 2019.

[5] E. S. Borden, K. H. Buetow, M. A. Wilson, and K. T. Hastings, "Cancer neoantigens: Challenges and future directions for prediction, prioritization, and validation," *Frontiers in Oncology*, vol. 12, 2022.

[6] X. Fang, Z. Guo, J. Liang, J. Wen, Y. Liu, X. Guan, and H. Li, "Neoantigens and their potential applications in tumor immunotherapy," *Oncology Letters*, vol. 23, no. 3, pp. 1–9, 2022.

[7] L. Mattos, M. Vazquez, F. Finotello, R. Lepore, E. Porta, J. Hundal, P. Amengual-Rigo, C. Ng, A. Valencia, J. Carrillo *et al.*, "Neoantigen prediction and computational perspectives towards clinical benefit: recommendations from the esmo precision medicine working group," *Annals of oncology*, vol. 31, no. 8, pp. 978–990, 2020.

[8] N. A. Mill, C. Bogaert, W. van Criekinge, and B. Fant, "neoms: Attention-based prediction of mhc-i epitope presentation," *bioRxiv*, 2022.

[9] B. Bulik-Sullivan, J. Busby, C. D. Palmer, M. J. Davis, T. Murphy, A. Clark, M. Busby, F. Duke, A. Yang, L. Young *et al.*, "Deep learning using tumor hla peptide mass spectrometry datasets improves neoantigen identification," *Nature biotechnology*, vol. 37, no. 1, pp. 55–63, 2019.

[10] M. Bassani-Sternberg, S. Pletscher-Frankild, L. J. Jensen, and M. Mann, "Mass spectrometry of human leukocyte antigen class i peptidomes reveals strong effects of protein abundance and turnover on antigen presentation*[s]," *Molecular & Cellular Proteomics*, vol. 14, no. 3, pp. 658–673, 2015.

[11] M. Yadav, S. Jhunjhunwala, Q. T. Phung, P. Lupardus, J. Tanguay, S. Bumbaca, C. Franci, T. K. Cheung, J. Fritsche, T. Weinschenk *et al.*, "Predicting immunogenic tumour mutations by combining mass spectrometry and exome sequencing," *Nature*, vol. 515, no. 7528, pp. 572–576, 2014.

[12] S. Kim, H. S. Kim, E. Kim, M. Lee, E.-C. Shin, and S. Paik, "Neopepsee: accurate genome-level prediction of neoantigens by harnessing sequence and amino acid immunogenicity information," *Annals of Oncology*, vol. 29, no. 4, pp. 1030–1036, 2018.

[13] T. J. O'Donnell, A. Rubinsteyn, and U. Laserson, "Mhcflurry 2.0: improved pan-allele prediction of mhc class i-presented peptides by incorporating antigen processing," *Cell systems*, vol. 11, no. 1, pp. 42–48, 2020.

[14] B. Reynisson, B. Alvarez, S. Paul, B. Peters, and M. Nielsen, "Netmhcpan-4.1 and netmhciipan-4.0: improved predictions of mhc antigen presentation by concurrent motif deconvolution and integration of ms mhc eluted ligand data," *Nucleic acids research*, vol. 48, no. W1, pp. W449–W454, 2020.

[15] A. Rubinsteyn, J. Kodysh, I. Hodes, S. Mondet, B. A. Aksoy, J. P. Finnigan, N. Bhardwaj, and J. Hammerbacher, "Computational pipeline for the pgv-001 neoantigen vaccine trial," *Frontiers in immunology*, vol. 8, p. 1807, 2018.

[16] M. A. Wood, A. Nguyen, A. J. Struck, K. Ellrott, A. Nellore, and R. F. Thompson, "Neoepiscope improves neoepitope prediction with multivariant phasing," *Bioinformatics*, vol. 36, no. 3, pp. 713–720, 2020.

[17] Y. Zhang, G. Zhu, K. Li, F. Li, L. Huang, M. Duan, and F. Zhou, "Hlab: learning the bilstm features from the protbert-encoded proteins for the class i hla-peptide binding prediction," *Briefings in Bioinformatics*, 2022.

[18] S. Mei, F. Li, D. Xiang, R. Ayala, P. Faridi, G. I. Webb, P. T. Illing, J. Rossjohn, T. Akutsu, N. P. Croft *et al.*, "Anthem: a user customised tool for fast and accurate prediction of binding between peptides and hla class i molecules," *Briefings in Bioinformatics*, vol. 22, no. 5, p. bbaa415, 2021.

[19] Y. Hu, Z. Wang, H. Hu, F. Wan, L. Chen, Y. Xiong, X. Wang, D. Zhao, W. Huang, and J. Zeng, "Acme: pan-specific peptide–mhc class i binding prediction through attention-based deep neural networks," *Bioinformatics*, vol. 35, no. 23, pp. 4946–4954, 2019.

[20] D. Gfeller, J. Schmidt, G. Croce, P. Guillaume, S. Bobisse, R. Genolet, L. Queiroz, J. Cesbron, J. Racle, and A. Harari, "Improved predictions of antigen presentation and tcr recognition with mixmhcpred2. 2 and prime2. 0 reveal potent sars-cov-2 cd8+ t-cell epitopes," *Cell Systems*, vol. 14, no. 1, pp. 72–83, 2023.

[21] L. Y. Zhou, F. Zou, and W. Sun, "Prioritizing candidate peptides for cancer vaccines through predicting peptide presentation by hla-i proteins," *Biometrics*, vol. 79, no. 3, pp. 2664–2676, 2023.

[22] X. Tan, L. Xu, X. Jian, J. Ouyang, B. Hu, X. Yang, T. Wang, and L. Xie, "Pgnneo: A proteogenomics-based neoantigen prediction pipeline in noncoding regions," *Cells*, vol. 12, no. 5, p. 782, 2023.

[23] C. Wert-Carvajal, R. Sánchez-García, J. R. Macías, R. Sanz-Pamplona, A. M. Pérez, R. Alemany, E. Veiga, C. Ó. S. Sorzano, and A. Muñoz-Barrutia, "Predicting mhc i restricted t cell epitopes in mice with nap-cnb, a novel online tool," *Scientific reports*, vol. 11, no. 1, pp. 1–10, 2021.

[24] A. C. M. Coelho, A. L. Fonseca, D. L. Martins, P. B. Lins, L. M. da Cunha, and S. J. de Souza, "neoant-hill: an integrated tool for identification of potential neoantigens," *BMC Medical Genomics*, vol. 13, no. 1, pp. 1–8, 2020.

[25] Y. Li, G. Wang, X. Tan, J. Ouyang, M. Zhang, X. Song, Q. Liu, Q. Leng, L. Chen, and L. Xie, "Progeo-neo: a customized proteogenomic workflow for neoantigen prediction and selection," *BMC medical genomics*, vol. 13, no. 5, pp. 1–11, 2020.

[26] T.-Y. Wang, L. Wang, S. K. Alam, L. H. Hoeppner, and R. Yang, "Scanneo: identifying indel-derived neoantigens using rna-seq data," *Bioinformatics*, vol. 35, no. 20, pp. 4159–4161, 2019.

[27] X. Li, X. Lin, X. Mei, P. Chen, A. Liu, W. Liang, S. Chang, and J. Li, "Hla3d: an integrated structure-based computational toolkit for immunotherapy," *Briefings in bioinformatics*, vol. 23, no. 3, p. bbac076, 2022.

[28] J. Hundal, S. Kiwala, J. McMichael, C. A. Miller, H. Xia, A. T. Wollam, C. J. Liu, S. Zhao, Y.-Y. Feng, A. P. Graubert *et al.*, "pvactools: a computational toolkit to identify and visualize cancer neoantigens," *Cancer immunology research*, vol. 8, no. 3, pp. 409–420, 2020.

[29] R. O. Schenck, E. Lakatos, C. Gatenbee, T. A. Graham, and A. R. @miscNCIdictionary2022, author = NCI, title = National Cancer Institute Dictionary, year = 2022, url = https://www.cancer.gov/publications/dictionaries/genetics-dictionary, urldate = 2022-03-20 Anderson, "Neopredpipe: high-throughput neoantigen prediction and recognition potential pipeline," *BMC bioinformatics*, vol. 20, no. 1, pp. 1–6, 2019.

[30] E. T. Abualrous, J. Sticht, and C. Freund, "Major histocompatibility complex (mhc) class i and class ii proteins: impact of polymorphism on antigen presentation," *Current Opinion in Immunology*, vol. 70, pp. 95–104, 2021.

[31] T. Wei, J. Lu, T. Ma, H. Huang, J.-P. Kocher, and L. Wang, "Re-evaluate fusion genes in prostate cancer," *Cancer Informatics*, vol. 20, p. 11769351211027592, 2021.

[32] V. D. Yakushina, L. V. Lerner, and A. V. Lavrov, "Gene fusions in thyroid cancer," *Thyroid*, vol. 28, no. 2, pp. 158–167, 2018.

[33] S. Panicker, G. Chengizkhan, R. Gor, I. Ramachandran, and S. Ramalingam, "Exploring the relationship between fusion genes and micrornas in cancer," *Cells*, vol. 12, no. 20, p. 2467, 2023.

[34] Y. Lei, Y. Lei, X. Shi, and J. Wang, "Eml4-alk fusion gene in non-small cell lung cancer," *Oncology letters*, vol. 24, no. 2, pp. 1–6, 2022.

[35] Y. Zhang, J. Sun, Y. Song, P. Gao, X. Wang, M. Chen, Y. Li, and Z. Wu, "Roles of fusion genes in digestive system cancers: Dawn for cancer precision therapy," *Critical Reviews in Oncology/Hematology*, vol. 171, p. 103622, 2022.

[36] I. Panagopoulos, K. Andersen, I. M. R. Johannsdottir, F. Micci, and S. Heim, "Novel mycbp:: Ehd2 and runx1:: Znf780a fusion genes in t-cell acute lymphoblastic leukemia," *Cancer Genomics & Proteomics*, vol. 20, no. 1, pp. 51–63, 2023.

[37] J. Zhang, E. R. Mardis, and C. A. Maher, "Integrate-neo: a pipeline for personalized gene fusion neoantigen discovery," *Bioinformatics*, vol. 33, no. 4, p. 555, 2017.

[38] Z. Wei, C. Zhou, Z. Zhang, M. Guan, C. Zhang, Z. Liu, and Q. Liu, "The landscape of tumor fusion neoantigens: a pan-cancer analysis," *IScience*, vol. 21, pp. 249–260, 2019.

[39] T.-C. Chang, R. A. Carter, Y. Li, Y. Li, H. Wang, M. N. Edmonson, X. Chen, P. Arnold, T. L. Geiger, G. Wu *et al.*, "The neoepitope landscape in pediatric cancers," *Genome medicine*, vol. 9, pp. 1–12, 2017.

[40] Y. Tang, Y. Wang, J. Wang, M. Li, L. Peng, G. Wei, Y. Zhang, J. Li, and Z. Gao, "Truneo: an integrated pipeline improves personalized true tumor neoantigen identification," *BMC Bioinformatics*, vol. 21, 11 2020.

[41] A. J. Rech, D. Balli, A. Mantero, H. Ishwaran, K. L. Nathanson, B. Z. Stanger, and R. H. Vonderheide, "Tumor immunity and survival as a function of alternative neopeptides in human cancer," *Cancer immunology research*, vol. 6, no. 3, pp. 276–287, 2018.

[42] Y. Wang, T. Shi, X. Song, B. Liu, and J. Wei, "Gene fusion neoantigens: Emerging targets for cancer immunotherapy," *Cancer Letters*, vol. 506, pp. 45–54, 2021.

[43] D. Rieder, G. Fotakis, M. Ausserhofer, R. Geyeregger, W. Paster, Z. Trajanoski, and F. Finotello, "nextneopi: a comprehensive pipeline for computational neoantigen prediction," pp. 1131–1132, 2022. [Online]. Available: https://doi.org/10.1093/bioinformatics/btab759

[44] K. Diao, J. Chen, T. Wu, X. Wang, G. Wang, X. Sun, X. Zhao, C. Wu, J. Wang, H. Yao, C. Gerarduzzi, and X.-S. Liu, "Seq2neo: a comprehensive pipeline for cancer neoantigen immunogenicity prediction," *Int J Mol Sci*, vol. 23, no. 19, 2022.

[45] J. Kodysh and A. Rubinsteyn, "Openvax: An open-source computational pipeline for cancer neoantigen prediction," *Methods in molecular biology*, vol. 2120, pp. 147–160, 2020. [Online]. Available: https://doi.org/10.1007/978-1-0716-0327-7_10

[46] N. Patwardhan, S. Marrone, and C. Sansone, "Transformers in the real world: A survey on nlp applications," *Information*, vol. 14, no. 4, p. 242, 2023.

[47] J. Cheng, K. Bendjama, K. Rittner, and B. Malone, "Bertmhc: improved mhc–peptide class ii interaction prediction with transformer and multiple instance learning," *Bioinformatics*, vol. 37, no. 22, pp. 4172–4179, 2021.

[48] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, P. Chen, J. Canny, P. Abbeel, and Y. Song, "Evaluating protein transfer learning with tape," *Advances in neural information processing systems*, vol. 32, 2019.

[49] H.-C. Gasser, G. Bedran, B. Ren, D. Goodlett, J. Alfaro, and A. Rajan, "Interpreting bert architecture predictions for peptide presentation by mhc class i proteins," *arXiv preprint arXiv:2111.07137*, 2021.

[50] F. Wang, H. Wang, L. Wang, H. Lu, S. Qiu, T. Zang, X. Zhang, and Y. Hu, "Mhcroberta: pan-specific peptide–mhc class i binding prediction through transfer learning with label-agnostic protein sequences," *Briefings in Bioinformatics*, vol. 23, no. 3, p. bbab595, 2022.

[51] R. Vita, S. Mahajan, J. A. Overton, S. K. Dhanda, S. Martini, J. R. Cantrell, D. K. Wheeler, A. Sette, and B. Peters, "The immune epitope database (iedb): 2018 update," *Nucleic acids research*, vol. 47, no. D1, pp. D339–D343, 2018.

[52] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger *et al.*, "Prottrans: Toward understanding the language of life through self-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 10, pp. 7112–7127, 2021.

[53] V. M. A. Machaca, "Neoantigen detection using transformers and transfer learning in the cancer immunology context," in *International Conference on Practical Applications of Computational Biology & Bioinformatics*. Springer, 2023, pp. 97–102.

[54] Y. Chu, Y. Zhang, Q. Wang, L. Zhang, X. Wang, Y. Wang, D. R. Salahub, Q. Xu, J. Wang, X. Jiang *et al.*, "A transformer-based model to predict peptide–hla class i binding and optimize mutated peptides for vaccine design," *Nature Machine Intelligence*, vol. 4, no. 3, pp. 300–311, 2022.

[55] Y. Jing, S. Zhang, and H. Wang, "Dapnet-hla: Adaptive dual-attention mechanism network based on deep learning to predict non-classical hla binding sites," *Analytical Biochemistry*, vol. 666, p. 115075, 2023.

[56] M. Nielsen, M. Andreatta, B. Peters, and S. Buus, "Immunoinformatics: predicting peptide–mhc binding," *Annual Review*, 2020.

[57] S. Mei, F. Li, A. Leier, T. T. Marquez-Lago, K. Giam, N. P. Croft, T. Akutsu, A. I. Smith, J. Li, J. Rossjohn *et al.*, "A comprehensive review and performance evaluation of bioinformatics tools for hla class i peptide-binding prediction," *Briefings in bioinformatics*, vol. 21, no. 4, pp. 1119–1135, 2020.

[58] M. Wang, L. Kurgan, and M. Li, "A comprehensive assessment and comparison of tools for hla class i peptide-binding prediction," *Briefings in Bioinformatics*, p. bbad150, 2023.

[59] V. E. Machaca, V. Goyzueta, M. Cruz, and Y. Tupac, "Deep learning and transformers in mhc-peptide binding and presentation towards personalized vaccines in cancer immunology: A brief review," in *International Conference on Practical Applications of Computational Biology & Bioinformatics*. Springer, 2023, pp. 14–23.

[60] N. Hashemi, B. Hao, M. Ignatov, I. C. Paschalidis, P. Vakili, S. Vajda, and D. Kozakov, "Improved prediction of mhc-peptide binding using protein language models," *Frontiers in Bioinformatics*, vol. 3, 2023.

[61] M. Kalemati, S. Darvishi, and S. Koohi, "Capsnet-mhc predicts peptide-mhc class i binding based on capsule neural networks," *Communications Biology*, vol. 6, no. 1, p. 492, 2023.

[62] Z. Ye, S. Li, X. Mi, B. Shao, Z. Dai, B. Ding, S. Feng, B. Sun, Y. Shen, and Z. Xiao, "Stmhcpan, an accurate star-transformer-based extensible framework for predicting mhc i allele binding peptides," *Briefings in Bioinformatics*, vol. 24, no. 3, p. bbad164, 2023.

[63] C. Chen, Z. Qiu, Z. Yang, B. Yu, and X. Cui, "Jointly learning to align and aggregate with cross attention pooling for peptide-mhc class i binding prediction," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2021, pp. 18–23.

[64] Y. Ye, J. Wang, Y. Xu, Y. Wang, Y. Pan, Q. Song, X. Liu, and J. Wan, "Mathla: a robust framework for hla-peptide binding prediction integrating bidirectional lstm and multiple head attention mechanism," *BMC bioinformatics*, vol. 22, no. 1, pp. 1–12, 2021.

[65] Z. Liu, J. Jin, Y. Cui, Z. Xiong, A. Nasiri, Y. Zhao, and J. Hu, "Deepseqpanii: an interpretable recurrent neural network model with attention mechanism for peptide-hla class ii binding prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

[66] X. Yang, L. Zhao, F. Wei, and J. Li, "Deepnetbim: deep learning model for predicting hla-epitope interactions based on network analysis by harnessing binding and immunogenicity information," *BMC bioinformatics*, vol. 22, no. 1, pp. 1–16, 2021.

[67] J. Jin, Z. Liu, A. Nasiri, Y. Cui, S.-Y. Louis, A. Zhang, Y. Zhao, and J. Hu, "Deep learning pan-specific model for interpretable mhc-i peptide binding prediction with improved attention mechanism," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 7, pp. 866–883, 2021.

[68] Z. Chen, M. R. Min, and X. Ning, "Ranking-based convolutional neural network models for peptide-mhc class i binding prediction," *Frontiers in Molecular Biosciences*, vol. 8, p. 634836, 2021.

[69] G. Venkatesh, A. Grover, G. Srinivasaraghavan, and S. Rao, "Mhcattnnet: predicting mhc-peptide bindings for mhc alleles classes i and ii using an attention-based deep neural model," *Bioinformatics*, vol. 36, no. Supplement_1, pp. i399–i406, 2020.

[70] J. Wu, W. Wang, J. Zhang, B. Zhou, W. Zhao, Z. Su, X. Gu, J. Wu, Z. Zhou, and S. Chen, "Deephlapan: a deep learning approach for neoantigen prediction considering both hla-peptide binding and immunogenicity," *Frontiers in Immunology*, p. 2559, 2019.

[71] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli *et al.*, "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, no. 6637, pp. 1123–1130, 2023.

[72] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart *et al.*, "The pfam protein families database in 2019," *Nucleic acids research*, vol. 47, no. D1, pp. D427–D432, 2019.

[73] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium, "Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 2015.

[74] U. Consortium *et al.*, "Uniprot: the universal protein knowledgebase," *Nucleic acids research*, vol. 46, no. 5, p. 2699, 2018.

[75] R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas *et al.*, "The pfam protein families database: towards a more sustainable future," *Nucleic acids research*, vol. 44, no. D1, pp. D279–D285, 2016.

[76] U. Consortium, "Uniprot: a worldwide hub of protein knowledge," *Nucleic acids research*, vol. 47, no. D1, pp. D506–D515, 2019.

[77] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[78] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, 2021.

[79] A. Merchant, E. Rahimtoroghi, E. Pavlick, and I. Tenney, "What happens to bert embeddings during fine-tuning?" *arXiv preprint arXiv:2004.14448*, 2020.

[80] J. Lee, R. Tang, and J. Lin, "What would elsa do? freezing layers during transformer fine-tuning," *arXiv preprint arXiv:1911.03090*, 2019.

[81] O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky, "Revealing the dark secrets of bert," *arXiv preprint arXiv:1908.08593*, 2019.

[82] R. Anil, B. Ghazi, V. Gupta, R. Kumar, and P. Manurangsi, "Large-scale differentially private bert," *arXiv preprint arXiv:2108.01624*, 2021.

[83] Y. Zhang, Y. Han, S. Cao, G. Dai, Y. Miao, T. Cao, F. Yang, and N. Xu, "Adam accumulation to reduce memory footprints of both activations and gradients for large-scale dnn training," *arXiv preprint arXiv:2305.19982*, 2023.

[84] Z. Huang, B. Jiang, T. Guo, and Y. Liu, "Measuring the impact of gradient accumulation on cloud-based distributed training," in *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2023, pp. 344–354.

[85] M. Mosbach, M. Andriushchenko, and D. Klakow, "On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines," *arXiv preprint arXiv:2006.04884*, 2020.

[86] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[87] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.