Allele-conditional attention mechanism for HLA-peptide complex binding affinity prediction

Rodrigo Hormazabal[*], **Doyeong Hwang**[*], **Kiyoung Kim**,
Sehui Han, Kyunghoon Bae, Honglak Lee
LG AI Research
{rodrigo, doyeong.hwang, elgee.kim, hansse.han
k.bae, honglak}@lgresearch.ai

Abstract

The Human Leukocyte Antigen (HLA) complex plays a crucial role in adaptive immune responses for cancer immunology. Due to the complex interactions governing the binding process of peptides in the HLA surface and the intrinsic polymorphic nature of the HLA complex, one of the main bottlenecks for cancer vaccine design is the accurate prediction of binding epitopes for specific alleles. Data-driven approaches using binding experiments' information have shown to be effective for high-throughput screening of candidates instead of expensive docking methods. However, there is still no consensus on how to most effectively represent amino acid sequences and model the long interaction patterns present in these complexes. Recently, attention-based models have been explored to improve this task, allowing for higher flexibility by introducing weaker inductive biases into the models, however, carrying a critical trade-off between expressiveness and data efficiency. We propose an allele-conditional attention mechanism for binding prediction and show how constraining the attention between the HLA context and peptide sequences improves performance while requiring fewer parameters than standard transformer-like models. We thoroughly study the impact of different attention schemes and pooling methods on the task of binding affinity prediction and benchmark widely utilized deep learning architectures. In addition, we show that patterns in string representation space can provide insights and encode information that correlates with the underlying spatial interactions between HLA class I and peptide amino acids without any extra docking simulations.

1 Introduction

The Human leukocyte antigen (HLA), the human version of the Major Histocompatibility Complex (MHC), plays an important role in adaptive immunity. HLA class I molecules mainly bind to cleaved and transported peptides with a length between 8 to 10 amino acids derived from protein fragments in cells, subsequently presenting them on the cell surface. Cytotoxic T cells release cytotoxins when they recognize certain peptides, epitopes, presented by HLA class I molecules on the surface of abnormal cells such as cancer cells[1]. Epitopes produced by genetic mutations on cancer cells, called neoepitopes, can be used to develop personalized cancer vaccines (Figure 1). Even though there is a lot recent work in personalized cancer vaccines, it is still challenging to identify neoepitopes from each cancer patient's specific HLA genotype and mutations[2].

Over the past decade, the performance of HLA binding prediction approaches has advanced considerably through the use of deep learning algorithms and, the increase data availability on open-source database such as the *"Immune Epitope Database"* (IEDB)[3]. NetMHCPan[4; 5; 6] was the first artificial neural network approach to the task of MHC-peptide binding prediction, which was followed by various on newer architectures trying to exploit regularities in amino acid sequences, such as

[*]These authors contributed equally to this work.

Machine Learning for Structural Biology Workshop, NeurIPS 2022.

recurrent neural networks (RNN)[7] an convolution neural networks (CNN)[8; 9]. Recently, models have seen further improvements by introducing attention mechanisms[10; 11; 12] and the use of large pretrained self-supervised models such as Evolutionary Scale Modeling (ESM)[13] and Bidirectional Encoder Representations from Transformers (BERT)[14]. However, with the increasing number of studies exploring the use of attention mechanisms to represent the residue-level interactions between protein and peptide, there is still not consensus on what is the most parameter-efficient way of encoding these interactions, balancing the trade-offs between attentional context and data needs.

In this study, we use propose a simplified conditional attention mechanism that uses peptide information to query and attend on the HLA structure, outperforming traditional more complex concatenated or cross-attention approaches. We provide a unified implementation and dataset to analyze and compare across different attention schemes, allow for fair performance assessment between different approaches. We derive a context-informed peptide structure position importance within the HLA sequences by deriving the importance of each amino-acid within the peptide and HLA sequences at each position through the attention weights, comparing the differences in strong and weak binding affinity. In addition, we also explore the learned weights in the cross-attention heads and show how peptide amino acids query information from the HLA sequences, following spatial patterns without any explicit docking information.

2 Background

2.1 Personalized cancer vaccines

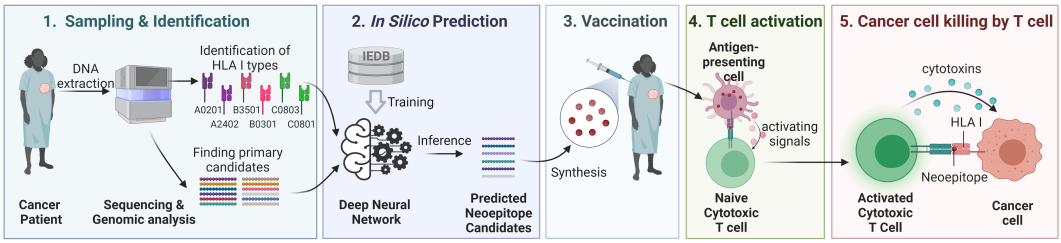

Figure 1: Process and mechanism of personalized cancer vaccine therapy.

Personalized cancer vaccine therapy is a promising approach to help trigger T-cell immune responses in the human body[15]. As shown in Figure 1, the first step in cancer vaccine design is to identify the patient's HLA genotype and mutations in cancer through sequencing and genomic analysis of normal and cancer genome. Primary neoepitope candidates are short peptides with at least one amino acid variation derived from cancer genome, since cytotoxic T cells can recognize only non-self epitopes[16]. Subsequently, these candidates can be scored and prioritized by predicting their HLA-specific binding affinity and their ability to induce immune responses through the use of models trained with experimental data. The top dozens of candidates are then synthesized to make a vaccine and injected into the patient's body. Subsequently, naive cytotoxic T cells that recognize these neoepitopes are activated by Antigen-presenting cells in lymph nodes. These activated T cells recognize HLA-neoepitope complexes, which in turn can lead to the destruction of cancer cells.

2.2 Binding affinity prediction

In order to assess the performance of binding affinity prediction architectures, we benchmark several models widely used in previous works (RNN and CNN-based) and different types of transformer-like attention mechanisms. Task specific details, such as dataset, featurization and the preprocessing pipeline utilized in this work are detailed below.

IEDB Database The raw MHC class I binding affinity data was obtained from IEDB[3]. IEDB is an open-source resource which catalogs experimental data on antibody and epitopes. Available binding experimental data for HLA-peptide pairs includes discrete classification labels (e.g, `[positive-high, positive-mid, positive-low, negative]`), continuous concentration values (commonly in nM scale) and the related experiment specific information. In our benchmarks, only data points containing concentration values were used. Our curation process pipeline is detailed in Appendix A, which follows guidelines gathered from previous works and adds modifications to ensure mostly reliable assays are retained.

Featurization Three featurization schemes to represent of protein/peptide amino acid sequences were tested to explore their impact on the models' performance: Learned token embeddings (`LE`), *Amino acid index database* (AA Index)[17] combined with BLOSUM[18] matrix (`AA`), and pre-trained protein-representation embeddings, such as the *"Evolutionary Scale Modeling "* (`ESM`)[19]. Pretrained representations for proteins were only used to represent HLA sequences. LE refers to randomly initialized embedding layers that map each amino acid token to a dense representations. AA index features represent various amino acids' physico and biochemical properties. Due to the fact that these descriptors are highly correlated with each other, we only use the first principal components in combination with the BLOSUM62 matrix concatenated to each amino acid token representation. `ESM` features are extracted from the pretrained Transformer protein language model developed by Rives et al.[19], specifically we use the `esm-1b` weights trained with the UniRef50[20] database. Both AA and ESM representations are projected with a dense layer to match the hidden dimension of the transformer models. ESM representations are only used for HLA, since more complex long range interactions are more likely to occur on longer amino acid sequences.

3 Method

Model architecture and training There are several approaches to account for the interaction between multiple sequences, in this case `[HLA] + [peptide]`. Previous methods either; (1) concatenate both sequences together using a special `[SEP]` token and then apply self-attention to the concatenated sequence or (2) apply a cross attention mechanism between both sequences. However, generally HLA sequences are much longer than peptides, while also having less variability in terms of structures, ergo carrying less information to determine binding affinity. Also, we normally look to find good peptide candidates that could server as neoepitopes, while keeping the HLA-alleles as targets. Followed by these intuitions, we therefore propose an architecture that uses HLA sequences as a context that conditions the interactions of peptides and the protein surface. First, a representation for peptide sequences is generated through self attention layers, which is then conditionally enhanced by attending on the protein sequence information. In this way peptide representations can sparsely query the HLA sequence at the most important locations. It's important to note that peptide sequences could also choose to ignore the impact of proteins through the use of the skip connections available in the attention layers. We show that this not only keeps HLA representations more stable while training (by not directly modifying is underlying embeddings), but allows for the network to focus its capacity on the peptide representation and how it "queries" different parts of the HLA sequence. Intuitively, this also allows for the peptide to implicitly find binding sites within the HLA sequence, which can help estimating the binding affinity.

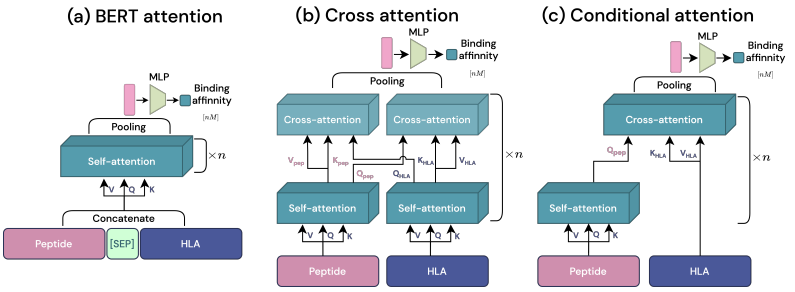

Figure 2: Architectures of attention-based models

Model construction variations We benchmark three attention-based models, which detailed architecture can be seen on Figure 2. Also, a GRU-based model and a CNN-based model are reported as baselines. In the proposed allele-conditional attention model, peptide sequences go through a self-attention where amino acids g. Output sequences from self-attention then attend to HLA sequences to acquire meaningful interactions in peptide-HLA binding. HLA-aware peptide token representations from the cross attention module are then pooled and projected to MLP layers for prediction. Following the architecture by Cheng et. al.[14], BERT-attention model applies self-attention on the concatenated peptide-HLA sequences, where the output sequences are then pooled and used for prediction. As for the cross-attention model, we implement the attention module presented by Chen et. al.[11], where self-attention is applied on each amino acid sequence separately and then cross-attention is applied on each counterpart, for both peptide and HLA. sequences. Peptide and HLA sequences

from cross-attention are pooled respectively, concatenated and projected through MLP layers. For pooling the context-aware attention outputs, we use learnable weights to aggregate sequence into a fixed length. We also conduct experiments across different methods of pooling on (Appendix A.1).

GRU[7] and CNN[21] models encode peptide and HLA sequences with bidirectional-GRU and CNN, respectively. Outputs of GRU/CNN-based encoders are concatenated and projected to MLP layers. Details with the architecture of GRU and CNN models are reported on Appendix A.2.

4 Results

Metrics Following previous work [11][14][12], in conjunction with standard regression metrics such as MAE, we report Spearman's correlation coefficients hoping to shed some light on the performance of each architecture across the whole distribution of epitopes. Mean absolute error(MAE) is reported to give a more intuitive indicator of the mean deviation from the predictions. To better compare with traditional classification-based approaches for binding affinity prediction, we also include AUC-ROC and AUC-PR scores, calculated with a threshold set to `500 nM` (standard threshold used in previous works citechen2021jointly). Lastly, we report the mean binding affinity of `top@100` & `top@50` instances queried by their predicted affinity values as a ranking score. This metric aligns well with real setups where only a few candidates can be tested experimentally, and false positives are costly. `top@K` affinity highlights the ability of each model robustly rank peptides without making mistakes due to overconfidence.

Table 1: Performances of model architectures across type of input embeddings. All entries use learnable weights for pooling sequences.

		Regression metrics		**Classification metrics**		**Ranking metrics**	
Featurization scheme	**Model**	**Spearman's rank correlation ρ ↑**	**Mean Absolute error (MAE)↓**	**AUC-ROC↑**	**AUC-PR↑**	**Average BA@TOP100↑**	**Average BA@TOP50↑**
Peptide: **LE** HLA: **LE**	GRU	0.688 ± 0.027	0.160 ± 0.019	0.854 ± 0.027	0.782 ± 0.032	0.877 ± 0.038	0.901 ± 0.036
	CNN	0.681 ± 0.014	0.164 ± 0.011	0.854 ± 0.020	0.784 ± 0.045	0.887 ± 0.077	0.919 ± 0.007
	BERT att.	0.711 ± 0.053	0.172 ± 0.017	0.870 ± 0.033	0.799 ± 0.072	0.857 ± 0.056	0.844 ± 0.048
	Cross att.	0.714 ± 0.032	0.199 ± 0.025	0.868 ± 0.020	0.795 ± 0.057	0.874 ± 0.058	0.911 ± 0.036
	Conditional att.	0.735 ± 0.027	0.161 ± 0.012	0.881 ± 0.018	0.811 ± 0.060	0.882 ± 0.053	0.900 ± 0.036
Peptide: **AA** HLA: **AA**	GRU	0.714 ± 0.010	0.153 ± 0.013	0.868 ± 0.018	0.802 ± 0.042	0.921 ± 0.054	0.959 ± 0.035
	CNN	0.684 ± 0.019	0.164 ± 0.011	0.855 ± 0.022	0.786 ± 0.043	0.879 ± 0.055	0.904 ± 0.052
	BERT att.	0.709 ± 0.025	0.200 ± 0.016	0.865 ± 0.019	0.790 ± 0.060	0.855 ± 0.032	0.863 ± 0.019
	Cross att.	0.722 ± 0.034	0.224 ± 0.018	0.875 ± 0.025	0.802 ± 0.056	0.842 ± 0.065	0.811 ± 0.079
	Conditional att.	0.746 ± 0.025	0.162 ± 0.011	0.885 ± 0.017	0.817 ± 0.053	0.900 ± 0.059	0.940 ± 0.064
Peptide: **AA** HLA: **ESM**	GRU	0.631 ± 0.058	0.175 ± 0.017	0.827 ± 0.036	0.748 ± 0.062	0.855 ± 0.074	0.888 ± 0.082
	CNN	0.676 ± 0.020	0.166 ± 0.012	0.851 ± 0.023	0.782 ± 0.043	0.873 ± 0.065	0.925 ± 0.076
	BERT att.	0.713 ± 0.035	0.188 ± 0.025	0.871 ± 0.021	0.802 ± 0.057	0.876 ± 0.037	0.913 ± 0.027
	Cross att.	0.727 ± 0.037	0.222 ± 0.032	0.874 ± 0.020	0.804 ± 0.056	0.839 ± 0.072	0.835 ± 0.053
	Conditional att.	**0.773 ± 0.025**	**0.144 ± 0.013**	**0.899 ± 0.019**	**0.840 ± 0.044**	**0.942 ± 0.052**	**0.974 ± 0.046**

Performance analysis As shown in Table 1, the proposed allele-conditional attention outperforms its counterparts, BERT-attention and Cross-attention, across all featurization schemes. This is likely due to the cost of learning a full attention set on both combined sequences, even though most of the variability happens only over peptides. Also, HLA sequences tend to show differences in a few positions across alleles, keeping most of their structure consistent. Concatenating both sequences before contextualizing tokens (BERT) or after (cross-attention) leads to peptides being relatively underrepresented over the full sequences. Thus, changes in the peptide representation have weaker effects on the final output activations. In contrast, allele-conditional attention results in an output length equal to the peptide sequence length, which is an order of magnitude shorter than protein sequences. This implies that information carried in peptide changes is more directly represented in the output activations. Furthermore, this scheme reduces computational complexity since the sequence length becomes much shorter after cross-attention.

GRU and CNN-based models show competitive performance on ranking and regression metrics compared to other transformer-based models. However, as spearman's correlation coefficients hint, this is likely due to these models focusing on common and not correctly capturing relative binding strengths across peptides. In contrast, conditional attention tends to perform better on these metrics, showing improvements in relative affinity and ranking. The performance gains of AA+ESM over other

featurization schemes(LE+LE, AA+AA) are most significant in the conditional-attention approach, achieving the best scores for all reported metrics. In addition, our allele-conditional attention scheme only attends to some positions of HLA sequences necessary for prediction. This lets the model effectively leverage richer information acquired from the pre-trained protein language model in HLA sequences, compared to GRU and CNN-based models.

Attention-weighted peptide positions analysis Per-position amino acid distribution is commonly reported on HLA-peptide binding affinity datasets for positive and negative samples. These give some insights into the priors for each amino acid at all positions. However, this is not very informative to assess the impact of amino acids at specific positions since these distributions have no contextual information, neither from the peptide itself nor the alleles. After models are trained, attention scores can be obtained for each peptide amino acid, which can then be used to calculate a new context-aware distribution for the amino acids weighted by the attention scores (Figure 3). These attention-weighted distributions can be calculated for positive and negative samples on the test set. Also, the difference between these results highlights amino acids at certain positions within the peptide sequences suspected to be crucial for binding affinity interactions. We also report the highlighted amino acids within the HLA sequences in Figure 6 in Appendix A.2.

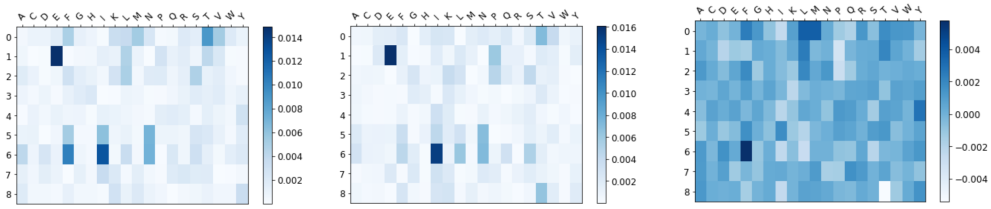

Figure 3: Attention weighted amino acid distribution for peptide sequences. These plots highlight positions that relate to positive binding (left), negative binding (middle), and positions that relatively show the highest correlation with the type of interaction (right). Calculated over entire dataset, global pattern extraction. `6F`, `0L`, `0M` stronger towards positive binding, and `8T`, `1P`, `2P`, `6L` stronger towards negative binding.

Attention correlation with distance in 3D space We analyzed the learned attention weights of the allele-conditional attention network and found several heads that align well with the HLA-peptide complex 3D structure. A 3D visualization of some attention heads is shown in Figure 4 for a single protein example (HLA0201: 7EU2, measured using x-ray diffraction [22]). In Figure 4 it can be seen that these heads point towards the vicinities of certain peptide positions, which hints that just from pure sequence information, implicit structural patterns can be found between nearby amino acids in space, with any docking information.

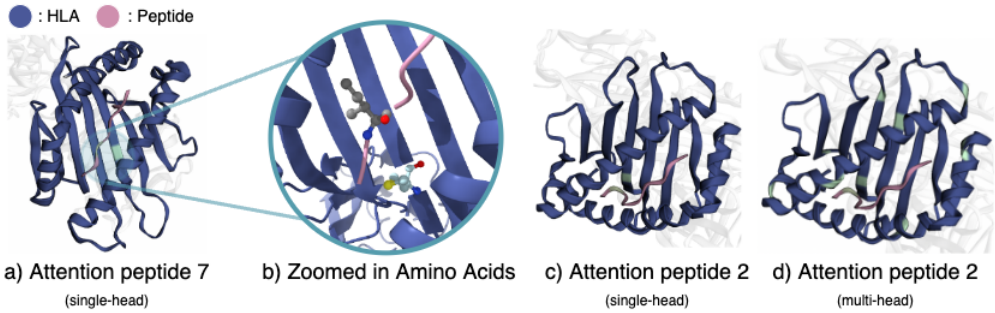

Figure 4: Visualization of allele-conditional attention heads for peptide positions two and seven in 3D space. Protein amino acids highlighted by cyan color are thresholded at 0.7. This shows that attention weights capture implicit information regarding 3D proximity and learn how to attend to different parts of the 3D HLA structure depending on peptide positions.

5 Discussion

Following our attention weights analysis, it can be seen that it might be beneficial to explicitly take into account the 3D structure within the model. This has been attempted before [23], however data is generally sparse on docking information for the peptide and HLA complex. A pretraining approach on individual sequences that learn a better representation for docking prediction together with a predictive model on binding affinity might create a more efficient solution.

Further work can also be extended to HLA class II or immunogenicity prediction [24]. There are in the end multiple processes that determine the effectiveness of a peptide, that might all jointly have to be considered when generating peptide candidates. The attention analysis and work presented here can hopefully guide further developments into these different areas.

References

[1] J. Rossjohn, S. Gras, J. J. Miles, S. J. Turner, D. I. Godfrey, and J. McCluskey, "T cell antigen receptor recognition of antigen-presenting molecules," *Annual review of immunology*, vol. 33, pp. 169–200, 2015.

[2] C. A. Brennick, M. M. George, W. L. Corwin, P. K. Srivastava, and H. Ebrahimi-Nik, "Neoepitopes as cancer immunotherapy targets: key challenges and opportunities," *Immunotherapy*, vol. 9, no. 4, pp. 361–371, 2017.

[3] 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, 2019.

[4] M. Nielsen, C. Lundegaard, T. Blicher, K. Lamberth, M. Harndahl, S. Justesen, G. Røder, B. Peters, A. Sette, O. Lund *et al.*, "Netmhcpan, a method for quantitative predictions of peptide binding to any hla-a and-b locus protein of known sequence," *PloS one*, vol. 2, no. 8, p. e796, 2007.

[5] M. Nielsen and M. Andreatta, "Netmhcpan-3.0; improved prediction of binding to mhc class i molecules integrating information from multiple receptor and peptide length datasets," *Genome medicine*, vol. 8, no. 1, pp. 1–9, 2016.

[6] 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.

[7] Y. Heng, Z. Kuang, S. Huang, L. Chen, T. Shi, L. Xu, and H. Mei, "A pan-specific gru-based recurrent neural network for predicting hla-i-binding peptides," *ACS omega*, vol. 5, no. 29, pp. 18 321–18 330, 2020.

[8] T. J. O'Donnell, A. Rubinsteyn, M. Bonsack, A. B. Riemer, U. Laserson, and J. Hammerbacher, "Mhcflurry: open-source class i mhc binding affinity prediction," *Cell systems*, vol. 7, no. 1, pp. 129–132, 2018.

[9] Y. S. Vang and X. Xie, "Hla class i binding prediction via convolutional neural networks," *Bioinformatics*, vol. 33, no. 17, pp. 2658–2665, 2017.

[10] 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.

[11] 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.

[12] 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.

[13] N. Hashemi, B. Hao, M. Ignatov, I. Paschalidis, P. Vakili, S. Vajda, and D. Kozakov, "Improved predictions of mhc-peptide binding using protein language models," *bioRxiv*, 2022.

[14] 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.

[15] C. S. Shemesh, J. C. Hsu, I. Hosseini, B.-Q. Shen, A. Rotte, P. Twomey, S. Girish, and B. Wu, "Personalized cancer vaccines: clinical landscape, challenges, and opportunities," *Molecular Therapy*, vol. 29, no. 2, pp. 555–570, 2021.

[16] Y. Takahama, "Journey through the thymus: stromal guides for t-cell development and selection," *Nature Reviews Immunology*, vol. 6, no. 2, pp. 127–135, 2006.

[17] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa, "Aaindex: amino acid index database, progress report 2008," *Nucleic acids research*, vol. 36, no. suppl_1, pp. D202–D205, 2007.

[18] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks." *Proceedings of the National Academy of Sciences*, vol. 89, no. 22, pp. 10 915–10 919, 1992.

[19] 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, p. e2016239118, 2021.

[20] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu, "Uniref: comprehensive and non-redundant uniprot reference clusters," *Bioinformatics*, vol. 23, no. 10, pp. 1282–1288, 2007.

[21] Z. Liu, Y. Cui, Z. Xiong, A. Nasiri, A. Zhang, and J. Hu, "Deepseqpan, a novel deep convolutional neural network model for pan-specific class i hla-peptide binding affinity prediction," *Scientific reports*, vol. 9, no. 1, pp. 1–10, 2019.

[22] H. Zhang, S. Deng, L. Ren, P. Zheng, X. Hu, T. Jin, and X. Tan, "Profiling cd8+ t cell epitopes of covid-19 convalescents reveals reduced cellular immune responses to sars-cov-2 variants," *Cell reports*, vol. 36, no. 11, p. 109708, 2021.

[23] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher, and N. F. Rajani, "Bertology meets biology: interpreting attention in protein language models," *arXiv preprint arXiv:2006.15222*, 2020.

[24] G. Li, B. Iyer, V. S. Prasath, Y. Ni, and N. Salomonis, "Deepimmuno: deep learning-empowered prediction and generation of immunogenic peptides for t-cell immunity," *Briefings in bioinformatics*, vol. 22, no. 6, p. bbab160, 2021.

[25] A. T. Nguyen, C. Szeto, and S. Gras, "The pockets guide to hla class i molecules," *Biochemical Society Transactions*, vol. 49, no. 5, pp. 2319–2331, 2021.

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

A Appendix

A.1 Additional results

Pooling variations For BERT, Cross and Conditional-attention models, three pooling schemes were considered to combine the information from the amino acid contextualized representations: average pooling, learned weighted pooling, and a class token pooling. Average pooling averages a sequence of token representations sequence-wise. Learned weighted pooling uses a learnable set of weights to aggregate sequences into a fixed length. Class token pooling adds a [CLS] at the start of each sequence. One extra attention layer is added on top of the main encoder and the [CLS] tokens are used for prediction.

Table 2: Performances of model architectures across type of input embeddings. All entries use learnable weights for pooling sequences.

		Regression metrics		Classification metrics		Ranking metrics	
Attention scheme	**Model**	**Spearman's rank correlation ρ ↑**	**Mean Absolute error (MAE) ↓**	**AUC-ROC ↑**	**AUC-PR ↑**	**Average BA@TOP100 ↑**	**Average BA@TOP50 ↑**
Cross attention	`[CLS]` token	0.721 ± 0.022	0.184 ± 0.014	0.871 ± 0.020	0.803 ± 0.048	0.883 ± 0.052	0.913 ± 0.034
	Mean pooling	0.711 ± 0.037	0.195 ± 0.019	0.870 ± 0.018	0.796 ± 0.069	0.875 ± 0.067	0.893 ± 0.067
	Learned weighting	0.727 ± 0.037	0.222 ± 0.032	0.874 ± 0.020	0.804 ± 0.056	0.839 ± 0.072	0.835 ± 0.053
BERT attention	`[CLS]` token	0.652 ± 0.052	0.195 ± 0.018	0.840 ± 0.033	0.760 ± 0.055	0.831 ± 0.020	0.849 ± 0.033
	Mean pooling	0.706 ± 0.023	0.185 ± 0.024	0.870 ± 0.020	0.798 ± 0.052	0.871 ± 0.065	0.893 ± 0.076
	Learned weighting	0.713 ± 0.035	0.188 ± 0.025	0.871 ± 0.021	0.802 ± 0.057	0.876 ± 0.037	0.913 ± 0.027
Conditional attention	`[CLS]` token	0.725 ± 0.041	0.163 ± 0.017	0.877 ± 0.027	0.813 ± 0.034	0.882 ± 0.053	0.898 ± 0.079
	Mean pooling	0.744 ± 0.027	0.164 ± 0.010	0.885 ± 0.014	0.816 ± 0.060	0.904 ± 0.051	0.941 ± 0.040
	Learned weighting	$\mathbf{0.773 \pm 0.025}$	$\mathbf{0.144 \pm 0.013}$	$\mathbf{0.899 \pm 0.019}$	$\mathbf{0.840 \pm 0.044}$	$\mathbf{0.942 \pm 0.052}$	$\mathbf{0.974 \pm 0.046}$

As can be seen in Table 2, Conditional-attention model outperforms BERT-attention and Cross-attention model across different types of pooling mechanisms. Among pooling types, learned weighted pooling shows the best performance across different attention architectures.

Classification vs regression labels: converting labels Generally, there are two main approaches for binding affinity prediction: classification and regression-based. On the one hand, classification tasks typically use either binary labels or binned affinity categories (positive-low, positive-intermediate, positive-high, negative). On the other hand, regression tasks include experimental values for binding affinity (typically in the range of 0-50000nM), which indicate how well the peptides bind to HLA proteins. A regression-based approach thus can give a better learning signal of the experimental binding affinity, allowing for better ranking of the final predicted peptide candidates compared to classification-based methods that optimize for categorical predictions.

A.2 Training details

Dataset curation process The whole sequence length of HLA class I molecules is approximately 365 amino acids; however, the alpha chain involved in epitope binding is about 300 residues long. In particular, binding residues are known to be distributed between positions from the 5th to 171st of the alpha chain[25]. Therefore, We extract sequences from positions 2 to 182 of the HLA class I alpha chains, which denote amino acids that are structurally close to binding residues. The dataset used in this work was extracted directly from the "Immune Epitope Database" (IEDB)[3], which was then curated with the following steps: i) assays with only HLA class I (`'HLA-A'`, `'HLA-B'`, `'HLA-C'`) remain. ii) keep only peptides with lengths between 8 to 14 residues iii) peptides with non-amino acid tokens, e.g., `'X'`, `'B'` are removed. iv) allowed MHC types/assay methods/assay techniques are purified MHC, cellular MHC/competitive, direct/radioactivity, fluorescence. v) according to assay references, instances generated through simulations are removed. The curated dataset is then split into five folds, where each fold is stratified with a number of assays sharing the same HLA.

Hyper-parameter settings and training details We used `AdamW`[26] optiimzer for all experiments. `ReduceLROnPlateau` scheduler was used for CNN and GRU models, while the other three attention-

based models used cosine annealing with warmup due to the necessity of a warmup regime when training transformer-like models. Most of our experiments were focused on the regression task, since on real-world scenarios ranking peptides is a priority to select promising candidates.

We tested random searched over a fixed range of hyper-parameters settings for all models and report the settings of the best performing attention-based, GRU- and CNN-based models. Architecutre related hyper-parameters were set to make comparisons as fair as possible in terms of model capacity. For maximum learning rate in the attention-based models, we report better performances in the range between 0.0005, 0.001.

Table 3: Performances of model architectures across type of input embeddings. All entries use learnable weights for pooling sequences.

	Hyperparameter	**Type**	**Value**
Common hyperparameters	Dropout	`float`	0.3
	Activation layers	`class`	`ReLU`
	Hidden dimension	`int`	256
	Mean pooling	`int`	1024
	Training epochs	`int`	1500
	Weight decay	`float`	$1{\cdot}10^{-8}$
GRU & CNN specific	Number of layers	`int`	3
	Scheduler	`class`	`ReduceLROnPlateau`
	Initial LR	`float`	$5{\cdot}10^{-4}$
	Minimum LR	`float`	$1{\cdot}10^{-7}$
	WD factor	`float`	0.5
	Patience	`int`	20
Attention-based specific	Number of layers	`int`	4
	Number of heads	`int`	4
	Scheduler	`class`	`CosineAnnealingWarmRestarts`
	Max LR	`float`	$\{5{\cdot}10^{-4}, 1 \cdot 10^{-3}\}$
	Warm-up epochs	`int`	750

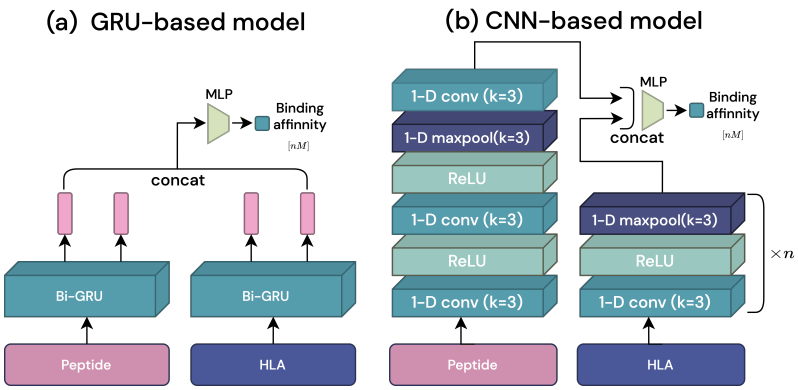

Figure 5: Architectures of GRU/CNN-based models

GRU/CNN-based architectures details GRU-based model uses Bi-GRU model to encode peptide and HLA sequences, respectively. Output vectors from the start/end of the peptide/HLA Bi-GRU models are concatenated and projected through MLP layers to return predictions. The CNN-based model uses 1-D convolutional and max-pool layers with `ReLU` activations to encode peptide and HLA sequences, respectively. Each CNN encoder returns a vector, which is concatenated and projected through MLP layers. Detailed architecture of the GRU- and CNN-based models can be found in Figure 5.

A.3 Attention-weighted HLA positions analysis

Per-position residue distributions weighted by the attention scores are also calculated over the amino acids in HLA proteins (Figure 6). Similar to Figure 3, highlighted amino acids at specific positions within the peptide sequence are perceived to be relatively important. The plots highlight the top 15 positions derived from the attention weight analysis for both positive and negative binding affinity for each amino acid. (thresholded at 0.15).

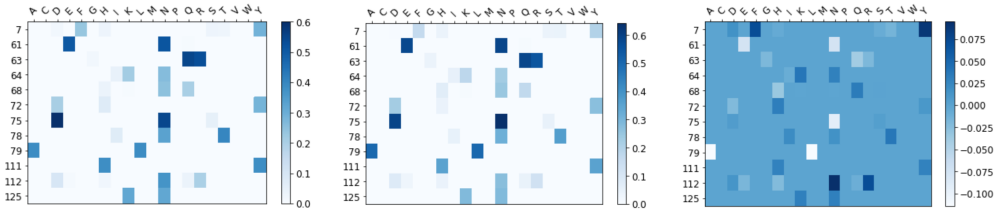

Figure 6: Attention weighted amino acid distribution for peptide sequences. These plots highlight positions that relate to positive binding (left), negative binding (middle), and positions that relatively show the highest correlation with the type of interaction (right). Calculated over entire dataset, global pattern extraction. 6F, 0L, 0M stronger towards positive binding, and 8T, 1P, 2P, 6L stronger towards negative binding.

A.4 Implementation

All implementations and shell scripts needed to train and reproduce the results shown in this work can be found at https://github.com/LGMILab/allele_conditional_attention.