Towards Personalized Cancer Immunotherapy: A Deep Learning Approach for Tumor-Specific T Cell Receptor Discovery

Sara Farmahini Farahani
Student ID: 7016100

Supervisors:
Prof. Dr. Volkhard Helms
Prof. Dr. Sebastian Vollmer

Advisors:
Dr. Gerrit Großmann
Dr. Maximilian Sprang

German Research Center for Artificial Intelligence
Department of Bioinformatics
Saarland University

A thesis submitted in fulfillment of the requirements for the degree
Master of Science in Bioinformatics

February 2026

I would like to dedicate this thesis to my family and friends.

Declaration

I, **Sara Farmahini Farahani**, declare that this thesis titled, *"Towards Personalized Cancer Immunotherapy: A Deep Learning Approach for Tumor-Specific T Cell Receptor Discovery"* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help and I used ChatGPT to correct lingual errors.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.
- I hereby confirm the congruence of the contents of the printed data and the electronic version of the thesis.

Signed: _____

Date: _____

Sara Farmahini Farahani
Student ID: 7016100
February 2026

Acknowledgements

I would like to acknowledge my supervisors, Prof. Dr. Volkhard Helms and Prof. Dr. Sebastian Vollmer, for overseeing this thesis.

I am especially grateful to my advisors, Dr. Gerrit Großmann and Dr. Maximilian Sprang, for their guidance, constructive discussions, technical support, and continuous assistance throughout the development of this work.

I also thank the German Research Center for Artificial Intelligence (DFKI) and Saarland University for providing the academic environment and computational resources necessary for this research.

Finally, I would like to thank my family for their support during my studies.

Abstract

Personalized cancer immunotherapy aims to harness a patient's immune system to selectively eliminate tumor cells. A central challenge in this approach is the recognition of tumor-specific mutations, known as *neoantigens*, by T cell receptors (TCRs). Accurately predicting TCR–neoantigen interactions remains a substantial computational challenge in the design of individualized therapies.

Recent deep learning models have reported strong performance in TCR–peptide binding prediction. However, systematic benchmarking has revealed a persistent generalization gap: models often perform well on peptides seen during training but degrade when evaluated on previously unseen peptides. This limitation is especially critical in the neoantigen setting, where each patient presents a largely unique mutational landscape. In addition, common negative sampling strategies—particularly random or mismatched sampling—can introduce label ambiguity, especially when biologically plausible cross-reactive peptides are assigned as negatives.

In this thesis, we systematically examine how supervision design, model architecture, and evaluation strategy influence robustness. We introduce a similarity-aware controlled negative sampling framework based on peptide clustering, explicitly avoiding the assignment of near-neighbor peptides as negatives. Compared to conventional sampling approaches, this supervision-aligned strategy is associated with improved generalization under peptide-level distribution shift.

Our backbone model, a Transformer architecture using AAIndex-based encoding, enables us to study the interaction between representation learning and structured supervision. Under strict hard-split evaluation (completely unseen peptides) and structurally defined RMSD-based splits, the proposed framework outperforms NetTCR-2.0 and ERGO-II, improving AUROC by approximately 10–11 percentage points in the evaluated settings. Notably, the improvements are most pronounced under structural separation, consistent with increased robustness under structural distribution shift.

Finally, we apply the framework to a clinically motivated case study in non-small cell lung cancer (NSCLC). By integrating somatic mutation catalogs with MHC binding prediction, we rank candidate TCR sequences under patient-specific constraints. Although these predictions remain model-dependent and require structural and functional validation, this work provides a supervision-aware and structurally stress-tested computational framework for TCR–neoantigen prioritization in personalized immunotherapy.

Code and pipeline implementations are available at:
https://github.com/SaraFarmahini/TCR_neo

Contents

Chapter 1
Introduction

1.1 Clinical Motivation

Personalized cancer immunotherapy aims to redirect a patient's own immune system toward tumor-specific targets, replacing broadly cytotoxic treatments with precision immune interventions [55]. Approaches such as adoptive cell transfer (ACT) and T cell receptor (TCR)-engineered therapies seek to expand or engineer cytotoxic T cells capable of selectively recognizing malignant cells [54, 17]. At a mechanistic level, T cells survey the body by using their TCRs to recognize short peptide fragments presented on the cell surface by major histocompatibility complex (MHC) molecules. Therapeutic efficacy therefore depends on identifying peptide targets that can be recognized with high specificity while sparing healthy tissue [55].

A key constraint in personalized immunotherapy is target selection. Therapeutic T cells must recognize cancer cells with high specificity while avoiding damage to normal tissues. Early strategies focused on tumor-associated antigens (TAAs), which are enriched in malignant cells but are not exclusively tumor-specific [66, 11]. Because these antigens are derived from normal host proteins, therapies directed against them carry a risk of cross-reactivity and immune-mediated toxicity [53, 45]. These limitations underscore an important design principle: effective immunotherapy requires molecular targets that are both immunogenic and strictly tumor-restricted. Consequently, the field has increasingly shifted toward identifying antigens that arise uniquely in cancer cells [28, 62].

Neoantigen peptides arising from tumor-specific somatic mutations represent such targets. By definition, they are absent from normal tissues and therefore evade central tolerance [74]. As non-self entities, neoantigens can elicit high-affinity T cell responses and have been associated with durable tumor regression in multiple clinical settings [74]. Consequently, neoantigen-directed immunotherapy has emerged as a promising framework for truly personalized cancer treatment.

The neoantigen discovery pipeline can be formulated as a multi-stage filtering process. Tumor sequencing identifies somatic mutations, which are translated into candidate mutant peptides. Peptide–MHC binding predictors, such as NetMHCpan, are then used

to prioritize peptides likely to be presented on the tumor cell surface [49]. Although this substantially reduces the candidate space, MHC presentation alone does not guarantee immunogenicity, as only a small fraction of presented peptides are recognized by T cells. The subsequent identification of TCRs capable of selectively binding a given neoantigen remains experimentally labor-intensive and low-throughput [41].

This final step—predicting TCR–neoantigen recognition—represents the primary computational challenge addressed in this thesis. Formally, the problem can be framed as a sequence-pair interaction prediction task: given a TCR sequence and a candidate peptide sequence, determine whether the pair forms a specific binding interaction. In the clinical setting, two constraints arise. First, models must generalize beyond peptides observed during training, as each patient presents a unique mutational landscape. Second, in the neoantigen context, models must distinguish between mutant and wild-type peptides that may differ by only a single amino acid substitution. This constitutes a fine-grained sequence discrimination problem under limited supervision. Consequently, data construction and evaluation design strongly influence whether measured performance reflects molecular generalization rather than peptide memorization.

Addressing these challenges requires modeling strategies that reduce reliance on peptide overlap and instead promote generalization across sequence space, together with evaluation frameworks that assess performance on previously unseen peptides and mutation-level discrimination tasks.

1.2 The Biological Challenge

Modeling TCR–peptide recognition is challenging due to the combinatorial diversity of the TCR repertoire. Each TCR consists of two protein chains, the α and β chains, which together form the antigen-binding interface that contacts peptide–MHC complexes. Both chains contribute to specificity, and receptor identity depends on their paired combination.

The sequences encoding these chains are generated during T cell development through a stochastic recombination process known as V(D)J recombination. In this process, discrete gene segments—called Variable (V), Diversity (D), and Joining (J) segments—are randomly selected and assembled to form the variable region of the receptor. For the β chain, one V, one D, and one J segment are combined; for the α chain, one V and one J segment are joined. Additional sequence variability is introduced at the junctions between these segments through random nucleotide insertions and deletions. Because the α and β chains are generated independently and subsequently paired, receptor diversity expands combinatorially [16].

Through the combined effects of segment recombination, junctional diversification, and chain pairing, the immune system can theoretically generate on the order of 10^{14}–10^{19} distinct TCR sequences [71] (Figure 1.1). Although only a small fraction of this theoretical space is realized in any individual, each person still harbors hundreds of millions of unique T cells at a given time [71].

This immense diversity implies that TCR–peptide recognition occurs within a vast, high-dimensional sequence space that cannot be exhaustively sampled experimentally. The problem can therefore be formulated as predicting interactions between two variable-length biological sequences under extreme combinatorial complexity.

Machine learning provides a scalable framework for addressing this challenge. Advances

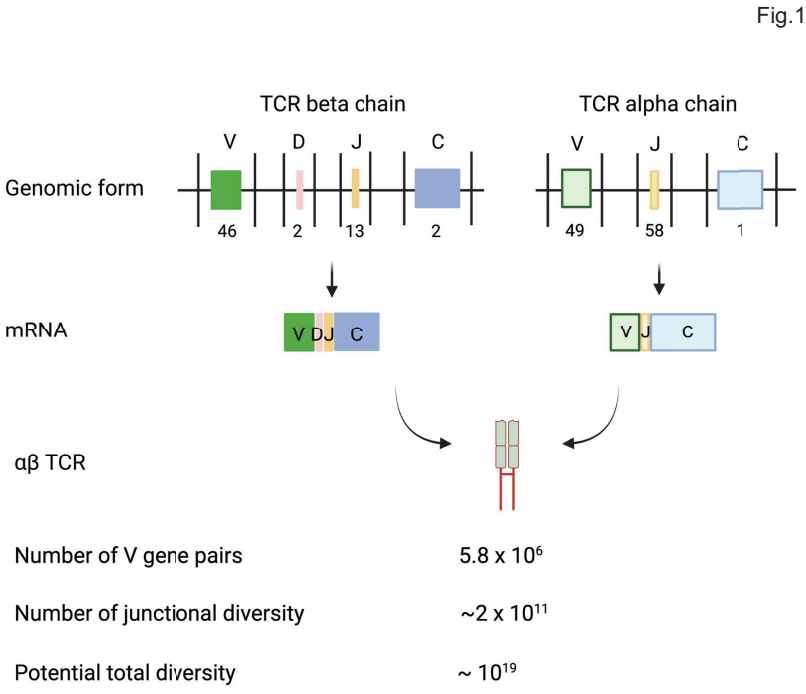

Figure 1.1. Schematic illustration of V(D)J recombination and the generation of TCR diversity. Adapted from [71].

in high-throughput repertoire sequencing and single-cell technologies have generated rapidly growing public databases of paired TCR–epitope interactions [41]. These datasets enable the training of deep learning models that attempt to infer binding specificity directly from sequence information. The central hypothesis is that such models can identify transferable sequence features within TCR complementarity-determining regions and peptide epitopes that govern molecular recognition. However, the task remains difficult due to the multifactorial nature of TCR binding and the limited and uneven coverage of epitopes in currently available datasets.

1.3 Problem Statement

1.3.1 The Generalization Gap: Unseen Peptides

Current TCR–peptide prediction models often show reduced performance when evaluated on previously unseen peptides. The task is typically formulated as a binary sequence-pair classification problem: given a TCR sequence and a peptide sequence, predict whether the pair forms a binding interaction.

Many existing evaluations rely on random train–test splits in which peptides appearing in the test set may also be present during training. Under such conditions, models often achieve strong performance. However, when evaluation protocols enforce peptide-level separation—commonly referred to as *hard splits*, where no peptide in the test set appears in the training set—performance typically degrades substantially [74, 20].

This behavior suggests that models may rely on peptide-specific signals rather than learning representations that generalize across sequence space. The problem can be interpreted as a distribution shift setting, in which the model is evaluated on peptide sequences not observed during training. In personalized immunotherapy, where each patient presents a distinct set of mutation-derived peptides, robust generalization to unseen peptides is required.

1.3.2 The Challenge of Clinical Specificity

In addition to generalization, clinical application requires selective targeting of tumor-specific antigens. Neoantigens arise from somatic mutations and are not encoded in the germline genome [29]. Because they differ from their wild-type counterparts, they may escape central tolerance and be recognized as non-self [9].

Therapeutic safety requires discrimination between mutant and corresponding wild-type peptides. A clinically viable TCR should recognize the tumor-derived mutant peptide while avoiding binding to the closely related wild-type peptide present in healthy tissue. In many cases, the two sequences differ by only a single amino acid substitution. Failure to distinguish between them can result in severe off-target toxicity [45]. The clinically relevant task therefore extends beyond general binding prediction to mutation-sensitive discrimination under minimal sequence variation.

Structural studies illustrate why this requirement is difficult. TCR specificity is often governed by a small number of critical contact residues at the binding interface. Even minor atomic-level differences—such as a single side-chain modification—can alter the geometry or energetics of interaction and determine whether recognition occurs [50, 12] (Figure 1.2).

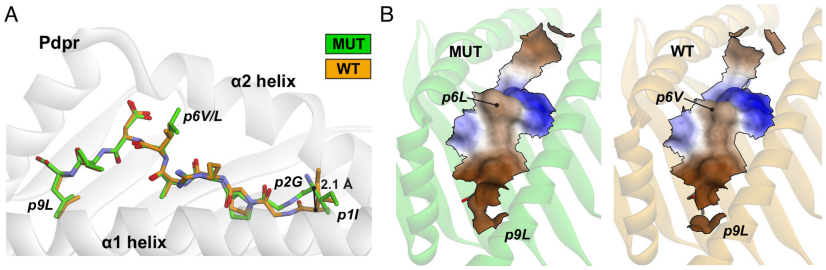

Figure 1.2. Structural illustration of how a single amino acid mutation can alter TCR recognition by modifying key contact interactions at the binding interface. Adapted from [12].

1.3.3 Limitations of Random Negative Sampling

A further methodological challenge concerns how negative training examples are constructed. In supervised learning, models require both positive (binding) and negative (non-binding) examples. Because experimentally validated non-binding TCR–peptide pairs are scarce, many studies generate negatives by randomly pairing TCRs with unrelated peptides [13, 30, 57].

Random negative construction can introduce systematic bias. Because mismatched pairs are frequently highly dissimilar from true binders, models may achieve high performance

by relying on coarse sequence-level differences rather than learning interaction-relevant features [20, 14].

Importantly, such random sampling rarely includes near-identical mutant–wild-type peptide pairs. As a result, models are not trained to resolve the fine-grained distinctions required for clinical safety. Performance under random negative sampling may therefore overestimate real-world applicability [40, 10].

1.3.4 Scope and Assumptions: MHC Restriction

In vivo, TCRs recognize peptides presented by MHC molecules on the cell surface. Computational prediction of peptide–MHC binding has become a relatively mature field, with highly accurate tools available [49].

In this thesis, we assume that candidate peptide–MHC complexes have already been identified using established predictors. We focus instead on modeling the sequence-intrinsic interaction between the TCR—specifically its complementarity-determining region 3 of the β chain (CDR3β), the most variable region involved in peptide contact—and the peptide sequence itself.

This MHC-agnostic abstraction isolates the most computationally challenging component of the problem: peptide-specific TCR recognition under stringent generalization and mutation-sensitive specificity constraints.

Taken together, these considerations motivate evaluation frameworks that explicitly enforce peptide-level generalization and modeling strategies that prioritize fine-grained discrimination over coarse memorization.

1.4 Positioning Within the Personalized Immunotherapy Pipeline

Personalized cancer immunotherapy involves a multi-stage pipeline that integrates tumor genomics, antigen prediction, immune screening, and cellular engineering [29, 62, 46]. The process begins with tumor biopsy and whole-exome sequencing to identify somatic mutations unique to the patient's malignancy. These mutations are translated into candidate mutant peptides, which are subsequently evaluated for their ability to bind patient-specific MHC molecules using established predictors such as NetMHCpan [49]. This step substantially reduces the search space by prioritizing peptides with high presentation likelihood.

However, MHC binding alone is insufficient to guarantee immunogenicity. Only a small fraction of presented neoepitopes elicit productive T cell responses [29]. Consequently, the subsequent and more challenging stage of the pipeline involves identifying TCR capable of recognizing these candidate neoantigens with sufficient specificity and affinity. Experimental approaches for TCR discovery, including multimer staining and high-throughput screening, remain labor-intensive, low-throughput, and costly [4]. This step therefore constitutes a bottleneck in the translation of genomic data into actionable immunotherapeutic strategies.

The computational task addressed in this thesis targets this bottleneck. Assuming that candidate peptide–MHC complexes have already been identified through established presentation predictors, we model the sequence-intrinsic interaction between the TCR

CDR3β region and the peptide. The objective is to prioritize TCR sequences with high predicted specificity for a given neoantigen while discriminating against recognition of the corresponding wild-type peptide.

By improving generalization to previously unseen peptides and introducing mutation-aware fine-tuning strategies, the proposed framework aims to reduce reliance on exhaustive experimental screening. In practical terms, the model functions as a prioritization engine within the immunotherapy pipeline, narrowing the candidate TCR search space and guiding downstream validation efforts. While it does not replace experimental confirmation, it provides a computational layer that enhances efficiency and reduces the combinatorial burden inherent in neoantigen-specific TCR discovery.

1.5 Contributions

This thesis makes the following contributions to the computational study of TCR–peptide interaction prediction:

- **Systematic evaluation under strict generalization settings.** We conduct a structured benchmarking of multiple deep learning architectures for TCR–peptide binding prediction using unseen-peptide (hard split) evaluation protocols. This evaluation explicitly assesses model generalization beyond peptide memorization, addressing a central limitation of prior studies that rely on random data splits.
- **Analysis and mitigation of negative sampling bias.** We analyze the influence of negative sampling strategies on reported model performance and show that purely random negative sampling can lead to overly optimistic estimates of predictive accuracy. To address this issue, we introduce biologically motivated hard negative examples that more closely reflect realistic non-binding scenarios, thereby improving the robustness and interpretability of model evaluation.
- **Sequence-based modeling framework focused on CDR3β–peptide interactions.** We develop and evaluate a sequence-only modeling framework that isolates the intrinsic interaction between the TCR CDR3β region and antigenic peptides, independent of upstream peptide–MHC binding prediction. This design mirrors current clinical pipelines and enables focused investigation of the most challenging component of TCR specificity prediction.
- **Neoantigen-specific fine-tuning strategy using wild-type peptide controls.** We propose a neoantigen-oriented fine-tuning strategy in which experimentally supported neoantigen–TCR pairs are contrasted directly against their corresponding wild-type peptides derived from healthy proteins. This setup explicitly targets the clinically relevant task of discriminating tumor-specific mutations from self-antigens.
- **Application to a clinically relevant NSCLC case study.** We apply the proposed framework to a non-small cell lung cancer (NSCLC) case study, demonstrating its ability to prioritize TCR CDR3β sequences with high predicted specificity for NSCLC-associated neoantigens. This application illustrates the practical utility of the approach for downstream experimental prioritization and hypothesis generation.

Together, these contributions establish a rigorous and transparent framework for the development and evaluation of sequence-based TCR–peptide binding models under realistic generalization scenarios, and offer practical guidance for improving robustness in neoantigen-specific prediction tasks.

1.6 Thesis Organization

This thesis is organized into five chapters, each addressing a distinct aspect of TCR–peptide interaction prediction, ranging from biological motivation and methodological foundations to model development, evaluation, and application.

1.6.1 Chapter 1: Introduction and Motivation

This chapter presents the biological and clinical motivation for TCR–peptide interaction prediction in the context of personalized cancer immunotherapy. It introduces the relevant immunological background, formulates the central challenge of generalization to unseen peptides, discusses key methodological issues such as negative sampling bias, and outlines the scope and contributions of this thesis.

1.6.2 Chapter 2: Background and Related Work

This chapter reviews existing computational approaches for TCR–epitope and TCR–peptide binding prediction. It covers traditional sequence-based methods, recent deep learning models, and commonly used evaluation protocols, with particular emphasis on generalization performance, negative sampling strategies, and limitations in neoantigen-focused settings. Key gaps in the current state of the art that motivate the proposed work are identified.

1.6.3 Chapter 3: Proposed Approach

This chapter describes the datasets used in this study, including data sources, preprocessing steps, and sequence encoding strategies. It details the modeling frameworks and training procedures employed for TCR–peptide binding prediction, introduces the negative sampling strategies, and explains the evaluation protocols designed to assess model generalization under unseen-peptide conditions.

1.6.4 Chapter 4: Results and Discussion

This chapter presents the experimental setup and results of the systematic benchmarking study. Model performance is evaluated under different data split strategies and negative sampling schemes using appropriate quantitative metrics. Comparisons with existing methods are provided, followed by a focused analysis of the neoantigen-specific fine-tuning strategy and its application to an NSCLC case study.

1.6.5 Chapter 5: Conclusion and Future Work

The final chapter summarizes the main findings and contributions of the thesis and discusses their implications for computational modeling of TCR specificity. Limitations of the proposed approach are examined, and directions for future research are outlined, including extensions to richer biological contexts and integration with downstream experimental validation pipelines.

Chapter 2
Background and Related Work

2.1 Biological Foundations of TCR Specificity

The adaptive immune system can be formulated as a complex recognition problem [59]. Unlike standard string-matching algorithms, TCR recognition is probabilistic and governed by physicochemical, structural, and thermodynamic constraints [6]. This section describes the structural organization of the TCR and the physical principles underlying neoantigen recognition [19, 43].

2.1.1 The Structural Hierarchy of the T-Cell Receptor

The TCR is a heterodimeric protein complex embedded in the plasma membrane. The vast majority of T cells involved in specific antigen recognition are $\alpha\beta$ T cells, consisting of two distinct polypeptide chains: the alpha (α) and beta (β) chains. Structurally, each chain is partitioned into a constant (C) domain, which acts as a structural anchor, and a variable (V) domain, which is responsible for antigen recognition [43]. The antigen-binding site is formed by the assembly of the Vα and Vβ domains. Rather than forming a planar surface, this site consists of a three-dimensional interface shaped by six CDR loops [43].

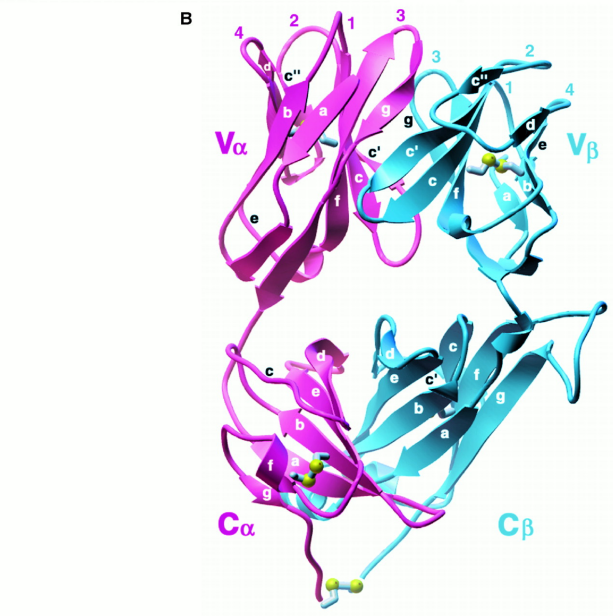

Figure 2.1. Backbone ribbon representation of the 2C TCR. The α chain is in pink (residues 1-213), and the β chain is in blue. This figure was taken and adapted from [18].

2.1.2 The Complementarity Determining Regions (CDRs)

The specificity of the TCR is localized almost entirely to the complementarity-determining region (CDR) loops. However, these loops do not contribute equally to the recognition event [58]:

- **Germline-encoded loops (CDR1 and CDR2):** These loops are encoded entirely by the V gene segments. Crystallographic studies show that they predominantly contact the conserved α-helices of the MHC molecule. In a predictive model, they primarily determine MHC restriction (the "docking frame") rather than peptide identity [58].

- **Hypervariable loop (CDR3):** The CDR3 loops (CDR3α and CDR3β) are generated somatically via V(D)J recombination. Positioned centrally over the peptide, they make the majority of energetic contacts with the antigen side chains. Consequently, the CDR3 loops dictate the peptide specificity of the TCR [72].

This functional separation motivates modeling approaches that prioritize CDR3 sequences when predicting peptide specificity.

2.1.3 Structural Dominance of the Beta Chain

Both TCR chains contribute to peptide–MHC binding. However, multiple studies indicate that the CDR3β loop provides strong predictive signal for antigen specificity [58]. This difference arises from combinatorial diversity: whereas the α chain undergoes VJ recombination, the β chain incorporates an additional diversity (D) segment during

VDJ recombination. This additional recombination step increases sequence entropy and length variability (6–23 amino acids), enabling the CDR3β loop to form a three-dimensional interface that frequently contacts central peptide residues [63]. For this reason, many predictive models, including the approach adopted in this thesis, focus on CDR3β as a principal sequence determinant of specificity.

2.1.4 MHC Restriction: The Presentation Platform

MHC class I molecules act as biological filters that select and display short peptide sequences on the cell surface. The binding groove is closed at both ends, restricting peptide length (typically 8–10 amino acids) and imposing positional constraints [47, 26].

Two functionally distinct residue categories can be defined:

- **Anchor residues**: Amino acids located at defined positions within the peptide (commonly P2 and the C-terminal position, PΩ) that insert into allele-specific pockets of the MHC binding groove. These residues determine peptide binding affinity and contribute to stable presentation. Their engagement with the groove constrains which peptides can be displayed on the cell surface [47, 26].
- **Epitope (TCR-facing) residues**: Amino acids that protrude from the MHC groove toward the solvent and are accessible to the TCR. These residues contribute minimally to MHC binding but are the primary determinants of TCR recognition and immunogenicity [8, 26].

Peptides longer than the canonical 8–10 amino acids often bulge centrally while remaining anchored at their termini, reinforcing the structural separation between MHC anchoring residues and TCR-facing peptide surfaces [26].

For machine learning, this distinction implies a natural decomposition: anchor residues encode allele-specific presentation constraints, whereas epitope residues encode interaction-relevant features for TCR specificity prediction.

2.1.5 The Neoantigen Distinction

A central objective of this thesis is to distinguish wild-type (WT) self-peptides from neoantigens. T cells are tolerized to self-derived peptides during thymic selection; therefore, a neoantigen must differ sufficiently from its wild-type counterpart to escape this tolerance and elicit an immune response [15].

A structural example is the HHAT–p8F neoepitope:

- **WT (KQWLVWLLL):** The leucine at position 8 permits a relatively flexible peptide conformation.
- **Neo (KQWLVWLFL):** Substitution with phenylalanine at position 8 introduces steric bulk, constraining the upstream tryptophan (P6) into a rigid conformation that enhances complementarity to the TCR [15].

Consistent with this mechanism, neoantigen immunogenicity has been associated with the degree of sequence divergence from the corresponding wild-type peptide [68]. Neoepitopes with limited divergence from self tend to show reduced T cell recognition, in line with tolerance-mediated constraints on self-reactive clones.

To quantify mutation-induced peptide divergence, three physicochemical descriptors were employed:

- **BLOSUM**: Sequence substitution severity was quantified using BLOSUM62-derived mutation penalties [24].
- **Aliphatic index**: Hydrophobic side-chain composition was measured via the aliphatic index as defined by Ikai [27]:

  $$AI = 100\,(x_A + 2.9x_V + 3.9(x_I + x_L))\,,$$

  reflecting the relative volume contribution of aliphatic residues.
- **Boman index**: Interaction propensity differences were captured using the Boman index, originally proposed to estimate protein–protein interaction potential [7].

Together, these descriptors provide complementary biochemical measures of neoantigen divergence from self.

Furthermore, predictive performance was shown to improve when focusing on the **Interaction Core (ICORE)**—the optimal MHC-binding submer—rather than the full-length peptide sequence [68]. As illustrated in Figure 2.2, restricting analysis to the peptide core that directly interfaces with the MHC and TCR enables more accurate modeling of sequence-intrinsic features governing antigen presentation and T cell recognition.

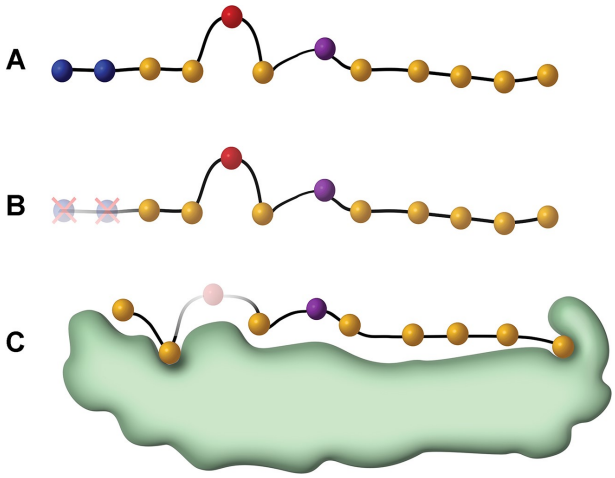

Figure 2.2. Schematic of the Interaction Core (ICORE) concept. The full peptide (A) undergoes processing to reveal a nested submer with optimal MHC-binding potential (B), which forms the core interface for TCR recognition (C). This figure was taken and adapted from [69].

These structural and physicochemical considerations indicate that TCR–peptide recognition cannot be reduced to simple sequence matching. Effective predictive models must account for localized changes in peptide structure and chemistry. These principles inform the modeling and evaluation framework described in Chapter 3.

2.2 Computational Foundations

Having established the biological determinants of TCR specificity, we now introduce the computational framework used to model these interactions. This involves defining sequence encodings, selecting model architectures, and specifying evaluation criteria.

2.2.1 Computational Representations of Sequences

Sequence encoding determines how amino acid information is represented for model input and therefore shapes the inductive biases available during learning. Different encoding schemes capture distinct aspects of amino acid similarity, including symbolic identity, evolutionary substitution patterns, and physicochemical properties.

In this thesis, five encoding strategies are evaluated, as illustrated in Figure 2.3. These approaches vary in representational density and biological prior integration, enabling comparison of how encoding choice influences TCR–peptide binding prediction.

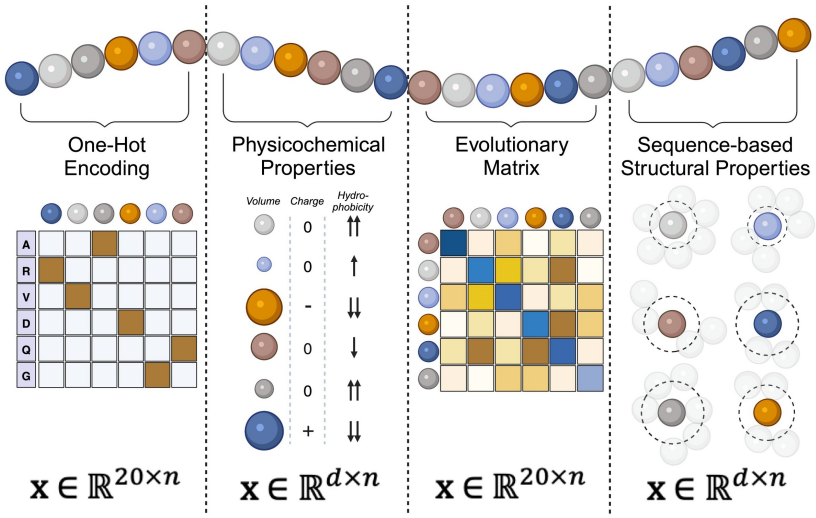

Figure 2.3. Overview of feature encoding strategies. Schematic comparison of the methods used to transform peptide sequences into numerical vectors: **(Left)** one-hot encoding creates sparse, orthogonal binary vectors; **(Middle-left)** physicochemical properties map residues to dense vectors based on intrinsic attributes such as volume and charge, corresponding to Atchley factors; **(Middle-right)** evolutionary matrices capture amino acid substitution probabilities, corresponding to BLOSUM encodings; **(Right)** sequence-based structural properties encode spatial features such as solvent accessibility and structural preference, corresponding to Kidera factors and solvent-accessible surface area (SASA) descriptors. This figure was taken and adapted from [23].

One-Hot Encoding

One-hot encoding is the simplest and most sparse representation of biological sequences, in which each amino acid is treated as a distinct categorical variable. In this scheme, the 20 standard amino acids are represented by binary vectors of length 20, with a single active entry indicating identity (e.g., Alanine = $[1, 0, \ldots, 0]$, Cysteine = $[0, 1, \ldots, 0]$), as illustrated

in Figure 2.3 (left). This representation encodes no evolutionary or physicochemical relationships between residues and therefore assumes orthogonality in the feature space. Deep learning architectures such as convolutional neural networks can, in principle, learn discriminative features from this representation. However, amino acids that are chemically similar (e.g., leucine and isoleucine) are encoded as equally dissimilar as chemically distinct pairs (e.g., leucine and arginine) [2].

BLOSUM Matrices (Evolutionary Information)

BLOSUM (BLOcks SUbstitution Matrix) encoding incorporates evolutionary information derived from observed substitution frequencies in homologous protein sequences [24]. BLOSUM62 assigns log-odds scores that quantify how frequently one amino acid replaces another relative to random expectation. These scores provide a measure of functional interchangeability between residues: positive values indicate substitutions observed more frequently than expected by chance, whereas negative values indicate substitutions that are rarely tolerated (Table 2.1).

Evolutionary substitution patterns reflect functional constraints on protein structure and stability. Amino acids that are frequently interchanged without loss of function often share similar physicochemical properties. Encoding peptide sequences using BLOSUM scores therefore embeds evolutionary constraints into the feature representation. A substitution with a high BLOSUM score indicates functional similarity and may preserve TCR recognition, whereas low-scoring substitutions are more likely to alter binding behavior [6]. This inductive bias is illustrated in Figure 2.3 (middle-right).

Table 2.1. The BLOSUM62 substitution matrix. Values represent the log-odds score of substituting one amino acid for another. Positive scores (e.g., diagonal values) indicate evolutionarily conserved replacements, while negative scores indicate unlikely substitutions. [24]

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9	−1	−1	−3	0	−3	−3	−3	−4	−3	−3	−3	−3	−1	−1	−1	−1	−2	−2	−2
S	−1	4	1	−1	1	0	1	0	0	0	−1	−1	0	−1	−2	−2	−2	−2	−2	−3
T	−1	1	5	−1	0	−2	0	−1	−1	−1	−2	−1	−1	−1	−1	−1	0	−2	−2	−2
P	−3	−1	−1	7	−1	−2	−2	−1	−1	−1	−2	−2	−1	−2	−3	−3	−2	−4	−3	−4
A	0	1	0	−1	4	0	−2	−2	−2	−1	−2	−1	−1	−1	−1	−1	−1	−2	−2	−3
G	−3	0	−2	−2	0	6	0	−1	−2	−2	−2	−2	−2	−3	−4	−4	−3	−3	−3	−2
N	−3	1	0	−2	−2	0	6	1	0	0	−1	−2	0	−2	−3	−3	−3	−3	−2	−4
D	−3	0	−1	−1	−2	−1	1	6	2	0	−2	−1	−1	−3	−3	−4	−3	−3	−3	−4
E	−4	0	−1	−1	−2	−2	0	2	5	2	−2	0	−1	−2	−3	−3	−2	−3	−2	−3
Q	−3	0	−1	−1	−1	−2	0	0	2	5	0	1	1	−1	−3	−3	−3	−3	−1	−2
H	−3	−1	−2	−2	−2	−2	−1	−2	−2	0	8	−1	−2	−1	−3	−3	−3	−1	2	−2
R	−3	−1	−1	−2	−1	−2	−2	−1	0	1	−1	5	−2	−3	−3	−3	−2	−3	−2	−3
K	−3	0	−1	−1	−1	−2	0	−1	−1	1	−2	−2	5	−1	−3	−2	−2	−3	−2	−3
M	−1	−1	−1	−2	−1	−3	−2	−3	−2	−1	−1	−3	−1	5	−2	−1	−2	−1	−1	−1
I	−1	−2	−1	−3	−1	−4	−3	−3	−3	−3	−3	−3	−3	−2	4	2	1	0	−1	−3
L	−1	−2	−1	−3	−1	−4	−3	−4	−3	−3	−3	−3	−2	−1	2	4	2	0	−1	−2
V	−1	−2	0	−2	−1	−3	−3	−3	−2	−3	−3	−2	−2	−2	1	2	4	−1	−1	−2
F	−2	−2	−2	−4	−2	−3	−3	−3	−3	−3	−1	−3	−3	−1	0	0	−1	6	3	1
Y	−2	−2	−2	−3	−2	−3	−2	−3	−2	−1	2	−2	−2	−1	−1	−1	−1	3	7	2
W	−2	−3	−2	−4	−3	−2	−4	−4	−3	−2	−2	−3	−3	−1	−3	−2	−2	1	2	11

AAIndex and Atchley Factors

Amino acid properties can be represented using numerical descriptors from the AAIndex database [31]. This resource includes over 500 indices derived from experimental

measurements and computational analyses, describing physicochemical attributes such as hydrophobicity, steric bulk, and thermodynamic stability.

Their dimensionality generally necessitates reduction or selective integration prior to model training.

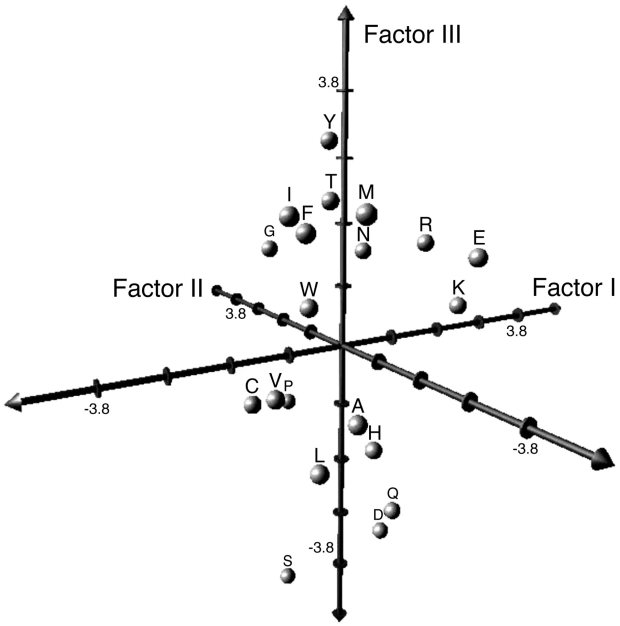

Figure 2.4. Plot of scores on Factors I–III for 20 amino acids. This figure was taken and adapted from [3].

Table 2.2. Five factor solution scores for the 54 selected amino acid attributes [3].

Amino acid	Factor I	Factor II	Factor III	Factor IV	Factor V
A	−0.591	−1.302	−0.733	1.570	−0.146
C	−1.343	0.465	−0.862	−1.020	−0.255
D	1.050	0.302	−3.656	−0.259	−3.242
E	1.357	−1.453	1.477	0.113	−0.837
F	−1.006	−0.590	1.891	−0.397	0.412
G	−0.384	1.652	1.330	1.045	2.064
H	0.336	−0.417	−1.673	−1.474	−0.078
I	−1.239	−0.547	2.131	0.393	0.816
K	1.831	−0.561	0.533	−0.277	1.648
L	−1.019	−0.987	−1.505	1.266	−0.912
M	−0.663	−1.524	2.219	−1.005	1.212
N	0.945	0.828	1.299	−0.169	0.933
P	0.189	2.081	−1.628	0.421	−1.392
Q	0.931	−0.179	−3.005	−0.503	−1.853
R	1.538	−0.055	1.502	0.440	2.897
S	−0.228	1.399	−4.760	0.670	−2.647
T	−0.032	0.326	2.213	0.908	1.313
V	−1.337	−0.279	−0.544	1.242	−1.262
W	−0.595	0.009	0.672	−2.128	−0.184
Y	0.260	0.830	3.097	−0.838	1.512

To address this limitation, Atchley factors provide a compact and interpretable representation of amino acid physicochemical space. Derived via factor analysis of 494 AAIndex descriptors, the five orthogonal factors capture the dominant variance in amino acid chemistry [3]. Table 2.2 and Figure 2.4 illustrate the organization of amino acids within this reduced factor space, revealing clustering patterns consistent with shared physicochemical properties.

Factor I reflects polarity and hydrophobicity, properties relevant for hydrophobic burial at the TCR–peptide interface. Factor II captures secondary structure propensity. Factor III encodes molecular volume and steric bulk, which influence geometric compatibility within the MHC binding groove. Factor IV relates to codon diversity and heat capacity, and Factor V represents electrostatic charge, contributing to salt-bridge formation and long-range interactions [3]. This encoding corresponds to the dense physicochemical representation shown in Figure 2.3 (middle-left).

Explicit Physicochemical Properties and Kidera Factors

Specific physicochemical attributes can also be selected based on biological hypotheses. Commonly used descriptors in TCR–peptide prediction include the Kyte–Doolittle hydrophobicity scale, isoelectric point (pI), and solvent-accessible surface area (SASA), each capturing a distinct aspect of amino acid behavior relevant to molecular recognition [34]. These descriptors correspond to the structural and surface-based encoding strategies illustrated in Figure 2.3 (right).

Kidera factors encode amino acids using ten orthogonal vectors derived from factor analysis of 188 physicochemical properties [32]. These factors describe structural preferences, including tendencies toward flat or extended conformations, which may influence

peptide behavior within the MHC binding groove.

Explicit physicochemical descriptors are especially informative in the context of neoantigen recognition, where immunogenicity often arises from small, localized changes in amino acid properties rather than large-scale sequence differences. In such cases, a single mutation that alters hydrophobicity, charge, or steric bulk at a TCR-facing residue may be sufficient to break immune tolerance and trigger recognition [15]. Encoding these properties directly allows models to capture such mechanistic effects and complements broader representations based on evolutionary or statistical similarity.

2.3 Negative Sampling and Label Noise in Interaction Prediction

Supervised learning for interaction prediction is complicated by the absence of experimentally validated true negatives. In TCR–peptide binding datasets, non-binding pairs are typically unobserved, and synthetic negatives are generated through random pairing [57, 30]. Random negative construction can include biologically plausible or highly similar peptide–TCR pairs, increasing the risk of label ambiguity. Such ambiguity may encourage models to rely on peptide frequency or coarse sequence patterns rather than learning interaction-relevant biochemical features [20]. Similarity-aware negative construction is therefore important for improving generalization under peptide-level distribution shift.

2.4 Transformer Architectures for Biological Sequences

Transformer architectures, originally introduced for natural language processing [65], are now widely used for modeling biological sequences. Unlike convolutional neural networks (CNNs), which primarily capture local motif patterns, or recurrent neural networks (RNNs), which process sequences sequentially, transformers use self-attention mechanisms that allow each residue in a sequence to attend to all other residues simultaneously. This global receptive field enables modeling of long-range dependencies, which are particularly relevant in protein and immune receptor sequences, where residues that are distant in primary sequence may contribute cooperatively to structural stability and functional specificity. Large-scale pretrained protein language models have further demonstrated the power of this architecture in biological domains. In particular, the evolutionary scale modeling (ESM) framework showed that transformer models trained on hundreds of millions of protein sequences can learn representations that implicitly encode structural, evolutionary, and functional information [51]. Such models have been shown to recover signals related to secondary structure, residue contacts, and mutational sensitivity without explicit structural supervision. This suggests that large-scale sequence modeling can capture features correlated with underlying biological constraints. Extensions such as the MSA Transformer further incorporate evolutionary information from multiple sequence alignments to enhance structural signal representation [48].

Recent work positions transformers as large-scale representation models in bioinformatics, capable of learning transferable sequence embeddings across tasks and protein families. For TCR–peptide binding prediction, this inductive bias is relevant because TCR recognition involves distributed interactions across complementarity-determining regions and peptide residues, including non-local dependencies that are difficult to cap-

ture with short convolutional filters. Self-attention provides a mechanism for modeling such long-range interaction patterns directly from sequence information.

2.4.1 Transfer Learning in Protein Sequence Modeling

Pretraining models on large biological sequence corpora using self-supervised objectives can capture general structural and evolutionary patterns that transfer to downstream tasks with limited labeled data [22]. In applications involving TCR–peptide specificity, transfer learning has been used to overcome the scarcity of labeled examples. For instance, the pMHC-TCR binding prediction network (pMTnet) employs a transfer learning framework to predict TCR binding specificity for neoantigens and other antigens using sequence data and class I MHC alleles, demonstrating improved predictive performance in settings where experimentally validated training labels are limited [39]. Models such as TABR-BERT similarly leverage pretrained representations to enhance generalization in epitope recognition tasks. These examples illustrate how transfer learning can stabilize training and enable task-specific adaptation when labeled neoantigen data are scarce, complementing purely supervised approaches.

2.5 Materials, Data Resources, and Computational Tools

This section summarizes the datasets, computational tools, and nomenclature conventions used throughout the experimental pipeline.

2.5.1 TCR–Peptide Interaction and Epitope Databases

The following resources provide experimentally validated TCR–peptide interaction data used for training, benchmarking, and evaluation:

- **VDJdb**: A database of TCR sequences with known antigen specificity, aggregating experimentally validated TCR–peptide–MHC interactions from published studies. It provides standardized CDR3 reporting and assigns confidence scores [56].
- **McPAS-TCR**: A manually compiled catalogue of pathology-associated TCR sequences, including epitopes derived from infectious pathogens and cancer [61].
- **NetTCR-2.0 Dataset**: Refers both to a deep learning prediction model and its associated training dataset. The dataset is widely used as a benchmark for TCR–peptide binding prediction [30, 44].
- **Immune Epitope Database (IEDB)**: A comprehensive repository of experimentally validated B-cell and T-cell epitopes across infectious disease, allergy, autoimmunity, and cancer contexts [67].
- **TChard Dataset**: A benchmark dataset introduced by Grazioli et al., integrating samples from IEDB, VDJdb, McPAS-TCR, and the NetTCR-2.0 repository. It was designed to evaluate peptide-shifted generalization under similarity-controlled splits [21, 20].
- **NeoTCR**: A public repository of neoantigen–TCR interaction pairs with mutation annotations, used in this thesis for mutation-aware fine-tuning and neoantigen-specific modeling [42].

- **COSMIC (Catalogue of Somatic Mutations in Cancer)**: A database of somatic mutations in human cancer. In this thesis, COSMIC is used to derive primary neoantigen candidates for the NSCLC case study [60].

2.5.2 MHC Binding Prediction and Structural Modeling Tools

- **NetMHCpan**: A state-of-the-art predictor of peptide binding affinity to MHC class I molecules. It reports an **EL rank** (ensemble learner percentile rank), where lower values indicate stronger predicted binding (e.g., EL rank ≤ 0.5 is commonly considered a strong-binder threshold) [49]. In this thesis, NetMHCpan is used within the NSCLC pipeline to filter candidate neoantigens based on predicted MHC binding strength.
- **ESMFold**: A deep learning–based protein structure prediction model that infers atomic-level structure from a single amino acid sequence using a pretrained protein language model [37]. Here, predicted Cα coordinates are used to compute structural similarity and to construct distance-based evaluation splits inspired by Grazioli et al.

2.5.3 Hyperparameter Optimization Framework

- **Optuna**: A hyperparameter optimization framework supporting define-by-run search spaces and automated pruning of unpromising trials to accelerate convergence [1].
- **Tree-structured Parzen Estimator (TPE)**: A sequential model-based optimization algorithm used as the default sampler in Optuna. TPE models the search space using kernel density estimation and selects new hyperparameter configurations by maximizing expected improvement. It is used in this thesis for hyperparameter optimization (HPO) [5].

2.5.4 Gene and Structural Nomenclature

- **TRB**: Refers to the TCR beta chain gene locus under IMGT/HUGO nomenclature. CDR3β sequences are derived from somatically rearranged TRB genes [36].
- **HLA (Human Leukocyte Antigen)**: The gene complex encoding human MHC class I and class II molecules. Allele names (e.g., HLA-A*02:01) follow standardized WHO nomenclature [52].
- **RMSD (Root Mean Square Deviation)**: A structural similarity metric measuring the average deviation between atomic coordinates after optimal superposition. In this thesis, RMSD is used to quantify structural distance between predicted peptide conformations.

2.6 Computational Models for TCR–Peptide Binding

A wide range of computational models have been proposed for predicting TCR–peptide interactions, differing in their choice of input representations, neural architectures, and

evaluation protocols. Early approaches relied on hand-crafted similarity measures and motif-based rules [44, 13], while more recent methods employ deep learning architectures to model nonlinear relationships between TCR sequences and peptide antigens [30, 57]. Common architectural paradigms include CNNs, RNNs, and hybrid models that combine sequence encoders with learned interaction functions [30, 58].

Most existing models share two central assumptions: the use of synthetically generated negative samples and evaluation under random or weakly constrained data splits [20, 70]. As a result, reported performance often reflects memorization of peptide identity rather than true generalization to unseen epitopes [20].

In the following subsections, we focus on two representative and widely used deep learning frameworks—NetTCR-2.0 and ERGO-II—which serve as reference baselines in this thesis and are evaluated under rigorous peptide-shifted protocols consistent with prior benchmarking studies [20].

2.6.1 ERGO and ERGO-II

ERGO (Enhanced Repertoire Generation and Optimization) and its extended variants are deep learning frameworks developed for predicting TCR–peptide binding from sequence data [57]. These models formulate TCR recognition as a supervised binary classification task, in which a TCR CDR3 sequence and a peptide sequence are jointly evaluated to determine binding likelihood.

In the original ERGO formulation, CDR3β sequences and peptides are encoded as ordered amino acid sequences and processed using recurrent neural networks, specifically long short-term memory (LSTM) architectures, to capture sequential dependencies along each sequence [57]. The latent representations of the TCR and peptide are subsequently concatenated and passed through fully connected layers to produce a binding probability.

Subsequent extensions of the ERGO framework investigated the contribution of additional immunological features, including the TCR α chain CDR3, V and J gene usage, and MHC typing, to peptide binding prediction [58]. These studies demonstrated that while incorporating auxiliary features can improve performance in some settings, the CDR3β sequence remains the dominant determinant of predictive signal.

A defining characteristic of ERGO-based models is their reliance on synthetically generated negative samples, typically constructed through random pairing of TCRs and peptides not observed to interact. Model training and evaluation in the original studies were primarily conducted using random data splits, in which peptide identities may overlap between training and test sets. This evaluation setup yields strong reported performance. However, later benchmarking work has shown that such protocols can overestimate generalization due to peptide memorization effects [20].

Due to their widespread adoption, clear architectural design, and strong baseline performance, ERGO-based models serve as representative recurrent neural network approaches in this thesis. They are therefore used as reference models in subsequent benchmarking analyses conducted under more stringent peptide-level and distance-based evaluation protocols.

2.6.2 NetTCR and NetTCR-2.0

NetTCR is a CNN-based framework originally proposed for predicting TCR–peptide interactions using sequence information [30]. NetTCR-2.0 represents a refined and ex-

tended version of this approach, incorporating architectural improvements and expanded training data to enhance predictive performance [44].

In NetTCR-2.0, TCR CDR3β sequences and peptide sequences are first transformed into fixed-length numerical representations and subsequently processed through multiple convolutional layers (Fig. 2.5). These convolutional filters act as local motif detectors, learning short, position-independent amino acid patterns that are informative for binding. The resulting feature maps are aggregated using pooling operations and passed through fully connected layers to produce a final binding probability.

The convolutional design of NetTCR-2.0 emphasizes local sequence patterns rather than long-range dependencies. This inductive bias reflects the hypothesis that short amino acid motifs within the CDR3β loop and the peptide contribute substantially to TCR–peptide interactions. As in ERGO-based models, training relies on synthetically generated negative samples, and evaluation in the original studies is conducted primarily using random data splits [30, 44].

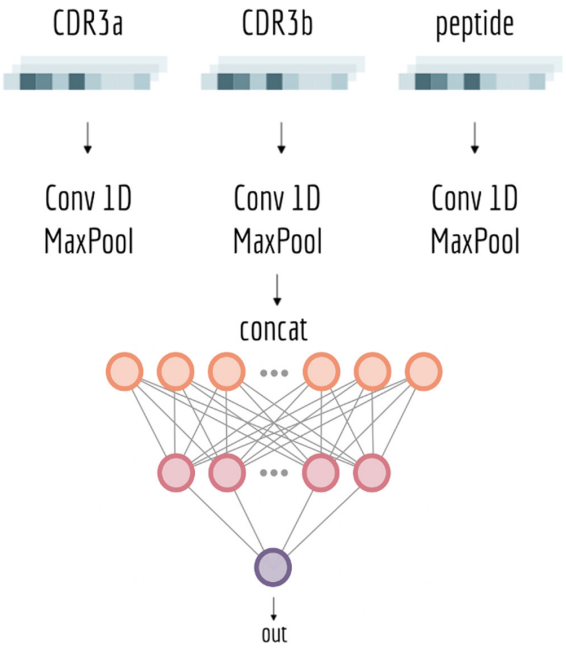

Figure 2.5. Schematic overview of the NetTCR-2.0 architecture. CDR3β and peptide sequences are encoded as fixed-length inputs and processed through convolutional layers that detect local sequence motifs. Feature maps are aggregated via pooling and combined in fully connected layers to predict TCR–peptide binding. This figure was taken and adapted from [44].

2.7 Evaluation Framework

Reliable assessment of TCR–peptide binding models requires evaluation protocols that explicitly test generalization beyond peptide memorization. Standard random train–test splits often allow identical or highly similar peptides to appear in both training and test sets, leading to overly optimistic performance estimates. To address this limitation, this thesis adopts the evaluation framework proposed by Grazioli et al., which was

specifically designed to expose memorization effects and quantify model robustness under controlled peptide distribution shifts. This framework forms the basis for all comparative analyses presented in the subsequent results chapter.

2.7.1 Grazioli Evaluation I: Random and Hard Peptide Splits

The first stage of the Grazioli evaluation framework introduces progressively stricter peptide-level separation between training and test data [20]. In addition to a standard random split, a *hard peptide split* is defined, in which all peptide identities present in the test set are completely excluded from the training set. The three regimes are referred to as Random Split (RS), Hard Split (HS), and Distance Split (DS).

Figure 2.6 illustrates the impact of this evaluation regime on two widely used models, ERGO-II and NetTCR-2.0, across multiple performance metrics. While both models achieve high scores under random splits, a substantial degradation in performance is observed under the hard split condition. This performance gap highlights the extent to which random-split evaluations may conflate memorization with true predictive capability.

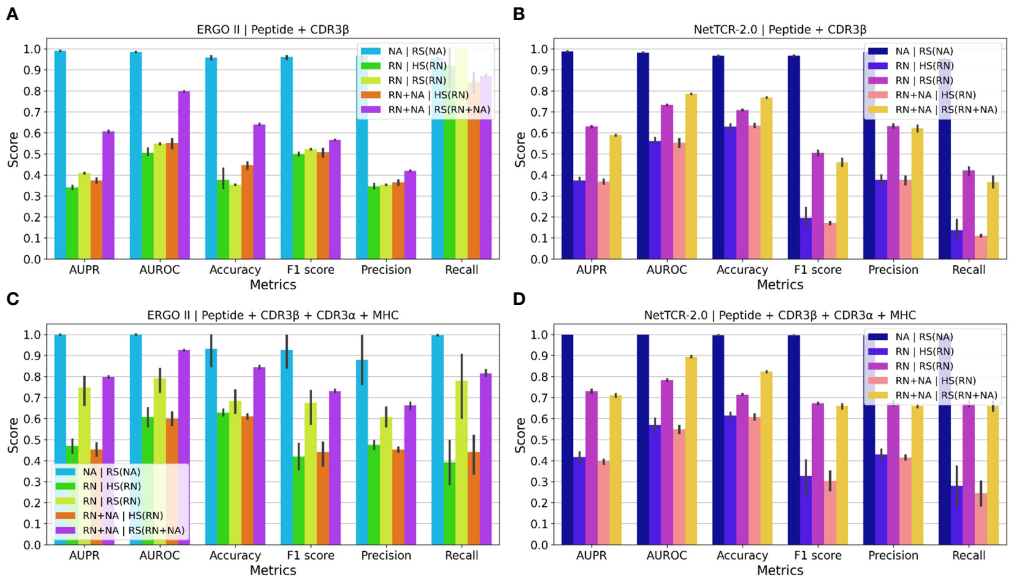

Figure 2.6. Comparison of ERGO-II and NetTCR-2.0 performance under random (RN) and hard peptide-level (NA) splits, reproduced from the Grazioli evaluation protocol. Panels show results for models trained on peptide+CDR3β inputs (A,B) and extended inputs including CDR3α and MHC information (C,D). Across all settings, performance drops substantially under hard splits, indicating reduced generalization to unseen peptides. This figure was taken and adapted from [20]

These findings motivate the need for more granular evaluation regimes that control not only peptide identity overlap, but also peptide similarity.

2.7.2 Grazioli Evaluation II: Distance-Based Peptide Splits

To move beyond the binary seen/unseen distinction imposed by the hard split, Grazioli et al. introduce a distance-aware evaluation framework termed the *Distance Split (DS)* [10]. The central idea is to quantify how far test peptides are from the training distribution and to assess model performance as this distance increases, thereby providing a controlled spectrum of out-of-distribution generalization difficulty.

As a prerequisite for this analysis, peptide similarity is measured using complementary distance metrics capturing different biological aspects. Sequence-based distances are computed using Levenshtein edit distance and BLOSUM substitution scores, while structural similarity is quantified using backbone RMSD derived from predicted peptide conformations. Figure 2.7 shows that sequence-based distances (Levenshtein and BLOSUM) are moderately correlated, whereas RMSD exhibits near-zero correlation with both, indicating that structural dissimilarity captures information largely orthogonal to sequence similarity. This observation motivates the use of both sequence- and structure-based distances within the DS framework.

The Distance Split algorithm stratifies peptides according to the percentile of their median distance to all other peptides, defining progressively harder test regimes. In contrast to the hard split, DS enforces that test peptides are unseen during training while simultaneously controlling how dissimilar they are from training peptides. This enables systematic evaluation across interpolation-like settings (low distance) and true extrapolation scenarios (high distance).

Figure 2.8 summarizes the empirical impact of distance-based splitting on multiple state-of-the-art models, including NetTCR-2.0, NetTCR-2.2, AVIB, and ERGO-II. Under RMSD-based splits, performance decreases as peptide structural distance increases, indicating reduced extrapolation capacity across structurally divergent peptides. In contrast, BLOSUM-based splits show an inverse trend, with larger sequence distances often associated with improved generalization. These observations suggest that robustness under random or hard splits does not fully capture model behavior under structured distribution shift and motivate the use of similarity-aware evaluation protocols for benchmarking.

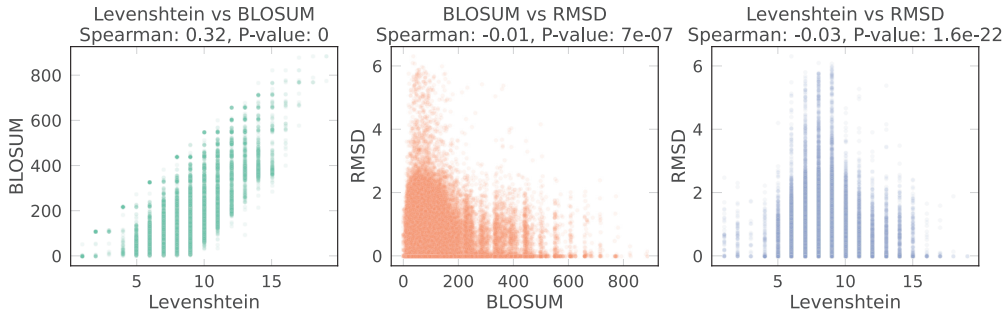

Figure 2.7. Correlation analysis between peptide distance metrics. Pairwise relationships between Levenshtein, BLOSUM, and RMSD distances are shown along with Spearman correlation coefficients. Sequence-based distances (Levenshtein and BLOSUM) exhibit moderate correlation, while RMSD shows near-zero correlation with sequence metrics, indicating that structural distance captures information largely orthogonal to sequence similarity. This figure was taken and adapted from [10].

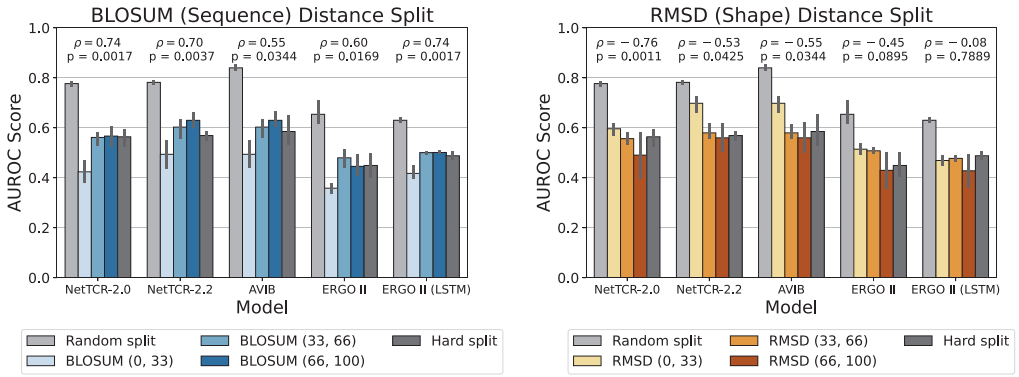

Figure 2.8. Model performance under Random Split (RS), Hard Split (HS), and Distance Split (DS) evaluation protocols. AUROC scores are reported for sequence-based (BLOSUM) and structure-based (RMSD) distance splits across increasing distance percentiles. While random splits yield optimistic performance estimates, distance-based splits reveal systematic degradation under RMSD shifts and contrasting trends under BLOSUM shifts, highlighting the dependence of generalization on the nature of peptide dissimilarity. This figure was taken and adapted from [10].

Chapter 3
Proposed Approach

3.1 Overview

This chapter describes the computational methodology used to develop, evaluate, and specialize a TCR–peptide binding prediction model. The experimental design follows the chronological order in which datasets, supervision strategies, and model configurations were constructed and compared.

The chapter proceeds in three parts. First, a unified positive interaction dataset is assembled, peptide similarity is modeled, and multiple negative-sampling strategies are generated using similarity-controlled clustering. These candidate supervision configurations are evaluated under peptide-shifted benchmarking to determine an appropriate training dataset.

Second, using the selected dataset, alternative model architectures and sequence encodings are compared under identical evaluation conditions. The best-performing configuration is subsequently optimized through systematic hyperparameter search and loss-function ablation. A parallel control experiment using the heterogeneous TChard dataset is conducted under the identical optimization and evaluation protocol to isolate the effect of supervision design.

Finally, the selected backbone model is adapted to a mutation-specific setting using a curated neoantigen/wild-type paired dataset. Mutation-derived physicochemical distance features are introduced and evaluated during fine-tuning. The resulting mutation-aware model is then applied within a case-study pipeline for NSCLC neoantigen prioritization.

All evaluation procedures follow the peptide-level splitting and similarity-controlled benchmarking protocol described in Chapter 2. Unless otherwise stated, model selection is based on validation performance, and test sets remain held out until final evaluation.

Software Dependencies

The implementation was developed using the following software and libraries:

- **Python 3** (3.9–3.12): Core scripting and data processing.
- **PyTorch** 1.12.1: Model training and inference.
- **Optuna**: Hyperparameter optimization.
- **Biopython** 1.78: Sequence alignment and structural superposition.
- **ESMFold** / fair-esm 1.0.2: Peptide structure prediction.
- **NetMHCpan** 4.2: MHC class I binding prediction.

3.2 Backbone Dataset Construction

3.2.1 Positive Interaction Dataset Construction and Diversity

The foundation of the framework is a set of experimentally validated TCR–peptide interactions assembled from multiple public repositories. Source datasets are filtered to remove incomplete records and restricted to valid TRB CDR3β sequences paired with peptide epitopes. All datasets are merged into a unified positive interaction set, and duplicate CDR3β–peptide pairs are removed.

The final positive collection contains 103,935 rows, 1,838 unique peptides, and 99,494 unique CDR3β sequences. The source composition is dominated by VDJdb (78.0%), followed by McPAS (12.0%), NetTCR2 (9.9%), and a small NetTCR2 sample subset (0.03%). Each positive instance consists of a CDR3β amino acid sequence paired with a peptide sequence and is assigned a binary label of 1. This unified positive dataset is kept fixed throughout all subsequent experiments and serves as the reference set for negative sampling.

3.2.2 Diversity Characterization Prior to Negative Construction

To quantify structural properties of the positive repertoire before negative generation, exploratory data analysis (EDA) was performed on source composition, peptide length distribution, and CDR3β diversity.

Peptide lengths range from 7 to 25 amino acids (mean = 9.18; median = 9), with most peptides being canonical 9-mers. CDR3β lengths range from 4 to 38 amino acids (mean = 14.33; median = 14). The CDR3β rank–abundance profile is strongly long-tailed, with 97.19% singletons and a small repeated core (maximum frequency = 90), indicating high diversity and that most CDR3β sequences occur only once.

These structural characteristics inform subsequent split design and negative construction. The concentration of observations within a subset of peptides and the sparsity of clonotype repetition may affect model evaluation under random splitting. Accordingly, peptide-controlled splitting and similarity-aware negative construction are considered in later sections.

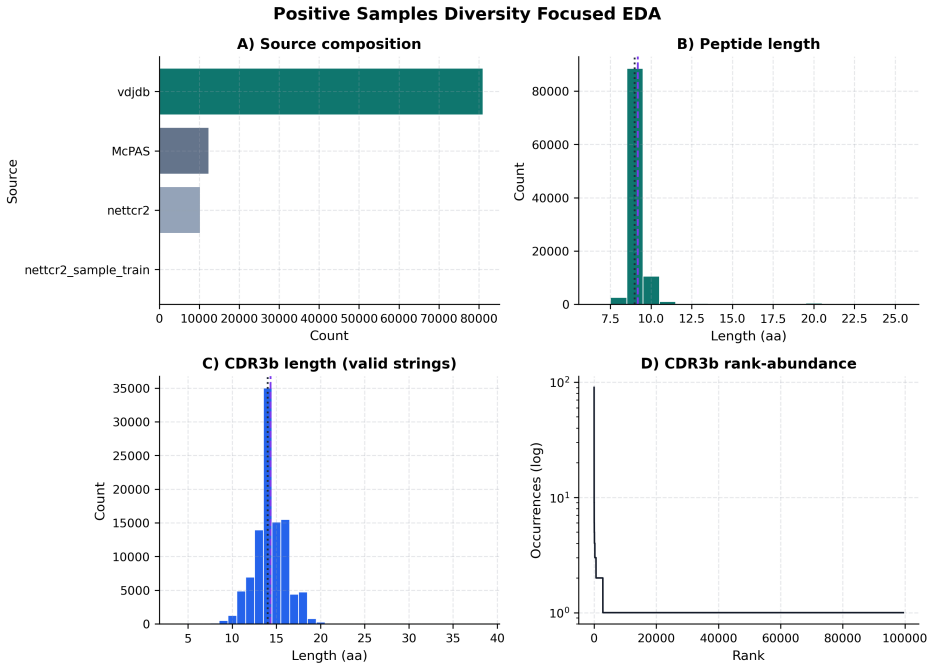

Figure 3.1. Diversity-oriented exploratory analysis of the unified positive interaction dataset: (A) source distribution, (B) peptide length distribution, (C) CDR3β length distribution, and (D) CDR3β rank-abundance curve. These diagnostics inform split design, padding limits, and leakage-aware evaluation strategy.

Figure 3.2 characterizes the global structure of peptide similarity under BLOSUM-based normalized alignment.

Panel (A) shows the distribution of pairwise normalized similarity scores across all unique peptides. The distribution is highly skewed toward low similarity values, with the majority of peptide pairs exhibiting near-zero similarity. This indicates that most peptides are mutually dissimilar in sequence space. A long but sparse tail toward higher similarity values reflects the presence of smaller neighborhoods of related peptides.

Panel (B) visualizes the similarity matrix for a stratified subset of peptides, reordered by hierarchical clustering. The strong diagonal corresponds to self-similarity (normalized to 1.0). Off-diagonal regions are predominantly low-similarity (dark), consistent with the distribution in Panel (A). However, localized blocks of elevated similarity are visible, indicating the presence of peptide clusters sharing higher sequence similarity.

Together, these observations suggest that peptide space is globally sparse but locally structured: most peptides are dissimilar, while subsets form similarity-defined neighborhoods. This structural organization motivates the use of clustering to control negative sample construction. By identifying and grouping locally similar peptides, cluster-controlled sampling can systematically exclude near-neighbor peptides when constructing negatives.

3.2.3 Peptide Similarity and Clustering

Before generating negative samples, peptide space is modeled by pairwise similarity and then partitioned by hierarchical clustering. This section describes how the BLOSUM similarity matrix was computed and how the dendrogram and cluster assignments were obtained. The resulting clusters are used in the next section to define cluster-controlled negative sampling.

BLOSUM-Based Peptide Similarity

Pairwise similarity between peptides is computed from the unique peptide set derived from the positive interaction data. For each pair of peptide sequences p_1, p_2, a global alignment is performed using the Needleman–Wunsch algorithm with the BLOSUM62 substitution matrix and a linear gap penalty. The raw alignment score $S(p_1, p_2)$ is then normalized by the maximum of the two self-alignment scores, $\max\{S(p_1, p_1), S(p_2, p_2)\}$, so that the similarity is comparable across peptides of different lengths and lies in a consistent range. This yields a symmetric similarity matrix over all unique peptides, which is saved for clustering and analysis.

Figure 3.2 summarizes the outcome: the distribution of pairwise similarities is shown in panel (A), and a stratified subset of the similarity matrix is shown in panel (B). The subset is obtained by sampling peptides proportionally from each cluster (at threshold 0.40) so that the visualization is representative of the full peptide space, then reordered by hierarchical clustering to reveal block structure. The concentration of mass in the similarity distribution and the block structure in the matrix reflect the fact that many peptides fall into local neighborhoods of biochemical similarity, which motivates the use of clusters to control negative sampling.

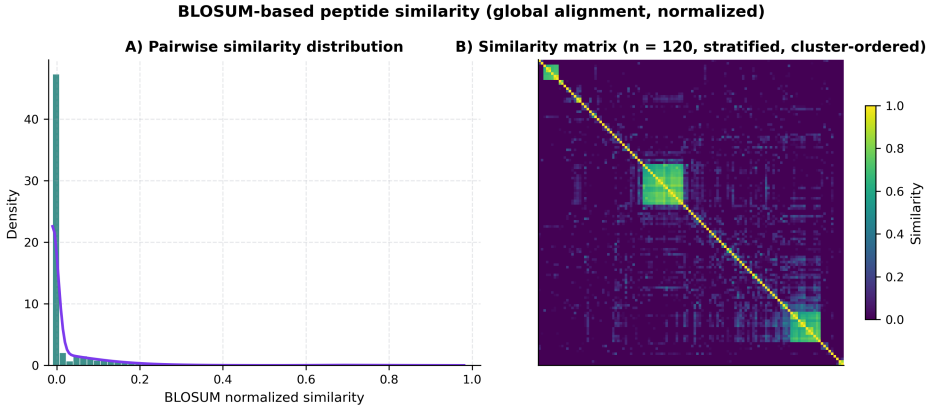

Figure 3.2. BLOSUM-based peptide similarity: (A) distribution of pairwise normalized similarities; (B) similarity matrix for a stratified subset of 120 peptides (proportionally sampled by cluster), reordered by hierarchical clustering to reveal block structure.

Hierarchical Clustering of Peptides

A peptide distance matrix is obtained by taking distance as $1 -$ similarity, so that similar peptides have small distance. Agglomerative hierarchical clustering is then applied with

average linkage. The resulting dendrogram is cut at different distance thresholds to produce clusterings of varying granularity: lower thresholds yield more and smaller clusters, while higher thresholds yield fewer, larger clusters. Importantly, each cut height produces a different clustering (different number of clusters and different partition). At distance thresholds of 0.20, 0.40, 0.60, and 0.80, the resulting cluster counts are 1373, 1133, 892, and 423, respectively, for the present data set. (Figure 3.3).

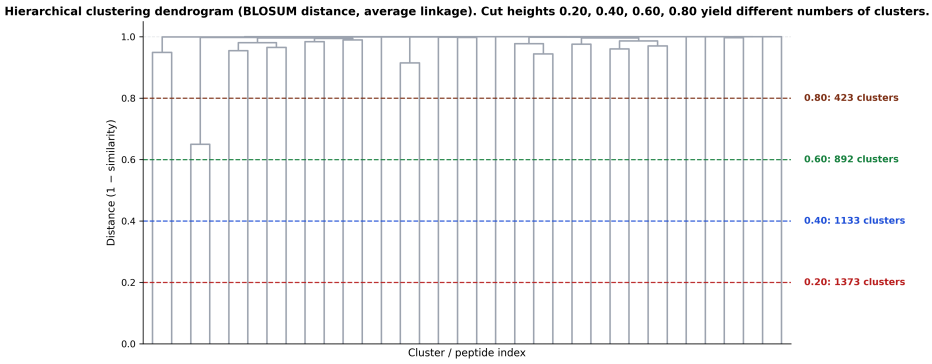

Figure 3.3. Hierarchical clustering dendrogram (BLOSUM distance, average linkage): (A) full peptide set truncated by level with leaf counts; (B) zoomed view on a subset showing the last merges. Dashed lines indicate example cut thresholds (0.30, 0.40, 0.50).

3.2.4 Negative Sample Generation

Because experimentally validated non-binding TCR–peptide pairs are largely unavailable, negative samples are constructed synthetically. Given the set of positive interactions observed $\mathcal{P}$, negatives are generated by pairing TCR sequences with peptides that are not recorded as binders. The method used for negative construction determines the statistical structure of the training data and may influence the behavior of the model under distribution shift.

To examine this effect systematically, multiple datasets are generated by varying the peptide similarity clustering threshold (Section 3.2.3). This procedure produces both randomly constructed and similarity-controlled negative sets, enabling comparative evaluation of supervision design.

Baseline: Random Negative Sampling

In the baseline strategy, negatives are formed by randomly pairing each CDR3β sequence with peptides such that $(t, p') \notin \mathcal{P}$. This guarantees that no experimentally validated positive interaction for a given TCR is mislabeled as negative.

However, random pairing does not constrain peptide similarity. Peptides that are highly similar to known binders, or that bind other TCRs, may be selected as negatives. This may result in dissimilar negatives as well as potential similarity-driven ambiguity in the supervision signal.

Cluster-Controlled Negative Sampling (CCNS)

To reduce similarity-driven ambiguity, a cluster-controlled sampling strategy is introduced. For each positive pair (t, p), negative peptides are sampled under the constraints $(t, p') \notin \mathcal{P}$ and $p' \notin \mathrm{Cluster}(p)$, where $\mathrm{Cluster}(p)$ denotes the peptide cluster defined at a given BLOSUM distance threshold.

By excluding peptides from the same similarity cluster as the positive peptide, this strategy limits the selection of near-neighbor negatives. The influence of peptide similarity on model performance can then be assessed systematically across datasets constructed with different clustering thresholds.

Figure 3.4 compares random and cluster-controlled negative construction. Under random sampling, negatives may originate from the same peptide cluster as known binders. Cluster-controlled sampling blocks such within-cluster pairings.

Cluster-controlled negatives remain synthetic and do not constitute experimentally validated non-binding interactions. The objective is solely to modify the similarity structure of the supervision signal.

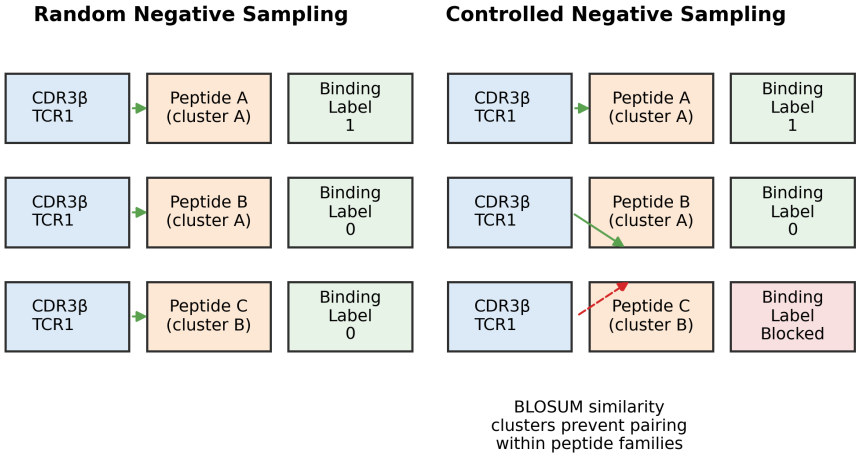

Figure 3.4. Schematic comparison of random versus controlled negative sampling. Random mismatching can yield negatives from the same peptide cluster as a known binder, while controlled sampling uses BLOSUM-based clustering to block within-cluster negatives.

3.3 Cluster Threshold Selection via Grazioli Evaluation

Before comparing model architectures or sequence encodings, an initial experiment was conducted to determine an appropriate negative-sampling configuration. The objective of this step was to identify which dataset, constructed using different negative-sampling strategies, yielded more stable generalization under peptide-shifted evaluation. The selected dataset was subsequently fixed for all remaining experiments.

3.3.1 Experimental Setup

A single shallow transformer architecture with a physicochemical encoding was used across all candidate datasets. The training configuration was kept identical in all cases: binary cross-entropy loss, identical optimization settings, and identical early-stopping criteria. No architectural, encoding, or loss variations were introduced at this stage.

3.3.2 Candidate Datasets

The evaluated datasets included the random negative baseline and cluster-controlled negative sampling (CCNS) constructed at BLOSUM distance thresholds of 0.20, 0.40, 0.60, and 0.80 (Section 3.2.3). Each dataset was processed under the same Grazioli-compliant preprocessing and split-generation procedure.

3.3.3 Evaluation Protocol

For each dataset, the model was trained from scratch and the checkpoint with the highest validation AUROC was retained. Evaluation was performed under the hard peptide-identity split and BLOSUM-based distance split (Distance Split 3), using AUROC and AUPR as performance metrics. These splits were selected to assess peptide-level generalization under distribution shift.

3.3.4 Dataset Selection

The dataset associated with the clustering threshold that demonstrated the most consistent performance across these splits was selected as the preferred negative-sampling configuration.

3.3.5 Subsequent Use

The selected dataset was fixed for all subsequent experiments in this chapter, including architecture comparison, encoding comparison, hyperparameter optimization, and loss-function ablation. No additional dataset selection was performed after this step.

The selected clustering threshold is reported in the Results chapter.

3.3.6 Control Dataset (TChard)

To assess whether cluster-controlled negative sampling yields better generalization than standard heterogeneous training data, we use the TChard dataset[1] [21] as an external control. It is a widely adopted TCR–peptide binding benchmark introduced by Grazioli et al. [20], aggregating 566,218 samples from heterogeneous sources including IEDB, VDJdb, McPAS-TCR, and the NetTCR-2.0 repository.

The TChard control dataset comprises 566,218 peptide–CDR3β pairs in total. Of these, 414,051 entries (73.13%) are labeled as negative interactions, while 151,958 entries correspond to positive binding pairs.

[1] https://zenodo.org/records/6962043

The negative class itself is heterogeneous. Among the 414,051 negative samples, 265,223 (64.06%) are generated by random mismatch between CDR3β and peptide sequences. These randomized negatives therefore constitute 46.8% of the entire dataset. The remaining 148,828 negatives (35.94% of the negative class) originate from experimentally validated non-binding assays.

Consequently, the TChard supervision signal combines synthetic mismatches and experimentally supported non-binders. Peptide similarity is not explicitly controlled during negative construction. This difference provides a basis for comparison with the cluster-controlled sampling strategy introduced in this thesis.

3.4 Architecture and Encoder Selection

Following dataset selection (Section 3.3), all subsequent experiments were conducted using the fixed negative-sampling configuration. No further dataset modification or selection was performed beyond this point.

3.4.1 Sequence Encoding

CDR3β and peptide sequences were encoded using a unified framework that allows any encoder to be paired with any model architecture. Five encoding schemes were implemented and evaluated in the model-selection stage.

AAIndex. Each amino acid is represented by a five-dimensional vector derived from selected AAIndex physicochemical properties (hydrophobicity, polarity, charge, molecular weight, and an additional descriptor). Values are min–max normalized per dimension across the amino acid alphabet [31].

Atchley. Atchley factors provide five-dimensional statistical descriptors derived from a broad set of physicochemical and structural attributes. Each amino acid is mapped to a fixed five-dimensional vector, normalized across the alphabet [3].

BLOSUM62. Each amino acid is represented by the corresponding row of the BLOSUM62 substitution matrix (20 dimensions), normalized to the unit interval. This encoding reflects evolutionary substitution likelihoods [24].

One-hot. Each amino acid is represented by a 20-dimensional binary vector with a single non-zero entry. This encoding serves as a representational baseline [2].

Physicochemical. A three-dimensional encoding capturing hydrophobicity, polarity, and charge per amino acid. Values are normalized across the alphabet [34].

For all encodings, sequences were padded or truncated to fixed maximum lengths (CDR3β: 50 residues; peptide: 30 residues). Padding masks were applied for architectures that require explicit masking (e.g., transformer). Identical maximum lengths were used across all encoder–architecture combinations to ensure comparability.

3.4.2 Model Architecture

Three backbone architectures were compared under identical training and evaluation conditions. All architectures operate on encoded CDR3β and peptide sequences and output a single binding logit.

CNN. Two independent 1D convolutional branches process the CDR3β and peptide sequences. Each branch consists of two convolutional layers (kernel size 3) with ReLU activation followed by adaptive average pooling. The resulting sequence representations are concatenated and passed through a multi-layer perceptron (MLP) with dropout to produce a scalar logit [35].

LSTM. Two bidirectional LSTM branches process the sequences independently. Final forward and backward hidden states are concatenated to form sequence representations, which are then combined and passed through an MLP to produce a scalar logit [25].

Transformer. Linear projections map encoded sequences to a shared hidden dimension (e.g., 128). Learned positional embeddings are added. Separate transformer encoder stacks process CDR3β and peptide sequences independently. Padding masks exclude padded positions from attention and pooling. Masked mean pooling produces fixed-length representations, which are concatenated and passed through an MLP to produce a scalar logit Transformer [65].

In all architectures, training uses binary cross-entropy with logits. At evaluation time, a sigmoid transformation is applied to obtain predicted binding probabilities.

3.4.3 Evaluation Protocol

All architecture–encoder combinations were evaluated under the peptide-shifted benchmarking framework described in Chapter 2.

Evaluation Metrics

Model performance is evaluated using standard binary classification metrics.

Let TP, FP, TN, and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively.

- **Precision:**
$$\text{Precision} = \frac{TP}{TP + FP}$$
- **Recall (Sensitivity):**
$$\text{Recall} = \frac{TP}{TP + FN}$$
- **Accuracy:**
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
- **F1-score:**
$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

In addition, threshold-independent ranking metrics are reported:

- **AUROC:** Area under the receiver operating characteristic curve.
- **AUPR:** Area under the precision–recall curve.

Split Regimes

Three split regimes are used:

- Random split
- Hard peptide-identity split
- Distance-Based Split

Random Split. Samples were assigned to training, validation, and test partitions using an 80/10/10 ratio. Peptide identities were allowed to overlap across partitions.

Hard Peptide-Identity Split. Although the hard split removes peptide identity overlap between partitions, it does not explicitly control the degree of similarity between training and test peptides.

Distance-Based Split. Distance-based splits extend identity-level separation by additionally stratifying peptides according to similarity. Two complementary distance metrics were employed:

- Sequence-level similarity was quantified using global alignment scores derived from the BLOSUM62 substitution matrix, computed via BioPython's `pairwise2` implementation with gap penalties of -10 (opening) and -1 (extension), matching the alignment configuration described by Grazioli et al.
- Structural similarity was quantified using Cα root mean square deviation (RMSD) following optimal superposition of ESMFold-predicted peptide structures. Superposition was performed using BioPython's `Superimposer`, which is methodologically equivalent to the PyMOL-based Cα alignment procedure used in the original framework.

For each metric, all pairwise peptide distances were computed to form a symmetric peptide–peptide distance matrix. The matrix was then pivoted into peptide-by-peptide form, and for each peptide the median distance to all other peptides was calculated using an estimator:

$$d_i = \mathrm{nanmedian}(D_i),$$

where D_i denotes the vector of distances between peptide i and all other peptides. This median-based aggregation replicates the grouping strategy described in the original work, ensuring that peptides are stratified according to their overall similarity profile rather than individual pairwise relationships.

Peptides were subsequently partitioned into three percentile strata based on their median distance values:

- low-distance (0–33%),
- intermediate-distance (33–66%),
- high-distance (66–100%).

For each stratum, an independent split was generated. Within each distance regime, the test and validation sets contain only peptides belonging to the selected percentile interval, while the training set may include peptides from outside that interval.

To avoid isolated test peptides, at least one peptide within a defined similarity threshold was assigned to the training set, ensuring controlled overlap in feature space.

Within each distance-based regime, splits were constructed using a 90/5/5 ratio for training, validation, and test sets, respectively. All percentile thresholds and distance configurations were defined once prior to model comparison and subsequently fixed across all experiments. No split parameters were tuned per architecture or encoding strategy, thereby ensuring that performance differences reflect model capacity rather than evaluation tailoring.

3.4.4 Architecture and Encoder Comparison Procedure

All encoder–architecture combinations were trained independently on the fixed dataset. Model checkpoints were selected based on validation AUROC. Final evaluation was performed under the full peptide-shifted protocol (random, hard identity, and distance-based splits).

This comparison step identifies the architecture and encoding configuration that demonstrates the most stable generalization under peptide-level distribution shift. The selected configuration is subsequently used for hyperparameter optimization and mutation-aware fine-tuning.

3.5 Hyperparameter Optimization

After selecting the transformer architecture as the primary backbone, a systematic hyperparameter optimization procedure was conducted to identify a stable training configuration under the selected evaluation regime. Using the fixed cluster-threshold dataset and the selected encoder–architecture configuration, optimization was performed using Optuna with a Tree-structured Parzen Estimator (TPE) sampler and a MedianPruner for early stopping of underperforming trials. The optimization procedure was implemented using a dedicated Optuna training pipeline.

The hyperparameters jointly optimized during HPO included architectural parameters (hidden dimension, number of transformer layers, attention heads), optimization parameters (learning rate, weight decay, dropout, batch size), and imbalance-handling parameters (training class ratio and loss-weighting strategy). The complete search space and sampling distributions are detailed in Appendix A.1. The following paragraphs describe the rationale for imbalance handling.

3.5.1 Optimization Objective

The objective of HPO was to maximize validation AUROC under the hard peptide-identity split. AUROC was chosen as the optimization target to ensure stable ranking behavior under class imbalance, while AUPR was tracked as a secondary metric for analysis but not directly optimized.

Validation and test sets retained their natural prevalence throughout HPO. Only the training set was resampled per trial.

3.5.2 Class Imbalance as an Explicit Hyperparameter

Following literature findings that TCR–peptide predictors are highly sensitive to class imbalance and negative sampling regimes, the positive-to-negative ratio in the training set was treated as an explicit hyperparameter rather than being fixed a priori.

The positive class ratio (`pos_ratio`) was selected from four predefined values: 0.5 (1:1), 1/3 (1:2), 0.2 (1:4), and 0.1 (1:9).

Importantly, only the training set was resampled per trial.

This design prevents evaluation distortion while allowing robustness analysis across biologically plausible imbalance regimes.

3.5.3 Loss Weighting Strategy

To avoid double correction effects, loss weighting was also treated as a hyperparameter with three options:

- No weighting
- Inverse-frequency weighting
- Square-root inverse-frequency weighting

For near-balanced training ratios (e.g., 1:1), weighting was automatically disabled to prevent overcorrection.

3.5.4 Training Protocol per Trial

For each trial, model parameters were initialized from scratch without pretrained weights. Optimization was performed using the Adam optimizer with ReduceLROnPlateau learning-rate scheduling. Models were trained for up to 50 epochs, with early stopping and Optuna pruning used to terminate underperforming configurations.

To reduce reuse of identical negative subsets across trials, a trial-dependent random seed was used when resampling the training data. As a result, different trials were exposed to different negative subsets even under identical class ratios.

3.5.5 Final Model Selection

After optimization, the best hyperparameter configuration (highest validation AUROC) was retrained on the full training set using the selected imbalance ratio and loss strategy. The final model was then evaluated on the held-out test set under the same hard peptide-identity split regime used for optimization.

3.5.6 Parallel Optimization on the TChard Control Dataset

To enable a controlled comparison between similarity-controlled negative sampling and heterogeneous supervision, the same hyperparameter optimization procedure was applied to the TChard control dataset (Section 3.3.6).

The transformer architecture with AAIndex encoding was used without modification. The identical Optuna search space, optimization objective (validation AUROC under the hard peptide-identity split), training protocol, class-ratio sampling strategy, and loss-weighting configurations were employed.

No hyperparameter ranges were altered for TChard. Validation and test splits were handled identically, and only the training set was resampled per trial.

This parallel optimization ensures that any performance differences observed between the cluster-controlled dataset and TChard reflect differences in supervision structure rather than differences in optimization strategy.

3.6 Loss Function Ablation Study

After selecting the final backbone architecture (transformer) and encoder (AAIndex) through model comparison and hyperparameter optimization (HPO), we conducted a dedicated ablation study to assess the impact of the training loss formulation on generalization performance. Because the dataset exhibits substantial class imbalance under Grazioli-compliant splits, the choice of loss function directly influences optimization dynamics and the balance between easy and hard examples.

All loss configurations were evaluated on the fixed transformer+AAIndex architecture under identical data splits and early-stopping criteria. Model selection was based on validation loss, and the best checkpoint was restored at the end of training.

3.6.1 Loss Configurations

Four loss configurations were compared:

(1) BCE-with-logits (plain). This configuration uses `BCEWithLogitsLoss`, where the model outputs raw logits and no sigmoid is applied in the forward pass. Class imbalance is addressed using a positive-class weight defined as

$$\texttt{pos\_weight} = \frac{1 - \texttt{pos\_ratio}}{\texttt{pos\_ratio}},$$

computed from the training split. The loss is applied in logit space with `reduction='none'`, followed by per-sample weighting before averaging.

(2) BCE-with-logits + resampling. This configuration uses the same `BCEWithLogitsLoss` formulation and class-weighting scheme as above, but training batches are constructed using positive-class resampling such that the effective positive ratio is approximately 0.5. This balanced-batch strategy was identified during HPO as a potentially stabilizing regime. While class weights remain defined from the training distribution, the optimization dynamics differ due to resampled batch composition.

(3) Focal loss (mild). To reduce the influence of easy examples and focus training on harder cases, we evaluated a focal loss formulation:

$$\mathcal{L} = \alpha_t (1 - p_t)^{\gamma} \cdot \text{BCE},$$

with $\alpha = 0.5$ and $\gamma = 0.5$. In this setting, the model outputs probabilities via a sigmoid activation before loss computation. The focal modulation term $(1 - p_t)^\gamma$ down-weights well-classified examples, though only mildly due to the small γ value.

(4) Focal loss (standard). A more aggressive focal configuration was also evaluated, using $\alpha = 0.25$ and $\gamma = 2.0$, following the original focal loss formulation. This setting strongly down-weights easy examples and shifts training emphasis toward hard or misclassified samples. As in the mild focal variant, the model outputs probabilities via sigmoid before applying the loss.

3.7 NetTCR-2.0 Ablation

To contextualize our chosen encoder and architecture relative to a widely used baseline, we conducted an ablation that mirrors the design of NetTCR-2.0 [44] (BLOSUM encoding with a CNN backbone) while varying encoder and architecture under the same Grazioli-compliant evaluation protocol.

Four controlled variants were evaluated:

1. Original NetTCR-2.0-style (BLOSUM + CNN)
2. Encoder substitution only (AAIndex + CNN)
3. Architecture substitution only (BLOSUM + transformer)
4. Both substitutions (AAIndex + transformer)

All four variants were trained and evaluated on the same cluster-threshold dataset and under the same peptide-shifted splits described in Section 3.3, including both BLOSUM- and RMSD-based distance regimes. Results are reported in Section 4.9 and compared to the proposed model (transformer + AAIndex after HPO and loss ablation).

3.8 Neoantigen Fine-Tuning Dataset Construction

To enable transfer learning for mutation-specific TCR recognition, a curated neoantigen–CDR3β dataset was assembled from three public repositories: NeoTCR [42], McPAS -TCR [61], and CEDAR [33] (The Cancer Epitope Data Resource).

The objective of this dataset is to provide paired neoantigen and wild-type peptides for the same TCR, enabling supervised discrimination of mutation-induced specificity.

3.8.1 Data Sources

NeoTCR. NeoTCR [42][2] was downloaded from its public GitHub repository and converted to a unified tabular format. It provides CDR3β sequences, neoantigen peptides, mutation annotations (e.g. "K160N"), HLA alleles, tumor type, and literature references. All entries containing valid CDR3β and peptide sequences were retained.

[2] https://github.com/lyotvincent/NeoTCR

McPAS-TCR. McPAS-TCR [61][3] was filtered by keyword matching within the `Pathology`, `Remarks`, and `Epitope.peptide` fields to identify neoantigen-associated entries.

CEDAR / IEDB receptor table. CEDAR (The Cancer Epitope Database and Analysis Resource) [33][4] contributed additional peptide–CDR3β associations. Potential Neo/WT candidate pairs were identified when the same CDR3β appeared with two peptides differing by a single amino acid, consistent with point mutations.

3.8.2 Dataset Integration and Deduplication

All source tables were merged using standardized column names. Exact duplicate CDR3β–neoantigen pairs were removed, and a `source` field was retained to preserve dataset provenance.

After deduplication, the merged dataset contained 1,313 unique neoantigen–CDR3β pairs with binder label = 1.

3.8.3 wild-type Peptide Recovery and Negative Label Assumption

Because the fine-tuning objective requires paired neoantigen and wild-type peptides for the same TCR, a structured wild-type recovery pipeline was implemented.

wild-type peptides were obtained through a three-stage procedure:

1. **Direct mutation reversal.** Mutation annotations were parsed to reconstruct the wild-type residue by reversing the reported amino acid substitution.
2. **UniProt-based retrieval.** When direct reconstruction was not feasible, protein or gene identifiers were used to retrieve the full wild-type sequence from UniProt [64][5], followed by local alignment to extract the corresponding peptide context.
3. **NEPdb lookup.** When automated reconstruction failed, NEPdb [73][6] was queried to recover the matching wild-type peptide.

After reconstruction and conflict resolution, the dataset enforces a strict pairing structure: for each CDR3β sequence, exactly one neoantigen peptide (label = 1) and one corresponding wild-type peptide (label = 0) are retained.

wild-type pairs are not experimentally validated non-binders. They are treated as putative negatives under the immunological principle of central tolerance, whereby high-affinity TCRs recognizing self-peptides are typically eliminated or inactivated during thymic selection. Accordingly, mutation-specific TCRs are expected to preferentially bind neoantigens rather than their wild-type counterparts, while acknowledging that cross-reactivity cannot be entirely excluded.

[3] https://friedmanlab.weizmann.ac.il/McPAS-TCR/
[4] https://cedar.iedb.org/
[5] https://www.uniprot.org/
[6] https://nep.whu.edu.cn/

3.8.4 Final Dataset Statistics

The resulting fine-tuning dataset contains:

- 2,626 total peptide–CDR3β pairs
- 1,313 unique CDR3β sequences
- 572 unique peptides
- 1,313 neoantigen pairs (label = 1)
- 1,313 wild-type pairs (label = 0)

The dataset is balanced at the pair level (approximately 1:1 Neo/WT) and is composed of NeoTCR (59.3%), McPAS-TCR (34.0%), and CEDAR (6.7%). Figure 3.5 summarizes the composition and structural properties of the fine-tuning dataset.

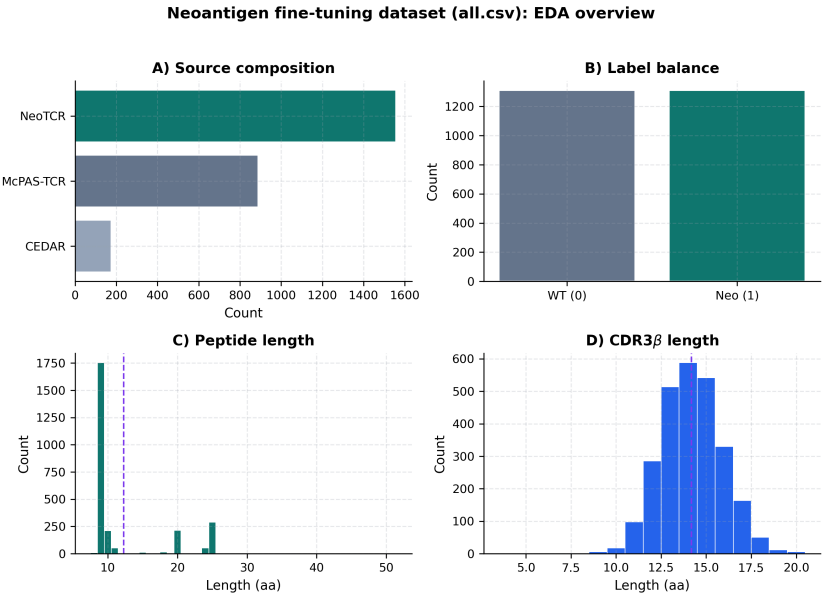

Figure 3.5. Exploratory data analysis of the neoantigen fine-tuning dataset: source contribution, label balance, and peptide and CDR3β length distributions.

3.8.5 Feature Definitions

To quantify the biochemical impact of each somatic mutation, three complementary physicochemical distance metrics were computed between each neoantigen peptide and its corresponding wild-type counterpart.

BLOSUM Distance. For each aligned residue position, the substitution penalty was computed using the BLOSUM62 matrix as

$$S_{aa,aa} - S_{aa,bb},$$

where aa denotes the amino acid in the neoantigen and bb the corresponding wild-type residue. These penalties were summed across the peptide. Higher values indicate greater evolutionary divergence.

Aliphatic Index Distance. The aliphatic index (AI) of each peptide was calculated as

$$AI = 100\,(x_A + 2.9x_V + 3.9(x_I + x_L))\,,$$

and the mutation-induced shift was defined as

$$|AI_{\text{neo}} - AI_{\text{WT}}|\,.$$

Boman Distance. The Boman index estimates mean per-residue protein–protein interaction potential. The distance was computed as

$$|B_{\text{neo}} - B_{\text{WT}}|\,.$$

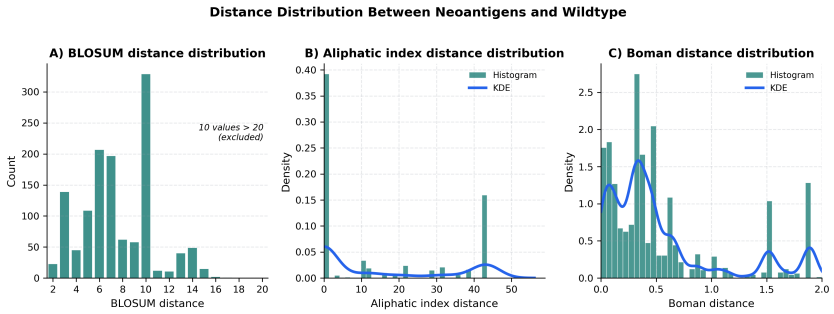

Figure 3.6. Distribution of mutation-derived physicochemical distances between neoantigen and wild-type peptides in the fine-tuning dataset: (A) BLOSUM62 substitution distance; (B) aliphatic index shift; (C) Boman interaction potential shift. Axes are restricted to the principal data range for visualization clarity; extreme outliers are excluded and annotated.

To characterize the physicochemical dissimilarity between neoantigen and wild-type peptides in the fine-tuning dataset, the three mutation-derived distance metrics defined in Section 3.8.5 were computed for each matched (neoantigen, wild-type) pair. Each pair consists of a neoantigen peptide (label = 1) and its reconstructed wild-type counterpart (label = 0), matched by CDR3β sequence.

Figure 3.6 summarizes the empirical distributions of the BLOSUM substitution distance, aliphatic index shift, and Boman interaction shift across all Neo/WT pairs. Axes are restricted to the principal data range for visualization clarity; extreme outliers are excluded and annotated. All values were retained during training.

3.8.6 Feature Ablation

The three delta features were evaluated individually and in combination during fine-tuning to determine their contribution to mutation-specific discrimination. The following configurations were tested:

- Each feature individually (1-by-1)

- All pairwise combinations (2-by-2)
- All three features combined

This ablation design isolates the incremental contribution of each physicochemical descriptor and determines whether combined mutation signals improve paired neoantigen discrimination.

3.8.7 Transfer Learning Strategy

The pretrained backbone remains a binary TCR–peptide binding predictor trained with binary cross-entropy. Fine-tuning preserves the same architecture and output head. Neoantigen (label = 1) and wild-type (label = 0) pairs are treated as standard binary examples, while evaluation uses paired accuracy.

Paired accuracy is defined as the proportion of CDR3β cases for which the predicted binding score of the neoantigen exceeds that of its corresponding wild-type peptide.

Fine-tuning proceeds in two stages. During an initial warm-up phase (five epochs), transformer encoder layers are frozen and only the MLP head is updated. Subsequently, the encoder is unfrozen and trained with a lower learning rate (1×10^{-6}) than the classification head (1×10^{-5}).

The selected mutation-derived delta features (BLOSUM distance and Boman distance) are injected at the penultimate layer. Gaussian noise is added during training for regularization.

Train, validation, and test splits are grouped by CDR3β sequence to prevent leakage of paired samples across partitions. Optimization uses AdamW [38] with cosine learning-rate decay and early stopping. AdamW is an adaptive gradient method that decouples weight decay from the gradient-based parameter update, preventing the implicit coupling between L2 regularization and adaptive learning-rate scaling observed in standard Adam. This decoupled formulation improves regularization stability and generalization performance. The loss function is `BCEWithLogitsLoss` with optional class weighting.

This strategy is designed to align the learning objective with neoantigen prioritization while preserving general TCR–peptide interaction representations learned during backbone training.

3.9 Case Study: NSCLC Neoantigen Prioritization

To demonstrate downstream application of the mutation-aware model described in Section 3.8.7, we designed a case-study pipeline for NSCLC neoantigen prioritization. This section describes the construction of the neoantigen set, the MHC filtering procedure, and the scoring framework used for TCR ranking. No evaluation or interpretation is presented here; quantitative outcomes are reported in Chapter 4.

3.9.1 Neoantigen Construction and MHC Filtering

Somatic mutation data were obtained from the COSMIC resistance mutation catalogue (GRCh38) [60]. After cleaning and deduplication, mutant protein sequences were recon-

structed by applying Human Genome Variation Society (HGVS) protein-level notation (HGVSP) substitutions to Ensembl-derived wild-type sequences.

From each mutant protein, all 9-mer peptides spanning the mutated residue were generated, reflecting canonical MHC class I presentation length constraints (8–11 amino acids). Only mutant peptides differing from their corresponding wild-type 9-mers at the mutated residue were retained.

All candidate neoantigen 9-mers were evaluated using NetMHCpan 4.2 across a fixed HLA class I allele panel. Peptides with EL rank ≤ 0.5 were classified as strong binders and retained for downstream analysis.

3.9.2 Mutation-Derived Feature Characterization in NSCLC

For each strong-binding neoantigen peptide p, the three mutation-derived physicochemical deltas defined in Section 3.8.5 (BLOSUM substitution distance, aliphatic index shift, and Boman interaction shift) were computed relative to the corresponding wild-type peptide.

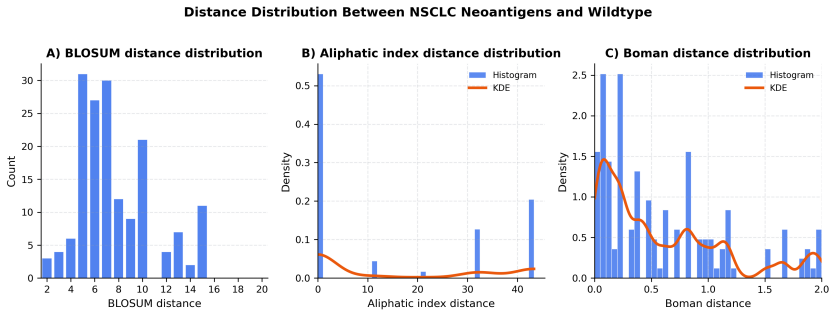

Figure 3.7. Distribution of mutation-derived physicochemical distances between NSCLC neoantigen peptides and their corresponding wild-type counterparts: (A) BLOSUM62 substitution distance; (B) aliphatic index shift; (C) Boman interaction potential shift. Axes are restricted to the principal data range for visualization clarity; extreme outliers are excluded and annotated.

Figure 3.7 shows the empirical distribution of mutation-induced distances in the NSCLC neoantigen set. The overall profiles are qualitatively similar to those observed in the fine-tuning Neo/WT dataset (Figure 3.6), indicating that the case-study mutations lie largely within the physicochemical regime represented during mutation-aware model adaptation.

BLOSUM distances predominantly fall within moderate substitution ranges, suggesting that most NSCLC mutations correspond to conservative or moderately disruptive amino-acid replacements. The Boman distance exhibits a right-skewed distribution with substantial spread, reflecting heterogeneous shifts in predicted interaction propensity. In contrast, the aliphatic index shift is strongly concentrated near zero, indicating limited alteration of bulk hydrophobic side-chain composition for many mutations.

For downstream inference, only the two features selected during Neo/WT ablation (Section 4.10)—BLOSUM and Boman distance—were injected into the deployed model. The aliphatic index shift was retained solely for descriptive analysis.

3.9.3 Mutation-Aware TCR Scoring and Ranking

The fine-tuned mutation-aware transformer model was applied in inference mode. For each neoantigen peptide p, all candidate CDR3β sequences $\{t_j\}_{j=1}^{M}$ from a curated TCR repertoire of 197,850 unique sequences were evaluated.

Each interaction was assigned a scalar score

$$s_j = f_\theta(t_j, p, \Delta p),$$

where f_θ denotes the fixed fine-tuned model and Δp contains the normalized BLOSUM and Boman mutation deltas.

For each neoantigen, the top-ranked TCR was defined as

$$t^\star(p) = \arg\max_{t_j \in \mathcal{T}} f_\theta(t_j, p, \Delta p).$$

The model output s_j represents a logit (raw score). For interpretability and threshold-based classification, logits were transformed into binding probabilities using the sigmoid function

$$\hat{y}_j = \sigma(s_j) = \frac{1}{1 + e^{-s_j}}.$$

Here, $\hat{y}_j \in (0, 1)$ denotes the predicted probability of binding under the trained binary classifier. The default decision boundary corresponds to $s_j \geq 0$ (equivalently $\hat{y}_j \geq 0.5$).

Chapter 4
Results and Discussion

This chapter evaluates the proposed controlled negative sampling framework for TCR–peptide binding prediction and examines its ability to support generalizable representation learning under Grazioli-compliant evaluation settings. The primary objective is not simply to maximize performance on a fixed dataset, but to determine whether biologically informed negative construction produces models whose ranking behavior remains stable when tested on previously unseen peptides. Particular emphasis is placed on precision-oriented metrics that reflect realistic prioritization scenarios in immunological discovery.

Beyond general TCR–peptide interaction prediction, the ultimate goal of this work is neoantigen-specific TCR prioritization for cancer immunotherapy. The neoantigen fine-tuning dataset and the NSCLC/COSMIC–NetMHCpan pipeline were introduced in Chapter 3. The backbone model developed and validated in this chapter is subsequently fine-tuned via transfer learning on the curated neoantigen dataset and applied to score candidate CDR3β sequences. The analyses presented here first establish the generalization capacity of the backbone model before mutation-aware specialization.

Two complementary evaluation settings are used throughout this chapter. First, the controlled-negative datasets defined in Chapter 3 isolate the effect of negative sample similarity under Grazioli-compliant conditions. Second, the external baseline dataset (`ds.csv`; Section 3.3.6 serves as a heterogeneous control to assess whether observed performance trends persist under alternative supervision regimes.

After identifying a stable backbone configuration under these evaluation settings, robustness is further assessed through loss-function ablation. The model is then specialized via mutation-aware fine-tuning on the curated neoantigen (Neo/WT) dataset. Finally, the mutation-aware model is deployed in an independent NSCLC case study, where COSMIC-derived neoantigens filtered using NetMHCpan are prioritized and matched to candidate CDR3β sequences without further retraining.

4.1 Datasets Used in Evaluation

All experiments in this chapter use the positive interaction set, controlled-negative datasets, and external baseline dataset defined in Chapter 3 (Sections 3.2.1, 3.2.4, and 3.3.6). Dataset composition, source distributions, and summary statistics are reported there.

4.2 Impact of Negative Sampling Strategy on Aggregate Performance

We next examine how negative sample construction influences model robustness under Grazioli-compliant evaluation. Figures 4.1 and 4.2 summarize AUROC and AUPR across random split, hard split, and distance-controlled splits using both BLOSUM- and RMSD-based distance definitions.

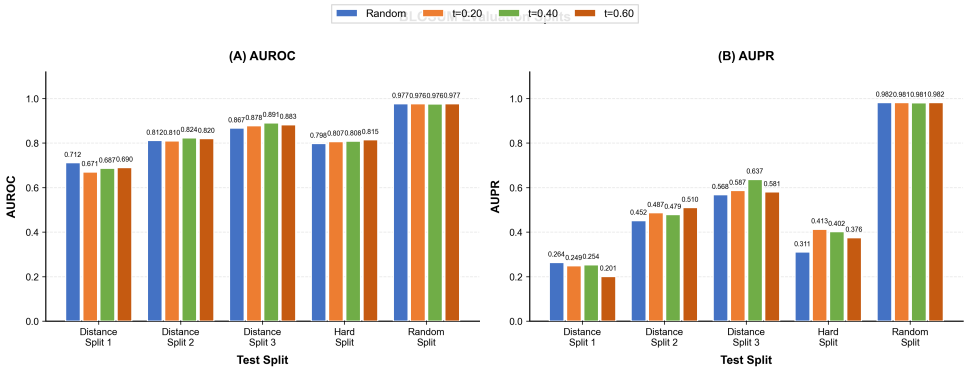

Figure 4.1. Performance comparison under BLOSUM-based splits.

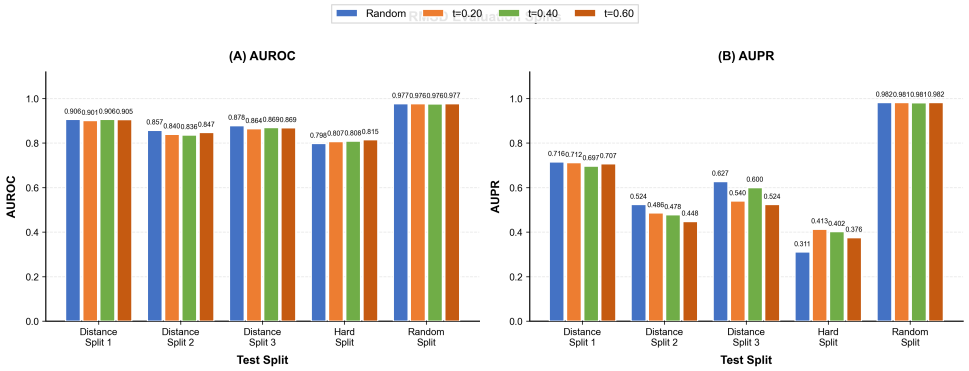

Figure 4.2. Performance comparison under RMSD-based splits.

4.2.1 Global Performance Trends Across Splits

Across all negative sampling regimes, performance is highest under the random split (AUROC ≈ 0.97; AUPR ≈ 0.98), reflecting peptide overlap between training and test sets. As expected, performance decreases under hard and distance-based splits, confirming that peptide-level separation induces a genuine distribution shift rather than trivial memorization.

Importantly, AUROC differences across negative sampling strategies are relatively modest. For example, under BLOSUM Distance Split 3, AUROC varies only within a narrow range (0.867–0.891). This indicates that coarse separability between binders and non-binders remains broadly similar across sampling strategies.

In contrast, AUPR reveals substantially larger differences, particularly under challenging splits. Because TCR–peptide prediction operates under severe class imbalance, AUPR provides a more sensitive measure of early retrieval quality and ranking robustness.

4.2.2 BLOSUM-Based Distance Splits

Under BLOSUM-based splits (Figure 4.1), AUPR increases progressively from Distance Split 1 to Distance Split 3 across most sampling strategies. For example, for $t = 0.40$, AUPR rises from 0.254 (DS1) to 0.637 (DS3). This behavior is consistent with the observations of Grazioli et al., who reported that increasing sequence distance does not necessarily make the task harder and may even improve generalization performance in some regimes.

A likely explanation is that sequence-based dissimilarity alone does not fully capture structural divergence. When peptides are sequence-distant but still structurally compatible, models may rely on higher-level interaction features rather than superficial sequence similarity. Thus, BLOSUM-based distance shifts may not impose the same level of extrapolation difficulty as structural shifts.

Notably, among clustering thresholds, $t = 0.40$ consistently achieves the highest AUPR under the most challenging BLOSUM regime (Distance Split 3: 0.637), outperforming both random negatives (0.568) and more extreme clustering thresholds. This suggests that intermediate clustering reduces ambiguous near-neighbor negatives while preserving sufficient peptide diversity.

4.2.3 RMSD-Based Distance Splits: Structural Generalization Challenge

In contrast, RMSD-based splits (Figure 4.2) exhibit a different pattern. Here, performance degradation is more pronounced under structurally defined distribution shifts. Compared to BLOSUM splits, AUPR values under RMSD Distance Split 2 and Distance Split 3 are lower and less monotonic.

This behavior aligns with the central finding of Grazioli et al.: structural (shape-based) distance induces a more severe generalization challenge than sequence distance. Because RMSD captures three-dimensional conformational divergence rather than amino-acid substitution patterns alone, RMSD-based splits require the model to extrapolate beyond both sequence and structural similarity regimes.

Importantly, the hardest overall regime remains the hard split combined with structural

separation, where AUPR drops substantially. This confirms that complete peptide exclusion from training remains a stronger constraint than percentile-based sequence distance alone.

4.2.4 What Makes This Dataset Optimal for Generalization?

Across both BLOSUM and RMSD splits, the controlled dataset constructed at clustering threshold $t = 0.40$ consistently demonstrates the most favorable balance between discrimination and robustness. Under BLOSUM Distance Split 3, $t = 0.40$ achieves the highest AUPR (0.637). Under RMSD splits and the hard split, it maintains competitive or superior performance relative to other thresholds.

This pattern can be interpreted as a trade-off between label noise and negative diversity:

- At low thresholds ($t = 0.20$), clustering fragments the peptide space excessively, reducing diversity and potentially encouraging overly coarse discrimination.
- At high thresholds ($t = 0.60$), near-neighbor peptides are admitted into the negative set. Given the known cross-reactivity of TCR recognition, some of these peptides may represent biologically plausible binders, introducing supervision ambiguity.
- The intermediate threshold ($t = 0.40$) appears to optimally exclude highly similar peptides while preserving structural diversity, thereby reducing label uncertainty without oversimplifying the learning problem.

Thus, the improvement observed at $t = 0.40$ is consistent with reduced negative-label ambiguity combined with preserved generalization capacity, rather than artificial task simplification.

4.3 Comparative Performance of Model Architectures and Encodings

Figures 4.3 and 4.4 compare CNN, LSTM, and Transformer architectures across five sequence encoding schemes under a hard peptide split using the controlled $t = 0.40$ dataset.

4.3.1 Architecture-Level Comparison

Across encodings, performance differences between architectures are more pronounced in AUPR than in AUROC. Transformer-based models consistently achieve the highest AUPR, whereas AUROC values remain comparatively similar across architectures. For example, with AAIndex encoding, AUROC ranges from 0.492 (CNN) to 0.578 (Transformer), while AUPR ranges from 0.657 (CNN) to 0.710 (Transformer). Notably, CNN combined with physicochemical encoding achieves the highest AUROC (0.598), slightly exceeding the Transformer configurations in global separability.

The modest spread in AUROC suggests that all architectures learn broadly similar decision boundaries separating binders from non-binders. However, the larger spread in AUPR indicates meaningful differences in ranking concentration under class imbalance. In particular, Transformers place a greater fraction of true binders among the highest-scoring predictions, even when overall separability remains comparable.

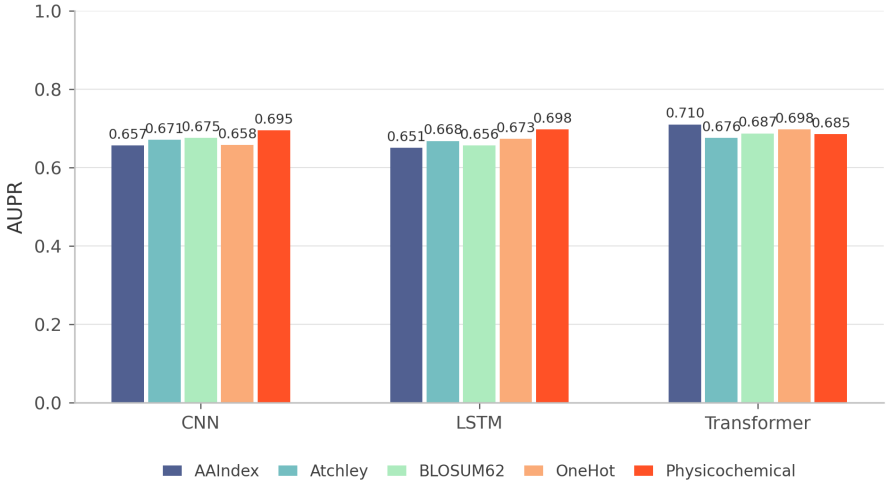

Figure 4.3. AUPR comparison across architectures and encodings under the hard peptide split.

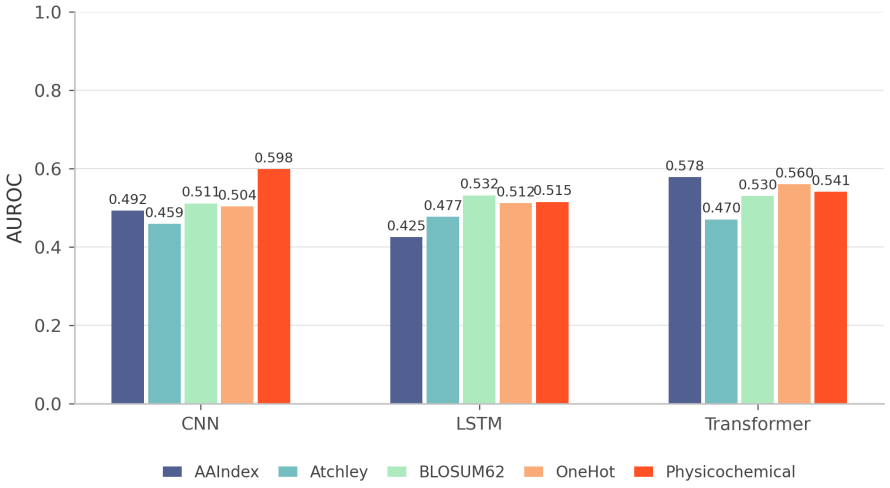

Figure 4.4. AUROC comparison across architectures and encodings under the hard peptide split.

4.3.2 Interpreting the AUROC–AUPR Divergence

The divergence between AUROC and AUPR reflects the distinct sensitivities of the two metrics. AUROC evaluates ranking performance across the entire score distribution and is relatively insensitive to class prevalence. In contrast, AUPR emphasizes precision among top-ranked predictions and is directly influenced by the positive class proportion.

Given the approximately 1:9 class imbalance in the controlled dataset, AUROC can remain stable even if precision among high-confidence predictions varies. AUPR is therefore more informative in this setting, as it reflects performance in the region most relevant for downstream candidate prioritization.

4.3.3 Why CNN + Physicochemical Performs Strongly in AUROC

Physicochemical encoding explicitly embeds residue-level properties such as hydrophobicity, polarity, and steric volume—features known to correlate with general binding propensity. When combined with convolutional filters that efficiently detect short local motifs, this representation appears effective at learning coarse biochemical discriminators. This likely explains the marginal AUROC advantage observed for CNN + physicochemical encoding: the model separates classes well in aggregate by exploiting local biochemical patterns.

4.3.4 Why Transformer + AAIndex Improves AUPR

In contrast, Transformer + AAIndex achieves the highest AUPR across configurations. AAIndex provides curated substitution and structural similarity descriptors, enabling smoother physicochemical continuity in the input space. The Transformer architecture, through self-attention, models long-range and distributed residue interactions rather than relying primarily on local motifs.

Under a hard peptide split—where test peptides are entirely unseen during training—models must generalize beyond memorized motif patterns. Distributed interaction modeling becomes more important for correctly ranking novel peptide–TCR combinations at the top of the score distribution. The improved AUPR therefore suggests that Transformer + AAIndex more effectively concentrates true binders among high-confidence predictions, even if its global separability is similar to that of CNN + physicochemical encoding.

4.3.5 Model Selection Rationale

Although CNN + physicochemical encoding achieves slightly higher AUROC, the difference is small and confined to global class separability. Transformer + AAIndex provides consistently superior AUPR, which directly reflects early precision under class imbalance and is more aligned with the practical objective of TCR prioritization.

Because TCR–peptide prediction is inherently a ranking task in which only a limited number of top candidates can be experimentally validated, early retrieval performance is prioritized over marginal improvements in overall separability. For this reason, AUPR is treated as the primary selection criterion.

Based on these considerations, the Transformer + AAIndex configuration is selected for hyperparameter optimization and subsequent neoantigen-specific fine-tuning.

4.4 Hyperparameter Optimization

4.4.1 Hyperparameter Optimization on the Controlled Dataset

Hyperparameter optimization was performed for the Transformer + AAIndex configuration using the training and validation subsets of a fixed hard split of the $t = 0.40$ controlled-negative dataset. The held-out test split was not accessed during optimization. The search space jointly explored architectural parameters (hidden dimension, number of layers, attention heads), optimization settings (learning rate, weight decay, dropout), and imbalance-handling strategies (training class ratio and loss weighting), rather than fixing these a priori.

Figure 4.5 illustrates the validation AUROC trajectory across trials.

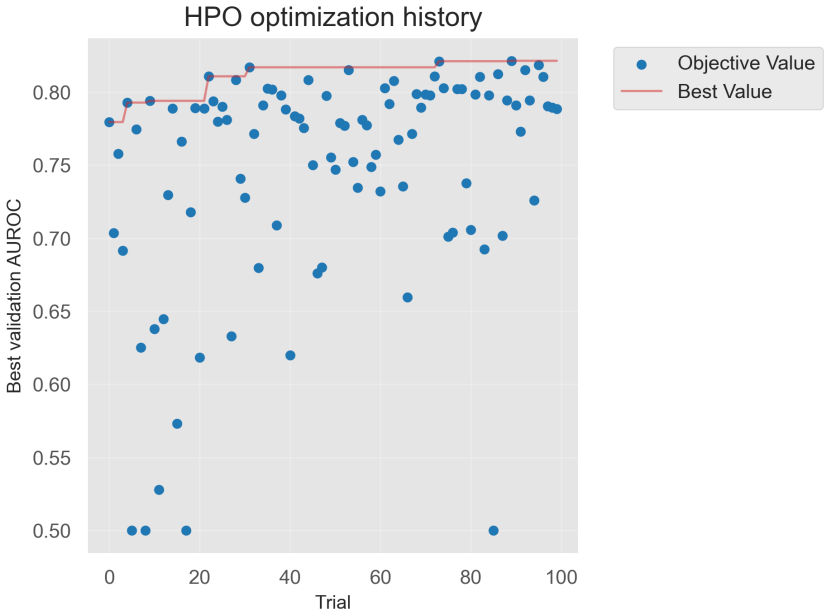

Figure 4.5. Optimization history for HPO on the controlled $t = 0.40$ dataset.

Validation AUROC increases rapidly within the first set of trials and subsequently stabilizes around a narrow performance band. The best-performing configuration is identified early in the search, and later trials predominantly explore nearby parameterizations with comparable validation performance.

This pattern indicates a relatively smooth and well-behaved optimization landscape. Rather than requiring extreme hyperparameter tuning to reach competitive performance, multiple configurations converge to similar AUROC values. Such behavior is consistent with stable supervision: when negative samples are constructed to reduce boundary ambiguity, the effective signal-to-noise ratio during training increases, and model performance becomes less sensitive to precise architectural choices.

Notably, the full HPO procedure on the controlled dataset required approximately 22 hours and 46 minutes to complete. Given the number of trials and early convergence behavior, this runtime reflects both computational cost and the fact that many trials

reached pruning or early stopping conditions relatively quickly once the performance plateau was identified.

The full hyperparameter search space is detailed in Appendix Section A.1.

4.4.2 Hyperparameter Optimization on the Baseline `ds.csv` Dataset

To evaluate whether the observed optimization dynamics were specific to the controlled-negative construction, the identical HPO procedure was repeated on the larger baseline dataset `ds.csv`, previously introduced in Section 3.3.6.

Figure 4.6 shows the corresponding validation AUROC trajectory.

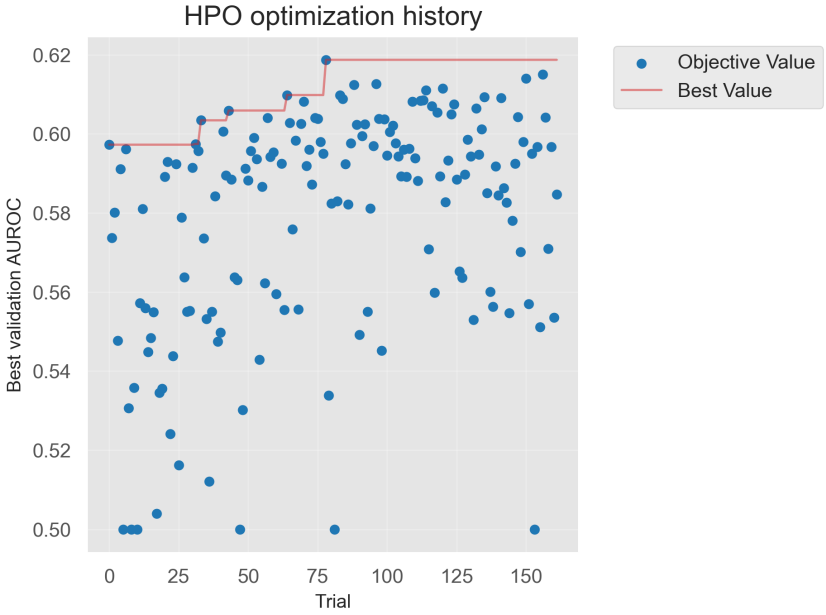

Figure 4.6. Optimization history for HPO on the `ds.csv` baseline dataset.

In contrast to the controlled dataset, optimization on `ds.csv` exhibits greater variance across trials and a slower approach to its performance ceiling. Although validation AUROC improves over trials, it saturates at a lower maximum value compared to the $t = 0.40$ dataset, and high-performing configurations are more sparsely distributed throughout the search space.

Importantly, the HPO procedure on `ds.csv` required approximately **70 hours and 32 minutes**, more than three times longer than the controlled dataset optimization. While part of this increase is attributable to the larger dataset size and higher negative-to-positive ratio, the extended runtime also reflects slower convergence and fewer early-pruned trials reaching stable plateaus quickly.

Despite containing substantially more samples, `ds.csv` does not yield higher validation AUROC under identical optimization conditions. This suggests that dataset size alone does not determine learnability. The baseline dataset includes randomly generated and assay-derived negatives as well as a stronger class imbalance, which may introduce greater supervision ambiguity near the decision boundary. Under such conditions, the

optimization landscape becomes more irregular: small changes in hyperparameters can lead to larger fluctuations in validation performance, and high-performing regions occupy a smaller fraction of the search space. This contrast is visually supported by the parallel-coordinate plots provided in Appendix Section A.2.

4.5 Summary of HPO Findings

Taken together, the HPO experiments reveal systematic differences between the controlled-negative dataset and the baseline dataset:

- The controlled $t = 0.40$ dataset yields rapid convergence, a higher validation AUROC ceiling, and a relatively narrow band of near-optimal configurations.
- The baseline `ds.csv` dataset exhibits slower convergence, greater variance across trials, and a lower performance ceiling despite its larger size.
- Optimization on the controlled dataset completes substantially faster (22h46m vs. 70h32m), reflecting both reduced computational burden and a smoother effective optimization landscape.

These observations suggest that negative sample structure plays a critical role not only in evaluation performance but also in shaping the geometry of the optimization process. By reducing near-neighbor ambiguity and constraining negative similarity, the controlled-negative dataset is consistent with an increase in the effective signal-to-noise ratio during training. This is associated with more stable model selection dynamics and improved validation performance under hard peptide splits.

Based on these results, the best-performing configuration obtained from the $t = 0.40$ controlled dataset is selected as the reference model for subsequent ablation studies and neoantigen-specific fine-tuning.

4.5.1 AUROC Comparison Across Models After HPO

To contextualize the effect of hyperparameter optimization under controlled supervision, we compared the HPO-optimized Transformer+AAIndex model trained on the $t = 0.40$ controlled-negative dataset against (i) the same architecture trained on the heterogeneous `ds.csv` baseline, and (ii) two established baselines, NetTCR-2.0 and ERGO2.

Figure 4.7 presents AUROC comparisons under BLOSUM-based distance splits, while Figure 4.8 shows the corresponding results under RMSD-based structural splits.

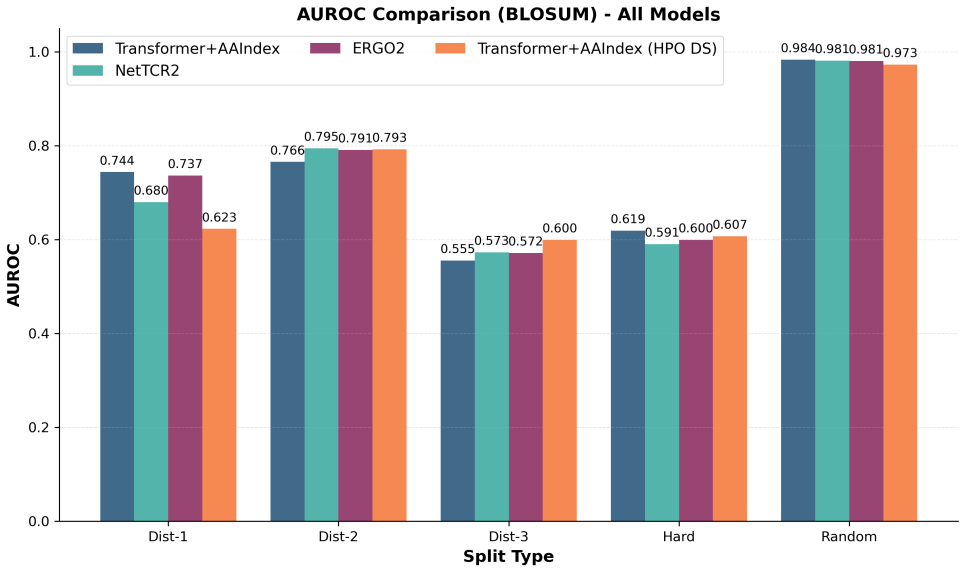

Figure 4.7. AUROC comparison across models under BLOSUM-based distance splits after hyperparameter optimization. The Transformer+AAIndex model trained on the controlled $t = 0.40$ dataset is compared to the same architecture trained on the heterogeneous `ds.csv` baseline, as well as NetTCR-2.0 and ERGO2. Sequence-level distance splits (Dist-1 to Dist-3) and the hard peptide split evaluate generalization under increasing similarity constraints.

Under BLOSUM distance regimes, the HPO-optimized controlled model consistently achieves competitive or superior AUROC across peptide-shifted splits (Dist-1, Dist-2, Dist-3, and Hard). In particular, performance degradation under increasingly strict distance regimes remains moderate relative to the baseline dataset, indicating improved stability under sequence-level distribution shift. Random splits yield near-ceiling AUROC for all models, confirming that identity-level overlap inflates performance and does not meaningfully discriminate architectures.

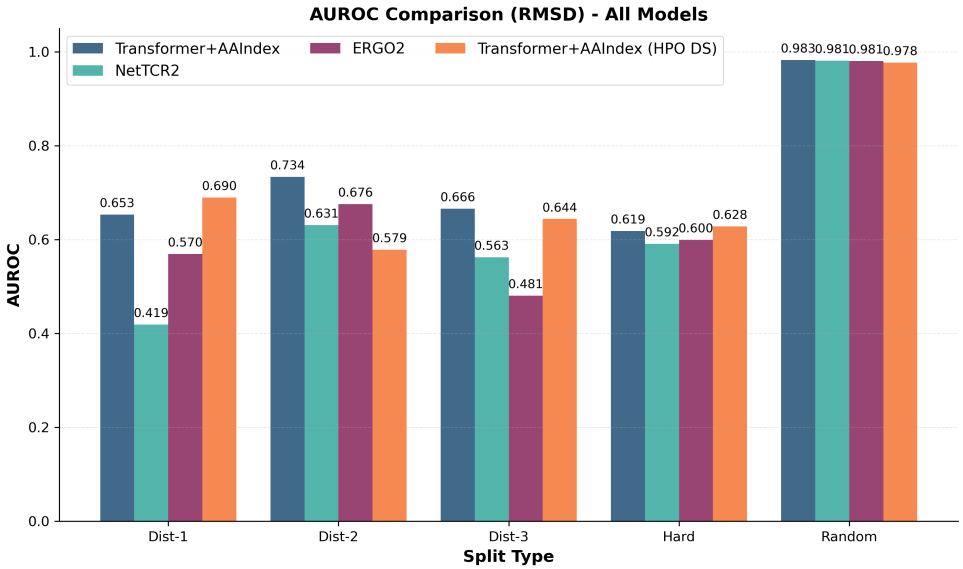

Figure 4.8. AUROC comparison across models under RMSD-based structural distance splits after hyperparameter optimization. Structural splits evaluate robustness to three-dimensional peptide dissimilarity. The controlled-negative HPO model maintains more stable AUROC under structural shift compared to the heterogeneous baseline and prior architectures.

Under RMSD-based structural splits, performance differences become more pronounced. The controlled HPO model maintains stronger AUROC under Dist-1 and Dist-3 compared to NetTCR2 and ERGO2, suggesting improved robustness to structural dissimilarity. In contrast, the baseline `ds.csv` model exhibits greater variability and a more pronounced drop under structural shifts. These findings reinforce that negative-sample structure influences not only validation dynamics during optimization (Section 4.4.1) but also downstream generalization behavior.

Notably, although NetTCR2 and ERGO2 achieve comparable AUROC under Random splits, their performance declines more sharply under peptide- and structure-controlled regimes. This pattern is consistent with the hypothesis that similarity-uncontrolled negative construction may lead to decision boundaries that are less stable under distribution shift.

Together, these results indicate that hyperparameter optimization amplifies the benefits of similarity-controlled supervision rather than compensating for heterogeneous negative construction. The controlled $t = 0.40$ dataset yields a model that is both easier to optimize (Section 4.4.1) and more stable under peptide-level generalization tests.

4.6 Loss Function Ablation

Complete numerical results for AUROC, AUPR, and Accuracy across all loss variants and evaluation splits are provided in Appendix A.4 (Tables A.1–A.6).

To disentangle the effect of loss formulation from architectural and encoding choices, we conducted a systematic ablation over loss variants using the HPO-selected Transformer + AAIndex configuration. The following loss functions were evaluated:

- BCE-with-logits (BCEWithLogitsLoss)
- BCE-with-logits + resampling
- Focal loss (mild configuration)
- Focal loss (standard configuration)

Performance was evaluated across all split types under both BLOSUM- and RMSD-based distance regimes. Results are summarized in Figures 4.9 and 4.10.

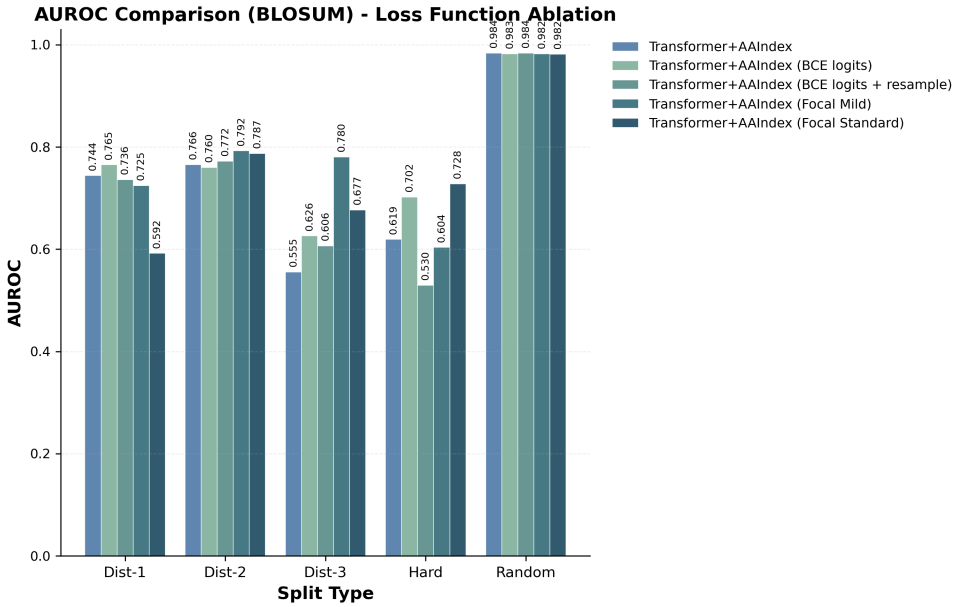

Figure 4.9. Loss function comparison under BLOSUM-based splits (AUROC).

4.6.1 Results Under BLOSUM Distance Splits

Under BLOSUM-based splits, AUROC values reveal distinct robustness patterns across loss formulations as peptide similarity decreases.

- *Hard split:* BCE-with-logits achieves 0.702, substantially outperforming resampling (0.530) and focal mild (0.604), and remaining close to focal standard (0.728). Notably, the resampling variant exhibits pronounced degradation, indicating instability under peptide-level distribution shift.
- *Distance_split_1:* BCE-with-logits reaches 0.765, exceeding focal standard (0.592), focal mild (0.725), and resampling (0.736), demonstrating strong robustness under moderate sequence separation.
- *Distance_split_2:* Focal mild (0.792) and focal standard (0.787) slightly outperform BCE-with-logits (0.760). However, this improvement is split-specific and does not persist under harder evaluation regimes.

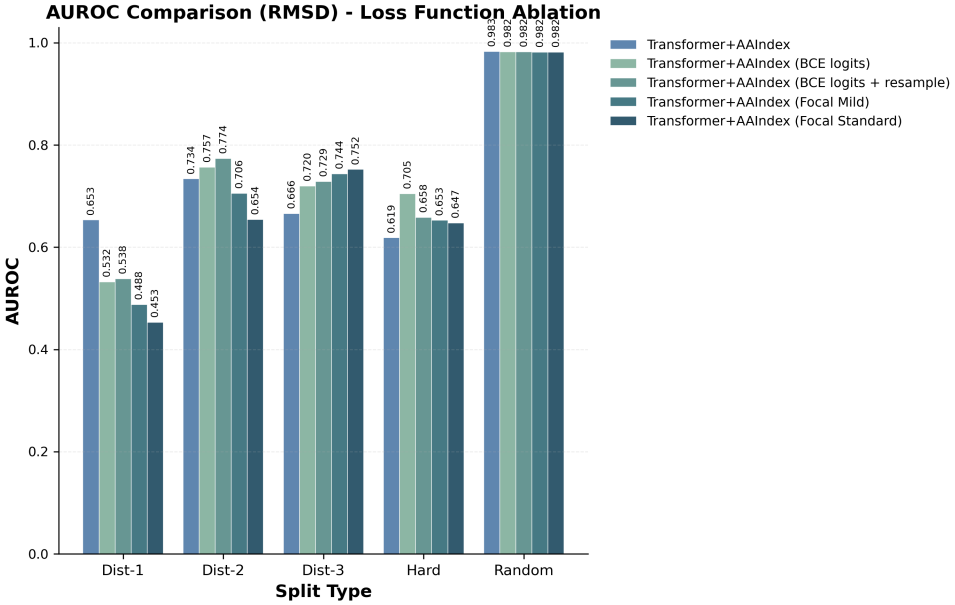

Figure 4.10. Loss function comparison under RMSD-based splits (AUROC).

- *Distance_split_3:* Focal mild attains 0.781, whereas BCE-with-logits reaches 0.626. Although focal mild performs strongly here, its performance varies considerably across other splits, whereas BCE-with-logits maintains moderate but stable behavior.
- *Random split:* All loss variants achieve near-ceiling AUROC ≈ 0.98, indicating that differences between loss functions largely vanish when peptide overlap between training and testing is allowed. This confirms that loss choice primarily influences robustness under distribution shift rather than performance in the memorization-friendly regime.

Overall, BCE-with-logits exhibits the most consistent AUROC behavior across progressively harder BLOSUM splits. In contrast, focal variants display split-dependent variability, occasionally achieving higher AUROC but lacking stability across evaluation regimes.

Detailed numerical results are provided in Appendix A.4 (Tables A.1–A.3).

4.6.2 Results Under RMSD Distance Splits

Under RMSD-based splits, which impose structural rather than purely sequence-level separation, similar but slightly attenuated stability patterns are observed.

- *Hard split:* BCE-with-logits achieves 0.705, outperforming focal mild (0.653), focal standard (0.647), and resampling (0.658), indicating improved robustness under structural distribution shift.
- *Distance_split_1:* BCE-with-logits (0.532) performs comparably to resampling (0.539)

and clearly exceeds focal mild (0.488) and focal standard (0.453), which show substantial degradation under this structural constraint.

- *Distance_split_2:* Resampling (0.774) slightly outperforms BCE-with-logits (0.757), while focal variants remain lower (0.706 and 0.654). As in the BLOSUM setting, localized improvements do not generalize consistently across splits.
- *Distance_split_3:* Focal standard reaches 0.752 and focal mild 0.744, with BCE-with-logits remaining competitive at 0.720. However, this advantage does not extend to harder splits such as the RMSD hard split.
- *Random split:* All loss functions again converge to ≈ 0.98, confirming that loss formulation has negligible effect when structural overlap between train and test peptides exists.

Across RMSD splits, BCE-with-logits avoids severe performance collapse and maintains comparatively stable AUROC under increasing structural dissimilarity, whereas focal and resampling strategies show more pronounced split-dependent fluctuations.

Corresponding results are reported in Appendix A.4 (Tables A.4–A.6).

4.6.3 Interpretation

The differential behavior of the evaluated loss functions can be interpreted in the context of Grazioli-compliant peptide-shifted evaluation, which introduces distributional variation between training and test partitions by construction. Under such conditions, models must maintain stable global discrimination despite changes in peptide similarity structure and class composition.

Focal loss and resampling strategies adapt optimization dynamics by amplifying specific subsets of training examples. While such mechanisms can improve separability under homogeneous distributions, they may yield split-dependent performance when evaluated under varying similarity regimes.

In contrast, BCE-with-logits optimizes a calibrated probabilistic objective without dynamically altering gradient scaling or effective class priors. This produces smoother decision boundaries and more consistent global ranking performance across both sequence- and structure-based splits.

Overall, BCE-with-logits demonstrates the most reproducible AUROC behavior under Grazioli-compliant evaluation, suggesting that stability-oriented objectives are advantageous under peptide-level distribution shift.

4.7 Final Model Choice

Based on architecture comparison, encoding evaluation, hyperparameter optimization, and loss-function ablation, the final selected configuration is:

- **Architecture:** Transformer
- **Encoding:** AAIndex (5-dimensional)
- **Hidden dimension:** 256

- **Transformer layers:** 2
- **Attention heads:** 16
- **Feedforward dimension:** 256
- **Dropout:** 0.125
- **Loss function:** BCE-with-logits
- **Batch size:** 64
- **Learning rate:** 1.02×10^{-5}
- **Weight decay:** 5.6×10^{-6}
- **Evaluation regime:** Hard split of the $t = 0.40$ controlled-negative dataset

This configuration demonstrates comparatively stable AUROC across similarity regimes while maintaining robustness under peptide-level distribution shift.

4.8 Benchmarking Against Established Methods

We next compared the final selected configuration (Transformer + AAIndex + BCE-with-logits) against established TCR–peptide prediction models, including NetTCR-2.0 and ERGO-II, under Grazioli-compliant BLOSUM- and RMSD-based splits. Results are shown in Figures 4.11 and 4.12.

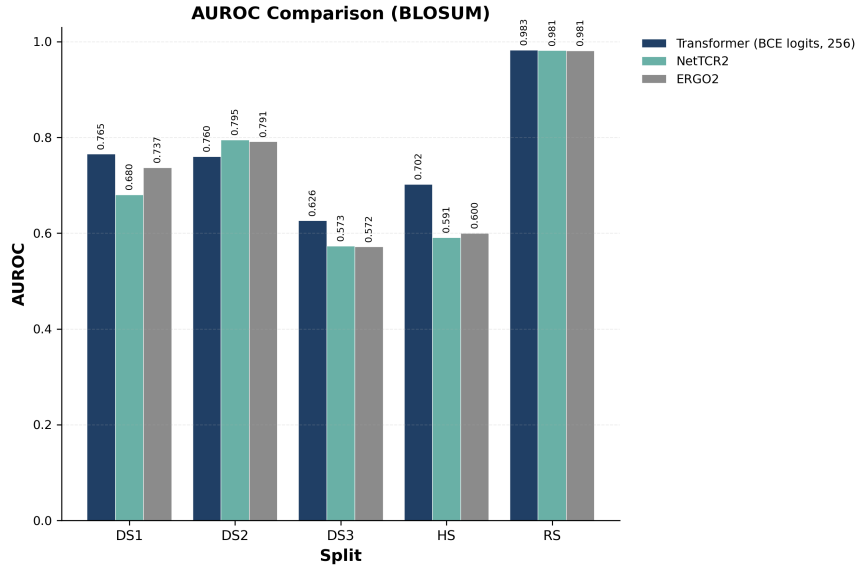

Figure 4.11. AUROC comparison under BLOSUM-based splits.

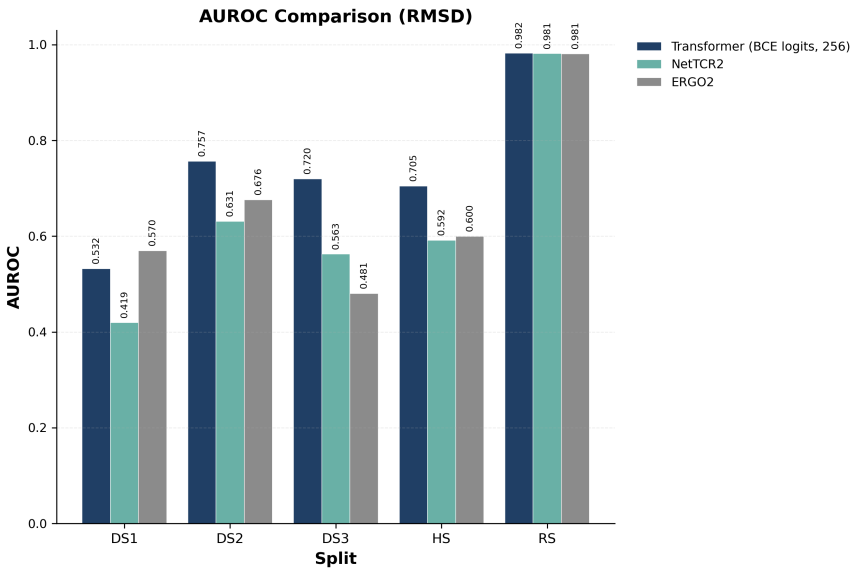

Figure 4.12. AUROC comparison under RMSD-based splits.

4.8.1 BLOSUM-Based Splits

Under sequence-based (BLOSUM) splits, the proposed model achieves the highest AUROC in three of five evaluation regimes (DS1, DS3, and Hard Split), while remaining competitive under DS2 and Random Split.

Specifically:

- *DS1:* The proposed model attains 0.765, outperforming NetTCR-2.0 (0.680) and ERGO-II (0.737).
- *DS2:* NetTCR-2.0 (0.795) and ERGO-II (0.791) slightly exceed the proposed model (0.760), though differences remain moderate.
- *DS3:* The proposed model achieves 0.626, improving upon both NetTCR-2.0 (0.573) and ERGO-II (0.572).
- *Hard Split:* The proposed configuration reaches 0.702, substantially exceeding NetTCR-2.0 (0.591) and ERGO-II (0.600).
- *Random Split:* All models approach ceiling performance ($\approx$0.98), indicating minimal separability differences when peptide overlap is allowed.

Performance gains are most pronounced under the Hard Split and DS3, which impose stronger peptide-level dissimilarity. This pattern is consistent with improved robustness under sequence-based distribution shift.

4.8.2 RMSD-Based Splits

Under structure-based (RMSD) splits, the proposed model achieves the strongest AUROC in four of five evaluation regimes.

- *DS1:* ERGO-II (0.570) exceeds the proposed model (0.532), though the latter remains above NetTCR-2.0 (0.419).
- *DS2:* The proposed model achieves 0.757, outperforming NetTCR-2.0 (0.631) and ERGO-II (0.676).
- *DS3:* The proposed model attains 0.720, substantially exceeding NetTCR-2.0 (0.563) and ERGO-II (0.481).
- *Hard Split:* The proposed configuration reaches 0.705, improving upon NetTCR-2.0 (0.592) and ERGO-II (0.600).
- *Random Split:* All models again converge near 0.98.

Notably, improvements under DS2, DS3, and Hard splits indicate that the proposed model maintains global separability under structural dissimilarity constraints, not merely under sequence-based similarity.

4.8.3 Interpretation

Across sequence-based (BLOSUM) similarity splits, the proposed configuration outperforms established state-of-the-art models (NetTCR-2.0 and ERGO-II) in the majority of evaluation regimes, particularly under the more stringent DS3 and Hard split conditions. Under these peptide-level distribution shifts, performance gains are most pronounced, whereas all models converge under random splits, indicating that differences are not attributable to memorization effects.

Importantly, these improvements should not be interpreted as evidence of any single component in isolation. Rather, they are consistent with the interaction of three complementary design choices: (i) Transformer-based global interaction modeling, which enables distributed residue-level dependency learning beyond local motif detection; (ii) the incorporation of physicochemical prior knowledge through AAIndex encoding, which provides structured biochemical information beyond raw sequence representation; and (iii) the controlled negative sampling framework, which reduces similarity-driven supervision artifacts and aligns training with Grazioli-compliant evaluation. Together, these elements contribute to improved robustness under sequence-based distribution shift.

4.9 NetTCR-2.0 Ablation Results

The NetTCR-2.0 ablation (Section 3.7) compared encoder and architecture variants under Grazioli-compliant evaluation. Figures 4.13 and 4.14 show AUROC across BLOSUM- and RMSD-based splits for the following configurations:

- **Original NetTCR-2.0 (BLOSUM + CNN):** The baseline architecture using BLOSUM50 encoding and convolutional layers as proposed in the original model.
- **Encoder substitution (AAIndex + CNN):** The original CNN architecture combined with AAIndex physicochemical encoding instead of BLOSUM.
- **Architecture substitution (BLOSUM + Transformer):** Replacement of the CNN with a transformer architecture while retaining BLOSUM encoding.

- **Combined substitution (AAIndex + Transformer):** Simultaneous replacement of both the encoding scheme and architecture.
- **Proposed final model (AAIndex + Transformer + BCE-with-logits):** The fully optimized configuration including contextual modeling, physicochemical encoding, regularization, and hyperparameter tuning under Grazioli-compliant splits.

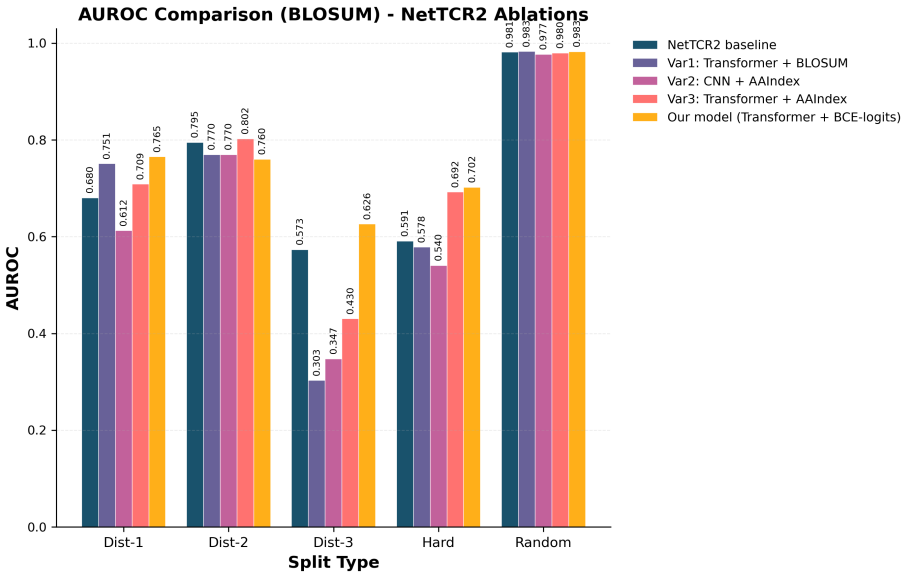

Figure 4.13. NetTCR-2.0 ablation: AUROC under BLOSUM-based splits. Comparison of encoder and architecture variants.

4.9.1 Ablation Under BLOSUM Distance Splits

Under sequence-based distance splits (Figure 4.13), clear robustness differences emerge as peptide similarity decreases.

The original NetTCR-2.0 configuration shows moderate AUROC in Distance_split_1 and Distance_split_2, but declines in Distance_split_3, indicating sensitivity to stronger sequence-level separation. Replacing the CNN with a transformer while retaining BLOSUM encoding produces split-dependent behavior. Performance improves in Distance split 1 and Distance split 2, where peptide dissimilarity remains moderate. However, in Distance split 3 and Hard split—corresponding to the highest levels of sequence separation—performance declines relative to the original CNN baseline.

Substituting AAIndex for BLOSUM while preserving the CNN architecture yields improvement in Distance Split 3 compared with Transformer + BLOSUM, but does not consistently outperform the baseline across all splits.

The combined Transformer + AAIndex variant further improves robustness in Distance split 1, 2 and hard split, yet performance remains worse than baseline in Distance split 3.

In contrast, the proposed final model achieves the most consistent AUROC profile across Distance split 1, Distance split 3, and Hard split, while remaining competitive in Distance

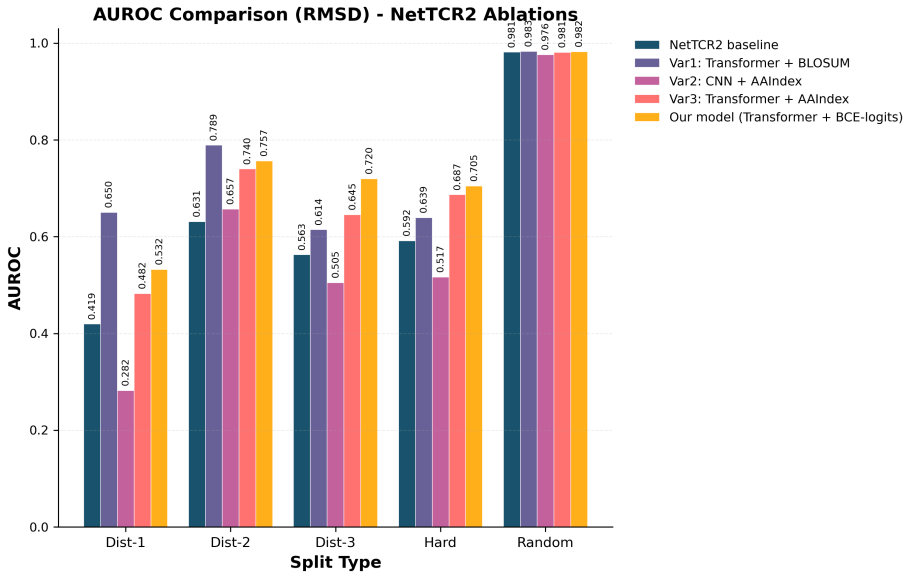

Figure 4.14. NetTCR-2.0 ablation: AUROC under RMSD-based splits. Comparison of encoder and architecture variants.

split 2 and Random split. This indicates enhanced stability under increasing sequence dissimilarity.

4.9.2 Ablation Under RMSD Distance Splits

Under RMSD-based splits (Figure 4.14), which enforce structural rather than purely sequence-level separation, robustness differences become more pronounced.

The original NetTCR-2.0 baseline exhibits marked degradation in Distance split 1, reflecting sensitivity to structural distribution shift. Introducing a transformer architecture (with either BLOSUM or AAIndex encoding) improves performance in several RMSD splits, particularly Distance split 2 and Distance split 3, indicating improved adaptability to structural dissimilarity.

However, none of the intermediate ablation variants uniformly dominate across all splits. The proposed final model maintains the most stable AUROC behavior across Distance split 3 and Hard split, while preserving competitive performance in Random split. Importantly, the model avoids the pronounced collapse observed in certain CNN-based configurations under stronger structural separation.

4.9.3 Interpretation

Under BLOSUM-based splits, the Transformer + BLOSUM variant exhibits improved AUROC in Distance split 1 and Distance split 2 relative to the original CNN baseline. However, this improvement does not persist under stronger distribution shift. In Distance_split_3 and Hard split—corresponding to the highest peptide dissimilarity—the same variant performs worse than the original NetTCR-2 configuration.

This split-dependent behavior indicates that architectural substitution alone does not

ensure robustness under increasing sequence separation. While contextual modeling may capture sequence patterns effectively in moderate similarity regimes, these gains do not consistently translate into generalization under stronger dissimilarity constraints. The degradation observed in Distance split 3 and Hard split is consistent with regime-specific fitting, where performance improvements on simpler splits do not extend to more difficult evaluation settings.

A structural comparison between the NetTCR-2 variants and the proposed final model reveals several relevant differences.

First, the original NetTCR-2 CNN architecture employs five parallel convolutional branches per modality (kernel sizes 1, 3, 5, 7, 9), followed by global max pooling and a shallow 32-unit classification layer. These multi-scale convolutional filters act as localized motif detectors. While effective under moderate similarity, such fixed receptive fields may be sensitive to training distribution characteristics and less adaptable when peptide similarity between train and test sets is substantially reduced.

Second, several architectural and capacity-related differences distinguish the NetTCR-2 variants from the proposed final model. While the Transformer variants of NetTCR-2 introduce contextual modeling, they retain a relatively shallow classification head and shorter maximum sequence lengths. In contrast, the proposed final model incorporates a deeper classification module, extended sequence lengths, and hyperparameters selected through systematic optimization under Grazioli-compliant evaluation.

Third, the negative sampling strategies differ substantially. NetTCR-2 constructs negative samples by combining experimentally validated non-binding TCR–peptide pairs with artificially mismatched (non-cognate) TCR–peptide pairings, where artificial negatives are used during training. In contrast, the proposed model was optimized using a controlled negative sampling framework designed to reduce similarity-driven artifacts and better reflect evaluation under similarity-constrained splits. Hyperparameter optimization was performed directly under this controlled-negative regime.

These methodological differences suggest that improved robustness of the proposed model arises from the interaction of contextual modeling, classifier capacity, regularization strategy, and alignment between training objective and similarity-controlled evaluation. While certain NetTCR-2 variants improve AUROC in less restrictive splits, they do not consistently generalize to Distance_split_3 and Hard split. In contrast, the proposed model maintains more stable AUROC as peptide dissimilarity increases.

Overall, the results indicate that robustness under Grazioli-compliant evaluation depends not only on architectural design but also on negative sampling strategy and optimization alignment under distribution shift.

4.10 Feature Selection for Neo/WT Fine-Tuning

Following the architectural, optimization, and loss-function ablations reported in Sections 4.3.1 and 4.6, a final backbone configuration was established that demonstrated stable generalization under Grazioli-compliant evaluation (Section 3.4.3), particularly under the Hard split regime. Because the Hard split enforces complete peptide-level separation between training and evaluation sets, it most closely approximates the downstream clinical objective of prioritizing TCRs for previously unseen neoantigens.

Having validated that the backbone model learns similarity-robust TCR–peptide representations under controlled-negative supervision, we next investigated whether it could

be further specialized to distinguish neoantigens from their corresponding wild-type counterparts. This stage constitutes a refinement phase rather than a re-optimization of global binding prediction. The goal is not to increase AUROC under similarity-based splits, but to enhance *mutation-aware discrimination* within a paired Neo/WT framework using the curated dataset described in Section 3.8.

4.10.1 Delta Feature Ablation for Neo/WT Adaptation

As described in Section 3.8, three mutation-derived physicochemical delta features were computed for each Neo/WT pair: BLOSUM substitution distance, Boman index shift, and aliphatic index shift. These quantities quantify mutation-induced deviation from self along complementary biochemical axes. To determine the most informative feature subset under limited Neo/WT supervision, we conducted a systematic ablation across eight configurations:

- No delta features (sequence-only adaptation),
- Each single delta feature,
- All pairwise combinations,
- Full three-feature combination.

All configurations were trained under identical fine-tuning protocols (Section 3.8), including encoder freezing during warm-up, grouped CDR3β splits, differential learning rates, and BCE-with-logits optimization. Model selection was based on validation paired accuracy, and final evaluation was performed using **paired accuracy** on the held-out test set, defined as the proportion of CDR3β cases for which the predicted binding score of the neoantigen exceeds that of its corresponding wild-type peptide.

Figure 4.15 summarizes test paired accuracy across all configurations.

A clear performance hierarchy emerges. The **BLOSUM + Boman** combination achieves the highest paired accuracy (0.847), substantially outperforming both the sequence-only baseline (0.779) and the full three-feature configuration (0.656).

4.10.2 Focused Recovery with Conservative Backbone Updates

To specialize the proposed backbone model for mutation-specific discrimination, we fine-tuned the Transformer-based binding predictor (Section 4.7) on the curated Neo/WT dataset described in Section 3.8 using the transfer-learning strategy detailed in Section 3.8.7. Fine-tuning was initialized from the best-performing checkpoint obtained on the $t = 0.40$ controlled-negative dataset, thereby preserving the similarity-aware representations learned during backbone training.

Given the limited size and high sequence similarity of the Neo/WT dataset (Section 3.8), fine-tuning was conducted under conservative update constraints to minimize catastrophic forgetting. Specifically, the backbone encoder was warm-started and updated with a learning rate scaled to one-hundredth of the classification head (`backbone_lr` = `head_lr`/100), and early-stopping patience was extended to allow gradual convergence. Both the mutation-aware (BLOSUM + Boman) and sequence-only (no-delta) configurations were trained under identical schedules and evaluation conditions.

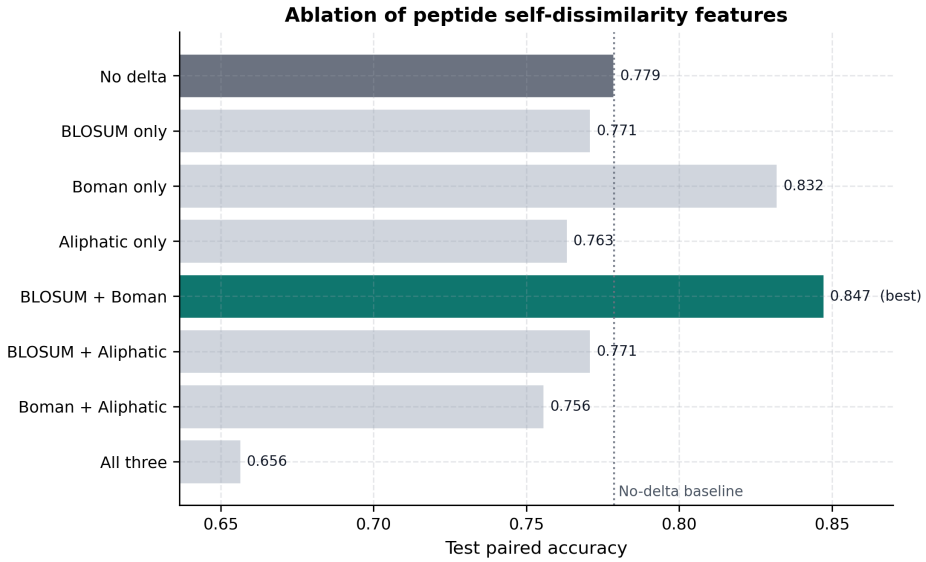

Figure 4.15. Ablation of mutation-aware delta feature subsets under paired Neo/WT evaluation. Test paired accuracy is shown for each configuration. The combination of BLOSUM and Boman distances yields the highest performance.

Figure 4.16 shows learning curves for validation paired accuracy, test paired accuracy, test AUROC, and test AUPR.

The mutation-aware BLOSUM + Boman configuration demonstrates consistently superior performance across all evaluation metrics. While the no-delta model converges rapidly and stops early (approximately 10 epochs), the BLOSUM + Boman variant exhibits a longer, more stable optimization trajectory, reaching peak validation paired accuracy at epoch 18 (0.847) and continuing to refine until early stopping at epoch 38.

At the selected checkpoint (epoch 18), the BLOSUM + Boman configuration achieves:

- Paired accuracy: 0.840
- Accuracy: 0.725
- AUROC: 0.788
- AUPR: 0.793
- Loss: 0.561

In contrast, the baseline configuration without mutation-aware delta features reaches:

- Paired accuracy: 0.794
- Accuracy: 0.611
- AUROC: 0.683
- AUPR: 0.707
- Loss: 0.630

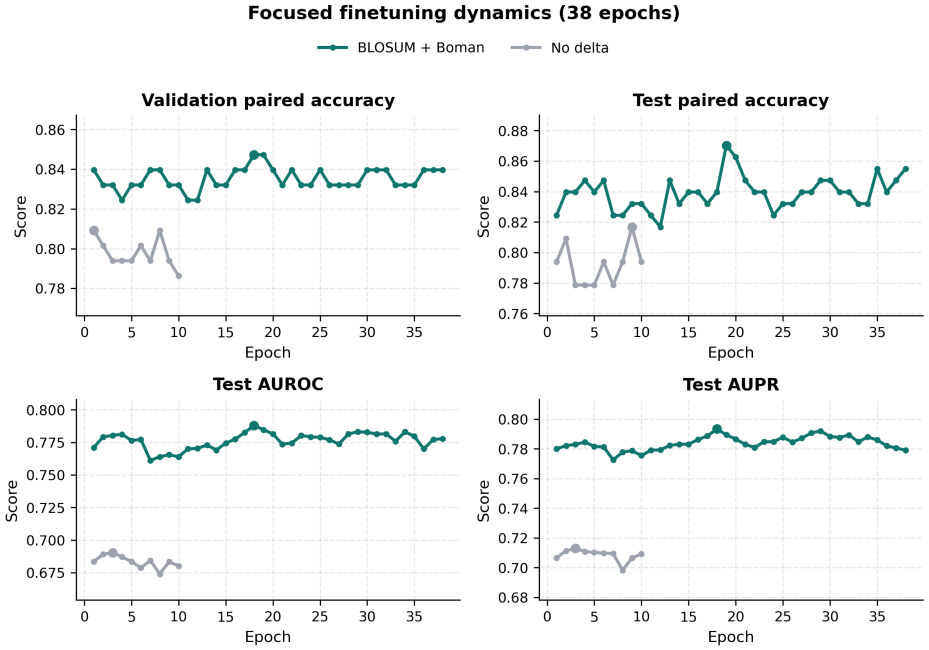

Figure 4.16. Focused fine-tuning learning curves (BLOSUM + Boman vs no-delta). Validation paired accuracy, test paired accuracy, test AUROC, and test AUPR are shown across epochs.

The consistent margin across all ranking metrics confirms that performance gains are attributable to mutation-aware feature injection rather than differences in training schedule or convergence dynamics.

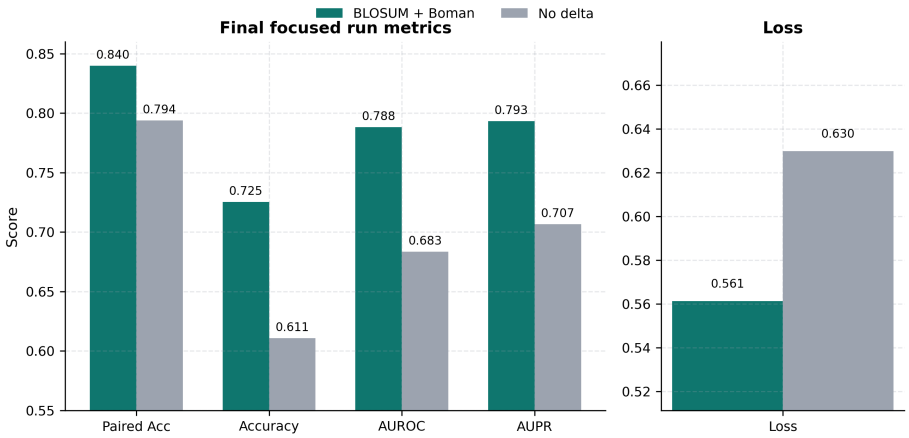

Figure 4.17. Final metric comparison for focused fine-tuning runs (BLOSUM + Boman vs no-delta). Left: higher-is-better metrics. Right: loss (lower is better).

4.10.3 Interpretation

Collectively, the fine-tuning results indicate that mutation-aware conditioning via the BLOSUM + Boman delta features improves Neo/WT discrimination relative to sequence-only adaptation under controlled transfer-learning constraints. Paired accuracy increases from 0.794 to 0.840, accompanied by consistent gains in AUROC and AUPR. The extension of improvements to threshold-independent metrics, despite optimization being guided primarily by paired accuracy, suggests coherent refinement of ranking behavior rather than metric-specific overfitting. The more gradual convergence profile observed for the BLOSUM + Boman configuration further indicates stable adaptation under highly similar Neo/WT sequence pairs.

At the same time, these improvements must be interpreted in light of the limited size and constrained diversity of the neo/WT fine-tuning dataset (1,313 paired cases; Section 3.8). In a data-limited regime with highly similar paired sequences, the introduction of hand-crafted physicochemical deltas may disproportionately influence the learned decision boundary. Although compact mutation descriptors can inject biologically meaningful signal, they also introduce potential risk of feature-induced bias or overfitting to dataset-specific mutation patterns. The observed degradation under certain feature combinations highlights the sensitivity of this regime to feature dimensionality and supervision noise.

The superior performance of the BLOSUM + Boman combination is biologically plausible, as it captures both evolutionary substitution severity and mutation-induced shifts in interaction propensity. However, alignment with immunogenicity hypotheses does not establish mechanistic causality. Rather, the results demonstrate that carefully selected mutation-aware descriptors can stabilize fine-tuning and improve mutation-conditioned ranking within the available dataset.

Importantly, this specialization operates as a statistical refinement layered onto the similarity-aware backbone established through controlled-negative training. It does not independently validate mutation-driven immunogenic mechanisms, nor does it establish broad generalization to diverse neoantigen landscapes. Instead, it represents a proof-of-concept demonstration that mutation-aware transfer can enhance paired discrimination while preserving backbone representation stability.

4.11 Case Study: Mutation-Aware TCR Prioritization for NSCLC Neoantigens

The preceding sections established a mutation-aware model with improved generalization under Grazioli-compliant evaluation and fine-tuned on Neo/WT data (Section 4.10.2). We now test this model in an independent case study: prioritization of TCR–neoantigen pairs for NSCLC. The pipeline (Section 3.9) applies the fixed model in inference-only mode to a cohort of NSCLC neoantigens that were not used for training. Below we report the quantitative outcomes; per-neoantigen results—BLOSUM distance, Boman distance, best-matching CDR3β, and model score—are tabulated in the Appendix (Table A.7, Section A.5).

The pipeline uses two distinct notions of *strong binder*, applied in sequence. First, MHC strong binders are neoantigen peptides predicted by NetMHCpan to bind MHC class I with high affinity, defined by an eluted ligand (EL) percentile rank ≤ 0.5, where EL rank reflects the predicted binding strength relative to a large set of random natural peptides. This set is a *prerequisite* for the next step.

Second, for each such peptide we rank candidate TCRs with the mutation-aware model and may classify the top pair as a TCR–peptide strong binder when the model score exceeds a chosen threshold. Thus, MHC strong binders (peptide–MHC) are the input to TCR prioritization, whereas TCR–peptide strong binders (TCR–neoantigen pairs) are the model's high-scoring output. The following sections present results for both stages.

4.11.1 Pipeline Overview and Stage Counts

Figure 4.18 provides a unified overview of the complete pipeline. Each box represents one processing stage; the number inside denotes the sample count at that stage, and arrows indicate the flow of data.

Starting from COSMIC mutation records, the pipeline resolves wild-type protein sequences for 70 unique transcripts and generates 382 primary neoantigens (full-length wild-type and mutant pairs). From these, 3,418 mutation-spanning 9-mer pairs are extracted. After deduplication, 1,739 unique 9-mers are submitted to NetMHCpan.

Filtering to strong binders (EL rank ≤ 0.5) yields 167 unique strong-binder neoantigens, each associated with one best-matching CDR3β sequence predicted by the fine-tuned model.

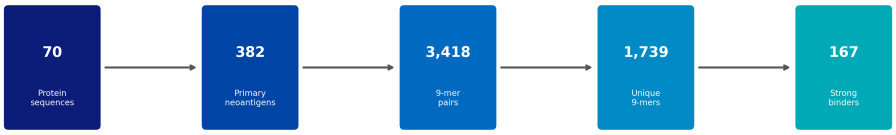

Figure 4.18. Quantitative overview of the NSCLC neoantigen prioritization pipeline. Each stage shows the number of retained entries after successive filtering and transformation steps.

4.11.2 MHC Strong Binders: Composition and EL Rank Distribution

The 167 neoantigens retained after the NetMHCpan filter are *MHC strong binders*: each has EL rank ≤ 0.5 for at least one HLA allele. Figure 4.19 shows the distribution of these peptide–allele pairs across HLA alleles. Figure 4.20 shows the EL rank distribution within this set; all values lie at or below the 0.5 threshold, with clustering in the high-affinity regime. This MHC strong-binder set is the prerequisite input for the TCR prioritization step below.

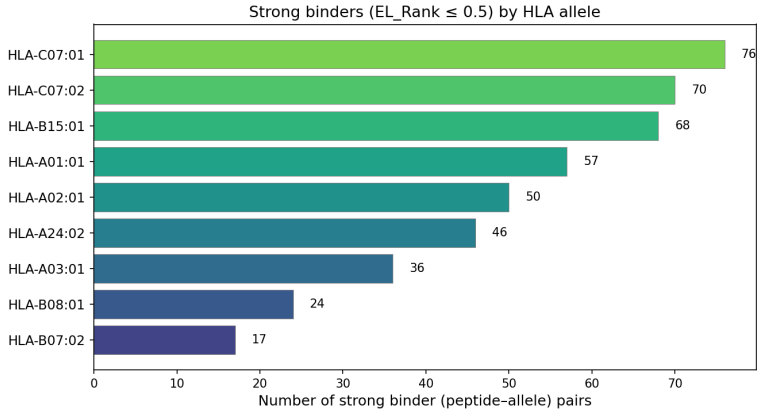

Figure 4.19. Number of strong-binder peptide–allele pairs per HLA allele (EL rank ≤ 0.5).

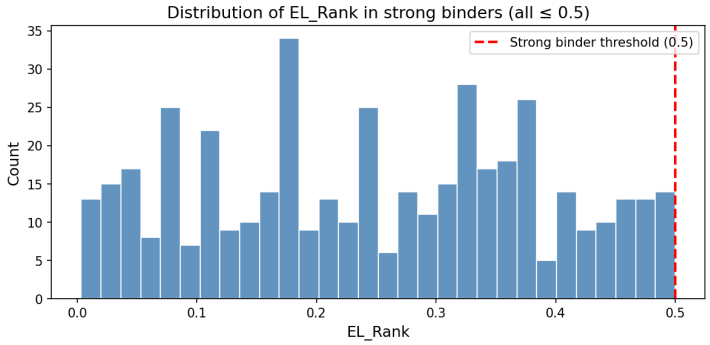

Figure 4.20. Distribution of NetMHCpan EL ranks among retained strong binders. The vertical line marks the ≤ 0.5 threshold.

4.11.3 Distribution of Best-Scoring CDR3β Interactions

For each of the 167 MHC strong binders, the mutation-aware model was used to select the highest-scoring CDR3β from the curated repertoire. Figure 4.21 shows the distribution of these best logits and binding probabilities; full scores are in the Appendix (Table A.7). Logit values span a broad dynamic range, indicating heterogeneity in predicted TCR–peptide compatibility. The probability distribution is right-skewed, suggesting that a subset of neoantigens has comparatively strong predicted TCR matches within the repertoire.

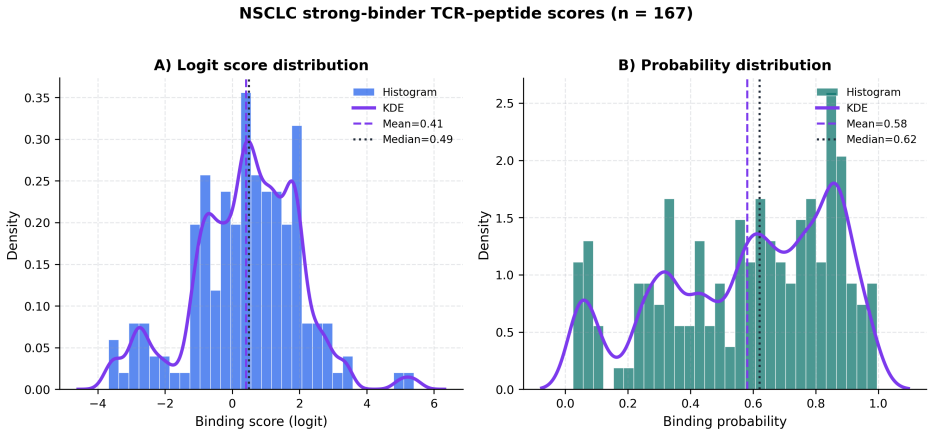

Figure 4.21. Distribution of best CDR3β binding scores for 167 NSCLC strong-binding neoantigens: (A) logit scores; (B) corresponding binding probabilities.

4.11.4 TCR–Peptide Score Thresholding and Counts

Among the 167 best-scoring TCR–neoantigen pairs (one per MHC strong binder), the proportion classified as model-predicted binders depends on the selected decision threshold. The model outputs a logit (raw score), which is converted to a probability via the sigmoid function. The standard decision boundary of the trained binary classifier is logit ≥ 0, corresponding to predicted binding probability ≥ 0.5.

Table 4.1 reports the number and fraction of pairs exceeding several logit thresholds. Under the default classifier boundary (logit ≥ 0), 107 of 167 pairs (64.1%) are classified as predicted binders. Increasing the threshold progressively restricts the set (e.g., 64 pairs at logit ≥ 1, 21 pairs at logit ≥ 2), reflecting increasing confidence under the model's scoring function.

Table 4.1. Number of TCR–neoantigen pairs exceeding selected logit (and probability) thresholds. $N = 167$ pairs.

Threshold	Count	%
Logit ≥ 0 (prob. ≥ 0.50)	107	64.1
Logit ≥ 0.5 (prob. ≥ 0.62)	86	51.5
Logit ≥ 1 (prob. ≥ 0.73)	64	38.3
Logit ≥ 1.5 (prob. ≥ 0.82)	40	24.0
Logit ≥ 2 (prob. ≥ 0.88)	21	12.6

These thresholds reflect internal classifier calibration rather than experimentally validated affinity cut-offs. The choice of threshold is therefore application-dependent: lower thresholds retain higher recall, whereas stricter cut-offs prioritize higher-confidence predictions within the model's scoring framework.

4.11.5 Interpretation

From 3,418 mutation-spanning 9-mer peptides, NetMHCpan filtering reduced the candidate set to 167 predicted strong MHC-I binders (EL rank ≤ 0.5), corresponding to approximately 4.9% of mutation-derived peptides.

Within this MHC-constrained set, the mutation-aware model assigns heterogeneous scores to candidate TCRs. The broad dynamic range of best-scoring logits indicates non-uniform and structured compatibility patterns across neoantigens rather than trivial or collapsed predictions. Under the classifier's default decision boundary (logit ≥ 0), 107 of 167 pairs are classified as predicted binders, with stricter thresholds progressively restricting this subset.

Importantly, this NSCLC deployment should be interpreted as a proof-of-concept transfer application. The model was fine-tuned on a curated Neo/WT dataset and subsequently applied in inference mode to an independent set of COSMIC-derived tumor mutations. The emergence of structured ranking behavior suggests that mutation-aware representations learned during fine-tuning transfer coherently to unseen tumor-derived peptides. However, these predictions remain statistical in nature and are not conclusive evidence of immunogenicity, physical binding affinity, or therapeutic suitability.

Because the candidate TCR repertoire partially overlaps with data sources used during backbone training, this case study does not evaluate generalization to entirely novel immune repertoires. Rather, it demonstrates feasibility of applying a mutation-aware scoring function to patient-derived mutation data under biologically constrained presentation conditions. Experimental validation using independent repertoires and functional assays would be required to assess biological relevance.

Chapter 5
Summary and Future Work

5.1 Summary of Contributions

This thesis focused on the methodological foundations of sequence-based CDR3β–peptide interaction modeling in the context of personalized cancer immunotherapy. In precision oncology, prioritizing candidate T-cell receptors for tumor-specific neoantigens remains a major challenge. The results suggest that architectural changes alone do not explain performance differences. In addition, the way negative samples are constructed during training plays an important role in generalization, a factor that has received comparatively little attention in previous studies.

In many previous works, negative samples are generated by randomly pairing TCRs with peptides that have not been experimentally reported as binders. This approach implicitly assumes that all unobserved peptide–TCR pairs represent true non-binding interactions. However, available datasets are incomplete and do not contain all biologically valid binders confirmed by functional assays.

To reduce this potential source of label noise, a controlled negative sampling framework was introduced based on peptide clustering and similarity-aware exclusion. Instead of labeling all unseen peptide–TCR pairs as non-binders, peptide similarity was explicitly considered during negative construction. Similarity was quantified using BLOSUM substitution scores, which approximate evolutionary relationships between peptides. The underlying assumption was that peptides with high evolutionary similarity may share physicochemical properties and could plausibly bind to the same TCR. Treating such near-neighbor peptides as negatives may therefore introduce ambiguous supervision.

The importance of controlling peptide similarity at evaluation time was previously emphasized by Grazioli et al., who introduced similarity-constrained dataset splits to prevent information leakage. However, existing TCR–peptide prediction models did not incorporate such similarity control during training. Negative samples were typically generated without accounting for evolutionary similarity between peptides.

In this work, similarity control was integrated directly into the negative sampling process,

thereby aligning supervision design with similarity-aware evaluation principles. This distinguishes the proposed approach from prior models, which focused primarily on architectural modifications while retaining conventional negative construction strategies.

By excluding highly similar peptides from negative assignment, the framework aimed to improve the consistency of training signals. The model was subsequently evaluated under peptide-level distribution shifts. Although training was performed using BLOSUM-based similarity control, the model also demonstrated consistent and robust behavior under RMSD-based splits in the Grazioli evaluation framework. This indicates that the supervision strategy supported generalization not only under sequence-based similarity constraints but also under structurally defined distribution shifts.

Under the hard peptide-identity split, the proposed model achieved an AUROC of approximately 0.70, compared to approximately 0.59–0.60 for NetTCR2 and ERGO2, corresponding to an absolute improvement of roughly 10–11 percentage points.

These gains cannot be attributed solely to the controlled negative sampling strategy. Architectural differences likely also contributed. In contrast to the CNN-based architecture of NetTCR-2.0 and the LSTM/autoencoder-based sequence encoders used in ERGO-II, the proposed model employs a transformer backbone, which enables contextual modeling of residue interactions. In addition, the encoding strategy was modified from BLOSUM-based or learned latent representations to AAIndex-derived physicochemical descriptors, providing structured biochemical information across eight dimensions.

The individual contribution of each component—negative sampling, architecture, and encoding—cannot be precisely disentangled. Ablation experiments were conducted on the NetTCR-2.0 baseline by modifying the encoder alone, the architecture alone, and both components simultaneously. However, even when combining transformer-based modeling with AAIndex encoding within the NetTCR-2.0 training framework, the full performance improvement observed in the final model was not reproduced. This suggests that performance gains likely arise from the interaction between supervision design, architectural capacity, and feature representation rather than from any single modification in isolation.

Given the stronger generalization of the proposed model under Grazioli's hard peptide splits, it was used as the basis for Neo/WT specialization. This evaluation setting closely resembles the neoantigen scenario, where peptides are typically unseen during training.

The model was subsequently fine-tuned on a paired Neo/WT dataset to improve discrimination between mutant peptides and their wild-type counterparts. Owing to the limited size of this dataset, additional mutation-aware features were introduced, inspired by the dissimilarity-to-self concept. These features were intended to capture physicochemical differences between mutant and wild-type sequences and to reduce potential bias arising from reliance solely on CDR3β–neoantigen sequence information.

Nevertheless, the small size of the Neo/WT dataset remains a limitation. Although the results indicate improved mutation-sensitive ranking, the findings should be interpreted cautiously due to limited statistical power and the potential risk of overfitting.

5.2 Limitations

Despite the improvements in peptide-level generalization, several limitations remain.

A central limitation of the proposed model is the absence of explicit MHC modeling.

TCR recognition occurs in the context of peptide–MHC (pMHC) complexes, and MHC binding can substantially alter the three-dimensional conformation of the presented peptide. Consequently, predicting TCR–peptide interactions without incorporating MHC context may overlook important structural determinants of recognition.

In the present framework, peptide–MHC binding affinity was not modeled directly. Instead, NetMHCpan was used as a preprocessing step to filter candidate neoantigens. Although NetMHCpan is currently one of the most established tools for MHC–peptide affinity prediction, reliance on an external predictor introduces an additional layer of approximation and potential error propagation.

Furthermore, high predicted TCR–neoantigen affinity does not guarantee therapeutic suitability. Candidate TCRs must also be evaluated for off-target cross-reactivity, stability, degradation, and potential safety risks. Even a highly specific TCR may recognize unintended wild-type peptides, which could lead to severe adverse effects. These safety considerations require extensive experimental validation and large-scale cross-reactivity datasets, which remain limited.

Another important limitation concerns data availability. True non-binding TCR–peptide pairs with experimentally confirmed negative results are scarce. As a result, negative sampling strategies rely on assumptions rather than validated non-interactions. Similarly, the Neo/WT paired dataset used for mutation-aware fine-tuning is relatively small. Single amino acid substitutions can strongly influence immunogenicity, yet the limited number of observed mutation contexts constrains statistical power and increases uncertainty in mutation-specific generalization.

5.3 Future Directions

Future work should focus on deepening the biological realism, structural integration, and clinical interpretability of the proposed framework in order to move closer to decision-support utility in personalized immunotherapy.

5.3.1 Integration of Geometric Information

TCR recognition depends on three-dimensional complementarity within the peptide–MHC complex, not only on linear sequence similarity. In this work, structural effects were approximated through RMSD-based evaluation and physicochemical encodings. However, the model remains primarily sequence-based.

A natural next step would be to incorporate explicit structural information from predicted TCR–peptide–MHC complexes, such as contact maps or structural embeddings. Graph-based or geometric deep learning methods could model spatial relationships more directly. This would better reflect the biological reality of antigen recognition and may improve generalization under structural distribution shifts, which are common in tumor heterogeneity.

5.3.2 Mechanistic Interpretability of Learned Representations

Although the model performs better under structural splits, its internal representations remain largely statistical. It is still unclear whether the model truly captures biologically meaningful interaction patterns.

Future work should examine attention maps, embedding structure, and residue-level contributions more systematically. This could help determine whether the model identifies known CDR3–peptide contact regions or relies mainly on dataset-specific correlations. Without clearer interpretability, it remains difficult to assess whether the model reflects real biological mechanisms.

5.3.3 Explicit Modeling of Cross-Reactivity

TCR cross-reactivity is a fundamental property of adaptive immunity. A single TCR can recognize multiple related peptides. This is not noise, but a natural consequence of structural and energetic constraints.

Current models treat binding as a binary outcome. However, this simplifies a complex biological reality. Future approaches could use contrastive or metric-learning methods to model cross-reactive relationships more explicitly.

Instead of predicting binding as strictly positive or negative, specificity could be represented along a similarity spectrum. This may better reflect biological recognition patterns and improve generalization across different antigen classes.

5.3.4 Integration into the Personalized Immunotherapy Pipeline

In a clinical setting, TCR–peptide prediction is only one part of a much larger workflow. It should be combined with other types of data, such as immunopeptidomics results, tumor expression levels, HLA information, and patient-specific TCR repertoire sequencing. Bringing these data sources together would place computational TCR scoring in a more realistic biological context.

At the same time, experimental validation remains essential. Computational models can help narrow down candidates, but they cannot replace functional assays. These predictions should therefore be viewed as a first filtering step rather than definitive biological confirmation.

5.4 Closing Perspective

This work explored how methodological choices influence the robustness of sequence-based CDR3β–peptide interaction modeling in a setting relevant to personalized immunotherapy. The results highlight that careful supervision design and evaluation under realistic distribution shifts are essential when developing predictive models for unseen neoantigens. At the same time, the study underscores the limits of purely sequence-based approaches in capturing the full biological complexity of TCR recognition. Important aspects such as MHC context, structural dynamics, and cross-reactivity remain open challenges. Computational modeling can meaningfully support candidate prioritization, but it should be viewed as an assisting tool within a broader experimental and clinical framework. Continued integration of biological knowledge, improved datasets, and structurally informed modeling will be necessary to move closer to reliable decision support in personalized cancer immunotherapy.

Appendix A
Additional Analyses

A.1 HPO Search Space

The following hyperparameters were optimized jointly during HPO (Chapter 4):

Optimization parameters:

- Learning rate: 10^{-6} to 10^{-1} (log-uniform)
- Weight decay: 10^{-6} to 10^{-3} (log-uniform)
- Batch size: $\{16, 32, 64, 128\}$
- Dropout: 0.1 to 0.5

Transformer architecture parameters:

- Hidden dimension: $\{64, 128, 256\}$
- Number of encoder layers: 1 to 4
- Number of attention heads: $\{4, 8, 16\}$ (constrained such that hidden dimension is divisible by head count)
- Feed-forward dimension: $\{128, 256, 512\}$

A.2 HPO Parallel-Coordinate Plots

The main Results chapter summarizes the optimization history (validation AUROC over trials) for HPO on both the controlled $t = 0.40$ dataset and the baseline `ds.csv` dataset. For completeness, this section provides the corresponding parallel-coordinate plots, which visualize how top-performing trials are distributed across the hyperparameter

space. On the controlled dataset, high-performing configurations concentrate around moderate dropout, shallow Transformer depth, and balanced or mildly imbalanced positive ratios, with multiple loss strategies appearing among the top trials. On `ds.csv`, top trials span a broader region with less consistent structure, reflecting a noisier optimization landscape and supporting the conclusion that negative sample construction strongly influences learnability.

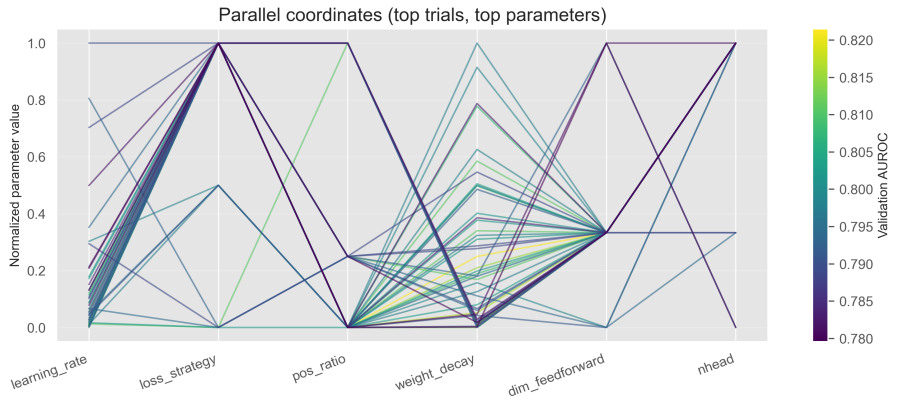

Figure A.1. Parallel-coordinate plot of hyperparameter trials on the controlled $t = 0.40$ dataset. Each line represents one trial; axes correspond to hyperparameters (e.g. dropout, depth, learning rate, class ratio). Top-performing trials cluster in a coherent region, indicating a well-posed optimization landscape.

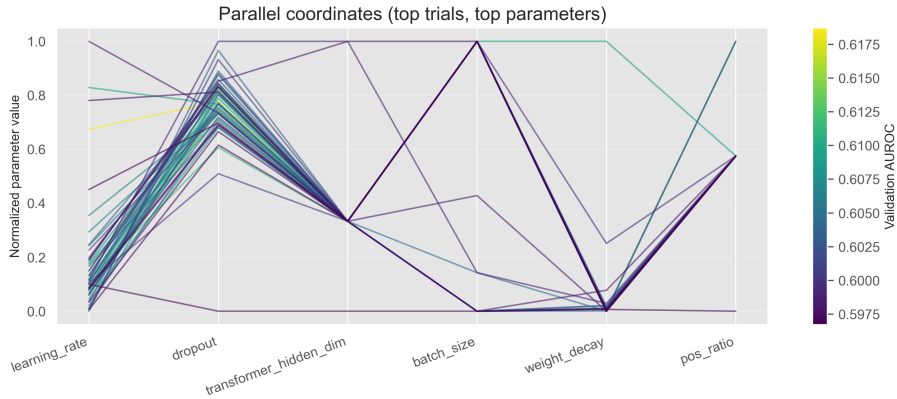

Figure A.2. Parallel-coordinate plot of hyperparameter trials on the `ds.csv` baseline dataset. Compared to the controlled dataset, top trials are more spread across the hyperparameter space, with no narrow optimal regime, consistent with a noisier supervision signal.

A.3 HPO Model Comparison: AUPR

The main Results chapter (Section 4.5.1) reports AUROC comparisons across all models for the HPO comparison under BLOSUM- and RMSD-based distance splits. For completeness, this section provides the corresponding AUPR comparisons under the same

evaluation settings. AUPR is particularly informative under class imbalance and reflects precision–recall trade-offs that can differ from AUROC-based rankings. The figures below show that the relative ordering of models and datasets (controlled $t = 0.40$ vs. `ds.csv`) under AUPR is largely consistent with the AUROC findings: the controlled-negative dataset yields stable AUPR across split types, while the heterogeneous `ds.csv` baseline tends to achieve lower AUPR under peptide-shifted evaluation.

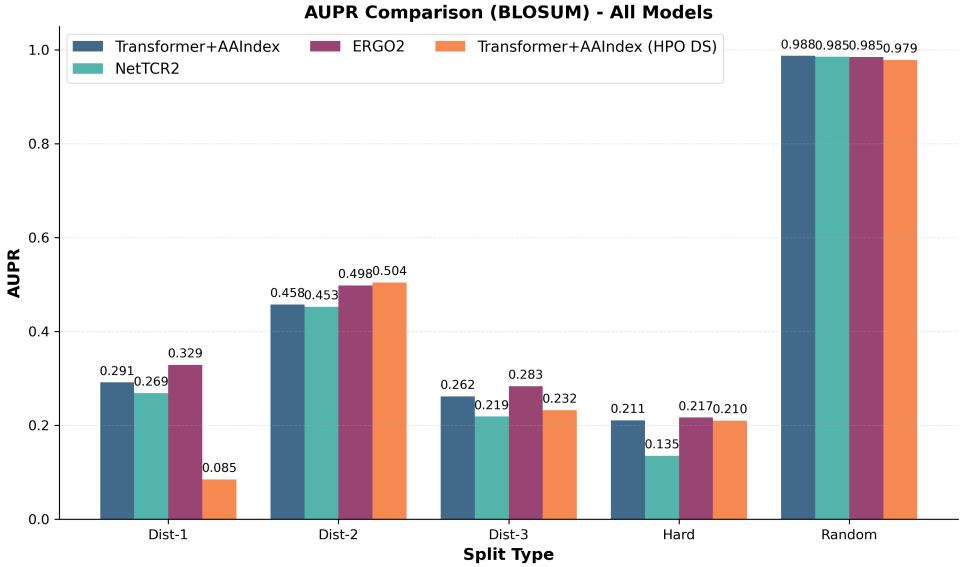

Figure A.3. AUPR comparison across all models under BLOSUM-based distance splits (HPO comparison). Complements the AUROC view in the main chapter and confirms that the controlled-negative setup maintains better precision–recall performance under peptide-shifted evaluation.

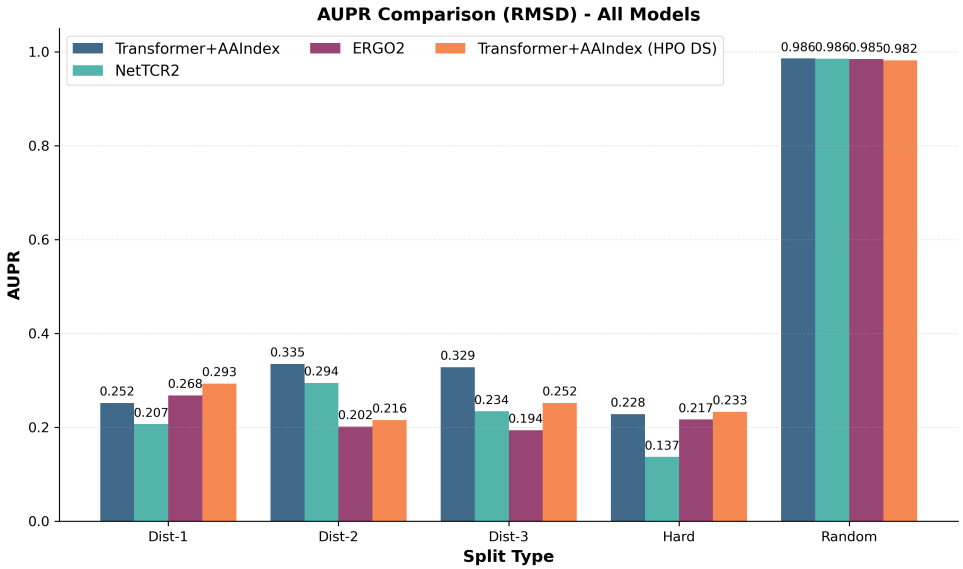

Figure A.4. AUPR comparison across all models under RMSD-based distance splits (HPO comparison). Structural (RMSD) distance splits show similar trends to the sequence-based (BLOSUM) setting, with the $t = 0.40$ controlled dataset yielding more consistent AUPR across split types.

A.4 Loss Ablation: AUROC, AUPR, and Accuracy

The main Results chapter (Section 4.6) discusses the impact of loss formulation (BCE with logits, BCE with resampling, focal mild, focal standard) on generalization under peptide-shifted evaluation. For reference, this section provides the full numerical comparison of AUROC, AUPR, and Accuracy across all loss variants and evaluation splits (BLOSUM- and RMSD-based).

A.4.1 BLOSUM-based splits

Table A.1–A.3 report AUROC, AUPR, and Accuracy for each loss variant under BLOSUM distance and hard/random splits.

Table A.1. Loss ablation: AUROC under BLOSUM-based evaluation splits.

Split	BCE logits	BCE+resamp.	Focal mild	Focal std
Distance Split 1	0.765	0.736	0.725	0.592
Distance Split 2	0.760	0.772	0.792	0.787
Distance Split 3	0.626	0.606	0.781	0.677
Hard Split	0.702	0.530	0.604	0.728
Random Split	0.983	0.984	0.982	0.982

A.4.2 RMSD-based splits

Table A.4–A.6 report the same metrics under RMSD-based evaluation splits.

Table A.2. Loss ablation: AUPR under BLOSUM-based evaluation splits.

Split	BCE logits	BCE+resamp.	Focal mild	Focal std
Distance Split 1	0.359	0.247	0.206	0.068
Distance Split 2	0.427	0.401	0.469	0.493
Distance Split 3	0.289	0.288	0.385	0.338
Hard Split	0.259	0.114	0.212	0.264
Random Split	0.986	0.987	0.986	0.986

Table A.3. Loss ablation: Accuracy under BLOSUM-based evaluation splits.

Split	BCE logits	BCE+resamp.	Focal mild	Focal std
Distance Split 1	0.959	0.941	0.920	0.890
Distance Split 2	0.933	0.910	0.929	0.907
Distance Split 3	0.786	0.790	0.760	0.787
Hard Split	0.858	0.880	0.848	0.870
Random Split	0.950	0.952	0.949	0.941

Table A.4. Loss ablation: AUROC under RMSD-based evaluation splits.

Split	BCE logits	BCE+resamp.	Focal mild	Focal std
Distance Split 1	0.532	0.539	0.488	0.453
Distance Split 2	0.757	0.774	0.706	0.654
Distance Split 3	0.720	0.729	0.744	0.752
Hard Split	0.705	0.658	0.653	0.647
Random Split	0.982	0.982	0.982	0.982

Table A.5. Loss ablation: AUPR under RMSD-based evaluation splits.

Split	BCE logits	BCE+resamp.	Focal mild	Focal std
Distance Split 1	0.267	0.276	0.237	0.173
Distance Split 2	0.377	0.276	0.205	0.192
Distance Split 3	0.490	0.384	0.451	0.467
Hard Split	0.265	0.221	0.231	0.219
Random Split	0.986	0.986	0.986	0.985

Table A.6. Loss ablation: Accuracy under RMSD-based evaluation splits.

Split	BCE logits	BCE+resamp.	Focal mild	Focal std
Distance Split 1	0.816	0.812	0.762	0.813
Distance Split 2	0.889	0.812	0.829	0.828
Distance Split 3	0.823	0.798	0.800	0.806
Hard Split	0.857	0.846	0.841	0.881
Random Split	0.951	0.950	0.948	0.936

A.5 NSCLC Strong Binders: Best CDR3β per Neoantigen

Table A.7: NSCLC strong-binder neoantigens with best-matching CDR3β and binding scores from the mutation-aware model. BLOSUM and Boman are neo–WT distances.

Peptide (neo)	WT	BLOSUM distance	Boman distance	Best CDR3β	Score
ILYEAYVMA	ILDEAYVMA	7	0.81	CSVGGTSGRGVINEQFF	-0.85
KRQGWQTFL	KRQGWRTFL	10	1.04	SSRDRAETQY	2.15
YLGNHMNIA	YLGNHMNIV	7	0.39	CSSRFRENSVNNIYF	1.12
NHMNIANLL	NHMNIVNLL	7	0.39	CASSPLRGSPSGANVLT	-1.09
TFSCHFYHF	TFSCHFYDF	5	0.45	SSRDRAETQY	3.11
YHFFNQAEW	YDFFNQAEW	5	0.45	CASSSHPGRSSGANVLT	1.82
YIIIEFMTY	YIITEFMTY	8	0.63	CSVEDLRRSSYNEQFF	1.85
RVIMHDSNY	RDIMHDSNY	15	1.16	ASSLSYDPFGLAF	0.28
VIMHDSNYV	DIMHDSNYV	15	1.16	CASSRLDRDSYEQYF	1.93
VLHESNSPY	VLHECNSPY	5	0.24	CVQGGNRVV	-3.15
RTDITVKHK	RTDITMKHK	13	0.02	CASSYVSREGPGGSLHF	0.46
TLKEGTMEV	TLKEDTMEV	5	0.97	CASGPIEQF	-3.64
YIITELMTY	YIITEFMTY	5	0.06	CFASSKGPAGGPRAQFF	-0.88
EYLEKKNVI	EYLEKKNFI	13	0.06	ASSLSGEQY	-0.38
SRMMTGDTY	SRLMTGDTY	2	0.15	CASSLVGPPGEAFF	0.32
SRLLTGDTY	SRLMTGDTY	2	0.15	CASSLWSIGGGGFIEAFF	0.38
LMAGGDLKY	LMAGGDLKS	6	0.22	CASSLWSIGGGGFIEAF	-1.79
LPRFILMEL	LPRFILLEL	2	0.15	ASSMRSSEPQH	1.35
YIITEVMTY	YIITEFMTY	13	0.06	CFASSKGPAGGPRAQFF	-1.39
HYGEVYEGV	QYGEVYEGV	8	0.10	CSVGGISGRGVINEQFF	-2.16
KLNHQNIVR	KFNHQNIVR	5	0.06	CASSSLDPTPVSEKLFF	0.79
GRVAKIADF	GRVAKIGDF	5	0.20	CASSPIGGMGTGELFF	0.27
KIADFGMAR	KIGDFGMAR	5	0.20	ASRVGPEAF	1.61
ICDFGLARY	ICDFGLARD	7	0.81	CSVGGISGRGVINEQFF	-1.08
RYIMSDSNY	RDIMSDSNY	7	0.81	CSVGGISGRGVINEQFF	-1.30
YIMSDSNYV	DIMSDSNYV	7	0.81	CASSEYPGPGANVLTF	0.45
RVIMSDSNY	RDIMSDSNY	15	1.16	CASSSHPGRSSGANVLT	-0.72
VIMSDSNYV	DIMSDSNYV	15	1.16	CASSPLRGSPSGANVLT	-0.46
FIMSDSNYV	DIMSDSNYV	12	1.22	CASSPLRGSPSGANVLT	-0.43
TQISSATEY	TQISSAMEY	6	0.45	CASSLWSIGGGGFIEAF	-3.16
SSATEYLEK	SSAMEYLEK	6	0.45	CSAPTLGLAGAPDGELF	0.04
AVKTLPEVY	AVKTLPEVC	7	0.01	CASGFIGASGRGTGELF	-0.07
YSEQDELDF	CSEQDELDF	7	0.01	CFASSKGPAGGPRAQFF	0.41
QSDFVAQGY	QSDSVAQGY	10	0.63	CDSSSHLDREETQYF	1.74
VLHECNSLY	VLHECNSPY	3	0.31	CASSSHPGRSSGANVLT	1.23
RLEAFLTPK	RLEAFLTQK	5	0.62	ASRVGPEAF	1.12
TPKAKVGEL	TQKAKVGEL	5	0.62	ASRVGPEAF	0.64
KPIIIGHHA	KPIIIGRHA	6	1.14	SSRDRAETQY	2.37
ELLAKDYRM	ELLEKDYRM	6	0.54	ASSMRSSEPQH	1.27
KLGGGQYGV	KLGGGQYGE	15	0.92	ASRVGPEAF	0.87
VVYEGVWKK	EVYEGVWKK	15	0.92	CASSLWSIGGGGFIEAF	-3.72
KLGGGQYGK	KLGGGQYGE	9	0.12	SSRDRAETQY	2.84
QYGKVYEGV	QYGEVYEGV	9	0.12	CASSLWSIGGGGFIEAF	-2.92
KVYEGVWKK	EVYEGVWKK	9	0.12	CASSLWSIGGGGFIEAF	-2.45

Continued on next page

Table A.7 – continued

Peptide (neo)	WT	BLOSUM distance	Boman distance	Best CDR3β	Score
TLKENTMEV	TLKEDTMEV	5	0.23	CFASSKGPAGGPRAQFF	-1.29
TLKEDTMAV	TLKEDTMEV	6	0.54	CASGFIGASGRGTGELF	-1.18
VADFGLSRF	VADFGLSRL	10	0.06	CASSDFRDRGHNEKLFF	2.26
SRFMTGDTY	SRLMTGDTY	10	0.06	CASSAFGGFGELFF	0.13
FMTGDTYTA	LMTGDTYTA	10	0.06	CTSSELSLCTSRANVLTF	-0.24
SAMEYLGKK	SAMEYLEKK	5	0.74	CASSQFPETGEGGANVLT	0.54
EYLGKKNFI	EYLEKKNFI	5	0.74	ASRVGPEAF	1.26
YTARAGAKF	YTAHAGAKF	8	1.14	CAIPLLPGASTGELFF	-0.97
RAGAKFPIK	HAGAKFPIK	8	1.14	ASRVGPEAF	1.95
KVNHQNIVR	KFNHQNIVR	13	0.06	CASSRKPGQGKSGELFF	0.63
LLLSWMKEV	LLLSRMKEV	10	1.89	SSRDRAETQY	4.55
WMKEVGKVF	RMKEVGKVF	10	1.89	CASGFIGASGRGTGELF	-1.05
FRSGFRQTL	FRSGSRQTL	10	0.63	CASTPIGPSGRIIGELFF	-0.05
ITFSCHFYY	ITFSCHFYD	7	0.81	CASSSHPGRSSGANVLT	-0.08
TFSCHFYYF	TFSCHFYDF	7	0.81	SSRDRAETQY	3.02
YYFFNQAEW	YDFFNQAEW	7	0.81	CASSSHPGRSSGANVLT	1.77
AWDLYYHVL	AWDLYYHVF	5	0.06	ASSLSYDPFGLAF	-0.14
LYYHVLRRI	LYYHVFRRI	5	0.06	CASSSHPGRSSGANVLT	2.37
VLRRISKQL	VFRRISKQL	5	0.06	CFASSKGPAGGPRAQFF	-1.30
FLMEALINS	FLMEALIIS	7	1.09	CASSLWSIGGGGFIEAF	-3.15
FLMEALISS	FLMEALIIS	4	0.73	CAIPLLPGASTGELFF	-3.35
FLMEALITS	FLMEALIIS	7	0.63	CASSLWSIGGGGFIEAF	-3.12
LYGLLLEML	LYDLLLEML	5	0.97	SASQEVWFPGYSNQPQH	0.67
QYDLLLEML	LYDLLLEML	6	0.93	CASSSYRDRESGANVLT	1.46
EYLEKKNII	EYLEKKNFI	9	0.10	ASSLSGEQY	-0.29
YIIVEFMTY	YIITEFMTY	13	0.47	CASSDFRDRGHNEKLFF	1.54
YIITERMTY	YIITEFMTY	10	1.91	CSSRFRENSVNNIYF	1.46
ALMSELEVL	ALMSELKVL	7	0.12	CSAPTLGLAGAPDGELF	-1.35
MSELEVLSY	MSELKVLSY	7	0.12	SAREGLAGGSYNEQF	1.06
ARDIKNDSY	ARDIKNDSN	6	0.58	CAIPLLPGASTGELFF	-0.25
YTAPAGAKF	YTAHAGAKF	7	0.52	CASSRIATSLQETQY	-0.01
APAGAKFPI	AHAGAKFPI	7	0.52	CASGFIGASGRGTGELF	-0.42
RLEACLTQK	RLEAFLTQK	8	0.39	CASSPLGGKNTGELFF	0.68
NSPYIVDFY	NSPYIVGFY	6	0.97	CAISEFKTGGFGTEAFF	1.00
IVDFYGAFY	IVGFYGAFY	6	0.97	CDSSSHLDREETQYF	1.86
HRKTRHVNI	LRKTRHVNI	8	0.83	CATSDHRDSGNTIYF	-0.40
VFGFVLITF	AFGFVLITF	14	0.39	SARVDRDSYEQYV	1.70
VLITFSYHF	VLITFSCHF	7	0.01	SSRDRAETQY	3.20
TFSYHFYDF	TFSCHFYDF	7	0.01	SARDSQDDFGTDTQY	2.02
HIMSDSNYV	DIMSDSNYV	5	0.45	CASSPLRGSPSGANVLT	0.69
ARELHQFAF	ARELHQFTF	4	0.08	CASSSHPGRSSGANVLT	0.38
ARELHQLTF	ARELHQFTF	5	0.06	CSVEDLRRSSYNEQF	1.77
ARELHQFSF	ARELHQFTF	3	0.09	CSGRDRAQSSYEQYF	1.88
QLMPFGSLL	QLMPFGCLL	5	0.24	SSRDRAETQY	4.98
ARNMYDKEY	ARDMYDKEY	5	0.23	CDSSSHLDREETQYF	1.21
GLARVMYDK	GLARDMYDK	15	1.16	CASSLWSIGGGGFIEAF	-0.06
ARVMYDKEY	ARDMYDKEY	15	1.16	ASSLNHEQF	0.55
RVMYDKEYY	RDMYDKEYY	15	1.16	CASSKKLKRGETQYF	1.87
VVGAAGVGK	VVGAGGVGK	5	0.20	ASSFGSEAF	0.77
VVGAGRVGK	VVGAGGVGK	8	1.66	CASSPIGGMGTGELFF	0.02
AGRVGKSAL	AGGVGKSAL	8	1.66	CASSLWSIGGGGFIEAF	-2.29
EYMANGSLL	EYMANGCLL	5	0.24	ASSLSYDPFGLAF	0.26

Continued on next page

Table A.7 – continued

Peptide (neo)	WT	BLOSUM distance	Boman distance	Best CDR3β	Score
MANGSLLNY	MANGCLLNY	5	0.24	CASSPLRGSPSGANVLT	0.21
SLLNYLREM	CLLNYLREM	5	0.24	CASSQIETRATNEKLF	0.73
EYMANGRLL	EYMANGCLL	9	1.52	ASSPDRGYEQY	0.35
MANGRLLNY	MANGCLLNY	9	1.52	CSSRFRENSVNNIYF	1.07
RLLNYLREM	CLLNYLREM	9	1.52	CASSQIETRATNEKLF	0.72
ITEYMANGY	ITEYMANGC	7	0.01	CFASSKGPAGGPRAQFF	1.13
EYMANGYLL	EYMANGCLL	7	0.01	ASSVDTGENTEAF	0.54
MANGYLLNY	MANGCLLNY	7	0.01	SYQESSYGYT	2.61
YLLNYLREM	CLLNYLREM	7	0.01	SASQEVWFPGYSNQPQH	2.49
EYMANGFLL	EYMANGCLL	12	0.39	CSSRHLSGSGETQYF	0.56
MANGFLLNY	MANGCLLNY	12	0.39	SARVDRDSYEQYV	2.55
FLLNYLREM	CLLNYLREM	12	0.39	CSARDLDRDGTDTQYF	1.93
HYDLLLEML	LYDLLLEML	8	0.83	CASSRRRYPQSRANVLT	1.85
STVQLITQF	STVQLITQL	10	0.06	CASSSHPGRSSGANVLT	1.45
QLITQFMPF	QLITQLMPF	10	0.06	CASSSYRDRESGANVLT	2.09
QLITQHMPF	QLITQLMPF	8	0.83	CASSSYRDRESGANVLT	1.14
ARDMHDKEY	ARDMYDKEY	7	0.36	CASSPWPGTGDTEAFF	0.81
SPLPERAHL	SPLPERAHP	3	0.31	ASRVGPEAF	1.04
LPERAHLEV	LPERAHPEV	3	0.31	CVQGGNRVV	-4.15
VVGARGVGK	VVGAGGVGK	8	1.66	ASRVGPEAF	1.41
VVGAVGVGK	VVGAGGVGK	13	0.19	ASRVGPEAF	1.27
VVGASGVGK	VVGAGGVGK	4	0.38	CASSLFPLAGGPIGELFF	0.18
FRSGSQQTL	FRSGSRQTL	10	1.04	CASSLWSIGGGGFIEAF	-2.50
HRVFVNQSL	HRVFVNRSL	10	1.04	CSSRHLSGSGETQYF	0.73
QRMPFGCLL	QLMPFGCLL	10	1.97	ASRVGPEAF	0.44
QLITQPMPF	QLITQLMPF	4	0.31	CSSRFRENSVNNIYF	1.49
QPMPFGCLL	QLMPFGCLL	4	0.31	SSRDRAETQY	2.47
QLITQVMPF	QLITQLMPF	14	0.12	CASSSHPGRSSGANVLT	1.39
ASGAFGTVY	GSGAFGTVY	5	0.20	ASRVGPEAF	1.19
TQNQKVGEL	TQKQKVGEL	6	0.12	CFASSKGPAGGPRAQFF	-2.25
ILYIGDDIF	ILYIGDHIF	6	0.45	CASSSHPGRSSGANVLT	-1.20
KHKNIIKLL	KHKNIINLL	9	0.12	CSAPTLGLAGAPDGELF	-1.19
QSNSVAQGY	QSDSVAQGY	5	0.23	CSAPTLGLAGAPDGELF	0.03
ESDILAQGF	QSDILAQGF	9	0.12	CDSSSHLDREETQYF	1.83
QSDMLAQGF	QSDILAQGF	8	0.18	SARVDRDSYEQYV	1.68
LAELYKHLY	LAELYKHLD	7	0.81	CDSSSHLDREETQYF	1.02
HLYSSSNER	HLDSSSNER	7	0.81	CASSLWSIGGGGFIEAF	-3.63
RTFLVIPEF	RTFLVIPEL	10	0.06	SASQEVWFPGYSNQPQH	2.87
VIPEFAQEL	VIPELAQEL	10	0.06	ASSLDRDLSSYNEQF	0.30
EFAQELHVW	ELAQELHVW	10	0.06	CAISEFKTGGFGTEAFF	0.57
YIINEFMTY	YIITEFMTY	6	0.45	CASSSHPGRSSGANVLT	0.63
SSAEIHQAF	SFAEIHQAF	5	0.63	CASSSHPGRSSGANVLT	-0.10
ESLAYNKFY	ESLAYNKFS	6	0.22	SSTGGGRYGYT	2.38
SLAYNKFYI	SLAYNKFSI	6	0.22	SARVDRDSYEQYV	2.02
LAYNKFYIK	LAYNKFSIK	6	0.22	CASSLFPLAGGPIGELFF	-0.22
KFYIKSDVW	KFSIKSDVW	6	0.22	CASSQLDSFPEPGELFF	0.53
YIKSDVWAF	SIKSDVWAF	6	0.22	CASSQFPETGEGGANVLT	-0.08
GLARDIKNY	GLARDIKND	7	0.81	CASGFIGASGRGTGELF	-0.74
RQTLFASQM	RQTLFASQV	6	0.02	SARDSQDDFGTDTQY	2.17
TLFASQMMR	TLFASQVMR	6	0.02	CASSLWSIGGGGFIEAF	-1.12
SQMMRYADL	SQVMRYADL	6	0.02	CDSSSHLDREETQYF	2.58
VWTNKSSLF	VWTDKSSLF	5	0.23	CASSEVPFF	-0.51

Continued on next page

Table A.7 – continued

Peptide (neo)	WT	BLOSUM distance	Boman distance	Best CDR3β	Score
RLDIDETEY	LLDIDETEY	10	1.97	CASSSHPGRSSGANVLT	-0.43
KHKNIIHLL	KHKNIINLL	4	0.22	CASGFIGASGRGTGELF	-1.15
KPITIGVHA	KPITIGRHA	15	1.85	CSAPTLGLAGAPDGELF	-0.89
GVHAHGDQY	GRHAHGDQY	15	1.85	SARVDRDSYEQYV	1.63
SPNGTIQNI	SPNGTIRNI	10	1.04	SARVDRDSYEQYV	1.20
LLLSGMKEV	LLLSRMKEV	6	1.66	CASSLWSIGGGGFIEAF	-3.21
GMKEVGKVF	RMKEVGKVF	6	1.66	CSAPTLGLAGAPDGELF	-0.58
VLHECNSSY	VLHECNSPY	5	0.38	CSVGGISGRGVINEQFF	0.84
NSSYIVGFY	NSPYIVGFY	5	0.38	CASSPLRGSPSGANVLT	1.22
NSPYIVVFY	NSPYIVGFY	13	0.19	CAIPGRPGGVFGANVLT	0.99
TQKQKEGEL	TQKQKVGEL	10	0.92	CVQGGNRVV	-2.84
NLMDLQKKL	NLVDLQKKL	6	0.02	CASSSYRDRESGANVLT	1.83
DSPYIVGFY	NSPYIVGFY	6	0.23	CFASSKGPAGGPRAQFF	0.44
LLLSLMKEV	LLLSRMKEV	7	1.97	CSVEDRGPPGSGANVLT	-1.15
SLMKEVGKV	SRMKEVGKV	7	1.97	CASSPLRGSPSGANVLT	0.34
LMKEVGKVF	RMKEVGKVF	7	1.97	CASGFIGASGRGTGELF	-0.78
LVDAWKPLF	LVDARKPLF	10	1.89	CSSIRSSDTQ	0.22
NFALVEHLL	NFALVGHLL	6	0.74	ASSPPYSGANVLT	-0.88
LVEHLLKGY	LVGHLLKGY	6	0.74	CASSSHPGRSSGANVLT	0.91

Bibliography

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. arXiv preprint. (2019). https://arxiv.org/abs/1907.10902 arXiv:1907.10902. (cited on page 2.5.3)

[2] Ehsan Asgari and Mohammad R. K. Mofrad. 2015. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLOS ONE* 10, 11 (2015), e0141287. DOI:http://dx.doi.org/10.1371/journal.pone.0141287 (cited on pages 2.2.1 and 3.4.1)

[3] W. R. Atchley, J. Zhao, A. D. Fernandes, and T. Drüke. 2005. Solving the protein sequence metric problem. *PNAS* 102, 18 (2005), 6395–6400. DOI:http://dx.doi.org/10.1073/pnas.0408677102 (cited on pages 2.4, 2.2, 2.2.1 and 3.4.1)

[4] Amalie K. Bentzen and Sine Reker Hadrup. 2017. Evolution of MHC-based technologies used for detection of antigen-responsive T cells. *Cancer Immunology, Immunotherapy* 66 (2017), 657–666. DOI:http://dx.doi.org/10.1007/s00262-017-1971-2 (cited on page 1.4)

[5] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems*, Vol. 24. 2546–2554. (cited on page 2.5.3)

[6] Michael E. Birnbaum, Juan L. Mendoza, Dhruv K. Sethi, Shen Dong, Jacob Glanville, Jessica Dobbins, Engin Özkan, Mark M. Davis, Kai W. Wucherpfennig, and K. Christopher Garcia. 2014. Deconstructing the Peptide–MHC Specificity of T Cell Recognition. *Cell* 157, 5 (2014), 1073–1087. DOI:http://dx.doi.org/10.1016/j.cell.2014.03.047 (cited on pages 2.1 and 2.2.1)

[7] H. G. Boman. 2003. Antibacterial peptides: basic facts and emerging concepts. *Journal of Internal Medicine* 254, 3 (2003), 197–215. DOI:http://dx.doi.org/10.1046/j.1365-2796.2003.01228.x (cited on page 2.1.5)

[8] J. J. A. Calis, M. Maybeno, J. A. Greenbaum, D. Weiskopf, A. D. De Silva, A. Sette, C. Keşmir, and B. Peters. 2013. Properties of MHC class I presented peptides that enhance immunogenicity. *PLoS Computational Biology* 9, 10 (2013), e1003266. DOI:http://dx.doi.org/10.1371/journal.pcbi.1003266 (cited on page 2.1.4)

[9] B. J. Cameron and others. 2013. Identification of a T cell receptor that recognizes a peptide derived from the wild-type p53 protein. *Science* 342, 6164 (2013), 1501–1504. DOI:http://dx.doi.org/10.1126/science.1240987 (cited on page 1.3.2)

[10] Leonardo V. Castorina, Filippo Grazioli, Pierre Machart, Anja Mösch, and Federico Errica. 2025. Assessing the Generalization Capabilities of TCR Binding Predictors via Peptide Distance Analysis. *PLOS ONE* 20, 5 (2025), e0324011. DOI:http://dx.doi.org/10.1371/journal.pone.0324011 (cited on pages 1.3.3, 2.7.2, 2.7 and 2.8)

[11] Pierre G. Coulie, Benoît J. Van den Eynde, Pierre van der Bruggen, and Thierry Boon. 2014. Tumour antigens recognized by T lymphocytes: at the core of cancer immunotherapy. *Nature Reviews Cancer* 14, 2 (2014), 135–146. DOI:http://dx.doi.org/10.1038/nrc3670 (cited on page 1.1)

[12] J. M. Custodio, C. M. Ayres, T. J. Rosales, and B. M. Baker. 2023. Structural and physical features that distinguish tumor-controlling from inactive cancer neoepitopes. *PNAS* 120, 51 (2023), e2312057120. DOI:http://dx.doi.org/10.1073/pnas.2312057120 (cited on pages 1.3.2 and 1.2)

[13] P. Dash and others. 2017. Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature* 547 (2017), 89–93. DOI:http://dx.doi.org/10.1038/nature22383 (cited on pages 1.3.3 and 2.6)

[14] C. Dens and others. 2023. The pitfalls of negative data bias for the T-cell epitope specificity challenge. *Nature Machine Intelligence* 5 (2023), 1060–1062. DOI:http://dx.doi.org/10.1038/s42256-023-00727-0 (cited on page 1.3.3)

[15] J. R. Devlin, J. A. Alonso, C. M. Ayres, G. L. J. Keller, S. Bobisse, C. W. Vander Kooi, G. Coukos, D. Gfeller, A. Harari, and B. M. Baker. 2020. Structural dissimilarity from self drives neoepitope escape from immune tolerance. *Nature Chemical Biology* 16, 11 (2020), 1269–1276. DOI:http://dx.doi.org/10.1038/s41589-020-0610-1 (cited on pages 2.1.5 and 2.2.1)

[16] Quentin M. Eastman, Tzu-Ming Leu, and David G. Schatz. 1996. Initiation of V(D)J recombination in vitro obeying the 12/23 rule. *Nature* 380, 6569 (March 1996), 85–88. DOI:http://dx.doi.org/10.1038/380085a0 (cited on page 1.2)

[17] A. D. Fesnak, C. H. June, and B. L. Levine. 2016. Engineered T cells: the promise and challenges of cancer immunotherapy. *Nature Reviews Cancer* 16 (2016), 566–581. DOI:http://dx.doi.org/10.1038/nrc.2016.97 (cited on page 1.1)

[18] K. Christopher Garcia, Marina Degano, Larry R. Pease, Mingdong Huang, Per A. Peterson, Luc Teyton, and Ian A. Wilson. 1996. Structural basis of plasticity in T cell receptor recognition of a self peptide–MHC antigen. *Science* 279, 5354 (1996), 1166–1172. DOI:http://dx.doi.org/10.1126/science.279.5354.1166 (cited on page 2.1)

[19] K. C. Garcia, L. Teyton, and I. A. Wilson. 1999. Structural basis of T cell recognition. *Annual Review of Immunology* 17 (1999), 369–397. DOI:http://dx.doi.org/10.1146/annurev.immunol.17.1.369 (cited on page 2.1)

[20] Filippo Grazioli, Anja Mösch, Pierre Machart, Kun Li, Ismail Alqassem, Timothy J. O'Donnell, and Martin R. Min. 2022a. On TCR Binding Predictors Failing to Generalize to Unseen Peptides. *Frontiers in Immunology* 13 (2022), 1014256. DOI:http://dx.doi.org/10.3389/fimmu.2022.1014256 (cited on pages 1.3.1, 1.3.3, 2.3, 2.5.1, 2.6, 2.6.1, 2.7.1, 2.6 and 3.3.6)

[21] Filippo Grazioli, Anja Mösch, Pierre Machart, Kun Li, Ismail Alqassem, Timothy J. O'Donnell, and Martin R. Min. 2022b. TChard: TCR-peptide/-pMHC binding dataset. (2022). DOI:http://dx.doi.org/10.5281/zenodo.6962043 (cited on pages 2.5.1 and 3.3.6)

[22] Fei Guo, Renchu Guan, Yaohang Li, Qi Liu, Xiaowo Wang, Can Yang, and Jianxin Wang. 2025. Foundation Models in Bioinformatics. *National Science Review* 12, 4 (April 2025), nwaf028. DOI:http://dx.doi.org/10.1093/nsr/nwaf028 Published: 25 January 2025. (cited on page 2.4.1)

[23] Avid Harding-Larsen, Jonathan Funk, Niklas Gesmar Madsen, Hani Gharabli, Carlos G. Acevedo-Rocha, Stanislav Mazurenko, and Ditte Hededam Welner. 2024. Protein representations: Encoding biological information for machine learning in biocatalysis. *Biotechnology Advances* 77 (2024), 108459. DOI:http://dx.doi.org/10.1016/j.biotechadv.2024.108459 (cited on page 2.3)

[24] Steven Henikoff and Jorja G. Henikoff. 1992. Amino acid substitution matrices from protein blocks. *Proc. Nat. Acad. Sci. U. S. A.* 89, 22 (November 1992), 10915–10919. DOI:http://dx.doi.org/10.1073/pnas.89.22.10915 (cited on pages 2.1.5, 2.2.1, 2.1 and 3.4.1)

[25] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. (cited on page 3.4.2)

[26] J. R. Hopkins, B. J. MacLachlan, S. Harper, A. K. Sewell, and D. K. Cole. 2022. Unconventional modes of peptide-HLA-I presentation change the rules of TCR engagement. *Discovery Immunology* 1, 1 (2022), kyac001. DOI:http://dx.doi.org/10.1093/discim/kyac001 (cited on page 2.1.4)

[27] A. Ikai. 1980. Thermostability and aliphatic index of globular proteins. *Journal of Biochemistry* 88, 6 (1980), 1895–1898. DOI:http://dx.doi.org/10.1093/oxfordjournals.jbchem.a133168 (cited on page 2.1.5)

[28] Carl H. June, Stanley R. Riddell, and Ton N. Schumacher. 2015a. Adoptive cellular therapy: a race to the finish line. *Science Translational Medicine* 7, 280 (2015), 280ps7. DOI:http://dx.doi.org/10.1126/scitranslmed.aaa3643 (cited on page 1.1)

[29] Carl H. June, Stanley R. Riddell, and Ton N. Schumacher. 2015b. Adoptive cellular therapy: a race to the finish line. *Science Translational Medicine* 7, 280 (2015), 280ps7. DOI:http://dx.doi.org/10.1126/scitranslmed.aaa3643 (cited on pages 1.3.2 and 1.4)

[30] V. Jurtz and others. 2018. NetTCR: sequence-based prediction of TCR binding to peptide–MHC complexes. *Bioinformatics* 34, 23 (2018), 4129–4137. DOI:http://dx.doi.org/10.1093/bioinformatics/bty500 (cited on pages 1.3.3, 2.3, 2.5.1, 2.6 and 2.6.2)

[31] S. Kawashima and others. 2008. AAindex: amino acid index database. *Nucleic Acids Research* 36 (2008), D202–D205. DOI:http://dx.doi.org/10.1093/nar/gkm998 (cited on pages 2.2.1 and 3.4.1)

[32] A. Kidera, Y. Konishi, M. Oka, T. Ooi, and H. A. Scheraga. 1985. Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *Journal of Protein Chemistry* 4 (1985), 23–55. DOI:http://dx.doi.org/10.1007/BF01025492 (cited on page 2.2.1)

[33] Zeynep Kosaloğlu-Yalçın, Nina Blazeska, Randi Vita, Hannah Carter, Morten Nielsen, Stephen Schoenberger, Alessandro Sette, and Bjoern Peters. 2023. The Cancer Epitope Database and Analysis Resource (CEDAR). *Nucleic Acids Research* 51, D1 (2023), D845–D852. DOI:http://dx.doi.org/10.1093/nar/gkac902 (cited on pages 3.8 and 3.8.1)

[34] Jack Kyte and Russell F. Doolittle. 1982. A Simple Method for Displaying the Hydropathic Character of a Protein. *Journal of Molecular Biology* 157, 1 (1982), 105–132. DOI:http://dx.doi.org/10.1016/0022-2836(82)90515-0 (cited on pages 2.2.1 and 3.4.1)

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. (cited on page 3.4.2)

[36] Marie-Paule Lefranc and others. 2005. IMGT unique numbering for immunoglobulin and T cell receptor variable domains and Ig superfamily V-like domains. *Developmental & Comparative Immunology* 29, 3 (2005), 185–203. DOI:http://dx.doi.org/10.1016/j.dci.2004.07.003 (cited on page 2.5.4)

[37] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, and others. 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 379, 6637 (2023), 1123–1130. DOI:http://dx.doi.org/10.1126/science.ade2574 (cited on page 2.5.2)

[38] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*. (cited on page 3.8.7)

[39] T. Lu and others. 2021. Deep learning-based prediction of the T cell receptor–antigen specificity using transfer learning (pMTnet). *Nature Machine Intelligence* (2021). pMHC-TCR binding prediction network demonstrates transfer learning application. (cited on page 2.4.1)

[40] T. Lu, Z. Zhang, J. Zhu, and others. 2021. Deep learning-based prediction of the T cell receptor–antigen binding specificity. *Nature Machine Intelligence* 3, 10 (2021), 864–875. DOI:http://dx.doi.org/10.1038/s42256-021-00383-2 (cited on page 1.3.3)

[41] Yang Lu, Yifan Wang, Min Xu, Bo Xie, Yu Yang, Hong Xu, and Shuang Suo. 2025. Assessment of Computational Methods in Predicting TCR–Epitope Binding Recognition. Nature Methods. (2025). DOI:http://dx.doi.org/10.1038/s41592-025-02910-0 Epub ahead of print. (cited on pages 1.1 and 1.2)

[42] Vincent Lyot and others. 2024. NeoTCR Database. https://github.com/lyotvincent/NeoTCR. (2024). Accessed: 2025-01-27. (cited on pages 2.5.1, 3.8 and 3.8.1)

[43] R. A. Mariuzza, P. Agnihotri, and J. Orban. 2020. The structural basis of T-cell receptor (TCR) activation: An enduring enigma. *Journal of Biological Chemistry* 295, 4 (2020), 914–925. DOI:http://dx.doi.org/10.1074/jbc.REV119.009411 (cited on pages 2.1 and 2.1.1)

[44] Anna Montemurro, Viktor Schuster, Helle R. Povlsen, Amalie K. Bentzen, Vanessa Jurtz, William D. Chronister, Amanda Crinklaw, Sine R. Hadrup, Ole Winther, Bjoern Peters, Lea E. Jessen, and Morten Nielsen. 2021. NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. *Communications Biology* 4, 1 (2021), 1060. DOI:http://dx.doi.org/10.1038/s42003-021-02610-3 (cited on pages 2.5.1, 2.6, 2.6.2, 2.5 and 3.7)

[45] Richard A. Morgan, Natarajan Chinnasamy, Daniel Abate-Daga, Alexis Gros, Paul F. Robbins, Zhiya Zheng, Mark E. Dudley, Steven A. Feldman, James C. Yang, Robert M. Sherry, Gia Q. Phan, Mark S. Hughes, Udai S. Kammula, Andrew D. Miller, Craig J. Hessman, A. Andrew Stewart, Nicholas P. Restifo, Maritza M. Quezado, Mahesh Alimchandani, Adam Z. Rosenberg, Avindra Nath, Tianxia Wang, Bibiana Bielekova, Susanne C. Wuest, Naveen Akula, Francis J. McMahon, Stefanie Wilde, Bernhard Mosetter, Dolores J. Schendel, Christine M. Laurencot, and Steven A. Rosenberg. 2013. Cancer regression and neurological toxicity following anti-MAGE-A3 TCR gene therapy. *Journal of Immunotherapy* 36, 2 (2013), 133–151. DOI:http://dx.doi.org/10.1097/CJI.0b013e3182829903 (cited on pages 1.1 and 1.3.2)

[46] Patrick A. Ott, Zhihao Hu, Dilsah B. Keskin, and others. 2017. An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature* 547 (2017), 217–221. DOI:http://dx.doi.org/10.1038/nature22991 (cited on page 1.4)

[47] K. C. Parker, M. A. Bednarek, L. K. Hull, U. Utz, B. Cunningham, H. J. Zweerink, W. E. Biddison, and J. E. Coligan. 1992. Sequence motifs important for peptide binding to the human MHC class I molecule, HLA-A2. *Journal of Immunology* 149, 11 (1992), 3580–3587. (cited on page 2.1.4)

[48] Roshan Rao, Julian Meier, Corinna Völz, Christopher Bauer, Tom Sercu, and Alexander Rives. 2021. MSA Transformer. *bioRxiv* (2021). DOI:http://dx.doi.org/10.1101/2021.02.12.430858 (cited on page 2.4)

[49] B. Reynisson, B. Alvarez, S. Paul, B. Peters, and M. Nielsen. 2020. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation. *Nucleic Acids Research* 48, W1 (2020), W449–W454. DOI:http://dx.doi.org/10.1093/nar/gkaa379 (cited on pages 1.1, 1.3.4, 1.4 and 2.5.2)

[50] T. P. Riley, G. L. J. Keller, A. R. Smith, and B. M. Baker. 2019. Structure Based Prediction of Neoantigen Immunogenicity. *Frontiers in Immunology* 10 (2019), 2047. DOI:http://dx.doi.org/10.3389/fimmu.2019.02047 (cited on page 1.3.2)

[51] Alexander Rives, Julian Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jure Liu, David Guo, Markus Ott, C Lawrence Zitnick, Jerry Ma, and Rob Fergus. 2021. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* 118, 15 (2021), e2016239118. DOI:http://dx.doi.org/10.1073/pnas.2016239118 (cited on page 2.4)

[52] James Robinson, Jason A. Halliwell, James D. Hayhurst, Paul Flicek, Peter Parham, and Steven G. Marsh. 2015. The IPD and IMGT/HLA database: allele variant databases. *Nucleic Acids Research* 43, Database issue (2015), D423–D431. DOI:http://dx.doi.org/10.1093/nar/gku1161 Epub 2014 Nov 20. (cited on page 2.5.4)

[53] Maarten C. J. Roex, Laura Hageman, Stephanie A. J. Veld, Esther van Egmond, Carlijn Hoogstraten, Christian Stemberger, Lars Germeroth, Hermann Einsele, J. H. Frederik Falkenburg, and Inge Jedema. 2020. A minority of T cells recognizing tumor-associated antigens presented in self-HLA can provoke antitumor reactivity. *Blood* 136, 4 (2020), 455–467. DOI:http://dx.doi.org/10.1182/blood.2019004443 (cited on page 1.1)

[54] S. A. Rosenberg and N. P. Restifo. 2015. Adoptive cell transfer as personalized immunotherapy for human cancer. *Science* 348, 6230 (2015), 62–68. DOI:http://dx.doi.org/10.1126/science.aaa4967 (cited on page 1.1)

[55] P. Sharma and J. P. Allison. 2015. The future of immune checkpoint therapy. *Science* 348, 6230 (2015), 56–61. DOI:http://dx.doi.org/10.1126/science.aaa8172 (cited on page 1.1)

[56] Mikhail Shugay, Dmitriy V. Bagaev, Ivan V. Zvyagin, Renske M. Vroomans, Jeremy C. Crawford, Garry Dolton, Ekaterina A. Komech, Anna L. Sycheva, Anna E. Koneva, Evgeniy S. Egorov, and others. 2018. VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Research* 46, D1 (2018), D419–D427. DOI:http://dx.doi.org/10.1093/nar/gkx760 (cited on page 2.5.1)

[57] I. Springer and others. 2020. Prediction of specific TCR–peptide binding from large dictionaries of TCR–peptide pairs. *Frontiers in Immunology* 11 (2020), 1803. DOI:http://dx.doi.org/10.3389/fimmu.2020.01803 (cited on pages 1.3.3, 2.3, 2.6 and 2.6.1)

[58] Ido Springer, Nili Tickotsky, and Yoram Louzoun. 2021. Contribution of T Cell Receptor Alpha and Beta CDR3, MHC Typing, V and J Genes to Peptide Binding Prediction. *Frontiers in Immunology* 12 (2021), 664514. DOI:http://dx.doi.org/10.3389/fimmu.2021.664514 (cited on pages 2.1.2, 2.1.3, 2.6 and 2.6.1)

[59] C. Szeto, C. A. Lobos, A. T. Nguyen, and S. Gras. 2020. TCR Recognition of Peptide–MHC-I: Rule Makers and Breakers. *International Journal of Molecular Sciences* 22, 1 (2020), 68. DOI:http://dx.doi.org/10.3390/ijms22010068 (cited on page 2.1)

[60] John G. Tate, Sally Bamford, Harry C. Jubb, Zbyslaw Sondka, David M. Beare, Nidhi Bindal, Harry Boutselakis, Charlotte G. Cole, Celestino Creatore, Elisabeth Dawson, and others. 2019. COSMIC: the Catalogue Of Somatic Mutations In Cancer. *Nucleic Acids Research* 47, D1 (2019), D941–D947. DOI:http://dx.doi.org/10.1093/nar/gky1015 (cited on pages 2.5.1 and 3.9.1)

[61] N. Tickotsky, T. Sagiv, J. Prilusky, E. Shifrut, and N. Friedman. 2017. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics* 33, 18 (2017), 2924–2929. DOI:http://dx.doi.org/10.1093/bioinformatics/btx286 (cited on pages 2.5.1, 3.8 and 3.8.1)

[62] E. Tran, P. F. Robbins, Y. C. Lu, T. D. Prickett, J. J. Gartner, L. Jia, A. Pasetto, Z. Zheng, S. Ray, E. M. Groh, I. R. Kriley, and S. A. Rosenberg. 2016. T-Cell Transfer Therapy Targeting Mutant KRAS in Cancer. *New England Journal of Medicine* 375, 23 (2016), 2255–2262. DOI:http://dx.doi.org/10.1056/NEJMoa1609279 (cited on pages 1.1 and 1.4)

[63] Y. Tsuchiya, Y. Namiuchi, H. Wako, and H. Tsurui. 2018. A study of CDR3 loop dynamics reveals distinct mechanisms of peptide recognition by T-cell receptors exhibiting different levels of cross-reactivity. *Immunology* 153, 4 (2018), 466–478. DOI:http://dx.doi.org/10.1111/imm.12849 (cited on page 2.1.3)

[64] UniProt Consortium. 2025. UniProt: the Universal Protein Knowledgebase in 2025. *Nucleic Acids Research* 53, D1 (2025), D609–D617. DOI:http://dx.doi.org/10.1093/nar/gkae1010 (cited on page 2)

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Advances in Neural Information Processing Systems*. (cited on pages 2.4 and 3.4.2)

[66] N. Vigneron. 2015. Human Tumor Antigens and Cancer Immunotherapy. *BioMed Research International* 2015 (2015), 948501. DOI:http://dx.doi.org/10.1155/2015/948501 (cited on page 1.1)

[67] Randi Vita, Swapnil Mahajan, James A. Overton, Sandeep K. Dhanda, Sheridan Martini, J. Randall Cantrell, Daniel K. Wheeler, Alessandro Sette, and Bjoern Peters. 2019. The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Research* 47, D1 (2019), D339–D343. DOI:http://dx.doi.org/10.1093/nar/gky1006 (cited on page 2.5.1)

[68] Y. R. Wan, Z. Koşaloğlu-Yalçın, B. Peters, and M. Nielsen. 2024b. A large-scale study of peptide features defining immunogenicity of cancer neo-epitopes. *NAR Cancer* 6, 1 (2024), zcae002. DOI:http://dx.doi.org/10.1093/narcan/zcae002 (cited on page 2.1.5)

[69] Yat-tsai Richie Wan, Zeynep Koşaloğlu-Yalçin, Bjoern Peters, and Morten Nielsen. 2024a. A large-scale study of peptide features defining immunogenicity of cancer neo-epitopes. *NAR Cancer* 6, 1 (2024), zcae002. DOI:http://dx.doi.org/10.1093/narcan/zcae002 (cited on page 2.2)

[70] Anna Weber, Jannis Born, and Maria Rodriguez Martinez. 2021. TITAN: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics* 37, Supplement_1 (2021), i237–i244. DOI:http://dx.doi.org/10.1093/bioinformatics/btab294 (cited on page 2.6)

[71] Nan-Ping Weng. 2023. Numbers and odds: TCR repertoire size and its age changes impacting on T cell functions. *Seminars in Immunology* 69 (2023), 101810. DOI:http://dx.doi.org/10.1016/j.smim.2023.101810 (cited on pages 1.2 and 1.1)

[72] K. W. Wucherpfennig, M. J. Call, L. Deng, and R. Mariuzza. 2009. Structural alterations in peptide–MHC recognition by self-reactive T cell receptors. *Current Opinion in Immunology* 21, 6 (2009), 590–595. DOI:http://dx.doi.org/10.1016/j.coi.2009.07.008 (cited on page 2.1.2)

[73] J. Xia, P. Bai, W. Fan, Q. Li, Y. Li, D. Wang, L. Yin, and Y. Zhou. 2021. NEPdb: A Database of T-Cell Experimentally-Validated Neoantigens and Pan-Cancer Predicted Neoepitopes for Cancer Immunotherapy. *Frontiers in Immunology* 12 (2021), 644637. DOI:http://dx.doi.org/10.3389/fimmu.2021.644637 (cited on page 3)

[74] N. Xie, G. Shen, W. Gao, and others. 2023. Neoantigens: promising targets for cancer therapy. *Signal Transduction and Targeted Therapy* 8 (2023), 9. DOI:http://dx.doi.org/10.1038/s41392-022-01270-x (cited on pages 1.1 and 1.3.1)