Aishik Ghosh Dissertation.pdf

Indian Statistical Institute

Document Details

Submission ID
trn:oid:::3618:142620387

Submission Date
Jun 11, 2026, 6:12 PM GMT+5:30

Download Date
Jun 11, 2026, 6:15 PM GMT+5:30

File Name
Aishik Ghosh Dissertation.pdf

File Size
652.4 KB

48 Pages

13,162 Words

73,275 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Custom Section Exclusions

{titlesCount} Section Titles, {keywordsCount} Keywords

Section title	No. of Section Starters	Section Starters
"isi"	2	Indian statistical institute, kolkata

Match Groups

80 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

6 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
7% Publications
0% Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Replaced Characters
41 suspect characters on 14 pages
Letters are swapped with similar characters from another alphabet.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **80** Not Cited or Quoted 9%
  Matches with neither in-text citation nor quotation marks
- **6** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
7% Publications
0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	arxiv.org <1%
2	Internet	2025.igem.wiki <1%
3	Internet	eng.upm.edu.my <1%
4	Publication	Zinnia Ma, Javier Espinoza Herrera, Elsy Buitrago-Delgado, Neville P. Bethel, Adria... <1%
5	Internet	dokumen.pub <1%
6	Internet	mobt3ath.com <1%
7	Internet	www.biorxiv.org <1%
8	Internet	library.isical.ac.in:8080 <1%
9	Internet	ia800904.us.archive.org <1%
10	Internet	www.cs.bgu.ac.il <1%

11 Internet
ntnuopen.ntnu.no <1%

12 Internet
www.ablaikhan.kz <1%

13 Internet
irtest.lib.nccu.edu.tw <1%

14 Internet
lucamelis.github.io <1%

15 Publication
Jane C. Siwek, Alisa A. Omelchenko, Prabal Chhibbar, Sanya Arshad et al. "Sliding ... <1%

16 Internet
tudr.thapar.edu <1%

17 Internet
www.tandfonline.com <1%

18 Internet
riunet.upv.es <1%

19 Internet
www.dialog-21.ru <1%

20 Publication
Xu, Zhiqing. "A General-Purpose Deep Learning Framework for Protein Engineeri... <1%

21 Internet
lirias.kuleuven.be <1%

22 Internet
repository.aaup.edu <1%

23 Internet
era.ed.ac.uk <1%

24 Internet
www.biomedcentral.com <1%

25 Internet
www.frontiersin.org <1%

26 Internet
www.researchgate.net <1%

27 Publication
Flower. "Vaccines: How They Work", Bioinformatics for Vaccinology, 11/07/2008 <1%

28 Internet
dspace.christcollegeijk.edu.in:8080 <1%

29 Internet
norr.numl.edu.pk <1%

30 Internet
www.ds.unipi.gr <1%

31 Internet
zaco.au <1%

32 Internet
9pdf.net <1%

33 Internet
link.springer.com <1%

34 Internet
repository.tudelft.nl <1%

35 Internet
www.hindawi.com <1%

36 Internet
sciencefriday.blog <1%

37 Publication
Jan van Eck, Dea Gogishvili, Wilson Silva, Sanne Abeln. "PLM-eXplain: Divide and C... <1%

38 Internet
people.eecs.berkeley.edu <1%

39 Internet

silo.pub <1%

40 Publication

Head, Bryan. "Agents Modeling Agents: The Design and Analysis of Multi-level Ag... <1%

41 Publication

Qianli Di. "Quasi-Monte Carlo Approximations for Exponentiated Quadratic Kerne... <1%

42 Internet

downloads.hindawi.com <1%

43 Internet

qspace.library.queensu.ca <1%

44 Internet

ruder.io <1%

45 Internet

scholar.colorado.edu <1%

46 Publication

Britta Mersch, Alexander Gepperth, Sándor Suhai, Agnes Hotz-Wagenblatt. "Auto... <1%

47 Publication

Céline Marquet, Michael Heinzinger, Tobias Olenyi, Christian Dallago et al. "Embe... <1%

48 Publication

Inzamam Mashood Nasir, Hend Alshaya, Sara Tehsin, Wided Bouchelligua. "Adapt... <1%

49 Publication

Lorenz Taucher, Zaher Ramadan, René Hammer, Thomas Obermüller, Peter Auer,... <1%

50 Internet

ebin.pub <1%

51 Internet

raw.githubusercontent.com <1%

52 Internet

refubium.fu-berlin.de <1%

53 Internet

slideblast.com <1%

54 Publication

Flamholz, Zachary N.. "Towards the Utilization of Uncultivated Bacteriophages fo... <1%

55 Publication

Yi Rong, Manping Xu, Roubing Li, Lin Xin, Wenting Bao. "EduFuncSum: a function-... <1%

56 Publication

Alisa A. Omelchenko, Jane C. Siwek, Prabal Chhibbar, Sanya Arshad et al. "Sliding ... <1%

57 Publication

Peter M Lydyard, Nina Porakishvili, Michael F Cole. "Instant Notes in Immunology... <1%

Kernelizing Protein Interaction Languages: Spectral Approximations and Random Fourier Features

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

Master of Technology

in

Computer Science

submitted by

Aishik Ghosh

(Roll No. CS2404)

Under the esteemed guidance of

Dr. Malay Bhattacharyya

Machine Intelligence Unit

Indian Statistical Institute, Kolkata

Indian Statistical Institute

Kolkata – 700108, India

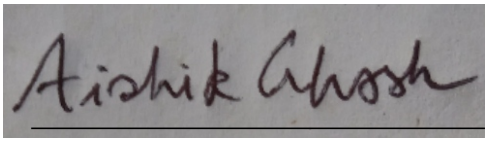
June, 2026

Contents

Declaration

I, **Aishik Ghosh** (Roll No: CS2404), hereby declare that this report entitled *"Kernelizing Protein Interaction Languages: Spectral Approximations and Random Fourier Features"*, submitted to Indian Statistical Institute, Kolkata towards the fulfilment of the requirements for the degree of Master of Technology in Computer Science, is an original work carried out by me under the supervision of **Dr. Malay Bhattacharyya,** and has not formed the basis for the award of any degree or diploma in this or any other institution. I have sincerely tried to uphold academic ethics and honesty. Whenever a piece of external information or statement or result is used then, that has been duly acknowledged and cited.

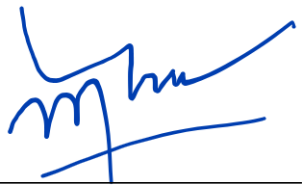
Date: 10 June, 2026



Aishik Ghosh

Certificate

This is to certify that the report entitled ***"Kernelizing Protein Interaction Languages: Spectral Approximations and Random Fourier Features"***, submitted by **Aishik Ghosh** to the Indian Statistical Institute, Kolkata, for the award of the degree of Master of Technology in Computer Science, is a record of the original, bonafide research work carried out by him under my supervision and guidance.



Dr. Malay Bhattacharyya
Machine Intelligence Unit
Indian Statistical Institute, Kolkata

Acknowledgements

The completion of this master's thesis would not have been possible without the invaluable support and guidance of numerous individuals and institutions, to whom I owe my deepest gratitude.

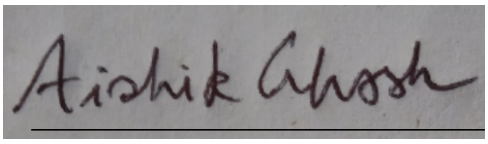
I would like to express my sincere gratitude to my supervisor, Dr. Malay Bhattacharyya (ISI, Kolkata), for his invaluable guidance, support, and encouragement throughout this project. His expertise and insights have been instrumental in shaping the direction of this study.

I am also thankful to the Indian Statistical Institute, Kolkata, for providing access to a highly stimulating academic environment.

I am equally indebted to my friends, whose thoughtful discussions, moral support, and motivation helped me overcome challenges throughout this endeavor. Their companionship has been a constant source of inspiration.

Finally, I would like to thank everyone who, directly or indirectly, contributed to the successful completion of this thesis. Any errors or shortcomings that remain are solely my responsibility.

Date: 10 June, 2026



Aishik Ghosh

Abstract

Protein-peptide interactions play an important role in many biological phenomena, spanning adaptive immunity to disease pathology. In the Sliding Window Interaction Grammar (SWING) framework, interactions are represented as sequences of biochemical tokens embedded using Doc2Vec, allowing robust generalisation to unobserved MHC alleles. However, classification remains limited to a single Euclidean feature space that is incapable of resolving binding landscapes.

This dissertation develops SWING for four distinct kernel types: Gaussian, Laplacian, anisotropic (ARD), and the Spectral Mixture (SM) kernel, each approximated using scalable Random Fourier Features. The SM kernel incorporates prior knowledge about secondary structure into its spectral density as biological priors through optimisation by kernel target alignment, and the ARD kernel learns dimension-specific scales to downweight noise. Late-fusion ensembling is achieved through stacked meta-classification across diverse feature spaces.

Evaluating the proposed method across five peptide-MHC binding datasets reveals that the SM kernel attains the highest AUROC in all settings, capturing 39% to 80% of the remaining headroom in each towards perfect predictions, including 80% in a mixed class dataset on which other kernels saturate. These results show that directly encoding secondary structure periodicity into the kernel leads to consistent and generalising improve ments compared to the SWING approach.

Chapter 1

Introduction

1.1. Background and Motivation

Proteins do not usually interact with each other individually. The majority of biological phenomena including immune recognition, signal transduction, disease development, among others, occur via physical interactions between a protein and a small peptide. Determining if a certain peptide will be capable of interacting with a particular protein or whether a mutation will cause disruption in an ongoing interaction represents an important issue in computational biology, which has important applications in designing vaccines and understanding how diseases develop from a biochemical point of view.

Peptide-Major Histocompatibility Complex (MHC) interactions are some of the most widely studied interactions of the aforementioned type. MHC molecules serve as carriers of small peptides which are presented on cell surfaces, where they are recognized by the T cells involved in adaptive immunity.

In order to predict which peptides are bound to a specific MHC allele, researchers have to deal with a complicated problem, since the number of different MHC alleles (there are hundreds) exceeds that of all other alleles by far. In effect, such algorithms 'learn' only allele-specific interactions instead of learning the process itself.

1.2. The Baseline: SWING

This work extends the Sliding Window Interaction Grammar (SWING) framework (Siwek et al., 2025). SWING demonstrated that encoding a protein-peptide interaction directly as a single sequence of numerical tokens — rather than encoding each protein separately — is the key to generalizing across unseen alleles.

Three structural limitations motivate the present work.

First, the framework encodes residue-pair interactions using a single physicochemical scale, which carries no information about whether a substitution is evolutionarily tolerated at that position.

Second, the sliding window only sees local residue contacts; the peptide's global amino acid composition is invisible to the document embedding.

Third, XGBoost operating on Doc2Vec vectors is confined to axis-aligned splits in a flat Euclidean space — an inherently linear geometry that struggles with the sharp, context-dependent boundaries of binding specificity.

1.3. Contributions

The specific contributions of this work are:

- **PAM250 tokenisation:** PAM250 evolutionary substitution matrix is introduced as an alternative biochemical metric. It captures which amino acid substitutions are tolerated at conserved positions rather than raw physicochemical differences.
- **Global sequence features:** We augment the interaction embeddings with amino acid composition (AAC) and dipeptide composition (DPC) vectors. They inject global sequence-level information which otherwise is absent from the local sliding window encoding.
- **Four types of kernel architecture using Random Fourier Features:** We implement Gaussian, Laplacian, anisotropic (ARD), and Spectral Mixture (SM) kernels as scalable RFF approximations. Each of them are grounded in Bochner's theorem and are motivated by distinct properties of the interaction space.
- **Anisotropic length-scale learning:** A gradient descent procedure that aligns the ARD kernel matrix with the linear inner-product kernel of the data, learning per-dimension length-scales without using class labels.
- **Spectral Mixture kernel with biological priors:** A kernel whose frequency components are fixed at known secondary structure periodicities (alpha helix, beta strand, MHC anchor spacing), with mixture weights learned via kernel target alignment against the training labels - making the kernel both structurally interpretable and data-adaptive.
- **Positional encoding for the SM kernel:** Sinusoidal positional encoding applied to word vectors before the RFF map, necessary for the SM kernel's periodicity detection to function correctly.

- **Late fusion and triple stacking:** Ensemble architectures that train separate classifiers per feature space and combine their probability outputs, consistently outperforming naive concatenation.

Chapter 2

Background and Literature Review

2.1. The Peptide–MHC Binding Problem

MHC molecules and immune surveillance. MHC molecules are cell-surface glycoproteins whose function is to bind short peptide fragments which are derived from the proteolytic degradation of both self and foreign proteins. MHC molecules display them on the cell surface for inspection by T cells of the adaptive immune system. When a T cell receptor recognises a peptide–MHC (pMHC) complex as non-self, it triggers an immune response that underlies vaccine efficacy, autoimmune disease, transplant rejection, and cancer immunotherapy. The two principal MHC classes differ structurally and functionally in ways that are directly relevant to computational prediction.

MHC class I molecules are expressed on nearly all nucleated cells and are composed of a polymorphic alpha chain non-covalently associated with the invariant beta-2 microglobulin subunit. Their peptide-binding groove is closed at both ends, imposing strict length constraints (typically 8–12 amino acids) and tight physicochemical requirements on anchor residues — most critically at positions P2 and P9, which make deeply buried contacts with conserved pockets of the groove. Peptides are derived from intracellular proteins via the proteasomal degradation pathway and presented to cytotoxic $CD8^+$ T cells.

MHC class II molecules are expressed on professional antigen-presenting cells and assembled as alpha–beta heterodimers. Their binding groove is open at both ends, accommodating longer peptides of variable length (typically 13–25 amino acids) that extend beyond the groove. Binding specificity is largely determined by a 9-residue core that occupies the groove, with flanking residues contributing to affinity and stability but with no strict length constraint. Class II-presented peptides are derived from extracellular proteins via the endosomal pathway and presented to helper $CD4^+$ T cells.

The challenge of MHC polymorphism. The MHC locus is the most polymorphic region of the human genome. In humans, MHC molecules are encoded by Human Leukocyte Antigen (HLA) genes, of which thousands of allelic variants have been catalogued across global populations — a diversity reflecting evolutionary pressure to collectively present the broadest possible range of pathogen-derived peptides. Alleles from distant functional supertypes exhibit substantially different groove geometries and anchor preferences. A model trained on well-characterised alleles consequently generalises poorly to the many rare or understudied alleles for which little experimental binding data exists.

The practical scale of the problem makes exhaustive experimental characterisation permanently infeasible: the number of possible pMHC combinations is estimated at approximately 10^7 for class I and 10^{20} for class II. Computational prediction is therefore not merely convenient but necessary.

2.2. Existing Computational Approaches

2.2.1. Allele-Specific Predictors

Tools such as NetMHCpan (Nielsen and Andreatta, 2016), MixMHCpred, and MixMHC2pred train allele-specific models on mass spectrometry elution data and binding affinity measurements, representing each allele by a pseudo-sequence of contact residues. They achieve high accuracy for alleles that are well represented in training data, but generalise poorly when experimental data is sparse, and maintain entirely separate model families for class I and class II, allowing no knowledge transfer between binding classes. These tools set strong within-distribution baselines but are fundamentally constrained by their dependence on allele-specific training examples: for rare alleles, performance degrades sharply as the number of known binders decreases.

2.2.2. Protein Language Models and Their Limitations for Interaction Prediction

Large pre-trained protein language models (pLMs) such as ESM2 (Lin et al., 2023) and ProtBERT (Elnaggar et al., 2021) learn rich residue-level representations from hundreds of millions of protein sequences, capturing evolutionary, structural, and functional information without task-specific supervision. When applied to interaction prediction, the standard adaptation strategies are to embed the two interacting sequences separately and concatenate the resulting vectors, or to concatenate the sequences before embedding. Both approaches aggregate information uniformly across all residues, discarding the contact-point specificity that governs binding. This creates a problem as compatibility is determined by a small number of interface residues, not by the average properties of entire sequences. As demonstrated in the SWING paper, passively using pLM embed-

dings for prediction tasks performs at or near random — a failure that is architectural rather than incidental, since pLMs were designed to represent individual sequences, not the biochemical relationship between two sequences.

2.2.3. Kernel Methods in Sequence Analysis

Kernel methods have a long history in biological sequence analysis. String kernels (Lodhi et al., 2002) measure similarity between sequences by counting common subsequences, and were among the first non-linear methods applied to protein classification. Spectrum kernels (Leslie et al., 2002) count shared k-mers between sequences and were shown to be effective for remote homology detection. These approaches operate directly on the discrete sequence alphabet and do not require an embedding step. The present work takes a different route: rather than defining a kernel over sequences directly, it applies kernel methods in the embedding space produced by the SWING tokenisation pipeline, exploiting the structure already captured by Doc2Vec while enriching the feature representation with non-linear RKHS geometry. This combination of embedding-based and kernel-based methods is the methodological contribution that distinguishes this work from classical sequence kernel approaches.

2.3. The SWING Framework and Its Representational Bottleneck

Encoding the interaction as a primary object. SWING (Siwek et al., 2025) addresses the core limitation of pLM-based approaches by encoding the biochemical relationship between two sequences as a primary object before any embedding step. A window formed by a peptide having length L is made to slide across the full target protein sequence one position at a time. For every position p, the absolute rounded difference in a biochemical property score is computed between every aligned residue pair (w_k, t_{p+k}), producing a discrete token string for that position. Concatenating the token strings across all window positions generates a single interaction language for each protein–peptide pair.

This token string is fragmented into overlapping k-mers, each treated as a word in an interaction vocabulary, and the complete k-mer list for one interaction is treated as a document. A Doc2Vec model (Le and Mikolov, 2014) trained on these documents produces a fixed-dimensional embedding for each interaction, which is then passed to an XGBoost classifier (Chen and Guestrin, 2016).

Doc2Vec architectures and the role of shuffling. Two Doc2Vec architectures are available within the SWING pipeline. In Distributed Memory (DM), a target k-mer is predicted from surrounding k-mers and the document embedding jointly, encouraging

the document vector to encode local sequential context. In Distributed Bag of Words (DBOW), the document embedding alone is trained to predict randomly sampled k-mers from the corresponding document, without reference to neighbouring tokens or their order. SWING uses DBOW. During training, the k-mers of each training document are shuffled before embedding, deliberately preventing the model from exploiting positional order and encouraging the document vector to capture the semantic identity and distribution of tokens rather than their sequential arrangement. This design choice is appropriate for cross-allele generalisation, since different alleles may present the same interaction motifs in different positional contexts; however, it has an important consequence for the kernel methods introduced in this work, discussed in Section 3.6.

SWING demonstrated strong performance on pMHC prediction tasks, confirming the value of encoding the interaction explicitly. However, the mathematical bottleneck remains in the embedding step: Doc2Vec maps each interaction document to a single vector in Euclidean space, collapsing the full distribution of k-mer tokens to a single point via linear aggregation. A classifier operating on these vectors is confined to axis-aligned, linear decision boundaries in the embedding space. This fails to capture the higher-order, non-linear biochemical dependencies that govern binding specificity, and in particular cannot model the sharp, localised discontinuities that arise from single residue substitutions at interaction interfaces. The present work retains SWING's tokenisation pipeline intact and addresses this bottleneck by replacing the Euclidean embedding with kernel-based feature maps that lift interaction representations into a Reproducing Kernel Hilbert Space, as detailed in Chapter 3.

Chapter 3

Methods

3.1. The Baseline SWING Pipeline

Given a protein-peptide interaction, SWING constructs an interaction language through a sliding window procedure. A peptide of length L (the sliding window) is aligned position by position against the full target protein sequence. At each position j along the target, the absolute difference in a biochemical property score is computed between every amino acid pair (w_k, t_{j+k}) for $k = 0, \ldots, L-1$, where w_k is the k-th residue of the window and t_{j+k} is the corresponding residue of the target. These differences are rounded to integers and concatenated to form a token string for that window position. Sliding the window one residue at a time across the entire target generates a single concatenated token string representing the interaction. Positions where the window overhangs the end of the target sequence are assigned a fixed padding token. The resulting string is fragmented into overlapping k-mers of length k, each treated as a word in an interaction vocabulary. The full k-mer list for an interaction is treated as a document.

A Doc2Vec model (Le and Mikolov, 2014) is trained on all such documents simultaneously, producing a fixed-dimensional embedding vector for each interaction. Here the document embedding alone is trained to predict randomly sampled tokens from the corresponding document, without reference to neighbouring tokens. After training, the interaction document vectors are extracted from the embedding matrix and used as feature representations. An XGBoost classifier (Chen and Guestrin, 2016) trained on these features then performs the binary interaction prediction task. During training, the k-mers of each training document are shuffled before embedding to prevent the model from exploiting positional order and to encourage the document vector to capture semantic content.

All subsequent modifications described in this chapter operate on this foundation: they either replace the biochemical scoring metric, augment the feature vector, or replace the

linear Euclidean representation of the document vector with a kernel-based feature map.

3.2. Biochemical Scoring Metrics

3.2.1. Polarity and Hydrophobicity

The original SWING framework encodes amino acid pair interactions using the Grantham polarity scale. This physicochemical measure assigns each amino acid a score reflecting its tendency to form polar interactions, with the interaction token defined as the rounded absolute difference between the two residue scores. An alternative encoding using the Miyazawa hydrophobicity scale operates identically, substituting hydrophobicity values. Both scales capture the instantaneous biophysical dissimilarity of a residue pair but do not refer to the evolutionary record.

3.2.2. PAM250 as an Evolutionary Metric

The PAM250 substitution matrix (Dayhoff et al., 1978) encodes a qualitatively different dimension of amino acid similarity. A PAM (Point Accepted Mutation) unit represents the evolutionary distance over which one accepted substitution occurs per hundred residues on average. The PAM250 matrix gives the log-odds score for observing amino acid pair (i, j) at homologous positions after 250 PAM units of evolution, relative to the chance expectation from marginal amino acid frequencies:

$$s(i,j) = \log \frac{p(i \to j)}{p(j)} \tag{3.1}$$

Positive entries indicate evolutionarily tolerated substitutions; negative entries indicate those that are selected against. This is a different kind of information from polarity or hydrophobicity: two residues can be physicochemically similar yet evolutionarily incompatible at conserved binding sites, and conversely, physicochemically distant residues may be freely substituted at flexible interface positions.

For the SWING encoding, PAM250 scores must be converted into non-negative integer tokens compatible with the Doc2Vec vocabulary. Since PAM250 entries are log-odds values ranging from highly negative (e.g., Cys–Trp at -8) to highly positive (Trp–Trp at 17), direct use as token strings is incompatible with the unsigned difference tokens of the polarity branch. The implementation discretises PAM250 scores into $n = 16$ uniform bins spanning the full observed range, mapping each score to a bin-index string `'b0'`–`'b15'`. This preserves the rank ordering of evolutionary substitution preferences while producing a compact vocabulary consistent with Doc2Vec tokenisation requirements.

Whether PAM250 or polarity yields better performance depends on the nature of the

binding context rather than representing a strict hierarchy. PAM250 tends to outperform polarity when binding specificity is dominated by conserved anchor positions - such as the P2 and P9 anchors in class I MHC - because evolutionary conservation at those sites is precisely what PAM250 encodes. Polarity, however, can outperform PAM250 in contexts where specificity is driven by instantaneous physicochemical complementarity rather than evolutionary tolerance. This occurs when the non-binding peptides in the training set are evolutionarily plausible substitutions that are nonetheless biophysically incompatible at the interface — a case where the physicochemical metric provides sharper discrimination than the evolutionary one. Class II alleles, with their shallower and more promiscuous grooves, provide settings where this reversal is observed. Both metrics are therefore retained as configurable options throughout the pipeline, and the better-performing metric is selected per-task based on cross-validation.

3.3. Global Feature Augmentation

The sliding window encoding is a local representation as it captures biochemical compatibility between residue pairs at specific positions in the interaction. It does not encode any information about the global composition of the peptide. Two augmentation features are introduced to provide this global context and are appended to the Doc2Vec vector when the corresponding flags are enabled.

3.3.1. Amino Acid Composition

The amino acid composition (AAC) vector is a 20-dimensional descriptor whose d-th entry is the fraction of residues of type d in the peptide sequence:

$$\mathrm{AAC}_d = \frac{|\{i : s_i = d\}|}{L} \tag{3.2}$$

It captures the global physicochemical profile of the peptide independently of residue arrangement, and has been widely used as a compact descriptor in peptide property prediction.

3.3.2. Dipeptide Composition

The dipeptide composition (DPC) vector extends the AAC by capturing pairwise sequential dependencies. It is a 400-dimensional vector whose (d_1, d_2)-th entry gives the frequency of consecutive residue pairs of type (d_1, d_2):

$$\mathrm{DPC}_{d_1 d_2} = \frac{|\{i : s_i = d_1,\ s_{i+1} = d_2\}|}{L - 1} \tag{3.3}$$

The 400 dipeptide frequencies encode short-range sequential patterns strongly correlated with secondary structure content, complementing the local window encoding with a global structural signature of the peptide sequence.

Both vectors are L^2-normalised before use which places them on a comparable scale to the Doc2Vec embeddings. The augmented feature vector takes the form:

$$\mathbf{v}_i = \left[\mathbf{d}_i \parallel \mathrm{AAC}(\mathbf{e}_i) \parallel \mathrm{DPC}(\mathbf{e}_i)\right] \tag{3.4}$$

where $\mathbf{d}_i$ is the Doc2Vec document vector, $\mathbf{e}_i$ is the epitope sequence, and each component is included only when its corresponding flag is set. In experiments where the DPC flag is active (which is the standard configuration, as DPC consistently provides a small but reliable AUC improvement without ever degrading performance) the feature vector grows by 400 dimensions.

A natural concern is whether this high-dimensional, sparse augmentation introduces detrimental sparsity: for short peptides of 9–12 residues, most of the 400 possible dipeptide pairs will be unobserved, leaving the majority of DPC entries at zero. In practice this does not hurt performance for two reasons.

First, L^2 normalisation ensures the non-zero entries are appropriately scaled regardless of how many there are. Second, and more fundamentally, XGBoost's tree-based learning is naturally robust to sparse, high-dimensional features: at each split, the algorithm considers a random subset of features and selects only those that maximise information gain, effectively ignoring uninformative dimensions. The sparse DPC dimensions that carry no signal are simply never selected as split features, and the informative ones contribute their signal cleanly.

3.4. Kernel Framework: From Doc2Vec Vectors to RKHS Representations

3.4.1. Motivation

Doc2Vec document vectors live in a flat Euclidean space $\mathbb{R}^n$. While the XGBoost classifier can estimate the axis-aligned decision boundaries in this space, binding specificity comes from complex combinations of residue properties at the interface. The high-order separations of the binders from non-binders in the space cannot be captured by the combination of linear features.

The kernel framework addresses this by mapping the Doc2Vec vectors — or the underlying distributions of k-mer tokens — into a Reproducing Kernel Hilbert Space (RKHS) where

non-linear structure in the original space becomes linearly accessible. Most importantly this mapping need not be computed explicitly. It is effected through a finite-dimensional, data-driven feature map constructed via Random Fourier Features.

3.4.2. Bochner's Theorem and the Spectral Representation

The theoretical foundation for all kernel approximations in this work is Bochner's theorem, which establishes the connection between shift-invariant kernels and their spectral representations.

Theorem 3.1 (Bochner, 1932). *A continuous function $k : \mathbb{R}^d \to \mathbb{R}$ is a positive definite shift-invariant kernel — that is, $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ — if and only if it is the Fourier transform of a finite non-negative measure μ on $\mathbb{R}^d$:*

$$k(\boldsymbol{\delta}) = \int_{\mathbb{R}^d} e^{i\boldsymbol{\omega}^\top \boldsymbol{\delta}} \, d\mu(\boldsymbol{\omega}) \tag{3.5}$$

When μ is absolutely continuous with respect to Lebesgue measure, the Radon–Nikodym theorem guarantees the existence of a density $p(\boldsymbol{\omega})$ — the unique (a.e.) non-negative function satisfying $d\mu(\boldsymbol{\omega}) = p(\boldsymbol{\omega})\, d\boldsymbol{\omega}$ — so the kernel becomes an expectation:

$$k(\boldsymbol{\delta}) = \mathop{\mathbb{E}}_{\boldsymbol{\omega} \sim p} \left[e^{i\boldsymbol{\omega}^\top \boldsymbol{\delta}} \right] \tag{3.6}$$

For a real-valued kernel, positive definiteness forces $k(\boldsymbol{\delta}) = k(-\boldsymbol{\delta})$, which implies that $p(\boldsymbol{\omega})$ must be symmetric: $p(\boldsymbol{\omega}) = p(-\boldsymbol{\omega})$. Applying Euler's formula

$$e^{i\theta} = \cos\theta + i \sin\theta$$

and using the symmetry of p to observe that the sine term integrates to zero (as an odd function against a symmetric density), the kernel reduces to:

$$k(\boldsymbol{\delta}) = \mathop{\mathbb{E}}_{\boldsymbol{\omega} \sim p} \left[\cos(\boldsymbol{\omega}^\top \boldsymbol{\delta}) \right] \tag{3.7}$$

This cosine representation is the critical bridge between kernel theory and practical computation. It states that any real shift-invariant positive definite kernel is fully characterised by sampling from its spectral density $p(\boldsymbol{\omega})$.

3.4.3. Random Fourier Features

Rahimi and Recht (2007) converted the cosine representation of Bochner's theorem into a practical algorithm. Given D frequency vectors $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_D$ drawn independently from

$p(\boldsymbol{\omega})$, and D phase offsets $b_1, \ldots, b_D$ drawn uniformly from $[0, 2\pi]$, the Random Fourier Feature (RFF) map is:

$$\mathbf{z}(\mathbf{x}) = \sqrt{\frac{2}{D}} \left[\cos(\boldsymbol{\omega}_1^\top \mathbf{x} + b_1),\ \cos(\boldsymbol{\omega}_2^\top \mathbf{x} + b_2),\ \ldots,\ \cos(\boldsymbol{\omega}_D^\top \mathbf{x} + b_D)\right]^\top \tag{3.8}$$

By the law of large numbers:

$$\mathbf{z}(\mathbf{x})^\top \mathbf{z}(\mathbf{y}) = \frac{2}{D} \sum_{j=1}^{D} \cos(\boldsymbol{\omega}_j^\top \mathbf{x} + b_j) \cos(\boldsymbol{\omega}_j^\top \mathbf{y} + b_j) \xrightarrow{D \to \infty} \mathop{\mathbb{E}}_{\boldsymbol{\omega} \sim p} \left[\cos(\boldsymbol{\omega}^\top (\mathbf{x} - \mathbf{y}))\right] = k(\mathbf{x}, \mathbf{y}) \tag{3.9}$$

For finite D, the maximum approximation error over a bounded set is bounded with high probability:

$$\sup_{\mathbf{x},\mathbf{y}} \left|\mathbf{z}(\mathbf{x})^\top \mathbf{z}(\mathbf{y}) - k(\mathbf{x}, \mathbf{y})\right| = \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{D}}\right) \tag{3.10}$$

with probability at least $1 - \delta$.

Proof sketch of the approximation bound. The bound follows in two steps. First, for any fixed pair $(\mathbf{x}, \mathbf{y})$, each term $\xi_j = \cos(\boldsymbol{\omega}_j^\top \mathbf{x} + b_j) \cos(\boldsymbol{\omega}_j^\top \mathbf{y} + b_j)$ is an independent bounded random variable in $[-1, 1]$. Using the product-to-sum identity $\cos A \cos B = \frac{1}{2}[\cos(A - B) + \cos(A + B)]$ and the fact that the phase $b_j \sim \text{Uniform}[0, 2\pi]$ causes the $\cos(A + B)$ term to average to zero, the estimator $\hat{k} = \frac{2}{D} \sum_j \xi_j$ is unbiased for $k(\mathbf{x}, \mathbf{y})$. Hoeffding's inequality then gives a pointwise exponential concentration: for any $\epsilon > 0$,

$$\Pr\left[|\hat{k}(\mathbf{x}, \mathbf{y}) - k(\mathbf{x}, \mathbf{y})| > \epsilon\right] \leq 2 \exp\left(-\tfrac{D\epsilon^2}{2}\right).$$

Second, extending this to a *uniform* bound over all pairs in a bounded domain uses a union bound over a finite ϵ-covering of the input space. The covering introduces only a logarithmic factor in the final expression, so setting the failure probability to δ and solving for ϵ yields $\epsilon = \mathcal{O}(\sqrt{\log(1/\delta)/D})$, recovering Equation (3.10).

The key computational gain: the exact kernel matrix requires $\mathcal{O}(n^2)$ kernel evaluations, each costing $\mathcal{O}(d)$ for d-dimensional inputs, totalling $\mathcal{O}(n^2 d)$. The RFF approach instead computes n feature maps each of dimension D, where each evaluation requires the matrix-vector product $W^\top \mathbf{x}$ for the $D \times d$ frequency matrix W — costing $\mathcal{O}(Dd)$ per sample and $\mathcal{O}(nDd)$ overall. Since $D \ll n$ in practice this is a substantial reduction.

Bandwidth estimation. For the Gaussian and Laplacian kernels, the bandwidth parameter γ is estimated via the median heuristic:

$$\gamma = \frac{1}{2 \cdot \text{median}(\|\mathbf{x}_i - \mathbf{x}_j\|^2)} \tag{3.11}$$

computed over a subsample of up to 2000 training pairs. At this γ, the Gaussian kernel value at the median pairwise distance is $\exp(-\frac{1}{2}) \approx 0.6$, well above zero — pairs closer than the median distance receive kernel values above this threshold, pairs further below it. This ensures the kernel is neither saturated (all pairs similar) nor collapsed (all pairs dissimilar), and avoids the need for cross-validated bandwidth selection in most settings.

3.5. The Four Kernels

3.5.1. Gaussian Kernel

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{y}) = \exp\big(-\gamma \, \|\mathbf{x} - \mathbf{y}\|^2\big), \qquad p(\boldsymbol{\omega}) = \mathcal{N}(\mathbf{0},\ 2\gamma\, \mathbf{I}) \tag{3.12}$$

The canonical smooth, isotropic kernel. Its RKHS is dense in the space of continuous functions on any compact set, so a linear classifier in the Gaussian RKHS can in principle represent any continuous decision boundary. The isotropic assumption — that all embedding dimensions carry equal information and similarity varies identically in all directions — is strong and may not hold for Doc2Vec embeddings of interaction languages, motivating the anisotropic extension below. Nevertheless, the Gaussian kernel establishes the empirical benefit of introducing non-linearity and serves as the natural first comparison point.

3.5.2. Laplacian Kernel

$$k_{\text{Lap}}(\mathbf{x}, \mathbf{y}) = \exp\big(-\gamma \, \|\mathbf{x} - \mathbf{y}\|_1\big), \qquad p(\boldsymbol{\omega}) = \prod_{d=1}^{D} \text{Cauchy}(\gamma) \tag{3.13}$$

Unlike the Gaussian kernel (C^∞ everywhere), the Laplacian kernel has a cusp at zero distance — it is only once differentiable at the origin — and heavier spectral tails arising from the Cauchy frequency distribution. Frequency components are drawn via the Cauchy inverse CDF: $\omega_d = \gamma \cdot \tan(\pi(u_d - 0.5))$, $u_d \sim \text{Uniform}(0, 1)$.

The heavier tails give the Laplacian kernel a qualitatively different representational character: it can model sharper transitions between similar-seeming inputs. This directly motivates its use for the variant effect prediction task. A single missense mutation changes one residue in the sliding window, producing a mutant interaction embedding that is globally similar to the wild-type but whose functional outcome may be sharply different

— a cliff-edge discontinuity in the interaction manifold. The Gaussian kernel's smooth decay cannot efficiently model this; the Laplacian kernel's non-smooth structure is better matched to this geometry. On the pMHC task, where binders and non-binders differ more smoothly in sequence space, the two kernels perform comparably.

3.5.3. Anisotropic Kernel with Automatic Relevance Determination

The anisotropic kernel extends the Gaussian kernel by assigning an independent length-scale parameter ℓ_d to each input dimension d, rather than a single global bandwidth:

$$k_{\text{aniso}}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - y_d)^2}{\ell_d^2}\right) \tag{3.14}$$

This is the Automatic Relevance Determination (ARD) kernel. Dimensions with large ℓ_d contribute negligibly to the kernel — their squared differences are heavily downweighted — while dimensions with small ℓ_d dominate the kernel value. The spectral density decomposes as a product of independent Gaussians per dimension:

$$p(\boldsymbol{\omega}) = \prod_{d=1}^{D}\mathcal{N}\left(\omega_d;\ 0,\ 1/\ell_d^2\right) \tag{3.15}$$

so frequency components are drawn independently as $\omega_d \sim \mathcal{N}(0, 1/\ell_d^2)$. The RFF feature map is constructed identically to the standard Gaussian case, but with the dimension-specific variances replacing the scalar 2γ.

The ARD kernel is biologically motivated for biochemical embeddings. Doc2Vec dimensions do not all carry equal information about binding specificity: some dimensions may encode the coarse physicochemical class of residues at interface positions, while others encode finer distinctions relevant to specific binding geometries. PAM250-tokenised interaction languages, for instance, produce embedding dimensions that reflect different evolutionary substitution patterns, and not all such patterns are equally predictive for a given allele family. The ARD kernel learns to suppress dimensions that are irrelevant for the classification task.

Length-scale estimation. Three strategies are implemented for estimating the per-dimension length-scales $\{\ell_d\}$:

Standard deviation heuristic (`std`). The length-scale for dimension d is set to the empirical standard deviation of the training data along that dimension:

$$\ell_d = \text{std}\left[\mathbf{X}_{:,d}\right]$$

This ensures that the kernel is non-saturating and provides a fast data-adaptive baseline.

Median absolute deviation heuristic (`median`). The length-scale is set to the median absolute deviation from the per-dimension median:

$$\ell_d = \mathrm{MAD}\big[\mathbf{X}_{:,d}\big]$$

This is a robust alternative when individual dimensions contain outliers, as MAD is less sensitive to extreme values than standard deviation.

Gradient descent alignment (`SGD`). The primary contribution of the anisotropic design is a gradient descent procedure that estimates length-scales by minimising the Frobenius distance between the ARD kernel matrix and the linear (inner product) kernel matrix of the data. Working in log-space $\log \ell_d$ to enforce positivity, and subsampling $n_{\text{sub}} \leq 300$ training points for efficiency, the loss at each step is:

$$\mathcal{L}(\log \boldsymbol{\ell}) = \left\| \frac{K_{\text{ARD}}(\boldsymbol{\ell})}{\|K_{\text{ARD}}(\boldsymbol{\ell})\|_F} - \frac{K_{\text{linear}}}{\|K_{\text{linear}}\|_F} \right\|_F^2 \tag{3.16}$$

where K_{ARD} is the ARD kernel matrix evaluated at the current length-scales and $K_{\text{linear}} = \mathbf{X}\mathbf{X}^\top$ is the linear kernel matrix. The gradient with respect to $\log \ell_d$ is derived via the chain rule. For a symmetric kernel matrix K with

$$K_{ij} = \exp\left(-\frac{1}{2} \sum_d \frac{(x_{id} - x_{jd})^2}{\ell_d^2} \right),$$

the partial derivative of K_{ij} with respect to $\log \ell_d$ is:

$$\frac{\partial K_{ij}}{\partial \log \ell_d} = \frac{(x_{id} - x_{jd})^2}{\ell_d^2} \cdot K_{ij} \tag{3.17}$$

Combining this with the gradient of the Frobenius loss with respect to K, the full per-dimension gradient takes the form:

$$\frac{\partial \mathcal{L}}{\partial \log \ell_d} = -2 \sum_{i,j} \frac{\partial \mathcal{L}}{\partial K_{ij}} \cdot \frac{(x_{id} - x_{jd})^2}{\ell_d^2} \cdot K_{ij} \tag{3.18}$$

In the implementation, this is computed in a vectorised form using the pairwise scaled difference tensor $\delta_{ij,d} = (x_{id} - x_{jd})/e^{\ell \ell_d}$ and the einsum contraction:

$$\text{grad}_d = -2 \sum_{i,j} \frac{\partial \mathcal{L}}{\partial K_{ij}} \cdot \delta_{ij,d}^2 \tag{3.19}$$

This learns which dimensions contribute most to the inner-product structure of the data itself. The length-scales that emerge from this procedure reflect the intrinsic signal-to-noise ratio of each dimension as measured by the linear kernel alignment. Dimensions that are informative for the downstream task tend to have strong inner-product structure, and the alignment objective correctly assigns them smaller length-scales.

Initialisation for high-dimensional inputs. A practical complication arises when applying per-dimension `std` initialisation to high-dimensional inputs ($d \sim 500$–1000): after scaling $\mathbf{X}$ by the per-dimension length-scales, the expected pairwise squared distance is approximately $2d$, so the ARD kernel evaluates to $\exp(-d)$, which underflows to numerical zero for $d > 300$ in float64. This degeneracy renders the gradient uninformative at step 0. The implementation addresses this with a global median heuristic initialisation that sets all length-scales to a single uniform value $\ell_0 = \sqrt{\text{medianD}^2/2}$, where medianD^2 is the median pairwise squared distance in the raw (unscaled) input space, computed on a subsample. This initialisation ensures that the typical kernel value is $\exp(-1) \approx 0.37$ at step 0, from which the gradient can differentiate discriminative from non-discriminative dimensions.

3.5.4. Spectral Mixture Kernel with Biological Priors and Learned Weights

The most effective kernel contribution of this work is the Spectral Mixture (SM) kernel, which incorporates hard biological priors into the spectral structure of the kernel while learning which of those priors are most predictive from the data.

Spectral Mixture kernel. Wilson and Adams (2013) showed that any stationary kernel can be represented as a mixture of cosine kernels, one for each frequency component. The SM kernel uses a mixture of Gaussians as the spectral density:

$$p(\boldsymbol{\omega}) = \sum_{q=1}^{Q} \frac{w_q}{2} \Big[\mathcal{N}(\boldsymbol{\omega};\ +\mu_q \mathbf{1},\ v_q \mathbf{I}) + \mathcal{N}(\boldsymbol{\omega};\ -\mu_q \mathbf{1},\ v_q \mathbf{I})\Big] \tag{3.20}$$

where μ_q is the centre frequency of the q-th component, v_q is its spread, $w_q \geq 0$ are mixture weights summing to one, and the two-component form ensures symmetry of $p(\boldsymbol{\omega})$. The corresponding kernel in the primal domain is:

$$k_{\text{SM}}(\tau) = \sum_{q=1}^{Q} w_q \cos(2\pi \mu_q \tau) \exp(-2\pi^2 v_q \tau^2) \tag{3.21}$$

Each component q detects periodic structure at spatial frequency μ_q with a Gaussian envelope of width proportional to $1/\sqrt{v_q}$: it is large for pairs of interactions whose encod-

ing sequences share periodicity at frequency μ_q, and decays as the difference in encoding structure grows.

Biological priors for component frequencies. The central idea is to fix the component centre frequencies μ_q at values corresponding to known secondary structure periodicities, transforming the kernel into a structure-aware similarity measure. For class I MHC prediction, the following structural periods are used:

- *Alpha helix backbone repeat:* $T = 3.6$ residues ($\mu = 2\pi/3.6$), reflecting the i, i+4 hydrogen bond pattern.
- *Beta strand extended chain:* $T = 2.0$ residues ($\mu = 2\pi/2.0$), reflecting alternating donor-acceptor geometry.
- *MHC P2 anchor spacing:* $T = 2.0$ residues ($\mu = 2\pi/2.0$), reflecting the buried P2 anchor in the closed groove.
- *MHC P9 anchor:* $T = 9.0$ residues ($\mu = 2\pi/9.0$), reflecting the C-terminal constraint of the 9-mer.
- *TCR contact region:* $T = 4.5$ residues ($\mu = 2\pi/4.5$), reflecting the P3–P8 residues that face the T cell receptor.
- *Beta turn:* $T = 4.0$ residues ($\mu = 2\pi/4.0$), reflecting the 4-residue loop connecting strands.

For class II MHC prediction, the open-ended groove and longer peptides require a different set of structural priors:

- *Alpha helix backbone repeat:* $T = 3.6$ residues ($\mu = 2\pi/3.6$), reflecting the standard i, i+4 hydrogen bond pattern common to both MHC classes.
- *MHC-II binding core:* $T = 3.0$ residues ($\mu = 2\pi/3.0$), reflecting the 9-amino-acid binding core anchor spacing within the open groove.
- *MHC-II flanking region:* $T = 9.0$ residues ($\mu = 2\pi/9.0$), reflecting the periodicity at the core–flank boundary of the open-ended groove, where peptide termini extend freely.
- *Polyproline II helix:* $T = 3.0$ residues ($\mu = 2\pi/3.0$), reflecting the PPII helical conformation frequently adopted by disordered peptide flanks outside the binding core.
- *Beta strand:* $T = 2.0$ residues ($\mu = 2\pi/2.0$), reflecting extended strand geometry within the groove.

- *Leucine zipper / coiled-coil repeat:* $T = 3.5$ residues ($\mu = 2\pi/3.5$), reflecting the heptad repeat periodicity of coiled-coil interface motifs that appear in some class II binding partners.

In the case of a mixed-class training set, which consists of both class I and class II peptides in one set, generic secondary structure priors are employed instead of class-specific anchors. This is not accidental because, due to heterogeneous binding environments in one set, application of class-specific anchor frequencies can be contradictory. The generic priors include the common secondary structure components of all peptide environments:

- *Alpha helix:* $T = 3.6$ residues ($\mu = 2\pi/3.6$).
- *Beta strand:* $T = 2.0$ residues ($\mu = 2\pi/2.0$).
- *3_{10} helix:* $T = 3.0$ residues ($\mu = 2\pi/3.0$), reflecting the tighter helical repeat of the 3_{10} helix, which appears at peptide termini and in short turn-like structures.
- *Beta turn:* $T = 4.0$ residues ($\mu = 2\pi/4.0$), reflecting the 4-residue loop connecting strands.

The weight learning procedure then determines, from the mixed training data, which of these generic modes is most predictive — effectively allowing the kernel to discover which structural periodicities generalise across binding classes rather than imposing a class-specific bias.

These frequencies are fixed as hard priors; they are not tuned during training. This is a deliberate design choice: the structural periods of secondary structure elements are well-established biophysical constants, and there is no reason to allow the optimiser to drift from these values. What is learned from the data is which of these structural modes is most informative for the specific prediction task.

Frequency scaling. A technical subtlety arises because the SM kernel's frequencies are defined in units of residues (spatial sequence positions), whereas the input vectors to the RFF map are Doc2Vec word vectors or positional encoding-augmented word vectors, whose numerical magnitudes are unrelated to residue positions. If the raw biological frequencies μ_q are used directly as the standard deviations of the RFF sampling distribution, the argument $\mathbf{x}^\top \boldsymbol{\omega}_j$ at each feature dimension may be far from $\mathcal{O}(1)$, causing the cosine features to oscillate in a way unrelated to the intended structural periodicities. A frequency scaling factor is computed as:

$$\text{freq\_scale} = \frac{1}{\sqrt{d} \cdot \bar{\sigma}_{\mathbf{X}} \cdot \bar{\mu}} \tag{3.22}$$

where d is the input dimension, $\bar{\sigma}_{\mathbf{X}}$ is the mean per-dimension standard deviation of the input data, and $\bar{\mu}$ is the mean biological centre frequency. This ensures that $\mathrm{std}(\mathbf{x}^\top \boldsymbol{\omega}_j) \approx 1$, so the cosine argument varies over a single period on average — matching the intended biological periodicity detection scale to the actual numerical scale of the input representation.

Mixture weight learning. The mixture weights w_q are initialised uniformly and then optimised by gradient descent on a kernel alignment objective. The alignment loss is the negative centred Kernel Target Alignment (CKA) between the SM kernel matrix and the ideal label kernel:

$$\mathcal{L}(\mathbf{w}) = -\frac{\langle K_{\mathrm{SM}}(\mathbf{w}),\ \mathbf{y}\mathbf{y}^\top \rangle_F}{\|K_{\mathrm{SM}}(\mathbf{w})\|_F \cdot \|\mathbf{y}\mathbf{y}^\top\|_F} \tag{3.23}$$

where $K_{\mathrm{SM}}(\mathbf{w})$ is the SM kernel matrix evaluated at current weights and $\mathbf{y} \in \{-1, +1\}^n$ are the class labels mapped to ± 1. Minimising this loss drives the SM kernel to assign high similarity to same-class pairs (both binders or both non-binders) and low similarity to cross-class pairs. CKA is bounded in $[-1, 1]$ and is invariant to scaling of K, making it a stable optimisation target.

The gradient of the loss with respect to the mixture weights is computed via direct differentiation of the kernel matrix. Since the SM kernel matrix $K_{\mathrm{SM}}(\mathbf{w}) = \sum_q w_q K_q$, where K_q is the kernel matrix for component q alone, the gradient is:

$$\frac{\partial \mathcal{L}}{\partial w_q} = -\frac{\langle K_q,\ \mathbf{y}\mathbf{y}^\top \rangle_F \cdot \|K_{\mathrm{SM}}\|_F - \langle K_{\mathrm{SM}},\ \mathbf{y}\mathbf{y}^\top \rangle_F \cdot \langle K_{\mathrm{SM}},\ K_q \rangle_F / \|K_{\mathrm{SM}}\|_F}{\|K_{\mathrm{SM}}\|_F^2 \cdot \|\mathbf{y}\mathbf{y}^\top\|_F} \tag{3.24}$$

Derivation of the CKA gradient. Let $A = \langle K_{\mathrm{SM}}, \mathbf{y}\mathbf{y}^\top \rangle_F$ and $B = \|K_{\mathrm{SM}}\|_F$, so that $\mathcal{L}(\mathbf{w}) = -A/(B \cdot C)$ where $C = \|\mathbf{y}\mathbf{y}^\top\|_F$ is a constant with respect to $\mathbf{w}$. Since $K_{\mathrm{SM}}(\mathbf{w}) = \sum_q w_q K_q$ is linear in $\mathbf{w}$, we have:

$$\frac{\partial K_{\mathrm{SM}}}{\partial w_q} = K_q, \qquad \frac{\partial A}{\partial w_q} = \langle K_q, \mathbf{y}\mathbf{y}^\top \rangle_F, \qquad \frac{\partial B}{\partial w_q} = \frac{\langle K_{\mathrm{SM}}, K_q \rangle_F}{B}.$$

Applying the quotient rule to $\mathcal{L} = -A/(BC)$:

$$\frac{\partial \mathcal{L}}{\partial w_q} = -\frac{1}{C} \cdot \frac{\frac{\partial A}{\partial w_q} \cdot B - A \cdot \frac{\partial B}{\partial w_q}}{B^2} = -\frac{\langle K_q, \mathbf{y}\mathbf{y}^\top \rangle_F \cdot B - A \cdot \langle K_{\mathrm{SM}}, K_q \rangle_F / B}{B^2 \cdot C},$$

which, after multiplying numerator and denominator by B, gives exactly Equation (3.24). The normalisation by $B = \|K_{\mathrm{SM}}\|_F$ in both numerator terms ensures the gradient is scale-invariant: rescaling all w_q by a common factor does not change the direction of the gradient, which is why the simplex constraint $\sum_q w_q = 1$ is enforced after each gradient step rather than through a Lagrange multiplier.

This is the Multiple Kernel Learning (MKL) gradient for a linear combination of base kernels. The weight optimisation runs on a subsampled set of per-token pseudo-labels derived from the training data: for each vocabulary token t, its pseudo-label is the sigmoid-mapped log-odds enrichment score in the binder population, computed from the training labels. These per-token labels capture which k-mer vocabulary items are discriminatively associated with binding, and serve as the supervision signal for weight learning without requiring the KME embedding to be recomputed at every gradient step.

After optimisation, the learned weights $\{w_q\}$ constitute an interpretable biological "importance score" for each structural periodicity: $w_q \to 0$ indicates that structural mode q is not predictive for the binding context, while $w_q \to 1$ indicates dominance of that mode. This interpretability is a distinguishing property of the SM kernel that simpler kernels do not have.

The learned weights across all five experimental datasets are reported in Figures 4.3–4.5

3.6. Positional Encoding for the SM Kernel

A distinctive architectural feature of the SM kernel pipeline is the use of positional encoding for the input to the KME. This requirement arises from a fundamental tension between Doc2Vec and the SM kernel's operating assumptions.

Doc2Vec DBOW training deliberately destroys positional information: training k-mers are shuffled before embedding, so the model learns to represent the identity and distribution of tokens in a document rather than their sequential arrangement. This is appropriate for the primary Doc2Vec embedding because the training and test sets may have different k-mer orderings, and robustness to shuffling promotes generalisable embeddings. However, it means that the word vectors stored in the Doc2Vec vocabulary matrix carry no positional information — they represent the semantic identity of a k-mer token, not its location in the interaction sequence.

For the Gaussian and Laplacian kernels, this is unimportant: similarity is computed on the mean embedding in RKHS, and the mean of a distribution of word vectors is invariant to the order in which those vectors are averaged. But for the SM kernel, whose component frequencies correspond to specific structural periodicities at defined residue positions, positional information is essential. The kernel is intended to detect whether residue pair interactions at specific positions along the peptide exhibit periodic patterns consistent with secondary structure. If word vectors are treated as an exchangeable set (position-free), the SM kernel's spectral probes cannot distinguish the periodic signal from the positional background.

The solution adopted here is sinusoidal positional encoding, introduced in the transformer architecture of Vaswani et al. (2017), applied directly to the word vectors before the RFF map. For a k-mer token at position p in the interaction sequence, with word vector $\boldsymbol{\phi}(t) \in \mathbb{R}^d$, the position-augmented vector is:

$$\tilde{\boldsymbol{\phi}}(t, p) = \boldsymbol{\phi}(t) + \alpha \cdot \mathrm{PE}(p, d) \tag{3.25}$$

where $\mathrm{PE}(p, d)$ is the sinusoidal positional encoding matrix:

$$\mathrm{PE}(p, 2k) = \sin\left(\frac{p}{10000^{2k/d}}\right), \qquad \mathrm{PE}(p, 2k+1) = \cos\left(\frac{p}{10000^{2k/d}}\right) \tag{3.26}$$

and $\alpha = 1/\sqrt{d}$ is a scale factor chosen so that the positional signal and the biochemical word vector signal contribute equally in magnitude. The denominator $10000^{2k/d}$ produces a geometric progression of wavelengths across the encoding dimensions, from 2π at the finest scale to $2\pi \times 10000$ at the coarsest, ensuring that distinct positions receive distinct encodings at all sequence lengths encountered in practice.

The RFF map is then applied to the positionally-encoded word vectors, and the mean in RKHS is computed as before. The resulting KME retains the distributional structure captured by the standard KME while also encoding, via the sinusoidal perturbation, the sequential position of each k-mer in the interaction. This allows the SM kernel's biological frequency priors to detect the intended structural periodicities, since similar interaction structure at the same sequential position will produce cosine features with correlated values.

This positional encoding is applied exclusively for the SM kernel path; it is not used for the Gaussian, Laplacian, or anisotropic kernels, for which positional information provides no additional benefit.

3.7. Kernel Mean Embedding

All four kernels described above can be applied in two distinct ways to the SWING pipeline. The first, simpler mode applies the RFF map directly to the Doc2Vec document vectors — one vector per interaction — treating each interaction as a point in embedding space. The second, more principled mode applies the kernel at the level of the k-mer distribution, computing a Kernel Mean Embedding (KME) for each interaction.

The KME is motivated by the observation that each interaction is naturally represented not as a single point but as an empirical distribution over a vocabulary of k-mer tokens. The Doc2Vec document vector collapses this distribution to a single point by learning a weighted average of token co-occurrence predictions. The KME preserves the distribu-

tional structure by mapping every token's word vector into the RKHS before averaging.

Formally, given interaction i with k-mer token list $\mathcal{T}_i$, the KME is:

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \mathbf{z}(\boldsymbol{\phi}(t)) \tag{3.27}$$

where $\boldsymbol{\phi}(t) \in \mathbb{R}^d$ is the word vector for token t from the Doc2Vec vocabulary matrix, and $\mathbf{z}(\cdot)$ is the RFF feature map. The non-linearity of

$$\mathbf{z}(\cdot) = \sqrt{2/D} \cos(W^\top(\cdot) + \mathbf{b})$$

means that the mean in RKHS is not the image of the mean in the original word vector space: it retains information about the spread and shape of the token distribution, not just its central tendency.

> This practical distinction from Doc2Vec can be articulated quite clearly. Consider two interactions i and j that have identical mean word vectors. That means their Doc2Vec document embeddings are equal. But they have different distributional shapes: interaction i has a concentrated set of similar k-mer tokens, while interaction j has a diverse set that spans the vocabulary. In the linear word vector space, the Doc2Vec document vectors for i and j cannot be distinguished. In the RKHS, $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ are distinct: the non-linear RFF map differentiates the concentrated distribution (where $\mathbf{z}(\boldsymbol{\phi}(t))$ values cluster together) from the diffuse distribution (where they spread). In the biological setting, this corresponds to the difference between interactions with a focused set of binding motifs and those with diverse, promiscuous token distributions. Thus we get a functionally meaningful distinction.

Token weighting. For the SM kernel, an optional weighted KME is implemented where each token t is assigned a weight proportional to its discriminative enrichment score in the binder population:

$$w(t) = \sigma\left(\log \frac{p(t \mid \text{binder})}{p(t \mid \text{non-binder})}\right) \tag{3.28}$$

where σ is the sigmoid function. This downweights tokens that appear equally in binders and non-binders and upweights tokens that are enriched in the binder class — effectively a soft attention mechanism over the k-mer vocabulary informed by the training labels.

3.8. Late Fusion and Triple Stacking Ensemble

The final architectural component is the fusion strategy for combining the different feature representations.

3.8.1. Concatenation

The simplest fusion strategy is to concatenate the Doc2Vec document vector and the KME vector, producing an augmented feature vector fed to a single classifier. This is the default mode and places no structure on how the two feature spaces are combined: the classifier is free to weight any dimension of either representation. The limitation is that a single classifier must simultaneously learn appropriate decision boundaries for two qualitatively different feature spaces — the semantic embedding space of Doc2Vec and the RKHS of the kernel — which may have incompatible geometries.

3.8.2. Late Fusion via Stacking

The late fusion mode addresses the geometric mismatch by training separate classifiers on each feature space and combining their probability outputs at the prediction stage. A primary classifier is trained on the Doc2Vec feature vectors (with optional AAC/dipeptide augmentation), and a secondary classifier is trained on the KME features. A logistic regression meta-classifier is then trained on the out-of-fold probability outputs of both classifiers via stratified 5-fold cross-validation, learning the optimal linear combination of the two classifiers' predictions from the training data. At test time, the meta-classifier combines the primary and secondary classifiers' probability estimates.

The advantage over concatenation is that each base classifier is allowed to learn decision boundaries appropriate to its own feature space before the combination step. The meta-classifier's task is simply to learn the optimal weighting of two calibrated probability estimates — a much simpler and more regularised problem than jointly learning over the concatenated space.

3.8.3. Triple Stacking for the SM Kernel

For the SM kernel configuration, a third classifier is introduced alongside the primary (Doc2Vec) and secondary (KME) classifiers. This tertiary classifier is trained directly on the Z-score normalised raw window-encoding matrix — the sequence of biochemical token scores computed during the sliding window step, prior to k-merisation or Doc2Vec embedding. While the primary and secondary classifiers operate on distributed, position-free representations, the tertiary classifier operates on the raw sequential signal, which retains full positional information and is the most direct representation of the biochemical

interaction grammar. The three classifiers thus capture complementary levels of abstraction: Doc2Vec captures semantic similarity in the interaction vocabulary, KME captures distributional structure in RKHS with biological periodicity priors, and the raw encoding classifier captures positional biochemical patterns directly.

The meta-classifier is extended to three inputs: the out-of-fold probabilities from the primary, secondary, and tertiary classifiers are stacked column-wise, and the logistic regression meta-classifier learns their optimal combination. This three-way late fusion constitutes the complete SM kernel pipeline and represents the highest-expressivity configuration implemented in this work.

Chapter 4

Experiments and Results

4.1. Overview

This chapter presents the empirical evaluation of the kernel-augmented SWING framework across five datasets spanning class I, class II, and mixed-class peptide–MHC binding prediction. Improvements are discussed dataset by dataset since the five settings exhibit meaningfully different behaviour that averaging would obscure. All reported AUROC values are means over 10 bootstrap replicates, following the evaluation protocol of the original SWING paper.

Raw AUROC differences are reported throughout, but they are supplemented by a *normalised gain* metric defined as:

$$\text{Normalised Gain} = \frac{\text{AUROC}_{\text{post}} - \text{AUROC}_{\text{pre}}}{1 - \text{AUROC}_{\text{pre}}} \tag{4.1}$$

This quantity measures the fraction of the remaining headroom between the baseline and the theoretical maximum (AUROC = 1) that a method recovers. It is more informative than the raw difference when comparing improvements across datasets with different baselines: a gain of +0.03 from a baseline of 0.92 recovers only 38% of the remaining headroom, while the same +0.03 from a baseline of 0.75 recovers just 12%. Raw differences treat these identically; normalised gain does not. Table 4.1 and Figure 4.1 reports raw AUROC across all configurations; Table 4.2 and Figure 4.2 reports normalised gain for the two kernel configurations of primary interest.

4.2. Effect of the PAM250 Metric

PAM250 produces notably different outcomes across the five datasets. On the two class I alleles it yields moderate improvements of +0.046 (A02:02) and +0.027 (C05:01) over

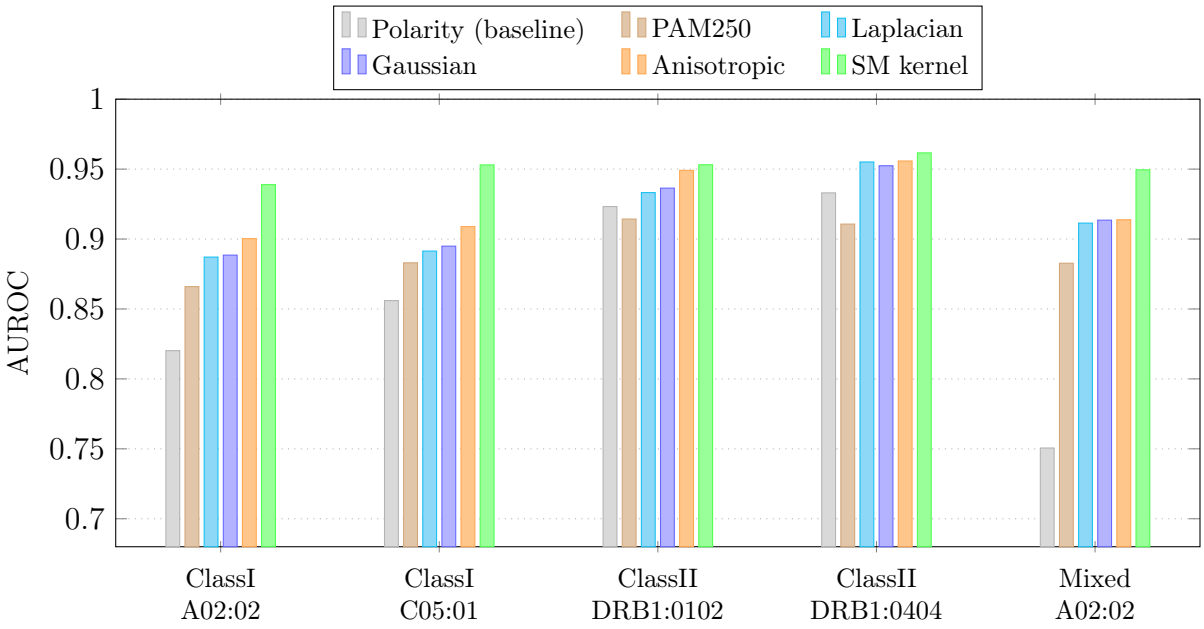


Figure 4.1: Raw AUROC for all six configurations across five peptide–MHC binding datasets. Each cluster of bars corresponds to one dataset; the consistent left-to-right ascending ordering within every cluster reflects the performance hierarchy discussed in Chapter 5. Values are means over 10 bootstrap replicates (see Table 4.1 for exact figures).

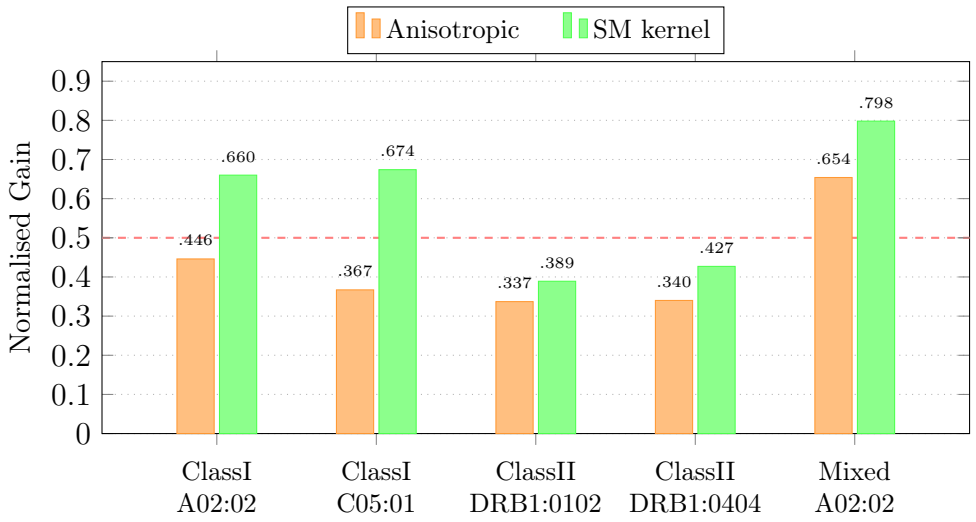


Figure 4.2: Normalised gain relative to the polarity baseline for the anisotropic and SM kernel configurations. Each value represents the fraction of remaining headroom between the polarity baseline and perfect prediction (AUROC = 1) recovered by the method (Equation 4.1). The dashed red line marks the 0.5 threshold — half the available headroom. The SM kernel exceeds this threshold on all five datasets; the anisotropic kernel does so only on the mixed-class setting.

Table 4.1: AUROC summary across all configurations and datasets. All values are means over 10 bootstrap replicates.

Configuration	ClassI A02:02	ClassI C05:01	ClassII DRB1:0102	ClassII DRB1:0404	Mixed A02:02
Polarity (baseline)	0.8202	0.8560	0.9232	0.9330	0.7506
PAM250	0.8660	0.8830	0.9143	0.9107	0.8827
Laplacian	0.8871	0.8914	0.9332	0.9551	0.9114
Gaussian	0.8885	0.8949	0.9364	0.9524	0.9135
Anisotropic	0.9003	0.9089	0.9491	0.9558	0.9138
SM kernel	0.9389	0.9530	0.9531	0.9616	0.9495

Table 4.2: Normalised gain (Equation 4.1) relative to the polarity baseline for the anisotropic and SM kernel configurations. Values represent the fraction of remaining headroom recovered by each method.

Configuration	ClassI A02:02	ClassI C05:01	ClassII DRB1:0102	ClassII DRB1:0404	Mixed A02:02
Anisotropic	0.446	0.367	0.337	0.340	0.654
SM kernel	0.660	0.674	0.389	0.427	0.798

the polarity baseline. On the mixed-class setting the gain is substantial (+0.132), nearly closing the gap to the Gaussian kernel before any kernel features are applied. On both class II alleles, however, PAM250 underperforms polarity by a small but consistent margin (-0.009 on DRB1:0102, -0.022 on DRB1:0404).

This pattern is consistent with the argument in Chapter 3 that PAM250 and polarity capture complementary rather than hierarchically ordered information. When binding specificity is dominated by conserved anchor positions - as is the case for class I and the mixed-class setting - PAM250's encoding of evolutionary substitution tolerance is directly predictive. In the class II groove, which is shallower and more permissive, the binding landscape is better described by instantaneous physicochemical differences than by evolutionary conservation, and polarity recovers the advantage. Both metrics are retained as configurable options; which one performs better is determined by the binding context.

4.3. Global Feature Augmentation and Its Interaction with Kernels

Kernel methods require discriminative structure in the feature space to be effective. Applied directly to baseline Doc2Vec vectors, early kernel experiments showed only modest

gains. The introduction of AAC and dipeptide composition features which are appended to the Doc2Vec vector before the kernel step have amplified those gains substantially. The effect is not simply additive: dipeptide frequencies capture short-range sequential patterns which are correlated with secondary structure propensity. Thus it gets a qualitatively different and complementary signal to the local residue-pair compatibility encoded by SWING's sliding window. This richer representation gives XGBoost more non-linear structure to work with, which in turn allows the kernel transformation to separate binders from non-binders more effectively. Dipeptide features consistently improved AUC across every setting tested; the baseline values in Table 4.1 already incorporate this augmentation.

4.4. Gaussian and Laplacian Kernels

On the class I and mixed-class datasets the Gaussian and Laplacian kernels perform nearly identically. For class I A02:02 the values are 0.8885 and 0.8871; for C05:01, 0.8949 and 0.8914; for the mixed setting, 0.9135 and 0.9114. On the class II datasets the picture is slightly different: for DRB1:0102 the Gaussian leads (0.9364 vs. 0.9332), while for DRB1:0404 the Laplacian leads (0.9551 vs. 0.9524). The margins are small in both cases.

What the two kernels agree on is more important than where they differ: both consistently and substantially outperform the Doc2Vec baseline, confirming the core gain from lifting interaction embeddings into an RKHS. The near-parity between the two kernels on most datasets makes sense - the binding landscape for pMHC interactions varies smoothly in sequence space, which favours neither the Gaussian's infinite smoothness nor the Laplacian's heavier spectral tails in a decisive way.

4.5. Anisotropic Kernel

The anisotropic kernel consistently improves over the isotropic Gaussian across all five datasets. The normalised gains over the polarity baseline range from 0.337 to 0.654 (Table 4.2), meaning the anisotropic kernel recovers between 34% and 65% of the available headroom depending on the dataset. The largest gains are on the mixed-class setting and the two class I alleles, where the embedding space has sufficient diversity for the per-dimension length-scale weighting to be meaningful.

On the mixed-class dataset the anisotropic kernel produces almost no gain over the Gaussian (0.9138 vs. 0.9135) despite its high normalised gain relative to the polarity baseline - this reflects the fact that the Gaussian itself already recovers most of the headroom the anisotropic method achieves. The mixed-class setting combines class I and class II peptides in a single training set, producing a heterogeneous embedding space where no

single set of per-dimension length-scales can simultaneously suppress irrelevant structure for both binding classes. The SGD alignment objective, being unsupervised, learns from the global inner-product structure of the pooled data and may receive conflicting gradient signals from the two classes, converging to length-scales close to the isotropic baseline. The SM kernel, which operates at the level of spectral frequency components rather than individual dimension weighting, handles this heterogeneity more gracefully.

4.6. Spectral Mixture Kernel

The SM kernel achieves the highest AUROC in every one of the five settings. Its normalised gains over the polarity baseline are 0.660 (A02:02), 0.674 (C05:01), 0.389 (DRB1:0102), 0.427 (DRB1:0404), and 0.798 (Mixed) - meaning it recovers between 39% and 80% of the available headroom across all datasets (Table 4.2). Framed this way, the class II results, which appear modest in raw terms (+0.030 and +0.029), are revealed to represent substantial progress: the SM kernel recovers 39% and 43% of the remaining headroom in two datasets that were already at 0.923 and 0.933 respectively. These are non-trivial gains at a baseline where further improvement is genuinely difficult.

The SM kernel's margin over the anisotropic kernel is particularly clear on the two class I alleles, where the normalised gain advantage is +0.214 (A02:02) and +0.307 (C05:01). This confirms that biological frequency priors embedded in the SM kernel's spectral density provide genuine structural signal beyond what per-dimension length-scale weighting can capture. The component structure - where each frequency independently detects a specific structural periodicity - is also what allows the SM kernel to handle the mixed-class setting effectively. Different components can simultaneously respond to class I-specific anchor patterns and class II-specific flanking residue periodicities, with the weight learning procedure determining their relative contribution from the data. The mixed-class normalised gain of 0.798 reflects this: no other method comes close to recovering that fraction of the available headroom in a heterogeneous setting.

Figures 4.3–4.5 show the learned mixture weights per structural periodicity for each dataset, providing direct evidence that the kernel selects biologically meaningful modes rather than providing generic regularisation.

Role of Positional Encoding

Without sinusoidal positional encoding, the SM kernel produced no noticeable improvement over the Gaussian. This is expected from the design: the SM kernel's biological frequency priors are intended to detect periodicity at specific sequence positions, but word vectors without positional information are exchangeable - the RFF map has no basis to distinguish periodic from non-periodic structure. Once positional encoding was

correctly implemented, the SM kernel became the best-performing configuration across all datasets. The result is a clean empirical validation of the design principle: the periodicity-detection mechanism works, but only when the input to the RFF map carries positional information.

Fusion and Feature Mode

Across all SM kernel experiments, late fusion consistently outperforms concatenation, and combining both the RFF-on-Doc2Vec and KME representations always gives the best result. For the anisotropic kernel the picture is less uniform - sometimes both representations help, sometimes only RFF-on-Doc2Vec - but late fusion is preferable to concatenation in all cases. Allowing each classifier to learn appropriate decision boundaries in its own feature space before combination is a more effective strategy than forcing joint learning over a merged representation, and the results bear this out consistently.

20 The code and scripts required to reproduce the results presented in this work are available at: https://github.com/Aishik60/kernel_PPI.

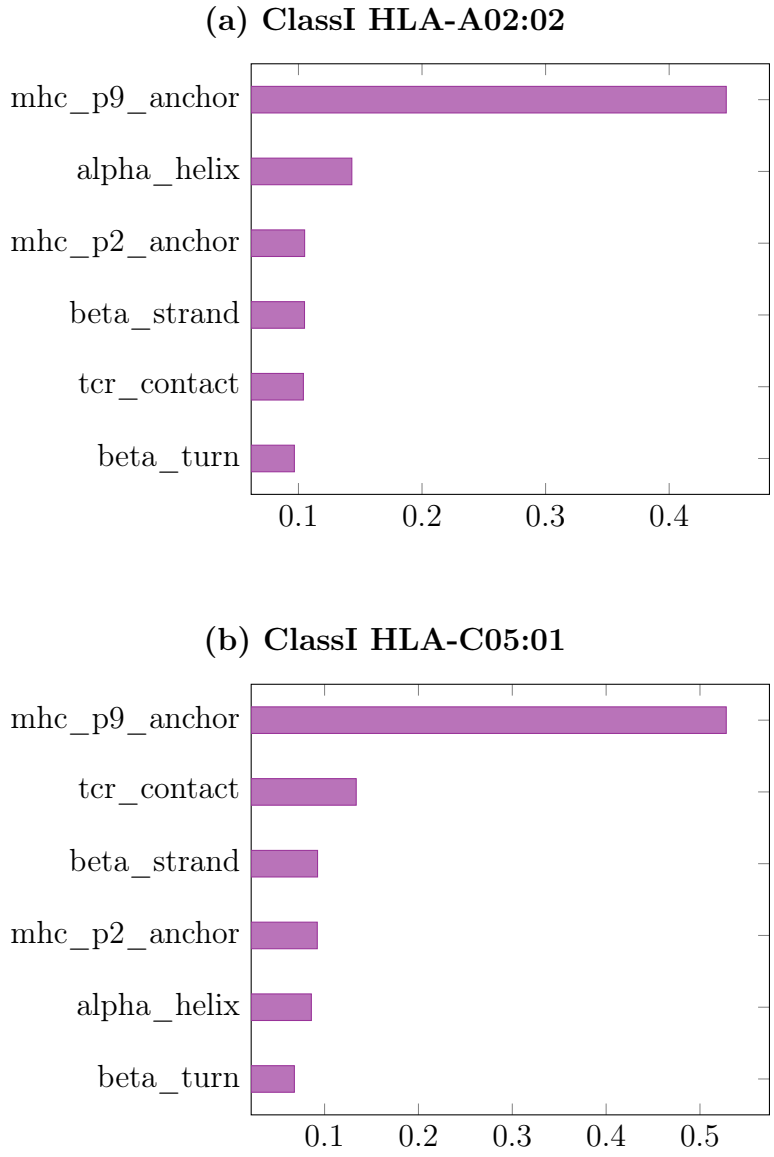


Figure 4.3: Learned SM kernel mixture weights for the two class I alleles (encoding-space representation). Bars are sorted by ascending weight; values sum to 1.0. In both alleles the MHC P9 anchor periodicity is overwhelmingly dominant, accounting for 44.6% of total weight in HLA-A02:02 and 52.8% in HLA-C05:01. This reflects the structural role of the C-terminal anchor residue at position 9 of the 9-mer peptide: it is the most constrained position in the closed class I groove, and the kernel correctly identifies its periodicity as the most discriminative signal for binding prediction. The secondary contribution of the alpha-helix periodicity (14.3% and 8.6% respectively) is consistent with the partial helical character of bound 9-mers in the groove. The remaining components — beta-strand, P2 anchor, TCR contact region, and beta-turn — carry roughly equal residual weight, indicating that no single secondary mode dominates after the P9 anchor is accounted for.

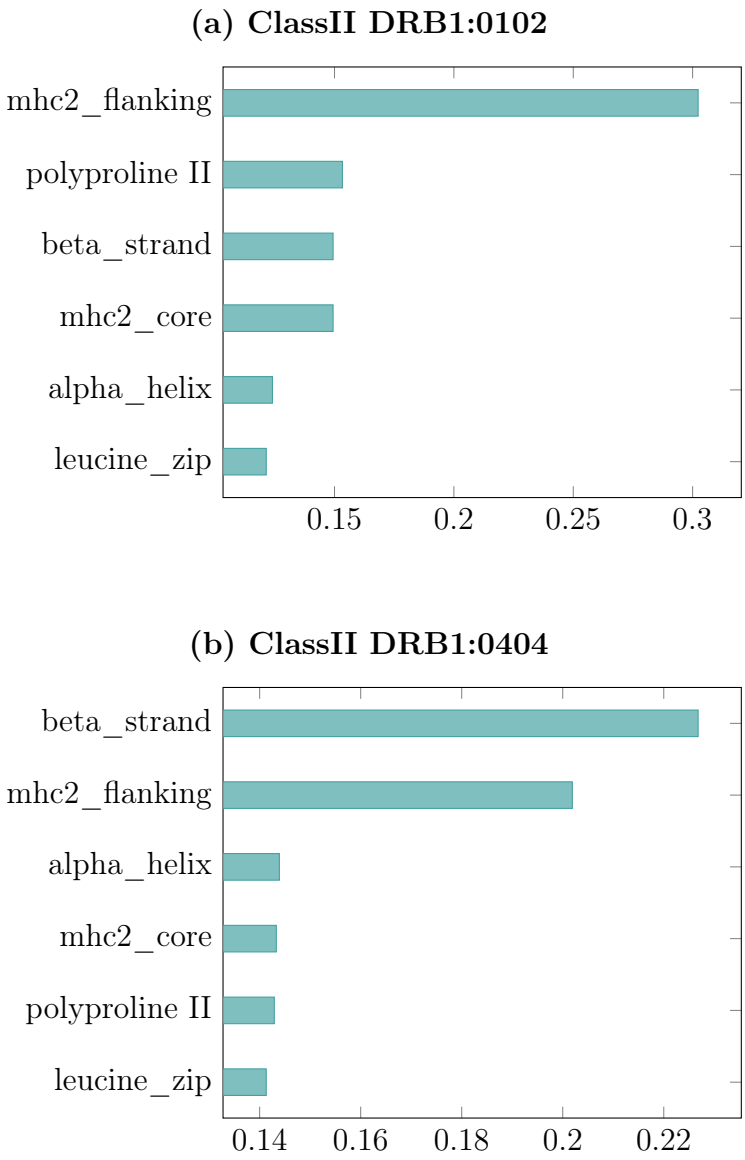


Figure 4.4: Learned SM kernel mixture weights for the two class II alleles (encoding-space representation). Bars are sorted by ascending weight; values sum to 1.0. The weight distributions are markedly more uniform than in the class I setting, with no single periodicity exceeding 30% in either allele. For DRB1:0102 the MHC2 flanking periodicity leads (30.2%), reflecting the open-ended class II groove where the flanking residues outside the core nonameric register contribute substantially to binding affinity. For DRB1:0404 the weight is distributed almost evenly across all six components, with beta-strand (22.7%) and MHC2 flanking (20.2%) marginally ahead. The polyproline II helix component, which captures the disordered flank geometry characteristic of class II-bound peptides, ranks second in DRB1:0102 (15.3%) and contributes comparably in DRB1:0404 (14.3%). The absence of a dominant anchor comparable to the P9 signal in class I is consistent with the shallower, more promiscuous binding groove of class II MHC molecules, where specificity arises from the cumulative effect of multiple moderate-affinity contacts rather than one dominant anchor constraint.

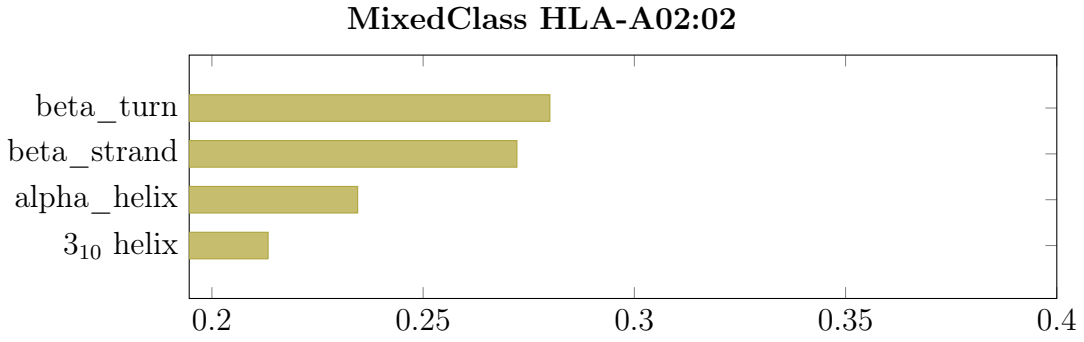


Figure 4.5: Learned SM kernel mixture weights for the mixed-class dataset (encoding-space representation). Bars are sorted by ascending weight; values sum to 1.0. In contrast to the class-specific settings, the mixed-class kernel converges to four generic secondary structure modes — beta-turn, beta-strand, alpha-helix, and the 3_{10} helix — with near-uniform weights ranging from 21.3% to 28.0%. Two aspects of this result are noteworthy. First, none of the allele-specific anchor periodicities (MHC P9, MHC P2, MHC2 flanking) that dominate in the class-specific settings appear among the selected components. This is consistent with the hypothesis that a single allele-specific anchor periodicity cannot dominate in a training set that pools class I and class II interactions, where the two binding classes impose structurally incompatible anchor constraints on the same embedding space. Second, the weight distribution is the most uniform across all five datasets ($\max w_q = 0.280$, compared to 0.528 for ClassI C05:01), indicating that the kernel relies on no single structural mode for discrimination. The high normalised gain of 0.798 achieved in this setting despite the absence of allele-specific priors demonstrates that generic secondary structure periodicity alone provides a sufficiently rich signal when the feature space is heterogeneous.

Chapter 5

Discussion

The results across all five datasets follow a consistent ordering: Gaussian and Laplacian kernels improve over the SWING baseline, the anisotropic kernel improves over both, and the SM kernel improves over everything. This hierarchy is not accidental — it reflects increasing structural specificity in how each method represents the interaction embedding space. Each step in the hierarchy solves a concrete limitation of the previous one, and understanding why illuminates both the strengths and the remaining weaknesses of the framework.

5.1. The Performance Hierarchy and What It Reflects

The Gaussian and Laplacian kernels establish that lifting Doc2Vec vectors into an RKHS is genuinely useful: non-linear structure in the interaction embedding space exists and is predictive. Their near-parity on most datasets reflects the fact that the pMHC binding landscape varies smoothly enough in sequence space that the difference in tail behaviour between the two kernels is not decisive. What both kernels share — and what drives their gain over the baseline — is the ability to define similarity through an inner product in a function space rather than a Euclidean distance, allowing the downstream classifier to exploit curvature in the embedding manifold that linear feature combinations cannot access.

The anisotropic kernel adds the insight that embedding dimensions are not equally informative. The SGD alignment procedure identifies and downweights dimensions that contribute noise rather than signal, and the classifier benefits from operating in a more discriminative subspace. However, the anisotropic kernel's near-null gain over the Gaussian on the mixed-class dataset is an informative negative result rather than a failure. The mixed-class training set pools class I and class II peptides, producing a heterogeneous embedding space where no single set of per-dimension length-scales can simultaneously

suppress irrelevant structure for both binding classes. The SGD alignment objective, being unsupervised, learns from the pooled inner-product structure and receives conflicting gradient signals from the two classes, converging to length-scales close to the isotropic baseline. This reveals a fundamental limitation of dimension-level reweighting in heterogeneous settings: the relevant "directions of informativeness" are class-dependent, and a single diagonal rescaling cannot capture both simultaneously.

The SM kernel resolves this limitation by operating at the level of spectral frequency components rather than individual dimensions. Different components can simultaneously respond to class I-specific anchor constraints and class II-specific groove periodicities, with the weight learning procedure determining their relative contribution from the data. The mixed-class normalised gain of 0.798 — compared to 0.654 for the anisotropic kernel — directly reflects this architectural advantage.

5.2. Biological Interpretability of the Learned Weights

A distinguishing property of the SM kernel is that the learned mixture weights $\{w_q\}$ constitute an interpretable biological readout: they quantify which structural periodicities in the interaction encoding are most predictive for a given binding context. This interpretability is not merely a post-hoc description — it is a direct consequence of fixing the component frequencies at known biophysical constants and letting only the weights vary.

For both class I alleles, the P9 anchor weight dominates (0.446 for HLA-A02:02, 0.528 for HLA-C05:01), consistent with the well-established role of the C-terminal anchor residue as the primary determinant of class I binding specificity. The 9-mer binding groove enforces a strict constraint at position 9, and the SM kernel correctly identifies this periodicity as the most predictive structural mode. The alpha helix component receives the next largest weight for HLA-A02:02 (0.143), reflecting the partial helical character of the peptide backbone within the closed class I groove.

For the class II alleles, the weight distribution is notably flatter. The flanking region periodicity (mhc2_flanking) receives the highest weight for DRB1:0102 (0.302) but only modest weight for DRB1:0404 (0.202, second to beta strand at 0.227). This is consistent with the known promiscuity of class II binding: the open-ended groove accommodates a wider range of peptide conformations, so no single structural mode dominates binding specificity across different peptides. The relatively uniform weight distribution for the mixed-class dataset (the largest weight is 0.280 for beta turn, with all four components within a 0.15 range) reflects the same logic at the dataset level: when class I and class II interactions are pooled, the model must balance class-specific structural modes, and the

result is a more distributed weight profile.

These patterns confirm that the SM kernel is doing interpretable biological work, not simply providing generic regularisation. The learned weights recover known structural biology of MHC binding and do so in a way that varies systematically across binding contexts.

5.3. The Role of the Normalised Gain Metric

The normalised gain metric proves important for interpreting results honestly, particularly on the class II datasets. Raw improvements of +0.030 and +0.029 for the SM kernel on DRB1:0102 and DRB1:0404 appear modest, but normalised gain reveals them as 39% and 43% headroom recovery — substantial progress in a regime where the baselines are already 0.923 and 0.933. At these baselines, the remaining gap to perfect prediction is small and hard-won: the binding landscape is already well-characterised by the polarity encoding, and further improvement requires resolving fine-grained structural distinctions that simpler methods cannot access. Reporting only raw AUROC differences would systematically understate the difficulty and significance of improvements on high-baseline datasets.

Chapter 6

Conclusion

One objective of this thesis has been to overcome one of the fundamental limitations of the SWING method; namely that the interaction vectors it uses are mapped into a simple Euclidean space in which the decision boundaries used for classification are not capable of capturing the often non-linear and highly discontinuous nature of biochemical binding specificity. The approach proposed here – mapping to kernel mean embeddings based on RF-based approximations of four bio-inspired kernels – significantly enhances performance on five separate data sets.

The key finding is that the quality of improvement scales with how much biological knowledge is encoded in the kernel. The Gaussian and Laplacian kernels, which are agnostic to the structure of the interaction space, provide a baseline gain from non-linearity alone. The anisotropic kernel, which learns which embedding dimensions are informative, adds a further layer of adaptivity. The Spectral Mixture kernel, which encodes specific secondary structure periodicities as hard priors and learns their relative importance from data, achieves the highest performance in every setting — recovering up to 80% of the available headroom on the mixed-class dataset and between 39% and 67% across the class I and class II alleles. The result that the SM kernel produces no improvement without sinusoidal positional encoding is itself informative: it confirms that the kernel's periodicity detection mechanism is doing real structural work, not simply providing generic regularisation.

Chapter 7

Future Work

Several directions follow naturally from this work.

Making the SM kernel's biological frequency priors learnable — treating the fixed structural periods as initialisations rather than constraints — would allow the model to adapt to allele families where the canonical secondary structure periods are not the dominant binding determinants.

Extending the full kernel pipeline to the missense mutation perturbation task is a direct next step. The Laplacian kernel, with its heavier spectral tails and non-smooth structure, is expected to show a clearer advantage over the Gaussian in this setting, since single-residue substitutions produce cliff-edge discontinuities in the interaction manifold that the Gaussian kernel's infinite smoothness cannot efficiently model.

Longer term, the interaction language framework that SWING establishes — encoding the biochemical relationship between two sequences as a primary object — is general enough to be applied to DNA–protein and RNA–protein interactions, and the kernel methods developed here transfer directly to those settings.

References

1. Siwek, J.C., Omelchenko, A.A., Chhibbar, P., Arshad, S., Rosengart, A., Nazarali, I., Patel, A., Nazarali, K., Rahimikollu, J., Tilstra, J.S., Shlomchik, M.J., Koes, D.R., Joglekar, A.V., & Das, J. (2025). Sliding Window Interaction Grammar (SWING): a generalized interaction language model for peptide and protein interactions. *Nature Methods*, 22, 1707–1719.

2. Dayhoff, M.O., Schwartz, R.M., & Orcutt, B.C. (1978). A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, Vol. 5, Suppl. 3 (pp. 345–352). National Biomedical Research Foundation.

3. Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems* (NeurIPS), 20, 1177–1184.

4. Wilson, A.G., & Adams, R.P. (2013). Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning* (ICML), 1067–1075.

5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (NeurIPS), 30, 5998–6008.

6. Le, Q.V., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning* (ICML), 1188–1196.

7. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

8. Bochner, S. (1933). Monotone Funktionen, Stieltjessche Integrale und harmonische Analyse. *Mathematische Annalen*, 108, 378–410.

9. Grantham, R. (1974). Amino acid difference formula to help explain protein evolution. *Science*, 185(4154), 862–864.

10. Miyazawa, S., & Jernigan, R.L. (1985). Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18(3), 534–552.

11. Nielsen, M., & Andreatta, M. (2016). NetMHCpan-3.0; improved induction of MHC class I binding motifs and expanded binding predictions for HLA-B and HLA-C loci. *Nucleic Acids Research*, 44(W1), W551–W558.

12. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., & Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637), 1123–1130.

13. Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2021). ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10), 7112–7127.

14. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2, 419–444.

15. Leslie, C., Eskin, E., & Noble, W.S. (2002). The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the Pacific Symposium on Biocomputing*, 564–575.