universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Exploration of 3D Pharmacophore Kernels and Multi-Instance Learning for Bioactivity Prediction

verfasst von | submitted by

Lukka Ladmann B.Sc.

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 910
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Computational Science
Betreut von | Supervisor:	Univ.-Prof. Mag. Dr. Thierry Langer

Acknowledgements

First and foremost, I would like to thank Univ.-Prof. Mag. Dr. Thierry Langer for motivating me to join his research group and for giving me the opportunity to contribute towards his research with this exciting project.

I am also sincerely grateful to my co-supervisor, Assoz. Prof. Dipl.-Inf. Dr. Nils Morten Kriege, whose extensive expertise of computer science—especially the mathematics behind graph theory and kernel methods—opened up new perspectives for me and greatly enriched my understanding of these fields.

Additionally, I wish to express my deep gratitude to Daniel Rose, M.Sc., for generously devoting his time, patiently answering my countless questions, and sharing his knowledge in cheminformatics, while also providing me with invaluable feedback on my code and manuscript. The guidance and influence of both Nils Kriege and Daniel Rose played a major role in shaping the direction of this thesis.

Moreover, I am deeply thankful to the people and city of Vienna, which became my second home over the course of my master's studies.

Finally, I need to express my heartfelt thanks to my friends and my brother for their support and honesty, and above all, to my parents, without whom I would have never been able to accomplish this work.

Abstract

This thesis investigates 3D pharmacophore-based strategies for bioactivity prediction by combining multi-instance learning (MIL) with a Pharmacophore Kernel (PK). The PK compares sets of three-point pharmacophores - including both label matching and distance-based similarities - across multiple conformers of each molecule. In comparison with established 2D descriptors (ECFP: Extended Connectivity Fingerprints) and 3D descriptors (RDF: Radial Distribution Function, TGT: Typed Graph Triangles) on the BACE dataset, PK can outperform traditional methods when sufficient conformers and training data are available. Furthermore, we investigate the performance of our models under data-scarce conditions and report a surprisingly robust performance of the TGT fingerprint. Our results highlight the importance of explicitly modeling molecular flexibility, although the use of the pharmacophore kernel comes with significant computational costs. Future work should focus on more diverse datasets and thorough conformer count selection. It should also extend the PK to more general subgraph matching kernels, further improving the training of machine learning models for bioactivity prediction using 3D pharmacophores.

Kurzfassung

In dieser Arbeit werden 3D-Pharmakophor-basierte Strategien zur Bioaktivitätsvorhersage untersucht, indem Multi-Instanz-Lernen (MIL) mit einem Pharmakophor-Kernel (PK) kombiniert wird. Der PK vergleicht Mengen an Drei-Punkt-Pharmakophoren – basierend auf Label-Matching und distanzbasierter Ähnlichkeit – über mehrere Konformere eines Moleküls hinweg. Auf dem BACE-Datensatz übertrifft PK etablierte 2D-Deskriptoren (ECFP) sowie 3D-Deskriptoren (RDF, TGT), sofern ausreichend Konformere und Trainingsdaten vorliegen. Zudem zeigt die Evaluierung unter datenarmen Bedingungen eine überraschend robuste Leistung des TGT-Fingerprints. Unsere Ergebnisse unterstreichen den Wert einer expliziten Modellierung molekularer Flexibilität, wobei die Nutzung des Pharmacophorkernels jedoch einen hohen Rechenaufwand erfordert. Zukünftige Arbeiten sollten vielfältigere Datensätze berücksichtigen und die Auswahl der Konformeranzahl optimieren. Zudem sollte der PK auf umfassendere Subgraph Matching Kernel ausgeweitet werden, um das Training von Machine-Learning-Modellen zur Vorhersage von Bioaktivität mithilfe von 3D-Pharmakophoren weiter zu verbessern.

Contents

List of Figures

List of Tables

Chapter 1

Introduction & Foundations

1.1 Pharmacophores

In the official IUPAC definition from 1998, a pharmacophore is described as "the ensemble of steric and electronic features that is necessary to ensure the optimal supramolecular interactions with a specific biological target structure and to trigger (or to block) its biological response"[1]. Rather than representing an actual molecule or functional group, it serves as an abstract concept encompassing the *common molecular interaction capacities* shared by a set of bioactive compounds. In this sense, a pharmacophore identifies the *largest common denominator* of biologically relevant features, distinguishing it from simpler structural functionalities or skeletal motifs that do not necessarily capture the essential inter- and intramolecular interactions underlying bioactivity[2].

By focusing on key chemical features (e.g., hydrogen-bond donors, acceptors, and aromatic rings) in three-dimensional space—often represented as spheres, planes, or vectors—pharmacophore models enable *scaffold hopping* (the replacement of a molecule's core structure with a novel chemical motif while preserving its biological activity[3]), leading to increased structural diversity in screening for new hits[4]. The goal of scaffold hopping is to replace the chemical core structure by a different chemical motif while keeping the biological activity of the molecule. As pharmacophores define chemical features essential for biological activity, they can be successfully employed to guide scaffold replacements. By emphasizing spatial arrangements rather than specific chemical structures, pharmacophores effectively highlight the essential interactions necessary for bioactivity. Key examples of feature types, their geometric representations, and typical interactions are provided in Table 1.1.

Table 1.1: Overview of common pharmacophoric feature types, their typical geometric representations, complementary features, and frequently observed interactions in biological systems[4].

Feature type	**Geom. representation**	**Complementary feature type(s)**	**Interaction type(s)**
Hydrogen-bond acceptor (HBA)	vector or sphere	HBD	hydrogen bonding
Hydrogen-bond donor (HBD)	vector or sphere	HBA	hydrogen bonding
Aromatic (AR)	plane or sphere	AR, PI	π-stacking, cation–π
Positive ionizable (PI)	sphere	AR, NI	ionic, cation–π
Negative ionizable (NI)	sphere	PI	ionic
Hydrophobic (H)	sphere	H	hydrophobic contact

1.1.1 Virtual Screening

Pharmacophore-based techniques currently are an integral part of many computer-aided drug design workflows, and have been successfully and extensively applied for virtual screening[4, 5]. Pharmacophore

models, derived via receptor- or ligand-based approaches, abstract essential non-bonded interactions between ligands and targets, making them both computationally efficient and intuitive for scientists[6]. One major application of these models is virtual screening of large compound libraries to discover novel compounds with desired pharmacophoric features crucial for biological activity[4, 7]. Virtual screening encompasses a series of *in silico* approaches designed to analyze large databases of compounds and identify promising hit candidates. These methods often complement high-throughput screening (HTS) campaigns by either preceding experimental assays, aiding in the selection of narrower candidate subsets, or following HTS to recover compounds that might have been overlooked ("latent hits"). Traditionally, virtual screening is partitioned into two main categories:

- **Ligand-based screening:** Relies on the 2D or 3D structures, molecular descriptors, or known active (and sometimes inactive) compounds to identify novel molecules of interest by applying similarity-based searches, common substructure detection, pharmacophore modeling, or shape-comparison techniques[8].
- **Structure-based (receptor-based) screening:** Involves, for example, computationally docking compounds into a target binding site and ranking them according to one or multiple scoring functions. The flexibilities of the target protein and ligand can be incorporated in varying degrees, depending on computational resources and the nature of the protein.

By appropriately managing the ligand and receptor conformational variability and refining the hits via post-processing steps, virtual screening can significantly accelerate the drug discovery pipeline while saving resources and offering more targeted experimentation compared to purely empirical HTS methods.

Pharmacophore-based screening yields structurally diverse hits, enables fast searching of large databases, and typically achieves higher hit rates compared to random sampling, enriching active compounds in smaller subsets. This computationally efficient method is widely used as a prefilter before applying resource-intensive techniques like MD simulations[9]. The approach also identifies diverse chemotypes not easily matched by traditional methods, offering valuable ideas for novel lead development and optimization[4].

1.2 From SVMs to Pharmacophore Kernels

Support Vector Machines (SVMs) are powerful supervised learning methods primarily used for classification but also applicable to regression, outlier detection, and related tasks. In their simplest linear form, SVMs learn a decision boundary, $\langle w, x\rangle + b = 0$, that maximizes the margin between two classes (labeled -1 and $+1$) while minimizing classification errors. For complex or nonvectorial data, SVMs employ the *kernel trick*, replacing inner products $\langle x_i, x_j\rangle$ with a kernel function $k(x_i, x_j)$. This enables SVMs to handle diverse data representations, including molecular structures[10].

Molecules are often represented as *molecular graphs*, where nodes correspond to non-hydrogen atoms and edges represent covalent bonds. Chemical Graph Theory (CGT) applies graph-theoretical methods to study molecular connectivity and structure, using graphs to model atoms and their bonds[11]. Formally, we define a *molecular graph* as:

$$\mathcal{G} = (V, E),$$

where

- V is the set of vertices (nodes) representing non-hydrogen atoms,
- E is the set of edges representing covalent bonds,
- $n = |V|$ is the number of vertices (atoms), and
- $m = |E|$ is the number of edges (bonds).

Molecular structures can be represented in two or three dimensions. In 2D, molecules are described by labeled graphs where vertices correspond to atoms (excluding hydrogen) and edges denote covalent bonds, or by various 2D descriptors[12]. In 3D, atoms are positioned in $\mathbb{R}^3$ with defined interatomic distances and angles. While directly converting these structures into fixed-length vectors can be challenging, kernel methods circumvent this by requiring only a valid similarity measure between molecules. If a kernel function satisfies positive semi-definiteness (p.s.d.), it can be integrated into an SVM framework without explicitly constructing feature vectors[13]. This flexibility makes SVMs particularly effective for molecular data analysis.

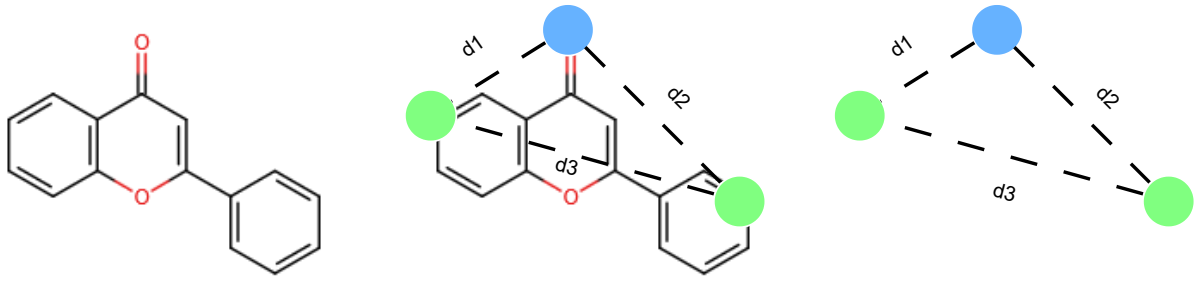

Figure 1.1: Visualization of triplets of atoms, showing three atoms forming a triangle with associated pairwise distances, adapted from[10].

1.2.1 Walk Kernels and Product Graphs

Molecules can be analyzed using *walk kernels* in 2D. A walk of length n on a labeled graph is a sequence of n adjacent vertices (possibly revisiting vertices or edges). The walk kernel counts such walks and sums over matches in their label sequences. Equivalently, one may form a *product graph* $\mathcal{G} \times \mathcal{G}'$, where

$$\mathcal{G} = (V, E) \quad \text{and} \quad \mathcal{G}' = (V', E')$$

are the molecular graphs of two molecules (with V and V' representing the sets of atoms, each with associated labels, and E and E' the sets of bonds). In this product graph, vertices are pairs (v, v') with $v \in V$ and $v' \in V'$ for which the atom labels match, and counting walks of length n corresponds to counting matching walks in the original graphs. In 3D, one may consider *complete* atom-based graphs, where every pair of atoms is connected by an edge encoding the Euclidean distance. This idea is key to the *pharmacophore kernel* described below.

1.2.2 Pharmacophore Kernel

In the pharmacophore kernel introduced by Mahé et al.[10], a *pharmacophore* is defined as a triplet of atoms arranged in 3D space to form a triangle with pairwise distances. This explicit definition (focusing on triplets with associated labels and distances) contrasts with the classical IUPAC notion of a pharmacophore as an abstract ensemble of steric and electronic features. Figure 1.1 illustrates how three atoms form such a triplet.

Let $\mathcal{M}$ denote a molecule represented as a set of atoms in $\mathbb{R}^3$, where each atom i is described by

$$(\mathbf{x}_i, \ell_i),$$

with $\mathbf{x}_i \in \mathbb{R}^3$ as its coordinate vector and ℓ_i its label (e.g., atom type, partial charge). A pharmacophore is then defined as

$$p = \big((\mathbf{x}_1, \ell_1),\, (\mathbf{x}_2, \ell_2),\, (\mathbf{x}_3, \ell_3)\big)$$

(with the corresponding pairwise distances d_1, d_2, d_3 implicitly defined). Define the set of all such triplets extracted from $\mathcal{M}$ as

$$\mathcal{P}(\mathcal{M}) = \{\, p \mid p \text{ is a triplet of distinct atoms in } \mathcal{M} \,\}.$$

To compare two molecules $\mathcal{M}$ and $\mathcal{M}'$, the pharmacophore kernel is given by

$$K(\mathcal{M}, \mathcal{M}') = \sum_{p \in \mathcal{P}(\mathcal{M})} \sum_{p' \in \mathcal{P}(\mathcal{M}')} K_P(p, p'), \tag{1.1}$$

where $K_P(p, p')$ measures the similarity between triplets p and p'.

We decompose $K_P(p, p')$ into an intrinsic similarity term K_I and a spatial similarity term K_S:

$$K_P(p, p') = K_I(p, p') \cdot K_S(p, p'), \tag{1.2}$$

with

$$p = \big((\mathbf{x}_1, \ell_1),\, (\mathbf{x}_2, \ell_2),\, (\mathbf{x}_3, \ell_3)\big) \quad \text{and} \quad p' = \big((\mathbf{x}'_1, \ell'_1),\, (\mathbf{x}'_2, \ell'_2),\, (\mathbf{x}'_3, \ell'_3)\big).$$

Intrinsic Similarity. The intrinsic similarity compares the atom labels:

$$K_I(p, p') = \prod_{i=1}^{3} K_{\text{Feat}}(\ell_i, \ell'_i), \tag{1.3}$$

where a common choice for K_{Feat} is the Dirac kernel:

$$K_{\text{Dirac}}(\ell, \ell') = \begin{cases} 1, & \text{if } \ell = \ell', \\ 0, & \text{otherwise.} \end{cases} \tag{1.4}$$

Spatial Similarity. The spatial similarity compares the pairwise distances between atoms:

$$K_S(p, p') = \prod_{i=1}^{3} K_{\text{Dist}}\Big(\|\mathbf{x}_i - \mathbf{x}_{i+1}\|,\, \|\mathbf{x}'_i - \mathbf{x}'_{i+1}\|\Big), \tag{1.5}$$

where $\|\mathbf{x}_i - \mathbf{x}_{i+1}\|$ denotes the Euclidean distance (with $i + 1$ taken modulo 3). A common choice for K_{Dist} is the Gaussian RBF kernel:

$$K_{\text{RBF}}(\mathbf{r}, \mathbf{s}) = \exp\Big(-\frac{\|\mathbf{r} - \mathbf{s}\|^2}{2\sigma^2}\Big), \tag{1.6}$$

with $\sigma > 0$ controlling sensitivity to distance differences.

Because both K_{Dirac} and K_{RBF} are positive definite, their product yields a valid kernel for comparing pharmacophores.

The direct computation of $K(\mathcal{M}, \mathcal{M}')$ is expensive—its complexity is

$$O\Big((|\mathcal{M}| \cdot |\mathcal{M}'|)^3\Big),$$

where $|\mathcal{M}|$ and $|\mathcal{M}'|$ are the numbers of atoms in the respective molecules. To reduce this cost, Mahé et al.[10] propose a matrix-based method: a square matrix $\mathbf{M}$ of size $|\mathcal{M}| \cdot |\mathcal{M}'|$ is constructed, with each entry encoding the product of intrinsic and spatial similarities between atoms. The pharmacophore kernel is then computed as $\text{trace}(\mathbf{M}^3)$, effectively counting closed walks of length 3 while reducing redundant comparisons. Efficiency is further enhanced by restricting $\mathbf{M}$ to its nonzero entries (i.e., when atom labels match).

Finally, note that this pharmacophore kernel is essentially a 3D extension of walk-count graph kernels. By treating each molecule as a complete, atom-based graph—where every pair of atoms is connected by an edge labeled with the Euclidean distance—a triplet of atoms corresponds exactly to a self-returning walk of length 3, making the kernel formulation equivalent to counting such walks[14].

1.3 Implementation and Limitations of the Pharmacophore Kernel

ChemCpp[15] is a C++ library that provides classes and tools for managing molecular data in MDL (MOL/SDF) formats. It provides (i) file manipulation utilities and (ii) kernel tools for various graph-based approaches. Among these, the pharmacophore kernel[10] is implemented, along with marginalized graph kernels[16], their extensions[17], Tanimoto kernels[18], and tree-pattern-based graph kernels[19].

Holder[20] identifies two major limitations of that pharmacophore kernel implementation:

1. **Computational complexity:** Exhaustive comparison of all three-point atomic pharmacophores is computationally costly. We address this issue by using three-point pharmacophores derived from pharmacophore features rather than raw atomic data. Abstracting molecules to the pharmacophore level reduces the number of vertices, thereby lowering computational complexity. Further efficiency gains are achieved by exploiting sparse adjacency matrices via the Triangular Distance Kernel and applying row-wise parallelization to the Gram matrix computation.

2. **Conformational flexibility:** Structural flexibility can be a major source of unreliability in similarity assessments, as freely rotatable bonds can alter interatomic distances by several angstroms. While including multiple conformations per molecule can better capture structural flexibility, a critical challenge arises: only a specific conformer might mediate the observed binding activity. Simply adding all conformers of an active compound as separate active instances in the dataset introduces bias, as inactive or irrelevant conformations would be incorrectly labeled as active. Instead, ligands should be represented as bags of conformations, where the entire bag (molecule) carries a single activity label. This requires proper aggregation strategies to combine conformational contributions meaningfully.

1.4 Conformational Flexibility in Molecular Systems

Molecules in biological systems exhibit inherent conformational flexibility, adopting multiple shapes that influence their interactions and functions[21]. Generating representative conformer ensembles is challenging due to the high dimensionality of conformational space and computational costs.

In computational drug design and 3D virtual screening, algorithms must capture relevant molecular conformations when binding to biomacromolecules. Tools like pharmacophore-based and shape-focused screening engines rely on precomputed multiconformational databases, where compounds are represented by compact conformer ensembles[22]. Small molecules, with numerous conformational degrees of freedom, may adopt distinct binding poses, including strained states related to transition states. Generating accurate conformer ensembles remains a key challenge[22].

This work employs confgen from CDPKit[23], which has been published under the name CONFORGE[24], to generate high-quality conformer ensembles, ensuring a balance between computational efficiency and molecular diversity.

1.5 Multi-Instance Learning and Drug Activity Prediction

Multi-instance learning (MIL)[25] is well-suited for drug activity prediction, where molecules are represented as "bags" of conformers (instances). This approach is particularly relevant in drug activity prediction, where molecules are represented by multiple conformations resulting from bond rotations. A molecule is active if at least one conformer binds to the target; otherwise, it is inactive. MIL accounts for molecular flexibility, improving pharmacophore-based bioactivity predictions.

Gärtner et al.[26] introduced MIL kernels, which aggregate similarity scores between instances within and across bags, enabling Support Vector Machines (SVMs) to classify structured molecular data. Vishwanathan et al.[27] further refined this approach with the Most Optimal Assignment Kernel (MOAK), approximating optimal assignment while preserving p.s.d., making it well-suited for handling multiple molecular conformations.

We extend these concepts by integrating pharmacophore kernels into an MIL framework, treating conformations as instances and aggregating their similarities.

1.6 Aim of the Thesis

The primary goal of this thesis is to explore strategies for bioactivity prediction by incorporating 3D pharmacophore features into a kernel-based machine learning framework. Conventional approaches often fail to capture the spatial flexibility and stereoelectronic nuances of pharmacophores. In response, we employ a pharmacophore kernel that uses feature data generated by CDPKit, emphasizing chemical and geometric information essential for molecular interactions. Conformational flexibility is addressed through a multi-instance learning strategy, aggregating information from multiple conformers by combining them to enhance predictive accuracy.

The **main contributions** of this thesis can be summarized as follows:

- **Pharmacophore Kernel extension:**
  Instead of relying on raw atomic data, this thesis incorporates domain-specific pharmacophore features (e.g., hydrogen-bond donors/acceptors, hydrophobic sites, etc.) into a kernel representation. By employing a triangular distance-based similarity, the method captures crucial 3D interactions while reducing computational cost and maintaining predictive performance.
- **Integration of multi-instance learning:**
  A range of approaches, including `mean`, `max`, and `softmax` aggregation, as well as specialized kernels such as the Multi-Instance Kernel `MI` and the Most Optimal Assignment Kernel (`MOAK`), is used to handle multiple conformers in a single molecule. This framework enhances similarity assessments for flexible structures by focusing on relevant conformations or combining their contributions.
- **Empirical evaluation:**
  The methods are demonstrated and benchmarked on the BACE dataset.

Chapter 2

Methods

2.1 Pharmacophore Kernel (PK) Computation

The PK computation begins by iterating through pairs of conformers from two conformer ensembles—collections of molecular conformations representing different spatial arrangements of the same molecule arising from rotatable bonds. In this approach, a pharmacophore is treated according to the IUPAC definition as an ensemble of steric and electronic features responsible for biological activity, rather than the triplet-of-atoms definition used in Mahé et al.'s work[10]. The pairwise Euclidean distances between these matched features are computed. With these distances in hand, the algorithm constructs a fused (product) graph by initializing a square matrix, where each entry corresponds to a pair of matched features.

It computes an edge weight by combining:

- a feature-type matching kernel (Dirac kernel, K_{Dirac}), and
- a distance-based kernel (Triangular kernel, $K_{\text{Triangular}}$).

This fused adjacency matrix is then converted into a sparse representation using the Compressed Sparse Row (CSR) format to optimize memory usage and computational efficiency. The resulting weighted adjacency matrix encodes both the three-dimensional distance relationships and the feature-type constraints, thereby capturing essential chemical and geometric similarities between the two conformers.

To capture higher-order relationships among matched pharmacophore features, the algorithm raises the weighted adjacency matrix $\mathbf{M}$ to the power of three and then computes its trace. Mathematically, this is expressed as

$$K(m, m') = \operatorname{trace}\big(\mathbf{M}^3\big), \tag{2.1}$$

where $K(m, m')$ denotes the kernel value between two conformers m and m' and the trace is defined as the sum of the diagonal elements of a matrix[28]. The trace of $\mathbf{M}^3$ effectively counts the number of closed walks of length three in the product graph, thereby providing a global measure of similarity based on third-order interactions.

Next, the resulting kernel values are normalized to ensure comparability across different pharmacophore conformers. Following the scheme proposed by Perret et al.[29], the normalized kernel matrix $\widetilde{\mathbf{K}}$ is given by

$$\widetilde{\mathbf{K}}[i, j] \;=\; \frac{K[i, j]}{\sqrt{K[i, i]\, K[j, j]}}. \tag{2.2}$$

Here, $K[i, j]$ is the kernel value between conformers i and j, and $K[i, i]$ (respectively $K[j, j]$) denotes the self-similarity of conformer i (respectively j). Dividing by the geometric mean of the self-similarities places the values on a common scale, effectively producing a normalized kernel matrix that is consistent and comparable across all pairs of pharmacophore conformers.

2.2 Handling Conformational Flexibility

Flexible molecules pose additional challenges due to the variability in pharmacophore distances caused by rotational freedom of rotatable bonds. Conformations of the same molecule may differ significantly.

Including multiple conformers per molecule or pharmacophore can improve coverage but might also introduce bias if not all conformers reflect observed bioactivity.

In this thesis, we adopt Multiple Instance Learning (MIL), where each molecule is treated as a "bag" of conformers (instances). To compare two molecules, we compute pairwise similarity scores for each conformer in one molecule/pharmacohore against each conformer in the other, then apply one of several aggregation strategies:

- Mean aggregation: Averages all pairwise similarity scores.
- Maximum aggregation: Focuses on the most similar conformers.
- Softmax aggregation: Computes a weighted average, assigning higher weights to larger scores via the softmax function.
- Multi-Instance kernel (MI)[26]: Sums pairwise similarities across all conformations (instances) in both bags.
- Most Optimal Assignment Kernel (MOAK)[27]: Uses exponential weights in the logarithmic semiring to combine multiple matchings while maintaining p.s.d.

2.2.1 Multi-Instance Kernel (MI)

The Multi-Instance Kernel was introduced by Gärtner et al.[26] and is designed for data represented as bags of instances rather than single feature vectors. In our setting, each molecule (or pharmacophore) is treated as a bag, where each conformer is an instance. Biological activity may depend on one or more specific conformers. Summing the pairwise similarities across all conformers in two bags can capture a broader range of potentially active geometries.

Formally, the MI kernel is defined as

$$k_{\mathrm{MI}}(\mathcal{X}, \mathcal{Y}) \;=\; \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}} \Big(k_I(\mathbf{x}, \mathbf{y})\Big)^p, \tag{2.3}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are bags (sets) of conformers, $\mathbf{x}$ and $\mathbf{y}$ denote individual conformers, $k_I(\mathbf{x}, \mathbf{y})$ is the kernel measuring similarity between individual conformers, and p is a positive integer that controls how strongly the kernel focuses on highly similar conformer pairs. By aggregating pairwise similarities in this manner, the MI kernel inherently considers conformational diversity when comparing molecules or pharmacophores.

2.2.2 Most Optimal Assignment Kernel (MOAK)

The Most Optimal Assignment Kernel extends kernel-based similarity to structured objects, including sets of molecular conformers, while guaranteeing positive semi-definiteness (p.s.d.). Building on R-convolution kernels in the logarithmic semiring with a temperature parameter $\beta > 0$, MOAK can approximate the optimal matching of conformers between two molecules without losing p.s.d.[27].

The MOAK kernel is defined as:

$$k(x, x') \;:=\; \sum_{\tilde{x}\in R^{-1}(x)} \sum_{\tilde{x}'\in R^{-1}(x')} \exp\Big(\beta\, \kappa(\tilde{x}, \tilde{x}')\Big), \tag{2.4}$$

where $R^{-1}(x)$ and $R^{-1}(x')$ represent the sets of valid decompositions of x and x' (e.g., different conformers or substructures), κ is a local similarity measure, and β balances the emphasis on high-similarity matches, ensuring that the kernel emphasizes the most relevant contributions.

Fröhlich et al.[30] propose an *optimal assignment (OA) kernel* for comparing chemical molecules by finding a one-to-one matching of atoms that maximizes local similarities. Although it can incorporate neighborhood- and distance-based comparisons in polynomial time, the OA kernel is *not* guaranteed to be p.s.d.[31] In contrast, MOAK remains p.s.d. for finite β, due to its use of the logarithmic semiring and exponential transformation[27]. At high values of β, MOAK approximates the OA kernel by emphasizing the single best conformer-to-conformer mapping[31].

The MI and MOAK kernels enable support vector machines (SVMs) and other kernel methods to handle multi-instance datasets, such as molecules or pharmacophores with multiple conformers. By systematically aggregating pairwise similarities, these kernels focus on the conformers most relevant to biological activity. Their ability to efficiently compare flexible molecules while maintaining p.s.d.—a key requirement for kernel-based methods—makes them particularly relevant to our work.

2.3 PK Extension and Triangular Kernel Integration

We extend the PK framework by incorporating pharmacophore features instead of atom features, integrating molecular conformations, and evaluating different conformer aggregation approaches to enhance virtual screening. Unlike the original work by Mahé et al.[10], which focused on individual atoms, this thesis employs pharmacophore features generated by CDPKit[23]—including hydrogen bond donors, acceptors, hydrophobic regions, aromatic rings, positive and negative ionizable sites, and halogen bond donors and acceptors. Emphasizing the structural and spatial arrangements of these features aims for more accurate bioactivity prediction.

To improve computational efficiency and scalability, this thesis adopts a triangular kernel function for spatial similarity computations, replacing the Gaussian radial basis function (RBF) kernel traditionally used for pairwise distance comparisons. The Triangular Kernel, $K_{\text{Triangular}}$, compares distances by assessing their proximity within a tolerance parameter C and sets similarity to zero when the distance difference exceeds C:

$$K_{\text{Triangular}}(d, d') = \begin{cases} 1 - \dfrac{|d - d'|}{C} & \text{if } |d - d'| \leq C, \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

Here, C is the tolerance parameter, and d and d' denote the pairwise distances being compared.

The Triangular Kernel is compactly supported, meaning that beyond a certain distance threshold, its similarity value is zero. By contrast, the Gaussian RBF kernel is globally supported and never truly drops to zero. According to Kriege[32], exploiting this compact support leads to sparser product graphs and thus significantly reduces the runtime when computing graph-based kernels. This is because the resulting matrices contain many zero entries, which, when combined with data structures and algorithms designed for sparse matrices, enable faster computations. In addition, classification accuracy remains comparable or even improves on multiple datasets. Empirical evaluations (see Table 2.1) confirm that the Triangular Kernel yields similar or slightly better performance and substantially reduces computational overhead.

Parallelized computations further enhance scalability, enabling efficient handling of large datasets with extensive conformer ensembles. However, this approach requires access to increased computational resources (e.g., additional CPUs or GPU nodes) to fully realize its benefits.

Table 2.1: Performance metrics (mean ± std, across 4 random states) for PK RBF and PK triangular.

Metric	PK RBF	PK triangular
Accuracy	0.79 ± 0.01	0.80 ± 0.01
CV accuracy	0.81 ± 0.01	0.81 ± 0.01
F1	0.77 ± 0.02	0.78 ± 0.02
ROC AUC	0.79 ± 0.01	0.80 ± 0.01
MCC	0.58 ± 0.03	0.59 ± 0.03

2.3.1 Gram Matrix

In general, kernels are functions that take two objects (e.g., data points, vectors, or pharmacophores) as input and return a scalar value that quantifies the similarity between the objects. For a dataset, the pairwise kernel values for all data points are compiled into a matrix called the *kernel matrix* or *Gram matrix*[33].

Let $\mathcal{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$ represent a dataset of m points, each being an n-dimensional vector, i.e., $\mathbf{x}^{(i)} \in \mathbb{R}^n$. A kernel K maps pairs of points to a real-valued similarity measure

$$K\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\big) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}. \tag{2.6}$$

The Gram matrix $\mathbf{G}$ is an $m \times m$ symmetric and positive semi-definite matrix, defined as

$$\mathbf{G}_{i,j} = K\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\big), \tag{2.7}$$

where each entry $\mathbf{G}_{i,j}$ represents the kernel value for the corresponding data points $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$[34].

2.3.2 Gram Matrix Computation for Molecular Conformers

In our implementation of the Gram matrix computation, each entry represents the kernel value computed between the conformers of two molecules or pharmacophores being compared. Specifically:

- If each molecule has a single conformer, the matrix entry $\mathbf{G}_{i,j}$ is the kernel value between the conformer of the i-th molecule and the conformer of the j-th molecule.
- For molecules with multiple conformers, the Gram matrix entry becomes a *list* (or array) of kernel values. Each element in the list corresponds to the kernel value between one conformer of the first molecule and each conformer of the second molecule. For example, if the first molecule has n conformers and the second has m conformers (where n and m may differ, as molecules can exhibit varying numbers of conformers), the entry $\mathbf{G}_{i,j}$ will be an array of size $n \times m$, representing all pairwise comparisons between the conformers.

2.4 Fingerprint Representation

Molecular descriptors are numerical indices that encode a molecule's structure and physicochemical properties, serving as key variables in chemoinformatics and modeling[35]. They range from simple atom or fragment counts (e.g., molecular weight, hydrogen bond donors/acceptors) to complex theoretical calculations based on quantum chemistry or graph theory. In the field of Quantitative Structure–Activity Relationship (QSAR) modeling these descriptors link molecular structure to biological or physicochemical behavior, enabling predictions for untested compounds using known activities of similar molecules[36, 35].

While molecules are often represented as 2D or 3D structures, statistical and machine learning methods require finite-dimensional vector inputs. Molecular descriptors provide these numeric representations, capturing chemical information such as elemental composition, steric, electronic, or topological properties. This enables models to predict biological activity, toxicity, and pharmacokinetics. Careful selection of descriptors is crucial to streamline predictions and reduce complexity in drug discovery[13].

Descriptors are categorized based on their representation, with 2D (topological) and 3D (geometrical) descriptors being particularly relevant. 2D descriptors, derived from molecular graphs, encode atom connectivity and bond types, ignoring spatial metrics like bond angles. They include topological indices reflecting connectivity or structural complexity[35]. In contrast, 3D descriptors, derived from spatial (x, y, z) coordinates, capture molecular shape, size, and steric features, providing insights into conformational influences on activity or properties[35].

Pharmacophore-based fingerprints differ from molecular fingerprints by focusing on combinations of pharmacophoric features (e.g., hydrogen bond donors, acceptors) rather than atom-level details. Both molecular and pharmacophore-based descriptors encode structural and functional attributes into compact vectors, facilitating data-driven modeling for bioactivity prediction.

2.4.1 Extended Connectivity Fingerprints (ECFP)

Circular fingerprints are representations of chemical structures by atom neighborhoods and have been widely applied in QSAR analysis. A widely used class of circular fingerprints are ECFPs, based on the Morgan algorithm. Heavy atoms (i.e., non-hydrogen atoms) are encoded into multiple circular layers, with each layer extending up to a specified diameter[12].

ECFP fingerprints encode molecular structures into fixed-length bit vectors by recursively enumerating atomic neighborhoods within a specified radius, effectively capturing relevant 2D substructural features[37]. In this work, we employ ECFP4, which uses a radius of 2 bond steps, to capture the local environments around each atom and generate descriptors suitable for subsequent modeling tasks. ECFP4 fingerprints were used as the baseline molecular representation for machine learning model training and bioactivity prediction. This fingerprint was implemented for molecules with the `CircularFingerprintGenerator` from the CDPKit `CDPL.Descr` class[23].

While ECFPs capture 2D substructural features, Radial Distribution Function (RDF) descriptors provide a complementary approach by encoding 3D spatial information.

2.4.2 Radial Distribution Function (RDF)

Radial Distribution Function descriptors are based on the distance distribution in the geometrical representation of a molecule[35]. These descriptors represent molecular structure as a radial scattering curve

derived from interatomic distance spectra, which can be calculated for molecules of any complexity provided their Cartesian coordinates are known[38]. RDF fingerprints encode the spatial arrangement of atoms or pharmacophore features. They analyze pairwise distances, weight them based on specific criteria, and bin them into intervals to produce a continuous distribution. This approach captures the 3D structural information of molecules in a quantitative and interpretable manner, making RDF descriptors highly useful for molecular similarity analysis, virtual screening, and machine learning applications. The RDF descriptors for molecules were generated using the `MoleculeRDFDescriptorCalculator`, while those for pharmacophores were created using the `PharmacophoreRDFDescriptorCalculator` from the `CDPL.Descr` class in the CDPKit. The `PharmacophoreRDFDescriptorCalculator` computes RDF descriptors from pharmacophore features to capture molecular properties in 3D space. It analyzes pairwise distances between features, such as hydrophobic regions, aromatic rings, hydrogen bond donors, and hydrogen bond acceptors, with higher weights assigned to pairs of the same feature type. The RDF calculation is parameterized by a smoothing factor, scaling factor, start radius, radius increment, and step count, which collectively control the resolution and shape of the descriptor. For each feature type, distances are binned, and their weighted contributions are summed to generate the RDF fingerprint[39].

2.4.3 Typed Graph Triangles (TGT)

The *Typed Graph Triangle* descriptor encodes sets of three pharmacophore features (triangles) by their types and 3D inter-feature distances. Its reference implementation is found in the CDPKit `CDPL.Descr` class in `N-Point3DPharmacophoreFingerprintGenerator`[40].

1. **Extract features:** Identify all pharmacophore features and their 3D coordinates.
2. **Subset enumeration:** Enumerate subsets of size 3 of these features.
3. **Distance binning:** For each pair within the subset, compute the Euclidean distance and map it to a discrete bin.
4. **Canonical ordering:** If multiple features share the same type, reorder them consistently to ensure uniqueness.
5. **Hashing to bit index:**
   - Combine the binned distances and feature types.
   - Hash the result using the SHA-1 algorithm, which generates a unique 160-bit hash value for each input.
   - Convert the hash to a fingerprint bit index (by modulus).
   - Set that bit in the output bitset.

Repeating the above for all feature triplets yields a sparse bit vector (size m), where each set bit encodes one unique *typed graph triangle* in 3D space.

2.5 BACE Dataset

The BACE dataset, obtained from MoleculeNet[41], is a comprehensive resource designed for bioactivity prediction, specifically targeting human β-secretase 1 (BACE-1) inhibitors. It includes both quantitative (IC50 values) and qualitative (binary labels) binding results derived from experimental values reported in scientific literature over the past decade. The compounds were divided into binary classes—active ($IC_{50} \leq 100$ nM) and inactive—using a self-imposed cutoff for the experimental IC_{50}[42]. This collection comprises 1513 compounds, represented as SMILES strings with 2D structures, making it suitable for classification tasks. Among these, 691 compounds are labeled as *actives* and 822 as *inactives*. Initially, the data is split into *actives* and *inactives* files, which are processed independently through the steps described in 2.6.

2.6 Data Processing and Preparation

To prepare the BACE dataset for analysis, a series of preprocessing, conformer generation, and pharmacophore generation steps were performed. These steps are crucial for ensuring the quality and consistency of the data, enabling accurate bioactivity prediction.

- **Preprocessing:** Molecules were standardized, charges minimized, and counter ions and solvents removed. Protonation and deprotonation adjusted molecules to physiological pH, while duplicate entries were eliminated to avoid redundancy.
- **Conformer generation:** High-quality 3D conformers were generated using `confgen` from CDPKit with the CONFORGE[24] algorithm. This process produced ensembles of varying sizes (e.g., 1, 2, 4, 5, 8, 12, 20, 25, 50, 100) to evaluate the impact of conformational diversity. While the algorithm aims to produce the specified number of conformers per molecule, final counts may vary slightly.
- **Pharmacophore generation:** Pharmacophores were created from the generated conformers using `psdcreate` from CDPKit. They were stored in pharmacophore-screening databases (`.psd`) for downstream analysis.

The final dataset and descriptors form the basis for model training and evaluation (see Appendix A for details on hyperparameters and performance metrics).

Chapter 3

Results & Discussion

3.1 Overview of the Evaluation Workflow

The evaluation begins by establishing a 2D baseline with Extended Connectivity Fingerprints (ECFP), focusing on single-conformer molecules. This benchmark highlights the competitive performance of a well-established fingerprint. Next, we introduce 3D descriptors using single-conformer models. For molecules, we use molecular RDF fingerprints (RDF Mol); for pharmacophores, we use pharmacophore RDF fingerprints (RDF Pharm), Typed Graph Triangles (TGT), and the Pharmacophore Kernel (PK). These models incorporate spatial and stereoelectronic features, allowing us to compare their predictive accuracy to the baseline.

Building on these single-conformer insights, we then consider multiple conformers per molecule in RDF Mol to capture a broader range of 3D structural flexibility. Extending this multi-conformer approach to RDF Pharm enables a direct comparison of molecule-based versus pharmacophore-based representations under increased conformational diversity. Subsequently, TGT is introduced as a pharmacophore-focused fingerprint that enumerates typed triangles of pharmacophoric features. Finally, the PK is presented to investigate a more extensive kernel-based comparison of three-point pharmacophore features.

In this way, we quantify the impact of (1) going beyond 2D descriptors, (2) adding multiple conformers, and (3) utilizing pharmacophore-focused kernel-based methods on the predictive performance of bioactivity classification. Throughout this step-by-step progression, we focus on the Matthews correlation coefficient (MCC) for brevity; however, other metrics (Accuracy, F1, and ROC-AUC) were evaluated as well. Detailed numerical results, including those additional metrics, are provided in the appendix C.1.

3.2 Single-Conformer Comparison Among Models

This section compares the performance of all five models—ECFP, RDF Mol, RDF Pharm, TGT, and PK—when using only a single conformer per molecule or pharmacophore.

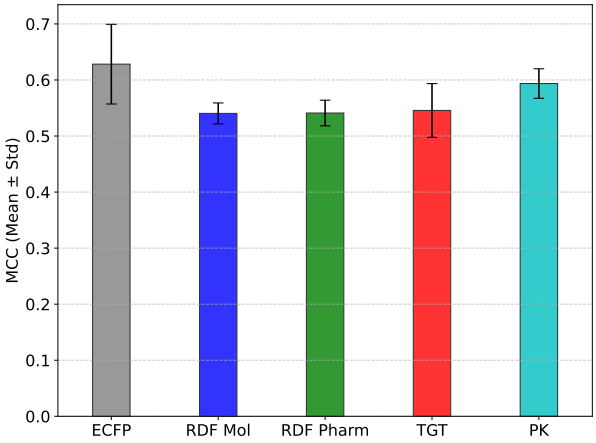

Figure 3.1: Comparison of MCC (*mean* ± *std*) across five models using only one conformer. The highest performance was achieved by the ECFP fingerprint, followed by the PK.

From Figure 3.1, it is clear that the ECFP descriptor consistently achieves a higher MCC than the other approaches. This finding aligns with prior work suggesting that 2D fingerprint-based methods can outperform 3D shape-based alternatives in certain contexts[43]. Nevertheless, the PK method follows closely, indicating that a carefully designed 3D kernel might have the potential to approach the strong performance of ECFP. The RDF Mol, RDF Pharm, and TGT methods exhibit slightly lower MCC values, emphasizing that the specific representation and kernel design can substantially impact classification accuracy.

Overall, these single-conformer results highlight both the reliability of ECFP and the potential of 3D kernel-based methods to narrow the gap in performance. In the subsequent sections, we explore whether using multiple conformers—and hence capturing more extensive 3D structural information—can further increase performance.

3.3 Multiple Conformers Comparison

3.3.1 RDF Mol vs. RDF Pharm

The inclusion of multiple conformers for RDF descriptors on full molecular structures enhances MCC compared to single-conformer models, though performance remains below the ECFP baseline (MCC = 0.63). Aggregation methods such as Multi-Instance Kernel (`MI`), Most Optimal Assignment Kernel (`MOAK`), and `mean` achieve moderate gains, with `MI` peaking at 0.60 for 16 conformers. Notably, the `max` aggregator demonstrates significantly weaker performance, likely due to its failure to ensure a positive semi-definite (p.s.d.) kernel matrix – a critical requirement for kernel-based methods. Despite these improvements, RDF descriptors are not yet able to match ECFP in predictive performance, suggesting their structural and chemical feature encoding require further optimization to bridge the gap with ECFP's bioactivity modeling.

Transitioning to pharmacophore-level RDF descriptors yields slight improvements over molecular RDF fingerprints for most aggregations. As shown in Figure 3.2, pharmacophore-based RDF approaches and molecule-based ones both reach their highest MCC values at 16 and 25 conformers.

Figure 3.3 highlights that `mean`, `MOAK`, and `MI` consistently achieve higher MCC scores than `softmax` or `max`, underscoring the value of multi-instance aggregation strategies. For all aggregations, except `MI`, pharmacophore RDF outperforms molecular RDF in the prediction on the BACE dataset, suggesting that pharmacophoric features may generally provide more accurate representations than atomic features in this setting. However, the gains remain modest, with pharmacophore RDF only marginally outperforming molecular RDF in most cases. The `max` aggregator continues to underperform for both representations. Overall, while pharmacophore-level RDF offers incremental improvements, the results emphasize the need for further 3D representations, to better utilize conformational diversity.

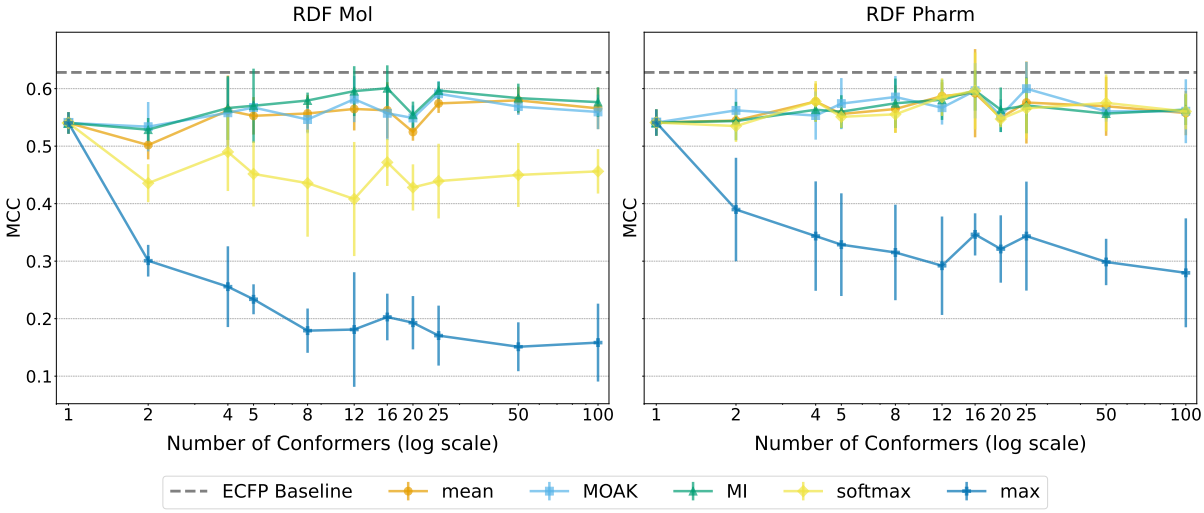

Figure 3.2: Side-by-side MCC comparison for RDF on molecules vs. RDF on pharmacophores across various conformer counts. Pharmacophore-level RDF slightly improves with more conformers but still lags behind the ECFP baseline in most cases.

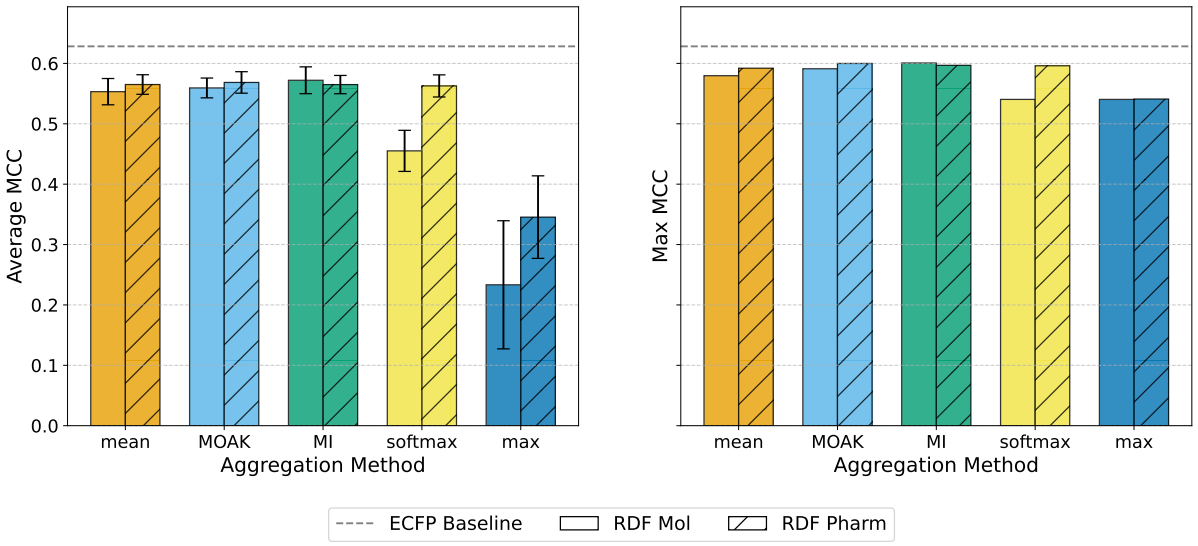

Figure 3.3: Average and maximum MCC values for each aggregation scheme comparing RDF on molecules vs. RDF on pharmacophores. The average and maximum values were calculated for each aggregation method across all available conformer counts.

3.3.2 RDF Pharm vs. TGT

TGT fingerprints encode triplets of pharmacophore features and thus make use of 3D information more effectively than pairwise RDF. Figure 3.4 shows TGT benefiting more from additional conformers—particularly with `MOAK` and `MI` aggregators—occasionally matching or surpassing the ECFP baseline (e.g., at 16 conformers). This advantage can stem from TGT's explicit encoding of higher-order spatial relationships among features. Figure 3.5 further highlights that `MOAK` and `MI` consistently perform better than simpler aggregation schemes, reinforcing TGT's value in capturing 3D pharmacophore geometry better.

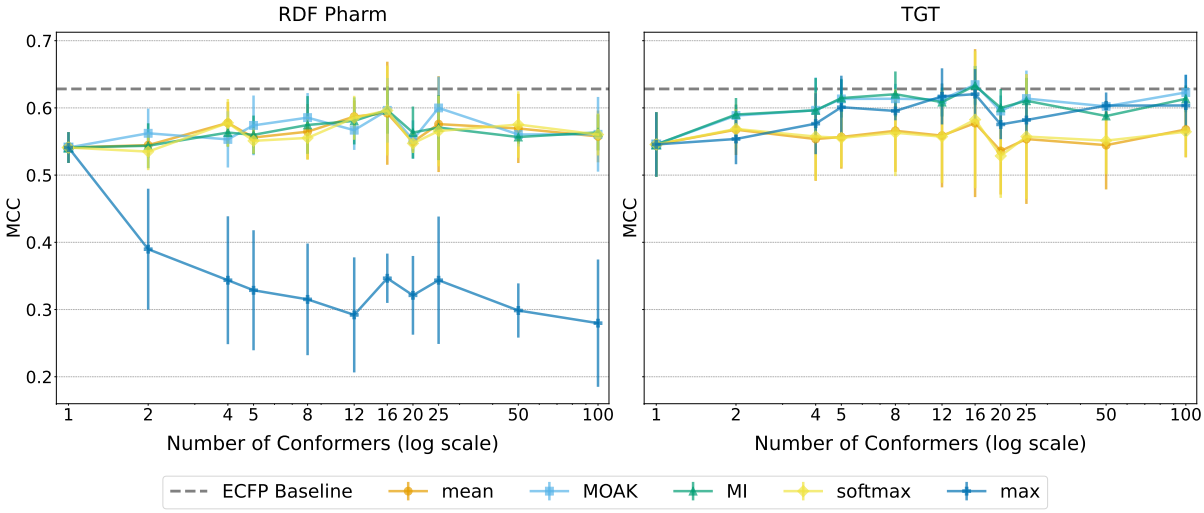

Figure 3.4: MCC comparison for RDF Pharm vs. TGT across conformer counts. TGT performance can equal or exceed ECFP with sufficient conformers.

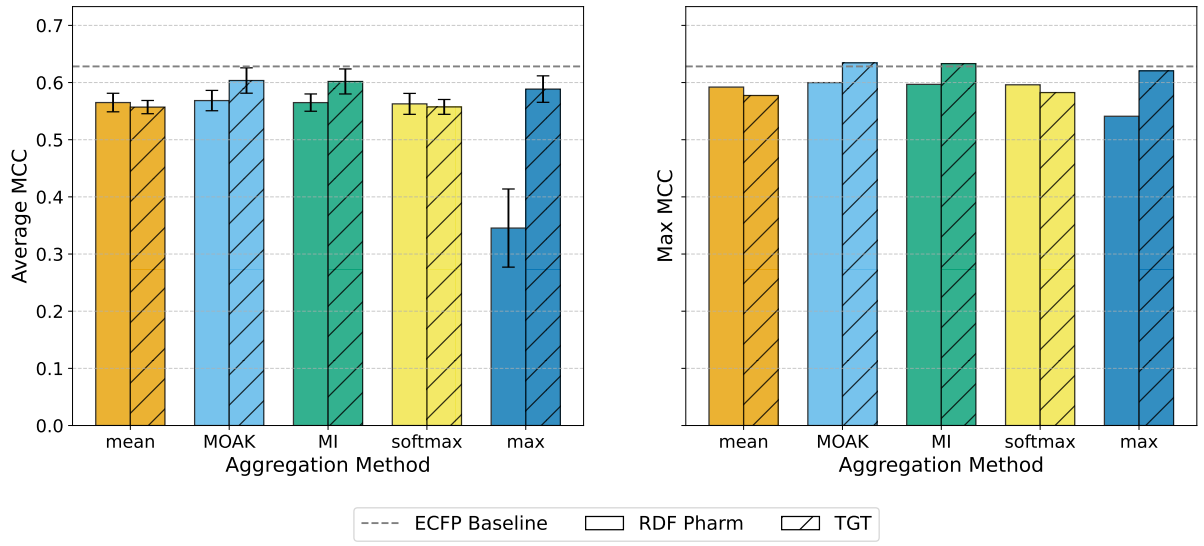

Figure 3.5: Average vs. best MCC values per aggregation across all available numbers of conformers for RDF on pharmacophores vs. TGT. TGT with `MOAK` or `MI` surpasses or nears ECFP performance.

3.3.3 TGT vs. PK

Figure 3.6 illustrates how PK—by directly comparing three-point pharmacophore features within a graph-based kernel—can outperform TGT and ECFP. With `MOAK`, `MI`, or `max` aggregation, PK can exceed the ECFP baseline at moderate conformer counts (e.g., 2, 5, 16). Indicating that a tailored 3D kernel on pharmacophore triplets can capture spatial and chemical nuances more effectively than simpler fingerprint approaches. Notably, the `max` aggregator achieves peak performance at 16 conformers (MCC = 0.66), despite lacking guaranteed p.s.d. However, given the high performance observed, we assume that the resulting kernel matrix is positive semi-definite in this case. A rigorous validation of this assumption is left for future work. Figure 3.7 further emphasizes PK's superior performance, showing higher average and maximum scores across aggregation methods and its consistent ability to use conformational information from the BACE dataset.

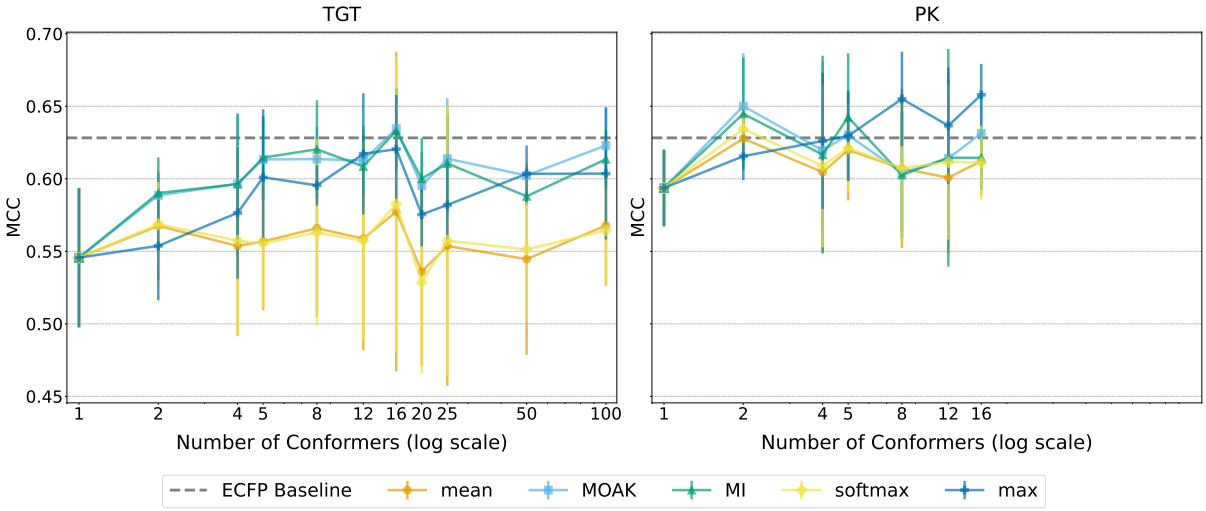

Figure 3.6: MCC comparison of TGT vs. PK across different conformer counts. PK offers higher performance.

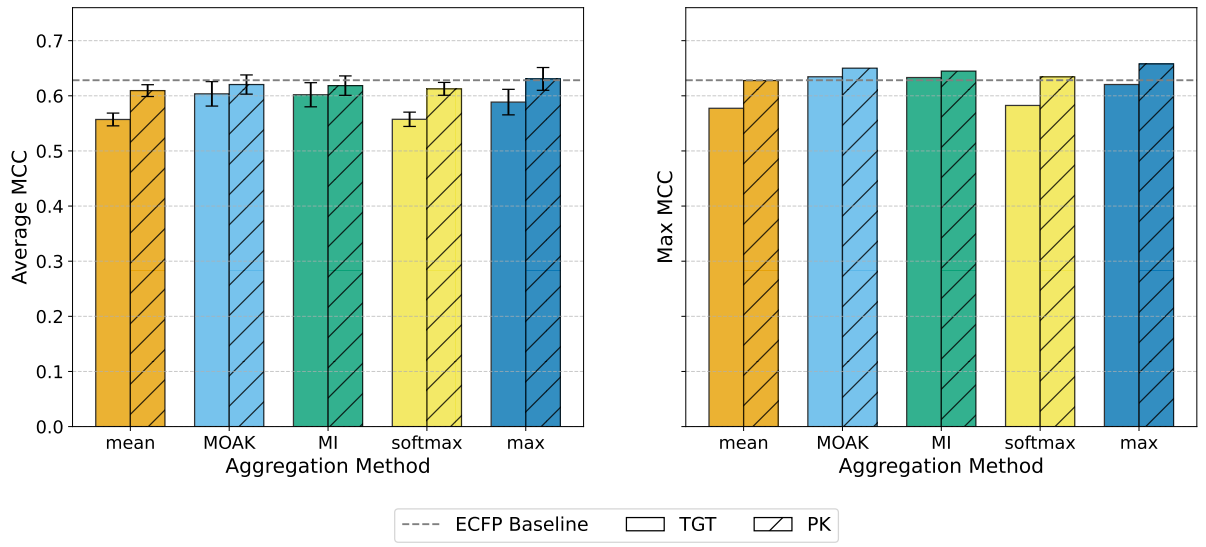

Figure 3.7: Comparison of TGT vs. PK showing average vs. best MCC across aggregation methods. Average and maximum values were calculated per aggregation method across all available conformer counts. PK performs better than TGT across all aggregations.

3.4 Comparative Analysis of PK Against Previous Methods

To unify our observations, we limited each molecule to 16 conformers. Compared to fingerprint-based methods, PK is substantially more expensive. It does not scale linearly with the number of conformers but instead grows quadratically, because each conformer is compared pairwise. Hence, restricting to 16 conformers helps keep computations feasible while still accounting for molecular flexibility.

We compared RDF (molecules and pharmacophores), TGT, and the PK. As shown in Figure 3.8, PK achieves the highest MCC in nearly all aggregators, with its `max` aggregation reaching 0.66 (at a conformer count of 16)—surpassing the ECFP baseline. Even on average, PK remains the top performer, indicating that its direct, kernel-based comparison of three-point pharmacophore features more effectively captures 3D spatial and chemical interactions. Unlike TGT, which also encodes pharmacophore triplets but uses simpler hashing, PK uses a full graph-kernel framework, applying distance- and label-based weighting at each comparison. This richer 3D modeling and the corresponding computational overhead limit PK to fewer conformers (here, 16). This improved accuracy over simpler RDF-based or TGT

representations confirms the added value of a kernel-driven approach to 3D pharmacophore similarity. Figure 3.1 also shows that with one conformer alone, PK still outperforms other single-conformer methods but cannot exceed the ECFP baseline—further underscoring the importance of conformational diversity for maximizing PK's predictive power.

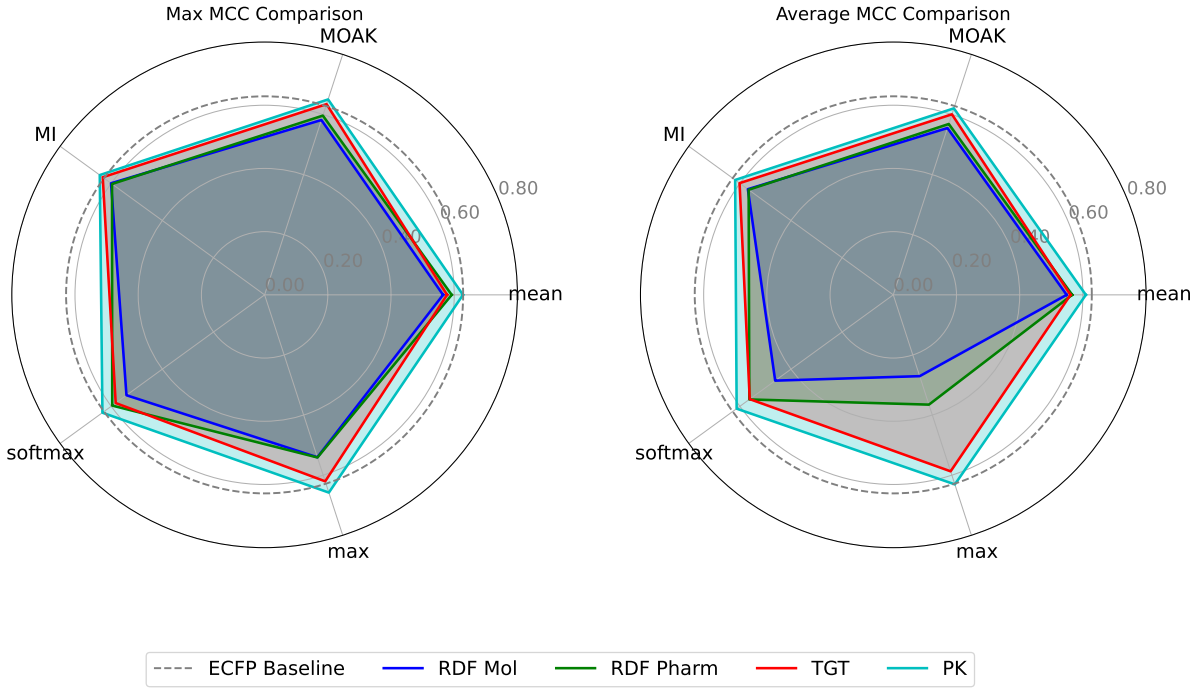

Figure 3.8: Radar plot comparing MCC (max and average) for each method (up to 16 conformers considered). PK consistently provides the highest scores across aggregators.

3.5 Impact of Training Data Size on Model Performance

For each method, we evaluated decreasing training-data fractions using a fixed limit of up to 16 conformers per molecule or pharmacophore. We selected 16 because each method performed optimally or near-optimally at that count, and using a lower conformer count might have negatively impacted performance, especially under limited-data conditions. This uniform choice allowed fair comparisons across all methods.

One of Kernel-SVMs characteristic is to deliver robust performance with limited data points—a critical advantage in drug discovery where researchers often work with just 10–20 samples for modeling. This capability stems from their dual formulation, which enables efficient handling of high-dimensional data[44]. By focusing on only support vectors to define margins and decision boundaries in the transformed feature space, a relative small dataset can be sufficient for achieving accurate prediction results. The kernel trick further enhances computational efficiency by operating on pairwise dot products rather than explicit high-dimensional coordinates[45].

To evaluate model robustness under data scarcity, we trained models on progressively smaller subsets of the training data while keeping hyperparameters and test sets fixed. This approach not only simulates real-world low-data scenarios but also reduces computational costs, particularly relevant for the PK.

Figure 3.9 illustrates MCC trends as training data decreases. While all methods degrade with fewer samples, PK and TGT achieve better prediction accuracy then RDF-based approaches, which fall below the ECFP baseline. Notably, TGT surpasses ECFP and PK for specific aggregation strategies, performing the best out of all methods as the traing data size decreases.

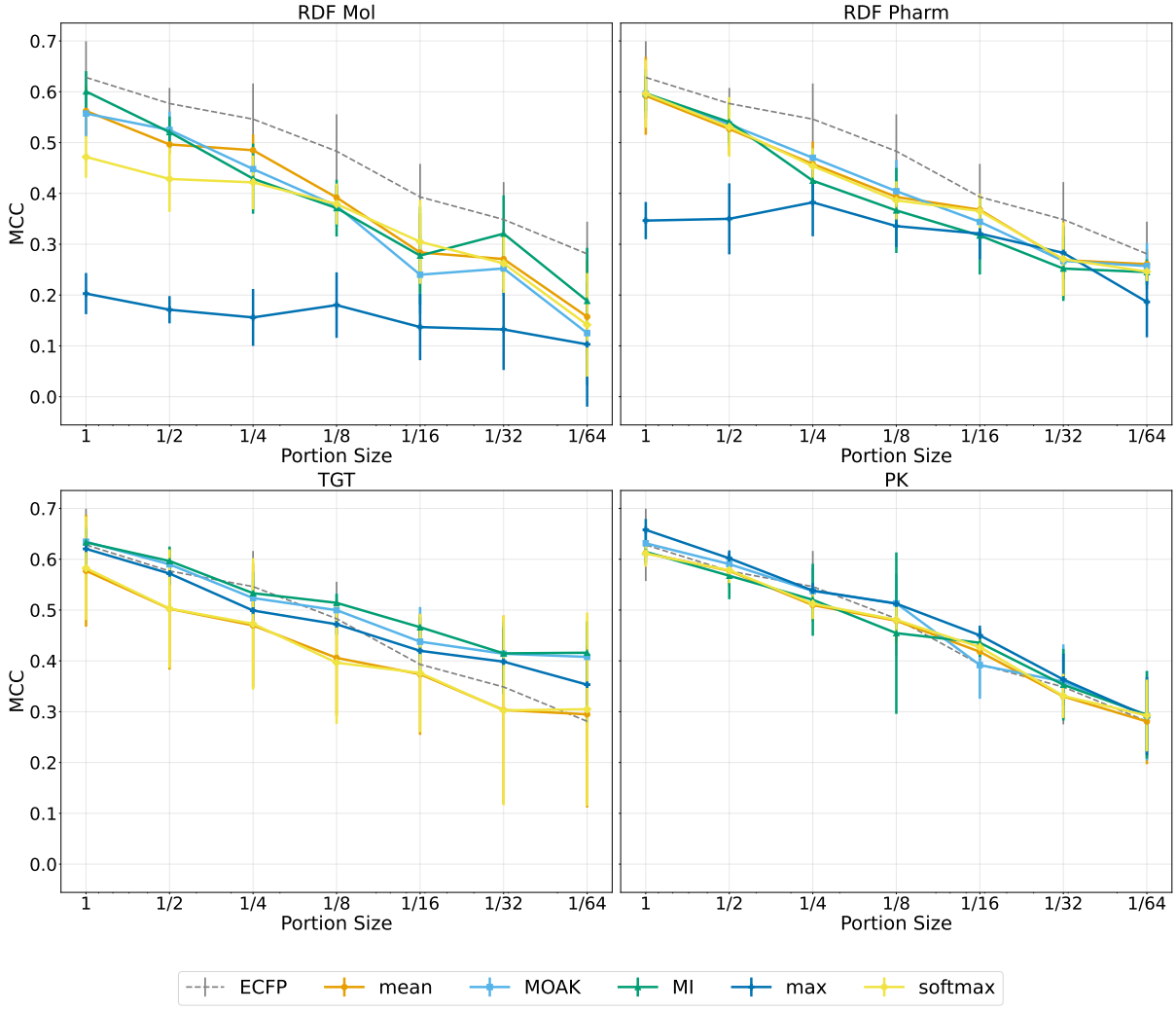

Figure 3.9: MCC trends for RDF Mol, RDF Pharm, TGT and PK as the training data fraction decreases.

Figure 3.10 highlights performance at the smallest training fraction (1/64). TGT obtains the highest MCC for `MOAK` and `mi`, indicating stronger robustness under extreme data scarcity than the other

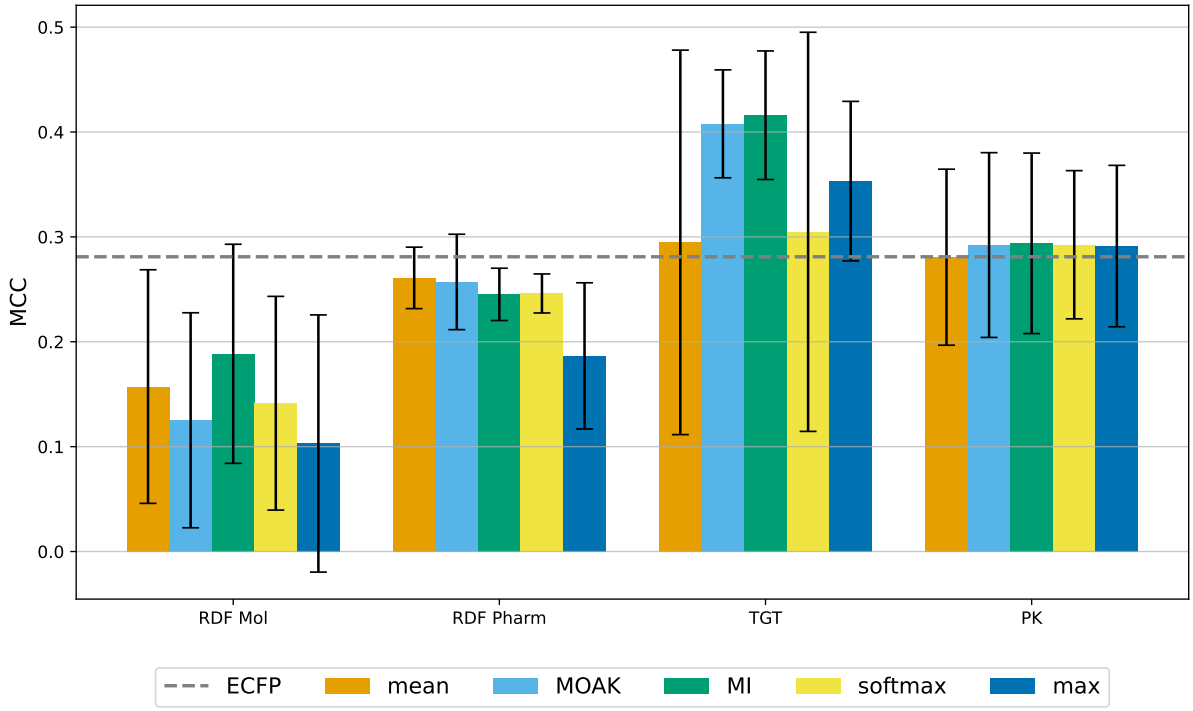

Figure 3.10: Grouped bar chart of MCC values at 1/64 of the training data across four datasets (RDF Mol, RDF Pharm, TGT, and PK). Each group shows five aggregation methods (`mean`, `max`, `softmax`, `MOAK`, and `MI`), with error bars indicating the standard deviation from multiple random splits.

methods. RDF Mol is weakest across nearly all aggregators, and although RDF Pharm improves on RDF Mol, it still trails behind ECFP. Meanwhile, PK slightly surpasses ECFP but does not exceed TGT, which contradicts earlier findings (Section 3.4) where PK achieved superior accuracy. Hence, for very limited data, TGT appears to offer the best predictive strength among the methods tested. The previous observation that PK outperformed TGT on the full training set could not be confirmed under this restricted data regime.

Chapter 4

Conclusion & Future Perspectives

4.1 Conclusion

This thesis explored the integration of 3D pharmacophore features and multi-instance learning (MIL) for bioactivity prediction, benchmarking kernel-based methods against traditional 2D and 3D molecular descriptors. While the Extended Connectivity Fingerprint (ECFP) remained a competitive baseline, the proposed Pharmacophore Kernel (PK) showed the potential to outperform it with suitable aggregation strategies.

Single-conformer models struggled to exceed ECFP, highlighting the limitations of purely static 3D representations. However, incorporating multiple conformers significantly increased predictive accuracy, emphasizing the importance of conformational diversity in molecular similarity. Comparisons among pharmacophore-based approaches revealed that PK, through its graph-kernel framework, exceeded both Typed Graph Triangle (TGT) descriptors and Radial Distribution Function (RDF) fingerprints *when trained on the entire dataset*. This advantage likely arises from PK's ability to weight pharmacophore triangles by both feature type and distance, offering a more structured and chemically meaningful comparison than TGT's hashing-based method.

In contrast, when training data was limited to smaller subsets, PK no longer surpassed TGT or ECFP. These conclusions stem from a single dataset and thus must be treated cautiously. Experimental measurements inherently contain noise, and design choices in data preparation, kernel parameterization, and fingerprint construction can all influence outcomes.

4.2 Future Perspectives

While the presented study demonstrates the potential of the Pharmacophore Kernel for bioactivity prediction, additional validation is essential. Larger and more diverse datasets, such as LIT-PCBA[46], will help clarify the generalizability of PK.

Selecting the optimal number of conformers and identifying suitable aggregation methods remain crucial for high predictive performance. Further exploration of alternative kernel functions could refine PK's capabilities and address potential biases arising from design decisions, including data preparation strategies, fingerprint parameters, and overall model architectures. It is also important to recognize the inherent noise in experimental measurements, which can introduce uncertainties in bioactivity prediction. Consequently, broader parameter sweeps and evaluations on multiple, more varied datasets would provide a deeper understanding of PK's strengths and the limitations of its application in drug discovery workflows.

To advance the practical application of pharmacophore-based models, the number of conformers per molecule should be systematically optimized alongside other hyperparameters within a nested cross-validation framework. First, PK Gram matrices for all candidate conformer counts would be precomputed, thus decoupling the computationally expensive conformer generation step from subsequent analyses. An inner cross-validation loop within the training set would then perform a grid search over both conformer counts and aggregation strategies, using averaged validation scores to identify the most promising configuration. The best combination would be retrained on the complete training set and finally evaluated on a held-out test set. This procedure generates validated hyperparameter guidelines (specifically, the optimal aggregation method and conformer count) for subsequent use on novel com-

pound collections without dedicated test sets. In practice, researchers can thus implement these validated parameters when training and validating on new active/inactive compound collections, ensuring robust model configuration without the need for external test data.

Bioactivity prediction accuracy could be further improved by extending PK with Subgraph Matching Kernels (SMK)[32], which generalize triplet-based approaches to capture larger pharmacophore subgraphs. In essence, PK—restricted to triplets—can be viewed as a special case of subgraph matching. By relaxing the triplet restriction, SMKs can incorporate higher-order structural motifs, unveiling more nuanced structure–activity correlations. Rather than only comparing three-point pharmacophores, an SMK-based approach would enumerate larger subgraphs (e.g., four or more nodes) and account for richer spatial and chemical interactions. Incorporating multiple conformers, similar to the approach in this study, would enable SMKs to adapt more flexibly to pharmacophore variability. Although expanding beyond triplets inevitably increases computational complexity, adopting strategies to improve efficiency and using the ability of kernel methods to make reliable predictions even with smaller datasets can keep runtimes feasible. SMK-based methods could thus capture deeper structure–activity relationships than triplet-based kernels to aim for better predictive accuracy in real-world drug discovery applications.

Bibliography

[1] C. G. Wermuth, C. R. Ganellin, P. Lindberg, and L. A. Mitscher. Glossary of terms used in medicinal chemistry (iupac recommendations 1998). *Pure Appl. Chem.*, 70:1129–1143, 1998.

[2] M. Wieder, A. Garon, U. Perricone, S. Boresch, T. Seidel, A. M. Almerico, and T. Langer. Common hits approach: Combining pharmacophore modeling and molecular dynamics simulations. *J. Chem. Model. Inf.*, 57(2):365–385, 2017.

[3] G. Hessler and K.-H. Baringhaus. The scaffold hopping potential of pharmacophores. *Drug Discov. Today Technol.*, 7(4):e263–e269, 2010.

[4] T. Seidel, O. Wieder, A. Garon, and T. Langer. Applications of the pharmacophore concept in natural product inspired drug design. *Mol. Inform.*, 39(11):e2000059, 2020.

[5] T. Seidel, S. D. Bryant, G. Ibis, G. Poli, and T. Langer. 3d pharmacophore modeling techniques in computer-aided molecular design using ligandscout. In *Tutorials in Chemoinformatics*, pages 279–309. 2017.

[6] T. Seidel, D. A. Schuetz, A. Garon, and T. Langer. The pharmacophore concept and its applications in computer-aided drug design. In *Progress in the Chemistry of Organic Natural Products*, volume 110, pages 99–133. 2019.

[7] G. Wolber and T. Langer. Ligandscout: 3-d pharmacophores derived from protein-bound ligands and their use as virtual screening filters. *J. Chem. Inf. Model.*, 45(1):160–169, 2005.

[8] G. Wolber, T. Seidel, F. Bendix, and T. Langer. Molecule-pharmacophore superpositioning and pattern matching in computational drug design. *Drug Discov. Today*, 13(1–2):23–29, 2008.

[9] G. Sliwoski, S. Kothiwale, J. Meiler, and E. W. Lowe. Computational methods in drug discovery. *Pharmacol. Rev.*, 66(1):334–395, 2014.

[10] P. Mahé, L. Ralaivola, V. Stoven, and J.-P. Vert. The pharmacophore kernel for virtual screening with support vector machines. *J. Chem. Model. Inf.*, 46(5):2003–2014, 2006.

[11] E. Estrada and D. Bonchev. Chemical graph theory. In *Mathematical Chemistry*, chapter 13.1. CRC Press, December 2013.

[12] L. David, A. Thakkar, R. Mercado, and O. Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *J. Cheminform.*, 12(56), 2020.

[13] P. Mahé and J.-P. Vert. Virtual screening with support vector machines and structure kernels. Technical report, Xerox Research Center Europe and Centre for Computational Biology, École des Mines de Paris, Meylan, France; Fontainebleau, France, February 2008. 6, chemin de Maupertuis, 38240 Meylan; 35, rue Saint Honoré, 77305 Fontainebleau.

[14] T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Lecture Notes in Computer Science*, pages 129–143, 2003.

[15] J.-L. Perret, P. Mahé, J.-P. Vert, T. Akutsu, M. Kanehisa, and N. Ueda. Chemcpp: A c++ toolbox for chemoinformatics. Version 1.0.2, November 2007. Available at: `http://sourceforge.net/projects/chemcpp/`. Accessed: 2025-01-17.

[16] H. Kashima, K. Tsuda, and A. Inokuchi. Kernels for graphs. In *Kernel Methods in Computational Biology*. 2004.

[17] P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Graph kernels for molecular structure-activity relationship analysis with support vector machines. *J. Chem. Model. Inf.*, 2005. Ecole des Mines de Paris, Fontainebleau, France; Bioinformatics Center, Kyoto University, Uji, Japan.

[18] L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi. Graph kernels for chemical informatics. *Neural Netw.*, 18(8):1093–1110, 2005.

[19] P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. Technical report, HAL, 2006. hal-00095488.

[20] Thomas Holder. Implementierung eines pharmakophor-kernels. Studienarbeit, Eberhard-Karls-Universität Tübingen, Wilhelm-Schickard-Institut für Informatik, Lehrstuhl Rechnerarchitektur, 2008.

[21] J. Koča. Travelling through conformational space: an approach for analyzing the conformational behaviour of flexible molecules. *Prog. Biophys. Mol. Biol.*, 70(2):137–173, 1998.

[22] N.-O. Friedrich, F. Flachsenberg, A. Meyder, K. Sommer, J. Kirchmair, and M. Rarey. Conformator: A novel method for the generation of conformer ensembles. *J. Chem. Model. Inf.*, 59(2):731–742, 2019.

[23] T. Seidel and O. Wieder. Chemical data processing toolkit (cdpkit): An open-source cheminformatics toolkit. `https://cdpkit.org`, 2024. Source code: `https://github.com/molinfo-vienna/CDPKit`. Documentation licensed under GFDL V1.2-or-later. Accessed: 2025-01-15.

[24] T. Seidel, C. Permann, O. Wieder, S. M. Kohlbacher, and T. Langer. High-quality conformer generation with conforge: Algorithm and performance assessment. *J. Chem. Model. Inf.*, 63(17):5549–5570, 2023.

[25] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, 1997.

[26] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-instance kernels. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pages 179–186, 2002.

[27] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *J. Mach. Learn. Res.*, 11:1201–1242, 2010.

[28] R. G. Mortimer and S. M. Blinder. Operators, matrices, and group theory. In R. G. Mortimer and S. M. Blinder, editors, *Mathematics for Physical Chemistry (Fifth Edition)*, pages 159–179. Elsevier, 2024.

[29] J.-L. Perret and P. Mahé. Chemcpp user-guide. `https://chemcpp.sourceforge.net/doc/chemcpp_user-guide.pdf`, October 2006. Accessed: 2025-01-22.

[30] H. Fröhlich, J. K. Wegner, F. Sieker, and A. Zell. Kernel functions for attributed molecular graphs – a new similarity-based approach to adme prediction in classification and regression. *QSAR Comb. Sci.*, 25(4):317–326, 2006.

[31] J.-P. Vert. The optimal assignment kernel is not positive definite. *CoRR*, abs/0801.4061, 2008.

[32] Nils Morten Kriege. *Comparing Graphs: Algorithms & Applications*. PhD thesis, Technische Universität Dortmund, Fakultät für Informatik, 2015.

[33] J. A. Mohr, B. J. Jain, and K. Obermayer. Molecule kernels: A descriptor- and alignment-free quantitative structure–activity relationship approach. *J. Chem. Model. Inf.*, 48:1868–1881, 2008.

[34] B. Engelhardt. Kernels and kernel methods. Lecture Notes, STA561: Probabilistic Machine Learning, Duke University, October 9 2013. Scribes: Dai, Y.; Lu, L.; Wu, W.

[35] R. Todeschini, V. Consonni, and P. Gramatica. Chemometrics in qsar. In S. D. Brown, R. Tauler, and B. Walczak, editors, *Comprehensive Chemometrics*, pages 129–172. Elsevier, 2009.

[36] E. N. Muratov, J. Bajorath, R. P. Sheridan, I. V. Tetko, D. Filimonov, V. Poroikov, T. I. Oprea, I. I. Baskin, A. Varnek, A. Roitberg, O. Isayev, S. Curtalolo, D. Fourches, Y. Cohen, A. Aspuru-Guzik, D. A. Winkler, D. Agrafiotis, A. Cherkasov, and A. Tropsha. Qsar without borders. *Chem. Soc. Rev.*, 49:3525–3564, Jun 2020.

[37] M. Wójcikowski, M. Kukiełka, M. M. Stepniewska-Dziubińska, and P. Siedlecki. Development of a protein–ligand extended connectivity (plec) fingerprint and its application for binding affinity predictions. *Bioinformatics*, 35(8):1334–1341, 2019.

[38] O. A. Raevskii, S. V. Trepalin, and A. N. Razdol'skii. New qsar descriptors calculated from interatomic interaction spectra. *Pharm. Chem. J.*, 34(12):19–22, December 2000.

[39] T. Seidel et al. Cdpkit: Pharmacophorerdfdescriptorcalculator. `https://github.com/molinfo-vienna/CDPKit/blob/master/Libs/C++/Source/CDPL/Descr/PharmacophoreRDFDescriptor.cpp`, 2023. Accessed: 2025-01-30.

[40] MolInfo Vienna. Cdpkit - npoint3dpharmacophorefingerprintgenerator. `https://github.com/molinfo-vienna/CDPKit`. Accessed on 2025-01-23.

[41] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, et al. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.*, 9(2):513–530, 2018.

[42] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *J. Chem. Model. Inf.*, 56(10):1936–1949, 2016.

[43] V. Venkatraman, V. I. Pérez-Nueno, L. Mavridis, and D. W. Ritchie. Comprehensive comparison of ligand-based virtual screening tools against the dud data set reveals limitations of current 3d methods. *J. Chem. Model. Inf.*, 50(12):2079–2093, 2010.

[44] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, 2000.

[45] Aarti Singh. Support vector machines - dual formulation and kernel trick. Machine Learning 10-315 Lecture. Carnegie Mellon University, School of Computer Science, Oct 28, 2020.

[46] Laboratoire d'innovation Thérapeutique. Lit-pcba: A dataset for virtual screening and machine learning. `https://drugdesign.unistra.fr/LIT-PCBA/`, 2023. Accessed: 2025-01-29.

[47] Microsoft. Visual studio code. `https://code.visualstudio.com/`. Accessed: 2025-01-16.

[48] Jupyter notebook. `https://jupyter.org/`. Accessed: 2025-01-19.

[49] Python Software Foundation. Python language reference, version 3.10. `https://www.python.org/`. Accessed: 2025-01-21.

[50] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Appendix A

Hyperparameter Optimization, Training, and Evaluation

A.1 Hyperparameter Optimization

We performed a grid search on a Support Vector Classifier (SVC), tuning:

- Regularization parameter $C \in \{0.01, 0.1, 1, 10, 100\}$
- Kernel parameter $\gamma \in \{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.1, 1, 10\}$

For Fingerprint-based models (ECFP, RDF, TGT), we used the RBF kernel, where γ controls the width of the Gaussian function.
In the PK model, we employed the same range for γ, which serves as the tolerance parameter in the triangular distance-based kernel.

Additonally, we tuned two multi-conformer aggregation kernels:

- `MOAK` (Most Optimal Assignment Kernel): $\beta \in \{0.001, 0.01, 0.1, 1, 10\}$.
- `MI` (Multi-Instance Kernel): $p \in \{0.001, 0.01, 0.1, 1, 10\}$.

A.2 Training and Evaluation Process

We repeated the following procedure four times with different random seeds:

1. Outer split: Stratify the dataset into 75% training and 25% testing.
2. Inner Cross-Validation: Perform a 5-fold stratified split on the training set to select the best hyperparameters based on average accuracy.
3. Final testing: Retrain a model with the optimal parameters on the entire training set, then evaluate it on the held-out test set.

A.3 Performance Metrics

We quantified performance using:

- Accuracy: Proportion of correctly predicted labels.
- F1-Score: Harmonic mean of precision and recall.
- ROC-AUC: Area under the ROC curve.
- MCC: Matthews correlation coefficient.

Appendix B

Tools

Computational experiments were conducted on a remote server, a 64-bit machine running Rocky Linux 9.4. The server is equipped with 15 GiB of system memory and an Intel Core i5-6600K CPU (3.50 GHz, 64-bit, SMP, vsyscall32). The primary development environment consisted of Visual Studio Code (VS Code)[47], Jupyter Notebook[48], and a conda environment with Python 3.10.14[49]. We employed a Support Vector Classifier (SVC) from scikit-learn (v. 1.4.2)[50].

Appendix C

Supplementary Tables

C.1 Full Data Evaluation Results

Table C.1: Results for RDF Mol

Conformer Count	Aggregation	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
1	mean	0.77 ± 0.01	0.74 ± 0.01	0.77 ± 0.01	0.54 ± 0.02
1	MOAK	0.77 ± 0.01	0.74 ± 0.01	0.77 ± 0.01	0.54 ± 0.02
1	MI	0.77 ± 0.01	0.74 ± 0.01	0.77 ± 0.01	0.54 ± 0.02
1	softmax	0.77 ± 0.01	0.74 ± 0.01	0.77 ± 0.01	0.54 ± 0.02
1	max	0.77 ± 0.01	0.74 ± 0.01	0.77 ± 0.01	0.54 ± 0.02
2	mean	0.75 ± 0.01	0.72 ± 0.01	0.75 ± 0.01	0.5 ± 0.02
2	MOAK	0.77 ± 0.02	0.74 ± 0.02	0.77 ± 0.02	0.53 ± 0.04
2	MI	0.77 ± 0.01	0.74 ± 0.01	0.76 ± 0.01	0.53 ± 0.02
2	softmax	0.72 ± 0.02	0.68 ± 0.01	0.72 ± 0.02	0.44 ± 0.03
2	max	0.65 ± 0.02	0.61 ± 0.01	0.65 ± 0.01	0.3 ± 0.03
4	mean	0.78 ± 0.03	0.76 ± 0.03	0.78 ± 0.03	0.56 ± 0.06
4	MOAK	0.78 ± 0.03	0.76 ± 0.04	0.78 ± 0.03	0.56 ± 0.06
4	MI	0.79 ± 0.03	0.76 ± 0.03	0.78 ± 0.03	0.57 ± 0.06
4	softmax	0.75 ± 0.03	0.72 ± 0.04	0.74 ± 0.03	0.49 ± 0.07
4	max	0.63 ± 0.04	0.59 ± 0.03	0.63 ± 0.03	0.26 ± 0.07
5	mean	0.78 ± 0.02	0.75 ± 0.02	0.77 ± 0.02	0.55 ± 0.03
5	MOAK	0.79 ± 0.01	0.76 ± 0.01	0.78 ± 0.01	0.57 ± 0.01
5	MI	0.79 ± 0.03	0.76 ± 0.04	0.78 ± 0.03	0.57 ± 0.06
5	softmax	0.73 ± 0.03	0.69 ± 0.03	0.72 ± 0.03	0.45 ± 0.06
5	max	0.62 ± 0.01	0.58 ± 0.01	0.62 ± 0.01	0.23 ± 0.03
8	mean	0.78 ± 0.02	0.75 ± 0.02	0.78 ± 0.02	0.56 ± 0.03
8	MOAK	0.78 ± 0.01	0.75 ± 0.01	0.77 ± 0.01	0.55 ± 0.02
8	MI	0.79 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	0.58 ± 0.01
8	softmax	0.72 ± 0.05	0.69 ± 0.05	0.72 ± 0.05	0.44 ± 0.09
8	max	0.59 ± 0.02	0.54 ± 0.02	0.59 ± 0.02	0.18 ± 0.04
12	mean	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
12	MOAK	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.58 ± 0.04
12	MI	0.8 ± 0.02	0.78 ± 0.02	0.8 ± 0.02	0.6 ± 0.04
12	softmax	0.71 ± 0.05	0.66 ± 0.06	0.7 ± 0.05	0.41 ± 0.1
12	max	0.6 ± 0.05	0.54 ± 0.05	0.59 ± 0.05	0.18 ± 0.1
16	mean	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.56 ± 0.05
16	MOAK	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.56 ± 0.05
16	MI	0.8 ± 0.02	0.78 ± 0.02	0.8 ± 0.02	0.6 ± 0.04
16	softmax	0.74 ± 0.02	0.71 ± 0.02	0.74 ± 0.02	0.47 ± 0.04
16	max	0.61 ± 0.02	0.56 ± 0.03	0.6 ± 0.02	0.2 ± 0.04
20	mean	0.76 ± 0.01	0.74 ± 0.01	0.76 ± 0.01	0.52 ± 0.02
20	MOAK	0.78 ± 0.01	0.75 ± 0.02	0.77 ± 0.01	0.55 ± 0.02
20	MI	0.78 ± 0.01	0.76 ± 0.01	0.78 ± 0.01	0.55 ± 0.02
20	softmax	0.72 ± 0.02	0.69 ± 0.02	0.71 ± 0.02	0.43 ± 0.04
20	max	0.6 ± 0.02	0.56 ± 0.02	0.6 ± 0.02	0.19 ± 0.05
25	mean	0.79 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	0.57 ± 0.02
25	MOAK	0.8 ± 0.01	0.78 ± 0.01	0.8 ± 0.01	0.59 ± 0.02
25	MI	0.8 ± 0.01	0.78 ± 0.01	0.8 ± 0.01	0.6 ± 0.02
25	softmax	0.72 ± 0.03	0.69 ± 0.04	0.72 ± 0.03	0.44 ± 0.06
25	max	0.59 ± 0.03	0.54 ± 0.04	0.58 ± 0.03	0.17 ± 0.05
50	mean	0.79 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	0.58 ± 0.02
50	MOAK	0.79 ± 0.01	0.76 ± 0.01	0.78 ± 0.01	0.57 ± 0.01
50	MI	0.79 ± 0.01	0.77 ± 0.02	0.79 ± 0.01	0.58 ± 0.03
50	softmax	0.73 ± 0.03	0.7 ± 0.03	0.72 ± 0.03	0.45 ± 0.06
50	max	0.58 ± 0.02	0.55 ± 0.03	0.58 ± 0.02	0.15 ± 0.04
100	mean	0.78 ± 0.02	0.76 ± 0.01	0.78 ± 0.02	0.57 ± 0.04
100	MOAK	0.78 ± 0.01	0.75 ± 0.01	0.78 ± 0.01	0.56 ± 0.03
100	MI	0.79 ± 0.01	0.76 ± 0.02	0.79 ± 0.01	0.58 ± 0.03
100	softmax	0.73 ± 0.02	0.7 ± 0.03	0.73 ± 0.02	0.46 ± 0.04
100	max	0.58 ± 0.03	0.55 ± 0.03	0.58 ± 0.03	0.16 ± 0.07

Table C.2: Results for RDF Pharm

Conformer Count	Aggregation	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
1	mean	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
1	MOAK	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
1	MI	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
1	softmax	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
1	max	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
2	mean	0.77 ± 0.01	0.75 ± 0.01	0.77 ± 0.01	0.54 ± 0.01
2	MOAK	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
2	MI	0.77 ± 0.02	0.75 ± 0.02	0.77 ± 0.02	0.54 ± 0.03
2	softmax	0.77 ± 0.01	0.75 ± 0.02	0.77 ± 0.01	0.53 ± 0.03
2	max	0.7 ± 0.04	0.66 ± 0.06	0.69 ± 0.05	0.39 ± 0.09
4	mean	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.58 ± 0.03
4	MOAK	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.55 ± 0.04
4	MI	0.78 ± 0.01	0.76 ± 0.01	0.78 ± 0.01	0.56 ± 0.02
4	softmax	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.58 ± 0.04
4	max	0.67 ± 0.05	0.64 ± 0.05	0.67 ± 0.05	0.34 ± 0.1
5	mean	0.78 ± 0.01	0.75 ± 0.01	0.78 ± 0.01	0.56 ± 0.02
5	MOAK	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.57 ± 0.04
5	MI	0.78 ± 0.01	0.75 ± 0.02	0.78 ± 0.02	0.56 ± 0.03
5	softmax	0.78 ± 0.01	0.75 ± 0.01	0.77 ± 0.01	0.55 ± 0.02
5	max	0.67 ± 0.05	0.64 ± 0.04	0.66 ± 0.04	0.33 ± 0.09
8	mean	0.78 ± 0.02	0.77 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
8	MOAK	0.79 ± 0.02	0.78 ± 0.02	0.79 ± 0.02	0.59 ± 0.04
8	MI	0.79 ± 0.02	0.77 ± 0.03	0.79 ± 0.02	0.57 ± 0.04
8	softmax	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.03
8	max	0.66 ± 0.04	0.62 ± 0.06	0.66 ± 0.04	0.32 ± 0.08
12	mean	0.79 ± 0.01	0.77 ± 0.02	0.79 ± 0.01	0.59 ± 0.02
12	MOAK	0.79 ± 0.02	0.76 ± 0.01	0.78 ± 0.01	0.57 ± 0.03
12	MI	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.58 ± 0.04
12	softmax	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.59 ± 0.03
12	max	0.65 ± 0.04	0.61 ± 0.05	0.65 ± 0.04	0.29 ± 0.09
16	mean	0.8 ± 0.04	0.78 ± 0.05	0.8 ± 0.04	0.59 ± 0.08
16	MOAK	0.8 ± 0.02	0.78 ± 0.03	0.8 ± 0.02	0.6 ± 0.05
16	MI	0.8 ± 0.02	0.78 ± 0.02	0.8 ± 0.02	0.6 ± 0.04
16	softmax	0.8 ± 0.03	0.78 ± 0.04	0.8 ± 0.03	0.6 ± 0.07
16	max	0.68 ± 0.02	0.63 ± 0.01	0.67 ± 0.02	0.35 ± 0.04
20	mean	0.78 ± 0.01	0.76 ± 0.01	0.77 ± 0.01	0.55 ± 0.01
20	MOAK	0.78 ± 0.01	0.76 ± 0.02	0.78 ± 0.01	0.55 ± 0.03
20	MI	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
20	softmax	0.77 ± 0.01	0.76 ± 0.01	0.77 ± 0.01	0.55 ± 0.01
20	max	0.66 ± 0.03	0.62 ± 0.03	0.66 ± 0.03	0.32 ± 0.06
25	mean	0.79 ± 0.03	0.77 ± 0.04	0.79 ± 0.04	0.58 ± 0.07
25	MOAK	0.8 ± 0.02	0.78 ± 0.03	0.8 ± 0.02	0.6 ± 0.05
25	MI	0.79 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.57 ± 0.05
25	softmax	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.03	0.57 ± 0.05
25	max	0.67 ± 0.05	0.64 ± 0.04	0.67 ± 0.05	0.34 ± 0.09
50	mean	0.79 ± 0.02	0.76 ± 0.03	0.78 ± 0.03	0.57 ± 0.05
50	MOAK	0.78 ± 0.01	0.75 ± 0.01	0.78 ± 0.01	0.56 ± 0.02
50	MI	0.78 ± 0.01	0.75 ± 0.01	0.77 ± 0.01	0.56 ± 0.01
50	softmax	0.79 ± 0.02	0.77 ± 0.03	0.79 ± 0.03	0.58 ± 0.05
50	max	0.65 ± 0.02	0.61 ± 0.02	0.65 ± 0.02	0.3 ± 0.04
100	mean	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
100	MOAK	0.78 ± 0.03	0.75 ± 0.03	0.78 ± 0.03	0.56 ± 0.06
100	MI	0.78 ± 0.01	0.75 ± 0.01	0.78 ± 0.01	0.56 ± 0.03
100	softmax	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.03
100	max	0.64 ± 0.05	0.59 ± 0.05	0.64 ± 0.05	0.28 ± 0.09

Table C.3: Results for TGT

Conformer Count	Aggregation	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
1	mean	0.78 ± 0.02	0.75 ± 0.03	0.77 ± 0.02	0.55 ± 0.05
1	MOAK	0.78 ± 0.02	0.75 ± 0.03	0.77 ± 0.02	0.55 ± 0.05
1	MI	0.78 ± 0.02	0.75 ± 0.03	0.77 ± 0.02	0.55 ± 0.05
1	softmax	0.78 ± 0.02	0.75 ± 0.03	0.77 ± 0.02	0.55 ± 0.05
1	max	0.78 ± 0.02	0.75 ± 0.03	0.77 ± 0.02	0.55 ± 0.05
2	mean	0.79 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.57 ± 0.04
2	MOAK	0.8 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	0.59 ± 0.02
2	MI	0.8 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	0.59 ± 0.02
2	softmax	0.79 ± 0.01	0.76 ± 0.01	0.78 ± 0.01	0.57 ± 0.03
2	max	0.78 ± 0.02	0.75 ± 0.02	0.78 ± 0.02	0.55 ± 0.04
4	mean	0.78 ± 0.03	0.76 ± 0.04	0.78 ± 0.03	0.55 ± 0.06
4	MOAK	0.8 ± 0.02	0.78 ± 0.03	0.8 ± 0.02	0.6 ± 0.05
4	MI	0.8 ± 0.02	0.78 ± 0.03	0.8 ± 0.02	0.6 ± 0.05
4	softmax	0.78 ± 0.03	0.76 ± 0.04	0.78 ± 0.03	0.56 ± 0.07
4	max	0.79 ± 0.02	0.77 ± 0.03	0.79 ± 0.02	0.58 ± 0.05
5	mean	0.78 ± 0.02	0.75 ± 0.03	0.78 ± 0.02	0.56 ± 0.05
5	MOAK	0.81 ± 0.01	0.79 ± 0.02	0.81 ± 0.01	0.61 ± 0.03
5	MI	0.81 ± 0.01	0.79 ± 0.02	0.81 ± 0.01	0.61 ± 0.03
5	softmax	0.78 ± 0.02	0.75 ± 0.03	0.78 ± 0.02	0.56 ± 0.04
5	max	0.8 ± 0.02	0.78 ± 0.03	0.8 ± 0.02	0.6 ± 0.05
8	mean	0.78 ± 0.03	0.77 ± 0.03	0.78 ± 0.03	0.57 ± 0.06
8	MOAK	0.81 ± 0.01	0.79 ± 0.01	0.81 ± 0.01	0.61 ± 0.02
8	MI	0.81 ± 0.02	0.8 ± 0.02	0.81 ± 0.02	0.62 ± 0.03
8	softmax	0.78 ± 0.03	0.76 ± 0.04	0.78 ± 0.03	0.56 ± 0.06
8	max	0.8 ± 0.01	0.78 ± 0.01	0.8 ± 0.01	0.6 ± 0.01
12	mean	0.78 ± 0.04	0.75 ± 0.06	0.78 ± 0.04	0.56 ± 0.08
12	MOAK	0.81 ± 0.01	0.79 ± 0.01	0.81 ± 0.01	0.61 ± 0.02
12	MI	0.81 ± 0.01	0.79 ± 0.01	0.8 ± 0.01	0.61 ± 0.02
12	softmax	0.78 ± 0.03	0.75 ± 0.05	0.78 ± 0.04	0.56 ± 0.07
12	max	0.81 ± 0.02	0.79 ± 0.02	0.81 ± 0.02	0.62 ± 0.04
16	mean	0.79 ± 0.05	0.77 ± 0.07	0.79 ± 0.06	0.58 ± 0.11
16	MOAK	0.82 ± 0.01	0.8 ± 0.01	0.82 ± 0.01	0.63 ± 0.03
16	MI	0.82 ± 0.01	0.8 ± 0.01	0.82 ± 0.01	0.63 ± 0.03
16	softmax	0.79 ± 0.05	0.77 ± 0.06	0.79 ± 0.05	0.58 ± 0.1
16	max	0.81 ± 0.02	0.8 ± 0.02	0.81 ± 0.02	0.62 ± 0.04
20	mean	0.77 ± 0.03	0.75 ± 0.05	0.77 ± 0.03	0.54 ± 0.06
20	MOAK	0.8 ± 0.01	0.78 ± 0.01	0.8 ± 0.01	0.6 ± 0.02
20	MI	0.8 ± 0.01	0.79 ± 0.02	0.8 ± 0.01	0.6 ± 0.03
20	softmax	0.77 ± 0.03	0.74 ± 0.04	0.76 ± 0.03	0.53 ± 0.06
20	max	0.79 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	0.58 ± 0.02
25	mean	0.78 ± 0.05	0.75 ± 0.08	0.77 ± 0.05	0.55 ± 0.1
25	MOAK	0.81 ± 0.02	0.79 ± 0.03	0.81 ± 0.02	0.61 ± 0.04
25	MI	0.81 ± 0.02	0.79 ± 0.02	0.81 ± 0.02	0.61 ± 0.03
25	softmax	0.78 ± 0.04	0.75 ± 0.07	0.78 ± 0.05	0.56 ± 0.09
25	max	0.79 ± 0.01	0.77 ± 0.02	0.79 ± 0.02	0.58 ± 0.03
50	mean	0.77 ± 0.03	0.75 ± 0.04	0.77 ± 0.03	0.54 ± 0.07
50	MOAK	0.8 ± 0.01	0.78 ± 0.01	0.8 ± 0.01	0.6 ± 0.01
50	MI	0.79 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	0.59 ± 0.01
50	softmax	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.03	0.55 ± 0.05
50	max	0.8 ± 0.01	0.78 ± 0.01	0.8 ± 0.01	0.6 ± 0.02
100	mean	0.78 ± 0.02	0.76 ± 0.04	0.78 ± 0.02	0.57 ± 0.04
100	MOAK	0.81 ± 0.01	0.79 ± 0.01	0.81 ± 0.01	0.62 ± 0.03
100	MI	0.81 ± 0.01	0.79 ± 0.01	0.81 ± 0.01	0.61 ± 0.02
100	softmax	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.56 ± 0.04
100	max	0.8 ± 0.02	0.79 ± 0.02	0.8 ± 0.02	0.6 ± 0.05

Table C.4: Results for PK

Conformer Count	Aggregation	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
1	mean	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
1	MOAK	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
1	MI	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
1	softmax	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
1	max	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.02
2	mean	0.77 ± 0.01	0.75 ± 0.01	0.77 ± 0.01	0.54 ± 0.01
2	MOAK	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
2	MI	0.77 ± 0.02	0.75 ± 0.02	0.77 ± 0.02	0.54 ± 0.03
2	softmax	0.77 ± 0.01	0.75 ± 0.02	0.77 ± 0.01	0.53 ± 0.03
2	max	0.7 ± 0.04	0.66 ± 0.06	0.69 ± 0.05	0.39 ± 0.09
4	mean	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.58 ± 0.03
4	MOAK	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.55 ± 0.04
4	MI	0.78 ± 0.01	0.76 ± 0.01	0.78 ± 0.01	0.56 ± 0.02
4	softmax	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.58 ± 0.04
4	max	0.67 ± 0.05	0.64 ± 0.05	0.67 ± 0.05	0.34 ± 0.1
5	mean	0.78 ± 0.01	0.75 ± 0.01	0.78 ± 0.01	0.56 ± 0.02
5	MOAK	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.57 ± 0.04
5	MI	0.78 ± 0.01	0.75 ± 0.02	0.78 ± 0.02	0.56 ± 0.03
5	softmax	0.78 ± 0.01	0.75 ± 0.01	0.77 ± 0.01	0.55 ± 0.02
5	max	0.67 ± 0.05	0.64 ± 0.04	0.66 ± 0.04	0.33 ± 0.09
8	mean	0.78 ± 0.02	0.77 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
8	MOAK	0.79 ± 0.02	0.78 ± 0.02	0.79 ± 0.02	0.59 ± 0.04
8	MI	0.79 ± 0.02	0.77 ± 0.03	0.79 ± 0.02	0.57 ± 0.04
8	softmax	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.03
8	max	0.66 ± 0.04	0.62 ± 0.06	0.66 ± 0.04	0.32 ± 0.08
12	mean	0.79 ± 0.01	0.77 ± 0.02	0.79 ± 0.01	0.59 ± 0.02
12	MOAK	0.79 ± 0.02	0.76 ± 0.01	0.78 ± 0.01	0.57 ± 0.03
12	MI	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.58 ± 0.04
12	softmax	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.02	0.59 ± 0.03
12	max	0.65 ± 0.04	0.61 ± 0.05	0.65 ± 0.04	0.29 ± 0.09
16	mean	0.8 ± 0.04	0.78 ± 0.05	0.8 ± 0.04	0.59 ± 0.08
16	MOAK	0.8 ± 0.02	0.78 ± 0.03	0.8 ± 0.02	0.6 ± 0.05
16	MI	0.8 ± 0.02	0.78 ± 0.02	0.8 ± 0.02	0.6 ± 0.04
16	softmax	0.8 ± 0.03	0.78 ± 0.04	0.8 ± 0.03	0.6 ± 0.07
16	max	0.68 ± 0.02	0.63 ± 0.01	0.67 ± 0.02	0.35 ± 0.04
20	mean	0.78 ± 0.01	0.76 ± 0.01	0.77 ± 0.01	0.55 ± 0.01
20	MOAK	0.78 ± 0.01	0.76 ± 0.02	0.78 ± 0.01	0.55 ± 0.03
20	MI	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
20	softmax	0.77 ± 0.01	0.76 ± 0.01	0.77 ± 0.01	0.55 ± 0.01
20	max	0.66 ± 0.03	0.62 ± 0.03	0.66 ± 0.03	0.32 ± 0.06
25	mean	0.79 ± 0.03	0.77 ± 0.04	0.79 ± 0.04	0.58 ± 0.07
25	MOAK	0.8 ± 0.02	0.78 ± 0.03	0.8 ± 0.02	0.6 ± 0.05
25	MI	0.79 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.57 ± 0.05
25	softmax	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.03	0.57 ± 0.05
25	max	0.67 ± 0.05	0.64 ± 0.04	0.67 ± 0.05	0.34 ± 0.09
50	mean	0.79 ± 0.02	0.76 ± 0.03	0.78 ± 0.03	0.57 ± 0.05
50	MOAK	0.78 ± 0.01	0.75 ± 0.01	0.78 ± 0.01	0.56 ± 0.02
50	MI	0.78 ± 0.01	0.75 ± 0.01	0.77 ± 0.01	0.56 ± 0.01
50	softmax	0.79 ± 0.02	0.77 ± 0.03	0.79 ± 0.03	0.58 ± 0.05
50	max	0.65 ± 0.02	0.61 ± 0.02	0.65 ± 0.02	0.3 ± 0.04
100	mean	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.04
100	MOAK	0.78 ± 0.03	0.75 ± 0.03	0.78 ± 0.03	0.56 ± 0.06
100	MI	0.78 ± 0.01	0.75 ± 0.01	0.78 ± 0.01	0.56 ± 0.03
100	softmax	0.78 ± 0.02	0.76 ± 0.02	0.78 ± 0.02	0.56 ± 0.03
100	max	0.64 ± 0.05	0.59 ± 0.05	0.64 ± 0.05	0.28 ± 0.09

C.2 Evaluation Results for Fractions of Training Data

Table C.5: RDF Mol Fraction Results (16 confs.)

Conformer Count	Fraction	Aggregation	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
16	1	mean	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.56 ± 0.05
16	1	MOAK	0.78 ± 0.02	0.76 ± 0.03	0.78 ± 0.02	0.56 ± 0.05
16	1	MI	0.8 ± 0.02	0.78 ± 0.02	0.8 ± 0.02	0.6 ± 0.04
16	1	max	0.61 ± 0.02	0.56 ± 0.03	0.6 ± 0.02	0.2 ± 0.04
16	1	softmax	0.74 ± 0.02	0.71 ± 0.02	0.74 ± 0.02	0.47 ± 0.04
16	0.5	mean	0.75 ± 0.01	0.72 ± 0.01	0.75 ± 0.01	0.5 ± 0.02
16	0.5	MOAK	0.76 ± 0.02	0.74 ± 0.02	0.76 ± 0.02	0.52 ± 0.04
16	0.5	MI	0.76 ± 0.01	0.74 ± 0.02	0.76 ± 0.02	0.52 ± 0.03
16	0.5	max	0.59 ± 0.01	0.54 ± 0.01	0.58 ± 0.01	0.17 ± 0.03
16	0.5	softmax	0.72 ± 0.03	0.68 ± 0.03	0.71 ± 0.03	0.43 ± 0.06
16	0.25	mean	0.75 ± 0.02	0.71 ± 0.02	0.74 ± 0.02	0.49 ± 0.03
16	0.25	MOAK	0.73 ± 0.02	0.7 ± 0.02	0.72 ± 0.02	0.45 ± 0.03
16	0.25	MI	0.72 ± 0.03	0.69 ± 0.05	0.71 ± 0.04	0.43 ± 0.07
16	0.25	max	0.58 ± 0.03	0.52 ± 0.03	0.58 ± 0.03	0.16 ± 0.06
16	0.25	softmax	0.71 ± 0.03	0.68 ± 0.03	0.71 ± 0.03	0.42 ± 0.05
16	0.125	mean	0.7 ± 0.01	0.66 ± 0.02	0.69 ± 0.01	0.39 ± 0.02
16	0.125	MOAK	0.69 ± 0.02	0.65 ± 0.03	0.69 ± 0.02	0.38 ± 0.04
16	0.125	MI	0.69 ± 0.03	0.66 ± 0.03	0.68 ± 0.03	0.37 ± 0.06
16	0.125	max	0.59 ± 0.03	0.55 ± 0.02	0.59 ± 0.03	0.18 ± 0.06
16	0.125	softmax	0.69 ± 0.02	0.65 ± 0.03	0.69 ± 0.02	0.38 ± 0.04
16	0.0625	mean	0.65 ± 0.04	0.6 ± 0.04	0.64 ± 0.04	0.28 ± 0.08
16	0.0625	MOAK	0.62 ± 0.04	0.57 ± 0.03	0.62 ± 0.04	0.24 ± 0.08
16	0.0625	MI	0.64 ± 0.05	0.6 ± 0.05	0.64 ± 0.05	0.28 ± 0.1
16	0.0625	max	0.58 ± 0.03	0.5 ± 0.03	0.57 ± 0.03	0.14 ± 0.07
16	0.0625	softmax	0.66 ± 0.04	0.61 ± 0.05	0.65 ± 0.04	0.3 ± 0.08
16	0.03125	mean	0.64 ± 0.03	0.58 ± 0.04	0.63 ± 0.03	0.27 ± 0.06
16	0.03125	MOAK	0.63 ± 0.03	0.57 ± 0.02	0.62 ± 0.02	0.25 ± 0.05
16	0.03125	MI	0.66 ± 0.04	0.62 ± 0.05	0.66 ± 0.04	0.32 ± 0.08
16	0.03125	max	0.57 ± 0.04	0.5 ± 0.06	0.57 ± 0.04	0.13 ± 0.08
16	0.03125	softmax	0.64 ± 0.03	0.58 ± 0.04	0.63 ± 0.03	0.26 ± 0.06
16	0.015625	mean	0.59 ± 0.05	0.51 ± 0.07	0.58 ± 0.05	0.16 ± 0.11
16	0.015625	MOAK	0.57 ± 0.05	0.49 ± 0.06	0.56 ± 0.05	0.13 ± 0.1
16	0.015625	MI	0.6 ± 0.05	0.54 ± 0.06	0.59 ± 0.05	0.19 ± 0.1
16	0.015625	max	0.56 ± 0.06	0.49 ± 0.07	0.55 ± 0.06	0.1 ± 0.12
16	0.015625	softmax	0.58 ± 0.05	0.49 ± 0.06	0.57 ± 0.05	0.14 ± 0.1

Table C.6: RDF Pharm Fraction Results (16 confs.)

Conformer Count	Fraction	Aggregation	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
16	1	mean	0.8 ± 0.04	0.78 ± 0.05	0.8 ± 0.04	0.59 ± 0.08
16	1	MOAK	0.8 ± 0.02	0.78 ± 0.03	0.8 ± 0.02	0.6 ± 0.05
16	1	MI	0.8 ± 0.02	0.78 ± 0.02	0.8 ± 0.02	0.6 ± 0.04
16	1	max	0.68 ± 0.02	0.63 ± 0.01	0.67 ± 0.02	0.35 ± 0.04
16	1	softmax	0.8 ± 0.03	0.78 ± 0.04	0.8 ± 0.03	0.6 ± 0.07
16	0.5	mean	0.77 ± 0.03	0.74 ± 0.03	0.76 ± 0.03	0.53 ± 0.05
16	0.5	MOAK	0.77 ± 0.02	0.74 ± 0.03	0.77 ± 0.02	0.54 ± 0.04
16	0.5	MI	0.77 ± 0.02	0.74 ± 0.03	0.77 ± 0.02	0.54 ± 0.04
16	0.5	max	0.68 ± 0.04	0.64 ± 0.03	0.67 ± 0.03	0.35 ± 0.07
16	0.5	softmax	0.77 ± 0.03	0.74 ± 0.03	0.76 ± 0.03	0.53 ± 0.06
16	0.25	mean	0.73 ± 0.02	0.7 ± 0.03	0.73 ± 0.02	0.46 ± 0.04
16	0.25	MOAK	0.74 ± 0.01	0.71 ± 0.01	0.73 ± 0.01	0.47 ± 0.02
16	0.25	MI	0.72 ± 0.02	0.68 ± 0.03	0.71 ± 0.02	0.43 ± 0.05
16	0.25	max	0.69 ± 0.03	0.65 ± 0.03	0.69 ± 0.03	0.38 ± 0.07
16	0.25	softmax	0.73 ± 0.02	0.69 ± 0.03	0.73 ± 0.02	0.45 ± 0.04
16	0.125	mean	0.7 ± 0.02	0.65 ± 0.03	0.69 ± 0.02	0.39 ± 0.04
16	0.125	MOAK	0.7 ± 0.03	0.67 ± 0.04	0.7 ± 0.03	0.4 ± 0.06
16	0.125	MI	0.69 ± 0.04	0.64 ± 0.05	0.68 ± 0.04	0.37 ± 0.08
16	0.125	max	0.67 ± 0.02	0.6 ± 0.02	0.66 ± 0.02	0.34 ± 0.04
16	0.125	softmax	0.7 ± 0.02	0.65 ± 0.03	0.69 ± 0.02	0.39 ± 0.04
16	0.0625	mean	0.69 ± 0.01	0.64 ± 0.02	0.68 ± 0.01	0.37 ± 0.03
16	0.0625	MOAK	0.67 ± 0.01	0.64 ± 0.01	0.67 ± 0.01	0.34 ± 0.02
16	0.0625	MI	0.66 ± 0.04	0.62 ± 0.04	0.66 ± 0.04	0.32 ± 0.08
16	0.0625	max	0.66 ± 0.02	0.6 ± 0.03	0.65 ± 0.02	0.32 ± 0.05
16	0.0625	softmax	0.69 ± 0.02	0.64 ± 0.02	0.68 ± 0.02	0.37 ± 0.03
16	0.03125	mean	0.64 ± 0.04	0.58 ± 0.02	0.63 ± 0.04	0.27 ± 0.08
16	0.03125	MOAK	0.64 ± 0.01	0.57 ± 0.01	0.63 ± 0.01	0.27 ± 0.02
16	0.03125	MI	0.63 ± 0.03	0.56 ± 0.05	0.62 ± 0.03	0.25 ± 0.06
16	0.03125	max	0.65 ± 0.03	0.57 ± 0.05	0.64 ± 0.03	0.28 ± 0.05
16	0.03125	softmax	0.64 ± 0.04	0.58 ± 0.02	0.63 ± 0.03	0.27 ± 0.07
16	0.015625	mean	0.63 ± 0.01	0.56 ± 0.07	0.62 ± 0.01	0.26 ± 0.03
16	0.015625	MOAK	0.62 ± 0.02	0.61 ± 0.05	0.63 ± 0.02	0.26 ± 0.05
16	0.015625	MI	0.62 ± 0.01	0.58 ± 0.05	0.62 ± 0.01	0.25 ± 0.02
16	0.015625	max	0.6 ± 0.04	0.55 ± 0.04	0.59 ± 0.03	0.19 ± 0.07
16	0.015625	softmax	0.62 ± 0.01	0.58 ± 0.05	0.62 ± 0.01	0.25 ± 0.02

Table C.7: TGT Fraction Results (16 confs.)

Conformer Count	Fraction	Aggregation	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
16	1	mean	0.79 ± 0.05	0.77 ± 0.07	0.79 ± 0.06	0.58 ± 0.11
16	1	MOAK	0.82 ± 0.01	0.8 ± 0.01	0.82 ± 0.01	0.63 ± 0.03
16	1	MI	0.82 ± 0.01	0.8 ± 0.01	0.82 ± 0.01	0.63 ± 0.03
16	1	max	0.81 ± 0.02	0.8 ± 0.02	0.81 ± 0.02	0.62 ± 0.04
16	1	softmax	0.79 ± 0.05	0.77 ± 0.06	0.79 ± 0.05	0.58 ± 0.1
16	0.5	mean	0.75 ± 0.06	0.7 ± 0.12	0.75 ± 0.07	0.5 ± 0.12
16	0.5	MOAK	0.8 ± 0.01	0.78 ± 0.01	0.79 ± 0.01	0.59 ± 0.02
16	0.5	MI	0.8 ± 0.01	0.78 ± 0.01	0.8 ± 0.01	0.6 ± 0.03
16	0.5	max	0.79 ± 0.01	0.76 ± 0.02	0.79 ± 0.01	0.57 ± 0.02
16	0.5	softmax	0.75 ± 0.06	0.7 ± 0.11	0.75 ± 0.07	0.5 ± 0.12
16	0.25	mean	0.73 ± 0.06	0.67 ± 0.13	0.73 ± 0.07	0.47 ± 0.12
16	0.25	MOAK	0.76 ± 0.02	0.74 ± 0.02	0.76 ± 0.02	0.52 ± 0.04
16	0.25	MI	0.77 ± 0.02	0.74 ± 0.02	0.76 ± 0.02	0.53 ± 0.04
16	0.25	max	0.75 ± 0.01	0.72 ± 0.01	0.75 ± 0.01	0.5 ± 0.02
16	0.25	softmax	0.74 ± 0.07	0.67 ± 0.14	0.73 ± 0.08	0.47 ± 0.13
16	0.125	mean	0.7 ± 0.06	0.6 ± 0.17	0.69 ± 0.07	0.41 ± 0.11
16	0.125	MOAK	0.75 ± 0.02	0.72 ± 0.02	0.75 ± 0.01	0.5 ± 0.03
16	0.125	MI	0.76 ± 0.01	0.72 ± 0.02	0.75 ± 0.01	0.51 ± 0.02
16	0.125	max	0.74 ± 0.01	0.69 ± 0.01	0.73 ± 0.01	0.47 ± 0.02
16	0.125	softmax	0.7 ± 0.06	0.6 ± 0.16	0.69 ± 0.07	0.4 ± 0.12
16	0.0625	mean	0.69 ± 0.06	0.56 ± 0.21	0.67 ± 0.08	0.37 ± 0.12
16	0.0625	MOAK	0.72 ± 0.03	0.68 ± 0.04	0.72 ± 0.03	0.44 ± 0.07
16	0.0625	MI	0.73 ± 0.01	0.68 ± 0.03	0.73 ± 0.01	0.47 ± 0.02
16	0.0625	max	0.71 ± 0.02	0.64 ± 0.06	0.7 ± 0.03	0.42 ± 0.04
16	0.0625	softmax	0.69 ± 0.06	0.56 ± 0.21	0.67 ± 0.08	0.38 ± 0.12
16	0.03125	mean	0.66 ± 0.08	0.5 ± 0.29	0.65 ± 0.09	0.3 ± 0.19
16	0.03125	MOAK	0.71 ± 0.03	0.66 ± 0.04	0.7 ± 0.03	0.41 ± 0.06
16	0.03125	MI	0.71 ± 0.03	0.65 ± 0.05	0.7 ± 0.03	0.41 ± 0.05
16	0.03125	max	0.7 ± 0.03	0.61 ± 0.08	0.68 ± 0.04	0.4 ± 0.06
16	0.03125	softmax	0.66 ± 0.08	0.49 ± 0.29	0.65 ± 0.09	0.3 ± 0.19
16	0.015625	mean	0.65 ± 0.07	0.47 ± 0.28	0.64 ± 0.09	0.29 ± 0.18
16	0.015625	MOAK	0.7 ± 0.02	0.66 ± 0.05	0.7 ± 0.03	0.41 ± 0.05
16	0.015625	MI	0.71 ± 0.03	0.66 ± 0.06	0.7 ± 0.03	0.42 ± 0.06
16	0.015625	max	0.67 ± 0.04	0.56 ± 0.12	0.66 ± 0.05	0.35 ± 0.08
16	0.015625	softmax	0.66 ± 0.07	0.49 ± 0.29	0.64 ± 0.09	0.3 ± 0.19

Table C.8: PK Fraction Results (16 confs.)

Conformer Count	Fraction	Aggregation	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
16	1	mean	0.81 ± 0.01	0.79 ± 0.02	0.81 ± 0.01	0.61 ± 0.02
16	1	MOAK	0.82 ± 0.01	0.8 ± 0.01	0.81 ± 0.01	0.63 ± 0.03
16	1	MI	0.81 ± 0.01	0.78 ± 0.02	0.81 ± 0.01	0.61 ± 0.02
16	1	max	0.83 ± 0.01	0.82 ± 0.01	0.83 ± 0.01	0.66 ± 0.02
16	1	softmax	0.81 ± 0.01	0.79 ± 0.02	0.8 ± 0.01	0.61 ± 0.03
16	0.5	mean	0.79 ± 0.01	0.76 ± 0.03	0.78 ± 0.02	0.58 ± 0.03
16	0.5	MOAK	0.8 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	0.59 ± 0.02
16	0.5	MI	0.78 ± 0.03	0.74 ± 0.06	0.78 ± 0.03	0.57 ± 0.05
16	0.5	max	0.8 ± 0.01	0.78 ± 0.02	0.8 ± 0.01	0.6 ± 0.02
16	0.5	softmax	0.79 ± 0.01	0.76 ± 0.02	0.79 ± 0.01	0.58 ± 0.03
16	0.25	mean	0.76 ± 0.02	0.71 ± 0.04	0.75 ± 0.02	0.51 ± 0.03
16	0.25	MOAK	0.77 ± 0.02	0.74 ± 0.03	0.77 ± 0.02	0.54 ± 0.05
16	0.25	MI	0.76 ± 0.04	0.69 ± 0.11	0.75 ± 0.05	0.52 ± 0.07
16	0.25	max	0.77 ± 0.01	0.74 ± 0.02	0.77 ± 0.01	0.54 ± 0.01
16	0.25	softmax	0.76 ± 0.02	0.71 ± 0.04	0.75 ± 0.02	0.51 ± 0.03
16	0.125	mean	0.74 ± 0.02	0.68 ± 0.05	0.73 ± 0.03	0.48 ± 0.04
16	0.125	MOAK	0.76 ± 0.03	0.71 ± 0.04	0.75 ± 0.03	0.51 ± 0.06
16	0.125	MI	0.73 ± 0.08	0.61 ± 0.23	0.71 ± 0.1	0.45 ± 0.16
16	0.125	max	0.76 ± 0.01	0.71 ± 0.02	0.75 ± 0.01	0.51 ± 0.01
16	0.125	softmax	0.74 ± 0.02	0.68 ± 0.05	0.73 ± 0.02	0.48 ± 0.03
16	0.0625	mean	0.71 ± 0.02	0.61 ± 0.08	0.69 ± 0.03	0.42 ± 0.02
16	0.0625	MOAK	0.7 ± 0.03	0.63 ± 0.07	0.69 ± 0.04	0.39 ± 0.07
16	0.0625	MI	0.72 ± 0.01	0.66 ± 0.03	0.71 ± 0.02	0.44 ± 0.03
16	0.0625	max	0.72 ± 0.01	0.66 ± 0.03	0.71 ± 0.01	0.45 ± 0.02
16	0.0625	softmax	0.71 ± 0.02	0.62 ± 0.07	0.7 ± 0.02	0.43 ± 0.02
16	0.03125	mean	0.66 ± 0.02	0.5 ± 0.08	0.64 ± 0.03	0.33 ± 0.04
16	0.03125	MOAK	0.68 ± 0.04	0.58 ± 0.1	0.67 ± 0.04	0.36 ± 0.07
16	0.03125	MI	0.68 ± 0.04	0.56 ± 0.09	0.66 ± 0.04	0.35 ± 0.07
16	0.03125	max	0.68 ± 0.02	0.58 ± 0.05	0.67 ± 0.02	0.36 ± 0.05
16	0.03125	softmax	0.66 ± 0.02	0.5 ± 0.08	0.64 ± 0.03	0.33 ± 0.04
16	0.015625	mean	0.63 ± 0.04	0.45 ± 0.13	0.61 ± 0.05	0.28 ± 0.08
16	0.015625	MOAK	0.64 ± 0.04	0.55 ± 0.15	0.63 ± 0.05	0.29 ± 0.09
16	0.015625	MI	0.64 ± 0.04	0.55 ± 0.15	0.63 ± 0.05	0.29 ± 0.09
16	0.015625	max	0.64 ± 0.03	0.52 ± 0.12	0.63 ± 0.04	0.29 ± 0.08
16	0.015625	softmax	0.64 ± 0.03	0.46 ± 0.13	0.62 ± 0.04	0.29 ± 0.07

Table C.9: ECFP Fraction Results

Conformer Count	Fraction	Accuracy (mean ± std)	F1 (mean ± std)	ROC AUC (mean ± std)	MCC (mean ± std)
1	1	0.82 ± 0.04	0.8 ± 0.04	0.81 ± 0.03	0.63 ± 0.07
1	0.5	0.79 ± 0.02	0.77 ± 0.01	0.79 ± 0.01	0.58 ± 0.03
1	0.25	0.78 ± 0.03	0.75 ± 0.04	0.77 ± 0.03	0.55 ± 0.07
1	0.125	0.74 ± 0.04	0.71 ± 0.04	0.74 ± 0.04	0.48 ± 0.07
1	0.0625	0.7 ± 0.03	0.64 ± 0.04	0.69 ± 0.03	0.39 ± 0.07
1	0.03125	0.68 ± 0.04	0.6 ± 0.06	0.67 ± 0.04	0.35 ± 0.07
1	0.015625	0.64 ± 0.03	0.6 ± 0.05	0.64 ± 0.03	0.28 ± 0.06

Appendix D

List of Abbreviations

2D Two-dimensional

3D Three-dimensional

CGT Chemical Graph Theory

CDPKit Chemical Data Processing Kit

ECFP Extended Connectivity Fingerprints

GPU Graphics Processing Unit

HTS High-Throughput Screening

MIL Multi-Instance Learning

MCC Matthews Correlation Coefficient

MOAK Most Optimal Assignment Kernel

MI Multi-Instance Kernel

p.s.d. Positive Semi-Definite

PK Pharmacophore Kernel

RBF Radial Basis Function

RDF Radial Distribution Function

ROC-AUC Receiver Operating Characteristic – Area Under the Curve

CSR Compressed Sparse Row

SVM Support Vector Machine

SVC Support Vector Classifier

SHA-1 Secure Hash Algorithm 1

SMILES Simplified Molecular Input Line Entry System

SMK Subgraph Matching Kernel

TGT Typed Graph Triangles

VS Code Visual Studio Code

CPU Central Processing Unit