Imperial College London

Department of Computing

Topological Graph Learning with Attention for Sparse Binding Affinity Prediction on BELKA

Author:
Guo Jing Yang

Supervisor:
Dr Tolga Birdal

Submitted in partial fulfillment of the requirements for the MSc degree in Artificial Intelligence of Imperial College London

September 2025

Abstract

We investigate ligand-only, *topological* graph neural networks for binding affinity ranking on BELKA, a large scaffold-shifted benchmark spanning BRD4, HSA, and sEH. Our backbone is a GPS-CC (*General, Powerful, Scalable Graph Transformer on Cellular Complexes*) architecture, a **hybrid MPNN + Transformer** design in which local message passing is interleaved with (optionally biased) global attention, instantiated on a cell complex so that, beyond node–edge updates, the model can perform higher-order message passing over 2-cells (ring "faces"). The backbone supports multi-scale positional structure via **Laplacian eigenvector encodings** and **random-walk structural encodings**, with optional **barycentric subdivision** features that ground faces in a topological hierarchy.

Under capacity-controlled ablations, the best-performing model is a *plain node–edge MPNN* equipped with Laplacian/RW encodings and trained with **Asymmetric Loss (ASL)**. On a 1M subset it attains $\mathrm{MAP} = 0.21207 \pm 0.01291$, improving to 0.24088 ± 0.00329 on 10M molecules. Scaling gains are explained by increased scaffold coverage, while fingerprint-level similarity to the full 98M pool is already saturated, suggesting diminishing but real returns at full scale.

A central contribution is a *loss-function study* tailored to BELKA's extreme label sparsity. We compare balanced BCE, focal loss, ASL, and ASL+center loss. ASL (via asymmetric focusing (γ^+, γ^-) and a negative margin m) selectively down-weights easy negatives while preserving gradients on rare positives, yielding the strongest MAP and the most stable training; ASL+center offers partial cohesion of positives but underperforms ASL, and focal/BCE trail further. We also observe that attention-enabled variants require a lower learning rate to avoid numerical instability, consistent with their poorer optimisation dynamics.

Contrary to common practice, **global attention** and **higher-order (face/barycentric)** channels *reduce* MAP. A targeted diagnostic suite: entropy trajectories, long-range attention ratios, distance–attention correlations, and logit/bias scale logging shows that (i) dense attention creates non-physical "virtual edges" that blur local chemical cues already captured by Laplacian/RW encodings; (ii) QK magnitudes overwhelm bounded SPD biases, producing unstable softmax regimes and weak structural grounding; and (iii) face features act as shortcut-like signals (reliance revealed by α-gating) yet are noise-insensitive (σ-sweeps), failing under scaffold shift. Bucket-wise analysis confirms strong performance in *share* and near-zero MAP in *kin0*, underscoring that chemically grounded locality with ASL-driven optimisation outperforms unconstrained global mixing on BELKA.

Finally, despite being trained on only $\sim 1\%$–10% of the data, our models achieve MAP scores of 0.212–0.241, already competitive with some of the most stable BELKA Kaggle submissions (0.277–0.286) and within reach of the private leaderboard winner (0.306). This trajectory suggests that scaling to the full dataset could yield state-of-the-art performance while preserving robustness, highlighting the importance of scaffold coverage and representation control.

Acknowledgments

I would like to thank my supervisor, Dr Tolga Birdal, my second marker, Dr Pedro Mediano, my coursemates and members of the Imperial CIRCLE group for their invaluable guidance, encouragement, and constructive feedback throughout this project. I am also grateful to the Imperial College Research Computing Service and Imperial College Department of Computing for providing access to the HPC and GPU clusters that enabled large-scale experimentation. Special thanks go to the maintainers of open-source libraries such as PyTorch, PyTorch Geometric, RDKit, and Weights & Biases, which formed the backbone of this work. Finally, I acknowledge Leash-Bio (provider of the BELKA dataset), Polaris (where BELKA is hosted) and the Kaggle community for providing a challenging benchmark that motivated much of this research.

Contents

Chapter 1

Introduction

Predicting **protein–ligand interactions (PLIs)** is a central challenge in computational drug discovery, as it underpins both the identification of new therapeutics and the repurposing of existing drugs. Accurate prediction of binding poses and affinities enables *virtual screening*: the computational prioritisation of candidate molecules for experimental validation, thereby accelerating early-stage drug development and reducing associated costs [1, 2]. Traditional approaches to PLIs have relied on **molecular docking**, where candidate ligands are sampled into the protein's binding pocket and scored using empirical or physics-based energy functions [3–5]. Classical docking pipelines are mechanistically interpretable, but their performance is hindered by limited scoring accuracy, rigid protein models that poorly capture induced fit, and high computational cost in large-scale virtual screening [6, 7]. More recent docking frameworks, such as DiffDock and FABind, retain the docking paradigm but replace the sampling or scoring modules with deep learning components (e.g. generative diffusion models for pose generation or graph neural networks for ligand–pocket interaction modelling), substantially improving accuracy while mitigating some of the computational burden [8, 9].

Recent advances in artificial intelligence have begun to transform this landscape. A comprehensive review by Sim et al. [10] highlights progress across four pillars of PLI modelling: ligand binding site prediction, binding pose estimation, scoring function development, and large-scale virtual screening. AI models now integrate sequence embeddings, geometric deep learning, and generative diffusion techniques, markedly improving efficiency and accuracy compared with classical docking methods. Despite these gains, generalising across chemically and structurally diverse protein–ligand pairs remains a significant challenge.

Several AI-driven models exemplify the field's trajectory. Interformer employs an interaction-aware mixture density network within a Graph–Transformer framework to explicitly model non-covalent interactions, improving pose accuracy and affinity prediction [11]. DeepRLI formulates PLI prediction as a multi-objective problem, jointly optimising docking, scoring, and screening with a graph transformer backbone [12]. PIGNet2 combines a physics-informed graph neural network with a tailored data augmentation strategy. This design integrates physics-based inductive bias (empirical scoring) with machine learning flexibility, helping the model generalise more effectively to diverse protein–ligand interactions [13, 14]. Meanwhile, generative methods such as FlowDock employ geometric flow matching to model protein flexibility, achieving higher success rates than single-sequence AlphaFold 3 in blind docking benchmarks and placing among the leading approaches in the CASP16 ligand docking category [15]. Advances in large-scale foundation models reinforce these developments: in particular, AlphaFold 3 extends beyond pure protein folding to accurately model complexes with ligands, nucleic acids, small molecules, and ions through its transformer-based Pairformer and diffusion refinement modules. This marked improvement in protein–ligand structure prediction highlights that the vast diversity of chemical space can be addressed within a unified deep learning framework, eliminating the need to separate protein folding from ligand docking artificially [16].

Overall, the current landscape of PLI prediction is characterised by the convergence of traditional structure-based approaches with modern deep learning. Graph neural networks, transformers, and diffusion-based generative models are redefining how binding affinity and pose are estimated, with clear implications for virtual screening pipelines. Nevertheless, improving generalisation across chemical space remains a pressing research direction, motivating further exploration of architectures that combine scalability with structural awareness[10].

Alongside these structure-based approaches, there has been rapid progress in structure-agnostic drug–target interaction (DTI) models. Unlike docking, which depends on experimentally resolved or predicted 3D protein structures, these structure-agnostic methods operate on one- or two-dimensional representations such as SMILES strings, amino acid sequences, or residue–residue contact maps. The advantage of these methods lies in their scalability: sequence repositories like UniProt and large-scale bioactivity databases such as BindingDB provide millions of ligand–target measurements, enabling the training of high-capacity machine learning models at scale [2, 17–20].

Sequence-based DTI models treat SMILES strings and amino-acid chains as text. Convolutional or recurrent networks, such as DeepDTA's dual-CNN pipeline [17], learn local patterns in sequences, while transformer architectures like TransformerCPI [18] capture long-range dependencies via self-attention. These methods train quickly on large datasets and scale effectively, but may miss the topological nuances of molecular graphs or discontinuous binding motifs in proteins.

The second paradigm uses **graph-based models**: ligands are represented as molecular graphs (atoms = nodes, bonds = edges), and proteins are encoded as 2D contact or distance matrices describing residue–residue interactions [21, 22]. The contact matrix is often derived from predicted protein structures, providing a coarse abstraction of 3D geometry [1]. Graph neural networks (GNNs), including message passing neural networks (MPNNs), graph convolutional networks (GCNs), and graph attention networks (GATs), embed ligand and protein graphs separately before combining them via cross-graph interaction modules. Models such as GraphDTA have demonstrated improved affinity prediction by explicitly capturing molecular topology [19]. A flagship example is AttentionSiteDTI, which frames each predicted binding site as a separate graph and employs a self-attention mechanism to highlight which pocket–atom interactions drive binding affinity. This yields both interpretability and high predictive accuracy without explicit docking [1].

Despite these advances, the relative strengths and weaknesses of sequence- and graph-based paradigms remain unresolved. Transformers excel at modelling long-range dependencies and can scale well in practice, but may underexploit explicit topological structure; GNNs are naturally suited for encoding molecular graphs yet often lag in throughput and optimisation ease compared to attention- or convolution-based sequence models [23, 24]. This tension is particularly relevant in light of the recent "Predict New Medicines with BELKA" Kaggle competition (2024), which introduced the largest publicly available DTI dataset to date: 133 million ligand–protein binding measurements across three targets. Notably, a *majority* of prize-winning solutions employed **sequence-based 1D CNN encoders** over SMILES sequences, sometimes paired with lightweight attention or MLP heads, underscoring the efficiency and scalability of convolutional pipelines for ultra-large-scale training and inference [25]. At the same time, these solutions paid little attention to explicit topological structure, raising questions about whether combining scalable sequence encoders with richer graph or topological representations could yield further gains.

1.1 Motivation

Large-scale datasets such as BELKA have revealed both the strengths and limitations of current drug–target interaction (DTI) models. Sequence-based approaches, particularly convolutional pipelines over SMILES, excel in efficiency and have scaled to hundreds of millions of samples. Yet their reliance on statistical correlations raises concerns about generalisation and their ability to capture the structural principles of binding. Conversely, more structurally grounded methods, such as graph neural networks, encode molecular topology explicitly, but often fail to train reliably at the scale demanded by BELKA.

This tension motivates the central question of this thesis: how can we combine the scalability of sequence encoders with the structural fidelity of topology-aware methods? Recent advances in topological deep learning (TDL) extend molecular graphs to higher-order representations, capturing rings, cavities, and other motifs neglected in atom–bond graphs. When paired with transformer modules, which are adept at modelling long-range dependencies, these models offer a promising route to scalable yet structurally meaningful DTI prediction.

Our aim is not only to pursue higher accuracy but also to develop architectures that are computationally efficient, interpretable, and capable of generalising across diverse regions of chemical space. BELKA provides a demanding test case: it requires throughput on millions of samples while retaining sensitivity to the structural motifs that underlie binding specificity.

1.2 Contributions

This project makes the following contributions:

1. **Hybrid architectures.** We design models integrating topological neural networks with transformer layers, enabling joint capture of higher-order motifs and long-range dependencies.

2. **Benchmarking at scale.** We provide the first systematic comparison of GNN, TNN, hybrid transformer-GNN and hybrid transformer-TNN architectures on BELKA, assessing accuracy, generalisation, and throughput.

3. **Topology-aware features.** We combined molecular descriptors that enrich standard atom-bond representations, including barycentric subdivisions, ring encodings, and structural positional embeddings.

4. **Scalability and interpretability analysis.** We evaluate training efficiency at the 10M scale and perform topology-aware attention diagnostics, including entropy, long-range attention ratios, and Pearson correlation with structural distances, to explain predictions in chemically meaningful terms.

5. **Ablation and dataset similarity studies.** We design controlled ablations comparing models with and without higher-order cell representations, as well as with and without barycentric subdivisions, to isolate the impact of these topological features. In addition, we analyse scaffold overlap and fingerprint similarity between the test subset and the full dataset to assess distributional shifts and validate generalisation.

6. **Reproducible pipeline** We release a modular PyTorch/PyG framework tailored for HPC-like environments, ensuring reproducibility and extensibility for future work.

1.3 Report Structure

Chapter 2 introduces the terminology, background concepts, dataset, evaluation metrics, and mathematical notation that underpin the report. Chapter 3 then details the preprocessing pipeline and feature engineering applied to the input data, followed by Chapter 4, which describes the model architecture. Chapter 5 outlines the different loss functions employed, while Chapter 6 presents the methodology, training configurations, optimisation strategies, and scoring metrics. Chapter 7 reports and evaluates the experimental results, and finally, Chapter 8 concludes the report and highlights the limitations and possible directions for future work.

Chapter 2

Background

2.1 Terminology

2.1.1 Graph Neural Network (GNNs)

Graph Neural Network (GNNs) are a class of deep learning architectures specifically tailored for graph-structured data, where entities and their relations are represented as nodes and edges, respectively. At the core of a GNN is the Message Passing Neural Networks (MPNN) framework. The forward pass consists of two phases. In each iteration t, node features are updated by aggregating information from neighbouring nodes and edges through learnable message functions, M_t, followed by node update functions, U_t, that integrate the messages into new node embeddings [26]. This can be represented in mathematical form

$$m_v^{t+1} = \sum_{w \in N(v)} m_{wv}^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{wv}, e_{vw}) \tag{2.1}$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{2.2}$$

where m_v^{t+1} is the message received by node v at time $t+1$, h_v^t denotes the hidden state of node v at time t, e_{wv} denotes the feature of the directed edge from w to v, $N(v)$ denotes the neighbour of v in graph $\mathcal{G}$. A final readout function, R, then pools node-level information according to Eq. (2.3) to generate graph-level representations, $\hat{y}$, for downstream tasks [27].

$$\hat{y} = R(\{h_v^T \mid v \in \mathcal{G}\}). \tag{2.3}$$

The message function, M_t, update function U_t and readout function R are all learned differentiable functions. R operates on the set of node features and must be permutation invariant for the MPNN to be invariant to graph isomorphism. One could also learn edge features by introducing hidden states for all edges h_{vw}^t in the graph and updating them according to Eq. (2.1) and Eq. (2.2).

Variants such as graph convolutional networks (GCNs), which generalise spectral convolution to graphs, graph attention networks (GATs) [28], which weight neighbour contributions via self-attention, and graph recurrent networks (GRNs), which use recurrent units for message integration, have all demonstrated superior performance across domains [29]. These architectures have become the de facto choice for representing complex relational structures, including social networks, knowledge graphs, and molecular graphs, where they excel at capturing topology and feature interactions.

In structure-agnostic drug discovery, accurate prediction of ligand molecular properties, such as electronic, geometric, physicochemical, and even quantum mechanical attributes [30], is crucial for prioritising and optimising candidate compounds prior to experimental validation. In this setting, GNNs typically represent molecules as undirected graphs, with nodes encoding atomic types and edges encoding bond types, without requiring explicit knowledge of the protein target. Additional features such as 3D coordinates or interatomic distances may be incorporated if available, but are not mandatory. Hybrid architectures like GPS++ combine message-passing neural networks (MPNNs) with global Transformer-based attention layers, allowing the model to capture both local chemical environments and long-range molecular interactions in a target-agnostic manner.

Equivariant GNNs. Molecular systems are subject to the symmetries of 3D Euclidean space; their properties should be independent of their absolute position and orientation (translation and rotation invariance), or transform predictably with these operations (equivariance). Mathematically, equivariance is defined as follows. Let $T_a : X \to X$ denote a family of transformations on X indexed by elements $a \in A$. A function $\phi : X \to Y$ is said to be *equivariant* with respect to g if there exists a corresponding transformation $S_a : Y \to Y$ on the output space such that

$$\phi(T_a(x)) \;=\; S_a\big(\phi(x)\big). \tag{2.4}$$

A SE(3)-equivariant GNNs are designed to have translation, rotation and permutation equivariance with respect to an input set of points, while an E(3)-equivariant GNN has all the equivariance properties of SE(3) plus an additional reflection equivariance [31–33]. A GNN can be turned into an EGNN by modifying the message function, M_t and updating the node's coordinates as a vector field in a radial direction (i.e. weighted sum). Specifically, Eq. (2.1) becomes

$$m_v^{t+1} = \sum_{w \in N(v)} m_{vw}^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}, e_{wv}, \left\|\mathbf{x}_v^t - \mathbf{x}_w^t\right\|^2) \tag{2.5}$$

where $\mathbf{x}_v^t$ is the n-dimensional coordinates of the node v at time t. The position of each node is updated using

$$\mathbf{x}_v^{t+1} = \mathbf{x}_v^t + C \sum_{w \neq v} (\mathbf{x}_v^t - \mathbf{x}_w^t)\, X_t(m_{vw}^t) \tag{2.6}$$

where $\mathbf{x}_v^t$ is the position of node v at time t and X_t is a weights function that output the importance weight of m_{vw}^t (a scalar). C is often chosen to be $\frac{1}{N-1}$, where N is the total number of nodes.

In structure-agnostic drug discovery, equivariance and invariance remain critical for reliable molecular property prediction, even without explicit protein structural information. If a molecule is rotated or translated, the GNN's output (e.g. a predicted property or a learned molecular representation) should transform consistently (for equivariant properties) or remain unchanged (for invariant properties). This ensures that predictions do not depend on arbitrary coordinate choices, which is essential for learning robust chemical representations from large ligand libraries. Models such as equivariant GNNs extend traditional message-passing by incorporating symmetry-aware updates that respect the Euclidean group, allowing them to capture 3D geometric and physicochemical properties in a physically meaningful way. While structure-based approaches like FABind exploit equivariance to model protein–ligand binding [9], in the structure-agnostic setting, these models focus on learning accurate ligand embeddings that are robust to isometric transformations, enabling downstream tasks such as property prediction, virtual screening, or docking pre-filtering. By enforcing equivariance and invariance, structure-agnostic GNNs provide a principled framework for modelling molecular geometry without relying on protein context, thereby improving generalisation across diverse chemical space.

2.1.2 Topological Deep Learning

Topological Deep Learning (TDL) extends traditional deep learning by operating on data endowed with rich, non-Euclidean structures, such as graphs, simplicial complexes, and CW (cell) complexes, rather than just regular grids or sequences. Central to TDL are Topological Neural Networks (TNNs), which generalise message-passing schemes from Graph Neural Networks (GNNs) to higher-order domains, allowing the model to aggregate information not only over nodes and edges but also over triangles, tetrahedra, and beyond. These architectures leverage algebraic topology tools (e.g., cochains, chain complexes) to embed multi-scale relational information directly into the learning process, yielding representations sensitive to the data's global and local topological features [34, 35].

In drug–target interaction (DTI) tasks, capturing subtle, long-range dependencies, such as ring structures, binding pockets, and non-local electrostatic interactions, is crucial. Geometric Deep Learning (GDL), which typically models molecules as graphs of atoms linked by bonds, is inherently limited to pairwise edges and often requires deep, stacked layers (or heuristic graph rewiring) to propagate signals across distant parts of the molecule. TDL, by contrast, natively encodes higher-order relational structure through cells of dimension ≥ 2 (e.g., loops for aromatic rings), enabling direct message passing across these motifs. For instance, the Cellular Transformer (CT) demonstrates consistent performance gains on MoleculeNet benchmarks by attending over aggregated cell-level features (e.g. rings, loops, fused motifs) without ad-hoc rewiring or virtual nodes, thereby capturing global interactions vital to binding affinity prediction [36–38]. Similarly, CW Networks lifts molecular graphs into CW complexes to model induced cycles, achieving state-of-the-art expressivity and compressing long-distance dependencies that GNNs struggle with [39].

2.1.3 Cellular Complex

The domains of TDL generalise the pairwise relations of graphs to part-whole and set-type relations that permit the representation of more complex relational structures. **Cellular Complexes** extend graphs and simplicial complexes by capturing higher-order part–whole relationships through the hierarchical construction of *cells*. In this framework, nodes are regarded as *rank-0 cells*, which can be joined to form edges. These edges (*rank-1 cells*) can in turn be combined into faces, while faces (*rank-2 cells*) may be assembled into volumes (*rank-3 cells*), and so forth [34]. A regular 2-dimensional cellular complex can capture ring structures in molecules such as benzene rings.

The part-whole structure of cellular complexes induces a dependency between the rank of a cell and its cardinality: a cell of rank r must contain at least $r + 1$ cells of rank $r - 1$. For instance, a face ($r = 2$) contains at least three edges ($r - 1 = 1$). Let $\mathcal{X}$ denote the domain (topological space) of a cellular complex. The *k-skeleton* of $\mathcal{X}$, written $\mathcal{X}^{(k)}$, is the subcomplex formed by all cells of dimension at most k. We represent the hidden state of a cell $x \in \mathcal{X}$ at layer (time step) t as $\mathbf{h}_x^{t,(r)}$, where r specifies the rank of x [34].

To define the MPNN framework of a Topological Neural Network (TNN), we need to construct the neighbourhood of a cell. The boundary relation defines the neighbourhood structure.

Definition 1 *The* ***boundary relation*** *is defined by $\sigma \prec \tau$ whenever $\sigma < \tau$ and there is no intermediate cell δ satisfying $\sigma < \delta < \tau$ [39].*

This relation allows us to characterise four types of local adjacencies in cell complexes, which serve as the fundamental components of our message-passing framework.

Definition 2 *(Cell complex adjacencies). Let $\mathcal{X}$ be a cell complex and $\sigma \in P_{\mathcal{X}}$ a cell, where $P_{\mathcal{X}}$ is the indexing set equipped with a poset structure (i.e., $\tau \leq \sigma \iff \mathcal{X}_\tau \subseteq \overline{\mathcal{X}_\sigma}$). Following [39], we define:*

1. ***Boundary adjacencies:*** *$\mathcal{B}(\sigma) = \{\tau \mid \tau \prec \sigma\}$. These are lower-dimensional cells lying on the boundary of σ. For example, the boundary cells of an edge are its vertices.*
2. ***Co-boundary adjacencies:*** *$\mathcal{C}(\sigma) = \{\tau \mid \sigma \prec \tau\}$. These are higher-dimensional cells for which σ is part of their boundary. For example, the co-boundary cells of a vertex are the edges incident to it.*
3. ***Lower adjacencies:*** *$\mathcal{N}_\downarrow(\sigma) = \{\tau \mid \exists\delta \text{ such that } \delta \prec \sigma \text{ and } \delta \prec \tau\}$. These are cells of the same dimension as σ that share a lower-dimensional boundary cell. For example, two edges are lower adjacent if they meet at a common endpoint.*
4. ***Upper adjacencies:*** *$\mathcal{N}_\uparrow(\sigma) = \{\tau \mid \exists\delta \text{ such that } \sigma \prec \delta \text{ and } \tau \prec \delta\}$. These are cells of the same dimension as σ that both lie on the boundary of a common higher-dimensional cell. For example, two edges are upper adjacent if they form sides of the same face.*

Boundary relations in a cell complex can be represented using incidence matrices, denoted by $\mathbf{B}$. In particular, the matrix $\mathbf{B}_r$ encodes which rank-$(r-1)$ cells bound each rank-r cell. Formally, $\mathbf{B}_r$ is an $n_{r-1} \times n_r$ matrix, where n_r is the number of rank-r cells for $r \geq 1$, and is defined by

$$(\mathbf{B}_r)_{i,j} = \begin{cases} \pm 1 & \text{if } x_i^{(r-1)} \prec x_j^{(r)}, \\ 0 & \text{otherwise,} \end{cases} \tag{2.7}$$

where $x_i^{(r-1)}$ and $x_j^{(r)}$ are cells of ranks $r-1$ and r, respectively. The ± 1 sign encodes the orientation associated with the boundary relation, as required in cellular complexes [34].

Incidence matrices provide a convenient way to encode the four neighbourhood structures described in Def. 2. For instance, the boundary adjacency matrix of rank 1 is given by $\mathbf{B}_1$, whose (i, j) entry is non-zero whenever the j^{th} edge is incident to the i^{th} node. The corresponding co-boundary adjacency matrix is simply the transpose of the boundary matrix, e.g. $\mathbf{B}_1^T$ for rank 1 [36].

In this framework, the usual graph Laplacian is written as $\mathbf{L}_{\uparrow,0}$ and is defined by $\mathbb{H}_0 = \mathbf{L}_{\uparrow,0} = \mathbf{B}_1\mathbf{B}_1^T$. Its higher-order generalisation is the r-Hodge Laplacian,

$$\mathbb{H}_r = \mathbf{L}_{\downarrow,r} + \mathbf{L}_{\uparrow,r}, \tag{2.8}$$

where the *lower Laplacian* $\mathbf{L}_{\downarrow,r} = \mathbf{B}_r^T\mathbf{B}_r$ encodes lower adjacencies of rank-r cells, and the *upper Laplacian* $\mathbf{L}_{\uparrow,r} = \mathbf{B}_{r+1}\mathbf{B}_{r+1}^T$ encodes their upper adjacencies. For the K-skeleton, the $K-$Hodge Laplacian reduces to $\mathbb{H}_K = \mathbf{L}_{\downarrow,K}$ since rank-K cells can only be lower adjacent. Similarly, the graph Laplacian $\mathbb{H}_0$ contains only the upper term, because vertices can only be upper adjacent [40].

The degree matrix generalises accordingly into lower and upper variants: $\mathbf{D}_{\downarrow,r} = \mathrm{diag}(\mathbf{B}_r^T\mathbf{B}_r)$, $\mathbf{D}_{\uparrow,r} = \mathrm{diag}(\mathbf{B}_{r+1}\mathbf{B}_{r+1}^T)$, with the standard graph degree matrix corresponding to $\mathbf{D}_{\uparrow,0}$.

Finally, we define the *lower* and *upper* adjacency matrices of rank r as

$$\mathbf{A}_{\downarrow,r} = \mathbf{D}_{\downarrow,r} - \mathbf{L}_{\downarrow,r}, \qquad \mathbf{A}_{\uparrow,r} = \mathbf{D}_{\uparrow,r} - \mathbf{L}_{\uparrow,r}. \tag{2.9}$$

For example, $(\mathbf{A}_{\downarrow,1})_{i,j}$ is non-zero if the i^{th} and j^{th} edges share a common node, while $(\mathbf{A}_{\uparrow,0})_{i,j}$ (the standard graph adjacency matrix) is non-zero if the i^{th} and j^{th} nodes are connected by an edge [36].

2.1.4 Barycentric Subdivision

Barycentric subdivision refines a simplicial complex Δ without altering its underlying topological space. An *abstract simplicial complex* Δ on a finite vertex set V is a nonempty collection of subsets of V such that

- $\{v\} \in \Delta$ for every $v \in V$,
- if $G \in \Delta$ and $F \subseteq G$, then $F \in \Delta$.

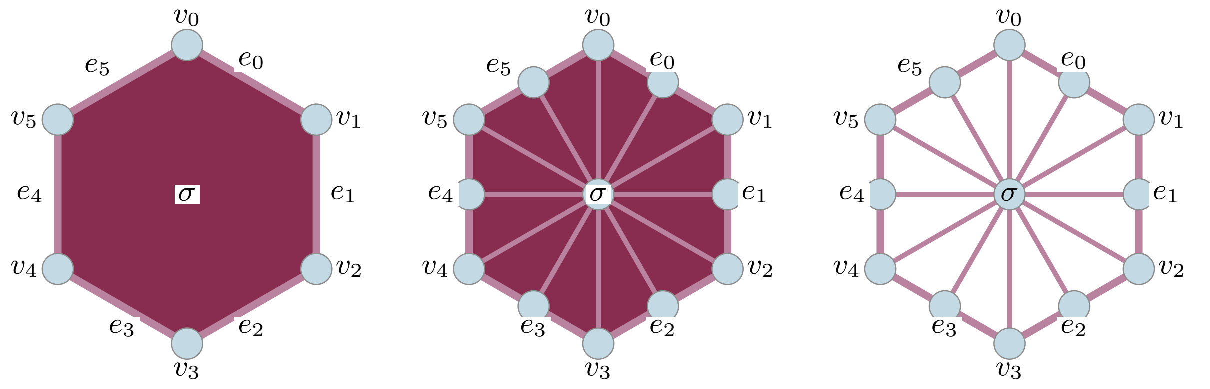
The elements of Δ are called *faces* (or simplices), and the maximal faces are referred to as *facets*. Note that here “faces” denote simplices of any dimension, in contrast to our earlier usage where “face” specifically meant a 2-cell. To avoid confusion, outside this subsection, the term “face” will always refer to a 2-cell. For our purposes, abstract simplicial complexes and simplicial complexes can be treated interchangeably. A formal justification is given in [41].

Two key ingredients are the **face poset** and the **order complex**. The face poset $P(\Delta)$ is the set of (typically nonempty) faces of Δ, ordered by inclusion $\subseteq$. Given a poset P, the order complex $\Delta(P)$ is the simplicial complex whose vertices are elements of P, with faces corresponding to chains (totally ordered subsets) of P. Barycentric subdivision is then defined as

$$\mathrm{Sd}(\Delta) = \Delta\big(P(\Delta)\big). \tag{2.10}$$

Concretely, this means: list all faces of Δ (which become the vertices of $\mathrm{Sd}(\Delta)$); then, for every chain of $k+1$ nested faces $F_0 \subset \cdots \subset F_k$, include one k-simplex in $\mathrm{Sd}(\Delta)$ [42].

Geometrically, the same subdivision can be realised by placing a new vertex at the **barycenter** of each face of Δ, and then forming convex hulls along chains of nested faces. Under this construction, an n-simplex subdivides into $(n+1)!$ smaller n-simplices [43]. The geometric realisations $|\Delta|$ and $|\mathrm{Sd}(\Delta)|$ are homeomorphic, so barycentric subdivision preserves the homotopy type of the complex while producing a finer, more regular mesh [41]. The process can be iterated to improve mesh regularity or to facilitate combinatorial arguments. As a simple example, for a single triangle, the face poset consists of 3 vertices, 3 edges, and 1 triangle; chains of the form $v \subset e \subset t$ yield six smaller triangles that exactly tile the original. Fig. 2.1 illustrates this construction for a benzene ring, together with the corresponding 1-skeleton barycentric subdivision graph.



(a) Cellular complex version of the benzene ring

(b) Barycentric subdivision of benzene ring.

(c) 1-skeleton barycentric subdivision graph of benzene ring.

Figure 2.1: Comparison of the original cellular complex, its barycentric subdivision, and the 1-skeleton barycentric subdivision. Each original cell of Δ is represented as a node in the barycentric subdivision.

2.1.5 Bemis–Murcko scaffolds

A Bemis–Murcko scaffold is a widely used representation of the **core** structure of a molecule. It is obtained by stripping away side chains and retaining only the ring systems and the linker framework that holds them together. This abstraction reduces a complex molecule to its fundamental chemotype, allowing comparisons at the level of broad structural families rather than fine substituents. Scaffold analysis is valuable because many medicinal chemistry tasks, such as scaffold hopping or scaffold-based splits in benchmarks, hinge on whether models generalise to unseen core structures [44].

2.1.6 Morgan fingerprints

A Morgan fingerprint (often referred to as ECFP, Extended Connectivity Fingerprint) is a circular molecular fingerprint that encodes the local atomic environment around each atom up to a given radius. The algorithm iteratively hashes atom neighbourhoods into fixed-length bit vectors, producing a compact binary representation of molecular structure. Morgan fingerprints are robust, interpretable, and computationally efficient, making them a standard choice for similarity searching, clustering, and as input features for machine learning models [45].

2.2 Dataset

The Big Encoded Library for Chemical Assessment (BELKA) is a massive public dataset released by Leash Biosciences for the NeurIPS 2024 “Predict New Medicines with BELKA” competition on Kaggle. BELKA captures binary binding labels for roughly 100 million small molecules (down from 133 million after splitting) screened against three protein targets (BRD4, EPHX2/sEH, and ALB/HSA) using DNA-encoded chemical library (DEL) technology. In total, BELKA encompasses on the order of 4.25 billion raw sequencing measurements across multiple selection rounds and replicates, dwarfing prior public benchmarks such as bindingdb ($\approx$2.8 million measurements) by a factor of $\approx$1,000 [46, 47].

BELKA’s experimental pipeline relies on combinatorial DEL synthesis followed by iterative affinity selections. For the main triazine-based library (AMA014), three rounds of selection were run in triplicate per target; an orthogonal “kinase0” library underwent a single round in duplicate. After each round, bound molecules are eluted and sequenced, producing raw read counts that are normalised to counts per billion (cpb):

$$cpb = \frac{raw\ counts}{total\ reads} \times 10^9$$

This normalisation facilitates comparison across experiments.

2.2.1 Library Design and Dataset Statistics

BELKA combines two DEL sub-libraries:

Table 2.1: Summary of Library Screening Data

Library	Chemistry Core	Rounds × Replicates	Total Raw Reads
AMA014	Triazine	3 rounds × 3 replicates	∼3.6 B
kinase0	Heterocycle	1 round × 2 replicates	∼0.6 B

After each selection round, bound molecules are eluted and sequenced. The three-round, triplicate design for AMA014 enhances enrichment signal at the cost of increased sequencing volume; kinase0 serves as an orthogonal test of out-of-distribution generalisation [25].

Table 2.2: Core Dataset Statistics

Statistic	Value
Training molecule examples	98 M per protein
Validation molecule examples	200K per protein
Test molecule examples	360K per protein
Positive binder ratio	∼0.5 %
Number of atoms per molecule	(min 20, average 41, max 72)
Number of bonds per molecule	(min 21, average 45, max 83)
Number of rings per molecule	(min 1, average 5, max 14)

2.2.2 Molecular Representations

The structure of a single row in the BELKA dataset is summarised in Tab. 2.3. The *molecule score* quantifies the binding affinity of a molecule to the target protein. It is computed from the enrichment of counts per molecule relative to a background condition, with scores above a cutoff threshold indicating binding. The `split_group` column contains three distinct values:

- *share* — molecules that share both the triazine core and building blocks across the training, validation, and test sets.
- *non-share* — molecules that share the triazine core but not the same building blocks with the training set; this group appears only in the validation and test sets.
- *kin0* — molecules derived from a different DNA-encoded library, with different cores, building blocks, and attachment chemistries compared to the training/validation sets. This group is present only in the test set and is used to evaluate true out-of-distribution generalisation [46].

Table 2.3: BELKA Dataset Column Descriptions

Column	**Description**	**Data Type**
`molecule_smiles`	Full-molecule SMILES string	str
`buildingblock1_smiles`	SMILES of the first attachment block	str
`buildingblock2_smiles`	SMILES of the second attachment block	str
`buildingblock3_smiles`	SMILES of the third attachment block	str
`molecule_score_HSA`	score measuring binding affinity to ALB/HSA	float32
`molecule_score_sEH`	score measuring binding affinity to EPHX2/sEH	float32
`molecule_score_BRD4`	score measuring binding affinity to BRD4	float32
`binds_HSA`	Binary indicator of binding to ALB/HSA	float16
`binds_sEH`	Binary indicator of binding to EPHX2/sEH	float16
`binds_BRD4`	Binary indicator of binding to BRD4	float16
`split`	train or val or test set	string
`split_group`	share or non-share or kin0	string

With its unprecedented scale and fully open access, BELKA provides a rigorous benchmark for assessing graph neural networks, topological neural networks, and hybrid architectures, while also serving as a testbed for exploring the scalability and structure-awareness of models in drug discovery [25, 46].

2.2.3 SMILES format

SMILES (Simplified Molecular Input Line Entry System) is a compact ASCII notation, introduced by David Weininger in 1988, for encoding molecular graphs as linear strings: atoms are denoted by element symbols (e.g., C, N, O), bonds by characters such as "=" or "#", branches by parentheses, and ring closures by numeric labels (e.g., benzene as "c1ccccc1") [48]. By reducing 2D or 3D molecular structures to plain text, SMILES enables highly efficient storage, transmission, and parsing by cheminformatics toolkits (e.g., RDKit, Open Babel), supporting rapid substructure searches, descriptor generation, and coordinate reconstruction [49].

A key feature is *canonicalization*: deterministic algorithms produce a unique SMILES string per molecule, facilitating duplication checks and database indexing. The notation is also extensible, encoding stereochemistry ("@" for chiral centres), isotopes (e.g., "[13C]"), formal charges, and aromaticity, ensuring unambiguous representation of rich structural details [49].

2.3 Cellular complex representation

Each small molecule can be represented as a cellular complex $\mathcal{C}$ of arbitrary rank. From cellular homology, however, we know that $H_k(X^n) = 0 \quad \forall k > \dim(X)$ if X is embedded in $\mathbb{R}^d$, where X is a CW complex and $H_k(X)$ denotes the k^{th} homology group. Intuitively, a CW complex (or cell complex) is a topological space obtained by glueing cells together, and $H_k(X)$ measures the presence of *k-dimensional holes* in X [50]. Since small molecules live in $\mathbb{R}^3$, it follows that $H_k(X) = 0$ for $k > 3$. In practice, nontrivial $H_3(X)$ rarely arises for small molecules, so we restrict attention to rank-2 complexes, $\mathcal{C}_2 = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, consisting of nodes $\mathcal{V}$, edges $\mathcal{E}$, and faces $\mathcal{F}$.

Concretely, each node $i \in \mathcal{V}$ corresponds to an atom, each edge $(u, v) \in \mathcal{E}$ represents a chemical bond, and each face $r_{\text{set}} \in \mathcal{F}$ represents a chemical ring, with r_{set} denoting the set of bond indices that form the cycle. A chemical ring arises whenever the complex contains an induced cycle; this is captured by a skeleton-preserving lifting transformation that attaches 2-cells to all induced cycles [39]. Because bonds are undirected, but graph and topological neural networks typically treat edges as directed, we represent each bond by a pair of bidirectional edges. Although some works also model functional groups as rank-2 cells [38], we exclude them due to the difficulty of consistently defining chemical features across diverse functional groups and the irregular topologies that complicate graph lifting. The resulting complex is characterised by $N = |\mathcal{V}|$ nodes, $E = |\mathcal{E}|$ edges, and $F = |\mathcal{F}|$ faces.

In implementation, the cellular complex is encoded using two index matrices: the *edge index* and the *face index*. The edge index has shape $[2, |\mathcal{E}|]$, with the first row giving source node indices and the second row giving target node indices. The face index has shape $[2, |\mathcal{F}| \times |\mathcal{E}_F|]$, where $|\mathcal{E}_F|$ is the number of edges per face. Here, the first row stores the face indices, and the second row stores the corresponding edge indices.

2.4 Evaluation Metrics

Attention Entropy

For an attention head with distribution a_{ij} over neighbours $j \in \mathcal{N}(i)$, the entropy is defined as

$$H_i = -\sum_{j \in \mathcal{N}(i)} a_{ij} \log a_{ij}. \tag{2.11}$$

Low entropy indicates sharp focus on a few neighbours, which is desirable when those neighbours correspond to chemically salient interactions. Conversely, high entropy reflects a diffuse, almost uniform distribution, suggesting that the head is not discriminating between relevant and irrelevant neighbours. Entropy has been widely used to probe whether attention modules capture meaningful structure or collapse into noise [51, 52].

Long-Range Attention Ratio (pair-weighted)

Let $a_{ij}^{(b,h)}$ denote the post-softmax attention from node i to j in head h, and let $\mathrm{SPD}(i,j)$ be the shortest-path distance in the (graph). For a cutoff d_c (we report $d_c \in \{8,16\}$), we define the pair-weighted long-range ratio as

$$R_{\text{long}}(d_c) \;=\; \frac{\displaystyle\sum_{h=1}^{H}\sum_{i,j\in\mathcal{V}} \mathbb{1}[\mathrm{SPD}(i,j) > d_c]\; a_{ij}^{(h)}}{\displaystyle\sum_{h=1}^{H}\sum_{i,j\in\mathcal{V}} \mathbb{1}[\mathrm{SPD}(i,j) > 0]\; a_{ij}^{(h)}}. \tag{2.12}$$

Here $\mathcal{V}_b = \{(i,j):\ \mathrm{SPD}_b(i,j) \neq -1,\ i \neq j\}$ are the reachable, non-self pairs (padding/unreachable pairs are excluded). This metric is a literal fraction of total *non-self* attention mass that goes to pairs farther than d_c; hence $R_{\text{long}}(d_c) \in [0,1]$ and it is monotone in d_c (e.g., $R_{\text{long}}(16) \leq R_{\text{long}}(8)$). In molecular graphs, most signal is local, but incorporating distance-aware biases (e.g., via SPD) allows attention to capture global structural context when beneficial (rings, long-range substituent effects, etc.). A balanced profile is therefore essential: excessive long-range focus may drown out local chemistry, while ignoring it risks missing important global cues [52].

Pearson Correlation

The Pearson correlation coefficient is a statistical measure of the linear relationship between two random variables X and Y. It is defined as

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,\,Y)}{\sigma_X\,\sigma_Y}, \tag{2.13}$$

where σ_X and σ_Y are the respective standard deviations. The coefficient $\rho_{X,Y}$ takes values in $[-1,1]$, with positive values indicating a direct linear relationship, negative values an inverse relationship, and values near zero suggesting little to no linear association. This metric is widely used to assess whether variations in one variable correspond systematically to variations in another [53].

Jaccard similarity

The Jaccard index (or Jaccard similarity coefficient) is a set-based metric that quantifies the overlap between two sets. Given two sets A and B, it is defined as $|A\cap B|/|A\cup B|$. In molecular applications, it is often used to measure how much the scaffold set of one collection overlaps with that of another. A high Jaccard value indicates that the two collections share many scaffolds relative to the size of their union, while a low value reflects distinct chemotype coverage [54].

Tanimoto similarity

The Tanimoto coefficient is the standard similarity measure for comparing molecular fingerprints (binary bit vectors). For two bitstrings x and y, it is defined as

$$T(x,y) = \frac{x \cdot y}{\|x\|^2 + \|y\|^2 - x \cdot y}, \tag{2.14}$$

where $x \cdot y$ is the number of shared **on** bits. Intuitively, it is the ratio of common features to total features present across both molecules. Tanimoto similarity ranges from 0 (no shared features) to 1 (identical fingerprints), and is widely used for nearest-neighbour searches, clustering, and library design [55].

Maximum Mean Discrepancy (MMD)

MMD is a statistical distance between probability distributions, defined in terms of expectations under a kernel function k. For datasets X and Y,

$$\mathrm{MMD}^2(X,Y) = \mathbb{E}[k(x,x')] + \mathbb{E}[k(y,y')] - 2\,\mathbb{E}[k(x,y)]. \tag{2.15}$$

When k is the Tanimoto kernel on molecular fingerprints, MMD quantifies how different two molecular collections are in terms of their fingerprint distributions. The Tanimoto similarity has been shown to be a Mercer kernel, and is therefore a positive definite kernel on non-negative feature vectors such as molecular fingerprints [56], making it valid for use within the MMD framework. A lower MMD indicates that the subset and full dataset are closer in their fine-grained chemical makeup [57].

2.5 General Mathematical Notation

In this report, we adopt the following notation: vectors are denoted in bold lowercase (e.g., $\mathbf{v}$), matrices in bold uppercase (e.g., $\mathbf{A}$), vector elements as v_i, and matrix elements as A_{ij}. Unless stated otherwise, vectors are row vectors. Vertical concatenation (stacking of vectors) of a family of vectors $\mathbf{v}_k$ is written as $[\mathbf{v}_k]_{k\in\mathcal{K}}$ with $\mathcal{K} = \{1, 2, \dots, K\}$, or equivalently $[\mathbf{v}_1; \mathbf{v}_2; \dots; \mathbf{v}_K]$. Horizontal concatenation (along the feature dimension) is denoted with a vertical bar, e.g. $[\mathbf{v}_1 \,|\, \mathbf{v}_2 \,|\, \dots \,|\, \mathbf{v}_K]$.

At each layer ℓ, every node i is assigned a feature vector $\mathbf{h}_i^\ell \in \mathbb{R}^{d_\text{node}}$, and stacking these across all N nodes yields the node feature matrix

$$\mathbf{H}^\ell = [\mathbf{h}_1^\ell; \dots ; \mathbf{h}_N^\ell] \in \mathbb{R}^{N \times d_\text{node}}.$$

Similarly, each edge $(u,v) \in \mathcal{E}$ has a feature vector $\mathbf{e}_{uv}^\ell \in \mathbb{R}^{d_\text{edge}}$, and collecting them gives

$$\mathbf{E}^\ell = \left[\mathbf{e}_{uv}^\ell \text{ for } (u,v) \in \mathcal{E}\right] \in \mathbb{R}^{|\mathcal{E}| \times d_\text{edge}}.$$

Each face $r_\text{set} \in \mathcal{F}$ is associated with a feature vector $\mathbf{f}_{r_\text{set}}^\ell \in \mathbb{R}^{d_\text{face}}$, and stacking these yields

$$\mathbf{F}^\ell = \left[\mathbf{f}_{r_\text{set}}^\ell \text{ for } r_\text{set} \in \mathcal{F}\right] \in \mathbb{R}^{|\mathcal{F}| \times d_\text{face}}.$$

In addition, there is a global feature vector $\mathbf{g}^\ell \in \mathbb{R}^{d_\text{global}}$ at layer ℓ. In our implementation, the feature dimensions are fixed as $d_\text{node} = 256$, $d_\text{edge} = 128$, $d_\text{face} = 128$, and $d_\text{global} = 64$.

Chapter 3

Preprocessing and Feature Engineering

3.1 Preprocessing

We treat our task as a multi-class classification. Therefore, for each molecule, we horizontally concatenate the protein's binding label into a row vector in the following order [binds_BRD4 | binds_HSA | binds_sEH]. Therefore, our y label has a size of $\mathbb{R}^{1\times3}$ for each entry. The dataset used [Dy] as a stand-in for the DNA linker in molecule_smiles. Before featurization, we wrote a function to perform a chemically aware removal of the BELKA-specific '[Dy]' (DNA tag placeholder) at the molecular graph level so that downstream feature generation operates on a valid, tag-free ligand. It normalises bare 'Dy' to '[Dy]', parses with RDKit, deletes the '[Dy]' atom and its incident bonds, and sanitises the molecule. This is necessary because leaving '[Dy]' makes RDKit treat it as dysprosium (Z=66), corrupting atomic-number/period/group features and breaking parsing. Graph-level deletion is superior to regex edits (which can produce invalid SMILES/topology), to dummy atom (no period/hybridisation), and to substituting real atoms (invented chemistry). The approach preserves topology (removes a terminal leaf), keeps hydrogen counts consistent, and is robust to multiple/annotated tags.

3.2 Feature Engineering

To transform the $molecule\_smiles$ string into cell features suitable for GNN/TNN models, we adopt feature engineering strategies inspired by the GPS++ framework [58]. In addition to standard chemical descriptors (e.g., atomic degree), we incorporate structural and positional information. The initialisation consists of five components: node states, edge states, face states (excluded in the baseline), global states, and attention biases.

3.2.1 Chemical Features

We first used the RDKit library to generate a mol object, an instance of rdkit.Chem.rdchem.Mol class that encodes atoms, bonds, connectivity and so on. We do not represent hydrogen atoms explicitly as nodes in the graph (hydrogen suppression). There are two reasons for this. First, RDKit stores the implicit hydrogen count of each atom as an internal attribute by default, and the number of hydrogen counts will be one of the encoded features per atom. Second, this approach drastically reduces graph size and speeds up training and evaluation without minimal information loss. Heavy atom connectivities capture most of the chemical reactivity and binding information [59, 60].

We used RDKit to generate chemical features for atoms, bonds, and rings, and encoded them into row vectors $\mathbf{h}^{atom}, \mathbf{e}^{bond}, \mathbf{f}^{ring}$ via one-hot encoding. For atoms, we extracted 12 categorical features: atomic number, group, period, degree, implicit valence electrons, number of hydrogens, number of radical electrons, formal charge, hybridisation type, aromaticity, ring membership, and chirality. This feature set is denoted by $\mathbb{A}$. For bonds, we considered 4 categorical features: bond type, stereochemical configuration, conjugation, and ring membership, denoted by $\mathbb{B}$. For rings, we extracted 5 categorical features: ring size, heteroatom count, saturation, fusion status, and average electronegativity, denoted by $\mathbb{F}$ [36, 38]. The complete list of extracted features and their ranges is shown in Tab. 3.1 (atoms), Tab. 3.2 (bonds), and Tab. 3.3 (rings).

To ensure robustness, we introduced a *misc* category as a default encoding for any unseen values. The encoding ranges were determined by scanning the whole dataset to collect all unique values for each feature type. All features are represented using **one-hot encoding**.

Table 3.1: Atom features

Type	Encoding
Atomic Number	5, 6, 7, 8, 9, 14, 16, 17, 35, 53
Group	13, 14, 15, 16, 17
Period	2, 3, 4, 5
Degree	0, 1, 2, 3, 4
Formal Charge	-1, 0, 1
Implicit valence electrons	0, 1, 2, 3
Number of hydrogens	0, 1, 2, 3
Number of radical electrons	0, 1, 2, 3
Hybridisation	SP, SP2, SP3
Is aromatic	0, 1
Is in ring	0, 1
Is chiral center	0, 1

Table 3.2: Edge features

Type	Encoding
Bond type	Single, Double, Triple, Aromatic
Bond stereo	Stereonone, Stereoe
Is conjugated	0, 1
Is in ring	0, 1

To map the categorical chemical features into a continuous space and reduce dimensionality, we assign a learnable embedding vector to each category, sum the embeddings across all features, and process the result with an MLP. This yields $\mathbf{h}^{atom} \in \mathbb{R}^{d_{\text{node}}}$, $\mathbf{e}^{bond} \in \mathbb{R}^{d_{\text{edge}}}$, and $\mathbf{f}^{ring} \in \mathbb{R}^{d_{\text{face}}}$.

$$\forall i: \quad \mathbf{h}_i^{\text{atom}} = \text{Dropout}_{0.18}\left(\text{MLP}_{\text{node}}\left(\sum_{j \in \mathbb{A}_i} \text{Embed}_{64}(j)\right)\right) \in \mathbb{R}^{d_{\text{node}}} \qquad (3.1)$$

Table 3.3: Ring features. Ring average electronegativity is encoded to the closest value in the encoding list.

Type	Encoding
Ring size	3, 4, 5, 6, 7
Het count	0, 1, 2, 3, 4
Saturated	0, 1
Has fusion	0, 1
Average electronegativity	2.5, 2.55, 2.60, 2.65, 2.70, 2.75, 2.80, 2.85, 2.90, 2.95

$$\forall(u,v): \quad \mathbf{e}_{uv}^{\text{bond}} = \text{Dropout}_{0.18}\left(\text{MLP}_{\text{edge}}\left(\sum_{j\in\mathbb{B}_{uv}} \text{Embed}_{64}(j)\right)\right) \in \mathbb{R}^{d_{\text{edge}}} \tag{3.2}$$

$$\forall(r_{set}): \quad \mathbf{f}_{r_{set}}^{\text{ring}} = \text{Dropout}_{0.18}\left(\text{MLP}_{\text{face}}\left(\sum_{j\in\mathbb{F}_{r_{set}}} \text{Embed}_{64}(j)\right)\right) \in \mathbb{R}^{d_{\text{face}}} \tag{3.3}$$

where $\text{Embed}_{64}(j) \in \mathbb{R}^{64}$ denotes the operation that selects the j^{th} row from a learnable embedding matrix. Some encoding values may be absent from the training partition; in such cases, the corresponding embedding row remains untrained, which does not introduce bias. The embedding matrix has one row for each possible encoding, plus an additional *misc* entry, as defined for the categorical features in Tab. 3.1, Tab. 3.2, and Tab. 3.3. The detailed MLP architecture is described in Sec. 4.1.

3.3 Positional Encodings

Graph neural networks (GNNs) and topological neural networks (TNNs) are inherently permutation invariant, which prevents them from assigning canonical positional information to nodes [61]. As a result, standard MP-GNNs/TNNs often struggle to distinguish between isomorphic graphs (graphs that are structurally identical up to relabelling) [62]. In addition, their reliance on local message passing limits long-range awareness: information must propagate step by step, requiring many layers to reach distant parts of the graph. This process can dilute signals, leading to oversquashing [63].

Positional encodings address these issues by breaking graph symmetries, thereby enhancing the model's discriminative power beyond the 1-Weisfeiler–Lehman (1-WL) test in certain cases [63]. They also enable shallower architectures and faster convergence, since global context can be injected directly rather than accumulated over many propagation steps. This reduces both oversmoothing and training time, while improving performance on structure-sensitive tasks such as molecular property prediction [37, 64].

In our model, we incorporate three forms of positional information: Laplacian eigenvector (and eigenvalues) encodings, random-walk structural encodings, and barycentric subdivision encodings [37].

3.3.1 Graph Laplacian positional encodings

In Transformer architectures, sinusoidal positional encodings (PEs) can be interpreted as eigenfunctions of the Euclidean Laplace operator. Graphs, however, lack a natural linear ordering on which such sinusoids can be defined. Instead, the eigenvectors of the graph Laplacian are employed as positional encodings, as they extend the notion of sine and cosine modes to arbitrary graph structures. These eigenvectors capture frequency-like information across the graph, analogous to how sinusoids represent frequencies along a line. Consequently, graph Laplacian eigenvectors are the natural counterpart to sinusoidal PEs on sequences [64]. By encoding global modes of variation, they provide distance-aware information and enrich the model's expressive capacity [63, 64]. Moreover, pairing eigenvalues with eigenvectors is crucial, since together they characterise important physical properties of molecular graphs and reflect underlying distance metrics [64]. The graph Laplacian is formulated as a global positional encoding as follows:

$$\forall i: \quad \mathbf{h}_i^{\text{LapVec}} = \text{MLP}_{\text{encoder}}\left(\mathbf{U}[i,\, 2\,\ldots\, k^{\text{Lap}}]\right) \in \mathbb{R}^{32}, \quad \text{where } \mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = \mathbf{U}^{\top}\mathbf{\Lambda}\mathbf{U} \tag{3.4}$$

$$\forall i: \quad \mathbf{h}_i^{\text{LapVal}} = \text{MLP}_{\text{encoder}}\left(\frac{\mathbf{\Lambda}'}{||\mathbf{\Lambda}'||}\right) \in \mathbb{R}^{32}, \quad \text{where } \mathbf{\Lambda}' = \text{diag}(\mathbf{\Lambda})[2\,\ldots\, k^{\text{Lap}}] \tag{3.5}$$

where $\mathbf{I}$ denotes the identity matrix, $\mathbf{D}$ the degree matrix, $\mathbf{A}$ the adjacency matrix, and $\mathbf{L}_{sym}$ the symmetrically normalised Laplacian. The matrix $\mathbf{U}$ is orthonormal, with its columns corresponding to the normalised eigenvectors. When constructing $\mathbf{D}$, we treat the graph as undirected: each edge (u, v) contributes once to the degree of both u and v, even if the adjacency representation contains both (u, v) and (v, u). While GPS++ [58] originally employed the unnormalised Laplacian, $\mathbf{L} = \mathbf{D} - \mathbf{A}$, we opted for $\mathbf{L}_{sym}$, as it prevents high-degree (heavy) nodes from dominating the spectral modes. This yields more balanced, degree-invariant positional encodings. In practice, this choice may have little effect, as both variants have been used interchangeably in the literature without strong justification [58, 61], likely because the network can learn the appropriate eigenvector scaling. The exact MLP layer configurations used for encoding are given in Sec. 3.6.

A graph $\mathcal{G}$ with $N = |\mathcal{V}|$ nodes admits at most N linearly independent eigenvectors. Since N varies across graphs, fixing a common number k of eigenvectors requires k to be no larger than the smallest graph size in the dataset. This creates a bottleneck: large graphs are forced to use only a small subset of their available eigenvectors, underutilising their spectral information. To address this and produce fixed shape inputs, we adopt a truncation/padding strategy: eigenvalues and eigenvectors are standardised to a fixed size of $k^{Lap} = 8$, discarding the trivial first eigenvalue $\mathbf{\Lambda}_{11} = 0$ [58, 64]. We focus on the smaller eigenvalues, as they capture more global connectivity structure.

Finally, eigenvectors are subject to a sign ambiguity: each eigenvector can be flipped independently without affecting validity, resulting in 2^k possible sign configurations for k eigenvectors. To account for this, we randomise eigenvector signs at every training epoch, allowing the network to learn invariance to this ambiguity [65].

3.3.2 Random walk positional encodings

Random walk positional encoding (RWPE) leverages the diffusion process of random walks on a graph. Specifically, it encodes the probability that a random walk starting at node i returns to the same node after a given number of steps. This captures distinctive information about the local neighbourhood structure, where the locality is controlled by the step count k [58]. We did not construct the full random walk matrix over all node pairs to reduce computational overhead. Instead, we computed return probabilities for walk lengths ranging from 1 to 16 steps $(k^{RW} = 16)$.

$$\forall i: \quad \mathbf{h}_i^{\text{RW}} = \text{MLP}_{\text{encoder}}\left(\left[(\mathbf{P}^1)_{ii},\ (\mathbf{P}^2)_{ii},\ \cdots,\ (\mathbf{P}^{k^{\text{RW}}})_{ii}\right]\right) \in \mathbb{R}^{32}, \quad \text{where } \mathbf{P} = \mathbf{D}^{-1}\mathbf{A} \quad (3.6)$$

where $\mathbf{P}$ is the transition matrix. The exact MLP layer configurations used for encoding are given in Sec. 3.6.

RWPE enhances discriminative power by distinguishing (i) structurally different nodes and (ii) non-isomorphic graphs that 1-WL and standard MP-GNNs cannot separate. It yields unique node representations provided that each node has a distinct k-hop neighbourhood for sufficiently large k [65]. Although this condition does not hold universally, it is empirically satisfied for most nodes. Even when only approximately met, using larger values of k enables RWPE to capture meaningful higher-order and global positional information across the graph.

3.3.3 Barycentric subdivision Laplacian positional encoding

A straightforward extension of the graph Laplacian to cellular complexes is the Hodge Laplacian. However, there is no established normalisation scheme for the Hodge Laplacian that simultaneously preserves random-walk semantics, orientation invariance, spectral bounds, and comparability across complexes. For $k > 0$, normalisation is not "plug-and-play": one must choose degree weightings for each rank of cells and decide how to balance contributions from lower- and upper-adjacency terms ($B_k^\top B_k$ vs. $B_{k+1}B_{k+1}^\top$). Different choices lead to different operators and spectra [66, 67].

Using the unnormalised operator L_k introduces further issues. Eigenvalues become **scale-dependent** (sensitive to mesh refinement, cell sizes, or counts), hurting cross-complex comparability and stability. For $k > 0$, L_k also entangles cochains and orientations: eigenvectors can flip signs under reorientation, and the harmonic subspace (linked to Betti numbers) may inject ambiguous or task-irrelevant signals [37, 67, 68].

To avoid these complications, we followed [37] and applied the normalised Laplacian operator to the 1-skeleton of the barycentric subdivision of the cellular complex, $\mathcal{X}$. In this construction, the 1-skeleton of $\mathcal{X}$ is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where vertices $\mathcal{V}$ correspond to the cells of $\mathcal{X}$, and an edge $(u, v) \in \mathcal{E}$ exists whenever one cell lies in the closure of the other.

Employing graph Laplacian positional encodings on this 1-skeleton yields a single node-level spectrum that integrates information from all cell ranks. This integrates seamlessly into standard GNNs or Transformers without the need for separate p-form branches. The resulting encoding is global, orientation-free, and computationally simpler than Hodge-based alternatives (no edge/face orientations). Moreover, the additional comparability edges introduced by barycentric subdivision act as shortcuts, mitigating over-squashing effects [69]. Importantly, this approach requires solving only one eigenproblem, rather than separate eigenproblems for each L_p. The barycentric subdivision Laplacian positional encoding is as follows:

$$\begin{aligned}
\forall i: \quad & \mathbf{h}_i^{\text{BSLapVec}} = \text{MLP}_{\text{encoder}}\left(\mathbf{U}_{\mathbf{node}}[i,\, 2\, \ldots\, k^{\text{BSLap}}]\right) \in \mathbb{R}^{32}, \\
& \qquad \text{where } \mathbf{L}_{sym} = \mathbf{I} - \mathbf{D_{BS}}^{-\frac{1}{2}} \mathbf{A_{BS}} \mathbf{D_{BS}}^{-\frac{1}{2}} = \mathbf{U}^{\top} \mathbf{\Lambda} \mathbf{U} && (3.7) \\
\forall i: \quad & \mathbf{e}_i^{\text{BSLapVec}} = \text{MLP}_{\text{encoder}}\left(\mathbf{U}_{\mathbf{edge}}[i,\, 2\, \ldots\, k^{\text{BSLap}}]\right) \in \mathbb{R}^{32}, && (3.8) \\
\forall i: \quad & \mathbf{f}_i^{\text{BSLapVec}} = \text{MLP}_{\text{encoder}}\left(\mathbf{U}_{\mathbf{face}}[i,\, 2\, \ldots\, k^{\text{BSLap}}]\right) \in \mathbb{R}^{32}, && (3.9) \\
\forall i: \quad & \mathbf{g}^{\text{BSLapVal}} = \text{MLP}_{\text{encoder}}\left(\frac{\mathbf{\Lambda}'}{||\mathbf{\Lambda}'||}\right) \in \mathbb{R}^{64}, \quad \text{where } \mathbf{\Lambda}' = \text{diag}(\mathbf{\Lambda})[2\, \ldots\, k^{\text{BSLap}}] && (3.10)
\end{aligned}$$

where $\mathbf{A_{BS}}$ and $\mathbf{D_{BS}}$ denote the adjacency and degree matrices of the 1-skeleton of the barycentric subdivision graph. As before, the graph is treated as undirected when computing degrees. $\mathbf{U}_{node}$ represents the subset of eigenvectors indexed by nodes, with analogous definitions for $\mathbf{U}_{edge}$ and $\mathbf{U}_{face}$. The full matrix $\mathbf{U}$ is orthonormal, and we fix $k^{\text{BSLap}} = 8$. Since the eigenvalues are derived from the 1-skeleton of the barycentric subdivision, which incorporates cells of all ranks, we treat this encoding as global.

3.3.4 Barycentric subdivision random walk positional encoding

We construct similar random walk positional encodings as in Sec. 3.3.2 for all ranks of cells.

$$\begin{aligned}
\forall i: \quad & \mathbf{h}_i^{\text{RW}} = \text{MLP}_{\text{encoder}}\left(\left[(\mathbf{P}^1_{node})_{ii},\, (\mathbf{P}^2_{node})_{ii},\, \cdots,\, (\mathbf{P}^{k^{\text{BSRW}}}_{node})_{ii}\right]\right) \in \mathbb{R}^{32}, \\
& \qquad \text{where } \mathbf{P} = \mathbf{D_{BS}}^{-1} \mathbf{A_{BS}} && (3.11) \\
\forall i: \quad & \mathbf{e}_i^{\text{RW}} = \text{MLP}_{\text{encoder}}\left(\left[(\mathbf{P}^1_{edge})_{ii},\, (\mathbf{P}^2_{edge})_{ii},\, \cdots,\, (\mathbf{P}^{k^{\text{BSRW}}}_{edge})_{ii}\right]\right) \in \mathbb{R}^{32}, && (3.12) \\
\forall i: \quad & \mathbf{f}_i^{\text{RW}} = \text{MLP}_{\text{encoder}}\left(\left[(\mathbf{P}^1_{face})_{ii},\, (\mathbf{P}^2_{face})_{ii},\, \cdots,\, (\mathbf{P}^{k^{\text{BSRW}}}_{face})_{ii}\right]\right) \in \mathbb{R}^{32}, && (3.13)
\end{aligned}$$

where $\mathbf{D_{BS}}$ is the degree matrix of the 1-skeleton barycentric subdivision graph, and $\mathbf{P}_{node}$ denotes the subset of random-walk vectors indexed by nodes, with $\mathbf{P}_{edge}$ and $\mathbf{P}_{face}$ defined analogously. We fix $k^{\text{BSRW}} = 16$.

3.3.5 Local graph centrality encoding (structural encoding)

Graph centrality encodings aim to quantify the importance of each node within the graph. A common choice is degree centrality, where nodes are embedded based on their connectivity. In this approach, each node is assigned two embedding vectors corresponding to its in-degree and out-degree. Since our graphs are undirected, we merge these into a single embedding vector, as defined in Eq. 3.14.

$$\forall i: \quad \mathbf{h}_i^{\text{Cent}} = \text{Embed}_{64}(D^{in}_{ii}) + \text{Embed}_{64}(D^{out}_{ii}) = \text{Embed}_{64}(D_{ii}) \in \mathbb{R}^{64} \qquad (3.14)$$

where D^{in}_{ii} denotes the in-degree and D_{ii} the total degree of node i. Since $D^{in}_{ii} = D^{out}_{ii} = \frac{1}{2} D_{ii}$, we redefine D_{ii} as $\frac{1}{2} D_{ii}$ to avoid redundant use of the embedding space. Centrality encodings provide attention layers with explicit node-importance signals through the queries and keys, enabling the model to better capture semantic correlations and relative importance [70]. We cap the maximum degree at 5, yielding 6 distinct embeddings in total, with higher-degree atoms clipped to 5.

3.3.6 Shortest path attention attention bias (spatial encoding)

To incorporate structural information from a graph or cellular complex, we define a function $\phi(\mathcal{V}_i, \mathcal{V}_j) : N \times N \to \mathbb{R}$ that measures the spatial relation between nodes i and j in graph $\mathcal{G}$. In this work, we set $\phi(\mathcal{V}_i, \mathcal{V}_j)$ to be the shortest-path distance between $\mathcal{V}_i$ and $\mathcal{V}_j$ whenever a path exists [70], and assign a default value of -1 if the nodes are disconnected. Using this definition, we construct the shortest-path distance (SPD) map $\Delta \in \mathbb{N}^{N\times N}$, where Δ_{ij} is the number of edges in the shortest path from node i to node j, or -1 if no such path exists. Each integer distance is then embedded as a scalar attention bias term, yielding the SPD attention bias map $\mathbf{B}^{SPD} \in \mathbb{R}^{N\times N}$, as shown in Eq.3.15 [58].

$$\forall i, j : \quad B_{ij}^{\text{SPD}} = \text{Embed}_1(\Delta_{ij}) \in \mathbb{R} \tag{3.15}$$

For clarity of notation, we assume single-headed attention in this report; in practice, one bias is learned per distance for each head in the multi-headed setting.

We construct two attention bias maps: one on the original graph (without higher-order lifting) and another on the 1-skeleton of the barycentric subdivision. The number of embeddings is set to 35 for the original graph and 60 for the barycentric subdivision. In the SPD map, disconnected node pairs (value -1) are mapped to the final embedding index. For connected nodes whose shortest-path distance exceeds the threshold $(\text{number of embeddings} - 3)$, we assign them to the second-to-last embedding. The subtraction of three accounts for special indices reserved for zero distance, disconnected pairs (-1), and all SPD values exceeding the threshold.

These bias terms allow the Transformer to adaptively modulate attention according to the structural information of the graph or cellular complex. For instance, if the embedding learns a monotonic function that increases with shortest-path distance, the model may attend more strongly to distant nodes than to nearby ones [70].

3.3.7 Learnable Positional and Structural Representations

We initialise nodes with structural (centrality) and positional encodings (LapPE/RWPE), allowing the model to refine these representations throughout all layers rather than appending fixed encodings only before the attention module. The Learnable Structural and Positional Encodings (LSPE) framework demonstrates that injecting learnable PEs early makes message passing and attention position-aware, mitigates overfitting, and achieves higher accuracy than fixed or late-concatenated encodings [65].

3.4 3D coordinates

While incorporating 3D coordinates into the features offers clear advantages and was also explored in GPS++ [58], the original dataset does not provide them, and generating accurate 3D structures from the `molecules_smiles` strings is computationally expensive for a large dataset like BELKA. For this reason, we chose not to include 3D coordinates in our feature set.

3.5 Cell Representation

Graph Transformers are highly expressive but inherently graph-agnostic: without additional signals, attention cannot determine a node's position or structural role. The GPS framework proposes two complementary cues to address this. **Positional encodings (PEs)** inject *where* information such as global coordinates (e.g., Laplacian eigenvectors) or relative distances (e.g., shortest paths), allowing attention to focus on relevant but distant nodes and overcoming the locality limits of message passing. **Structural encodings (SEs)** inject *what/role* information such as degree, centrality, or substructure identifiers, breaking 1-WL symmetries and enabling the model to distinguish nodes that are feature-wise identical. Together, PEs and SEs act as soft inductive biases that enhance expressivity, guide pairwise attention scores, and mitigate over-squashing by enabling multi-hop communication earlier. Importantly, learning these encodings across layers, rather than fixing them, resolves challenges such as Laplacian sign ambiguity and allows positions to adapt to the task. Empirically, combining learned PE/SE with hybrid local–global blocks yields strong, scalable performance across diverse graphs and datasets [65, 71]. Following this recipe, we integrate all the feature sources described above when initialising cell representations in the first layer of our GNN/TNN models.

$$\mathbf{H}^0 = \mathrm{Dense}\left(\left[\mathbf{H}^{\text{atom}} \,\middle|\, \mathbf{H}^{\text{LapVec}} \,\middle|\, \mathbf{H}^{\text{LapVal}} \,\middle|\, \mathbf{H}^{\text{RW}} \,\middle|\, \mathbf{H}^{\text{Cent}}\right]\right) \quad \in \mathbb{R}^{N\times d_{\text{node}}} \tag{3.16}$$

$$\mathbf{E}^0 = \mathrm{Dense}\left(\left[\mathbf{E}^{\text{bond}}\right]\right) \quad \in \mathbb{R}^{E\times d_{\text{edge}}} \tag{3.17}$$

$$\mathbf{F}^0 = \mathrm{Dense}\left(\left[\mathbf{F}^{\text{ring}}\right]\right) \quad \in \mathbb{R}^{F\times d_{\text{face}}} \tag{3.18}$$

$$\mathbf{g}^0 = \mathrm{Embed}_{d_{\text{global}}}(0) \quad \in \mathbb{R}^{d_{\text{global}}} \tag{3.19}$$

$$\mathbf{B} = \mathbf{B}^{\text{SPD}} \quad \in \mathbb{R}^{N\times N} \tag{3.20}$$

where Dense denotes a linear (projection) layer that maps a high-dimensional vector to the desired output dimension, and $\mathbf{X}^{\text{feature}}$ refers to the matrix obtained by vertically stacking the feature vectors $\mathbf{x}_i^{\text{feature}}$, where each $\mathbf{x}_i$ is a feature vector of the same rank. For instance, $\mathbf{H}^{\text{atom}}$ is constructed as $[\mathbf{h}_1^{\text{atom}}; \mathbf{h}_2^{\text{atom}}; \ldots; \mathbf{h}_N^{\text{atom}}]$, with $N = |\mathcal{V}|$. We refer to these as the original cell representations. We also evaluated a variant of the cell representations that incorporates barycentric subdivision positional encodings.

$$\mathbf{H}^0 = \mathrm{Dense}\left(\left[\mathbf{H}^{\text{atom}} \mid \mathbf{H}^{\text{BSLapVec}} \mid \mathbf{H}^{\text{BSRW}} \mid \mathbf{H}^{\text{Cent}}\right]\right) \quad \in \mathbb{R}^{N\times d_{\text{node}}} \tag{3.21}$$

$$\mathbf{E}^0 = \mathrm{Dense}\left(\left[\mathbf{E}^{\text{bond}} \,\middle|\, \mathbf{E}^{\text{BSLapVec}} \,\middle|\, \mathbf{E}^{\text{BSRW}}\right]\right) \quad \in \mathbb{R}^{E\times d_{\text{edge}}} \tag{3.22}$$

$$\mathbf{F}^0 = \mathrm{Dense}\left(\left[\mathbf{F}^{\text{ring}} \,\middle|\, \mathbf{F}^{\text{BSLapVec}} \,\middle|\, \mathbf{F}^{\text{BSRW}}\right]\right) \quad \in \mathbb{R}^{F\times d_{\text{face}}} \tag{3.23}$$

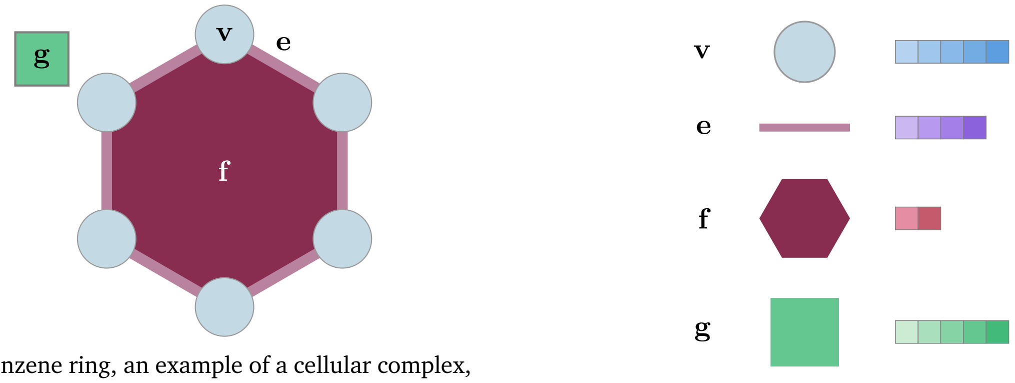
$$\mathbf{g}^0 = \mathbf{g}^{\text{BSLapVal}} \quad \in \mathbb{R}^{d_{\text{global}}} \tag{3.24}$$

$$\mathbf{B} = \mathbf{B}^{\text{SPD}} \quad \in \mathbb{R}^{N\times N} \tag{3.25}$$

Similarly, $\mathbf{H}^{\text{BSLapVec}}$ is defined as $[\mathbf{h}_1^{\text{BSLapVec}}; \mathbf{h}_2^{\text{BSLapVec}}; \ldots; \mathbf{h}_N^{\text{BSLapVec}}]$, where $N = |\mathcal{V}|$. Together, $[\mathbf{H}^{\text{BSLapVec}}; \mathbf{E}^{\text{BSLapVec}}; \mathbf{F}^{\text{BSLapVec}}]$ form $\mathbf{U}$, whose columns are the normalised eigenvectors of the 1-skeleton barycentric subdivision graph. This variant is called the barycentric subdivision (BS) cell representations.

We consider three variants of this setup. The first excludes the face matrix $\mathbf{F}^0$ (denoted *CR-R0*) from the original cell representations. The second is the original cell representations (denoted

CR-R2). The third is the BS cell representations (denoted *CR-BS*). Fig. 3.1 shows an example of the cell representation of a cellular complex.



(a) A benzene ring, an example of a cellular complex, $\mathbf{v}$, $\mathbf{e}$, $\mathbf{f}$ and $\mathbf{g}$ each represent the node, edge, face and global state.

(b) Cells $(\mathbf{v}, \mathbf{e}, \mathbf{f}, \mathbf{g})$ with their correspoinding feature vectors.

Figure 3.1: Example of a cellular complex and its cell representation.

3.6 MLP encoder

The $\text{MLP}_{\text{encoder}}$ used in Laplacian PE, RWPE and their barycentric subdivision variants is a two-layer MLP that projects features to a fixed-dimension latent space. It has the following architecture [58]

$$\begin{aligned} & y = \text{MLP}_{\text{encoder}}(x), && \text{where } x \in \mathbb{R}^h && (3.26) \\ \text{computed as} \quad & \bar{x} = \text{ReLU}(\text{Dense}(\text{LayerNorm}(x))) && \in \mathbb{R}^{2h} && (3.27) \\ & y = \text{Dropout}_{0.18}(\text{Dense}(\text{LayerNorm}(\bar{x}))) && \in \mathbb{R}^{32} && (3.28) \end{aligned}$$

where `LayerNorm` denotes the normalisation operation as defined in [72].

3.7 Code implementation

Owing to the enormous size of the dataset, all cell representations were precomputed on Imperial HPC's multi-core CPUs to ensure scalability and efficiency. The resulting PyTorch tensors were subsequently compressed and stored in cloud storage for streamlined access during training.

Chapter 4

Architecture

Our model, the **General, Powerful, Scalable Graph Transformer on Cellular Complexes (GPS-CC)**, extends the GPS++ architecture [58] with modifications that enable message passing over cellular complexes [71]. Conceptually, GPS-CC is a hybrid between message passing neural networks (MPNNs) and graph Transformers: MPNNs capture local chemical interactions with strong inductive biases, while Transformers model long-range dependencies and global context. This makes GPS++ strike an effective balance: it biases attention with structural and positional information, includes a global state for efficient long-range communication, and scales to millions of graphs without sacrificing training stability [58]. Its partially equivariant design ensures predictions remain robust to geometric variations, a key property when working with conformationally diverse small molecules. To adapt GPS++ to higher-rank structures, we introduced a hierarchical message passing procedure across nodes, edges, and faces in an ordered sequence. Each rank is enriched with chemical features as well as structural and positional encodings, yielding expressive higher-order molecular representations.

The choice of GPS++ as a foundation is motivated by both theoretical and practical considerations. Pure MPNNs excel at encoding local topology but suffer from oversmoothing, oversquashing, and bounded expressivity under the Weisfeiler–Lehman (WL) test; stacking them before a Transformer, as in GraphTrans [73], risks irreversible information loss early in training. Conversely, Transformers are powerful for modelling global interactions but lack efficient inductive bias for local structure. The GPS hybrid architecture overcomes these limitations by running local message passing and global attention **in parallel**. This prevents premature smoothing, grounds attention in structural context, and maintains global flexibility.

From a scalability perspective, GPS++ combines linear-complexity MPNNs ($O(E)$) that operate on dense local aggregation with efficient attention mechanisms restricted to nodes ($O(N)$), making the overall cost effectively linear for sparse graphs like molecules, as the number of edges is proportional to the number of nodes in sparse graphs. This design allows GPS++ to scale to millions of molecular graphs, a critical requirement for the BELKA dataset, which spans $\sim$100M molecules under scaffold shifts. At the same time, GPS++ achieves high expressivity: structural and positional encodings strengthen local reasoning, while attention alleviates oversmoothing and oversquashing by enabling long-range communication. Notably, graph Transformers with full eigenvector encodings are universal function approximators, and GPS++ inherits this universality without discarding edge information [71].

As demonstrated in [71], GPS and GPS++ balance **expressivity, scalability, and generality**, achieving state-of-the-art results across molecular and biological benchmarks. These properties make GPS++ particularly well-suited for BELKA, where scaffold diversity and dataset scale pose unique challenges. Moreover, the modular design of GPS++ makes it naturally compatible with our cellular-complex extension and a promising stepping stone toward the Cellular Transformer paradigm, enabling richer higher-order topological reasoning at scale [37].

4.1 MLP

The architecture for MLP_c where $c \in \{node, edge, face\}$ consists of two layers and is specified as follows:

$$\begin{aligned} \mathbf{y} &= \mathrm{MLP}_c(\mathbf{x}) && (4.1)\\ \text{computed as} \quad \tilde{\mathbf{x}} &= \mathrm{GELU}(\mathrm{Dense}(\mathbf{x})) \in \mathbb{R}^{4d_c} && (4.2)\\ \mathbf{y} &= \mathrm{Dense}(\mathrm{LayerNorm}(\tilde{\mathbf{x}})) \in \mathbb{R}^{d_c} && (4.3) \end{aligned}$$

where `GELU` denotes the activation function introduced in [74].

4.2 GPS-CC Block

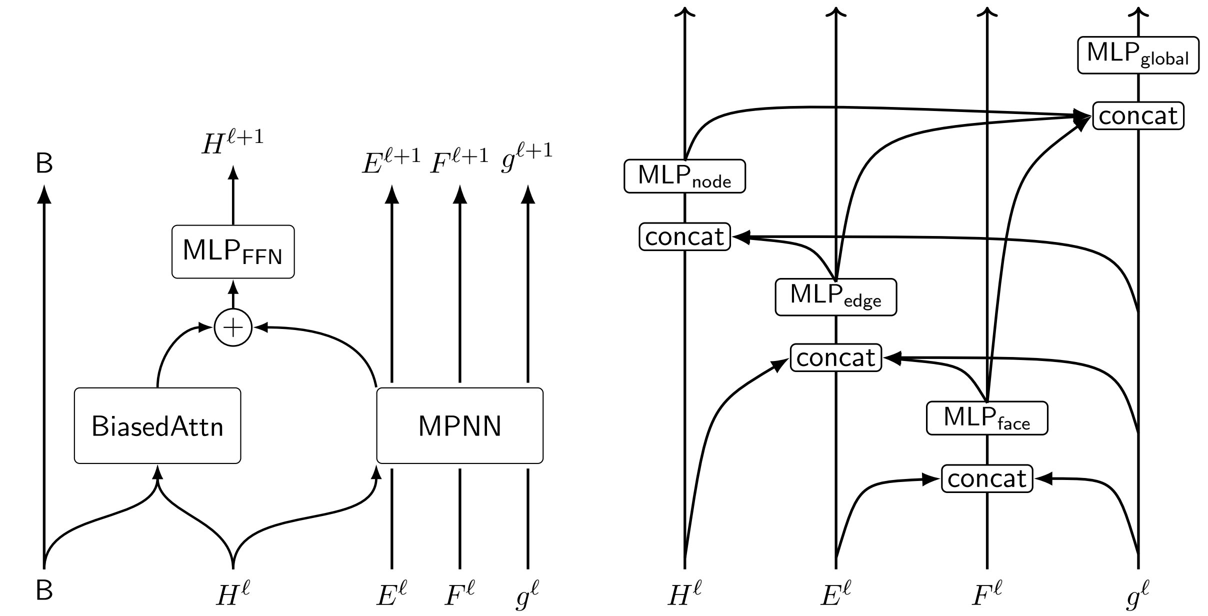
The core GPS-CC block integrates message-passing and attention layers, which operate in parallel and are subsequently summed before being processed by a feed-forward MLP. This composite block is stacked 16 times. The architecture is preceded by an encoder that projects input features into the appropriate latent embedding space and followed by a decoder that produces binding predictions for the three target proteins [58]. The GPS-CC block is defined as follows for layer $\ell > 0$

$$\begin{aligned} \mathbf{H}^{\ell+1}, \mathbf{E}^{\ell+1}, \mathbf{F}^{\ell+1}, \mathbf{g}^{\ell+1} &= \mathrm{GPS-CC}\left(\mathbf{H}^{\ell}, \mathbf{E}^{\ell}, \mathbf{F}^{\ell}, \mathbf{g}^{\ell}, \mathbf{B}\right) && (4.4)\\ \text{computed as} \quad \mathbf{Y}^{\ell}, \mathbf{E}^{\ell+1}, \mathbf{F}^{\ell+1}, \mathbf{g}^{\ell+1} &= \mathrm{MPNN}\left(\mathbf{H}^{\ell}, \mathbf{E}^{\ell}, \mathbf{F}^{\ell}, \mathbf{g}^{\ell}\right), && (4.5)\\ \mathbf{Z}^{\ell} &= \mathrm{BiasedAttn}\left(\mathbf{H}^{\ell}, \mathbf{B}\right), && (4.6)\\ \forall i: \quad \mathbf{h}_i^{\ell+1} &= \mathrm{FFN}\left(\mathbf{y}_i^{\ell} + \mathbf{z}_i^{\ell}\right) && (4.7) \end{aligned}$$

The GPS-CC block shown here contained the face matrix. For the *CR-R0* variant, the $\mathbf{F}^{\ell}$ is removed. Fig. 4.1 visually presents the architecture of our GPS-CC Block, with each component described in detail in the following sections.

4.3 Molecular Message Passing

Our MPNN module used a modified internal block configuration of the message-passing neural network. Specifically, we used a Full GN block architecture, where the global features update the node's, edges', and faces' features and vice versa. This differs from a relatively static update in a message-passing neural network, where the global features are updated by aggregating


B
$H^{\ell+1}$
$E^{\ell+1}$ $F^{\ell+1}$ $g^{\ell+1}$
$\mathrm{MLP}_{\mathrm{FFN}}$
+
BiasedAttn
MPNN
B
H^{ℓ}
E^{ℓ} F^{ℓ} g^{ℓ}
$\mathrm{MLP}_{\mathrm{global}}$
concat
$\mathrm{MLP}_{\mathrm{node}}$
concat
$\mathrm{MLP}_{\mathrm{edge}}$
concat
$\mathrm{MLP}_{\mathrm{face}}$
concat
H^{ℓ}
E^{ℓ}
F^{ℓ}
g^{ℓ}

(a) Architecture of GPS-CC block. It consists of a `BiasedAttn` module and `MPNN` module, a `FFN` then merges them.

(b) `MPNN` module. The arrow direction indicates where the message is passed to.

Figure 4.1: GPS-CC Block architecture.

nodes' features [75]. A **Full Graph Network (Full GN) block** generalises message passing to operate over multiple types of entities with their own attributes, such as nodes, edges, and global features, while supporting flexible aggregation and update functions for each type. This is better for tasks needing a dynamic global context, such as our molecule property prediction task.

When extended to *cellular complexes* with ranks 0, 1, and 2 (nodes, edges, and faces), this framework naturally accommodates higher-order relationships by defining message-passing channels along the *boundary* and *coboundary* operators between adjacent ranks, as well as optional same-rank adjacencies. Each rank maintains its feature vector and update function, and a global block aggregates information across all ranks, enabling both local (rank-adjacent) and global (graph-level) reasoning. This design preserves the structured relational inductive biases emphasised by Battaglia et al. [75], while expanding the representational capacity to higher-dimensional topological domains.

Such an architecture offers several benefits. Working directly with a cell complex's combinatorial structure captures geometric and topological information beyond what ordinary graphs can express. Compared to purely 1-dimensional message passing, incorporating rank two entities enables explicit modelling of surface-like structures and 2-cells, improving expressiveness in higher-order domains. Specifically, our MPNN, a cellular Weisfeiler–Lehman network (CWNs), has the same expressive limit as the Cellular Weisfeiler–Lehman (CWL) test. Like GNNs and WL, CWNs match CWL's power if they are deep enough and use local aggregators that can learn injective mappings [39]. In addition, GNNs need L message-passing steps to capture interactions between nodes L hops apart, which limits efficiency for long-range dependencies. CWNs overcome this by using higher-dimensional cells (e.g., 2-cells) that act as shortcuts, allowing long-range interactions to be captured with a constant, small number of layers (e.g., 3 for ring structures), regardless of graph size [39].

The Full GN formulation ensures that information can propagate between any relevant parts of the complex through boundary chains, coboundary chains, and global aggregation, while keeping the update rules modular and permutation-equivariant. This combination supports richer representations of complex relational systems and can be adapted to various scientific and geometric learning tasks.

4.3.1 Message passing in CW networks

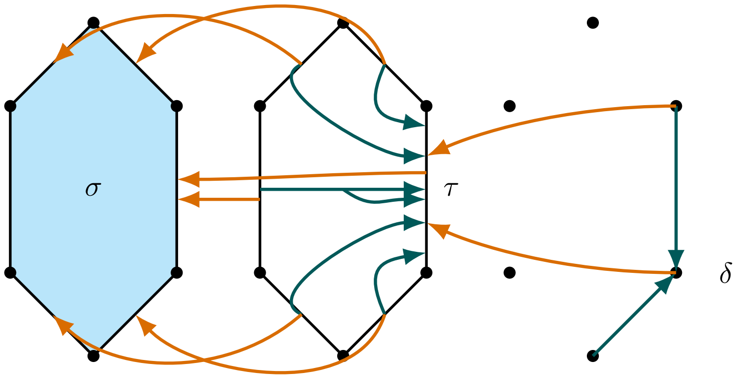


Figure 4.2: Hierarchical illustration of the message passing procedure. Orange arrows denote boundary messages received by cells at levels σ and τ, while teal arrows represent upper-adjacency messages received by cells at levels τ and δ.

Definition 3 *The Cellular Weisfeiler–Lehman, when restricted to exclude coboundary and lower-adjacency relations, retains the same expressive power for distinguishing non-isomorphic cell complexes as the full variant with all adjacency types. [39]*

As established in Def. 3, we can consider only boundary and upper-adjacency relations without reducing the model's expressiveness. Accordingly, cells in our cellular complex receive two types of messages:

$$m_{\mathcal{B}}^{t+1}(\sigma) = \mathrm{AGG}_{\tau\in\mathcal{B}(\sigma)}\left(M_{\mathcal{B}}\left(h_\sigma^t, h_\tau^t\right)\right), \tag{4.8}$$

$$m_{\uparrow}^{t+1}(\sigma) = \mathrm{AGG}_{\tau\in\mathcal{N}_\uparrow(\sigma),\, \delta\in\mathcal{C}(\sigma,\tau)}\left(M_{\uparrow}\left(h_\sigma^t, h_\tau^t, h_\delta^t\right)\right). \tag{4.9}$$

where $\mathcal{B}$ and $\uparrow$ are the boundary and upper adjacencies. σ, τ, δ are cells, C is the coboundary, as stated in Def. 2.

The first type of message passes information from atoms to bonds and from bonds to rings. The second type captures interactions between atoms connected by a bond and between bonds belonging to the same ring. For this second adjacency, atom-to-atom communication incorporates the features of the connecting bond, while bond-to-bond communication incorporates the features of the ring they share. The update step then aggregates these two classes of incoming messages to update the cell features [39]:

$$h_\sigma^{t+1} = U\left(h_\sigma^t, m_{\mathcal{B}}^t(\sigma), m_\uparrow^{t+1}(\sigma)\right). \tag{4.10}$$

A graphical illustration of the hierarchical message passing framework can be seen in Fig. 4.2.

More concretely, the message passing framework for our CW network is as follows:

$$\begin{aligned}
\mathbf{H}^{\ell+1}, \mathbf{E}^{\ell+1}, \mathbf{F}^{\ell+1}, \mathbf{g}^{\ell+1} &= \mathrm{MPNN}\left(\mathbf{H}^{\ell}, \mathbf{E}^{\ell}, \mathbf{F}^{\ell}, \mathbf{g}^{\ell}\right), && (4.11)\\
\forall(r_{set}): \quad \bar{\mathbf{f}}^{\ell}_{r_{set}} &= \mathrm{MLP}_{\text{face}}\left(\left[\mathbf{f}^{\ell}_{r_{set}} \,\middle|\, \sum_{(u,v)\in r_{set}} \mathbf{e}^{\ell}_{uv} \,\middle|\, \mathbf{g}^{\ell}\right]\right), && (4.12)\\
\forall(u,v): \quad m_{\mathcal{B}}^t(\mathbf{e}_{uv}) &= \left[\mathbf{x}^{\ell}_u \,\middle|\, \mathbf{x}^{\ell}_v\right] && (4.13)\\
\forall(u,v): \quad m_\uparrow^{t+1}(\mathbf{e}_{uv}) &= \left[\sum_{(u,v)\in r}\left[\bar{\mathbf{f}}^{\ell}_r \,\middle|\, \frac{1}{\sum_{(m,n)\in r\setminus(u,v)} 1} \sum_{(m,n)\in r\setminus(u,v)} \mathbf{e}^{\ell}_{mn}\right]\right] && (4.14)\\
\forall(u,v): \quad \bar{\mathbf{e}}^{\ell}_{uv} &= \mathrm{Dropout}_{0.0035}\left(\mathrm{MLP}_{\text{edge}}\left(\left[\mathbf{e}^{\ell}_{uv} \,\middle|\, m_{\mathcal{B}}^t(\mathbf{e}_{uv}) \,\middle|\, m_\uparrow^{t+1}(\mathbf{e}_{uv}) \,\middle|\, \mathbf{g}^{\ell}\right]\right)\right), && (4.15)\\
\forall i: \quad \bar{\mathbf{h}}^{\ell}_i &= \mathrm{MLP}_{\text{node}}\left(\left[\mathbf{h}^{\ell}_i \,\middle|\, \sum_{(u,i)\in\mathcal{E}} \left[\bar{\mathbf{e}}^{\ell}_{ui} \,\middle|\, \mathbf{h}^{\ell}_u\right] \,\middle|\, \sum_{(i,v)\in\mathcal{E}} \left[\bar{\mathbf{e}}^{\ell}_{iv} \,\middle|\, \mathbf{h}^{\ell}_v\right] \,\middle|\, \mathbf{g}^{\ell}\right]\right), && (4.16)\\
\bar{\mathbf{g}}^{\ell} &= \mathrm{MLP}_{\text{global}}\left(\left[\mathbf{g}^{\ell} \,\middle|\, \sum_{j\in\mathcal{V}} \bar{\mathbf{h}}^{\ell}_j \,\middle|\, \sum_{(u,v)\in\mathcal{E}} \bar{\mathbf{e}}^{\ell}_{uv} \,\middle|\, \sum_{(r_{set})\in\mathcal{F}} \bar{\mathbf{f}}^{\ell}_{r_{set}}\right]\right), && (4.17)\\
\forall i: \quad \mathbf{h}^{\ell+1}_i &= \mathrm{LayerNorm}\left(\mathrm{Dropout}_{0.3}\left(\bar{\mathbf{h}}^{\ell}_i\right)\right) + \mathbf{h}^{\ell}_i, && (4.18)\\
\forall(u,v): \quad \mathbf{e}^{\ell+1}_{uv} &= \bar{\mathbf{e}}^{\ell}_{uv} + \mathbf{e}^{\ell}_{uv}, && (4.19)\\
\forall(r_{set}): \quad \mathbf{f}^{\ell+1}_{r_{set}} &= \bar{\mathbf{f}}^{\ell}_{r_{set}} + \mathbf{f}^{\ell}_{r_{set}}, && (4.20)\\
\mathbf{g}^{\ell+1} &= \mathrm{Dropout}_{0.35}\left(\bar{\mathbf{g}}^{\ell}\right) + \mathbf{g}^{\ell}. && (4.21)
\end{aligned}$$

where $\mathrm{Dropout}_p$ denotes elementwise masking with probability p [76, 77]. Equation 4.14 defines the upper-adjacency message received by an edge. We adopt mean aggregation: for each edge e_{uv}, all incident faces are considered. For every such face, we concatenate its attribute with the mean of the attributes of all edges in that face, excluding e_{uv} itself. The final edge update is then obtained by summing over all faces containing $\mathbf{e}_{uv}$ [39].

To clarify the notation in Eq. 4.12 and Eq. 4.14:

- $\sum_{(u,v)\in r_{set}}$ refers to the sum of all edge attributes in face $\mathbf{f}_{r_{set}}$,
- $\sum_{(u,v)\in r}$ denotes the sum of all face attributes associated with edge $\mathbf{e}_{uv}$, and
- $\sum_{(m,n)\in r\setminus(u,v)}$ represents the sum of all edge attributes in face $\mathbf{f}_r$ excluding $\mathbf{e}_{uv}$.

We use mean aggregation rather than summation to ensure comparability of aggregated edge attributes across rings of different sizes in the upper adjacency message. Ring size information is already encoded in the chemical features of each ring. Dropout is not applied in Eq. 4.12, since ligands typically contain only a small number of rings.

We did not add explicit cyclic or orientation encodings for molecular rings because the combination of random walk PE, Laplacian PE, shortest path distances, and centrality is sufficient for capturing long-range and cyclic dependencies without enumerating ring structures. Furthermore, orientation information in molecular rings is not chemically meaningful: molecules are modelled as undirected graphs, and cyclic orientation (clockwise vs. counterclockwise) is an artefact of representation rather than a physical property. Aromaticity and conjugation effects, which are the chemically relevant aspects of rings, are already embedded in atom/bond features and can be propagated effectively through message passing and global attention layers.

Finally, explicit cycle encodings increase computational overhead (requiring cycle basis enumeration or face index construction) and can be redundant with existing shortest-path distance, centrality, and spectral encodings. We believe the hybrid MPNN–Transformer design of GPS-CC is expressive enough to model ring-dependent interactions from these features directly.

For variant *CR-R0*, Eq.4.12, 4.14, 4.20 is removed and any references to them (in Eq. 4.15, Eq. 4.17 are also removed. The message passing framework contains a hierarchical structure where the aggregation and update start from the highest cell rank, flow to the next highest rank, and end with the global state update [39].

The message passing framework is similar to the GPS++ paper [58] in which we used horizontal concatenation as the `AGGREGATION` and MLP as the `UPDATE`.

4.4 Biased Attention Transformer

The biased attention transformer block follows the standard attention mechanism [78], but introduces an additional bias term to the attention scores before the softmax operation [70]. This bias encodes structural priors from the input graph. In our case, the bias is derived from the shortest-path distance map, as defined in Sec. 3.3.6. The equation is as follows:

$$\begin{aligned}\mathbf{Z}^{\ell} &= \mathrm{MultiHeadBiasedAttn}(\mathbf{H}, \mathbf{B}) \\ &= \mathrm{Concat}_{i=1}^{h}\left(\mathrm{Dropout}_{0.3}\left(\mathrm{Softmax}\left(\frac{(\mathbf{H}\mathbf{W}_Q^{(i)})(\mathbf{H}\mathbf{W}_K^{(i)})^{\top}}{\sqrt{d_{\text{head}}}} + \mathbf{B}\right)\right)(\mathbf{H}\mathbf{W}_V^{(i)})\right)\mathbf{W}_O, \\ d_{\text{head}} &= \frac{d_{\text{node}}}{h}, \quad \mathbf{H}\mathbf{W}_Q^{(i)}, \mathbf{H}\mathbf{W}_K^{(i)}, \mathbf{H}\mathbf{W}_V^{(i)} \in \mathbb{R}^{N \times d_{\text{head}}}, \quad \mathbf{B} \in \mathbb{R}^{N \times N}, \quad \mathbf{W}_O \in \mathbb{R}^{h d_{\text{head}} \times d_{\text{node}}}.\end{aligned} \tag{4.22}$$

where $h = 32$ and $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ and $\mathbf{W}_O$ are learnable weights. We then performed GraphDropout, residual update and LayerNorm on $\mathbf{Z}^{\ell}$.

$$\mathbf{Z}^{\ell} = \text{LayerNorm}(\text{GraphDropout}_{\frac{t}{L}0.3}(\mathbf{Z}^{\ell}) + \mathbf{H}^{\ell} \quad \in \mathbb{R}^{N \times d_{\text{node}}} \tag{4.23}$$

GraphDropout means some graphs are masked, and the remaining graphs are upscale by $1 - p$, where p is the dropout rate, similar to regular Dropout.

Transformers are introduced into graph neural networks to overcome key limitations of standard message passing neural networks (MPNNs). While MPNNs propagate information only through local neighbourhoods, requiring many layers to capture long-range dependencies (often leading to oversmoothing and vanishing gradients), they also remain limited in expressiveness, being bounded by the Weisfeiler–Lehman test and overly reliant on adjacency structure. In contrast, Transformers provide a global receptive field from the outset, allowing any node to attend directly to all others in a single step. Their attention mechanism adaptively learns which nodes or edges are most relevant for the task, making them more expressive and scalable for complex graph learning problems. However, a vanilla transformer lacks inherent structural bias, treating all nodes uniformly and thus performing poorly on graph-structured data. To address this limitation, positional encodings, structural encodings, and attention biases are incorporated into the feature representations or attention layers to inject graph structural awareness [58, 70].

4.5 Feed Forward Network

The feed-forward network module has the following architecture:

$$
\begin{aligned}
&& \mathbf{y} &= \mathrm{FFN}(\mathbf{x}) && && (4.24)\\
&\text{computed as} & \bar{\mathbf{x}} &= \mathrm{Dropout}_p(\mathrm{GELU}(\mathrm{Dense}(\mathbf{x}))) && \in \mathbb{R}^{4d_{\text{node}}}, && (4.25)\\
&& \mathbf{y} &= \mathrm{GraphDropout}_{\frac{t}{L}0.3}(\mathrm{Dense}(\bar{\mathbf{x}})) + \mathbf{x} && \in \mathbb{R}^{d_{\text{node}}}, && (4.26)\\
&& \mathbf{y} &= \mathrm{LayerNorm}\left(\mathbf{y}\right) && \in \mathbb{R}^{d_{\text{node}}}. && (4.27)
\end{aligned}
$$

where we set $p = 0$ for our model. The sum of node features from MPNN and BiasedAttention was taken before passing into the feed-forward network.

4.6 Output Decoder

We used the same MLP architecture as in Sec. 4.1 for our output decoder. We apply a **global mean pooling** operation over the node features output by the GPS-CC block to obtain the graph-level representation used as input to the decoder. Compared to global sum pooling, mean pooling normalises for graph size, preventing larger molecules from dominating the representation simply due to having more atoms. This choice is particularly important in the BELKA binding affinity prediction task, where ligands vary considerably in size: the model should capture the distributional patterns of atom-level features rather than their absolute counts. Prior work has also noted that sum pooling may bias models towards graph size, while mean pooling offers more stable representations across molecules of different scales [79]. Alternative pooling strategies (e.g., attention-based pooling) may introduce additional parameters and potential overfitting. In contrast, mean pooling provides a robust and size-invariant aggregation that aligns with the goal of learning comparable affinity predictions across diverse ligands.

4.7 Encode Process Decode

In molecular property prediction, the encode–process–decode architecture is well-suited because it mirrors the hierarchical reasoning needed to extract meaningful information from molecular graphs. The **encoder** maps raw molecular inputs, G_{inp}: atoms, bonds, and possibly higher-order structures into a latent graph representation that captures local chemical features (e.g., atom types, bond orders) and global molecular context (e.g., size, charge). This ensures the model starts from a chemically informed embedding space rather than sparse raw descriptors. The **process** stage, implemented as repeated GN blocks, then performs iterative message passing, allowing information about local electronic environments, substructures like rings, and long-range interactions to propagate across the molecular graph. This iterative refinement is crucial in chemistry, where properties such as polarity, aromaticity, or reactivity often depend on relationships between distant parts of the molecule (Gilmer et al., 2017; Battaglia et al., 2018). The transformer attention mechanism here allows the model to refine its learned structural information. Finally, the **decoder** projects the processed graph into the desired output space, G_{out}, binding affinity in our case, effectively translating the learned representation into task-specific predictions. Fig. 4.3 shows our complete GPS-CC architecture using the various GN blocks.

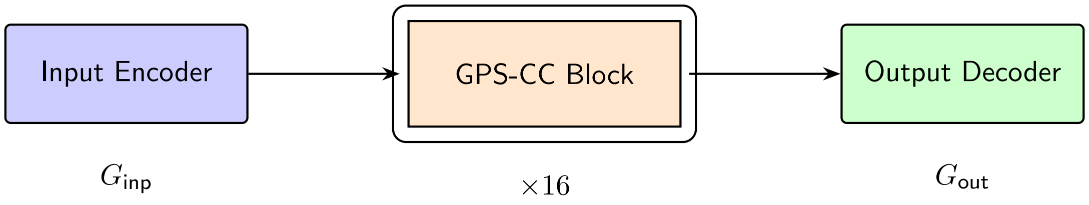


Figure 4.3: Encode-process-decode architecture, G_{inp} represents the raw input (SMILES string) and G_{out} represents the binding predictions of the three target proteins.

The separation of roles in encode–process–decode offers both **modularity** and **inductive bias**: the encoder enforces a chemically meaningful embedding, the process stage ensures expressive and scalable propagation of structural information, and the decoder tailors the representation to the prediction task. This design also naturally supports deeper reasoning than a single GN block, as the core's recurrent application enables the model to capture interactions at multiple scales from local atom-bond relations to global molecular properties while maintaining computational efficiency.

Tab. 4.1 reports the distribution of parameters across submodules. The counts correspond to the configuration with attention, face features, and barycentric subdivision encoding enabled. Variants that omit attention, face features, or barycentric subdivision encoding are expected to require fewer parameters overall. We could see that the GPS-CC blocks contained the most parameters.

Table 4.1: Breakdown of the number of trainable parameters in the GPS-CC model. The majority of parameters are concentrated in the GPS-CC blocks, with MPNN layers dominating the count. Percentages are given relative to the total of $\sim$51M parameters.

Name	**Parameters (M)**	**Percentage**
Encoders	0.70	1.37%
GPS-CC (16 Blocks)	49.99	98.11%
MPNN	37.35	73.31%
BiasedAttn	4.22	8.28%
FFN	8.42	16.52%
Decoder	0.27	0.53%
Total	50.95	100%

Chapter 5

Loss Function

The BELKA dataset exhibits a strong imbalance between positive and negative labels, with negative samples outnumbering positives by roughly 200:1. This imbalance introduces numerical instability during training, as the rare positive labels produce disproportionately large gradient updates under standard loss functions. To mitigate the model's sensitivity to this effect, we evaluate several alternative loss formulations and compare their performance. Here, the y labels are binary indicators specifying whether a small molecule binds to each of the target proteins, framed as a multi-label classification problem.

5.1 Balanced Binary Cross Entropy

The first loss function we implemented is the weighted binary cross-entropy.

$$\mathcal{L}_{\text{ML-WBCE}} = -\frac{1}{NC} \sum_{i=1}^{N} \sum_{c=1}^{C} \Big[w_c \, y_{ic} \, \log(\hat{p}_{ic}) \; + \; (1 - y_{ic}) \, \log\big(1 - \hat{p}_{ic}\big) \Big]. \tag{5.1}$$

where $y_{ic} \in \{0, 1\}$ is the true label of i sample for class c, $\hat{p}_{ic}$ is the predicted label of i sample for class c, w_c is the weighting for the positive class c, assuming the negative labels have a weighting of 1, N is the number of samples and C is the number of class. The weighting is calculated as $w_c = \frac{n_c}{p_c}$, where n_c is the number of negative labels for class c and p_c is the number of positive labels for class c.

Weighted binary cross-entropy helps handle class imbalance by increasing the loss contribution of underrepresented classes, usually the positives. It encourages the model to pay attention to rare events, improving recall without major changes to training. In multi-label problems, different weights can be applied per class for more control.

5.2 Focal Loss

A limitation of balanced cross-entropy loss is that easily classified (negative) samples still contribute non-negligible loss values and thus dominate the gradient. Although it balances the relative weight of positive and negative samples, it does not distinguish between easy and hard examples. Focal loss addresses this issue by reshaping the objective to down-weight easy cases and concentrate training on hard, misclassified samples [80]. This is achieved by introducing a modulating factor $(1-p)^{\gamma}$, where γ is a tunable hyperparameter. The equation of focal loss is as follows:

$$\mathcal{L}_{\text{FL}} = -\frac{1}{NC}\sum_{i=1}^{N}\sum_{c=1}^{C}\Big[\alpha_c\, y_{ic}\,(1-\hat{p}_{ic})^{\gamma}\,\log(\hat{p}_{ic}) + (1-\alpha_c)\,(1-y_{ic})\,\hat{p}_{ic}^{\gamma}\,\log(1-\hat{p}_{ic})\Big] \tag{5.2}$$

where α_c is the balancing factor for class c, similar to the w_c in Sec. 5.1, γ is the focusing factor. We set $\alpha_c = 0.80$ for all c and $\gamma = 2.0$ in our code. When an example is misclassified (p_{ic} small), the loss is nearly unchanged, but as $p_{ic} \to 1$, the loss contribution approaches zero. The parameter γ controls how strongly easy examples are down-weighted: $\gamma = 0$ recovers standard cross-entropy, while larger γ increases the focus on hard examples. For example, with $\gamma = 2$, predictions with $p_{ic} = 0.9$ or 0.968 have 100x and 1000x lower loss than cross-entropy. This makes focal loss particularly effective for correcting misclassified or minority-class examples [80].

5.3 Asymmetric Loss

Asymmetric Loss (ASL) is a variant of focal loss designed for extreme class imbalance, as often occurs in multi-label problems. It introduces two key mechanisms: (i) **asymmetric focusing**, where separate focusing parameters are used for positives and negatives, γ_+ for positives and γ_- for negatives, with $\gamma_- > \gamma_+$ to aggressively suppress easy negatives while keeping gradients for rare positives; and (ii) **asymmetric probability shifting**, where a margin m discards very easy negatives by setting their contribution to zero if $p < m$. Together, these modifications prevent easy negatives from dominating the gradient while ensuring positives retain meaningful learning signals [81].

In molecular binding prediction, where true binders are rare and non-binders are abundant, standard cross-entropy or focal loss often fail: a high γ suppresses positive gradients, while a low γ lets negatives overwhelm the model. ASL addresses this by using a small γ_+ (often zero) so positives train under cross-entropy, a larger γ_- to down-weight trivial non-binders, and a margin m to discard easy negatives entirely. This makes ASL especially suited to binding datasets: it highlights scarce positive binding signals, ignores floods of trivial negatives, and reduces the effect of mislabeled non-binders, ultimately enabling the model to learn more discriminative molecular features.

$$\mathcal{L}_{ASL} = -\frac{1}{NC}\sum_{i=1}^{N}\sum_{c=1}^{C}\Big[y_{i,c}\,(1-\hat{p}_{i,c})^{\gamma_+}\,(\log(\hat{p}_{i,c})) + (1-y_{i,c})\,(\hat{p}_{m,i,c})^{\gamma_-}\,(\log(1-\hat{p}_{m,i,c}))\Big] \tag{5.3}$$

$$\text{with} \quad \hat{p}_{i,c} = \sigma(z_{i,c}), \quad \hat{p}_{m,i,c} = \max(\hat{p}_{i,c} - m, 0), \quad y_{i,c} \in \{0,1\} \tag{5.4}$$

where $z_{i,c}$ is the logit of sample i label c. σ is the sigmoid function. For our implemntation, we used $\gamma_+ = 0, \gamma_- = 2, m = 0.03$.

5.4 Center Loss

Center loss introduces class-specific prototypes (centres) in the embedding space and penalises the distance between each ligand's embedding and its corresponding binding class centre. In a multi-label setting, a ligand that binds multiple proteins will be pulled towards each of their centres. This enforces **intra-class compactness** while preserving **inter-class separability**, so that ligands bind to the same protein cluster more tightly, even when their raw molecular structures are quite different [82]. For the multi-label case, each sample i can belong to a set of classes $T_i \subseteq \{1, \ldots, C\}$, represented by a binary target matrix $t_{ij} \in \{0, 1\}$. The loss is then defined as

$$\mathcal{L}_{CL}^{\text{multi}} = \frac{1}{2\,N_+} \sum_{i=1}^{m} \sum_{j=1}^{C} t_{ij}\, \|x_i - c_j\|_2^2, \tag{5.5}$$

where $N_+ = \sum_{i=1}^{m} \sum_{j=1}^{C} t_{ij}$ is the total number of positive (sample, class) pairs in the batch.

We implemented Center Loss by maintaining a set of class centres in the embedding space and penalising the squared distance between sample embeddings and their corresponding centres. The samples' embeddings and centres have dimension $d \in \mathbb{R}^{64}$. The centres are updated using an exponential moving-average rule as shown in Eq. 5.6, stabilising training and reducing the impact of noisy or rare samples. The loss supports both single-label and multi-label settings: in the single-label case, each sample is penalised based on the squared Euclidean distance to its class centre; in the multi-label case, distances are computed against all positive class centres for a given sample, and averaged across positives. During training, centres are updated efficiently by aggregating batch-wise statistics. This design ensures stable centre updates, handles imbalanced batches gracefully through the EMA weighting, and can be seamlessly integrated with other classification losses such as BCE, focal loss or asymmetric loss.

$$\Delta c_j = \frac{\sum_{i=1}^{m} t_{ij}\,(x_i - c_j)}{\epsilon + \sum_{i=1}^{m} t_{ij}}, \qquad c_j \leftarrow c_j + \alpha\, \Delta c_j, \tag{5.6}$$

where only centres with at least one positive assignment ($\sum_i t_{ij} > 0$) are updated, we used $\alpha = 0.2$.

5.5 Combining Loss Function

In protein–ligand binding prediction, we often want the model to assign correct binary labels (binds vs. not-binds for each protein target) and learn **structured embeddings** of ligands that reflect biological activity. Standard binary cross-entropy (BCE) loss or its variants, such as focal loss and asymmetric loss, can achieve classification accuracy, but they do not constrain how embeddings are arranged in the latent space. This can lead to embeddings that are scattered within the same binding class, especially when different ligands exhibit diverse chemotypes yet share the same binding outcome.

Balanced BCE, focal loss, or asymmetric loss provides the **discriminative decision signal**, handling class imbalance and separating positives from negatives. In contrast, center loss provides a complementary **metric regularizer** that tightens positive clusters (and, if desired, each label's cluster in multi-label tasks). This combo typically improves **recall at a fixed precision**, yields **more stable thresholds** (better calibration) when positives are scarce, and avoids the mining complexity of triplet/contrastive losses [82]. In practice, BCE/focal/asymmetric handle inter-class separation while center loss sharpens intra-class compactness.

Together, they encourage the network to produce **robust, discriminative embeddings** that capture binding specificity, handle extreme imbalance across targets, and improve generalisation to new protein–ligand pairs. In our training pipeline, we added center loss to our classification loss with a hyperparameter λ as the weighting.

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{BCE/Focal/ASL}} + \lambda \mathcal{L}_{\text{CL}} \tag{5.7}$$

where $\lambda = 0.01$ for our implementation.

5.6 Other Loss Function

Support Vector Data Description (SVDD) is a classical one-class classification method that learns a minimum-volume hypersphere in feature space enclosing the training data. Given embeddings $\phi(x)$, SVDD learns a center c and radius R by solving:

$$\begin{aligned} \min_{R,c,\{\xi_i\}} \quad & R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \|\phi(x_i) - c\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \end{aligned} \tag{5.8}$$

where $\nu \in (0, 1]$ is a regularization parameter. In deep one-class classification [83], the kernel embedding is replaced by a neural network $f_\theta(x)$, leading to the simplified objective (assuming the data is normally distributed)

$$\mathcal{L}_{\text{SVDD}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \|f_\theta(x_i) - c\|^2. \tag{5.9}$$

Why We Considered SVDD. Given the extreme class imbalance in BELKA binding prediction (few positive binders versus many negatives), SVDD appeared at first as a promising candidate. Its principle of compactly enclosing "normal" data within a hypersphere suggested a way to separate rare positives from abundant negatives, by treating positives as potential anomalies in latent space. This perspective aligns with the anomaly-detection roots of SVDD and motivated our initial consideration of this loss.

Why We Do Not Use SVDD. However, the Deep SVDD framework [83] demonstrates that the method is only valid under strict conditions: the hypersphere center c must be initialised away from the trivial all-zero solution, neural networks must not contain bias terms, and activation functions must not be bounded (e.g. sigmoid, $\tanh$), in order to avoid collapse. Our architecture for BELKA prediction violates some of these assumptions, since it includes bias terms, employs residual and attention layers, and does not enforce the specific initialisation of c. Consequently, applying SVDD loss would likely produce degenerate embeddings or a trivial collapse. We therefore adopt classification-oriented objectives (BCE, focal loss, asymmetric loss), optionally regularised with metric learning approaches such as center loss, which are better aligned with multi-label, imbalanced binding prediction.

Chapter 6

Methods

Due to computational constraints, we trained our model on a subset of the BELKA dataset rather than the whole dataset, which was infeasible to process within a reasonable timeframe. To preserve the inherent label imbalance, the downsampling procedure was designed to maintain approximately the original positive-to-negative ratio. This ensured that the training subset remained representative of the full dataset.

6.1 Features

All chemical features of atoms, bonds, and rings, along with positional, structural, and centrality encodings, were precomputed on the CPUs. When a feature required an MLP or an embedding map, that component was deferred to training, while feature concatenation and projection were also performed on-the-fly. The precomputed features were stored in PyTorch tensor format and compressed into zip archives. Multiprocessing was employed to accelerate preprocessing; computing features for 10 million molecules required approximately 30 minutes on a 64-core processor. To reduce RAM usage during training, tensors were stored in the smallest feasible data type (typically 'int8' for categorical features and 'float32' for continuous features) and upcast to higher precision only during training. We also stored the original cell and barycentric subdivision (BS) cell representations separately. For 10M molecules, the original cell representation required $\sim$370 GB before compression and $\sim$32 GB after compression, while the BS cell representation required $\sim$550 GB before compression and $\sim$62 GB after compression.

6.2 Model training

6.2.1 Data Loader

To efficiently handle large and variable-sized molecular graphs during training, we developed a custom dataloader that extends the standard PyTorch Geometric `Collater` and integrates a token-bucket batch sampler. Unlike the default collater, our implementation can correctly batch higher-order structures such as face indices (rank-2 cells), which are not natively supported. We achieve this by padding shortest-path distance (SPD) matrices with -1 to the maximum graph size in each batch and by offsetting face indices and edge indices when combining multiple graphs, ensuring that structural consistency is preserved across the batch. In addition, the token-bucket sampler groups graphs by node and edge counts to balance GPU memory usage, though in practice it yields only minor gains, as most batches already contain similar node counts. The gain may be more apparent if the batch size is smaller. For simplicity, we defaulted to fixed-graph batch sizes during training. These extensions enable scalable training of GNNs on protein–ligand datasets with both graph-level and higher-order topological features.

6.2.2 Training Configuration

For model optimisation, we employed the AdamW optimiser with fused CUDA kernels where available, in preference to standard Adam. AdamW decouples weight decay from the gradient update, which improves generalisation and avoids Adam's tendency to over-regularise when weight decay is used. Training was carried out with a warmup–decay learning rate schedule, where the learning rate is linearly increased from zero to **0.0003** during the first 10 warmup epochs, and then decayed linearly towards zero over 450 total epochs. To further enhance efficiency, training was performed under mixed precision (AMP) using PyTorch's autocast and GradScaler. The GradScaler dynamically scales losses to avoid underflow when using half precision, ensuring stable gradients while benefiting from reduced memory consumption and faster tensor operations. Gradients were clipped to a maximum norm of five, preventing instability from exploding gradients, and centres (when present) were renormalised after each update to maintain numerical stability. To avoid overfitting and reduce unnecessary computation, we also employed early stopping with a patience of 15 epochs, halting training when validation performance failed to improve within that window. Logging was integrated via Weights and Biases (WandB) to monitor training loss, learning rate, and additional metrics across iterations and epochs. This setup balances training stability, efficiency, and scalability, making it well-suited for large graph neural network models.

6.2.3 Numerical Precision

To accelerate training while reducing memory consumption, we adopted mixed precision training [84]. This technique combines single-precision (FP32) and half-precision (FP16) representations during computation, enabling faster matrix multiplications and reduced memory bandwidth usage without compromising model accuracy. Critical operations such as gradient accumulation and loss scaling are maintained in FP32 to preserve numerical stability, while less sensitive operations are executed in FP16 for efficiency. In addition, we configured PyTorch to use high-precision matrix multiplications for FP32 operations, which further optimised throughput on modern GPU hardware. Together, these adjustments improved training scalability and efficiency.

6.2.4 Hardware and Dataset

We trained our models on two different sizes of the dataset. The larger subset has 10 million data points (SMILES string), and the smaller subset has 1 million data points. The larger subset was trained on the Nvidia L40 48GB GDDR6. It has a memory bandwidth of 864GB/s and delivers a total of 91.6 teraFLOPS of FP32 compute and 183 teraFLOPS of TF32 compute [85]. The smaller subset was trained on the Nvidia A30 24GB HBM2. It has a memory bandwidth of 933GB/s and delivers a total of 10.3 teraFLOPS of FP32 compute and 82 teraFLOPS of TF32 compute [86]. We used the full validation and test dataset for both subsets.

6.2.5 Training speed and memory

Apart from the GPU computer capability, the batch size and GPU memory can also affect the model's total training time. A large batch size can reduce the number of steps, which affects total training time, but at a higher GPU memory utilisation. In practice, we observe that having a large batch size (and hence fewer steps) results in shorter total training time than vice versa. All the model variants have similar training time, except for the variant with and without attention. For the smaller subset and a batch size of 512 (number of graphs), it took 12 minutes to train one epoch with 42% memory utilisation without attention and 25 minutes with 92% memory utilisation with attention on the A30. For the larger subset and a batch size of 1024 (number of graphs), it took 83 minutes to train one epoch with 42% memory utilisation without attention and 175 minutes with 95% memory utilisation with attention on the L40. The total number of epochs (before early stopping is triggered) is around 30 without attention and 35 with attention.

6.2.6 Scoring metric

The **Average Precision (AP)** score is a ranking-based evaluation metric summarising the trade-off between precision and recall. It is defined as the weighted mean of the precision values at each point where recall increases, with the weight being the change in recall. Formally, for a ranked list of predictions,

$$\mathrm{AP} = \sum_{k=1}^{N} P(k) \cdot \Delta R(k), \tag{6.1}$$

where $P(k)$ is the precision when considering the top k samples and $\Delta R(k)$ is the increase in recall from adding the k-th sample [87]. This formulation is mathematically equivalent to integrating the area under the precision–recall curve. However, our implementation of AP from `scikit-learn` does not use interpolation between points, as a linear interpolation of points on the precision-recall curve provides an overly optimistic measure of classifier performance [88, 89]. Instead, it is calculated using the stepwise area between each sample step. Unlike simple accuracy, AP is insensitive to the large number of negatives in binding datasets and directly rewards models for ranking true binders higher than non-binders. This makes AP robust to class imbalance and particularly well suited to molecular binding prediction tasks, where the quality of the top-ranked predictions is of primary importance.

We use the same scoring metric as in the original BELKA Kaggle competition for a fair comparison. Model performance was evaluated using Mean Average Precision (MAP), which was elected as a more stable alternative to top-k precision while still reflecting the accuracy of the highest-ranked predictions, where experimental validation is most costly. To mitigate biases introduced by varying hit rates across different dataset splits, AP was computed separately for each protein target (sEH, HSA, BRD4) and each split group (shared building blocks, non-shared building blocks, and proprietary library (kin0)). This yields nine AP values, which are averaged with equal weight to form the final score. Notably, two-thirds of the score derives from molecules outside the training distribution, which either incorporate novel building blocks or originate from a proprietary library, evaluating the ability of models to generalise to unseen chemistry. This balanced metric is more representative in benchmarking molecular representations and architectures that capture transferable binding principles.

Chapter 7

Results and Evaluation

7.1 Ablation Study

Most ablation studies were conducted on a smaller subset of the dataset; experiments on larger subsets are explicitly noted. We found that models with attention required a slightly reduced learning rate (0.00025 vs. 0.0003) to avoid NaN losses during training, whereas configurations without attention trained robustly under the default schedule. The results remain comparable because early stopping was applied in all cases. Consequently, attention-based models are typically trained for more epochs, compensating for the lower learning rate, an effect we consistently observed in practice. Owing to resource and time limitations, we did not perform multiple independent seed runs for every variant; however, we prioritised running several seeds for the primary variants under comparison. Accordingly, the reported MAP scores are shown without standard deviations except where multiple seeds were evaluated.

7.1.1 Attention and Higher Order features

For model variants without attention, we removed the `BiasedAttn` and `FFN` module from the GPS-CC block. For the *CR-R0, CR-R2, CR-BS* variants, the differences in message passing framework are described in Sec. 4.3.1. Tab. 7.1 showed the MAP score of each variant and its standard deviation calculated across five different seed runs.

Table 7.1: Impact of attention and higher order features on MAP score. The MAP score and its standard deviation are calculated across five runs. MAP scores without standard deviation mean that only a single run is performed.

Size	Attention	With Face	Barycentric	Loss	MAP (± std)
1M	X	X	X	ASL	**0.21207 ± 0.01291**
1M	X	✓	X	ASL	0.19026 ± 0.00680
1M	X	✓	✓	ASL	0.18331 ± 0.01315
1M	✓	X	X	ASL	0.18167 ± 0.01540
1M	✓ (w/o bias)	X	X	ASL	0.18915 ± 0.03008
1M	✓	✓	X	ASL	0.17583
1M	✓	✓	✓	ASL	0.17150

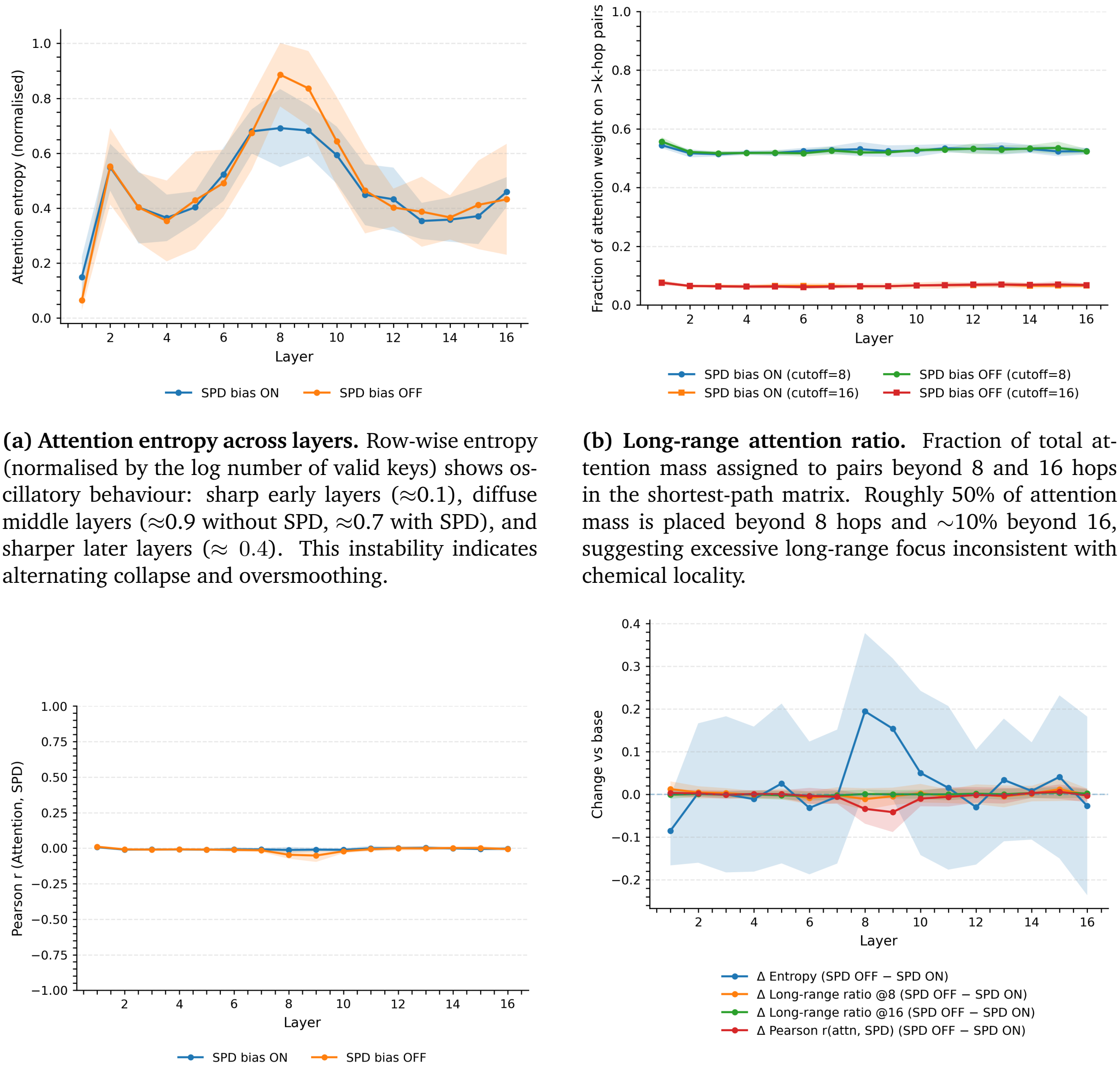
Why No Attention is better than Attention?

We observed that the plain MPNN model without attention and the 2-cell features performed best, with a score of 0.21207 ± 0.01291. In GPS-CC on the BELKA dataset, the MPNN-only variant systematically outperforms versions that include global attention, even when the latter is biased by shortest-path distances. The main reason is that the baseline node and edge features already include rich structural descriptors such as Laplacian eigenvector encodings, random-walk structural features, and centrality measures. These positional encodings effectively provide the model with local and medium-range relational information, exactly what global attention is meant to capture. Adding a dense attention mechanism brings little new information but substantially increases capacity and gradient variance, making the model more challenging to optimise. This is particularly acute as the positive labels are sparse, and some labels are noisy in our dataset. The attention mechanism may overfit to these noises, as it is more sensitive. Moreover, shortest-path distance bias is only a soft prior: when overwhelmed by QK^T magnitudes, it fails to constrain attention, while if scaled too strongly, it collapses the mechanism onto trivial self-neighbours. In either case, the resulting distribution is overly smooth, leading to diluted probability scores and a drop in top-k precision, the component most critical to MAP on BELKA.

In addition, binding propensity in these small molecular graphs is dominated by short-range chemistry (functional groups, ring context, local stereochemistry). The MPNN already encodes true edges; dense self-attention introduces non-physical “virtual edges” that blur locality. On molecules, unconstrained attention frequently routes information across distant atoms that have no chemical interaction, effectively creating shortcuts that break the relational geometry MPNNs respect.

To gain a deeper insight into why the attention mechanism underperforms in GPS-CC on the BELKA dataset, we analysed the internal behaviour of the attention heads using three complementary diagnostics (Fig 7.1). Attention entropy (Fig. 7.1a) measures how concentrated or diffuse each head’s distribution is over neighbours, providing insight into whether the module focuses sharply on a few chemically relevant interactions or spreads weight indiscriminately. Long-range attention ratios (Fig. 7.1b) quantify structural preference by summing the fraction of total attention mass assigned to pairs separated by more than 8 hops (ratio-8) or more than 16 hops (ratio-16) in the shortest-path distance matrix, capturing whether the mechanism balances local bonding information with global structural context. Finally, Pearson correlation between attention weights and shortest-path distances (Fig 7.1c) measures the extent to which attention strength is aligned with graph distance. A strong negative correlation suggests that nearer nodes receive systematically higher attention, aligning with the inductive bias that local interactions dominate in chemistry. Weak or inconsistent correlation indicates the model is not exploiting distance as intended. Using the model configuration without face and barycentric features, these metrics were logged over five forward passes on the test set (batch size 4096) and averaged, capturing the sharpness, spatial scale, and structural grounding of the learned attention patterns.

Chemically meaningful binding interactions often depend on a balance of local context (bonded atoms, rings) and selected longer-range relationships (through-space contacts). As shown in Fig. 7.1a, the entropy of the attention weights fluctuates substantially across layers, starting very low in the early blocks ($\sim$0.2, nearly one-hot distributions), rising to a diffuse regime around



(a) Attention entropy across layers. Row-wise entropy (normalised by the log number of valid keys) shows oscillatory behaviour: sharp early layers (≈0.1), diffuse middle layers (≈0.9 without SPD, ≈0.7 with SPD), and sharper later layers (≈ 0.4). This instability indicates alternating collapse and oversmoothing.

(b) Long-range attention ratio. Fraction of total attention mass assigned to pairs beyond 8 and 16 hops in the shortest-path matrix. Roughly 50% of attention mass is placed beyond 8 hops and ∼10% beyond 16, suggesting excessive long-range focus inconsistent with chemical locality.

(c) Pearson correlation with SPD distance. Correlation between attention weights and shortest-path distances is weak and inconsistent across layers, indicating that the SPD bias does not effectively ground attention in graph structure.

(d) Per-layer differences between models with and without SPD. Changes in entropy, long-range attention ratios (8, 16 hops), and Pearson correlation with SPD (w/o SPD relative to w/ SPD) are shown. Shaded regions indicate propagated standard deviations; the dashed line marks baseline parity.

Figure 7.1: Diagnosis metrics on attention

the middle (∼0.8), and then collapsing again to sharper distributions in later layers (0.3–0.5). This oscillatory behaviour suggests that the softmax distribution is unstable: at shallow depths, attention collapses to a single dominant neighbour, in the middle, it oversmooths by spreading across nearly all nodes, and at deeper depths, it returns to overly sharp focus. Such swings undermine the intended balance between local and global information flow.

A key factor is the relative scale of the dot-product logits and the structural bias. To confirm this, we logged the ranges of raw logits and the SPD bias for two forward passes. We observed that while the SPD bias remains bounded in the range of roughly ± 4 with standard deviation ≈ 1, the raw logits quickly expand in magnitude, reaching standard deviations of 15-20 in intermediate layers and several hundred in deeper ones. Consequently, the bias has a negligible effect where it should provide inductive guidance, and the softmax saturates towards near-deterministic or overly diffuse regimes depending on the layer. This explains both the weak influence of the bias and the observed entropy oscillations.

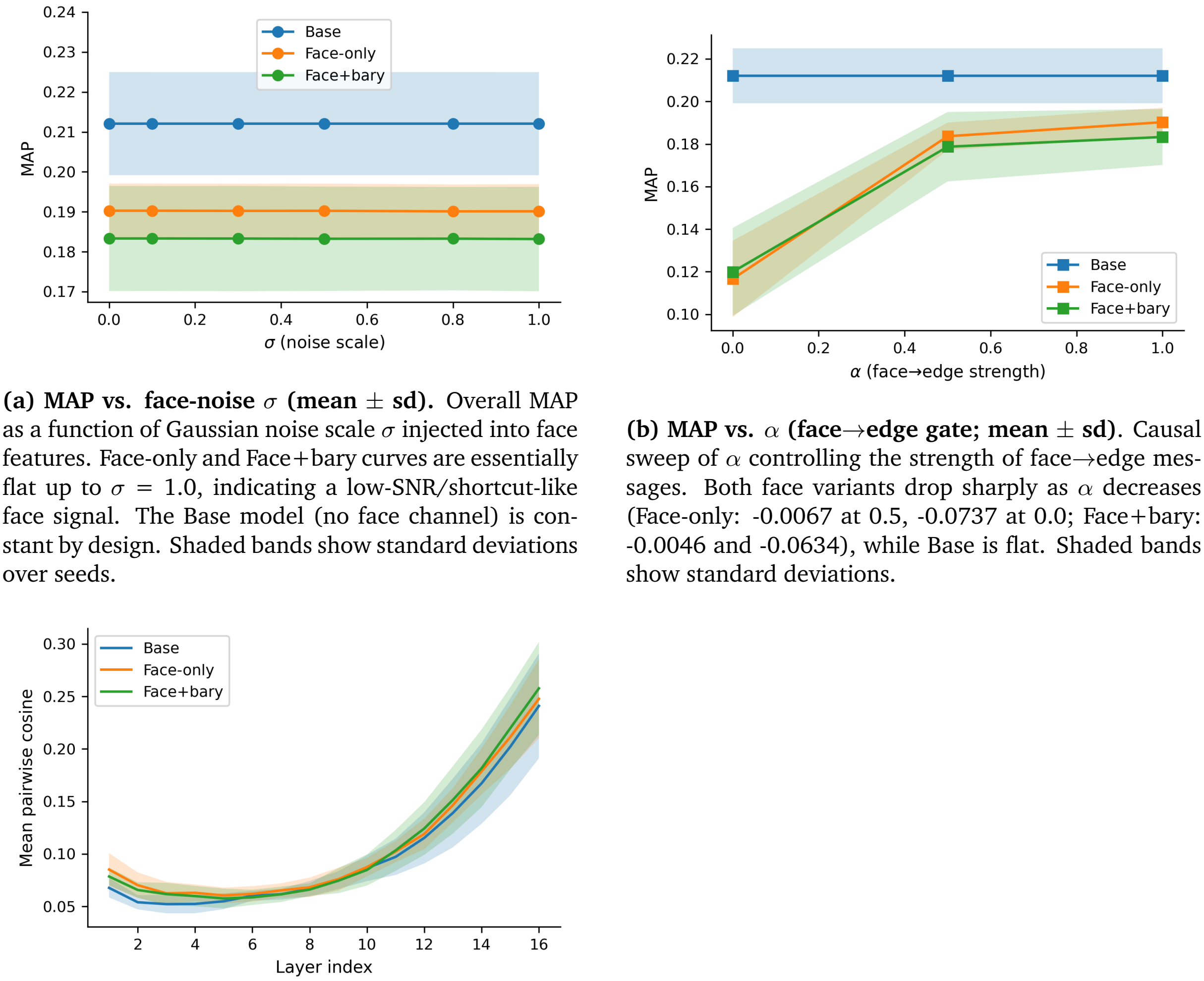
Further evidence comes from the long-range ratio and correlation diagnostics. Fig 7.1b shows that roughly 50% of the total attention mass is placed on pairs more than 8 hops apart, and about 10% extends beyond 16 hops. For molecular graphs, where chemically meaningful interactions are typically local, this excessive long-range focus suggests that attention is not capturing realistic structural dependencies. Meanwhile, Fig 7.1c demonstrates no correlation between the SPD bias values and the learned attention weights, confirming that the intended structural prior does not shape the distribution. Per-layer differences of each diagnosis metric can be seen in Fig. 7.1d.

Finally, comparing runs with and without SPD bias reveals that the overall oscillatory shape of the entropy curve is unchanged: both collapse at shallow layers, diffuse around the middle, and sharpen again at depth. However, without SPD, the middle-layer entropy peaks at nearly 0.9 (almost uniform attention), whereas with SPD, the peak is moderated to about 0.7. Thus, while the bias does act as a weak regulariser of entropy, its effect is too small to improve generalisation. In fact, the MAP score confirms this: the SPD-biased variant underperforms slightly (0.18167 ± 0.01540) compared to the unbiased one (0.18915 ± 0.03008). This suggests that the bias operates more as a mild stabiliser than a meaningful inductive prior, damping the most extreme oversmoothing but not producing useful structural grounding.

Taken together, these results explain why enabling attention worsens performance on BELKA: instead of refining chemically relevant local interactions while selectively integrating global structure, the attention heads amplify indiscriminate long-range signals, leading to oversmoothing and degraded mean average precision compared to the MPNN baseline.

Why plain MPNN is better than MPNN with higher-order topological features?

Here, we compared three architectural variants of the GPS-CC framework on the BELKA dataset: (i) a **Base** model without face features or barycentric subdivision encodings, (ii) a **Face-only** model incorporating ring-level face features, and (iii) a **Face+bary** model combining face features with barycentric subdivision encodings. All models were trained under identical conditions (no attention and asymmetric loss), with equivalent parameter counts and FLOPs, to ensure differences reflected architectural choices rather than capacity.



(a) MAP vs. face-noise σ (mean $\pm$ sd). Overall MAP as a function of Gaussian noise scale σ injected into face features. Face-only and Face+bary curves are essentially flat up to $\sigma = 1.0$, indicating a low-SNR/shortcut-like face signal. The Base model (no face channel) is constant by design. Shaded bands show standard deviations over seeds.

(b) MAP vs. α (face$\rightarrow$edge gate; mean $\pm$ sd). Causal sweep of α controlling the strength of face$\rightarrow$edge messages. Both face variants drop sharply as α decreases (Face-only: -0.0067 at 0.5, -0.0737 at 0.0; Face+bary: -0.0046 and -0.0634), while Base is flat. Shaded bands show standard deviations.

(c) Oversmoothing curves (mean pairwise cosine vs. layer; mean $\pm$ sd). Layer-wise mean pairwise cosine of node embeddings. Base rises slowest (best diversity), Face-only rises faster, and Face+bary fastest (strongest collapse). Final-layer means: 0.241 (Base), 0.248 (Face-only), 0.258 (Face+bary). Shaded bands are standard deviations across variances.

Figure 7.2: Diagnosis metrics on higher-order features

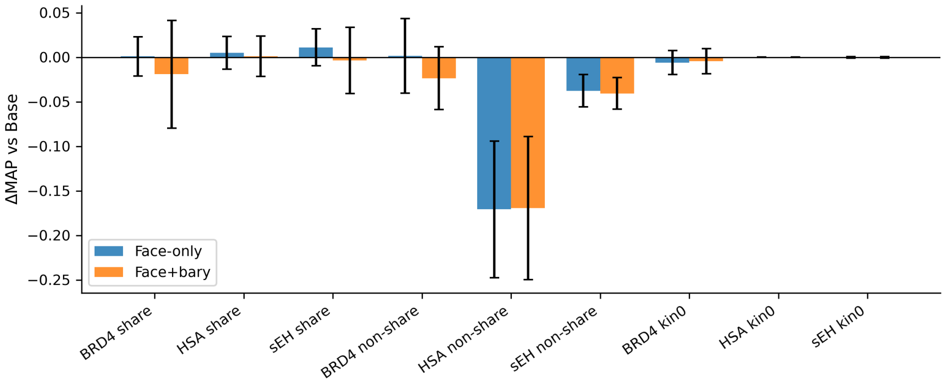


Figure 7.3: Bucket-wise ΔMAP vs. Base (mean ± sd). Per-bucket differences relative to the Base model (bars with error bars). Largest negative deltas occur in **HSA non-share** and **sEH non-share** (≈ -0.15), pinpointing failures under scaffold shift; share and kin0 buckets remain near zero. Error bars combine variances of the compared models in quadrature.

To assess robustness of the face channel, we performed a **face-noise sensitivity sweep**. During evaluation only on the test set, Gaussian noise was added to the face feature embeddings with scale σ relative to the per-feature standard deviation. We swept $\sigma \in \{0.0, 0.1, 0.3, 0.5, 0.8, 1.0\}$, recomputing mean average precision (MAP) at each level. Flat or improving curves indicate that faces encode fragile or shortcut-like information, whereas consistent degradation under noise suggests they carry a stable chemical signal.

We next performed an **α-gate sweep** to test the contribution of face-to-edge message passing causally. A multiplicative factor α was applied to the aggregated face-to-edge messages in the MPNN layer. After training with α=1.0, we re-ran the evaluation with $\alpha \in \{1.0, 0.5, 0.0\}$. If performance remains constant as $\alpha \to 0$, face messages are redundant; a sharp drop indicates that the model internally depends on them, even if this does not yield stronger generalisation.

We logged **cosine similarity per layer** to diagnose representation collapse. After training, models were switched to evaluation mode and run on five test batches. For each layer, node embeddings were normalised and pairwise cosine similarity was computed on a random subset of up to 4,096 nodes. The mean similarity traces oversmoothing: faster rise indicates greater homogenization of node embeddings.

Finally, all metrics were reported using the official **bucket-wise MAP** evaluation. The dataset is partitioned into nine buckets: BRD4 share, HSA share, sEH share, BRD4 non-share, HSA non-share, sEH non-share, BRD4 kin0, HSA kin0, and sEH kin0. For each bucket and target, average precision was computed, and MAP was defined as the mean across all buckets and targets. This ensures equal weighting of easy and hard regimes, and highlights where architectural changes most affect generalisation under scaffold shift.

We observed that across the three variants, the **Base** model without faces or barycentric subdivision encoding achieves the highest MAP (0.21207 ± 0.01291), followed by **Face-only** (0.19026

± 0.00680), with **Face+bary** lowest (0.18331 ± 0.01315). This ordering is consistent across seeds and sets the backdrop for the mechanistic probes.

Fig. 7.2a shows the MAP of each variant against increasing face noise. Injecting Gaussian noise into face features up to $\sigma = 1.0$ leaves performance essentially unchanged: Face-only goes from 0.19026 at $\sigma = 0.0$ to 0.19013 at $\sigma = 1.0$, Face+bary goes from 0.18331 at $\sigma = 0.0$ to 0.18317 at $\sigma = 1.0$. The flat curves indicate that, although the models *use* the face pathway, it carries **low-SNR/shortcut-like** signal that random perturbations do not disrupt. The Base model is unaffected by construction.

Fig. 7.2b shows the MAP of each variant against changes in α. Gating the face$\to$edge messages (α) shows strong dependence for both face variants while leaving the Base model invariant. Face-only degrades from 0.19036 ($\alpha = 1.0$) to 0.18365 ($\alpha = 0.5$) and 0.11664 ($\sigma = 0.0$). Face+bary drops from 0.18331 $\to$ 0.17873 $\to$ 0.11995. Thus, the network relies on the face pathway, but that reliance does not translate into better generalisation (per the flat σ sweep).

Fig. 7.2c shows the cosine similarity of each layer. Mean pairwise cosine similarity across layers rises slowest for Base and fastest for Face+bary (final-layer means: 0.24103 (Base) $<$ 0.24784 (Face-only) $<$ 0.25774 (Face+bary)). This indicates that explicit face features, and especially face+bary, **accelerate representation collapse**, aligning with the MAP ordering.

Fig. 7.3 shows the changes in MAP score for each protein class and split group for all three variants. Bucket-wise ΔMAP relative to Base shows the largest deficits in the **HSA non-share** and **sEH non-share** buckets (≈ -0.15 for both face variants), i.e., the hardest scaffold-shift regimes. Share and kin0 buckets are near-neutral with only small ± wiggles. This localisation explains the leaderboard shake-ups: the face channel encourages **non-transferable shortcuts** that fail under distribution shift, and barycentric grounding further increases mixing (and oversmoothing) without recovering OOD accuracy.

Although our ablation results ultimately favour the baseline model without faces or barycentric encodings, there is clear chemical motivation for exploring these higher-order features. Face attributes provide explicit 2-cell information corresponding to rings, fused aromatic systems, and other cyclic motifs, while barycentric subdivision encodings ground these faces within a multiscale positional basis. These structures are known to play key roles in protein–ligand binding, particularly for recognition of aromatic scaffolds and heteroatom-rich cycles, and are only partially captured by classical atom, bond, and positional encodings. In principle, such priors should strengthen inductive bias toward chemically meaningful motifs. However, in the BELKA dataset the added signals appear noisy or redundant with simpler aromaticity flags, and they exacerbate oversmoothing under scaffold shift, leading to the observed performance degradation relative to the node–edge backbone. This highlights the tension between incorporating chemically motivated higher-order structure and maintaining transferable representations across diverse scaffolds.

In conclusion, all probes agree that (i) Base generalises best by avoiding a fragile face channel and preserving embedding diversity; (ii) Face-only provides some usable ring-level signal (α-dependent) but remains noise-insensitive and mildly oversmoothed; (iii) Face+bary oversmooths the most and underperforms particularly on non-share buckets. Overall, the evidence supports the consistent ranking Base $>$ Face-only $>$ Face+bary and attributes the gap primarily to **oversmoothing and shortcutting** introduced by the face pathway, amplified by barycentric encodings.

When extending these comparisons to attention-enabled variants, the same trend persists but with further degradation. In line with our earlier observation that attention alone reduces performance in GPS-CC, adding faces or faces with barycentric subdivision on top of attention compounds this effect. The interaction between attention and higher-order features appears particularly detrimental: the combination of attention with face features already lowers MAP, while attention with both face and barycentric encodings performs worst overall. This outcome is consistent with our mechanistic probes, which showed that attention accelerates oversmoothing and amplifies indiscriminate long-range signals. Thus, when attention is active, both face features and barycentric subdivision act as amplifiers of these weaknesses, leading to the steepest declines in generalisation under scaffold shift.

Notes on training stability

The standard deviations further highlight differences in training stability across settings. Without attention, the baseline model (no face, no barycentric, asymmetric loss) achieves both the highest mean MAP and a moderate spread (±0.013). Adding face features alone reduces the mean MAP but yields very consistent performance (±0.007), indicating that the degradation is systematic rather than stochastic. By contrast, combining face and barycentric features lowers the mean further and increases the variance (±0.013), making the results less stable. When attention is introduced, the variance rises substantially (±0.015), and it becomes even larger in the absence of the SPD bias (±0.030). Interestingly, the SPD bias lowers the mean MAP but clearly improves reproducibility, showing a trade-off between accuracy and stability.

7.1.2 Loss Function

Table 7.2: Impact of loss function choice on MAP score

Size	Attention	With Face	Barycentric	Loss	MAP
1M	X	X	X	ASL	**0.21207 ± 0.01291**
1M	X	X	X	ASL+CL	0.17923
1M	X	X	X	Focal	0.17772
1M	X	X	X	BCE	0.14935

Tab. 7.2 showed the MAP score for different loss functions but the same model configurations (no attention, no 2-cell features, no barycentric). Asymmetric Loss (ASL) achieves the strongest performance, 0.21207, on BELKA because it directly addresses extreme class imbalance. Unlike balanced binary cross-entropy (BCE), which only reweights positive and negative samples globally, ASL introduces decoupled focusing parameters (γ^+, γ^-) and a negative margin, m, that selectively down-weights easy negatives while preserving rare positive gradients [81]. This ensures that model updates concentrate on hard positives and confusing negatives near the decision boundary, improving precision in the top-ranked predictions (region that Mean Average

Precision (MAP) emphasises most strongly).

Adding Center Loss (CL) on positives alone provides an additional regularisation signal by pulling binders closer in embedding space [82]. This yields a modest improvement compared to focal loss, since it enforces some cohesion among positives and prevents excessive dispersion. However, BELKA positives are highly heterogeneous across scaffolds and binding modes. A single class-level centre thus collapses distinct positive sub-modes, reducing structural diversity without pushing negatives further away. As a result, ASL alone maintains more discriminative margins, while ASL+CL trades some of that fine separation for intra-positive compactness that does not directly benefit ranking. Moreover, under mixed-precision training, the combination is less numerically stable: scarce positive updates make the centre estimates noisy (high-variance gradients), which interacts poorly with gradient scaling and requires a lower learning rate (we used 0.0002, with the same warmup and decay schedule) to avoid NaN losses.

Between the weaker baselines, focal loss is more effective than balanced BCE because it adaptively focuses learning on hard examples rather than simply rescaling classes. This prevents the model from being dominated by the vast majority of easy negatives, improving the recall of rare binders [80]. Still, focal loss lacks ASL's asymmetry and margin shift, so ASL and ASL+CL outperform it. Finally, balanced BCE performs the worst because static weighting is too crude for BELKA's extreme imbalance and fails to concentrate learning where it matters for ranking.

7.1.3 Scaling

Table 7.3: Impact of dataset subset size on MAP score. MAP score for 1M and its standard deviation is calculated across five runs, whereas the 10M is calculated across two runs.

Size	Attention	With Face	Barycentric	Loss	MAP
1M	X	X	X	ASL	0.21207 ± 0.01291
10M	X	X	X	ASL	**0.24088 ± 0.00329**

We compared the model's performance trained on the 1 million and 10 million data subsets. The configurations are shown in Tab. 7.3. The 1 million subset achieved an average MAP score of 0.21207 (± 0.01291) over five runs, whereas the 10 million subset achieved an average MAP score of 0.24088 (± 0.00329) over two runs, showing a clear benefit from scaling by an order of magnitude.

To evaluate how representative our current **subset (10 M)** is relative to both the **full training pool** and the **test set**, we assess representativeness in two complementary ways: (i) *scaffold overlap*, which measures coverage at the chemotype level, and (ii) *fingerprint-level similarity*, using maximum mean discrepancy (MMD^2) and average nearest-neighbour (AvgNN) Tanimoto similarity. Together, these metrics indicate whether the subset faithfully captures the diversity and distribution of the full dataset, and allow us to estimate the potential performance improvements that might be gained by scaling to the full 98M molecules.

First, we preprocessed the molecules by normalising and removing the stand-in DNA linker [Dy], converted to canonical SMILES, and featurized into ECFP-4 fingerprints (2048 bits, radius 2, chiral) using RDKit. Each set was cached with fingerprints, popcounts, and Murcko scaffolds. **Scaffold overlap** metrics were computed exactly as set intersections. **Average nearest-neighbour (AvgNN) Tanimoto** similarity was estimated by sampling 20,000 queries and comparing against a capped reference of 1–2 M molecules in 32k blocks. **Maximum Mean Discrepancy (MMD^2)** was estimated under the Tanimoto kernel by subsampling 50k molecules per set and averaging over 1.5 million random pairs per expectation (self, self, cross).

Table 7.4: Comparison of 1M and 10M training subsets to the full 100M pool and the full test set. Values show scaffold overlap (coverage and Jaccard), MMD^2 in fingerprint space, and AvgNN Tanimoto similarity.

Pair	Cov. A∈B	Cov. B∈A	Jaccard	MMD^2	AvgNN A→B	AvgNN B→A
Test vs Training						
Test vs 1M subset	0.114	0.091	0.054	0.063	0.436	0.669
Test vs 10M subset	0.170	0.031	0.027	0.063	0.436	0.669
Test vs Full (100M)	0.193	0.010	0.010	0.063	0.445	0.669
Subset vs Full						
1M vs Full	1.000	0.067	0.067	≈0	0.767	0.743
10M vs Full	1.000	0.294	0.294	≈0	0.768	0.743

Tab. 7.4 showed the comparison of 1M and 10M training subsets to the full training and test set in terms of scaffold overlap and fingerprint-level similarity using Jaccard, MMD^2 and AvgNN. The comparisons reveal that both the 1 M and 10 M subsets are indistinguishable from the full 100 M pool in fingerprint space (subset–full MMD^2 ≈ 0), and their distance to the test set is the same (MMD^2 ≈ 0.063 for both test–subset and test–full). However, scaffold overlaps tell a different story: the 1 M subset covers only 6.7% of the full's scaffolds, while the 10 M subset covers 29.4%. This difference is also visible in overlaps with the test set: the Jaccard index is 5.4% for test–1 M subset but only 2.7% for test–10 M subset (full is 1.0%), reflecting that while the larger subset covers a broader portion of training chemotypes, the test introduces many scaffolds absent from both. AvgNN results are consistent across subset sizes: test molecules find moderately close neighbours in training (mean ≈ 0.44), while subsets and full are much closer to each other (mean ≈ 0.75). Uncertainty (±) reflects the standard error of the estimate. For AvgNN, this is $\mathrm{SE}(\bar{s}) = \hat{\sigma}/\sqrt{N}$ with $\hat{\sigma}^2 = \frac{1}{N-1}\sum_i (s_i - \bar{s})^2$. For MMD^2, we approximate SE using per-sample kernel contributions a_i, b_j as $\mathrm{SE}\big(\mathrm{MMD}^2\big) \approx \sqrt{\hat{\sigma}_a^2/m + \hat{\sigma}_b^2/n}$. With $N, m, n \sim 10^6$, the resulting errors are roughly on the order of 10^{-3}, so these conclusions are robust.

In practice, this indicates that scaling from 1 M to 10 M already yields tangible MAP improvements due to better scaffold coverage, and moving to the full 100 M should continue this trend but with diminishing returns. Since the 10 M subset already captures most of the distributional properties of the full dataset, further gains from scaling are expected to come mainly from exposing the model to rarer scaffolds and chemotypes. Given the MAP of 0.24088 on 10 M, one might expect only modest but real improvements from training on all 100 M molecules—likely on the order of **0.01–0.02 absolute MAP** ($\approx 5 - 10\%$ relative), primarily through improved generalisation to novel scaffolds. rather than shifts in overall fingerprint statistics.

7.2 Generalisation

Tab. 7.5 and 7.6 showed the mean and standard deviation of the AP score for each protein and split group across five runs for the model without an attention mechanism and with an attention mechanism (with bias), respectively. 2-cell features and barycentric subdivision encodings were disabled, and we used ASL as the loss function.

Table 7.5: Tab showed the mean average precision (MAP) score and its standard deviation for each protein class and split group across five random seeds with attention **disabled**. The final row reported the MAP for each subgroup, and the corresponding standard deviation measures the variability between the three protein class means.

Protein Class	Share	Non-share	kin0
BRD4	0.42044 ± 0.01403	0.07503 ± 0.03196	0.00983 ± 0.01302
HSA	0.24633 ± 0.01245	0.29617 ± 0.07516	0.00118 ± 0.00006
sEH	0.73958 ± 0.01996	0.11829 ± 0.01526	0.00173 ± 0.00081
Mean	0.46878 ± 0.25015	0.16316 ± 0.11720	0.00425 ± 0.00484

Table 7.6: Tab showed the mean average precision (MAP) score and its standard deviation for each protein class and split group across five random seeds with attention **enabled**. The final row reported the MAP for each subgroup, and the corresponding standard deviation measures the variability between the three protein class means.

Protein Class	Share	Non-share	kin0
BRD4	0.32490 ± 0.05310	0.04412 ± 0.01585	0.00265 ± 0.00069
HSA	0.18865 ± 0.04213	0.25776 ± 0.09858	0.00120 ± 0.00011
sEH	0.68298 ± 0.03721	0.13074 ± 0.02930	0.00205 ± 0.00051
Mean	0.39884 ± 0.25532	0.14421 ± 0.10746	0.00197 ± 0.00073

The results in Tab 7.5 and 7.6 indicate that the model achieves its highest predictive performance in the *Share* group, where ligands share the same scaffold and building blocks with the training data. Performance drops substantially in the *Non-share* group, where only the scaffold is preserved, and it is almost negligible in the *kin0* group, where ligands share no structural similarity to the training compounds. This trend highlights the limited ability of the model to generalise beyond close analogues of the training data: strong predictive accuracy is largely confined to compounds with high structural overlap, while extrapolation to novel chemical space remains challenging. The consistently low variance across runs further suggests that this behaviour is systematic rather than stochastic.

For BRD4, the model performs reasonably well in the *Share* group. Still, its predictive accuracy collapses almost entirely in the *kin0* group, suggesting that the model relies heavily on close structural analogues to capture BRD4 binding patterns [90, 91].

For HSA, performance is more balanced across *Share* and *Non-share* groups, with the highest MAP in the latter. This reflects the fact that HSA binding is often driven by broader physicochemical properties (e.g., lipophilicity, hydrophobic interactions) rather than precise substructural motifs, allowing the model to generalise better when only the core is preserved [92].

However, generalisation still fails when neither scaffold nor building blocks are shared.

For sEH, the model shows the strongest performance overall, with MAP exceeding 0.73 in the *Share* group. While accuracy drops in the *Non-share* group, the relative retention of signal compared to BRD4 indicates that sEH binding determinants are somewhat more transferable across related scaffolds [93], or it could be due to stochastic variation in downsampling. Yet again, the near-zero MAP in the *kin0* group confirms that extrapolating entirely novel chemotypes remains extremely challenging.

Another interesting observation is that the degradation patterns differ between attention, face and face+barycentric variants. With attention enabled, the decline in MAP is broad and consistent across all buckets (Tab. 7.6), supporting the interpretation that attention induces general oversmoothing rather than failing in specific regimes. In contrast, the face or face+barycentric variant shows a heterogeneous effect: performance in the share buckets is marginally higher (face) than or similar (face+barycentric) to the baseline, though the improvement lies within the standard deviation and may reflect noise, while in the non-share buckets the degradation is substantial. As confirmed by Fig. 7.3, these larger losses offset any gains, leading to worse overall generalisation. This suggests that face and barycentric encodings act as weak inductive hints that can help in-distribution cases, but behave as scaffold-specific shortcuts that harm transfer to unseen scaffolds.

Taken together, these observations reinforce that the model's generalisation capacity is highly dependent on the nature of the binding target: for proteins like BRD4, where binding relies on specific hydrogen-bonding motifs [90, 91], predictive power is scaffold-restricted; for HSA, where global physicochemical properties largely govern binding [92], extrapolation is slightly more forgiving; and for enzymes like sEH, where active-site features admit a broader range of chemotypes [93], partial generalisation is possible. At the same time, our ablations show that architectural choices modulate these trends: attention induces a broad decline consistent with oversmoothing, while face and barycentric encodings provide weak in-distribution benefits but act as scaffold-specific shortcuts that harm transfer to novel scaffolds. These findings highlight that while current ML models can achieve impressive performance on seen datasets, their predictive power does not reliably extend to novel targets or diverse chemical scaffolds. Addressing this generalisation gap by improving coverage of chemical space, refining inductive biases, and incorporating mechanistic insights into ligand–protein recognition remains a central challenge for advancing robust and transferable affinity prediction models [10].

7.3 Discussion

Across ablations, the **MPNN-only** backbone with ASL is consistently strongest (Tab. 7.1, 7.2), while **global attention** and **higher-order (face/barycentric)** channels degrade MAP via two related mechanisms: (i) dense attention introduces non-physical "virtual edges" that amplify indiscriminate long-range flow (Fig. 7.1b), and (ii) both attention and face+bary accelerate *representation collapse* (Fig. 7.2c), reducing discriminability under scaffold shift. SPD bias mildly stabilises entropy (Fig. 7.1a) but does not anchor attention to distance (Fig. 7.1c) nor improve MAP. Scaling from 1M to 10M improves MAP primarily through better scaffold coverage (Tab. 7.4); however, generalisation remains strongly scaffold-dependent, with performance collapsing in *kin0* (Tab. 7.5, 7.6).

Beyond reporting scores, this study introduces a targeted diagnostic suite: *entropy trajectories*, *long-range attention ratios*, *distance–attention correlations*, *logit/bias scale logging* and *causal face interventions* (σ-noise and α-gating). These reveal *why*, contrary to common practice, attention and higher-order channels do not help on BELKA: (a) **local chemistry dominates** ranking, so unconstrained dense mixing blurs useful short-range cues; (b) **logit–bias mismatch** (large QK vs. bounded SPD) drives unstable softmax regimes and weak structural grounding; (c) **shortcut-prone faces** are heavily relied upon (sensitive to α) yet noise-insensitive (σ-flat), consistent with failure under scaffold shift. These patterns suggest that *local, chemically grounded* message passing with careful regularisation is preferable to adding global channels that are hard to constrain on BELKA.

Finally, when viewed against the BELKA Kaggle leaderboard, our results appear competitive despite being trained on a fraction of the data. In the BELKA Kaggle competition, the test set was split into 35% public and 65% private, with the *kin0* group absent from the public split. Because the exact indices used are unavailable, we cannot directly compare to the leaderboard. However, since our evaluation covers the whole test set, we expect the performance to represent where we would stand on the private leaderboard. The official competition revealed large rank shifts between public and private test sets, highlighting the instability of many high-performing submissions. By contrast, our models display consistent behaviour across subsets: with only $\sim 1\%$ and $\sim 10\%$ of the data, we already achieve MAP scores of 0.21207 and 0.24088. Given that the most stable leaderboard entries scored in the 0.27729–0.28557 range and the top model achieved 0.30619 [94], this trajectory suggests that scaling to the full dataset could yield state-of-the-art performance while preserving robustness, addressing precisely the variance and overfitting issues that diagnostics here identify in attention and higher-order channels.

Chapter 8

Future Work and Conclusion

8.1 Limitations

This study is constrained first by data and evaluation. Due to resource limits, models were trained on 1M/10M subsets rather than the full ~ 100M pool. While fingerprint-space MMD^2 indicates these subsets resemble the full pool (Tab. 7.4), scaffold coverage remains limited (6.7% for 1M; 29.4% for 10M), reducing exposure to rare chemotypes. The validation split is small and not scaffold-balanced, so early stopping may track the validation scaffold mix more than true out-of-distribution behaviour. Moreover, the benchmark enforces a hard scaffold shift; MAP in *kin0* is near zero, so absolute performance on entirely novel chemical space should be interpreted cautiously.

A second limitation arises from modelling assumptions. The approach is ligand-only and therefore protein-agnostic: we do not incorporate pocket geometry or sequence/structure embeddings, which restricts target-conditional reasoning (e.g., differences between BRD4 and HSA). Ligand descriptors are primarily 2D (atom/bond/positional encodings); without 3D conformers, partial charges, or explicit geometry, through-space contacts and stereospecific effects are only weakly captured.

Finally, there are optimisation and metric limitations. When enabled, attention logits overwhelm the SPD bias, producing unstable softmax regimes (Fig. 7.1a); we did not systematically explore mitigations such as temperature/scale clipping, alternative normalisation, or sparsification. To keep comparisons fair, we matched parameters/FLOPs and kept tuning modestly, so alternatives (different normalisers or residual gating, depth/width trade-offs, or MAP-oriented surrogates) were not exhaustively assessed. MAP is our primary endpoint; conclusions might differ for enrichment and calibration–focused objectives (e.g., $\mathrm{EF}_{1\%}$, Brier) [95, 96]. Lastly, positives in BELKA are sparse and heterogeneous, and potential label noise may interact unfavourably with high-capacity components; we did not perform explicit noise-robust training or label auditing.

8.2 Future Work

Several promising avenues remain for future research. First, scaling training from the current 10M subset to the full ~ 100M pool would broaden scaffold coverage and increase exposure to rarer chemotypes. At the same time, evaluation protocols could be tightened through scaffold-balanced validation or cross-validation, as well as group-aware sampling or reweighting strategies. Such refinements would help ensure that early stopping and model selection align with true out-of-distribution behaviour, particularly relevant to robustness in challenging subsets such as *kin0*.

On the modelling side, future work could explore alternatives to global mean pooling. Gated or attention-based pooling layers with explicit locality constraints (e.g., degree/distance gates or top-k sparsity) may prevent the creation of virtual long-range shortcuts. Similarly, moderate depth combined with stronger regularisation such as residual gating, PairNorm [97], or GraphNorm [98] may mitigate oversmoothing. For attention mechanisms, constrained variants (e.g., scale clipping of QK logits or distance-aware sparse masks with learnable SPD ceilings) may yield a more balanced trade-off between expressivity and stability. Higher-order structural information could also be incorporated through learned motif or ring aggregators, particularly when paired with anti-smoothing penalties and the diagnostic tools developed in this work.

A further direction is to incorporate target-side information without requiring full docking. Candidate approaches include pocket graphs or pretrained protein sequence/structure embeddings, integrated via ligand–pocket cross-attention. Lightweight geometric surrogates such as distance-geometry conformers, partial charges, or ring planarity may additionally provide low-cost ways of encoding through-space effects.

Finally, optimisation and objectives may be better aligned with ranking-based evaluation. MAP-oriented or listwise losses, together with broader sweeps of ASL hyperparameters (γ^+, γ^-, m), could provide more effective training signals. Robustness could be enhanced through techniques such as negative-label smoothing, co-teaching, calibration (e.g., temperature scaling), or consistency regularisation. Pretraining remains an underexplored opportunity: masked atom/bond prediction, contrastive graph learning, and multi-target training are all potential strategies for transferring general chemical regularities prior to BELKA fine-tuning.

Overall, these directions suggest a rich landscape of opportunities for advancing ligand-only models in protein–ligand interaction prediction, both by extending dataset scale and by developing architectures and training strategies that better capture chemical and structural complexity.

8.3 Conclusion

This work systematically evaluates GPS-CC variants on BELKA under controlled capacity and training schedules. Three conclusions emerge. **First**, a *plain MPNN* with rich but local positional/structural encodings and *ASL* loss achieves the strongest and most stable MAP. Adding *global attention* or *higher-order face/barycentric subdivision* channels consistently reduces performance, largely via oversmoothing and the introduction of non-physical long-range shortcuts; SPD bias stabilises attention entropy but does not translate to accuracy gains. **Second**, *scaling* from 1M to 10M improves MAP by covering more scaffolds; nonetheless, generalisation remains scaffold-dependent, with performance collapsing in *kin0*, highlighting the difficulty of extrapolating to novel chemotypes. **Third**, the gap to robust OOD performance appears driven less by raw model capacity and more by *representation control and target conditioning*: enforcing locality, resisting collapse, and injecting protein context are likely to yield the largest returns.

Beyond ranking variants, we contribute a *causal+diagnostic* account of *why* attention and higher-order channels underperform on BELKA: QK logits overwhelm bounded SPD priors (logit–bias mismatch), producing unstable attention and weak distance alignment; face/bary pathways act as shortcut features, models *rely* on them (sensitive to α) yet they are *noise-insensitive* (flat in σ), consistent with failure under scaffold shift. These findings advocate for architectures that prioritise chemically grounded locality and regularised information flow, complemented by larger and more representative training data. Future extensions such as training on the full dataset, replacing global mean pooling with gated or sparse alternatives, integrating protein embeddings, modestly reducing depth with stronger normalisation/gating, and exploring MAP-aligned objectives offer a practical path toward improving generalisation under scaffold shift on BELKA.

Finally, when set against the original BELKA Kaggle leaderboard, our results appear competitive despite being trained on only a fraction of the data. The competition showed significant rank shifts between public and private test sets, indicating instability in many high-performing entries. By contrast, our models exhibit consistent behaviour across subsets: with only $\sim 1\%$ and $\sim 10\%$ of the training pool, we already achieve MAP scores of 0.21207 and 0.24088, compared to 0.277–0.286 for the most stable leaderboard entries and 0.306 for the top model. This trajectory suggests that scaling our approach to the full dataset could approach state-of-the-art performance while maintaining robustness, addressing the variance and instability issues plaguing existing leaderboard solutions.

Bibliography

[1] Yazdani-Jahromi M, Yousefi N, Tayebi A, Kolanthai E, Neal CJ, Seal S, et al. AttentionSiteDTI: an interpretable graph-based model for drug-target interaction prediction using NLP sentence-level relation classification. Briefings in Bioinformatics. 2022;23(4):bbac272. pages 1, 2

[2] Yang Y, Cheng F. Artificial intelligence streamlines scientific discovery of drug–target interactions. British Journal of Pharmacology. 2025. pages 1, 2

[3] Morris GM, Lim-Wilby M. Molecular docking. Molecular modeling of proteins. 2008:365-82. pages 1

[4] Pinzi L, Rastelli G. Molecular docking: shifting paradigms in drug discovery. International journal of molecular sciences. 2019;20(18):4331. pages

[5] Wikipedia. Docking (molecular); 2025. Accessed: 2025-05-22. https://en.wikipedia.org/wiki/Docking_(molecular). pages 1

[6] Grinter SZ, Zou X. Challenges, applications, and recent advances of protein-ligand docking in structure-based drug design. Molecules. 2014;19(7):10150-76. pages 1

[7] Lexa KW, Carlson HA. Protein flexibility in docking and surface mapping. Quarterly reviews of biophysics. 2012;45(3):301-43. pages 1

[8] Corso G, Stärk H, Jing B, Barzilay R, Jaakkola T. Diffdock: Diffusion steps, twists, and turns for molecular docking. arXiv preprint arXiv:221001776. 2022. pages 1

[9] Pei Q, Gao K, Wu L, Zhu J, Xia Y, Xie S, et al. Fabind: Fast and accurate protein-ligand binding. Advances in Neural Information Processing Systems. 2023;36:55963-80. pages 1, 7

[10] Sim J, Kim D, Kim B, Choi J, Lee J. Recent advances in AI-driven protein-ligand interaction predictions. Current Opinion in Structural Biology. 2025;92:103020. pages 1, 2, 52

[11] Lai H, Wang L, Qian R, Huang J, Zhou P, Ye G, et al. Interformer: an interaction-aware model for protein-ligand docking and affinity prediction. Nature Communications. 2024;15(1):10223. pages 1

[12] Lin H, Zhu J, Wang S, Li Y, Pei J, Lai L. DeepRLI: a multi-objective framework for universal protein–ligand interaction prediction. Digital Discovery. 2025. pages 1

[13] Moon S, Zhung W, Yang S, Lim J, Kim WY. PIGNet: a physics-informed deep learning model toward generalized drug–target interaction predictions. Chemical Science. 2022;13(13):3661-73. pages 1

[14] Moon S, Hwang SY, Lim J, Kim WY. PIGNet2: a versatile deep learning-based protein–ligand interaction prediction model for binding affinity scoring and virtual screening. Digital Discovery. 2024;3(2):287-99. pages 1

[15] Morehead A, Cheng J. Flowdock: Geometric flow matching for generative protein-ligand docking and affinity prediction. ArXiv. 2025:arXiv-2412. pages 2

[16] Abramson J, Adler J, Dunger J, Evans R, Green T, Pritzel A, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature. 2024;630(8016):493-500. pages 2

[17] Öztürk H, Özgür A, Ozkirimli E. DeepDTA: deep drug–target binding affinity prediction. Bioinformatics. 2018;34(17):i821-9. pages 2

[18] Chen L, Tan X, Wang D, Zhong F, Liu X, Yang T, et al. TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. Bioinformatics. 2020;36(16):4406-14. pages 2

[19] Nguyen T, Le H, Quinn TP, Nguyen T, Le TD, Venkatesh S. GraphDTA: predicting drug–target binding affinity with graph neural networks. Bioinformatics. 2021;37(8):1140-7. pages 2

[20] Diego US. UniProt Accession ID; 2025. Accessed: 2025-05-28. `https://www.bindingdb.org/rwd/bind/ByUniProtids.jsp`. pages 2

[21] Jiang M, Li Z, Zhang S, Wang S, Wang X, Yuan Q, et al. Drug–target affinity prediction using graph neural network and contact maps. RSC advances. 2020;10(35):20701-12. pages 2

[22] Zheng S, Li Y, Chen S, Xu J, Yang Y. Predicting drug–protein interaction using quasi-visual question answering system. Nature Machine Intelligence. 2020;2(2):134-40. pages 2

[23] Dwivedi VP, Rampášek L, Galkin M, Parviz A, Wolf G, Luu AT, et al. Long range graph benchmark. Advances in Neural Information Processing Systems. 2022;35:22326-40. pages 3

[24] Thakoor S, Tallec C, Azar MG, Azabou M, Dyer EL, Munos R, et al. Large-scale representation learning on graphs via bootstrapping. arXiv preprint arXiv:210206514. 2021. pages 3

[25] Bio L. NeurIPS 2024 - Predict New Medicines with BELKA; 2025. Accessed: 2025-05-31. `https://www.kaggle.com/competitions/leash-BELKA`. pages 3, 11, 12

[26] Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: International conference on machine learning. PMLR; 2017. p. 1263-72. pages 5

[27] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, et al. Graph neural networks: A review of methods and applications. AI open. 2020;1:57-81. pages 5

[28] Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y, et al. Graph attention networks. stat. 2017;1050(20):10-48550. pages 5

[29] Reiser P, Neubert M, Eberhard A, Torresi L, Zhou C, Shao C, et al. Graph neural networks for materials science and chemistry. Communications Materials. 2022;3(1):93. pages 5

[30] Wang Y, Li Z, Barati Farimani A. Graph neural networks for molecules. In: Machine learning in molecular sciences. Springer; 2023. p. 21-66. pages 6

[31] Thomas N, Smidt T, Kearnes S, Yang L, Li L, Kohlhoff K, et al. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. arXiv preprint arXiv:180208219. 2018. pages 6

[32] Fuchs F, Worrall D, Fischer V, Welling M. Se (3)-transformers: 3d roto-translation equivariant attention networks. Advances in neural information processing systems. 2020;33:1970-81. pages

[33] Satorras VG, Hoogeboom E, Welling M. E (n) equivariant graph neural networks. In: International conference on machine learning. PMLR; 2021. p. 9323-32. pages 6

[34] Papillon M, Sanborn S, Hajij M, Miolane N. Architectures of topological deep learning: A survey of message-passing topological neural networks. arXiv preprint arXiv:230410031. 2023. pages 7, 8

[35] Wikipedia. Topological deep learning; 2025. Accessed: 2025-05-31. `https://en.wikipedia.org/wiki/Topological_deep_learning`. pages 7

[36] Barsbey M, Ballester R, Demir A, Casacuberta C, Hernández-García P, Pujol-Perich D, et al. Higher-Order Molecular Learning: The Cellular Transformer. In: ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design; 2025. . pages 7, 8, 9, 17

[37] Ballester R, Hernández-García P, Papillon M, Battiloro C, Miolane N, Birdal T, et al. Attending to Topological Spaces: The Cellular Transformer. arXiv preprint arXiv:240514094. 2024. pages 18, 20, 26

[38] Battiloro C, Tec M, Dasoulas G, Audirac M, Dominici F, et al. E (n) equivariant topological neural networks. arXiv preprint arXiv:240515429. 2024. pages 7, 13, 17

[39] Bodnar C, Frasca F, Otter N, Wang Y, Lio P, Montufar GF, et al. Weisfeiler and lehman go cellular: Cw networks. Advances in neural information processing systems. 2021;34:2625-40. pages 7, 8, 13, 27, 28, 29, 30

[40] Barbarossa S, Sardellitti S. Topological signal processing over simplicial complexes. IEEE Transactions on Signal Processing. 2020;68:2992-3007. pages 9

[41] Wachs ML. Poset topology: tools and applications. arXiv preprint math/0602226. 2006. pages 9, 10

[42] Wikipedia. Subdivision (simplicial complex); 2025. Accessed: 2025-08-11. `https://en.wikipedia.org/wiki/Subdivision_(simplicial_complex)`. pages 9

[43] Wikipedia. Barycentric subdivision; 2025. Accessed: 2025-08-11. `https://en.wikipedia.org/wiki/Barycentric_subdivision`. pages 10

[44] Bemis GW, Murcko MA. The properties of known drugs. 1. Molecular frameworks. Journal of medicinal chemistry. 1996;39(15):2887-93. pages 10

[45] Rogers D, Hahn M. Extended-connectivity fingerprints. Journal of chemical information and modeling. 2010;50(5):742-54. pages 10

[46] Polaris. leash-bio/BELKA-v1; 2025. Accessed: 2025-05-31. `https://polarishub.io/datasets/leash-bio/belka-v1`. pages 11, 12, 65

[47] Quigley IK, Blevins A, Halverson BJ, Wilkinson N. Belka: The big encoded library for chemical assessment. In: NeurIPS 2024 Competition Track; 2024. . pages 11, 65

[48] Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of chemical information and computer sciences. 1988;28(1):31-6. pages 12

[49] Daylight Chemical Information Systems I. SMILES - A Simplified Chemical Language; 2025. Accessed: 2025-05-31. `https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html`. pages 12, 13

[50] Hatcher A. Algebraic Topology. Cambridge University Press; 2002. pages 13

[51] Kovaleva O, Romanov A, Rogers A, Rumshisky A. Revealing the dark secrets of BERT. arXiv preprint arXiv:190808593. 2019. pages 14

[52] Clark K, Khandelwal U, Levy O, Manning CD. What does bert look at? an analysis of bert's attention. arXiv preprint arXiv:190604341. 2019. pages 14

[53] Benesty J, Chen J, Huang Y, Cohen I. Pearson correlation coefficient. In: Noise reduction in speech processing. Springer; 2009. p. 1-4. pages 14

[54] Jaccard P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat. 1901;37:547-79. pages 14

[55] Willett P, Barnard JM, Downs GM. Chemical similarity searching. Journal of chemical information and computer sciences. 1998;38(6):983-96. pages 15

[56] Ralaivola L, Swamidass SJ, Saigo H, Baldi P. Graph kernels for chemical informatics. Neural networks. 2005;18(8):1093-110. pages 15

[57] Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A. A kernel two-sample test. The journal of machine learning research. 2012;13(1):723-73. pages 15

[58] Masters D, Dean J, Klaser K, Li Z, Maddrell-Mander S, Sanders A, et al. Gps++: An optimised hybrid mpnn/transformer for molecular property prediction. arXiv preprint arXiv:221202229. 2022. pages 16, 19, 20, 22, 24, 25, 26, 30, 31

[59] Apodaca RL. Hydrogen Suppression in Cheminformatics; 2025. Accessed: 2025-05-31. `https://depth-first.com/articles/2020/05/18/hydrogen-suppression-in-cheminformatics/`. pages 17

[60] Wojtuch A, Danel T, Podlewska S, Maziarka Ł. Extended study on atomic featurization in graph neural networks for molecular property prediction. Journal of Cheminformatics. 2023;15(1):81. pages 17

[61] Dwivedi VP, Bresson X. A generalization of transformer networks to graphs. arXiv preprint arXiv:201209699. 2020. pages 18, 19

[62] Dwivedi VP, Joshi CK, Luu AT, Laurent T, Bengio Y, Bresson X. Benchmarking graph neural networks. Journal of Machine Learning Research. 2023;24(43):1-48. pages 18

[63] Brüel-Gabrielsson R, Yurochkin M, Solomon J. Rewiring with positional encodings for graph neural networks. arXiv preprint arXiv:220112674. 2022. pages 18, 19

[64] Kreuzer D, Beaini D, Hamilton W, Létourneau V, Tossou P. Rethinking graph transformers with spectral attention. Advances in Neural Information Processing Systems. 2021;34:21618-29. pages 18, 19

[65] Dwivedi VP, Luu AT, Laurent T, Bengio Y, Bresson X. Graph neural networks with learnable structural and positional representations. arXiv preprint arXiv:211007875. 2021. pages 19, 20, 22, 23

[66] Schaub MT, Benson AR, Horn P, Lippner G, Jadbabaie A. Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. SIAM Review. 2020;62(2):353-91. pages 20

[67] Ziegler C, Skardal PS, Dutta H, Taylor D. Balanced Hodge Laplacians optimize consensus dynamics over simplicial complexes. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2022;32(2). pages 20

[68] Horak D, Jost J. Spectra of combinatorial Laplace operators on simplicial complexes. Advances in Mathematics. 2013;244:303-36. pages 20

[69] Topping J, Di Giovanni F, Chamberlain BP, Dong X, Bronstein MM. Understanding oversquashing and bottlenecks on graphs via curvature. arXiv preprint arXiv:211114522. 2021. pages 21

[70] Ying C, Cai T, Luo S, Zheng S, Ke G, He D, et al. Do transformers really perform badly for graph representation? Advances in neural information processing systems. 2021;34:28877-88. pages 22, 30, 31

[71] Rampášek L, Galkin M, Dwivedi VP, Luu AT, Wolf G, Beaini D. Recipe for a general, powerful, scalable graph transformer. Advances in Neural Information Processing Systems. 2022;35:14501-15. pages 23, 25, 26

[72] Ba JL, Kiros JR, Hinton GE. Layer normalization. arXiv preprint arXiv:160706450. 2016. pages 24

[73] Wu Z, Jain P, Wright M, Mirhoseini A, Gonzalez JE, Stoica I. Representing long-range context for graph neural networks with global attention. Advances in neural information processing systems. 2021;34:13266-79. pages 25

[74] Hendrycks D, Gimpel K. Gaussian error linear units (gelus). arXiv preprint arXiv:160608415. 2016. pages 26

[75] Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:180601261. 2018. pages 27

[76] Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:12070580. 2012. pages 29

[77] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014;15(1):1929-58. pages 29

[78] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems. 2017;30. pages 30

[79] Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? arXiv preprint arXiv:181000826. 2018. pages 31

[80] Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2980-8. pages 35, 49

[81] Ridnik T, Ben-Baruch E, Zamir N, Noy A, Friedman I, Protter M, et al. Asymmetric loss for multi-label classification. In: Proceedings of the IEEE/CVF international conference on computer vision; 2021. p. 82-91. pages 35, 48

[82] Wen Y, Zhang K, Li Z, Qiao Y. A discriminative feature learning approach for deep face recognition. In: European conference on computer vision. Springer; 2016. p. 499-515. pages 36, 37, 49

[83] Ruff L, Vandermeulen R, Goernitz N, Deecke L, Siddiqui SA, Binder A, et al. Deep one-class classification. In: International conference on machine learning. PMLR; 2018. p. 4393-402. pages 37

[84] Micikevicius P, Narang S, Alben J, Diamos G, Elsen E, Garcia D, et al. Mixed precision training. arXiv preprint arXiv:171003740. 2017. pages 39

[85] RCS I. GPU Jobs; 2025. Accessed: 2025-08-25. `https://icl-rcs-user-guide.readthedocs.io/en/latest/hpc/queues/gpu-jobs/`. pages 39

[86] nvidia. Nvidia A30 Tensor Core GPU; 2022. Accessed: 2025-08-25. `https://www.nvidia.com/content/dam/en-zz/Solutions/data-center/products/a30-gpu/pdf/a30-datasheet.pdf`. pages 39

[87] scikit learn. average_precision_score; 2025. Accessed: 2025-08-25. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html. pages 40

[88] Flach P, Kull M. Precision-recall-gain curves: PR analysis done right. Advances in neural information processing systems. 2015;28. pages 40

[89] Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd international conference on Machine learning; 2006. p. 233-40. pages 40

[90] Gundelach L, Fox T, Tautermann CS, Skylaris CK. BRD4: quantum mechanical protein–ligand binding free energies using the full-protein DFT-based QM-PBSA method. Physical Chemistry Chemical Physics. 2022;24(41):25240-9. pages 51, 52

[91] Tahir A, Alharthy RD, Naseem S, Mahmood N, Ahmed M, Shahzad K, et al. Investigations of structural requirements for BRD4 inhibitors through ligand-and structure-based 3D QSAR approaches. Molecules. 2018;23(7):1527. pages 51, 52

[92] Fasano M, Curry S, Terreno E, Galliano M, Fanali G, Narciso P, et al. The extraordinary ligand binding properties of human serum albumin. IUBMB life. 2005;57(12):787-96. pages 51, 52

[93] Shen HC, Hammock BD. Discovery of inhibitors of soluble epoxide hydrolase: a target with multiple potential therapeutic indications. Journal of medicinal chemistry. 2012;55(5):1789-808. pages 52

[94] Blevins AD. Closing Thoughts and Future Directions; 2025. Accessed: 2025-09-09. https://www.kaggle.com/competitions/leash-BELKA/discussion/518936. pages 53

[95] Truchon JF, Bayly CI. Evaluating virtual screening methods: good and bad metrics for the “early recognition” problem. Journal of chemical information and modeling. 2007;47(2):488-508. pages 54

[96] Wikipedia. Brier score; 2025. Accessed: 2025-09-08. https://en.wikipedia.org/wiki/Brier_score. pages 54

[97] Zhao L, Akoglu L. Pairnorm: Tackling oversmoothing in gnns. arXiv preprint arXiv:190912223. 2019. pages 55

[98] Cai T, Luo S, Xu K, He D, Liu Ty, Wang L. Graphnorm: A principled approach to accelerating graph neural network training. In: International Conference on Machine Learning. PMLR; 2021. p. 1204-15. pages 55

Declaration

Use of Generative AI

I acknowledge the use of ChatGPT-5, o3, o4-mini and 4o (OpenAI, https://chatgpt.com) and Gemini 2.5 Pro (Google, https://gemini.google.com) as generative AI tools to assist with drafting text, summarising technical concepts, refining explanations, and suggesting code snippets during the development of my research. The tool was used to improve clarity, generate outlines, and provide alternative formulations of my own ideas, but all intellectual contributions, analysis, and final written work are my own. I confirm that no AI-generated content has been presented as original research results, and I take full responsibility for the accuracy and integrity of the submitted work.

Ethical Considerations

This project develops machine learning models for protein-ligand binding affinity prediction using the publicly available BELKA dataset from the Kaggle challenge and Polaris. The dataset consists of binding affinity between proteins and ligands and molecular features of ligands, and does not include identifiable human subject data, clinical samples, or genetic sequence information. Consequently, the research does not require approval from a medical ethics board.

All analyses are performed on open, pre-processed molecular data at the structural and chemical representation level. Ligands are represented through derived structural features rather than genetic material, ensuring compliance with data protection standards such as GDPR.

The purpose of this research is methodological: to investigate machine learning approaches for binding affinity prediction. Model outputs are predictive scores that may inform drug discovery but cannot substitute for experimental validation or clinical evaluation. The work is conducted exclusively for therapeutic and scientific advancement, with awareness of the potential dual-use nature of molecular design technologies.

Experiments were run on institutional high-performance computing resources with attention to efficiency and responsible resource allocation. Overall, the project poses no ethical risks relating to human participants, genetic privacy, or clinical interventions.

Sustainability

In conducting this research, I took deliberate steps to ensure that the work was carried out in an environmentally responsible and energy-efficient manner. All experiments were run on Imperial's high-performance computing (HPC) clusters, the Imperial Department of Computing GPU Cluster and Lab workstation. These facilities are optimised for parallelised workloads and cooling efficiency, making them significantly more energy-efficient than local computation.

I used CPUs rather than GPUs for feature extraction and preprocessing since many cheminformatics tasks parallelise more efficiently with multiprocessing. This choice reduced idle GPU time and ensured that each type of hardware was used for tasks where it is most energy-efficient. Preprocessing was performed once, stored in compressed shards, and reused, avoiding repeated heavy computation.

At the software level, I adopted strategies to minimise wasted computation. Mixed-precision training in PyTorch reduced floating-point requirements and memory footprint. Early stopping, gradient clipping, and learning-rate scheduling prevented unstable or unnecessarily long runs. Code was always tested on small subsets before scaling to full datasets. In HPC and GPU Cluster, I used job arrays with carefully specified resource requests, checkpointing to resume interrupted jobs, and optimised data loading to minimise I/O overhead.

Beyond efficiency, I emphasised reproducibility, indirectly reducing computational waste by avoiding duplicating failed or ambiguous experiments. This included maintaining structured configuration files, clear metrics logging (via Weights & Biases), and saving intermediate checkpoints. These practices ensure that experiments can be repeated reliably by me or others without rerunning extensive exploratory jobs.

Together, these choices minimised the environmental impact of training while ensuring that the research was conducted responsibly, efficiently, and reproducibly.

Availability of Data and Materials

The dataset we used is the BELKA dataset [47], which is publicly available at [46].