Ca' Foscari
University
of Venice

Master Degree
in Computer Science and Information Technology

Final Thesis

Analysis of metabolic network representations through graph neural networks

Supervisor
Dr. Marta Simeoni

Co-supervisor
Dr. Alessandro Bicciato

Graduand
Andrea Moruzzi
Matriculation Number 890550

Academic Year
2025 / 2026

To my grandfather,
who sparked my passion to question,
dissect, and understand the world.

Abstract

The computational modeling of metabolism requires specific representation techniques to capture the inherent complexity of biochemical networks. This thesis investigates the application of Graph Learning (GL) methodologies to classify and analyze metabolic networks derived from the KEGG database. The research focuses on three distinct levels of topological abstraction: Reaction Graphs (RGs), which model individual biochemical reactions as nodes; Metabolic Directed Acyclic Graphs (mDAGs), obtained by condensing strongly connected components (SCCs) into Metabolic Building Blocks (MBBs) to eliminate cycles; and Abstract Metabolic Networks (AMNs), which provide a high-level modular view of pathways.

Methodologically, we adapt and compare two graph learning paradigms from the recent literature: the Graph Kernel Neural Network (GKNN), combining stochastic masks with graph kernels, and the Kolmogorov-Arnold Graph Neural Network (KA-GNN), which uses learnable univariate functions on the edges. Within the kernel pipeline, we introduce two original contributions: a cycle-aware Weisfeiler-Lehman variant (En-WL), designed to neutralize the redundancy of reversible reactions, and a family of node relabeling strategies (Just Reactions, JR; Separate Enzymes, SE; Enzymes and Reactions, ER).

Empirical results demonstrate that both frameworks achieve high classification accuracy, with the KA-GNN outperforming the discrete model on highly heterogeneous clades like Protists. When stress-tested against severe class imbalances and topological noise across the complete KEGG dataset, the KA-GNN shows exceptional generalization robustness. Finally, a structural decoding of the learned masks reveals that the models autonomously reconstruct cohesive biological pathways rather than exploiting spurious correlations.

Contents

List of Figures

List of Tables

List of Acronyms

AMN	Abstract Metabolic Network
BiGG	Biochemical Genetic and Genomic
BRENDA	Braunschweig Enzyme Database
DAG	Directed Acyclic Graph
DD	Decomposition DAG
DGCNN	Deep Graph Convolutional Neural Network
EC	Enzyme Commission number
En-WL	Enhanced Weisfeiler-Lehman
En-WL-ER	Enhanced Weisfeiler-Lehman Enzymes-Reaction
ER	Enzymes and Reactions
GKC	Graph Kernel Convolution
GKNN	Graph Kernel Neural Network
GL	Graph Learning
GNN	Graph Neural Network
HGP	Human Genome Project
IUBMB	International Union of Biochemistry and Molecular Biology
JR	Just Reactions
KA-GNN	Kolmogorov-Arnold Graph Neural Network
KAN	Kolmogorov-Arnold Network
KEGG	Kyoto Encyclopedia of Genes and Genomes
KO	KEGG Orthology
MBB	Metabolic Building Block
mDAG	Metabolic Directed Acyclic Graph

MLP	Multi-Layer Perceptron
NN	Neural Network
ODD	Ordered Decompositional DAG
ODD-ER	Ordered Decompositional DAG Enzymes-Reaction
ODE	Ordinary differential equation
PGDB	Pathway/Genome Database
PPI	Protein-Protein Interaction
RG	Reaction Graph
SBML	Systems Biology Markup Language
SCC	Strongly Connected Component
SE	Separate Enzymes
TCA	Tricarboxylic Acid Cycle
WL	Weisfeiler-Lehman
WL-ER	Weisfeiler-Lehman Enzymes-Reaction
WL-JR	Weisfeiler-Lehman Just-Reaction
WL-SE	Weisfeiler-Lehman Separate-Enzymes

Chapter 1

Introduction

Every living cell sustains itself through metabolism, the vast collection of chemical reactions that convert matter and energy, build cellular components and allow organisms to grow, reproduce and respond to their environment. These reactions do not occur in isolation: the product of one transformation typically becomes the substrate for the next, so that thousands of enzyme-catalyzed steps organize themselves into the interconnected sequences known as metabolic pathways [1, 2]. While classical biochemistry historically investigated these routes as discrete, isolated units, systems biology has shifted the focus toward a holistic paradigm. Under this view, cellular metabolism is studied as a unified, interconnected system, and its global behavior is analyzed in terms of interactions between components, rather than the properties of each individual part of the system [3]. While several mathematical frameworks can model this complexity, abstracting the cell's chemistry into a graph representation offers an approach to formalize and analyze the overall structure of these metabolic networks. This network-driven framework relies on KEGG [4, 5]. Among the available metabolic repositories (such as BioCyc, MetaCyc and Braunschweig Enzyme Database (BRENDA) [6]), KEGG was selected due to its comprehensive standard annotations of pathways, reactions, and enzymes across thousands of sequenced organisms.

Beyond describing a single organism, the network view also opens a comparative dimension. Since metabolic networks are shaped by evolution, their structure reflects an organism's phylogenetic history. Close evolutionary relatives typically share similar metabolic pathways, suggesting that taxonomic identity can be inferred from network topology. However, translating raw biochemical data into a formal graph representation requires specific modeling choices, as the same network can be abstracted at different levels of complexity.

In this work, we consider three representations organized along these different levels of abstraction. At the most granular end, Reaction Graphs (RGs) provide a detailed view of the network, modeling individual biochemical reactions as nodes connected through shared metabolites, thereby preserving the organism's full enzymatic repertoire and its topological cycles. At an intermediate level sit Metabolic Directed Acyclic Graphs (mDAGs), obtained by condensing the strongly connected components of an RG into Metabolic Building Blocks (MBBs) to remove cycles and provide a directional ordering [7, 8]. Finally, at the highest level of abstraction, Abstract Metabolic Networks (AMNs) collapse entire pathways into single nodes, yielding a compact, modular view of pathway connections [9, 10]. These three formalisms are applied to a core dataset of 8904 organisms retrieved from the KEGG repository, generating three distinct graph datasets (one for each representation). This collection is highly unbalanced across the six considered taxonomic groups, as it is heavily dominated by Bacteria, which account for 7615 organisms, followed by Animals (535), Archaea (405), Fungi (154), Plants

(139) and Protists (56). This structural imbalance motivated the creation of smaller, balanced subsets (Sub_300 and Sub_600) for training and validation, while the full, unbalanced dataset is retained to test model scalability.

The choice of formalism determines which structural information survives encoding and, consequently, how much of the original biological signal a learning algorithm can exploit. A central focus of this thesis is whether the compressed nature of the more abstract representations preserves enough information for accurate taxonomic classification, or whether the finer, more detailed reaction-level view proves ultimately more informative.

The relational and irregular nature of these graphs makes them poorly suited to traditional machine-learning methods, which flatten their input into fixed-size vectors and thereby discard the connectivity that defines a metabolite's functional role. Graph Learning (GL) [11] offers a more natural alternative, and within it two broad families coexist. Graph kernels [12] compare networks by decomposing them into simpler substructures and measuring how frequently those substructures recur, producing interpretable and structurally stable similarity measures. The Weisfeiler-Lehman (WL) [13] kernel is the best-known representative of this family. Graph Neural Networks (GNNs) [14], in contrast, learn continuous representations through iterative message passing, trading transparency for greater expressive power. More recent architectures combine the two worlds. The Graph Kernel Neural Network (GKNN) [15] embeds graph kernels into a neural pipeline through learnable structural masks, and has recently been applied to metabolic network classification by [16]. The Kolmogorov-Arnold Graph Neural Network (KA-GNN) [17] replaces the fixed activations of conventional GNNs with learnable univariate functions placed on the network edges. How these two paradigms compare on metabolic data, and how their behavior interacts with the chosen level of graph abstraction, remains an open question that we try to address in this thesis. Furthermore, the standard WL refinement is poorly suited to the dense, label-rich, and frequently reversible nature of biochemical networks. This mismatch necessitates the development of graph kernels and node encodings specifically tailored to the metabolic domain.

Motivated by these gaps, this thesis investigates whether the structure of metabolic networks, encoded at three levels of abstraction, is sufficient to classify organisms into their six taxonomic kingdoms, and how a kernel-based paradigm and a learnable one compare on this task. Classification accuracy is employed to quantitatively evaluate expressiveness: a representation that still supports accurate prediction must have preserved enough of the evolutionary signal embedded in the metabolism. The study extends the framework of [16], which evaluated the Deep Graph Convolutional Neural Network (DGCNN) [18] and GKNN models on the same KEGG-derived representations, and uses it as a direct benchmark.

The thesis is organized as follows. Chapter 2 presents a brief introduction to the biological foundations of metabolism and the three graph-based representations (AMNs, mDAGs and RGs) compared throughout the work. Chapter 3 formalizes the graph-learning framework and the GKNN and KA-GNN architectures, and introduces Enhanced Weisfeiler-Lehman (En-WL), a cycle-aware variant of the WL kernel that neutralizes the redundancy induced by reversible reactions. Chapter 4 describes the KEGG-derived case study together with a family of node relabeling strategies (Just Reactions (JR), Separate Enzymes (SE) and Enzymes and Reactions (ER)) that encode reaction and enzymes information compactly. Chapter 5 reports the classification results, the head-to-head comparison between the GKNN and the newly introduced KA-GNN, benchmarked against the DGCNN and GKNN baselines of [16], together with a robustness and mask-interpretability analysis on the full, imbalanced dataset. Finally, Chapter 6 summarizes the findings and outlines directions for future work.

Chapter 2

Background on Metabolism

Metabolism represents the totality of chemical reactions occurring within a living organism to maintain life, allow for growth and reproduction, and respond to environmental stimuli. These reactions are organized into intricate, interconnected pathways that transform energy and matter with extraordinary precision [3].

In this chapter, we explore the foundations of metabolic modeling. We first introduce the fundamental principles of metabolism from a scientific point of view, and then examine the standard representations used to map biochemical reactions into graph-theoretical frameworks, ranging from highly detailed reaction-centric models to abstract functional networks. Finally, we discuss the inherent challenges of modeling such data, including noise and size, and justify the necessity of GL as the most suitable paradigm for capturing the relational and evolutionary signals embedded within metabolic data.

2.1 Introduction to Metabolism

At its most fundamental level, the "chemistry" of life is defined by the unique structural versatility of carbon-based molecules. This chemical flexibility allows for the creation of a vast diversity of organic compounds, ranging from simple sugars to complex polymers, which serve as the essential building blocks for energy storage, structural integrity, and information transfer [2].

Metabolism is the complex biochemical system that orchestrates these molecular resources, representing the complete set of life-sustaining chemical reactions that occur within the cells of living organisms. The main functions of metabolism include the conversion of food to energy, the construction of cellular components, and the elimination of waste products. Metabolism is the key to managing growth, reproduction, cellular maintenance and response to environmental changes within all cellular organisms [2].

Rather than a collection of independent events, metabolism functions as a highly coordinated and dynamic system where the product of one reaction becomes the substrate for the next, creating a continuous sequence of biochemical transformations. These contiguous flows of reactions are called "metabolic pathways" and each step of this path brings a specific chemical transformation to the cells [1]. These enzyme-catalyzed reactions are traditionally categorized into two interconnected metabolic pillars: *catabolism* (release of energy), the breakdown of organic matter to harvest energy and reducing power, and *anabolism* (consumption of energy), the building up of complex cellular components such as proteins and nucleic acids, also called biosynthesis.

Catabolism is the degradative phase, in which organic nutrients are converted into simpler products, releasing energy in the process. In contrast, anabolism is the biosynthetic phase, where

simple precursors are assembled into more complex molecules including lipids, polysaccharides, proteins and nucleic acids [2].

Central to this process are metabolites, the small molecules that are the intermediates and products of metabolic activity. The transformation of these molecules is governed by enzymes, specialised biological catalysts that ensure reaction specificity. A classic example is the enzyme *hexokinase*, which specifically targets glucose to initiate its breakdown; in a computational context, such enzymes act as functional operators that transform one chemical state into another [1].

In this vast and intricate network of reactions, the flow of matter and energy is tightly regulated through homeostasis, the dynamic self-regulatory process by which biological systems maintain a stable internal state amidst constant environmental fluctuations. It can be seen as a kinetic steady-state where the rate of metabolite production is precisely balanced by the rate of its consumption [19].

The inherent complexity of these interconnected pathways suggests that metabolism is far more than a simple sequence of reactions: it is a massive, integrated system where the output of one process is inextricably linked to the input of dozens of others. To capture this complexity, systems biology translates these biochemical realities into formal mathematical structures known as *Metabolic Networks*. In this framework, the cell's chemistry is abstracted into diverse graph-like structures where the nodes and edges represent metabolites and their biochemical transformations according to the specific perspective chosen.

This definition of metabolism as a network allows us to move beyond the traditional study of isolated pathways to analyze the global topology and properties of the entire system [3]. This mathematical abstraction is not merely a visualization tool, but a necessary prerequisite for applying computational methods (such as Graph Learning) to predict how the system responds to genetic or environmental perturbations. However, it is crucial to acknowledge that such graph-based abstraction are static and structural. By focusing exclusively on the topological blueprint, this representation abstracts away the real-time dynamics of metabolic networks, omitting critical biochemical parameters such as reaction kinetics, enzyme saturation states, and thermodynamic constraints. To construct such a representation, high-quality and structured data is required, leading to the necessity of comprehensive biological databases like KEGG [4, 5].

2.2 Biological Pathways and Systems Biology

The traditional study of metabolism has long relied on the concept of isolated biological pathways, each representing a discrete sequence of reactions that perform specific cellular functions. However, the need to understand biology at the system level (i.e., the emergence of "Systems Biology") has shifted the paradigm toward a holistic view, where metabolism is treated as a single, integrated network. This discipline emphasizes that the behavior of the metabolic network is governed by the topology of its interactions rather than the sum of its individual components [20]. The key pillars of this systemic perspective, along with the inherent abstractions required by a graph-theoretical representation, are:

- *Contextual Functionality*: A metabolite's biological significance is defined by its connectivity within the global network. A molecule such as pyruvate is not merely a chemical structure but a critical junction (or "metabolic hub") that bridges multiple pathways, including Glycolysis and the Tricarboxylic Acid Cycle (TCA) cycle [3].

- *Structural Robustness*: In a living cell, metabolic networks can reroute biochemical fluxes if a specific enzyme is inhibited or a nutrient is scarce, ensuring homeostasis under

environmental stress [21]. However, within a static graph representation, this robustness is translated exclusively into **topological redundancy**. A graph-theoretical model lacks physical quantities and kinetic rates; therefore, it cannot simulate active flux rerouting, but it successfully maps the existence of alternative paths and structural bypasses that make such biological resilience structurally possible.

- *Predictive Logic*: A system-level view allows researchers to conceptualize how perturbations, such as genetic mutations or drug introductions, propagate through the metabolic system to affect distant reactions [20]. Nevertheless, it must be noted that while dynamic or stoichiometric models simulate quantitative cascade effects, a purely topological graph can only predict structural reachability and connectivity changes. The predictive logic of a structural graph is thus limited to identifying which pathways are topologically disconnected or compromised when specific nodes are removed, rather than quantifying the active metabolic rate variations.

By adopting this holistic perspective of metabolism, it becomes possible to analyze the systemic properties of the cell, providing a fertile ground for the application of computational models. However, the efficacy of such models is strictly dependent on the construction of a comprehensive and accurate structural representation. Consequently, the essential prerequisite for any system-level analysis is the acquisition of high-quality metabolic data, starting from biological databases that catalog these molecular interactions.

2.3 Metabolic Knowledge Bases

The construction of metabolic networks requires curated biological databases that aggregate decades of biochemical research. Several specialized repositories exist, each offering a different perspective on metabolic data. Some of the most widely used include:

- **KEGG** [6]: A prominent bioinformatics resource used to link genomes to biological functions and environments. It serves as a comprehensive catalog of pathways, genes, proteins, and reactions. The organization of this resource is detailed in Section 2.3.1.
- **BioCyc** [6]: A collection of over 160 Pathway/Genome Databases (PGDBs). It organizes information on genomes and cellular networks to facilitate computational analysis and editing via software tools.
- **MetaCyc** [6]: A highly curated, non-redundant reference database containing experimentally elucidated metabolic pathways and enzymes from thousands of different organisms.
- **ENZYME** [6]: A nomenclature-focused repository based on the recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (IUBMB). It provides specific data on enzyme functions, reactions, and inhibitors.
- **BRENDA** [6]: The largest publicly available enzyme information system worldwide. It features manually curated data on functional parameters, kinetics, stability, and molecular properties extracted from primary literature.
- **Biochemical Genetic and Genomic (BiGG)** [6]: A knowledge base of genome-scale metabolic reconstructions. It uses standard nomenclature to allow for comparison across various organisms and supports export to the Systems Biology Markup Language (SBML) format.

- **Reactome** [6]: An open-source, manually curated database focused specifically on human biological processes and pathways. It systematically associates human proteins with molecular and cellular functions.
- **KaPPA-View4** [6]: A specialized metabolic pathway database that allows users to visualize and overlay omics data (such as gene-to-gene correlations) directly onto metabolic pathway maps, with a strong focus on plant species.

2.3.1 KEGG Database

The core of the data integration process in this research is the KEGG database [4]. Started in 1995, it has evolved into a comprehensive resource that organizes biological information into a hierarchical structure of databases. The project was initiated by Professor Minoru Kanehisa at Kyoto University and has become a cornerstone in the field of bioinformatics. Under the ongoing Japanese Human Genome Project (HGP), the KEGG database has been developed as a computational model of biological information systems, representing data in terms of molecular interaction and reaction networks. The model is manually curated and continuously updated, making it a reliable source for metabolic data.

KEGG is organized as a multi-layered suite of databases designed to bridge genomic information with high-level systemic functions. In its current version, it consists of several databases categorized into four main information blocks [4, 5]:

- **Systems Information**: Contains the PATHWAY database, which provides graphical representations of cellular processes such as metabolism, signal transduction, and the cell cycle. It also includes BRITE for functional hierarchies and MODULE for conserved units of gene sets.
- **Genomic Information**: Centered around the GENES database, which catalogs genes for completely sequenced genomes. This category also includes the KEGG Orthology (KO) system, which serves as a "pivot" for linking genomic data to functional pathways.
- **Chemical Information**: Includes the LIGAND suite of databases, which stores data on chemical compounds (COMPOUND database), enzymatic reactions (REACTION database) and enzyme molecules (ENZYME database).
- **Health Information**: A recent expansion containing databases like DISEASE, DRUG and NETWORK, which link molecular networks to human health and clinical data.

To provide a comprehensive view of the KEGG architecture, it is essential to examine how these categories interact to form a unified biological knowledge base. The database is built upon a fundamental philosophy of linking "blueprints" to "entities" through a multi-tier hierarchy that facilitates the transition from molecular data to systemic understanding. A critical aspect of KEGG's methodology is the KO system. Each entry in a molecular network is assigned a KO identifier (K-numbers) that represents a functional ortholog (genes or proteins that perform the same function across different species). This allows for the computational expansion of "reference" pathways into "organism-specific" ones. This integration across these layers is made possible by a rigorous system of cross-references. For instance, a reaction (R-numbers) is linked to the compounds it transforms (C-numbers) and the functional orthologs that catalyze it (K-numbers), which are in turn mapped to specific genes in a given organism.

KEGG has significantly expanded its scope, now containing over 8400 genomes. Recent updates emphasize taxonomy-based analysis, allowing researchers to examine how functional

links in pathways and physical links on chromosomes are conserved across different taxonomic groups. Key new tools include, for example, the KEGG Genome Browser, for analyzing the physical locations of genes on the chromosome; the Brite Hierarchy Viewer for Taxonomy, enabling global views of organism groups; and the Viral Datasets, containing viral genes and functionally important viral peptides [22].

The adoption of the KEGG database as the primary information source for this research is justified by its unparalleled status as a curated repository of biochemical pathways. Its rigorous annotation of enzymes and reactions provides the foundational data necessary for the structural modeling of metabolism. Using its standardized identifiers, we ensure that our metabolic reconstructions are both reproducible and biologically grounded. Furthermore, this choice facilitates a direct methodological alignment with the work of [16], where this comparison was first introduced. By maintaining consistency in the data source, we can effectively benchmark the performance of our models against established findings in the field. The granular nature of KEGG data is instrumental in the transition from raw biochemical lists to the three graph-based representations of metabolism (Reaction Graphs (RGs), Metabolic Directed Acyclic Graphs (mDAGs) and Abstract Metabolic Networks (AMNs)) described in the next sections. This structured approach allows for a systematic evaluation of how different levels of topological abstraction impact the efficiency of GNNs in metabolic classification tasks.

2.4 Metabolic Networks Representations

The structural information curated within the KEGG database provides a comprehensive map of the biochemical interactions that define life. However, to leverage this information within a computational or machine learning framework, these biological relationships must be translated into formal mathematical representations. Depending on the specific analytical objective, various modeling paradigms can be adopted. For instance, Ordinary differential equations (ODEs) are extensively used to model deterministic reaction kinetics, Petri nets are employed to analyze concurrent structural properties and reachability, and Bayesian networks offer a probabilistic framework to capture conditional dependencies and uncertainties within metabolic pathways. Nevertheless, when the primary goal shifts toward representation learning and structural pattern recognition across full-scale networks, graph-based representations emerge as a suitable paradigm. Translating metabolic relationships into formal graphs allows us to preserve the underlying topology while leveraging modern geometric deep learning and graph kernel methods for taxonomic classification. This translation process is not trivial, as it involves a trade-off between biological granularity and computational complexity. Depending on the level of detail retained, the resulting network can emphasize different properties of the metabolism, from the stoichiometry of individual reactions to the high-level modularity of functional pathways. This section explores three distinct graph-based formalisms used to model metabolic networks: AMNs, mDAGs, and RGs. Each representation offers a unique perspective on metabolic systems, capturing different levels of structural and functional complexity. These models form a hierarchical spectrum of abstraction: while AMNs provide a high-level overview focused on the topological organization of functional pathways, RGs offer a granular view by incorporating the complete set of biochemical reactions. Between these extremes, mDAGs serve as an intermediate representation, balancing biological detail with an acyclic, oriented structural organization.

2.4.1 Basic Graph Concept

In the context of this research, a metabolic network is translated into a graph where biological entities and their interactions are mapped to mathematical abstractions. Formally, a graph G is defined as an ordered pair:

$$G = (V, E) \tag{2.1}$$

where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of vertices (or nodes) representing biological units, and E is a set of edges representing the relationships, transformations, or associations between those units [23]. In this general formulation, the nature of the edges in E determines whether the network structural architecture is directed, undirected, or mixed.

A fundamental concept in graph theory is the notion of adjacency. Two vertices $u, v \in V$ are said to be *neighbors* (or adjacent) if they are connected by an edge within the set E. This neighborhood relationship underpins the fundamental logic of structural network analysis and message passing algorithms. Depending on the specific formalization of the edge set E, the concept of neighborhood adapts as follows.

An **undirected graph** is a graph $G = (V, E)$ in which the edges have no orientation. Formally, each edge $e \in E$ is an unordered pair of vertices, denoted as $e = \{u, v\}$. In this context, the neighborhood relation is inherently symmetric: if u is a neighbor of v, then v is strictly a neighbor of u. In biological modeling, undirected graphs are typically employed to represent reciprocal physical associations, such as Protein-Protein Interaction (PPI) networks.

A **directed graph** (or digraph) is a graph $G = (V, E)$ where each edge possesses a specific orientation. Formally, each edge $e \in E$ is defined as an ordered pair of vertices, denoted as $e = (u, v)$, indicating a directional link that points from the source node u to the target node v. In a directed graph, the neighborhood concept is asymmetric and branches into two distinct sets: the set of incoming neighbors (predecessors) and the set of outgoing neighbors (successors). Since biochemical reactions are inherently directional, transforming substrates into products, the use of directed edges allows for the authentic mathematical representation of metabolic flux directionality.

To capture the complex, multi-component architectures of biological pathways, more advanced graph topologies are required. A **bipartite graph** is a special type of graph where the set of vertices V is partitioned into two disjoint and independent sets, U and W, such that every edge $(u, w) \in E$ connects a vertex in U to one in W. It is important to understand that no edges exist between vertices within the same set.
This representation is particularly relevant for the KEGG database structure. Indeed, a metabolic network can be naturally modeled as a bipartite graph:

- C-numbers: first set of nodes represents Metabolites
- R-numbers: second set represents Reactions

An edge exists only between a metabolite and a reaction, indicating the metabolite is either a substrate or a product of that specific transformation.

Finally, an **hypergraph** is a generalization of a graph in which an edge can join any number of vertices. Formally, an hypergraph is an ordered pair $H = (V, E)$, where V is a finite set of vertices and E is a set of non-empty subsets of V, called hyperedges ($e \in E$). In a directed hypergraph, each hyperedge is defined as an ordered pair $e = (T, H)$, where $T \subseteq V$ represents the tail (source nodes) and $H \subseteq V$ represents the head (target nodes).
From a purely biochemical perspective, the directed hypergraph constitutes the mathematically ideal representation for metabolic networks. Traditional and bipartite graphs are fundamentally

limited to binary relations, meaning an edge can only connect a single node to another (or a metabolite to a single reaction). However, biological reality is inherently multi-variant: a single metabolic reaction often involves multiple substrates simultaneously transforming into multiple products (e.g., $A + B \xrightarrow{\text{Enzyme}} C + D$).

By mapping the entire set of substrates to the tail T and the entire set of products to the head H of a single hyperedge, an hypergraph preserves the exact stoichiometry and the collective nature of biochemical transformations without artificially fragmenting them into isolated binary pairs. While the computational complexity and the sparsity of current hypergraph learning algorithms often necessitate a conversion into simpler graph formats, as detailed in the subsequent sections, acknowledging this hypertopological nature is essential to fully grasp the multi-component dependencies embedded within the metabolism of life.

2.4.2 Reaction Graphs (RGs)

The RG is the most granular and detailed representation of metabolic networks employed in this research. While the raw biological data in KEGG is inherently bipartite as described in the previous section, the RG is constructed as a unipartite projection onto the reaction space.
In this representation, metabolism is modeled as a directed graph where [24]:

- **Nodes** (V): Each vertex represents an individual biochemical reaction, uniquely identified by its KEGG R-number.
- **Edges** (E): A directed edge exists from reaction R_i to reaction R_j if they are linked by a shared metabolite flow. Specifically, an edge (R_i, R_j) is established if at least one product of reaction R_i serves as a substrate for reaction R_j.

RGs provide a detailed topological map of the enzymatic reactions of an organism. As an example, the RG representation of the Glycolysis pathway in Homo Sapiens is shown in Figure 2.1. Each node corresponds to a specific enzymatic reaction (KEGG R-number), and edges represent the directed sharing of metabolites between reactions. The figure illustrates the topological complexity and density typical of this granular representation. As shown, while the RG formulation naturally exhibits a higher edge count compared to downstream abstractions like AMNs and mDAGs, it cannot be mathematically classified as a dense network; rather, it remains a sparse graph where connectivity is strictly dictated by shared metabolites.
Furthermore, it is important to note that this representation does not preserve the full biochemical detail of the organism's metabolic repertoire. Because metabolites are abstracted into structural edges to establish direct connections between reaction nodes, their explicit identities, structural properties, and stoichiometric values are omitted. By focusing on the connectivity between enzymes, it highlights the sequential dependencies required to transform nutrients into cellular components [25].
This high level of detail introduces significant topological complexity:

1. *Connectivity and Density*: Since many reactions share common metabolites, RGs tend to be highly connected, resulting in an adjacency matrix that can be computationally expensive to process for certain GNN architectures.
2. *Currency Metabolites*: A common challenge in RG construction is the presence of "currency metabolites" (e.g., H_2O, ATP, etc.). These molecules participate in hundreds of reactions, potentially creating "shortcut" edges that do not represent a meaningful biological pathway [26]. To ensure the quality of the graph, these ubiquitous compounds are

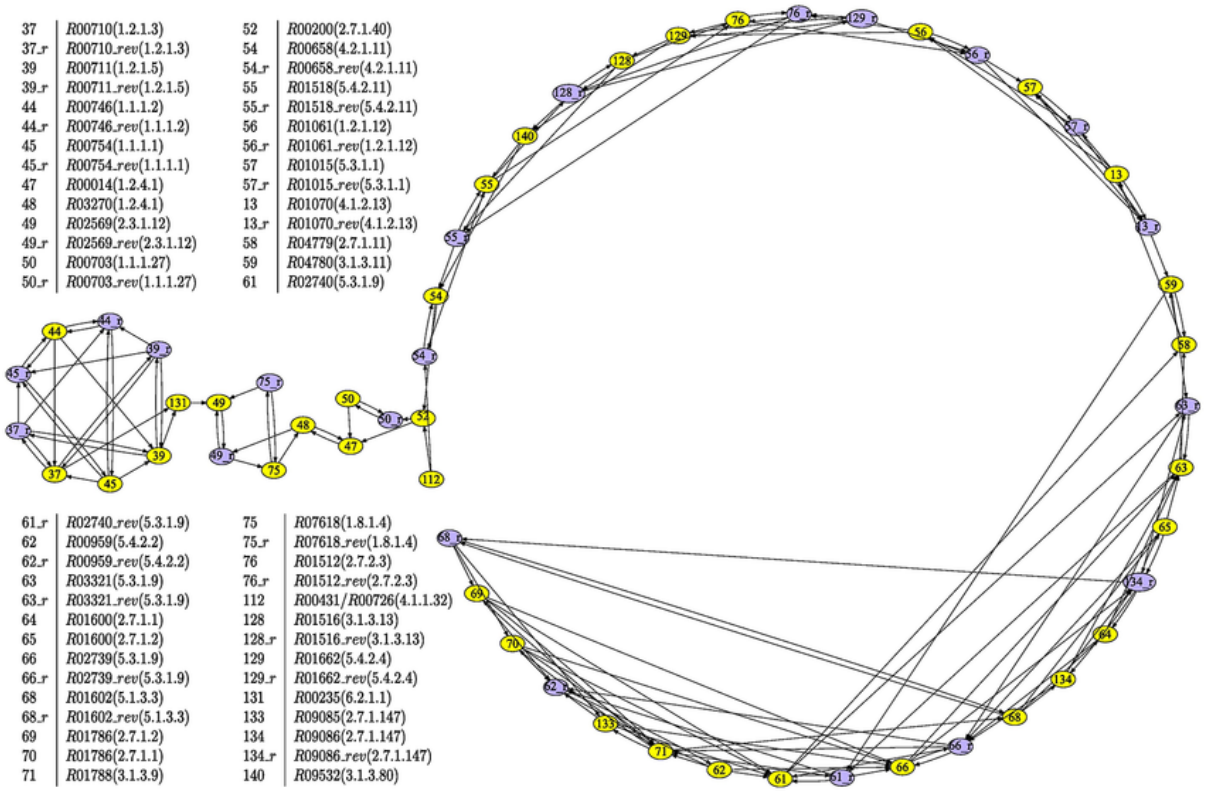


Figure 2.1: The reaction graph corresponding to the metabolic pathway of the glycolysis in *Homo sapiens*. Nodes in this graph are the reactions of the pathway, depicted in blue are the reverse of a reversible reaction and in yellow the reaction itself (taken from [7]).

often filtered out during the projection process to preserve the specificity of the metabolic signal.

3. *Cycles and Feedback Loops*: Unlike the more abstract models, RGs fully retain the cyclic nature of metabolism (e.g., TCA cycle). While biologically accurate, these cycles prevent the graph from having a clear directed structure, making the extraction of long-range dependencies more difficult for simpler algorithms [27].

Despite these challenges, the RG remains the fundamental benchmark for metabolic modeling, as it preserves the maximum amount of information regarding the enzymatic repertoire encoded in an organism's genome.

2.4.3 Metabolic Directed Acyclic Graphs (mDAGs)

mDAGs represent an intermediate level of abstraction, designed to organize metabolic reactions into a hierarchical structure. Unlike RGs, which preserve all cyclic dependencies, mDAG is a Directed Acyclic Graph (DAG), meaning that it contains no directed cycles: for every node $v \in V$, there is no directed path that starts and ends at v.

The construction of an mDAG from KEGG data requires a systematic resolution of metabolic cycles such as, for example, the *TCA cycle* or the *Pentose Phosphate pathway*. The mDAG can be viewed as a condensation of the RG where all the Strongly Connected Components (SCCs) are grouped into single nodes. With SCC we identify a subgraph such that every pair of nodes in it is strongly connected (equivalence relation). These SCCs in the reaction graph R_g are called MBBs, and are the basic structure of the new metabolic DAG [8].

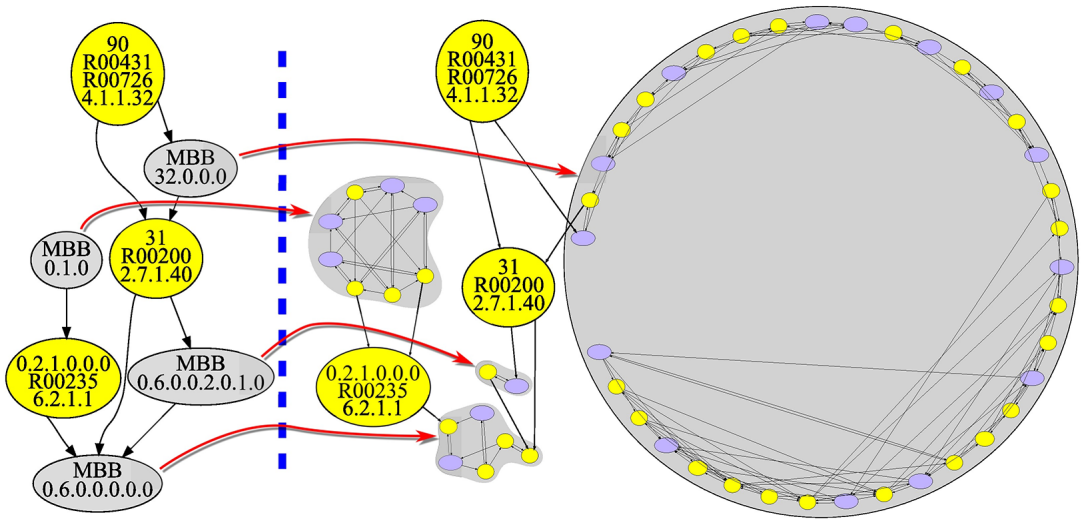


Figure 2.2: Relation between the reaction graph of the glycolysis pathway in *Homo sapiens* and its corresponding mDAG. The mDAG has seven nodes which are the corresponding MBBs in the reaction graph. Notice that three of them are yellow nodes, that is, a MBB with only one reaction, and four of them are grey nodes which are MBBs with more than one reaction (taken from [7]).

Figure 2.2 shows the relation between the reaction graph of the Glycolysis pathway in *Homo sapiens* and its corresponding mDAG. In the second formalization, the mDAG has seven MBBs, three of them with only one reaction (yellow nodes) and four of them with more than one reaction (grey nodes).

The process of transformation is called "condensation" and involves:

- **Cycle Decomposition**: Identifying SCCs within the reaction network and breaking feedback loops. This is often achieved by identifying "key" metabolites or reactions that act as entry and exit points for the cycle.
- **Source-to-Sink Orientation**: Organizing the network into a layered structure where "source" nodes flow towards "sink" nodes.
- **Topological Sorting**: Due to the acyclic property, the nodes in an mDAG can be linearly ordered such that for every directed edge (u, v), u comes before v in the ordering. This property is particularly advantageous for GNNs, as it defines a clear direction for information propagation.

As illustrated in Figure 2.3, the mDAG provides a cleaner, "cascade-like" view of metabolic pathways, potentially reducing the complexity of the network and allowing the graph models to better learn the causal dependencies of biochemical transformations. The network structure has been processed to eliminate directed cycles, establishing a clear order from input substrates to metabolic outputs.

2.4.4 Abstract Metabolic Networks (AMNs)

The AMN representation employs an undirected graph to represent the highest level of topological abstraction in this study. While RGs and mDAGs focus on the connectivity between individual enzymatic reactions, the AMN shifts the focus toward the modular organization of metabolism. The construction of an AMN involves a process of functional coarse-graining [9]:

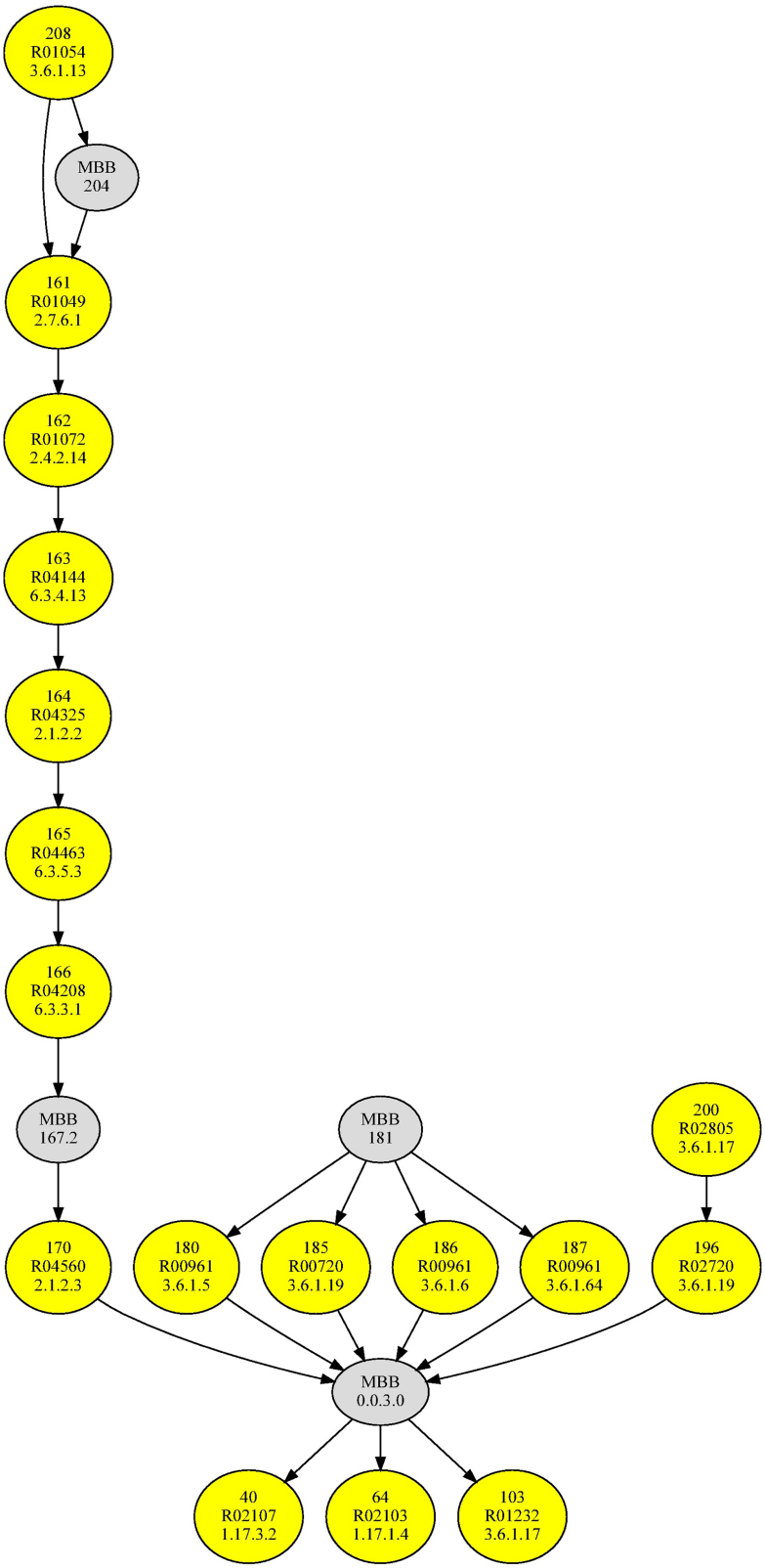


Figure 2.3: The condensed mDAG structure of the Purine metabolism pathway in *Homo sapiens*. Nodes in this topological representation correspond to individual MBBs extracted by contracting the cyclic components of the underlying reaction graph. To highlight network modularity, a dual-color coding scheme is adopted: yellow nodes denote singleton MBBs containing exactly one linear or unidirectional reaction, whereas grey nodes represent macro-nodes encapsulating complex sub-networks or strongly connected cyclic pathways with multiple interconnected reactions.(Adapted from: [7]).

- **Nodes**: Each vertex represents an entire metabolic pathway (as defined in the KEGG database). In this view, an organism's metabolism is reduced to a set of high-level functional units.

- **Edges**: An edge between two pathway nodes is established if they share one or more intermediate metabolites. To ensure that the topology reflects meaningful biological interactions, ubiquitous molecules known as "currency compounds" are systematically excluded from the intersection calculation to prevent the creation of spurious or non-informative edges.

The core value of this representation lies in its expressiveness. Despite the loss of reaction-level detail, the topological organization of AMNs preserves a robust biological and evolutionary signal. The spatial arrangement of pathways effectively discriminates macro-evolutionary events, allowing for a clear distinction between domains of life and across different kingdoms [9].

From a computational perspective, the AMN offers two crucial advantages for GNNs training:

1. **Noise Filtering**: By aggregating reactions into functional units, the AMN acts as a structural filter, emphasizing the global "metabolic fingerprint" of the organism by aggregating local reaction details.

2. **Efficiency**: The reduced size of the graph, as we will see in the experiments, enables a low computational overhead while maintaining the model's ability to identify taxonomic relationships.

The architectural logic of AMN is visually summarized in Figure 2.4. As depicted, the network bypasses individual enzymatic steps to map the interdependencies between core metabolic modules, such as *Glycolysis*, *Pentose Phosphate* and *Histidine Metabolism*. Clearly, AMN capture the "metabolic backbone" of an organism, where edges define the flow of essential compounds between major functional units rather than simple chemical transformations.

2.5 Challenges in Computational Modeling

The transition from raw biochemical data to predictive models involves several hurdles. While the representations described in the previous sections (RGs, mDAGs and AMNs) provide a structural framework, their analysis remains non-trivial due to the inherent nature of biological data.

2.5.1 Data sparsity and noise

One of the primary challenges in modeling metabolic networks is the uneven quality and density of the available data. Databases such as KEGG are gold standards, but they are also subject to several forms of "noise":

- **Annotation Gaps**: Many organisms, especially non-model species, have incomplete metabolic reconstructions. This leads to "gaps" in the network where reactions or enzymes are missing, potentially breaking the connectivity of the graph [28, 29].

- **Currency Compounds**: As we have seen in the previous sections, the presence of ubiquitous metabolites like H_2O or ATP can create a sort of "shortcut" edges that do not

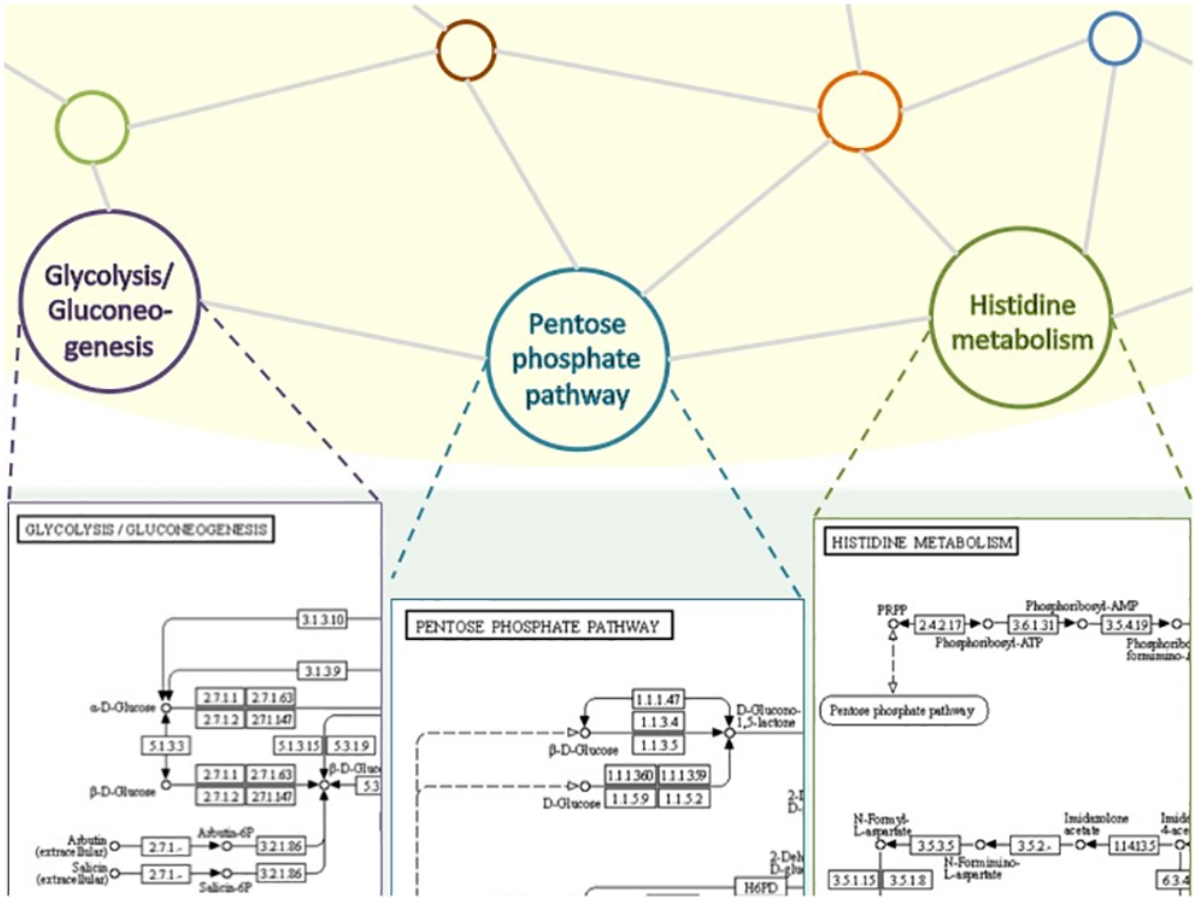


Figure 2.4: A two-level view of the metabolism: the *Structural level*, which corresponds to AMN and the *Functional level*, which highlights the functional role of each pathway through their interaction (Adapted from: [10]).

represent specific functional relationships. If not properly filtered, this noise can lead to a "small-world" effect [26] where every node appears connected to every other node, obscuring the actual metabolic signal.

- **Structural Complexity**: RGs are often extremely dense and high-dimensional. For GNNs, this density can lead to the *oversmoothing* problem, where node representations become indistinguishable after a few layers of computation.

2.5.2 The need for graph learning

The inherent complexity of metabolic structures suggests that traditional machine learning methods (often relying on "flattening" data into vectors) may be insufficient to fully decode the biological information embedded in these systems. Such methods tend to ignore the relational context that defines the actual functional role of cellular components.

In this context, the adoption of GL and, more specifically GNNs, becomes a fundamental requirement. The primary motivation is that metabolic data is naturally organized as a network, so the mathematical framework used to analyze it should reflect this topology.

Unlike standard Neural Networks (NNs), GNNs leverage the "expressiveness" of the graph, leading even abstract models like AMNs to retain deep evolutionary signals [9]. By propagating information through the edges of the network, these models can learn how the position of a reaction relative to its neighbors determines its systemic importance, a feature that is invisible to non-relational models.

Chapter 3

Graph Learning Methods

The goal of this chapter is to formalize the computational framework chosen to classify metabolic networks. The discussion is organized as follows: first, we establish the foundations of learning on graphs, with a particular emphasis on the message passing paradigm and its biological relevance. We then delve into Graph Kernel methods [12], introducing a novel, cycle-aware variation of the WL kernel (En-WL) specifically designed to address the symmetries of reversible metabolic reactions, together with the Ordered Decompositional DAG (ODD) kernel [30] for hierarchical flow analysis. Finally, we present the neural networks employed in this study: the Graph Kernel Neural Network (GKNN) [15] and the Kolmogorov-Arnold Graph Neural Network (KA-GNN) [17].

3.1 Foundations of Representation Learning

The core objective of representation learning is to automatically transform raw input data into a mathematical format (typically a feature map) that preserves the essential information required for a specific task, such as classification or clustering. In the context of metabolic systems, this process is particularly challenging due to the non-linearly interconnected nature of biological networks.

Traditionally, Bioinformatics relied on manual feature engineering, where experts predefined sets of descriptors (e.g., the presence or absence of specific enzymes) to characterize an organism. However, these "flat" representations often fail to capture the structural dependencies and the systemic organization of life. Representation learning overcomes these limitations by shifting the focus from individual components to the relational context in which the metabolic components operate.

By mapping metabolic networks into a latent space, we can compute distances between organisms that reflect their functional and evolutionary divergence. As established in Section 2.5.2, in this study we leverage the concept of **structural expressiveness** [9], which suggests that the arrangement of metabolic information contains a deeper signal than the mere sum of its parts. Effective representation learning on graphs must, therefore, ensure that the chosen embedding or kernel maintains the "topological signature", allowing the model to distinguish between different biological taxonomic groups based on their "metabolic blueprint" rather than isolated biochemical reactions.

3.2 Learning on Graphs

Learning on graph-structured data presents unique challenges compared to Euclidean data (such as images or sequences). Unlike a grid of pixels, a metabolic network is characterized by an irregular topology where the number of neighbors for each node is variable and the global structure lacks a fixed orientation. To effectively learn from these data, each metabolic network representation must be **permutation invariant**, meaning the output should not change regardless of the order in which nodes are presented [15]. The most common approach to achieve this is through the learning of structural representations that capture both the individual properties of the nodes and the global arrangement of the network. This is achieved by exploiting the connectivity of the graph to propagate information across its topology.

3.2.1 Message Passing Paradigm

The Message Passing Paradigm is the fundamental mechanism underlying most modern GNNs and many topological kernel methods, as we will see in the next section. The core idea is that the representation of a node should be influenced not only by its own features but also by the features of its neighborhood.

Formally, for a node v in a graph G, the message-passing process at each iteration k consists of two main steps:

1. **Aggregation**: The node collects "messages" (feature vectors) from all its immediate neighbors $N(v)$. This is typically done using a permutation-invariant function (such as sum, mean or max) to ensure that the order of neighbors does not affect the results:

$$m_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\}\right)$$

2. **Combination**: The aggregated message is then combined with the node's own previous state to produce a new representation:

$$h_v^{(k)} = \text{UPDATE}^{(k)}\left(h_v^{(k-1)}, m_v^{(k)}\right)$$

In the context of metabolic networks, this paradigm allows the model to understand the **functional context** of a reaction or a pathway. For instance, as explained in Figure 3.1, a node representing the "Citrate Cycle" will update its state by aggregating information from its connected neighbors like "Glycolysis" or "Pyruvate metabolism". By repeating this process over multiple iterations, the model can capture increasingly larger structural patterns, eventually encoding the global metabolic "fingerprint" of an organism [9].

3.3 Graph Kernel Methods

While GNNs learn representations through iterative optimization, Graph Kernels offer an alternative paradigm based on predefined similarity functions [12]. A graph kernel is a function $K(G, G')$ that computes the similarity of two graphs represented as vectors of features in a high-dimensional feature space.

In the context of metabolic networks, kernels are particularly valuable due to their interpretability and structural stability, meaning that the kernel function is robust against minor

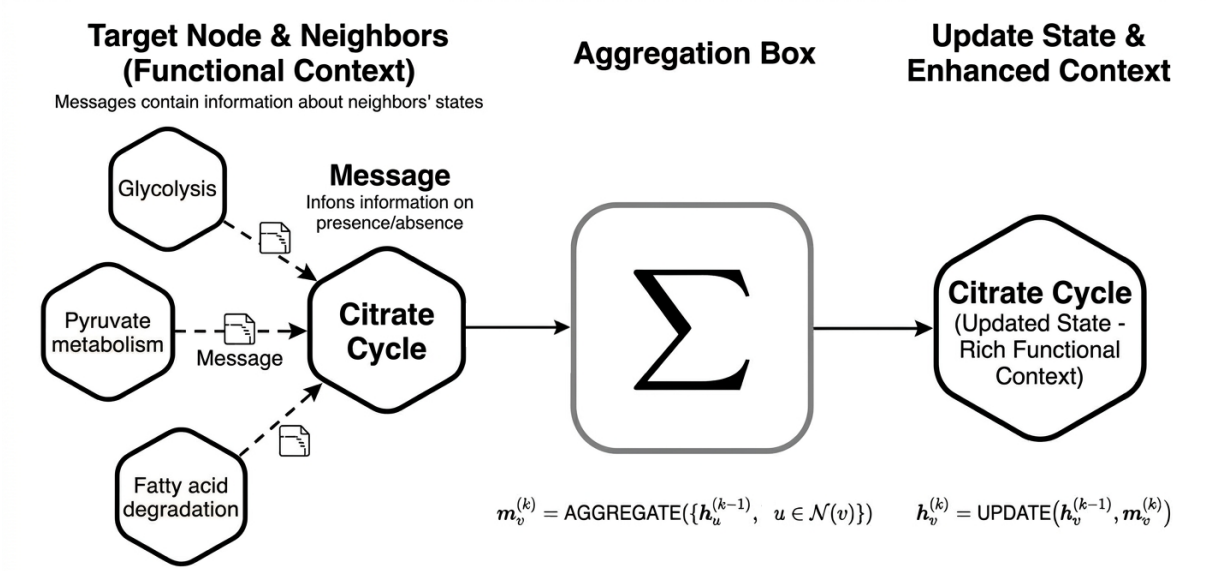


Figure 3.1: Biological adaptation of the Message Passing framework. A target hexagram (Citrate Cycle) aggregates information from its immediate topological neighbors. This process maps to the formal equation shown, where the neighbors' feature vectors $h_u^{(k-1)}$ form an aggregated message $m_v^{(k)}$, which is combined with the target node's initial state $h_v^{(k-1)}$ to produce the updated representation $h_v^{(k)}$.

topological perturbations. However, it is important to note that this comparison is inherently non-exact. Since checking for exact graph isomorphism is computationally infeasible, graph kernels rely on approximation strategy: they evaluate similarity by decomposing the complex, global graphs into simpler substructures, such as subtrees or paths, and comparing their relative frequencies. Consequently, while the localized structural decomposition provides a mathematically rigorous way to handle the noise and sparsity typical of biological data [9], it measures the overlap of local metabolic motifs rather than verifying a global structural identity between the two systems.

3.3.1 Weisfeiler-Lehman (WL)

The WL Subtree Kernel [13] is one of the most efficient and widely used graph kernels for labeled graphs. It is based on the *WL Color Refinement*, a classic algorithm originally designed to address the **Graph Isomorphism** problem, the challenge of determining whether two graphs are topologically identical despite different node orderings. The algorithm operates through an iterative labeling process that systematically "unfolds" the local structure of each node into a subtree. The procedural logic can be divided into four fundamental stages:

1. **Initial Labeling**: Each node v is assigned an initial label $l^{(0)}(v)$ based on its biological identity;

2. **Label Collection**: At each iteration $h > 0$, every node v collects the labels of its immediate neighbors $\mathcal{N}(v)$ into a multiset:

$$M^{(h)}(v) = \{\{l^{(h-1)}(u) : u \in \mathcal{N}(v)\}\}$$

   The use of a multiset is crucial as it preserves the frequency of each neighbor's label, capturing the exact local connectivity;

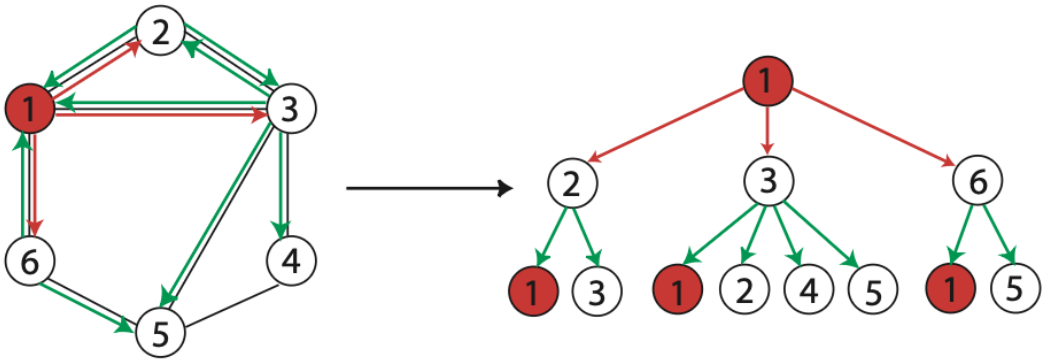


Figure 3.2: A subtree pattern of height 2 rooted at the node 1. Note the repetitions of nodes in the unfolded subtree pattern on the right (Source: [13])

3. **Label Refinement (Hashing)**: The node's current label and its neighbors' multiset are concatenated and hashed into a new unique label. To ensure a deterministic implementation, the elements of the multiset $M^{(h)}(v)$ are first sorted in a canonical order (e.g., lexicographically) to form a unique sequence. This sequence is then concatenated with the node's own previous label and processed by an injective hash function:

$$l^{(h)}(v) = \text{HASH}\left(l^{(h-1)}(v), \text{sort}(M^{(h)}(v))\right)$$

   This guarantees that the hashing process remains invariant to the original arbitrary permutation of the graph's adjacency list, assigning identical labels if and only if the nodes share the same structural history up to iteration h.

4. **Feature Vector Construction**: After H iterations, the algorithm produces a sequence of labels for each node. The kernel then constructs a global feature vector $\phi(G)$ by counting the occurrences of each unique label across all iterations:

$$\phi(G) = \left(c_1(G), c_2(G), \ldots, c_{|\Sigma|}(G)\right)$$

   where $c_i(G)$ is the count of the i-th label in the graph.

The similarity between two metabolic networks G and G' is computed as the dot product: $K_{WL}(G,G') = \langle \phi(G), \phi(G') \rangle$. Mathematically, the WL kernel captures subtree patterns: after h iterations, a node's label represents the entire structure of its neighborhood up to distance h. This allows the model to detect conserved structural "motifs" or topological patterns, such as specific enzymatic sequences, that are essential for accurate taxonomic classification.

The recursive decomposition mechanism of the WL kernel is depicted in Figure 3.2. Starting from an initial labeling (node 1 in red), the algorithm iteratively maps the topological context of each node into a rooted subtree of depth H. This process ensures that nodes with distinct neighborhood structures receive different hash labels, forming the backbone of the subsequent graph-level feature vector.

Limitations of the standard WL Kernel

Despite its widespread use, the standard WL kernel suffers from specific structural and categorical biases when applied to biochemical networks, which can degrade the quality of the learned representations:

- **Redundancy in 2-node Cycles (Backtracking)**: Metabolic networks, particularly in their bipartite representations, are densely populated with reversible reactions ($A \rightleftharpoons B$). In a standard WL refinement, these bidirectional edges induce a "ping-pong" effect, where a node v continuously receives its own previous state back from its neighbor u. This backtracking mechanism introduces a significant structural bias, inflating the similarity baseline among structurally distinct neighborhoods and artificially flattening the expressive capacity of the generated features.

- **Label Sensitivity**: The WL hash function treats every unique categorical label as an entirely independent entity, lacking any metric of partial similarity between different node annotations. Consequently, the kernel is highly sensitive to nominal mismatches. Two isolated reactions that are biochemically or structurally analogous, but annotated with distinct discrete labels, will be treated as completely orthogonal by the hashing process, preventing the model from recognizing underlying functional overlaps.

3.3.2 Enhanced Weisfeiler-Lehman (En-WL)

To overcome the backtracking limitations inherent in 2-node cycles, we implemented a custom, cycle-aware variation of the standard WL kernel. Instead of permanently contracting reversible edges (which would result in a loss of valuable information) we directly modified the core message-passing mechanism of the algorithm.

Our approach introduces a dynamic edge-pruning strategy that depends on the current iteration step, effectively breaking the infinite loop while preserving the local topology. The mechanism is structured in two distinct phases:

1. **Context Acquisition (Iteration $i = 0$)**: During the first step of the refinement, the aggregation proceeds exactly as in the standard WL algorithm. Every node v receives the labels from all its neighbors $u \in \mathcal{N}(v)$, including those connected by bidirectional edges. We can consider this step as an initial "handshake" that makes each node aware of its self-context and captures this information in the node's signature.

2. **Backtracking Prevention (Iteration $i > 0$)**: From the second iteration onwards, a topological filter is dynamically applied. When updating the label of node v, we evaluate each neighbor u. If the edge between them forms a 2-node cycle (i.e., both directed edges (u,v) and (v,u) exist in the graph), u is explicitly excluded from the aggregation neighborhood of v.

Formally, the filtered neighborhood $\mathcal{N}_f^{(i)}(v)$ used for the hash function at iteration i is defined as:

$$\mathcal{N}_f^{(i)}(v) = \begin{cases} \mathcal{N}(v) & \text{if } i = 0 \\ u \in \mathcal{N}(v) \mid \neg((v,u) \in E \wedge (u,v) \in E) & \text{if } i > 0 \end{cases} \tag{3.1}$$

As demonstrated in Figure 3.3, by cutting off the reciprocal message passing immediately after the initial exchange, this Cycle-Aware WL kernel completely eliminates the "ping-pong" effect. The network retains its structural integrity, and the kernel can successfully aggregate deeper, more complex structural hierarchies without being degraded by the noise of local oscillations. This ultimately leads to more robust and discriminative graph representations for metabolic networks.

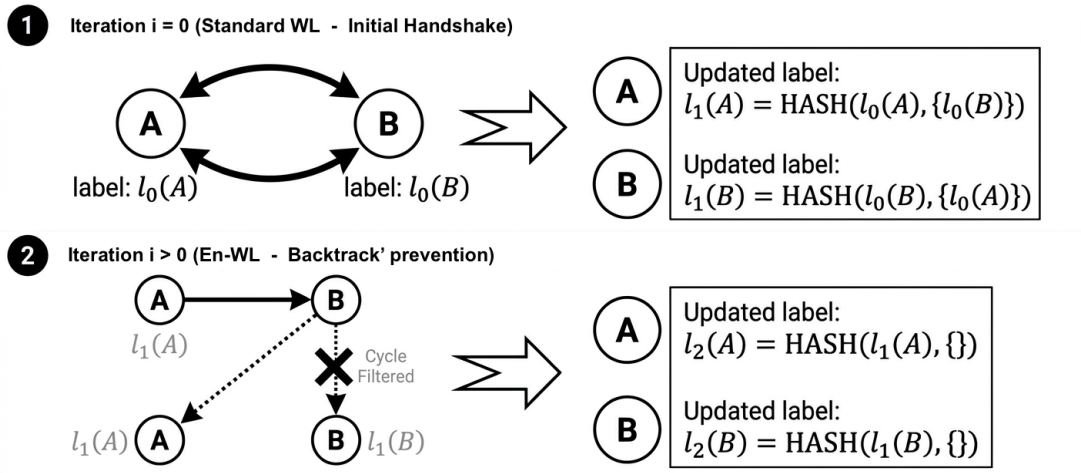


Figure 3.3: Schematic representation of the Backtracking Prevention mechanism in the En-WL kernel solution. At iteration $i = 0$, the algorithm performs an initial handshake, aggregating all neighbors to establish the node's baseline functional identity. From iteration $i > 0$ onwards, a dynamic topological filter is applied: edges forming 2-node cycles are pruned, effectively preventing redundant information from oscillating between nodes A and B. This ensures that the hash function generates unique, hierarchical structural labels instead of repeating local information exchange.

3.3.3 Ordered Decompositional DAG (ODD)

While the WL kernel and its variations focus on isotropic neighborhood aggregations, they often fail to capture the macro-level directional flows and hierarchical pathways embedded within metabolic systems. Although the majority of individual biochemical reactions are fundamentally reversible and function in both directions, metabolic networks as a whole drive substrates from primary nutrient inputs to essential biomass outputs.

To fully exploit this underlying directional logic without ignoring local reversibility, we employ the ODD Kernel. This framework decomposes the cyclic and bidirectional metabolic graph into a collection of ordered, directed acyclic graphs (DAGs), allowing the model to capture sequential path-based dependencies that characterize systemic metabolic flows.

The ODD kernel is specifically designed to operate on Directed Acyclic Graphs (DAGs), such as the mDAGs described in Section 2.4.3. Its main objective is to decompose a complex directed structure into a set of simpler, ordered substructures that respect the partial ordering of nodes.

The core mechanisms of the ODD kernel involve three key steps [30]:

1. **DAG Decomposition**: The input is decomposed into all its possible rooted sub-structures (Decomposition DAGs (DDs)). Since the graph is acyclic, these structures represent valid biological sequences of reactions without infinite loops.

2. **Ordered Aggregation**: Unlike the standard WL kernel, which aggregates all adjacent nodes into a single direction-agnostic multiset, the ODD kernel explicitly accounts for the structural role and direction of the edges. It distinguishes between "predecessor" nodes (substrates) and "successor" nodes (products), ensuring that the direction is preserved in the generated feature space.

3. **Recursive Kernel Computation**: The similarity between two DAGs is computed recursively by comparing their substructures. For two nodes $v \in G$ and $v' \in G'$, the kernel $K(v, v')$ is calculated based on the similarity of their labels and the recursive similarity of their respective successors.

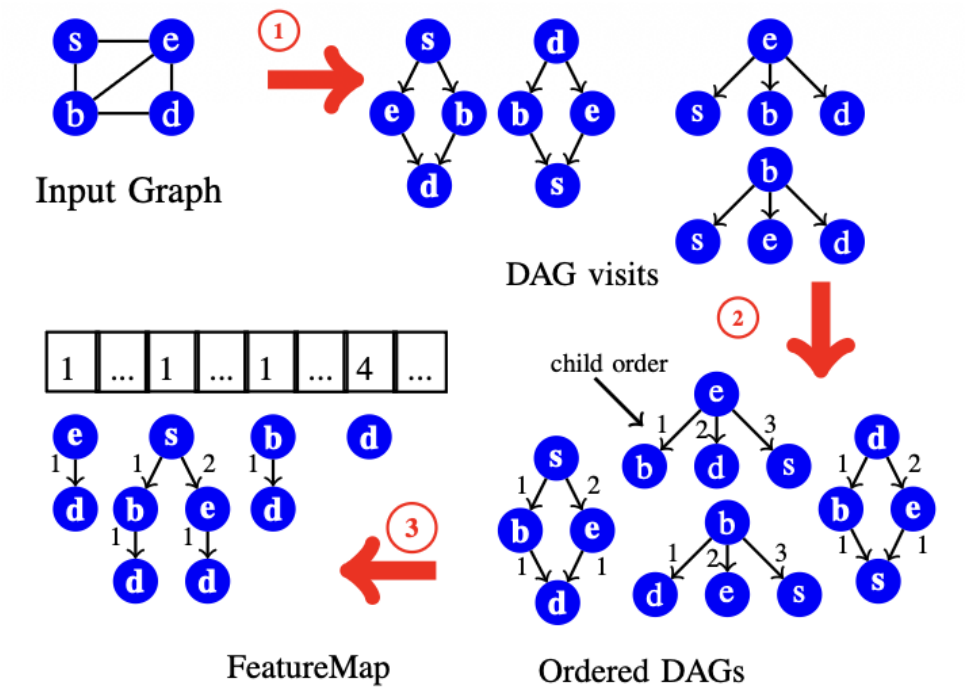


Figure 3.4: ODD kernel summary. 1) Decomposition of a graph into its DDs; 2) definition of a total ordering among the children of each node; 3) generation of an explicit FeatureMap extracting all proper subtrees from the set of Ordered DDs. (Source: [30])

The procedural framework and the decomposition logic of the ODD kernel are illustrated in Figure 3.4. Starting from an input directed graph, the algorithm performs a systematic traversal to generate a family of associated rooted DAGs. These sub-DAGs are structured by establishing a strict partial ordering based on the reachability of successor nodes, which formally preserves the directional hierarchy of the edges. Finally, a recursive matching mechanism evaluates the structural similarity between the decomposed sub-DAGs, yielding a robust feature map. In our context, this representation allows the model to map and compare the underlying topologies of different organisms without being constrained by local cyclic redundancies.

The application of the ODD kernel to biological datasets addresses the "backtracking" problem exposed in Section 3.3.1 by construction. By resolving SCCs and removing cycles during the graph construction phase, we allow the kernel to focus on the metabolic structure. Indeed, while the standard WL kernel struggles with circular motifs due to the message-pinging effect, the ODD kernel bypasses these loops to map the topological dependencies and directional constraints governing metabolic pathways. This approach avoids the constraints of strict linearity, providing a complementary structural perspective that effectively enhances the overall expressiveness of our framework.

3.4 GKNN and KA-GNN architectures

After extracting structural features through the graph kernels described in the previous section, we need a robust framework to perform supervised classification. While traditional GNNs learn representation through continuous message-passing between nodes, our approach focuses on Graph Learning methodologies, specifically employing the GKNN [15] and the KA-GNN [17] approach to integrate graph kernel into the predictive pipeline. By leveraging these advanced techniques, we try to bridge the gap between classical graph theory and deep learning, allowing

our models to map topological structures for accurate taxonomic classification.

3.4.1 Graph Kernel Neural Network (GKNN)

The GKNN framework redefines the convolution operator for graph-structured data by replacing Euclidean filters with graph kernels. Unlike standard GNNs that rely on continuous message-passing, GKNN operates directly on the discrete structural properties of the network [15].

The architecture introduces a structural version of the convolution operator, called Graph Kernel Convolution (GKC), which avoids the need for explicit node embeddings by working directly on the set of graphs. Unlike standard GNNs where filters are vectors of weights, a GKC layer is defined by a set of structural masks $\{\mathscr{M}_1,\ldots,\mathscr{M}_m\}$ where each $\mathscr{M}_i = (V_{\mathscr{M}}, X_{\mathscr{M}}, E_{\mathscr{M}})$ is a first-order stochastic graph. The edges $e \in E_{\mathscr{M}_i}$ are sampled from independent Bernoulli trials with parameters θ_e while each node label $x \in X_{\mathscr{M}_i}$ is sampled from the corresponding discrete probability distribution over a dictionary $\mathbf{D}$ with parameters $\phi_{xd} \forall d \in \mathbf{D}$, indicating the probability of x assuming value d. Parameters θ and ϕ are the learnable parameters of the GKC layer, making the masks adaptive to the data.

Given an input metabolic graph $G = (V, X, E)$, the layer computes the structural similarity between the local neighborhood of each node v and the masks. For a node $v \in V$ let $\mathscr{N}^r_{\mathscr{G}}(v)$ be the subgraph of radius r centered at vertex v. The output feature vector $z(v)$ for node v is computed as:

$$z_i(v) = \mathscr{K}(\mathscr{M}_i, \mathscr{N}^r_{\mathscr{G}}(v)), \tag{3.2}$$

where $\mathscr{K}(\cdot,\cdot)$ is a valid positive definite graph kernel, such as WL, En-WL or the ODD kernel. This operation effectively maps the discrete topological space into a continuous vector space through the kernel similarity.

The continuous feature output vector $z(v) \in \mathbb{R}^m_+$ contains the graph kernel responses between the structural masks and the subgraph centered on the node v. For graph kernels requiring discrete labels, the model interprets $z(v)$ as an unnormalized probability distribution over a set of m labels from which we can derive a new set of labels $x(v)$ for node v, used as input for the next GKC layer.

Letting $\mathbf{Z}^l$ denote the output features of the l-th layer, the final graph embedding (focusing on graph-level tasks) is obtained by the concatenation (indicated as $\cdot \mid \cdot$) of node-wise features:

$$\mathbf{Z} = \square\left(\mathbf{Z}^1 | \ldots | \mathbf{Z}^L\right)$$

This embedding is then passed through a Multi-Layer Perceptron (MLP) to obtain the final classification. In this context, the MLP acts as a non-linear classifier that operates on the structural feature space defined by the kernels, mapping the learned topological signatures into the target taxonomic labels. This allows the model to capture complex dependencies between different metabolic substructures that are relevant for distinguishing biological groups.

The overall architectural flow is illustrated in Figure 3.5. As shown, the input metabolic graph G undergoes one or more GKC layers, where the kernel responses between local subgraphs $\mathscr{N}^r_{\mathscr{G}}(v)$ and the learned structural masks $\mathscr{M}_i$ are computed. The resulting node-level features are aggregated through the global concatenation operator $\square$, which captures topological information. This unified embedding $\mathbf{Z}$ is finally passed to the MLP for taxonomic classification, completing the end-to-end learning process.

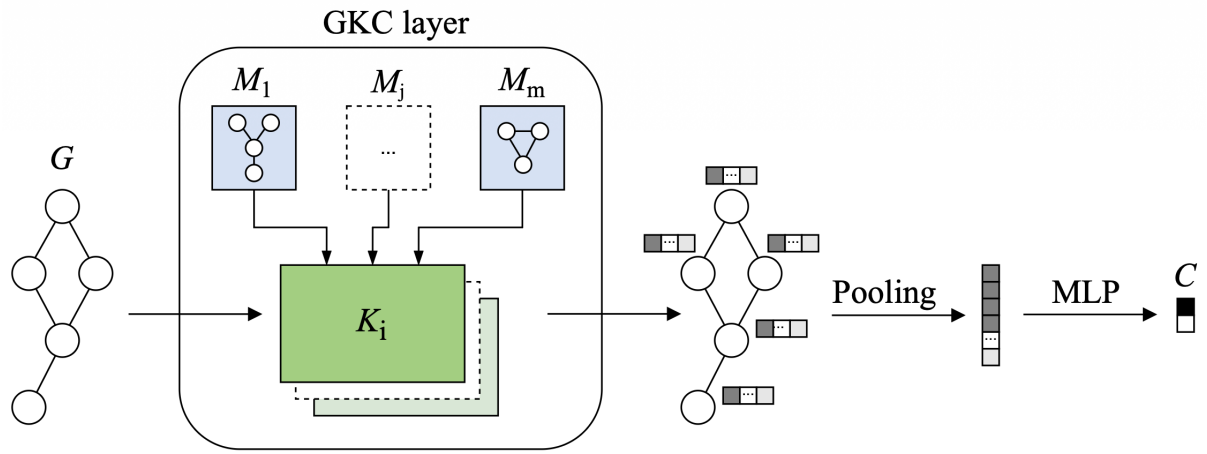


Figure 3.5: GKNN Architecture. The input graph is fed into one or more GKC layers, where subgraphs centered at each node are compared against a series of structural masks using a kernel function. This generates a new set of real-valued feature vectors associated with graph nodes. A graph-level feature vector is obtained by pooling the node features, which is then fed to an MLP to produce the final classification label (Adapted from: [15])

The Learning Problem and Loss Function

The training of a GKNN is formulated as a supervised learning problem where, given a set of B training graphs $\{\mathscr{G}_1,\ldots,\mathscr{G}_B\}$ and their associated labels y_i, the goal is to minimize a composite objective function. The total loss is defined as:

$$loss = loss_{CE} + \lambda loss_{JSD}, \tag{3.3}$$

where $loss_{CE}$ is the **standard cross-entropy** loss between the predicted class probabilities and the ground truth:

$$loss_{CE} = \sum_{i=1}^{B} \text{cross-entropy}(\mathbb{E}_{\Theta}[g_{\mathscr{M}_1,\ldots\mathscr{M}_m,\Phi}(\mathscr{G}_i)], y_i) \tag{3.4}$$

In Equation 3.4, g represents the entire neural network (including GKC layers and the MLP) parameterized by Φ and by the edge probabilities Θ of the masks $\mathscr{M}_i$. The goal is to optimize these parameters in order to minimize the cross-entropy loss.

The second term, $loss_{JSD}$, is a **Jensen-Shannon Divergence** loss designed to prevent the masks from collapsing into similar responses. Let $P_i = \{\alpha z_i(v) \mid v \in V\}$ be the probability distribution induced by the i-th mask over the nodes of a graph, where α is a scaling factor. The JSD loss is defined as:

$$loss_{JSD} = -H\left(\sum_{i=1}^{m} P_i\right) + \sum_{i=1}^{m} H(P_i) \tag{3.5}$$

where $H(P)$ is the Shannon entropy. Maximizing this term forces the learned distributions P_i to be as far from one another as possible, ensuring that each structural mask specializes in a unique metabolic motif.

3.4.2 Kolmogorov-Arnold GNN (KA-GNN)

The KA-GNN framework introduces a fundamentally innovative approach to graph representation learning by replacing the MLPs traditionally used in GNNs with Kolmogorov-Arnold Networks (KANs). While standard GNNs apply fixed activation functions at the node level

and learn weights on edges, KANs are grounded on the Kolmogorov-Arnold superposition theorem, which states that any multivariate continuous function can be decomposed as a finite composition of univariate continuous functions. This theoretical foundation enables accurate and interpretable modeling of complex functions, yielding a more expressive architecture. The adoption of learnable activation functions on each edge of the network leads us to embed KAN modules into GNN by creating a unified framework called KA-GNN that includes node embedding, message passing and readout [17].

Fourier-based KAN layer

The specific KAN variant adopted in KA-GNN is *Fourier-based KAN*, which parametrizes the learnable univariate activation functions to effectively capture both low-frequency and high-frequency structural patterns in graphs. Starting from the superposition theorem reported above, KANs generalize it by constructing each layer as a sum of learnable univariate functions.

For an input vector x, the Fourier-based KAN activations function is defined as:

$$f(x) = \sum_{k=1}^{K} \left(a_k \cos(kx) + b_k \sin(kx)\right) \tag{3.6}$$

where K is the number of Fourier harmonics (a hyperparameter controlling expressiveness) and a_k, b_k are learnable coefficients used to dynamically adjust the shape of the activation function during training. Unlike standard neurons that apply a fixed non-linearity after a linear projection, each activation in a KAN is itself a learnable function, making the network adaptive at the level of individual inputs. While original KAN implementations rely on B-splines, we employ a Fourier-based parametrization for its superior computational efficiency and the basic allowance to function as a learnable filter, where the number of harmonics provides a constraint on model complexity. This Fourier-based approach is particularly suited for biological networks, as it allows the model to act as a learnable filter in the domain. By tuning the coefficients a_k and b_k, the KA-GNN can emphasize specific topological frequencies, effectively distinguishing between ubiquitous metabolic pathways (low-frequency) and organism-specific interactions (high-frequencies).

KA-GNN Architecture

The distinguishing feature of KA-GNN is the systematic replacement of MLPs with KAN operators at all three stages of the GNN pipeline: Node embedding initialization, message passing and graph-level readout. The flowchart of the described architecture is illustrated in Figure 3.6.

Node initialization (Stage 1) The input feature vector of each node $v \in V$ is first projected into a hidden space by incorporating both node features and the features of its neighborhood $\mathcal{N}(v)$ through KAN layer:

$$\mathbf{h}_v^{(0)} := \text{KAN}\left(\mathbf{f}_v \oplus \left(\frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \mathbf{f}_{vu}\right)\right) \tag{3.7}$$

where $\oplus$ denotes the vector concatenation and $\mathbf{f}_{vu}$ represents the message or feature vector derived from the neighbor u to the central node v.

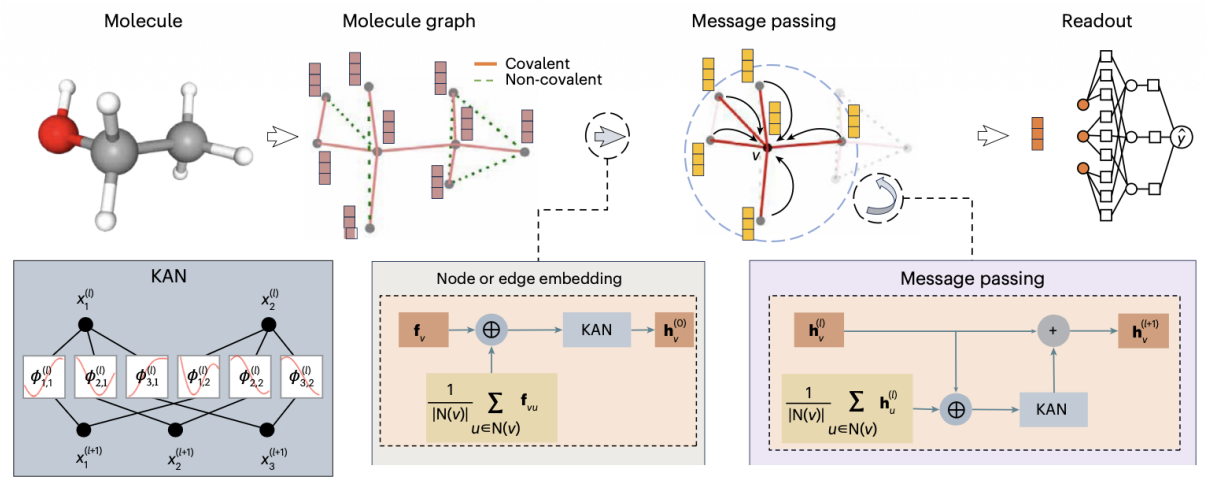


Figure 3.6: KA-GNN Architecture. Node features are first projected into a hidden space via a `KAN` layer (Stage 1). Iterative message passing is then performed through `KAN` blocks with residual connections (Stage 2), followed by average pooling and a final `KAN` readout producing the classification (Stage 3). (Adapted from: [17])

Message passing (Stage 2) For each of the $L-1$ layers, messages are computed by applying the KAN layer to the neighborhood of each node:

$$\mathbf{h}_v^{(l+1)} := \mathbf{h}_v^{(l)} + \mathrm{KAN}\left(\mathbf{h}_v^{(l)} \oplus \left(\frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} \mathbf{h}_u^{(l)}\right)\right) \tag{3.8}$$

In this way, the feature interactions are dynamically adjusted and the information across layers are preserved during the execution.

Graph-level readout (Stage 3) After L message-passing steps, graph-level representations are obtained by average pooling over all node embeddings:

$$\mathbf{y} := \mathrm{KAN}\left(\frac{1}{|V|}\sum_{v\in V} \mathbf{h}_v^{(L)}\right) \tag{3.9}$$

The KAN-based readout dynamically learns complex patterns in the chosen graph. By substituting KANs at every stage, the model gains flexibility in learning complex frequency-based representations of graph topology, with the number of harmonics K acting as a direct control over the expressiveness of each layer.

The original KA-GNN extracts node features from graphs using continuous atomic descriptors. Since our metabolic graphs use discrete integer node labels with no associated continuous attributes, we replace this feature extraction with a trainable embedding layer. To provide immediate local context, each embedding is concatenated with the mean embedding of its direct neighbors before being passed to the first KAN layer.

Chapter 4

Case study

This chapter presents the empirical evaluation of the proposed graph representation frameworks applied to a real-world biological classification task. The primary objective is to investigate how distinct topological abstractions (specifically AMNs, mDAGs and RGs) impact the predictive performance, computational efficiency, and generalization capabilities of machine learning architectures. To achieve this, we cross-evaluate and contrast two distinct learning paradigms: the structural GKNN [15], representing graph kernel-based methods, and our proposed KA-GNN framework [17], which leverages the expressiveness of Kolmogorov-Arnold Networks within a message-passing architecture.

The core task focused on predicting the taxonomic kingdom of an organism strictly from its underlying metabolic network structure. This classification problem serves as a robust proxy to evaluate whether compressed graph representations successfully preserve the deep evolutionary and functional signatures natively embedded within metabolic pathways, validating their expressiveness and network properties.

4.1 Dataset details

The empirical evaluation of the proposed graph representations relies on a comprehensive collection of metabolic network compiled from the KEGG repositories. This section provides a detailed quantitative overview of the core dataset, details the sub-sampling and stratification strategies designed to mitigate taxonomic imbalance, and formalizes the notation adopted to uniquely identify the different dataset size variants and taxonomic distributions throughout the benchmarking phase.

4.1.1 Dataset Overview and Subsets

The dataset used in this study consists of metabolic data sourced from the KEGG database [4, 5]. The complete dataset comprises 8904 organisms distributed across six biological kingdoms: Animals, Plants, Fungi, Protists, Bacteria and Archaea.

As reported in Table 4.1, the distribution is heavily skewed, with Bacteria accounting for 85.52 % of the total. This class imbalance biases the standard overall accuracy; indeed, a majority-class classifier selecting Bacteria for every single input would artificially achieve an accuracy of approximately 85.52 %, failing to evaluate the model's performance on minority classes such as Plants (1.56 %) or Protists (0.63 %).

To enable a balanced evaluation, two subsets were constructed by capping the maximum number of graphs per kingdom. The first subset limits each kingdom to 300 graphs, resulting in

Kingdom	Total	%Total	Sub_300	%Sub_300	Sub_600	%Sub_600
Animals	535	6.01%	300	24.02%	535	28.32%
Plants	139	1.56%	139	11.13%	139	7.36%
Fungi	154	1.73%	154	12.33%	154	8.15%
Protists	56	0.63%	56	4.48%	56	2.96%
Bacteria	7615	85.52%	300	24.02%	600	31.76%
Archaea	405	4.55%	300	24.02%	405	21.44%
Total	**8904**	**100%**	**1249**	**100%**	**1889**	**100%**

Table 4.1: Distribution of organisms across the six biological kingdoms in the full dataset and in the two balanced subsets.

a total of 1249 graphs. The second subset allows up to 600 graphs per kingdom, yielding 1889 graphs. We need to specify that for kingdoms with fewer samples than the respective cap (e.g., Plants with 139 organisms), all available graphs are used. Table 4.1 summarizes the full dataset distribution alongside both subsets used in the analysis.

To maintain a consistent notation, each dataset variant is designated by its representation acronym followed by the cardinality of the largest class group. For example, `RGs_300` denotes the Reaction Graphs (RGs) subset capped at 300 samples per kingdom, while `AMN_9000` refers to the complete dataset in the Abstract Metabolic Network (AMN) representation.

Both methods evaluated in this work, the Graph Kernel Neural Network (GKNN) (Section 3.4.1) and the Kolmogorov-Arnold Graph Neural Network (KA-GNN) (Section 3.4.2), are initially trained on the two balanced subsets. To demonstrate the generalization capability of the models and ensure they do not simply overfit to the artificial balanced distribution, the best-performing configurations from the training phase are subsequently evaluated on the full, unbalanced dataset of 8904 graphs. Furthermore, it is important to note that, prior to any training or evaluation, the RGs datasets undergo a preprocessing step concerning node relabeling strategies, the details of which are described in Section 4.2.

4.1.2 Graph-level basic structural analysis

Analyzing the intrinsic topological properties of the graphs provides a better understanding of the dataset beyond class distributions. The three representations introduced in Section 2.4, encode metabolic information at different levels of granularity, which is directly reflected in the structural statistics of their respective graph collections.

Table 4.2 summarizes the average number of nodes, isolated nodes, node degree and edges across all graphs in the full dataset, broken down by representation and kingdom. The data confirm a clear hierarchical ordering in terms of node cardinality: AMNs are the most compact structures, averaging between 60 and 96 nodes, whereas RGs are the most detailed, reaching averages of up to 1880 nodes for Plants. In terms of vertex count, mDAGs occupy an intermediate position, consistent with their construction as condensation of the underlying RGs (Section 2.4.3). However, this hierarchy does not strictly hold for edge counts. For instance, in Bacteria and Fungi, AMNs exhibit a slightly higher average number of edges than mDAGs. This occurs because AMNs condense metabolic pathways into denser, highly interconnected abstractions, whereas mDAGs preserve a sparse, more fragmented backbone inherited from the underlying RGs.

Repr.	Kingdom	Avg. Nodes	Avg. Isolated	Avg. Max Degree	Avg. Edges
AMN	Animals	83	13	14	358
	Plants	96	13	17	556
	Fungi	80	11	17	479
	Protists	60	10	11	231
	Bacteria	79	10	17	490
	Archaea	67	8	15	364
mDAGs	Animals	798	108	18	783
	Plants	823	108	19	848
	Fungi	514	99	18	478
	Protists	342	80	11	285
	Bacteria	440	82	14	416
	Archaea	365	80	10	325
RGs	Animals	1847	96	14	4850
	Plants	1880	97	16	5031
	Fungi	1353	86	14	3800
	Protists	742	67	10	1675
	Bacteria	1060	71	14	2740
	Archaea	679	61	11	1435

Table 4.2: Summary of node and edge statistics across the three metabolic network representations for the six biological kingdoms. Values are averaged over all graphs in the full dataset.

4.1.3 Vertex connectivity analysis

The distribution of vertex connectivity constitutes a fundamental topological descriptor of a network. In metabolic graphs, the degree of a node reflects its structural integration: in a Reaction Graph (RG), it indicates the number of other reactions with which it shares direct metabolite flows, whereas in an Abstract Metabolic Network (AMN), it represents the macro-level functional interconnections between pathways. The shape of this distribution directly influences the behavior of graph learning methods. The GKNN operates by comparing local subgraphs centered at each node against structural masks, meaning that local connectivity peaks govern the complexity of individual neighborhoods. Conversely, the KA-GNN aggregates neighborhood embeddings via message passing, making it sensitive to extreme structural asymmetries where a few heavily centralized nodes can dominate the learned representations.

When analyzing aggregate statistics across a large collection of graphs, a crucial methodological distinction must be made. Rather than pooling every individual vertex degree across the entire dataset, the global and per-kingdom profiles illustrated in Figure 4.1 capture the distribution of the *maximum degree achieved within each individual graph* (Graph Max Degree). This distinction successfully reconciles the apparent mathematical tension between the box plot central tendencies and the standard topological averages reported in Table 4.2.

As shown in the global distribution of Figure 4.1, mDAGs exhibit the highest absolute topological asymmetry, reaching extreme outlier peak values up to 202 in the full dataset [16]. This sharp tail is driven by core metabolic intermediates that act as global structural consolidators within large MBBs, where the cycle condensation algorithms collapse feedback loops and merge numerous converging reaction flows onto single pivot nodes. However, despite these prominent

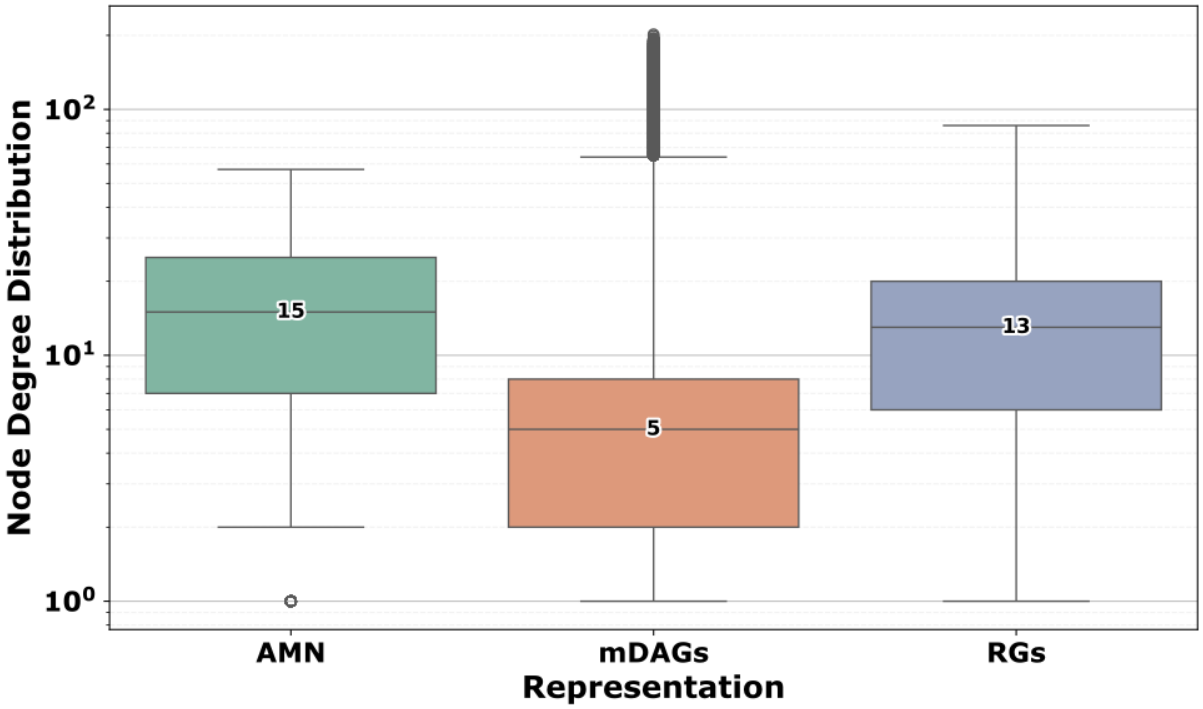


Figure 4.1: Box Plot of graph maximum degree distribution across the three representations (full dataset). The logarithmic scale on the y-axis highlights variations in peak localized connectivity. mDAGs display a heavily skewed distribution characterized by extreme outliers up to 202, while AMN and RGs exhibit more balanced profiles with higher median maximum values [16].

outliers, mDAGs present a considerably lower median peak degree of 5. This demonstrates that while mDAGs develop highly centralized hubs in their core machinery, the majority of the graphs in the collection retain an otherwise sparse and streamlined backbone.

Conversely, RGs and AMNs display a more distributed and uniform maximum degree profile across their graph collections. RGs display a median graph maximum degree of 13, capping their absolute peak at 86, reflecting a topology where connectivity is distributed smoothly along linear metabolic pathways without collapsing into artificial macro-hubs. AMNs smooth out these structural spikes even further due to their higher functional abstraction, shifting their median peak toward 15 while strictly capping their maximum absolute degree at 57.

These architectural differences directly affect the performance of the graph learning methods. Methodologies relying on local neighborhood comparisons via graph kernels, such as the GKNN framework, can suffer from feature dilution when encountering these high-degree nodes, as heavily centralized neighborhoods might obscure fine-grained structural differences. Conversely, the KA-GNN architecture effectively processes these distributional asymmetries: its learnable Fourier-based activation function acts as adaptive filters, allowing the network to untangle asymmetric topological configurations by separating low-frequency structural backbones from high-frequency localized variations induced by prominent hubs.

When examining the per-kingdom breakdowns in Figure 4.2, a consistent taxonomic trend emerges regarding the absolute size of the graphs, while localized connectivity reveals distinct patterns. The average graph sizes (Table 4.2) of Bacteria and Archaea are systematically smaller compared to those of multicellular Eukaryotes like Animals and Plants, reflecting the structural compactness and lack of tissue-specific compartmentalization in prokaryotic genomes.

However, this reduction in scale does not translate to a uniform reduction in peak node degree. Within the AMN and RG formalisms, Bacteria retain high average maximum degrees

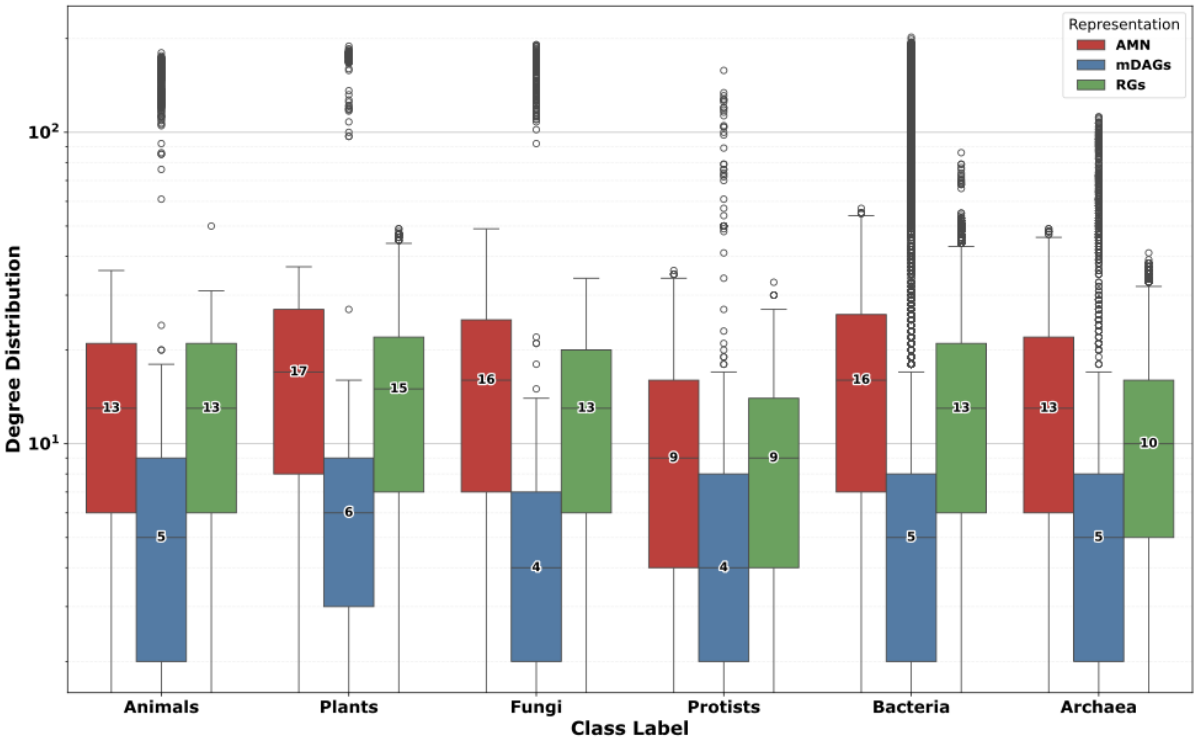


Figure 4.2: Box Plots of graph maximum degree distribution for the six biological kingdoms across the three representations. Each box highlights the spread, central tendencies, and peak outliers of maximum graph connectivity, visualized on a logarithmic y-axis to clarify cross-representation patterns [16].

(matching or exceeding those of Animals), highlighting heavily centralized hub structures within a tighter backbone. Conversely, the Archaea kingdom exhibits a systematic attenuation in peak degree across all representations. Rather than an inherent pathway simplicity, this lower baseline is further amplified by the current gap in comprehensive enzymatic annotations for non-model organisms in global databases, which artificially truncates the peripheral branches of their metabolic networks.

4.1.4 Connected Components Analysis

The presence of multiple connected components in a metabolic network graph is deeply tied to the underlying graph abstraction model. In Reaction Graphs (RGs), a weakly connected component corresponds to a set of biochemical reactions linked through shared metabolites. Conversely, isolated nodes represent reactions or modular units that share no annotated metabolic flow with the rest of the organism's system.

Quantifying this structural fragmentation across different abstractions and taxonomic kingdoms is crucial, as it directly impacts the computational capabilities of graph learning frameworks. The GKNN computes kernel similarities over localized subgraphs, meaning that a disconnected or isolated node has a zero-degree neighborhood. Similarly, KA-GNN aggregate neighborhood embeddings via message passing, consequently, highly fragmented graphs result in disconnected subgraphs and isolated components, preventing them from receiving or propagating structural signals across the network.

Table 4.3 reports the connected component statistics across representations and kingdoms for the full dataset. When interpreting these metrics, the *Giant (%)* column serves as the primary topological gauge for global network integration, as it quantifies the exact fraction of total nodes

Repr.	Kingdom	Avg. Comp.	Avg. Giant	Giant (%)	Isolation (%)
AMN	Animals	14.9	67.3	81.6%	16.0%
	Plants	14.3	82.7	86.2%	13.8%
	Fungi	11.8	68.9	86.2%	13.7%
	Protists	11.1	50.0	82.1%	16.9%
	Bacteria	11.6	68.0	85.9%	13.4%
	Archaea	8.7	59.0	87.8%	11.7%
mDAGs	Animals	166.8	419.5	52.0%	14.4%
	Plants	150.2	529.2	64.1%	13.3%
	Fungi	127.9	316.7	60.7%	19.8%
	Protists	108.9	169.3	47.2%	24.7%
	Bacteria	104.5	286.7	63.8%	19.2%
	Archaea	104.0	196.1	53.1%	22.1%
RGs	Animals	166.8	1394.5	75.4%	5.4%
	Plants	150.2	1546.7	82.1%	5.2%
	Fungi	127.9	1120.6	81.4%	6.6%
	Protists	108.9	520.8	66.5%	10.1%
	Bacteria	104.5	875.8	80.3%	7.2%
	Archaea	104.0	470.8	68.5%	9.3%

Table 4.3: Connected component statistics across representations and kingdoms for the `Full dataset` (8904 graphs). For each graph the two quantities are computed with respect to its total number of nodes and then averaged over all graphs of the kingdom. *Giant (%)* is the fraction of nodes belonging to the largest weakly connected component, while *Isolation (%)* is the fraction of degree-zero (isolated) nodes, i.e. the number of isolated nodes divided by the total number of nodes in the graph.

successfully clustered within the single largest interconnected core. Empirical analyses on the smaller subsets (300 and 600 samples) yielded nearly identical distributions, demonstrating that the observed topological characteristics are highly stable and independent of the sample size.

The empirical values reveal a counter-intuitive topological behavior across the three abstractions, which is visually summarized in the giant component heatmap of Figure 4.3. Indeed, RGs emerge as remarkably cohesive structures. Despite having a high absolute number of connected components (averaging around 104–166 depending on the kingdom), they retain a very large giant component that encompasses 66 % to 82 % of the entire network, coupled with a minimal ratio of isolated nodes (5 % - 10 %). This indicates that the main metabolic core in RGs remains heavily integrated, and the multiple secondary components are mostly small, peripheral pathways or individual reaction steps.

In contrast, mDAGs exhibit the most severe structural fragmentation within the dataset, a phenomenon clearly visible in Figure 4.3 where the entire mDAG column shifts toward significantly lighter and less integrated color gradients. While they share the exact same average number of components as RGs (as a direct artifact of their structural construction), their giant component significantly reduced, dropping down to 47.2 % in Protists, 52.0 % in Animals, and 53.1 % in Archaea. Concurrently, the fraction of isolated nodes spikes dramatically, reaching nearly 25 % in Protists. This behavior is a direct consequence of the DAG conversion process: by breaking feedback loops, resolving strongly connected components (SCCs), and enforcing an acyclic ordering, the mDAGs abstraction intentionally isolates specific metabolic building

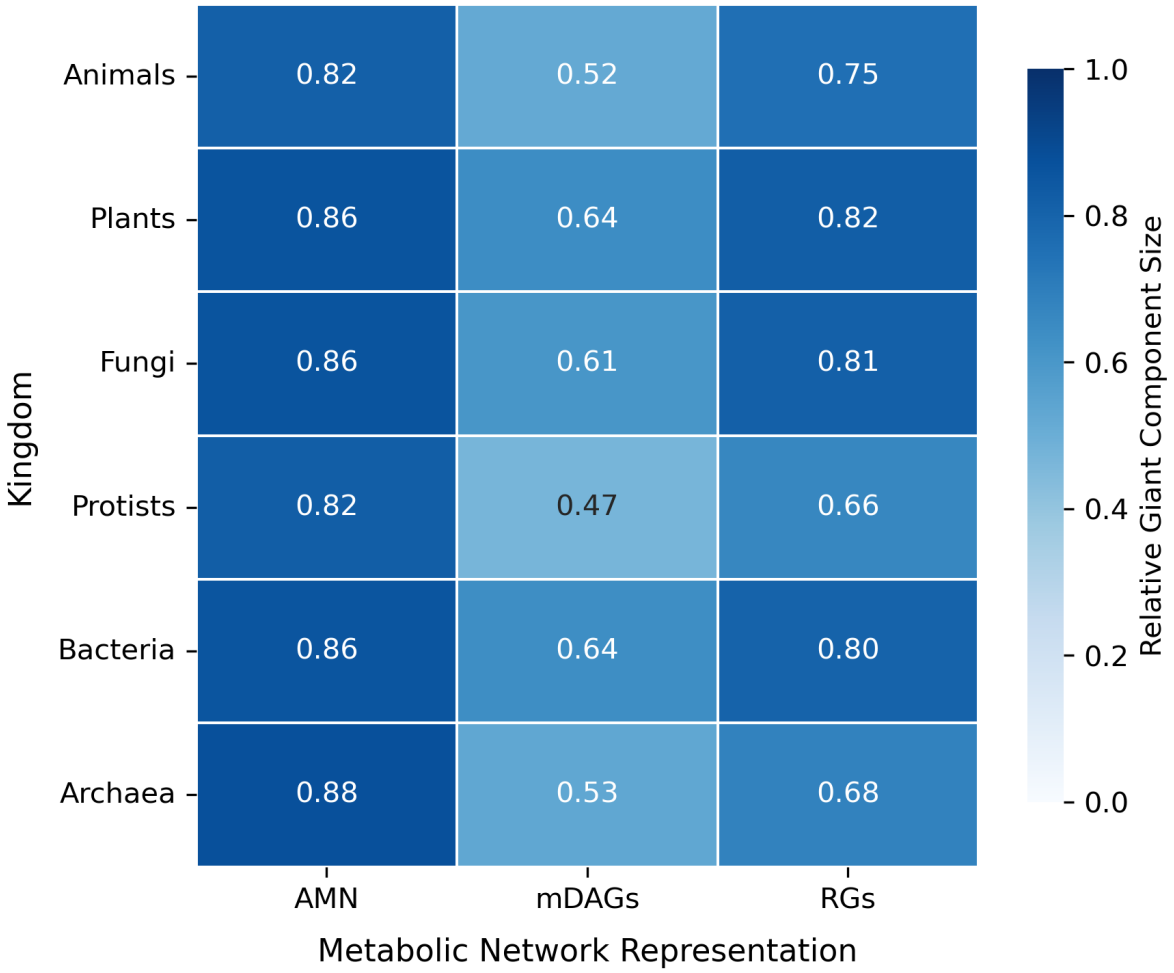


Figure 4.3: Average fraction of nodes belonging to the giant connected component for each combination of representation and kingdom (full dataset). The colorbar on the right maps the *Relative Giant Component Size* (from 0.0 to 1.0), where darker cells indicate higher global topological integration.

blocks (MBBs), heavily fragmenting the network backbone across all taxonomic groups. Finally, AMNs display a completely different scale, characterized by extremely small graphs (reflected in the low absolute size of the giant component, e.g., 50.0 nodes in Protists) but a highly compressed, globalized connectivity.

When examining the horizontal taxonomic trends across Figure 4.3, a critical biological anomaly emerges in the baseline representations. The lowest connectivity scores are consistently achieved by Protists and Archaea, a pattern particularly evident within the RG column where their giant components drop below 70 %. Rather than biological simplicity alone, this pattern stems from smaller genomes and missing annotations in non-model organisms, whose highly divergent pathways remain less characterized [28, 29]. Conversely, Bacteria maintain highly integrated cores and large giant components across all formalisms (63.8 % in mDAGs and 80.3 % in RGs), performing comparably to Plants. This trend is heavily driven by the extensive curation, high structural homogeneity, and functional characterization that bacterial genomes receive in current metabolic databases, matching the density of heavily integrated plant networks despite the substantial difference in physical node count [29].

4.2 Node relabeling strategies

This section describes three node relabeling strategies developed for applying a GKNN framework with the WL kernel to a dataset of metabolic RGs. Each strategy assigns a different numerical representation to each node, encoding different aspects of the biochemical information available in the labels of the chosen dataset.

In the original format, each node is identified by a single integer label. When the WL kernel is applied, these labels are iteratively refined by aggregating the labels of the neighborhood of the analyzed node, producing a hierarchy of "color" histograms that capture the local graph structure. The RGs dataset contains metabolic reactions graphs. As described in Section 2.4.2, each node represents a reaction identified in the raw data by a label string such as "R00710(1.2.3.4)" or "R04969 | R04970(1.3.1.9 | 1.3.1.10)". These strings encode two distinct types of biochemical information:

1. **Reaction identifier** (e.g., R00710, R04969), which name the metabolic step catalyzed at that node. We can even find a trailing "_rev" suffix that indicates the reaction in the reverse direction.

2. **Enzyme representation** (e.g., 1.2.3.4), the Enzyme Commission number (EC), which classify the enzyme catalyzing the reaction according to a four-level hierarchy (L1.L2.L3.L4).

A single node may carry more than one reaction and/or enzyme when separated by a pipe character ("|"), indicating that the node is involved in multiple transformations. The four encodings described in this section differ in how they encode these two dimensions of each label and in the trade-off between vector dimensionality for one-hot encoding in WL iteration and semantic meaning.

To ensure absolute methodological transparency, the direct association between each relabeling method and the resulting feature vocabulary size is centralized in Table 4.4.

Relabeling	Biochemical Encoding	Vocabulary Size
Baseline (§4.2.1)	Atomic raw string (Reaction + EC + Direction)	6273 labels
JR (§4.2.2)	Reaction identity only (Direction collapsed)	5163 reactions
SE (§4.2.3)	Hierarchical EC levels only (4-block multi-hot)	419 bits
ER (§4.2.4)	Reaction identity, EC levels, explicit direction bit	5583 bits

Table 4.4: Direct association between the proposed node relabeling strategies and their respective feature space dimensions.

4.2.1 Baseline

This type of encoding is the starting point from which all three relabeling strategies depart. It treats each unique raw label as it appears in the mapping CSV, so as an atomic symbol with no internal structure exploited. Each distinct label string (e.g., "R00710(1.2.3.4)",

“R00710_rev(1.2.3.4)”, “R04969 | R04970(1.3.1.9 | 1.3.1.10)”), is marked with a unique integer ID during preprocessing, resulting in a list of 6273 different labels across the full dataset. The baseline makes no distinction between the reaction part and the enzyme part of a label, neither between forward and reverse variants. Two nodes labeled “R00710(1.2.3.4)” and “R00710_rev(1.2.3.4)” receive completely different one-hot vectors, even though they differ only in reaction direction and share the same enzyme.

This encoding is maximally expressive in the sense that no information is lost or merged, indeed every distinct label string gets its own vector. However, there are different costs:

- *High dimensionality* and *sparsity*. The kernel sees two nodes as completely dissimilar unless they share the exact same label string.
- *No generalization across related reactions*. As in the example above, two reactions that represent the same biochemical step in opposite direction are treated as unrelated.
- *No generalization across related enzymes*. Two nodes catalyzed by (1.2.1.3) and (1.2.1.4) have no shared features in this representation, while in concrete they share three levels out of four.

The goal of the three relabeling strategies (JR, SE, ER) is to address these limitations by deliberately decomposing the label string and encoding only the information that should be relevant for classification task.

4.2.2 Just Reactions (JR)

The JR encoding focuses exclusively on reaction identity, entirely discarding enzyme information. Its key design choices are:

- The enzyme part “(1.2.3.4)” is removed from the string;
- The “_rev” suffix is removed, so a forward and a reverse variant of the same reaction are treated as the same base reaction;
- Pipe-separated reaction are kept joined as a single string, so for example “R00431 | R00726” becomes a distinct label from “R00431” alone;

After collecting all unique cleaned strings, each gets an integer up to 5163, that corresponds to the total number of different reactions without considering the reversed ones.

Reaction can be considered as the most direct structural descriptor of a metabolic graph node. In a metabolic network, two nodes sharing the same reaction perform an identical biochemical transformation, regardless of where they appear in the network. In this way, when the kernel elaborates this type of information, it is able to distinguish nodes by what they do rather than their topological position. By collapsing the “_rev” variants we tried to reduce sparsity and ensure that structurally equivalent transformations are recognized as equivalent by the kernel. The JR representation is sparse and high-dimensional, but it preserves the basic identity of each reaction.

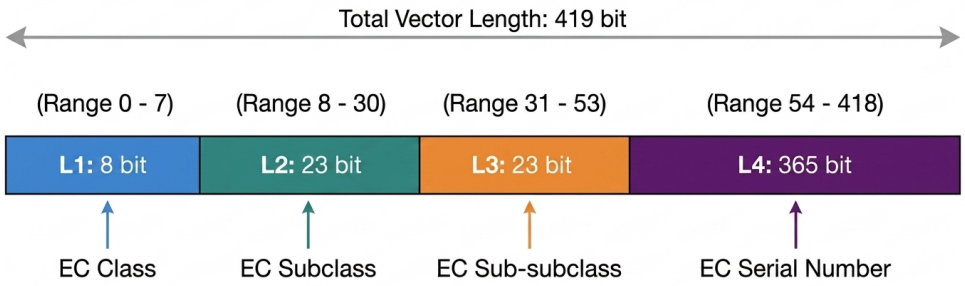


Figure 4.4: Schematic representation of the SE encoding. The vector is partitioned into four blocks, each representing a level of the EC hierarchy.

4.2.3 Separate Enzymes (SE)

The SE encoding focuses exclusively on enzymatic classification, discarding reaction information. Instead of treating the enzyme numbers as an atomic string, the system decomposes it into its four hierarchical levels and encodes each level independently with a separate block. For nodes with multiple enzymes (with pipe), the encoding is multi-hot, so all active levels across all enzymes are set to 1. The resulting one-hot vector has the structure shown in Figure 4.4.

Finally, the total vector size is equal to:

$$8\ (\text{L1}) + 23\ (\text{L2}) + 23\ (\text{L3}) + 365\ (\text{L4}) = 419\ \text{bits}. \tag{4.1}$$

The mapping from the values to one-hot indices is pre-computed and stored as space-separated integers, then loaded directly at training time without any further encoding step. This technique allows us to guarantee a consistent 419-dimensional encoding across all different dimensions of datasets. The non-digit entries are mapped to the value “0” as a placeholder for the error.

Enzymatic classification captures the biochemical function of a reaction at multiple levels of granularity. Two reactions may have entirely different identifiers and yet be catalyzed by enzymes with the same class, indicating that they perform the same type of chemistry. The SE allows the kernel to recognize this functional similarity. This decomposition into four levels is particularly suitable for kernel as the WL, which works by accumulating information from the neighbors. At iteration 0, nodes are distinguished by their full 419-bit representation. At iteration 1, the kernel aggregates neighborhoods, so if two nodes differ at L4 but share the other three levels, they will produce similar label distributions.

The multi-hot construction for multi-enzyme nodes reflects the biological reality that some reactions can be catalyzed by different enzymes. With this type of encoding, all enzymes present at a node contribute to the final vector.

4.2.4 Enzymes and Reactions (ER)

The ER encoding is a combined, powerful representation that integrates both reaction identity and enzymatic classification into a single vector. It is designed to capture information that JR and SE are not able to express alone. The resulting one-hot vector has the structure shown in Figure 4.5.

Finally, the total vector size is equal to:

$$\dim(\mathbf{v}_{ER}) = \underbrace{1}_{\text{rev bit}} + \underbrace{5163}_{R_{\text{base}}} + \underbrace{(8+23+23+365)}_{EC\ \text{levels}} = 5583\ \text{bits} \tag{4.2}$$

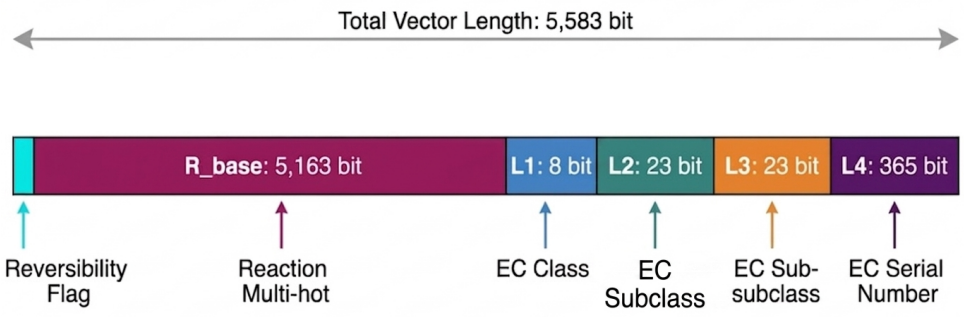


Figure 4.5: Schematic representation of the ER encoding. The vector is partitioned into six blocks, the first for the reverse reaction, the second one represent the numeric base of the reaction and the last four represent each a level of the EC hierarchy.

The 5163 different base reactions are obtained by removing the 721 "_rev" entries from the initial dataset of 5884 raw reaction strings. In the ER encoding, reversibility is represented as a single dedicated bit. Because ER vectors are very sparse, they are stored in a "compact sparse format", indeed each node is represented only by the indices of the bits set to 1 in the one-hot encoding, space-separated. It will be the data loader in a second moment to reconstruct the full dense tensor before saving the processed file. This technique significantly reduces disk usage.

Vector Representation Example

Node A: R04969 | R04970 (1.3.1.9 | 1.3.1.10) [forward]
Node B: R04969_rev | R04970_rev (1.3.1.9 | 1.3.1.10) [reverse]

Sparse Indices (Node A): $\{880, 881, 5164, 5174, 5195, 5226, 5227\}$

- Bit 0 absent: rev $= 0$ (indicates forward direction).
- Bits 880, 881: Map to $R_{\text{base}}[879]$ and $R_{\text{base}}[880]$, corresponding to reactions R04969 and R04970.
- Bit 5164: Corresponds to $L1[0]$ (EC class 1).
- Bit 5174: Corresponds to $L2[2]$ (EC subclass 3).
- Bit 5195: Corresponds to $L3[0]$ (EC sub-subclass 1).
- Bit 5226,5227: Corresponds to $L4[8]$ and $L4[9]$ (EC serial number 9 for enzyme 1.3.1.9 and EC serial number 10 for enzyme 1.3.1.10).

Sparse Indices (Node B): $\{880, 881, 5164, 5174, 5195, 5226, 5227\}$

- Bit 0 present: rev $= 1$ (indicates reverse direction).
- Remaining bits: Identical to Node A, representing the same base reactions and EC classification.

The two nodes are represented by vectors that differ by exactly one bit, giving the WL kernel a precise signal. The ER encoding is motivated by the observation that JR and SE capture complementary information. JR tells the kernel exactly which reaction is happening at a node,

but treats forward and reverse as equivalent and discards enzymatic context. SE instead tells the kernel what type of chemistry is performed and by which enzyme, but doesn't know the reaction identity and direction. From a dimensionality perspective, the combined 5583-bit vector of ER is less than the original 6273-bit one-hot vector over unique label strings (Baseline).

4.3 Experimental Setup and Configurations

All experiments were conducted following a unified evaluation protocol to ensure a fair and reproducible comparison across models, graph representations, and dataset variants. The evaluation framework adopts a 10-fold stratified cross-validation strategy, where the same dataset splits are shared across both models and all graph representations. After each fold, the best-performing model checkpoint, selected based on the validation loss, is evaluated on the held-out test fold. Performance is reported as the mean classification accuracy $\pm$ standard error across the 10 folds.

To manage the high computational complexity of evaluation, a multi-stage grid search strategy was adopted across the available dataset scales. First, an exhaustive hyperparameter exploration was conducted using the smaller balanced subset (`Sub_300`), benchmarking every possible parameter combination. Second, for the larger balanced subset (`Sub_600`), the search space was restricted to approximately half of the total configurations. Specifically, optimization focused on architectures with 3 layers, as preliminary runs on the smaller subset demonstrated that deeper configurations (5 layers) consistently led to lower performance while increasing the execution time. Finally, only the top-performing configurations identified from the `Sub_600` subset runs were evaluated on the full unbalanced KEGG dataset to assess performance scaling. A comprehensive overview of the hardware environments, hyperparameter search spaces, and validation parameters is summarize in Table 4.5.

GKNN Configuration

The GKNN was executed on the AMD EPYC infrastructure outlined in Table 4.5. The full hyperparameter grid explored across the benchmarks included:

- **Masks** (number of structural masks per GKC layer): $\{32, 64\}$;
- **Nodes** (size of each structural mask): $\{16, 24\}$;
- **Layers** (number of GKC layers): $\{3, 5\}$.

The learning rate was fixed at 0.001 for all experiments. Training was capped at 300 epochs, with early stopping triggered when the validation loss showed no improvement for 50 consecutive epochs. For the RGs representation, the GKNN was evaluated under eight kernel and node encoding combinations: WL (Baseline), Weisfeiler-Lehman Just-Reaction (WL-JR), Weisfeiler-Lehman Separate-Enzymes (WL-SE), Weisfeiler-Lehman Enzymes-Reaction (WL-ER), En-WL, Enhanced Weisfeiler-Lehman Enzymes-Reaction (En-WL-ER), ODD, and Ordered Decompositional DAG Enzymes-Reaction (ODD-ER).

This extensive exploration was required to address the high structural complexity of RGs. Conversely, for the AMNs and mDAGs abstractions, only the baseline WL and ODD kernels were evaluated, as both representations achieved a high level of accuracy already in their standard configurations, making further node relabeling expansions computationally redundant.

Configuration Feature	**GKNN**	**KA-GNN**
Hardware and Computational Cost		
Processing Unit	AMD EPYC 7413 CPU	NVIDIA Tesla T4 GPU / A40
Average Runtime	Up to $\sim$48 h	$\sim$10 min
Hyperparameter Search Space		
Masks	$\{32, 64\}$	$\{32, 64\}$
Nodes	$\{16, 24\}$	$\{16, 24\}$
Layers	$\{3, 5\}$	$\{3, 5\}$
Harmonics (K)	—	$\{1, 3\}$
Training and Optimization		
Learning Rate (η)	0.001 (Fixed)	0.001 (Fixed)
Maximum Epoch	300	300
Early Stopping (Patience)	50 epochs	50 epochs
Graph Kernels		
AMN / mDAGs	WL, ODD	—
RGs	WL, WL-JR, WL-SE, WL-ER, En-WL, En-WL-ER, ODD, ODD-ER	—
Evaluation Setup		
Validation Strategy	10-fold cross-validation	
Performance Metric	Mean accuracy $\pm$ standard error	

Table 4.5: Summary of experimental configurations, hyperparameters, and training protocols for the GKNN and KA-GNN frameworks.

Computational cost varied substantially across kernel configurations. Full grid searches using WL, WL-JR, and En-WL on the `RGs` representations required up to approximately 48 h per configuration, prompting the programmatic restriction to 3-layer structures on the larger `RGs_600` dataset variant. Conversely, WL-ER and WL-SE configurations exhibited significantly shorter runtimes (up to $\sim$24 h), as their more compact feature representations reduce the kernel computation overhead.

KA-GNN Configuration

The KA-GNN was executed on the GPU cluster node described in Table 4.5, with each experiment completing in approximately 10 min. The comprehensive hyperparameter grid was configured as follows:

- **Masks** (hidden dimension of the KAN layers): $\{32, 64\}$;
- **Nodes** (neighborhood embedding size): $\{16, 24\}$;
- **Layers** (number of message passing steps): $\{3, 5\}$;
- **Grid Feat (K)** (number of Fourier harmonics in the KAN activation function): $\{1, 3\}$.

The same training protocol was applied: a learning rate of 0.001, a maximum of 300 epochs, and early stopping with a patience of 50 epochs.

Computing environment. All experiments were run on the VHPC cluster of Ca' Foscari University of Venice, whose compute nodes run Ubuntu 24.04.4 LTS (kernel 6.8.0) on AMD EPYC 7413 processors. The kernel-based GKNN was executed on the CPU partition (`longjobs`): each grid-search job was allocated 4 CPU cores and up to 128 GB of RAM, with the heaviest reaction-graph configurations requiring up to about two days of wall-clock time. The KA-GNN was trained on the GPU partition (`gpus`), using a single GPU per run. The nodes of this partition are equipped with NVIDIA Tesla T4 (16 GB) and NVIDIA A40 (48 GB) GPUs. Because the GKNN ran on CPU cores and the KA-GNN on a GPU, the runtimes reported in Table 4.5 reflect the cost of each pipeline on its own hardware and are not a direct algorithmic speed comparison.

Software environment Two separate software stacks were used. The GKNN pipeline was frozen at Python 3.8.12, PyTorch 1.8.1, PyTorch Geometric 1.7.0, NumPy 1.23.5, SciPy 1.10.1 and Scikit-learn 1.2.1, to guarantee backward compatibility with the baseline executions [16]. The KA-GNN pipeline was run in a dedicated environment based on Python 3.10 with PyTorch 2.1.2 and CUDA 12.1.

Chapter 5

Results and Discussion

This chapter presents the experimental results and discussion of the proposed learning architectures on large-scale metabolic network classification. The core objective is to analyze how different graph abstraction paradigms interact with discrete (GKNN) and continuous (KA-GNN) mechanisms when mapping complex biochemical topologies to taxonomic clades. We evaluate our frameworks across three distinct structural representations (AMNs, mDAGs, and RGs) to investigate the trade-offs between data compression, model expressiveness, and structural regularization in biochemical domains. The analysis follows a step-by-step progression to isolate individual model components before performing broader comparisons (cross-model comparison and fulltest evaluation). The exploration begins with the discrete GKNN framework, establishing a clear baseline for how matching masks and color-refinement regularizations manage topological noise under changing network granularities. We then compare this approach with KA-GNN formulation, analyzing how replacing standard linear weights with learnable, univariate functional edges affects scalability across different mask capacities and grid resolutions (K).

With both architectural characterizations established, Section 5.3 provides a comparative synthesis that directly contrasts the two different frameworks. By contrasting color-refinement heuristics with the continuous activations, we identify when learnable parametric updates perform more effectively than fixed structural kernels. Finally, Section 5.4 evaluates the models on the full original KEGG dataset to test their generalization. This section provides an analysis of classification errors across different taxonomic groups and examines the learned mask weights to verify if the models focus on documented biological pathways.

5.1 GKNN Classification Results

This section presents the classification performance of the GKNN framework across the three metabolic graph representations. We analyze how core architectural hyperparameters and node relabeling strategies affect the mean classification accuracy on the `Sub_300` and `Sub_600` subsets. This evaluation highlights how the choice of network abstraction influences the model's ability to learn from metabolic topologies.

5.1.1 AMNs experiments

As detailed in Section 2.4, AMNs represent the highest level of functional abstraction, collapsing complete metabolic pathways into individual nodes. This structural compression yields compact graphs characterized by low vertex counts. The empirical results for the GKNN architecture

Masks	Nodes	Layers	Kernel	Sub_300	Sub_600
32	16	3	WL	98.40 ± 0.24	98.52 ± 0.31
			ODD	98.56 ± 0.44	$\mathbf{99.42 \pm 0.15}$
		5	WL	97.28 ± 0.59	—
			ODD	97.84 ± 0.27	—
32	24	3	WL	97.28 ± 0.54	98.94 ± 0.11
			ODD	98.24 ± 0.33	99.05 ± 0.33
		5	WL	93.84 ± 2.16	—
			ODD	97.44 ± 0.73	—
64	16	3	WL	98.72 ± 0.24	98.99 ± 0.41
			ODD	98.48 ± 0.37	98.94 ± 0.28
		5	WL	93.42 ± 1.87	—
			ODD	96.08 ± 1.10	—
64	24	3	WL	98.16 ± 0.53	99.21 ± 0.23
			ODD	98.16 ± 0.41	99.36 ± 0.17
		5	WL	91.91 ± 2.19	—
			ODD	95.12 ± 0.82	—

Table 5.1: GKNN classification accuracy (%) on Abstract Metabolic Networks (AMNs). Results report the mean accuracy $\pm$ standard error across 10 stratified folds.

utilizing the standard Weisfeiler-Lehman (WL) and Ordered Decompositional DAG (ODD) kernels are reported in Table 5.1. Across all tested configurations, the pathway label features encoded within AMNs prove to be highly discriminative for taxonomic classification, consistently achieving accuracies above 90 % on both data subsets. A clear trend emerges when looking at the model depth. For all parameter combinations, increasing the number of layers from 3 to 5 leads to a decrease in classification accuracy. For example, in the 64-mask and 24-node configuration with the WL kernel, performance drops from 98.16 ± 0.53 to 91.91 ± 2.19. This decrease suggests that a deeper neural classifier (5 layers) introduces unnecessary model complexity for this task. Since the features extracted from AMNs are already highly discriminative, simpler configurations with 3 layers are sufficient and less prone to overfitting. Consequently, the best results are obtained with 3 layers, where the ODD kernel achieves the highest accuracy of 99.42 ± 0.15 on the `Sub_600` subset.

5.1.2 mDAGs experiments

The mDAG representation introduces a more fragmented network structure. As established by the topological analysis in Section 4.1, the acyclic condensation process isolates specific functional blocks (MBBs), reducing the size of the main connected component and increasing the number of isolated vertices. Table 5.2 reports the performance of the GKNN model on mDAGs. This structural fragmentation leads to higher variance and lower peak accuracies compared to the results obtained on AMNs. Even though mDAGs include finer structural and reaction-level details, their disconnected nature makes it harder for the graph kernels to extract meaningful patterns, making taxonomic classification more difficult.

The results show an interesting interaction between the choice of the kernel and the hidden

Masks	Nodes	Layers	Kernel	Sub_300	Sub_600
32	16	3	WL	91.20 ± 1.27	96.61 ± 0.41
			ODD	87.44 ± 2.60	94.87 ± 0.74
		5	WL	79.44 ± 6.21	—
			ODD	87.03 ± 2.35	—
32	24	3	WL	84.13 ± 4.52	92.59 ± 2.50
			ODD	87.76 ± 3.16	94.39 ± 0.96
		5	WL	84.55 ± 3.33	—
			ODD	83.10 ± 3.52	—
64	16	3	WL	92.15 ± 1.09	$\mathbf{97.09 \pm 0.42}$
			ODD	86.75 ± 3.52	96.08 ± 0.61
		5	WL	78.30 ± 2.59	—
			ODD	74.39 ± 6.87	—
64	24	3	WL	83.44 ± 4.58	94.65 ± 0.67
			ODD	91.47 ± 2.81	94.97 ± 0.93
		5	WL	54.10 ± 7.02	—
			ODD	68.93 ± 5.39	—

Table 5.2: GKNN classification accuracy (%) on Metabolic Directed Acyclic Graphs (mDAGs). Results report the mean accuracy $\pm$ standard error across 10 stratified folds.

layer configuration. With a hidden layer size of 16 nodes, the WL kernel achieves higher performance in the 3-layer setups. Specifically, it reaches the highest accuracy of 97.09 ± 0.42 on the Sub_600 subset using 64 masks, outperforming ODD (96.08 ± 0.61). However, when the hidden layer capacity is increased to 24 nodes, ODD performs better in several configurations. For instance, on the Sub_300 subset with 64 masks, ODD scores 91.47 ± 2.81 compared to the 83.44 ± 4.58 of WL. This suggests that the features extracted by ODD benefit from a slightly larger classification network to be effectively separated.

Increasing the model depth to 5 layers degrades performance across all configurations. The largest drop occurs in the 64-mask and 24-node setting under the WL kernel, where accuracy falls from 83.44 ± 4.58 to 54.10 ± 7.02 on the Sub_300 subset. This degradation suggests that a deeper classifier may be more prone to overfitting on these fragmented topologies. In this case, the high number of isolated nodes potentially limits the availability of distinctive structural features, making it harder for a model to generalize well.

The ODD kernel does not consistently outperform WL at 5 layers; for example, WL maintains a small advantage in the 32-mask / 24-node configuration (84.55 % versus 83.10 %). However, ODD provides a more stable baseline under deep configurations. Its lowest accuracy (68.93 ± 5.39) is higher than the worst result of WL (54.10 ± 7.02). Furthermore, ODD mitigates the sharpest drops, showing an improvement of about 15 percentage points over WL in the 64-mask / 24-node setup. This stability may suggest that the directional decomposition of the ODD kernel matches the acyclic nature of mDAGs, extracting subgraphs that are more robust to overfitting when model complexity increases.

In summary, the WL kernel provides the highest classification accuracy when using a shallow 3-layer setup and a smaller nodes configuration. Conversely, the ODD kernel offers more stable performance as the size and depth of the classifier increase, making it less sensitive to the topological fragmentation of mDAGs.

5.1.3 RGs experiments

The RG representation provides the most granular topological mapping within the dataset, capturing fine-grained reaction and enzyme-level interactions. However, as explained in Section 3.3.1, this detailed formulation introduces structural redundancies, particularly the presence of two-node cycles caused by reversible biochemical pathways. To address these redundancies, we evaluated the GKNN architecture using different node relabeling strategies (Section 4.2) and a customized graph kernel (Section 3.3.2).

Effect of 3-layer propagation

Table 5.3 reports the classification accuracy for all kernel and encoding combinations under a constrained 3-layer propagation radius. Across these configurations, the ER encoding combined with WL or En-WL kernel emerges as the most consistent strong performer. It achieves the highest `Sub_300` accuracy of 91.75 ± 1.37 (WL-ER configuration, 64 masks, 16 nodes) and the highest `Sub_600` maximum of 94.55 ± 1.31 (En-WL-ER configuration, 32 masks, 24 nodes). This latter configuration is the most balanced of the two, pairing a high `Sub_300` score (91.52 ± 0.96) with the best `Sub_600` result.

These results support the choice of the ER encoding. By representing both the reaction identity and the enzymatic classification together, the model can exploit complementary information that neither JR nor SE capture individually. The reaction component identifies the specific biochemical transformation, while the four-level EC hierarchy encodes the class of the enzyme. This dual representation allows the model to leverage both specific and general functional similarities across the network.

The baseline WL encoding, which treats each raw label string as an atomic symbol with no internal decomposition, performs worse across all configurations, with accuracies ranging from 78.14 ± 4.40 to 83.99 ± 3.64 on `Sub_300`. This performance gap is likely due to the high dimensionality and sparsity of the baseline label space (6273 unique strings), which makes it harder for the model to generalize across similar reactions that differ only in direction or minor enzymatic variants.

The WL-JR encoding, which retains reaction identity while discarding enzyme information and collapses forward and reverse variants, yields mixed results. On some configurations with a larger mask, it improves over the baseline, reaching 90.55 ± 1.77 with 64 masks and 24 nodes on `Sub_300`. This suggests that direction-invariant reaction identity provides a useful signal. However, its inconsistency across different parameter configurations indicates that omitting the enzymatic context limits its performance when the reaction space alone cannot fully separate the complex metabolic profiles.

The WL-SE encoding, which contains exclusively the four-level EC hierarchy, achieves more stable results than WL-JR, with accuracies ranging between 78.54 ± 1.84 and 84.87 ± 2.31 on `Sub_300`. The lower variance reflects the more compact and structured nature of this feature space. By aggregating enzymatic information across all four EC levels, the encoding allows the model to recognize functional similarities between reactions that share an enzyme class or subclass, even when their specific identifiers differ. However, the complete omission of reaction identity introduces a performance ceiling.

Regarding the customized En-WL kernel, which implements the cycle-aware backtracking prevention described in Section 3.3.2, the results are encouraging but highly dependent on the hyperparameter regime. On `Sub_300`, En-WL performs comparably to or slightly above the standard WL. A clear improvement is observed within the 32-mask and 16-node profile, where En-WL reaches $92.91 \pm 2.65\%$ on `Sub_600` compared to $86.76 \pm 3.93\%$ for the standard

Masks	Nodes	Layers	Kernel	Sub_300	Sub_600
32	16	3	WL	78.16 ± 7.10	86.76 ± 3.93
			WL-JR	81.12 ± 6.76	89.68 ± 4.04
			WL-SE	81.36 ± 3.78	87.36 ± 2.84
			WL-ER	90.31 ± 1.01	93.70 ± 0.85
			En-WL	80.06 ± 4.41	92.91 ± 2.65
			En-WL-ER	90.79 ± 1.01	93.33 ± 0.93
			ODD	90.88 ± 2.75	80.79 ± 5.92
			ODD-ER	80.35 ± 5.98	81.69 ± 5.29
32	24	3	WL	83.99 ± 3.64	89.26 ± 4.46
			WL-JR	68.39 ± 7.79	83.70 ± 4.21
			WL-SE	81.42 ± 2.57	87.61 ± 2.40
			WL-ER	91.52 ± 0.79	94.39 ± 1.21
			En-WL	83.03 ± 5.63	77.56 ± 4.65
			En-WL-ER	91.52 ± 0.96	$\mathbf{94.55 \pm 1.31}$
			ODD	71.58 ± 7.19	93.38 ± 2.75
			ODD-ER	—	—
64	16	3	WL	82.55 ± 4.31	87.62 ± 4.20
			WL-JR	82.04 ± 5.49	88.99 ± 4.23
			WL-SE	84.87 ± 2.31	84.60 ± 1.84
			WL-ER	$\mathbf{91.75 \pm 1.37}$	85.59 ± 2.76
			En-WL	84.56 ± 6.37	78.41 ± 6.70
			En-WL-ER	87.44 ± 2.50	87.99 ± 2.83
			ODD	86.13 ± 5.10	84.86 ± 4.05
			ODD-ER	88.79 ± 2.23	91.04 ± 4.07
64	24	3	WL	78.14 ± 4.40	89.13 ± 5.05
			WL-JR	90.55 ± 1.77	82.91 ± 6.96
			WL-SE	78.54 ± 1.84	82.06 ± 2.50
			WL-ER	86.80 ± 3.31	93.43 ± 1.16
			En-WL	85.74 ± 5.51	90.69 ± 4.79
			En-WL-ER	88.64 ± 2.25	87.07 ± 3.15
			ODD	72.09 ± 8.04	86.35 ± 4.31
			ODD-ER	—	—

Table 5.3: GKNN classification accuracy (%) on Reaction Graphs (RGs) under a constrained 3-layer propagation radius. Results report the mean accuracy $\pm$ standard error across 10 stratified folds.

WL. This supports the hypothesis that the "ping-pong" effect induced by two-node cycles does degrade the discriminative quality of the learned representations, and that the dynamic edge-pruning strategy effectively mitigates this issue. Crucially, the full potential emerges evaluating the En-WL in combination with the ER encoding. The model achieves its absolute peak performance on the `Sub_600` dataset, slightly outperforming all other configurations. This results shows us that combining chemical node refinements with cycle-aware graph kernels is beneficial.

The ODD kernel achieves competitive results on `Sub_300`, reaching 90.88 ± 2.75 with 32 masks and 16 nodes. However, its performance on `Sub_600` is less consistent, dropping to $80.79 \pm 5.92\%$ under the same configuration. This instability may reflect a difficulty in generalizing to the higher topological variation of the larger subset. When combined with advanced encodings like ER, the ODD results partially stabilize, though the model remains constrained by the loss of local cyclic contexts during the graph condensation phase.

Effect of 5-layer configuration

Table 5.4 shows a substantial decrease in performance when the propagation depth is increased to 5 layers. Across most kernel and encoding combinations, the classification accuracy drops significantly. For instance, the baseline WL falls to 39.65 ± 4.89 under the 16-node budget and reaches 30.67 ± 3.71 in the 32-mask, 24-node configuration. These values approach the majority-class baseline for this six-class task. A similar drop is observed for the En-WL kernel (36.61 ± 4.39). This sharp decrease in accuracy indicates that a deeper 5-layer classifier introduces excessive model complexity for this task, leading to severe overfitting. Because the feature space derived from RGs is large and sparse, increasing the depth of the neural classifier causes the model to fit to training noise rather than generalizable taxonomic patterns, uniforming the final predictions.

The choice of node encoding strategy significantly influences how well the model resists this degradation. The hybrid WL-ER scheme consistently emerges as the top-performing configuration under this 5-layer setup, sustaining an accuracy of 75.59 ± 4.08 in the 16-node regime and achieving a maximum of 83.74 ± 3.79 with 24 nodes. A moderate robustness is also maintained by the separate enzyme encoding (WL-SE), which preserves stable accuracies between 53.09 ± 4.30 and 71.17 ± 4.00 on `Sub_300`. The relative resilience of WL-SE and WL-ER suggests that incorporating the structured four-level EC classification helps constrain the feature space. By grouping nodes into distinct functional classes rather than relying on raw reaction strings, these encodings provide more structured and stable features that mitigate the overfitting induced by the deeper classifier architecture.

The behavior of the ODD kernel under a 5-layer configuration varies based on the hyperparameter setup. In the 32-mask regimes, ODD shows a severe performance drop, falling below 40% (e.g., scoring 29.54 ± 3.37 at 24 nodes). However, when the configuration is expanded to 64 masks and 24 nodes, ODD partially recovers to an accuracy of 54.17 ± 10.03, outperforming both the standard WL baseline (51.34 ± 7.13) and the WL-JR variant (40.10 ± 5.90). This partial recovery suggests that a larger classification capacity allows the model to better exploit the directional features extracted by the ODD decomposition, helping to isolate useful signals even within an unfavorable hyperparameter regime.

In summary, the results reported in Table 5.4 confirm that a shallow 3-layer architecture is the optimal choice for this framework. While the 3-layer setup effectively learns from the metabolic features, deeper configurations suffer from severe overfitting, making specialized encodings like WL-ER necessary to preserve a discriminative signal.

Masks	Nodes	Layers	Kernel	Sub_300	Sub_600
32	16	5	WL	39.65 ± 4.89	—
			WL-JR	44.37 ± 4.42	—
			WL-SE	67.67 ± 7.88	61.30 ± 4.25
			WL-ER	75.59 ± 4.08	—
			En-WL	38.84 ± 4.48	—
			En-WL-ER	—	—
			ODD	37.00 ± 5.29	—
			ODD-ER	—	—
32	24	5	WL	30.67 ± 3.71	—
			WL-JR	31.39 ± 3.43	—
			WL-SE	71.17 ± 4.00	66.13 ± 6.54
			WL-ER	83.74 ± 3.79	—
			En-WL	36.61 ± 4.39	—
			En-WL-ER	—	—
			ODD	29.54 ± 3.37	—
			ODD-ER	—	—
64	16	5	WL	54.04 ± 8.52	—
			WL-JR	43.30 ± 8.65	—
			WL-SE	53.09 ± 4.30	57.59 ± 5.66
			WL-ER	54.79 ± 8.38	—
			En-WL	45.09 ± 8.21	—
			En-WL-ER	—	—
			ODD	38.58 ± 8.22	—
			ODD-ER	—	—
64	24	5	WL	51.34 ± 7.13	—
			WL-JR	40.10 ± 5.90	—
			WL-SE	60.85 ± 2.28	56.44 ± 5.08
			WL-ER	59.06 ± 8.56	—
			En-WL	45.15 ± 8.15	—
			En-WL-ER	—	—
			ODD	54.17 ± 10.03	—
			ODD-ER	—	—

Table 5.4: GKNN classification accuracy (%) on Reaction Graphs (RGs) under a constrained 5-layer propagation radius. Results report the mean accuracy $\pm$ standard error across 10 stratified folds.

Masks	Nodes	Grid Feat	Layers	Sub_300 (%)	Sub_600 (%)
32	16	1	3	98.24 ± 0.46	99.10 ± 0.21
			5	95.84 ± 0.33	97.19 ± 0.40
		3	3	97.28 ± 0.30	98.31 ± 0.44
			5	95.28 ± 0.50	97.14 ± 0.39
32	24	1	3	98.64 ± 0.36	98.99 ± 0.25
			5	94.08 ± 0.48	96.40 ± 0.48
		3	3	97.84 ± 0.46	98.78 ± 0.27
			5	94.96 ± 0.40	97.25 ± 0.39
64	16	1	3	$\mathbf{98.88 \pm 0.34}$	$\mathbf{99.15 \pm 0.20}$
			5	96.56 ± 0.53	97.78 ± 0.32
		3	3	98.48 ± 0.33	98.94 ± 0.27
			5	97.04 ± 0.34	97.83 ± 0.37
64	24	1	3	98.56 ± 0.31	98.99 ± 0.32
			5	96.24 ± 0.32	97.46 ± 0.50
		3	3	98.40 ± 0.29	99.15 ± 0.26
			5	97.20 ± 0.38	98.25 ± 0.27

Table 5.5: KA-GNN classification accuracy (%) on Abstract Metabolic Networks (AMNs). The Grid Feat column reports the number of Fourier harmonics K. Results report the mean accuracy $\pm$ standard error across 10 stratified folds.

5.2 KA-GNN Classification Results

This section presents the classification results of the Kolmogorov-Arnold Graph Neural Network (KA-GNN) architecture. By replacing traditional Multi-Layer Perceptron (MLP) layers with learnable, univariate Fourier-based activation functions positioned directly on the edges of the neural layers, the KA-GNN provides non-linear, adaptive aggregation. The model is evaluated across all three metabolic network representations (AMNs, mDAGs and RGs) under varying configurations of masks, hidden nodes and layer depths. In addition to the standard hyperparameters, the evaluation introduces the harmonic resolution parameter K, reported in the `Grid Feat` column of the result tables. This setting controls the number of Fourier series harmonics used to determine the frequency resolution and expressive capacity of the learnable edge-activation functions. Overall, the KA-GNN architecture shows strong performance across the tasks, demonstrating higher stability than the baseline models when the network depth increases.

5.2.1 AMNs experiments

The empirical results reported in Table 5.5 evaluate the performance of the Fourier-based KA-GNN architecture on metabolic network representations. The classification task on AMNs shows stable performance, with all tested configurations consistently yielding mean accuracies above 94%. This high performance suggests that the high-level functional wiring captured by the abstract representation preserves clear taxonomic signals that are easily learned by the model. An analysis of the harmonic configuration shows a similar performance distribution between $K = 1$ and $K = 3$. Under a shallow 3-layer setup, a minimal harmonic resolution ($K = 1$) is sufficient

Masks	Nodes	Grid Feat	Layers	Sub_300 (%)	Sub_600 (%)
32	16	1	3	86.47 ± 1.31	92.01 ± 0.66
			5	71.58 ± 1.75	78.14 ± 0.81
		3	3	84.71 ± 0.78	91.16 ± 0.82
			5	79.26 ± 1.04	83.17 ± 0.92
32	24	1	3	90.47 ± 0.90	96.40 ± 0.27
			5	78.47 ± 1.68	89.73 ± 0.60
		3	3	88.55 ± 0.43	93.70 ± 0.50
			5	78.39 ± 1.30	84.70 ± 0.70
64	16	1	3	91.91 ± 0.54	96.61 ± 0.35
			5	85.35 ± 0.92	92.54 ± 0.58
		3	3	90.31 ± 0.75	95.55 ± 0.43
			5	84.71 ± 1.01	89.89 ± 0.43
64	24	1	3	93.20 ± 0.60	$\mathbf{97.04 \pm 0.29}$
			5	86.87 ± 0.88	94.87 ± 0.43
		3	3	$\mathbf{93.43 \pm 0.82}$	96.88 ± 0.53
			5	85.51 ± 0.94	91.06 ± 0.54

Table 5.6: KA-GNN classification accuracy (%) on Metabolic Directed Acyclic Graphs (mDAGs). The Grid Feat column reports the number of Fourier harmonics K. Results report the mean accuracy $\pm$ standard error across 10 stratified folds.

to reach nearly optimal classification boundaries. For instance, the global maximum on the Sub_300 subset is achieved with 64 masks and 16 nodes, peaking at 98.88 ± 0.34, while the same configuration reaches 99.15 ± 0.20 on the Sub_600 subset. Increasing the harmonic resolution to $K = 3$ does not lead to generalized performance shifts, but it introduces localized adjustments. For example, within the deepest parameter profile (64 masks, 24 nodes, 5 layers), increasing the number of harmonics to $K = 3$ mitigates the performance drop, raising the accuracy from $96.24 \pm 0.32\%$ to $97.20 \pm 0.38\%$. This indicates that while a single low-order Fourier term is sufficient for shallow architectures, additional harmonics can provide a localized regularization benefit when network depth increases. Furthermore, the KA-GNN architecture demonstrates higher stability against depth-induced degradation compared to the GKNN baselines, avoiding severe performance drops when moving from 3 to 5 layers.

5.2.2 mDAGs experiments

Table 5.6 reports the classification performance of the KA-GNN architecture on Metabolic Directed Acyclic Graphs (mDAGs). The experimental results show high classification accuracy across both subsets, with performance peaking at 93.43 ± 0.82 on Sub_300 and reaching a global maximum of 97.04 ± 0.29 on Sub_600 under the 64-mask, 24-node configuration. These results indicate that contracting cyclic metabolic pathways into condensed nodes, while preserving the overall directional flow of the network, provides a discriminative topological signal for taxonomic classification.

An analysis of the harmonic resolution shows an interaction between K and the model depth. Unlike the stable distribution observed in the AMNs experiments, increasing the harmonic capacity from $K = 1$ to $K = 3$ yields visible effects under a 5-layer configuration. In the

smallest profile (32 masks, 16 nodes), a minimal harmonic resolution ($K = 1$) experiences a more pronounced performance drop at 5 layers, falling to 71.58 ± 1.75 on `Sub_300` and 78.14 ± 0.81 on `Sub_600`. Introducing more Fourier terms ($K = 3$) helps mitigate this drop, raising performance to 79.26 ± 1.04 and 83.17 ± 0.92, respectively.

However, as the hidden layer size and mask count increase, this trend reverses, and the lower harmonic resolution ($K = 1$) systematically outperforms the $K = 3$ setup. For instance, under the 64-mask and 24-node constraint at 5 layers, shifting to $K = 3$ decreases accuracy from 94.87 ± 0.43 to 91.06 ± 0.54 on the larger subset. This behavior suggests an over-parameterization threshold: when the model already has sufficient capacity via masks and nodes, a simpler edge activation ($K = 1$) provides a better regularizing effect, preventing the Fourier terms from overfitting.

Finally, the data shows that the model benefits from a larger mask and node allocation to stabilize its predictions on the larger subset. When moving from 32 to 64 masks under a 5-layer setup, the standard error decreases, and the mean accuracy margins expand by up to approximately 14 percentage points. This behavior suggests that the higher structural heterogeneity of the condensed mDAG topologies in `Sub_600` requires a larger number of structural masks to be effectively captured.

5.2.3 RGs experiments

Table 5.7 reports the classification performance of the KA-GNN architecture on RGs. Operating under a shallow 3-layer propagation limit, the model achieves high peak accuracies, reaching 98.16 ± 0.29 on `Sub_300` (with 64 masks and 16 nodes) and a global maximum of 98.89 ± 0.25 on `Sub_600` (under 32 masks and 24 nodes). These results demonstrate that when the classifier depth is restricted, the non-linear transformations computed by the Fourier-based edges can effectively exploit the granular, reaction-level and enzyme-level interactions captured by the representation.

However, when expanding the classifier depth to 5 layers, the data reveals an inversion of the trends observed in the previous graph abstractions. In this domain, increasing K from 1 to 3 does not mitigate performance drops, but instead accelerates depth-induced degradation. This phenomenon is most evident in the 32-mask, 16-node configuration, where shifting K from 1 to 3 causes the 5-layer accuracy to collapse from 77.11 ± 1.44 down to $62.93 \pm 1.83\%$ on `Sub_300`. A similar decrease is visible on `Sub_600`, where the same structural configuration drops from 87.46 ± 0.74 to 72.16 ± 1.12. This performance drop probably indicates severe overfitting as model complexity increases. At 5 layers, the total number of learnable parameters in the network scales significantly compared to the 3-layer setup. Using a minimal harmonic setting ($K = 1$) limits the capacity of the edge activations, which helps mitigate overfitting. Conversely, expanding the parameterization to $K = 3$ increases the degrees of freedom, causing the Fourier functions to fit to local noise within the dense RG topologies. This degradation can be partially mitigated by increasing the hidden layer size. When expanding the budget to 64 masks, the larger hidden dimensionality helps the network find a more stable optimization path. Indeed, under 64 masks and 16 nodes, moving to $K = 3$ yields an accuracy of 80.15 ± 1.40 on `Sub_300`, recovering from the collapse observed in the 32-mask setup. In summary, while the KA-GNN architecture achieves near-optimal results in shallow configurations, scaling it to 5 layers on dense graphs requires strict parameter control, favoring a lower number of harmonics ($K = 1$) to prevent severe overfitting.

5.3 Cross-model Comparison

It is now necessary to provide a unified comparative analysis of the two architectural paradigms evaluated in this work: the kernel-driven Graph Kernel Neural Network (GKNN), which relies on discrete structural matching, and the Kolmogorov-Arnold Graph Neural Network (KA-GNN), which uses learnable Fourier-based edge transformation functions. By analyzing their performance across different metabolic formulations, network depths, and dataset granularities, we identify the specific topological conditions under which each framework excels. Finally, both frameworks are evaluated against the continuous, end-to-end DGCNN baselines previously established on this taxonomic classification benchmark [16].

5.3.1 Topological abstractions and architectural trade-offs

The empirical results across the three metabolic formulations (AMNs, mDAGs and RGs) demonstrate that the selection of the optimal paradigm depends on the level of abstraction applied to the underlying biochemical pathways. On AMNs, where the metabolic graph is compressed into high-level functional paths, both paradigms achieve high accuracies. The GKNN combined with the ODD kernel reaches an accuracy of $99.42 \pm 0.15\%$ on the `Sub_600` dataset, while the KA-GNN peaks at $99.15 \pm 0.20\%$. This minimal performance gap indicates that when graph preprocessing abstracts away explicit reaction labels and local redundancies, the taxonomic signal becomes readily accessible. Consequently, the high expressive capacity of learnable edge transformations provides no significant advantage over discrete Weisfeiler-Lehman (WL) refinement histograms within simplified structural environments.

Conversely, a clear architectural divergence occurs in the uncompressed domain of RGs. On these networks, the standard WL baseline under the GKNN framework struggles with the raw feature space, requiring manual node regularizers like WL-ER to reach an accuracy of 91.75 ± 1.37 on `Sub_300` and En-WL-ER to obtain 94.55 ± 1.31 on `Sub_600`. In contrast, the KA-GNN achieves a peak of 98.16 ± 0.29 on `Sub_300` and 98.89 ± 0.25 on `Sub_600` using the raw, unrefined label space. This confirms that learnable Fourier-based transformations are significantly more effective at capturing patterns within complex graphs than discrete structural kernels, which can be confounded by high neighborhood density and shared metabolites.

5.3.2 Expressive Capacity and Overfitting

The interaction between model depth (3 vs. 5 layers) and internal parameter capacity highlights a critical trade-off in both paradigms. Within the GKNN framework, expanding the classifier depth to 5 layers triggers a severe performance drop. As proposed in Section 5.1.3, this vulnerability is probably caused by severe overfitting. To maintain taxonomic discrimination at higher depths, this framework requires external constraints such as restricting the node label space or condensing the networks into cycle-free topologies (mDAG).

Conversely, the KA-GNN framework exhibits higher resilience to depth-induced performance degradation. Because the Fourier-based transformations are localized on the edges, the network can map distinct non-linear transformations to individual connections, preventing features from homogenizing too rapidly. For instance, on RGs configured at 5 layers, the KA-GNN sustains an accuracy of 85.75 ± 0.90 on the `Sub_300` dataset (64 masks, 16 nodes, $K = 1$), whereas the corresponding GKNN WL baseline falls to 54.04 ± 8.52.

However, this structural resilience depends heavily on the chosen harmonic resolution K. Increasing the number of harmonics to $K = 3$ improves architectural stability only within the

highly constrained regimes of condensed mDAGs (32 masks, 16 nodes), lifting the 5-layer accuracy from 71.58 ± 1.75 to 79.26 ± 1.04 on `Sub_300`. On uncompressed RGs, the $K = 3$ configuration induces severe overfitting: at 32 masks and 16 nodes, the 5-layer accuracy decreases from 77.11 ± 1.44 ($K = 1$) to 62.93 ± 1.83.

5.3.3 Comparison with Baselines frameworks

To evaluate the broader impact of the proposed GKNN and KA-GNN frameworks, Table 5.8 compares our peak configurations against both the continuous end-to-end DGCNN architecture and the standard GKNN baselines established on this exact taxonomic benchmark [16]. One architectural caveat applies to this comparison: the baseline GKNN from prior work was evaluated exclusively with the standard WL kernel and over a restricted hyperparameter grid (masks $\in \{8, 16, 32\}$, nodes $\in \{6, 12\}$), whereas our search space explores cycle-aware kernels and alternative node encodings over an expanded capacity range (masks $\in \{32, 64\}$, nodes $\in \{16, 24\}$).

The continuous DGCNN architecture proves competitive only on the heavily abstracted AMNs (95.92 ± 0.42), where the pathway-level representation exposes a highly compressed structural signal. Its performance on the finer granularities, however, is less stable, falling to 70.60 ± 7.66 on mDAGs (its weakest representation) and recovering only partially to 82.74 ± 6.48 on RGs. The larger standard errors observed on both abstractions (± 7.66 and ± 6.48) confirm that the end-to-end continuous convolution struggles to generalize effectively across dense metabolic graph topologies.

The GKNN baseline represents a stronger reference point, reaching 98.41 on AMNs, 94.55 on mDAGs, and 84.81 on RGs. The discrete kernel successfully isolates localized structural contexts, as evidenced by standard errors dropping under 1% in two out of three representations. Its primary limitation manifests in the uncompressed RG domain, where the unrefined WL process is probably confounded by the reaction-level redundancy of the raw label space.

Both of our proposed frameworks achieve systematic improvements over these baselines across all three representations, with the performance margin widening precisely where prior methods exhibit degradation. Our GKNN setup, equipped with the cycle-aware En-WL kernel and the node relabelings, lifts the RG accuracy from 84.81 to 94.55, which represents a gain of nearly 10 percentage points over the baseline GKNN and almost 12 percentage points over DGCNN. Concurrently, it improves performance on AMNs ($+1.01\%$) and mDAGs ($+2.54\%$).

Furthermore, our KA-GNN pushes the RG results even higher to 98.89 ± 0.25, exceeding the baseline GKNN by approximately 14 percentage points and the DGCNN baseline by over 16 percentage points, all while operating directly on the raw label space without relying on hand-crafted node relabeling heuristics. Finally, the choice between the two proposed frameworks depends on the network representation. The GKNN shows a marginal advantage on highly compressed abstractions, performing best on AMNs (99.42% versus 99.15%) and resulting in a comparable accuracy on mDAGs (97.09% versus 97.04%). Conversely, the KA-GNN yields better results on uncompressed RGs (98.89% versus 94.55%). As evidenced by these empirical trends, when the graph topology is dense and retains reaction-level details, learnable edge activations offer a more effective alternative to discrete structural matching mechanisms.

Masks	Nodes	Grid Feat	Layers	Sub_300 (%)	Sub_600 (%)
32	16	1	3	97.28 ± 0.30	98.52 ± 0.33
			5	77.11 ± 1.44	87.46 ± 0.74
		3	3	77.18 ± 1.33	92.96 ± 0.80
			5	62.93 ± 1.83	72.16 ± 1.12
32	24	1	3	97.28 ± 0.42	$\mathbf{98.89 \pm 0.25}$
			5	75.18 ± 1.00	88.41 ± 0.86
		3	3	77.66 ± 0.70	87.35 ± 0.76
			5	65.49 ± 0.80	78.56 ± 0.89
64	16	1	3	$\mathbf{98.16 \pm 0.29}$	98.73 ± 0.18
			5	85.75 ± 0.90	91.69 ± 0.70
		3	3	83.03 ± 0.92	91.11 ± 0.64
			5	80.15 ± 1.40	84.38 ± 0.47
64	24	1	3	97.60 ± 0.36	98.78 ± 0.25
			5	84.71 ± 0.98	90.68 ± 0.55
		3	3	79.91 ± 1.07	91.27 ± 0.74
			5	79.19 ± 0.86	84.81 ± 0.77

Table 5.7: KA-GNN classification accuracy (%) on Reaction Graphs (RGs). The Grid Feat column reports the number of Fourier harmonics K. Results report the mean accuracy $\pm$ standard error across 10 stratified folds.

Model Paradigm	AMNs (%)	mDAGs (%)	RGs (%)
DGCNN [16]	95.92 ± 0.42	70.60 ± 7.66	82.74 ± 6.48
GKNN [16]	98.41 ± 0.33	94.55 ± 0.78	84.81 ± 3.53
GKNN	$\mathbf{99.42 \pm 0.15}$	$\mathbf{97.09 \pm 0.42}$	94.55 ± 1.31
KA-GNN	99.15 ± 0.20	97.04 ± 0.29	$\mathbf{98.89 \pm 0.25}$

Table 5.8: Taxonomic classification accuracy on the Sub_600 datasets best performance, contrasting the baseline architectures of [16] with our proposed frameworks. All values are reported as mean accuracy $\pm$ standard error.

5.4 Generalization and Interpretability

Evaluating the performance of the proposed graph-based architectures requires verifying their generalization capabilities under different dataset scales and analyzing the biochemical relevance of their internal representations. In large-scale metabolic modeling, models can be susceptible to performance degradation due to topological noise or overfitting to dataset-specific patterns. To demonstrate how the proposed paradigms address these aspects, this section evaluates the GKNN and KA-GNN frameworks across three dimensions:

- **Robustness to Dataset Scale and Imbalance:** We assess the scaling properties of both architectures by testing their configurations on the original KEGG repository (*Full Test* with 8904 organisms), which features a larger scale and high class imbalance. This includes a per-kingdom analysis on the RGs domain, comparing all combinations of node relabeling strategies and graph kernels.
- **Prediction analysis:** Through an analysis of class-resolved confusion matrices on the RG domain, we map the error distributions across specific taxonomic lineages, highlighting where the architectures resolve classification ambiguities.
- **Mask Interpretability:** We analyze the decision-making process of the matching masks by evaluating their prototype weights and loss-increment profiles (ΔLoss). This analysis shows whether the optimization process assigns higher importance to functionally cohesive metabolic subsystems.

By combining empirical robustness testing with an analysis of the mask weights, these investigations verify that the models capture meaningful topological patterns that correlate with taxonomic classifications rather than relying on localized memorization.

5.4.1 Robustness to Dataset Scale and Imbalance

To assess the generalization capabilities of the proposed frameworks, we test their performance by transitioning from the class-balanced subsets of `Sub_300` and `Sub_600` to the original KEGG dataset (*Full Test*). The comparative performance for the GKNN is documented in Tables 5.9 and 5.10, while the KA-GNN is proposed in Tables 5.11 and 5.12.

The empirical evaluation reveals an upward shift in mean classification accuracy across most representations when moving from the balanced subsets to the full testing environment. This behavior is directly linked to the composition of the complete KEGG repository. In the KEGG database, taxonomic kingdoms such as *Plants*, *Fungi*, and *Protists* contain fewer than 600 available organisms in total. Consequently, when the models are trained on the `Sub_600` subset, they are exposed to the near-totality of these domains during the cross-validation process. Consequently, the transition to the *Full Test* does not introduce new data variations for these kingdoms. The scaling challenge is almost entirely concentrated on the *Bacteria* lineage, which contains thousands of organisms. Because the models perform well on this expanded bacterial majority, the overall raw accuracy on the *Full Test* is mathematically driven up by this single class.

Evaluating the models trained on the smaller `Sub_300` subset (Table 5.9) provides a clearer test of generalization. In this setup, other kingdoms (specifically *Animals* and *Archaea*) exceed the 300 training sample limit, forcing the models to classify entirely unseen organisms across multiple lineages. On abstracted domains like AMNs and mDAGs, the framework still achieves

Graph repr.	Mechanism	Optimal Config (h,n,l)	Sub_300 Acc. (%)	Full Test Acc. (%)
AMNs	WL	$(64,16,3)$	98.72 ± 0.24	98.81 ± 0.11
	ODD	$(32,16,3)$	98.56 ± 0.44	98.66 ± 0.28
mDAGs	WL	$(64,16,3)$	92.15 ± 1.09	93.32 ± 1.10
	ODD	$(64,24,3)$	91.47 ± 2.81	87.65 ± 6.15
RGs	**WL**			
	$\rightarrow$ WL (Base)	$(32,24,3)$	83.99 ± 3.64	80.48 ± 7.69
	$\rightarrow$ WL-SE	$(64,16,3)$	84.87 ± 2.31	81.01 ± 3.77
	$\rightarrow$ WL-JR	$(64,24,3)$	90.55 ± 1.77	87.48 ± 3.93
	$\rightarrow$ WL-ER	$(64,16,3)$	91.75 ± 1.37	90.45 ± 5.11
	En-WL			
	$\rightarrow$ En-WL (Base)	$(64,24,3)$	85.74 ± 5.51	82.53 ± 8.19
	$\rightarrow$ En-WL-ER	$(32,24,3)$	91.52 ± 0.96	93.80 ± 1.53
	ODD			
	$\rightarrow$ ODD (Base)	$(32,16,3)$	90.88 ± 2.75	93.31 ± 1.23
	$\rightarrow$ ODD-ER	$(64,16,3)$	88.79 ± 2.23	92.38 ± 1.67

Table 5.9: Taxonomic classification accuracy (%$\pm$ standard error) of the GKNN framework, contrasting the balanced Sub_300 subset against the complete *Full Test* environment across different structural kernel variants and graph abstractions.

stable performance. For example, as shown in Table 5.10, the GKNN with the WL kernel on mDAGs goes from $97.09 \pm 0.42\%$ on Sub_600 to $98.15 \pm 0.31\%$ on the *Full Test*. However, the limitations of this scaling appear when analyzing the ODD kernel on the mDAGs representation within the Sub_300 framework (Table 5.9). In this case, the accuracy drops from $91.47 \pm 2.81\%$ to $87.65 \pm 6.15\%$, showing that the model is more sensitive to scale shifts when trained on fewer data samples.

In the uncompressed Reaction Graphs (RGs) domain, where the network structure is preserved, explicit architectural trends emerge between the structural refinement methodologies. Within the WL family on Sub_600, the baseline kernel accuracy increases from 89.26% to 96.16%, with a standard error of $\pm 1.52\%$. Incorporating node-label regularizations helps control this variance: the enzyme-reaction variant (WL-ER) achieves $96.80 \pm 3.22\%$ on the full environment, while its extension En-WL-ER yields $96.90 \pm 0.97\%$. This indicates that specific node encodings help the discrete color-refinement heuristics maintain consistent classification boundaries under scaling.

Conversely, the ODD family responds differently to node-label modifications. The baseline ODD decomposition, which handles cyclic patterns through strongly connected component condensation, generalizes from 93.38% on Sub_600 to $96.01 \pm 0.77\%$ on the *Full Test*, showing a stability level comparable to the WL baseline. However, augmenting ODD with the ER encoding (ODD-ER) reduces the mean accuracy to $92.39 \pm 2.57\%$. A similar decrease is present in the Sub_300 configuration (Table 5.9), where the baseline ODD reaches $93.31 \pm 1.23\%$, whereas ODD-ER drops to $92.38 \pm 1.67\%$.

The empirical evaluation of the KA-GNN framework demonstrates a high capacity to

Graph repr.	Mechanism	Optimal Config (h,n,l)	Sub_600 Acc. (%)	Full Test Acc. (%)
AMNs	WL	$(64,16,3)$	98.99 ± 0.41	99.41 ± 0.15
	ODD	$(32,16,3)$	99.42 ± 0.15	99.35 ± 0.15
mDAGs	WL	$(64,16,3)$	97.09 ± 0.42	98.15 ± 0.31
	ODD	$(64,16,3)$	96.08 ± 0.61	97.05 ± 0.58
RGs	**WL**			
	$\rightarrow$ WL (Base)	$(32,24,3)$	89.26 ± 4.46	96.16 ± 1.52
	$\rightarrow$ WL-SE	$(32,24,3)$	87.61 ± 2.40	92.04 ± 1.59
	$\rightarrow$ WL-JR	$(32,16,3)$	89.68 ± 4.04	95.20 ± 1.46
	$\rightarrow$ WL-ER	$(32,24,3)$	94.39 ± 1.21	96.80 ± 3.22
	En-WL			
	$\rightarrow$ En-WL (Base)	$(32,16,3)$	92.91 ± 2.65	94.67 ± 2.24
	$\rightarrow$ En-WL-ER	$(32,24,3)$	94.55 ± 1.31	96.90 ± 0.97
	ODD			
	$\rightarrow$ ODD (Base)	$(32,24,3)$	93.38 ± 2.75	96.01 ± 0.77
	$\rightarrow$ ODD-ER	$(64,16,3)$	91.04 ± 4.07	92.39 ± 2.57

Table 5.10: Taxonomic classification accuracy (%$\pm$ standard error) of the GKNN framework, contrasting the balanced Sub_600 subset against the complete *Full Test* environment across different structural kernel variants and graph abstractions.

Graph repr.	Optimal Config (h,n,l,K)	Sub_300 Acc. (%)	Full Test Acc. (%)
AMNs	$(64,16,3,1)$	98.88 ± 0.34	98.98 ± 0.17
	$(64,16,5,1)$	96.56 ± 0.53	96.67 ± 0.62
mDAGs	$(64,24,3,3)$	93.43 ± 0.82	92.72 ± 1.16
	$(64,24,5,3)$	85.51 ± 0.94	77.06 ± 1.13
RGs	$(32,24,3,1)$	97.28 ± 0.42	99.01 ± 0.14
	$(32,24,5,1)$	75.18 ± 1.00	72.24 ± 2.23

Table 5.11: Taxonomic classification accuracy (%$\pm$ standard error) of the KA-GNN framework, contrasting the balanced Sub_300 reference against the complete *Full Test* distribution across varying network depths (l) and harmonic grid dimensions (K).

Graph repr.	Optimal Config (h,n,l,K)	Sub_600 Acc. (%)	Full Test Acc. (%)
AMNs	$(64,24,3,3)$	99.15 ± 0.26	99.33 ± 0.11
	$(64,24,5,3)$	98.25 ± 0.27	98.91 ± 0.18
mDAGs	$(64,24,3,1)$	97.04 ± 0.29	99.13 ± 0.08
	$(64,24,5,1)$	94.87 ± 0.43	98.25 ± 0.17
RGs	$(32,24,3,1)$	98.89 ± 0.25	99.60 ± 0.07
	$(32,24,5,1)$	88.41 ± 0.86	91.78 ± 1.42

Table 5.12: Taxonomic classification accuracy (%± standard error) of the KA-GNN framework, contrasting the balanced Sub_600 reference against the complete *Full Test* distribution across varying network depths (l) and harmonic grid dimensions (K).

maintain classification stability under dataset scaling. As reported in Table 5.12, at a standard depth of 3 layers, the architecture achieves its highest classification score on the RGs *Full Test* domain, reaching a mean accuracy of $99.60 \pm 0.07\%$. A similar trend is visible for the configuration optimized on the smaller Sub_300 subset (Table 5.11), which generalizes to the complete dataset with an accuracy of $99.01 \pm 0.14\%$. This generalization behavior suggests that parameterizing edge and node updates via learnable univariate functions helps the network retain key structural cues when moving to larger and more skewed distributions. By substituting rigid linear transformations with functions parameterized over harmonic grids (K), the model avoids the representational limitations frequently observed in discrete kernels.

However, the results also show that performance scaling is highly sensitive to the chosen network depth. Increasing the architecture to 5 layers generally leads to lower generalization metrics and higher standard errors. This degradation is particularly evident in the uncompressed RG domain for the Sub_300 subset, where accuracy drops to $72.24 \pm 2.23\%$ on the *Full Test*. A comparable, though less severe, drop is visible in the Sub_600 reference. This pattern suggests that deeper configurations are more susceptible to overfitting on the specific density patterns of the smaller training subsets, making shallower configurations ($l = 3$) a more reliable choice.

To evaluate model performance across individual taxonomic lineages, Figure 5.1 presents a radar chart comparing the classification accuracy per kingdom on the RGs domain. This evaluation represents the *Full Test* environment, measuring the generalization performance of the best models previously optimized on the balanced Sub_300 subset across all evaluated combinations of graph kernels and node relabeling strategies for the GKNN approach, alongside the KA-GNN framework.

The radar chart highlights that performance across the majority of the classes (*Animals*, *Plants*, *Fungi*, *Bacteria*, *Archaea*) remains consistently high for almost all configurations, with scores generally exceeding 80%. The primary differentiator among the architectures is the highly variable *Protists* class. On this domain, different configuration show a significant drop in performance, falling toward the 20% accuracy threshold. This class-by-clade breakdown highlights a paradox regarding the ER solution. Although this configuration achieves some of the highest overall accuracy scores across the entire benchmark, the radar chart reveals that its generalization is highly uneven. Specifically, WL-ER displays a sharp performance drop on the *Protists* class, representing the lowest score among all tested architectures. Finally, the KA-GNN framework maintains the most balanced and stable profile across the entire plot, preserving an accuracy of approximately 95% even on the complex *Protists* class. This result suggests that

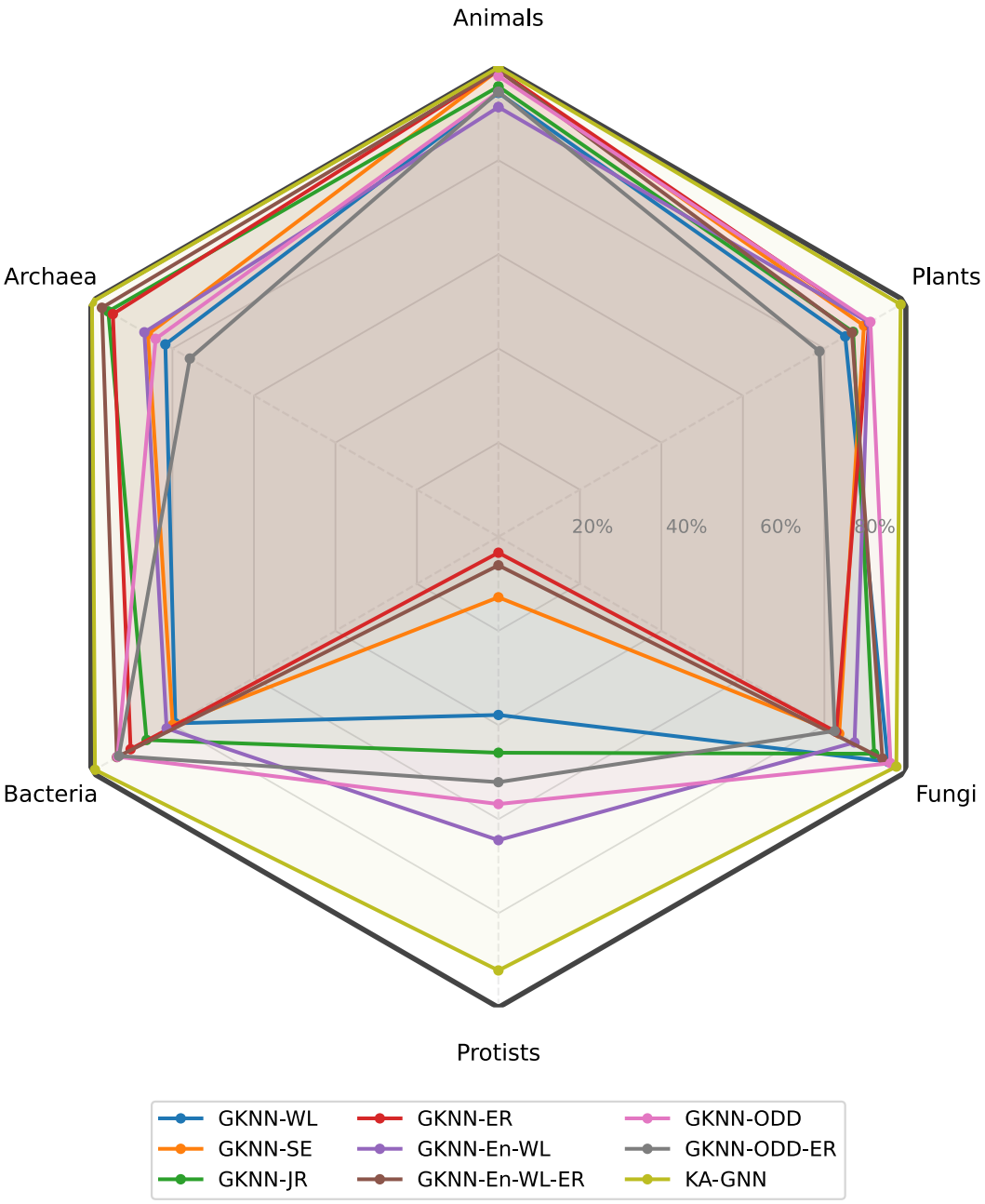


Figure 5.1: Radar chart displaying the classification accuracy per kingdom on the KEGG *Full Test* using the best configuration of the `Sub_300` subset on RGs. The plot shows all the combinations of node relabeling strategies and graph kernels within the GKNN framework against the KA-GNN baseline.

continuous, learnable edge activations are better suited to handle topological variations and generalize effectively across highly skewed class distributions.

5.4.2 Prediction analysis

To investigate the classification behavior of the proposed architecture, Figure 5.2 illustrates the row-normalized confusion matrices computed over the aggregated 10-fold cross-validation split on the RG domain for the peak configurations of both the GKNN (WL-ER kernel) and KA-GNN. The breakdown across taxonomic categories shows that the global accuracy differences are driven by specific biological clades. For highly distinct metabolic domains, both architectures perform near perfectly. The KA-GNN achieves a per-class true positive rate of 1.00 for *Animals*, *Bacteria* and *Archaea*. The GKNN performs competitively on *Bacteria* (0.99) and *Animals* (0.98), but misclassifies 0.07 of the *Archaea* samples as *Bacteria*. This confusion between the two prokaryotic groups may reflect the fact that, at the level of graph topology, Archaea and Bacteria share a number of common reaction patterns.

A similar trend is visible across multi-cellular eukaryotes (*Animals*, *Plants*, and *Fungi*). The KA-GNN correctly classifies 1.00 of *Animals*, 0.97 of *Plants*, and 0.95 of *Fungi*. Conversely, the GKNN shows a symmetric error pattern, misclassifying 0.04 of *Plants* as *Fungi* and 0.04 of *Fungi* as *Plants*. This overlap in the confusion matrix may reflect the fact that *Plants* and *Fungi* share a subset of common metabolic reactions, which can make their graph representations difficult to distinguish for a discrete kernel.

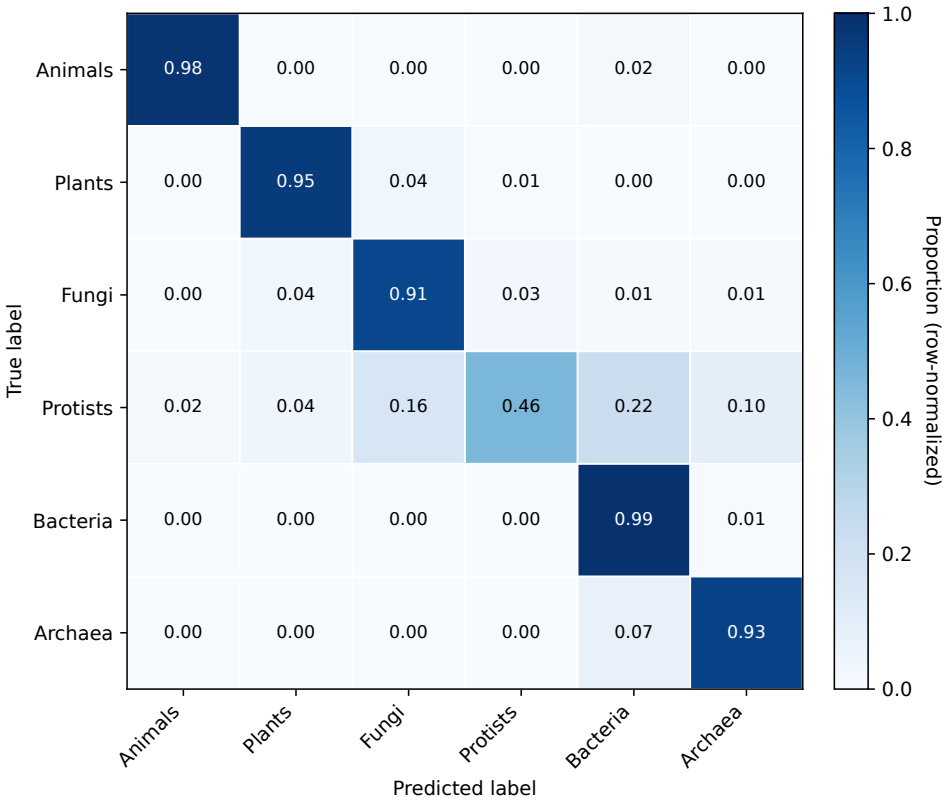
The main difference between the two models occurs in the *Protists* class. Within the dataset, *Protists* represent a highly heterogeneous and minority taxonomic group characterized by a high structural diversity among its metabolic networks. This high topological variance limits the discrete mechanism of the GKNN, yielding a true positive rate of 0.46 and scattering the misclassified samples across *Bacteria* (0.22), *Fungi* (0.16), and *Archaea* (0.10). The KA-GNN handles this group more effectively, achieving a true positive rate of 0.78 compared to 0.46 for the GKNN. Additionally, its misclassifications remain within the eukaryotic domain (`Fungi` and `Plants`), whereas the GKNN scatters errors also into `Bacteria` and `Archaea`. This suggests that the continuous edge transformations provide a more flexible representation for heterogeneous groups.

5.4.3 Mask Interpretability

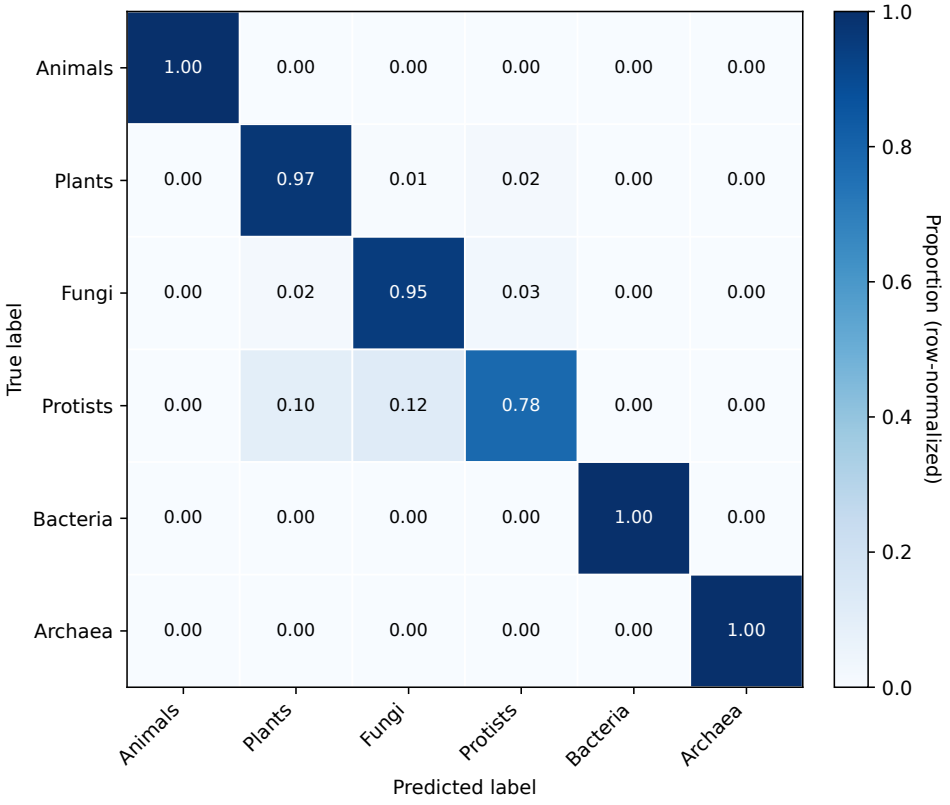
To evaluate the decision-making process of the GKNN framework, we analyze the structural properties and taxonomic alignment of the learned matching masks. Because the models were trained in `feature` mode, each mask acts as a collection of feature prototypes optimized over the input dataset vocabulary.

We visualize these representations across all three metabolic formulations (AMNs, mDAGs, RGs) under both the WL and ODD kernels in Figure 5.3, 5.4 and 5.5. The top panels of these figures display the mean weight that the mask prototypes assign to each input topological feature. Here, a positive weight denotes an *excitatory* feature whose presence in a node's local neighborhood amplifies the mask's activation, whereas a negative weight represents an *inhibitory* feature that suppresses it.

The bottom panels measure the global classification impact of each individual mask. This importance metric is calculated by ablating the mask within the pooled graph representation and recording the resulting increment in cross-entropy loss (ΔLoss) on the training set. A high ΔLoss indicates a critical structural filter upon which the network relies for taxonomic



(a) GKNN (Best Config)



(b) KA-GNN (Best Config)

Figure 5.2: Taxonomic confusion matrices on the `Sub_600` Reaction Graphs (RGs), comparing the error distributions of (a) the best GKNN configuration (32 masks, 24 nodes, 3 layers with WL-ER mechanisms) and (b) the best KA-GNN (32 masks, 24 nodes, 3 layers with $K = 1$).

discrimination. Because the total cross-entropy loss is computed as an average of per-graph losses, this increment can be decomposed into the exact contributions of the six biological kingdoms. This class breakdown reveals functional specialization: a single-color bar represents a specialized mask dedicated to isolating a unique taxonomic clade, whereas a multi-colored bar represents a shared, domain-general structural motif.

When evaluating the RG domain, where each input feature corresponds to an explicit KEGG reaction identifier, the learned masks show a clear alignment with known biochemical pathways. By mapping the highly-weighted functional features back to their official KEGG pathway annotations and Enzyme Commission number (EC) definitions [31], we observe that the optimization process selects reactions that belong to the same functional metabolic categories, rather than picking unrelated biochemical steps.

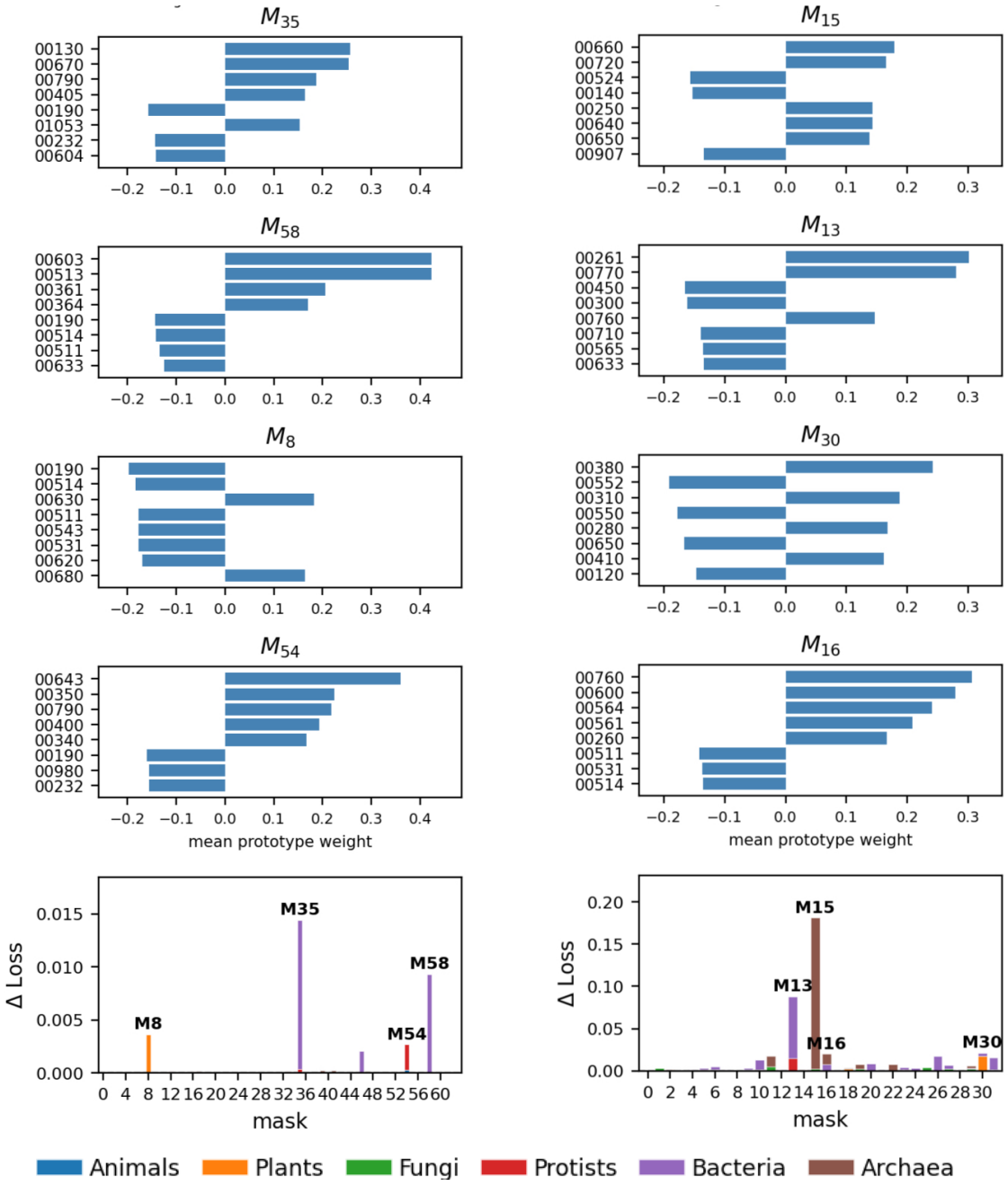
Under the WL kernel (Figure 5.5 (a)), mask M_5 is dominated by branched-chain-amino-acid aminotransferase reaction (EC 2.6.1.42), which are characteristic of amino-acid metabolism. Mask M_3 groups nucleotide-metabolism reactions, primarily by nucleoside-diphosphate kinase (EC 2.7.4.6), ribonucleoside-diphosphate reductase (EC 1.17.4.1) and aspartate carbamoyltransferase (EC 2.1.3.2). Mask M_{31} isolates a fatty-acid module, coupling fatty-acid activation (long-chain acyl-CoA synthetase, EC 6.2.1.3) with biosynthetic cycle components (acetyl-CoA carboxylase, EC 6.4.1.2; 3-oxoacyl-ACP reductase, EC 1.1.1.100; and 3-hydroxyacyl-ACP dehydratase, EC 4.2.1.59).

The same behavior is consistent under the ODD kernel (Figure 5.5 (b)), where mask M_{10} concentrates on thiamine biosynthesis (EC 2.5.1.3, EC 4.1.99.17 and EC 2.7.4.7) and mask M_2 isolates molybdenum-cofactor biosynthesis (EC 2.10.1.1, EC 4.6.1.17 and EC 2.7.7.75). The masks thus function as detectors of functionally cohesive metabolic pathways rather than arbitrary feature collections. In broader terms, this biochemical alignment provides evidence that the model is not relying on spurious, dataset-specific correlations to achieve high classification accuracy. Instead, the optimization process leverages core biological subsystems as fundamental components for taxonomic discrimination.

The per-mask importance panels reveal the distribution of these decisions. Although the model optimizes multiple feature detectors, taxonomic classification is driven by a small subset of masks, as shown by the sharp spikes in the importance profiles.

A mask's importance is not necessarily proportional to how sharply its weights are tuned. On RGs, M_{23} under the WL kernel and M_7 under the ODD kernel display flat, lower prototype-weight profiles, yet they produce the largest ΔLoss. This indicates that the model relies on a broad, generalist mask spread across co-occurring reactions for its baseline predictions.

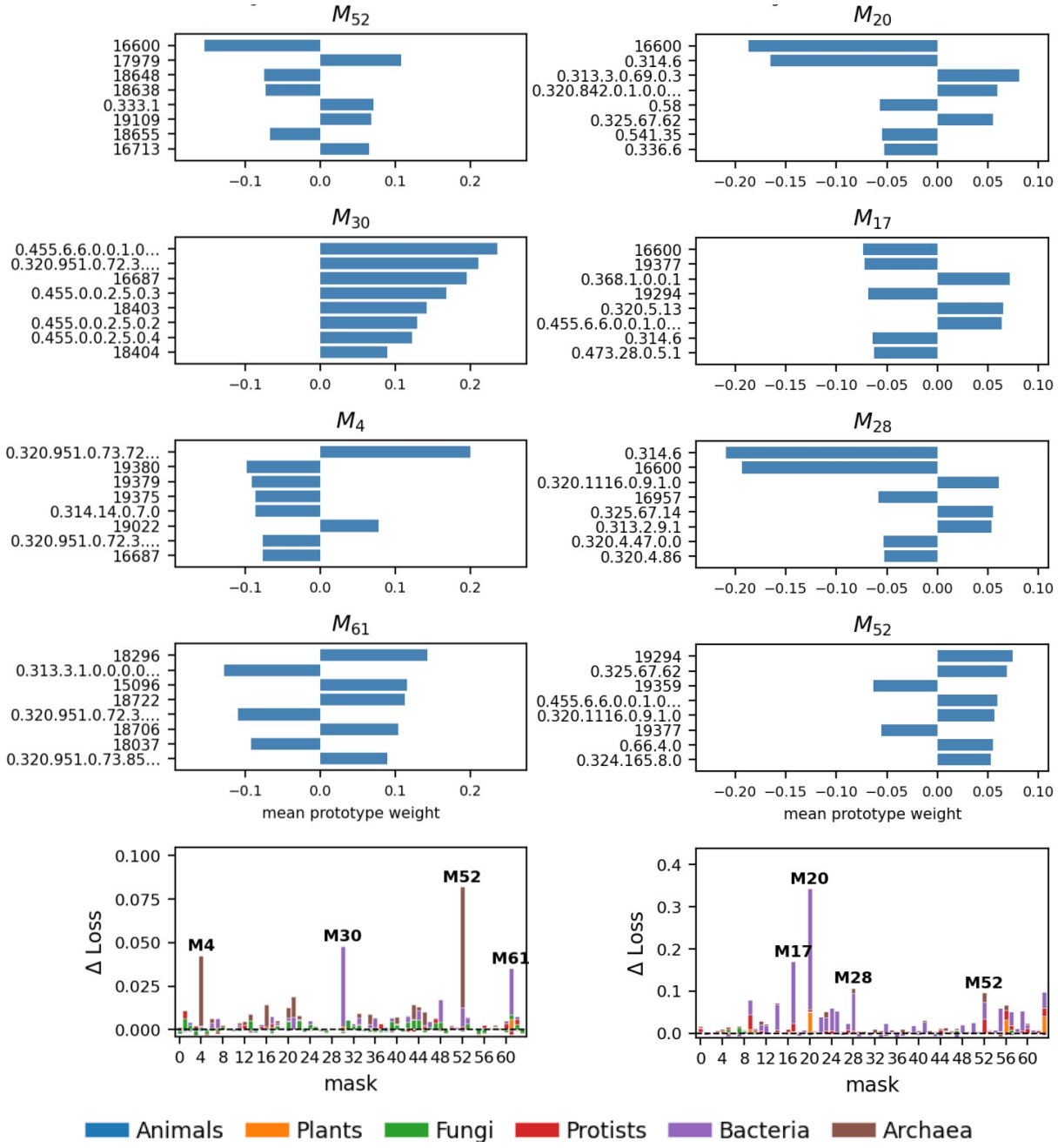
Finer taxonomic distinctions are then resolved by a division of labor among the remaining masks. Decomposing each mask's ΔLoss into the contributions of individual kingdom reveals that these auxiliary, high-impact masks behave as dedicated kingdom "specialists". On AMN, for instance, M_{35} and M_{58} are devoted almost entirely to *Bacteria*, while M_8 and M_{54} are tuned to *Plants* and *Protists*, respectively. No single mask specializes in the Animal class. Because animal identity is encoded redundantly across multiple masks, their individual ablation results in a negligible ΔLoss per mask. Consequently, their segment is not visible in the plots, despite the near-perfect classification accuracy achieved for this kingdom. To classify an organism, the network leverages the consensus of the broad generalist mask alongside a predictable set of these highly specialized diagnostic pathways.



(a) WL kernel

(b) ODD kernel

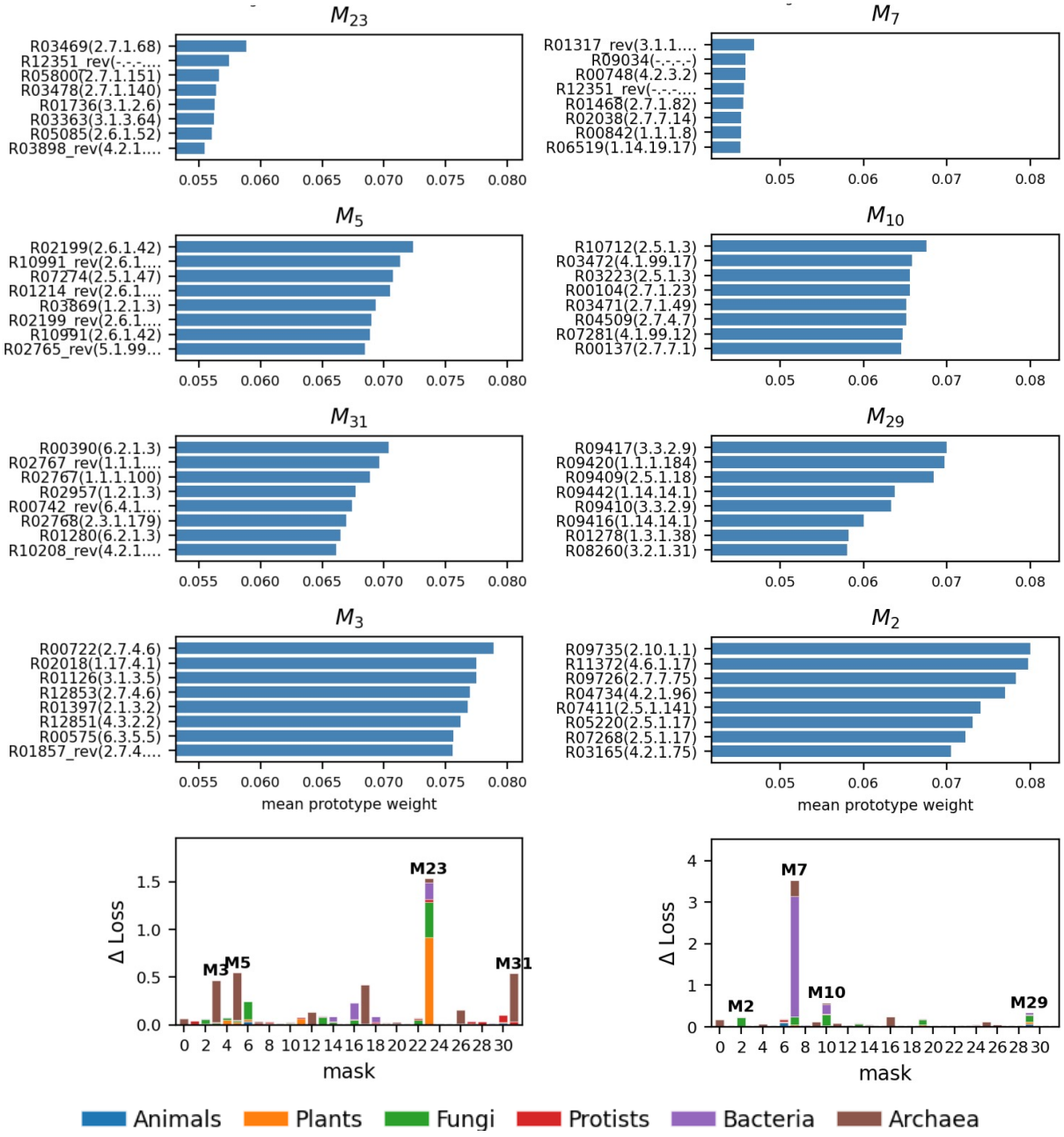
Figure 5.3: Masks interpretability for the **AMNs dataset**, comparing (a) WL and (b) ODD kernels. For each kernel, the four upper panels show the learned masks ranked by their impact on the loss. Each mask is summarized by the mean weight that its prototypes assign to the features (shown in the y axis). The bottom panel reports the increase in classification loss (ΔLoss) caused by ablating each mask, decomposed by biological kingdom (each coloured segment is the contribution of that kingdom's graphs, and the segments sum exactly to the mask's total ΔLoss).



(a) WL

(b) ODD

Figure 5.4: Masks interpretability for the **mDAGs dataset**, comparing (a) WL and (b) ODD kernels. For each kernel, the four upper panels show the learned masks ranked by their impact on the loss. Each mask is summarized by the mean weight that its prototypes assign to the features shown in the y axis. The bottom panel reports the increase in classification loss (ΔLoss) caused by ablating each mask, decomposed by biological kingdom (each coloured segment is the contribution of that kingdom's graphs, and the segments sum exactly to the mask's total ΔLoss).



(a) WL

(b) ODD

Figure 5.5: Masks interpretability for the **RGs dataset**, comparing (a) WL and (b) ODD kernels. For each kernel, the four upper panels show the learned masks ranked by their impact on the loss. Each mask is summarized by the mean weight that its prototypes assign to the features shown in the y axis. The bottom panel reports the increase in classification loss (ΔLoss) caused by ablating each mask, decomposed by biological kingdom (each coloured segment is the contribution of that kingdom's graphs, and the segments sum exactly to the mask's total ΔLoss).

Kernel Regimes comparison

While both kernels exhibit similar qualitative behavior, they differ in the concentration of their importance distributions. Under the ODD kernel, a single mask absorbs a larger share of the total importance than under the WL kernel: the peak rises to $\Delta\text{Loss} \approx 3.5$ (M_7) against ≈ 1.5 (M_{23}) on RGs, to ≈ 0.34 (M_{20}) against ≈ 0.08 (M_{52}) on mDAG, and to ≈ 0.18 (M_{15}) against ≈ 0.014 (M_{35}) on AMN. Therefore, the ODD kernel yields a more concentrated, lower-redundancy representation, whereas the WL kernel distributes the classification cues across a wider set of masks.

Although the absolute ΔLoss of separately trained models is not directly comparable due to different baseline losses, the more peaked shape of the ODD importance distribution remains a consistent trend. This concentration interacts with a cross-dataset trend: the large, label-rich RG domain operates under a centralized regime where a single "generalist" mask dominates and the absolute ΔLoss is largest. The mDAG representation distributes importance across a small, constrained group of masks with comparable values, while the compact AMN space exhibits a highly distributed and redundant profile, with ΔLoss values an order of magnitude smaller (particularly under the WL kernel, where removing any single mask yields minimal changes in the loss).

The orientation of the prototype weights changes with the representation as well. On RGs, the dominant features are almost exclusively positive, meaning the masks accumulate excitatory evidence based on the presence of metabolic pathways. On mDAGs and AMNs, the masks are instead contrastive, balancing positive and negative weights, with specific features recurring as inhibitors shared by several masks (e.g., feature 190 on AMN; features 16600 and 0.314.6 on mDAG under ODD). In these abstractions, the model exploits both the presence and the verified absence of specific features to secure taxonomic boundaries. This structural variation reflects how discriminative the features are within each specific graph layer.

Chapter 6

Conclusions and Future Works

This thesis investigated how well an organism's taxonomic classification can be predicted from its metabolic network structure. Based on the idea that related organisms share similar metabolic wiring, we set up a classification task using data from the KEGG repository. We used classification accuracy to evaluate how well each graph representation preserves biological information across different levels of abstraction.

Our study investigated these approaches on a dataset of 8904 organisms from the KEGG repository, evaluating both the full, unbalanced collection and smaller, balanced subsets (`Sub_300` and `Sub_600`). Within this setup, we focused on two main areas. First, we analyzed three representations across a spectrum of abstraction: granular Reaction Graphs (RGs), intermediate Metabolic Directed Acyclic Graphs (mDAGs) and highly compressed Abstract Metabolic Networks (AMNs). Second, we compared two different learning approaches: the kernel-driven Graph Kernel Neural Network (GKNN) [15] and the learnable Kolmogorov-Arnold Graph Neural Network (KA-GNN) [17]. Along the way, we introduced three original contributions: the cycle-aware Enhanced Weisfeiler-Lehman (En-WL) kernel to handle the redundancy of reversible reactions, a set of node-relabeling strategies (Just Reactions (JR), Separate Enzymes (SE) and Enzymes and Reactions (ER)) to better encode KEGG labels, and the first adaptation of the KA-GNN architecture to our metabolic graphs.

The central finding of the research is that metabolic structure alone carries a clear signature of taxonomic identity, and this signal remains visible even under significant structural compression. Even the AMN representation, which collapses entire pathways into a single node, proved sufficient for near-perfect classification. On the `Sub_600` subset, the GKNN reached $99.42 \pm 0.15\%$ accuracy using the ODD kernel, while the KA-GNN achieved $99.15 \pm 0.20\%$. This suggests that abstracting networks to the pathway level preserves sufficient discriminative features for classification. The comparison between models showed that there is no single "best" architecture, as performance heavily depends on the specific topological abstraction. On the highly compressed AMNs, the two paradigms performed almost identically (both scoring above 99%) probably because pathway-level features are distinct enough to allow both architectures to easily resolve taxonomic boundaries without suffering from local topological noise. However, the mDAGs proved to be a harder middle ground: on this representation, the discrete GKNN reached a peak of 97.09%, but struggled with the fragmentation and structural disconnections introduced by the acyclic decomposition. In contrast, the continuous edge transformations of the KA-GNN adapted far more naturally to the hierarchical, directional flow of the mDAGs, achieving a comparable 97.04%. We also observed a clear difference when testing deeper architectures. While a shallow three-layer setup was optimal for both frameworks, they failed differently beyond this threshold. The GKNN suffered a severe oversmoothing collapse on

RGs (which we tried to fix using our encodings) and degraded on disconnected mDAGs. In contrast, the KA-GNN proved significantly more resilient to depth. However, its Fourier basis resolution (K) interacts with model depth in opposite ways depending on the representation: it improved stability on mDAGs, but caused overfitting on dense RGs, meaning $K = 1$ worked best as a regularizer. Our proposed structural encodings proved highly effective. The En-WL kernel confirmed that the "ping-pong" effect of two-node cycles causes a slight performance degradation, and its combination with the ER encoding (En-WL-ER) further consolidates this gain, establishing a new global maximum of $94.55 \pm 1.31\%$ on the balanced `Sub_600` RGs, even surpassing WL-ER and confirming that cycle-aware pruning and enzymatic relabeling can be complementary sources of robustness. Two additional analyses provide supporting evidence for the quality of what the models have learned. In the robustness test on the complete dataset, the KA-GNN achieved $99.60 \pm 0.07\%$ on RGs with very low variance. As discussed in Chapter 5, this result should be interpreted carefully, since for most taxonomic groups the Full Test does not introduce genuinely unseen organisms. On the interpretability side, a clade-by-clade breakdown showed that the performance gap between models was driven almost entirely by the single hardest group: the Protists. The KA-GNN raised the true positive rate for Protists from 0.46 to 0.78, making mistakes only within the eukaryotic domain. Most importantly, when we decoded the learned masks of the GKNN, we found that the optimization had autonomously rediscovered core biological pathways (such as amino-acid, nucleotide, and fatty-acid metabolisms). The fact that a model trained only on taxonomic labels isolates functionally cohesive enzyme groups provides evidence that the optimization aligns with verified biological pathways.

Naturally, these results come with some caveats. They depend entirely on KEGG annotations, which are not uniform across taxonomic groups. Groups like Archaea and Protists often have smaller genomes or simpler metabolic pathways, which naturally leads to smaller and less connected graphs. This means the models might partially classify organisms based on these differences in graph size and connectivity, rather than complex topological patterns. Additionally, all three representations are static and structural, capturing which reactions are present but omitting dynamic data like reaction rates. In practice, this means the models are largely blind to differences that are purely regulatory or kinetic. Consequently, our high accuracy only proves that the models can separate organisms based on their available enzymes, ignoring how those networks actually function in real life. Finally, the classification task is limited to six broad taxonomic groups, which may not reflect finer-grained biological distinctions. These limitations suggest several directions for future research. First, transitioning from unipartite reaction graphs to hypergraph learning would preserve the multi-substrate nature of biochemical reactions, quantifying the information lost in binary edge projections. Second, evaluating the learned Fourier activations of the KA-GNN could clarify which structural frequencies encode taxonomic information. Finally, testing these representations at finer taxonomic ranks than the kingdom level would evaluate the resolution limits of metabolic topology. Additionally, exploring hybrid architectures that employ kernel functions as the learnable univariate activations of the Kolmogorov-Arnold layers, such as Kernel-KANs [32] and Radial-Basis-Function KANs (RBF-KANs) [33], represents a promising direction for future research.

In summary, the results show that metabolic topology alone carries a consistent signal of taxonomic identity, and that the best approach depends on the level of graph abstraction: discrete kernels work reliably on compressed representations, while continuous learnable edges scale better on dense ones. In particular, the GKNN masks, trained only to predict a kingdom, assign the highest weights to reactions belonging to known metabolic pathways such as amino-acid, nucleotide, and fatty-acid metabolism. This is consistent with the model having learned biologically meaningful features, though it does not rule out other contributing factors.

Bibliography

1. Berg, J., Tymoczko, J. & Stryer, L. Biochemistry. **5,** 568–593, 648–649 (2002).
2. Nelson, D. & Cox, M. Lehninger principles of biochemistry. **7,** 10–28, 481–488 (2017).
3. Barabási, A. & Oltvai, Z. Network Biology. *Nature Reviews Genetics* **5** (2004).
4. Kanehisa, M. & Goto, S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* **28** (2000).
5. Kanehisa, M., Furumichi, M., Sato, Y., Kawashima, M. & Ishiguro-Watanabe, M. KEGG for taxonomy-based analysis of pathways and genomes. *Nucleic Acids Research* **52** (2024).
6. Jing, L., Shah, F., Mohamad, M., Hamran, N., A.H.M., S., Deris, S. & Alashwal, H. Database and Tools for Metabolic Network Analysis. *Biotechnology and Bioprocess Engineering* (2014).
7. Alberich, R., Castro, J., Llabrés, M. & Palmer-Rodrígues, P. Metabolomics analysis: Finding out metabolic building blocks. *PLOS* (2017).
8. García, I., Chouaia, B., Llabrés, M., Palmer-Rodrígues, P. & Simeoni, M. Analysing the expressiveness of metabolic networks representations. *Artificial Life and Evolutionary Computation* (2024).
9. García, I., Chouaia, B., Llabrés, M., Palmer-Rodrígues, P. & Simeoni, M. Exploring the expressiveness of abstract metabolic networks. *PLOS ONE* (2023).
10. Cocco, N., Llabrés, M., Reyes-Prieto, M. & Simeoni, M. MetNet: A two-level approach to reconstructing and comparing metabolic networks. *PLOS ONE* (2021).
11. Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S. & Liu, H. Graph Learning: A Survey. *IEEE Transactions on Artifical Intelligence* (2021).
12. Kriege, N., Johansson, F. & Morris, C. A survey on graph kernels. *Springer* (2020).
13. Shervashidze, N., Schweitzer, P., Jan van Leeuwen, E., Mehlhorn, K. & Borgwardt, K. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research* (2011).
14. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. & Yu, P. S. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks* (2019).
15. Cosmo, L., Minello, G., Bicciato, A., Bronstein, M., Rodolà, E., Rossi, L. & Torsello, A. Graph Kernel Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* **36** (2025).
16. Kordi, R. Application of Graph Learning Methods to Metabolic Networks (2024).
17. Li, L., Zhang, Y., Wang, G. & Xia, K. Kolmogorov-Arnold graph neural networks for molecular property prediction. *Nature Machine Intelligence* **7** (2025).

18. Zhang, M., Cui, Z., Neumann, M. & Chen, Y. An end-to-end deep learning architecture for graph classification. *The Thirty-Second AAAI Conference on Artifical Intelligence (AAAI-18)* (2018).

19. Klipp, E., Liebermeister, W., Wierling, C. & Kowald, A. Systems Biology. **2,** 4–8 (2016).

20. Kitano, H. Systems Biology: A Brief Overview. *SCIENCE* **295** (2002).

21. Palsson, B. Systems Biology: Constraint-based Reconstruction and Analysis (2015).

22. *KEGG: Kyoto Encyclopedia of Genes and Genomes* `https://www.kegg.jp`. [Accessed 16-03-2026].

23. Pavlopoulos, G., Secrier, M., Moschopoulos, C., Soldatos, T., Kossida, S., Aerts, J., Schneider, R. & Bagos, P. Using graph theory to analyze biological networks. *BigData Mining* (2011).

24. Lacroix, V., Cottret, L., Thébault, P. & Sagot, M.-F. An introduction to metabolic networks and their structural analysis. *PNAS,* 14–16 (2008).

25. Arita, M. The metabolic world of Escherichia coli is not small. *PNAS* (2004).

26. Huss, M. & Holme, P. Currency and commodity metabolites. *IET Systems Biology* (2006).

27. Ma, H. & Zeng, A. Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *PNAS,* 274–275 (2002).

28. Prigent, S., Frioux, C., Dittami, S. M., Thiele, S., Larhlimi, A., Collet, G., Gutknecht, F., Got, J., Eveillard, D., Bourdon, J., Plewniak, F., Tonon, T. & Siegel, A. Meneco, a Topology-Based Gap-Filling Tool Applicable to Degraded Genome-Wide Metabolic Networks. *PLOS Computational Biology* **13** (2017).

29. Lobb, B., Tremblay, B. J.-M., Moreno-Hagelsieb, G. & Doxey, A. C. An assessment of genome annotation coverage across the bacterial tree of life. *Microbial Genomics* **6** (2020).

30. Da San Martino, G., Navarin, N. & Sperduti, A. A tree-based kernel for graphs with continuous attributes. *IEEE Transactions on Neural Networks and Learning Systems* (2016).

31. McDonald, A., Boyce, S. & Tipton, K. F. ExplorEnz: the primary source of the IUBMB enzyme list. *Nucleic Acids Research* **37** (2008).

32. Alhusseiny, A. Kernel-KAN: Kernel Kolmogorov Arnold Networks. *Available at SSRN 5501938* (2025).

33. Chao, Z., Liu, X., Wu, Z. & Li, X. RBF-KAN: Radial Basis Function-Kolmogorov-Arnold Network. *IEEE Internet of Things Journal* **13** (2026).