Quantum Hackathon Challenge: Kernel Methods for Molecular Classification

Team: QFennecs

Team Members:
Rihab HOCEINI (@Rihab HC), Raouf Ould Ali (@Raouf SC), Amieur Lilya Fatima-Zohra (@ohlilyaaa), Widad Hassina Belkadi (@belkadiWidad_Algeria), Hamza Abderaouf KHENTACHE (@¬¬)

25.10.2025 (V1.0)

1 Introduction

Many real-world datasets are inherently relational and are best represented as graphs, where nodes denote entities and edges capture their relationships. This structure preserves essential topology and connectivity information, making graph representations powerful across domains such as chemistry [1, 2], biology [3], and plant systems [4]. Graph-based learning methods have thus become central to tasks like molecular property prediction, as they exploit domain-specific relationships for improved interpretability and accuracy. However, the performance of these models strongly depends on how graphs are represented before learning. Beyond raw connectivity, specialized *graph feature maps* [5] are needed to transform each graph into informative, high-dimensional embeddings that capture both structural and semantic properties crucial for accurate prediction.

2 Methodology

The overall molecular classification pipeline consists of five main stages: dataset loading, graph conversion, cross-validation, feature map extraction, and SVM kernel-based classification.

2.1 Step 1. Dataset Loading

The five benchmark molecular datasets (*AIDS*, *PTC-MR*, *MUTAG*, *NCI1*, *PROTEINS*) are imported using the `TUDataset` interface from PyTorch Geometric. Each dataset contains a collection of molecular graphs $G_i = (V_i, E_i)$ with atom and bond attributes and a corresponding graph-level target label y_i.

2.2 Step 2. Graph Conversion.

Each molecule is modeled as an undirected graph $G = (V, E)$ using `NetworkX`, where nodes V represent atoms and edges E denote chemical bonds with associated multiplicities. This abstraction provides a consistent molecular representation across domains (*chemistry*, *toxicology*, *biology*). To ensure compatibility among datasets with differing atom and bond conventions, a unified node–edge attribute mapping is defined as follows.

A. Node Mapping. Each atom is identified by a categorical label corresponding to its chemical symbol. To ensure consistency across domains, we defined explicit lookup tables for each dataset as follows:

Table 1: Sub Keys of atom type mappings used for different molecular graph datasets.

Dataset	Atom Type Mapping
MUTAG	{0 : C, 1 : N, 2 : O, 3 : F, 4 : I, 5 : Cl, 6 : Br}
AIDS	{0 : C, 1 : O, 2 : N, 3 : Cl, 4 : F, 5 : S, 6 : Se, 7 : P, 8 : Na, 9 : I, 10 : Co, 11 : Br, 12 : Li, 13 : Si, 14 : Mg, 15 : Cu, 16 : As, 17 : B, 18 : Pt, 19 : Ru, 20 : K, 21 : Pd, 22 : Au, 23 : Te, 24 : W, 25 : Rh, 26 : Zn, 27 : Bi, 28 : Pb, 29 : Ge, 30 : Sb, 31 : Sn, 32 : Ga, 33 : Hg, 34 : Ho, 35 : Tl, 36 : Ni, 37 : Tb}
PTC-MR	{0 : C, 1 : N, 2 : O, 3 : Cl, 4 : F, 5 : S, 6 : Br, 7 : P, 8 : I, 9 : Na, 10 : K, 11 : Li, 12 : Ca, 13 : Cu, 14 : Mg, 15 : As, 16 : B, 17 : Sn}
PROTEINS	{1 : Helix, 2 : sheet, 3 : turn}

Each node thus stores an attribute `atom_type` corresponding to its symbol, e.g., `atom_type='O'` for oxygen.

B. Edge Mapping. Edges represent chemical bonds between atoms, each annotated with a real-valued attribute `bond_order` determined by the bond type (single $\rightarrow$ 1.0, double $\rightarrow$ 2.0, triple $\rightarrow$ 3.0, aromatic $\rightarrow$ 1.5). If bond information is missing, a single bond (`bond_order`=1.0) is assumed. This encoding preserves molecular topology and chemical structure for downstream feature extraction.

2.3 Step 3. Cross-Validation Split.

All molecular graphs are partitioned into 10 folds using stratified cross-validation to maintain label balance. Each iteration trains on 90% of the data and tests on the remaining 10%.

2.4 Step 4. Feature Map Extraction.

For each training fold, a feature extractor generates a high-dimensional embedding $\phi(G)$ for every molecular graph. Three feature extraction methods are implemented, which we present in increasing order of complexity and robustness:

2.4.1 Chemical Laplacian based Feature Maps

To capture both global and local structural information, we employ a feature extractor based on the *Chemical Laplacian*. This representation integrates chemical semantics with graph structure by weighting the Laplacian matrix using atomic importance $\alpha(v_i)$ and bond strength $w(i, j)$, embedding both atom diversity and bond multiplicity into the spectral domain. The Laplacian eigenvalues provide compact global descriptors of molecular connectivity and stability. To complement these, we include **topological features** (global structure), **bridge-tree features** (hierarchical connectivity), and **Weisfeiler–Lehman histograms** (local substructures). Together, they form a multi-scale, isomorphism-invariant representation sensitive to both local and

global chemical patterns. However, since this method relies on atom- and bond-level semantics, it performs best on chemical graphs (e.g., *MUTAG*, *NCI1*) and is less suitable for biological datasets like *PROTEINS*, where nodes represent amino acids and edges lack chemical bond meaning. To overcome this, we introduce the attention-based feature map.

2.4.2 Attention Based Feature Maps

The proposed feature map integrates attention mechanisms at both the node and edge levels to enhance classical graph-based descriptors with adaptive weighting. Its objective is to capture heterogeneous contributions of atoms and bonds to the overall molecular structure allowing the model to focus on chemically and structurally relevant substructures.

A. Node-Level Attention. Each node $v_i \in V$ represents an atom described by its type, degree, and local neighborhood features. The node attention module learns to weight atomic importance through

$$\mathbf{a}_i = \mathrm{softmax}\left(\frac{(\mathbf{x}_i\mathbf{W}_q)(\mathbf{x}_j\mathbf{W}_k)^\top}{\sqrt{d}}\right),$$

where $\mathbf{x}_i$ is the node feature, and $\mathbf{W}_q, \mathbf{W}_k$ are learned projections. The resulting attention weights highlight chemically relevant atoms—such as heteroatoms, bridge atoms, or functional groups, capturing local saliency and topological influence within the molecular graph.

B. Edge-Level Attention. Analogously, the edge attention module assigns weights to chemical bonds based on bond type (single, double, triple, aromatic), the atom types of the endpoints, and their local connectivity. Given the edge feature vector $\mathbf{e}_{uv}$, the attention score is computed as

$$\alpha_{uv} = \sigma(\mathbf{e}_{uv}\mathbf{W}_e\mathbf{W}_e^\top\mathbf{e}_{uv}^\top),$$

where $\sigma(\cdot)$ denotes the sigmoid activation and $\mathbf{W}_e$ is the learned edge projection. This term reflects the structural importance of each bond and enables differentiation between rigid (double, aromatic) and flexible (single) connections in the molecular graph.

C. Integration with Structural Features. Attention weights modulate classical graph descriptors to enhance chemical and structural relevance:

- **Weisfeiler–Lehman histograms:** attention-weighted substructure refinements.
- **Spectral descriptors:** Laplacian eigenvalues capturing global connectivity.
- **Topological features:** weighted degree, clustering, and component statistics.

- **Pharmacophore patterns:** attention-weighted atom–pair distances.
- **Random walk statistics:** attention-guided local traversal patterns.
- **Persistence features:** stability of substructures under attention-weighted filtrations.

In the experiments, we evaluate the impact of these features to the performance of the feature extraction and classification.

D. Rationale and Informative Properties. This hybrid approach fuses *learned attention* with *graph descriptors* (structural, spectral, topological), producing interpretable and robust representations. It provides:

- *Chemical interpretability:* highlights chemically relevant atoms and bonds.
- *Domain adaptability:* learns structural importance without relying on predefined semantics.
- *Structural sensitivity:* distinguishes molecules with similar topology but distinct chemistry.
- *Noise robustness:* attenuates non-informative atoms via soft weighting.

Overall, the attention-based feature map yields expressive, domain-general graph embeddings that enhance molecular prediction.

2.4.3 Quantum Based Feature Maps

To explore quantum-enhanced representations, we designed a `QURIQuantumFeatureExtractor` leveraging the *QURI Parts* framework for quantum simulation. This module encodes molecular graphs as quantum circuits and extracts entanglement and spectral observables that serve as graph-level descriptors.

A. Graph-to-Quantum Encoding Each molecular graph $G = (V, E)$ is transformed into a quantum *graph state*, where each atom corresponds to a qubit and each chemical bond induces a Controlled-Z (CZ) entangling gate. All qubits are initialized in superposition using Hadamard gates:

$$|G\rangle = \prod_{(i,j)\in E} CZ_{ij} \prod_{v_i \in V} H_i |0\rangle^{\otimes |V|}.$$

This state captures molecular connectivity through multi-qubit entanglement.

B. Quantum Observables. From this state, several quantum and spectral properties are derived:

- **Entanglement Entropy:** the von Neumann entropy across molecular partitions, quantifying electronic delocalization and correlation.
- **Quantum Walk Overlap:** the return probability of a continuous-time quantum walk over G, reflecting dynamic coherence and connectivity.
- **Molecular Hamiltonian Expectation:** approximate electronic observables (total energy, HOMO–LUMO gap) obtained from the eigenvalues of the adjacency matrix, following a Hückel-like model.

C. Interpretative Value. These quantum observables act as attention-like weights, emphasizing regions of high entanglement or energy variation. As such, they introduce *quantum features* into molecular feature maps, highlighting chemically significant substructures beyond classical graph connectivity.

2.5 Step 5. Classification Training and Evaluation.

After the feature extraction phase, each feature vector is normalized using `StandardScaler`. The same scaling parameters are then applied to the test set. This step ensures that all features contribute equally during the training phase and prevents dominance of features with larger numeric ranges.
The standardized feature matrices are used to train a support vector machine (SVM) with an RBF kernel. Hyperparameters like (C, γ) are tuned by grid search within each fold. Performance is assessed via accuracy and F1-score.

3 Experiments and Results

We conducted extensive experiments across five benchmark datasets. Due to space limitations, Table 2 reports only the best-performing configurations for each feature extraction method and dataset.
The **Quantum + Attention-Based** model consistently achieves the **highest F1-scores and CV accuracies**, especially on *MUTAG* and *AIDS*, indicating that integrating quantum observables with attention-driven features provides richer molecular representations. The **Attention-Based** method alone performs strongly on datasets with clear functional group patterns , confirming that adaptive node and bond weighting effectively captures chemical saliency. The **Chemical Laplacian** baseline performs reasonably well, particularly on smaller graphs

Table 2: Comparison of F1-score and cross-validation (CV) accuracy across feature extraction methods and datasets.

Dataset	Chemical Laplacian		Attention-Based		Quantum + Attention-Based	
	F1	CV Acc.	F1	CV Acc.	F1	CV Acc.
MUTAG	0.8	0.82	0.83	0.85	**0.90**	**0.86**
PTC-MR	0.52	0.59	0.51	0.58	**0.53**	**0.60**
NCI1			0.70	0.73	**0.73**	**0.74**
AIDS	0.84	0.95	0.98	0.98	**0.99**	**0.98**
PROTEINS			0.63	0.66	**0.60**	**0.67**

, showing that spectral topology remains informative, but lacks adaptivity to local chemical variations.

4 Conclusion and Future Works

This work presented three complementary feature extraction methods — Chemical Laplacian, Attention-Based, and Quantum-Based feature maps — for molecular graph classification. The results showed that while the chemical-aware models effectively capture molecular topology and reactivity, the hybrid **Quantum + Attention** approach achieved the best overall performance. Moreover, the strong results on the **PROTEINS** dataset highlight the potential of these chemically informed representations to generalize to other domains of graph-structured data.

5 Appendix

A. Benchmark Datasets To evaluate the performance of our proposed feature maps design methods, we employed five widely used molecular graph classification benchmarks: **AIDS**[6], **PROTEINS**[7], **NCI1**[8], **PTC-MR**[9], and **MUTAG**[10]. Each dataset represents molecules as graphs, where nodes correspond to atoms or amino acids, and edges represent chemical bonds or spatial interactions. Below is a summary of their key characteristics.

Table 3: Summary of benchmark molecular graph datasets.

Dataset	**#Graphs**	**Avg. Nodes**	**Avg. Edges**	**Target Label (Y)**	**Domain**
MUTAG	188	17	19	Mutagenic (1) / Non-mutagenic (0)	Chemistry
AIDS	2000	15	16	Active (1) / Inactive (0)	Chemistry
NCI1	4110	30	32	Active (1) / Inactive (0)	Chemistry
PTC-MR	344	25	26	Toxic (1) / Non-toxic (0)	Toxicology
PROTEINS	1113	39	73	Enzyme (1) / Non-enzyme (0)	Biology

References

[1] P. Bongini, N. Pancino, A. Bendjeddou, F. Scarselli, M. Maggini, and M. Bianchini, "Composite graph neural networks for molecular property prediction," *International Journal of Molecular Sciences*, vol. 25, no. 12, p. 6583, 2024.

[2] T. Lutchyn, M. Mardal, and B. Ricaud, "Efficient learning of molecular properties using graph neural networks enhanced with chemistry knowledge," 2025.

[3] R. Li, X. Yuan, M. Radfar, P. Marendy, W. Ni, T. J. O'Brien, and P. M. Casillas-Espinosa, "Graph signal processing, graph neural network and graph learning on biological data: a systematic review," *IEEE Reviews in Biomedical Engineering*, vol. 16, pp. 109–135, 2021.

[4] X. Zhao, B. Chen, M. Ji, X. Wang, Y. Yan, J. Zhang, S. Liu, M. Ye, and C. Lv, "Implementation of large language models and agricultural knowledge graphs for efficient plant disease detection," *Agriculture*, vol. 14, no. 8, p. 1359, 2024.

[5] H. Cai, H. Zhang, D. Zhao, J. Wu, and L. Wang, "Fp-gnn: a versatile deep learning architecture for enhanced molecular property prediction," *Briefings in bioinformatics*, vol. 23, no. 6, 2022.

[6] K. Riesen and H. Bunke, "Iam graph database repository for graph based pattern recognition and machine learning," in *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*, pp. 287–297, 2008.

[7] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. suppl_1, pp. i47–i56, 2005.

[8] N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 347–375, 2008.

[9] H. Toivonen, A. Srinivasan, R. D. King, S. Kramer, and C. Helma, "Statistical evaluation of the predictive toxicology challenge 2000–2001," *Bioinformatics*, vol. 19, no. 10, pp. 1183–1193, 2003.

[10] A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity," *Journal of medicinal chemistry*, vol. 34, no. 2, pp. 786–797, 1991.