Journal of Big Data

RESEARCH Open Access

Inverse link prediction with graph convolutional networks for knowledge-preserving sparsification in cheminformatics

Elnaz Bangian Tabrizi[1], Mehrdad Jalali[1,2,3*] and Mahboobeh Houshmand[1]

*Correspondence:
mehrdad.jalali@kit.edu

[1] Department of Computer Engineering, Ma.C., Islamic Azad University, Mashhad, Iran
[2] Institute of Functional Interfaces (IFG), Karlsruhe Institute of Technology (KIT), Eggenstein-Leopoldshafen, Germany
[3] Applied Data Science and Artificial Intelligence, SRH University Heidelberg, Heidelberg, Germany

Abstract: Large-scale cheminformatics datasets, such as those used in drug discovery and materials science, are often represented as dense similarity graphs; however, their complexity hinders scalable analysis and interpretability. We propose a novel Inverse Link Prediction (ILP) framework, powered by Graph Neural Networks (GNNs), for knowledge-preserving graph sparsification, using Metal–Organic Framework (MOF) datasets as a case study. The framework comprises four key components: (1) Graph Convolutional Networks (GCNs) to predict edge importance based on node features, (2) ILP to compute inverse weights identifying redundant edges, (3) dual-weight analysis to integrate initial similarity weights with GCN-derived weights, and (4) modularity optimization to prune edges while preserving community structures and domain knowledge. Validated on MOF similarity graphs, the sparsified graphs maintain structural integrity and support robust performance across both graph-based (GCN, GraphRAGE) and non-graph-based (Gradient Boosting Trees, Logistic Regression, Naïve Bayes, Deep Neural Networks) machine learning models for tasks such as pore limiting diameter prediction. This Inverse Link Prediction with Graph Convolutional Networks (ILP-GCN) framework offers a scalable and interpretable solution for cheminformatics, with broad applications in material discovery and beyond.

Keywords: Graph sparsification, Inverse link prediction, GCN, Metal–organic frameworks, Network analysis, Complex networks, MOF characterization, Computational material science, Energy applications

Introduction

Background and motivation

Large-scale cheminformatics datasets, encompassing molecular and material structures, are often modeled as dense similarity graphs to capture complex relationships between them [1, 2]. However, these graphs, with millions of edges, pose significant computational challenges, including high processing costs and reduced interpretability, limiting scalable analysis. In many scientific domains, especially cheminformatics, data is represented using dense similarity graphs that capture intricate structural and chemical

relationships. Metal–Organic Frameworks (MOFs) are a fascinating class of porous materials characterized by their diverse structures and intricate coordination between metal ions or clusters and organic ligands [3, 4]. These materials have garnered significant interest due to their potential applications in various fields, such as gas storage, catalysis, and environmental remediation. The ability to accurately predict MOF properties is critical, given the vast number and complex nature of MOF structures, which introduce significant challenges regarding computational time and accuracy [5, 6].

The advent of machine learning methods has revolutionized the predictive modeling of MOF properties by leveraging computational algorithms to interpret and learn from MOF data [5]. Graph Neural Networks (GNNs) have emerged as powerful tools for capturing complex patterns in network structures by integrating node attributes and topological features [7]. Our prior work addressed some of these challenges by developing MOFGalaxyNet, a novel framework that constructs a social network based on MOF data [1]. This method has enabled us to expedite the prediction of MOF properties by simplifying the intricate network of MOF interactions, similar to our previous work where we utilized it to predict gas adsorption [2]. However, a significant limitation was encountered: only a fraction of the available MOF data could be mapped due to the extensive size of the generated social network. This limitation necessitates exploring more advanced methods to refine and expand upon our initial model, ensuring that a broader array of MOF data can be efficiently processed and analyzed.

As social networks grow in size and complexity, methods for simplifying them while keeping their key features intact are urgently needed [8]. Sparsification has become a crucial technique in this area, focusing on reducing connections within a network without changing its core structure. This approach improves data management and makes network analysis tasks more manageable. Our research builds upon foundational work in sparsification and has demonstrated that networks can be simplified without compromising their essential spectral characteristics [9]. This method has led to numerous valuable applications in graph algorithms, underscoring the importance of preserving specific mathematical properties of networks, such as the eigenvalues and eigenvectors of the graph Laplacian. Building on this research, we investigate how sparsification can be applied to large social networks, highlighting its role in making networks easier to navigate and maintain. This approach involves innovative methods, such as utilizing advanced computational models to preserve essential network features while reducing complexity. Recent work [10] has introduced a machine learning-enhanced molecular dynamics (MD) framework for predicting solvent-accessible surface area (SASA) in nanoparticle drug delivery systems, aligning with our sparsification approach by integrating learning-based surrogates to preserve key structural properties in large-scale graphs. This study significantly broadened our investigation by utilizing a large Cambridge Structural Database (CSD) dataset, containing approximately 14,000 metal–organic frameworks (MOFs), which allowed us to delve more deeply into sparsification within our network model, MOFGalaxyNet [11].

Problem statement and contribution

The growth of large social networks necessitates efficient techniques such as sparsification, which simplify network complexity by strategically removing redundant

connections while preserving essential structural characteristics [12]. This process not only aids in efficient data storage but also enhances computational performance across various network analysis tasks [13]. One of the foundational works in this area introduced spectral sparsification, demonstrating that networks could be simplified while preserving their spectral properties [9]. Their work paved the way for numerous applications in graph algorithms, emphasizing the importance of maintaining the eigenvalues and eigenvectors of the graph Laplacian. By further advancing this field, sparsification in the context of large social networks was explored, underscoring the technique's utility in maintaining network navigability and community structure [14]. Their empirical investigations demonstrated that sparsified networks could effectively replicate the original network's community detection and shortest path calculations, thereby affirming the technique's value in social network analysis. Recent studies continue to push the boundaries of sparsification in large social networks, shedding light on innovative approaches and the multifaceted benefits of these techniques. For instance, the authors [15] explored the location of social influence sources based on network sparsification and stratification, introducing a novel method to preserve the distribution of node degrees in sparsified networks. Additionally, another study introduced a novel method for graph sparsification utilizing generative adversarial networks, illustrating how sparsified networks can retain essential structural and functional characteristics [16].

Graph sparsification aims to reduce the size of a graph while preserving essential properties relevant to tasks such as graph convolutional networks (GCN), thus improving the efficiency of the GCN algorithm [17, 18]. The inGRASS algorithm efficiently sparsifies large graphs incrementally, with nearly linear setup time and O(logN) update time per change, employing a multilevel resistance embedding framework to identify critical edges and detect redundancies [19]. However, identifying a suitable sparsification method for GCNs is challenging due to trade-offs between preserving different graph properties. The authors [20] evaluated 12 sparsification algorithms across 16 graph metrics on 14 real-world graphs, shedding light on their effectiveness in preserving properties crucial for Graph Convolutional Network (GCN) performance. These findings underscore the importance of aligning sparsification methods with GCN algorithms and propose a framework for ongoing evaluation and optimization. Traditional methods, such as Edge Betweenness Centrality, often discard critical chemical and topological information, thereby hindering applications like material discovery [8].

Current techniques, including node importance estimation and density-based clustering, are not inherently designed for sparsification, often resulting in increased computational costs or failing to support predictive workflows [21–24]. Fuzzy clustering-based deep learning models, such as FDAGC and FCGCN, enhance clustering quality by incorporating fuzzy logic into GNNs, thereby addressing oversmoothing issues but not directly addressing sparsification [25, 26]. Evaluating sparsified graphs involves assessing their ability to approximate original structures while maintaining efficiency, with methods such as graph denoising and optimization-based estimators showing promise [27, 28].

Large-scale cheminformatics datasets, represented as dense similarity graphs, pose significant computational challenges due to their millions of results, resulting in high processing costs and reduced interpretability [1]. This study addresses these challenges by proposing an Inverse Link Prediction (ILP) framework using GCNs

(ILP-GCN) within MOFGalaxyNet. Unlike traditional link prediction, which forecasts future links [14, 29], ILP identifies redundant edges for removal without disrupting network integrity, thereby optimizing storage and computation [30–32]. Applied to a CSD dataset of approximately 14,000 MOFs, ILP-GCN selectively prunes edges while preserving community structures, enhancing scalability and interpretability for tasks like guest accessibility prediction [1, 33]. Figure 1 illustrates the general architecture of ILP-GCN, showing the process from input graph to sparsified graph via GCN encoder and inverse link prediction.

In the following sections, we detail our proposed ILP-based sparsification framework and demonstrate its effectiveness on large-scale MOF similarity graphs.

ILP-based sparsification framework

In this study, we introduce an innovative approach to elucidate the intricate network of MOFs through the development of MOFGalaxyNet, which was previously introduced in our work [1] and is powered by graph sparsification via inverse link prediction (ILP) and Graph Convolutional Network (GCN). The general architecture of the method is illustrated in Fig. 2, which begins with the comprehensive collection and preprocessing of data from the Cambridge Structural Database, followed by the vectorization of more than 14,000 MOFs based on their unique structural and chemical properties. This information forms an adjacency matrix encapsulating the relationships within MOFGalaxyNet, which is subsequently visualized to highlight essential structural hubs. We apply ILP with GCN models to identify and remove less critical links, refining the network while preserving its structural integrity. The sparsification process is meticulously evaluated to ensure that the essential characteristics of the network are maintained.

Furthermore, we utilized machine learning methods to predict key MOF properties, incorporating the pore-limiting diameter (PLD) as a critical feature. Sparsified datasets were employed to validate the accuracy and reliability of these predictive models. This streamlined and computationally efficient dataset, the Pure MOF dataset, is thus prepared for advanced material characterization, setting a new paradigm at the intersection of complex network analysis and computational material science, particularly for applications targeting environmental and energy challenges. Subsequent sections will elaborate on the comprehensive details of this methodology.

Generating MOFGalaxyNet

MOF Data Preparation The dataset consists of 14,296 MOFs, each characterized by unique SMILES strings for organic linkers [34] and atomic descriptors for metal ions.

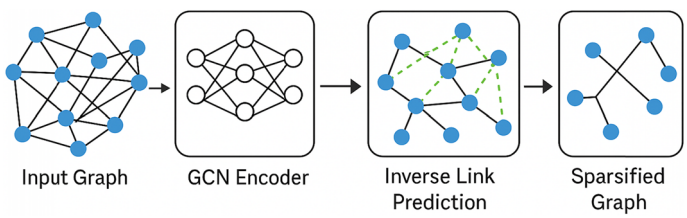


Fig. 1 General architecture of ILP-GCN, illustrating the process from Input Graph through GCN Encoder and Inverse Link Prediction to Sparsified Graph

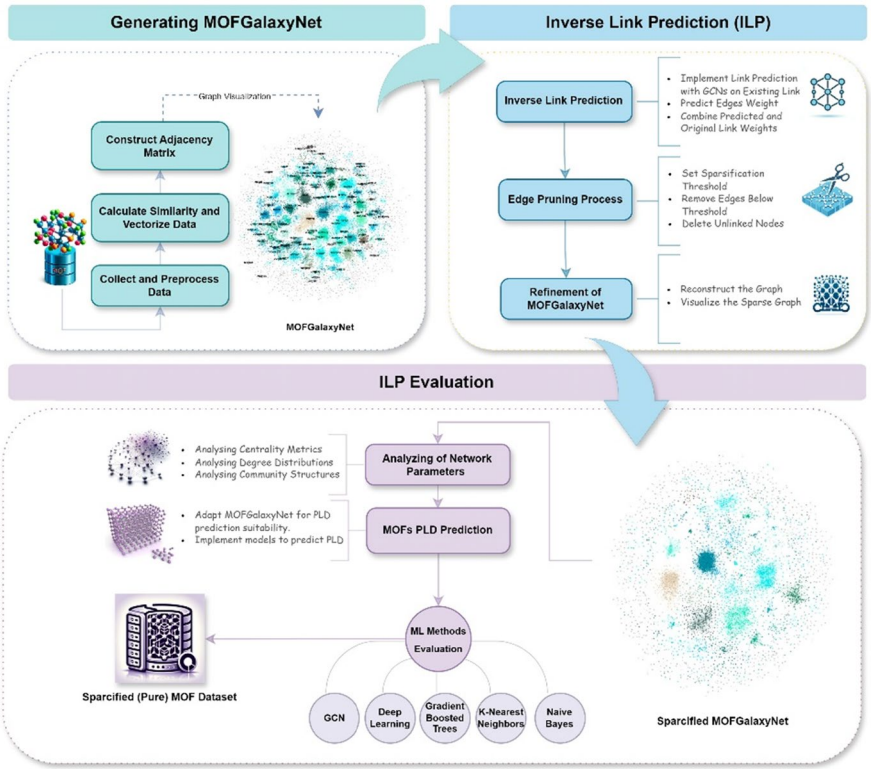


Fig. 2 Overview of the proposed ILP-GCN framework. The system starts with a dense MOF similarity graph, followed by edge scoring via a GCN-based model. Inverse link prediction identifies weak or redundant edges, which are pruned to generate a sparse graph. The resulting graph is evaluated for structural integrity and downstream predictive performance

We refined our vectorization and similarity calculation methods to accommodate this dataset, enhancing the network's construction and predictive analysis.

Vectorization of MOF Data Vectorization transforms MOF descriptors into a numerical format suitable for machine learning analysis. For each MOF, a vector v is constructed as follows:

$$v = (SMILES, AN, AW, AR, ME, P, EA) \tag{1}$$

where SMILES represents the organic linker structure. AN (atomic number), AW (atomic weight), AR (atomic radius), ME (mulliken electronegativity), P (polarizability), and EA (electron affinity) represent the atomic descriptors of metal ions [6]. Each descriptor contributes to the multidimensional vector v, encapsulating the structural and chemical properties of the MOF.

Dataset Summary The dataset features a broad diversity, covering 53 distinct metal types and over 3000 unique linkers, together with key geometric descriptors such as the Pore Limiting Diameter (PLD), Largest Cavity Diameter (LCD), and Largest Free Sphere (LFS). To ensure data quality, MOFs with missing information or non-physical pore characteristics were removed, and all continuous features were normalized using Min–Max scaling. The detailed distribution of structural properties and preprocessing statistics are provided in Section S5 of ESI.

Similarity Calculation The similarity between MOFs is a critical step in constructing the adjacency matrix for MOFGalaxyNet. We calculated similarity using two distinct approaches for linkers and metals.

In the context of metal–organic frameworks (MOFs), organic linkers are crucial components that connect metal ions or clusters to form the framework's structure. These linkers are typically represented using SMILES codes, a notation that encodes the molecular structure as a string of characters. The SMILES code provides a compact and human-readable way to describe the arrangement of atoms within a molecule.

For example, the organic linker benzene-1,4-dicarboxylic acid (BDC) is considered a common ligand in MOF construction. Its SMILES code is "O=C(O)c1 ccc(cc1)C(=O) O". This code succinctly represents the molecular structure, where "O=C(O)" denotes the carboxylic acid group, "c1 ccc(cc1)" represents the benzene ring, and the entire expression outlines the connectivity of these components within the molecule.

When assessing the similarity between two MOFs based on their organic linkers, the SMILES strings of the linkers are utilized to generate molecular fingerprints through the Morgan fingerprint method. These fingerprints are binary vectors that capture the presence or absence of certain structural features within a molecule. The similarity score, SIM_{Linker}, between two MOFs can then be calculated by comparing their fingerprints:

$$SIM_{Linker}\left(MOF_i, MOF_j\right) = \frac{Fingerprint(MOF_i).Fingerprint\left(MOF_j\right)}{\|Fingerprint(MOF_i)\|.\|Fingerprint\left(MOF_j\right)\|} \tag{2}$$

This calculation essentially measures how similar the structural features of two linkers are, with a higher score indicating greater similarity. The use of SMILES codes and Morgan Fingerprints enables a detailed and efficient comparison of organic linkers within MOFs, facilitating the identification of MOFs with similar properties for further analysis or application[35].

The similarity between the metal ion descriptors of two MOFs is determined using the cosine similarity, denoted as SIM_{Metal}:

$$SIM_{Metal}\left(MOF_i, MOF_j\right) = \frac{V_{Metal_i}.V_{Metal_j}}{\left\|FV_{Metal_i}\right\|.\left\|V_{Metal_j}\right\|} \tag{3}$$

Where V_{Metal_i} and V_{Metal_j} are the vector representations of the metal ions in MOF i and MOF j, respectively.

The adjacency matrix A is a fundamental component in constructing MOFGalaxyNet and represents the connections (edges) between nodes (MOFs) based on their similarities. The process of forming A incorporates the calculated similarities between MOF pairs, both for organic linkers and metal ions, into a unified framework. Here, we detail the mathematical formulation for generating A in the context of the expanded MOFGalaxyNet.

Given a set of MOFs,$\{MOF_1, MOF_2, \ldots, MOF_N\}$, where N is the total number of MOFs in the dataset, the adjacency matrix A is an $N \times N$ matrix where each element A_{ij} represents the edge weight (similarity) between the MOF i and MOF j. The edge weight is calculated as follows:

$$A_{ij} = SIM_{MOF}\left(MOF_i, MOF_j\right) \tag{4}$$

where $SIM_{MOF}(MOF_i, MOF_j)$ is the overall similarity between two MOFs, defined as a weighted combination of their linker and metal similarities:

$$A_{ij} = SIM_{MOF}(MOF_i, MOF_j) = \alpha \times SIM_{Linker}(MOF_i, MOF_j) + (1-\alpha) \times SIM_{Metal}(MOF_i, MOF_j) \quad (5)$$

To refine A and mitigate the complexity of MOFGalaxyNet, we apply a thresholding operation. All of the elements in A below a specified similarity threshold $\varnothing$ are set to zero, effectively removing weak links from the network:

$$\begin{cases} A_{ij} if A_{ij} \geq \varnothing \\ 0 otherwise \end{cases} \quad (6)$$

The resulting matrix serves as the adjacency matrix for the enhanced MOFGalaxyNet, encapsulating the significant relationships within the MOF dataset. This matrix underpins the network's structure, facilitating subsequent analyses, including centrality measures and community detection, to unravel the intricate web of MOF similarities and interactions. In Algorithm 1, the pseudocode of the process is illustrated. This pseudocode outlines an algorithm for constructing a graph-based representation of MOFs based on the similarity of their organic linkers. This approach involves initializing structures, loading MOF data, generating fingerprints for similarity comparison, calculating similarity scores, applying a threshold to these scores, and constructing a graph where edges represent significant similarities.

Algorithm 1: Pseudocode for graph-based MOF representation, detailing data loading, similarity fingerprinting, and threshold-based graph construction.

```
Input: MOFDataset, SimilarityThreshold φ
Output: Graph G representing MOF similarities

1. Initialize:
  - Create an empty list for MOFs
  - Create an empty adjacency matrix A

2. Load MOF Data:
  For each MOF in the dataset:
    - Extract the SMILES code for the organic linker
    - Store the SMILES code and associated MOF metadata in the MOFs list

3. Generate Fingerprints:
  For each MOF in MOFs list:
    - Generate the Morgan Fingerprint from the SMILES code
    - Update the MOF data in the list to include its fingerprint

4. Calculate Similarity Scores:
  For i = 0 to length(MOFs) - 1:
    For j = i + 1 to length(MOFs):
      - Calculate SIM_Linker using the fingerprints of MOF[i] and MOF[j]
      - Update the adjacency matrix A[i][j] and A[j][i] with the calculated similarity score

5. Apply Thresholding to Adjacency Matrix:
  - Define a similarity threshold φ
  For each element A[i][j] in the adjacency matrix:
    If A[i][j] < φ:
      - Set A[i][j] = 0 (removing weak links)

6. Construct Graph from Adjacency Matrix:
  - Initialize an empty graph G
  For each nonzero element A[i][j] in the adjacency matrix:
    - Add an edge between MOF[i] and MOF[j] to graph G with weight A[i][j]

7. Output the Graph:
  - Return graph G representing the network of MOFs based on linker similarity
```

In our study, we constructed a MOFGalaxyNet graph to explore the intricate relationships among MOFs based on linker structure similarities, applying a stringent similarity threshold of 0.9 derived from previous work [1]. This thresholding yielded a graph with 12,561 nodes and 414,650 edges, as summarized in the network specifications (see Table 1), reflecting a dense network with an average degree of 66.022 and highlighting the substantial structural resemblances among MOFs. Notably, MOFGalaxyNet exhibited a high modularity of 0.834, revealing the presence of 876 distinct communities within the network, which underscores the diversity and specialization among MOF structures. Furthermore, network characteristics such as an average path length of 2.979 and a network diameter of 16, alongside an average weighted degree of 114.965, suggest a "small-world" phenomenon with efficient connectivity and the existence of central MOFs that may play pivotal roles within their communities. The degree distribution of the network, illustrated in the log–log plot (see Fig. 3), further corroborates the "small-world" nature of the network, demonstrating the skewness and scale-free properties that characterize the network. Structural analysis of MOFGalaxyNet, therefore, not only sheds light on the complex interrelationships between different MOFs but also opens avenues for targeted material design and the discovery of novel MOF applications by navigating the vast landscape of MOF structures.

Following the construction of MOFGalaxyNet, we visualized an intricate network to elucidate the diverse communities of MOFs based on their similarities. The graph, represented in Fig. 4, employs a color-coded scheme to differentiate between the myriad MOF communities; each color demarcates a distinct cluster. These clusters emerged through a rigorous application of our similarity threshold, set to capture the nuanced relationships that define each community within the broader MOF landscape.

The graphical representation showcases a network marked by densely interconnected nodes, indicating central hubs. These pivotal hubs suggest that the structural or functional dynamics of MOFs may significantly determine their community. The gradation from a highly connected core to less dense peripheries illustrates the network's connectivity spectrum, highlighting MOFs ranging from highly interlinked to those with distinctive or specialized features.

Distinct coloration for each community provides visual clarity and establishes a preliminary taxonomy, stratifying the expansive MOF domain into more manageable subdomains for further exploration. This color-based stratification is instrumental for researchers, allowing for identifying specific communities that merit further analysis, potentially revealing shared properties or guiding the design of novel MOF materials.

Table 1 Network Specifications of MOFGalaxyNet

Specification	Value
Nodes	12,561
Edges	414,650
Similarity threshold	0.9
Average degree	66.022
Modularity	0.834
Communities	876
Average path length	2.979
Network diameter	16
Average weighted degree	114.965

The visualization corroborates the 'small-world' network trait, underscoring efficient and concise connectivity within MOFGalaxyNet, a characteristic inferred from our prior statistical analysis. For material scientists, this graph provides a strategic vantage point for the MOF universe, accentuating pivotal nodes that could be instrumental in advancing the synthesis of innovative MOFs or refining existing ones for diverse applications.

As we transition from the initial complex structure of MOFGalaxyNet, the next phase of our study addresses the challenges of size and intricacy. The dense network, while comprehensive, necessitates a refinement process—graph sparsification using inverse link prediction (ILP). This method carefully trims the network, preserving its fundamental topology and easing the analysis. Implementing ILP ensures that the network's core attributes are maintained for more efficient navigation and investigation. The forthcoming section will detail this sparsification approach, paving the way for an enhanced understanding of MOF interconnectivity.

Theory definition: graph sparsification using inverse link prediction (ILP)

In this work, we define inverse link prediction (ILP) as a rigorous methodology designed for the sparsification of complex graphs, symbolized by $G = (V, E)$, where V and E denote the sets of vertices and edges, respectively. The crux of ILP lies in its strategic focus on the discriminative removal of edges from G, predicated on a predictive model that assesses the significance of each edge, e_{ij}, in maintaining the graph's structural cohesion and functional efficacy. This approach employs the predictive ability of GCN to evaluate the importance of connections within the graph:

$$S_{GCN}\left(e_{ij}\right) = GCN(v_i, v_j|\theta) \tag{7}$$

The prediction score $S_{GCN}\left(e_{ij}\right) =$ derived from the GCN model offers a deep learning-based perspective on edge significance, complementing the traditional ILP approach. These scores are inversely used to compute the ILP-GCN-derived weights, which quantify the importance of edges based on both ILP and GCN insights:

$$W_{ILP-GCN}\left(e_{ij}\right) = \frac{\alpha}{S_{GCN}\left(e_{ij}\right) + \varepsilon} \tag{8}$$

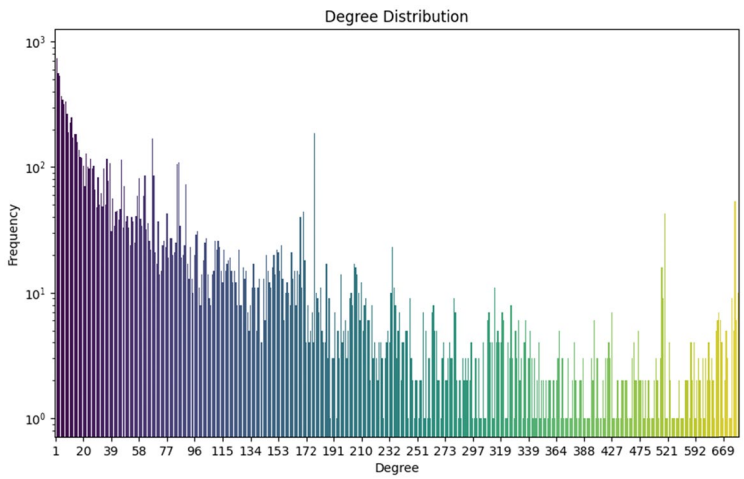


Fig. 3 Log–log plot of the degree distribution in MOFGalaxyNet

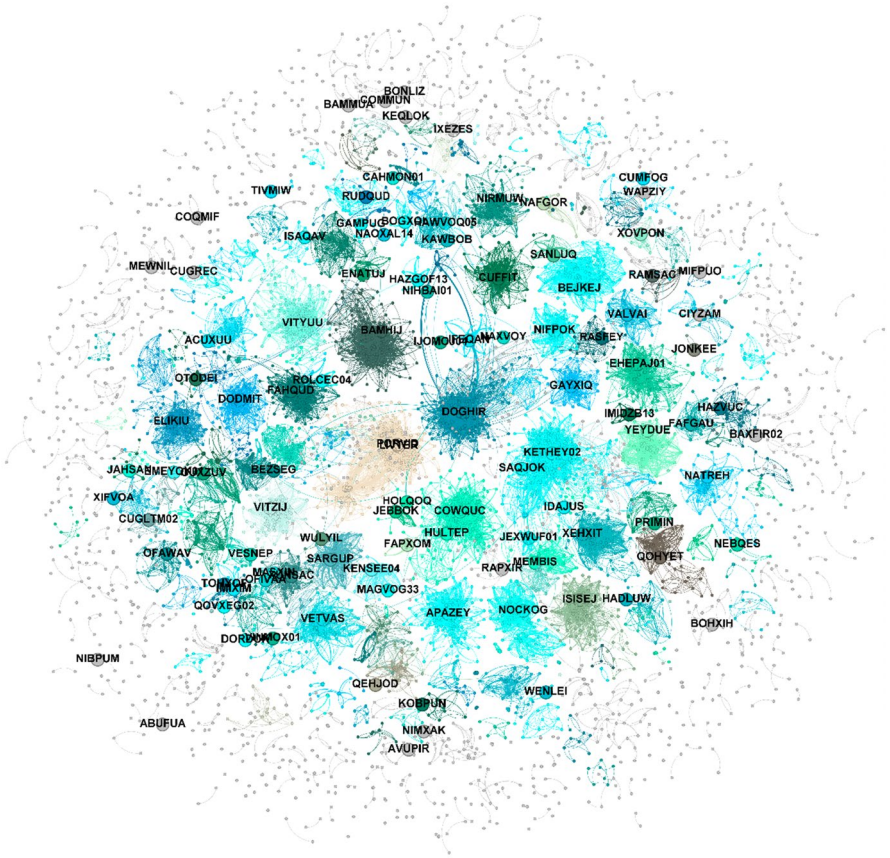


Fig. 4 Visualization of MOFGalaxyNet depicting the network of MOFs based on structural and compositional similarities. Each color represents a distinct community within the network, highlighting the dense interconnectivity and the central hubs of activity. The variation in node saturation and edge density reflects the degree of connectivity, illustrating the spectrum from highly interlinked core MOFs to peripheral frameworks with unique attributes. This graph serves as a strategic tool for identifying key MOF materials and their potential roles within the network

Where α is a normalization factor and ε is a small constant to ensure numerical stability.

Dual-Weight Analysis for Comprehensive Edge Assessment The dual-weight analysis integrates initial graph weights,$W_{initial}(e_{ij})$, representing inherent edge significance, with ILP-GCN-derived weights,$W_{ILP-GCN}(e_{ij})$, reflecting both predictive and deep learning insights. This analysis yields a final weight for each edge,$W_{final}(e_{ij})$, which guides the sparsification process:

$$W_{final}\left(e_{ij}\right) = \gamma . W_{initial}\left(e_{ij}\right) + (1-\gamma) . W_{ILP-GCN}\left(e_{ij}\right) \tag{9}$$

The parameter γ balances the contributions of the initial and enhanced ILP-GCN-derived weights, ensuring a nuanced approach to edge retention.

Modularity Optimization for Sparsification This methodology employs modularity optimization to evaluate the contribution of each edge to the overall community structure of the graph utilizing the final composite weights $W_{final}\left(e_{ij}\right)$. This step prioritizes the preservation of edges critical for maintaining the integrity of the graph's community structure:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij}.W_{final}\left(e_{ij}\right) - \frac{\left(k_i.W_{ki-GCN}\right)\left(k_j.W_{kj-GCN}\right)}{2m} \right] \delta\left(c_i, c_j\right) \tag{10}$$

In this context, A_{ij} represents the adjacency matrix indicating edge presence between nodes i, $W_{final}\left(e_{ij}\right)$ is the final weight of edge e_{ij} after applying the inverse link prediction (ILP) enhanced by graph convolutional network (GCN) insights,k_i and k_j denote the degrees of nodes i and j, respectively, weighted by GCN-derived insights W_{ki-GCN} and W_{kj-GCN}, and $\delta\left(c_i, c_j\right)$ is the Kronecker delta function, signifying node pair i and j membership in the same community.

By maximizing Q, we aim to identify the graph partitioning that best preserves dense intracommunity connections while reducing complexity by removing edges with the lowest modularity contribution. The algorithm iteratively removes edges to optimize the network's modularity Q. The process halts when further removals fail to improve the modularity, indicating that the desired sparsity level has been reached. This level reflects the optimal balance between reducing graph complexity and preserving essential network properties.

Based on the final edge weight formulation $W_{final}\left(e_{ij}\right)$ described in Eq. (9), we introduce a sparsification threshold $\phi \in [0, 1]$ such that edges for which $W_{final}\left(e_{ij}\right) < \phi$ are pruned from the graph. The choice of ϕ is not arbitrary but is determined by analyzing how global graph metrics evolve with increasing threshold values. In particular, we observe that modularity $Q(\phi)$, a measure of community structure, initially increases with ϕ as weak and noisy links are removed, then declines as structurally important edges are pruned. This behavior suggests a unimodal trend for $Q(\phi)$. Accordingly, the optimal threshold is ϕ^* selected where $Q(\phi)$ reaches its maximum or begins to plateau, thus balancing the trade-off between network simplification and the preservation of essential community structures.

To further justify this choice, we systematically analyzed the evolution of modularity and other network metrics across thresholds (see Table S2 and Figure S2 in the ESI). The selected thresholds correspond to points where modularity maximizes or remains stable while still enabling significant sparsification, confirming the effectiveness and robustness of this threshold selection strategy.

Empirical results from Table 2 support this rationale, showing that modularity improves consistently even with substantial reductions in edge count, confirming the robustness of this threshold selection criterion.

By implementing this sparsification threshold, we not only streamline the graph but also ensure that the removal process is informed by a deep learning-based understanding of edge significance, thereby enhancing the robustness and relevance of our sparsified model. Throughout the paper, we refer to this term numerous times to emphasize its foundational role in our sparsification approach, underscoring its importance in aligning with the overarching goals of preserving essential topological features and optimizing the graph's community structure as measured by Q.

Through this ILP framework, as enriched by GCN insights, we achieve a sparsified graph $\acute{G} = \left(\acute{V}, \acute{E}\right)$ where $\acute{E} \subset E$ and $\acute{V} \subset V$. This subset $V\prime$ includes only those vertices connected by the edges in $E\prime$, effectively maintaining the essential topological

features of the graph amidst reduced complexity. This approach ensures that any vertices not connected by the sparsified edges are also removed, streamlining the graph further. This theoretical underpinning provides a robust foundation for our novel approach to graph sparsification, exemplified within our MOFGalaxyNet model. This highlights the transformative potential of ILP in simplifying complex networks without sacrificing their intrinsic value.

It is important to note that the GCN model employed for inverse link prediction (ILP) during the sparsification process is distinct from the GCN model used later in the prediction phase. In the ILP stage, the GCN is utilized specifically for predicting and consequently removing non-essential links within the network, which is crucial for effective sparsification. Conversely, in the prediction phase, we employ a GCN to predict MOF properties from the sparsified data, and this GCN model is then compared directly with other machine learning methods. This distinction ensures that the GCN's role in sparsification does not bias the subsequent prediction results.

Figure 5 shows an example of a complex network that leverages graph convolutional network (GCN) methodologies and inverse link prediction (ILP) for enhanced link prediction and graph sparsification. Figure 5a shows the initial state of MOFGalaxyNet with edges assigned weights that indicate similarities between MOFs. Transitioning to Fig. 5b, we implement GCN algorithms to adjust the edge weights around a specified node, MOF5—distinctly colored in red, indicating it as the analytical nucleus with its neighbors highlighted in green. In Fig. 5c, MOFGalaxyNet undergoes dual-weighting integration, merging the original and GCN-calculated weights and maintaining a focus on MOF5 and its immediate network. The final visualization, shown in Fig. 5d, demonstrates the network after applying ILP, which selectively prunes less significant edges and effectively simplifies MOFGalaxyNet while preserving its essential topological features. This series exemplifies the capabilities of ILP within our proposed MOFGalaxyNet, demonstrating its utility in reducing network complexity and highlighting critical relational structures centered on targeted nodes.

The integration of ILP and GCN in our sparsification framework is motivated by the limitations of traditional methods such as Edge Betweenness Centrality (EBC), which rely solely on structural heuristics and often neglect the rich chemical and topological features embedded in MOF data. While such methods can identify central or redundant edges based on topology, they fail to leverage feature-driven patterns critical to material science applications. GCN, on the other hand, learns meaningful node embeddings that capture both local and global structural properties as well as chemical descriptors. By using these embeddings within ILP, we enable a data-driven assessment of edge importance that transcends simplistic heuristics. This

Table 2 Network Sparsification Effects of Increasing Sparsification Thresholds in MOFGalaxyNet

Specification	Original	Threshold 0.90	Threshold 0.95	Threshold 0.98
Nodes	12,561	8999	6559	2153
Edges	414650	38365	12548	1920
Average degree	66.02	8.53	3.83	1.78
Modularity	0.832	0.85	0.87	0.94
Communities	871	686	748	574
% Reduction in nodes	–	28.36%	47.78%	82.86%
% Reduction in edges	–	90.75%	96.97%	99.54%
% Reduction in average degree	–	87.07%	94.20%	97.31%

combination ensures that sparsification is not only efficient but also chemically and structurally informed. As a result, we preserve domain-relevant community structures, improve interpretability, and significantly reduce complexity without compromising the network's predictive utility. This synergy between ILP and GCN represents a novel approach in MOF network analysis, offering a scalable and scientifically grounded framework for sparsification.

In addition to the theoretical formulation, we provide the complete implementation of the ILP-based sparsification procedure as Algorithm 2. This algorithm takes as input the MOF graph, node feature matrix, and relevant hyperparameters, and outputs the sparsified graph alongside the list of removed edges and final computed edge weights. The process consists of (i) training a GCN-based link predictor, (ii) computing inverse link prediction (ILP) weights, (iii) integrating initial and predicted weights, and (iv) optimizing the graph modularity during the sparsification phase. The full pseudo-code and detailed steps of the algorithm are provided in the ESI (Section S3). The time complexity of the overall sparsification procedure is formally analyzed as $O\left(e.n.f.h + m.f.h + \mid \mathrm{T} \mid \cdot \mathrm{m} \cdot \log \mathrm{n}\right)$ where e is the number of training epochs, n is the number of nodes, f is the node feature dimension, h is the GCN hidden size, mmm is the number of edges, and $\mid T \mid$ is the number of tested thresholds. This ensures scalability for large MOF graphs, as confirmed by our practical runtime results (see Table S1 and Section S4 of the ESI).

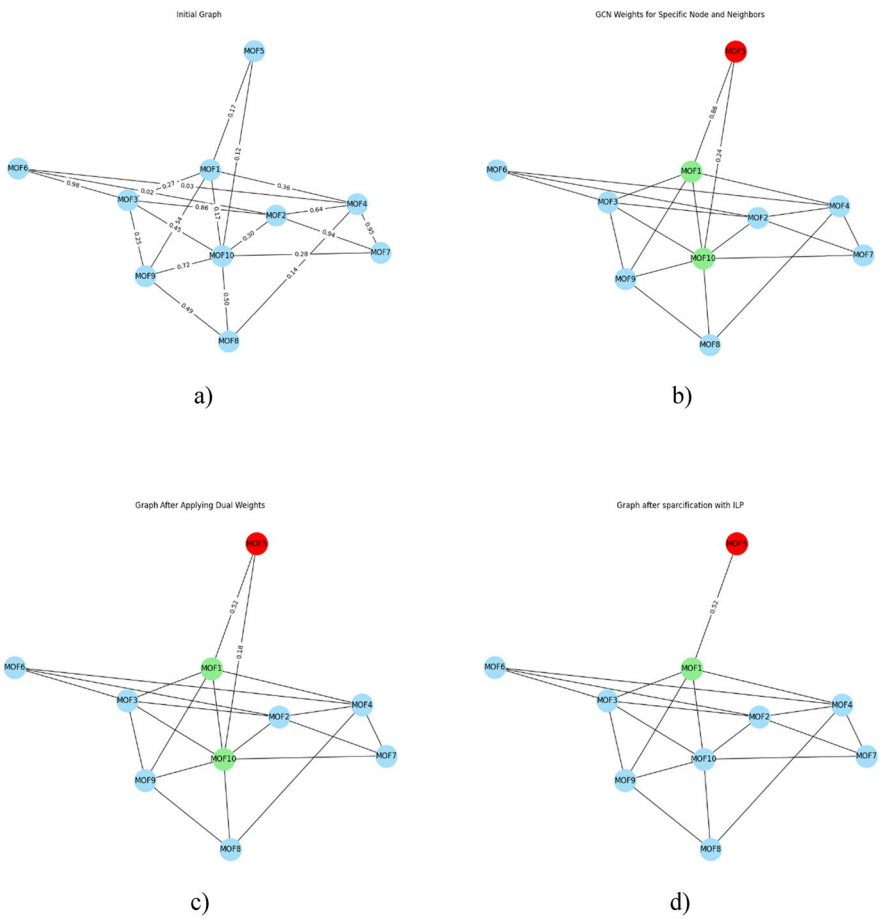


Fig. 5 Evolution of MOFGalaxyNet: **a** initial weights, **b** GCN adjustments around MOF5, **c** dual-weight integration, **d** post-ILP sparsification

Algorithm 2 Pseudo-code of the ILP-based sparsification framework combining GCN-driven link prediction, inverse weight computation, and modularity-guided edge pruning for constructing knowledge-preserving sparsified graphs from MOF similarity networks.

Input:
- **Graph G with nodes V and edges E** // **Input graph, undirected, representing MOF similarities**
- **Node features X** // **Matrix of node features for GCN input**
- **Small constant ε** // **Small positive scalar to prevent division by zero**
- **Normalization factor α** // **Scalar to scale ILP-GCN weights**
- **Balance factor γ** // **Scalar (0 ≤ γ ≤ 1) to weigh initial vs. ILP-GCN weights**
- **Thresholds T** // **List of sparsification thresholds to test**

Output:
- **Sparsified Graph G'** // **Reduced graph after sparsification**
- **List of Removed_Edges** // **List of edges pruned during sparsification**
- **Final weights W_final** // **Final edge weights for remaining edges in G'**

1. Initialize:
- Load graph G with nodes V (|V| = n) and edges E (|E| = m)
  // G is a NetworkX graph; n is number of nodes (MOFs), m is number of edges (similarities)
- Define ε, α, γ, and thresholds T (|T| = number of thresholds)
  // ε (float), α (float), γ (float) are scalars; T is a list of floats (e.g., [0.9, 0.95, 0.98])
- Initialize Removed_Edges = []
  // List to store edges removed during sparsification; initially empty
- Extract node features X (n × f, where f = feature dimension)
  // X is a 2D float array (n rows, f cols); e.g., f = 1027 (fingerprints + metal + geometric features)

2. Train GCN Model for Link Prediction:
- Split E into training (E_train) and validation (E_val) sets
  // E_train and E_val are lists of edge tuples (v_i, v_j); e.g., 80% train, 20% validation
- Initialize GCN model with parameters Θ (e.g., l layers, h hidden units)
  // Θ is a set of weight matrices (e.g., f × h, h × h, h × 1); l = layers, h = hidden units (e.g., 64)
- For each epoch e in E_epochs (total e epochs):
  // e is an integer (e.g., 100); loops over training iterations
  - Forward pass: Compute predictions S_GCN for E_train using X and adjacency A
    // S_GCN is a float array of scores; A is n × n adjacency matrix from G
  - Backward pass: Optimize Θ using loss on E_train
    // Updates Θ via gradient descent; loss = binary cross-entropy
  - Validate on E_val
- Time Complexity: O(e * n * f * h)
  // e = epochs, n = nodes, f = feature dim, h = hidden units; scales with node count and feature size

3. Compute GCN-based Link Prediction Scores: // Part of Edge Weight Calculation
- For each edge e_ij in E:
  // e_ij is a tuple (v_i, v_j); loops over all m edges
  - S_GCN(e_ij) = GCN(v_i, v_j | Θ, X)
    // S_GCN(e_ij) is a float (0 to 1) predicting edge likelihood; uses node features X[v_i], X[v_j]
- Time Complexity: O(m * f * h)
  // m = edges; computes GCN prediction per edge, linear in edge count

4. Compute ILP-GCN-derived Weights: // Part of Edge Weight Calculation
- For each edge e_ij in E:
  // Loops over m edges to compute inverse weights
  - I(e_ij) = 1 / (S_GCN(e_ij) + ε)
    // I(e_ij) is a float; inverse of GCN score for ILP, ε ensures stability
  - W_ILP-GCN(e_ij) = α / (S_GCN(e_ij) + ε)
    // W_ILP-GCN(e_ij) is a float; normalized inverse weight
- Time Complexity: O(m)
  // Linear in m; simple arithmetic per edge

5. Integrate Initial Graph Weights:
- For each edge e_ij in E:
  // Loops over m edges to retrieve initial weights
  - Retrieve W_initial(e_ij) from G
    // W_initial(e_ij) is a float from G's edge data (e.g., similarity score)

6. Compute Final Weights for Edge Retention:
- For each edge e_ij in E:
  // Loops over m edges to combine weights
  - W_final(e_ij) = γ * W_initial(e_ij) + (1 - γ) * W_ILP-GCN(e_ij)
    // W_final(e_ij) is a float; weighted combination for sparsification decision

7. Optimize Modularity for Sparsification:
- Compute initial modularity Q of G using W_final
  // Q is a float; measures community structure quality using greedy algorithm
- For each threshold t in T:
  // t is a float (e.g., 0.9); loops over |T| thresholds
  - G_temp = G.copy()
    // G_temp is a NetworkX graph; temporary copy for testing threshold
  - For each edge e_ij in E:
    // Loops over m edges to check against threshold
    - If W_final(e_ij) < t:
      - Temporarily remove e_ij from G_temp
  - Recalculate Q_temp for G_temp
    // Q_temp is a float; recomputes modularity on sparsified graph
  - If Q_temp > Q:
    - Permanently update G_temp, add e_ij to Removed_Edges
    - Q = Q_temp
  - Else:
    - Discard changes to G_temp
- Time Complexity: O(|T| * m * log(n))
  // |T| = number of thresholds; modularity computation per threshold scales with edges and log(nodes)

8. Output Sparsified Graph:
- G' = G_temp after final threshold
  // G' is the final NetworkX graph after sparsification
- Return G', Removed_Edges, W_final
  // Returns graph, edge list, and weights

Overall Time Complexity for Key Steps: O(e * n * f * h + m * f * h + |T| * m * log(n))

Results

ILP network sparsification across different thresholds

In this study, we systematically analyzed the effect of graph sparsification using Inverse Link Prediction (ILP) across a wide range of sparsification thresholds. As detailed in Table S2 of the Electronic Supplementary Information (ESI), we investigated thresholds from 0.0 up to 0.99 in increments of 0.01, providing a comprehensive view of how key network properties—including node and edge counts, average degree, modularity, and the number of communities—evolve under progressive sparsification.

For clarity and conciseness, we selected three representative thresholds: 0.90, 0.95, and 0.98, as summarized in Table 2. These thresholds were chosen to illustrate three distinct stages of the sparsification process:

- At threshold 0.90, the network is reduced to 8999 nodes and 38,365 edges, corresponding to reductions of approximately 28.4% and 90.8% in nodes and edges, respectively. The modularity increased from 0.83 to 0.85, reflecting the emergence of more defined community structures.
- Increasing the threshold to 0.95 results in a further reduction to 6559 nodes and 12,548 edges, leading to reductions of 47.8% and 97.0% in nodes and edges, respectively. The modularity continues to improve to 0.87, indicating enhanced community separation.
- Finally, at threshold 0.98, the network undergoes substantial sparsification, retaining only 2153 nodes and 1920 edges (corresponding to 82.9% and 99.5% reductions, respectively), while still maintaining a high modularity of 0.94.

In addition to the representative thresholds (0.90, 0.95, and 0.98) discussed here, we conducted a comprehensive sensitivity analysis using a fine-grained sweep of thresholds from 0.00 to 0.99 in 0.01 increments. The evolution of structural metrics—including modularity, average degree, and community count—across this range is presented in Table S2 and Figure S2 of the ESI, supporting the robustness of our threshold selection strategy.

Figure 6 shows the network topology transformation in MOFGalaxyNet with various sparsification thresholds. The visualization captured the network at four distinct phases of sparsification. Panel (a) shows the original, densely interconnected MOFGalaxyNet, with a profusion of nodes and edges illustrating complex molecular interactions. In Panel (b), the network is displayed after applying a sparsification threshold of 0.9, which is noticeably less dense, with reduced nodes and edges, indicating the elimination of less significant connections while retaining more essential linkages. Panel (c) furthers this refinement at a threshold of 0.95, where the sparsity of the network is more pronounced, isolating clusters of high-similarity nodes and discarding even more peripheral connections. Finally, Panel (d) reveals a starkly sparsified network at a threshold of 0.98, distilling the network to its core constituents. This panel shows a significant reduction in network complexity, leaving only the most crucial nodes and edges, representing the most substantial similarities and the fundamental framework of the original network.

Validation of the sparsified MOF dataset

Summary of validation strategy

To assess the effectiveness and integrity of the sparsified MOFGalaxyNet, we implemented a multi-level validation framework incorporating both graph-based and non-graph-based machine learning models. The primary objective was to determine whether the sparsification process, driven by inverse link prediction (ILP), preserves the essential topological and compositional features necessary for accurate downstream prediction of MOF properties, particularly pore limiting diameter (PLD). Our validation begins with the deployment of a Graph Convolutional Network (GCN) on the original unsparsified graph, establishing a performance baseline. We then evaluate how the model performs across various sparsification thresholds (0.90, 0.95, and 0.98), examining how network reduction influences predictive capacity. Performance is assessed using classification accuracy, loss curves, confusion matrices, and class label distribution metrics. In parallel, we analyze the effect of sparsification on training efficiency and model generalizability by tracking computational time, overfitting trends, and class-wise performance shifts. To ensure robustness, we extend our evaluation to a suite of alternative learning methods, including GraphRAGE, Deep Neural Networks (DNN), Gradient Boosting Trees (GBT), Logistic Regression (LR), and Naïve Bayes (NB), each applied to both the original and sparsified graphs. Through this comprehensive validation strategy, we aim to demonstrate that ILP-based sparsification can reduce computational complexity without compromising the predictive quality of the network, thereby supporting efficient and scalable cheminformatics workflows.

GCN-based validation before and after sparsification

Our first step involved using a GCN model to predict the PLD size categories of MOFs, which are crucial for determining their applicability in various domains, such as gas storage or separation. The GCN model was built upon the preprocessed node features extracted from the molecular fingerprints of organic linkers, metal ion descriptors, and other relevant structural features. The GCN model could effectively predict PLD size categories by learning low-dimensional MOFs representations that capture their structural and compositional nuances.

Training the GCN involved splitting the dataset into training and testing sets, ensuring a diverse representation of the MOFs. The model's performance, reflected through its accuracy in classifying MOFs into their correct PLD size categories, validated the utility of the sparsified dataset. Notably, the model achieved good accuracy, highlighting the integrity of the dataset after sparsification and its potential for facilitating efficient and accurate predictions of MOF properties.

Our validation efforts began by predicting pore limiting diameter (PLD) sizes using a graph convolutional network (GCN) on the original graph structure before any sparsification techniques were applied. The GCN architecture used was consistent throughout our validation process. In deploying GCN to predict PLD sizes in MOFs, the GCN model employs a complex architecture designed to interpret graph-structured data effectively. This model incorporates molecular fingerprints from organic linkers, integer-encoded metal names, and geometric descriptors (e.g., largest cavity diameter and largest free sphere) to enhance the input data. The PLDs were classified as nonporous

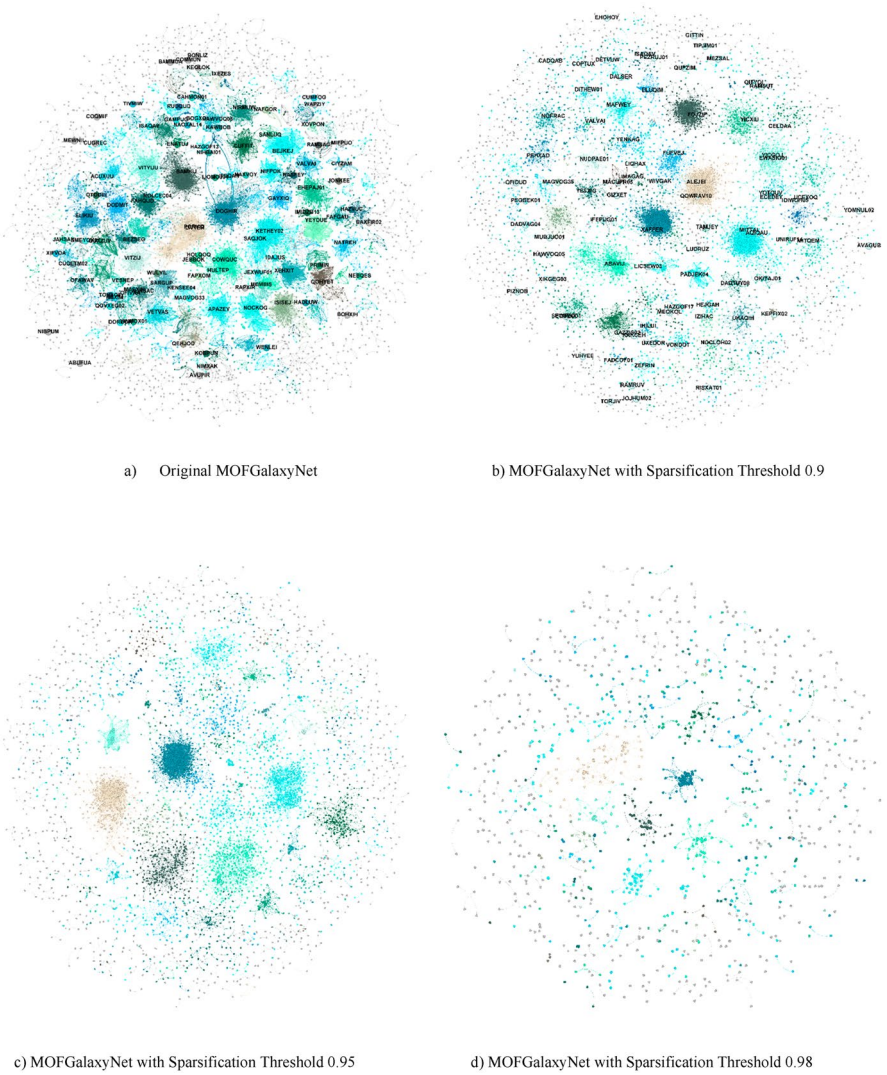


Fig. 6 Sparsification of Illustrating MOFGalaxyNet: From Original Complexity to Refined Simplicity at Thresholds of 0.9, 0.95, and 0.98

(PLD ≤ 2.4 Å), small (2.4 Å < PLD ≤ 4.4 Å), medium (4.4 Å < PLD ≤ 5.9 Å), or large (PLD > 5.9 Å). For example, the PLD for IRMOF-10 (refcode: LIHFAK) is 12.07725 Å, categorizing it among large-pore MOFs; HKUST-1 (refcode: FIQCEN) has a PLD of 5.23 Å, placing it in the medium-pore category; UiO-66 (refcode: RUBTAK) features a PLD of 3.99 Å, classifying it as a small pore; and Ni-Asp-bipy is categorized as a nonporous MOF (see Fig. 7).

Before applying machine learning techniques, Fig. 8 offers a distinct depiction of how class labels are distributed within the original dataset and those modified by different levels of sparsification. The graph illustrates that the proportions of various class labels remain consistent after the application of sparsification, demonstrating the efficacy of our chosen sparsification method. As the thresholds are adjusted from 0.9 to 0.98, the distributions of nonporous, small, medium, and large pore classes exhibit negligible changes. This uniformity is critical because it suggests that the sparsification process

discriminates the dataset's core structure. Machine learning algorithms must have equitable class representations for training. This balance is fundamental to developing dependable predictive models by mitigating potential biases toward overrepresented classes. Therefore, this visual evidence reinforces the validity of our sparsification technique, ensuring that the dataset's integral features remain intact for impartial and effective machine learning analyses.

We chose to utilize GCN for modeling due to its inherent ability to efficiently process and analyze graph-structured data. GCNs are particularly adept at maintaining critical informational integrity in sparsified graphs, demonstrating that the essential structural and feature-related information can be preserved even after removing certain edges and nodes. This capability is crucial for our studies involving metal–organic frameworks (MOFs), where understanding the nuanced relationships within complex molecular structures is vital. By applying GCN, we can effectively demonstrate that our sparsification techniques retain the key characteristics necessary for accurate modeling and analysis, thus proving that our approach to reducing computational complexity does not sacrifice the quality of insights derived from the data. This validation supports using sparsification to manage large datasets, ensuring that our models are both computationally efficient and robust in their predictive capabilities.

In our study, we employed a robust GCN architecture designed to handle graph-structured data effectively. This architecture includes two primary input layers: one for node features and another for the adjacency matrix. Both inputs are critical for capturing the intricate relationships and properties within the MOFs.

The node features are processed through dense layers with ReLU activation functions to introduce non-linearity, which is essential for deep learning models. To combat overfitting, L2 regularization with a coefficient of 0.01 and dropout layers with rates of 0.5 and 0.3 are applied sequentially after the first and second dense layers, respectively. These regularization techniques prevent the model from memorizing the training data, thereby enhancing its ability to generalize to new, unseen data.

Similarly, the adjacency matrix input is processed through its own set of dense layers, following the same regularization and dropout strategies. This parallel processing of node features and adjacency data ensures that node properties and connectivity are equally emphasized in the learning process.

After processing the inputs separately, their outputs are concatenated to integrate information from the node features and the adjacency matrix. This combined data then passes through a final dense layer with a softmax activation function, which classifies the nodes into various PLD size categories, reflective of the diverse properties of MOFs.

The model is compiled using the Adam optimizer with a specifically chosen low learning rate of 0.0009 to ensure smooth and stable convergence. Categorical crossentropy is used as the loss function, suitable for the multi-class classification task.

An early stopping mechanism is implemented to ensure the model's robustness against overfitting and to fine-tune its performance. This callback monitors the validation loss during training, ceasing further training if no improvement is observed and restoring the best model weights observed during the process. This early stopping mechanism is vital for maintaining the delicate balance between learning the intricate patterns in the

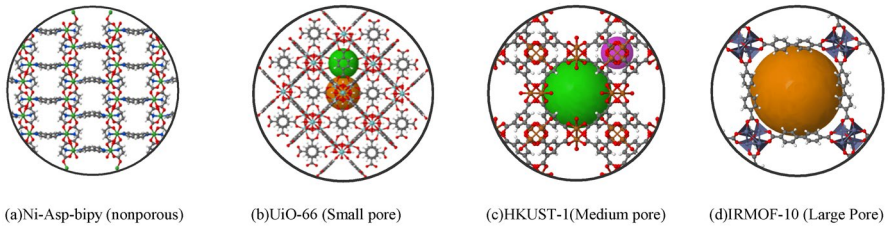


Fig. 7 illustrates four examples of MOFs featuring varying PLD sizes. **a** IRMOF-10 (refcode: LIHFAK), exhibiting a PLD of 12.07725 Å, is classified as a large-pore MOF. **b** HKUST-1 (refcode: FIQCEN), with a PLD of 5.23 Å, falls into the category of medium-pore MOFs. **c** UiO-66 (refcode: RUBTAK) has a PLD of 3.99 Å, characterizing it as a small pore. Finally, **d** Ni-Asp-bipy represents a nonporous MOF

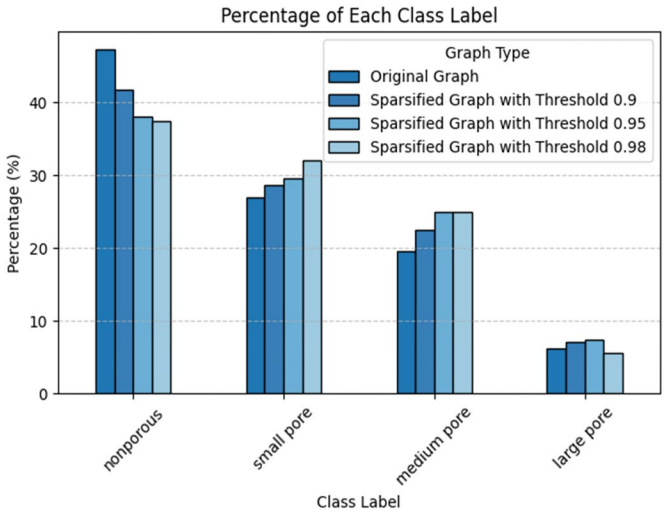


Fig. 8 Distribution of class labels in the original and sparsified graphs. This bar chart compares the percentages of nonporous, small-pore, medium-pore, and large-pore class labels across the original MOFGalaxyNet graph and its sparsified versions at thresholds of 0.9, 0.95, and 0.98

training data and the model's ability to generalize these patterns to new, unseen data, thereby enhancing the reliability and robustness of our predictive model.

For training and evaluation, the dataset is split into training and testing sets, with 20% of the data reserved for testing. This separation guarantees that the model's performance metrics, such as accuracy and the detailed confusion matrix, are evaluated on data not seen during the training phase. The confusion matrix, generated during the testing phase, provides a granular view of the model's performance across different MOF classes, illustrating specific areas where the model excels or may be confused between classes.

This comprehensive setup of the GCN architecture is designed to balance complexity with robustness. It ensures that our model not only learns effectively from the complex network dynamics of MOF data but also generalizes well to new scenarios, making it ideal for practical applications in computational material science.

Following the establishment of our GCN architecture, we evaluated the effectiveness of the models using the original graph structure before any sparsification. The results are encapsulated in the accompanying graphs (Fig. 9), illustrating the training

and validation loss (Fig. 9a), accuracy (Fig. 9b), and a confusion matrix detailing the classification outcomes (Fig. 9c). The training and validation loss graph demonstrates a rapid decline in loss during the initial epochs, indicating efficient early learning by the model. Although the training loss decreases significantly and then stabilizes, the validation loss follows a similar trend, with a minor uptick towards the end, potentially indicating the onset of overfitting. Early stopping was implemented during training to halt further iterations when the validation loss ceased to improve, thereby preventing the model from overfitting and optimizing its generalization capabilities on unseen data. The accuracy graph reveals a swift rise in training accuracy, showing that the model effectively learns from the dataset. The validation accuracy peaks at approximately 75.7%, reflecting the model's robust capacity to generalize from the complete graph data without sparsification.

The confusion matrix provides insights into the model's performance across four PLD size categories: nonporous, small, medium, and large. There is a strong concentration of correct predictions along the matrix's diagonal, highlighting the model's overall precision. The nonporous class shows an accuracy of approximately 85.4%, indicating high reliability in this category. The large pore class also demonstrates robust accuracy at 92.4%. Conversely, the medium pore class has a lower accuracy of 47.2%, reflecting the challenges in classifying this category due to the inherent complexities of its structural and chemical diversity. The small pore class, with an accuracy of 63.1%, shows some misclassifications with medium pores, suggesting opportunities for improving model precision in this area.

These results, derived from a GCN applied to an unaltered graph structure, confirm that the model is quite adequate, with a final accuracy of 75.7%. The early stopping protocol during training has contributed to this performance by ensuring that the model does not learn from noise or redundant data, thereby enhancing its predictive reliability for future material science applications involving complex graph-data structures. The analysis highlights the model's adept handling of varied MOF classifications and suggests strategies for refining classification accuracy between closely related categories.

After implementing a sparsification threshold of 0.90, resulting in a 27.2% reduction in nodes and a 90.7% reduction in edges, the predictive accuracy of the graph convolutional network (GCN) model increased remarkably to 94%. With the application of a sparsification threshold of 0.9, the training and validation loss graphs (Fig. 10a) show a refined learning curve. Early stopping is employed to prevent overfitting, as indicated by the stable validation loss following an initial sharp decrease, suggesting optimal model tuning. The accuracy graph (Fig. 10b) demonstrates an improved learning efficacy, with the validation accuracy stabilizing at an impressive 92%, indicating superior generalization from the sparsified graph data. This increase, compared to the non-sparsified dataset, underscores the effectiveness of reducing data complexity by focusing the model's learning on the most informative features.

The confusion matrix (Fig. 10c) at a sparsification threshold of 0.9 shows varied performance across the four PLD size categories: nonporous, small, medium, and large. The nonporous and large pore classes retain high reliability, achieving accuracies of 84.5% and 88.7%, respectively, with minimal changes from the original dataset. The small pore

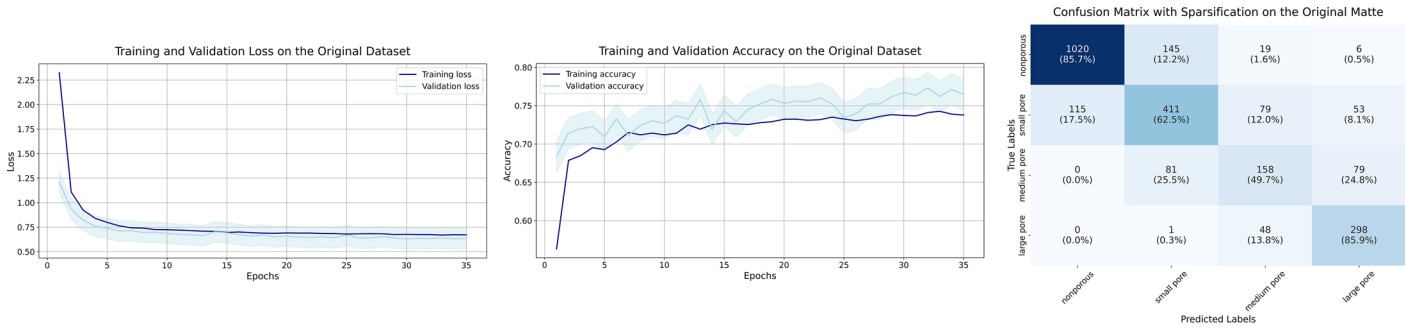


Fig. 9 Model Performance Metrics Without Network Sparse. Part **a** depicts the models' learning curve through training and validation loss, part **b** shows the accuracy trajectory over epochs, and part **c** provides a detailed confusion matrix for the classification results, all of which demonstrate the models' baseline efficiency

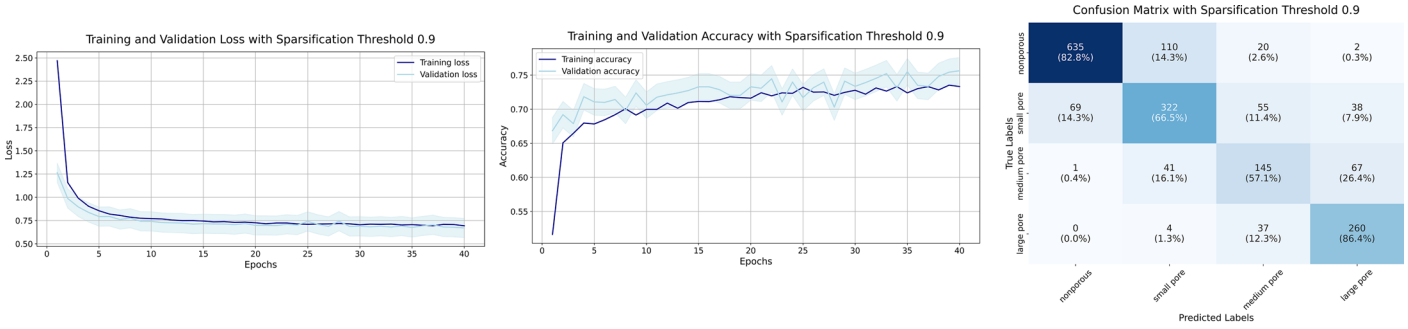
class shows an accuracy of 63.8%, while the medium pore class accuracy drops to 26.8%, indicating that this class faces challenges in accurate classification. These results suggest that, while a 0.9 sparsification threshold reduces computational demands, it does not improve classification accuracy for all categories equally. This emphasizes the need for careful tuning of sparsification parameters to balance efficiency and accuracy across different classes.

After raising the sparsification threshold to 0.95, we observed a pronounced decrease of 48.1% in nodes and 97.0% in edges within the network. Upon applying a sparsification threshold of 0.95, the training and validation loss curves (Figure 11a) show a more pronounced reduction in loss, with early stopping implemented to counteract overfitting. The validation loss stabilizes lower than previous models, indicating successful adjustment to a more sparsified dataset. In the accuracy graph (Figure 11b), validation accuracy further improves, achieving a peak of about 94%, highlighting the model's enhanced ability to generalize from even more reduced data complexity. This boost in validation accuracy, compared to less sparsified datasets, emphasizes the model's efficiency in capturing essential information with fewer connections. Figure 11c for the sparsification threshold of 0.95 reveals varied performance across the four PLD size categories: nonporous, small, medium, and large. The nonporous and large pore classes retain high reliability, achieving approximately 85.7% and 82.7% accuracy, respectively, even with reduced data. However, accuracy for the medium pore class decreases significantly to around 45.8%, and the small pore class accuracy drops to 65.7%. These shifts suggest that while overall model efficiency has improved, the small and medium pore categories are notably impacted by the sparsification, likely due to the loss of discriminative information. This highlights the trade-offs at higher sparsification thresholds, balancing complexity reduction with predictive accuracy.

Elevating the sparsification threshold to an extreme of 0.98 led to a drastic reduction in network density, decreasing the nodes and edges by 83.3% and 99.5%, respectively. Despite this, the graph convolutional network (GCN) maintained a high prediction accuracy of 92.66%. However, this seemingly strong performance may conceal the risk of overfitting—a common issue when training with sparser datasets.

Upon setting a sparsification threshold of 0.98, the training and validation loss curves (Fig. 12a) display a sharp and quick reduction in training loss, stabilizing at a very low level, indicative of an effective fit to the drastically reduced dataset. Early stopping comes into play as seen in the slight uptick in validation loss, preventing the model from overfitting on the much sparser data. The accuracy graph (Fig. 12b) shows that the training accuracy climbs swiftly and plateaus, with the validation accuracy reaching approximately 95%, illustrating remarkable generalization capability on a highly sparsified dataset.

The confusion matrix (Fig. 12c) at this sparsification level shows that the accuracy across all categories—nonporous, small, medium, and large—holds relatively steady or improves slightly despite the extreme reduction in data connections. The nonporous and large pore classes maintain high accuracy, above 80%, indicating the model's continued effectiveness in these categories. However, the medium and small pore classes show a slight decrease in accuracy, now recorded around 67.9% and 12.8% respectively. This reduction may suggest that while the model still performs well overall, the extreme



(a) **(b)** **(c)**

Fig. 10 Enhanced GCN model performance post-sparsification. **a** and **b** depict the training and validation loss and accuracy graphs after sparsification, showing optimized learning and generalization. **c** displays the confusion matrix, highlighting an increase in accurate predictions and illustrating the refined predictive performance of the GCN model on a sparsified MOFGalaxyNet at a threshold of 0.90

Table 3 Accuracy of models across thresholds

Threshold	GCN	GraphRAGE	GB Trees	LR	Naïve Bayes	DNN
Original	0.757	0.725	0.672	0.754	0.318	0.768
0.90	0.735	0.714	0.688	0.733	0.332	0.736
0.95	0.721	0.728	0.692	0.713	0.335	0.708
0.98	0.768	0.768	0.655	0.712	0.463	0.745

Table 4 Kappa of models across thresholds

Threshold	GCN	GraphRAGE	GB Trees	LR	Naïve Bayes	DNN
Original	0.644	0.615	0.467	0.626	0.157	0.655
0.90	0.623	0.593	0.548	0.614	0.171	0.622
0.95	0.610	0.621	0.579	0.601	0.161	0.596
0.98	0.667	0.667	0.509	0.591	0.290	0.640

sparsification has potentially removed some nuanced information necessary for distinguishing between the more challenging categories of medium and small pores.

These results underscore the nuanced impact of high sparsification thresholds: while they can significantly enhance model efficiency by reducing computational demands, they may also risk losing critical information that affects the model's ability to differentiate between more subtly distinct classes. This balancing act between sparsification and model performance highlights the importance of selecting an optimal threshold that minimizes data complexity without compromising the essential predictive qualities of the model.

Graph sparsification using Inverse Link Prediction (ILP) significantly enhances the computational efficiency of Graph Convolutional Networks (GCNs) in characterizing metal–organic frameworks (MOFs) while preserving predictive accuracy. By incrementally increasing the sparsification threshold, the number of nodes and edges in the network is substantially reduced, leading to a marked decrease in computational complexity. For instance, a threshold of 0.98 achieves an 82% reduction in training time compared to the original graph, dropping from approximately 263.84 s to 47.43 s. Despite this drastic reduction, GCN models maintain high predictive accuracy, with a sparsification threshold of 0.9 yielding stable validation accuracy that surpasses the unsparsified dataset. This indicates that ILP effectively retains the most informative network features, allowing robust predictions of MOF properties, such as pore limiting diameter (PLD), across various size categories.

The sparsification process, however, introduces trade-offs that require careful threshold selection to balance efficiency and information retention. At a threshold of 0.9, the network achieves significant complexity reduction while preserving excellent predictive performance across nonporous, small, medium, and large pore classes, with minimal loss of critical structural information. Higher thresholds, such as 0.95 and 0.98, further reduce computational demands but risk losing nuanced details, particularly for medium and small pore categories, leading to variability in classification accuracy. A threshold of 0.9 emerges as the optimal choice, offering a robust compromise that enhances

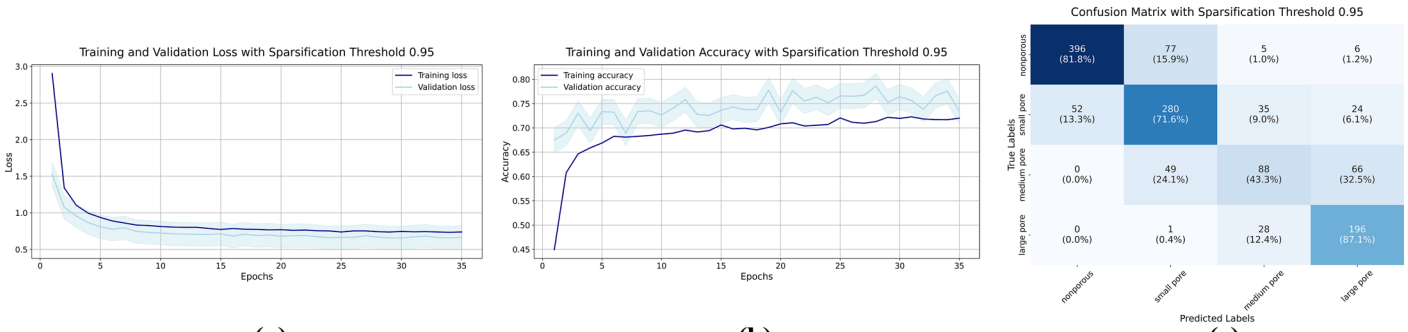
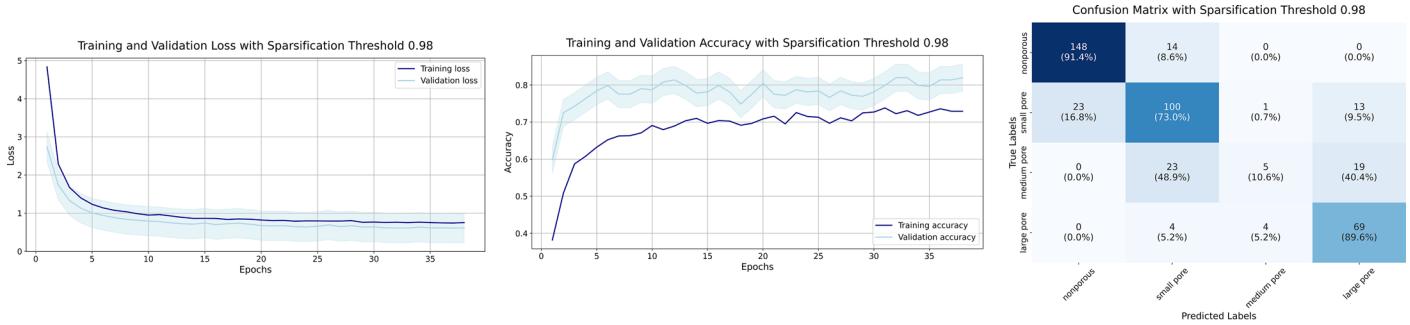


Fig. 11 Efficacy of the GCN model at a sparsification threshold of 0.95. Panel **a** shows the consistent training and validation loss, **b** depicts the efficient learning process as indicated by training and validation accuracy, and **c** presents a confusion matrix illustrating predictive performance across different pore sizes on a sparser MOFGalaxyNet



(a) **(b)** **(c)**

Fig. 12 GCN Performance at a Maximum Sparsification Threshold of 0.98. The figure reveals (**a**) the training and validation loss trends, with premature stabilization hinting at overfitting; **b** the sharp increase and high plateau in training accuracy contrasted with validation volatility; and **c** the confusion matrix indicating high classification accuracy within the substantially reduced dataset

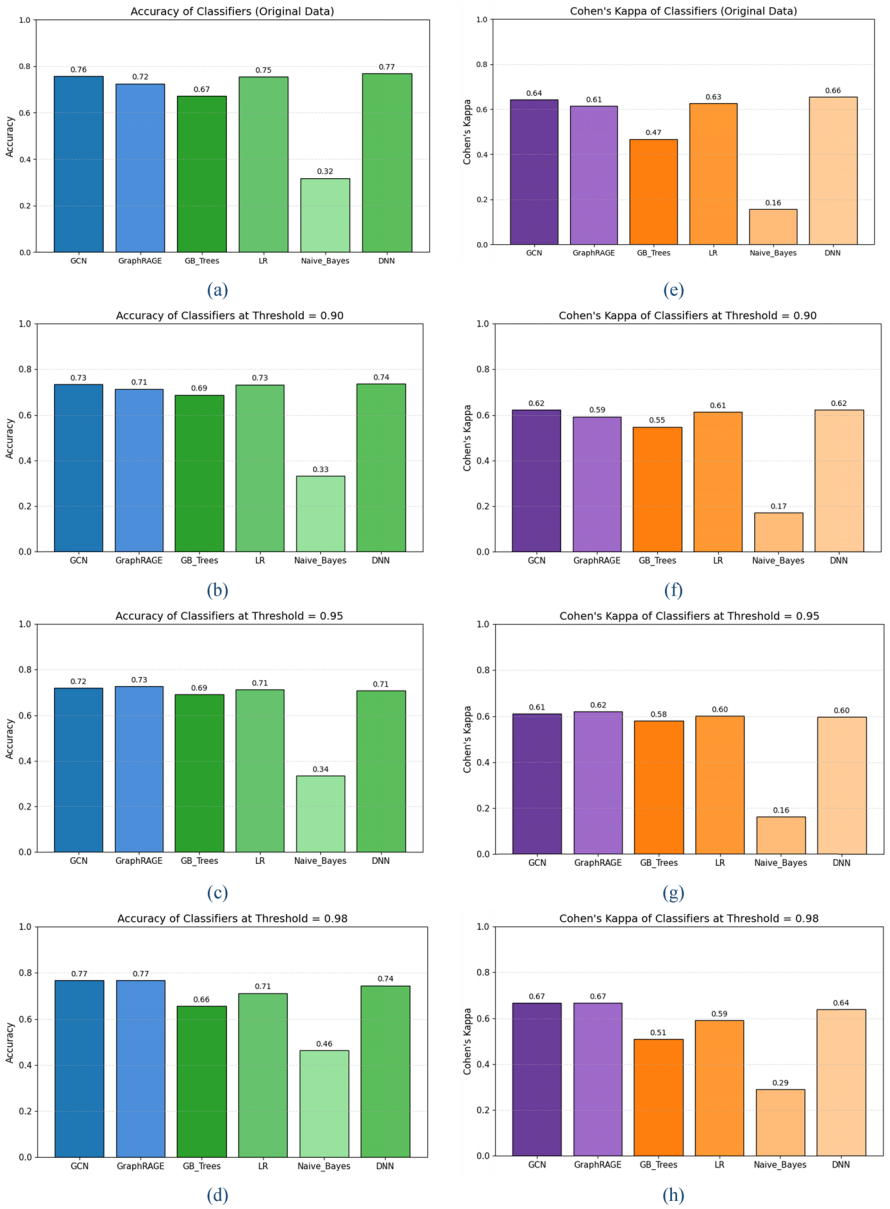


Fig. 13 Comparative performance of machine learning classifiers on MOFGalaxyNet before and after ILP sparsification. **a**–**d** display accuracy and (**e**–**h**) present Cohen's Kappa, illustrating the resilience of graph-based and non-graph-based models across thresholds of 0.90, 0.95, and 0.98

computational efficiency while maintaining the model's ability to distinguish complex MOF structures, thus supporting scalable and accurate analysis in computational materials science.

Validation with alternative ML models

Following our initial validation of the sparsified MOFGalaxyNet graph using GCN as detailed earlier in Sect. "Validation of the sparsified MOF dataset", we proceeded to evaluate its predictive capacity with a broader range of machine learning methods.

To rigorously assess the sparsified graph both before and after Inverse Link Prediction (ILP)-based sparsification, we conducted a comprehensive evaluation employing a diverse array of models. This included additional graph-based models, such as GraphRAGE, which leverage the graph's topological structure, alongside non-graph-based models—including Deep Neural Networks (DNN), Gradient Boosting Trees (GB Trees), Logistic Regression (LR), and Naïve Bayes (NB)—which rely solely on node feature vectors without structural context. This multifaceted approach aimed to establish a performance baseline using the original graph, evaluate the robustness of sparsified versions across multiple thresholds, and determine whether ILP preserves sufficient structural and feature-based information to support varied machine learning paradigms. Such an extensive analysis is essential, as applications in computational materials science often require adaptability, balancing the computational efficiency of feature-based methods with the topological insights provided by graph-based approaches, depending on research goals and resource constraints.

Performance was assessed using two complementary metrics: Accuracy, which quantifies the proportion of correctly classified samples, and Cohen's Kappa [36], which measures classification agreement while correcting for chance, offering a more nuanced evaluation in the presence of class imbalance—such as that observed in the Pore Limiting Diameter (PLD) categories of MOFGalaxyNet. Together, these metrics provide a robust framework for interpreting model efficacy, ensuring that conclusions account for both overall predictive success and resilience to distributional biases inherent in the dataset.

Evaluation of the original MOFGalaxyNet graph, characterized by its dense connectivity, revealed distinct performance profiles across the model suite, as summarized in Tables 3 and 4 Graph-based models like GCN and GraphRAGE demonstrated strong predictive capabilities, leveraging the rich structural information to effectively discern complex relationships among metal–organic frameworks (MOFs). Their success can be attributed to their ability to exploit neighborhood connectivity and propagate information through the graph's extensive edge network, aligning with their architectural strengths.

Among non-graph-based models, DNN emerged as a standout, achieving performance comparable to or exceeding that of graph-based counterparts. This suggests that the node feature vectors—encompassing molecular fingerprints, metal descriptors, and geometric properties—encode substantial predictive power, capturing relational information implicitly without requiring explicit structural input. LR also performed admirably, reinforcing the notion that linear relationships within the feature space are sufficient for robust classification in this context. Conversely, GB Trees displayed moderate success, limited by their inability to incorporate topological cues, while NB lagged significantly, its assumption of feature independence proving incompatible with the interdependent nature of MOF-derived features.

These baseline findings underscore the complementary roles of graph-based and non-graph-based approaches. The former excels by harnessing the graph's structural complexity, while the latter demonstrates that carefully engineered features can independently yield high predictive fidelity, albeit with varying degrees of sensitivity to the dataset's intrinsic characteristics.

Subsequent analysis explored model performance on ILP-sparsified graphs at thresholds of 0.90, 0.95, and 0.98, corresponding to progressive edge reductions (Table 2). The results, detailed in Tables 3 and 4 and illustrated in Fig. 13, elucidate the interplay between sparsification intensity and predictive capability. At the 0.90 threshold, where edge reduction is moderate, most models exhibited only a slight decline in performance compared to the baseline. Graph-based models like GCN and GraphRAGE maintained robust outcomes, benefiting from the preservation of high modularity, which sustains key community structures despite the loss of connectivity. This resilience suggests that ILP effectively retains edges critical to graph-based learning.

Non-graph-based models mirrored this trend, with DNN and LR sustaining strong performance, indicating that the feature vectors remain highly informative even after significant structural pruning. GB Trees showed modest gains, likely due to a simplified feature space enhancing decision-tree efficiency, while NB remained consistently weak, its limitations persisting across all conditions.

At the 0.95 threshold, further edge reduction introduced more pronounced effects. GCN experienced a slight performance dip, reflecting the loss of finer structural details, though its reliance on preserved modularity mitigated severe degradation. GraphRAGE, conversely, showed improvement, its neighborhood-aggregation approach proving advantageous in sparser topologies where focused connectivity enhances generalization. Among non-graph models, DNN and LR retained competitive efficacy, though with marginal declines, suggesting a gradual erosion of nuanced relational cues embedded in the features. GB Trees peaked at this threshold, capitalizing on the streamlined feature set, while NB showed no significant progress.

At the extreme 0.98 threshold, marked by a drastic edge reduction, graph-based models like GCN and GraphRAGE exhibited a notable rebound, achieving performance levels comparable to the original graph. This resurgence is likely driven by the highly modular structure, which concentrates learning on core relationships, enhancing predictive focus. DNN and LR continued to perform well, underscoring the robustness of the feature extraction process, while GB Trees declined, possibly reaching a sparsity limit beyond which its adaptability wanes. NB showed a surprising uptick, though it remained the weakest performer, benefiting marginally from the simplified structure but still hindered by its foundational assumptions.

The performance trajectories, visualized in Fig. 13, reveal several critical insights. Graph-based models demonstrate remarkable adaptability to sparsification, with GCN and GraphRAGE thriving at higher thresholds due to ILP's preservation of modular communities, which align with their learning paradigms. Non-graph models, particularly DNN and LR, exhibit resilience across thresholds, affirming that the feature vectors retain substantial discriminative power even as structural complexity diminishes. The moderate success of GB Trees and the persistent weakness of NB further emphasize the importance of model alignment with dataset properties.

Sparsification introduces a trade-off between complexity reduction and information retention. The 0.90 threshold strikes an effective balance, offering significant computational efficiency while preserving predictive fidelity across most models. At 0.95, increased sparsity begins to erode subtle structural cues, though performance remains viable. The 0.98 threshold, while boosting graph-based models through focused

modularity, risks overfitting, potentially compromising generalization for certain PLD categories.

To more clearly highlight these effects, we have explicitly presented the comparative performance of all models before and after sparsification. This comparison illustrates that complex models, such as GCN, GraphRAGE, GBT, and DNN, remain robust across varying sparsity levels, while simpler models, like Logistic Regression and Naïve Bayes, show noticeable drops in performance at higher thresholds. These trends emphasize ILP's strength in preserving essential predictive structures, particularly for expressive models, while revealing the sensitivity of linear and probabilistic models to network reduction.

This validation confirms that ILP-sparsified graphs retain sufficient information to support both graph-based and non-graph-based machine learning. ILP emerges as a versatile sparsification strategy that enables scalable MOF analysis with minimal loss of predictive power, and the 0.90 threshold is recommended as the most effective balance between efficiency and accuracy.

Network integrity assessment after sparsification

Evaluating the structural integrity of MOFGalaxyNet following Inverse Link prediction (ILP)-based sparsification is crucial to ensure that essential network properties are preserved while achieving computational efficiency. This assessment employed heatmaps of adjacency matrices and a suite of centrality metrics to analyze the impact of sparsification thresholds (0.90, 0.95, and 0.98) on network topology and connectivity, balancing the need for simplification with the retention of functional characteristics relevant to MOF analysis.

Heatmaps derived from adjacency matrices provide a visual representation of network connectivity across sparsification levels, as shown in Fig. 14. The original network (Fig. 14a) exhibits a dense interaction pattern, reflecting the intricate web of MOF similarities. At the 0.90 threshold (Fig. 14b), the structure remains largely intact, retaining most primary connections, which supports comprehensive analysis but incurs higher computational costs. The 0.95 threshold (Fig. 14c) achieves a balanced reduction, preserving significant linkages while streamlining the network, suggesting an effective compromise for MOF studies where both accuracy and efficiency are paramount. At the stringent 0.98 threshold (Fig. 14d), the heatmap reveals a markedly simplified structure, highlighting only the most critical connections. While this enhances computational tractability, it risks omitting less prominent yet potentially relevant interactions, necessitating careful threshold selection based on resource constraints and analytical objectives.

Centrality metrics further elucidate the preservation of network integrity by quantifying the roles of nodes (MOFs) within the original and sparsified graphs [37, 38]. The degree distribution analysis, illustrated in Fig. 15, confirms that MOFGalaxyNet retains its characteristic scale-free topology across sparsification thresholds of 0.90, 0.95, and 0.98. The density curves reveal a long-tailed distribution, with the original graph exhibiting a pronounced peak at lower degrees and a gradual decline, indicative of a few highly connected nodes amidst a majority with fewer connections. As sparsity increases, the main plot shows a progressive reduction in the density of high-degree nodes, reflecting the removal of less critical edges. However, the zoomed-in view (degrees 0–40)

highlights that the density of low-degree nodes remains largely consistent across all thresholds, ensuring that the network's foundational connectivity is preserved. This selective edge pruning maintains the core structural patterns, allowing the sparsified network to closely approximate the original's topological essence.

Closeness centrality (Fig. 16) in MOFGalaxyNet measures the average similarity distance from a MOF to all other MOFs, indicating its efficiency in transmitting characteristics across the network [39]. The density curves reveal that the distribution of closeness centrality in the original graph remains largely consistent after sparsification at thresholds of 0.90, 0.95, and 0.98, with the curves showing substantial overlap across all levels. This preservation suggests that the network's interconnectedness is maintained despite significant edge reduction, ensuring that MOFs retain their ability to facilitate efficient similarity-based interactions throughout MOFGalaxyNet.

Eccentricity[40] (Fig. 17) in MOFGalaxyNet measures the greatest distance from a node (MOF) to any other node within the network, reflecting the extent of reachability across the graph. The density curves demonstrate that the distribution of eccentricity values in the original graph remains largely consistent after sparsification at thresholds of 0.90, 0.95, and 0.98, with the curves exhibiting significant overlap. This similarity indicates that MOFs in the sparsified networks maintain their relative proximity to one another, preserving the network's essential structure and ensuring that overall connectivity and reachability are sustained despite the reduction in edges.

Betweenness centrality [39] in MOFGalaxyNet quantifies the frequency with which a node (MOF) acts as a bridge along the shortest paths connecting other nodes. This measure is essential for identifying nodes that play pivotal roles in facilitating interactions across the network. The distribution of betweenness centrality in Fig. 18 reveals that sparsification primarily reduces the centrality of nodes with lower values, while effectively preserving the central positions of highly influential nodes. This selective reduction ensures that essential network connectivity and structural integrity remain robust, confirming the effectiveness of sparsification in maintaining critical interactions within the MOF network.

Eigenvector centrality [41] in MOFGalaxyNet quantifies the influence of a node (MOF) based on the importance of its neighboring nodes, where nodes connected to highly influential neighbors exhibit higher centrality scores. The eigenvector centrality distribution in Fig. 19 illustrates that the network maintains its core structural and influential patterns even after sparsification, with similar overall trends evident across different thresholds. Furthermore, the figure includes an inset plot specifically zooming into eigenvector centrality values between approximately 0.55 and 0.7. This inset highlights in detail how sparsification minimally affects nodes within this mid-to-high centrality range, confirming that influential nodes preserve their significant roles within the network despite the reduction of overall complexity.

in summary, the analysis of various centrality measures—degree, closeness, betweenness, eigenvector, and eccentricity—demonstrates that graph sparsification via inverse link prediction effectively reduces the complexity of MOFGalaxyNet while preserving essential structural characteristics. Comparisons between original and sparsified graphs, supported by detailed analyses, reveal that crucial connectivity patterns and centrality distributions remain intact despite significant reductions in nodes and edges. This

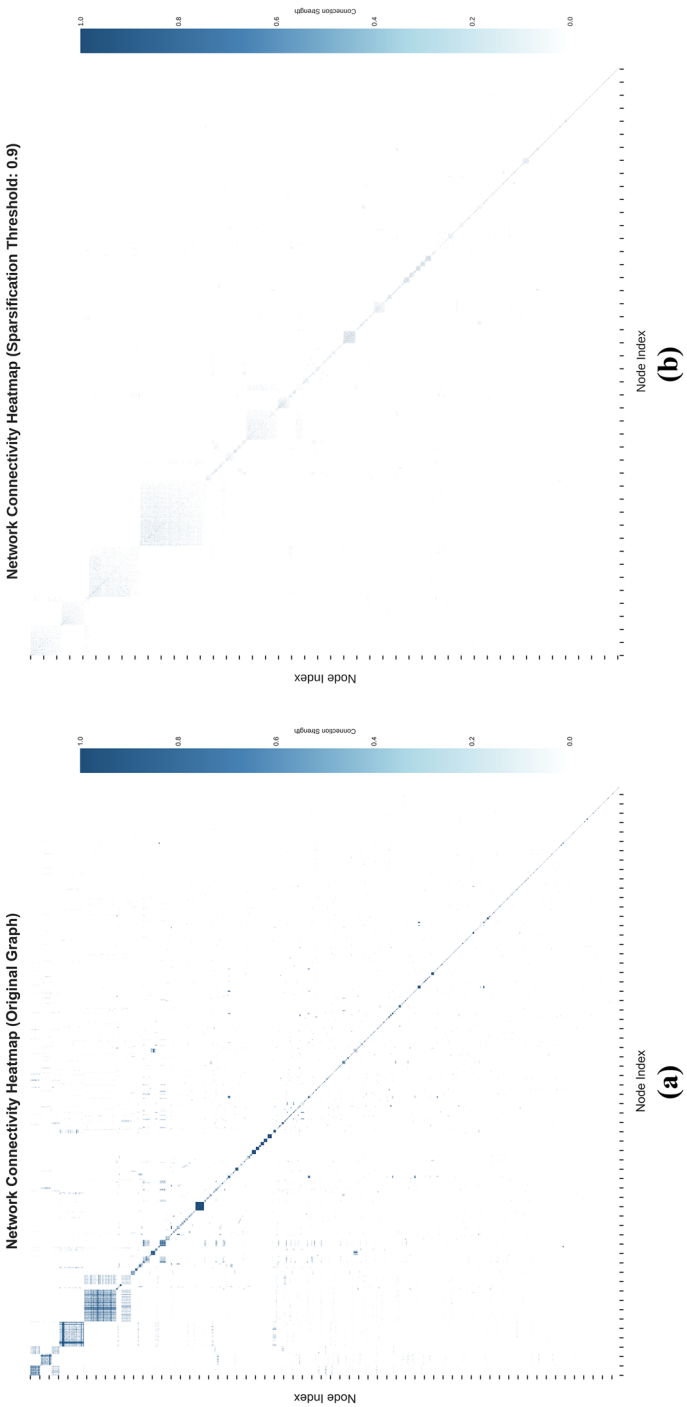


Fig. 14 Heatmap of the adjacency matrix in the original and spatialized graphs

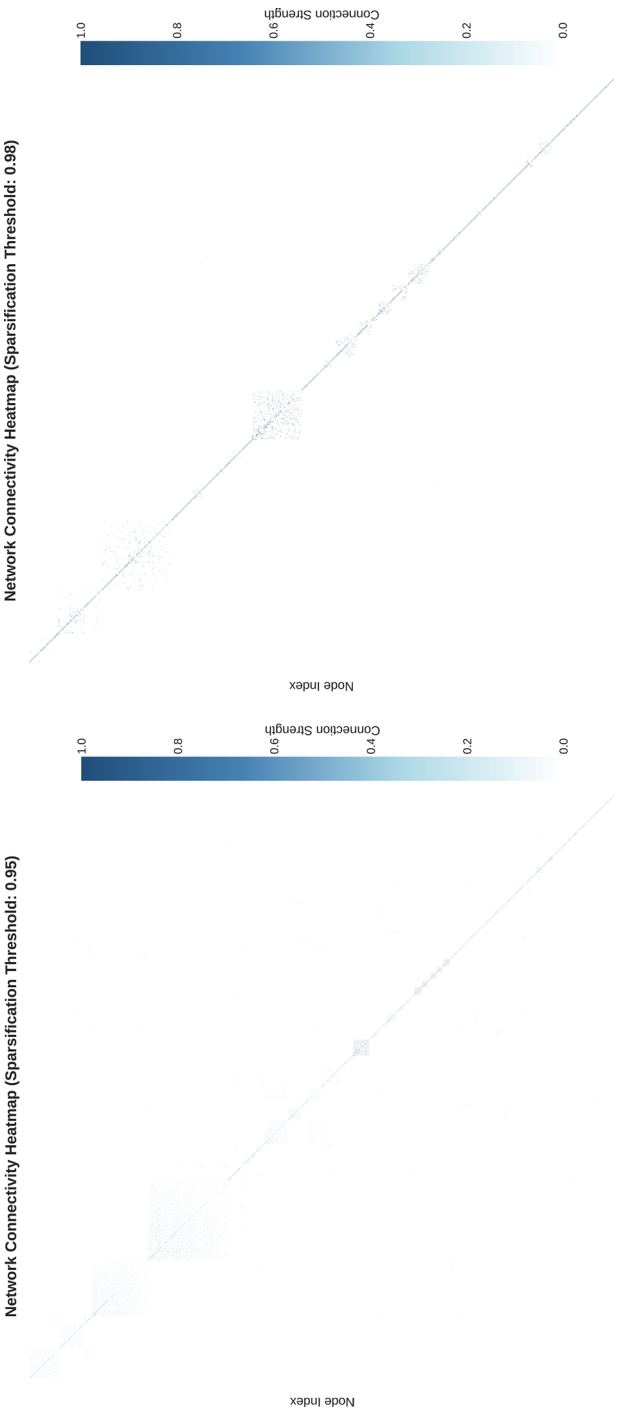


Fig. 14 (continued)

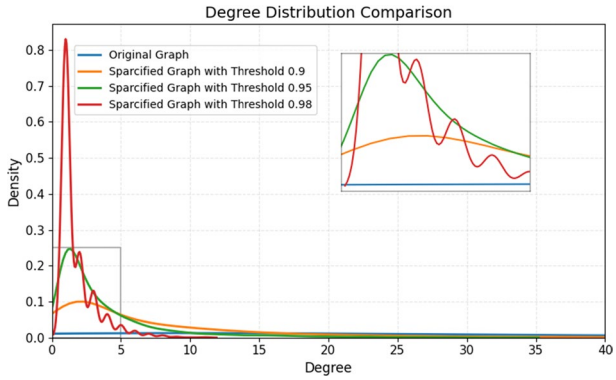


Fig. 15 Degree distribution comparison between the original and sparsified graphs at different thresholds, with a zoomed-in view

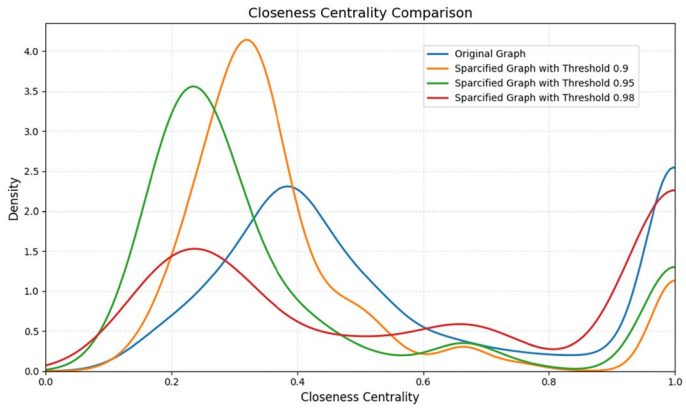


Fig. 16 Closeness centrality comparison between the original and sparsified graphs at different thresholds

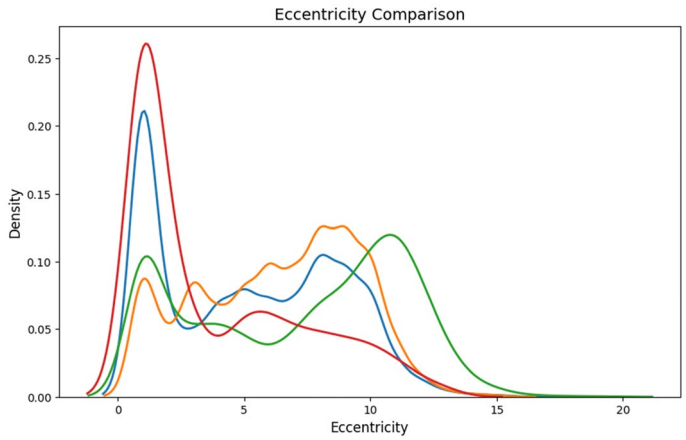


Fig. 17 Eccentricity comparison between the original and sparsified graphs at different thresholds

retention of influential nodes ensures that critical network properties persist, thereby enabling accurate and efficient characterization of MOFs. Consequently, the sparsification approach not only alleviates computational complexity but also sustains network

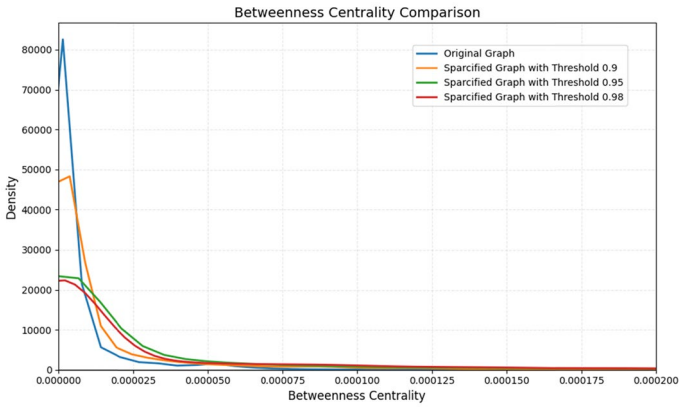


Fig. 18 Betweenness centrality comparison between the original and sparsified graphs at different thresholds

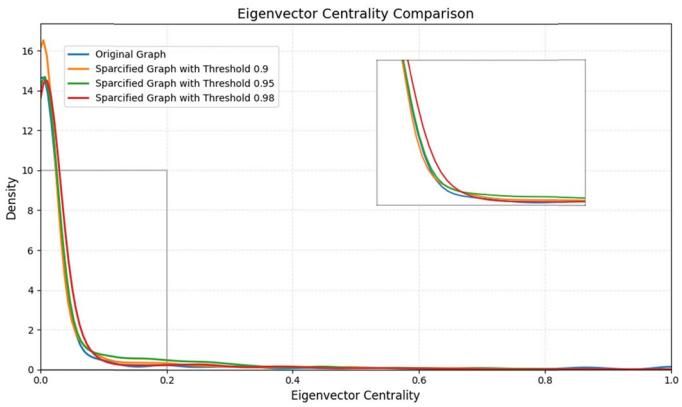


Fig. 19 Eigencentrality compares the original and sparsified graphs at different thresholds

integrity and functionality, validating its utility for advanced materials science and network-based analysis.

To further demonstrate the effectiveness of our proposed ILP framework beyond standalone sparsification performance, we conducted a comparative analysis against a traditional baseline method, Edge Betweenness Centrality (EBC), as described in the following subsection.

Comparison with classical sparsification methods

To further validate the performance of the proposed ILP-based sparsification, we performed a comparative analysis against the classical Edge Betweenness Centrality (EBC) sparsification method, a widely used baseline in network analysis. EBC identifies critical edges using global betweenness scores but comes with considerable computational costs and often leads to unintended loss of network structure when applied to large-scale graphs such as MOF similarity networks.

For a fair comparison, we aligned both methods based on equivalent edge reduction ratios. As shown in Tables S1 and S4 and visualized in Figures S2 and S4 of the Electronic

Supplementary Information (ESI), ILP consistently outperforms EBC by achieving higher modularity across all sparsification levels, reducing node loss compared to EBC, providing a more controlled reduction in average node degree, and significantly lowering computational time. Moreover, the modularity trend, node and edge reduction percentages, and average degree curves (Figure S4) clearly indicate that ILP better preserves community structures, even under strong sparsification. The complexity comparison in Table S1 further highlights the superior efficiency of ILP, which achieves sparsification with linear-time complexity relative to the number of edges, in contrast to the $O(ElogE)$ complexity of EBC.

These results confirm that ILP provides an effective balance between graph simplification, structural preservation, and computational efficiency, making it well-suited for large-scale MOF network analysis.

Case studies on sparsified network applications

When predicting the properties of a new MOF, it is essential to format the MOF's representation consistently with the training data used for model development. Should the new MOF feature have a distinct structure, preprocessing may be required to transform it into a compatible graph format for GCN analysis. To ensure both consistency and precision in our predictions, we employ the same preprocessing protocols and hyperparameters that were utilized during the training phase.

Our GCN model evaluation included four MOFs—CIGXIA, RUBTAK02, SABVUM01, and VELVOY—as depicted in Fig. 20. We extracted the metal and linker details from these MOFs to update the adjacency matrix, integrating similarities between the new and existing building blocks within the matrix. This updated matrix, now enriched with fresh building unit data, was subsequently used to predict guest accessibility. Interestingly, the predictions aligned with the pore limiting diameter (PLD) classifications of the respective MOFs.

To validate these predictions, we calculated the porosity of the four MOFs using Zeo++, a software tool designed for analyzing the geometric properties of porous materials. The analysis spanned MOFs with varying PLDs compared against actual sizes postsparsification. Our evaluation was carried out at three different sparsification thresholds: 0.9, 0.95, and 0.98. For instance, the MOF CIGXIA consistently fell into the "Larg" category across all the thresholds, perfectly matching the empirical data. This consistency underscores the robustness of the models, particularly for MOFs genuinely categorized as "large". Conversely, the MOF RUBTAK02 showed significant prediction variability and was identified accurately only as "Larg" at the highest threshold (0.98). This suggests that higher sparsification levels may be necessary for precise predictions of larger MOFs, albeit at the risk of misclassifying smaller variants. For MOF SABVUN01, the predictions were as accurate as those for 'Smal' at the lower thresholds, aligning with the actual size after recategorization. However, at the highest threshold (0.98), the prediction inaccurately shifted to "Larg", indicating potential overfitting at this level of sparsification. On the other hand, predictions for MOF VELVOY remained consistent across all thresholds, where the model reliably identified it as "Small". This consistent accuracy across various thresholds highlights the reliability of MOFs with smaller PLDs. Our comparative analysis revealed varying degrees of accuracy across different thresholds. While

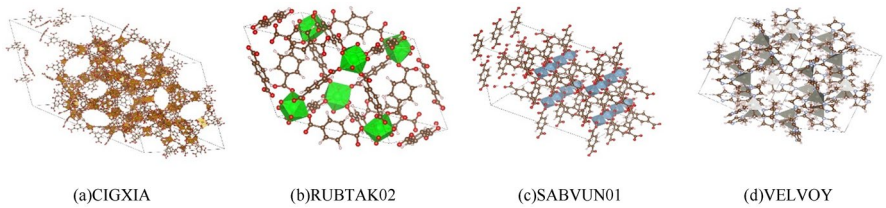


Fig. 20 Visualizations of MOFs used in GCN model performance evaluation, illustrating the diversity in structure and porosity: **a** CIGXIA, represented by a widespread porous network; **b** RUBTAK02, characterized by prominent green linker nodes and dense metal clusters; **c** SABVUN01, characterized by a compact and regular framework; and **d** VELVOY, characterized by a sparser lattice structure. These MOFs serve as test cases to predict guest accessibility correlating with pore-limiting diameters (PLDs)

the highest threshold (0.98) tends to yield accurate predictions for larger MOFs, it risks misclassifying smaller MOFs as belonging to larger categories. However, lower thresholds (0.9 and 0.95) offer a more conservative approach, effectively capturing smaller categories but occasionally missing larger ones. Operating at a sparsification threshold of 0.9 provides an optimal balance, accurately categorizing both small- and medium-sized MOFs while minimizing significant misclassifications. This approach, conservative yet precise, is suitable for a wide range of MOF sizes and is recommended for further applications in predicting the characteristics of MOFs in unseen datasets.

Conclusion and future direction

This study successfully introduced a novel method of graph sparsification through inverse link prediction (ILP) enhanced by the use of GCN, representing a significant advance in the computational analysis of metal–organic frameworks (MOFs). Key achievements include enhanced computational efficiency, where the application of ILP and GCN enabled strategic pruning of the network, resulting in up to a 90% reduction in the number of edges. This reduction correlates with a potential decrease in computational complexity while maintaining high predictive accuracy for critical MOF properties such as the pore limiting diameter (PLD). Unlike traditional sparsification or feature selection methods, ILP strategically preserves essential structural relationships within the MOF network, allowing for significant computational savings while retaining accuracy, an advantage not typically achievable with other approaches. Additionally, the refined MOFGalaxyNet model facilitated a deeper understanding of MOF properties, enhancing applications in energy storage and environmental remediation. The process also generated a machine learning-ready dataset rich in structural and functional insights, providing a robust platform for rapid material discovery and characterization, and enabling scientists to apply predictive analytics efficiently, significantly accelerating the innovation cycle in materials science.

Potential future applications of our inverse link prediction-based sparsification framework include its extension to biological network analysis, where reducing complexity while preserving essential structure is crucial. In drug repositioning, sparsifying heterogeneous biological networks could help retain key regulatory relationships, similar to the regulation-aware graph learning strategy proposed for drug repurposing over biological networks [42]. For drug–drug interaction prediction, applying

sparsification to biomedical knowledge graphs while maintaining spatial and semantic coherence may enhance interpretability and efficiency, as demonstrated by the spatial-aware capsule-based graph neural network model [43]. In protein–protein interaction prediction, our approach could be integrated with community-preserving algorithms like the mixed membership stochastic blockmodel, which effectively infers latent biological complexes within protein interaction networks [44].

Additionally, our future work aims to explore the integration of optimization algorithms within the sparsified MOF similarity network to guide the selection of optimal frameworks for specific applications, such as targeted drug delivery. For example, our previously developed nature-inspired algorithms—the Lotus Effect Optimization Algorithm (LEA) [45] and the Multimodal Lotus Effect Algorithm (MLEA) [46]—have shown strong performance in engineering design problems. These could be adapted to search for the best-fitted MOFs by optimizing multiple criteria (e.g., guest accessibility, pore size, chemical compatibility) within the sparsified graph structure. This line of inquiry offers promising potential for coupling network simplification with design-oriented MOF screening in pharmaceutical and biomedical domains.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40537-025-01220-8.

Supplementary material 1.

Acknowledgements
We extend our sincere gratitude to Prof. Christof Wöll (Karlsruhe Institute of Technology) for his invaluable support in providing MOF materials and for inspiring us with his insights into the captivating phenomenon of the Galaxy of MOFs. His expertise and guidance have greatly enriched our research. We are truly grateful for his contributions to this work.

Author contribution
Conceptualization, M.J. (Mehrdad Jalali) and E.B.T. (Elnaz Bangian Tabrizi); methodology, M.J.; software, M.J. and M.H. (Mahboobeh Houshmand); validation, M.J., E.B.T, and M.H.; formal analysis, M.J.; investigation, M.J.; resources, M.J.; data curation, M.J.; writing—original draft preparation, M.J.; writing—review and editing, E.B.T. and M.H.; visualization, M.J.; supervision, M.J. and M.H.; project administration, M.J.; funding acquisition, M.J. All authors have read and agreed to the published version of the manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL.

Data availability
No datasets were generated or analysed during the current study.

Code availability
The source code used for the analyses presented in this study is publicly available for reproducibility and further research. The code, including scripts for data processing, graph sparsification, and machine learning models, can be accessed at the following GitHub repository: https://github.com/MehrdadJalali-KIT/InverseLinkPredcition.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests

The authors declare no competing interests.**Received:** 18 January 2025 **Accepted:** 21 June 2025
Published online: 17 July 2025

References

1. Jalali M, Wonanke AD, Wöll C. MOFGalaxyNet: a social network analysis for predicting guest accessibility in metal–organic frameworks utilizing graph convolutional networks. J Cheminform. 2023;15(1):94.

2. Jalali M, Tsotsalas M, Wöll C. MOFSocialNet: exploiting metal-organic framework relationships via social network analysis. Nanomaterials. 2022;12(4):704.
3. James SL. Metal-organic frameworks. Chem Soc Rev. 2003;32(5):276–88.
4. Chai L, Li R, Sun Y, Zhou K, Pan J. MOF-derived carbon-based materials for energy-related applications. Adv Mater. 2025;37:2413658.
5. Han Z, et al. Development of the design and synthesis of metal–organic frameworks (MOFs)–from large scale attempts, functional oriented modifications, to artificial intelligence (AI) predictions. Chem Soc Rev. 2025;54(1):367–95.
6. Ma Q, et al. Computational design of metal-organic frameworks for sustainable energy and environmental applications: bridging theory and experiment. Mater Sci Eng, B. 2025;311:117765.
7. Yang Z, Yu Q, Zhan Y, Liu J. Incorporating edge convolution and correlative self-attention into graph neural network for material properties prediction. Mach Learn Sci Technol. 2025;6(1):015020.
8. Borgatti SP, Agneessens F, Johnson JC, Everett MG. Analyzing social networks. 2024.
9. Spielman DA and Teng S-H. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, 2004; pp. 81–90.
10. Jahandoost A, Dashti R, Houshmand M, Hosseini SA. Utilizing machine learning and molecular dynamics for enhanced drug delivery in nanoparticle systems. Sci Rep. 2024;14(1):26677.
11. Groom CR, Bruno IJ, Lightfoot MP, Ward SC. The Cambridge structural database. Struct Sci. 2016;72(2):171–9.
12. Bader DA, Meyerhenke H, Sanders P, Wagner D. Graph partitioning and graph clustering. Providence: American Mathematical Society Providence; 2013.
13. Satuluri V, Parthasarathy S, Ruan Y. Local graph sparsification for scalable clustering. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, 2011; pp. 721–732.
14. Ahmed NK, Neville J, Kompella R. Network sampling: from static to streaming graphs. ACM Trans Knowl Discov Data (TKDD). 2013;8(2):1–56.
15. Shao Y, Chen L, Chen Y, Liu W. Social influence source locating based on network sparsification and stratification. Expert Syst Appl. 2022;208:118087.
16. Wu H-Y and Chen Y-L. Graph sparsification with generative adversarial network. in 2020 IEEE international conference on data mining (ICDM), IEEE. 2020; pp. 1328–1333
17. Chen Y et al. Demystifying graph sparsification algorithms in graph properties preservation. 2023. arXiv preprint arXiv:2311.12314.
18. Peng H et al.Towards sparsification of graph neural networks. in 2022 IEEE 40th International Conference on Computer Design (ICCD), IEEE. 2022; pp. 272–279
19. Aghdaei A and Feng Z. inGRASS: incremental graph spectral sparsification via low-resistance-diameter decomposition, 2024. *arXiv preprint* arXiv:2402.16990,
20. Chen Y, et al. Demystifying graph sparsification algorithms in graph properties preservation. Proc VLDB Endowment. 2023;17(3):427–40.
21. Han-huai P, Lin-wei W, Hao L, Abdollahi M. Identifying influential nodes in complex networks: a semi-local centrality measure based on augmented graph and average shortest path theory. Telecommun Syst. 2025;88(1):25.
22. Esfandiari S, Moosavi MR. Identifying influential nodes in complex networks through the k-shell index and neighborhood information. J Comput Sci. 2025;84:102473.
23. Ruan Y, Liu S, Tang J, Guo Y, Yu T. GLC: a dual-perspective approach for identifying influential nodes in complex networks," Expert Systems with Applications, 2024; p. 126292
24. Ahmadzadeh D, Jalali M, Ghaemi R, Kheirabadi M. GraphDBSCAN: optimized DBSCAN for noise-resistant community detection in graph clustering. Future Internet. 2025;17(4):150.
25. Yang Y, et al. Fuzzy-based deep attributed graph clustering. IEEE Trans Fuzzy Syst. 2023;32(4):1951–64.
26. Yang Y, Li G, Li D, Zhang J, Hu P, Hu L. Integrating fuzzy clustering and graph convolution network to accurately identify clusters from attributed graph. IEEE Transactions on Network Science and Engineering, 2024.
27. Parchas P, Papailiou N, Papadias D, Bonchi F. Uncertain graph sparsification. IEEE Trans Knowl Data Eng. 2018;30(12):2435–49.
28. Rey S, Tenorio VM, Marqués AG. Robust graph filter identification and graph denoising from signal observations. IEEE Trans Signal Process. 2023;71:3651–66.
29. Zhang M and Chen Y. Link prediction based on graph neural networks. Advances in neural information processing systems, 2018; vol. 31
30. Lü L, Zhou T. Link prediction in complex networks: a survey. Phys A Stat Mech Appl. 2011;390(6):1150–70.
31. Kumar A, Singh SS, Singh K, Biswas B. Link prediction techniques, applications, and performance: a survey. Physica A Stat Mech Appl. 2020;553:124289.
32. Arrar D, Kamel N, Lakhfif A. A comprehensive survey of link prediction methods. J Supercomput. 2024;80(3):3902–42.
33. Pétuya R, et al. Machine-learning prediction of metal-organic framework guest accessibility from linker and metal chemistry. Angew Chem Int Ed. 2022;61(9): e202114573.
34. Weininger D. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. J Chem Inf Comput Sci. 1988;28(1):31–6.
35. Bajusz D, Rácz A, Héberger K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? J Cheminform. 2015;7:1–13.
36. Cohen J. A coefficient of agreement for nominal scales. Educ Psychol Measur. 1960;20(1):37–46.
37. Yang J, Chen Y. Fast computing betweenness centrality with virtual nodes on large sparse networks. PLoS ONE. 2011;6(7): e22557.
38. Martin T, Zhang X, Newman ME. Localization and centrality in networks. Phys Rev E. 2014;90(5):052808.
39. Zhang J and Luo Y. Degree centrality, betweenness centrality, and closeness centrality in social network. in *2017* 2nd international conference on modelling, simulation and applied mathematics (MSAM2017), 2017, pp. 300–303: Atlantis press.
40. Hage P, Harary F. Eccentricity and centrality in networks. Soc Netw. 1995;17(1):57–63.

41. Bonacich P. Some unique properties of eigenvector centrality. Soc Netw. 2007;29(4):555–64.
42. Zhao B-W, et al. Regulation-aware graph learning for drug repositioning over heterogeneous biological network. Inf Sci. 2025;686:121360.
43. Su X et al. Knowledge graph neural network with spatial-aware capsule for drug-drug interaction prediction. IEEE journal of biomedical and health informatics, 2024.
44. Wang X, et al. PPISB: a novel network-based algorithm of predicting protein-protein interactions with mixed membership stochastic blockmodel. IEEE/ACM Trans Comput Biol Bioinf. 2022;20(2):1606–12.
45. Dalirinia E, Jalali M, Yaghoobi M, Tabatabaee H. Lotus effect optimization algorithm (LEA): a lotus nature-inspired algorithm for engineering design optimization. J Supercomput. 2024;80(1):761–99.
46. Dalirinia E, Yaghoobi M, Tabatabaee H, Chandna S, Jalali M. Multimodal lotus effect algorithm for engineering optimization problems. Eng Rep. 2025;7(4): e70137.

Publisher's Note