Astero, Maryam; Rousu, Juho

Enhancing atom mapping with multitask learning and symmetry-aware deep graph matching

Published in:
Journal of Cheminformatics

DOI:
10.1186/s13321-025-01030-3

Published: 01/12/2025

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY-NC-ND

Please cite the original version:
Astero, M., & Rousu, J. (2025). Enhancing atom mapping with multitask learning and symmetry-aware deep graph matching. *Journal of Cheminformatics*, *17*(1), 1-17. Article 87. https://doi.org/10.1186/s13321-025-01030-3

Astero and Rousu
Journal of Cheminformatics (2025) 17:87
https://doi.org/10.1186/s13321-025-01030-3

Journal of Cheminformatics

METHODOLOGY

Open Access

Enhancing atom mapping with multitask learning and symmetry-aware deep graph matching

Maryam Astero[1]* and Juho Rousu[1]*

Abstract

Atom mapping involves identifying the correspondence between individual atoms in reactant molecules and their counterparts in product molecules. This process is crucial for gaining deeper insight into reaction mechanisms, such as defining reaction templates and determining which chemical bonds are formed or broken during a reaction. However, reliable atom mapping data are often limited or incomplete within chemical databases, rendering manual annotation impractical for large-scale datasets. To address this limitation, we propose the Symmetry-Aware Multitask Atom Mapping Network (SAMMNet), a model designed to automatically infer atom correspondences by incorporating an auxiliary self-supervised task during training. SAMMNet employs molecular graph representations and leverages graph neural networks to capture both general and task-specific features, enabling enhanced predictive performance. Our experimental results demonstrate that the multitask learning framework, coupled with symmetry-aware atom mapping, improves accuracy and robustness in atom mapping predictions. This makes our method a promising advancement for computational chemistry and related fields.

Scientific Contribution

This study introduces SAMMNet, a novel Symmetry-Aware Multitask Atom Mapping Network, advancing atom mapping methodologies by integrating multitask learning and post-prediction symmetry refinement. Unlike prior approaches, SAMMNet leverages auxiliary self-supervised tasks to enhance molecular graph representations, improving mapping accuracy while addressing imbalanced reactions through graph padding techniques.

Keywords Atom mapping, Graph matching, Multitask learning, Graph representation learning

Introduction

In a chemical reaction, reactant molecules are converted into product molecules by rearranging their atoms. This transformation involves breaking and forming bonds, altering the structure and distribution of atoms within the molecules. The process of identifying the correspondence of atoms between reactants and products is described by atom mapping, which tracks how atoms are redistributed during the reaction.

In computer-aided synthesis, atom mapping is used to extract reaction rules from known chemical processes and predict the outcomes of new or unknown reactions.

Communicated by Barbara Zdrazil.

*Correspondence:
Maryam Astero
maryam.astero@aalto.fi
Juho Rousu
juho.rousu@aalto.fi
[1] Computer Science, Aalto University, Konemiehentie 2, 02150 Espoo, Finland

Furthermore, the importance of atom mapping extends beyond theoretical chemistry, playing a pivotal role in fields such as drug design, computational chemistry, and reaction prediction [1–4].

Despite its significance, atom mapping data is often incomplete or unavailable in many chemical reaction databases, limiting its practical utility. The labor-intensive nature of manually determining atom mappings for large datasets has driven the development of computational methods to automate this process.

Various computational approaches, both heuristic and machine learning-based, have been proposed to address the atom mapping problem. Heuristic methods, such as graph isomorphism [5, 6] and optimization-based techniques [7–9], rely on chemical principles and predefined rules to determine atom correspondences. Graph isomorphism algorithms compare molecular graphs to identify identical substructures, while optimization methods minimize bond changes or maximize structural similarity between reactants and products. Later, hybrid approaches [10, 11] that combine graph isomorphism with optimization methods have demonstrated enhanced efficiency and accuracy, particularly in handling complex reactions.

In contrast, deep learning approaches offer a data-driven alternative that excels in handling complex reactions. These models learn from large datasets, identifying atom mappings based on patterns extracted from reactants and products. Two main strategies have emerged: sequence-based and graph-based models. Sequence-based models treat atom mapping as a sequence translation problem using SMILES (Simplified Molecular Input Line Entry System) strings. A notable example is RXNMapper [12], which applies an unsupervised technique inspired by natural language processing (NLP), where attention mechanisms infer atom mappings without labeled data. However, sequence-based models face limitations due to the non-unique nature of SMILES strings and their inability to account for molecular symmetry, which can result in inconsistent atom mappings.

Graph-based models represent molecular structures as graphs, where atoms are nodes and bonds are edges, providing a representation that aligns more naturally with molecular structure. Graph-based models, such as GraphormerMapper [13] and our previous work AMNet [14], can incorporate molecular features that produce richer and more informative embeddings. However, the combined complexity of graph-based and standard transformers in GraphormerMapper poses computational challenges. AMNet, on the other hand, addresses molecular symmetry by identifying topologically equivalent atoms, thereby improving mapping accuracy for reactions with complex or symmetrical molecular structures. Despite its promise, AMNet's design does not support processing *imbalanced reactions*, where discrepancies in atom counts arise between reactants and products due to the omission of reagents, solvents, or catalysts on the product side. While such imbalances are rare in natural reactions, they frequently occur in chemical databases due to incomplete data. Thus, a robust atom mapping algorithm must be capable of handling cases where only a subset of atoms is present.

Recent efforts to improve atom mapping have explored methods to augment data-driven approaches with additional sources of knowledge. For instance, Chen et al. [15] proposed an innovative framework that combines human expertise with machine learning to enhance atom mapping accuracy in organic reactions. While this method demonstrates considerable potential, its reliance on continuous human input poses challenges related to scalability, increased costs, and the introduction of potential biases.

Building on AMNet [14], in this paper, we use molecular graphs to model reactants and products, and frame atom mapping as a graph matching problem. The goal is to identify an optimal alignment between reactant and product atoms while preserving the chemical structure and properties of the reaction.

To further improve atom mapping performance and eliminate reliance on human input, we enhance our framework by adopting multitask learning (MTL) [16]. By leveraging auxiliary self-supervised tasks derived from the data, MTL enables the model to improve atom mapping accuracy without external intervention. This approach strengthens the model's ability to learn robust, generalizable representations, leading to enhanced prediction performance while maintaining scalability and efficiency.

We expand our framework to address the inherent challenges associated with imbalanced reaction datasets, where disparities in atom counts between reactants and products often arise due to the omission of auxiliary components such as reagents, solvents, or catalysts. To mitigate these issues, we employ a padding strategy, augmenting the smaller graph (typically the product) with zero entries. This ensures that the adjacency matrices and node features of both reactants and products are dimensionally consistent, facilitating accurate pairwise similarity computations and graph alignment. By harmonizing graph dimensions, our model effectively processes imbalanced reactions, mitigating errors or biases associated with size mismatches and maintaining the integrity of atom mapping tasks.

A notable departure from our previous work with AMNet [14] lies in how symmetry-aware refinement is applied. In AMNet, symmetry detection was embedded directly into the training process, potentially constraining the learning dynamics and limiting the model's adaptability. In this work, we use symmetry detection as a post-prediction enhancement. Specifically, an adaptation of the Weisfeiler–Lehman test [17] is employed to identify molecular symmetry after generating initial mappings. This post-prediction refinement allows the model to focus on learning flexible and generalizable features during training. By refining the mappings after prediction, we improve the model's ability to capture nuanced patterns and address challenges posed by symmetric molecular structures, enhancing both accuracy and adaptability.

Furthermore, we conduct a comprehensive evaluation of three training strategies-vanilla training, transfer learning [18], and multitask learning-to assess their effectiveness on atom mapping tasks. Vanilla training serves as a baseline, relying solely on labeled data to learn atom mappings. Transfer learning introduces a pretraining phase to acquire generalizable features, followed by fine-tuning for the specific task. In contrast, multitask learning simultaneously optimizes multiple objectives, leveraging shared representations to enhance accuracy and robustness. This systematic comparison highlights the strengths and limitations of each approach, underscoring the advantages of MTL in the advancement of atom mapping in chemical reaction modeling. Our main contributions are as follows:

- Development of a Multitask Learning Approach: We propose a novel multitask learning method that leverages multiple tasks to improve atom mapping predictions.
- Graph Matching for Imbalanced Reactions: We address the challenge of imbalanced reactions in molecular graphs using graph matching techniques.
- Enhanced Atom Mapping Accuracy: We incorporate post-prediction symmetry detection to improve mapping accuracy.
- Comprehensive Comparison of Training Methods: We provide an extensive evaluation of training strategies, including vanilla training, transfer learning, and multitask learning, for atom mapping.

Preliminaries

Problem formulation

Molecules can be naturally represented as graphs, where atoms are nodes and bonds are edges. Consequently, the atom mapping task can be framed as a graph matching problem, where the goal is to find the optimal correspondence between atoms in two graphs, such as reactants and products in a chemical reaction.

In this study, we address imbalanced reactions, where the number of atoms in the reactants and products differs. This imbalance often arises because datasets are based on reactions reported in scientific literature and patents, which primarily document the main products, omitting less significant byproducts [19, 20]. As a result, these datasets overrepresent "ideal" reactions and lack details on alternate reaction pathways. To address this issue, we propose reversing the conventional mapping order. Instead of mapping atoms from reactants to products, we map atoms in products to subgraphs of the reactants. In simpler terms, this approach involves aligning the molecular graph of the products with corresponding substructures in the reactants, ensuring a more flexible and inclusive representation of the reaction.

We represent a chemical reaction as a pair of graphs – one for the reactants and one for the products, each of which may be disconnected. The task is to identify a mapping function $\mathrm{M} : V_P \rightarrow V_R$ that connects each atom in the product molecules to its corresponding atom in the reactant molecules.

Figure 1 illustrates a graph representation of an imbalanced chemical reaction, with five components on the reactant graph and one on the product graph. The mapping function M assigns a unique label to each atom in the product molecules, linking it to the corresponding atom in the reactant molecules. This mapping ensures connectivity and preserves atom types. However, because of the presence of topologically equivalent atoms, more than one valid atom mapping may exist.

Graph matching

Graph matching is the process of finding an optimal correspondence between the nodes of two graphs. Given a source graph (G_R) and a target graph (G_P), each represented by nodes V, a binary adjacency matrix A, a node feature matrix X, and an edge feature matrix E, the goal of graph matching is to establish a mapping that aligns nodes and edges between the two graphs. This alignment is represented by a binary correspondence matrix M, where each entry indicates whether a node in G_R corresponds to a node in G_P.

Graph matching can be formulated as a quadratic assignment problem (QAP) [21–24], where the objective is to maximize the similarity score between corresponding nodes. Ideally, this involves a bijective mapping, where each node in G_R corresponds uniquely to a node in G_P, allowing a similarity score between two graphs to be expressed as a distance metric between their adjacency matrices.

However, achieving a bijective mapping is not always possible. For example, if the two graphs differ in size, some nodes in the larger graph will lack correspondences

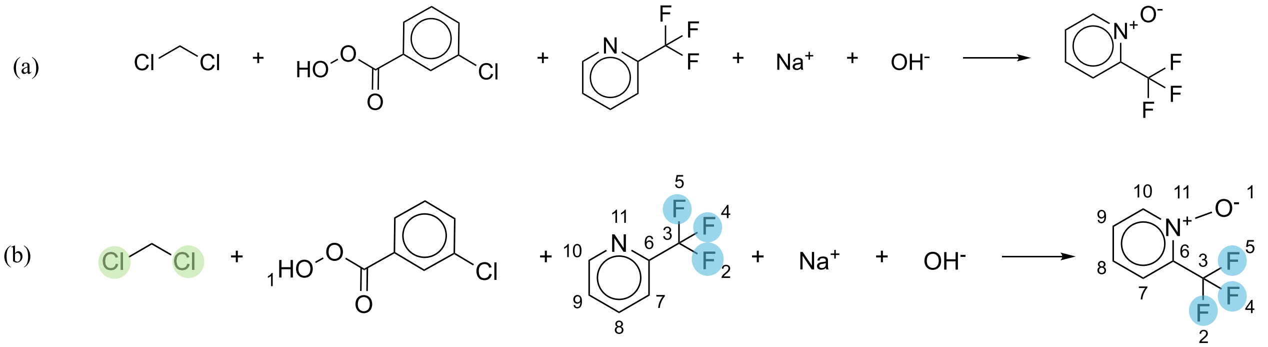

Fig. 1 **a** An example of an imbalanced chemical reaction, where auxiliary components such as solvents or reagents are omitted from the product side; **b** One possible atom mapping, in which atoms of the same color are topologically equivalent. Note that only the main reaction components are atom mapped

in the smaller graph. Additionally, symmetries within one or both graphs may lead to multiple possible correspondences for each node. These challenges introduce complexities in finding a unique matching solution, and we explore these cases further in Sect. 3.4.

To achieve node correspondence, a permutation matrix π is used to reorder the nodes in G_P to align with those in G_R. The objective is to find the permutation matrix $\pi \in 0, 1^{|V_R|\times|V_P|}$ that maximizes the similarity score, computed starting from the adjacency matrix of reactants, A_R – with elements $(A_R)_{i,j}$, and products, A_P – with elements $(A_P)_{i,j}$:

$$\pi^\star = \arg\max_{\pi} \sum_{i,j} (A_R)_{i,j}(A_P)_{\pi(i,j)}. \tag{1}$$

This discrete, non-convex optimization problem is GI-hard, and finding the global optimum is challenging. To make this problem more tractable, the discrete permutation constraint can be relaxed by replacing π with a continuous correspondence matrix $M \in [0,1]^{|V_R|\times|V_P|}$. The entry $M_{i,i'}$ indicates the probability that node i in G_R maps to node i' in G_P.

The relaxed problem then seeks an optimal correspondence matrix $M^\star$ that maximizes the similarity score between the two graphs. This can be achieved by solving equation 2 [23, 24].

$$\begin{aligned} M^\star &= \arg\max_{M} \sum_{i,j} \sum_{i',j'} (A_R)_{i,j}(A_P)_{i',j'} M_{i,i'} M_{j,j'} \\ &\text{s.t.} \quad 0 \leq \sum_{i' \in V_P} M_{i,i'} \leq 1\,, \forall i \in V_R, \\ &0 \leq \sum_{j \in V_R} M_{j,j'} \leq 1\,, \forall j' \in V_P. \end{aligned} \tag{2}$$

Graph matching models aim to learn a function that predicts the optimal correspondence between nodes in two graphs, given their respective features and structure. This involves selecting a discriminant function $f(G_R, G_P, M, w)$ that maps graph pairs (G_R, G_P) to a space of possible mappings $\mathcal{M}$, and finding an optimal function that maximizes:

$$F_w(G_R, G_P) = \arg\max_{M \in \mathcal{M}} f(G_R, G_P, M, w). \tag{3}$$

Thus, solving this optimization problem approximates the solution to the graph matching problem.

To train the function $F_w(G_R, G_P)$, the empirical risk (average loss) on a training set must be minimized. The loss function penalizes deviations between the predicted and correct correspondence matrices. For each row in the correspondence matrix, representing each atom in the reactant graph, a discrete probability distribution over corresponding atoms in the product graph is computed. This is minimized using a negative log-likelihood (NLL) loss:

$$\mathcal{L} = -\sum_{i \in V_R} \log\left(M_{i,\pi_{\mathrm{gt}}(i)}\right), \tag{4}$$

where $\pi_{gt}(i)$ represents the index of the ground truth mapping.

Graph Neural Networks have demonstrated significant potential in graph matching tasks due to their ability to learn and encode complex structural and relational information in graph data [24–27]. By iteratively updating node embeddings based on structural affinities, GNNs facilitate accurate node correspondences across graphs.

Graph neural networks

Graph Neural Networks (GNNs) are designed for graph-structured data, capturing intricate relationships between nodes. GNNs perform graph representation learning by embedding nodes or graphs into a vector space that preserves the graph's structure, with connected nodes remaining close in the embedding space while unconnected nodes are pushed apart.

In GNNs, nodes exchange information with their neighbors through a process called message passing, which updates each node's features using information from its neighbors. For molecular graphs, node features describe atomic properties, edge features represent bond types, and the adjacency matrix represents connectivity. During message passing, features of each node i and its neighbors j are aggregated as shown in Equation 5:

$$h_i^{(t)} = \text{update}(h_i^{(t-1)}, \text{aggregate}(h_i^{(t-1)}, h_j^{(t-1)}, e_{ij}^{(t-1)})), \tag{5}$$

where $h_i^{(0)}$ and $e_{ij}^{(0)}$ are initial node and edge features, respectively. The update function is differentiable, and aggregate is a permutation-invariant operator, such as mean, max, or sum.

Various GNN architectures utilize different aggregation and update functions. For example, GIN (Graph Isomorphism Networks) [28] uses summation-based aggregation, enhancing its capacity to distinguish graph structures but increasing computational demands. GCN (Graph Convolutional Networks) [29] averages neighbors' features, making it efficient for local structure learning. GraphSAGE [30] samples neighbors before aggregation, using mean, LSTM [31], or pooling. GAT (Graph Attention Networks) [32] employs attention to dynamically weigh neighbors during aggregation.

Through message passing, GNNs encode the graph's structure and features into node embeddings that capture structural and semantic information, making them effective for tasks involving graph comparison and matching.

SAMMNet: symmetry-aware multitask atom mapping network

We propose SAMMNet (Symmetry-Aware Multitask Atom Mapping Network), a novel multitask learning (MTL) framework that integrates an auxiliary task within a graph neural network (GNN) architecture with shared parameters. The shared parameters enable the model to capture intricate dependencies between the complex molecular structures of reactants and products. By leveraging MTL, SAMMNet enhances molecular representations through the simultaneous learning of complementary tasks; this ultimately enhances predictive atom mapping performance.

MTL strengthens model robustness by leveraging auxiliary tasks, which serve as an implicit form of regularization. These tasks allow the model to focus on various structural and contextual aspects of the graphs, resulting in richer feature learning. Examples of these tasks include node (atom) classification [33, 34], where the model predicts individual atom characteristics, such as atom type, to better understand local chemical environments and atomic properties. Edge prediction [30, 33, 35] focuses on identifying the presence or type of bonds between nodes, capturing critical connectivity and spatial relationships within molecular graphs. Context prediction [33] involves inferring neighboring subgraph structures based on an anchor subgraph, providing insights into subgraph-level interactions. Similarly, tasks like molecular property prediction [36, 37] enable the model to learn global graph-level features related to molecule-wide properties such as solubility or toxicity.

Among these tasks, node classification is particularly effective for atom mapping due to its strong synergy with this objective. By learning about local atomic environments and chemical properties, node classification directly complements the atom mapping process, enabling greater accuracy and robustness in identifying atom correspondences. This makes it a practical and efficient choice for enhancing graph-based molecular models.

Figure 2 illustrates the overview of the SAMMNet. Here, A_P and X_P represent the adjacency matrix and node features of the product molecules, while A_R and X_R correspond to the reactant molecules. Additionally, $\tilde{A}_P$ and $\tilde{X}_P$ denote the adjacency matrix and node features of the product molecules with masked atoms, which are used during the auxiliary node classification task in multitask learning.

Main task: atom mapping

The core objective of SAMMNet is to perform atom mapping by aligning atoms in the reactant and product molecules. The process begins by transforming reactant and product molecular structures into graph representations, where atoms serve as nodes and bonds as edges, preserving their structural and relational properties. These graph representations are then processed by a Graph Neural Network (GNN) to generate node embeddings that encapsulate molecular features. The resulting embeddings, denoted as H_p and H_R, are computed using a GNN as follows:

$$\begin{aligned} H_P &= \text{GNN}(A_P, X_P) \\ H_R &= \text{GNN}(A_P, X_P). \end{aligned} \tag{6}$$

Once node embeddings are obtained, pairwise similarity scores between nodes are calculated using the dot product of H_P and H_R, represented as $\hat{M} = \langle H_P, H_R \rangle$. This similarity matrix, $\hat{M}$, is further refined through Sinkhorn normalization to yield a doubly stochastic matrix M,

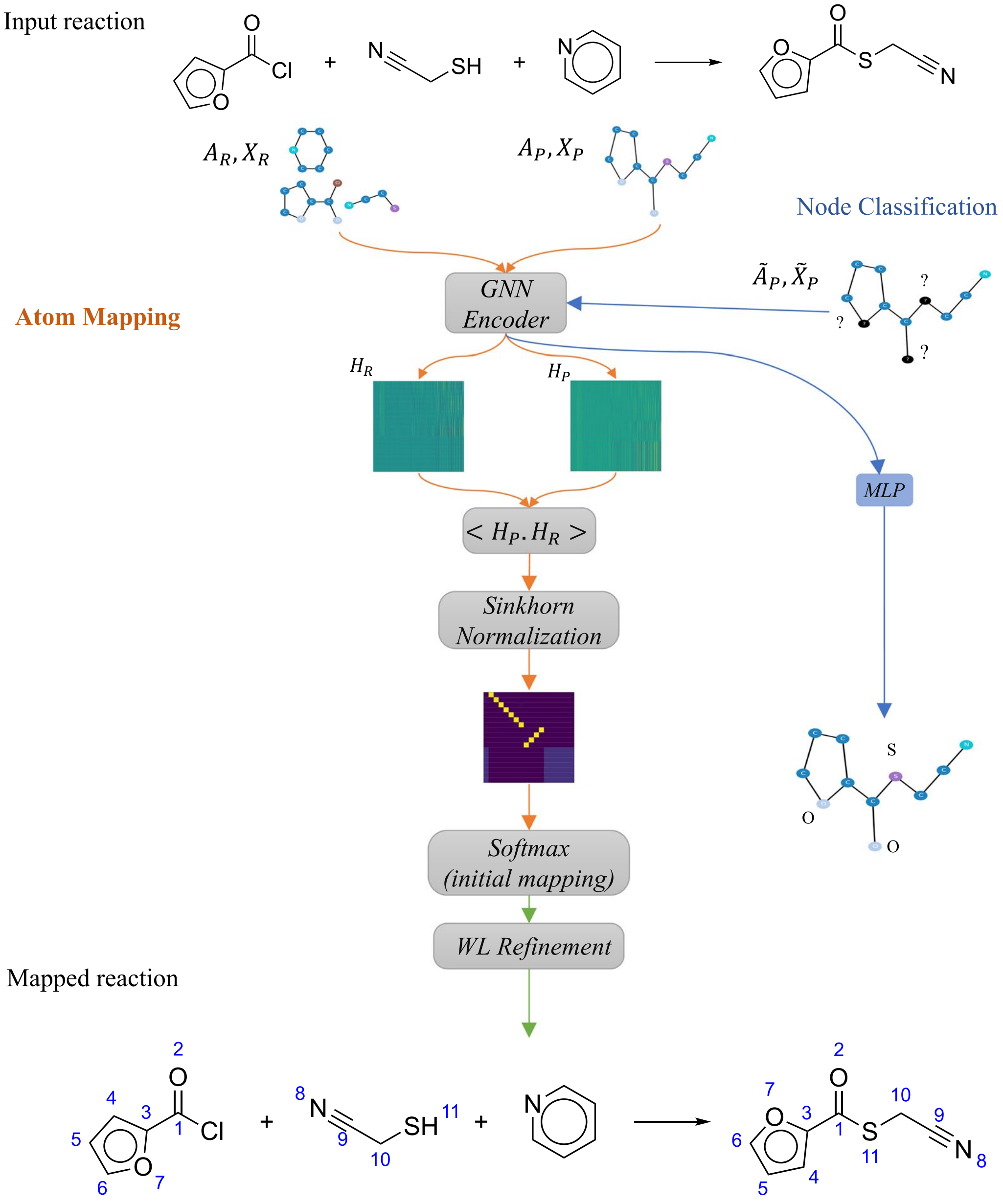

Fig. 2 Overview of the SAMMNet Framework and its Multitask Learning Strategy. The model processes input molecular graphs through a shared GNN encoder and then branches into two distinct supervision pathways. The Atom Mapping branch (orange path) computes pairwise node similarity between reactant and product atoms, refines the similarity matrix using the Sinkhorn algorithm for optimal assignment, and incorporates symmetry-aware refinement via the Weisfeiler–Lehman (WL) test (green path). The Node Classification branch (blue path) masks a subset of atom features and trains the model to predict the original atom types. The individual loss components from these tasks are combined into a weighted sum, enabling joint optimization of the entire framework

which ensures probabilistic alignment between reactant and product atoms:

$$M = \text{Sinkhorn}(\hat{M}). \tag{7}$$

The most likely correspondences are determined using the Argmax function on M, identifying optimal atom mappings. SAMMNet applies a post-prediction, symmetry-aware refinement step, leveraging the Weisfeiler–Lehman test to resolve ambiguities caused by molecular symmetry.

Auxiliary task: node classification

In SAMMNet, the auxiliary task is node classification, implemented with a masking strategy inspired by BERT

[38] and similar approaches [33, 39]. Specifically, 15% of the atoms in the product graph are masked, and the model predicts the types of these atoms using a linear classifier (MLP in Fig. 2) applied to the GNN-generated embeddings.

The node classification task enhances the SAMMNet model's performance for atom mapping by providing a synergistic relationship with the core task. By predicting atom types based on local chemical environments, node classification improves the model's understanding of atomic interactions and bonding patterns. This task enriches the model's node embeddings, which captures more nuanced structural features of molecules. Additionally, the inclusion of node classification acts as a regularization effect, balancing the learning process and preventing overfitting.

Model training and loss optimization

SAMMNet optimizes two objectives: atom mapping (AM) and node classification (NC). The total loss function is a weighted combination of the losses from these two tasks, encouraging the model to balance its learning across both objectives. The overall loss function is defined as follows:

$$\mathcal{L}_{\text{total}} = \lambda_{AM} \cdot \mathcal{L}_{\text{AM}} + \lambda_{NC} \cdot \mathcal{L}_{\text{NC}}, \tag{8}$$

where:

- $\mathcal{L}_{\text{AM}}$ is the negative log-likelihood loss for the atom mapping task, defined as:

  $$\mathcal{L}_{\text{AM}} = -\sum_{i \in V_P} \log(M_{i,\pi_{\text{gt}}(i)}), \tag{9}$$

  here $\pi_{\text{gt}}(\cdot)$ denotes the ground truth correspondence matrix, indicating the correct atom matches between product and reactant graphs.
- $\mathcal{L}_{\text{NC}}$ is the cross-entropy loss for the node classification task, computed as follows:

  $$\mathcal{L}_{\text{NC}} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i), \tag{10}$$

  in this equation, N represents the number of classes (unique atoms), y_i is the true label of the atom, and $\hat{y}_i$ is the predicted probability for class i.
- λ_{AM} and λ_{NC} are weighting factors that control the contribution of each task to the total loss.

By jointly optimizing these objectives, the model can take advantage of the information shared between tasks, resulting in an improvement in the quality of predictions.

Symmetry-aware refinement

Molecular symmetry arises when a molecule has indistinguishable components, such as atoms or groups, due to its structural arrangement. Recognizing molecular symmetry helps resolve ambiguities in atom mapping, ensuring more precise alignment between reactants and products, particularly in complex reactions involving symmetric molecules. These symmetric features result in topologically equivalent atoms, which are chemically indistinguishable because they share identical chemical environments, bonding patterns, and reactivity during chemical processes. An illustration of equivalent atoms is provided in Fig. 1b.

Accurately identifying and mapping such atoms is critical, as it significantly improves the accuracy and efficiency of atom mapping, making it a key component of chemical reaction modeling.

To identify topologically equivalent atoms, we adapted the Weisfeiler–Lehman (WL) test-a widely recognized algorithm for determining graph isomorphism. The WL test operates by iteratively refining node labels within a graph based on each node's local neighborhood structure. In our approach, we apply the WL test to a single molecular graph, treating two atoms as topologically equivalent if they share the same element and have identical three-hop neighboring atoms. Further details on this identification process are provided in [14].

Unlike AMNet [14], which integrated symmetry-aware refinement directly into the training process, our current approach applies symmetry detection as a post-prediction enhancement. This post-prediction refinement offers greater adaptability by avoiding potential constraints on learning dynamics that may arise from embedding symmetry-awareness during training. By refining the predicted mappings, we achieve more nuanced and precise alignments of atoms, thereby enhancing the overall accuracy and reliability of our atom mapping framework.

Handling imbalanced reactions

SAMMNet addresses the challenges of imbalanced reactions, where discrepancies in atom counts between reactants and products arise from missing reagents, solvents, or catalysts. To mitigate this, the smaller graph (typically the product graph) is padded with zero entries to match the size of the reactant graph. This ensures consistency in adjacency matrices and node features, facilitating accurate pairwise similarity computations and graph alignment.

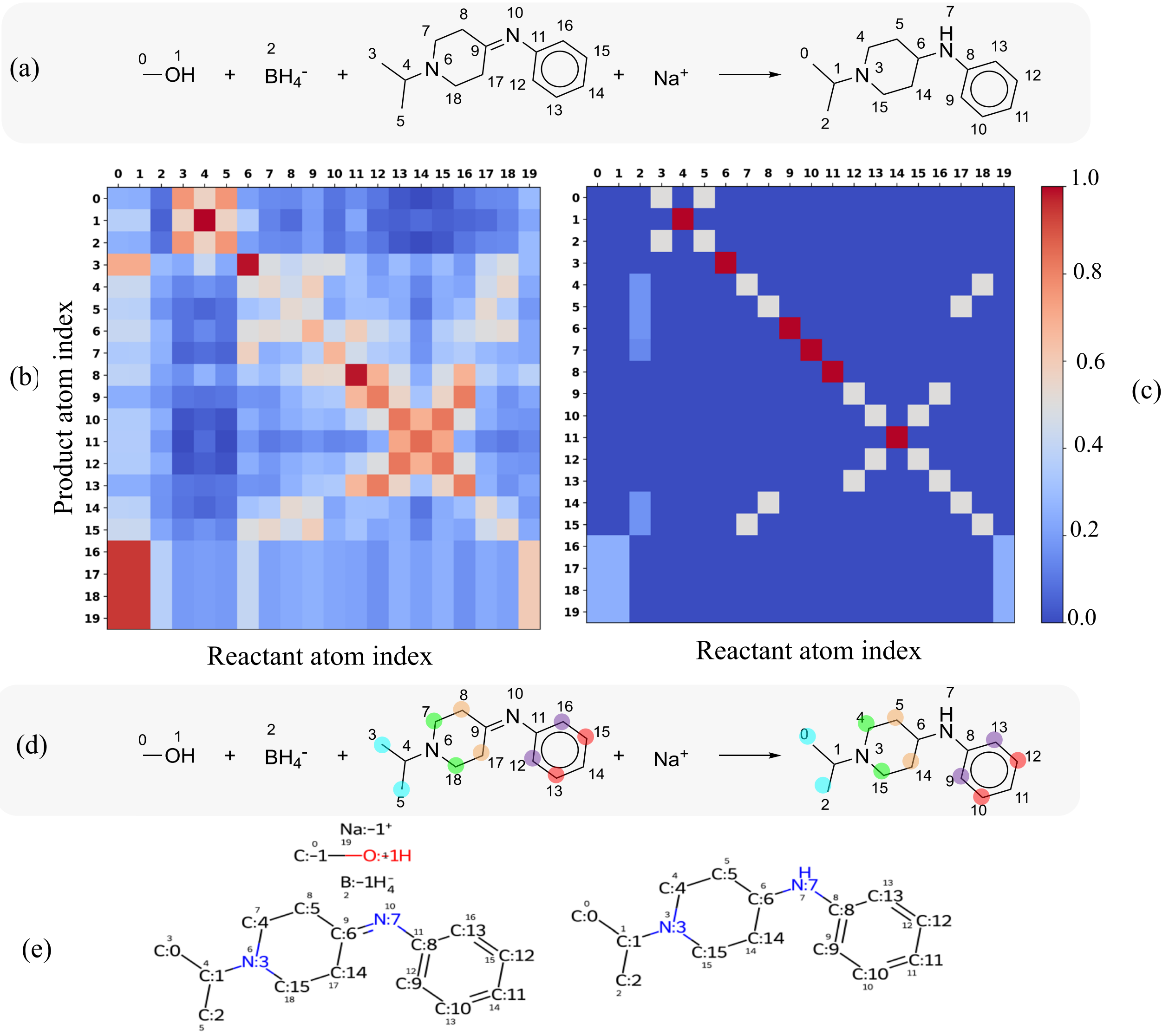

Fig. 3 Overview of the atom mapping process using the SAMMNet using GIN as the backbone. **a** A sample chemical reaction depicting reactant and product molecules, with numbers indicating atom indices. **b** The similarity matrix represents potential matches based on structural and feature similarities between atoms. **c** The soft matching output, obtained through Sinkhorn normalization of the similarity matrix, refines atom correspondences probabilistically. **d** Identification of equivalent atoms, where topologically equivalent atoms are highlighted with the same color to signify their symmetrical relationships within the molecular graph. **e** Predicted atom mappings where the small numbers indicate atom indices, while atom labels represent the final atom mapping predictions

By standardizing input dimensions, SAMMNet remains robust across both balanced and imbalanced reactions, preserving the fidelity of the atom mapping process. This flexibility makes SAMMNet particularly well-suited for processing diverse chemical datasets.

An illustrative example

Figure 3 illustrates the complete SAMMNet workflow applied to a sample chemical reaction (Fig. 3a), where the reactant and product molecules are represented as input graphs. The process begins with the computation of a similarity matrix (Fig. 3b) that evaluates potential atom matches based on structural and feature-based similarities between the reactant and product graphs. This matrix is refined through Sinkhorn normalization to produce a soft matching matrix (Fig. 3c), which probabilistically aligns atoms between the two molecular graphs.

To further enhance mapping accuracy, SAMMNet identifies equivalent atoms within the molecular graphs (Fig. 3d). By recognizing and grouping symmetrically

Table 1 Dataset splits used in this study

Dataset	Training	Validation	Testing
USPTO-50 K	40,328	4,836	4,836
Golden	–	–	1,851

equivalent atoms, this step resolves ambiguities that arise in mapping, especially for molecules with symmetric structures. Importantly, structural consistency is maintained through considerations for minimum edits. For instance, if atom 4 in the product is mapped to atom 7 in the reactant, then atom 5 in the product must align with atom 8 rather than atom 17 to preserve the molecular structure.

Finally, SAMMNet generates the predicted atom mappings (Fig. 3e), showcasing accurate and consistent correspondences between reactant and product atoms.

Experiments

Dataset

The atom mapping data used in this study is based on the USPTO-50 K dataset, originally derived by Lowe through data mining of reactions from United States Patent and Trademark Office (USPTO) patents [19]. Schneider et al. [40] refined and filtered these reactions to yield approximately 50,000 atom-mapped examples. Detailed reaction statistics and preprocessing steps are provided in Appendix 1.

During preprocessing, we removed reactions with duplicate products. Additionally, many reactions exhibit an imbalance in atom counts between reactants and products due to missing reagents, solvents, or catalysts in the product graph. To address this, we zero-padded the smaller graph (typically the product) to match the size of the corresponding reactant graph.

To ensure a robust evaluation, we randomly split the USPTO-50 K dataset into training, validation, and test sets with an 8:1:1 ratio. The validation set is used during training for model selection and early stopping. This splitting was repeated five times to account for potential data distribution variance.

To assess the model's robustness and generalization capabilities, we also tested it on the Golden dataset, which includes 1,851 curated reaction examples. For this benchmark, the model was trained solely on the USPTO-50 K training set and evaluated on the Golden dataset.

Table 1 summarizes the data usage across all experiments.

To generate molecular graphs, we used a comprehensive range of atom and bond features. These features were computed using the RDKit open-source package and encoded as one-hot vectors, which were then concatenated to form a detailed representation of the molecular structure. This feature vector encapsulates intricate details about atoms and bonds, enabling the model to capture the complexities of the molecular structures. For a complete list of the atom features, please refer to the AMNet paper [14].

Evaluation

To evaluate each model's effectiveness, we report both the average accuracy and the symmetry-aware accuracy of the predictions on the test dataset. The average atom mapping accuracy is calculated by averaging the accuracies of the predicted atom mappings for each reaction across the entire test set. Symmetry-aware accuracy is computed by first identifying topologically equivalent atoms, following a method similar to [14]. A predicted atom is considered correctly mapped if it belongs to the set of equivalent atoms and its neighbors align with the neighbors of these equivalent atoms.

To ensure the robustness and reliability of our results, we repeated each experiment using five different dataset splits. This approach minimizes the potential influence of any particularly favorable or unfavorable splits on the reported performance metrics.

Setup

To explore the generalization capabilities and robustness of our approach, we employed three different GNN architectures to establish correspondences between molecular graphs. Using various GNNs enables a comprehensive comparison of how architectural differences influence performance on atom mapping accuracy.

The choice of architecture is particularly important in multitask learning, where models must generalize effectively across multiple related tasks while maintaining robust performance. Previous studies have demonstrated that GNN architecture significantly influences the success of graph-based tasks. For instance, Hu et al. [33] showed that pretraining on certain GNN variants, such as GIN [28], can significantly enhance performance on downstream tasks. In contrast, architectures like GAT [32] may experience performance degradation post-pretraining [33]. Thus, selecting the appropriate architecture is vital for optimal results.

To ensure fair and reliable comparisons, we maintained consistent hyperparameter settings across all models, standardizing the experimental setup. After experimenting with several configurations, we determined that an embedding dimension of 512 and employing three message-passing layers yielded optimal results, as these values have been shown to provide a balanced trade-off between model performance and computational efficiency. Model optimization was performed using the ADAM optimizer with a fixed learning rate of 0.0001. To prevent overfitting, we employed early stopping during the training phase. Additionally, we incorporated the Jumping Knowledge (JK) technique [41], which aggregates node embeddings across multiple message-passing iterations. This technique enhances the model's capacity to capture complex graph structures and improves node feature representation. All models were implemented using PyTorch and the PyTorch Geometric libraries [42].

Comparison of training strategies

To comprehensively evaluate the effectiveness of our proposed multitask learning framework, we compare it against two alternative training strategies: vanilla training and transfer learning. This comparison allows us to assess the impact of including an auxiliary node classification task, as well as the benefits of pretraining in improving model performance for atom mapping.

Multitask learning (MTL)

In the multitask learning setup, the model is trained jointly on two objectives-atom mapping and node classification-throughout the entire training process. Both losses are optimized simultaneously. The goal is to enhance the model's atom mapping performance by introducing an auxiliary task that provides structural regularization and richer representations.

The atom mapping loss is optimized using the objective in Eq. 9, while the node classification component is trained using the loss in Eq. 10. The combined loss Eq. 8 encourages the GNN to learn atomic-level features that are beneficial to both tasks. This approach is fully end-to-end and does not rely on transfer or staged learning.

Vanilla training

The vanilla training baseline focuses exclusively on the atom mapping task, using a single loss function without any auxiliary tasks. The GNN processes the input molecular graphs to produce atom-level embeddings, from which a similarity matrix is computed between reactant and product atoms. This matrix is refined via the Sinkhorn algorithm to produce a valid correspondence. The training objective is to minimize the negative log-likelihood of the true atom mapping, as defined in Eq. 9. This simple yet strong baseline allows us to isolate the impact of multitask or pretraining strategies.

Transfer learning

To evaluate the benefit of task-specific pretraining, we also implement a transfer learning strategy. This approach separates training into two stages: a pretraining phase, where the model learns general molecular representations, followed by a fine-tuning phase focused on atom mapping.

During pretraining, the model performs a self-supervised node classification task using a masking strategy inspired by BERT. Specifically, 15% of the atoms in the product graph are masked, and the model learns to predict their atom types based on their surrounding chemical environment. This is optimized using the node classification loss in Eq. 10, and encourages the GNN to capture general structural features of molecules.

In the fine-tuning phase, the model weights from pretraining are used to initialize training for the atom mapping task. This phase optimizes the atom mapping loss (Eq. 9) in the same way as the vanilla model. The transfer learning strategy thus allows us to assess how well knowledge learned in a general pretraining task can improve performance on the specific downstream task of atom mapping.

Although both pretraining and fine-tuning phases use the same dataset, they serve different task objectives-node classification and atom mapping, respectively. Since the input data remains the same but the target tasks differ, there is no risk of overfitting; the model learns general molecular representations during pretraining and then specializes in atom-to-atom correspondence during fine-tuning.

Appendices 2 and 3 provide architectural and implementation details for both vanilla and transfer learning models.

Results and discussion

We compared the MTL approach against vanilla and transfer learning (TL) strategies, followed by a benchmark evaluation of our best-performing model using the golden dataset. The evaluation metrics included initial atom mapping accuracy and symmetry-aware accuracy, which accounts for the correct assignment of topologically equivalent atoms. To ensure robustness and reliability, each experiment was repeated five times with different dataset splits, and the average results were reported to minimize the impact of potential data biases.

Performance on USPTO-50 K dataset

The results on the USPTO-50 K dataset demonstrate the superiority of multitask learning (MTL) over both vanilla and transfer learning (TL) approaches across various GNN architectures. Table 2 presents a comprehensive comparison, showing that MTL consistently outperforms its counterparts in both initial and symmetry-aware accuracy.

MTL consistently demonstrates superior performance, with the GIN model achieving an initial accuracy of 88.51% and a symmetry-aware accuracy of 97.37%, marking a substantial improvement. Similarly, both GCN and GraphSAGE models showed notable

Table 2 Performance comparison on USPTO-50 K dataset across different training approaches

Model	Initial accuracy (%) ± std	Symmetry-aware accuracy (%) ± std
SAMMNet		
GIN	**88.51** ± 0.07	**97.37** ±0.06
GCN	87.18 ± 0.12	95.66 ± 0.08
GraphSAGE	88.20 ± 0.11	97.02 ± 0.05
Vanilla		
GIN	87.64 ± 0.09	96.46 ± 0.06
GCN	86.34 ± 0.11	94.89 ± 0.08
GraphSAGE	86.21 ±0.1	95.32 ±0.07
Transfer learning		
GIN	86.65 ± 0.1	95.5 ± 0.04
GCN	85.15 ± 0.12	93.73 ± 0.08
GraphSAGE	84.78 ± 0.14	93.87 ± 0.05

Bold values indicate the best-performing results for each evaluation metric

Table 3 SAMMNet performance for different λ_{AM} and λ_{NC} configurations

λ_{AM}	λ_{NC}	Description	Initial accuracy (%) ± std	Symmetry-aware accuracy (%) ± std
0.5	0.5	Balanced MTL	87.3 ± 0.09	96.2 ± 0.14
0.3	0.7	Emphasis on NC	86.7 ± 0.1	95.0 ± 0.05
0.7	0.3	Emphasis on AM	**88.51** ± 0.07	**97.37** ± 0.06
1.0	0.0	Pure Vanilla training	87.6 ± 0.08	96.5 ± 0.07
0.0	1.0	Pure node classification	69.9 ± 1.2	76.6 ± 0.9

Bold values indicate the best-performing results for each evaluation metric

gains under MTL, suggesting that simultaneous training on multiple tasks helps the models develop more robust and generalized feature representations. The enhanced performance of MTL can be attributed to its ability to learn shared representations, which reduces overfitting and improves generalization by exposing the models to related tasks. These results underscore the effectiveness of multitask learning in handling complex molecular structures and improving atom mapping accuracy. These findings align with previous studies, such as [43–45], which highlight the success of MTL when the auxiliary tasks are well-aligned with the graph structure.

In contrast, the vanilla approach yielded lower results. The GIN model achieved an initial accuracy of 87.64% and a symmetry-aware accuracy of 96.46%, making it the top performer in this category. Meanwhile, GCN and GraphSAGE models attained slightly lower initial accuracies of 86.34% and 86.21%, respectively.

The transfer learning (TL) approach resulted in lower initial and symmetry-aware accuracies across all models compared to the vanilla approach. While GIN remained the top performer under TL, its performance slightly declined, reaching an initial accuracy of 86.65% and a symmetry-aware accuracy of 95.5%. Both GCN and GraphSAGE showed similar reductions in accuracy, indicating that the transferred knowledge might not fully align with the atom mapping task. This could be due to challenges in domain adaptation and the risk of catastrophic forgetting during fine-tuning. These observations are consistent with findings in existing literature, such as [39], which question the assumption that GNN pretraining is universally beneficial for molecular representations.

Balancing task contributions

To assess the impact of balancing the contributions of atom mapping (AM) and node classification (NC) tasks within the SAMMNet framework, we conducted experiments by varying the values of λ_{AM} (weight for the atom mapping loss) and λ_{NC} (weight for the node classification loss) in our multitask learning objective function. The goal of these experiments was to understand the trade-offs between the two tasks and evaluate how different weight configurations influence model performance. We use GIN as the GNN backbone. The other hyperparameter remained unchanged.

The performance metrics for each configuration are summarized in Table 3.

The results in Table 3 demonstrate that the highest performance was achieved when placing greater emphasis on the atom mapping task ($\lambda_{AM} = 0.7$, $\lambda_{NC} = 0.3$). This finding indicates that while auxiliary tasks, such as node classification, provide useful contextual information and regularization benefits, prioritizing the primary task of atom mapping is critical for maximizing accuracy. Emphasizing atom mapping allows the model to focus more intensively on learning precise atom correspondences, which is essential for the overall success of the framework.

In contrast, balanced multitask learning ($\lambda_{AM} = 1$, $\lambda_{NC} = 1$) yielded high but slightly lower performance, suggesting that while the auxiliary task aids learning, an equal weighting may dilute the model's attention on the primary task. Similarly, configurations with a heavier emphasis on node classification ($\lambda_{AM} = 0.3$, $\lambda_{NC} = 0.7$) underperformed compared to the atom mapping-focused setting, indicating that auxiliary tasks alone cannot compensate for the loss of primary task focus.

Vanilla training showed slightly lower performance, emphasizing the necessity of integrating auxiliary tasks

Table 4 Symmetry-aware accuracy of SAMMNet and AMNet for different configurations

Configuration	SAMMNet (%) ± std	AMNet (%) ± std
Train/Test on USPTO-50 K	97.37 ± 0.06	NA
Train on USPTO-15 K/Test on USPTO-50 K	91.78 ± 0.09	NA
Train on USPTO-50 K/Test on USPTO-15 K	95.42 ± 0.12	NA
Train/Test on USPTO-15 K	**98.02**± 0.05	97.30 ± 0.10

"NA" denotes that the value is unavailable for the configuration

Bold values indicate the best-performing results for each evaluation metric

to enrich the learning process. Pure node classification ($\lambda_{AM} = 0$, $\lambda_{NC} = 1$) demonstrated limited performance on the primary task, emphasizing the need for balance in multitask learning.

Impact of reaction completeness on SAMMNet performance: a comparison with AMNet

In this subsection, we explore the impact of reaction completeness (balanced vs. imbalanced reactions) on SAMMNet's training and testing performance. To provide a comprehensive analysis, we examine how variations in reaction completeness influence the model's ability to map atoms accurately under different conditions. This experiment also allows to compare SAMMNet with AMNet, our previous proposal specifically designed for balanced reactions, to evaluate SAMMNet's flexibility and ability to generalize beyond controlled conditions.

The experiments reported in previous sections are performed by using the USPTO-50 K dataset, which predominantly consists of imbalanced reactions and is therefore unsuitable for testing AMNet. To benchmark SAMMNet against AMNet under balanced conditions, we employed the USPTO-15 K dataset, which contains 15,000 reactions with complete atom correspondence [14].

We evaluated SAMMNet and AMNet using the following configurations:

1. Train/Test on USPTO-50 K (imbalanced reactions).
2. Train on USPTO-15 K (balanced reactions)/Test on USPTO-50 K.
3. Train on USPTO-50 K/Test on USPTO-15 K.
4. Train/Test on USPTO-15 K (balanced reactions).
5. AMNet results from [14] on USPTO-15 K (balanced reactions).

Table 4 summarizes the performance of SAMMNet and AMNet across various training and testing configurations, emphasizing the impact of reaction completeness. When trained and tested on the imbalanced USPTO-50 K dataset, SAMMNet achieves a high symmetry-aware accuracy of 97.37%, demonstrating its robustness in handling typical mixed datasets. Its performance declines to 91.78% when trained on the balanced USPTO-15 K and tested on the mixed USPTO-50 K dataset. While this level of accuracy remains suitable for practical applications, this drop suggests that balanced training data alone may not sufficiently prepare the model for the challenges of imbalanced datasets.

Notably, SAMMNet achieves its highest accuracy of 98.02% when trained and tested on the balanced USPTO-15 K dataset, outperforming AMNet (97.30%). This result underscores SAMMNet's ability to capture richer molecular representations through its multitask learning framework.

Benchmarking SAMMNet: performance evaluation on the golden dataset against state-of-the-art methods

To ensure a fair comparison between SAMMNet and a state-of-the-art method, we selected the Golden dataset to guarantee no overlap between the training and test sets across models. To further evaluate the effectiveness of our proposed multitask learning model, we compared it specifically with RXNMapper, a state-of-the-art atom mapping method, using the golden dataset. We chose to focus on RXNMapper for this comparison due to its status as a widely recognized and popular baseline in the field of automated atom mapping, supported by its strong performance on large datasets.

RXNMapper maps product atoms to reactant atoms, often resulting in an unintended permutation of atom order. To ensure a fair comparison between RXNMapper's predictions and manually curated data, we standardized its output to mitigate the effects of this permutation, similar to our prior approach outlined in [14]. Additional details on the standardization process can also be found in [14].

We assessed each method's accuracy by evaluating the complete alignment of predicted atom mappings with the ground truth mapped reactions. Specifically, a method was considered accurate if the predicted atom correspondence matched exactly with the ground truth correspondences. Our proposed model achieved a symmetry-aware accuracy of 86.3% in atom mapping predictions, while RXNMapper correctly mapped atoms for 84.5% of reactions.

Conclusion and future work

This study highlights the efficacy of multitask learning strategies for complex atom mapping tasks by systematically comparing it, vanilla training, and transfer learning approaches.

Our findings demonstrate that MTL is a powerful framework for enhancing graph-based models by mitigating overfitting, capturing nuanced relationships, and improving accuracy, particularly when combined with symmetry-aware atom mapping. SAMMNet's multitask paradigm, which simultaneously trains on node classification and atom mapping tasks, enables the development of richer, more generalized molecular representations. This dual-task approach leverages inherent regularization, reduces overfitting risks, and enhances generalizability, even in complex reactions, highlighting its versatility across diverse scenarios.

Moving forward, we plan to extend the application of MTL to a broader range of molecular datasets and more complex auxiliary tasks. Additionally, we aim to develop strategies to minimize domain shift and mitigate catastrophic forgetting within transfer learning frameworks, thereby enhancing the performance of graph-based models across various tasks. This line of research holds the potential to drive significant advancements in chemical reaction modeling and related scientific fields.

Appendix 1: USPTO-50k dataset statistics

This section provides an overview of atom and molecule distributions in reactants and products across a dataset of chemical reactions, highlighting key patterns and variations (Figs. 4, 5, 6).

Appendix 2: Vanilla model architecture

The vanilla model focuses exclusively on predicting atom correspondences, without leveraging auxiliary tasks or pretraining. This model refines atom mappings through a similarity function and Sinkhorn normalization, which adjusts the correspondences matrix into a doubly stochastic format, ensuring each product node is uniquely matched with a reactant node.

Figure 7 illustrates the architecture of the vanilla model, detailing how it processes molecular graphs to predict atom mappings.

Appendix 3: Transfer learning model architecture

The transfer learning model adapts a Graph Neural Network to leverage pre-trained representations and adapt them for atom mapping. This approach begins with a pre-training phase on node classification to capture general molecular features. The pre-trained GNN is then fine-tuned on the specific atom mapping task using the

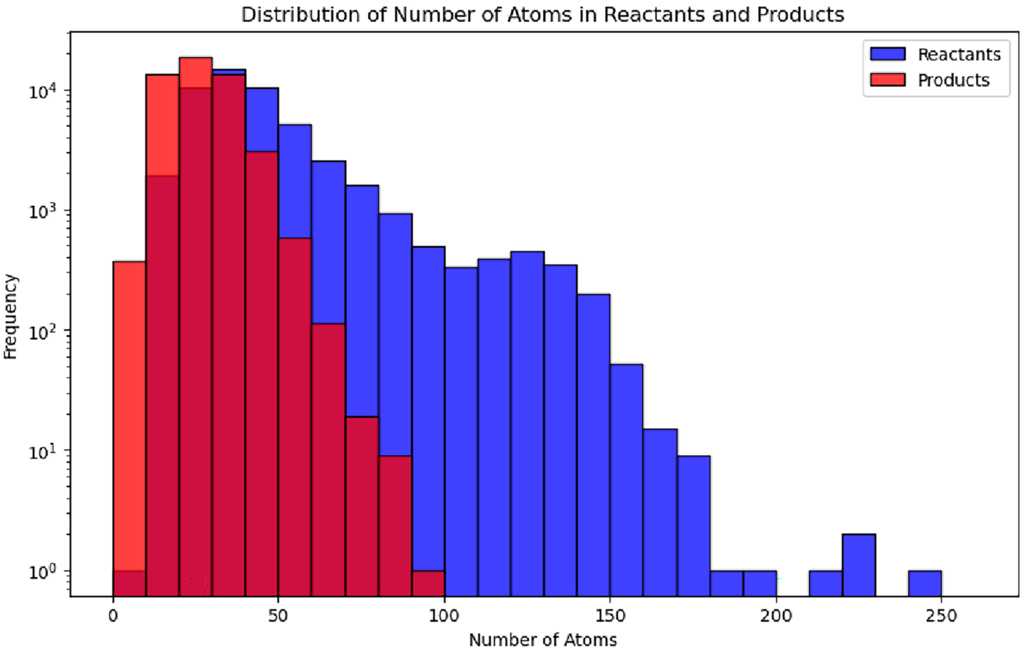

Fig. 4 Logarithmic distribution of atom counts in reactants and products

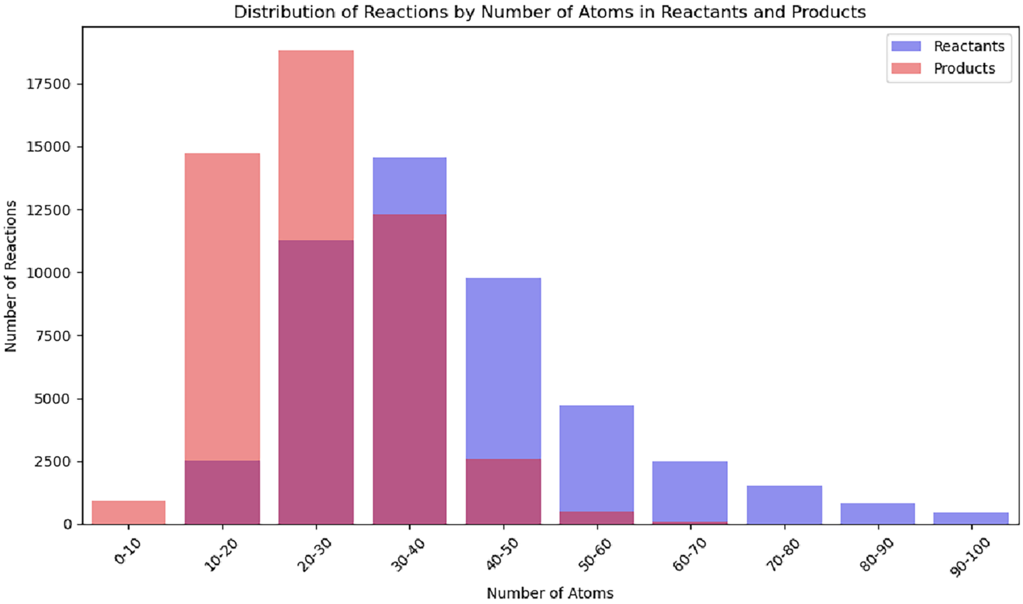

Fig. 5 Distribution of reactions binned by the number of atoms in reactants and products

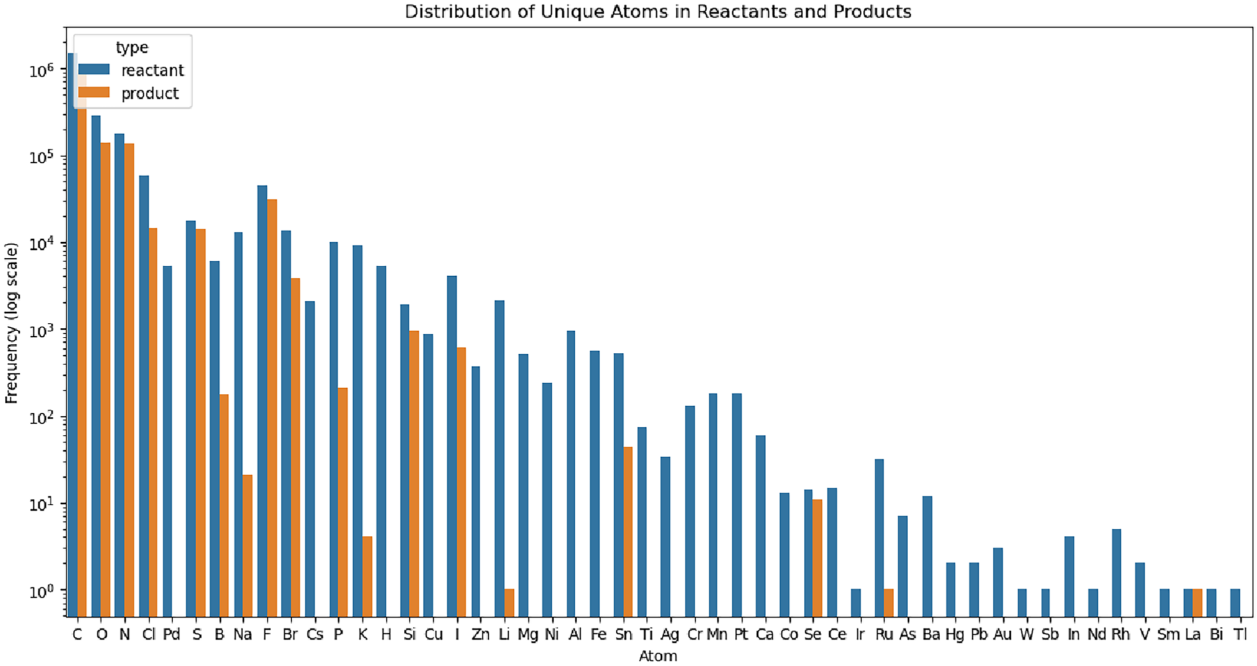

Fig. 6 Logarithmic distribution of atom frequencies in reactants and products. The bar heights indicate the relative abundance of each atom, scaled to log to accommodate a wide range of frequencies

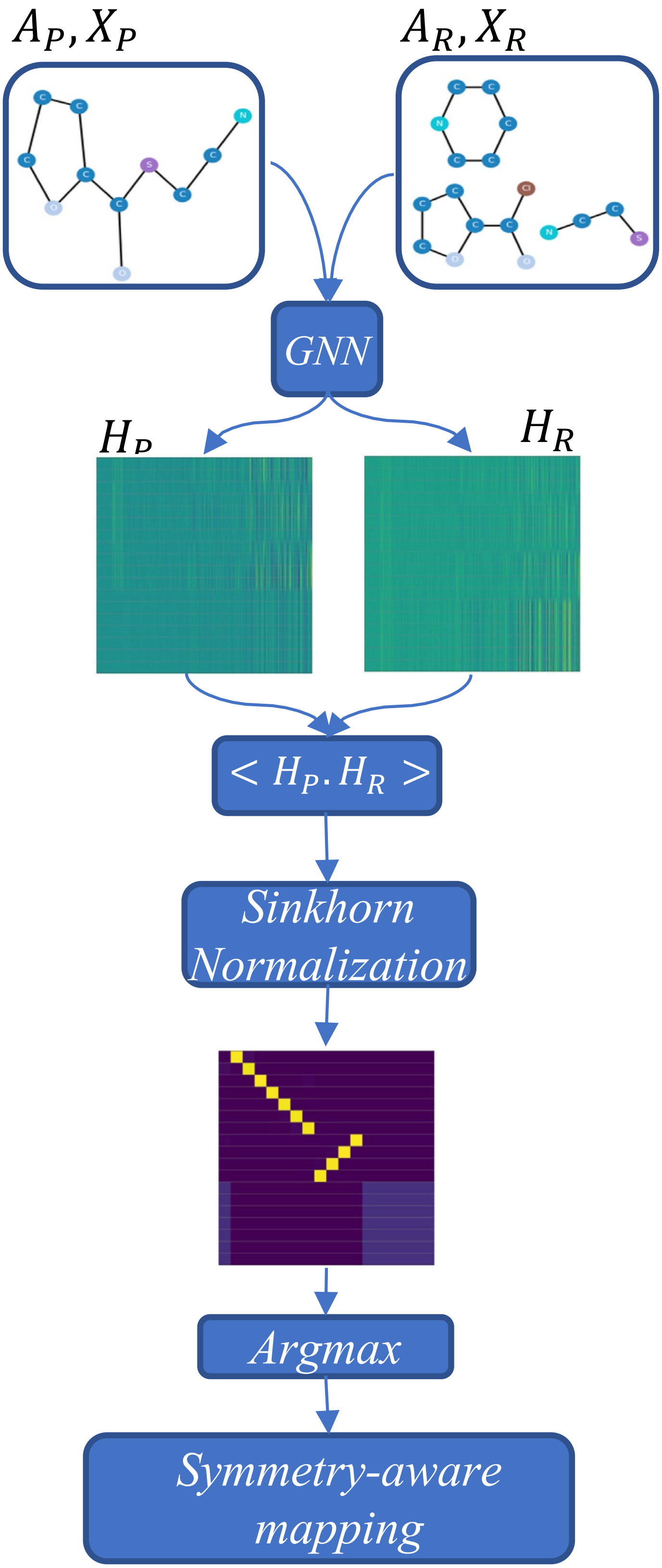

Fig. 7 Overview of the vanilla model architecture

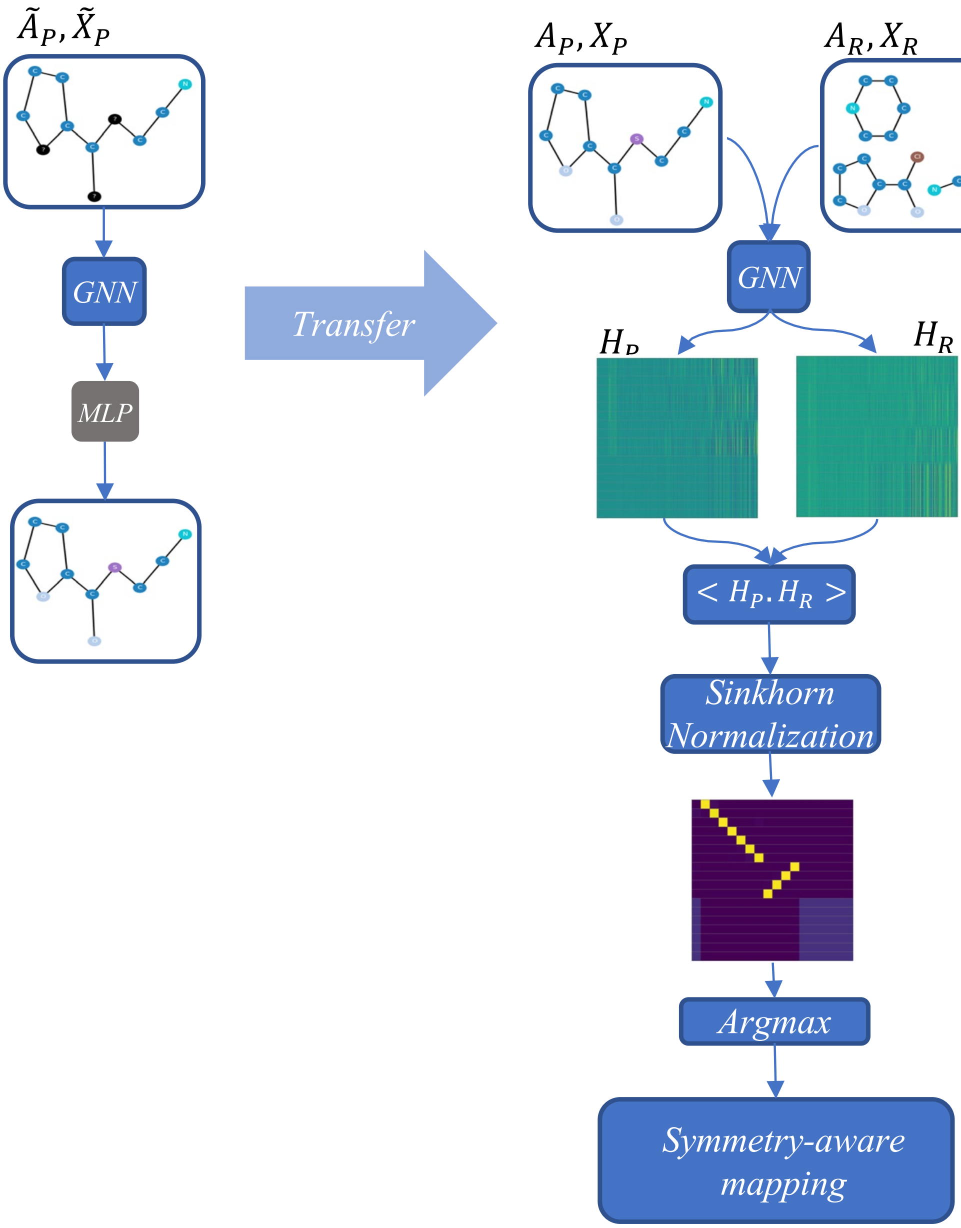

Fig. 8 Overview of the transfer learning model architecture

transfer learning technique. This process allows the model to leverage prior knowledge and generalize better to new, unseen datasets.

Figure 8 illustrates the architecture of the transfer learning model, showing how it processes molecular graphs and adapts pre-trained features for accurate atom mapping.

Acknowledgements

M.A. acknowledges Elena Casiraghi for her invaluable assistance in reviewing this work and providing constructive feedback. We acknowledge the generous support from the Wihuri Foundation as well as the Jane and Aatos Erkko Foundation (BIODESIGN project), which contributed to the advancement of this study. Additionally, this research has in part been funded by the Research Council of Finland (Grants 339421 and 345802).

Author contributions

M.A. contributed to the conceptualization, model development, experimental analysis, and manuscript preparation. J.R. provided support in conceptualization, supervision, and critical review of the manuscript. All authors have carefully reviewed and approved the final version of the manuscript.

Availability of data and materials

For further reference, the code used in this study is available on GitHub at https://github.com/maryamastero/SAMMNet.

Declarations

Competing interests

The authors declare no competing interests.

Received: 17 December 2024 Accepted: 12 May 2025
Published online: 30 May 2025

References

1. Jin W, Coley C, Barzilay R, Jaakkola T (2017) Predicting organic reaction outcomes with Weisfeiler–Lehman network. In: Advances in neural information processing systems, vol 30
2. Coley CW, Green WH, Jensen KF (2018) Machine learning in computer-aided synthesis planning. Acc Chem Res 51(5):1281–1289
3. Acharyya RK, Rej RK, Nanda S (2018) Exploration of ring rearrangement metathesis reaction: a general and flexible approach for the rapid construction [5, n]-fused bicyclic systems en route to linear triquinanes. J Org Chem 83(4):2087–2103
4. David L, Thakkar A, Mercado R, Engkvist O (2020) Molecular representations in ai-driven drug discovery: a review and practical guide. J Cheminf 12(1):56
5. Raymond JW, Willett P (2002) Maximum common subgraph isomorphism algorithms for the matching of chemical structures. J Comput Aided Mol Des 16:521–533
6. Hattori M, Okuno Y, Goto S, Kanehisa M (2003) Heuristics for chemical compound matching. Genome Inform 14:144–153
7. Heinonen M, Lappalainen S, Mielikäinen T, Rousu J (2011) Computing atom mappings for biochemical reactions without subgraph isomorphism. J Comput Biol 18(1):43–58
8. Jochum C, Gasteiger J, Ugi I (1980) The principle of minimum chemical distance (pmcd). Angew Chem Int Ed Engl 19(7):495–505
9. Mann M, Nahar F, Schnorr N, Backofen R, Stadler PF, Flamm C (2014) Atom mapping with constraint programming. Algorithms Mol Biol 9:1–12
10. Fooshee D, Andronico A, Baldi P (2013) Reactionmap: an efficient atom-mapping algorithm for chemical reactions. J Chem Inf Model 53(11):2812–2819
11. Jaworski W, Szymkuć S, Mikulak-Klucznik B, Piecuch K, Klucznik T, Kaźmierowski M, Rydzewski J, Gambin A, Grzybowski BA (2019) Automatic mapping of atoms across both simple and complex chemical reactions. Nat Commun 10(1):1434
12. Schwaller P, Hoover B, Reymond J-L, Strobelt H, Laino T (2021) Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. Sci Adv 7(15):eabe4166
13. Nugmanov R, Dyubankova N, Gedich A, Wegner JK (2022) Bidirectional graphormer for reactivity understanding: neural network trained to reaction atom-to-atom mapping task. J Chem Inf Model 62(14):3307–3315
14. Astero M, Rousu J (2024) Learning symmetry-aware atom mapping in chemical reactions through deep graph matching. J Cheminf 16(1):46
15. Chen S, An S, Babazade R, Jung Y (2024) Precise atom-to-atom mapping for organic reactions via human-in-the-loop machine learning. Nat Commun 15(1):2250
16. Caruana R (1997) Multitask learning. Mach Learn 28:41–75
17. Weisfeiler B, Leman A (1968) The reduction of a graph to canonical form and the algebra which appears therein.nti. Series 2(9):12–16
18. Bozinovski S (1976) Reminder of the first paper on transfer learning in neural networks. Informatica 44(3):2020
19. Lowe DM (2012) Extraction of chemical structures and reactions from the literature. Ph.D. thesis, University of Cambridge
20. Lin A, Dyubankova N, Madzhidov TI, Nugmanov RI, Verhoeven J, Gimadiev TR, Afonina VA, Ibragimova Z, Rakhimbekova A, Sidorov P et al (2022) Atom-to-atom mapping: a benchmarking study of popular mapping algorithms and consensus strategies. Mol Inf 41(4):2100138
21. Cho M, Alahari K, Ponce J (2013) Learning graphs to match. In: Proceedings of the IEEE international conference on computer vision, pp 25–32
22. Gold S, Rangarajan A (1996) A graduated assignment algorithm for graph matching. IEEE Trans Pattern Anal Mach Intell 18(4):377–388
23. Caetano TS, McAuley JJ, Cheng L, Le QV, Smola AJ (2009) Learning graph matching. IEEE Trans Pattern Anal Mach Intell 31(6):1048–1058
24. Fey M, Lenssen JE, Morris C, Masci J, Kriege NM (2020) Deep graph matching consensus. arXiv preprint arXiv:2001.09621
25. Li Y, Gu C, Dullien T, Vinyals O, Kohli P (2019) Graph matching networks for learning the similarity of graph structured objects. In: International conference on machine learning. PMLR, pp 3835–3845
26. Yan J, Yang S, Hancock ER (2020) Learning for graph matching and related combinatorial optimization problems. In: International joint conferences on artificial intelligence organization, pp 4988–4996
27. Chen H, Luo Z, Zhang J, Zhou L, Bai X, Hu Z, Tai C-L, Quan L (2021) Learning to match features with seeded graph matching network. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6301–6310
28. Xu K, Hu W, Leskovec J, Jegelka S (2018) How powerful are graph neural networks? arXiv preprint arXiv:1810.00826
29. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
30. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in neural information processing systems, vol 30
31. Hochreiter S (1997) Long short-term memory. Neural computation. MIT-Press, London
32. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. arXiv preprint arXiv:1710.10903
33. Hu W, Liu B, Gomes J, Zitnik M, Liang P, Pande V, Leskovec J (2019) Strategies for pre-training graph neural networks. arXiv preprint arXiv:1905.12265
34. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. In: International conference on machine learning. PMLR, pp 1263–1272
35. Kipf TN, Welling M (2016) Variational graph auto-encoders. arXiv preprint arXiv:1611.07308
36. Gasteiger J, Groß J, Günnemann S (2020) Directional message passing for molecular graphs. arXiv preprint arXiv:2003.03123
37. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V (2018) Moleculenet: a benchmark for molecular machine learning. Chem Sci 9(2):513–530
38. Devlin J (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
39. Sun R, Dai H, Yu AW (2022) Does gnn pretraining help molecular representation? Adv Neural Inf Process Syst 35:12096–12109
40. Schneider N, Stiefl N, Landrum GA (2016) What's what: the (nearly) definitive guide to reaction role assignment. J Chem Inf Model 56(12):2336–2346
41. Xu K, Li C, Tian Y, Sonobe T, Kawarabayashi K, Jegelka S (2018) Representation learning on graphs with jumping knowledge networks. In: International conference on machine learning. PMLR, pp 5453–5462
42. Fey M, Lenssen JE (2019) Fast graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428
43. You Y, Chen T, Wang Z, Shen Y (2020) When does self-supervision help graph convolutional networks? In: International conference on machine learning. PMLR, pp 10871–10880
44. Dey V, Ning X (2024) Enhancing molecular property prediction with auxiliary learning and task-specific adaptation. J Cheminf 16(1):85
45. Liu S, Qu M, Zhang Z, Cai H, Tang J (2022) Structured multi-task learning for molecular property prediction. In: International conference on artificial intelligence and statistics. PMLR, pp 8906–8920

Publisher's Note