Chemical complexity challenge: Is multi-instance machine learning a solution?

Dmitry Zankov, Timur Madzhidov, Alexandre Varnek, Pavel Polishchuk

Received: 8 January 2023 | Revised: 1 November 2023 | Accepted: 7 November 2023

DOI: 10.1002/wcms.1698

ADVANCED REVIEW

WIREs COMPUTATIONAL MOLECULAR SCIENCE WILEY

Chemical complexity challenge: Is multi-instance machine learning a solution?

Dmitry Zankov[1] | **Timur Madzhidov**[2] | **Alexandre Varnek**[1,3] | **Pavel Polishchuk**[4]

[1]ICReDD, Hokkaido University, Sapporo, Japan

[2]Chemistry Solutions, Elsevier, Oxford, United Kingdom

[3]Laboratory of Chemoinformatics, University of Strasbourg, Strasbourg, France

[4]Institute of Molecular and Translational Medicine, Faculty of Medicine and Dentistry, Palacky University Olomouc, Olomouc, Czech Republic

Correspondence
Pavel Polishchuk, Institute of Molecular and Translational Medicine, Faculty of Medicine and Dentistry, Palacky University Olomouc, Olomouc, Czech Republic.
Email: pavlo.polishchuk@upol.cz

Funding information
European Regional Development Fund, Grant/Award Number: CZ.02.1.01/0.0/0.0/16_019/0000868; Ministerstvo Školství, Mládeže a Tělovýchovy, Grant/Award Number: CZ.02.1.01/0.0/0.0/18_046/0016118

Edited by: Peter R. Schreiner, Editor-in-Chief

Abstract

Molecules are complex dynamic objects that can exist in different molecular forms (conformations, tautomers, stereoisomers, protonation states, etc.) and often it is not known which molecular form is responsible for observed physicochemical and biological properties of a given molecule. This raises the problem of the selection of the correct molecular form for machine learning modeling of target properties. The same problem is common to biological molecules (RNA, DNA, proteins)—long sequences where only key segments, which often cannot be located precisely, are involved in biological functions. Multi-instance machine learning (MIL) is an efficient approach for solving problems where objects under study cannot be uniquely represented by a single instance, but rather by a set of multiple alternative instances. Multi-instance learning was formalized in 1997 and motivated by the problem of conformation selection in drug activity prediction tasks. Since then MIL has found a lot of applications in various domains, such as information retrieval, computer vision, signal processing, bankruptcy prediction, and so on. In the given review we describe the MIL framework and its applications to the tasks associated with ambiguity in the representation of small and biological molecules in chemoinformatics and bioinformatics. We have collected examples that demonstrate the advantages of MIL over the traditional single-instance learning (SIL) approach. Special attention was paid to the ability of MIL models to identify key instances responsible for a modeling property.

This article is categorized under:

Data Science > Chemoinformatics

Data Science > Artificial Intelligence/Machine Learning

KEYWORDS

artificial intelligence, bioinformatics, chemoinformatics, machine learning, multi-instance learning

1 | INTRODUCTION

In "structure–property" modeling each molecule is encoded by a set of numerical chemical descriptors used as an input to machine learning (ML) algorithms establishing a correlation between descriptors and the property or biological activity. One of the key limitations of traditional "structure–property" modeling is the requirement that each molecule has to be represented by a single instance with a fixed conformation, stereoconfiguration, protonation, and tautomeric states and associated with a single vector of descriptors. However, a molecule is a dynamic object that simultaneously exists in many forms/instances in equilibrium, which raises the problem of the selection of molecular form(s) responsible for the observed property. This creates a contradiction that is resolved most often by a representation of molecules as two-dimensional (2D) molecular graphs as well as through the standardization of molecular forms whose importance was repeatedly shown.[1] Descriptors derived from this representation describe mainly the atomic composition and topology of a chosen fixed molecular form. The 2D descriptors ignore the spatial molecular structure of compounds and their conformational flexibility but to some extent can encode stereoconfiguration. However, they cannot represent mixtures of stereoisomers which is a highly important issue for drug development because different stereoisomers can trigger different responses and it is not always possible to estimate which stereoisomer is preferable, for example, with molecular docking (the three-dimensional [3D] structure of a protein can be unavailable). Canonicalization (canonical tautomeric representation, molecule neutralization, etc.) is a somewhat artificial choice of molecular representation because the equilibrium or transition between molecular forms may exist. Therefore, some important structural information that could increase the predictive ability of models may be lost. The modeling of such systems today is either impossible or very difficult.

A similar problem exists in the modeling of functions of biological molecules (RNA, DNA, proteins), that are sequences of monomer units (amino acids or nucleotides). However, only particular segments of these sequences are responsible for the interaction between biological molecules, and experimental information on the exact location of key segments is often not available. This also leads to the problem of many alternative representations of biological molecules, which is often neglected in traditional modeling approaches.

This problem can be handled by multi-instance machine learning (MIL).[2] The main idea of MIL[2] is to represent an object as a set of alternative instances (called a *bag*), each encoded by its vector of features. In contrast to traditional single-instance learning (SIL) (Figure 1a), the task is to establish a correlation between the *bag* of the instances and the *bag* label (Figure 1b).

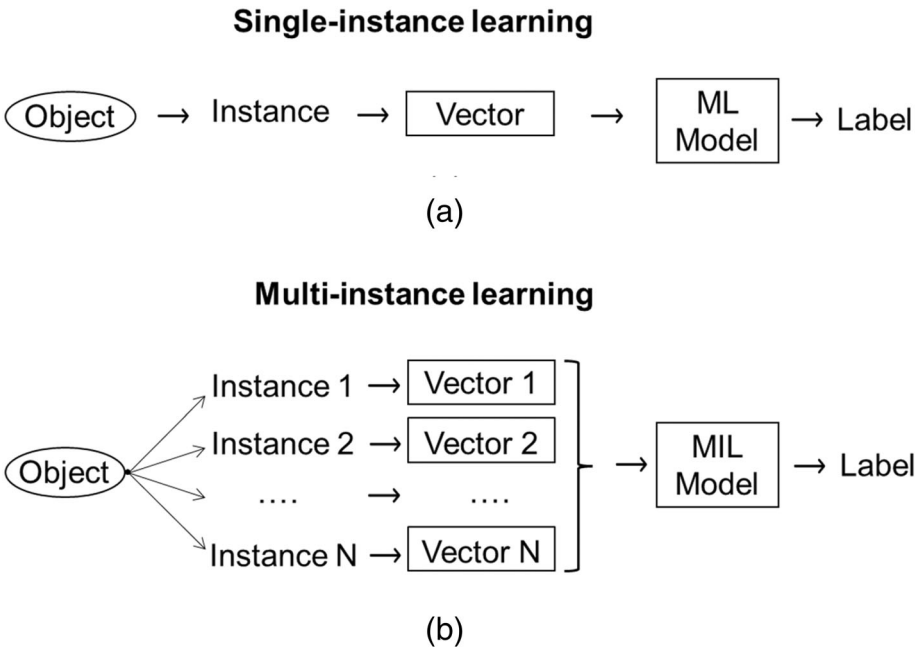

FIGURE 1 Single-instance versus multi-instance machine learning. An object can be a molecule, protein, DNA, or RNA while instances can be conformations, atoms, isoforms, subsequences, and so on. In single-instance modeling, an object is represented by a single instance chosen by a researcher and is encoded by a feature vector. In multi-instance learning, every object is represented by a set (bag) of instances where each instance is encoded by its own feature vector.

Some of the problems studied can be attributed to multi-instance learning problem[3] but were not formalized as MIL in the original papers. These were chemical structure determination by mass spectroscopy (multiple interpretations for each peak in the mass-abundance curve),[4] adaptive alignment in drug activity prediction (multiple conformations of a molecule),[5] modeling DNA promoter sequences (multiple transcriptional start sites in DNA promoter),[6] phoneme recognition (multiple segments of spoken letters),[7] and recognition of handwritten characters (multiple pose or location of the characters).[8] In 1997, Dietterich et al.[2] formalized the MIL problem. Their article was motivated by the drug activity prediction problem, which is related to MIL by the fact that a molecule can be represented by multiple alternative conformations, and it is not known which one is responsible for the observed biological effect. In the same paper,[2] they proposed an algorithm for the direct solution of multi-instance problems (see details in Section 5.1).

Since the seminal paper of Dietterich et al,[2] numerous MIL algorithms have been developed and applied in various domains, such as drug discovery (pharmacy), classification of text documents (information retrieval), classification of images (computer vision), speaker identification (signal processing), bankruptcy prediction (economy), and so on.[9–11] However, MIL still has not become a popular approach in chemoinformatics and only a few studies on its application to structure–property modeling have been reported so far.[2,5,12–21] In bioinformatics, MIL has attracted significantly more attention, because of a large number of tasks[22–35] perfectly fitting the MIL framework.

The main goal of MIL algorithms is to establish correlations between bags of instances and bag labels. Another important question concerns the identification of key instances that determine or have the greatest contribution to the label of the bag. The key instance detection (KID) problem was formulated in Ref. [36] and it is even more challenging than the prediction of the bag label since not all MIL algorithms can solve it. Also, many MIL algorithms ignore the relationship between instances in a bag because they consider the instances as independent and identically distributed (i.i.d) samples.[37] In this context, instances are i.i.d. if they have the same probability distribution and are mutually independent. In the majority of chemoinformatics and bioinformatics tasks, instances can be considered as mutually independent, for example, prediction of biological activity based on ensembles of conformations. However, in some cases, this should not be neglected, for example, prediction of a property of a molecule using atoms as instances. In this case, atoms may mutually influence each other which affects the modeling property. These aspects of MIL distinguish it from traditional SIL and require specific solutions, which were addressed in many studies.[38]

Despite the attractiveness of the MIL approach for some tasks, no comprehensive review of its application to the modeling of molecular properties/functions has been published so far. Here, we describe the MIL framework and algorithms, as well as some applications in chemoinformatics and bioinformatics.

2 | ORIGINS OF MULTI-INSTANCE LEARNING

Multi-instance learning is a suitable learning framework for tasks where the modeled object is difficult to represent with a single instance and a feature vector. The sort of problems, where an object is associated with multiple alternative representations, can be attributed to *polymorphism ambiguity* (Figure 2a). In "structure–property" modeling, this type of ambiguity arises when a molecule can be represented by alternative instances, such as conformations, tautomers,

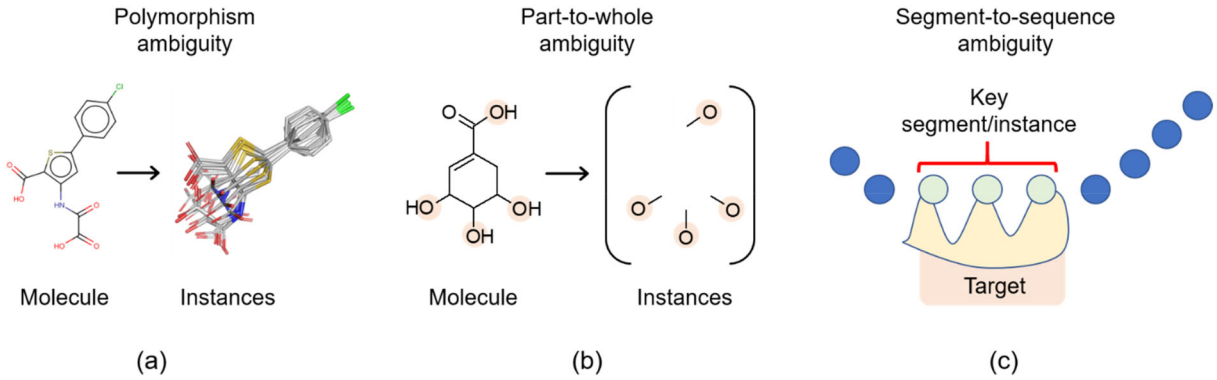

FIGURE 2 Types of ambiguity in tasks related to modeling molecular properties and functions: (a) polymorphism ambiguity, (b) part-to-whole ambiguity, and (c) segment-to-sequence ambiguity.

protonation states, and so on. The wrong choice of the key molecular form in SIL can result in the poor performance of predictive models. Thus, Masand et al.[39] demonstrated that the selection of the tautomeric form of a compound significantly influences the descriptor selection process, as well as the performance of QSAR models. According to a study by Toropova et al.,[40] accounting for both the keto- and enol-forms improves the prediction accuracy of the anxiolytic activity, compared with the models accounting for only one of the two tautomeric forms.

Another problem where MIL is applicable is characterized by a *part-to-whole ambiguity* (Figure 2b) where only one or several parts of a modeled object are responsible for its observed property. In this context, a molecule can be thought of as a set of atoms (instances), where its physicochemical or biological properties are generally influenced by a single atom or a group of atoms, and it is often not known which atoms determine the observed property of a molecule.[12] Such problems can be solved using "local" descriptors[41] describing atoms responsible for a particular property, for example, halogen or hydrogen basicity. In this case, H-bonded and halogen-bonded atoms should be explicitly labeled.

MIL is also a quite popular modeling approach in bioinformatics, where modeled objects—biological molecules (RNA, DNA, and proteins) are sequences of monomeric units (nucleotides or amino acids), and often only a particular segment of a sequence is responsible for the biological function of a whole molecule, but the exact location of such segments may be unknown. In the MIL framework, biological molecules can be represented by multiple alternative segments (instances) encoded by a special feature vector. This type of problem can be attributed to *segment-to-sequence ambiguity* (Figure 2c).

Other multi-instance problems include multi-multi-instance learning,[42] multi-instance multi-label learning,[43] key instance detection in multi-instance learning,[36] multi-instance clustering,[44] and multi-instance ranking.[20] Mutli-multi-instance learning encodes objects as nested bags, for example in text categorization a text can be represented as a bag of sentences, and each sentence as a bag of words.[42] Relatively to chemoinformatics, we may suppose to represent a molecule as a set of tautomers where every tautomer is represented by a set of conformations. Thus, a model will decide which tautomers and conformations are relevant for a modeling property. Other mentioned methods are extensions or adaptations of conventional ones applicable to single-instance learning. Comprehensive reviews of the MIL concept and its applications in regular ML tasks can be found in reviews.[9–11,38,45–49]

3 | MULTI-INSTANCE LEARNING ALGORITHMS

The growing number of MIL algorithms requires their systematization. This review follows a categorization of algorithms similar to one proposed by Amores[45] and Herrera[11] (other types of categorization of MI algorithms are described in Refs. 11,46,50,51) and distinguishes two major groups of MIL algorithms—instance-based (instance-level) and bag-based (bag-level). We chose this categorization because it is widely accepted in the MIL community and corresponds to chemoinformatics and bioinformatics applications.

The instance-based algorithms consider each instance as a separate training object generate predictions for each instance in the bag, and then apply a predefined rule (an aggregation function) to aggregate the instance predictions into a prediction for the whole bag. In contrast, the bag-based algorithms consider the whole bag as a training object and provide a prediction for the bag without explicit predictions for individual instances. The bag-based algorithms can be based (i) on the definition of a distance between bags,[52] bag similarities and kernels,[53] bag dissimilarities,[54] or (ii) on the aggregation of instances to obtain a bag representation using pooling operations (mean, weighted mean, sum, etc.).

Here, we consider three types of MIL algorithms: wrappers, conventional, and neural-network algorithms. Wrappers transform a multi-instance problem into a single-instance one. Conventional algorithms correspond to either multi-instance adaptations of classical single-instance algorithms (tree-based methods, SVM-based methods, nearest neighbors, etc.) or original algorithms specially designed for solving MIL problems. Also, there exist many specific architectures of neural networks adapted to MIL problems.

3.1 | Benchmarking datasets

In the domain of chemoinformatics, two benchmarking datasets were widely accepted and used by the MIL community to validate models. These datasets (MUSK1 and MUSK2) are related to the prediction of molecule bioactivity, particularly to the prediction of the human perception of the musk odor. They were collected and published by Dietterich et al. in their seminal papers.[2,5] The MUSK1 dataset contains 47 musk and 45 non-musk compounds. The MUSK2

dataset contains 39 musk and 63 non-musk compounds. These datasets share 72 common compounds. Compounds were collected from literature and the authors kept only those compounds that occurred in at least two publications and all musk judgments agreed. For compounds from the MUSK1 dataset only low-energy conformations were retained and the total number of kept conformations was 476. For compounds from the MUSK2 dataset, the authors kept all generated conformations to better outline the complexity of the task. The total number of conformations was 6598. MUSK1 and MUSK2 datasets are accessible by these links: https://archive.ics.uci.edu/dataset/74/musk+version+1, https://archive.ics.uci.edu/dataset/75/musk+version+2. These datasets were widely used to validate approaches which will be discussed below and therefore we described them in detail.

The MUSK datasets are of limited size, making them unsuitable for benchmarking contemporary neural network models, which typically necessitate substantial training data. Within the fields of chemoinformatics and bioinformatics, there are no other widely accepted benchmark datasets. Recent efforts under the Therapeutics Data Commons (TDC) initiative[55] have yielded datasets attributed to multi-instance problems, accessible at https://tdcommons.ai/multi_pred_tasks/overview. These datasets contained information about pairs of molecules causing particular responses or associated with a particular attribute. Examples include drug–drug interactions causing side effects or synergistic effects (DDI and DrugSyn datasets), drug-protein pairs with associated affinity values (DTI dataset), protein–protein interactions (PPI dataset and others), and more. It is worth noting that the authors of these datasets have not provided information regarding what should be considered instances in each case. In certain scenarios, components of a pair may be interpreted as instances. For instance, when a specific side effect results from the combination of two drugs in DDI dataset, these drugs can be considered as individual instances, as the role and the contribution of each drug to the effect remain unknown. In the PPI dataset, individual proteins can be represented by sets (bags) of subsequences (instances). However, the determination of what constitutes an instance is challenging in some cases. For instance, the TDC Catalyst dataset supplies information about a mixture of reactants and a mixture of products, with the task objective being the prediction of the suitable catalyst mixture. In this case, mixtures of reactants and products can hardly be considered as individual instances due to their specific roles in a chemical reaction. Furthermore, it is imperative to acknowledge the overall quality of these datasets. Notably, the TDC Catalyst dataset contains numerous solvents falsely labeled as catalysts. Other issues related to data curation and normalization have also been observed. Consequently, users are advised to exercise caution when utilizing these datasets.

Many other datasets were widely used for validation of MIL models but all of them are from other domains (image analysis, text classification, etc.) and we do not include their descriptions here, but an interested reader may find references for them in these publications.[9,45]

3.2 | Multi-instance wrappers

Multi-instance wrappers transform multi-instance data into a single-instance representation used as input to the machine learning algorithm. These types of algorithms are universal and can be coupled with any ML approach. There are two types of *wrapper* algorithms: instance-based and bag-based wrapper algorithms (Figure 3).

In *Instance-wrapper* (Figure 3a) a label of a bag is assigned to all its instances and a conventional single-instance model is built to predict labels of individual instances. The predicted label for a new bag results from averaging the instances predictions or application of other aggregation functions, e.g. max function.

In the *Bag-wrapper* (Figure 3b) algorithm, a single vector representing the bag is generated by aggregation of instances, usually by regular or weighted averaging. Then, any ML algorithm can be applied to train the model on aggregated representations of bags. In the prediction mode, all instances of a new bag are aggregated into a single representation, which is used as input to obtain a prediction for the given bag.

3.3 | Conventional MIL algorithms

Several regular machine learning algorithms were adapted to process raw multi-instance data: maximum likelihood-based methods,[56–59] decision rules and tree-based methods,[60–63] SVM-based methods,[48] and evolutionary-based methods.[64]

For example, Citation-kNN[52] is an extended version of the kNN algorithm for the bag space. In Citation-kNN, the classification of a new bag is based on the nearest bag from the training set, whereas the Hausdorff distance is used as

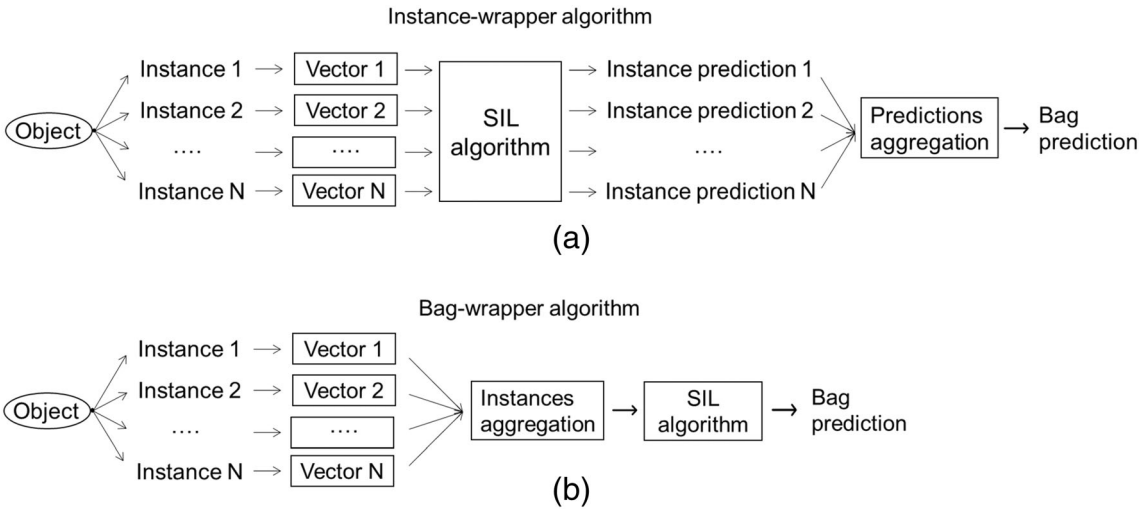

FIGURE 3 (a,b) Modeling workflow implying instance- and bag-wrapper MIL algorithms. An object can be a molecule, protein, DNA, or RNA while instances can be conformations, atoms, isoforms, subsequences, and so on. Every object is represented by a set (bag) of instances where each instance is encoded by its feature vector. SIL algorithm denotes any conventional single-instance learning algorithm (e.g., Random Forest or Support Vector Machine). Prediction and instance aggregation can be performed by the *mean* function, however, other pooling functions may be also applicable.

the distance function between the bags. The Hausdorff distance is the largest distance out of the set of all distances between each instance of one bag to the closest instance of the other bag. In other words, two bags are close if every instance of either bag is close to some instance of the other bag. Another example is the bag-level kernels, which can be used in a standard SVM to optimize the margin between bag classes. For example in Ref. 53, each bag is transformed to a minimax vector based on the minimum and maximum feature values of instances in each bag. Then any instance-level kernel and standard SVM can be applied to find the optimal margin between classes. ID3-MI and RipperMI[60] are the MIL extensions of the decision tree, and decision rules approach. They use all instances as individual and independent training samples and a bag is predicted as positive if at least one instance was predicted positive, otherwise a bag is predicted as negative.

Other algorithms were specially designed to solve MIL problems, for example, Diverse Density (DD)[56] algorithms. This is a maximum likelihood-based approach that implements an assumption that positive instances occupy a specific area in the feature space. For example, one has a 2D feature space, where individual instances are depicted as shapes (Figure 4). The algorithm searches for points of high diverse density (e.g., point A) where many different positive bags are close to those points and instances of negative bags are far away. This point is a prototype instance that is a generalization of instances of positive bags. If any instance of a given bag is closer to the prototype instance than a threshold, the bag is classified as positive. Expectation–Maximization Diverse Density (EM-DD)[57] uses the EM algorithm to locate the prototype instances more efficiently. There exist several other MI algorithms based on the Diverse Density approach, such as DD-SVM[65] and MILES.[66]

3.4 | Neural network-based MIL algorithms

Neural networks perform multi-instance learning in an end-to-end manner in which a bag with a various number of instances serves as an input. The modern approaches often generate bag embedding, that is, latent vector representing the bag based on the vector representation of instances, but in early approaches, other adaptations of neural networks were applied. Multi-instance neural networks (MI-NN) were first described by Ramon et al.[67] for classification problems. To calculate the bag label probability, they suggested aggregating computed instance probabilities by the log-sum-exp operator. Zhou et al.[68] modified MI-NN by employing a loss function capturing the nature of multi-instance learning, that is, weights of the network are updated for each training bag rather than for each training instance. Later, this neural network architecture was improved by adopting feature scaling with Diverse Density and feature reduction

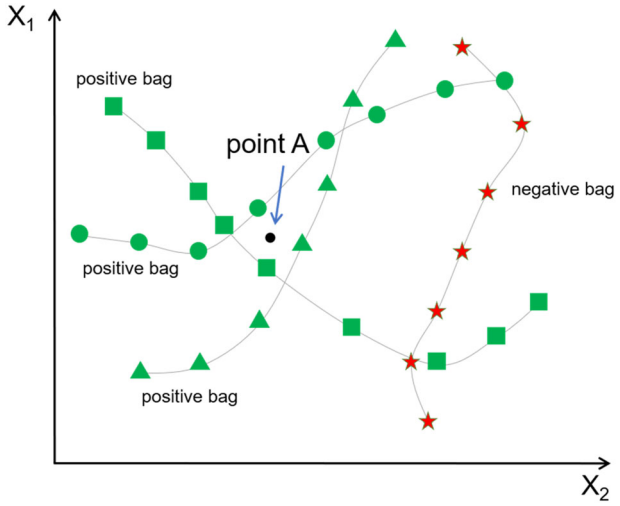

FIGURE 4 Illustration of the Diversity Density approach in a two-dimensional feature space. Point A is a point of high diversity density where instances of many positive bags (green) are close and instances of negative bags (red) are far. Gray lines connecting instances of every object were just used to better highlight instances belonging to the same object.

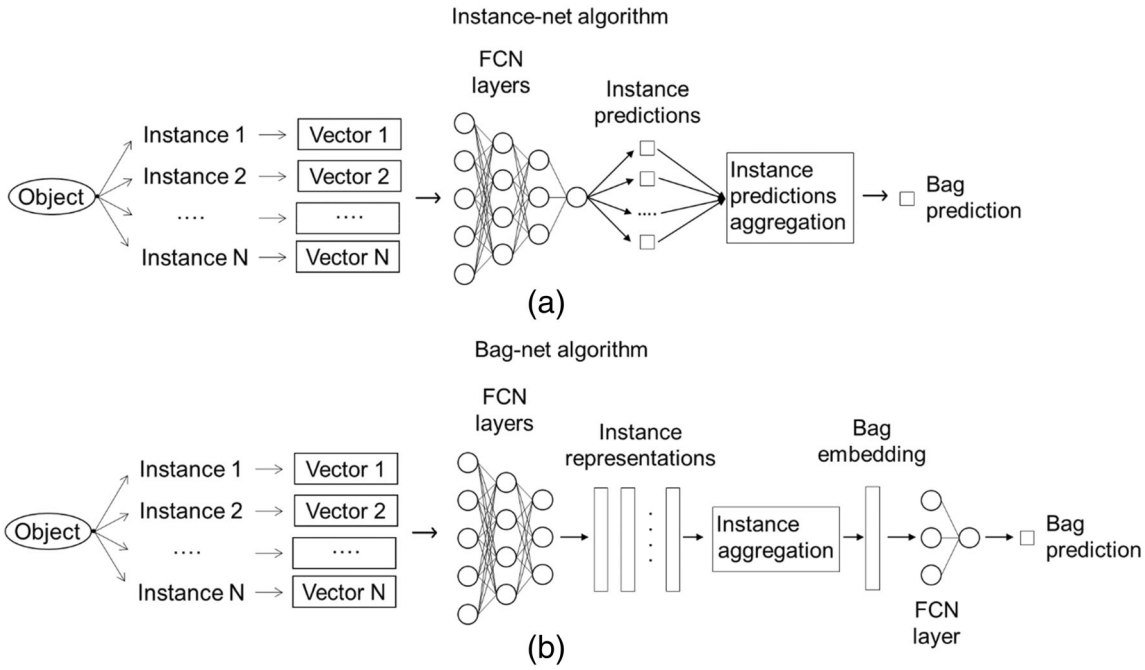

FIGURE 5 (a,b) The architecture of instance- and bag-based multi-instance neural networks. An object can be a molecule, protein, DNA, or RNA while instances can be conformations, atoms, isoforms, subsequences, and so on. Every object is represented by a set (bag) of instances where each instance is encoded by its feature vector. Commonly used aggregation functions are *max*, *sum*, *log-sum-exp*, and *mean*.

by principal component analysis.[69] Ensemble neural networks[70] and RBF neural networks[71] were suggested to tackle MIL problems. Zhang et al.[72] reported an implementation in MI-NN as a loss function for the MIL regression task.

In contrast to the abovementioned MI-NN, Wang et al.[73] focused on generating bag representations from instances. The bag-level network (Bag-net, Figure 5b) generates instance latent vectors that are further aggregated using *max*, *sum*, or *log-sum-exp* pooling operators into bag representation which in turn is used to make the bag label prediction by the last fully-connected layer with one output neuron. The instance-level (Instance-net, Figure 5a) network generates instance labels, that are then aggregated by a pooling operator. These two types of neural networks demonstrated similar classification accuracy on benchmark datasets.[73] Integration in MI-NN of some popular deep learning tricks (deep

supervision and residual connections) improved the classification accuracy. According to Wang et al.,[73] bag-level networks (Figure 5b) outperform instance-level networks (Figure 5a) on popular MIL benchmark datasets belonging to localized content-based image retrieval and text categorization tasks. However, for bioactivity prediction benchmarks MUSK1 and MUSK2 they performed comparably well: classification accuracy was 0.889 and 0.858 for instance-level networks and 0.887 and 0.859 for bag-level networks for MUSK1 and MUSK2 datasets, respectively.

Traditional pooling operators (*max*, *sum*, or *log-sum-exp*) have a clear limitation, that is, they are pre-defined and non-learnable. The *max*-pooling operator could be effective in aggregating instance scores but might be inappropriate for the aggregation of instance feature vectors in bag-level algorithms. Similarly, the *mean* pooling operator might be unsuitable to aggregate instance scores but could succeed in generating the aggregated bag representation.

Ilse et al.[74] proposed an attention-based pooling operator, that replaces pre-defined pooling operators with a trainable attention network that can generate instance weights, which quantify the importance of each instance and its contribution to the aggregated bag representation (Figure 6). A dynamic pooling[75] was inspired by the *Routing Algorithm* from *Capsule Networks*[76] and iteratively updates instance contribution to aggregated bag representation. In Ref. 77 a new pooling operator based on the LSTM recurrent neural network was proposed. The LSTM memory mechanism allows to accumulation of information after processing each instance representation to iteratively update the bag representation. This approach achieved a mean error rate of 2.04–7.4 in such classification tasks as multiple digit occurrence, single digit counting, and outlier detection in the MNIST data set[78] and outperformed attention-based[74] (mean error rate = 11.9–37.4) and dynamic pooling[75] methods (mean error rate = 25.4–40.9). However, it was not applied to chemoinformatics or bioinformatics tasks.

Set Transformer[79,80] based on a *multi-head self-attention* mechanism was also proposed as a tool for multi-instance learning. This method outperformed the mean pooling and attention-based architectures in some artificial and real-world datasets.[79,80] It was also demonstrated that Graph Convolutional Neural Networks (GCNNs) can also be used as permutation-invariant operators that improve instance representations by exploring relationships between them.[81] Klambauer et al.[82] demonstrated that Hopfield networks can solve the multi-instance problem of immune repertoire classification, in which bags are extremely large and may consist of hundreds of thousands of instances—immune receptors, represented by amino acid sequences. They demonstrated that the update rule of Hopfield networks is essentially a key-value attention mechanism—a basis of *Transformer* architectures[80] and suggested a new transformer-like attention-based pooling that allows for processing extremely large bags and extracting key instances. To avoid overfitting, they proposed also a special instance-level dropout regularization technique.

There are also other interesting examples of multi-instance neural networks. Tu et al.[83] proposed a multi-instance learning approach based on graph neural networks. In this approach, each bag of instances is converted to an undirected graph which is processed by Graph Neural Network (GNN) to learn the aggregated bag representation. In Ref.

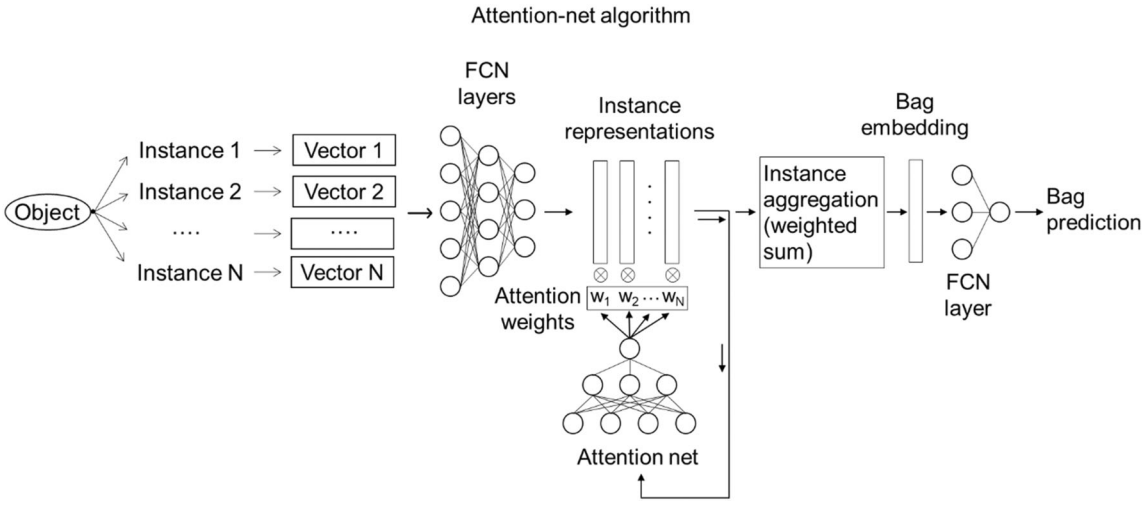

FIGURE 6 Attention-net multi-instance learning algorithm. An object can be a molecule, protein, DNA, or RNA while instances can be conformations, atoms, isoforms, subsequences, and so on. Every object is represented by a set (bag) of instances where each instance is encoded by its feature vector. An attention net was incorporated into the model architecture to predict the weights of individual instances and aggregate instance embeddings by a weighted sum.

84 each bag is converted into an unordered sequence of instances, which is processed by the recurrent neural network, that can memorize the instances.

3.5 | Key instance detection algorithms

The main goal of MIL algorithms is to predict labels for bags. However, it is also desirable to identify key instances that primarily contribute to the label of the bag (Figure 7). The solutions to the key instance detection (KID) problem[36] can efficiently be used to interpret MIL models. Following,[85] interpretation approaches of MIL models can be divided into *model-specific* and *model-agnostic*.

Model-specific KID approaches include MIL algorithms that infer instance labels or estimate the importance of instances (instance weights). Instance-based algorithms rely on some process, which determines the labels of instances in a bag. Provided by such algorithms[56,57,66,86–89] instance labels are aggregated to derive bag labels. In these algorithms, the key instances can be naturally identified by considering assigned instance labels.

Bag-based algorithms[36,90,91] can be coupled with a specially designed mechanism for the identification of key instances. For example, multi-instance neural networks can include a pooling operator, which aggregates instance representations and can also serve as a detector of key instances. Ilse et al.[74] proposed a pooling operator based on the attention mechanism,[92] which was implemented as a two-layered neural network followed by the *softmax* function that receives instance scores and generates instance weights that sum to 1. Li et al.[93] proposed a deep multiple instance selection framework (DMIS) based on hard attention[94] with Gumbel softmax or Gumbel top-k functions. In contrast to soft attention where continuous attention weights are assigned to the instances (including negative ones), the proposed approach selects several key instances filtering out potential negative (non-key) instances. It was shown that focusing on a small number of key instances may improve overall prediction accuracy. For example, the DMIS approach reached a classification accuracy of 0.907 on the MUSK2 dataset while state-of-the-art approaches had 0.836–0.903.[93] Shin et al.[95] applied a neural network inversion mechanism[96] in the MIL classification problem and demonstrated that it can significantly improve the KID performance. In the image classification tasks (MNIST, colon cancer, and breast cancer) the approach achieved F1 scores of 0.65, 0.75, and 0.23, while conventional attention reached 0.29, 0.33 and 0.15, respectively.

There are also other types of pooling operators that can be used for KID. For example, for this purpose, Gaussian pooling[97] applies the Gaussian radial basis function. Dynamic pooling[75] iteratively updates the contribution of each instance during each feed-forward step in neural network training and highlights the key instances. Tu et al.[83] implemented an approach, where each instance in a bag is a node in a graph processed by a graph neural network (GNN) converted to a fixed-dimensional representation by differentiable graph clustering pooling. This approach can capture relationships between instances in a bag, which in some cases can improve KID performance.[83]

However, the robustness of KID mechanisms in model-specific approaches is still an open question, because validation of KID solutions requires labeled data at the instance level but the amount of such data is still scarce. Haab[98] addressed this issue and concluded that models with high prediction accuracy of a target property can poorly identify a key instance, but also demonstrated that the robustness of KID models can be increased by employing an ensemble of models rather than a single model.

The *model-agnostic* KID approach for the interpretation of MIL models in classification tasks was proposed in.[85] This approach considers methods ignoring relationships between instances and those that recognize such relationships. The former implies simple strategies such as single instance prediction or *one instance removed* prediction or their

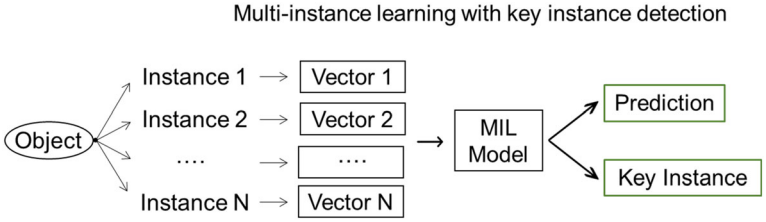

FIGURE 7 Multi-instance learning with key instance detection. A MIL model may predict not only a label for an input object (a molecule, protein, or DNA) but may also predict the most relevant instance (a conformation, an atom, an isoform, a subsequence, etc.).

combination. The latter is represented by the Multiple Instance Learning Local Interpretations (MILLI) technique, which is similar to the popular single-instance LIME[99] and KernelSHAP[100] methods for model interpretation. Interestingly, it was shown[85] that the model-agnostic approaches perform significantly better in the identification of key instances than model-specific KID mechanisms of popular MIL algorithms. On the MNIST dataset, MILLI approaches reached a normalized discounted cumulative gain (a measure of how well-supporting instances are ranked on top) of 0.942 while other approaches resulted in 0.630–0.833.

4 | RELATED CONCEPTS

Many other approaches were proposed to solve multi-instance problems with more straightforward strategies and without involving multi-instance algorithms. For example, many attempts to solve these problems in chemistry were related to the modeling of biological activity based on ensembles of conformations. Various approaches such as 4D-QSAR,[101,102] Quasar,[103] *dynamic* QSAR,[104] *ensemble* QSAR,[105] multi-conformational structure-based QSAR,[106] and many others[101] were proposed for this purpose.

The 4D-QSAR suggested by Hopfinger et al.[102] was the first attempt to account for the flexibility of molecules represented by ensembles of conformations obtained from molecular dynamics simulations. Individual conformations of a molecule were aligned within a grid and occupancy of every cell in a grid by polar, non-polar, hydrogen-bond donor or acceptor atoms is calculated and averaged over all conformations to encode a molecule. This representation was further used to build a PLS model to predict a biological response of compounds.

Bak and Polanski[107] suggested a modification of Hopfinger's approach. Occupancies calculated for individual conformations are mapped on a self-organizing Kohonen map of a fixed size. Every molecule is encoded by a sum of occupancies or mean charge values in each neuron and the PLS method enhanced by iterative variable elimination is applied to establish a correlation with a property.

Within the Quasar approach, Vedani and Dobler tried to solve the issue of alignment and suggested creating a quasi-receptor around training set molecules represented by multiple conformations. A quasi-receptor is generated by placing different features (hydrophobic, H-bond donor/acceptor, etc.) around training set conformations and the positions of these features were iteratively optimized. Afterwards, the binding energy of individual conformations to a quasi-receptor was calculated followed by a calculation of the binding energy of a molecule as a Boltzmann average. Further, these values were used to establish a linear model with observed binding energies of corresponding compounds.

Mekenyan[104] proposed a *dynamic* QSAR framework where a rule-based system is used to screen conformation ensembles for selecting an effective set of conformations according to the desired property. Predicted property values for individual conformations were averaged or a Boltzmann average can be computed to get a predicted value for a molecule. The authors also demonstrated that averaging of descriptors did not allow to improve model performance.

The 4D-SiRMS approach was proposed by Kuz'min et al. as alignment-independent 4D-QSAR method.[108,109] Individual conformations were encoded by 3D simplexes which are the number of identical tetraatomic fragments with fixed composition, topology, and chirality. Descriptors were averaged across individual conformations of compounds using the Boltzmann distribution and models were built using the PLS method.

All described approaches follow essentially the same strategy as the earliest MIL solutions: converting a multi-instance learning problem into a single-instance one. The majority of approaches can be attributed to bag-wrappers where descriptors were aggregated to represent a compound. Technically this was implemented in different ways, for instance, averaging or concatenation of conformation descriptors.[104] A more physically sound method is a Boltzmann averaging of conformation ensembles,[108,109] but its efficiency is limited by the accuracy of the estimation of conformation energies. To account for ambiguity in the 2D structure representation of molecules, Bonachera et al.[110] averaged descriptor vectors of microspecies (protonation or tautomer forms) taking into account their predicted abundance. This approach is similar to the bag-wrapper algorithm which tackles polymorphism ambiguity caused by protonation equilibria.

Another example of related approaches is graph convolution neural networks (GCNNs) considering a molecule as a molecular graph. Individual atoms are featurized and then atomic embeddings are enriched considering the embeddings of the neighboring atoms by application of permutationally invariant operators. For instance, the simplest Kippf graph convolution[111] updates atomic embeddings using the repeated summation of embeddings of the given atom with neighboring atoms embeddings. Afterwards, vector representations of individual atoms are pooled into a single feature vector by choosing a maximum value for each variable (max pooling). This vector represents a whole molecule and is used to train a model and make a prediction. This is not widely recognized but, in fact, GCNNs perfectly fit the MIL framework where a molecule is a bag and instances are atoms. GCNN models create embeddings of atoms (instances)

which afterward are aggregated into a single embedding used to predict a property of a molecule. This corresponds to the bag-level MIL approach which does effectively the same. We refer readers to some comprehensive reviews of the graph convolutions published recently in Refs. 112–114.

5 | MULTI-INSTANCE LEARNING APPLICATIONS

5.1 | Polymorphism ambiguity modeling

5.1.1 | Bioactivity modeling with conformation ensembles

One of the first ideas of modeling biological activity with multiple conformations and a MIL-like algorithm was implemented in *Compass*,[5] an algorithm that automatically selects bioactive conformations and their alignments. *Compass* is based on a neural network that increases the accuracy of biological activity prediction by the iterative selection of more probable (bioactive-like) conformation. *A compass* was applied to predict the human perception of the musk odor of 102 molecules from the MUSK2 dataset. The single conformation model demonstrated a performance of 71% while the model considering multiple conformations achieved 91% prediction accuracy in the cross-validation experiment.

Then, Dietterich et al.[2] formalized the problem of multi-instance learning and proposed an axis-parallel hyper-rectangles (APR) classification algorithm. All conformations of all molecules are aligned, 162 evenly distributed rays are emanated from the origin and the distance from the origin to the surface along each ray is a descriptor value. Thus 162 shape descriptors are calculated for every conformation. Then the algorithm searches for lower and upper boundaries along each ray which define the virtual wall of a receptor. Optimal hyper-rectangles should match at least one conformation of positive molecules and avoid all conformations of negative ones. The APR algorithm was tested on MUSK1 and MUSK2 datasets and compared with conventional neural network and C4.5 decision tree methods. The latter treated all instances of positive examples as positively labeled during training and predicted a molecule as positive if at least one instance was predicted positively. The APR approach was superior in both cases. It achieved a classification accuracy of 92.4% and 89.2% on MUSK1 and MUSK2, respectively, while neural networks (NN) and decision trees (DT) which ignored the multi-instance nature of the problem achieved 75.0% and 67.7% (NN) and 68.5% and 58.8% (DT).

Other examples of applications of MIL for bioactivity modeling are scarce. In Ref. 14 a molecule was represented by a set of conformations encoded by binary pharmacophore features used as an input to multi-instance regression. The bioactivity of a given molecule was assessed by weighted averaging of the predicted activities of its conformations. The experiments on three datasets (23 dopamine agonists, 31 thermolysin inhibitors, and 41 thrombin inhibitors) demonstrated that the multiple conformation models always outperform single conformation models (dopamine agonists: RMSE = 0.87 vs. 1.25, thermolysin inhibitors: RMSE = 1.27 vs. 1.37, and thrombin inhibitors: RMSE = 1.28 vs. 1.36).

The popular MILES (multiple-instance learning via embedded instance selection) algorithm was successfully applied to the classification of bioactive compounds against GSK-3 (266 active and 258 negative molecules), P-gp (122 active and 128 inactive molecules), and cannabinoid receptors (data set I had 253 active and 284 inactive molecules; data set II had 307 active and 188 inactive molecules).[15] Each molecule was represented by a set of conformations encoded by pharmacophore fingerprints. For comparison purposes, the authors used conventional modeling approaches: 1-norm SVM, decision tree, and Random Forest. To represent molecules in these cases they aggregated bit vectors of individual conformations for a molecule using logical OR. Thus, reference models can also be recognized as MIL models but with bag-level aggregation rather than instance-level aggregation of the MILES approach. For all four datasets, MILES showed high performance with a classification accuracy of 0.898–0.978 whereas the reference approaches resulted in 0.698–0.919 accuracy. Also, it was demonstrated that MILES can be used to recognize bioactive conformations. The MILES model was able to recognize experimental bioactive conformations for 10 out of 12 test molecules from the GSK-3 data set, which was intentionally included in the bag of the generated conformations.

In our recent study, we proposed a MIL-kmeans algorithm to build 3D multi-conformational models.[16] A molecule (bag) was represented by an ensemble of its conformations (instances), each encoded by 3D pharmacophore descriptors. The latter were used to cluster all instances in the training set. Then, for each molecule, a new binary descriptor vector was generated. Its length was equal to the number of clusters. A bit was assigned 1 if, at least, one conformation of a compound fell into the corresponding cluster or 0 otherwise. In such a way, this approach transforms multi-instance data into single-instance representation. Then Random Forest algorithm was applied to build classification models based on these bit vectors. This approach demonstrated some competitive performance on datasets composed from exclusively chiral molecules: balanced accuracy for 2D/MIL models for serotonin 1a receptor—0.79/0.77, dopamine D2

receptor—0.83/0.81, alpha-1b adrenergic receptor—0.69/0.74, mu opioid receptor—0.83/0.81, JAK2 kinase—0.85/0.86, 11-beta-hydroxysteroid dehydrogenase 1—0.83/0.82. However, it was outperformed by conventional 2D models in a larger scale testing on 163 datasets extracted from ChEMBL which contained preferably achiral molecules: 2D models were better in 124 cases, whereas in other cases both models demonstrated accuracy close to random.

Afterward, we extended our studies[17,18] and performed a large-scale benchmark of single-instance and multi-instance regression models for the prediction of the biological activity of molecules on an updated 175 datasets from the ChEMBL database. 3D multi-conformation models outperformed 3D single-conformation models in 98% of cases (average $R^2 = 0.524$ vs. 0.024) and conventional 2D models in 70% of cases[17] (average $R^2 = 0.524$ vs. 0.464). Also, we observed that the performance of 3D multi-conformation models depends on the type of multi-instance algorithm. In particular, we found that *Instance-wrapper* outperformed more sophisticated multi-instance attention-based neural networks in 84% of datasets (average $R^2 = 0.524$ vs. 0.468). On the other hand, the above study demonstrated that attention-based neural networks can successfully identify bioactive-like conformations (i.e., solve key instance detection problems) even better than the popular AutoDock Vina docking program.[115] This study justified applicability of MIL approaches to solve chemical problems and their competitiveness to other methods. We tried to establish factors which make datasets more suitable for conventional 2D modeling or multi-conformation modeling using MIL approaches and found that 3D multi-conformation MIL models perform better on datasets composed from more rigid molecules. This may be explained by conformation sampling issues and the insufficient number of representative conformations.

5.1.2 | Catalysts enantioselectivity modeling with conformation ensembles

Modeling of enantioselectivity of chiral organic catalysts has recently attracted a lot of attention and at least two 3D multiple conformation approaches[19,116,117] have been proposed to generate Quantitative Structure–Selectivity Relationships models. For example, Denmark's group reported a 3D grid-based approach,[116,117] in which each catalyst conformation was represented by Average Steric Occupancy descriptors (averaged steric occupancy vectors of conformations similar to the Bag-wrapper approach). They demonstrated that multiple conformation models outperform single conformation ones (MAE of 0.21 vs. 0.26 kcal/mol) in the prediction of enantioselectivity for the test set of phosphoric acid catalysts.

We reported the first application of MIL to 3D modeling of enantioselectivity[19] for the phosphoric acid catalysts from Zahrt et al.[116] Each catalyst was represented by an ensemble of conformations and 3D models were built using *Instance-wrapper*, *Bag-wrapper*, *Instance-net*, *Bag-net*, and *Attention-net algorithms* (Section 3). Similar to our previous study on the modeling of biological activities,[17] the *Instance-wrapper* algorithm outperformed other multi-instance algorithms and single conformation models in predicting enantioselectivity. The MIL approach resulted in a mean absolute error of 0.25–0.26 kcal/mol for predicted $\Delta\Delta G$ while 2D single-instance gave 0.39–0.40 kcal/mol on external test sets.[19]

In the recent study[13] we performed more rigorous validation of MIL approaches on four datasets: (i) asymmetric addition of thiols to imines catalyzed by chiral phosphoric acid catalysts (PAC data set), (10) (ii, iii) asymmetric alkylation of glycine-derived Schiff bases catalyzed by ammonium salts (APTC-1 and APTC-2 datasets),[118,119] and (iv) asymmetric protonation of carboxylic acids catalyzed by chiral disulfonimides (DSI dataset).[120] It was demonstrated that 3D multi-conformer models outperform any 3D single-instance model regardless of the 3D descriptors used. *Instance-wrapper* showed the best performance in PAC (MAE = 0.33 kcal/mol), APTC-1 (MAE = 0.11 kcal/mol), and DSI datasets (MAE = 0.17 kcal/mol) while *Bag-wrapper* was better in APTC-2 (MAE = 0.13 kcal/mol). These results were comparable to or better than the performance of best 2D models obtained in the same study: PAC 0.28 kcal/mol, APTC-1 0.16 kcal/mol, APTC-2 0.22 kcal/mol, DSI 0.23 kcal/mol (all values are MAE). We also showed that MIL models better predict enantioselectivity beyond the training set on the example of the PAC dataset where reactions with enantiomeric excess (*ee*) below 80% were used as the training set and reactions with $ee \geq 80\%$ composed the test set. The 3D multi-conformer model built using the quantile loss function achieved R^2_{test} 0.36 and a Ranking accuracy of 0.79, while the 3D single conformer model had 0.01 and 0.67 and the best 2D model had −0.07 and 0.69, respectively.

5.2 | Part-to-whole ambiguity modeling

5.2.1 | Property modeling with atoms as instances

A molecule can be thought of as a collection of interconnected atoms. However, it is frequently unclear which specific atoms are responsible for the observed particular molecule's property. This problem can be treated within the *part-to-*

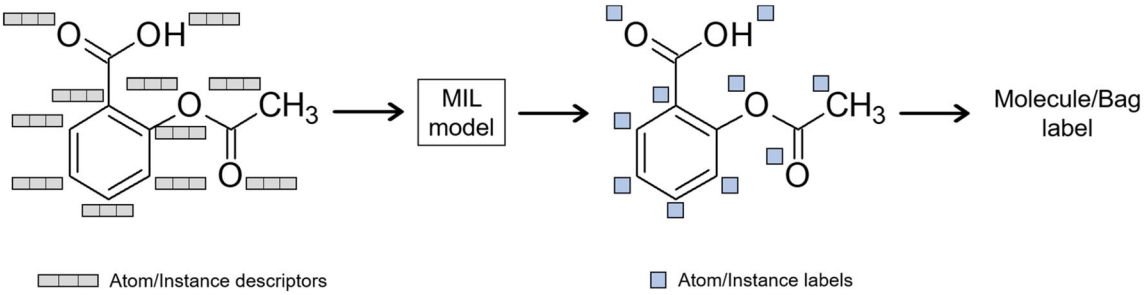

FIGURE 8 A general approach to multi-instance modeling of properties of molecules represented by atom instances. Vectors of atoms can include physico-chemical or quantum-chemical descriptors or can be extracted using graph neural networks.[12]

whole ambiguity framework where each atom of a molecule is represented by a separate vector of atom descriptors (Figure 8).

Often, the problem of identification of fragments responsible for a particular property is solved using approaches for model interpretation,[121] for example, similarity maps[122] or masking of atoms/fragments.[123] MIL can be applied in a case when the assignment of an atom responsible for a given activity, like acidity or basicity, is complicated. If such a key atom is known, the models can be built using local atomic descriptors.[41,124–126]

Bergeron et al.[20,21] introduced the multi-instance ranking (MIRank) approach to identify metabolic sites of molecules (i.e., atomic groups from which a hydrogen atom is removed during the enzymatic transformation). The authors suggested grouping topologically equivalent hydrogens into individual bags and labeling bags as positive, if any of the hydrogens within the bag are metabolized, and negative otherwise. Each hydrogen atom was represented by a set of quantum chemical descriptors such as the charge, the surface area, hydrophobic moment, and so on, calculated from 3D structures of molecules. This resulted in different descriptor representations of hydrogens within the same bag and allowed the formulation of the task as a multi-instance problem. Using a dataset of 227 drugs, drug candidates, and other biologically active compounds that were metabolized by cytochrome CYP3A4.[127] It was demonstrated that the MIRank model performed slightly better than the linear multi-instance classification model[20] (classification accuracy 70.9% vs. 67.1%). Later, Bergeron et al.[21] upgraded their algorithm and validated it on an extended database of 10 CYP datasets comprising 28–397 compounds collected from the literature. The models achieved ranking accuracy of 57.3%–75.2% for individual cytochromes and the model which was trained on the single combined dataset (923 compounds) achieved even higher accuracy 77.4%, while the random baseline expectations were 16.5%–35.6% for individual cytochromes and 19.6% for the combined dataset.

Recently, Xiong et al.[12] developed a multi-instance graph neural network to predict both the macro-pKa of the molecule and the micro-pKa of individual atoms. In their approach, a molecule was a bag, which contained instances of the ionizable atoms of this molecule. Each atom was described by a vector of features extracted with a graph neural network. The extracted instance features were used to predict the micro-pKa of atoms, which were then aggregated to derive a macro-pKa. The model (Graph-pKa) was tested on the dataset of 16,595 compounds associated with 17,489 pKa values. The Graph-pKa model was compared with baseline models (SVM, RF, XGBoost, and ANN machine learning models) where each compound was encoded by a set of molecular fingerprints. As a result, Graph-pKa achieved a MAE of pKa prediction of 0.55 on the test set, while baseline models showed prediction accuracy within MAE = 0.63–0.72.

In Table 1 we summarized all currently published chemoinformatics studies where MIL approaches were applied.

5.3 | Segment-to-sequence ambiguity modeling

5.3.1 | Protein–protein interactions

Protein–protein interactions (PPI) play an important role in biological processes. In general, only particular segments of proteins (domains) are involved in the interaction between the proteins and, therefore, determine their functional response. For this reason, knowledge of such domains enables the prediction of new PPI.

TABLE 1 Applications of multi-instance learning approaches in chemoinformatics.

Paper	Year	Task	Datasets	Representation	Algorithms
Jain et al.[5]	1994	Bioactivity of molecules (musk strength)	MUSK1 (102 molecules)	Multiple conformations of the molecule	Instance-level neural network
Dietterich et al.[2]	1997	Bioactivity of molecules (musk strength)	MUSK1 (92 molecules) and MUSK2 (102 molecules)	Multiple conformations of the molecule	Axis-parallel rectangles (APR): standard APR, outside-in APR, inside-out APR
Davis et al.[14]	2007	Binding affinity of molecules	Dopamine agonists, thermolysin inhibitors, and thrombin inhibitors	Multiple conformations of the molecule	Multi-instance regression
Bergeron et al.[42]	2008	Identification of metabolic sites of molecules	227 compounds metabolized by cytochrome CYP3A4	Equivalent hydrogens as a bag	MIRank
Bergeron et al.[127]	2012	Identification of metabolic sites of molecules	10 CYP datasets	Equivalent hydrogens as a bag	Upgraded MIRank
Fu et al.[15]	2012	Inhibitory activities of molecules	Inhibitors against GSK-3, P-gp, and CBrs receptors	Multiple conformations of the molecule	MILES
Nikonenko et al.[16]	2021	Bioactivity of molecules	162 ChEMBL datasets	Multiple conformations of the molecule	MIL-kmeans
Zankov et al.[17]	2021	Bioactivity of molecules	175 ChEMBL datasets	Multiple conformations of the molecule	Instance-Wrapper, Bag-Wrapper, Instance-Net, Bag-Net, Bag-AttentionNet
Zankov et al.[19]	2021	Enantioselectivity of organic catalysts	Phosphoric acid catalysts	Multiple conformations of catalysts	Instance-Wrapper, Bag-Wrapper, Instance-Net, Bag-Net, Bag-AttentionNet
Xiong et al.[12]	2022	Macro- and micro-pKa of molecules	16,595 compounds associated with 17,489 pKa values	Instances of the ionizable atoms as a bag	Multi-instance graph neural network
Zankov et al.[13]	2023	Enantioselectivity of organic catalysts	Phosphoric acid catalysts; two datasets on phase-transfer catalysts; disulfonimides	Multiple conformations of catalysts	Instance-Wrapper, Bag-Wrapper, Instance-Net, Bag-Net, Bag-AttentionNet

Experimental PPI data provide some information on the interacting protein pair and the type of interaction (activation, ingestion, phosphorylation, dissociation, etc.), but the important details concerning interacting domains (key domains) are often unavailable. This scenario fits the MIL framework, where each potential domain pair is an instance (Figure 9) and the whole collection of domain pairs in a given protein–protein complex constitutes a bag. At least, one of these domain pairs defines a type of interaction (e.g., phosphorylation) (Figure 9). If the proteins do not interact, there is no pair of interacting domains in the bag.

For a dataset of 1279 PPI records labeled with 10 different interaction types, Yamakawa et al.[128] considered the simplified classification task—whether a given PPI is phosphorylation or not. To solve this problem, they proposed a Voting Diverse Density (VDD) algorithm based on the Diverse Density (DD) method. The main idea of the DD approach is to find a prototype point in the feature space that is close to at least one instance from every positive bag and far away from any instances in negative bags (Figure 4). Prototype points are found according to a diverse density score, which is a measure of how many different positive bags have instances near the prototype point. The authors observed that the DD algorithm is sensitive to the contribution of negative instances to the diverse density score. To solve this problem,

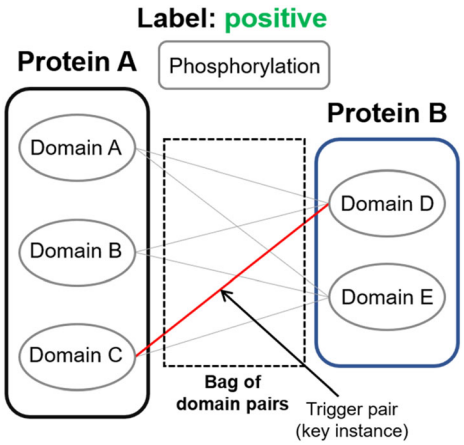

FIGURE 9 Probable domain-domain pairs of interacting proteins. A pair of proteins (an object) is represented by a set of corresponding domain pairs (instances).

they proposed a weighted voting system, in which many positive instances that are near the prototype point should receive a higher score, even if there are a few negative instances close to that point. Then they compared the VDD model with other popular six MIL algorithms (Citation-kNN, mi-SVM, EM-DD, MI-SVM, Diversity Density, Iterdiscrim-APR) from the MILL toolkit (A Multiple Instance Learning Library).[129] VDD model demonstrated classification accuracy of 0.852, while competitive MIL algorithms performed within 0.565–848 accuracy. Although some alternative MIL algorithms are very close in accuracy to VDD (the closest one Citation-kNN with a classification accuracy of 0.848), the VDD model building time was reported to be around 70 s, whereas, for the other algorithms, the time varied between 900 and 1500 s.[128]

Multi-domain proteins can realize many different functions. To predict the biological functions of proteins, Wu et al.[27] used a Multi-Instance Multi-Label (MIML) framework, where protein domains (instances) and the protein (bag) were associated with multiple biological functions (multiple labels). They demonstrated that their ensemble MIML learning approach (an ensemble multi-instance multi-label learning framework, EnMIMLNN) outperformed most of the other state-of-the-art MIML algorithms (MIMLNN,[130] MIMLSVM+,[131] En-MIMLSVM,[132] MIMLBoost,[132] MIMLkNN,[133] DBA[134]) on seven real-world genome data sets from the main biological systems: two bacteria genomes (*Geobacter sulfurreducens*, *Azotobacter vinelandii*), two archaea genomes (*Haloarcula marismortui*, *Pyrococcus furiosus*) and three eukaryote genomes (*Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Drosophila melanogaster*) (average Hamming Loss: 0.009 for EnMIMLNN and 0.009–0.064 for others).

5.3.2 | Gene–gene interactions

A gene in a DNA sequence induces the biosynthesis of a given protein with an inherent structure and biological function. Proteins produced by different genes can be functionally related and their combination may determine a particular phenotype. This is a so-called gene–gene interaction. However, an alternative splicing mechanism makes possible the synthesis of several isoforms of a given protein from the same gene. These isoforms have a similar amino acid sequence and structure but may exhibit different biological functions. This substantially complicates gene–gene interaction modeling and may lead to the erroneous classification of interaction between genes as negative if the corresponding proteins were considered in their canonical isoforms (widely expressed or the longest sequence) while alternative isoforms are responsible for the effect.

This case can be handled within the MIL framework, in which a gene (bag) generates several protein isoforms (instances). The interaction between a gene–gene pair is positive if at least one of the isoform-isoform interactions (IIIs) is positive. To address these problems Li et al.[135] proposed a single-instance bag MIL (SIB-MIL) algorithm based on the Bayesian network classifier. SIB-MIL works at the instance level and assigns to each instance (isoform pair) a probability to be positive (interactive). In SIB-MIL, the Bayesian network classifier is initially trained on positive bags with single-instance (gene pairs with a single pair of isoforms) and negative instances from negative bags. The obtained

classifier is then used to assign probability scores to the remaining isoform pairs in multi-instance bags. Using the obtained probability scores, a witness (key instance) is selected from each positive bag and labeled as positive. The instances with the highest probability score from the negative bags are labeled as negative. Updated labels are used to retrain the Bayesian network classifier. The instance labels are updated until the accuracy of the validation set stops improving. At the gene-pair level, the label of a bag is defined as the maximum probability score of its instances.

Then, Zeng et al. proposed a DMIL-III method[30] based on a deep neural network with convolutional layers. DMIL-III consists of several convolution layers and the last layer with a sigmoid activation function, which quantifies the probability that an isoform pair is interacting or not. These isoform pair probabilities are then processed by the maximum pooling operator to obtain the final predicted label for the given bag (gene–gene pair). DMIL-III neural network was tested in the PPI dataset of 26,344 positive gene bags (at least one isoform pair is interacting) and 20,910 negative gene bags (none of the isoform pair is interacting), corresponding to 177,456 positive and 130,138 negative isoform pairs, respectively. DMIL-III (classification accuracy of 0.94) was shown to significantly outperform the described SIB-MIL (classification accuracy of 0.54) and mi-SVM multi-instance algorithms (classification accuracy of 0.64).

Typically, PPIs and IIIs databases contain information about identified interactions, whereas classification algorithms require also negative examples, which are usually generated artificially. This strategy often results in a significant excess of negative examples over positive ones, leading to imbalanced datasets. For this reason, Zeng et al.[22] implemented a novel loss function to handle the imbalanced data and proposed the IDMIL-III method. They also enhanced the IDMIL-III with an attention mechanism, to identify the interacting isoform pairs from a positive gene bag. For algorithm comparison, they used the multi-isoform gene pairs dataset derived from the Human Protein Reference Database (HPRD).[136] IDMIL-III achieves an F1 value of 95.4% at the gene-level prediction, which is 42.2% higher than that of SIB-MIL, and 3.8% higher than that of DMIL-III.

5.3.3 | MHC-II-peptide interactions

Major histocompatibility complex (MHC) proteins are a large set of cell surface proteins that are essential for the adaptive immune system. MHC proteins of class I and II bind a short peptide fragment (epitope) obtained from cytosolic or extracellular proteins, correspondingly, and present it at the cell membrane to cytotoxic T cells. This will trigger a response from the immune system against a particular non-self-protein. In the context of vaccine design, it is very important to know which peptides bind to MHC to initiate the desired immune response. MHC proteins have a binding groove where peptide fragments bind. MHC-I has a closed groove and usually binds peptides of lengths between 9 and 11 amino acids. In contrast, the binding groove of an MHC-II protein is open at both ends and can bind peptides commonly with lengths from 11 to 30 amino acids.[137] On the other hand, it was established that 9-mer segment of the peptide is responsible for the MHC-II binding but the sequence itself is hard to identify experimentally.

Multi-instance learning was adapted to predict peptide binding activity to MHC-II in classification[138] and regression tasks.[139] Both approaches generated bags of segments of nine amino acids using the sliding window approach (Figure 10). In study,[138] an SVM classifier with a normalized set kernel was used as the multi-instance method.[138] This method was tested on the MHCII benchmark dataset collected by Wang[140] and describing 10,017 experimentally measured peptide MHCII binding affinities for 14 human and 2 mouse MHC class II types and was demonstrated to perform on the level of the conventional state-of-the-art approaches. In study,[139] the prediction of MHC-II binding activity was considered as a regression problem. For this purpose, the popular multi-instance MILES algorithm[66] was adopted for the regression task by replacing the 1-norm SVM classifier with a support vector regression (SVR). To be compared with other methods, the proposed MHCMIR method was also tested in classification mode, when predicted binding affinities were converted to binary labels by specified thresholds. As a result, MHCMIR outperformed other competing methods on 4 out of 16 Wang's MHCII benchmark dataset subsets.[140] Also, it was demonstrated, that MHCMIR can identify key instances (peptides binding cores).

A new MIL approach for predicting MHC-II binding was proposed in which flanking amino acids (11-mers) were considered in addition to the 9-mer segments.[26] Also, the authors used experimental information that amino acids at positions 1, 4, 6, 7, and 9 may be crucial for peptide binding and integrated this information into the learning algorithm. In addition, their study revealed that amino acids at position 2 may also influence peptide binding.

Often, experimental methods cannot identify which member of the MHC-II protein family is bound to a given peptide. Cheng et al.[23] formulated the MIL problem, where the bag contains multiple MHC-II proteins. The bag is positive

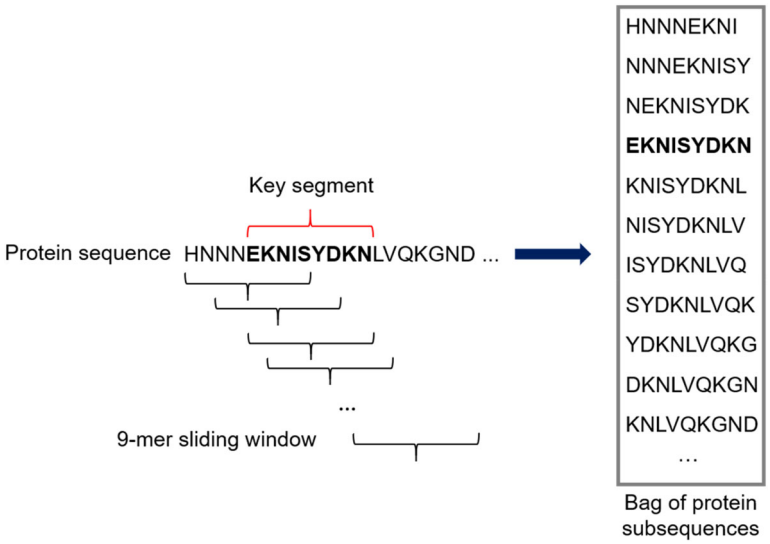

FIGURE 10 Sliding window approach for the generation of a bag of subsequences (instances) from biological molecules (e.g., proteins), which are modeling objects.

if at least one MHC-II protein binds a given peptide and negative if there are no binding MHC-II proteins in the bag. They used a combined dataset of single-allele and multi-allele data[141] to train a *Transformer-based* neural network BERTMHC.[23] Their model achieved an AUROC value of 0.72 on the independent test set in the task of classification of peptides as binders and non-binders while state-of-the-art methods performed worse: NetMHCIIpan3.2 (AUROC = 0.68), PUFFIN (AUROC = 0.69), and MHCnuggets (AUROC = 0.58).

5.3.4 | Calmodulin–protein interactions

Calmodulin (CaM) is a calcium-binding protein of 148 amino acids that can interact with more than 300 proteins and peptides,[142] thereby regulating many biological processes. The biological significance of CaM and the high diversity of proteins interacting with CaM motivated the development of computational methods for predicting both the proteins able to bind to CaM (interaction prediction problem) and the binding sites of these proteins (binding site prediction problem).

Minhas et al.[33] proposed the MI-1 SVM algorithm for CaM binding site prediction and tested it on the dataset of 153 proteins with 185 experimentally annotated binding sites. In a single-instance scenario, the subsequences annotated as binding sites were labeled as positive examples, and all other parts of the protein (obtained using a sliding window approach) as negative. However, experimental methods do not always accurately determine the position of the binding site, which introduces ambiguity into the learning process of the classification model. Therefore, within the multi-instance framework, all subsequences overlapping the binding site formed a positive bag, and all other subsequences formed a negative bag.

They tested one single-instance algorithm (vanilla SVM) two multi-instance algorithms mi-SVM and a new MI-1 SVM designed by them. In mi-SVM, first, all instances of positive bags are assigned positive labels, and all instances of negative bags are labeled as negative. Then, a standard SVM is trained on the obtained single-instance data and used to update the labels for the instances. If no instance in a positive bag is assigned a positive label, the algorithm chooses the instance in the bag having the largest score (discriminant function value in SVM) and sets its label as positive. This algorithm is based on the condition that in positive beg there must be at least one positive instance. They formulated the relaxed condition, in which at least one instance in the true binding site needs to score higher than the negative instances from the same protein (bag). Finally instance labels are updated until is repeated until the instance labels stop changing. As a result, the authors concluded that MI-1 SVM outperformed mi-SVM (AUC = 96.9 vs. 96.2) and single-instance SVM (AUC = 96.9 vs. 95.9) in the classification of binding sites. However, the authors also concluded[33] that the accuracy of MI-1 SVM for CaM interaction prediction is still low since the presence of a CaM-binding pattern in a

protein does not guarantee efficient binding of the pro-protein to CaM. Therefore, they considered the prediction of CaM binding proteins and their binding sites as separate tasks and proposed the CaMELS approach, which achieved state-of-the-art accuracy in both tasks.[139]

5.3.5 | Modeling genomic sequences

Transcription of genes is the process of copying a DNA sequence into an RNA molecule. A Transcription Factor (TF) is a special protein that binds to a DNA sequence and activates or suppresses the expression of certain genes. Regions of DNA sequences that are bound by a transcription factor are called Transcription Factor Binding Sites (TFBS). Modern experimental techniques[29] enable the identification of DNA segments that are bound by the TF protein, but the precise identification of TFBS is still a challenge. A DNA sequence may contain one or more binding sites and, usually, the exact location of the TFBS is not known, although preference information is sometimes available. Therefore, it is natural to represent the DNA sequence as a bag of possible binding sites. A bag is generated by a sliding window approach (Figure 10) of length n through the whole DNA sequence. In the MIL classification setting, a bag (DNA sequence) is positive if it contains at least one TFBS and negative if it contains no TFBS. The typical length of a TFBS is 6–12 bp, which is reflected in the length of the subsequences (instances) included in the bag.

The in vitro protein binding microarray (PBM) experiments allow high-throughput screening of DNA sequences that bind to a given TF. The typical length of DNA sequences in such experiments is 35 bp, whereas TFBS lengths normally vary from 6 to 12 bp. PBM data provide an excellent source for the modeling of TF-DNA interactions and predicting in vivo binding. To model in vitro binding, Gao and Ruan[24] used a dataset of measured binding affinities of DNA sequences against 20 mouse TFs. This dataset was obtained from the Dialogue on Reverse-Engineering Assessment and Methods (DREAM) competition.[143] They compared SIL (whole DNA sequence) and MIL (bag of DNA subsequences) models. For building MIL models, they used the essentially *Instance-wrapper* algorithm implemented in the WEKA package with the C4.5 decision tree as the wrapped machine learning algorithm. Individual PBM probe sequences were represented as a bag of all possible binding sites (instances) of the length 5–8 bp. The MIL models outperformed the corresponding SIL models for each of the 20 mouse TFs (average AUC score 0.94 vs. 0.71). Later Gao and Ruan[32] proposed a MIL version of the TeamD algorithm (one of the best single-instance algorithms in the DREAM5 competition), which models each subsequence (instance) of DNA separately. Using a PBM dataset of 86 mouse TFs from their previous work,[24] they demonstrated that for 78 of the 86 TFs, MIL-TeamD outperformed SIL-TeamD (average AUC score 0.94 vs. 0.90).

To predict TF-DNA binding, Zhang[144] considered the DeepBind[34] algorithm based on a deep convolutional neural network (CNN) earlier used to predict DNA– and RNA–protein binding and proposed its MIL version called Weakly-Supervised CNN (WSCNN). A single-instance learning algorithm (SIL-CNN) had the same architecture as DeepBind. WSCNN first divides each DNA sequence into multiple subsequences (instances) with a sliding window, then separately models each instance using CNN, and finally fuses the predicted scores of all instances in the same bag using different fusion operators (*Max*, *Average*, *Linear Regression*, and *Top-Bottom* Instances). They took the same PBM dataset of 86 mouse TFs[32] and found that the WSCNN (MIL-CNN) model with an average Pearson correlation coefficient (R_P) of 0.534 performed better than SIL-CNN (average $R_P = 0.491$) and the MIL-TeamD (average $R_P = 0.414$) model.

RNA modification is the process of chemical modification of the nucleotides in synthesized RNA. Traditional supervised learning approaches for predicting RNA modification sites require base-resolution data, which are often not available. Huang et al.[145] proposed the MIL framework based on a deep convolutional network, called weakly supervised learning framework (WeakRM), to predict RNA modification sites based on low-resolution datasets. Each RNA was considered as a bag consisting of regions (instances) obtained by a sliding window approach (Figure 10). Each instance is converted to a feature matrix by the one-hot encoding method of each nucleotide in the subsequence. The instance features are processed by convolutional layers and by gated attention (a three-layer neural network) to obtain a weighted summation of instance embeddings. Then the final prediction is generated based on bag-level representation. WeakRM outperformed described above WSCNN for three different types of RNA modification and was demonstrated to be able to identify regions containing the RNA modifications (key instances).[145] AUROC values for WeakRM and WSCNN were 0.896 versus 0.862 for the prediction of methylation of guanine at the N7 position, 0.909 versus 0.889 for the prediction of hydroxylmethylation of cytidine at position 5, and 0.935 versus 0.912 for acetylation of N4 position of cytidine.

5.3.6 | miRNA–mRNA interactions

mRNA regulates the synthesis of the peptides during gene expression, while microRNAs (short non-coding RNA with 18–25 nucleotides) bind to the specific sites of the target mRNA, and deactivate a part of the latter or initiate its degradation and thereby inhibit gene expression. mRNA has a large number of potential binding sites (PBS) that can be bound by given miRNA. Experimental identification of functional binding sites (FBS, 2–8 nucleotide segments) is an expensive process. In this context, computational approaches for predicting miRNA targets and their binding sites are highly desirable. In the MIL framework, each miRNA–mRNA pair is considered a bag, and each PBS of target mRNA is treated as an instance. In the classification task, a bag is positive if it contains at least one FBS (key instance), and negative if there is no FBS in the bag (given that miRNA–mRNA does not interact).

Bandyopadhyay et al.[35] developed the MBSTAR (Multiple instance learning of Binding Sites of miRNA TARgets) approach, which is based on the MIL Random Forest algorithm (MIL-RF) and can predict both miRNA-mRNA pairs (bag predictions) and target binding sites (instance predictions). The performance of MBSTAR was compared with other MIL methods (Diverse Density [DD], Expectation–Maximization DD [EM-DD], Citation kNN, and multiple instance SVM [MI-SVM]) and conventional state-of-the-art miRNA target prediction tools (TargetScan, miRanda, MirTarget2, and SVMicrO) on a dataset consisting of 9531 positive miRNA–mRNA interactions and 973 negative interactions. On the target level (miRNA–mRNA pairs prediction) MBSTAR (classification accuracy of 0.720) outperformed other MIL methods (classification accuracy of 0.685–0.486) and conventional tools (see ROC plots in original study[35]). It was found that MBSTAR achieved the highest *F*-Score of 0.337 in binding site prediction compared with conventional methods (0.274–0.049) and target level classification accuracy of 78.24% (other methods showed 57.77%–16.3%) for the validated positive interactions.

6 | TOOLKITS AND SOFTWARE

Due to the rapid development of MI methods in recent decades, many of their open-source implementations in different programming languages and tools have been proposed (Table 2). Here, we briefly review the most popular ones.

WEKA[146] is a freely available software for data analysis and visualization, as well as machine learning modeling. WEKA is written completely in Java and has a simple API and user-friendly graphical interface. It supports several popular MI classifiers, including the aforementioned CitationKNN, Diverse Density algorithm, multi-instance extensions of SVM, and wrappers.

Knowledge Extraction based on Evolutionary Learning (KEEL),[147] is another open-source machine learning software written in Java and supported by a graphical interface. KEEL provides a set of tools for building predictive models using machine learning algorithms, including some MIL algorithms. Thus, it provides different variations of the APR algorithm and several popular multi-instance methods, such as EM-DD, G3PMI, CitationKNN, and methods based on evolutionary algorithms.

MATLAB implementations of multi-instance algorithms can be found in the Matlab Toolbox for Multiple Instance Learning.[148] Multiple-Instance Learning Python Toolbox[149] is inspired by MATLAB Toolbox and provides popular multi-instance algorithms written in Python.

Various multi-instance modifications of SVM[48] methods are available online in Python. Also, some implementations of multi-instance deep neural networks can be obtained from GitHub repositories: classical MI-NN (3D-MIL-QSAR), MI-NN with attention mechanisms (AttentionDeepMIL), graph MI-NN (Graph neural networks), and Transformer-based multi-instance architectures (Set Transformer) (Table 2).

7 | COMPUTATIONAL COMPLEXITY AND COST

Additional costs for MIL model building may come from two sources. The first one is the generation of instances if they are not available. This can be very efficient in the case of the generation of subsequences of a protein/DNA sequence by the sliding window approach or relatively expensive in the case of enumeration of conformations of molecules. Conformation generation may become a bottleneck of a whole modeling pipeline and take even more time than model building itself. However, more efficient conformation generation approaches may partly solve this issue and greatly reduce these costs.[150]

TABLE 2 Multi-instance learning toolkits and software.

Tool	Programming language	Link	Description
WEKA	Java	https://waikato.github.io/weka-wiki/multi_instance_classification/	Contains a module of multi-instance classification algorithms (at least 14 algorithms) as part of the WEKA tool
KEEL	Java	https://sci2s.ugr.es/keel/category.php?cat=mul	Contains some multi-instance learning classification algorithms (APR, CitationKNN, DD, etc.)
Multiple Instance Learning Matlab toolbox	Matlab	https://github.com/DMJTax/mil	Multi-instance learning classification algorithms
Multiple-Instance Learning Python Toolbox	Python	https://github.com/jmarrietar/MILpy	Multi-instance learning classification algorithms
MILL	Matlab	https://www.cs.cmu.edu/~juny/MILL/	Contains some multi-instance learning classification algorithms (APR, DD, Citation-kNN, etc.)
MISVM	Python	https://github.com/garydoranjr/misvm	Python implementation of numerous support vector machine (SVM) algorithms for the multiple-instance (MI) learning framework
AttentionDeepMIL	Python	https://github.com/AMLab-Amsterdam/AttentionDeepMIL	PyTorch implementation of attention-based deep multiple Instance learning neural network
Set Transformer	Python	https://github.com/juho-lee/set_transformer	PyTorch implementation of the paper Set Transformer
Graph neural networks	Python	https://github.com/KostiukIvan/Multiple-instance-learning-with-graph-neural-networks	Multi-instance learning with graph neural networks
3D-MIL-QSAR	Python	https://github.com/cimm-kzn/3D-MIL-QSAR	QSAR modeling based on conformation ensembles using a multi-instance learning approach

Another issue is the complexity of different MIL representations and their processing.[9,45] Let B be the number of bags in a training set (e.g., molecules), N the number of instances (e.g., conformations), and D the dimensionality of a feature vector representing the instances. In the case of a bag-level representation where one has to compute the distance between bags, the cost is $O(B^2 \times N^2 \times D)$. This is, for example, the Citation-kNN approach. The computational costs in this case will increase very quickly with increasing the number of modeling objects and instances.

Models based on embedding individual instances in a bag into a single feature vector are more efficient and are scaled linearly with the number of bags and instances: $O(B \times N \times D)$. An example of these models is bag-wrappers. Regarding instance-based approach, for example, instance-wrappers, the main costs are created by the number of training set instances which is N times greater than for single-instance models. Therefore, approaches based on bag embeddings are the most computationally efficient followed by instance-base approaches and bag-level representation approaches. To make the latter approaches applicable to datasets with a large number of instances the size of datasets can be reduced, for example, by selection of most representative instances by clustering, however, this may also reduce the accuracy of models.

8 | PERSPECTIVES

Applications of MIL approaches are not limited to those described above. MIL can be applied to ensembles of any other molecular form, for example, tautomers. In some cases, a minor tautomer of a ligand binds to a biological target and triggers the biological response.[151] It was demonstrated[39] that accounting for tautomerism may significantly affect the performance of machine learning models for anxiolytic activity,[40] logP, and pKa prediction[152] as well as retrieval information on structure–activity relationships.[153] So far, there are no applications of MIL to model molecular properties

using a set of tautomeric forms, but this seems an attractive way to improve the performance of modeling molecular properties dependent on the underlying tautomeric form.

The application of MIL algorithms to sets of atoms-as-instances is another promising direction, which enables the identification of key atoms of a molecule that determine its properties. Although many approaches for atom-based interpretation of QSAR/QSPR models already exist,[121] very few examples of related MIL applications have been reported so far.[12,20,21] However, they can rapidly emerge taking into account that popular graph convolution methods[113,114,154] implicitly use the MIL technique. Notice that MIL could be effectively coupled with quantum chemical descriptors associated with particular atoms.

Many interesting venues for MIL application can be found when the modeled property is associated with different mechanisms of action. For example, the toxicity of molecules is usually related to the interaction of a molecule with a set of biological targets only some of which trigger a particular response. Another possible application is the modeling of properties of molecular mixtures. Unlike existing single-instance approaches,[155,156] a mixture can be considered as a bag of individual components representing instances. Establishing relationships between individual components and key instance detection by dedicated MIL algorithms would represent an additional benefit.

A unique feature of MIL approaches is the ability to identify key instances—molecular forms or segments associated with an observed property or a function of a molecule. Several studies have demonstrated successful examples of the identification of possible bioactive conformations of a molecule.[12,15,17] This is a remarkable achievement—the identification of a bioactive conformation of a molecule without information about a receptor. However, more rigorous studies are still necessary to better investigate and unlock the full potential of the identification of key instances. Partly, the validation and development of MIL algorithms for solving the KID problem are constrained by the limited amount of experimental data on active molecular forms and we anticipate the development of new benchmarks to stimulate the progress in this direction.

There is still a lack of widely accepted MIL benchmarking datasets relevant in chemistry and biology domain. Therefore, with the penetration of MIL approaches to chemo- and bioinformatics, we anticipate the publication of new well-characterized datasets suitable for benchmarking of contemporary neural network models requiring larger datasets.

9 | CONCLUSIONS

A molecule is a dynamic object representing an ensemble of different forms (conformations, tautomers, etc.) in equilibrium, and, therefore a machine learning method used to model its physico-chemical properties or biological activities should be able to handle such molecular complexity. In this context, Multi-Instance Learning considering an object as an ensemble of instances represents a promising alternative to regular machine learning techniques. Numerous studies demonstrate that MIL beats conventional single-instance learning. In particular, this concerns the modeling of the biological activity of molecules and enantioselectivity of chiral catalysts where the MIL-based 3D multi-conformation models outperform the 3D single-conformation ones accounting for the lowest-energy conformations only. 3D multi-conformation regression MIL models also outperformed state-of-the-art 2D models in many cases. However, due to the higher computational costs of the former models we recommend building 2D models first and then, if they fail, building 3D MIL models. In a limited number of chemoinformatics studies, instance wrappers were best performing among other MIL approaches for the solution of regression tasks. For classification tasks, there are not enough systematic studies to conclude whether these models outperform conventional 2D ones. More benchmarking studies are required to answer the question about the applicability and performance of different MIL models in different settings. MIL models allow also the identification of bioactive conformations, although there is still a need for deeper studies in this direction. In bioinformatics, MIL has extensively been applied to a wide range of problems such as gene–gene interactions, protein–protein interactions, protein-peptide interactions, and modeling of genomic sequences.

Despite its attractiveness, MIL has not become a very popular approach in chemoinformatics. This could be explained by several reasons. The first one concerns the availability of easy-to-use open-source tools (in contrast to a plethora of tools for classical single-instance machine learning) helping to realize different MIL scenarios. Another reason is related to the fact that the problem of multiple conformations which is most often considered by researchers can be solved by transforming a multi-instance task into a single-instance task. Because of their simplicity, such strategies as conformation descriptors averaging (or weighted averaging) are still popular, although they do not demonstrate high

performance compared with conventional 2D approaches. However, it should be noted, that the majority of these popular approaches which were not recognized as MIL follows the concept of bag-wrappers and can be considered as a branch of MIL. Also, some methodological developments are required regarding multiple representations in MIL, for example, approaches for feature/descriptor selection or determination of the applicability domain of MIL models. Recent achievements in neural network models revived the interest in MIL modeling as they propose flexible architectures of neural networks to address particular problems.

To sum up, multi-instance learning is a well-established machine learning approach. Although MIL handles molecular complexity issues much better than regular single-instance machine-learning methods, this approach still has not received the deserved attention of chemists and biologists. As reported here case studies show that MIL lives up to researchers' expectations. We hope that this review will help to broaden the application of MIL approaches in chemistry and biology.

AUTHOR CONTRIBUTIONS

Dmitry Zankov: Conceptualization (equal); formal analysis (lead); investigation (lead); visualization (lead); writing – original draft (lead). **Timur Madzhidov:** Conceptualization (equal); funding acquisition (equal); writing – review and editing (equal). **Alexandre Varnek:** Conceptualization (equal); funding acquisition (equal); writing – review and editing (equal). **Pavel Polishchuk:** Conceptualization (equal); funding acquisition (equal); writing – review and editing (lead).

FUNDING INFORMATION

PP acknowledges the support of the European Regional Development Fund (Project ENOCH no. CZ.02.1.01/0.0/0.0/16_019/0000868) and by the Ministry of Education, Youth and Sports of Czech Republic (project CZ-OPENSCREEN, CZ.02.1.01/0.0/0.0/18_046/0016118).

CONFLICT OF INTEREST STATEMENT

The authors have declared no conflicts of interest for this article.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Dmitry Zankov https://orcid.org/0000-0002-6201-3347
Timur Madzhidov https://orcid.org/0000-0002-3834-6985
Alexandre Varnek https://orcid.org/0000-0003-1886-925X
Pavel Polishchuk https://orcid.org/0000-0001-5088-8149

RELATED WIREs ARTICLES

Machine learning methods in chemoinformatics
Machine learning in drug design: Use of artificial intelligence to explore the chemical structure-biological activity relationship

REFERENCES

1. Fourches D, Muratov E, Tropsha A. Trust, but verify: on the importance of chemical structure curation in cheminformatics and QSAR modeling research. J Chem Inf Model. 2010;50(7):1189–204.
2. Dietterich TG, Lathrop RH, Lozano-Pérez T. Solving the multiple instance problem with axis-parallel rectangles. Artif Intell. 1997;89(1–2):31–71.
3. Maron O. Learning from ambiguity [dissertation]. Massachusetts Institute of Technology. 1992.
4. Buchanan BG, Feigenbaum EA. Dendral and meta-dendral: their applications dimension. Artif Intell. 1978;11(1-2):5–24.
5. Jain AN, Dietterich TG, Lathrop RH, Chapman D, Critchlow RE, Bauer BE, et al. Compass: a shape-based machine learning tool for drug design. J Comput Aided Mol Des. 1994;8(6):635–52.
6. Norton SW. Learning to recognize promoter sequences in *E. coli* by modeling uncertainty in the training data. In: Proceedings of the national conference on artificial intelligence. 1994:657–63.
7. Aikawa K. Phoneme recognition using time-warping neural networks. J Acoust Soc Jpn. 1992;13(6):395–402.
8. Rumelhart DE. A self-organizing integrated segmentation and recognition neural net. Aerosp Sensing. 1992;1991(4):496–503.

9. Carbonneau MA, Cheplygina V, Granger E, Gagnon G. Multiple instance learning: a survey of problem characteristics and applications. Pattern Recogn. 2018;77:329–53.
10. Babenko B. Multiple instance learning: algorithms and applications.In: International conference on machine learning. 2008;1–19.
11. Herrera F, Ventura S, Bello R, Cornelis C, Zafra A, Sánchez-Tarragó D, et al. Multiple instance learning: foundations and algorithms. Cham: Springer; 2016. p. 1–233.
12. Xiong J, Li Z, Wan G, Fu Z, Zhong F, Xu T, et al. Multi-instance learning of graph neural networks for aqueous pKa prediction. Bioinformatics. 2022;38(3):792–8.
13. Zankov D, Madzhidov T, Polishchuk P, Sidorov P, Varnek A. Multi-instance learning approach to the modeling of enantioselectivity of conformationally flexible organic catalysts. J Chem Inf Model. 2023;63:6629–41.
14. Davis J, Costa VS, Ray S, Page D. An integrated approach to feature invention and model construction for drug activity prediction. In: ACM international conference proceeding series. 2007:217–24.
15. Fu G, Nan X, Liu H, Patel RY, Daga PR, Chen Y, et al. Implementation of multiple-instance learning in drug activity prediction. BMC Bioinform. 2012;13:S3.
16. Nikonenko A, Zankov D, Baskin I, Madzhidov T, Polishchuk P. Multiple conformer descriptors for QSAR modeling. Mol Inform. 2021;40(11):2060030.
17. Zankov DV, Matveieva M, Nikonenko AV, Nugmanov RI, Baskin II, Varnek A, et al. QSAR modeling based on conformation ensembles using a multi-instance learning approach. J Chem Inf Model. 2021;61(10):4913–23.
18. Zankov DV, Shevelev MD, Nikonenko AV, Polishchuk PG, Rakhimbekova AI, Madzhidov TI. Multi-instance learning for structure-activity modeling for molecular properties. In: Van der Aalst WMP, Batagelj V, Ignatov DI, Khachay M, Kuskova V, Kutuzov A, et al., editors. Communications in Computer and Information Science. 8th International Conference Analysis of Images, Social networks and Texts. Kazan: Springer; 2020. p. 62–71.
19. Zankov D, Polishchuk P, Madzhidov T, Varnek A. Multi-instance learning approach to predictive modeling of catalysts enantioselectivity. Synlett. 2021;32(18):1833–6.
20. Bergeron C, Zaretzki J, Breneman C, Bennett KP. Multiple instance ranking. In: Proceedings of the 25th international conference on machine learning. New York, New York, USA: ACM Press; 2008:48–55.
21. Bergeron C, Moore G, Zaretzki J, Breneman CM, Bennett KP. Fast bundle algorithm for multiple-instance learning. IEEE Trans Pattern Anal Mach Intell. 2012;34(6):1068–79.
22. Yu G, Zeng J, Wang J, Zhang H, Zhang X, Guo M. Imbalance deep multi-instance learning for predicting isoform–isoform interactions. Int J Intell Syst. 2021;36(6):2797–824.
23. Cheng J, Bendjama K, Rittner K, Malone B. BERTMHC: improved MHC–peptide class II interaction prediction with transformer and multiple instance learning. Bioinformatics. 2021;37(22):4172–9.
24. Gao Z, Ruan J. A structure-based multiple-instance learning approach to predicting in vitrotranscription factor-DNA interaction. BMC Genomics. 2015;16(4):S3.
25. Zhang YP, Zha Y, Li X, Zhao S, Du X. Using the multi-instance learning method to predict protein-protein interactions with domain information. Lect Notes Comput Sci. 2014;8818:249–59.
26. Xu Y, Luo C, Qian M, Huang X, Zhu S. MHC2MIL: a novel multiple instance learning based method for MHC-II peptide binding prediction by considering peptide flanking region and residue positionss. BMC Genomics. 2014;15(S9):S9.
27. Wu JS, Huang SJ, Zhou ZH. Genome-wide protein function prediction through multi-instance multi-label learning. IEEE/ACM Trans Comput Biol Bioinform. 2014;11(5):891–902.
28. Zhang Y, Chen Y, Bao W, Cao Y. A hybrid deep neural network for the prediction of in-vivo protein-DNA binding by combining multiple-instance learning. Lect Notes Comput Sci. 2021;12838:374–84.
29. Emamjomeh A, Choobineh D, Hajieghrari B, MahdiNezhad N, Khodavirdipour A. DNA–protein interaction: identification, prediction and data analysis. Mol Biol Rep. 2019;46(3):3571–96.
30. Zeng J, Yu G, Wang J, Guo M, Zhang X. DMIL-III: isoform-isoform interaction prediction using deep multi-instance learning method. In: Proceedings – 2019 IEEE international conference on bioinformatics and biomedicine, BIBM 2019. 2019:171–6.
31. Pan X, Bin SH. Predicting RNA-protein binding sites and motifs through combining local and global deep convolutional neural networks. Bioinformatics. 2018;34(20):3427–36.
32. Gao Z, Ruan J. Computational modeling of in vivo and in vitro protein-DNA interactions by multiple instance learning. Bioinformatics. 2017;33(14):2097–105.
33. Abbasi WA, Asif A, Andleeb S, Minhas F. CaMELS: in silico prediction of calmodulin binding proteins and their binding sites. Proteins Struct Funct Bioinform. 2017;85(9):1724–40.
34. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol. 2015;33(8):831–8.
35. Bandyopadhyay S, Ghosh D, Mitra R, Zhao Z. MBSTAR: multiple instance learning for predicting specific functional binding sites in microRNA targets. Sci Rep. 2015;5(1):1–12.
36. Liu G, Wu J, Zhou ZH. Key instance detection in multi-instance learning. J Mach Learn Res. 2012;25:253–68.
37. Zhou ZH, Xu JM. On the relation between multi-instance learning and semi-supervised learning. In: ACM international conference proceeding series. New York, NY: ACM. 2007:1167–74. (ICML'07; vol. 227).
38. Carbonneau MA. Multiple instance learning under real-world conditions [PhD Thesis]. The Université du Québec. 2017:1–271.

39. Masand VH, Mahajan DT, Ben Hadda T, Jawarkar RD, Alafeefy AM, Rastija V, et al. Does tautomerism influence the outcome of QSAR modeling? Med Chem Res. 2014;23(4):1742–57.
40. Toropova AP, Toropov AA, Benfenati E, Gini G, Leszczynska D, Leszczynski J. QSAR modeling of anxiolytic activity taking into account the presence of keto- and enol-tautomers by balance of correlations with ideal slopes. Cent Eur J Chem. 2011;9(5):846–54.
41. Glavatskikh M, Madzhidov T, Solov'ev V, Marcou G, Horvath D, Graton J, et al. Predictive models for halogen-bond basicity of binding sites of polyfunctional molecules. Mol Inform. 2016;35(2):70–80.
42. Tibo A, Jaeger M, Frasconi P. Learning and interpreting multi-multi-instance learning networks. J Mach Learn Res. 2020;21:191–3.
43. Zhou ZH, Zhang ML. Multi-instance multi-label learning with application to scene classification. Adv Neural Inf Process Syst. 2007; 1609–16.
44. Kriegel HP, Pryakhin A, Schubert M. An EM-approach for clustering multi-instance objects. In: Pacific-Asia conference on knowledge discovery and data mining. 2006:139–48.
45. Amores J. Multiple instance classification: review, taxonomy and comparative study. Artif Intell. 2013;201:81–105.
46. Foulds J, Frank E. A review of multi-instance learning assumptions. Knowl Eng Rev. 2010;25(1):1–25.
47. Dooly DR, Zhang Q, Goldman SA, Amar RA. Multiple-instance learning of real-valued data. J Mach Learn Res. 2003;3(4-5):651–78.
48. Doran G, Ray S. A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. Mach Learn. 2014;97(1-2):79–102.
49. Fatima S, Ali S, Kim HC. A comprehensive review on multiple instance learning. Electronics. 2023;12(20):4323.
50. Xu X. Statistical learning in multiple instance problems [Internet]. Hamilton, New Zealand: University of Waikato; 2003.
51. Foulds J. Learning instance weights in multi-instance learning. Hamilton, New Zealand: University of Waikato; 2008.
52. Wang J, Zucker JD. Solving multiple-instance problem: a lazy learning approach. In: Proceedings 17th IEEE international conference on machine learning. 2000:1119–25.
53. Gärtner T, Flach PA, Kowalczyk A, Smola AJ. Multi-instance kernels. Proceedings of the Nineteenth International Conference onMachine Learning. ICML; 2002. p. 179–86.
54. Cheplygina V, Tax DMJ, Loog M. Multiple instance learning with bag dissimilarities. Pattern Recogn. 2015;48(1):264–75.
55. Huang K, Fu T, Gao W, Zhao Y, Roohani Y, Leskovec J, et al. Artificial intelligence foundation for therapeutic science. Nat Chem Biol. 2022;18(10):1033–6.
56. Maron O, Lozano-Pérez T. A framework for multiple-instance learning. In: Advances in neural information processing systems. 1998: 570–6.
57. Zhang Q, Goldman SA. Em-dd: an improved multiple-instance learning technique. In: Advances in neural information processing systems. 2002:1073–80.
58. Ray S, Craven M. Supervised versus multiple instance learning: an empirical comparison. In: ICML 2005 – proceedings of the 22nd international conference on machine learning. 2005:697–704.
59. Auer P, Ortner R. A boosting approach to multiple instance learning. Lect Notes Artif Intell. 2004;3201:63–74.
60. Chevaleyre Y, Zucker JD. Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. Application to the mutagenesis problem. Lect Notes Artif Intell. 2001;2056:204–14.
61. Blockeel H, Page D, Srinivasan A. Multi-instance tree learning. In: ICML 2005 – Proceedings of the 22nd international conference on machine learning. 2005:57–64.
62. Bjerring L, Frank E. Beyond trees: adopting MITI to learn rules and ensemble classifiers for multi-instance data. Lect Notes Comput Sci. 2011;7106:41–50.
63. Leistner C, Saffari A, Bischof H. MIForests: multiple-instance learning with randomized trees. Lect Notes Comput Sci. 2010;6316: 29–42.
64. Zafra A, Ventura S. G3P-MI: a genetic programming algorithm for multiple instance learning. Inf Sci. 2010;180(23):4496–513.
65. Chen Y, Wang JZ. Image categorization by learning and reasoning with regions. J Mach Learn Res. 2004;5:913–39.
66. Chen Y, Bi J, Wang JZ. MILES: multiple-instance learning via embedded instance selection. IEEE Trans Pattern Anal Mach Intell. 2006;28(12):1931–47.
67. Ramon J, de Raedt L. Multi-instance neural networks. In: Proceedings of the ICML-2000 workshop on attribute-value and relational learning. 2000:53–60.
68. Zhou ZH, Zhang ML. Neural networks for multi-instance learning. In: Proceedings of the international conference on intelligent information technology, Beijing, China. 2002.
69. Zhang ML, Zhou ZH. Improve multi-instance neural networks through feature selection. Neural Process Lett. 2004;19(1):1–10.
70. Zhang ML, Zhou ZH. Ensembles of multi-instance neural networks. In: IFIP advances in information and communication technology. 2005. p. 471–4.
71. Zhang ML, Zhou ZH. Adapting RBF neural networks to multi-instance learning. Neural Process Lett. 2006;23(1):1–26.
72. Zhang ML, Zhou ZH. Multi-instance regression algorithm based on neural network. Ruan Jian Xue Bao. 2003;14(7):1238–42.
73. Wang X, Yan Y, Tang P, Bai X, Liu W. Revisiting multiple instance neural networks. Pattern Recogn. 2018;74:15–24.
74. Ilse M, Tomczak JM, Welling M. Attention-based deep multiple instance learning. 35th Int Conf Mach Learn ICML 2018. 2018;5: 3376–91.
75. Yan Y, Wang X, Fang J, Liu W, Huang J, Zhu J, et al. Deep multi-instance learning with dynamic pooling. ACML. 2018;95:662–77.
76. Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. In: Advances in neural information processing systems. 2017:3857–67.

77. Wang K, Oramas J, Tuytelaars T. In defense of LSTMs for addressing multiple instance learning problems. Lect Notes Comput Sci. 2021;12627:444–60.
78. LeCun Y, Cortes C. MNIST handwritten digit database. Florham Park: AT & T Labs; 2010. p. 23.
79. Lee J, Lee Y, Kim J, Kosiorek AR, Choi S, Teh YW. Set transformer: a framework for attention-based permutation-invariant neural networks. Int Conf Mach Learn. 2018;97:3744–53.
80. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017; 30:5999–6009.
81. Yin S, Peng Q, Li H, Zhang Z, You X, Liu H, et al. Multi-instance deep learning with graph convolutional neural networks for diagnosis of kidney diseases using ultrasound imaging. Lect Notes Comput Sci. 2019;11840:146–54.
82. Widrich M, Schäfl B, Pavlovic M, Ramsauer H, Gruber L, Holzleitner M, et al. Modern Hopfield networks and attention for immune repertoire classification. Adv Neural Inf Process Syst. 2020;33:18832–45.
83. Tu M, Huang J, He X, Zhou B. Multiple instance learning with graph neural networks [Internet]. 2019.
84. D'avila Garcez AS, Zaverucha G. Multi-instance learning using recurrent neural networks. In: The 2012 International Joint Conference on Neural Networks (IJCNN). 2012:10–5.
85. Early J, Evers C, Ramchurn S. Model agnostic interpretability for multiple instance learning. International Conference on Learning Representations. 2022. https://openreview.net/forum?id=KSSfF5lMIAg. Accessed 21 Nov 2023.
86. Andrews S, Tsochantaridis I, Hofmann T. Support vector machines for multiple-instance learning. In: Proceedings of the 15th international conference on neural information processing systems. Cambridge, MA: MIT Press. 2002:577–84.
87. Wang D, Li J, Zhang B. Multiple-instance learning via random walk. Lect Notes Comput Sci. 2006;4212:473–84.
88. Zhou D, Schölkopf B, Hofmann T. Semi-supervised learning on directed graphs. Adv Neural Inf Process Syst. 2005;17:1633–40.
89. Carbonneau MA, Granger E, Raymond AJ, Gagnon G. Robust multiple-instance learning ensembles using random subspace instance selection. Pattern Recogn. 2016;58:83–99.
90. Liu J, Qiao R, Li Y, Li S. Witness detection in multi-instance regression and its application for age estimation. Multimed Tools Appl. 2019;78(23):33703–22.
91. Li YF, Kwok JT, Tsang IW, Zhou ZH. A convex method for locating regions of interest with multi-instance learning. Lect Notes Comput Sci. 2009;5782:15–30.
92. Lin Z, Feng M, Dos Santos CN, Yu M, Xiang B, Zhou B, et al. A structured self-attentive sentence embedding. In: Conference paper in 5th international conference on learning representations (ICLR 2017). 2017.
93. Li XC, Zhan DC, Yang JQ, Shi Y. Deep multiple instance selection. Sci China Inf Sci. 2021;64(3):1–15.
94. Xu K, Ba JL, Kiros R, Cho K, Courville A, Salakhutdinov R, et al. Show, attend and tell: neural image caption generation with visual attention. In: 32nd international conference on machine learning, ICML 2015. 2015:2048–57.
95. Shin B, Cho J, Yu H, Choi S. Sparse network inversion for key instance detection in multiple instance learning. In: Proceedings – international conference on pattern recognition. 2020:4083–90.
96. Kindermann J, Linden A. Inversion of neural networks by gradient descent. Parallel Comput. 1990;14(3):277–86.
97. Looks M, Herreshoff M, Hutchins D, Norvig P. Deep multiple instance learning with Gaussian weighting. In: ICLR 2020. 2020:1–12.
98. Haab J, Deutschmann N, Martínez MR. Is attention interpretation? A quantitative assessment on sets. In: XKDD'22, September 19th 2022, Grenoble, France. 2022;1(1).
99. Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?" explaining the predictions of any classifier. In: NAACL-HLT 2016–2016 conference of the north American chapter of the association for computational linguistics: human language technologies, proceedings of the demonstrations session. 2016:97–101.
100. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst. 2017;30:4766–75.
101. Bak A. Two decades of 4d-qsar: a dying art or staging a comeback? Int J Mol Sci. 2021;22(10):5212.
102. Hopfinger AJ, Wang S, Tokarski JS, Jin B, Albuquerque M, Madhav PJ, et al. Construction of 3D-QSAR models using the 4D-QSAR analysis formalism. J Am Chem Soc. 1997;119(43):10509–24.
103. Vedani A, Dobler M. Multi-dimensional QSAR in drug research. Progress in drug research. Basel: Birkhäuser Basel; 2000. p. 105–35.
104. Mekenyan OG, Ivanov JM, Veith GD, Bradbury SP. Dynamic QSAR: a new search for active conformations and significant stereo-electronic indices. Quant Struct Relationships. 1994;13(3):302–7.
105. Pissurlenkar RRS, Khedkar VM, Iyer RP, Coutinho EC. Ensemble QSAR: a QSAR method based on conformational ensembles and metric descriptors. J Comput Chem. 2011;32(10):2204–18.
106. Adekoya A, Dong X, Ebalunode J, Zheng W. Development of improved models for phosphodiesterase-4 inhibitors with a multi-conformational structure-based QSAR method. Curr Chem Genom. 2009;3(1):54–61.
107. Bak A, Polanski J. Modeling robust QSAR 3: SOM-4D-QSAR with iterative variable elimination IVE-PLS: application to steroid, azo dye, and benzoic acid series. J Chem Inf Model. 2007;47(4):1469–80.
108. Kuz'min VE, Artemenko AG, Polischuk PG, Muratov EN, Hromov AI, Liahovskiy AV, et al. Hierarchic system of QSAR models (1D–4D) on the base of simplex representation of molecular structure. J Mol Model. 2005;11(6):457–67.
109. Kuz'min VE, Artemenko AG, Muratov EN. Hierarchical QSAR technology based on the simplex representation of molecular structure. J Comput Aided Mol Des. 2008;22:403–21.
110. Bonachéra F, Parent B, Barbosa F, Froloff N, Horvath D. Fuzzy tricentric pharmacophore fingerprints. 1. Topological fuzzy pharmacophore triplets and adapted molecular similarity scoring schemes. J Chem Inf Model. 2006;46(6):2457–77.

111. Welling M, Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: International conference on learning representations (ICLR 2017). 2016.
112. Jiang D, Wu Z, Hsieh CY, Chen G, Liao B, Wang Z, et al. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. J Chem. 2021;13(1):1–23.
113. Reiser P, Neubert M, Eberhard A, Torresi L, Zhou C, Shao C, et al. Graph neural networks for materials science and chemistry. Commun Mater. 2022;3(1):1–18.
114. Wieder O, Kohlbacher S, Kuenemann M, Garon A, Ducrot P, Seidel T, et al. A compact review of molecular property prediction with graph neural networks. Drug Discov Today Technol. 2020;37:1–12.
115. Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem. 2009;31(2): 455–61.
116. Zahrt AF, Henle JJ, Rose BT, Wang Y, Darrow WT, Denmark SE. Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning. Science. 2019;363(6424):eaau5631.
117. Henle JJ, Zahrt AF, Rose BT, Darrow WT, Wang Y, Denmark SE. Development of a computer-guided workflow for catalyst optimization. Descriptor validation, subset selection, and training set analysis. J Am Chem Soc. 2020;142(26):11578–92.
118. Melville JL, Lovelock KRJ, Wilson C, Allbutt B, Burke EK, Lygo B, et al. Exploring phase-transfer catalysis with molecular dynamics and 3D/4D quantitative structure – selectivity relationships. J Chem Inf Model. 2005;45(4):971–81.
119. Melville JL, Andrews BI, Lygo B, Hirst JD. Computational screening of combinatorial catalyst libraries. Chem Commun. 2004;4(12): 1410–1.
120. Mandrelli F, Blond A, James T, Kim H, List B. Deracemizing α-branched carboxylic acids by catalytic asymmetric protonation of bis-silyl ketene acetals with water or methanol. Angew Chem – Int Ed. 2019;58(33):11479–82.
121. Polishchuk P. Interpretation of quantitative structure—activity relationship models: past, present, and future. J Chem Inf Model. 2017; 57(11):2618–39.
122. Riniker S, Landrum GA. Open-source platform to benchmark fingerprints for ligand-based virtual screening. J Chem. 2013;5(5):1–17.
123. Polishchuk PG, Kuźmin VE, Artemenko AG, Muratov EN. Universal approach for structural interpretation of qsar/qspr models. Mol Inform. 2013;32(9-10):843–53.
124. You H, Kim GE, Na CH, Lee S, Lee CJ, Cho KH, et al. An empirical model for gas phase acidity and basicity estimation. SAR QSAR Environ Res. 2014;25(2):91–115.
125. Oliferenko AA, Oliferenko PV, Huddleston JG, Rogers RD, Palyulin VA, Zefirov NS, et al. Theoretical scales of hydrogen bond acidity and basicity for application in QSAR/QSPR studies and drug design. Partitioning of aliphatic compounds. J Chem Inf Comput Sci. 2004;44(3):1042–55.
126. Habibi-Yangjeh A, Danandeh-Jenagharad M, Nooshyar M. Application of artificial neural networks for predicting the aqueous acidity of various phenols using QSAR. J Mol Model. 2006;12(3):338–47.
127. Sheridan RP, Korzekwa KR, Torres RA, Walker MJ. Empirical regioselectivity models for human cytochromes P450 3A4, 2D6, and 2C9. J Med Chem. 2007;50(14):3173–84.
128. Yamakawa H, Maruhashi K, Nakao Y. Predicting types of protein-protein interactions using a multiple-instance learning model. Lect Notes Comput Sci. 2007;4384:42–53.
129. Yang J. MILL: A multiple instance learning library [Internet].
130. Zhou ZH, Zhang ML, Huang SJ, Li YF. Multi-instance multi-label learning. Artif Intell. 2012;176(1):2291–320.
131. Li YX, Ji S, Kumar S, Ye J, Zhou ZH. Drosophila gene expression pattern annotation through multi-instance multi-label learning. IEEE/ACM Trans Comput Biol Bioinform. 2011;9(1):98–112.
132. Xu XS, Xue X, Zhou ZH. Ensemble multi-instance multi-label learning approach for video annotation task. In: Proceedings of the 19th ACM international conference on multimedia. 2011:1153–6.
133. Zhang ML. A k-nearest neighbor based multi-instance multi-label learning algorithm. In: Proceedings – International Conference on Tools with Artificial Intelligence, ICTAI. 2010. p. 207–12.
134. Yang SH, Zha H, Hu BG. Dirichlet-Bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora. Adv Neural Inf Process Syst. 2009;22.
135. Li HD, Menon R, Eksi R, Guerler A, Zhang Y, Omenn GS, et al. A network of splice isoforms for the mouse. Sci Rep. 2016;6(1):1–11.
136. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, et al. Human protein reference database—2009 update. Nucleic Acids Res. 2009;37(suppl_1):D767–72.
137. Rammensee HG, Friede T, Stevanović S. MHC ligands and peptide motifs: first listing. Immunogenetics. 1995;41(4):178–228.
138. Pfeifer N, Kohlbacher O. Multiple instance learning allows MHC class II epitope predictions across alleles. Lect Notes Comput Sci. 2008;5251:210–21.
139. El-Manzalawy Y, Dobbs D, Honavar V. Predicting MHC-II binding affinity using multiple instance regression. IEEE/ACM Trans Comput Biol Bioinform. 2011;8(4):1067–79.
140. Wang P, Sidney J, Dow C, Mothé B, Sette A, Peters B. A systematic assessment of MHC class II peptide binding predictions and evaluation of a consensus approach. PLoS Comput Biol. 2008;4(4):e1000048.
141. Reynisson B, Alvarez B, Paul S, Peters B, Nielsen M. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. Nucleic Acids Res. 2021;48(W1): W449–54.

142. Andrews C, Xu Y, Kirberger M, Yang JJ. Structural aspects and prediction of calmodulin-binding proteins. Int J Mol Sci. 2021;22(1): 1–26.
143. Stolovitzky G, Monroe D, Califano A. Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference. Ann N Y Acad Sci. 2007;1115(1):1–22.
144. Zhang Q, Zhu L, Bao W, Huang DS. Weakly-supervised convolutional neural network architecture for predicting protein-DNA binding. IEEE/ACM Trans Comput Biol Bioinform. 2020;17(2):679–89.
145. Huang D, Song B, Wei J, Su J, Coenen F, Meng J. Weakly supervised learning of RNA modifications from low-resolution epitranscriptome data. Bioinformatics. 2021;37(suppl_1):I222–30.
146. Witten IH, Frank E, Geller J. Data mining: practical machine learning tools and techniques with Java implementations. SIGMOD Rec. 2002;31(1):76–7.
147. Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, et al. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Mult Valued Log Soft Comput. 2011;17(2-3):255–87.
148. Tax DMJ. {MIL}: A matlab toolbox for multiple instance learning [Internet]. 2013.
149. Arrieta JM. MILpy: multiple-instance learning python toolbox [Internet]. 2016.
150. Seidel T, Permann C, Wieder O, Kohlbacher SM, Langer T. High-quality conformer generation with CONFORGE: algorithm and performance assessment. J Chem Inf Model. 2023;63:5549–70.
151. Milletti F, Vulpetti A. Tautomer preference in PDB complexes and its impact on structure-based drug discovery. J Chem Inf Model. 2010;50(6):1062–74.
152. Baker CM, Kidley NJ, Papachristos K, Hotson M, Carson R, Gravestock D, et al. Tautomer standardization in chemical databases: deriving business rules from quantum chemistry. J Chem Inf Model. 2020;60(8):3781–91.
153. Masand VH, Mahajan DT, Gramatica P, Barlow J. Tautomerism and multiple modelling enhance the efficacy of QSAR: antimalarial activity of phosphoramidate and phosphorothioamidate analogues of amiprophos methyl. Med Chem Res. 2014;23(11):4825–35.
154. Xiong J, Xiong Z, Chen K, Jiang H, Zheng M. Graph neural networks for automated de novo drug design. Drug Discov Today. 2021; 26(6):1382–93.
155. Muratov EN, Varlamova EV, Artemenko AG, Polishchuk PG, Kuz'Min VE. Existing and developing approaches for QSAR analysis of mixtures. Mol Inform. 2012;31(3-4):202–21.
156. Belfield SJ, Firman JW, Enoch SJ, Madden JC, Erik Tollefsen K, Cronin MTD. A review of quantitative structure-activity relationship modelling approaches to predict the toxicity of mixtures. Comput Toxicol. 2022;25:100251.

How to cite this article: Zankov D, Madzhidov T, Varnek A, Polishchuk P. Chemical complexity challenge: Is multi-instance machine learning a solution? WIREs Comput Mol Sci. 2024;14(1):e1698. https://doi.org/10.1002/wcms.1698