Cluster-Aware Few-Shot Molecular Property Prediction With Factor Disentanglement

Haodong Zhang, Tao Ren, Yifan Wang, Fanchun Meng, Wei Ju, and Ying Tian

***Abstract*—Molecular property prediction plays a crucial role in drug discovery, but is always challenged by the limited number of effective labels. Compared with existing methods, we argue that the auxiliary properties of the molecule and the heterogeneous structure of different property prediction tasks have always been ignored. In this article, we propose a novel framework termed Meta-DREAM for few-shot molecular property prediction, which tailors to learning the transferable knowledge within different clusters of tasks. Specifically, we first construct a heterogeneous molecule relation graph (HMRG) with molecule–property and molecule–molecule relations to utilize many-to-many correlations between properties and molecules. The meta-learning episode can, then, be reformulated as a subgraph of HMRG. Next, we propose a disentangled graph encoder to explicitly discriminate the underlying factors of the task. In addition, we introduce a soft clustering module to group each factorized task representation into appropriate clusters and preserve knowledge generalization within a cluster and customization among clusters. In this way, each disentangled factor serves as a cluster-aware parameter gate for the task-specific meta-learner. Extensive experiments on five commonly used molecular datasets show that Meta-DREAM consistently outperforms existing state-of-the-art methods and verifies the effectiveness of each module.**

***Index Terms*—Disentangled representation learning, few-shot learning, meta-learning, molecular property prediction.**

I. Introduction

DRUG discovery is a critical industry closely linked to human health, but it is also notoriously labor-intensive and time-consuming. A key task in drug discovery is molecular property prediction. During the initial stages of lead optimization, it is essential for researchers to select candidates and conduct virtual screening from a large number of molecules to prevent the misallocation of resources on molecules that probably lack the required properties. As the number of available molecules increases and the cost of developing new drugs continues to rise, the use of deep learning technologies to accelerate this process is becoming increasingly important.

Actually, there are several deep learning-based approaches to learning about molecules and their properties [1], [2]. Early works regard molecules as strings via simplified molecular-input line-entry (*SMILES* [3]) and leverage sequential models to learn molecular representation for property prediction [4], [5]. To more effectively capture the pharmacological characteristics of molecules, each molecule can be represented as a graph and employs graph neural networks (GNNs) to learn molecular structures [6], [7]. These deep learning-based methods achieve effective molecular property prediction while highly relying on large-scale labeled training samples. Since the cost of obtaining molecular properties with screening experiments is expensive, it is often the case that only a limited number of molecules exhibit a particular set of properties [7], [8].

In parallel, the challenge of learning in domains with limited data has been progressively considered by few-shot learning. A prominent solution is the meta-learning paradigm, focusing on developing a learner that is effective at adapting to new tasks. Typically, matching networks [9], which aims to learn a distance metric for prediction in a nonparametric way, have been utilized for few-shot molecular property prediction [8]. And model-agnostic meta-learning (MAML) algorithm [10], which proposes to effectively initialize a base learner that can be quickly adapted to new tasks, has also led numerous follow-up studies [7], [11], [12], [13], [14].

However, these existing works still remain unsatisfactory due to the following limitations.

1) Neglect multiple properties of the molecule. Unlike typical few-shot learning scenarios such as image classification, a molecule can exhibit multiple properties at once, which often correlate with one another [15], [16]. For example, Metformin, a drug used to treat diabetes, may cause digestive side effects (such as diarrhea and stomach pain) and metabolic changes (such as decreased vitamin B12 levels) [17].
2) Unable to capture the varied complexities specific to different tasks. As shown in Fig. 1(a), most existing meta-learning methods assume the transferable knowl-

Received 16 May 2024; revised 30 December 2024 and 4 April 2025; accepted 8 July 2025. Date of publication 15 August 2025; date of current version 31 October 2025. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant N2217003, in part by the Fundamental Research Funds for the Central Universities in University of International Business and Economics (UIBE) under Grant 23QN02, in part by the Natural Science Foundation of Liaoning Province under Grant 2023010411-JH3/101, in part by Liaoning Provincial Department of Science and Technology Fund Project under Grant 2022-YGJC-43, and in part by the Key Laboratory of Optical Information and Simulation Technology of Liaoning Province. *(Corresponding authors: Ying Tian; Yifan Wang.)*

Haodong Zhang, Tao Ren, and Fanchun Meng are with the Software College, Northeastern University, Shenyang 110819, China (e-mail: 2110496@stu.neu.edu.cn; chinarentao@163.com; chinafcmeng@163.com).

Yifan Wang is with the School of Information Technology and Management, University of International Business and Economics, Beijing 100029, China (e-mail: yifanwang@uibe.edu.cn).

Wei Ju is with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: juwei@scu.edu.cn).

Ying Tian is with the Department of Otorhinolaryngology, First Affiliated Hospital of China Medical University, Shenyang 110819, China (e-mail: ty31505@163.com).

Digital Object Identifier 10.1109/TNNLS.2025.3590240

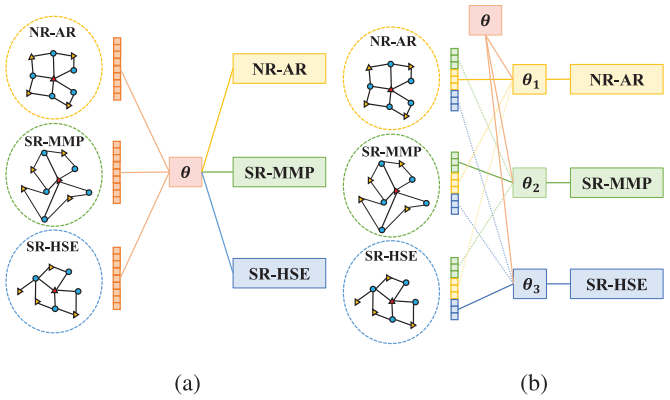


Fig. 1. Example of the difference between the proposed Meta-DREAM and previous approaches on few-shot molecular property prediction. NR-AR, SR-MMP, and SR-HSE present different property prediction tasks from Tox21 dataset. (a) Previous method. (b) Meta-DREAM.

edge is shared globally across all tasks. This assumption can lead to suboptimal performance when these meta-learning algorithms are applied to molecular property prediction tasks that originate from varied distributions. For example, a straightforward method to distinguish between acidity and alkalinity in organic compounds is to identify carboxyl and hydroxyl groups. However, when predicting another property, such as a compound's fluorescence, the distinguishing patterns might rely on entirely different functional groups, such as aromatic rings or conjugated systems. This highlights that different molecular properties often exhibit distinct and prominent patterns, requiring unique sets of features for effective prediction.

Recently, there has been a growing interest in disentangled representation learning on heterogeneous data, which focuses on developing factorized representations that uncover and clarify the fundamental or explanatory characteristics of the data. In the case of meta-learning for molecular property prediction, since humans tend to organize tasks into similarity-based clusters and focus on learning within these clusters rather than across them, it is suggested that disentangled representation learning be used to cluster tasks into multiple states based on task similarity, as shown in Fig. 1(b). In this way, when faced with a new molecular property prediction task, it can quickly use the relevant knowledge from the cluster it is aligned with, or form a new cluster if it differs significantly from existing ones.

Toward this end, we propose *Meta-DREAM*, a novel Meta-learning architecture based on Disentangled REpresentation leArning framework for Molecular property prediction, which enables the customization of knowledge across different clusters of tasks to enhance the effectiveness of meta-learning. Specifically, we first construct a heterogeneous molecule relation graph (HMRG) to model the many-to-many relations among properties and molecules. And an episode of a task can be reformulated as a subgraph, including corresponding molecule nodes, target property, and auxiliary property nodes. Then, a disentangled graph encoder is proposed to learn the subgraph representation, which discerns the task into different aspects. Next, we introduce a soft clustering method to accurately assign each task to locate the appropriate cluster for each task. In this way, a globally shared parameter initialization is tailored for each cluster through a parameter gate, which then serves as the base initialization for all molecular property prediction tasks within that cluster.

To summarize, in this article, we make the following contributions.

1) *Conceptual:* We consider the auxiliary properties and heterogeneity of the molecular prediction tasks and construct a molecule relation graph so that the many-to-many correlations of properties and molecules could be used under the few-shot learning scenario.
2) *Methodological:* We propose a novel meta-learning architecture based on the disentangled representation framework and learn the task clustering structure, enabling the explicit customization of transferable knowledge for distinct clusters of tasks.
3) *Experimental:* We conduct comprehensive experiments on five benchmark datasets to evaluate Meta-DREAM. The experimental results demonstrate the effectiveness of our proposed framework against existing baselines for few-shot molecular property prediction.

II. Related Work

A. Meta-Learning

Meta-learning, as a popular framework coming into the public eye, has been proven to be an effective solution to the few-shot problem [18], [19], [20], [21]. Meta-learning falls into three categories: 1) model-based approaches leverage augmented long-term memory storing in base model to draw meta-knowledge [22], [23]; 2) metric-based approaches aim at learning a generative metrics to measure the distance between samples [24], [25]; and 3) optimization-based improves the optimization to learn an appropriate initialization for rapid transferring to new tasks [10], [26], [27]. For example, MANN [22] uses long short term memory (LSTM) to gather memories learned from previous tasks and serve to new task. A prototypical network [24] learns a metric space to quantify the prototype representation of each class and classifies samples by measuring distance. MAML [10] provides a two-step training framework to learn initialized parameters capable of rapid transferring to new tasks for meta-learner.

Motivated by the blossom of meta-learning, many efforts have tried to incorporate meta-learning into GNN to tackle the challenge of lacking valid samples in real scenario [28], [29], [30], [31], [32]. For example, TPN [28] proposes a transductive inference framework to learn propagate labels from labeled instances to unlabeled instances. GPN [29] constructs prototypes of each class and proposes a meta-learner to propagate messages between prototypes. Chauhan et al. [30] propose a super-class prototypical network based on graph spectral measures for the few-shot graph classification task. However, most of these works are not tailored for molecular property prediction, which is challenged by the complexity of domain knowledge and data structure.

B. Disentangled Representation Learning

Disentangled representation learning focuses on developing factorized representations that effectively identify and separate the underlying explanatory factors hidden within observed data [33]. The method allows models to better understand the complexities and interrelations within the data, enhancing their ability to perform detailed analyses or adjustments of these underlying factors. Existing efforts in disentangled representation learning have primarily applied to fields such as computer vision [34], [35], recommendation system [36], [37], [38], [39], [40], and natural language processing [41], [42]. For example, β-VAE [34] introduces a hyperparameter β as the weight of Kullback-Leibler (KL) divergence to strike a balance between the independence of the latent variables and the accuracy of the reconstruction. IDEL [41] optimizes the upper bound of mutual information to disentangle the style and content of texts, which performs well on text-style transfer and conditional text generation. MacridVAE [43] leverages macro- and microlevels of representation disentanglement to understand hierarchical user intentions.

Recently, the usage of disentangled representation learning on graph-structured data has gained notable attention [44], [45], [46], [47]. For example, DisenGCN [44] utilizes a neighborhood routing mechanism that divides the node's neighborhood into distinct compartments to learn disentangled representations. DisenHAN [45] iteratively identifies key aspects of different relations in a heterogeneous information network (HIN) and semantically propagates the corresponding message. DGCL [46] introduces a contrastive learning framework for disentangled graph representations. However, how to learn a disentangled representation for few-shot learning tasks, especially for molecular property prediction, is largely unexplored.

C. Few-Shot Molecular Property Prediction

Due to the diversity of molecular properties and the scarcely available effective labels [48], [49], few-shot molecular property prediction has become an emerging field of research [7], [13], [14], [16], [50]. These approaches aim to develop a prediction model through a series of predictive tasks related to property estimation, with the capability to extend its application to predict unseen properties based on a limited number of labeled molecules. Meta-MGNN [7] applies self-supervised module and self-attentive weights to identify the heterogeneity of different tasks. PAR [16] constructs a property-aware embedding function to learn a task-relevant substructure-aware space. HSL-RG [13] uses graph kernels to generate molecular structural information and adopts a task-adaptive algorithm to provide customization knowledge for different tasks. GS-Meta [15] constructs a molecule-property relation graph and introduces a sampling scheduler based on reinforcement learning to ensure the independence of sampled tasks. PACIA [51] uses a parameter-efficient adapter to adapt the encoder at the task level and the predictor at the query level.

Although these prior approaches achieve satisfactory performance on extensive benchmarks, they mainly concentrate on excavating universal meta-knowledge between different tasks and devote to improve the transferring to new tasks without considering tailored initialization with customized knowledge. Here, we provide a scheme to leverage task-specific knowledge via the lens of clustering.

III. Preliminary and Problem Definition

A. Graph Neural Network

GNNs [52], [53], [54], [55] have received increasing attention due to their ability to effectively handle structured data, which is capable of learning representations for graph components such as nodes, edges, and the entire graph via message-passing mechanism. Specifically, let the embedding of a node i at layer $(l-1)$ represented as $h_i^{(l-1)}$. In the lth layer of a GNN, the embeddings of i's neighbors are aggregated first and then merged with i's previous layer embedding to iteratively update i's representation. Formally, $h_i^{(l)}$ can be calculated as follows:

$$\begin{aligned} h_i^{(l)} &= \mathrm{GNN}^{(l)}\left(h_i^{(l-1)}, h_j^{(l-1)}, h_{i,j}^{(l-1)}, w_{i,j}^{(l)} | j \in \mathcal{N}(i)\right) \\ &= \mathcal{C}^{(l)}\left(h_i^{(l-1)}, \mathcal{A}^{(l)}\left(h_j^{(l-1)}, h_{i,j}^{(l-1)}, w_{i,j}^{(l)} | j \in \mathcal{N}(i)\right)\right) \end{aligned} \tag{1}$$

where $h_i^{(l)}$ indicates the embedding of i at layer $l \in \{1, \ldots, L\}$, $h_{i,j}^{(l-1)}$ is the edge embedding initialized according to edge type, $\mathcal{N}(i)$ denotes the neighbors of i, and $w_{i,j}^{(l)}$ is the edge weight. At each layer, $\mathcal{A}^{(l)}$ and $\mathcal{C}^{(l)}$ are the message aggregating and embedding updating function, respectively. After the final layer L, a readout function aggregates all node representations to produce the graph-level representations as follows:

$$h_{\mathcal{G}} = \mathrm{READOUT}\left(\left\{h_i^{(L)}\right\}_{i \in \mathcal{V}}\right) \tag{2}$$

where READOUT could be averaging or other graph-level pooling functions and $h_{\mathcal{G}}$ is the graph-level representation.

B. Model-Agnostic Meta-Learning

MAML [10] provides a feasible framework to train neural network with very few samples. The key of MAML is to cultivate the generalization knowledge of the meta-learner to find an optimal initialization for the model via two optimization loops: 1) *outer loop:* updates the neural network's initial parameters, also known as meta-initialization, for fast adaptation to new tasks and 2) *inner loop:* take the meta-initialization of the outer loop to perform gradient updates over each task separately for rapid adaptation.

Given a distribution $\mathcal{D}$ over tasks, B tasks from $\mathcal{D}$ are randomly sampled to form a mini-batch $\{\mathcal{T}_t\}_{t=1}^{B}$ for an episode, where each task $\mathcal{T}_t$ consists of a support set $\mathcal{S}_t$ and a query set $\mathcal{Q}_t$. Let f_θ denote the meta-learner with parameters θ and $\theta^t = \theta$, and the inner loop update on $\mathcal{T}_t$ can be presented as

$$\tilde{\theta}^t = \theta^t - \alpha \nabla_{\theta^t} \mathcal{L}_{\mathcal{S}_t}\left(f_{\theta^t}\right) \tag{3}$$

where $\mathcal{L}_{\mathcal{S}_t}$ is the loss function on the support set in the inner loop, and $\tilde{\theta}^t$ signifies θ after gradient updates for task $\mathcal{T}_t$. The updated parameters $\{\tilde{\theta}^t\}_{t=1}^{B}$ are then applied to the query set of

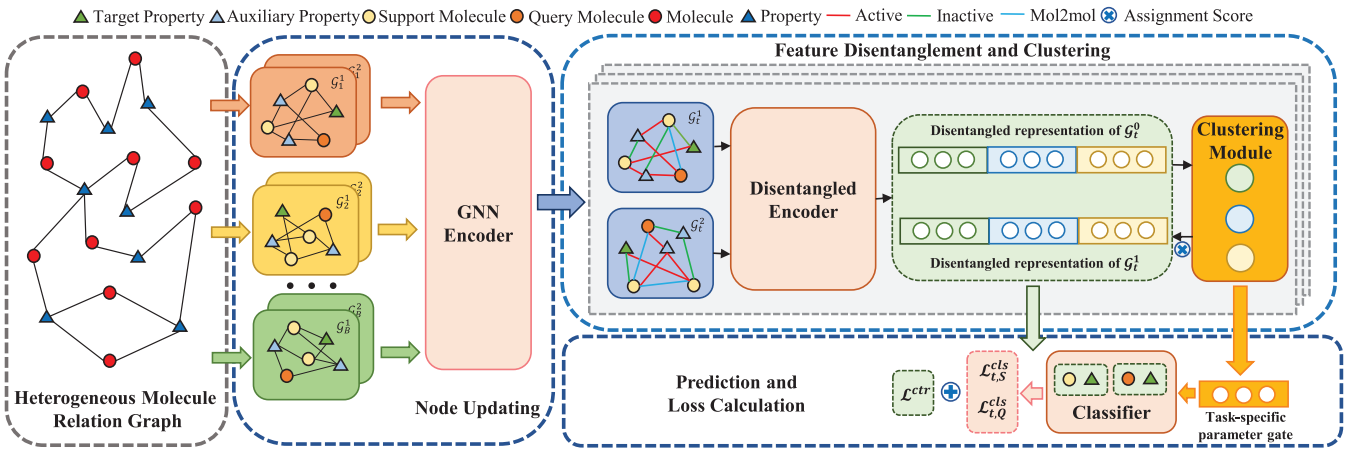


Fig. 2. Two-way one-shot overview of our proposed model. First, HMRG is constructed, and target tasks within subgraphs are randomly sampled to form a training batch with B episode pairs. Second, each subgraph is successively processed by the GNN encoder and disentangled encoder to achieve a graph representation containing K independent factors and contrastive loss L^{ctr} is output. Then, the soft clustering module is applied to compute the assignment score of each latent factor. Finally, the cluster-aware factorized representation is transformed into task-specific initialization for meta-classifier and output classification loss L^{cls}.

corresponding tasks. In this way, the outer loop loss function can be obtained, and the optimization of θ is defined as

$$\theta = \theta - \eta \nabla_\theta \sum_{t=1}^{B} \mathcal{L}_{\mathcal{Q}_t}(f_{\theta'}) \tag{4}$$

where $\mathcal{L}_{\mathcal{Q}_t}$ is the loss on the query set after inner updates.

C. Problem Definition

Following the previous work [10], task set in few-shot molecular property prediction is defined as $\{\mathcal{T}\}$, where each task $\mathcal{T}_t$ focuses on predicting a certain property p_t of molecules. Set of training tasks is denoted as $\mathcal{D}_{\text{train}} = \{(x_i, y_{i,t}) | t \in \mathcal{T}_{\text{train}}\}$, where x_i is a molecule and $y_{i,t}$ is the corresponding label of x_i on $\mathcal{T}_t$, and $\mathcal{T}_{\text{train}}$ is the task set during training step. Similarly, set of testing tasks is defined as $\mathcal{D}_{\text{test}} = \{(x_i, y_{i,t}) | t \in \mathcal{T}_{\text{test}}\}$. Specifically, training task set and testing task set are disjoint, namely, $\{\mathcal{T}_{\text{train}}\} \cap \{\mathcal{T}_{\text{test}}\} = \emptyset$, and $\{\mathcal{T}_{\text{train}}\} \cup \{\mathcal{T}_{\text{test}}\} = \{\mathcal{T}\}$. Besides, support set on $\mathcal{T}_t$ is defined as $\mathcal{S}_t = \{(x_i^s, y_{i,t}^s)\}|_{i=1}^{2N}$ that consists of N positive labels (i.e., $y = 1$) and N negative labels (i.e., $y = 0$), while query set on $\mathcal{T}_t$ is defined as $\mathcal{Q}_t = \{(x_i^q, y_{i,t}^q)\}|_{i=1}^{M}$ that consists of M molecules. In this way, a two-way N-shot episode is formulated. With the introduction of a meta-learning setting, the objective of our problem is to learn a meta-classifier on $\mathcal{D}_{\text{train}}$, which is capable of rapid transferring to novel properties with few samples on $\mathcal{D}_{\text{test}}$. In practice, batches of episodes $\{E_t\}_{t=1}^{B}$ are randomly sampled from $\{\mathcal{T}_{\text{train}}\}$, where $E_t = \mathcal{S}_t \cup \mathcal{Q}_t$; $\mathcal{S}_t$ participates in inner loop, while $\mathcal{Q}_t$ participates in outer loop.

IV. Method

A. Overview

As illustrated in Fig. 2, the overall architecture of our proposed framework consists of four parts. Given the many-to-many relationships between molecules and properties, we construct an HMRG, which encompasses molecules and properties as different types of nodes and their corresponding relations as edges. Then, the target task is randomly sampled from $\{\mathcal{T}_{\text{train}}\}$ as a subgraph; we utilize a GNN to encode the subgraph and simultaneously evaluate the similarity of molecules to establish the *mol2mol* relationships. Next, we introduce a disentangled encoder to factorize the subgraph representation in a self-supervised contrastive learning approach. The method aims to uncover independent factors that highlight the latent characteristics of the target task, and the contrastive loss is formulated as $\mathcal{L}^{\text{ctr}}$. Subsequently, the soft clustering module clusters the disentangled task representation, tailoring customization knowledge for task-specific initialization of the property classifier. Finally, pairs of molecule node and target property node are sent to classifier for molecular property prediction, and the classification loss can be calculated as $\mathcal{L}^{\text{cls}}$.

B. Heterogeneous Molecule Relation Graph

To capitalize the relation structure among properties and molecules, we construct an HMRG with explicit molecule–property relations and molecule–molecule relations, allowing us to integrate auxiliary properties of molecules for the molecular property prediction. Formally, HMRG is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{V}_M \cup \mathcal{V}_P$ indicates node set consisting of molecule node set $\mathcal{V}_M$ and property node set $\mathcal{V}_P$ from the entire dataset, and $\mathcal{E} = \mathcal{E}_{\text{MP}} \cup \mathcal{E}_{\text{MM}}$ indicates the edge set consisting of molecule–property relation set $\mathcal{E}_{\text{MP}}$ and molecule–molecule relation set $\mathcal{E}_{\text{MM}}$. Note that $\mathcal{V}$ and $\mathcal{E}_{\text{MP}}$ are provided by the dataset, while $\mathcal{E}_{\text{MM}}$ is updated during training. Following these definitions, we incorporate episodic meta-learning into HMRG. Given a specific property node $p_t \in \mathcal{V}_P$, we sample a subgraph of HMRG centered on p_t with $2N$ neighboring molecules. Hence, the support set $\mathcal{S}_t$ can be reformulated as

$$\mathcal{S}_t = \left\{\left(x_i, e_{i,t}, p_t\right) y_{i,t} | x_i \in \mathcal{N}(p_t)\right\}_{i=1}^{2N} \tag{5}$$

where $e_{i,t}$ denotes the edge between molecule x_i and property node p_t, and $\mathcal{N}(p_t)$ is the neighboring node set of p_t. Similarly,

the query set on HMRG can be defined as

$$\mathcal{Q}_t = \left\{\left(x_j, p_t\right), y_{j,t} | x_j \in \mathcal{N}\left(p_t\right)\right\}. \tag{6}$$

In this way, an episode of meta-learning E_t can be represented as $E_t = \mathcal{S}_t \cup \mathcal{Q}_t$. Since molecules possess multiple properties, other available properties can be utilized when predicting a novel property of the same molecule. In order to take advantage of auxiliary information that reflects the inherent properties of the molecule, N_a available property nodes are sampled and gathered to form an auxiliary subgraph given by $\mathcal{S}_t$ and $\mathcal{Q}_t$

$$\mathcal{G}_t^a = \left\{\left(x_i, e_{i,a}, p_a\right), y_{i,a} | p_a \in \mathcal{N}\left(x_i\right) \setminus p_t\right\}|_{a=1}^{N_a}. \tag{7}$$

Besides, considering that molecules with a certain degree of similarity tend to share common property characteristics, a novel edge type called *mol2mol* is introduced to illustrate intermolecular relationships within the sampled subgraph

$$\mathcal{G}_t^m = \left\{\left(x_i, e_{i,j}, x_j\right) | e_{i,j} \in \mathcal{E}_{\mathrm{MM}}\right\}. \tag{8}$$

By randomly selecting auxiliary properties and establishing the relationships between molecules (i.e., x_i and x_j) and these properties, along with the intermolecular relationships, the episode can be reformulated as

$$E_t \sim \mathcal{G}_t\left(\mathcal{V}_t, \mathcal{E}_t\right) = \mathcal{S}_t \cup \mathcal{Q}_t \cup \mathcal{G}_t^a \cup \mathcal{G}_t^m. \tag{9}$$

Based on $\mathcal{G}_t$, we apply a graph-based encoder f_{mol} [53] and a task embedding layer f_{pro} to encode molecule $x_i \in \mathcal{V}_M$ and property $p_i \in \mathcal{V}_P$, respectively,

$$h_{x_i} = f_{\mathrm{mol}}\left(x_i\right) h_{p_t} = f_{\mathrm{pro}}\left(p_t\right) \tag{10}$$

where $h_{x_i}, h_{p_t} \in \mathbb{R}^d$ is embedded as d-dimensional representations in the same space. And the initialized representation of node i, namely, $h_i^{(0)}$ in $\mathcal{G}_t$, can be represented as

$$h_i^{(0)} = \begin{cases} h_{x_i}, & for\ i \in \mathcal{V}_M \\ h_{p_i}, & for\ i \in \mathcal{V}_P. \end{cases} \tag{11}$$

Given $\mathcal{E} = \mathcal{E}_{\mathrm{MP}} \cup \mathcal{E}_{\mathrm{MM}}$, there are three kinds of edge type in $\mathcal{G}_t$: *active* and *inactive* for $e_{i,j} \in \mathcal{E}_{\mathrm{MP}}$, and *mol2mol* for $e_{i,j} \in \mathcal{E}_{\mathrm{MM}}$. Hence, an edge embedding layer is applied to obtain the initialized representation $h_{i,j}^{(0)}$ of $e_{i,j}$

$$h_{i,j}^{(0)} = f_{\mathrm{edge}}\left(e_{i,j}\right). \tag{12}$$

After the embedding initialization overall subgraph $\mathcal{G}_t$, a message-passing mechanism is adopted to iteratively update the node feature with aggregated neighbor information as follows:

$$h_i^{(l)} = \mathrm{GNN}^{(l)}\left(h_i^{(l-1)}, h_j^{(l-1)}, h_{i,j}^{(l-1)}, \omega_{i,j}^{(l-1)} | j \in \mathcal{N}\left(i\right)\right) \tag{13}$$

where $h_i^{(l)} \in \mathbb{R}^d$ is the embedding of node i at lth layer, $h_{i,j}^{(l)} \in \mathbb{R}^d$ denotes the embedding of edge between nodes i and j at (l)th layer according to the node type, and $\omega_{i,j}^{(l)} \in \mathbb{R}$ denotes the corresponding edge weight. Then, a relation estimator is utilized to predict the correlation weight between molecules, namely, $e_{i,j} \in \mathcal{E}_{\mathrm{MM}}$, defined as

$$\alpha_{i,j}^{(l)} = \sigma\left(\mathrm{MLP}\left(\exp\left(-\left|h_i^{(l-1)} - h_j^{(l-1)}\right|\right)\right)\right) \tag{14}$$

where $\sigma(\cdot)$ is the sigmoid function, and MLP denotes the multilayer perception. We select the top-n predicted $\alpha_{i,j}^{(l)}$ and keep the corresponding molecule pairs as edges in *mol2mol* type. Accordingly, the weight of edges $\omega_{i,j}^{(l)}$ can be represented as

$$\omega_{i,j}^{(l)} = \begin{cases} \alpha_{i,j}^{(l)}, & for\ e_{i,j} \in \mathcal{E}_{\mathrm{MM}} \\ 1, & \text{otherwise} \end{cases}. \tag{15}$$

With the combination of the above steps, the final node feature $h_i^{(L)}$ is obtained after L iterations and forwarded to the disentangled encoder for further process.

C. Disentangled Graph Encoder

As previous works [7], [15], [16] assume that the transferable knowledge is shared across all tasks, they tend to overlook the heterogeneity of different sampled few-shot molecular property prediction tasks from $\mathcal{T}$. In fact, a meta-learner should balance the generalization of knowledge across tasks and the heterogeneity inherent to each task, which is crucial for effective and rapid knowledge transfer. Therefore, we are motivated to design a disentangled graph encoder to uncover the various underlying explanatory factors within the task, create a factorized representation, and ensure that different factorized representations are effectively tailored to the corresponding transferable knowledge across tasks.

To achieve this, we suppose that each subgraph can be factorized into K independent aspects and presented as a combination of related latent factors after processing by a multichannel graph disentanglement layer. For each channel k, a $\mathrm{GNN}_k = \{\mathrm{GNN}_k^{(m)}\}_{m=1}^M$ encoder with M disentangled layers is utilized to gather factorwise information as follows:

$$h_{i,k}^{(L+M)} = \mathrm{GNN}_k\left(h_i^{(L)}, h_j^{(L)}, h_{i,j}^{(L)}, \omega_{i,j}^{(L)} | j \in \mathcal{N}\left(i\right)\right) \tag{16}$$

where $h_{i,k}^{(L+M)} \in \mathbb{R}^{(d/K)}$ is the node embedding that is only pertinent to the kth latent factor. And each channel outputs the factorized node representation with a separate MLP

$$z_{i,k} = \mathrm{MLP}_k\left(h_{i,k}^{(L+M)}\right) \tag{17}$$

where $z_{i,k} \in \mathbb{R}^{(d/K)}$ is the representation of node i at kth channel. Compared to existing graph encoders, which generally take a holistic approach, our disentangled graph encoder enables the identification and isolation of heterogeneous aspects of the subgraph, facilitating the localization of the cluster to which the task belongs.

Furthermore, we consider that subgraphs sampled from the same task (i.e., $\mathcal{G}_t^1$ and $\mathcal{G}_t^2$) should be consistent with each other and semantically pulled close in corresponding factorwise representation, while subgraphs sampled from different tasks (i.e., $\mathcal{G}_t^1$ and $\mathcal{G}_{t'}^2$) are regarded as separate episodes from different tasks and should be pushed far away. Toward this end, we reformulate an episode E_t consisting of a pair of subgraph $(\mathcal{G}_t^1, \mathcal{G}_t^2)$ and adopt a self-supervised objective to train a disentangled encoder in a contrastive learning manner. Given the final representation of node i at kth channel $z_{i,k}$, we first compute the graph representation as follows:

$$g_{t,k} = \mathrm{READOUT}_k\left(\left\{z_{i,k}\right\}_{i=1}^{|\mathcal{V}_t|}\right) = \frac{1}{|\mathcal{V}_t|}\sum_{i=1}^{|\mathcal{V}_t|} z_{i,k} \tag{18}$$

where $g_{t,k} \in \mathbb{R}^{(d/K)}$ is the kth channel representation of subgraph centered on target property t. Then, we define K learnable prototype vectors $\{u_k\}_{k=1}^{K}$ corresponding to latent factors, and the probability of the kth latent factor reflected in subgraph $\mathcal{G}_t$ can be

$$p_\theta\left(k|\mathcal{G}_t\right)) = \frac{\exp\left(\cos\left(g_{t,k}, u_k\right)\right)}{\sum_{k=1}^{K}\left(\exp\left(\cos\left(g_{t,k}, u_k\right)\right)\right)} \tag{19}$$

where $\cos(\cdot,\cdot)$ is the cosine similarity. Meanwhile, the probability of kth latent factor pertinent to the subgraphs sampled from the same task can be expressed as

$$p_\theta\left(t|\mathcal{G}_t, k\right) = \frac{\exp\left(\cos\left(g_{t,k}^1, g_{t,k}^2\right)\right)}{\sum_{t'=1}^{T}\left(\exp\left(\cos\left(g_{t,k}^1, g_{t',k}^2\right)\right)\right)} \tag{20}$$

where $g_{t,k}^1$ and $g_{t,k}^1$ are the disentangled graph representations of the pair of subgraphs sampled from the same task t. Hence, factorwise contrastive learning is conducted by instance discrimination of the same task, defined as

$$p_\theta\left(t|\mathcal{G}_t\right)) = \mathbb{E}_{p_\theta(k|\mathcal{G}_t)}\left[p_\theta\left(t|\mathcal{G}_t, k\right)\right]. \tag{21}$$

Then, the learning objective is to maximize the joint probability $\prod_{t=1}^{T} p_\theta(t|\mathcal{G}_t)$ over the dataset $\{\mathcal{G}_t\}_{t=1}^{T}$. However, the objective is difficult to maximize due to the latent factor. We reformulate it with an expectation–maximization (EM) algorithm by calculating the posterior distribution

$$p_\theta\left(k|\mathcal{G}_t, t\right) = \frac{p_\theta\left(k \mid \mathcal{G}_t\right) p_\theta\left(t|\mathcal{G}_t, k\right)}{\sum_{k=1}^{K} p_\theta\left(k|\mathcal{G}_t\right) p_\theta\left(t|\mathcal{G}_t, k\right)}. \tag{22}$$

Nevertheless, the computation of $p_\theta(k|\mathcal{G}_t, t)$ is subjected to the summation over the dataset with the term $\sum_{t'=1}^{T}(\exp(\cos(g_{t,k}^1, g_{t',k}^2)))$, which results in high computation cost. Instead, we resort to maximizing the evidence lower bound (ELBO) of $\log p_\theta(t|\mathcal{G}_t)$, given by

$$\begin{aligned}\log p_\theta\left(t|\mathcal{G}_t\right) \geq\ & \mathbb{E}_{q_\theta(k|\mathcal{G}_t,t)}\left[\log p_\theta\left(t|\mathcal{G}_t, k\right)\right] \\ & - D_{\mathrm{KL}}\left(q_\theta\left(k|\mathcal{G}_t, t\right) \parallel p_\theta\left(k|\mathcal{G}_t\right)\right)\end{aligned} \tag{23}$$

where $q_\theta(k|\mathcal{G}_t, t)$ is a variational distribution to approximate posterior distribution $p_\theta(k|\mathcal{G}_t, t)$ and is calculated by

$$q_\theta\left(k|\mathcal{G}_t, t\right) = \frac{p_\theta\left(k \mid \mathcal{G}_t\right)) \hat{p}_\theta\left(t \mid \mathcal{G}_t, k\right)}{\sum_{k=1}^{K} p_\theta\left(k|\mathcal{G}_t\right)) \hat{p}_\theta\left(t \mid \mathcal{G}_t, k\right)} \tag{24}$$

where $\hat{p}_\theta(t|\mathcal{G}_t, k)$ is the instance discrimination under latent factor calculated on a mini-batch of size B

$$\hat{p}_\theta\left(t|\mathcal{G}_t, k\right) = \frac{\exp\left(\cos\left(g_{t,k}^1, g_{t,k}^2\right)\right)}{\sum_{t'=1}^{B}\exp\left(\cos\left(g_{t,k}^1, g_{t',k}^2\right)\right)}. \tag{25}$$

As a result, we can reformulate the factorwise contrastive learning loss $\mathcal{L}^{\mathrm{ctr}}$ over a mini-batch as

$$\begin{aligned}\mathcal{L}^{\mathrm{ctr}} =\ & \sum_{t=1}^{B} \mathbb{E}_{q_\theta(k|\mathcal{G}_t,t)}\left[\log p_\theta\left(t|\mathcal{G}_t, k\right)\right] \\ & - D_{\mathrm{KL}}\left(q_\theta\left(k|\mathcal{G}_t, t\right) \parallel p_\theta\left(k|\mathcal{G}_t\right)\right)\end{aligned} \tag{26}$$

where $\mathcal{L}^{\mathrm{ctr}}$ enforces the independence of learned graph representation $\{g_{t,k}\}_{k=1}^{K}$ and participates in the outer loop during the meta-learning process.

D. Clustering-Based Initialization

The uncertainty and heterogeneity of different tasks in meta-learning restrict the effectiveness of universal knowledge transfer between tasks [56], making it challenging to achieve suitable initializations for new tasks. Thus, given the disentangled graph representation of the task, we propose a soft clustering module to locate the cluster to which the task belongs and optimize the initial parameters of latent factors, which serve as the customization knowledge of different clusters of tasks. In particular, we evaluate the assignment score of each task $s_{t,k} \in \mathbb{R}$ to cluster and group tasks into separate clusters. Given the extracted graph representation of the task, we regard latent factor prototype vectors $\{u_k\}_{k=1}^{K}$ as K cluster centers. Therefore, the assignment score of $g_{t,k}$ in cluster k can be parameterized by (19), namely, $s_{t,k} = p_\theta(k|\mathcal{G}_t))$.

As a result of the assignment, the weighted graph representation at kth factors $w_{t,k} \in \mathbb{R}^{(d/K)}$ is, then, given by

$$w_{t,k} = s_{t,k} \tanh\left(W_s g_{t,k} + b_s\right) \tag{27}$$

where W_s and b_s are the learnable parameters shared among factors. Subsequently, $\{g_{t,k}\}_{k=1}^{K}$ and $\{w_{t,k}\}_{k=1}^{K}$ are, respectively, concatenated at the factor level and packed up together to attain the task-specific parameter gate o_t

$$o_t = \sigma\left(W_g\left(g_t \oplus w_t\right) + b_g\right) \tag{28}$$

where $\oplus$ is a concatenation operator that not only preserves but also enhances the cluster-specific properties of the parameter gate. W_g and b_g are the learnable parameters.

With these efforts, the final graph representation of the task is believed to be cluster-specific, where a similar task achieves a similar graph representation while different tasks trigger disparate ones. We condense the representation into the parameter gate as customization knowledge. Then, the initial parameter θ_{cls} of meta-classifier is adapted to task-specific initialization θ_{cls}^t with parameter gate: $\theta_{\mathrm{cls}}^t = \theta_{\mathrm{cls}} \circ o_t$, where $\circ$ is the elementwise product. The optimization of the classifier during the inner loop is, then, reformulated as follows:

$$\tilde{\theta}_{\mathrm{cls}}^t = \theta_{\mathrm{cls}}^t - \lambda_{\mathrm{inner}} \nabla_{\theta_{\mathrm{cls}}} \mathcal{L}_{t,\mathcal{S}}^{\mathrm{cls}}\left(\theta_{\mathrm{cls}}\right) \tag{29}$$

where λ_{inner} is the inner loop learning rate, and $\mathcal{L}_{t,\mathcal{S}}^{\mathrm{cls}}$ is the classification loss on support set.

E. Training and Testing

In this section, we discuss the optimization strategy in Meta-DREAM. We first demonstrate the classification loss function $\mathcal{L}^{\mathrm{cls}}$, and then illustrate the process of inner loop and outer loop in the training and testing stages.

Following the soft clustering module, the disentangled node embedding $z_{i,k}$ of $\mathcal{G}_t$ is weighted by assignment score and concatenated at factor level by: $z_i = \mathrm{CONCAT}(\{s_{t,k} \cdot z_{i,k}\}_{k=1}^{K})$. Then, the embedding of molecule node z_i and the target property node z_t are concatenated to predict the label

$$\hat{y}_{i,t} = \sigma\left(f_{\mathrm{cls}}\left(z_i \oplus z_t\right)\right) \tag{30}$$

where $\hat{y}_{i,t}$ is the predicted label of molecule node i in terms of target property t, and f_{cls} is the classifier parameterized by

Algorithm 1 Meta-Training Algorithm of Meta-DREAM

Input: Heterogeneous Molecule Relation Graph (HMRG)
Output: Parameter Set Θ

while not done **do**
  Sample B episode pairs $\{(\mathcal{G}_t^1, \mathcal{G}_t^2)\}_{t=1}^B$ from HMRG where $t \in \mathcal{T}_{train}$;
  for $t = 1, 2, \ldots, B$ **do**
    Compute node embedding h_i^L in $\mathcal{G}_t^1, \mathcal{G}_t^2$ respectively;
    Compute disentangled node embedding $z_{i,k}$ and disentangled graph embedding $g_{i,k}$ on both $\mathcal{G}_t^1, \mathcal{G}_t^2$;
    Compute task assignment score $s_{t,k}$ by Eqn.(19);
    Compute parameter gate o_t by Eqn.(28);
    Calculate classification loss on support set $\mathcal{L}_{t,\mathcal{S}}^{\text{cls}}$ by Eqn.(31) on both $\mathcal{G}_t^1, \mathcal{G}_t^2$;
    Do inner loop update for classifier parameter by Eqn.(29) and rest of parameters by Eqn.(32);
    Calculate classification loss on query set $\mathcal{L}_{t,\mathcal{Q}}^{\text{cls}}$ on both $\mathcal{G}_t^1$ and $\mathcal{G}_t^2$;
  end for
  Compute contrastive loss $\mathcal{L}^{ctr}$ by Eqn.(26);
  Do outer loop parameter update by Eqn.(34);
end while

θ_{cls}. The classification loss of $\mathcal{G}_t$ in inner loop, which focuses on support set $\mathcal{S}_t$, is given by

$$\mathcal{L}_{t,\mathcal{S}}^{\text{cls}} = -\sum_{\mathcal{S}_t} \left(y \log \hat{y} + (1-y) \log (1-\hat{y}) \right) \tag{31}$$

where y and $\hat{y}$ are the ground truth and prediction of labels. Similarly, classification loss in the outer loop, which focuses on the query set $\mathcal{Q}_t$, can be calculated as $\mathcal{L}_{t,\mathcal{Q}}^{\text{cls}}$.

Aside from θ_{cls}, which is updated by (29), the rest of the parameters are updated in the inner loop optimization by gradient descent as follows:

$$\theta \leftarrow \theta - \lambda_{\text{inner}} \nabla_\theta \mathcal{L}_{t,\mathcal{S}}^{\text{cls}} \tag{32}$$

where θ denotes the parameter set, including molecule and property embedding layer, GNN layers, relation estimator, disentangled graph encoder, and soft clustering module. After the parameter update in the inner loop, the classification loss on the query set $\mathcal{L}_{t,\mathcal{Q}}^{\text{cls}}$ is computed and packed up with contrastive loss $\mathcal{L}^{\text{ctr}}$ in the outer loop as follows:

$$\mathcal{L} = \frac{1}{2B} \sum_{t=1}^{B} \left(\mathcal{L}_{t,\mathcal{Q}}^{\text{cls}\ 1} + \mathcal{L}_{t,\mathcal{Q}}^{\text{cls}\ 2} \right) + \alpha \mathcal{L}^{\text{ctr}} \tag{33}$$

where B is the batch size of episodes, $\mathcal{L}_{t,\mathcal{Q}}^{\text{cls}\ 1}$ denotes the classification loss of $\mathcal{G}_t^1$ on query set, $\mathcal{L}^{\text{ctr}}$ is defined by (26), and α is a hyperparameter to balance the influence of contrastive loss. The outer loop is, then, conducted to optimize the whole meta-learner with learning rate λ_{outer}

$$\Theta \leftarrow \Theta - \lambda_{\text{outer}} \nabla_\Theta \mathcal{L} \tag{34}$$

where $\Theta = \theta \cup \theta_{\text{cls}}$ for simplicity. The overall procedure of our meta-learning framework is described in Algorithm 1.

In the testing stage, a novel task is sampled from $\mathcal{T}_{test}$ with auxiliary properties from $\mathcal{T}_{\text{train}}$ and molecules from $\mathcal{D}_{test}$. After identical subgraph construction and feature extraction process, classification loss $\mathcal{L}_{t,\mathcal{S}}^{\text{cls}}$ is obtained and Θ is only fine-tuned by (29) and (32).

F. Complexity Analysis

Given HRMG with $|\mathcal{T}|$ total tasks, we sample the subgraph $\mathcal{G}_t$ with batch size B. The average number of nodes and edges in $\mathcal{G}_t$ are $|\mathcal{V}_t|$ and $|\mathcal{E}_t|$, respectively, the number of support samples per task is N, the dimension of representation is d, and the number of GNN layer and factor is L and K, respectively. The computational consumption is mainly composed of two parts: 1) base learner training phase and 2) meta-training phase. For 1), the complexity of the GNN-based encoder is $O(|\mathcal{E}_t|Ld)$, the complexity of the disentangled graph encoder is $O(|\mathcal{V}_t|d)$, the complexity of the clustering module is $O(d)$, and the complexity of the binary classifier is $O(\text{Nd})$. To sum up, we have the complexity of 1): $O(|\mathcal{T}|d(|\mathcal{E}_t|L + |\mathcal{V}_t| + N + 1))$. For 2), the complexity consists of the contrastive loss $\mathcal{L}^{\text{ctr}}$ and the optimization of parameters. Since the latter is independent of the dataset scale, we focus only on the complexity of the contrastive loss, which is $O(BNd)$. In total, the overall complexity of our model is $O(|\mathcal{T}|d(|\mathcal{E}_t|L + |\mathcal{V}_t| + N + 1) + BNd)$. This complexity is linearly related to the product of sample and task numbers ($|\mathcal{T}| \ll |\mathcal{V}_t|$), which aligns with other meta-learning methods for few-shot molecular property prediction.

V. Experiment

In this section, we conduct comprehensive experiments on five commonly used benchmark datasets of molecular property prediction to illustrate the following research problems.

1) *RQ1:* How does our proposed Meta-DREAM perform over other baselines for few-shot molecular prediction?
2) *RQ2:* How does each key component of Meta-DREAM influence the performance of the model?
3) *RQ3:* How do different hyperparameters in Meta-DREAM influence the performance of the model?
4) *RQ4:* Can we intuitively visualize the effectiveness of the disentangled encoder and the soft-clustering module?

A. Experimental Setup

1) Datasets: The experiments of our Meta-DREAM are conducted on five few-shot molecular benchmarks datasets from MoleculeNet [57] following the data splits in previous work [15]. Table I presents the details of these datasets.

1) *Tox21* [58] contains 8014 compounds and their interactions with 12 biological properties, including stress response mechanisms and nuclear receptors.
2) *SIDER* [59] contains the relationship between 1427 approved drugs and 27 categories of side effects.
3) *MUV* [60] provides 17 tasks with 93 127 compounds for the validation of virtual screening.
4) *ToxCast* [61] contains 8615 compounds with 617 toxicity labels based on high-throughput screening.
5) *PCBA* [62] contains bio-activity profiles of 437 929 small molecules generated by 128 high-throughput screening.
6) *ENZYMES* [63] is a macromolecule dataset, where each graph represents the structure of an enzyme.

2) Baselines: We compare our model against various approaches for few-shot molecular property prediction. The detailed information of baselines is listed as follows.

1) *Siamese* [64] identifies whether input molecule pairs belong to the same category for the prediction of query properties by employing dual convolutional neural networks.
2) *ProtoNet* [24] introduces class prototypes and learns to compare the distance between each input and prototype in a metric-based manner.
3) *MAML* [10] applies two-step gradient descent to learn a good initialization of model parameters, allowing the model to quickly adjust to novel tasks.
4) *TPN* [28] constructs a relation graph based on the similarity of the inputs and performs label propagation across the graph for prediction.
5) *EGNN* [65] also constructs a relation graph from input samples based on the similarity and learns to predict edge labels for relational graphs.
6) *IterRefLSTM* [8] adopts a variant of MatchingNet [9] and combines residual LSTM embedding with GNNs for molecular property prediction.
7) *PAR* [16] refines the representations of input data through class prototypes and implements a label propagation strategy for closely related inputs within a relational graph.
8) *GS-Meta* [15] constructs a graph of molecule property relations and reformulates the learning process of few-shot molecule property prediction via an episode scheduler.
9) *PACIA* [51] adjusts the encoder at the task level and the predictor at the query level, utilizing a hierarchical adaptation mechanism.
10) *APN* [66] uses human-defined attributes to guide the excavation of generalization knowledge within the molecular graph for the model.

3) Evaluation Metrics: Following the previous works [13], [15], we adopt ROC-AUC to measure the prediction performance of molecular properties in the query set of the meta-testing task. Across ten experimental runs for each dataset, we documented the mean and standard deviation of the ROC-AUC.

4) Implementation: We implement our model based on PyTorch and PyTorch Geometric. Specifically, we use a five-layer graph isomorphism network (GIN) to encode $\mathcal{G}_t$ with the hidden size $d = 300$. We use a two-layer disentangled encoder and a two-layer MLP for classification. We fix the influence factor α of contrastive loss as 0.02, and the learning rates of inner loop and outer loop, namely, λ_{inner} and λ_{outer}, are set as 0.5 and 0.001, respectively. Given the varied dataset sizes and unknown optimal latent factors, we tuned the number of latent factors K within the range [1,2,3,4,5,6,10].

B. Performance Comparison (RQ1)

Table II summarizes the performance of our Meta-DREAM against all baselines. Our main conclusions are as follows.

TABLE I
STATISTICS OF THE DATASETS USED IN THE EXPERIMENTS

Dataset	Tasks	Support Tasks	Query Tasks	Compounds
Tox21	12	9	3	8014
SIDER	27	21	6	1427
MUV	17	12	5	93127
ToxCast	617	451	178	8615
PCBA	128	118	10	437929
ENZYMES	6	4	2	600

1) Compared to traditional methods (i.e., Siamese, ProtoNet, and MAML), graph-based methods (i.e., TPN, EGNN, IterRefLSTM, PAR, GS-Meta, and Meta-DREAM) generally achieve better performance. For example, Meta-DREAM obtains 7.9% improvement against MAML on the ten-shot scenario on Tox21. This indicates that utilizing the topological structure significantly enhances the quality of molecular representations.
2) Our Meta-DREAM consistently outperforms other baselines over all six datasets. Specifically, our method achieves the average increment ratio of 0.90% and 1.94% compared to the runner-up baseline GS-Meta in both ten-shot and one-shot scenarios, respectively. This demonstrates that introducing the disentangled encoder and the clustering-based initialization of specific tasks can effectively improve the model performance. Based on the promising performance, we also include a discussion on the potential applications in drug discovery.
3) Notice that the improvement varies across different datasets. For example, our model reaches a growth at 1.29% on ToxCast but 0.12% on MUV. The reason for the discrepancy in the improvement rate can be mainly reduced to the ratio of missing labels in different datasets. Specifically, there are 84.2% missing labels on MUV, which prevents the full utilization of auxiliary properties.

C. Ablation Study (RQ2)

To illustrate the effect of key components introduced in our model, we conduct ablation studies from two aspects.

1) For the disentangled graph representation learning process, we consider the following variants: *w/o CTR*: remove contrastive loss L^{ctr}; *w/o m2m*: remove *mol2mol* edges in HMRG; and *w/o E*: remove edge type in message passing.
2) For task-specific initialization, we consider the following variants without soft clustering module: *Variant 1*: it sets a random distribution of s_t, k and *Variant 2*: it sets a uniform distribution of s_t, k. The results are shown in Table III.

1) Effect of Disentangled Graph Learning: Our HMRG construction and disentangled graph encoder module contribute to achieving better performance. Among the three variants, removing *mol2mol* edges leads to an apparent decline of the performance, demonstrating that intermolecular edges calculated by similarity effectively facilitate the usage of auxiliary information. Meanwhile, considering the edge type during

TABLE II

ROC-AUC SCORES ON BENCHMARK DATASETS. THE BEST IS MARKED WITH BOLDFACE, AND THE SECOND BEST IS MARKED WITH UNDERLINE

Method	Tox21		SIDER		MUV		ToxCast		PCBA		ENZYMES	
	10-shot	1-shot	10-shot	1-shot	10-shot	1-shot	10-shot	1-shot	10-shot	1-shot	10-shot	1-shot
Siamese	$80.40_{(0.35)}$	$65.00_{(1.58)}$	$71.10_{(4.32)}$	$51.43_{(3.31)}$	$59.96_{(5.13)}$	$50.00_{(0.17)}$	-	-	-	-	-	-
ProtoNet	$74.98_{(0.32)}$	$65.58_{(1.72)}$	$64.54_{(0.89)}$	$57.50_{(2.34)}$	$65.88_{(4.11)}$	$58.31_{(3.18)}$	$68.87_{(0.43)}$	$58.55_{(0.52)}$	$64.93_{(1.94)}$	$55.79_{(1.45)}$	-	-
MAML	$80.21_{(0.24)}$	$75.74_{(0.48)}$	$70.43_{(0.76)}$	$67.81_{(1.12)}$	$63.90_{(2.28)}$	$60.51_{(3.12)}$	$68.30_{(0.59)}$	$61.12_{(0.94)}$	$66.22_{(1.31)}$	$62.04_{(1.73)}$	-	-
TPN	$76.05_{(0.24)}$	$60.16_{(1.18)}$	$67.84_{(0.95)}$	$62.90_{(1.38)}$	$65.22_{(5.82)}$	$50.00_{(0.51)}$	-	-	-	-	$60.82_{(0.95)}$	$58.40_{(1.75)}$
EGNN	$81.21_{(0.16)}$	$79.44_{(0.22)}$	$72.87_{(0.73)}$	$70.79_{(0.95)}$	$65.20_{(2.08)}$	$62.18_{(1.76)}$	$74.02_{(1.11)}$	$64.17_{(0.89)}$	$69.92_{(1.85)}$	$62.14_{(1.58)}$	$65.84_{(0.73)}$	$61.07_{(1.08)}$
IterRefLSTM	$81.10_{(0.17)}$	$80.97_{(0.10)}$	$69.63_{(0.31)}$	$71.73_{(0.14)}$	$49.56_{(5.12)}$	$48.54_{(3.12)}$	-	-	-	-	$62.37_{(1.81)}$	$59.62_{(1.06)}$
PAR	$82.06_{(0.12)}$	$80.46_{(0.13)}$	$74.68_{(0.31)}$	$71.87_{(0.48)}$	$66.48_{(2.12)}$	$64.12_{(1.18)}$	$74.78_{(1.53)}$	$69.45_{(1.24)}$	$70.05_{(0.94)}$	$67.77_{(1.04)}$	$68.56_{(1.26)}$	$64.90_{(2.37)}$
GS-Meta	$\underline{85.85}_{(0.26)}$	$\underline{84.32}_{(0.89)}$	$\underline{83.72}_{(0.54)}$	$\underline{82.84}_{(0.67)}$	$\underline{67.11}_{(1.95)}$	$\underline{64.70}_{(2.88)}$	$\underline{81.55}_{(0.19)}$	$\underline{80.03}_{(0.26)}$	$\underline{72.16}_{(0.71)}$	$\underline{70.03}_{(1.56)}$	$\underline{70.92}_{(0.65)}$	$\underline{67.69}_{(0.95)}$
PACIA	$84.25_{(0.31)}$	$82.77_{(0.15)}$	$82.40_{(0.26)}$	$77.72_{(0.34)}$	$66.80_{(2.65)}$	$64.91_{(3.18)}$	$72.38_{(0.96)}$	$69.89_{(1.17)}$	$69.73_{(0.67)}$	$67.50_{(1.29)}$	$70.57_{(0.82)}$	$67.36_{(0.74)}$
APN	$84.54_{(0.36)}$	$80.40_{(0.23)}$	$79.02_{(0.72)}$	$75.07_{(0.38)}$	$65.41_{(1.72)}$	$62.67_{(2.35)}$	$76.18_{(1.62)}$	$73.25_{(2.45)}$	$70.09_{(1.32)}$	$66.84_{(1.74)}$	$69.53_{(1.24)}$	$64.67_{(1.59)}$
Meta-DREAM	$\mathbf{86.56}_{(0.29)}$	$\mathbf{85.61}_{(0.06)}$	$\mathbf{84.52}_{(0.18)}$	$\mathbf{83.88}_{(0.24)}$	$\mathbf{67.19}_{(1.89)}$	$\mathbf{65.83}_{(1.20)}$	$\mathbf{82.60}_{(0.21)}$	$\mathbf{81.30}_{(0.11)}$	$\mathbf{73.11}_{(0.49)}$	$\mathbf{72.54}_{(0.71)}$	$\mathbf{72.08}_{(0.76)}$	$\mathbf{68.74}_{(1.32)}$

TABLE III

ABLATION STUDY ON DISENTANGLED GRAPH LEARNING AND TASK-SPECIFIC INITIALIZATION

Method	SIDER		Tox21	
	10-shot	1-shot	10-shot	1-shot
w/o CTR	84.40(↓ 0.12)	83.68(↓ 0.20)	86.16(↓ 0.40)	85.46(↓ 0.15)
w/o m2m	84.29(↓ 0.23)	83.32(↓ 0.56)	86.36(↓ 0.20)	84.83(↓ 0.78)
w/o E	84.35(↓ 0.17)	83.64(↓ 0.24)	86.23(↓ 0.33)	85.27(↓ 0.34)
Variant 1	83.02(↓ 1.50)	82.23(↓ 1.65)	84.70(↓ 1.86)	83.33(↓ 2.28)
Variant 2	84.20(↓ 0.32)	83.37(↓ 0.51)	86.17(↓ 0.39)	85.24(↓ 0.37)
Ours	**84.52**	**83.88**	**86.56**	**85.61**

message passing improves model performance to specifically capture interactions and dependencies between nodes. Removing contrastive loss also leads to a model performance decrease, emphasizing its crucial role in effectively improving disentangled graph representation learning.

2) Effect of Task-Specific Initialization: There is also an apparent performance drop in Variant 1, indicating that the random strategy struggles to identify the appropriate cluster of subgraphs and generate a proper initialization. Similarly, Variant 2 adopts a uniform strategy and distributes tasks equally across all clusters. The strategy overlooks the distinct characteristics of each task, resulting in a negative impact on model performance. In a nutshell, these variants positively demonstrate that the soft clustering module can effectively contribute to the formation of task-specific initialization and guide the meta-training process.

D. Parameter Sensitivity (RQ3)

We also examine the sensitivity of the proposed Meta-DREAM to various hyperparameters. Specifically, we investigate the effect of varying numbers of latent factors, disentangled layers for the disentangled graph encoder, and auxiliary properties for the training and testing stages.

1) Effect of the Number of Latent Factors: To analyze whether Meta-DREAM can benefit from factor disentanglement, we study the performance of the model with varying numbers of latent factors. In particular, we search the number of factor graphs K in the range of {1,2,3,4,5,6,10}. Fig. 3(a) shows the performance with respect to different numbers of latent factors for the graph disentangled encoder. We find the following.

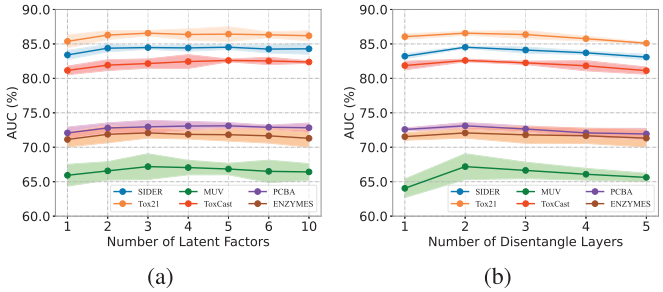


Fig. 3. Performance comparison with respect to different numbers of (a) latent factors and (b) disentangle layers.

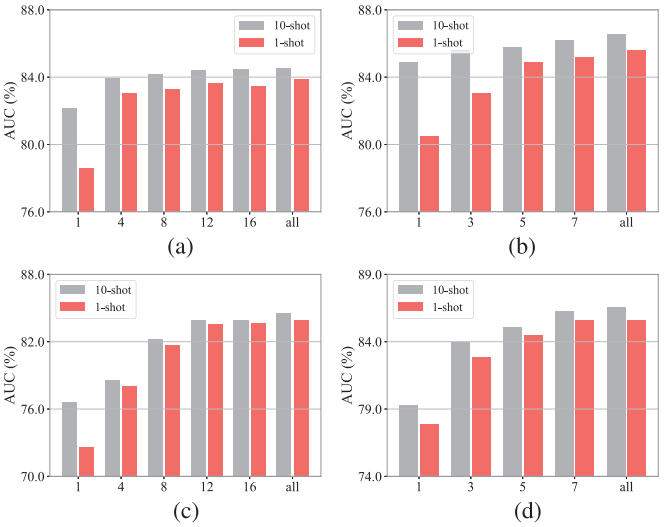
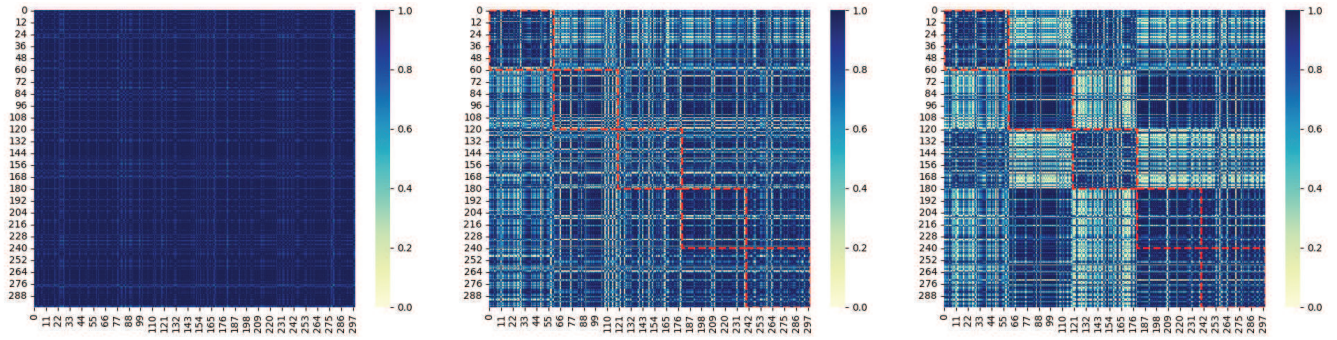


Fig. 4. Performance comparison with respect to different numbers of sampled auxiliary properties in the training (top) and testing (bottom) stages. (a) Training stage of SIDER. (b) Training stage of Tox21. (c) Testing stage of SIDER. (d) Testing stage of Tox21.

1) When K=1, the encoder can be degraded into an entangled representation module with poor performance. Increasing K can substantially enhance the model performance. This observation underscores that feature disentanglement contributes to excavating heterogeneous factors and attaining better task-specific representations.
2) The optimal value of K varies across datasets based on their sizes. For example, in Tox21, which contains 12 tasks, the best performance is achieved with K = 3. In



(a) (b) (c)

Fig. 5. Analysis of feature correlation on graphs with five latent factors. We calculate the absolute values of the correlation between elements of the same 300-D graph representation with five latent factors. Compared to GS-Meta and w/o CTR, the correlation of Meta-DREAM shows distinct factor-aware blocks on the diagonal, indicating the disentangled graph encoder factorizes the graph representation into five independent aspects. (a) GS-Meta. (b) w/o CTR. (c) Ours.

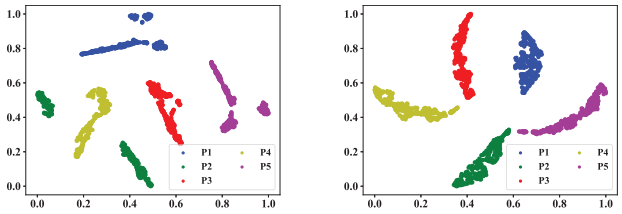
contrast, SIDER, with 27 tasks, shows optimal performance at $K = 5$. Despite this, our model appears to be less sensitive to K once its value reaches a certain extent.

2) Effect of the Number of Disentangled Layers: By stacking various numbers of disentangled layers, we further explore the optimal composition of the disentangled graph encoder. Specifically, we vary the number of disentangled layers M in the range of {1,2,3,4,5}. Fig. 3(b) shows the performance with respect to different numbers of disentangled layers for message passing. We have the following observations.

1) When $M = 1$, the encoder struggles to aggregate enough local information for the factor disentanglement, leading to a noticeable decline in performance. The best performance is achieved when $M = 2$, illustrating the importance of capturing subgraph structure for factor disentanglement by stacking multiple layers.
2) Too many disentangled layers (i.e., $M > 2$) may hurt the model performance. This confirms that two disentangled layers are enough for our datasets. Excessive stacking of message-passing-based disentangled layers will lead to over-smoothing and introduce noise to the model.

3) Effect of the Number of Auxiliary Properties: To further investigate whether Meta-DREAM could benefit from auxiliary properties, we study the model performance by varying the number of sampled auxiliary properties. Specifically, we consider two scenarios for discussion: 1) keep all auxiliary properties during the testing stage and vary the number during the training stage and 2) keep all auxiliary properties during the training stage and vary the number during the testing stage. Fig. 4 verifies the performance by varying numbers of auxiliary properties in the training stage and testing stage, respectively. We discover the following.

1) The performance consistently improves in both scenarios as the number of auxiliary properties increases. This indicates that integrating auxiliary properties helps to exploit comprehensive properties that reflect the correlation between molecules and the target property.
2) Decreasing the number of auxiliary properties in the training stage has less impact on the performance than in the testing stage. This provides a feasible way to deal with numerous auxiliary properties (i.e., PCBA with 128 properties), as sampling some of these properties during training can effectively train the model without significantly reducing its performance.



(a) (b)

Fig. 6. t-SNE visualization of the graph representation learned in GS-Meta and the gated weight learned in our model. (a) GS-Meta. (b) Meta-DREAM.

E. Case Study and Visualization (RQ4)

To comprehensively verify the effectiveness of our Meta-DREAM, we conduct three qualitative analyses, including the correlations between the elements in the learned representations, the assignment of clusters, and the visualization of the gated weight.

1) Feature Correlation Analysis: Fig. 5 shows the feature correlation between elements of graph representations generated by GS-Meta, w/o CTR, and Meta-DREAM with five latent factors, respectively. It can be observed that graph representations of GS-Meta are highly entangled. Meanwhile, Meta-DREAM demonstrates a more blockwise correlation pattern than w/o CTR, where elements on the diagonal show a higher correlation. This phenomenon reveals that our model, equipped with the disentangled graph encoder, is capable of extracting mutually exclusive task information for different customization knowledge transfers, achieving a better performance in the few-shot molecular property prediction task.

2) Visualization of Task Structure: To intuitively explore the inherent heterogeneous structure among tasks, we randomly select five tasks with 300 related subgraphs in the ToxCast dataset, and show the t-distributed stochastic neighbor embedding (t-SNE) visualization of the graph representations

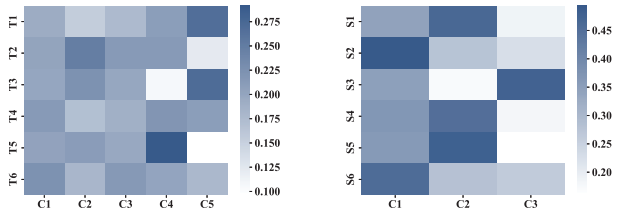


Fig. 7. Visualization of soft-assignment score $s_{t,k}$ of (a) ToxCast and (b) SIDER. Based on the scale of the datasets, we adopt five clusters (i.e., $C1$–$C5$) and three clusters (i.e., $C1$–$C3$) to categorize the randomly sampled properties of ToxCast (i.e., $T1$–$T6$) and SIDER (i.e., $S1$–$S6$), respectively.

learned in GS-Meta and the gated weight learned in our model. For simplicity, we abbreviate five labels as $P1$, $P2$, $P3$, $P4$, and $P5$. As shown in Fig. 6(a), various types of tasks primarily activate distinct clusters within the learned graph representations of GS-Meta, illustrating the heterogeneous nature of task structures. Compared with GS-Meta, the gated weight result shown in Fig. 6(b) further highlights the capability of our soft clustering module to distinguish tasks across different clusters (i.e., $P1$ and $P2$), facilitating the task-specific initial parameter learning within these clusters.

3) Task Clustering Analysis: Recalling to the clustering procedure in Section IV-D, we apply K clusters, with their centers defined by the latent factor prototype vectors (i.e., $\{u_k\}_{k=1}^{K}$). Subsequently, we calculate the assignment score of $g_{t,k}$ in cluster k using (19). This approach allows us to achieve the soft clustering of $\mathcal{G}_t$ related to task t. Here, we randomly select six tasks from the ToxCast and SIDER datasets, which are grouped into five and three clusters, respectively, and visualize the assignment score distribution for each task. As shown in Fig. 7, each task is assigned within different combinations of clusters. For instance, there is often a substantial interplay of interrelated complications observed between $S1$ (renal and urinary disorders), $S4$ (cardiac disorders), and $S5$ (nervous system disorders), indicating strong latent correlations among these properties. Hence, $S1$, $S4$, and $S5$ show similar activation patterns on $C2$. In contrast, $S2$ (pregnancy, puerperium, and perinatal conditions) and $S3$ (ear and labyrinth disorders) are highly distinct tasks, exhibiting separate activation on $C1$ and $C3$, respectively. By disentangling heterogeneous tasks, the factorized representations reflect the intrinsic characteristics of the tasks and activate the corresponding clusters in a customized manner. The results unequivocally demonstrate that Meta-DREAM is capable of assigning appropriate soft clustering combinations based on the underlying characteristics of different tasks, allowing for the customization of cluster-aware knowledge for task-specific initialization.

VI. Conclusion

In this article, we propose a novel framework termed Meta-DREAM for few-shot molecular property prediction, which aims at learning customization knowledge of tasks under each cluster and preserving knowledge generalization via factor disentanglement. Specifically, we construct an HMRG and reformulate the meta-learning episode as a subgraph of HMRG. Further, we leverage the proposed disentangled graph encoder to uncover the underlying different factors within the task. Based on this, a soft clustering module is proposed to group tasks into separate clusters and learn the cluster-aware task-specific initialization under each factor. Experiments on five benchmark datasets demonstrate the efficacy of our Meta-DREAM. In the future, we will extend our framework to multiclass and regression scenarios by readapting the soft-clustering module.

References

[1] Y. Song, S. Zheng, Z. Niu, Z. Fu, Y. Lu, and Y. Yang, "Communicative representation learning on attributed molecular graphs," in *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2020, pp. 2831–2838.

[2] F. Yin et al., "Molecular contrastive learning with chemical element knowledge graph," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 4, pp. 3968–3976.

[3] D. Weininger, A. Weininger, and J. L. Weininger, "SMILES. 2. algorithm for generation of unique SMILES notation," *J. Chem. Inf. Comput. Sci.*, vol. 29, no. 2, pp. 97–101, May 1989.

[4] S. Wang, Y. Guo, Y. Wang, H. Sun, and J. Huang, "SMILES-BERT: Large scale unsupervised pre-training for molecular property prediction," in *Proc. 10th ACM Int. Conf. Bioinf., Comput. Biol. Health Informat.*, Sep. 2019, pp. 429–436.

[5] S. Zheng, X. Yan, Y. Yang, and J. Xu, "Identifying structure–property relationships through SMILES syntax analysis with self-attention mechanism," *J. Chem. Inf. Model.*, vol. 59, no. 2, pp. 914–923, Feb. 2019.

[6] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1263–1272.

[7] Z. Guo et al., "Few-shot graph learning for molecular property prediction," in *Proc. Web Conf.*, Apr. 2021, pp. 2559–2567.

[8] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, "Low data drug discovery with one-shot learning," *ACS Central Sci.*, vol. 3, no. 4, pp. 283–293, Apr. 2017.

[9] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Dec. 2016, pp. 3637–3645.

[10] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1126–1135.

[11] C. Q. Nguyen, C. Kreatsoulas, and K. M. Branson, "Meta-learning GNN initializations for low-resource molecular property prediction," 2020, *arXiv:2003.05996*.

[12] Z. Meng, Y. Li, P. Zhao, Y. Yu, and I. King, "Meta-learning with motif-based task augmentation for few-shot molecular property prediction," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, Jan. 2023, pp. 811–819.

[13] W. Ju et al., "Few-shot molecular property prediction via hierarchically structured learning on relation graphs," *Neural Netw.*, vol. 163, pp. 122–131, Jun. 2023.

[14] Q. Lv, G. Chen, Z. Yang, W. Zhong, and C. Y.-C. Chen, "Meta learning with graph attention networks for low-data drug discovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 11218–11230, 2023.

[15] X. Zhuang, Q. Zhang, B. Wu, K. Ding, Y. Fang, and H. Chen, "Graph sampling-based meta-learning for molecular property prediction," in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, Aug. 2023, pp. 4729–4737.

[16] Y. Wang, A. Abuduweili, Q. Yao, and D. Dou, "Property-aware relation networks for few-shot molecular property prediction," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 17441–17454.

[17] M. Fatima, S. Sadeeqa, and S. U. Rashid Nazir, "Metformin and its gastrointestinal problems: A review," *Biomed. Res.*, vol. 29, no. 11, pp. 2285–2289, 2018.

[18] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[19] J. Gordon, J. F. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner, "Meta-learning probabilistic inference for prediction," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 3915–3924.

[20] N. Lai, M. Kan, C. Han, X. Song, and S. Shan, "Learning to learn adaptive classifier–predictor for few-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3458–3470, Aug. 2021.

[21] H. Chi et al., "Meta discovery: Learning to discover novel classes given very limited data," 2021, *arXiv:2102.04002*.
[22] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 1842–1850.
[23] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–12.
[24] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.
[25] H. Zhang, H. Li, and P. Koniusz, "Multi-level second-order few-shot learning," *IEEE Trans. Multimedia*, vol. 25, pp. 2111–2126, 2023.
[26] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*.
[27] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? Towards understanding the effectiveness of MAML," 2019, *arXiv:1909.09157*.
[28] Y. Liu et al., "Learning to propagate labels: Transductive propagation network for few-shot learning," 2018, *arXiv:1805.10002*.
[29] L. Liu, T. Zhou, G. Long, J. Jiang, and C. Zhang, "Learning to propagate for graph meta-learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Sep. 2019, pp. 1037–1048.
[30] J. Chauhan, D. Nathani, and M. Kaul, "Few-shot learning on graphs via super-classes based on graph spectral measures," 2020, *arXiv:2002.12815*.
[31] X. Lin et al., "Structure-aware prototypical neural process for few-shot graph classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4607–4621, 2022.
[32] I. Spinelli, S. Scardapane, and A. Uncini, "A meta-learning approach for training explainable graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4647–4655, 2022.
[33] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[34] I. Higgins et al., "Beta-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent.*, Apr. 2017.
[35] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, Dec. 2016, pp. 2172–2180.
[36] Y. Wang et al., "DisenCTR: Dynamic graph-based disentangled representation for click-through rate prediction," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 2314–2318.
[37] Y. Wang et al., "Deep graph mutual learning for cross-domain recommendation," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, Jan. 2022, pp. 298–305.
[38] Y. Qin et al., "DisenPOI: Disentangling sequential and geographical influence for point-of-interest recommendation," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, Feb. 2023, pp. 508–516.
[39] H. Li et al., "DisCo: Graph-based disentangled contrastive learning for cold-start cross-domain recommendation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, Apr. 2025, pp. 12049–12057.
[40] Y. Wang et al., "GMR-rec: Graph mutual regularization learning for multi-domain recommendation," *Inf. Sci.*, vol. 703, Jun. 2025, Art. no. 121946.
[41] P. Cheng et al., "Improving disentangled text representation learning with information-theoretic guidance," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7530–7541.
[42] Y. Wang et al., "DisenCite: Graph-based disentangled representation learning for context-specific citation generation," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, pp. 11449–11458.
[43] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, "Learning disentangled representations for recommendation," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Dec. 2019, pp. 5712–5723.
[44] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, May 2019, pp. 4212–4221.
[45] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, and M. Zhang, "DisenHAN: Disentangled heterogeneous graph attention network for recommendation," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2020, pp. 1605–1614.
[46] H. Li, X. Wang, Z. Zhang, Z. Yuan, H. Li, and W. Zhu, "Disentangled contrastive learning on graphs," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 34, Dec. 2021, pp. 21872–21884.
[47] Y. Wang, X. Luo, C. Chen, X.-S. Hua, M. Zhang, and W. Ju, "DisenSemi: Semi-supervised graph classification via disentangled representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 8192–8204, May 2025.
[48] W. Ju et al., "A survey of data-efficient graph learning," in *Proc. 33rd Int. Joint Conf. Artif. Intell.*, Aug. 2024, pp. 8104–8113.
[49] W. Ju et al., "A survey of graph neural networks in real world: Imbalance, noise, privacy and OOD challenges," 2024, *arXiv:2403.04468*.
[50] Q. Lv, G. Chen, Z. Yang, W. Zhong, and C. Y. Chen, "Meta-MolNet: A cross-domain benchmark for few examples drug discovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, pp. 4849–4863, Mar. 2024.
[51] S. Wu, Y. Wang, and Q. Yao, "PACIA: Parameter-efficient adapter for few-shot molecular property prediction," in *Proc. 33rd Int. Joint Conf. Artif. Intell.*, Aug. 2024, pp. 5208–5216.
[52] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2016.
[53] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, Oct. 2018.
[54] W. Ju et al., "A comprehensive survey on deep graph representation learning," *Neural Netw.*, vol. 173, May 2024, Art. no. 106207.
[55] W. Ju et al., "Hypergraph-enhanced dual semi-supervised graph classification," in *Proc. Int. Conf. Mach. Learn.*, May 2024, pp. 22594–22604.
[56] T. Ren et al., "MHGC: Multi-scale hard sample mining for contrastive deep graph clustering," *Inf. Process. Manage.*, vol. 62, no. 4, Jul. 2025, Art. no. 104084.
[57] Z. Wu et al., "MoleculeNet: A benchmark for molecular machine learning," *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, 2018.
[58] R. R. Tice, C. P. Austin, R. J. Kavlock, and J. R. Bucher, "Improving the human hazard characterization of chemicals: A Tox21 update," *Environ. Health Perspect.*, vol. 121, no. 7, pp. 756–765, Jul. 2013.
[59] M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, and P. Bork, "A side effect resource to capture phenotypic effects of drugs," *Mol. Syst. Biol.*, vol. 6, no. 1, p. 343, Jan. 2010.
[60] S. G. Rohrer and K. Baumann, "Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data," *J. Chem. Inf. Model.*, vol. 49, no. 2, pp. 169–184, Feb. 2009.
[61] A. M. Richard et al., "ToxCast chemical landscape: Paving the road to 21st century toxicology," *Chem. Res. Toxicology*, vol. 29, no. 8, pp. 1225–1251, Aug. 2016.
[62] Y. Wang et al., "Pubchem's bioassay database," *Nucleic acids Res.*, vol. 40, no. D1, pp. D400–D412, 2012.
[63] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "TUDataset: A collection of benchmark datasets for learning with graphs," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2020.
[64] G. Koch, R. S. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. Int. Conf. Mach. Learn.*, 2015, vol. 2, no. 1, pp. 1–30.
[65] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11–20.
[66] L. Hou, H. Xiang, X. Zeng, D. Cao, L. Zeng, and B. Song, "Attribute-guided prototype network for few-shot molecular property prediction," *Briefings Bioinf.*, vol. 25, no. 5, p. 394, Jul. 2024.

Haodong Zhang received the B.S. and M.S. degrees in software engineering from the Software College, Northeastern University, Shenyang, China, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree in software engineering.

His research interests include graph representation learning, graph neural networks, and corresponding applications in drug discovery and out-of-distribution detection.

Tao Ren received the B.S., M.S., and Ph.D. degrees in engineering from Northeastern University, Shenyang, China, in 2003, 2005, and 2007, respectively.

He held a post-doctoral position in computer science from Northeastern University in 2013. He is currently a Professor with Northeastern University. He is in charge of 30 projects, such as the National Natural Science Foundation of China. He has authored or co-authored more than 100 high-qualified academic articles in several high-ranking journals or conferences. Furthermore, he has published five books and holds more than 40 Chinese patents. His main research interests include graph representation learning, machine learning, and its applications.

Yifan Wang received the B.S. and M.S. degrees in software engineering from Northeastern University, Shenyang, Liaoning, China, in 2014 and 2017, respectively, and the Ph.D. degree in computer science from Peking University, Beijing, China, in 2023.

He is currently an Assistant Professor with the School of Information Technology and Management, University of International Business and Economics, Beijing. His research interests include graph representation learning, graph neural networks, disentangled representation learning, and corresponding applications such as drug discovery and recommender systems.

Fanchun Meng is currently pursuing the Ph.D. degree in software engineering with the Software College, Northeastern University, Shenyang, China.

He is mainly engaged in deep learning in time series.

Wei Ju received the B.S. degree in mathematics from Sichuan University, Chengdu, Sichuan, China, in 2017, and the Ph.D. degree from the School of Computer Science, Peking University, Beijing, China, in 2022.

He worked as a Post-Doctoral Research Fellow with the School of Computer Science, Peking University. He is currently an Associate Professor with the College of Computer Science, Sichuan University. He has published more than 60 articles in top-tier venues. His current research interests lie primarily in the area of machine learning on graphs including graph representation learning and graph neural networks, and interdisciplinary applications such as recommender systems, bioinformatics, drug discovery, and spatiotemporal analysis.

Dr. Ju won the Best Paper Finalist in IEEE ICDM 2022.

Ying Tian received the B.S., M.S., and Ph.D. degrees in clinical medicine from China Medical University, Shenyang, China, in 2004, 2007, and 2016, respectively.

In 2015, she joined Loma Linda University, Loma Linda, CA, USA, as a Visiting Scholar for further studies. She is currently an Associate Professor and an Associate Chief Physician with the Department of Otorhinolaryngology, First Affiliated Hospital of China Medical University, Shenyang. She holds several academic positions, has participated in three research projects, has authored or co-authored numerous high-quality articles, and holds one Chinese patent. Her main research direction is medical basic and clinical research.