MolCPT: Molecule Continuous Prompt Tuning to Generalize Molecular Representation Learning

Cameron Diao[*†] Kaixiong Zhou[*†] Xiao Huang[‡] Xia Hu[†]

Abstract

Molecular representation learning is crucial for the problem of molecular property prediction, where graph neural networks (GNNs) serve as an effective solution due to their promising structure modeling capability. Since labeled data is often scarce and expensive to obtain, it is a great challenge for GNNs to generalize in the extensive molecule space. Recently, the training paradigm of "pre-train, fine-tune" has been leveraged to improve the generalization performance of GNNs. It uses self-supervised information to pre-train GNNs, and then performs fine-tuning to optimize the downstream task with only a few labels. However, it has been shown that the pre-training does not always provide statistically significant improvement, especially for self-supervised learning with random structural masking. Particularly, the molecular graph is characterized by motif subgraphs, which are frequently occurring and related to molecular properties. To leverage the task-related motifs, we first propose a novel paradigm of "pre-train, prompt, fine-tune" for molecular representation learning, named molecule continuous prompt tuning (MolCPT). Based on the pre-trained model, MolCPT defines a motif prompting function to project the standalone input into an expressive prompt, which augments the molecular graph with meaningful motifs in the continuous embedding space. In this way, the motif prompt provides more structural patterns to aid the downstream classifier in identifying molecular properties. By optimizing downstream classification loss, the motif embeddings are encoded with semantic knowledge for exact molecular analysis. Extensive experiments on various benchmark datasets show that MolCPT efficiently generalizes pre-trained GNNs for molecular property classification with or even without a few fine-tuning steps. Our code is in: https://anonymous.4open.science/r/GraphCL-7105.

1 Introduction

Molecular property prediction is a fundamental task in many fields, such as the predictions of quantum mechanics, physical chemistry, and toxicity [21]. By viewing the atoms and bonds as the nodes and edges, respectively, various graph neural networks (GNNs) have been proposed to model the structure information of molecular graphs [9, 19, 22]. By treating the molecule as a computation graph, GNNs learn the low-dimensional node embeddings by passing messages along edges. The molecule representations are then read out (e.g., sum of node embeddings) to estimate properties.

Along with the exploration of GNN variants for different applications, supervised training is notorious for requiring a sufficiently large amount of input data and label pairs. This places a great burden on molecular representation learning, where the labels are limited compared with the enormous space of possible molecules. Recent efforts have turned to the "pre-train, fine-tune" learning strategy. In particular, they pre-train GNNs with self-supervised learning tasks (e.g., contrastive learning [23] and masking predictive methods [7]) or the open-source molecule collection [2]. The pre-trained model is expected to improve generalization performance on downstream label-limited tasks, with careful fine-tuning.

However, the pre-trained model does not necessarily encode the semantic structure instrumental for molecular analysis. Different from social graphs, the molecule is often characterized by motifs, which are frequently occurring subgraph patterns that are indicative of molecular properties [20]. For example, Benzene ring is a functional motif of organic molecules that indicates aromaticity. The vanilla pre-training, e.g., comparing contrastive examples through random structure masking [7], is ill-suited for learning meaningful motifs and distinguishing between different molecules. Even worse, it is studied that applying self-supervised pre-training does not guarantee significant improvements without careful experimental setup [17]. Under the "pre-train, fine-tune" learning framework, many laborious efforts have been exerted to tailor and align the pre-trained objective with the downstream task.

Instead of being trapped in pre-training objective engineering, we explore the novel strategy of "pre-train, prompt, fine-tune" to generalize molecular represen-

[*] Both authors contribute equally.

[†] Department of Computer Science, Rice University, {cwd2, Kaixiong.Zhou, xia.hu}@rice.edu

[‡] The Hong Kong Polytechnic University, xiaohuang@comp.polyu.edu.hk

tation learning. The prompt technique is first proposed in natural language processing (NLP) [10], where the prompting function augments the input text with knowledge description related to the downstream task. In this way, the pre-trained model will be transferred and generalized effectively on downstream applications. For example, given the pre-trained language model, we consider product review classification with raw input text (e.g., "Absolutely a cost-effective product."). The prompting function reformulates the input text as prompt by appending the task related description (e.g., "Absolutely a cost-effective product. Whether it is good or not?"), which prompts the pre-trained model to produce the desired results. Motivated from the wide success of prompts in NLP [13, 15], we aim at defining the motif prompt to aid pre-trained GNNs deployed on any new molecular analysis.

Despite the conceptual simplicity, it is non-trivial to design the motif prompting function due to the following two challenges. First, it is unclear how to append the molecular graph with corresponding motifs to construct a prompt. Compared with sequential semantic text, graphs are formulated as unordered nodes and their physical connections. The motif prompt requires domain rules or differential algorithms to combine the molecule and its motifs, which cannot be applied to other molecular datasets with different bonding formulas. Second, it is challenging to learn the motif vocabulary containing expressive patterns for molecular property prediction. Most motif detection tools are unsupervised and rely on counting the frequency of subgraphs. The derived motifs may contain too much noise to strongly correlate with the concerned task.

To tackle the above challenges, we propose molecule continuous prompt tuning (MolCPT) in Figure 1 to generalize the pre-trained GNNs and enhance downstream molecular representation learning. Under the paradigm of "pre-train, prompt, fine-tune", without loss of generality, we first adopt self-supervised learning to pre-train the backbone model. We then propose a motif prompting function to augment the molecular graph with motifs in the continuous representation space, and prompt the downstream classifier to easily recognize molecular properties. Specifically, we make three key contributions through MolCPT:

- To flexibly prompt molecular information (challenge 1), our motif prompting function infers continuous representations of the molecular graph and its motifs. Instead of connecting the molecules and motifs in discrete structure space, we concatenate their representation vectors to prepare for downstream classification. This removes the requirement that the motifs should be structurally connected with the molecule, according to specific domain knowledge.
- To denoise the motif vocabulary (challenge 2), we optimize the motif prompt module on the downstream task. Specifically, we treat the motif representations as trainable embeddings, and apply attention networks to organically combine the molecule and motif representations. Via fine-tuning, the motif embeddings learn to store molecular property-related knowledge.
- We evaluate MolCPT on a series of molecular graph benchmarks. Our experimental results demonstrate that the motif prompt efficiently generalizes pre-trained GNNs to identify molecular properties. The average ROC-AUC improvement is up to 14.13%, and the fine-tuning epochs can be reduced to just 50.

2 Preliminary on GNNs and Pre-training

2.1 Molecular Property Prediction A molecule is abstracted as a topological graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the atoms (nodes) and biochemical bonds (edges) within the molecule, respectively. Considering node $v \in \mathcal{V}$, we use $x_v \in \mathbb{R}^d$ to represent the node's initial features, and adopt $\mathcal{N}_v$ to denote the set of its direct neighbors. Let $\mathcal{T} = \{(G, y), \cdots\}$ denote the training set of molecule and label pairs. The molecular property prediction task is to learn the molecular representations and map them to their corresponding label. The molecular representation should generalize over the extensive space even with a small training set of labeled samples.

2.2 Graph Neural Networks GNNs learn the molecular topological structure and atom features with hidden node embeddings, which are read out to generate the molecular representation. Specifically, following the message passing strategy [6], GNNs learn the node embedding by recursively aggregating its neighborhood and combining with itself. At the k-th layer, the embedding learning of node v is formulated as:

$$x_v^{(k)} = \text{AGGRE}(\{x_{v'}^{(k-1)}, v' \in \mathcal{N}_v \cup v\}, \theta^{(k)}). \quad (2.1)$$

$x_v^{(k)} \in \mathbb{R}^d$ is the embedding vector at the k-th layer; $x_v^{(0)} = x_v$ at the initial layer; $\theta^{(k)} \in \mathbb{R}^{d \times d}$ is a trainable weight to encode atom features; and AGGRE denotes the aggregation and combination function of node embeddings (e.g., through sum, mean or max pooling). Suppose the number of graph convolutional layers is K. To facilitate the following expression, we use $x_v^{(K)} \triangleq f_\theta(G, v)$ to represent the final node representation learned from K-layer GNNs, where $\theta = \{\theta^{(1)}, \cdots, \theta^{(K)}\}$ is the set of trainable parameters.

To obtain the molecule representation used for property prediction, a readout function (e.g., sum or mean pooling) is applied to pool all the node embeddings as: $h_G \triangleq f_\theta(G) = \text{READOUT}(\{x_v^{(K)}, v \in \mathcal{V}\})$.

2.3 Pre-train and Fine-tune One of the key challenges in molecular property prediction is posed by limited and imbalanced labels, which tend to cause GNNs to overfit. To improve generalization capability, pre-training methods have been adopted to learn robust and transferable knowledge of molecular graphs. Notably, most of them randomly mask part of the molecule, and then pre-train the GNN to recover it (i.e., in context predictive learning [7]) or maximize the mutual information between the original molecule and masked one (i.e., in contrastive learning [20, 23])

Given the pre-trained model f_θ, it serve as initialization to fine-tune on molecular property prediction. Mathematically, model f_θ is connected with a new classifier head p_φ, often a multi-layer perceptron (MLP) with parameters φ. They are fine-tuned together as:

$$\begin{aligned} &\min_{\theta,\varphi} && \textstyle\sum_{(G,y)\in\mathcal{T}} \mathcal{L}(p_\varphi(f_\theta(G)); y), \\ &\text{s.t.} && \theta^{\text{init}} = \theta^{\text{pre}}. \end{aligned} \tag{2.2}$$

The constraint means the GNN parameters are initialized as the pre-trained model. $\mathcal{L}$ is the classification loss function, such as cross-entroy loss.

3 Molecule Continuous Prompt Tuning

Although many pre-training methods have been proposed to learn the transferable knowledge, it has been observed that the generalization improvement depends a lot on the experimental setting. For example, the pre-training objective needs to be tailored according to the downstream problem, while the hyperparameters are required to be tuned laboriously [17]. Particularly, by unifying the experimental hyperparameters, applying self-supervised pre-training only achieves the marginal improvement or even performs worse than without pre-training. This is because the self-supervised pre-training mainly relies on the random context masking for both the predictive learning and contrastive learning. It is difficult for such general pre-training to extract motifs, which are the frequently appearing and indicative for the downstream molecular properties. For example, Benzene rings are the elemental subgraphs for organic molecules, while carbon rings and NO2 groups are prone to be mutagenic. Although a number of motif related works has been studied previously, they are mainly used for the traditional unsupervised embedding learning [11, 14]. Some recent solutions have been proposed to leverage the motifs for pre-training. They rely on the motif learning module and domain rule for the masking or generation of motifs to pre-train GNNs [20, 24], which is inefficient and non-general to the diverse downstream analysis. Thus it is worthy to study how to use motifs in an easy plug-in way for any molecular property classification problem.

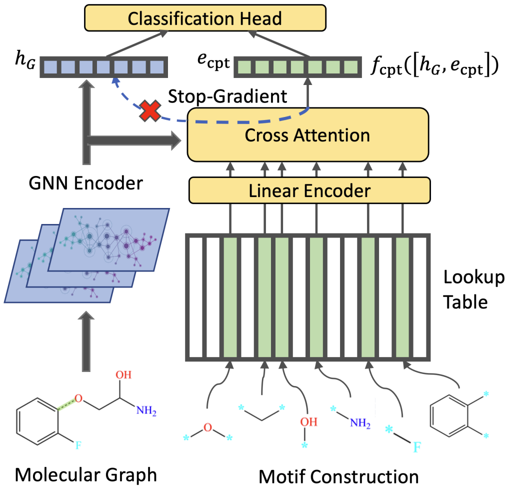

Figure 1: Overview of MolCPT motif prompt generation. Molecule graph G passes through the GNN encoder to obtain h_G. Concurrently, G is fragmented into several motifs. MolCPT looks up the motifs' embeddings, and passes them into attention module to obtain e_{cpt}. h_G and e_{cpt} are concatenated to produce a motif prompt as input to the classification head.

In this work, we shift to investigate the motif prompt under the new training paradigm of "pre-train, prompt, fine-tune". The prompt technique is firstly proposed in NLP, which reformulates the input text by appending the indicative descriptions according to the downstream application (see example in Introduction). Given a suite of appropriate prompts, a single model pre-trained with the general self-supervised fashion can be adapted to diverse problems. This advantage frees one from the laborious design of specific pre-training objective for each task, and allows the recycle of previous model in literature. Our MolCPT is proposed to leverage the motifs to prompting the input molecules, and generalize the pre-trained GNNs to each given task.

3.1 Pre-train, Prompt, Fine-tune Before going to the technical details of MolCPT, we first mathematically define the motif prompting function, and then lay out the whole pipeline of "pre-train, prompt, fine-tune".

Definition 3.1. (Graph Motif) *Consider molecule graph $G = (\mathcal{V}, \mathcal{E})$. We define its set of motifs as:*

$$\mathcal{M}_G \triangleq \{M^{(1)}, \cdots, M^{(n)}\}, \tag{3.3}$$

where $M^{(j)}$ denotes the j-th motif, and n denotes the *total number of motifs belonging to molecule G. Since motifs are defined as the subgraphs of G, each motif $M^{(j)} = (\mathcal{V}^{(j)}, \mathcal{E}^{(j)})$ where $\mathcal{V}^{(j)} \subset \mathcal{V}$ and $\mathcal{E}^{(j)} \subset \mathcal{E}$.*

DEFINITION 3.2. (MOTIF PROMPTING FUNCTION) *A motif prompting function f_{prompt} is used to reformulate the input molecule by appending the series of corresponding motifs:*

$$G' = f_{\text{prompt}}(G, \mathcal{M}_G), \tag{3.4}$$

where G' is named as motif prompt, which is a new hypergraph generated by highlighting the key subgraph patterns to indicate the molecular property prediction. Several possible solutions could be adopted to define f_{prompt}. For example, one can simply concatenate them as disjoint hypergraph, or defines rules to specify the connections between the motifs and their molecule. Our MolCPT is illustrated in Figure 1, which realizes the new learning paradigm of "pre-train, prompt, fine-tune" via the following three steps:

- **Model pre-training.** As introduced in Section 2.3, we use the general pre-training objective to learn backbone GNNs. This removes the engineering efforts to extensively design and pre-train model for molecule analysis at hand.
- **Prompt addition.** According to Eq. (3.4), we reformulate the standalone molecule as motif prompt by appending the indicative motifs.
- **Prompt fine-tuning.** We use motif prompt G' to replace the original graph G in the fine-tuning objective of Eq. (2.2). All the parameters (or only the new weights involved in prompting function and classification head) are efficiently updated to optimize the downstream loss with a few epochs.

As pointed out above, to generalize the prompt technique to molecule analysis, the challenges lie in how to define the expressive prompting function to connect motifs and molecule, and how to fine-tune the prompting function to indicate the downstream applications. Particularly, MolCPT consists of three functional modules: motif-based corpus generation to extract the meaningful motif subgraphs, continuous prompting function to reformulate the input data, and motif constrained learning to fine-tune model.

3.2 Motif-based Corpus Generation

Considering the countless connection modes between atoms in the real-world datasets, the set of possible motifs may be too large to represent the key knowledge of molecular properties. MolCPT first preprocesses the molecule dataset to construct a limited vocabulary of motifs, which serves as prompting corpus to inform molecule.

In general, the motif fragmentation should following two formulas: (i) The produced motifs must contains the semantic structure information, e.g. by extracting meaningful functional groups from the dataset. (ii) The produced motifs must occur frequently enough. MolCPT uses the fragmentation method provided by Motif-based Graph Self-Supervised Learning (MGSSL) framework [25]. MGSSL adapts the Breaking of Retrosynthetically Intereesting Chemical Substructures (BRICS) algorithm [3] to form the motif vocabulary. By further filtering out the low frequent motifs, we mathematically define the motif vocabulary as follows:

DEFINITION 3.3. (MOTIF VOCABULARY) *For dataset $\mathcal{T} = \{(G, y), \cdots\}$, let $\mathcal{M}_G$ denote the corresponding set of fragmented motifs from each graph G. A motif vocabulary is the union of frequent motifs:*

$$\mathcal{M}_{\text{mol}} \triangleq (\cup_{G \in \mathcal{T}} \{M^{(j)} \in \mathcal{M}_G : |M^{(j)}| \geq t\}) \cup \{M^{(0)}\}, \tag{3.5}$$

where t is a threshold hyperparameter to select the frequently appearing motifs, and $|M^{(j)}|$ refers to the number of molecules in the whole dataset containing motif $M^{(j)}$. Notably, $M^{(0)}$ is an empty motif containing none of nodes or edges. After the threshold filtering, there may exist molecules without any frequent motif associated with them. We include the empty motif to characterize and prompt these molecules.

3.3 Continuous Prompting Function

Unlike the sequential prompting function in NLP that directly concatenates the discrete words, it is difficult for the graph prompting function to connect the structured motifs to their molecule. The disjoint combination overlooks their inherent correlations and results in poor performance. Although there are some related works applying the deterministic rules or differentiable algorithms to build up the connections, they either rely on the domain knowledge, or deteriorate the training efficiency due to the large amount of candidate links.

Instead of optimizing over discrete structure, we propose the continuous prompting function to concatenate the motifs and molecule embeddings in the hidden representation space. Specifically, it is defined below:

DEFINITION 3.4. (CONTINUOUS PROMPTING FUNCTION) *A continuous prompting function f_{cpt} concatenates the embeddings of molecule G and its frequent motifs, and generates the prompt embedding as:*

$$h'_G = f_{\text{cpt}}(h_G, \{e^{(j)}, \text{for } M^{(j)} \in \mathcal{M}_G \cap \mathcal{M}_{\text{mol}}\}). \tag{3.6}$$

h_G and h'_G is the original and resulted prompt embedding of molecule graph G, respectively. $e^{(j)} \in \mathbb{R}^d$

is the embedding of motif $M^{(j)}$, such as one-hot embedding. We consider the frequently appearing fragmented motifs, which are defined by the intersection set of $\mathcal{M}_G \cap \mathcal{M}_{\text{mol}}$. For the molecule without any meaningful motif, we use the embedding of empty motif to unify the prompt learning. To obtain the expressive prompt embedding h'_G, we free the motif embeddings as trainable table, and parameterize the continuous prompting function with attention module as follows.

Motif Embedding Table. We adopt a trainable table to maintain the motif embeddings, which are fine-tuned together to learn the indicative motif knowledge. Mathematically, we define the motif embedding table $\mathbf{E}_{\text{mol}} \in \mathbb{R}^{|\mathcal{M}_{\text{mol}}| \times d}$, where $|\mathcal{M}_{\text{mol}}|$ denotes the cardinality of motif vocabulary. Each motif embedding $e^{(j)}$ is look up correspondingly from table.

A proper embedding initialization is crucial for the molecular property prediction. One naive strategy is to randomly initialize and rely on training to learn the semantic meaning. However, the random initialization fails to distinguish the diverse motifs at the initial stage, and may converge to poor generalization area. To indicate the molecule analysis with the meaningful motif prompt, we propose to initialize them via inferring the pre-trained model. Recall from Definition 3.1 that each motif $M^{(j)}$ has a subgraph structure $(\mathcal{V}^{(j)}, \mathcal{E}^{(j)})$. Thus, by treating the motif as input, the pre-trained GNNs could be used to infer the subgraph embedding and initialize $e^{(j)}$ as: $e^{(j)} \triangleq f_\theta(M^{(j)})$. Notably, we initialize the empty motif $M^{(0)}$ as $e^{(0)} = \vec{\mathbf{0}}$, where $\vec{\mathbf{0}}$ denotes the all-zeros vector representing the semantic absence.

Molecule-motif Cross Attention. Depending on the chemical bonds between the motifs and the molecule, a common sense is the diverse motifs weight differently to the final molecular properties. Although the continuous prompting function removes the connection constraints between motifs and molecules, it would lead to the poor topological encoding between them and the drop of performance. Besides, given the various numbers of motifs among molecules, one is incapable of directly concatenating their embeddings and then feeding them into the downstream classifier.

To address these problems, we leverage the multi-head attention module to weight the motifs, and generate the abstract motif representation for one molecule. Mathematically, let $\mathbf{E}_G$ denote the motif embedding matrix by considering the contained motifs in $\mathcal{M}_G \cap \mathcal{M}_{\text{mol}}$ for molecule G, where the motif embeddings are stacked row-wisely. The abstract motif representation $e_{\text{cpt}} \in \mathbb{R}^d$ to prompt molecule G is given by:

$$e_{\text{cpt}} = \text{softmax}\left(\frac{(\mathbf{W}_q h_G)^\top (\mathbf{E}_G \mathbf{W}_k)}{\sqrt{d}}\right) \mathbf{E}_G \mathbf{W}_v, \tag{3.7}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v \in \mathbb{R}^{d \times d}$ are the query, key, and value transformation matrices, respectively. In particular, we treat the graph representation a query to attend on the related motifs, and obtain the abstract motif representation indicating the molecular properties. Finally, the continuous prompting function f_{cpt} in Eq. (3.6) is instantiated as: $h'_G = \text{Concate}(h_G, e_{\text{cpt}}) \in \mathbb{R}^{2d}$. h'_G is used as input to classifier p_φ to estimate properties.

Given the above motif attention, we further adopt an empirical trick to freeze the backward gradients on graph representation h_G. In other word, we decouple the computations of molecule graph representation and abstract motif embedding, and stop the backward gradients crossing between them. The rationale behind the gradient freezing are explained from two perspectives: (i) This allows MolCPT to learn the informative motif prompts flexibly without restricting to preserve knowledge held in the pre-trained GNNs; (ii) The decoupling allows ones to directly infer the existing GNNs, and only fine-tune the light weights in prompting function and classifier. Similar to NLP, the motif prompt is an efficient plug-in technique to extend a specific model to solve a great number of molecular analysis problems.

3.4 Motif Constrained Learning Based on the continuous prompt embedding h'_G, we reformulate the fine-tuning objective as:

$$\tag{3.8} \begin{array}{cl} \min_{\theta,\varphi,\mathbf{E}_{\text{mol}},\mathbf{W}_{q,k,v}} & \sum_{(G,y)\in\mathcal{T}} \mathcal{L}(p_\varphi(h'_G); y) + \lambda ||\mathbf{E}_{\text{mol}}||_2, \\ \text{s.t.} & \theta^{\text{init}} = \theta^{\text{pre}}. \end{array}$$

λ is a loss hyperparameter, and $||\mathbf{E}_{\text{mol}}||_2$ denotes its L2 norm. We note that only a small fraction of related motif embeddings are updated for a batch of molecules. Since some motifs involve frequently during the fine-tuning process, we use L2 penalty to constrain the scale of their embeddings to be comparable with the others.

As analyzed in the last section, model parameter θ could also be fixed to facilitate the deployment of any pre-trained GNNs. MolCPT is then simplified into the MLP-based model with two extra potential advantages: (i) *Fine-tuning efficiency.* By storing the molecule graph representations, we can avoid repeating the time-consuming graph convolutions each time for the new analysis tasks on the same dataset. (ii) *Engineering efficiency.* The indicative motif prompt decouples the co-design of pre-training and fine-tuning, and accelerate the engineering deployment. The traditional co-design has to select the proper pre-training objective and then test the fine-tuning performance, which requires tedious trials and repeated graph convolutional computations. In MolCPT, we only needs to design the indicative prompt to adapt any a given pre-trained GNNs to each problem, which is as efficient as MLP tuning.

4 Related Work

Pre-training GNNs. Graph pre-training aims to capture significant structural patterns in the input graph distribution, in a self-supervised manner [7]. A growing number of graph pre-training modules are making significant advancements in various problem domains. Deep Graph Infomax [18] pre-trains GNNs to maximize mutual information between subgraph representations and high-level graph summaries. Hu et al. [7] pre-trains on the level of individual nodes as well as entire graphs. GraphCL [23] uses contrastive learning with domain-specific graph augmentations, significantly improving transfer learning on molecular tasks. Certain pre-training tasks are designed specifically for molecular property prediction. MGSSL [25] introduces a motif generation pre-training task that captures substructures in molecular graphs. MolCLR [20] proposes different graph augmentations for contrastive learning to capture the general molecular structure.

Prompt in Natural Language Processing. A growing number of researchers have adopted a new approach for applying pre-trained language models on downstream tasks, called "pre-train, prompt, predict" [10]. This approach does not use a task-specific pre-training objective, but instead defines task-specific prompting functions that align the original pre-training task with the help of textual prompts. Prompts can be engineered for better downstream performance, in two different ways. Manual Template Engineering creates prompt templates from human intuition [13]. Automated Template Learning learns prompt templates as part of the finetuning procedure, and can learn over discrete [4,15] or continuous prompts [12]. Although there have been some prompted related works proposed in graph domain, they mainly target at node classification in the knowledge or social graphs [1,16].

5 Experiments

We investigate how molecular representations augmented by MolCPT perform on downstream prediction tasks. Specifically, we raise the following questions: **Q1:** Compared with standard supervised and pre-training baselines, how effective is MolCPT at boosting molecular property prediction scores? **Q2:** How efficient does the motif prompting function adapt the pre-trained model for downstream application? **Q3:** How does the choice of motif vocabulary affect downstream performance, with and without the vocabulary filtering step? **Q4:** How does the prompt initialization strategy affect prompt tuning? We provide extra experiments in the appendix to verify MolCPT is more powerful when stacked with other prompt techniques.

5.1 Datasets We evaluate MolCPT on eight benchmark datasets of molecular property classification contained in MoleculeNet [21], namely BBBP, BACE, ClinTox, Tox21, SIDER, HIV, MUV, and ToxCast. Details for these datasets are described in Appendix A.

5.2 Baselines To evaluate the proposed MolCPT based on "pre-train, prompt, fine-tune", we mainly compare with two categories of baselines, including the supervised algorithms without pre-training and the pre-trained approaches without prompt.

Supervised baselines. We consider state-of-the-art GNNs widely used for molecular property prediction, which are updated only via the training label set. Particularly, we include GCN [9], GIN [22], GAT [19], and GraphSAGE [6].

Pre-trained baselines. This section refers to the training paradigm of "pre-train, fine-tune", where the backbone GNNs are first pre-trained with self-supervised techniques and then fine-tuned with the training label set. We consider five typical pre-trained approaches, including Infomax [18] maximizing the mutual information between nodes and their corresponding graph, AttrMasking [7] predicting the masked node features, ContextPred [7] predicting the masked topology context, GraphCL [23] contrasting the graphs and their randomly perturbed counterparts, and MolCLR [20] leveraging motif elements to generate contrasting pairs.

5.3 Implementation of MolCPT To demonstrate the generality of MolCPT to the pre-training objectives, we choose three pre-trained baselines to implement the plug-in prompt technique. Specifically, we adopt AttrMasking, GraphCL and MolCLR due to their superior performances for molecule analysis. To guarantee the fair comparison, we use the same settings provided in their open repositories, including the 5-layer GIN backbone, pre-training objectives, and the pre-training as well as fine-tuning hyperparameters. Besides, MolCPT uses 4 attention heads by default. We tune the key hyperparameters of motif filtering threshold t and loss penalty λ to fully demonstrate the prompt advantage.

5.4 Molecule Property Prediction Studies We compare MolCPT with the baseline models in Table 1 to answer **Q1**. We make the following observations:

❶ *MolCPT generally boosts the model performance for the molecular property prediction.* It should be noted that the pre-trained approaches are prone to outperform the supervised ones by encoding the prior topological knowledge. Based upon AttrMasking, GraphCL, and MolCLR frameworks, our MolCPT further generalize the pre-trained model to indicate the molecular properties with the motif prompt. Particularly, compared with

Table 1: Results on eight MoleculeNet tasks by averaging 5 runs and measuring in ROC-AUC (%). Results on MUV and ToxCast are omitted for MolCLR because its open-sourced repository does not support evaluation.

Frameworks	Methods	Datasets							
		BBBP	**BACE**	**ClinTox**	**Tox21**	**SIDER**	**HIV**	**MUV**	**ToxCast**
Supervised	GCN [9]	64.9±3.0	73.6±3.0	65.8±4.5	74.9±0.8	60.0±1.0	75.7±1.1	73.2±1.4	63.3±0.9
	GIN [22]	65.8±4.5	70.1±5.4	58.0±4.4	74.0±0.8	57.3±1.6	75.3±1.9	71.8±2.5	63.4±0.6
	GAT [19]	66.2±2.6	69.7±6.4	58.5±3.6	75.4±0.5	60.9±1.4	72.9±1.8	66.6±2.2	64.6±0.6
	GraphSAGE [6]	66.2±2.6	69.7±6.4	58.5±3.6	75.4±0.5	60.9±1.4	72.9±1.8	66.6±2.2	64.6±0.6
Pre-train	Infomax [18]	68.8±0.8	75.9±1.6	69.9±3.0	75.3±0.5	58.4±0.8	76.0±0.7	75.3±2.5	62.7±0.4
	AttrMasking [7]	64.3±2.8	79.3±1.6	71.8±4.1	**76.7±0.4**	61.0±0.7	77.2±1.1	74.7±1.4	64.2±0.5
	ContextPred [7]	68.0±2.0	79.6±1.2	65.9±3.8	75.7±0.7	60.9±0.6	77.3±1.0	75.8±1.7	63.9±0.6
	GraphCL [23]	70.1±0.3	73.8±0.1	80.8±0.6	73.8±0.3	59.3±0.2	**77.3±0.9**	70.1±1.1	61.7±0.1
	MolCLR [20]	73.9±0.2	78.9±0.6	83.7±1.8	72.3±0.3	60.5±0.1	76.9±0.3	—	—
Prompt	AttrMasking+MolCPT	66.2±1.3	**79.7±0.3**	85.9±1.1	76.4±0.2	60.0±0.5	76.4±0.2	**76.5±1.0**	**64.8±0.2**
	GraphCL+MolCPT	72.9±0.7	77.1±0.5	84.3±0.9	75.7±0.3	**62.1±0.3**	75.2±0.4	76.2±0.6	63.4±0.1
	MolCLR+MolCPT	**74.7±0.3**	79.2±1.1	**86.5±1.1**	72.6±0.1	61.4±0.3	76.9±0.1	—	—

GraphCL, MolCPT improves the average test scores by up to **6.10%** on 7 out of 8 tasks. Compared with MolCLR, MolCPT improves scores by up to **2.85%** on all 6 tasks, and compared with AttrMasking, MolCPT improves scores by up to **14.13%** on 5 out of 8 tasks. These results also empirically validate that MolCPT is agnostic to the choice of pre-training framework. One could easily reuse the general pre-training objective, and quickly extend it to the downstream problem by designing the adaptive motif prompt.

❷ *MolCPT reduces the negative transfer resulted from the misaligned pre-training and fine-tuning objectives.* We find that GraphCL exhibits the negative transfer and performs even poorly than vanilla GCN on Tox21, SIDER, MUV, and ToxCast datasets. The authors attribute this issue to the ill-posed graph augmentations that corrupt chemical structure information during pre-training. In other word, the pre-training objective with random topological masking is not in line with the molecule analysis. The "pre-train, fine-tune" strategy thus often requires extensive efforts to design the pre-training objectives for each specific problem. In this work, MolCPT bridges the objective misalignment with the motif prompting function, which augments the input molecule with indicative and related motifs. This supports our hypothesis that the continuous motif embeddings can learn the semantic property knowledge from the downstream application.

5.5 Freezing Weight We further investigate **Q2** by fixing the pre-trained weights, which aims to evaluate whether the motif prompt could adapt the agnostic pre-trained model to the problem at hand. We consider two weight freezing cases: (i) Freeze the pre-trained GNNs, and only train the new classification head and continuous prompting function; (ii) Freeze pre-trained GNNs and the randomly initialized classifier, and only fine-tune prompt. The first case mimics the real-world settings, where one is freed to update the parameters involved in complex graph convolutions. The second ablation targets at extremely estimating whether MolCPT alone can prompt the pre-trained GNNs to the downstream application. We list the results in Table 2 and make the following observations:

❸ *Motif prompt facilitates the accurate estimation of molecular properties, without needing to fine-tune backbone GNNs.* In general, with the frozen pre-trained GNN, MolCPT improves the average test scores by **4.21%** above MolCLR, and **1.98%** above GraphCL. Notably, when freezing the pre-trained GNNs as well as classification head, GraphCL and MolCLR are degenerated to random guessing, and constrained around the ROC-AUC score of 0.5. In contrast, by tuning the plug-in continuous prompt embeddings, our MolCPT encodes the task-relevant knowledge to indicate the pre-trained GNNs towards property identification. For the case of freezing only the pre-trained GNNs, MolCPT has the poor results on HIV and MUV. One of the possible reasons is the motif vocabularies are much larger in these two benchmarks, which poses great challenge to the standalone tuning of motif prompts.

5.6 Impact of Fine-tuning Time To further answer **Q2**, we reduce the fine-tuning epochs to just 50, and evaluate whether the motif prompt could quickly adapt pre-trained model for molecular property prediction. The results are listed in Table 3, where we observe:

❹ *MolCPT tends to delivers the desired classification performance even with limited fine-tuning epochs.* This is because the motif prompting function bridges the pre-training and fine-tuning objectives. The pre-trained GNNs are efficiently tailored for the new task with the motif prompt, which is highly related to the molecular properties. We illustrate loss dynamic in Figure 3, where MolCPT converges quickly to a stable generation area. Recalling the last experimental observa-

Table 2: ROC-AUC (%) results for models with freezing pre-trained GNNs and/or classification head.

Freeze Weights	Methods	BBBP	BACE	ClinTox	Tox21	SIDER	HIV	MUV	ToxCast
Pre-trained	GraphCL [23]	60.2±0.0	66.2±0.0	57.0±0.0	67.9±0.0	54.7±0.0	**70.5**±0.0	**70.8**±0.0	56.3±0.0
	MolCLR [20]	67.3±0.5	61.28±0.4	44.0±0.7	69.15±0.5	**57.5**±0.3	67.8±0.6	—	—
	GraphCL + MolCPT	60.5±0.8	**74.1**±0.5	**73.4**±0.8	67.4±0.7	55.9±0.3	64.5±0.8	65.7±2.2	**57.9**±0.3
	MolCLR + MolCPT	**68.8**±0.3	68.5±2.3	67.6±1.1	**70.9**±0.2	55.8±0.5	60.7±0.4	—	—
Pre-trained & Classification Head	GraphCL [23]	55.3±0.0	51.6±0.0	59.3±0.0	49.0±0.0	49.7±0.0	42.6±0.0	48.9±0.0	50.6±0.0
	MolCLR [20]	47.9±4.3	56.1±6.4	47.8±1.6	51.5±1.0	49.9±1.0	52.6±2.4	—	—
	GraphCL + MolCPT	51.6±0.1	71.1±0.2	**73.8**±0.7	**63.2**±0.7	**52.7**±0.4	**63.5**±0.4	**50.3%**±1.2	**53.3**±0.1
	MolCLR + MolCPT	**64.2**±1.6	**71.4**±1.3	54.7±3.0	58.0±1.8	50.3±0.7	49.3±3.4	—	—

Table 3: ROC-AUC results by evaluating backbone models and MolCPT restricted to 50 finetuning epochs.

Methods	BBBP	BACE	ClinTox	Tox21	SIDER	HIV	MUV	ToxCast
GraphCL	70.4±0.4	73.6±0.1	79.3±0.7	73.7±0.5	58.0±0.5	**78.2±0.7**	70.9±0.3	61.7±0.1
MolCLR	**73.7±0.4**	78.3±0.6	81.1±0.1	72.3±0.1	61.2±0.2	75.5±0.5	—	—
GraphCL + MolCPT	71.3±0.8	78.0±0.5	84.7±1.1	**75.5±0.2**	59.5±0.2	75.7±0.3	**75.9±0.6**	**63.7±0.1**
MolCLR + MolCPT	72.6±0.7	**79.2±0.7**	**85.1±1.1**	72.7±0.2	**61.7±0.3**	75.0±0.5	—	—

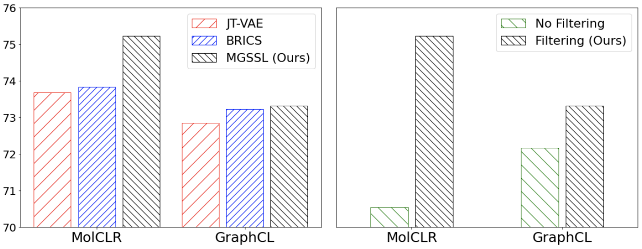

Figure 2: Ablation studies of motif detection method (Left) and filtering (Right), averaging over all datasets.

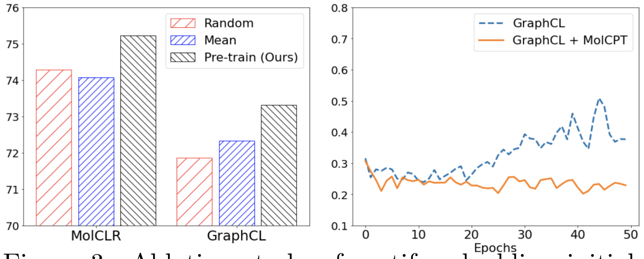

Figure 3: Ablation study of motif embedding initialization (Left), averaging over all datasets. Validation losses of GraphCL and GraphCL + MolCPT on the Tox21 dataset (Right).

tion, we reach the conclusion that MolCPT enables the pre-trained model to be efficiently and reliably inferred with or even without a few fine-tuning epochs.

5.7 Number of Key Motifs To answer **Q3**, we study two key components affecting the size of motif vocabulary in MolCPT: the motif fragmentation method, and the filtering threshold. By ablating these components on backbones GraphCL and MolCLR, we have the following observations visualized in Figure 2:

❺ *The motif fragmentation method heavily affects downstream task performance.* We evaluate three different fragmentation methods, in order of increasing granularity: JT-VAE [8], MGSSL (ours) [25], and BRICS [3]. Granularity is an important consideration when generating the motif vocabulary: [25] finds that coarse-grained motifs have lower occurrence frequencies, preventing the model from learning motif embeddings suited for downstream application. On the other hand, fine-grained motifs are typically single atoms or bonds, which capture no semantic meaning. We find that the intermediate granularity of MGSSL produces the most informative motifs (Figure 1).

❻ *The filtering threshold improves model's generalization capability on downstream tasks.* Without a filtering step, the motif vocabulary contains rare motifs whose embeddings are updated on very few molecules. These rare motif embeddings cannot correlate with the property related knowledge, and thus fail to indicate the molecule analysis.

5.8 Prompt Initialization Method We investigate **Q4** by testing three different initialization methods for motif embeddings $\mathbf{E}_{\mathrm{mol}}$: random embedding, mean embedding, and pre-trained embedding (ours). For the random embedding, we used uniform Xavier initialization [5]. We define the mean embedding initialization as follows: Given a motif $M^{(j)}$ in the motif vocabulary, its embedding is initialized by the mean representation of all molecules $M^{(j)}$ belongs to. We show the ablation result in Figure 3, and observe:

❼ *Choosing the right initialization method is crucial for learning motif embeddings that capture semantic meaning.* Given the large volume of molecules in a dataset, the mean embedding initialization has the comparable performance with the random one. The random or mean initialization cannot distinguish between different motifs at the starting phase. Thus the motif prompt fails to provide the semantic knowledge to identify the molecular properties. We propose to initialize them by inferring the subgraph embeddings of motifs, which is simple but surprisingly effective.

6 Conclusion

We propose MolCPT, the first "pre-train, prompt, finetune" training paradigm for molecular property prediction. Specifically, MolCPT proposes the motif prompting function to bridge the gap between pre-training and fine-tuning objectives, where the key motifs are used as corpus to augment the input molecule. Towards freeing from the complex link connections between motifs and their molecule, MolCPT instead uses the continuous prompting function. It concatenate the motifs and their molecule at the hidden embedding dimension. A proper initialization method, attention module, and constrained learning are adopted to encode the motif embeddings with the semantic knowledge related to the molecular property. Extensive experiments on eight benchmarks show the generality, effectiveness, and efficiency to improve pre-trained models.

References

[1] X. Chen, N. Zhang, X. Xie, S. Deng, Y. Yao, C. Tan, F. Huang, L. Si, and H. Chen, Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction, in Proceedings of the ACM Web Conference, 2022.

[2] S. Chithrananda, G. Grand, and B. Ramsundar, Chemberta: large-scale self-supervised pretraining for molecular property prediction, arXiv preprint arXiv:2010.09885, (2020).

[3] J. Degen, C. Wegscheid-Gerlach, A. Zaliani, and M. Rarey, On the art of compiling and using 'drug-like' chemical fragment spaces, ChemMedChem, 3 (2008), pp. 1503–1507.

[4] T. Gao, A. Fisch, and D. Chen, Making pre-trained language models better few-shot learners, arXiv preprint arXiv:2012.15723, (2020).

[5] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.

[6] W. Hamilton, Z. Ying, and J. Leskovec, Inductive representation learning on large graphs, Advances in neural information processing systems, 30 (2017).

[7] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, Strategies for pre-training graph neural networks, 2020.

[8] W. Jin, R. Barzilay, and T. Jaakkola, Junction tree variational autoencoder for molecular graph generation, 2018.

[9] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, 2016.

[10] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021.

[11] N. Pržulj, Biological network comparison using graphlet degree distribution, Bioinformatics, 23 (2007), pp. e177–e183.

[12] G. Qin and J. Eisner, Learning how to ask: Querying lms with mixtures of soft prompts, 2021.

[13] T. Schick and H. Schütze, Exploiting cloze questions for few shot text classification and natural language inference, 2020.

[14] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, Efficient graphlet kernels for large graph comparison, in Artificial intelligence and statistics.

[15] T. Shin, Y. Razeghi, R. L. Logan, E. Wallace, and S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, 2020.

[16] M. Sun, K. Zhou, X. He, Y. Wang, and X. Wang, Gppt: Graph pre-training and prompt tuning to generalize graph neural networks, in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, Association for Computing Machinery, 2022, p. 1717–1727.

[17] R. Sun, Does gnn pretraining help molecular representation?, arXiv:2207.06010, (2022).

[18] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, Deep graph infomax., ICLR (Poster), 2 (2019), p. 4.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, Graph attention networks, 2017.

[20] Y. Wang, J. Wang, Z. Cao, and A. B. Farimani, Molecular contrastive learning of representations via graph neural networks, Nature Machine Intelligence, 4 (2022), pp. 279–287.

[21] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, Moleculenet: A benchmark for molecular machine learning, 2017.

[22] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, How powerful are graph neural networks?, 2018.

[23] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, Graph contrastive learning with augmentations, in Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 5812–5823.

[24] Z. Yu and H. Gao, Molecular representation learning via heterogeneous motif graph neural networks, in International Conference on Machine Learning, PMLR, 2022, pp. 25581–25594.

[25] Z. Zhang, Q. Liu, H. Wang, C. Lu, and C.-K. Lee, Motif-based graph self-supervised learning for molecular property prediction, NeurIPS, 2021.