# SELF-ATTENTIVE RATIONALIZATION FOR GRAPH CONTRASTIVE LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

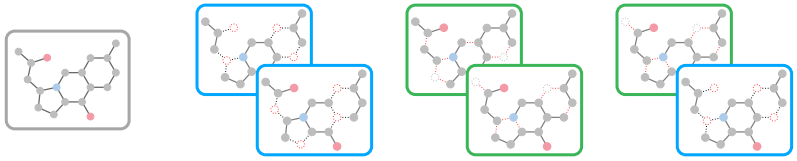
Graph augmentation is the key component to reveal instance-discriminative features of a graph as its rationale in graph contrastive learning (GCL). And existing rationale-aware augmentation mechanisms in GCL frameworks roughly fall into two categories and suffer from inherent limitations: (1) non-heuristic methods with the guidance of domain knowledge to preserve salient features, which require expensive expertise and lack generality, or (2) heuristic augmentations with a co-trained auxiliary model to identify crucial substructures, which face not only the dilemma between system complexity and transformation diversity, but also the instability stemming from the co-training of two separated sub-models. Inspired by recent studies on transformers, we propose Self-attentive Rationale guided Graph Contrastive Learning (SR-GCL), which integrates rationale finder and encoder together, leverages the self-attention values in transformer module as a natural guidance to delineate semantically informative substructures from both node- and edge-wise views, and contrasts on rationale-aware augmented pairs. On real-world biochemistry datasets, visualization results verify the effectiveness of self-attentive rationalization, and the performance on downstream tasks demonstrates the state-of-the-art performance of SR-GCL for graph model pre-training.

## 1 INTRODUCTION

Graph augmentation is a crucial enabler for graph contrastive learning (GCL) (You et al., 2020; Qiu et al., 2020; Zhu et al., 2020). It pre-trains the model to yield instance-discriminative representations by contrasting augmented samples without hand-annotated labels against each other. To achieve this goal, early studies (You et al., 2020; 2021; Qiu et al., 2020; Zhu et al., 2020) conduct random corruptions in topological structures (*i.e.,* nodes and edges) or attributes to construct contrastive pairs. However, such random corruptions, especially on salient substructures, easily cause a semantic gap between two augmented views of the same anchor graph, misguiding the following contrastive optimization procedure (Wang et al., 2021; Li et al., 2022).

There has been recent interest in rationale discovery (Chang et al., 2020; Suresh et al., 2021; Li et al., 2022) as graph augmentation to resolve the aforementioned problem. We systematize these studies as rationale-aware augmentation, where a rationale exhibits a graph's instance-discriminative information from the others. The dominant paradigm often consists of two subsequent modules: the rationale discovery function and the rationale encoder, which aim at creating the rationale-aware views and yielding their representations to contrast, respectively. To find rationales, early studies turn to domain knowledge to highlight the salient parts of graphs (Zhu et al., 2021; Liu et al., 2022). For instance, Rong et al. (2020) leverage RDkit (Landrum, 2010), an assistant software of chemistry, to capture crucial functional groups with high activity in molecules. However, such expertise is expensive or even inaccessible in some scenarios (Tang et al., 2014). Besides, bringing in too much prior knowledge might harm generalization (Wang et al., 2022). To mitigate this problem, recent works (Suresh et al., 2021; Li et al., 2022) introduce an auxiliary model instead to automatically identify rationales, which is termed the rationale finder and co-trained with the rationale encoder. In this ad-hoc scheme, however, we reveal two inherent limitations:

- Typically, the finder is tailor-made for one single transformation of graph data (Suresh et al., 2021; Li et al., 2022), forcing the focus on either node- or edge-wise rationales (*e.g.,* Figures 1(b) and 1(c)). And we argue that a high-performing finder is supposed to be equipped with plural views



(a) Original graph (b) Node-wise views (c) Edge-wise views (d) Dual views

Figure 1: Rationale-aware graph augmentation preserves instance-discriminative features in original graph (*e.g.,* 1(a)). Existing frameworks tailor-make an auxiliary model for one single transformation (*e.g.,* 1(b) or 1(c)). SR-GCL constructs both node- and edge-wise rationale-aware views (*e.g.,* 1(d)).

(a) Separate finder-encoder scheme (b) Integrated scheme

Figure 2: Separate finder-encoder in 2(a) may lead to instability and biased representation, while SR-GCL integrates them together in 2(b) to pursue high-quality rationales and representations.

from both node and edge (*e.g.,* Figure 1(d)). The lack of view diversity confines the rationale-aware augmentations to one transformation, while leaving the cross-transformation untouched. Worse still, it degenerates the effectiveness of contrastive learning, because studies (Chen et al., 2020; You et al., 2020) find that "no single transformation suffices to learn good representations". Here we ascribe this crux of finder to the lack of transformation diversity.

- As illustrated in Figure 2(a), the finder, aiming at discovering rationales, separates from the consequent encoder, which specializes in encoding them. While conceptually appealing, we hypothesize that these separate modules cooperate with each other to pursue high-quality rationales unsmoothly. Because the supervision signal of the finder is remotely generated by the contrastive optimization of the encoder, much of which is weak. Moreover, co-optimizing two sub-models could make the pre-training more complicated and time-consuming but less stable.

To resolve these limitations, we draw inspiration from the transformers (Vaswani et al., 2017) to reshape the finder-encoder scheme. Despite originally being proposed for language (Devlin et al., 2019) and vision tasks (Dosovitskiy et al., 2021), transformers are attracting a surge of interest in graph area (Wu et al., 2021; Chen et al., 2022; Rampásek et al., 2022). At the core is the self-attention operation, which models pairwise connections between tokens and yields high-quality representations. We find self-attention de facto a natural mechanism to concurrently discover and condense rationale information from both edge- and node-wise transformations. By prepending a special token as its proxy and treating its nodes as other tokens, self-attention is able to elegantly indicate the importance of each node and each edge. Sampling nodes and edges based on the importance scores (*i.e.,* heterogeneous transformations) allows us to generate both the node- and edge-wise subgraphs (*i.e.,* rationales) simultaneously. Moreover, self-attention can directly output the rationale representations without additional modules. In stark contrast to the prior finder-encoder scheme, the "self-attentive rationalization" not only accomplishes diverse rationales in one shot, but also integrates the functions of rationale discovery and encoding together.

With the Self-attentive Rationalization as graph augmentation, we incorporate it into GCL and name the framework **SR-GCL**. Specifically, two augmented views stem from the node- and edge-wise rationales, respectively; subsequently, the contrastive optimization pulls close representations of contrastive pair augmented from the same anchor graph and pushes away those of different anchors by minimizing contrastive loss. Compared with conventional GCL methods, our SR-GCL collates and conflates the instance-discriminative information across both node- and edge-wise transformations to construct rationale-aware contrastive pairs from dual views. Henceforth, we find that such a strategy improves the generalization performance of pre-trained model on downstream tasks, while simultaneously interpreting the contribution of each node/edge to instance-discrimination. Extensive experiments show that our SR-GCL sets the new state-of-the-art for graph pre-training across a number of molecule benchmark datasets in (Wu et al., 2018a). We release our codes at https://anonymous.4open.science/r/SR-GCL-EDD3.

# 2 PRELIMINARY

## 2.1 GRAPH CONTRASTIVE LEARNING

As a prevalent line of self-supervised learning (SSL) (Chen et al., 2020; He et al., 2022), GCL has been a rising topic among researchers in the graph area. It pre-trains the backbone model to yield instance-discriminative representations by contrasting augmented views of graph samples against each other in large label-free datasets, thus benefiting the downstream supervised finetuning. The typical paradigm of existing GCL frameworks consists of two modules: (1) graph-structure data augmentation and (2) contrastive optimization.

### 2.1.1 GRAPH AUGMENTATION

Graph augmentation transforms a graph in two correlated but different views. Early studies (You et al., 2020; 2021; Qiu et al., 2020; Zhu et al., 2020) instantiate it by random mechanisms, which corrupt the graph structures or attributes with uniform probability. For example, GraphCL (You et al., 2020) constructs augmented views with strategies of node dropping, edge perturbation, or attribute masking in a random fashion. GCC (Qiu et al., 2020) augments nodes' $r$-ego network by random walks with restarts. Nonetheless, overlooking the importance of keeping saliency properties during augmentation may result in potential loss of semantic information, misleading following contrastive optimization and undermining the goal of instance-discrimination.

**Rationale-guided Augmentation.** To tackle this problem, various methods (Zhu et al., 2021; Liu et al., 2022) have been proposed to preserve salient semantics in augmented views, which we term as rationale-guided augmentation. A research line leverages external domain knowledge to bridge the potential semantic gap between augmented views. For instance, in the social network domain, GCA (Zhu et al., 2021) proposes to utilize node centrality measures to capture crucial connective structures. Still, expertise could be expensive or even inaccessible in some scenarios (Tang et al., 2014), making the application very limited and the performances of generalizing to unseen domains are easily damaged by these over-specific guidance (Wang et al., 2022). The limitations above call for rationale-aware but automatic augmentation mechanisms.

**Rationale Finder.** More recently, GCL frameworks (Suresh et al., 2021; Li et al., 2022; Xu et al., 2022) adopt auxiliary models, *aka.* rationale finder, to conduct heuristic rationale discovery. Specifically, AD-GCL (Suresh et al., 2021) hires another graph model as the rationale finder to identify salient edges for edge-wise views. RGCL (Li et al., 2022) captures semantically informative substructures to create node-wise views with a GNN-based rationale finder. However, the single-view (*i.e.,* only node- or edge-wise) confines the model's performance to a sub-optimal position, which is consistent with the observations in (Chen et al., 2020; You et al., 2020). To improve the diversity of views, a naive solution is to adopt multiple auxiliary finders, each of which specialises one view. Nonetheless, you cannot have your cake and eat it – by this way, more diversity on rationale-aware transformations demands more complexity on systems (*i.e.,* more auxiliary finders).

**Rationale Encoder.** Apart from the rationale finder, existing rationale-aware GCL frameworks separate the rationale finder from the encoder (*i.e.,* the backbone model). The back-propagation of gradients to the rationale encoder is natural when optimizing the contrastive loss of representations yielded by it. Nevertheless, to enable the propagation to the rationale finder, AD-GCL Suresh et al. (2021) leverages the Gumbel-Max reparameterization, which may lead to instability stemming from a large variance of gradients (Jang et al., 2017). And RGCL (Li et al., 2022) mixes the outputs of rationale encoder and finder to generate graph representations to contrast, making the output of the encoder biased – the rationale finder will be discarded after pre-training. Here, we blame the separation of rationale finder and encoder for the aforementioned instability and biased representations.

### 2.1.2 CONTRASTIVE OPTIMIZATION

With contrastive pairs in hand, GCL performs in instance-discriminative task by optimizing contrastive loss, including NCE (Wu et al., 2018b), InfoNCE (van den Oord et al., 2018) and NT-Xent (Chen et al., 2020). During this procedure, the agreement between two augmented views of the same anchor graph is encouraged, while the divergence between those of different anchors is enforced.

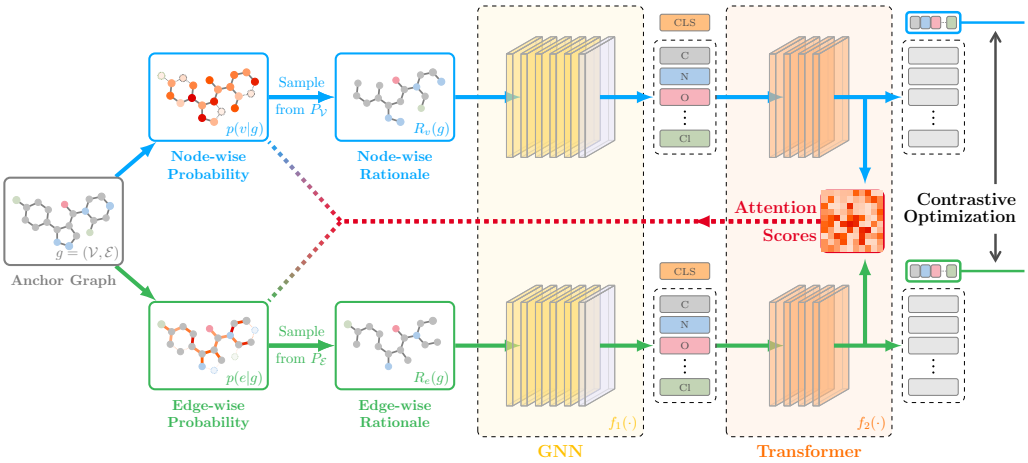



Figure 3: SR-GCL framework, which constructs node- and edge-wise rationales (*i.e.,* $R_v(g)/R_e(g)$) by sampling from corresponding probability distributions (*i.e.,* $P_\mathcal{V}(\cdot|G)/P_\mathcal{E}(\cdot|G)$). Then contrastive optimization is performed to encourage the agreement between views of the same anchor.

### 2.2 Graph Transformer

Transformer has achieved remarkable success in various domains (Vaswani et al., 2017; Devlin et al., 2019; Dosovitskiy et al., 2021). To capture global relations and avoid over-smoothing, researchers also attempt to apply it in graph area (Dwivedi & Bresson, 2020; Kreuzer et al., 2021; Mialon et al., 2021; Ying et al., 2021; Wu et al., 2021; Chen et al., 2022; Rampásek et al., 2022).

The self-attention mechanism attracts researchers to adapt it to graph models. However, graph is not naturally in agreement with the input format of transformers, but a data structure with topology and high regularity. Thus, early studies transform a graph into a sequence of nodes and only implicitly incorporate topological information by adding absolute (Dwivedi & Bresson, 2020; Kreuzer et al., 2021), relative (Mialon et al., 2021) or structural (Ying et al., 2021) encoding.

More recently, hybrid structures have been proposed to integrate GNN and transformer. In Graph-Trans Wu et al. (2021), the GNN module first learns local structure to generate token features, and then the transformer module focuses on capturing global relationships. SAT (Chen et al., 2022) leverages a GNN to encode structure information, while a transformer captures interactions between nodes. More recently, GPS (Rampásek et al., 2022) provides a comprehensive recipe for hybrid models, including transformer structures and encoding methods.

## 3 Methodology

The necessity of preserving instance-discriminative semantics when augmenting graphs has been revealed in (Suresh et al., 2021; Li et al., 2022). However, they only focus on discovering rationales from a single view of node/edge, thus resulting in sub-optimal performance (Chen et al., 2020; You et al., 2020). Thus, we are motivated to propose a novel pre-training framework, Self-attentive Rationale guided Graph Contrastive Learning (SR-GCL). As shown in Figure 3, it utilizes the self-attention map generated by the transformer module to discover the rationales from dual views: both node- and edge-wise, and then conducts contrastive optimization to learn instance-discriminative representations. The algorithm of SR-GCL is also provided in Appendix A.

### 3.1 Graph Augmentation with Self-attentive Rationalization

We refer to a graph as $g = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are node and edge set of it, respectively. Let $\mathbf{x}_v \in \mathbb{R}^d$ denote the $d$-dimension node attributes for node $v \in \mathcal{V}$. Moreover, with a slight abuse of notation, uppercase $G$ and $R(G)$ represent random variables of graph and its rationale. Correspondingly,

lowercase $g$ and $R(g)$ are samples of the aforementioned variables. And conditional probabilistic function and conditional probabilistic distribution are denoted by $p(\cdot|\cdot)$ and $P(\cdot|\cdot)$, respectively.

#### 3.1.1 SELF-ATTENTION FOR DUAL VIEWS

Treating the node set of a graph as a sequence of tokens, the input/output of a transformer layer can be denoted by $\mathbf{X} = \left[\mathbf{x}_{\text{cls}}, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{|\mathcal{V}|}\right]$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the representation of the $i$-th node with a hidden dimension of $d$ and $\mathbf{x}_{\text{cls}} \in \mathbb{R}^d$ is a special trainable token prepended as the graph-level representation for contrastive optimization in pre-training or classification in downstream finetuning. At the core of transformer is the self-attention mechanism, which establishes connections between each token pair to yield high-quality representations. The self-attention and forward propagation in $l$-th transformer layer can be abstracted as:

$$a^l_{v,u} = \frac{\kappa(\mathbf{x}^l_v, \mathbf{x}^l_u)}{\sum_{w\in\mathcal{V}} \kappa(\mathbf{x}^l_v, \mathbf{x}^l_w)}, \qquad \mathbf{x}^{l+1}_v = \sum_{u\in\mathcal{V}} a^l_{v,u}(\mathbf{W}^l_V \mathbf{x}^l_u), \quad (1)$$

$$\kappa(\mathbf{x}, \mathbf{x}') := \exp\left(\left\langle \mathbf{W}^l_Q \mathbf{x}, \mathbf{W}^l_K \mathbf{x}' \right\rangle \Big/ \sqrt{d}\right), \quad (2)$$

where $a_{v,u}$ evaluates the contribution of node $u$ to the representation of node $v$, $\mathbf{W}_{Q/K/V}$ are trainable projection matrices in $\mathbb{R}^{d\times d}$ and $\langle\cdot,\cdot\rangle$ is the dot product operation.

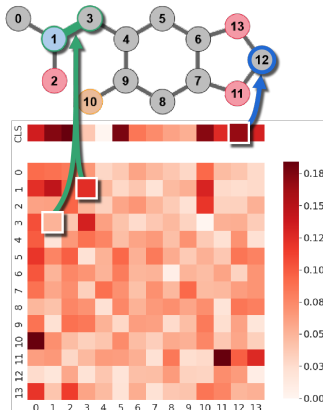



Figure 4: Self-attention map.

For intelligibility, we reshape the attention values yielded by the inherent self-attention mechanism in transformer into a 2-D attention map in Figure 4, which delineates the discriminative power of each node and edge. Specifically, for a graph $g$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$, the importance of node $v$ and edge $e$ (*i.e.,* $p_v$ and $p_e$) can be approximated as the normalized attention score of node $v$ to $\langle$CLS$\rangle$ token and that of edge $e$'s two endpoints $u_e$ and $v_e$, respectively:

$$p_v = \frac{\kappa(\mathbf{x}_{\text{cls}}, \mathbf{x}_v)}{\sum_{u\in\mathcal{V}} \kappa(\mathbf{x}_{\text{cls}}, \mathbf{x}_u)}, \qquad p_e = \frac{\kappa(\mathbf{x}_{u_e}, \mathbf{x}_{v_e})}{\sum_{e'\in\mathcal{E}} \kappa(\mathbf{x}_{u_{e'}}, \mathbf{x}_{v_{e'}})}, \quad (3)$$

where $\kappa(\cdot,\cdot)$ is the exponential kernel function defined in equation 2, $\kappa(\mathbf{x}_{\text{cls}}, \mathbf{x}_v)$ evaluates node $v$'s attribution to the graph-level representation, *i.e.,* the $\langle$CLS$\rangle$ token, and $\kappa(\mathbf{x}_{u_e}, \mathbf{x}_{v_e})$ reflects the volume of information propagated on edge $e$. With the contributions of each node and edge to instance-discrimination in hand, we can further conduct probabilistic sampling to generate both node- and edge-wise rationale-aware views for contrastive optimization.

#### 3.1.2 NODE- & EDGE-WISE RATIONALE GENERATION

Following RGCL (Li et al., 2022), we first approximate the distribution of rationale-aware views and then conduct probabilistic sampling to maintain the diversity of views.

**Node- and edge-wise rationale-aware views.** Given a graph $g$, the probability distribution of its node-wise rationale-aware view $P_{\mathcal{V}}$ is approximated as:

$$P_{\mathcal{V}}(R(G) = R(g)|G = g) = \prod_{v\in\mathcal{V}_R} p(v|g) \prod_{v\in\mathcal{V}\setminus\mathcal{V}_R} (1 - p(v|g)), \quad (4)$$

where $\mathcal{V}$ and $\mathcal{V}_R$ are the node sets of $g$ and rationale $R(g)$, respectively; $p(v|g)$ denotes the probability of $v$ being included into $R(g)$, which quantizes how semantically informative it is. Similarly, the edge-wise probability distribution $P_{\mathcal{E}}$ can be formulated as:

$$P_{\mathcal{E}}(R(G) = R(g)|G = g) = \prod_{e\in\mathcal{E}_R} p(e|g) \prod_{e\in\mathcal{E}\setminus\mathcal{E}_R} (1 - p(e|g)), \quad (5)$$

where $\mathcal{E}$ and $\mathcal{E}_R$ are the edge sets of $g$ and $R(g)$, respectively; $p(e|g)$ reflects the discriminative power of edge $e$. And $p(v|g)$ (or $p(e|g)$) is parameterized by $p_v$ (or $p_e$) in equation 3 at the first layer of the transformer module as in GraphTrans (Wu et al., 2021).

**Sampling with fixed ratios.** To construct anchor graph $g$'s node-wise rationale-aware view, we sample from $P_{\mathcal{V}}(\cdot|G=g)$ under the constraint of a fixed node sampling ratio $\rho_v$:

$$R_v(g) \sim P_{\mathcal{V}}(\cdot|G=g) \quad \text{s.t.} \ |\mathcal{V}_R| = \rho_v \cdot |\mathcal{V}|. \tag{6}$$

After obtaining sampled nodes, the edges between them are kept to construct a complete subgraph. Analogically, edge-wise rationale-aware view can be generated with another edge sampling ratio $\rho_e$:

$$R_e(g) \sim P_{\mathcal{E}}(\cdot|G=g) \quad \text{s.t.} \ |\mathcal{E}_R| = \rho_e \cdot |\mathcal{E}|, \tag{7}$$

where the endpoints of sampled edges are preserved, while the others are discarded.

Without involving any auxiliary models, two rationale-aware views with different transformations (*i.e.,* node- and edge-wise) are generated under the guidance of self-attention rationalization.

### 3.2 Representation Learning and Contrastive Optimization

Following prior studies (Wu et al., 2021; Rampásek et al., 2022), we instantiate the backbone model as an integrator of GNN and transformer, which has powerful representation ability and comparatively low computation cost. Furthermore, with the rationale-aware views (*i.e.,* $R_v(g)/R_e(g)$), we follow the pipeline of representation learning and contrastive optimization to empower the backbone model with instance-discriminative ability.

**Rationale representation learning.** First, the GNN module $f_1(\cdot)$ takes in a subgraph $R(g)$ with $|\mathcal{V}_R|$ nodes and outputs node-level representations $\mathbf{h}_i \in \mathbb{R}^{d_{\text{GNN}}}$ with hidden dimensions of $d_{\text{GNN}}$, which are then converted into a series of tokens summarized as $\mathbf{X}^0$, formally:

$$\left[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{|\mathcal{V}_R|}\right] = f_1(R(g)), \quad \mathbf{X}^0 = \left[\mathbf{x}_{\text{cls}}, \mathbf{W}\mathbf{h}_1, \mathbf{W}\mathbf{h}_2, ..., \mathbf{W}\mathbf{h}_{|\mathcal{V}_R|}\right], \tag{8}$$

where $\mathbf{W} \in \mathbb{R}^{d_{\text{TF}} \times d_{\text{GNN}}}$ is a linear projection layer, $\mathbf{x}_{\text{cls}} \in \mathbb{R}^{d_{\text{TF}}}$ is the $\langle$CLS$\rangle$ token appended to the sequence, whose representation at the final layer is viewed as the graph-level representation. The token series are then fed into the transformer module $f_2(\cdot)$ consisting of Self-Attention (SA), Layer Normalization (LN), and Multi-Layer Perceptron (MLP) blocks with residual connections. The forward propagation in $l$-th transformer layer can be formulated as:

$$\mathbf{X}^{l+1} = f_2(\mathbf{X}^l) = \text{MLP}(\text{LN}(\mathbf{Y}^l)) + \mathbf{Y}^l, \qquad \mathbf{Y}^l = \text{SA}(\text{LN}(\mathbf{X}^l)) + \mathbf{X}^l, \tag{9}$$

where SA operation computes as Equation 1 with multi-heads.

As illustrated in Figure 3, the backbone model follows the aforementioned procedure to yield the graph-level representations for node- and edge-wise rationale-aware views:

$$\mathbf{z}_v = f_2(f_1(R_v(g))), \qquad \mathbf{z}_e = f_2(f_1(R_e(g))), \tag{10}$$

where $f_1(\cdot)$ and $f_2(\cdot)$ are GNN and transformer module, respectively; $\mathbf{z}_v$ and $\mathbf{z}_e$ are the graph-level representations of node- and edge-wise views. Then we follow the standard operation of projecting them to a latent space through a MLP-based projection head with $l_2$ norm $h(\cdot)$:

$$h(\mathbf{z}) = L_2(\text{MLP}(\mathbf{z})), \qquad \mathbf{r}_v = h(\mathbf{z}_v), \qquad \mathbf{r}_e = h(\mathbf{z}_e). \tag{11}$$

The transformer module plays the roles of both the rationale finder, whose attention map guides rationale-aware augmentations of both node- and edge-wise views, and the encoder, who generates graph representations for contrastive optimization. And without adopting any auxiliary model, the natural integration of the rationale finder and encoder makes the training stable and smooth.

**Contrastive optimization.** Finally, we adopt Info-NCE (van den Oord et al., 2018) to maximize the mutual information between node- and edge-wise rationale-aware views of the same anchor graph:

$$\min_{f_1, f_2, h} \mathcal{L} = \mathbb{E}_{g \in \mathcal{G}}\, l(g), \qquad l(g) = -\log \frac{\exp\left({\mathbf{r}_1^+}^\top \mathbf{r}_2^+ / \tau\right)}{\exp\left({\mathbf{r}_1^+}^\top \mathbf{r}_2^+ / \tau\right) + \sum_{\mathbf{r}^- \in \mathcal{R}_g^-} \exp\left({\mathbf{r}_1^+}^\top \mathbf{r}^- / \tau\right)}, \tag{12}$$

where $\mathbf{r}_1^+$ and $\mathbf{r}_2^+$ are generated from the views of the same anchor (*i.e.,* positive views), $\mathcal{R}_g^-$ summarizes the representations of the other graphs' views in the minibatch (*i.e.,* negative views). After pre-training, the projection head $h(\cdot)$ will be thrown away. And it's worth mentioning that SR-GCL can be easily adapted to any transformer-based graph model with self-attention mechanism.

Table 1: Test ROC-AUC scores (%) of pre-trained models on downstream benchmark datasets. G represents the commonly used GIN model in previous baselines, while H is the hybrid model of GIN and transformer in our paper. Statistics of G models are from their original papers and those of H models are obtained by replanting the H model into their official released pre-training codes. red and blue indicate the best and the second best performance on each dataset, respectively.

| Dataset | | BBBP | Tox21 | ToxCast | SIDER | ClinTox | MUV | HIV | BACE | AVG. | GAIN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No Pre-Train | G | 65.8 ±4.5 | 74.0 ±0.8 | 63.4 ±0.6 | 57.3 ±1.6 | 58.0 ±4.4 | 71.8 ±2.5 | 75.3 ±1.9 | 70.1 ±5.4 | 67.0 | - |
| | H | 68.12±2.94 | 73.48±0.32 | 62.67±0.56 | 59.44±1.26 | 69.45±3.12 | 72.18±1.16 | 74.41±1.45 | 74.23±4.62 | 68.24 | 1.24 |
| Infomax | G | 68.8 ±0.8 | 75.3 ±0.5 | 62.7 ±0.4 | 58.4 ±0.8 | 69.9 ±3.0 | 75.3 ±2.5 | 76.0 ±0.7 | 75.9 ±1.6 | 70.3 | 3.3 |
| | H | 69.15±0.68 | 74.62±0.45 | 61.85±0.75 | 60.14±0.72 | 74.78±2.66 | 74.83±1.52 | 74.95±1.26 | 76.32±1.78 | 70.83 | 3.83 |
| EdgePred | G | 67.3 ±2.4 | 76.0 ±0.6 | 64.1 ±0.6 | 60.4 ±0.7 | 64.1 ±3.7 | 74.1 ±2.1 | 76.3 ±1.0 | 79.6 ±1.2 | 70.3 | 3.3 |
| | H | 69.32±1.24 | 76.78±0.49 | 64.43±0.72 | 60.15±0.62 | 73.61±2.49 | 72.41±1.26 | 75.87±0.82 | 78.46±1.87 | 71.37 | 4.37 |
| AttrMasking | G | 64.3 ±2.8 | 76.7 ±0.4 | 64.2 ±0.5 | 61.0 ±0.7 | 71.8 ±4.1 | 74.7 ±1.4 | 77.2 ±1.1 | 79.3 ±1.6 | 71.1 | 4.1 |
| | H | 66.42±3.16 | 77.22±0.67 | 64.89±0.89 | 60.88±0.45 | 77.28±3.88 | 74.12±1.98 | 77.23±1.51 | 79.56±2.13 | 72.20 | 5.20 |
| ContextPred | G | 68.0 ±2.0 | 75.7 ±0.7 | 63.9 ±0.6 | 60.9 ±0.6 | 65.9 ±3.8 | 75.8 ±1.7 | 77.3 ±1.0 | 79.6 ±1.2 | 70.9 | 3.9 |
| | H | 68.52±1.12 | 75.29±0.56 | 63.23±0.46 | 61.29±0.75 | 74.90±3.18 | 70.92±2.36 | 76.82±1.13 | 80.16±1.84 | 71.39 | 4.39 |
| GraphCL | G | 69.68±0.67 | 73.87±0.66 | 62.40±0.57 | 60.53±0.88 | 75.99±2.65 | 69.80±2.66 | 78.47±1.22 | 75.38±1.44 | 70.77 | 3.77 |
| | H | 70.41±1.06 | 73.82±1.04 | 63.06±0.41 | 60.38±1.28 | 77.79±3.02 | 73.81±2.02 | 75.62±0.89 | 78.28±1.08 | 71.64 | 4.64 |
| GraphLoG | G | 72.5 ±0.8 | 75.7 ±0.5 | 63.5 ±0.7 | 61.2 ±1.1 | 76.7 ±3.3 | 76.0 ±1.1 | 77.8 ±0.8 | 83.5 ±1.2 | 73.4 | 6.4 |
| | H | 67.59±1.58 | 75.95±0.80 | 63.63±0.57 | 59.85±2.11 | 79.10±3.24 | 72.84±1.79 | 72.54±1.60 | 83.55±1.96 | 71.88 | 4.88 |
| AD-GCL | G | 70.01±1.07 | 76.54±0.82 | 63.07±0.72 | 63.28±0.79 | 79.78±3.52 | 72.30±1.61 | 78.28±0.97 | 78.51±0.80 | 72.72 | 5.72 |
| | H | 67.88±0.97 | 72.97±1.19 | 63.19±0.44 | 60.18±0.95 | 78.83±1.60 | 75.62±2.54 | 75.54±1.29 | 73.37±1.55 | 70.95 | 3.95 |
| JOAO | G | 70.22±0.98 | 74.98±0.29 | 62.94±0.48 | 59.97±0.79 | 81.32±2.49 | 71.66±1.43 | 76.73±1.23 | 77.34±0.48 | 71.90 | 4.90 |
| | H | 70.78±0.62 | 75.25±1.46 | 63.79±0.84 | 61.04±0.87 | 78.68±2.97 | 75.67±2.57 | 76.98±1.16 | 77.49±1.23 | 72.46 | 5.46 |
| RGCL | G | 71.42±0.66 | 75.20±0.34 | 63.33±0.17 | 61.38±0.61 | 83.38±0.91 | 76.66±0.99 | 77.90±0.80 | 76.03±0.77 | 73.16 | 6.16 |
| | H | 71.36±0.85 | 75.72±0.40 | 63.93±0.36 | 60.91±0.56 | 80.02±1.59 | 75.94±1.19 | 77.81±0.62 | 79.92±1.02 | 73.20 | 6.20 |
| SR-GCL | H | 72.60±0.89 | 77.30±0.84 | 65.86±0.61 | 61.46±1.36 | 81.84±1.60 | 77.76±1.18 | 78.52±0.74 | 81.58±0.97 | 74.62 | 7.62 |

# 4 EXPERIMENTS

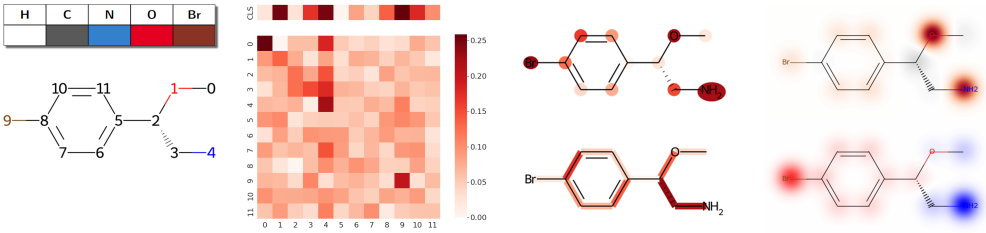
In this section, we conduct extensive experiments on benchmark biochemical molecule datasets to demonstrate the state-of-the-art performance of SR-GCL. Also, we visualize the rationales to illustrate the effectiveness of the self-attentive rationalization mechanism proposed by us. Finally, we provide analyses for hyperparameter sensitivity.

## 4.1 PERFORMANCE COMPARISON

**Setup.** Following the settings in (Hu et al., 2020), the backbone model is pre-trained on large scale label-free dataset – ZINC-2M (Sterling & Irwin, 2015) and then finetuned on 8 benchmark datasets with multi graph-level classification tasks in MoleculeNet (Wu et al., 2018a). Due to the limitation of space, we include statistics of datasets in Appendix B.

**Baselines.** We compare SR-GCL with prior competitive graph pre-training baselines, including Infomax (Velickovic et al., 2019) , EdgePred (Hamilton et al., 2017), AttrMasking Hu et al. (2020), ContextPred (Hu et al., 2020), GraphCL (You et al., 2020), GraphLoG (Xu et al., 2021), AD-GCL (Suresh et al., 2021), JOAO (You et al., 2021) and RGCL (Li et al., 2022). We also give briefs for the aforementioned baselines in Appendix C.

**Performance on downstream tasks.** With the pre-trained backbone model in hand, we append an MLP-based classifier and then finetune it on downstream small-scale datasets in MoleculeNet (Wu et al., 2018a). In table 1, we present all statistics from their original papers with the backbone of commonly used GIN model (Xu et al., 2019), which is denoted as G. To make a fair comparison, we also re-run all baselines with the same hybrid model of GNN and transformer in our paper, which we denote as H. On each dataset, we run finetuning procedures 10 times with random seeds from 0 to 9 to collect the test ROC-AUC of the epoch with the highest validation score, the mean and standard deviation statistics of which are showcased in Table 1. The detailed model structures and the computational efficiency comparison of commonly used GIN and hybrid model are included in Appendix D and E for your reference.




(a) (b) (c) (d)

Figure 5: 5(a): a molecule graph. 5(b): corresponding attention map. 5(c): node- (top) and edge-wise (bottom) attention, where darker color indicates higher attention weights. 5(d): the contribution of each atom to Gasteiger partial charge (top) and partition coefficient (bottom).

Table 2: Ablation Study

| Dataset | BBBP | Tox21 | ToxCast | SIDER | ClinTox | MUV | HIV | BACE | AVG. |
|---|---|---|---|---|---|---|---|---|---|
| GraphCL | 70.41±1.06 | 73.82±1.04 | 63.06±0.41 | 60.38±1.28 | 77.79±3.02 | 73.81±2.02 | 75.62±0.89 | 78.28±1.08 | 71.64 |
| SR-GCL node-wise | 73.01±0.94 | 75.12±1.68 | 65.69±1.02 | 61.21±0.93 | 80.47±3.55 | 77.92±3.12 | 77.89±0.68 | 81.33±0.92 | 74.08 |
| SR-GCL edge-wise | 71.62±1.87 | 74.98±0.96 | 64.87±0.65 | 61.32±1.59 | 80.29±2.68 | 75.11±1.63 | 77.64±1.68 | 79.69±0.87 | 73.19 |
| SR-GCL dual-views | 72.60±0.89 | 77.30±0.84 | 65.86±0.61 | 61.46±1.36 | 81.84±1.60 | 77.76±1.18 | 78.52±0.74 | 81.58±0.97 | 74.62 |

We get following observations from the comparison of SR-GCL with multiple baselines:

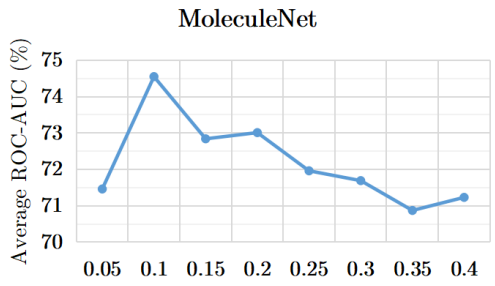
1. **SR-GCL outperforms existing pre-training baselines.** SR-GCL empowers the backbone hybrid model with an average performance gain of 6.38 on 8 downstream tasks. And compared with other pre-training strategies, it also achieves best result on 5 out of 8 datasets and 3 second bests – the first two positions on BACE are both GraphLoG models, which verifies the effectiveness of SR-GCL as a method of self-supervised learning benefiting backbone's representation ability.

2. **Rationale-aware augmentations benefits GCL.** Compared with vanilla GCL framework with random augmentations – GraphCL, frameworks with rationale-aware augmentations (*e.g.,* AD-GCL, RGCL and SR-GCL) achieves better performance, demonstrating the crucial role of augmentation mechanism to reveal the instance-discriminative substructures in anchor graphs.

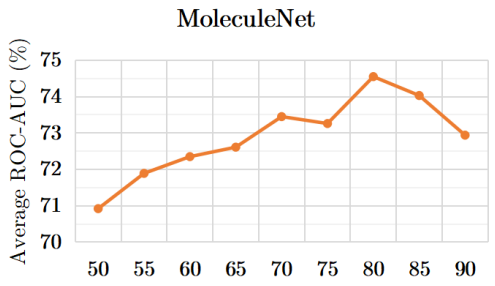
3. **Dual-view is more effective than single-view.** With the advantage of dual-view from both node and edge, SR-GCL outperforms those rationale-aware but single-view frameworks (*i.e.,* RGCL and AD-GCL), by a clear margin, which is consistent with the observation in (Chen et al., 2020): the diversity of transformations benefits contrastive learning.

### 4.2 Effectiveness of Self-attentive Rationalization

We pick out a molecule from pre-training dataset – ZINC-2M at random for visualization. Figure 5(a) and 5(b) show the original graph and corresponding attention map generated by the pre-trained model, respectively. To be specific, the aforementioned parameterized probability value $p(v_i|g)$ and $p(e_{jk}|g)$, where $v_i$ is the $i$-th node and $e_{jk}$ is the edge between node $j$ and $k$ in 5(a), correspond to entry $(\langle \text{CLS} \rangle, i)$ and $(j, k)$ in 5(b). To make it more intelligible, the attention weights are mapped to the molecule graph in 5(c), on top and bottom of which are node- and edge-wise view (*i.e.,* $p(v|g)$ and $p(e|g)$), respectively. 5(d) gives the calculation of each atom to Gasteiger partial charge (top) and partition coefficient (bottom), which indicate the asymmetric distribution of electrons in chemical bonds and the lipophilicity/hydrophilicity (red/blue shades) of the molecule, respectively. The shade of each atom reflects its impact on chemical properties – instance-discriminative power. The comparison of 5(c) and 5(d) shows high precision of our proposed self-attentive rationalization, justifying its reliability. We also provide more visualization results in Appendix F.




(a) $\tau$




(b) $\rho$ (%)

Figure 6: Sensitivity *w.r.t.* hyperparameters $\tau$ and $\rho$.

### 4.3 Ablation Study

As shown in Table 2, we ablate on the structure of SR-GCL to showcase:

1. **Effectiveness of rationale-aware augmentation.** Hybrid model based GraphCL (You et al., 2020) can be viewed as the vanilla GCL method with random augmentations. The performance margin between SR-GCL and GraphCL demonstrates the crucial role of rationale-aware augmentations to reveal instance-discriminative features in anchor graphs.
2. **Advantage of diverse views.** SR-GCL node-/edge-wise conducts rationale-aware augmentations but has only node-/edge-wise views. The complete SR-GCL with dual-views outperforms them, verifying the advantage brought by the diversity of views in GCL.

### 4.4 Hyperparameter Sensitivity

The average performances of SR-GCL on 8 MoleculeNet datasets *w.r.t.* hyperparameters $\tau$ and $\rho$ are presented in Figure 6. The sharp pike in 6(a) indicates the close relationship between SR-GCL's performance and hyperparameters $\tau$ (Wang & Liu, 2021). Thus, we suggest to tune it carefully around $0.1$. As for hyperparameter controlling the scale of augmented views, we set $\rho_v = \rho_e = \rho$ to make the scales of each contrastive pair close. The results show that SR-GCL achieves the best performance when $\rho$ is set to a large value around $80\%$, which is consistent with Li et al. (2022).

## 5 Conclusion

In this paper, we draw inspiration from the self-attention mechanism in transformers and propose a novel framework for transformer-based graph models, Self-attentive Rationale guided Graph Contrastive Learning (SR-GCL), which reshape the separated design of rationale finder and encoder in existing rationale-aware GCL frameworks. In SR-GCL, the self-attention values in transformer module are leveraged to guide the construction of both node- and edge-wise rationale-aware contrastive pairs, making (1) the graph transformations diverse, and (2) the rationale finder naturally integrated with encoder. Visual inspections on real-world molecule datasets demonstrate the effectiveness of self-attentive rationalization proposed by us and extensive experiments on downstream tasks show that SR-GCL empowers the backbone model with the ability of yielding instance-discriminative representations, thus setting the new state-of-the-art for graph pre-training.

**Limitations.** The application of SR-GCL is constrained in graph-level tasks and we will explore the possibility to adapt self-attentive rationalization in node-level tasks, in which graphs usually contain hundreds of nodes or more. Concretely speaking, vanilla transformers have a $O(N^2)$ space complexity *w.r.t.* the number of nodes in a graph, making its application limited to graphs with dozens of nodes. The prepositional obstacle of applying self-attentive mechanism in node-level tasks is to study the graph transformer models with linear space complexity. Meanwhile, it's also promising to investigate the rationale discovery in graph pre-training frameworks of context prediction (He et al., 2022; Tan et al., 2022; Hou et al., 2022).

## REFERENCES


Shiyu Chang, Yang Zhang, Mo Yu, and Tommi S. Jaakkola. Invariant rationalization. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1448–1458. PMLR, 2020.

Dexiong Chen, Leslie O'Bray, and Karsten M. Borgwardt. Structure-aware transformer for graph representation learning. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3469–3489. PMLR, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699, 2020.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pp. 1024–1034, 2017.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *KDD*, pp. 594–604. ACM, 2022.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *ICLR*, 2020.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *NeurIPS*, pp. 21618–21629, 2021.

Greg Landrum. Rdkit, open-source cheminformatics software, 2010. URL https://www.rdkit.org/.

Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 13052–13065. PMLR, 2022.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *ICLR*. OpenReview.net, 2022.

Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers. *CoRR*, abs/2106.05667, 2021.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: graph contrastive coding for graph neural network pre-training. In *KDD*, pp. 1150–1160. ACM, 2020.

Ladislav Rampásek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *CoRR*, abs/2205.12454, 2022.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In *NeurIPS*, 2020.

Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. In *NeurIPS*, pp. 15920–15933, 2021.

Qiaoyu Tan, Ninghao Liu, Xiao Huang, Rui Chen, Soo-Hyun Choi, and Xia Hu. MGAE: masked autoencoders for self-supervised learning on graphs. *CoRR*, abs/2201.02534, 2022.

Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, pp. 37, 2014.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pp. 5998–6008, 2017.

Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *ICLR (Poster)*. OpenReview.net, 2019.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*, pp. 2495–2504. Computer Vision Foundation / IEEE, 2021.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. In *International Conference on Learning Representations*, 2021.

Zhanghao Wu, Paras Jain, Matthew A. Wright, Azalia Mirhoseini, Joseph E. Gonzalez, and Ion Stoica. Representing long-range context for graph neural networks with global attention. In *NeurIPS*, pp. 13266–13279, 2021.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018a.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pp. 3733–3742. Computer Vision Foundation / IEEE Computer Society, 2018b.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11548–11558. PMLR, 2021.

Xinyi Xu, Cheng Deng, Yaochen Xie, and Shuiwang Ji. Group contrastive self-supervised learning on graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *NeurIPS*, pp. 28877–28888, 2021.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *NeurIPS*, 33:5812–5823, 2020.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12121–12132. PMLR, 2021.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *CoRR*, abs/2006.04131, 2020.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *WWW*, pp. 2069–2080, 2021.

# A ALGORITHM OF SR-GCL

**Algorithm 1** SR-GCL algorithm

**Initialize:** dataset $\{g_m : m = 1, 2, ..., M\}$, GNN module $f_1(\cdot)$, transformer module $f_2(\cdot)$, projector $h(\cdot)$, sampling ratio $\rho_v$ & $\rho_e$ and temperature $\tau$.
**for all** sampled minibatch of data $\{g_n : n = 1, 2, ..., N\}$ **do**
  **for** $n = 1$ **to** $N$ **do**
    $\mathbf{H} = f_1(g_n), \quad \mathbf{X} = [\mathbf{x}_{\text{cls}}, \mathbf{WH}]$ ▷ equation 8
    $\{p(v|g_n) = \frac{\kappa(\mathbf{x}_{\text{cls}}, \mathbf{x}_v)}{\sum_{u \in \mathcal{V}} \kappa(\mathbf{x}_{\text{cls}}, \mathbf{x}_u)} : v \in \mathcal{V}\}$ ▷ equation 3
    $\{p(e|g_n) = \frac{\kappa(\mathbf{x}_{u_e}, \mathbf{x}_{v_e})}{\sum_{e' \in \mathcal{E}} \kappa(\mathbf{x}_{u_{e'}}, \mathbf{x}_{v_{e'}})} : e \in \mathcal{E}\}$ ▷ equation 3
    $R_v(g) \sim P_{\mathcal{V}}(\cdot|G = g_n)) \quad \text{s.t.} \ |\mathcal{V}_R| = \rho_v \cdot |\mathcal{V}|$ ▷ equation 6
    $R_e(g) \sim P_{\mathcal{E}}(\cdot|G = g_n) \quad \text{s.t.} \ |\mathcal{E}_R| = \rho_e \cdot |\mathcal{E}|$ ▷ equation 7
    $\mathbf{r}'_n = h(f_2(f_1(R_v(g)))), \quad \mathbf{r}''_n = h(f_2(f_1(R_e(g))))$
  **end for**
  **for** $n = 1$ **to** $N$ **do**
    $\mathcal{R}_n^- = \{\mathbf{r}'_i, \mathbf{r}''_i : i = 1, 2, ..., n-1, n+1, ..., N\}$
    $l^n = -\log \frac{\exp\left(\mathbf{r}'^{\top}_n \mathbf{r}''_n / \tau\right)}{\exp\left(\mathbf{r}'^{\top}_n \mathbf{r}''_n / \tau\right) + \sum_{\mathbf{r}^- \in \mathcal{R}_g^-} \exp\left(\mathbf{r}'^{\top}_n \mathbf{r}^- / \tau\right)}$ ▷ equation 12
  **end for**
  Update $f(\cdot)$, $r(\cdot)$, and $h(\cdot)$ to minimize $\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} l^n$
**end for**
**return** Encoder $f_2 \circ f_1$

# B DATASETS

Table 3: Statistics for ZINC-2M and MoleculeNet datasets.

| DATASETS | CATEGORY | UTILIZATION | GRAPHS# | AVG. N# | AVG. DEGREE |
|---|---|---|---|---|---|
| ZINC-2M | BIOCHEMICAL MOLECULES | PRE-TRAINING | 2,000,000 | 26.62 | 57.72 |
| BBBP | BIOCHEMICAL MOLECULES | FINETUNING | 2,039 | 24.06 | 51.90 |
| TOX21 | BIOCHEMICAL MOLECULES | FINETUNING | 7,831 | 18.57 | 38.58 |
| TOXCAST | BIOCHEMICAL MOLECULES | FINETUNING | 8,576 | 18.78 | 38.52 |
| SIDER | BIOCHEMICAL MOLECULES | FINETUNING | 1,427 | 33.64 | 70.71 |
| CLINTOX | BIOCHEMICAL MOLECULES | FINETUNING | 1,477 | 26.15 | 55.76 |
| MUV | BIOCHEMICAL MOLECULES | FINETUNING | 93,087 | 24.23 | 52.55 |
| HIV | BIOCHEMICAL MOLECULES | FINETUNING | 41,127 | 25.51 | 54.93 |
| BACE | BIOCHEMICAL MOLECULES | FINETUNING | 1,513 | 34.08 | 73.71 |

# C BASELINES

We compare with following baselines to verify the state-of-the-art performance of our SR-GCL:

- **Infomax** (Velickovic et al., 2019) Infomax conducts mutual information maximization between patches (*i.e.,* subgraphs centered around nodes of interest) and high-level graphs.
- **EdgePred** (Hamilton et al., 2017) EdgePred pre-trains a backbone model by generating node representations of previously unseen data with text attributes to make edge predictions.

- **Attribute Masking** (Hu et al., 2020) Attribute Masking masks attributes of node/edge and make backbone to predict them based on their neighboring structure.
- **Context Prediction** (Hu et al., 2020) Context Prediction trains the backbone to map nodes appearing in similar contexts to nearby representations by predicting with subgraphs.
- **GraphCL** (You et al., 2020) GraphCL is a vanilla GCL method, constructing augmented views with strategies of node dropping, edge perturbation or attribute masking in a random fashion.
- **GraphLoG** (Xu et al., 2021) GraphLoG preserves the local similarity and leverages prototypes to capture the clusters with global semantics.
- **AD-GCL** (Suresh et al., 2021) With a GNN-based augmenter, AD-GCL adopts adversarial graph augmentation strategies to minimize redundant information in GCL.
- **JOAO** (You et al., 2021) JOAO automatically selects data augmentations from a strategy pool to construct contrastive pairs for GCL.
- **RGCL** (Li et al., 2022) RGCL hires an auxiliary model to reveal semantically important nodes to construct augmented views for contrastive optimization.

## D Backbone Model Architectures

Table 4: Backbone model architectures

| Models | GNN in baselines | Hybrid model |
|---|---|---|
| GNN type | GIN | GIN |
| Neuron# | [300,300,300,300,300] | [300,300,300,300,300] |
| Trans in/out dimension | - | 128 |
| Trans projection dimension | - | 512 |
| Trans layer# | - | 4 |
| Trans heads# | - | 4 |
| Pooling | mean | ⟨CLS⟩ |
| Projector Neuron# | [300,300] | [128,128] |
| Optimizer | Adam | AdamW |
| Learning rate | 0.001 | 0.0001 |

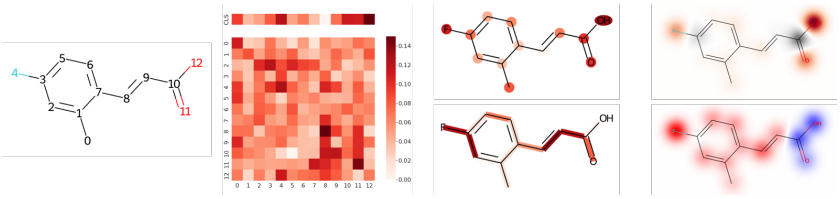
## E Computational Efficiency

To compare the computational efficiency of commonly used GIN model in existing baselines with hybrid model, we evaluate the number of parameters and the propagation runtime for each mini-batch of 256 samples on RTX 3090. The statistics in Table 5 show that the transformer module adds an overhead of $46.63\%$ to whole contrastive learning procedure and $49.91\%$ to inference (*i.e.,* forward) time, which we argue, considering the great improvement of representation ability and interpretability, is acceptable.

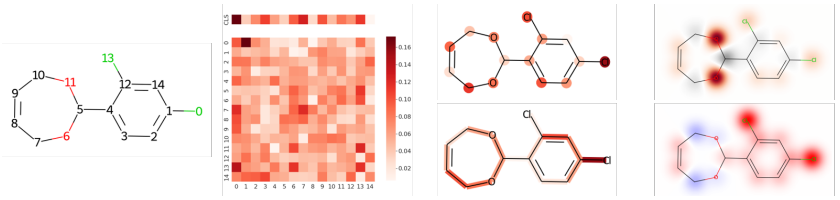
| Backbone | Forward time (ms) | Backward time (ms) | Cumulative time (ms) | Parameter # |
|---|---|---|---|---|
| GIN | $15.55_{\pm 9.72}$ | $22.49_{\pm 2.27}$ | $38.04_{\pm 10.07}$ | 1.88M |
| Hybrid Model | $23.97_{\pm 11.39}$ | $36.51_{\pm 3.32}$ | $60.47_{\pm 11.95}$ | 2.71M |

Table 5: Computational efficiency

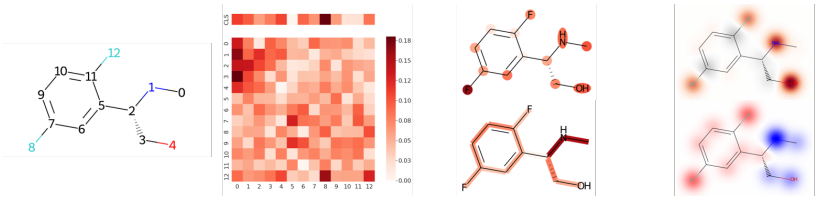
## F SAMPLES OF SELF-ATTENTIVE RATIONALIZATION



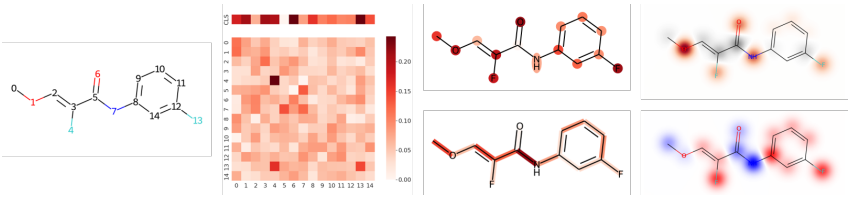
(a) Cc1cc(F)ccc1/C=C/C(=O)O



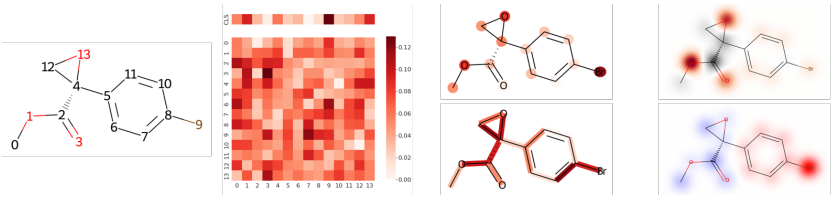
(b) Clc1ccc(C2OCC=CCO2)c(Cl)c1



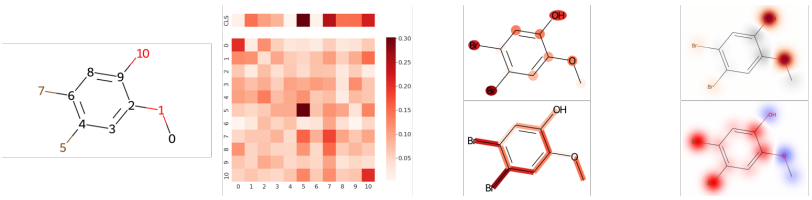
(c) CN[C@H](CO)c1cc(F)ccc1F



(d) CO/C=C(\F)C(=O)Nc1cccc(F)c1



(e) COC(=O)[C@]1(c2ccc(Br)cc2)CO1



(f) COc1cc(Br)c(Br)cc1O

Figure 7: Visualization of more molecules