CURRICULUM-AWARE TRAINING FOR DISCRIMINATING MOLECULAR PROPERTY PREDICTION MODELS

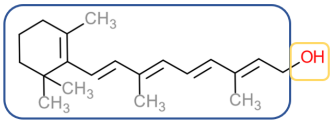
Anonymous authors
Paper under double-blind review

ABSTRACT

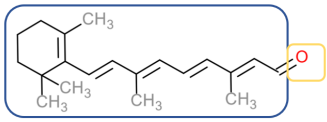
Despite their wide application across various fields, current molecular property prediction models struggle with the challenge of activity cliff, which refers to the situation where molecules with similar chemical structures display remarkable different properties. This phenomenon hinders existing models' ability to learn distinctive representations for molecules with similar chemical structures, and results in inaccurate predictions on molecules with activity cliff. To address this limitation, we first present empirical evidence demonstrating the ineffectiveness of standard training pipelines on molecules with activity cliff. We propose a novel approach that reformulates molecular property prediction as a node classification problem, introducing two innovative tasks at both the node and edge levels to improve learning outcomes for these challenging molecules with activity cliff. Our method is versatile, allowing seamless integration with a variety of base models, whether pre-trained or randomly initialized. Extensive evaluation across different molecular property prediction datasets validate the effectiveness of our approach.

1 INTRODUCTION

Molecular property prediction aims to determine the properties of specific molecules directly from the chemical structures. It plays a crucial role in various fields, including drug discovery (Stokes et al., 2020), material science (Chanussot et al., 2021; Tran et al., 2022) and bioinformatics (Narayanan et al., 2002; Zhou et al., 2023). Despite its broad application, recent studies (van Tilborg et al., 2022; Deng et al., 2023) reveal that current models often fail to generate sufficiently discriminative molecular representation, and sometimes even perform worse than models with fixed representation (e.g., molecular fingerprints). Such limitation arises as existing machine learning models tend to produce similar representations for chemically similar molecules. When two molecules with analogous structures exhibit different properties, accurately predicting their properties becomes particularly challenging due to the indistinguishable representations. Such phenomenon is commonly referred as activity cliff (AC) (Stumpfe et al., 2019; Tamura et al., 2023; Dablander et al., 2023), which is prevalent across various molecular property datasets. An example from the Tox21 data set (Wu et al., 2018) is shown in Figure 1. Here, the two molecules only have minor differences (shown in the two yellow boxes), but their responses to ER, ATAD5 and HSE receptors are all different.



(a) Retinol.



(b) Retinal.

Figure 1: Examples of two molecules with AC

While numerous studies (Maggiora, 2006; van Tilborg et al., 2022; Graff et al., 2023; Deng et al., 2023) have verified the intuition that AC causes difficulty for existing molecular property prediction models, their analysis only focuses on the *inference* stage, and it remains unclear why existing models fail to learn discriminating molecular representation in the *training* stage. Similar to the inference stage, training a model to differentiate between structurally similar molecules with distinct properties inherently presents challenges. Nevertheless, no existing work considers how to address such challenges, and standard training pipelines only lead to models without enough abilities to distinguish molecules with AC.

Motivated by the shortcoming of existing training algorithms in obtaining discriminative molecular representation, in this paper, we propose a novel training algorithm to enhance learning from molecules with AC. Through extensive empirical analysis, we first demonstrate that standard training algorithms struggle to accurately fit molecules with AC during the training phase, and this challenge persists across different model backbones and pre-training tasks. In response, we propose a new training algorithm that focuses on improving model's discriminative power by effectively learning from molecules with AC. Our method reformulates molecular property prediction as a node classification problem on graph, where each node represents a molecule, and edges are defined by similarities in their chemical structures. Then we introduce two tasks at the node and edge levels respectively. For the node-level task, we employ a curriculum learning approach that considers both loss and AC information to select informative molecules for model training. For the edge-level task, we introduce a novel pairwise modeling task to align the model directly with AC on different molecular properties. The proposed method can be integrated with different base models, including both random-initialized and pre-trained models. Empirical results on diverse molecular property prediction data sets demonstrate the effectiveness of our proposed method.

Our contributions are summarized as follows:

- We are the first to investigate why existing molecular property prediction models fail to produce discriminative molecular representation. Using molecules with AC as representatives, our results reveal that standard training pipelines struggle to accurately fit these molecules, a limitation observed in both randomly initialized and pre-trained models.
- We propose to re-formulate molecular property prediction as a node classification problem. Under this formulation, we introduce two novel tasks on node and edge levels respectively to learn from molecules with AC more effectively and produce models with enough discriminative ability.
- Empirical results on diverse molecular property data sets demonstrate that our proposed method improves the performance for both random-initialized and pre-trained models.

2 Related Works

2.1 Molecular Property Prediction with Graph Neural Networks

Molecular property prediction predicts the molecular properties from a molecular graph, in which each node is an atom and each edge is a chemical bond between atoms. Naturally, various graph learning architectures can be applied to this task. Pioneering works (Merkwirth & Lengauer, 2005; Gilmer et al., 2017) use the message-passing graph neural networks (GNN) (Velickovic et al., 2017; Xu et al., 2018). However, the GNN may not be able to capture long-range dependencies (Rampášek et al., 2022). Instead, recently, transformer models (Vaswani et al., 2017) are used to model long-range interactions between nodes (Ying et al., 2021; Rampášek et al., 2022). On the quantum-chemical regression task, EGT (Hussain et al., 2022) uses global self-attention to update both the node and edge representations. This allows unconstrained dynamic long-range interactions between nodes, and results in better performance.

Despite the use of different architectures, another way to improve performance is by using different graph pre-training tasks. Most of these works consider how to effectively use the geometric information contained in the 3D conformers of different molecules (Townshend et al., 2019; Axelrod & Gomez-Bombarelli, 2022). For example, Klicpera et al. (2020) uses the relative 3D information (such as bond length and bond angle) derived from the absolute Cartesian coordinates. GemNet (Gasteiger et al., 2021) further captures information from the dihedral angle to uniquely define all relative atom positions. SphereNet (Liu et al., 2021) proposes a generic framework for the 3D graph network, and designs a spherical message passing mechanism. 3D Infomax (Stärk et al., 2022) proposes to maximize mutual information between the 3D structures and representations from the GNN, enabling the model to produce implicit 3D information that can be useful for the downstream tasks. 3D-PGT (Wang et al., 2023b) proposes a multi-task 3D pre-training framework, that predicts bond length, bond angle and dihedral angle from molecular graphs. UniMol (Zhou et al., 2023) proposes to jointly use the 3D position recovery task and masked atom prediction task for pre-training, and achieves state-of-the-art performance on various molecular property prediction benchmarks.

The negative impacts of AC to molecular property prediction have long been investigated (Maggiora, 2006; van Tilborg et al., 2022; Graff et al., 2023; Deng et al., 2023). However, they focus on the inference stage, while we propose to confront such negative impacts with a novel training algorithm. Some other works (Horvath et al., 2016; Iqbal et al., 2021; Park et al., 2022; Zhang et al., 2023; Wu, 2024) predict whether a given pair of molecules have AC. These works focus on different application as we consider the original molecular property prediction problem.

2.2 Curriculum Learning

Generally, curriculum learning (CL) (Wang et al., 2022) first trains a learning model with easier training samples, so that the model can easily obtain a coarse decision boundary. The model is then refined by harder samples later in the training process. As an easy-to-use plug-in, curriculum learning can improve generalization performance of various models in a wide range of scenarios, including computer vision (Guo et al., 2018), natural language processing (Platanios et al., 2019; Liu et al., 2020) and reinforcement learning (Narvekar et al., 2017).

Curriculum learning has also been applied to graph learning (Wei et al., 2022; Wang et al., 2023a). CLNode (Wei et al., 2022) proposes to jointly consider the loss and node labels in algorithm design for curriculum learning on node classification. MotifNet (Wang et al., 2023a) uses curriculum learning for motif-based graph learning, and sorts different motifs based on their difficulty levels. CurrMG (Gu et al., 2022) further considered using curriculum learning in molecular property prediction. Nevertheless, their approach only yields limited improvements as they only consider the prediction error and molecular structure for each molecule separately, and our method considers the pairwise relation between molecules.

3 Case Studies on Molecules with Activity Cliff

To see how existing models suffer from limited abilities to distinguish molecules with similar chemical structures, we take the set of molecules with AC as examples. Loosely speaking, AC refers to a pair of molecules with similar structures but distinct properties. Its exact definition depends on how we characterize structural similarity. In the following, we build upon the definition of matched molecule pairs (Dablander et al., 2023).

Definition 3.1 (Matched Molecule Pair: Dablander et al. (2023))**.** A *matched molecule pair* is a pair of molecules that share a common *structural core* (which contains at least twice as many heavy atoms[1] as in the variable parts) but differ by small *variable parts* (which contains no more than 13 heavy atoms) from the chemical transformation of bond cutting on exocyclic bonds.

The definition of AC can then be given as follows:

Definition 3.2 (Activity Cliff (AC))**.** Activity cliff refers to a matched molecule pair with different labels with respect to a given property.

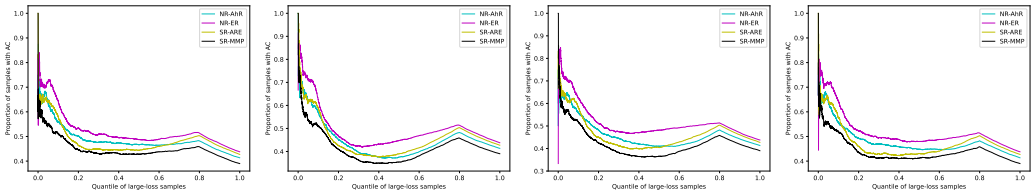
Note that the definition of AC depends on the property being considered. For a pair of molecules with similar chemical structures, it may exhibit activity cliff for one property but not another.

While many works have demonstrated the difficulty of making accurate predictions on molecules with AC (Maggiora, 2006; van Tilborg et al., 2022; Graff et al., 2023; Deng et al., 2023), it remains unclear why such difficulty arises, and why existing models cannot produce discriminating representations on these molecules. To empirically investigate these issues, we consider four tasks from the Tox21 data set (Wu et al., 2018) that predict a molecule's response to different receptors (AhR, ER, ARE and MMP receptors). We use two models that are very commonly used for molecular property prediction: GIN (Xu et al., 2018) as a representative for message-passing neural networks, and GraphGPS (Rampášek et al., 2022) as a representative for attention-based graph learning models. Despite training GIN or GraphGPS models from scratch, we also include two recent state-of-the-art pre-training models: 3D-PGT (Wang et al., 2023b) and Uni-Mol (Zhou et al., 2023), which both use attention-based graph learning models similar to GraphGPS.

Figure 2 first shows the proportion of molecules with AC among molecules with the top-$n\%$ loss values. While only about 40% of all samples in the Tox21 data set have activity cliff (Table 9

[1]Heavy atoms are atoms other than hydrogen.

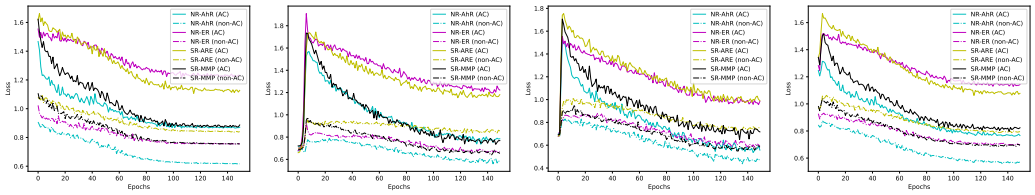
in Appendix B), **molecules with AC make up a significantly larger proportion of large-loss molecules** (about 60% in samples with the top-10% loss). In other words, activity cliff is a critical source for samples that are not accurately learnt, which also indicates the inability of current models to distinguish structural similar molecules.



(a) GIN. (b) GraphGPS. (c) Pre-trained 3D-PGT. (d) Pre-trained Uni-Mol.

Figure 2: Proportion of molecules with AC among molecules with top-$n\%$ loss values.

Figure 3 shows the average training loss for the top 10%-loss molecules with and without AC. We can see that **even for these "hard" molecules, molecules with AC have significantly larger training losses** than those without AC. Moreover, the average loss on molecules with AC is still way larger than zero even near convergence, which indicates these models do not even fit these molecules well with standard training pipelines.



(a) GIN. (b) GraphGPS. (c) pre-trained 3D-PGT. (d) pre-trained Uni-Mol.

Figure 3: Training losses of large-loss molecules with and without activity cliffs in four model training setups.

While difficult to train on, molecules with AC can also be useful for learning an accurate decision boundary: to accurately classify a pair of molecules with activity cliff, the decision boundary is expected to separate them even though they have very similar chemical structures. As such, we conduct an experiment that only selects molecules with AC (denoted as "AC") or only selects molecules without AC (denoted as "non-AC") as training samples. We use both randomly initialized GraphGPS model (Rampášek et al., 2022) and 3D-PGT pretrained model (Wang et al., 2023b). The ROC-AUC metrics on different data sets are shown in Table 1. As expected, selecting only molecules without AC generally yields worse performances than training on all molecules. Selecting molecules with AC leads to some improvements the Tox21 and ToxCast data set, but the improvements are still limited, as we ignore information from molecules without AC.

Table 1: ROC-AUC on different molecular property prediction data sets when only using molecules with/without AC for training.

Method	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
GraphGPS	71.5	68.5	56.4	66.9	76.9	**67.0**	**71.1**	**77.0**
GraphGPS (AC)	**71.8**	**69.2**	**56.5**	68.8	**77.6**	**67.0**	67.8	72.2
GraphGPS (non-AC)	67.8	66.9	56.3	**69.2**	75.8	66.3	67.4	74.8
3D PGT	73.8	69.2	**60.6**	**69.4**	**80.9**	**72.1**	**79.4**	**69.4**
3D PGT (AC)	**74.0**	**70.1**	59.7	67.3	79.9	68.6	69.1	68.7
3D PGT (non-AC)	68.6	68.9	58.6	64.6	79.1	65.7	77.3	69.1

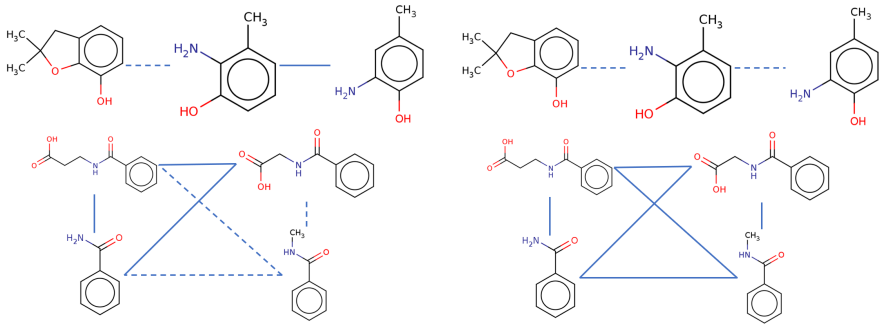
Since molecules with AC are both difficult and useful for model training, in the next section, we propose a more effective approach to learn from molecules with activity cliff and obtain a molecular property prediction model with discriminating molecular representation.

4 EFFECTIVE LEARNING FROM SAMPLES WITH ACTIVITY CLIFF

Based on the observations in section 3, we propose a novel training algorithm to effectively learn from molecules with AC for more discriminative molecular representation. We first propose to formulate molecular property prediction as a node classification problem, with the graph structure induced by structural similarity in section 4.1. In section 4.2, we propose a novel sample selection method to gradually select hard molecules with AC for training. We further propose a novel edge-level task to align the model with AC on different properties in section 4.3.

4.1 MOLECULAR PROPERTY PREDICTION AS NODE CLASSIFICATION

Given a set of molecules, the definition of matched molecule pairs (Definition 3.1) naturally induces a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In this graph, each molecule corresponds to a node, whose features correpond to molecular representation obtained by pre-trained models. Two nodes (molecules) are connected if they are a matched molecule pair. For example, Figure 4 shows the subgraph (for 7 molecules) based on two property prediction tasks from the *Tox21* data set: predicting a molecule's response to the ARE and MMP receptors. As mentioned in section 3, a matched molecule pair may have activity cliff for one property (dashed edges in Figure 4) but not for the other (solid edges). The graph $\mathcal{G}$ allows us to formulate molecular property prediction as a node classification problem, where the node labels describe the properties of different molecules. Such graph formulation is different from similar ideas in literature (Zhuang et al., 2023; Zhao et al., 2024) as they did not conisder the AC information inside the graph formulation, reflected by different types of edges in Figure 4.



(a) Responses to ARE receptors. (b) Responses to MMP receptors.

Figure 4: Visualization of the graph structure. Edges (dashed and thick lines) between two molecules show that they have similar structures as defined in Definition 3.1. Dashed lines indicate they have different properties (as in the subfigure captions).

4.2 NODE-LEVEL TASK FOR MOLECULES WITH ACTIVITY CLIFF

Since molecules with AC are more difficult to learn from (section 3), we consider the use of curriculum learning (Wang et al., 2022), which first selects easier samples and then harder samples to gradually train a better model. However, from Figure 3, even for molecules with similar losses, molecules with AC are still more difficult to learn than molecues without AC. As such, we propose a weighted curriculum learning algorithm that jointly considers AC and molecule loss. Specifically, we define a weighted loss $\hat{\ell}_i(\boldsymbol{w})$ for a given molecule i as $\hat{\ell}_i(\boldsymbol{w}) = p_i \ell_i(\boldsymbol{w})$, where $\ell_i(\boldsymbol{w})$ denotes the original loss on molecule i (e.g., cross-entropy loss for classification tasks, or squared loss for regression tasks), and p_i is the weight on molecule i defined as:

$$p_i = \begin{cases} 1 & \text{molecule } i \text{ has activity cliff} \\ p & \text{molecule } i \text{ does not have activity cliff} \end{cases} \tag{1}$$

with $p < 1$ (i.e., molecules with AC have higher weights than those without AC). Thus, when two samples have the same loss values (i.e., equally difficult for the model), we select molecules with AC first. At iteration t, let the sampled mini-batch be $\mathcal{B}$. We select a subset $\hat{\mathcal{B}}$ of large-loss samples in $\mathcal{B}$:

$$\hat{\mathcal{B}}(\boldsymbol{w}) = \{i | i \in \mathcal{B}, \hat{\ell}_i(\boldsymbol{w}) \geq R(t) \text{ percentile of loss in } \mathcal{B}\}.$$

In other words, $R(t)$ controls the percentage of easy molecules that are discarded at iteration t, as we focus more on hard molecules that are not fit well. The loss on $\hat{\mathcal{B}}$, namely, $\mathcal{L}(\boldsymbol{w}; \hat{\mathcal{B}}(\boldsymbol{w})) = \frac{1}{|\hat{\mathcal{B}}(\boldsymbol{w})|} \sum_{i \in \hat{\mathcal{B}}(\boldsymbol{w})} \hat{\ell}_i(\boldsymbol{w})$, is then used to update the model, which allows the model to gradually focus more on difficult molecules with AC that are more useful for making accurate prediction.

4.3 Edge-level Task for Activity Cliff Pairs

While the aforementioned sample selection method can better learn from molecules with AC, it only considers each molecule separately. However, AC is defined for a pair of molecules, and they may affect the predictions of each other. As such, we propose to also introduce an edge-level task: Specifically, for each edge $e_{ij} = (v_i, v_j)$ in $\mathcal{G}$, we define the loss:

$$\ell_{e_{ij}}(\boldsymbol{w}) = -(y_i - y_j)(\hat{y}_i(\boldsymbol{w}) - \hat{y}_j(\boldsymbol{w})). \tag{2}$$

For classification tasks, when we have $y_i = y_j$, it indicates that these two molecules have the same label and does not form an AC pair. Otherwise, if $y_i \neq y_j$, we must have $y_i - y_j = \pm 1$. When $y_i = 1$ (i.e., $y_i - y_j = 1$ and $y_j = 0$), (2) minimizes $-(\hat{y}_i - \hat{y}_j)$, which corresponds to maximizing $\hat{y}_i$ and minimizing $\hat{y}_j$. When $y_i = 0$ (i.e., $y_i - y_j = -1$ and $y_j = 1$), (2) minimizes $(\hat{y}_i - \hat{y}_j)$, which corresponds to minimizing $\hat{y}_i$ and maximizing $\hat{y}_j$. Similar deduction can be obtained for regression tasks as well, and we also draw the predictions of both molecules with activity cliff towards the ground truth. The total edge-level loss over all matched molecule pairs can then be written as:

$$\mathcal{L}_e(\boldsymbol{w}; \mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum\nolimits_{e_{ij} \in \mathcal{A}} \ell_{e_{ij}} = \frac{1}{|\mathcal{A}|} \sum\nolimits_{e_{ij} \in \mathcal{A}} -(y_i - y_j)(\hat{y}_i(\boldsymbol{w}) - \hat{y}_j(\boldsymbol{w})), \tag{3}$$

where $\mathcal{A} \subset \mathcal{E}$ is the set of all matched molecule pairs. The following Proposition further expresses the gradient of the edge-level loss $\mathcal{L}_e(\boldsymbol{w}; \mathcal{A})$ in terms of the $\frac{\partial \hat{y}_i(\boldsymbol{w})}{\partial \boldsymbol{w}}$ for each molecule i.

Proposition 4.1. $\frac{\partial \mathcal{L}_e(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{1}{|\mathcal{A}|} \sum_i -n_i(2y_i - 1)\frac{\partial \hat{y}_i(\boldsymbol{w})}{\partial \boldsymbol{w}}$, *where n_i is the number of AC pairs containing molecule i.*

In other words, the gradient of $\mathcal{L}_e$ is a weighted sum of $\frac{\partial \hat{y}_i(\boldsymbol{w})}{\partial \boldsymbol{w}}$'s. The weight on each $\frac{\partial \hat{y}_i(\boldsymbol{w})}{\partial \boldsymbol{w}}$ depends on the number of AC pairs containing molecule i, which does not change throughout training. Nevertheless, different AC pairs may not be equally important for learning discriminative molecular representation. Some pairs are easily separated, while other pairs may be more difficult to distinguish. Therefore, we propose to also employ curriculum learning into the edge-level task, and change the edge loss in (3) to:

$$\mathcal{L}_e(\boldsymbol{w}; \hat{\mathcal{A}}) = \frac{1}{|\hat{\mathcal{A}}|} \sum\nolimits_{e_{ij} \in \hat{\mathcal{A}}} \ell_{e_{ij}}(\boldsymbol{w}) = \frac{1}{R(t)|\mathcal{A}|} \sum\nolimits_{e_{ij} \in \hat{\mathcal{A}}} \ell_{e_{ij}}(\boldsymbol{w}), \tag{4}$$

where $\hat{\mathcal{A}} = \{e_{ij} | e_{ij} \in \mathcal{A}, \ell_{e_{ij}} \geq R(t)$ percentile of loss in $\mathcal{A}\}$. Using $\hat{\mathcal{A}}$ instead of $\mathcal{A}$ then allows us to focus more on AC pairs $e_{ij} \in \mathcal{A}$ with larger loss $\ell_{e_{ij}}$, which correspond to less well-separated pairs that are more important for model update.

Algorithm 1 Learning with Activity Cliff (LAC).

```
Initialize prediction model f with parameter w (random initialization or pre-trained weights);
for t = 0, ..., T − 1 do
    Draw a mini-batch B from molecule data set D;
    Obtain molecule pairs in B with activity cliff (denoted A)
    Determine R(t);
    Select R(t) × |B| large-loss samples B̂ from B based on network f's predictions;
    Select R(t) × |A| pairs of molecule Â with activity pairs and compute L_e in (4).
    Update w = w − η∇_w(L(w; B̂) + αL_e(w; Â));
end for
```

4.4 Complete Algorithm

The complete algorithm, which will be called Learning with Activity Cliff (LAC), is shown in Algorithm 1. Compared with standard curriculum learning algorithms (Wang et al., 2022) that may be applied to training molecular property prediction models, it has the following key differences:

- Algorithm 1 involves training on two different tasks, while existing works only consider curriculum learning on one task (namely the node-level task).
- We also propose a novel design of curriculum in Algorithm 1 based on AC information, which is unique for molecular property prediction.

Note that the proposed method can be used to train different base models, including both randomly-initialized or pre-trained models. It also introduces a hyper-parameter $R(t)$ that controls the number of large-loss samples. We will study its effect on model performance in more detail in section 5.4.

5 EXPERIMENTS

In this section, we verify the performance of proposed method on both classification data sets (section 5.1) popularly used in existing works (Stärk et al., 2022; Wang et al., 2023b; Zhou et al., 2023) and regression data sets (section 5.2) that are more common in real-world application (van Tilborg et al., 2022). Section 5.3 contains ablation studies on each component of our proposed method to verify their effectiveness, and section 5.4 studies the impact of hyper-parameters that defines $R(t)$. Section 5.5 further visualizes the loss distribution on molecules with AC for different methods, and section 5.6 provides some case studies to better understand the proposed method.

5.1 EXPERIMENTS ON CLASSIFICATION DATA SETS

We first conduct experiments on classification tasks from the MoleculeNet data set (Wu et al., 2018). The proposed method LAC is combined with the following baseline methods: (i) training from scratch with GIN (Xu et al., 2018) and GraphGPS (Rampášek et al., 2022) model, and (ii) using model initialization from the following pre-training methods: GraphMVP (Liu et al., 2022), 3D Infomax (Stärk et al., 2022), 3D-PGT (Wang et al., 2023b) and UniMol (Zhou et al., 2023). The statistics on the data sets used are in Table 9, and detailed settings of our experiments can be found in Appendix B.

The ROC-AUC on different molecular property data sets are shown in Table 2. Results demonstrate that our proposed method LAC improves the final performance for all base models considered. Using the proposed method, UniMol pre-trained model achieves the best performance on all data sets despite ToxCast, where 3D-PGT pre-trained model performs the best. Also, the improvements partially depend on the ratio of AC samples in specific data set. For example, comparing Tox21 and MUV, the improvement in Tox21 is generally larger than MUV as AC is more popular in Tox21.

Table 2: ROC-AUC on different molecular property prediction data sets. The best performance for each task is marked in bold.

Method	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
GIN	74.9	61.6	58.0	71.0	72.6	65.4	58.8	75.3
GIN+LAC	**75.6**	**62.2**	**58.3**	**72.4**	**74.8**	**65.9**	**61.6**	**76.1**
GraphGPS	71.5	68.5	56.4	66.9	76.9	67.0	71.1	77.0
GraphGPS+LAC	**74.0**	**73.7**	**60.4**	**71.3**	**82.5**	**67.7**	**72.4**	**77.6**
GraphMVP	75.9	63.1	63.9	77.7	81.2	72.4	79.1	77.0
GraphMVP+LAC	**76.7**	**70.1**	**64.5**	**78.1**	**81.6**	**72.9**	**80.2**	**77.8**
3D-PGT	73.8	69.2	60.6	69.4	80.9	72.1	79.4	69.4
3D-PGT+LAC	**75.2**	**74.0**	**61.0**	**75.1**	**84.5**	**72.4**	**79.6**	**69.5**
UniMol	79.6	69.6	65.9	82.1	85.7	72.9	91.9	80.8
UniMol+LAC	**80.2**	**72.5**	**66.2**	**82.7**	**86.4**	**73.6**	**92.2**	**80.9**

5.2 EXPERIMENTS ON REGRESSION DATA SETS

While the definition for activity cliff is more straight-forward for classification tasks as the label takes 0/1 values, recent works (van Tilborg et al., 2022; Deng et al., 2023) also consider how to define activity cliff in regression data sets. Following (van Tilborg et al., 2022), we select several data sets from the ChEMBL database (Zdrazil et al., 2023) that describe the bioactivity

values (continuous values) of different molecules to a specific target and have large proportion of molecules with activity cliff. We train a MLP model with ECFP molecular fingerprints as it performs the best on these data sets under the training pipeline in (van Tilborg et al., 2022). The mean absolute error (MAE) on different data sets are shown in Table 3. Results demonstrate that our proposed method LAC can also improve the model performance on regression tasks.

Table 3: MAE on different molecular property prediction data sets. The best performance for each task is marked in bold.

Target	5-HT1A	MOR	D3R	FXR	HRH3
ChEMBL ID	214	233	234	2047	264
MLP(ECFP)	0.692	0.845	0.669	0.796	0.672
MLP(ECFP)+LAC	**0.656**	**0.827**	**0.635**	**0.762**	**0.657**

5.3 Ablation Studies

We first compare the model performances with only the node-level task in section 4.2 or the edge-level task in section 4.3. The ROC-AUC on different data sets are shown in Table 4. We can see that both the pairwise task and curriculum learning on samples can generally improve the model performances, and curriculum learning on samples often has more significant improvements. The only exception is the MUV data set, where only using pairwise task achieves slightly worse performances. That is because MUV data set contains limited molecules with activity cliff, as is shown in Table 9 in Appendix B. Combining both components achieves the best overall performances across all data sets.

Table 4: Ablation studies on different components in the proposed method LAC. The evaluation metric is ROC-AUC (Larger is better).

Base model	node-level	edge-level	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
GraphGPS	×	×	71.5	68.5	56.4	66.9	76.9	67.0	71.1	77.0
	×	√	72.0	69.8	58.6	67.2	79.3	66.6	71.6	77.1
	√	×	73.8	73.0	59.3	69.5	81.3	67.1	72.1	77.4
	√	√	**74.0**	**73.7**	**60.4**	**71.3**	**82.5**	**67.7**	**72.4**	**77.6**
3D PGT	×	×	73.8	69.2	60.6	69.4	80.9	72.1	79.4	69.4
	×	√	74.0	70.2	60.2	69.1	81.8	69.4	77.4	68.6
	√	×	74.6	73.0	**61.0**	72.2	83.1	72.2	**79.6**	**69.5**
	√	√	**75.2**	**74.0**	**61.0**	**75.1**	**84.5**	**72.4**	**79.6**	**69.5**

Table 5 compares the model performances with different p values in (1). Setting $p = 1$ corresponds to only using the loss as difficulty measure and does not distinguish molecules with/without AC, while setting $p = 0$ corresponds to only using molecules with AC for training in Table 1. We can see that choosing $p < 1$ generally improves upon the baseline with $p = 1$ which demonstrates the effectiveness of AC information to select informative molecules for training. Nevertheless, setting p too small may still be harmful to model performance as we neglect molecules without AC. In our previous experiments, we set $p = 0.5$ as it achievs the best overall performance.

Table 5: Ablation studies on the effect of activity cliff weights for curriculum learning on samples. The evaluation metric is ROC-AUC (Larger is better).

Base model	p	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
GraphGPS	1.0	73.5	72.6	60.3	**72.9**	80.0	65.0	71.1	77.0
	0.75	73.8	72.9	60.4	72.3	81.7	67.3	71.9	77.5
	0.5	**74.0**	**73.7**	**60.4**	71.3	**82.5**	**67.7**	**72.4**	**77.6**
	0.25	71.6	70.3	57.9	69.8	77.4	67.1	70.1	73.4
	0	67.8	66.9	56.3	69.2	75.8	66.3	67.8	72.2
3D PGT	1.0	74.2	73.0	60.7	72.9	81.5	70.5	79.4	69.4
	0.75	74.7	73.7	60.9	74.6	83.8	72.1	**79.6**	69.2
	0.5	**75.2**	**74.0**	**61.0**	**75.1**	**84.5**	**72.4**	**79.6**	**69.5**
	0.25	72.4	71.9	59.2	70.4	81.3	71.8	73.6	69.1
	0	68.6	68.9	58.6	64.6	79.1	65.7	69.1	68.7

Table 6 compares the model performances on whether to use curriculum learning for pairwise task. We see that using curriculum learning for pairwise task further improves the performance than using the naive pairwise task for most data sets.

Table 6: Ablation studies on the effect of curriculum learning for pairwise task. The evaluation metric is ROC-AUC (Larger is better).

Base model	Pairwise curriculum	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
GraphGPS	×	73.0	73.2	59.7	70.6	81.6	**67.8**	71.9	77.5
	√	**74.0**	**73.7**	**60.4**	**71.3**	**82.5**	67.7	**72.4**	**77.6**
3D PGT	×	74.0	73.5	60.8	71.0	83.6	70.4	78.9	69.1
	√	**75.2**	**74.0**	**61.0**	**75.1**	**84.5**	**72.4**	**79.6**	**69.5**

Table 7: ROC-AUC on different data sets with different types of $R(t)$ schedules. 3D-PGT pre-trained model is used.

Schedule	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
linear	**75.2**	**74.0**	**61.0**	**75.1**	84.5	**72.4**	**79.6**	**69.5**
root	74.5	73.2	59.5	71.0	83.5	70.0	79.3	69.2
geometric	75.0	73.7	60.5	75.0	**85.0**	71.2	**79.6**	69.3

5.4 Impacts of $R(t)$

In this section, we empirically investigate how different choice of $R(t)$ schedules in Algorithm 1 may impact the final performance. We consider the following three different functions for $R(t)$: (i) linear: $R(t) = \lambda \min(t/(\gamma T), 1)$, (ii) root: $R(t) = \lambda \min(\sqrt{t/(\gamma T)}, 1)$, and (iii) geometric: $R(t) = \lambda(2^{\min(t/(\gamma T),1)} - 1)$. The linear function increases the difficulty of training samples at a uniform rate; the root function introduces more *hard samples* in fewer epochs, while the geometric function trains for a greater number of epochs on the subset of *easy samples*. We set $\gamma = 0.1$ and $\lambda = 0.2$ for all these schedule types, and the ROC-AUC on different data sets with different schedule types are in Table 7. Generally, linear schedule achieves the best overall performances, and geometric schedule achieves comparable performances with linear schedule (sometimes even outperforms it).

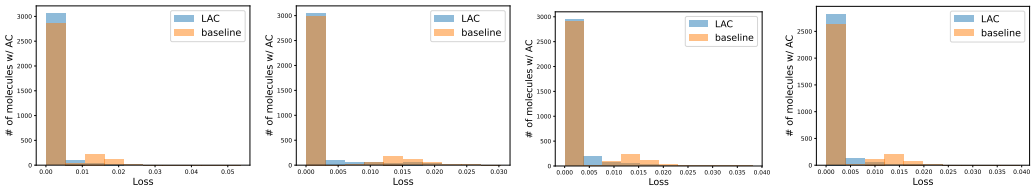
Table 8 compares the ROC-AUC on Tox21 data set using linear schedule with different hyper-parameters γ and λ. We see that the model performances are generally stable on a wide range of γ and λ values.

Table 8: ROC-AUC on Tox21 data set with different λ and γ for LAC. 3D-PGT pre-trained model is used.

	λ=0.1	λ=0.2	λ=0.3	λ=0.4	λ=0.5
γ=0.1	**75.1**	**75.2**	**75.1**	73.8	**75.2**
γ=0.2	74.0	**75.0**	**75.1**	72.7	74.3
γ=0.3	73.0	**75.0**	**75.0**	72.3	72.2
γ=0.4	73.7	73.1	74.1	72.1	72.2
γ=0.5	74.6	72.5	72.3	72.6	73.5

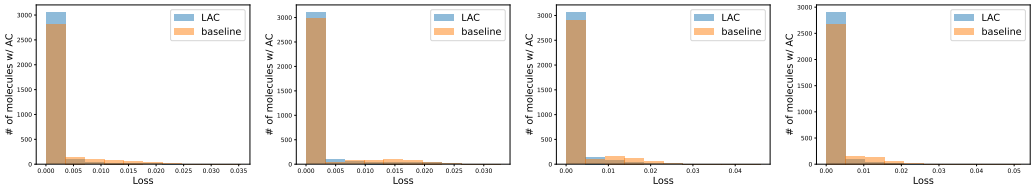
5.5 Visualize Loss Distributions for Molecules with Activity Cliff

We further visualize the final training loss distributions of all molecules with AC using models trained with the baseline training algorithm (i.e., simply training on all samples using cross-entropy loss for classification) or with our proposed algorithm LAC. For fair comparison, here we use the same cross-entropy loss on each molecule with AC. We consider using both randomly initialized GraphGPS model or a 3D-PGT pre-trained model, and the final loss distributions on molecules with AC for different tasks (properties) are shown in Figure 5 and Figure 6, respectively. We can see that models trained by the baseline algorithm (blue columns) have inaccurate predictions on part of molecules with AC, while the proposed method LAC (orange columns) can reduce the loss for these samples. LAC improves the performance for both randomly-initialized model and pre-trained model.



(a) NR-AhR. (b) NR-ER. (c) SR-ARE. (d) SR-MMP.

Figure 5: Loss distributions on molecules with AC for different tasks in Tox21 data set. Randomly initialized GraphGPS model is used.

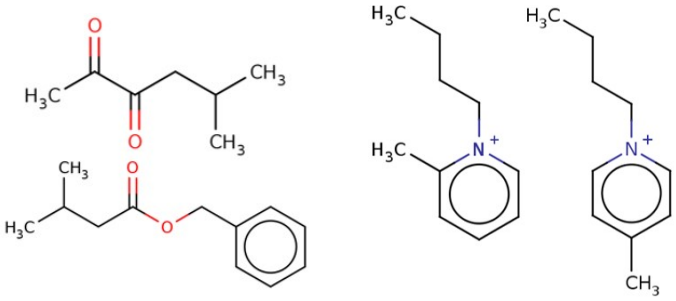


(a) NR-AhR. (b) NR-ER. (c) SR-ARE. (d) SR-MMP.

Figure 6: Loss distributions on molecules with AC for different tasks in Tox21 data set. 3D-PGT pre-trained GraphGPS model is used.

5.6 Case Studies

Finally, we choose some examples to better illustrate how the proposed method LAC can improve upon existing molecular property prediction model. We choose the UniMol pre-trained model (Zhou et al., 2023) as it overall achieves the best performance on different data sets. As in Figure 7(b), without our proposed method, UniMol cannot correctly classify molecules with AC when the structural differences are very small, even if it can handle easier pairs like in Figure 7(a). With tasks from two levels, LAC further improves the model performance to accurately classify two molecules in Figure 7(b)



(a) Example 1: correctly classified by both UniMol and UniMol+LAC

(b) Example 2: wrongly classified by UniMol but correctly classified by UniMol+LAC

Figure 7: Examples of molecules with AC. LAC improves upon existing methods to obtain more accurate predictions on molecules with AC.

6 Conclusion

In this paper, we propose to improve the performance of molecular property prediction models from the perspective of activity cliff (AC). We first use empirical results with different tasks and models to demonstrate that standard training pipeline cannot fit molecules with AC well. By re-formulating the original problem as a problem on a graph, we propose a novel training algorithm LAC that uses both node and edge-level tasks to effectively learn from molecules with AC. Extensive empirical results on different data sets demonstrate that the proposed method significantly improves the performance of different baseline methods.

REFERENCES

Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):1–14, 2022.

Lowik Chanussot, Abhishek Das, Siddharth Goyal, Weihua Lavril, et al. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, 2021.

Markus Dablander, Thierry Hanser, Renaud Lambiotte, and Garrett M. Morris. Exploring QSAR models for activity-cliff prediction. *Journal of Cheminformatics*, 15, April 2023.

Jianyuan Deng, Zhibo Yang, Hehe Wang, Iwao Ojima, Dimitris Samaras, and Fusheng Wang. A systematic study of key elements underlying molecular property prediction. *Nature Communications*, 14:6395, October 2023.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34: 6790–6802, 2021.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

David E. Graff, Edward O. Pyzer-Knapp, Kirk E. Jordan, Eugene I. Shakhnovich, and Connor W. Coley. Evaluating the roughness of structure–property relationships using pretrained molecular representations. *Digital Discovery*, 2:1452–1460, 2023.

Yaowen Gu, Si Zheng, Zidu Xu, Qijin Yin, Liang Li, and Jiao Li. An efficient curriculum learning-based strategy for molecular graph learning. *Briefings in Bioinformatics*, 23(3), April 2022.

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Computer Vision – ECCV 2018*, pp. 139–154, 2018.

Dragos Horvath, Gilles Marcou, Alexandre Varnek, Shilva Kayastha, Antonio de la Vega de León, and Jürgen Bajorath. Prediction of activity cliffs using condensed graphs of reaction representations, descriptor recombination, support vector machine classification, and support vector regression. *Journal of Chemical Information and Modeling*, 56(9):1631–1640, 2016.

Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. Global self-attention as a replacement for graph convolution. In *KDD*, pp. 655–665, 2022.

Javed Iqbal, Martin Vogt, and Jürgen Bajorath. Prediction of activity cliffs on the basis of images using convolutional neural networks. *Journal of Computer-Aided Molecular Design*, 35:1157 – 1164, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

Tao Lin, Sebastian U. Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't Use Large Mini-Batches, Use Local SGD. 2018.

Tao Lin, Lingjing Kong, Sebastian Stich, and Martin Jaggi. Extrapolation for Large-batch Training in Deep Learning. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Shengchao Liu, Hanchen Wang, Weiyang Liu, et al. Pre-training molecular graph representation with 3d geometry. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. Norm-based curriculum learning for neural machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks. *arXiv preprint arXiv:2102.05013*, 2021.

Gerald M. Maggiora. On outliers and activity cliffswhy qsar often disappoints. *Journal of Chemical Information and Modeling*, 46(4):1535–1535, 2006.

Christian Merkwirth and Thomas Lengauer. Automatic generation of complementary descriptors with molecular graph networks. *Journal of Chemical Information and Modeling*, 45(5):1159–1168, 2005.

Ajit Narayanan, Edward C Keedwell, and Björn Olsson. Artificial intelligence techniques for bioinformatics. *Applied bioinformatics*, 1:191–222, 2002.

Sanmit Narvekar, Jivko Sinapov, and Peter Stone. Autonomous task sequencing for customized curriculum design in reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2017.

Junhui Park, Gaeun Sung, SeungHyun Lee, SeungHo Kang, and ChunKyun Park. Acgcn: Graph convolutional networks for activity cliff prediction between matched molecular pairs. *Journal of Chemical Information and Modeling*, 62(10):2341–2351, 2022.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom Michael Mitchell. Competence-based curriculum learning for neural machine translation. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *arXiv preprint arXiv:2205.12454*, 2022.

Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*, pp. 20479–20502. PMLR, 2022.

Jonathan M Stokes, Kevin Yang, Kyle Swanson, , et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

Dagmar Stumpfe, Huabin Hu, and Jürgen Bajorath. Evolving concept of activity cliffs. *ACS Omega*, 4(11):14360–14368, 2019.

Shunsuke Tamura, Tomoyuki Miyao, and Jürgen Bajorath. Large-scale prediction of activity cliffs using machine and deep learning methods of increasing complexity. *Journal of Cheminformatics*, 15, January 2023.

Raphael Townshend, Rishi Bedi, Patricia Suriana, and Ron Dror. End-to-end learning on 3d protein structure for interface prediction. *Advances in Neural Information Processing Systems*, 32, 2019.

Richard Tran, Janice Lan, et al. The open catalyst 2022 (oc22) dataset and challenges for oxide electrocatalysis. *arXiv preprint arXiv:2206.08917*, 2022.

Derek van Tilborg, Alisa Alenicheva, and Francesca Grisoni. Exposing the limitations of molecular machine learning with activity cliffs. *Journal of Chemical Information and Modeling*, 62(23): 5938–5951, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017.

Daixin Wang, Zhiqiang Zhang, Yeyu Zhao, Kai Huang, Yulin Kang, and Jun Zhou. Financial default prediction via motif-preserving graph neural network with curriculum learning. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023a.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2022.

Xu Wang, Huan Zhao, Wei-wei Tu, and Quanming Yao. Automated 3d pre-training for molecular property prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 2419–2430, 2023b.

Xiaowen Wei, Xiuwen Gong, Yibing Zhan, Bo Du, Yong Luo, and Wenbin Hu. Clnode: Curriculum learning for node classification. *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2022.

Fang Wu. A semi-supervised molecular learning framework for activity cliff estimation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 6080–6088. International Joint Conferences on Artificial Intelligence Organization, 8 2024.

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*, 9:513–530, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Tie-Yan Liu, et al. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, Maria Paula Magarinos, Nicolas Bosc, Ricardo Arcila, Tevfik Kizilören, Anna Gaulton, A Patrícia Bento, Melissa F Adasme, Peter Monecke, Gregory A Landrum, and Andrew R Leach. The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, 11 2023.

Ziqiao Zhang, Bangyi Zhao, Ailin Xie, Yatao Bian, and Shuigeng Zhou. Activity Cliff Prediction: Dataset and Benchmark, February 2023. arXiv:2302.07541 [cs, q-bio].

Bangyi Zhao, Weixia Xu, Jihong Guan, and Shuigeng Zhou. Molecular property prediction based on graph structure learning. *Bioinformatics*, 40(5):btae304, 05 2024.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023.

Xiang Zhuang, Qiang Zhang, Bin Wu, Keyan Ding, Yin Fang, and Huajun Chen. Graph sampling-based meta-learning for molecular property prediction. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 2023.

A PROOFS

Proof for Proposition 4.1. We consider the gradient of the edge-level objective, as is given by:

$$\begin{aligned}\frac{\partial \mathcal{L}_e}{\partial \boldsymbol{w}} &= \frac{1}{|\mathcal{A}|}\sum\nolimits_{(i,j)\in\mathcal{A}} \frac{\partial \ell_{e_{ij}}}{\partial \boldsymbol{w}} \\ &= \frac{1}{|\mathcal{A}|}\sum\nolimits_{(i,j)\in\mathcal{A}} -(y_i - y_j)(\frac{\partial \hat{y}_i}{\partial \boldsymbol{w}} - \frac{\partial \hat{y}_j}{\partial \boldsymbol{w}})\end{aligned}$$

As is previously analyzed, the sign of $\frac{\partial \hat{y}_i}{\partial \boldsymbol{w}}$ should only depend on the ground truth y_i, thus we have:

$$\begin{aligned}\frac{\partial \mathcal{L}_e(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{1}{|\mathcal{A}|}\sum\nolimits_{i:(i,j)\in\mathcal{A}} -(2y_i - 1)\frac{\partial \hat{y}_i}{\partial \boldsymbol{w}} \\ &= \frac{1}{|\mathcal{A}|}\sum\nolimits_{i} -n_i(2y_i - 1)\frac{\partial \hat{y}_i}{\partial \boldsymbol{w}}\end{aligned}$$

□

B EXPERIMENT DETAILS

All experiments are run on a single NVIDIA RTX A6000 GPU. For all experiments in this work, we use the Adam optimizer (Kingma & Ba, 2015), and follow its default hyper-parameters: learning rate η is set 0.001, first-order momentum weight β_1 is set to 0.9, and the second-order momentum weight β_2 is set to 0.99. The batch size is set to 256 for all data sets.

Unless otherwise specified, we set the $R(t)$ schedule as $R(t) = \lambda \min(t/(\gamma T), 1)$ with $\lambda = 0.2$ and $\gamma = 0.1$, and the weight α for pairwise loss $\mathcal{L}_e$ is set to 0.1. The data splits of all data sets in our experiments follow the scaffold split in (Wang et al., 2023b).

All data sets used in our experiments are released under MIT license. Some statistics on data sets used in experiments are in Table 9.

Table 9: Summary for data sets used in experiments.

Data sets	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
# molecules	7831	8521	1427	93087	1513	2039	1477	41127
# MMP pairs	3212114	3802710	11935	2243595	15894	24105	7080	20740266
# AC pairs	315841	381260	3183	2610	1470	1186	1912	2484912

C ADDITIONAL EMPIRICAL RESULTS

C.1 MORE DETAILED EXPERIMENT RESULTS FOR CLASSIFICATION

Table 10 shows the performance with or without the proposed method LAC on different classification data sets with deviation. We can see that LAC consistently improves upon different backbone models.

Table 10: ROC-AUC with deviation in parenthesis on different data sets.

Method	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
GraphMVP	75.9(0.5)	63.1(0.4)	63.9(1.2)	77.7(0.6)	81.2(0.9)	72.4(1.6)	79.1(2.8)	77.0(1.2)
GraphMVP+LAC	76.7(0.6)	70.1(0.6)	64.5(1.1)	78.1(0.4)	81.6(0.8)	72.9(1.2)	80.2(1.9)	77.8(1.4)
3D-PGT	73.8(0.2)	69.2(1.1)	60.6(1.1)	69.4(1.2)	80.9(1.4)	72.1(0.9)	79.4(0.9)	69.4(0.4)
3D-PGT+LAC	75.2(0.2)	74.0(0.3)	61.0(0.1)	75.1(0.3)	84.5(0.6)	72.4(0.2)	79.6(0.8)	69.5(0.4)
UniMol	79.6(0.5)	69.6(0.1)	65.9(1.3)	82.1(1.3)	85.7(0.2)	72.9(0.6)	91.9(1.8)	80.8(0.3)
UniMol+LAC	80.2(0.6)	72.5(0.8)	66.2(1.1)	82.7(0.9)	86.4(0.3)	73.6(0.7)	92.2(1.6)	80.9(0.4)

Table 11 compares the ROC-AUC scores only for molecules with/without AC on different data sets. For simplicity, we use UniMol as the base model as it generally performs best on all these data sets. We can see that while both methods achieve worse performance on molecules with AC, the proposed method improves more on the prediction accuracy for the molecules with AC, comparing the row on "improvements of LAC" for molecules with/without AC. Meanwhile, LAC also slightly improves the prediction accuracy for molecules without AC.

Table 11: ROC-AUC (with deviation in parenthesis) for molecules with/without AC on different data sets.

Method	AC	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
UniMol	$\surd$	67.4(1.5)	59.7(0.1)	65.7(1.2)	69.6(1.2)	81.4(0.5)	67.9(1.2)	90.7(1.8)	78.6(0.3)
UniMol+LAC	$\surd$	71.2(1.4)	63.9(1.3)	66.1(1.0)	70.8(0.9)	84.2(0.8)	70.7(1.0)	91.5(1.6)	78.9(0.4)
Improvement of LAC (%)	$\surd$	5.64	7.04	0.61	1.72	3.43	4.12	0.88	0.38
UniMol	$\times$	82.4(0.4)	79.1(0.1)	80.0(0.1)	82.2(1.2)	86.8(0.1)	74.0(0.4)	92.6(1.8)	82.2(0.2)
UniMol+LAC	$\times$	82.7(0.2)	80.3(0.6)	80.0(0.1)	82.6(0.8)	87.0(0.1)	74.2(0.6)	92.8(1.6)	82.5(0.4)
Improvement of LAC (%)	$\times$	0.36	1.52	0	0.49	0.23	0.27	0.22	0.36

C.2 Effects of Batch Size

Table 12 compares the performance with different batch sizes for LAC on both GraphGPS and 3D-PGT model. While we set the batch size to be 256 for all data sets in our experiments, we can see that setting the batch size either too large (1024) or too small (64) may not lead to the best performance. Setting the batch size too small cannot cover enough activity cliff pairs in the edge-level loss of our method, hence cannot utilize this task well and may even leads to performance worse than the standard training pipeline (e.g., the Tox21 data). While setting the batch size larger leads to some improvement on large data sets like MUV or ToxCast, it leads to even worse performance for other data sets with limited molecules like Sider or BBBP. Such observation agrees with existing theoretical works on stochastic optimization for neural networks (Lin et al., 2018; 2020), as they demonstrate that large batch sizes can lead to worse generalization performance. Therefore, although setting the batch size to be larger can include more activity cliff pairs in a single batch, it may still not lead to better performance on all data sets.

Table 12: Ablation studies on the effect of batch size. The evaluation metric is ROC-AUC (Larger is better).

Method	Batch size	Tox21	ToxCast	Sider	MUV	Bace	BBBP	ClinTox	HIV
GraphGPS+LAC	64	72.9	72.1	59.7	70.7	81.9	67.1	72.2	77.3
	256	**74.0**	73.7	**60.4**	71.3	**82.5**	**67.7**	**72.4**	**77.6**
	1024	73.9	**73.8**	58.2	**71.6**	81.5	66.4	71.7	77.4
3D PGT+LAC	64	74.9	73.8	60.5	73.9	83.8	72.1	79.3	69.1
	256	**75.2**	**74.0**	**61.0**	75.1	**84.5**	**72.4**	**79.6**	**69.5**
	1024	75.0	**74.0**	60.1	**75.2**	81.3	71.8	76.6	69.2

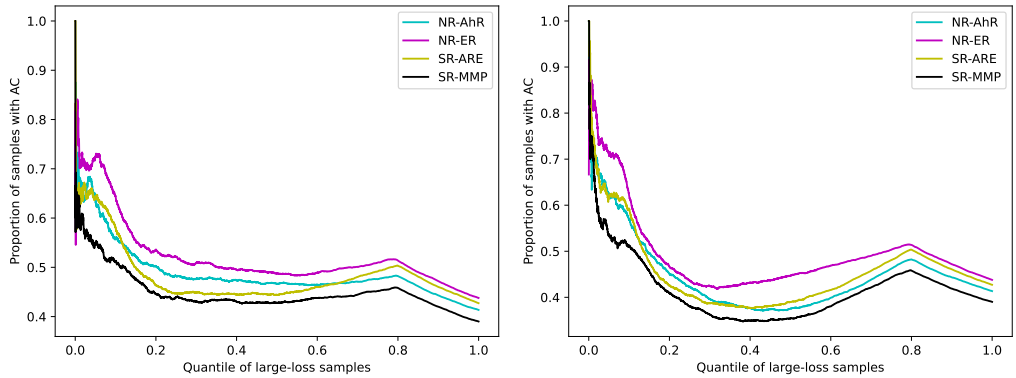
C.3 Time Cost on Activity Cliff Detection

Table 13 compares the total time cost in fine-tuning for the standard training pipeline and our proposed method LAC. Note that compared to standard training, LAC involves an additional process of finding all activity cliff pairs, therefore we show its time cost in two parts in parenthesis, where the first number represents the time cost of finding all activity cliff pairs and the second number represents the time cost of fine-tuning in Algorithm 1. We can see that the time cost for our method is almost the same as the standard training pipeline. In other words, the new node and edge-level tasks do not incur much additional time cost. Also, the time cost of finding all activity cliff pairs is generally limited compared to fine-tuning.

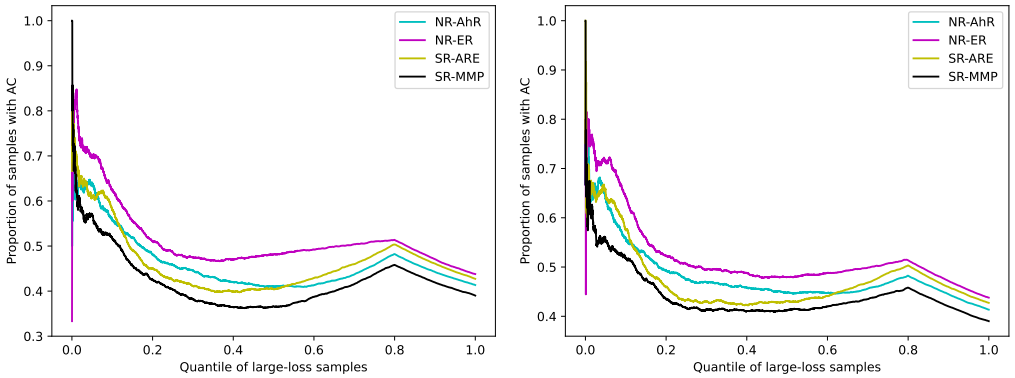
Table 13: CPU time cost (in minutes) of standard training pipeline and the proposed method LAC when fine-tuning 3D-PGT/UniMol model.

Data sets	Tox21	Sider	Bace	BBBP
3D-PGT	156	54	62	69
3D-PGT+LAC	196 (37+159)	58 (3+55)	70 (5+65)	73 (4+69)
UniMol	208	77	83	91
UniMol+LAC	247 (37+210)	80 (3+77)	88 (5+83)	97 (4+93)

C.4 Enlarged Figures in Section 3 and Additional Motivation Results



(a) GIN. (b) GraphGPS.

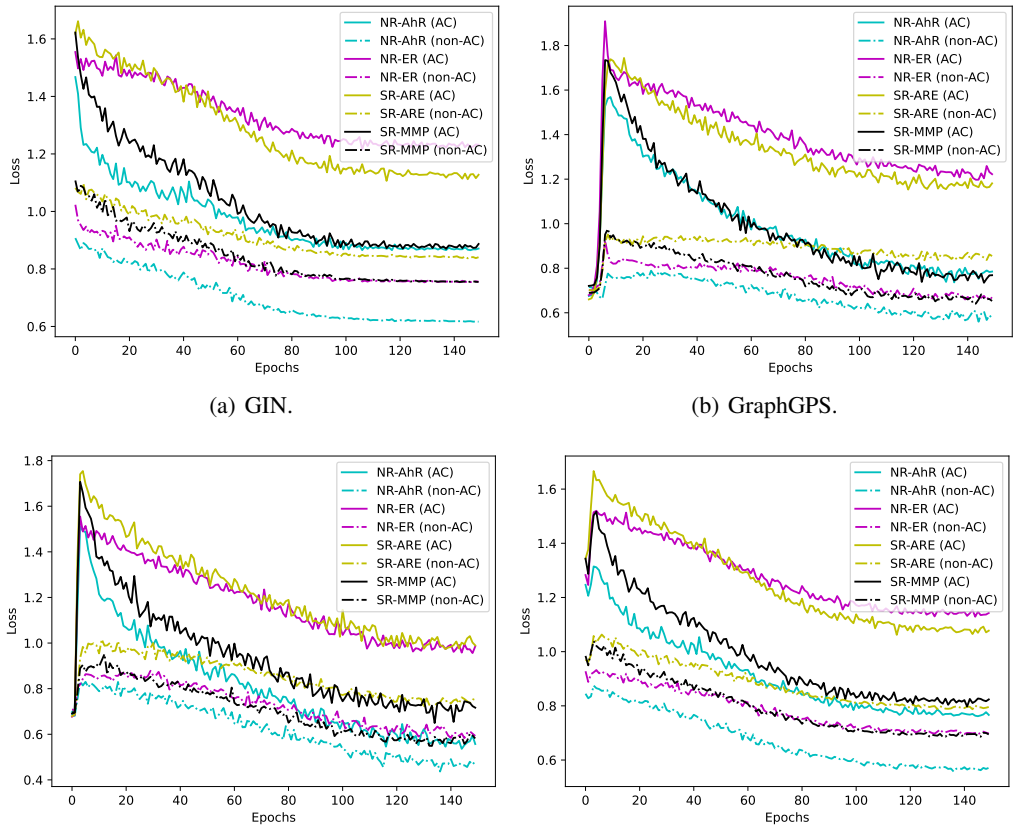


(c) Pre-trained 3D-PGT. (d) Pre-trained Uni-Mol.

Figure 8: (Larger version of Figure 2) Proportion of molecules with AC among molecules with top-$n\%$ loss values.

Figure 10 shows the average training losses for molecules with AC and molecules without AC. As can be seen, for all the four setups, **molecules with AC have significantly larger training losses than molecules without AC**. This demonstrates that molecules with AC are more difficult to learn due to their similar structures yet different properties. Moreover, from Figures 10(c) and 10(d), we can see that this phenomenon also exists for the 3D-PGT and Uni-Mol pre-trained models. In other words, molecules with AC are still more difficult to learn during fine-tuning of pre-trained models.

Since a pair of molecules with activity cliff have large difference in their properties, they may have larger influence on the prediction of each other during training. To demonstrate this, Figure 11 shows the average difference of training losses ("loss gap") between molecules with activity cliff. As can be seen, **AC leads to loss gaps between two molecules**, which also indicates that all these models fail to accurately classify both molecules, as in such cases the loss gap should be small (both with small loss). Instead, current models make the same prediction for these two molecules with AC. Only one



(a) GIN. (b) GraphGPS.

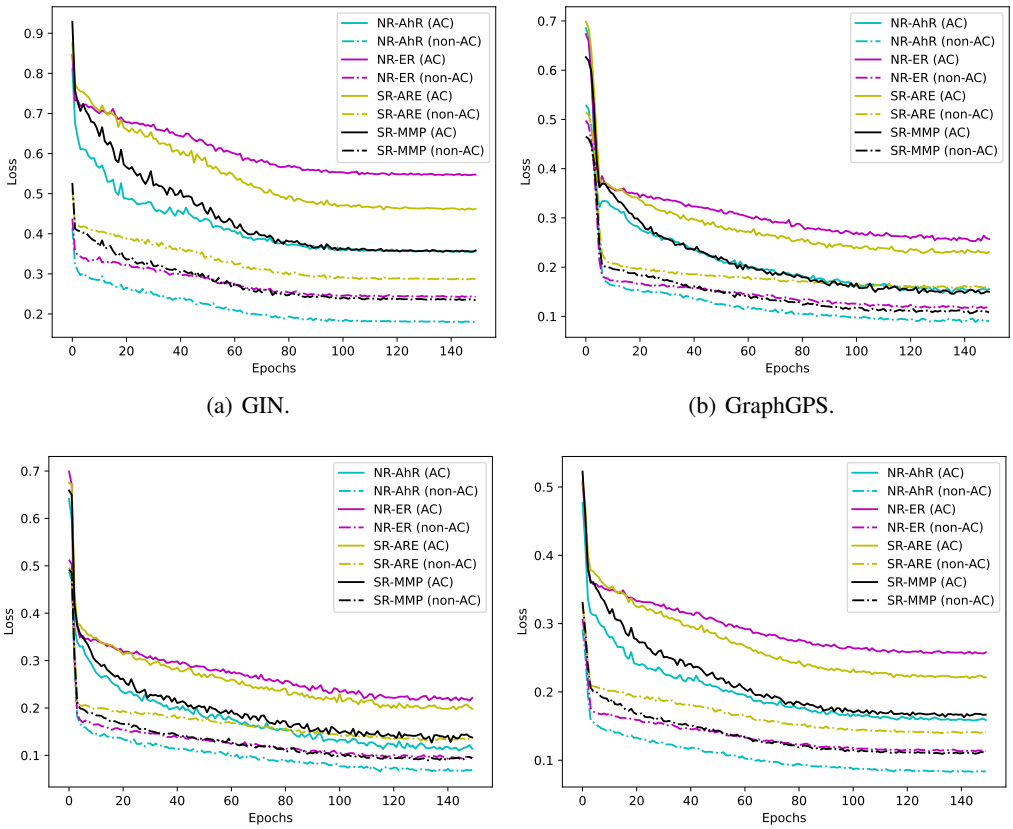
(c) pre-trained 3D-PGT. (d) pre-trained Uni-Mol.

Figure 9: (Larger version of Figure 3) Training losses of large-loss molecules with and without activity cliffs in four model training setups.

molecule is correctly classified with small loss, while another molecule has large loss that leads to the large loss gap.

Table 14: Training loss distributions with activity cliff on Tox21 data for train a GraphGPS model from scratch.

Task	# pairs with activity cliff	# pairs with loss gaps (ratio)	smallest loss gap
NR-AR	20686	19465 (94.1%)	0.361
NR-AR-LBD	12770	12757 (99.9%)	0.252
NR-AhR	81253	70967 (87.3%)	0.023
NR-Aromatase	12214	10237 (83.8%)	1.096
NR-ER	99097	93721 (94.6%)	0.011
NR-ER-LBD	31601	29745 (94.1%)	0.055
NR-PPAR-gamma	16581	15423 (93.0%)	0.339
SR-ARE	107397	101056 (94.1%)	0.102
SR-ATAD5	27128	25068 (92.4%)	0.290
SR-HSE	46875	43508 (92.8%)	0.261
SR-MMP	78148	60694 (77.7%)	0.009
SR-p53	25793	23942 (92.8%)	0.369



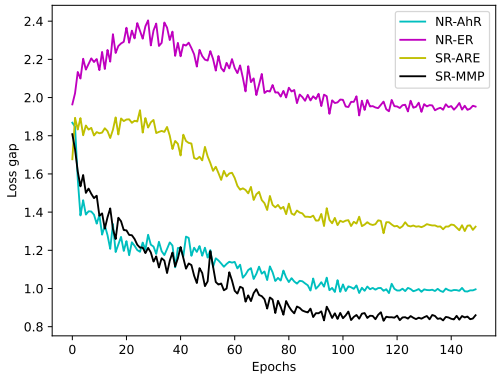
(a) GIN. (b) GraphGPS.

(c) Pre-trained 3D-PGT. (d) Pre-trained Uni-Mol.

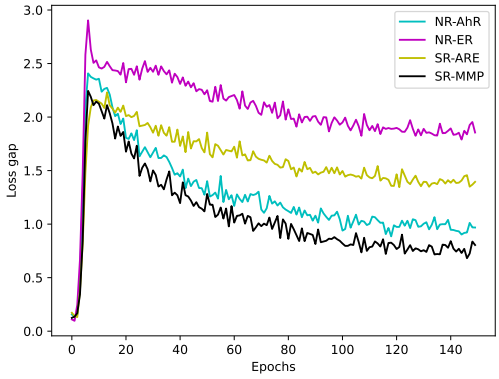
Figure 10: Training losses of molecules with and without activity cliffs in four model training setups.

Table 15: Training loss distributions with activity cliff on Tox21 data for fine-tuning 3D-PGT model.

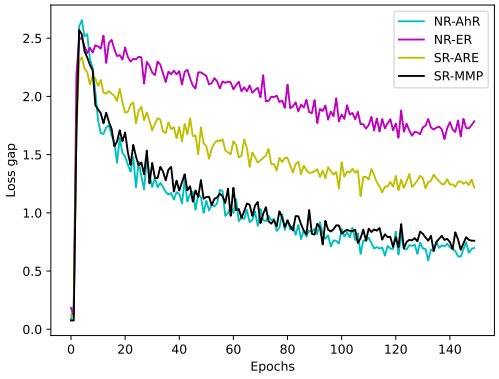
Task	# pairs with activity cliff	# pairs with loss gaps (ratio)	smallest loss gap
NR-AR	20686	19465 (94.1%)	0.498
NR-AR-LBD	12770	12757 (99.9%)	0.488
NR-AhR	81253	57066 (70.2%)	0.009
NR-Aromatase	12214	10230 (83.8%)	0.185
NR-ER	99097	88503 (89.3%)	0.031
NR-ER-LBD	31601	29407 (93.1%)	0.059
NR-PPAR-gamma	16581	15422 (93.0%)	0.347
SR-ARE	107397	99881 (93.0%)	0.025
SR-ATAD5	27128	24075 (88.7%)	0.067
SR-HSE	46875	43794 (93.4%)	0.253
SR-MMP	78148	60367 (77.2%)	0.006
SR-p53	25793	23944 (92.8%)	0.332



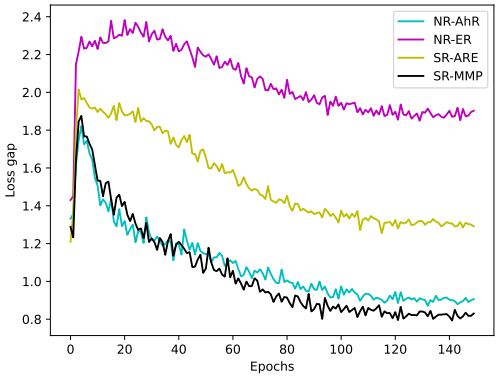
(a) GIN.



(b) GraphGPS.

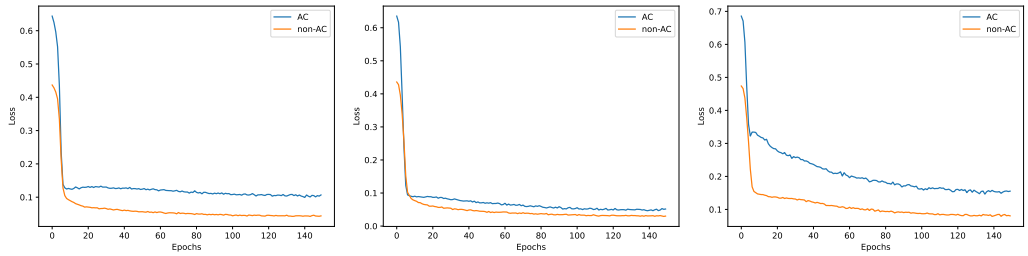


(c) pre-trained 3D-PGT.



(d) pre-trained Uni-Mol.

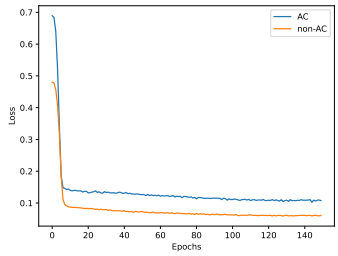
Figure 11: Loss gaps of molecules with AC for different tasks during model training.



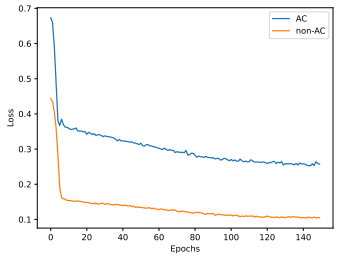
(a) NR-AR.

(b) NR-AR-LBD.

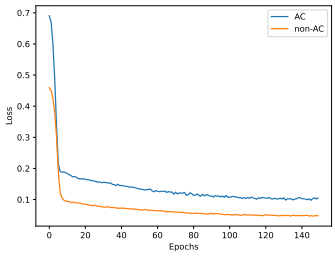
(c) NR-AhR.



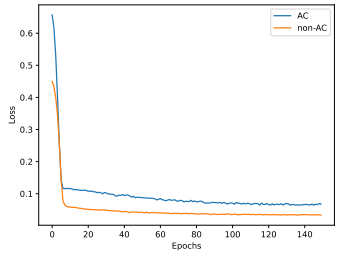
(d) NR-Aromatase.



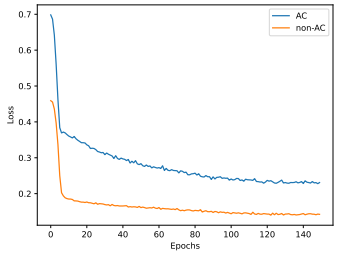
(e) NR-ER.



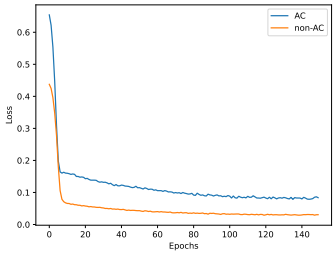
(f) NR-ER-LBD.



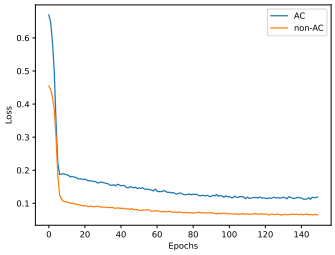
(g) NR-PPAR-gamma.



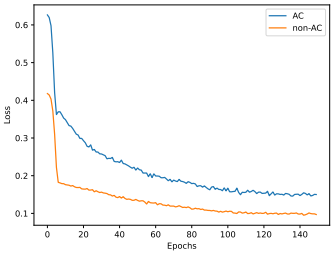
(h) SR-ARE.



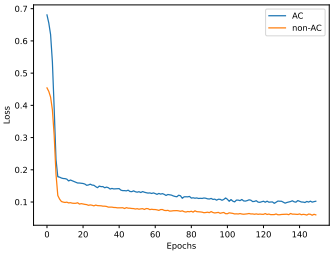
(i) SR-ATAD5.



(j) SR-HSE.

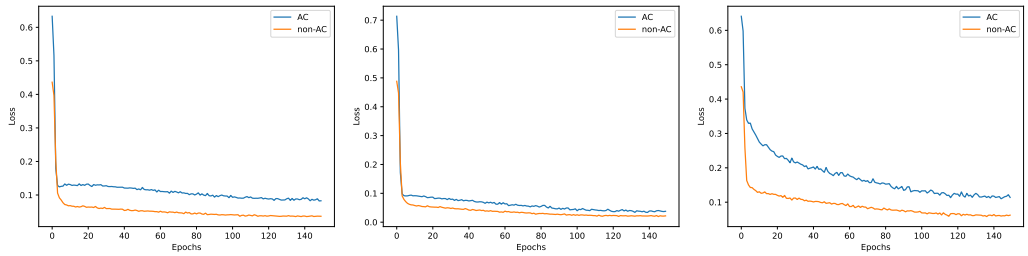


(k) SR-MMP.



(l) SR-p53.

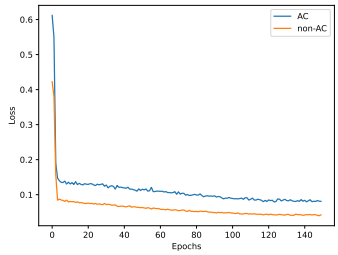
Figure 12: Training loss of molecules in/not in activity cliffs for randomly initialized GraphGPS model.



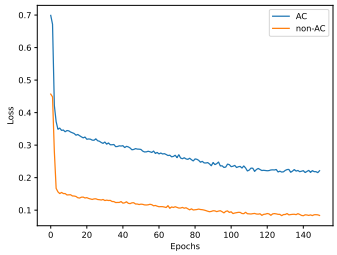
(a) NR-AR.

(b) NR-AR-LBD.

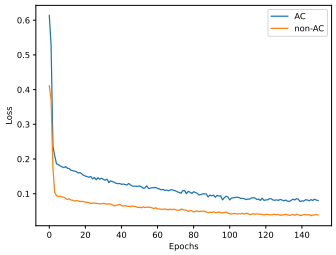
(c) NR-AhR.



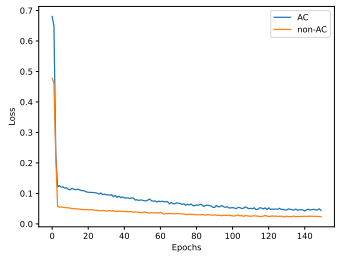
(d) NR-Aromatase.



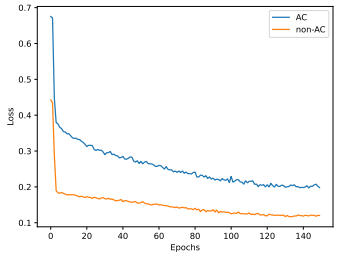
(e) NR-ER.



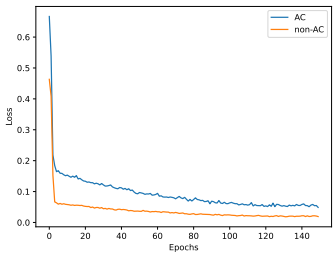
(f) NR-ER-LBD.



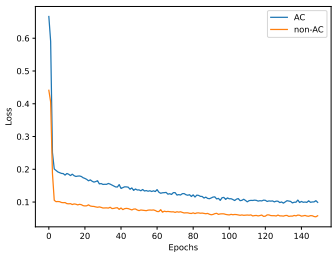
(g) NR-PPAR-gamma.



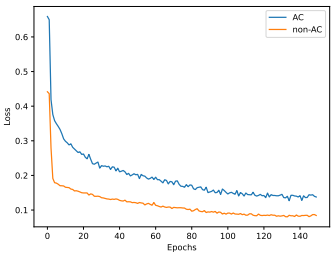
(h) SR-ARE.



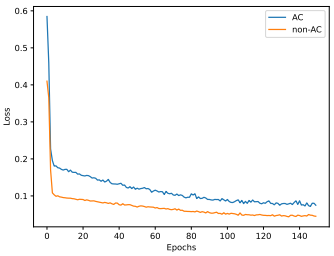
(i) SR-ATAD5.



(j) SR-HSE.



(k) SR-MMP.



(l) SR-p53.

Figure 13: Training loss of molecules in/not in activity cliffs for 3D-PGT pre-trained model.