
Beyond Efficiency: Molecular Data Pruning for Enhanced Generalization

Dingshuo Chen^{1,2} Zhixun Li³ Yuyan Ni⁴ Guibin Zhang⁵ Ding Wang^{1,2}
Qiang Liu^{1,2} Shu Wu^{1,2*} Jeffrey Xu Yu³ Liang Wang^{1,2}

¹New Laboratory of Pattern Recognition

State Key Laboratory of Multimodal Artificial Intelligence Systems
Institute of Automation, Chinese Academy of Sciences

² School of Artificial Intelligence, University of Chinese Academy of Sciences

³Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong

⁴Academy of Mathematics and Systems Science, Chinese Academy of Sciences

⁵Tongji University

✉ Primary contact: dingshuo.chen@cripac.ia.ac.cn

Abstract

With the emergence of various molecular tasks and massive datasets, how to perform efficient training has become an urgent yet under-explored issue in the area. Data pruning (DP), as an oft-stated approach to saving training burdens, filters out less influential samples to form a coresnet for training. However, the increasing reliance on pretrained models for molecular tasks renders traditional in-domain DP methods incompatible. Therefore, we propose a Molecular data Pruning framework for enhanced Generalization (MolPeg), which focuses on the *source-free data pruning* scenario, where data pruning is applied with pretrained models. By maintaining two models with different updating paces during training, we introduce a novel scoring function to measure the informativeness of samples based on the loss discrepancy. As a plug-and-play framework, MolPeg realizes the perception of both source and target domain and consistently outperforms existing DP methods across four downstream tasks. Remarkably, it can surpass the performance obtained from full-dataset training, even when pruning up to 60-70% of the data on HIV and PCBA dataset. Our work suggests that the discovery of effective data-pruning metrics could provide a viable path to both **enhanced efficiency** and **superior generalization** in transfer learning.

1 Introduction

The research enthusiasm for developing molecular foundation models is steadily increasing [1–5], attributed to its foreseeable performance gains with ever-larger model and amounts of data, as observed neural scaling laws [6, 7] and emergence ability [8] in other domains. However, the computational and storage burdens are daunting in model training [9], hyperparameter tuning, and model architecture search [10–12]. It is therefore urgent to ask for training-efficient molecular learning in the community.

Data pruning (DP), in a natural and simple manner, involves the selection of the most influential samples from the entire training dataset to form a coresnet as *paragons* for model training. The primary goal is to alleviate training costs by striking a balance point between efficiency and performance compromise. A trend in this field is developing data influence functions [13–15], training dynamic metrics [16–19], and coresnet selection [20–22] for lossless - although typically compromised - model

*Corresponding author: Shu Wu (shu.wu@nlpr.ia.ac.cn).

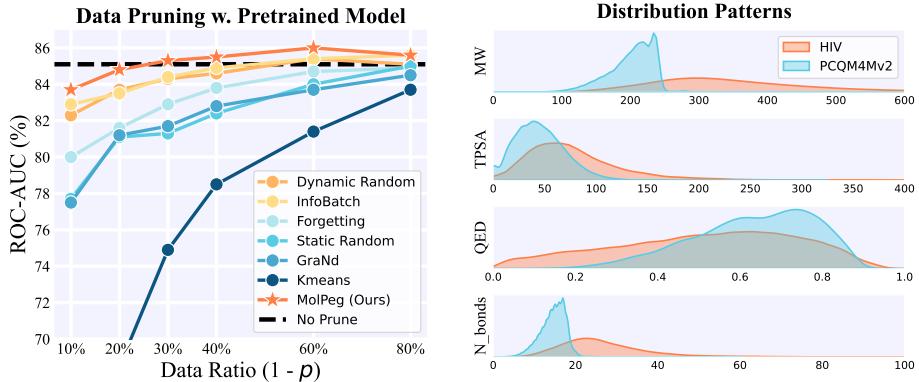


Figure 1: **(Left)** The performance comparison of different data pruning methods in HIV dataset under source-free data pruning setting. **(Right)** Distribution patterns of four important molecular features - molecular weight (MW), topological polar surface area (TPSA), Quantitative Estimate of Drug-likeness (QED) and number of bonds - in PCQM4Mv2 [33] and HIV [34] dataset, which are used for pretraining and finetuning, respectively.

generalization. When it comes to molecular tasks, transfer learning, particularly the *pretrain-finetune* paradigm, has been regarded as the de-facto standard for enhanced training stability and superior performance [23–27]. However, existing DP methods are purposed for train-from-scratch setting, i.e., the model is randomly initialized and trained on the selected coresset. A natural question arises as to *whether or not current DP methods remain effective when applied with pre-trained models*. Experimental analysis, as illustrated in Figure 1 (Left), suggests a negative answer. Most existing pruning strategies exhibit inferior results relative to the performance achieved with the full dataset, even falling short of simple random pruning.

In contrast to the existing DP approaches, which focus solely on a single target domain, the incorporation of pretrained model introduces an additional source domain, thereby inevitably exposing us to the challenge of distribution shift [28, 29]. Unfortunately, this is especially severe in molecular tasks, owing to the limited diversity of large-scale pretraining datasets compared to the varied nature of downstream tasks. As illustrated in Figure 1 (Right), we investigate the distribution patterns of several important molecular properties across the upstream and downstream datasets following Beaini et al. [30]. The observed disparities impede the model generalization, thus making DP with pretrained models a highly non-trivial task. We define this out-of-domain DP setting as *source-free data pruning*. It entails removing data from downstream tasks leveraging pre-trained models while remaining agnostic to the specifics of the pre-training data.

Of particular relevance to this work are approaches that propose DP methods for transfer learning [31, 32], which also target cross-domain scenarios. Despite the promising results they achieved, these methods select pretraining samples based on downstream data distribution, which necessitates reevaluation of previously selected samples and retraining heavy models as new samples involving, undermining the goal of achieving generalization and universality in pretraining. To this end, we take a step towards designing a DP method under the source-free data pruning setting to achieve **efficient and effective** model training, which aligns better with practical deployment for molecular tasks.

In this work, we propose a Molecular data Pruning framework for enhanced Generalization, which we term MolPeg for brevity. The core idea of MolPeg is to achieve cross-domain perception via maintaining an online model and a reference model during training, which places emphasis on the target and source domain, respectively. Besides, we design a novel scoring function to simultaneously select easy (representative) and hard (challenging) samples by comparing the absolute discrepancy between model losses. We further take a deep dive into the theoretical understanding and glean insight on its connection with the previous DP strategies. Note that our proposed MolPeg framework is generic, allowing for seamless integration of off-the-shelf pretrained models and architectures. To the best of our knowledge, this is the first work that studies how to perform data pruning for molecular learning from a transfer learning perspective. Our contributions can be summarized as follows:

- We analyze the challenges of efficient training in the molecular domain and formulate a tailored DP problem for transfer learning, which better aligns with the practical requirements of molecular pre-trained models.

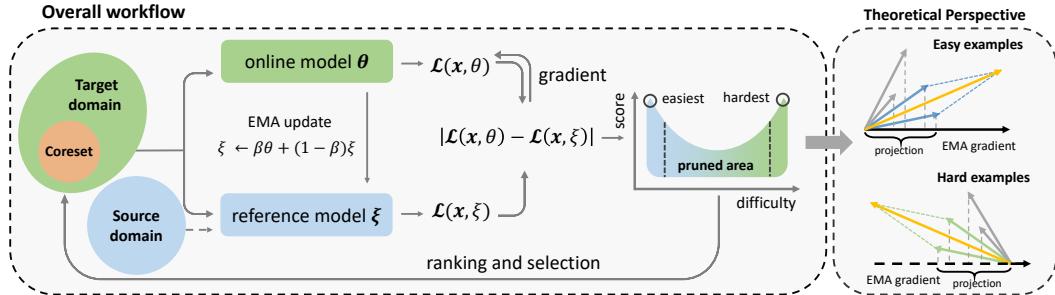


Figure 2: The overall framework of MolPeg. **(Left)** We maintain an online model and a reference model with different updating paces, which focus on the **target** and **source** domain, respectively. After model inference, the samples are scored by the absolute loss discrepancy and selected in ascending order. The easiest and hardest samples are given the largest score and selected to form the coreset. **(Right)** The selection process of MolPeg can be interpreted from a gradient projection perspective. Samples with low projection norms (grey) are discarded, while those with high norms are kept.

- We propose an efficient data pruning framework that can perceive both the source and target domains. It can achieve lightweight and effective DP without the need for retraining, facilitating easy adaptation to varied downstream tasks. We also provide a theoretical understanding of MolPeg and build its connections with existing DP strategies.
- We conduct extensive experiments on 4 downstream tasks, spanning different modalities, pretraining strategies, and task settings. Our method can surpass the full-dataset performance when up to 60%-70% of the data is pruned, which validates the effectiveness of our approach and unlocks a door to enhancing model generalization with fewer samples.

2 Preliminaries

In this section, we take a detour to revisit the traditional data pruning setting and *pretrain-finetune* paradigm before introducing the problem formulation of source-free data pruning.

Problem statement of traditional data pruning. Consider a learning scenario where we have a large training set denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, consisting of input-output pairs (x_i, y_i) , where $x_i \in \mathcal{X}$ represents the input and $y_i \in \mathcal{Y}$ denotes the ground-truth label corresponding to x_i . Here, \mathcal{X} and \mathcal{Y} refer to the input and output spaces, respectively. The objective of traditional data pruning is to identify a subset $\hat{\mathcal{D}} \subset \mathcal{D}$, that captures the most informative instances. The model trained on this subset $\hat{\mathcal{D}}$ should yield a slightly inferior or comparative performance to the model trained on the entire training set \mathcal{D} . Thus they need to strike a balance between efficiency and performance.

Revisit on transfer learning. Given source and target domain datasets \mathcal{D}_S and \mathcal{D}_T , the goal of pretraining is to obtain a high-quality feature extractor f in a supervised or unsupervised manner. While in the finetuning phase, we aim to adapt the pretrained f in conjunction with output head g to the target dataset \mathcal{D}_T .

Considering the proficiency of molecular pre-trained models in capturing meaningful chemical spaces, their widespread usage in enhancing performance across diverse molecular tasks has become commonplace. This necessitates a reassessment of the conventional approach to DP within the molecular domain and, more broadly, within the field of transfer learning. Previous attempts [31, 32] in data pruning for transfer learning primarily focus on trimming upstream data, selecting samples that closely match the distribution of downstream tasks to align domain knowledge. However, this necessitates retraining the model from scratch, which is notably ill-suited for the molecular domain, where the continual influx of new molecules introduces novel functionalities and structures. To this end, we propose a tailored DP problem for molecular transfer learning.

Problem formulation (Source-free data pruning). *Given a target domain dataset \mathcal{D}_T and a pretrained feature extractor parameterized by θ_S , we aim to identify a subset $\hat{\mathcal{D}}_T \subset \mathcal{D}_T$ for training, while being agnostic of the source domain dataset \mathcal{D}_S , to maximize the model generalization.*

3 Methodology

As with generic data pruning pipelines, the MolPeg framework is divided into two stages, scoring and selection. In the first stage, we define a scoring function to measure the informativeness of samples and apply it to the training set. In the subsequent stage, given the sample scores, we rank them in ascending order and maintain the high-ranking samples for training. Note that our pruning method is dynamically performed during the training process, rather than conducted before training.

We next introduce the MolPeg framework in detail. We track the training dynamics of two models with different update paces. For each training sample, we measure the difference in loss between the two models to quantify its importance, and then make the final selection based on this metric. In the following parts, we first intuitively introduce our design of the scoring function. Then, we further explore the theoretical support behind the effectiveness of the MolPeg. The overall framework is illustrated in Figure 2.

3.1 The MolPeg framework

The design of the scoring function addresses two key issues, (1) how to achieve the perception of source and target domain and (2) how to measure the informativeness of the samples.

Cross-domain perception. Since we are unable to access the upstream dataset, the pre-trained model serves as the only entry point of the source domain. During the finetuning stage, apart from *online encoder* undergoing gradient optimization via back-propagation, we further maintain a *reference encoder* updated with exponential moving average (EMA) to perceive the cross-domain knowledge. Note that both encoders are initialized by pretrained model $\theta_0 = \xi_0 = \theta^S$, where θ_t and ξ_t denotes the parameters of online and reference model at batch step t , respectively. They are updated as follows:

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}_t, \theta_t) \quad \xi_t = \beta \theta_t + (1 - \beta) \xi_{t-1} \quad (1)$$

where α is the learning rate and $\beta \in [0, 1)$ is the pace coefficient that controls the degree of history preservation. Here $\hat{\mathcal{D}}_t$ is the selected finetuning dataset for epoch t , and $\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}_t, \theta_t)$ denotes the average gradient $\frac{1}{|\hat{\mathcal{D}}_t|} \sum_{x_i \in \hat{\mathcal{D}}_t} \nabla_{\theta} \mathcal{L}(x_i, \theta_t)$ for short. Intuitively, We control the influence of target domain on the reference encoder via EMA. With a small update pace β , the online encoder prioritizes target domain, while the reference encoder emphasizes source domain.

Informativeness measurement and selection. By far we explicitly represent the inaccessible source domain knowledge with the help of the reference model, facilitating us to further quantify the informativeness of each sample in the cross-domain context. Our motivation for measuring the sample informativeness comes from a recent work that improves the neural scaling laws [35]. They suggest that the best pruning strategy depends on the amount of initial data. When the data volume is large, retaining the hardest samples yields better pruning results than retaining the easiest ones; the conclusion is the opposite when the data volume is small. This contrasts with the conclusion that only the hardest samples should be selected [16]. From an intuitive perspective, simple samples are more representative, allowing the model to adapt to downstream tasks more quickly, while hard samples are crucial for model generalization since they are considered *supporting vectors* near the decision boundaries. This debate highlights that in data pruning, how to perform a mixture of easy and hard samples is a critical factor. As shown in Figure 3, when 60% samples in the HIV dataset are pruned, simply selecting the easiest or hardest samples leads to a performance drop in later epochs.

Therefore, we opt to retain both easy and hard samples instead of singularly removing one type. To measure the information gap between domains, we adopt both *online* and *reference encoder* to infer each sample and calculate the absolute loss discrepancy between them:

$$\hat{\mathcal{D}}_t = \{x \in \mathcal{D}_t \mid |\mathcal{L}(x, \theta_t) - \mathcal{L}(x, \xi_t)| \geq \delta\}, \quad (2)$$

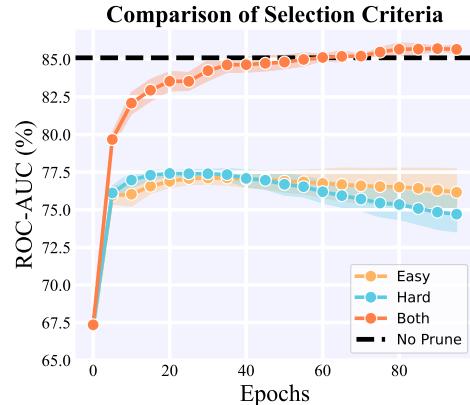


Figure 3: Performance comparison of selection criteria on HIV dataset when pruning 40% samples.

Algorithm 1: Molecular Data Pruning for Enhanced Generalization (MolPeg)

1 Inputs:

$\mathcal{D} = \{(\mathbf{x}_i, y_i, s_i)\}_{i=1}^{|\mathcal{D}|}$: dataset with the score for each example ($s_i = 1, \forall s_i \in \mathcal{D}$);
 α : learning rate; β : EMA update pace;
 p : data pruning ratio ($p < 1$); T : total number of training epochs;
 f_θ : pretrained encoder parameterized by θ

```

2  $t \leftarrow 0$ ;
3 while  $t \leq T$  do
4    $K \leftarrow p \cdot |\mathcal{D}|$ ;           /* Get the number of remaining samples */
5    $\hat{\mathcal{D}}_t \leftarrow \text{TopK}(s)$ ; /* Rank and Select the top-K samples for training */
6    $s_i \leftarrow \|\mathcal{L}(f_\theta(\mathbf{x}_i), y_i) - \mathcal{L}(f_\xi(\mathbf{x}_i), y_i)\|, \forall (\mathbf{x}_i, y_i, s_i) \in \hat{\mathcal{D}}_t$ ; /* Scoring the samples */
7    $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\hat{\mathcal{D}}_t, \theta)$ ;           /* Gradient update for online model */
8    $\xi \leftarrow \beta \theta + (1 - \beta) \xi$ ;           /* EMA update for reference model */
9    $t \leftarrow t + 1$ 
10 return

```

where $\mathcal{D}_t \in \mathcal{D}^T$ comprises the target domain data sampled for batch step t and $\hat{\mathcal{D}}_t \in \mathcal{D}_t$ comprises the data selected by MolPeg. δ is not a constant, but a threshold determined by the pruning ratio. Specifically, the rank of δ in the absolute loss discrepancy sequence $\{\|\mathcal{L}(\mathbf{x}_i, \theta_t) - \mathcal{L}(\mathbf{x}_i, \xi_t)\|\}_{i=1}^{|\mathcal{D}_t|}$ is $|\hat{\mathcal{D}}_t|$, i.e. pruning ratio $\times |\mathcal{D}_t|$. It is easy to infer that a positive loss discrepancy, i.e. $\mathcal{L}(\mathbf{x}, \theta_t) - \mathcal{L}(\mathbf{x}, \xi_t) > 0$, indicates the model struggles to accurately distinguish the sample, identifying it as hard one. Conversely, a negative loss discrepancy indicates that the model can easily improve its accuracy, marking it as an easy sample. Therefore, intuitively, we dynamically assess the learning difficulty of samples during the training process. By measuring the absolute value of the loss discrepancy, we keep the simplest (most representative) and the hardest (most challenging) samples, which are integrated as the most informative ones (Orange line in Figure 3). We also provide the pseudo-code of MolPeg in Algorithm 1.

3.2 Theoretical Understanding

In this section, we explore the theoretical underpinnings of the data selection process in MolPeg. Recall that our scoring function is defined by loss discrepancy, we further make use of Taylor expansion on the designed scoring function. Then, from the gradient perspective, i.e., the first-order expansion term, we derived the following propositions and the complete proof is provided in the Appendix E.

Assumption 1 (Slow parameter updating). *Assume the learning rate is small enough, so that the parameter update $\Delta\theta_t = \theta_{t+1} - \theta_t$ is small for every time step, i.e. $\|\Delta\theta_t\| \leq \epsilon, \forall t \in \mathbb{N}$, ϵ is a small constant.*

Proposition 1 (Interpretation of loss discrepancy). *With Assumption 1, the loss discrepancy can be approximately expressed by the dot product between the data gradient and the “EMA gradient”:*

$$\mathcal{L}(\mathbf{x}, \xi_t) - \mathcal{L}(\mathbf{x}, \theta_t) = \alpha \nabla_\theta \mathcal{L}(\mathbf{x}, \theta_t) \mathbf{v}_t^{EMA} + O(\epsilon^2), \quad (3)$$

where \mathbf{v}_t^{EMA} denotes $\sum_{j=1}^t (1 - \beta)^j \nabla_\theta \mathcal{L}(\hat{\mathcal{D}}_{t-j}, \theta_{t-j})$, i.e. the weighted sum of the historical gradients, which we termed as “EMA gradient”.

It indicates that the scoring function is essentially influenced by the magnitude of the dot product between the data gradient and the EMA gradient, as illustrated in Figure 2 (right). Given the EMA gradient, the size of the dot product is influenced by two factors: the norm of $\nabla_\theta \mathcal{L}(\mathbf{x}, \theta_t)$ and the angle between the two vectors. (i) A larger norm of the current data gradient is more likely to be selected, which resembles the criteria of GraNd score [19]. More connections to several well-known scoring functions are provided in the appendix F. (ii) If the current gradient direction closely aligns with the (opposite) EMA gradient direction, it often indicates an easy (hard) optimization of the sample, corresponding to the goal of selecting simple and hard samples in the previous analysis. Conversely, samples with gradient directions orthogonal to the EMA gradient are discarded.

In the following proposition, we examine the gradient of the selected samples and analyze simple and hard samples separately. Since the selection is performed at each fixed batch time step, we focus on one step of selection and omit the common time subscript t . Note that this result involves certain simplifications and approximations, and a formal version is provided in the appendix.

Proposition 2 (Gradient projection interpretation of MolPeg, informal). *Let $\mathcal{D}^+ \subseteq \mathcal{D}$ and $\hat{\mathcal{D}}^+ \subseteq \hat{\mathcal{D}}$ denote the sets of samples for which the dot products between the data gradients and the "EMA gradient" are positive. Then, the gradient of the selected "simple" samples can be expressed as:*

$$\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^+, \theta) = \nabla_{\theta} \mathcal{L}(\mathcal{D}^+, \theta) + a \mathbf{v}^{EMA}, a \geq 0. \quad (4)$$

Similarly, we define $\mathcal{D}^- \in \mathcal{D}$ and $\hat{\mathcal{D}}^- \in \hat{\mathcal{D}}$ as samples that have negative dot products, then

$$\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^-, \theta) = \nabla_{\theta} \mathcal{L}(\mathcal{D}^-, \theta) + b \mathbf{v}^{EMA}, b \leq 0. \quad (5)$$

$a = 0$ and $b = 0$ holds if and only if the loss discrepancy across \mathcal{D}^+ and \mathcal{D}^- is uniform respectively, which are uncommon scenarios.

Therefore, our data selection strategy essentially increases the weight of the (opposite) EMA gradient direction in the data gradient for easy (hard) samples. When \mathcal{D}^+ predominates, indicating a majority of simple samples in the dataset, this simplified model is akin to the momentum optimization strategy, which utilizes the sum of the current data gradient and the weighted EMA gradient to update the model parameters. This suggests that retaining simple samples may enhance optimization stability, allowing the model to overcome saddle points and local minima [36]. However, our method differs from the momentum optimization strategy in two key aspects. Firstly, we preserve directions opposite to the EMA gradient to target hard and forgettable samples. Secondly, our EMA gradient, which records the gradient of the coreset rather than the entire set, can retain more historical information under the same update pace.

4 Experimental Settings

4.1 Datasets and tasks

To comprehensively validate the effectiveness of our proposed MolPeg, we conduct experiments on three datasets, i.e., HIV [34], PCBA [37], MUV [38] and QM9 [39], covering four types of molecular tasks. These tasks span two molecular modalities—2D graph and 3D geometry—as well as two types of supervised tasks, i.e., classification and regression.

Given the potential issues of over-fitting and spurious correlations that may arise with limited samples when a large pruning ratio is adopted, we focus on relatively large-scale datasets containing at least 40K molecules. Below, we briefly summarize the information of the datasets. For a more detailed description and statistics of the dataset, please refer to Appendix A.

4.2 Implementation details

In this section, we provide a succinct overview of the implementation details for our experiments, including backbone models for different modalities, training details and evaluation protocols.

Backbone models. Given the two modalities involved in our experiment, we need corresponding backbone models for data modeling. Below is a concise introduction to the backbone models. For a more comprehensive understanding of the model architecture, please refer to the Appendix D.

- For 2D graphs, we utilize the Graph Isomorphism Network (GIN) [40] as the encoder. To ensure the generalizability of our research findings, we adopt the commonly recognized experimental settings proposed by Hu et al. [41], with 300 hidden units in each layer, and a 50% dropout ratio. The number of layers is set to 5.
- For 3D geometries, we employ two widely used backbone models, PaiNN [42] and SchNet [43], as the encoders for different datasets. For SchNet, we set the hidden dimension and the number of filters in continuous-filter convolution to 128. The interatomic distances are measured with 50 radial basis functions, and we stack 6 interaction layers. For PaiNN, we adopt the setting with 128 hidden dimensions, 384 filters, 20 radial basis functions, and stack 3 interaction layers.

Table 1: The performance comparison to state-of-the-art methods on HIV and PCBA in terms of ROC-AUC (%), \uparrow and Average Precision (%), \uparrow . We highlight the best-performing results in **boldface**. The performance difference with whole dataset training is highlighted with **blue** and **orange**, respectively.

Dataset		HIV						PCBA						
		90	80	70	60	40	20	90	80	70	60	40	20	
Static	Hard Random	77.7 _{17.4}	81.1 _{4.0}	81.3 _{3.8}	82.4 _{2.7}	84.0 _{1.1}	85.0 _{0.1}	14.6 _{11.7}	18.7 _{7.6}	21.1 _{5.2}	23.2 _{3.1}	25.3 _{1.0}	26.2 _{0.1}	
	CD	77.5 _{17.6}	80.9 _{4.2}	81.5 _{3.6}	82.7 _{2.4}	83.4 _{1.7}	84.9 _{0.2}	14.7 _{11.6}	18.0 _{8.3}	20.8 _{5.5}	21.9 _{4.4}	25.1 _{1.2}	26.0 _{0.3}	
	Herding	63.6 _{21.5}	72.0 _{13.1}	73.9 _{11.2}	76.9 _{8.2}	82.2 _{2.9}	84.7 _{0.4}	8.1 _{18.2}	10.6 _{15.7}	11.7 _{14.6}	13.7 _{12.6}	17.2 _{9.1}	22.6 _{3.7}	
	K-Means	61.8 _{23.3}	68.5 _{16.6}	74.9 _{10.2}	78.5 _{6.6}	81.4 _{3.7}	83.7 _{1.4}	12.8 _{13.5}	16.7 _{9.6}	19.6 _{6.7}	21.4 _{4.9}	24.1 _{2.2}	25.8 _{0.5}	
	Least Confidence	79.2 _{15.9}	81.0 _{4.1}	82.4 _{2.7}	82.8 _{2.3}	83.2 _{1.9}	85.1 _{0.0}	14.4 _{11.9}	19.2 _{7.1}	21.6 _{4.7}	23.2 _{3.1}	25.7 _{1.0}	26.0 _{0.3}	
	Entropy	78.7 _{16.4}	81.1 _{4.0}	81.3 _{3.8}	82.4 _{2.7}	84.3 _{0.8}	85.2 _{0.1}	14.6 _{11.7}	18.4 _{7.9}	21.4 _{4.9}	23.2 _{3.1}	25.5 _{0.8}	26.7 _{0.4}	
	Forgetting	80.0 _{15.1}	81.6 _{3.5}	82.9 _{2.2}	83.8 _{1.3}	84.7 _{0.4}	84.9 _{0.3}	15.3 _{11.0}	18.9 _{7.4}	21.3 _{5.0}	22.3 _{4.0}	25.3 _{1.0}	26.1 _{0.2}	
	GraNd-4	77.5 _{17.6}	81.2 _{3.9}	81.7 _{3.4}	82.8 _{2.3}	83.7 _{1.4}	84.5 _{0.6}	14.7 _{11.6}	18.4 _{7.9}	21.1 _{5.2}	22.6 _{3.7}	25.5 _{0.8}	26.2 _{0.1}	
	GraNd-20	80.1 _{15.0}	82.5 _{2.6}	83.0 _{2.1}	83.9 _{1.2}	84.7 _{0.4}	84.9 _{0.2}	15.8 _{10.5}	19.4 _{6.9}	22.0 _{4.3}	23.1 _{3.2}	25.7 _{1.0}	26.0 _{0.3}	
	DeepFool	76.8 _{18.3}	80.9 _{4.2}	81.5 _{3.6}	82.0 _{3.1}	83.1 _{2.0}	84.6 _{0.5}	13.9 _{12.4}	17.5 _{8.8}	20.9 _{5.4}	22.2 _{4.1}	24.9 _{1.4}	25.9 _{0.4}	
	Craig	76.5 _{18.6}	80.8 _{4.3}	81.3 _{3.8}	82.5 _{2.6}	83.8 _{1.3}	85.0 _{0.1}	14.5 _{11.8}	18.7 _{7.6}	21.3 _{5.0}	22.9 _{3.4}	25.1 _{1.2}	26.0 _{0.3}	
	Glister	80.9 _{14.2}	82.3 _{12.8}	83.4 _{11.7}	84.0 _{11.1}	84.9 _{0.2}	85.2 _{0.1}	15.5 _{10.8}	18.8 _{7.5}	21.6 _{4.7}	23.2 _{3.1}	25.3 _{1.0}	26.1 _{0.2}	
	Influence	76.5 _{18.6}	80.5 _{4.6}	81.7 _{3.4}	82.5 _{2.6}	83.4 _{1.7}	84.2 _{0.9}	13.7 _{12.6}	17.9 _{8.4}	20.5 _{5.8}	22.1 _{4.2}	24.5 _{1.6}	25.4 _{0.9}	
	EL2N-20	79.8 _{15.3}	82.0 _{3.1}	83.5 _{1.6}	84.0 _{1.1}	85.4 _{0.3}	85.1 _{0.0}	14.7 _{11.6}	19.1 _{7.2}	21.7 _{4.6}	22.5 _{3.8}	25.5 _{0.8}	26.1 _{0.2}	
	DP	77.9 _{17.2}	80.1 _{5.0}	82.5 _{2.6}	83.7 _{1.4}	84.6 _{0.5}	85.0 _{0.1}	14.1 _{12.2}	18.2 _{8.1}	20.9 _{5.4}	22.8 _{3.5}	25.1 _{1.2}	25.9 _{0.4}	
Dynamic	Soft Random	82.3 _{12.8}	83.7 _{1.4}	84.3 _{0.8}	84.6 _{0.5}	85.0 _{0.1}	85.1 _{0.0}	16.1 _{10.2}	19.2 _{7.1}	21.0 _{5.3}	22.3 _{4.0}	24.2 _{2.1}	25.4 _{0.9}	
	ϵ -greedy	82.5 _{12.6}	83.2 _{1.9}	83.7 _{1.4}	84.1 _{1.0}	84.8 _{0.3}	85.1 _{0.0}	16.5 _{9.8}	19.8 _{6.5}	20.3 _{6.0}	21.5 _{4.8}	23.8 _{2.5}	25.2 _{1.1}	
	UCB	82.6 _{12.5}	83.0 _{2.1}	83.5 _{1.6}	83.9 _{1.2}	84.5 _{0.6}	84.7 _{0.4}	16.7 _{9.6}	20.2 _{6.1}	22.0 _{4.3}	23.5 _{2.8}	24.9 _{1.4}	26.1 _{0.2}	
	InfoBatch ¹	82.9 _{12.2}	83.5 _{1.6}	84.4 _{0.7}	84.9 _{0.2}	85.4 _{0.3}	85.2 _{0.1}	19.9 _{6.4}	22.8 _{3.5}	24.5 _{1.8}	25.5 _{0.8}	26.8 _{0.5}	27.0 _{0.7}	
	MolPeg	83.7_{11.4}	84.8_{0.3}	85.3_{0.2}	85.5_{0.4}	86.0_{0.9}	85.6_{0.5}		20.7_{5.6}	23.9_{2.4}	25.6_{0.7}	26.4_{0.1}	26.8_{0.5}	27.0_{0.7}
	Whole Dataset	85.1 \pm 0.3						26.3 \pm 0.1						

¹ To make a fair comparison, we remove the annealing operation in the InfoBatch, since it uses the full dataset for training at later epochs.

Training details. We adhere to the settings proposed by [41] for our experiments. In classification tasks, the dataset is randomly split, with an 80%/10%/10% partition for training, validation and testing, respectively. In regression tasks, the QM9 dataset is divided into 110K molecules for training, 10K for validation, and another 10K for testing. The Adam optimizer [44] is employed for training with a batch size of 256. For classification tasks, the learning rate is set at 0.001 and we opt against using a scheduler. For regression tasks, we align with the original experimental settings of PaiNN and SchNet, setting the learning rate to 5×10^{-4} and incorporating a cosine annealing scheduler.

Evaluation protocols. We conduct a series of experiments between model performance and varied data quantities. Specifically, we divide the pruning ratio into six proportional subsets: [20%, 40%, 60%, 70%, 80%, 90%], and for each configuration, we randomly select five seeds and report the mean performance. For HIV datasets, performance is measured using the Area Under the ROC-Curve (ROC-AUC), while reporting the performance on PCBA in terms of Average Precision (AP) —higher values in both metrics indicate better performance. When assessing quantum property predictions in the QM9 dataset, the Mean Absolute Error (MAE) is used as the performance metric, with lower values indicating better accuracy. The additional results on MUV dataset are demonstrated in Table 5.

5 Empirical Studies

5.1 Empirical analysis on classification tasks

Our empirical studies for classification tasks utilize the 2D graph modality as the input. We employ GIN as the backbone model and adopt GraphMAE [45] for model pre-training on the PCQM4Mv2 dataset. For a comprehensive comparison, we select the following two groups of DP methods as primary baselines in our experiments: static DP and dynamic DP, following [46]. The majority of previous methods fall into the former group, from which we select 14 competitive and classic DP methods as baselines, i.e., hard random pruning, CD [47], Herding [18], K-means [35], Least

Table 2: The performance comparison to state-of-the-art methods on QM9 dataset in terms of MAE (↓). We highlight the best- and the second-performing results in **boldface** and underlined, respectively.

Dataset	QM9-U0 (meV)						QM9-Zpve (meV)						
	Pruning Ratio %	90	80	70	60	40	20	90	80	70	60	40	20
Random		85.0	45.7	<u>34.2</u>	30.9	<u>19.2</u>	15.7	4.94	3.09	<u>2.53</u>	<u>2.26</u>	1.93	1.65
DP		136.0	68.5	39.8	32.3	20.8	16.1	8.56	6.29	3.62	2.36	2.05	1.68
InfoBatch		116.0	57.0	36.4	<u>30.1</u>	20.4	<u>15.6</u>	6.26	4.61	3.22	2.34	<u>1.91</u>	<u>1.64</u>
MolPeg		92.4	48.2	32.4	26.1	17.7	14.3	5.40	3.18	2.51	2.24	1.86	1.62

Confidence [48], Entropy [48], Forgetting [16], GraNd [19], EL2N [19], DeepFool [49], Craig [50], Glister [51], Influence [14] and DP [15]. Since dynamic pruning remains a niche topic, we identify four methods, to the best of our knowledge, to serve as baselines, i.e., soft random pruning, ϵ -greedy [52], UCB [52] and InfoBatch [46], with MolPeg also falling into this category. Please refer to Appendix C for a more detailed introduction to the baselines.

Performance comparison. Empirical results for DP methods are presented in Table 1. Our systematic study suggests the following trends: (i) *Dynamic DP strategies significantly outperform static DP strategies.* Soft random, as a fundamental baseline in dynamic DP, consistently outperforms the baselines of static groups across almost all pruning ratios, even surpassing some strong competitors such as Glister and GraNd. We also observe that the performance advantage of dynamic DP becomes more pronounced when the pruning ratio is relatively large. Intuitively, compared to fixing a subset for training, dynamic pruning can perceive the full dataset during training, thereby possessing a larger receptive field and naturally yielding better performance. As more data samples are retained, the ability of both groups to perceive the full training set converges, leading to smaller performance differences between them. (ii) *MolPeg achieves the state-of-the-art performance across all proportions.* On the HIV dataset, we can achieve nearly lossless pruning by removing 80% of the samples, surpassing other baseline methods significantly. Similarly, on the larger-scale PCBA dataset, we can still achieve lossless pruning by removing 60% of the data. (iii) *MolPeg brings superior generalization performance compared to fine-tuning on the full dataset.* For example, on the HIV dataset, we achieve an ROC-AUC performance of 86 when pruning 40% of the data, surpassing the 85.1 achieved with training on the full dataset. This indicates that appropriate data pruning can better aid model generalization given a pre-trained model. However, as more downstream data is introduced, the improvement brought by our method diminishes, as shown by the 20% pruning proportion, due to introducing data samples that hinder model generalization. More empirical results on MUV dataset

Efficiency comparison. In addition to performance, time efficiency is another crucial indicator for DP. We conduct a performance-efficiency comparison of various DP methods on the HIV dataset at a 60% pruning ratio, as shown in Figure 4. We define *time efficiency* as the reciprocal of the runtime multiplied by 1000. A higher value of this metric indicates greater efficiency. We can observe that despite MolPeg experiencing slight efficiency loss compared to random pruning, it demonstrates superior pruning performance. Compared to the current SOTA baseline model, InfoBatch, our method achieves better model generalization with comparable efficiency. Conversely, static pruning methods incur 1.6x to 2.1x greater time costs than random pruning, with model performance stagnating or declining. This underscores that MolPeg achieves superior performance with minimal efficiency costs. Despite increased memory usage introduced by the reference model, EMA is commonly used to stabilize molecular training, which allows our method to utilize EMA-saved models without added memory overhead.

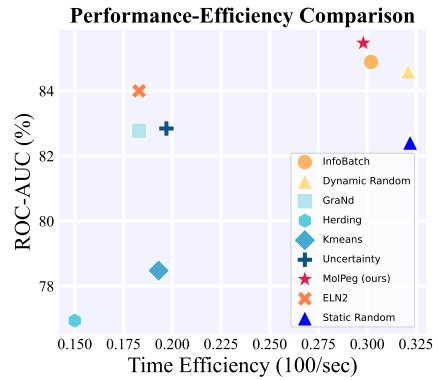


Figure 4: Performance and efficiency comparison between different DP methods. Pretrained models are fine-tuned on the HIV dataset at a 60% pruning ratio.

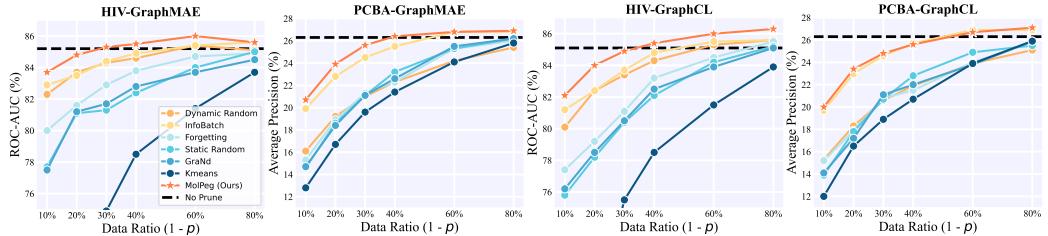


Figure 5: Data pruning trajectory given by downstream performance (%). Here the source models are pretrained on the PCQM4Mv2 dataset with GraphMAE and GraphCL strategies, respectively.

5.2 Results on QM9 dataset

Since regression is another common type of downstream molecular task, we also present the empirical results of MolPeg on two properties using the QM9 dataset, alongside comparisons with state-of-the-art methods. To ensure a fair comparison of experimental results, we employ the commonly used 3D geometry modality for modeling. We adopt GeoSSL [53] as the pretraining strategy and PaiNN as the backbone model, following the settings outlined by Liu et al [53]. Empirical results are presented in Table 2. It can be observed that MolPeg consistently outperforms other DP methods. However, all DP methods unexpectedly demonstrate inferior performance than random pruning in certain pruning ratios (80% and 90%). We speculate this phenomenon is attributed to the PCQM4Mv2 dataset used for pre-training and the QM9 dataset having a close match in the distribution patterns of molecular features. Thus, any non-uniform sampling methods would lead to biased data pruning which exacerbates distribution shift and hinders domain generalization.

5.3 Sensitivity Analysis

We further conduct extensive sensitivity analysis to validate the robustness of MolPeg across different pre-training strategies, molecular modalities, pre-training datasets and hyperparameter choices. All experiments below are conducted on the HIV dataset.

Robustness evaluation across pretraining strategies. Given that MolPeg primarily targets scenarios involving pre-trained models, it is necessary to compare its robustness when applied with different pre-training strategies. Without loss of generality, we select two representative pre-training strategies: generative self-supervised learning (SSL) and contrastive self-supervised learning, both of which dominate the field of molecular pre-training. Specifically, in addition to the results based on GraphMAE [45] (generative SSL) presented in Table 1, we also conduct experiments based on GraphCL (contrastive SSL) [54] whose results are shown in Figure 5. We can observe that MolPeg achieves optimal performance on both pre-training methods across different pruning ratios. Promisingly, it demonstrates better model generalization than training on the full dataset, indicating insensitivity to pre-training strategies of our proposed framework, thus allowing for convenient plug-in application to other pre-trained models in different molecular tasks.

Robustness evaluation across modalities. The selection of molecular modality has long been a contentious issue in the field. To validate the effectiveness of MolPeg across different molecular modalities, we present a comparison of pruning results using 3D geometry in the HIV dataset as shown in Table 3. We pretrain the SchNet [43] on the PCQM4Mv2 dataset, and keep other settings the same as in Section 4.2. It is evident from the results that the MolPeg framework, consistent with the conclusions drawn in Section 5.1, continues to outperform dynamic random pruning and enhance the model generalization ability. At a 40% pruning ratio, MolPeg also surpasses the performance achieved with training on the full dataset. This demonstrates the robustness of our proposed DP method across molecular modalities.

Robustness evaluation across pretraining datasets. In source-free transfer learning, pretrained model is a hard-encoded module, and their variations naturally lead to performance changes. Therefore,

Table 3: Performance with 3D modality on HIV dataset.

Pruning Ratio %	60	40	20
Random Pruning	80.1 _{±1.3}	80.8 _{±0.6}	81.2 _{±0.2}
MolPeg	81.9 _{±0.5}	82.3 _{±0.9}	82.2 _{±0.8}
Whole Dataset			81.4 _{±1.7}

it is necessary to evaluate the robustness of MolPeg when pretrained with different pre-training datasets. We conduct additional experiments on the HIV dataset using two pretrained models of varying quality, obtained from the ZINC15 [55] and QM9 datasets, respectively. Compared to the PCQM4Mv2 dataset used in the Section 5.1, these two datasets are smaller in scale and exhibit more pronounced distribution shifts, resulting in poorer pretraining quality. We observe the following trends from Table 5: (i) In the case of fine-tuning on the entire dataset, models pretrained on ZINC15 and QM9 show significantly inferior performance, even worse than training from scratch, indicating poor quality of pretrained models. (ii) MolPeg still achieves the best performance with these two pretrained models. This demonstrates the robustness of MolPeg across pretraining datasets.

How to choose β . Since EMA is a crucial component of our framework, it is necessary to evaluate how to choose a proper β . We conduct an empirical analysis on the HIV dataset across three pruning ratios, i.e., [0.1, 0.4, 0.8], and consider a candidate list covering the value ranges of β : [0.001, 0.01, 0.1, 0.5, 0.9]. Intuitively, a smaller β implies a slower parameter update pace in the reference model. When $\beta = 0$, it signifies using a frozen pre-trained model as the reference. The experimental results corresponding to the variation of β are illustrated in Figure 6. Empirical results indicate that the overall performance shows only moderate sensitivity to parameter change. However, typically, when $\beta = 0.5$, the model tends to achieve better performance and smaller standard deviation. Hence, for our primary experiments, we opt to default to $\beta = 0.5$.

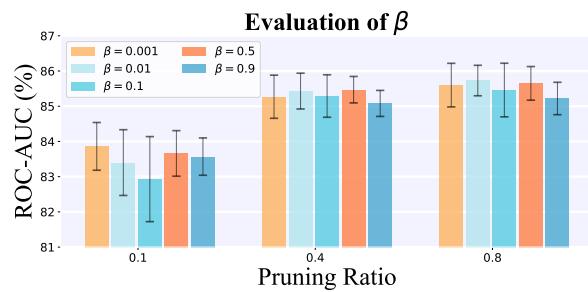


Figure 6: Performance bar chart of different choices of hyper-parameter β on HIV dataset. The error bar is measured in standard deviation and plotted in grey color.

6 Conclusion

In this work, we propose MolPeg, a novel molecular data pruning framework designed to enhance generalization without the need for source domain data, thereby addressing the limitations of existing in-domain data pruning (DP) methods. Our approach leverages two models with different update paces to measure the informativeness of samples. Through extensive experiments across four downstream tasks involving both classification and regression tasks, we demonstrate that MolPeg not only achieves lossless pruning but also outperforms full dataset training in certain scenarios. This underscores the potential of MolPeg to optimize training efficiency and improve the generalization of pre-trained models in the molecular domain. Our contributions highlight the importance of considering source domain information in DP methods and pave the way for more efficient and scalable training paradigms in molecular machine learning.

Broader impacts. Given that our application tasks fall within the molecular domain, improper use of methods for tasks such as molecular property prediction may result in significant deviations. This could impact subsequent applications of the molecules in drug development or materials design, especially in predicting properties like toxicity and stability. We recommend further experimental validation of key molecules after using the model to ensure the reliability of the results. We provide further discussions in Appendix H.

7 Acknowledgements

This work is jointly supported by National Science and Technology Major Project (2023ZD0120901) and National Natural Science Foundation of China (62372454).

References

- [1] Oscar Méndez-Lucio, Christos Nicolaou, and Berton Earnshaw. Mole: a molecular foundation model for drug discovery. *arXiv preprint arXiv:2211.02657*, 2022. [1](#)
- [2] Yizhen Luo, Kai Yang, Massimo Hong, Xingyi Liu, and Zaiqing Nie. Molfm: A multimodal molecular foundation model. *arXiv preprint arXiv:2307.09484*, 2023.
- [3] Jinho Chang and Jong Chul Ye. Bidirectional generation of structure and properties through a single molecular foundation model. *Nature Communications*, 15(1):2323, 2024.
- [4] Kerstin Kläser, Błażej Banaszewski, Samuel Maddrell-Mander, Callum McLean, Luis Müller, Ali Parviz, Shenyang Huang, and Andrew Fitzgibbon. Minimol: A parameter-efficient foundation model for molecular learning. *arXiv preprint arXiv:2404.14986*, 2024.
- [5] Nathan C Frey, Ryan Soklaski, Simon Axelrod, Siddharth Samsi, Rafael Gomez-Bombarelli, Connor W Coley, and Vijay Gadepally. Neural scaling of deep chemical models. *Nature Machine Intelligence*, 5(11):1297–1305, 2023. [1](#)
- [6] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. [1](#)
- [7] Dingshuo Chen, Yanqiao Zhu, Jieyu Zhang, Yuanqi Du, Zhixun Li, Qiang Liu, Shu Wu, and Liang Wang. Uncovering neural scaling laws in molecular representation learning. *Advances in Neural Information Processing Systems*, 36, 2023. [1](#)
- [8] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. [1](#)
- [9] 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*, pages 1263–1272. PMLR, 2017. [1](#)
- [10] Babatounde Moctard Oloulade, Jianliang Gao, Jiamin Chen, Raeed Al-Sabri, and Zhenpeng Wu. Cancer drug response prediction with surrogate modeling-based graph neural architecture search. *Bioinformatics*, 2023. [1](#)
- [11] Yijian Qin, Xin Wang, Ziwei Zhang, Pengtao Xie, and Wenwu Zhu. Graph neural architecture search under distribution shifts. In *International Conference on Machine Learning*, pages 18083–18095. PMLR, 2022.
- [12] Shengli Jiang, Shiyi Qin, Reid C Van Lehn, Prasanna Balaprakash, and Victor M Zavala. Uncertainty quantification for molecular property predictions with graph neural architecture search. *arXiv preprint arXiv:2307.10438*, 2023. [1](#)
- [13] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33:14879–14890, 2020. [1](#), [15](#)
- [14] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017. [8](#)
- [15] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. *arXiv preprint arXiv:2205.09329*, 2022. [1](#), [8](#)
- [16] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. [1](#), [4](#), [8](#), [15](#), [19](#)
- [17] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. [15](#)
- [18] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th annual international conference on machine learning*, pages 1121–1128, 2009. [7](#)
- [19] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021. [1](#), [5](#), [8](#), [15](#), [18](#)

[20] Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable bayesian logistic regression. *Advances in neural information processing systems*, 29, 2016. 1

[21] Trevor Campbell and Tamara Broderick. Automated scalable bayesian inference via hilbert coresets. *Journal of Machine Learning Research*, 20(15):1–38, 2019.

[22] Sungyun Kim, Sangmin Bae, and Se-Young Yun. Coreset sampling from open-set for fine-grained self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7537–7547, 2023. 1, 16

[23] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022. 2

[24] Jinhua Zhu, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Unified 2d and 3d pre-training of molecular representations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2626–2636, 2022.

[25] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.

[26] Haisong Gong, Qiang Liu, Shu Wu, and Liang Wang. Text-guided molecule generation with diffusion language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 109–117, 2024.

[27] Yanqiao Zhu, Dingshuo Chen, Yuanqi Du, Yingze Wang, Qiang Liu, and Shu Wu. Molecular contrastive pretraining with collaborative featurizations. *Journal of Chemical Information and Modeling*, 64(4):1112–1122, 2024. 2

[28] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. Good: A graph out-of-distribution benchmark. *Advances in Neural Information Processing Systems*, 35:2059–2073, 2022. 2

[29] Xin Sun, Liang Wang, Qiang Liu, Shu Wu, Zilei Wang, and Liang Wang. Dive: Subgraph disagreement for graph out-of-distribution generalization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2794–2805, 2024. 2

[30] Dominique Beaini, Shenyang Huang, Joao Alex Cunha, Zhiyi Li, Gabriela Moisescu-Pareja, Oleksandr Dymov, Samuel Maddrell-Mander, Callum McLean, Frederik Wenkel, Luis Müller, et al. Towards foundational models for molecular learning on large-scale multi-task datasets. In *The Twelfth International Conference on Learning Representations*, 2023. 2

[31] Yihua Zhang, Yimeng Zhang, Aochuan Chen, Jiancheng Liu, Gaowen Liu, Mingyi Hong, Shiyu Chang, Sijia Liu, et al. Selectivity drives productivity: Efficient dataset pruning for enhanced transfer learning. *Advances in Neural Information Processing Systems*, 36, 2023. 2, 3, 16

[32] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227, 2023. 2, 3, 16

[33] Maho Nakata and Tomomi Shimazaki. Pubchemqc project: a large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57(6):1300–1308, 2017. 2, 15

[34] AIDS Antiviral Screen Data. 2, 6, 15

[35] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022. 4, 7

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 6

[37] Yanli Wang, Jewen Xiao, Tugba O Suzek, Jian Zhang, Jiayao Wang, Zhigang Zhou, Lianyi Han, Karen Karapetyan, Svetlana Dracheva, Benjamin A Shoemaker, et al. Pubchem’s bioassay database. *Nucleic acids research*, 40(D1):D400–D412, 2012. 6, 15

[38] Sebastian G. Rohrer and Knut Baumann. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. *J. Chem. Inf. Model.*, 49(2):169–184, 2009. 6, 15

[39] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012. 6, 15

[40] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018. 6, 16

[41] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019. 6, 7

[42] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021. 6, 17

[43] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017. 6, 9, 16

[44] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[45] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604, 2022. 7, 9

[46] Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, et al. Infobatch: Lossless training speed up by unbiased dynamic data pruning. *arXiv preprint arXiv:2303.04947*, 2023. 7, 8, 15, 16, 18

[47] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 137–153. Springer, 2020. 7, 15

[48] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019. 8, 15, 16

[49] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018. 8, 15

[50] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020. 8, 15

[51] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 8, 15

[52] Ravi S Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning. *arXiv preprint arXiv:2111.12621*, 2021. 8, 16

[53] Shengchao Liu, Hongyu Guo, and Jian Tang. Molecular geometry pretraining with se (3)-invariant denoising distance matching. *arXiv preprint arXiv:2206.13602*, 2022. 9

[54] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020. 9

[55] Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015. 10

[56] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020. 15

[57] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014. 15

[58] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 15

[59] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019. [15](#)

[60] Greg Landrum et al. Rdkit: Open-source cheminformatics software. 2016. URL <http://www.rdkit.org/>, <https://github.com/rdkit/rdkit>, 149(150):650, 2016. [15](#)

[61] Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022. [15](#)

[62] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012. [15](#)

[63] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-gan: Speeding up gan training using core-sets. In *International Conference on Machine Learning*, pages 9005–9015. PMLR, 2020. [15](#)

[64] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2022. [15](#)

[65] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021. [15](#)

[66] Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021. [15](#)

[67] Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in neural information processing systems*, 34:14488–14501, 2021. [15](#)

[68] Vishal Kaushal, Suraj Kothawade, Ganesh Ramakrishnan, Jeff Bilmes, and Rishabh Iyer. Prism: A unified framework of parameterized submodular information measures for targeted data subset selection and summarization. *arXiv preprint arXiv:2103.00128*, 2021. [16](#)

[69] Suraj Kothawade, Nathan Beck, Krishnateja Killamsetty, and Rishabh Iyer. Similar: Submodular information measures based active learning in realistic scenarios. *Advances in Neural Information Processing Systems*, 34:18685–18697, 2021.

[70] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR, 2015. [16](#)

[71] Noveen Sachdeva, Carole-Jean Wu, and Julian McAuley. Svp-cf: Selection via proxy for collaborative filtering data. *arXiv preprint arXiv:2107.04984*, 2021. [16](#)

[72] Jiarong Xu, Renhong Huang, Xin Jiang, Yuxuan Cao, Carl Yang, Chunping Wang, and Yang Yang. Better with less: A data-active perspective on pre-training graph neural networks. *Advances in Neural Information Processing Systems*, 36:56946–56978, 2023. [16](#)

[73] Boris Weisfeiler and Andrei Leman. A Reduction of a Graph to a Canonical Form and an Algebra Arising During This Reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968. [16](#)

[74] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. [20](#)

[75] Liang Wang, Xiang Tao, Qiang Liu, and Shu Wu. Rethinking graph masked autoencoders through alignment and uniformity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15528–15536, 2024.

[76] Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*, 2023.

[77] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1725–1735, 2024. [20](#)

A Datasets and Tasks

In the following, we will elaborate on the adopted datasets and the statistics are summarized in Table 4.

Table 4: Statistics of datasets used in experiments.

	Dataset	Data Type	#Molecules	Avg. #atoms	Avg. #bonds	#Tasks	Avg. degree
Finetuning	Pre-training PCQM4Mv2	SMILES	3,746,620	14.14	14.56	-	2.06
	HIV	SMILES	41,127	25.51	27.47	1	2.15
	PCBA	SMILES	437,929	25.96	28.09	92	2.16
	QM9-U0	SMILES, 3D	130,831	18.03	18.65	1	2.07
	QM9-ZPVE	SMILES, 3D	130,831	18.03	18.65	1	2.07

- **PCQM4Mv2** is a quantum chemistry dataset curated by Hu et al. [56] based on the PubChemQC project [33]. It comprises 3,746,620 molecules and is extensively utilized in molecular pretraining tasks. We also adopt this widely recognized dataset for our molecular pretraining endeavors.
- **HIV** dataset is designed to evaluate the ability of molecular compounds to inhibit HIV replication [34] in a binary classification setting, consisting of 41,127 organic molecules.
- **PCBA** is a dataset consisting of biological activities of small molecules generated by high-throughput screening [37]. It contains 437,929 molecules with annotations of 92 classification tasks.
- **QM9** is a comprehensive dataset, structured for regression tasks, that provides geometric, energetic, electronic and thermodynamic properties for a subset of GDB-17 database, comprising 134 thousand stable organic molecules with up to nine heavy atoms [39]. In our experiments, we delete 3,054 uncharacterized molecules which failed the geometry consistency check [57]. We include the U0 and ZPVE in our experiment, which cover properties related to stability, and thermodynamics. These properties collectively capture important aspects of molecular behavior and can effectively represent various energetic and structural characteristics within the QM9 dataset.
- **MUV** (Maximum Unbiased Validation) group was selected from PubChem BioAssay via a refined nearest neighbor analysis approach, which is specifically designed for validation of virtual screening techniques [38].

B Computing infrastructures

Software infrastructures. All of the experiments are implemented in Python 3.7, with the following supporting libraries: PyTorch 1.10.2 [58], PyG 2.0.3 [59], RDKit 2022.03.1 [60].

Hardware infrastructures. We conduct all experiments on a computer server with 8 NVIDIA GeForce RTX 3090 GPUs (with 24GB memory each) and 256 AMD EPYC 7742 CPUs.

C Related work

Data pruning (DP) has been an ongoing research topic since the rise of deep learning. Traditional data pruning strategies often focus solely on the task dataset, exploring ways to represent the distribution of the entire dataset with fewer data points, thereby reducing training costs. However, with the recent advancements in transfer learning, focusing solely on the task dataset has become insufficient. Consequently, some data pruning strategies have been developed for transfer learning scenarios. We classify these strategies into in-domain data pruning and cross-domain data pruning.

In-domain data pruning. Most existing data pruning methods fall into this category. We further divide them into two groups: static data pruning and dynamic data pruning following [46]. Static data pruning aims to select a subset of data that remains unchanged throughout the training process, while dynamic data pruning methods consider that the optimal data subset evolves dynamically during training. Guo et al. [61] classify existing static data pruning methods based on their scoring function into the following categories: geometry [47, 62, 17, 63, 64], uncertainty [48], loss [16, 19, 46], decision boundary [49, 65], gradient matching [66, 50], bilevel optimization [13, 51, 67],

submodularity [68–70], and proxy [71, 48]. Despite dynamic data pruning is still in its early stages, it has demonstrated superior performance. Raju et al. [52] propose two dynamic pruning methods called UCB and ϵ -greedy. These methods define an uncertainty value and calculate the estimated moving average. During each pruning period, ϵ -greedy or UCB is used to select a fraction of the samples with the highest scores, and training is then conducted on these selected samples for that period. Recently, InfoBatch [46] achieves lossless pruning based on loss distribution and rescales the gradients of the remaining samples to approximate the original gradient. However, all of these methods place much emphasis on the target domain while ignoring the widespread use of transfer learning.

Cross-domain data pruning. We observe that with the use of pretraining, there is an additional source domain alongside the target domain. The key issue now is how to effectively utilize the information from both domains for data pruning in the context of transfer learning. To effectively address downstream tasks, a straightforward approach is to measure the distribution shift between the upstream and downstream data, and then prune the pretraining dataset to align its distribution with that of the downstream dataset [32, 31, 72, 22]. However, this method requires retraining the pretrained model for each different downstream task, which contradicts the intended *pretrain-finetune* paradigm. Therefore, we propose the problem of *source-free data pruning* which is aligned with practical usage of transfer learning.

D Backbone Model

D.1 Embedding 2D graphs

Graph Isomorphism Network (GIN) [40] is a simple and effective model to learn discriminative graph representations, which is proved to have the same representational power as the Weisfeiler-Lehman test [73]. Recall that each molecule is represented as $\mathcal{G} = (\mathbf{A}, \mathbf{X}, \mathbf{E})$, where \mathbf{A} is the adjacency matrix, \mathbf{X} and \mathbf{E} are features for atoms and bonds respectively. The layer-wise propagation rule of GIN can be written as:

$$\mathbf{h}_i^{(k+1)} = f_{\text{atom}}^{(k+1)} \left(\mathbf{h}_i^{(k)} + \sum_{j \in \mathcal{N}(i)} \left(\mathbf{h}_j^{(k)} + f_{\text{bond}}^{(k+1)}(\mathbf{E}_{ij}) \right) \right), \quad (6)$$

where the input features $\mathbf{h}_i^{(0)} = \mathbf{x}_i$, $\mathcal{N}(i)$ is the neighborhood set of atom v_i , and $f_{\text{atom}}, f_{\text{bond}}$ are two MultiLayer Perceptron (MLP) layers for transforming atoms and bonds features, respectively. By stacking K layers, we can incorporate K -hop neighborhood information into each center atom in the molecular graph. Then, we take the output of the last layer as the atom representations and further use the mean pooling to get the graph-level molecular representation:

$$\mathbf{z}^{\text{2D}} = \frac{1}{N} \sum_{i \in \mathcal{V}} \mathbf{h}_i^{(K)}. \quad (7)$$

D.2 Embedding 3D geometries

SchNet [43]. We use the SchNet [43] as the encoder for the 3D geometries in HIV dataset. SchNet models message passing in the 3D space as continuous-filter convolutions, which is composed of a series of hidden layers, given as follows:

$$\mathbf{h}_i^{(k+1)} = f_{\text{MLP}} \left(\sum_{j=1}^N f_{\text{FG}}(\mathbf{h}_j^{(t)}, \mathbf{r}_i, \mathbf{r}_j) \right) + \mathbf{h}_i^{(t)}, \quad (8)$$

where the input $\mathbf{h}_i^{(0)} = \mathbf{a}_i$ is an embedding dependent on the type of atom v_i , $f_{\text{FG}}(\cdot)$ denotes the filter-generating network. To ensure rotational invariance of a predicted property, the message passing function is restricted to depend only on rotationally invariant inputs such as distances, which satisfying the energy properties of rotational equivariance by construction. Moreover, SchNet adopts radial basis functions to avoid highly correlated filters. The filter-generating network is defined as follow:

$$f_{\text{FG}}(\mathbf{x}_j, \mathbf{r}_i, \mathbf{r}_j) = \mathbf{x}_j \cdot e_k(\mathbf{r}_i - \mathbf{r}_j) = \mathbf{x}_j \cdot \exp(-\gamma \|\mathbf{r}_i - \mathbf{r}_j\|_2 - \mu\|_2^2). \quad (9)$$

Similarly, for non-quantum properties prediction concerned in this work, we take the average of the node representations as the 3D molecular embedding:

$$\mathbf{z}^{3D} = \frac{1}{N} \sum_{i \in V} \mathbf{h}_i^{(K)}, \quad (10)$$

where K is the number of hidden layers.

PaiNN [42]. We use the PaiNN [42] as the encoder for the 3D geometries in QM9 dataset. PaiNN identify limitations of invariant representations in SchNet and extend the message passing formulation to rotationally equivariant representations, attaining a more expressive SE(3)-equivariant neural network model.

E Proof of Theoretical Analyses

Assumption 1 (Slow parameter updating) *Assume the learning rate is small enough, so that the parameter update $\Delta\theta_t = \theta_{t+1} - \theta_t$ is small for every time step, i.e. $\|\Delta\theta_t\| \leq \epsilon$, $\forall t \in \mathbb{N}$, ϵ is a small constant.*

Lemma 1. *With the assumption of slow parameter update, we can prove that $\|\xi_t - \theta_t\| \leq \frac{1-\beta}{\beta} \epsilon$.*

Proof.

$$\begin{aligned} \xi_t - \theta_t &= (1-\beta)\xi_{t-1} - (1-\beta)\theta_t \\ &= (1-\beta)(\xi_{t-1} - \theta_{t-1}) - (1-\beta)\Delta\theta_{t-1} \\ &= - \sum_{j=1}^t (1-\beta)^j \Delta\theta_{t-j}. \end{aligned} \quad (11)$$

For the first two equations, we respectively use the definition of EMA parameter update in equation 1 and the definition of $\Delta\theta$. For the third equation, we iteratively employed the results from the previous two steps, along with the initial condition $\xi_0 = \theta_0$. With Assumption 1, we have

$$\|\xi_t - \theta_t\| \leq \sum_{j=1}^t (1-\beta)^j \epsilon \leq \frac{1-\beta}{\beta} \epsilon \quad (12)$$

□

For the following results, we use the default setting in experiment $\beta = 0.5$, i.e. $\|\xi_t - \theta_t\| \leq \epsilon$.

Proposition 1 (Interpretation of loss discrepancy) *With Assumption 1, the loss discrepancy can be approximately expressed by the dot product between the data gradient and the “EMA gradient”:*

$$\mathcal{L}(\mathbf{x}, \xi_t) - \mathcal{L}(\mathbf{x}, \theta_t) = \alpha \nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta_t) \mathbf{v}_t^{EMA} + O(\epsilon^2), \quad (13)$$

where \mathbf{v}_t^{EMA} denotes $\sum_{j=1}^t (1-\beta)^j \nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}_{t-j}, \theta_{t-j})$, i.e. the weighted sum of the historical gradients, which we termed as “EMA gradient”.

Proof. From Lemma 1, since $\|\xi_t - \theta_t\|$ is small, we can use Taylor expansion of the loss function at θ_t :

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \xi_t) - \mathcal{L}(\mathbf{x}, \theta_t) &= \nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta_t) (\xi_t - \theta_t) + O(\|\xi_t - \theta_t\|^2) \\ &= \nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta_t) \sum_{j=1}^t (1-\beta)^j \nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}_{t-j}, \theta_{t-j}) + O(\|\epsilon\|^2), \end{aligned} \quad (14)$$

where we use equation 11 and the definition of online parameter update in equation 1. □

Proposition 2 (Gradient projection interpretation of MolPeg) *In the context of neglecting higher-order small quantities, we define $\mathcal{D}^+ \in \mathcal{D}$ and $\hat{\mathcal{D}}^+ \in \hat{\mathcal{D}}$ as samples that have positive dot products between the data gradient and the “EMA gradient”, then*

$$\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^+, \theta) = \nabla_{\theta} \mathcal{L}(\mathcal{D}^+, \theta) + a \mathbf{v}^{EMA} + c \mathbf{v}_{\perp}^{EMA}, a \geq 0, c \in \mathbb{R}. \quad (15)$$

Similarly, we define $\mathcal{D}^- \in \mathcal{D}$ and $\hat{\mathcal{D}}^- \in \hat{\mathcal{D}}$ as samples that have negative dot products, then

$$\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^-, \theta) = \nabla_{\theta} \mathcal{L}(\mathcal{D}^-, \theta) + b \mathbf{v}^{EMA} + d \mathbf{v}_{\perp}^{EMA}, b \leq 0, d \in \mathbb{R}. \quad (16)$$

Equality holds if and only if the absolute loss discrepancy $|\mathcal{L}(\mathbf{x}, \xi_t) - \mathcal{L}(\mathbf{x}, \theta_t)|$ across \mathcal{D}^+ and \mathcal{D}^- is uniform. This is a rare situation, and in such a case, our data selection strategy degenerates to random selection on \mathcal{D}^+ and \mathcal{D}^- .

Discussion about c and d is provided after the proof.

Proof. In the context of neglecting higher-order small quantities, MolPeg selects data with large loss discrepancies, meaning large dot products by using Proposition 1. That is $\forall \mathbf{x} \in \hat{\mathcal{D}}$ and $\forall \mathbf{x}' \in \mathcal{D} \setminus \hat{\mathcal{D}}$, we have

$$|\nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta) \cdot \mathbf{v}^{EMA}| \geq |\nabla_{\theta} \mathcal{L}(\mathbf{x}', \theta) \cdot \mathbf{v}^{EMA}|. \quad (17)$$

Then for samples in \mathcal{D}^+ and $\hat{\mathcal{D}}^+$, we have

$$\frac{1}{|\hat{\mathcal{D}}^+|} \sum_{x \in \hat{\mathcal{D}}^+} \nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta) \cdot \mathbf{v}^{EMA} \geq \frac{1}{|\mathcal{D}^+|} \sum_{x \in \mathcal{D}^+} \nabla_{\theta} \mathcal{L}(\mathbf{x}', \theta) \cdot \mathbf{v}^{EMA} > 0 \quad (18)$$

That is $\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^+, \theta) \cdot \mathbf{v}^{EMA} \geq \nabla_{\theta} \mathcal{L}(\mathcal{D}^+, \theta) \cdot \mathbf{v}^{EMA} > 0$ for short. Thus when projecting $\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^+, \theta) - \nabla_{\theta} \mathcal{L}(\mathcal{D}^+, \theta)$ on \mathbf{v}^{EMA} , the coefficient a is $(\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^+, \theta) - \nabla_{\theta} \mathcal{L}(\mathcal{D}^+, \theta)) \cdot \mathbf{v}^{EMA} \geq 0$.

Similarly,

$$\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^-, \theta) \cdot \mathbf{v}^{EMA} \leq \nabla_{\theta} \mathcal{L}(\mathcal{D}^-, \theta) \cdot \mathbf{v}^{EMA} < 0 \quad (19)$$

Then $c \triangleq (\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^-, \theta) - \nabla_{\theta} \mathcal{L}(\mathcal{D}^-, \theta)) \cdot \mathbf{v}^{EMA} \leq 0$.

The condition for $a = 0$ ($c = 0$) is that the equality in equation 17 holds for samples in \mathcal{D}^+ (\mathcal{D}^-). \square

Since our selection strategy does not constrain the direction perpendicular to the EMA gradient, we consider a simplified model where b and d are treated as random variables with an expectation of zero. Consequently, in the sense of expectation, equation 4 and equation 5 hold. The feasibility of this simplified model is demonstrated as follows. Assume that $\nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta) \cdot \mathbf{v}_{\perp}^{EMA}$ for all samples are independent and identically distributed random variables with expectation μ and variance σ^2 . When the sample sizes $|\mathcal{D}^+|$ and $|\hat{\mathcal{D}}^+|$ are sufficiently large, the central limit theorem implies that $\frac{1}{|\mathcal{D}^+|} \sum_{x \in \mathcal{D}^+} \nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta) \cdot \mathbf{v}_{\perp}^{EMA}$ is approximately a Gaussian distribution $\mathcal{N}\left(\mu, \frac{\sigma^2}{|\mathcal{D}^+|}\right)$, and similarly, $\frac{1}{|\hat{\mathcal{D}}^+|} \sum_{x \in \hat{\mathcal{D}}^+} \nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta) \cdot \mathbf{v}_{\perp}^{EMA}$ is approximately a Gaussian distribution $\mathcal{N}\left(\mu, \frac{\sigma^2}{|\hat{\mathcal{D}}^+|}\right)$. The expectation of their difference $\mathbb{E}c = \mathbb{E}\nabla_{\theta} \mathcal{L}(\hat{\mathcal{D}}^+, \theta) \cdot \mathbf{v}_{\perp}^{EMA} - \mathbb{E}\nabla_{\theta} \mathcal{L}(\mathcal{D}^+, \theta) \cdot \mathbf{v}_{\perp}^{EMA} = 0$. Similarly, we can prove $\mathbb{E}d = 0$.

F Connections to Existing DP Methods

F.1 MolPeg & GraNd [19]

In the pretraining scenario, where the initialization is fixed, the GraNd score is defined as the norm of the gradient $\|\nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta_t)\|$. With Assumption 1, we can deduce $\|\xi_t - \theta_t\| \leq \epsilon$ as shown in equation 12, then the data selected by MolPeg satisfies $\delta \leq |\mathcal{L}(\mathbf{x}, \theta_t) - \mathcal{L}(\mathbf{x}, \xi_t)| = |\nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta_t)(\theta_t - \xi_t) + O(\epsilon^2)| \leq \epsilon \|\nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta_t)\| + O(\epsilon^2)$. The data we select has a lower bound on the GraNd score $\|\nabla_{\theta} \mathcal{L}(\mathbf{x}, \theta_t)\| \geq O(\frac{\delta}{\epsilon})$, making it more likely to be chosen by the GraNd score.

F.2 MolPeg & Infobatch [46]

Our strategy employs relative loss scales rather than absolute values, enabling a more flexible adaptation for transfer scenarios. For simple downstream samples for pretraining model, where $\mathcal{L}(\mathbf{x}, \xi_t)$ is small, both Infobatch and MolPeg eliminate samples with small online loss which are regarded as redundant for finetuning. However, for difficult samples for pretraining model, where $\mathcal{L}(\mathbf{x}, \xi_t)$ is large, our method diverges from Infobatch by preserving the crucial samples for transfer learning.

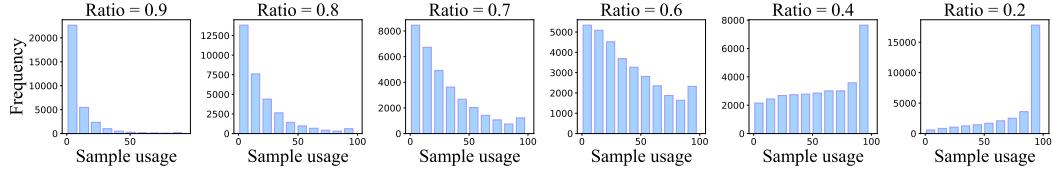
F.3 MolPeg & Forgetting [16]

If we consider classification tasks and use accuracy loss, our method tends to select samples near the classification boundary. This can be related to the forgetting method, which aims to select samples that have been forgotten (i.e., initially classified correctly and then incorrectly) multiple times. For simplicity, let's explain this in the context of binary classification under Assumption 1. Further assume the class prediction probability f is l -Lipschitz continuous with respect to the parameters θ , where $f = (f^{(0)}, f^{(1)})$ and $f^{(0)} + f^{(1)} = 1$, we have $\|f(\mathbf{x}, \theta) - f(\mathbf{x}, \xi)\| \leq l\epsilon$. The loss function $\mathcal{L}(\mathbf{x}, \theta) = |\text{argmax}_i\{f(\mathbf{x}, \theta)^{(i)}\} - y|$, $y \in \{0, 1\}$ is not continuous at the classification boundary where $f^{(0)}(\mathbf{x}, \theta) = f^{(1)}(\mathbf{x}, \theta) = 0.5$. Consequently, only when the sample is located near the classification boundary $f^{(0)}(\mathbf{x}, \theta) \in (0.5 - l\epsilon/\sqrt{2}, 0.5 + l\epsilon/\sqrt{2})$, exhibit a non-zero loss discrepancy.

G Additional Empirical Results

G.1 The frequency of sample usage

Figure 7: Statics of frequency of sample usage in HIV datasets. The x-axis represents the number of times samples are used throughout the entire training process



Unlike static DP, although we use a fixed proportion of samples in each iteration, we monitor the training dynamics of the entire dataset. Our method naturally avoids the issue of completely ignoring crucial samples, as almost **all samples** are used to varying degrees. Coordinating the use of samples in each iteration to achieve better generalization is the main contribution of MolPeg. In Figure 7, we visualize the frequency of sample usage in the HIV datasets, showing that almost all samples are used for training even with an aggressive pruning ratio adopted, with crucial samples being used more frequently.

G.2 Robustness evaluation of different pretraining datasets

Table 5: The performance comparison on HIV with different pre-training datasets of varying quality in terms of ROC-AUC (%), \uparrow)

Pre-train Dataset		QM9 (Low quality)			ZINC (medium quality)		
	Pruning Ratio %	90	70	40	90	70	40
Static	Hard Random	74.8 _[7.9]	78.1 _[4.6]	80.5 _[2.2]	78.2 _[6.0]	80.9 _[3.3]	83.1 _[1.1]
	Forgetting	76.5 _[6.2]	79.4 _[3.3]	80.9 _[1.8]	79.6 _[4.6]	81.5 _[2.7]	83.7 _[0.5]
	GraNd	77.3 _[5.4]	80.2 _[2.5]	80.8 _[1.9]	80.2 _[4.0]	81.7 _[2.5]	83.9 _[0.3]
	Glister	75.9 _[6.8]	78.6 _[4.1]	81.1 _[1.6]	78.6 _[5.6]	81.2 _[3.0]	83.5 _[0.7]
Dynamic	Soft Random	79.1 _[3.6]	81.3 _[1.4]	82.5 _[0.2]	81.8 _[2.4]	83.2 _[1.0]	84.0 _[0.2]
	UCB	79.5 _[3.2]	81.8 _[0.9]	82.5 _[0.2]	82.2 _[2.0]	83.1 _[1.1]	83.9 _[0.3]
	InfoBatch	80.5 _[2.2]	81.7 _[1.0]	83.1 _[0.4]	82.5 _[1.7]	83.8 _[0.4]	84.5 _[0.3]
	MolPeg	81.5 _[1.2]	82.9 _[0.2]	83.9 _[1.2]	83.2 _[1.0]	84.5 _[0.3]	85.6 _[1.4]
Whole Dataset		82.7 \pm 0.5			84.2 \pm 0.2		
Train from Scratch		83.8 \pm 0.2					

G.3 Results on MUV dataset

In Table 5, we have supplemented with additional experiments on the MUV dataset, following the same experimental setup described in Section 5.1. We observe that MolPeg still achieves state-of-the-art performance on the MUV dataset, further validating the effectiveness of our method.

Table 6: The performance comparison to state-of-the-art methods on MUV in terms of ROC-AUC (%), \uparrow .

MUV	Static Pruning				Dynamic Pruning			
	Pruning Ratio %	Random	Forgetting	GraNd-20	Glister	Random	UCB	InfoBatch
90	73.8 _{±6.4}	75.3 _{±4.9}	75.6 _{±4.6}	75.5 _{±4.7}	76.3 _{±3.9}	76.8 _{±3.4}	78.0 _{±2.2}	79.8 _{±0.4}
70	75.7 _{±4.5}	76.8 _{±3.4}	77.2 _{±3.0}	77.6 _{±2.6}	77.9 _{±2.3}	78.2 _{±2.0}	79.1 _{±1.1}	80.7 _{±0.5}
40	78.4 _{±1.8}	78.5 _{±1.7}	77.9 _{±2.3}	78.3 _{±1.9}	79.0 _{±1.2}	79.5 _{±0.7}	80.5 _{±0.3}	81.2 _{±1.0}
Whole Dataset					80.2 \pm 0.2			

H Discussions

Limitations and future works. Our data pruning strategy is specifically designed for molecular downstream tasks, but source-free data pruning is a task setting with broad applications in other fields as well. For example, in large language models (LLMs) and heavy-weight models, pretraining data is often difficult for users to obtain or even kept confidential [74–77]. However, we have not validated our method in these more general scenarios. Therefore, verifying the effectiveness of MolPeg in more general tasks is one of our future research directions, e.g., graph, natural language and vision scenarios. Additionally, as the first work designed for the source-free data pruning setting, we have only made simple attempts at perceiving upstream and downstream knowledge via loss discrepancy. In the future, we will explore how to better utilize knowledge from both the source and target domains to achieve data pruning, which leaves significant potential to be explored.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification:

Guidelines: It accurately reflect our contribution and scope.

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss the limitation in Appendix [H](#).

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We provide the complete proof in Appendix [E](#).

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide our codes and README file in supplementary materials to ensure reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: We provide our codes and README file in supplementary materials to ensure reproducibility.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: **[Yes]**

Justification: We provide the implementation details in Section 4.2

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: **[Yes]**

Justification: We report the error bars in Section 5.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: We provide the computational resources in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: We provide the discussion of broader impacts in Appendix H.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly respect and credit the existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We provide asset with a CC-BY 4.0 license.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.