GraphSculptor: Sculpting Pre-training Coreset for Graph Self-supervised Learning

Chuang Liu[1], **Zelin Yao**[2], **Xueqi Ma**[3], **Luzhi Wang**[1*], **Mukun Chen**[4]
Pinghua Xu[5] and **Wenbin Hu**[2,6,7]
[1]College of Artificial Intelligence, Dalian Maritime University, Dalian, China
[2]School of Computer Science, Wuhan University, Wuhan, China
[3]School of Computing and Information Systems, The University of Melbourne, Melbourne, Australia
[4]School of Advanced Interdisciplinary Studies, Hunan University of Technology and Business, Changsha, China
[5]Sangfor Technologies Inc., Shenzhen, China
[6]Wuhan University Shenzhen Research Institute, Shenzhen, China
[7]Hubei Jiangxia Laboratory, Wuhan, China
{chuangliu, zelinyao, cmk0910, xupinghua, hwb}@whu.edu.cn, xueqim@student.unimelb.edu.au, wangluzhi0@gmail.com

Abstract

Graph self-supervised learning (SSL) typically relies on large-scale unlabeled datasets, heavily inflating computational costs. However, empirical evidence suggests that these datasets contain substantial redundancy—our analysis reveals that uniformly subsampling 50% of graphs retains over 96% of downstream performance. To exploit this redundancy, we introduce GraphSculptor for pre-training coreset construction. Unlike methods dependent on additional training-time signals or limited solely to topological statistics, GraphSculptor provides a label-free solution that constructs coresets via two complementary perspectives: intrinsic structure and contextual semantics. Concretely, structural diversity is quantified using intrinsic graph statistics, yielding a structural feature vector for each graph, while semantic diversity is captured by utilizing a pre-trained language model to encode descriptions generated via graph-to-text. GraphSculptor integrates these signals into a unified metric space and performs cluster-aware selection to preserve joint structural–semantic diversity. We further derive a theoretical bound on the loss gap between coreset and full-data pre-training, offering theoretical motivation for our selection formulation. Extensive experiments demonstrate that GraphSculptor effectively "sculpts" the dataset: a 10% coreset achieves 99.6% of full-data performance while reducing pre-training time by nearly 90%, offering a scalable solution for data-efficient graph pre-training.

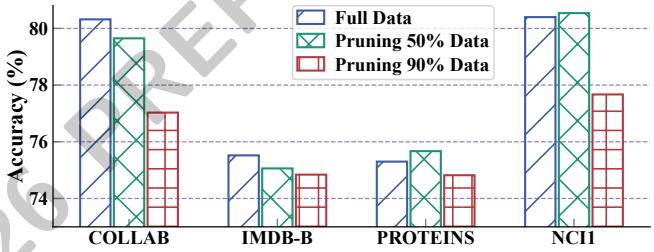

Figure 1: **Substantial redundancy in graph SSL datasets.** We pre-train GraphMAE on uniformly subsampled subsets (retaining 50% and 10% of graphs) and evaluate on four benchmark datasets. Performance degrades only mildly under random pruning; in particular, when retaining 50% of the pre-training graphs, the worst-case relative drop across these datasets is below 4%. This observation suggests that a sizeable fraction of the dataset provides marginal utility for pre-training and motivates principled coreset curation.

1 Introduction

Graph self-supervised learning (SSL) has become a cornerstone for representation learning on graph-structured data, achieving strong performance across domains from molecular science to social network analysis [You *et al.*, 2020; Xia *et al.*, 2022a; Hou *et al.*, 2022]. Recent progress is often driven by scaling up unlabeled pre-training datasets, implicitly assuming that more pre-training graphs will reliably translate into better representations. In this work, we revisit this assumption and show that large graph datasets can contain substantial redundancy, motivating a more compute-efficient, data-centric alternative.

A simple diagnostic (Figure 1) shows that graph SSL is empirically robust to uniform random pruning: removing 50% of pre-training graphs preserves over 96% of downstream transfer performance. This indicates diminishing returns of data scaling and substantial redundancy, raising the question: *can*

*Corresponding Author

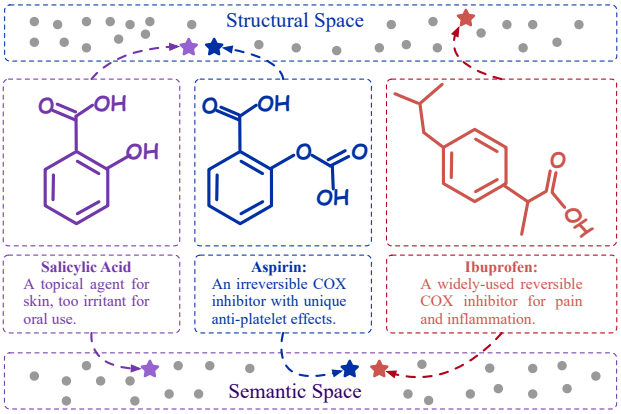

Figure 2: **Structure vs. semantics can disagree.** Molecules that are structurally similar (*e.g.*, Aspirin vs. Salicylic Acid) may be semantically distant in terms of their textual descriptions and associated functions, while structurally dissimilar ones (*e.g.*, Aspirin vs. Ibuprofen) can be semantically closer. This motivates measuring diversity beyond topology by incorporating semantic embeddings derived from graph-to-text.

we identify a much smaller subset that remains representative of the full pre-training dataset?

To answer this question, we take a data-centric view of graph SSL: we aim to improve pre-training efficiency by selecting informative pre-training graphs instead of increasing data scale. Prior efforts to reduce training data often rely on training-time signals such as gradients or task supervision to identify "important" examples [Xu *et al.*, 2023a; Zhang *et al.*, 2024]. However, in graph SSL pre-training, labels are typically unavailable, and computing gradient-based criteria can introduce additional optimization overhead and may be tied to specific model/optimizer choices, potentially limiting the achievable wall-clock savings and generality. To this end, we introduce **GraphSculptor**, a label-free and model-agnostic framework that extracts an information-rich coreset from a large pre-training collection. GraphSculptor measures diversity from two complementary views—structure via intrinsic graph statistics and semantics via graph-to-text followed by a frozen text encoder—and then selects a coreset that preserves joint structural–semantic diversity under a target budget. Our objective is to substantially reduce pre-training cost while maintaining downstream transfer performance comparable to full-data pre-training.

The core idea of GraphSculptor is to curate pre-training data using a dual-view notion of diversity. We characterize each graph from two complementary perspectives: **structure** and **semantics**. For the structural view, we compute a compact vector of intrinsic, multi-scale graph statistics and aim to cover diverse topological regimes, rather than over-representing the most frequent patterns. At the same time, structure alone may be insufficient: in many domains, graphs with similar topology can correspond to different domain properties (*e.g.*, molecules sharing similar scaffolds may have different biological activities; see Figure 2). To capture such domain-level variation, we map each graph to a textual description (graph-to-text) and encode it with a pre-trained language model, yielding a semantic embedding that serves as a complementary proxy beyond topology-based summaries.

Given the fused structural and semantic embeddings, GraphSculptor selects a budgeted coreset in a cluster-aware manner. Specifically, it clusters graphs in the joint space and then selects representative graphs across clusters, prioritizing clusters that are both diverse internally and well separated from others, to preserve joint structural–semantic diversity under the target budget. The selection in GraphSculptor is performed once per pre-training dataset as an offline preprocessing step: we compute structural/semantic embeddings for all graphs and derive a fixed coreset that can be reused across pre-training runs. This one-time cost is amortized when the same dataset is used to pre-train different models or under different SSL objectives, and it avoids any training-time gradient-based scoring, making the procedure label-free, model-agnostic, and easy to cache and deploy. Our main contributions are:

- We propose GraphSculptor, a label-free and model-agnostic coreset construction framework for graph SSL that jointly considers structural statistics and semantic embeddings.
- We provide a loss-gap bound for coreset pre-training versus full-data pre-training, supporting our selection principle.
- Experiments show strong efficiency–accuracy trade-offs: using 10% of the pre-training graphs retains up to 99.6% of full-data performance while reducing pre-training time by nearly 90%.

2 Related Work

Graph Pretraining. Self-supervised graph pre-training has become a standard approach for leveraging unlabeled graphs and mitigating label scarcity across domains [Xia *et al.*, 2022b]. Existing methods broadly fall into two paradigms: contrastive pre-training, which aligns representations across augmented views to learn invariances [You *et al.*, 2020; Xia *et al.*, 2022a], and generative pre-training, which learns by reconstructing masked components of graphs [Li *et al.*, 2023; Liu *et al.*, 2026; Liu *et al.*, 2024]. The latter line, represented by GraphMAE and its variants [Hou *et al.*, 2022; Hou *et al.*, 2023], is often adopted due to its relatively simple training pipeline without explicit negative sampling. Subsequent work extends the reconstruction targets to richer structural signals (*e.g.*, edges and degrees) [Li *et al.*, 2023; Tan *et al.*, 2023; Shi *et al.*, 2023; Liu *et al.*, 2025] or combines reconstruction with contrastive objectives [Liu *et al.*, 2023; Zheng and Jia, 2024; Wang *et al.*, 2024; Shen *et al.*, 2025]. While these approaches differ in objectives and architectures, they typically pre-train on the full available dataset and benefit from larger-scale data when feasible. This practice can incur substantial computational cost and limits the iteration speed and scalability of graph pre-training. In contrast, our work takes a data-centric perspective and asks how to curate a compact yet informative pre-training subset that preserves the utility of full-dataset pre-training. Therefore, our method is orthogonal to the choice of pre-training objective and can be applied on top of existing graph pre-training frameworks.

Graph Data Reduction. The high computational cost of learning on large-scale graph datasets has motivated research

on reducing training data, broadly including graph dataset distillation/condensation and graph selection. Distillation methods such as GCond [Jin *et al.*, 2022b] and DosCond [Jin *et al.*, 2022a] aim to synthesize a compact surrogate dataset, often formulated as bilevel optimization to match training dynamics (*e.g.*, gradients). While effective in some settings, these formulations can be computationally demanding and sensitive to modeling choices, prompting more tractable alternatives: KIDD [Xu *et al.*, 2023b] replaces gradient matching with kernel-based objectives, SGDD [Yang *et al.*, 2023] emphasizes preserving structural properties, and MIRAGE [Gupta *et al.*, 2024] condenses representative substructures with improved model-agnosticism. Graph selection, which directly chooses a subset from the original dataset, is closer to our setting. Many existing selection approaches adopt dynamic in-training strategies that iteratively update the subset using model-dependent signals, such as training loss [Chen *et al.*, 2024], gradient norms [Zhang *et al.*, 2024], or predictive uncertainty [Xu *et al.*, 2023a]. Although these signals can be informative, the resulting procedures are tightly coupled to specific training dynamics and typically require repeated evaluations over training, increasing overhead and reducing their suitability as a one-shot, model-agnostic preprocessing step [Li *et al.*, 2025]. In contrast, GraphSculptor performs one-shot, offline coreset selection without training-time signals, enabling label-free and model-agnostic preprocessing before pre-training.

3 Method

GraphSculptor is a data-centric framework for constructing a compact pre-training coreset from a large unlabeled graph dataset. As illustrated in Figure 3, it follows two stages: (1) **Holistic Graph Representation**, which computes for each graph a fused representation that combines structural and semantic views; and (2) **Principled Coreset Curation**, which selects a budgeted subset via cluster-based selection in the fused space to preserve structural–semantic diversity. In particular, the semantic view is obtained by mapping each graph to a textual description (graph-to-text) and encoding it with a frozen text encoder, providing signals complementary to topology-based statistics. The entire procedure is performed offline as a one-time preprocessing step before SSL pre-training.

Preliminaries. We consider an unlabeled graph dataset $\mathcal{D} = \{G_i\}_{i=1}^{M}$, where each graph $G_i = (\mathcal{V}_i, \mathcal{E}_i)$ is associated with an adjacency matrix $\mathbf{A}_i$ and node features $\mathbf{X}_i$. Our goal is to select a coreset $\mathcal{D}_{\text{core}} \subset \mathcal{D}$ with a fixed budget $|\mathcal{D}_{\text{core}}| = M_{\text{target}} = \lfloor p_{\text{target}}\, M \rfloor$, where $p_{\text{target}} \in (0, 1]$ denotes the retention ratio. We construct $\mathcal{D}_{\text{core}}$ to remain representative of $\mathcal{D}$ in the proposed fused (structural–semantic) representation space under this budget, *i.e.*, to preserve the diversity of graphs as measured in that space.

3.1 Holistic Graph Representation

We represent each graph by jointly capturing (i) intrinsic topology and (ii) semantic context. Concretely, for each G_i we compute a structural feature vector $\mathbf{z}_{S,i} \in \mathbb{R}^{d_S}$ and a semantic embedding $\mathbf{z}_{T,i} \in \mathbb{R}^{d_T}$, where d_S and d_T denote their respective feature dimensions. We then combine them into a fused representation used for downstream clustering and coreset selection.

Structural Characterization

To capture structural diversity, we characterize each graph using a compact set of graph-level descriptors. The design goal is to balance scalability and discriminative power: the descriptors should be inexpensive to compute at dataset scale while still reflecting salient structural variations relevant to data curation. We therefore adopt a modular structural feature extractor that can incorporate domain-specific descriptors when beneficial, while using a lightweight default configuration in our main experiments.

For each graph G_i, we define a structural feature map $\psi_S : \mathcal{G} \to \mathbb{R}^{d_S}$ that outputs a fixed-dimensional descriptor $\mathbf{z}_{S,i}$. Specifically,

$$\mathbf{z}_{S,i} = \psi_S(G_i) = [\phi_1(G_i), \phi_2(G_i), \ldots, \phi_{d_S}(G_i)]^\top, \quad (1)$$

where each $\phi_j : \mathcal{G} \to \mathbb{R}$ computes a scalar structural statistic. Concretely, ψ_S is instantiated with a streamlined set of descriptors, engineered to strike an optimal balance between discriminative expressivity and computational scalability: (i) Basic Topological Invariants. We explicitly encode the fundamental scale and sparsity of each graph via its node count $|\mathcal{V}_i|$, edge count $|\mathcal{E}_i|$, and average degree $2|\mathcal{E}_i|/|\mathcal{V}_i|$. (ii) Global Positional Encodings. To transcend local aggregation metrics and capture long-range structural dependencies, we incorporate Positional Encodings. Specifically, we leverage random-walk based positional identifiers [Ying *et al.*, 2021], which provide a compact yet informative signature of global connectivity, ensuring the structural feature vector remains distinct across topologically diverse graphs.

The resulting $\mathbf{z}_{S,i}$ serves as the structural view used in the fused representation. We use a lightweight default configuration in our main experiments for efficiency, and ψ_S can be extended with additional domain-specific descriptors when they are available and useful.

Semantic Contextualization

Beyond structure, many graph datasets carry domain semantics that may not be fully captured by structural descriptors alone. For molecular graphs, compounds with highly similar scaffolds can exhibit markedly different biochemical effects (*e.g.*, Aspirin vs. Salicylic Acid). Similarly, in citation networks, two papers may have nearly identical local citation neighborhoods (*e.g.*, comparable in-/out-degree and co-citation patterns) while belonging to entirely different research topics. To incorporate this complementary signal, we leverage natural-language descriptions by mapping each graph to text and then encoding the text with a pre-trained text encoder, yielding a semantic embedding that is orthogonal to purely structural statistics. We define a semantic feature map as a two-stage composition:

$$\psi_T \;=\; f_{\text{T2E}} \circ f_{\text{G2T}}, \quad (2)$$

which maps each graph to a dense semantic embedding.

Graph-to-Text conversion (f_{G2T}). We map each graph G_i to a short textual descriptor $t_i = f_{\text{G2T}}(G_i)$ using domain-specific instantiations. For molecular graphs, we leverage the available SMILES strings and employ a biomedical text

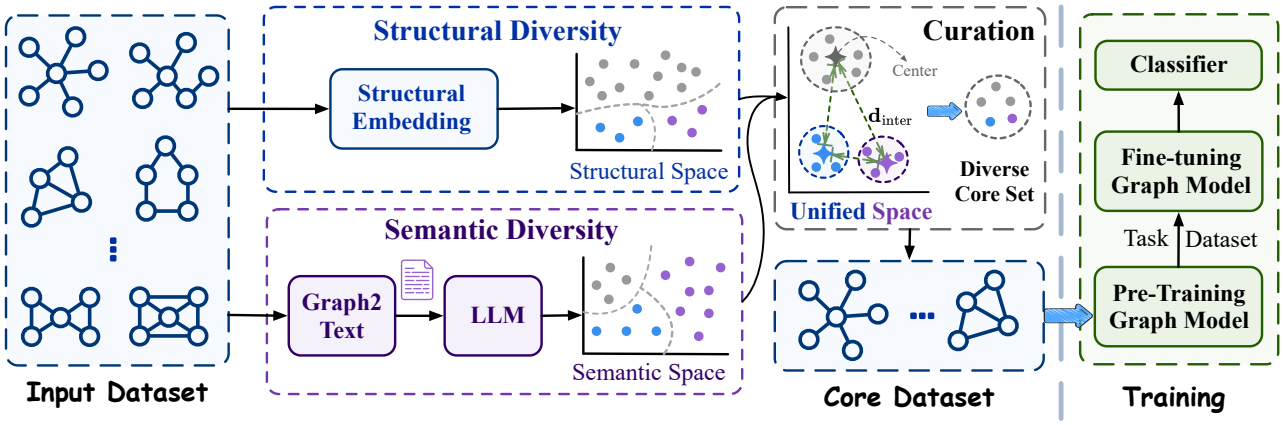

Figure 3: **The GraphSculptor workflow.** We construct a coreset by jointly optimizing for Structural Diversity and Semantic Diversity. These are fused into a unified space where our curation module selects an information-rich subset for efficient and effective pre-training.

generation model (*e.g.*, BioT5 [Pei *et al.*, 2023]) to generate a concise title/description. For social-network graphs, where no canonical textual field is provided, we first serialize the graph into an adjacency-list (edge-list) representation with a fixed, deterministic ordering (*e.g.*, sorting nodes and neighbors by index to reduce permutation effects). We then prepend a fixed dataset-level background description that specifies only the domain context (without labels or task information), and use a general-purpose LLM (*e.g.*, ChatGPT or Gemini) to produce a bounded-length, sociology-oriented title. We use a fixed prompt template and deterministic decoding, and cache the generated texts to ensure reproducibility.

Text-to-Embedding encoding (f_{T2E}). We encode t_i with a pre-trained text encoder to obtain a semantic embedding. In our main experiments on biochemical graphs, we use SciBERT [Beltagy *et al.*, 2019] as the encoder; for general text, standard encoders such as BERT [Devlin *et al.*, 2019] can be used. Formally, given a text encoder $\text{Encoder}(\cdot)$, we use the final-layer hidden state of the special classification token `[CLS]` as a pooled representation:

$$\mathbf{z}_{T,i} = f_{\text{T2E}}(t_i) = \text{Encoder}(t_i)_{[\text{CLS}]} \in \mathbb{R}^{d_T}. \tag{3}$$

During coreset construction, the text encoder is kept frozen to avoid introducing any training-time signals and to keep the procedure lightweight. The resulting embeddings provide a semantics-aware proxy derived from natural language, complementing structural descriptors by capturing descriptive similarity that may not be reflected by topology alone.

Unified Representation Space

To enable holistic coreset selection, we project the distinct structural and semantic signals into a unified metric space. Direct fusion of the structural descriptor $\mathbf{z}_{S,i} \in \mathbb{R}^{d_S}$ and the semantic embedding $\mathbf{z}_{T,i} \in \mathbb{R}^{d_T}$ is non-trivial due to their disparate modalities and potentially incommensurable magnitudes. To prevent high-variance features from dominating the distance computation, we employ feature-wise standardization (z-score normalization) across the dataset, yielding normalized descriptors $\hat{\mathbf{z}}_{S,i}$ and $\hat{\mathbf{z}}_{T,i}$. We then synthesize the holistic representation $\mathbf{h}_i$ via concatenation:

$$\mathbf{h}_i = \text{Concat}(\hat{\mathbf{z}}_{S,i},\, \hat{\mathbf{z}}_{T,i})\,. \tag{4}$$

In this unified space, the Euclidean distance aggregates both structural similarity and semantic affinity, preventing modal dominance and ensuring that coreset selection targets joint structural–semantic diversity.

3.2 Principled Coreset Curation

We construct the coreset via a cluster-aware selection scheme in the fused space. We first partition graphs by K-means, then prioritize clusters using a diversity score, and finally select representative graphs within each cluster under a fixed budget.

❶ Unsupervised Clustering in the Fused Space. Given fused representations $\{\mathbf{h}_i\}_{i=1}^{M}$, we apply K-means clustering and obtain clusters $\mathcal{C} = \{C_1, \ldots, C_K\}$ with centroids $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ by minimizing the within-cluster squared Euclidean distance:

$$\arg\min_{\mathcal{C}} \sum_{k=1}^{K} \sum_{\mathbf{h}_i \in C_k} \|\mathbf{h}_i - \boldsymbol{\mu}_k\|_2^2\,. \tag{5}$$

❷ Diversity Scoring. An effective coreset should balance *mass coverage* (so that high-density regions of the empirical distribution are not under-represented) and *diversity* to avoid collapsing onto frequent but low-informative patterns. To combine these factors in a simple and scale-robust manner, we define the cluster score in the log domain:

$$\Omega_j = \log(\pi_j) + w \log(d_{\text{intra},j}) + (1-w)\log(d_{\text{inter},j}), \tag{6}$$

where $\pi_j = |C_j|/M$ captures the cluster mass and $w \in [0,1]$ controls the trade-off between within-cluster dispersion and cross-cluster separation. This log-domain formulation ensures numerical stability and balances the magnitude differences between density and structural metrics. We next normalize these weights to obtain cluster-wise sampling proportions.

❸ Cluster-wise Selection. We convert $\{\Omega_j\}$ into cluster-wise sampling proportions:

$$\tilde{\Omega}_j = \frac{\exp(\Omega_j/\tau)}{\sum_{k=1}^{K}\exp(\Omega_k/\tau)}, \qquad \tau > 0, \tag{7}$$

where we use $\tau = 0.5$ by default. We then assign each cluster a quota n_j such that $\sum_{j=1}^{K} n_j = M_{\text{target}}$; in implementation, we set $n_j = \lfloor M_{\text{target}} \tilde{\Omega}_j \rfloor$ for $j = 1, \ldots, K-1$ and allocate the remainder to the last cluster.

Within each cluster C_j, we perform centroid-biased sampling to obtain $\mathcal{S}_j$. For $G_i \in C_j$, we define

$$p_{i|j} = \frac{\exp\left(-\frac{\|\mathbf{h}_i - \boldsymbol{\mu}_j\|_2^2}{2\sigma_j^2}\right)}{\sum_{i' \in C_j} \exp\left(-\frac{\|\mathbf{h}_{i'} - \boldsymbol{\mu}_j\|_2^2}{2\sigma_j^2}\right)}. \quad (8)$$

and sample n_j graphs without replacement from C_j according to $\{p_{i|j}\}$:

$$\mathcal{D}_{\text{core}} = \bigcup_{j=1}^{K} \mathcal{S}_j, \qquad \mathcal{S}_j \sim \text{WRS}\big(C_j, n_j; \{p_{i|j}\}\big), \quad (9)$$

where $\text{WRS}(\cdot)$ denotes weighted random sampling without replacement, *i.e.*, drawing n_j samples from C_j with probabilities proportional to $\{p_{i|j}\}$. Since global coverage is mainly ensured by the cross-cluster budget allocation (Eq. (6)), this within-cluster step favors typical instances near the centroid while retaining randomness. The temperature σ_j controls the sampling sharpness, interpolating between uniform sampling ($\sigma_j \to \infty$) and hard nearest-to-centroid selection ($\sigma_j \to 0$).

3.3 Theoretical Analysis

We provide theoretical motivation for our cluster-aware allocation by analyzing optimization under a standard NTK-style linearization around the initialization W_0. Here, W denotes the parameters of the pre-training model and W_0 is their initialization. Concretely, we consider the linearized per-sample loss $\tilde{\ell}(W; G) = \ell(W_0; G) + \langle \nabla \ell(W_0; G), W - W_0 \rangle$, so that the optimization dynamics are governed by initialization gradients. Under L-smoothness and bounded-gradient assumptions, we bound the expected pre-training loss gap between training on the selected coreset and training on the full dataset. The bound highlights the role of within-cluster gradient variability (heterogeneity), suggesting that clusters with larger variability should receive higher sampling proportions.

Theorem 1 (Expected Pre-training Loss Gap under Linearization). *Let $\mathcal{L}(W) = \frac{1}{M}\sum_{i=1}^{M} \ell(W; G_i)$ be an L-smooth loss over $\mathcal{D} = \{G_1, \ldots, G_M\}$. Assume $\|\nabla \ell(W_0; G)\|_F \le B_g$ for all $G \in \mathcal{D}$ and the linearized regime defined by $\tilde{\ell}(\cdot;\cdot)$ above. Suppose $\mathcal{D}$ is partitioned into K disjoint clusters $\{C_k\}_{k=1}^{K}$ with proportions $\pi_k = |C_k|/M$. Let M_{target} be the coreset size, and let M_k be the number of selected samples from cluster C_k, inducing $q_k = M_k/M_{target}$ (with $q_k > 0$ and $\sum_k q_k = 1$). Define the within-cluster gradient variance at W_0 as*

$$\mathcal{V}_k^2 = \mathbb{E}_{G \in C_k}\Big[\big\|\nabla \ell(W_0; G) - \mathbb{E}_{G' \in C_k}[\nabla \ell(W_0; G')]\big\|_F^2\Big].$$

Let $W_{full}^{(T)}$ and $W_{core}^{(T)}$ denote the parameters after T gradient steps with learning rate η when optimizing on the full dataset and on the sampled coreset, respectively. Then the expected pre-training loss gap satisfies:

$$\mathbb{E}_{\mathcal{D}_{core}}\Big[\mathcal{L}(W_{core}^{(T)}) - \mathcal{L}(W_{full}^{(T)})\Big] \le \frac{LT^2\eta^2}{2\,M_{target}} \sum_{k=1}^{K} \frac{\pi_k^2\,\mathcal{V}_k^2}{q_k}. \quad (10)$$

Remark. Minimizing the upper bound in Eq. (10) subject to $\sum_{k=1}^{K} q_k = 1$ yields $q_k^\star \propto \pi_k \mathcal{V}_k$, *i.e.*, larger clusters with higher gradient variability should receive more budget.

Discussion. In practice, $\mathcal{V}_k$ is not available without training-time gradient evaluations. We therefore adopt offline, representation-based proxies computed once in our fused structural–semantic space. Under a mild regularity condition that initialization gradients vary smoothly with the fused representation (*e.g.*, $\|\nabla \ell(W_0; G_i) - \nabla \ell(W_0; G_{i'})\|_F \le L_h \|\mathbf{h}_i - \mathbf{h}_{i'}\|_2$), the within-cluster gradient variability can be controlled by within-cluster dispersion in the fused space, motivating the mass–intra component $\log(\pi_k) + w \log(d_{\text{intra},k})$. We further incorporate inter-cluster separation $d_{\text{inter},k}$ as a practical correction to discourage allocating excessive budget to highly overlapping clusters, leading to our score Ω_k (Eq. (6)). Sampling proportions are then obtained via temperature-scaled softmax, implementing the mass–variability principle suggested by the theory while remaining label-free and avoiding training-time signals.

4 Experiments

4.1 Experimental Setup

Datasets and Tasks. Our evaluation covers two learning paradigms. **Pre-training and Transfer Learning.** For graph SSL, we pre-train on ZINC, which contains ∼250,000 commercially available molecules [Sterling and Irwin, 2015]. We then transfer the pre-trained models to eight binary classification tasks from MoleculeNet [Wu *et al.*, 2018]. For all MoleculeNet tasks, we use the standard scaffold splits to evaluate generalization. **Supervised Learning.** To assess the generality of our selection strategy beyond SSL pre-training, we additionally conduct supervised graph classification on large-scale OGB molecular benchmarks, OGBG-MOLHIV and OGBG-MOLPCBA [Hu *et al.*, 2020], following the official data splits provided by OGB. We further include two social graph benchmarks from TUDataset [Morris *et al.*, 2020], COLLAB and IMDB-BINARY, using the standard evaluation protocol. We benchmark GraphSculptor against 13 static data selection baselines, covering Random sampling, classical coreset methods (k-Center, Herding, CD), and representative label-informed pruning approaches (Influence, DP). .

Implementation Details. For SSL pre-training on ZINC, we employ GraphMAE [Hou *et al.*, 2022] and evaluate via linear probing on MoleculeNet. For supervised learning on OGB, we train GraphGPS [Rampasek *et al.*, 2022] models from scratch on selected coresets. GraphSculptor performs graph-to-text translation and semantic embedding in an offline manner. For molecular graphs, we use BioT5-base [Pei *et al.*, 2023] for graph-to-text translation and SciBERT [Beltagy *et al.*, 2019] as the frozen text encoder. For social-network datasets from TUDataset (COLLAB and IMDB-BINARY), we use a general-purpose LLM (*e.g.*, Gemini) for graph-to-text translation and a BERT as the encoder. The number of clusters K is treated as a hyperparameter. All reported results are the mean and standard deviation over 5 random seeds.

Data Remaining	Method	BBBP	Tox21	ToxCast	SIDER	Clintox	MUV	HIV	BACE	Avg.
	Full Dataset	$71.7_{\pm 0.7}$	$75.6_{\pm 0.6}$	$63.8_{\pm 0.4}$	$60.4_{\pm 1.0}$	$81.0_{\pm 1.9}$	$77.3_{\pm 2.1}$	$76.5_{\pm 1.3}$	$83.3_{\pm 0.8}$	73.7
70% Dataset Remaining	Random	$70.8_{\pm 0.7}$	$75.4_{\pm 0.1}$	$64.1_{\pm 0.5}$	$60.5_{\pm 0.7}$	$81.8_{\pm 1.3}$	$76.2_{\pm 0.9}$	$77.1_{\pm 1.3}$	$82.5_{\pm 0.6}$	73.6 ↓ 0.1
	CD	$71.5_{\pm 0.4}$	$75.2_{\pm 0.4}$	$64.0_{\pm 0.4}$	$60.0_{\pm 0.7}$	$84.2_{\pm 1.0}$	$74.9_{\pm 1.9}$	$76.7_{\pm 0.3}$	$82.6_{\pm 1.3}$	73.6 ↓ 0.1
	K-Center	$72.5_{\pm 0.5}$	$74.6_{\pm 0.6}$	$63.5_{\pm 0.4}$	$60.3_{\pm 0.5}$	$82.9_{\pm 1.7}$	$74.8_{\pm 0.9}$	$77.1_{\pm 1.5}$	$83.0_{\pm 1.2}$	73.6 ↓ 0.1
	Herding	$72.4_{\pm 0.4}$	$75.3_{\pm 0.2}$	$63.6_{\pm 0.3}$	$59.0_{\pm 0.4}$	$83.8_{\pm 1.0}$	$75.1_{\pm 2.2}$	$77.4_{\pm 0.6}$	$82.7_{\pm 0.5}$	73.7 ↓ 0.0
	GraphSculptor	$\mathbf{72.8}_{\pm 0.4}$	$75.3_{\pm 0.5}$	$63.4_{\pm 0.4}$	$\mathbf{62.5}_{\pm \mathbf{0.3}}$	$81.9_{\pm 1.3}$	$75.2_{\pm 1.6}$	$\mathbf{77.9}_{\pm 1.3}$	$\mathbf{83.8}_{\pm 1.0}$	**74.1** ↑ 0.4
50% Dataset Remaining	Random	$71.4_{\pm 0.9}$	$75.1_{\pm 0.5}$	$63.5_{\pm 0.4}$	$60.6_{\pm 0.7}$	$83.4_{\pm 2.5}$	$74.3_{\pm 2.0}$	$77.1_{\pm 1.1}$	$81.7_{\pm 0.5}$	73.4 ↓ 0.3
	CD	$71.8_{\pm 0.3}$	$75.7_{\pm 0.7}$	$63.3_{\pm 0.4}$	$60.0_{\pm 0.7}$	$83.7_{\pm 2.0}$	$73.6_{\pm 1.8}$	$77.2_{\pm 0.7}$	$83.0_{\pm 1.5}$	73.4 ↓ 0.3
	K-Center	$72.4_{\pm 0.3}$	$74.4_{\pm 0.4}$	$63.5_{\pm 0.6}$	$59.8_{\pm 1.1}$	$83.2_{\pm 1.1}$	$76.4_{\pm 1.9}$	$76.2_{\pm 1.3}$	$81.9_{\pm 0.7}$	73.5 ↓ 0.2
	Herding	$71.6_{\pm 1.0}$	$74.5_{\pm 0.4}$	$63.8_{\pm 0.3}$	$60.6_{\pm 0.5}$	$83.6_{\pm 1.4}$	$74.8_{\pm 2.5}$	$77.6_{\pm 0.5}$	$82.1_{\pm 0.8}$	73.6 ↓ 0.1
	GraphSculptor	$72.0_{\pm 0.4}$	$\mathbf{76.5}_{\pm 0.5}$	$\mathbf{64.3}_{\pm 0.5}$	$60.5_{\pm 0.8}$	$82.1_{\pm 1.5}$	$74.4_{\pm 1.8}$	$\mathbf{77.9}_{\pm 1.3}$	$82.8_{\pm 0.8}$	**73.8** ↑ 0.1
10% Dataset Remaining	Random	$68.5_{\pm 0.8}$	$74.2_{\pm 0.7}$	$62.9_{\pm 0.7}$	$59.4_{\pm 1.3}$	$83.9_{\pm 1.4}$	$73.1_{\pm 3.8}$	$76.6_{\pm 1.5}$	$80.5_{\pm 1.2}$	72.4 ↓ 1.3
	CD	$70.9_{\pm 0.9}$	$74.0_{\pm 1.0}$	$63.4_{\pm 0.8}$	$59.0_{\pm 1.6}$	$82.7_{\pm 2.1}$	$73.6_{\pm 2.4}$	$76.7_{\pm 1.7}$	$80.5_{\pm 1.2}$	72.6 ↓ 1.1
	K-Center	$70.6_{\pm 0.9}$	$74.2_{\pm 0.5}$	$63.1_{\pm 0.4}$	$59.7_{\pm 1.1}$	$80.5_{\pm 3.0}$	$73.8_{\pm 2.2}$	$76.0_{\pm 1.0}$	$81.0_{\pm 0.9}$	72.4 ↓ 1.3
	Herding	$70.5_{\pm 0.9}$	$74.0_{\pm 0.6}$	$63.1_{\pm 0.3}$	$57.6_{\pm 1.6}$	$83.5_{\pm 2.7}$	$72.6_{\pm 1.4}$	$76.7_{\pm 1.4}$	$80.0_{\pm 1.1}$	72.3 ↓ 1.4
	GraphSculptor	$\mathbf{71.1}_{\pm 0.5}$	$\mathbf{74.7}_{\pm 0.7}$	$\mathbf{64.0}_{\pm 0.5}$	$\mathbf{60.1}_{\pm 1.1}$	$\mathbf{84.8}_{\pm 1.8}$	$\mathbf{74.0}_{\pm 1.2}$	$\mathbf{77.1}_{\pm 0.6}$	$\mathbf{81.2}_{\pm 0.6}$	**73.4** ↓ 0.3

Table 1: Transfer learning performance on 8 MoleculeNet benchmarks (ROC-AUC %). Results are grouped by sampling ratio. Full Dataset refers to the performance using the original graph dataset. Bold indicates the best result within each sampling ratio.

4.2 Main Result in Graph SSL

We evaluate whether a curated subset can match full-data pre-training for graph SSL. Table 1 reports transfer results on eight MoleculeNet benchmarks (ROC-AUC). GraphSculptor remains competitive across budgets and achieves the highest average ROC-AUC in Table 1 under each reported ratio: 74.1 at 70% remaining (vs. 73.7 for full-data pre-training), 73.8 at 50% remaining, and 73.4 at 10% remaining (vs. 72.3–72.6 for classical coreset baselines such as Herding, K-Center, and CD). Two observations stand out. First, GraphSculptor matches or slightly exceeds full-data pre-training at moderate budgets, suggesting that removing redundant graphs may sharpen the effective pre-training signal. Second, its advantage becomes more pronounced under aggressive compression, consistent with our theory that the loss-gap bound scales with $1/M_{\text{target}}$ and depends on the allocation across clusters, and with our design that uses fused structural–semantic clustering and cluster-aware budget allocation to better cover diverse, non-redundant regions.

4.3 Generality in Supervised Learning

We further evaluate the generality of GraphSculptor on supervised graph classification. On OGBG-MOLHIV and OGBG-MOLPCBA, GraphSculptor performs label-free selection while several strong baselines leverage labels for pruning; nevertheless, GraphSculptor remains competitive and often yields better accuracy under the same budget. For example, on MOLHIV it matches or slightly exceeds the full-data result at the 70% budget, and at 50% it achieves 77.0 on HIV and 25.3 on PCBA, outperforming Random subsets even at larger retention ratios. Beyond molecular graphs, we also test social-network benchmarks from TUDataset (Table 3). On COLLAB, GraphSculptor consistently improves over Random and classical coreset baselines across budgets, reaching 81.0/81.3 at 50%/70% and closely tracking the full-data reference (81.4). On IMDB-BINARY, the gains are more modest, which is expected given the small dataset size and limited redundancy. Nonetheless, GraphSculptor is most effective in the low-budget regime, outperforming Random and achieving the best (or tied-best) accuracy at 20%–30% retention, while at higher budgets it becomes comparable to classical coreset methods such as K-Center and Herding.

4.4 Further Discussion

Ablation Study. We study the contribution of the two diversity views in GraphSculptor by comparing the full method with two ablations: Structure-only, which performs clustering and selection using only structural descriptors $\mathbf{z}_S$, and Semantics-only, which uses only semantic embeddings $\mathbf{z}_T$. As shown in Figure 4, the full model achieves the best performance across all budgets, while removing either view leads to a consistent degradation, indicating that the two views provide complementary signals for coreset curation. In our setting, dropping the semantic view typically incurs a larger performance drop than dropping the structural view, suggesting that language-based semantics can capture information not well reflected by structural statistics alone. Overall, combining both structural and semantic cues yields a more reliable criterion than either single view.

Efficiency Analysis. Figure 5 summarizes the end-to-end efficiency of GraphSculptor. The method incurs a one-time preprocessing cost for graph-to-text and text encoding (about

Method	OGBG-MOLHIV (ROC-AUC: 78.7)				OGBG-MOLPCBA (AP: 27.2)			
	20%	30%	50%	70%	20%	30%	50%	70%
Random	69.3 ↓ 9.4	72.7 ↓ 6.0	73.4 ↓ 5.3	75.6 ↓ 3.1	19.4 ↓ 7.8	21.7 ↓ 5.5	23.9 ↓ 3.3	26.3 ↓ 0.9
CD	72.6 ↓ 6.1	73.0 ↓ 5.7	75.3 ↓ 3.4	76.7 ↓ 2.0	18.0 ↓ 9.2	20.7 ↓ 6.5	21.7 ↓ 5.5	26.4 ↓ 1.0
Herding	69.5 ↓ 9.2	73.3 ↓ 5.4	74.5 ↓ 4.2	75.9 ↓ 2.8	13.3 ↓ 13.9	14.0 ↓ 13.2	17.8 ↓ 9.4	23.0 ↓ 4.2
K-Center	67.2 ↓ 11.5	70.8 ↓ 7.9	72.6 ↓ 6.1	73.9 ↓ 4.8	16.9 ↓ 10.3	19.4 ↓ 7.8	22.8 ↓ 4.4	26.1 ↓ 1.1
SVP	73.9 ↓ 4.8	74.2 ↓ 4.5	75.8 ↓ 2.9	77.3 ↓ 1.4	19.4 ↓ 7.6	21.9 ↓ 5.3	23.5 ↓ 3.7	26.0 ↓ 1.1
Margin	74.0 ↓ 4.7	74.4 ↓ 4.3	75.8 ↓ 2.9	77.5 ↓ 1.2	18.8 ↓ 8.1	21.5 ↓ 5.7	23.9 ↓ 3.3	27.0 ↓ 0.2
Forgetting	74.2 ↓ 4.5	74.8 ↓ 3.9	75.9 ↓ 3.1	76.9 ↓ 1.6	19.4 ↓ 7.5	21.9 ↓ 5.3	23.3 ↓ 3.9	26.8 ↓ 0.4
GraNd-4	73.8 ↓ 4.9	74.2 ↓ 4.5	75.8 ↓ 3.4	77.5 ↓ 1.2	18.0 ↓ 9.2	21.2 ↓ 6.3	23.6 ↓ 3.6	26.9 ↓ 0.3
DeepFool	72.2 ↓ 6.5	73.3 ↓ 5.4	74.9 ↓ 3.4	75.5 ↓ 3.2	17.6 ↓ 9.6	21.3 ↓ 6.2	23.2 ↓ 4.0	26.5 ↓ 1.0
CRAIG	73.5 ↓ 5.0	74.4 ↓ 4.3	75.9 ↓ 2.7	77.5 ↓ 1.2	**22.5** ↓ 4.5	24.5 ↓ 2.5	24.7 ↓ 2.2	27.1 ↓ 1.0
GLSTER	73.6 ↓ 4.9	75.0 ↓ 3.7	75.8 ↓ 2.8	77.0 ↓ 0.7	**22.5** ↓ 4.5	**24.8** ↓ 2.2	24.2 ↓ 2.7	27.0 ↓ 0.2
Influence	72.9 ↓ 5.6	73.7 ↓ 5.0	74.8 ↓ 3.9	77.4 ↓ 1.3	17.7 ↓ 9.3	23.5 ↓ 3.7	23.5 ↓ 3.7	26.6 ↓ 0.6
EL2N-20	74.0 ↓ 4.7	75.5 ↓ 3.2	76.9 ↓ 1.9	77.7 ↓ 1.1	19.1 ↓ 8.1	22.9 ↓ 3.9	25.0 ↓ 2.0	26.6 ↓ 1.1
DP	72.0 ↓ 6.7	74.1 ↓ 4.6	76.0 ↓ 2.7	76.9 ↓ 1.8	19.6 ↓ 7.6	21.5 ↓ 5.7	24.9 ↓ 2.3	26.4 ↓ 0.8
GraphSculptor	**74.6** ↓ 4.1	**75.6** ↓ 3.1	**77.0** ↓ 1.7	**78.9** ↑ 0.2	19.5 ↓ 7.7	22.4 ↓ 4.8	**25.3** ↓ 1.9	**27.4** ↑ 0.2

Table 2: Supervised Learning results on OGBG-HIV and OGBG-PCBA.

Method	COLLAB (81.4)				IMDB-BINARY (77.4)			
	20%	30%	50%	70%	20%	30%	50%	70%
Random	75.1	76.2	80.4	80.9	73.8	74.2	74.6	75.4
CD	74.5	76.4	80.5	80.7	73.5	74.2	75.0	75.5
Herding	75.6	76.5	80.7	80.2	74.0	74.5	75.2	76.0
K-Center	75.4	75.8	78.6	79.8	74.2	74.7	**75.5**	**76.1**
GraphSculptor	**76.0**	**76.6**	**81.0**	**81.3**	**74.2**	**75.0**	75.3	75.8

Table 3: Supervised Learning results on COLLAB and IMDB-BINARY.

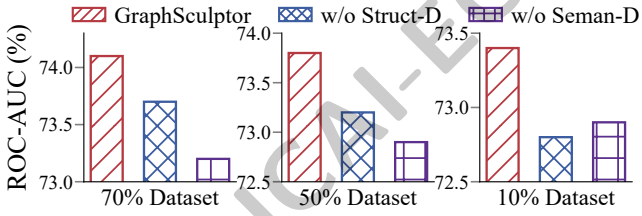

Figure 4: **Ablation study** of GraphSculptor. "w/o Struct-D" denotes the absence of structural diversity; "w/o Seman-D" denotes the lack of semantic diversity.

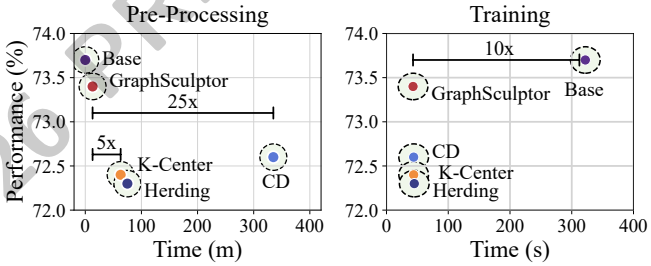

Figure 5: **Efficiency analysis**.

30 minutes in our setup), but this cost is amortized across pre-training runs and is substantially lower than several training-signal-based pruning baselines (5–25× faster in wall-clock time). More importantly, coreset pre-training provides a direct reduction in training time proportional to the retained data budget: for example, using a 10% coreset cuts pre-training time by nearly 90% compared to full-data pre-training, while maintaining comparable downstream performance (and occasionally yielding small gains). Overall, GraphSculptor offers a favorable performance–efficiency trade-off.

5 Conclusion

We revisit the prevailing practice of scaling unlabeled graph corpora for self-supervised pre-training and observe notable redundancy in our benchmark evaluation. We propose GraphSculptor, a label-free and model-agnostic framework for constructing compact pre-training coresets by combining a structural view based on intrinsic graph statistics with a complementary semantic view derived from graph-to-text and a frozen text encoder. We further provide a loss-gap bound under a standard linearized optimization regime, offering theoretical motivation for cluster-aware budget allocation. Empirically, GraphSculptor achieves favorable performance–efficiency trade-offs across molecular and social graph benchmarks. Overall, our results suggest that diversity-driven, offline data curation can serve as a practical preprocessing step for more efficient scaling of graph SSL. As future work, we will explore more reliable semantic abstractions for text-scarce graphs and strengthen the theoretical analysis in training dynamics.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China (No. 62502065, 62476203), the National Key Research and Development Program of China (2023YFC2705705), the Key Project of Traditional Chinese Medicine Joint Fund of Hubei Provincial Natural Science Foundation (No. 2025AFD47), the Hubei Province Science and Technology Innovation Plan Project (No. 2025BCB035), the Guangdong Provincial Natural Science Foundation General Project (No. 2025A1515012155), the Shenzhen Natural Science Foundation Project (No. JCYJ20250604122534006, JCYJ20230807090211021), the Hubei Jiangxia Laboratory Development Project (No. 2025JXKF034), the New Cornerstone Science Foundation through the XPLORER PRIZE, and the Innovative Research Group Project of Hubei Province under Grants 2024AFA017.

References

[Beltagy *et al.*, 2019] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[Chen *et al.*, 2024] Dingshuo Chen, Zhixun Li, Yuyan Ni, Guibin Zhang, Ding Wang, Qiang Liu, Shu Wu, Jeffrey Xu Yu, and Liang Wang. Beyond efficiency: Molecular data pruning for enhanced generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Assoc. Comput. Linguistics*, 2019.

[Gupta *et al.*, 2024] Mridul Gupta, Sahil Manchanda, HARIPRASAD KODAMANA, and Sayan Ranu. Mirage: Model-agnostic graph distillation for graph classification. In *The Twelfth International Conference on Learning Representations*, 2024.

[Hou *et al.*, 2022] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2022.

[Hou *et al.*, 2023] Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proc. ACM Web Conf. 2023*, 2023.

[Hu *et al.*, 2020] 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. *arXiv:2005.00687*, 2020.

[Jin *et al.*, 2022a] Wei Jin, Xianfeng Tang, Haoming Jiang, Zheng Li, Danqing Zhang, Jiliang Tang, and Bing Yin. Condensing graphs via one-step gradient matching. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[Jin *et al.*, 2022b] Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, and Neil Shah. Graph condensation for graph neural networks. In *International Conference on Learning Representations*, 2022.

[Li *et al.*, 2023] Jintang Li, Ruofan Wu, Wangbin Sun, Liang Chen, Sheng Tian, Liang Zhu, Changhua Meng, Zibin Zheng, and Weiqiang Wang. What's behind the mask: Understanding masked graph modeling for graph autoencoders. In *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2023.

[Li *et al.*, 2025] Ting-Wei Li, Ruizhong Qiu, and Hanghang Tong. Model-free graph data selection under distribution shift, 2025.

[Liu *et al.*, 2023] Zhiyuan Liu, Yaorui Shi, An Zhang, Enzhi Zhang, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. Rethinking tokenizer and decoder in masked graph modeling for molecules. In *Thirty-seventh Conf. Neural Inf. Process. Syst.*, 2023.

[Liu *et al.*, 2024] Chuang Liu, Yuyao Wang, Yibing Zhan, Xueqi Ma, Dapeng Tao, Jia Wu, and Wenbin Hu. Where to mask: Structure-guided masking for graph masked autoencoders. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 2180–2188. International Joint Conferences on Artificial Intelligence Organization, 2024.

[Liu *et al.*, 2025] Yang Liu, Deyu Bo, Wenxuan Cao, Yuan Fang, Yawen Li, and Chuan Shi. Graph positional autoencoders as self-supervised learners. In *SIGKDD*, 2025.

[Liu *et al.*, 2026] Chuang Liu, Zelin Yao, Xueqi Ma, Mukun Chen, Luzhi Wang, Jia Wu, and Wenbin Hu. Hi-gmae: Hierarchical graph masked autoencoders. In *Proceedings of the ACM Web Conference 2026*, page 1250–1261, 2026.

[Morris *et al.*, 2020] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop Graph Representation Learn. Beyond (GRL+ 2020)*, 2020.

[Pei *et al.*, 2023] Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1102–1123, December 2023.

[Rampasek *et al.*, 2022] Ladislav Rampasek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Conf. Workshop Neural Inf. Process. Syst.*, 2022.

[Shen *et al.*, 2025] Weixuan Shen, Xiaobo Shen, and Shirui Pan. Adversarial contrastive graph masked autoencoder against graph structure and feature dual attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[Shi *et al.*, 2023] Yucheng Shi, Yushun Dong, Qiaoyu Tan, Jundong Li, and Ninghao Liu. Gigamae: Generalizable graph masked autoencoder via collaborative latent space reconstruction. In *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, 2023.

[Sterling and Irwin, 2015] Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *J. Chem. Inf. Model.*, 2015.

[Tan *et al.*, 2023] Qiaoyu Tan, Ninghao Liu, Xiao Huang, Soo-Hyun Choi, Li Li, Rui Chen, and Xia Hu. S2gae: Self-supervised graph autoencoders are generalizable learners with graph masking. In *Proc. Sixteenth ACM Int. Conf. Web Search Data Mining*, 2023.

[Wang *et al.*, 2024] Liang Wang, Xiang Tao, Qiang Liu, and Shu Wu. Rethinking graph masked autoencoders through alignment and uniformity. In *Proc. AAAI Conf. Artif. Intell.*, 2024.

[Wu *et al.*, 2018] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.*, 2018.

[Xia *et al.*, 2022a] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proc. ACM Web Conf. 2022*, 2022.

[Xia *et al.*, 2022b] Jun Xia, Yanqiao Zhu, Yuanqi Du, and Stan Z Li. A survey of pretraining on graphs: Taxonomy, methods, and applications. *arXiv:2202.07893*, 2022.

[Xu *et al.*, 2023a] 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. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Xu *et al.*, 2023b] Zhe Xu, Yuzhong Chen, Menghai Pan, Huiyuan Chen, Mahashweta Das, Hao Yang, and Hanghang Tong. Kernel ridge regression-based graph dataset distillation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 2850–2861, 2023.

[Yang *et al.*, 2023] Beining Yang, Kai Wang, Qingyun Sun, Cheng Ji, Xingcheng Fu, Hao Tang, Yang You, and Jianxin Li. Does graph distillation see like vision dataset counterpart? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Ying *et al.*, 2021] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Conf. Workshop Neural Inf. Process. Syst.*, 2021.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances Neural Inf. Process. Syst.*, volume 33, pages 5812–5823, 2020.

[Zhang *et al.*, 2024] Guibin Zhang, Haonan Dong, Yuchen Zhang, Zhixun Li, Dingshuo Chen, Kai Wang, Tianlong Chen, Yuxuan Liang, Dawei Cheng, and Kun Wang. GDer: Safeguarding efficiency, balancing, and robustness via prototypical graph pruning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[Zheng and Jia, 2024] Yimei Zheng and Caiyan Jia. Protomgae: Prototype-aware masked graph auto-encoder for graph representation learning. *ACM Trans. Knowl. Discovery Data*, 2024.