On Representation Knowledge Distillation for Graph Neural Networks

Chaitanya K. Joshi, Fayao Liu, Xu Xun, Jie Lin, Chuan Sheng Foo

***Abstract*—Knowledge distillation is a learning paradigm for boosting resource-efficient graph neural networks (GNNs) using more expressive yet cumbersome teacher models. Past work on distillation for GNNs proposed the Local Structure Preserving loss (LSP), which matches *local* structural relationships defined over edges across the student and teacher's node embeddings. This paper studies whether preserving the *global* topology of how the teacher embeds graph data can be a more effective distillation objective for GNNs, as real-world graphs often contain latent interactions and noisy edges. We propose Graph Contrastive Representation Distillation (G-CRD), which uses contrastive learning to *implicitly* preserve global topology by aligning the student node embeddings to those of the teacher in a shared representation space. Additionally, we introduce an expanded set of benchmarks on large-scale real-world datasets where the performance gap between teacher and student GNNs is non-negligible. Experiments across 4 datasets and 14 heterogeneous GNN architectures show that G-CRD consistently boosts the performance and robustness of lightweight GNNs, outperforming LSP (and a global structure preserving variant of LSP) as well as baselines from 2D computer vision. An analysis of the representational similarity among teacher and student embedding spaces reveals that G-CRD balances preserving local and global relationships, while structure preserving approaches are best at preserving one or the other.**

***Index Terms*—Knowledge Distillation, Graph Neural Networks, Geometric Deep Learning, Contrastive Learning.**

I. Introduction

GRAPH Neural Networks (GNNs) [1]–[5] generalize convolutional networks from 2D computer vision to irregular data structures such as graphs, sets, and 3D point clouds. Recent years have seen impactful applications of GNNs in fields ranging from social networks [6], [7] to biomedical discovery [8]–[10]. While the community has recently focused on large-scale datasets [11]–[13], more expressive architectures [14]–[18] as well as improving generalization via self-supervised learning [19]–[22], there has been an emerging line of work on lightweight, resource-efficient GNNs [23]–[30] that achieve high performance under computation and memory constraints. Boosting the performance and reliability of such lightweight models is critical for accelerating a myriad of applications, including real-time recommendations on social networks, safety-critical 3D perception for autonomous robots, and encrypted models for proprietary biomedical data.

A promising and generic approach for improving lightweight deep learning models is Knowledge Distillation (KD) [31]. KD is a teacher-student learning paradigm that transfers knowledge from high performance but resource-intensive teacher models to resource-efficient students. Pioneered by Hinton et al. [31], *logit*-based distillation trains the student to match the output logits of teachers, in addition to standard supervised learning. Recent work has attempted to go beyond logit-based distillation by transferring *representational* knowledge from the teacher to the student through the design of loss functions that align the latent embedding spaces of the teacher and student [32], see Fig.1(a) for an intuitive overview.

Unlike for 2D image data, representation distillation for GNNs has largely been unexplored. To the best of our knowledge, the only representation distillation method for GNNs is the pioneering work of Yang et al. [33] that proposes a **Local Structure Preserving** (LSP) loss to encourage the student to mimic the pairwise similarities with immediate neighbours present in the teacher's node embedding space. Thus, structural similarities are preserved over *local* graph edges. However, this is not guaranteed to preserve the *global* topology of how the teacher embeds graphs as this ignores latent interactions among disconnected nodes, see Fig.1(b).

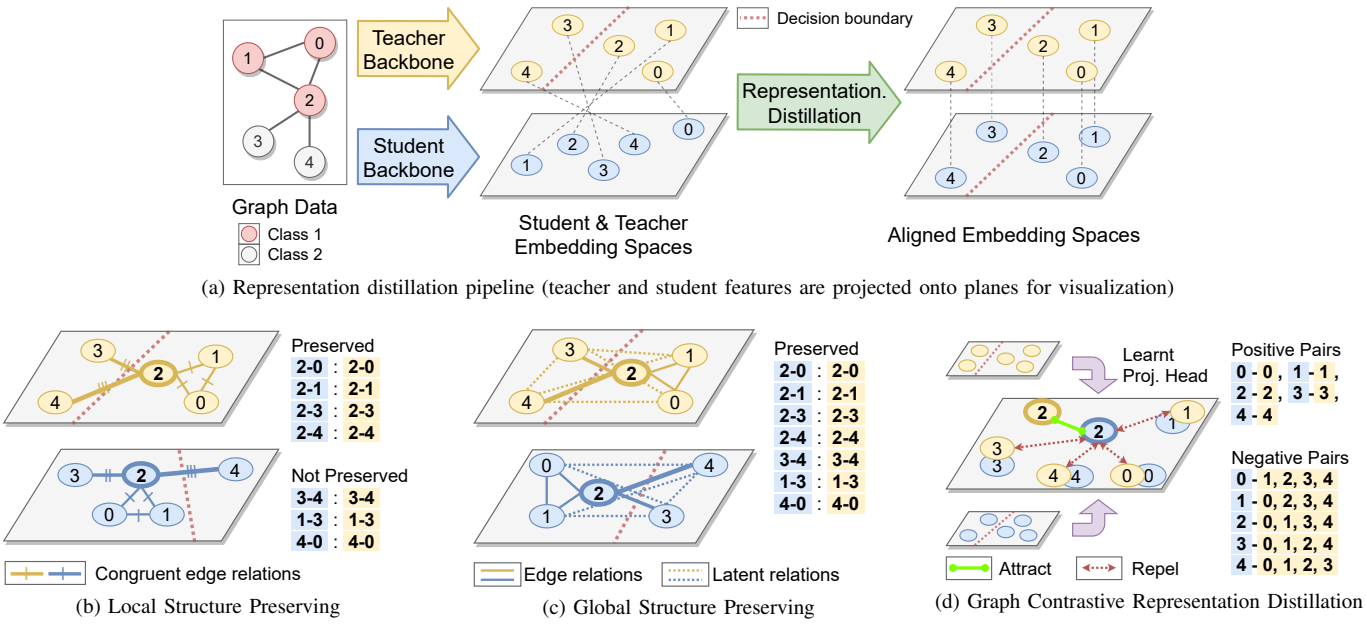
Modelling latent interactions is often critical for solving tasks on real-world graphs, which tend to be incomplete or noisy as edges are determined heuristically [34]. For *e.g.*, the 3D interactions of diametrically opposite atoms in molecular graphs may determine overall properties, and edges in social or biological networks correspond to only reported or experimental observations. By using expressive and deep GNNs as teachers, the topology of the teacher node embedding space is able to better capture the full structure of the underlying data, including latent interactions. Learning the same information via lightweight or shallow student GNNs may not be possible due to be mismatch in representational capacity. Thus, the ideal representation distillation technique would transfer *global* structural information from the teacher to the student.

To preserve global relationships, it is natural to extend LSP to *explicitly* consider all possible pairwise similarities among node features. However, we found this **Global Structure Preserving** (GSP) approach to be challenging to scale as the number of possible relationships may explode, see Fig.1(c).

Thus, we introduce a new distillation objective that *implicitly* preserves global topology by aligning the student and teacher node feature vectors via contrastive learning [35]–[37]. We formulate our objective as a node-level contrastive task on pairwise relationships *across* the teacher and student embedding spaces, see Fig.1(d). We term this objective **Graph Contrastive Representation Distillation (G-CRD)**, as it generalizes CRD [38] from sample-level 2D image classification tasks to fine-

Submitted to IEEE TNNLS Special Issue on Deep Neural Networks for Graphs: Theory, Models, Algorithms and Applications.

Work done when all authors were at Institute for Infocomm Research, A*STAR, Singapore. CKJ is now at University of Cambridge, UK. CSF is also affiliated with Centre for Frontier AI Research, A*STAR



(a) Representation distillation pipeline (teacher and student features are projected onto planes for visualization)

(b) Local Structure Preserving

(c) Global Structure Preserving

(d) Graph Contrastive Representation Distillation

Fig. 1: **Graphical overview of representation distillation for GNNs.** (a) Representational knowledge is transferred by aligning the latent node embedding space of the student to that of the teacher. (b) **Local Structure Preserving** loss [33] considers pairwise relationships over graph edges, but may not preserve global topology of the teacher's embedding space due to latent interactions among disconnected nodes. (c) Preserving all pairwise relationships, *i.e.* **Global Structure Preserving** loss, better preserves the global topology of the embedding space, but can be challenging to optimize/scale up due to an explosion of possible pairs. (d) Different from relation-based approaches, **Graph Contrastive Representation Distillation** transfers representational knowledge by contrastive learning among positive/negative pairwise relations *across* the teacher and student embedding spaces via learnt projection heads.

grained node-level tasks on graphs. G-CRD preserves global relationships by training the student to spatially align its node embeddings with the corresponding teacher node embedding, termed positive samples. For example, in Fig.1(d), the student's embedding for node 2 is pushed to the teacher's embedding for node 2 by maximizing the learnt similarity metric. Additionally, the student's embedding for node 2 is repelled from all other node embeddings from the teacher, termed negative samples, by minimizing their similarities. The number of negative samples can be varied to ensure scalability.

We compare G-CRD to LSP, its global variant GSP, and baseline techniques from 2D computer vision across a diverse range of model architectures and tasks from Open Graph Benchmarks [11] and S3DIS [39]. Crucially, our evaluation focuses on out-of-distribution generalization on real-world data, where the performance gap between cumbersome teachers and lightweight students is non-negligible. We believe this is critical for testing the efficacy and robustness of knowledge distillation, but was missing from the LSP study [33] which used small-scale and synthetic datasets with negligible performance gaps between teachers and students. In addition to our technical contributions, we hope that our expanded set of benchmarks can form the basis for comparison in future work.

Our contributions are summarized as follows:

- We study GNN representation distillation from the perspective of preserving global topology. We introduce Graph Contrastive Representation Distillation (G-CRD), the first contrastive distillation technique specialized for GNNs which trains students to implicitly preserve the global topology of the teacher's node embedding space.
- We benchmark GNN distillation on large-scale datasets which test out-of-distribution generalization. Our experiments compare 6 distillation techniques across 4 tasks and 14 architectures for teachers and students. This extends the LSP study which used synthetic datasets with negligible performance gap between teachers and students.
- Training lightweight GNNs with G-CRD consistently outperform the structure preserving objectives, LSP and GSP, as well as baselines adapted from 2D computer vision. We further analyze the robustness, transferability, quantizability, and representational similarity of teacher and student embeddings in order to unpack the efficacy of G-CRD over the structure preserving approaches.

II. Related Work

Graph Neural Networks (GNNs). GNNs [1]–[5] generalize convolutional networks from 2D computer vision to graph structured data. GNNs serve as powerful feature extractors for real-world and real-time applications across diverse domains, including social networks [6], [7], biomedicine [8]–[10], 3D perception [3], [15], natural language processing [40], [41], and operations research [42], [43]. Our study complements an emerging line of work on efficient GNNs, including lightweight models [23]–[25], neural architecture search [26], [27] and quantization techniques [28], [29].

Knowledge Distillation (KD). KD [31] is a learning paradigm for improving the performance of lightweight 'student' models by matching their outputs to those from cumbersome 'teacher' models. Recent literature has focused on augmenting Hinton et al.'s [31] logit-based approach with

representational knowledge from the teacher's latent feature vectors. Following the taxonomy in [32], *feature-based* distillation techniques train the student to imitate the teacher's intermediate feature vectors or attention maps via direct regression [44]–[47]. Due to mismatches in representational capacity between the teacher and student, feature-based distillation may not improve the performance of lightweight students. As an alternative, *relation-based* distillation preserves metrics among feature vectors from the teacher's representation space to that of the student, *e.g.* semantic similarity graphs, pairwise distances or angular relationships among samples from the dataset [48]–[50]. The Local Structure Preserving loss (LSP) [33] and its global variant (GSP) extend relation-based distillation for graph data.

Motivated by the success of contrastive objectives for self-supervised learning [35]–[37], Tian et al. [38] proposed Contrastive Representation Distillation (CRD) for 2D image classification. In this work, we found that the sample-level CRD objective is not effective for distilling node-level representations built by GNNs. We introduce Graph Contrastive Representation Distillation (G-CRD), a node-level contrastive distillation objective tailored to GNNs.

Knowledge Distillation and GNNs. Local Structure Preserving loss (LSP) [33] is the only distillation technique specifically designed for GNNs and message passing style models, but was evaluated on small-scale synthetic datasets. This work extends the LSP study in two ways: (1) We introduce G-CRD, a new representation distillation objectives that preserve global topology and consistently outperform LSP; and (2) We benchmark GNN distillation on large-scale datasets which evaluate for out-of-distribution generalization, as well as across a wider range of teacher-student combinations.

The original logit-based KD [31] has been applied to GNNs to improve the training of lightweight models such as graph-agnostic MLPs or label propagation [51]–[53]. These works do not introduce new GNN-specific distillation techniques. Data-free or teacher-free self-distillation [**?**], [54] has also been used to regularize the training of GNNs in the semi-supervised and self-supervised learning paradigm.

III. Preliminaries

A. Graph Representation Learning

Graph Neural Networks (GNNs) take as input an unordered set of nodes and the graph connectivity among them, and learn latent node representations for them via iterative feature aggregation or message passing across local neighborhoods [5]. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of n nodes, and $\mathcal{E}$ is a set of edges associated with the nodes. Each node $i \in \mathcal{V}$ is associated with a d-dimensional initial feature vector f_i, which is iteratively updated by the backbone GNN via a generic message passing or graph convolution operation:

$$f_i^{\ell+1} = \textsc{Upd}\left(f_i^{\ell}, \underset{(i,j)\in\mathcal{E}}{\textsc{Agg}}\Big(\textsc{Msg}\big(f_i^{\ell}, f_j^{\ell}, e_{ij}\big)\Big)\right), \tag{1}$$

where $\textsc{Msg}, \textsc{Upd}$ are learnable transformations such as multi-layer perceptrons, e_{ij} are optional edge features for each pair of linked nodes i, j, and $\textsc{Agg}$ is a permutation-invariant local neighborhood aggregation function such as summation, maximization, averaging, or weighted averaging via attention.

We are usually interested in making predictions for each node in the graph, *e.g.* node classification on social networks or semantic segmentation of 3D point clouds. Thus, the final feature vectors after L layers of message passing f_i^L are passed to a linear classifier to obtain logits z_i, and the neural network is trained end-to-end via a cross-entropy loss $\mathcal{H}$ with the groundtruth class labels y_i: $\mathcal{L}_{\textsc{Sup}} = \mathcal{H}\left(y_i, \text{softmax}\left(z_i\right)\right)$.

For graph-level tasks, such as molecular property prediction, the feature vectors of all nodes are pooled via a permutation-invariant function $\textsc{Pool}$ to obtain a global feature vector $f_G = \textsc{Pool}\left(f_i^L | i \in \mathcal{V}\right)$, and then passed to a linear classifier. Notably, representation learning still occurs at the node-level for graph-level prediction tasks [55], unlike 2D image classification where pooling layers learn feature vectors for entire images.

GNNs and 3D point clouds. The message passing framework in (1) can be used to present a unified 'geometric' view of deep learning on non-Euclidean data structures such as 3D point clouds, irregular voxel grids, meshes, and sets [40], [56]. In this work, we consider 3D point cloud networks [57], [58] which process sets of points as nodes. The edges originating from each point are heuristically determined by the k-nearest neighbors or radius ball queries in the 3D coordinate space. We additionally consider sparse 3D voxel networks [59]–[61] and use trilinear interpolation to convert voxel-level features to point-level features before distillation.

B. Knowledge Distillation

Knowledge Distillation (KD) transfers *dark knowledge* from high capacity teacher models (which are cumbersome to deploy) to efficient students via matching their output logits in addition to supervised learning. At each node $i \in \mathcal{V}$, logit-based KD [31] uses the cross-entropy loss or KL-divergence to match the output of a softmax of the logits of the student z_i^S and teacher z_i^T, scaled by temperature τ_1:

$$\mathcal{L}_{\text{KD}} = \sum_{i\in\mathcal{V}} \mathcal{H}\left(\text{softmax}\left(z_i^T/\tau_1\right), \text{softmax}\left(z_i^S/\tau_1\right)\right). \tag{2}$$

In this work, we study auxiliary representation distillation techniques that augment or replace logit-based distillation with *representational knowledge* using the teacher and student features, $f^T \in \mathbb{R}^{d^T}$ and $f^S \in \mathbb{R}^{d^S}$, respectively.[1] Our overall objective function for training the student model via the logits as well as latent feature vectors of the teacher is as follows:

$$\mathcal{L} = (1-\alpha)\,\mathcal{L}_{\textsc{Sup}} \;+\; \alpha\tau_1^2\;\mathcal{L}_{\text{KD}} \;+\; \beta\;\mathcal{L}_{\text{Aux}}, \tag{3}$$

where α, β are balancing weights for the logit-based KD loss $\mathcal{L}_{\text{KD}}$ and auxiliary representation distillation loss $\mathcal{L}_{\text{Aux}}$, respectively. Next, we will describe how $\mathcal{L}_{\text{Aux}}$ is instantiated as different representation distillation techniques such as $\mathcal{L}_{\text{LSP}}$, $\mathcal{L}_{\text{GSP}}$, and $\mathcal{L}_{\text{G-CRD}}$.

[1]Following best practices [38], [47], we consider feature vectors from the penultimate layer before the prediction head for representation distillation.

C. Local Structure Preserving Distillation

We briefly summarize LSP [33], a GNN representation distillation objective which trains the student model to preserve the local structure of graph data from the teacher's node embedding space. The local structure around each node is defined as the set of parameterized pairwise distances to its neighboring nodes in the latent feature space. The similarity between a pair of linked nodes can be computed by kernel functions (which we tune for):

$$\mathcal{K}\left(f_i, f_j\right) = \begin{cases} \|f_i - f_j\|_2^2, & \text{Euclidean distance,} \\ f_i' \cdot f_j, & \text{Linear kernel,} \\ \left(f_i' \cdot f_j + c\right)^d, & \text{Polynomial kernel,} \\ e^{-\frac{1}{2\sigma}\|f_i - f_j\|^2}, & \text{RBF kernel.} \end{cases} \tag{4}$$

Thus, the local structure around each node $i \in \mathcal{V}$ is the softmax probability distribution of the similarities among f_i and its neighbors' features $f_j \; \forall (i,j) \in \mathcal{E}$. LSP trains the student model to mimic the local structure from the teacher's embedding space via KL-divergence:

$$\mathcal{L}_{\text{LSP}} = \sum_{i \in \mathcal{V}} \mathcal{D}_{\text{KL}}\Big(\underset{(i,j)\in\mathcal{E}}{\text{softmax}}\big(\mathcal{K}(f_i^S, f_j^S)\big) \,\|\, \underset{(i,j)\in\mathcal{E}}{\text{softmax}}\big(\mathcal{K}(f_i^T, f_j^T)\big)\Big). \tag{5}$$

D. Global Structure Preserving Distillation

The purely local LSP objective over pre-defined edges does not preserve the global topology of how the teacher embeds the graph, as it does not account for latent interactions among disconnected nodes. We would like to *explicitly* train the student to preserve the global topology in order to better distill representational knowledge from the teacher. To achieve this, we first introduce a simple extension of LSP, the Global Structure Preserving loss (GSP), which matches *all* pairwise similarities among node features via mean squared error:

$$\mathcal{L}_{\text{GSP}} = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \left\| \mathcal{K}\left(f_i^T, f_j^T\right) - \mathcal{K}\left(f_i^S, f_j^S\right) \right\|_2^2. \tag{6}$$

While theoretically more powerful than LSP, GSP may be computationally inefficient and involves two key design choices: (See the Appendix for ablation studies.)

MSE over KL-divergence We omit normalizing the similarities via softmax followed by KL-divergence, as this led to very sparse signals over all pairwise similarities when converted to probabilities. We use MSE to match the raw teacher and student pairwise similarity matrices as it worked better empirically.

Scalability and sub-sampling. For a set of n nodes, GSP needs to compute n^2 pairwise similarities for both the teacher and student features. We often need to sub-sample large graphs or 3D point clouds when computing $\mathcal{L}_{\text{GSP}}$ due to GPU memory constraints instead of storing all possible pairwise similarities. For each experiment, random sub-sampling was done to retain as many nodes as possible subject to GPU memory limitations.

IV. Graph Contrastive Representation Distillation

An alternative to *explicit* structure preserving techniques such as LSP and GSP is to directly align the node features from the student to those of the teacher. An objective on the features would be a stronger constraint than over pairwise relationships, resulting in *implicitly* preserving relationships over pre-defined as well as latent edges, and thereby preserving global topology. However, we and others [33] found direct feature mimicking approaches [44], [45] from 2D computer vision to be ineffective for GNNs due to mismatch in representational capacity.

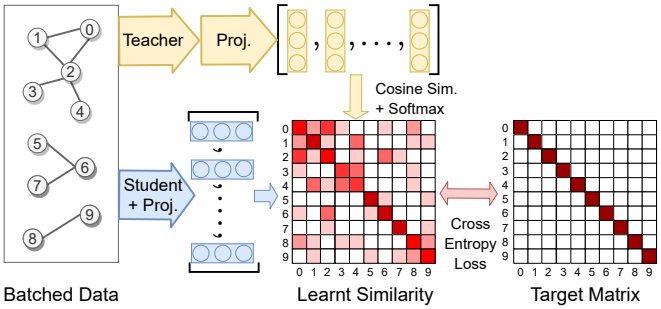


Fig. 2: **Graph Contrastive Representation Distillation.** G-CRD is a node-level contrastive learning task of identifying positive node correspondences across the teacher and student embedding spaces, while pushing away a set of distractor nodes. Here, we define negative samples for each student node feature vector as all the other node features vectors from the teacher within the same mini-batch.

Thus, we introduce Graph Contrastive Representation Distillation (G-CRD) which formulates representation distillation as a contrastive learning task on pairwise relationships *across* the teacher and student embedding spaces. Intuitively, we want to maximize the similarity among pairs of student and teacher feature vectors corresponding to the same node, *i.e.* f_i^S, f_i^T (positive samples), while pushing away the feature vectors of pairs of unmatched nodes, *i.e.* f_i^S, f_j^T (negative samples). To achieve this, we adapt the InfoNCE objective [35], [37] to the teacher-student paradigm and pose representation distillation as the task of classifying positive pairs among the set of distractors via a temperature-scaled softmax function:

$$\mathcal{L}_{\text{G-CRD}} = -\sum_{i \in \mathcal{V}} \log \frac{\exp\left(\frac{h(f_i^S, f_i^T)}{\tau_2}\right)}{\exp\left(\frac{h(f_i^S, f_i^T)}{\tau_2}\right) + \sum_{j \neq i} \exp\left(\frac{h(f_i^S, f_j^T)}{\tau_2}\right)}, \tag{7}$$

where $h : \{d^S, d^T\} \to [0, 1]$ is a learnt similarity metric, and τ_2 is a scalar temperature parameter. From an information theoretic perspective, the InfoNCE-style loss aligns the teacher and student embedding spaces by maximizing the mutual information among them. Fig.2 illustrates G-CRD.

G-CRD generalizes Contrastive Representation Distillation [38] from sample-level 2D image classification tasks to fine-grained node-level tasks on graphs, and involves key design choices tailored for GNNs:

Projection heads. We define the similarity metric as a learnt cosine similarity between teacher and student features projected to a common representation space:

$$h(f^S, f^T) = \frac{P^S\left(f^S\right)' \cdot P^T\left(f^T\right)}{\|P^S\left(f^S\right)\| \cdot \|P^T\left(f^T\right)\|}, \tag{8}$$

where the projection heads P^S, P^T can be multi-layer perceptrons (MLPs) composed of linear transformations to a common dimension d (usually that of the student feature) followed by batch normalization and non-linear activation. However,

independent MLPs on node feature vectors are 'structure-agnostic' [23] – they cannot adapt the shared representation space to the graph structure of the underlying data. Thus, we introduce structure-aware projection heads, which use a single GCN layer [1][2] followed by batch normalization and non-linear activation:

$$P\left(f_i\right) = \begin{cases} \text{GCN}\left(f_i, f_j | (i,j) \in \mathcal{E}\right), & \text{Structure-aware,} \\ \text{MLP}\left(f_i\right), & \text{Structure-agnostic,} \end{cases} \tag{9}$$

The GCN projection can learn neighborhood-dependent projections via one message passing step, while having the same number of parameters as the structure-agnostic MLP. We tune the choice of projection head as a hyperparameter.

Node-level contrastive learning. We formulate G-CRD as a node-level contrastive learning task of identifying positive node correspondences across the teacher and student embedding spaces, while pushing away a set of distractors. G-CRD adapts the well-studied InfoNCE loss in (7). Contrastive Representation Distillation (CRD) [38] is a related representation distillation objective for 2D image classification models:

$$\begin{aligned} \mathcal{L}_{\text{CRD}} = -\max_{h} \Big(& \mathbb{E}_{q(T,S|C=1)}[\log h(T,S)] + \\ & N \mathbb{E}_{q(T,S|C=0)}[1 - \log(h(T,S))] \Big), \end{aligned} \tag{10}$$

where T, S are random variables for teacher and student features, $q(T, S|C = 1)$ is their joint distribution (positives), $q(T, S|C = 0)$ is their marginal distribution (negatives), and $h : \{d^S, d^T\} \rightarrow [0, 1]$ is an auxiliary 'critic' model which returns an unnormalized similarity score. Crucially, CRD is tailored for 2D image classification and contrasts among global/per-sample feature vectors. It additionally requires a specialized memory bank [36] to provide a large number N of negative samples. On the other hand, GNNs build representations at the node-level for both node-level prediction tasks as well as global-level tasks [55]. Thus, it is not possible to directly apply CRD to GNNs beyond graph-level prediction. Our experiments show how a naive application of CRD for GNNs is ineffective, and ablate the impact of our design choices in formulating G-CRD.

Mini-batch negative sampling. Contrastive learning among GNN node features from teachers and students alleviates the need for specialized negative sampling procedures: we simply define negative samples for each student node feature as all the other node features from the teacher within the same mini-batch. This is markedly different from recent contrastive pre-training objectives for GNNs, which rely on handcrafted data augmentation [19], [20] or sub-graph sampling procedures [21], [22] to generate multiple views of graphs.

Unlike sample-level CRD, our approach does not require extremely large batch sizes or negative samples, as the combined cardinality of all nodes within a mini-batch is sufficient for boosting student performance. G-CRD is robust across a range of graph and batch sizes – from 25 nodes per graph in mini-batches of 32, up to single giant graphs with over 1.9 million nodes.

[2]We chose GCN as it is one of the most well studied and lightweight layers.

V. Experimental Setup

A. Datasets

We benchmark distillation techniques across a range of tasks on single large-scale networks, batches of small graphs, as well as batches of 3D point clouds, summarized in Tab.I. As we are motivated by real-world and real-time applications involving noisy and shifting data distributions, we make the following considerations for our benchmarks:

Challenging for student models. Past work on distillation for GNNs [33] used small-scale PPI [2] (node classification) and ModelNet [65] (3D point cloud classification), both of which are considered saturated by the community [11], [66]. The performance gap between cumbersome teachers and lightweight student models is negligible for these datasets, which makes them unsuitable benchmarks for comparing techniques. (See the Appendix for an investigation on PPI.)

We evaluate node classification on ARXIV and the Microsoft Academic Graph (MAG) [11], [64], which are 70× and 800× larger than PPI, respectively, and consider the more challenging semantic segmentation task on 3D point cloud scenes from S3DIS [39] that are over 10× denser than CAD object scans from ModelNet. We additionally evaluate graph classification on MOLHIV from OGB/MoleculeNet [63].

Out-of-distribution evaluation. Unlike PPI and ModelNet, all datasets involve realistic and carefully curated train-test splitting procedures to evaluate out-of-distribution generalization. ARXIV and MAG follow a temporal split, MOLHIV trains on common molecular scaffolds while testing on rare ones, and S3DIS is split according to 6 distinct areas.

B. Teacher and Student Architectures

Tab.II summarizes our choice of teachers and students, which are made with the following considerations about deploying GNNs: (1) *depth* – more message passing rounds allow models to access larger sub-graphs around each node, at the cost of training and inference time; (2) *hidden channels* – especially for giant graphs, increasing hidden channels boosts performance as well as memory consumption, and may require specialized sampling procedures; and (3) *architectural complexity* – principled geometric priors [58] or attention mechanisms over edges [4], [16] improve model expressivity but tend to have higher inference latency and memory usage.

Unlike in [33], we benchmark distillation techniques across *heterogeneous* architecture families with significant mismatch in terms of parameter count, expressive power, and inference time; *e.g.* for social networks, we distil from Graph Attention Networks [4] to simple GCNs [1]. For 3D point clouds, we distil from Kernel Point ConvNet [58] teachers to lightweight PointNet++ [57] and voxel-based MinkowskiNet [59]. We give further rationale for our choice of teacher-student pairs for each experiment in Sec.VI.

C. Training and Evaluation

Our implementation is built upon PyTorch Geometric [68], PyTorch Points 3D [62], and Open Graph Benchmark [11]. We follow the best practices and guidelines for each dataset.

TABLE I: **Summary of datasets.** ARXIV, MAG, and MOLHIV were accessed via OGB [11] and S3DIS via PyTorch Points 3D [62].

Name	#Samples	Avg. #Nodes	Avg. #Edges	Split Scheme	Split Ratio	Prediction	Task	Metric
MOLHIV [63]	41,127	25.5	27.5	Scaffold	80/10/10	Graph-level	Binary clf.	ROC-AUC
S3DIS [39]	5,845	20,000	-	Rooms	52/19/29	Point-level	Multi-class clf. (16)	mIoU, mAcc
ARXIV [11]	1	169,343	1,166,243	Time	54/18/28	Node-level	Multi-class clf. (40)	Accuracy
MAG [64]	1	1,939,743	21,111,007	Time	85/9/6	Node-level	Multi-class clf. (349)	Accuracy
PPI [2]	24	2,372	34,113	Random	84/8/8	Node-level	Multi-label (128) Binary clf.	F1

TABLE II: **Summary of teacher and student model architectures.** Peak GPU usage and average inference latency are reported over test set graphs with fixed batch size 1 with single-threading on a single GPU (RTX3090 for ARXIV/MAG, V100 for MOLHIV/S3DIS). For MAG, we exclude $136,440,832$ extra embedding dictionary parameters from #Param. and their lookup time in I.Time.

	Name	Type	#Layer, #Hidden	#Parameters		Peak GPU Usage		Average I.Time	
MOLHIV	GIN-E [19]	Teacher	5, 300	3,336,306		1.2GB		9ms	
	PNA [16]	Teacher	5, 150	2,433,901		1.2GB		18ms	
	GCN [1]	Student	2, 100	40,801	(↓98%)	1.1GB	(↓9%)	4ms	(↓78%)
	GIN [14]	Student	2, 50	10,951	(↓99%)	1.1GB	(↓9%)	2ms	(↓89%)
S3DIS	SPVCNN [60]	Teacher	U-Net	21,777,421		1.8GB		88ms	
	KP-FCNN [58]	Teacher	U-Net	14,082,688		4.8GB		227ms	
	PointNet++ [57]	Student	U-Net, SSG	1,400,269	(↓90%)	1.2GB	(↓75%)	22ms	(↓90%)
	MinkNet [59]	Student	U-Net, 20%cr.	852,512	(↓94%)	1.2GB	(↓75%)	81ms	(↓64%)
ARXIV	GAT [4]	Teacher	3, 750	1,441,580		6.8GB		93ms	
	GCN [1]	Student	2, 256	43,816	(↓97%)	2.2GB	(↓68%)	11ms	(↓88%)
	GraphSage [2]	Student	2, 256	86,824	(↓94%)	2.1GB	(↓69%)	5ms	(↓95%)
	SIGN [23]	Student	3, 512	3,566,128	(↑147%)	9.3GB	(↑36%)	5ms	(↓95%)
MAG	R-GCN [67]	Teacher	3, 512	5,575,540		20.8GB		189ms	
	R-GCN [67]	Student	2, 32	169,428	(↓97%)	11.1GB	(↓47%)	134ms	(↓29%)

Training. Teacher models are trained via the conventional supervised learning paradigm and the final weights for distillation are selected by early stopping. Student models are trained via the knowledge distillation pipeline described in Sec.III. For the OGB datasets, we follow their example implementations and training setups for both teacher and student models. For S3DIS, the teacher models are trained for 600 epochs following their original learning rate strategies, while the student models are trained for 300 epochs and use an exponential learning rate strategy. We use the conventional data preparation and augmentation procedures for S3DIS via PyTorch Points 3D.

Evaluation. For the OGB datasets, we use the official evaluators and report the average test performance across 8/10 random seeds (each teacher-student pair is re-trained for each seed). For S3DIS, we follow best practices in the literature: all models are trained once, and we report the average mIoU and mAcc over 10 voting runs on the held-out Area-5 test set using the evaluation protocol from PyTorch Points 3D.

Baselines and hyperparameters. Following Yang et al. [33], we compare the GNN representation distillation techniques G-CRD, GSP and LSP to logit-based KD [31] (2) as well as FitNet [44] and Attention Transfer (AT) [45], two feature mimicking baselines adapted from 2D computer vision which are formulated as: $\mathcal{L}_{\text{Aux}} = \sum_{i \in \mathcal{V}} \|P^T(f_i^T) - P^S(f_i^S)\|_2^2$ (FitNet uses L2 normalized projection head, AT uses attention mapping). We tune the loss balancing weights α, β in (3) for all techniques on the validation set. For KD, we tune $\alpha \in \{0.8, 0.9\}, \tau_1 \in \{4, 5\}$. For FitNet, AT, LSP, and GSP, we tune $\beta \in \{100, 1000, 10000\}$ and the kernel in (4) (only LSP, GSP). For G-CRD, we tune $\beta \in \{0.01, 0.05\}, \tau_2 \in \{0.05, 0.075, 0.1\}$ and the projection head in (9). When comparing representation distillation methods, we set α to 0 in order to ablate performance, as in [38], and reduce β by one order of magnitude.

VI. Results

A. Molecular Graph Property Prediction

We consider the graph-level property prediction task over batches of molecular graphs. Reducing the inference latency of GNNs for molecules speeds up high throughput virtual screening [63]. Additionally, virtual screening on proprietary data will require homomorphically encrypted models, which further demand low layer count and hidden size [69]. As teachers, we consider 5-layer deep GIN-E [19] and PNA [16] augmented with virtual nodes, two strong OGB baselines. Notably, PNA is the most expressive message passing GNN but explicitly materializes messages over graph edges, leading to higher memory requirement and inference latency. Our student architectures are 2-layer GCN [1] and GIN [14], which are comparatively less expressive and do not use virtual nodes (and edge features in GIN), while having low inference latency.

In Tab.III, we compare logit-based KD and the representation distillation techniques across a range of teacher-student pairs. **We find that the implicit G-CRD technique consistently improves over the supervised student's performance and outperforms the explicit global structure preserving approach, GSP. In turn, GSP outperforms the purely local approach, LSP.** While all other distillation techniques do offer minor performance boosts over the supervised student, G-CRD is the only one which improves over the KD baseline for most teacher-student combinations.

B. 3D Point Cloud Semantic Segmentation

Semantic segmentation of 3D scene graphs is a safety critical real-time task with applications in autonomous driving and robotics. Models process 3D scenes as sets of points or voxel grids, and make a dense prediction by assigning semantic categories to each point/unit. Recent state-of-the-art architectures involve strong geometric priors and complex architectures, leading to increased inference latency and GPU memory requirement. Here, we consider distilling two powerful models, Kernel Point Convolution (KP-FCNN, rigid kernel) [58] and Sparse Point-Voxel CNN (SPVCNN) [60], [61], into simpler models with low inference latency and memory usage: PointNet++ with single-scale grouping [57] and voxel-based MinkowskiNet [59] at low 20% channel ratio.

In Tab.IV, we see trends that are consistent with the previous section: G-CRD consistently improves the performance of lightweight students compared to GSP, and is particularly effective for boosting PointNet++. **Notably, due to a large mismatch in teacher-student representation capacity, feature mimicking losses, FitNet and AT, may worsen the student's performance over purely supervised learning.**

TABLE III: **Molecular graph classification on MOLHIV** (metric: ROC-AUC (%)). **Bold**/underlined denote the best/second best performing distillation technique for each column. The arrows (↑)/(↓) denote performance improvement/regression compared to logit-based Knowledge Distillation (KD). We report the average performance and std. across 8 random seeds.

	Teacher (#Layer,#Param): **Student** (#Layer,#Param):	**GIN-E** (5L,3.3M) **GCN** (2L,15K)	**PNA** (5L,2.4M) **GCN** (2L,15K)	**GIN-E** (5L,3.3M) **GCN** (2L,40K)	**PNA** (5L,2.4M) **GCN** (2L,40K)	**PNA** (5L,2.4M) **GIN** (2L,10K)
Sup.	Supervised Teacher	77.69 ±1.61	77.48 ±1.71	77.69 ±1.61	77.48 ±1.71	77.48 ±1.71
Sup.	Supervised Student	73.02 ±1.46	73.02 ±1.46	73.65 ±1.50	73.65 ±1.50	73.03 ±2.02
Distillation	KD [31]	74.08 ±1.03	74.13 ±1.72	75.25 ±1.71	74.45 ±1.27	73.42 ±2.14
Distillation	FitNet [44]	73.62 ±1.05 (↓)	73.65 ±1.25 (↓)	74.52 ±1.33 (↓)	74.39 ±1.46 (↓)	72.88 ±0.89 (↓)
Distillation	AT [45]	73.85 ±0.85 (↓)	73.64 ±1.50 (↓)	74.94 ±0.97 (↓)	73.89 ±1.92 (↓)	73.87 ±2.28 (↑)
Distillation	LSP [33]	73.58 ±1.29 (↓)	73.24 ±1.67 (↓)	75.04 ±1.20 (↓)	74.43 ±1.58 (↓)	70.74 ±1.82 (↓)
Distillation	GSP	72.83 ±1.30 (↓)	73.74 ±0.93 (↓)	75.12 ±1.27 (↓)	75.09 ±1.48 (↑)	69.68 ±2.88 (↓)
Distillation	G-CRD (Ours)	**74.34** ±1.44 (↑)	**75.11** ±0.73 (↑)	**75.53** ±1.64 (↑)	**75.89** ±0.80 (↑)	**75.77** ±2.02 (↑)

TABLE IV: **3D Semantic segmentation on S3DIS** (metric: mIoU, mAcc). **Bold**/underlined denote the best/second best performing distillation technique for each column in terms of mIoU. The arrows (↑)/(↓) denote performance improvement/regression compared to logit-based Knowledge Distillation (KD). We report the average performance over 10 voting runs.

	Teacher (#Param): **Student** (#Param):	**KP-FCNN** (14.0M) **PointNet, SSG** (1.4M)	**SPVCNN** (21.8M) **PointNet, SSG** (1.4M)	**KP-FCNN** (14.0M) **MinkNet, 20%cr.** (0.8M)	**SPVCNN** (21.8M) **MinkNet, 20%cr.** (0.8M)
Sup.	Supervised Teacher	62.70, 69.54	64.58, 71.71	62.70, 69.54	64.58, 71.71
Sup.	Supervised Student	49.67, 57.51	49.67, 57.51	55.29, 64.14	55.29, 64.14
Distillation	KD [31]	51.89, 59.48	51.81, 59.22	56.00, 64.90	55.78, 64.51
Distillation	FitNet [44]	49.37, 57.35 (↓)	49.85, 57.78 (↓)	48.94, 57.26 (↓)	53.62, 63.14 (↓)
Distillation	AT [45]	51.82, 59.38 (↓)	49.57, 57.13 (↓)	56.02, 64.87 (↑)	54.84, 63.78 (↓)
Distillation	LSP [33]	50.69, 58.49 (↓)	51.07, 58.57 (↓)	54.20, 63.59 (↓)	55.20, 64.34 (↓)
Distillation	GSP	53.00, 61.15 (↑)	51.68, 60.35 (↓)	55.50, 65.19 (↓)	54.77, 63.88 (↓)
Distillation	G-CRD (Ours)	**53.15, 61.15** (↑)	**53.27, 61.29** (↑)	**56.07, 64.87** (↑)	**55.83, 65.03** (↑)

C. Node Classification on Social Networks

Reducing model depth and feature dimensions boosts inference and reduces memory usage for real-time applications on large-scale graphs. For ARXIV, a homogeneous citation network where models classify the subject of each node/paper, we use a strong 3-layer GAT [4] teacher. GAT has high memory requirement during training and inference due to its attention mechanism. We distil into 2-layer GCN [1] and GraphSage [2], which are comparatively more scalable but lack the expressivity of attention. We also distil to structure-agnostic SIGN [23] designed for parallelized large-graph processing. LSP cannot be used for SIGN as it does not make use of graph structure.

Experiments on the giant heterogeneous Microsoft Academic Graph (MAG) of authors, papers, and institutions further tests the scalability of distillation. We distil among Relational-GCNs [67] at different depth and feature sizes. We use the GraphSAINT [30] sampler to fit GPU memory.

In Tab.V, when considering each technique independently, logit-based KD is the best while G-CRD is the second best. We believe this can be explained by the homophily phenomenon for node labels on social networks [72]. The soft labels from the teacher model thus providing a more informative signal for knowledge transfer than the latent representations. On combining KD with representation distillation, KD + G-CRD consistently outperforms all other pairs. **GSP is the worst performing technique for MAG, demonstrating the inability of explicit global topology preservation to scale to large graphs.** Note that performance gains from distillation may seem marginal, but are significant as metrics are averaged over several thousand nodes (48K for ARXIV, 42K for MAG).

D. Does Distilation Preserve Topology?

Across Tab.III, V, IV, we have observed that G-CRD, which implicitly preserves global topology from the teacher to the student embedding space, outperforms explicit structure preserving approaches LSP and GSP, as well as feature mimicking baselines FitNet and AT. In order to unpack the efficacy of G-CRD beyond performance metrics, we measure how well distillation techniques preserve both global and local topology from the teacher to the student node embedding space in Tab.VI. To quantify global representational similarity between the teacher and student, we use the recently proposed Centered Kernel Alignment score (CKA) [70] between two embedding sets, as well as the classical Mantel Test [71], [73][3] of Pearson correlation between all pairwise cosine distances from two embedding sets (as in the GSP loss). Additionally, we quantify local structural similarity via another Mantel Test which only considers distances over pre-defined edges (as in the LSP loss).

Our results largely follow the intuitions developed in Fig.1: In terms of global topology and representational similarity, students trained with FitNet, GSP, and G-CRD have high correlations to the teacher. On the other hand, LSP is the most correlated for local topology over existing graph edges, but relatively poorer at preserving global topology. **Overall, we see that the implicit contrastive approach G-CRD strikes a balance between preserving both local and global relationships, while the explicit LSP/GSP are best at preserving only one or the other.**

[3]We use the implementation from scikit-bio (skbio.stats.distance.mantel).

TABLE V: **Node classification on ARXIV and MAG** (metric: Accuracy (%)). **Bold**/underlined denote the best/second best performing distillation technique for each column. The arrows (↑)/(↓) denote performance improvement/regression compared to logit-based Knowledge Distillation (KD). We report the average performance and std. across 10 random seeds.

	Dataset:	ARXIV	ARXIV	ARXIV	MAG
	Teacher (#Layer,#Param):	**GAT** (3L,1.4M)	**GAT** (3L,1.4M)	**GAT** (3L,1.4M)	**R-GCN** (3L,5.5M)
	Student (#Layer,#Param):	**GCN** (2L,44K)	**GraphSage** (2L,87K)	**SIGN** (3L,3.5M)	**R-GCN** (2L,170K)
Sup.	Supervised Teacher	73.91 ±0.12	73.91 ±0.12	73.91 ±0.12	49.48 ±0.35
Sup.	Supervised Student	71.25 ±0.28	70.97 ±0.23	71.98 ±0.16	46.22 ±0.31
Distillation	KD [31]	71.55 ±0.25	**71.44** ±0.10	**72.26** ±0.11	**46.65** ±0.20
Distillation	FitNet [44]	71.38 ±0.17 (↓)	70.78 ±0.25 (↓)	71.98 ±0.13 (↓)	46.15 ±0.24 (↓)
Distillation	AT [45]	70.44 ±0.28 (↓)	70.17 ±0.11 (↓)	71.99 ±0.14 (↓)	46.09 ±0.27 (↓)
Distillation	LSP [33]	71.52 ±0.22 (↓)	70.95 ±0.22 (↓)	-	46.23 ±0.41 (↓)
Distillation	GSP	71.41 ±0.31 (↓)	70.98 ±0.33 (↓)	71.99 ±0.12 (↓)	46.04 ±0.15 (↓)
Distillation	G-CRD (Ours)	**71.64** ±0.16 (↑)	71.15 ±0.12 (↓)	72.10 ±0.10 (↓)	46.42 ±0.20 (↓)
KD + Dist.	KD + FitNet [44]	71.10 ±0.19 (↓)	71.11 ±0.17 (↓)	72.31 ±0.08 (↑)	46.60 ±0.36 (↓)
KD + Dist.	KD + AT [45]	70.91 ±0.31 (↓)	71.06 ±0.19 (↓)	72.27 ±0.11 (↑)	46.59 ±0.41 (↓)
KD + Dist.	KD + LSP [33]	71.35 ±0.23 (↓)	71.34 ±0.17 (↓)	-	46.73 ±0.31 (↑)
KD + Dist.	KD + GSP	71.39 ±0.19 (↓)	71.51 ±0.18 (↑)	72.27 ±0.16 (↑)	46.49 ±0.18 (↓)
KD + Dist.	KD + G-CRD (Ours)	**71.57** ±0.23 (↑)	**71.59** ±0.15 (↑)	**72.32** ±0.11 (↑)	**46.78** ±0.32 (↑)

TABLE VI: **Topological similarity measures among teacher and student embeddings on ARXIV**. We report average metrics across 10 random seeds for teacher and student validation set node embeddings. CKA Score (range: $[0, 1]$) measures global representational similarity between embedding sets. Global Mantel Test (range: $[-1, 1]$) measures Pearson correlation between the pairwise cosine distance matrices of embedding sets, *i.e.* all pairwise relationships or global structural similarity. Local Mantel Test measures local structural similarity by only considering distances over pre-defined edges. Colors denote the **highest**/**second**/**third** highest similarity for each column.

Similarity Metric:	**CKA Score** [70]		**Global Mantel Test** [71]		**Local Mantel Test** [71]	
Teacher (#Layer,#Param):	**GAT** (3L,1.4M)	**GAT** (3L,1.4M)	**GAT** (3L,1.4M)	**GAT** (3L,1.4M)	**GAT** (3L,1.4M)	**GAT** (3L,1.4M)
Student (#Layer,#Param):	**GCN** (2L,44K)	**GraphSage** (2L,87K)	**GCN** (2L,44K)	**GraphSage** (2L,87K)	**GCN** (2L,44K)	**GraphSage** (2L,87K)
Sup. Teacher	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
Sup. Student	0.655 ±0.011	0.609 ±0.007	0.680 ±0.011	0.623 ±0.008	0.695 ±0.005	0.521 ±0.007
KD [31]	0.716 ±0.011	0.721 ±0.003	0.733 ±0.007	0.730 ±0.004	0.710 ±0.006	0.589 ±0.007
KD + FitNet [44]	**0.740** ±0.006	**0.764** ±0.005	**0.760** ±0.005	**0.765** ±0.005	**0.736** ±0.008	0.576 ±0.009
KD + AT [45]	0.570 ±0.020	0.696 ±0.013	0.593 ±0.015	0.687 ±0.012	0.669 ±0.009	0.478 ±0.007
KD + LSP [33]	0.714 ±0.006	0.708 ±0.005	0.697 ±0.011	0.683 ±0.005	**0.772** ±0.006	**0.648** ±0.008
KD + GSP	**0.746** ±0.010	**0.752** ±0.004	**0.756** ±0.006	**0.757** ±0.003	0.722 ±0.007	**0.594** ±0.009
KD + G-CRD (Ours)	**0.725** ±0.007	**0.727** ±0.003	**0.750** ±0.004	**0.753** ±0.003	**0.742** ±0.004	**0.596** ±0.006

E. Ablation Study of Contrastive Loss Design

This section highlights the key design choices that distinguish G-CRD from Contrastive Representation Distillation (CRD) [38], which introduced a contrastive distillation objective for 2D image classification. CRD is specialized for global per-sample level tasks: it contrasts linearly projected global representations and uses the loss in (10). On the other hand, G-CRD performs contrastive learning at the node-level using an InfoNCE-style loss (7) and a non-linear projection head design (9); it is flexible to handle both node and global-level tasks.

In Tab.VII, we compare G-CRD and our adaptation of CRD to GNNs. In the first set of rows, we see that contrasting representations at the node-level (as in G-CRD) significantly outperforms global-level contrastive learning (as in CRD) as well as sample-wise node-level (only contrasting among nodes belonging to the same sample). In the second set of rows, we find that the G-CRD loss outperforms the CRD objective across graph as well as node-level tasks when fixing the contrastive learning at the node-level. In the third set of rows, we show that the structure-aware GCN projection head boosts performance over a structure-agnostic MLP projection. Finally, we show that the full G-CRD implementation consistently outperforms a naive CRD implementation at graph-level tasks.

TABLE VII: **Ablation study for contrastive loss design** (metric: ROC-AUC (%) for MOLHIV; Accuracy (%) for ARXIV).

Repr.	**Loss**	**Proj.**	**Dataset**: MOLHIV **PNA** (2.4M) **GCN** (15K)	MOLHIV **PNA** (2.4M) **GCN** (40K)	ARXIV **GAT** (1.4M) **GCN** (44K)
Nodes	$\mathcal{L}_{\text{G-CRD}}$	MLP	**74.72** ±0.64	**75.47** ±0.79	71.56 ±0.13
Nodes (s.w.)	$\mathcal{L}_{\text{G-CRD}}$	MLP	74.09 ±1.65	74.74 ±0.86	N.A.
Global	$\mathcal{L}_{\text{G-CRD}}$	MLP	73.32 ±1.24	73.87 ±1.43	N.A.
Nodes	$\mathcal{L}_{\text{G-CRD}}$	MLP	**74.72** ±0.64	**75.47** ±0.79	**71.56** ±0.13
Nodes	$\mathcal{L}_{\text{CRD}}$	MLP	73.62 ±1.97	75.20 ±1.18	71.48 ±0.15
Nodes	$\mathcal{L}_{\text{G-CRD}}$	GCN	**75.11** ±0.73	**75.89** ±0.80	**71.64** ±0.16
Nodes	$\mathcal{L}_{\text{G-CRD}}$	MLP	74.72 ±0.64	75.47 ±0.79	71.56 ±0.13
Full G-CRD: Nodes + $\mathcal{L}_{\text{G-CRD}}$		+ GCN Proj.	**75.11** ±0.73	**75.89** ±0.80	71.64 ±0.16
Full CRD: Global + $\mathcal{L}_{\text{CRD}}$		+ Lin. Proj.	73.67 ±1.55	73.37 ±2.08	N.A.

F. Quantizability of Distilled Representations

Quantization of model weights and activations to lower precision arithmetic is a complementary compression technique to distillation. Models at lower precision such as 8-bit integers have significantly faster inference latency and lower memory usage. However, quantization aware training (QAT) is known to degrade performance. In Tab.VIII, we explore whether distillation can improve the performance of a lightweight quantized 2-layer GIN model when trained with vanilla QAT as well as DegreeQuant [29], a GNN-specific QAT. We find that QAT with G-CRD enables quantized models to retain a

TABLE VIII: **Molecular graph classification on MOLHIV** under INT8 quantization aware training (metric: ROC-AUC (%)).

	Teacher (#Param):	**PNA** @FP32 (2.4M)	**PNA** @FP32 (2.4M)
	Student (#Param):	**GIN** @INT8 (10K)	**GIN** @INT8 (10K)
	Training Scheme:	Vanilla QAT	DegreeQuant [29]
Sup.	Teacher	77.48 ±1.71	77.48 ±1.71
Sup.	Student	70.62 ±3.05	71.38 ±2.38
Distillation	KD [31]	72.93 ±1.14	71.74 ±1.97
Distillation	FitNet [44]	73.03 ±1.89 (↑)	71.08 ±3.86 (↓)
Distillation	AT [45]	72.76 ±1.02 (↓)	69.31 ±2.12 (↓)
Distillation	LSP [33]	70.91 ±2.23 (↓)	68.94 ±4.23 (↓)
Distillation	GSP	66.44 ±4.06 (↓)	68.84 ±2.29 (↓)
Distillation	G-CRD (Ours)	**73.65** ±2.09 (↑)	**73.50** ±1.13 (↑)

TABLE IX: **Molecular graph classification/regression on small-scale MOL* datasets** (metric: ROC-AUC (%) for BACE, SIDER; MSE for ESOL). We perform linear probing on frozen node representations from GCN (2L,40K) initialized randomly or pre-trained on MOLHIV via supervised learning or various distillation techniques.

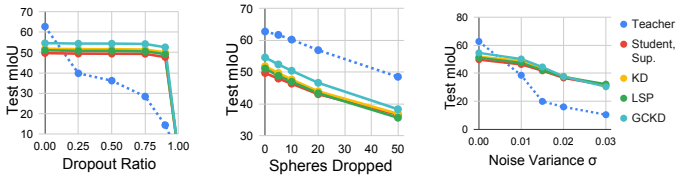
Initialization	MOLBACE	MOLSIDER	MOLESOL
Random	67.03 ±1.63	55.35 ±1.05	2.097 ±0.128
Supervised	71.32 ±2.48	58.40 ±1.18	1.907 ±0.064
KD [31]	72.24 ±2.38	**59.50** ±0.73	1.948 ±0.084
FitNet [44]	71.82 ±4.17 (↓)	57.31 ±1.13 (↓)	2.032 ±0.137 (↓)
AT [45]	69.73 ±3.57 (↓)	57.62 ±0.43 (↓)	1.820 ±0.062 (↑)
LSP [33]	70.03 ±2.12 (↓)	58.67 ±0.71 (↓)	1.998 ±0.058 (↓)
GSP	70.53 ±1.74 (↓)	58.57 ±1.13 (↓)	1.948 ±0.070 (↑)
G-CRD (Ours)	**72.46** ±2.19 (↑)	56.81 ±1.42 (↓)	**1.812** ±0.112 (↑)

large portion of their performance at 8-bit integer precision as compared to other distillation techniques. Significantly, GIN trained with G-CRD at INT8 (73.65% ROC-AUC) can still outperform its purely supervised counterpart at full 32-bit floating point precision (73.03%, see column 6 in Tab.III).

G-CRD is well suited for QAT because the projection heads ensure that contrastive distillation take place at full precision even when the student model is at low INT8 (the heads do not interfere with the inference phase, which always uses INT8). On the other hand, structure preserving approaches LSP and GSP are particularly ill suited as they match relationships from low INT8 feature vectors (student) to full precision (teacher).

G. Transferability of Distilled Representations

In biomedical discovery, we are interested in models' ability to generalize or extrapolate to unseen regions of chemical space. In addition to evaluating on out-of-distribution molecular scaffolds from MOLHIV's test set, we further test the transferability of distilled models on smaller scale molecule datasets from OGB: MOLBACE (1,513 samples), MOLSIDER (1,427 samples), and MOLESOL (1,128 samples) in Tab.IX. We perform linear probing on frozen node features from a 2-layer GCN model. The GCN feature extractor can either be initialized randomly, pre-trained on MOLHIV via supervised learning, or pre-trained via various distillation techniques with a 5-layer PNA teacher. Encouragingly, we find that distillation leads to more transferable representations than random or purely supervised initialization. Overall, G-CRD boosts the transferability of student models for 2 out of 3 datasets with structurally different molecules as well as task semantics.



(a) Sparse scans (b) Occluded scans (c) Noisy scans

Fig. 3: **Robustness analysis for 3D semantic segmentation.**

H. Robustness of Distilled Representations

Beyond clean test set performance, lightweight 3D segmentation models often have to deal with 'dirty' data such as sparse, occluded or noisy scans. In Fig.3, we evaluate the impact of distillation on the robustness of PointNet++ across three common challenging scenarios (we also show the KP-FCNN teacher for comparisson): (1) *Sparse scans* – randomly dropout a percentage of points for each scan; (2) *Partial and occluded scans* – dropout all points sampled within a number of random spheres of fixed radius of 0.5m for each scan; and (3) *Noisy scans* – add independent random noise to the 3D coordinates of each point with a variance factor σ.

Training the lightweight student with G-CRD consistently improves its robustness compared to purely supervised training as well as logit-based KD [31] and LSP [33]. Promisingly, lightweight students can be more robust to sparse or noisy scans than cumbersome teachers.

VII. Conclusion

In this work, we study representation distillation for GNNs by training lightweight models to preserve global topology of embeddings from more expressive teacher models. We introduce Graph Contrastive Representation Distillation (G-CRD), the first contrastive distillation technique specialized for GNNs. G-CRD uses contrastive learning to implicitly align the student node embeddings to those of the teacher, preserving structural relationships among graph edges as well as latent interactions among disconnected nodes.

Additionally, we introduce an expanded set of benchmarks on large-scale real-world datasets where the performance gap between teacher and student GNNs is non-negligible. This was missing from the LSP study which used synthetic and saturated datasets. Our experiments reveal that training lightweight GNN models with G-CRD consistently improves their performance, robustness, and quantizability compared to explicit structure preserving approaches (LSP, GSP) as well as baselines adapted from 2D computer vision. We further unpack the efficacy of G-CRD over GSP and LSP through the lens of representational similarity of teacher and student embedding spaces.

Currently, distillation techniques for GNNs are neither as effective nor as well understood as their counterparts in 2D computer vision. We hope that our techniques and benchmarks can form the basis for comparison in future work on this emerging research direction.

ACKNOWLEDGEMENTS

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds (Project No.A19E3b0099 and Project No. A20H6b0151). We would like to thank Efe Camci, Vijay Prakash Dwivedi, Yoon Ji Wei, Edwin Khoo, Hannes Stärk, Shyam A. Tailor, and Wanyue Zhang for helpful comments and discussions.

REFERENCES

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[2] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017.
[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM TOG*, 2019.
[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *ICLR*, 2018.
[5] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint*, 2018.
[6] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *KDD*, 2018.
[7] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," *arXiv preprint*, 2019.
[8] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann *et al.*, "A deep learning approach to antibiotic discovery," *Cell*, 2020.
[9] P. Gainza, F. Sverrisson, F. Monti, E. Rodola, D. Boscaini, M. Bronstein, and B. Correia, "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning," *Nature Methods*, 2020.
[10] Y. Long, M. Wu, C. K. Kwoh, J. Luo, and X. Li, "Predicting human microbe–drug associations via graph convolutional network with conditional random field," *Bioinformatics*, 2020.
[11] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *arXiv preprint*, 2020.
[12] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, and J. Leskovec, "Ogb-lsc: A large-scale challenge for machine learning on graphs," in *KDD Cup*, 2021.
[13] R. Addanki, P. W. Battaglia, D. Budden, A. Deac, J. Godwin, T. Keck, W. L. S. Li, A. Sanchez-Gonzalez, J. Stott, S. Thakoor *et al.*, "Large-scale graph representation learning with very deep gnns and self-supervision," in *KDD Cup*, 2021.
[14] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.
[15] G. Li, M. Müller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *ICCV*, 2019.
[16] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, "Principal neighbourhood aggregation for graph nets," *NeurIPS*, 2020.
[17] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," *arXiv preprint*, 2020.
[18] L. Bai, L. Cui, Y. Jiao, L. Rossi, and E. Hancock, "Learning backtrackless aligned-spatial graph convolutional networks for graph classification," *IEEE TPAMI*, 2020.
[19] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," in *ICLR*, 2019.
[20] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *NeurIPS*, 2020.
[21] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *KDD*, 2020.
[22] Y. Jiao, Y. Xiong, J. Zhang, Y. Zhang, T. Zhang, and Y. Zhu, "Sub-graph contrast for scalable self-supervised graph representation learning," in *ICDM*, 2020.
[23] F. Frasca, E. Rossi, D. Eynard, B. Chamberlain, M. Bronstein, and F. Monti, "Sign: Scalable inception graph neural networks," in *ICML Workshop on Graph Representation Learning and Beyond*, 2020.
[24] S. A. Tailor, F. L. Opolka, P. Liò, and N. D. Lane, "Adaptive filters and aggregator fusion for efficient graph convolutions," in *MLSys GNNSys Workshop*, 2021.
[25] G. Li, M. Müller, B. Ghanem, and V. Koltun, "Training graph neural networks with 1000 layers," in *ICML*, 2021.
[26] Y. Gao, H. Yang, P. Zhang, C. Zhou, and Y. Hu, "Graph neural architecture search." in *IJCAI*, 2020.
[27] Y. Zhao, D. Wang, X. Gao, R. Mullins, P. Liò, and M. Jamnik, "Probabilistic dual network architecture search on graphs," in *AAAI Workshop*, 2021.
[28] Y. Zhao, D. Wang, D. Bates, R. Mullins, M. Jamnik, and P. Liò, "Learned low precision graph neural networks," *arXiv preprint*, 2020.
[29] S. A. Tailor, J. Fernandez-Marques, and N. D. Lane, "Degree-quant: Quantization-aware training for graph neural networks," in *ICLR*, 2021.
[30] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "Graphsaint: Graph sampling based inductive learning method," in *ICLR*, 2020.
[31] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint*, 2015.
[32] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, 2021.
[33] Y. Yang, J. Qiu, M. Song, D. Tao, and X. Wang, "Distilling knowledge from graph convolutional networks," in *CVPR*, 2020.
[34] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," in *ICLR*, 2021.
[35] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint*, 2018.
[36] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *CVPR*, 2018.
[37] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.
[38] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *ICLR*, 2019.
[39] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *CVPR*, 2016.
[40] C. Joshi, "Transformers are graph neural networks," *The Gradient*, 2020.
[41] D. Ghosal, N. Majumder, S. Poria, N. Chhaya, and A. Gelbukh, "Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation," in *EMNLP*, 2020.
[42] C. K. Joshi, T. Laurent, and X. Bresson, "An efficient graph convolutional network technique for the travelling salesman problem," *arXiv preprint*, 2019.
[43] Q. Cappart, D. Chételat, E. Khalil, A. Lodi, C. Morris, and P. Veličković, "Combinatorial optimization and reasoning with graph neural networks," in *IJCAI*, 2021.
[44] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint*, 2014.
[45] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *arXiv preprint*, 2016.
[46] Q. Li, S. Jin, and J. Yan, "Mimicking very efficient network for object detection," in *CVPR*, 2017.
[47] G.-H. Wang, Y. Ge, and J. Wu, "In defense of feature mimicking for knowledge distillation," *arXiv preprint*, 2020.
[48] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan, "Knowledge distillation via instance relationship graph," in *CVPR*, 2019.
[49] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *CVPR*, 2019.
[50] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *ICCV*, 2019.
[51] B. Yan, C. Wang, G. Guo, and Y. Lou, "Tinygnn: Learning efficient graph neural networks," in *KDD*, 2020.
[52] C. Yang, J. Liu, and C. Shi, "Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework," *arXiv preprint*, 2021.
[53] S. Zhang, Y. Liu, Y. Sun, and N. Shah, "Graph-less neural networks: Teaching old mlps new tricks via distillation," *arXiv preprint*, 2021.
[54] Y. Chen, Y. Bian, X. Xiao, Y. Rong, T. Xu, and J. Huang, "On self-distilling graph neural network," *arXiv preprint*, 2020.
[55] D. Mesquita, A. Souza, and S. Kaski, "Rethinking pooling in graph neural networks," in *NeurIPS*, 2020.
[56] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," 2021.
[57] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NIPS*, 2017.
[58] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," *CVPR*, 2019.
[59] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *CVPR*, 2019.

[60] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel cnn for efficient 3d deep learning," in *NeurIPS*, 2019.

[61] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *ECCV*, 2020.

[62] T. Chaton, C. Nicolas, S. Horache, and L. Landrieu, "Torch-points3d: A modular multi-task frameworkfor reproducible deep learning on 3d point clouds," in *3DV*, 2020.

[63] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical science*, 2018.

[64] K. Wang, I. Shen, C. Huang, C.-H. Wu, Y. Dong, and A. Kanakia, "Microsoft academic graph: when experts are not enough," *Quantitative Science Studies*, 2020.

[65] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015.

[66] M. A. Uy, Q.-H. Pham, B.-S. Hua, D. T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *ICCV*, 2019.

[67] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European semantic web conference*, 2018.

[68] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint*, 2019.

[69] A. Al Badawi, J. Chao, J. Lin, C. F. Mun, S. J. Jie, B. H. M. Tan, X. Nan, A. M. M. Khin, and V. Chandrasekhar, "Towards the alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus," *IEEE Transactions on Emerging Topics in Computing*, 2020.

[70] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *ICML*, 2019.

[71] N. Mantel, "The detection of disease clustering and a generalized regression approach," *Cancer research*, 1967.

[72] Q. Huang, H. He, A. Singh, S.-N. Lim, and A. Benson, "Combining label propagation and simple models out-performs graph neural networks," in *ICLR*, 2021.

[73] P. Legendre and L. Legendre, *Numerical ecology*. Elsevier, 2012.

APPENDIX

A. Results on PPI

In Tab.X, we attempt to reproduce the results from Yang et. al. [33], but our codebase[4] showed significant improvements over their numbers for the same teacher-student pair. We found negligible performance difference between the two models, even under supervised learning. Thus, we redo the experiment with a more reasonable student architecture (a 2 layer GAT instead of 5 layers) and find that G-CRD leads to the best performance, followed by LSP. We believe small-scale and saturated datasets such as PPI with random splits are not suitable for benchmarking distillation techniques.

TABLE X: **Node classification on PPI** (metric: F1 (%)).

	Teacher (#Layer,#Param):	**GAT** (3L,3.6M)	**GAT** (3L,3.6M)
	Student (#Layer,#Param):	**GAT** (5L,160K)	**GAT** (2L,180K)
		Reported / Our Codebase	Our Codebase
Sup.	Supervised Teacher	97.6 / 97.95 ±0.10	97.95 ±0.10
Sup.	Supervised Student	95.7 / 97.90 ±0.45	87.55 ±1.33
Distillation	KD [31]	- / 97.93 ±0.26	88.30 ±1.81
Distillation	FitNet [44]	95.6 / 97.74 ±0.34 (↓)	87.46 ±0.78 (↓)
Distillation	AT [45]	95.4 / 97.02 ±0.76 (↓)	86.82 ±1.27 (↓)
Distillation	LSP [33]	96.1 / 97.81 ±0.31 (↓)	88.37 ±1.89 (↑)
Distillation	GNN-SD [54]	**96.2** / -	-
Distillation	G-CRD (Ours)	- / **98.42** ±0.14 (↑)	**88.38** ±1.36 (↑)

[4]We built our implementation upon the official PyG example for PPI.

TABLE XI: **Average training time per epoch and GPU usage for distillation techniques.** Time to complete one epoch and GPU usage during training are reported over training set graphs on a single RTX8000 GPU, averaged over 100 epochs.

	Dataset:	ARXIV		MOLHIV	
	Teacher (#Param):	**GAT** (1.4M)		**PNA** (2.4M)	
	Student (#Param):	**GCN** (44K)		**GCN** (40K)	
		Avg. E.Time	Avg. GPU	Avg. E.Time	Avg. GPU
Sup.	Teacher	1.190s	14.7GB	42.660s	2.3GB
Sup.	Student	0.003s	3.6GB	10.137s	2.0GB
Distillation	KD [31]	0.004s	4.2GB	24.086s	2.1GB
Distillation	FitNet [44]	0.006s	5.5GB	25.535s	2.1GB
Distillation	AT [45]	0.004s	5.0GB	24.499s	2.1GB
Distillation	LSP [33]	0.100s	13.1GB	26.484s	2.1GB
Distillation	GSP	0.027s	30.4GB	45.781s	2.3GB
Distillation	G-CRD (MLP Head)	0.028s	6.0GB	25.464s	2.1GB
Distillation	G-CRD (GCN Head)	0.066s	6.4GB	27.878s	2.1GB

TABLE XII: **Ablation study for GSP**. (metric: ROC-AUC (%) for MOLHIV; Accuracy (%) for ARXIV).

Kernel	**Dataset**: **Metric**	MOLHIV **GIN-E** (3.3M) **GCN** (15K)	MOLHIV **GIN-E** (3.3M) **GCN** (40K)	ARXIV **GAT** (1.4M) **GraphSage** (87K)
Euclidean	MSE	62.73 ±3.30	68.67 ±2.42	70.89 ±0.22
Linear	MSE	**73.01** ±1.00	74.90 ±1.60	70.98 ±0.33
Polynomial	MSE	72.58 ±0.82	74.07 ±1.60	71.00 ±0.18
RBF	MSE	72.83 ±1.30	**75.12** ±1.27	**71.05** ±0.20
RBF	KL-div.	72.46 ±0.81	74.33 ±2.24	70.96 ±0.08
RBF	MSE	**72.83** ±1.30	**75.12** ±1.27	**71.05** ±0.20

B. Training Time and GPU Usage

In Tab.XI, we report the average training time per epoch as well as GPU usage for distillation techniques. For the large-scale ARXIV graph, GSP uses significantly higher GPU memory than other approaches due to storing all-to-all pairwise similarity matrices for both teacher and student node embeddings. For our implementation of LSP (in PyG), the computation of a sparse softmax over local edges in the ARXIV graph is expensive in terms of time cost.

For mini-batched training on molecular graphs from MOLHIV, GSP has a high time cost due to building pairwise similarity matrices for each sample. Although the size of molecular graphs is very small, GSP's GPU memory cost is higher than other approaches, too.

Compared to LSP and GSP, G-CRD is scalable in terms training time and memory usage while consistently improving the performance of student models. Naturally, G-CRD with the one-layer GCN projection head has a higher time cost than the conventional MLP projection head. As a reminder, we propose to treat the choice of projection head as a hyperparameter.

C. Ablation Study for GSP

In Tab.XII, we ablate the choice of kernel and metric for GSP, which was proposed as a global structure preserving extension of LSP [33]. Like LSP, the RBF kernel lead to the best overall performance. Unlike LSP, we found directly matching teacher and student pairwise similarity matrices via MSE to perform better than normalizing the similarities via softmax followed by KL-divergence. We attribute this to the sparsity of the signal in the resulting matrices after the softmax.