Return to Basics: Very Simple Graph Contrastive Learning via Noise Cancellation Principle

Yanan Zhao, Feng Ji *, **Jingyang Dai, Jiaze Ma, Wee Peng Tay**
Nanyang Technological University,

Abstract

Graph Contrastive Learning (GCL) has shown strong promise for unsupervised graph representation learning, yet its effectiveness remains limited on heterophilic graphs, where connected nodes often belong to different classes. Existing methods rely on complex augmentation schemes, intricate encoders, or negative sampling, which raises the question of whether such complexity is truly necessary in this challenging setting. In this work, we revisit the foundations of supervised and unsupervised learning on graphs and uncover a simple yet effective principle for GCL: mitigating node feature noise by aggregating it with structural features derived from the graph topology. This phenomenological study suggests that the original node features and the graph structure naturally provide two complementary views for contrastive learning. Building on this insight, we propose an embarrassingly simple GCL model that uses a GCN encoder to capture structural features and an MLP encoder to isolate node feature noise. Our design requires neither data augmentation nor negative sampling, yet achieves state-of-the-art results on heterophilic benchmarks with minimal computational and memory overhead, while offering advantages in homophilic graphs in terms of complexity, scalability, and robustness. Moreover, a variant of GCN-MLP based on the same principle also achieves SOTA performance on homophilic datasets. We provide theoretical justification for our approach and validate its effectiveness through extensive experiments. The implementation code is available at `https://github.com/YananZhao0630/BackToBasics-GCL`

1 Introduction

Contrastive learning is a powerful unsupervised representation learning technique. It learns informative embeddings by pulling similar instances closer and pushing dissimilar ones apart, often using views generated by different encoders. Without requiring extensive labels, it has achieved strong performance in computer vision, natural language processing, and recommendation systems [1–3]. Its extension to graph-structured data is known as Graph Contrastive Learning (GCL).

The central idea of GCL is to design encoders that produce distinct yet semantically meaningful graph views. While this paradigm has shown strong promise, its effectiveness on heterophilic graphs, where connected nodes often belong to different classes, remains limited. To overcome this challenge, existing methods typically introduce additional complexity. Augmentation-based approaches generate views through perturbations such as edge removal or feature masking [4–8], often rely on heuristic pipelines that may distort graph semantics. For example, EPAGCL [8] constructs augmented views by adding or dropping edges according to weights derived from the Error Passing Rate (EPR). In contrast, augmentation-free approaches shift the complexity to the encoder, requiring sophisticated designs to extract distinct representations from the same input. PolyGCL [9] applies polynomial filters

*Correspondence to: Feng Ji <jifeng@ntu.edu.sg>.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

to generate low-pass and high-pass spectral views, while SDMG [10] employs two dedicated low-frequency encoders to facilitate diffusion-based learning. Despite these intricacies, both approaches often continue to rely on negative sampling during training, which adds further complexity. We refer readers to Appendix A for additional related work. To illustrate the trade-off between complexity

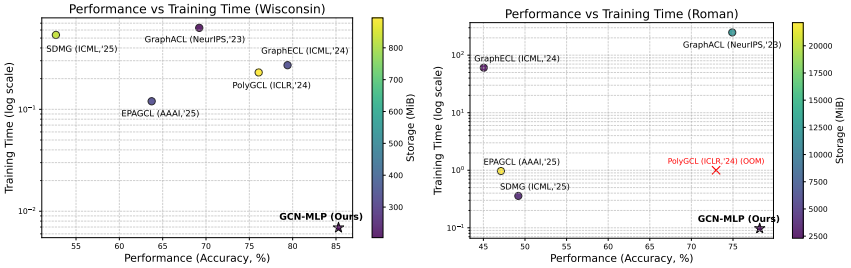


Figure 1: Performance–complexity trade-off of GCL methods on the Wisconsin and Roman. Accuracy is plotted against training time (log scale, in seconds), with marker color indicating storage cost. Our GCN-MLP achieves the best performance with minimal complexity, while OOM cases are marked with red crosses. OOM refers to out of memory on an NVIDIA RTX A5000 GPU (24GB).

and node classification performance in recent GCL methods, Fig. 1 plots accuracy against training time for each epoch, with marker color indicating storage cost. They are on the more challenging heterophilic datasets: the Wisconsin dataset and the (large-scale) Roman dataset. GraphACL'23 and PolyGCL'24 gain higher accuracy at the expense of greater complexity, while GraphECL'24, EPAGCL'25, and SDMG'25 reduce training time and storage but suffer degraded performance. This trend raises two natural questions:

Q1 *Have recent advances in GCL substantially improved performance on heterophilic graphs?*

Q2 *Are increasingly elaborate designs truly necessary, or can simpler models achieve comparable or better results?*

To address these questions, we revisit the essence of node classification. In the *ideal* case, classification is trivial when nodes of the same class share identical features. In practice, however, node features from the same class are better modeled as realizations from a common distribution. The *noise*, defined as the deviation of the feature from the distribution mean, introduces variability that complicates classification. Hence, we want to "mitigate noise", which might be (partially) achieved by aggregating features across nodes of the same class, akin to a law-of-large-numbers effect [11]. In homophilic graphs, models such as GCN leverage neighborhood aggregation under the assumption that neighbors are likely to share the same class. In contrast, for heterophilic graphs, effective strategies involve identifying non-neighboring but same-class nodes for aggregation [12]. This is usually much harder [13], as the graph topology does not provide direct information for aggregation. While labeled data provides supervision to guide class separation, unsupervised settings require stronger noise mitigation to ensure that features cluster well by class. In summary, heterophilic GCL suffers from limited guidance from both the graph structure and node labels.

However, beyond aggregating features across nodes, an alternative way to mitigate noise is to generate multiple feature representations for the same node. Our strategy is motivated by the observation that: cancellation is stronger in the sum of two vectors when they are less correlated. The key, therefore, is to construct diverse feature views such that their associated "noise" is preferably less correlated. For graph-structured data, two natural sources arise: the original node features independent of the graph topology and the embeddings from aggregating over the graph structure. We hope that their respective noises, termed *feature noise* and *structural noise*, are weakly correlated for cancellation.

Building on the above intuition, we adopt an embarrassingly *simple* GCL architecture that uses only a *GCN* and an *MLP* as view-generation encoders. The resulting GCN-MLP framework requires neither data augmentation nor negative sampling and is readily applicable to a broad range of graph datasets. This design is especially advantageous in heterophilic settings, where node features and graph topology are typically weakly correlated. We describe the method as "simple" because of its minimal and interpretable construction. Its effectiveness is illustrated in Fig. 1, while more comprehensive evaluations are presented in Section 5. We further note that the GCN encoder can be replaced by other

GNN variants, and that strongly homophilic datasets may benefit from encoders more expressive than GCN; a detailed discussion is provided in Section 4.

We emphasize that the central contribution of this work is a **noise-cancellation** principle for GCL, rather than the introduction of new encoder architectures. The key hypothesis is that, when feature noise and structural noise are weakly correlated, contrastive alignment followed by simple linear fusion yields stronger noise suppression. Accordingly, our design objective is to construct two views whose noise components are minimally correlated. The GCN-MLP framework provides a concise and effective instantiation of this idea: the GCN captures structure-informed representations together with structural noise, whereas the MLP preserves feature-space information with its associated feature noise, producing complementary views for contrastive learning. In this respect, our perspective departs from mainstream GCL research, which has largely emphasized increasingly complex encoders and augmentation strategies.

In summary, our main contributions are as follows:

- We introduce a noise-cancellation principle for GCL and suggest an augmentation-free GCN-MLP framework based on this principle. Our framework is simple, flexible, and computationally efficient. We further provide theoretical support for the selected contrastive views, elucidating the basis of the model's simplicity and robustness.
- We analyze the mechanisms underlying the model's strong performance on heterophilic datasets, and we show that the same framework remains cost-effective, scalable, and robust on homophilic datasets.
- We conduct extensive experiments across diverse benchmarks and demonstrate consistent advantages in predictive accuracy, computational efficiency, and robustness to adversarial perturbations.

This work is motivated in part by our concurrent work [14, 15], where we observed that many view designs "almost" predominantly rely on the nodes' own features (e.g., when the diffusion order parameter is very small). This observation prompts us to investigate whether a simple MLP is sufficient to design a complementary view.

2 Noise Cancellation for GCL

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the node set $\mathcal{V} = \{v_1, \ldots, v_N\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each node v_i is associated with a feature vector $\mathbf{x}_i$, which is collected as the i-th row in the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$. The graph structure is encoded by a symmetric weighted matrix $\mathbf{A} = (a_{ij})_{1 \le i,j \le n} \in \mathbb{R}^{N \times N}$, where a_{ij} is the edge weight between v_i and v_j. The complete graph data is denoted by $\mathcal{X} = (\mathbf{A}, \mathbf{X})$.

GCL belongs to the category of *unsupervised representation learning*, where no labels are available during training. For unsupervised learning, the goal is to train an encoder f_θ that maps each node v_i and its context in $\mathcal{X}$ to a representation $\mathbf{z}_i = f_\theta(\mathcal{X}, v_i) \in \mathbb{R}^F$, where F denotes the feature dimension. The resulting embedding matrix $\mathbf{Z} \in \mathbb{R}^{N \times F}$ is then used for downstream tasks such as *node classification*, which is our main focus.

For GCL, self-supervision is achieved by enforcing consistency between representations $\mathbf{Z}_1$ and $\mathbf{Z}_2$ obtained from different encoders f_{θ_1} and f_{θ_2}. As a guiding principle, the encoders f_{θ_1} and f_{θ_2} should represent different "graph views". The concept of a *graph view* is not universally defined and is open to interpretation, while a *local-global* dichotomy is popular [9]. For each node v_i, the final feature representation is a weighted sum $\beta \mathbf{z}_{1,i} + (1-\beta)\mathbf{z}_{2,i}$, where $\mathbf{z}_{1,i}$ and $\mathbf{z}_{2,i}$ are the i-th row of $\mathbf{Z}_1$ and $\mathbf{Z}_2$, respectively, and $0 < \beta < 1$. Most models simply choose $\beta = 0.5$ to avoid discriminating against any graph view.

We take a step back and examine the main challenge of unsupervised learning. In the ideal situation where the features are *noiseless*, i.e., $\mathbf{x}_i = \mathbf{x}_j$ if and only if v_i and v_j have the same label, the classification becomes trivial. However, this never happens for real datasets. More specifically, we formalize the discussion as follows.

Assume that for each label class c, the feature for a node with label c is generated according to a class-specific distribution γ_c.

Definition 1. *Let c be a class label and $\mathcal{V}_c$ be the set of nodes of label c. Define the* class centroid $\mathbf{x}_c = \mathbb{E}_{\mathbf{x} \sim \gamma_c}[\mathbf{x}]$*, and the* noise *of $v_i \in \mathcal{V}_c$ to be $\mathbf{n}_i = \mathbf{x}_i - \mathbf{x}_c$.*

In practice, a proxy for $\mathbf{x}_c$ is the empirical centroid $\widehat{\mathbf{x}}_c = (\sum_{v_i \in \mathcal{V}_c} \mathbf{x}_i)/|\mathcal{V}_c|$. Unlike in the ideal situation, $\mathbf{n}_i$ can have a large norm, which prohibits effective separation of nodes from different classes. To address this challenge, we may aim for a small ratio between the norms of the noise and class centroid, termed *noise-to-class centroid ratio* (NCR), which leads to the following two natural strategies (see more discussions at the end of the section):

S1 *Enlarge the norm of the class centroids of the output representation.*
S2 *Reduce the norm of the noise of the output representation.*

To explain how these strategies might be implemented, we consider the following simple observation (see Appendix B for the proof).

Proposition 1. *For two different representations f_{θ_1} and f_{θ_2} in the same space $\mathbb{R}^F$, let $\mathbf{z}_{1,c}, \mathbf{z}_{2,c} \neq 0$ be their respective centroids. Then, for $0 < \beta < 1$, the norm of the aggregated feature centroid $\mathbf{z}_c = \beta\mathbf{z}_{1,c} + (1-\beta)\mathbf{z}_{2,c}$ increases as the cosine similarity between $\mathbf{z}_{1,c}$ and $\mathbf{z}_{2,c}$ increases while keeping the centroid norms fixed.*
Moreover, suppose $|\mathcal{V}_c| = n_c$, and for $r = 1, 2$, let $\mathbf{z}'_{r,c}$ be the empirical centroid of the features $\{\mathbf{z}_{r,i} : 1 < i \leq n_c\}$ (i.e., excluding the node v_1). Let $\mathbf{n}'_{r,1} = \mathbf{z}_{r,1} - \mathbf{z}'_{r,c}$ be the deviation of v_1's feature from $\mathbf{z}'_{r,c}$. Then, the norm of the output noise $\mathbf{n}_1 = \mathbf{z}_1 - \mathbf{z}_c$, where $\mathbf{z}_1 = \beta\mathbf{z}_{1,1} + (1-\beta)\mathbf{z}_{2,1}$, is a non-decreasing function of the cosine similarity between $\mathbf{n}'_{1,1}$ and $\mathbf{n}'_{2,1}$ while keeping the deviation norms fixed.

Intuitively, recall that we seek cancellation between $\mathbf{n}_{1,i}$ and $\mathbf{n}_{2,i}$, while avoiding it between $\mathbf{z}_{1,c}$ and $\mathbf{z}_{2,c}$. The observation suggests that to generate output features $\mathbf{Z}$ with a small NCR, we need to ensure that $\mathbf{z}_{1,c}$ and $\mathbf{z}_{2,c}$ are strongly correlated, while for each v_i, their respective noise (to centroid) $\mathbf{n}_{1,i}$ and $\mathbf{n}_{2,i}$ are weakly correlated (see empirical evidence in Appendix D.1). Supporting evidence regarding the effect of our model on NCR is given in Appendix D.2.

On implementing the strategies Contrastive learning is deemed to amplify the class centroid via a contrastive loss (minimizing pairwise feature cosine similarity). More specifically, the learning process seeks to align the centroids of distinct views. The dedicated loss encourages these centroids to form a small angle, thereby reducing cancellation during aggregation.

However, a similar approach to reducing noise through a dedicated loss is not as straightforward, since labels are unavailable during training. Centroids, and hence noise, cannot be computed explicitly. Instead, the idea is to design encoders with different characteristics so that their respective noise is intrinsically less correlated. We provide the motivation in the next section.

3 Feature and Structural Noise

As we have envisioned in the previous section, we motivate the model design aiming for noise reduction. Any graph dataset naturally consists of two pieces of information: *features* and the *graph structure*. We formalize earlier discussions and associate each with a notion of "noise". We analyze their correlations, in alignment with the objective of noisy reduction as discussed in the previous section. Let $\widetilde{\mathbf{A}}_{\mathcal{G}}$ be the asymmetric normalized adjacency matrix.[2] For a matrix $\mathbf{M}$, we use $\mathbf{M}_i$ to denote the i-th row vector of $\mathbf{M}$.

Definition 2. *For any feature matrix $\mathbf{X}$, its associated* feature noise *of a node v_i with label c is $\mathbf{n}_i = \mathbf{x}_i - \mathbf{x}_c$. For a fixed $k > 0$, the k-*hop structural noise *(or simply the* structural noise*) of v_i with class label c is the feature noise $\mathbf{n}_i^{(k)}$ associated with the transformed feature matrix $\widetilde{\mathbf{A}}_{\mathcal{G}}^k\mathbf{X}$, defined as follows:*

$$\mathbf{n}_i^{(k)} = \left(\widetilde{\mathbf{A}}_{\mathcal{G}}^k\mathbf{X}\right)_i - \mathbf{M}_i \tag{1}$$

with

$$\mathbf{M}_i = \mathbb{E}\left[\frac{1}{|\mathcal{V}_c|}\sum_{v_j \in \mathcal{V}_c}\left(\widetilde{\mathbf{A}}_{\mathcal{G}}^k\mathbf{X}\right)_j\right] = \frac{1}{|\mathcal{V}_c|}\sum_{v_j \in \mathcal{V}_c}\left(\widetilde{\mathbf{A}}_{\mathcal{G}}^k\overline{\mathbf{X}}\right)_j, \tag{2}$$

where $\overline{\mathbf{X}}$ is the mean feature matrix whose j-th row is $\mathbf{x}_c$ if v_j has label c.

[2]Notice that the usual GCN uses the symmetric normalized adjacency matrix. However, the asymmetric version is more convenient for theoretical analysis. In Appendix D.3, we demonstrate that using either version has comparable performance.

Observe that when $k = 0$, the 0-hop structural noise reduces to ordinary feature noise associated with $\mathbf{X}$. We single out this case since no graph information is involved. Recall that our objective is to obtain features with less correlated noise. Although k-hop structural noise still contains the original feature noise, its effect is attenuated, as a consequence of the following result.

Proposition 2. *Given the feature matrix $\mathbf{X}$, let $\overline{\mathbf{X}}$ be the mean feature matrix, where the i-th row is $\mathbf{x}_c$ if v_i has label c. Then as k increases, the features $\widetilde{\mathbf{A}}_{\mathcal{G}}^k\mathbf{X}$ is close to $\widetilde{\mathbf{A}}_{\mathcal{G}}^k\overline{\mathbf{X}}$, with high probability.*

We provide its formal version Proposition 3 and the proof in Appendix B. Intuitively, as $\overline{\mathbf{X}}$ is *unambiguous* in the sense that it trivially separates all classes, the "uncertainty" or "noise" of $\widetilde{\mathbf{A}}_{\mathcal{G}}^k\overline{\mathbf{X}}$ is solely from different neighborhood structures of distinct nodes. Therefore, approximately, the "noise" of $\widetilde{\mathbf{A}}_{\mathcal{G}}^k\mathbf{X}$ is also due to distinct neighborhood structures. The result suggests that feature noise $\mathbf{n}_i$ and structural noise $\mathbf{n}_i^{(k)}, k > 0$ are indeed of different characteristics. In other words, the operator $\widetilde{\mathbf{A}}_{\mathcal{G}}^k$ effectively "replaces" feature noise with structural noise. This decoupling between feature and structural noise becomes more pronounced when the graph construction depends only weakly on the initial node features. We may verify the above numerically as follows, even for a relatively small $k = 2$. For a node v_i, we compute its feature and 2-hop structural noise $\mathbf{n}_i$ and $\mathbf{n}_i^{(2)}$. We reshuffle the index so that the components of $\mathbf{n}_i$ are ordered increasingly, while the same indexing is applied to $\mathbf{n}_i^{(2)}$. Sample examples are shown in Fig. 2. We see that $\mathbf{n}_i^{(2)}$ displays a more random behavior with the given feature index ordering. From the examples in Fig. 2, we expect that the correlation between $\mathbf{n}_i^{(2)}$ and $\mathbf{n}_i$ can be reduced due to cancellation from the random spikes. This is desirable, as discussed in Section 2.

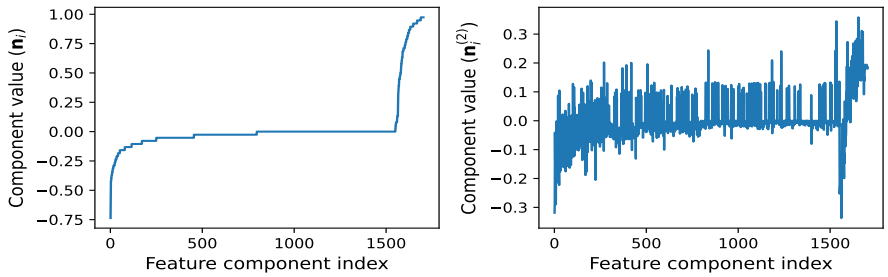


Figure 2: For a sample node v_i, the noise $\mathbf{n}_i$ (left panel) and $\mathbf{n}_i^{(2)}$ (right panel) of a random selected node v_i from the Cornell dataset. We see that $\mathbf{n}_i^{(2)}$ displays a more random behavior.

Observation 1. *Consider the empirical average correlation $E_k = \sum_{i=1}^N \langle \mathbf{n}_i, \mathbf{n}_i^{(k)} \rangle / N$ between $\{\mathbf{n}_i\}$ and $\{\mathbf{n}_i^{(k)}\}$. E_k can be decomposed as $E_k = D_k + H_k$, where D_k can be reduced as k increases, while the term H_k has zero expectation, i.e., $\mathbb{E}[H_k] = 0$.*

A detailed treatment of Observation 1 is deferred to Appendix B. In particular, Corollary 1 establishes that $D_{2l+2} \leq D_{2l}$ for all $l \geq 0$, a result that is especially relevant given the widespread use of two-layer GCNs.

To generate a "secondary view" to supplement the initial features, it suffices to consider features transformed via k-step message passing. The above theoretical discussions suggest that we may consider a *simple MLP* and a *simple GCN* [16] as the view generation encoders. With k being the number of GCN layers, a moderate choice such as $k = 2$, which conforms to common practices, is usually sufficient to generate less correlated structural noise (see evidence in Fig. 2). We provide more details on the model in the next section.

4 A Proposition: A Very Simple Model for GCL

Based on the previous theoretical developments and intuitions, we suggest a very simple model for GCL. The strategy is to use a k-layer *GCN* and an *MLP* as view-generation encoders. Consider a graph $\mathcal{G}$ with adjacency matrix $\mathbf{A}$ and node features $\mathbf{X}$, collectively denoted as $\mathcal{X} = (\mathbf{A}, \mathbf{X})$. The k-layer *GCN* captures structural features together with their inherent structural noise, producing the

view

$$\mathbf{H}^{(0)} = \mathbf{X}, \;\; \mathbf{H}^{(\ell+1)} = \sigma\left(\widetilde{\mathbf{A}}_{\mathcal{G}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}\right), \ell = 0, \ldots, k-1,$$

where $\widetilde{\mathbf{A}}_{\mathcal{G}}$ is the normalized adjacency matrix with self-loops, $\mathbf{W}^{(\ell)}$ are learnable weight matrices, and $\sigma(\cdot)$ is a nonlinear activation (e.g., ReLU). The output of the GCN after k layers,

$$\mathbf{Z}_s = \mathbf{H}^{(k)}$$

is a representation with prominent structural noise. In parallel, the *MLP* serves as an encoder that isolates feature noise, generating the feature-noise representation

$$\mathbf{Z}_f = \text{MLP}(\mathbf{X}).$$

Together, $\mathbf{Z}_s$ and $\mathbf{Z}_f$ form two complementary views used for contrastive learning. The learnable parameters of the model are the weight matrices of the GCN and the MLP, while the number of GCN layers k and MLP layers L are treated as hyperparameters. In practice, we adopt $L = 1$ for the MLP, which is a simple and efficient choice and aligns with common practice. To optimize these parameters, we adopt the standard cosmean contrastive loss $\mathcal{L}$ [17] between $\mathbf{Z}_s$ and $\mathbf{Z}_f$, i.e.,

$$\mathcal{L}(\mathbf{Z}_s, \mathbf{Z}_f) = 1 - \frac{1}{N}\sum_{i=1}^{N} \frac{\langle \mathbf{Z}_{s,i}, \mathbf{Z}_{f,i} \rangle}{\|\mathbf{Z}_{s,i}\|_2 \|\mathbf{Z}_{f,i}\|_2},$$

where $\langle \cdot, \cdot \rangle$ is the inner product, and $\|\cdot\|_2$ denotes the ℓ_2-norm. It is deemed to align the feature vectors for "amplifying class centroids" (see the first strategy in Section 2).

For downstream tasks, we compute a weighted average of the two views as $\mathbf{Z} = \beta\mathbf{Z}_s + (1-\beta)\mathbf{Z}_f$, where β is either set to 0.5 or tuned based on validation accuracy. This aggregation is effective only if the noise components of $\mathbf{Z}_s$ and $\mathbf{Z}_f$ are not strongly correlated. As in Observation 1, the structure-noise view $\mathbf{Z}_s$ and the feature-noise view $\mathbf{Z}_f$ are less correlated. Consequently, their combination allows the signal components to be reinforced while their independent noise components are partially canceled out. This highlights the importance of constructing diverse feature views that capture different sources of noise, as feature noise and structural noise exhibit inherently different characteristics.

Heterophily vs. homophily Although GCN-MLP is applicable to both homophilic and heterophilic graphs, its accuracy improvements are expected to be larger under heterophily (cf. Observation 1). In homophilic graphs, GCN already performs strongly because neighborhood aggregation typically reinforces class-consistent information while suppressing feature noise. As a result, structural and attribute signals are highly aligned, and the marginal benefit of integrating two views is reduced. However, homophily is inherently local. Even highly homophilic datasets contain boundary nodes with heterophilic neighborhoods, which are more difficult to classify. To evaluate performance on these challenging cases, we focus on nodes with heterophily ratio equal to 1 (i.e., all neighbors have different labels) and report their test accuracy. As shown in Table 1, GCN-MLP consistently outperforms GraphACL, a strong contrastive baseline, on these difficult nodes in homophilic datasets. Moreover, in homophilic settings, GCN-MLP also offers substantial gains in computational and memory efficiency (see Section 5.5) and maintains strong robustness (see Section 5.4).

In contrast, heterophilic graphs frequently connect nodes from different classes, so message passing tends to amplify structural noise while attenuating feature noise. GCN-MLP exploits this regime by explicitly decoupling these two noise sources: the MLP captures feature-space information together with feature noise, whereas the GCN captures structure-informed representations together with structural noise. Because these noise components are typically less correlated under heterophily, combining the two views reinforces informative signals and partially cancels independent noise, leading to strong performance on heterophilic benchmarks (see Section 5.2).

Table 1: Test accuracy on heterophilic nodes

	Cora	Citeseer	Pubmed
GraphACL	36.31±0.01	31.06±0.87	43.43±0.03
GCN-MLP	39.11±0.65	32.95±0.13	54.07±0.57

Table 2: Different GNNs as structural encoder.

	Cora	Pubmed	Photo
GCN-MLP	77.26±0.14	79.00± 0.03	93.41± 0.87
GAT-MLP	78.83±0.45	79.52± 0.19	92.12± 0.83
GRAND-MLP	84.49±0.14	81.11± 0.20	94.12± 0.70

Model variants: improvement for homophilic datasets We emphasize that our core contribution is the *noise-cancellation principle*, rather than a specific instantiation GCN-MLP architecture. Intuitively, GCN aggregation averages out feature noise (cf. Proposition 2), replacing it with structural noise that the MLP can isolate—provided the two noise sources are largely decoupled, as is typical outside strongly homophilic settings. Hence, any GNN with a similar averaging effect (e.g., GAT, GRAND [18]) could serve as the structural encoder; as shown in Table 2. For example, replacing GCN with GRAND, a continuous counterpart of GCN that captures additional *longer-range structural information*, improves the accuracy on Cora to 84.49%, outperforming most baselines (see Table 4). Nevertheless, GCN remains preferable when efficiency is prioritized, e.g., on Cora, GCN-MLP requires only 0.04s per training epoch, compared with 0.28s for GRAND-MLP.

Robustness Real-world graphs often contain noisy features and imperfect topology (e.g., missing edges) [19]. GCN-MLP improves robustness by contrasting two complementary views: a graph-independent MLP view, which is insensitive to edge perturbations, and a topology-aware GCN view. Their alignment strengthens useful representations while suppressing uncorrelated noise, making the model robust to both feature noise and structural perturbations, as confirmed by the black-box and white-box attack results in Section 5.4.

5 Experiments

5.1 Experimental setup

Datasets and splits We first focus on heterophilic datasets to verify that our method is particularly well-suited for weak homophily, including Wisconsin, Cornell, Texas, Actor, Crocodile, and the filtered Squirrel/Chameleon, where duplicates are removed to avoid train-test leakage [20]. We further test scalability on **large-scale** heterophilic datasets, including Amazon-ratings, Roman-empire, and Arxiv-year, and report homophilic results on Cora, Citeseer, Pubmed, Computer, Photo and **large-scale** Ogbn-Arxiv. All datasets use standard public splits with details in Appendix C.1.

Baselines We compare GCN-MLP with 15 unsupervised baselines, covering classical and recent SOTAs models: DGI [21], GMI [22], MVGRL [23], GRACE [5], CCA-SSG [24], BGRL [17], AFGRL [25], DSSL [7], SP-GCL [26], GraphACL [13], GraphECL [27], PolyGCL [9], LOHA [28], EPAGCL [8] and SDMG [10]. Detailed are given in Appendix C.2.

Evaluation protocol We evaluate representation quality mainly on **node classification**: following the standard linear evaluation protocol, we train a linear classifier on the frozen representations and report the test accuracy. We also assess GCN-MLP on the **graph classification** and **link prediction** (see Appendix D.5) to demonstrate its generalization beyond node-level settings.

Setup All methods are trained and evaluated under a unified protocol to ensure fairness and reproducibility. We randomly initialize model parameters and train the encoder with the Adam and report the mean and standard deviation over ten seeds. For fairness, all methods tune learning rate, weight decay, and hidden dimension using validation accuracy only, following standard baseline settings [13]. Missing or non-standard splits baselines [7, 9, 28, 10] are reproduced with the authors' official code.

5.2 Overall performance

Node classification results on heterophilic and homophilic datasets are reported in Tables 3 and 4. On heterophilic graphs, GCN-MLP achieves clear state-of-the-art performance, outperforming GraphACL, PolyGCL, GraphECL, and other baselines by a large margin. This supports our strategy of mitigating feature noise by combining it with representations dominated by weakly correlated structural noise. Further visualizations and discussions on the relation between performance and noise correlation are provided in Appendix D.1.

On homophilic graphs, the performance gains of GCN-MLP are less pronounced, as discussed in Section 4. Although it underperforms on Cora, it remains competitive on the other datasets. As further noted in Section 4, the proposed noise-cancellation principle is not tied to a specific backbone: replacing GCN with GRAND—a continuous variant that captures longer-range structural dependencies—yields GRAND-MLP, which outperforms most baselines on homophilic benchmarks. Nevertheless, GCN-MLP remains more cost-effective, requiring substantially less computation time and memory (Table 10).

Table 3: Node classification results(%) on heterophilic datasets. The best and the second-best result under each dataset are highlighted in **red** and **blue**, respectively.

Method	DGI	CCA-SSG	BGRL	DSSL	SP-GCL	GraphACL	PolyGCL	GraphECL	LOHA	EPAGCL	SDMG	GCN-MLP
Squirrel	40.60±0.35	**41.23±1.77**	40.13±2.16	37.69±2.07	40.11±2.20	35.51±2.03	33.07±0.94	35.89±2.62	34.46±1.69	40.28±1.59	36.82±0.77	**43.89±1.62**
Chameleon	42.57±0.71	39.46±3.10	42.55±2.35	40.95±3.35	44.49±2.59	38.59±2.81	41.79±2.45	36.11±1.15	**45.45±1.83**	35.43±1.28	41.55±0.73	**46.48±0.45**
Crocodile	51.25±0.51	56.77±0.39	53.87±0.65	62.98±0.51	61.72±0.21	**66.17±0.24**	65.95±0.59	52.52±3.01	66.09±0.69	66.14±0.62	65.38±0.37	**66.47±1.20**
Actor	28.30±0.76	27.82±0.60	28.80±0.54	28.15±0.31	28.94±0.69	30.03±0.13	34.37±0.69	**35.80±0.89**	33.69±0.73	30.02±0.91	26.74±0.13	**36.79±0.91**
Wisconsin	55.21±1.02	58.46±0.96	51.23±1.17	62.25±0.55	60.12±0.39	69.22±0.40	76.92±1.48	**79.41±0.65**	74.27±1.99	63.73±3.95	52.68±1.21	**84.90±0.20**
Cornell	45.33±6.11	52.17±1.04	50.33±2.29	53.15±1.28	52.29±1.21	59.33±1.48	44.11±1.12	**66.24±1.82**	55.35±1.32	53.78±3.72	45.59±0.67	**70.78±0.87**
Texas	58.53±2.98	59.89±0.78	52.77±1.98	62.11±1.53	59.81±1.33	71.08±0.34	72.51±1.95	**73.76±2.02**	67.75±2.70	68.92±5.95	53.60±2.67	**78.14±0.57**
Roman	63.71±0.63	67.35±0.61	68.66±0.39	71.70±0.54	70.88±0.35	**74.91±0.28**	72.97±0.25	45.05±1.57	OOM	47.11±0.87	49.20±0.51	**78.21±0.39**
Amazon	42.72±0.42	41.23±0.25	41.17±0.25	42.12±0.78	42.04±0.68	OOM	**44.29±0.43**	36.88±1.25	38.45±0.20	OOM	43.70±0.36	**45.42±0.47**
Arxiv-year	39.26±0.72	37.38±0.41	43.02±0.62	45.80±0.57	44.11±0.35	**47.21±0.39**	43.07±0.23	OOM	OOM	OOM	OOM	**46.15±0.08**

Table 4: Node classification results(%) on homophilic datasets. For each model, we show the average performance ranking across all the datasets.

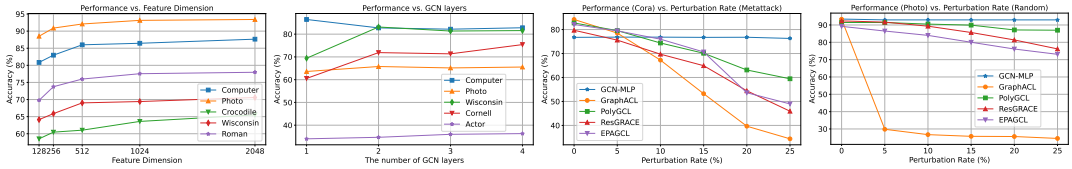
Method	DGI	GMI	MVGRL	GRACE	CCA-SSG	BGRL	AFGRL	SP-GCL	GraphACL	PolyGCL	LOHA	EPAGCL	SDMG	GCN-MLP	GRAND-MLP
Cora	82.30±0.60	82.70±0.20	82.90±0.71	76.80±0.90	84.00±0.40	82.70±0.60	82.31±0.42	83.16±0.13	**84.20±0.31**	82.74±0.14	81.22±0.17	82.14±0.89	83.60±0.60	77.26±0.14	**84.49±0.14**
Citeseer	71.80±0.70	73.01±0.30	72.61±0.70	71.72±0.62	73.10±0.30	71.10±0.80	68.70±0.30	71.96±0.42	**73.63±0.22**	71.82±0.45	71.89±0.63	71.94±0.57	**73.20±0.50**	70.12±0.44	72.98±0.52
Pubmed	76.80±0.60	80.11±0.22	79.41±0.31	79.51±1.10	81.00±0.40	79.60±0.50	79.71±0.21	79.16±0.84	**82.02±0.15**	77.31±0.27	78.09±0.29	**81.28±0.62**	80.00±0.40	79.00±0.03	81.11±0.20
Computer	83.95±0.47	82.21±0.34	87.52±0.11	86.51±0.32	88.74±0.28	89.69±0.37	89.90±0.31	89.68±0.19	89.80±0.25	86.54±0.45	79.05±0.32	76.81±0.79	**90.40±0.20**	87.65±1.10	**89.97±0.65**
Photo	91.61±0.22	90.72±0.21	91.72±0.10	92.50±0.22	93.14±0.14	92.90±0.30	93.25±0.33	92.49±0.31	93.31±0.19	91.45±0.35	86.46±0.41	93.05±0.23	**94.10±0.20**	93.41±0.88	**94.12±0.70**
Ogbn-arxiv	70.32±0.25	OOM	OOM	65.10±0.50	69.37±0.20	71.21±0.20	**71.64±0.24**	OOM	68.25±0.22	OOM	OOM	OOM	70.60±0.20	69.81±0.07	70.27±0.14
Avg. rank.	11.00	8.80	8.40	11.00	5.17	7.67	7.00	7.80	3.50	9.17	12.80	8.80	**3.17**	9.67	**3.00**

5.3 Ablation and hyperparameter analysis

To illustrate the complementary roles of the two encoders, the GCN captures structural features with their inherent noise, while the MLP isolates feature noise, together producing complementary views. Ablation results on Cora, Chameleon, Roman, and Arxiv-year (Table 5) show that both GCN-GCN and MLP-MLP underperform GCN-MLP, confirming the benefit of combining structural and feature-based views. Key hyperparameters follow existing literature or common GNN/GCL practice.

Table 5: Node classification results (%) across different datasets and design configurations.

Method	Cora	Chameleon	Roman	Arxiv-year
MLP-MLP	64.37±0.31	42.13±4.52	65.55±0.48	35.64±0.28
GCN-GCN	56.23±0.54	34.49±2.72	32.83±0.28	40.82±0.18
GCN-MLP	77.26±0.14	46.01±4.42	77.13±0.46	46.15±0.08



(a) Acc. vs. Feature dim. (b) Acc. vs GCN layers (c) Acc. vs. Metattack (d) Acc. vs Rand. attack

Figure 3: Comparison across feature dimension, GCN layers, Metattack, and Random attack

Feature dimension In the GCN-MLP model, the linear layers in both GCN and MLP expand the feature dimension, thus increasing the representation capacity and allowing the model to capture more complex patterns, as discussed in [13]. As shown in Fig. 3a, enlarging the feature dimension consistently improves performance on both homophilic and heterophilic graphs, with especially pronounced gains on the latter.

The number of GCN layers We study the effect of the GCN depth k in the GCN-MLP (Fig. 3b). On heterophilic graphs, larger k improves performance by lowering the cosine similarity between feature and structural noise (Observation 1), making the two views more complementary. On homophilic graphs (e.g., Computer), neighborhood aggregation already reinforces class-consistent information and suppresses noise, so the structural and feature views are more strongly correlated and their combination brings limited additional benefit. Most improvements occur from $k = 1$ to $k = 2$, while deeper models give diminishing returns. Thus, $k = 2$, consistent with common practice [16], offers a good trade-off between performance and complexity.

Augmentation & Negative sampling techniques To access whether GCN-MLP benefits from additional training tricks, we test two common operations: data augmentation (e.g., edge removal and feature masking) and InfoNCE negative sampling (as in GRACE) on both homophilic and heterophilic datasets. As shown in Table 6, neither brings clear improvement over the original model, suggesting that GCN-MLP is already near-optimal without augmentation or negatives, This supports an Occam's Razor design: simple, lightweight, and efficient while retaining strong performance.

Table 6: Acc. w.r.t. additional techniques

	Crocodile	Roman	Pubmed	Photo
GCN-MLP	66.47±1.20	78.21±0.39	79.00±0.03	93.41±0.87
GCN-MLP(+ Aug.)	65.56±1.01	78.25±0.49	79.05±0.22	93.48±0.82
GCN-MLP(+ Neg.)	65.71±0.95	78.16±0.45	78.67±0.13	93.40±0.96
GCN-MLP(+ Aug. & Neg.)	65.71±0.55	78.20±0.49	78.58±0.04	93.46±0.97

Table 7: Sensitiveness of GCN-MLP w.r.t. β.

	Best accuracy(β)	$\beta = 0.5$
Crocodile	66.47±1.20 (0.3)	65.44±1.20
Roman	78.21±0.39 (0.4)	77.98±0.37
Citeseer	71.72±0.13 (0.8)	70.12±0.44
Photo	93.41±0.87 (0.7)	92.05±1.15

The selection of β and sensitivity analysis The combination weight β balances structural and feature views and can be adjusted to graph homophily. Homophilic graphs generally favor larger β, as neighborhood aggregation reinforces class-consistent information, whereas heterophilic graphs may prefer smaller β to suppress structural noise. In practice, we either fix $\beta = 0.5$ by default or tune it by validation with step size 0.1. As shown in Table 7, $\beta = 0.5$ performs close to the tuned optimum across datasets, indicating that GCN-MLP is robust to β and requires little hyperparameter tuning.

5.4 Robustness study

We evaluate the claimed robustness of GCN-MLP through node classification under both black-box and white-box adversarial attacks on standard benchmarks, including homophilic (Photo) and heterophilic (Actor, Wisconsin, Texas) graphs. GCN-MLP is compared against eight baselines: a robust supervised method (FROND [29]), three robust GCL methods (GCL-Jac [30], Ariel [31], Res-GRACE [32]), and five state-of-the-art GCL models (GraphACL, PolyGCL, LOHA, EPAGCL, SDMG). Additional results on more datasets are provided in Appendix D.4.

Attacks methods We consider four black-box topology attacks in the evasion setting: Random, PRBCD [33], Nettack [34], and Metattack [35]. We also evaluate two white-box attacks, PGD [36] and PRBCD, which jointly perturb graph structure and node features. All models are trained on clean graphs, with perturbations applied only at inference. Attack details are provided in Appendix C.3.

Robust results We evaluate GCN-MLP under perturbation rates from 0% to 25% using Metattack on Cora (Fig. 3c) and Random attack on Photo (Fig. 3d). Although GCN-MLP is weaker on clean Cora, it becomes more robust as perturbations increase, outperforming strong baselines such as GraphACL at higher attack rates. This robustness comes from its dual-view design: the MLP provides a structure-independent feature view resilient to edge perturbations, while the GCN provides a structure-aware view. Their contrastive alignment helps preserve consistent signals and suppress adversarial noise when the graph structure is corrupted. Results in Tables 8 and 9 further confirm its strong robustness, especially under white-box attacks. More robust results are in Appendix D.4.

Table 8: Black-box attack robust accuracy results(%) on graph evasion attack (under 20% perturbation) for node classification.

Dataset	Attack	FROND	GCL-Jac	Ariel	Res-GRACE	GraphACL	PolyGCL	LOHA	EPAGCL	SDMG	GCN-MLP
Photo	*clean*	92.93±0.46	91.46±0.50	85.75±1.21	92.23±1.22	93.31±0.19	91.45±0.35	86.46±0.41	93.05±0.23	**94.10±0.20**	**93.41±0.88**
	Random	89.90±1.21	86.40±0.74	80.62±1.53	87.79±1.93	26.61±0.05	**90.17±0.99**	85.83±1.12	84.08±1.50	89.90±0.78	**92.94±0.58**
	PRBCD	88.58±1.05	85.24±1.30	80.58±1.62	85.39±4.19	29.13±0.95	**89.65±0.39**	86.35±1.07	80.60±2.72	89.42±0.96	**92.84±0.28**
	Metattack	89.61±1.13	86.20±1.06	82.76±1.11	85.46±1.56	28.42±0.74	**91.06±1.36**	86.56±0.89	85.65±0.56	90.78±0.99	**91.14±0.68**
	Nettack	91.17±1.35	90.50±0.63	85.28±0.91	91.51±1.40	32.84±0.25	**91.29±1.15**	87.40±0.89	89.59±1.05	90.29±0.56	**92.34±0.52**
Wisconsin	*clean*	67.84±3.84	43.53±6.19	56.08±4.31	52.35±7.18	69.22±0.40	**76.08±3.33**	76.05±6.08	63.73±3.95	52.68±1.21	**85.10±2.35**
	Random	69.61±4.49	44.71±6.43	51.18±5.44	51.76±6.27	51.56±5.63	75.23±3.13	**76.47±4.12**	59.02±4.59	51.18±0.98	**85.29±1.81**
	PRBCD	67.65±5.28	44.71±6.72	55.88±4.41	51.37±6.67	52.55±5.13	74.60±3.14	**75.29±4.12**	60.39±6.61	50.98±0.78	**84.90±2.33**
	Metattack	64.51±5.98	43.53±4.09	50.98±4.64	50.59±6.06	52.15±5.08	**76.67±3.92**	74.71±4.31	60.39±4.79	51.67±1.47	**84.90±1.76**
	Nettack	70.78±6.17	44.71±5.32	55.29±5.02	50.00±5.70	53.73±5.16	**77.65±3.92**	75.49±3.73	59.02±2.97	51.57±1.57	**85.10±2.00**
Actor	*clean*	**35.08±1.08**	29.25±1.21	24.36±1.11	30.72±0.72	30.03±0.13	34.37±0.69	33.69±0.73	30.02±0.91	26.74±0.13	**36.56±0.93**
	Random	**35.15±0.78**	27.59±1.12	25.64±1.02	30.16±1.09	28.36±1.95	25.41±0.72	34.19±0.59	28.92±1.03	27.09±0.68	**36.19±0.77**
	PRBCD	**35.04±0.90**	27.76±1.66	24.95±0.89	30.48±1.28	28.37±1.95	27.21±0.64	26.23±0.79	28.66±2.01	26.79±0.82	**36.47±1.05**
	Metattack	**32.34±7.10**	28.00±1.10	25.54±0.75	30.34±1.04	28.45±1.26	28.29±0.42	26.97±0.65	29.65±1.12	26.78±0.91	**36.56±1.12**
	Nettack	**34.97±0.88**	28.87±0.73	25.51±0.95	30.86±0.96	28.60±1.20	25.96±0.86	27.20±0.74	30.05±0.81	26.72±0.79	**36.14±0.67**

Table 9: White-box attack robust accuracy (%) under 15% perturbation for node classification.

Dataset	Attack	FROND	Res-GRACE	GraphACL	PolyGCL	LOHA	EPAGCL	SDMG	GCN-MLP
Photo	*clean*	92.03±1.27	92.50±0.17	93.31±0.19	91.45±0.35	86.46±0.41	93.05±0.23	**94.10±0.20**	**93.41±0.88**
	PGD	14.18±5.16	45.46±2.05	30.94±3.62	22.93±2.73	**59.35±0.43**	5.02±2.85	11.98±0.81	**90.56±0.54**
	PRBCD	15.08±7.52	40.72±3.70	28.56±1.16	15.86±1.99	**52.82±3.45**	9.91±1.73	8.71±6.22	**75.42±0.69**
Texas	*clean*	**74.86±3.21**	54.59±5.51	71.08±0.34	72.51±1.95	67.75±2.70	68.92±4.05	53.60±2.67	**78.38±4.68**
	PGD	**69.46±7.16**	28.85± 8.55	21.62±7.20	57.84±14.05	47.30±16.45	13.78±6.78	30.49±4.32	**75.68±6.28**
	PRBCD	**65.14±3.91**	26.49±11.22	23.78±8.99	51.35±10.26	32.70±10.50	14.32±6.29	19.43±4.39	**67.03±5.10**

Table 10: Comparison of training time (s), inference time (s), and storage (MiB) across different datasets.

Method	Training Time (s)				Inference Time (s)				Storage (MiB)			
	Cora	Wisconsin	Roman	Arxiv-year	Cora	Wisconsin	Roman	Arxiv-year	Cora	Wisconsin	Roman	Arxiv-year
GraphACL	0.06	0.08	100.32	5403.97	0.004	0.01	0.15	1.06	932	1670	14008	53960
PolyGCL	0.34	0.23	OOM	OOM	7.46	9.73	OOM	OOM	4098	894	OOM	OOM
GraphECL	0.02	0.27	60.58	OOM	0.50	0.04	0.04	OOM	112	336	2306	OOM
EPAGCL	0.13	0.12	0.97	OOM	2.81	2.63	3.14	OOM	649	329	22127	OOM
SDMG	0.06	0.54	0.36	OOM	1.22	0.28	0.55	OOM	2888	824	3676	OOM
GCN-MLP	0.04	0.007	0.097	3.96	0.02	0.001	0.031	1.89	2396	588	5850	44368

5.5 Complexity analysis

The training time complexity of GCN-MLP consists of three parts: solving the GCN encoder, the MLP encoder, and computing the contrastive loss. For a graph with N nodes and $|\mathcal{E}|$ edges, each GCN layer requires $\mathcal{O}(|\mathcal{E}|h + Nh^2)$ operations, where h is the hidden feature dimension, leading to a total of $\mathcal{O}(k(|\mathcal{E}|h + Nh^2))$ for k layers. The MLP encoder adds $\mathcal{O}(LNh^2)$ for L layers. The contrastive loss further requires pairwise similarity computations $\mathcal{O}(N)$, resulting in a total training complexity of $\mathcal{O}(k(|\mathcal{E}|h + LNh^2) + N)$. Training/inference time and memory comparisons with baselines (e.g., GraphACL, PolyGCL, GraphECL, EPAGCL, and SDMG) are shown in Table 10. GCN-MLP consistently achieves much lower training, inference costs and avoids out-of-memory issues, even on large-scale graphs (e.g., Arxiv-year).

6 Conclusion

In conclusion, we revisit the foundations of graph contrastive learning and reveal that original node features and graph structure naturally serve as two complementary views, eliminating the need for augmentation, negative sampling, or complex encoders. Based on this insight, our embarrassingly simple GCN-MLP framework achieves state-of-the-art results on both heterophilic and homophilic benchmarks with minimal computational and memory overhead. Theoretical analysis and extensive experiments validate the effectiveness and generality of our approach, demonstrating that principled simplicity can outperform architectural complexity in graph contrastive learning.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, and J. Clark, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[2] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. Guo, and M. G. Azar, "Bootstrap your own latent-a new approach to self-supervised learning," in *Advances Neural Inf. Process. Syst.*, 2020.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[4] Y. Zhang, Y. Chen, Z. Song, and I. King, "Contrastive cross-scale graph knowledge synergy," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2023, p. 3422–3433.

[5] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep graph contrastive representation learning," *arXiv preprint arXiv:2006.04131*, 2020.

[6] ——, "Graph contrastive learning with adaptive augmentation," in *Proc. Web Conf.*, 2021, pp. 2069–2080.

[7] T. Xiao, Z. Chen, Z. Guo, Z. Zhuang, and S. Wang, "Decoupled self-supervised learning for graphs," in *Advances Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 620–634.

[8] Y. Xu, S. Huang, H. Zhang, and X. Li, "Why does dropping edges usually outperform adding edges in graph contrastive learning?" in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 21 824–21 832.

[9] J. Chen, R. Lei, and Z. Wei, "PolyGCL: Graph contrastive learning via learnable spectral polynomial filters," in *Proc. Int. Conf. Learn. Representations*, 2024.

[10] J. Zhu, L. He, C. Gao, D. Hou, Z. Su, P. S. Yu, J. Kurths, and F. Hellmann, "SDMG: Smoothing your diffusion models for powerful graph representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2025.

[11] F. Ji, Y. Zhao, K. Zhao, H. Meng, J. Yang, and W. P. Tay, "Rethinking graph neural networks from a geometric perspective of node features," in *Proc. Int. Conf. Learn. Representations*, 2025.

[12] J. Linkerhägner, C. Shi, and I. Dokmanić, "Joint graph rewiring and feature denoising via spectral resonance," in *Proc. Int. Conf. Learn. Representations*, 2025.

[13] T. Xiao, H. Zhu, Z. Chen, and S. Wang, "Simple and asymmetric graph contrastive learning without augmentations," in *Advances Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 16 129–16 152.

[14] Y. Zhao, F. Ji, K. Zhao, X. Li, Q. Kang, W. Liang, Y. Alkhatib, X. Jian, and W. P. Tay, "Simple graph contrastive learning via fractional-order neural diffusion networks," *arXiv preprint arXiv:2504.16748*, 2025.

[15] Y. Zhao, F. Ji, J. Dai, K. Jiang, J. Ma, K. Zhao, and W. P. Tay, "Adaptive multi-view graph contrastive learning via fractional continuous dynamics," in *Advances Neural Inf. Process. Syst.*, 2026.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[17] S. Thakoor, C. Tallec, M. G. Azar, M. Azabou, E. L. Dyer, R. Munos, P. Veličković, and M. Valko, "Large-scale representation learning on graphs via bootstrapping," in *Proc. Int. Conf. Learn. Representations*, 2022.

[18] B. P. Chamberlain, J. Rowbottom, M. Goronova, S. Webb, E. Rossi, and M. M. Bronstein, "GRAND: Graph neural diffusion," in *Proc. Int. Conf. Mach. Learn.*, 2021.

[19] S. Lee, F. Ji, W. P. Tay, and K. Xia, "Graph neural networks with a distribution of parametrized graphs," in *Proc. Int. Conf. Mach. Learn.*, 2024.

[20] O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, and L. Prokhorenkova, "A critical look at evaluation of gnns under heterophily: Are we really making progress?" in *Proc. Int. Conf. Learn. Representations*, 2023.

[21] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Representations*, 2019.

[22] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, "Graph representation learning via graphical mutual information maximization," in *Proc. Web Conf.*, 2020.

[23] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn.*, Jul 2020, pp. 4116–4126.

[24] H. Zhang, Q. Wu, J. Yan, D. Wipf, and P. S. Yu, "From canonical correlation analysis to self-supervised graph neural networks," in *Advances Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 76–89.

[25] N. Lee, J. Lee, and C. Park, "Augmentation-free self-supervised learning on graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2022, p. 7372–7380.

[26] H. Wang, J. Zhang, Q. Zhu, W. Huang, K. Kawaguchi, and X. Xiao, "Single-pass contrastive learning can work for both homophilic and heterophilic graph," *Trans. Mach. Learn. Res.*, 2023.

[27] T. Xiao, H. Zhu, Z. Zhang, Z. Guo, C. C. Aggarwal, S. Wang, and V. G. Honavar, "Efficient contrastive learning for fast and accurate inference on graphs," in *Proc. Int. Conf. Mach. Learn.*, Jul 2024, pp. 54 363–54 381.

[28] Z. Zou, Y. Jiang, L. Shen, J. Liu, and X. Liu, "Loha: Direct graph spectral contrastive learning between low-pass and high-pass views," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, 2025, pp. 13 492–13 500.

[29] Q. Kang, K. Zhao, Q. Ding, F. Ji, X. Li, W. Liang, Y. Song, and W. P. Tay, "Unleashing the potential of fractional calculus in graph neural networks with FROND," in *Proc. Int. Conf. Learn. Representations*, Vienna, Austria, 2024.

[30] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin, "Topology attack and defense for graph neural networks: An optimization perspective," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020.

[31] S. Feng, B. Jing, Y. Zhu, and H. Tong, "Adversarial graph contrastive learning with information regularization," in *Proc. Web Conf.*, 2022.

[32] M. Lin, T. Xiao, E. Dai, X. Zhang, and S. Wang, "Certifiably robust graph contrastive learning," in *Advances Neural Inf. Process. Syst.*, 2024.

[33] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2018, pp. 2847–2856.

[34] S. Geisler, T. Schmidt, H. Sirin, D. Zügner, A. Bojchevski, and S. Günnemann, "Robustness of graph neural networks at scale," in *Advances Neural Inf. Process. Syst.*, 2021, pp. 7637–7649.

[35] D. Zügner and S. Günnemann, "Adversarial attacks on graph neural networks via meta learning," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[37] J. Chen and G. Kou, "Attribute and structure preserving graph contrastive learning," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 6, pp. 7024–7032, Jun. 2023.

[38] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Advances Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 5812–5823.

[39] J. Chen, G. Zhu, Y. Qi, C. Yuan, and Y. Huang, "Towards self-supervised learning on graphs with heterophily," in *Proc. ACM Int. Conf. Inf. & Knowledge Management*, 2022, p. 201–211.

[40] N. Liu, X. Wang, D. Bo, C. Shi, and J. Pei, "Revisiting graph contrastive learning from the perspective of graph spectrum," in *Advances Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 2972–2983.

[41] H. Zhang, Q. Wu, Y. Wang, S. Zhang, J. Yan, and P. S. Yu., "Localized contrastive learning on graphs," *arXiv preprint arXiv:2212.04604*, 2022.

[42] Y. Li, H. Zhang, and Y. Yuan, "Edge contrastive learning: An augmentation-free graph contrastive learning model," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 18 575–18 583.

[43] Z. Wang, Z. Zhang, C. Zhang, and Y. Ye, "Training mlps on graphs without supervision," in *Proc. ACM Int. Conf. Web Search Data Min.*, 2025.

[44] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.

[45] J. Kohler and A. Lucchi, "Sub-sampled cubic regularization for non-convex optimization," in *International Conference on Machine Learning*, 2017.

[46] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[47] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," *J. Complex Netw.*, 2021.

[48] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2020.

[49] O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, and L. Prokhorenkova, "A critical look at evaluation of GNNs under heterophily: Are we really making progress?" in *Proc. Int. Conf. Learn. Representations*, 2023.

[50] D. Lim, F. Hohne, X. Li, S. L. Huang, V. Gupta, O. Bhalerao, and S. N. Lim, "Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods," in *Advances Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20 887–20 902.

[51] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. Int. ACM SIGIR Conf. Research and Develop. Inform. Retrieval*, 2015, p. 43–52.

[52] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.

[53] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *Proc. Int. Conf. Learn. Representations*, 2019.

[54] Q. Ji, J. Li, J. Hu, R. Wang, C. Zheng, and F. Xu, "Rethinking dimensional rationale in graph contrastive learning from causal perspective," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, pp. 12 810–12 820.

[55] W. Shiao, Z. Guo, T. Zhao, E. E. Papalexakis, Y. Liu, and N. Shah, "Link prediction with non-contrastive learning," in *Proc. Int. Conf. Learn. Representations*, 2023.

[56] Y. Hao, X. Cao, Y. Fang, X. Xie, and S. Wang, "Inductive link prediction for nodes having only attribute information," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 1209–1215.

A Related works

A.1 Graph contrastive learning without augmentation

Deep Graph Infomax (DGI) [21] is a seminal framework in graph contrastive learning, which maximizes mutual information (MI) between local node features and a global graph representation. It aggregates node features into a global embedding using a readout function, and employs a discriminator to distinguish positive samples from the original graph against negative samples generated by shuffling node features. This corruption serves as an augmentation, boosting robustness and generalization. Contrastive Multi-view Representation Learning (MVGRL) [23] extends this idea by leveraging multiple graph views generated through different graph diffusion processes. Its discriminator contrasts node-level and graph-level embeddings across views, leading to richer representations. Cross-Scale Contrastive Graph Knowledge Synergy (CGKS) [4] further advances this line by constructing a graph pyramid of coarse-grained views and employing a joint optimization strategy with pairwise contrastive loss to transfer knowledge across scales.

GRACE [5] adopts a different strategy by producing two graph views via edge removal and node feature masking, then maximizing agreement between their node embeddings. It further enhances contrastive learning with both inter-view and intra-view negative pairs. GCA [6] improves upon GRACE by designing adaptive augmentations guided by topological and semantic priors. Moving beyond dual views, ASP [37] introduces three complementary perspectives, i.e., original, attribute, and global, into a joint contrastive learning framework, strengthening representation quality across these perspectives.

GraphCL [38] systematizes augmentation strategies tailored to graph data. To handle non-homophilous graphs, DSSL [7] and HGRL [39] exploit global and high-order information. While HGCL relies on augmentations, DSSL assumes an underlying graph generation process, which may not align with real-world scenarios. Despite these advances, augmentation-based methods have notable limitations: their performance is highly sensitive to the chosen augmentations, with no universally optimal strategy. In addition, they tend to bias the encoder toward low-frequency components, while overlooking high-frequency information that is essential for learning on heterophilic graphs [40]. More recently, EPAGCL [8] combines edge addition and deletion, generating augmented views by adding or dropping edges according to weights derived from the Error Passing Rate (EPR).

To overcome the drawbacks of augmentation-based methods, augmentation-free approaches have been proposed. Graphical Mutual Information (GMI) [22] directly estimates MI between input features and representations of nodes and edges, eliminating the need for data augmentation. L-GCL [41] also avoids augmentations but focuses primarily on homophilic graphs. SP-GCL [26] overcomes this by capturing both low- and high-frequency components, making it effective for heterophilic structures. GraphACL [13] further removes reliance on both augmentations and homophily assumptions, achieving robust performance across varying graph types. More recent methods adopt spectral strategies to replace augmentation entirely. PolyGCL [9] employs learnable polynomial filters to construct spectral views with varying frequency responses. LOHA [28] directly contrasts natural low- and high-pass components in the spectral domain to facilitate contrastive learning. Similarly, AFECL [42] introduces an edge-centric contrastive framework that operates without any form of augmentation. SDMG [10] employs two dedicated low-frequency encoders to extract global signals, promoting a diffusion-based self-supervised learning scheme.

Although SimMLP [43] and GraphECL [27] also employ GCN and MLP as dual encoders, our GCN-MLP framework differs fundamentally in both motivation and contrastive formulation. SimMLP and GraphECL distill knowledge from a teacher GNN into a student MLP trained solely on node features, with the aim of integrating rich structural information into the MLP. Only the MLP outputs are used for downstream tasks, with the primary goal of accelerating inference by replacing GNN computation. In contrast, GCN-MLP is not a distillation model but is grounded in a new principle: feature noise and structural noise are weakly correlated, and their contrastive interaction leads to stronger noise cancellation. Therefore, our design goal is to construct two views whose noise components are as uncorrelated as possible. The GCN-MLP architecture is a simple yet effective instantiation of this principle, where the GCN encodes structural information with its associated structural noise, and the MLP isolates feature noise. This naturally facilitates noise cancellation through contrastive learning and linear combination, resulting in cleaner and more discriminative features for node classification, particularly on challenging heterophilic graphs.

A.2 Graph contrastive learning without negative sample pairs

Building on the success of BYOL for image data, BGRL [17] eliminates the need for negative samples in graph contrastive learning. It generates two graph augmentations through random node feature masking and edge masking, using an online encoder and a target encoder. The objective is to maximize the cosine similarity between the online encoder's prediction and the target encoder's embedding. To prevent mode collapse and ensure stable training, a stop-gradient operation is applied to the target encoder.

Augmentation-Free Graph Representation Learning (AFGRL) [25] addresses the limitations of augmentation-dependent methods like BGRL and GCA [6], where representation quality heavily depends on the choice of augmentation schemes. Building on the BGRL framework, AFGRL eliminates the need for augmentations by generating positive samples directly from the original graph for each node. This approach captures both local structural information and global semantics. However, it introduces higher computational costs.

Inspired by Canonical Correlation Analysis (CCA) methods [44], CCA-SSG [24] introduces an unsupervised learning framework for graphs without relying on negative sample pairs. It maximizes the correlation between two augmented views of the same input while decorrelating the feature dimensions within a single view's representation.

These advancements highlight promising alternatives to traditional graph contrastive learning methods. Employing augmentation-free frameworks or innovative masking strategies mitigates challenges associated with negative sample selection and augmentation dependency, offering robust solutions for graph representation learning.

B Discussions and proofs

Proof of Proposition 1. We have the following simple observation: let $\mathbf{v}_1, \mathbf{v}_2$ and $0 < \beta < 1$. Then $\|\beta\mathbf{v}_1 + (1-\beta)\mathbf{v}_2\|$ is a non-decreasing function of the cosine similarity between $\mathbf{v}_1$ and $\mathbf{v}_2$, while keeping their norms fixed. This follows from laws of cosines: $\|\beta\mathbf{v}_1 + (1-\beta)\mathbf{v}_2\|^2 = \beta^2\|\mathbf{v}_1\|^2 + (1-\beta)^2\|\mathbf{v}_2\|^2 + 2\beta(1-\beta)\|\mathbf{v}_1\|\|\mathbf{v}_2\|\cos(\alpha)$, where α is the angle between $\mathbf{v}_1, \mathbf{v}_2$. Therefore, $\|\beta\mathbf{v}_1 + (1-\beta)\mathbf{v}_2\|$ is non-decreasing in $\cos(\alpha)$.

The first part of Proposition 1 follows from letting $\mathbf{v}_1 = \mathbf{z}_{1,c}$ and $\mathbf{v}_2 = \mathbf{z}_{2,c}$. For the second part of Proposition 1, notice that $\mathbf{n}_1 = \beta\mathbf{n}'_{1,1} + (1-\beta)\mathbf{n}'_{2,1}$. Hence, it suffices to apply the above observation with $\mathbf{v}_1 = \mathbf{n}'_{1,1}$ and $\mathbf{v}_2 = \mathbf{n}'_{2,1}$. □

We next formalize Proposition 2 and give the proof. We first make some assumptions regarding the feature model. For each class c, we assume that $\mathbf{x}_i$ are i.i.d. samples according to γ_c.[3] Recall that the mean of γ_c is $\mathbf{x}_c$, we further assume the variance is bounded by a constant $\sigma^2 < \infty$, which is independent of c.

Consider a fixed node v and $k \geq 1$ as the hop number. Let N_v^k be the k-hop neighbors of v, and $N_{v,c}^k$ be the set of nodes in N_v^k whose label is c. For each $v_i \in N_v$, let w_i be the number of distinct k-hop paths from v_i to v. Denote $W_v = \sum_{v_i \in N_v^k} a_i$ and $W_{v,c} = \sum_{v_i \in N_{v,c}^k} a_i$.

To state the formal version of Proposition 2, we introduce two quantities:

$$b = \max_{c \in C} \max_{v_i \in N_{v,c}^k} \frac{a_i |N_{v,c}^k|}{W_{v,c}}; \text{ and } N_k = \max_{c \in C} |N_{v,c}^k| \Big(\frac{W_{v,c}}{W_v}\Big)^{-1} = |N_v^k| \max_{c \in C} \frac{|N_{v,c}^k|}{|N_v^k|} \Big(\frac{W_{v,c}}{W_v}\Big)^{-1}.$$

Notice b and N_k depend on how balanced a_i's are for distinct $v_i \in N_v^k$. To give a feeling of their physical meanings, if a_i's are almost balanced, then $b \sim 1$ and $N_k \sim |N_v^k|$. We have the following specific result with the above notations:

Proposition 3. *Let $\mathbf{y}_v$ be the feature vector of v in the difference matrix $\widetilde{\mathbf{A}}_{\mathcal{G}}^k \mathbf{X} - \widetilde{\mathbf{A}}_{\mathcal{G}}^k \overline{\mathbf{X}}$. Then for any $\epsilon > 0$ that is sufficiently small, we have*

$$\mathbb{P}(\|\mathbf{y}_v\| \leq |C|\epsilon) \geq \big(1 - \exp(-\frac{N_k \epsilon}{8b^2\sigma^2} + \frac{1}{4})\big)^{|C|}.$$

[3] More generally, we only need $\mathbf{x}_i$ to be independent for Proposition 3 to hold, which is required for the vector Bernstein inequality. Here, the identical distribution is emphasized for nodes with a common label c.

Proof. By the same argument in the proof of [11, Theorem 1(a)] using the Cauchy-Schwarz inequality, we have the following bound:

$$\|\mathbf{y}_v\|^2 \leq \sum_{c\in C} \frac{W_{v,c}}{W_v} \left\| \mathbf{x}_c - \frac{1}{W_{v,c}} \sum_{v_i \in N^k_{v,c}} a_i \mathbf{x}_i \right\|^2, \tag{3}$$

where $\mathbf{x}_i$ is the feature of v_i in the feature matrix $\mathbf{X}$.

The difference term can be equivalently expressed as follows:

$$\mathbf{x}_c - \frac{1}{W_{v,c}} \sum_{v_i \in N^k_{v,c}} a_i \mathbf{x}_i = \frac{1}{|N^k_{v,c}|} \sum_{v_i \in N^k_{v,c}} \frac{a_i |N^k_{v,c}|}{W_{v,c}} (\mathbf{x}_c - \mathbf{x}_i).$$

For each v_i above, consider the random variable $\frac{a_i|N^k_{v,c}|}{W_{v,c}}(\mathbf{x}_c - \mathbf{x}_i)$. It has a zero mean, and the variance is bounded by $(\frac{a_i|N^k_{v,c}|}{W_{v,c}})^2\sigma^2 \leq b^2\sigma^2 < \infty$.

Therefore, by the vector Bernstein inequality [45, Lemma 18], we have

$$\mathbb{P}\Big(\|\mathbf{x}_c - \frac{1}{W_{v,c}} \sum_{v_i \in N^k_{v,c}} a_i \mathbf{x}_i\| \geq \epsilon'_c\Big) \leq \exp(-\frac{|N^k_{v,c}|\epsilon'^2_c}{8b^2\sigma^2} + \frac{1}{4}),$$

for any ϵ'_c sufficiently small.

Let $\epsilon'_c = \big(\epsilon \frac{W_v}{W_{v,c}}\big)^{1/2}$. The above inequality implies that

$$\begin{aligned}
&\mathbb{P}\left(\|\mathbf{x}_c - \frac{1}{W_{v,c}} \sum_{v_i \in N^k_{v,c}} a_i \mathbf{x}_i\|^2 \leq \epsilon \frac{W_v}{W_{v,c}} \right) \\
\geq& 1 - \exp(-\frac{\epsilon |N^k_{v,c}| \cdot W_v/W_{v,c}}{8b^2\sigma^2} + \frac{1}{4}) \\
\geq& 1 - \exp(-\frac{N_k \epsilon}{8b^2\sigma^2} + \frac{1}{4}).
\end{aligned}$$

Combining this result with (3), we obtain the stated lower bound of $\mathbb{P}(\|\mathbf{y}_v\| \leq |C|\epsilon)$. □

Intuitively, N_k should grow as the k-hop neighborhood of the node v, which increases as k increases. Accordingly, the lower bound in the result also increases. Therefore, correspondingly, the "error term" $\mathbf{y}_v$ has a large chance of being bounded by $|C|\epsilon$, which explains Proposition 2.

We now discuss Observation 1. From the above proof, we notice that the term $\|\mathbf{v}_1\|\|\mathbf{v}_2\|\cos(\alpha)$ (in the law of cosines) is essentially the inner product $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle$, which plays the key role in the analysis of $\|\beta\mathbf{v}_1 + (1-\beta)\mathbf{v}_2\|^2$. Therefore, for the rest of the appendix, we use $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle$ to quantify the correlation between $\mathbf{v}_1$ and $\mathbf{v}_2$.

Recall that the (asymmetric) normalized Laplacian matrix $\widetilde{\mathbf{L}}_{\mathcal{G}}$ is defined as $I_N - \widetilde{\mathbf{A}}_{\mathcal{G}}$, where I_N is the identity matrix. It is conjugate to the symmetric normalized Laplacian matrix, and hence it admits an orthogonal eigenbasis with the same eigenvalues as its symmetric counterpart. Specifically, $\widetilde{\mathbf{L}}_{\mathcal{G}} = \mathbf{W}_{\mathcal{G}} \mathbf{\Lambda}_{\mathcal{G}} \mathbf{W}^{\intercal}_{\mathcal{G}}$, where columns $\mathbf{w}_i, i \leq N$ of $\mathbf{W}_{\mathcal{G}}$ are eigenvectors, and their associated eigenvalues are $\lambda_i, 1 \leq N$. They are ordered increasingly and form the diagonal of $\mathbf{\Lambda}_{\mathcal{G}}$.

For the j-th feature component, let $\mathbf{m}_j$ be the column vector whose i-th entry is the j-th component of $\mathbf{n}_i$, the feature noise of node v_i. Consider $\widetilde{\mathbf{L}}_{\mathcal{G}}$ as the graph shift operator [46]. Then the i-th Fourier coefficient $\widehat{\mathbf{m}}_j(i)$ of $\mathbf{m}_j$ is the number $\langle \mathbf{w}_i, \mathbf{m}_j \rangle$. According to the general principle of graph signal processing, if $\mathbf{m}_j$ is smooth, then $\widehat{\mathbf{m}}_j(i)$ is relatively small for large i and relatively large for small i.

Theorem 1. *Let the empirical average correlation between the feature noise and structural noise be*

$$E_k = \frac{1}{N} \sum_{1 \leq i \leq N} \langle \mathbf{n}_i, \mathbf{n}_i^{(k)} \rangle.$$

There is a decomposition $E_k = D_k + H_k$ such that the following holds:

(a)

$$D_k = \frac{1}{N} \sum_{1 \le i \le N} (1 - \lambda_i)^k \sum_{1 \le j \le N} \widehat{\mathbf{m}}_j(i)^2,$$

where $\{\lambda_i : i = 1, \ldots, N\}$ *are the eigenvalues of the normalized Laplacian* $\tilde{\mathbf{L}}_{\mathcal{G}}$ *and* $\{\widehat{\mathbf{m}}_j(i) : i, j = 1, \ldots, N\}$ *are the Fourier coefficients of the feature noise matrix.*

(b) The term H_k *takes the form*

$$H_k = \frac{1}{N} \sum_{1 \le i \le N} \langle \mathbf{n}_i, \mathbf{g}_i \rangle,$$

such that $\mathbf{g}_i$ *depends only on the graph topology and class centroids. Furthermore,* $\mathbb{E}[H_k] = 0$.

Proof. Let $\mathbf{N}$ be the matrix whose i-th row is $\mathbf{n}_i$, denoted by $(\mathbf{N})_i$. Recall that $\mathbf{n}_i = \mathbf{x}_i - \mathbf{x}_c$, where c is the class label of v_i. We re-express $\mathbf{n}_i^{(k)}$ in (1) as

$$\begin{aligned} \mathbf{n}_i^{(k)} &= (\widetilde{\mathbf{A}}_{\mathcal{G}}^k \mathbf{X})_i - (\widetilde{\mathbf{A}}_{\mathcal{G}}^k \overline{\mathbf{X}})_i + (\widetilde{\mathbf{A}}_{\mathcal{G}}^k \overline{\mathbf{X}})_i - \mathbf{M}_i \\ &= (\widetilde{\mathbf{A}}_{\mathcal{G}}^k \mathbf{N})_i + \mathbf{g}_i, \end{aligned} \tag{4}$$

where $\mathbf{g}_i$ is the i-th row of $\widetilde{\mathbf{A}}_{\mathcal{G}}^k \overline{\mathbf{X}} - \mathbf{M}$. Therefore, we have

$$E_k = \frac{1}{N} \sum_{1 \le i \le N} \langle \mathbf{n}_i, (\widetilde{\mathbf{A}}_{\mathcal{G}}^k \mathbf{N})_i \rangle + \frac{1}{N} \sum_{1 \le i \le N} \langle \mathbf{n}_i, \mathbf{g}_i \rangle.$$

Therefore, we have $E_k = D_k + H_k$, with the following respective expressions:

$$D_k = \frac{1}{N} \sum_{1 \le i \le N} \langle \mathbf{n}_i, (\widetilde{\mathbf{A}}_{\mathcal{G}}^k \mathbf{N})_i \rangle, \text{ and } H_k = \frac{1}{N} \sum_{1 \le i \le N} \langle \mathbf{n}_i, \mathbf{g}_i \rangle.$$

It suffices to show that they have the stated properties in (a) and (b).

For (a), we may express

$$N D_k = \mathrm{Tr}(\mathbf{N}(\widetilde{\mathbf{A}}_{\mathcal{G}}^k \mathbf{N})^\intercal) = \mathrm{Tr}((\widetilde{\mathbf{A}}_{\mathcal{G}}^k \mathbf{N})^\intercal \mathbf{N}).$$

Notice that $\widetilde{\mathbf{A}}_{\mathcal{G}} = I_N - \widetilde{\mathbf{L}}_{\mathcal{G}}$ has the same eigenvectors as $\widetilde{\mathbf{L}}_{\mathcal{G}}$, while the eigenvalues are of the form $1 - \lambda_i$. Let $\mathbf{\Gamma}_{\mathcal{G}}$ be the diagonal matrix whose diagonal entries are $1 - \lambda_i, i \le N$. Then we have the following:

$$N D_k = \mathrm{Tr}((\mathbf{W}_{\mathcal{G}} \mathbf{\Gamma}_{\mathcal{G}}^k \mathbf{W}_{\mathcal{G}}^\intercal \mathbf{N})^\intercal \mathbf{N}) = \mathrm{Tr}((\mathbf{W}_{\mathcal{G}}^\intercal \mathbf{N})^\intercal \mathbf{\Gamma}_{\mathcal{G}}^k (\mathbf{W}_{\mathcal{G}}^\intercal \mathbf{N})).$$

Notice that the (i, j)-th entry of $\mathbf{W}_{\mathcal{G}}^\intercal \mathbf{N}$ is the Fourier coefficient $\widehat{\mathbf{m}}_j(i)$. Therefore the i-th diagonal entry of $(\mathbf{W}_{\mathcal{G}}^\intercal \mathbf{N})^\intercal \mathbf{\Gamma}_{\mathcal{G}}^k (\mathbf{W}_{\mathcal{G}}^\intercal \mathbf{N})$ is $(1 - \lambda_i)^k \sum_{1 \le k \le N} \widehat{\mathbf{m}}_j(i)^2$. Therefore, we have:

$$N D_k = \mathrm{Tr}((\mathbf{W}_{\mathcal{G}}^\intercal \mathbf{N})^\intercal \mathbf{\Gamma}_{\mathcal{G}}^k (\mathbf{W}_{\mathcal{G}}^\intercal \mathbf{N})) = \sum_{1 \le i \le N} (1 - \lambda_i)^k \sum_{1 \le k \le N} \widehat{\mathbf{m}}_j(i)^2.$$

This proves the claim for (a).

To show (b), we note that $\mathbb{E}[\langle \mathbf{n}_i, \mathbf{g}_i \rangle] = \langle \mathbb{E}[\mathbf{n}_i], \mathbf{g}_i \rangle = 0$ since $\mathbf{g}_i$ is deterministic from (4) and $\mathbb{E}[\mathbf{n}_i] = 0$. Therefore, $\mathbb{E}[H_k] = 0$. □

We can say more about the summand D_k. The eigenvalues $\lambda_i, 1 \le i \le N$ are known to belong to $[0, 2]$. Hence, the following holds:

Corollary 1. *For all* $l \ge 0$*, the sequence* D_{2l} *is monotonically decreasing, i.e.,* $D_{2l+2} \le D_{2l}$.

For general k, the trend depends on the size $\widehat{\mathbf{m}}_j(i)$ for different i. In particular, if the signal is more concentrated on the low-frequency components, i.e., $1-\lambda_i \geq 0$, then an average reduction in D_k should be observed for general k. In the homophilic setting, due to smoothness, the signal is likely concentrated for those frequency components where $1-\lambda_i \approx 1$. Therefore, the reduction in D_k is expected to be less pronounced.

If the summand D_k is (made) small, then the average correlation E_k is dominated by H_k. Since $\mathbf{g}_i$ depends only on the graph structure and class centroids, it is deterministic once the graph and labels are fixed; therefore, we obtain $\mathbb{E}[H_k] = 0$. Consequently, when $E_k \approx H_k$, then $\mathbf{n}_i^{(k)}$ is effectively replaced by $\mathbf{g}_i$ in the correlation computation. As $\mathbf{g}_i$ has no expected alignment with the initial feature noise $\mathbf{n}_i$, the two components are decoupled in expectation, resulting in weak correlation between them.

C Experimental details

C.1 Details of datasets

We refer the reader to Table 11 for detailed statistics of the datasets. Detailed descriptions of the datasets are given below:

Table 11: Statistics of heterophilic and homophilic graph datasets

Dataset	Nodes	Edges	Classes	Node Features	Data splits
Texas	183	309	5	1793	48%/32%/20%
Cornel	183	295	5	1703	48%/32%/20%
Wisconsin	251	466	5	1703	48%/32%/20%
Squirrel-filtered	2205	46557	5	2089	48%/32%/20%
Chameleon-filtered	864	7754	5	2325	48%/32%/20%
Actor	7600	33391	5	932	48%/32%/20%
Roman-empire	22662	32927	18	300	50%/25%/25%
Amazon-ratings	24492	186100	5	300	50%/25%/25%
Arxiv-year	169343	1166243	5	128	50%/25%/25%
Cora	2708	5429	7	1433	standard
Citeseer	3327	4732	6	3703	standard
PubMed	19717	88651	3	500	standard
Computer	13752	574418	10	767	10%/10%/80%
Photo	7650	119081	8	745	10%/10%/80%
Ogbn-arxiv	169343	1166243	40	128	standard

Texas, Wisconsin and Cornell [47]. These datasets are webpage networks collected by Carnegie Mellon University from computer science departments at various universities. In each network, nodes represent web pages, and edges denote hyperlinks between them. Node features are derived from bag-of-words representations of the web pages. The task is to classify nodes into five categories: student, project, course, staff, and faculty.

Chameleon, Crocodile and Squirrel [47]. These datasets represent Wikipedia networks, with nodes corresponding to web pages and edges denoting hyperlinks between them. Node features are derived from prominent informative nouns on the pages, while node labels reflect the average daily traffic of each web page. The *Squirrel-filtered* and *Chameleon-filtered* variants remove duplicate nodes to prevent training–test leakage [20].

Actor [48]. This dataset is an actor-induced subgraph extracted from the film-director-actor-writer network. Nodes represent actors, and edges indicate their co-occurrence on the same Wikipedia page. Node features are derived from keywords on the actors' Wikipedia pages, while labels categorize the actors into five groups based on the content of their Wikipedia entries.

For **Texas, Wisconsin, Cornell, Chameleon, Crocodile, Squirrel, and Actor** datasets, we utilize the raw data provided by Geom-GCN [48] with the standard fixed 10-fold split for our experiments. These datasets are available for download at: `https://github.com/graphdml-uiuc-jlu/geom-gcn`.

Roman-empire [49] is a heterophilous graph derived from the English Wikipedia article on the Roman Empire. Each node represents a word (possibly non-unique) in the text, with features based on word embeddings. Node classes correspond to syntactic roles, with the 17 most frequent roles as

distinct classes, and all others grouped into an 18th class. Following [49], we use the fixed 10 random splits with a 50%/25%/25% ratio for training, validation, and testing.

Arxiv-year [50] is a citation network derived from a subset of the Microsoft Academic Graph, focusing on predicting the publication year of papers. Nodes represent papers, and edges indicate citation relationships. Node features are computed as the average of word embeddings from the titles and abstracts. Following [50], the dataset is split into training, validation, and testing sets with a 50%/25%/25% ratio.

Cora, Citeseer, and Pubmed [16]. These datasets are among the most widely used benchmarks for node classification. Each dataset represents a citation graph with high homophily, where nodes correspond to documents and edges represent citation relationships. Node class labels reflect the research field, and node features are derived from a bag-of-words representation of the abstracts. The public dataset split is used for evaluation, with 20 nodes per class designated for training, and 500 and 1,000 nodes fixed for validation and testing, respectively.

Computer and Photo [17, 51]. These datasets are co-purchase graphs from Amazon, where nodes represent products, and edges connect products frequently bought together. Node features are derived from product reviews, while class labels correspond to product categories. Following the experimental setup in [41], the nodes are randomly split into training, validation, and testing sets, with proportions of 10%, 10%, and 80%, respectively.

C.2 Baselines

DGI [21]: Deep Graph InfoMax (DGI) is an unsupervised learning method that maximizes mutual information between node embeddings and a global graph representation. It employs a readout function to generate the graph-level summary and a discriminator to distinguish between positive (original) and negative (shuffled) node-feature samples, enabling effective graph representation learning.

GMI [22]: Graphical Mutual Information (GMI) measures the mutual information between input graphs and hidden representations by capturing correlations in both node features and graph topology. It extends traditional mutual information computation to the graph domain, ensuring comprehensive representation learning.

MVGRL [23]: Contrastive Multi-View Representation Learning (MVGRL) leverages multiple graph views generated through graph diffusion processes. It contrasts node-level and graph-level representations across these views using a discriminator, enabling robust multi-view graph representation learning.

GRACE [5]: Graph contrastive representation learning (GRACE) model generates two correlated graph views by randomly removing edges and masking features. It focuses on contrasting node embeddings across these views using contrastive loss, maximizing their agreement while incorporating inter-view and intra-view negative pairs, without relying on injective readout functions for graph embeddings.

CCA-SSG [24]: Canonical Correlation Analysis inspired Self-Supervised Learning on Graphs (CCA-SSG) is a graph contrastive learning model that enhances node representations by maximizing the correlation between two augmented views of the same graph while reducing correlations across feature dimensions within each view.

BGRL [17]: Bootstrapped Graph Latents (BGRL) is a graph representation learning method that predicts alternative augmentations of the input using simple augmentations, eliminating the need for negative examples.

AFGRL [25]: Augmentation-Free Graph Representation Learning (AFGRL) builds on the BGRL framework, avoiding augmentation schemes by generating positive samples directly from the original graph. This approach captures both local structural and global semantic information, offering an alternative to traditional graph contrastive methods, though at the cost of increased computational complexity.

DSSL [7]: Decoupled self-supervised learning (DSSL) is a flexible, encoder-agnostic representation learning framework that decouples diverse neighborhood contexts using latent variable modeling, enabling unsupervised learning without requiring augmentations.

SP-GCL [26]: Single-Pass Graph Contrastive Learning (SP-GCL) is a single-pass graph contrastive learning method that leverages the concentration property of node representations, eliminating the need for graph augmentations.

GraphACL [13]: Graph Asymmetric Contrastive Learning (GraphACL) is a simple and effective graph contrastive learning approach that captures one-hop neighborhood context and two-hop monophily similarities in an asymmetric learning framework, without relying on graph augmentations or homophily assumptions.

PolyGCL [9]: It is a graph contrastive learning pipeline that leverages polynomial filters with learnable parameters to generate low-pass and high-pass spectral views, achieving contrastive learning without relying on complex data augmentations.

GraphECL [27]: It is a simple and efficient contrastive learning method that eliminates message passing during inference by coupling an MLP with a GNN, enabling the MLP to efficiently mimic the GNN's computations, but this design limits representational flexibility and still relies on negative samples for training.

LOHA [28]: It is a self-supervised graph spectral contrastive framework that directly contrasts low-pass and high-pass views based on their natural distinct specialties without additional data augmentations.

EPAGCL [8]: Error-Passing-based Graph Contrastive Learning (EPAGCL) is an augmentation-based GCL model that generates views by adding or dropping edges according to weights derived from the Error Passing Rate (EPR).

SDMG [10]: Smooth Diffusion Model for Graphs (SDMG) is a novel self-supervised framework that learns recognition-oriented representations without labels, employing two dedicated low-frequency encoders, one for node features and another for topology, to distill global low-frequency signals.

C.3 Attack methods

We consider four *black-box* topology attacks in the evasion setting: Random, PRBCD [33], Nettack [34], and Metattack [35]. Additionally, we further consider two *white-box* attacks (i.e., PGD [36] and PRBCD) that jointly perturb both the graph structure and node features. A detailed description of these attack methods is provided below.

Random attack: Adds noisy edges by randomly selecting node pairs across the graph. The number of edges inserted is determined by a perturbation ratio with respect to the original edge count.

PRBCD: The Projected Randomized Block Coordinate Descent (PRBCD) attack perturbs the adjacency matrix $\mathbf{A}$ by iteratively adding or removing edges to maximize the classification loss of a surrogate GNN (e.g., GCN). It employs a projected randomized block coordinate descent strategy with a fixed budget of edge modifications, ensuring efficient and scalable adversarial perturbations. In the white-box setting, PRBCD extends naturally to jointly perturb node features by exploiting full access to model parameters and gradients.

Nettack: A targeted structure-based attack designed to mislead node classification. It manipulates the graph by removing edges to same-class nodes, thereby lowering classification confidence, and by adding edges to different-class nodes to trigger misclassification. Using a surrogate GNN for guidance, it greedily selects the most impactful edge modifications within a fixed budget.

Metattack: A global structure-based attack that perturbs the adjacency matrix $\mathbf{A}$ by leveraging meta-gradients of a surrogate GNN. It modifies the graph to maximize overall classification loss, thereby degrading performance across all nodes.

PGD: The Projected Gradient Descent (PGD) attack is a white-box method that jointly perturbs graph structure and node features to maximize the target model's classification loss. It applies iterative gradient-based updates within a fixed perturbation budget, projecting modifications back into the feasible space after each step. With full access to model parameters and gradients, PGD delivers strong and precise attacks.

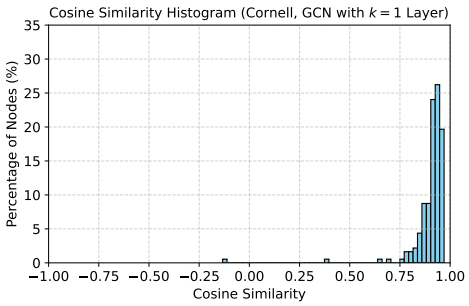
D More numerical results

D.1 Performance and noise correlation

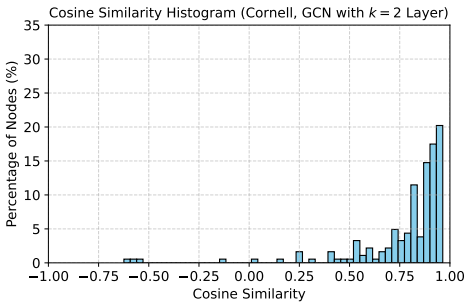
We illustrate that if the two noise sources, namely feature and structural noise, are less correlated, then the resulting GCN-MLP has a better performance. For this, we empirically verify that aggregating feature representations with weakly correlated structural representations helps mitigate feature noise.

We visualize the cosine similarity histograms between structural features (together with inherent structural noise, captured by the GCN) and node feature noise (isolated by the MLP) on three datasets: Cornell, Roman, and Cora, with $k = 1$ or $k = 2$ GCN layers. The results are shown in Fig. 4, Fig. 5, and Fig. 6, respectively.

In general, we always observe that higher accuracy is associated with weaker correlation. Taking Cornell as an example, when the GCN has $k = 2$ layers, the cosine similarity histogram shifts from being concentrated near 1 (strong correlation) toward 0 (weak correlation), compared with $k = 1$. Performance improves significantly, which agrees with our discussions.

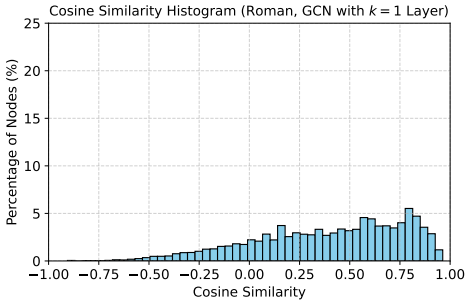


(a) Performance (Cornell): 58.38 ± 3.86

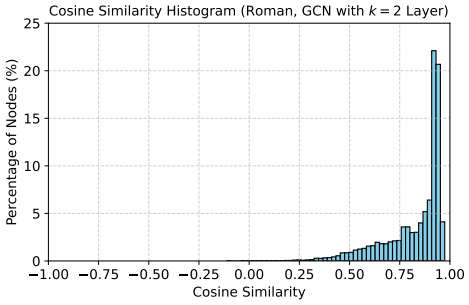


(b) Performance (Cornell): 75.14 ± 3.58

Figure 4: Performance v.s. noise correlation on Cornell



(a) Performance (Roman): 77.74 ± 0.44

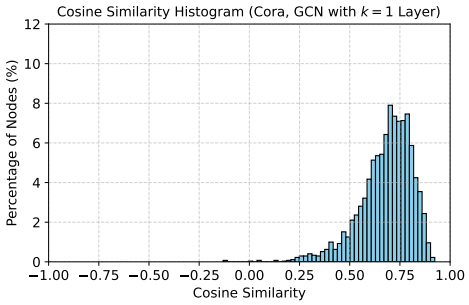


(b) Performance (Roman): 67.12 ± 0.51

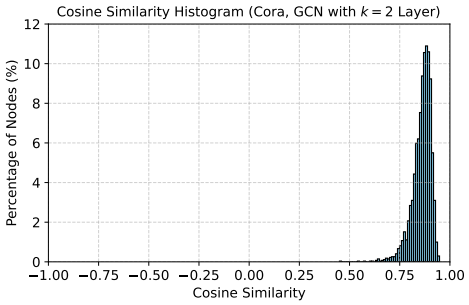
Figure 5: Performance v.s. noise correlation on Roman

D.2 Quantitative Analysis for Noise Cancellation

To quantitatively support the proposed noise-cancellation principle, we analyze node representations before and after graph aggregation using a class-wise centroid-to-deviation ratio, which can be viewed as an empirical counterpart of the NCR introduced in the main text. For each class, the class centroid is treated as the class-consistent signal component, while the deviation of each node representation from its centroid is treated as a noise-like residual. We compute the ratio between the centroid energy and the average residual energy, together with its dB value. A larger ratio indicates that the representations are more concentrated around the class prototype and thus exhibit less within-class residual variation.



(a) Performance (Cora): 77.37 ± 0.18



(b) Performance (Cora): 71.31 ± 0.30

Figure 6: Performance v.s. noise correlation on Cora

We evaluate this measure on heterophilic graphs, including Wisconsin and Roman. As shown in Table 12, the ratio consistently increases after graph aggregation. This indicates that aggregation enhances the class-consistent component while suppressing within-class residual variation, providing quantitative evidence for the proposed noise-cancellation effect.

Table 12: Class-wise Centroid-to-deviation ratio

	Before	After
Wisconsin	1.61/2.08db	2.74/4.37db
Roman	0.91/-0.40db	5.04/7.02db

D.3 Symmetric and asymmetric normalized adjacency matrices for GCN

Recall that the symmetric normalized adjacency matrix $\widehat{\mathbf{A}}_{\mathcal{G}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is commonly used for the basic GCN architecture. However, our theoretical analysis is based on the asymmetric version $\widetilde{\mathbf{A}}_{\mathcal{G}} = \mathbf{D}^{-1}\mathbf{A}$. Therefore, to bridge the gap, we compare their empirical performance with the exact same model architecture, except that different matrices are used for message passing. The results are shown in Table 13. We notice that the approaches have comparable performance on most datasets.

Table 13: Node classification results (%) for GCNs with different message passing matrices.

	Texas	Chameleon	Wisconsin	Cornell	Crocodile	Actor	Roman
$\widehat{\mathbf{A}}_{\mathcal{G}}$	78.14±0.57	46.48±0.45	84.90±0.20	70.78±0.87	66.47± 1.20	36.79± 0.91	78.21± 0.39
$\widetilde{\mathbf{A}}_{\mathcal{G}}$	78.22±1.34	43.41±0.20	84.14±0.37	73.11±1.19	66.32± 1.06	36.30± 0.79	77.64± 0.61

D.4 More robustness results

To further assess the robustness of GCN-MLP, we perform node classification under black-box attacks on additional homophilic and heterophilic datasets (e.g., Photo, Citeseer, Wisconsin, Cornell, Texas, and Actor). As shown in Table 14 and Table 15, the results reinforce the robustness of GCN-MLP across a broader range of benchmarks.

D.5 Graph classification and link prediction results

While most GCL methods target node-level representation learning and do not provide a straightforward graph-level extension, we assess GCN-MLP's generality by applying a simple, non-parametric readout (MeanPooling) to obtain graph-level embeddings. We evaluate this configuration on two standard graph-classification benchmarks, Proteins and DD, and compare against recent graph-level GCL models such as GraphCL and DRGCL as well as strong node-level baselines adapted to the graph-level setting (e.g., GraphACL). GCN-MLP achieves competitive performance compared with node-level baselines and yields results on par with specialized graph-level contrastive methods,

Table 14: Black-box attack robust accuracy results(%) on graph evasion attack for node classification.

Dataset	Attack	FROND	GCL-Jac	Ariel	Res-GRACE	GraphACL	PolyGCL	LOHA	EPAGCL	SDMG	GCN-MLP
Photo	*clean*	92.93±0.46	91.46±0.50	85.75±1.21	92.23±1.22	93.31±0.19	91.45±0.35	86.46±0.41	93.05±0.23	**94.10±0.20**	**93.41±0.88**
	Random	89.90±1.21	86.40±0.74	80.62±1.53	87.79±1.93	26.61±0.05	**90.17±0.99**	85.83±1.12	84.08±1.50	89.90±0.78	**92.94±0.58**
	PRBCD	88.58±1.05	85.24±1.30	80.58±1.62	85.39±4.19	29.13±0.95	**89.65±0.39**	86.35±1.07	80.60±2.72	89.42±0.96	**92.84±0.28**
	Metattack	89.61±1.13	86.20±1.06	82.76±1.11	85.46±1.56	28.42±0.74	**91.06±1.36**	86.56±0.89	85.65±0.56	90.78±0.99	**91.14±0.68**
	Nettack	91.17±1.35	90.50±0.63	85.28±0.91	91.51±1.40	32.84±0.25	**91.29±1.15**	87.40±0.89	89.59±1.05	90.29±0.56	**92.34±0.52**
Citeseer	*clean*	71.37±1.34	70.52±0.65	50.89±3.76	71.72±0.62	**73.60±0.70**	71.82±0.45	71.95±0.45	71.94±0.57	**73.20±0.50**	70.12±0.44
	Random	70.23±1.40	57.26±4.20	44.98±3.45	56.69±2.63	68.13±0.44	**71.58±0.24**	**71.70±0.29**	63.10±1.08	71.47±0.47	69.90±0.00
	PRBCD	71.47±1.29	58.30±4.11	46.02±3.16	58.86±266	70.52±1.16	71.19±0.61	**71.60±0.63**	64.54±2.00	71.28±0.51	69.92±0.04
	Metattack	67.94±1.42	57.51±5.21	36.68±3.76	36.20±5.62	20.50±0.28	**71.78±0.42**	42.99±4.02	47.24±2.67	58.52±0.54	**69.92±0.04**
	Nettack	70.05±1.10	59.40±4.17	46.45±3.16	58.18±2.65	**71.93±1.10**	70.33±0.50	**71.01±0.34**	65.27±1.21	70.88±0.78	69.90±0.00
Wisconsin	*clean*	67.84±3.84	43.53±6.19	56.08±4.31	52.35±7.18	69.22±0.40	**76.92±1.48**	74.27±1.99	63.73±3.95	52.68±1.21	**85.10±2.35**
	Random	69.61±4.49	44.71±6.43	51.18±5.44	51.76±6.27	51.56±5.63	75.23±3.13	**76.47±4.12**	59.02±4.59	51.18±0.98	**85.29±1.81**
	PRBCD	67.65±5.28	44.71±6.72	55.88±4.41	51.37±6.67	52.55±5.13	74.60±3.14	**75.29±4.12**	60.39±6.61	50.98±0.78	**84.90±2.33**
	Metattack	64.51±5.98	43.53±4.09	50.98±4.64	50.59±6.06	52.15±5.08	**76.67±3.92**	74.71±4.31	60.39±4.79	51.67±1.47	**84.90±1.76**
	Nettack	70.78±6.17	44.71±5.32	55.29±5.02	50.00±5.70	53.73±5.16	**77.65±3.92**	75.49±3.73	59.02±2.97	51.57±1.57	**85.10±2.00**
Cornell	*clean*	**63.24±9.38**	42.97±6.78	51.89±6.71	51.08±5.19	59.33±1.48	44.11±1.12	55.25±1.32	53.78±3.78	45.59±0.67	**73.78±5.68**
	Random	**63.24±7.27**	37.30±4.49	40.00±4.95	49.19±4.15	42.97±8.10	43.78±5.14	45.68±3.51	54.32±6.33	45.49±7.72	**73.78±5.68**
	PRBCD	**64.86±5.27**	41.62±9.83	48.38±6.33	48.92±5.98	46.22±9.66	44.59±4.05	51.08±3.24	53.24±6.74	45.14±7.65	**73.78±5.68**
	Metattack	**67.03±5.51**	38.65±6.63	49.73±7.85	49.73±6.07	45.14±6.87	42.43±4.87	48.11±5.14	55.68±5.86	45.22±8.33	**73.78±5.68**
	Nettack	**66.49±6.53**	41.08±7.03	50.54±6.95	49.19±5.24	49.73±7.45	43.78±3.24	52.43±3.51	51.89±3.59	44.68±7.97	**73.78±5.68**
Texas	*clean*	**74.32±5.16**	57.57±5.68	61.35±6.63	57.84±5.69	71.08±0.34	72.51±1.95	67.75±2.70	68.92±4.05	53.60±2.67	**77.57±4.37**
	Random	72.70±4.59	55.41±6.97	55.14±5.82	54.59±8.18	56.22±5.95	**73.51±2.16**	64.59±2.97	73.51±3.24	53.92±3.27	**77.03±5.30**
	PRBCD	**74.05±6.53**	57.57±5.14	58.38±9.06	57.84±5.16	57.03±4.67	67.30±4.87	64.59±3.24	65.95±4.32	53.51±2.14	**77.30±6.30**
	Metattack	**72.97±5.41**	55.41±7.38	55.95±5.14	56.49±5.33	58.11±6.14	68.92±4.32	66.49±2.70	63.24±4.55	53.38±2.27	**78.11±5.98**
	Nettack	**73.24±5.33**	56.22±6.49	61.08±7.17	56.49±6.89	56.76±5.70	71.08±4.86	65.41±2.97	64.59±4.26	53.92±3.27	**77.84±5.10**
Actor	*clean*	**35.08±1.08**	29.25±1.21	24.36±1.11	30.72±0.72	30.03±0.13	34.37±0.69	33.69±0.73	30.02±0.91	26.74±0.13	**36.56±0.93**
	Random	**35.15±0.78**	27.59±1.12	25.64±1.02	30.16±1.09	28.36±1.95	25.41±0.72	34.19±0.59	28.92±1.03	27.09±0.68	**36.19±0.77**
	PRBCD	**35.04±0.90**	27.76±1.66	24.95±0.89	30.48±1.28	28.37±1.95	27.21±0.64	26.23±0.79	28.66±2.01	26.79±0.82	**36.47±1.05**
	Metattack	**32.34±7.10**	28.00±1.10	25.54±0.75	30.34±1.04	28.45±1.26	28.29±0.42	26.97±0.65	29.65±1.12	26.78±0.91	**36.56±1.12**
	Nettack	**34.97±0.88**	28.87±0.73	25.51±0.95	30.86±0.96	28.60±1.20	25.96±0.86	27.20±0.74	30.05±0.81	26.72±0.79	**36.14±0.67**

Table 15: White-box attack robust accuracy (%) under 15% perturbation for node classification.

Dataset	Attack	FROND	Res-GRACE	GraphACL	PolyGCL	LOHA	EPAGCL	SDMG	GCN-MLP
Photo	*clean*	92.03±1.27	92.50±0.17	93.31±0.19	91.45±0.35	86.46±0.41	93.05±0.23	**94.10±0.20**	**93.41±0.88**
	PGD	14.18±5.16	45.46±2.05	30.94±3.62	22.93±2.73	**59.35±0.43**	5.02±2.85	11.98±0.81	**90.56±0.54**
	PRBCD	15.08±7.52	40.72±3.70	28.56±1.16	15.86±1.99	**52.82±3.45**	9.91±1.73	8.71±6.22	**75.42±0.69**
Texas	*clean*	**74.86±3.21**	54.59±5.51	71.08±0.34	72.51±1.95	67.75±2.70	68.92±4.05	53.60±2.67	**78.38±4.68**
	PGD	**69.46±7.16**	28.85± 8.55	21.62±7.20	57.84±14.05	47.30±16.45	13.78±6.78	30.49±4.32	**75.68±6.28**
	PRBCD	**65.14±3.91**	26.49±11.22	23.78±8.99	51.35±10.26	32.70±10.50	14.32±6.29	19.43±4.39	**67.03±5.10**
Wisconsin	*clean*	67.84±3.84	52.35±7.18	69.22±0.40	**76.92±1.48**	74.27±1.99	63.73±3.95	52.68±1.21	**85.10±2.35**
	PGD	62.16±6.01	30.45±6.89	8.04±3.75	66.27±6.06	**66.86±7.52**	18.43±10.57	15.29±17.60	**80.98±4.64**
	PRBCD	57.06±5.71	27.56±8.36	11.37±3.56	**60.59±5.65**	57.06±7.36	33.92±11.20	18.24±12.15	**73.53±4.04**
Cornell	*clean*	**63.24±9.38**	51.08± 5.19	59.33±1.48	44.11±1.12	55.35±1.32	53.78±3.72	45.59±0.67	**73.78±5.68**
	PGD	44.36±8.34	30.58± 6.13	29.46±10.23	38.65±6.17	48.11±5.10	20.00±9.98	39.46±11.67	**53.78±8.92**
	PRBCD	43.81±8.00	28.13± 5.47	37.03±8.74	36.76± 7.47	44.05±7.65	20.54±13.20	13.24±8.24	**52.16±9.52**

demonstrating that our proposed GCN-MLP framework generalizes effectively beyond node-level tasks.

Table 16: Graph classification results (%); The first 4 rows are from node-level GCL methods adapted to graph-level tasks, and the next 3 rows are from graph-level models.

Method	Proteins	DD	PTC-MR	MUTAG	Avg. rank.
MVGRL [52]	74.02±0.30	75.20±0.40	- -	89.20±1.30	5.33
GraphACL[13]	73.50±0.70	- -	- -	89.40±2.00	5.50
SimMLP[43]	**75.30±0.10**	78.40±0.50	60.30±1.10	87.70±0.20	3.63
SDMG [10]	73.16±0.16	72.66±3.16	56.70±2.02	**91.58±0.28**	5.00
InfoGraph[53]	74.44±0.40	72.85±1.70	**61.70±1.40**	89.00± 1.10	4.50
GraphCL[38]	74.39±0.45	**78.62±0.40**	- -	86.80± 1.30	4.67
DRGCL[54]	75.20±0.60	78.40±0.70	- -	89.50± 0.60	**2.83**
GCN-MLP	**75.41±0.35**	77.00±0.45	**62.27±1.44**	**89.56± 0.85**	**2.00**

Table 17: Link prediction results.

Method	CiteSeer	Pubmed
DEAL	93.70 ± 0.30	96.60 ± 0.10
T-BGRL	95.30 ± 0.30	–
AFECL	96.60 ± 0.20	–
GCN-MLP	95.48 ± 0.69	94.46 ± 0.11

For link prediction, we following AFECL's (unsupervised) protocol (a link-prediction-specific GCL method [42]). Specifically, we freeze the pretrained encoder and train an edge classifier using the learned node embeddings. Results (See Tab. 4 below) are comparable to AFECL [42], TBGRL [55] and DEAL [56], supporting the generality of our proposed noise-cancellation principle to edge-level tasks.

E Limitations and future work

Following standard evaluation protocols in graph contrastive learning, this work considers static graph settings. The core contribution, the noise cancellation principle, provides a valuable new theoretical perspective for GCL, departing from the mainstream reliance on complex encoders. This principle is orthogonal to the static/dynamic distinction, and extending it to dynamic graphs is a promising future direction.

F Broader impacts

This work introduces a new principle for graph contrastive learning: noise cancellation via cross-view aggregation, where node feature noise can be mitigated by combining it with structural signals derived from the graph topology. Based on this principle, we design an augmentation-free, negative-free contrastive framework that treats node attributes and graph structure as two inherently complementary views, implemented with a simple GCN–MLP dual encoder. This perspective challenges the prevailing reliance on heavy augmentations and complex objectives, and offers a practical route to robust representation learning, especially on heterophilic graphs, at low computational and memory cost. The societal impact of this work depends on a commitment to ethical standards and responsible use, ensuring that advancements in contrastive learning lead to positive outcomes without exacerbating biases, inequality, or misuse in sensitive applications.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The paper's contributions and scope are detailed in the abstract and introduction Section 1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have discussed the limitations in Section E.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The theorems proposed in Section 3 are supported by detailed proofs provided in Appendix B.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have included the implementation details in Appendix C and the hyperparameters in Section 5.3.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [Yes]

Justification: We have provided the source code in the supplementary material.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We have included the implementation details in Appendix C and the hyperparameters in Section 5.3.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report mean and standard deviation values in our main experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the computing cost in Section 5.5.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We have discussed broader impacts in Section F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the original paper that produced the code package or dataset.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

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

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.