Spectral Graph Sparsification Preserves Representation Geometry in Graph Neural Networks

Sanjukta Krishnagopal
Department of Computer Science, University of California Santa Barbara
sanjukta@ucsb.edu

Abstract

Spectral graph sparsification is a classical tool for reducing graph complexity while preserving Laplacian quadratic forms. In graph neural networks (GNNs), sparsification is often used to accelerate computation while maintaining predictive performance. In this work, we study a complementary representation-level question: does sparsification preserve the geometry of learned embeddings?

For polynomial-filter GNNs, we prove that any ϵ-spectral sparsifier induces $O(\epsilon)$ perturbations in polynomial graph filters, multilayer hidden representations, and their Gram matrices. These guarantees imply stability of squared pairwise distances, class means, and covariance structure in embedding space. We further establish finite-time training stability: under smoothness and boundedness assumptions, gradient descent on dense and sparsified graphs produces weight trajectories whose separation grows at most proportionally to the sparsification distortion.

Empirically, effective-resistance sparsification validates the predicted perturbation chain on synthetic graphs and preserves hidden representation geometry on real datasets. In our experiments, the Gram matrix and training dynamics show low divergence even under substantial sparsification, consistent with the predicted stability under spectral sparsification. Hidden Gram preservation strongly predicts neighborhood preservation and class-centroid stability across FashionMNIST, Cora, and Paul15. Together, these results show that spectral sparsification preserves not only graph operators, but also the representation geometry that supports downstream use of GNN embeddings for interpretability.

1 Introduction

Graph neural networks (GNNs) are a standard framework for learning from relational data, with applications in citation networks, molecules, recommendation systems, and scientific data analysis [Kipf and Welling, 2017, Hamilton et al., 2017b, Defferrard et al., 2016]. Their effectiveness, however, comes with a scalability cost: training and inference often depend strongly on the size and density of the underlying graph. This is especially true for architectures whose propagation rules are expressed through graph operators, such as spectral and polynomial-filter GNNs [Bruna et al., 2014, Defferrard et al., 2016].

Spectral graph sparsification offers a principled way to reduce this cost. Classical sparsification theory shows that a graph can be replaced by a much sparser graph whose Laplacian $\widetilde{L}$ satisfies

$$(1-\varepsilon)L \preceq \widetilde{L} \preceq (1+\varepsilon)L,$$

thereby preserving quadratic forms and spectral structure up to controlled distortion [Spielman and Srivastava, 2011]. Recent work has begun to use such ideas for efficient GNNs, including effective-resistance- and Laplacian-based sparsification methods for spectral and attention-style architectures

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

[Ding et al., 2025]. Most of this literature evaluates sparsification through operator approximation, runtime, or predictive accuracy.

In this paper, we ask a representation-centered question: *does spectral sparsification preserve the geometry of learned hidden representations?* This question is important because GNNs are not only predictors; they are also representation learners [Hamilton et al., 2017a]. Their hidden states encode pairwise similarities, class separation, and global latent geometry. Two sparse graphs may yield similar accuracy while inducing substantially different embeddings, which can affect downstream uses such as retrieval, clustering, transfer learning, and interpretation. Recent work on feature evolution and neural collapse in GNNs further emphasizes that graph topology can shape representation geometry during training [Kothapalli et al., 2023, Papyan et al., 2020]. We focus on polynomial-filter GNNs, a broad class of spectral graph neural networks in which each layer applies a finite-degree polynomial in a graph operator before feature mixing and nonlinear activation [Defferrard et al., 2016]. This class includes Chebyshev and other finite-hop spectral filters, and it also captures first-order propagation layers when the graph shift is written as a degree-one polynomial operator. In particular, ChebNet is explicitly polynomial and a standard GCN layer is degree one in the graph shift, so both fall within the analyzed class; architectures with learned or feature-dependent graph operators, such as attention-based propagation, are not directly covered by the present guarantees. It is therefore a natural setting for studying spectral sparsification, since the sparsifier directly perturbs the graph operator appearing in the layer. Building on prior analyses of GNN stability and transferability under graph perturbations [Gama et al., 2020, Ruiz et al., 2021, 2023], we move beyond operator-level guarantees and analyze hidden representation preservation directly. We prove that spectral sparsification stabilizes multilayer embeddings and their Gram matrices, yielding preservation of pairwise distances, class means, and covariance structure. We also analyze learning dynamics under gradient descent in a one-layer model and give a multilayer extension under stronger smoothness and boundedness assumptions, connecting to recent work on GNN training behavior on growing graphs [Krishnagopal and Ruiz, 2023].

Contributions.

- We introduce a representation-geometric perspective on graph sparsification for GNNs.
- We prove multilayer embedding stability and Gram-matrix stability under spectral sparsification.
- We establish finite-time training-dynamics stability in a one-layer setting, and a multilayer extension under stronger smoothness assumptions.
- We show empirically, using effective-resistance sparsification as a concrete testbed, on synthetic and real data, that hidden-Gram preservation predicts neighborhood and class-geometry preservation in learned embeddings, improving interpretability and propagation fidelity.

Relation to prior work. Our work connects several lines of research. First, spectral graph sparsification provides classical operator-level guarantees for replacing a graph by a sparse spectral approximation [Spielman and Srivastava, 2011]. Second, recent sparsification-for-GNN methods use these ideas to improve efficiency, focusing mainly on propagation fidelity and predictive performance [Ding et al., 2025]. Third, prior work studies the stability and transferability of GNNs under graph perturbations, graph shifts, and graph-size changes [Gama et al., 2020, Ruiz et al., 2021, 2023]. Fourth, recent work studies the geometry of learned features in graph representation learning, including neural collapse in GNNs [Kothapalli et al., 2023, Papyan et al., 2020]. Finally, our viewpoint is also related to large-graph asymptotic analyses of GNN transferability and learning dynamics [Krishnagopal and Ruiz, 2023]. Our contribution lies at the intersection of these directions: we study whether principled spectral sparsification preserves the geometry of learned representations, not only the propagation operator or task accuracy.

2 Preliminaries and Main Results

2.1 Setup and Notation

Let $G = (V, E)$ be a graph on n nodes, and let $L \in \mathbb{R}^{n\times n}$ denote a symmetric positive semidefinite graph operator, such as the combinatorial or normalized Laplacian. For symmetric matrices $A, B \in$

$\mathbb{R}^{n\times n}$, we write $A \preceq B$ if $B - A$ is positive semidefinite. We use $\|\cdot\|_2$ for the operator norm and $\|\cdot\|_F$ for the Frobenius norm.

We say that $\widetilde{L}$ is an ε-spectral sparsifier of L if

$$(1-\varepsilon)L \preceq \widetilde{L} \preceq (1+\varepsilon)L, \qquad 0 < \varepsilon < 1. \tag{1}$$

We study a class of K-layer polynomial filter graph neural networks of the form

$$H^{(0)} = X, \qquad H^{(k+1)} = \sigma_k\big(p_k(L)H^{(k)}W_k\big), \qquad k = 0, \dots, K-1, \tag{2}$$

where $X \in \mathbb{R}^{n\times d_0}$ is the input feature matrix, $H^{(k)} \in \mathbb{R}^{n\times d_k}$ is the hidden representation at layer k, $W_k \in \mathbb{R}^{d_k\times d_{k+1}}$ is the trainable weight matrix, p_k is a real polynomial, and σ_k is applied entrywise.

The corresponding network on the sparsified graph is defined by

$$\begin{aligned} \widetilde{H}^{(0)} &= X, \\ \widetilde{H}^{(k+1)} &= \sigma_k\big(p_k(\widetilde{L})\widetilde{H}^{(k)}W_k\big). \end{aligned} \tag{3}$$

We denote the final embeddings by

$$Z := H^{(K)}, \qquad \widetilde{Z} := \widetilde{H}^{(K)}.$$

Our analysis is based on the following assumptions.

(A1) Polynomial filters. For each k, $p_k(x) = \sum_{r=0}^{d_k^\star} a_{k,r}x^r$ is a polynomial of finite degree $d_k^\star$.

(A2) Lipschitz nonlinearities. Each activation $\sigma_k : \mathbb{R} \to \mathbb{R}$ is L_{σ_k}-Lipschitz.

(A3) Bounded weights. There exists $B_W > 0$ such that $\|W_k\|_2 \le B_W$ for all k.

(A4) Bounded inputs. The input features satisfy $\|X\|_F \le B_X$.

(A5) Bounded operators. There exists $B_L > 0$ such that $\|L\|_2, \|\widetilde{L}\|_2 \le B_L$.

Assumptions (A1)–(A5) are standard in perturbation analyses of graph-based learning systems and are sufficient for the stability results below. They match the finite-degree filter, Lipschitz nonlinearity, and bounded-operator conditions commonly used in GNN stability and transferability analyses [Gama et al., 2020, Ruiz et al., 2021, 2023].

2.2 Propagation Stability

We first show that spectral sparsification preserves polynomial graph filters.

Theorem 1 (Polynomial filter stability)**.** *Let $p(x) = \sum_{r=0}^{d} a_r x^r$ be a polynomial, and let $L, \widetilde{L}$ satisfy* (1)*. Under (A5),*

$$\|p(L) - p(\widetilde{L})\|_2 \le C_p\,\varepsilon, \tag{4}$$

where

$$C_p := \sum_{r=1}^{d} |a_r|\, r\, B_L^r.$$

Theorem 1 is the operator-level ingredient in our analysis: it says that spectral closeness of the graph implies closeness of the propagation operators used by polynomial-filter GNNs. The full proof is given in Appendix A.2.

2.3 Representation Stability

We next propagate the operator perturbation through a multilayer nonlinear architecture.

Theorem 2 (Multilayer representation stability)**.** *Suppose that* (1) *holds and assumptions (A1)–(A5) are satisfied. Then there exists a constant $C_{\mathrm{rep}} > 0$, depending only on the network depth, filter coefficients, Lipschitz constants, and bounds B_X, B_W, B_L, such that*

$$\|Z - \widetilde{Z}\|_F \le C_{\mathrm{rep}}\,\varepsilon. \tag{5}$$

The constant in Theorem 2 can be written explicitly from the layerwise recursion in Appendix A.3. If

$$\alpha_k = L_{\sigma_k} B_W C_{p_k} B'_k, \qquad \beta_k = L_{\sigma_k} B_W M_k,$$

then one may take

$$C_{\mathrm{rep}} = \sum_{j=0}^{K-1} \alpha_j \prod_{\ell=j+1}^{K-1} \beta_\ell.$$

This also makes the graph-size dependence explicit. For bounded per-node features of fixed dimension, $\|X\|_F = O(\sqrt{n})$. With a normalized or scaled Laplacian ($B_L = O(1)$), fixed depth, and fixed filter degrees, the forward bounds satisfy $B'_k = O(\sqrt{n})$ and hence $\|Z - \widetilde{Z}\|_F = O(\sqrt{n}\,\varepsilon)$. Thus there is no hidden exponential dependence on graph size. For an unnormalized combinatorial Laplacian the dependence remains polynomial through $B_L = \|L\|_2$.

The result shows that spectral sparsification controls not only the graph operator but also the learned embedding itself. The full proof is given in Appendix A.3.

2.4 Representation Geometry

The main conceptual contribution of this paper is that sparsification preserves the geometry of the learned embedding space.

Theorem 3 (Gram matrix stability)**.** *Under the assumptions of Theorem 2, there exist constants* $C_{\mathrm{gram},2}, C_{\mathrm{gram},F} > 0$ *such that*

$$\|ZZ^\top - \widetilde{Z}\widetilde{Z}^\top\|_2 \le C_{\mathrm{gram},2}\,\varepsilon, \qquad \|ZZ^\top - \widetilde{Z}\widetilde{Z}^\top\|_F \le C_{\mathrm{gram},F}\,\varepsilon.$$

Theorem 3 upgrades representation stability to Gram-matrix stability in both spectral and Frobenius norms. Since the Gram matrix encodes all pairwise inner products between node embeddings, this implies preservation of distances and class-level statistics such as centroids and covariances in latent space. The proof is the direct factorization $ZZ^\top - \widetilde{Z}\widetilde{Z}^\top = Z(Z - \widetilde{Z})^\top + (Z - \widetilde{Z})\widetilde{Z}^\top$, followed by the bounded forward-state estimates from Theorem 2. Under the normalized/scaled graph-size regime described above, $\|Z\|_F, \|\widetilde{Z}\|_F = O(\sqrt{n})$, so the corresponding worst-case Gram perturbation is $O(n\varepsilon)$ rather than exponential in n. The full proof is given in Appendix A.4.

The following corollaries formalize this implication.

Corollary 1 (Pairwise distance preservation)**.** *Let* $z_i^\top$ *and* $\widetilde{z}_i^\top$ *denote the* i*th rows of* Z *and* $\widetilde{Z}$*, respectively. Then for all* $i, j \in \{1, \dots, n\}$*,*

$$\left| \|z_i - z_j\|_2^2 - \|\widetilde{z}_i - \widetilde{z}_j\|_2^2 \right| \le 4C_{\mathrm{gram},2}\,\varepsilon. \tag{6}$$

Now suppose that the nodes are partitioned into C classes with index sets $S_1, \dots, S_C$, where $n_c := |S_c|$. Define the class mean and empirical covariance for class c by

$$\mu_c := \frac{1}{n_c} \sum_{i \in S_c} z_i, \qquad \Sigma_c := \frac{1}{n_c} \sum_{i \in S_c} (z_i - \mu_c)(z_i - \mu_c)^\top, \tag{7}$$

and define $\widetilde{\mu}_c, \widetilde{\Sigma}_c$ analogously using $\widetilde{Z}$.

Corollary 2 (Class-statistics stability)**.** *Under the assumptions of Theorem 2, for each class c,*

$$\|\mu_c - \widetilde{\mu}_c\|_2 \le \frac{1}{\sqrt{n_c}} \|Z - \widetilde{Z}\|_F, \tag{8}$$

and, if the row norms of Z *and* $\widetilde{Z}$ *are uniformly bounded by* B_Z*,*

$$\|\Sigma_c - \widetilde{\Sigma}_c\|_2 \le \frac{8B_Z}{\sqrt{n_c}} \|Z - \widetilde{Z}\|_F. \tag{9}$$

In particular, both quantities are $O(\varepsilon)$*.*

Corollary 2 controls several representation diagnostics used in practice. Examples include the displacement of class centroids, pairwise distances and angles between centroids, within-class variance $\mathrm{tr}(\Sigma_c)$, leading covariance eigenvalues, between-/within-class scatter ratios, and nearest-centroid margins. These quantities measure whether classes remain separated, compact, and similarly oriented after sparsification. Neural-collapse-style diagnostics are also covered, since they are built from the same objects: class means, within-class covariances, and normalized centroid geometry. On bounded sets, any such diagnostic that is locally Lipschitz as a function of $\{\mu_c, \Sigma_c\}_{c=1}^{C}$ therefore changes by at most $O(\varepsilon)$ under spectral sparsification.

Proofs of Corollaries 1 and 2 are collected in Appendix A.5.

Theorems 1–3 establish the implication chain

$$L \approx \widetilde{L} \implies p(L) \approx p(\widetilde{L}) \implies Z \approx \widetilde{Z} \implies ZZ^\top \approx \widetilde{Z}\widetilde{Z}^\top.$$

2.5 Training Dynamics

We also study how sparsification perturbs training trajectories. In a one-layer model

$$f(W; L) = \sigma\big(p(L)XW\big), \qquad J(W; L) = \tfrac{1}{2}\|f(W; L) - Y\|_F^2,$$

In addition to (A1)–(A5), we make the following standard assumptions:

(A6) Smooth activation. $\sigma \in C^1(\mathbb{R})$, σ' is bounded, and σ' is Lipschitz.

(A7) Bounded iterates. The gradient descent iterates remain in a bounded set:

$$\|W_t\|_2, \|\widetilde{W}_t\|_2 \le R \qquad \text{for all } t \le T.$$

Assumptions (A6)–(A7) are finite-time regularity conditions used to control perturbations of the gradient field: smoothness makes the gradient locally Lipschitz, while bounded iterates restrict the analysis to the finite trajectory actually traversed by gradient descent. **(A8) Bounded targets.** The target matrix satisfies $\|Y\|_F \le B_Y$.

Let

$$W_{t+1} = W_t - \eta \nabla_W J(W_t; L), \qquad \widetilde{W}_{t+1} = \widetilde{W}_t - \eta \nabla_W J(\widetilde{W}_t; \widetilde{L}),$$

with common initialization $W_0 = \widetilde{W}_0$.

Theorem 4 (One-layer training stability)**.** *Under these assumptions, with constants allowed to depend on B_Y, there exist constants $L_W, L_S > 0$ such that for all $t \le T$,*

$$\|W_{t+1} - \widetilde{W}_{t+1}\|_F \le (1 + \eta L_W)\|W_t - \widetilde{W}_t\|_F + \eta L_S \|p(L) - p(\widetilde{L})\|_2. \tag{10}$$

Consequently, by Theorem 1,

$$\|W_t - \widetilde{W}_t\|_F \le C_{\mathrm{train}} \varepsilon \sum_{j=0}^{t-1} (1 + \eta L_W)^j, \tag{11}$$

for a constant C_{train} independent of ε.

Theorem 4 shows that sparsification perturbs the gradient field only slightly, and hence yields finite-time stability of gradient descent trajectories in the one-layer setting. The full proof is deferred to Appendix A.6.

Multilayer training dynamics. The one-layer result extends to full-batch multilayer gradient descent on bounded finite-time trajectories under analogous smoothness and boundedness assumptions. Let

$$\Theta := (W_0, \dots, W_{K-1}), \qquad F(\Theta; L) := H^{(K)},$$

where

$$H^{(0)} = X, \qquad H^{(k+1)} = \sigma_k\big(p_k(L) H^{(k)} W_k\big),$$

and let

$$J(\Theta; L) := \tfrac{1}{2}\|F(\Theta; L) - Y\|_F^2.$$

In addition to (A1)–(A5) and (A8), we make the following standard assumptions:

(A6′) **Smooth activations.** For each k, $\sigma_k \in C^1(\mathbb{R})$, σ_k' is bounded, and σ_k' is Lipschitz.

(A7′) **Bounded iterates.** There exists $R > 0$ such that for all $t \le T$ and all k,

$$\|W_k^t\|_2,\ \|\widetilde{W}_k^t\|_2 \le R.$$

Under (A1)–(A5), this weight bound implies uniform finite-time bounds on the hidden states by the same forward recursion used in Appendix A.3; hidden-state boundedness therefore need not be imposed as an independent assumption.

Let $\Theta^t, \widetilde{\Theta}^t$ be the corresponding gradient descent iterates with common initialization. We assume smoothness and boundedness for the activation functions and weights, ensuring that perturbations do not grow uncontrollably over finite training steps.

Proposition 1 (Multilayer training stability)**.** *Under these assumptions, there exist constants $L_\Theta, L_S > 0$ such that for all $t \le T$,*

$$\|\Theta^{t+1} - \widetilde{\Theta}^{t+1}\|_F \le (1 + \eta L_\Theta)\|\Theta^t - \widetilde{\Theta}^t\|_F + \eta L_S \sum_{k=0}^{K-1} \|p_k(L) - p_k(\widetilde{L})\|_2. \tag{12}$$

Consequently, by Theorem 1,

$$\|\Theta^t - \widetilde{\Theta}^t\|_F \le \eta L_S \left(\sum_{k=0}^{K-1} C_{p_k}\right) \varepsilon \sum_{j=0}^{t-1} (1 + \eta L_\Theta)^j \le C_{\text{train}}^{(K)} \varepsilon \sum_{j=0}^{t-1} (1 + \eta L_\Theta)^j,$$

where $C_{\text{train}}^{(K)}$ is independent of ε.

The proof (see A.7) follows the same strategy as Theorem 4: one shows that the full gradient field is Lipschitz in the parameters and Lipschitz in the collection of graph filters $\{p_k(L)\}_{k=0}^{K-1}$ on bounded sets, and then combines these bounds with Theorem 1.

3 Experiments

We evaluate the central claim of the paper: in practically relevant sparsification regimes, principled graph sparsification preserves not only graph operators, but also the geometry of learned hidden representations. Our experiments combine controlled synthetic tests, which directly probe the perturbation predictions of Section 2, with practical representation-learning tests on real datasets.

3.1 Setup

We evaluate sparsification on synthetic and real graph datasets. The synthetic experiments use weighted stochastic block models (SBMs) and geometric k-NN graphs constructed to exhibit clear community or manifold structure. The real-data experiments use FashionMNIST [Xiao et al., 2017], Cora [Sen et al., 2008], and Paul15 [Paul et al., 2015], a single-cell RNA-seq dataset profiling myeloid progenitor cells. The full real-data graphs contain 2,200 FashionMNIST nodes, 2,708 Cora nodes, and 2,730 Paul15 cells. Node features are either class-structured synthetic features, for controlled experiments, or the standard features provided with each dataset; details are given in Appendix B.1.

For the theory-aligned experiments in Figures 1 and 2, we use deterministic GNN models to isolate the effect of graph sparsification. Specifically, we use a polynomial graph filter followed by two linear feature-mixing layers for forward-map evaluation, and one- and two-layer deterministic GNNs for training-dynamics experiments. Dense and sparsified models have matched initialization. This deterministic setup is an experimental control, not a restriction of the theory: it isolates the change $L \mapsto \widetilde{L}$ from dropout, minibatching, or unmatched randomness. Standard GCNs remain within the polynomial-filter class analyzed above.

Sparsification is performed using effective-resistance sampling. Effective-resistance sparsification samples edges with probability proportional to their leverage scores $w_e R_e$, where w_e is the edge weight and R_e is the effective resistance between the endpoints of edge e, and then reweights sampled edges so that the sparse Laplacian remains a spectrally accurate approximation of the

original Laplacian [Spielman and Srivastava, 2011, Spielman and Teng, 2011]. This choice provides a concrete testbed for the theory, but our results apply to any method producing a graph $\widetilde{L}$ satisfying the spectral comparison (1). The sparsification level is controlled by the edge-sampling budget, with larger budgets producing denser, lower-distortion sparsifiers. Spectral sparsification is intended to be a computationally efficient preprocessing step: classical constructions are nearly linear in the original graph size and effective-resistance sampling yields sparse spectral approximations with $O(n \log n/\varepsilon^2)$ edges [Spielman and Srivastava, 2011, Spielman and Teng, 2011]. Replacing m edges by $\widetilde{m}$ edges reduces edge-dependent message-passing and storage costs from $O(md)$ to $O(\widetilde{m}d)$ per layer for feature width d. The generalized-eigenvalue calculation used below to report ϵ_{emp} is only a theory-facing diagnostic and is not required by the sparsification or GNN training procedure. Full architectural, optimization, and sparsification details are given in Appendices B.2 and B.3.

3.2 Validation of representation stability

Geometric						
ϵ_{emp}	0.37	0.38	0.41	0.54	0.66	0.75
frac.	0.90	0.87	0.87	0.70	0.58	0.53

SBM						
ϵ_{emp}	0.41	0.42	0.43	0.43	0.56	0.71
frac.	0.96	0.96	0.92	0.89	0.77	0.66

Table 1: Mean empirical spectral distortion ϵ_{emp} and retained edge fraction (frac.) for the synthetic sparsification experiments, averaged over 5 sparsifier draws.

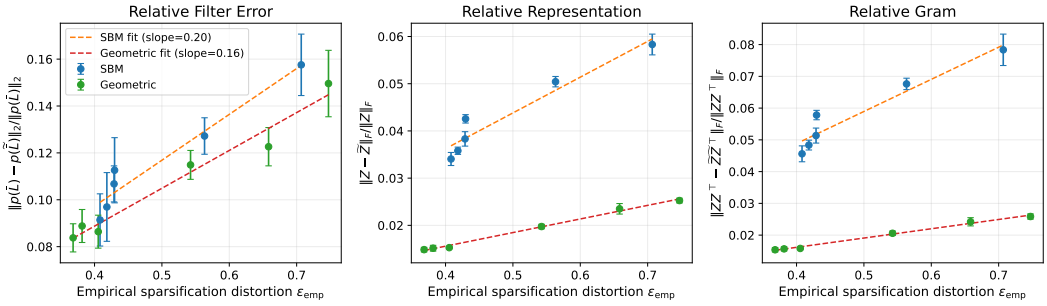

Figure 1: **Controlled validation of representation stability.** We measure relative polynomial-filter error, relative hidden-representation error, and relative hidden-Gram error for effective resistance sparsified weighted SBM and geometric weighted k-NN graphs. The graph operator is the scaled combinatorial Laplacian $\bar{L} = L/\|L\|_2$, and the polynomial filter is $p(\bar{L}) = I - 0.6\bar{L} + 0.15\bar{L}^2$. Across both graph families, all three quantities increase smoothly with ϵ_{emp}, consistent with the stability predictions of Theorems 1–3. Markers show means over five sparsifier draws at each selected budget, and vertical error bars denote one standard deviation.

On weighted SBM and geometric graph families, we construct effective-resistance sparsifiers with varying degrees of sparsification, controlled by varying the sampling budget q. For synthetic graphs, we measure empirical spectral distortion ϵ_{emp} from the generalized eigenvalue envelope of the dense and sparse Laplacians. If $\lambda_{\min}$ and $\lambda_{\max}$ are the extremal generalized eigenvalues of $(\widetilde{L}, L)$ on the nonconstant subspace, we compute

$$\epsilon_{\text{emp}} = \max\{1 - \lambda_{\min}, \lambda_{\max} - 1\}.$$

Table 1 reports the corresponding fraction of retained edges.

We then evaluate the quantities appearing in the theory: relative polynomial-filter error, relative hidden-representation error, and relative hidden-Gram error. The synthetic forward map uses the scaled Laplacian graph operator $\bar{L} = L/\|L\|_2$ and polynomial filter

$$p(\bar{L}) = I - 0.6\bar{L} + 0.15\bar{L}^2.$$

The relative errors reported here are normalized versions of the absolute perturbations controlled in Section 2. Because the dense-graph denominator is fixed with respect to sparsification, this normalization preserves the predicted $O(\varepsilon)$ dependence whenever the denominator is nonzero. Since

the Gram matrix encodes pairwise inner products, stability of $ZZ^\top$ directly reflects preservation of embedding geometry.

Figure 1 validates the perturbation chain of Section 2. Across both graph families, all three quantities increase smoothly and approximately linearly with ϵ_{emp}, consistent with the $O(\epsilon)$ scaling predicted by Theorems 1–3. The relative representation and Gram distortions remain small $(0.03 - 0.08)$ even in substantial-distortion regimes where only roughly half the edges remain ($\epsilon_{\text{emp}} \sim 0.7$). We interpret these large-ϵ_{emp} points as empirical stress tests rather than as a regime in which the perturbative constants are expected to be sharp.

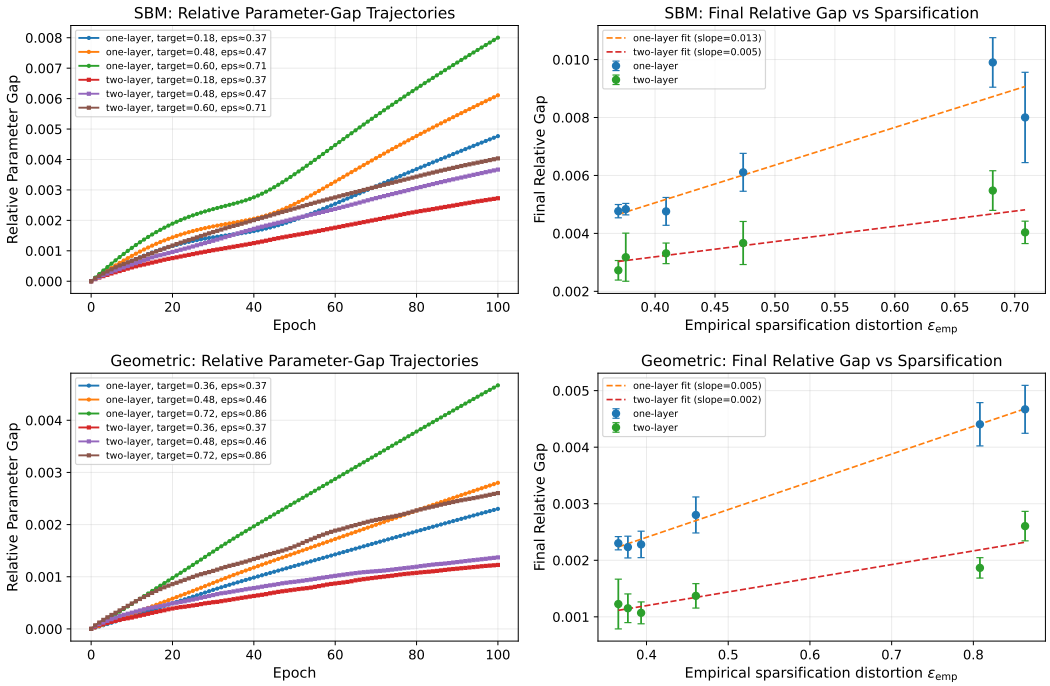

Figure 2: **Training-trajectory perturbation under graph sparsification.** Relative parameter-gap trajectories between models trained on the dense graph and on effective-resistance sparsifiers for one-layer and two-layer deterministic polynomial-filter GNNs on SBM and geometric graph families. Both experiments use the same scaled polynomial-Laplacian graph shift as Figure 1. Left: parameter-gap trajectories over training epochs for representative empirical distortion levels. Right: final relative parameter gap versus empirical sparsification distortion ϵ_{emp}. Markers show means over five sparsifier draws at fixed target distortion, vertical error bars denote one standard deviation. The monotone increase with ϵ_{emp} is consistent with the finite-time perturbation picture of Theorem 4 and Proposition 1. The relative parameter gap is a trajectory-level diagnostic in parameter space and should not be interpreted as graph or representation distortion.

3.3 Training dynamics under graph sparsification

We test whether the finite-time perturbation behavior predicted by Theorem 4 and Proposition 1 is visible in controlled training experiments. For each dense graph and sparsifier, we train dense-graph and sparse-graph models from the same initialization and compare their parameter trajectories over time. The relative parameter gap is defined as the Frobenius norm of the difference between the weight matrices of the models trained on the dense and sparsified graphs:

$$\text{Relative Parameter Gap} = \frac{\|\Theta^t - \widetilde{\Theta}^t\|_F}{\|\Theta^t\|_F}.$$

with the analogous one-layer definition when there is only one weight matrix. This is a trajectory-level diagnostic in parameter space. It should not be interpreted as graph or representation distortion.

Figure 2 shows that, across SBM and geometric graph families, the relative parameter gap grows smoothly over training and is consistently larger for sparsifiers with larger empirical spectral distortion ϵ_{emp}. The final relative gap also increases approximately linearly with ϵ_{emp}. This behavior is consistent with the recurrence in Theorem 4: sparsification introduces an $O(\epsilon)$ perturbation to the gradient field at each step, and these perturbations accumulate over finite training time.

In this polynomial-Laplacian setup, the absolute parameter gaps remain small. Thus, Figure 2 should be read as qualitative evidence for the predicted monotone finite-time dependence on sparsification distortion, while the main representation-level validation is provided by Figure 1 and the real-data geometry experiments below.

3.4 Hidden-geometry preservation on real data

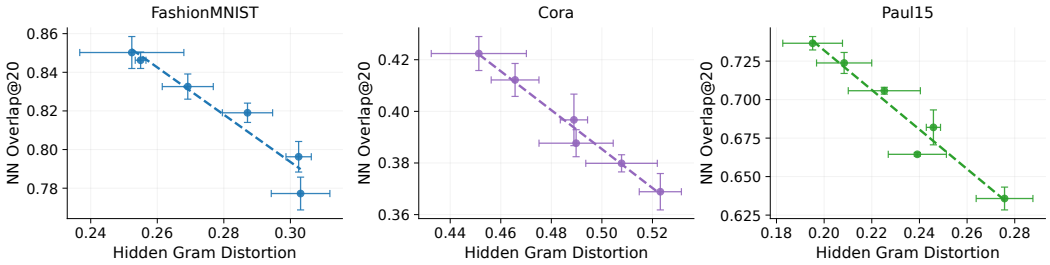

Figure 3: **Hidden Gram preservation predicts neighborhood preservation on real data.** For each sparsification budget and dataset, we repeat the sparsifier with three random seeds and compute both hidden Gram distortion and mean 20-NN overlap on held-out test nodes. Horizontal error bars show one standard deviation of hidden Gram distortion, and vertical error bars show one standard deviation of 20-NN overlap. Across FashionMNIST, Cora, and Paul15, worse hidden Gram distortion is associated with worse neighborhood preservation, suggesting geometry preservation is a useful representation-fidelity metric.

Our main practical question is whether preserving hidden Gram geometry also preserves useful local and class-level structure in representation space. Since the Gram matrix controls pairwise inner products, Theorem 3 and Corollary 1 predict stability of pairwise embedding geometry. Corollary 2 further predicts stability of coarse class statistics such as centroids and within-class spread. For real datasets, we therefore focus directly on representation-level distortion, using hidden Gram distortion as the geometry-preservation metric.

Figure 3 tests the local-structure side of this prediction. The 20-NN overlap measures the fraction of each node's nearest neighbors in the dense embedding that remain nearest neighbors after sparsification, averaged over held-out nodes. At zero distortion the overlap is 1 by construction, providing the natural no-perturbation baseline. Across FashionMNIST, Cora, and Paul15, larger hidden Gram distortion is consistently associated with lower 20-NN overlap. The relationship is strongly monotone on all three datasets, with Spearman correlations -0.92, -0.90, and -0.92, respectively. Because 20-NN overlap measures preservation of each node's local embedding neighborhood, these results show that Gram distortion tracks a practically meaningful notion of embedding fidelity.

Figure 4 tests the class-level side of the same picture. For each dataset, we fit a two-dimensional PCA projection on the dense embedding and project both dense and sparse class centroids into that same basis. Across all three datasets, sparse centroids remain close to their dense counterparts, indicating that sparsification preserves coarse class organization in addition to local neighborhoods.

Appendix B.5 provides an additional Procrustes-aligned embedding visualization, and Appendix B.6 evaluates another downstream diagnostic for representation geometry preservation: a knowledge-distillation task in which a dense-graph teacher provides soft labels for a student trained on the sparsified graph.

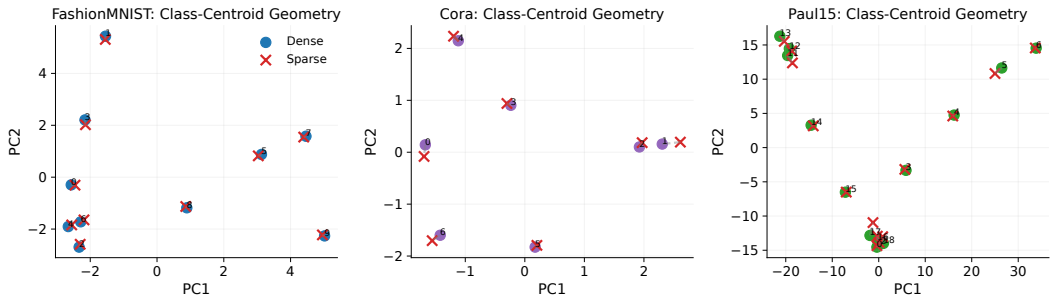

Figure 4: **Class-centroid geometry under sparsification.** Within each dataset, both dense and sparse centroids are projected into the same PCA basis. Small centroid displacements indicate that sparsification preserves class geometry, consistent with the class-statistics stability in Corollary 2.

4 Conclusion

We studied graph sparsification from a representation-geometric perspective. We proved that spectral sparsification preserves not only polynomial graph filters, but also multilayer hidden embeddings and their Gram matrices. These bounds imply stability of pairwise distances, class means, and covariance structure in embedding space. We also studied the training dynamics and established finite-time stability of gradient descent trajectories in controlled one-layer and multilayer settings. Empirically, we showed that representation and training perturbations scale smoothly with empirical spectral distortion on synthetic graphs, and that hidden Gram preservation predicts neighborhood and class-geometry preservation on real datasets. Across our datasets, hidden Gram matrices, embedding-geometry diagnostics, and finite-time training trajectories exhibited substantially smaller deviations than the underlying graph-level spectral distortion. These results suggest that sparsification should be evaluated not only by efficiency or predictive accuracy, but also by the geometry it preserves. This points toward a representation-aware view of graph compression for interpretable graph learning.

Limitations: Our results assume that the sparsified graph satisfies an ϵ-spectral approximation, which, despite being widely used, may hold only approximately in practical implementations. The analysis also relies on boundedness, Lipschitz, and smoothness assumptions, and the training-dynamics guarantees apply to controlled finite-time full-batch settings. The guarantees directly cover fixed polynomial graph operators (including GCN- and ChebNet-style propagation) but not learned, feature-dependent graph operators. Our experiments are deliberately controlled and modest in scale; evaluating larger benchmarks, additional architectures and sparsifiers, more aggressive compression, and nodewise rather than aggregate representation deviations remains important future work. Finally, while we show that geometric quantities are preserved and relate them to several downstream diagnostics, a deeper understanding of how representation geometry controls generalization remains an important direction for future work.

Broader impact: Geometric representation preserving sparsification may reduce the computational cost of graph learning and make large-scale graph models more accessible. However, sparsification may disproportionately affect low-degree, peripheral, or minority nodes, potentially degrading performance for underrepresented groups in sensitive applications. Spectral or representation-level guarantees do not by themselves ensure fairness, privacy, or robustness.

Code availability. Code can be found at `https://github.com/chimeraki/Spectral-Sparsification-Preserves-Representation-Geometry`.

References

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*, 2014. URL `https://openreview.net/forum?id=DQNsQf-UsoDBa`.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29*, pages 3844–3852, 2016.

Haipeng Ding, Zhewei Wei, and Yuhang Ye. Large-scale spectral graph neural networks via Laplacian sparsification: Technical report. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 213–223, 2025. doi: 10.1145/3690624.3709241.

Fernando Gama, Joan Bruna, and Alejandro Ribeiro. Stability of graph neural networks to relative perturbations. In *ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 9070–9074, 2020. doi: 10.1109/ICASSP40776.2020.9053186.

William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 40(3):52–74, 2017a.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30*, pages 1024–1034, 2017b.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=SJU4ayYgl`.

Vignesh Kothapalli, Tom Tirer, and Joan Bruna. A neural collapse perspective on feature evolution in graph neural networks. In *Advances in Neural Information Processing Systems 36*, pages 74985–75016, 2023.

Sanjukta Krishnagopal and Luana Ruiz. Graph neural tangent kernel: Convergence on large graphs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17827–17841, 2023.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020. doi: 10.1073/pnas.2015509117.

Franziska Paul, Ya'ara Arkin, Amir Giladi, Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Deborah Winter, David Lara-Astiaso, Meital Gury, Assaf Weiner, Eyal David, Nadav Cohen, Felicia Kathrine Bratt Lauridsen, Simon Haas, Andreas Schlitzer, Alexander Mildner, Florent Ginhoux, Steffen Jung, Andreas Trumpp, Bo Torben Porse, Amos Tanay, and Ido Amit. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell*, 163(7): 1663–1677, 2015. doi: 10.1016/j.cell.2015.11.013.

Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Graph neural networks: Architectures, stability, and transferability. *Proceedings of the IEEE*, 109(5):660–682, 2021. doi: 10.1109/JPROC.2021.3055400.

Luana Ruiz, Luiz F. O. Chamon, and Alejandro Ribeiro. Transferability properties of graph neural networks. *IEEE Transactions on Signal Processing*, 71:3474–3489, 2023. doi: 10.1109/TSP.2023.3297848.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. doi: 10.1137/080734029.

Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011. doi: 10.1137/08074489X.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A Proofs

This appendix contains full proofs of the results stated in Section 2. We retain the notation introduced there.

A.1 Auxiliary bounds

We first record a standard identity for matrix powers.

Lemma 1 (Telescoping identity)**.** *For any square matrices A, B and any integer $r \geq 1$,*

$$A^r - B^r = \sum_{j=0}^{r-1} A^{r-1-j}(A-B)B^j.$$

Proof. The claim follows by induction on r. The case $r = 1$ is immediate. If it holds for r, then

$$A^{r+1} - B^{r+1} = A(A^r - B^r) + (A-B)B^r,$$

and substituting the induction hypothesis yields the result. □

We will also use the following consequence of spectral sparsification.

Lemma 2. *If $(1-\varepsilon)L \preceq \widetilde{L} \preceq (1+\varepsilon)L$ and $\|L\|_2 \leq B_L$, then*

$$\|L - \widetilde{L}\|_2 \leq \varepsilon B_L.$$

Proof. For any unit vector x,

$$-\varepsilon x^\top L x \leq x^\top(\widetilde{L} - L)x \leq \varepsilon x^\top L x.$$

Since $0 \preceq L$ and $\|L\|_2 \leq B_L$, we have $x^\top L x \leq B_L \|x\|_2^2 = B_L$. Hence

$$\left|x^\top(\widetilde{L} - L)x\right| \leq \varepsilon B_L$$

for every unit vector x. Because $\widetilde{L} - L$ is symmetric,

$$\|\widetilde{L} - L\|_2 = \sup_{\|x\|_2=1} \left|x^\top(\widetilde{L} - L)x\right| \leq \varepsilon B_L.$$

□

A.2 Proof of Theorem 1

Proof of Theorem 1. Let

$$p(x) = \sum_{r=0}^{d} a_r x^r.$$

Then

$$p(L) - p(\widetilde{L}) = \sum_{r=1}^{d} a_r (L^r - \widetilde{L}^r).$$

By Lemma 1,

$$L^r - \widetilde{L}^r = \sum_{j=0}^{r-1} L^{r-1-j}(L - \widetilde{L})\widetilde{L}^j.$$

Taking operator norms and using $\|L\|_2, \|\widetilde{L}\|_2 \leq B_L$,

$$\|L^r - \widetilde{L}^r\|_2 \leq \sum_{j=0}^{r-1} \|L\|_2^{r-1-j} \|L - \widetilde{L}\|_2 \|\widetilde{L}\|_2^j \leq r B_L^{r-1} \|L - \widetilde{L}\|_2.$$

Applying Lemma 2,

$$\|L^r - \widetilde{L}^r\|_2 \leq r B_L^r \varepsilon.$$

Therefore,

$$\|p(L) - p(\widetilde{L})\|_2 \leq \sum_{r=1}^{d} |a_r| \, \|L^r - \widetilde{L}^r\|_2 \leq \left(\sum_{r=1}^{d} |a_r| \, r \, B_L^r \right) \varepsilon.$$

This is exactly (4). □

A.3 Proof of Theorem 2

Proof of Theorem 2. For each layer define

$$A_k := p_k(L), \qquad \widetilde{A}_k := p_k(\widetilde{L}).$$

Since $\|L\|_2, \|\widetilde{L}\|_2 \le B_L$ and each p_k has fixed finite degree, there exist constants

$$M_k := \sup_{\|S\|_2 \le B_L} \|p_k(S)\|_2 < \infty$$

such that

$$\|A_k\|_2, \|\widetilde{A}_k\|_2 \le M_k.$$

We first record uniform forward bounds. Let $B_0 := B_X$. Since σ_k is Lipschitz,

$$\|H^{(k+1)}\|_F \le L_{\sigma_k}\|A_k H^{(k)} W_k\|_F + |\sigma_k(0)|\sqrt{n d_{k+1}}.$$

Using $\|A_k\|_2 \le M_k$ and $\|W_k\|_2 \le B_W$, we get

$$\|H^{(k+1)}\|_F \le L_{\sigma_k} M_k B_W \|H^{(k)}\|_F + |\sigma_k(0)|\sqrt{n d_{k+1}}.$$

The same bound holds for $\widetilde{H}^{(k+1)}$. Hence, by induction, there are constants B'_k independent of ε such that

$$\|H^{(k)}\|_F, \|\widetilde{H}^{(k)}\|_F \le B'_k \qquad \text{for all } k = 0, \dots, K.$$

By Theorem 1,

$$\|A_k - \widetilde{A}_k\|_2 \le C_{p_k}\varepsilon.$$

Let

$$\Delta_k := \|H^{(k)} - \widetilde{H}^{(k)}\|_F.$$

Since $H^{(0)} = \widetilde{H}^{(0)} = X$, we have $\Delta_0 = 0$.

Using the layer definitions, Lipschitz continuity of σ_k, and adding and subtracting $\widetilde{A}_k H^{(k)} W_k$, we obtain

$$\begin{aligned}\Delta_{k+1} &= \|\sigma_k(A_k H^{(k)} W_k) - \sigma_k(\widetilde{A}_k \widetilde{H}^{(k)} W_k)\|_F \\ &\le L_{\sigma_k}\Big(\|(A_k - \widetilde{A}_k)H^{(k)} W_k\|_F + \|\widetilde{A}_k (H^{(k)} - \widetilde{H}^{(k)}) W_k\|_F\Big). \end{aligned} \tag{13}$$

Applying the weight and operator bounds to (13) gives

$$\Delta_{k+1} \le L_{\sigma_k} B_W \Big(\|A_k - \widetilde{A}_k\|_2 \|H^{(k)}\|_F + \|\widetilde{A}_k\|_2 \Delta_k\Big).$$

Using Theorem 1 and the uniform forward bounds gives

$$\Delta_{k+1} \le \alpha_k \varepsilon + \beta_k \Delta_k,$$

where

$$\alpha_k := L_{\sigma_k} B_W C_{p_k} B'_k, \qquad \beta_k := L_{\sigma_k} B_W M_k.$$

Since $\Delta_0 = 0$, iterating this finite recursion gives

$$\Delta_K \le C_{\mathrm{rep}}\varepsilon,$$

where C_{rep} depends only on the stated network and boundedness constants, which proves (5). □

A.4 Proof of Theorem 3

Proof. Let $\Delta := Z - \widetilde{Z}$. Then

$$ZZ^\top - \widetilde{Z}\widetilde{Z}^\top = Z\Delta^\top + \Delta\widetilde{Z}^\top.$$

For the spectral norm,

$$\begin{aligned}\|ZZ^\top - \widetilde{Z}\widetilde{Z}^\top\|_2 &\le \|Z\Delta^\top\|_2 + \|\Delta\widetilde{Z}^\top\|_2 \\ &\le (\|Z\|_2 + \|\widetilde{Z}\|_2)\|\Delta\|_F.\end{aligned}$$

For the Frobenius norm,

$$\begin{aligned}\|ZZ^\top - \widetilde{Z}\widetilde{Z}^\top\|_F &\le \|Z\Delta^\top\|_F + \|\Delta\widetilde{Z}^\top\|_F \\ &\le (\|Z\|_2 + \|\widetilde{Z}\|_2)\|\Delta\|_F.\end{aligned}$$

By the boundedness argument used in the proof of Theorem 2, both $\|Z\|_2$ and $\|\widetilde{Z}\|_2$ are bounded independently of ε. Theorem 2 gives $\|\Delta\|_F \le C_{\mathrm{rep}}\varepsilon$. Therefore both claimed bounds follow. □

A.5 Proofs of Corollaries 1 and 2

Proof of Corollary 1. Recall that

$$\|z_i - z_j\|_2^2 = \langle z_i, z_i\rangle + \langle z_j, z_j\rangle - 2\langle z_i, z_j\rangle.$$

The same identity holds for $\widetilde{z}_i, \widetilde{z}_j$. Therefore,

$$\begin{aligned}\left|\|z_i - z_j\|_2^2 - \|\widetilde{z}_i - \widetilde{z}_j\|_2^2\right| \le |\langle z_i, z_i\rangle - \langle \widetilde{z}_i, \widetilde{z}_i\rangle| + |\langle z_j, z_j\rangle - \langle \widetilde{z}_j, \widetilde{z}_j\rangle| \\ + 2|\langle z_i, z_j\rangle - \langle \widetilde{z}_i, \widetilde{z}_j\rangle|.\end{aligned}$$

Each inner-product difference is an entry of the matrix $ZZ^\top - \widetilde{Z}\widetilde{Z}^\top$, hence is bounded by its operator norm. Applying Theorem 3 gives the result. □

Proof of Corollary 2. Let $\Delta := Z - \widetilde{Z}$, and let Δ_i denote its ith row. Then

$$\mu_c - \widetilde{\mu}_c = \frac{1}{n_c}\sum_{i\in S_c}\Delta_i.$$

By Cauchy–Schwarz,

$$\|\mu_c - \widetilde{\mu}_c\|_2 \le \frac{1}{n_c}\sum_{i\in S_c}\|\Delta_i\|_2 \le \frac{1}{\sqrt{n_c}}\left(\sum_{i\in S_c}\|\Delta_i\|_2^2\right)^{1/2} \le \frac{1}{\sqrt{n_c}}\|\Delta\|_F.$$

This proves (8).

For the covariance bound, define

$$a_i := z_i - \mu_c, \qquad \widetilde{a}_i := \widetilde{z}_i - \widetilde{\mu}_c.$$

Then

$$\Sigma_c - \widetilde{\Sigma}_c = \frac{1}{n_c}\sum_{i\in S_c}(a_i a_i^\top - \widetilde{a}_i\widetilde{a}_i^\top).$$

Using

$$a_i a_i^\top - \widetilde{a}_i\widetilde{a}_i^\top = a_i(a_i - \widetilde{a}_i)^\top + (a_i - \widetilde{a}_i)\widetilde{a}_i^\top,$$

we obtain

$$\|a_i a_i^\top - \widetilde{a}_i\widetilde{a}_i^\top\|_2 \le (\|a_i\|_2 + \|\widetilde{a}_i\|_2)\|a_i - \widetilde{a}_i\|_2.$$

If all row norms of Z and $\widetilde{Z}$ are bounded by B_Z, then

$$\|a_i\|_2, \|\widetilde{a}_i\|_2 \le 2B_Z.$$

Moreover,

$$\|a_i - \widetilde{a}_i\|_2 \le \|z_i - \widetilde{z}_i\|_2 + \|\mu_c - \widetilde{\mu}_c\|_2.$$

Therefore,

$$\|\Sigma_c - \widetilde{\Sigma}_c\|_2 \le \frac{4B_Z}{n_c}\sum_{i\in S_c}\left(\|z_i - \widetilde{z}_i\|_2 + \|\mu_c - \widetilde{\mu}_c\|_2\right).$$

Using Cauchy–Schwarz for the first term and the mean bound for the second term,

$$\frac{1}{n_c}\sum_{i\in S_c}\|z_i - \widetilde{z}_i\|_2 \le \frac{1}{\sqrt{n_c}}\|\Delta\|_F, \qquad \|\mu_c - \widetilde{\mu}_c\|_2 \le \frac{1}{\sqrt{n_c}}\|\Delta\|_F.$$

Combining these yields

$$\|\Sigma_c - \widetilde{\Sigma}_c\|_2 \le \frac{8B_Z}{\sqrt{n_c}}\|\Delta\|_F.$$

Finally, Theorem 2 gives $\|\Delta\|_F = O(\varepsilon)$. □

A.6 Proof of Theorem 4

We first establish two Lipschitz properties of the gradient.

Lemma 3 (Gradient Lipschitz in weights)**.** *Under (A1), (A5), (A6), and (A7), there exists $L_W > 0$ such that*

$$\|\nabla_W J(W; L) - \nabla_W J(V; L)\|_F \le L_W \|W - V\|_F$$

for all W, V in the bounded region $\{\|W\|_2, \|V\|_2 \le R\}$.

Proof. Let $S := p(L)$. The gradient can be written as

$$\nabla_W J(W; L) = X^\top S\Big((\sigma(SXW) - Y) \odot \sigma'(SXW)\Big).$$

This expression is a composition of affine maps, matrix products, and pointwise C^1 nonlinearities. The Lipschitz assumption on σ' ensures that these compositions have uniformly bounded first derivatives on the prescribed bounded parameter/state region. Under assumptions (A4)–(A8), all quantities appearing in the gradient remain in a bounded set, and on bounded sets both σ and σ' are Lipschitz. Hence the map $W \mapsto \nabla_W J(W; L)$ is locally Lipschitz on that bounded set. Therefore there exists $L_W > 0$ such that

$$\|\nabla_W J(W; L) - \nabla_W J(V; L)\|_F \le L_W \|W - V\|_F.$$

□

Lemma 4 (Gradient Lipschitz in the graph operator)**.** *Under (A1), (A4)–(A7), there exists $L_S > 0$ such that*

$$\|\nabla_W J(W; L) - \nabla_W J(W; \widetilde{L})\|_F \le L_S \|p(L) - p(\widetilde{L})\|_2$$

for all W with $\|W\|_2 \le R$.

Proof. Writing $S := p(L)$ and $\widetilde{S} := p(\widetilde{L})$, the same gradient formula shows that $\nabla_W J(W; \cdot)$ is locally Lipschitz in the propagation operator on bounded sets. Since W remains bounded and σ, σ' are bounded and Lipschitz on bounded sets, there exists $L_S > 0$ such that

$$\|\nabla_W J(W; L) - \nabla_W J(W; \widetilde{L})\|_F \le L_S \|p(L) - p(\widetilde{L})\|_2.$$

□

Proof of Theorem 4. Starting from the gradient descent updates,

$$W_{t+1} - \widetilde{W}_{t+1} = W_t - \widetilde{W}_t - \eta\Big(\nabla_W J(W_t; L) - \nabla_W J(\widetilde{W}_t; \widetilde{L})\Big).$$

Taking Frobenius norms and using the triangle inequality,

$$\|W_{t+1} - \widetilde{W}_{t+1}\|_F \le \|W_t - \widetilde{W}_t\|_F + \eta \|\nabla_W J(W_t; L) - \nabla_W J(\widetilde{W}_t; \widetilde{L})\|_F.$$

Split the gradient difference as

$$\begin{aligned}\nabla_W J(W_t; L) - \nabla_W J(\widetilde{W}_t; \widetilde{L}) = &\big(\nabla_W J(W_t; L) - \nabla_W J(\widetilde{W}_t; L)\big) \\ &+ \big(\nabla_W J(\widetilde{W}_t; L) - \nabla_W J(\widetilde{W}_t; \widetilde{L})\big).\end{aligned}$$

Applying Lemmas 3 and 4,

$$\|W_{t+1} - \widetilde{W}_{t+1}\|_F \le (1 + \eta L_W)\|W_t - \widetilde{W}_t\|_F + \eta L_S \|p(L) - p(\widetilde{L})\|_2.$$

This proves (10). Since $W_0 = \widetilde{W}_0$, unrolling the recursion yields

$$\|W_t - \widetilde{W}_t\|_F \le \eta L_S \|p(L) - p(\widetilde{L})\|_2 \sum_{j=0}^{t-1} (1 + \eta L_W)^j.$$

Finally, Theorem 1 implies

$$\|p(L) - p(\widetilde{L})\|_2 \le C_p \varepsilon,$$

which gives the claimed bound. □

A.7 Proof of Proposition 1

In this subsection we prove the multilayer analogue of Theorem 4. For a parameter tuple

$$\Theta = (W_0, \dots, W_{K-1}),$$

define the hidden states recursively by

$$\begin{aligned} H^{(0)} &= X, \\ H^{(k+1)} &= \sigma_k\big(S_k H^{(k)} W_k\big), \qquad S_k := p_k(L), \end{aligned} \tag{14}$$

and similarly, on the sparsified graph,

$$\begin{aligned} \widetilde{H}^{(0)} &= X, \\ \widetilde{H}^{(k+1)} &= \sigma_k\big(\widetilde{S}_k \widetilde{H}^{(k)} W_k\big), \qquad \widetilde{S}_k := p_k(\widetilde{L}). \end{aligned} \tag{15}$$

Let

$$F(\Theta; L) := H^{(K)}, \qquad J(\Theta; L) := \frac{1}{2}\|F(\Theta; L) - Y\|_F^2,$$

and equip parameter space with the product Frobenius norm

$$\|\Theta\|_F^2 := \sum_{k=0}^{K-1} \|W_k\|_F^2.$$

For each layer k, define the preactivation

$$U_k := S_k H^{(k)} W_k.$$

We also introduce the backpropagated state gradients

$$\Delta_K := H^{(K)} - Y,$$

and for $k = K-1, \dots, 0$,

$$\Delta_k = S_k^\top \Big(\big(\Delta_{k+1} \odot \sigma_k'(U_k)\big) W_k^\top \Big). \tag{16}$$

With this notation, the gradient with respect to W_k is

$$\nabla_{W_k} J(\Theta; L) = (H^{(k)})^\top S_k^\top \big(\Delta_{k+1} \odot \sigma_k'(U_k)\big). \tag{17}$$

We first prove two Lipschitz properties of the full gradient.

Lemma 5 (Gradient Lipschitz in the parameters)**.** *Under assumptions (A1)–(A5), (A6′), (A7′), and (A8) there exists a constant $L_\Theta > 0$ such that*

$$\|\nabla_\Theta J(\Theta; L) - \nabla_\Theta J(\Phi; L)\|_F \le L_\Theta \|\Theta - \Phi\|_F$$

for all parameter tuples Θ, Φ in the bounded region determined by (A7′).

Proof. From the forward recursion (14), the backward recursion (16), and the gradient formula (17), all forward states, backward states, and gradient blocks are compositions of affine maps, matrix products, and pointwise C^1 nonlinearities. Under assumptions (A1)–(A5), (A6′), (A7′), and (A8) these quantities remain in a bounded set, on which all such maps are Lipschitz. Therefore the full gradient map $\Theta \mapsto \nabla_\Theta J(\Theta; L)$ is locally Lipschitz on that bounded set, so there exists $L_\Theta > 0$ such that

$$\|\nabla_\Theta J(\Theta; L) - \nabla_\Theta J(\Phi; L)\|_F \le L_\Theta \|\Theta - \Phi\|_F.$$

□

Lemma 6 (Gradient Lipschitz in the graph filters)**.** *Under assumptions (A1)–(A5), (A6′), and (A7′), there exists a constant $L_S > 0$ such that*

$$\|\nabla_\Theta J(\Theta; L) - \nabla_\Theta J(\Theta; \widetilde{L})\|_F \le L_S \sum_{k=0}^{K-1} \|S_k - \widetilde{S}_k\|_2$$

for all parameter tuples Θ in the bounded region determined by (A7′).

Proof. Fix Θ. Using the same parameter tuple on both graphs, the only perturbation comes from replacing $\{S_k\}_{k=0}^{K-1}$ by $\{\widetilde{S}_k\}_{k=0}^{K-1}$. The forward states, backward states, and gradient blocks are again compositions of affine maps, matrix products, and pointwise C^1 nonlinearities, and remain bounded under (A1)–(A5), (A6′), and (A7′). Hence the full gradient is locally Lipschitz in the collection of graph filters on bounded sets. Therefore there exists $L_S > 0$ such that

$$\|\nabla_\Theta J(\Theta; L) - \nabla_\Theta J(\Theta; \widetilde{L})\|_F \le L_S \sum_{k=0}^{K-1} \|S_k - \widetilde{S}_k\|_2.$$

□

Proof of Proposition 1. This is the same perturbation argument as in Theorem 4, now applied in product parameter space and with the graph perturbation measured through the collection $\{p_k(L) - p_k(\widetilde{L})\}_{k=0}^{K-1}$.

Starting from the gradient descent updates,

$$\Theta^{t+1} - \widetilde{\Theta}^{t+1} = \Theta^t - \widetilde{\Theta}^t - \eta\Big(\nabla_\Theta J(\Theta^t; L) - \nabla_\Theta J(\widetilde{\Theta}^t; \widetilde{L})\Big). \tag{18}$$

Taking Frobenius norms and using the triangle inequality,

$$\|\Theta^{t+1} - \widetilde{\Theta}^{t+1}\|_F \le \|\Theta^t - \widetilde{\Theta}^t\|_F + \eta\|\nabla_\Theta J(\Theta^t; L) - \nabla_\Theta J(\widetilde{\Theta}^t; \widetilde{L})\|_F. \tag{19}$$

Split the gradient difference as

$$\begin{aligned}\nabla_\Theta J(\Theta^t; L) - \nabla_\Theta J(\widetilde{\Theta}^t; \widetilde{L}) = &\Big(\nabla_\Theta J(\Theta^t; L) - \nabla_\Theta J(\widetilde{\Theta}^t; L)\Big) \\ &+ \Big(\nabla_\Theta J(\widetilde{\Theta}^t; L) - \nabla_\Theta J(\widetilde{\Theta}^t; \widetilde{L})\Big).\end{aligned} \tag{20}$$

Applying Lemmas 5 and 6 gives

$$\|\Theta^{t+1} - \widetilde{\Theta}^{t+1}\|_F \le (1 + \eta L_\Theta)\|\Theta^t - \widetilde{\Theta}^t\|_F + \eta L_S \sum_{k=0}^{K-1} \|S_k - \widetilde{S}_k\|_2.$$

This proves (12).

Since $\Theta^0 = \widetilde{\Theta}^0$, unrolling the recursion yields

$$\|\Theta^t - \widetilde{\Theta}^t\|_F \le \eta L_S \left(\sum_{k=0}^{K-1} \|S_k - \widetilde{S}_k\|_2\right) \sum_{j=0}^{t-1} (1 + \eta L_\Theta)^j. \tag{21}$$

Finally, by Theorem 1, for each k,

$$\|S_k - \widetilde{S}_k\|_2 = \|p_k(L) - p_k(\widetilde{L})\|_2 \le C_{p_k}\varepsilon.$$

Therefore

$$\sum_{k=0}^{K-1} \|S_k - \widetilde{S}_k\|_2 \le \left(\sum_{k=0}^{K-1} C_{p_k}\right)\varepsilon,$$

Substituting this estimate into Eqn. 21 gives

$$\|\Theta^t - \widetilde{\Theta}^t\|_F \le \eta L_S \left(\sum_{k=0}^{K-1} C_{p_k}\right)\varepsilon \sum_{j=0}^{t-1} (1 + \eta L_\Theta)^j,$$

which proves the claimed bound with

$$C_{\text{train}}^{(K)} = \eta L_S \sum_{k=0}^{K-1} C_{p_k}.$$

□

B Additional Experimental Details

B.1 Dataset construction

Synthetic SBM. We generate a four-block weighted stochastic block model with block sizes $(80, 80, 80, 80)$, within-block edge probability 0.20, and cross-block edge probability 0.05.

Synthetic geometric graph. We generate four Gaussian feature clusters with 80 nodes per class and 20-dimensional features. A weighted k-NN graph with $k = 30$ is built from these features using Gaussian kernel weights.

FashionMNIST. We load the training split, subsample 220 examples per class in the full setting, flatten the images, reduce to 50 PCA dimensions, and build a weighted 15-NN graph.

Cora. We use the Planetoid Cora benchmark and take the citation graph as the adjacency.

Paul15. We use the Paul15 single-cell dataset from Scanpy. Labels are extracted from an available cluster annotation, features are reduced to at most 64 dimensions when needed, and a weighted 15-NN graph is built from expression features.

Splits. Class-balanced random masks are used throughout. For FashionMNIST, we use 20 train and 40 validation examples per class. For Cora, we use 20 train and 50 validation examples per class. For Paul15, we use 20 train and 30 validation examples per class. Remaining nodes are used for testing. On synthetic datasets, analogous class-balanced masks are used.

B.2 Architectures and optimization

Synthetic theory-facing models. For the controlled sparsification experiments of Figure 1, we use a fixed deterministic two-stage polynomial-filter map so that the forward distortion induced by sparsification can be measured without confounding from stochastic optimization. We use the scaled combinatorial Laplacian

$$\bar{L} = L/\|L\|_2$$

as the graph operator, and apply the same dense-graph scaling to the sparsified Laplacian. The polynomial filter is

$$p(\bar{L}) = I - 0.6\bar{L} + 0.15\bar{L}^2.$$

The map consists of this polynomial graph filter followed by two deterministic feature-mixing stages.

For the training-dynamics experiments of Figure 2, we use the same polynomial-Laplacian graph shift. Dense and sparse trajectories differ only through replacing $p(\bar{L})$ by $p(\widetilde{\bar{L}})$, with matched initialization.

For the training-dynamics plots, the number of training epochs is 100. The learning rates are 0.01 for the one-layer model and 0.003 for the two-layer model, with weight decay 10^{-3} and gradient clipping.

Teacher models. For the synthetic practical experiments, the teacher is a deterministic two-layer GCN trained on the dense graph. The hidden dimension is 48 for the SBM and 32 for the geometric graph. Teachers are trained for 120 epochs with Adam, learning rate 0.01, and weight decay 5×10^{-4}.

For the real-data experiments, we use a two-layer GCN with ReLU activation and no dropout. When the input feature dimension is at most 64, we use hidden dimension 48, which is the case for the datasets considered here. In the full runs, teachers are trained for 20 epochs with Adam, learning rate 0.01, and weight decay 5×10^{-4}.

B.3 Sparsification details

Synthetic graphs. In the controlled synthetic experiments, we use exact effective-resistance sampling on the weighted SBM and geometric graphs. The sampling budget is the number of edge samples drawn with replacement before reweighting. We parameterize it as

$$q = \lfloor c\, n \log n \rfloor,$$

where larger c draws more samples and typically yields a denser, lower-distortion sparsifier. Rather than reporting only a fixed budget sweep, we search over a grid of budget multipliers c and, for each prescribed target level, select a sparsified graph whose realized empirical distortion ϵ_{emp} is closest to that target. Each selected target level is then repeated with multiple sparsifier draws. In the base setting we use 5 repetitions per selected target level.

For each sparsified graph $\widetilde{G}$, we compute the empirical graph distortion ϵ_{emp} from the generalized eigenvalue envelope of the dense and sparse Laplacians on the orthogonal complement of the all-ones vector. Specifically, if

$$\lambda_{\min} \leq \frac{x^\top \widetilde{L} x}{x^\top L x} \leq \lambda_{\max} \qquad \text{for all } x \perp \mathbf{1},$$

then we report

$$\epsilon_{\mathrm{emp}} := \max\{1 - \lambda_{\min},\, \lambda_{\max} - 1\}.$$

Real datasets. For real-data experiments, we use approximate effective-resistance sampling, where resistance scores are estimated using a truncated Laplacian eigenspace. In the full runs, the approximation rank is 16, the budget multipliers are

$$c \in \{0.45, 0.60, 0.80, 1.00, 1.25, 1.50\}, \tag{22}$$

and each budget is repeated 3 times with different sparsifier seeds. The empirical graph distortion is approximated by probing quadratic-form distortion with random test vectors.

B.4 Metric definitions

On synthetic graphs, we report filter error, relative hidden representation error, and relative hidden Gram error. On real data, the primary geometry metric is relative hidden Gram distortion,

$$D_{\mathrm{Gram}}(Z, \widetilde{Z}) = \frac{\|ZZ^\top - \widetilde{Z}\widetilde{Z}^\top\|_F}{\|ZZ^\top\|_F}. \tag{23}$$

To measure local geometry preservation, we compute mean 20-nearest-neighbor overlap on held-out nodes.

Let $z_i^\top$ denote the ith row of Z, i.e., the hidden embedding of node i. We define the similarity between nodes i and j by the cosine similarity of their hidden embeddings,

$$\cos(z_i, z_j) := \frac{\langle z_i, z_j \rangle}{\|z_i\|_2 \|z_j\|_2}.$$

Let $N_k(i; Z)$ denote the set of the k nodes with largest cosine similarity to node i in the embedding Z. The nearest neighbor overlap is given by

$$D_{\mathrm{NN@}k}(Z, \widetilde{Z}) = \frac{1}{|S|} \sum_{i \in S} \frac{|N_k(i; Z) \cap N_k(i; \widetilde{Z})|}{k}, \tag{24}$$

with $k = 20$ in all reported experiments. In the real-data script, these geometry metrics are evaluated on a random subsample of at most 500 held-out test nodes to keep the computation lightweight.

Compute resources. All experiments were run on an HP Spectre x360 laptop with an Intel Core i7-10750H CPU, 16 GB RAM, and an NVIDIA GeForce GTX 1650 Ti with Max-Q Design GPU (4 GB GDDR6 VRAM). In practice, the provided code ran primarily on CPU, with approximate runtimes of 5–15 minutes per real dataset and about 20–40 minutes for the full synthetic suite.

B.5 Additional qualitative and downstream figures

Figure 5 provides a qualitative view of the dense and sparse hidden embeddings after projection to two principal components and alignment by an optimal orthogonal Procrustes transform. The substantial overlap after alignment indicates that much of the dense–sparse discrepancy is explained by an approximately rigid transformation rather than severe geometric collapse.

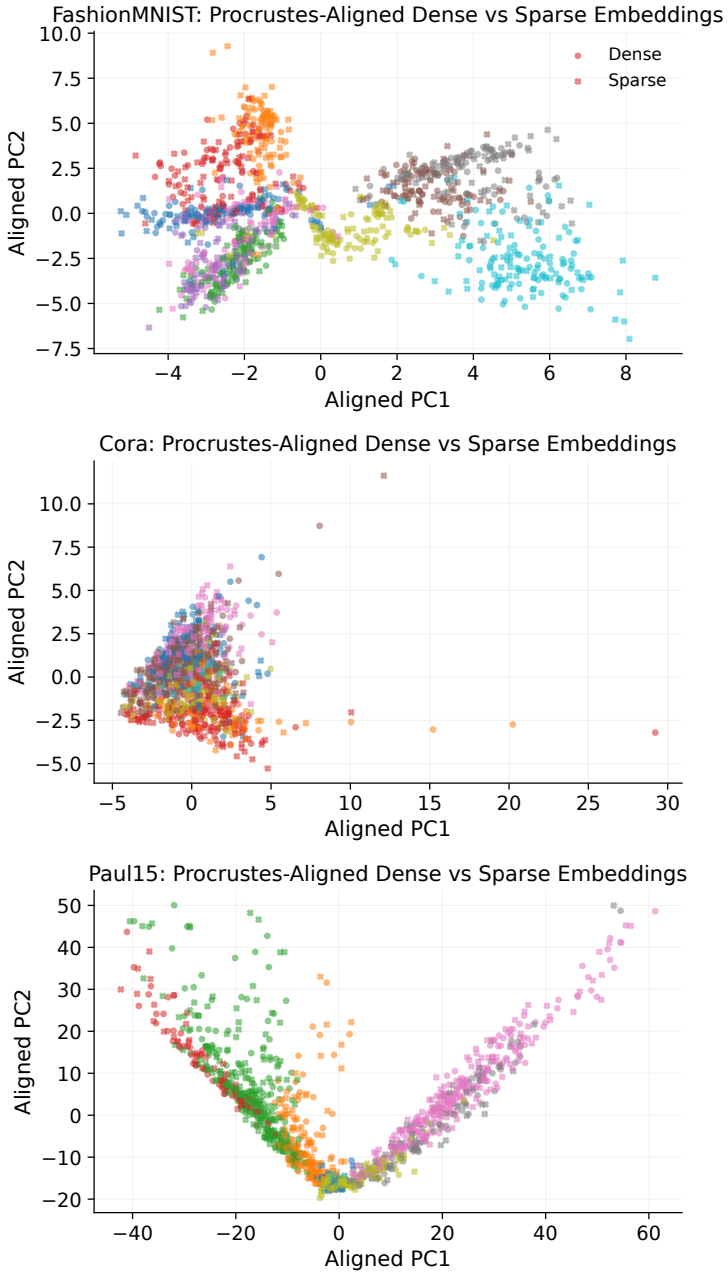

Figure 5: **Procrustes-aligned dense and sparse embeddings.** Dense and sparse hidden embeddings on held-out nodes after projection to two principal components and optimal orthogonal alignment of the sparse embedding to the dense one. The substantial overlap after alignment indicates that much of the dense–sparse discrepancy is explained by an approximately rigid transformation rather than severe deformation of the learned representation geometry.

B.6 Knowledge distillation under graph distortion

This experiment provides a downstream diagnostic complementary to the representation-geometry metrics in the main text. Here, we study teacher–student transfer on the synthetic graphs. A teacher is first trained on the dense graph. We then sparsify the graph and train a student on the sparse graph using a mixture of cross-entropy loss and KL-based knowledge distillation from the dense teacher. Figure 6 shows that larger empirical graph distortion leads to larger KL divergence in teacher behavior, and that larger teacher KL is associated with worse KD student accuracy. This provides a downstream interpretation of representation distortion: graph perturbations alter the soft predictive information available for transfer.

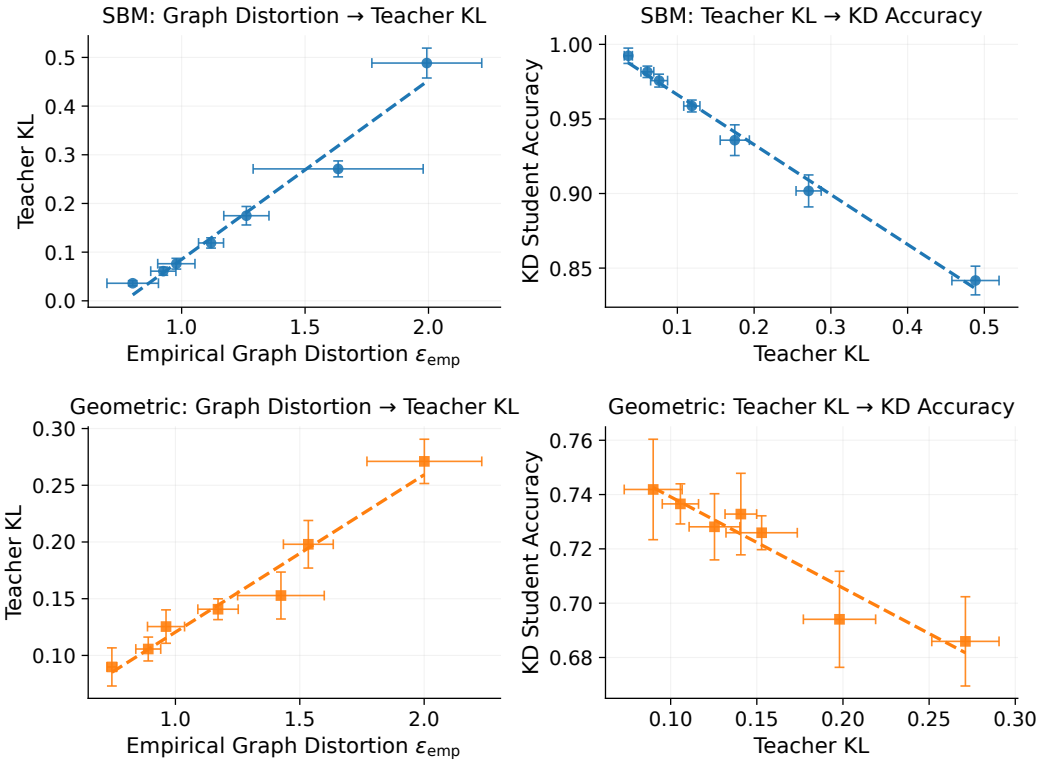

Figure 6: **Teacher–student consequences of graph distortion.** On synthetic SBM and geometric graphs, larger empirical graph distortion produces larger teacher KL divergence, and larger KL is associated with lower KD student accuracy. This shows that graph distortion affects not only hidden geometry but also the transferred soft decision structure. Error bars denote one standard deviation across four repeated runs.

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 abstract and introduction accurately state the paper's main theoretical and empirical contributions: stability of polynomial graph filters, multilayer hidden representations, and Gram matrices under spectral sparsification; finite-time training-dynamics stability in simplified settings; and empirical evidence that representation-geometry preservation tracks practical embedding fidelity on synthetic and real datasets.

   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: The paper includes a dedicated limitations discussion in Section 4. It notes that the theory is proved for polynomial-filter GNNs under boundedness, Lipschitz, and smoothness assumptions; that training-dynamics guarantees are established only in simplified deterministic settings; that experiments instantiate the framework using effective-resistance-based sparsification as a concrete testbed; and that the empirical study does not yet cover a broader range of architectures, larger-scale benchmarks, alternative sparsifiers, or highly compressed regimes.

   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 assumptions are stated explicitly in Section 2, including (A1)–(A8) and the multilayer variants (A6′)–(A7′), and complete proofs are provided in Appendix A for Theorems 1–4, Proposition 1, Corollaries 1–2, and the auxiliary lemmas they rely on.

   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: Sections 3.1–3.2 and Appendix B specify the datasets, graph constructions, train/validation/test splits, model classes, perturbation levels, sparsification budgets, metrics, optimization settings, and repetition counts used for the reported experiments.

   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: The paper provides detailed experimental specifications and a public code repository at `https://github.com/chimeraki/Spectral-Sparsification-Preserves-Representation-Geometry` with code for reproducing the reported experiments. Public datasets used in the experiments are standard benchmark datasets.

   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: The experimental setting is described in Section 3 and Appendix B, including dataset construction, data splits, hidden dimensions, optimizers, learning rates, training epochs, sparsification budgets, and evaluation metrics.

   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: The paper reports repeated runs and error bars where appropriate. In the full synthetic setting, the controlled perturbation experiments use 5 repetitions per perturbation level, and the synthetic knowledge-distillation experiment uses 4 repeated runs. In the full real-data setting, each sparsification budget is repeated 3 times with different sparsifier seeds. Reported error bars correspond to one standard deviation across repeated runs at fixed perturbation level or sparsification budget.

   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: Appendix B reports the hardware used for all experiments (HP Spectre x360 laptop, Intel Core i7-10750H CPU, 16 GB RAM, NVIDIA GeForce GTX 1650 Ti with Max-Q Design GPU, 4 GB GDDR6 VRAM), notes that the code ran primarily on CPU in practice, and provides order-of-magnitude runtime information for the real-data and synthetic experiment suites.

   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: The work is theoretical and empirical on synthetic and standard benchmark datasets, does not involve human subjects, and does not introduce a deployment system or other high-risk application.

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: The paper includes a broader-impacts paragraph discussing both potential benefits, such as improved efficiency and accessibility of graph-based learning methods, and possible risks if such methods are later deployed in sensitive decision-making settings without sufficient fairness, robustness, or privacy evaluation.

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 does not release a high-risk model, scraped dataset, or other asset with an elevated misuse profile; its core contribution is theoretical and experimental analysis.

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: The paper cites the original benchmark datasets, papers, and prior methods used in the experiments. The datasets employed are standard public research assets.

    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 submission does not introduce a new dataset, benchmark, pretrained model, or standalone released asset as a core contribution.

    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 or 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 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: LLMs are not part of the paper's core scientific methodology. Any LLM use was limited to writing, editing, formatting, and coding assistance rather than method development or experimental decision-making.

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.