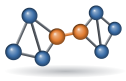
Distilling Graph Geometry: Knowledge Gap from GNNs to MLPs



Zhewei Chen[1] **Hao Zhu**[2*] **Jiaojiao Jiang**[3] **Ahad N. Zehmakan**[1*]
[1]Australian National University [2]CSIRO Technology [3]University of New South Wales
{zhewei.chen, ahadn.zehmakan}@anu.edu.au
allenhaozhu@gmail.com jiaojiao.jiang@unsw.edu.au

Abstract

GNN-to-MLP distillation aims to retain the predictive accuracy of a message-passing teacher while deploying a graph-free MLP at inference. Existing methods mainly transfer node-wise predictions or use confidence-based reweighting, but they do not specify where the student should preserve the teacher's graph-induced geometry. We show that this omission leads to two spectral failure modes in the student's representation space. On sparse graphs, the student suffers from *spectral underfit*, missing high-energy teacher directions induced by aggregation. On dense graphs, it suffers from *spectral overfit*, retaining spurious directions that the teacher has collapsed through aggregation. Motivated by an energy-weighted teacher–student alignment objective, we propose **Graph Geometry-aware MLP (G^2MLP)**, a training-time distillation framework guided by Ollivier–Ricci curvature. Curvature acts as a local proxy for allocating supervision between prediction-level and representation-level alignment. The deployed model is a standard MLP with no graph access at inference. Across node-classification benchmarks, G^2MLP improves over graph-free distillation baselines in most settings, reduces the teacher–student rank gap in both regimes, and transfers to Graph Transformer teachers and link prediction.

1 Introduction

Graph Neural Networks (GNNs) [Gilmer et al., 2017] are the standard tool for learning on graphs, but their reliance on message passing makes inference slow in latency-sensitive applications such as fraud detection [Wang et al., 2021]. A growing line of work distils GNN knowledge into a structure-free Multi-Layer Perceptron (MLP), enabling fast graph-free inference [Zhang et al., 2022, Guo et al., 2023, Tian et al., 2023]. Most methods follow GLNN [Zhang et al., 2022] and match teacher logits with a KL loss, sometimes injecting structural priors through positional encodings [Tian et al., 2023, Wang et al., 2023], codebooks [Yang et al., 2024], or neighborhood alignment cues [Hu et al., 2021, Wu et al., 2023a]. These methods improve the student's information source but treat graph structure as uniformly useful: every node receives the same distillation pressure regardless of topology.

We argue that this uniform view misses an important source of teacher–student mismatch. Message passing changes representations differently across graph regions. Boundary nodes often contain graph-induced directions that are hard to infer from features alone, while dense interior nodes are often smoothed toward class-level prototypes. As shown in Figure 1, this produces two spectral failure modes in GNN-to-MLP distillation. On sparse graphs, the student has lower effective rank than the teacher, which we call *spectral underfit*. On dense graphs, the teacher may collapse to a low-rank geometry, while the feature-only student retains unsupported variation, which we call *spectral overfit*.

*Corresponding authors.

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

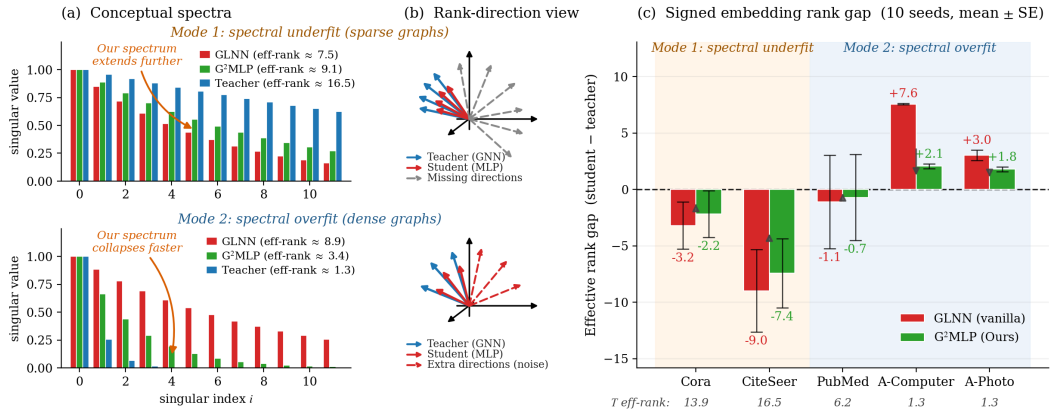


Figure 1: **Two spectral failure modes of GNN-to-MLP distillation.** **(a)** Singular-value spectra: GLNN exhibits *spectral underfit* on sparse graphs and *spectral overfit* on dense graphs. **(b)** Rank-direction sketch. **(c)** Per-dataset rank gap; bands mark the dominant mismatch channel (Table 8).

Our analysis in Section 3 formalizes this phenomenon through the mismatch between the GNN kernel and the MLP kernel. The resulting energy-weighted alignment objective $\mathcal{L}^*$ explains why pointwise KL distillation is insufficient: two students can match the same logits while inducing different representation geometries, so a practical surrogate should allocate supervision where the spectral error concentrates. Ollivier–Ricci curvature (ORC) provides such a local signal: low-curvature edges mark boundary regions where random-walk neighborhoods disagree, while high-curvature edges mark dense regions where aggregation strongly smooths representations.

We instantiate this principle in **Graph Geometry-aware MLP (G^{2}MLP)**. G^{2}MLP uses ORC to approximate this principled alignment: low-curvature nodes receive stronger prediction-level supervision to correct spectral underfit, while high-curvature nodes receive stronger representation-level neighborhood alignment to reduce spectral overfit. The representation term uses a closed-form diagonal Gaussian $\mathcal{W}_2$ surrogate. All graph-dependent quantities are used only during training, so the deployed model remains a vanilla MLP. Our contributions are:

- We identify spectral underfit and spectral overfit as two failure modes of GNN-to-MLP distillation, and connect them to boundary and interior graph regions.
- We derive an energy-weighted alignment objective $\mathcal{L}^*$ that characterizes teacher–student geometric mismatch and motivates curvature-dependent allocation of distillation supervision.
- We propose G^{2}MLP, a curvature-guided distillation framework that combines prediction- and representation-level alignment while preserving graph-free MLP inference.

2 Related Work

GNN-to-MLP knowledge distillation. GLNN [Zhang et al., 2022] established the paradigm of distilling a GNN teacher into an MLP student through soft-label KL [Hinton et al., 2015], eliminating graph access at inference. Follow-up work enriches the student's input or supervision target: positional encodings (NOSMOG [Tian et al., 2023]), structural codebooks (VQGraph [Yang et al., 2024]), neighborhood alignment (FF-G2M [Wu et al., 2023a]), node-pair grouping (MSN-GDM [Dong et al., 2025]), and hardness-aware reweighting (KRD [Wu et al., 2023b], HGMD [Wu et al., 2024], AdaGMLP [Lu et al., 2024]); SimMLP [Wang et al., 2025] uses a self-supervised alignment target. These methods either treat nodes uniformly or derive node weights from teacher confidence; none ties reweighting to graph topology or to a named failure mode of uniform KD.

Optimal transport for distillation. WCoRD [Chen et al., 2021] and EMD+IPOT [Lohit and Jones, 2022] transfer knowledge via discrete Wasserstein distances between batch features through cross-instance matching. WKD [Lv et al., 2024] introduces continuous Gaussian $\mathcal{W}_2$ as a Riemannian metric for feature distillation, sharper than geometry-unaware KL [Peyré and Cuturi, 2019]. We extend WKD's Gaussian-$\mathcal{W}_2$ template to the graph setting, fitting it per node over neighborhood representations and weighting it by curvature.

Curvature on graphs. Ollivier–Ricci curvature (ORC) [Ollivier, 2009] measures the $\mathcal{W}_1$ distance between neighborhood walks of adjacent nodes, and has been used for community detection [Sia et al., 2019] and information-flow analysis [Topping et al., 2022]. In GNNs, Nguyen et al. [2023] link positive curvature to over-smoothing and negative curvature to over-squashing, motivating rewiring (BORF) and dropout (CurvDrop [Liu et al., 2023]) schemes; Hevapathige et al. [2025] adapt depth via learnable Bakry–Émery curvature. These works modify the GNN itself; we instead use curvature to stratify distillation pressure.

3 Why Uniform Distillation Fails under Curvature Heterogeneity

We develop a kernel-motivated analysis of GNN-to-MLP distillation. Our goal is not to derive an exact computable form of the GNTK, but to identify which parts of the teacher geometry a graph-free student tends to miss and how graph curvature can guide a tractable surrogate. The argument proceeds in three steps: the teacher kernel contains graph-dependent aggregation terms absent from the MLP kernel; this mismatch appears as two opposite spectral failure modes; and ORC provides a local graph signal for allocating supervision where these modes concentrate. The result is an energy-weighted target objective $\mathcal{L}^*$ and a curvature-guided approximation developed in Section 4.

Teacher and Student Kernel Mismatch. NTK theory [Jacot et al., 2018] characterizes infinitely wide networks trained by gradient flow: at convergence, the network output lies in the function space induced by a fixed kernel, the network's NTK. For a K-layer GNN with feature aggregation $\mathcal{A}$ and activation σ, the GNTK $\mathbf{K}^{(K)}$ admits a recursion of the schematic form

$$\mathbf{K}^{(k)}(v,u) = \underbrace{\mathcal{A}\Big[\Sigma^{(k-1)}\Big](v,u)}_{\text{aggregation over } (\mathcal{N}(v),\mathcal{N}(u))} + \dot{\Sigma}^{(k)}(v,u)\,\mathcal{A}\Big[\mathbf{K}^{(k-1)}\Big](v,u), \tag{1}$$

where $\Sigma^{(k)}$ is the post-activation covariance and $\dot{\Sigma}^{(k)}$ its derivative kernel; the full recursion follows Du et al. [2019] and is given in Appendix B. Here $\mathcal{G}$ denotes the input graph, $\mathbf{X}$ the node feature matrix, and $\mathcal{N}(v)$ the 1-hop neighborhood of v. The aggregation term sums covariance over neighborhood pairs, so $\mathbf{K}^{(K)}$ depends jointly on $(\mathbf{X}, \mathcal{G})$. The MLP student's NTK $\mathbf{K}^{\mathrm{MLP}}$ is the aggregation-free counterpart and depends only on $\mathbf{X}$. Thus $\Delta^{(K)} = \mathbf{K}^{(K)} - \mathbf{K}^{\mathrm{MLP}}$ captures the graph-dependent component introduced by message passing. The following statement is used qualitatively to motivate local surrogates; we do not claim that the full GNTK is sparse or that curvature exactly estimates its eigenspectrum.

Lemma 3.1 (Aggregation Locality)**.** *For a K-layer message-passing GNN with Lipschitz activation, the graph-dependent aggregation component of $\mathbf{K}^{(K)}$ is generated through paths contained in K-hop receptive fields. Hence the graph-induced contribution associated with a node pair (v,u) depends on the overlap and feature similarity of their K-hop neighborhoods. The full kernel need not be sparse, since feature covariance may remain nonzero for distant nodes.*

3.1 Two Failure Modes and the Ideal Distillation Objective

Spectral distortion. We characterize the student's coverage of the teacher's geometry in the teacher's eigenbasis. Let $\mathbf{K}^{(K)} = \sum_i \lambda_i^K \mathbf{u}_i \mathbf{u}_i^\top$, with $\lambda_1^K \geq \cdots \geq \lambda_N^K \geq 0$. For each direction with $\lambda_i^K > 0$, define

$$\rho_i = \frac{\mathbf{u}_i^\top \mathbf{K}^{\mathrm{MLP}} \mathbf{u}_i}{\lambda_i^K}. \tag{2}$$

When $\rho_i < 1$, the student has insufficient energy along a teacher direction, which we call *spectral underfit*; when $\rho_i > 1$, the student retains energy in directions that the teacher kernel weakly supports, which we call *spectral overfit*. We summarize the diagonal mismatch in the teacher eigenbasis by

$$\mathcal{D} = \sum_i (\rho_i - 1)^2 \lambda_i^K. \tag{3}$$

This quantity does not compare all off-diagonal kernel entries. It measures how much MLP energy differs from teacher energy along teacher eigendirections, with high-energy teacher directions receiving larger weight. Understanding where these discrepancies arise is therefore key to allocating distillation effort.

Define Ollivier–Ricci curvature as $\kappa(i,j) = 1 - W_1(\mu_i, \mu_j)$, where μ_v is the lazy random-walk distribution on the closed neighborhood $\mathcal{N}^+(v)$. Let $\bar{\kappa}_v$ average $\kappa(v,u)$ over $u \in \mathcal{N}(v)$, and let $\widetilde{\kappa}_v = (\bar{\kappa}_v - \mu_\kappa)/\sigma_\kappa$. For a Lipschitz feature map ϕ, define the neighborhood embedding $\bar{\mathbf{x}}_v = \sum_u \mu_v(u)\phi(\mathbf{x}_u)$. The following proposition links ORC to local aggregation disagreement under explicit metric assumptions.

Proposition 3.2 (ORC as a Local Mismatch Proxy)**.** *Suppose ϕ is L-Lipschitz with respect to the ground metric used in W_1. Then for any edge (i,j),*

$$\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2 \le L\, W_1(\mu_i, \mu_j) = L(1 - \kappa(i,j)). \tag{4}$$

Suppose in addition that on a set of edges $\mathcal{E}' \subseteq \mathcal{E}$, with constants $\eta \in (0,1]$ and $m > 0$, (C1) *the displacements transported by an optimal coupling $\pi^\star$ of (μ_i, μ_j) do not cancel, $\|\mathbb{E}_{\pi^\star}[\phi(\mathbf{x}_X) - \phi(\mathbf{x}_Y)]\|_2 \ge \eta\, \mathbb{E}_{\pi^\star}\|\phi(\mathbf{x}_X) - \phi(\mathbf{x}_Y)\|_2$, and* (C2) *$\phi$ is lower-Lipschitz on the transported pairs, $\|\phi(\mathbf{x}_a) - \phi(\mathbf{x}_b)\|_2 \ge m\, d_{\mathcal{G}}(a,b)$ whenever $\pi^\star(a,b) > 0$. Then for every $(i,j) \in \mathcal{E}'$,*

$$m\eta\,(1 - \kappa(i,j)) \le \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2 \le L\,(1 - \kappa(i,j)). \tag{5}$$

Proposition 3.2 makes ORC a locally computable proxy for where graph-dependent teacher information is likely to matter, without requiring construction or diagonalization of $\Delta^{(K)}$. This motivates the curvature-stratified weights in Section 4.3.

Why pointwise KL is insufficient. The standard distillation loss $\mathcal{L}_{\mathrm{KD}} = \frac{1}{|\mathcal{V}|}\sum_v D_{\mathrm{KL}}(\hat{\mathbf{y}}_v^T \| \hat{\mathbf{y}}_v^S)$ sums node-wise divergences that depend on the teacher's marginal predictive distribution. Two students may match the same teacher logits while inducing different neighborhood geometries in hidden space. Consequently, pointwise KL does not directly control the graph-induced representation geometry encoded by $\mathbf{K}^{(K)}$. Reducing spectral distortion therefore requires an additional objective sensitive to neighborhood-level teacher-student alignment.

The target alignment loss. A natural energy-weighted target is

$$\mathcal{L}^* = \|\mathbf{f}^T - \mathbf{f}^S\|^2_{\mathbf{K}^{(K)}} = \sum_i \lambda_i^K \left(\hat{f}_i^T - \hat{f}_i^S\right)^2, \tag{6}$$

where $\hat{f}_i^T = \mathbf{u}_i^\top \mathbf{f}^T$ and $\|\mathbf{f}\|^2_{\mathbf{K}^{(K)}} = \mathbf{f}^\top \mathbf{K}^{(K)} \mathbf{f}$ denotes the finite-dimensional teacher-kernel energy form, not the RKHS norm. The weights λ_i^K emphasize teacher directions with high functional energy. In spectral underfit, this prioritizes teacher directions that the MLP does not recover; in spectral overfit, it downweights near-null teacher directions where the student retains unsupported variation. Directly using $\mathcal{L}^*$ is impractical because it requires storing and diagonalizing the full teacher kernel. We therefore use it as a principled target and approximate it with local, curvature-guided alignment in Section 4.

3.2 Curvature Stratifies the Failure Modes

Mode-conditional failure structure. Proposition 3.2 indicates that curvature separates locally incoherent boundary neighborhoods from coherent interior neighborhoods. Let $\mathbf{H}^T, \mathbf{H}^S \in \mathbb{R}^{N \times d_h}$ denote penultimate-layer representations, and define the demeaned effective rank $\mathrm{erank}(\mathbf{H}) = \exp(-\sum_i p_i \log p_i)$, where $p_i \propto \sigma_i^2(\mathbf{H} - \bar{\mathbf{H}})$. The rank gap $\Delta r = \mathrm{erank}(\mathbf{H}^S) - \mathrm{erank}(\mathbf{H}^T)$ reduces each learned representation's spectrum to one number and ignores its directions, so it is used as an observable diagnostic of representation geometry, not as an exact estimator of the teacher-kernel spectrum. Writing $\Delta_{ij}\bar{\mathbf{x}} = \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j$ for the neighborhood contrast of an edge, the two regimes read as follows.

- **Spectral underfit ($\Delta r < 0$).** On sparse graphs with heterogeneous curvature, aggregation creates teacher directions that the feature-only student does not recover. Take ϕ in Proposition 3.2 to be the teacher's prediction map. If the student under-responds at an edge, $\|\Delta_{ij}\bar{\mathbf{z}}^S\|_2 \le (1-\zeta)\|\Delta_{ij}\bar{\mathbf{z}}^T\|_2$ with $\zeta \in (0,1]$, the reverse triangle inequality and Eq. (5) give a prediction-contrast mismatch of at least $\zeta m\eta(1 - \kappa(i,j))$; for ζ, m and η common to the edges compared, this guarantee is largest on low-curvature edges.
- **Spectral overfit ($\Delta r > 0$).** On dense graphs with relatively homogeneous curvature, aggregation collapses teacher representations toward low-rank class prototypes. Take ϕ

to be the teacher's representation map. If the feature-only student retains a neighborhood representation contrast $c_{ij} = \|\Delta_{ij}\bar{\mathbf{h}}^S\|_2$, measured on a scale common with the teacher, that does not shrink with curvature, the upper bound in Eq. (4) leaves an excess of at least $c_{ij} - L(1-\kappa(i,j))$, which grows toward high curvature.

Empirical localization. Appendix F.1 attributes the spectral mismatch to individual nodes through the per-direction ratio ρ_i of demeaned, trace-normalized representations, which splits it into an underfit part ($\rho_i < 1$) and an overfit part ($\rho_i > 1$), and bins nodes by ORC quintile. For the vanilla-KL student, overfit energy is higher in the highest than in the lowest quintile on all five benchmarks (Figure 2; channel totals in Table 8), whereas underfit energy is spread across quintiles rather than concentrated at low curvature; at the edge level, the prediction mismatch across lowest-curvature edges exceeds that across highest-curvature edges on the underfit-dominated graphs. Which mode dominates is thus a graph-level property, while curvature localizes the overfit channel within a graph.

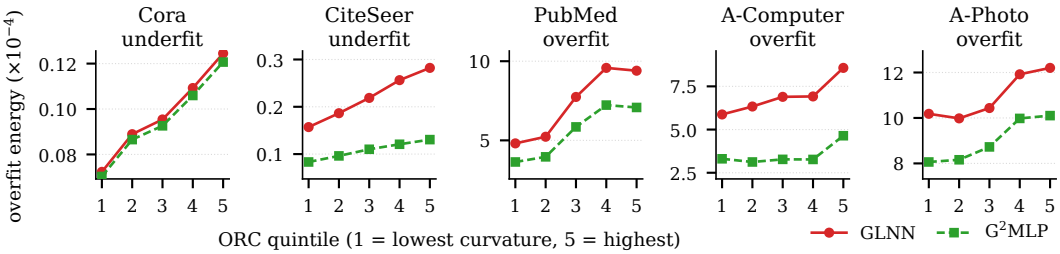


Figure 2: Mean overfit energy per ORC quintile of the nodes for the vanilla-KL student (GLNN) and G^{2}MLP (SAGE teacher; Appendix F.1). Every panel plots the overfit channel; the label under each dataset name is the graph's dominant mismatch channel (Table 8).

Implication for the loss design. The two regimes call for corrections at opposite ends of the curvature spectrum: under underfit, the guaranteed prediction mismatch is largest on low-curvature edges; under overfit, the guaranteed excess representation variation grows toward high curvature. A uniformly weighted loss does not distinguish them, since pressure on boundary-discriminative directions is also pressure on already-smoothed interior regions. This motivates a curvature-stratified objective in which prediction-side and representation-side alignment receive opposite curvature-dependent emphasis, as developed in Section 4.

4 G^{2}MLP: Curvature-Guided Graph Wasserstein Alignment

The target objective $\mathcal{L}^*$ in Eq. (6) is computationally intractable because it requires the full teacher kernel spectrum. We approximate it through three practical relaxations: global kernel alignment is replaced by local neighborhood distribution matching, discrete optimal transport is replaced by a diagonal Gaussian $\mathcal{W}_2$ surrogate, and uniform node weighting is replaced by ORC-dependent weighting. These steps are tractable approximations, not exact equivalences. Figure 3 summarizes the three steps within the distillation pipeline.

4.1 From Global Alignment to Local Neighborhood Measures

The first source of intractability is the global scope of $\mathbf{K}^{(K)}$. By Lemma 3.1, the graph-dependent part of the teacher kernel is generated through message-passing neighborhoods, even though the full kernel may also contain nonlocal feature covariance. We therefore approximate the energy-weighted alignment in $\mathcal{L}^*$ by matching local teacher and student representation distributions.

For each node v, let μ_v be the lazy random walk distribution over $\mathcal{N}^+(v)$, defined as $\mu_v(u) = \frac{1}{2}\mathbf{1}\{u=v\} + \frac{1}{2|\mathcal{N}(v)|}\mathbf{1}\{u \in \mathcal{N}(v)\}$, and define the local representation measures

$$\nu_v^T = \sum_{u\in\mathcal{V}} \mu_v(u)\,\delta_{\mathbf{r}_u^T}, \qquad \nu_v^S = \sum_{u\in\mathcal{V}} \mu_v(u)\,\delta_{\mathbf{r}_u^S}, \tag{7}$$

where $\delta_{\mathbf{x}}$ denotes the Dirac measure at point $\mathbf{x}$ and $\mathbf{r}_u = [\mathbf{h}_u^\top, \mathbf{z}_u^\top]^\top$ concatenates the penultimate hidden features and output logits of node u. Matching ν_v^T and ν_v^S in Wasserstein-2 distance encourages

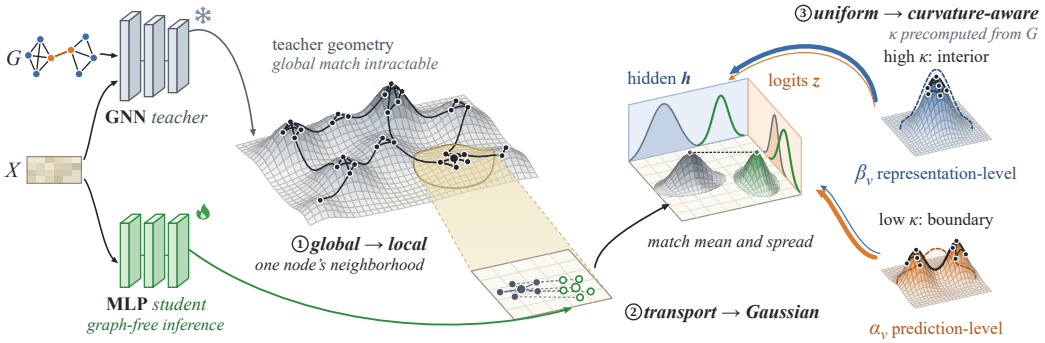


Figure 3: **Overview of G^{2}MLP.** A frozen GNN teacher is distilled into a graph-free MLP student through three practical relaxations of global alignment (Section 4): (1) teacher and student representations are compared over each node's neighborhood; (2) each neighborhood is summarized by a diagonal Gaussian; (3) Ollivier–Ricci curvature redistributes emphasis between the feature side (β_v, interior nodes) and the logit side (α_v, boundary nodes).

agreement in both local means and local dispersion. Aggregating over nodes gives the *Graph Wasserstein objective*

$$\mathcal{L}_{\mathrm{GW}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathcal{W}_2^2\big(\nu_v^T, \nu_v^S\big)\,. \tag{8}$$

Unlike pointwise KL, this term depends on neighborhood representation geometry: students with identical nodewise predictions can receive different gradients if their local representation distributions differ. In practice, we use the uniform one-hop proxy $q_v(u) = |\mathcal{N}^+(v)|^{-1}\mathbf{1}\{u \in \mathcal{N}^+(v)\}$ for computational simplicity, and write $\widetilde{\nu}_v$ for the resulting empirical measure.

4.2 Efficient Neighborhood Alignment via Moment Matching

Each $\mathcal{W}_2^2(\widetilde{\nu}_v^T, \widetilde{\nu}_v^S)$ is an optimal transport problem over $|\mathcal{N}^+(v)|$ atoms, costing $O(|\mathcal{N}^+(v)|^3)$ per node. To obtain a scalable surrogate, we match the first two marginal moments of the teacher and student neighborhood measures. This approximation is most reliable in high-curvature neighborhoods, where adjacent nodes have large structural overlap and the local representation distribution is often concentrated. We use the diagonal Gaussian approximation

$$\widetilde{\nu}_v^T \approx \mathcal{N}(\mathbf{m}_v^T, \mathrm{diag}(\boldsymbol{\sigma}_v^T)), \quad \widetilde{\nu}_v^S \approx \mathcal{N}(\mathbf{m}_v^S, \mathrm{diag}(\boldsymbol{\sigma}_v^S)), \tag{9}$$

where $\mathbf{m}_v = \sum_{u \in \mathcal{N}^+(v)} q_v(u)\mathbf{r}_u$ and $\boldsymbol{\sigma}_v = \sum_{u \in \mathcal{N}^+(v)} q_v(u)(\mathbf{r}_u - \mathbf{m}_v)^2$ denotes the coordinate-wise variance. The resulting closed-form distance is

$$\mathcal{W}_{2,\mathrm{diag}}^2(\widetilde{\nu}_v^T, \widetilde{\nu}_v^S) = \|\mathbf{m}_v^T - \mathbf{m}_v^S\|_2^2 + \|\sqrt{\boldsymbol{\sigma}_v^T + \epsilon} - \sqrt{\boldsymbol{\sigma}_v^S + \epsilon}\|_2^2. \tag{10}$$

This reduces the per-node cost to $O(d)$. The diagonal covariance follows standard practice in Gaussian-$\mathcal{W}_2$ distillation [Lv et al., 2024]; it avoids the $O(d^2)$ cost of full covariances while retaining coordinate-wise dispersion. The surrogate may be less accurate for multimodal or strongly heterophilic neighborhoods, which we treat as a limitation. Since $\mathbf{r}_u = [\mathbf{h}_u^\top, \mathbf{z}_u^\top]^\top$, the distance in Eq. (10) splits additively into a feature-side term over $\mathbf{h}$ and a logit-side term over $\mathbf{z}$, enabling the curvature-dependent weighting in Section 4.3. We denote this objective $\mathcal{L}_{\mathrm{diag}}$.

4.3 Curvature-Adaptive Weighting

The remaining gap between $\mathcal{L}_{\mathrm{diag}}$ and $\mathcal{L}^*$ is that $\mathcal{L}_{\mathrm{diag}}$ weights every node uniformly, while $\mathcal{L}^*$ emphasizes high-energy teacher directions. Section 3.2 suggests that these directions are associated with different curvature regimes in the two failure modes: low-curvature boundary regions are important for recovering missing decision-relevant teacher directions, whereas high-curvature interior regions are important for suppressing unsupported student variation after teacher smoothing. We

therefore use ORC as a local proxy for allocating the alignment budget, rather than as an exact estimator of kernel eigenvalues.

We instantiate this proxy with opposite monotone weights for the logit and feature components:

$$\alpha_v = \frac{\exp(-\gamma_z \widetilde{\kappa}_v)}{\frac{1}{|\mathcal{V}|}\sum_u \exp(-\gamma_z \widetilde{\kappa}_u)}, \qquad \beta_v = \frac{\exp(+\gamma_h \widetilde{\kappa}_v)}{\frac{1}{|\mathcal{V}|}\sum_u \exp(+\gamma_h \widetilde{\kappa}_u)}, \tag{11}$$

where α_v upweights prediction-side alignment at low-curvature boundary nodes and β_v upweights representation-side alignment at high-curvature interior nodes. The normalization enforces $\frac{1}{|\mathcal{V}|}\sum_v \alpha_v = \frac{1}{|\mathcal{V}|}\sum_v \beta_v = 1$, so curvature redistributes emphasis without changing the overall loss scale. Hyperparameters $\gamma_z, \gamma_h > 0$ control the strength of stratification.

Let $\widetilde{\nu}_{h,v}$ and $\widetilde{\nu}_{z,v}$ denote the hidden and logit marginals of $\widetilde{\nu}_v$. The resulting *Curvature-Adaptive Wasserstein* objective is

$$\mathcal{L}_{\mathrm{CAW}} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \left[\beta_v \, \mathcal{W}^2_{2,\mathrm{diag}}\big(\widetilde{\nu}^T_{h,v}, \widetilde{\nu}^S_{h,v}\big) + \alpha_v \, \mathcal{W}^2_{2,\mathrm{diag}}\big(\widetilde{\nu}^T_{z,v}, \widetilde{\nu}^S_{z,v}\big)\right], \tag{12}$$

which is computable in $O((|\mathcal{V}| + |\mathcal{E}|)\cdot d)$ per iteration.

The objective in Eq. (12) keeps the approximation aligned with the two regimes: low-curvature nodes receive stronger prediction-level correction, while high-curvature nodes receive stronger neighborhood representation matching where the Gaussian moment surrogate is most appropriate.

Objective Function. The full G^{2}MLP training objective is

$$\mathcal{L} = \lambda_{\mathrm{CE}}\mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{KD}}\mathcal{L}_{\mathrm{KD}} + \lambda_{\mathrm{CAW}}\frac{\mathcal{L}_{\mathrm{CAW}}}{\mathrm{sg}(\mathcal{L}_{\mathrm{CAW}}) + \epsilon_{\mathrm{sg}}}, \tag{13}$$

where $\mathcal{L}_{\mathrm{KD}} = \frac{1}{|\mathcal{V}|}\sum_v D_{\mathrm{KL}}(\hat{\mathbf{y}}^T_v \| \hat{\mathbf{y}}^S_v)$ and $\mathcal{L}_{\mathrm{CE}}$ is cross-entropy on labeled nodes $\mathcal{V}^L$, with $\lambda_{\mathrm{CE}} = 0$ in all node-classification and Graph Transformer experiments. The stop-gradient normalization rescales the CAW gradient by a batch-level constant without changing its direction, making its magnitude comparable to entropy-scaled losses. The KL term preserves probability-simplex calibration, while $\mathcal{L}_{\mathrm{CAW}}$ aligns local teacher-student geometry.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate on six node classification benchmarks: three citation networks (Cora, Citeseer, Pubmed), two Amazon co-purchase graphs (A-Computer, A-Photo) [Shchur et al., 2018], and the large-scale OGB graph ogbn-Arxiv [Hu et al., 2020]. The first five follow the CPF setting [Yang et al., 2021]; ogbn-Arxiv follows the standard OGB split. Dataset statistics are given in Appendix E.

Settings. We follow the protocol of GLNN [Zhang et al., 2022], evaluating the result in two regimes: *transductive*, where all nodes are visible during training but test labels are hidden; and *inductive*, where a disjoint set of test nodes (and their incident edges) is held out from training. The interpolated *production* score $\mathrm{prod} = 0.2\times\mathrm{ind} + 0.8\times\mathrm{tran}$ [Tian et al., 2023] summarizes a realistic deployment mix of seen and unseen nodes.

Baselines. We focus on the canonical GNN-to-MLP distillation setup, defined by two conditions: (a) the deployed student consumes **only the original node features** $\mathbf{x}_v$, and (b) the teacher is a **fixed, independently trained GNN**; training-time graph access is not a criterion, since every distillation method uses the graph through its teacher. The GNN teacher is GraphSAGE [Hamilton et al., 2017] throughout, following GLNN, NOSMOG. We compare against: vanilla MLP; GLNN [Zhang et al., 2022], soft-label KL distillation; KRD [Wu et al., 2023b], reliability-based sample reweighting; FF-G2M [Wu et al., 2023a], low- and high-frequency distillation. NOSMOG [Tian et al., 2023], whose student consumes graph-derived positional encodings at inference, VQGraph [Yang et al., 2024], which trains the teacher jointly with a codebook, HGMD [Wu et al., 2024], a hardness-aware reweighting scheme, and AdaGMLP [Lu et al., 2024], a boosted ensemble of MLP students, are compared separately in Section 5.3.

Implementation. We use a 2-layer GraphSAGE teacher and a 2-layer MLP student with matched hidden dimension, following standard practice [Zhang et al., 2022]; ogbn-Arxiv uses the OGB-standard 3-layer architecture for both teacher and student. All models are trained with Adam [Kingma and Ba, 2015] and early stopping on validation accuracy. Student learning rate, weight decay, and dropout follow the GLNN-released configurations and are not tuned; our loss-balance hyperparameters are set per dataset. Hyperparameter ranges and per-dataset optimizer settings are in Appendix E. We report mean accuracy ± standard deviation over 10 unique random seeds.

5.2 Main Results

Table 1: Node classification accuracy (%), **transductive** setting. **Bold**: best graph-free student.

	Teacher	Graph-free students				
Dataset	SAGE	MLP	GLNN	KRD	FF-G2M	**G^{2}MLP**
Citeseer	70.49±1.53	58.50±1.86	71.22±1.50	72.84±1.70	72.85±1.59	**74.51±2.35**
Pubmed	75.56±2.06	68.39±3.09	75.59±2.46	77.01±3.11	76.56±3.41	**78.17±2.75**
Cora	80.64±1.57	59.18±1.60	80.26±1.66	82.27±1.31	82.38±1.41	**82.54±1.87**
A-computer	82.82±1.37	67.62±2.21	82.71±1.18	82.87±0.87	83.67±1.04	**83.87±1.09**
A-photo	90.85±0.87	77.29±1.79	91.95±1.04	91.95±1.44	92.51±0.56	**92.85±0.78**
Arxiv	70.73±0.35	55.67±0.24	63.75±0.48	59.26±0.51	58.51±0.35	**64.57±0.31**

Transductive Setting. G^{2}MLP achieves the best graph-free accuracy on all six datasets, improving over the strongest baseline (FF-G2M, KRD, or GLNN depending on the dataset) by 0.16 to 1.66 pp. G^{2}MLP further surpasses the GraphSAGE teacher on five of the six datasets, with gains up to 4.02 pp on Citeseer; only on ogbn-Arxiv does the student fall short of the teacher (-6.16 pp), as feature-only inference is known to be hard on this graph [Zhang et al., 2022]. Component ablation (Appendix E.4) confirms that gains arise from curvature stratification.

Table 2: Node classification accuracy (%) under the **production** setting; conventions as in Table 1.

		Teacher	Graph-free students				
Dataset	Eval	SAGE	MLP	GLNN	KRD	FF-G2M	**G^{2}MLP**
Citeseer	prod	68.06	58.49	69.08	71.38	**71.89**	71.67
	ind	69.14±2.99	59.31±4.56	68.48±2.38	69.78±3.04	69.75±3.16	**71.85±2.27**
	tran	67.79±2.80	58.29±1.94	69.23±2.39	71.77±2.81	**72.12±2.69**	71.62±1.31
Pubmed	prod	74.77	68.39	74.67	76.00	73.98	**77.45**
	ind	75.07±2.89	68.28±3.25	74.52±2.95	75.17±3.11	73.49±7.91	**77.30±1.48**
	tran	74.70±2.33	68.42±3.06	74.70±2.75	76.20±3.00	74.10±7.78	**77.49±1.25**
Cora	prod	79.53	59.18	77.82	75.74	78.60	**79.73**
	ind	81.03±1.71	59.44±3.36	73.21±1.50	70.26±1.94	72.02±1.43	**75.90±1.87**
	tran	79.16±1.60	59.12±1.49	78.97±1.56	77.11±1.44	80.01±1.41	**80.69±0.73**
A-computer	prod	82.73	67.62	82.10	81.17	82.69	**84.29**
	ind	82.83±1.51	67.69±2.20	80.27±2.11	79.15±1.82	80.52±1.56	**82.67±0.93**
	tran	82.70±1.34	67.60±2.23	82.56±1.80	81.67±1.92	83.23±1.36	**84.69±0.85**
A-photo	prod	90.45	77.29	91.34	91.84	92.35	**92.51**
	ind	90.56±1.47	77.44±1.50	89.50±1.12	90.04±1.12	**90.70±0.76**	90.61±0.50
	tran	90.42±0.68	77.25±1.90	91.80±0.49	92.29±0.63	92.77±0.24	**92.98±0.51**
Arxiv	prod	70.69	55.35	63.50	59.32	59.60	**63.90**
	ind	70.69±0.58	55.29±0.63	59.04±0.46	57.32±0.31	57.02±0.43	**59.82±0.42**
	tran	70.69±0.39	55.36±0.34	64.61±0.15	59.82±0.27	60.24±0.23	**64.92±0.26**

Inductive and Production Settings. G^{2}MLP achieves the best graph-free production accuracy on 5/6 datasets, with gains of 0.16–1.60 pp over the strongest baseline. The advantage is largest in the inductive regime, where G^{2}MLP improves by 2.07–2.69 pp on Cora, Citeseer, Pubmed, and A-computer, more than under transductive evaluation, so curvature-stratified alignment transfers to unseen nodes. On A-computer (ind), G^{2}MLP reaches 82.67%, essentially matching the SAGE teacher (82.83%) without graph access; on Citeseer and Pubmed (prod), G^{2}MLP surpasses the teacher by 3.61 and 2.68 pp respectively.

5.3 Comparison Beyond the Fixed-Teacher Protocol

Table 9 in Appendix F.2 covers the methods that relax the fixed-teacher protocol. Run with our teachers, NOSMOG sits within ± 0.6 pp of G^{2}MLP on the CPF graphs, although our loss uses neither its representation-distillation nor its adversarial component. On ogbn-Arxiv NOSMOG's positional input is decisive: 69.72 under our teachers (71.65 published) against our 64.57, while our student with the same DeepWalk input reaches 71.52 ± 0.30. G^{2}MLP is comparable to or higher than VQGraph as reproduced by TINED [Zhou et al., 2025], whose codebook-trained teacher makes this a different-teacher comparison. Under our teachers and student width it exceeds HGMD-M and AdaGMLP on all five CPF benchmarks; on ogbn-Arxiv AdaGMLP's K-student ensemble reaches 65.51 against ours, at K times the inference cost of our student.

5.4 GT-to-MLP Distillation

Graph Transformers (GTs) incur quadratic attention cost at inference, making distillation to MLP students particularly attractive. We evaluate three GT teachers: GT [Dwivedi and Bresson, 2020] with local masked attention, GraphGPS [Rampášek et al., 2022] with hybrid MPNN+global attention, and NAGphormer [Chen et al., 2023] with per-node hop-token attention. Experiments use the same five datasets as Section 5.2 except ogbn-Arxiv, which exceeds GraphGPS dense-attention memory. The student is the same MLP used throughout.

Table 3: Transformer-teacher to MLP distillation. T: teacher; S: GLNN student; $S^\dagger$: G^{2}MLP student. **Bold**: best graph-free student per teacher.

	GT			GraphGPS			NAGphormer		
Dataset	T	S	$S^\dagger$	T	S	$S^\dagger$	T	S	$S^\dagger$
Citeseer	71.16±2.31	71.85±2.45	**72.35±1.85**	68.07±2.62	69.38±2.30	**69.68±2.05**	70.09±1.81	71.24±1.64	**71.82±1.29**
Pubmed	74.09±2.48	74.61±2.48	**74.76±2.83**	71.77±2.81	71.68±2.61	**72.03±2.58**	76.19±2.86	76.85±3.34	**77.06±3.49**
Cora	78.53±2.08	78.98±1.90	**79.21±2.09**	76.21±1.91	76.06±1.99	**76.77±1.97**	80.88±1.77	81.00±1.45	**81.48±1.53**
A-computer	83.89±1.59	84.53±1.42	**84.72±1.32**	78.30±1.70	78.99±1.73	**80.12±1.55**	82.78±1.92	83.02±1.87	**83.31±1.77**
A-photo	90.94±0.87	91.90±0.56	**92.06±0.82**	90.41±1.10	91.03±0.95	**91.32±0.93**	91.45±0.82	92.07±0.74	**92.49±0.65**

Results. The MLP student meets or exceeds its transformer teacher on $13/15$ cells ($S \geq T$: GT $5/5$, GraphGPS $3/5$, NAGphormer $5/5$), and G^{2}MLP improves over the vanilla-KL student on every cell ($S^\dagger > S$ on $15/15$) at the same student architecture and inference cost. Teacher configurations, the $S \geq T$ reversal, and the GraphGPS dense-vs-sparse comparison are in Appendix G. GCN and GAT teachers (Appendix F.3) extend the evaluation to six teacher architectures: G^{2}MLP improves over the vanilla-KL student on Cora and Citeseer under GCN and on Cora and Pubmed under GAT, and the remaining cells are within noise.

5.5 Extension to Link Prediction

Link prediction (LP) tests whether the framework generalizes beyond label fitting: supervision lives on edges and the student must produce embeddings whose inner product reproduces the teacher's edge ranking. The teacher is a 2-layer GraphSAGE trained with binary cross-entropy on training edges, and the student is the same 2-layer MLP as in node classification. We score edges by $s(u, v) = \langle \mathbf{e}_u, \mathbf{e}_v \rangle$ and train the student with an embedding-MSE distillation term and an edge-BCE term (the *multi-task* regime; Appendix H gives the protocol and ablates either term). Table 4 reports mean±std of test AUC / AP over 10 seeds.

Result. G^{2}MLP wins on all three datasets in both AUC and AP, with the largest gain on Cora (+4.70 AUC over KRD, +5.47 AP) and narrow margins on Citeseer (+0.30 / +0.51) and Pubmed (+0.04 / +0.58). All three G^{2}MLP rows exceed the SAGE teacher (Cora +2.61, Citeseer +2.41, Pubmed +2.00 AUC); on Cora and Citeseer, where the plain MLP stays below the teacher, the graph-free student thus preserves and extends the teacher's edge-ranking knowledge. The plain MLP is competitive only where node features alone carry the LP signal (Pubmed); on Citeseer it loses 7 AUC to GLNN, so the embedding-distillation term contributes structural information the edge-BCE term alone cannot recover. On Citeseer and Pubmed, where KRD already exceeds the teacher, the margin over KRD is small but positive in both AUC and AP; all methods share the same splits and teacher checkpoints, so the comparison is paired.

Table 4: Link prediction (multi-task regime); conventions as in Table 1.

	Cora		Citeseer		Pubmed	
Method	AUC	AP	AUC	AP	AUC	AP
SAGE (teacher, ref)	81.29±1.99	80.54±2.69	81.21±3.15	81.80±3.87	88.36±1.65	86.83±1.54
MLP	77.92±1.78	76.23±2.01	76.47±2.08	76.13±2.74	90.29±0.38	89.34±0.44
GLNN	78.61±1.68	77.05±1.76	83.32±2.34	83.71±2.53	90.03±0.38	89.12±0.32
KRD	79.20±1.65	77.89±2.31	83.31±2.23	83.81±2.42	90.32±0.42	89.38±0.51
G^{2}MLP (Ours)	**83.90±1.10**	**83.36±0.99**	**83.62±2.33**	**84.32±2.43**	**90.36±1.29**	**89.96±1.07**

5.6 Inference Efficiency and Scale

G^{2}MLP shares the same MLP architecture as GLNN: curvature reweighting is applied entirely at training time, so deployment reduces to a feature-vector $\rightarrow$ logit forward pass with no graph access. As Figure 4 shows, this yields a $58\times$ speedup over the 3-layer GraphSAGE teacher under standard layer-wise full inference (8.6 vs. 499.7 ms). In accuracy, G^{2}MLP recovers 59.1% of the teacher's gain over a vanilla MLP, ahead of GLNN (53.7%, same student) and matching the non-canonical TINED [Zhou et al., 2025] (58.2%, deeper student), placing it on the speed–accuracy Pareto frontier.

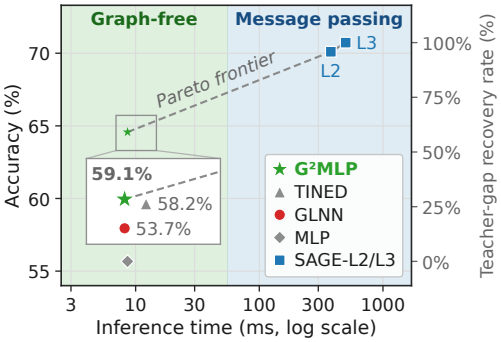


Figure 4: Accuracy vs. inference time on ogbn-Arxiv, with the teacher-gap recovery rate (right axis). Inset: distilled students (8.6–9.7 ms). L2/L3: 2-/3-layer SAGE.

Scale and precomputation cost. Exact Ollivier–Ricci curvature requires an optimal-transport solve per edge, $O(|\mathcal{E}| \cdot \bar{d}^3)$ for average degree $\bar{d}$, computed once per graph. On ogbn-Arxiv and ogbn-Products we use an $O(1)$-per-edge common-neighbor approximation (Appendix F.4); on ogbn-Products the weights build in 276 seconds on one CPU node and the objective trains end to end on one GPU from a per-epoch node and edge subsample. Over ten seeds on the public OGB split, G^{2}MLP reaches 70.03±0.30 there, above the best published graph-free result, TINED's 69.48±0.25, each number being its method's own best configuration. Appendix F.4 compares the approximation and three Forman-type curvatures [Sreejith et al., 2016, Topping et al., 2022] with exact ORC under one recipe: all agree within 0.3 pp on average, so surrogates can replace exact ORC on large graphs.

6 Conclusion

We presented Graph Geometry-aware MLP (G^{2}MLP), a distillation framework that targets the two spectral failure modes of uniform KD, spectral underfit and spectral overfit, by combining curvature-stratified reweighting with graph Wasserstein alignment at training time only. On node classification, G^{2}MLP reduces the embedding rank gap in both regimes and improves over graph-free baselines on most benchmarks at the inference cost of a vanilla MLP. The same recipe transfers to Graph Transformer teachers and to link prediction. Future work will extend curvature-stratified principle to dynamic graphs, whose curvature evolves, and to heterogeneous graphs.

Acknowledgments

The work was supported by the Professional Development (PD) Funding account (R.CE201.PD00) and University Research Scholarships, with computational resources provided by the ANU School of Computing. We thank the reviewers and area chair for their constructive feedback. We dedicate this work to the memory of the late Professor Qing Wang (Australian National University), who was not only a mentor and teacher, but also a guiding light and a motherly presence. She lit the torch illuminating the first author's research; though no longer with us, her light forever guides the way.

References

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A tokenized graph transformer for node classification in large graphs. In *International Conference on Learning Representations (ICLR)*, 2023.

Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein contrastive representation distillation. In *CVPR*, pages 16296–16305, 2021.

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. *Advances in neural information processing systems*, 22, 2009.

Rui Dong, Jiaxing Li, Weihuang Zheng, and Youyong Kong. Suit the node pair to the case: A multi-scale node pair grouping strategy for graph-mlp distillation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 2775–2783, 2025.

Simon S Du, Kangcheng Hou, Russ R Salakhutdinov, Barnabas Poczos, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *Advances in neural information processing systems*, 32, 2019.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699, 2020. URL `https://arxiv.org/abs/2012.09699`.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pages 1607–1616. PMLR, 2018.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

Zhichun Guo, William Shiao, Shichang Zhang, Yozen Liu, Nitesh V Chawla, Neil Shah, and Tong Zhao. Linkless link prediction via relational distillation. In *ICML*, 2023.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

Asela Hevapathige, Ahad N Zehmakan, and Qing Wang. Depth-adaptive graph neural networks via learnable Bakry–Émery curvature. In *KDD*, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Yang Hu, Haoxuan You, Zhecan Wang, Zhicheng Wang, Erjin Zhou, and Yue Gao. Graph-MLP: Node classification without message passing in graph. *arXiv:2106.04051*, 2021.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Yang Liu, Chuan Zhou, Shirui Pan, Jia Wu, Zhao Li, Hongyang Chen, and Peng Zhang. CurvDrop: A Ricci curvature based approach to prevent graph neural networks from over-smoothing and over-squashing. In *Proceedings of the ACM Web Conference 2023*, pages 221–230, 2023.

Suhas Lohit and Michael Jones. Model compression using optimal transport. In *WACV*, pages 2764–2773, 2022.

Weigang Lu, Ziyu Guan, Wei Zhao, and Yaming Yang. AdaGMLP: AdaBoosting GNN-to-MLP knowledge distillation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*, pages 2060–2071, 2024. doi: 10.1145/3637528.3671699.

Jiaming Lv, Haoyuan Yang, and Peihua Li. Wasserstein distance rivals Kullback-Leibler divergence for knowledge distillation. In *NeurIPS*, 2024.

Khang Nguyen, Nong Minh Hieu, Vinh Duc Nguyen, Nhat Ho, Stanley Osher, and Tan Minh Nguyen. Revisiting over-smoothing and over-squashing using Ollivier–Ricci curvature. In *ICML*, pages 25956–25979, 2023.

Chien-Chun Ni, Yu-Yao Lin, Feng Luo, and Jie Gao. Community detection on networks with Ricci flow. *Scientific Reports*, 9:9984, 2019.

Yann Ollivier. Ricci curvature of Markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems 35*, 2022.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv:1811.05868*, 2018.

Jayson Sia, Edmond Jonckheere, and Paul Bogdan. Ollivier–Ricci curvature-based method to community detection in complex networks. *Scientific Reports*, 9(1):9800, 2019.

R. P. Sreejith, Karthikeyan Mohanraj, Jürgen Jost, Emil Saucan, and Areejit Samal. Forman curvature for complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(6):063206, 2016.

Yijun Tian, Chuxu Zhang, Zhichun Guo, Xiangliang Zhang, and Nitesh Chawla. Learning MLPs on graphs: A unified view of effectiveness, robustness, and efficiency. In *ICLR*, 2023.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *ICLR*, 2022.

Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, and Zhenyu Guo. APAN: Asynchronous propagation attention network for real-time temporal graph embedding. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2628–2638, 2021.

Yiwei Wang, Bryan Hooi, Yozen Liu, and Neil Shah. Graph explicit neural networks: Explicitly encoding graphs for efficient and accurate inference. In *WSDM*, 2023.

Zehong Wang, Zheyuan Zhang, Chuxu Zhang, and Yanfang Ye. Training MLPs on graphs without supervision. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 697–706, 2025.

Lirong Wu, Haitao Lin, Yufei Huang, Tianyu Fan, and Stan Z. Li. Extracting low-/high- frequency knowledge from graph neural networks and injecting it into MLPs: An effective GNN-to-MLP distillation framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10351–10360, 2023a.

Lirong Wu, Haitao Lin, Yufei Huang, and Stan Z. Li. Quantifying the knowledge in GNNs for reliable distillation into MLPs. In *ICML*, 2023b.

Lirong Wu, Yunfan Liu, Haitao Lin, Yufei Huang, and Stan Z. Li. Teach harder, learn poorer: Rethinking hard sample distillation for GNN-to-MLP knowledge distillation. In *CIKM*, pages 2554–2563, 2024.

Cheng Yang, Jiawei Liu, and Chuan Shi. Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework. In *Proceedings of the Web Conference 2021*, pages 1227–1237, 2021.

Ling Yang, Ye Tian, Minkai Xu, Zhongyi Liu, Shenda Hong, Wei Qu, Wentao Zhang, Bin Cui, Muhan Zhang, and Jure Leskovec. VQGraph: Rethinking graph representation space for bridging GNNs and MLPs. In *ICLR*, 2024.

Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, 2016.

Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old MLPs new tricks via distillation. In *ICLR*, 2022.

Ziang Zhou, Zhihao Ding, Jieming Shi, Qing Li, and Shiqi Shen. Tined: Gnns-to-mlps by teacher injection and dirichlet energy distillation. In *International Conference on Machine Learning*, pages 78616–78632. PMLR, 2025.

A Notation and Preliminaries

Graph and feature notation. We use throughout:

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ undirected; $d_{\mathcal{G}}$ the shortest-path metric on $\mathcal{V}$.
- For $w \in \mathcal{V}$: feature $\mathbf{x}_w \in \mathbb{R}^F$, hidden state $\mathbf{h}_w \in \mathbb{R}^H$, pre-softmax logits $\mathbf{z}_w \in \mathbb{R}^C$, and degree $|\mathcal{N}(w)|$.
- Closed k-hop neighborhood: $\mathcal{N}_k^+(v) := \{w \in \mathcal{V} : d_{\mathcal{G}}(v, w) \le k\}$, with $\mathcal{N}^+(v) := \mathcal{N}_1^+(v) = \{v\} \cup \mathcal{N}(v)$.
- Lazy random-walk distribution with laziness $\alpha = 1/2$: $\mu_v := \frac{1}{2}\delta_v + \frac{1}{2|\mathcal{N}(v)|}\sum_{u \in \mathcal{N}(v)} \delta_u$.
- Mean aggregation (under any feature map ϕ): $\bar{\mathbf{x}}_v := \mathbb{E}_{u \sim \mu_v}[\phi(\mathbf{x}_u)]$.
- Ollivier–Ricci curvature on edge $(u, v) \in \mathcal{E}$: $\kappa(u, v) := 1 - W_1^{d_{\mathcal{G}}}(\mu_u, \mu_v)$.

Kantorovich–Rubinstein duality. For any probability measures ν_1, ν_2 on $(\mathcal{V}, d_{\mathcal{G}})$,

$$W_1^{d_{\mathcal{G}}}(\nu_1, \nu_2) \;=\; \sup_{\mathrm{Lip}_{d_{\mathcal{G}}}(f) \le 1} \big[\mathbb{E}_{\nu_2}[f] - \mathbb{E}_{\nu_1}[f]\big], \tag{14}$$

where $\mathrm{Lip}_{d_{\mathcal{G}}}(f) := \sup_{a \ne b} |f(a) - f(b)| / d_{\mathcal{G}}(a, b)$.

Standing assumption. Throughout the proofs we assume **(A1)**: the feature map $w \mapsto \mathbf{x}_w$ (or, more generally, a Lipschitz embedding ϕ of node features) is L-Lipschitz with respect to $d_{\mathcal{G}}$: $\|\phi(\mathbf{x}_a) - \phi(\mathbf{x}_b)\|_2 \le L \cdot d_{\mathcal{G}}(a, b)$ for all $a, b \in \mathcal{V}$.

B Full GNTK Recursion and Proofs

B.1 Full GNTK Recursion

For an infinitely wide K-layer GNN, the full GNTK recursion following Du et al. [2019] is as follows. Initialize $\Sigma^{(0)}(v, u) = \mathbf{x}_v^\top \mathbf{x}_u / F$ and $\Theta^{(0)}(v, u) = \Sigma^{(0)}(v, u)$. For each layer $k = 1, \dots, K$, both kernels are first aggregated over neighborhood pairs,

$$\bar{\Sigma}^{(k)}(v, u) = \frac{1}{|\mathcal{N}^+(v)||\mathcal{N}^+(u)|} \sum_{v' \in \mathcal{N}^+(v)} \sum_{u' \in \mathcal{N}^+(u)} \Sigma^{(k-1)}(v', u'), \tag{15}$$

$$\bar{\Theta}^{(k)}(v, u) = \frac{1}{|\mathcal{N}^+(v)||\mathcal{N}^+(u)|} \sum_{v' \in \mathcal{N}^+(v)} \sum_{u' \in \mathcal{N}^+(u)} \Theta^{(k-1)}(v', u'),$$

and the nonlinearity is then applied,

$$\Theta^{(k)}(v, u) = \Sigma^{(k)}(v, u) + \dot{\Sigma}^{(k)}(v, u)\, \bar{\Theta}^{(k)}(v, u), \tag{16}$$

where $\Sigma^{(k)}(v, u) = \mathbb{E}_{f \sim \mathcal{N}(0, \bar{\Lambda}_{vu}^{(k)})}[\sigma(f_v)\sigma(f_u)]$ is the post-activation covariance and $\dot{\Sigma}^{(k)}(v, u) = \mathbb{E}_{f \sim \mathcal{N}(0, \bar{\Lambda}_{vu}^{(k)})}[\sigma'(f_v)\sigma'(f_u)]$ its derivative kernel (arc-cosine kernels for ReLU [Cho and Saul, 2009]), with $\bar{\Lambda}_{vu}^{(k)}$ the 2×2 matrix formed by $\bar{\Sigma}^{(k)}(v, v)$, $\bar{\Sigma}^{(k)}(v, u)$, and $\bar{\Sigma}^{(k)}(u, u)$. The MLP NTK $\mathbf{K}^{\mathrm{MLP}}$ is the special case in which both aggregations are replaced by the identity, $\bar{\Sigma}^{(k)}(v, u) = \Sigma^{(k-1)}(v, u)$ and $\bar{\Theta}^{(k)}(v, u) = \Theta^{(k-1)}(v, u)$, so $\mathbf{K}^{\mathrm{MLP}}(v, u)$ depends only on $\mathbf{x}_v, \mathbf{x}_u$.

B.2 Proof of Lemma 3.1 (Aggregation Locality)

We make precise the structural claim of Lemma 3.1: the GNTK entry $\Theta^{(K)}(v,u)$ depends on node features only through the K-hop closed neighborhoods of v and u, while the MLP NTK $\mathbf{K}^{\mathrm{MLP}}(v,u)$ depends only on $\mathbf{x}_v$ and $\mathbf{x}_u$ themselves. Their difference $\Delta^{(K)}(v,u) := \Theta^{(K)}(v,u) - \mathbf{K}^{\mathrm{MLP}}(v,u)$ is therefore the graph-induced contribution generated by paths in the K-hop receptive fields of v and u. The kernel need not be sparse, since the input covariance $\Sigma^{(0)}$ is generically nonzero for any pair.

Proof. We prove the support claim by induction on the layer index k:

$$\Sigma^{(k)}(v,u) \text{ and } \Theta^{(k)}(v,u) \text{ depend on features only through } \{\mathbf{x}_w : w \in \mathcal{N}_k^+(v) \cup \mathcal{N}_k^+(u)\}.$$

Base case ($k = 0$). $\Sigma^{(0)}(v,u) = \mathbf{x}_v^\top \mathbf{x}_u / F$ is a function of $\mathbf{x}_v$ and $\mathbf{x}_u$ alone, i.e. of features in $\mathcal{N}_0^+(v) \cup \mathcal{N}_0^+(u) = \{v,u\}$. The same holds for $\Theta^{(0)} = \Sigma^{(0)}$.

Inductive step. Suppose the claim holds at layer $k-1$. Fix any pair (v,u). By Eq. (15), $\bar{\Sigma}^{(k)}(v,u)$ and $\bar{\Theta}^{(k)}(v,u)$ average $\Sigma^{(k-1)}(v',u')$ and $\Theta^{(k-1)}(v',u')$ over $(v',u') \in \mathcal{N}^+(v) \times \mathcal{N}^+(u)$. By the inductive hypothesis, each summand depends on features only through $\mathcal{N}_{k-1}^+(v') \cup \mathcal{N}_{k-1}^+(u')$. Since $v' \in \mathcal{N}^+(v)$ implies $d_{\mathcal{G}}(v,v') \le 1$, the triangle inequality gives $\mathcal{N}_{k-1}^+(v') \subseteq \mathcal{N}_k^+(v)$, and analogously $\mathcal{N}_{k-1}^+(u') \subseteq \mathcal{N}_k^+(u)$. Hence $\bar{\Sigma}^{(k)}(v,u)$ and $\bar{\Theta}^{(k)}(v,u)$, and likewise the diagonal entries $\bar{\Sigma}^{(k)}(v,v)$ and $\bar{\Sigma}^{(k)}(u,u)$, depend on features only through $\mathcal{N}_k^+(v) \cup \mathcal{N}_k^+(u)$. The kernels $\Sigma^{(k)}(v,u)$ and $\dot{\Sigma}^{(k)}(v,u)$ are deterministic functions of $\bar{\Lambda}_{vu}^{(k)}$ [Cho and Saul, 2009], so they inherit this dependence, and so does $\Theta^{(k)}(v,u) = \Sigma^{(k)}(v,u) + \dot{\Sigma}^{(k)}(v,u)\,\bar{\Theta}^{(k)}(v,u)$ by Eq. (16).

MLP comparison. Replacing both aggregations in Eq. (15) by the identity gives the MLP recursion, in which $\Sigma_{\mathrm{MLP}}^{(k)}(v,u)$ and $\Theta_{\mathrm{MLP}}^{(k)}(v,u)$ depend only on previous-layer entries at the pairs (v,v), (v,u), and (u,u). Iterating yields $\mathbf{K}^{\mathrm{MLP}}(v,u) = \Theta_{\mathrm{MLP}}^{(K)}(v,u)$ as a function of $\mathbf{x}_v, \mathbf{x}_u$ alone.

The Lemma follows on identifying $\Theta^{(K)}(v,u) = [\mathbf{K}^{(K)}]_{vu}$ and $\Delta^{(K)}(v,u) = \Theta^{(K)}(v,u) - \mathbf{K}^{\mathrm{MLP}}(v,u)$: $\Delta^{(K)}$ is determined by features in $\mathcal{N}_K^+(v) \cup \mathcal{N}_K^+(u)$ but not, in general, by those at $\{v,u\}$ alone — unlike $\mathbf{K}^{\mathrm{MLP}}$. □

B.3 Proof of Proposition 3.2 (ORC as a Local Mismatch Proxy)

Proof. Fix an edge $(i,j) \in \mathcal{E}$. By definition, $\bar{\mathbf{x}}_i = \mathbb{E}_{X\sim\mu_i}[\phi(\mathbf{x}_X)]$ and $\bar{\mathbf{x}}_j = \mathbb{E}_{Y\sim\mu_j}[\phi(\mathbf{x}_Y)]$.

Step 1 (Coupling lift). Let π be any coupling of (μ_i,μ_j), i.e. a joint distribution on $\mathcal{V}\times\mathcal{V}$ with marginals μ_i and μ_j. Linearity of expectation gives

$$\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j \;=\; \mathbb{E}_{(X,Y)\sim\pi}[\phi(\mathbf{x}_X) - \phi(\mathbf{x}_Y)]\,.$$

Step 2 (Jensen + Lipschitz). Applying the triangle inequality for the Bochner integral (Jensen for the convex norm $\|\cdot\|_2$),

$$\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2 \;\le\; \mathbb{E}_{(X,Y)\sim\pi}[\|\phi(\mathbf{x}_X) - \phi(\mathbf{x}_Y)\|_2] \;\le\; L\cdot\mathbb{E}_{(X,Y)\sim\pi}[d_{\mathcal{G}}(X,Y)]\,,$$

where the second inequality uses (A1).

Step 3 (Wasserstein and ORC substitution). Taking the infimum over couplings $\pi \in \Pi(\mu_i,\mu_j)$:

$$\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2 \;\le\; L\cdot\inf_{\pi\in\Pi(\mu_i,\mu_j)}\mathbb{E}_\pi[d_{\mathcal{G}}(X,Y)] \;=\; L\cdot W_1^{d_{\mathcal{G}}}(\mu_i,\mu_j).$$

Substituting the ORC definition $W_1^{d_{\mathcal{G}}}(\mu_i,\mu_j) = 1-\kappa(i,j)$ for adjacent (i,j) yields the upper bound in Eq. (5).

Step 4 (Lower bound under (C1)–(C2)). Let $\pi^\star$ be the optimal coupling of (C1), so that $\mathbb{E}_{\pi^\star}[d_{\mathcal{G}}(X,Y)] = W_1^{d_{\mathcal{G}}}(\mu_i,\mu_j)$. By Step 1 and (C1),

$$\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2 = \big\|\mathbb{E}_{\pi^\star}[\phi(\mathbf{x}_X) - \phi(\mathbf{x}_Y)]\big\|_2 \;\ge\; \eta\,\mathbb{E}_{\pi^\star}\|\phi(\mathbf{x}_X) - \phi(\mathbf{x}_Y)\|_2.$$

Since $\pi^\star(X,Y) > 0$ almost surely, (C2) gives $\|\phi(\mathbf{x}_X) - \phi(\mathbf{x}_Y)\|_2 \ge m\,d_{\mathcal{G}}(X,Y)$, hence $\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2 \ge \eta m\, W_1^{d_{\mathcal{G}}}(\mu_i,\mu_j) = \eta m\,(1-\kappa(i,j))$. □

On conditions (C1) and (C2). Both conditions are sufficient rather than necessary for the two-sided bound in Eq. (5); the mode-conditional reading of Section 3.2 uses the lower bound for the prediction map under underfit and the upper bound for the representation map under overfit.

C Derivation of the Multi-Granularity Decomposition

We derive the closed-form decomposition that justifies separating the joint Gaussian $\mathcal{W}_2^2$ alignment in Eq. (10) into independent feature-side and logit-side terms, and that motivates the curvature-stratified weighting Eq. (11).

Joint Gaussian model. For each node v and neighbor $u \in \mathcal{N}^+(v)$, stack the hidden representation and pre-softmax logits: $\mathbf{r}_u = (\mathbf{h}_u^\top, \mathbf{z}_u^\top)^\top \in \mathbb{R}^{H+C}$. Fit a Gaussian $\mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$ to $\{\mathbf{r}_u : u \in \mathcal{N}^+(v)\}$ from empirical moments, with block-form covariance

$$\boldsymbol{\Sigma}_v \;=\; \begin{pmatrix} \boldsymbol{\Sigma}_v^{hh} & \boldsymbol{\Sigma}_v^{hz} \\ \boldsymbol{\Sigma}_v^{zh} & \boldsymbol{\Sigma}_v^{zz} \end{pmatrix}. \tag{17}$$

Assumption D1 (Negligible cross-covariance). We model $\boldsymbol{\Sigma}_v^{hz} \approx \mathbf{0}$. This is a modeling simplification rather than an empirical fact: the logits are computed from the hidden representations, so the exact cross-covariance is not zero in general. The simplification is consistent with the diagonal surrogate of Eq. (10), which drops all off-diagonal entries, and it is what allows the two blocks to be weighted separately.

Assumption D2 (Diagonal within blocks). Following WKD [Lv et al., 2024], we assume $\boldsymbol{\Sigma}_v^{hh}$ and $\boldsymbol{\Sigma}_v^{zz}$ are diagonal. This reduces the $(H+C)\times(H+C)$ matrix square root from cost $O((H+C)^3)$ per node to coordinate-wise square roots in cost $O(H+C)$.

Decomposition. Under (D1)+(D2), the closed-form $\mathcal{W}_2^2$ between two joint Gaussians on $\mathbb{R}^{H+C}$ [Lv et al., 2024] factorizes additively along the hidden / logit blocks:

$$\mathcal{W}_2^2(\mathcal{N}_T(v), \mathcal{N}_S(v)) \;=\; \underbrace{\mathcal{W}_2^2\big(\mathcal{N}_T^h(v), \mathcal{N}_S^h(v)\big)}_{\text{feature-side}} \;+\; \underbrace{\mathcal{W}_2^2(\mathcal{N}_T^z(v), \mathcal{N}_S^z(v))}_{\text{logit-side}}, \tag{18}$$

where each diagonal-Gaussian term reduces to $\|\boldsymbol{\mu}_T - \boldsymbol{\mu}_S\|_2^2 + \|\sqrt{\boldsymbol{\sigma}_T} - \sqrt{\boldsymbol{\sigma}_S}\|_2^2$. This is the additive structure exploited in Eq. (10) of the main body and consumed in the curvature-adaptive objective Eq. (12).

Implementation note. Decomposition Eq. (18) separates the two blocks but does not by itself prescribe their relative weighting. The curvature-adaptive weights α_v, β_v in Eq. (11) apply a sign-opposite stratification of the two terms ($\beta_v \propto e^{+\gamma_h \widetilde{\kappa}_v}$ emphasizes feature-side alignment at high-curvature interior nodes; $\alpha_v \propto e^{-\gamma_z \widetilde{\kappa}_v}$ emphasizes logit-side alignment at low-curvature boundary nodes), mirroring the two failure-mode regimes characterized in Section 3.2. The sign convention therefore follows the theoretical analysis for node classification, and the only curvature-temperature hyperparameters introduced here are γ_h, γ_z.

D Training Algorithm

Algorithm 1 summarizes the two-stage G^{2}MLP training pipeline. Stage 1 trains a standard GNN teacher and precomputes node-level Ollivier–Ricci curvature once as a graph-level preprocessing step. Stage 2 trains the MLP student under the curvature-adaptive objective. At inference time, only the student MLP is deployed; no graph access, neighborhood fetching, or sampling is required.

The stop-gradient normalization $\mathrm{sg}(\mathcal{L}_{\mathrm{CAW}}) + \epsilon_{\mathrm{sg}}$ in line 10 keeps the CAW gradient direction unchanged but rescales its magnitude to match entropy-scaled losses, stabilizing the loss scale across datasets with different feature norms and reducing the sensitivity of λ_{CAW} selection.

Algorithm 1 G^{2}MLP Training Pipeline

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, teacher GNN T, student MLP S_ϕ, joint-loss weights $\lambda_{\mathrm{CE}}, \lambda_{\mathrm{KD}}, \lambda_{\mathrm{CAW}}$, curvature temperatures γ_h, γ_z

Stage 1: Teacher pre-training and graph preprocessing
Train teacher T with cross-entropy until validation convergence
Precompute node-level ORC $\{\kappa_v\}_{v\in\mathcal{V}}$ (cached to disk)
Cache teacher soft labels $\{\hat{\mathbf{y}}_v^T\}$ and hidden representations $\{\mathbf{h}_v^T\}$
Stage 2: Student training
for epoch $= 1, \ldots, E$ **do**
  Forward student: $\mathbf{h}_v^S = S_\phi(\mathbf{x}_v)$, $\mathbf{z}_v^S =$ classifier head$(\mathbf{h}_v^S)$
  Compute curvature-adaptive weights: $\beta_v \propto e^{+\gamma_h \widetilde{\kappa}_v}$, $\alpha_v \propto e^{-\gamma_z \widetilde{\kappa}_v}$ (Eq. (11))
  Compute curvature-adaptive Wasserstein loss $\mathcal{L}_{\mathrm{CAW}}$ (Eq. (12))
  Compute full objective $\mathcal{L} = \lambda_{\mathrm{CE}}\mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{KD}}\mathcal{L}_{\mathrm{KD}} + \lambda_{\mathrm{CAW}} \cdot \mathcal{L}_{\mathrm{CAW}}/(\mathrm{sg}(\mathcal{L}_{\mathrm{CAW}}) + \epsilon_{\mathrm{sg}})$ (Eq. (13))
  Update ϕ by minimizing $\mathcal{L}$ with Adam
end for
Inference: deploy S_ϕ only (no graph access required)

E Datasets, Baselines, and Implementation Details

E.1 Datasets and Splits

We evaluate on the six node-classification benchmarks summarized in Table 5. Following CPF [Yang et al., 2021] and GLNN [Zhang et al., 2022], the citation networks [Yang et al., 2016] and the Amazon co-purchase graphs [Shchur et al., 2018] are used as the largest connected component of each published graph, and the table reports these graphs; the OGB graph follows the standard release of Hu et al. [2020]. The homophily ratio h is the edge-level fraction $h = |\{(u, v) \in \mathcal{E} : y_u = y_v\}|/|\mathcal{E}|$.

Table 5: Statistics of the benchmark graphs. $\bar{d}$ is the mean degree; h is edge homophily.

Dataset	#Nodes	#Edges	#Feat.	#Cls.	$\bar{d}$	h
Cora	2,485	5,069	1,433	7	4.1	0.80
Citeseer	2,110	3,668	3,703	6	3.5	0.74
Pubmed	19,717	44,324	500	3	4.5	0.80
A-computer	13,381	245,778	767	10	36.7	0.78
A-photo	7,487	119,043	745	8	31.8	0.83
ogbn-arxiv	169,343	1,166,243	128	40	13.7	0.65

The benchmarks differ in their curvature profiles. Under exact ORC, the mean edge curvature is negative on all five CPF graphs: most strongly on the citation graphs Pubmed (-0.29) and Cora (-0.17), and closest to zero on the dense co-purchase graph A-photo (-0.02, with about half of its edges positively curved); Citeseer (-0.09) and A-computer (-0.08) lie in between.

Splits and evaluation protocol. For Cora, Citeseer, Pubmed, A-computer, and A-photo, we use the **CPF** split protocol of Yang et al. [2021]: 10 random splits per dataset with the standard CPF train/validation/test ratios; we report mean$\pm$std over the 10 splits. For ogbn-arxiv, we use the canonical OGB time-based split [Hu et al., 2020] (training on papers published until 2017, validation on 2018, and test on 2019 onward) and vary 10 seeds for model initialization and dropout.

E.2 Baselines

We compare against the canonical GNN-to-MLP distillation baselines grouped by inference-time graph access. For all baselines we use the standard student configurations released by the original authors and refrain from inference-time tricks such as enlarged hidden dimensions or increased depth.

Vanilla baseline. *MLP*: the student architecture with identical optimization (no distillation signal).

Graph-free distillation students. *GLNN* [Zhang et al., 2022]: soft-label KL distillation with no architectural change to the student. *KRD* [Wu et al., 2023b]: adds reliability-based per-node

sample weighting on top of GLNN; reliability $r_v = 1 - \delta_v^p$ is estimated from the teacher's noise sensitivity following the released protocol. *FF-G2M* [Wu et al., 2023a]: distillation of the low- and high-frequency components of the teacher's knowledge.

E.3 G^{2}MLP Implementation and Hyperparameters

Teacher and student architectures. Unless stated otherwise (Appendix F.3), the GraphSAGE teacher uses the GCN aggregator. On Cora, Citeseer, Pubmed, A-computer, and A-photo it has 2 layers (learning rate 10^{-2}, weight-decay 5×10^{-4}, dropout 0, fan-out 5,5 for neighbor sampling); on ogbn-arxiv it follows the OGB-standard 3-layer configuration (hidden dimension 256, dropout 0.2, weight-decay 0, batch normalization, fan-out 5,10,15). It is trained for 200 epochs. The MLP student has the same depth and hidden dimension as the teacher (2 layers and 128 on the five CPF datasets, 3 layers and 256 on ogbn-arxiv); it is trained for 500 epochs with Adam under the combined objective in Eq. (13). Both use early stopping on validation accuracy (patience 50). Per-dataset student optimizer settings (learning rate η, weight-decay, dropout) follow the GLNN-released configurations (Table 6); these are held fixed across all our G^{2}MLP runs and are not retuned.

Table 6: Student MLP optimizer hyperparameters.

Dataset	η	weight-decay	dropout
Cora	10^{-2}	5×10^{-3}	0.6
Citeseer	10^{-2}	10^{-3}	0.1
Pubmed	5×10^{-3}	0	0.4
A-computer	10^{-3}	2×10^{-3}	0.3
A-photo	5×10^{-3}	2×10^{-3}	0.3
ogbn-arxiv	10^{-2}	0	0.2

Hyperparameter search. The only G^{2}MLP-specific knobs we sweep are

$$\lambda_{\mathrm{CAW}} \in \{10^{-2},\ 10^{-1},\ 5\times10^{-1}\},$$
$$\textsc{caw_transform} \in \{\textsc{none},\ \textsc{student},\ \textsc{both}\},$$

set per dataset. All remaining knobs are fixed across all datasets: $\lambda_{\mathrm{KD}} = 1$, $\lambda_{\mathrm{CE}} = 0$ (pure distillation), $\gamma_h = \gamma_z = 1$, temperature $\tau = 1$. ORC is computed with lazy-walk parameter $\alpha = 0.5$, and the detach-normalized L_{CAW} uses $\epsilon_{\mathrm{sg}} = 10^{-8}$. On the Amazon graphs a jointly trained, hidden-dimension-preserving `Linear` + `ReLU` head transforms $\mathbf{h}_v^S$ only for the $\mathcal{W}_2$ term and is discarded at inference, leaving the classifier path unchanged; under an identical configuration, removing it changes accuracy by at most 0.34 pp (A-computer) and 0.06 pp (A-photo).

E.4 Component Ablation

We ablate G^{2}MLP along two axes: (i) W_2 alignment vs. KL distillation under uniform weights, and (ii) which side of the representation is curvature-stratified. All variants share the same MLP architecture, optimizer, and 10-seed protocol as the main results.

Table 7: Component ablation on transductive node classification. Baseline reproduced from Table 1.

Variant	Cora	Citeseer	Pubmed
GLNN baseline	80.26±1.66	71.22±1.50	75.59±2.46
+ W_2 uniform (no curvature)	80.86±2.30	71.07±2.19	75.22±2.15
+ curvature, logit-side only	81.78±1.55	73.87±2.09	77.27±2.92
+ curvature, feature-side only	81.34±1.88	73.99±2.02	77.14±2.93
Full G^{2}MLP (both sides)	**82.54±1.87**	**74.51±2.35**	**78.17±2.75**

W_2 alignment alone is neutral relative to GLNN (Table 7); single-side curvature stratification yields most of the improvement ($+1.5$ to $+2.8$ over GLNN); the two sides are complementary, validating the sign-opposite stratification predicted by Section 3.2.

E.5 Compute Environment

Hardware. All node-classification, Graph Transformer, and link-prediction experiments run on a single NVIDIA GeForce RTX 3090 (24 GB GDDR6X, Ampere) on a shared SLURM cluster, with an Intel Xeon Gold 6136 host (3.0 GHz, dual-socket) and 256 GB system RAM.

Software and code. Ubuntu 24.04 LTS (kernel 6.8); CUDA 12.4 / cuDNN 9.1; Python 3.10.20; PyTorch 2.6.0; DGL 2.5.0; `GraphRicciCurvature` [Ni et al., 2019] for ORC precomputation. Code is available at `https://github.com/CHENZhewei928/G2MLP-NeurIPS26`.

Wall-clock cost. For the original experimental suite, approximate wall-clock costs are: ≈80 GPU-hours for the node-classification experiments (6 datasets × 2 settings × 5 methods × 10 seeds); ≈40 GPU-hours for the Graph Transformer experiments (see Appendix G); ≈15 GPU-hours for the link-prediction experiments (see Appendix H). Inference-latency measurements (Figure 4) follow the standard layer-wise full-inference protocol on the same hardware.

F Additional Experiments

The experiments in this appendix use the protocol of Appendix E unless stated otherwise.

F.1 Curvature-Binned Diagnostics

Node-level attribution. Let $\widetilde{\mathbf{H}}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the thin SVD of the demeaned teacher representation scaled to unit trace, and $\widetilde{\mathbf{H}}^S$ the student representation scaled the same way. With $\rho_i = \mathbf{u}_i^\top \widetilde{\mathbf{H}}^S \widetilde{\mathbf{H}}^{S\top} \mathbf{u}_i / \sigma_i^2$, the mismatch attributed to node v is

$$d_v^{\text{under}} = \sum_{\rho_i < 1} (1-\rho_i)^2 \sigma_i^2 \, u_i(v)^2, \qquad d_v^{\text{over}} = \sum_{\rho_i > 1} (\rho_i - 1)^2 \sigma_i^2 \, u_i(v)^2,$$

which sum over nodes to the global underfit and overfit mismatch. Nodes are binned into five equal-population quintiles of mean incident ORC. The students are the vanilla-KL student and G^{2}MLP distilled from the Table 1 SAGE teachers (5 seeds; mean and standard error). Table 8 reports the totals of both channels (the bands of Figure 1c follow the dominant channel; Pubmed's mean rank gap is within one standard error of zero), and Figure 2 the overfit channel of both students per quintile. G^{2}MLP reduces the dominant channel on four of the five datasets, by 15.5% to 49.1%, with Cora flat.

Table 8: Total underfit and overfit energy ($\times 10^{-4}$, 5 seeds) of the vanilla-KL student (GLNN) and G^{2}MLP with the relative change; the dominant channel of each dataset is in bold.

		underfit		overfit	
Dataset	regime	GLNN	G^{2}MLP	GLNN	G^{2}MLP
Cora	underfit	**3.39**	**3.41** (+0.6%)	0.49	0.48 (−2.9%)
Citeseer	underfit	**2.46**	**2.08** (−15.5%)	1.10	0.54 (−50.9%)
Pubmed	overfit	0.15	0.11 (−22.6%)	**36.8**	**27.8** (−24.5%)
A-computer	overfit	3.44	2.78 (−19.1%)	**34.6**	**17.6** (−49.1%)
A-photo	overfit	4.71	4.73 (+0.5%)	**54.7**	**45.0** (−17.7%)

Edge-level test of the lower bound. Eq. (5) predicts, under student boundary under-response, that the teacher–student prediction mismatch across an edge grows as the edge curvature decreases. For an edge e let $U_e = \|\Delta_e \mathbf{z}^T - \Delta_e \mathbf{z}^S\|_2^2$ be the squared disagreement of the prediction contrasts across e, and $\Delta_U = \mathbb{E}[U_e \mid \text{lowest-}\kappa \text{ quintile}] / \mathbb{E}[U_e \mid \text{highest-}\kappa \text{ quintile}]$ the ratio between the extreme edge quintiles. For vanilla-KL students of the Table 1 SAGE teachers on the two underfit-dominated graphs (5 seeds, mean and 95% interval), $\Delta_U = 1.29 \pm 0.14$ on Cora and 1.94 ± 0.66 on Citeseer: the ratio exceeds one with the interval excluding one on both graphs, and likewise under independently trained GCN teachers. The node-level underfit energy is a basis-projected quantity and is spread across quintiles, whereas the bound concerns an edge-level boundary quantity, and at the edge level the low-curvature concentration appears in U_e.

F.2 Comparison Beyond the Fixed-Teacher Protocol

Table 9 reports the methods compared outside the protocol of Tables 1 and 2 (conditions (a) and (b) of Section 5.1). HGMD-M re-implements HGMD's loss inside our pipeline with our SAGE checkpoints, splits, and student; HGMD (released) runs the authors' code with its own teacher, splits, and student. NOSMOG relaxes (a), since its student consumes DeepWalk positional encodings at inference; NOSMOG (our teachers) runs the official code under our teachers and splits (‡: its released ogbn-Arxiv configuration enables neither the adversarial nor the representation-distillation component). The DeepWalk control of Section 5.3 changes only the student input. VQGraph relaxes (b), since its teacher is trained jointly with a codebook, so student-level differences also reflect a different teacher; its row quotes the reproduction by TINED [Zhou et al., 2025] under the same GraphSAGE-teacher protocol. AdaGMLP [Lu et al., 2024] deploys a boosted ensemble of K MLP students, so its inference cost scales with K; its row runs the official student code with our teachers, splits, and student width, with its method hyperparameters selected on validation per dataset.

Table 9: Transductive accuracy (%) of methods outside the protocol of Table 1, mean±std over 10 seeds; "—" marks a cell the corresponding pipeline does not provide.

Method	Cora	Citeseer	Pubmed	A-computer	A-photo	Arxiv
G^{2}MLP (Table 1)	82.54±1.87	74.51±2.35	78.17±2.75	83.87±1.09	92.85±0.78	64.57±0.31
HGMD-M (matched)	81.80±1.61	73.23±2.05	77.99±2.48	82.10±1.27	91.47±1.44	63.59±0.34
HGMD (released)	82.61±1.19	73.46±1.06	81.03±0.66	—	92.49±1.22	—
NOSMOG (our teachers)	83.07±1.65	74.40±1.97	78.76±1.58	83.65±1.21	92.58±1.35	69.72±0.33‡
NOSMOG (published)	83.04±1.26	73.78±1.54	77.34±2.36	84.04±1.01	93.36±0.69	71.65±0.29
VQGraph [Zhou et al., 2025]	78.66±1.21	74.66±1.23	73.02±3.51	80.16±2.02	92.32±1.74	—
AdaGMLP (our teachers)	82.06±1.91	73.80±1.88	77.38±3.01	83.44±1.21	92.40±1.79	65.51±0.28

F.3 Other Teacher Architectures

Table 10 reports GCN and GAT teachers on the five CPF benchmarks under the GLNN protocol. Under the GCN teacher G^{2}MLP improves over the vanilla-KL student by 0.68 pp on Cora and 0.33 pp on Citeseer, and under the GAT teacher by 1.35 pp on Cora and 0.72 pp on Pubmed; every remaining GCN and GAT cell is within noise (at most 0.26 pp in magnitude). Together with GraphSAGE (Table 1) and the three Graph Transformer teachers (Table 3), the curvature-stratified loss improves distillation under six teacher architectures without re-tuning its mechanism.

Table 10: Transductive accuracy (%) with GCN and GAT teachers (10 seeds; student hidden width matched to the teacher). Bold marks the better student; the A-photo cells under GAT are tied.

Teacher	Student	Cora	Citeseer	Pubmed	A-computer	A-photo
GCN	GLNN	81.38±2.03	72.08±2.08	**76.62±3.03**	83.22±2.06	**92.98±0.70**
	G^{2}MLP	**82.06±1.38**	**72.41±1.69**	76.36±2.87	**83.37±1.99**	92.82±0.89
GAT	GLNN	80.25±2.75	72.07±1.85	75.89±2.61	**82.95±1.10**	92.86±1.16
	G^{2}MLP	**81.60±1.84**	**72.22±1.55**	**76.61±2.83**	82.92±1.26	92.86±1.21

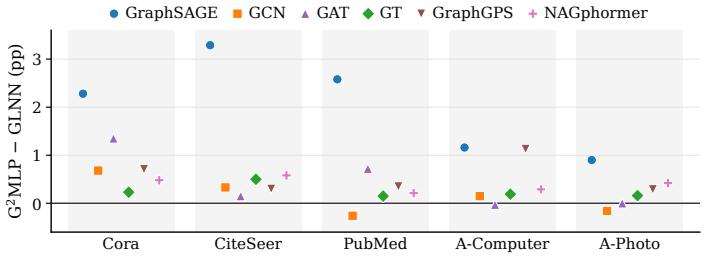


Figure 5: Test-accuracy gain of the G^{2}MLP student over the vanilla-KL student (pp) for six teacher architectures (10 seeds; Tables 1, 3, and 10); the horizontal line marks equal accuracy.

F.4 Curvature Computation Ablation

The common-neighbor approximation replaces the per-edge optimal transport by $W_1(\mu_i, \mu_j) \approx \alpha + (1-\alpha)\big(1 - |N(i) \cap N(j)|/\sqrt{d_i d_j}\big)$, which costs $O(1)$ per edge and is the curvature used on ogbn-Arxiv and ogbn-Products. Swapping the curvature computation among exact ORC, this approximation, augmented Forman, classic 1d Forman, and balanced Forman [Sreejith et al., 2016, Topping et al., 2022] under an identical training recipe (10 seeds on each of the five CPF benchmarks per curvature), the five computations agree within 0.3 pp on average, with a maximum single-dataset deviation of 1.25 pp, so any curvature that preserves the ordering from bridges to communities suffices for the allocation mechanism.

F.5 Robustness to Feature Outliers

We replace a fraction p of the node-feature entries with heavy-tailed Cauchy noise (scale equal to one feature standard deviation) and retrain every method from scratch at each level (5 seeds; Cora and A-computer as the underfit- and overfit-dominated regimes). Figure 6 shows the result. G^{2}MLP tracks the vanilla-KL student within noise at every level (largest gap 0.67 pp in either direction), so the diagonal-moment surrogate introduces no measurable heavy-tail fragility relative to plain logit distillation; on A-computer, where the representation-level term carries the most weight, G^{2}MLP loses 10.6 pp from clean to $p = 20\%$ against 14.4 pp for the plain MLP. Even at $p = 20\%$, both distilled students keep an advantage of about 15 pp over the feature-only MLP on both datasets. The theory does not cover strongly contaminated or multimodal neighborhoods, which we state as a limitation in Appendix I.

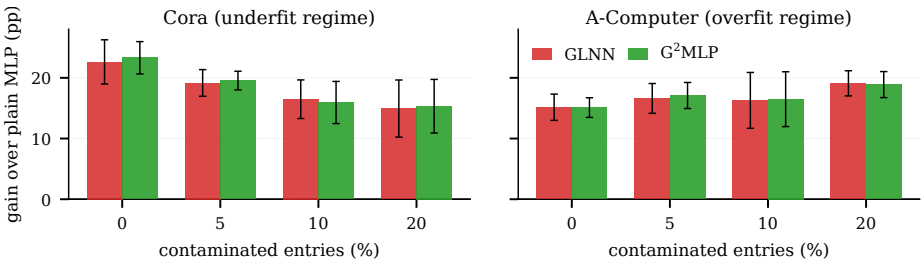


Figure 6: Mean test-accuracy gain of each distilled student over the feature-only MLP (pp) as a fraction of the feature entries is replaced by Cauchy noise (5 seeds; bars: std of the per-seed gain).

G Graph Transformer Teacher Experiments

This appendix consolidates implementation and analysis details for the Graph Transformer (GT) to MLP distillation experiments in Section 5.4.

G.1 GT Architectures and Configurations

We evaluate three GT architectures as teachers, matching the columns of Table 3 in the main body: vanilla `GT` [Dwivedi and Bresson, 2020], `GraphGPS` [Rampášek et al., 2022], and `NAGphormer` [Chen et al., 2023]. Each teacher is trained for 500 epochs with Adam and uses the same hidden dimension as our GraphSAGE teacher, so that students are compared at matched width; the remaining hyperparameters are set per dataset. The GraphGPS teacher uses dense $N \times N$ attention as its canonical default; we additionally analyze a sparse edge-softmax variant in Appendix G.4. Across all GT teachers, the student is a 2-layer MLP with hidden dimension matched to the teacher and is trained with the G^{2}MLP objective of the main experiments; the CAW weight and the curvature temperature are chosen per (teacher, dataset) cell from a small grid.

G.2 Why Students Can Exceed Their GT Teachers

Table 3 shows that the MLP student meets or exceeds its GT teacher on $13/15$ cells. This $S \geq T$ reversal arises because GTs at the dataset sizes considered (2K–20K nodes) are difficult to train well: the small label budget (140 / 120 / 60 nodes on Cora / Citeseer / Pubmed) provides limited supervision

for $O(N^2)$ attention interactions, so randomly initialized full attention often produces near-uniform patterns that the 500-epoch training schedule does not escape. Distillation acts as a regularizer in this regime: the student inherits the teacher's prediction targets while shedding the teacher's overfit attention noise, with the curvature-adaptive weighting in G^{2}MLP further focusing the alignment budget on neighborhoods where the teacher's geometry is reliable. The same phenomenon has been observed for "born-again" distillation in vision under analogous data-scarce regimes [Furlanello et al., 2018]; our contribution is showing that it persists with GT teachers and that the gain extends to MLP-only inference.

G.3 Why NAGphormer Is Often the Strongest GT Teacher

NAGphormer is the strongest teacher on most CPF benchmarks because its hop-token embedding produces a permutation-invariant summary at every hop distance, which (i) regularizes the attention pattern when the training signal is small, and (ii) injects graph structure into the token sequence rather than relying on attention alone to recover it.

G.4 GraphGPS: Sparse vs. Dense Attention

GraphGPS canonically uses dense $N \times N$ attention; sparse edge-softmax is a common alternative that restricts attention to graph edges. On the dataset sizes considered ($N \in [2{,}000, 20{,}000]$), both variants fit comfortably in 24 GB GPU memory and converge in 500 epochs. The dense vs. sparse comparison in Table 11 is therefore not a memory tradeoff but an inductive-bias one: edge-softmax acts as an explicit locality prior, while dense attention must learn locality from scratch.

Table 11: G^{2}MLP vs. GLNN with the GraphGPS teacher under both canonical dense ($N \times N$) and sparse (edge-softmax) attention. Conventions as in Table 3: T teacher, S GLNN student, $S^\dagger$ G^{2}MLP student. **Bold**: best graph-free student per teacher.

	GraphGPS (dense, canonical)			GraphGPS (sparse, edge-softmax)		
Dataset	T	S	$S^\dagger$	T	S	$S^\dagger$
Cora	76.21±1.91	76.06±1.99	**76.77±1.97**	78.12±2.12	77.86±2.09	**78.74±1.87**
Citeseer	68.07±2.62	69.38±2.30	**69.68±2.05**	68.88±2.25	70.29±2.13	**70.44±2.13**
Pubmed	71.77±2.81	71.68±2.61	**72.03±2.58**	75.61±2.13	**76.08±2.34**	75.97±2.21
A-computer	78.30±1.70	78.99±1.73	**80.12±1.55**	81.84±0.88	82.05±1.18	**82.55±0.86**
A-photo	90.41±1.10	91.03±0.95	**91.32±0.93**	90.34±1.21	91.12±1.04	**91.26±1.02**

Discussion. The dense GraphGPS teacher is intrinsically weaker than vanilla GT or NAGphormer on small citation graphs, for the reason given in Appendix G.2; the MPNN branch of GraphGPS carries most of the model's capacity. Even so, with the dense teacher G^{2}MLP improves over GLNN and surpasses the teacher on all five datasets; with the sparse teacher it surpasses the teacher on all five and improves over GLNN on four (Pubmed: -0.11 pp). The sparse variant is a stronger teacher than dense on four of the five graphs (e.g. A-computer $T = 81.84$ vs. 78.30; Cora $T = 78.12$ vs. 76.21; A-photo is a tie, 90.34 vs. 90.41). The G^{2}MLP gains over GLNN hold in nine of the ten cells across the two attention modalities.

H Link Prediction: Pure-Distillation Ablation

This appendix accompanies Section 5.5 of the main body, which reports only the multi-task regime (KD + edge-BCE). Here we ablate the edge-BCE term to isolate the contribution of embedding distillation alone, and analyze the resulting cross-regime gap.

Setup. We use the largest-connected-component CPF graphs of Table 5. Teacher and student architectures follow Appendix E.3. Edges are split 85/5/10 into train/validation/test with the standard `RandomLinkSplit` protocol and 1:1 negative sampling; all models train for up to 500 epochs with best-validation-AUC checkpointing. Per-dataset MLP optimizer settings are those of Table 6. KRD's reliability weights use 10 noise realizations of magnitude 1.0, bin count 50, and momentum 0.99 [Wu et al., 2023b]. The teacher is trained once per seed and shared across both regimes.

G^{2}MLP-LP hyperparameter search. G^{2}MLP-LP introduces three task-specific hyperparameters in addition to the curvature temperatures γ_h, γ_z shared with node classification: the KD weight λ_{KD}, the CAW weight λ_{CAW}, and the sign of γ_h in the feature-side weight β_v of Eq. (11). The node-classification analysis (Section 3.2) motivates the $\beta_v \propto e^{+\gamma_h \widetilde{\kappa}_v}$ convention for label-fitting tasks; link prediction is governed by a different inductive signal (inner-product edge ranking), so we treat the sign as task-specific and set it per dataset. We refer to the two choices as the *community* direction ($\beta_v \propto e^{+\gamma_h \widetilde{\kappa}_v}$, emphasizing high-curvature interior nodes) and the *bridge* direction ($\beta_v \propto e^{-\gamma_h \widetilde{\kappa}_v}$, emphasizing low-curvature boundary nodes). The logit-side direction α_v is held fixed at its node-classification convention ($\alpha_v \propto e^{-\gamma_z \widetilde{\kappa}_v}$, emphasizing boundary nodes) for all LP experiments. We select per dataset from a 72-config grid on the multi-task split: $\lambda_{\mathrm{KD}} \in \{0.05, 0.1, 0.2, 0.5, 1, 2\}$, $\lambda_{\mathrm{CAW}} \in \{10^{-3}, 10^{-2}, 10^{-1}\}$, $\gamma_h \in \{1, 10\}$, β_v direction $\in \{\text{community}, \text{bridge}\}$. Other settings are held fixed at the validated configuration from node-classification (Appendix E.3). Selected values:

Dataset	λ_{KD}	λ_{CAW}	γ_h	β_v direction
Cora	0.20	10^{-2}	10	bridge
Citeseer	0.05	10^{-3}	10	community
Pubmed	0.05	10^{-2}	1	community

The dataset-dependent shape is informative. Cora prefers a strong distillation weight ($\lambda_{\mathrm{KD}} = 0.2$) and the *bridge* direction, consistent with its sparse, high-curvature-variance structure where bridge nodes carry disproportionate LP-relevant information. Citeseer and Pubmed prefer weaker distillation and the *community* direction, where neighborhood Gaussians are denser and statistically more reliable, and LP supervision benefits from interior alignment. Baselines (MLP, GLNN, KRD) use a single global hyperparameter set; G^{2}MLP-LP's per-dataset selection mirrors KRD's own per-dataset p-power tuning in the original paper [Wu et al., 2023b].

Pure regime: ablating the edge-BCE term. In the pure-distillation regime the student sees only the teacher's frozen embeddings (MSE loss); no edge labels are available. On Citeseer and Pubmed the GLNN / KRD pure-KD students already exceed the teacher, clearly on Citeseer (84.26 AUC vs. 81.21) and marginally on Pubmed (88.63 vs. 88.36), showing that the teacher's embeddings carry essentially sufficient LP information; the edge-BCE term in the multi-task default changes Citeseer AUC by less than one point and adds 1.4–1.7 AUC on Pubmed. On Cora, however, the GLNN / KRD pure students collapse to 56–57 AUC, a ≈25-point drop from the teacher's 81.29: the teacher's embeddings encode label-discriminative directions but lose the geometric anisotropy needed for inner-product LP scoring on this small, sparse graph. ***G^{2}MLP-LP eliminates this collapse***: with curvature-adaptive reweighting and neighborhood Gaussian $\mathcal{W}_2$ alignment, the pure student recovers to 81.77 AUC on Cora (+24.5 over KRD; matches the teacher), and stays competitive with KRD on the other two graphs. This is direct evidence that curvature-aware geometry transfers more LP-relevant structure than uniform or reliability-weighted MSE.

Table 12: Link prediction (pure-distillation regime, no edge-BCE). Conventions as in Table 4.

	Cora		Citeseer		Pubmed	
Method	AUC	AP	AUC	AP	AUC	AP
SAGE (teacher, ref)	81.29±1.99	80.54±2.69	81.21±3.15	81.80±3.87	88.36±1.65	86.83±1.54
GLNN	56.18±5.79	56.81±4.55	83.99±2.03	84.23±2.16	88.62±1.57	87.17±1.58
KRD	57.23±5.53	57.41±5.02	**84.26±2.56**	**84.46±2.59**	88.63±1.56	87.18±1.57
G^{2}MLP (Ours)	**81.77±2.25**	**80.28±2.84**	82.89±3.61	83.15±4.39	**88.82±1.46**	**87.43±1.35**

Cross-regime gap. Subtracting the pure-regime AUC (Table 12) from the multi-task AUC (Table 4) on the same teacher and split yields, for GLNN: +22.4 on Cora, −0.7 on Citeseer, +1.4 on Pubmed; KRD is almost identical (+22.0/ − 1.0/ + 1.7). Where the gap is large and positive (Cora), the edge-BCE term is doing genuine work for the uniform-MSE baselines. G^{2}MLP-LP closes most of the Cora gap on its own (gap +2.1), confirming that curvature-aware geometry already encodes the structural signal that uniform MSE needs the BCE term to supply.

I Limitations and Future Directions

Diagonal Gaussian neighborhood surrogate. The diagonal Gaussian in Eq. (12) is a scalable moment-matching surrogate, not an assumption that all graph neighborhoods are exactly Gaussian (Section 4.2, Assumption D2 in Appendix C). For strongly multimodal neighborhoods it may underrepresent the alignment cost by collapsing several modes onto one mean and per-coordinate variance. Mixture or low-rank covariance surrogates are natural extensions.

Scope of the theoretical results. Lemma 3.1 establishes aggregation locality, and Proposition 3.2 gives an ORC-dependent two-sided bound under sufficient conditions; ORC is a proxy for allocating supervision, not an estimator of kernel eigenvalues. These results do not prove that all spectral underfit lies at low curvature or all spectral overfit at high curvature. A spectral curriculum that schedules supervision from the student's empirical NTK spectrum and the teacher kernel's energy along it would be more direct; whether it transfers kernel structure rather than outputs is open.

Heterophily and contaminated features. Curvature-stratified alignment presupposes locally concentrated neighborhoods. Under strong heterophily, where a node's neighbors belong to different communities, this premise fails and the mechanism degrades gracefully: it neither helps nor materially hurts relative to uniform distillation. Under heavy-tailed feature contamination (Appendix F.5) G^2MLP likewise tracks the vanilla-KL student; the theory covers neither case.

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's three claims — spectral misalignment via two failure modes, curvature-adaptive Wasserstein alignment, and transfer to GT teachers and link prediction — are supported respectively by Section 3, Section 4 with Tables 1–2, and Sections 5.4–5.5.

   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: Appendix I discusses three limitations: the diagonal Gaussian neighborhood surrogate, the scope of the theoretical results, and heterophily and contaminated features.

   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: Assumptions are collected in Appendix A; proofs of Lemma 3.1 and Proposition 3.2 are in Appendix B; the multi-granularity decomposition is derived in Appendix C.

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: Appendix E reports datasets, splits, baselines, hyperparameter ranges, and the compute environment; code is available at the repository listed in Appendix E.5.

   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: All datasets are public benchmarks (CPF release of Yang et al. [2021]; OGB release of Hu et al. [2020]); code is available at the URL in Appendix E.5.

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: Section 5 and Appendix E.3 describe the experimental protocols, architectures, optimiser, hyperparameter ranges, and ORC parameters; the swept hyperparameters are set per dataset.

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 main result tables report mean$\pm$std over random seeds, following the CPF and OGB protocols of prior work; seed counts and the reported uncertainty are stated with each experiment.

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: Hardware (single NVIDIA RTX 3090, 24 GB; shared SLURM cluster), software stack (CUDA 12.4, PyTorch 2.6, DGL 2.5), and wall-clock cost ($\approx$80 + 40 + 15 GPU-hours for node classification, GT, and LP experiments respectively) are reported in Appendix E.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 use only public graph benchmarks with no personally identifiable information and find no item of the Code of Ethics that applies adversely.

   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: [N/A]

Justification: This is foundational research on knowledge distillation for graph neural networks; it does not introduce new applications or capabilities beyond those of the underlying GNN/MLP models.

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: No new datasets, pretrained models, or other high-risk assets are released.

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: All datasets (CPF, OGB) and reference implementations (GLNN, KRD, FF-G2M) are cited at first use; the software stack (PyTorch, DGL, `GraphRicciCurvature`) is listed in Appendix E.5; all are used under their respective open-source licenses.

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: [Yes]

    Justification: Code is available at the repository listed in Appendix E.5.

    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: No crowdsourcing or human-subject research is involved.

    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: No human-subject research is involved; IRB review is not applicable.

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: [Yes]

    Justification: LLMs were used as writing and coding assistants: editing text, drafting the plotting scripts behind the result figures, and checking bibliography entries against their sources. The method, experiments, analyses, and claims were authored and verified by the authors; LLMs were not used in any component of the methodology or experimental pipeline.

    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.