GEOMETRY-AWARE GRAPH COARSENING VIA OLLIVIER-RICCI CURVATURE

Anonymous authors
Paper under double-blind review

ABSTRACT

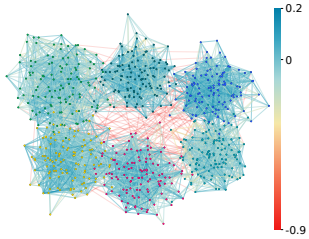
We introduce Ollivier-Ricci Graph Coarsening (ORGC), a geometry-aware graph coarsening framework that formulates node contraction through Ricci-curvature optimization. Existing methods commonly use adjacency rules or Laplacian preservation, which do not model metric-measure geometry explicitly. Gromov-Wasserstein coarsening makes this geometry explicit, but its global distance and assignment computations make efficient implementation difficult. ORGC instead adopts Ollivier-Ricci curvature as an efficient geometric primitive for comparing Markov neighborhoods. We define Ricci Geometric Coherence (RGC) over non-singleton coarse clusters, which measures all-pairs curvature agreement among nodes that are actually contracted together. ORGC derives a contraction-forest lower bound $\mathcal{L}_h$ for this objective and exactly maximizes this bound through a Kruskal-style greedy algorithm. We further establish a boundary-preservation guarantee for ORGC under a forest-rank separation condition. Experiments on structural fidelity, graph classification, molecular regression, and runtime scaling show that ORGC preserves graph structure and downstream utility while remaining efficient.

1 INTRODUCTION

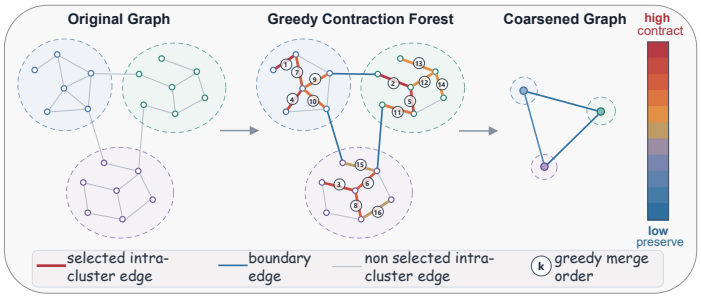
Graphs provide a common representation for relational data in social and scientific interaction networks (Wasserman & Faust, 1994; Newman, 2001), molecular and biological systems (Barabasi & Oltvai, 2004; Méndez-Lucio et al., 2024), and knowledge graphs used for structured reasoning (Sun et al., 2024). They encode both entities and their dependencies, supporting learning, reasoning, retrieval, and simulation over structured data. Recent graph pretraining and graph foundation models further broaden this role through cross-domain transfer, language interfaces, and instruction tuning (Liu et al., 2024; Tang et al., 2024; Liu et al., 2025). However, relational structure becomes expensive to process as graphs grow. Message passing scales with the number of edges, graph descriptors require repeated structural comparisons, and large knowledge graphs can be costly to query and visualize. Graph coarsening addresses this problem by replacing $G = (\mathcal{V}, \mathcal{E}, A)$ with a smaller graph $G_c = (\mathcal{V}_c, \mathcal{E}_c, A_c)$ whose nodes summarize groups of original nodes (Karypis & Kumar, 1995; Loukas, 2019; Jin et al., 2020; Cai et al., 2021). The resulting graph should reduce computation while retaining the relational structure required by downstream analysis.

Many coarsening methods construct coarse graphs from local connectivity patterns. Classical multilevel and partitioning methods merge nodes according to adjacency and matching heuristics (Karypis & Kumar, 1995). These methods are efficient, but adjacency alone does not identify the geometric role of a contraction. An edge inside a coherent region and an edge crossing a bottleneck can have the same weight, even though contracting them has different effects on diffusion, boundary structure, and downstream representations. Another line of work treats the graph Laplacian as the preservation target (Loukas & Vandergheynst, 2018; Loukas, 2019; Jin et al., 2020). These methods preserve Laplacian eigenvalues, eigenspaces, Rayleigh quotients, and related quadratic forms. Such objectives are principled because Laplacian behavior controls diffusion and smooth graph signals. However, they model graph geometry indirectly and can require eigenspace computation or iterative Laplacian-preservation procedures. A coarse graph with favorable spectral error can still merge neighborhoods whose metric-measure geometry differs substantially.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Ollivier-Ricci curvature visualization.



(b) ORGC contraction forest.

Figure 1: Motivation and procedure of ORGC. (a) Intra-community edges tend to have higher Ollivier-Ricci curvature, while cross-community edges often have lower curvature. (b) ORGC selects high-curvature acyclic contractions and leaves low-curvature boundary edges uncontracted in the illustration.

Geometry-based methods, such as Gromov-Wasserstein (GW) coarsening, model graphs as metric-measure spaces (Mémoli, 2011; Peyré et al., 2016). GW compares pairwise graph distances, but constructing global distances and optimizing assignments can be costly. Approximate and reformulated GW coarsening methods address this cost (Chen et al., 2023; Taveras et al., 2026), creating a trade-off between computational efficiency and fidelity to the original objective. This motivates a coarsening rule that uses geometric information without global assignment optimization.

Ollivier-Ricci curvature provides a suitable geometric primitive. It originates from metric-measure geometry and compares how probability mass around nearby points spreads under Wasserstein distance (Ollivier, 2009). On a graph, each node is associated with a Markov measure over its neighborhood, and the curvature between two nodes quantifies the transport discrepancy between their measures relative to graph distance. High curvature indicates geometrically coherent neighborhoods, whereas low curvature often occurs near bridges, bottlenecks, and transitions between graph regions. Prior work has exploited this geometry for graph subsampling (Wu et al., 2023) and graph clustering and pooling (Feng & Weber, 2024). In particular, ORC-Pool uses iterative Ricci flow to guide cluster assignment, with its GNN implementation further optimizing a differentiable pooling objective to learn assignments. This introduces computational costs from iterative reweighting and assignment optimization. Curvature itself, however, already provides a geometric criterion for node contraction. As illustrated in Figure 1, intra-community edges tend to have higher curvature, whereas edges across structural boundaries tend to have lower curvature. This observation motivates an efficient, training-free graph coarsening method that directly uses Ollivier-Ricci curvature to select contractions.

We formalize this idea through Ricci Geometric Coherence (RGC), defined as the average pairwise curvature within each connected, non-singleton coarse cluster and aggregated into a partition-level objective. We propose Ollivier-Ricci Graph Coarsening (ORGC), which computes depth-h curvature on input edges and constructs coarse clusters through a contraction forest. For a fixed target size, a Kruskal-style algorithm exactly maximizes a forest lower bound on the RGC objective.

Our contributions are summarized as follows:

- We introduce a geometry-aware view of graph coarsening based on Ollivier-Ricci curvature, using curvature to identify geometrically coherent contractions.
- We define Ricci Geometric Coherence (RGC), which quantifies pairwise agreement between Markov neighborhoods within connected non-singleton coarse clusters, and derive a tractable contraction-forest lower bound.
- ORGC exactly optimizes the proposed forest lower bound through a Kruskal-style procedure, and we establish a boundary-preservation guarantee under a forest-rank separation condition.
- Experiments on structural fidelity, graph classification, molecular regression, and runtime scaling show leading aggregate performance among compressed-graph baselines.

2 BACKGROUND

2.1 GRAPH COARSENING

Let $G = (\mathcal{V}, \mathcal{E}, A)$ be an undirected weighted graph with N nodes and c_G connected components. Coarsening assigns its nodes to $n < N$ coarse nodes through $Q \in \{0,1\}^{N \times n}$, with one nonzero per row. The coarse adjacency is

$$A_c = Q^\top A Q.$$

Thus $(A_c)_{ab}$ sums the weights of edges between clusters C_a and C_b.

We primarily assess whether coarsening preserves graph structure, alongside runtime and downstream utility.

2.2 WASSERSTEIN DISTANCE

Optimal transport compares distributions while accounting for distances between their supports (Kantorovich, 1958; Villani, 2009; Peyré & Cuturi, 2019). For probability measures μ, ν on a finite metric space $(\mathcal{X}, d)$, the 1-Wasserstein distance is

$$W_{1,d}(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \sum_{x,y \in \mathcal{X}} d(x,y)\pi(x,y).$$

Here $\Pi(\mu, \nu)$ contains couplings with marginals μ and ν. For nodes in the same connected component, ORGC uses hop shortest-path distance as the ground metric:

$$d_G(x,y) = \text{hop shortest-path distance between } x \text{ and } y. \tag{1}$$

2.3 OLLIVIER-RICCI CURVATURE ON GRAPHS

Ollivier-Ricci curvature measures the contraction and expansion of local neighborhoods under a Markov transition. For any distinct vertices u and v in the same connected component, it is defined as (Ollivier, 2009; Lin et al., 2011; Jost & Liu, 2014)

$$\kappa(u,v) = 1 - \frac{W_{1,d}(m_u, m_v)}{d(u,v)}.$$

Let P be the random-walk transition matrix on G. For an unweighted graph, $P_{uv} = 1/d_u$ when $v \in \mathcal{N}(u)$. For a nonnegative weighted graph, we interpret A_{uv} as connection strength rather than metric length, and set

$$P_{uv} = \frac{A_{uv}}{\sum_z A_{uz}}.$$

Given laziness $\alpha \in [0,1]$, a standard graph-local lazy measure is

$$m_u^\alpha = \delta_u\big(\alpha I + (1-\alpha)P\big),$$

where δ_u is the point mass at u. This measure describes the local lazy random-walk neighborhood around u. Edge weights affect the Markov kernel, while the transport cost is measured by the graph-support hop metric in Eq. 1. The resulting graph Ollivier-Ricci curvature is

$$\kappa_\alpha(u,v) = 1 - \frac{W_{1,d_G}(m_u^\alpha, m_v^\alpha)}{d_G(u,v)}.$$

This definition applies within each connected component. On every input edge, $d_G(u,v) = 1$, which is the form used for contraction.

3 METHOD

In this section, we propose Ollivier-Ricci Graph Coarsening (ORGC). Building on the principle that high Ollivier-Ricci curvature indicates coherent Markov geometry, ORGC defines Ricci Geometric Coherence to score candidate non-singleton coarse clusters. It then derives a contraction-forest lower bound and solves the resulting objective through a Kruskal-style contraction procedure.

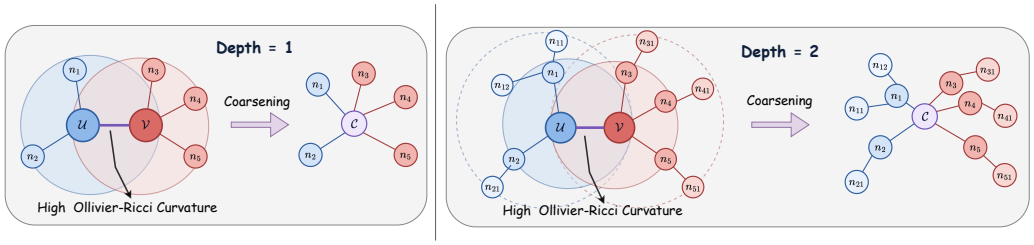


Figure 2: Depth-1 and depth-2 Markov geometry around the input edge (u, v) whose curvature is evaluated. Depth-2 expands the measure through non-backtracking two-hop transitions.

3.1 Depth-h Ollivier-Ricci Curvature

Standard graph Ollivier-Ricci curvature compares one-step lazy Markov neighborhoods, which fixes its geometric range to immediate neighbors. We make this range adjustable through a depth-h non-backtracking stopping measure. Let P be the transition matrix and let $\alpha \in [0, 1]$ be a stopping parameter. Starting from u, a fraction α of the surviving mass stops at each intermediate depth, while the remaining mass follows P without immediately returning to the preceding node. Let $p_{uv}^{(t)}$ denote the endpoint probability after t such transitions. If no non-backtracking continuation exists, such as upon reaching a degree-one endpoint, the remaining mass stays at that endpoint. The resulting measure is

$$m_u^{\alpha,h} = \alpha\delta_u + \sum_{t=1}^{h-1} \alpha(1-\alpha)^t \sum_{v\in\mathcal{V}} p_{uv}^{(t)}\delta_v + (1-\alpha)^h \sum_{v\in\mathcal{V}} p_{uv}^{(h)}\delta_v.$$

The final term collects the mass that reaches depth h without stopping. The appendix gives the pathwise transition rule and verifies the depth-1 and depth-2 forms used in the implementation. At $h = 1$, this recovers the standard lazy neighborhood measure. At $h = 2$, surviving mass reaches second-order neighbors, while excluding immediate reversals prevents the added step from returning directly to the source. The stopping parameter α controls how much mass remains near the source. For distinct nodes u and v in the same connected component, the corresponding curvature is

$$\kappa_{\alpha,h}(u, v) = 1 - \frac{W_{1,d_G}(m_u^{\alpha,h}, m_v^{\alpha,h})}{d_G(u, v)}.$$

The algorithm evaluates curvature only on input edges, whereas RGC below uses the pairwise definition within connected clusters. This paper considers $h \in \{1, 2\}$, as illustrated in Figure 2.

3.2 Ricci Geometric Coherence

Ollivier-Ricci curvature measures the geometric agreement between node-local Markov neighborhoods. Based on this signal, we propose Ricci Geometric Coherence (RGC) to quantify the internal Ricci geometry of a coarse cluster. A desirable coarse cluster should group nodes with high mutual curvature, indicating that their Markov neighborhoods are geometrically compatible. For a connected candidate cluster $C \subseteq \mathcal{V}$ with $|C| \geq 2$, RGC is defined as

$$\mathrm{RGC}_h(C) = \frac{1}{\binom{|C|}{2}} \sum_{\{u,v\}\subset C} \kappa_{\alpha,h}(u, v),$$

The score is defined directly from Ollivier-Ricci curvature. A large $\mathrm{RGC}_h(C)$ indicates that the nodes in C have coherent Markov geometry, whereas a small value indicates that the cluster mixes structurally incompatible regions.

Let $\Pi_n^{\mathrm{conn}}(\mathcal{V})$ denote partitions into n connected clusters. For $\mathcal{C} \in \Pi_n^{\mathrm{conn}}(\mathcal{V})$, let $\mathcal{C}_{\mathrm{ns}} = \{C \in \mathcal{C} : |C| \geq 2\}$. When $c_G \leq n < N$, $\mathcal{C}_{\mathrm{ns}}$ is nonempty and all within-cluster distances are finite. The ideal objective averages Ricci geometric coherence over the coarse clusters created by nontrivial contractions.

$$\mathcal{C}^\star \in \operatorname*{arg\,max}_{\mathcal{C}\in\Pi_n^{\mathrm{conn}}(\mathcal{V})} J_{\mathrm{RGC}}^h(\mathcal{C}), \qquad J_{\mathrm{RGC}}^h(\mathcal{C}) = \frac{1}{|\mathcal{C}_{\mathrm{ns}}|} \sum_{C\in\mathcal{C}_{\mathrm{ns}}} \mathrm{RGC}_h(C). \tag{2}$$

Algorithm 1 Ollivier-Ricci Graph Coarsening

Require: Graph adjacency A, target node count $c_G \leq n < N$, laziness α, depth $h \in \{1, 2\}$
Ensure: Coarse adjacency A_c, assignment matrix Q, cluster index vector idx
Build P from A and set hop metric d_G on the graph support
Build local Markov measures $m_u^{\alpha,h}$ for all $u \in \mathcal{V}$
for each edge $(u, v) \in \mathcal{E}$ **do**
  Compute $\kappa_{\alpha,h}(u, v) = 1 - W_{1,d_G}(m_u^{\alpha,h}, m_v^{\alpha,h})$ since $d_G(u, v) = 1$
end for
Sort edges in nonincreasing order of $\kappa_{\alpha,h}$
Initialize union-find with one component per node and $F \leftarrow \emptyset$
for each sorted edge (u, v) **do**
  if u and v are in different union-find components **then**
    Add (u, v) to F and union the two components
    if number of active components equals n **then**
      break
    end if
  end if
end for
Construct idx from the final union-find components
Construct Q from idx
$A_c \leftarrow Q^\top A Q$
return A_c, Q, idx

Directly maximizing Eq. 2 over all target-size partitions is combinatorial and requires evaluating all node pairs inside candidate clusters. This motivates a contraction-based objective that retains a certified connection to RGC while avoiding an explicit partition search.

3.3 RGC Lower-bound Contraction Forest

Every node pair in a non-singleton forest component is connected by a unique path of selected contraction edges. A forest thus records how each connected coarse cluster is formed. For a forest F of $N - n$ input edges, let $\mathcal{C}(F)$ denote its component partition. We summarize the geometry of its contractions by the total Ricci deficit

$$D_h(F) = \sum_{e \in F} \big(1 - \kappa_{\alpha,h}(e)\big).$$

This yields the following forest-level RGC certificate.

$$\mathcal{L}_h(F) = 1 - D_h(F) = 1 - \sum_{e \in F} \big(1 - \kappa_{\alpha,h}(e)\big). \tag{3}$$

A large $\mathcal{L}_h(F)$ favors geometrically compatible contractions. Proposition 1 establishes

$$\mathcal{L}_h(F) \leq J^h_{\mathrm{RGC}}(\mathcal{C}(F)).$$

ORGC selects a target-size contraction forest by solving

$$F^\star = \arg \max_{\substack{F \subseteq \mathcal{E}:\ |F| = N-n, \\ F \text{ is a forest}}} \mathcal{L}_h(F). \tag{4}$$

The induced partition is $\mathcal{C}(F^\star)$. Since $|F| = N - n$, $\mathcal{L}_h(F) = n - N + 1 + \sum_{e \in F} \kappa_{\alpha,h}(e)$. Feasible forests form a fixed-rank truncation of a graphic matroid, so sorting input edges by curvature and accepting acyclic edges exactly maximizes Eq. 4. The exchange proof is in the appendix.

Algorithm 1 computes the curvatures of all input edges and processes them in descending order. It selects edges between distinct union–find components until completing $N - n$ contractions, which preserves acyclicity and keeps every output cluster connected.

3.4 Theoretical Properties

Proposition 1 (Contraction-Forest Lower Bound). *Let F be any contraction forest whose components are connected by input edges, and let $\mathcal{C}(F)$ be its component partition. Then all within-component pairwise curvatures are finite and*

$$\mathcal{L}_h(F) \leq J^h_{\mathrm{RGC}}(\mathcal{C}(F)).$$

Theorem 2 (Optimality of the Forest Lower Bound). *For $c_G \leq n < N$, the edge set returned by Algorithm 1 is an optimal solution to Eq. 4.*

Proofs and an empirical ranking comparison between $\mathcal{L}_h$ and all-pairs RGC are given in the appendix.

We next characterize when the curvature ordering is sufficient to preserve a prescribed partition.

Theorem 3 (Boundary Preservation under Forest-Rank Separation). *Let $\mathcal{B} = \{B_1, \ldots, B_K\}$ be a reference partition of $\mathcal{V}$, and let E_{in} and E_{out} denote its within-block and between-block input edges. For a fixed depth h, define*

$$\tau^h_{\mathrm{out}} = \max_{e \in E_{\mathrm{out}}} \kappa_{\alpha,h}(e), \quad E^{+,h}_{\mathrm{in}} = \left\{ e \in E_{\mathrm{in}} : \kappa_{\alpha,h}(e) > \tau^h_{\mathrm{out}} \right\}.$$

with $\tau^h_{\mathrm{out}} = -\infty$ when $E_{\mathrm{out}} = \emptyset$. Let $r_G(S)$ be the maximum cardinality of an acyclic subset of $S \subseteq \mathcal{E}$. If

$$r_G\left(E^{+,h}_{\mathrm{in}}\right) \geq N - n,$$

then the forest returned by ORGC satisfies

$$F_{\mathrm{ORGC}} \cap E_{\mathrm{out}} = \emptyset, \qquad \mathcal{C}_{\mathrm{ORGC}} \preceq \mathcal{B}.$$

Here $\mathcal{C}_{\mathrm{ORGC}} \preceq \mathcal{B}$ means each output cluster lies in one reference block. The rank condition ensures that edges above the largest boundary curvature can complete all $N - n$ contractions. For $n = K$, those edges contain a spanning tree in each block. The proof is in the appendix and holds for any fixed h.

3.5 Complexity Analysis

Let $M = |\mathcal{E}|$ and Δ be the maximum degree. ORGC solves one local transport problem per input edge. If $T_{\mathrm{loc}}(s)$ includes both transport and ground-cost construction on support size s, the curvature stage costs $O(MT_{\mathrm{loc}}(\Delta))$ for $h = 1$ and $O(MT_{\mathrm{loc}}(\Delta^2))$ for $h = 2$ in the worst case. Sorting adds $O(M \log M)$. We use POT's exact EMD routine (Flamary et al., 2021); The appendix provides a detailed analysis of the transport and distance computation costs.

4 Experiments

4.1 Evaluation Protocol

ORGC is evaluated through complementary protocols that measure intrinsic structural fidelity, downstream utility, and runtime, following prior graph coarsening evaluations (Loukas & Vandergheynst, 2018; Loukas, 2019; Jin et al., 2020; Chen et al., 2023). Coarsening is structure-only and does not use node features. ORGC uses $\alpha = 0.5$, and each local transport problem is solved by exact EMD through the POT routine `ot.emd2`, without entropic regularization (Flamary et al., 2021). The coarsening ratio r denotes the retained coarse-node fraction, with $n = \lceil rN \rceil$, so each reported ratio directly specifies the compression level. All stochastic results are reported over 10 repeated runs.

Following Chen et al. (2023), we evaluate against several classical graph coarsening baselines, including VNGC, VEGC, MGC, SGC (Loukas, 2019; Jin et al., 2020), and Kernel Graph Coarsening (KGC). FULL denotes the uncoarsened reference. ORGC-d1 and ORGC-d2 use $h = 1$ and $h = 2$, respectively.

4.2 Graph Structural Evaluation

Task and setup. This task isolates graph compression from downstream modeling choices by evaluating normalized-Laplacian structure after contraction. Karate Club (Zachary, 1977) and Les Miserables (Knuth, 1993) are obtained from NetworkX (Hagberg et al., 2008), while Cora and Citeseer (Sen et al., 2008) are obtained from DGL (Wang et al., 2019). All graphs are converted to unweighted, undirected, symmetric graphs, and ratios $r \in \{0.7, 0.5, 0.3\}$ are evaluated.

Metrics and configuration. The metrics are REE, RE, and J^1_{RGC}. REE compares the first n normalized-Laplacian eigenvalues of the original and coarse graphs, while RE measures the normalized squared Frobenius discrepancy between the original Laplacian and its lifted coarse counterpart (Loukas & Vandergheynst, 2018; Liu et al., 2018). Both errors are minimized. J^1_{RGC} is the non-singleton partition objective in Eq. 2, evaluated with $h = 1$ and maximized. Each entry is first averaged over 10 runs for each graph and ratio. For REE and RE, the main table reports equal-weight graph averages at each ratio, and Avg. is the equal-weight average over the three ratios. For J^1_{RGC}, Avg. first averages the 12 graph-ratio values within each run and then reports the mean across the 10 runs. The appendix provides the exact operators, formulas, and near-zero threshold.

Table 1: Structural fidelity on Karate Club, Les Miserables, Cora, and Citeseer. REE and RE are lower-is-better. J^1_{RGC} is higher-is-better. Bold and underline indicate the best and second-best coarsening results in each column.

	REE↓				RE↓				J^1_{RGC} ↑			
Method	r=0.7	r=0.5	r=0.3	Avg.	r=0.7	r=0.5	r=0.3	Avg.	r=0.7	r=0.5	r=0.3	Avg.
VNGC	0.210	0.359	0.821	0.463	0.909	1.515	2.869	1.764	0.364	0.345	0.372	0.360
VEGC	0.120	0.255	0.847	0.407	0.801	1.486	2.242	1.509	0.424	0.374	0.414	0.404
MGC	0.257	0.536	1.784	0.859	1.110	2.422	4.122	2.551	0.433	0.374	0.319	0.375
SGC	0.277	0.575	1.504	0.786	8.687	9.758	9.448	9.297	0.369	0.367	0.350	0.362
KGC	0.476	1.275	3.634	1.795	50.370	130.497	163.670	114.846	0.545	0.454	0.377	0.459
ORGC-d1	**0.108**	0.205	0.488	0.267	0.742	**1.187**	1.769	**1.233**	0.550	0.459	0.436	0.482
ORGC-d2	0.109	**0.195**	**0.472**	**0.259**	**0.737**	1.199	**1.769**	1.235	**0.557**	**0.488**	**0.449**	**0.498**

Results. Table 1 summarizes the structural errors over the four graphs and the aggregate non-singleton RGC score. ORGC gives the strongest aggregate REE and RE among compressed methods. ORGC-d2 also attains the highest aggregate non-singleton RGC score.

4.3 Graph Classification

Table 2: NetLSD-1NN graph classification accuracy at $r = 0.2$ (mean ± std over 10 runs). Bold and underline mark the best and second-best compressed methods; FULL is the uncoarsened reference.

Method	MUTAG	PTC	PROTEINS	MSRC	IMDB	Tumblr	Mean
VNGC	73.42±1.80	52.27±1.81	64.13±1.73	14.04±1.15	52.13±1.67	50.24±2.18	51.04
VEGC	83.34±1.38	54.30±1.99	60.84±0.75	**17.42**±0.89	66.00±1.30	49.33±0.92	55.20
MGC	69.83±0.95	55.18±1.53	64.20±0.91	16.73±1.28	66.42±2.27	48.34±2.12	53.45
SGC	82.25±1.92	54.95±2.11	63.98±0.85	15.12±2.82	65.73±1.48	49.19±1.64	55.20
KGC	80.55±2.72	52.87±3.12	64.36±1.93	17.20±3.66	65.53±2.05	49.57±3.40	55.01
ORGC-d1	**84.67**±1.09	53.10±1.10	**65.26**±0.71	16.69±1.34	66.28±1.35	50.85±2.19	56.14
ORGC-d2	83.48±1.98	**55.35**±1.55	63.85±0.82	17.36±0.74	**66.48**±1.17	**51.74**±2.15	**56.38**
FULL	85.46±0.92	54.67±1.46	66.92±0.63	20.77±1.14	69.87±1.25	51.71±2.44	58.23

Task and setup. This task tests whether coarsening preserves graph-level discriminative structure. MUTAG, PTC, PROTEINS, MSRC, IMDB, and Tumblr are taken from TUDataset (Morris et al., 2020) and coarsened at $r = 0.2$. Each coarsened graph is represented by NetLSD (Tsitsulin et al.,

2018) and classified with 1-nearest neighbor, so accuracy evaluates the structural descriptor retained after compression.

Results. Table 2 reports NetLSD-1NN accuracy at $r = 0.2$. ORGC achieves the strongest aggregate classification performance among compressed methods, suggesting that curvature-guided contractions effectively retain graph-level structural information relevant to distinguishing graph classes.

4.4 Molecular Regression

Task and setup. This task evaluates coarsening as preprocessing for molecular property prediction with a graph convolutional network (Kipf & Welling, 2017; Gilmer et al., 2017). AQSOL (Sorkun et al., 2019; Dwivedi et al., 2023) and ZINC (Irwin et al., 2012; Dwivedi et al., 2023) are coarsened at $r = 0.3$, after which the same GCGCN training pipeline is applied to all methods. Test MAE is the primary metric, and Train MAE characterizes fitting behavior.

Table 3: Molecular regression with GCGCN at $r = 0.3$ (mean $\pm$ std). FULL is the uncoarsened reference.

	AQSOL		ZINC	
Method	Test MAE	Train MAE	Test MAE	Train MAE
VNGC	1.383±0.016	0.641±0.025	0.695±0.010	0.413±0.025
VEGC	1.349±0.019	0.680±0.026	0.633±0.007	0.430±0.011
MGC	1.446±0.014	0.646±0.039	0.676±0.007	0.399±0.018
SGC	1.453±0.011	0.658±0.039	0.674±0.007	0.415±0.020
KGC	1.409±0.037	0.692±0.031	0.715±0.011	0.480±0.019
ORGC-d1	**1.315**±0.016	**0.577**±0.029	0.615±0.007	0.388±0.019
ORGC-d2	1.320±0.015	0.591±0.026	**0.599**±0.008	**0.372**±0.020
FULL	1.383±0.024	0.529±0.015	0.391±0.006	0.283±0.015

Results. Table 3 reports Test and Train MAE on AQSOL and ZINC, where lower values indicate lower prediction error. ORGC obtains the lowest compressed-graph Test MAE on both datasets and improves Train MAE relative to the compressed baselines. FULL remains the uncoarsened reference, especially on ZINC, while the variation across depths indicates that the appropriate geometric range is data dependent.

4.5 Analysis

Runtime Scaling. We evaluate runtime on synthetic Barabási–Albert graphs with average degree approximately 8, 50–5000 nodes, and $r = 0.5$. We report mean coarsening time over 5 runs, excluding graph generation and downstream learning. All methods receive the same graph in each run. Baselines use their original implementations, and curves terminate at the largest size completed within the common time budget.

Table 4: Mean Markov transition consistency errors on Citeseer.

	$r = 0.7$		$r = 0.5$		$r = 0.3$	
Method	E_1	E_2	E_1	E_2	E_1	E_2
VNGC	.1311	.1629	.2047	.2099	.2233	.2246
VEGC	.1202	.1280	.1984	.1955	.1844	.2000
MGC	.0602	.0622	.1410	.1447	.2157	.2227
SGC	.1371	.1530	.2289	.2173	.2550	.2029
KGC	.1906	.1933	.3269	.3306	.4230	.4096
ORGC-d1	.0615	.0601	.1401	.1196	.1560	.1311
ORGC-d2	**.0595**	**.0563**	**.1374**	**.1161**	**.1405**	**.1232**

Figure 3 shows that ORGC-d1 achieves the lowest runtime across the tested sizes, consistent with the small one-hop transport supports of low-degree nodes. ORGC-d2 is slower because its broader Markov neighborhoods enlarge local transport problems, particularly near high-degree hubs, but remains competitive at larger scales. Detailed runtimes are provided in the appendix.

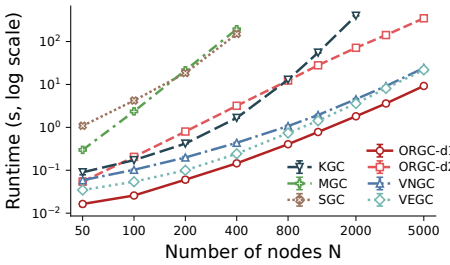


Figure 3: Runtime scaling on Barabasi-Albert graphs, averaged over 5 runs.

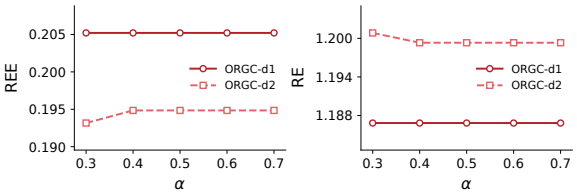


Figure 4: Sensitivity to α at $r = 0.5$. Curves show mean REE and RE over four graphs and 10 paired runs.

Parameter Stability. We vary $\alpha \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ on Karate Club, Les Miserables, Cora, and Citeseer at $r = 0.5$, keeping the remaining ORGC-d1 and ORGC-d2 settings unchanged. Each configuration uses 10 runs. For each run, we first average each metric equally over the four graphs, then compute the mean and standard deviation across seeds. Figure 4 plots the mean REE and RE. Both metrics remain stable over the tested range, with nearly unchanged values for ORGC-d1 and small variations for ORGC-d2, indicating limited sensitivity to α in this setting.

Markov geometry preservation. To assess Markov geometry preservation independently of curvature and RGC, we evaluate random-walk transition consistency on Citeseer at $r \in \{0.7, 0.5, 0.3\}$. Let $Q \in \{0,1\}^{N\times n}$ denote the cluster assignment matrix and $P = D^{-1}A$ the original random-walk transition matrix. For all methods, we construct $A_c = Q^\top AQ$ and $P_c = D_c^{-1}A_c$, retaining coarse self-loops to represent within-cluster transitions. Zero-degree nodes are assigned self-transition probability one. We compare walking before aggregation, P^tQ, with walking after aggregation, QP_c^t, using the node-averaged total variation error. This discrepancy quantifies how closely the coarse random walk reproduces the original graph's cluster-level transition probabilities.

$$E_t = \frac{1}{2N}\sum_{i=1}^{N}\sum_{c=1}^{n}\left|(P^tQ)_{ic} - (QP_c^t)_{ic}\right|, \quad t \in \{1,2\}.$$

Each original node receives equal weight, and lower $E_t \in [0,1]$ indicates better transition consistency. As shown in Table 4, ORGC-d1 and ORGC-d2 generally achieve the lowest errors among the evaluated methods, supporting their preservation of short-range Markov dynamics across compression ratios and walk lengths.

5 Limitations

Although ORGC uses an effective geometry-aware signal, its curvature scores are computed from bounded-depth Markov neighborhoods and remain local. Increasing h broadens the geometric range but enlarges transport supports, while a fixed depth may not match heterogeneous structural scales. Moreover, the current structure-only implementation does not use node or edge attributes and may miss feature-level distinctions between structurally similar neighborhoods.

6 Conclusion

ORGC uses Ollivier–Ricci curvature to construct geometrically coherent coarse clusters. Its forest lower bound connects a pairwise cluster objective to an exactly solvable edge-selection problem, while the rank condition characterizes boundary preservation. Across structural, downstream, and runtime evaluations, ORGC combines strong graph fidelity with efficient coarsening. The candidate-forest diagnostic further supports the surrogate as a ranking signal for all-pairs RGC.

Future research will derive probabilistic conditions for forest-rank separation under random graph models, combine structural and feature costs through fused Gromov-Wasserstein constructions (Vayer et al., 2019), extend ORGC to directed graphs through asymmetric curvature and contraction rules, and integrate the method into graph neural networks as a learnable coarsening component.

Statement on the Use of AI Tools

Generative AI tools assisted with manuscript drafting and revision, LaTeX formatting, figure preparation, code development, and the presentation and checking of experimental and theoretical material. The authors remain responsible for the claims, proofs, experiments, and final text.

Ethics Statement

This work uses publicly available graph datasets and does not collect or process new human-subject data. The reported results may inherit biases, sampling artifacts, and other limitations present in the underlying datasets and graph models.

Reproducibility Statement

The experimental protocol, coarsening settings, training hyperparameters, evaluation metrics, and random seeds are documented in Section 4.1 and Appendix B.1. The implementation and experiment configuration are included in the supplementary material.

References

Albert-Laszlo Barabasi and Zoltan N. Oltvai. Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004. doi: 10.1038/nrg1272.

Chen Cai, Dingkang Wang, and Yusu Wang. Graph coarsening with neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uxpzitPEooJ.

Yifan Chen, Rentian Yao, Yun Yang, and Jie Chen. A gromov-Wasserstein geometric view of spectrum-preserving graph coarsening. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5257–5281. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23ak.html.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, pp. 2292–2300, 2013.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

Amy Feng and Melanie Weber. Graph pooling via Ricci flow. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=xpBHp9WFvk.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021. URL https://www.jmlr.org/papers/v22/20-451.html.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, pp. 11–15, 2008.

John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52 (7):1757–1768, 2012. doi: 10.1021/ci3001277.

Yu Jin, Andreas Loukas, and Joseph JaJa. Graph coarsening with preserved spectral properties. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4452–4462. PMLR, 2020. URL https://proceedings.mlr.press/v108/jin20a.html.

Jürgen Jost and Shiping Liu. Ollivier's ricci curvature, local clustering and curvature-dimension inequalities on graphs. *Discrete & Computational Geometry*, 51(2):300–322, 2014. doi: 10.1007/s00454-013-9558-1.

L. V. Kantorovich. On the translocation of masses. *Management Science*, 5(1):1–4, 1958. doi: 10.1287/mnsc.5.1.1.

George Karypis and Vipin Kumar. Analysis of multilevel graph partitioning. In *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing*, pp. 29, 1995. doi: 10.1145/224170.224229.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Donald E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. ACM Press, 1993.

Yong Lin, Linyuan Lu, and Shing-Tung Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal*, 63(4):605–627, 2011. doi: 10.2748/tmj/1325886283.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4IT2pgc9v6.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, and Chuan Shi. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):5023–5044, 2025. doi: 10.1109/TPAMI.2025.3548729.

Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph summarization methods and applications: A survey. *ACM Computing Surveys*, 51(3):62:1–62:34, 2018. doi: 10.1145/3186727.

Andreas Loukas. Graph reduction with spectral and cut guarantees. *Journal of Machine Learning Research*, 20(116):1–42, 2019. URL https://www.jmlr.org/papers/v20/18-680.html.

Andreas Loukas and Pierre Vandergheynst. Spectrally approximating large graphs with smaller graphs. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3237–3246. PMLR, 2018. URL https://proceedings.mlr.press/v80/loukas18a.html.

Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487, 2011. doi: 10.1007/s10208-011-9093-5.

Oscar Méndez-Lucio, Christos A. Nicolaou, and Berton Earnshaw. MolE: A foundation model for molecular graphs using disentangled attention. *Nature Communications*, 15(1):9431, 2024. doi: 10.1038/s41467-024-53751-y.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001. doi: 10.1073/pnas.98.2.404.

Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009. doi: 10.1016/j.jfa.2008.11.001.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2664–2672. PMLR, 2016.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008. doi: 10.1609/aimag.v29i3.2157.

Murat Cihan Sorkun, Abhishek Khetan, and Süleyman Er. AqSolDB, a curated reference set of aqueous solubility and 2d descriptors for a diverse set of compounds. *Scientific Data*, 6(1):143, 2019. doi: 10.1038/s41597-019-0151-1.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/10a6bdcabbd5a3d36b760daa295f63c1-Abstract-Conference.html.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. GraphGPT: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 491–500. ACM, 2024. doi: 10.1145/3626772.3657775.

Carlos A Taveras, Santiago Segarra, and César A Uribe. Gromov-wasserstein graph coarsening. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 646–650. IEEE, 2026.

Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander M. Bronstein, and Emmanuel Müller. NetLSD: Hearing the shape of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2347–2356. ACM, 2018.

Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Optimal transport for structured data with application on graphs. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6275–6284. PMLR, 2019. URL https://proceedings.mlr.press/v97/titouan19a.html.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der Mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-540-71050-9.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994. doi: 10.1017/CBO9780511815478.

Shushan Wu, Huimin Cheng, Jiazhang Cai, Ping Ma, and Wenxuan Zhong. Subsampling in large graphs using ricci curvature. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=w9WUQkBvpI.

Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

A Additional Theoretical Results

A.1 Depth-h Non-backtracking Markov Measure

We specify the endpoint distributions used in the depth-h measure from the main paper. For the first transition from u, let $p_{uv}^{(1)} = P_{uv}$. After traversing a directed edge (x, y), define the non-backtracking continuation kernel

$$P_{yz}^{\neg x} = \frac{A_{yz}}{\sum_{w \in \mathcal{N}(y) \setminus \{x\}} A_{yw}}, \qquad z \in \mathcal{N}(y) \setminus \{x\},$$

whenever the denominator is positive. For an unweighted graph, this kernel is uniform over $\mathcal{N}(y) \setminus \{x\}$. If this set is empty, the path remains at y for the remaining transitions.

For a non-backtracking path $\pi = (x_0 = u, x_1, \ldots, x_t)$, its probability is

$$\Pr(\pi) = P_{ux_1} \prod_{j=1}^{t-1} P_{x_j x_{j+1}}^{\neg x_{j-1}}.$$

The endpoint probability $p_{uv}^{(t)}$ is the sum of $\Pr(\pi)$ over paths ending at v, including paths held at a terminal endpoint. At depths $0, \ldots, h-1$, a fraction α of the surviving mass stops and the remaining fraction continues. All mass that reaches depth h stops there, which gives

$$m_u^{\alpha,h} = \alpha \delta_u + \sum_{t=1}^{h-1} \alpha(1-\alpha)^t \sum_{v \in \mathcal{V}} p_{uv}^{(t)} \delta_v + (1-\alpha)^h \sum_{v \in \mathcal{V}} p_{uv}^{(h)} \delta_v.$$

For $h = 1$, this reduces to the standard lazy measure

$$m_u^{\alpha,1} = \alpha \delta_u + (1-\alpha) \sum_{v \in \mathcal{V}} P_{uv} \delta_v.$$

For $h = 2$, it retains intermediate mass on one-hop endpoints and propagates the remainder through one additional non-backtracking transition. The experiments use $\alpha = 1/2$, so the surviving mass is split equally between the one-hop and two-hop endpoints. This construction matches the evaluated depth-1 and depth-2 implementations and does not use ordinary backtracking transitions.

A.2 Proof of Proposition 1

Proof. Let $\mathcal{C}_{\mathrm{ns}}(F) = \{C \in \mathcal{C}(F) : |C| \geq 2\}$. Since $n < N$, every feasible forest contains at least one selected edge and $\mathcal{C}_{\mathrm{ns}}(F)$ is nonempty. For a component $C \in \mathcal{C}_{\mathrm{ns}}(F)$, define

$$D_C = \sum_{e \in F_C} \big(1 - \kappa_{\alpha,h}(e)\big).$$

For any unordered pair $\{u, v\} \subset C$, let $P_{F_C}(u, v)$ denote the unique path between u and v in the tree F_C. Since each edge $e = (a, b) \in F_C$ is an original graph edge, $d_G(a, b) = 1$, and the definition of Ollivier-Ricci curvature gives

$$W_{1,d_G}(m_a^{\alpha,h}, m_b^{\alpha,h}) = 1 - \kappa_{\alpha,h}(a, b).$$

By the triangle inequality of W_1 along the tree path,

$$W_{1,d_G}(m_u^{\alpha,h}, m_v^{\alpha,h}) \leq \sum_{e \in P_{F_C}(u,v)} \big(1 - \kappa_{\alpha,h}(e)\big) \leq D_C.$$

Because $u \neq v$, the hop distance satisfies $d_G(u, v) \geq 1$. Hence

$$1 - \kappa_{\alpha,h}(u, v) = \frac{W_{1,d_G}(m_u^{\alpha,h}, m_v^{\alpha,h})}{d_G(u, v)} \leq D_C,$$

which implies $\kappa_{\alpha,h}(u,v) \geq 1 - D_C$. Averaging over all unordered pairs inside C gives

$$\mathrm{RGC}_h(C) \geq 1 - D_C.$$

Since $D_h(F) = \sum_{C \in \mathcal{C}_{\mathrm{ns}}(F)} D_C \geq 0$ and $|\mathcal{C}_{\mathrm{ns}}(F)| \geq 1$, averaging the componentwise inequality gives

$$\begin{aligned} J^h_{\mathrm{RGC}}(\mathcal{C}(F)) &= \frac{1}{|\mathcal{C}_{\mathrm{ns}}(F)|} \sum_{C \in \mathcal{C}_{\mathrm{ns}}(F)} \mathrm{RGC}_h(C) \\ &\geq 1 - \frac{D_h(F)}{|\mathcal{C}_{\mathrm{ns}}(F)|} \geq 1 - D_h(F) \\ &= \mathcal{L}_h(F). \end{aligned}$$

□

A.3 Proof of Theorem 2

Proof. Let $q = N - n$ be the number of contractions required by the target coarse size, and let $\mathcal{F}_{N-n} = \{F \subseteq \mathcal{E} : |F| = q,\ (\mathcal{V}, F) \text{ is acyclic}\}$. Since $c_G \leq n$, this set is nonempty. Define

$$\mathcal{I} = \{F \subseteq \mathcal{E} :\ (\mathcal{V}, F) \text{ is acyclic}\}.$$

The family $\mathcal{I}$ is the independent-set family of the graphic matroid on G. The feasible forests in the main-paper objective are exactly the independent sets in $\mathcal{I}$ with cardinality q. Equivalently, they are the bases of the rank-q truncation of this graphic matroid.

For any $F \in \mathcal{F}_{N-n}$, the forest has n connected components and $N-n$ selected edges. The definition of $\mathcal{L}_h(F)$ therefore gives

$$\mathcal{L}_h(F) = 1 - \sum_{e \in F} \big(1 - \kappa_{\alpha,h}(e)\big) = n - N + 1 + \sum_{e \in F} \kappa_{\alpha,h}(e).$$

The additive term $n - N + 1$ is constant over $\mathcal{F}_{N-n}$, so maximizing $\mathcal{L}_h(F)$ is equivalent to maximizing $\sum_{e \in F} \kappa_{\alpha,h}(e)$.

Algorithm 1 in the main paper sorts all edges in nonincreasing order of $w(e) = \kappa_{\alpha,h}(e)$ and scans them in that order. It adds an edge if and only if the edge preserves acyclicity of the selected set, and it stops after q selected edges. Therefore the algorithm is precisely the greedy algorithm for the equivalent linear objective over the rank-q truncation of a graphic matroid.

For completeness, we spell out the exchange argument. Let

$$F_{\mathrm{ORGC}} = \{g_1, \ldots, g_q\}$$

be the greedy edge set in the order selected by the algorithm. Let O_0 be a maximizer of the equivalent linear objective over feasible edge sets of size q. We prove by induction that, after the i-th greedy selection, there exists an optimal feasible edge set O_i such that

$$\{g_1, \ldots, g_i\} \subseteq O_i.$$

The claim is true for $i = 0$ by taking O_0 itself. Suppose it is true for $i - 1$. If $g_i \in O_{i-1}$, set $O_i = O_{i-1}$ and the claim follows. Otherwise, since O_{i-1} is a basis of the rank-q truncation and $\{g_1, \ldots, g_{i-1}, g_i\}$ is independent, the matroid basis-exchange property gives an edge

$$f \in O_{i-1} \setminus \{g_1, \ldots, g_{i-1}\}$$

such that

$$O_i = (O_{i-1} \setminus \{f\}) \cup \{g_i\}$$

is also feasible. Moreover, $\{g_1, \ldots, g_{i-1}, f\}$ is independent because it is a subset of the independent set O_{i-1}. Hence f was a feasible candidate at the moment when greedy selected g_i. Since greedy scans edges by nonincreasing weight and chooses the feasible edge with largest available weight, we have $w(g_i) \geq w(f)$. Therefore

$$\sum_{e \in O_i} w(e) = \sum_{e \in O_{i-1}} w(e) - w(f) + w(g_i) \geq \sum_{e \in O_{i-1}} w(e).$$

Because O_{i-1} is optimal, O_i is also optimal. This completes the induction.

At $i = q$, the greedy set F_{ORGC} is contained in an optimal feasible set O_q, and both sets have cardinality q. Thus $F_{\mathrm{ORGC}} = O_q$, so Algorithm 1 in the main paper maximizes $\sum_{e\in F} \kappa_{\alpha,h}(e)$ over all feasible contraction forests of size $N - n$. By the equivalence above, it also maximizes $\mathcal{L}_h(F)$. This proves Theorem 2. □

A.4 Proof of Theorem 3

Proof. Let $q = N - n$ and let $H^+ = (\mathcal{V}, E_{\mathrm{in}}^{+,h})$ be the subgraph formed by within-block edges whose curvature is strictly greater than τ_{out}^h. Every edge in $E_{\mathrm{in}}^{+,h}$ precedes every edge in E_{out} in the nonincreasing curvature order used by ORGC.

Consider the Kruskal scan restricted to the prefix $E_{\mathrm{in}}^{+,h}$. The accepted edges form a maximal forest of H^+, whose cardinality equals the graphic-matroid rank $r_G(E_{\mathrm{in}}^{+,h})$. Under the theorem condition, this prefix contains an acyclic subset of at least q edges, so ORGC accepts q edges and reaches the target size before scanning any edge in E_{out}. Consequently,

$$F_{\mathrm{ORGC}} \subseteq E_{\mathrm{in}}^{+,h} \subseteq E_{\mathrm{in}},$$

which proves $F_{\mathrm{ORGC}} \cap E_{\mathrm{out}} = \emptyset$.

Each union–find component is initially a singleton and is merged only through within-block edges. Every final component is therefore contained in one block of $\mathcal{B}$, which gives

$$\mathcal{C}_{\mathrm{ORGC}} \preceq \mathcal{B}.$$

□

A.5 Detailed Complexity Analysis

Let $N = |\mathcal{V}|$, $M = |\mathcal{E}|$, $\Delta = \max_{u\in\mathcal{V}} d_u$, and $q = N - n$. Define

$$b_h(u) = |\operatorname{supp}(m_u^{\alpha,h})|, \qquad b_h = \max_{u\in\mathcal{V}} b_h(u),$$

and let

$$s_{uv}^{(h)} = \left|\operatorname{supp}(m_u^{\alpha,h}) \cup \operatorname{supp}(m_v^{\alpha,h})\right|$$

be the combined support size for input edge (u, v). The local transport problem depends on $s_{uv}^{(h)}$ rather than directly on N.

The sparse transition matrix P is constructed in $O(M)$ time. For $h = 1$, the measure at u is supported on $\{u\} \cup \mathcal{N}(u)$, so

$$b_1(u) \le d_u + 1, \qquad b_1 = O(\Delta).$$

For $h = 2$, the support is contained in the non-backtracking two-hop neighborhood, and

$$b_2(u) \le 1 + d_u + \sum_{x\in\mathcal{N}(u)} (d_x - 1) \le 1 + d_u\Delta = O(\Delta^2),$$

which gives $b_2 = O(\Delta^2)$ in the worst case. Moreover, $s_{uv}^{(h)} \le b_h(u) + b_h(v) \le 2b_h$.

Ground-cost construction must be counted separately from the transport solve. For an input edge (u, v), every pair of support nodes is at graph distance at most $2h + 1$ through (u, v). The implementation therefore caches truncated hop distances with cutoffs 3 for $h = 1$ and 5 for $h = 2$. Let $T_{\mathrm{dist}}^{(h)}$ be the total cost of building this cache. Once cached, extracting the local ground-cost block requires $O((s_{uv}^{(h)})^2)$ entries.

Let $T_{\mathrm{EMD}}(s)$ denote the runtime of the exact EMD solver on combined support size s. The implementation calls `ot.emd2` from POT (Flamary et al., 2021), which uses a network-simplex backend. EMD is an optimization problem rather than a single algorithm, so its precise complexity depends on the backend, initialization, and support geometry. We therefore retain $T_{\mathrm{EMD}}(s)$ instead of assigning a universal exponent to exact EMD.

The complete curvature cost is

$$C_h = O\left(T_{\text{dist}}^{(h)} + \sum_{(u,v)\in\mathcal{E}} \left[T_{\text{EMD}}(s_{uv}^{(h)}) + (s_{uv}^{(h)})^2\right]\right).$$

The support bounds give

$$\text{ORGC-d1:} \quad C_1 = O\Big(T_{\text{dist}}^{(1)} + M[T_{\text{EMD}}(O(\Delta)) + \Delta^2]\Big),$$
$$\text{ORGC-d2:} \quad C_2 = O\Big(T_{\text{dist}}^{(2)} + M[T_{\text{EMD}}(O(\Delta^2)) + \Delta^4]\Big).$$

These expressions expose the broader support of depth 2 without tying the analysis to one EMD implementation.

After curvature computation, sorting costs $O(M \log M)$ and the union–find scan costs $O(M\alpha(N))$, where $\alpha(\cdot)$ is the inverse Ackermann function. Constructing the sparse assignment matrix and $A_c = Q^\top AQ$ costs $O(N + M)$. The full runtime is therefore

$$O(M + C_h + M \log M + M\alpha(N) + N).$$

For a sparse bounded-degree graph and fixed h, the local supports and truncated-distance neighborhoods are bounded. Under these conditions, $T_{\text{dist}}^{(h)} = O(N + M)$ and the curvature stage is $O(M)$ for a fixed exact solver implementation, while sorting contributes $O(M \log M)$.

Implementation notes. Curvature scores are independent across input edges and can be computed in parallel. The evaluated implementation uses exact EMD because its local transport problems are small. Entropic regularization such as Sinkhorn (Cuturi, 2013) and more aggressive distance caching can reduce cost on high-degree graphs, but they change the transport solver and are not used for the reported ORGC results.

Table 5: Full graph-by-ratio REE results on Karate Club, Les Miserables, Cora, and Citeseer. Results are mean ± standard deviation over 10 repeated runs. Lower is better.

	Karate Club			Les Miserables		
Method	r=0.7	r=0.5	r=0.3	r=0.7	r=0.5	r=0.3
VNGC	0.26±0.05	0.50±0.11	0.58±0.01	0.12±0.00	0.21±0.00	0.45±0.00
VEGC	0.12±0.00	0.16±0.00	0.35±0.00	0.10±0.00	0.16±0.00	0.35±0.02
MGC	0.21±0.00	0.36±0.00	0.65±0.00	0.12±0.00	0.19±0.00	0.40±0.00
SGC	0.20±0.00	0.31±0.03	0.35±0.11	0.12±0.01	0.21±0.00	0.39±0.04
KGC	0.23±0.03	0.42±0.06	0.90±0.13	0.22±0.02	0.56±0.04	1.33±0.07
ORGC-d1	0.13±0.00	0.21±0.00	0.36±0.03	0.08±0.00	0.13±0.00	0.22±0.00
ORGC-d2	0.13±0.00	0.18±0.00	0.26±0.00	0.08±0.00	0.13±0.00	0.18±0.00

	Cora			Citeseer		
Method	r=0.7	r=0.5	r=0.3	r=0.7	r=0.5	r=0.3
VNGC	0.16±0.00	0.33±0.00	0.74±0.04	0.29±0.01	0.41±0.01	1.52±0.12
VEGC	0.13±0.00	0.31±0.01	0.93±0.05	0.13±0.00	0.39±0.02	1.76±0.08
MGC	0.27±0.00	0.58±0.00	1.32±0.00	0.42±0.00	1.02±0.00	4.76±0.00
SGC	0.25±0.00	0.67±0.01	0.45±0.01	0.54±0.01	1.11±0.01	4.84±0.04
KGC	0.44±0.01	1.11±0.03	2.82±0.05	1.01±0.10	3.01±0.34	9.49±0.43
ORGC-d1	0.12±0.00	0.26±0.00	0.63±0.00	0.11±0.00	0.22±0.00	0.74±0.00
ORGC-d2	0.12±0.00	0.26±0.00	0.66±0.00	0.11±0.00	0.22±0.00	0.79±0.00

B Additional Experimental Results

B.1 Reproducibility Details

All stochastic experiments use the fixed seeds 12 through 21. Results over repeated runs are reported as the mean and standard deviation across these ten seeds. Experiments are conducted on a server with 256 GB RAM, a 64-core Intel Xeon Gold 5218 CPU, and an NVIDIA RTX 3090 GPU.

B.2 Structural Fidelity with Variability

Tables 5–7 report the complete graph-by-ratio results underlying Table 1 of the main paper. Each entry is the mean ± standard deviation over 10 repeated runs. The aggregated values in the main paper are computed from the unrounded run-level results by assigning equal weight to the four graphs at each ratio and then averaging over the three ratios.

B.3 Cross-task Discussion

The evaluations test complementary effects of the same structural objective: intrinsic Laplacian and Ricci fidelity, descriptor-based classification, GNN regression, and coarsening runtime. ORGC performs strongly across these settings without adapting its objective to a downstream model. The depth results should be interpreted as a locality tradeoff rather than task-specific dominance: increasing h broadens geometric context but enlarges transport supports and need not improve every graph. Thus h can be selected with the validation protocol used for the coarsening ratio. Comparisons among compressed methods further isolate performance under the same retained-node budget, while FULL remains an uncompressed reference.

B.4 Forest-Objective Informativeness

We evaluate whether the global linear objective $\mathcal{L}_h(F)$ preserves the ranking induced by the non-singleton all-pairs objective $J^h_{\mathrm{RGC}}(\mathcal{C}(F))$. The evaluation uses all 1,000 IMDB-BINARY graphs at $r = 0.2$ and considers $h \in \{1, 2\}$ separately. For each graph and depth, we construct $K =$

Table 6: Full graph-by-ratio RE results on Karate Club, Les Miserables, Cora, and Citeseer. Results are mean ± standard deviation over 10 repeated runs. Lower is better.

	Karate Club			Les Miserables		
Method	r=0.7	r=0.5	r=0.3	r=0.7	r=0.5	r=0.3
VNGC	0.87±0.09	1.41±0.31	1.72±0.08	0.75±0.00	1.23±0.00	2.80±0.00
VEGC	0.71±0.00	1.47±0.00	1.98±0.00	0.69±0.00	1.51±0.00	2.14±0.09
MGC	1.08±0.00	2.43±0.00	3.63±0.00	1.24±0.00	1.55±0.00	1.95±0.00
SGC	1.07±0.13	1.93±0.26	2.48±0.43	0.82±0.09	1.91±0.01	2.59±0.32
KGC	2.08±0.51	5.18±0.67	7.28±1.33	5.66±0.45	11.18±1.59	11.42±1.42
ORGC-d1	0.64±0.00	1.18±0.00	1.82±0.04	0.68±0.00	0.98±0.00	1.37±0.00
ORGC-d2	0.64±0.00	1.24±0.00	1.70±0.00	0.66±0.00	0.94±0.00	1.36±0.00

	Cora			Citeseer		
Method	r=0.7	r=0.5	r=0.3	r=0.7	r=0.5	r=0.3
VNGC	0.90±0.01	1.84±0.16	4.02±2.08	1.12±0.13	1.58±0.02	2.93±1.03
VEGC	0.83±0.00	1.45±0.01	2.61±0.22	0.98±0.00	1.51±0.01	2.23±0.03
MGC	1.01±0.00	2.40±0.00	4.69±0.00	1.12±0.00	3.31±0.00	6.22±0.00
SGC	14.87±0.75	18.07±3.98	8.74±1.57	17.98±0.95	17.13±2.04	23.98±5.28
KGC	100.07±3.76	263.42±8.16	341.08±9.84	93.67±14.02	242.21±38.48	294.91±26.07
ORGC-d1	0.77±0.00	1.26±0.00	2.09±0.02	0.88±0.00	1.33±0.00	1.80±0.03
ORGC-d2	0.77±0.00	1.26±0.00	2.19±0.00	0.88±0.00	1.35±0.00	1.82±0.00

Table 7: Full graph-by-ratio J^1_{RGC} results on Karate Club, Les Miserables, Cora, and Citeseer. Results are mean ± standard deviation over 10 repeated runs. Higher is better.

	Karate Club			Les Miserables		
Method	r=0.7	r=0.5	r=0.3	r=0.7	r=0.5	r=0.3
VNGC	0.31±0.01	0.31±0.01	0.35±0.04	0.26±0.00	0.29±0.00	0.33±0.00
VEGC	0.43±0.00	0.41±0.00	0.38±0.00	0.42±0.00	0.38±0.00	0.38±0.01
MGC	0.47±0.00	0.44±0.00	0.41±0.00	0.54±0.00	0.50±0.00	0.41±0.00
SGC	0.42±0.02	0.40±0.02	0.40±0.02	0.47±0.02	0.45±0.00	0.38±0.02
KGC	0.40±0.07	0.38±0.05	0.33±0.06	0.40±0.08	0.28±0.10	0.25±0.08
ORGC-d1	0.46±0.00	0.41±0.01	0.39±0.00	0.56±0.00	0.45±0.00	0.44±0.00
ORGC-d2	0.45±0.00	0.45±0.00	0.41±0.00	0.57±0.00	0.50±0.00	0.43±0.00

	Cora			Citeseer		
Method	r=0.7	r=0.5	r=0.3	r=0.7	r=0.5	r=0.3
VNGC	0.37±0.01	0.27±0.00	0.27±0.00	0.51±0.01	0.51±0.01	0.54±0.01
VEGC	0.28±0.01	0.23±0.01	0.29±0.02	0.56±0.00	0.48±0.01	0.60±0.01
MGC	0.32±0.00	0.24±0.00	0.16±0.00	0.40±0.00	0.32±0.00	0.30±0.00
SGC	0.13±0.00	0.25±0.00	0.27±0.00	0.47±0.00	0.37±0.00	0.36±0.00
KGC	0.50±0.02	0.40±0.03	0.33±0.02	0.88±0.04	0.75±0.07	0.59±0.08
ORGC-d1	0.47±0.00	0.39±0.00	0.34±0.00	0.71±0.00	0.60±0.00	0.58±0.00
ORGC-d2	0.49±0.00	0.40±0.00	0.36±0.00	0.72±0.00	0.60±0.00	0.60±0.00

34 feasible forests with the same target size. The candidate set contains one ORGC forest, one reverse-curvature forest, 16 random Kruskal forests, and 16 noisy-curvature forests. The noisy forests perturb edge curvatures with Gaussian noise at $\sigma \in \{0.05, 0.1, 0.2, 0.5\}$, with four forests generated at each level.

For every candidate forest, we compute $\mathcal{L}_h(F)$ and $J^h_{\mathrm{RGC}}(\mathcal{C}(F))$ exactly as defined in the main paper. For each graph, their ranking agreement is measured by

$$\rho_h = \mathrm{Spearman}\left(\{\mathcal{L}_h(F_k)\}_{k=1}^K, \{J^h_{\mathrm{RGC}}(\mathcal{C}(F_k))\}_{k=1}^K\right).$$

Table 8 reports the mean and standard deviation of ρ_h over the 1,000 graphs.

Table 8: Ranking agreement between the linear forest objective and non-singleton all-pairs RGC on IMDB-BINARY at $r = 0.2$ (mean $\pm$ standard deviation over 1,000 graphs).

h	Spearman$(\mathcal{L}_h, J^h_{\mathrm{RGC}})$
1	0.868 ± 0.060
2	0.845 ± 0.115

The strong positive correlations show that $\mathcal{L}_h$ provides an informative ranking signal for the original non-singleton RGC objective over the sampled feasible forests. This diagnostic supports the use of the linear lower bound as a tractable surrogate, without implying exact optimization of J^h_{RGC}.

B.5 Molecular Regression Protocol

We evaluate molecular graph regression on AQSOL and ZINC using fixed train, validation, and test splits from the benchmarking GNN pipeline. AQSOL contains 7,985, 998, and 999 graphs in the three splits, and ZINC contains 10,000, 1,000, and 1,000 graphs. All methods coarsen only molecular topology and retain the original graph-level regression label.

For a molecular graph with adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, atom features $\mathbf{X} \in \{0,1\}^{N\times d_a}$, and hard assignment matrix $\mathbf{Q} \in \{0,1\}^{N\times n}$, we use $n = \min\{N, \max\{5, \lceil 0.3N\rceil\}\}$. Let $\mathbf{s} = \mathbf{Q}^\top \mathbf{1}_N$ and $\bar{\mathbf{Q}} = \mathbf{Q}\,\mathrm{diag}(\mathbf{s})^{-1}$. Coarse-node features and weighted adjacency are

$$\mathbf{X}_c = \bar{\mathbf{Q}}^\top \mathbf{X}, \qquad \mathbf{A}_c = \bar{\mathbf{Q}}^\top \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{Q}.$$

Thus, each coarse node stores the average atom-type distribution of its cluster, and the GCN uses entries of $\mathbf{A}_c$ as edge weights. For graph readout, a coarse node j receives weight $s_j n/N$, which preserves the scale of the original node average.

All methods use the same GCGCN backbone with four graph-convolution layers, hidden and output dimensions of 145, residual connections, batch normalization, mean readout, and no dropout. We train with ℓ_1 loss and Adam at learning rate 10^{-3}, batch size 128, and zero weight decay. The learning rate is halved after ten validation-plateau epochs, and training stops when it falls below 10^{-5}, after 1,000 epochs, or at the 12-hour time limit. Results are reported as mean and standard deviation over seeds 12 through 21.

B.6 Definitions of REE and RE

We follow the spectral and reconstruction criteria used in graph coarsening evaluations (Loukas & Vandergheynst, 2018; Liu et al., 2018) and state the implementation used for all reported values. For adjacency matrices $A \in \mathbb{R}^{N\times N}$ and $A_c \in \mathbb{R}^{n\times n}$, define the normalized Laplacians

$$L = I - D^{-1/2}AD^{-1/2}, \qquad L_c = I - D_c^{-1/2}A_cD_c^{-1/2}.$$

Let $0 \le \lambda_1 \le \cdots \le \lambda_N$ and $0 \le \lambda_{c,1} \le \cdots \le \lambda_{c,n}$ be their ordered eigenvalues. The implementation compares the first n eigenvalues and excludes original eigenvalues below $\epsilon = 10^{-9}$. With

$$\mathcal{I}_\epsilon = \{i \in \{1,\dots,n\} : |\lambda_i| \ge \epsilon\},$$

the Relative Eigenvalue Error is

$$\mathrm{REE} = \frac{1}{|\mathcal{I}_\epsilon|}\sum_{i\in\mathcal{I}_\epsilon} \frac{|\lambda_{c,i} - \lambda_i|}{|\lambda_i|}.$$

If $\mathcal{I}_\epsilon$ is empty, the implementation returns $n^{-1}\sum_{i=1}^n |\lambda_{c,i} - \lambda_i|$.

Table 9: Structural fidelity REE at $r = 0.2$ on graph classification datasets, using the same 10 repeated runs as the main-paper classification accuracy experiment. Lower is better.

Method	MUTAG	PTC	PROTEINS	MSRC	IMDB	Tumblr	Mean
VNGC	1.783±0.000	1.394±0.000	1.375±0.000	1.562±0.000	0.087±0.000	1.117±0.000	1.220
VEGC	1.078±0.000	1.069±0.000	1.038±0.000	0.774±0.000	0.327±0.000	1.115±0.000	0.900
MGC	6.740±0.000	7.171±0.000	1.574±0.000	1.512±0.000	0.168±0.000	1.782±0.000	3.158
SGC	3.953±0.051	4.901±0.031	1.426±0.010	1.081±0.014	0.451±0.003	0.620±0.012	2.072
KGC	2.473±0.040	2.588±0.086	1.778±0.017	1.078±0.023	0.249±0.003	1.097±0.013	1.544
ORGC-d1	1.648±0.027	1.382±0.019	0.983±0.002	1.150±0.001	0.147±0.000	0.432±0.003	0.957
ORGC-d2	1.467±0.018	1.117±0.009	0.899±0.001	1.083±0.000	0.152±0.000	0.463±0.000	0.863

Table 10: Structural fidelity RE at $r = 0.2$ on graph classification datasets, using the same 10 repeated runs as the main-paper classification accuracy experiment. Lower is better.

Method	MUTAG	PTC	PROTEINS	MSRC	IMDB	Tumblr	Mean
VNGC	1.611±0.000	1.590±0.000	1.483±0.000	1.735±0.000	1.197±0.000	5.391±0.000	2.168
VEGC	1.700±0.000	1.643±0.000	1.601±0.000	2.009±0.000	2.146±0.000	1.879±0.000	1.830
MGC	6.811±0.000	5.673±0.000	2.506±0.000	1.930±0.000	1.724±0.000	8.992±0.000	4.606
SGC	4.248±0.043	4.331±0.024	2.251±0.016	2.285±0.015	1.856±0.007	4.186±0.052	3.193
KGC	1.913±0.013	1.907±0.016	2.137±0.009	2.406±0.043	1.590±0.003	8.075±0.383	3.005
ORGC-d1	1.677±0.005	1.609±0.003	1.398±0.000	1.780±0.001	1.555±0.047	2.006±0.002	1.671
ORGC-d2	1.608±0.004	1.569±0.001	1.392±0.000	1.661±0.000	1.546±0.051	1.818±0.002	1.599

For the hard assignment matrix $Q \in \{0,1\}^{N\times n}$, the lifted normalized Laplacian is

$$L_{\text{lift}} = QL_cQ^\top.$$

The reported Reconstruction Error is

$$\text{RE} = \frac{1}{N}\left\|L - L_{\text{lift}}\right\|_F^2.$$

The factor $1/N$ makes the squared reconstruction discrepancy comparable across graph sizes. Both REE and RE are computed from normalized Laplacians and are minimized.

B.7 Structural Fidelity on Classification Datasets

To complement the graph classification accuracy results in the main paper, Tables 9 and 10 report REE and RE on the same datasets at $r = 0.2$. The results use the same 10 repeated runs as the classification experiment.

ORGC obtains the best compressed-graph aggregate REE and RE, showing that its classification performance is accompanied by strong spectral structural fidelity.

B.8 Runtime Scaling Details

Table 11 reports the wall-clock coarsening time used to produce the runtime-scaling figure in the main paper. Values are mean ± standard deviation in seconds over 5 runs. A dash indicates that the method did not finish that graph size within the time budget.

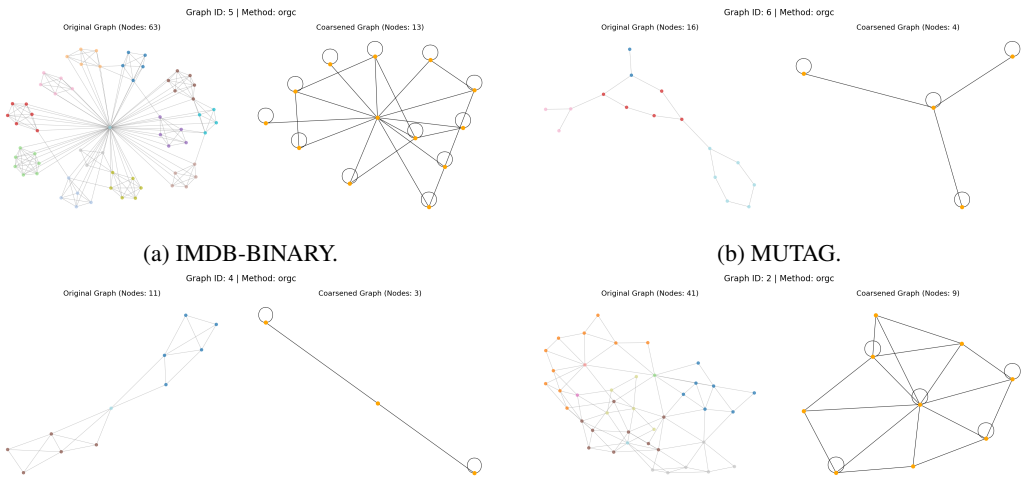
B.9 Qualitative Coarsening Examples

Figure 5 presents representative ORGC outputs on four graph classification datasets at a retained-node ratio of approximately 0.2. Each panel compares the input graph with the induced coarse graph. Node colors in the input graph identify contraction clusters, and self-loops in the coarse graph summarize edges internal to those clusters. The examples cover community-dominated, molecular, protein, and image-region graphs, illustrating how ORGC compresses graphs with distinct structural regimes while retaining their coarse connectivity patterns.

Table 11: Runtime scaling values for the runtime-scaling figure in the main paper. Values are wall-clock coarsening time in seconds, reported as mean ± standard deviation over 5 runs.

Method	$N = 50$	$N = 100$	$N = 200$	$N = 400$
ORGC-d1	0.016±0.002	0.026±0.001	0.060±0.008	0.15±0.002
ORGC-d2	0.054±0.004	0.21±0.008	0.80±0.024	3.18±0.13
VNGC	0.059±0.007	0.10±0.003	0.20±0.009	0.44±0.010
VEGC	0.035±0.001	0.054±0.000	0.100±0.002	0.24±0.007
KGC	0.090±0.011	0.17±0.001	0.42±0.003	1.66±0.028
MGC	0.30±0.004	2.36±0.037	21.4±0.79	194±1.77
SGC	1.09±0.005	4.21±0.060	18.4±0.13	150±0.93

Method	$N = 800$	$N = 1200$	$N = 2000$	$N = 3000$	$N = 5000$
ORGC-d1	0.40±0.011	0.78±0.028	1.81±0.042	3.60±0.17	9.22±0.17
ORGC-d2	12.3±0.79	27.9±3.83	71.5±5.76	142±14.2	348±22.9
VNGC	1.07±0.035	1.94±0.069	4.53±0.15	9.13±0.11	24.3±0.53
VEGC	0.74±0.035	1.42±0.066	3.60±0.13	8.10±0.14	21.9±0.45
KGC	12.9±0.17	55.1±1.33	399±2.72	–	–
MGC	–	–	–	–	–
SGC	–	–	–	–	–



(a) IMDB-BINARY.

(b) MUTAG.

(c) PROTEINS.

(d) MSRC.

Figure 5: Representative ORGC coarsening outputs at $r \approx 0.2$. Each panel compares the original graph on the left with the induced coarse graph on the right.