From Nodes to Pixels: Topological and Structural Two-View Graph Imaging

Md Joshem Uddin*
Department of Mathematics
University of Georgia
Athens, GA 30602, USA

Soham Changani*
Dept of Math. Sciences
University of Texas at Dallas
Richardson, TX 75080, USA

Sai Karthik Navuluru
Dept of Elec. & Comp. Eng.
University of Texas at Dallas
Richardson, TX 75080, USA

Cuneyt Gurcan Akcora
AI Institute
University of Central Florida
Orlando, FL 32816, USA

Baris Coskunuzer
Dept of Math. Sciences
University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Graph learning still lacks a broadly reusable input interface comparable to patches in vision or tokens in language. Unlike images or text, graphs are irregular, vary in size and density, and are invariant under many equivalent node labelings, making it difficult to standardize inputs across tasks and architectures. We introduce G2IMAGE, a fixed-budget graph imaging framework that converts each graph into two complementary image-like views. TOPOGRID provides a stable topological view by encoding an intrinsic bifiltration as a compact multipersistence surface grid, while GRAPHGRID provides a higher-bandwidth structural view by aggregating edge densities between groups defined by intrinsic node scores. Both views are permutation-invariant and come with resolution-controlled stability guarantees. Because the resulting representations are fixed-size tensors, they can be processed by lightweight 2D encoders and aligned through a supervised cross-view contrastive objective. Across graph classification and molecular prediction benchmarks, G2IMAGE achieves strong average performance among the compared methods, improves over single-view and standard fusion variants, and extends to attribute-rich molecular graphs without changing the overall architecture. These results suggest that fixed-budget graph images offer a practical and reusable interface for graph representation learning.

1 Introduction

Graphs represent molecules, biological networks, and social systems, but graph learning still lacks a reusable input interface analogous to patches in vision [DBK+20] or tokens in language [SHB16]. Images and text can be converted into fixed-budget representations that decouple backbone design from raw input size [HCX+22]. Graphs are irregular, vary in size and density, and admit many equivalent node labelings [WPC+20, ZCH+20]. Consequently, graph models often operate directly on the combinatorial object, tying model design to neighborhood aggregation, pooling, structural encodings, and dataset-specific preprocessing [EPBM20, DJL+23].

This creates a tension in graph classification. Message-passing GNNs handle variable-size graphs natively, but their bias is tied to local aggregation and can be sensitive to graph size, density, and pooling [XHLJ18, HYL17]. Graph transformers improve expressivity, but often require additional

*Equal contribution.

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

structural encodings or costly attention over graph elements [Y+21, RGD+22]. Handcrafted summaries are stable and fixed-size, but may discard too much relational information. We seek an intermediate interface: a fixed-budget, permutation-invariant graph representation that preserves mesoscopic structure, exposes complementary signals, and can be processed by standard neural backbones without designing a new graph operator.

We propose G2IMAGE, a two-view graph imaging interface that maps each graph to a pair of compact tensors. The first view, TOPOGRID, constructs an intrinsic bifiltration from permutation-equivariant node functions and encodes it as a multipersistence surface tensor [BL22, LCB23] with Betti-0, Betti-1, vertex-count, and edge-count channels. This provides a stable topological view of connectivity and cycle structure at a controlled resolution. The second view, GRAPHGRID, groups nodes by quantized intrinsic score signatures, aggregates edge densities between groups, and pools the resulting density matrix to a fixed $k \times k$ grid. This produces a permutation-invariant structural image that captures relational patterns not fully represented by topological surfaces alone.

Because both views are fixed-size tensors, G2IMAGE uses lightweight 2D encoders rather than task-specific graph operators. The views are not simply concatenated: we align them with a supervised cross-view contrastive objective [KTW+20], treating the topological and structural images of the same graph as positive pairs and same-label graphs as additional positives. This encourages agreement between a stable topological compression and a higher-bandwidth structural map while preserving task-discriminative information through the classification loss.

The framework is deliberately non-injective: at fixed resolution, distinct graphs may map to the same image. This reflects an explicit tradeoff: G2IMAGE gives up full reconstruction for a fixed-budget, permutation-invariant, resolution-controlled interface. We formalize this through invariance and stability results: TOPOGRID is locally constant away from threshold crossings, while GRAPHGRID is stable to edge perturbations under a fixed bin partition, with an explicit bin-swap penalty.

We evaluate G2IMAGE on graph classification benchmarks and molecular property prediction tasks [MKB+20, HFZ+20]. Across seven TU benchmarks, G2IMAGE achieves the best overall AvgRank among 16 compared methods, while both TOPOGRID and GRAPHGRID remain competitive as standalone representations. On seven OGB molecular datasets, G2IMAGE-CHEM, a feature-aware extension of G2IMAGE, achieves the best overall performance, winning outright on five of seven tasks, and confirms that the fixed-budget interface generalizes to attribute-rich molecular graphs without architectural redesign. Ablations show that multipersistence improves over single-filtration summaries, the structural grid has predictable resolution sensitivity, and supervised cross-view alignment improves over standard fusion.

Contributions.

- **Fixed-budget graph imaging.** We introduce G2IMAGE, a permutation-invariant interface that maps variable-size graphs to compact image-like tensors compatible with standard 2D encoders.
- **Two complementary graph views.** TOPOGRID encodes multipersistence surfaces from an intrinsic bifiltration, while GRAPHGRID encodes block-pooled edge densities between quantized intrinsic node groups.
- **Resolution-controlled stability.** We show that both views are invariant to node relabeling, that TOPOGRID is locally constant away from threshold crossings, and that GRAPHGRID admits an edge-perturbation bound with a bin-swap term.
- **Empirical validation.** A supervised cross-view contrastive objective improves over standalone views and standard fusion across graph classification and molecular prediction tasks.

2 Background

2.1 Related Work

Graph Classification. Graph classification is dominated by message-passing GNNs coupled with permutation-invariant pooling, which aggregate node representations into a graph-level embedding [KW17, HYL17, XHLJ18, VCC+18]. Graph transformers and hybrids such as GPS incorporate global attention to better capture long-range interactions beyond local neighborhoods [Y+21, RGD+22]. Graph classification uses standard benchmarks where careful evaluation protocols

are important since small datasets and confounders such as graph size and density can induce shortcut learning and unstable rankings [DJL+23, EPBM20, LCS+24].

Graph-to-Grid Methods. Graph-to-grid methods convert graphs into fixed-size tensors by imposing an ordering and then applying CNN-style processing. Patchy-SAN constructs normalized receptive fields via graph labeling and neighborhood ordering [NAK16, SSVL+11], while SortPooling orders nodes globally before applying convolutional models [ZCNC18]. Hierarchical pooling and coarsening also produce fixed-budget summaries by aggregating nodes into clusters [YYM+18, CVJ+18, WPC+20]. A common limitation is sensitivity to tie-breaking and near-symmetries, where small perturbations can change orderings, and aggressive pooling can discard relational detail. Differing from earlier work, our GRAPHGRID is permutation-invariant by construction: we quantize permutation-equivariant node scores, aggregate edges by bin groups, and pool the resulting group-level density matrix into a fixed $k \times k$ map, eliminating sensitivity to within-group ordering and tie-breaking.

Topology and multipersistence for graph learning. TDA provides structural descriptors via persistent homology that complement message passing by capturing multi-scale connectivity patterns [DW22, CA24]. Multiparameter persistence combines multiple filtrations and is typically summarized through stable, grid-based vectorizations rather than canonical barcodes [BL22, LCB23]. Recent work develops practical multipersistence representations for learning (EMP) [CSDA+24], graph-integrated encoders (Graphcode) [RK24], low-dimensional representations (TopER) [TTAC25] and stability and differentiable constructions [LSC+23, SSL+24]. Existing approaches typically treat topology as an auxiliary feature appended to a GNN; we instead use multipersistence surfaces as a fixed-budget image-like view that is directly processed by 2D encoders.

Multi-view learning and contrastive alignment. Multi-view learning aligns different views of the same object to learn invariant representations. In vision, contrastive and supervised-contrastive objectives (e.g., SimCLR [CKNH20], SupCon [KTW+20]) bring positive pairs together and improve class structure [vLV18], and dual-encoder alignment has been popularized at scale [R+21, JYX+21]. In graphs, analogous objectives rely on multiple augmentations, leading to strong self-supervised baselines such as GraphCL, GRACE, and BGRL [YCS+20, ZXY+20, TTA+22]. Distillation-style alternatives can be more stable in smaller-batch regimes [HVD15, GSA+20]. Our setting differs from augmentation-based graph contrastive methods in a fundamental way: the two views are not stochastic perturbations of the same graph representation, but structurally heterogeneous fixed-budget images encoding complementary information. Aligning a stable topological compression with a higher-bandwidth structural map differs from aligning two augmented adjacency matrices.

2.2 Persistent Homology and Multiparameter Persistence

Persistent homology (PH) summarizes how topological features such as connected components and cycles appear across a filtration [DW22]. In graph learning, PH provides structural descriptors that can complement message passing by capturing connectivity and cycle patterns beyond local aggregation [CA24]. A standard PH pipeline builds a nested sequence of subgraphs or clique complexes from a node or edge filtration function, computes persistence diagrams in each homology dimension, and vectorizes them for learning using summaries such as persistence images, landscapes, silhouettes, or Betti curves [AAJ+22]. We review single-parameter PH in Appendix B.

Our method uses a two-parameter version of this construction. Given two intrinsic node functions $f, g : \mathcal{V} \to \mathbb{R}$, such as heat kernel signature and degree centrality, we define a sublevel bifiltration $\mathcal{V}_{(\alpha,\beta)} = \{v \in \mathcal{V} : \ f(v) \le \alpha, \ g(v) \le \beta\}, \quad \mathcal{G}_{(\alpha,\beta)} = \mathcal{G}[\mathcal{V}_{(\alpha,\beta)}], \quad$ and pass to clique complexes $\widehat{\mathcal{G}}_{(\alpha,\beta)}$. Discretizing thresholds $\{\alpha_i\}_{i=1}^{M}$ and $\{\beta_j\}_{j=1}^{N}$ gives a grid-indexed bifiltration $\{\widehat{\mathcal{G}}_{ij}\}$ with $\widehat{\mathcal{G}}_{ij} \subseteq \widehat{\mathcal{G}}_{i'j'}$ whenever $i \le i'$ and $j \le j'$. For each homology dimension k, this defines a bi-graded persistence module over the grid poset. Unlike the single-parameter case, general multiparameter persistence does not admit a complete barcode-style decomposition [BL22]. Practical graph-learning pipelines therefore rely on stable finite summaries, such as surfaces, rank-based summaries, and learned vectorizations [LCB23, RK24]. TOPOGRID follows this fixed-summary approach. Instead of encoding a complete multiparameter invariant, it records compact Betti-0, Betti-1, vertex-count, and edge-count surfaces on the discretized bifiltration, yielding a fixed-budget, permutation-invariant topological image with controlled resolution. Additional MP background and design motivation are given in Appendix C.

3 G2Image: Two-View Graph Imaging

Goal and rationale. We seek fixed-size graph representations that capture meaningful mesoscopic structure while remaining stable under node relabeling and small perturbations. By *intrinsic*, we mean permutation-equivariant functions of the input graph that depend only on its structure and not on node identities or external ordering. We propose a two-view imaging interface that maps each graph to two compact, image-like tensors that can be processed by standard 2D encoders. The first view, TOPOGRID, provides a stable, compressive topological view via multipersistence surfaces from an intrinsic bifiltration. The second view, GRAPHGRID, provides a higher-bandwidth structural view by grouping nodes according to intrinsic score signatures, aggregating edge densities between groups, and pooling the resulting group-level matrix to a fixed resolution. Because both are fixed-size tensors, we can couple them with lightweight 2D encoders and an explicit two-view alignment objective (Section 3.3). Expressivity limits and failure cases of both views are characterized in Appendix G.

3.1 TOPOGRID View

Overview. TOPOGRID maps each graph to a compact multipersistence surface tensor from an intrinsic bifiltration. It is a stable, compressive view that complements the higher-bandwidth GRAPHGRID in our two-view framework (Section 3.3). We call GRAPHGRID higher-bandwidth because it retains a $k \times k$ map of inter-group edge densities, whereas TOPOGRID captures structure via low-dimensional counts on an $M \times N$ grid.

Multipersistence surfaces from intrinsic bifiltrations. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple graph. Following Section 2.2, choose two intrinsic node functions $f, g : \mathcal{V} \to \mathbb{R}$ capturing complementary structure. In this paper we use $f = f_{\mathrm{hks}}$ (heat kernel signature at a fixed diffusion time) and $g = f_{\mathrm{degc}}$ (normalized degree centrality). Both scores are permutation-equivariant and depend only on graph structure; we normalize them within each graph prior to quantile binning. Given threshold grids $\{\alpha_i\}_{i=1}^{M}$ and $\{\beta_j\}_{j=1}^{N}$, define $\mathcal{V}_{ij} = \{v \in \mathcal{V} : f(v) \le \alpha_i,\ g(v) \le \beta_j\}$, $\mathcal{G}_{ij} = \mathcal{G}[\mathcal{V}_{ij}]$, and let $\widehat{\mathcal{G}}_{ij} = \mathrm{Cl}(\mathcal{G}_{ij})$ be the clique complex of $\mathcal{G}_{ij}$. For efficiency, we truncate $\widehat{\mathcal{G}}_{ij}$ to dimension 2, and compute only $k \in \{0, 1\}$.

We summarize the bifiltration with four surface channels on the (i, j) grid: $S_{ij}^{(0)} = \beta_0\left(\widehat{\mathcal{G}}_{ij}\right)$, $S_{ij}^{(1)} = \beta_1\left(\widehat{\mathcal{G}}_{ij}\right)$, $S_{ij}^{(V)} = |\mathcal{V}_{ij}|$, $S_{ij}^{(E)} = |\mathcal{E}_{ij}|$, where $\mathcal{E}_{ij} = \{(u, v) \in \mathcal{E} : u, v \in \mathcal{V}_{ij}\}$ is the induced edge set. Stacking yields a fixed-size topological image per graph, $\mathcal{X}(\mathcal{G}) = \left[S^{(0)}, S^{(1)}, S^{(V)}, S^{(E)}\right] \in \mathbb{R}^{4 \times M \times N}$.

Leakage-safe grids and normalization. We construct the threshold grids using empirical quantiles computed from the training split, and normalize each surface channel using training-split statistics (mean and variance) that are cached and reused at evaluation time. Quantile grids also provide a consistent semantics: the cell (i, j) corresponds to a percentile region of (f, g) rather than an absolute scale, reducing sensitivity to dataset-specific score ranges. This choice also ensures reproducible preprocessing and prevents inadvertent use of evaluation statistics.

TOPOGRID encoder. Given $\mathcal{X}(\mathcal{G})$, we apply a small image encoder (CNN) to obtain an embedding $z_{\mathrm{top}}(\mathcal{G})$. This encoder is not intended to be architecturally novel; the contribution is the fixed-budget topological imaging interface that yields predictable compute and integrates naturally with standard 2D representation learners.

3.2 GRAPHGRID View

Overview. We construct a fixed-size GRAPHGRID that summarizes a graph via deterministic, structure-driven binning of intrinsic node scores, followed by bin-aggregation and block pooling of edge densities into a $k \times k$ structural map. GRAPHGRID is designed to complement TOPOGRID: TOPOGRID provides a stable, compressive topological view, while GRAPHGRID provides a higher-bandwidth structural view. Because both are fixed-size tensors, we can couple them with standard two-view learning objectives using lightweight 2D encoders. While graph-to-grid constructions have appeared in prior work, our use pairs topology-guided imaging with an explicit two-view objective and resolution-controlled stability. This block-pooled representation should not be interpreted as a best low-rank approximation. Rather, it is a fixed-resolution summary of edge densities between

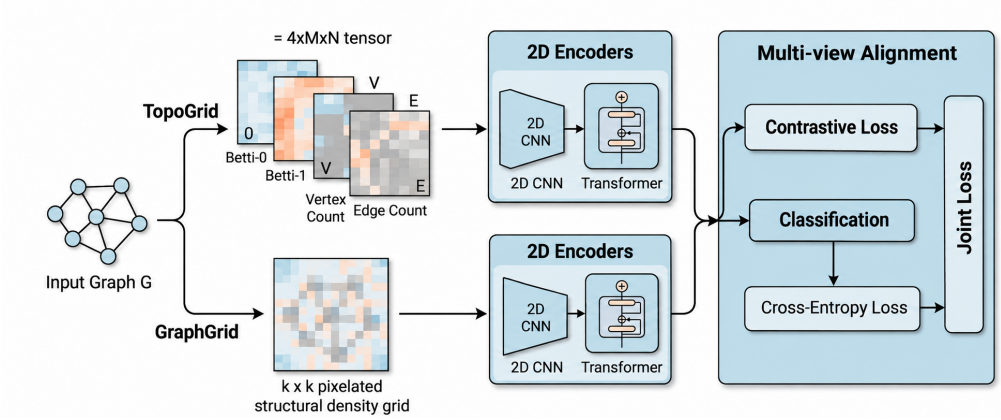

Figure 1: **G2IMAGE overview.** From an input graph $\mathcal{G}$, TOPOGRID builds a $4 \times M \times N$ multipersistence surface tensor and GRAPHGRID builds a $k \times k$ structural density grid. Two lightweight 2D encoders (2D CNN extracts local features, which are flattened into tokens and processed by a Transformer encoder) produce embeddings that are aligned with a supervised cross-view contrastive loss and trained with cross-entropy for graph classification.

structurally similar node groups. In Remark A.3, we discuss when coarse resolutions preserve useful structural signal.

Stable Graph Imaging: from Graphs to Fixed-Size Structural Maps Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$. Our goal is not to compute a canonical form or solve graph isomorphism. Instead, we define an intrinsic and deterministic map that converts $\mathcal{G}$ into a fixed-size structural summary.

Intrinsic node scoring and bin-aligned grouping. For each vertex $v \in \mathcal{V}$, we compute an intrinsic signature $\Phi_{\mathcal{G}}(v) = \big(s_{\mathrm{diff}}(v),\ s_{\mathrm{HKS}}(v),\ \mathrm{kcore}(v),\ \deg(v)\big) \in \mathbb{R}^4$, where all components are permutation-equivariant functions of $\mathcal{G}$, capturing complementary diffusion, spectral, mesoscopic, and local structural properties. Here s_{diff} denotes a diffusion-based centrality score. We normalize each coordinate within-graph and discretize it into B quantile bins, yielding $\widetilde{\Phi}_{\mathcal{G}}(v) \in \{1, \dots, B\}^4$ and bin groups $\mathcal{C}_\tau(\mathcal{G}) := \{v \in \mathcal{V} : \widetilde{\Phi}_{\mathcal{G}}(v) = \tau\}$. Let $\tau_1 \prec \cdots \prec \tau_m$ denote the nonempty signatures in lexicographic order and write $\mathcal{C}_p := \mathcal{C}_{\tau_p}$. Because the construction below aggregates within $\mathcal{C}_p$, it is invariant to permutations *within* each bin group and, since Φ_G is permutation-equivariant, invariant to node relabeling. We use this set as a lightweight mix of diffusion, mesoscopic, and local centrality cues; other permutation-equivariant scalar scores can be substituted without changing the pipeline.

Binning and normalization. Binning uses within-graph quantiles (fixed B across a dataset) and requires no split-specific statistics. We normalize each GRAPHGRID channel using mean and variance computed on the training split and cache these statistics for evaluation.

Bin-aggregated and block-pooled adjacency images. We form a bin-aggregated density matrix $D(\mathcal{G}) \in \mathbb{R}^{m \times m}$: $\quad D(\mathcal{G})_{pq} = \dfrac{1}{|\mathcal{C}_p||\mathcal{C}_q|} \sum_{u \in \mathcal{C}_p} \sum_{v \in \mathcal{C}_q} A_{uv}$.

To obtain a fixed $k \times k$ image, we apply uniform block pooling by partitioning $\{1, \dots, m\}$ into k consecutive blocks $\mathcal{B}_1, \dots, \mathcal{B}_k$ and setting $\quad I_k(\mathcal{G})_{ij} = \dfrac{1}{|\mathcal{B}_i||\mathcal{B}_j|} \sum_{p \in \mathcal{B}_i} \sum_{q \in \mathcal{B}_j} D(\mathcal{G})_{pq}$.

In practice $m \ll B^4$, and $D(\mathcal{G})$ can be computed efficiently by accumulating edge counts between bin groups in a single pass over E.

3.3 G2Image: Multi-View Alignment and Training

Encoders. Let $\mathcal{X}_{\mathrm{top}}(\mathcal{G})$ denote the TOPOGRID tensor from Section 3.1 and let $\mathcal{X}_{\mathrm{img}}(\mathcal{G})$ denote the GRAPHGRID tensor (single I_k or a multi-resolution stack). We use lightweight 2D encoders to obtain embeddings $\quad z_{\mathrm{top}}(\mathcal{G}) = f_{\mathrm{top}}(\mathcal{X}_{\mathrm{top}}(\mathcal{G})), \quad z_{\mathrm{img}}(\mathcal{G}) = f_{\mathrm{img}}(\mathcal{X}_{\mathrm{img}}(\mathcal{G}))$, with optional projection heads and ℓ_2 normalization.

Multi-view training objective. We train the model to (i) align the two views of the same graph and (ii) learn class-discriminative representations for graph classification. We use a supervised contrastive objective with cross-view positives [KTW+20]: for each graph $\mathcal{G}$, the pair $(\mathcal{X}_{\mathrm{top}}(\mathcal{G}), \mathcal{X}_{\mathrm{img}}(\mathcal{G}))$ is a

positive instance pair, and all same-label pairs in the minibatch are treated as additional positives. Different-label graphs act as negatives. We combine this alignment with standard supervised learning by optimizing $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{align}}$, where $\mathcal{L}_{\mathrm{CE}}$ is cross-entropy on labels (using either view-specific heads or a fused head) and $\mathcal{L}_{\mathrm{align}}$ is the cross-view supervised contrastive loss.

3.4 Invariance and Resolution-Controlled Stability

Invariance and resolution-controlled stability. G2IMAGE is not designed to be an injective graph representation. Instead, it constructs fixed-budget summaries whose behavior is controlled by the chosen discretization. We formalize three properties: both views are invariant to node relabeling; TOPOGRID is locally constant away from threshold-crossing events; and GRAPHGRID is Lipschitz in edge perturbations under a fixed bin partition, with an explicit penalty for bin swaps. Neither view is injective at fixed resolution: explicit non-isomorphic collision constructions, threshold-crossing bounds, and block-pooling error analysis are deferred to Appendices A and G.

Proposition 3.1 (Permutation invariance)**.** *Let π be any permutation of the vertices of $\mathcal{G}$. If the intrinsic score functions used by* TOPOGRID *and* GRAPHGRID *are permutation-equivariant, then $\mathcal{X}_{\mathrm{top}}(\pi\mathcal{G}) = \mathcal{X}_{\mathrm{top}}(\mathcal{G})$ and $\mathcal{X}_{\mathrm{img}}(\pi\mathcal{G}) = \mathcal{X}_{\mathrm{img}}(\mathcal{G})$.*

Proposition 3.2 (Local constancy of TOPOGRID)**.** *Fix $\mathcal{G}$ and grids $\{\alpha_i\}, \{\beta_j\}$ with margins $m_f = \min_{v,i} |f(v) - \alpha_i|$ and $m_g = \min_{v,j} |g(v) - \beta_j|$. If $\|f - \tilde{f}\|_\infty < m_f$ and $\|g - \tilde{g}\|_\infty < m_g$, then $\mathcal{X}_{\mathrm{top}}(\mathcal{G}; f, g) = \mathcal{X}_{\mathrm{top}}(\mathcal{G}; \tilde{f}, \tilde{g})$.*

Lemma 3.3 (Bin stability)**.** *Let $\gamma = \min_{v,\ell,b} |\Phi_{\mathcal{G}}(v)_\ell - q_{\ell,b}|$ where $q_{\ell,b}$ is the b-th bin boundary for coordinate ℓ. If $\max_v \|\Phi_{\mathcal{G}}(v) - \Phi_{\mathcal{G}'}(v)\|_\infty < \gamma$, then $\mathrm{BS}(\mathcal{G}, \mathcal{G}') = 0$.*

Theorem 3.4 (Stability of GRAPHGRID)**.** *Let $\mathcal{G}, \mathcal{G}'$ share a vertex set with adjacencies $A, A' \in [0, W]^{n \times n}$. Let $b_{\min} := \min_a |\mathcal{B}_a|$ be the minimum nonempty pooling-block size, $c_{\min} := \min_p |\mathcal{C}_p|$ the minimum nonempty bin-group size under the reference partition of $\mathcal{G}$, and $\mathrm{BS}(\mathcal{G}, \mathcal{G}') := \frac{1}{n}|\{v : \widetilde{\Phi}_{\mathcal{G}}(v) \neq \widetilde{\Phi}_{\mathcal{G}'}(v)\}|$. Then, for an absolute constant C,*

$$\|I_k(\mathcal{G}) - I_k(\mathcal{G}')\|_1 \;\le\; \frac{\|A - A'\|_1}{b_{\min}^2\, c_{\min}^2} \;+\; \min\Big(2Wk^2,\; CWk^2\, n\, \mathrm{BS}(\mathcal{G}, \mathcal{G}')\Big).$$

In particular, if no vertex changes bin then $\mathrm{BS}(\mathcal{G}, \mathcal{G}') = 0$ and the bound reduces to $\|I_k(\mathcal{G}) - I_k(\mathcal{G}')\|_1 \le (b_{\min}\, c_{\min})^{-2} \|A - A'\|_1$.

These results make the tradeoff explicit: G2IMAGE sacrifices full graph reconstruction in exchange for fixed-budget, resolution-controlled summaries that are permutation-invariant, locally stable, and compatible with standard 2D encoders.

4 Experiments

4.1 Setup

Datasets. We evaluate G2IMAGE on two complementary benchmark suites. For TU-style graph classification [MKB+20], we use seven datasets spanning three domains: MUTAG, BZR, PTC-MR, and COX2 (molecular graphs), PROTEINS (bioinformatics), and IMDB-BINARY and IMDB-MULTI (social networks). These datasets vary widely in graph size and density and cover both binary and multi-class settings. To assess generalization beyond small TU benchmarks, we additionally evaluate on seven molecular property prediction datasets from OGB [HFZ+20]: ogbg-molbace, ogbg-molbbbp, ogbg-molclintox, ogbg-molsider, ogbg-moltoxcast, ogbg-moltox21, and ogbg-molhiv. These datasets are substantially larger, include class imbalance, and span single-task and multi-task prediction settings (e.g., 617 tasks for ogbg-moltoxcast, 27 for ogbg-molsider), providing a more demanding test of the fixed-budget imaging interface. Dataset statistics are summarized in Table 1.

Table 1: Graph classification datasets.

Datasets	#Graphs	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Classes
TU Benchmarks				
MUTAG	188	17.93	19.79	2
BZR	405	35.75	38.36	2
PTC-MR	344	14.29	14.69	2
COX2	467	41.22	43.45	2
PROTEINS	1113	39.06	72.82	2
IMDB-B	1000	19.77	96.53	2
IMDB-M	1500	13.00	65.94	3
OGB Molecular Benchmarks				
SIDER	1427	33.64	35.36	27
ClinTox	1478	30.77	32.54	2
BACE	1513	34.08	36.89	2
BBBP	2039	24.06	25.95	2
ToxCast	8576	18.78	19.26	617
Tox21	7831	18.60	36.60	12
HIV	41127	25.53	27.48	2

Table 3: **G2Image vs. SOTA.** Graph classification accuracy comparing G2IMAGE to state-of-the-art baselines across seven benchmarks. We also report **AvgRank** (lower is better), computed by ranking methods per dataset (excluding missing entries) and averaging ranks across datasets. Results are highlighted as **best**, **second-best**, and **third-best**.

Method	MUTAG	BZR	PTC-MR	COX2	PROTEINS	IMDB-B	IMDB-M	AvgRank↓
GCN [KW17]	87.89±5.29	86.83±4.25	62.57±4.13	84.26±2.89	74.11±1.79	60.60±3.44	50.47±4.38	10.00
GAT [VCC+18]	87.37±4.21	83.66±4.09	63.14±5.92	**85.11**±2.13	74.38±1.33	61.70±1.49	49.93±3.09	9.71
GIN [XHLJ18]	83.68±8.95	82.68±3.53	54.86±7.86	80.85±5.47	72.05±3.57	61.30±1.85	47.93±2.99	13.93
GSAGE [HYL17]	83.60±9.60	77.53±3.73	60.10±4.70	79.01±2.42	73.00±4.50	68.80±4.50	47.60±3.50	13.86
DiffPool [YYM+18]	87.50±3.35	51.83±2.17	45.75±5.25	59.67±3.32	70.60±2.74	58.71±1.48	42.08±1.93	16.29
DropGIN [PMFW21]	85.79±9.72	77.80±2.55	55.14±8.86	78.72±0.95	55.14±8.86	55.00±3.71	37.33±3.16	16.36
GPRGNN [CPLM21]	91.05±2.41	84.88±2.39	62.57±5.49	78.72±0.00	68.30±4.76	61.90±2.84	49.80±2.78	11.57
TFGW [VCFC+22]	**93.68**±2.11	84.39±2.93	61.14±5.74	83.19±3.08	74.29±2.14	62.20±2.18	50.67±2.92	8.57
WWLS [FHSK23]	88.30±1.23	88.02±0.61	67.32±1.09	81.58±0.91	75.35±0.74	75.08±0.31	51.61±0.62	6.57
MP-HSM [LSC+23]	85.60±5.30	–	–	77.10±3.00	74.60±2.10	74.80±2.50	47.90±3.20	12.00
EPIC [HLAK24]	82.44±0.70	88.78±2.30	–	**85.53**±1.60	69.06±1.00	71.70±1.00	47.93±1.30	10.75
TopoGCL [C+24]	90.09±0.93	87.17±0.83	63.43±1.13	81.45±0.55	**77.30**±0.89	74.67±0.32	**52.81**±0.31	5.93
PGOT [QTLL24]	**95.79**±3.16	84.15±4.26	63.43±7.65	**85.74**±3.44	71.52±2.27	60.60±4.94	43.07±5.54	9.57
GPSE [CLLG+24]	87.19±8.66	80.49±4.18	–	78.37±2.62	72.15±3.66	69.30±3.61	47.40±5.40	13.83
TopER [TTAC25]	90.99±6.64	90.13±4.14	–	82.01±4.59	74.58±3.92	73.20±3.43	50.00±4.02	7.50
MultiNet [QBC+25]	89.81±1.46	–	62.65±0.88	–	76.40±0.87	**76.49**±0.60	51.93±0.25	5.60
TOPOGRID	92.60±6.73	**92.82**±3.61	**68.58**±4.64	82.91±5.81	**77.72**±3.66	**76.60**±2.58	51.00±2.86	**3.71**
GRAPHGRID	93.63±4.64	**90.37**±2.82	**70.93**±5.53	84.83±5.69	76.19±2.24	75.20±2.99	**52.07**±4.29	**3.57**
G2IMAGE	**94.18**±5.98	**91.84**±2.98	**72.38**±4.17	84.62±5.15	**76.46**±3.15	**77.30**±1.55	**52.73**±3.57	**2.29**

Representations. For each graph $\mathcal{G}$, G2IMAGE builds two complementary fixed-budget grids. TOPOGRID is a $4\times10\times10$ multipersistence tensor from the (HKS, degree-centrality) bifiltration, with channels $(\beta_0, \beta_1, |\mathcal{V}|, |\mathcal{E}|)$. GRAPHGRID computes permutation-equivariant node scores (PageRank, HKS, k-core, degree), quantizes them into within-graph bins ($B = 10$), aggregates edge densities between bin groups, and average-pools the resulting matrix to a 10×10 grid. For OGB datasets, G2IMAGE-CHEM augments the framework by building an additional image from node features through binning and pooling while leaving the imaging pipeline and encoders unchanged, highlighting the flexibility of our framework to incorporate attribute information when available.

Implementations. We use two lightweight 2D encoders, one per view, producing embeddings $z_{\text{top}}(\mathcal{G})$ and $z_{\text{img}}(\mathcal{G})$. Training minimizes a joint objective $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{CON}}$ with $\lambda = \{0.1, 0.2, 0.5\}$, where $\mathcal{L}_{\text{CE}}$ is cross-entropy on graph labels and $\mathcal{L}_{\text{CON}}$ is an NT-Xent objective over cross-view positive pairs. We optimize with AdamW (weight decay 10^{-4}) and gradient clipping (norm 1.0). Resolution ablations are reported in Section 4.3. Further implementation details are given in Appendix D. All experiments were conducted on an NVIDIA H100 NVL GPU with 768 GB RAM. Code is available at `https://github.com/sohamchangani/G2Image`.

Computational complexity and runtime. G2IMAGE performs a one-time, cacheable graph-to-tensor conversion; all subsequent training operates on fixed-size tensors. Given fixed (M, N, k), the forward and backward cost of the 2D encoders and alignment loss is independent of $|V|$ and $|E|$ and depends only on the backbone and batch size.

Table 2: Runtime Analysis (in seconds)

Dataset	TopoGrid	GraphGrid	SupCon	Total
MUTAG	130.8	0.2	80.5	212.7
PROTEINS	538.1	3.0	594.1	1154.4
IMDB-B	1066.3	1.3	287.5	1362.8

For TOPOGRID, each of the MN cells requires $O(|V_{ij}|+|E_{ij}|)$ time for β_0 via connected components and sparse matrix rank computation for β_1 [EP14]. For GRAPHGRID, within-graph quantile binning costs $O(|V| \log |V|)$ per score, edge accumulation costs $O(|E|)$, and block pooling adds negligible $O(k^2)$ overhead. End-to-end training time is thus linear in the number of graphs per epoch with constants fixed by the chosen tensor resolutions. Preprocessing and training wall-clock times for representative datasets of varying size and density, are reported in Table 2.

Evaluation protocols. For TU benchmarks, we use 10-fold cross-validation with fixed random seeds, select checkpoints by validation accuracy within each fold, and report mean±std test accuracy over all folds. For OGB benchmarks, we use the standard random train/validation/test splits with the provided evaluator and report mean±std ROC-AUC over 5 runs.

4.2 Results

G2IMAGE vs. SOTA. Table 3 shows that G2IMAGE achieves the best overall performance across seven TU benchmarks, with the lowest AvgRank (2.29), indicating consistent gains rather than

Table 4: **OGBG-MOL Benchmarks.** Performance comparison across multiple OGB molecular graph classification datasets. G2IMAGE* is chemistry-aware instantiation incorporating OGB atom and bond features; see Appendix D. AvgRank is computed per column (lower is better). AvgGap is the mean difference to the best result per dataset (lower is better).

Method # Graphs	SIDER 1427	ClinTox 1478	BACE 1513	BBBP 2039	TOXCAST 8575	TOX21 7831	HIV 41127	AvgRank↓	AvgGap↓
GCN	58.20±2.59	75.56±3.12	71.90±4.04	65.60±1.43	56.78±3.04	74.68±1.09	72.90±1.12	5.57	5.01
GIN	57.73±3.46	76.10±3.00	**72.43±4.36**	**65.76±1.91**	60.33±2.00	**75.63±1.94**	**76.75±1.36**	4.14	3.71
GSAGE	**58.95±1.97**	**81.90±5.47**	72.21±4.04	64.00±2.16	**66.64±1.92**	75.17±1.94	73.05±1.47	**3.86**	**2.68**
GAT	**59.67±2.34**	**84.07±4.52**	**72.89±4.00**	**66.29±3.06**	**64.71±1.49**	**76.03±1.72**	**75.41±3.55**	**2.00**	**1.66**
GPS	59.55±1.53	76.14±4.68	67.47±3.86	65.06±1.52	58.66±0.69	70.97±0.43	70.44±1.08	5.71	6.34
SGFormer	58.32±0.80	78.07±1.26	66.24±4.78	**66.08±1.31**	63.13±0.23	72.97±0.45	72.21±1.33	4.86	4.81
G2IMAGE*	**60.13±1.85**	**81.72±1.02**	**76.85±1.22**	65.34±1.72	**69.13±1.03**	**77.32±1.60**	**76.89±1.03**	**1.86**	**0.47**

isolated wins on individual datasets. G2IMAGE attains the top accuracy on PTC-MR and IMDB-B, while remaining among the top two methods on MUTAG, BZR, and IMDB-M. Both standalone views are competitive in their own right: TOPOGRID is best on BZR and PROTEINS, and GRAPHGRID achieves the second-best AvgRank (3.57) across all datasets. These results support the central claim that combining a stable topological summary with a higher-bandwidth structural grid produces a robust fixed-budget interface that generalizes across molecular, bioinformatics, and social graph regimes. We discuss scalability, expressivity, and evaluation scope limitations in Appendix G.

G2IMAGE-CHEM on OGB Molecular Benchmarks. G2IMAGE is modular: domain adaptation requires only substituting the intrinsic scoring functions, with the imaging pipeline unchanged. We instantiate this as G2IMAGE-CHEM by incorporating OGB atom and bond features into the scoring pipeline (Appendix D), and evaluate on seven OGB datasets against GCN, GIN, GraphSAGE, GAT, GPS, and SGFormer under the standard random-split protocol [HFZ+20]. As shown in Table 4, G2IMAGE-CHEM achieves the best AvgRank (1.86) and the lowest AvgGap (0.47) across all methods, winning outright on SIDER, BACE, ToxCast, TOX21, and HIV, and remaining within 2.4 points of the best on ClinTox. The one exception is BBBP, where the fixed-budget representation remains competitive but falls short of the best result, suggesting that it may miss fine-grained local structural cues critical for blood–brain barrier permeability; expressivity limits are discussed in Appendix G. Overall, these results confirm that the fixed-budget interface generalizes to attribute-rich molecular graphs at OGB scale without any architectural redesign.

TOPOGRID vs. multipersistence baselines. Table 5 compares TOPOGRID against a broad set of multiparameter persistence (MP) vectorizations and learning pipelines, including classical summaries (MP-Kernel [CFK+19], MP-Land [Vip20], MP-Im [CB20]), rank-based representations (GRIL [CMSD23], MP-Hil, MP-Euler [OS21]), and recent learning-oriented MP encoders (MP-SB [LSC+23], EMP [CSDA+24], MPTOP [SSL+24], GraphCode [RK24]). TOPOGRID is consistently among the top methods across all five benchmarks, attaining the best accuracy on COX2, MUTAG (tied), and PROTEINS, with particularly large margins on COX2 and PROTEINS. On social benchmarks, TOPOGRID ranks second on IMDB-B and close to best on IMDB-M. These results suggest that compact multipersistence surface grids preserve the essential MP signal for graph classification while providing a fixed-budget interface that is competitive with richer MP summaries and specialized MP pipelines at substantially lower representational complexity. The fixed-budget design choices motivating this tradeoff are discussed further in Appendix C.

Table 5: **TopoGrid vs. multipersistence baselines for graph classification.** Accuracy results comparing TopoGrid to recent MP models across benchmark datasets. The **best**, **second**, and **third** performances are highlighted.

Dataset	MP-Kernel	MP-Land	MP-Im	GRIL	MP-Hil	MP-Euler	MP-SB	EMP	MPTOP	GraphCode	TopoGrid
COX2	**79.9±1.8**	79.0±3.3	77.9±2.7	79.8±2.9	78.2±1.7	77.3±1.1	78.4±0.7	**79.9±0.8**	NA	78.7±4.9	**82.9±5.8**
IMDB-B	68.2±1.2	71.2±2.0	71.1±2.1	65.2±2.6	72.0±1.9	73.0±4.5	**75.1±3.4**	74.4±0.5	**78.4±5.1**	65.4±2.7	**76.6±2.6**
IMDB-M	46.9±2.6	46.2±2.3	46.7±2.7	NA	49.1±1.6	50.0±0.8	**51.1±1.3**	48.0±0.4	**51.2±4.4**	NA	**51.0±2.9**
MUTAG	86.1±5.2	84.0±6.8	85.6±7.3	87.8±4.2	87.2±5.8	87.2±4.3	**89.9±4.3**	88.8±0.6	**92.6±5.1**	86.4±6.1	**92.6±2.9**
PROTEINS	67.5±3.1	65.8±3.3	67.3±3.5	70.9±3.1	70.2±2.1	70.7±1.9	**73.9±1.7**	72.8±0.5	NA	**73.6±2.6**	**77.7±3.7**

4.3 Ablation Studies and Sensitivity Analysis

Standalone and alternative fusion. Table 6 shows that TOPOGRID and GRAPHGRID individually capture complementary cues, motivating explicit cross-view alignment. Early fusion via 5 CHANNELS yields only modest gains over the stronger standalone branch, indicating that naive channel stacking does not reliably exploit complementarity. CONCAT and LATE FUSION improve on several

Table 6: **Standalone and Fusion Models.** Standalone and fusion results when combining TOPOGRID and GRAPHGRID. **Best**, **second**, and **third** performances are highlighted. SUPCON row represent our fusion model for G2IMAGE.

Method	MUTAG	BZR	PTC-MR	COX2	PROTEINS	IMDB-B	IMDB-M	Avg.	Av.Rank↓
TOPOGRID	92.60±6.73	**92.82±3.61**	68.58±4.64	82.91±5.81	**77.72±3.66**	**76.60±2.58**	51.00±2.86	77.46	3.86
GRAPHGRID	**93.63±4.64**	90.37±2.82	**70.93±5.53**	**84.83±5.69**	76.19±2.24	75.20±2.99	**52.07±4.29**	77.60	4.14
5 CHANNELS	93.10±5.77	**91.84±3.54**	69.15±4.39	82.48±6.46	**77.63±3.78**	**76.70±2.90**	51.53±3.77	77.49	4.07
CONCAT	**94.18±6.84**	**92.32±3.96**	69.75±4.68	82.70±6.65	**77.53±4.08**	76.30±1.73	51.60±3.06	**77.77**	**3.36**
LATE FUSION	**94.15±6.47**	90.85±3.56	**70.03±5.37**	**85.68±4.41**	76.64±2.31	76.00±2.37	**52.53±3.60**	**77.98**	**3.29**
SUPCON	**94.18±5.98**	**91.84±2.98**	**72.38±4.17**	**84.62±5.15**	76.46±3.15	**77.30±1.55**	**52.73±3.57**	**78.50**	**2.29**

datasets but are inconsistent across benchmarks. G2IMAGE (SUPCON) provides the most uniform improvements, achieving the best results on MUTAG, PTC-MR, IMDB-B, and IMDB-M while remaining near the top elsewhere. Aligning the two fixed-budget views with a supervised contrastive objective is strictly more effective than standard fusion strategies.

TOPOGRID variants. Table 7 compares TOPOGRID variants. Single-persistence MLP summaries (SP-HKS, SP-DC) underperform multipersistence, showing that the bifiltration captures complementary structure. Flattening the MP grid and using an MLP further degrades performance, suggesting that the CNN exploits local spatial and cross-channel patterns in the $4 \times 10 \times 10$ tensor. The full MP+CNN model is strongest overall.

Table 7: **TopoGrid vs. topological variants.** Accuracy comparison of single-persistence variants (SP-HKS, SP-Degree Centrality) and multipersistence (HKS + Degree Centrality) variants across seven benchmarks.

Method	MUTAG	BZR	PTC-MR	COX2	PROTEINS	IMDB-B	IMDB-M
SP-HKS + MLP	91.49±5.38	84.95±7.19	65.71±4.78	79.20±5.13	76.82±3.65	73.70±3.03	49.13±2.91
SP-DC + MLP	87.22±7.17	83.70±7.89	62.54±11.20	78.99±5.51	76.92±3.29	72.70±3.72	49.73±3.77
MP + MLP	89.39±7.79	88.90±6.34	63.97±6.13	82.86±4.34	77.10±3.03	75.50±2.33	49.80±3.77
MP + CNN (TOPOGRID)	**92.60**±6.73	**92.82**±3.61	**68.58**±4.64	**82.91**±5.81	**77.72**±3.66	**76.60**±2.58	**51.00**±2.86

TOPOGRID channel ablation. Table 8 shows that Betti-0/1 surfaces are already competitive, while node and edge count channels add complementary global statistics. Combining all four channels gives the best overall results.

Table 8: **TOPOGRID channel ablation.** CNN performance using Betti-only channels, node/edge-count channels, and all four channels of the $4 \times 10 \times 10$ TOPOGRID tensor.

Method	MUTAG	BZR	PTC-MR	COX2	PROTEINS	IMDB-B	IMDB-M
Betti only (2 channels)	90.96±6.68	91.85±2.21	66.28±4.72	**83.13**±6.14	75.20±3.41	72.20±2.44	47.20±2.76
Nodes+Edges (2 channels)	91.02±7.81	91.60±2.02	**68.62**±3.93	81.42±6.03	76.10±3.29	75.70±2.15	50.60±1.75
All (4 channels)	**92.60**±6.73	**92.82**±3.61	68.58±4.64	82.91±5.81	**77.72**±3.66	**76.60**±2.58	**51.00**±2.86

Sensitivity to GRAPHGRID resolution and binning. Table 9 shows that GRAPHGRID is robust to discretization. A 10×10 grid performs best overall: coarser grids lose relational detail, while finer grids add sparsity. Likewise, $B = 10$ bins are most consistent; fewer bins underrepresent structure, and more bins add sparsity without clear gains.

Further ablations, including sensitivity to the choice of score functions, the supervised contrastive objective, and size- and density-controlled splits, are reported in Appendix F.

Table 9: **GraphGrid sensitivity analysis.** Resolution ablation (top): binning fixed at $B = 10$, grid resolution varied. Binning ablation (bottom): resolution fixed at 10×10, number of bins varied. The default configuration (10×10, $B = 10$) is shown in the last row. All values are accuracy (%, mean±std).

	Setting	MUTAG	BZR	PTC-MR	COX2	PROTEINS	IMDB-B	IMDB-M
Resolution	28×28	89.91±7.60	85.94±3.60	69.46±3.12	82.48±5.28	**76.64±3.01**	74.30±3.26	51.40±3.68
	64×64	90.96±6.25	85.69±3.60	67.71±4.79	78.22±7.01	75.47±3.48	74.20±2.48	**52.60±3.81**
Binning	No bins	90.44±7.36	88.13±2.73	66.54±5.12	83.13±5.28	**76.64±2.87**	**76.60±2.94**	50.07±3.05
	5 bins	90.96±5.79	86.41±3.73	68.86±5.71	81.85±6.33	75.56±2.65	73.10±3.33	51.47±3.22
	20 bins	92.54±6.85	88.88±3.00	68.89±3.49	81.20±6.22	76.01±2.26	74.00±2.32	50.87±3.18
Default (10×10, 10 bins)		**93.63±4.64**	**90.37±2.82**	**70.93±5.53**	**84.83±5.69**	76.19±2.24	75.20±2.99	52.07±4.29

5 Conclusion

In this work, we introduced G2IMAGE, a fixed-budget graph imaging interface that converts variable-size graphs into compact, permutation-invariant tensors. It combines two complementary views: TOPOGRID, which captures stable topological structure through multipersistence surfaces, and GRAPHGRID, which captures higher-bandwidth structure through block-pooled edge-density maps. With standard 2D encoders and supervised cross-view alignment, G2IMAGE provides a reusable alternative to graph-specific architectures while retaining control over resolution and stability. Empirically, it achieves strong performance across graph classification and molecular prediction benchmarks, with consistent gains over single-view and standard fusion variants. More broadly, fixed-budget graph images offer a practical interface for reusable graph representation learning and transfer across heterogeneous graph domains.

Acknowledgements

This work was partially supported by National Science Foundation under grant DMS-2229417 and by Simons Foundation under grant SFM-21269. The authors acknowledge the Texas Advanced Computing Center (TACC) at UT Austin for providing computational resources that have contributed to the research results reported within this paper.

References

[AAJ+22] Dashti Ali, Aras Asaad, Maria-Jose Jimenez, Vidit Nanda, Eduardo Paluzo-Hidalgo, and Manuel Soriano-Trigueros. A survey of vectorization methods in topological data analysis. *arXiv preprint arXiv:2212.09703*, 2022.

[BL22] Magnus Bakke Botnan and Michael Lesnick. An introduction to multiparameter persistence. *arXiv preprint arXiv:2203.14289*, 2022.

[C+24] Yuzhou Chen et al. Topogcl: Topological graph contrastive learning. In *AAAI*, volume 38, pages 11453–11461, 2024.

[CA24] Baris Coskunuzer and Cüneyt Gürcan Akçora. Topological methods in machine learning: A tutorial for practitioners. *arXiv preprint arXiv:2409.02901*, 2024.

[CB20] Mathieu Carrière and Andrew Blumberg. Multiparameter persistence image for topological machine learning. *NeurIPS*, 33, 2020.

[CFK+19] René Corbet, Ulderico Fugacci, Michael Kerber, Claudia Landi, and Bei Wang. A kernel for multi-parameter persistent homology. *Computers & graphics: X*, 2:100005, 2019.

[CKNH20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[CLLG+24] Semih Cantürk, Renming Liu, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder. In *International Conference on Machine Learning*, pages 5533–5566. PMLR, 2024.

[CMSD23] X Cheng, S Mukherjee, S Samaga, and TK Dey. Gril: A 2-parameter persistence based vectorization for machine learning. In *Proc. ICML 2023 workshop TAGML*, 2023.

[CPLM21] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021.

[CSDA+24] Yuzhou Chen, Ignacio Segovia-Dominguez, Cuneyt Gurcan Akcora, Zhiwei Zhen, Murat Kantarcioglu, Yulia Gel, and Baris Coskunuzer. Emp: Effective multidimensional persistence for graph representation learning. In *Learning on Graphs Conference*, pages 24–1. PMLR, 2024.

[CVJ+18] Cătălina Cangea, Petar Veličković, Nikola Jovanović, Thomas Kipf, and Pietro Liò. Towards sparse hierarchical graph classifiers. *arXiv preprint arXiv:1811.01287*, 2018.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DJL+23] 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.

[DW22] Tamal Krishna Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022.

[Eis13] David Eisenbud. *Commutative algebra: with a view toward algebraic geometry*, volume 150. Springer Science & Business Media, 2013.

[EP14] Herbert Edelsbrunner and Salman Parsa. On the computational complexity of betti numbers: reductions from matrix rank. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*, pages 152–160. SIAM, 2014.

[EPBM20] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *ICLR*, 2020.

[FHSK23] Zhongxi Fang, Jianming Huang, Xun Su, and Hiroyuki Kasai. Wasserstein graph distance based on l1–approximated tree edit distance between weisfeiler–lehman subtrees. In *AAAI*, volume 37, pages 7539–7549, 2023.

[GSA+20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[HCX+22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[HFZ+20] 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. *NeurIPS*, 33:22118–22133, 2020.

[HLAK24] Jaeseung Heo, Seungbeom Lee, Sungsoo Ahn, and Dongwoo Kim. Epic: graph augmentation with edit path interpolation via learnable cost. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 4116–4126, 2024.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[HYL17] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *NeurIPS*, 30, 2017.

[JYX+21] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, volume 139, pages 4904–4916. PMLR, 2021.

[KTW+20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[KW17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

[LCB23] David Loiseaux, Mathieu Carrière, and Andrew J Blumberg. A framework for fast and stable representations of multiparameter persistent homology decompositions. *NeurIPS*, 2023.

[LCS+24] Zhengdao Li, Yong Cao, Kefan Shuai, Yiming Miao, and Kai Hwang. Rethinking the effectiveness of graph classification datasets in benchmarks for assessing gnns. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 2144–2152, 2024.

[LHZ23] Yecheng Lyu, Xinming Huang, and Ziming Zhang. Revisiting 2D convolutional neural networks for graph-based applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6909–6922, 2023.

[LSC+23] David Loiseaux, Luis Scoccola, Mathieu Carrière, Magnus Bakke Botnan, and Steve Oudot. Stable vectorization of multiparameter persistent homology using signed barcodes as measures. *NeurIPS*, 2023.

[LSC+24] David Loiseaux, Luis Scoccola, Mathieu Carrière, Magnus Bakke Botnan, and Steve Oudot. Stable vectorization of multiparameter persistent homology using signed barcodes as measures. *NeurIPS*, 36, 2024.

[LW15] Michael Lesnick and Matthew Wright. Interactive visualization of 2-D persistence modules. *arXiv preprint arXiv:1512.00180*, 2015.

[MKB+20] 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.

[NAK16] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023. PMLR, 2016.

[OS21] Steve Oudot and Luis Scoccola. On the stability of multigraded betti numbers and hilbert functions. *arXiv preprint arXiv:2112.11901*, 2021.

[PMFW21] Pál András Papp, Karolis Martinkus, Lukas Faber, and Roger Wattenhofer. Dropgnn: Random dropouts increase the expressiveness of graph neural networks. *Advances in Neural Information Processing Systems*, 34:21997–22009, 2021.

[QBC+25] Xinya Qin, Lu Bai, Lixin Cui, Ming Li, Hangyuan Du, and Edwin Hancock. Multinet: Adaptive multi-viewed subgraph convolutional networks for graph classification. In *NeurIPS*, 2025.

[QTLL24] Chen Qian, Huayi Tang, Hong Liang, and Yong Liu. Reimagining graph classification from a prototype view with optimal transport: Algorithm and theorem. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2444–2454, 2024.

[R+21] Alec Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021.

[RGD+22] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

[RK24] Florian Russold and Michael Kerber. Graphcode: Learning from multiparameter persistent homology using graph neural networks. In *NeurIPS*, 2024.

[SHB16] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics*, pages 1715–1725, 2016.

[SOG09] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009.

[SSL+24] Luis Scoccola, Siddharth Setlur, David Loiseaux, Mathieu Carrière, and Steve Oudot. Differentiability and optimization of multiparameter persistent homology. In *ICML*, 2024.

[SSVL+11] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[TTA+22] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. In *International Conference on Learning Representations*, 2022.

[TTAC25] Astrit Tola, Funmilola Mary Taiwo, Cuneyt Gurcan Akcora, and Baris Coskunuzer. Toper: Topological embeddings in graph representation learning. *NeurIPS*, 2025.

[VCC+18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR*, 2018.

[VCFC+22] Cédric Vincent-Cuaz, Rémi Flamary, Marco Corneli, Titouan Vayer, and Nicolas Courty. Template based graph neural network with optimal transport distances. *Advances in Neural Information Processing Systems*, 35:11800–11814, 2022.

[Vip20] Oliver Vipond. Multiparameter persistence landscapes. *Journal of Machine Learning Research*, 21:61–1, 2020.

[vLV18] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[WPC+20] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[XHLJ18] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ICLR*, 2018.

[Y+21] Chengxuan Ying et al. Do transformers really perform badly for graph representation? *NeurIPS*, 34:28877–28888, 2021.

[YCS+20] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[YYM+18] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, volume 31, pages 4800–4810. Curran Associates, Inc., 2018.

[ZCH+20] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[ZCNC18] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[ZXY+20] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

Appendix

A Stability Results

This appendix provides full proofs for the invariance and stability results stated in Section 3.4. The results collectively characterize the mathematical behavior of G2IMAGE as a function of its inputs: what is preserved exactly, what changes continuously, and where information loss can occur. We organize the proofs into two subsections, one for each view.

A.1 Proofs for TOPOGRID

We begin with the two exact preservation results. Permutation invariance holds because both views are defined entirely through permutation-equivariant node scores: the bifiltration vertex sets and bin groups depend on score values, not vertex identities, so any relabeling of vertices leaves every tensor entry unchanged. Local constancy holds because threshold membership is a discrete event: as long as no score crosses a grid boundary, the induced subgraphs, clique complexes, and hence all surface channels are identical under perturbation.

Proposition 3.1 [Permutation invariance] *Let π be any permutation of the vertices of $\mathcal{G}$. If the intrinsic score functions used by* TOPOGRID *and* GRAPHGRID *are permutation-equivariant, then $\mathcal{X}_{\text{top}}(\pi\mathcal{G}) = \mathcal{X}_{\text{top}}(\mathcal{G})$ and $\mathcal{X}_{\text{img}}(\pi\mathcal{G}) = \mathcal{X}_{\text{img}}(\mathcal{G})$.*

Proof of Proposition 3.1. Both HKS and degree centrality are permutation-equivariant: their values depend only on graph structure, not vertex labels. For TOPOGRID, the bifiltration vertex sets $\mathcal{V}_{ij} = \{v : f(v) \le \alpha_i,\ g(v) \le \beta_j\}$ depend only on the values of (f, g), not vertex identities. Hence the induced subgraphs, clique complexes, Betti numbers, and count channels are unchanged under any permutation π, giving $\mathcal{X}_{\text{top}}(\pi\mathcal{G}) = \mathcal{X}_{\text{top}}(\mathcal{G})$. For GRAPHGRID, the bin groups $\mathcal{C}_\tau = \{v : \widetilde{\Phi}_{\mathcal{G}}(v) = \tau\}$ are defined by binned score values; since $\Phi_{\mathcal{G}}$ is permutation-equivariant, the bin partition and density matrix $D(\mathcal{G})$ are identical under π, hence so is $I_k(\mathcal{G})$. □

Proposition 3.2 [Local constancy of TOPOGRID] *Fix $\mathcal{G}$ and grids $\{\alpha_i\}, \{\beta_j\}$ with margins $m_f = \min_{v,i} |f(v) - \alpha_i|$ and $m_g = \min_{v,j} |g(v) - \beta_j|$. If $\|f - \tilde{f}\|_\infty < m_f$ and $\|g - \tilde{g}\|_\infty < m_g$, then $\mathcal{X}_{\text{top}}(\mathcal{G}; f, g) = \mathcal{X}_{\text{top}}(\mathcal{G}; \tilde{f}, \tilde{g})$.*

Proof of Proposition 3.2. Since $\|f - \tilde{f}\|_\infty < m_f$, no vertex v crosses any f-threshold α_i: the membership $f(v) \le \alpha_i$ is preserved for every v and i. The same argument applies to the g-direction using $\|g - \tilde{g}\|_\infty < m_g$. Therefore $\mathcal{V}_{ij}(f, g) = \mathcal{V}_{ij}(\tilde{f}, \tilde{g})$ for all (i, j), and hence the induced subgraphs, clique complexes, Betti numbers $S^{(0)}, S^{(1)}$, and count channels $S^{(V)}, S^{(E)}$ are all unchanged. □

Local constancy characterizes the best case: exact preservation whenever scores stay within their grid cells. The following remark characterizes the worst case: when nodes do cross thresholds, how much can the representation change? The answer is local — changes are bounded by the number of crossing vertices and their neighborhood structure, not by global graph properties.

Remark A.1 (Threshold-crossing sensitivity). When the margin conditions of Proposition 3.2 are violated, let $Q = \{v : \exists\, i, j \text{ s.t. membership of } v \text{ in } \mathcal{V}_{ij} \text{ changes}\}$ be the set of threshold-crossing vertices. The count channels satisfy

$$\begin{aligned} \|S^{(V)}(\mathcal{G}; f, g) - S^{(V)}(\mathcal{G}; \tilde{f}, \tilde{g})\|_1 &\le MN\,|Q|, \\ \|S^{(E)}(\mathcal{G}; f, g) - S^{(E)}(\mathcal{G}; \tilde{f}, \tilde{g})\|_1 &\le MN \sum_{v \in Q} \deg(v). \end{aligned}$$

For the Betti channels, adding or removing a vertex v and its incident edges can change β_0 and β_1 by at most a constant times $1 + \deg(v)$ per affected grid cell, yielding the crude bound

$$\|S^{(r)}(\mathcal{G}; f, g) - S^{(r)}(\mathcal{G}; \tilde{f}, \tilde{g})\|_1 \le C\,MN \sum_{v \in Q} (1 + \deg(v)), \qquad r \in \{0, 1\}.$$

All changes are thus localized to threshold-crossing vertices and their incident edges.

A.2 Proofs for GRAPHGRID

The GRAPHGRID stability analysis has a different character from the TOPOGRID analysis. Rather than threshold crossings, the relevant events are *bin swaps*: changes to a vertex's binned score signature $\widetilde{\Phi}_{\mathcal{G}}(v)$ that reassign it to a different bin group. When no bin swaps occur, the bin partition is fixed and GRAPHGRID is Lipschitz in the edge perturbation $\|A - A'\|_1$. When bin swaps do occur, the additional error is controlled by the fraction of reassigned vertices. The lemma below identifies the condition under which bin swaps cannot occur; the theorem then gives the general bound.

Setup and notation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ share vertex set $\mathcal{V}$ with $n = |\mathcal{V}|$, and let $A, A' \in \mathbb{R}^{n\times n}$ with $A_{uv}, A'_{uv} \in [0, W]$. Let $\mathcal{C}_1, \dots, \mathcal{C}_m$ be the nonempty bin groups under $\widetilde{\Phi}_{\mathcal{G}}$ in lexicographic order, with $c_{\min} := \min_p |\mathcal{C}_p|$ the minimum bin-group size. The density matrix $D(\mathcal{G}) \in \mathbb{R}^{m\times m}$ has entries

$$D(\mathcal{G})_{pq} = \frac{1}{|\mathcal{C}_p||\mathcal{C}_q|} \sum_{u\in\mathcal{C}_p} \sum_{v\in\mathcal{C}_q} A_{uv}.$$

The pooled image $I_k(\mathcal{G})$ partitions $\{1, \dots, m\}$ into k consecutive blocks $\mathcal{B}_1, \dots, \mathcal{B}_k$ of bin groups, with $b_{\min} := \min_a |\mathcal{B}_a|$. For balanced nonempty blocks, $b_{\min} \geq \lfloor m/k \rfloor$. If $m < k$, we use m nonempty blocks and zero-pad the remaining entries. Define $\mathrm{BS}(\mathcal{G}, \mathcal{G}') := \frac{1}{n}|\{v : \widetilde{\Phi}_{\mathcal{G}}(v) \neq \widetilde{\Phi}_{\mathcal{G}'}(v)\}|$.

Lemma 3.3 [Bin stability] *Let* $\gamma = \min_{v,\ell,b} |\Phi_{\mathcal{G}}(v)_\ell - q_{\ell,b}|$ *where* $q_{\ell,b}$ *is the* b*-th bin boundary for coordinate* ℓ*. If* $\max_v \|\Phi_{\mathcal{G}}(v) - \Phi_{\mathcal{G}'}(v)\|_\infty < \gamma$*, then* $\mathrm{BS}(\mathcal{G}, \mathcal{G}') = 0$*.*

Proof of Lemma 3.3. Under the stated condition, no score coordinate of any vertex crosses a bin boundary, so $\widetilde{\Phi}_{\mathcal{G}}(v) = \widetilde{\Phi}_{\mathcal{G}'}(v)$ for all $v \in V$, giving $\mathrm{BS}(\mathcal{G}, \mathcal{G}') = 0$ by definition. □

When $\mathrm{BS}(\mathcal{G}, \mathcal{G}') = 0$, the bin partition is shared between $\mathcal{G}$ and $\mathcal{G}'$ and the stability of I_k reduces entirely to controlling how edge edits propagate through the density matrix and block pooling. The following theorem handles both the fixed-partition case and the general case with bin swaps.

Theorem 3.4 [Stability of GRAPHGRID] Let $\mathcal{G}, \mathcal{G}'$ share a vertex set with adjacencies $A, A' \in [0, W]^{n\times n}$. Let $b_{\min} := \min_a |\mathcal{B}_a|$ be the minimum nonempty pooling-block size, $c_{\min} := \min_p |\mathcal{C}_p|$ the minimum nonempty bin-group size under the reference partition of $\mathcal{G}$, and $\mathrm{BS}(\mathcal{G}, \mathcal{G}') := \frac{1}{n}|\{v : \widetilde{\Phi}_{\mathcal{G}}(v) \neq \widetilde{\Phi}_{\mathcal{G}'}(v)\}|$. Then, for an absolute constant C,

$$\|I_k(\mathcal{G}) - I_k(\mathcal{G}')\|_1 \;\leq\; \frac{\|A - A'\|_1}{b_{\min}^2\, c_{\min}^2} \;+\; \min\Big(2Wk^2,\; CWk^2\, n\, \mathrm{BS}(\mathcal{G}, \mathcal{G}')\Big).$$

In particular, if no vertex changes bin then $\mathrm{BS}(\mathcal{G}, \mathcal{G}') = 0$ and the bound reduces to $\|I_k(\mathcal{G}) - I_k(\mathcal{G}')\|_1 \leq (b_{\min}\, c_{\min})^{-2}\|A - A'\|_1$.

Proof of Theorem 3.4. We decompose the change in I_k into two terms: one from edge edits under a fixed bin partition, and one from bin-group changes.

(I) Edge edits under fixed bin partition. Assume $\mathrm{BS}(\mathcal{G}, \mathcal{G}') = 0$, so the bin groups are identical between $\mathcal{G}$ and $\mathcal{G}'$. Then $D(\mathcal{G})$ and $D(\mathcal{G}')$ share the same index set with the same group sizes. For each bin pair (p, q),

$$|D(\mathcal{G})_{pq} - D(\mathcal{G}')_{pq}| \;\leq\; \frac{1}{|\mathcal{C}_p||\mathcal{C}_q|} \sum_{u\in\mathcal{C}_p} \sum_{v\in\mathcal{C}_q} |A_{uv} - A'_{uv}|.$$

Summing over all (p, q) and using $|\mathcal{C}_p||\mathcal{C}_q| \geq c_{\min}^2$,

$$\|D(\mathcal{G}) - D(\mathcal{G}')\|_1 \;\leq\; \frac{1}{c_{\min}^2}\|A - A'\|_1.$$

Each pooled entry satisfies

$$|I_k(\mathcal{G})_{ab} - I_k(\mathcal{G}')_{ab}| \;\leq\; \frac{1}{|\mathcal{B}_a||\mathcal{B}_b|} \sum_{p\in\mathcal{B}_a} \sum_{q\in\mathcal{B}_b} |D(\mathcal{G})_{pq} - D(\mathcal{G}')_{pq}|.$$

Summing over (a, b) and using $|\mathcal{B}_a||\mathcal{B}_b| \geq b_{\min}^2$,

$$\text{(I)} := \|I_k(\mathcal{G}) - I_k(\mathcal{G}')\|_1 \;\leq\; \frac{1}{b_{\min}^2}\|D(\mathcal{G}) - D(\mathcal{G}')\|_1 \;\leq\; \frac{\|A - A'\|_1}{b_{\min}^2\, c_{\min}^2}.$$

(II) Bin-group changes. Let $S = \{v : \widetilde{\Phi}_{\mathcal{G}}(v) \neq \widetilde{\Phi}_{\mathcal{G}'}(v)\}$, so $|S| = n\,\mathrm{BS}(\mathcal{G}, \mathcal{G}')$. Since $I_k(\mathcal{G})$ and $I_k(\mathcal{G}')$ each have k^2 entries bounded in $[0, W]$, the trivial bound gives $\|I_k(\mathcal{G}) - I_k(\mathcal{G}')\|_1 \leq 2Wk^2$. Alternatively, each reassigned vertex in S affects the density entries of its old and new bin groups. Since the pooled image has k^2 entries each bounded in $[0, W]$, the contribution per reassigned vertex to $\|I_k(\mathcal{G}) - I_k(\mathcal{G}')\|_1$ is at most CWk^2. Summing over $|S| = n\,\mathrm{BS}(\mathcal{G}, \mathcal{G}')$ reassigned vertices,

$$\text{(II)} \;\leq\; \min\Big(2Wk^2,\; CWk^2\, n\,\mathrm{BS}(\mathcal{G}, \mathcal{G}')\Big).$$

Combining. Adding (I) and (II) gives

$$\|I_k(\mathcal{G}) - I_k(\mathcal{G}')\|_1 \;\leq\; \frac{\|A - A'\|_1}{b_{\min}^2\, c_{\min}^2} \;+\; \min\Big(2Wk^2,\; CWk^2\, n\,\mathrm{BS}(\mathcal{G}, \mathcal{G}')\Big),$$

as claimed. Setting $\mathrm{BS}(\mathcal{G}, \mathcal{G}') = 0$ yields the fixed-partition bound immediately. □

Remark A.2 (Block-pooling approximation error). The information discarded by GRAPHGRID at resolution k has a precise characterization. Let $P_k D(\mathcal{G})$ denote the blockwise-constant reconstruction of $D(\mathcal{G})$ from $I_k(\mathcal{G})$, where each entry $D(\mathcal{G})_{pq}$ for $p \in \mathcal{B}_a$, $q \in \mathcal{B}_b$ is replaced by $\bar{D}_{ab} = I_k(\mathcal{G})_{ab}$. Then

$$\|D(\mathcal{G}) - P_k D(\mathcal{G})\|_F^2 = \sum_{a,b=1}^{k} \sum_{p \in \mathcal{B}_a} \sum_{q \in \mathcal{B}_b} \big(D(\mathcal{G})_{pq} - \bar{D}_{ab}\big)^2.$$

The discarded information is exactly the within-block variance of the bin-level density matrix. Increasing k reduces this error but increases sparsity and model variance, directly motivating the resolution ablation in Section 4.3. This provides a direct interpretation of the resolution parameter k without appealing to a low-rank approximation argument.

Remark A.3 (Resolution and block-pooling error). The resolution parameter k controls the information discarded by GRAPHGRID. Let $D(\mathcal{G}) \in \mathbb{R}^{m \times m}$ be the bin-level density matrix and let $P_k D(\mathcal{G})$ be its blockwise-constant reconstruction from the pooled image $I_k(\mathcal{G})$. Then

$$\|D(\mathcal{G}) - P_k D(\mathcal{G})\|_F^2 = \sum_{a,b=1}^{k} \sum_{p \in \mathcal{B}_a} \sum_{q \in \mathcal{B}_b} \big(D(\mathcal{G})_{pq} - \bar{D}_{ab}\big)^2,$$

where $\bar{D}_{ab}$ is the average density in block (a, b). Thus the information removed by pooling is exactly the within-block variance of the bin-level density matrix. For nested refinements of the partition, increasing k makes this aggregation error non-increasing, but may also increase sparsity and model variance. This motivates the resolution ablation.

B Persistent Homology Background

Persistent homology (PH) is a standard tool in topological data analysis for summarizing topological features across scales [DW22]. In graph learning, PH is commonly used to construct descriptors of connectivity, cycles, and higher-order organization that may not be captured by local message passing alone [CA24].

Single-parameter filtrations. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and let $f : \mathcal{V} \to \mathbb{R}$ be a node filtration function. Given thresholds $\mathcal{I} = \{\epsilon_1 < \epsilon_2 < \cdots < \epsilon_N\}$, the sublevel filtration activates

$$\mathcal{V}_i = \{v \in \mathcal{V} : f(v) \leq \epsilon_i\}, \qquad \mathcal{E}_i = \{(u, v) \in \mathcal{E} : u, v \in \mathcal{V}_i\}.$$

This yields a nested sequence

$$\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \cdots \subseteq \mathcal{G}_N = \mathcal{G}.$$

Passing to clique complexes gives a simplicial filtration

$$\widehat{\mathcal{G}}_1 \subseteq \widehat{\mathcal{G}}_2 \subseteq \cdots \subseteq \widehat{\mathcal{G}}_N.$$

Superlevel filtrations are defined analogously using $f(v) \geq \epsilon_i$. Edge-based filtrations can also be built from edge weights, curvature scores, or other edge functions.

As the filtration evolves, topological features appear and disappear. Their birth and death times form persistence diagrams $\mathrm{PD}_k(\mathcal{G})$ indexed by homology dimension k. For graphs and clique complexes, $k = 0$ captures connected components, while $k = 1$ captures cycles. Common graph filtration functions include degree, centrality, betweenness, heat kernel signatures (HKS) [SOG09], curvature, and domain-specific attributes.

Vectorization. Persistence diagrams are multisets in $\mathbb{R}^2$ and must be vectorized before they can be used in standard learning pipelines. Common vectorizations include persistence images, landscapes, silhouettes, and Betti curves [AAJ+22]. These summaries trade completeness for stability, finite dimensionality, and computational tractability.

C Multiparameter Persistence: Background and Design Motivation

Multiparameter persistent homology (MP-PH) extends PH by filtering a graph with more than one function. This can capture complementary structural information that may be invisible to any single filtration parameter. However, MP-PH is also substantially harder to summarize than ordinary one-dimensional persistence, because general multiparameter modules do not admit a complete barcode-style representation.

Bifiltrations on graphs. Let $f, g : \mathcal{V} \to \mathbb{R}$ be two node functions that capture complementary graph structure, such as diffusion geometry through HKS and local connectivity through degree. Define the coordinate-wise partial order on $\mathbb{R}^2$ by

$$(\alpha, \beta) \preceq (\alpha', \beta') \quad \Longleftrightarrow \quad \alpha \leq \alpha' \text{ and } \beta \leq \beta'.$$

The pair (f, g) defines the sublevel bifiltration

$$\mathcal{V}_{(\alpha,\beta)} = \{v \in \mathcal{V} : \ f(v) \leq \alpha, \ g(v) \leq \beta\}, \qquad \mathcal{G}_{(\alpha,\beta)} = \mathcal{G}[\mathcal{V}_{(\alpha,\beta)}].$$

Passing to clique complexes gives a family $\{\widehat{\mathcal{G}}_{(\alpha,\beta)}\}_{(\alpha,\beta)\in\mathbb{R}^2}$ that is monotone with respect to $\preceq$. Discretizing thresholds $\{\alpha_i\}_{i=1}^{M}$ and $\{\beta_j\}_{j=1}^{N}$ yields a grid-indexed bifiltration $\{\widehat{\mathcal{G}}_{ij}\}$ satisfying

$$\widehat{\mathcal{G}}_{ij} \subseteq \widehat{\mathcal{G}}_{i'j'} \qquad \text{whenever} \qquad i \leq i', \ j \leq j'.$$

For each homological dimension k, the assignments

$$M_k : (i, j) \longmapsto H_k(\widehat{\mathcal{G}}_{ij}), \qquad (i, j) \preceq (i', j') \longmapsto H_k(\widehat{\mathcal{G}}_{ij}) \hookrightarrow H_k(\widehat{\mathcal{G}}_{i'j'})$$

form a bi-graded persistence module over the grid poset. Useful summaries include the Betti surface

$$\beta_k(i, j) = \dim H_k(\widehat{\mathcal{G}}_{ij})$$

and the rank invariant

$$r_k\big((i, j) \preceq (i', j')\big) = \operatorname{rank} H_k(\widehat{\mathcal{G}}_{ij} \to \widehat{\mathcal{G}}_{i'j'}).$$

Why one-dimensional persistence admits barcodes. In the single-parameter setting, filtration indices lie in a totally ordered set. Topological features can therefore be assigned consistent birth and death times. For pointwise finite-dimensional modules indexed by a total order, the structure theorem gives an interval decomposition, which underlies persistence barcodes and diagrams [BL22].

Why multiparameter persistence is harder. In the two-parameter setting, the index set is a partially ordered set rather than a total order. There is no global linear notion of birth and death, and a direct barcode-style decomposition does not exist for general multiparameter modules [BL22]. Algebraic invariants such as minimal free resolutions and graded Betti numbers are well studied [Eis13], but they do not yield a simple, complete, and easily vectorizable analogue of persistence diagrams.

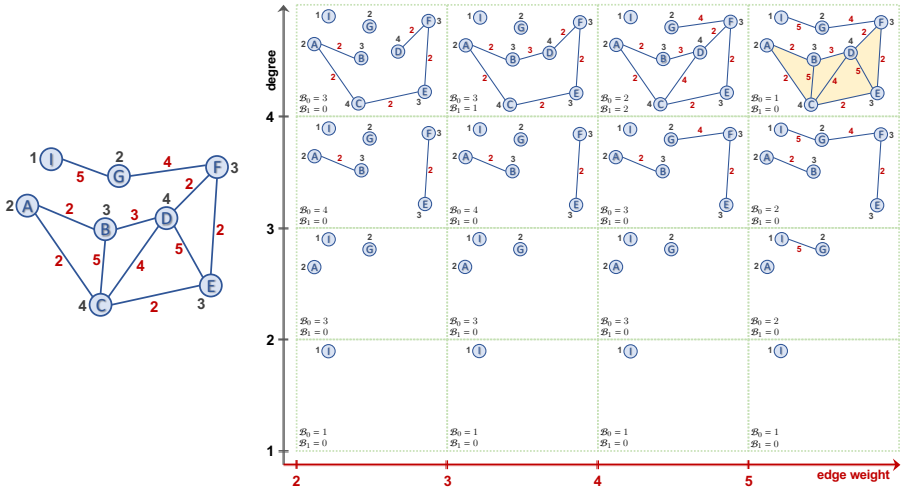

Figure 2: **Multiparameter persistence.** In the two-parameter setting, the filtration index set is a poset rather than a total order. Each grid cell (i, j) corresponds to the subgraph induced by nodes satisfying both threshold conditions. This prevents a direct barcode-style decomposition and motivates fixed-grid summaries such as TopoGrid.

Practical summaries and tradeoffs. Since a complete discrete invariant is generally unavailable, practical methods use stable partial summaries. Slicing methods restrict the bifiltration to one-dimensional directions and aggregate ordinary persistence across slices [LW15, CB20]. These methods are flexible, but the choice of directions and aggregation affects the resulting representation. Landscape- and image-based extensions build vectorizations more directly from multiparameter invariants [Vip20], but can be computationally expensive. Recent learning-oriented pipelines [LCB23, LSC+24, CMSD23, CSDA+24, RK24, SSL+24] show that MP information is useful for graph learning, but often require nontrivial preprocessing or specialized architectures.

Motivation for TopoGrid. TopoGrid occupies a fixed-budget point in this design space. We discretize an intrinsic bifiltration on a small $M \times N$ grid and record low-dimensional surface summaries as image-like channels. In our implementation, these channels are Betti-0, Betti-1, vertex count, and edge count surfaces. This design has three practical advantages. First, the output size is fixed and predictable. Second, the grid structure is directly compatible with standard 2D encoders. Third, the representation is locally stable at fixed resolution: perturbations of the filtration scores that do not cross grid boundaries leave the tensor unchanged, while boundary crossings affect only localized cells through the crossing vertices and their incident edges.

The cost is loss of completeness. TopoGrid does not preserve the full multiparameter persistence module, nor is it intended to reconstruct the input graph. It is a compact, stable, and permutation-invariant topological image designed for representation learning.

D Implementation Details

Encoder architecture. Both TopoGrid and GraphGrid are processed by independent lightweight 2D encoders with the same architecture. Each encoder first applies two convolutional blocks, where each block consists of a 3×3 convolution with padding 1, batch normalization, ReLU activation, and dropout. The resulting feature map is flattened into a sequence of spatial tokens and projected to a d_{model}-dimensional embedding. We prepend a learnable `[CLS]` token and process the sequence using a Transformer encoder with 2 layers and 4 attention heads. The final `[CLS]` representation is used as the view-level graph embedding.

Two-view contrastive classifier. Given the two fixed-budget images $X_{\text{top}}(\mathcal{G})$ and $X_{\text{img}}(\mathcal{G})$, the two encoders produce embeddings $h_{\text{top}}, h_{\text{img}} \in \mathbb{R}^{d_{\text{model}}}$. Each embedding is passed through a projection head to obtain contrastive representations $z_{\text{top}}, z_{\text{img}} \in \mathbb{R}^{d_{\text{proj}}}$. For classification, we concatenate the two encoder embeddings and apply dropout followed by a linear classifier. The model

is trained with a joint objective

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{CON}},$$

where $\mathcal{L}_{\mathrm{CE}}$ is the supervised classification loss and $\mathcal{L}_{\mathrm{CON}}$ aligns the two views using a cross-view contrastive objective.

G2Image-Chem for OGB datasets. For OGB molecular benchmarks, we extend G2IMAGE by incorporating atom-level attributes through an additional feature-driven view. Specifically, node features are used to construct an attribute-based image via a GRAPHGRID-style binning and pooling procedure. For TOPOGRID, the bifiltration functions are adapted to chemistry-aware signals by using atomic number and formal charge as intrinsic node functions, as these attributes are known to capture key chemical properties in molecular graphs. With this extension, the framework operates on three complementary views capturing topological, structural, and attribute-level information, which are aligned using the same cross-view contrastive objective. The encoder architecture and optimization procedure remain unchanged.

Hyperparameters. We perform a grid search over the following hyperparameters: learning rate $\in \{10^{-4}, 5\times10^{-4}, 10^{-3}\}$, CNN channel size $\in \{16, 32, 64\}$, Transformer hidden dimension $d_{\text{model}} \in \{32, 64, 128\}$, and dropout $\in \{0.0, 0.3\}$. For all TU experiments, we fix the Transformer encoder to 2 layers with 4 attention heads. Based on the performance on TU datasets and computational considerations for larger OGB benchmarks, we fix the architecture for OGB experiments to 64 CNN channels, $d_{\text{model}} = 128$, dropout $= 0.0$, learning rate $= 10^{-3}$ and weight decay $= 10^{-4}$. We additionally tune the contrastive loss weight $\lambda \in \{0.1, 0.2, 0.5\}$.

E Fusion Baselines

We compare G2IMAGE (using Supervised Contrastive fusion) against three fusion strategies that combine TOPOGRID ($T(\mathcal{G}) \in \mathbb{R}^{4\times10\times10}$) and GRAPHGRID ($I(\mathcal{G}) \in \mathbb{R}^{1\times10\times10}$, $k = 10$) without cross-view alignment. All variants use the same 2D encoder architecture described in Appendix D and are trained with cross-entropy only (no contrastive term). Let f_{top} and f_{graph} be 2D encoders outputting $\mathbb{R}^{d_{\mathrm{top}}}$ and $\mathbb{R}^{d_{\mathrm{graph}}}$, and let $\mathrm{MLP}(\cdot)$ denote a feedforward classification head.

5 Channels (early fusion). GRAPHGRID is concatenated channel-wise with TOPOGRID, yielding a single $5\times10\times10$ input tensor:

$$X_{5\mathrm{ch}}(\mathcal{G}) = \mathrm{concat}_{\mathrm{ch}}\big(T(\mathcal{G}),\, I(\mathcal{G})\big) \in \mathbb{R}^{5\times10\times10}, \quad \hat{y}(\mathcal{G}) = \mathrm{MLP}\big(f_{5\mathrm{ch}}(X_{5\mathrm{ch}}(\mathcal{G}))\big).$$

This forces a single encoder to jointly learn from topological and structural channels without any explicit alignment signal.

Concat (feature-level fusion). Two independent encoders produce separate embeddings that are concatenated before classification:

$$z_{\mathrm{top}}(\mathcal{G}) = f_{\mathrm{top}}(T(\mathcal{G})), \quad z_{\mathrm{graph}}(\mathcal{G}) = f_{\mathrm{graph}}(I(\mathcal{G})),$$
$$\hat{y}(\mathcal{G}) = \mathrm{MLP}\big([z_{\mathrm{top}}(\mathcal{G});\, z_{\mathrm{graph}}(\mathcal{G})]\big).$$

This preserves view-specific representations but mixes information only at the embedding level.

Late Fusion (per-view projection before concatenation). Each embedding is passed through a lightweight view-specific MLP ϕ before concatenation:

$$h_{\mathrm{top}}(\mathcal{G}) = \phi_{\mathrm{top}}(z_{\mathrm{top}}(\mathcal{G})) \in \mathbb{R}^{d'}, \quad h_{\mathrm{graph}}(\mathcal{G}) = \phi_{\mathrm{graph}}(z_{\mathrm{graph}}(\mathcal{G})) \in \mathbb{R}^{d'},$$
$$\hat{y}(\mathcal{G}) = \mathrm{MLP}\big([h_{\mathrm{top}}(\mathcal{G});\, h_{\mathrm{graph}}(\mathcal{G})]\big).$$

This adds a small amount of view-specific nonlinearity before mixing while keeping the two encoders independent.

Relation to G2IMAGE (SUPCON). All three baselines above rely on a fixed fusion operator applied after independent encoding. G2IMAGE differs by training both encoders jointly with a supervised cross-view contrastive objective (Section 3), which explicitly encourages the two embeddings to be mutually consistent before they are combined for classification. The ablation results in Table 6 show that this alignment step — not the fusion operator itself — is the primary source of G2IMAGE's consistent gains.

F Additional Ablation Studies and Sensitivity Analysis

GraphGrid function sensitivity Table 10 shows that GraphGrid is largely insensitive to the choice of score functions. We replace the default set (PageRank, HKS, k-core, and degree) with four alternative combinations drawn from a broad pool of local, spectral, and centrality-based descriptors. Across all three benchmarks, accuracy stays within a narrow band: the spread between the best and worst configuration is about 1.2 points on PROTEINS (75.47–76.64) and 1.3 points on IMDB-B (74.80–76.10). On MUTAG it is 2.6 points (91.55–94.15), which is still small relative to the fold-to-fold standard deviations of up to 6.47 on this small dataset. For every dataset, the default configuration lies within one standard deviation of every alternative, so none of the observed differences is distinguishable from run-to-run variability. Although the default is not the top performer on any single dataset, it is consistently competitive, which suggests that the default choice was not tuned to any particular benchmark. This robustness indicates that GraphGrid's performance comes mainly from its grid-based representation of the filtration space rather than from any specific scoring function, so practitioners can substitute functions suited to their domain without a meaningful loss in accuracy.

Table 10: Sensitivity of GraphGrid to the choice of score functions. Mean $\pm$ standard deviation of classification accuracy (%). The default configuration uses PageRank, HKS, k-core, and degree.

Score functions	**MUTAG**	**PROTEINS**	**IMDB-B**
deg, k-core, betweenness, closeness	91.55 ± 0.67	75.47 ± 3.26	76.10 ± 3.51
HKS, deg, betweenness, closeness	93.63 ± 5.21	75.74 ± 2.18	75.60 ± 3.29
HKS, betweenness, closeness, PageRank	94.15 ± 6.47	76.19 ± 2.51	74.80 ± 2.18
eigencentrality, clustering coef., HKS, deg	93.10 ± 5.27	76.64 ± 3.24	76.10 ± 2.62
GRAPHGRID (default)	93.63 ± 4.64	76.19 ± 2.24	75.20 ± 2.99

Effect of cross-view alignment. Table 11 isolates the contribution of aligning the two heterogeneous views. Training either view alone with supervised contrastive loss gives similar but limited performance, and neither dominates across datasets: TopoGrid is stronger on PROTEINS, while the two are comparable on MUTAG and IMDB-B. Simply encoding both views with separate within-view SupCon objectives, without cross-view positives, yields modest improvements on MUTAG (91.52) and IMDB-B (75.90) over either single view, which suggests the two views carry complementary information. However, this combination does not by itself exploit that complementarity, since the two embedding spaces are never tied together. Introducing cross-view positives, so that a graph's topological and structural representations are pulled together while embeddings of other classes are pushed apart, gives the largest gains. Relative to within-view SupCon, G2IMAGE improves accuracy by 2.66 points on MUTAG, 1.17 on PROTEINS, and 1.40 on IMDB-B, an average of 1.74 points. On IMDB-B it also yields the lowest variance across folds (± 1.55). Because the within-view baseline uses the same two encoders and the same number of views, these gains cannot be attributed to additional capacity or input information. They come from the cross-view alignment objective, which forces two representations of very different character, a topological image and a graph-structural image, to agree on class-level semantics. On PROTEINS, G2IMAGE (76.46) is on par with the best single view (76.73), but it improves substantially over the two-view baseline without alignment, consistent with alignment being the active ingredient.

Table 11: Ablation of the supervised contrastive objective (accuracy %, mean $\pm$ std). All rows use SupCon. Best per dataset in bold.

Configuration	**MUTAG**	**PROTEINS**	**IMDB-B**
TopoGrid (single view)	89.91 ± 9.24	$\mathbf{76.73} \pm 3.84$	75.30 ± 2.41
GraphGrid (single view)	89.88 ± 6.90	74.75 ± 2.08	74.00 ± 2.76
Two-view, within-view only	91.52 ± 7.16	75.29 ± 2.84	75.90 ± 2.07
G2IMAGE (cross-view)	$\mathbf{94.18} \pm 5.98$	76.46 ± 3.15	$\mathbf{77.30} \pm 1.55$

Comparison with graph-to-grid and graph-as-image methods. Table 12 compares G2IMAGE with methods that, like ours, transform graphs into CNN-compatible representations: Patchy-SAN [NAK16], DGCNN with SortPooling [ZCNC18], and GPGL [LHZ23]. These methods follow a similar 10-fold cross-validation protocol, so we report the results published in the respective papers. G2IMAGE attains the highest mean accuracy on all five datasets, with the largest margins on PTC-MR and IMDB-B. On PROTEINS, the margin over GPGL is negligible (76.46 vs. 76.44), so the two are effectively on par there.

The approaches also differ in how the grid-like input is built. Patchy-SAN selects an ordered sequence of nodes, extracts fixed-size local neighborhoods, normalizes them, and passes the resulting tensors to convolutional layers. DGCNN applies graph convolutions followed by SortPooling, which produces a fixed-length ordered node sequence for one-dimensional convolution. GPGL places vertices on an optimized two-dimensional grid before applying a CNN. These methods therefore depend on node ordering, learned node representations, or graph-layout optimization. In contrast, G2IMAGE directly constructs two complementary fixed-size graph-level tensors from permutation-equivariant intrinsic quantities: TOPOGRID captures multipersistence surfaces, and GRAPHGRID summarizes inter-group edge densities. It therefore requires neither a per-node ordering nor a graph-drawing layout, and it comes with permutation-invariance and resolution-controlled stability guarantees. The two views are then integrated through supervised cross-view contrastive alignment.

Table 12: Comparison with graph-to-grid and graph-as-image methods (accuracy %, mean ± std, 10-fold cross-validation). Baseline results are taken from the respective papers. Best per dataset in bold. "–" indicates that the result was not reported.

Model	MUTAG	PTC-MR	PROTEINS	IMDB-B	IMDB-M
Patchy-SAN	88.95 ± 4.37	62.29 ± 5.68	75.00 ± 2.51	71.00 ± 2.29	45.23 ± 2.84
DGCNN	85.83 ± 1.66	58.59 ± 2.47	75.54 ± 0.94	70.03 ± 0.86	47.83 ± 0.85
GPGL	92.48 ± 5.53	–	76.44 ± 1.56	74.90 ± 4.01	51.99 ± 2.36
G2IMAGE	$\mathbf{94.18} \pm 5.98$	$\mathbf{72.38} \pm 4.17$	$\mathbf{76.46} \pm 3.15$	$\mathbf{77.30} \pm 1.55$	$\mathbf{52.73} \pm 3.57$

Size- and density-controlled evaluation. To rule out size and density shortcuts, we evaluate G2IMAGE under distribution shifts in these attributes. The OGB results in Table 4 already provide one structure-controlled evaluation, since OGB uses scaffold splitting, which places structurally distinct molecules in different train and test subsets and is harder than random splitting. Under this protocol, G2IMAGE* attains the best average rank. In addition, we construct controlled splits on MUTAG, PTC-MR, PROTEINS, and IMDB-B. For each attribute, we sort the graphs and hold out the top 30% for testing, so that the train and test sets differ systematically in graph size or in density. Table 13 compares G2IMAGE with a plain graph-statistics baseline, GCN, and GAT. G2IMAGE achieves the highest mean accuracy on all eight splits, while GCN and GAT degrade sharply on several of them (for example, GAT reaches only about 41–42% on IMDB-B). The plain-statistics baseline is competitive only on MUTAG and IMDB-B, whose labels are known to correlate with size and degree, so a size or density model retains signal precisely where that signal is informative. On these two datasets the margins over it are small (1.4–2.2 points on MUTAG and 0.1–1.3 points on IMDB-B), and G2IMAGE is best mainly by being robust rather than by a large gap. On PTC-MR and PROTEINS, where simple statistics are far weaker, G2IMAGE leads the strongest baseline by 2.5–5.2 points. Together, these results indicate that the advantage of G2IMAGE does not reduce to a size or density shortcut.

G Resolution Tradeoffs and Limitations

G2IMAGE is a fixed-budget graph interface and makes an explicit compression tradeoff. It preserves mesoscopic topological and structural information in compact, permutation-invariant tensors, while discarding some fine-grained combinatorial detail. We summarize the main tradeoffs below.

Expressivity and non-injectivity. G2IMAGE is not a canonical graph form or graph isomorphism test. At fixed resolution, distinct graphs can share the same representation. TOPOGRID collides when all bifiltration cells have matching Betti-0, Betti-1, vertex-count, and edge-count summaries; GRAPHGRID collides when graphs share the same binned score profile and block-pooled inter-group

Table 13: Accuracy (%, mean ± std) under size- and density-controlled splits. Graphs are sorted by the controlled attribute and the top 30% are held out for testing. Best per row in bold.

Dataset	Control	Plain graph-stats	GCN	GAT	G2IMAGE
MUTAG	size	84.64 ± 0.87	68.21 ± 0.91	66.79 ± 2.90	$\mathbf{86.07} \pm 1.34$
	density	82.55 ± 8.90	65.82 ± 1.67	70.91 ± 2.57	$\mathbf{84.73} \pm 2.18$
PTC-MR	size	58.10 ± 1.20	56.00 ± 3.43	57.71 ± 2.45	$\mathbf{61.14} \pm 1.85$
	density	57.86 ± 2.35	57.67 ± 3.17	59.61 ± 1.69	$\mathbf{62.14} \pm 3.07$
PROTEINS	size	65.27 ± 1.45	69.82 ± 0.58	67.01 ± 0.91	$\mathbf{75.03} \pm 1.59$
	density	65.69 ± 9.80	69.88 ± 3.03	70.42 ± 5.20	$\mathbf{74.25} \pm 0.73$
IMDB-B	size	70.47 ± 0.40	57.93 ± 0.39	41.93 ± 0.57	$\mathbf{70.60} \pm 1.25$
	density	69.27 ± 0.44	58.48 ± 0.45	41.46 ± 0.57	$\mathbf{70.53} \pm 0.76$

edge densities. The two views reduce collisions relative to either view alone, since both topological summaries and density patterns must match. Highly regular or symmetry-rich graphs may remain challenging, while higher resolution, richer intrinsic scores, or domain-specific channels can improve discriminability within the same pipeline.

Relation to WL expressivity. G2IMAGE is not designed as a Weisfeiler–Leman isomorphism test. Its contribution is a different inductive bias: a fixed-budget, permutation-invariant, image-like representation that combines topological and structural summaries for reusable 2D encoders. The relevant question is not injectivity, but whether the retained mesoscopic information is predictive for downstream graph classification.

Scalability. The main preprocessing cost comes from TOPOGRID, which computes intrinsic scores and Betti summaries over an $M \times N$ grid. This cost is paid once and cached; training then uses fixed-size tensors. For larger graphs, exact spectral scores such as HKS may dominate, but approximate diffusion scores, randomized eigensolvers, or other permutation-equivariant scores can be substituted without changing the model. GRAPHGRID is lightweight, requiring only score binning and one edge-aggregation pass.

Domain adaptation. The base framework uses structural scores such as degree, HKS, PageRank, and k-core, and does not require attributes. In attribute-rich domains, these can be augmented with domain-aware scores. Our molecular instantiation, G2IMAGE-CHEM, uses atom and bond information while preserving the same imaging pipeline, encoders, and objective.

Evaluation scope. Our evaluation covers standard graph classification and OGB molecular prediction benchmarks, spanning molecular, biological, and social graphs. Extending G2IMAGE to larger long-range benchmarks, graph regression, and pretraining or transfer settings is a natural next step.

H Broader Impacts

G2IMAGE introduces a standardized, fixed-budget graph-to-image interface that lowers the barrier to applying mature 2D vision pipelines to graph problems in chemistry, biology, and materials science, with explicit resolution controls and stability guarantees that make the representation auditable and reproducible. The primary risk of misuse lies in applying the framework to sensitive relational data such as social or communication networks, where improved graph classification could enable user profiling or surveillance, and where degree- and PageRank-based scoring functions may amplify structural biases present in the underlying data. We recommend that practitioners evaluate subgroup performance when applying G2IMAGE to graphs representing people, avoid high-stakes deployment without domain-expert review of the chosen scoring functions and discretization, and document grid resolutions and normalization statistics to ensure reproducibility; additionally, the stability guarantees should be understood as bounds at fixed resolution rather than unconditional robustness claims, since perturbations that cause nodes to cross grid thresholds can produce discontinuous representation changes.

NeurIPS Paper Checklist

1. **Claims**

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions and scope: a fixed-budget two-view graph imaging interface, invariance/stability properties, cross-view alignment, and empirical validation on graph classification and molecular prediction benchmarks.

2. **Limitations**

   Answer: [Yes]

   Justification: The paper discusses non-injectivity, resolution dependence, scalability, expressivity, and failure cases in the main text and Appendix G.

3. **Theory assumptions and proofs**

   Answer: [Yes]

   Justification: The invariance and stability results state their assumptions explicitly, and the complete proofs are provided in Appendix A.

4. **Experimental result reproducibility**

   Answer: [Yes]

   Justification: The paper specifies datasets, preprocessing, train/test protocols, model components, optimization settings, evaluation metrics, and code availability.

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

   Answer: [Yes]

   Justification: The experiments use public TU and OGB datasets, and the paper provides an anonymized code link for reproducing the main results.

6. **Experimental setting/details**

   Answer: [Yes]

   Justification: Section 4 describes datasets, representations, training objective, optimizer, hyperparameters, validation protocol, and evaluation metrics.

7. **Experiment statistical significance**

   Answer: [Yes]

   Justification: Results are reported as mean $\pm$ standard deviation over repeated runs or folds, as described in the evaluation protocol.

8. **Experiments compute resources**

   Answer: [Yes]

   Justification: The paper reports complexity, representative runtime, hardware type, memory, and total compute.

9. **Code of ethics**

   Answer: [Yes]

   Justification: The work uses public benchmark datasets and does not involve human subjects, private data, deceptive systems, or high-risk deployment.

10. **Broader impacts**

    Answer: [Yes]

    Justification: The broader impacts are discussed in Appendix H.

11. **Safeguards**

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped personal datasets, or dual-use systems requiring special safeguards.

12. **Licenses for existing assets**

Answer: [Yes]

Justification: The paper cites the original TU and OGB benchmark sources and uses publicly available datasets and baselines.

13. **New assets**

    Answer: [Yes]

    Justification: The released asset is code for the proposed method; documentation and reproduction instructions are provided through the anonymized repository.

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or research with human participants.

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

    Answer: [N/A]

    Justification: No human-subject experiments are conducted, so IRB approval is not applicable.

16. **Declaration of LLM usage**

    Answer: [N/A]

    Justification: No large language models are used as part of the proposed method, experiments, data generation, or evaluation.