GRAPHVEC: CROSS-DOMAIN GRAPH VECTORIZATION FOR GRAPH-LEVEL REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning transferable graph representations across heterogeneous domains is challenging because graph datasets differ in topology, node-attribute semantics, feature dimensions, and attribute availability. We propose **GraphVec**, a language-model-free framework that maps heterogeneous graphs into fixed-dimensional representations for graph-level tasks. GraphVec constructs multi-scale global relational graphs and spectral embeddings to replace incomparable raw attributes with comparable relational features, and further aligns them through a density-maximization orthogonal transformation with provable convergence. A GIN–Graph Transformer backbone equipped with hierarchical reference distribution encoding captures node-distribution information beyond standard pooling. We further establish an end-to-end generalization bound connecting cross-domain contrastive pretraining to few-shot classification on unseen domains. Experiments on 13 datasets against more than 15 baselines show that GraphVec consistently performs strongly in cross-domain few-shot graph classification and graph clustering, while also yielding competitive representations for few-shot node classification.

1 INTRODUCTION

Graph data is a fundamental and widely prevalent form of structured data, representing entities as nodes and their relationships as edges. It plays a crucial role in diverse domains, including social networks, biological systems, citation networks, recommendation systems, and knowledge graphs. Given its ability to model complex relational patterns, graph data analysis has become a key focus in machine learning and data mining. In node-level tasks, the training set is usually a single but large graph, on which each node represents a sample. Node-level tasks include node embedding or representation (Grover & Leskovec, 2016; Cai et al., 2018), node classification (Kipf & Welling, 2017), node clustering (Wang et al., 2024a), link prediction (Martínez et al., 2016), etc. For example, node classification might involve categorizing users in a social network, while link prediction could be used to recommend new connections.

Graph-level tasks treat each graph as a sample and include graph comparison (Kobler et al., 2012), representation learning (Sun et al., 2020), classification (Xu et al., 2019), clustering (Cai et al., 2024), and generation (Liao et al., 2019). Applications include graph classification for molecular property prediction (Gilmer et al., 2017; Wang & Fan, 2024) and graph generation for creating drug-like molecules (Hoogeboom et al., 2022). Graph comparison measures similarity using kernels (Gärtner et al., 2003; Vishwanathan et al., 2010; Shervashidze et al., 2011), distances (Bunke, 1997; Zeng et al., 2009; Mémoli, 2011; Bento & Ioannidis, 2018), or deep learning methods (Sun & Fan, 2024). Our focus is graph-level representation learning, which encodes entire graphs into compact embeddings for downstream tasks (You et al., 2020; 2021; Sun et al., 2023b). Representative approaches range from substructure-based graph2vec (Narayanan et al., 2017) to mutual-information and contrastive methods such as InfoGraph, GraphCL, automated graph contrastive learning, and GCC (Sun et al., 2020; You et al., 2020; 2021; Qiu et al., 2020).

Reusing graph representation models across tasks and datasets can reduce repeated model selection and parameter tuning while making use of knowledge learned from other datasets. However, such reuse is difficult because graph structures and node attributes have different meanings across domains (Galkin et al., 2023). In molecular graphs, nodes and edges describe atoms and bonds, and node attributes encode chemical properties. In social networks, they describe users, their connections,

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

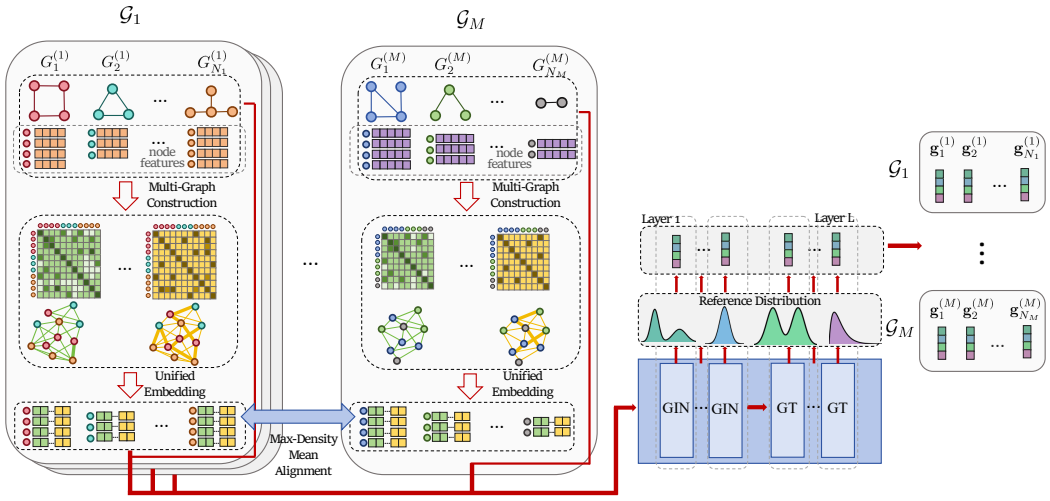

Figure 1: Flow-chart of GraphVec. $\mathcal{G}_1, \ldots, \mathcal{G}_M$ are M datasets from different domains. The model represents each graph $G_i^{(j)}$ as a single vector $\mathbf{g}_i^{(j)}$ that can be used in graph-level downstream tasks.

and user profiles. These differences in structure, feature semantics, and feature dimensions make it difficult to apply a single pretrained model across datasets.

Graph pretraining addresses this by learning representations adaptable to downstream tasks. Graph prompt learning, for example, adapts pretrained models by reducing the gap between pretraining and task-specific objectives (Sun et al., 2022; Fang et al., 2023; Liu et al., 2023b; Fu et al., 2025). For transfer across domains, however, adaptation must also accommodate differences in graph structure and input features. Research on graph foundation models (GFMs) further considers learning transferable representations from diverse datasets (Liu et al., 2025). Existing GFMs address this heterogeneity through different approaches. LLM-based methods use text to represent graph structures and attributes, allowing language models to process information from different datasets (Fatemi et al., 2023; Liu et al., 2023a; Kong et al., 2025; Tang et al., 2024). Non-LLM methods instead develop graph-based representations for cross-domain transfer (Sun et al., 2025; Yuan et al., 2025b), alongside advances in graph pretraining and adaptation (Qiu et al., 2020; Yu et al., 2025a; Fu et al., 2025).

Despite this progress, obtaining transferable graph-level representations remains challenging. Textual descriptions can lose fine-grained structural and node-attribute information (Yu et al., 2025a), and using LLMs adds substantial computational cost. Meanwhile, much of the work on pretrained graph models focuses on node-level tasks (Zhao et al., 2024b; Wang et al., 2025a), while transfer across datasets for graph-level tasks has received comparatively less attention (Yu et al., 2025c; Fu et al., 2025). These limitations motivate a model that can learn from heterogeneous graph datasets without relying on language models and produce graph embeddings useful in unseen domains.

We propose GraphVec for cross-domain graph-level representation learning, as illustrated in Figure 1. GraphVec constructs multi-scale relational graphs from node attributes and aligns their spectral features through density maximization, providing inputs for a shared GNN across datasets. To capture node-distribution information beyond pooling, it compares node representations at multiple GNN layers with shared learnable reference distributions. Our contributions are as follows.

- We propose GraphVec, a language-model-free framework for cross-domain graph vectorization without requiring comparable raw node attributes. Its key technical components include multi-scale relational spectral feature construction, density-maximization orthogonal alignment, and hierarchical reference distribution encoding.
- We provide theoretical guarantees for GraphVec by establishing the monotonic improvement and asymptotic stationarity of the density-maximization alignment algorithm, together with an end-to-end generalization bound connecting cross-domain contrastive pretraining to few-shot classification on unseen domains.

- Extensive experiments on cross-domain few-shot graph classification and graph clustering demonstrate the effectiveness of GraphVec against recent graph pretraining and prompt-learning methods, supported by comprehensive ablation, robustness, and scalability studies.

2 Related Work

Graph pretraining Many studies have explored the "pre-train and adaptation" paradigm for graph models, leveraging message-passing or transformer-based GNN backbones. These approaches typically employ contrastive or generative self-supervised learning for pretraining, followed by fine-tuning a subset of model parameters to adapt to downstream tasks or datasets (Liu et al., 2025). Contrastive methods, including GCC (Qiu et al., 2020), InfoGraph (Sun et al., 2020), DGI (Veličković et al., 2019), SimGRACE (Xia et al., 2022), and GCOPE (Zhao et al., 2024a), maximize agreement between augmented views to learn transferable representations, while generative methods (Hou et al., 2022; 2023) pre-train via graph reconstruction or property prediction. Recently, graph prompt tuning (Sun et al., 2022; Fang et al., 2023; Sun et al., 2023a; Liu et al., 2023b; Fu et al., 2025) has emerged to bridge the pretraining–downstream gap, and many recent GFMs adopt this paradigm (Yuan et al., 2025b; Yu et al., 2025b). However, these works mainly emphasize adaptation, leaving the problem of learning unified graph representations underexplored. Due to space limitations, we defer more discussion of **Graph pretraining** and the related work on **GFMs** to Appendix B.

3 Methodology

3.1 Cross-Domain Graph Vectorization: Problem Setup

First of all, the major notations used in this paper are shown in Table 4. Let $\mathcal{D} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_M\}$ be a union of M datasets of labeled graphs from M different domains, where $\mathcal{G}_j = \{(G_1^{(j)}, y_1^{(j)}), (G_2^{(j)}, y_2^{(j)}), \ldots, (G_{N_j}^{(j)}, y_{N_j}^{(j)})\}$. Here, each graph $G_i^{(j)}$ is denoted as $G_i^{(j)} = (\mathbf{A}_i^{(j)}, \mathbf{X}_i^{(j)}, y_i^{(j)})$, where $\mathbf{A}_i^{(j)} \in \mathbb{R}^{n_i^{(j)} \times n_i^{(j)}}$ denotes the adjacency matrix, $\mathbf{X}_i^{(j)} \in \mathbb{R}^{n_i^{(j)} \times d^{(j)}}$ denotes the node attribute matrix, $n_i^{(j)}$ denotes the number of nodes, $d^{(j)}$ denotes the number of attributes, and $y_i^{(j)}$ denotes the graph label. Our goal is to use $\mathcal{D}$ to train a cross-domain graph representation model, denoted as $F : \mathbb{G} \to \mathbb{R}^r$ to represent any graph from the space $\mathbb{G}$ as an r-dimensional vector that is useful in downstream tasks such as graph classification, where $\mathbb{G}$ denotes the set of all graphs in the form of $(\mathbf{A}, \mathbf{X})$. Therefore, F serves as a universal graph representation model. We need to address these challenges:

- **Attributes inconsistency** The node attributes of graphs from different domains are different and not comparable at all. Thus the node attributes in $\mathcal{D}$ cannot be fed into F directly.
- **Attributes absence** Many graph datasets do not contain node attributes, making them very different from graph datasets with node attributes. The heuristic method of constructing node attributes, such as using node degrees, does not comply with the semantic attributes of other datasets.
- **Information loss in pooling** Although there have been a few advanced graph pooling methods (Liu et al., 2022), converting nodes' embeddings into a single vector cannot fully utilize the information.

3.2 Domain-Agnostic Relational Features via Global Multi-Graphs

As mentioned, the node features of different domain graph datasets may vary significantly in semantics and dimensions. To capture domain-invariant features, we focus on the relationships among nodes across the entire dataset rather than the original features. The reason is that in many machine learning problems, using the relationships between samples or a graph constructed from the dataset can provide effective solutions. For instance, in spectral clustering (Ng et al., 2001), we use a similarity graph rather than the original features; in kernel support vector machine (Cortes & Vapnik, 1995), we can use a Gaussian kernel matrix, which is a similarity matrix of the data points.

Multi-Scale Global Relational Graphs For each graph dataset $\mathcal{G}_j$, $j \in [M]$, we propose to construct a similarity graph over all nodes in the dataset using a Gaussian kernel function, i.e.,

$$\mathbf{K}_{\lambda}^{(j)} = \left[\exp\left(-\frac{\|\mathbf{x}_u - \mathbf{x}_v\|_2^2}{2\lambda\mu^2}\right)\right]_{u,v=1}^{\bar{N}_j}, \quad \bar{N}_j = \sum_{i=1}^{N_j} n_i^{(j)} \tag{1}$$

where $\mathbf{x}_i$ is the i-th row of $\mathbf{X} := \big|\big|_{i=1}^{N_j} \mathbf{X}_i^{(j)}$ (vertical concatenation), μ is the mean of the pairwise distances between all nodes in the dataset, and λ controls the bandwidth of the kernel. This setting ensures *translation, rotation, and scaling invariance*, which is important to extract comparable features across diverse datasets. $\mathbf{K}_{\lambda}^{(j)}$ is the adjacency matrix of this global graph of the nodes in $\mathcal{G}_j$. Note that a single $\mathbf{K}_{\lambda}^{(j)}$ exploits partial information of the node attributes of $\mathcal{G}_j$ and the optimal setting of λ remains an open problem. Therefore, we use a number of different values for λ, e.g. $\lambda_1, \lambda_2, \ldots, \lambda_Q$, to construct **multiple global graphs** for the nodes in $\mathcal{G}_j$:

$$\boldsymbol{\mathcal{K}}^{(j)} := \left\{\mathbf{K}_{\lambda_1}^{(j)}, \mathbf{K}_{\lambda_2}^{(j)}, \ldots, \mathbf{K}_{\lambda_Q}^{(j)}\right\}, \quad j \in [M] \tag{2}$$

Note that the diversity of $\boldsymbol{\mathcal{K}}^{(j)}$ can be further enhanced if more kernel families, e.g., $k(\mathbf{x}_u, \mathbf{x}_v) = \exp(-\alpha\|\mathbf{x}_u - \mathbf{x}_v\|_1)$, are considered.

Multi-Scale Relational Spectral Embedding For $\mathbf{K}_{\lambda_q}^{(j)}$, we compute $\bar{d}$-dimensional node embeddings using singular value decomposition (SVD):

$$\mathbf{Z}_{\lambda_q}^{(j)} = \mathbf{U}_{\bar{d}}\boldsymbol{\Sigma}_{\bar{d}}^{1/2}, \quad \mathbf{K}_{\lambda_q}^{(j)} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\top} \tag{3}$$

where $\mathbf{U}_{\bar{d}} \in \mathbb{R}^{\bar{N}_j \times \bar{d}}$ is composed of the first $\bar{d}$ columns of $\mathbf{U}$ and $\boldsymbol{\Sigma}_{\bar{d}}$ is a diagonal matrix consisting of the first (largest) $\bar{d}$ singular values. Then the final node feature matrix is obtained by concatenating embeddings from all scales, i.e.,

$$\mathbf{Z}^{(j)} = \left[\mathbf{Z}_{\lambda_1}^{(j)}, \ldots, \mathbf{Z}_{\lambda_Q}^{(j)}\right] = \begin{bmatrix} \mathbf{Z}_1^{(j)} \\ \vdots \\ \mathbf{Z}_{N_j}^{(j)} \end{bmatrix} \in \mathbb{R}^{\sum_{i=1}^{N_j} n_i^{(j)} \times Q\bar{d}}, \tag{4}$$

where $j \in [M]$. For datasets without node attributes, we generate node attributes using the truncated SVD of the self-looped adjacency matrix, i.e.,

$$\mathbf{X}_i^{(j)} = \mathrm{SVD}\left(\mathbf{A}_i^{(j)} + \mathbf{I}_{n_i^{(j)}}\right) \in \mathbb{R}^{n_i \times \bar{d}} \tag{5}$$

where $\mathrm{SVD}(\cdot)$ returns the singular vectors corresponding to the top-$\bar{d}$ singular values, similar to (3). Then we apply (1), (2), (3), and (4) to $\mathbf{X}_i^{(j)}$ to generate unified node embeddings. For large datasets, the Nyström approximation (Williams & Seeger, 2000) can be employed to accelerate the computation of the kernel matrix and SVD.

3.3 Convergent Density-Maximization Alignment for Spectral Features

In SVD, individual singular vectors have arbitrary signs (Bro et al., 2008). This sign ambiguity may make the embeddings of two similar graphs very different, leading to significant difficulties in both the training and testing stages. Moreover, if two singular values are the same, the order of the corresponding singular vectors cannot be determined, which further increase the difficulty in learning. In machine learning, to ensure learnability and generalization, we require that the training samples and the testing samples are from the same distribution, or at least, their means are similar. Therefore, we proposed to align the mean embeddings of different graphs via maximizing the density.

Specifically, for each scale λ_q, we compute the spectral mean $\boldsymbol{\mu}_j^{(\lambda_q)} = \bar{N}_j^{-1}(\mathbf{Z}_{\lambda_q}^{(j)})^{\top}\mathbf{1}_{\bar{N}_j}$ for $j \in [M]$. We drop the superscript (λ_q) below when considering one fixed scale. For each dataset j, we introduce an orthonormal matrix $\mathbf{R}_j \in \mathbb{R}^{\bar{d}\times\bar{d}}$. Right-multiplying the node features by $\mathbf{R}_j^{\top}$ preserves their within-domain Gram matrix, while transforming the spectral mean to $\mathbf{R}_j\boldsymbol{\mu}_j$. We align these

Algorithm 1 Max-Density Mean Alignment

Input: $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_M$; $\gamma_{\mathrm{A}} > 0$; $\eta > 0$; T.
Initialization: $\mathbf{R}_j^{(0)} = \mathbf{I}_{\bar{d}}, \forall j \in [M]$
$w_{ij} = \exp(-\gamma_{\mathrm{A}}(\|\boldsymbol{\mu}_i\|^2 + \|\boldsymbol{\mu}_j\|^2))$, $(i,j) \in [M] \times [M]$
for $t = 1$ to T **do**
  $k_{ij}^{(t-1)} = \exp(2\gamma_{\mathrm{A}} \langle \mathbf{R}_i^{(t-1)} \boldsymbol{\mu}_i, \mathbf{R}_j^{(t-1)} \boldsymbol{\mu}_j \rangle)$, $(i,j) \in [M] \times [M]$
  for $i = 1$ to M **do**
    $\mathbf{H}_i^{(t)} = \sum_j w_{ij} k_{ij}^{(t-1)} \mathbf{R}_j^{(t-1)} \boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top + \eta \mathbf{R}_i^{(t-1)}$
    SVD: $\mathbf{H}_i^{(t)} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^\top$
    $\mathbf{R}_i^{(t)} = \mathbf{U}_i \mathbf{V}_i^\top$
  end for
end for
Output: $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_M$

Algorithm 2 pretraining model

Input: Train datasets: $\mathcal{D} = \{\mathcal{G}_j\}_{j=1}^M$
Compute $\{\mathbf{Z}^{(j)}\}_{j=1}^M$ using (3) and (4)
$\{\mathbf{R}_j^{\lambda_q}\}_{j=1}^M \leftarrow$ Algorithm 1
$\mathbf{Z}_{\lambda_q}^{(j)} \leftarrow \mathbf{Z}_{\lambda_q}^{(j)} {\mathbf{R}_j^{(\lambda_q)}}^\top$, $q \in [Q]$, $j \in [M]$
repeat
  for $j = 1$ to M **do**
    for $i = 1$ to S **do**
      Sample $\{\mathbf{G}_i^{(j)} = (\mathbf{Z}_i^{(j)}, \mathbf{A}_i^{(j)})\}_{i \in \mathcal{B}}$
      $\mathbf{g}_i^{(j)} \leftarrow F_{\mathcal{W},\mathcal{V},\gamma}(\mathbf{A}_i^{(j)}, \mathbf{Z}_i^{(j)}), i \in \mathcal{B}$
      $(\mathcal{W}, \mathcal{V}) \leftarrow (\mathcal{W}, \mathcal{V}) - \alpha_1 \nabla_{(\mathcal{W},\mathcal{V})} \mathcal{L}$
      $\gamma \leftarrow \gamma - \alpha_2 \nabla_\gamma \mathcal{L}$
    end for
  end for
until Convergence conditions are met
Output: Pretrained model $F_{\mathcal{W},\mathcal{V},\gamma}$

means by maximizing their density. The density of each mean vector can be calculated by the kernel density estimation (Parzen, 1962):

$$\hat{p}(\boldsymbol{\mu}) = \frac{1}{M} \sum_{j=1}^{M} \frac{1}{(2\pi h)^{\bar{d}/2}} \exp\left(-\frac{\|\boldsymbol{\mu} - \boldsymbol{\mu}_j\|^2}{2h}\right) \tag{6}$$

where we use the Gaussian kernel with hyperparameter h. Let $\mathcal{R}$ be the set of all orthonormal matrices of size $\bar{d} \times \bar{d}$, i.e., $\mathcal{R} = \{\mathbf{R} \in \mathbb{R}^{\bar{d} \times \bar{d}} : \mathbf{R}^\top \mathbf{R} = \mathbf{I}_{\bar{d}}\}$. Then we maximize the total density of the M mean vectors:

$$\underset{\mathbf{R}_j \in \mathcal{R}, j \in [M]}{\text{maximize}} \ \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{1}{(2\pi h)^{\bar{d}/2}} \exp\left(-\frac{\|\mathbf{R}_i \boldsymbol{\mu}_i - \mathbf{R}_j \boldsymbol{\mu}_j\|^2}{2h}\right) \tag{7}$$

Letting $\gamma_{\mathrm{A}} = \frac{1}{2h}$, where the subscript distinguishes the alignment bandwidth from the learnable RBF parameter γ in the reference-distribution module, (7) is equivalent to

$$\underset{\mathbf{R}_j \in \mathcal{R}, j \in [M]}{\text{maximize}} \ \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \exp\left(-\gamma_{\mathrm{A}} \|\mathbf{R}_i \boldsymbol{\mu}_i - \mathbf{R}_j \boldsymbol{\mu}_j\|^2\right) \triangleq \mathcal{L}\left(\{\mathbf{R}_j\}_{j=1}^M\right). \tag{8}$$

For the convergence analysis, let $L := M\mathcal{L}$ denote the equivalent unnormalized objective; L and $\mathcal{L}$ have the same maximizers and the same monotonicity.

The optimization is non-trivial due to the orthonormal constraints and the exponential functions. We propose an efficient algorithm in Algorithm 1. The detailed derivation for the algorithm is introduced in Appendix C. Theorem 3.1 provides a convergence guarantee for the optimization.

Theorem 3.1 (Monotonic convergence of Algorithm 1)**.** *Let $\nu := \max_{j \in [M]} \|\boldsymbol{\mu}_j\|_2$ and suppose $\eta > 4\gamma_{\mathrm{A}} \nu^4 M^{3/2} + \nu^2 M$. Then $\{L(\{\mathbf{R}_j^{(t)}\}_{j=1}^M)\}_{t \geq 0}$ is monotonically non-decreasing and convergent. Moreover, $\sum_{j=1}^M \|\mathbf{R}_j^{(t)} - \mathbf{R}_j^{(t-1)}\|_F^2 \to 0$ as $t \to \infty$. Every accumulation point of $\{\mathbf{R}_j^{(t)}\}_{j=1}^M$ is a stationary point of the constrained maximization problem.*

Once $\mathbf{R}_1, \ldots, \mathbf{R}_M$ are optimized, we modify the embeddings of the $M \times Q$ global graphs as

$$\mathbf{Z}_{\lambda_q}^{(j)} \leftarrow \mathbf{Z}_{\lambda_q}^{(j)} {\mathbf{R}_j^{(\lambda_q)}}^\top, \quad q \in [Q], \ j \in [M], \tag{9}$$

where $\mathbf{R}_j^{(\lambda_q)}$ denotes the transformation obtained for dataset j at the q-th kernel scale. Using (4) and (9), we obtain the modified embeddings $\mathbf{Z}_1^{(j)}, \ldots, \mathbf{Z}_{N_j}^{(j)}$, $j \in [M]$. Note that Algorithm 1 can also be applied to the generated attributes by (5) of graphs without inherent node attributes.

3.4 LOCAL–GLOBAL NODE ENCODING WITH GIN AND GRAPH TRANSFORMER

To design a universal graph representation model F, we incorporate two main components: a GIN module f (Xu et al., 2019) and a graph transformer (GT) (Rampášek et al., 2022) module g. We build a GIN encoder followed by a graph transformer encoder $g_\psi \circ f_\theta(\cdot)$, where θ and ψ are the parameters. The GIN encoder specializes in learning local representations of the structure of a node's immediate neighborhood, while the transformer computes all pairwise node interactions, enabling global reasoning through attention mechanisms. The node representations of $G_i^{(j)}$ is

$$\mathbf{H}_i^{(j)} = g_\psi \circ f_\theta\left(\mathbf{A}_i^{(j)}, \mathbf{Z}_i^{(j)}\right) \Big\| f_\theta\left(\mathbf{A}_i^{(j)}, \mathbf{Z}_i^{(j)}\right) \tag{10}$$

where $i \in [N_j]$ and $j \in [M]$. For convenience, we denote $\mathcal{W} = \{\psi, \theta\}$.

3.5 HIERARCHICAL REFERENCE DISTRIBUTION ENCODING

Standard graph-level pooling may discard distributional information in node embeddings. To better preserve such information, we introduce a *layer-wise reference distribution encoding mechanism*, where learnable reference distributions are attached to multiple GIN and GT layers. Each graph is represented by its similarities to these reference distributions, capturing node-distribution patterns at different structural ranges and abstraction levels.

Specifically, suppose for a certain layer $l \in [L]$ in the backbone, we have R reference discrete distributions $\{\mathbf{V}_1^{(l)}, \mathbf{V}_2^{(l)} \ldots, \mathbf{V}_R^{(l)}\} \triangleq \mathcal{V}^{(l)}$, each $\mathbf{V}_b^{(l)} \in \mathbb{R}^{m \times d}$ is a learnable set of m reference nodes and can be interpreted as a discrete reference distribution in the latent space. To obtain the graph representations from node embedding matrix $\mathbf{H}_i^{(j,l)}$, we measure the similarity between the graph $G_i^{(j)}$ and the reference distributions $\{\mathbf{V}_b^{(l)}\}_{b=1}^R$. Letting ξ be a similarity measure between two distributions, the similarity between the graph $G_i^{(j)}$ and the reference distribution $\mathbf{V}_b^{(l)}$ is

$$s_i^{(j,l)} = \xi\left(\mathbf{H}_i^{(j,l)}, \mathbf{V}_b^{(l)}\right), \quad b \in [R] \tag{11}$$

We let ξ be the negative kernelized Maximum Mean Discrepancy (MMD) (Gretton et al., 2012):

$$s_{i,b}^{(j,l)} = -\text{MMD}\left(\mathbf{H}_i^{(j,l)}, \mathbf{V}_b^{(l)}\right), \tag{12}$$

where we use the Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2\right)$ with a learnable parameter γ. The expanded expression is given in Appendix D.1. We apply the RD module after every layer in both the GT and GIN backbones, allowing the final graph embedding to capture hierarchical information from neighbors at different ranges. The final graph embedding $\mathbf{g}_i^{(j)} \in \mathbb{R}^r$ of graph i in dataset j combines the similarity vector $\mathbf{s}_i^{(j)} = \Big\|_{l \in [L]} \mathbf{s}_i^{(j,l)}$ from different layers with a readout vector $\mathbf{p}_i^{(j)}$, i.e.,

$$\mathbf{g}_i^{(j)} = \mathbf{s}_i^{(j)} \big\| \mathbf{p}_i^{(j)}, \quad i \in [N_j], j \in [M] \tag{13}$$

where $\mathbf{p}_i^{(j)} = \text{READOUT}(\mathbf{H}_i^{(j)})$ is obtained using a graph-level pooling operation.

3.6 SUPERVISED AND UNSUPERVISED CROSS-DATASET PRETRAINING

Different datasets $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_M$ may contain varying numbers of classes. To unify training across datasets and avoid changing classifiers during pretraining, we use a fixed-size mini-batch contrastive objective. For dataset j, let $\mathcal{B}_j$ contain B graph views and let $C_j(i) = \{u \in \mathcal{B}_j \setminus \{i\} : y_u = y_i\}$ be the supervised positive set. Define $\mathcal{I}_j^{\text{SCL}} = \{i \in \mathcal{B}_j : |C_j(i)| > 0\}$. The supervised contrastive loss (SCL) is

$$\mathcal{L}_{\text{SCL}}(\mathcal{W}, \mathcal{V}, \gamma) = -\frac{1}{M} \sum_{j=1}^{M} \frac{1}{\max\{1, |\mathcal{I}_j^{\text{SCL}}|\}} \sum_{i \in \mathcal{I}_j^{\text{SCL}}} \frac{1}{|C_j(i)|} \sum_{u \in C_j(i)} \log p_{iu}^{(j)}, \tag{14}$$

where $p_{iu}^{(j)} = \exp(\zeta(\mathbf{g}_i^{(j)}, \mathbf{g}_u^{(j)})) / \sum_{v \in \mathcal{B}_j, v \neq i} \exp(\zeta(\mathbf{g}_i^{(j)}, \mathbf{g}_v^{(j)}))$, $\mathbf{g}_i^{(j)} = F_{\mathcal{W},\mathcal{V},\gamma}(\mathbf{A}_i^{(j)}, \mathbf{Z}_i^{(j)})$, and $\zeta(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v} / (\varrho \|\mathbf{u}\| \|\mathbf{v}\|)$. If $\mathcal{I}_j^{\text{SCL}}$ is empty, its batch contribution is defined as zero. This objective pulls same-class graphs together and pushes different-class graphs apart; see Algorithm 2.

GraphVec can be pretrained without labels. Let $P_j(i)$ be the designated augmented views of the same original graph and $\mathcal{I}_j^{\mathrm{UCL}} = \{i \in \mathcal{B}_j : |P_j(i)| > 0\}$. The unsupervised contrastive loss (UCL) is

$$\mathcal{L}_{\mathrm{UCL}}(\mathcal{W}, \mathcal{V}, \gamma) = -\frac{1}{M}\sum_{j=1}^{M} \frac{1}{\max\{1, |\mathcal{I}_j^{\mathrm{UCL}}|\}} \sum_{i \in \mathcal{I}_j^{\mathrm{UCL}}} \frac{1}{|P_j(i)|} \sum_{u \in P_j(i)} \log p_{iu}^{(j)}. \tag{15}$$

The same mini-batch denominator $p_{iu}^{(j)}$ is used in both modes. The fixed empty-positive convention is also the one analyzed in Appendix I.1.1.

3.7 Inductive Graph Embedding at Test Time

For each test graph, we compute the cross-kernel between its nodes and the target support nodes. Using the spectral bases and alignment matrices computed from the support set, we obtain

$$\widetilde{\mathbf{Z}}_{i,\lambda_q}^{\text{test}} = \mathbf{K}_{i,\lambda_q}^{\text{test,train}} \mathbf{V}_{q,\bar{d}} \boldsymbol{\Sigma}_{q,\bar{d}}^{-1/2} (\mathbf{R}_q^{\text{train}})^{\top}. \tag{16}$$

The features from all scales are concatenated into $\widetilde{\mathbf{Z}}_i^{\text{test}}$ and fed to the pretrained encoder:

$$\mathbf{g}_i^{\text{test}} = F_{\mathcal{W},\mathcal{V},\gamma}\left(\mathbf{A}_i^{\text{test}}, \widetilde{\mathbf{Z}}_i^{\text{test}}\right). \tag{17}$$

The support bases and alignment matrices remain fixed during inference. Each test graph can therefore be embedded individually, without access to other test graphs. Details are in Appendix E.2.

3.8 Generalization Analysis of GraphVec

We analyze the complete inductive pipeline of GraphVec: contrastive pretraining on M source datasets, alignment using only the training split S of an unseen target dataset, and few-shot training of a linear classifier $\widehat{\mathbf{W}}_{\mathrm{c}}$. Let $a \in \{\mathrm{SCL}, \mathrm{UCL}\}$ denote the supervised or unsupervised pretraining mode, B the number of contrastive views, and $L_B = \log(B-1) + 2/\varrho$. For scale λ_q, let $\mathbf{m}_j^{(q)}$ and $\mathbf{m}_T^{(q)}$ denote the aligned source and target spectral means and define $\mathfrak{a}_q = \frac{1}{M}\sum_{j=1}^{M}\left[1 - \exp\left(-\gamma_{\mathrm{A}}\|\mathbf{m}_T^{(q)} - \mathbf{m}_j^{(q)}\|_2^2\right)\right]$, where $\mathbf{m}_T^{(q)}$ is computed from the target training split only. For the classifier learned from S, define its population cross-entropy risk on a fresh target graph by $\mathcal{R}_T^{\mathrm{CE}} := \mathbb{E}_{(G,y)\sim\mathbb{P}_T}\left[\mathrm{CE}\left(\widehat{\mathbf{W}}_{\mathrm{c}}[\mathbf{g}_S(G); 1], y\right)\right]$, where $\mathbf{g}_S(G)$ is obtained using the spectral map and alignment fitted from S. We abbreviate $L_{\mathrm{shift}} := L_{\mathrm{shift}}(\widehat{\boldsymbol{\omega}})$ for the deployed encoder.

Theorem 3.2 (Generalization of GraphVec on an unseen target domain)**.** *Let $\delta \in (0,1)$ and allocate failure probability $\delta/2$ to source pretraining and $\delta/2$ to target-head estimation. Under the assumptions in Appendix H, with probability at least $1-\delta$,*

$$\mathcal{R}_T^{\mathrm{CE}} \le \frac{1}{\kappa_T(B,\boldsymbol{\pi})} \min\left\{ L_B,\ \mathcal{C}_{\mathrm{pre}}^{a}(\delta/2) + \frac{8L_{\mathrm{shift}}}{\varrho}\sqrt{\frac{\sum_{q=1}^{Q}\mathfrak{a}_q}{1-e^{-4\gamma_{\mathrm{A}}}}} + \Delta_{\mathrm{res}}^{a} + \mathbf{1}\{a = \mathrm{UCL}\}\epsilon_{\mathrm{aug}\to\mathrm{sem,T}} \right\} + \epsilon_{\mathrm{head}}(\delta/2), \tag{18}$$

where $\kappa_T(B,\boldsymbol{\pi}) > 0$ quantifies class coverage in a random target mini-batch, $\mathcal{C}_{\mathrm{pre}}^{a}(\delta/2)$ is the source contrastive certificate defined in Appendix H, $\Delta_{\mathrm{res}}^{a}$ is the residual task-relevant target–source discrepancy, and $\epsilon_{\mathrm{head}}(\delta/2)$ is the target-head term defined in (68). For UCL, $\epsilon_{\mathrm{aug}\to\mathrm{sem,T}}$ accounts for the difference between augmentation-defined and semantic positive pairs. Furthermore, $\mathbb{P}_T(\widehat{y}(G) \neq y) \le \min\{1, \mathcal{R}_T^{\mathrm{CE}}/\log 2\}$.

The proof is nontrivial for four reasons: mini-batch contrastive terms cannot be treated as independent graph pairs; UCL augmentations are dependent views of the same graph; the kernel/SVD/alignment map is itself learned from data; and the contrastive objective must be linked to a separately trained target classifier. The proof, in Appendix H, addresses these issues with a block PAC–Bayes argument, explicit corrections for learned preprocessing, an alignment-sensitive transfer inequality, and a random-batch contrastive-to-classification bridge.

The theorem separates three sources of generalization error. First, larger source sample sizes tighten the explicit statistical component of $\mathcal{C}_{\mathrm{pre}}^{a}$ through S_{eff}. Second, the proposed density-maximization

Table 1: 50-shot graph classification performance comparison with different pretrained models. We color the **best** and **second best** models. The compared numbers of in-domain experiments are from EdgePrompt (Fu et al., 2025). We only demonstrate the most competitive results reported in EdgePrompt here due to space limitations. Full comparison can be found in Appendix F.2.

pretraining	Tuning Methods	ENZYMES	DD	NCI1	NCI109	Mutagenicity	Average
in-dataset	Classifier Only (You et al., 2020)	$27.07_{\pm 1.04}$	$61.77_{\pm 2.40}$	$61.27_{\pm 3.64}$	$62.12_{\pm 1.10}$	$67.36_{\pm 0.71}$	55.92
	GraphPrompt (Liu et al., 2023b)	$26.87_{\pm 1.47}$	$62.58_{\pm 1.84}$	$62.45_{\pm 1.52}$	$62.41_{\pm 0.69}$	$68.03_{\pm 0.78}$	56.47
	ALL-in-one (Sun et al., 2023a)	$25.73_{\pm 1.18}$	$65.16_{\pm 1.47}$	$58.52_{\pm 1.59}$	$62.01_{\pm 0.66}$	$64.43_{\pm 1.00}$	55.17
	GPF (Fang et al., 2023)	$28.53_{\pm 1.76}$	$65.64_{\pm 0.70}$	$61.45_{\pm 3.13}$	$61.90_{\pm 1.26}$	$67.19_{\pm 0.74}$	56.94
	GPF-plus (Fang et al., 2023)	$27.33_{\pm 2.01}$	$67.20_{\pm 1.56}$	$61.61_{\pm 2.89}$	$62.84_{\pm 0.23}$	$67.69_{\pm 0.64}$	57.33
	EdgePrompt (Fu et al., 2025)	$29.33_{\pm 2.30}$	$63.97_{\pm 2.14}$	$62.02_{\pm 3.02}$	$62.02_{\pm 1.03}$	$67.55_{\pm 0.85}$	56.98
	EdgePrompt+ (Fu et al., 2025)	$32.67_{\pm 2.53}$	$67.72_{\pm 1.62}$	**$67.07_{\pm 1.96}$**	**$66.53_{\pm 1.30}$**	$68.31_{\pm 1.36}$	60.46
	GraphCL (You et al., 2020)	$30.50_{\pm 1.16}$	$62.89_{\pm 2.19}$	$62.49_{\pm 1.95}$	$61.68_{\pm 0.93}$	$66.62_{\pm 1.87}$	56.84
	GeMax (Sun et al., 2024)	$31.20_{\pm 5.92}$	$56.79_{\pm 1.12}$	$59.62_{\pm 1.41}$	$59.55_{\pm 0.85}$	$65.18_{\pm 1.98}$	54.46
cross-dataset	GCN (Kipf & Welling, 2017)	$43.33_{\pm 1.05}$	$65.84_{\pm 2.77}$	$61.36_{\pm 2.00}$	$62.17_{\pm 0.66}$	$60.46_{\pm 1.75}$	58.63
	BRIDGE (Yuan et al., 2025b)	$36.67_{\pm 5.96}$	$64.95_{\pm 3.38}$	$63.50_{\pm 2.27}$	$61.78_{\pm 1.63}$	$65.12_{\pm 2.83}$	58.40
	GFT (Wang et al., 2024b)	$34.61_{\pm 3.12}$	$56.00_{\pm 1.77}$	$59.16_{\pm 6.25}$	$60.50_{\pm 2.71}$	$67.82_{\pm 3.18}$	55.61
	RiemannGFM (Sun et al., 2025)	$34.27_{\pm 1.72}$	$68.74_{\pm 1.31}$	$55.10_{\pm 2.24}$	$59.86_{\pm 1.30}$	$62.56_{\pm 4.04}$	56.11
GraphVec		**$51.00_{\pm 3.22}$**	**$75.94_{\pm 2.70}$**	**$67.32_{\pm 1.51}$**	**$67.90_{\pm 1.67}$**	**$68.57_{\pm 1.62}$**	**66.14**
Unsupervised GraphVec		**$48.33_{\pm 2.36}$**	**$74.02_{\pm 1.26}$**	$66.11_{\pm 2.30}$	$64.34_{\pm 2.38}$	**$68.38_{\pm 2.88}$**	**64.23**

alignment directly reduces the term involving $\sum_q \mathfrak{a}_q$ while preserving within-domain spectral Gram geometry; domain differences not explained by this alignment remain in Δ^a_{res}. Third, more labeled target graphs reduce the explicit finite-sample component of ϵ_{head} through $N^{\text{c}}_{\text{eff}}$. Importantly, architecture dependence enters through localized complexity and sensitivity quantities rather than a worst-case product of layerwise norms. The complete parameter-dependent form, including $Q, \bar{d}, \lambda_q, L, R, m, d, \gamma, \gamma_{\text{A}}, \varrho$, GIN/GT complexity, and target sample size, is given in Appendix H.

4 Experiments

4.1 Few-Shot Graph Classification

Datasets and Baselines Following (Fu et al., 2025), we use five datasets from TUDataset (Morris et al., 2020), including ENZYMES, DD, NCI1, NCI109, and Mutagencity, to conduct few-shot graph classification experiments. We evaluate our methods against baselines under two distinct settings:

1) In-dataset setting: In this setting, the training and testing sets are partitioned within the same dataset, and the model is fine-tuned using few-shot samples. We employ SimGRACE (Xia et al., 2022), the most competitive pretraining strategy reported in EdgePrompt for pretraining and adopt seven different tuning mechanisms, including prompt-tuning methods such as GraphPrompt, All-In-One (Sun et al., 2023a), GPF (Fang et al., 2023), and GPF-plus (Fang et al., 2023) as well as standard classifier training. We also compare GraphVec with 2 graph representation methods, including the classical method GraphCL (You et al., 2020) and the recent method GeMax (Sun et al., 2024).

2) Cross-dataset setting: This setting involves fine-tuning and testing on datasets that were unseen during the pretraining phase. Specifically, we adopt a leave-one-out strategy where each of the five datasets serves as the downstream target for testing, while the remaining four datasets are leveraged for pretraining. In cross domain setting we compared GraphVec with 3 recent strong GFM baselines, GFT (Wang et al., 2024b), BRIDGE (Yuan et al., 2025b), and RiemannGFM (Sun et al., 2025), and 1 classic GNN (Kipf & Welling, 2017). GraphVec is pretrained and evaluated under this setting.

To further evaluate the transferability of GraphVec in different domains, we conduct more experiments on 4 social network datasets, including COLLAB, REDDIT-BINARY, IMDB-BINARY, IMDB-MULTI and 3 computer vision datasets including Letter-med, COIL-RAG and Cuneiform using GraphVec pretrained on 5 bio-chemical datasets mentioned above. Our GraphVec is compared with 2 graph prompt based methods: ProNoG (Yu et al., 2025b) and EdgePrompt+ (Fu et al., 2025), 2 LLM-based GFMs: OFA (Liu et al., 2023a) and GOFA (Kong et al., 2025), and 4 non-LLM-based GFMs: GFT (Wang et al., 2024b), SAMGPT (Yu et al., 2025a), BRIDGE (Yuan et al., 2025b), and RiemannGFM (Sun et al., 2025). For a fairer graph-level comparison, we additionally equip GFT and RiemannGFM with DiffPool (Ying et al., 2018) as an advanced pooling mechanism. More details about the settings and datasets can be found in Appendix E.3 and E.1.

Table 2: Few-shot graph classification on social network and computer vision datasets. 5-shot and 1-shot settings are used on COIL-RAG and Cuneiform, due to insufficient samples in some classes.

Dataset	COLLAB 50-shot	REDDIT-B 50-shot	IMDB-B 50-shot	IMDB-M 50-shot	Letter-med 50-shot	COIL-RAG 5-shot	Cuneiform 1-shot
ProNoG (Yu et al., 2025b)	$46.88_{\pm 3.14}$	$74.33_{\pm 2.05}$	$60.8_{\pm 5.19}$	$40.53_{\pm 0.66}$	$56.98_{\pm 5.83}$	$34.97_{\pm 7.62}$	$10.00_{\pm 7.92}$
EdgePrompt+(Fu et al., 2025)	$\mathbf{68.76}_{\pm 1.60}$	$74.60_{\pm 1.60}$	$\mathbf{71.17}_{\pm 1.07}$	$46.60_{\pm 0.50}$	$74.66_{\pm 1.69}$	$5.60_{\pm 0.21}$	$18.22_{\pm 0.72}$
OFA (Liu et al., 2023a)	$33.07_{\pm 0.43}$	$50.02_{\pm 0.99}$	$51.06_{\pm 0.67}$	$34.33_{\pm 0.50}$	$20.19_{\pm 0.37}$	$20.48_{\pm 0.79}$	$19.94_{\pm 0.67}$
GOFA (Kong et al., 2025)	$38.31_{\pm 1.02}$	OOM	$49.49_{\pm 0.84}$	$34.67_{\pm 0.61}$	$11.11_{\pm 0.25}$	$3.84_{\pm 0.18}$	$6.25_{\pm 0.31}$
SAMGPT (Yu et al., 2025a)	$67.20_{\pm 0.91}$	$55.77_{\pm 0.90}$	$49.50_{\pm 0.50}$	$46.00_{\pm 6.30}$	$56.90_{\pm 0.74}$	$57.34_{\pm 1.66}$	$52.00_{\pm 6.96}$
GFT+DiffPool (Wang et al., 2024b)	$63.44_{\pm 1.35}$	OOM	$65.83_{\pm 1.12}$	$46.30_{\pm 0.94}$	$37.83_{\pm 0.71}$	$7.68_{\pm 0.42}$	$10.34_{\pm 0.39}$
RiemannGFM+DiffPool (Sun et al., 2025)	$40.45_{\pm 0.88}$	OOM	$55.53_{\pm 0.72}$	$34.31_{\pm 0.57}$	$45.76_{\pm 0.63}$	$12.45_{\pm 0.55}$	$11.05_{\pm 0.48}$
BRIDGE (Yuan et al., 2025b)	$54.52_{\pm 3.73}$	$57.80_{\pm 8.79}$	$50.20_{\pm 6.27}$	$36.60_{\pm 5.48}$	$35.40_{\pm 2.09}$	$27.02_{\pm 6.90}$	$23.07_{\pm 9.42}$
GraphVec	$\underline{68.09}_{\pm 2.99}$	$\mathbf{81.52}_{\pm 1.50}$	$\underline{68.39}_{\pm 4.06}$	$\mathbf{46.70}_{\pm 0.99}$	$\mathbf{85.60}_{\pm 1.44}$	$\mathbf{74.20}_{\pm 0.77}$	$\mathbf{55.86}_{\pm 8.15}$
Unsupervised GraphVec	$64.82_{\pm 1.87}$	$\underline{80.62}_{\pm 2.18}$	$66.61_{\pm 1.66}$	$45.00_{\pm 1.81}$	$\underline{80.63}_{\pm 1.87}$	$\underline{72.22}_{\pm 1.66}$	$\underline{53.79}_{\pm 5.97}$

Results The results are reported in Tables 1 and 2. As shown in Table 1, supervised GraphVec consistently outperforms all baseline methods, while unsupervised GraphVec achieves the second-best average performance. GraphVec also remains competitive against in-domain baselines equipped with different fine-tuning strategies under the cross-dataset setting. More importantly, on social-network and computer-vision datasets that differ substantially from the pretraining datasets in both semantics and structure, GraphVec consistently outperforms the compared cross-domain baselines. These results demonstrate that GraphVec learns transferable graph representations across heterogeneous domains without relying on specially designed downstream tuning mechanisms. Compared with supervised GraphVec, the unsupervised variant shows only a moderate performance decrease on most datasets, with larger gaps mainly on COLLAB and Letter-med, and still outperforms all four cross-domain baselines on 4 of the 7 target datasets.

4.2 Graph Clustering

Table 3: Graph clustering performance on ENZYMES, NCI1, COLLAB, REDDIT-BINARY, REDDIT-MULTI. The comparison numbers are from AMGC (Yang et al., 2025).

Method	ENZYMES ACC	ENZYMES NMI	NCI1 ACC	NCI1 NMI	COLLAB ACC	COLLAB NMI	REDDIT-BINARY ACC	REDDIT-BINARY NMI	REDDIT-MULTI ACC	REDDIT-MULTI NMI
InfoGraph +KM	$22.1_{\pm 1.0}$	$2.4_{\pm 0.5}$	$54.1_{\pm 2.2}$	$1.3_{\pm 1.1}$	$59.6_{\pm 1.8}$	$14.4_{\pm 3.0}$	$51.3_{\pm 2.1}$	$2.3_{\pm 0.4}$	$20.3_{\pm 0.9}$	$0.5_{\pm 0.2}$
InfoGraph +SC	$23.8_{\pm 0.5}$	$4.6_{\pm 0.7}$	$54.9_{\pm 1.7}$	$0.9_{\pm 0.6}$	$60.9_{\pm 2.5}$	$15.4_{\pm 3.3}$	$50.8_{\pm 1.3}$	$1.6_{\pm 0.6}$	$24.7_{\pm 1.3}$	$4.8_{\pm 0.6}$
GraphCL +KM	$21.5_{\pm 0.2}$	$1.6_{\pm 0.1}$	$55.4_{\pm 1.7}$	$0.5_{\pm 0.3}$	$58.0_{\pm 1.2}$	$17.8_{\pm 2.0}$	$51.9_{\pm 3.3}$	$3.4_{\pm 1.2}$	$25.3_{\pm 0.9}$	$5.3_{\pm 0.3}$
GraphCL +SC	$25.3_{\pm 0.3}$	$4.8_{\pm 0.4}$	$50.8_{\pm 1.6}$	$0.6_{\pm 0.6}$	$57.8_{\pm 0.6}$	$17.0_{\pm 1.3}$	$55.9_{\pm 2.1}$	$3.2_{\pm 1.0}$	$27.3_{\pm 1.3}$	$5.4_{\pm 0.8}$
JOAO + KM	$21.7_{\pm 0.4}$	$4.9_{\pm 0.4}$	$51.1_{\pm 0.4}$	$0.4_{\pm 0.2}$	$58.3_{\pm 1.5}$	$18.7_{\pm 2.6}$	$54.3_{\pm 2.9}$	$4.2_{\pm 1.8}$	$26.6_{\pm 0.6}$	$3.6_{\pm 1.2}$
JOAO + SC	$24.4_{\pm 1.4}$	$3.2_{\pm 0.7}$	$51.5_{\pm 3.0}$	$0.9_{\pm 1.2}$	$58.2_{\pm 0.9}$	$17.1_{\pm 2.1}$	$55.9_{\pm 1.2}$	$6.7_{\pm 2.0}$	$25.6_{\pm 0.6}$	$2.5_{\pm 0.2}$
GLCC(Ju et al., 2023)	$24.4_{\pm 1.4}$	$3.2_{\pm 0.7}$	$60.9_{\pm 2.3}$	$5.3_{\pm 1.9}$	$60.3_{\pm 0.6}$	$18.2_{\pm 1.3}$	$\mathbf{67.6}_{\pm 3.4}$	$9.2_{\pm 2.6}$	$32.4_{\pm 2.1}$	$11.8_{\pm 1.3}$
AMGC(Yang et al., 2025)	$\mathbf{26.7}_{\pm 2.0}$	$\mathbf{5.2}_{\pm 1.3}$	$\mathbf{62.7}_{\pm 3.0}$	$\mathbf{6.4}_{\pm 1.9}$	$\mathbf{61.2}_{\pm 1.0}$	$\mathbf{20.5}_{\pm 1.6}$	$64.3_{\pm 1.9}$	$\mathbf{12.1}_{\pm 3.3}$	$\mathbf{35.5}_{\pm 2.3}$	$\mathbf{16.1}_{\pm 0.9}$
GraphVec	$\mathbf{29.1}_{\pm 0.4}$	$\mathbf{7.7}_{\pm 0.3}$	$\mathbf{64.8}_{\pm 0.0}$	$\mathbf{6.5}_{\pm 0.0}$	$\mathbf{61.8}_{\pm 0.0}$	$\mathbf{21.2}_{\pm 0.0}$	$\mathbf{71.6}_{\pm 0.0}$	$\mathbf{20.7}_{\pm 0.0}$	$\mathbf{40.0}_{\pm 0.2}$	$\mathbf{17.3}_{\pm 0.1}$

To further evaluate the quality and transferability of GraphVec representations, we apply spectral clustering (Ng et al., 2001) to the pretrained graph embeddings. All target datasets in our clustering experiments are excluded from GraphVec pretraining. In particular, the ENZYMES and NCI1 results are obtained using separate models pretrained on the remaining four source datasets after excluding ENZYMES and NCI1, respectively. As shown in Table 3, GraphVec achieves the best overall performance, further demonstrating its effectiveness for graph-level tasks and its transferability to unseen domains and structures.

Node classification, pretraining scale, low-shot learning, attribute semantic preservation, clustering, ablations, robustness, and efficiency are in **Appendices F.3, F.5-F.7, F.11, F.14, F.18, and F.19**.

5 Conclusions

This paper presented GraphVec, a cross-domain graph vectorization framework that learns fixed-dimensional graph representations from multiple source datasets. GraphVec combines multi-scale relational feature construction, density-based alignment, and reference distribution encoding to improve transfer across heterogeneous graph domains. Experiments on few-shot graph classification and graph clustering demonstrate its strong cross-domain performance. A current limitation is the lack of zero-shot capability without textual information, which could be addressed in future work.

AI USE STATEMENT

We used generative AI tools to assist with polishing the manuscript, improving its organization and clarity, and identifying and checking relevant references. All AI-assisted content was reviewed and revised by the authors, and technical statements and references were checked before inclusion. The authors take full responsibility for the final manuscript and its scientific content.

REFERENCES

Ery Arias-Castro, Adel Javanmard, and Bruno Pelletier. Perturbation bounds for procrustes, classical scaling, and trilateration, with applications to manifold learning. *Journal of Machine Learning Research*, 21(15):1–37, 2020.

Jose Bento and Stratis Ioannidis. A family of tractable graph distances. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 333–341. SIAM, 2018.

Rasmus Bro, Evrim Acar, and Tamara G Kolda. Resolving the sign ambiguity in the singular value decomposition. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 22(2):135–140, 2008.

Horst Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern recognition letters*, 18(8):689–694, 1997.

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE T-KDE*, 30(9):1616–1637, 2018.

Jinyu Cai, Yunhe Zhang, Jicong Fan, Yali Du, and Wenzhong Guo. Dual contrastive graph-level clustering with multiple cluster perspectives alignment. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 3770–3779, 8 2024. Main Track.

Jatin Chauhan, Deepak Nathani, and Manohar Kaul. Few-shot learning on graphs via super-classes based on graph spectral measures. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2002.12815.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36:52464–52489, 2023.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.

Ben Finkelshtein, İsmail İlkan Ceylan, Michael Bronstein, and Ron Levie. Equivariance everywhere all at once: A recipe for graph foundation models. *arXiv preprint arXiv:2506.14291*, 2025.

Xingbo Fu, Yinhan He, and Jundong Li. Edge prompt tuning for graph neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=92vMaHotTM.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.04562*, 2023.

Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pp. 129–143. Springer, 2003.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pp. 1263–1272. PMLR, 2017.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pp. 855–864, 2016.

Kaveh Hassani. Cross-domain few-shot graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6856–6864, 2022.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 594–604, 2022.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM web conference 2023*, pp. 737–746, 2023.

Wei Ju, Yiyang Gu, Binqi Chen, Gongbo Sun, Yifang Qin, Xingyuming Liu, Xiao Luo, and Ming Zhang. Glcc: A general framework for graph-level clustering. In *AAAI*, volume 37, pp. 4391–4399, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

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

Johannes Kobler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Springer Science & Business Media, 2012.

Lecheng Kong, Jiarui Feng, Hao Liu, Chengsong Huang, Jiaxin Huang, Yixin Chen, and Muhan Zhang. GOFA: A generative one-for-all model for joint graph language modeling. In *International Conference on Learning Representations*, pp. 40792–40823, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/652c104b5b0652a03684efeaf805463b-Paper-Conference.pdf.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. *Advances in neural information processing systems*, 32, 2019.

Chuang Liu, Yibing Zhan, Jia Wu, Chang Li, Bo Du, Wenbin Hu, Tongliang Liu, and Dacheng Tao. Graph pooling for graph neural networks: Progress, challenges, and opportunities. *arXiv preprint arXiv:2204.07321*, 2022.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149*, 2023a.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*, pp. 417–428, 2023b.

Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)*, 49(4):1–33, 2016.

Andreas Maurer. A vector-contraction inequality for Rademacher complexities. *arXiv preprint arXiv:1605.00251*, 2016. doi: 10.48550/ARXIV.1605.00251. URL https://arxiv.org/abs/1605.00251.

Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11:417–487, 2011.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *NeurIPS*, 14, 2001.

Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1150–1160, 2020.

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.

Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *JMLR*, 12(9), 2011.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1908.01000.

Li Sun, Zhenhao Huang, Suyang Zhou, Qiqi Wan, Hao Peng, and Philip Yu. Riemanngfm: Learning a graph foundation model from riemannian geometry. In *Proceedings of the ACM on Web Conference 2025*, pp. 1154–1165, 2025.

Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1717–1727, 2022.

Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2120–2131, 2023a.

Yan Sun and Jicong Fan. Mmd graph kernel: Effective metric learning for graphs via maximum mean discrepancy. In *ICLR*, 2024.

Ziheng Sun, Chris Ding, and Jicong Fan. Lovász principle for unsupervised graph representation learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 58290–58311. Curran Associates, Inc., 2023b.

Ziheng Sun, Xudong Wang, Chris Ding, and Jicong Fan. Learning graph representation via graph entropy maximization. In *Forty-first International Conference on Machine Learning*, 2024.

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

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *ICLR*, 2019.

S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *JMLR*, 11:1201–1242, 2010.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? In *Advances in Neural Information Processing Systems*, volume 36, pp. 30840–30861. Curran Associates, Inc., 2023. doi: 10.52202/075280-1345. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/622afc4edf2824a1b6aaf5afe153fa93-Paper-Conference.pdf.

Kai Wang and Siqiang Luo. Towards graph foundation models: The perspective of zero-shot reasoning on knowledge graphs. *arXiv preprint arXiv:2410.12609v1*, 2024. URL https://arxiv.org/abs/2410.12609v1.

Shiping Wang, Jinbin Yang, Jie Yao, Yang Bai, and William Zhu. An overview of advanced deep graph node clustering. *IEEE Transactions on Computational Social Systems*, 11(1):1302–1314, 2024a. doi: 10.1109/TCSS.2023.3242145.

Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. *arXiv preprint arXiv:2502.02017*, 2025a.

Xingliang Wang, Zemin Liu, Junxiao Han, and Shuiguang Deng. Rag4gfm: Bridging knowledge gaps in graph foundation models through graph retrieval augmented generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Zehong Wang, Zheyuan Zhang, Nitesh Chawla, Chuxu Zhang, and Yanfang Ye. Gft: Graph foundation model with transferable tree vocabulary. *Advances in Neural Information Processing Systems*, 37: 107403–107443, 2024b.

Zixiao Wang and Jicong Fan. Graph classification via reference distribution learning: Theory and practice. In *Advances in Neural Information Processing Systems*, volume 37, pp. 137698–137740. Curran Associates, Inc., 2024.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM web conference 2022*, pp. 1070–1079, 2022.

Hongteng Xu, Jiachang Liu, Dixin Luo, and Lawrence Carin. Representing graphs via gromov-wasserstein factorization. *IEEE T-PAMI*, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

Jinbin Yang, Jinyu Cai, Yunhe Zhang, Sujia Huang, and Shiping Wang. Towards adaptive masked structural learning for graph-level clustering. *IEEE Transactions on Network Science and Engineering*, 2025.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823, 2020.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *ICML*, pp. 12121–12132. PMLR, 2021.

Xingtong Yu, Zhenghao Liu, Yuan Fang, Zemin Liu, Sihong Chen, and Xinming Zhang. Generalized graph prompt: Toward a unification of pre-training and downstream tasks on graphs. *IEEE Transactions on Knowledge and Data Engineering*, 2024a.

Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Multigprompt for multi-task pre-training and prompting on graphs. In *Proceedings of the ACM Web Conference 2024*, pp. 515–526, 2024b.

Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. Samgpt: Text-free graph foundation model for multi-domain pre-training and cross-domain adaptation. In *Proceedings of the ACM on Web Conference 2025*, pp. 1142–1153, 2025a.

Xingtong Yu, Jie Zhang, Yuan Fang, and Renhe Jiang. Non-homophilic graph pre-training and prompt learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1844–1854, 2025b.

Xingtong Yu, Chang Zhou, Zhongwei Kuai, Xinming Zhang, and Yuan Fang. Gcot: Chain-of-thought prompt learning for graphs. *arXiv preprint arXiv:2502.08092*, 2025c.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S Yu. Graver: Generative graph vocabularies for robust graph foundation models fine-tuning. *arXiv preprint arXiv:2511.05592*, 2025a.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S Yu. How much can transfer? bridge: Bounded multi-domain graph foundation model with generalization guarantees. In *Forty-second International Conference on Machine Learning*, 2025b.

Zhiping Zeng, Anthony KH Tung, Jianyong Wang, Jianhua Feng, and Lizhu Zhou. Comparing stars: On approximating graph edit distance. *Proceedings of the VLDB Endowment*, 2(1):25–36, 2009.

Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4443–4454, 2024a.

Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On learning invariant representations for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7523–7532. PMLR, 2019.

Jianan Zhao, Hesham Mostafa, Michael Galkin, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. Graphany: A foundation model for node classification on any graph. *arXiv preprint arXiv:2405.20445*, 2024b.

A NOTATIONS

Table 4: Notations

Symbol	Meaning	Symbol	Meaning
x	a real number	$\mathbf{x}$	a vector
$\mathbf{X}$	a matrix	$\mathbf{I}_n$	identity matrix of size $n \times n$
G	a graph	$\mathbf{g}$	vector representation of G
$\mathcal{G}$	a set of graphs	$\mathcal{D}$	a dataset
$\|\mathbf{x}\|$	the Euclidean norm of $\mathbf{x}$	$\|\mathbf{x}\|_1$	the ℓ_1 norm of $\mathbf{x}$
$[M]$	the set $\{1, 2, \ldots, M\}$	$\mathbf{X}\|\mathbf{Y}$ or $[\mathbf{X}, \mathbf{Y}]$	vertical concatenation
$\|\mathbf{X}\|_F$	Frobenius norm of matrix	$\|\mathbf{X}\|_2$	spectral norm of matrix

B MORE ABOUT RELATED WORK

Language Model-Free GFMs Many studies have explored training GFMs using the "pre-train and adaptation" paradigm, leveraging message-passing-based or transformer-based GNNs as backbones. These approaches typically employ contrastive or generative self-supervised learning for pretraining, followed by fine-tuning a subset of model parameters to adapt to downstream tasks or datasets (Liu et al., 2025). Contrastive methods (Qiu et al., 2020; Sun et al., 2020; Veličković et al., 2019; Xia et al., 2022) typically aim to produce generalized graph representations through maximizing the agreement between different augmentations of the same instance. For example, GraphCL (You et al., 2020) designs four types of graph data augmentations to learn invariant representations under specialized perturbations. GCOPE (Zhao et al., 2024a) employs a graph contrastive learning framework and introduces coordinators which are some virtual nodes that function as dynamic bridges between disparate graph datasets. Focused on node-level tasks, it effectively mitigates negative transfer effects when pretraining graph models on cross-domain datasets. In the meantime, generative methods pre-train GNNs through graph reconstruction or property prediction. For instance, GraphMAEs (Hou et al., 2022; 2023) employed the reconstruction of features with masking strategies. Recently, graph prompt tuning methods (Sun et al., 2022; Fang et al., 2023; Sun et al., 2023a; Liu et al., 2023b) have been proposed as an adaptation mechanism to bridge the gap between pretraining tasks and downstream tasks. GraphPrompt (Liu et al., 2023b) converts the pretraining task and downstream tasks to follow the same template based on subgraph similarity and uses learnable prompt vectors to implement different aggregation schemes for readout in different downstream tasks. EdgePrompt (Fu et al., 2025) manipulates input graphs by learning prompt vectors for edges and incorporates the edge prompts through message passing in the pretrained GNN models. As an effective adaptation mechanism, these methods have been widely adopted in subsequent GFMs (Yuan et al., 2025b;a; Wang et al., 2025a; Yu et al., 2024b; 2025a). Recently, several notable Graph Foundation Models have been introduced. RiemannGFM (Sun et al., 2025) embeds nodes into a Riemann manifold using structural vocabulary of trees as circles, enable structral transferability across domains. TS-GNN (Finkelshtein et al., 2025) investigates symmetries that a graph foundation model must respect, which is mainly designed for node-level tasks. RAG4GFM (Wang et al., 2025b) applies the Retrieval-Augmented Generation (RAG) paradigm to Graph Foundation Models, allowing them to dynamically access and integrate graph knowledge at inference time. SCORE (Wang & Luo, 2024) utilizes KGs as a unified topological structure to tackle diverse tasks.

LLM-based GFMs These models utilize the strong capacity of large language models to conduct graph analysis. For instance, GraphQA (Fatemi et al., 2023) converts graph connectivity into textual descriptions and uses LLMs to answer graph reasoning questions. By enriching these prompts with domain-specific context, GraphQA can effectively learn cross-domain structural representations, essentially serving as a structural GFM. Similar approaches include NLGraph (Wang et al., 2023), which tackles tasks like shortest path finding by translating graphs into text, demonstrating another viable pathway for unified structure learning. For unifying node feature representations, the One For All (OFA) framework (Liu et al., 2023a) offers an innovative solution. It aggregates diverse graph datasets into a unified text-attributed graph (TAG) format, then leverages LLMs to jointly learn

feature representations that transcend domain boundaries. This approach effectively bridges the gap between heterogeneous graph data sources.

GFM for Graph-Level Tasks As mentioned before, most of the existing GFMs are designed for node-level tasks. There are a few studies that focus on graph-level tasks across domains. For instance, Chauhan et al. (2020) tries to pretrain GNNs on certain classes of a dataset and conduct few-shot classification on the remaining classes within the same dataset. Hassani (2022) adopts a meta-learning approach to learn model initialization for few-shot graph classification. These graph-level models are usually small and not general. Some GFMs can be adapted to graph-level tasks. For instance, GraphPrompt (Liu et al., 2023b), GraphPrompt+ (Yu et al., 2024a), and EdgePrompt (Fu et al., 2025) use learnable prompts to adjust graph-level pooling for obtaining domain-adaptive graph embeddings. Other works such as SAMGPT (Yu et al., 2025a), ProNoG (Yu et al., 2025b) MultiGPrompt (Yu et al., 2024b) , and BRIDGE (Yuan et al., 2025b) are also designed to effectively perform graph classification, but they mainly build on well-designed node embedding and use simple global pooling to apply the model to graph-level tasks.

C Derivation of Algorithm 1

Recall that we aim to solve

$$\underset{\mathbf{R}_j \in \mathcal{R},\, j \in [M]}{\text{maximize}} \;\; \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \exp\left(-\gamma_{\mathrm{A}} \|\mathbf{R}_i \boldsymbol{\mu}_i - \mathbf{R}_j \boldsymbol{\mu}_j\|_2^2\right) \triangleq \mathcal{L}\left(\{\mathbf{R}_j\}_{j=1}^{M}\right), \tag{19}$$

where

$$\mathcal{R} = \left\{ \mathbf{R} \in \mathbb{R}^{\bar{d} \times \bar{d}} : \mathbf{R}^\top \mathbf{R} = \mathbf{I}_{\bar{d}} \right\}. \tag{20}$$

The factor $1/M$ does not affect the maximizer. Therefore, for notational simplicity, we consider the equivalent unnormalized objective

$$L(\mathbf{R}) = \sum_{i=1}^{M} \sum_{j=1}^{M} \exp\left(-\gamma_{\mathrm{A}} \|\mathbf{R}_i \boldsymbol{\mu}_i - \mathbf{R}_j \boldsymbol{\mu}_j\|_2^2\right), \tag{21}$$

where

$$\mathbf{R} := \{\mathbf{R}_j \in \mathcal{R}\}_{j=1}^{M}. \tag{22}$$

Since each $\mathbf{R}_j$ is orthogonal, we have

$$\|\mathbf{R}_j \boldsymbol{\mu}_j\|_2 = \|\boldsymbol{\mu}_j\|_2. \tag{23}$$

Thus,

$$\begin{aligned} L(\mathbf{R}) &= \sum_{i=1}^{M} \sum_{j=1}^{M} \exp\left[-\gamma_{\mathrm{A}} \left(\|\boldsymbol{\mu}_i\|_2^2 + \|\boldsymbol{\mu}_j\|_2^2\right)\right] \exp\left(2\gamma_{\mathrm{A}} \langle \mathbf{R}_i \boldsymbol{\mu}_i, \mathbf{R}_j \boldsymbol{\mu}_j \rangle\right) \\ &= \sum_{i=1}^{M} \sum_{j=1}^{M} w_{ij} \exp\left(2\gamma_{\mathrm{A}} \langle \mathbf{R}_i \boldsymbol{\mu}_i, \mathbf{R}_j \boldsymbol{\mu}_j \rangle\right), \end{aligned} \tag{24}$$

where

$$w_{ij} := \exp\left[-\gamma_{\mathrm{A}} \left(\|\boldsymbol{\mu}_i\|_2^2 + \|\boldsymbol{\mu}_j\|_2^2\right)\right]. \tag{25}$$

Let $\mathbf{R}^{(t-1)} = \{\mathbf{R}_j^{(t-1)}\}_{j=1}^{M}$ be the current iterate. Define

$$k_{ij}^{(t-1)} := \exp\left(2\gamma_{\mathrm{A}} \left\langle \mathbf{R}_i^{(t-1)} \boldsymbol{\mu}_i, \mathbf{R}_j^{(t-1)} \boldsymbol{\mu}_j \right\rangle\right). \tag{26}$$

The Euclidean gradient of L with respect to $\mathbf{R}_i$ at $\mathbf{R}^{(t-1)}$ is

$$\nabla_{\mathbf{R}_i} L(\mathbf{R}^{(t-1)}) = 4\gamma_{\mathrm{A}} \sum_{j=1}^{M} w_{ij} k_{ij}^{(t-1)} \mathbf{R}_j^{(t-1)} \boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top. \tag{27}$$

For compactness, denote

$$\mathbf{A}_i^{(t)} := \sum_{j=1}^{M} w_{ij} k_{ij}^{(t-1)} \mathbf{R}_j^{(t-1)} \boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top. \tag{28}$$

Then

$$\nabla_{\mathbf{R}_i} L(\mathbf{R}^{(t-1)}) = 4\gamma_{\mathrm{A}} \mathbf{A}_i^{(t)}. \tag{29}$$

At iteration t, we maximize the following surrogate:

$$\widetilde{L}_t(\mathbf{R}) = \sum_{i=1}^{M} \left\langle \nabla_{\mathbf{R}_i} L(\mathbf{R}^{(t-1)}), \mathbf{R}_i - \mathbf{R}_i^{(t-1)} \right\rangle - 2\gamma_{\mathrm{A}} \eta \sum_{i=1}^{M} \|\mathbf{R}_i - \mathbf{R}_i^{(t-1)}\|_F^2, \tag{30}$$

subject to $\mathbf{R}_i \in \mathcal{R}$ for all $i \in [M]$. Substituting (29) into (30), we obtain

$$\widetilde{L}_t(\mathbf{R}) = 4\gamma_{\mathrm{A}} \sum_{i=1}^{M} \left\langle \mathbf{A}_i^{(t)}, \mathbf{R}_i - \mathbf{R}_i^{(t-1)} \right\rangle - 2\gamma_{\mathrm{A}} \eta \sum_{i=1}^{M} \|\mathbf{R}_i - \mathbf{R}_i^{(t-1)}\|_F^2. \tag{31}$$

Because both $\mathbf{R}_i$ and $\mathbf{R}_i^{(t-1)}$ are orthogonal, we have

$$\|\mathbf{R}_i - \mathbf{R}_i^{(t-1)}\|_F^2 = 2\bar{d} - 2\left\langle \mathbf{R}_i^{(t-1)}, \mathbf{R}_i \right\rangle. \tag{32}$$

Therefore, after dropping constants independent of $\mathbf{R}$ and removing the positive factor $4\gamma_{\mathrm{A}}$, maximizing (31) is equivalent to

$$\underset{\mathbf{R}_i \in \mathcal{R},\, i \in [M]}{\text{maximize}} \sum_{i=1}^{M} \left\langle \mathbf{A}_i^{(t)} + \eta \mathbf{R}_i^{(t-1)}, \mathbf{R}_i \right\rangle. \tag{33}$$

Let

$$\mathbf{H}_i^{(t)} := \mathbf{A}_i^{(t)} + \eta \mathbf{R}_i^{(t-1)} = \sum_{j=1}^{M} w_{ij} k_{ij}^{(t-1)} \mathbf{R}_j^{(t-1)} \boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top + \eta \mathbf{R}_i^{(t-1)}. \tag{34}$$

Then the subproblem separates over i as

$$\mathbf{R}_i^{(t)} = \underset{\mathbf{R}_i^\top \mathbf{R}_i = \mathbf{I}_{\bar{d}}}{\arg\max} \left\langle \mathbf{H}_i^{(t)}, \mathbf{R}_i \right\rangle. \tag{35}$$

This is the classical orthogonal Procrustes problem (Schönemann, 1966). Let the singular value decomposition of $\mathbf{H}_i^{(t)}$ be

$$\mathbf{H}_i^{(t)} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^\top. \tag{36}$$

Then the optimal solution is

$$\mathbf{R}_i^{(t)} = \mathbf{U}_i \mathbf{V}_i^\top. \tag{37}$$

This gives the update rule used in Algorithm 1.

D Details of GIN and Graph Transformer based Model

To design a universal graph representation model F, we incorporate two main components: a GNN module f and a graph transformer module g. We build a GIN encoder followed by a graph transformer encoder $g_\psi \circ f_\theta(\cdot)$. The GNN encoder specializes in learning local representations of the structure of a node's immediate neighborhood, while the transformer computes all pairwise node interactions, enabling global reasoning through attention mechanisms. Specifically, we adopt the Graph Isomorphism Network (GIN) (Xu et al., 2019) as the GNN encoder, and its l-th layer can be formulated as

$$f^{(l)}\left(\mathbf{A}_i^{(j)}, \mathbf{Z}_i^{(j)}\right) = \mathrm{MLP}^{(l)}\left(\left(\tilde{\mathbf{A}}_i^{(j)} + \epsilon \mathbf{I}\right) \cdot \mathbf{Z}_i^{(j)}\right) \tag{38}$$

where $\tilde{\mathbf{A}}_i^{(j)}$ is the adjacency matrix of $G_i^{(j)}$ with self-loops, ϵ is a hyperparameter, $\mathrm{MLP}^{(l)}$ is a multilayer perceptron (MLP) in layer l, and the parameters to optimize are denoted as θ.

The graph transformer (GT) module consists of a self-attention mechanism and a feed-forward network, which is usually an MLP. Let $\mathbf{\Gamma}_i^{(j)} \in \mathbb{R}^{n_i \times d}$ represent the matrix of hidden states, and $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ of size $d \times d'$ be projection matrices, the self-attention mechanism is

$$\operatorname{attn}\left(\mathbf{\Gamma}_i^{(j)}\right) = \operatorname{softmax}\left(\frac{(\mathbf{\Gamma}_i^{(j)}\mathbf{W}_Q)(\mathbf{\Gamma}_i^{(j)}\mathbf{W}_K)^\top}{\sqrt{d'}}\right)(\mathbf{\Gamma}_i^{(j)}\mathbf{W}_V) \tag{39}$$

which is further transformed to $\hat{\mathbf{\Gamma}}_i^{(j)} = \operatorname{Norm}\left(\mathbf{\Gamma}_i^{(j)} + \operatorname{attn}\left(\mathbf{\Gamma}_i^{(j)}\right)\right)$. Then the l-th transformer block can be formulated as

$$g^{(l)}\left(\mathbf{\Gamma}_i^{(j)}\right) = \operatorname{Norm}\left(\hat{\mathbf{\Gamma}}_i^{(j)} + \operatorname{FFN}\left(\hat{\mathbf{\Gamma}}_i^{(j)}\right)\right) \tag{40}$$

We denote the parameters of the transformer module as ψ. Finally, we concatenate the outputs of the GIN and GT, leading to the following node representations of $G_i^{(j)}$:

$$\mathbf{H}_i^{(j)} = g_\psi \circ f_\theta\left(\mathbf{A}_i^{(j)}, \mathbf{Z}_i^{(j)}\right) \Big\| f_\theta\left(\mathbf{A}_i^{(j)}, \mathbf{Z}_i^{(j)}\right), \quad i \in [N_j], \quad j \in [M]. \tag{41}$$

For convenience, we let $\mathcal{W} = \{\psi, \theta\}$, which is the set of all parameters of the GIN and GT.

D.1 MMD Expansion for Reference Distribution Encoding

The negative MMD similarity between a graph's node representations and a reference distribution can be expanded as

$$\begin{aligned} s_{i,b}^{(j,l)} &= -\text{MMD}\left(\mathbf{H}_i^{(j,l)}, \mathbf{V}_b^{(l)}\right) \\ &= -\Big\|\frac{1}{n_i}\sum_{p=1}^{n_i}\phi\left(\mathbf{h}_p^{(j,l)}\right) - \frac{1}{m}\sum_{q=1}^{m}\phi\left(\mathbf{v}_q^{(b,l)}\right)\Big\|_2 \\ &= -\Big[\frac{1}{n_i^2}\sum_{p,p'\in[n_i]}k\left(\mathbf{h}_p^{(j,l)}, \mathbf{h}_{p'}^{(j,l)}\right) + \frac{1}{m^2}\sum_{q,q'\in[m]}k\left(\mathbf{v}_q^{(b,l)}, \mathbf{v}_{q'}^{(b,l)}\right) \\ &\quad - \frac{2}{mn_i}\sum_{p\in[n_i],q\in[m]}k\left(\mathbf{h}_p^{(j,l)}, \mathbf{v}_q^{(b,l)}\right)\Big]^{\frac{1}{2}} \end{aligned} \tag{42}$$

where ϕ is the high-dimensional feature map induced by a kernel function, $\mathbf{h}_p^{(j)}$ is the p-th row of $\mathbf{H}_i^{(j)}$, $\mathbf{v}_q^{(b)}$ is the q-th row of $\mathbf{V}_b$, and $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2\right)$ is the Gaussian kernel with a learnable parameter γ.

E Details about Experimental Settings

E.1 Datasets

The basic information and statistics of the graph datasets we used in the experiments are shown in Table 5. In our experiments, the concatenation of the original node attributes and node labels in the datasets is used as initial input node features.

E.2 Details of Model Testing in Few-Shot Graph Classification

Let $\mathcal{G}^{\text{Down}} = \{\mathcal{G}^{\text{train}}, \mathcal{G}^{\text{test}}\}$ denote the target training/support and query sets, respectively. For each scale $q \in [Q]$, construct $\mathbf{K}_{\lambda_q}^{\text{train}}$ using only support nodes and compute

$$\mathbf{K}_{\lambda_q}^{\text{train}} = \mathbf{U}_q\mathbf{\Sigma}_q\mathbf{V}_q^\top, \qquad \mathbf{Z}_{\lambda_q}^{\text{train}} = \mathbf{U}_{q,\bar{d}}\mathbf{\Sigma}_{q,\bar{d}}^{1/2}. \tag{43}$$

At each scale, keep all pretrained source transformations fixed and optimize only the target transformation $\mathbf{R}_q^{\text{train}}$ using the same SVD/Procrustes update as Algorithm 1. Align the support features as $\widetilde{\mathbf{Z}}_{\lambda_q}^{\text{train}} = \mathbf{Z}_{\lambda_q}^{\text{train}}(\mathbf{R}_q^{\text{train}})^\top$ and concatenate them across scales. The pretrained encoder maps each

Table 5: Dataset Statistics.

Dataset	Domain	#Graphs	#Avg.Nodes	Node features	#Classes	Task
ENZYMES	Bioinformatics	600	32.63	Yes	6	Graph Classification/Graph Clustering
NCI1	Small molecules	4110	29.87	Yes	2	Graph Classification/Graph Clustering
NCI109	Small molecules	4127	29.68	Yes	2	Graph Classification
DD	Bioinformatics	1178	284.32	Yes	2	Graph Classification
Mutagenicity	Small molecules	4337	30.32	Yes	2	Graph Classification
COLLAB	Social networks	5000	74.49	No	2	Graph Classification/Graph Clustering
REDDIT-BINARY	Social networks	2000	429.63	No	2	Graph Classification/Graph Clustering
REDDIT-MULTI	Social networks	4999	508.52	No	5	Graph Clustering
IMDB-BINARY	Social networks	1000	19.77	No	2	Graph Classification
IMDB-MULTI	Social networks	1500	13.00	No	3	Graph Classification
Letter-med	Computer vision	2250	4.67	Yes	15	Graph Classification
COIL-RAG	Computer vision	3900	3.01	Yes	100	Graph Classification
Cuneiform	Computer vision	267	21.27	Yes	30	Graph Classification

support graph to $\mathbf{g}_i^{\text{train}} = F_{\mathcal{W},\mathcal{V},\gamma}(\mathbf{A}_i^{\text{train}}, \widetilde{\mathbf{Z}}_i^{\text{train}})$, which is used to train the downstream softmax classifier. The support-derived preprocessing statistics, spectral bases, and alignment matrices are then fixed for query inference.

For a query graph G_i^{test}, compute $\mathbf{K}_{i,\lambda_q}^{\text{test,train}}$ between its nodes and the support nodes, using the fixed support-derived kernel normalization. Apply (16) and concatenate the aligned features as $\widetilde{\mathbf{Z}}_i^{\text{test}} = [\widetilde{\mathbf{Z}}_{i,\lambda_1}^{\text{test}}, \ldots, \widetilde{\mathbf{Z}}_{i,\lambda_Q}^{\text{test}}]$. The query embedding is $\mathbf{g}_i^{\text{test}} = F_{\mathcal{W},\mathcal{V},\gamma}(\mathbf{A}_i^{\text{test}}, \widetilde{\mathbf{Z}}_i^{\text{test}})$. No query–query kernel, test-set mean, or separately estimated $\mathbf{R}^{\text{test}}$ is used. Algorithm 1 is not rerun on query graphs. Processing queries individually or in batches only partitions the rows of the cross-kernel and yields the same per-graph representations. Algorithm 3 summarizes the procedure.

Algorithm 3 Inductive few-shot graph classification

Input: Support set $\mathcal{G}^{\text{train}}$ with labels, query graphs $\mathcal{G}^{\text{test}}$, pretrained encoder $F_{\mathcal{W},\mathcal{V},\gamma}$, and source means and alignment matrices at each scale.

for each scale $q \in [Q]$ **do**
  Construct the support kernel and spectral features using (1) and (43).
  Keep the pretrained source transformations fixed and optimize only $\mathbf{R}_q^{\text{train}}$ using the SVD/Procrustes update of Algorithm 1.
  $\widetilde{\mathbf{Z}}_{\lambda_q}^{\text{train}} \leftarrow \mathbf{Z}_{\lambda_q}^{\text{train}} (\mathbf{R}_q^{\text{train}})^\top$.
end for
Concatenate aligned support features across scales to obtain $\widetilde{\mathbf{Z}}^{\text{train}}$.
Freeze support preprocessing statistics, spectral bases, and $\{\mathbf{R}_q^{\text{train}}\}_{q=1}^{Q}$.
$\mathbf{g}_i^{\text{train}} \leftarrow F_{\mathcal{W},\mathcal{V},\gamma}(\mathbf{A}_i^{\text{train}}, \widetilde{\mathbf{Z}}_i^{\text{train}})$ for each support graph.
Train the softmax classifier f_c on the support embeddings and labels.
for each query graph G_i^{test} **do**
  Compute $\mathbf{K}_{i,\lambda_q}^{\text{test,train}}$ against fixed support nodes for every $q \in [Q]$.
  Project and align using (16); concatenate across scales to obtain $\widetilde{\mathbf{Z}}_i^{\text{test}}$.
  $\hat{\mathbf{y}}_i^{\text{test}} \leftarrow f_c \circ F_{\mathcal{W},\mathcal{V},\gamma}(\mathbf{A}_i^{\text{test}}, \widetilde{\mathbf{Z}}_i^{\text{test}})$.
end for

Output: Predicted graph labels $\{\hat{\mathbf{y}}_i^{\text{test}}\}$.

E.3 Inplementation Details

In our experiments, we use 6 Gaussian kernels with different $\lambda_q \in \{0.25, 0.5, 1, 2, 5, 10\}$. For all kernel matrices and the adjacency matrix, the truncated dimension of SVD $\bar{d}$ is set as 32. For each global graph obtained by Gaussian kernels, we use 6-GIN encoder to encode node features from different global graphs respectively. We implement each GIN encoder with 3 graph convolutional layers. The size of each hidden layer in GIN is set to 128. The graph transformer module consists of 3 equally wide layers, each containing 4 attention heads, with the dimension of each attention head set as 48. In the pretraining stage, all modules are optimized using Adam optimizer (Kingma & Ba, 2015) with fixed learning rate $\alpha_1 = 0.0005$ and a weight decay factor of 10^{-5}, trained for 50 epochs. The Gaussian kernel parameter γ in the reference layer employs a separate learning rate $\alpha_2 = 0.1$. The batch size for all datasets is fixed to 64.

Sign convention for structural features For datasets without original node attributes, we fix the sign of each retained singular vector from the truncated SVD of $\mathbf{A} + \mathbf{I}$ in (5). For each vector $\mathbf{u}$, we choose between $\mathbf{u}$ and $-\mathbf{u}$ to maximize the number of nonnegative entries before using it to construct the node features.

Few-shot learning settings In the downstream tasks of few-shot graph classification, the classifier is a softmax classifier, which follows the setting in EdgePrompt (Fu et al., 2025). Regarding data splitting, we randomly choose 50 graphs in each class for training, and the remaining samples are used for testing. The number of epochs is set to 500, and the learning rate of the classifier is set to 0.001 for graph few-shot training. As the k-shot tasks are balanced classification, we employ accuracy as the evaluation metric following EdgePrompt.
For ProNoG, we used the provided checkpoint from the official open-source repository as the pretrained model. For BRIDGE, GFT, and RiemannGFM, we followed the recommended settings in their paper to pretrain the model. The official repository of RiemannGFM does not support graph classification, we extend it to graph level task by using mean pooling. In the downstream adaptation stage, we adopted the recommended hyperparameters for both methods. Experiments on COLLAB, REDDIT-B, IMDB-B, IMDB-M, and Letter-med are conducted under 50-shot setting following experiments in our paper. For COIL-RAG and Cuneiform, due to a lack of enough samples per class, we adopt 5-shot and 1-shot settings, respectively. Since the three baselines do not handle datasets without node attributes, to ensure fair comparison, we handle social network datasets without node attributes (COLLAB, REDDIT-B, IMDB-B, IMDB-M) uniformly across all methods. Following our proposed approach, we generate node attributes using truncated SVD on A+I (adjacency matrix with self-loops) as input for all baseline models. For these cross-domain few-shot classification experiments, our models are pretrained on the five bio-chemical datasets (ENZYMES, DD, NCI1, NCI109, Mutagenicity), which differ significantly from the target social-network and computer-vision datasets in both semantics and structure.

In cross domain experiments, for graph-prompt-based baselines, which are designed to train and test within a single dataset, we preprocess the raw node attributes of each target dataset by PCA truncation to 32 dimensions; if the feature dimension is smaller than 32, we zero-pad it to 32 dimensions. Other baselines follow the same cross-domain protocol as GraphVec: they are pretrained on the five bio-chemical datasets (ENZYMES, DD, NCI1, NCI109, Mutagenicity) and evaluated on target graphs from other domains. In the downstream adaptation stage, we adopt the recommended hyperparameters for each baseline. Experiments on COLLAB, REDDIT-B, IMDB-B, IMDB-M, and Letter-med are conducted under the 50-shot setting. For COIL-RAG and Cuneiform, due to a lack of enough samples per class, we adopt 5-shot and 1-shot settings, respectively. SAMGPT is trained for 20 epochs, GOFA is fine-tuned for one epoch from the official checkpoint, and the remaining baselines are trained for 50 epochs. For LLM-based methods, raw graphs must first be converted into text-attributed graphs. Since the node features in our datasets are numerical rather than natural-language attributes, this conversion may produce highly similar text embeddings. In practice, we observe different degrees of representation collapse, which may explain the weak performance of LLM-based methods on several datasets. For datasets without node attributes, we handle them uniformly across all methods by generating structural node attributes using truncated SVD on $\mathbf{A}+\mathbf{I}$, following the same setting as GraphVec.

Unsupervised pretraining settings Following You et al. (2020), we construct 3 augmentations using dropping nodes with a ratio of 0.1, permuting edges with a ratio of 0.1, and extracting subgraph for each graph before the global multi-graph construction and max-density mean alignment.

We conduct all experiments on a 14 vCPU Intel(R) Xeon(R) Gold 6348 CPU with one Nvidia A800-80G GPU, CUDA 11.8. We repeat five times with different random seeds and report the average results with standard deviation calculated by the numpy library function.

F MORE RESULTS

F.1 INTUITIVE EXAMPLE OF THE GLOBAL GRAPH CONSTRUCTION

Here we provide an intuitive example of synthetic data to show that our graph construction could be domain-agnostic. Suppose we have four datasets $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$ drawn from the following four distributions respectively: 1) $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ (2D Gaussian); 2) $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ (2D Gaussian); 3) $\mathcal{N}(\mathbf{1}, 2\mathbf{I}_3)$ (3D Gaussian); 4) $\mathcal{N}(-\mathbf{2}, \mathbf{I}_2) + \mathcal{N}(\mathbf{2}, \mathbf{I}_2)$ (2D Gaussian mixture model). Thus, $\mathcal{D}_2$ can be regarded as a dataset from the same domain as $\mathcal{D}_1$, while $\mathcal{D}_3$ and $\mathcal{D}_4$ are from different domains. We calculate the Gromov-Wasserstein distances between the weighted graphs constructed from the four datasets using the method proposed in our paper. The results are shown in the following table (average of 5 runs). We see that the distance between $\mathcal{D}_1$ and $\mathcal{D}_3$ is close to that between $\mathcal{D}_1$ and $\mathcal{D}_2$, meaning that the features generated by our multi-graph alignment method are indeed domain agnostic. The distance between $\mathcal{D}_1$ and $\mathcal{D}_4$ is much larger than that between $\mathcal{D}_1$ and $\mathcal{D}_3$, meaning that our method can effectively identify the topological difference between the datasets.

Table 6: Gromov-Wasserstein distances between synthetic datasets (average of 5 runs)

	$\mathcal{D}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$	$\mathcal{D}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$	$\mathcal{D}_3 \sim \mathcal{N}(\mathbf{1}, 2\mathbf{I}_3)$	$\mathcal{D}_4 \sim \mathcal{N}(-\mathbf{2}, \mathbf{I}_2) + \mathcal{N}(\mathbf{2}, \mathbf{I}_2)$
$\mathcal{D}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$	0	0.004	0.015	0.069
$\mathcal{D}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$	—	0	0.015	0.069
$\mathcal{D}_3 \sim \mathcal{N}(\mathbf{1}, 2\mathbf{I}_3)$	—	—	0	0.081
$\mathcal{D}_4 \sim \mathcal{N}(-\mathbf{2}, \mathbf{I}_2) + \mathcal{N}(\mathbf{2}, \mathbf{I}_2)$	—	—	—	0

F.2 FULL RESULTS OF TABLE 1

The full compared numbers in the baseline of Table 1 are provided in Table 7.

F.3 EXTENSION TO NODE CLASSIFICATION TASK

Our model can also be extended to node-level tasks by retraining it with a node-level contrastive loss objective. Specifically, we maintain the construction of global multi-graphs and the mean alignment module from our original framework, while removing the reference distribution layers since graph-level representations are not required here. The node embeddings are obtained directly from the outputs of both the Graph Transformer and GIN modules. These embeddings are then fed into a linear classifier to perform the downstream node classification task.

The model was evaluated on 4 node classification datasets: Cora, CiteSeer, PubMed, and ogbn-arxiv. We follow a leave-one-dataset-out protocol: for each target dataset, GraphVec is pretrained on the remaining three datasets and then evaluated on the target dataset under the 5-shot setting. The node embeddings produced by the pretrained encoder are used to train a linear classifier with the labeled nodes, and the remaining nodes are used for testing. The results are shown as follows. This adaptation demonstrates that GraphVec can also be effectively extended to node-level tasks.

F.4 THE IMPACT OF NYSTRÖM APPROXIMATION

To address the computational complexity associated with large-scale graphs, we employ the Nyström approximation during pretraining. To systematically evaluate its impact, we pre-train GraphVec using the Nyström method with varying sample sizes, while keeping the pretraining datasets consistent with our main experiments. In the following table, we specifically report the wall-clock time required for constructing the global multi-graphs, and evaluate downstream performance via 50-shot graph classification accuracy on the Letter-med dataset. We see that, when the sample size is less than 2000, the time used for constructing global multi-graphs is acceptable. When adding the sample size to 4000, the wall-clock time increases sharply while the performance improvement is marginal (less than 0.1%).

Table 7: 50-shot graph classification performance comparison with different pretrained models. We color the **best** and **second best** models. The compared numbers of in-domain experiments are from EdgePrompt (Fu et al., 2025).

pretraining	Tuning Methods	ENZYMES	DD	NCI1	NCI109	Mutagenicity	Average
GraphCL	Classifier Only	$30.50_{\pm 1.16}$	$62.89_{\pm 2.19}$	$62.49_{\pm 1.95}$	$61.68_{\pm 0.93}$	$66.62_{\pm 1.87}$	56.84
	GraphPrompt (Liu et al., 2023b)	$27.83_{\pm 1.61}$	$64.33_{\pm 1.79}$	$63.19_{\pm 1.71}$	$62.18_{\pm 0.48}$	$67.62_{\pm 0.65}$	57.03
	ALL-in-one (Sun et al., 2023a)	$25.92_{\pm 0.55}$	$66.54_{\pm 1.82}$	$57.52_{\pm 2.61}$	$62.74_{\pm 0.78}$	$63.43_{\pm 2.53}$	55.23
	GPF (Fang et al., 2023)	$30.08_{\pm 1.25}$	$64.54_{\pm 2.22}$	$62.66_{\pm 1.83}$	$62.29_{\pm 0.90}$	$66.54_{\pm 1.85}$	57.22
	GPF-plus (Fang et al., 2023)	$31.00_{\pm 1.50}$	$67.26_{\pm 2.29}$	$64.56_{\pm 1.10}$	$62.84_{\pm 0.22}$	$66.82_{\pm 1.63}$	58.50
	EdgePrompt (Fu et al., 2025)	$29.50_{\pm 1.57}$	$64.16_{\pm 2.13}$	$63.05_{\pm 2.11}$	$62.59_{\pm 0.93}$	$66.87_{\pm 1.88}$	57.23
	EdgePrompt+ (Fu et al., 2025)	$34.00_{\pm 1.25}$	$67.98_{\pm 2.05}$	$66.30_{\pm 2.54}$	$66.52_{\pm 0.91}$	$67.47_{\pm 2.37}$	60.45
SimGRACE	Classifier Only	$27.07_{\pm 1.04}$	$61.77_{\pm 2.40}$	$61.27_{\pm 3.64}$	$62.12_{\pm 1.10}$	$67.36_{\pm 0.71}$	55.92
	GraphPrompt (Liu et al., 2023b)	$26.87_{\pm 1.47}$	$62.58_{\pm 1.84}$	$62.45_{\pm 1.52}$	$62.41_{\pm 0.69}$	$68.03_{\pm 0.78}$	56.47
	ALL-in-one (Sun et al., 2023a)	$25.73_{\pm 1.18}$	$65.16_{\pm 1.47}$	$58.52_{\pm 1.59}$	$62.01_{\pm 0.66}$	$64.43_{\pm 1.00}$	55.17
	GPF (Fang et al., 2023)	$28.53_{\pm 1.76}$	$65.64_{\pm 0.70}$	$61.45_{\pm 3.13}$	$61.90_{\pm 1.26}$	$67.19_{\pm 0.74}$	56.94
	GPF-plus (Fang et al., 2023)	$27.33_{\pm 2.01}$	$67.20_{\pm 1.56}$	$61.61_{\pm 2.89}$	$62.84_{\pm 0.23}$	$67.69_{\pm 0.64}$	57.33
	EdgePrompt (Fu et al., 2025)	$29.33_{\pm 2.30}$	$63.97_{\pm 2.14}$	$62.02_{\pm 3.02}$	$62.02_{\pm 1.03}$	$67.55_{\pm 0.85}$	56.98
	EdgePrompt+ (Fu et al., 2025)	$32.67_{\pm 2.53}$	$67.72_{\pm 1.62}$	**$67.07_{\pm 1.96}$**	**$66.53_{\pm 1.30}$**	$68.31_{\pm 1.36}$	60.46
EP-GPPT	Classifier Only	$29.08_{\pm 1.35}$	$62.12_{\pm 2.82}$	$56.85_{\pm 4.35}$	$62.27_{\pm 0.78}$	$66.30_{\pm 1.78}$	55.32
	GraphPrompt (Liu et al., 2023b)	$26.67_{\pm 1.60}$	$61.61_{\pm 1.91}$	$58.77_{\pm 0.97}$	$62.16_{\pm 0.89}$	$66.37_{\pm 1.17}$	55.12
	ALL-in-one (Sun et al., 2023a)	$24.92_{\pm 1.33}$	$63.61_{\pm 2.12}$	$59.14_{\pm 2.12}$	$59.70_{\pm 1.37}$	$64.86_{\pm 1.60}$	54.45
	GPF (Fang et al., 2023)	$28.33_{\pm 1.73}$	$63.48_{\pm 2.08}$	$58.14_{\pm 4.16}$	$62.52_{\pm 1.39}$	$66.10_{\pm 0.96}$	55.71
	GPF-plus (Fang et al., 2023)	$29.25_{\pm 1.30}$	$66.92_{\pm 2.34}$	$62.93_{\pm 3.23}$	$64.13_{\pm 1.42}$	$67.57_{\pm 1.45}$	58.16
	EdgePrompt (Fu et al., 2025)	$28.33_{\pm 3.41}$	$64.03_{\pm 2.26}$	$59.85_{\pm 3.15}$	$62.98_{\pm 1.44}$	$66.36_{\pm 1.22}$	56.31
	EdgePrompt+ (Fu et al., 2025)	$32.75_{\pm 2.26}$	$66.16_{\pm 1.60}$	$63.58_{\pm 2.07}$	$65.15_{\pm 1.60}$	$68.35_{\pm 1.57}$	59.20
EP-GraphPrompt	Classifier Only	$31.33_{\pm 3.22}$	$62.58_{\pm 2.40}$	$62.09_{\pm 2.31}$	$60.19_{\pm 1.71}$	$65.13_{\pm 0.81}$	55.32
	GraphPrompt (Liu et al., 2023b)	$30.20_{\pm 1.93}$	$64.72_{\pm 1.98}$	$62.57_{\pm 1.45}$	$62.32_{\pm 0.95}$	$65.85_{\pm 0.65}$	57.13
	ALL-in-one (Sun et al., 2023a)	$29.07_{\pm 1.16}$	$65.60_{\pm 2.38}$	$58.67_{\pm 2.42}$	$57.69_{\pm 1.08}$	$64.66_{\pm 0.76}$	55.14
	GPF (Fang et al., 2023)	$30.93_{\pm 1.76}$	$66.21_{\pm 1.66}$	$61.80_{\pm 2.78}$	$62.27_{\pm 1.18}$	$65.61_{\pm 0.59}$	57.36
	GPF-plus (Fang et al., 2023)	$30.67_{\pm 3.06}$	$67.50_{\pm 2.45}$	$62.59_{\pm 2.09}$	$61.98_{\pm 1.60}$	$65.51_{\pm 1.10}$	57.65
	EdgePrompt (Fu et al., 2025)	$30.80_{\pm 2.09}$	$65.87_{\pm 1.35}$	$61.75_{\pm 2.49}$	$62.33_{\pm 1.65}$	$65.77_{\pm 0.90}$	57.30
	EdgePrompt+ (Fu et al., 2025)	$33.27_{\pm 2.71}$	$67.47_{\pm 2.14}$	$65.06_{\pm 1.84}$	$64.64_{\pm 1.57}$	$66.42_{\pm 1.31}$	59.37
cross-domain	GCN (Kipf & Welling, 2017)	$43.33_{\pm 1.05}$	$65.84_{\pm 2.77}$	$61.36_{\pm 2.00}$	$62.17_{\pm 0.66}$	$60.46_{\pm 1.75}$	58.63
	BRIDGE (Yuan et al., 2025b)	$36.67_{\pm 5.96}$	$64.95_{\pm 3.38}$	$63.50_{\pm 2.27}$	$61.78_{\pm 1.63}$	$65.12_{\pm 2.83}$	58.40
	GFT (Wang et al., 2024b)	$34.61_{\pm 3.12}$	$56.00_{\pm 1.77}$	$59.16_{\pm 6.25}$	$60.50_{\pm 2.71}$	$67.82_{\pm 3.18}$	55.61
	RiemannGFM (Sun et al., 2025)	$34.27_{\pm 1.72}$	$68.74_{\pm 1.31}$	$55.10_{\pm 2.24}$	$59.86_{\pm 1.30}$	$62.56_{\pm 4.04}$	56.11
GraphVec		**$51.00_{\pm 3.22}$**	**$75.94_{\pm 2.70}$**	**$67.32_{\pm 1.51}$**	**$67.90_{\pm 1.67}$**	**$68.57_{\pm 1.62}$**	**66.14**
Unsupervised GraphVec		$48.33_{\pm 2.36}$	**$74.02_{\pm 1.26}$**	$66.11_{\pm 2.30}$	$64.34_{\pm 2.38}$	**$68.38_{\pm 2.88}$**	**64.23**
GraphVec w/o mean alignment		**$49.33_{\pm 1.48}$**	$73.14_{\pm 1.16}$	$65.80_{\pm 1.40}$	$65.21_{\pm 2.05}$	$67.00_{\pm 1.31}$	64.09

Table 8: 5-shot node classification results on node-level tasks

Methods	**Cora**	**CiteSeer**	**Pubmed**	**ogbn-arxiv**
GPPT (Sun et al., 2022)	41.28 ± 6.24	35.32 ± 1.27	53.41 ± 3.99	17.73 ± 1.66
GraphPrompt (Liu et al., 2023b)	31.65 ± 3.33	26.98 ± 1.24	44.18 ± 5.57	16.11 ± 1.42
ALL-in-one (Sun et al., 2023a)	31.57 ± 2.86	29.76 ± 1.53	46.89 ± 5.35	17.89 ± 1.21
GPF (Fang et al., 2023)	37.56 ± 3.81	29.74 ± 1.73	48.16 ± 3.32	17.64 ± 1.18
GPF-plus (Fang et al., 2023)	28.87 ± 3.18	26.65 ± 1.91	43.02 ± 4.59	17.39 ± 1.27
EdgePrompt (Fu et al., 2025)	37.26 ± 4.53	29.83 ± 1.01	45.49 ± 3.27	17.82 ± 1.59
EdgePrompt + (Fu et al., 2025)	56.41 ± 3.62	43.49 ± 2.62	61.51 ± 4.91	17.78 ± 2.12
GraphVec	$\mathbf{58.66 \pm 1.51}$	$\mathbf{45.45 \pm 1.26}$	$\mathbf{65.72 \pm 2.43}$	$\mathbf{23.75 \pm 1.39}$

Table 9: Accuracy–efficiency trade-off of Nyström approximation on Letter-med.

# Nyström samples	100	1000	2000	4000
Wall-clock time (s)	34.82	48.57	63.55	716.61
Accuracy (%)	81.50 ± 2.21	82.47 ± 0.92	84.27 ± 1.10	84.33 ± 1.60

F.5 Impact of Number of pretraining Datasets

Figure 2 shows the change of classification accuracy when the number of datasets used in pretraining increases from 1 to 4. We can see that with more datasets used in pretraining, the performance in downstream tasks becomes better. This result indicates that the generalization ability of graph embeddings generated by our GraphVec can benefit from the increase in the number of training datasets, which is an important capability for cross domain pretrained graph model. It can also be observed that even using model pretrained on only 1 dataset, GraphVec still outperforms other baselines shown in Table 1.

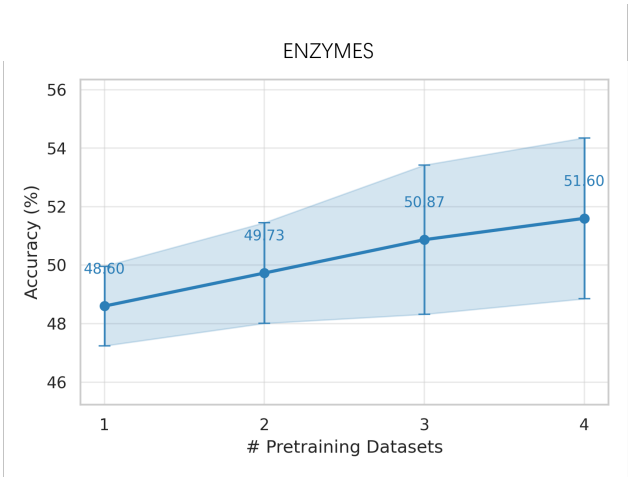

Figure 2: The change of classification accuracy in ENZYMES when the number of datasets used in pretraining increases from 1 to 4.

F.6 Few-shot Learning with Fewer Labeled Samples

As shown in Figure 3, the classification accuracy of our method GraphVec increases as the number of labeled samples increases. Our GraphVec with 20-shot even outperforms the competitors with 50-shot in Table 1 of the main paper.

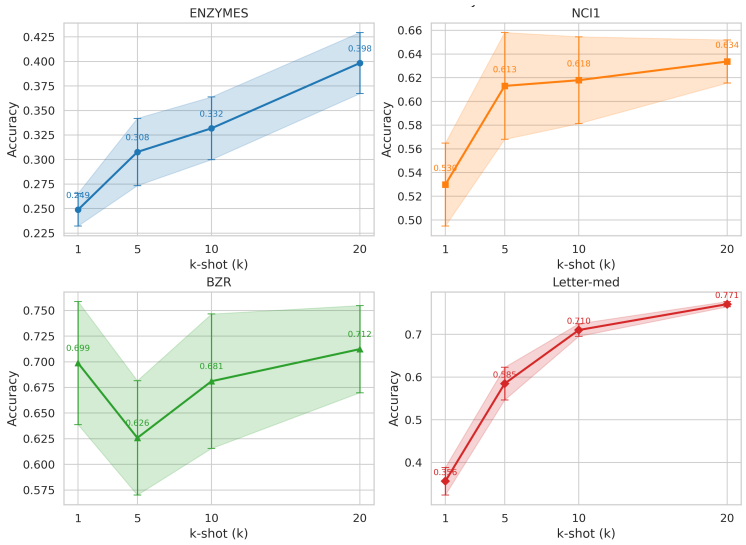

Figure 3: Classification accuracy trends of our method GraphVec with varying k values in few-shot learning across four datasets (PROTEINS, NCI109, DD, and Mutagenicity), shaded area represents standard deviation

F.7 Comparison between Original Attributes and Similarity-based Features

Table 10: Supervised graph classification accuracy of a standard GIN using original node attributes and our similarity-based features. The train/validation/test split is 8/1/1.

Feature Type	PROTEINS	NCI1	NCI109	ENZYMES
Original node attributes	72.81 ± 4.08	78.88 ± 1.50	81.60 ± 1.62	66.33 ± 6.09
Similarity-based features	71.43 ± 2.33	80.00 ± 1.99	78.31 ± 2.24	67.00 ± 5.81

Table 10 shows that replacing the original attributes with the features produced by our global multi-graph construction leads to comparable supervised performance on datasets with semantically meaningful node attributes. The transformed features slightly outperform the raw attributes on NCI1 and ENZYMES, while remaining competitive on PROTEINS and NCI109. This result suggests that the proposed feature construction does not simply discard useful information; rather, it preserves a substantial portion of the task-relevant signal while converting heterogeneous attributes into a more transferable relational representation. In these experiments, mean alignment is applied to remove the sign ambiguity induced by SVD.

F.8 Comparison with Backbone-only Variants

Table 11: 50-shot cross-dataset graph classification accuracy on the five biochemical datasets. GraphVec denotes the full model, while GIN and GT are trained under the same evaluation protocol.

Method	ENZYMES	DD	NCI1	NCI109	Mutagenicity
GIN	43.53	60.53	59.45	60.84	64.34
GT	44.53	65.52	54.43	55.20	61.48
GraphVec	51.00	75.94	67.32	67.90	68.57

Table 11 addresses a potential confound: whether the performance gain mainly comes from the expressive backbone rather than from the proposed pretraining and alignment design. The answer is negative. Although GraphVec is built on top of GIN and graph transformer components, the full model consistently outperforms either backbone alone on all five datasets, with particularly

large margins on DD, NCI1, and NCI109. Therefore, the improvement cannot be attributed only to backbone choice; the multi-graph feature construction, mean alignment, and reference distribution module contribute materially to the final performance.

F.9 Comparison with Simpler Cross-domain Feature Alternatives

Table 12: Cross-domain feature ablation on computer vision datasets. All models use the same mean alignment, backbone, reference layers, and training strategy; only the input feature construction is changed.

Feature Type	Letter-med	COIL-RAG	Cuneiform
Original node attributes	55.33 ± 1.78	58.12 ± 1.97	33.56 ± 3.52
PCA features	56.73 ± 1.93	10.29 ± 0.45	28.05 ± 7.07
Global multi-graph features	85.60 ± 1.44	74.20 ± 0.77	55.86 ± 8.15

Table 12 further shows that the advantage of GraphVec is not explained by dimensional alignment alone. When the global multi-graph features are replaced by zero-padded original attributes or PCA-based features, while keeping all other components unchanged, cross-domain performance drops substantially on all three computer vision datasets. The degradation is especially severe for PCA on COIL-RAG and Cuneiform, indicating that simple linear dimension reduction does not yield a transferable relational space. In contrast, the proposed global multi-graph construction provides much stronger cross-domain invariance, which supports the role of kernelized sample relationships as the key mechanism for bridging domain-specific feature spaces.

F.10 Evaluation on Generated Node Attributes SVD$(\mathbf{A} + \mathbf{I})$

For the dataset without original node attributes, the classification mainly relies on discriminating between different structures of graphs. While the topologically derived features may not carry explicit domain semantics like chemical properties or pixel coordinates, they can be viewed as a form of generic node attribute derived from the graph connectivity. The core of our methodology, specifically the global multi-graph, is designed to bridge the inherent semantic gaps between different domains, regardless of whether the original features are rich in semantics or purely structural.

To evaluate the impact of the feature generation method , we conducted a controlled experiment to replace truncated SVD with two kinds of node centrality. The results are shown in the Table 13.

Table 13: Evaluation on Generated Node Attributes.

	Degree Centrality	Betweenness Centrality	Degree + Betweenness	Our Method
REDDIT - B	73.95 ± 2.30	74.92 ± 1.68	77.79 ± 2.76	81.52 ± 1.50

As shown in the table, our method demonstrates a clear advantage over using only a single type of centrality, and also achieves a marginal improvement compared to combining both centrality measures. It is worth noting that the attributes generated by our approach can be viewed as a form of structural encoding. While other types of structural encodings may also be effective, the performance gain observed here can be largely attributed to our proposed global multi-graph construction, which enhances the model's ability to capture feature information from different spaces.

F.11 More Graph Clustering Results

Pretraining datasets For all graph-clustering results in Table 3 and Tables 14–15, the target dataset is excluded from GraphVec pretraining. For ENZYMES, the model is pretrained on DD, NCI1, NCI109, and Mutagenicity; for NCI1, it is pretrained on ENZYMES, DD, NCI109, and Mutagenicity. The remaining clustering datasets are not among the five source datasets and are evaluated using the model pretrained on all five.

To further evaluate the performance of GraphVec in graph clustering task, we conduct experiments on 4 more datasets. The results are shown in Table 14. We also provide the full version of Table 3 with NMI in Table 15

Table 14: Graph clustering results on PTC-MM, MUTAG, COX2 and BZR

Dataset	PTC-MM			MUTAG			COX2			BZR		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
GraphCL+SC	62.09 ± 0.56	2.14 ± 0.43	3.36 ± 0.87	73.22 ± 2.66	$\mathbf{32.19} \pm 2.05$	23.44 ± 2.45	75.01 ± 2.12	1.24 ± 0.37	$\mathbf{2.39} \pm 2.28$	72.88 ± 1.66	$\mathbf{1.90} \pm 0.38$	$\mathbf{3.47} \pm 0.59$
GWF (Xu et al., 2022)+SC	53.02 ± 1.66	0.36 ± 0.28	0.21 ± 0.09	73.92 ± 4.30	18.35 ± 3.85	24.48 ± 4.69	58.83 ± 4.46	1.16 ± 0.41	1.45 ± 1.21	52.76 ± 0.80	3.47 ± 1.16	-0.71 ± 0.32
GLCC (Ju et al., 2023)	61.61 ± 0.24	0.63 ± 0.41	1.24 ± 1.38	71.99 ± 3.08	13.18 ± 6.93	16.89 ± 8.28	77.37 ± 1.11	0.02 ± 0.03	-0.30 ± 0.42	63.62 ± 9.79	1.18 ± 0.60	1.12 ± 0.97
Our Method	$\mathbf{65.74} \pm 0.00$	$\mathbf{4.35} \pm 0.00$	$\mathbf{6.31} \pm 0.00$	$\mathbf{81.38} \pm 0.00$	31.00 ± 0.00	$\mathbf{38.96} \pm 0.00$	$\mathbf{78.58} \pm 0.00$	$\mathbf{2.37} \pm 0.00$	2.19 ± 0.00	$\mathbf{77.28} \pm 0.00$	0.16 ± 0.00	1.50 ± 0.00

Table 15: Graph clustering performance on ENZYMES, NCI1, COLLAB, REDDIT-BINARY, REDDIT-MULTI. The comparison numbers are from AMGC (Yang et al., 2025).

Method	ENZYMES			NCI1			COLLAB			REDDIT-BINARY			REDDIT-MULTI		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
RW +SC	$17.0_{\pm 0.0}$	$0.7_{\pm 0.0}$	$0.3_{\pm 0.0}$	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
WL +SC	$21.0_{\pm 0.0}$	$3.1_{\pm 0.0}$	$1.5_{\pm 0.0}$	$50.1_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$53.2_{\pm 0.0}$	$2.0_{\pm 0.0}$	$0.5_{\pm 0.0}$	$57.6_{\pm 0.0}$	$9.0_{\pm 0.0}$	$2.2_{\pm 0.0}$	$18.7_{\pm 0.0}$	$9.0_{\pm 0.0}$	$4.0_{\pm 0.0}$
WL-OA +SC	$20.0_{\pm 0.0}$	$1.4_{\pm 0.0}$	$0.3_{\pm 0.0}$	$53.2_{\pm 0.0}$	$0.9_{\pm 0.0}$	$0.8_{\pm 0.0}$	$54.2_{\pm 0.0}$	$0.2_{\pm 0.0}$	$2.6_{\pm 0.0}$	$53.8_{\pm 0.0}$	$5.6_{\pm 0.0}$	$3.8_{\pm 0.0}$	$20.9_{\pm 0.0}$	$9.6_{\pm 0.0}$	$3.2_{\pm 0.0}$
SP +SC	$22.0_{\pm 0.0}$	$2.6_{\pm 0.0}$	$1.7_{\pm 0.0}$	$50.1_{\pm 0.0}$	$0.1_{\pm 0.0}$	$0.0_{\pm 0.0}$	$48.7_{\pm 0.0}$	$17.9_{\pm 0.0}$	$13.9_{\pm 0.0}$	$57.8_{\pm 0.0}$	$2.2_{\pm 0.0}$	$2.2_{\pm 0.0}$	$20.3_{\pm 0.0}$	$6.1_{\pm 0.0}$	$0.1_{\pm 0.0}$
LT +SC	$17.0_{\pm 0.0}$	$0.4_{\pm 0.0}$	$0.0_{\pm 0.0}$	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
GK +SC	$17.1_{\pm 0.1}$	$0.8_{\pm 0.3}$	$0.0_{\pm 0.0}$	$52.9_{\pm 0.9}$	$0.7_{\pm 1.4}$	$0.3_{\pm 0.6}$	$56.8_{\pm 1.4}$	$15.5_{\pm 1.9}$	$9.3_{\pm 2.1}$	$50.3_{\pm 0.3}$	$0.2_{\pm 0.1}$	$0.0_{\pm 0.0}$	$18.7_{\pm 0.9}$	$7.2_{\pm 0.3}$	$0.3_{\pm 0.1}$
InfoGraph +KM	$22.1_{\pm 1.0}$	$2.4_{\pm 0.5}$	$1.3_{\pm 0.5}$	$54.1_{\pm 2.2}$	$1.3_{\pm 1.1}$	$0.9_{\pm 0.9}$	$59.6_{\pm 1.8}$	$14.4_{\pm 3.0}$	$6.6_{\pm 2.3}$	$51.3_{\pm 2.1}$	$2.3_{\pm 0.4}$	$0.6_{\pm 0.2}$	$20.3_{\pm 0.9}$	$0.5_{\pm 0.2}$	$0.0_{\pm 0.0}$
InfoGraph +SC	$23.8_{\pm 0.5}$	$4.6_{\pm 0.7}$	$2.2_{\pm 0.4}$	$54.9_{\pm 1.7}$	$0.9_{\pm 0.6}$	$1.0_{\pm 0.8}$	$60.9_{\pm 2.5}$	$15.4_{\pm 3.3}$	$9.3_{\pm 3.5}$	$50.8_{\pm 1.3}$	$1.6_{\pm 0.6}$	$0.6_{\pm 0.0}$	$24.7_{\pm 1.3}$	$4.8_{\pm 0.6}$	$3.2_{\pm 0.6}$
GraphCL +KM	$21.5_{\pm 0.2}$	$1.6_{\pm 0.1}$	$0.9_{\pm 0.1}$	$55.4_{\pm 1.7}$	$0.5_{\pm 0.3}$	$1.0_{\pm 0.9}$	$58.0_{\pm 1.2}$	$17.8_{\pm 2.0}$	$11.3_{\pm 0.6}$	$51.9_{\pm 3.3}$	$3.4_{\pm 1.2}$	$0.2_{\pm 0.0}$	$25.3_{\pm 0.9}$	$5.3_{\pm 0.3}$	$4.3_{\pm 0.6}$
GraphCL +SC	$25.3_{\pm 0.3}$	$4.8_{\pm 0.4}$	$2.0_{\pm 0.3}$	$50.8_{\pm 1.6}$	$0.6_{\pm 0.6}$	$1.1_{\pm 0.8}$	$57.8_{\pm 0.6}$	$17.0_{\pm 1.3}$	$10.1_{\pm 0.7}$	$55.9_{\pm 2.1}$	$3.2_{\pm 1.0}$	$0.3_{\pm 0.2}$	$27.3_{\pm 1.3}$	$5.4_{\pm 0.8}$	$4.2_{\pm 1.1}$
JOAO + KM	$21.7_{\pm 0.4}$	$4.9_{\pm 0.4}$	$2.1_{\pm 0.2}$	$51.1_{\pm 0.4}$	$0.4_{\pm 0.2}$	$0.1_{\pm 0.0}$	$58.3_{\pm 1.5}$	$18.7_{\pm 2.6}$	$11.1_{\pm 1.8}$	$54.3_{\pm 2.9}$	$4.2_{\pm 1.8}$	$0.8_{\pm 0.3}$	$26.6_{\pm 0.6}$	$3.6_{\pm 1.2}$	$2.5_{\pm 0.2}$
JOAO + SC	$24.4_{\pm 1.4}$	$3.2_{\pm 0.7}$	$1.7_{\pm 0.8}$	$51.5_{\pm 3.0}$	$0.9_{\pm 1.2}$	$0.4_{\pm 1.2}$	$58.2_{\pm 0.9}$	$17.1_{\pm 2.1}$	$10.6_{\pm 0.8}$	$55.9_{\pm 1.2}$	$6.7_{\pm 2.0}$	$1.4_{\pm 0.6}$	$25.6_{\pm 0.6}$	$2.5_{\pm 0.2}$	$3.4_{\pm 0.3}$
GLCC	$24.4_{\pm 1.4}$	$3.2_{\pm 0.7}$	$1.7_{\pm 0.8}$	$60.9_{\pm 2.3}$	$5.3_{\pm 1.9}$	$3.6_{\pm 2.6}$	$60.3_{\pm 0.6}$	$18.2_{\pm 1.3}$	$12.1_{\pm 0.9}$	$67.6_{\pm 3.4}$	$9.2_{\pm 2.6}$	$8.7_{\pm 1.7}$	$32.4_{\pm 2.1}$	$11.8_{\pm 1.3}$	$8.2_{\pm 1.6}$
AMGC	$26.7_{\pm 2.0}$	$5.2_{\pm 1.3}$	$2.8_{\pm 0.7}$	$62.7_{\pm 3.0}$	$6.4_{\pm 1.9}$	$6.4_{\pm 3.6}$	$61.2_{\pm 1.0}$	$20.5_{\pm 1.6}$	$12.9_{\pm 0.9}$	$64.3_{\pm 1.9}$	$12.1_{\pm 3.3}$	$10.5_{\pm 2.7}$	$35.5_{\pm 2.3}$	$16.1_{\pm 0.9}$	$12.0_{\pm 0.7}$
GraphVec	$29.1_{\pm 0.4}$	$7.7_{\pm 0.3}$	$3.5_{\pm 0.2}$	$64.8_{\pm 0.0}$	$6.5_{\pm 0.0}$	$8.7_{\pm 0.0}$	$61.8_{\pm 0.0}$	$21.2_{\pm 0.0}$	$18.9_{\pm 0.0}$	$71.6_{\pm 0.0}$	$20.7_{\pm 0.0}$	$18.6_{\pm 0.0}$	$40.0_{\pm 0.2}$	$17.3_{\pm 0.1}$	$12.1_{\pm 0.2}$

F.12 Visualization of aligned node embeddings

To qualitatively inspect whether the proposed alignment places node embeddings from heterogeneous datasets into a comparable coordinate system, we visualize the aligned node embeddings from biochemical, social-network, and computer-vision datasets using t-SNE. As shown in Figure 4, the embeddings from different datasets are not separated into completely isolated domain-specific regions. Instead, several datasets occupy overlapping or adjacent regions in the projected space, suggesting that the global multi-graph construction and density-maximization mean alignment reduce the gross distributional mismatch caused by heterogeneous raw attributes and feature dimensions.

The visualization also shows that the aligned space does not collapse all datasets into an indistinguishable distribution. Some datasets still form relatively compact local clusters or dataset-specific bands, which is expected because different domains preserve different structural and semantic regularities. Thus, the goal of the alignment is not to erase domain information, but to map node-level features into a shared relational space where cross-domain training becomes feasible. Since t-SNE is a qualitative projection and can distort global distances, this figure should be viewed as supporting evidence complementary to the downstream classification, clustering, and ablation results.

We further compare the node embeddings learned by GraphVec-FM with those obtained from two representative non-LLM-based GFMs, RiemannGFM and GFT. Figure 5 visualizes node embeddings from NCI109 and IMDB-MULTI, where colors denote datasets and markers denote graph classes. Compared with RiemannGFM and GFT, GraphVec-FM produces embeddings with stronger cross-dataset mixing while still preserving visible class-level structure. This qualitative comparison suggests that the proposed global multi-graph construction and density-based alignment better reduce dataset-specific feature mismatch, rather than merely separating nodes according to their source dataset. As with all t-SNE visualizations, this result is qualitative and should be interpreted together with the quantitative transfer results.

F.13 Scalability to Large-scale Dataset

The construction and decomposition of global multi-graphs can be computationally intensive for large training datasets; the Nyström approximation in Section 3.2 can reduce this cost. During downstream inference, the support spectral bases and alignment matrices are fixed. We control peak memory by

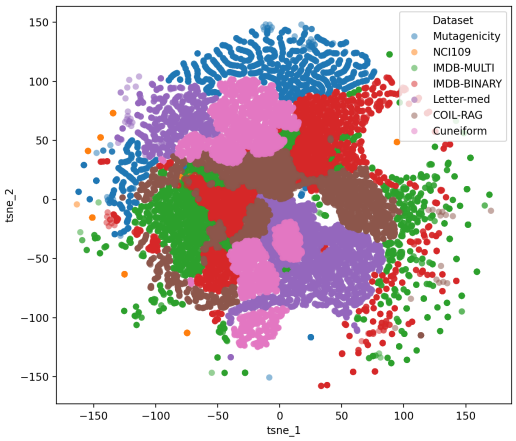

Figure 4: T-SNE visualization of aligned node embeddings of datasets from different domains.

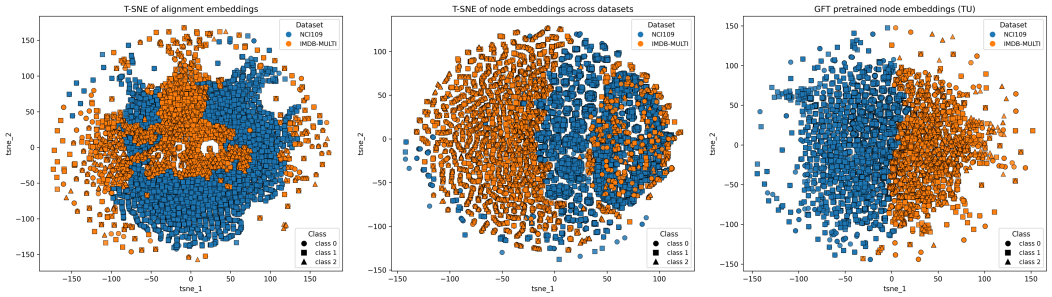

Figure 5: T-SNE visualization of node embeddings obtained by different pretrained models. From left to right are node embeddings obtained from GraphVec-FM, RiemannGFM, and GFT, respectively.

Table 16: Resource consumption for large-scale dataset evaluation. Note that COLOR-3 exhibits higher RAM consumption compared to reddit_threads, primarily due to its larger average number of nodes.

Dataset	**Graphs**	**Avg. Nodes**	**RAM**	**Time**
COLOR-3	10500	61.31	23.17GB	626.84s
reddit_threads	203088	23.93	6.01GB	3139.53s

computing the test-to-training cross-kernel projection in batches using these shared bases, without constructing batch-specific global graphs or recomputing alignment.

To further validate the scale ability during evaluation, we test the time and memory consumption to evaluate on two large datasets COLOR-3 and reddit_threads (Morris et al., 2020) with more than 10k graphs and 20k graphs respectively. For these experiments, query graphs are processed in blocks of 128 using the fixed support bases and the cross-kernel projection in (16). The resulting resource usage is summarized in Table 16, demonstrating that the overhead remains manageable at this scale.

F.14 Ablation Study

To verify the effectiveness of our proposed methods and modules, we conduct ablation study on global multi-graph construction, mean alignment algorithm, and reference layer. For the global multi-graph, we vary the number of multi-graphs from 1-6. Figure 6 and Figure 7 demonstrate the impact of the number of kernel parameters on downstream few-shot graph classification accuracy. These results were obtained by incrementally increasing the set of Gaussian kernel bandwidths from $[0.25]$ to the full set $[0.25, 0.5, 1, 2, 5, 10]$ used in our main experiments. It can be observed

that classification accuracy improves with a greater number of global multi-graphs, particularly for datasets with original continuous node attributes. This observation further illustrates that the global multi-graphs constructed by using different kernel parameters help capture patterns from original features.

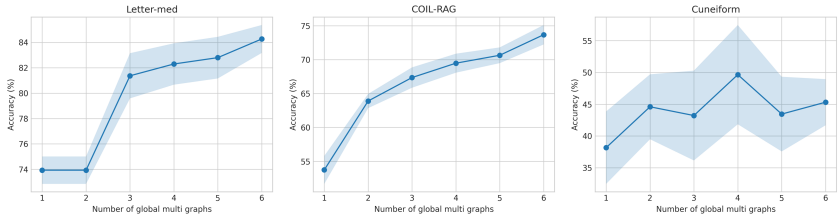

Figure 6: The few-shot graph classification accuracy in datasets with node attributes when the number of global multi-graphs increases from 1 to 6.

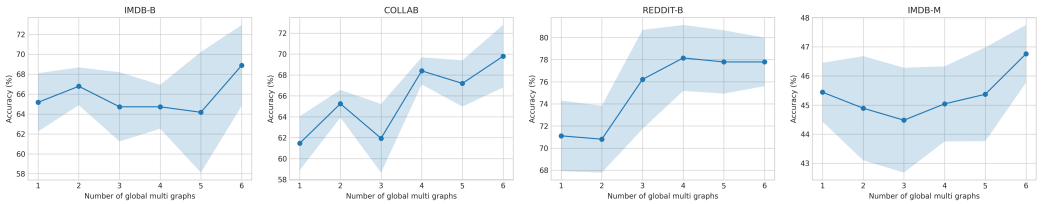

Figure 7: The few-shot graph classification accuracy in datasets without node attributes when the number of global multi-graphs increases from 1 to 6.

We conducted experiments using simple pooling without reference layers, and the results are presented in Table 17. After removing the reference layer module, the performance on all datasets shows degradation, especially on COLLAB, IMDB-BINARY, and Cuneiform. By removing the mean alignment, the performance on COLLAB, Letter-Med, and Cuneiform shows an evident decrease. Similarly, we also conduct experiments that remove the alignment module and both the alignment module and the reference layer. The overall impact of the two modules is shown in Table 18.

Table 17: Ablation study of reference layer and mean alignment module.

Model Variant	Dataset						
	COLLAB (50-shot)	REDDIT-B (50-shot)	IMDB-B (50-shot)	IMDB-M (50-shot)	Letter-med (50-shot)	COIL-RAG (5-shot)	Cuneiform (1-shot)
Original Model	**68.09** $\pm$ 2.99	**81.52** $\pm$ 1.50	**68.39** $\pm$ 4.06	**46.70** $\pm$ 0.99	**85.60** $\pm$ 1.44	**74.20** $\pm$ 0.77	**55.86** $\pm$ 8.15
Mean Readout Only	65.04 $\pm$ 2.75	77.79 $\pm$ 3.11	61.78 $\pm$ 2.59	46.07 $\pm$ 2.60	83.17 $\pm$ 0.99	72.87 $\pm$ 1.26	41.04 $\pm$ 3.64
w/o alignment	64.90 $\pm$ 1.81	76.74 $\pm$ 5.43	66.06 $\pm$ 4.44	45.78 $\pm$ 1.46	81.87 $\pm$ 1.31	72.14 $\pm$ 0.35	42.87 $\pm$ 5.18

Table 18: The individual effect of alignment algorithm and reference layer on downstream cross-domain graph classification. The reported performance is averaged on 7 datasets.

Alignment	Reference layer	Avg. Acc. (%)
✓	✓	68.62
✓	×	63.95
×	✓	64.37
×	×	63.16

F.15 Kernel Function Analysis

We further study the effect of different kernel functions in the multi-graph construction module. Specifically, we compare the Laplacian kernel, polynomial kernel, and RBF kernel for constructing the global graphs over node attributes. As shown in Table 19, the RBF kernel achieves the best or

Table 19: Analysis of different kernel functions used in the multi-graph construction module. The RBF kernel is used as the default choice in GraphVec.

Kernel	COLLAB	IMDB-BINARY	IMDB-MULTI	REDDIT-BINARY	Letter-med	COIL-RAG	Cuneiform
Laplacian	65.28 ± 1.38	69.83 ± 1.79	46.33 ± 2.09	86.03 ± 3.23	83.47 ± 2.05	74.74 ± 0.96	47.00 ± 4.92
Polynomial	56.33 ± 3.81	60.56 ± 2.06	40.56 ± 3.24	71.53 ± 4.65	82.57 ± 1.42	62.55 ± 2.53	41.43 ± 4.79
RBF	68.09 ± 2.99	68.39 ± 4.06	46.70 ± 0.99	81.52 ± 1.50	85.60 ± 1.44	74.20 ± 0.77	55.86 ± 8.15

competitive performance on most datasets, including COLLAB, IMDB-MULTI, Letter-med, and Cuneiform. Although the Laplacian kernel performs better on IMDB-BINARY, REDDIT-BINARY, and COIL-RAG, its performance drops substantially on Cuneiform compared with the RBF kernel. The polynomial kernel performs consistently worse than the other two kernels, suggesting that its induced similarity may be less suitable for capturing local node-attribute relationships across heterogeneous graph domains.

Overall, the RBF kernel provides the most stable performance across datasets. This is consistent with the motivation of our multi-graph construction strategy: the RBF kernel captures smooth local similarity in the node attribute space and is less dependent on the absolute scale or linear structure of the original attributes. Therefore, we use the RBF kernel as the default kernel function in GraphVec.

F.16 Comparison with Other Alignment Approaches

We further compare our density-maximization mean alignment with the alignment strategy adopted in AnyGraph under the same backbone and evaluation protocol. Although both methods rely on SVD-based representations, the two alignment mechanisms are fundamentally different. AnyGraph applies SVD directly to original node attributes and is mainly designed to handle heterogeneous feature dimensions, whereas our method first constructs kernelized global graphs and then aligns the resulting relational embeddings by explicitly maximizing the density of dataset means under orthogonal transformations.

The results in Table 20 show that our alignment strategy improves performance on all six cross-domain benchmarks. The gain is modest on IMDB-MULTI, but it becomes substantial on datasets with larger domain gaps, especially the computer vision datasets Letter-med, COIL-RAG, and Cuneiform. In particular, the large margin on COIL-RAG and Cuneiform suggests that directly aligning raw attribute spaces is not sufficient when datasets differ strongly in feature semantics, while our density-based alignment is better at matching datasets in a shared relational space. These results provide empirical support that the proposed alignment module is an important contributor to the transferability of GraphVec across domains.

Table 20: Comparison between the alignment strategy used in AnyGraph and our density-maximization mean alignment. Our alignment consistently improves cross-domain graph classification performance, especially on computer vision graph datasets.

Alignment Method	COLLAB	IMDB-BINARY	IMDB-MULTI	Letter-med	COIL-RAG	Cuneiform
AnyGraph's Alignment	62.93 ± 2.27	63.61 ± 2.92	45.33 ± 1.93	72.17 ± 1.25	29.05 ± 3.35	10.46 ± 2.09
Our Alignment	68.09 ± 2.99	68.39 ± 4.06	46.70 ± 0.99	85.60 ± 1.44	74.20 ± 0.77	55.86 ± 8.15

F.17 Hyperparameter Analysis

We analyze the sensitivity of GraphVec with respect to five key hyperparameters: the contrastive temperature ϱ, the SVD embedding dimension $\bar{d}$, the alignment bandwidth γ_{A}, the number of reference distributions R, and the backbone depths $(l_{\mathrm{GIN}}, l_{\mathrm{GT}})$. Specifically, ϱ controls the sharpness of the supervised contrastive objective, $\bar{d}$ denotes the dimensionality of the SVD-based node embeddings used in the multi-graph feature alignment module, γ_{A} controls the Gaussian weighting in the density-maximization alignment, while the reference-distribution RBF parameter γ is learned during pretraining, R is the number of learnable reference distributions in the reference distribution module, and l_{GIN} and l_{GT} denote the numbers of GIN and graph transformer layers, respectively.

Table 21: Hyperparameter sensitivity analysis of the contrastive temperature ϱ. The underlined column denotes the default setting used in our experiments.

Dataset	$\varrho = 0.05$	$\underline{\varrho = 0.1}$	$\varrho = 0.15$	$\varrho = 0.2$	$\varrho = 0.5$
IMDB-MULTI	46.93 ± 0.83	46.70 ± 0.99	46.78 ± 1.11	48.11 ± 0.55	46.81 ± 1.17
IMDB-BINARY	67.83 ± 1.53	68.39 ± 4.06	68.22 ± 3.15	67.06 ± 2.01	68.89 ± 4.09
COLLAB	66.10 ± 2.35	68.09 ± 2.99	68.06 ± 1.32	66.63 ± 0.96	66.46 ± 2.19
Letter-med	85.37 ± 0.73	85.60 ± 1.44	85.20 ± 0.83	85.73 ± 0.98	86.03 ± 1.62
REDDIT-BINARY	86.19 ± 2.29	81.52 ± 1.50	86.06 ± 1.29	84.14 ± 2.24	86.71 ± 1.13
COIL-RAG	74.35 ± 0.46	74.20 ± 0.77	72.86 ± 1.10	71.11 ± 1.45	73.97 ± 1.36
Cuneiform	51.73 ± 3.25	55.86 ± 8.15	50.13 ± 4.97	48.95 ± 5.25	51.48 ± 3.65

Table 22: Hyperparameter sensitivity analysis of the SVD embedding dimension $\bar{d}$. The underlined column denotes the default setting used in our experiments.

Dataset	$\bar{d} = 8$	$\bar{d} = 16$	$\underline{\bar{d} = 32}$	$\bar{d} = 64$	$\bar{d} = 128$
IMDB-MULTI	46.00 ± 1.97	47.37 ± 1.06	46.70 ± 0.99	45.37 ± 1.40	43.30 ± 3.93
IMDB-BINARY	67.06 ± 2.60	65.72 ± 1.24	68.39 ± 4.06	68.56 ± 1.85	68.00 ± 1.38
COLLAB	64.64 ± 1.96	66.74 ± 1.51	68.09 ± 2.99	67.92 ± 1.31	69.53 ± 1.37
Letter-med	80.50 ± 2.14	82.10 ± 1.70	85.60 ± 1.44	85.50 ± 1.27	86.60 ± 1.01
REDDIT-BINARY	79.42 ± 1.59	83.01 ± 4.56	81.52 ± 1.50	82.80 ± 2.07	78.75 ± 3.24
COIL-RAG	64.97 ± 1.43	70.72 ± 1.12	74.20 ± 0.77	40.05 ± 29.18	3.37 ± 0.96
Cuneiform	52.91 ± 5.25	51.81 ± 6.39	55.86 ± 8.15	20.17 ± 19.50	6.67 ± 2.04

For the contrastive temperature ϱ, the performance is generally stable across a broad range. The default value $\varrho = 0.1$ achieves competitive results on most datasets, e.g., 68.39 on IMDB-BINARY, 68.09 on COLLAB, 85.60 on Letter-med, and 74.20 on COIL-RAG. Although a few datasets obtain slightly higher scores with other values, such as IMDB-MULTI at $\varrho = 0.2$ and REDDIT-BINARY at $\varrho = 0.5$, the differences are mostly moderate. This indicates that the contrastive objective is not overly sensitive to the temperature, and $\varrho = 0.1$ provides a reasonable default trade-off across datasets.

The embedding dimension $\bar{d}$ has a more pronounced effect. Moderate dimensions, especially $\bar{d} = 32$, perform consistently well and are used as the default setting. Increasing $\bar{d}$ beyond this point does not consistently improve performance and can severely degrade results on some datasets. For example, COIL-RAG drops from 74.20 at $\bar{d} = 32$ to 40.05 at $\bar{d} = 64$ and 3.37 at $\bar{d} = 128$, while Cuneiform drops from 55.86 to 20.17 and 6.67, respectively. This suggests that overly high-dimensional SVD features may introduce noise or instability, particularly on small or structurally sparse datasets. Therefore, $\bar{d} = 32$ is a justified default: it is sufficiently expressive while avoiding the instability observed with larger dimensions.

For γ_{A}, the model is relatively robust within the tested range. In GraphVec, γ_{A} is the hyperparameter in the density-maximization mean alignment objective, which controls the strength of the exponential weighting over the distances between aligned mean embeddings. The default value $\gamma_{\mathrm{A}} = 0.01$ gives strong results on IMDB-BINARY, COLLAB, Letter-med, COIL-RAG, and Cuneiform. Larger values occasionally improve individual datasets, such as COIL-RAG at $\gamma_{\mathrm{A}} = 10.0$ and Cuneiform at $\gamma_{\mathrm{A}} = 1.0$, but they do not yield consistent gains across all datasets. This suggests that using a very large γ_{A} may over-emphasize small pairwise differences among mean embeddings during alignment, making the alignment process more dataset-dependent. The results therefore support using a small default value $\gamma_{\mathrm{A}} = 0.01$, which provides stable cross-domain alignment across heterogeneous graph datasets.

The number of reference distributions R is also not highly sensitive. The default $R = 64$ performs competitively across datasets, achieving 68.09 on COLLAB, 85.60 on Letter-med, 74.20 on COIL-RAG, and 55.86 on Cuneiform. Increasing R to 128 improves some datasets, such as IMDB-BINARY and REDDIT-BINARY, but slightly hurts others such as COLLAB and Letter-med. This suggests that a larger reference set can improve expressiveness, but the gain is dataset-dependent. We therefore choose $R = 64$ as a balanced setting between representation capacity and robustness.

Table 23: Hyperparameter sensitivity analysis of the alignment bandwidth γ_{A}. The underlined column denotes the default setting used in our experiments.

Dataset	$\underline{\gamma_{\mathrm{A}} = 0.01}$	$\gamma_{\mathrm{A}} = 0.1$	$\gamma_{\mathrm{A}} = 0.5$	$\gamma_{\mathrm{A}} = 1.0$	$\gamma_{\mathrm{A}} = 10.0$
IMDB-MULTI	46.70 ± 0.99	44.35 ± 1.20	45.93 ± 2.22	42.22 ± 0.56	46.57 ± 1.02
IMDB-BINARY	68.39 ± 4.06	68.19 ± 0.14	67.78 ± 0.28	68.89 ± 1.94	68.33 ± 0.83
COLLAB	68.09 ± 2.99	63.20 ± 0.21	65.41 ± 2.63	65.77 ± 0.31	66.91 ± 0.93
Letter-med	85.60 ± 1.44	85.17 ± 0.17	85.50 ± 0.50	83.58 ± 1.25	85.50 ± 0.33
REDDIT-BINARY	81.52 ± 1.50	82.63 ± 2.26	84.16 ± 0.72	80.53 ± 2.61	83.82 ± 2.23
COIL-RAG	74.20 ± 0.77	74.19 ± 2.58	73.96 ± 0.65	75.31 ± 1.31	76.27 ± 0.27
Cuneiform	55.86 ± 8.15	52.32 ± 0.00	56.12 ± 1.27	59.92 ± 2.95	55.27 ± 0.42

Table 24: Hyperparameter sensitivity analysis of the number of reference distributions R. The underlined column denotes the default setting used in our experiments.

Dataset	$R = 8$	$R = 16$	$R = 32$	$\underline{R = 64}$	$R = 128$
IMDB-MULTI	45.41 ± 0.75	45.11 ± 2.52	44.59 ± 0.81	46.70 ± 0.99	47.00 ± 1.30
IMDB-BINARY	68.56 ± 1.57	67.17 ± 1.35	67.72 ± 1.82	68.39 ± 4.06	70.39 ± 1.59
COLLAB	66.04 ± 2.08	66.10 ± 2.96	67.09 ± 1.12	68.09 ± 2.99	66.35 ± 0.72
Letter-med	84.67 ± 1.18	84.57 ± 1.51	85.27 ± 1.83	85.60 ± 1.44	85.33 ± 1.48
REDDIT-BINARY	81.92 ± 2.18	82.76 ± 4.63	83.51 ± 1.55	81.52 ± 1.50	84.90 ± 3.41
COIL-RAG	73.03 ± 1.43	74.14 ± 2.46	73.78 ± 1.40	74.20 ± 0.77	74.08 ± 0.65
Cuneiform	53.76 ± 5.30	54.68 ± 3.89	53.67 ± 4.05	55.86 ± 8.15	55.36 ± 2.77

Table 25: Hyperparameter sensitivity analysis of the backbone depths $(l_{\mathrm{GIN}}, l_{\mathrm{GT}})$. The underlined column denotes the default setting used in our experiments.

Dataset	$\underline{l_{\mathrm{GIN}} = 3,\ l_{\mathrm{GT}} = 3}$	$l_{\mathrm{GIN}} = 4,\ l_{\mathrm{GT}} = 3$	$l_{\mathrm{GIN}} = 5,\ l_{\mathrm{GT}} = 3$	$l_{\mathrm{GIN}} = 3,\ l_{\mathrm{GT}} = 4$	$l_{\mathrm{GIN}} = 3,\ l_{\mathrm{GT}} = 5$
IMDB-MULTI	46.70 ± 0.99	46.73 ± 1.22	46.91 ± 1.37	46.98 ± 1.52	46.67 ± 1.32
IMDB-BINARY	68.39 ± 4.06	67.22 ± 2.16	60.56 ± 2.56	69.44 ± 1.18	69.44 ± 1.20
COLLAB	68.09 ± 2.99	62.37 ± 6.01	65.91 ± 2.74	66.43 ± 0.42	68.56 ± 0.72
Letter-med	85.60 ± 1.44	83.78 ± 1.73	81.89 ± 0.48	86.11 ± 0.98	87.28 ± 0.61
REDDIT-BINARY	81.52 ± 1.50	82.70 ± 1.99	84.43 ± 0.56	82.49 ± 2.77	84.24 ± 2.74
COIL-RAG	74.20 ± 0.77	69.44 ± 1.13	65.46 ± 0.38	71.97 ± 0.32	72.41 ± 1.65
Cuneiform	55.86 ± 8.15	49.23 ± 5.60	46.27 ± 1.70	55.84 ± 5.02	53.16 ± 5.21

Finally, we evaluate the backbone depths $(l_{\mathrm{GIN}}, l_{\mathrm{GT}})$. The default setting $(3, 3)$ is competitive, while deeper or more asymmetric configurations provide mixed results. For instance, $(3, 5)$ improves Letter-med and REDDIT-BINARY but degrades COIL-RAG and Cuneiform compared with the default. Similarly, increasing l_{GIN} to 4 or 5 does not consistently improve performance and can substantially hurt COLLAB, COIL-RAG, and Cuneiform. These results suggest that excessive depth may introduce over-smoothing or optimization difficulty, especially under cross-domain transfer. Thus, the default shallow configuration $(l_{\mathrm{GIN}}, l_{\mathrm{GT}}) = (3, 3)$ is a reasonable choice for stable generalization.

Overall, GraphVec is relatively insensitive to ϱ, γ_{A}, and R, while the embedding dimension $\bar{d}$ and backbone depth require moderate settings. The selected default configuration $\varrho = 0.1$, $\bar{d} = 32$, $\gamma_{\mathrm{A}} = 0.01$, $R = 64$, and $(l_{\mathrm{GIN}}, l_{\mathrm{GT}}) = (3, 3)$ provides a robust balance across social network and computer vision datasets, rather than being tuned for a single dataset.

F.18 Robustness Evaluation on Noisy Input Graphs

To validate our model's performance on noisy graph data, we randomly added/deleted 10% edges to 50% of the test graphs during the few-shot test phase, and the results are shown in Table 26. It can be observed that there is only a slight decrease in terms of accuracy when the input graphs are perturbed or noisy, which demonstrates the robustness of our model.

Table 26: Few-shot graph classification performance comparison between original input and perturbed input

Dataset	REDDIT-B 50-shot	IMDB-B 50-shot	IMDB-M 50-shot	Letter-med 50-shot	COIL-RAG 5-shot	Cuneiform 1-shot
Original Graphs	$\mathbf{81.52} \pm 1.50$	$\mathbf{68.39} \pm 4.06$	$\mathbf{46.70} \pm 0.99$	$\mathbf{85.60} \pm 1.44$	$\mathbf{74.20} \pm 0.77$	$\mathbf{55.86} \pm 8.15$
50% Perturbed Graphs	77.68 ± 2.30	65.22 ± 1.07	46.07 ± 1.55	83.47 ± 3.22	72.13 ± 1.93	44.14 ± 3.82

F.19 Time and Memory Consumption

To evaluate the computational cost and runtime of model pretraining, we conducted experiments on two datasets of different scales: the larger deezer_ego_net dataset (9,629 graphs) and the smaller ENZYMES dataset (600 graphs), each for 10 epochs. The wall-clock time and peak GPU/RAM memory usage are presented in the Table 27. We also compared the wall-clock time and memory cost of pretraining with ProNoG (Yu et al., 2025b) and BRIDGE (Yuan et al., 2025b) on the same dataset. It can be observed that our GraphVec requires less training time, especially on relatively large datasets. GraphVec demands more memory consumption, which is primarily due to the computation and storage of the global graph.

Table 27: Performance and resource utilization comparison

	deezer_ego_net			**ENZYMES**		
Method	wall-clock time (s)	GPU Peak Memory (GB)	RAM Peak Memory (GB)	wall-clock time (s)	GPU Peak Memory (GB)	RAM Peak Memory (GB)
ProNoG	1194.61	0.12	1.1	69.54	0.05	0.96
BRIDGE	1600.28	0.31	1.27	116.99	0.18	1.17
Our Method	899.22	1.04	46.42	76.15	0.26	8.35

To ensure fair comparison, since ProNoG and BRIDGE process 4 graphs at once, we set our model's batch size to 4. All experiments are conducted on 14 vCPU Intel(R) Xeon(R) Gold 6348 CPU with one Nvidia A800-80G GPU, CUDA 11.8.

G PROOF FOR THEOREM 3.1

Proof. Define

$$\phi_{ij}(\mathbf{R}) = w_{ij} \exp(2\gamma_{\mathrm{A}} \langle \mathbf{R}_i \boldsymbol{\mu}_i, \mathbf{R}_j \boldsymbol{\mu}_j \rangle) \,. \tag{44}$$

Since every $\mathbf{R}_i$ is orthogonal, we have

$$0 < \phi_{ij}(\mathbf{R}) = \exp\bigl(-\gamma_{\mathrm{A}} \|\mathbf{R}_i \boldsymbol{\mu}_i - \mathbf{R}_j \boldsymbol{\mu}_j\|_2^2\bigr) \le 1. \tag{45}$$

The Euclidean gradient of $\mathcal{L}$ with respect to $\mathbf{R}_i$ is

$$\nabla_{\mathbf{R}_i} \mathcal{L}(\mathbf{R}) = 4\gamma_{\mathrm{A}} \sum_{j=1}^{M} \phi_{ij}(\mathbf{R}) \mathbf{R}_j \boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top. \tag{46}$$

We first bound the Lipschitz constant of $\nabla\mathcal{L}$. For two feasible points $\mathbf{R}$ and $\mathbf{S}$, let

$$\delta_i = \|\mathbf{R}_i - \mathbf{S}_i\|_F, \qquad D = \left(\sum_{i=1}^{M} \delta_i^2 \right)^{1/2}. \tag{47}$$

By the mean-value theorem and the bound $\phi_{ij}(\mathbf{R}) \le 1$,

$$|\phi_{ij}(\mathbf{R}) - \phi_{ij}(\mathbf{S})| \le 2\gamma_{\mathrm{A}} \nu^2 (\delta_i + \delta_j). \tag{48}$$

Therefore,

$$\begin{aligned}
&\|\nabla_{\mathbf{R}_i} \mathcal{L}(\mathbf{R}) - \nabla_{\mathbf{R}_i} \mathcal{L}(\mathbf{S})\|_F \\
&\le 4\gamma_{\mathrm{A}} \sum_{j=1}^{M} \left[|\phi_{ij}(\mathbf{R}) - \phi_{ij}(\mathbf{S})| \|\mathbf{R}_j \boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top\|_F + \phi_{ij}(\mathbf{S}) \|(\mathbf{R}_j - \mathbf{S}_j) \boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top\|_F \right] \\
&\le 4\gamma_{\mathrm{A}} \sum_{j=1}^{M} \left[2\gamma_{\mathrm{A}} \nu^4 (\delta_i + \delta_j) + \nu^2 \delta_j \right] \\
&\le \left(16\gamma_{\mathrm{A}}^2 \nu^4 M + 4\gamma_{\mathrm{A}} \nu^2 \sqrt{M} \right) D.
\end{aligned} \tag{49}$$

Taking the Frobenius norm over all blocks gives

$$\begin{aligned}
\|\nabla\mathcal{L}(\mathbf{R}) - \nabla\mathcal{L}(\mathbf{S})\|_F &= \left(\sum_{i=1}^{M} \|\nabla_{\mathbf{R}_i} \mathcal{L}(\mathbf{R}) - \nabla_{\mathbf{R}_i} \mathcal{L}(\mathbf{S})\|_F^2 \right)^{1/2} \\
&\le \left(16\gamma_{\mathrm{A}}^2 \nu^4 M^{3/2} + 4\gamma_{\mathrm{A}} \nu^2 M \right) \|\mathbf{R} - \mathbf{S}\|_F.
\end{aligned} \tag{50}$$

Thus $\nabla\mathcal{L}$ is L_∇-Lipschitz continuous with

$$L_\nabla = 16\gamma_{\mathrm{A}}^2 \nu^4 M^{3/2} + 4\gamma_{\mathrm{A}} \nu^2 M. \tag{51}$$

By the smoothness of $\mathcal{L}$, we have

$$\begin{aligned}
\mathcal{L}(\mathbf{R}^{(t)}) \ge \mathcal{L}(\mathbf{R}^{(t-1)}) &+ \sum_{i=1}^{M} \left\langle \nabla_{\mathbf{R}_i} \mathcal{L}(\mathbf{R}^{(t-1)}), \mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)} \right\rangle \\
&- \frac{L_\nabla}{2} \sum_{i=1}^{M} \|\mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)}\|_F^2.
\end{aligned} \tag{52}$$

Let

$$\mathbf{A}_i^{(t-1)} = \sum_{j=1}^{M} w_{ij} k_{ij}^{(t-1)} \mathbf{R}_j^{(t-1)} \boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top. \tag{53}$$

Then

$$\nabla_{\mathbf{R}_i} \mathcal{L}(\mathbf{R}^{(t-1)}) = 4\gamma_{\mathrm{A}} \mathbf{A}_i^{(t-1)}. \tag{54}$$

Algorithm 1 computes

$$\mathbf{R}_i^{(t)} = \arg\max_{\mathbf{R}_i^\top \mathbf{R}_i = \mathbf{I}} \left\langle \mathbf{A}_i^{(t-1)} + \eta \mathbf{R}_i^{(t-1)}, \mathbf{R}_i \right\rangle. \tag{55}$$

Hence,

$$\left\langle \mathbf{A}_i^{(t-1)} + \eta \mathbf{R}_i^{(t-1)}, \mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)} \right\rangle \geq 0. \tag{56}$$

Since both $\mathbf{R}_i^{(t)}$ and $\mathbf{R}_i^{(t-1)}$ are orthogonal,

$$\left\langle \mathbf{R}_i^{(t-1)}, \mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)} \right\rangle = -\frac{1}{2} \|\mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)}\|_F^2. \tag{57}$$

Therefore,

$$\left\langle \mathbf{A}_i^{(t-1)}, \mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)} \right\rangle \geq \frac{\eta}{2} \|\mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)}\|_F^2. \tag{58}$$

Substituting this into the smoothness inequality gives

$$\mathcal{L}(\mathbf{R}^{(t)}) \geq \mathcal{L}(\mathbf{R}^{(t-1)}) + \left(2\gamma_{\mathrm{A}} \eta - \frac{L_\nabla}{2} \right) \sum_{i=1}^{M} \|\mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)}\|_F^2. \tag{59}$$

Since $\eta > L_\nabla / (4\gamma_{\mathrm{A}})$, we have

$$2\gamma_{\mathrm{A}} \eta - \frac{L_\nabla}{2} > 0. \tag{60}$$

Thus $\mathcal{L}(\mathbf{R}^{(t)})$ is non-decreasing.

Moreover, because $0 < \phi_{ij}(\mathbf{R}) \leq 1$,

$$0 < \mathcal{L}(\mathbf{R}) \leq M^2. \tag{61}$$

Hence $\{\mathcal{L}(\mathbf{R}^{(t)})\}_{t \geq 0}$ converges. Summing the ascent inequality over t gives

$$\sum_{t=1}^{\infty} \sum_{i=1}^{M} \|\mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)}\|_F^2 < \infty. \tag{62}$$

Therefore,

$$\sum_{i=1}^{M} \|\mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)}\|_F^2 \to 0. \tag{63}$$

Finally, the feasible set is compact because it is a product of orthogonal groups. Hence the sequence has accumulation points. Let $\mathbf{R}^\star$ be any accumulation point. Since

$$\mathbf{R}_i^{(t)} - \mathbf{R}_i^{(t-1)} \to \mathbf{0}, \tag{64}$$

the optimality condition of the Procrustes update passes to the limit and gives

$$\mathbf{R}_i^\star = \arg\max_{\mathbf{R}_i^\top \mathbf{R}_i = \mathbf{I}} \langle \mathbf{A}_i^\star + \eta \mathbf{R}_i^\star, \mathbf{R}_i \rangle. \tag{65}$$

This implies the first-order stationarity condition for the original constrained maximization problem. Therefore, every accumulation point is stationary. □

H Final Generalization Guarantee and Discussion

H.1 End-to-end theorem and proof

The proof follows the same sequence as the GraphVec pipeline: we first certify source-domain contrastive generalization, then transfer this guarantee to the unseen target domain, convert target semantic contrastive risk into the risk of a feasible linear classifier, and finally control the classifier learned from the finite target support set. The supporting results are organized in exactly this order in Appendix I.

Theorem H.1 (GraphVec contrastive transfer followed by few-shot classification)**.** *Fix a target domain $\mathbb{P}_T$, independent of the source-training records, and a mode $a \in \{\mathrm{SCL}, \mathrm{UCL}\}$. Suppose:*

1. *The source certificate $\mathcal{C}^a_{\mathrm{pre}}(\delta_{\mathrm{pre}})$ in (102), or another valid certificate for the same deterministic deployed encoder and centered block loss, holds with failure probability δ_{pre}.*
2. *The inductive feature maps and the common-shift condition satisfy Theorem I.15. For UCL, define the augmentation-to-semantic term as in (119); it is not assumed to be zero.*
3. *Target semantic evaluation uses $B \geq 3$ iid fresh labeled graphs with positive class probabilities $\boldsymbol{\pi}$. The target training set follows one of the sampling/weighting schemes of Appendix I.6.*
4. *The comparison features and head class satisfy (130), the trained head satisfies (135), and the source encoder is not refitted using target labels. For a raw-feature head, choose $a_0 > 0$ such that $\sqrt{C_T}/(\varrho a_0) \leq B_{\mathrm{c}}$, with finite ϵ_{rad} in (129).*

Then, with probability at least $1 - \delta_{\mathrm{pre}} - \delta_{\mathrm{c}}$,

$$\begin{aligned}\mathcal{R}^{\mathrm{CE}}_T(\widehat{\mathbf{W}}_{\mathrm{c}}) \leq \frac{U^a_T}{\kappa_T(B, \boldsymbol{\pi})} + \epsilon_{\mathrm{rad}} + 2\mathfrak{G}_{\mathrm{c}} \\ + 2\sqrt{2}B_{\mathrm{c}}(d_{\mathrm{emp}} + d_{\mathrm{pop}}) + \epsilon_{\mathrm{opt}},\end{aligned} \tag{66}$$

where U^a_T is (122), i.e.,

$$U^a_T = \min\left\{ L_B, \mathcal{C}^a_{\mathrm{pre}}(\delta_{\mathrm{pre}}) + \frac{8L_{\mathrm{shift}}(\widehat{\boldsymbol{\omega}})}{\varrho}\sqrt{\frac{\sum_q \mathfrak{a}_q}{1 - e^{-4\gamma_{\mathrm{A}}}}} + \Delta^a_{\mathrm{res}} + \mathbf{1}\{a = \mathrm{UCL}\}\epsilon_{\mathrm{aug}\to\mathrm{sem},T} \right\}. \tag{67}$$

For compact notation, define the complete target-head contribution

$$\epsilon_{\mathrm{head}}(\delta_{\mathrm{c}}) := \epsilon_{\mathrm{rad}} + 2\mathfrak{G}_{\mathrm{c}} + 2\sqrt{2}B_{\mathrm{c}}(d_{\mathrm{emp}} + d_{\mathrm{pop}}) + \epsilon_{\mathrm{opt}}, \tag{68}$$

where $\mathfrak{G}_{\mathrm{c}}$ contains the confidence parameter δ_{c} as in (133). Hence (66) is equivalently $\mathcal{R}^{\mathrm{CE}}_T(\widehat{\mathbf{W}}_{\mathrm{c}}) \leq U^a_T/\kappa_T(B, \boldsymbol{\pi}) + \epsilon_{\mathrm{head}}(\delta_{\mathrm{c}})$. For a head on normalized graph embeddings, use its corresponding comparison features, set $\epsilon_{\mathrm{rad}} = 0$, and require $B_{\mathrm{c}} \geq \sqrt{C_T}/\varrho$ so the comparison head belongs to the class. The target misclassification probability satisfies

$$\mathbb{P}_T(\widehat{y}(G) \neq y) \leq \min\left\{1, \frac{\textit{right side of (66)}}{\log 2}\right\}. \tag{69}$$

Proof. **Step 1: source empirical-to-population generalization.** By the source certificate in (102), on an event $\mathcal{E}_{\mathrm{pre}}$ of probability at least $1 - \delta_{\mathrm{pre}}$,

$$\mathcal{L}^{a,\circ}_{\mathrm{src}}(\widehat{\boldsymbol{\omega}}) \leq \mathcal{C}^a_{\mathrm{pre}}(\delta_{\mathrm{pre}}). \tag{70}$$

The construction of this certificate is given in Appendix I.2: the multi-positive objective is treated as a bounded contrastive block loss, its overlap is handled through a block PAC–Bayes argument, data-dependent source preprocessing is charged explicitly, and the Gibbs certificate is converted to the deployed deterministic encoder.

Step 2: transfer the source guarantee to target contrastive risk. Condition on a realization in $\mathcal{E}_{\mathrm{pre}}$. Applying the alignment-sensitive transfer theorem Theorem I.15 to the deployed encoder and then using (70) gives

$$\mathcal{L}^{a,\circ}_T \leq \mathcal{C}^a_{\mathrm{pre}}(\delta_{\mathrm{pre}}) + \mathcal{E}_{\mathrm{A}} + \Delta^a_{\mathrm{res}}. \tag{71}$$

Step 3: obtain target semantic contrastive risk. For SCL, (71) already controls the semantic target risk. For UCL, Proposition I.16 converts augmentation-defined positives to semantic same-class positives at the additional cost $\epsilon_{\mathrm{aug}\to\mathrm{sem},T}$. Together with $0 \leq \mathcal{L}^{\mathrm{SCL},\circ}_T \leq L_B$, both modes therefore satisfy

$$\mathcal{L}^{\mathrm{SCL},\circ}_T \leq U^a_T, \tag{72}$$

with U^a_T defined in (67).

Step 4: convert semantic contrastive risk to a feasible classifier risk. For the fitted target map, Theorem I.19 applies to fresh iid target batches and gives

$$\mathcal{R}_T^{\mathrm{mean}} \le \frac{\mathcal{L}_T^{\mathrm{SCL},\circ}}{\kappa_T(B,\boldsymbol{\pi})} \le \frac{U_T^a}{\kappa_T(B,\boldsymbol{\pi})}. \tag{73}$$

If the deployed softmax head uses raw graph vectors, Proposition I.20 constructs a feasible classifier in $\mathcal{H}_T$ with population risk at most the last quantity plus ϵ_{rad}. For normalized graph vectors, the mean classifier itself is feasible and $\epsilon_{\mathrm{rad}} = 0$.

Step 5: control the classifier actually learned from the target support set. Let $\mathcal{E}_{\mathrm{c}}$ be the target-head uniform-convergence event of Theorem I.22, which has conditional probability at least $1 - \delta_{\mathrm{c}}$. Substituting the feasible comparison classifier from Step 4 into Theorem I.22 yields

$$\mathcal{R}_T^{\mathrm{CE}}(\widehat{\mathbf{W}}_{\mathrm{c}}) \le \frac{U_T^a}{\kappa_T(B,\boldsymbol{\pi})} + \epsilon_{\mathrm{rad}} + 2\mathfrak{G}_{\mathrm{c}} + 2\sqrt{2}B_{\mathrm{c}}(d_{\mathrm{emp}} + d_{\mathrm{pop}}) + \epsilon_{\mathrm{opt}},$$

which is (66).

Step 6: combine confidence events and convert to classification error. The event $\mathcal{E}_{\mathrm{c}}$ has conditional probability at least $1 - \delta_{\mathrm{c}}$ for every source realization satisfying the stated conditions. Therefore a union bound with $\mathcal{E}_{\mathrm{pre}}$ gives probability at least $1 - \delta_{\mathrm{pre}} - \delta_{\mathrm{c}}$; independence of the two events is unnecessary. Any additional high-probability estimates used to upper-bound map, noise, or Monte Carlo terms must be included in this failure budget. Finally, if softmax misclassifies a graph, the true-class probability satisfies $p_y \le 1/2$, so pointwise $\mathbf{1}\{\widehat{y} \ne y\} \le -\log p_y / \log 2$. Taking expectations and using the trivial upper bound one proves (69). □

Meaning of the result. The result is for a *specified previously unseen target domain*, with its residual shift displayed explicitly. It is not a distribution-free average guarantee over arbitrary new domains. Such an average-domain statement would need a meta-distribution assumption and would count source domains at the corresponding statistical level. Here source graph count controls source estimation, while the target-domain difference is retained in Δ_{res}^a and, for UCL, in $\epsilon_{\mathrm{aug}\to\mathrm{sem},T}$.

What the theorem does not imply. The formula does not guarantee nonvacuity, does not order the two pretraining modes, and does not assert that adding a kernel scale or retaining more spectral components always helps. It does not imply graph-clustering consistency from a classification bound. Small measured training loss, a finite KL, and a monotonic alignment objective are not by themselves sufficient to set the other displayed terms to zero.

Why the proof is nontrivial. The result cannot be obtained by simply adding a source generalization bound to a classifier bound. The proof must preserve the dependence structure of the multi-positive mini-batch loss, account for source and target preprocessing learned from the same graph records used downstream, separate the alignment- controlled part of domain shift from the residual task-relevant discrepancy, and establish an explicit bridge from semantic contrastive risk to the risk of a feasible linear classifier under random mini-batches. These ingredients are proved in the same order as the six steps above in Appendix I.

H.2 Architecture-expanded readable form of the bound

For interpretation and presentation, it is useful to substitute the architecture-dependent quantities into the complete guarantee while keeping only genuinely data-dependent discrepancies as named residual terms. The following display is algebraically the same guarantee as Theorem H.1 and Corollary I.31; it introduces no additional assumption beyond those already stated there and in Propositions I.29 and I.30.

For $a \in \{\mathrm{SCL}, \mathrm{UCL}\}$, define

$$L_B = \log(B-1) + \frac{2}{\varrho}, \qquad S_{\mathrm{eff}} = M \min_{j\in[M]} \left\lfloor \frac{N_j}{b} \right\rfloor, \qquad A_{\mathrm{arch}} = \sqrt{1 + 2LR + B_{\mathrm{p}}^2}. \tag{74}$$

Then, with the same confidence statement as Corollary I.31, the target-domain misclassification probability obeys

$$\begin{aligned}\mathbb{P}_T\big(\widehat{y}(G)\neq y\big)\le \min\Bigg\{1,\frac{1}{\log 2}\Bigg[\frac{1}{\kappa_T(B,\boldsymbol{\pi})}\min\Bigg\{L_B,\ \mathcal{C}^{a}_{\mathrm{pre}}(\delta_{\mathrm{pre}})\\ +\frac{8}{\varrho s_0}\sqrt{\left(2\gamma R\sum_{l=1}^{L}J_l^2+J_{\mathrm{p}}^2\right)\frac{\sum_{q=1}^{Q}\mathfrak{a}_q}{1-e^{-4\gamma_{\mathrm{A}}}}}\\ +\Delta^{a}_{\mathrm{res}}+\mathbf{1}\{a=\mathrm{UCL}\}\epsilon_{\mathrm{aug}\to\mathrm{sem},T}\Bigg\}\\ +\epsilon_{\mathrm{rad}}+8B_{\mathrm{c}}A_{\mathrm{arch}}\sqrt{\frac{C_T}{N^{\mathrm{c}}_{\mathrm{eff}}}}\\ +2\left(\log C_T+2B_{\mathrm{c}}A_{\mathrm{arch}}\right)\sqrt{\frac{\log(2/\delta_{\mathrm{c}})}{2N^{\mathrm{c}}_{\mathrm{eff}}}}\\ +2\sqrt{2}B_{\mathrm{c}}(d_{\mathrm{emp}}+d_{\mathrm{pop}})+\epsilon_{\mathrm{opt}}\Bigg]\Bigg\}.\end{aligned}\tag{75}$$

For a classifier trained on normalized graph vectors, replace A_{arch} by $\sqrt{2}$ and set $\epsilon_{\mathrm{rad}}=0$, subject to the corresponding head-radius condition already stated in Theorem H.1.

The remaining source-pretraining certificate in (75) can itself be written explicitly. Let

$$\overline{r}_a=\min\left\{L_B,\widehat{\mathcal{L}}^{a,\circ}_{\mathrm{src},\widehat{\mathcal{T}}}(\mathsf{Q})+\epsilon_{\mathrm{map,emp}}(\mathsf{Q})\right\},\tag{76}$$

and set

$$\begin{aligned}\mathcal{C}^{a}_{\mathrm{pre}}(\delta_{\mathrm{pre}})=\min\Bigg\{L_B,\,L_B\min\left\{1,\frac{1-\exp\left[-t\overline{r}_a/L_B-\dfrac{\mathrm{KL}(\mathsf{Q}\|\Pi)+\log(1/\delta_{\mathrm{pre}})}{S_{\mathrm{eff}}}\right]}{1-e^{-t}}\right\}\\ +\epsilon_{\mathrm{map,pop}}(\mathsf{Q})+\epsilon_{\mathrm{det}}(\widehat{\boldsymbol{\omega}},\mathsf{Q})\Bigg\}.\end{aligned}\tag{77}$$

This is simply the learned-preprocessing PAC–Bayes certificate followed by the Gibbs-to-deterministic correction. If Monte Carlo is used to estimate the empirical Gibbs risk, its valid upper correction from Appendix I.2.2 is added to $\widehat{\mathcal{L}}^{a,\circ}_{\mathrm{src},\widehat{\mathcal{T}}}$.

For the product diagonal-Gaussian prior/posterior specialization in Proposition I.28,

$$\mathrm{KL}(\mathsf{Q}\|\Pi)=\sum_{h\in\mathcal{H}_{\mathrm{par}}}\left[\frac{\|\widehat{\boldsymbol{\omega}}_h-\boldsymbol{\omega}_{0,h}\|_2^2}{2s_h^2}+\frac{p_h}{2}\left(\frac{\sigma_h^2}{s_h^2}-1+\log\frac{s_h^2}{\sigma_h^2}\right)\right].\tag{78}$$

In particular,

$$\begin{aligned}\mathrm{KL}_{\mathcal{V}}=\frac{\|\widehat{\mathcal{V}}-\mathcal{V}_0\|_F^2}{2s_{\mathcal{V}}^2}\\ +\frac{LRmd}{2}\left(\frac{\sigma_{\mathcal{V}}^2}{s_{\mathcal{V}}^2}-1+\log\frac{s_{\mathcal{V}}^2}{\sigma_{\mathcal{V}}^2}\right),\end{aligned}\tag{79}$$

while the GIN and GT blocks use the same expression with their actual parameter counts p_{GIN} and p_{GT}, whose dependence on $Q,\bar{d}$, depths, widths, and attention parameterization is described in Appendix I.8.1. Thus (75) makes explicit the roles of $Q,\bar{d},\lambda_q,L,R,m,d,\gamma,\gamma_{\mathrm{A}},\varrho$, the GIN/GT parameterization, batch size, source sample sizes, target-shot count, and classifier capacity, while retaining $\Delta^{a}_{\mathrm{res}}$, the learned-map errors, spectral conditioning, and realized parameter displacements as data- and solution-dependent quantities rather than replacing them by worst-case architecture-wide norm products.

Interpretation. Equation (75) separates four effects: (i) source-pretraining fit and complexity through $\mathcal{C}^a_{\mathrm{pre}}$; (ii) cross-domain compatibility through the alignment discrepancy and Δ^a_{res}; (iii) amplification of remaining mismatch through the learned sensitivities J_l, J_{p} and the reference parameters γ, R; and (iv) target few-shot estimation through $N^{\mathrm{c}}_{\mathrm{eff}}$ and B_{c}. The formula is not a monotonic ranking of hyperparameters: changing any architectural quantity also changes the empirical loss, posterior displacement, spectral quantities, residual discrepancy, and sensitivity constants.

H.3 Implications for GraphVec parameters

The expanded bound separates four effects. Source sample sizes and block size enter through S_{eff}; model size and posterior displacement enter through the blockwise KL; kernel scales and spectral truncation affect the map errors and spectral conditioning; the number of reference distributions R, reference layers L, and reference-kernel parameter γ affect both representation sensitivity and classifier feature magnitude; alignment quality enters through $\sum_q \mathfrak{a}_q$; and the number of labeled target graphs enters through $N^{\mathrm{c}}_{\mathrm{eff}}$. These dependencies express tradeoffs rather than monotone hyperparameter rankings, because changing an architectural parameter also changes the fitted empirical risk, posterior location, residual shift, and sensitivity constants.

I SUPPORTING LEMMAS AND PROOFS

Proof roadmap. The supporting results are arranged in the same order as the end-to-end proof:

$$\widehat{\mathcal{L}}^{a,\circ}_{\mathrm{src}} \longrightarrow \mathcal{C}^{a}_{\mathrm{pre}} \longrightarrow U^{a}_{T} \longrightarrow \mathcal{R}^{\mathrm{mean}}_{T} \longrightarrow \mathcal{R}^{\mathrm{CE}}_{T}(\widehat{\mathbf{W}}_{\mathrm{c}}).$$

Step 0 prepares a bounded loss and the common statistical notation. Step 1 constructs the source certificate. Step 2 transfers it to the unseen target through the alignment term and residual discrepancy. Step 3 handles the extra UCL-to-semantic conversion. Step 4 turns semantic contrastive risk into the risk of a feasible linear classifier. Step 5 controls the classifier actually trained from the finite target support set. Technical perturbation and architecture-dependent specializations are collected afterwards so that they do not interrupt this main proof chain.

I.1 STEP 0: STATISTICAL SETUP AND A BOUNDED CONTRASTIVE LOSS

Notation. The source collection is $\mathcal{D} = \{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$, where $\mathcal{G}_j = \{(G_i^{(j)}, y_i^{(j)})\}_{i=1}^{N_j}$ and $G_i^{(j)} = (\mathbf{A}_i^{(j)}, \mathbf{X}_i^{(j)})$. The multi-scale kernel parameters are $\lambda_1, \ldots, \lambda_Q$, the spectral dimension is $\bar{d}$, and $\bar{N}_j = \sum_i n_i^{(j)}$. The backbone/reference encoder remains $F_{\mathcal{W},\mathcal{V},\gamma}$, with output $\mathbf{g} \in \mathbb{R}^r$. The contrastive temperature remains ϱ. The scalar R is the number of reference distributions, whereas $\mathbf{R}_j^{(\lambda_q)}$ is an orthogonal alignment matrix.

We use $\gamma_{\mathrm{A}} > 0$ for the density-alignment bandwidth and reserve $\gamma > 0$ for the learnable RBF parameter in the reference-distribution module. Also, μ_j below denotes the scalar mean pairwise attribute distance in domain j, whereas $\boldsymbol{\mu}_j^{(\lambda_q)}$ is its vector-valued spectral mean.

Let $\boldsymbol{\omega} = (\mathcal{W}, \mathcal{V}, \log\gamma)$ and abbreviate $F_{\boldsymbol{\omega}} = F_{\mathcal{W},\mathcal{V},\gamma}$. The target domain is indexed by T and has training set $\mathcal{G}^{\mathrm{train}}$ of size N_{train}.

Inductive target protocol. All target kernel scales, spectral bases, and alignment matrices are estimated from $\mathcal{G}^{\mathrm{train}}$ and then frozen. In particular,

$$\mathbf{Z}_{\lambda_q}(G) = \mathbf{K}^{G,\mathrm{train}}_{\lambda_q} \mathbf{U}^{\mathrm{train}}_{\bar{d}} (\boldsymbol{\Sigma}^{\mathrm{train}}_{\bar{d}})^{-1/2}, \qquad \overline{\mathbf{Z}}_{\lambda_q}(G) = \mathbf{Z}_{\lambda_q}(G)(\mathbf{R}_T^{(\lambda_q)})^{\top}. \tag{80}$$

Here $\mathbf{U} = \mathbf{V}$ can be chosen because the exact Gaussian kernel is symmetric positive semidefinite. The same $\mathbf{R}_T^{(\lambda_q)}$ is applied to all future test graphs. There is *no* separately estimated $\mathbf{R}^{\mathrm{test}}$. The kernel used for each cross-kernel evaluation must have the same frozen bandwidth as the training kernel.

Use $\mathcal{T}_j$ to denote the entire frozen domain-specific feature map (kernel, spectral coordinates, and alignment), and write

$$\mathbf{g}_{\boldsymbol{\omega},j}(G) = F_{\boldsymbol{\omega}}(\mathbf{A}_G, \mathcal{T}_j(G)), \qquad \mathbf{e}_{\boldsymbol{\omega},j}(G) = \frac{\mathbf{g}_{\boldsymbol{\omega},j}(G)}{\|\mathbf{g}_{\boldsymbol{\omega},j}(G)\|_2}. \tag{81}$$

Because the contrastive score uses cosine normalization, the denominator must be nonzero. A numerical stabilization or a zero-output convention would require its own explicit version of the analysis; neither is silently inserted here.

Additional assumptions. Statistical independence of original graphs, a precise mini-batch sampler, positive-empty handling, prior independence, and bounded sensitivity of the *normalized* frozen encoder are stated where used. They do not follow from the architecture alone. Graphs may have arbitrary internal node/edge dependence. Augmented views of one graph are *not* counted as independent graphs. No correspondence between individual source and target nodes, no shared class names across datasets, and no equality of domain distributions is assumed in the transfer theorem.

Relation to the model specification. The analysis uses the kernel, SVD, orthogonal transformations, negative-MMD reference coordinates, cosine temperature, and positive-set definitions stated in the method. The source PAC–Bayes analysis is derived directly for the multi-example contrastive denominator rather than replacing it by an order-two pairwise loss. Related theory includes contrastive mean classifiers, contrastive PAC–Bayes, and U-statistic treatment of reused contrastive samples; none is invoked as an unverified shortcut.

I.1.1 CENTERED MULTI-POSITIVE CONTRASTIVE LOSS

Fix a mode $a \in \{\mathrm{SCL}, \mathrm{UCL}\}$ and $B \geq 3$ graph views. For SCL, $C(i) = \{u \neq i : y_u = y_i\}$; for UCL, $P(i)$ consists of designated augmented views of the same original graph. Write $P_a(i) = C(i)$ or $P(i)$, respectively, and $p_i = |P_a(i)|$. Positives must belong to the denominator and exclude the anchor itself. Let

$$S_{iv} = \mathbf{e}_i^\top \mathbf{e}_v \in [-1, 1], \qquad s_{iv} = S_{iv}/\varrho. \tag{82}$$

For $p_i > 0$, the fixed-size mini-batch contrastive objective in (14)–(15) has anchor loss

$$\ell_i^a = \log \sum_{v \neq i} \exp(s_{iv}) - \frac{1}{p_i} \sum_{u \in P_a(i)} s_{iu}. \tag{83}$$

This is the average of the log losses over positives, not the logarithm of an averaged positive probability. The analysis below therefore matches the fixed-size mini-batch objective stated in the main text.

Positive-empty and averaging conventions. For a random supervised mini-batch, an anchor may satisfy $C(i) = \varnothing$. Consistent with the main-text objective, an invalid anchor contributes zero, and a batch is either averaged over all B anchors or over valid anchors only; a batch with no valid anchor contributes zero. The convention must be fixed before evaluating a risk. Both conventions satisfy all results below; the class-coverage theorem explicitly accommodates both. Sampling only valid batches changes the sampling distribution and is not automatically covered by the iid-batch bridge.

Definition I.1 (Centered multi-positive loss)**.** For a valid anchor define

$$\ell_i^{a,\circ} = \ell_i^a - \log p_i, \qquad \ell_{a,B}^\circ = \sum_i w_i \ell_i^{a,\circ}, \tag{84}$$

where $w_i = 1/B$ on valid anchors under all-anchor averaging, or $w_i = 1/|\{i : p_i > 0\}|$ under valid-anchor averaging. Invalid weights are zero. The centered batch loss is zero if no anchor is valid.

The subtracted quantity is determined by the graph labels/augmentation identities and the batch, not by $\mathcal{W}, \mathcal{V}, \gamma$. Consequently,

$$\nabla_{\boldsymbol{\omega}} \ell_{a,B}^\circ = \nabla_{\boldsymbol{\omega}} \ell_{a,B}. \tag{85}$$

This is an exact change of the evaluation baseline, not a changed training objective. Raw and centered risks differ by the explicitly recorded expected positive-count baseline. Source risks are domain-balanced and therefore average over the M domains.

Lemma I.2 (Boundedness and score stability)**.** *Let*

$$L_B = \log(B-1) + \frac{2}{\varrho}. \tag{86}$$

Then $0 \leq \ell_{a,B}^\circ \leq L_B$*. For the same positive sets and averaging convention,*

$$|\ell_{a,B}^\circ(S) - \ell_{a,B}^\circ(S')| \leq \frac{2}{\varrho} \|S - S'\|_{\max}. \tag{87}$$

If corresponding unit graph representations differ by at most ϵ*, then*

$$|\ell_{a,B}^\circ(\mathbf{e}_1, \ldots, \mathbf{e}_B) - \ell_{a,B}^\circ(\mathbf{e}'_1, \ldots, \mathbf{e}'_B)| \leq \frac{4\epsilon}{\varrho}. \tag{88}$$

Proof. For each valid anchor, the arithmetic–geometric mean inequality gives

$$\sum_{v \neq i} e^{s_{iv}} \geq \sum_{u \in P_a(i)} e^{s_{iu}} \geq p_i \exp\left(\frac{1}{p_i} \sum_{u \in P_a(i)} s_{iu} \right).$$

Taking logarithms proves $\ell_i^a \geq \log p_i$. On the other hand, $\log \sum_{v \neq i} e^{s_{iv}} \leq \log(B-1) + 1/\varrho$ and the negative positive-score average is at most $1/\varrho$. Since $p_i \geq 1$, this proves the upper bound for each centered anchor. Weighted averaging, with total weight at most one, proves the batch bound.

If $\|u - v\|_\infty \le d$, then $e^{-d}\sum_k e^{v_k} \le \sum_k e^{u_k} \le e^d \sum_k e^{v_k}$. Therefore log-sum-exp changes by at most d. The positive-score average also changes by at most d, with $d = \|S - S'\|_{\max}/\varrho$. The constant $\log p_i$ is unchanged. This proves (87). Finally,

$$|\mathbf{e}_i^\top \mathbf{e}_v - (\mathbf{e}'_i)^\top \mathbf{e}'_v| \le \|\mathbf{e}_i - \mathbf{e}'_i\|_2 + \|\mathbf{e}_v - \mathbf{e}'_v\|_2 \le 2\epsilon.$$

Substitution proves (88). □

Definition I.3 (Contrastive configuration distance)**.** A configuration is $(S, \mathcal{P})$, where $\mathcal{P}$ records all positive sets and thus all valid-anchor weights under the fixed convention. On a fixed batch size define

$$d_B\big((S,\mathcal{P}),(S',\mathcal{P}')\big) = \|S - S'\|_{\max} + \frac{\varrho L_B}{2}\mathbf{1}\{\mathcal{P} \ne \mathcal{P}'\}. \tag{89}$$

This is a metric on score/mask configurations (or a pseudometric on the underlying graphs). The indicator is deliberately conservative. It prevents an unproved identification of class positives and augmentation positives.

Corollary I.4 (Risk transfer in configuration space)**.** *For any two configuration distributions ρ, ρ' on the same B and convention,*

$$|\mathbb{E}_\rho \ell^\circ_{a,B} - \mathbb{E}_{\rho'} \ell^\circ_{a,B}| \le \frac{2}{\varrho} W_{1,d_B}(\rho, \rho'). \tag{90}$$

The symbol a only indicates the mask-generating protocol; the loss is the same measurable function of a configuration.

Proof. Change scores while holding the first mask fixed and apply (87). Then change masks while holding scores fixed. The latter difference is at most L_B by boundedness, and zero for equal masks. Thus the loss is $2/\varrho$-Lipschitz for (89). For any coupling Γ of ρ, ρ', integrate the pointwise inequality to obtain $(2/\varrho)\mathbb{E}_\Gamma d_B$. Take the infimum over couplings. No existence of an optimal coupling is needed. □

I.2 Step 1: Source contrastive generalization

This step turns the empirical mini-batch contrastive objective into a population-risk certificate for the deterministic pretrained encoder. The three issues are, in order, dependence among overlapping contrastive blocks, data-dependent source preprocessing, and the gap between a PAC–Bayes Gibbs predictor and the deployed checkpoint.

I.2.1 Sampling units and the empirical object

Assumption I.5 (Independent base records and a fixed comparison map)**.** For domain j, the N_j original graph records are iid from $\mathbb{P}_j$, and records in different domains are independent. A record may include its independently generated collection of augmentation seeds/views. Internal views remain one record. A contrastive block uses b distinct base records and produces B views; for ordinary supervised batches $b = B$. For v fixed views per original graph, $B = vb$. Any additional block randomness is averaged in the loss kernel. Its permutation symmetrization is used if necessary. Comparison preprocessing maps $\mathcal{T}_j^0$ are fixed independently of these base records, possibly conditionally on independent auxiliary data. The prior Π is also independent of them.

The b in the effective sample size is *not* the number of positive pairs or the number of augmented views. Different domains need not share a graph distribution. A full-dataset denominator rather than a fixed-size mini-batch corresponds to a high-order kernel; its sample size cannot be silently replaced by a pairwise count.

For each $\boldsymbol{\omega}$ let $h^0_{j,\boldsymbol{\omega}}(Z_1, \dots, Z_b) \in [0, L_B]$ be the expected centered block loss with map $\mathcal{T}_j^0$. The Z_i here denote graph records and should not be confused with the bold spectral matrix $\mathbf{Z}$. Let

$$\widehat{\mathcal{L}}^{a,\circ}_{j,0}(\boldsymbol{\omega}) = \binom{N_j}{b}^{-1} \sum_{I \subset [N_j]: |I| = b} h^0_{j,\boldsymbol{\omega}}(Z_I), \tag{91}$$

$$\mathcal{L}^{a,\circ}_{j,0}(\boldsymbol{\omega}) = \mathbb{E}_{Z_1,\dots,Z_b \sim \mathbb{P}_j} h^0_{j,\boldsymbol{\omega}}(Z_1, \dots, Z_b). \tag{92}$$

Use $M^{-1}\sum_j$ for source averages and $\mathbb{E}_{\boldsymbol{\omega}\sim\mathsf{Q}}$ for Gibbs averages. Conditional on a fixed observed pool, (91) is the expected loss of a uniformly sampled block without replacement. It is not automatically an average of logged SGD losses from changing parameter values. A fixed final-model evaluation or an appropriate Monte Carlo estimate is needed.

Set

$$k = \min_j \left\lfloor \frac{N_j}{b} \right\rfloor \geq 1, \qquad S_{\text{eff}} = Mk. \tag{93}$$

This is the conservative effective size of the following block-matching proof. It is not asserted to be optimal among all U-statistic bounds.

Theorem I.6 (Bounded contrastive-block PAC–Bayes)**.** *Under Assumption I.5, fix $t > 0$ and $\delta \in (0,1)$. With probability at least $1-\delta$ over the source records, simultaneously for every $\mathsf{Q} \ll \Pi$,*

$$\mathcal{L}^{a,\circ}_{\text{src},0}(\mathsf{Q}) \leq \mathfrak{B}_t\left(\widehat{\mathcal{L}}^{a,\circ}_{\text{src},0}(\mathsf{Q}), \mathrm{KL}(\mathsf{Q}\|\Pi) + \log(1/\delta)\right), \tag{94}$$

where

$$\mathfrak{B}_t(r,c) = L_B \min\left\{1, \frac{1-\exp[-tr/L_B - c/S_{\text{eff}}]}{1-e^{-t}}\right\}, \qquad 0 \leq r \leq L_B. \tag{95}$$

For a finite predetermined grid $\mathcal{T}$ of t values, one may minimize over the grid after replacing $\log(1/\delta)$ by $\log(|\mathcal{T}|/\delta)$.

Proof. **Step 1: random disjoint blocks.** For each domain draw a uniform random permutation π_j of $[N_j]$ and use the first kb positions to form k disjoint blocks of size b. Let $\widetilde{L}_{\boldsymbol{\pi}}(\boldsymbol{\omega})$ be the average of the Mk block losses, divided by L_B. Each block position is uniformly distributed over size-b subsets. Therefore, conditionally on the observed source pool,

$$\mathbb{E}_{\boldsymbol{\pi}}\widetilde{L}_{\boldsymbol{\pi}}(\boldsymbol{\omega}) = \widehat{\mathcal{L}}^{a,\circ}_{\text{src},0}(\boldsymbol{\omega})/L_B. \tag{96}$$

This identity remains true when k is smaller than the number of all possible blocks in a larger domain.

Step 2: exponential moment for fixed parameters. For $x \in [0,1]$, convexity gives $e^{-tx} \leq 1 - x(1-e^{-t})$. For a fixed collection of permutations, the normalized block losses X_{jr} are independent, and their means are $p_j = \mathcal{L}^{a,\circ}_{j,0}(\boldsymbol{\omega})/L_B$. Consequently,

$$\begin{aligned}\mathbb{E}\exp\left(-t\sum_{j,r} X_{jr}\right) &\leq \prod_{j=1}^{M}[1-(1-e^{-t})p_j]^k \\ &\leq [1-(1-e^{-t})\bar{p}]^{Mk}, \qquad \bar{p} = \frac{1}{M}\sum_j p_j,\end{aligned}$$

where the second inequality is arithmetic–geometric mean. Define $\Phi_t(p) = -t^{-1}\log[1-p(1-e^{-t})]$. It follows that

$$\mathbb{E}\exp\{tS_{\text{eff}}[\Phi_t(\bar{p}) - \widetilde{L}_{\boldsymbol{\pi}}(\boldsymbol{\omega})]\} \leq 1.$$

By (96) and Jensen's inequality for the exponential, the same inequality holds after replacing $\widetilde{L}$ by the complete normalized U-statistic. This is the step that handles overlap between contrastive blocks without declaring them independent.

Step 3: prior integration and a simultaneous event. Let

$$f_{\mathcal{D}}(\boldsymbol{\omega}) = tS_{\text{eff}}\left[\Phi_t\left(\mathcal{L}^{a,\circ}_{\text{src},0}(\boldsymbol{\omega})/L_B\right) - \widehat{\mathcal{L}}^{a,\circ}_{\text{src},0}(\boldsymbol{\omega})/L_B\right].$$

Independence of the prior, Tonelli's theorem, and the preceding moment bound imply $\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\Pi}e^{f_{\mathcal{D}}} \leq 1$. Markov's inequality therefore gives the event $\mathbb{E}_{\Pi}e^{f_{\mathcal{D}}} \leq 1/\delta$ with probability at least $1-\delta$.

On this event, for all $\mathsf{Q} \ll \Pi$,

$$\mathbb{E}_{\mathsf{Q}} f_{\mathcal{D}} \leq \mathrm{KL}(\mathsf{Q}\|\Pi) + \log \mathbb{E}_{\Pi} e^{f_{\mathcal{D}}} \leq \mathrm{KL}(\mathsf{Q}\|\Pi) + \log(1/\delta). \tag{97}$$

For completeness, let $q = d\mathsf{Q}/d\Pi$. Jensen gives $\mathbb{E}_{\mathsf{Q}}\log(e^f/q) \leq \log\int_{\{q>0\}} e^f\, d\Pi \leq \log\int e^f\, d\Pi$. This proves the first inequality in (97); the second integral need not equal the restricted integral when the posterior has smaller support.

Step 4: posterior averaging and inversion. The function Φ_t is increasing and convex, since

$$\Phi_t'(p) = \frac{1 - e^{-t}}{t[1 - p(1 - e^{-t})]} > 0, \quad \Phi_t''(p) = \frac{(1 - e^{-t})^2}{t[1 - p(1 - e^{-t})]^2} > 0.$$

Apply Jensen to $\mathbb{E}_{\mathsf{Q}}\Phi_t$ in (97) and divide by tS_{eff}. Solving the resulting inequality for $\mathcal{L}^{a,\circ}_{\mathrm{src},0}(\mathsf{Q})/L_B$ gives the expression in (95). If the untruncated expression exceeds one, use boundedness instead. Finally, a union bound over fixed $t \in \mathcal{T}$ proves the grid statement. □

Unequal sizes and other samplers. One can use all disjoint blocks with weights $\alpha_j = \lfloor N_j/b \rfloor / \sum_h \lfloor N_h/b \rfloor$; the same proof then has sample size $\sum_j \lfloor N_j/b \rfloor$ but controls that *different* weighted source risk. Class-balanced batch construction, hard-positive mining, or repeated augmented views are not iid blocks of independent views. Their base-record sampler must be explicitly specified; the U-statistic kernel may then require stratification or further corrections.

I.2.2 The same training graphs may construct the preprocessing

The actual maps $\widehat{\mathcal{T}}_j = \mathcal{T}_j(\mathcal{D})$ may depend on all source graphs, including across domains through alignment. In that case the fixed-map theorem cannot be applied by simply conditioning on those maps. We now give an explicit correction instead of making that invalid step.

For a realized data set and a graph block, couple the actual-map and comparison-map evaluations using the same records and augmentation randomness. Let $D_j(\boldsymbol{\omega}; Z_{1:b})$ be the conditional expected maximum, over the B views, of

$$\|\mathbf{e}_{\boldsymbol{\omega},\widehat{\mathcal{T}}_j}(G) - \mathbf{e}_{\boldsymbol{\omega},\mathcal{T}_j^0}(G)\|_2.$$

Define $\epsilon_{\mathrm{map,emp}}(\mathsf{Q})$ as $4/\varrho$ times its posterior expectation and domain-balanced complete U-average on the observed pool. Define $\epsilon_{\mathrm{map,pop}}(\mathsf{Q})$ analogously using fresh base records with the actual fitted maps frozen. These are explicit representation discrepancies, not assumed convergence rates. More explicitly,

$$\epsilon_{\mathrm{map,emp}}(\mathsf{Q}) = \frac{4}{M\varrho}\mathbb{E}_{\mathsf{Q}} \sum_{j=1}^{M} \binom{N_j}{b}^{-1} \sum_{I:|I|=b} D_j(\boldsymbol{\omega}; Z_I), \tag{98}$$

$$\epsilon_{\mathrm{map,pop}}(\mathsf{Q}) = \frac{4}{M\varrho}\mathbb{E}_{\mathsf{Q}} \sum_{j=1}^{M} \mathbb{E}_{Z_{1:b}\sim\mathbb{P}_j^b} D_j(\boldsymbol{\omega}; Z_{1:b}). \tag{99}$$

The second expectation uses fresh records and keeps the maps fitted on the observed pool fixed. No conditioning argument is used to make the original training records independent after preprocessing.

Corollary I.7 (Source certificate with learned preprocessing)**.** *On the event of Theorem I.6 with $\delta = \delta_{\mathrm{pre}}$, define*

$$\overline{r}_{\mathsf{Q}} = \min\{L_B, \widehat{\mathcal{L}}^{a,\circ}_{\mathrm{src},\widehat{\mathcal{T}}}(\mathsf{Q}) + \epsilon_{\mathrm{map,emp}}(\mathsf{Q})\},$$
$$c_{\mathsf{Q}} = \mathrm{KL}(\mathsf{Q}\|\Pi) + \log(1/\delta_{\mathrm{pre}}).$$

Then

$$\mathcal{L}^{a,\circ}_{\mathrm{src},\widehat{\mathcal{T}}}(\mathsf{Q}) \le \min\{L_B, \mathfrak{B}_t(\overline{r}_{\mathsf{Q}}, c_{\mathsf{Q}}) + \epsilon_{\mathrm{map,pop}}(\mathsf{Q})\}. \tag{100}$$

Proof. Apply (88) on every coupled block, average over the observed pool or fresh draws as appropriate, and then over the posterior. This gives

$$\widehat{\mathcal{L}}^{a,\circ}_{\mathrm{src},0}(\mathsf{Q}) \le \widehat{\mathcal{L}}^{a,\circ}_{\mathrm{src},\widehat{\mathcal{T}}}(\mathsf{Q}) + \epsilon_{\mathrm{map,emp}}(\mathsf{Q}),$$

$$\mathcal{L}^{a,\circ}_{\mathrm{src},\widehat{\mathcal{T}}}(\mathsf{Q}) \le \mathcal{L}^{a,\circ}_{\mathrm{src},0}(\mathsf{Q}) + \epsilon_{\mathrm{map,pop}}(\mathsf{Q}).$$

Use Theorem I.6, monotonicity of $\mathfrak{B}_t$ in its first argument, and boundedness. The comparisons are pathwise, so the maps may have been estimated on the same data without any false independence claim. □

These terms vanish for a genuinely independent fixed reference construction. They can also be bounded by spectral/out-of-sample perturbation plus (141), followed by a bounded encoder input sensitivity. For example, if a normalized encoder obeys $\|\mathbf{e}(A,Z)-\mathbf{e}(A,Z')\| \le L_Z\|Z-Z'\|_F/\sqrt{n}$ and single-node multi-scale errors are uniformly at most d_j, then both map corrections are at most $4L_Z M^{-1}\sum_j d_j/\varrho$. That sensitivity and the approximation errors must actually be established; mean-objective descent is not a bound on $\|\widehat{R}O - R_0\|$.

I.2.3 From Gibbs risk to the deterministic pretrained encoder

Let $\widehat{\boldsymbol{\omega}}$ be the deployed parameter vector. For the same frozen actual maps define

$$\epsilon_{\mathrm{det}}(\widehat{\boldsymbol{\omega}},\mathsf{Q}) = \frac{4}{M\varrho}\sum_j \mathbb{E}_{\boldsymbol{\omega}\sim\mathsf{Q}}\mathbb{E}_{\text{fresh block},j}\max_i \|\mathbf{e}_{\widehat{\boldsymbol{\omega}},j}(G_i)-\mathbf{e}_{\boldsymbol{\omega},j}(G_i)\|_2. \tag{101}$$

By (88), the deterministic source risk is at most the Gibbs source risk plus (101). Thus, if the right side of (100) is denoted $\mathcal{C}^a_{\mathrm{Gibbs}}(\mathsf{Q})$,

$$\mathcal{L}^{a,\circ}_{\mathrm{src}}(\widehat{\boldsymbol{\omega}}) \le \mathcal{C}^a_{\mathrm{pre}}(\delta_{\mathrm{pre}}) := \min\{L_B, \mathcal{C}^a_{\mathrm{Gibbs}}(\mathsf{Q}) + \epsilon_{\mathrm{det}}(\widehat{\boldsymbol{\omega}},\mathsf{Q})\}. \tag{102}$$

No claim is made that a concentrated posterior automatically makes this term small uniformly on unseen graphs. It must be bounded or left explicit.

For diagonal Gaussians with fixed prior center $\boldsymbol{\omega}_0$ and scales s_k, and posterior mean $\widehat{\boldsymbol{\omega}}$ and scales σ_k, the KL is

$$\mathrm{KL}(\mathsf{Q}\|\Pi) = \frac{1}{2}\sum_{k=1}^{p_{\mathrm{net}}}\left[\frac{\sigma_k^2+(\widehat{\omega}_k-\omega_{0,k})^2}{s_k^2} - 1 + \log\frac{s_k^2}{\sigma_k^2}\right]. \tag{103}$$

This follows by taking the posterior expectation of the difference of the two Gaussian log densities: the quadratic expectations are σ_k^2 around the posterior mean and $\sigma_k^2+(\widehat{\omega}_k-\omega_{0,k})^2$ around the prior mean. A Dirac posterior at a point has infinite KL relative to a continuous Gaussian prior. With a predetermined discrete prior assigning that implemented parameter vector positive mass, a Dirac posterior instead has KL $-\log\Pi\{\widehat{\boldsymbol{\omega}}\}$ and $\epsilon_{\mathrm{det}} = 0$; the resulting code-length penalty can still be large.

Monte Carlo certification. After fixing the posterior, independently draw K posterior/block/domain samples from the empirical evaluation distribution. Their centered losses are conditionally iid in $[0, L_B]$. Hoeffding gives an upper correction $L_B\sqrt{\log(1/\delta_{\mathrm{MC}})/(2K)}$ for the empirical Gibbs risk. Its elementary exponential proof uses $\mathbb{E}e^{\lambda(X-\mathbb{E}X)} \le e^{\lambda^2 L_B^2/8}$ and Markov, optimized at $\lambda = 4\epsilon/L_B^2$ for the sample mean after rescaling by K. Fresh Monte Carlo randomness is needed after posterior tuning, or its selection must be accounted for. Map/noise corrections require their own valid upper bounds; evaluating only the uncorrected empirical risk is not sufficient.

A source certificate obtained from independent held-out blocks of the fully frozen deterministic pipeline is also valid and can replace $\mathcal{C}^a_{\mathrm{pre}}$ in what follows. The transfer and classification results depend on a correct source certificate, not on the use of PAC–Bayes specifically.

I.3 Step 2: From source contrastive risk to target contrastive risk

The transfer step first records the spectral facts needed to control aligned means, and then isolates the part of the target–source discrepancy explicitly affected by density-maximization alignment.

I.3.1 Kernel, spectral, and Nyström facts used in transfer

The following are deterministic facts. They do not assume that different datasets contain corresponding nodes or have identical distributions. Perturbation comparisons are only made when two matrices are defined on the *same* anchor system.

Proposition I.8 (Invariance and relational stability)**.** *For a fixed nonconstant node collection, let* $D_{uv} = \|\mathbf{x}_u-\mathbf{x}_v\|_2^2/\mu^2$, $\mu>0$, *and* $K_{\lambda,uv} = e^{-D_{uv}/(2\lambda)}$. *If* $\mathbf{x}'_u = a\mathbf{B}\mathbf{x}_u+\mathbf{b}$ *with* $a>0$ *and* $\mathbf{B}^\top\mathbf{B}=\mathbf{I}$ *on the data-difference span, then* $\mathbf{K}'_\lambda = \mathbf{K}_\lambda$. *The output dimension of* $\mathbf{B}$ *may differ from its input dimension. More generally, on the same indexed node collection,*

$$\|\mathbf{K}'_\lambda - \mathbf{K}_\lambda\|_F \le \frac{1}{2\lambda}\|\mathbf{D}'-\mathbf{D}\|_F. \tag{104}$$

Proof. All pairwise distances and their mean are multiplied by a, so their normalized squares are unchanged. For the second assertion, the derivative of $u \mapsto e^{-u/(2\lambda)}$ on $u \geq 0$ has absolute value at most $1/(2\lambda)$. Apply the mean-value inequality entrywise, square, sum, and take a square root. □

This is not an assertion of arbitrary nonlinear invariance or unconditional contraction: $1/(2\lambda)$ can exceed one. Nor does it justify subtracting kernel matrices with different node counts.

Proposition I.9 (Exact spectral geometry and out-of-sample norm control)**.** *Let $\mathbf{K}_\lambda = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top \succeq 0$ be an exact Gaussian Gram matrix on $\bar{N}$ training anchors. Retain $\bar{d}$ positive eigenvalues and set $\mathbf{Z} = \mathbf{U}_{\bar{d}}\mathbf{\Sigma}_{\bar{d}}^{1/2}$. Then*

$$\mathbf{Z}\mathbf{Z}^\top = \mathbf{K}_{\lambda,\bar{d}}, \tag{105}$$

$$\|\mathbf{K}_\lambda - \mathbf{Z}\mathbf{Z}^\top\|_F^2 = \sum_{r>\bar{d}} \sigma_r^2, \tag{106}$$

$$\|\mathbf{z}_u\|_2 \leq 1. \tag{107}$$

For any *new node x, exact out-of-sample extension using the same kernel,*

$$\mathbf{z}(x)^\top = \mathbf{k}_x^\top \mathbf{U}_{\bar{d}}\mathbf{\Sigma}_{\bar{d}}^{-1/2}, \tag{108}$$

also satisfies $\|\mathbf{z}(x)\|_2 \leq 1$. Consequently, the concatenated Q-scale node feature has norm at most $\sqrt{Q}$, and each single-scale empirical spectral mean has norm at most one. Orthogonal alignment preserves all these bounds and the within-domain Gram matrix.

Proof. Multiplication gives (105); orthogonality of eigenvectors gives (106). Since the discarded eigenvalues are nonnegative, $\mathbf{K}_\lambda - \mathbf{K}_{\lambda,\bar{d}} \succeq 0$. Its diagonal is nonnegative and $K_{\lambda,uu} = 1$, proving (107).

For the stronger out-of-sample statement, let ϕ_λ be the kernel feature map in its reproducing-kernel Hilbert space. For retained r define

$$v_r = \sigma_r^{-1/2} \sum_{u=1}^{\bar{N}} U_{ur}\phi_\lambda(x_u).$$

Then $\langle v_r, v_s\rangle = (\sigma_r\sigma_s)^{-1/2}\mathbf{U}_r^\top \mathbf{K}_\lambda \mathbf{U}_s = \mathbf{1}\{r = s\}$. Also $z_r(x) = \langle \phi_\lambda(x), v_r\rangle$. By the Pythagorean theorem for orthogonal projection,

$$\sum_{r=1}^{\bar{d}} z_r(x)^2 \leq \|\phi_\lambda(x)\|^2 = k_\lambda(x,x) = 1.$$

The multi-scale bound follows by summing squares. A mean is a convex combination of vectors in the unit ball. Lastly, $(\mathbf{Z}\mathbf{R}^\top)(\mathbf{Z}\mathbf{R}^\top)^\top = \mathbf{Z}\mathbf{Z}^\top$ and $\|\mathbf{R}z\| = \|\mathbf{z}\|$. □

The out-of-sample conclusion requires a consistent positive-semidefinite kernel construction. It does not automatically cover an arbitrary numerical approximation with inconsistent training and cross-kernel matrices. The feature-space projection residual has average squared norm $\bar{N}^{-1}\sum_{r>\bar{d}}\sigma_r$ on the training anchors; this follows by summing $1 - \|\mathbf{z}_u\|^2$. This is distinct from the squared Frobenius Gram residual in (106).

I.3.2 Alignment discrepancy and mean-mismatch control

For each scale, define the actual training-reference spectral means and their aligned values by

$$\boldsymbol{\mu}_j^{(\lambda_q)} = \frac{1}{\bar{N}_j}(\mathbf{Z}_{\lambda_q}^{(j)})^\top \mathbf{1}, \quad \mathbf{m}_j^{(q)} = \mathbf{R}_j^{(\lambda_q)}\boldsymbol{\mu}_j^{(\lambda_q)}, \quad \mathbf{m}_T^{(q)} = \mathbf{R}_T^{(\lambda_q)}\boldsymbol{\mu}_T^{(\lambda_q)}. \tag{109}$$

The target mean is computed on $\mathcal{G}^{\text{train}}$, not on test graphs. These vectors are reference statistics; the proof does not pretend that an empirical training-node mean equals the mean of a fresh target graph. Set $\mathbf{m}_j = [\mathbf{m}_j^{(1)}; \ldots; \mathbf{m}_j^{(Q)}]$.

Definition I.10 (Alignment discrepancy)**.** For fixed source references, define

$$\mathfrak{a}_q(\mathbf{R}_T^{(\lambda_q)}) = \frac{1}{M}\sum_{j=1}^{M}\left[1 - e^{-\gamma_{\mathrm{A}}\|\mathbf{m}_T^{(q)} - \mathbf{m}_j^{(q)}\|_2^2}\right], \qquad \mathfrak{a} = \sum_{q=1}^{Q}\mathfrak{a}_q. \tag{110}$$

Minimizing $\mathfrak{a}_q$ is exactly maximizing the average Gaussian density of the transformed target mean relative to the fixed source means. It is the frozen-source counterpart of the density-maximization objective in the main method.

Lemma I.11 (The density objective controls a mean-mismatch bound)**.** *Under exact spectral/Nyström features,*

$$\frac{1}{M}\sum_{j=1}^{M}\|\mathbf{m}_T - \mathbf{m}_j\|_2 \le 2\sqrt{\frac{\mathfrak{a}}{1 - e^{-4\gamma_{\mathrm{A}}}}}. \tag{111}$$

Proof. By Proposition I.9, every single-scale reference mean has norm at most one, before and after rotation. Thus $d_{jq}^2 = \|\mathbf{m}_T^{(q)} - \mathbf{m}_j^{(q)}\|_2^2 \in [0, 4]$. Concavity of $h(t) = 1 - e^{-\gamma_{\mathrm{A}} t}$ on $[0, 4]$ implies $h(t) \ge th(4)/4$. Hence

$$\frac{1}{M}\sum_{j} d_{jq}^2 \le \frac{4\mathfrak{a}_q}{1 - e^{-4\gamma_{\mathrm{A}}}}.$$

Sum this inequality over q, then use $M^{-1}\sum_j \sqrt{\sum_q d_{jq}^2} \le \sqrt{M^{-1}\sum_j \sum_q d_{jq}^2}$. □

If each scale uses a different $\gamma_{\mathrm{A},q}$, replace $\mathfrak{a}/(1 - e^{-4\gamma_{\mathrm{A}}})$ by $\sum_q \mathfrak{a}_q/(1 - e^{-4\gamma_{\mathrm{A},q}})$.

I.3.3 A MONOTONICITY GUARANTEE FOR TARGET-ONLY ALIGNMENT

Proposition I.12 (Frozen-source target update)**.** *Fix a scale and abbreviate* $\boldsymbol{\mu}_T^{(\lambda_q)} = \mu$ *and* $\mathbf{m}_j^{(q)} = m_j$*. For* $R^\top R = I$ *let*

$$J(R) = \frac{1}{M}\sum_{j} e^{-\gamma_{\mathrm{A}}\|R\mu - m_j\|^2}, \qquad S(R) = \frac{1}{M}\sum_{j} e^{-\gamma_{\mathrm{A}}\|R\mu - m_j\|^2} m_j \mu^\top.$$

For any fixed $\eta > 0$*, update*

$$R^+ = \arg\max_{V^\top V = I} \langle S(R) + \eta R, V\rangle_F. \tag{112}$$

It is computed by the same polar/SVD step as Algorithm 1. Then

$$J(R^+) - J(R) \ge \gamma_{\mathrm{A}}\eta\|R^+ - R\|_F^2 \ge 0. \tag{113}$$

Repeated updates have convergent objective values, vanishing successive increments, and stationary accumulation points. This proposition concerns one target with frozen sources*; it is not a replacement for the joint-source convergence theorem.*

Proof. On the orthogonal group, J equals the restriction of the convex function

$$\widetilde{J}(V) = \frac{1}{M}\sum_{j} e^{-\gamma_{\mathrm{A}}(\|\mu\|^2 + \|m_j\|^2)} e^{2\gamma_{\mathrm{A}} m_j^\top V \mu}.$$

Its Euclidean gradient at an orthogonal R is $2\gamma_{\mathrm{A}} S(R)$. Convexity gives $J(R^+) - J(R) \ge 2\gamma_{\mathrm{A}}\langle S(R), R^+ - R\rangle_F$. Optimality in (112) gives $\langle S(R) + \eta R, R^+ - R\rangle_F \ge 0$. Since both matrices are orthogonal, $\langle R, R^+ - R\rangle_F = -\|R^+ - R\|_F^2/2$. These two inequalities prove (113).

Since $0 < J \le 1$, summing (113) over iterations yields $\sum_t \|R^{(t+1)} - R^{(t)}\|_F^2 < \infty$. Compactness gives accumulation points. Along a convergent subsequence, the successive iterate has the same limit. Continuity of S and the closed optimality inequality imply that a limit R^* maximizes $\langle S(R^*) + \eta R^*, V\rangle$ over the orthogonal group. Its tangent first-order condition drops the ηR^* term, since $\langle R^*, R^*\Omega\rangle = 0$ for skew-symmetric Ω. This is exactly the tangent first-order condition of J. □

Algorithm 1 uses unnormalized sums in its update. Dividing the entire linear subproblem by M gives (112) with a rescaled positive η. If a target self-term is retained, it adds a nonnegative multiple of $R\mu\mu^\top$ to the subproblem. Its inner product with $R^+ - R$ equals $-\|(R^+ - R)\mu\|^2/2$, so the same ascent conclusion remains valid. The proof does not require the larger joint-source step restriction.

I.3.4 From alignment to the actual contrastive risk

Assumption I.13 (Common-shift sensitivity of the normalized encoder)**.** For the fixed $\boldsymbol{\omega}$ under analysis there is a finite $L_{\mathrm{shift}}(\boldsymbol{\omega})$ such that, on all graph inputs and comparison shifts used below,

$$\|\mathbf{e}_{\boldsymbol{\omega}}(\mathbf{A}, \mathbf{Z} + \mathbf{1}u^\top) - \mathbf{e}_{\boldsymbol{\omega}}(\mathbf{A}, \mathbf{Z} + \mathbf{1}v^\top)\|_2 \le L_{\mathrm{shift}}(\boldsymbol{\omega})\|u - v\|_2. \tag{114}$$

All involved outputs are nonzero. The constant is common to the comparisons, including before/after alignment if those are compared.

This is a directional input-sensitivity assumption, not a claimed architecture-free numerical certificate. It may be large. To obtain it from a differentiable raw encoder, a positive output-norm lower bound and a bound on the derivative in the common-shift directions suffice. A measured Jacobian on finitely many samples alone is not a uniform proof of this assumption. For example, put $v(u) = F_{\boldsymbol{\omega}}(A, Z + \mathbf{1}u^\top)$ and suppose along every relevant line segment that $\|v(u)\| \ge s_0 > 0$ and $\|D_u v(u)\|_2 \le L_0$. Where differentiable,

$$D_u \frac{v(u)}{\|v(u)\|} = \frac{I - e(u)e(u)^\top}{\|v(u)\|} D_u v(u),$$

whose operator norm is at most L_0/s_0. Integrating along the segment proves (114) with $L_{\mathrm{shift}} \le L_0/s_0$ (also for locally Lipschitz piecewise differentiable maps using their almost-everywhere derivatives). These bounds must hold along the comparisons, not just at the endpoints.

For a frozen target map, let $\overline{\mathbf{Z}}_T(G)$ be the concatenated aligned feature matrix. Define the proof-only comparison input

$$\overline{\mathbf{Z}}_{T\to j}(G) = \overline{\mathbf{Z}}_T(G) + \mathbf{1}(\mathbf{m}_j - \mathbf{m}_T)^\top. \tag{115}$$

The graph, its label or augmentation identity, and its centered feature structure are unchanged. This is not an additional algorithmic operation.

Let $\mathcal{L}_j^{a,\circ}(\boldsymbol{\omega})$ be the population expectation of (84) on fresh batches from domain j, using its frozen map and its declared mode-a sampler. Let $\mathcal{L}_{\mathrm{src}}^{a,\circ} = M^{-1}\sum_j \mathcal{L}_j^{a,\circ}$. Define $\mathcal{L}_T^{a,\circ}$ and $\mathcal{L}_{T\to j}^{a,\circ}$ similarly. All compared configurations have the same number B of views and the same anchor-averaging convention. Let their score/mask distributions be $\rho_j^{a,\boldsymbol{\omega}}$ and $\rho_{T\to j}^{a,\boldsymbol{\omega}}$.

Definition I.14 (Residual task-relevant discrepancy)**.**

$$\Delta_{\mathrm{res}}^a(\boldsymbol{\omega}) = \frac{2}{M\varrho}\sum_{j=1}^{M} W_{1,d_B}\big(\rho_{T\to j}^{a,\boldsymbol{\omega}}, \rho_j^{a,\boldsymbol{\omega}}\big). \tag{116}$$

This retains topology, centered feature geometry, positive relations, and sampling differences not explained by the mean-reference displacement. Its definition needs no cross-domain correspondence between label names or nodes. It is not assumed to vanish, and may depend on the chosen rotations.

Theorem I.15 (Alignment-sensitive contrastive transfer)**.** *For each mode a, under Assumption I.13 and exact features as in Proposition I.9,*

$$\mathcal{L}_T^{a,\circ}(\boldsymbol{\omega}) \le \mathcal{L}_{\mathrm{src}}^{a,\circ}(\boldsymbol{\omega}) + \mathcal{E}_{\mathrm{A}}(\boldsymbol{\omega}) + \Delta_{\mathrm{res}}^a(\boldsymbol{\omega}), \tag{117}$$

where

$$\mathcal{E}_{\mathrm{A}}(\boldsymbol{\omega}) = \frac{8L_{\mathrm{shift}}(\boldsymbol{\omega})}{\varrho}\sqrt{\frac{\sum_{q=1}^{Q}\mathfrak{a}_q}{1 - e^{-4\gamma_{\mathrm{A}}}}}. \tag{118}$$

This is a deterministic inequality for every realized collection of frozen maps and reference means, even if those maps were fitted to their training sets. It is not, on its own, a finite-sample source certificate.

Proof. For every target batch and each source reference j, (114) bounds the maximum corresponding graph-representation change by $L_{\text{shift}}\|\mathbf{m}_T - \mathbf{m}_j\|_2$. The positive masks are unchanged under (115). Apply (88) and average to get

$$\mathcal{L}_T^{a,\circ} \le \mathcal{L}_{T\to j}^{a,\circ} + \frac{4L_{\text{shift}}}{\varrho}\|\mathbf{m}_T - \mathbf{m}_j\|_2.$$

By Corollary I.4,

$$\mathcal{L}_{T\to j}^{a,\circ} \le \mathcal{L}_j^{a,\circ} + \frac{2}{\varrho}W_{1,d_B}(\rho_{T\to j}^{a,\boldsymbol{\omega}}, \rho_j^{a,\boldsymbol{\omega}}).$$

Average both inequalities over j and insert Lemma I.11. □

What improvement means. With encoder and sources frozen, reducing $\mathfrak{a}$ reduces the explicit term (118), while preserving every within-domain input Gram matrix. It does *not* force Δ_{res}^a to decrease. For a common L_{shift}, the full right side improves whenever

$$\Delta_{\text{res,after}}^a - \Delta_{\text{res,before}}^a < \mathcal{E}_{\text{A},before} - \mathcal{E}_{\text{A},after}.$$

An improved upper bound is not a proof that realized test risk decreases at every update. Mean alignment leaves rotations orthogonal to a nonzero mean unidentified; if the mean is zero, it constrains none of those directions. This is consistent with the need to retain task-related shift in adaptation analysis Zhao et al. (2019).

I.4 Step 3: From UCL risk to target semantic contrastive risk

For SCL, Step 2 already controls the semantic target risk. For UCL, the augmentation-defined positive relation must first be compared with the same-class relation needed by the downstream classification bridge.

Fix the deployed encoder and its target-training-set-fitted map. The target *semantic* risk is the SCL risk on B independent fresh labeled target graphs, with $C(i)$ defined by equality of their target labels. The target *augmentation* risk is the UCL risk on the declared augmented-view sampler. The latter generally contains dependent views from fewer than B original graphs. Both are evaluation quantities; no test label is used in fitting the pipeline.

Let ρ_T^{SCL} and ρ_T^{UCL} be their score/mask configuration laws, with the same number B of views and the same averaging convention. If the two training modes use different batch sizes, make this construction separately at the batch size of each mode; the comparison SCL risk for a UCL encoder is evaluated at its B.

Proposition I.16 (An explicit augmentation-to-semantic correction)**.** *Define*

$$\epsilon_{\text{aug}\to\text{sem},T} = \frac{2}{\varrho}W_{1,d_B}(\rho_T^{\text{SCL}}, \rho_T^{\text{UCL}}). \tag{119}$$

Then

$$\mathcal{L}_T^{\text{SCL},\circ} \le \mathcal{L}_T^{\text{UCL},\circ} + \epsilon_{\text{aug}\to\text{sem},T}. \tag{120}$$

Proof. Both quantities are expectations of the same centered log-softmax function of a score/mask configuration. Apply Corollary I.4. The distance accounts for both view dependence and the difference in positive relations. □

For any explicit coupling of the two batch-generating procedures, this term is at most

$$\frac{2}{\varrho}\mathbb{E}\|S^{\text{sem}} - S^{\text{aug}}\|_{\max} + L_B\mathbb{P}(\mathcal{P}^{\text{sem}} \ne \mathcal{P}^{\text{aug}}).$$

This is a conservative sufficient bound, not an assertion that the correction is small. Label preservation by augmentations alone is not enough to make it zero, since augmented views of the same graph are not independent draws from a class-conditional graph distribution.

A sharper identity on the very same batch. When both positive sets are nonempty for an anchor and the candidates are identical, let $\overline{e}_i^{\mathrm{sem}}$ and $\overline{e}_i^{\mathrm{aug}}$ be their positive-embedding means. Then

$$\ell_i^{\mathrm{sem},\circ} - \ell_i^{\mathrm{aug},\circ} = \frac{\mathbf{e}_i^\top(\overline{e}_i^{\mathrm{aug}} - \overline{e}_i^{\mathrm{sem}})}{\varrho} + \log\frac{|P(i)|}{|C(i)|}. \tag{121}$$

This follows by cancelling the identical log denominator. The positive-count term must not be omitted after centering. The identity can sharpen a same-batch analysis, but does not replace the sampling-law correction in (119).

Combining Theorem I.15 with the source certificate gives the following target-semantic upper bound for either mode:

$$U_T^a = \min\left\{L_B, \mathcal{C}_{\mathrm{pre}}^a(\delta_{\mathrm{pre}}) + \mathcal{E}_{\mathrm{A}} + \Delta_{\mathrm{res}}^a + \mathbf{1}\{a = \mathrm{UCL}\}\epsilon_{\mathrm{aug}\to\mathrm{sem},T}\right\}, \qquad \mathcal{L}_T^{\mathrm{SCL},\circ} \le U_T^a. \tag{122}$$

The two encoders can have different values of every term. The extra correction does not assert that an unsupervised encoder always performs worse.

I.5 Step 4: From semantic contrastive risk to a feasible classifier

This step proves that a small target semantic contrastive risk implies the existence of a low-risk linear classifier on the learned graph embeddings, without requiring every class to appear in every random mini-batch.

This section proves a bridge for the multi-positive loss in (14) under ordinary iid target graph batches. It does not impose a batch that contains all classes. It does require that the downstream comparison concerns the same fixed representation of fresh graphs used in the semantic loss.

Let $C_T \ge 2$, with target class probabilities $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{C_T})$, each $\pi_c > 0$. Condition on all already fitted maps and the encoder. Define

$$\mathbf{c}_c = \mathbb{E}[\mathbf{e}(G) \mid y = c], \qquad \|\mathbf{c}_c\|_2 \le 1, \qquad (W^{\mathrm{mean}})_{c,:} = \mathbf{c}_c^\top/\varrho. \tag{123}$$

These $\mathbf{c}_c$ are class means of final *graph* embeddings, not the $\boldsymbol{\mu}_j^{(\lambda_q)}$ used for dataset-level node-feature alignment. Let $\mathcal{R}_T^{\mathrm{mean}}$ be the population softmax cross-entropy risk of W^{mean} on normalized $\mathbf{e}(G)$.

Lemma I.17 (A fractional-subset log-sum inequality)**.** *For $a_1, \dots, a_d \ge 0$, fix their order and set*

$$v_k = \log\left(1 + \sum_{h=1}^{k} a_h\right) - \log\left(1 + \sum_{h=1}^{k-1} a_h\right) \ge 0.$$

For $w_k \in [0,1]$,

$$\log\left(1 + \sum_k w_k a_k\right) \ge \sum_k w_k v_k. \tag{124}$$

Proof. For a subset S, add its elements in the fixed order. The increment from adding a_k is $\log(1 + a_k/(1 + \sum_{h<k, h\in S} a_h))$, which is at least v_k because the denominator is no larger than that of the complete prefix. Thus $\log(1 + \sum_{k\in S} a_k) \ge \sum_{k\in S} v_k$.

For $u \in [0,1]$, put $S_u = \{k : w_k \ge u\}$. Since $w_k = \int_0^1 \mathbf{1}\{k \in S_u\}\, du$, concavity of the logarithm gives

$$\log(1 + \sum_k w_k a_k) \ge \int_0^1 \log(1 + \sum_{k\in S_u} a_k)\, du \ge \int_0^1 \sum_{k\in S_u} v_k\, du = \sum_k w_k v_k.$$

□

Definition I.18 (Random-batch class coverage)**.** Let $(K_1, \dots, K_{C_T}) \sim \mathrm{Multinomial}(B-1; \boldsymbol{\pi})$ be the class counts of the reference graphs for an anchor. Define

$$\kappa_{yc} = \mathbb{E}\left[\mathbf{1}\{K_y > 0\}\min\left\{1, \frac{K_c}{K_y}\right\}\right], \qquad \kappa_T(B, \boldsymbol{\pi}) = \min_{y\ne c}\kappa_{yc}, \tag{125}$$

where the integrand is defined to be zero when $K_y = 0$.

For $B \geq 3$ and positive class probabilities, $\kappa_T > 0$: the event with one reference from y and all other references from c has positive probability and integrand one. In particular, B need not exceed C_T. The coefficient may nevertheless be small for rare classes or small batches.

Theorem I.19 (Centered SupCon controls a linear mean classifier)**.** *For B iid fresh target labeled graphs, either of the declared anchor averaging conventions satisfies*

$$\mathcal{R}_T^{\mathrm{mean}} \leq \frac{\mathcal{L}_T^{\mathrm{SCL},\circ}}{\kappa_T(B, \boldsymbol{\pi})}. \tag{126}$$

Consequently, any linear classifier class on normalized embeddings containing W^{mean} has optimal cross-entropy risk no larger than the right side.

Proof. **Step 1: conditional expectation of the actual loss.** Fix an anchor with embedding $\mathbf{e}$ and label y. Given the $B-1$ reference labels, let K_c be their counts. If $K_y > 0$, the centered anchor loss is

$$\log\left(\frac{1}{K_y}\sum_{v\neq i} e^{\mathbf{e}^\top \mathbf{e}_v/\varrho}\right) - \frac{1}{K_y}\sum_{u:y_u=y}\frac{\mathbf{e}^\top \mathbf{e}_u}{\varrho}.$$

The log-sum-exp is convex in its score vector. Conditional on the reference labels, each graph from class c has expected normalized embedding $\mathbf{c}_c$ and is independent of the anchor. Jensen, followed by linearity of the positive-score term, therefore gives

$$\mathbb{E}[\ell_i^{\mathrm{SCL},\circ} \mid \mathbf{e}, y, (y_v)_{v\neq i}] \geq \log\left[1 + \sum_{c\neq y}\frac{K_c}{K_y}\exp\left(\frac{\mathbf{e}^\top(\mathbf{c}_c - \mathbf{c}_y)}{\varrho}\right)\right]. \tag{127}$$

This is exactly why both the positive-count subtraction and conditional class sampling must be tracked. For $K_y = 0$, the anchor contribution is zero under the declared convention.

Step 2: allow classes to be missing. Fix an arbitrary order of the classes other than y and let $a_c = \exp(\mathbf{e}^\top(\mathbf{c}_c - \mathbf{c}_y)/\varrho)$. Let v_c be the increments in Lemma I.17. Then

$$\sum_{c\neq y} v_c = \log(1 + \sum_{c\neq y} a_c) = \mathrm{CE}(W^{\mathrm{mean}}\mathbf{e}, y).$$

For $K_y > 0$, replace K_c/K_y by its smaller clipped value $\min(1, K_c/K_y)$ in (127), and apply Lemma I.17. Including the invalid-anchor case, this yields

$$\mathbb{E}[\ell_i^{\mathrm{SCL},\circ} \mid \mathbf{e}, y] \geq \sum_{c\neq y}\kappa_{yc}v_c \geq \kappa_T\,\mathrm{CE}(W^{\mathrm{mean}}\mathbf{e}, y).$$

The multinomial law of the reference counts is independent of the fixed anchor. No relation between source and target class names is used.

Step 3: average anchors and graphs. Under all-anchor averaging, exchangeability makes the population batch loss equal to the population contribution of one anchor. Integrating the preceding inequality over $(\mathbf{e}, y)$ proves $\mathcal{L}_T^{\mathrm{SCL},\circ} \geq \kappa_T \mathcal{R}_T^{\mathrm{mean}}$. Under valid-anchor averaging, the centered batch loss is pointwise at least the all-anchor average, because every centered contribution is nonnegative and the divisor is no larger than B. The same lower bound thus holds. Finally, an infimum over a class containing W^{mean} cannot exceed its risk. □

An explicit, computable coefficient. For $n = B - 1$,

$$\kappa_{yc} = \sum_{u=1}^{n}\sum_{v=0}^{n-u}\frac{n!}{u!v!(n-u-v)!}\pi_y^u\pi_c^v(1-\pi_y-\pi_c)^{n-u-v}\min\{1, v/u\}. \tag{128}$$

Use the usual $0^0 = 1$ convention in this finite multinomial probability. For illustration, exact evaluation for uniform classes and $B = 64$ gives approximately $0.9124, 0.8605, 0.7856, 0.6229, 0.1938$ for $C_T = 2, 6, 15, 30, 100$, respectively. These are probability calculations, not measured GraphVec performance; the accompanying script reproduces them.

Balanced-reference special case. If an independently chosen anchor is accompanied by exactly q independently sampled references from *each* class, conditional Jensen immediately has all ratios $K_c/K_y = 1$. In that *different sampling protocol*, $\mathcal{R}_T^{\mathrm{mean}} \le \mathcal{L}_T^{\mathrm{SCL}} - \log q$. This recovers the simpler mean-classifier argument, but is not substituted for the random-batch theorem above.

I.5.1 THE DEPLOYED CLASSIFIER MAY USE UNNORMALIZED GRAPH VECTORS

The contrastive scores use normalized graph vectors, whereas the downstream softmax head is allowed to use the raw graph vectors. We therefore analyze the raw-feature head explicitly.

Proposition I.20 (A raw-feature comparison classifier)**.** *For any $a_0 > 0$, define a zero-bias raw-feature linear classifier by $W^0_{c,:} = \mathbf{c}_c^\top/(\varrho a_0)$. Its Frobenius norm is at most $\sqrt{C_T}/(\varrho a_0)$. Its raw-feature population cross-entropy obeys*

$$\mathcal{R}_T(W^0;\mathbf{g}) \le \mathcal{R}_T^{\mathrm{mean}} + \epsilon_{\mathrm{rad}}, \qquad \epsilon_{\mathrm{rad}} = \frac{2}{\varrho}\mathbb{E}_G\left|\frac{\|\mathbf{g}(G)\|_2}{a_0} - 1\right|. \tag{129}$$

Proof. For every class, the raw classifier score differs from the normalized mean classifier score by at most $|\|\mathbf{g}\|/a_0 - 1|/\varrho$, since $|\mathbf{c}_c^\top \mathbf{e}| \le 1$. Cross-entropy $\log\sum_c e^{u_c} - u_y$ is 2-Lipschitz in the maximum norm, by the same log-sum-exp argument as before. Average this score bound. The matrix-norm bound follows from $\|\mathbf{c}_c\| \le 1$. □

Thus a head class containing W^0 satisfies $\mathcal{R}_T^*(\mathbf{g}) \le \mathcal{L}_T^{\mathrm{SCL},\circ}/\kappa_T + \epsilon_{\mathrm{rad}}$. If the implemented head actually uses normalized embeddings, apply the normalized version directly and set $\epsilon_{\mathrm{rad}} = 0$; a suitable head radius is then at least $\sqrt{C_T}/\varrho$.

I.6 STEP 5: GENERALIZATION OF THE CLASSIFIER LEARNED ON THE TARGET SUPPORT SET

The previous step provides a feasible comparison classifier. This step controls the actual softmax head trained from the finite target support set, while explicitly accounting for the fact that the target preprocessing map is itself fitted from that same support set.

Condition only on the source pretraining data, not on the target training set. The deployed $\widehat{\boldsymbol{\omega}}$ is now fixed and independent of fresh target-training graphs. Let $\mathbf{g}_S(G)$ be the graph vector obtained with the map fitted to $S = \mathcal{G}^{\mathrm{train}}$. Introduce a comparison map $\mathcal{T}_T^0$ fixed independently of S, possibly depending on the target population distribution and the already fixed source information. Write $\mathbf{g}_0(G) = F_{\widehat{\boldsymbol{\omega}}}(A_G, \mathcal{T}_T^0(G))$. This map is only for analysis and is not substituted into the deployed method.

Include bias by setting $\mathbf{x}_S(G) = [\mathbf{g}_S(G); 1]$, $\mathbf{x}_0(G) = [\mathbf{g}_0(G); 1]$. Let

$$\mathcal{H}_T = \{\mathbf{W}_{\mathrm{c}} \in \mathbb{R}^{C_T \times (r+1)} : \|\mathbf{W}_{\mathrm{c}}\|_F \le B_{\mathrm{c}}\}, \qquad \|\mathbf{x}_0(G)\|_2 \le A_0. \tag{130}$$

B_{c}, A_0 are fixed analysis bounds. The bias counts in the norm. If a radius is selected after examining target data, use a predetermined grid and a union penalty, or explicitly charge that selection.

Iid and class-stratified sampling. For iid target sampling take weights $w_i = 1/N_{\mathrm{train}}$. To match a k-shot design, alternatively take independent samples within class c, of count $N_{\mathrm{train},c} \ge 1$, and use $w_i = \pi_c/N_{\mathrm{train},c}$. Define

$$N_{\mathrm{eff}}^{\mathrm{c}} = (\sum_i w_i^2)^{-1} = \left(\sum_c \frac{\pi_c^2}{N_{\mathrm{train},c}}\right)^{-1} \tag{131}$$

for the stratified case. Equal class counts k and uniform target class weights give $N_{\mathrm{eff}}^{\mathrm{c}} = C_T k = N_{\mathrm{train}}$. The target population risk and empirical weighting must agree. If training is unweighted balanced-class ERM but evaluation uses a different natural class prior, the mismatch is not silently discarded.

Let $\mathcal{R}_0(W) = \mathbb{E}\,\mathrm{CE}(W\mathbf{x}_0(G), y)$ and $\widehat{\mathcal{R}}_0(W) = \sum_i w_i\,\mathrm{CE}(W\mathbf{x}_0(G_i), y_i)$; define $\mathcal{R}_S, \widehat{\mathcal{R}}_S$ using $\mathbf{x}_S$. For the realized support-fitted map $\mathbf{g}_S$, $\mathcal{R}_S(W)$ is exactly the target population cross-entropy risk denoted $\mathcal{R}_T^{\mathrm{CE}}(W)$ in the final theorem. The new test graph in $\mathcal{R}_S$ is independent of S.

Lemma I.21 (A fixed-feature softmax uniform bound)**.** *Let $H_{\mathrm{c}} = \log C_T + 2B_{\mathrm{c}}A_0$. With probability at least $1-\delta_{\mathrm{c}}$ over the target training graphs, simultaneously for all $W \in \mathcal{H}_T$,*

$$|\mathcal{R}_0(W) - \widehat{\mathcal{R}}_0(W)| \le \mathfrak{G}_{\mathrm{c}}, \tag{132}$$

where

$$\mathfrak{G}_{\mathrm{c}} = 4B_{\mathrm{c}}A_0\sqrt{\frac{C_T}{N^{\mathrm{c}}_{\mathrm{eff}}}} + H_{\mathrm{c}}\sqrt{\frac{\log(2/\delta_{\mathrm{c}})}{2N^{\mathrm{c}}_{\mathrm{eff}}}}. \tag{133}$$

The same expression applies to iid samples with $N^{\mathrm{c}}_{\mathrm{eff}} = N_{\mathrm{train}}$.

Proof. **Step 1: loss range and Lipschitz constants.** Every logit has absolute value at most $B_{\mathrm{c}}A_0$, so the cross-entropy lies in $[0, H_{\mathrm{c}}]$. Its gradient in logits is $p - e_y$ with $\|p-e_y\|_2^2 = \sum_{c\neq y} p_c^2 + (1-p_y)^2 \le 2$. It is therefore $\sqrt{2}$-Lipschitz in Euclidean norm.

Step 2: complexity of the linear head only. For fixed labeled features let $\mathfrak{R}_w = \mathbb{E}_\epsilon \sup_{W\in\mathcal{H}_T} \sum_i w_i\epsilon_i \,\mathrm{CE}(W\mathbf{x}_{0,i}, y_i)$. The vector-contraction inequality, Corollary 4 of Maurer (2016), states that composing vector-valued evaluations with L-Lipschitz scalar losses increases this Rademacher sum by at most the factor $\sqrt{2}L$ when the right side uses independent signs for each coordinate. Apply it with $L = \sqrt{2}$, absorbing each positive w_i into the evaluation (equivalently, use $w_i\,\mathrm{CE}(v/w_i, y_i)$). Constants at zero have zero signed expectation. Thus

$$\begin{aligned}
\mathfrak{R}_w &\le 2\mathbb{E}_{\epsilon_{ic}} \sup_{\|W\|_F\le B_{\mathrm{c}}} \sum_{i,c} w_i\epsilon_{ic}(W\mathbf{x}_{0,i})_c \\
&= 2B_{\mathrm{c}}\mathbb{E}\left\| \left[\sum_i w_i\epsilon_{ic}\mathbf{x}_{0,i}^\top\right]_{c=1}^{C_T} \right\|_F \\
&\le 2B_{\mathrm{c}}\sqrt{C_T\sum_i w_i^2\|\mathbf{x}_{0,i}\|_2^2} \le 2B_{\mathrm{c}}A_0\sqrt{C_T/N^{\mathrm{c}}_{\mathrm{eff}}}.
\end{aligned}$$

The penultimate step is Cauchy–Schwarz/Jensen and cancellation of cross terms of independent signs. This is a complexity bound for the linear head, not a product of the backbone's layer norms.

Step 3: symmetrization and concentration. Introduce independent ghost graphs with the same per-observation law. Jensen, independent sign symmetrization of observation–ghost differences, and splitting the supremum give

$$\mathbb{E}\sup_W(\mathcal{R}_0(W) - \widehat{\mathcal{R}}_0(W)) \le 2\mathbb{E}\mathfrak{R}_w \le 4B_{\mathrm{c}}A_0\sqrt{C_T/N^{\mathrm{c}}_{\mathrm{eff}}}.$$

The reverse one-sided supremum has the same bound. This argument needs independence, not identical distributions, so it covers stratified graphs. Replacing the ith graph changes either supremum by at most $H_{\mathrm{c}}w_i$. The bounded-difference exponential inequality therefore bounds its upper tail by $\exp(-2u^2/\sum_i H_{\mathrm{c}}^2 w_i^2)$. Apply it to both one-sided suprema with failure probability $\delta_{\mathrm{c}}/2$ each, and use the union bound. This is (133). □

For a self-contained alternative not using vector contraction, an elementary covering proof is provided in Appendix I.9; it generally has worse explicit dimension dependence. Neither bound is asserted to be numerically tight for the given checkpoints without evaluating its parameters.

Define the actual-to-comparison representation errors

$$d_{\mathrm{emp}} = \sum_i w_i\|\mathbf{g}_S(G_i) - \mathbf{g}_0(G_i)\|_2, \qquad d_{\mathrm{pop}} = \mathbb{E}_G\|\mathbf{g}_S(G) - \mathbf{g}_0(G)\|_2. \tag{134}$$

Both use the same actual fitted map; d_{pop} is conditional on that realized map. They can be bounded by input-map perturbation and raw encoder sensitivity, but are not assumed to disappear just because the pretrained backbone is frozen.

Theorem I.22 (The classifier's excess risk with learned target preprocessing). *Suppose* $\widehat{\mathbf{W}}_{\mathrm{c}} \in \mathcal{H}_T$ *satisfies*

$$\widehat{\mathcal{R}}_S(\widehat{\mathbf{W}}_{\mathrm{c}}) \le \inf_{W \in \mathcal{H}_T} \widehat{\mathcal{R}}_S(W) + \epsilon_{\mathrm{opt}}. \tag{135}$$

On the event of Lemma I.21,

$$\mathcal{R}_S(\widehat{\mathbf{W}}_{\mathrm{c}}) \le \inf_{W \in \mathcal{H}_T} \mathcal{R}_S(W) + 2\mathfrak{G}_{\mathrm{c}} + 2\sqrt{2}B_{\mathrm{c}}(d_{\mathrm{emp}} + d_{\mathrm{pop}}) + \epsilon_{\mathrm{opt}}. \tag{136}$$

Proof. For all $W \in \mathcal{H}_T$, the Euclidean loss Lipschitz bound gives

$$|\mathcal{R}_S(W) - \mathcal{R}_0(W)| \le \sqrt{2}B_{\mathrm{c}}d_{\mathrm{pop}}, \quad |\widehat{\mathcal{R}}_S(W) - \widehat{\mathcal{R}}_0(W)| \le \sqrt{2}B_{\mathrm{c}}d_{\mathrm{emp}}.$$

Consequently, (132) implies the uniform bound $|\mathcal{R}_S(W) - \widehat{\mathcal{R}}_S(W)| \le D_{\mathrm{c}} := \mathfrak{G}_{\mathrm{c}} + \sqrt{2}B_{\mathrm{c}}(d_{\mathrm{emp}} + d_{\mathrm{pop}})$. Although the actual features depend on S, this conclusion follows from pathwise comparisons on the fixed-feature event, not from conditioning on $\mathcal{T}_T(S)$ and asserting independence. Now

$$\begin{aligned}\mathcal{R}_S(\widehat{W}_{\mathrm{c}}) &\le \widehat{\mathcal{R}}_S(\widehat{W}_{\mathrm{c}}) + D_{\mathrm{c}} \\ &\le \inf_W \widehat{\mathcal{R}}_S(W) + D_{\mathrm{c}} + \epsilon_{\mathrm{opt}} \\ &\le \inf_W \mathcal{R}_S(W) + 2D_{\mathrm{c}} + \epsilon_{\mathrm{opt}}.\end{aligned}$$

Uniformity over W permits a data-dependent optimal comparator in the last line. Expanding D_{c} proves the result. □

The value ϵ_{opt} can always be defined as the empirical excess of the actual trained head within the selected norm ball. It is not set to zero merely because Adam was run for 500 epochs. If regularization or class weighting differs from (135), account for that difference or prove the corresponding regularized oracle inequality.

I.7 Auxiliary stability bounds for preprocessing and representation maps

The next results are not separate links in the main proof chain. They provide concrete ways to control preprocessing and representation-sensitivity terms that remain explicit in the end-to-end theorem.

I.7.1 A fully specified finite-anchor spectral perturbation bound

Lemma I.23 (Square-root perturbation). *For positive definite matrices* A, B *of the same dimension,*

$$\|B^{1/2} - A^{1/2}\|_F \le \frac{\|B - A\|_F}{\sqrt{\lambda_{\min}(A)} + \sqrt{\lambda_{\min}(B)}}. \tag{137}$$

Proof. Write $X = B^{1/2} - A^{1/2}$. Then $B - A = B^{1/2}X + XA^{1/2}$. The Frobenius inner product with X is at least $(\sqrt{\lambda_{\min}(A)} + \sqrt{\lambda_{\min}(B)})\|X\|_F^2$. Cauchy–Schwarz bounds the same inner product by $\|X\|_F\|B - A\|_F$. Cancel $\|X\|_F$ when nonzero; otherwise the claim is immediate. □

Proposition I.24 (Spectral coordinates up to an orthogonal factor). *Let* $K, \widehat{K} \succeq 0$ *be on the same* $\bar{N}$ *anchors, with* $1 \le \bar{d} < \bar{N}$. *Put*

$$\epsilon = \|\widehat{K} - K\|_2, \quad a = \sigma_{\bar{d}}(K) > 0, \quad L = \|K\|_2, \quad \delta_{\mathrm{sp}} = \sigma_{\bar{d}}(K) - \sigma_{\bar{d}+1}(K) > 0.$$

Assume $\epsilon < \delta_{\mathrm{sp}}/2$. *Define*

$$t_{\mathrm{sp}} = \frac{\sqrt{2\bar{d}}\,\epsilon}{\delta_{\mathrm{sp}} - \epsilon}, \qquad \chi_{\mathrm{sp}} = \sqrt{\bar{d}}\,\epsilon + 2Lt_{\mathrm{sp}}. \tag{138}$$

There exists an orthogonal O *such that*

$$\|\widehat{Z}O - Z\|_F \le \sqrt{L + \epsilon}\,t_{\mathrm{sp}} + \frac{\chi_{\mathrm{sp}}}{\sqrt{a} + \sqrt{a - \epsilon}}. \tag{139}$$

Here $Z = U_{\bar{d}}\Sigma_{\bar{d}}^{1/2}$ *and similarly for* $\widehat{Z}$.

Proof. The variational characterization of eigenvalues and $-\epsilon I \preceq \widehat{K} - K \preceq \epsilon I$ imply $|\widehat{\sigma}_r - \sigma_r| \le \epsilon$ for every r. Let $U_\perp$ collect the discarded eigenvectors of K. For each retained $\widehat{u}_r$, project $\widehat{K}\widehat{u}_r = \widehat{\sigma}_r \widehat{u}_r$ onto $U_\perp$:

$$(\widehat{\sigma}_r I - \Sigma_\perp) U_\perp^\top \widehat{u}_r = U_\perp^\top (\widehat{K} - K)\widehat{u}_r.$$

The eigenvalues of the matrix on the left are at least $\delta_{\mathrm{sp}} - \epsilon$. Hence $\|U_\perp^\top \widehat{U}\|_F \le \sqrt{\bar{d}}\epsilon/(\delta_{\mathrm{sp}} - \epsilon)$.

Let the singular values of $U^\top \widehat{U}$ be $c_r \in [0, 1]$. The orthogonal Procrustes solution O gives

$$\|\widehat{U}O - U\|_F^2 = 2\sum_r (1 - c_r) \le 2\sum_r (1 - c_r^2) = 2\|U_\perp^\top \widehat{U}\|_F^2.$$

Set $\overline{U} = \widehat{U}O$ and $B_0 = O^\top \widehat{\Sigma} O$, $A_0 = \Sigma$. Then $B_0 = \overline{U}^\top \widehat{K}\overline{U}$ and $A_0 = U^\top K U$. Expanding their difference yields

$$\|B_0 - A_0\|_F \le \sqrt{\bar{d}}\epsilon + 2L\|\overline{U} - U\|_F \le \chi_{\mathrm{sp}}.$$

Their smallest eigenvalues are at least $a - \epsilon$ and a, respectively. Because $\widehat{Z}O = \overline{U}B_0^{1/2}$, write

$$\widehat{Z}O - Z = (\overline{U} - U)B_0^{1/2} + U(B_0^{1/2} - A_0^{1/2}).$$

Use $\|B_0^{1/2}\|_2 \le \sqrt{L + \epsilon}$ and Lemma I.23 to obtain (139). □

The proof is given explicitly to avoid confusing unweighted eigenspace perturbation with the eigenvalue-weighted spectral coordinates used by GraphVec. Related general Procrustes analyses appear in Arias-Castro et al. (2020).

Corollary I.25 (Perturbed out-of-sample coordinates)**.** *In the setting of Proposition I.24, let $k_x, \widehat{k}_x$ be cross-kernel vectors to the same anchors. Then, for the same O,*

$$\|\widehat{z}(x)^\top O - z(x)^\top\|_2 \le \frac{\|\widehat{k}_x - k_x\|_2}{\sqrt{a - \epsilon}} + \|k_x\|_2 \left[\frac{t_{\mathrm{sp}}}{\sqrt{a - \epsilon}} + \frac{\chi_{\mathrm{sp}}}{\sqrt{a}\sqrt{a - \epsilon}(\sqrt{a} + \sqrt{a - \epsilon})}\right]. \quad (140)$$

Proof. With $\overline{U}, B_0, A_0$ from the preceding proof, $\widehat{z}(x)^\top O = \widehat{k}_x^\top \overline{U} B_0^{-1/2}$. Add and subtract $k_x^\top \overline{U} B_0^{-1/2}$ and $k_x^\top U B_0^{-1/2}$. Bound the three differences by the triangle and submultiplicative inequalities. For the last one use

$$B_0^{-1/2} - A_0^{-1/2} = B_0^{-1/2}(A_0^{1/2} - B_0^{1/2})A_0^{-1/2}$$

and Lemma I.23. This gives exactly (140). □

Remark I.26 (The unknown spectral rotation is not the learned alignment). The existence of O in these results does not imply that mean alignment recovers it. For any comparison alignment R_0, the actual aligned coordinates satisfy the separate, elementary inequality

$$\|\widehat{z}^\top \widehat{R}^\top - z^\top R_0^\top\|_2 \le \|\widehat{z}^\top O - z^\top\|_2 + \|z\|_2 \|\widehat{R}O - R_0\|_2. \quad (141)$$

Indeed, insert $z^\top O^\top \widehat{R}^\top$ and use orthogonality. The second term must remain unless an additional identification/stability condition is proved. A small mean-alignment objective alone is not such a condition. Population spectral rates across independently sampled anchor sets would additionally require an operator-level sampling analysis; the finite-anchor result is not mislabeled as that analysis.

I.7.2 A STABILITY FACT FOR THE REFERENCE BRANCH

Proposition I.27 (Unsquared-MMD reference readout)**.** *For fixed references V_b, let $s_b(H) = -\mathrm{MMD}(H, V_b)$ as defined in the main method, using $k(x, x') = e^{-\gamma\|x - x'\|^2}$. For two n-node representations with a fixed node correspondence,*

$$\|\mathbf{s}(H) - \mathbf{s}(H')\|_2 \le \sqrt{\frac{2\gamma R}{n}}\|H - H'\|_F. \quad (142)$$

More generally, $|s_b(P) - s_b(P')| \le \mathrm{MMD}(P, P')$ for node distributions.

Proof. Write $m(P)$ for the kernel mean. The reverse triangle inequality gives $|\|m(P) - m(V_b)\| - \|m(P') - m(V_b)\|| \le \|m(P) - m(P')\|$. For the Gaussian feature map,

$$\|\phi(h) - \phi(h')\|^2 = 2(1 - e^{-\gamma\|h-h'\|^2}) \le 2\gamma\|h - h'\|^2.$$

Thus the empirical kernel-mean difference is at most $\sqrt{2\gamma}\, n^{-1} \sum_i \|h_i - h'_i\| \le \sqrt{2\gamma/n}\|H - H'\|_F$. Square and sum over the R coordinates. □

This proves a property of the reference branch, not of the entire GIN–Transformer–pooling composite, and it specifically applies to the unsquared MMD used by GraphVec.

I.8 Architecture-dependent specialization and parameter effects

The preceding theorem is solution-dependent: it contains empirical risks, posterior complexity, input sensitivity, and representation discrepancies. This section makes their dependence on GraphVec parameters explicit. It does not replace the actual architecture by a simpler network. Algebraic consequences of the reference readout are distinguished from additional sensitivity and norm assumptions needed to specialize the risk bound.

I.8.1 Parameter accounting and the terms they control

Continue to use Q for the number of global kernels, λ_q for their bandwidth factors, $\bar{d}$ for the spectral dimension, R for the number of reference distributions at an attachment site, and $m \times d$ for the size of each reference matrix. In this section L counts the reference attachment sites in the concatenation $\mathbf{s} = \|_{l=1}^{L} \mathbf{s}^{(l)}$. If parallel branches have separate reference readouts, each such readout is a separate site. Sharing parameters can reduce the number of free parameters, but does not remove coordinates from the concatenated output. The scalar γ remains the reference-kernel parameter; γ_{A} is the alignment-kernel parameter. No target-test statistics are used to fit either spectral coordinates or alignment matrices.

Table 28: Where architectural and training parameters enter the bound.

Parameter	Principal terms and mechanisms
$Q, \lambda_q, \bar{d}$	Multi-scale input norm, kernel and spectral errors, $\sum_q \mathfrak{a}_q$, and the sizes of the input-processing GIN branches.
R, m, d, L	Number of free reference parameters, block KL, reference-readout sensitivity, and the norm of the concatenated graph vector.
GIN/GT depths and widths	Parameter counts and displacements, intermediate sensitivities J_l, pooling sensitivity J_{p}, and pooling norm.
Attention heads	Attention parameter dimensions and the sensitivities of complete attention blocks; head count alone does not determine either.
$\gamma, \gamma_{\mathrm{A}}$	Reference sensitivity and $\log\gamma$ KL; alignment discrepancy and its conversion to a mean-mismatch bound.
ϱ, B, b	Loss range L_B, perturbation/transfer coefficients, class coverage κ_T, and the block effective size S_{eff}.
$M, N_j, N_{\mathrm{train},c}$	Source estimation and target-head estimation; these counts do not replace the residual target-domain discrepancy.
η, iterations, learning rates	Realized alignment progress, empirical risk, parameter displacement, posterior sensitivity, and optimization error; not additional independent statistical observations.

I.8.2 Blockwise PAC–Bayes complexity and model size

Proposition I.28 (A module-wise Gaussian complexity decomposition)**.** *Partition the free network coordinates into disjoint blocks indexed by $h \in \mathcal{H}_{\mathrm{par}}$, with p_h coordinates in block h. Suppose the prior and posterior factorize over these blocks, and within a block are isotropic Gaussians:*

$$\Pi_h = \mathcal{N}(\boldsymbol{\omega}_{0,h}, s_h^2 I_{p_h}), \qquad \mathsf{Q}_h = \mathcal{N}(\widehat{\boldsymbol{\omega}}_h, \sigma_h^2 I_{p_h}), \qquad s_h, \sigma_h > 0.$$

The prior centers and scales obey the independence requirement of Assumption I.5. Then

$$\mathrm{KL}(\mathsf{Q}\|\Pi) = \sum_{h\in\mathcal{H}_{\mathrm{par}}} \left\{ \frac{\|\widehat{\boldsymbol{\omega}}_h - \boldsymbol{\omega}_{0,h}\|_2^2}{2s_h^2} + \frac{p_h}{2}\left(\frac{\sigma_h^2}{s_h^2} - 1 + \log\frac{s_h^2}{\sigma_h^2}\right)\right\}. \tag{143}$$

In particular, independent blocks for $\mathcal{W}_{\mathrm{GIN}}$, $\mathcal{W}_{\mathrm{GT}}$, $\mathcal{V}$, and $\log\gamma$ give the sum of four corresponding divergences. Any other learned encoder tensors must also be included in one of the blocks.

Proof. For product distributions, the log density ratio is the sum of block log density ratios. Its posterior expectation is consequently the sum of block KL divergences. In block h, specialize (103) to common prior variance s_h^2 and common posterior variance σ_h^2. The sum of squared coordinate displacements is $\|\widehat{\boldsymbol{\omega}}_h - \boldsymbol{\omega}_{0,h}\|_2^2$, whereas the variance-mismatch summand is repeated p_h times. This gives (143). □

For *untied* reference matrices with common dimensions,

$$p_{\mathrm{RD}} = LRmd. \tag{144}$$

Thus an isotropic reference block has the explicit contribution

$$\begin{aligned}\mathrm{KL}_{\mathcal{V}} = {} & \frac{\|\widehat{\mathcal{V}} - \mathcal{V}_0\|_F^2}{2s_{\mathcal{V}}^2} \\ & + \frac{LRmd}{2}\left(\frac{\sigma_{\mathcal{V}}^2}{s_{\mathcal{V}}^2} - 1 + \log\frac{s_{\mathcal{V}}^2}{\sigma_{\mathcal{V}}^2}\right).\end{aligned} \tag{145}$$

Here the squared collection norm sums over all reference matrices. If the node-embedding dimension at site l is d_l, the count is $\sum_l Rmd_l$. Tied parameters are counted only once. The mean-shift and output-norm bounds below, however, still count all output sites.

GIN and GT parameters. To make the depth/width connection explicit without guessing the implementation, let $h_{q,l,t-1}$ and $h_{q,l,t}$ be the input and output widths of affine layer t in the MLP of GIN layer l in branch q. If there are ϑ_1 GIN layers, κ_1 affine layers per such MLP, and a bias in every affine layer, their parameter contribution is

$$p_{\mathrm{GIN,affine}} = \sum_{q=1}^{Q}\sum_{l=1}^{\vartheta_1}\sum_{t=1}^{\kappa_1}\left(h_{q,l,t-1}h_{q,l,t} + h_{q,l,t}\right). \tag{146}$$

Terms for absent biases are omitted; learned aggregation coefficients and normalization parameters, if present, are added. Input widths depend on d and on whether scales are processed separately or concatenated. Identical untied branches multiply their affine count by Q, whereas shared branches do not.

For the GT, count entries of the actual query, key, value, output-projection, feed-forward, normalization, and bias tensors over its ϑ_2 blocks. For example, in a conventional dense projection implementation with total width d_{GT}, four square attention projections contribute $4d_{\mathrm{GT}}^2$ weights per block before biases. Splitting a *fixed total width* into more heads does not multiply that particular count; keeping each head's width fixed and increasing the number of heads changes d_{GT} and hence the count. This is a conditional counting example, not an assertion that every tensor in GraphVec has this shape. If a feed-forward network has κ_2 affine layers per GT block, its weight-and-bias count is the sum of the corresponding width products and output widths, exactly as in (146), over $\vartheta_2\kappa_2$ affine layers. Use the actual count if depths differ.

The preprocessing matrices $\mathbf{R}_j^{(\lambda_q)}$ are not part of $\boldsymbol{\omega} = (\mathcal{W}, \mathcal{V}, \log\gamma)$ in this proof. Their data dependence is charged through the learned-map corrections, alignment term, and residual discrepancy, not silently ignored or counted as an additional network KL block. A different parameterization that places preprocessing in the hypothesis would need its own prior and complexity accounting.

Equation (143) puts these counts into $\mathcal{C}_{\mathrm{pre}}^a$ through (95), (100), and (102). More parameters do not by themselves determine the final certificate: empirical Gibbs loss, parameter displacement, variance mismatch, and deterministic/map corrections also change. If $\sigma_h = s_h$, the explicit variance-mismatch term is zero, but the displacement penalty and posterior risk need not be small. In particular,

expanding posterior variance is not guaranteed to reduce KL after it exceeds the prior variance. Prior scales selected using certification data require a valid mixture or selection penalty; they are not free tuning parameters.

I.8.3 Reference size, reference sensitivity, and graph-vector norm

Proposition I.29 (Reference-parameter sensitivity and output range)**.** *At one reference site fix the node matrix* $\mathbf{H}$*, the reference kernel* $k(x, x') = e^{-\gamma\|x-x'\|_2^2}$*, and two collections* $\mathcal{V} = \{\mathbf{V}_b\}_{b=1}^R$ *and* $\mathcal{V}' = \{\mathbf{V}'_b\}_{b=1}^R$ *with* m *nodes per reference. With a fixed correspondence between their reference rows,*

$$\|\mathbf{s}(\mathbf{H}, \mathcal{V}) - \mathbf{s}(\mathbf{H}, \mathcal{V}')\|_2 \le \sqrt{\frac{2\gamma}{m}}\,\|\mathcal{V} - \mathcal{V}'\|_F. \tag{147}$$

Every reference coordinate also satisfies

$$-\sqrt{2} \le s_b^{(l)} \le 0. \tag{148}$$

Consequently, for L *sites and* R *coordinates per site,*

$$\|\mathbf{g}\|_2^2 = \|\mathbf{s}\|_2^2 + \|\mathbf{p}\|_2^2 \le 2LR + \|\mathbf{p}\|_2^2. \tag{149}$$

If the comparison pooling vector in Appendix I.6 satisfies $\|\mathbf{p}_0(G)\|_2 \le B_{\mathrm{p}}$*, uniformly over its population inputs, then a valid bound for the bias-augmented comparison feature is*

$$A_{\mathrm{arch}} = \sqrt{1 + 2LR + B_{\mathrm{p}}^2}. \tag{150}$$

For the statistical head bound, B_{p} *is fixed independently of the target training set, possibly conditionally on the source training data.*

Proof. For one reference b, the reverse triangle inequality for kernel mean embeddings implies

$$|\,\mathrm{MMD}(\mathbf{H}, \mathbf{V}_b) - \mathrm{MMD}(\mathbf{H}, \mathbf{V}'_b)| \le \left\| \frac{1}{m}\sum_{r=1}^{m} [\phi(\mathbf{v}_r) - \phi(\mathbf{v}'_r)] \right\|.$$

The Gaussian feature-map identity in the proof of Proposition I.27 gives $\|\phi(v) - \phi(v')\| \le \sqrt{2\gamma}\|v - v'\|_2$. Triangle and Cauchy–Schwarz inequalities therefore bound the last display by $\sqrt{2\gamma/m}\|\mathbf{V}_b - \mathbf{V}'_b\|_F$. Squaring and summing over b proves (147).

For any two empirical probability measures P, V, the two within-measure terms in $\mathrm{MMD}^2(P, V)$ are at most one each, and the cross term being subtracted is nonnegative, since $0 < k \le 1$. Thus $0 \le \mathrm{MMD}^2(P, V) \le 2$, proving (148). GraphVec concatenates LR such coordinates and $\mathbf{p}$; summing their squares proves (149). Adding a bias coordinate of value one and using the assumed comparison-pooling bound proves (150). □

Neither the coordinate range nor its norm bound grows with m or d: MMD averages reference nodes instead of summing them. This does not make m, d statistically cost-free; they enter the learned-parameter KL through (144). Also, (147) improves as $m^{-1/2}$ only for a *fixed total* Frobenius perturbation. For iid perturbations of variance σ^2 per reference coordinate, its squared-norm upper bound has expectation $(2\gamma/m)(Rmd\sigma^2) = 2\gamma Rd\sigma^2$ at one site. Thus increasing m alone does not give a $1/\sqrt{m}$ statistical rate. These learned reference nodes are not new independent training graphs.

For a head using normalized graph vectors, the bias-augmented comparison input has norm $\sqrt{2}$, provided the comparison graph vectors are nonzero. Then use $A_0 = \sqrt{2}$ rather than (150), and set $\epsilon_{\mathrm{rad}} = 0$. For a raw-feature head, retain (129). In either case the classifier norm constraint and the comparison-head feasibility requirement of Theorem H.1 remain. The head complexity depends on the product $B_{\mathrm{c}}A_0$, so changing feature scale does not establish a benefit unless the head scale is treated consistently.

I.8.4 A LAYER-WISE SPECIALIZATION OF COMMON-SHIFT SENSITIVITY

Proposition I.30 (From backbone responses to an alignment penalty)**.** *Fix the deployed parameters $\widehat{\boldsymbol{\omega}}$. For an n-node graph define $\mathbf{H}^{(l)}(u)$ and $\mathbf{p}(u)$ as the node matrix at reference site l and the final pooling vector, respectively, when the concatenated spectral input is $\mathbf{Z}+\mathbf{1}u^\top$. Suppose these maps are locally Lipschitz and, along every line segment used in the comparisons of Theorem I.15, have almost-everywhere bounds*

$$\|D_u\mathbf{H}^{(l)}(u)\|_{2\to F}\le\sqrt{n}\,J_l,\qquad \|D_u\mathbf{p}(u)\|_2\le J_{\mathrm{p}},\qquad \|\mathbf{g}(u)\|_2\ge s_0>0. \tag{151}$$

The constants hold uniformly over the relevant graphs and segments; $\|\cdot\|_{2\to F}$ is the operator norm from Euclidean input to Frobenius output. Let

$$L_{\mathrm{raw,shift}}=\left(2\gamma R\sum_{l=1}^{L}J_l^2+J_{\mathrm{p}}^2\right)^{1/2}. \tag{152}$$

Then Assumption I.13 holds with

$$L_{\mathrm{shift}}\le\frac{L_{\mathrm{raw,shift}}}{s_0}. \tag{153}$$

Consequently, a valid architecture-dependent replacement for the alignment term in (118) is

$$\mathcal{E}_{\mathrm{A}}^{\mathrm{arch}}=\frac{8}{\varrho s_0}\sqrt{\left(2\gamma R\sum_{l=1}^{L}J_l^2+J_{\mathrm{p}}^2\right)\frac{\sum_{q=1}^{Q}\mathfrak{a}_q}{1-e^{-4\gamma_{\mathrm{A}}}}}. \tag{154}$$

Proof. Integrate the first derivative bound in (151) on a segment from u to v:

$$\|\mathbf{H}^{(l)}(u)-\mathbf{H}^{(l)}(v)\|_F\le\sqrt{n}J_l\|u-v\|_2.$$

References and γ are held fixed in this input comparison. By Proposition I.27, the output of reference site l therefore satisfies

$$\|\mathbf{s}^{(l)}(u)-\mathbf{s}^{(l)}(v)\|_2\le\sqrt{2\gamma R/n}\sqrt{n}J_l\|u-v\|_2=\sqrt{2\gamma R}J_l\|u-v\|_2.$$

The pooling bound integrates to $\|\mathbf{p}(u)-\mathbf{p}(v)\|_2\le J_{\mathrm{p}}\|u-v\|_2$. The output is a concatenation, so its squared difference is the sum of the squared branch differences. This proves a raw common-shift Lipschitz bound $L_{\mathrm{raw,shift}}$.

Along each segment the raw output is absolutely continuous. At almost every point where its derivative exists, putting $\mathbf{e}=\mathbf{g}/\|\mathbf{g}\|$ gives

$$D_u\mathbf{e}=\frac{I-\mathbf{e}\mathbf{e}^\top}{\|\mathbf{g}\|_2}D_u\mathbf{g}.$$

The projection in the numerator has norm at most one; the denominator is at least s_0. Integrating proves (153). This also covers the possible nondifferentiability of unsquared MMD by using its Lipschitz bound and almost-everywhere derivatives along the segment. Substitution into (118) proves (154). □

If all $J_l\le J_{\mathrm{back}}$, one may further replace $\sum_l J_l^2$ by LJ_{back}^2. This exposes LR explicitly, but does not say that J_{back} or s_0 is constant as architecture or training changes. With site-specific kernels or reference counts, $2\gamma R\sum_l J_l^2$ becomes $2\sum_l\gamma_l R_l J_l^2$. The explicit $1/\sqrt{n}$ of the reference layer has canceled the $\sqrt{n}$ in the common-shift perturbation, but J_l, J_{p} can still depend on graph size, degrees, topology, and the fitted parameters. No graph-size-independent theorem follows from that cancellation alone.

How GIN depth and weights affect J_l. For illustration, suppose a GIN block is exactly $H\mapsto\mathrm{MLP}_l((\widetilde{A}+\epsilon_l I)H)$. If the MLP contains κ_1 affine layers with 1-Lipschitz activations, its input Lipschitz factor is bounded by

$$L_{\mathrm{GIN},l}\le\|\widetilde{A}+\epsilon_l I\|_2\prod_{t=1}^{\kappa_1}\|\mathbf{W}_{l,t}^{\mathrm{GIN}}\|_2. \tag{155}$$

Indeed, left multiplication bounds the Frobenius norm by $\|\widetilde{A} + \epsilon_l I\|_2$, each right affine multiplication by its weight spectral norm, biases cancel in differences, and each activation has Lipschitz factor at most one. Chain composition gives $J_l \le L_{\mathrm{GIN},l} J_{l-1}$ on a sequential branch, starting with unit directional factor for the spectral input. Any additional normalization, residual, or learned aggregation operation must be included in the block. Parallel-branch and concatenation factors must respect the actual inputs; their omission cannot be justified by only listing depth.

How GT blocks and pooling affect these factors. For a complete transformer block $g_\psi^{(l)}$, a derivative norm bound $L_{\mathrm{GT},l}$ on the relevant hidden-state paths gives the analogous chain factor $J_l \le L_{\mathrm{GT},l} J_{l-1}$. This is the derivative of the *complete* block: query/key/value and output projections, attention, residual connections, feed-forward layers, and layer normalization all matter. In particular, the derivative of $H + \mathrm{attn}(H)$ is $I + D\mathrm{attn}(H)$, not just the attention derivative. For mean pooling from a final node matrix, $\|n^{-1}\mathbf{1}^\top \Delta H\|_2 \le \|\Delta H\|_F/\sqrt{n}$, so its common-shift factor is at most the corresponding final J_l. For sum pooling the analogous factor is nJ_l. Other pooling mechanisms require their own bound.

Global layer-norm products provide one conservative way to establish these bounds, but substituting them can recover the looseness of the old analysis. The new result permits solution-dependent bounds restricted to the relevant comparison directions. Empirical Jacobians or finite-difference measurements are diagnostic evidence, not uniform certificates for (151). Neither a positive s_0 nor a small J_l follows from the architecture specification alone.

I.8.5 The specialized classifier and end-to-end bound

Define

$$H_{\mathrm{c}}^{\mathrm{arch}} = \log C_T + 2B_{\mathrm{c}} A_{\mathrm{arch}}, \tag{156}$$

$$\mathfrak{G}_{\mathrm{c}}^{\mathrm{arch}} = 4B_{\mathrm{c}} A_{\mathrm{arch}} \sqrt{\frac{C_T}{N_{\mathrm{eff}}^{\mathrm{c}}}} + H_{\mathrm{c}}^{\mathrm{arch}} \sqrt{\frac{\log(2/\delta_{\mathrm{c}})}{2N_{\mathrm{eff}}^{\mathrm{c}}}}. \tag{157}$$

The r coordinates of the graph vector are LR reference coordinates plus $\dim(\mathbf{p})$ pooling coordinates, when no additional output operation is used. Although the vector-contraction bound above has no separate r factor, r influences norms and parameterization. The finite-net alternative in Proposition I.32 explicitly contains $C_T(r+1)$.

Corollary I.31 (Architecture-dependent form of the complete guarantee)**.** *Retain every statistical, sampling, feasibility, and optimization condition of Theorem H.1. In addition, assume the uniform comparison-pooling bound in Proposition I.29 and the common-shift bounds in Proposition I.30. Put*

$$\begin{aligned} U_{T,\mathrm{arch}}^{a} = \min\big\{ & L_B, \mathcal{C}_{\mathrm{pre}}^{a}(\delta_{\mathrm{pre}}) + \mathcal{E}_{\mathrm{A}}^{\mathrm{arch}} + \Delta_{\mathrm{res}}^{a} \\ & + \mathbf{1}\{a = \mathrm{UCL}\}\epsilon_{\mathrm{aug}\to\mathrm{sem},T} \big\}. \end{aligned} \tag{158}$$

The source certificate can use the block KL in (143), retaining all map, deterministic-predictor, and Monte Carlo corrections when applicable. With the same failure probability as in Theorem H.1,

$$\begin{aligned} \mathcal{R}_T^{\mathrm{CE}}(\widehat{\mathbf{W}}_{\mathrm{c}}) \le \frac{U_{T,\mathrm{arch}}^{a}}{\kappa_T(B,\boldsymbol{\pi})} + \epsilon_{\mathrm{rad}} + 2\mathfrak{G}_{\mathrm{c}}^{\mathrm{arch}} \\ + 2\sqrt{2}B_{\mathrm{c}}(d_{\mathrm{emp}} + d_{\mathrm{pop}}) + \epsilon_{\mathrm{opt}}. \end{aligned} \tag{159}$$

For a normalized-vector head, replace A_{arch} by $\sqrt{2}$ and set $\epsilon_{\mathrm{rad}} = 0$, subject to that version's comparison features and head-radius condition. The classification-error conversion in (69) remains valid.

Proof. Proposition I.30 proves the common-shift condition needed by Theorem I.15, with the explicit alignment penalty (154). Proposition I.29 supplies a valid comparison-feature radius in the fixed-feature head proof; repeating that proof with this radius gives (157). The KL identity only specializes the same source certificate. Substitute these three statements into the proof of Theorem H.1. The bridge, radial correction, learned-map comparison, and optimization steps are unchanged. If any additional bounds on J_l, s_0, B_{p} or other terms hold only on extra high-probability events, their failure probabilities must also be charged; the corollary does not make data-selected radii cost-free. □

I.8.6 How kernel scales and spectral dimension affect map errors

Bandwidth and scale count. Proposition I.8 gives the per-scale sensitivity $1/(2\lambda_q)$ to normalized squared-distance perturbations. Larger λ_q reduces this particular coefficient, but also changes empirical contrastive risk, eigenvalues, and retained task information. Directly from the kernel formula, $\mathbf{K}_{\lambda_q} \to \mathbf{1}\mathbf{1}^\top$ as $\lambda_q \to \infty$ on a fixed finite point set; a small perturbation coefficient is therefore not an argument for arbitrarily large bandwidth. Increasing Q changes the input norm bound to $\sqrt{Q}$ and adds terms to $\sum_q \mathfrak{a}_q$, as well as parameters for untied branches. Holding other quantities fixed and writing $\overline{\mathfrak{a}} = Q^{-1}\sum_q \mathfrak{a}_q$ makes the alignment contribution proportional to $\sqrt{Q\overline{\mathfrak{a}}}$. This is not a monotonicity claim when L_{shift}, the learned weights, or the residual shift also change.

Spectral truncation and conditioning. For a fixed kernel, the Gram residual $\sum_{r>\bar{d}} \sigma_r^2$ decreases as $\bar{d}$ increases, whereas the training-anchor feature-space projection residual is $\bar{N}^{-1}\sum_{r>\bar{d}} \sigma_r$; these are different errors. The stability bound (139) depends also on $\delta_{\mathrm{sp}} = \sigma_{\bar{d}} - \sigma_{\bar{d}+1}$ and $a = \sigma_{\bar{d}}$. Increasing $\bar{d}$ can retain smaller positive eigenvalues and does not guarantee a larger gap. The out-of-sample formula (140) exposes the additional inverse-eigenvalue factors. Thus a larger spectral representation can preserve more relationships while becoming more sensitive to kernel estimation and numerical perturbations. There is no direct implication that all retained relationships are useful for the downstream labels.

From node-map errors to the displayed statistical corrections. For a same-anchor comparison at scale q, let $\epsilon_{\mathrm{sp},q}(x)$ denote the right side of (140) and let $\epsilon_{R,q} = \|\widehat{R}_q O_q - R_{0,q}\|_2$. The unit node norm and (141) give

$$\|\widehat{z}_q(x)^\top \widehat{R}_q^\top - z_{0,q}(x)^\top R_{0,q}^\top\|_2 \le \epsilon_{\mathrm{sp},q}(x) + \epsilon_{R,q}. \tag{160}$$

Consequently a uniform multi-scale node error is

$$d_Z = \left[\sum_{q=1}^{Q} \big(\sup_x \epsilon_{\mathrm{sp},q}(x) + \epsilon_{R,q}\big)^2\right]^{1/2}. \tag{161}$$

If a raw encoder satisfies $\|F(A,Z) - F(A,Z')\|_2 \le L_Z^{\mathrm{raw}}\|Z - Z'\|_F/\sqrt{n}$ on these comparisons, then

$$d_{\mathrm{emp}}, d_{\mathrm{pop}} \le L_Z^{\mathrm{raw}} d_Z. \tag{162}$$

Indeed, each of the n node-row errors is at most d_Z, so the Frobenius input error is at most $\sqrt{n}d_Z$; apply the assumed raw sensitivity and average. The analogous normalized-encoder bound gives the source map corrections described after (100). Unlike L_{shift}, L_Z^{raw} concerns arbitrary node-wise perturbations and is not controlled by the common-shift result alone. All these comparisons require the same anchor system and an actual rotation error bound. Replacing them by population sample-size rates for independently resampled anchors needs an operator sampling and alignment stability analysis, not just the finite-anchor result.

The number of Nyström anchors changes computational cost, the approximation, and its spectral conditioning, but is not automatically an independent sample count: anchors taken from one graph can be dependent. The empirical and population map errors in the theorem remain explicit until appropriate estimator-specific bounds have been established.

I.8.7 Temperature, alignment bandwidth, batch size, and optimization

Contrastive temperature ϱ. The exact loss range is $L_B = \log(B-1) + 2/\varrho$, and the representation and residual-transfer coefficients carry $1/\varrho$. The raw-head radial correction and the norm of the normalized mean classifier also depend on $1/\varrho$; in particular its guaranteed feasibility uses $B_{\mathrm{c}} \ge \sqrt{C_T}/\varrho$ for normalized graph vectors. A smaller temperature increases these worst-case sensitivity terms at fixed representations, but it changes the trained representation and the empirical loss as well. The bound therefore does not select an optimal temperature by inspecting only $1/\varrho$.

Two distinct Gaussian parameters. For the reference kernel, γ enters $\sqrt{2\gamma R\sum_l J_l^2 + J_{\mathrm{p}}^2}$ and the KL of $\log\gamma$. The reference-coordinate amplitude bound remains $\sqrt{2}$ for every $\gamma > 0$. For the

alignment kernel, both the numerator $\mathfrak{a}_q$ and denominator $1 - e^{-4\gamma_A}$ depend on γ_A. Holding means and rotations fixed and putting $d_{jq} = \|\mathbf{m}_T^{(q)} - \mathbf{m}_j^{(q)}\|_2$, Taylor expansion of the finite sums gives

$$\lim_{\gamma_A \downarrow 0} \frac{\mathfrak{a}_q}{1 - e^{-4\gamma_A}} = \frac{1}{4M} \sum_{j=1}^{M} d_{jq}^2. \tag{163}$$

Specifically $1 - e^{-\gamma_A d_{jq}^2} = \gamma_A d_{jq}^2 + O(\gamma_A^2)$ and $1 - e^{-4\gamma_A} = 4\gamma_A + O(\gamma_A^2)$. Hence the ratio does not diverge simply because the denominator tends to zero. When γ_A also changes the optimized rotations, this fixed-configuration limit alone does not predict performance.

Batch size and independent sample counts. The loss bound uses B views, whereas $S_{\text{eff}} = M \min_j \lfloor N_j/b \rfloor$ uses b original graph records per block. Supervised batches have $b = B$ in the stated sampler; v views per original graph give $B = vb$. The random-batch bridge has its separate coefficient $\kappa_T(B, \boldsymbol{\pi})$, and rare classes or small batches can make it small. More candidate graphs can improve coverage, but neither κ_T nor the total certificate is asserted monotone in batch size here. The block effective size is specific to a conservative proof and is not evidence that larger batches necessarily generalize worse. Repeated augmentations do not multiply N_j.

The target-classifier effective size is $N_{\text{eff}}^{c} = (\sum_c \pi_c^2 / N_{\text{train},c})^{-1}$ for the stated stratified design. With k examples per class and uniform risk weights it is $C_T k$. Increasing the number of source graphs affects source estimation, whereas their residual relationship to a specified target domain is retained in Δ_{res}^{a}. Neither more source domains nor more target labels automatically removes representation incompatibility or the augmentation-to-semantic correction.

Alignment and network optimization parameters. With source means and encoder fixed, the target-only update proved above reduces $\mathfrak{a}_q$ under its stated conditions. Its proximal parameter η affects the attained rotations and progress per iteration, not the number of independent observations. Learning rates, weight decay, and pretraining duration enter this analysis through the resulting weights, empirical risk, KL, perturbation sensitivity, and residual shift. They have no separate closed-form guarantee in the present theorem. For example, weight decay is not itself a proved bound on J_l or $\|\widehat{\mathcal{W}} - \mathcal{W}_0\|_F$. The target-head training length can affect ϵ_{opt}; a finite number of optimization epochs does not justify setting that term to zero. Different regularized training objectives require the corresponding comparator correction as already specified after Theorem I.22.

Comparing hyperparameters without overstating the conclusion. A larger $R, m, \bar{d}, Q$ or depth can reduce empirical/approximation error while increasing statistical or sensitivity penalties. A valid comparison therefore evaluates all affected terms under the same confidence and sampling conventions. If selecting an architecture, head radius, prior scale, or other non-uniformly covered choice from the same certification data, use a predeclared family with the appropriate selection penalty. To interpret a measured curve, report at least its final centered risk, blockwise KL, spectral tail and conditioning, alignment discrepancies, raw output norms, and relevant head norms. Empirical sensitivity diagnostics must be labeled as diagnostics unless upgraded to uniform bounds. The architecture-dependent specialization explains identifiable tradeoffs; it does not, by itself, prove a numerically nonvacuous certificate or an optimal choice of these parameters.

I.9 A FULLY ELEMENTARY ALTERNATIVE FOR THE CLASSIFIER BOUND

Proposition I.32 (Finite-net classifier bound)**.** *Under the fixed-feature setup of Lemma I.21, put $p_c = C_T(r+1)$. For any fixed $u > 0$, with probability at least $1 - \delta_c$,*

$$\sup_{W \in \mathcal{H}_T} |\mathcal{R}_0(W) - \widehat{\mathcal{R}}_0(W)| \le 2\sqrt{2} A_0 u + H_c \sqrt{\frac{p_c \log(1 + 2B_c/u) + \log(2/\delta_c)}{2N_{\text{eff}}^{c}}}. \tag{164}$$

The head and complete theorems remain valid with this expression in place of $\mathfrak{G}_c$. To select the smaller of two independently proved bounds after seeing the data, allocate failure probability to both events.

Proof. Identify the matrix ball with a Euclidean ball in $\mathbb{R}^{p_c}$. Choose a maximal u-separated set with centers in the ball. The radius-$u/2$ balls about its centers are disjoint and lie in a radius-$B_c + u/2$

ball. Comparing volumes bounds the number of centers by $(1 + 2B_c/u)^{p_c}$. Maximality makes it a u-net.

For a fixed center, the independent weighted losses are in $[0, H_c]$. The two-sided Hoeffding tail is at most $2\exp[-2t^2/(H_c^2 \sum_i w_i^2)]$. Apply a union bound over net centers and solve for t using total failure δ_c. For any W choose a center W_0 within Frobenius distance u. The logit change is at most $A_0 u$ in Euclidean norm, and cross-entropy changes by at most $\sqrt{2}A_0 u$, both in the population and empirical average. Add the two approximation errors to the finite-net tail. □

I.9.1 Elementary concentration facts used above

For completeness, if X lies in an interval of length c, define $g(\lambda) = \log \mathbb{E}e^{\lambda(X-\mathbb{E}X)}$. Under exponential tilting, $g''(\lambda)$ is a variance of a variable in the same interval. That variance is at most $c^2/4$, since squared distance to the interval midpoint is at most $c^2/4$, and variance minimizes expected squared distance over centers. Since $g(0) = g'(0) = 0$, integration gives

$$\mathbb{E}e^{\lambda(X-\mathbb{E}X)} \le e^{\lambda^2 c^2/8}.$$

Independence and Markov's inequality then imply, for a weighted sum of independent $[0, H]$ variables,

$$\mathbb{P}\left(\left|\sum_i w_i(X_i - \mathbb{E}X_i)\right| \ge t\right) \le 2e^{-2t^2/(H^2 \sum_i w_i^2)}.$$

For a function of independent observations with coordinate replacement bounds c_i, expose observations successively and use the Doob martingale. Each conditional martingale increment has a conditional range of length at most c_i. Apply the same exponential bound conditionally and iterate the expectations. Markov optimization gives the one-sided bounded-difference bound $\exp[-2t^2/\sum_i c_i^2]$ used in Lemma I.21. Thus the only external complexity ingredient in that particular head proof is the precisely stated vector-contraction inequality; Proposition I.32 provides an alternative without it.