SPR: A GRAPH FOUNDATION MODEL FOR TRANSFERABLE GRAPH COGNITION THROUGH SPECTRAL PATTERNS AND RELATIONAL GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, Graph Foundation Models (GFMs) have attracted increasing attention for their great potential to learn unified and generalizable knowledge across diverse graphs, thereby supporting a wide range of open-ended graph scenarios. However, unlike natural language and images, graphs are less intuitive for humans to understand. Therefore, despite many initial explorations, a key problem remains unresolved: the transferable cognitive mechanism for graphs. Existing GFMs usually adopt intuitively defined mechanisms to uniformly encode transferable graph patterns, such as handcrafted structural templates (e.g., cycles and trees), fixed propagation operators, and predefined random-walk patterns. However, these mechanisms are often prescriptive and rigid, which may limit their ability to simultaneously model diverse graph patterns. This motivates the exploration of a more flexible, graph-native, and transferable cognitive mechanism. To achieve this, we analyze graphs from the spectral perspective, which provides a natural view of graphs, and propose SPR. SPR decomposes graph information into Chebyshev polynomial bases, facilitating the unified cognition of continuously varying graph spectral patterns. We further model cross-graph relational geometry to enhance the transferability of such graph cognition. Extensive experiments across diverse graphs and downstream scenarios demonstrate that SPR achieves more effective and transferable graph cognition.

1 INTRODUCTION

Graph data are prevalent in diverse real-world scenarios, such as recommender systems (Ying et al., 2018) and social networks (Perozzi et al., 2014), where entities interact through complex relations. Graphs provide a natural way to model such relational data by treating entities as nodes and their interactions as edges (Wu et al., 2021; Zhou et al., 2020). To learn from such data, Graph Neural Networks (GNNs) have become a dominant paradigm by leveraging message passing to aggregate information from neighboring nodes (Kipf & Welling, 2017). Yet conventional GNNs are typically developed for specific graphs or tasks, limiting their ability to reuse learned knowledge across different graph domains. Inspired by the transferability of foundation models in language and vision, recent research has moved toward Graph Foundation Models (GFMs), which aim to acquire transferable knowledge from multiple graphs and generalize it to diverse graph scenarios. (Wang et al., 2025b).

Following this trend, recent GFMs have explored several routes to enable knowledge sharing across diverse graphs. One direction seeks to unify graph semantics, typically by translating node and edge attributes into a common semantic space through textualization or language-model-based encoding, thereby reducing the incompatibility of feature spaces across domains (Liu et al., 2023a; He et al., 2025). A second line focuses on structural transfer, where graph-specific topology is abstracted into reusable structural units or encodings, such as computation-tree vocabularies, task trees, universal structural encodings, and other topology-derived descriptions (Wang et al., 2024; 2025c; Chen et al., 2026). In parallel, another stream aims to bridge different graph tasks by reformulating node-, edge-, and graph-level prediction into shared learning interfaces and adapting pre-trained knowledge through graph prompting, instruction tuning, or lightweight task-specific modules (Liu et al., 2023b;a; He et al., 2025). Despite their different formulations, these approaches share a common design principle:

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.

they first define a reusable form of graph knowledge and then build cross-graph pre-training and adaptation around it.

Despite these initial explorations, a key problem remains unresolved: the transferable cognitive mechanism for graphs. Unlike natural language and images, graphs are less intuitive for humans to understand and lack an obvious common form through which their underlying patterns can be perceived and organized. This difficulty is further amplified by substantial variations in domain-specific semantics and graph topology (Wu et al., 2021; Zhou et al., 2020). Consequently, cross-graph transfer requires more than learning a shared representation; it calls for a graph-native principle that determines how graph knowledge should be perceived, identified and organized as transferable across different domains. We refer to such a principle as a transferable cognitive mechanism for graphs. Existing GFMs have instantiated such mechanisms through different intuitively defined priors, including predefined structural vocabularies composed of trees and cycles, fixed propagation mechanisms, and random-walk-based structural descriptions (Sun et al., 2025b; Wang et al., 2026; Kim et al., 2025). Despite their different forms, these mechanisms prescribe particular ways of perceiving and encoding graph knowledge, leaving the underlying cognitive mechanism largely determined by human-designed priors.

This issue poses two key challenges: identifying an appropriate transferable cognitive mechanism for graphs, and enabling such a mechanism to be learned through multi-graph pre-training. The first challenge concerns what form this mechanism should take. Rather than being confined to a small set of predefined graph patterns, it should be sufficiently flexible to accommodate patterns that vary across different domains. As shown in Figure 1, raw node features exhibit markedly different spectral distributions across datasets (Zhang et al., 2023; Dong et al., 2025), suggesting that informative patterns may occupy different and continuously varying spectral regions rather than a few predetermined resolutions. The second challenge concerns how transferability can emerge from pre-training rather than being manually imposed in advance. Learning from multiple graphs should therefore go beyond fitting each graph independently and uncover common organization across them. In particular, similar patterns may play different roles depending on their relations to other patterns, making relational organization an important basis for cross-graph transfer.

To address these challenges, we propose SPR, a graph foundation model that develops a flexible and transferable cognitive mechanism from the spectral perspective. For the first challenge, the spectral domain provides a continuous space in which graph patterns with different characteristics can be expressed under a unified view, avoiding the need to prescribe a small set of discrete patterns or propagation scales. Based on this perspective, SPR decomposes graph information into Chebyshev polynomial bases and learns shared spectral responses, enabling continuously varying spectral patterns to be adaptively perceived across graphs. For the second challenge, SPR further learns cross-graph relational geometry during multi-graph pre-training. Rather than modeling the organization within each graph independently, it characterizes intrinsic pairwise relations in individual graphs and distills their common relational organization into a shared prototype geometry. This geometry provides a common relational reference that can be instantiated on different graphs, thereby supporting the transfer of learned graph cognition to unseen domains.

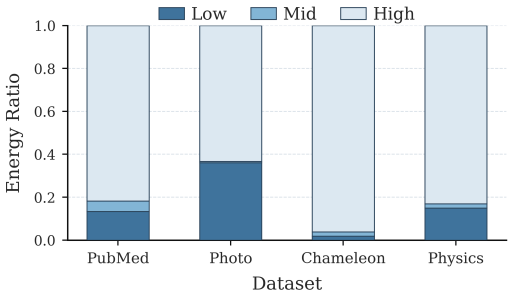


Figure 1: Spectral energy distribution of raw node features.

Our contributions are summarized as follows:

- We investigate transferable cognitive mechanisms for graphs and introduce a spectral perspective for understanding graph knowledge across domains, providing a flexible and graph-native view of diverse graph patterns.
- We instantiate this perspective in SPR through shared Chebyshev spectral responses that adaptively capture continuously varying graph patterns, together with cross-graph relational geometry that distills shared pairwise organization into a prototype geometry for knowledge transfer.

- Extensive experiments on diverse source and unseen graphs, covering both node- and graph-level tasks, demonstrate the effectiveness and transferability of SPR.

2 Related Work

Graph Foundation Models. Graph foundation models (GFMs) aim to learn reusable knowledge that can generalize across graphs and tasks. Early studies focus on reducing discrepancies in feature spaces and task formulations through unified representations, language-based encoders, or shared pre-training frameworks (Liu et al., 2024; He et al., 2025; Zhao et al., 2024a). Subsequent methods place increasing emphasis on structural transfer, introducing reusable structural vocabularies, graph prompts, structural tokens, and adaptive architectures to capture patterns that persist across different graphs (Wang et al., 2024; Yu et al., 2025; Chen et al., 2025). More recent work further explores graph-native forms of transferable knowledge, including recurring structural patterns, topology-aware representations, and shared geometric spaces (Sun et al., 2025a; 2026; He et al., 2026).

Spectral Graph Learning. Spectral graph learning represents graph signals through functions defined over the spectra of graph adjacency or Laplacian matrices (Bruna et al., 2014; Ortega et al., 2018). Early spectral GNNs parameterize filters directly in the spectral domain, while polynomial approximations such as Chebyshev filters avoid eigendecomposition and enable efficient localized computation (Defferrard et al., 2016). Later work develops adaptive polynomial filters and learnable frequency responses to better accommodate different graph structures and signal characteristics (Zhang et al., 2023; Dong et al., 2025). Recent studies further explore multi-resolution spectral bases and more flexible spectral parameterizations; for example, WaveGC learns general graph spectral wavelets through Chebyshev-order decomposition to model information over different spatial ranges (Liu et al., 2025).

Gromov–Wasserstein for Graph Learning. Gromov–Wasserstein (GW) distance compares relational spaces through their intrinsic pairwise structure without requiring explicit correspondence between their elements (Mémoli, 2011; Peyré et al., 2016). Early graph applications mainly use GW for graph matching, comparison, and soft structural alignment (Titouan et al., 2019). Its use has since expanded to graph coarsening, representation alignment, and structure-aware learning across graphs with different node sets or feature spaces. Recent work further applies GW to cross-graph transfer: GTRANS aligns and transfers structural patterns between source and target graphs through GW optimal transport, while related methods use fused GW to align graph representation spaces in distributed learning settings (Wang et al., 2025a; Fu et al., 2025).

3 Preliminaries

Gromov–Wasserstein Geometry. Gromov–Wasserstein (GW) geometry compares relational spaces through their intrinsic pairwise structure without requiring element-wise correspondence (Mémoli, 2011). For two graphs represented by structural cost matrices $C_1 \in \mathbb{R}^{n_1 \times n_1}$ and $C_2 \in \mathbb{R}^{n_2 \times n_2}$, with node distributions $p \in \Delta^{n_1}$ and $q \in \Delta^{n_2}$, the discrete GW discrepancy is

$$\mathrm{GW}(C_1, C_2, p, q) = \min_{T \in \Pi(p,q)} \sum_{i,i',j,j'} \left| C_1(i,i') - C_2(j,j') \right|^2 T_{ij} T_{i'j'}, \tag{1}$$

where $\Pi(p,q) = \{T \geq 0 \mid T\mathbf{1} = p,\ T^\top \mathbf{1} = q\}$. The transport T provides a soft correspondence between the two spaces, while the objective favors correspondences that preserve pairwise relations. This property makes GW suitable for comparing graphs with different node sets and feature spaces (Mémoli, 2011; Titouan et al., 2019).

Spectral Graph Filtering. Let $L = I - D^{-1/2} A D^{-1/2} = U \Lambda U^\top$ be the normalized graph Laplacian, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains Laplacian frequencies. A spectral filter acts on a graph signal X as

$$g_\theta(L) X = U g_\theta(\Lambda) U^\top X, \tag{2}$$

where g_θ specifies the response applied to different spectral components (Bruna et al., 2014; Ortega et al., 2018). Polynomial parameterizations such as Chebyshev filters approximate this response

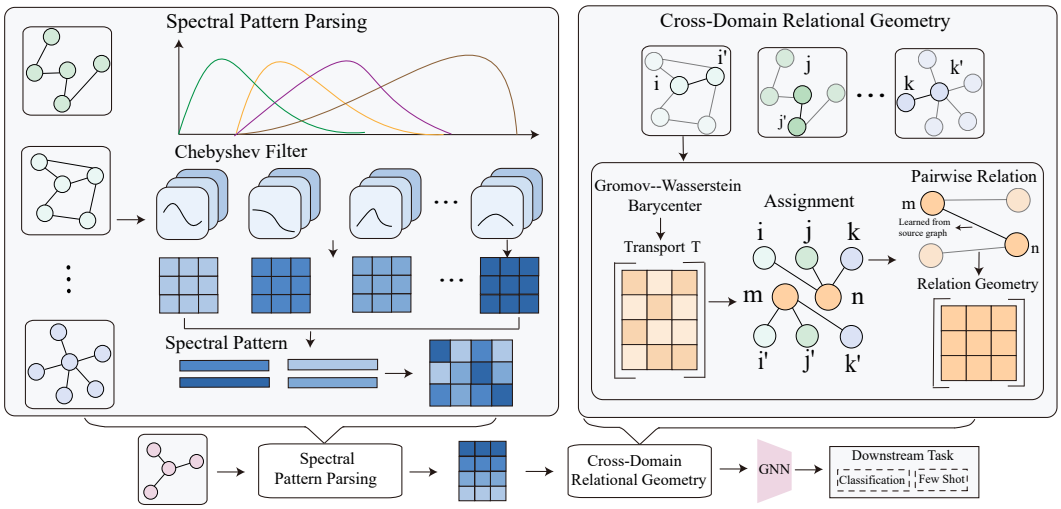


Figure 2: Overview of SPR.

without explicitly computing the eigendecomposition (Defferrard et al., 2016). In SPR, the spectral frontend is implemented on the normalized propagation operator $S = D^{-1/2}AD^{-1/2} = I - L$; hence a Laplacian frequency λ corresponds to the propagation-domain eigenvalue $\rho = 1 - \lambda$.

4 METHOD

SPR adopts a spectral view of transferable graph cognition with two components: spectral pattern parsing and cross-graph relational geometry. The former learns shared spectral responses instantiated on graph-specific operators, providing a unified yet adaptive representation across spectral regions. The latter aggregates intrinsic pairwise relations from source graphs into a shared prototype geometry. Both are integrated into a shared encoder for multi-graph self-supervised pre-training, as illustrated in Figure 2.

4.1 SPECTRAL PATTERN PARSING

Spectral pattern parsing represents graph information over a continuous spectral axis rather than through fixed structural units or propagation scales. As Figure 1 shows, different graphs exhibit distinct spectral characteristics. SPR therefore learns shared responses whose emphasis varies continuously over the spectrum, yielding a unified yet adaptive description across domains.

For a graph $G_m = (V_m, E_m)$ with node features $X_m \in \mathbb{R}^{n_m \times d_m}$, let A_m and D_m denote its adjacency and degree matrices. We use the normalized adjacency $S_m = D_m^{-1/2} A_m D_m^{-1/2}$ to characterize graph topology in the spectral domain. Instead of explicitly computing its eigendecomposition, SPR parameterizes spectral responses with Chebyshev polynomials, which can be evaluated efficiently through recursive propagation over S_m. Specifically, we construct the Chebyshev basis responses as

$$B_{m,0} = X_m, \qquad B_{m,1} = S_m X_m, \qquad B_{m,k} = 2S_m B_{m,k-1} - B_{m,k-2}, \quad k \geq 2. \tag{3}$$

Here, $B_{m,k} \in \mathbb{R}^{n_m \times d_m}$ denotes the k-th Chebyshev basis response and K is the maximum polynomial order. Each basis corresponds to a different polynomial transformation of S_m. Rather than associating individual basis orders with predetermined structural scales, SPR learns how these responses should be combined. For the α-th spectral filter, we compute

$$Z_{m,\alpha} = \sum_{k=0}^{K} \theta_{\alpha,k} B_{m,k}, \qquad \alpha = 1, \ldots, M_f, \tag{4}$$

where $\theta_{\alpha,k}$ denotes the learnable coefficient of the k-th basis, M_f is the number of spectral filters, and $Z_{m,\alpha}$ is the response produced by the α-th filter. The coefficients are shared across graphs.

The spectral interpretation of Eq. (4) becomes clearer by considering $S_m = U_m \operatorname{diag}(\rho_m) U_m^\top$, where ρ_m denotes the propagation-domain eigenvalues. Although this eigendecomposition is not explicitly computed, the filtering operation is equivalently written as

$$Z_{m,\alpha} = U_m g_\alpha(\rho_m) U_m^\top X_m, \qquad g_\alpha(\rho) = \sum_{k=0}^{K} \theta_{\alpha,k} T_k(\rho). \tag{5}$$

Here, U_m contains the eigenvectors of S_m, ρ denotes an eigenvalue of S_m, and $T_k(\rho)$ is the k-th Chebyshev polynomial. Since $S_m = I - L_m$, a Laplacian frequency λ corresponds to $\rho = 1 - \lambda$. The function $g_\alpha(\rho)$ therefore specifies the response of the α-th filter to different spectral components under the propagation-domain parameterization. Because g_α varies continuously with ρ, multiple shared response functions can emphasize complementary spectral regions without manually specifying frequency bands. Their realizations remain graph-dependent through S_m and X_m, allowing the same parameterization to transfer across graphs with different topology and feature spaces.

Since the filtered responses $Z_{m,\alpha}$ retain graph-specific feature dimensions, we apply the same truncated-SVD compression to each response and denote the resulting node-side rank-r embedding by $R_{m,\alpha} = \mathrm{SVD}_r^{\mathrm{node}}(Z_{m,\alpha}) \in \mathbb{R}^{n_m \times r}$. Here, $\mathrm{SVD}_r^{\mathrm{node}}(\cdot)$ denotes the $n_m \times r$ node representation returned by the rank-r truncated decomposition, using the same convention for every graph. This step only standardizes dimensionality across domains; the transferable spectral patterns are defined by the shared responses before compression. We then concatenate the reduced responses and project them into the encoder space:

$$X_{\mathrm{spec},m} = \mathrm{LN}\left([R_{m,1} \| \cdots \| R_{m,M_f}] W_{\mathrm{spec}} + b_{\mathrm{spec}}\right). \tag{6}$$

Here, $\|$ denotes concatenation, W_{spec} and b_{spec} are learnable projection parameters, and $\mathrm{LN}(\cdot)$ denotes layer normalization. The resulting representation integrates multiple learned spectral responses in a common feature space, providing a continuous and data-adaptive characterization of graph patterns across spectral regions.

4.2 Cross-Graph Relational Geometry

SPR further distills reusable relational organization from multiple source graphs through intrinsic pairwise geometry. Each graph is characterized by node-pair relations, whose common structure is aggregated into a shared prototype geometry. This avoids requiring aligned node identities or feature spaces and provides a common relational reference for transfer.

For each source graph G_m, we construct the heat kernel $H_m = \exp[-t_{\mathrm{gw}}(I - S_m)]$, where t_{gw} controls the diffusion scale. The corresponding diffusion-induced distance is

$$C_m(i,j) = \sqrt{H_m(i,i) + H_m(j,j) - 2H_m(i,j)}. \tag{7}$$

Here, $C_m \in \mathbb{R}^{n_m \times n_m}$ collects pairwise diffusion distances, providing an intrinsic graph description that does not require aligned node identities or feature dimensions.

Given the source geometries $\{C_m\}_{m=1}^M$, we learn a shared prototype geometry $C_\star \in \mathbb{R}^{p \times p}$ through a Gromov–Wasserstein barycenter:

$$C_\star = \arg\min_{C, \{T_m\}} \sum_{m=1}^{M} \lambda_m \sum_{i,i',k,l} \big(C_m(i,i') - C(k,l)\big)^2 T_m(i,k) T_m(i',l), \tag{8}$$

where p is the number of prototypes, w_m and b denote the node and prototype distributions, respectively, and $T_m \in \Pi(w_m, b) \subset \mathbb{R}^{n_m \times p}$ is the corresponding soft transport. The coefficient λ_m denotes the source-graph weight with $\sum_m \lambda_m = 1$ (Peyré et al., 2016). In practice, we optimize an entropically regularized GW approximation using the solver settings reported in Appendix D.

GW aligns pairwise relations rather than individual nodes: if (i, i') is softly associated with prototype pair (k, l), the objective encourages $C_m(i, i')$ and $C_\star(k, l)$ to agree. Thus, $C_\star$ represents recurring relational organization across source graphs.

This relational transfer admits a direct distortion guarantee.

Proposition 1 (Distortion-controlled geometry transfer)**.** *Let* $R_m = \mathrm{diag}(w_m)^{-1} T_m$ *and* $\overline{C}_m = R_m C_\star R_m^\top$*. Then*

$$\|C_m - \overline{C}_m\|^2_{w_m \otimes w_m} \le \mathcal{D}_{\mathrm{GW}}(C_m, C_\star, T_m). \tag{9}$$

Thus, a small GW distortion guarantees that the shared prototype geometry can be lifted back to each graph with bounded relational distortion. The proof is provided in Appendix A.1.

To incorporate the learned geometry into graph representations, we convert the prototype distances into relational affinities. Following our implementation, we define

$$A_\star(k,l) = \exp\left(-C_\star(k,l)/s_\star\right), \qquad s_\star = \mathrm{median}(C_\star), \tag{10}$$

followed by symmetrization and removal of self-connections. A diffusion kernel is then constructed over the resulting prototype graph:

$$K_\star = \exp\big(-L(A_\star)\big). \tag{11}$$

where $L(A_\star)$ is the graph Laplacian of the prototype affinity graph. The resulting $K_\star$ serves as the shared prototype interaction kernel.

For a source graph G_m, its correspondence with the shared prototype space allows the learned relational structure to be instantiated in the original node space:

$$\widetilde{P}_m = T_m K_\star T_m^\top. \tag{12}$$

The composition transfers prototype interactions back to the graph node space; a symmetrically normalized form of $\widetilde{P}_m$ is used by the encoder.

The same shared geometry naturally extends to unseen graphs. For a new graph G_{new}, we retain $C_\star$ and $K_\star$ and estimate a graph-to-prototype assignment T_{new}. We use GW alignment when the pairwise geometry is tractable; for large graphs, we use the scalable K-means-based sparse assignment described in Appendix D. The transferred relational structure is

$$\widetilde{P}_{\mathrm{new}} = T_{\mathrm{new}} K_\star T_{\mathrm{new}}^\top. \tag{13}$$

The shared prototype geometry remains fixed during transfer. The distortion guarantee above applies to GW-based alignment; the large-graph assignment is an efficiency-oriented approximation and is not covered by that guarantee.

The encoder combines local adjacency, graph-specific heat diffusion, and the transferred prototype operator through a node-wise gate.

4.3 Self-Supervised Pre-training

SPR is pretrained across the source graphs with node-level contrastive learning and masked spectral reconstruction. The shared spectral responses and encoder parameters are optimized jointly across graphs, while the prototype geometry is distilled from their graph-specific relational structures. Given two augmented graph views, the contrastive objective encourages corresponding nodes to maintain consistent representations:

$$\mathcal{L}_{\mathrm{node}} = -\frac{1}{N} \sum_i \log \frac{\exp(\mathrm{sim}(q_i, q_i')/\tau_c)}{\sum_j \exp(\mathrm{sim}(q_i, q_j')/\tau_c)}. \tag{14}$$

where q_i and q_i' denote the representations of node i from the two augmented views, $\mathrm{sim}(\cdot,\cdot)$ is the similarity function, and τ_c is the contrastive temperature.

To preserve the multi-scale information produced by spectral parsing, we hide the parsed spectral inputs of a node subset $\mathcal{M}$ before encoding and reconstruct their original, unmasked targets:

$$\mathcal{L}_{\mathrm{spec}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \|f_{\mathrm{rec}}(z_i) - x_{\mathrm{spec},i}\|_2^2, \tag{15}$$

where z_i is produced from the masked input, $x_{\mathrm{spec},i}$ is the corresponding unmasked spectral target, and f_{rec} is the reconstruction head. The overall pre-training objective is

$$\mathcal{L} = \lambda_{\mathrm{node}} \mathcal{L}_{\mathrm{node}} + \lambda_{\mathrm{spec}} \mathcal{L}_{\mathrm{spec}}. \tag{16}$$

where λ_{node} and λ_{spec} control the two objectives. The contrastive objective promotes representation consistency across graph perturbations, while spectral reconstruction encourages the encoder to preserve the spectral patterns extracted by spectral parsing. Implementation details and loss weights are provided in Appendix D.

Table 1: 1-shot performance comparison on node and graph classification. Results are accuracy (%, mean ± std); best and runner-up results are bolded and underlined, respectively.

	Self-supervised Pre-training			Graph Foundation Models					
	BGRL	GraphMAE2	GRACE	GCOPE	GraphAny	TIG	SAMGPT	GraphGlue	**SPR**
Source-Graph Node Classification									
Cora	36.85 ± 5.98	44.14 ± 9.22	41.34 ± 5.74	$\underline{50.59 \pm 9.05}$	39.40 ± 8.41	46.06 ± 10.78	39.65 ± 7.27	32.83 ± 6.24	$\mathbf{53.29 \pm 10.12}$
Photo	42.59 ± 9.21	57.88 ± 9.49	55.70 ± 8.40	49.90 ± 7.57	$\underline{61.53 \pm 9.87}$	53.95 ± 9.47	55.32 ± 9.01	48.33 ± 2.69	$\mathbf{64.63 \pm 10.19}$
CS	41.05 ± 7.42	65.56 ± 6.98	60.72 ± 6.94	63.04 ± 8.11	70.04 ± 6.99	$\underline{70.89 \pm 7.14}$	64.29 ± 8.41	67.43 ± 9.08	$\mathbf{71.99 \pm 6.87}$
Chameleon	21.16 ± 2.72	25.15 ± 4.77	23.52 ± 3.34	$\underline{27.66 \pm 4.41}$	24.91 ± 4.20	24.96 ± 4.82	24.94 ± 4.21	21.68 ± 1.79	$\mathbf{28.01 \pm 4.42}$
Held-out Graph Node Classification									
Citeseer	25.97 ± 3.21	38.86 ± 8.85	23.03 ± 2.74	39.58 ± 8.29	35.58 ± 6.99	$\underline{40.55 \pm 8.78}$	34.58 ± 7.12	28.70 ± 4.70	$\mathbf{45.44 \pm 9.68}$
Pubmed	37.69 ± 7.40	51.56 ± 8.81	39.80 ± 6.64	48.32 ± 8.36	$\underline{51.84 \pm 9.42}$	51.01 ± 6.54	49.20 ± 8.52	48.65 ± 6.27	$\mathbf{55.73 \pm 11.65}$
Physics	54.56 ± 10.27	$\underline{78.76 \pm 9.14}$	50.44 ± 10.98	58.03 ± 9.67	75.01 ± 11.41	68.14 ± 15.76	61.55 ± 13.04	69.99 ± 8.69	$\mathbf{83.84 \pm 7.66}$
Cornell	26.47 ± 9.91	31.36 ± 10.60	26.47 ± 7.49	35.25 ± 10.94	36.22 ± 9.52	29.36 ± 11.34	30.46 ± 7.77	$\underline{36.89 \pm 9.49}$	$\mathbf{44.52 \pm 12.37}$
Wisconsin	22.36 ± 9.18	33.61 ± 10.91	23.50 ± 9.89	$\underline{37.25 \pm 11.32}$	23.32 ± 8.61	31.01 ± 8.48	32.19 ± 7.27	36.17 ± 9.23	$\mathbf{52.81 \pm 15.06}$
Arxiv	4.15 ± 0.82	14.46 ± 3.47	7.72 ± 1.89	7.47 ± 1.43	12.64 ± 2.15	2.62 ± 1.01	$\underline{17.25 \pm 3.94}$	4.73 ± 0.85	$\mathbf{23.04 \pm 6.06}$
Products	11.21 ± 2.39	14.17 ± 4.34	13.07 ± 2.21	3.80 ± 0.78	10.97 ± 4.48	4.02 ± 1.37	$\underline{15.16 \pm 2.72}$	3.71 ± 0.86	$\mathbf{17.96 \pm 5.10}$
Graph-Level Transfer									
COLLAB	43.42 ± 11.74	42.64 ± 7.97	42.72 ± 8.08	38.06 ± 6.97	–	$\underline{50.27 \pm 1.19}$	46.46 ± 10.57	38.22 ± 5.76	$\mathbf{53.68 \pm 10.60}$
DD	51.14 ± 4.23	51.95 ± 5.68	51.69 ± 4.22	$\underline{59.61 \pm 13.10}$	–	52.08 ± 0.34	52.15 ± 5.20	51.81 ± 5.32	$\mathbf{74.84 \pm 11.98}$
ENZYMES	19.19 ± 2.93	20.07 ± 3.72	20.11 ± 3.52	19.75 ± 3.66	–	20.02 ± 0.54	$\underline{20.12 \pm 3.73}$	19.75 ± 3.44	$\mathbf{35.66 \pm 5.28}$
IMDB-B	53.44 ± 6.67	52.81 ± 6.82	53.09 ± 6.23	51.91 ± 6.50	–	$\underline{55.74 \pm 0.55}$	53.30 ± 6.56	53.56 ± 6.04	$\mathbf{64.50 \pm 11.89}$
PROTEINS	51.62 ± 5.96	53.37 ± 9.13	53.62 ± 8.83	$\underline{54.52 \pm 8.71}$	–	53.31 ± 0.31	53.41 ± 9.29	53.36 ± 9.41	$\mathbf{65.46 \pm 7.14}$

5 EXPERIMENTS

We evaluate SPR on node- and graph-level classification across diverse graph domains, with particular emphasis on whether the learned graph cognition remains reusable beyond the pre-training graphs. For node classification, SPR is pretrained on Cora, Photo, CS, and Chameleon, and evaluated on unseen graphs including Computers, Physics, Citeseer, Pubmed, Cornell, Wisconsin, ogbn-arxiv, and ogbn-products (Sen et al., 2008; Pei et al., 2020; Shchur et al., 2018; Hu et al., 2020). For graph classification, we evaluate on COLLAB, DD, ENZYMES, IMDB-B, and PROTEINS (Morris et al., 2020). We consider both a 1:1:8 train/validation/test split and the 1-shot setting, providing complementary evaluations of self-supervised representation quality and label-efficient transfer. We compare SPR with self-supervised graph learning methods, including GRACE, BGRL, and GraphMAE2 (Zhu et al., 2020; Thakoor et al., 2021; Hou et al., 2023), as well as graph foundation models, including GCOPE, GraphAny, SAMGPT, TIG, and GraphGlue (Zhao et al., 2024a;b; Yu et al., 2025; Zhao et al., 2026; Sun et al., 2026). Additional implementation and evaluation details are provided in Appendix D.

5.1 1-SHOT NODE AND GRAPH CLASSIFICATION

Table 1 reports the 1-shot results, where only one labeled example per class is available for downstream adaptation. This setting provides a stringent test of transferability because the limited supervision leaves little opportunity to relearn graph-specific patterns on the target domain.

For node classification, SPR achieves the best mean performance on all source and unseen graphs. Its average margin over the runner-up increases from 1.81 percentage points on source graphs to 6.52 on unseen graphs, with clear gains on Wisconsin, Cornell, and Arxiv. This is consistent with SPR's shared-yet-adaptive design: spectral responses are shared but instantiated through graph-specific operators, while the prototype geometry is aligned to each target graph, remaining effective under minimal supervision.

The graph-classification results provide a further test of whether the learned knowledge remains reusable beyond node-level prediction. SPR achieves the best performance on all five benchmarks, outperforming the runner-up by 10.78 percentage points on average. Since these datasets differ from the pre-training graphs and also require graph-level rather than node-level prediction, the consistent gains indicate that the learned representations are not tied to a particular source graph or prediction granularity. The results are also consistent with the role of relational geometry, which organizes pairwise relations in a shared prototype space that can be instantiated on new graphs.

Table 2: Self-supervised representation learning performance on node and graph classification. Results are accuracy (%, mean ± std); best and runner-up results are bolded and underlined, respectively.

	Self-supervised Pre-training			Graph Foundation Models					
	BGRL	GraphMAE2	GRACE	GCOPE	GraphAny	TIG	SAMGPT	GraphGlue	**SPR**
Source-Graph Node Classification									
Cora	81.22 ± 1.17	$\underline{82.38 \pm 1.10}$	77.28 ± 0.34	79.97 ± 0.50	78.83 ± 2.11	81.19 ± 1.26	81.79 ± 0.71	77.54 ± 0.64	$\mathbf{84.10 \pm 0.54}$
Photo	89.48 ± 0.93	90.24 ± 0.15	91.51 ± 0.23	80.65 ± 0.54	87.48 ± 3.44	$\underline{92.88 \pm 0.35}$	89.92 ± 0.65	87.57 ± 0.40	$\mathbf{93.60 \pm 0.19}$
CS	88.55 ± 0.97	91.82 ± 0.09	91.97 ± 0.08	86.13 ± 0.20	90.66 ± 0.50	$\underline{92.62 \pm 0.26}$	88.76 ± 0.20	86.14 ± 0.27	$\mathbf{93.45 \pm 0.26}$
Chameleon	26.95 ± 1.28	44.18 ± 2.90	35.54 ± 0.52	$\underline{53.10 \pm 0.83}$	$\mathbf{57.50 \pm 1.47}$	34.52 ± 2.80	30.19 ± 2.53	24.26 ± 3.48	46.18 ± 1.07
Held-out Graph Node Classification									
Computers	87.42 ± 2.24	79.81 ± 0.54	83.23 ± 0.03	75.04 ± 0.19	86.38 ± 0.36	$\underline{87.66 \pm 0.33}$	78.57 ± 0.16	83.91 ± 1.01	$\mathbf{88.85 \pm 0.46}$
Physics	85.97 ± 0.73	92.36 ± 0.08	90.35 ± 0.28	92.34 ± 0.11	85.55 ± 1.13	$\underline{95.36 \pm 0.13}$	90.95 ± 0.56	93.46 ± 0.34	$\mathbf{95.83 \pm 0.06}$
Citeseer	69.17 ± 1.41	71.39 ± 0.76	55.09 ± 1.71	$\underline{71.56 \pm 0.89}$	68.94 ± 1.51	70.18 ± 0.99	69.16 ± 0.37	62.80 ± 1.71	$\mathbf{72.81 \pm 0.73}$
Pubmed	77.85 ± 1.83	83.11 ± 0.59	74.25 ± 0.21	81.73 ± 0.22	84.56 ± 0.22	$\underline{85.34 \pm 0.28}$	79.07 ± 1.20	81.48 ± 0.31	$\mathbf{87.41 \pm 0.18}$
Cornell	47.03 ± 3.03	48.49 ± 2.05	44.60 ± 1.63	45.89 ± 14.54	$\underline{54.20 \pm 3.28}$	42.65 ± 7.85	49.54 ± 4.89	44.22 ± 1.67	$\mathbf{61.24 \pm 2.81}$
Wisconsin	46.87 ± 0.20	56.12 ± 1.59	43.28 ± 2.84	52.04 ± 11.10	$\underline{61.99 \pm 2.49}$	47.51 ± 6.69	49.47 ± 2.58	46.76 ± 2.26	$\mathbf{68.75 \pm 2.50}$
Arxiv	59.20 ± 0.45	57.19 ± 0.53	$\underline{66.80 \pm 0.62}$	65.24 ± 0.43	59.64 ± 0.52	16.14 ± 0.04	61.32 ± 0.77	64.14 ± 0.49	$\mathbf{70.21 \pm 0.71}$
Products	77.16 ± 0.87	$\underline{81.07 \pm 0.91}$	77.68 ± 0.64	58.58 ± 0.37	79.49 ± 0.68	27.31 ± 0.02	79.10 ± 0.82	80.88 ± 0.96	$\mathbf{81.89 \pm 1.14}$
Graph-Level Transfer									
COLLAB	$\underline{73.78 \pm 1.06}$	71.48 ± 1.13	72.29 ± 1.14	71.41 ± 1.10	–	72.34 ± 1.19	71.26 ± 1.25	70.74 ± 1.89	$\mathbf{93.73 \pm 0.61}$
DD	65.15 ± 2.55	66.38 ± 2.74	64.84 ± 1.33	$\underline{74.00 \pm 2.12}$	–	66.38 ± 1.71	65.32 ± 1.89	67.36 ± 1.35	$\mathbf{91.75 \pm 1.48}$
ENZYMES	24.17 ± 2.25	23.83 ± 1.43	$\underline{25.63 \pm 0.86}$	20.88 ± 2.35	–	22.13 ± 2.82	24.08 ± 1.63	23.25 ± 3.79	$\mathbf{49.29 \pm 4.62}$
IMDB-B	67.67 ± 3.98	67.10 ± 3.25	67.22 ± 2.80	66.02 ± 3.24	–	69.27 ± 1.83	$\underline{69.60 \pm 1.37}$	65.55 ± 2.13	$\mathbf{90.37 \pm 1.28}$
PROTEINS	63.95 ± 2.87	69.11 ± 0.74	65.99 ± 4.30	$\underline{72.53 \pm 0.61}$	–	66.44 ± 2.20	68.53 ± 1.13	65.90 ± 2.82	$\mathbf{89.16 \pm 1.76}$

5.2 Self-Supervised Representation Learning

Table 2 evaluates representations learned through self-supervised pre-training using a frozen encoder and linear probing. SPR is pretrained on four source graphs and evaluated on source and unseen node-classification graphs, together with graph-level benchmarks. Because the pretrained encoder is fixed during downstream evaluation, this setting directly measures the quality and reusability of the learned representations without relearning the encoder for each dataset.

On the source graphs, SPR obtains the best results on Cora, Photo, and CS, although it does not outperform GraphAny and GCOPE on Chameleon. The more important observation is its behavior beyond the pre-training domains: SPR achieves the best performance on all eight unseen graphs, including graphs that differ substantially in scale, feature space, and topology from the source graphs. The gains on Cornell and Wisconsin are particularly pronounced, while consistent improvements are also observed on much larger graphs such as Arxiv and Products. The fact that SPR does not uniformly dominate every source graph, yet performs consistently strongly on unseen graphs, suggests that its advantage is not simply stronger fitting to the pre-training domains. Instead, the learned representation remains effective as graph characteristics change, supporting flexible and transferable graph cognition under substantial domain variation.

For graph classification, SPR again achieves the best performance on all five benchmarks, exceeding the strongest competing result by 16.63–23.66 percentage points. This provides a complementary evaluation of representation reusability because the same pretrained model is evaluated under a different prediction granularity. Together with the unseen-node results, these findings show that the representations learned by SPR remain effective across different graph domains and downstream settings. They are also consistent with the proposed prototype geometry serving as a shared relational reference rather than being specialized to a particular source topology.

5.3 Ablation Study

Table 3 examines whether the two core components of SPR account for the transfer behavior observed above. Removing spectral pattern parsing consistently degrades performance, with particularly large drops of 11.45 points on Cornell, 19.00 points on COLLAB, and 9.08 points on ENZYMES, compared with only a modest decrease on the source graph Cora. This contrast indicates that the spectral component is especially important when the model encounters graphs or prediction settings beyond the source domains,

Table 3: Ablation of the key components of SPR.

Setting	Cora	Cornell	COLLAB	ENZYMES
SPR	84.10 ± 0.54	61.24 ± 2.81	93.73 ± 0.61	49.29 ± 4.62
w/o spectral	83.62 ± 0.12	49.79 ± 2.90	74.73 ± 1.70	40.21 ± 3.21
adj_only	80.31 ± 1.68	46.07 ± 4.75	83.18 ± 0.78	46.50 ± 4.17
heat_only	81.23 ± 0.54	43.04 ± 3.67	84.44 ± 0.36	49.08 ± 4.55
w/o proto	80.64 ± 1.45	46.48 ± 3.61	92.08 ± 0.27	47.33 ± 3.96
w/o adapt	81.18 ± 1.52	45.38 ± 3.25	92.77 ± 0.78	48.00 ± 5.27

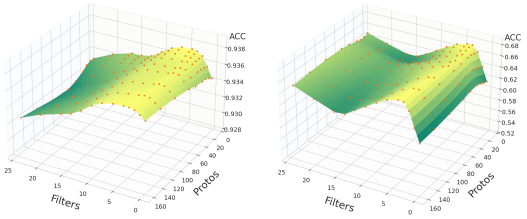
supporting its role in providing a flexible spectral representation that adapts to graph-specific spectral variation rather than merely improving source-graph fitting.

The relational component exhibits a complementary effect. Removing the prototype geometry causes a 14.76-point decrease on Cornell, while removing graph-specific adaptation to the prototype space leads to a 15.86-point drop. Both degradations are substantially larger than that observed on Cora. These results support two aspects of the proposed relational mechanism: the shared prototype geometry provides relational information learned across source graphs, and this geometry needs to be instantiated according to each graph rather than applied uniformly. Finally, both `adj_only` and `heat_only` underperform the full model, showing that local adjacency or graph-specific diffusion alone cannot recover the information introduced by spectral pattern parsing and the shared relational geometry. Overall, the ablations separately support the necessity of the flexible spectral mechanism and the cross-graph relational reference used in SPR.

5.4 Model Analysis

We further examine whether the learned spectral responses and prototype geometry behave as intended beyond their contribution to downstream accuracy.

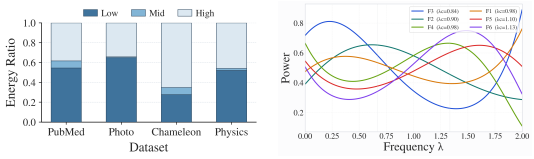
As shown in Figure 3, performance depends on both the number of spectral filters and the size of the prototype space. CS remains stable across most settings, whereas Wisconsin is more sensitive to the number of filters. Using too few filters degrades performance on both datasets, indicating that multiple spectral responses are needed to accommodate variation across spectral regions. In contrast, increasing the number of prototypes beyond a moderate size brings little additional benefit and can slightly reduce performance, suggesting that the recurring relational information can be summarized by a relatively compact prototype space.



(a) CS (b) Wisconsin

Figure 3: Effect of filters and prototypes on ACC.

Figure 4 further examines what is learned by spectral pattern parsing. The response functions exhibit distinct peaks and power-weighted centers, showing that different filters specialize in complementary spectral regions rather than collapsing to similar responses. More importantly, the resulting spectral representations exhibit different energy distributions across graphs even though the response functions are shared. This behavior directly reflects the design of SPR: the spectral mechanism is shared across domains, while its realization remains graph-dependent through the graph-specific operator and input features. Compared with the raw spectral distributions in Figure 1, these results show that SPR adaptively reorganizes spectral information rather than imposing a fixed smoothing pattern.



(a) Spectral distribution. (b) Spectral filter responses.

Figure 4: Analysis of the learned spectral responses in SPR.

6 Conclusion

This paper investigates transferable graph cognition from a spectral perspective. SPR learns shared Chebyshev spectral responses to flexibly characterize continuously varying graph patterns, while cross-graph relational geometry distills common pairwise organization from multiple source graphs into a shared prototype geometry. The shared spectral mechanism is instantiated on graph-specific operators, and the learned relational reference can be aligned to unseen graphs, together supporting graph cognition that is adaptive and transferable. Experiments across source and unseen graphs, node- and graph-level tasks, and few-shot settings support the effectiveness of the proposed design.

AI USE STATEMENT

Generative AI tools were used in a limited capacity for language polishing, minor rephrasing, improving manuscript organization and readability. Generative AI was not used to generate data or experimental results. All AI-assisted content was reviewed and verified by the authors, and all equations, citations, numerical results, technical descriptions, and scientific claims were independently checked by the authors. We take full responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

REFERENCES

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6203.

Haibo Chen, Xin Wang, Zeyang Zhang, Haoyang Li, Ling Feng, and Wenwu Zhu. Autogfm: Automated graph foundation model with adaptive architecture customization. In *Forty-second International Conference on Machine Learning*, 2025.

Jialin Chen, Haolan Zuo, Haoyu Wang, Siqi Miao, Pan Li, and Rex Ying. Towards A universal graph structural encoder. In Hakim Hacid, Yoelle Maarek, Francesco Bonchi, Ido Guy, and Emine Yilmaz, editors, *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pages 1457–1468. ACM, 2026. doi: 10.1145/3774904.3792656. URL https://doi.org/10.1145/3774904.3792656.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Yushun Dong, Patrick Soga, Yinhan He, Song Wang, and Jundong Li. Graph neural networks are more than filters: Revisiting and benchmarking from A spectral perspective. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=nWdQX5hOL9.

Lele Fu, Tianchi Liao, Sheng Huang, Bowen Deng, Chuanfu Zhang, Shirui Pan, and Chuan Chen. Unsupervised federated graph learning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/8c4c80d3eee8a56da1302bc6dcd2d953-Abstract-Conference.html.

Xiaodong He, Haolan He, Ruiyi Fang, Ming Sun, and Zhao Kang. Structure-centric graph foundation model via geometric bases. *CoRR*, abs/2605.08689, 2026. doi: 10.48550/ARXIV.2605.08689. URL https://doi.org/10.48550/arXiv.2605.08689.

Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 448–459, 2025.

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*, pages 737–746, 2023.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Jinwoo Kim, Xingyue Huang, Krzysztof Olejniczak, Kyungbin Min, Michael M. Bronstein, Seunghoon Hong, and .Ismail .Ilkan Ceylan. Flock: A knowledge graph foundation model via learning on random walks. *CoRR*, abs/2510.01510, 2025. doi: 10.48550/ARXIV.2510.01510. URL https://doi.org/10.48550/arXiv.2510.01510.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

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.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Nian Liu, Xiaoxin He, Thomas Laurent, Francesco Di Giovanni, Michael M. Bronstein, and Xavier Bresson. A general graph spectral wavelet convolution via chebyshev order decomposition. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/liu25y.html.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben, editors, *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 417–428. ACM, 2023b. doi: 10.1145/3543507.3583386. URL https://doi.org/10.1145/3543507.3583386.

Facundo Mémoli. Gromov-wasserstein distances and the metric approach to object matching. *Found. Comput. Math.*, 11(4):417–487, 2011. doi: 10.1007/S10208-011-9093-5. URL https://doi.org/10.1007/s10208-011-9093-5.

Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):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. *CoRR*, abs/2007.08663, 2020. URL https://arxiv.org/abs/2007.08663.

Antonio Ortega, Pascal Frossard, Jelena Kovacevic, José M. F. Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE*, 106(5):808–828, 2018. doi: 10.1109/JPROC.2018.2820126. URL https://doi.org/10.1109/JPROC.2018.2820126.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=S1e2agrFvS.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710. ACM, 2014. doi: 10.1145/2623330.2623732. URL https://doi.org/10.1145/2623330.2623732.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pages 2664–2672. PMLR, 2016.

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

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *CoRR*, abs/1811.05868, 2018. URL http://arxiv.org/abs/1811.05868.

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*, pages 1154–1165, 2025a.

Li Sun, Zhenhao Huang, Suyang Zhou, Qiqi Wan, Hao Peng, and Philip S. Yu. Riemanngfm: Learning a graph foundation model from riemannian geometry. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov, editors, *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pages 1154–1165. ACM, 2025b. doi: 10.1145/3696410.3714952. URL https://doi.org/10.1145/3696410.3714952.

Li Sun, Zhenhao Huang, Silei Chen, Lanxu Yang, Junda Ye, Sen Su, and Philip S. Yu. Multi-domain riemannian graph gluing for building graph foundation models. *CoRR*, abs/2603.00618, 2026. doi: 10.48550/ARXIV.2603.00618. URL https://doi.org/10.48550/arXiv.2603.00618.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 workshop on geometrical and topological representation learning*, pages 1–14. OpenReview. net, 2021.

Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pages 6275–6284. PMLR, 2019.

Yi Wang, Jitao Zhao, Di Jin, and Dongxiao He. Beyond feature and structure alignment: Learning transferable propagation knowledge for graph foundation models. *CoRR*, abs/2607.28980, 2026. doi: 10.48550/ARXIV.2607.28980. URL https://doi.org/10.48550/arXiv.2607.28980.

Yuyao Wang, Yu-Hung Cheng, Debarghya Mukherjee, and Huimin Cheng. Transfer learning on edge connecting probability estimation under graphon model. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025a. URL http://papers.nips.cc/paper_files/paper/2025/hash/30dee0ddc6c0b4907a9d587d3ce87979-Abstract-Conference.html.

Zehong Wang, Zheyuan Zhang, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. Gft: Graph foundation model with transferable tree vocabulary. *Advances in neural information processing systems*, 37:107403–107443, 2024.

Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, et al. Graph foundation models: A comprehensive survey. *arXiv preprint arXiv:2505.15116*, 2025b.

Zehong Wang, Zheyuan Zhang, Tianyi Ma, Nitesh V. Chawla, Chuxu Zhang, and Yanfang Ye. Towards graph foundation models: Learning generalities across graphs via task-trees. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025c. URL https://proceedings.mlr.press/v267/wang25eq.html.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32 (1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 974–983. ACM, 2018. doi: 10.1145/3219819.3219890. URL https://doi.org/10.1145/3219819.3219890.

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*, pages 1142–1153, 2025.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Zhou Qin, Weigao Wen, Hui Xue, Haoyang Li, and Wenwu Zhu. Spectral invariant learning for dynamic graphs under distribution shifts. *Advances in Neural Information Processing Systems*, 36:6619–6633, 2023.

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*, pages 4443–4454, 2024a.

Jianan Zhao, Zhaocheng Zhu, Mikhail Galkin, Hesham Mostafa, Michael Bronstein, and Jian Tang. Fully-inductive node classification on arbitrary graphs. *arXiv preprint arXiv:2405.20445*, 2024b.

Jitao Zhao, Yi Wang, Yawen Li, Dongxiao He, Di Jin, Zhiyong Feng, and Weixiong Zhang. Towards graph foundation model: Node feature transfer invariant modeling on general graphs. In Hakim Hacid, Yoelle Maarek, Francesco Bonchi, Ido Guy, and Emine Yilmaz, editors, *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pages 810–821. ACM, 2026. doi: 10.1145/3774904.3792236. URL https://doi.org/10.1145/3774904.3792236.

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

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

A THEORY OF PROTOTYPE TRANSFER

This section gives detailed proofs for two properties of prototype transfer. The first connects GW alignment to the fidelity of lifting the shared geometry back to a graph. The second studies the stability of the unnormalized prototype-induced operator $P_T = TK_\star T^\top$ with respect to transport estimation error. These results characterize complementary parts of the transfer mechanism rather than claiming an end-to-end guarantee for the normalized encoder operator.

For Proposition 1, let $T_m \in \Pi(w_m, b)$, so that

$$T_m \mathbf{1} = w_m, \qquad T_m^\top \mathbf{1} = b, \qquad T_m(i,k) \geq 0.$$

We assume $w_m(i) > 0$ on the node support considered below. Nodes with zero mass can be omitted because they contribute zero to the weighted error. We use

$$\|B\|^2_{w_m \otimes w_m} = \sum_{i,i'} w_m(i) w_m(i') B(i,i')^2$$

for the weighted squared Frobenius norm. For Proposition 2, $\|\cdot\|_2$ denotes the spectral norm and $\|\cdot\|_F$ the Frobenius norm.

A.1 PROOF OF DISTORTION-CONTROLLED GEOMETRY TRANSFER

We prove that the same pairwise distortion minimized by the GW objective upper-bounds the error obtained after lifting the shared prototype geometry back to the graph.

Step 1: Normalize the transport into conditional prototype assignments. Define

$$R_m = \mathrm{diag}(w_m)^{-1} T_m. \tag{17}$$

For any node i with $w_m(i) > 0$,

$$R_m(i,k) = \frac{T_m(i,k)}{w_m(i)} \geq 0. \tag{18}$$

Moreover, because $T_m \mathbf{1} = w_m$,

$$\begin{aligned} \sum_k R_m(i,k) &= \frac{1}{w_m(i)} \sum_k T_m(i,k) \\ &= \frac{w_m(i)}{w_m(i)} = 1. \end{aligned} \tag{19}$$

Therefore, each row $R_m(i,\cdot)$ is a probability distribution over the p prototypes. It can be interpreted as the conditional prototype assignment of node i.

Step 2: Express the lifted geometry as an expectation over prototype pairs. The lifted graph-space geometry is

$$\overline{C}_m = R_m C_\star R_m^\top. \tag{20}$$

Its (i,i')-th entry is

$$\begin{aligned} \overline{C}_m(i,i') &= e_i^\top R_m C_\star R_m^\top e_{i'} \\ &= \sum_{k,l} R_m(i,k)\, C_\star(k,l)\, R_m(i',l). \end{aligned} \tag{21}$$

For this fixed node pair, define the product distribution

$$\pi_{ii'}(k,l) = R_m(i,k) R_m(i',l). \tag{22}$$

Because both rows of R_m sum to one,

$$\sum_{k,l} \pi_{ii'}(k,l) = \left(\sum_k R_m(i,k)\right)\left(\sum_l R_m(i',l)\right) = 1. \tag{23}$$

Hence, if $K \sim R_m(i,\cdot)$ and $L \sim R_m(i',\cdot)$ are sampled independently, then

$$\overline{C}_m(i,i') = \mathbb{E}[C_\star(K,L) \mid i, i']\,. \tag{24}$$

Step 3: Bound the reconstruction error for one node pair. Since $C_m(i,i')$ is fixed once (i,i') is fixed, Eq. (24) gives

$$\begin{aligned} C_m(i,i') - \overline{C}_m(i,i') &= C_m(i,i') - \mathbb{E}[C_\star(K,L) \mid i,i'] \\ &= \mathbb{E}[C_m(i,i') - C_\star(K,L) \mid i,i']\,. \end{aligned} \tag{25}$$

Applying Jensen's inequality to the convex function $x \mapsto x^2$,

$$\begin{aligned} \big(C_m(i,i') - \overline{C}_m(i,i')\big)^2 &= \big(\mathbb{E}[C_m(i,i') - C_\star(K,L) \mid i,i']\big)^2 \\ &\le \mathbb{E}\Big[\big(C_m(i,i') - C_\star(K,L)\big)^2 \mid i,i'\Big]\,. \end{aligned} \tag{26}$$

Expanding this expectation under the product assignment in Eq. (22),

$$\big(C_m(i,i') - \overline{C}_m(i,i')\big)^2 \le \sum_{k,l} \big(C_m(i,i') - C_\star(k,l)\big)^2 R_m(i,k) R_m(i',l). \tag{27}$$

Thus, for each node pair, the squared error of the averaged lifted geometry is no larger than the expected squared relational mismatch before averaging.

Step 4: Aggregate over node pairs and recover the GW distortion. Multiply Eq. (27) by $w_m(i)w_m(i')$ and sum over all node pairs:

$$\begin{aligned} \|C_m - \overline{C}_m\|^2_{w_m \otimes w_m} &= \sum_{i,i'} w_m(i) w_m(i') \big(C_m(i,i') - \overline{C}_m(i,i')\big)^2 \\ &\le \sum_{i,i',k,l} w_m(i) w_m(i') \big(C_m(i,i') - C_\star(k,l)\big)^2 R_m(i,k) R_m(i',l). \end{aligned} \tag{28}$$

From the definition of R_m,

$$w_m(i) R_m(i,k) = T_m(i,k), \qquad w_m(i') R_m(i',l) = T_m(i',l). \tag{29}$$

Substituting Eq. (29) into Eq. (28),

$$\begin{aligned} \|C_m - \overline{C}_m\|^2_{w_m \otimes w_m} &\le \sum_{i,i',k,l} \big(C_m(i,i') - C_\star(k,l)\big)^2 T_m(i,k) T_m(i',l) \\ &= \mathcal{D}_{\mathrm{GW}}(C_m, C_\star, T_m). \end{aligned} \tag{30}$$

The last line is the GW relational distortion at T_m, which proves Proposition 1.

Proposition 2 (Transport-stable prototype propagation)**.** *Let $P_T = T K_\star T^\top$ and $P_{\widetilde{T}} = \widetilde{T} K_\star \widetilde{T}^\top$. Then*

$$\|P_T - P_{\widetilde{T}}\|_F \le \|K_\star\|_2 (\|T\|_2 + \|\widetilde{T}\|_2) \|T - \widetilde{T}\|_F. \tag{31}$$

If T and $\widetilde{T}$ are probability couplings, the bound reduces to $2\|K_\star\|_2 \|T - \widetilde{T}\|_F$.

The result applies to the unnormalized operator before the symmetric normalization used by the encoder.

A.2 Proof of Transport-Stable Prototype Propagation

We next bound the change in the induced node-space operator under perturbations of the graph-to-prototype transport.

Let

$$P_T = T K_\star T^\top, \qquad P_{\widetilde{T}} = \widetilde{T} K_\star \widetilde{T}^\top, \tag{32}$$

and define the transport perturbation

$$\Delta_T = T - \widetilde{T}. \tag{33}$$

$$P_T - P_{\widetilde{T}} = T K_\star T^\top - \widetilde{T} K_\star \widetilde{T}^\top. \tag{34}$$

Add and subtract $\widetilde{T}K_\star T^\top$:

$$\begin{aligned} P_T - P_{\widetilde{T}} &= TK_\star T^\top - \widetilde{T}K_\star T^\top + \widetilde{T}K_\star T^\top - \widetilde{T}K_\star \widetilde{T}^\top \\ &= (T-\widetilde{T})K_\star T^\top + \widetilde{T}K_\star (T-\widetilde{T})^\top \\ &= \Delta_T K_\star T^\top + \widetilde{T}K_\star \Delta_T^\top. \end{aligned} \tag{35}$$

We use the standard inequalities

$$\|AB\|_F \le \|A\|_F \|B\|_2, \qquad \|AB\|_F \le \|A\|_2 \|B\|_F.$$

For the first term in Eq. (35),

$$\begin{aligned} \|\Delta_T K_\star T^\top\|_F &\le \|\Delta_T\|_F \|K_\star T^\top\|_2 \\ &\le \|\Delta_T\|_F \|K_\star\|_2 \|T^\top\|_2 \\ &= \|K_\star\|_2 \|T\|_2 \|\Delta_T\|_F. \end{aligned} \tag{36}$$

Similarly,

$$\begin{aligned} \|\widetilde{T}K_\star \Delta_T^\top\|_F &\le \|\widetilde{T}K_\star\|_2 \|\Delta_T^\top\|_F \\ &\le \|\widetilde{T}\|_2 \|K_\star\|_2 \|\Delta_T\|_F, \end{aligned} \tag{37}$$

where we used $\|\Delta_T^\top\|_F = \|\Delta_T\|_F$.

Using the triangle inequality and Eqs. (36)–(37),

$$\begin{aligned} \|P_T - P_{\widetilde{T}}\|_F &\le \|\Delta_T K_\star T^\top\|_F + \|\widetilde{T}K_\star \Delta_T^\top\|_F \\ &\le \|K_\star\|_2 \|T\|_2 \|\Delta_T\|_F + \|\widetilde{T}\|_2 \|K_\star\|_2 \|\Delta_T\|_F \\ &= \|K_\star\|_2 \big(\|T\|_2 + \|\widetilde{T}\|_2\big) \|T - \widetilde{T}\|_F. \end{aligned} \tag{38}$$

Suppose T is a probability coupling, i.e., $T(i,k) \ge 0$ and

$$\sum_{i,k} T(i,k) = 1. \tag{39}$$

Then

$$\begin{aligned} \|T\|_2 &\le \|T\|_F \\ &= \left(\sum_{i,k} T(i,k)^2 \right)^{1/2} \\ &\le \sum_{i,k} T(i,k) \\ &= 1. \end{aligned} \tag{40}$$

The same argument gives $\|\widetilde{T}\|_2 \le 1$. Substituting these two inequalities into Eq. (38),

$$\boxed{\|P_T - P_{\widetilde{T}}\|_F \le 2\|K_\star\|_2 \|T - \widetilde{T}\|_F}. \tag{41}$$

This proves the second claim.

Consequence for the propagated representation. For a fixed node representation H, the operator perturbation produces

$$\|(P_T - P_{\widetilde{T}})H\|_F \le \|P_T - P_{\widetilde{T}}\|_F \|H\|_2. \tag{42}$$

Combining this with Eq. (38) gives

$$\|(P_T - P_{\widetilde{T}})H\|_F \le \|K_\star\|_2 \big(\|T\|_2 + \|\widetilde{T}\|_2\big) \|T - \widetilde{T}\|_F \|H\|_2. \tag{43}$$

Thus, transport perturbations also bound their effect on a fixed propagated representation.

Table 4: Notation used throughout the method and appendix.

Symbol	Shape / Type	Description
G_m	graph	The m-th graph domain.
V_m, E_m	node / edge set	Node set and edge set of G_m.
n_m, e_m	scalar	Number of nodes and edges in G_m.
A_m	$n_m \times n_m$	Adjacency matrix of G_m.
D_m	$n_m \times n_m$	Degree matrix of A_m.
L_m	$n_m \times n_m$	Normalized graph Laplacian.
S_m	$n_m \times n_m$	Normalized propagation operator used by the spectral frontend.
ρ_m	n_m	Propagation-domain eigenvalues of S_m; $\rho = 1 - \lambda$ for Laplacian frequency λ.
X_m	$n_m \times d_m$	Raw node feature matrix, where d_m may vary across domains.
M_f	scalar	Number of learnable Chebyshev spectral filters.
Z_α	$n_m \times d_m$	Filtered response produced by the α-th Chebyshev spectral filter.
K_c	scalar	Chebyshev polynomial order.
r	scalar	Truncated SVD rank for each spectral filter response.
d	scalar	Unified spectral feature dimension after parsing.
h	scalar	Hidden dimension of the propagation backbone.
d_z	scalar	Output embedding dimension.
C_m	$n_m \times n_m$	Heat-kernel-distance structural cost matrix for graph G_m.
p	scalar	Number of structural prototypes.
$C_\star$	$p \times p$	Shared GW barycenter cost matrix in prototype space.
$A_\star$	$p \times p$	Prototype affinity matrix derived from $C_\star$.
$K_\star$	$p \times p$	Prototype heat kernel derived from $C_\star$.
T_m	$n_m \times p$	Node-to-prototype transport plan for graph G_m.
w_m	n_m	Node mass distribution of graph G_m.
b	p	Prototype mass distribution.
λ_m	scalar	Weight of source graph G_m in the GW barycenter.
P_m	$n_m \times n_m$	Prototype-induced propagation operator on graph G_m.
$K_{t_\ell}^{(m)}$	$n_m \times n_m$	Heat-diffusion operator at layer ℓ.
$H^{(\ell)}$	$n_m \times h$	Node representation after layer ℓ.
$H_A^{(\ell)}$	$n_m \times h$	Local adjacency-path representation.
$H_K^{(\ell)}$	$n_m \times h$	Heat-diffusion-path representation.
$H_P^{(\ell)}$	$n_m \times h$	Prototype-path representation.
$g_i^{(\ell)}$	$\mathbb{R}^3$	Node-wise gate over the three propagation paths.
$\mathcal{M}$	set of nodes	Masked node set for spectral-feature reconstruction.

B NOTATIONS USED IN OUR METHOD

Table 4 summarizes the main notation used by SPR. Unless otherwise specified, all graphs are treated as undirected during propagation, and sparse matrices are used for adjacency-based operations.

C ALGORITHM

Algorithm 1 Spectral Prototype-Guided Pre-Training

Input: Source graphs $\{G_m = (V_m, E_m, X_m)\}_{m=1}^{M}$, domain weights $\{\lambda_m\}_{m=1}^{M}$;
  prototype number p, filter number M_f, Chebyshev order K_c, SVD rank r;
  layers L, heat times $\{t_\ell\}_{\ell=1}^{L}$.
Output: Pretrained encoder f_θ, shared prototype kernel $K_\star$.
(1) Shared prototype geometry
for each source graph G_m **do**
  $L_m \leftarrow I - D_m^{-\frac{1}{2}} A_m D_m^{-\frac{1}{2}}$
  $K_{t_{\mathrm{gw}}}^{(m)} \leftarrow \exp(-t_{\mathrm{gw}} L_m)$
  for $\ell = 1$ to L **do**
    $K_{t_\ell}^{(m)} \leftarrow \exp(-t_\ell L_m)$
  end for
  $C_m \leftarrow \mathrm{HeatKernelCost}(K_{t_{\mathrm{gw}}}^{(m)})$
end for
$C_\star, \{T_m\}_{m=1}^{M} \leftarrow \mathrm{GWBarycenter}(\{C_m\}_{m=1}^{M}, \{\lambda_m\}_{m=1}^{M}, p)$
$A_\star \leftarrow \mathrm{Affinity}(C_\star)$
$K_\star \leftarrow \exp(-L(A_\star))$
while not converged **do**
  for each source graph G_m **do**
    (2) Spectral frontend
    $S_m \leftarrow D_m^{-\frac{1}{2}} A_m D_m^{-\frac{1}{2}}$
    for $\alpha = 1$ to M_f **do**
      $Z_\alpha \leftarrow \sum_{k=0}^{K_c} \theta_{\alpha,k} T_k(S_m, X_m)$
      $\widetilde{Z}_\alpha \leftarrow \mathrm{SVD}_r^{\mathrm{node}}(Z_\alpha)$
    end for
    $X_{\mathrm{spec},m} \leftarrow \mathrm{Proj}([\widetilde{Z}_1 \| \cdots \| \widetilde{Z}_{M_f}])$
    (3) Prototype-guided graph propagation
    $P_m \leftarrow \mathrm{SymNorm}(T_m K_\star T_m^\top)$
    sample masked node set $\mathcal{M}$ and set $\widehat{X}_{\mathrm{spec},m} \leftarrow \mathrm{Mask}(X_{\mathrm{spec},m}, \mathcal{M})$
    $H_m^{(0)} \leftarrow \mathrm{InputProj}(\widehat{X}_{\mathrm{spec},m})$
    $H_{A,m}^{(0)}, H_{K,m}^{(0)}, H_{P,m}^{(0)} \leftarrow H_m^{(0)}, H_m^{(0)}, H_m^{(0)}$
    for $\ell = 1$ to L **do**
      $H_{A,m}^{(\ell)} \leftarrow \mathrm{GCN}_A^{(\ell)}(A_m, H_{A,m}^{(\ell-1)})$
      $H_{K,m}^{(\ell)} \leftarrow \mathrm{GCN}_K^{(\ell)}(K_{t_\ell}^{(m)}, H_{K,m}^{(\ell-1)})$
      $H_{P,m}^{(\ell)} \leftarrow \mathrm{GCN}_P^{(\ell)}(P_m, H_{P,m}^{(\ell-1)})$
      $[g_{A,m}^{(\ell)}, g_{K,m}^{(\ell)}, g_{P,m}^{(\ell)}] \leftarrow \mathrm{Softmax}(\mathrm{Gate}([H_{A,m}^{(\ell)} \| H_{K,m}^{(\ell)} \| H_{P,m}^{(\ell)}]))$
      $H_m^{(\ell)} \leftarrow g_{A,m}^{(\ell)} \odot H_{A,m}^{(\ell)} + g_{K,m}^{(\ell)} \odot H_{K,m}^{(\ell)} + g_{P,m}^{(\ell)} \odot H_{P,m}^{(\ell)}$
      $H_{A,m}^{(\ell)}, H_{K,m}^{(\ell)}, H_{P,m}^{(\ell)} \leftarrow H_m^{(\ell)}, H_m^{(\ell)}, H_m^{(\ell)}$
    end for
    $Z_m \leftarrow H_m^{(L)}$
    (4) Self-supervised losses
    sample a second edge-perturbed view and obtain Z'_m using the same spectral frontend and encoder
    $\mathcal{L}_{\mathrm{node}} \leftarrow \mathrm{InfoNCE}(Z_m, Z'_m)$
    $\mathcal{L}_{\mathrm{spec}} \leftarrow |\mathcal{M}|^{-1} \sum_{i \in \mathcal{M}} \| f_{\mathrm{rec}}(Z_{m,i}) - X_{\mathrm{spec},m,i} \|_2^2$
    $\mathcal{L}_m \leftarrow 1.5\mathcal{L}_{\mathrm{node}} + 0.3\mathcal{L}_{\mathrm{spec}}$
    Update the trainable encoder and spectral parameters by minimizing $\mathcal{L}_m$.
  end for
end while
return f_θ and $K_\star$

Algorithm 1 summarizes SPR pre-training. It first learns a shared GW prototype geometry and converts it into $K_\star$. Each source graph is then encoded by spectral parsing, followed by gated propagation over adjacency, heat diffusion, and the symmetrically normalized prototype-induced operator derived from $T_m K_\star T_m^\top$. The encoder is trained with node-level contrastive learning and spectral feature reconstruction.

D Implementation Details

Table 5: Main hyper-parameters.

Category	Hyper-parameter	Value
Pre-training domains	Source graphs	Cora, CS, Photo, Chameleon
Backbone	epoch	25
Backbone	Input / hidden / output dimensions	$128/1024/768$
Backbone	GCN layers	3
Backbone	Dropout	0.12
Backbone	Heat times	$[0.5, 1.0, 1.5]$
Spectral frontend	Number of Chebyshev filters M_f	6
Spectral frontend	Chebyshev order K_c	3
Spectral frontend	SVD rank r	32
Spectral frontend	projected dimensions for input	128
Prototype module	Number of prototypes p	96
Prototype module	GW diffusion time t_{gw}	0.58
Prototype module	GW entropy regularization ϵ_{gw}	0.005
Optimization	Optimizer	Adam
Optimization	Learning rate	1.2×10^{-4}
Optimization	Weight decay	10^{-5}
Loss	Loss weights	$1.5/0.3$ for node/feature
Downstream adaptation	K-means assignment temperature	0.2
Downstream adaptation	sparse prototype assignment	top-8
Downstream adaptation	Linear probing epochs	10000
Downstream adaptation	Linear probing learning rate	0.015

Baselines. The baselines cover self-supervised graph representation learning and graph foundation or transfer learning methods, including GraphMAE2, BGRL, GRACE, GCOPE, GraphAny, TIG, SAMGPT, and GraphGlue. We use released implementations and recommended settings for baseline-specific pre-training or adaptation, while evaluating all supported methods with the same downstream splits, shot construction, and accuracy metric described below. Datasets unsupported by a baseline are marked as missing.

Data splits and evaluation protocol. For node classification, we use stratified 1:1:8 train/validation/test splits and report mean and standard deviation over 10 seeds. For graph classification, we likewise use stratified 1:1:8 splits over graphs and report mean and standard deviation over 10 seeds. Node inputs are unified to 128 dimensions: COLLAB and IMDB-BINARY use degree-based one-hot features, while DD, ENZYMES, and PROTEINS use one-hot categorical node labels. Linear probing freezes the pretrained encoder; graph embeddings are obtained by mean pooling, $h_G = |V_G|^{-1} \sum_{i \in V_G} z_i$. For 1-shot node and graph classification, we sample one labeled example per class, keep the encoder frozen, and average results over 100 independently sampled support sets.

Running environment. All experiments are conducted on a Linux server with 125 GiB RAM and 8 NVIDIA GeForce RTX 5090 GPUs, each with 32 GB memory. We use Python 3.9.23, PyTorch 2.8.0 with CUDA 12.8, and DGL 1.1.3.

Pre-training details. The model is pre-trained for 25 epochs. The encoder uses input, hidden, and output dimensions $128/1024/768$, three propagation layers, dropout 0.12, heat times $[0.5, 1.0, 1.5]$, gate temperature 1.0, and path-stream mix 1.0. The spectral frontend applies $M_f = 6$ Chebyshev filters of order $K_c = 3$ to the normalized operator $S = D^{-1/2}AD^{-1/2}$ without self-loops. Each graph uses $r_G = \min(32, n_G, d_{\text{raw}})$ node-side truncated-SVD components per filter, zero-padded to 32 if needed, following the node-side SVD convention defined in the Method. With 6 filters, this gives a fixed 192-dimensional representation projected to 128 dimensions. Thus all graphs still produce the same $M_f r$-dimensional spectral representation before the final projection. The prototype module uses $p = 96$ prototypes, diffusion time $t_{\text{gw}} = 0.58$, and GW entropy regularization $\epsilon_{\text{gw}} = 0.005$. The encoder is trained using the self-supervised objective described in Sec. 4.3, with loss weights $1.5/0.3$ for node-level contrastive learning and spectral feature reconstruction, respectively. We optimize the model with Adam using learning rate 1.2×10^{-4} and weight decay 10^{-5}. Node contrast uses two edge-perturbed views and negative node pairs, masked reconstruction hides 25% of nodes and reconstructs X_{spec}.

Spectral energy visualization. All spectral visualizations are computed from graph-signal energy rather than classifier outputs. For graphs where eigendecomposition is feasible, we compute the normalized Laplacian $L = I - D^{-1/2}AD^{-1/2} = U\Lambda U^\top$. Given a row-normalized representation X, its normalized Fourier energy on eigenvector u_i is

$$E_i(X) = \frac{\|u_i^\top X\|_2^2}{\sum_j \|u_j^\top X\|_2^2}. \tag{44}$$

We sort eigenvalues increasingly and sum the first, middle, and last thirds of the energies as low-, mid-, and high-frequency ratios. The "spectral distribution after parsing" figure uses $X = X_{\text{spec}}$, showing how Chebyshev filtering and SVD compression redistribute graph-signal energy. For large graphs, we use a residual-shell proxy: with $P = D_{\tilde{A}}^{-1/2}\tilde{A}D_{\tilde{A}}^{-1/2}$ and $\tilde{A} = A + I$, we set $R_k = P^k X$, compute $s_k = \|R_k - R_{k+1}\|_F^2$ for $k = 0, \ldots, 7$ and $s_8 = \|R_8\|_F^2$. After normalization by $\sum_{k=0}^{8} s_k$, we use $s_0 + s_1$, $\sum_{k=2}^{7} s_k$, and s_8 as high-, mid-, and low-frequency proxies.

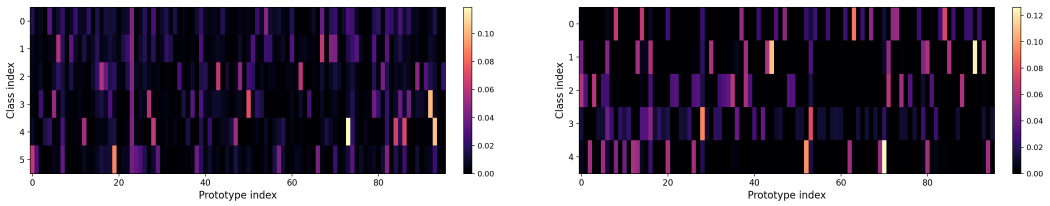
Prototype assignment visualization. We visualize class-prototype associations to examine how different classes use the learned global prototype space. For each target dataset, we first compute a soft node-to-prototype transport matrix $T \in \mathbb{R}^{N \times P}$ using the same KMeans-based transport procedure as in adaptation, where N is the number of nodes and P is the number of global prototypes. Each entry $T_{i,p}$ measures the assignment strength between node i and prototype p. We then row-normalize T to obtain per-node prototype assignment probabilities A, so that each node has a probability distribution over prototypes. For each class c, we average the assignment probabilities over all nodes belonging to that class:

$$M_{c,p} = \frac{1}{|\mathcal{V}_c|} \sum_{i \in \mathcal{V}_c} A_{i,p}, \qquad \mathcal{V}_c = \{i : y_i = c\}. \tag{45}$$

The resulting matrix $M \in \mathbb{R}^{C \times P}$ is visualized as a heatmap, where rows denote classes and columns denote global prototypes. Brighter entries indicate stronger association between a class and a prototype.

Figure 5 shows that different graphs and classes emphasize different subsets of the shared prototype space. We use this only as a qualitative view of prototype usage.

Learned filter-response visualization. The learned filter-response figure is computed directly from the trained Chebyshev coefficients, rather than from downstream predictions. Since the spectral frontend filters the normalized propagation operator $S = D^{-1/2}AD^{-1/2}$, and $S = I - L$ for the normalized Laplacian L, a Laplacian frequency $\lambda \in [0, 2]$ corresponds to the propagation-domain value $1 - \lambda$. Therefore, when the same learned filter is visualized on the Laplacian-frequency axis,



(a) Citeseer (b) Cornell

Figure 5: Prototype assignment on Citeseer and Cornell.

we write

$$\widehat{g}_\alpha(\lambda) = \sum_{k=0}^{K_c} \theta_{\alpha,k} T_k(1-\lambda), \tag{46}$$

which is exactly the Method response $g_\alpha(\rho)$ evaluated at $\rho = 1 - \lambda$. We evaluate $\widehat{g}_\alpha(\lambda)$ on a dense grid over $[0, 2]$ and plot the normalized power $|\widehat{g}_\alpha(\lambda)|^2$. To summarize each filter's dominant frequency preference, we also compute its power-weighted center. This visualization is used to verify whether the learned filters collapse to similar low-pass responses or instead cover complementary spectral regions.

E DATASETS

Table 6: Dataset statistics. For graph classification datasets, Nodes and Edges are averages per graph.

Dataset	Family	Graphs	Nodes	Edges	Node input	Classes
Node Classification Datasets						
Cora	Citation	1	2,708	10,556	1,433	7
Citeseer	Citation	1	3,327	9,104	3,703	6
Pubmed	Citation	1	19,717	88,648	500	3
Photo	Amazon co-purchase	1	7,650	238,162	745	8
Computers	Amazon co-purchase	1	13,752	491,722	767	10
CS	Coauthor	1	18,333	163,788	6,805	15
Physics	Coauthor	1	34,493	495,924	8,415	5
Cornell	WebKB	1	183	298	1,703	5
Wisconsin	WebKB	1	251	515	1,703	5
Chameleon	WikipediaNetwork	1	2,277	36,101	2,325	5
ogbn-arxiv	OGB citation	1	169,343	1,166,243	128	40
ogbn-products	OGB co-purchase	1	2,449,029	61,859,140	100	47
Graph Classification Datasets						
COLLAB	Social	5,000	74.49	2,457.78	Degree	3
DD	Bioinformatics	1,178	284.32	715.66	Label	2
ENZYMES	Bioinformatics	600	32.63	62.14	Label	6
IMDB-B	Social	1,000	19.77	96.53	Degree	2
PROTEINS	Bioinformatics	1,113	39.06	72.82	Label	2

For graph classification, "Degree" denotes degree-based one-hot node inputs used for the featureless COLLAB and IMDB-B datasets, while "Label" denotes one-hot encoded categorical node labels used as node features for DD, ENZYMES, and PROTEINS. All resulting node inputs are unified to 128 dimensions before being fed into SPR.

The node-classification benchmarks span citation, co-purchase, co-authorship, WebKB/Wikipedia, and large-scale OGB graphs; graph classification further evaluates transfer to whole-graph prediction. Dataset statistics are reported in Table 6.

Table 7: Runtime and GPU profiling results.

Dataset	Total Time	Forward Time	Linear Time	Transport Time	Peak GPU
ogbn-arxiv	12.5s	0.7s	3.1s	1.1s	3037 MiB
ogbn-products	53m 8.1s	51m 51.2s	1m 11.1s	5.8s	32099 MiB
Cora	8.4s	0.2s	1.8s	0.6s	911 MiB
Physics	1m 3.1s	0.8s	53.1s	2.1s	14307 MiB

F Complexity Analysis

Let n and e be the number of nodes and edges of an input graph, d_0 the raw feature dimension, d the unified spectral dimension, h the hidden dimension, d_z the output dimension, M_f the number of spectral filters, K_c the Chebyshev order, r the SVD rank, p the number of prototypes, and L the number of propagation layers.

Spectral parsing. The Chebyshev frontend applies M_f filters of order K_c. Since each Chebyshev recursion requires sparse multiplication with the normalized propagation operator S, the cost of computing all filter responses is $O(M_f K_c e d_0)$. For each filtered response, rank-r truncated SVD costs approximately $O(nd_0 r)$, giving $O(M_f n d_0 r)$ over all filters. The final projection from the concatenated $M_f r$-dimensional representation to d dimensions costs $O(nM_f rd)$. Therefore, the overall spectral parsing cost is

$$O(M_f K_c e d_0 + M_f n d_0 r + n M_f r d)\,. \tag{47}$$

The memory cost is $O(e + nd_0 + nM_f r)$, plus learnable parameters.

Prototype construction. For each source graph G_m, constructing a dense heat-kernel-distance matrix requires $O(n_m^2)$ memory. Exact heat-kernel computation by eigendecomposition costs $O(n_m^3)$, while the implementation uses an approximate normalized-adjacency surrogate for large graphs. In each GW update between a source graph and p prototypes, the linearized cost matrix costs $O(n_m^2 p + n_m p^2)$, and Sinkhorn normalization costs $O(n_m p)$ per iteration. With B barycenter outer iterations, G GW iterations, and S Sinkhorn iterations, the dominant prototype construction cost is

$$O\left(B \sum_m G(n_m^2 p + n_m p^2) + B \sum_m GS\, n_m p\right). \tag{48}$$

This cost is paid during pre-training and is amortized across downstream tasks.

Gated propagation. At each propagation layer, the adjacency path costs $O(eh)$. If the heat kernel is explicitly materialized, the heat path costs $O(n^2 h)$; for large graphs, the implementation instead uses sparse or approximate propagation. The prototype path is defined by $P = TK_\star T^\top$. Explicitly forming P would cost $O(np^2 + n^2 p)$, and applying it would cost $O(n^2 h)$. In practice, the operator is applied implicitly as $T(K_\star(T^\top H))$, which costs $O(nph + p^2 h)$ per layer, or $O(nkh + p^2 h)$ when each node keeps only k nonzero prototype assignments. The node-wise gate and hidden transformations cost $O(nh^2)$ when hidden-to-hidden projections dominate. Under sparse adjacency and sparse prototype transport, the total L-layer propagation cost is

$$O\big(L(eh + nkh + p^2 h + nh^2)\big)\,. \tag{49}$$

Adaptation to unseen graphs. For a regular-sized target graph, GW-based assignment has the same graph-to-prototype complexity as above. For a large target graph, the scalable K-means-based assignment costs $O(Inpd_t)$, where I is the number of iterations and d_t is the assignment feature dimension, followed by top-k sparsification. The downstream linear classifier costs $O(nd_z c)$ per epoch for c classes. The large-graph approximation is used for scalability and is not covered by the GW transport guarantee above.

Large-graph handling strategy. For large-scale graphs, we avoid dense $N \times N$ pairwise matrices and exact full-graph GW alignment. For ogbn-arxiv and ogbn-products, we keep the learned prototype

geometry fixed and replace only the target graph-to-prototype alignment with a K-means-based sparse assignment using 96 slots and top-k sparsification, together with sparse adjacency. This is a scalable approximation to the exact GW assignment and is not covered by Proposition 1. For ogbn-products, inference uses sampled subgraphs with batch size 1024 and fanouts $(10, 10, 5)$, followed by linear probing on frozen embeddings.