Signed Graph Pre-Training and Prompt Learning

Zihan Mei
Arizona State University
zihanmei@asu.edu

Rong Pan
Arizona State University
Rong.Pan@asu.edu

Yuzhou Chen
University of California, Riverside
Yuzhou.Chen@ucr.edu

Yixuan He*
Arizona State University
Yixuan.He@asu.edu

Abstract

Signed graphs arise in trust–distrust networks, financial correlation systems, biological interaction graphs, and many other domains in which edges can be positive or negative and may also be directed. While signed graph neural networks have improved task-specific learning, graph transfer learning on signed graphs remains underdeveloped. In this paper, we introduce TopoSIGN, a pioneering topology-guided graph pre-training and prompt learning framework for signed graphs. TopoSIGN combines a structural encoder built on the magnetic signed Laplacian with a novel persistent-homology branch that summarizes signed topology through Dowker-complex persistence images. The fused embeddings are then transferred to a prompt learning function. Experimental results on synthetic and real-world datasets demonstrate the efficacy of TopoSIGN in extracting useful structural information in signed graphs, as well as the adaptability and flexibility of the proposed general framework.

1 Introduction

Graphs with both positive and negative interactions appear naturally in applications where agreement and disagreement, promotion and inhibition, or attraction and repulsion coexist [1–3], including trust and distrust links in online social networks, competitive and synergistic relations in biological systems, and positively or negatively correlated financial assets. In many of these applications, edge polarity is as informative as connectivity itself, and in some cases, the graph is also directed [4], so both sign and orientation must be modeled jointly. These properties make signed graph representation learning fundamentally different from the standard unsigned setting.

Recent work on signed graph neural networks (GNNs) has made substantial progress on link sign prediction, node classification, and node clustering by explicitly modeling positive and negative relations, balance theory, or signed random walks [2, 4–8]. In particular, MSGNN [4] extends the magnetic Laplacian from directed graphs to a magnetic signed Laplacian, thereby handling directed signed graphs within one Hermitian spectral framework. However, most signed GNNs remain largely task-specific and are trained end-to-end for a single supervised objective. By contrast, graph pre-training, prompt learning, and graph foundation models aim to learn transferable representations that can be adapted to multiple downstream tasks with limited labels [9, 10]. This line of work has been highly active on unsigned graphs, but signed graphs have received far less attention. SGPT [11] is the only existing graph pre-training and fine-tuning work on signed graphs so far, but its goal is fundamentally different from ours: SGPT transfers knowledge from unsigned graph learning to signed learning tasks, while here we pretrain and fine-tune on signed tasks directly.

Topology is another underused source of signal in signed graph representation learning. Persistent homology (PH) and related topological data analysis (TDA) [12–17] tools provide multi-scale

*Corresponding author.

Mei et al., Signed Graph Pre-Training and Prompt Learning. *Proceedings of the Fifth Learning on Graphs Conference (LoG 2026)*, PMLR 338, Boston, MA, USA, November 20 – 22, 2026.

summaries of higher-order connectivity patterns beyond local neighborhoods. The recent directed-graph framework TopoDIG [18] combined a magnetic Laplacian encoder with a Dowker-complex topological branch [19] for pre-training and prompting on directed unsigned graphs. However, extending that idea to signed graphs is not trivial. In addition to a sign-aware GNN encoder, the TDA branch itself must be reconsidered: negative edges do not fit naturally into the unsigned filtration design used in prior work, and naively injecting signed weights into a scalar filtration can produce a poor match between the filtration semantics and the intended persistent-homology interpretation.

In this paper, we design for signed graph pre-training a two-branch structure, i.e., a spectral graph encoder plus a Dowker complex-based topological encoder, ensuring that they are aligned with the semantic needs of signed graphs. Concretely, we propose *TopoSIGN*, a **Topo**logy-guided pre-training and prompt learning model for **sign**ed graphs. First, we employ a signed directed GNN method, e.g., SSSNET [2] or MSGNN [4], for the structural encoder. Second, and more importantly, we introduce a signed degree-vector distance filtration that does not neglect signed information. That is, instead of assigning each node a single scalar degree in the filtration definition, we associate with each node a two-dimensional vector containing its positive and negative degrees and define a novel Dowker filtration through distances in this signed degree space. This allows us to retain signed information inside the simple yet effective topological branch without resorting to multiparameter persistence. After that, we fuse the learned embeddings of the two branches and apply a graph prompt learning paradigm [9, 20] for graph pre-training and fine-tuning.

The main contributions of this work are as follows.

- We formulate *TopoSIGN*, the first topology-guided graph pre-training and prompt-learning framework for signed graphs.
- We design for signed graphs a topology-empowered graph prompt function that improves the transfer and generalization capabilities of GNNs.
- We conduct experiments on both synthetic and real-world datasets with two instantiations of our framework and demonstrate its efficacy and flexibility. The framework is general and compatible with various signed GNNs and prompt learning functions.

2 Related work

2.1 Signed graph representation learning

Signed graph learning has a long history in spectral clustering, matrix factorization, and balance-theoretic embedding [21–23]. Neural approaches extend these ideas by designing message passing or spectral filters that distinguish positive and negative relations. SGCN uses balance theory to propagate separate representations through positive and negative neighborhoods [5]. SiGAT and SDGNN incorporate signed directed motifs and signed aggregation objectives for link sign prediction on directed signed networks [6, 24]. SNEA learns signed network embeddings with graph attention [25]. For clustering, SSSNET introduces a semi-supervised signed clustering objective that is not restricted to strong balance assumptions [2]. Spectral signed GNNs include SLGNN [26], frequency-based signed GNNs [27], SigMaNet [7], and MSGNN [4], which define signed or signed-directed Laplacian operators for signed representation learning. DSGC studies robust deep signed graph clustering through weak balance theory and graph denoising [8]. SE-SGformer develops a self-explainable signed graph transformer for link sign prediction with signed random-walk positional encodings [28]. CopulaLSP models inter-edge dependencies for scalable link sign prediction on signed graphs [29]. Here we pick MSGNN [4] as one of our structural backbones since it is simple and effective to complement the more computationally expensive topological branch for our task, and it can directly handle the tasks of our interest, i.e., link prediction and node clustering. SSSNET [2] is also picked due to its simplicity and efficacy.

2.2 Graph pre-training, prompting, and graph foundation models

Graph pre-training learns reusable representations through self-supervised objectives at node, edge, or graph level. Representative methods include contrastive approaches such as DGI [30]

Code is available at `https://github.com/Harrison-zh-M/TopoSIGN`.

and GraphCL [31], masked or reconstruction-based models such as GraphMAE [32] and GraphMAE2 [33], and task-specific pre-training frameworks such as S2PGNN [34] and BRep-BERT [35]. Prompt learning attempts to reduce the gap between pre-training and downstream tasks. GPPT uses edge prediction as the pretext task and introduces tokens to align node classification with pre-training [9]; GraphPrompt unifies graph pre-training and downstream tasks with prompt-based readout [36]; GPF learns universal graph prompts [37]; and All-in-One formulates multi-task graph prompting through meta-learning [10]. ProG provides a benchmark and systematic evaluation for graph prompt learning [20], while GCOPE [38] and OpenGraph [39] study cross-domain transfer and open graph foundation modeling, respectively. SAMGPT focuses on text-free multi-domain graph pre-training and cross-domain adaptation [40]. UniGraph learns a unified cross-domain model for text-attributed graphs [41]. GraphTOP adapts pre-trained GNNs by topology-oriented prompting through local edge rewiring [42]. TopoDIG [18] fuses topological features and a directed graph neural network encoder into the GPPT-like framework and extends the pre-training and prompt learning framework to directed graphs with asymmetric sending and receiving patterns. These methods establish strong transfer baselines, but most assume unsigned or text-attributed graphs and do not natively model signed Laplacian structure or signed topological filtrations. SGPT [11] is the only existing graph pre-training and fine-tuning work on signed graphs so far, but their goal is fundamentally different from ours: SGPT transfers knowledge from unsigned graph learning to signed learning tasks, while here we pretrain and fine-tune on signed tasks directly. Our approach gets inspiration from TopoDIG [18] by constructing a signed directed graph encoder and a topological encoder and feeding the concatenated outputs into a GPPT-like prompting framework.

2.3 Topological learning on graphs

Recent TDA-based graph representation learning uses persistent homology and related topological summaries to encode connectivity, cycles, and higher-order structures that are difficult for standard message passing to capture. Early neural and kernel-based methods include PersLay, which learns vectorized persistence-diagram representations and introduces graph topological signatures [43], P-WL, which augments Weisfeiler–Lehman subtree features with persistent cycle information [44], and Graph Filtration Learning, which learns a differentiable filtration function and uses persistent homology as a graph-level readout [45]. Later methods integrate topology more directly into GNN architectures: TRI-GNN rewires local graph neighborhoods using persistent homology and uses topological summaries as side information for node classification [46]; TOGL injects global persistent-homology features into message-passing GNNs and improves expressiveness beyond standard WL-limited aggregation [47]; and Wit-TopoPool uses persistent homology and witness complexes to design topology-aware pooling for graph classification [48]. More recent work shifts from graph-level summaries to localized and task-adaptive topological features. TTG-NN combines persistent homology, graph convolution, and tensor operations to capture local and global graph structure [49]; Yan et al. [50] use extended persistent homology on vicinity graphs to construct node- and edge-level topological features for node classification and link prediction; TensorMV-GCL incorporates extended persistent homology into multi-view graph contrastive learning [51]; and GraphTCL aligns GNN structural embeddings with persistent-homology embeddings through cross-view contrastive learning [52]. TopoDIG extends this direction to directed graphs by combining magnetic-Laplacian-based directed graph encoding with Dowker-complex-based topological features in a pre-training and prompting framework [18]. Despite this progress, almost all TDA-enhanced graph learning methods assume unsigned or merely weighted graphs. For signed graphs, existing topological work mainly characterizes structural balance using simplicial homology and cohomology [53], but does not provide a modern pre-training, prompting, or GNN representation-learning framework. Thus, signed topological graph learning remains largely open.

3 Methodology

3.1 Problem formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w, \mathbf{X})$ be a signed graph, possibly directed and weighted, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, $w : \mathcal{E} \to \mathbb{R} \setminus \{0\}$ assigns a nonzero signed weight to each edge, and $\mathbf{X} \in \mathbb{R}^{n \times F}$ is the node-feature matrix. We denote positive and negative edge sets by $\mathcal{E}^+ = \{(u, v) \in \mathcal{E} : w(u, v) > 0\}$ and $\mathcal{E}^- = \{(u, v) \in \mathcal{E} : w(u, v) < 0\}$. The signed adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is defined entrywise as $A_{uv} = w(u, v)$ if $(u, v) \in \mathcal{E}$ and zero otherwise. We further

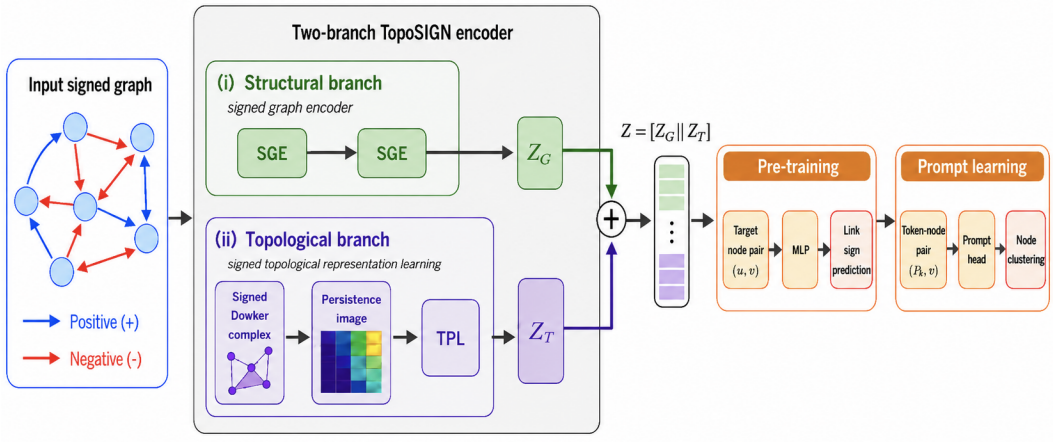


Figure 1: Overview of the TopoSIGN pre-training and prompting framework. Given an input signed (possibly also directed) graph with node attributes, TopoSIGN learns transferable representations through two complementary pathways (with the gray box). The first branch (i) applies signed graph encoder (SGE) layers to capture signed, directional, and local structural information. In parallel, the second branch (ii) constructs a Dowker complex to encode higher-order relational structures, which are transformed into compact topological embeddings via the topological representation learning (TPL) module using a projection layer. The structural and topological representations are fused to form joint node embeddings. Pre-training is performed using a pairwise prediction objective on both node-node and token-node pairs with a shared MLP and pretext loss, enabling topology-aware and prompt-compatible representations for downstream tasks.

define the positive and negative magnitude matrices $\mathbf{A}^+ = \max(\mathbf{A}, 0)$ and $\mathbf{A}^- = \max(-\mathbf{A}, 0)$, where the maximum is taken elementwise.

The pre-training task does not involve any node labels and is based completely on observed edges. Following [4], we consider different variations of the link prediction task for *signed and/or directed* networks. In our default pre-training task, link sign prediction (SP), one assumes that there is a link from u to v and aims to predict whether that link is positive or negative, i.e., whether $(u, v) \in \mathcal{E}^+$ or $(u, v) \in \mathcal{E}^-$. The downstream task is node clustering, whose goal is to partition the node set into a disjoint union of K clusters, $\mathcal{V} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{K-1}$.

Persistent Homology Persistent homology (PH) provides a principled framework for analyzing filtered simplicial complexes [54–56]. Early applications primarily focused on point clouds $\mathcal{X} \subset \mathbb{R}^N$, where Vietoris-Rips filtrations construct a sequence of nested complexes $\Delta_1(\mathcal{X}) \subset \Delta_2(\mathcal{X}) \subset \cdots$, enabling the tracking of topological features across multiple scales [57, 58]. The resulting persistence diagram $\mathcal{D}_q(\mathcal{X}) = \{(b_i, d_i)\}$ records the birth and death times of q-dimensional features (where $q \in \{0, 1, \dots\}$), with longer lifetimes $(d_i - b_i)$ typically interpreted as more structurally significant. PH has also been extended beyond point clouds to structured domains such as graphs and images. Unlike Euclidean data, graphs lack an intrinsic notion of scale, making the choice of filtration nontrivial. To address this, two principal approaches have been proposed. The first is the power-based filtration, which treats nodes as points in a metric space induced by graph distances and constructs a Vietoris-Rips filtration [59, 60]. While conceptually straightforward, this approach is often computationally expensive. The second, more practical approach is based on sublevel set filtrations, where a scalar function f defined on nodes or edges induces a sequence of nested subgraphs [45, 61]. These subgraphs are then lifted to simplicial complexes via clique expansions. A key interpretability distinction arises between these approaches. In power filtrations, persistence reflects geometric scale, whereas in sublevel set filtrations, it reflects variations in the function f. Consequently, long persistence does not necessarily correspond to large-scale structures. The standard PH pipeline for graphs consists of three steps, i.e., filtration, persistence computation, and vectorization. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a function $f : \mathcal{G} \to \mathbb{R}$, a sequence of thresholds $\{\alpha_i\}_{i=1}^n$ induces nested subgraphs $\mathcal{G}_1 \subset \cdots \subset \mathcal{G}_n$, $\mathcal{G}_i \subset \mathcal{G}$ satisfies $f(\mathcal{G}_i) \leq \alpha_i$. Each subgraph is lifted to its clique complex $\hat{\mathcal{G}}_i$, forming a filtration $\{\hat{\mathcal{G}}_i\}$. Persistence diagrams $\mathcal{D}_q(\mathcal{G}, f) = \{(b_j, d_j)\}$

summarize the birth and death of homological features $H_q(\hat{\mathcal{G}}_i)$, and can be converted into fixed-length representations via persistence images or persistence landscapes [62, 63].

Despite their effectiveness, these constructions often suffer from significant computational overhead, particularly for large graphs due to the combinatorial growth of simplicial complexes. To mitigate this challenge, in this paper, we adopt the Dowker complex [19], which constructs simplicial structures based on a bipartite relation between a subset of representative nodes (landmarks) and the remaining nodes (witnesses). That is, given a graph $\mathcal{G}$, a landmark set $L \subseteq \mathcal{V}$ and a witness set $W = \mathcal{V} \setminus L$ are defined, and simplices are formed based on proximity relations between landmarks and witnesses. This construction preserves essential topological features while significantly reducing computational complexity.

3.2 Overall framework of TopoSIGN

TopoSIGN consists of three modules: a structural signed GNN encoder (SGE, we apply MSGNN [4] and SSSNET [2] here), a signed topological encoder (with our novel design adapted from [18]), and a prompt module (we apply similar architectures as in [9]). The structural encoder maps the signed graph to node embeddings $\mathbf{Z}_{\mathcal{G}} \in \mathbb{R}^{n \times d_G}$. The topological encoder computes persistence-image features from a Dowker filtration and maps them to $\mathbf{Z}_{\mathcal{T}} \in \mathbb{R}^{n \times d_T}$. The fused node representation is $\mathbf{Z} = \left[\mathbf{Z}_{\mathcal{G}} \parallel \mathbf{Z}_{\mathcal{T}}\right] \in \mathbb{R}^{n \times (d_G + d_T)}$, where $[\cdot \parallel \cdot]$ denotes the concatenation function. The same fused representation is used for (link sign) pre-training and (cluster) prompt adaptation. A framework overview is provided in Fig. 1.

3.3 Signed graph encoder (SGE)

The SGE branch is responsible for modeling graph structural information in terms of sign, direction, and spectral smoothness over the full signed graph, and may contain one or more SGE layers using different SGE backbones from signed GNNs.

One instantiation of the structural branch of TopoSIGN uses the magnetic signed Laplacian introduced by [4] to encode signed local and global structure. First, define the symmetrized signed adjacency and degree matrices by

$$\widetilde{\mathbf{A}}_{uv} = \frac{1}{2}(A_{uv} + A_{vu}), \qquad \widetilde{\mathbf{D}}_{uu} = \frac{1}{2}\sum_{v=1}^{n}\big(\,|A_{uv}| + |A_{vu}|\,\big). \tag{1}$$

The directional asymmetry is encoded through the phase matrix $\mathbf{\Theta}^{(q)}_{uv} = 2\pi q\,(A_{uv} - A_{vu})$, where $q \in \mathbb{R}$ is the "charge parameter". The corresponding Hermitian adjacency matrix is $\mathbf{H}^{(q)} = \widetilde{\mathbf{A}} \odot \exp(\mathrm{i}\mathbf{\Theta}^{(q)})$, where $\odot$ denotes the Hadamard product. MSGNN then defines the unnormalized and normalized magnetic signed Laplacian matrices by

$$\mathbf{L}^{(q)}_U = \widetilde{\mathbf{D}} - \mathbf{H}^{(q)}, \qquad \mathbf{L}^{(q)}_N = \mathbf{I} - \left(\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}\right) \odot \exp(\mathrm{i}\mathbf{\Theta}^{(q)}). \tag{2}$$

[4] defines spectral convolution by diagonalizing a Hermitian Laplacian $\mathbf{L}$ and approximating the resulting filter with Chebyshev polynomials. In our instantiation, with $\mathbf{L} = \mathbf{L}^{(q)}_N$ and choosing the second-order polynomial approximation, the layer update becomes

$$\mathbf{X}^{(\ell)} = \sigma\Big(\mathbf{X}^{(\ell-1)}\mathbf{W}^{(\ell)}_{\text{self}} + \widetilde{\mathbf{L}}^{(q)}_N\mathbf{X}^{(\ell-1)}\mathbf{W}^{(\ell)}_{\text{neigh}} + \mathbf{B}^{(\ell)}\Big), \tag{3}$$

where $\mathbf{W}^{(\ell)}_{\text{self}}$ and $\mathbf{W}^{(\ell)}_{\text{neigh}}$ are learned weight matrices corresponding to the filter weights of different channels and $\mathbf{B}^{(\ell)}$ is the bias vector, $\widetilde{\mathbf{L}}^{(q)}_N = \frac{2}{\lambda_{\max}}\mathbf{L}^{(q)}_N - \mathbf{I}$ is the rescaled Laplacian with identity matrix $\mathbf{I}$, and σ is a nonlinear activation function. As in [4, 64], we use a complex version of the Rectified Linear Unit defined by $\sigma(z) = z$, if $-\pi/2 \le \arg(z) < \pi/2$, and $\sigma(z) = 0$ otherwise, where $\arg(\cdot)$ is the complex argument of $z \in \mathbb{C}$. After the final layer, the complex representation is unwound into a real-valued embedding by concatenating its real and imaginary parts. We use $q = 0.25$ by default.

Another instantiation we use is SSSNET (SIMPA) from [2]. Its core component is the SIMPA aggregation mechanism, which separately propagates information over positive and negative neighborhoods to preserve the distinct semantics of friendship and antagonism relations. Specifically,

SSSNET computes decoupled embeddings from multiple signed propagation paths and combines them to capture agreement, disagreement, and neutral interactions across the graph. Owing to its lightweight propagation structure and effectiveness on signed community detection benchmarks, SSSNET serves as a simple yet strong structural encoder in our framework.

3.4 Signed topological representation learning

The topological branch complements the SGE embedding with a multiscale signed summary derived from persistent homology on Dowker complexes. The key design choice is how to define a scalar filtration that remains meaningful for signed graphs.

Positive-subgraph baseline. The simplest sign-compatible baseline removes negative edges before topology is computed. Let $\mathcal{G}^+ = (\mathcal{V}, \mathcal{E}^+, \mathbf{A}^+)$ be the positive subgraph. Its degree for node u is $d^+(u) = \sum_{v=1}^{n} \left(A^+_{uv} + A^+_{vu}\right)$.

Given landmark and witness sets $L, W \subseteq \mathcal{V}$, we define a scalar landmark–witness distance by

$$\delta_+(\ell, w) = \left|d^+(\ell) - d^+(w)\right|. \tag{4}$$

For threshold $\varepsilon \geq 0$, the associated Dowker complex is

$$D^+_\varepsilon(L, W) = \{\sigma \subseteq L \mid \exists w \in W \text{ such that } \delta_+(\ell, w) \leq \varepsilon,\ \forall \ell \in \sigma\}. \tag{5}$$

Sweeping ε produces a one-parameter filtration. This baseline is useful because it keeps the PH semantics clean and provides a direct comparison point for any sign-aware topological design.

Signed degree-vector distance filtration. Our proposed topological variant uses the full signed graph instead of discarding negative edges. For each node u, define the positive and negative degree magnitudes

$$d^+(u) = \sum_{v=1}^{n} \left(A^+_{uv} + A^+_{vu}\right), \qquad d^-(u) = \sum_{v=1}^{n} \left(A^-_{uv} + A^-_{vu}\right). \tag{6}$$

The negative degree is nonnegative by construction because $\mathbf{A}^-$ stores absolute magnitudes of negative edges. We then associate to node u the two-dimensional signed degree vector

$$\mathbf{s}(u) = \begin{bmatrix} d^+(u) \\ d^-(u) \end{bmatrix} \in \mathbb{R}^2. \tag{7}$$

The landmark–witness distance is defined by Euclidean distance in this degree space

$$\delta_\pm(\ell, w) = \left\|\mathbf{s}(\ell) - \mathbf{s}(w)\right\|_2. \tag{8}$$

The resulting signed Dowker filtration is

$$D^\pm_\varepsilon(L, W) = \{\sigma \subseteq L \mid \exists w \in W \text{ such that } \delta_\pm(\ell, w) \leq \varepsilon,\ \forall \ell \in \sigma\}. \tag{9}$$

This construction keeps the filtration one-dimensional, which allows standard PH software and persistence-image vectorizations to be used unchanged, while preserving information about how much positive and negative connectivity each node carries. Nodes with similar total degree but very different sign composition are therefore distinguished by the filtration. Additionally, to further improve scalability, we adopt a landmark–witness strategy: a subset of nodes is selected as landmarks (e.g., via either degree- or centrality-based sampling), which serve as the backbone for the filtration, while the remaining nodes act as witnesses that attach to nearby landmarks according to their signed connectivity. This design preserves the essential signed structural patterns while reducing computational complexity.

Persistence images and topological encoder. From a filtration family we compute persistence diagrams in dimensions 0 and 1 and then convert them to persistence images (PIs) [62]. Since here we are interested in node-level representations, for each node, we construct topological features based on its two-hop ego-network (much smaller than the whole network). For each two-hop ego-network, we define a filtration based on node degrees. This yields a nested sequence of subgraphs, from which we compute H_0 (connectivity) and H_1 (cycles) persistence diagrams that capture the local topology across scales. Because these diagrams vary in size across nodes, we transform them into PIs—a fixed-dimensional vector representation. We achieve this by mapping the diagrams to *(birth, persistence)*

coordinates, convolving the points with a Gaussian kernel weighted by a persistence-dependent function, and discretizing the resulting surface into a $d_I \times d_I$ grid, where d_I denotes the resolution of the image. This ensures a consistent input size for downstream machine learning models while preserving the topological signatures of the local neighborhood. Flattening the grid into a vector of a fixed length $d_T = d_I^2$ gives the topological representations of each node.

Let $\mathbf{Z}_{\mathcal{T}} \in \mathbb{R}^{n \times d_T}$ contain all topological feature vectors in its rows, we then fuse it with the SGE embedding:

$$\mathbf{Z} = \left[\mathbf{Z}_{\mathcal{G}} \parallel \mathbf{Z}_{\mathcal{T}}\right] \in \mathbb{R}^{n \times (d_G + d_T)}. \tag{10}$$

The fused representation $\mathbf{Z}$ is used by both the pre-training loss and the downstream prompt head.

3.5 Pre-training with link sign prediction

The pre-training task is link sign prediction. Let $\Omega_{\text{tr}} \subseteq \mathcal{E}$ denote the training edge set produced by a signed edge split. Each training edge receives a binary sign label,

$$y_{uv} = \begin{cases} 1, & A_{uv} > 0, \\ 0, & A_{uv} < 0. \end{cases} \tag{11}$$

Given the fused node embeddings $\mathbf{z}_u$ and $\mathbf{z}_v$ from Eq. (10), an edge decoder $\psi_\zeta(\cdot)$ predicts the edge sign,

$$\hat{y}_{uv} = \text{sigmoid}(\cdot)\big(\psi_\zeta([\mathbf{z}_u \parallel \mathbf{z}_v])\big). \tag{12}$$

The default pre-training loss is binary cross-entropy,

$$\mathcal{L}_{\text{SP}} = -\sum_{(u,v)\in\Omega_{\text{tr}}} \left[y_{uv} \log \hat{y}_{uv} + (1 - y_{uv}) \log(1 - \hat{y}_{uv})\right]. \tag{13}$$

3.6 Cluster prompts for downstream adaptation

Due to the limited node classification labels in the signed graph learning literature, we focus on the task of node clustering, where abundant synthetic datasets are available [3]. To adapt the GPPT [9] idea to this setting, we replace class prompts with cluster prompts. Let M be the number of METIS [65] clusters (note these clusters typically differ from our downstream K clusters, but are constructed since varying task tokens in different densely-connected clusters may provide enhanced task embeddings). We introduce a learnable prompt bank

$$\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_M\}, \qquad \mathbf{p}_m \in \mathbb{R}^{d_P}. \tag{14}$$

For node u (belonging to a certain METIS cluster m) and a downstream cluster k, we construct a prompted token

$$\boldsymbol{\xi}_{u,k} = [\mathbf{p}_k^m \parallel \mathbf{z}_u], \tag{15}$$

and score cluster compatibility with a small task head g_ϕ,

$$s(u,k) = g_\phi(\boldsymbol{\xi}_{u,k}), \qquad \hat{c}_u = \arg\max_{k\in\{1,\ldots,K\}} s(u,k). \tag{16}$$

If a training set $\mathcal{S} \subseteq \mathcal{V}$ of labeled nodes is available, the adaptation objective is a negative log-likelihood (NLL) loss function as

$$\mathcal{L}_{\text{cluster}} = \sum_{u\in\mathcal{S}} \text{NLL}\big(\text{LogSoftmax}(s(u,\cdot)), c_u\big), \tag{17}$$

where $c_u \in \{1, \ldots, K\}$ is the seed-node cluster label. In practice, the pre-trained encoder can be frozen and only the prompt bank and task head are tuned, or a lightweight adapter can be updated jointly.

4 Experiments

We conduct experiments on both synthetic and real-world datasets to evaluate our proposed TopoSIGN framework against baselines and its variants. Additional results are provided in Appendix A, while implementation details are provided in Appendix B.

4.1 Datasets and evaluation metrics

To evaluate the effectiveness of the proposed TopoSIGN framework, we conduct a comprehensive comparison on both synthetic benchmarks (SDSBM from [4]) and real-world datasets (Rainfall and SP1500 from [2]). Rainfall and SP1500 represent the only publicly available real-world signed graphs that provide both ground-truth node labels and sufficient scale required for self-supervised pre-training and node clustering evaluation to the best of our knowledge. Dataset statistics are provided in Appendix Table 17. We adopt signed link prediction for self-supervised pre-training, as it serves as a foundational task to capture intrinsic signed topological patterns without label reliance; these learned structural representations are subsequently transferred via prompts to guide node clustering, a downstream task directly aligned with node ground-truth annotations. Node clustering performance is measured using the Adjusted Rand Index (ARI) [66] for node clustering tasks. ARI measures the similarity between predicted node clusters and ground-truth labels, where a score of 1 indicates a perfect match, and 0 indicates random clustering, making it a robust measure for evaluating how well a model captures structural communities.

Specifically, the synthetic SDSBM [4] graphs are generated to simulate diverse community structures and noise levels. We employ the F_1 meta-graph configuration to define community interactions, where a size ratio ρ characterizes the imbalance between the largest and smallest blocks, and p determines the overall edge density. To account for structural uncertainty, we incorporate directional noise γ and sign flip probability η, which perturb the topology and edge signs, respectively. In our experiments, we set the SDSBM under the following specific configuration:

- **Setting 1 (SDSBM-1)**:$F_1(\gamma = 0.25)$, $n = 1000$, $p = 0.1$, $\rho = 1.5$, and $\eta = 0.25$.
- **Setting 2 (SDSBM-2)**:$F_2(\gamma = 0.25)$, $n = 1000$, $p = 0.1$, $\rho = 1.5$, and $\eta = 0.25$.
- **Setting 3 (SDSBM-3)**:$F_1(\gamma = 0.1)$, $n = 1000$, $p = 0.1$, $\rho = 1.5$, and $\eta = 0.25$.
- **Setting 4 (SDSBM-4)**:$F_1(\gamma = 0)$, $n = 1000$, $p = 0.1$, $\rho = 1.5$, and $\eta = 0$. Note that setting 4 is a noiseless setting.

4.2 Baselines and experiment setups

Our main experiments are designed to answer two research questions: (1) How effective is TopoSIGN in transferring link prediction knowledge to node clustering, compared with other pre-training and fine-tuning methods? (2) How effective is TopoSIGN in learning node representations compared with other signed GNNs if we directly apply a semi-supervised node clustering task without any pre-training?

To answer these research questions, we compare TopoSIGN against two families of representative baselines. The first family consists of graph pre-training, prompting, and foundation-model methods: GPPT [9], All-in-one [10], GraphPrompt [36], GPF [37], SAMGPT [40], and TopoDIG [18]. The second family consists of signed graph clustering models: SSSNET [2], SigMaNet [7], MSGNN [4], and DSGC [8]. For a fair comparison on different architectures, we employ their original hyperparameter settings but use the same loss function for all methods during pre-training and the same NLL loss for prompt learning or semi-supervised node clustering. Hyperparameter settings are provided in Appendix B.3.

4.3 Results

Table 1 compares our proposed TopoSIGN (TopoMSGNN with SGE backbone MSGNN [4] and TopoSSSNET with SGE backbone SSSNET [2]) against existing graph pre-training and prompt learning methods. TopoSIGN achieves the strongest or near-strongest performance on the noisy SDSBM settings. On SDSBM-1, SDSBM-2, and SDSBM-3, TopoSSSNET obtains the best ARI among all pre-training and prompt learning methods, while TopoMSGNN is consistently competitive and improves over TopoDIG. This comparison is particularly informative because TopoDIG also contains a topological branch, but it was designed for directed unsigned graphs. The improvement of TopoSIGN over TopoDIG indicates that simply transferring a directed-graph topological pre-training framework to signed graphs is not enough; the filtration and structural encoder must be adapted to preserve sign information. The results on SDSBM-4 reveal a different pattern. SDSBM-4 is noiseless, and several unsigned prompt learning baselines perform well, especially SAMGPT, GPF, Gprompt, and All-in-one. This suggests that when the clustering signal is clean and strong, generic graph pre-training can already recover useful community structure, even without explicit signed topology.

In contrast, TopoSIGN is more advantageous in the noisy signed settings, where edge signs and higher-order signed connectivity patterns are harder to exploit through standard message passing alone. This supports the main motivation of TopoSIGN: the topological branch is most useful when local signed neighborhoods are perturbed and when transferable representations must capture more stable structural regularities.

The real-world datasets are more challenging. This may be due to the dense and highly structured nature of these real-world graphs, the mismatch between link-sign pre-training and the final clustering objective, and the fact that the current signed degree-vector filtration captures only one type of signed topology. Nevertheless, TopoSIGN remains competitive on Rainfall and shows that the proposed framework can be applied beyond synthetic signed block models. The weaker SP1500 results suggest that future work should consider richer signed-directed filtrations, edge-feature-aware topological summaries, and more adaptive prompt construction for dense financial networks.

Table 1: Pre-training and prompt learning performance comparison in terms of downstream node clustering ARI. We report the mean ARI among five runs ± standard deviation. The best method is marked in **bold** while the second best is marked with underline.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
GPPT	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
Gprompt	0.026 ± 0.027	0.018 ± 0.030	0.169 ± 0.141	0.401 ± 0.201	$\mathbf{0.160 \pm 0.047}$	0.066 ± 0.033
GPF	0.016 ± 0.015	0.026 ± 0.023	0.132 ± 0.116	$\underline{0.403 \pm 0.202}$	0.092 ± 0.075	$\mathbf{0.081 \pm 0.011}$
All-in-one	0.015 ± 0.016	0.036 ± 0.034	0.148 ± 0.099	0.403 ± 0.199	$\underline{0.124 \pm 0.024}$	0.046 ± 0.038
SAMGPT	0.008 ± 0.009	0.037 ± 0.044	0.122 ± 0.096	$\mathbf{0.477 \pm 0.045}$	0.090 ± 0.074	$\underline{0.078 \pm 0.006}$
TopoDIG	0.030 ± 0.022	0.094 ± 0.033	0.307 ± 0.057	0.162 ± 0.017	0.000 ± 0.000	0.010 ± 0.017
TopoMSGNN (Ours)	$\underline{0.032 \pm 0.030}$	$\underline{0.107 \pm 0.009}$	$\underline{0.372 \pm 0.042}$	0.164 ± 0.025	0.075 ± 0.061	0.000 ± 0.000
TopoSSSNET (Ours)	$\mathbf{0.076 \pm 0.031}$	$\mathbf{0.125 \pm 0.011}$	$\mathbf{0.379 \pm 0.036}$	0.157 ± 0.023	0.062 ± 0.057	0.000 ± 0.000

Table 2 evaluates the topological branch in semi-supervised clustering. Adding topological features consistently boosts several signed encoders. The gain is most pronounced for SSSNET across all datasets—particularly SDSBM-1–3, Rainfall, and SP1500—suggesting that persistence-image features complement its signed propagation by capturing multiscale local connectivity and cycle structures. Extensive evaluations across various backbone-prompt combinations (Appendix Tables 5–7) and low-data regimes (1, 3, and 5 shots; Appendix Tables 8–16) further confirm TopoSIGN's efficacy, especially under extreme label scarcity and on complex real-world graphs. Notably, TopoSIGN slightly degrades DSGC, likely because DSGC's topology-optimizing pre-processing makes additional topological features redundant or noisy.

Table 2: Semi-supervised node clustering performance (ARI) on real-world and synthetic signed datasets. We compare our approach against state-of-the-art signed GNNs.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
DSGC	0.134 ± 0.231	0.184 ± 0.143	0.197 ± 0.147	0.480 ± 0.245	0.102 ± 0.056	0.093 ± 0.012
SigMaNet	0.333 ± 0.080	0.230 ± 0.121	0.226 ± 0.141	0.705 ± 0.139	0.236 ± 0.130	0.075 ± 0.040
MSGNN	0.703 ± 0.047	0.714 ± 0.012	0.790 ± 0.014	0.988 ± 0.008	0.283 ± 0.183	0.116 ± 0.050
SSSNET	0.180 ± 0.244	0.052 ± 0.065	0.218 ± 0.178	0.431 ± 0.387	0.118 ± 0.236	0.009 ± 0.014
DSGC+Topo	$\underline{0.762 \pm 0.133}$	$\underline{0.777 \pm 0.160}$	$\underline{0.911 \pm 0.083}$	$\mathbf{0.989 \pm 0.009}$	$\underline{0.472 \pm 0.044}$	0.156 ± 0.024
SigMaNet+Topo	0.169 ± 0.015	0.248 ± 0.062	0.284 ± 0.034	0.458 ± 0.234	0.174 ± 0.042	0.072 ± 0.006
MSGNN+Topo	0.684 ± 0.022	0.559 ± 0.105	0.674 ± 0.054	$\underline{0.974 \pm 0.022}$	0.435 ± 0.084	$\underline{0.170 \pm 0.013}$
SSSNET+Topo	$\mathbf{0.923 \pm 0.030}$	$\mathbf{0.923 \pm 0.029}$	$\mathbf{0.937 \pm 0.047}$	0.967 ± 0.028	$\mathbf{0.481 \pm 0.034}$	$\mathbf{0.298 \pm 0.041}$

4.4 Ablation study

To demonstrate the efficacy of each component of our framework, we perform an ablation study to isolate the contribution of each component. From Table 3 in Appendix A, we conclude that without the topological branch or without SGE, the performance typically drops (especially for MSGNN). Using a signed filtration is better than a positive-sign subgraph only topological baseline. Further results and discussions on different pre-training tasks, other prompting functions, as well as additional SGE compatibility, are provided in Appendix A.

4.5 Computational Cost

We further analyze the computational cost of TopoSIGN by separating the one-time topological preprocessing from model training and evaluation. Persistence images are extracted offline from

node-level ego-networks and can be cached and reused across subsequent training and evaluation runs. Although this preprocessing introduces an additional upfront cost, the node-wise computation is naturally parallelizable, and incorporating the precomputed topological features incurs only moderate overhead during model training compared with the corresponding signed GNN backbones. The preprocessing cost becomes more pronounced for dense graphs such as SP1500, suggesting that more scalable topological feature extraction is an important direction for future work. Detailed preprocessing and training runtimes are reported in Appendix B.4.

4.6 Parameter Sensitivity

We also evaluate the sensitivity of TopoSIGN to two key design parameters of the topological branch: the ego-network radius and the persistence-image resolution. Overall, the prompt-learning results remain relatively stable when increasing the neighborhood radius from 2 to 3 hops and across different persistence-image resolutions, while the semi-supervised setting exhibits greater sensitivity to the image resolution. A larger neighborhood can improve performance for several signed GNN backbones but requires more expensive topological preprocessing, whereas overly fine or coarse persistence-image discretizations may respectively introduce sparse/noisy representations or lose fine-grained topological information. These observations support our default choices of a 2-hop ego-network and a pixel size of 1.0 as practical settings that balance performance, robustness, and computational cost. Full sensitivity results are provided in Appendix B.5.1 and B.5.2.

5 Conclusion

We introduced TopoSIGN, a pioneering topology-guided pre-training and prompt-learning framework for signed graphs. TopoSIGN combines signed GNN structural embeddings with persistent-homology features from signed Dowker filtrations, pre-trains on link sign prediction, and adapts to node clustering with cluster prompts. TopoSIGN is compatible with various signed GNN backbones and prompt functions. One key methodological component is a signed degree-vector filtration that retains positive and negative degree information while remaining compatible with standard one-parameter persistence. Future work includes richer signed-directed filtrations based on separate in/out positive and negative degrees, dynamic signed graph pre-training, edge-feature-aware signed topology, and scalable landmark selection for large signed networks. We do not envision severe negative societal impacts.

6 Acknowledgments

Yixuan He is supported by a Jetstream2 NAIRR AI Fellowship. This work used Jetstream2 at Indiana University [67] through allocation CIS251396 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program [68], which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650, 2010. 1, 21

[2] Yixuan He, Gesine Reinert, Songchao Wang, and Mihai Cucuringu. SSSNET: Semi-Supervised Signed Network Clustering. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 244–252. SIAM, 2022. 1, 2, 5, 8

[3] Yixuan He, Xitong Zhang, Junjie Huang, Benedek Rozemberczki, Mihai Cucuringu, and Gesine Reinert. Pytorch geometric signed directed: A software package on graph neural networks for signed and directed graphs. In *Learning on Graphs Conference*, pages 12–1. PMLR, 2024. 1, 7

[4] Yixuan He, Michael Perlmutter, Gesine Reinert, and Mihai Cucuringu. MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian. In *Learning on Graphs Conference*, pages 40–1. PMLR, 2022. 1, 2, 4, 5, 8

[5] Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE international conference on data mining (ICDM)*, pages 929–934. IEEE, 2018. 2

[6] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Sdgnn: Learning node representation for signed directed networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 196–203, 2021. 2

[7] Stefano Fiorini, Stefano Coniglio, Michele Ciavotta, and Enza Messina. Sigmanet: One laplacian to rule them all. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7568–7576, 2023. 2, 8

[8] Peiyao Zhao, Xin Li, Zeyu Zhang, Mingzhong Wang, Xueying Zhu, and Lejian Liao. Robust deep signed graph clustering via weak balance theory. In *Proceedings of the ACM on Web Conference 2025*, pages 3819–3830, 2025. 1, 2, 8

[9] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. GPPT: Graph pre-training and prompt tuning to generalize graph neural networks. In *ACM SIGKDD*, 2022. 1, 2, 3, 5, 7, 8

[10] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in One: Multi-task prompting for graph neural networks. In *ACM SIGKDD*, 2023. 1, 3, 8

[11] Zian Zhai, Qing Sima, Xiaoyang Wang, and Wenjie Zhang. Sgpt: Few-shot prompt tuning for signed graphs. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 4045–4055, 2025. 1, 3

[12] Larry Wasserman. Topological data analysis. *Annual review of statistics and its application*, 5 (2018):501–532, 2018. 1

[13] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4:667963, 2021.

[14] Gunnar Carlsson, Rick Jardine, Dmitry Feichtner-Kozlov, Dmitriy Morozov, Frédéric Chazal, Vin de Silva, Brittany Fasy, Jesse Johnson, Matt Kahle, Gilad Lerman, et al. Topological data analysis and machine learning theory. In *BIRS Workshop*, pages 1–11, 2012.

[15] Yuzhou Chen and Yulia Gel. Topological zigzag spaghetti for diffusion-based generation and prediction on graphs. In *Proceedings of the International Conference on Learning Representations*, 2025.

[16] Matthew Dixon, Yuzhou Chen, and Yulia R Gel. Topological k-means clustering in reproducing kernel hilbert spaces. *Electronic Journal of Statistics*, 19(1):204–239, 2025.

[17] Yuzhou Chen, Ignacio Segovia, and Yulia R Gel. Z-GCNETs: Time zigzags at graph convolutional networks for time series forecasting. In *Proceedings of the International conference on machine learning*, pages 1684–1694. PMLR, 2021. 1

[18] Peiyu Liang, Chenguang Yang, Yixuan He, Rong Pan, and Yuzhou Chen. Topology-guided graph pre-training and prompt learning on directed graphs. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=kMIdkLTys8. 2, 3, 5, 8

[19] Clifford H Dowker. Topology of metric complexes. *American journal of mathematics*, 74(3): 555–577, 1952. 2, 5

[20] Chenyi Zi, Haihong Zhao, Xiangguo Sun, Yiqing Lin, Hong Cheng, and Jia Li. Prog: A graph prompt learning benchmark. *Advances in Neural Information Processing Systems*, 37: 95406–95437, 2024. 2, 3

[21] Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. Sponge: A generalized eigenproblem for clustering signed networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1088–1098. PMLR, 2019. 2

[22] Dimitrios Rafailidis. Modeling trust and distrust information in recommender systems via joint matrix factorization with signed graphs. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 1060–1065, 2016.

[23] Zeyu Zhang, Peiyao Zhao, Xin Li, Jiamou Liu, Xinrui Zhang, Junjie Huang, and Xiaofeng Zhu. Signed graph representation learning: A survey. *arXiv preprint arXiv:2402.15980*, 2024. 2

[24] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Signed graph attention networks. In *International conference on artificial neural networks*, pages 566–577. Springer, 2019. 2

[25] Yu Li, Yuan Tian, Jiawei Zhang, and Yi Chang. Learning signed network embedding via graph attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4772–4779, 2020. 2

[26] Yu Li, Meng Qu, Jian Tang, and Yi Chang. Signed laplacian graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4444–4452, 2023. 2

[27] Yang Chen, Menglin Wang, Changhao Li, and Cheng Cheng. Graph fourier transforms on signed graphs. In *Proceedings of the 2024 12th International Conference on Communications and Broadband Networking*, pages 96–101, 2024. 2

[28] Lu Li, Jiale Liu, Xingyu Ji, Maojun Wang, and Zeyu Zhang. Self-explainable graph transformer for link sign prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12084–12092, 2025. 2

[29] Jinkyu Sung, Myunggeum Jee, and Joonseok Lee. A scalable inter-edge correlation modeling in copulagnn for link sign prediction. *arXiv preprint arXiv:2601.19175*, 2026. 2

[30] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *stat*, 1050:21, 2018. 2

[31] 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. 3

[32] 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*, pages 594–604, 2022. 3

[33] 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. 3

[34] Wang Zhili, DI Shimin, CHEN Lei, and ZHOU Xiaofang. Search to fine-tune pre-trained graph neural networks for graph-level tasks. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 2805–2819. IEEE, 2024. 3

[35] Yunzhong Lou, Xueyang Li, Haotian Chen, and Xiangdong Zhou. Brep-bert: Pre-training boundary representation bert with sub-graph node contrastive learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1657–1666, 2023. 3

[36] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. GraphPrompt: Unifying pre-training and downstream tasks for graph neural networks. In *Web Conference*, pages 417–428, 2023. 3, 8

[37] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 2023. 3, 8

[38] 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, 2024. 3

[39] Lianghao Xia, Ben Kao, and Chao Huang. Opengraph: Towards open graph foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2365–2379, 2024. 3

[40] 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. 3, 8

[41] 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. 3

[42] Xingbo Fu, Zhenyu Lei, Zihan Chen, Binchi Zhang, Chuxu Zhang, and Jundong Li. Graphtop: Graph topology-oriented prompting for graph neural networks. *arXiv preprint arXiv:2510.22451*, 2025. 3

[43] Mathieu Carriere, Frederic Chazal, Yuichi Ike, Theo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures.

In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2786–2796. PMLR, 2020. URL https://proceedings.mlr.press/v108/carriere20a.html. 3

[44] Bastian Rieck, Christian Bock, and Karsten Borgwardt. A persistent weisfeiler-lehman procedure for graph classification. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5448–5458. PMLR, 2019. URL https://proceedings.mlr.press/v97/rieck19a.html. 3

[45] Christoph Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In *International Conference on Machine Learning*, pages 4314–4323. PMLR, 2020. 3, 4

[46] Yuzhou Chen, Baris Coskunuzer, and Yulia Gel. Topological relational learning on graphs. *Advances in Neural Information Processing Systems*, 34:27029–27042, 2021. 3

[47] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=oxxUMeFwEHd. 3

[48] Yuzhou Chen and Yulia R. Gel. Topological pooling on graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7096–7103, 2023. doi: 10.1609/aaai.v37i6.25866. URL https://ojs.aaai.org/index.php/AAAI/article/view/25866. 3

[49] Tao Wen, Elynn Chen, and Yuzhou Chen. Tensor-view topological graph neural network. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4330–4338. PMLR, 2024. URL https://proceedings.mlr.press/v238/wen24a.html. 3

[50] Zuoyu Yan, Qi Zhao, Ze Ye, Tengfei Ma, Liangcai Gao, Zhi Tang, Yusu Wang, and Chao Chen. Enhancing graph representation learning with localized topological features. *Journal of Machine Learning Research*, 26(5):1–36, 2025. URL https://www.jmlr.org/papers/v26/23-1424.html. 3

[51] Yujia Wu, Junyi Mo, Elynn Chen, and Yuzhou Chen. Tensor-fused multi-view graph contrastive learning. In *Data Science: Foundations and Applications*, volume 15876 of *Lecture Notes in Computer Science*, pages 16–28. Springer, 2025. doi: 10.1007/978-981-96-8298-0_2. URL https://link.springer.com/book/10.1007/978-981-96-8298-0. 3

[52] Ahmet Sami Korkmaz, Selim Coskunuzer, and Md Joshem Uddin. Cross-view topology-aware graph representation learning, 2025. URL https://arxiv.org/abs/2512.02130. 3

[53] Baike She and Zhen Kan. Algebraic topological characterizations of structural balance in signed graphs. *Automatica*, 107:61–67, 2019. doi: 10.1016/j.automatica.2019.05.029. URL https://www.sciencedirect.com/science/article/pii/S0005109819302456. 3

[54] Herbert Edelsbrunner, John Harer, et al. Persistent homology-a survey. *Contemporary mathematics*, 453(26):257–282, 2008. 4

[55] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 347–356, 2004.

[56] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ data science*, 6(1):17, 2017. 4

[57] Thibault de Surrel, Felix Hensel, Mathieu Carrière, Théo Lacombe, Yuichi Ike, Hiroaki Kurihara, Marc Glisse, and Frédéric Chazal. Ripsnet: a general architecture for fast and robust estimation of the persistent homology of point clouds. In *Topological, algebraic and geometric learning workshops 2022*, pages 96–106. PMLR, 2022. 4

[58] Renata Turkes, Guido F Montufar, and Nina Otter. On the effectiveness of persistent homology. *Advances in Neural Information Processing Systems*, 35:35432–35448, 2022. 4

[59] Donald R Sheehy. Linear-size approximations to the vietoris-rips filtration. In *Proceedings of the twenty-eighth annual symposium on Computational geometry*, pages 239–248, 2012. 4

[60] Mehmet E Aktas, Esra Akbas, and Ahmed El Fatmaoui. Persistence homology of networks: methods and applications. *Applied Network Science*, 4(1):1–28, 2019. 4

[61] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *Advances in neural information processing systems*, 30, 2017. 4

[62] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *JMLR*, 2017. 5, 6

[63] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015. 5

[64] Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. MagNet: A Neural Network for Directed Graphs. In *Advances in Neural Information Processing Systems*, 2021. 5

[65] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998. 7

[66] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1): 193–218, 1985. 8

[67] David Y. Hancock, Jeremy Fischer, John Michael Lowe, Winona Snapp-Childs, Marlon Pierce, Suresh Marru, J. Eric Coulter, Matthew Vaughn, Brian Beck, Nirav Merchant, Edwin Skidmore, and Gwen Jacobs. Jetstream2: Accelerating cloud computing via jetstream. In *Practice and Experience in Advanced Research Computing (PEARC '21)*, pages 1–8, New York, NY, USA, 2021. Association for Computing Machinery. doi: 10.1145/3437359.3465565. 10

[68] Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. ACCESS: Advancing innovation: NSF's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing (PEARC '23)*, pages 1–4, New York, NY, USA, 2023. Association for Computing Machinery. doi: 10.1145/3569951.3597559. 10

[69] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017. 22

A Additional results

Here we provide additional ablation study results. Note that results on the same method may be slightly different due to different ways of setting random seeds in the scripts.

A.1 Analysis of Topological Components and Backbones

Table 3 ablates the two-branch TopoSIGN design under the signed link-prediction pre-training and prompt-learning clustering pipeline. The topology-only variant yields near-zero ARI, showing that persistent-homology features alone cannot replace a signed graph encoder. For the MSGNN backbone, TopoMSGNN improves over MSGNN-only on four of the five datasets and generally outperforms the positive-sign subgraph-only topological baseline, indicating that the signed degree-vector filtration provides useful complementary information and that negative edges should not be discarded before topological feature construction. For the SSSNET backbone, the effect is more mixed: TopoSSSNET improves on SDSBM-2 but underperforms SSSNET-only on several other datasets, suggesting that topological augmentation is backbone-dependent and may interact differently with signed aggregation mechanisms. Overall, the ablation confirms that the structural encoder is essential, the topology representation learning with signed filtration is generally preferable to positive-only topology, and the gain from topology depends on how well the topological features complement the chosen signed graph encoder.

In addition, we are interested in how different pre-training tasks may affect the quality of TopoSIGN's embeddings useful for node clustering. Table 4 provides a comparison of conducting one additional link prediction task. For signed directed graphs, we additionally consider multi-task variants in our ablation study

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{SP}} + \sum_{r \in \mathcal{R}} \alpha_r \mathcal{L}_r, \tag{18}$$

where $\mathcal{R}$ may include DP, 3C, 4C, and 5C relation-prediction objectives and α_r are hyperparameters (here we set them to be either 0 or 1). These auxiliary tasks test whether directional or non-edge discrimination improves downstream clustering transfer. In addition to our default task SP, in our second task, direction prediction (DP), one aims to predict whether $(u, v) \in \mathcal{E}$ or $(v, u) \in \mathcal{E}$ under

Table 3: Ablation study: effect of different topological filtration strategies on node clustering ARI on TopoMSGNN and TopoSSSNET. posTopo means the positive-subgraph only baseline filtration. MSGNN-, SSSNET-, or Topo- only means leaving out one branch in the framework.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall
MSGNN_only	0.014 ± 0.011	$\underline{0.102 \pm 0.021}$	0.372 ± 0.044	$\underline{0.164 \pm 0.019}$	$\underline{0.068 \pm 0.058}$
Topo_only	-0.001 ± 0.001	0.003 ± 0.007	0.007 ± 0.008	-0.000 ± 0.000	0.001 ± 0.001
MSGNN+posTopo	0.006 ± 0.006	0.099 ± 0.016	0.358 ± 0.049	0.164 ± 0.034	0.034 ± 0.046
TopoMSGNN	0.037 ± 0.026	0.092 ± 0.018	$\underline{0.381 \pm 0.046}$	$\mathbf{0.167 \pm 0.024}$	$\mathbf{0.076 \pm 0.063}$
SSSNET_only	$\mathbf{0.096 \pm 0.046}$	0.099 ± 0.025	$\mathbf{0.394 \pm 0.019}$	0.162 ± 0.023	0.054 ± 0.055
SSSNET+posTopo	0.057 ± 0.016	0.094 ± 0.028	0.365 ± 0.028	0.148 ± 0.021	0.056 ± 0.052
TopoSSSNET	$\underline{0.066 \pm 0.039}$	$\mathbf{0.118 \pm 0.020}$	0.374 ± 0.017	0.146 ± 0.028	0.025 ± 0.049

the assumption that exactly one of these two conditions holds. We also consider three-, four-, and five-class prediction problems. In the three-class problem (3C), the possibilities are $(u, v) \in \mathcal{E}$, $(v, u) \in \mathcal{E}$, or that neither (u, v) nor (v, u) are in $\mathcal{E}$. For the four-class problem (4C), the possibilities are $(u, v) \in \mathcal{E}^+$, $(u, v) \in \mathcal{E}^-$, $(v, u) \in \mathcal{E}^+$, and $(v, u) \in \mathcal{E}^-$. For the five-class problem (5C), we also add in the possibility that neither (u, v) nor (v, u) are in $\mathcal{E}$. In this ablation study, we test SP only against SP plus one of the other tasks. We can see from Table 4 that adding the pre-training task 3C is generally the most helpful for TopoMSGNN by also considering a class of non-existing edges to further learn about the graph structure. On the other hand, SP only seems to perform the second-best overall. On TopoSSSNET, however, different additions to SP may have different leading datasets, and having an extra task does boost the TopoSSSNET performance. For simplicity and due to the unclearness of which extra task is the most helpful, we stick to SP only for our main results.

Table 4: Pre-training task ablation: effect of different pre-training objectives on downstream node clustering ARI after prompt learning for TopoMSGNN. We report the mean ARI among five runs plus/minus one standard deviation. The best method is marked in **bold** while the second best is marked with underline.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4
TopoMSGNN_pretrain_SP	0.013 ± 0.025	0.100 ± 0.015	0.400 ± 0.025	0.164 ± 0.018
TopoMSGNN_pretrain_SP+DP	0.026 ± 0.006	0.090 ± 0.026	0.340 ± 0.080	0.029 ± 0.028
TopoMSGNN_pretrain_SP+3C	0.031 ± 0.024	0.115 ± 0.015	0.396 ± 0.043	0.117 ± 0.035
TopoMSGNN_pretrain_SP+4C	0.011 ± 0.010	0.113 ± 0.014	0.389 ± 0.066	0.074 ± 0.043
TopoMSGNN_pretrain_SP+5C	0.013 ± 0.014	0.107 ± 0.011	0.393 ± 0.040	0.072 ± 0.017
TopoSSSNET_pretrain_SP	0.066 ± 0.039	0.118 ± 0.020	0.374 ± 0.017	0.146 ± 0.028
TopoSSSNET_pretrain_SP+DP	$\mathbf{0.082 \pm 0.017}$	$\underline{0.129 \pm 0.004}$	0.389 ± 0.060	0.064 ± 0.026
TopoSSSNET_pretrain_SP+3C	$\underline{0.072 \pm 0.038}$	0.126 ± 0.020	$\mathbf{0.431 \pm 0.023}$	0.189 ± 0.150
TopoSSSNET_pretrain_SP+4C	0.060 ± 0.048	$\mathbf{0.132 \pm 0.015}$	0.399 ± 0.090	$\underline{0.195 \pm 0.118}$
TopoSSSNET_pretrain_SP+5C	0.068 ± 0.025	0.125 ± 0.016	$\underline{0.430 \pm 0.039}$	$\mathbf{0.205 \pm 0.101}$

A.2 Evaluation on Graph Prompting Paradigm

To further investigate the compatibility of our signed topological features with the emerging graph prompting paradigm, we evaluate four representative downstream prompting methods: GPPT, Gprompt, GPF, and All-in-One. In this experiment, the SGE+Topo part is employed as a fixed backbone to provide structural embeddings, and we compare how different prompt designs affect the final node clustering results. Comparing Table 5 and Table 1, we conclude that using our SGE+Topo indeed boosts the prompt learning frameworks in the signed clustering task.

A.3 Extended Ablation Studies and Few-Shot Experiments

To investigate whether our topological filtration strategies universally enhance baseline performance, we conduct comprehensive ablation studies across all combinations of backbones and downstream prompting frameworks, evaluating models both with and without topological information. We first evaluate overall performance under standard settings, with results reported in Tables 5–7. Furthermore, to assess whether topological features remain effective under data-scarce scenarios, we

conduct few-shot ablation experiments under 1-, 3-, and 5-shot settings. The detailed few-shot results are summarized in Tables 8–16.

Table 5: Prompt method comparison: performance of TopoMSGNN, TopoSSSNET, TopoDSGC and TopoSigMaNet combined with different graph prompting strategies. We report the mean ARI among five runs plus/minus one standard deviation. The best method is marked in **bold** while the second best is marked with underline.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+Topo+GPPT	0.032 ± 0.029	0.086 ± 0.032	0.399 ± 0.041	0.233 ± 0.131	0.079 ± 0.065	0.005 ± 0.010
MSGNN+Topo+Gprompt	0.670 ± 0.038	0.680 ± 0.031	0.768 ± 0.047	0.641 ± 0.013	0.129 ± 0.033	0.095 ± 0.012
MSGNN+Topo+GPF	0.679 ± 0.041	0.671 ± 0.042	0.767 ± 0.028	$\mathbf{0.876 \pm 0.153}$	0.142 ± 0.023	0.103 ± 0.016
MSGNN+Topo+All-in-one	0.658 ± 0.057	0.660 ± 0.032	0.748 ± 0.050	0.801 ± 0.157	0.159 ± 0.027	0.098 ± 0.006
SSSNET+Topo+GPPT	0.087 ± 0.029	0.139 ± 0.019	0.375 ± 0.042	0.156 ± 0.017	0.055 ± 0.055	0.000 ± 0.000
SSSNET+Topo+Gprompt	0.665 ± 0.057	0.792 ± 0.057	0.803 ± 0.070	0.651 ± 0.000	0.145 ± 0.057	0.108 ± 0.019
SSSNET+Topo+GPF	0.707 ± 0.040	0.754 ± 0.045	0.795 ± 0.048	0.720 ± 0.107	0.143 ± 0.025	$\mathbf{0.119 \pm 0.030}$
SSSNET+Topo+All-in-one	0.648 ± 0.034	0.774 ± 0.055	0.818 ± 0.056	0.706 ± 0.119	0.161 ± 0.039	0.117 ± 0.025
DSGC+Topo+GPPT	0.042 ± 0.037	0.104 ± 0.022	0.261 ± 0.051	0.154 ± 0.017	0.025 ± 0.049	0.000 ± 0.000
DSGC+Topo+Gprompt	0.950 ± 0.010	$\mathbf{0.826 \pm 0.035}$	0.897 ± 0.075	0.779 ± 0.162	0.161 ± 0.029	0.091 ± 0.008
DSGC+Topo+GPF	0.937 ± 0.021	0.767 ± 0.086	$\mathbf{0.911 \pm 0.043}$	0.845 ± 0.186	0.173 ± 0.014	0.093 ± 0.004
DSGC+Topo+All-in-one	$\mathbf{0.957 \pm 0.009}$	0.785 ± 0.018	0.892 ± 0.064	0.790 ± 0.172	$\mathbf{0.175 \pm 0.030}$	0.093 ± 0.009
SigMaNet+Topo+GPPT	-0.001 ± 0.002	0.012 ± 0.007	0.079 ± 0.037	0.204 ± 0.135	0.033 ± 0.019	0.006 ± 0.004
SigMaNet+Topo+Gprompt	0.056 ± 0.026	0.050 ± 0.038	0.094 ± 0.050	0.162 ± 0.091	0.104 ± 0.044	0.030 ± 0.017
SigMaNet+Topo+GPF	0.090 ± 0.059	0.133 ± 0.033	0.143 ± 0.039	0.214 ± 0.114	0.134 ± 0.030	0.051 ± 0.004
SigMaNet+Topo+All-in-one	0.077 ± 0.061	0.118 ± 0.030	0.141 ± 0.033	0.294 ± 0.020	0.123 ± 0.040	0.036 ± 0.011

Table 6: Prompt method comparison: performance of MSGNN, SSSNET, DSGC and SigMaNet combined with different graph prompting strategies. We report the mean ARI among five runs plus/minus one standard deviation. The best method is marked in **bold** while the second best is marked with underline.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+GPPT	0.012 ± 0.013	0.104 ± 0.016	0.368 ± 0.048	0.157 ± 0.016	0.073 ± 0.060	0.010 ± 0.012
MSGNN+Gprompt	0.640 ± 0.099	0.643 ± 0.117	0.764 ± 0.045	0.768 ± 0.132	0.151 ± 0.023	0.105 ± 0.008
MSGNN+GPF	0.676 ± 0.016	0.688 ± 0.019	0.756 ± 0.022	0.758 ± 0.134	0.161 ± 0.027	0.103 ± 0.015
MSGNN+All-in-one	0.660 ± 0.032	0.705 ± 0.016	0.787 ± 0.022	0.787 ± 0.167	0.169 ± 0.022	0.107 ± 0.007
SSSNET+GPPT	0.057 ± 0.026	0.120 ± 0.017	0.382 ± 0.041	0.164 ± 0.018	0.053 ± 0.065	0.000 ± 0.000
SSSNET+Gprompt	0.654 ± 0.023	0.779 ± 0.065	0.925 ± 0.049	0.716 ± 0.130	0.126 ± 0.027	0.096 ± 0.015
SSSNET+GPF	0.694 ± 0.054	0.773 ± 0.023	0.899 ± 0.074	0.749 ± 0.130	0.148 ± 0.020	0.102 ± 0.012
SSSNET+All-in-one	0.710 ± 0.043	0.775 ± 0.041	0.867 ± 0.054	0.805 ± 0.143	$\mathbf{0.178 \pm 0.051}$	$\mathbf{0.115 \pm 0.028}$
DSGC+GPPT	0.057 ± 0.029	0.106 ± 0.007	0.291 ± 0.062	0.210 ± 0.155	0.026 ± 0.053	0.000 ± 0.000
DSGC+Gprompt	0.927 ± 0.049	$\mathbf{0.866 \pm 0.041}$	$\mathbf{0.957 \pm 0.010}$	0.791 ± 0.171	0.164 ± 0.046	0.096 ± 0.006
DSGC+GPF	0.909 ± 0.035	0.828 ± 0.038	0.948 ± 0.023	0.858 ± 0.169	0.170 ± 0.031	0.084 ± 0.013
DSGC+All-in-one	$\mathbf{0.941 \pm 0.023}$	0.857 ± 0.032	0.956 ± 0.010	$\mathbf{0.860 \pm 0.171}$	0.155 ± 0.025	0.089 ± 0.007
SigMaNet+GPPT	-0.001 ± 0.004	0.006 ± 0.009	0.028 ± 0.051	0.082 ± 0.165	0.060 ± 0.121	0.001 ± 0.001
SigMaNet+Gprompt	0.114 ± 0.061	0.044 ± 0.088	0.186 ± 0.130	0.274 ± 0.357	0.107 ± 0.148	0.018 ± 0.028
SigMaNet+GPF	0.097 ± 0.101	0.157 ± 0.124	0.233 ± 0.131	0.664 ± 0.118	-0.000 ± 0.000	0.043 ± 0.035
SigMaNet+All-in-one	0.044 ± 0.061	0.093 ± 0.115	0.222 ± 0.119	0.466 ± 0.206	0.064 ± 0.079	0.033 ± 0.039

Table 7: Ablation study: performance improvement by adding topological information (Table 5 minus Table 6). Positive values indicate performance gains due to topological structures, while negative values show performance drops.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+GPPT	$+0.020$	-0.018	$+0.031$	$+0.076$	$+0.006$	-0.005
MSGNN+Gprompt	$+0.030$	$+0.037$	$+0.004$	-0.127	-0.022	-0.010
MSGNN+GPF	$+0.003$	-0.017	$+0.011$	$+0.118$	-0.019	0.000
MSGNN+All-in-one	-0.002	-0.045	-0.039	$+0.014$	-0.010	-0.009
SSSNET+GPPT	$+0.030$	$+0.019$	-0.007	-0.008	$+0.002$	0.000
SSSNET+Gprompt	$+0.011$	$+0.013$	-0.122	-0.065	$+0.019$	$+0.012$
SSSNET+GPF	$+0.013$	-0.019	-0.104	-0.029	-0.005	$+0.017$
SSSNET+All-in-one	-0.062	-0.001	-0.049	-0.099	-0.017	$+0.002$
DSGC+GPPT	-0.015	-0.002	-0.030	-0.056	-0.001	0.000
DSGC+Gprompt	$+0.023$	-0.040	-0.060	-0.012	-0.003	-0.005
DSGC+GPF	$+0.028$	-0.061	-0.037	-0.013	$+0.003$	$+0.009$
DSGC+All-in-one	$+0.016$	-0.072	-0.064	-0.070	$+0.020$	$+0.004$
SigMaNet+GPPT	0.000	$+0.006$	$+0.051$	$+0.122$	-0.027	$+0.005$
SigMaNet+Gprompt	-0.058	$+0.006$	-0.092	-0.112	-0.003	$+0.012$
SigMaNet+GPF	-0.007	-0.024	-0.090	-0.450	$+0.134$	$+0.008$
SigMaNet+All-in-one	$+0.033$	$+0.025$	-0.081	-0.172	$+0.059$	$+0.003$

Table 8: Prompt method comparison with topological information (1-shot with topo). We report the mean ARI among five runs plus/minus one standard deviation. The best method is marked in **bold** while the second best is marked with underline.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
TopoMSGNN+GPPT	0.024 ± 0.036	0.092 ± 0.038	0.334 ± 0.055	0.158 ± 0.034	0.137 ± 0.010	0.088 ± 0.022
TopoMSGNN+Gprompt	0.513 ± 0.067	0.572 ± 0.078	0.593 ± 0.159	0.858 ± 0.168	0.170 ± 0.045	0.107 ± 0.011
TopoMSGNN+GPF	0.504 ± 0.098	0.545 ± 0.056	0.620 ± 0.125	$\mathbf{0.967 \pm 0.024}$	0.194 ± 0.016	0.113 ± 0.011
TopoMSGNN+All-in-one	0.563 ± 0.087	0.541 ± 0.052	0.605 ± 0.158	0.835 ± 0.153	0.171 ± 0.014	0.113 ± 0.012
TopoSSSNET+GPPT	0.041 ± 0.012	0.101 ± 0.012	0.353 ± 0.038	0.160 ± 0.025	0.124 ± 0.022	0.069 ± 0.017
TopoSSSNET+Gprompt	0.667 ± 0.024	$\mathbf{0.755 \pm 0.042}$	0.799 ± 0.043	0.903 ± 0.129	$\underline{0.211 \pm 0.031}$	$\mathbf{0.118 \pm 0.021}$
TopoSSSNET+GPF	0.681 ± 0.034	0.738 ± 0.016	0.820 ± 0.063	$\underline{0.907 \pm 0.130}$	$\mathbf{0.216 \pm 0.018}$	0.106 ± 0.011
TopoSSSNET+All-in-one	0.670 ± 0.035	$\underline{0.742 \pm 0.049}$	0.817 ± 0.045	0.892 ± 0.124	0.168 ± 0.036	$\underline{0.115 \pm 0.010}$
TopoDSGC+GPPT	0.007 ± 0.011	0.085 ± 0.022	0.273 ± 0.054	0.156 ± 0.022	0.124 ± 0.028	0.082 ± 0.003
TopoDSGC+Gprompt	$\underline{0.935 \pm 0.015}$	0.668 ± 0.121	$\underline{0.850 \pm 0.073}$	0.835 ± 0.147	0.180 ± 0.024	0.100 ± 0.008
TopoDSGC+GPF	0.933 ± 0.015	0.660 ± 0.075	0.822 ± 0.102	0.860 ± 0.156	0.189 ± 0.016	0.096 ± 0.006
TopoDSGC+All-in-one	$\mathbf{0.944 \pm 0.017}$	0.701 ± 0.153	$\mathbf{0.850 \pm 0.082}$	0.853 ± 0.127	0.168 ± 0.013	0.090 ± 0.008
TopoSigMaNet+GPPT	-0.001 ± 0.003	0.001 ± 0.002	0.084 ± 0.054	0.215 ± 0.106	0.040 ± 0.031	0.009 ± 0.008
TopoSigMaNet+Gprompt	0.076 ± 0.013	0.129 ± 0.024	0.128 ± 0.031	0.235 ± 0.125	0.133 ± 0.022	0.030 ± 0.017
TopoSigMaNet+GPF	0.101 ± 0.038	0.136 ± 0.030	0.152 ± 0.025	0.329 ± 0.023	0.137 ± 0.020	0.034 ± 0.019
TopoSigMaNet+All-in-one	0.112 ± 0.032	0.112 ± 0.054	0.128 ± 0.032	0.300 ± 0.080	0.129 ± 0.021	0.035 ± 0.019

Table 9: Prompt method comparison without topological information (1-shot without topo).

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+GPPT	0.016 ± 0.009	0.081 ± 0.037	0.321 ± 0.053	0.166 ± 0.029	0.112 ± 0.057	0.084 ± 0.005
MSGNN+Gprompt	0.517 ± 0.084	0.547 ± 0.071	0.570 ± 0.134	$\underline{0.952 \pm 0.051}$	0.176 ± 0.017	$\underline{0.113 \pm 0.009}$
MSGNN+GPF	0.533 ± 0.085	0.534 ± 0.060	0.641 ± 0.081	0.948 ± 0.045	0.172 ± 0.023	0.113 ± 0.016
MSGNN+All-in-one	0.498 ± 0.026	0.564 ± 0.057	0.602 ± 0.068	$\mathbf{0.980 \pm 0.014}$	0.178 ± 0.014	0.108 ± 0.009
SSSNET+GPPT	0.035 ± 0.018	0.089 ± 0.034	0.361 ± 0.049	0.172 ± 0.019	0.134 ± 0.015	0.068 ± 0.008
SSSNET+Gprompt	0.648 ± 0.036	0.729 ± 0.030	0.791 ± 0.065	0.899 ± 0.129	$\mathbf{0.226 \pm 0.040}$	0.112 ± 0.008
SSSNET+GPF	0.721 ± 0.056	$\underline{0.755 \pm 0.064}$	0.814 ± 0.080	0.909 ± 0.131	0.194 ± 0.028	0.104 ± 0.009
SSSNET+All-in-one	0.648 ± 0.027	0.711 ± 0.062	0.797 ± 0.038	0.937 ± 0.099	0.180 ± 0.012	$\mathbf{0.116 \pm 0.016}$
DSGC+GPPT	0.024 ± 0.014	0.109 ± 0.015	0.329 ± 0.047	0.202 ± 0.159	0.127 ± 0.018	0.062 ± 0.029
DSGC+Gprompt	$\underline{0.916 \pm 0.032}$	0.748 ± 0.136	$\mathbf{0.930 \pm 0.024}$	0.861 ± 0.170	$\underline{0.204 \pm 0.015}$	0.100 ± 0.006
DSGC+GPF	$\mathbf{0.920 \pm 0.031}$	0.724 ± 0.078	$\underline{0.912 \pm 0.035}$	0.861 ± 0.170	0.195 ± 0.023	0.095 ± 0.003
DSGC+All-in-one	0.881 ± 0.031	$\mathbf{0.781 \pm 0.045}$	0.883 ± 0.070	0.860 ± 0.163	0.183 ± 0.027	0.094 ± 0.008
SigMaNet+GPPT	-0.001 ± 0.002	0.009 ± 0.013	0.005 ± 0.010	0.000 ± 0.000	0.021 ± 0.043	0.015 ± 0.024
SigMaNet+Gprompt	0.128 ± 0.092	0.044 ± 0.081	0.188 ± 0.155	0.557 ± 0.101	0.033 ± 0.067	0.029 ± 0.035
SigMaNet+GPF	0.095 ± 0.093	0.108 ± 0.132	0.081 ± 0.118	0.636 ± 0.115	0.000 ± 0.000	0.040 ± 0.040
SigMaNet+All-in-one	0.160 ± 0.094	0.049 ± 0.090	0.201 ± 0.166	0.628 ± 0.186	0.096 ± 0.128	0.030 ± 0.037

Table 10: Ablation study on 1-shot learning: performance improvement by adding topological information (1-shot with Topo minus 1-shot without Topo). Positive values indicate gains from topology, while negative values indicate performance drops.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+GPPT	$+0.008$	$+0.011$	$+0.013$	-0.008	$+0.025$	$+0.004$
MSGNN+Gprompt	-0.004	$+0.025$	$+0.023$	-0.094	-0.006	-0.006
MSGNN+GPF	-0.029	$+0.011$	-0.021	$+0.019$	$+0.022$	0.000
MSGNN+All-in-one	$+0.065$	-0.023	$+0.003$	-0.145	-0.007	$+0.005$
SSSNET+GPPT	$+0.006$	$+0.012$	-0.008	-0.012	-0.010	$+0.001$
SSSNET+Gprompt	$+0.019$	$+0.026$	$+0.008$	$+0.004$	-0.015	$+0.006$
SSSNET+GPF	-0.040	-0.017	$+0.006$	-0.002	$+0.022$	$+0.002$
SSSNET+All-in-one	$+0.022$	$+0.031$	$+0.020$	-0.045	-0.012	-0.001
DSGC+GPPT	-0.015	-0.024	-0.056	-0.046	-0.003	$+0.020$
DSGC+Gprompt	$+0.019$	-0.080	-0.080	-0.026	-0.024	0.000
DSGC+GPF	$+0.013$	-0.064	-0.090	-0.001	-0.006	$+0.001$
DSGC+All-in-one	$+0.063$	-0.080	-0.033	-0.007	-0.015	-0.004
SigMaNet+GPPT	0.000	-0.008	$+0.079$	$+0.215$	$+0.019$	-0.006
SigMaNet+Gprompt	-0.052	$+0.085$	-0.060	-0.322	$+0.100$	$+0.001$
SigMaNet+GPF	$+0.006$	$+0.028$	$+0.071$	-0.307	$+0.137$	-0.006
SigMaNet+All-in-one	-0.048	$+0.063$	-0.073	-0.328	$+0.033$	$+0.005$

Table 11: Prompt method comparison with topological information (3-shot with topo). We report the mean ARI among five runs plus/minus one standard deviation. The best method is marked in **bold** while the second best is marked with underline.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
TopoMSGNN+GPPT	0.030 ± 0.023	0.081 ± 0.017	0.384 ± 0.034	0.165 ± 0.016	0.130 ± 0.069	0.099 ± 0.007
TopoMSGNN+Gprompt	0.543 ± 0.059	0.548 ± 0.063	0.679 ± 0.082	0.848 ± 0.161	0.182 ± 0.007	0.113 ± 0.006
TopoMSGNN+GPF	0.539 ± 0.092	0.528 ± 0.027	0.726 ± 0.039	$\mathbf{0.954 \pm 0.042}$	0.178 ± 0.036	0.109 ± 0.014
TopoMSGNN+All-in-one	0.562 ± 0.061	0.532 ± 0.021	0.663 ± 0.104	0.842 ± 0.157	0.186 ± 0.006	0.109 ± 0.012
TopoSSSNET+GPPT	0.065 ± 0.050	0.118 ± 0.024	0.385 ± 0.045	0.191 ± 0.043	0.138 ± 0.014	0.067 ± 0.011
TopoSSSNET+Gprompt	0.671 ± 0.027	$\underline{0.798 \pm 0.042}$	0.827 ± 0.064	0.941 ± 0.029	0.194 ± 0.019	0.113 ± 0.006
TopoSSSNET+GPF	0.679 ± 0.035	$\mathbf{0.799 \pm 0.030}$	0.817 ± 0.048	0.950 ± 0.029	$\underline{0.219 \pm 0.057}$	$\underline{0.117 \pm 0.010}$
TopoSSSNET+All-in-one	0.679 ± 0.031	0.770 ± 0.036	0.854 ± 0.035	$\underline{0.950 \pm 0.025}$	$\mathbf{0.230 \pm 0.037}$	$\mathbf{0.126 \pm 0.032}$
TopoDSGC+GPPT	0.020 ± 0.021	0.106 ± 0.024	0.271 ± 0.092	0.153 ± 0.026	0.124 ± 0.028	0.083 ± 0.002
TopoDSGC+Gprompt	0.926 ± 0.017	0.762 ± 0.030	$\underline{0.870 \pm 0.069}$	0.899 ± 0.128	0.183 ± 0.025	0.095 ± 0.006
TopoDSGC+GPF	$\mathbf{0.934 \pm 0.024}$	0.723 ± 0.079	0.868 ± 0.055	0.857 ± 0.155	0.181 ± 0.012	0.095 ± 0.012
TopoDSGC+All-in-one	$\underline{0.929 \pm 0.027}$	0.724 ± 0.042	$\mathbf{0.874 \pm 0.082}$	0.886 ± 0.125	0.179 ± 0.021	0.089 ± 0.009
TopoSigMaNet+GPPT	-0.004 ± 0.003	0.007 ± 0.006	0.116 ± 0.049	0.182 ± 0.078	0.028 ± 0.022	0.003 ± 0.006
TopoSigMaNet+Gprompt	0.086 ± 0.039	0.140 ± 0.035	0.149 ± 0.017	0.243 ± 0.140	0.126 ± 0.035	0.055 ± 0.004
TopoSigMaNet+GPF	0.135 ± 0.033	0.127 ± 0.031	0.162 ± 0.029	0.346 ± 0.027	0.147 ± 0.017	0.051 ± 0.004
TopoSigMaNet+All-in-one	0.117 ± 0.035	0.139 ± 0.028	0.148 ± 0.020	0.321 ± 0.044	0.116 ± 0.032	0.052 ± 0.004

Table 12: Prompt method comparison without topological information (3-shot without topo).

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+GPPT	0.038 ± 0.031	0.091 ± 0.032	0.360 ± 0.039	0.163 ± 0.028	0.112 ± 0.059	0.087 ± 0.007
MSGNN+Gprompt	0.550 ± 0.115	0.560 ± 0.051	0.719 ± 0.063	0.921 ± 0.135	0.174 ± 0.005	0.113 ± 0.006
MSGNN+GPF	0.578 ± 0.045	0.513 ± 0.085	0.767 ± 0.033	0.946 ± 0.059	0.173 ± 0.019	0.106 ± 0.006
MSGNN+All-in-one	0.602 ± 0.072	0.567 ± 0.076	0.704 ± 0.064	$\mathbf{0.967 \pm 0.040}$	0.159 ± 0.021	0.105 ± 0.008
SSSNET+GPPT	0.052 ± 0.021	0.089 ± 0.045	0.365 ± 0.052	0.169 ± 0.019	0.132 ± 0.016	0.050 ± 0.028
SSSNET+Gprompt	0.672 ± 0.039	0.768 ± 0.062	0.811 ± 0.059	0.901 ± 0.133	$\mathbf{0.220 \pm 0.007}$	$\underline{0.126 \pm 0.020}$
SSSNET+GPF	0.724 ± 0.056	0.747 ± 0.048	0.853 ± 0.063	0.926 ± 0.127	$\underline{0.208 \pm 0.020}$	0.118 ± 0.012
SSSNET+All-in-one	0.647 ± 0.060	0.773 ± 0.055	0.796 ± 0.098	$\underline{0.950 \pm 0.060}$	0.201 ± 0.035	$\mathbf{0.128 \pm 0.016}$
DSGC+GPPT	0.033 ± 0.021	0.092 ± 0.018	0.344 ± 0.026	0.217 ± 0.119	0.149 ± 0.022	0.061 ± 0.032
DSGC+Gprompt	$\mathbf{0.928 \pm 0.029}$	$\mathbf{0.819 \pm 0.021}$	0.894 ± 0.092	0.861 ± 0.170	0.190 ± 0.008	0.098 ± 0.006
DSGC+GPF	0.914 ± 0.031	0.756 ± 0.034	$\underline{0.922 \pm 0.046}$	0.856 ± 0.167	0.189 ± 0.028	0.097 ± 0.005
DSGC+All-in-one	$\underline{0.922 \pm 0.020}$	$\underline{0.807 \pm 0.040}$	$\mathbf{0.934 \pm 0.036}$	0.863 ± 0.167	0.190 ± 0.016	0.094 ± 0.006
SigMaNet+GPPT	-0.001 ± 0.002	0.006 ± 0.007	0.008 ± 0.011	0.148 ± 0.180	0.003 ± 0.007	0.001 ± 0.003
SigMaNet+Gprompt	0.134 ± 0.081	0.048 ± 0.086	0.191 ± 0.157	0.663 ± 0.147	0.026 ± 0.052	0.023 ± 0.029
SigMaNet+GPF	0.094 ± 0.091	0.098 ± 0.120	0.080 ± 0.119	0.663 ± 0.115	0.000 ± 0.000	0.039 ± 0.038
SigMaNet+All-in-one	0.125 ± 0.079	0.046 ± 0.087	0.244 ± 0.228	0.603 ± 0.152	0.079 ± 0.104	0.031 ± 0.038

Table 13: Ablation study on 3-shot learning: performance improvement by adding topological information (3-shot with Topo minus 3-shot without Topo). Positive values represent gains from adding topological information.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+GPPT	-0.008	-0.010	$+0.024$	$+0.002$	$+0.018$	$+0.012$
MSGNN+Gprompt	-0.007	-0.012	-0.040	-0.073	$+0.008$	0.000
MSGNN+GPF	-0.039	$+0.015$	-0.041	$+0.008$	$+0.005$	$+0.003$
MSGNN+All-in-one	-0.040	-0.035	-0.041	-0.125	$+0.027$	$+0.004$
SSSNET+GPPT	$+0.013$	$+0.029$	$+0.020$	$+0.022$	$+0.006$	$+0.017$
SSSNET+Gprompt	-0.001	$+0.030$	$+0.016$	$+0.040$	-0.026	-0.013
SSSNET+GPF	-0.045	$+0.052$	-0.036	$+0.024$	$+0.011$	-0.001
SSSNET+All-in-one	$+0.032$	-0.003	$+0.058$	0.000	$+0.029$	-0.002
DSGC+GPPT	-0.013	$+0.014$	-0.073	-0.064	-0.025	$+0.022$
DSGC+Gprompt	-0.002	-0.057	-0.024	$+0.038$	-0.007	-0.003
DSGC+GPF	$+0.020$	-0.033	-0.054	$+0.001$	-0.008	-0.002
DSGC+All-in-one	$+0.007$	-0.083	-0.060	$+0.023$	-0.011	-0.005
SigMaNet+GPPT	-0.003	$+0.001$	$+0.108$	$+0.034$	$+0.025$	$+0.002$
SigMaNet+Gprompt	-0.048	$+0.092$	-0.042	-0.420	$+0.100$	$+0.032$
SigMaNet+GPF	$+0.041$	$+0.029$	$+0.082$	-0.317	$+0.147$	$+0.012$
SigMaNet+All-in-one	-0.008	$+0.093$	-0.096	-0.282	$+0.037$	$+0.021$

Table 14: Prompt method comparison with topological information (5-shot with topo). We report the mean ARI among five runs plus/minus one standard deviation. The best method is marked in **bold** while the second best is marked with underline.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
TopoMSGNN+GPPT	0.036 ± 0.020	0.068 ± 0.023	0.349 ± 0.075	0.160 ± 0.024	0.143 ± 0.018	0.099 ± 0.007
TopoMSGNN+Gprompt	0.603 ± 0.075	0.623 ± 0.041	0.691 ± 0.076	0.857 ± 0.167	0.177 ± 0.024	0.108 ± 0.011
TopoMSGNN+GPF	0.575 ± 0.084	0.568 ± 0.046	0.735 ± 0.044	$\mathbf{0.970 \pm 0.037}$	0.201 ± 0.017	0.108 ± 0.006
TopoMSGNN+All-in-one	0.615 ± 0.076	0.590 ± 0.071	0.687 ± 0.077	0.855 ± 0.165	0.170 ± 0.017	0.102 ± 0.007
TopoSSSNET+GPPT	0.091 ± 0.058	0.122 ± 0.032	0.381 ± 0.071	0.151 ± 0.025	0.130 ± 0.024	0.067 ± 0.031
TopoSSSNET+Gprompt	0.664 ± 0.031	0.772 ± 0.046	0.855 ± 0.044	0.913 ± 0.100	0.211 ± 0.028	$\mathbf{0.114 \pm 0.006}$
TopoSSSNET+GPF	0.709 ± 0.062	$\mathbf{0.785 \pm 0.037}$	0.836 ± 0.052	0.881 ± 0.120	0.217 ± 0.011	0.096 ± 0.015
TopoSSSNET+All-in-one	0.680 ± 0.042	0.775 ± 0.045	0.836 ± 0.047	0.954 ± 0.023	$\mathbf{0.223 \pm 0.030}$	$\mathbf{0.114 \pm 0.009}$
TopoDSGC+GPPT	0.029 ± 0.014	0.072 ± 0.041	0.202 ± 0.104	0.155 ± 0.019	0.081 ± 0.060	0.048 ± 0.039
TopoDSGC+Gprompt	0.914 ± 0.034	0.779 ± 0.037	0.879 ± 0.072	0.825 ± 0.146	0.176 ± 0.027	0.100 ± 0.006
TopoDSGC+All-in-one	0.922 ± 0.014	0.780 ± 0.056	$\mathbf{0.909 \pm 0.066}$	0.843 ± 0.158	0.168 ± 0.035	0.096 ± 0.003
TopoDSGC+GPF	$\mathbf{0.938 \pm 0.017}$	0.737 ± 0.111	0.907 ± 0.038	0.853 ± 0.169	0.182 ± 0.026	0.101 ± 0.002
TopoSigMaNet+GPPT	-0.001 ± 0.002	0.008 ± 0.004	0.111 ± 0.037	0.231 ± 0.093	0.039 ± 0.055	0.008 ± 0.007
TopoSigMaNet+Gprompt	0.069 ± 0.020	0.122 ± 0.044	0.136 ± 0.052	0.216 ± 0.079	0.116 ± 0.030	0.053 ± 0.004
TopoSigMaNet+GPF	0.118 ± 0.039	0.134 ± 0.026	0.161 ± 0.041	0.283 ± 0.049	0.118 ± 0.037	0.042 ± 0.017
TopoSigMaNet+All-in-one	0.114 ± 0.026	0.123 ± 0.024	0.150 ± 0.029	0.301 ± 0.048	0.140 ± 0.024	0.042 ± 0.017

Table 15: Prompt method comparison without topological information (5-shot without topo).

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+GPPT	0.030 ± 0.026	0.066 ± 0.024	0.363 ± 0.025	0.158 ± 0.025	0.140 ± 0.018	0.082 ± 0.012
MSGNN+Gprompt	0.596 ± 0.056	0.622 ± 0.033	0.715 ± 0.041	0.855 ± 0.166	0.163 ± 0.018	0.110 ± 0.005
MSGNN+GPF	0.575 ± 0.066	0.594 ± 0.055	0.773 ± 0.016	0.954 ± 0.045	0.170 ± 0.017	0.107 ± 0.006
MSGNN+All-in-one	0.579 ± 0.039	0.645 ± 0.035	0.724 ± 0.024	$\mathbf{0.960 \pm 0.034}$	0.143 ± 0.041	0.110 ± 0.007
SSSNET+GPPT	0.055 ± 0.016	0.099 ± 0.012	0.379 ± 0.065	0.163 ± 0.018	0.121 ± 0.043	0.067 ± 0.012
SSSNET+Gprompt	0.693 ± 0.035	0.773 ± 0.059	0.844 ± 0.037	0.891 ± 0.131	0.217 ± 0.039	0.104 ± 0.012
SSSNET+GPF	0.734 ± 0.056	0.798 ± 0.028	0.858 ± 0.058	0.921 ± 0.126	0.229 ± 0.015	0.107 ± 0.015
SSSNET+All-in-one	0.665 ± 0.035	0.760 ± 0.061	0.833 ± 0.043	0.943 ± 0.086	$\mathbf{0.240 \pm 0.011}$	$\mathbf{0.125 \pm 0.024}$
DSGC+GPPT	0.042 ± 0.019	0.110 ± 0.029	0.318 ± 0.040	0.232 ± 0.136	0.102 ± 0.051	0.053 ± 0.037
DSGC+Gprompt	$\mathbf{0.947 \pm 0.018}$	$\mathbf{0.849 \pm 0.026}$	0.926 ± 0.039	0.861 ± 0.170	0.180 ± 0.015	0.095 ± 0.006
DSGC+All-in-one	0.928 ± 0.021	0.840 ± 0.034	$\mathbf{0.953 \pm 0.018}$	0.865 ± 0.162	0.188 ± 0.023	0.088 ± 0.018
DSGC+GPF	0.941 ± 0.020	0.814 ± 0.016	0.945 ± 0.014	0.861 ± 0.170	0.198 ± 0.010	0.098 ± 0.007
SigMaNet+GPPT	-0.001 ± 0.003	0.004 ± 0.009	0.007 ± 0.009	0.146 ± 0.181	0.000 ± 0.000	0.003 ± 0.007
SigMaNet+Gprompt	0.143 ± 0.082	0.047 ± 0.085	0.195 ± 0.162	0.665 ± 0.151	0.095 ± 0.129	0.030 ± 0.036
SigMaNet+GPF	0.097 ± 0.095	0.101 ± 0.123	0.072 ± 0.106	0.662 ± 0.103	0.000 ± 0.000	0.039 ± 0.038
SigMaNet+All-in-one	0.186 ± 0.122	0.048 ± 0.089	0.249 ± 0.233	0.678 ± 0.178	0.071 ± 0.092	0.030 ± 0.036

Table 16: Ablation study on 5-shot learning: performance improvement by adding topological information (5-shot with Topo minus 5-shot without Topo). Positive values represent gains from adding topological information.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
MSGNN+GPPT	$+0.006$	$+0.002$	-0.014	$+0.002$	$+0.003$	$+0.017$
MSGNN+Gprompt	$+0.007$	$+0.001$	-0.024	$+0.002$	$+0.014$	-0.002
MSGNN+GPF	0.000	-0.026	-0.038	$+0.016$	$+0.031$	$+0.001$
MSGNN+All-in-one	$+0.036$	-0.055	-0.037	-0.105	$+0.027$	-0.008
SSSNET+GPPT	$+0.036$	$+0.023$	$+0.002$	-0.012	$+0.009$	0.000
SSSNET+Gprompt	-0.029	-0.001	$+0.011$	$+0.022$	-0.006	$+0.010$
SSSNET+GPF	-0.025	-0.013	-0.022	-0.040	-0.012	-0.011
SSSNET+All-in-one	$+0.015$	$+0.015$	$+0.003$	$+0.011$	-0.017	-0.011
DSGC+GPPT	-0.013	-0.038	-0.116	-0.077	-0.021	-0.005
DSGC+Gprompt	-0.033	-0.070	-0.047	-0.036	-0.004	$+0.005$
DSGC+All-in-one	-0.006	-0.060	-0.044	-0.022	-0.020	$+0.008$
DSGC+GPF	-0.003	-0.077	-0.038	-0.008	-0.016	$+0.003$
SigMaNet+GPPT	0.000	$+0.004$	$+0.104$	$+0.085$	$+0.039$	$+0.005$
SigMaNet+Gprompt	-0.074	$+0.075$	-0.059	-0.449	$+0.021$	$+0.023$
SigMaNet+GPF	$+0.021$	$+0.033$	$+0.089$	-0.379	$+0.118$	$+0.003$
SigMaNet+All-in-one	-0.072	$+0.075$	-0.099	-0.377	$+0.069$	$+0.012$

B Implementation details

B.1 Dataset details

Table 17 summarizes the dataset statistics, including the number of nodes, number of ground-truth clusters, positive and negative edge counts, and average degree for the four synthetic SDSBM settings and the two real-world signed graph datasets, Rainfall and SP1500.

Table 17: Dataset summary statistics, where SDSBM results are averaged over five random seeds and we also report one standard deviation.

Dataset	n	K	$\lvert\mathcal{E}^+\rvert$	$\lvert\mathcal{E}^-\rvert$	Avg. degree
SDSBM-1	1,000	3	$25{,}426 \pm 109$	$24{,}565 \pm 76$	49.99 ± 0.12
SDSBM-2	1,000	4	$21{,}949 \pm 101$	$27{,}958 \pm 128$	49.91 ± 0.13
SDSBM-3	1,000	4	$21{,}514 \pm 129$	$28{,}433 \pm 163$	49.95 ± 0.20
SDSBM-4	1,000	3	$25{,}913 \pm 196$	$24{,}013 \pm 144$	49.93 ± 0.23
Rainfall	306	6	64,408	29,228	306.00
SP1500	1,193	10	1,069,319	353,930	1193.00

B.2 Setup

Experiments were conducted on one compute node with one Nvidia H100 GPU with driver version 580.95.05 and CUDA version 13.0, 20 Intel(R) Xeon(R) Platinum 8468 CPUs and 234GB RAM.

B.3 Hyperparameters

All experiments use a unified training protocol to ensure fair comparison across methods. Table 18 details the hyperparameter settings. In particular:

Optimizer and training duration. All models are trained with the Adam optimizer (learning rate 0.01 by default, weight decay 5×10^{-4}, dropout 0.5) for up to 1,000 epochs with early stopping (patience 400 epochs). Pre-training fine-tuning stages use a shorter budget of 100 epochs and patience 40. All metrics are reported as the mean ± standard deviation over 5 independent training seeds (0, 10, 20, 30, 40).

Table 18: Hyperparameter settings used across all experiments.

Category	Hyperparameter	Value	Applies to
Architecture	GNN hidden dim (d)	32	All GNNs
	TPL output dim (d_T)	32	TopoSIGN, TopoDIG
	SigMaNet hidden (complex channels)	1	SigMaNet only
	GNN layers	2	All GNNs
	Chebyshev order (K)	1	MSGNN, TopoSIGN, TopoDIG, GFMs
	Magnetic charge (q)	0.25	MSGNN, TopoSIGN, TopoDIG, GFMs
	Laplacian normalization	symmetric	All MSConv-based models
	Dropout	0.5 (0.0 for SAMGPT)	All
Node clustering (baselines)	Optimizer	Adam	All
	Epochs	1,000	All
	Learning rate	0.01	All
	Weight decay	5×10^{-4}	All
	Early stopping patience	400 epochs	All
GFM pre-training	Optimizer	Adam	All GFMs
	Epochs	1,000	All GFMs
	Learning rate (fixed)	10^{-2}	All GFMs
	Early stopping patience	400 epochs	All GFMs
	Link-sign batch size	64	All GFMs
GFM fine-tuning	Optimizer	Adam	All GFMs
	Epochs	100	All GFMs
	Early stopping patience	40 epochs	All GFMs
	Learning rate (searched)	$\{10^{-2}, 5{\times}10^{-3}, 10^{-3}, 5{\times}10^{-4}, 10^{-4}\}$	selected by val ARI
PI computation (TDA)	k-hop neighborhood	2	TopoSIGN, TopoDIG
	Filtration max scale	10.0	TopoSIGN, TopoDIG
	Pixel size	1.0	TopoSIGN, TopoDIG
	PI grid / feature dim	$10 \times 10 = 100$	TopoSIGN, TopoDIG
	Top-k landmarks	80	TopoSIGN, TopoDIG
Data splits	Node train set	10%	All
	Node validation set	10%	All
	Node test set	80%	All
Evaluation	Training seeds	5 ($\{0, 10, 20, 30, 40\}$)	All
	Metric	ARI (mean $\pm$ std)	All

Data splits. The graph is split once per dataset instance using deterministic seeds. For node clustering, each graph is partitioned into 10% train (seed) nodes (labeled for NLL loss), 10% validation nodes (for ARI-based early stopping), and 80% test nodes (on which the final ARI is reported).

Hyperparameter selection. The only tuned hyperparameter is the learning rate during finetuning, searched over $\{10^{-2}, 5{\times}10^{-3}, 10^{-3}, 5{\times}10^{-4}, 10^{-4}\}$. All other hyperparameters are fixed across all methods and datasets: hidden dimension 32, topological hidden dimension 32, Chebyshev order K=1, magnetic charge q=0.25, 2 MSConv layers, normalization='sym', and persistence-image pixel size 1.0. The best learning rate is selected per method and dataset per seed by validation ARI. Multiclass node objectives use finite NLL loss on log-softmax outputs.

Balanced pre-training mini-batches. During pre-training, the GFM encoder is trained on the observed signed directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$, where $\mathcal{E}^+$ and $\mathcal{E}^-$ denote the positive and negative edge sets. We use all observed edges as the pre-training pool; no separate link split is held out, because pre-training is unsupervised and is fully decoupled from the node-clustering evaluation splits.

At each pre-training epoch we construct a mini-batch of size $B = 64$ by *stratified sign sampling*: we draw $B/2$ edges uniformly at random from $\mathcal{E}^+$ and $B/2$ from $\mathcal{E}^-$, yielding a balanced batch $\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^-$ with $|\mathcal{B}^+| = |\mathcal{B}^-| = B/2$. If one sign class is absent (e.g. for unsigned graphs), we revert to uniform random sampling over all edges. This balancing is important because real-world signed networks are often heavily skewed toward positive edges [1], and an unbalanced batch would bias the binary cross-entropy loss of the sign-prediction (SP) objective. Given node embeddings $\mathbf{Z}$ produced by the frozen backbone, an MLP decoder scores each edge $(u, v) \in \mathcal{B}$ as positive or negative via $\hat{y}_{uv} = \sigma(\text{MLP}([\mathbf{z}_u \| \mathbf{z}_v]))$, optimized with binary cross-entropy. The balanced batch ensures equal gradient contributions from both sign classes.

The structured auxiliary objectives used in Table 4 are derived from the same balanced batch $\mathcal{B}$, so their class distributions are naturally balanced as well:

- **SP** (sign prediction). Given node embeddings $\mathbf{Z}$ produced by the frozen backbone, an MLP decoder scores each edge $(u, v) \in \mathcal{B}$ as positive or negative via $\hat{y}_{uv} = \sigma(\text{MLP}([\mathbf{z}_u \| \mathbf{z}_v]))$, optimized with binary cross-entropy. The balanced batch ensures equal gradient contributions from both sign classes.

- **DP** (direction prediction). The reversed batch $\mathcal{B}_{\text{rev}} = \{(v, u) \mid (u, v) \in \mathcal{B}\}$ is constructed by flipping the source and destination indices. The combined pool $\mathcal{B} \cup \mathcal{B}_{\text{rev}}$ (size $2B$) is labeled 1 for forward and 0 for reversed edges; the 1:1 ratio is preserved exactly.
- **3C** (three-class sign and existence prediction). Non-edges $\mathcal{N}$ of size $|\mathcal{B}|$ are sampled via sparse negative sampling [69] from node pairs absent in $\mathcal{E}^+ \cup \mathcal{E}^-$. Labels are 0 (positive forward), 1 (negative forward), and 2 (non-edge), with $B/2$ edges per observed class and B non-edges, giving a $\frac{1}{4}$:$\frac{1}{4}$:$\frac{1}{2}$ class ratio.
- **4C** (four-class directed sign prediction). The reversed batch is concatenated to form $\mathcal{B} \cup \mathcal{B}_{\text{rev}}$ (size $2B$). Labels encode all four sign×direction combinations (positive forward, negative forward, positive reversed, negative reversed), each of size $B/2$, yielding a perfectly balanced four-class distribution.
- **5C** (five-class directed sign and existence prediction). Non-edges $\mathcal{N}$ (size B) are added to the 4C pool, giving five classes of sizes $B/2$, $B/2$, $B/2$, $B/2$, B; the four observed classes remain balanced.

The model is pre-trained for up to 1,000 epochs with Adam (learning rate 10^{-2}, weight decay 5×10^{-4}) and early stopping with patience 400 on the training loss. Fine-tuning uses a separate learning rate selected from $\{10^{-2}, 5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$ via validation ARI, with the pre-trained weights fixed as the starting point.

Evaluation metric. We report Adjusted Rand Index (ARI) on the test split (80% of nodes). ARI equals 1 for perfect agreement with ground-truth clusters and 0 for random clustering.

B.4 Runtime and Computational Cost

To provide a transparent breakdown of the computational cost, we explicitly report the runtime for preprocessing, model training, and the whole running time:

- **Training and Evaluation Runtime (After Preprocessing):** Table 19 reports the runtime (pre-training, prompt-tuning, and evaluation per run/seed) for graph prompt learning methods. Table 20 reports the training and evaluation runtime for standard signed GNNs and signed GNNs plus the topological branch in semi-supervised clustering.

- **Preprocessing Time:** Persistent Image (PI) extraction is computed only once as a preprocessing step prior to model training. The exact preprocessing time per dataset is detailed in Table 21. Because persistence diagrams are extracted independently for each node's local ego-network, this step is easily parallelizable across CPU cores.

- **Whole Running Time:** The total running time for our methods consists of the one-time preprocessing time in Table 21 plus the training and evaluation time in Table 19 (or Table 20). Once topological features are computed and stored, they do not add any computational cost during repeated training, tuning, or evaluation.

The empirical results reveal a clear and practical trade-off. As detailed in Table 21, the initial feature extraction takes roughly 2 to 32 minutes for sparse and moderately dense graphs (e.g., Rainfall, SDSBMs), though it scales up significantly on dense, complete graphs like SP1500. However, because each node's ego-network is processed independently, this upfront whole-time bottleneck is highly parallelizable across standard CPU cores.

More importantly, this decoupled design successfully shifts the computational burden away from the iterative model training and evaluation loops. As shown in Tables 19 and 20, once the topological summaries are precomputed, ingesting them introduces only a moderate overhead during training. For instance, incorporating the topological branch typically extends the training time by less than a factor of two compared to the bare signed GNN backbones, keeping the training-only cost firmly on the order of a few minutes per run. Consequently, while the whole running time on a newly seen dataset must account for the initial preprocessing phase, any subsequent operations only incur the highly competitive training-only runtime. We believe this demonstrates that TopoSIGN achieves performance gains while remaining computationally practical.

Table 19: Average runtime (seconds) per run across different seeds for graph prompt learning methods (corresponding to Table 1), covering pre-training, prompt-tuning, and evaluation **after** preprocessing. The shortest time is marked in **bold** while the second shortest time is underlined.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
GPPT	95 ± 11	109 ± 15	129 ± 13	118 ± 16	168 ± 12	285 ± 47
Gprompt	68 ± 16	73 ± 26	93 ± 7	77 ± 10	80 ± 17	126 ± 20
GPF	69 ± 7	$\mathbf{69 \pm 18}$	$\mathbf{81 \pm 19}$	84 ± 17	98 ± 10	128 ± 26
All-in-one	66 ± 13	70 ± 12	95 ± 5	$\mathbf{74 \pm 14}$	80 ± 14	123 ± 20
SAMGPT	$\mathbf{64 \pm 14}$	70 ± 16	85 ± 16	83 ± 13	$\mathbf{77 \pm 15}$	$\mathbf{97 \pm 10}$
TopoDIG	141 ± 12	203 ± 26	177 ± 19	179 ± 15	218 ± 28	304 ± 85
TopoMSGNN	324 ± 36	399 ± 42	304 ± 32	227 ± 17	270 ± 56	386 ± 102
TopoSSSNET	879 ± 104	656 ± 84	670 ± 56	629 ± 137	1003 ± 225	1380 ± 278

Table 20: Average training and evaluation runtime (seconds) per run across different seeds for signed GNNs and signed GNNs plus the topological branch (corresponding to Table 2), excluding one-time preprocessing. The shortest time is marked in **bold** while the second shortest time is underlined.

Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
DSGC	$\mathbf{54 \pm 3}$	$\mathbf{56 \pm 5}$	$\mathbf{56 \pm 2}$	$\mathbf{61 \pm 17}$	$\mathbf{56 \pm 2}$	$\mathbf{82 \pm 7}$
SigMaNet	140 ± 47	131 ± 38	90 ± 8	125 ± 45	135 ± 45	188 ± 45
MSGNN	94 ± 32	78 ± 11	109 ± 29	64 ± 2	88 ± 18	224 ± 73
SSSNET	100 ± 4	99 ± 1	106 ± 18	99 ± 1	130 ± 56	170 ± 22
DSGC+Topo	94 ± 32	120 ± 33	127 ± 23	81 ± 14	150 ± 26	201 ± 53
SigMaNet+Topo	158 ± 43	182 ± 32	162 ± 37	165 ± 47	130 ± 36	223 ± 51
MSGNN+Topo	148 ± 34	147 ± 36	134 ± 27	91 ± 13	185 ± 10	298 ± 43
SSSNET+Topo	166 ± 28	163 ± 29	195 ± 35	146 ± 17	264 ± 44	409 ± 78

B.5 Parameter Sensitivity Analysis

B.5.1 Sensitivity Analysis on Ego-Network Radius (k-hop)

In our main experiments (Table 1 and Table 2), the topological features are extracted from 2-hop ego-networks ($k = 2$) with a pixel size of 1.0. To investigate the sensitivity and robustness of TopoSIGN with respect to the ego-network neighborhood radius, we evaluate our framework under an expanded 3-hop context ($k = 3$, pixel_size $= 1.0$). Table 22 and Table 23 present the downstream node clustering ARI for graph prompt learning and semi-supervised signed GNNs, respectively. This comparison isolates the impact of higher-order multi-hop connectivity patterns against localized neighborhood topology.

The empirical results reveal a nuanced but consistent trend. In the prompt learning setting (Table 22), increasing the radius to $k = 3$ yields highly stable performance with only marginal fluctuations compared to $k = 2$. In the semi-supervised setting (Table 23), expanding the receptive field to $k = 3$ actually brings noticeable ARI improvements on several synthetic datasets (e.g., DSGC+Topo and SigMaNet+Topo on SDSBM-1 to 3), while maintaining comparable results on real-world networks like Rainfall and SP1500.

Despite the performance gains of $k = 3$ on certain baselines, we set $k = 2$ as the default in our main experiments to strike an optimal balance between topological discriminability and computational efficiency. Because a 2-hop ego-network is mathematically sufficient to capture the fundamental

Table 21: One-time Persistent Image (PI) preprocessing time (in seconds) across datasets. This step is executed once prior to training and can be parallelized across CPU cores.

Dataset	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
PI computation (s)	1917.12	1924.32	1928.89	1904.11	145.67	9435.03

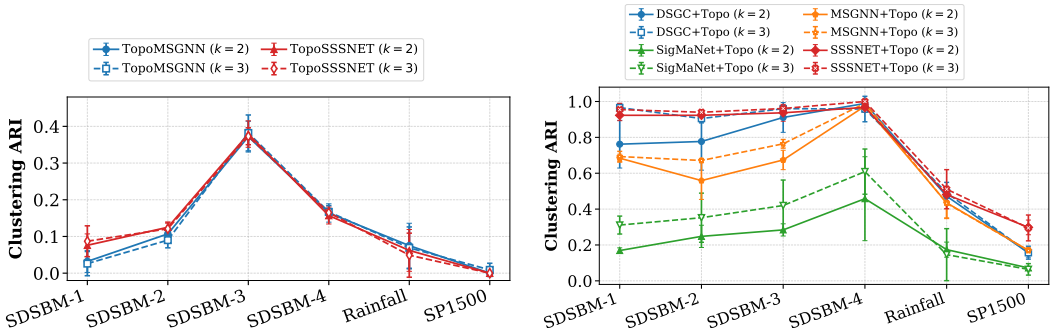
building blocks of structural balance (i.e., signed triangles and local cycles), it provides strong structural regularization without incurring the larger offline preprocessing cost required to extract persistence diagrams from 3-hop dense subgraphs.

Table 22: Pre-training and prompt learning performance comparison in terms of downstream node clustering ARI across different ego-network radii (k-hop). We report the mean ARI among five runs plus/minus one standard deviation.

k-hop	Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
$k=2$	TopoMSGNN	0.032 ± 0.030	0.107 ± 0.009	0.372 ± 0.042	0.164 ± 0.025	0.075 ± 0.061	0.000 ± 0.000
	TopoSSSNET	0.076 ± 0.031	0.125 ± 0.011	0.379 ± 0.036	0.157 ± 0.023	0.062 ± 0.057	0.000 ± 0.000
$k=3$	TopoMSGNN	0.026 ± 0.033	0.090 ± 0.021	0.382 ± 0.049	0.167 ± 0.017	0.069 ± 0.057	0.009 ± 0.018
	TopoSSSNET	0.087 ± 0.042	0.121 ± 0.018	0.373 ± 0.023	0.165 ± 0.009	0.049 ± 0.060	0.000 ± 0.000

Table 23: Semi-supervised node clustering performance (ARI) on real-world and synthetic signed datasets across different ego-network radii (k-hop). We report the mean ARI among five runs plus/minus one standard deviation.

k-hop	Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
$k=2$	DSGC+Topo	0.762 ± 0.133	0.777 ± 0.160	0.911 ± 0.083	0.989 ± 0.009	0.472 ± 0.044	0.156 ± 0.024
	SigMaNet+Topo	0.169 ± 0.015	0.248 ± 0.062	0.284 ± 0.034	0.458 ± 0.234	0.174 ± 0.042	0.072 ± 0.006
	MSGNN+Topo	0.684 ± 0.022	0.559 ± 0.105	0.674 ± 0.054	0.974 ± 0.022	0.435 ± 0.084	0.170 ± 0.013
	SSSNET+Topo	0.923 ± 0.030	0.923 ± 0.029	0.937 ± 0.047	0.967 ± 0.028	0.481 ± 0.034	0.298 ± 0.041
$k=3$	DSGC+Topo	0.966 ± 0.021	0.905 ± 0.024	0.960 ± 0.013	0.958 ± 0.071	0.491 ± 0.058	0.157 ± 0.036
	SigMaNet+Topo	0.311 ± 0.050	0.352 ± 0.137	0.420 ± 0.142	0.609 ± 0.126	0.146 ± 0.145	0.064 ± 0.032
	MSGNN+Topo	0.693 ± 0.029	0.671 ± 0.011	0.764 ± 0.024	0.990 ± 0.009	0.433 ± 0.085	0.169 ± 0.014
	SSSNET+Topo	0.955 ± 0.034	0.940 ± 0.014	0.961 ± 0.018	1.000 ± 0.000	0.511 ± 0.109	0.295 ± 0.072



(a) Graph Prompt Learning

(b) Semi-Supervised Signed GNNs

Figure 2: Downstream node clustering ARI sensitivity comparison across ego-network extraction radii ($k = 2$ vs. $k = 3$) over diverse datasets. Solid lines represent $k = 2$ and dashed lines represent $k = 3$.

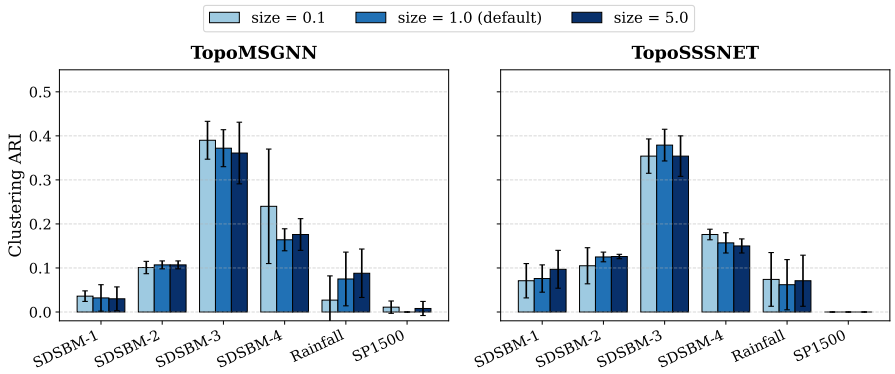
B.5.2 Sensitivity Analysis on Persistence Image Resolution (Pixel Size)

The discretization resolution of persistence images directly determines the granularity of the vectorized topological representations. To evaluate model sensitivity to this hyperparameter, we conduct experiments across varying grid resolutions with pixel_size $\in \{0.1, 1.0, 5.0\}$ while keeping the default 2-hop neighborhood ($k = 2$). Table 24 and Table 25 summarize the node clustering performance across pre-training and semi-supervised settings. These results demonstrate the stability of TopoSIGN across a reasonable range of topological feature discretizations and confirm that our framework does not require meticulous resolution tuning to achieve competitive performance.

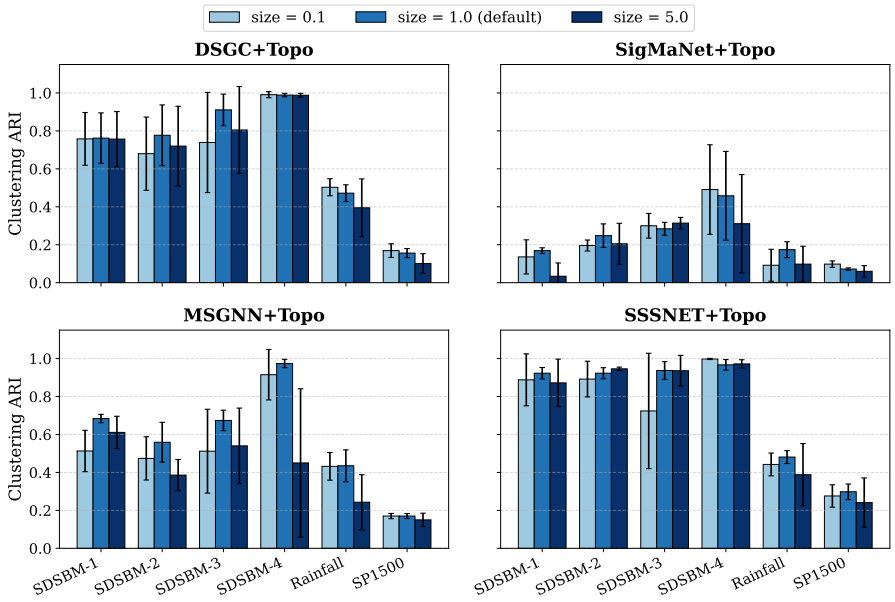
The empirical results reveal that while the framework functions across different resolutions, grid granularity does meaningfully impact downstream accuracy. In the prompt learning setting (Table 24), performance remains relatively stable across resolutions with minor fluctuations. However, in the semi-supervised setting (Table 25), deviating from the default resolution (pixel_size $= 1.0$) noticeably degrades performance on several datasets (e.g., MSGNN+Topo and SSSNET+Topo on SDSBM-3, or MSGNN+Topo on SDSBM-4 under size $= 5.0$).

This performance behavior aligns with the inherent mechanics of persistence images and reflects a trade-off in resolution granularity. A balanced grid (pixel_size $= 1.0$) acts as an effective topological

filter, aggregating nearby persistence points to produce dense, robust structural representations. Conversely, an overly fine grid (pixel_size $= 0.1$) produces highly sparse feature matrices susceptible to localized topological noise, while an overly coarse grid (pixel_size $= 5.0$) over-smoothes the topological landscape, sacrificing discriminative fine-grained structural features.



(a) Graph Prompt Learning



(b) Semi-Supervised Signed GNNs

Figure 3: Downstream node clustering ARI sensitivity comparison across persistence image discretization resolutions (pixel_size $\in \{0.1, 1.0, 5.0\}$) over diverse datasets. Bars from lighter to darker shades denote pixel_size $= 0.1$, 1.0 (default), and 5.0, respectively.

B.6 Ablation Study: Persistent Homology vs. Explicit Signed Degree Features

To isolate the performance gain specifically contributed by persistent homology compared to non-topological degree signals, we construct an explicit signed-degree baseline. Specifically, for each node, we extract a 2-dimensional signed net degree descriptor consisting of the net in-degree and net out-degree, defined as Net In $= d_{in}^{+} - d_{in}^{-}$ and Net Out $= d_{out}^{+} - d_{out}^{-}$. To stabilize the degree distribution while preserving the directional sign of the net balance, we apply a signed logarithmic compression: $\text{sign}(x) \cdot \log(1 + |x|)$ for each dimension $x \in \{\text{Net In}, \text{Net Out}\}$. We then evaluate the model performance when using this explicit degree representation in place of the persistent homology features. Tables 26 and 27 present the downstream node clustering ARI across datasets under the prompt-based learning and semi-supervised signed GNN settings, respectively. This comparison verifies whether multi-scale homological summaries (H_0 connectivity and H_1 cycles) provide structural benefits beyond directional signed net degree statistics.

Table 24: Pre-training and prompt learning performance comparison in terms of downstream node clustering ARI across different persistence image pixel sizes. We report the mean ARI among five runs plus/minus one standard deviation.

Pixel Size	Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
size $= 5.0$	TopoMSGNN	0.030 ± 0.027	0.107 ± 0.009	0.361 ± 0.070	0.176 ± 0.036	0.088 ± 0.055	0.008 ± 0.016
	TopoSSSNET	0.097 ± 0.043	0.126 ± 0.005	0.354 ± 0.046	0.150 ± 0.016	0.071 ± 0.058	0.000 ± 0.000
size $= 1.0$	TopoMSGNN	0.032 ± 0.030	0.107 ± 0.009	0.372 ± 0.042	0.164 ± 0.025	0.075 ± 0.061	0.000 ± 0.000
	TopoSSSNET	0.076 ± 0.031	0.125 ± 0.011	0.379 ± 0.036	0.157 ± 0.023	0.062 ± 0.057	0.000 ± 0.000
size $= 0.1$	TopoMSGNN	0.036 ± 0.012	0.101 ± 0.014	0.390 ± 0.043	0.240 ± 0.130	0.027 ± 0.055	0.011 ± 0.014
	TopoSSSNET	0.071 ± 0.039	0.105 ± 0.041	0.354 ± 0.039	0.176 ± 0.012	0.074 ± 0.061	0.000 ± 0.000

Table 25: Semi-supervised node clustering performance (ARI) on real-world and synthetic signed datasets across different persistence image pixel sizes. We report the mean ARI among five runs plus/minus one standard deviation.

Pixel Size	Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
size $= 5.0$	DSGC+Topo	0.757 ± 0.145	0.720 ± 0.210	0.805 ± 0.229	0.988 ± 0.010	0.395 ± 0.152	0.101 ± 0.052
	SigMaNet+Topo	0.034 ± 0.070	0.205 ± 0.108	0.314 ± 0.030	0.311 ± 0.259	0.098 ± 0.094	0.060 ± 0.030
	MSGNN+Topo	0.611 ± 0.085	0.386 ± 0.082	0.540 ± 0.199	0.450 ± 0.391	0.243 ± 0.145	0.150 ± 0.035
	SSSNET+Topo	0.872 ± 0.125	0.946 ± 0.009	0.936 ± 0.081	0.972 ± 0.022	0.388 ± 0.164	0.241 ± 0.130
size $= 1.0$	DSGC+Topo	0.762 ± 0.133	0.777 ± 0.160	0.911 ± 0.083	0.989 ± 0.009	0.472 ± 0.044	0.156 ± 0.024
	SigMaNet+Topo	0.169 ± 0.015	0.248 ± 0.062	0.284 ± 0.034	0.458 ± 0.234	0.174 ± 0.042	0.072 ± 0.006
	MSGNN+Topo	0.684 ± 0.022	0.559 ± 0.105	0.674 ± 0.054	0.974 ± 0.022	0.435 ± 0.084	0.170 ± 0.013
	SSSNET+Topo	0.923 ± 0.030	0.923 ± 0.029	0.937 ± 0.047	0.967 ± 0.028	0.481 ± 0.034	0.298 ± 0.041
size $= 0.1$	DSGC+Topo	0.758 ± 0.139	0.680 ± 0.193	0.739 ± 0.264	0.991 ± 0.016	0.503 ± 0.045	0.169 ± 0.036
	SigMaNet+Topo	0.136 ± 0.090	0.196 ± 0.029	0.300 ± 0.065	0.491 ± 0.236	0.092 ± 0.084	0.098 ± 0.017
	MSGNN+Topo	0.513 ± 0.109	0.474 ± 0.114	0.512 ± 0.221	0.915 ± 0.133	0.432 ± 0.073	0.170 ± 0.014
	SSSNET+Topo	0.888 ± 0.137	0.892 ± 0.094	0.724 ± 0.304	0.998 ± 0.003	0.442 ± 0.060	0.276 ± 0.059

Table 26: Pre-training and prompt learning performance comparison in terms of downstream node clustering ARI. We report the mean ARI among five runs plus/minus one standard deviation. Negative difference values in Δ are marked in **bold**.

Features	Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
Degree-based	TopoMSGNN	0.026 ± 0.031	0.098 ± 0.026	0.370 ± 0.041	0.235 ± 0.130	0.073 ± 0.060	0.001 ± 0.002
	TopoSSSNET	0.039 ± 0.019	0.111 ± 0.033	0.381 ± 0.019	0.133 ± 0.019	0.045 ± 0.055	0.015 ± 0.020
PH	**TopoMSGNN (Ours)**	0.032 ± 0.030	0.107 ± 0.009	0.372 ± 0.042	0.164 ± 0.025	0.075 ± 0.061	0.000 ± 0.000
	TopoSSSNET (Ours)	0.076 ± 0.031	0.125 ± 0.011	0.379 ± 0.036	0.157 ± 0.023	0.062 ± 0.057	0.000 ± 0.000
Δ (Degree $-$ PH)	TopoMSGNN	$\mathbf{-0.006}$	$\mathbf{-0.009}$	$\mathbf{-0.002}$	$+0.071$	$\mathbf{-0.002}$	$+0.001$
	TopoSSSNET	$\mathbf{-0.037}$	$\mathbf{-0.014}$	$+0.002$	$\mathbf{-0.024}$	$\mathbf{-0.017}$	$+0.015$

Table 27: Semi-supervised node clustering performance (ARI) on real-world and synthetic signed datasets. We compare our approach against state-of-the-art signed GNNs. We report the mean ARI among five runs plus/minus one standard deviation. The best method is marked in **bold** while the second best is marked with underline.

Features	Method	SDSBM-1	SDSBM-2	SDSBM-3	SDSBM-4	Rainfall	SP1500
Degree-based	DSGC+Topo	0.733 ± 0.128	0.760 ± 0.227	0.919 ± 0.057	$\mathbf{0.989 \pm 0.009}$	$\underline{0.491 \pm 0.058}$	0.157 ± 0.036
	SigMaNet+Topo	0.145 ± 0.090	0.271 ± 0.022	0.310 ± 0.066	0.527 ± 0.087	0.146 ± 0.145	0.064 ± 0.032
	MSGNN+Topo	0.677 ± 0.013	0.549 ± 0.082	0.695 ± 0.035	$\underline{0.975 \pm 0.018}$	0.433 ± 0.085	0.169 ± 0.014
	SSSNET+Topo	$\mathbf{0.948 \pm 0.015}$	$\underline{0.858 \pm 0.125}$	$\underline{0.922 \pm 0.087}$	0.971 ± 0.021	$\mathbf{0.511 \pm 0.109}$	$\underline{0.295 \pm 0.072}$
PH	DSGC+Topo	0.762 ± 0.133	0.777 ± 0.160	0.911 ± 0.083	$\mathbf{0.989 \pm 0.009}$	0.472 ± 0.044	0.156 ± 0.024
	SigMaNet+Topo	0.169 ± 0.015	0.248 ± 0.062	0.284 ± 0.034	0.458 ± 0.234	0.174 ± 0.042	0.072 ± 0.006
	MSGNN+Topo	0.684 ± 0.022	0.559 ± 0.105	0.674 ± 0.054	0.974 ± 0.022	0.435 ± 0.084	0.170 ± 0.013
	SSSNET+Topo	$\underline{0.923 \pm 0.030}$	$\mathbf{0.923 \pm 0.029}$	$\mathbf{0.937 \pm 0.047}$	0.967 ± 0.028	0.481 ± 0.034	$\mathbf{0.298 \pm 0.041}$