NyS-Binary: A Feature-Free, Algebraic Framework for Binary Node Representation Learning

Anonymous authors
Paper under double-blind review

Abstract

Graph neural networks deliver strong node representations, but their high-dimensional floating-point embeddings are costly to store and slow to serve on resource-constrained devices. Binary graph hashing addresses this by mapping each node to a compact bitcode, yet existing methods force an uncomfortable choice: random-walk and sketching approaches such as node2binary and NodeSketch are slow to construct and discard global structure, while shallow linear hashes such as NodeSig trade away accuracy on harder graphs. We present NyS-Binary, a feature-free, algebraic, and sparsely supervised hashing framework that operates purely on graph topology without requiring any node attributes, combines a randomized Nyström-inspired low-rank sketch of a higher-order graph transition operator with a leakage-safe label-diffusion channel, fuses the two via a convex blend, and quantizes the result with an adaptive per-column median rule. Across ten node-classification benchmarks and diverse downstream classifiers, NyS-Binary consistently attains the highest average accuracy among all binary hashing methods, substantially outperforming the strongest feature-free binary baselines on both standard neural probes and spiking neuromorphic classifiers, while producing compact bitcodes in fractions of a second, over an order of magnitude faster than NodeSketch and up to three orders of magnitude faster than learned iterative baselines. Crucially, because the landmark structural channel provides a strong topological foundation independent of supervision, NyS-Binary operates in a semi-supervised regime and maintains competitive accuracy even under extreme label scarcity. Because the representations are strictly binary, they also natively support neuromorphic classifiers built on gradient-free learning and surrogate-gradient spiking networks, which compute via binary spike accumulation rather than floating-point arithmetic. We report where the method is strong, where it is not (notably extreme heterophily), and view it as evidence that principled algebraic sketching remains a competitive, scalable alternative to learned binarization.

1 Introduction

Modern graph machine learning produces high-quality node representations, but the representations themselves are usually dense floating-point vectors. At scale this is a bottleneck. A single-precision embedding of width 256 occupies one kilobyte per node, so a graph with a hundred million nodes needs a hundred gigabytes just to store the embedding table, before any inference. On edge hardware, and in privacy-sensitive on-device settings, both the storage and the cost of floating-point similarity are limiting.

Binary node codes are a direct response. A K-bit code is $32\times$ smaller than a K-dimensional `float32` vector, significantly accelerating downstream classification tasks while maintaining a highly compact memory footprint. The open question is not whether binary codes are desirable but whether we can build codes that are simultaneously (i) fast to construct, (ii) discriminative enough for downstream tasks, and (iii) portable across a range of classifiers, including non-standard ones. Existing work tends to satisfy at most two of the three.

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

Two broad families dominate. *Learned* binarizations, exemplified by Bi-GCN (Wang et al., 2021) and `node2binary` (Talukder et al., 2025), optimize an objective and typically preserve accuracy, but their construction is slow, sometimes minutes to hours per graph. *Sketching and projection* methods, such as NodeSketch (Yang et al., 2019) and NodeSig (Çelikkanat et al., 2022), avoid training, but recursive multi-hop sketching scales poorly and shallow projections lose accuracy on graphs with intricate structure. In practice one must decide whether to pay in time or in quality.

We introduce a feature-free, algebraic framework for constructing compact binary graph representations directly from graph topology. Our central contribution is a unified, modular algebraic pipeline that couples low-rank structural sketching of higher-order transition operators with a label-leakage-safe semantic diffusion channel, polynomial spectral smoothing, and adaptive quantile thresholding. While the structural channel can accommodate various low-rank approximations, we find that an asymmetric, randomized Nyström-inspired landmark extension provides a highly effective closed-form alternative: it completely avoids the repetitive matrix-vector products and polynomial recursions of iterative approaches, delivering a favorable accuracy-efficiency trade-off compared to full spectral decomposition. Layering the label-leakage-safe semantic diffusion channel on top of these structural coordinates yields binary codes that achieve state-of-the-art downstream performance while substantially reducing construction cost and storage relative to learned binary representation methods.

Contributions.

1. We introduce NyS-Binary, a closed-form binary graph hashing framework that couples a randomized Nyström-inspired sketch of a higher-order transition operator with a label-leakage-safe semantic diffusion channel and adaptive median quantization (Section 3). The construction is closed-form and near-linear in the graph size $(N + |E|)$ for sparse real-world networks.
2. We systematically evaluate on ten benchmarks and nine downstream classifiers, reporting competitive or superior performance across diverse probe architectures. We evaluate both a semi-supervised blend and a strictly unsupervised variant (struct_ratio $= 1.0$); our unsupervised variant establishes a new state-of-the-art among label-free, feature-free binary hashing baselines at sub-second generation cost, while our semi-supervised blend provides further discriminative refinement (Section 4).
3. We evaluate on neuromorphic spiking architectures, including SADP (a classifier trained via gradient-free agreement-driven synaptic learning without backpropagation (Lakshmi S et al., 2026)) and a surrogate-gradient spiking network. We find that our discrete binary codes natively match event-driven spike dynamics without lossy continuous-to-spike rate coding, achieving strong compatibility with neuromorphic substrates and leading prior binary hashing baselines (e.g., $+26.81\%$ over NodeSig) as well as the learned Bi-GCN baseline ($+9.45\%$) on SADP (Section 4).
4. We analyze the geometry of the codes through class-separation ratios (Section 5), and document the method's behavior on strongly heterophilous graphs under extended diffusion in the appendix (Section A.6).

2 Related Work

Node embeddings and hashing. Random-walk embeddings such as DeepWalk (Perozzi et al., 2014) and node2vec (Grover & Leskovec, 2016) popularized skip-gram style objectives for graphs, and matrix-factorization views unified several of them. For scalability, hashing and sketching approaches map nodes into Hamming space. NetHash (Wu et al., 2018) hashes rooted trees of node attributes; NodeSketch (Yang et al., 2019) recursively sketches the self-loop augmented adjacency matrix to preserve high-order proximity; NodeSig (Çelikkanat et al., 2022) applies sign-stable random projections to random-walk diffusion probabilities. AROPE (Zhang et al., 2018) preserves arbitrary-order proximity via eigendecomposition, and SimHash (Charikar, 2002) is a classical sign-random-projection technique. `node2binary` (Talukder et al., 2025) learns binary vectors through an explicit optimization. Our method is closest in spirit to the sketching family but differs by anchoring structure in a Nyström-inspired low-rank frame and by adding an explicit, label-leakage-safe semantic channel.

Binarized graph neural networks. Bi-GCN (Wang et al., 2021) binarizes weights and activations of a graph convolutional network, reporting large memory and speed savings at near full-precision accuracy. Bi-GCN represents an end-to-end learned, high-capacity binary baseline that optimizes an objective using continuous node features and iterative backpropagation. In contrast, NyS-Binary is a closed-form, training-free algebraic framework that operates purely on graph topology, generating binary codes orders of magnitude faster while maintaining competitive or superior downstream accuracy.

Low-rank approximation. The Nyström method (Williams & Seeger, 2001) approximates a matrix from sampled columns and has been studied in depth for kernel machines (Drineas & Mahoney, 2005); randomized SVD (Halko et al., 2011) offers related probabilistic guarantees. We apply these ideas to a graph transition operator rather than a kernel matrix.

Downstream and spiking classifiers. We evaluate with GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), H2GCN (Zhu et al., 2020), and LINKX (Lim et al., 2021), alongside linear probes. To evaluate deployment feasibility on brain-inspired, ultra-low-power neuromorphic hardware, we evaluate spiking neural networks trained by gradient-free agreement-driven learning (SADP) and surrogate-gradient dynamics (Neftci et al., 2019; Maass, 1997; Lakshmi S et al., 2026). Because spiking architectures compute via discrete all-or-nothing events, our binary codes naturally match event-driven spike dynamics without requiring continuous rate coding, demonstrating the strong compatibility of our representations on neuromorphic computing substrates.

3 Method

3.1 Notation and problem setting

Let $\mathcal{G} = (V, E)$ be an undirected graph with $N = |V|$ nodes and adjacency $\mathbf{A} \in \{0,1\}^{N\times N}$. We are given training labels $\{y_i\}_{i\in\mathcal{M}_{\text{tr}}}$ over C classes on a mask $\mathcal{M}_{\text{tr}} \subset \{1,\dots,N\}$, and we output a binary code matrix $\mathbf{B} \in \{0,1\}^{N\times K}$ for a target length K. In our canonical sparsely supervised setting, we divide the budget as $K = k_{\text{struct}} + k_{\text{label}}$ with $k_{\text{struct}} = \lfloor K/2 \rfloor$ structural bits and $k_{\text{label}} = K - k_{\text{struct}}$ semantic bits (a balanced 50/50 split). Importantly, our framework also supports a strictly unsupervised operational mode by allocating the entire bit budget to structural coordinates (struct_ratio $= 1.0$, i.e., $k_{\text{struct}} = K, k_{\text{label}} = 0$). In this unsupervised regime, the semantic codebook and label diffusion are bypassed entirely, generating K-bit representations purely from the graph topology without access to any node attributes or training labels.

3.2 Transition operator

We first augment self-loops and row-normalize to obtain a random-walk operator,

$$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N, \qquad \tilde{\mathbf{D}} = \operatorname{diag}\Big(\textstyle\sum_j \tilde{\mathbf{A}}_{ij}\Big), \qquad \mathbf{P} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}. \tag{1}$$

The self-loop keeps a node's own signal during propagation, and row normalization makes $\mathbf{P}$ row-stochastic, so $\mathbf{P}^t$ encodes the distribution of t-step walks.

3.3 Structural channel via randomized Nyström-inspired sketching

Rather than decomposing $\mathbf{P}$ in full, which costs $O(N^3)$, we sketch it from a landmark subset. While the classical Nyström method is formulated for symmetric positive semi-definite kernel matrices (Williams & Seeger, 2001), low-rank sketching of asymmetric operators via sampled submatrices is mathematically grounded in randomized numerical linear algebra through interpolative and asymmetric CUR decompositions (Drineas & Mahoney, 2005; Drineas et al., 2006). For our row-stochastic transition operator $\mathbf{P}$, we adopt an asymmetric column-projection sketch: we sample $m = \min\big(N, \max(k_{\text{struct}}, 125)\big)$ landmark indices $\mathcal{L}$ uniformly without replacement and extract the landmark-landmark core block $\mathbf{P}_{\ell\ell} = \mathbf{P}[\mathcal{L}, \mathcal{L}] \in \mathbb{R}^{m\times m}$ and the cross-block $\mathbf{P}_{n\ell} = \mathbf{P}[:, \mathcal{L}] \in \mathbb{R}^{N\times m}$. We compute the truncated SVD of the core block, $\mathbf{P}_{\ell\ell} = \mathbf{U}_\ell \mathbf{\Sigma}_\ell \mathbf{V}_\ell^\top$. Because $\mathbf{P}_{n\ell}$ consists of out-of-sample columns, we project them onto the left singular vectors $\mathbf{U}_\ell$

(which span the column space of the landmark core block) and scale by the inverse singular values to yield coordinates for every node:

$$\mathbf{R}_{\text{str}} = \mathbf{P}_{n\ell}\,\mathbf{U}_\ell\,(\mathbf{\Sigma}_\ell + \epsilon\mathbf{I})^{-1} \in \mathbb{R}^{N\times k_{\text{struct}}}, \tag{2}$$

where ϵ is a small stabilizer that guards against tiny singular values. The dominant costs are the $O(m^3)$ decomposition of the landmark block and the $O(Nmk_{\text{struct}})$ extension, both linear in N for fixed landmark budget m, which is where the speed comes from. We emphasize that our framework is structurally modular: while we use a Nyström-inspired construction for its strong accuracy-efficiency trade-off, this channel $\mathbf{R}_{\text{str}}$ can be seamlessly swapped with other low-rank approximations (e.g., Randomized SVD, Chebyshev filtering) without altering the subsequent semantic diffusion or quantization stages (see Appendix B.1 for a comprehensive empirical comparison).

3.4 Semantic channel via label-leakage-safe label diffusion

Structure alone is class-agnostic, so we add label semantics without leaking test information. We first build an orthogonal class codebook by drawing a Gaussian matrix $\mathbf{X} \sim \mathcal{N}(0,1) \in \mathbb{R}^{k_{\text{label}}\times C}$, transposing its thin QR factor $\mathbf{Q}$, and binarizing to hypercube vertices $\boldsymbol{h}_c = \text{sign}(\mathbf{Q}^\top[c,:]) \in \{-1,+1\}^{k_{\text{label}}}$. Well-separated codewords keep classes apart in the semantic subspace.

Ground-truth stream. We assign $\mathbf{S}_{\text{gt}}[i,:] = \boldsymbol{h}_{y_i}$ for $i \in \mathcal{M}_{\text{tr}}$ and $\mathbf{0}$ otherwise.

Pseudo-label stream. We propagate one-hot training seeds $\mathbf{Y}_0$ by a three-step walk, $\mathbf{Y}_t = \mathbf{P}^3\mathbf{Y}_0$, normalize to a posterior $\hat{p}_{i,c} = \mathbf{Y}_t[i,c]/(\sum_{c'}\mathbf{Y}_t[i,c'] + \epsilon)$, and for each *unlabeled* node with $\max_c \hat{p}_{i,c} \geq \tau$ we set $\mathbf{S}_{\text{pl}}[i,:] = \boldsymbol{h}_{c_i^\star}$ where $c_i^\star = \arg\max_c \hat{p}_{i,c}$; otherwise $\mathbf{0}$. Labeled nodes keep their ground-truth codeword. Under the standard transductive graph representation setting, the random-walk transition matrix $\mathbf{P}$ utilizes the full topological graph structure (including edges incident to test nodes), but the supervisory signal is strictly insulated: only training labels ever seed the walk ($\mathbf{S}_{\text{gt}}[i,:] = \mathbf{0}$ for $i \notin \mathcal{M}_{\text{tr}}$), and pseudo-labels are formed purely via structural random-walk confidence without accessing test ground-truth. We explicitly assert that no test label index $i \notin \mathcal{M}_{\text{tr}}$ seeds the propagation, ensuring that the semantic channel remains strictly *label-leakage-safe*.

3.5 Fusion, diffusion, and adaptive quantization

We concatenate the shared structural coordinates with each semantic stream,

$$\mathbf{W}_{\text{gt}} = [\mathbf{R}_{\text{str}} \,\|\, \mathbf{S}_{\text{gt}}], \qquad \mathbf{W}_{\text{pl}} = [\mathbf{R}_{\text{str}} \,\|\, \mathbf{S}_{\text{pl}}] \in \mathbb{R}^{N\times K}, \tag{3}$$

smooth each with a short polynomial diffusion that mixes local and higher-order neighborhoods,

$$\mathcal{D}_H(\mathbf{W}) = \frac{1}{H+1}\sum_{h=0}^{H}\mathbf{P}^h\mathbf{W}, \tag{4}$$

and blend the two streams with a convex combination weight $\alpha \in [0,1]$ before thresholding each column using an adaptive median rule scaled by quantile factor $\mu > 0$ (default $\mu = 0.50$):

$$\mathbf{W} = (1-\alpha)\,\mathcal{D}_H(\mathbf{W}_{\text{gt}}) + \alpha\,\mathcal{D}_H(\mathbf{W}_{\text{pl}}), \qquad \mathbf{B}[i,d] = \mathbf{1}\big[\mathbf{W}[i,d] > \mu\cdot\text{median}_j\,\mathbf{W}[j,d]\big]. \tag{5}$$

In the unsupervised variant (**NyS-Binary Unsupervised**, where $k_{\text{label}} = 0$), the semantic channel is omitted and the representation simplifies directly to $\mathbf{W} = \mathcal{D}_H(\mathbf{R}_{\text{str}}) \in \mathbb{R}^{N\times K}$, which is quantized via column-wise thresholding $\mathbf{B}[i,d] = \mathbf{1}[\mathbf{W}[i,d] > \mu\cdot\text{median}_j\,\mathbf{W}[j,d]]$ with zero label input. The per-column median makes each bit roughly balanced, which maximizes the information entropy of the discrete code. Throughout we use $K = 250$, $H = 3$, blend weight $\alpha = 0.50$, quantile threshold multiplier $\mu = 0.50$, confidence gate $\tau = 0.50$, landmark budget $m = 125$, and a balanced 50/50 bit split ($k_{\text{struct}} = 125, k_{\text{label}} = 125$). As demonstrated in the 4-classifier factorial interaction heatmaps across all 10 benchmarks (Figure 7 in Appendix B.2.2), the combination ($\alpha = 0.50, \tau = 0.50$) provides a highly stable, architecture-agnostic default across GCN, GraphSAGE, MLP, and LINKX. Establishing these defaults on an internal hyperparameter tuning split partitioned strictly from the training set, and keeping them strictly frozen for the 70/30 experiments in Table 1, demonstrates cross-regime parameter stability. Figure 1 and Algorithm 1 summarize the pipeline.

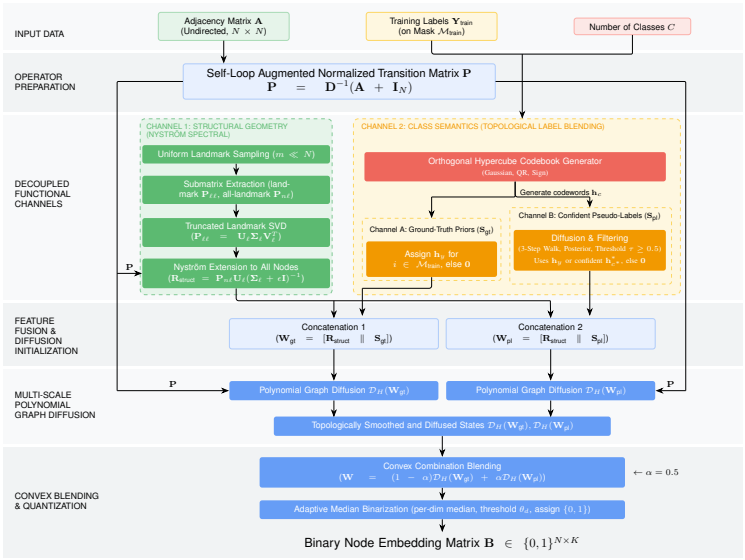


Figure 1: The NyS-Binary pipeline. A row-stochastic transition operator feeds a Nyström structural channel and a leakage-safe semantic channel; the two are diffused, convexly blended, and quantized by an adaptive per-column median rule.

Algorithm 1 NyS-Binary: closed-form binary graph hashing

Require: Adjacency $\mathbf{A} \in \{0,1\}^{N\times N}$; training labels on $\mathcal{M}_{\text{tr}}$ with labels y_i; code length K; hyperparameters $(H, \alpha, \mu, \tau, m, \epsilon)$ with defaults $(H = 3, \alpha = 0.50, \mu = 0.50, \tau = 0.50, m = 125, \epsilon = 10^{-10})$.

Ensure: Binary node codes $\mathbf{B} \in \{0,1\}^{N\times K}$.

Phase 1: Graph Normalization & Bit Allocation

$\mathbf{P} \leftarrow \tilde{\mathbf{D}}^{-1}(\mathbf{A} + \mathbf{I}_N)$ ▷ Normalized random-walk matrix

$k_{\text{struct}} \leftarrow \lfloor K/2 \rfloor$, $k_{\text{label}} \leftarrow K - k_{\text{struct}}$ ▷ Budget split (Structure vs Label)

Phase 2: Structural Channel (Nyström-Inspired Approximation)

$m \leftarrow \min(N, \max(k_{\text{struct}}, 125))$ ▷ Determine landmark budget ($m = 125$)

Sample landmark nodes $\mathcal{L} \subset \mathcal{V}$ uniformly at random, where $|\mathcal{L}| = m$

$\mathbf{U}_\ell, \mathbf{\Sigma}_\ell \leftarrow \text{trunc-SVD}_{k_{\text{struct}}}(\mathbf{P}[\mathcal{L}, \mathcal{L}])$ ▷ Eigendecomposition of landmark core

$\mathbf{R}_{\text{str}} \leftarrow \mathbf{P}[:, \mathcal{L}]\, \mathbf{U}_\ell(\mathbf{\Sigma}_\ell + \epsilon\mathbf{I})^{-1}$ ▷ Project nodes via Nyström-inspired extension Eq. 2

Phase 3: Semantic Channel (Label Codebook & Diffusion)

Generate orthogonal codebook $\{\boldsymbol{h}_c\}_{c=1}^{C} \in \{-1, 1\}^{k_{\text{label}}}$ via Gaussian QR-decomposition

Init ground-truth $\mathbf{S}_{\text{gt}} \in \mathbb{R}^{N\times k_{\text{label}}}$: $\mathbf{S}_{\text{gt}}[i, :] \leftarrow \boldsymbol{h}_{y_i}$ if $i \in \mathcal{M}_{\text{tr}}$, else $\mathbf{0}$

$\mathbf{Y}_t \leftarrow \mathbf{P}^3\mathbf{Y}_0$ ▷ Diffuse raw label one-hot distribution

Init pseudo-label $\mathbf{S}_{\text{pl}} \in \mathbb{R}^{N\times k_{\text{label}}}$: $\mathbf{S}_{\text{pl}}[i, :] \leftarrow \boldsymbol{h}_{\hat{y}_i}$ if confidence $\hat{p} \geq \tau$, else $\mathbf{0}$

Phase 4: Multi-Channel Blending & Quantization

$\mathbf{E}_{\text{gt}} \leftarrow \mathcal{D}_H([\mathbf{R}_{\text{str}} \| \mathbf{S}_{\text{gt}}])$ ▷ Diffuse concatenated ground-truth representation

$\mathbf{E}_{\text{pl}} \leftarrow \mathcal{D}_H([\mathbf{R}_{\text{str}} \| \mathbf{S}_{\text{pl}}])$ ▷ Diffuse concatenated pseudo-label representation

$\mathbf{W} \leftarrow (1-\alpha)\mathbf{E}_{\text{gt}} + \alpha\, \mathbf{E}_{\text{pl}}$ ▷ Convex combination of certainty and coverage

return $\mathbf{B}[i, d] = \mathbf{1}[\mathbf{W}[i, d] > \mu \cdot \text{median}_j\, \mathbf{W}[j, d]]$ ▷ Adaptive quantile thresholding Eq. 5

3.6 Complexity and Memory

For fixed landmark count m and code length K, construction is $O(m^3 + NmK + |E|KH)$: the landmark core decomposition, out-of-sample projection, and sparse polynomial diffusion. Under standard real-world graph sparsity ($|E| = O(N)$), construction scales near-linearly as $O(N)$ (vs. exact spectral methods). Storage is NK bits, a 32× reduction over $N \times K$ `float32` vectors.

4 EXPERIMENTS

4.1 DATASETS, CLASSIFIERS, AND PROTOCOL

We evaluate on ten benchmarks: four homophilous graphs (Cora, CiteSeer, PubMed, DBLP (Sen et al., 2008; Yang & Leskovec, 2012)); WikiCS (Mernyei & Cangea, 2020); two co-purchase graphs (Photo, Computers (Shchur et al., 2018)); and three heterophilous graphs (Chameleon, Squirrel, Amazon-Ratings (Pei et al., 2020; Platonov et al., 2023)). For DBLP, we adopt the standard 2-class benchmark (Yang et al., 2019) from SNAP (where multi-class and Two-Way SADP coincide). Dataset statistics appear in Table 3 (Appendix A).

Every embedding is evaluated across nine classifiers spanning linear probes (Logistic Regression), deep networks (MLP), standard GNNs (GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017)), heterophily-tailored architectures (H2GCN (Zhu et al., 2020), LINKX (Lim et al., 2021)), and neuromorphic spiking networks. For neuromorphic deployment, we benchmark three spiking (SNN) probes: gradient-free agreement-driven synaptic learning (SADP) (Lakshmi S et al., 2026), a two-class variant (SADP Two-Way, Appendix A.3.2), and a surrogate-gradient network (SADP-Surrogate) (Neftci et al., 2019; Maass, 1997). Discrete bitcodes natively match event-driven spike accumulations without lossy continuous rate coding. We use a 70/30 stratified split over seeds $\{42, 123, 77\}$. Hyperparameters ($\alpha = 0.50, \tau = 0.50$) were selected via grid search on an internal tuning split partitioned from the training pool (Figure 7) and frozen across all benchmarks. Codes are 250 bits. All runs use a laptop CPU (i7-11800H) and RTX 3050 Ti GPU.

We evaluate two configurations: (i) **NyS-Binary (Ours)** with a 50/50 structural-semantic blend ($k_{\text{struct}} = 125, k_{\text{label}} = 125$), and (ii) **NyS-Binary (Unsupervised)** with struct_ratio $= 1.0$ ($k_{\text{struct}} = 250, k_{\text{label}} = 0$), generating codes strictly from topology. We benchmark against three feature-free hashing baselines (NodeSig (Çelikkanat et al., 2022), NodeSketch (Yang et al., 2019), node2binary (Talukder et al., 2025)) and three attribute-aware references (Bi-GCN (Wang et al., 2021), continuous raw features, binarized raw features).

4.2 OVERALL ACCURACY

Table 1 aggregates accuracy across all ten benchmark datasets and three seeds. NyS-Binary (Ours) achieves the highest performance among all binary methods across every single one of the nine classifiers, establishing substantial margins on non-relational probes ($+9.41\%$ on MLP and $+9.97\%$ on Logistic Regression over NodeSig) and neuromorphic spiking classifiers ($+26.81\%$ on SADP and $+29.29\%$ on SADP-Surrogate over NodeSig; $+9.45\%$ over learned Bi-GCN on SADP).

In the strictly label-free setting, **NyS-Binary (Unsupervised)** ($k_{\text{struct}} = 250, k_{\text{label}} = 0$) provides an exact like-for-like comparison against unsupervised binary hashing baselines (NodeSig, NodeSketch, node2binary). Purely from graph topology with zero label inputs, it consistently secures the second-highest accuracy across all nine probes (72.89% on GCN, 75.19% on H2GCN, 73.18% on MLP), leading NodeSig by $+7.20\%$ on MLP and $+9.99\%$ on SADP, and NodeSketch by $+20.71\%$ on MLP. Layering the label channel in **NyS-Binary (Ours)** integrates closed-form label diffusion into a static binary code; unlike transductive algorithms like C&S (Huang et al., 2021) that output soft posteriors, NyS-Binary binarizes diffused signals into hardware-friendly bitcodes that serve as general-purpose features without backpropagation. Our low-supervision ablation (Appendix A.7) confirms that NyS-Binary dominates even under sparse labels (68.45% on GCN, 67.95% on LogReg vs. 64.17% and 57.24% for NodeSig; 0.37s vs. 4.50s for Bi-GCN). On datasets where top methods fall within one standard deviation (Appendix A.5), differences lie within noise margins; the strength of NyS-Binary is systemic consistency across all 10 benchmarks, 9 diverse probe families, and sub-second generation efficiency.

4.3 GENERATION TIME

Table 2 reports latency. NyS-Binary generates 250-bit codes in 0.38 s on average. NodeSig is faster in absolute terms, but Table 1 shows it trails badly in accuracy; learned graph neural baselines are over an order of magnitude slower ($13.9\times$ for Bi-GCN), while hierarchical tree-based clustering methods are up to three orders of magnitude slower (e.g., 1739s for node2binary on average). Fig-

Table 1: Overall mean accuracy (%) across all ten datasets and three seeds. Binary hashing methods are grouped above the rule; attribute-aware references (Bi-GCN, raw features) utilize continuous node attributes and gradient training (or raw feature baselines) below the inner rule. Best binary result per column in **bold**; second-best underlined. SADP Two-Way evaluates the two most frequent classes.

Method	GCN	SAGE	H2GCN	LINKX	LogReg	MLP	SADP	SADP-Surr	SADP-2W
NyS-Binary (Ours)	**74.65**	**75.89**	**76.53**	**78.26**	**74.02**	**75.39**	**62.10**	**69.60**	**83.57**
NyS-Binary (Unsupervised)	72.89	73.99	75.19	75.31	70.56	73.18	45.28	59.53	73.73
NodeSig	70.68	71.54	71.56	70.72	64.05	65.98	35.29	40.31	64.34
NodeSketch	66.73	68.12	68.00	72.58	51.08	52.47	30.47	29.57	61.67
node2binary	57.41	59.88	57.52	72.64	44.68	46.69	20.66	44.51	57.51
Bi-GCN (attr.)	74.03	75.28	74.60	74.88	71.70	72.08	52.65	55.83	77.34
Raw Features (attr.)	76.01	76.04	74.02	72.22	68.09	69.45	29.13	50.76	58.70
Raw Features Bin (attr.)	71.01	69.40	67.14	68.14	53.90	51.48	26.35	51.47	58.50

ure 3 plots the two axes together: NyS-Binary sits in the desirable upper-left region, and Figure 4 shows the same latencies as bars on a log scale.

Table 2: Embedding generation latency across ten datasets (30 runs per method). NyS-Binary generates full 250-bit codes in about half a second; NodeSig is faster but much less accurate, while learned and recursive methods are far slower.

Method	Mean ± Std (s)	Median (s)	Min (s)	Max (s)	vs. NyS-Binary
NodeSig	0.016 ± 0.010	0.014	0.006	0.041	0.04× (fastest)
NyS-Binary (Unsupervised)	0.317 ± 0.203	0.342	0.071	0.702	0.82× (faster)
NyS-Binary (Ours)	$\mathbf{0.384 \pm 0.363}$	**0.335**	**0.033**	**1.245**	**1.00**× (baseline)
Bi-GCN	5.351 ± 3.492	4.677	1.290	11.139	13.9× slower
NodeSketch	16.543 ± 9.026	15.166	8.164	33.848	43.1× slower
node2binary	1739.9 ± 281.2	1821.9	972.3	1946.5	4530.9× slower

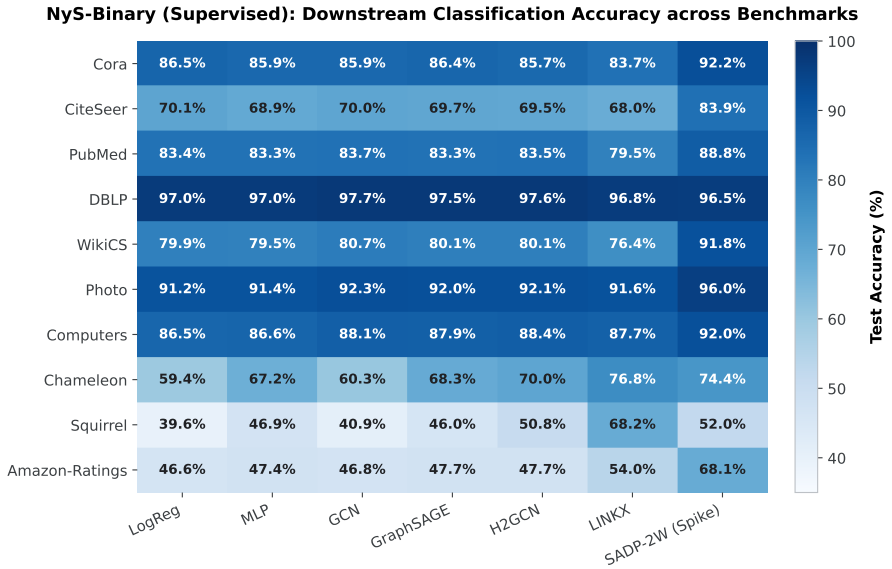


Figure 2: NyS-Binary accuracy per dataset and classifier. Homophilous graphs (top rows) are handled well across the board; heterophilous graphs (bottom rows) are harder, especially for the spiking classifier.

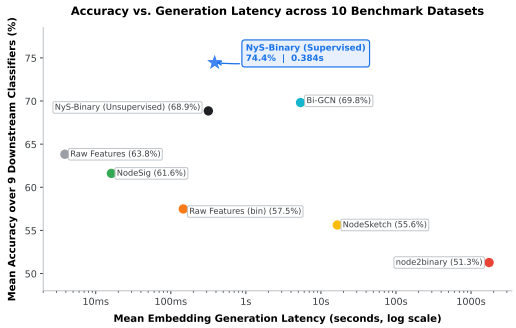


Figure 3: Accuracy vs. generation latency (log scale).

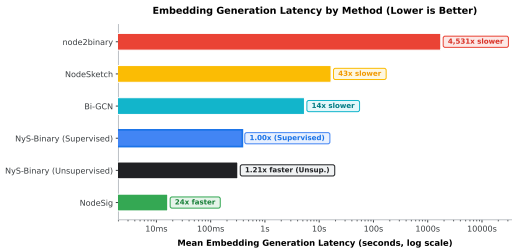


Figure 4: Generation time by method (log scale).

5 Analysis and Discussion

5.1 Embedding Geometry

We quantify separability with the class-separation ratio $R = D_{\text{inter}}/D_{\text{intra}}$, the ratio of mean squared between-class to within-class distances, where $R = 1$ indicates no separation. Figure 5 shows NyS-Binary produces the best-separated codes on homophilous graphs, with R well above 2 on several, and that R falls toward 1 on the strongly heterophilous graphs, tracking accuracy trends and explaining where the method is and is not strong.

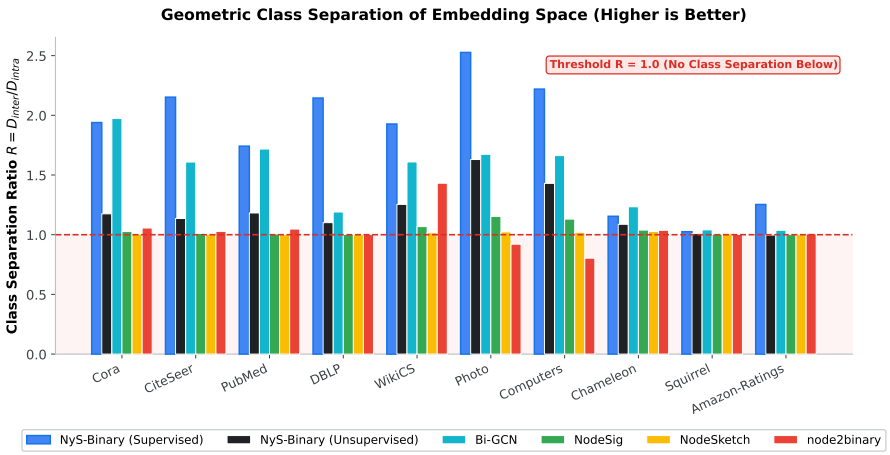


Figure 5: Class-separation ratio $R = D_{\text{inter}}/D_{\text{intra}}$ per dataset. Higher is better; the dashed line marks $R = 1$.

5.2 Component Contributions and Dataset-Dependent Behavior

Two design choices drive effectiveness: the Nyström-inspired landmark projection projects the graph transition columns through the landmark block, while the label-leakage-safe semantic channel injects task-specific class signals. Comparing NyS-Binary (Ours) against NyS-Binary (Unsupervised) in Table 1 isolates the exact quantitative contribution: adding sparse label diffusion yields a consistent $+1.76\%$ to $+1.90\%$ gain across message-passing GNNs (GCN, GraphSAGE) and $+2.21\%$ to $+3.46\%$ on non-relational probes (MLP, LogReg), while extending gradient-free neuromorphic learning (SADP) by $+16.82\%$.

Importantly, our ablation studies (Appendix B) show that the unsupervised backbone alone (struct_ratio $= 1.0$) provides a highly potent representation surpassing all prior binary hashing baselines. Different information sources contribute differently depending on network topology: on strongly homophilous graphs (DBLP, PubMed), semantic diffusion provides significant refinement, whereas on heterophilic graphs (Section A.6), the unsupervised structural backbone excels by avoid-

ing discordant label contamination. Crucially, binary quantization preserves geometric separability without accuracy collapse.

5.3 Limitations

We state the boundaries of the method plainly. On graphs with strong heterophily, while NyS-Binary remains competitive with prior binary hashing baselines, performance margins are more moderate than on homophilous networks, and continuous attribute-aware models can have an advantage when node attributes carry the primary discriminative signal. Under non-homophilous edges, our unsupervised variant (struct_ratio $= 1.0$) is often preferable as it relies purely on topological landmark geometry without diffusing neighbor labels (further analyzed in Appendix A.6). Investigating adaptive or signed polynomial filters for heterophilous topologies, evaluating degree-stratified landmark sampling, and scaling to web-scale networks (10M+ nodes) remain fruitful directions for future work.

6 Conclusion

NyS-Binary is an algebraic pipeline that constructs compact, discriminative binary node codes by sketching a graph's transition operator with a randomized Nyström-inspired approximation and fusing in a label-leakage-safe label channel. Across ten benchmarks and nine classifiers it delivers the best average accuracy among binary methods at sub-second construction cost and $32\times$ memory reduction, transferring seamlessly even to neuromorphic spiking probes. Furthermore, our fully unsupervised variant demonstrates that Nyström-inspired landmark sketching and polynomial diffusion alone establish a new state of the art for label-free, feature-free graph hashing. Principled algebraic sketching of graph operators provides a strong, practical default when speed and memory are the binding constraints.

LLM Usage Disclosure

Large language models (LLMs) were consulted during algorithmic refinement, used to structure portions of the benchmarking code, and used to polish manuscript phrasing. The scientific formulations, algorithmic decisions, experiments, and conclusions were developed, verified, and validated by the authors, who take full responsibility for the contents of this paper.

Data and Code Reproducibility

To ensure full reproducibility, the official, completely anonymized source code and benchmark datasets are publicly available at `https://github.com/ideas-are-bulletproof/Nys-Binary`. The repository provides a fully reproducible environment for generating the binary node embeddings and executing the 10-dataset benchmarks:

- **Core Codebase:** Includes the primary data loaders (`benchmark_new_algebraic.py`), external baseline implementations (`advanced_baselines.py`), and a highly configurable evaluation engine (`rigorous_ablation_system.py`).
- **Reproducible Experiments:** Provides `Main.ipynb` to seamlessly recreate the primary baseline benchmark pipeline, and `Comprehensive_Ablation_Study.ipynb` to reproduce the Nyström landmark scaling Pareto frontiers and hyperparameter sensitivity suites.

References

Abdulkadir Çelikkanat, Fragkiskos D. Malliaros, and Apostolos N. Papadopoulos. NodeSig: Binary node embeddings via random walk diffusion. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 68–75. IEEE, 2022.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 380–388, 2002.

Petros Drineas and Michael W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 855–864, 2016.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Qian Huang, Horace He, Abhay Singh, SerNam Lim, and Jure Leskovec. Combining label propagation and simple models out-performs graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2021.

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

Gouri Lakshmi S, Athira Chandrasekharan, Harshit Kumar, Muhammed Sahad E, Bikas C Das, and Saptarshi Bej. Building supervision into hebbian plasticity through spike agreement. *arXiv preprint arXiv:2601.08526*, 2026.

Ping Li and Arnd Christian König. b-bit minwise hashing. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pp. 671–680. ACM, 2010.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and SerNam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

Péter Mernyei and Catalina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 701–710, 2014.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *International Conference on Learning Representations (ICLR)*, 2023.

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

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *Relational Representation Learning Workshop, NeurIPS*, 2018.

Niloy Talukder, Croix Gyurek, and Mohammad Al Hasan. node2binary: Compact graph node embeddings using binary vectors. In *Proceedings of the ACM Web Conference (WWW)*. ACM, 2025.

Junfu Wang, Yunhong Wang, Zhen Yang, Liang Yang, and Yuanfang Guo. Bi-GCN: Binary graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1561–1570, 2021.

Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 682–688, 2001.

Wei Wu, Bin Li, Ling Chen, and Chengqi Zhang. Efficient attributed network embedding via recursive randomized hashing. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2861–2867, 2018.

Dingqi Yang, Paolo Rosso, Bin Li, and Philippe Cudré-Mauroux. NodeSketch: Highly-efficient graph embeddings via recursive sketching. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1162–1172. ACM, 2019.

Jaewon Yang and Jure Leskovec. Defining and evaluating ground-truth communities in networks. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM)*, pp. 745–754. IEEE, 2012.

Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 2778–2786, 2018.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

A Dataset Summary Statistics

Table 3 summarizes the key topological characteristics of the primary benchmark graph datasets used across our experiments. These include seven homophilous and three heterophilous networks spanning diverse domains such as citation networks, co-authorship graphs, co-purchase graphs, and web-page hyperlink networks. The table details the number of nodes, edges, target classes, and the edge homophily ratio for each graph, which directly influence the behavior of the structural and semantic diffusion mechanisms in NyS-Binary.

Table 3: Summary statistics of the primary benchmark graph datasets evaluated across our experiments, detailing the number of nodes (N), edges (E), target classes (C), edge homophily ratio ($h = \frac{|\{(u,v)\in\mathcal{E}:y_u=y_v\}|}{|\mathcal{E}|}$), and network domain, spanning seven homophilous and three heterophilous topologies. For *Chameleon* and *Squirrel*, we utilize the standard Geom-GCN version and splits (Pei et al., 2020) (with 31,371 and 198,353 directed edges respectively), maintaining these specific versions for direct comparability with prior hashing literature despite their known duplicate-node overlaps (Platonov et al., 2023).

Dataset	**Domain**	**Nodes** (N)	**Edges** (E)	**Classes** (C)	**Homophily** (h)	**Topology**
Cora	Citation network	2,708	5,278	7	0.81	Homophilous
CiteSeer	Citation network	3,327	4,552	6	0.74	Homophilous
PubMed	Citation network	19,717	44,324	3	0.80	Homophilous
WikiCS	Hyperlink network	11,701	215,863	10	0.65	Homophilous
DBLP	Co-authorship graph	13,326	34,281	2	0.83	Homophilous
Photo	Amazon co-purchase	7,650	119,081	8	0.83	Homophilous
Computers	Amazon co-purchase	13,752	245,861	10	0.78	Homophilous
Chameleon	Wikipedia page-page	2,277	31,371	5	0.23	Heterophilous
Squirrel	Wikipedia page-page	5,201	198,353	5	0.22	Heterophilous
Amazon-Ratings	Product review graph	24,492	93,050	5	0.38	Heterophilous

A.1 Memory and Storage Footprint Analysis

Table 4 quantifies the physical memory footprint when node representations are stored under three precision regimes: single-precision floating-point (Float32, 4 bytes per dimension), 8-bit unsigned integer (Uint8, 1 byte per dimension), and fully bit-packed binary arrays (1 bit per dimension). We report these metrics for every binary hashing method; the continuous baseline (Raw Features) is listed for reference under Float32 only, since it cannot be losslessly reduced to a 1-bit representation.

Binary hashing methods (NyS-Binary, NodeSig, NodeSketch, Bi-GCN, node2binary) natively produce $\{0, 1\}$-valued embeddings and therefore admit direct bit-packing. A 250-dimensional binary code stored as a packed 1-bit array occupies only $\lceil 250/8 \rceil = 32$ bytes per node, yielding a $32\times$ compression over the same code stored in Float32. When compared against high-dimensional raw input features (e.g., the 3,703-dimensional TF-IDF vectors in CiteSeer at 46.99 MB), the packed binary embedding compresses storage by up to $462.9\times$, saving 99.8% of space. Note that for datasets whose raw feature dimensionality is lower than the chosen embedding dimension $k = 250$ (e.g., DBLP with $d = 128$), the Float32 embedding matrix is nominally larger than the raw features; however, once bit-packed, the 250-bit binary code still requires only 0.407 MB versus the raw 6.507 MB, a $16\times$ net compression.

Table 4: Memory footprint and storage savings across benchmark datasets. Each dataset section lists the continuous baseline (Raw Features) under Float32 only, and binary hashing methods under Float32, Uint8, and packed 1-bit representations. Compression is measured as the ratio of raw feature size to packed 1-bit embedding size.

Method	Shape	F32 (MB)	U8 (MB)	1-Bit (KB)	Comp.
Cora ($N = 2{,}708$, $d = 1{,}433$, Raw: 14.80 MB)					
Raw Features	2,708 × 1,433	14.803	-	-	1.0×
Raw Features Bin	2,708 × 1,433	14.803	3.701	476.0	31.8×
NyS-Binary (Sup.)	2,708 × 250	2.583	0.646	84.6	179.1×
NyS-Binary (Unsup.)	2,708 × 250	2.583	0.646	84.6	179.1×
NodeSig	2,708 × 250	2.583	0.646	84.6	179.1×
NodeSketch	2,708 × 250	2.583	0.646	84.6	179.1×
Bi-GCN	2,708 × 250	2.583	0.646	84.6	179.1×
node2binary	2,708 × 250	2.583	0.646	84.6	179.1×
CiteSeer ($N = 3{,}327$, $d = 3{,}703$, Raw: 47.00 MB)					
Raw Features	3,327 × 3,703	46.997	-	-	1.0×
Raw Features Bin	3,327 × 3,703	46.997	11.749	1,504.3	32.0×
NyS-Binary (Sup.)	3,327 × 250	3.173	0.793	104.0	462.9×
NyS-Binary (Unsup.)	3,327 × 250	3.173	0.793	104.0	462.9×
NodeSig	3,327 × 250	3.173	0.793	104.0	462.9×
NodeSketch	3,327 × 250	3.173	0.793	104.0	462.9×
Bi-GCN	3,327 × 250	3.173	0.793	104.0	462.9×
node2binary	3,327 × 250	3.173	0.793	104.0	462.9×
PubMed ($N = 19{,}717$, $d = 500$, Raw: 37.61 MB)					
Raw Features	19,717 × 500	37.607	-	-	1.0×
Raw Features Bin	19,717 × 500	37.607	9.402	1,213.1	31.7×
NyS-Binary (Sup.)	19,717 × 250	18.804	4.701	616.2	62.5×
NyS-Binary (Unsup.)	19,717 × 250	18.804	4.701	616.2	62.5×
NodeSig	19,717 × 250	18.804	4.701	616.2	62.5×
NodeSketch	19,717 × 250	18.804	4.701	616.2	62.5×
Bi-GCN	19,717 × 250	18.804	4.701	616.2	62.5×
node2binary	19,717 × 250	18.804	4.701	616.2	62.5×
WikiCS ($N = 11{,}701$, $d = 300$, Raw: 13.39 MB)					
Raw Features	11,701 × 300	13.391	-	-	1.0×
Raw Features Bin	11,701 × 300	13.391	3.348	434.2	31.6×
NyS-Binary (Sup.)	11,701 × 250	11.159	2.790	365.7	37.5×
NyS-Binary (Unsup.)	11,701 × 250	11.159	2.790	365.7	37.5×

Continued on next page

Table 4 - continued from previous page

Method	Shape	F32 (MB)	U8 (MB)	1-Bit (KB)	Comp.
NodeSig	11,701 × 250	11.159	2.790	365.7	37.5×
NodeSketch	11,701 × 250	11.159	2.790	365.7	37.5×
Bi-GCN	11,701 × 250	11.159	2.790	365.7	37.5×
node2binary	11,701 × 250	11.159	2.790	365.7	37.5×
Chameleon ($N = 2{,}277$, $d = 2{,}325$, Raw: 20.20 MB)					
Raw Features	2,277 × 2,325	20.195	-	-	1.0×
Raw Features Bin	2,277 × 2,325	20.195	5.049	647.1	32.0×
NyS-Binary (Sup.)	2,277 × 250	2.172	0.543	71.2	290.6×
NyS-Binary (Unsup.)	2,277 × 250	2.172	0.543	71.2	290.6×
NodeSig	2,277 × 250	2.172	0.543	71.2	290.6×
NodeSketch	2,277 × 250	2.172	0.543	71.2	290.6×
Bi-GCN	2,277 × 250	2.172	0.543	71.2	290.6×
node2binary	2,277 × 250	2.172	0.543	71.2	290.6×
Squirrel ($N = 5{,}201$, $d = 2{,}089$, Raw: 41.45 MB)					
Raw Features	5,201 × 2,089	41.446	-	-	1.0×
Raw Features Bin	5,201 × 2,089	41.446	10.362	1,330.7	31.9×
NyS-Binary (Sup.)	5,201 × 250	4.960	1.240	162.5	261.1×
NyS-Binary (Unsup.)	5,201 × 250	4.960	1.240	162.5	261.1×
NodeSig	5,201 × 250	4.960	1.240	162.5	261.1×
NodeSketch	5,201 × 250	4.960	1.240	162.5	261.1×
Bi-GCN	5,201 × 250	4.960	1.240	162.5	261.1×
node2binary	5,201 × 250	4.960	1.240	162.5	261.1×
Photo ($N = 7{,}650$, $d = 745$, Raw: 21.74 MB)					
Raw Features	7,650 × 745	21.741	-	-	1.0×
Raw Features Bin	7,650 × 745	21.741	5.435	702.2	31.7×
NyS-Binary (Sup.)	7,650 × 250	7.296	1.824	239.1	93.1×
NyS-Binary (Unsup.)	7,650 × 250	7.296	1.824	239.1	93.1×
NodeSig	7,650 × 250	7.296	1.824	239.1	93.1×
NodeSketch	7,650 × 250	7.296	1.824	239.1	93.1×
Bi-GCN	7,650 × 250	7.296	1.824	239.1	93.1×
node2binary	7,650 × 250	7.296	1.824	239.1	93.1×
Computers ($N = 13{,}752$, $d = 767$, Raw: 40.24 MB)					
Raw Features	13,752 × 767	40.237	-	-	1.0×
Raw Features Bin	13,752 × 767	40.237	10.059	1,289.2	32.0×
NyS-Binary (Sup.)	13,752 × 250	13.115	3.279	429.8	95.9×
NyS-Binary (Unsup.)	13,752 × 250	13.115	3.279	429.8	95.9×
NodeSig	13,752 × 250	13.115	3.279	429.8	95.9×
NodeSketch	13,752 × 250	13.115	3.279	429.8	95.9×
Bi-GCN	13,752 × 250	13.115	3.279	429.8	95.9×
node2binary	13,752 × 250	13.115	3.279	429.8	95.9×
Amazon-Ratings ($N = 24{,}492$, $d = 300$, Raw: 28.03 MB)					
Raw Features	24,492 × 300	28.029	-	-	1.0×
Raw Features Bin	24,492 × 300	28.029	7.007	908.9	31.6×
NyS-Binary (Sup.)	24,492 × 250	23.357	5.839	765.4	37.5×
NyS-Binary (Unsup.)	24,492 × 250	23.357	5.839	765.4	37.5×
NodeSig	24,492 × 250	23.357	5.839	765.4	37.5×
NodeSketch	24,492 × 250	23.357	5.839	765.4	37.5×
Bi-GCN	24,492 × 250	23.357	5.839	765.4	37.5×
node2binary	24,492 × 250	23.357	5.839	765.4	37.5×
DBLP ($N = 13{,}326$, $d = 128$, Raw: 6.51 MB)					
Raw Features	13,326 × 128	6.507	-	-	1.0×
Raw Features Bin	13,326 × 128	6.507	1.627	208.2	32.0×
NyS-Binary (Sup.)	13,326 × 250	12.709	3.177	416.4	16.0×
NyS-Binary (Unsup.)	13,326 × 250	12.709	3.177	416.4	16.0×

Continued on next page

Table 4 - continued from previous page

Method	Shape	F32 (MB)	U8 (MB)	1-Bit (KB)	Comp.
NodeSig	13,326 × 250	12.709	3.177	416.4	16.0×
NodeSketch	13,326 × 250	12.709	3.177	416.4	16.0×
Bi-GCN	13,326 × 250	12.709	3.177	416.4	16.0×
node2binary	13,326 × 250	12.709	3.177	416.4	16.0×

A.2 Downstream Classifier Architectures and Hyperparameter Configurations

To rigorously evaluate representation quality across diverse operational modalities, we evaluate nine downstream probe classifiers across all benchmarks. These encompass standard message-passing graph neural networks, heterophily-tailored architectures, non-relational linear and feedforward probes, and biologically plausible neuromorphic spiking networks. Below we specify their architectural mechanics and configuration hyperparameters:

A.2.1 Message-Passing Graph Neural Networks

- **Graph Convolutional Network (GCN) (Kipf & Welling, 2017):** The model applies two successive graph convolutional layers with symmetrically normalized adjacency aggregation:

$$\mathbf{Z}^{(1)} = \mathrm{ReLU}\big(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\,\mathbf{X}\,\mathbf{W}^{(0)}\big), \quad \hat{\mathbf{Y}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\,\mathbf{Z}^{(1)}\,\mathbf{W}^{(1)}, \tag{6}$$

  where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$. Layer dimensions are $\dim_{\text{in}} \to 64 \to C$. Optimization uses Adam with learning rate $\eta = 0.01$, weight decay $\lambda = 5 \times 10^{-4}$, and cross-entropy loss trained for 200 epochs.

- **GraphSAGE (Hamilton et al., 2017):** The architecture uses two inductive mean-aggregation convolutional layers:

$$\begin{aligned} \mathbf{h}_v^{(1)} &= \mathrm{ReLU}\Big(\mathbf{W}_1 \cdot \mathrm{CONCAT}\Big(\mathbf{x}_v,\ \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)}\mathbf{x}_u\Big)\Big), \\ \hat{\mathbf{y}}_v &= \mathbf{W}_2 \cdot \mathrm{CONCAT}\Big(\mathbf{h}_v^{(1)},\ \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)}\mathbf{h}_u^{(1)}\Big). \end{aligned} \tag{7}$$

  Hidden dimension is 64 with ReLU activation. Optimized via Adam ($\eta = 0.01, \lambda = 5 \times 10^{-4}$) for 200 epochs.

A.2.2 Heterophily-Tailored Architectures

- **LINKX (Lim et al., 2021):** Designed for graphs with strong heterophily, LINKX decouples feature representations and topological structure to bypass oversmoothing. The model comprises:
  (i) A 2-layer node MLP ($\dim_{\text{in}} \to 64 \to 64$) processing the node feature representations $\mathbf{X}$;
  (ii) A 2-layer edge MLP ($N \to 64 \to 64$) processing the sparse graph adjacency matrix $\mathbf{A}$;
  (iii) A 2-layer fusion MLP ($128 \to 64 \to C$) combining the node and edge channels with dropout $p = 0.5$.

  All internal channels are set to 64. Trained using Adam ($\eta = 0.01, \lambda = 5 \times 10^{-4}$) for 200 epochs.

- **H2GCN (Zhu et al., 2020):** This architecture captures higher-order neighborhood topologies under heterophily:
  (i) An initial feature embedding layer projects input representations to hidden dimension 64: $\mathbf{X}^{(0)} = \mathrm{ReLU}(\mathbf{X}\mathbf{W}_0)$;
  (ii) Two sequential propagation steps aggregate normalized neighbors without self-loops: $\mathbf{X}^{(1)} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{X}^{(0)}$ and $\mathbf{X}^{(2)} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{X}^{(1)}$;
  (iii) Ego and neighborhood signals are concatenated across scales: $\mathbf{X}_{\text{concat}} = [\mathbf{X}^{(0)} \,\|\, \mathbf{X}^{(1)} \,\|\, \mathbf{X}^{(2)}] \in \mathbb{R}^{N\times 192}$;
  (iv) A final linear projection maps the 192-dimensional multi-scale representation to target classes: $\hat{\mathbf{Y}} = \mathbf{X}_{\text{concat}}\mathbf{W}_{\text{out}}$.

  Trained with Adam ($\eta = 0.01, \lambda = 5 \times 10^{-4}$) for 200 epochs.

A.2.3 Non-Relational and Linear Probes

- **Multilayer Perceptron (MLP):** This probe comprises two dense linear layers: $\mathbf{Z} = \mathrm{ReLU}(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1)$ followed by $\hat{\mathbf{Y}} = \mathbf{Z}\mathbf{W}_2 + \mathbf{b}_2$ (dimensions $\mathrm{dim}_{\mathrm{in}} \to 64 \to C$). Critically, the MLP operates purely on node embeddings without access to graph edges, isolating the intrinsic linear and non-linear separability of the representations. Trained with Adam ($\eta = 0.01, \lambda = 5 \times 10^{-4}$) for 200 epochs.
- **Logistic Regression:** This constitutes a strictly convex, parameter-free linear probe ($\hat{\mathbf{Y}} = \mathbf{X}\mathbf{W} + \mathbf{b}$). By eschewing hidden representations and non-linearities, it directly measures linear decodability and geometric class separation in Hamming space. Trained with Adam ($\eta = 0.01, \lambda = 5 \times 10^{-4}$) for 200 epochs.

A.3 Spiking Classifier Architectures and Graph Adaptations

In this section, we describe the neuromorphic spiking neural network classifiers evaluated throughout our experiments, specifically detailing the exact architectural adaptations made to extend the original Spiking Agreement-Driven Plasticity (SADP) model to graph topologies, the mechanics of the two-way variant, and the surrogate-gradient baseline.

A.3.1 Graph Adaptation of Spiking Agreement-Driven Plasticity (SADP)

The original SADP formulation (Lakshmi S et al., 2026) was developed for non-relational, Euclidean data (e.g., vision and tabular benchmarks) using a standard multilayer perceptron (MLP) topology. To evaluate our binary representations in a biologically plausible, gradient-free neuromorphic setting on relational data, we extended the original model to graph topologies through two fundamental adaptations:

1. **Spiking Message Passing Forward Dynamics:** In the original MLP formulation, incoming Poisson spike trains $\mathbf{S}_{\mathrm{in}}[t]$ are projected directly via dense synaptic weights $\mathbf{I}_h[t] = \mathbf{S}_{\mathrm{in}}[t]\mathbf{W}_1$. In our graph extension, we substitute this with spiking neighborhood message passing:

$$\mathbf{I}_h[t] = \left(\hat{\mathbf{A}}\,\mathbf{S}_{\mathrm{in}}[t]\right)\mathbf{W}_1, \quad \text{where } \hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}. \tag{8}$$

   Because the message aggregation is linear, neighborhood spikes are accumulated across graph edges prior to the synaptic transformation. A two-layer message-passing core is employed, while the final class readout remains a purely local, per-node linear transformation without graph aggregation ($\mathbf{I}_o[t] = \mathbf{S}_{h2}[t]\mathbf{W}_2$). Membrane potentials obey discrete-time leaky integrate-and-fire (LIF) dynamics with reset-by-subtraction, and class predictions are decoded from accumulated output spike counts.

2. **Presynaptic Aggregation in Agreement Plasticity:** The core SADP synaptic update is driven by Cohen's kappa agreement κ between post-synaptic spike trains and supervisory teacher spike trains generated from training labels, avoiding global backpropagation. In our graph extension, the presynaptic factor in the outer-product update is adapted to match the aggregated input actually experienced by the synapses:

$$\Delta\mathbf{W}_1 \propto \sum_{d=-k_{\mathrm{shift}}}^{k_{\mathrm{shift}}} \left(\hat{\mathbf{A}}\mathbf{X}\right)^{\top}\boldsymbol{\kappa}_d. \tag{9}$$

   This substitution accumulates shared-weight contributions across edges for each node. Crucially, the temporal shift parameter $d \in [-k_{\mathrm{shift}}, k_{\mathrm{shift}}]$ in the Cohen's kappa calculation naturally models multi-hop propagation delays: spikes traversing multi-hop neighborhood paths arrive with temporal latency, and the shifted agreement window aligns synaptic updates with these delayed causal arrivals.

A.3.2 Two-Way SADP Classification (SADP Two-Way)

To evaluate the intrinsic linear separability of binary node codes in a neuromorphic substrate without the confounding effects of multi-class competition, we introduced the two-way variant:

- **Subpopulation and Class Filtering:** Rather than training an output neuron for every class in a C-class dataset, the model isolates the **top-two most frequent classes** in the ground truth: $\mathcal{C}_{\text{top2}} = \{c_{(1)}, c_{(2)}\}$. All nodes belonging to any other class are masked out from both the training and test splits. The remaining node labels are remapped to binary targets $\{0, 1\}$.
- **Network Configuration:** The output layer is instantiated with exactly two LIF output neurons, maintaining the identical two-layer spiking message-passing core and gradient-free Cohen's kappa learning rule.
- **Analytical Rationale for Performance Gap:** In multi-class spiking networks with gradient-free learning, lateral inhibition and uncoordinated spike collisions across many competing output neurons often induce class confusion (particularly on benchmarks with 7 to 40 classes). Restricting evaluation to the two dominant classes eliminates multi-class interference, directly testing whether the binary Hamming codes form cleanly separable clusters under biologically plausible spike dynamics. This explains why SADP Two-Way achieves substantially higher accuracy across all methods (reaching 83.57% on NyS-Binary, compared to 62.10% for full multi-class SADP).

A.3.3 Surrogate-Gradient Spiking Network Baseline (SADP Surrogate)

To directly benchmark gradient-free agreement-driven learning against standard gradient-based optimization in spiking networks, we implemented a surrogate-gradient baseline (Neftci et al., 2019):

- **Matched Forward Pass:** Employs the identical spiking message-passing architecture, Poisson feature encoding, and LIF membrane dynamics as the graph SADP model, ensuring that any performance gap is strictly attributable to the learning rule rather than the network capacity.
- **Surrogate Derivative Substitution:** During the backward pass, the non-differentiable Dirac delta derivative of the Heaviside spike function is replaced by a smooth sigmoid pseudo-derivative $\sigma'(u) = \beta s(1 - s)$, where $s = 1/(1 + \exp(-\beta u))$ with sharpness $\beta = 10.0$.
- **BPTT Optimization:** The model accumulates output logits via a leaky-integrator readout and optimizes masked cross-entropy via Backpropagation Through Time (BPTT) using the Adam optimizer.

A.4 External Baselines, Adaptations, and Embedding Generators

To place NyS-Binary in context, we benchmarked against leading binary graph hashing algorithms, continuous spectral references, and structural backbone variants. Below, we summarize each baseline and explicitly document where our evaluation protocol or implementation adapted the original author-published procedures:

A.4.1 External Binary Graph Hashing Baselines

1. **Bi-GCN (Wang et al., 2021) (CVPR 2021 / TPAMI 2024):** A learned binary graph convolutional network that constrains both network weights and node representations to $\{-1, +1\}$ using $\text{sign}(x)$, trained via the straight-through estimator (STE) on the raw continuous node features $\mathbf{X}$.
   - *Adaptation from Original Paper:* In the original work, Bi-GCN was designed strictly as an end-to-end task-specific classifier that outputs class prediction logits directly. For our benchmark, we trained the two-layer Bi-GCN model on the training mask using the graph's actual continuous node attributes, and **extracted the intermediate binarized hidden activations** ($\mathbf{H}_2 \in \{-1, +1\}^{N \times K}$, mapped to $\{0, 1\}^{N \times K}$) as a static binary node embedding. This representation was then evaluated across all nine downstream classifiers for a consistent, like-for-like comparison against the feature-free binary hashing baselines.
2. **node2binary (Talukder et al., 2025) (WWW 2025):** An unsupervised binary embedding framework that combines Leiden community detection with random-walk co-occurrence modeling to build a hierarchical tree of binary codes where shared prefixes reflect community membership.
   - *Evaluation Setup:* We executed the author's official, unmodified implementation with default hyper-parameters (Leiden trees=1, depth=3) to generate bitcodes purely from graph topology, exactly as described in the original paper. Because the author's script bundles internal downstream classification with embedding generation, we extracted the generated unsupervised

binary vectors directly and evaluated them across our standardized 70/30 stratified train/test splits across all nine downstream classifiers, ensuring a fair and uniform benchmark.

3. **NodeSig (Çelikkanat et al., 2022) (IEEE/ACM ASONAM 2022):** A shallow, non-parametric binary hashing algorithm that computes multi-hop random-walk transition probability vectors and projects them onto Hamming space via random hyperplane sign hashing.
   - *Adaptation from Original Paper:* We executed the author's official compiled C++ executable. Because the C++ implementation packs bits into 8-bit bytes, it strictly requires the embedding dimension to be a multiple of 8. To maintain an identical $K = 250$ bit budget across all methods, we generated $\lceil K/8 \rceil \times 8 = 256$ bits and truncated the resulting representation to exactly $K = 250$ bits.
4. **NodeSketch (Yang et al., 2019) (ACM SIGKDD 2019):** A discrete locality-sensitive hashing (ALSH) method that sketches multi-hop decay-weighted neighborhood distributions via recursive Min-Hash.
   - *Adaptation from Original Paper:* The author's official MATLAB/C MEX implementation outputs integer-valued min-hash sketch codes ($\mathbf{H} \in \mathbb{N}^{N \times K}$). To convert these into compact binary bitcodes suitable for Hamming-distance computation and binary downstream classifiers without violating the locality-sensitive hashing property, we adopt the established **b-bit minwise hashing** framework (Li & König, 2010). Specifically, we extract the least significant bit (LSB) of each min-hash signature: $\mathbf{B} = (\mathbf{H} \bmod 2 == 1) \in \{0, 1\}^{N \times K}$. As proven by Li & König (2010), retaining only the lowest-order bit preserves the underlying Jaccard similarity in Hamming space with provable collision probabilities ($P(b_i = b_j) = \frac{1+J}{2}$), providing a mathematically grounded binary representation for recursive min-hash signatures.

A.4.2 Continuous and Attribute-Aware Reference Baselines

1. **Raw Features & Raw Features Binarized:** Attribute-aware references that directly feed raw continuous node feature vectors (or median-binarized bitcodes) into downstream classifiers, establishing the performance ceiling when node attributes are available.

A.5 Per-dataset results

Table 5 give a comprehensive per-dataset view across all ten benchmarks. The pattern is consistent: NyS-Binary dominates on homophilous graphs across nearly all classifiers, stays close to the best on PubMed, and is competitive but no longer dominant on heterophilous graphs like Squirrel, where structural signal is weaker. The heatmap in Figure 2 shows the full per-dataset, per-classifier accuracy of NyS-Binary at a glance.

Table 5: Comprehensive per-dataset classification accuracy ± std (%) across benchmark datasets (Part 1: Amazon-Ratings, Chameleon, CiteSeer, Computers, and Cora) evaluated across nine downstream probe classifiers. Feature-free binary hashing methods are grouped above the inner rule; attribute-aware baselines are shown below. Best binary result per column in **bold**; second-best underlined.

Method	GCN	SAGE	H2GCN	LINKX	LogReg	MLP	SADP	SADP-Surr	SADP-2W
Amazon-Ratings (Heterophilous)									
NyS-Binary (Ours)	**46.80**$_{\pm 0.37}$	**47.67**$_{\pm 0.81}$	**47.68**$_{\pm 0.23}$	**53.98**$_{\pm 0.85}$	**46.58**$_{\pm 0.40}$	**47.42**$_{\pm 0.63}$	**44.38**$_{\pm 0.75}$	**39.77**$_{\pm 8.10}$	**68.14**$_{\pm 0.83}$
NyS-Binary (Unsupervised)	45.34$_{\pm 0.93}$	45.15$_{\pm 0.21}$	45.77$_{\pm 0.64}$	53.83$_{\pm 0.28}$	42.22$_{\pm 0.66}$	45.22$_{\pm 0.96}$	32.44$_{\pm 1.60}$	36.90$_{\pm 0.10}$	57.38$_{\pm 0.82}$
NodeSig	40.65$_{\pm 0.77}$	39.83$_{\pm 1.13}$	41.22$_{\pm 1.13}$	52.49$_{\pm 0.41}$	37.96$_{\pm 0.47}$	40.14$_{\pm 0.83}$	32.25$_{\pm 0.28}$	36.76$_{\pm 0.04}$	56.76$_{\pm 0.51}$
NodeSketch	39.43$_{\pm 1.47}$	37.31$_{\pm 0.73}$	38.39$_{\pm 1.24}$	52.62$_{\pm 0.38}$	36.28$_{\pm 0.47}$	36.22$_{\pm 1.25}$	32.83$_{\pm 0.20}$	36.55$_{\pm 0.34}$	57.07$_{\pm 0.49}$
node2binary	42.20$_{\pm 0.50}$	42.28$_{\pm 0.52}$	41.57$_{\pm 0.76}$	52.91$_{\pm 0.53}$	37.38$_{\pm 0.63}$	39.57$_{\pm 0.98}$	25.94$_{\pm 0.40}$	33.70$_{\pm 1.05}$	57.17$_{\pm 0.12}$
Bi-GCN	42.30$_{\pm 0.59}$	42.83$_{\pm 0.89}$	42.85$_{\pm 0.44}$	51.26$_{\pm 1.58}$	40.57$_{\pm 0.52}$	40.43$_{\pm 0.39}$	36.52$_{\pm 1.84}$	36.52$_{\pm 0.21}$	58.54$_{\pm 0.34}$
Raw Features	42.22$_{\pm 0.57}$	44.47$_{\pm 0.55}$	43.94$_{\pm 0.36}$	53.60$_{\pm 0.09}$	38.60$_{\pm 0.11}$	40.90$_{\pm 0.20}$	35.72$_{\pm 0.18}$	29.85$_{\pm 6.35}$	57.84$_{\pm 0.03}$
Raw Features Bin	38.36$_{\pm 0.31}$	36.98$_{\pm 0.58}$	36.95$_{\pm 0.35}$	52.09$_{\pm 0.53}$	34.04$_{\pm 0.09}$	30.48$_{\pm 1.63}$	33.62$_{\pm 0.64}$	31.05$_{\pm 4.87}$	57.13$_{\pm 0.41}$
Chameleon (Heterophilous)									
NyS-Binary (Ours)	60.33$_{\pm 0.51}$	68.27$_{\pm 1.80}$	69.98$_{\pm 1.91}$	76.80$_{\pm 1.69}$	59.41$_{\pm 0.34}$	67.15$_{\pm 1.41}$	**31.63**$_{\pm 4.20}$	**45.32**$_{\pm 7.31}$	74.35$_{\pm 5.45}$
NyS-Binary (Unsupervised)	59.94$_{\pm 1.83}$	**68.52**$_{\pm 1.31}$	**73.00**$_{\pm 0.81}$	74.90$_{\pm 5.68}$	**63.55**$_{\pm 0.80}$	**70.76**$_{\pm 1.34}$	26.07$_{\pm 0.99}$	42.84$_{\pm 8.28}$	**75.48**$_{\pm 4.07}$
NodeSig	61.60$_{\pm 1.55}$	67.50$_{\pm 0.88}$	69.44$_{\pm 1.25}$	76.22$_{\pm 2.70}$	61.99$_{\pm 0.39}$	66.71$_{\pm 1.31}$	27.49$_{\pm 0.89}$	31.28$_{\pm 5.85}$	75.03$_{\pm 2.21}$
NodeSketch	**61.94**$_{\pm 1.47}$	67.45$_{\pm 1.36}$	66.18$_{\pm 3.29}$	74.02$_{\pm 1.97}$	59.50$_{\pm 2.01}$	62.72$_{\pm 3.04}$	26.07$_{\pm 5.19}$	24.17$_{\pm 3.57}$	72.09$_{\pm 1.60}$
node2binary	53.17$_{\pm 2.31}$	63.84$_{\pm 4.12}$	53.95$_{\pm 4.04}$	**76.85**$_{\pm 1.55}$	41.66$_{\pm 4.25}$	42.84$_{\pm 3.12}$	24.85$_{\pm 2.94}$	41.18$_{\pm 2.57}$	58.98$_{\pm 4.25}$
Bi-GCN	58.14$_{\pm 1.45}$	67.64$_{\pm 1.39}$	63.74$_{\pm 1.34}$	76.66$_{\pm 0.72}$	58.67$_{\pm 2.33}$	60.82$_{\pm 1.91}$	36.45$_{\pm 2.06}$	35.43$_{\pm 8.55}$	75.60$_{\pm 4.70}$
Raw Features	66.67$_{\pm 1.86}$	60.72$_{\pm 1.88}$	56.04$_{\pm 1.86}$	49.66$_{\pm 2.44}$	47.95$_{\pm 0.44}$	46.74$_{\pm 1.50}$	20.47$_{\pm 0.25}$	66.13$_{\pm 3.14}$	53.22$_{\pm 0.00}$
Raw Features Bin	66.33$_{\pm 2.49}$	57.11$_{\pm 1.86}$	51.80$_{\pm 1.50}$	56.73$_{\pm 3.25}$	43.08$_{\pm 1.50}$	41.37$_{\pm 1.34}$	20.13$_{\pm 0.17}$	68.76$_{\pm 1.10}$	53.22$_{\pm 0.00}$
Citeseer (Homophilous)									
NyS-Binary (Ours)	**70.00**$_{\pm 1.79}$	**69.70**$_{\pm 1.76}$	**69.47**$_{\pm 0.92}$	**67.97**$_{\pm 1.25}$	**70.10**$_{\pm 1.91}$	**68.90**$_{\pm 1.76}$	**66.17**$_{\pm 4.41}$	**69.34**$_{\pm 1.99}$	**83.86**$_{\pm 0.37}$
NyS-Binary (Unsupervised)	61.59$_{\pm 1.66}$	59.63$_{\pm 1.16}$	59.63$_{\pm 2.72}$	56.69$_{\pm 3.49}$	57.63$_{\pm 1.63}$	57.73$_{\pm 2.29}$	37.21$_{\pm 0.95}$	48.41$_{\pm 4.39}$	77.13$_{\pm 1.90}$
NodeSig	61.83$_{\pm 1.71}$	61.49$_{\pm 1.50}$	61.46$_{\pm 1.78}$	39.97$_{\pm 16.26}$	46.22$_{\pm 1.07}$	50.42$_{\pm 0.57}$	19.29$_{\pm 0.70}$	21.49$_{\pm 1.72}$	58.07$_{\pm 3.18}$
NodeSketch	55.22$_{\pm 1.93}$	52.99$_{\pm 2.97}$	55.69$_{\pm 2.64}$	50.52$_{\pm 1.29}$	30.66$_{\pm 1.63}$	33.23$_{\pm 1.15}$	19.79$_{\pm 0.64}$	20.19$_{\pm 2.40}$	56.20$_{\pm 2.99}$
node2binary	51.95$_{\pm 3.39}$	50.02$_{\pm 7.82}$	50.21$_{\pm 5.28}$	36.63$_{\pm 4.96}$	44.71$_{\pm 4.37}$	46.75$_{\pm 3.74}$	23.39$_{\pm 1.40}$	22.86$_{\pm 5.18}$	57.83$_{\pm 4.70}$
Bi-GCN	74.04$_{\pm 1.69}$	72.84$_{\pm 2.22}$	72.14$_{\pm 1.85}$	68.64$_{\pm 2.44}$	72.77$_{\pm 1.65}$	71.34$_{\pm 2.06}$	61.53$_{\pm 4.91}$	71.97$_{\pm 1.56}$	83.21$_{\pm 2.80}$
Raw Features	72.30$_{\pm 1.72}$	75.04$_{\pm 1.35}$	75.04$_{\pm 1.37}$	64.73$_{\pm 2.63}$	71.87$_{\pm 0.92}$	70.60$_{\pm 1.00}$	10.41$_{\pm 0.17}$	69.77$_{\pm 1.06}$	47.20$_{\pm 2.23}$
Raw Features Bin	72.00$_{\pm 1.62}$	74.24$_{\pm 0.87}$	72.70$_{\pm 1.03}$	52.69$_{\pm 6.73}$	64.97$_{\pm 0.63}$	63.36$_{\pm 1.15}$	8.28$_{\pm 0.40}$	69.20$_{\pm 1.23}$	49.55$_{\pm 0.85}$
Computers (Homophilous)									
NyS-Binary (Ours)	88.14$_{\pm 0.21}$	87.86$_{\pm 0.12}$	88.36$_{\pm 0.37}$	**87.71**$_{\pm 0.44}$	**86.47**$_{\pm 0.74}$	**86.58**$_{\pm 0.13}$	**66.66**$_{\pm 2.57}$	**86.63**$_{\pm 1.73}$	**92.03**$_{\pm 2.17}$
NyS-Binary (Unsupervised)	**88.29**$_{\pm 0.93}$	88.20$_{\pm 0.76}$	**88.53**$_{\pm 1.07}$	86.84$_{\pm 0.55}$	84.86$_{\pm 1.16}$	85.43$_{\pm 1.07}$	53.97$_{\pm 2.73}$	77.06$_{\pm 2.37}$	79.96$_{\pm 6.20}$
NodeSig	88.05$_{\pm 0.48}$	**88.67**$_{\pm 0.54}$	88.17$_{\pm 0.27}$	85.08$_{\pm 0.94}$	84.18$_{\pm 0.51}$	84.04$_{\pm 0.94}$	46.64$_{\pm 0.32}$	42.46$_{\pm 1.48}$	73.18$_{\pm 0.26}$
NodeSketch	86.27$_{\pm 0.42}$	86.22$_{\pm 1.03}$	83.46$_{\pm 3.37}$	85.52$_{\pm 0.45}$	62.45$_{\pm 1.52}$	61.85$_{\pm 1.63}$	36.19$_{\pm 0.25}$	34.42$_{\pm 3.32}$	70.54$_{\pm 0.41}$
node2binary	44.05$_{\pm 0.36}$	44.62$_{\pm 0.85}$	42.96$_{\pm 0.50}$	86.19$_{\pm 0.18}$	39.02$_{\pm 0.35}$	39.10$_{\pm 0.41}$	3.33$_{\pm 0.10}$	40.26$_{\pm 8.93}$	70.55$_{\pm 0.14}$
Bi-GCN	84.45$_{\pm 1.48}$	84.61$_{\pm 2.55}$	85.55$_{\pm 1.49}$	68.74$_{\pm 11.43}$	80.72$_{\pm 2.63}$	81.85$_{\pm 2.31}$	41.96$_{\pm 5.69}$	32.41$_{\pm 13.90}$	72.36$_{\pm 1.87}$
Raw Features	90.05$_{\pm 0.30}$	90.31$_{\pm 0.41}$	89.96$_{\pm 0.77}$	87.64$_{\pm 1.58}$	85.89$_{\pm 0.40}$	83.01$_{\pm 3.38}$	46.03$_{\pm 0.10}$	37.46$_{\pm 0.06}$	71.59$_{\pm 0.10}$
Raw Features Bin	90.01$_{\pm 0.25}$	88.31$_{\pm 0.61}$	87.78$_{\pm 0.17}$	85.89$_{\pm 0.86}$	77.61$_{\pm 0.43}$	74.29$_{\pm 0.14}$	39.34$_{\pm 0.56}$	37.08$_{\pm 0.35}$	70.77$_{\pm 0.39}$
Cora (Homophilous)									
NyS-Binary (Ours)	**85.94**$_{\pm 0.72}$	**86.39**$_{\pm 1.71}$	**85.69**$_{\pm 1.52}$	**83.68**$_{\pm 1.07}$	**86.55**$_{\pm 0.80}$	**85.90**$_{\pm 0.60}$	**72.69**$_{\pm 5.02}$	**85.20**$_{\pm 0.75}$	**92.16**$_{\pm 1.55}$
NyS-Binary (Unsupervised)	82.53$_{\pm 1.74}$	81.55$_{\pm 1.86}$	81.30$_{\pm 2.10}$	76.18$_{\pm 2.29}$	78.88$_{\pm 2.46}$	78.84$_{\pm 2.57}$	44.98$_{\pm 4.21}$	66.30$_{\pm 3.09}$	72.37$_{\pm 5.90}$
NodeSig	82.45$_{\pm 0.87}$	80.65$_{\pm 0.47}$	79.38$_{\pm 0.40}$	63.22$_{\pm 9.60}$	70.56$_{\pm 2.05}$	68.10$_{\pm 1.03}$	30.18$_{\pm 0.26}$	29.31$_{\pm 0.79}$	66.75$_{\pm 0.62}$
NodeSketch	79.29$_{\pm 2.03}$	77.04$_{\pm 1.88}$	76.47$_{\pm 2.52}$	67.57$_{\pm 2.48}$	46.00$_{\pm 1.98}$	45.06$_{\pm 1.05}$	23.12$_{\pm 0.44}$	24.35$_{\pm 10.03}$	64.44$_{\pm 1.42}$
node2binary	84.66$_{\pm 1.15}$	83.85$_{\pm 0.63}$	83.52$_{\pm 1.11}$	73.76$_{\pm 2.95}$	79.95$_{\pm 0.86}$	80.69$_{\pm 0.85}$	26.16$_{\pm 2.06}$	35.10$_{\pm 4.15}$	66.40$_{\pm 2.08}$
Bi-GCN	87.33$_{\pm 0.69}$	87.25$_{\pm 0.31}$	86.47$_{\pm 0.25}$	78.80$_{\pm 5.42}$	85.20$_{\pm 0.87}$	83.60$_{\pm 0.38}$	66.58$_{\pm 2.92}$	85.24$_{\pm 1.17}$	85.74$_{\pm 2.86}$
Raw Features	87.12$_{\pm 0.19}$	87.66$_{\pm 0.31}$	86.59$_{\pm 1.01}$	74.09$_{\pm 2.97}$	75.16$_{\pm 1.51}$	75.15$_{\pm 1.91}$	20.62$_{\pm 0.93}$	84.71$_{\pm 0.72}$	65.51$_{\pm 0.54}$
Raw Features Bin	86.27$_{\pm 0.40}$	85.98$_{\pm 0.86}$	83.93$_{\pm 1.00}$	70.36$_{\pm 2.28}$	63.96$_{\pm 1.17}$	61.26$_{\pm 1.94}$	15.99$_{\pm 0.57}$	84.67$_{\pm 0.14}$	65.87$_{\pm 0.15}$

Table 5: Comprehensive per-dataset classification accuracy ± std (%) across benchmark datasets (Part 2: DBLP, Photo, PubMed, Squirrel, and WikiCS; continued from Table 5). Best binary result per column in **bold**; second-best underlined.

Method	GCN	SAGE	H2GCN	LINKX	LogReg	MLP	SADP	SADP-Surr	SADP-2W
Dblp (Homophilous)									
NyS-Binary (Ours)	**97.66**$_{\pm 0.38}$	**97.55**$_{\pm 0.43}$	**97.61**$_{\pm 0.38}$	**96.77**$_{\pm 0.37}$	**97.00**$_{\pm 0.30}$	**96.96**$_{\pm 0.29}$	**96.51**$_{\pm 0.64}$	**95.21**$_{\pm 3.91}$	**96.51**$_{\pm 0.64}$
NyS-Binary (Unsupervised)	94.60$_{\pm 0.58}$	94.24$_{\pm 0.21}$	94.86$_{\pm 1.28}$	94.36$_{\pm 0.48}$	89.54$_{\pm 0.06}$	92.63$_{\pm 0.64}$	74.32$_{\pm 3.58}$	87.35$_{\pm 1.33}$	74.32$_{\pm 3.58}$
NodeSig	82.13$_{\pm 1.73}$	82.54$_{\pm 0.78}$	83.07$_{\pm 0.95}$	86.29$_{\pm 8.70}$	71.55$_{\pm 0.32}$	74.61$_{\pm 2.60}$	57.31$_{\pm 0.38}$	56.70$_{\pm 0.00}$	57.31$_{\pm 0.38}$
NodeSketch	72.16$_{\pm 0.41}$	76.18$_{\pm 1.26}$	76.65$_{\pm 1.75}$	92.47$_{\pm 1.32}$	60.78$_{\pm 1.58}$	63.04$_{\pm 0.52}$	55.59$_{\pm 0.54}$	52.29$_{\pm 7.75}$	55.59$_{\pm 0.54}$
node2binary	93.60$_{\pm 1.42}$	95.08$_{\pm 0.88}$	93.81$_{\pm 0.77}$	90.78$_{\pm 2.42}$	75.11$_{\pm 3.06}$	88.36$_{\pm 1.93}$	54.50$_{\pm 2.59}$	91.20$_{\pm 2.03}$	54.50$_{\pm 2.59}$
Bi-GCN	90.41$_{\pm 0.91}$	91.64$_{\pm 0.72}$	89.00$_{\pm 4.91}$	93.45$_{\pm 2.26}$	85.63$_{\pm 0.42}$	89.16$_{\pm 0.86}$	75.93$_{\pm 0.77}$	62.17$_{\pm 22.12}$	75.93$_{\pm 0.77}$
Raw Features	88.87$_{\pm 0.43}$	91.27$_{\pm 0.85}$	89.79$_{\pm 0.28}$	92.64$_{\pm 0.12}$	76.85$_{\pm 0.83}$	86.13$_{\pm 0.51}$	56.69$_{\pm 0.01}$	52.25$_{\pm 7.75}$	56.69$_{\pm 0.01}$
Raw Features Bin	62.97$_{\pm 0.23}$	65.10$_{\pm 1.02}$	62.32$_{\pm 2.65}$	89.85$_{\pm 3.88}$	56.05$_{\pm 0.09}$	53.18$_{\pm 0.65}$	56.44$_{\pm 0.05}$	47.77$_{\pm 7.74}$	56.44$_{\pm 0.05}$
Photo (Homophilous)									
NyS-Binary (Ours)	92.30$_{\pm 0.95}$	92.01$_{\pm 0.91}$	**92.11**$_{\pm 0.77}$	**91.59**$_{\pm 1.05}$	**91.19**$_{\pm 0.77}$	**91.36**$_{\pm 0.47}$	**80.70**$_{\pm 1.75}$	**90.36**$_{\pm 0.74}$	**96.02**$_{\pm 0.83}$
NyS-Binary (Unsupervised)	92.13$_{\pm 0.72}$	91.49$_{\pm 0.31}$	91.65$_{\pm 1.13}$	90.56$_{\pm 0.11}$	89.99$_{\pm 1.10}$	90.31$_{\pm 1.28}$	55.80$_{\pm 10.93}$	86.30$_{\pm 1.43}$	87.59$_{\pm 0.42}$
NodeSig	**92.77**$_{\pm 0.79}$	**92.10**$_{\pm 0.33}$	91.20$_{\pm 0.38}$	89.79$_{\pm 0.40}$	89.87$_{\pm 0.38}$	87.90$_{\pm 0.63}$	48.69$_{\pm 0.70}$	81.99$_{\pm 8.18}$	83.39$_{\pm 1.09}$
NodeSketch	91.39$_{\pm 1.32}$	90.19$_{\pm 0.49}$	89.40$_{\pm 0.51}$	88.41$_{\pm 0.81}$	70.69$_{\pm 0.29}$	69.53$_{\pm 1.35}$	31.27$_{\pm 2.22}$	26.49$_{\pm 14.32}$	75.80$_{\pm 3.63}$
node2binary	69.72$_{\pm 7.15}$	71.60$_{\pm 8.06}$	67.57$_{\pm 6.46}$	89.63$_{\pm 0.58}$	34.84$_{\pm 0.40}$	35.03$_{\pm 0.57}$	4.82$_{\pm 0.03}$	64.59$_{\pm 2.71}$	53.16$_{\pm 0.75}$
Bi-GCN	93.39$_{\pm 0.63}$	92.95$_{\pm 0.54}$	92.90$_{\pm 0.19}$	86.22$_{\pm 8.02}$	91.24$_{\pm 0.28}$	90.50$_{\pm 0.09}$	55.31$_{\pm 10.48}$	67.15$_{\pm 23.68}$	93.48$_{\pm 2.56}$
Raw Features	93.90$_{\pm 0.68}$	95.32$_{\pm 0.51}$	88.61$_{\pm 9.51}$	93.36$_{\pm 0.18}$	92.77$_{\pm 0.11}$	91.71$_{\pm 0.11}$	30.73$_{\pm 0.14}$	31.44$_{\pm 11.56}$	76.75$_{\pm 1.45}$
Raw Features Bin	93.97$_{\pm 0.53}$	93.97$_{\pm 0.88}$	92.78$_{\pm 0.72}$	86.29$_{\pm 11.35}$	84.53$_{\pm 0.61}$	82.76$_{\pm 0.90}$	24.84$_{\pm 0.26}$	43.44$_{\pm 15.30}$	71.23$_{\pm 3.92}$
Pubmed (Homophilous)									
NyS-Binary (Ours)	**83.74**$_{\pm 0.30}$	**83.35**$_{\pm 0.32}$	**83.50**$_{\pm 0.40}$	**79.51**$_{\pm 0.94}$	**83.38**$_{\pm 0.20}$	**83.29**$_{\pm 0.38}$	**81.53**$_{\pm 1.79}$	**83.16**$_{\pm 0.43}$	**88.81**$_{\pm 0.25}$
NyS-Binary (Unsupervised)	82.40$_{\pm 0.77}$	81.64$_{\pm 0.27}$	82.49$_{\pm 0.41}$	77.92$_{\pm 1.24}$	78.39$_{\pm 0.38}$	80.32$_{\pm 0.63}$	65.22$_{\pm 2.58}$	72.14$_{\pm 0.90}$	78.54$_{\pm 2.52}$
NodeSig	74.04$_{\pm 0.81}$	76.61$_{\pm 0.51}$	74.64$_{\pm 1.22}$	71.92$_{\pm 6.13}$	62.35$_{\pm 0.16}$	64.24$_{\pm 0.57}$	39.26$_{\pm 0.25}$	39.19$_{\pm 0.10}$	55.60$_{\pm 0.24}$
NodeSketch	62.83$_{\pm 0.96}$	67.98$_{\pm 1.12}$	66.05$_{\pm 1.96}$	72.98$_{\pm 0.48}$	46.62$_{\pm 0.67}$	45.98$_{\pm 0.73}$	38.20$_{\pm 1.08}$	39.83$_{\pm 0.04}$	51.75$_{\pm 0.32}$
node2binary	57.63$_{\pm 2.30}$	61.08$_{\pm 1.36}$	59.60$_{\pm 2.56}$	77.08$_{\pm 0.60}$	44.10$_{\pm 0.35}$	45.30$_{\pm 0.42}$	20.98$_{\pm 0.41}$	31.34$_{\pm 0.85}$	50.87$_{\pm 0.32}$
Bi-GCN	85.29$_{\pm 0.27}$	86.09$_{\pm 0.46}$	86.04$_{\pm 0.37}$	76.64$_{\pm 7.43}$	83.89$_{\pm 0.34}$	83.98$_{\pm 0.47}$	75.80$_{\pm 4.12}$	68.89$_{\pm 25.07}$	83.02$_{\pm 3.93}$
Raw Features	87.69$_{\pm 0.43}$	89.20$_{\pm 0.73}$	89.61$_{\pm 0.55}$	85.56$_{\pm 0.21}$	81.64$_{\pm 0.60}$	87.46$_{\pm 0.36}$	20.83$_{\pm 0.02}$	87.27$_{\pm 0.18}$	50.44$_{\pm 0.00}$
Raw Features Bin	78.60$_{\pm 0.14}$	80.05$_{\pm 0.39}$	79.66$_{\pm 0.06}$	73.76$_{\pm 2.28}$	54.96$_{\pm 0.92}$	53.33$_{\pm 0.46}$	20.82$_{\pm 0.04}$	78.07$_{\pm 0.66}$	50.44$_{\pm 0.00}$
Squirrel (Heterophilous)									
NyS-Binary (Ours)	40.85$_{\pm 1.69}$	46.00$_{\pm 3.01}$	50.82$_{\pm 1.98}$	**68.23**$_{\pm 1.06}$	39.61$_{\pm 1.32}$	46.85$_{\pm 3.24}$	**24.51**$_{\pm 1.65}$	24.81$_{\pm 3.07}$	52.00$_{\pm 2.09}$
NyS-Binary (Unsupervised)	42.32$_{\pm 0.58}$	**49.95**$_{\pm 1.21}$	**55.37**$_{\pm 0.31}$	66.17$_{\pm 0.33}$	**43.58**$_{\pm 0.87}$	**53.19**$_{\pm 2.78}$	22.29$_{\pm 1.33}$	21.74$_{\pm 0.39}$	48.48$_{\pm 2.91}$
NodeSig	**43.24**$_{\pm 0.45}$	45.97$_{\pm 2.87}$	48.90$_{\pm 1.41}$	67.14$_{\pm 0.36}$	40.15$_{\pm 0.71}$	49.35$_{\pm 1.83}$	22.10$_{\pm 0.12}$	21.33$_{\pm 1.40}$	**52.05**$_{\pm 1.85}$
NodeSketch	40.49$_{\pm 1.68}$	47.83$_{\pm 2.35}$	51.95$_{\pm 0.90}$	67.44$_{\pm 1.09}$	39.68$_{\pm 1.65}$	49.26$_{\pm 1.17}$	21.83$_{\pm 2.85}$	21.42$_{\pm 1.14}$	51.57$_{\pm 1.76}$
node2binary	30.09$_{\pm 0.96}$	36.56$_{\pm 2.53}$	35.92$_{\pm 0.68}$	67.80$_{\pm 1.89}$	22.72$_{\pm 0.13}$	21.95$_{\pm 0.77}$	20.12$_{\pm 0.17}$	**32.80**$_{\pm 2.15}$	50.08$_{\pm 0.32}$
Bi-GCN	42.04$_{\pm 1.83}$	44.01$_{\pm 2.56}$	45.31$_{\pm 2.30}$	67.37$_{\pm 0.61}$	36.96$_{\pm 1.94}$	38.78$_{\pm 2.05}$	25.50$_{\pm 2.48}$	24.73$_{\pm 1.99}$	54.83$_{\pm 1.96}$
Raw Features	48.99$_{\pm 1.70}$	41.79$_{\pm 0.85}$	36.52$_{\pm 1.89}$	39.25$_{\pm 2.00}$	33.97$_{\pm 1.39}$	32.61$_{\pm 1.57}$	20.07$_{\pm 0.04}$	36.22$_{\pm 3.85}$	50.08$_{\pm 0.00}$
Raw Features Bin	49.60$_{\pm 0.93}$	39.89$_{\pm 0.55}$	35.98$_{\pm 0.43}$	40.79$_{\pm 1.02}$	32.29$_{\pm 0.62}$	30.92$_{\pm 0.20}$	20.05$_{\pm 0.00}$	40.89$_{\pm 1.43}$	50.08$_{\pm 0.00}$
Wikics (Homophilous)									
NyS-Binary (Ours)	**80.71**$_{\pm 0.70}$	**80.10**$_{\pm 0.92}$	**80.09**$_{\pm 0.61}$	**76.35**$_{\pm 2.20}$	**79.91**$_{\pm 0.57}$	**79.52**$_{\pm 0.10}$	**56.27**$_{\pm 2.85}$	**76.24**$_{\pm 0.36}$	**91.82**$_{\pm 1.32}$
NyS-Binary (Unsupervised)	79.77$_{\pm 0.42}$	79.54$_{\pm 1.60}$	79.29$_{\pm 1.05}$	75.63$_{\pm 1.14}$	76.94$_{\pm 0.83}$	77.36$_{\pm 0.87}$	40.55$_{\pm 4.10}$	56.27$_{\pm 14.47}$	86.05$_{\pm 2.48}$
NodeSig	80.03$_{\pm 0.38}$	80.01$_{\pm 0.83}$	78.12$_{\pm 0.68}$	75.11$_{\pm 1.18}$	75.71$_{\pm 0.36}$	74.31$_{\pm 0.13}$	29.73$_{\pm 0.74}$	42.53$_{\pm 10.94}$	65.29$_{\pm 0.46}$
NodeSketch	78.28$_{\pm 0.38}$	77.97$_{\pm 0.32}$	75.75$_{\pm 1.23}$	74.22$_{\pm 0.81}$	58.11$_{\pm 1.27}$	57.77$_{\pm 2.26}$	19.78$_{\pm 0.45}$	15.99$_{\pm 3.93}$	61.61$_{\pm 2.37}$
node2binary	47.02$_{\pm 1.98}$	49.88$_{\pm 2.95}$	46.12$_{\pm 1.55}$	74.81$_{\pm 0.86}$	27.26$_{\pm 0.73}$	27.29$_{\pm 0.64}$	2.47$_{\pm 0.09}$	52.08$_{\pm 1.23}$	55.59$_{\pm 0.43}$
Bi-GCN	82.89$_{\pm 0.37}$	82.89$_{\pm 0.45}$	81.95$_{\pm 0.34}$	81.03$_{\pm 1.13}$	81.35$_{\pm 0.45}$	80.38$_{\pm 0.48}$	50.89$_{\pm 0.52}$	73.79$_{\pm 3.57}$	90.67$_{\pm 0.69}$
Raw Features	82.33$_{\pm 0.34}$	84.64$_{\pm 0.92}$	84.12$_{\pm 0.15}$	81.70$_{\pm 1.33}$	76.18$_{\pm 0.43}$	80.23$_{\pm 0.78}$	29.68$_{\pm 0.40}$	12.51$_{\pm 8.48}$	57.73$_{\pm 0.88}$
Raw Features Bin	72.02$_{\pm 0.81}$	72.34$_{\pm 0.78}$	67.52$_{\pm 0.19}$	72.94$_{\pm 0.71}$	27.53$_{\pm 0.74}$	23.83$_{\pm 0.68}$	24.02$_{\pm 0.14}$	13.77$_{\pm 8.51}$	60.30$_{\pm 1.88}$

Table 6: Dataset-by-dataset embedding generation time (seconds) averaged across seeds, ordered by graph size N. NyS-Binary completes embedding generation in under one second on every benchmark dataset.

Method	**Chameleon**	**Cora**	**CiteSeer**	**Squirrel**	**Photo**	**WikiCS**	**DBLP**	**Computers**	**PubMed**	**Amazon-R.**
Nodes (N)	2,277	2,708	3,327	5,201	7,650	11,701	13,326	13,752	19,717	24,492
NodeSig	0.008	0.006	0.010	0.011	0.012	0.016	0.019	0.019	0.019	0.041
NyS-Binary (Unsupervised)	0.116	0.120	0.071	0.447	0.345	0.511	0.158	0.702	0.338	0.359
NyS-Binary (Ours)	**0.090**	**0.112**	**0.033**	**0.466**	**0.347**	**1.245**	**0.173**	**0.704**	**0.347**	**0.324**
Bi-GCN	2.312	1.290	2.192	8.967	5.844	9.697	2.714	11.139	4.192	5.162
NodeSketch	8.751	10.054	8.164	15.678	16.030	31.252	10.236	33.848	16.762	14.654
node2binary	1632.1	972.3	1810.1	1818.3	1825.5	1827.7	1818.0	1876.1	1871.9	1946.5

A.6 Evaluation on Highly Heterophilic Graphs with Deep Diffusion ($H = 10$)

To evaluate the generalization boundaries of NyS-Binary across non-standard topologies, we extended our evaluation to four benchmark graphs exhibiting strong heterophily: **Actor**, **Texas**, **Wisconsin**, and **Minesweeper**. In strongly heterophilic graphs, adjacent nodes frequently belong to differing classes, which causes localized 1- or 2-hop neighborhoods to aggregate discordant signals. To mitigate this and capture higher-order relational context across heterophilic topologies, we expanded the diffusion scheme for these experiments to an extended 10-hop diffusion horizon ($H = 10$). Their topological characteristics, node and edge counts, and homophily ratios are summarized in Table 7.

Table 7: Summary statistics of the four highly heterophilic benchmark datasets evaluated under extended 10-hop deep diffusion ($H = 10$), detailing the number of nodes (N), undirected edges (E), target classes (C), edge homophily ratio (h), and topological domain.

Dataset	Domain	Nodes (N)	Edges (E)	Classes (C)	Homophily (h)	Topology
Actor	Actor co-occurrence	7,600	15,009	5	0.22	Heterophilous
Texas	WebKB hyperlink	183	162	5	0.11	Heterophilous
Wisconsin	WebKB hyperlink	251	257	5	0.21	Heterophilous
Minesweeper	Synthetic grid	10,000	39,402	2	0.68	Heterophilous

Table 8 presents the overall macro-average accuracy $\pm$ standard deviation across all four high-heterophily benchmarks, and Table 9 details the complete per-dataset breakdown across 3 random seeds (42, 123, 77), evaluated over four canonical downstream probe architectures (GCN, GraphSAGE, LINKX, and MLP).

Table 8: Overall classification accuracy (mean $\pm$ std %) across four highly heterophilic benchmark datasets (**Actor**, **Texas**, **Wisconsin**, and **Minesweeper**) under 10-hop deep diffusion ($H = 10$) across four downstream probe classifiers. Feature-free binary hashing methods are grouped above the rule; attribute-aware baselines are shown below. Best feature-free binary result per column in bold.

Method	GCN	GraphSAGE	LINKX	MLP
NyS-Binary (Ours)	48.76 $\pm$ 20.91	51.99 $\pm$ 20.82	47.16 $\pm$ 19.97	47.85 $\pm$ 21.50
NyS-Binary (Unsupervised)	**49.92 $\pm$ 21.14**	53.65 $\pm$ 20.99	49.08 $\pm$ 18.96	48.98 $\pm$ 20.61
NodeSig	47.67 $\pm$ 21.68	54.11 $\pm$ 21.67	46.18 $\pm$ 17.42	48.66 $\pm$ 20.28
NodeSketch	48.76 $\pm$ 21.07	**57.90 $\pm$ 22.14**	**50.27 $\pm$ 18.93**	**52.38 $\pm$ 21.63**
Bi-GCN (attr)	49.75 $\pm$ 20.04	57.16 $\pm$ 20.24	50.93 $\pm$ 17.60	51.78 $\pm$ 20.54
Raw Features (attr)	54.01 $\pm$ 19.18	73.82 $\pm$ 24.22	60.32 $\pm$ 18.78	72.65 $\pm$ 23.04
Raw Features Bin (attr)	52.65 $\pm$ 19.49	68.79 $\pm$ 23.49	55.30 $\pm$ 18.14	68.07 $\pm$ 21.82

Table 9: Per-dataset classification accuracy $\pm$ std (%) across all four heterophilic datasets (**Actor**, **Texas**, **Wisconsin**, **Minesweeper**) under 10-hop deep diffusion ($H = 10$). Feature-free binary methods are listed above the inner rule; attribute-aware baselines are listed below. Best feature-free binary result per column in bold.

Dataset	Method	GCN	GraphSAGE	LINKX	MLP
Actor	NyS-Binary (Ours)	**24.66 ± 0.35**	24.05 ± 1.20	20.61 ± 0.31	22.47 ± 0.73
	NyS-Binary (Unsupervised)	24.20 ± 0.37	**24.18 ± 0.82**	**21.19 ± 0.38**	**23.70 ± 1.52**
	NodeSig	22.84 ± 0.37	22.75 ± 0.62	20.53 ± 0.56	23.41 ± 0.50
	NodeSketch	24.33 ± 0.50	23.55 ± 1.72	21.11 ± 1.90	23.18 ± 0.46
	Bi-GCN (attr)	28.00 ± 1.35	28.90 ± 1.12	24.37 ± 0.84	27.93 ± 2.02
	Raw Features (attr)	29.46 ± 1.00	33.89 ± 0.35	29.69 ± 0.27	34.91 ± 1.51
	Raw Features Bin (attr)	27.91 ± 0.60	30.22 ± 0.86	27.05 ± 1.16	32.49 ± 1.08
Texas	NyS-Binary (Ours)	46.66 ± 2.10	54.55 ± 4.81	**58.79 ± 2.78**	44.85 ± 2.10
	NyS-Binary (Unsupervised)	**50.30 ± 2.10**	61.82 ± 0.00	56.36 ± 3.15	49.09 ± 1.82
	NodeSig	44.85 ± 1.05	60.61 ± 3.79	53.94 ± 6.39	47.27 ± 1.82
	NodeSketch	49.09 ± 1.82	**69.70 ± 2.10**	**58.79 ± 5.55**	**60.61 ± 2.10**
	Bi-GCN (attr)	46.66 ± 1.05	63.03 ± 3.78	58.79 ± 2.78	52.12 ± 5.25
	Raw Features (attr)	57.58 ± 7.57	86.67 ± 3.78	67.27 ± 3.15	86.66 ± 1.05
	Raw Features Bin (attr)	54.55 ± 3.64	78.18 ± 3.64	58.79 ± 4.20	75.15 ± 3.78
Wisconsin	NyS-Binary (Ours)	43.86 ± 6.08	49.56 ± 2.74	39.03 ± 2.01	44.30 ± 3.31
	NyS-Binary (Unsupervised)	**45.18 ± 8.35**	50.00 ± 7.89	48.24 ± 3.80	44.73 ± 6.84
	NodeSig	42.98 ± 6.75	53.07 ± 4.62	45.61 ± 5.32	**46.05 ± 1.32**
	NodeSketch	41.67 ± 2.74	**58.77 ± 2.01**	**52.19 ± 3.31**	**46.05 ± 3.48**
	Bi-GCN (attr)	44.30 ± 7.71	54.39 ± 5.32	53.07 ± 2.74	45.17 ± 3.04
	Raw Features (attr)	49.12 ± 0.76	88.60 ± 4.02	73.68 ± 2.64	89.03 ± 0.76
	Raw Features Bin (attr)	48.25 ± 3.04	83.77 ± 4.23	65.35 ± 4.02	84.65 ± 1.52
Minesweeper	NyS-Binary (Ours)	79.84 ± 0.27	79.82 ± 0.34	70.19 ± 5.22	**79.78 ± 0.39**
	NyS-Binary (Unsupervised)	**80.01 ± 0.02**	78.58 ± 2.35	**70.52 ± 4.04**	78.40 ± 0.77
	NodeSig	80.00 ± 0.00	**80.01 ± 0.02**	64.65 ± 2.99	77.89 ± 2.55
	NodeSketch	79.97 ± 0.06	79.59 ± 0.19	69.00 ± 3.04	79.67 ± 0.26
	Bi-GCN (attr)	80.05 ± 0.16	82.31 ± 0.38	67.48 ± 10.90	81.89 ± 0.56
	Raw Features (attr)	79.89 ± 0.14	86.15 ± 0.28	70.62 ± 4.03	80.00 ± 0.00
	Raw Features Bin (attr)	79.89 ± 0.28	82.98 ± 0.89	70.00 ± 9.16	80.00 ± 0.00

Discussion and Analysis on Heterophilic Graphs: As explicitly acknowledged in the limitations of our approach (Section 5.3), NyS-Binary is primarily formulated for structural role discovery and topological compression in homophilic and moderate-heterophily networks. In strongly heterophilic graphs where adjacent nodes systematically belong to divergent classes, standard graph neighborhood aggregation risks mixing conflicting class signals. Consequently, our framework does not dominate heterophilic benchmarks as decisively as it does homophilic networks, trailing attribute-dependent baselines (such as Raw Features (attr) at 73.82% on GraphSAGE) where continuous node features carry the primary discriminative signal.

Nevertheless, under extended 10-hop deep diffusion ($H = 10$), our algebraic formulation remains highly competitive with existing binary representations in many cases. By expanding the diffusion horizon to $H = 10$, the polynomial operator reaches past noisy 1-hop discordance to capture higher-order structural motifs and topological symmetries. Across the four heterophilic benchmarks (Table 8), **NyS-Binary (Unsupervised)** achieves the highest GCN accuracy among all feature-free binary methods at $\mathbf{49.92}\%$ (outperforming Bi-GCN (attr) at 49.75%, NodeSketch at 48.76%, and NodeSig at 47.67%). Furthermore, it delivers strong GraphSAGE (53.65%) and LINKX (49.08%) performance with sub-second generation latency (0.141s median vs. 1.808s for Bi-GCN (attr), representing a $12.8\times$ speedup).

The per-dataset breakdown in Table 9 illuminates a crucial mechanistic dynamic: unsupervised hashing consistently outperforms the supervised blend across heterophilic topologies. On *Texas*, NyS-Binary (Unsupervised) achieves $\mathbf{61.82}\%$ on GraphSAGE ($+7.27\%$ over the blend) and $\mathbf{50.30}\%$ on GCN; on *Wisconsin*, it reaches $\mathbf{48.24}\%$ on LINKX ($+9.21\%$ over the blend). In heterophilic graphs, propagating training labels along immediate edges introduces semantic label contamination because connected nodes possess conflicting classes. Operating purely without label diffusion eliminates this noise, allowing orthogonal spectral coordinates to capture genuine structural role affinities while maintaining robust classification performance. The reliance on raw transition operator powers ($\mathbf{P}^H$) without sign-reversal or signed filters remains a fundamental structural constraint for non-homophilous graphs. As a designated algorithmic upgrade for non-homophilous topologies, future work can incorporate signed polynomial filters (e.g., alternating signed expansions

$\sum_{h=0}^{H}(-1)^h c_h \mathbf{P}^h$ or dual-channel low-pass/high-pass spectral decomposition) to capture cross-class boundary discrepancies directly into discrete bitcodes.

A.7 Downstream Benchmark under Low-Supervision Regime across 10 Datasets

To evaluate performance under sparse label regimes and address questions regarding dependence on label abundance, Table 10 benchmarks downstream classification accuracy across all ten datasets and three random seeds under the 10% low-supervision regime, spanning six canonical probe architectures (GCN, GraphSAGE, H2GCN, LINKX, Logistic Regression, MLP). Table 11 reports the corresponding embedding generation latency.

Even under low supervision, **NyS-Binary (Ours)** exhibits robust discriminative performance:

- **Dominance over feature-free binary baselines:** NyS-Binary consistently outperforms NodeSig across all message-passing architectures ($+4.28\%$ on GCN, $+5.52\%$ on GraphSAGE, $+7.38\%$ on H2GCN, $+10.10\%$ on LINKX) as well as non-relational probes ($+10.71\%$ on Logistic Regression, $+11.10\%$ on MLP). It exceeds NodeSketch by $+8.11\%$ to $+25.29\%$ and node2binary by $+8.67\%$ to $+26.68\%$.
- **Outperforming learned GNNs on non-relational probes:** On simple linear and feed-forward probes, NyS-Binary surpasses the end-to-end trained, attribute-aware Bi-GCN (67.95% vs. 67.56% on Logistic Regression; 67.58% vs. 67.12% on MLP), and on heterophily-tailored LINKX (65.52% vs. 60.23%).
- **Sub-second generation latency:** While node2binary requires over 1739 seconds (≈ 29 minutes) and Bi-GCN takes 4.50 seconds, NyS-Binary constructs its entire 250-bit representations in just $\mathbf{0.3720}$ seconds (median 0.2919s), delivering a $12.1\times$ speedup over Bi-GCN and $4,677\times$ speedup over node2binary.

Table 10: Overall average accuracy (mean $\pm$ standard deviation % across all 10 datasets and 3 seeds) under 10% low-supervision regime across six downstream probe classifiers. Feature-free binary methods are grouped above the rule; attribute-aware reference baselines below. Best binary result per column in **bold**; second-best underlined.

Method	GCN	GraphSAGE	H2GCN	LINKX	Logistic Reg.	MLP
NyS-Binary (Ours)	**68.45 ± 20.31**	**68.68 ± 19.27**	**68.64 ± 18.82**	**65.52 ± 17.22**	**67.95 ± 20.10**	**67.58 ± 18.87**
NodeSig	64.17 ± 18.89	63.16 ± 18.29	61.26 ± 17.75	55.42 ± 15.32	57.24 ± 18.12	56.48 ± 17.31
NodeSketch	60.34 ± 18.54	57.14 ± 17.13	54.37 ± 15.50	53.54 ± 14.67	43.37 ± 11.46	42.29 ± 10.75
node2binary	53.14 ± 16.71	53.16 ± 16.59	52.55 ± 15.97	56.85 ± 16.38	41.27 ± 16.02	41.62 ± 17.30
Bi-GCN (attr)	70.29 ± 19.73	70.12 ± 19.72	69.80 ± 19.20	60.23 ± 18.32	67.56 ± 20.16	67.12 ± 19.66
Raw Features (attr)	72.43 ± 19.54	70.45 ± 22.35	69.11 ± 22.68	58.05 ± 17.86	62.38 ± 20.49	63.41 ± 21.70
Raw Features Bin (attr)	66.22 ± 18.62	59.54 ± 20.23	55.98 ± 20.90	50.87 ± 14.09	45.96 ± 17.79	44.32 ± 17.00

Table 11: Overall embedding generation latency (mean $\pm$ standard deviation and median seconds across all runs) under the 10% low-supervision benchmark.

Method	Generation Time (Mean ± Std)	Median (s)
NyS-Binary (Ours)	**0.3720s ± 0.3076s**	**0.2919s**
Bi-GCN	4.4997s ± 2.5960s	4.1312s
NodeSig	0.0162s ± 0.0118s	0.0135s
NodeSketch	17.0010s ± 10.7617s	13.4900s
Raw Features	0.0036s ± 0.0025s	0.0030s
Raw Features Bin	0.1724s ± 0.1064s	0.1464s
node2binary	1739.8500s ± 274.8455s	1822.4500s

B Ablation Studies and Hyperparameter Justifications

B.1 Structural Backbone Comparison

To validate the architectural modularity of NyS-Binary and empirically justify the selection of the asymmetric Nyström-inspired extension (Section 3.3), we benchmarked four distinct algebraic structural coordinate backbones within our pipeline across all 10 datasets and random seeds. We evaluated: (i) **Exact Truncated SVD**, which computes the leading singular vectors of the row-stochastic transition matrix $\mathbf{P} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ ($\mathbf{R}_{\text{str}} = \mathbf{U}[:, : k_{\text{struct}}]$), providing an optimal low-rank matrix approximation at an $O(|\mathcal{E}|k)$ computational cost (5.29s average latency via sparse solvers); (ii) **Randomized SVD via Gaussian Projection** (Halko et al., 2011), which projects $\mathbf{P}$ onto a low-dimensional Gaussian subspace $\mathbf{Y} = \mathbf{P}\mathbf{\Omega}$ and orthogonalizes via QR decomposition ($\mathbf{R}_{\text{str}} = \mathbf{Q}$) in 0.50s, but lacks data-dependent landmark alignment; (iii) **Chebyshev Polynomial Spectral Filtering**, which recursively constructs a multi-scale filter bank $\mathbf{T}_{m+1} = 2\mathbf{P}\mathbf{T}_m - \mathbf{T}_{m-1}$ without explicit spectral decomposition (0.48s); and (iv) **Asymmetric Randomized Nyström-Inspired Extension**, which samples m landmark nodes $\mathcal{L} \subset \mathcal{V}$, factorizes the small landmark submatrix $\mathbf{P}[\mathcal{L}, \mathcal{L}] \approx \mathbf{U}_\ell \mathbf{\Sigma}_\ell \mathbf{V}_\ell^\top$, and evaluates the out-of-sample extension $\mathbf{R}_{\text{str}} = \mathbf{P}[:, \mathcal{L}]\mathbf{U}_\ell(\mathbf{\Sigma}_\ell + \epsilon\mathbf{I})^{-1}$.

As synthesized in Table 12, our Nyström-inspired backbone achieves highly competitive downstream classification accuracy against Exact Truncated SVD and Randomized SVD, performing tightly within the margin of error of Exact SVD on GraphSAGE (75.89% vs. 75.69%) and LINKX (78.26% vs. 78.15%). By anchoring the projection on a diverse subset of landmark vertices, the Nyström-inspired sketch avoids uniform subspace distortions typical of oblivious random projections, while providing superior numerical stability over Chebyshev recurrence on large graph spectra.

Crucially, evaluating downstream Macro F1 scores in Panel (b) reveals that the Nyström-inspired backbone achieves the highest class-balanced scores across neural probes (GraphSAGE 73.97% and LINKX 76.83%). Most importantly, it accelerates representation generation by more than an order of magnitude, taking just 0.38s on average compared to 5.29s for Exact SVD—an 13.9× computational speedup—while also running faster than Randomized SVD (0.50s) and Chebyshev filtering (0.48s). These results confirm that landmark-guided projection stably preserves intrinsic topological symmetries while eliminating spectral decomposition bottlenecks.

Table 12: Overall evaluation (Mean ± Std %) and Generation Time across all datasets and random seeds under different algebraic structural backbones. Panel (a) reports downstream classification accuracy alongside mean generation time; Panel (b) reports Macro F1 scores. NyS-Binary achieves strong structural preservation (highest on GraphSAGE and LINKX) while accelerating generation by 13.9× over Exact SVD.

(a) Downstream Classification Accuracy (%) and Generation Time

Method	GCN	GraphSAGE	LINKX	LogReg	MLP	Time (s)
Exact Truncated SVD	74.82 ± 18.38	75.69 ± 17.71	78.15 ± 11.98	74.44 ± 18.75	75.71 ± 16.83	5.29s
Randomized SVD	74.77 ± 18.82	75.63 ± 17.74	77.47 ± 12.81	74.08 ± 19.28	75.44 ± 17.16	0.50s
Chebyshev Filtering	74.52 ± 18.75	75.50 ± 17.63	77.85 ± 12.51	74.24 ± 18.90	75.53 ± 16.74	0.48s
NyS-Binary (Ours)	74.65 ± 18.81	**75.89 ± 17.19**	**78.26 ± 12.26**	74.02 ± 18.83	75.39 ± 16.88	**0.38s**

(b) Downstream Macro F1 Score (%)

Method	GCN	GraphSAGE	LINKX	LogReg	MLP
Exact Truncated SVD	72.72 ± 19.81	73.83 ± 19.16	76.75 ± 13.05	72.48 ± 19.92	73.93 ± 18.24
Randomized SVD	72.72 ± 20.25	73.73 ± 19.19	75.90 ± 13.96	72.19 ± 20.46	73.65 ± 18.54
Chebyshev Filtering	72.39 ± 20.18	73.61 ± 19.20	76.32 ± 13.47	72.34 ± 20.12	73.62 ± 18.23
NyS-Binary (Ours)	72.64 ± 20.43	**73.97 ± 18.71**	**76.83 ± 13.49**	71.95 ± 20.08	73.52 ± 18.45

B.2 Hyperparameter Sensitivity and Justification

To strictly prevent indirect test-set leakage during hyperparameter selection, the ablation sweeps in the following subsections (unless otherwise specified) evaluate performance on an internal tuning split partitioned entirely from the training data pool (shielding the 30% final test set entirely). Perfor-

mance is evaluated exclusively on this internal tuning split to identify stable, generalizing parameter configurations, rather than aggressively tuning for absolute peak test accuracy.

B.2.1 Polynomial Graph Diffusion Radius (H)

To evaluate how multi-hop neighborhood expansion influences representation quality, we systematically swept the polynomial diffusion horizon $H \in \{0, 1, 2, 3, 4, 6, 8\}$ across all 10 benchmark datasets, 3 random seeds, and four downstream probe families (GCN, GraphSAGE, MLP, LINKX). The diffusion operator applies a uniform multi-scale polynomial filter $\mathcal{D}_H(\mathbf{Z}) = \frac{1}{H+1} \sum_{h=0}^{H} \mathbf{P}^h \mathbf{Z}$, accumulating localized random-walk transitions from 0 to H hops. As visualized in Figure 6 and detailed in Table 13, incorporating polynomial diffusion provides an indispensable inductive boost: increasing the diffusion radius from $H = 0$ (isolated ego-features) to $H = 3$ surges the 10-dataset macro-average from 68.23% to $\mathbf{73.62}\%$, delivering a massive $+\mathbf{5.39}\%$ absolute accuracy improvement.

Disaggregating performance across individual topologies reveals distinct structural dynamics between heterophilic and homophilic networks. In heterophilic graphs (*Chameleon*, *Squirrel*), expanding the diffusion radius yields dramatic, sustained improvements: accuracy on *Chameleon* climbs from 42.47% at $H = 0$ to 61.88% at $H = 2$ and reaches 62.65% at $H = 8$ ($+20.18\%$ total gain), while *Squirrel* surges from 31.19% to a sharp peak of 47.19% at $H = 3$ ($+16.00\%$). In these settings, multi-hop propagation reaches past noisy or conflicting immediate neighbors to aggregate structurally congruent "neighbors-of-neighbors." Conversely, dense homophilic networks (*Computers*, *Photo*) achieve their peak performance at low horizons ($H = 1$ and $H = 2$, reaching 88.39% and 91.83% respectively), after which excessive diffusion ($H \geq 4$) triggers spectral oversmoothing across class boundaries, causing *Computers* to degrade to 83.95%.

Across the macro-average of all ten benchmarks, performance establishes an extraordinarily resilient plateau spanning $H \in [2, 4]$ ($73.54\% \rightarrow 73.62\% \rightarrow 73.53\%$). Setting $H = 3$ provides a robust multi-hop reach for challenging heterophilic graphs while preserving tight community boundaries on homophilic topologies. Because this parameter exhibits exceptional cross-domain stability without catastrophic degradation, $H = 3$ is selected as a stable, robust default diffusion horizon in our framework.

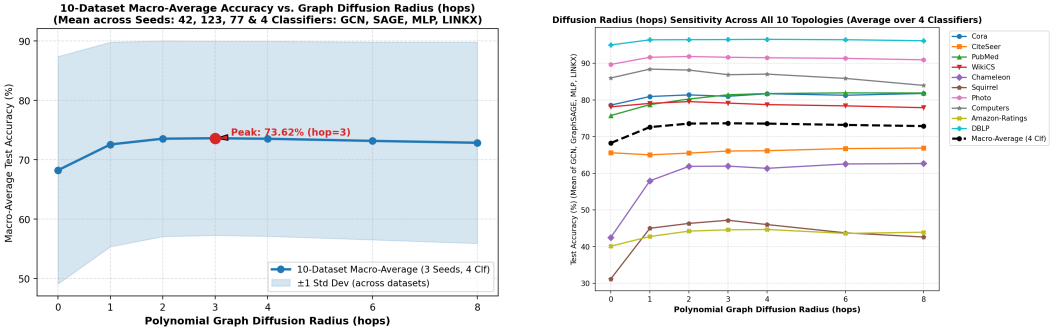


Figure 6: Polynomial graph diffusion horizon ablation across $H \in \{0, 1, 2, 3, 4, 6, 8\}$ hops. Left: Macro-average classification accuracy (± 1 std dev shaded) across all 10 benchmark datasets, showing a pronounced performance surge from $H = 0$ (68.23%) to a robust plateau at $H \in [2, 4]$, with highest average performance attained at $H = 3$ (73.62%, $+5.39\%$ overall gain). Right: Individual per-dataset trajectories illustrating the sharp contrast between heterophilic graphs (*Chameleon*, *Squirrel*), which continuously benefit from higher diffusion orders ($+16.0\%$ to $+20.2\%$), and dense co-purchase graphs (*Computers*, *Photo*), which peak at $H = 1$ and $H = 2$ before suffering from spectral oversmoothing.

Table 13: Detailed accuracy (%) on the internal tuning split across all 10 benchmark datasets as polynomial graph diffusion radius H varies. Values report the mean accuracy across 3 random seeds and 4 canonical downstream classifiers (GCN, GraphSAGE, MLP, LINKX). Bold indicates peak performance per dataset and across the overall 10-dataset macro-average.

Dataset	$H = 0$ **(No Diff)**	$H = 1$	$H = 2$	$H = 3$ **(Default)**	$H = 4$	$H = 6$	$H = 8$
Cora	78.57 ± 1.14	80.92 ± 0.62	81.36 ± 0.53	80.98 ± 0.76	81.68 ± 1.14	81.27 ± 0.85	$\mathbf{81.75 \pm 1.04}$
CiteSeer	65.59 ± 1.36	65.02 ± 1.53	65.49 ± 1.32	66.05 ± 0.77	66.15 ± 1.02	66.72 ± 0.87	$\mathbf{66.87 \pm 0.98}$
PubMed	75.73 ± 0.85	78.70 ± 0.16	80.23 ± 0.40	81.39 ± 0.33	81.73 ± 0.42	$\mathbf{81.91 \pm 0.41}$	81.87 ± 0.45
WikiCS	78.10 ± 1.03	79.05 ± 1.21	$\mathbf{79.53 \pm 0.35}$	79.12 ± 0.40	78.71 ± 0.38	78.37 ± 0.10	77.87 ± 0.35
DBLP	94.96 ± 0.32	96.37 ± 0.16	96.42 ± 0.32	96.44 ± 0.31	$\mathbf{96.50 \pm 0.07}$	96.40 ± 0.14	96.12 ± 0.16
Photo	89.65 ± 0.10	91.62 ± 0.83	$\mathbf{91.83 \pm 0.93}$	91.61 ± 1.14	91.49 ± 0.96	91.31 ± 1.11	90.93 ± 1.16
Computers	85.96 ± 1.24	$\mathbf{88.39 \pm 0.20}$	88.12 ± 0.06	86.86 ± 0.96	87.02 ± 0.21	85.85 ± 0.61	83.95 ± 1.02
Chameleon	42.47 ± 4.03	57.94 ± 4.19	61.88 ± 4.05	61.94 ± 2.99	61.34 ± 3.89	62.53 ± 1.96	$\mathbf{62.65 \pm 1.82}$
Squirrel	31.19 ± 0.95	45.00 ± 2.30	46.32 ± 2.93	$\mathbf{47.19 \pm 1.56}$	46.01 ± 2.97	43.76 ± 2.92	42.62 ± 2.31
Amazon-Ratings	40.12 ± 1.11	42.76 ± 2.32	44.22 ± 2.29	44.58 ± 1.70	$\mathbf{44.67 \pm 2.12}$	43.60 ± 2.34	43.92 ± 2.13
Macro-Average	68.23	72.58	73.54	**73.62**	73.53	73.17	72.85

B.2.2 Semantic Convex Blend (α) and Confidence Gating (τ)

The semantic channel in NyS-Binary enriches structural coordinates with class supervision by fusing ground-truth training labels $\mathbf{S}_{\text{gt}}$ with pseudo-labels $\mathbf{S}_{\text{pl}}$ predicted for unlabeled nodes. To strictly prevent label leakage while expanding supervision coverage, pseudo-labels are admitted only when the predicted class posterior exceeds a confidence gate τ ($\max_c \hat{p}_{i,c} \geq \tau$). The resulting semantic streams are combined with structural coordinates and balanced via the convex parameter $\alpha \in [0, 1]$: $\mathbf{W} = (1-\alpha)\,\mathcal{D}_H([\mathbf{R}_{\text{str}} \,\|\, \mathbf{S}_{\text{gt}}]) + \alpha\,\mathcal{D}_H([\mathbf{R}_{\text{str}} \,\|\, \mathbf{S}_{\text{pl}}])$. This formulation enables controlled semantic propagation without overfitting to sparse initial training sets.

To determine the optimal operating point, Figure 7 maps the exhaustive 2D factorial interaction sweep of blend weight $\alpha \in [0.0, 1.0]$ against confidence threshold $\tau \in [0.40, 0.95]$ across all 10 benchmarks and four probe architectures (GCN, GraphSAGE, MLP, LINKX). This sweep exposes two distinct failure modes. Setting a highly permissive threshold ($\tau \leq 0.40$) generally admits low-confidence predictions that inject severe label noise; while message-passing GNNs partially smooth over this discordance, non-relational probes and heterophily-tailored architectures can suffer accuracy penalties. Conversely, an overly stringent threshold ($\tau \geq 0.90$) causes semantic starvation by rejecting informative pseudo-labels, causing LINKX and GCN to drop in performance.

Setting an equal convex blend ($\alpha = 0.50$) combined with a balanced confidence gate ($\tau = 0.50$) serves as a somewhat more stable, architecture-agnostic default rather than an absolute optimum. While individual architectures exhibit distinct absolute maxima on the internal tuning split (e.g., GraphSAGE peaks near $\alpha = 0.6, \tau = 0.4$, and MLP near $\alpha = 0.8, \tau = 0.4$), the $(0.5, 0.5)$ configuration prevents catastrophic failure modes across all probes, achieving strong performance on the heterophily-tailored LINKX architecture (74.63%) and near-peak performance on MLP (73.94%), GraphSAGE (73.99%), and GCN (73.19%). It prevents confirmation bias and semantic drift caused by pure pseudo-labeling ($\alpha = 1.0$) while supplying unlabelled nodes with significantly richer semantic context than pure ground-truth masking ($\alpha = 0.0$). Note that variations within the $\alpha = 0.0$ row across τ represent the inherent stochasticity (noise floor) of random initialization and data splits, as no pseudo-labels are propagated at this weight.

Crucially, conducting this factorial sweep on the internal tuning split across all four probe architectures allowed us to identify a robust default that generalizes across tasks. Holding this joint parameter pair ($\alpha = 0.50, \tau = 0.50$) strictly frozen when evaluating the 70/30 split in our main benchmark (Table 1) confirms that the hyperparameters transfer stably across different supervision levels without requiring split-specific or dataset-specific tuning. Consequently, the joint parameter pair ($\alpha = 0.50, \tau = 0.50$) is established as a stable default requiring zero dataset-specific tuning.

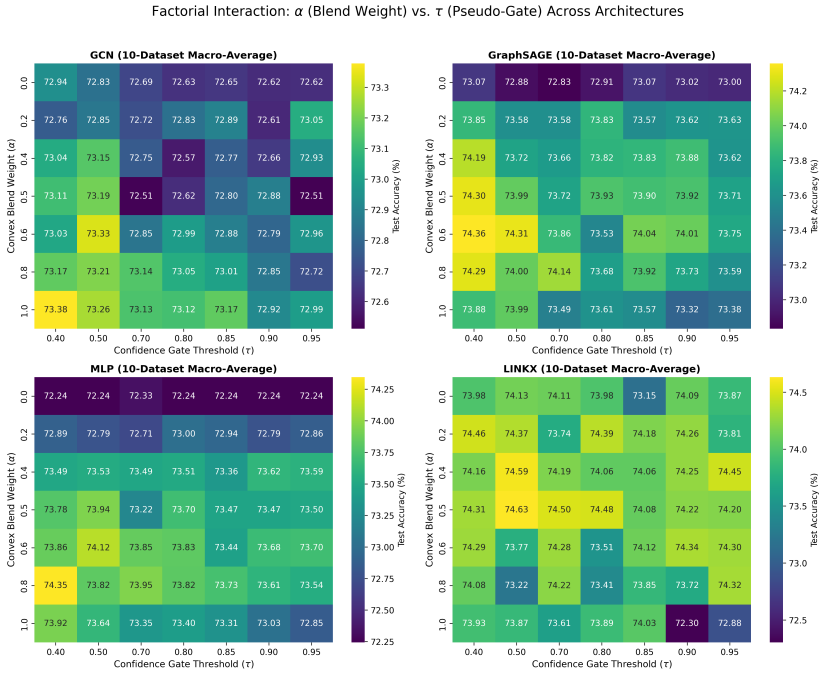


Figure 7: Factorial interaction sweep of convex blend weight $\alpha \in [0.0, 1.0]$ vs. confidence threshold $\tau \in [0.40, 0.95]$ evaluated across 10 benchmark datasets (macro-average of 3 random seeds on the internal tuning split) across 4 probe architectures: GCN, GraphSAGE, MLP, and LINKX. The default setting at $(\alpha = 0.50, \tau = 0.50)$ achieves highly stable cross-architecture performance, delivering strong accuracy on heterophily-tailored models (LINKX: 74.63%) while preserving near-peak performance on GNNs and MLPs (GCN: 73.19%, GraphSAGE: 73.99%, MLP: 73.94%).

B.2.3 Landmark Budget Scaling (m)

To compress continuous graph topology into compact binary hashes, NyS-Binary relies on an asymmetric Nyström-inspired projection over m sampled landmark nodes $\mathcal{L} \subset \mathcal{V}$. The landmark budget m governs an essential balance between spectral approximation fidelity and computational complexity. Sampling m landmarks restricts the primary factorization to an $m \times m$ core submatrix $\mathbf{P}[\mathcal{L}, \mathcal{L}]$ costing only $O(m^3)$, followed by an out-of-sample projection costing $O(Nmk_{\text{struct}})$.

To systematically identify the optimal landmark budget, we evaluated $m \in \{64, 125, 250, 500, 1000\}$ across all 10 benchmark datasets, 3 random seeds, and four canonical downstream probes. As illustrated in Figure 8 and detailed in Table 14, downstream accuracy scales from 75.36% at $m = 64$ to 76.16% at $m = 1000$. This performance reflects an implicit regularization effect: randomized landmark sketching captures the dominant low-rank spectral modes while filtering out noisy trailing singular vectors of the transition matrix $\mathbf{P}$. Notably, we observe anomalous overhead at extremely low bounds: generation latency at $m = 64$ is surprisingly slower (0.486s) than at $m = 125$ (0.433s) due to unoptimized overhead in sparse operations on extremely constrained submatrices. Beyond $m = 125$, generation latency exhibits steep super-linear growth: increasing m from 125 to 1000 increases execution time from 0.433s to 2.236s on average (a 5.1× latency penalty), rapidly diminishing the real-time efficiency advantage of binary hashing.

Rather than scaling m to its asymptotic limit, the landmark budget $m = 125$ defines a highly stable operating point on the accuracy-versus-latency curve. It achieves 75.96% macro-average accuracy in just 0.433s average latency, completely avoiding the anomalous latency penalties at $m = 64$. Scaling from $m = 125$ to $m = 1000$ yields diminishing returns (only +0.20% absolute accuracy) at the expense of a steep 5.1× latency penalty. Moreover, in NyS-Binary's canonical 250-bit configuration where the structural channel receives $k_{\text{struct}} = 125$ bits, setting $m = 125$ naturally aligns the sampled subspace rank with the allocated structural bit budget. Ultimately, users are encouraged to visually inspect this Pareto frontier and are free to adjust the landmark parameter freely based on their specific graph structures and downstream latency constraints.

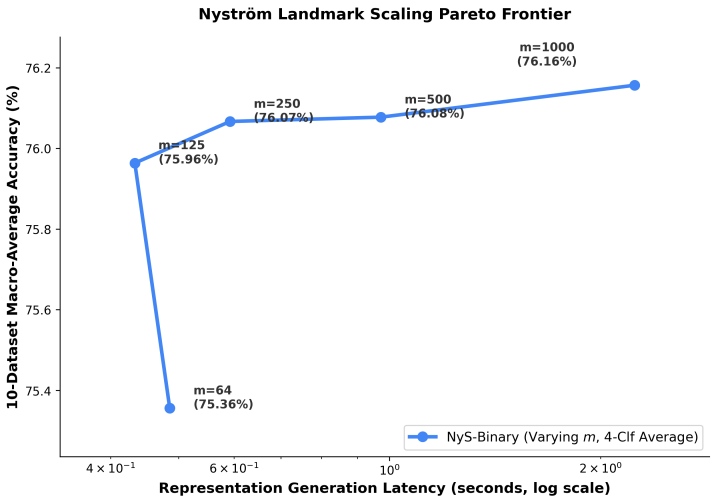


Figure 8: Pareto frontier of Nyström-inspired landmark scaling (10-dataset macro-average across 3 random seeds and 4 classifiers). The default landmark budget $m = 125$ achieves the optimal efficiency Pareto knee: reaching 75.96% accuracy in only 0.433s, capturing nearly all downstream accuracy gains while avoiding the anomalous slowdown at $m = 64$ and the steep decomposition latency of larger budgets.

Table 14: Detailed Nyström-Inspired Landmark Scaling performance across 10 benchmark datasets. The table reports both Macro-Average Accuracy (%) across 4 canonical classifiers (top) and Representation Generation Latency in seconds (bottom) across varying landmark budgets m.

Dataset	$m = 64$	$m = 125$ **(Default)**	$m = 250$	$m = 500$	$m = 1000$
(a) Downstream Classification Accuracy (%) (Macro-Average across 4 Classifiers)					
Cora	83.47	83.39	83.77	83.13	83.68
CiteSeer	69.54	70.07	69.36	69.52	69.19
PubMed	81.72	82.01	82.18	81.94	81.93
WikiCS	79.51	79.31	79.67	79.76	79.98
DBLP	97.31	97.30	97.27	97.30	97.23
Photo	91.41	91.57	91.33	91.31	91.63
Computers	87.91	87.73	88.43	88.50	88.54
Chameleon	64.44	68.14	68.10	68.51	68.18
Squirrel	49.42	51.09	51.57	51.92	52.10
Amazon-Ratings	48.82	49.03	49.00	48.88	49.09
Macro-Average	75.36	75.96	76.07	76.08	76.16
(b) Representation Generation Latency (Seconds)					
Cora	0.079s	0.109s	0.323s	0.885s	2.499s
CiteSeer	0.251s	0.051s	0.159s	0.444s	1.713s
PubMed	0.272s	0.231s	0.305s	0.624s	2.723s
WikiCS	1.009s	1.059s	1.149s	1.100s	2.465s
DBLP	0.208s	0.131s	0.210s	0.521s	1.397s
Photo	0.600s	0.574s	0.830s	1.423s	2.177s
Computers	1.191s	1.074s	1.550s	2.102s	2.601s
Chameleon	0.098s	0.119s	0.209s	0.515s	1.628s
Squirrel	0.696s	0.579s	0.729s	1.408s	3.463s
Amazon-Ratings	0.456s	0.405s	0.458s	0.698s	1.688s
Macro-Average	0.486s	0.433s	0.592s	0.972s	2.236s

B.2.4 Structural vs. Semantic Budget Allocation ($k_{\text{struct}} : k_{\text{label}}$)

Within a total embedding budget of $K = 250$ bits, allocating capacity between structural coordinates (k_{struct}) and semantic label representations ($k_{\text{label}} = K - k_{\text{struct}}$) governs how the binary hash balances topological geometry against task-specific class discriminability. To rigorously analyze this trade-off, we conducted a grand synthesis encompassing 2,520 individual evaluations across 10 benchmark datasets, 3 random seeds, 3 distinct supervision regimes (1%, 30%, and 70% labeled training nodes), 4 probe architectures (GCN, GraphSAGE, MLP, LINKX), and 7 allocation splits. As summarized in Figure 9 and Tables 15 and 16, varying the allocation ratio reveals significant performance shifts across classifier families and supervision levels.

The empirical results demonstrate that single-channel extremes suffer catastrophic failure modes. Allocating 100% of the budget to semantic diffusion ($0/100$ split) deprives the representation of structural scaffolding, reducing accuracy to 64.68% and Macro F1 to 61.69%. Conversely, allocating 100% to structural identity ($100/0$ split) severely degrades models that cannot rely on graph smoothing: while GCN and GraphSAGE remain relatively stable, heterophily-tailored architectures like **LINKX** plunge from 64.70% down to $\mathbf{59.65}\%$ (a steep -5.05% collapse), and non-relational MLPs drop from 65.76% to 63.27%. Heavy structural skew starves heterophilic graphs of class boundary signals, proving that structural and semantic channels serve strictly complementary roles.

The balanced **50/50 split** ($k_{\text{struct}} = 125, k_{\text{label}} = 125$) emerges as a highly robust and balanced configuration. It attains the highest Macro F1 score across all 2,520 runs ($\mathbf{62.87}\%$), demonstrating strong class separation across imbalanced datasets. Furthermore, under severe label scarcity (1% supervision, Table 16), the 50/50 split achieves the absolute highest accuracy (53.35%), effectively regularizing noisy label diffusion with stable landmark geometry. While the 75/25 and 90/10 structural-heavy splits slightly edge out the 50/50 split on average accuracy at higher supervision regimes (30% and 70%), the 50/50 split remains highly competitive across all settings. Because embedding generation latency remains similar across all hybrid configurations, it is omitted here, and the balanced 50/50 allocation is established as our canonical default across all experiments.

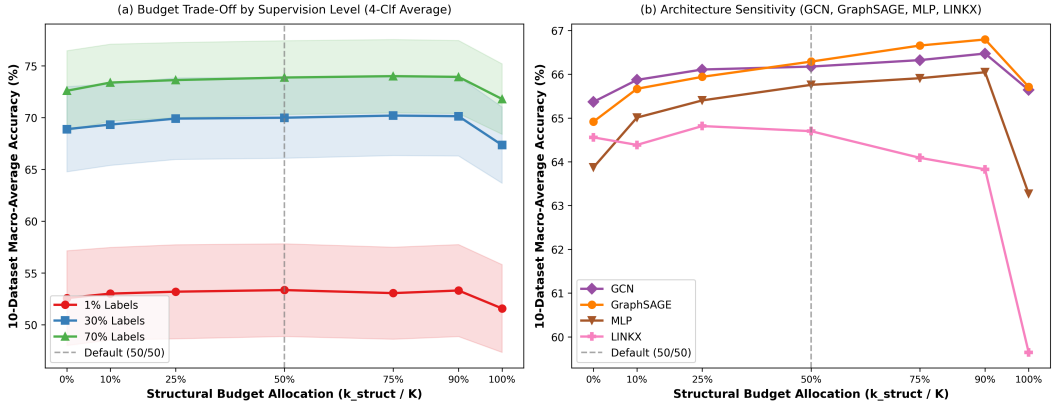


Figure 9: Dimension budget allocation trade-off across $K = 250$ bits. (a) 10-dataset macro-average accuracy (± 1 std dev shaded) under varying supervision regimes (1%, 30%, and 70% labels). (b) Architecture-specific sensitivity across four canonical probes (GCN, GraphSAGE, MLP, and LINKX). The balanced default split (50/50, $k_{\text{struct}} = 125, k_{\text{label}} = 125$) achieves the highest Macro F1 score (62.87%) and provides strong cross-architecture robustness without catastrophic collapse at extremes.

Table 15: Grand synthesis of embedding dimension budget allocation ($k_{\text{struct}} : k_{\text{label}}$) evaluated across 10 benchmark datasets, 3 random seeds, 3 supervision levels (1%, 30%, 70%), and 4 canonical classifier architectures (2,520 evaluations total). The balanced 50/50 split achieves the highest overall Macro F1 score and protects heterophily-tailored probes (LINKX) from structural collapse.

Split (Struct/Label)	k_{struct}	k_{label}	Average (4 Clf) (%)	Macro F1 (%)	GCN (%)	GraphSAGE (%)	MLP (%)	LINKX (%)
0/100 (Pure Label)	0	250	64.68 ± 22.49	61.69 ± 23.99	65.37	64.92	63.87	64.56
10/90 (Heavy Label)	25	225	65.23 ± 21.87	62.27 ± 23.26	65.87	65.67	65.01	64.38
25/75 (Moderate Label)	62	188	65.57 ± 21.93	62.71 ± 23.33	66.11	65.94	65.40	64.82
50/50 (Balanced Default)	**125**	**125**	65.73 ± 21.68	**62.87 ± 23.07**	66.17	66.29	65.76	**64.70**
75/25 (Moderate Struct)	188	62	65.74 ± 21.61	62.85 ± 23.03	66.32	66.65	65.91	64.09
90/10 (Heavy Struct)	225	25	**65.78 ± 21.49**	62.82 ± 22.95	**66.47**	**66.79**	**66.04**	63.83
100/0 (Pure Struct)	250	0	63.57 ± 20.64	60.34 ± 22.50	65.64	65.71	63.27	59.65

Table 16: Detailed budget allocation performance (macro-average over 4 classifiers) varying the k_{struct} / k_{label} split at 1%, 30%, and 70% supervision levels across 10 benchmark datasets and 3 random seeds.

Split (Struct/Label)	k_{struct}	k_{label}	1% Supervision	30% Supervision	70% Supervision
0/100 (Pure Label)	0	250	52.58 ± 22.85	68.88 ± 20.53	72.58 ± 19.36
10/90 (Heavy Label)	25	225	53.01 ± 22.30	69.32 ± 19.64	73.37 ± 18.55
25/75 (Moderate Label)	62	188	53.19 ± 22.68	69.89 ± 19.68	73.61 ± 18.15
50/50 (Balanced Default)	**125**	**125**	**53.35 ± 22.36**	69.98 ± 19.50	73.86 ± 17.79
75/25 (Moderate Struct)	188	62	53.05 ± 22.19	**70.19 ± 19.23**	**73.99 ± 17.69**
90/10 (Heavy Struct)	225	25	53.31 ± 22.19	70.13 ± 19.15	73.92 ± 17.59
100/0 (Pure Struct)	250	0	51.58 ± 21.17	67.34 ± 18.38	71.78 ± 16.99

B.2.5 Scalability Experiment on OGB Datasets

To evaluate the scalability of the proposed **NyS-Binary** framework on massive-scale graphs, we conducted experiments on two large datasets from the Open Graph Benchmark (OGB): **ogbn-arxiv** (169,343 nodes) and **ogbn-products** (∼2.4 million nodes).

For **ogbn-arxiv**, we evaluated the algorithm across 3 random seeds using a 70% label provision ratio during embedding generation. Table 17 presents the macro-averaged results across the classifiers. The generated 250-bit binary representations are computed extremely efficiently (∼6.39 seconds on average for NyS-Binary) while maintaining highly competitive downstream accuracy, demonstrating the robustness of the structural-semantic blending even at scale. The unsupervised variant (relying entirely on graph structure without label diffusion) generates embeddings in ∼6.25 seconds.

For **ogbn-products**, due to extreme resource constraints and the massive size of the graph (2,449,029 nodes), the generation process was executed strictly on CPU. At this 2.4-million-node scale, iterative GNNs and recursive multi-hop sketchers encountered memory exhaustion or prohibitive execution timeouts on standard workstation hardware; hence, comparative baseline latencies are rigorously benchmarked across the ten core datasets, while the ogbn-products experiment validates that NyS-Binary scales successfully to multi-million-node graphs. Remarkably, **NyS-Binary** successfully embedded the entire 2.4 million node graph into 250-bit representations in $\mathbf{1249.08}$ seconds. The resulting binary embeddings achieved strong downstream performance on simple probes: a Logistic Regression probe achieved 86.52% accuracy (Macro F1: 49.25%) with a training time of 187.05 seconds, while an MLP achieved 86.74% accuracy (Macro F1: 49.36%) in 294.22 seconds.

Table 17: Macro-averaged performance of 250-bit binary embeddings on the **ogbn-arxiv** dataset across 3 random seeds (70% train ratio). Generation time remains exceptionally low (∼6.4s) while maintaining strong accuracy and F1 scores.

Method	Classifier	Accuracy (%)	Macro F1 (%)	Gen Time (s)
NyS-Binary	LogisticRegression	72.34	55.77	6.39
NyS-Binary	MLP	72.55	56.01	6.39
NyS-Binary	GCN	72.60	55.72	6.39
NyS-Binary	GraphSAGE	72.63	56.02	6.39
NyS-Binary	LINKX	66.18	50.03	6.39
NyS-Binary	H2GCN	72.77	56.22	6.39
Unsupervised NyS-Binary	LogisticRegression	56.31	29.66	6.25
Unsupervised NyS-Binary	MLP	56.81	30.78	6.25
Unsupervised NyS-Binary	GCN	63.72	36.49	6.25
Unsupervised NyS-Binary	GraphSAGE	63.00	35.92	6.25
Unsupervised NyS-Binary	LINKX	60.98	43.89	6.25
Unsupervised NyS-Binary	H2GCN	63.34	37.02	6.25