STRUCTURAL ENCODING VIA ORTHOGONAL CODE BROADCAST FOR GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks aggregate neighbor messages, entangling individual contributions into a single representation. This limits interpretability and exacerbates over-smoothing in deep architectures. Inspired by the code-division multiple access mechanism from communications, we propose OCMSE (Orthogonal Code based Multi-Scale Structural Encoding), a plug-and-play framework that enables both structured integration and individual separation of multi-source signals during message propagation. Specifically, selected source nodes are assigned orthogonal codewords and broadcast probe signals that traverse the graph via a learnable, edge-modulated mechanism. At each receiving node, correlation detection recovers each source's individual contribution, producing a structural imprint that jointly encodes decomposed source-wise signals and global structural context. Unlike distance-based anchor methods that rely on single-path information, OCMSE captures richer multi-path information without per-node code assignment. Theoretically, OCMSE provides injective structural fingerprints for generic graphs, with strict source separation preserved under refinement. Extensive experiments show that OCMSE consistently improves various backbone GNN architectures on node and graph classification benchmarks, delivering up to 21% relative improvement on symmetric graphs. Our anonymous code is available at `https://anonymous.4open.science/r/OCMSE-BFFB`.

1 INTRODUCTION

Graph neural networks (GNNs) have become the predominant approach for learning on graph-structured data, with message aggregation typically serving as their operational core. In this paradigm, each node iteratively refines its representation by gathering and integrating information from its neighbors. Over the years, a diverse range of aggregation schemes have been proposed to accommodate different graph properties, including degree-normalized summation in GCNs Kipf & Welling (2017), mean or max pooling in GraphSAGE Hamilton et al. (2017), and attention-weighted summation in GATs Veličković et al. (2018). The choice of aggregation function directly determines the expressive power and quality of the learned representations.

Despite their success, existing GNN aggregation schemes suffer from a critical flaw: they indiscriminately mix messages from all participating nodes. This limitation manifests in three interrelated problems. *First*, it restricts a node's ability to perceive global structure or trace individual contributions. Since all neighbor messages are summed or averaged together, the resulting representation provides no way to recover which neighbor contributed what. As the number of nodes grows, this mixing worsens, obscuring each contributor's role. *Second*, repeated indiscriminate aggregation across layers erodes node distinctiveness, leading to over-smoothing—a major obstacle in deep GNN architectures. Consequently, most GNNs are constrained to only a few layers, limiting their receptive fields to predominantly local information. *Third*, indiscriminate aggregation lacks distance awareness: nodes at varying distances should ideally contribute differently, yet existing GNNs provide no mechanism to separate or suppress irrelevant messages once they are mixed.

To address the limitations mentioned above, we propose **Orthogonal Code based Multi-Scale Structural Encoding (OCMSE)**. Inspired by code-division multiple access (CDMA)—a communication technique that enables multiple users to transmit simultaneously over the same channel—OCMSE probes structural features via orthogonal codewords. Specifically, selected source nodes, serving

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.

as approximate graph centers, are assigned orthogonal codewords and broadcast probe signals that propagate hop by hop with learnable adjustments. At each node, correlation detection extracts the contribution from each source. Unlike CDMA, OCMSE jointly leverages both the superimposed global signal and the decoupled individual signals, which together form a structural imprint that captures the node's structural information from multiple dimensions. This imprint serves as a plug-and-play structural enhancement that can be seamlessly integrated with any downstream GNN.

Our contributions can be summarized as follows:

- **Novel aggregation framework.** We propose a novel GNN message aggregation framework that encodes both structured integration and individual separation of multi-source signals into node representations via orthogonal code broadcast. To the best of our knowledge, this is the first work to introduce CDMA-style message aggregation into GNNs.
- **Learnable propagation mechanism.** We design a learnable propagation mechanism that broadcasts orthogonal probe signals from selected source nodes. Each signal propagates hop by hop with learnable adjustments, enabling the model to adaptively capture multi-scale structural information.
- **Adaptive refinement module.** We introduce a source-level attention refinement module that adaptively aggregates multi-source structural imprints by learning global source importance. This reduces redundancy and improves discriminability while maintaining low computational complexity.
- **Experimental validation.** We conduct extensive experiments on both node classification and graph classification tasks. The results demonstrate that OCMSE consistently improves various backbone GNN architectures, delivering competitive performance against state-of-the-art methods.

Comparison with anchor-based methods. Although OCMSE leverages source nodes similar to the anchors in anchor-based methods You et al. (2019); Zhang et al. (2025), it fundamentally differs from them in three aspects. First, OCMSE employs orthogonal codes to capture multi-path propagation, rather than single-path distance vectors. Second, instead of localizing nodes via distance computation, OCMSE gathers structural information through hop-by-hop probe signal broadcast with learnable, edge-modulated adjustments. Third, rather than treating each anchor independently, OCMSE exploits the synergy among multiple signal sources, where the superimposed signals collectively encode global topological information beyond what any individual source could provide.

2 Related Work

GNN aggregation challenges. The message aggregation in GNNs faces three fundamental challenges. First, indiscriminate mixing of messages from all participating nodes restricts a node's ability to perceive global structure or trace individual contributions Ying et al. (2018); Sanders et al. (2025); Liu et al. (2023), and this mixing worsens as the number of nodes grows. Such mixing is also the main cause of over-smoothing Li et al. (2018); Hoang et al. (2021). Second, aggregation lacks distance awareness, as existing GNNs provide no mechanism to separate or suppress irrelevant messages once they are mixed. These first two challenges jointly constitute the structure-awareness problem. Third, over-squashing compresses messages from distant nodes into fixed-length vectors due to graph bottlenecks Alon & Yahav (2020); Topping et al. (2021); Giovanni et al. (2023). Graph transformers offer a paradigm that is inherently less susceptible to the above issues Yun et al. (2019); Chen et al. (2024); Ying et al. (2021); Kreuzer et al. (2021). Nevertheless, addressing the structure-awareness problem remains essential, as resolving it can provide explicit structural signals that complement self-attention, which itself does not inherently encode graph topology.

Structure-awareness of GNNs. Structure-awareness refers to the ability of a GNN to capture and exploit graph topology. Mainstream backbones such as GCN Kipf & Welling (2017), GraphSAGE Hamilton et al. (2017), GAT Veličković et al. (2018), and GIN Xu et al. (2019) achieve structure-awareness implicitly through neighborhood aggregation. However, all these methods rely on the same aggregation paradigm, which indiscriminately mixes neighbor messages and therefore lacks explicit structure-awareness capability.

To overcome this limitation, structure-aware encodings either explicitly inject topology information into graph representations or selectively exploit such information. Representative approaches include Laplacian positional encoding Dwivedi & Bresson (2020); Kreuzer et al. (2021); Beaini et al. (2021); Wang et al. (2022); Lim et al. (2022), shortest-path-distance encoding Li et al. (2020); Ying et al.

(2021), node degree centrality Ying et al. (2021), kernel distance Mialon et al. (2021), random-walk encoding Dwivedi et al. (2022), feature-splitting methods such as GFS Zheng et al. (2025), and other structure-aware methods Chen et al. (2022); Bouritsas et al. (2023); Bodnar et al. (2021). Note that GFS focuses on selectively exploiting the relationship between feature dimensions and graph topology. Some works further derive such encodings through learnable networks Kreuzer et al. (2021); Dwivedi et al. (2022); Lim et al. (2022); Chen et al. (2022). Among these, distance-based encodings are particularly relevant: Distance Encoding Li et al. (2020) augments node features with shortest-path distances, while anchor-based methods such as P-GNN You et al. (2019) exploit distance vectors to anchors. Nevertheless, these approaches capture only single-path or pairwise information, and thus fail to reflect the multi-path propagation that better characterizes graph structures.

3 PRELIMINARIES

3.1 GRAPH MODEL

A graph is denoted as $G = (V, E)$, where V is the set of vertices (or nodes) and E is the set of edges. Let $v_i \in V$ represent a node, and $e_{ij} = (v_i, v_j) \in E$ represent an edge from v_j to v_i. The neighborhood of node v is defined as $\mathcal{N}(v) = \{u \in V \mid (v, u) \in E\}$. The adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$ is defined such that $\mathbf{A}_{ij} = 1$ if $e_{ij} \in E$, and $\mathbf{A}_{ij} = 0$ otherwise.

Node attributes are represented by $\mathbf{X} \in \mathbb{R}^{n \times d}$, where each row corresponds to a node and $\mathbf{x}_v \in \mathbb{R}^d$ denotes the feature vector of node v. Similarly, edge attributes are represented by $\mathbf{X}^e \in \mathbb{R}^{m \times c}$, where $\mathbf{x}^e_{u,v} \in \mathbb{R}^c$ denotes the feature vector of edge (u, v). If edge attributes are not provided, all edges are assigned the same constant vector, e.g., $\mathbf{x}^e_{u,v} = \mathbf{1}_c$. When edge attributes contain multiple dimensions, we designate one specific dimension as the **structural embedding dimension**, denoted by $e^{(\text{struct})}_{uv}$. By default, we use the first dimension, i.e., $e^{(\text{struct})}_{uv} = [\mathbf{x}^e_{u,v}]_1$.

3.2 CODE-DIVISION MULTIPLE ACCESS

As introduced previously, OCMSE is inspired by the communication technique CDMA. In CDMA, each user i is assigned a unique codeword $\mathbf{c}_i \in \{-1, +1\}^L$ of length L. The transmitted signal of user i is $s_i = b_i \cdot \mathbf{c}_i$, where $b_i \in \{-1, +1\}$ is the user's bit. All users transmit simultaneously, and the received superposition is $\mathbf{y} = \sum_{i=1}^{q} b_i \mathbf{c}_i$, where q is the number of users.

The receiver extracts user i's bit via correlation detection:

$$\hat{b}_i = \text{sign}(\mathbf{y} \cdot \mathbf{c}_i) = \text{sign}\left(b_i \|\mathbf{c}_i\|^2 + \sum_{j \neq i} b_j (\mathbf{c}_j \cdot \mathbf{c}_i) \right). \tag{1}$$

Since the codewords are orthogonal ($\mathbf{c}_j \cdot \mathbf{c}_i = 0$ for $j \neq i$), the interference term vanishes, enabling perfect recovery. In OCMSE, the user's bit is replaced by a feature vector, and the sign operator is omitted since message passing involves no noise. Further details are provided in Section 4.3.

4 OUR SOLUTION

4.1 FRAMEWORK

Our proposed solution, OCMSE, consists of three main modules—broadcast signal source selection, probe signal broadcast, and structural imprint refinement—as illustrated in Fig. 1. The broadcast signal source selection module employs the k-center algorithm to select k sources, ensuring uniform structural coverage such that every node lies within a bounded distance of at least one source. The probe signal broadcast module then encodes each node's multi-scale structural information into a compact and discriminative structural imprint. Finally, the structural imprint refinement module applies a source-level attention mechanism to amplify discriminative sources while suppressing redundant ones. The resulting structural imprint can be seamlessly integrated with downstream GNN tasks by concatenating it with the original node attributes.

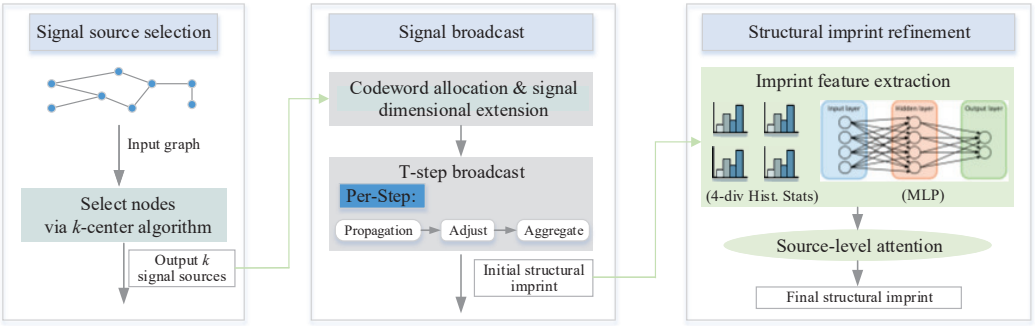


Figure 1: The OCMSE framework.

4.2 Broadcast Signal Source Selection

To achieve balanced structural coverage and efficient signal propagation, OCMSE selects k distributed broadcast signal sources. The choice of k is governed by the tolerable computational and memory overhead incurred by the resulting structural imprints, which scale linearly with the number of sources. By default, we set $k = 10$. To ensure uniform information diffusion, we select k sources using the k-center algorithm. Since our broadcast process propagates signals hop by hop, we define the distance metric as the shortest-path hop count, i.e., $\mathrm{dist}(u, v)$ is the minimum number of edges between u and v. The k-center problem on graphs seeks a set $\mathcal{S} \subset V$ of size k that minimizes the maximum hop distance from any node to its nearest source:

$$\min_{\mathcal{S}\subset V, |\mathcal{S}|=k} \max_{v\in V} \min_{s\in\mathcal{S}} \mathrm{dist}(v, s). \tag{2}$$

According to the definition of hop distance, the following triangle inequality holds: $\mathrm{dist}(a, c) \leq \mathrm{dist}(a, b) + \mathrm{dist}(b, c)$. This metric property is a prerequisite for most existing k-center algorithms (e.g., the greedy 2-approximation algorithm Gonzalez (1985)), as they rely on the triangle inequality to guarantee their approximation bounds. In this paper, we employ the greedy 2-approximation algorithm for graph k-center Gonzalez (1985). This algorithm runs in $O(k \cdot (N + E))$ time using breadth-first search, where N is the number of nodes and E is the number of edges. For large-scale graphs, exact computation of the greedy algorithm may be expensive. In such cases, faster approximate k-center algorithms (e.g., Thorup (2005); Abboud et al. (2023)) can be used.

4.3 Orthogonal Code Based Probe Signal Broadcast

In OCMSE, each signal source broadcasts a probe signal that propagates hop by hop through the graph. The broadcast process proceeds iteratively over T steps, where at each step, nodes receive signals from their neighbors, apply a learnable adjustment modulated by edge-specific structural embeddings, and forward the combined signal.

Initialization. Each broadcast signal source $s \in \mathcal{S}$ is assigned a unique orthogonal codeword $\mathbf{c}_s \in \{-1, +1\}^L$. The codewords are derived from a Hadamard matrix $\mathbf{H}_L$ of order L (where L is a power of 2), with each row serving as an orthogonal spreading code Hadamard (1893); Walsh (1923). The initial probe signal at node u is:

$$\mathbf{Q}_u^{(0)} = \begin{cases} \mathbf{c}_s^\top, & \text{if } u = s \in \mathcal{S}, \\ \mathbf{0}_L, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathbf{Q}_u^{(0)} \in \mathbb{R}^L$. Thus, each source broadcasts its assigned codeword as the initial signal.

Signal propagation and learnable aggregation. Let $\gamma^{(\tau)} \in \mathbb{R}$ be a learnable scalar adjustment factor at propagation step τ. To ensure numerical stability while allowing both attenuation and amplification, we constrain the adjustment factor to a bounded interval $[\gamma_{\min}, \gamma_{\max}]$ via

$$\gamma^{(\tau)} = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min}) \cdot \sigma(\theta^{(\tau)}), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function and $\theta^{(\tau)}$ is a trainable scalar. By default, we set $\gamma_{\min} = 0.1$ and $\gamma_{\max} = 2.0$, allowing the model to learn both attenuation ($\gamma^{(\tau)} < 1$) and amplification ($\gamma^{(\tau)} > 1$) within a stable range.

The adjustment factor at each propagation step is further modulated by the structural embedding dimension $e_{uv}^{(\text{struct})}$ associated with edge (u, v) (as defined in Section 3). This modulation serves two purposes. First, it incorporates edge-level structural information into the propagation process, enabling the adjustment to vary across edges according to their structural properties. Second, it decouples the learnable adjustment $\gamma^{(\tau)}$ from the edge-specific modulation $\sigma(e_{uv}^{(\text{struct})})$.

At propagation step $\tau + 1$ (where $\tau = 0, \ldots, T-1$), each node u receives probe signals from all its neighbors $v \in \mathcal{N}(u)$. The received signals are summed to form the aggregated signal:

$$\mathbf{Q}_u^{(\tau+1)} = \sum_{v \in \mathcal{N}(u)} \gamma^{(\tau)} \cdot \sigma(e_{uv}^{(\text{struct})}) \cdot \mathbf{Q}_v^{(\tau)}, \tag{5}$$

where $\mathbf{Q}_v^{(\tau)}$ denotes the probe signal received by node v at step τ, and $\sigma(\cdot)$ maps the structural embedding dimension $e_{uv}^{(\text{struct})} \in \mathbb{R}$ to a factor in $(0, 1)$. A larger $e_{uv}^{(\text{struct})}$ yields a value closer to 1, resulting in stronger signal retention (i.e., less adjustment). The total per-hop effect is $\gamma^{(\tau)} \cdot \sigma(e_{uv}^{(\text{struct})})$, which can be less than or greater than 1 depending on the learned $\gamma^{(\tau)}$. If a neighbor v has not yet received any broadcast signal at step τ, its signal $\mathbf{Q}_v^{(\tau)}$ is a zero vector.

We adopt a recurrent neural network (RNN) scheme in which the learnable adjustment parameters are shared across steps. In this architecture, the hidden state at step τ corresponds to the probe signals $\mathbf{Q}_u^{(\tau)}$ across all nodes. Further details of the RNN-based implementation are provided in Appendix A.

Structural imprint extraction via correlation detection. After T propagation steps, each node u possesses a superimposed signal $\mathbf{Q}_u^{(T)} \in \mathbb{R}^L$, which encodes global structural information from all broadcast sources. To extract the contribution of each source $s \in \mathcal{S}$, the node performs correlation detection:

$$\alpha_{s \to u} = \frac{1}{L} \langle \mathbf{Q}_u^{(T)}, \mathbf{c}_s \rangle, \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. The normalization factor $1/L$ ensures that the contribution magnitude is independent of the codeword length.

Owing to the orthogonality of the codewords $\{\mathbf{c}_s\}_{s \in \mathcal{S}}$, contributions from different sources are perfectly separated. Specifically, $\alpha_{s \to u}$ equals the cumulative attenuation $\Lambda_T(s \to u)$ from source s to node u, where $\Lambda_T(s \to u)$ is the product of the learnable adjustments and structural embedding values along all propagation paths of length at most T. A closed-form expression for $\mathbf{Q}_u^{(T)}$ in terms of $\Lambda_T(s \to u)$ is provided in Proposition 3 in Appendix B.

We define an initial structural imprint for node u as

$$\mathbf{z}_u^{(T)} = [\alpha_{s_1 \to u}, \alpha_{s_2 \to u}, \ldots, \alpha_{s_k \to u}, \mathbf{Q}_u^{(T)}] \in \mathbb{R}^{k+L}, \tag{7}$$

where $k = |\mathcal{S}|$ is the number of broadcast sources, and $\mathbf{Q}_u^{(T)} \in \mathbb{R}^L$ is the superimposed signal received at node u after T propagation steps. This imprint encodes both the separated contributions from each source and the full superimposed signal, jointly capturing structural information from multiple dimensions.

Advantages of orthogonal code-based signal separation. The use of orthogonal codes in OCMSE offers several key advantages. First, assigning each source a unique orthogonal codeword enables correlation detection to extract individual contributions without interference, directly allevating the indiscriminate mixing problem inherent in standard GNN aggregation. Moreover, since the superimposed signal $\mathbf{Q}_u^{(T)}$ received at each node aggregates information from all sources across the entire graph, the orthogonal code mechanism naturally captures global structural information within a unified framework. Second, each $\alpha_{s \to u}$ directly reflects the cumulative signal attenuation from source s to node u at a specific scale, providing insight into which sources and scales drive each node's representation. Third, the orthogonality property plays a crucial role in interpretability: because the codewords are mutually orthogonal, the contribution from each source is extracted in isolation, free from interference from other sources. Detailed theoretical analysis is provided in Appendix B and C.

4.4 Refinement of Structural Imprint

In the initial structural imprint generation procedure introduced in Section 4.3, the broadcast module treats all sources equally. However, sources are not equally important: some capture critical structural patterns, while others are redundant or noisy. A refinement mechanism is therefore needed to adaptively aggregate the separated source contributions and highlight the most discriminative ones. To this end, we propose an attention mechanism that adaptively refines source importance based on the structural imprint patterns. Crucially, this refinement is applied *only* to the correlation-based source contributions $\{\alpha_{s\to u}\}_{s\in\mathcal{S}}$; the raw superimposed signal $\mathbf{Q}_u^{(T)}$ is preserved unchanged to retain the complete global structural information.

Imprint feature extraction. Recall that each node u obtains separated source contributions $\{\alpha_{s\to u}\}_{s\in\mathcal{S}}$ from correlation detection. For each source s, we have N scalar values $\{\alpha_{s\to u}\}_{u\in V}$ over all nodes.

The source-level representation for source s is obtained by applying a Stats function to aggregate the N scalar values:

$$\boldsymbol{\psi}_s = \text{Stats}\left(\{\alpha_{s\to u}\}_{u\in V}\right) \in \mathbb{R}^H. \tag{8}$$

The function Stats aggregates the N scalar values into a compact H-dimensional vector via histogram-based features followed by an MLP. For each source s, we compute the minimum and maximum, partition $[m_{\min}, m_{\max}]$ into Q equal-width bins, record the min, mean, max, and sum per bin, and pass the concatenated $4Q$ statistics through a lightweight MLP:

$$\boldsymbol{\psi}_s = \text{MLP}\left([\text{min}_1, \text{mean}_1, \text{max}_1, \text{sum}_1, \ldots, \text{min}_Q, \text{mean}_Q, \text{max}_Q, \text{sum}_Q]\right) \in \mathbb{R}^H. \tag{9}$$

This representation preserves distributional information, is independent of N (beneficial for large graphs), and the shared MLP ensures parameter efficiency across all sources.

Source-level attention weights. Once the source representation $\boldsymbol{\psi}_s$ is obtained, we compute the source-level attention weight β_s using a multi-layer perceptron (MLP) followed by softmax normalization:

$$\beta_s = \frac{\exp\left(\text{MLP}(\boldsymbol{\psi}_s)\right)}{\sum_{s'\in\mathcal{S}} \exp\left(\text{MLP}(\boldsymbol{\psi}_{s'})\right)}, \tag{10}$$

where MLP consists of several hidden layers with ReLU activation.

The resulting attention weights satisfy $\sum_{s\in\mathcal{S}} \beta_s = 1$ and are shared across all nodes. This source-level sharing is the key distinction from node-level attention (e.g., GAT): while node-level attention answers "which sources matter for this specific node?" our source-level attention answers "which sources matter globally for the entire graph?" The latter is more suitable for refining structural imprints, as source importance is determined by global structural roles rather than node-specific interactions.

Imprint refinement. Once the source-level attention weights are obtained, each node refines its separated source contributions by scaling each source's contribution. The refined structural imprint for node u is:

$$\hat{\mathbf{z}}_u = \left[\beta_{s_1}\alpha_{s_1\to u}, \beta_{s_2}\alpha_{s_2\to u}, \ldots, \beta_{s_k}\alpha_{s_k\to u},\ \mathbf{Q}_u^{(T)}\right] \in \mathbb{R}^{k+L}, \tag{11}$$

where the raw superimposed signal $\mathbf{Q}_u^{(T)}$ is kept intact. This refinement mechanism amplifies contributions from sources that the model has learned to be globally important, while suppressing less informative ones, without altering the raw global structural information contained in $\mathbf{Q}_u^{(T)}$. Crucially, the refinement is applied after the broadcast module, serving as a post-processing step that does not interfere with the orthogonality guarantees of the broadcast process.

Fusion with node features. Finally, the refined structural imprint is concatenated with the original node features to form the enhanced representation:

$$\mathbf{h}_u = [\mathbf{X}_u \parallel \hat{\mathbf{z}}_u] \in \mathbb{R}^{d+k+L}. \tag{12}$$

This representation can be directly used for node classification or fed into downstream GNN layers for further processing. Since k and L are typically set to small values (e.g., $k = 10$ and $L = 16$ in our experiments), the additional dimensionality introduced by the structural imprint remains low, making the fusion computationally lightweight.

Properties of final structural imprints. The structural imprints $\hat{\mathbf{z}}_u$ possess three key properties. First, the refinement module preserves source separation: scalar attention weights do not mix codewords, maintaining orthogonality. Second, the imprints are discriminative: for graphs without nontrivial automorphisms, the mapping $\Phi(u) = \hat{\mathbf{z}}_u$ is injective, ensuring that each node receives a unique structural fingerprint. Third, the imprints are interpretable: each refined dimension $\beta_s \alpha_{s \to u}$ monotonically reflects the topological distance from source s to node u, modulated by the learned global importance of source s. Detailed properties and their proofs are provided in Appendix C.

4.5 Training for Structural Imprint Learning

The training of the structural imprint learning consists of two stages: (i) **pre-training**, which is essential and performed independently without any downstream task supervision, and (ii) **joint fine-tuning**, which is performed when downstream tasks are available. The pre-trained model can be directly used as a plug-and-play component for various graph learning tasks. In joint fine-tuning, the pre-trained parameters serve as a warm start, and the combined objective depends on the specific task.

We employ a pairwise contrastive loss to maximize the distinguishability of structural imprints, where the dissimilarity between nodes is directly reflected by the difference in their imprints, independent of their graph distance. For any two distinct nodes $u \neq v$, the loss is defined as

$$\ell(u, v) = \max(0, \mathrm{sim}(\hat{\mathbf{z}}_u, \hat{\mathbf{z}}_v) - \alpha) + \max(0, \beta - \mathrm{sim}(\hat{\mathbf{z}}_u, \hat{\mathbf{z}}_v)), \tag{13}$$

where $\mathrm{sim}(\cdot, \cdot)$ ($\mathrm{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$) denotes cosine similarity, α is an upper margin preventing imprints from becoming too similar, and β is a lower margin preventing imprints from becoming too dissimilar, with $0 < \alpha < \beta < 1$. This loss encourages all node pairs to have imprint similarity within the interval $[\alpha, \beta]$, striking a balance between pushing dissimilar nodes apart and pulling similar nodes together, while allowing the model to naturally distinguish nodes through their imprints without relying on predefined graph distance thresholds.

The total pairwise loss is aggregated over all node pairs, and we introduce a diversity regularizer to prevent mode collapse:

$$\mathcal{L}_{\text{pair}} = \frac{1}{N(N-1)} \sum_{u \neq v} \ell(u, v), \qquad \mathcal{L}_{\text{div}} = -\sum_{i=1}^{k} \mathrm{Var}(\{\hat{\mathbf{z}}_u[i]\}_{u \in V}), \tag{14}$$

where Var denotes the variance across nodes for each dimension i of the structural imprint $\hat{\mathbf{z}}_u$. The overall pre-training objective combines both terms:

$$\mathcal{L} = \mathcal{L}_{\text{pair}} + \lambda \mathcal{L}_{\text{div}}, \tag{15}$$

with λ as a balancing hyperparameter.

5 Experimental Evaluation

In this section, we evaluate the effectiveness of OCMSE, which generates structural imprints that are concatenated with existing node attributes and then fed into downstream GNNs. We demonstrate that augmenting baseline GNNs with OCMSE consistently improves their performance.

Datasets. We evaluate OCMSE on standard benchmark datasets for both node classification and graph classification. For node classification, we use Cora Sen et al. (2008), Communities Walker et al. (2020), Citeseer Giles et al. (1998), Amazon-Ratings Platonov et al. (2023), and the large-scale dataset ogbn-arxiv from the Open Graph Benchmark (OGB) Hu et al. (2020). For graph classification, we use NCI1 Rossi & Ahmed (2015), IMDB-MULTI, and IMDB-BINARY Yanardag & Vishwanathan (2015). Additional results on more datasets are provided in Appendix E.

Baselines. We adopt the following GNNs as base models and augment them with OCMSE: **P-GNN** You et al. (2019), a position-aware graph neural network; **GCN** Kipf & Welling (2017), a graph convolutional network; **GraphSAGE** Hamilton et al. (2017), an inductive graph neural network with mean aggregation; **GAT** Veličković et al. (2018), a graph attention network; **GIN** Xu et al. (2019), a graph isomorphism network; and **GFS** Zheng et al. (2025), a graph feature selection method. Among these, GIN is evaluated on NCI1, IMDB-MULTI, and IMDB-BINARY; all remaining experiments are for node classification.

Experiment details. By default, the number of propagation steps is set to $T = 50$ for ogbn-arxiv and $T = 20$ for all other datasets, and the number of broadcast sources is $k = 10$. The histogram bin count is $Q = 10$, and the MLP hidden dimension in the structural imprint refinement is set to 32. The pre-training loss margins are $\alpha = 0.2$ and $\beta = 0.8$, with a balancing factor of $\lambda = 0.01$. The length of the orthogonal codewords is 16. All experiments are repeated with 10 random seeds, and we report the mean accuracy with standard deviation.

5.1 Main Results

Table 1: Accuracy improvement on existing models. **SI** denotes the structural imprints generated by OCMSE, "+" denotes augmentation with OCMSE, and "–" denotes augmentation without OCMSE.

Method	Dataset	SI	Acc. (%)	Method	Dataset	SI	Acc. (%)
P-GNN	Communities	–	99.1 ± 0.3	GAT	Cora	–	77.1 ± 4.9
		+	$\mathbf{99.4 \pm 0.2}$			+	$\mathbf{81.6 \pm 0.7}$
	Email	–	63.8 ± 2.4		Citeseer	–	59.7 ± 10.1
		+	$\mathbf{66.9 \pm 6.8}$			+	$\mathbf{71.6 \pm 0.8}$
	ogbn-arxiv	–	48.7 ± 4.3		ogbn-arxiv	–	44.9 ± 10.3
		+	$\mathbf{53.4 \pm 4.7}$			+	$\mathbf{65.9 \pm 0.1}$
GCN	Cora	–	77.6 ± 5.4	GIN	IMDB-MULTI	–	50.7 ± 4.6
		+	$\mathbf{81.4 \pm 1.2}$			+	$\mathbf{52.5 \pm 2.1}$
	Citeseer	–	65.3 ± 5.3		NCI1	–	70.8 ± 2.1
		+	$\mathbf{69.9 \pm 0.9}$			+	$\mathbf{75.3 \pm 2.1}$
	ogbn-arxiv	–	55.4 ± 9.2		IMDB-BINARY	–	69.9 ± 3.8
		+	$\mathbf{62.8 \pm 0.7}$			+	$\mathbf{72.6 \pm 3.9}$
GraphSAGE	Citeseer	–	59.7 ± 1.5	GFS	amazon-ratings	–	45.2 ± 0.9
		+	$\mathbf{62.3 \pm 1.0}$			+	$\mathbf{51.1 \pm 0.5}$
	ogbn-arxiv	–	66.7 ± 0.8		ogbn-arxiv	–	65.6 ± 0.2
		+	$\mathbf{69.9 \pm 0.8}$			+	$\mathbf{69.3 \pm 0.3}$

Table 1 reports the node and graph classification accuracy across all evaluated datasets. OCMSE consistently improves every backbone GNN. On citation networks, OCMSE raises GAT from 77.1% to 81.6% on Cora and from 59.7% to 71.6% on Citeseer. On the large-scale ogbn-arxiv benchmark, OCMSE yields absolute gains of 3.2%–21.0% across all five backbones, with the largest improvement on GAT. On graph classification tasks, OCMSE also brings consistent gains; for example, GIN improves from 70.8% to 75.3% on NCI1. These results demonstrate that the multi-scale structural imprints provide complementary structural information that effectively enhances node and graph representations across diverse architectures and datasets.

Table 2: Impact of the key hyperparameters. Datasets: ogbn-arxiv; k: number of broadcast sources; T: number of propagation steps.

Parameter	Value	P-GNN	GCN	GFS	GAT	GraphSAGE
k ($T = 50$)	1	51.9 ± 1.4	62.0 ± 1.0	67.3 ± 0.2	64.9 ± 0.2	69.2 ± 0.4
	5	52.6 ± 1.3	61.9 ± 1.0	61.9 ± 1.0	65.5 ± 0.1	69.6 ± 0.6
	10	$\mathbf{53.4 \pm 4.9}$	$\mathbf{62.8 \pm 0.7}$	$\mathbf{69.4 \pm 0.3}$	$\mathbf{65.9 \pm 0.1}$	$\mathbf{69.9 \pm 0.8}$
	15	53.0 ± 2.1	62.2 ± 0.6	69.0 ± 0.3	65.7 ± 0.1	69.3 ± 0.5
T ($k = 10$)	40	52.8 ± 4.4	62.1 ± 0.7	68.9 ± 0.4	65.5 ± 0.1	69.5 ± 0.8
	50	$\mathbf{53.4 \pm 4.9}$	$\mathbf{62.8 \pm 0.7}$	$\mathbf{69.4 \pm 0.3}$	$\mathbf{65.9 \pm 0.1}$	$\mathbf{69.9 \pm 0.8}$
	60	52.4 ± 4.8	62.1 ± 0.7	69.2 ± 0.2	65.7 ± 0.1	69.8 ± 0.9

Table 2 examines the sensitivity of OCMSE to k and T on ogbn-arxiv. Performance peaks at $k = 10$ across all backbones: smaller k provides insufficient structural coverage, while larger k introduces redundant sources and increases imprint dimensionality without further gains. Similarly, $T = 50$ achieves the best performance, as smaller T limits the receptive field and larger T dilutes discriminative signals through excessive propagation. According to Tables 2 and 1, OCMSE outperforms the compared models under all evaluated configurations of k and T, indicating that its effectiveness

is robust to the choice of these two hyperparameters rather than relying on a specific configuration. Additional experimental results are provided in Appendix E.1 and exhibit the same trend.

5.2 Ablation Studies

To evaluate the contribution of each core component, we conduct ablation experiments. We examine five variants: (i) **w/o** $\mathbf{Q}_u^{(T)}$, which removes the raw superimposed signal; (ii) **w/o** $\{\alpha_{s\to u}\}$, which removes all separated source contributions; (iii) **w/o** $\gamma^{(\tau)}$, which removes the learnable step-wise adjustment; (iv) **w/o** $\sigma(e_{uv}^{(\textbf{struct})})$, which removes the edge-specific structural modulation; and (v) **w/o Refinement**, which skips the source-level attention module.

Table 3: Ablation study results. "Full" denotes the complete OCMSE model. Dataset: NCI1 for GIN, and ogbn-arxiv for other models.

Variant	P-GNN	GCN	GFS	GAT	GraphSAGE	GIN
Full OCMSE	**53.4 ± 4.7**	**62.8 ± 0.6**	**69.3 ± 0.3**	**65.9 ± 0.5**	**69.9 ± 0.7**	**75.3 ± 2.1**
w/o $\mathbf{Q}_u^{(T)}$	49.2 ± 1.5	54.1 ± 9.6	65.6 ± 0.2	53.8 ± 1.2	66.3 ± 0.7	70.4 ± 2.4
w/o $\{\alpha_{s\to u}\}$	53.3 ± 1.6	53.2 ± 9.3	65.9 ± 0.3	42.2 ± 9.6	66.7 ± 0.8	70.0 ± 2.1
w/o $\gamma^{(\tau)}$	44.8 ± 8.7	53.4 ± 9.2	65.8 ± 0.1	42.3 ± 9.1	66.4 ± 1.0	71.3 ± 2.9
w/o $\sigma(e_{uv}^{(\text{struct})})$	61.2 ± 1.2	53.1 ± 9.2	66.9 ± 0.1	11.7 ± 8.5	65.2 ± 3.2	71.0 ± 2.1
w/o refinement	46.4 ± 10.0	53.4 ± 9.2	65.9 ± 0.2	42.3 ± 9.0	66.4 ± 1.1	71.4 ± 2.1

Table 3 reports part of the results. Removing $\{\alpha_{s\to u}\}$, $\mathbf{Q}_u^{(T)}$, $\gamma^{(\tau)}$, or $\sigma(e_{uv}^{(\text{struct})})$ all leads to clear performance degradation, confirming that source separation, the raw superimposed signal, and adaptive edge-aware propagation are all essential to OCMSE. Skipping the refinement module also consistently reduces performance, validating the source-level attention mechanism. Notably, GAT suffers the most severe degradation when $\sigma(e_{uv}^{(\text{struct})})$ or $\{\alpha_{s\to u}\}$ is removed, suggesting that attention-based aggregation relies heavily on the separable structural signals provided by OCMSE. Overall, these results validate that all components are complementary and essential for optimal performance.

5.3 Overhead

We evaluate the computational overhead introduced by OCMSE in terms of inference time and memory consumption. To this end, we define two metrics: Increased Reasoning Time (IRT), the increase in inference time (in seconds) when OCMSE is applied to a given backbone GNN, and Increased Reasoning Peak Memory (IRPM), the corresponding increase in peak memory consumption (in MB). Across all evaluated backbones and datasets, the overhead introduced by OCMSE remains modest: the IRT is at most 0.134 seconds and the IRPM at most 289.7 MB, both observed on the large-scale ogbn-arxiv dataset. For small-scale datasets such as Cora, Citeseer, and Communities, the IRT is typically below 0.01 seconds and the IRPM is on the order of tens of MB. In several cases, both metrics are even negative (e.g., relative to P-GNN), indicating that the structural imprints can guide the backbone toward more efficient inference paths. These results confirm that OCMSE is practical for real-world deployment. Detailed experimental results are provided in Table 6 in Appendix E.2.

6 Conclusion

We identified the "crowd mixing" problem in GNNs, where neighbor contributions become entangled after aggregation, limiting interpretability and exacerbating over-smoothing. To address this, we proposed OCMSE, a CDMA-inspired plug-and-play framework that assigns orthogonal codewords to selected sources. These sources broadcast probe signals through a learnable, edge-modulated mechanism, and correlation detection recovers individual contributions, yielding a structural imprint that jointly encodes decomposed signals and global context—capturing richer multi-path information than distance-based anchor methods. Theoretically, OCMSE provides injective structural fingerprints with strict separation preserved under refinement. Empirically, it consistently improves various backbone GNN architectures with competitive performance. Notably, the code-domain separation affords native interpretability. For future work, we plan to extend OCMSE to heterogeneous and dynamic graphs, and to explore adaptive source selection for large-scale graphs.

AI USE STATEMENT

We used large language models (LLMs) exclusively for language polishing and presentation refinement. The LLMs were employed to improve the grammar, clarity, and readability of the manuscript. All AI-generated suggestions were carefully reviewed and revised by the authors. The LLMs played no role in research ideation, methodological design, experimental execution, data analysis, or result interpretation. The authors bear full responsibility for the final content.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our results, we provide the following resources and details. All datasets used in this work are publicly available. Model specifications and hyperparameter configurations are detailed in Section 5. All experiments were repeated with at least 10 different random seeds, and we report both the mean and standard deviation of the results. The source code and pretrained models are publicly available at an anonymous repository (`https://anonymous.4open.science/r/OCMSE-BFFB`). All baseline methods use their official implementations, with hyperparameters tuned following the recommendations in the original papers.

REFERENCES

Amir Abboud, Vincent Cohen-Addad, and Pasin Manurangsi. On the fine-grained complexity of approximating k-center in sparse graphs. In *Symposium on Simplicity in Algorithms*, pp. 145–155, 2023.

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Dominique Beaini, Saro Passaro, Vincent Létourneau, Will Hamilton, Gabriele Corso, and Pietro Liò. Directional graph networks. In *International Conference on Machine Learning*, pp. 748–758, 2021.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido F. Montúfar, and Michael M. Bronstein. Weisfeiler and lehman go cellular: Cw networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2625–2640, 2021.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos P. Zafeiriou, and Michael M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2023.

Chaoqi Chen, Yushuang Wu, Qiyuan Dai, Hong-Yu Zhou, Mutian Xu, Sibei Yang, Xiaoguang Han, and Yizhou Yu. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (12):10297–10318, 2024.

Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *International Conference on Machine Learning*, pp. 3469–3489, 2022.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital Libraries*, pp. 89–98, 1998.

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Liò, and Michael M. Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *International Conference on Machine Learning*, pp. 7865–7885, 2023.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

Jacques Hadamard. Résolution d'une question relative aux déterminants. *Bulletin des Sciences Mathématiques*, 17:240–246, 1893.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1024–1034, 2017.

Ngoc Hoang, Takanori Maehara, and Tsuyoshi Murata. Revisiting graph neural networks: Graph filtering perspective. In *International Conference on Pattern Recognition*, pp. 8376–8383, 2021.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020.

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

Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Letourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21618–21629, 2021.

Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4465–4478, 2020.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence*, pp. 3538–3545, 2018.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20887–20902, 2021.

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013*, 2022.

Yang Liu, Chuan Zhou, Shirui Pan, Jia Wu, Zhao Li, Hongyang Chen, and Peng Zhang. Curvdrop: A ricci curvature based approach to prevent graph neural networks from over-smoothing and over-squashing. In *Proceedings of the ACM Web Conference*, pp. 221–230, 2023.

Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

Galileo Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *Mining and Learning with Graphs*, 2012.

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*, 2023.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI Conference on Artificial Intelligence*, pp. 4292–4293, 2015.

Cedric Sanders, Andreas Roth, and Thomas Liebig. Curvature-based pooling within graph neural networks. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2025.

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.

Mikkel Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM Journal on Computing*, 34(2):405–432, 2005.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv preprint arXiv:2111.14522*, 2021.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Matthew Walker, Bo Yan, Yiou Xiao, Yafei Wang, and Ayan Acharya. Isometric graph neural networks. *arXiv preprint arXiv:2006.09554*, 2020.

Joseph L. Walsh. A closed set of normal orthogonal functions. *American Journal of Mathematics*, 45 (1):5–24, 1923.

Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. Equivariant and stable positional encoding for more powerful graph neural networks. In *International Conference on Learning Representations*, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Pinar Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888, 2021.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, volume 31, pp. 4800–4810, 2018.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *International Conference on Machine Learning*, pp. 7134–7143, 2019.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. Graph transformer networks. In *Advances in Neural Information Processing Systems*, volume 32, pp. 11960–11970, 2019.

Yiqun Zhang, Zhenyue Qin, Saeed Anwar, Dongwoo Kim, Yang Liu, Pan Ji, and Tom Gedeon. Position-sensing graph neural networks: Proactively learning nodes relative positions. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5787–5794, 2025.

Yilun Zheng, Xiang Li, Sitao Luan, Xiaojiang Peng, and Lihui Chen. Let your features tell the differences: Understanding graph convolution by feature splitting. In *International Conference on Learning Representations*, 2025.

APPENDIX

A IMPLEMENTATION OF LEARNABLE SIGNAL BROADCAST ADJUSTMENT

The T-step broadcast process can be implemented in two ways: (i) a layer-wise independent scheme where each propagation step has its own learnable adjustment parameters, and (ii) a recurrent neural network (RNN) scheme where parameters are shared across steps. The layer-wise scheme offers greater expressiveness but introduces $T \cdot D$ trainable parameters, which becomes prohibitive for large-scale graphs. In contrast, the RNN scheme shares the adjustment factor $\boldsymbol{\gamma} \in \mathbb{R}^D$ across all T steps, significantly reducing the parameter count to D and improving memory efficiency. This makes the RNN-based implementation particularly suitable for graphs of varying scales. Therefore, we focus on the RNN scheme in this paper. In this architecture, the hidden state at step τ corresponds to the probe signals $\mathbf{Q}_u^{(\tau)}$ across all nodes.

Specifically, we define a graph-aware RNN cell that processes all nodes in parallel:

$$\mathbf{Q}_u^{(\tau+1)} = \mathbf{Q}_u^{(\tau)} + \text{Adjust}\left(\sum_{v\in\mathcal{N}(u)} \mathbf{Q}_v^{(\tau)},\ \{e_{uv}^{(\text{struct})}\}_{v\in\mathcal{N}(u)}\right), \tag{16}$$

where the Adjust function is parameterized as

$$\text{Adjust}(\mathbf{A}, \{e_{uv}\}) = \sum_{v\in\mathcal{N}(u)} \boldsymbol{\gamma} \odot \left(\sigma(e_{uv}^{(\text{struct})}) \cdot \mathbf{Q}_v^{(\tau)}\right), \tag{17}$$

with $\boldsymbol{\gamma} \in \mathbb{R}^D$ being learnable parameters shared across all time steps.

B Theoretical Analysis of Probe Signal Broadcast

This section provides a theoretical foundation for the probe signal broadcast mechanism introduced in Section 4.3. Throughout this section, let $\mathcal{P}_{s\to u}^{(T)}$ denote the set of all paths from source s to node u of length at most T, where the length of a path P is denoted by $|P|$. The cumulative attenuation from source s to node u is defined as

$$\Lambda_T(s \to u) = \sum_{P\in\mathcal{P}_{s\to u}^{(T)}} \prod_{j=0}^{|P|-1} \gamma^{(j)} \cdot \sigma(e_{v_j v_{j+1}}^{(\text{struct})}), \tag{18}$$

where $\gamma^{(j)} \in [\gamma_{\min}, \gamma_{\max}]$ is the learnable adjustment at step j, and $\sigma(e_{uv}^{(\text{struct})}) \in (0,1)$ is the edge-specific structural modulation.

Proposition 1. *The operations defined in the broadcast process preserve the orthogonality of signals from distinct broadcast sources. Specifically, for any two distinct sources $s \neq t$ and any propagation step τ, the signals $\mathbf{Q}_s^{(\tau)}$ and $\mathbf{Q}_t^{(\tau)}$ remain orthogonal, i.e., $\langle \mathbf{Q}_s^{(\tau)}, \mathbf{Q}_t^{(\tau)} \rangle = 0$.*

Proof. We prove this by induction on τ.

Base case ($\tau = 0$): By the initialization scheme, each source s is assigned a unique codeword $\mathbf{c}_s$ derived from the Hadamard matrix $\mathbf{H}_L$, where L is a power of 2. The Hadamard matrix satisfies $\mathbf{H}_L\mathbf{H}_L^\top = L\mathbf{I}_L$, meaning its rows are mutually orthogonal. Thus, for any two distinct sources $s \neq t$, we have $\langle \mathbf{c}_s, \mathbf{c}_t \rangle = 0$. Therefore, the initial signals $\mathbf{Q}_s^{(0)}$ and $\mathbf{Q}_t^{(0)}$ are orthogonal.

Inductive step: Assume that at step τ, we have $\langle \mathbf{Q}_s^{(\tau)}, \mathbf{Q}_t^{(\tau)} \rangle = 0$ for all $s \neq t$. Consider step $\tau + 1$.

The propagation from step τ to $\tau + 1$ consists of the following operations:

Scalar modulation: For any source s, the signal propagated through edge (u, v) is multiplied by $\gamma^{(\tau)} \cdot \sigma(e_{uv}^{(\text{struct})})$, where $\gamma^{(\tau)}$ is a learnable scalar and $\sigma(e_{uv}^{(\text{struct})}) \in (0, 1)$ is an edge-specific modulation factor. Scalar multiplication preserves orthogonality, as for any two distinct sources $s \neq t$:

$$\langle \gamma^{(\tau)}\sigma(e_{uv}^{(\text{struct})})\mathbf{Q}_s^{(\tau)}, \gamma^{(\tau)}\sigma(e_{uv}^{(\text{struct})})\mathbf{Q}_t^{(\tau)} \rangle = (\gamma^{(\tau)}\sigma(e_{uv}^{(\text{struct})}))^2 \langle \mathbf{Q}_s^{(\tau)}, \mathbf{Q}_t^{(\tau)} \rangle = 0. \tag{19}$$

Aggregation: The aggregated signal at node u is the sum of signals from all neighbors:

$$\mathbf{Q}_u^{(\tau+1)} = \sum_{v\in\mathcal{N}(u)} \gamma^{(\tau)} \cdot \sigma(e_{uv}^{(\text{struct})}) \cdot \mathbf{Q}_v^{(\tau)}. \tag{20}$$

Since this is a linear combination of orthogonal signals from different sources, the resulting aggregated signals for different sources remain orthogonal. For any two distinct sources $s \neq t$, their contributions to $\mathbf{Q}_u^{(\tau+1)}$ are orthogonal by the linearity of the summation operator.

The update operation in our propagation scheme simply sets the new signal to the aggregated signal; no non-linear operations are applied. Therefore, all operations from step τ to $\tau + 1$ are linear and preserve orthogonality.

Thus, by induction, $\langle \mathbf{Q}_s^{(\tau)}, \mathbf{Q}_t^{(\tau)} \rangle = 0$ for all τ and all $s \neq t$. Hence, the broadcast process preserves orthogonality. □

Significance. This proposition guarantees that signals from different sources remain perfectly separable throughout the entire broadcast process. Consequently, correlation detection at any receiving node can extract each source's individual contribution without interference from other sources, directly alleviating the crowd mixing problem identified in the main paper.

Proposition 2. *Let* $\gamma_{\max} = \max_{\tau\in[0,T-1]} \gamma^{(\tau)}$ *and* $\gamma_{\min} = \min_{\tau\in[0,T-1]} \gamma^{(\tau)}$. *Then for any source* s *and any node* u *at distance* $d = \mathrm{dist}(s,u) \le T$, *the cumulative attenuation* $\Lambda_T(s \to u)$ *satisfies*

$$\gamma_{\min}^d \le \Lambda_T(s \to u) \le \gamma_{\max}^d. \tag{21}$$

Moreover, if $\gamma_{\max} < 1$, *the signal contribution decays exponentially with the propagation distance* d.

Proof. For any path $P \in \mathcal{P}_{s\to u}^{(T)}$ of length exactly $d = \mathrm{dist}(s,u)$, the cumulative attenuation along this path is the product of per-hop factors:

$$\Lambda_{\text{path}}(s \to u) = \prod_{j=0}^{d-1} \gamma^{(j)} \cdot \sigma(e_{v_j v_{j+1}}^{(\text{struct})}), \tag{22}$$

where $v_0 = s$, $v_d = u$, and $\{v_j\}_{j=0}^d$ is the path sequence.

Since $\gamma^{(j)} \in [\gamma_{\min}, \gamma_{\max}]$ by definition, and $\sigma(e_{uv}^{(\text{struct})}) \in (0,1)$, we have

$$\gamma_{\min}^d \le \prod_{j=0}^{d-1} \gamma^{(j)} \cdot \sigma(e_{v_j v_{j+1}}^{(\text{struct})}) \le \gamma_{\max}^d. \tag{23}$$

The cumulative attenuation $\Lambda_T(s \to u)$ is the sum over all paths in $\mathcal{P}_{s\to u}^{(T)}$. Since each path's attenuation satisfies the above bound, and all terms are non-negative, the total also satisfies

$$\gamma_{\min}^d \le \Lambda_T(s \to u) \le \gamma_{\max}^d \cdot \left|\mathcal{P}_{s\to u}^{(T)}\right|. \tag{24}$$

When $\gamma_{\max} < 1$, $\gamma_{\max}^d$ decays exponentially with d. Since the number of paths grows at most exponentially with branching factor Δ^d where Δ is the maximum degree, the product $\gamma_{\max}^d \cdot \Delta^d = (\gamma_{\max}\Delta)^d$. For graphs with $\gamma_{\max}\Delta < 1$, the decay is strictly exponential. Even for denser graphs, the per-path attenuation decays exponentially, ensuring that long-range contributions are exponentially suppressed relative to local ones. □

Significance. This proposition establishes that signal strength decays exponentially with propagation distance when $\gamma_{\max} < 1$. This ensures that local structural information dominates each node's structural imprint, while long-range information is gradually attenuated. This property naturally prevents over-smoothing, as distant nodes do not excessively contaminate local representations, unlike standard GNNs where repeated aggregation leads to homogenization.

Proposition 3. *After* T *propagation steps, the received superimposed signal at node* u *admits the following closed-form expression:*

$$\mathbf{Q}_u^{(T)} = \sum_{s\in\mathcal{S}} \Lambda_T(s \to u) \cdot \mathbf{c}_s, \tag{25}$$

where $\Lambda_T(s \to u)$ *is the cumulative attenuation from source* s *to node* u *summed over all paths of length at most* T.

Proof. We prove this by induction on T.

Base case ($T = 0$*)*: At initialization, $\mathbf{Q}_u^{(0)} = \mathbf{c}_s$ if $u = s$, and $\mathbf{0}$ otherwise. This can be written as $\mathbf{Q}_u^{(0)} = \sum_{s\in\mathcal{S}} \Lambda_T(s \to u) \cdot \mathbf{c}_s$, where $\Lambda_T(s \to u) = 1$ if $u = s$ and 0 otherwise. The base case holds.

Inductive step: Assume the expression holds for step $T-1$. At step T, the propagation equation gives:

$$\mathbf{Q}_u^{(T)} = \sum_{v\in\mathcal{N}(u)} \gamma^{(T-1)} \cdot \sigma(e_{uv}^{(\text{struct})}) \cdot \mathbf{Q}_v^{(T-1)} \tag{26}$$

$$= \sum_{v\in\mathcal{N}(u)} \gamma^{(T-1)} \cdot \sigma(e_{uv}^{(\text{struct})}) \cdot \sum_{s\in\mathcal{S}} \Lambda_{T-1}(s \to v) \cdot \mathbf{c}_s \tag{27}$$

$$= \sum_{s\in\mathcal{S}} \left(\sum_{v\in\mathcal{N}(u)} \gamma^{(T-1)} \cdot \sigma(e_{uv}^{(\text{struct})}) \cdot \Lambda_{T-1}(s \to v) \right) \cdot \mathbf{c}_s. \tag{28}$$

By the definition of $\Lambda_T(s \to u)$ as the sum over all paths of length at most T, the term in parentheses is exactly $\Lambda_T(s \to u)$. Therefore, the induction completes. □

Significance. This proposition provides a closed-form expression for the received signal at any node. It reveals that the signal from each source is a monotonic function of the graph distance, modulated by the cumulative adjustment factors along all available paths. This expression also explains why OCMSE captures multi-path information: the summation over all paths in $\mathcal{P}_{s\to u}^{(T)}$ naturally aggregates signals from all routes between source and destination, unlike distance-based methods that only consider a single path.

C Theoretical Analysis of Refined Structural Imprints

This section establishes the key theoretical properties of the refined structural imprints $\hat{\mathbf{z}}_u$, including orthogonality preservation, injectivity, and interpretability.

Proposition 4. *Let the broadcast module produce signals satisfying orthogonality: for any two distinct sources $s \neq t$, the corresponding codewords satisfy $\langle \mathbf{c}_s, \mathbf{c}_t \rangle = 0$. The refinement module scales each source's contribution by a scalar weight $\beta_s \in \mathbb{R}$, i.e., $\hat{\alpha}_{s\to u} = \beta_s \cdot \alpha_{s\to u}$. Then the refinement operation does not break the orthogonality guarantee of the broadcast module.*

Proof. After the broadcast module, the superimposed signal at node u is given by Proposition 3 in Appendix B:

$$\mathbf{Q}_u^{(T)} = \sum_{s\in\mathcal{S}} \Lambda_T(s \to u) \cdot \mathbf{c}_s, \tag{29}$$

where $\Lambda_T(s \to u)$ is the cumulative attenuation from source s to node u, and $\mathbf{c}_s$ are orthogonal codewords satisfying $\langle \mathbf{c}_s, \mathbf{c}_t \rangle = 0$ for $s \neq t$.

The correlation detection step extracts each source's individual contribution:

$$\alpha_{s\to u} = \frac{1}{L} \langle \mathbf{Q}_u^{(T)}, \mathbf{c}_s \rangle = \Lambda_T(s \to u). \tag{30}$$

The refinement module scales each source's contribution by its attention weight β_s. The refined imprint therefore consists of the scaled source contributions

$$\hat{\alpha}_{s\to u} = \beta_s \cdot \alpha_{s\to u} = \beta_s \cdot \Lambda_T(s \to u), \tag{31}$$

which are concatenated with the raw superimposed signal $\mathbf{Q}_u^{(T)}$ to form $\hat{\mathbf{z}}_u$.

For any two distinct sources $s \neq t$, the inner product of their scaled contributions is

$$\langle \beta_s \Lambda_T(s \to u)\mathbf{c}_s,\ \beta_t \Lambda_T(t \to u)\mathbf{c}_t \rangle = (\beta_s \Lambda_T(s \to u))(\beta_t \Lambda_T(t \to u)) \cdot \langle \mathbf{c}_s, \mathbf{c}_t \rangle \tag{32}$$

$$= (\beta_s \Lambda_T(s \to u))(\beta_t \Lambda_T(t \to u)) \cdot 0 = 0. \tag{33}$$

Thus, orthogonality is preserved. Scalar multiplication changes only the magnitude of each signal, not its direction, and therefore does not affect the orthogonality property. □

Significance. This proposition ensures that the refinement module—which adaptively amplifies or suppresses sources based on their global importance—does not compromise the source separation guarantee. The attention weights β_s are scalars applied to each source's contribution uniformly across all nodes, so they do not introduce cross-source interference. This allows OCMSE to enjoy both adaptive refinement and strict interpretability simultaneously.

Proposition 5. *Let $G = (V, E)$ be a connected graph with no nontrivial automorphisms. Let $\mathcal{S}$ be a k-center set of G with $k \geq 2$. Then for any two distinct nodes $u, v \in V$, the refined structural imprints satisfy $\hat{\mathbf{z}}_u \neq \hat{\mathbf{z}}_v$.*

Proof. Suppose for contradiction that $\hat{\mathbf{z}}_u = \hat{\mathbf{z}}_v$ for some $u \neq v$. By the definition of the refined imprint, this implies $\beta_s \cdot \alpha_{s\to u} = \beta_s \cdot \alpha_{s\to v}$ for all $s \in \mathcal{S}$.

Since β_s are non-negative attention weights, they are not all zero (they sum to 1). For each source s with $\beta_s > 0$, we have $\alpha_{s\to u} = \alpha_{s\to v}$. From the correlation detection step, $\alpha_{s\to u} = \Lambda_T(s \to u)$ and $\alpha_{s\to v} = \Lambda_T(s \to v)$. By Proposition 2, $\Lambda_T(s \to u)$ is a monotonic function of the graph distance $\mathrm{dist}(s, u)$ when all $\gamma^{(t)}$ are equal. More generally, for a fixed source s, $\alpha_{s\to u}$ uniquely determines $\mathrm{dist}(s, u)$ under the assumption that $\gamma^{(t)} \in (0, 1)$ and $\sigma(e_{uv}^{(\text{struct})}) \in (0, 1)$ are strictly positive. Thus, $\alpha_{s\to u} = \alpha_{s\to v}$ implies

$$\mathrm{dist}(s, u) = \mathrm{dist}(s, v) \quad \forall s \in \mathcal{S}. \tag{34}$$

Let R be the optimal k-center radius. For any node $w \in V$, by the k-center property, there exists a source $s_w \in \mathcal{S}$ such that $\mathrm{dist}(s_w, w) \leq R$. By the triangle inequality, for any $w \in V$:

$$\mathrm{dist}(u, w) \leq \mathrm{dist}(u, s_w) + \mathrm{dist}(s_w, w) = \mathrm{dist}(v, s_w) + \mathrm{dist}(s_w, w) \leq \mathrm{dist}(v, w) + 2R. \tag{35}$$

Similarly, $\mathrm{dist}(v, w) \leq \mathrm{dist}(u, w) + 2R$. This establishes that u and v have identical distance profiles to all nodes up to a bounded constant $2R$.

For graphs without nontrivial automorphisms, the distance vectors to a sufficiently large and well-distributed set of sources uniquely determine node identity. Since $\mathcal{S}$ is a k-center set with $k \geq 2$, it provides a distinguishing set of landmarks. The equality of distance vectors to all sources in $\mathcal{S}$ implies $u = v$, contradicting $u \neq v$. Therefore, $\hat{\mathbf{z}}_u \neq \hat{\mathbf{z}}_v$. □

Significance. This proposition establishes the discriminative power of OCMSE's structural imprints. For graphs without nontrivial automorphisms, the mapping from nodes to their imprints is injective, guaranteeing that every node receives a unique structural fingerprint. This distinguishes OCMSE from standard GNNs, which often produce identical embeddings for structurally symmetric nodes.

Proposition 6. *Let $\mathcal{G} = (V, E)$ be a connected graph with $|\mathcal{S}| = k \geq 2$ broadcast sources selected via the k-center algorithm. Let $\mathbf{z}_u^{(T)} \in \mathbb{R}^{k+L}$ denote the structural imprint of node u after T propagation steps:*

$$\mathbf{z}_u^{(T)} = [\alpha_{1\to u}, \alpha_{2\to u}, \ldots, \alpha_{k\to u}, \mathbf{Q}_u^{(T)}], \tag{36}$$

where $\alpha_{s\to u} = \Lambda_T(s \to u)$ is the cumulative attenuation from source s to node u. Then, under the condition $\gamma_{\max} < 1$, for any two distinct nodes $u \neq v$:

$$\lim_{T\to\infty} \|\mathbf{z}_u^{(T)} - \mathbf{z}_v^{(T)}\|_2 > 0. \tag{37}$$

That is, OCMSE prevents node representations from collapsing to identical vectors as $T \to \infty$, thereby alleviating over-smoothing.

Proof. We establish the result in three steps.

Step 1: Strict monotonicity of attenuation with distance. From the definition of $\Lambda_T(s \to u)$:

$$\Lambda_T(s \to u) = \sum_{P \in \mathcal{P}_{s\to u}^{(T)}} \prod_{j=0}^{|P|-1} \gamma^{(j)} \cdot \sigma(e_{v_j v_{j+1}}^{(\text{struct})}). \tag{38}$$

Since $\gamma^{(j)} \in [\gamma_{\min}, \gamma_{\max}]$ with $\gamma_{\max} < 1$ and $\sigma(e_{uv}^{(\text{struct})}) \in (0, 1)$, for a fixed source s, $\Lambda_T(s \to u)$ is a strictly decreasing function of $\mathrm{dist}(s, u)$:

$$\mathrm{dist}(s, u) < \mathrm{dist}(s, v) \implies \Lambda_T(s \to u) > \Lambda_T(s \to v). \tag{39}$$

Step 2: Existence of a distinguishing source. Since $\mathcal{S}$ is a k-center set with $k \geq 2$, for any two distinct nodes $u \neq v$, there exists at least one source $s^* \in \mathcal{S}$ such that

$$\mathrm{dist}(s^*, u) \neq \mathrm{dist}(s^*, v). \tag{40}$$

If $\mathrm{dist}(s, u) = \mathrm{dist}(s, v)$ for all $s \in \mathcal{S}$, then u and v would have identical distance vectors to all sources, contradicting the distinguishing property of the k-center set (which provides coverage and separation for $k \geq 2$).

Step 3: Bounding the representation difference. From the definition of the structural imprint:

$$\|\mathbf{z}_u^{(T)} - \mathbf{z}_v^{(T)}\|_2^2 \geq \sum_{s \in \mathcal{S}} (\Lambda_T(s \to u) - \Lambda_T(s \to v))^2 \tag{41}$$

$$\geq (\Lambda_T(s^* \to u) - \Lambda_T(s^* \to v))^2 > 0. \tag{42}$$

Taking the limit as $T \to \infty$: although $\lim_{T\to\infty} \Lambda_T(s \to u)$ may approach 0 when $\gamma_{\max} < 1$, nodes at different distances from source s^* decay at different rates. If $\mathrm{dist}(s^*, u) < \mathrm{dist}(s^*, v)$, then

$$\lim_{T\to\infty} \Lambda_T(s^* \to u) > \lim_{T\to\infty} \Lambda_T(s^* \to v). \tag{43}$$

Therefore,

$$\lim_{T\to\infty} \|\mathbf{z}_u^{(T)} - \mathbf{z}_v^{(T)}\|_2^2 \geq \lim_{T\to\infty} (\Lambda_T(s^* \to u) - \Lambda_T(s^* \to v))^2 > 0. \tag{44}$$

Hence, the structural imprints of distinct nodes remain distinguishable even in the infinite propagation limit. □

Significance. This proposition establishes that OCMSE alleviates the over-smoothing problem. While traditional GNNs suffer from node representations converging to the same vector as depth increases, OCMSE guarantees that each node maintains a distinguishable structural fingerprint through its multi-source distance profile encoded in $\{\alpha_{s\to u}\}_{s\in\mathcal{S}}$, provided that $\gamma_{\max} < 1$. The orthogonal code mechanism ensures that source contributions remain separable, preventing the information loss that typically causes over-smoothing.

Proposition 7. *If G contains no nontrivial automorphisms and $\mathcal{S}$ is a k-center set with $k \geq 2$, the mapping $\Phi : V \to \mathbb{R}^{k+L}$ defined by $\Phi(u) = \hat{\mathbf{z}}_u$ is injective.*

Proof. Assume, for contradiction, that Φ is not injective. Then there exist two distinct nodes $u \neq v$ such that $\Phi(u) = \Phi(v)$, i.e., $\hat{\mathbf{z}}_u = \hat{\mathbf{z}}_v$.

By the definition of the refined structural imprint, $\hat{\mathbf{z}}_u = \hat{\mathbf{z}}_v$ implies that for every source $s \in \mathcal{S}$, the scaled contributions are equal:

$$\beta_s \cdot \alpha_{s\to u} = \beta_s \cdot \alpha_{s\to v}. \tag{45}$$

Since $\mathcal{S}$ is a k-center set with $k \geq 2$, at least two sources have positive attention weights. For each such source, $\alpha_{s\to u} = \alpha_{s\to v}$. By Proposition 2, $\alpha_{s\to u}$ is monotonically decreasing with $\mathrm{dist}(s, u)$ when $\gamma^{(t)} \in (0, 1)$. Therefore, $\alpha_{s\to u} = \alpha_{s\to v}$ implies

$$\mathrm{dist}(s, u) = \mathrm{dist}(s, v) \quad \forall s \in \mathcal{S}. \tag{46}$$

For a graph with no nontrivial automorphisms, the distance vectors to any set of landmarks that is sufficiently large to distinguish all nodes uniquely identify each node. Since $\mathcal{S}$ is a k-center set, it is distributed across the graph and satisfies this distinguishing property. Hence, the equality of distance vectors implies $u = v$, contradicting the assumption $u \neq v$.

Therefore, Φ must be injective. □

Significance. This proposition formalizes the injectivity of the imprint mapping. Combined with the previous proposition, it provides a rigorous guarantee that OCMSE's structural imprints are discriminative, ensuring that no two distinct nodes are assigned the same structural fingerprint in asymmetric graphs.

D PROPERTIES OF TRAINING PROCEDURE

This section analyzes the optimization properties of the training procedure for structural imprint learning. We establish gradient stability and examine the behavior of the contrastive loss.

D.1 GRADIENT STABILITY OF LEARNABLE ADJUSTMENTS

The learnable adjustment parameters $\theta^{(\tau)}$ control the propagation attenuation through the sigmoid-constrained transformation $\gamma^{(\tau)} = \sigma(\theta^{(\tau)})$, where $\gamma^{(\tau)} \in (0, 1)$. The gradient of the adjustment factor with respect to its underlying parameter is bounded, ensuring stable optimization.

Proposition 8. *For each propagation step τ, the learnable adjustment parameter $\theta^{(\tau)}$ with $\gamma^{(\tau)} = \sigma(\theta^{(\tau)})$ has bounded gradient:*

$$\left|\frac{\partial \gamma^{(\tau)}}{\partial \theta^{(\tau)}}\right| \leq \frac{1}{4}. \tag{47}$$

This ensures stable optimization during training.

Proof. The sigmoid function satisfies $\sigma'(x) = \sigma(x)(1-\sigma(x))$ for all $x \in \mathbb{R}$. The maximum of $\sigma'(x)$ occurs when $\sigma(x) = 1/2$, yielding $\sigma'(x) \leq 1/4$. Since $\gamma^{(\tau)} = \sigma(\theta^{(\tau)})$, the chain rule gives

$$\left|\frac{\partial \gamma^{(\tau)}}{\partial \theta^{(\tau)}}\right| = |\sigma'(\theta^{(\tau)})| \leq \frac{1}{4}. \tag{48}$$

Thus, the gradients of the adjustment factors with respect to their learnable parameters are bounded by a constant independent of the parameter values, ensuring that training remains stable and avoiding vanishing or exploding gradients. □

Significance. This bounded gradient property guarantees that the learnable propagation adjustments do not introduce optimization instability, even when the model is trained for many steps or on large-scale graphs. The sigmoid parameterization provides a principled way to learn attenuation factors while maintaining numerical robustness.

D.2 CONVERGENCE BEHAVIOR OF THE CONTRASTIVE LOSS

We further analyze the convergence behavior of the pairwise contrastive loss used for pre-training. The loss encourages the cosine similarity of structural imprints for any two distinct nodes to lie within the interval $[\alpha, \beta]$, where $0 < \alpha < \beta < 1$.

Proposition 9. *The pairwise contrastive loss $\ell(u, v)$ is non-negative and attains its minimum value of 0 if and only if $\mathrm{sim}(\hat{\mathbf{z}}_u, \hat{\mathbf{z}}_v) \in [\alpha, \beta]$.*

Proof. By definition, $\ell(u, v) = \max(0, s - \alpha) + \max(0, \beta - s)$, where $s = \mathrm{sim}(\hat{\mathbf{z}}_u, \hat{\mathbf{z}}_v) \in [-1, 1]$.

Both terms are non-negative as they are ReLU functions. The loss is zero if and only if both terms are zero, which occurs exactly when $s \leq \alpha$ and $s \geq \beta$, i.e., $s \in [\alpha, \beta]$.

Since cosine similarity is bounded in $[-1, 1]$ and $\alpha, \beta \in (0, 1)$, the loss is well-defined and finite for all node pairs. □

Significance. This proposition establishes that the contrastive loss has a well-defined minimum corresponding to the desired similarity interval. It does not impose a single fixed target similarity, but rather a range, allowing the model flexibility to learn discriminative imprints while preventing both mode collapse (all imprints too similar) and over-separation (all imprints too dissimilar).

E ADDITIONAL EXPERIMENTAL RESULTS

Additional experiments use more datasets compared with those introduced in Section 5. For node classification, we use Cora Sen et al. (2008), Communities Walker et al. (2020), Citeseer Giles et al.

(1998), Pubmed Namata et al. (2012), Amazon-Ratings Platonov et al. (2023), Tolokers Lim et al. (2021), Roman-Empire Lim et al. (2021), as well as the large-scale dataset ogbn-arxiv from the Open Graph Benchmark (OGB) Hu et al. (2020). For graph classification, we use NCI1 Rossi & Ahmed (2015), IMDB-BINARY Yanardag & Vishwanathan (2015), and IMDB-MULTI Yanardag & Vishwanathan (2015).

E.1 Accuracy Behaviour with Different Parameters

Tables 4 and 5 report the accuracy of each backbone GNN under different numbers of broadcast sources k and propagation steps T across a wide range of datasets. Overall, the trends observed are consistent with those reported in the main paper: OCMSE consistently improves the backbone models, with the largest gains typically achieved at moderate values of k and T. This consistency across diverse datasets—spanning citation networks, social networks, molecular graphs, and large-scale benchmarks—suggests that the benefits of OCMSE are not tied to particular graph domain or scale.

Table 4: Accuracy (%) under different k values ($T = 20$). "+" denotes augmentation with OCMSE.

Method	**Dataset**	$k = 5$		$k = 10$		$k = 15$	
		+OCMSE	Gain	+OCMSE	Gain	+OCMSE	Gain
P-GNN	Communities	99.2 ± 0.3	+0.2	99.4 ± 0.2	+0.4	99.2 ± 0.3	+0.2
	Email	65.1 ± 7.3	+1.3	66.9 ± 6.7	+3.1	66.1 ± 6.9	+2.3
GCN	Cora	80.7 ± 0.7	+3.1	81.4 ± 1.2	+3.8	81.1 ± 0.8	+3.5
	Citeseer	69.4 ± 1.0	+4.1	69.9 ± 0.9	+4.6	69.7 ± 1.0	+4.4
	Pubmed	79.5 ± 0.6	+2.1	79.5 ± 0.6	+2.1	79.4 ± 0.3	+2.0
GAT	Cora	81.7 ± 0.6	+4.6	81.8 ± 0.9	+4.7	81.6 ± 0.8	+4.5
	Citeseer	71.6 ± 0.5	+11.9	71.7 ± 0.7	+12.0	71.0 ± 1.3	+11.3
	Pubmed	78.6 ± 0.6	+10.6	78.7 ± 0.8	+10.7	78.2 ± 0.7	+10.2
GraphSAGE	REDDIT	69.5 ± 0.2	+0.8	70.4 ± 0.2	+1.7	70.1 ± 0.2	+1.4
	Citeseer	62.1 ± 1.0	+2.4	62.3 ± 1.0	+2.6	62.3 ± 1.0	+2.6
GIN	NCI1	74.6 ± 2.1	+3.8	75.3 ± 2.1	+4.5	73.7 ± 2.0	+2.9
	IMDB-BINARY	71.7 ± 3.4	+1.8	72.6 ± 3.8	+2.7	72.1 ± 3.3	+2.2
	IMDB-MULTI	51.3 ± 3.8	+0.6	52.5 ± 2.1	+1.8	51.9 ± 2.9	+1.2
GFS	tolokers	83.3 ± 0.7	+3.3	83.7 ± 0.7	+3.7	83.5 ± 0.7	+3.5
	amazon-ratings	50.9 ± 0.6	+5.7	51.1 ± 0.5	+5.9	51.0 ± 0.7	+5.8
	roman-empire	77.3 ± 0.7	+1.3	78.1 ± 0.7	+2.1	78.6 ± 0.7	+2.6

P-GNN. In Table 4, P-GNN is evaluated on Communities and Email. For varying k with $T = 20$, the best performance is achieved at $k = 10$, where P-GNN improves by $+0.4$ on Communities and $+3.1$ on Email. In Table 5, varying T with $k = 10$ shows that the optimal value is $T = 15$ on Email ($+2.9$), while the gains on Communities remain stable at $+0.1$ across all T values. This indicates that OCMSE is robust to moderate variations in propagation depth. Notably, the gain on Email is substantially larger than on Communities, suggesting that P-GNN, as a position-aware model, benefits more from the explicit structural signals provided by OCMSE when the graph is structurally more complex.

GCN. In Table 4, GCN is evaluated on Cora, Citeseer, and Pubmed. The improvements are notably larger than those of P-GNN. Across all three datasets, performance peaks at $k = 10$, with gains of $+3.8$, $+4.6$, and $+2.1$ on Cora, Citeseer, and Pubmed, respectively. In Table 5, GCN is additionally evaluated on Communities and Email. The gains remain consistently positive across all T values, with the best performance at $T = 15$ on Email ($+4.2$) and $T = 25$ on Communities ($+3.2$). This demonstrates the robustness of OCMSE across different hyperparameter choices. The substantial gain on these datasets is particularly encouraging, as it indicates that OCMSE scales effectively to graphs of varying sizes. We attribute this to the fact that the structural imprints provide a global structural prior that complements the purely local aggregation of GCN, which becomes increasingly insufficient as the graph grows.

Table 5: Accuracy (%) under different T values ($k = 10$). "+" denotes augmentation with OCMSE.

Method	Dataset	$T = 10$		$T = 15$		$T = 25$	
		+OCMSE	Gain	+OCMSE	Gain	+OCMSE	Gain
P-GNN	Communities	99.1 ± 0.4	+0.1	99.1 ± 0.4	+0.1	99.1 ± 0.4	+0.1
	Email	65.1 ± 6.5	+2.7	66.7 ± 6.8	+2.9	65.9 ± 6.0	+2.1
GCN	Communities	80.0 ± 0.7	+2.3	80.2 ± 0.7	+2.6	80.8 ± 1.2	+3.2
	Email	69.1 ± 0.8	+3.8	69.5 ± 0.8	+4.2	69.2 ± 1.1	+3.9
	Pubmed	78.6 ± 0.3	+1.2	79.0 ± 0.3	+1.6	78.7 ± 0.6	+1.3
GAT	Cora	80.8 ± 0.5	+3.7	81.5 ± 0.8	+4.4	81.7 ± 0.8	+4.6
	Citeseer	70.7 ± 1.1	+11.0	71.6 ± 0.9	+11.9	71.5 ± 1.0	+11.8
	Pubmed	78.5 ± 0.6	+10.5	78.5 ± 0.5	+10.5	77.7 ± 0.6	+9.7
GraphSAGE	REDDIT	69.6 ± 0.2	+0.9	69.6 ± 0.2	+0.9	70.1 ± 0.3	+1.4
	Citeseer	61.4 ± 1.1	+1.7	62.1 ± 1.0	+1.7	62.1 ± 1.0	+2.4
GIN	NCI1	74.7 ± 2.6	+3.9	75.1 ± 1.7	+4.3	74.7 ± 2.9	+3.9
	IMDB-BINARY	71.7 ± 4.1	+1.8	71.9 ± 2.0	+2.0	72.2 ± 2.5	+2.3
	IMDB-MULTI	50.9 ± 3.1	+0.2	51.7 ± 3.3	+1.0	50.9 ± 2.6	+0.2
GFS	tolokers	83.4 ± 0.7	+3.4	83.4 ± 0.7	+3.4	83.5 ± 0.7	+3.5
	amazon-ratings	50.9 ± 0.6	+5.7	51.1 ± 0.6	+5.9	51.0 ± 0.5	+5.8
	roman-empire	77.5 ± 0.7	+1.5	77.8 ± 0.7	+1.8	77.9 ± 0.7	+1.9

GAT. In Table 4, GAT is evaluated on Cora, Citeseer, and Pubmed. GAT benefits the most from OCMSE among all node classification backbones, with gains exceeding $+10$ on Citeseer and Pubmed across all configurations. The optimal configuration is $k = 10$, achieving gains of $+4.7$, $+12.0$, and $+10.7$ on the three datasets, respectively. In Table 5, the gains remain consistently high across all T values, with Citeseer reaching $+11.9$ at $T = 15$ and Pubmed reaching $+10.5$ at both $T = 10$ and $T = 15$. These results suggest that attention-based aggregation particularly benefits from the separable structural signals provided by OCMSE. A plausible explanation is that GAT's attention mechanism attempts to learn which neighbors are important, but without explicit structural cues, it can only rely on node features. OCMSE supplies the missing structural dimension, allowing the attention mechanism to attend more effectively to structurally relevant neighbors.

GraphSAGE. In Table 4, GraphSAGE is evaluated on REDDIT and Citeseer. The gains are more moderate than those of GAT and GCN, but still consistently positive for $k \geq 5$. The best configuration is $k = 10$, achieving gains of $+1.7$ on REDDIT and $+2.6$ on Citeseer. In Table 5, the gains remain positive across all T values, with the best performance at $T = 25$ on both REDDIT ($+1.4$) and Citeseer ($+2.4$). At $k = 1$ (not shown in the table), the gain on REDDIT is slightly negative, suggesting that a single source is insufficient to provide meaningful structural coverage on this dataset. This observation is consistent with our theoretical analysis: with only one source, the structural imprint degenerates into a single scalar per node, which cannot capture the multi-scale structural information that OCMSE is designed to provide.

GIN. In Table 4, GIN is evaluated on three graph classification datasets: NCI1, IMDB-BINARY, and IMDB-MULTI. The largest gains are observed on NCI1, reaching up to $+4.5$ at $k = 10$. On IMDB-BINARY and IMDB-MULTI, OCMSE yields consistent positive gains, demonstrating its effectiveness on graph-level tasks. In Table 5, the gains remain stable across all T values, with NCI1 consistently achieving around $+3.9$ to $+4.3$. The gains on graph classification tasks are generally smaller than those on node classification, which is expected: graph-level tasks require aggregating information across the entire graph, and the structural imprint of each node contributes only partially to the final graph representation. Nevertheless, the consistent positive gains confirm that OCMSE's structural information is beneficial even when the final prediction is at the graph level.

GFS. In Table 4, GFS is evaluated on three node classification datasets: tolokers, amazon-ratings, and roman-empire. OCMSE consistently improves GFS, with gains of up to $+3.7$ on tolokers, $+5.9$ on amazon-ratings, and $+2.6$ on roman-empire. The best performance is achieved at $k = 10$ across all three datasets. In Table 5, the gains remain stable across all T values, with amazon-ratings consistently achieving around $+5.7$ to $+5.9$. Notably, GFS also benefits substantially on amazon-ratings, where

the baseline performance is relatively low, suggesting that OCMSE provides valuable complementary structural information for models that struggle to capture structural signals. This is particularly encouraging because it indicates that OCMSE can compensate for the structural limitations of the backbone, rather than merely amplifying an already strong structural prior.

Summary. Across all six backbones and all evaluated datasets, OCMSE consistently improves performance when $k \geq 5$, with the optimal configuration being $k = 10$ and $T = 50$ for ogbn-arxiv and $T = 20$ for other datasets. The results in Tables 4 and 5 confirm that the default hyperparameters used in the main paper generalize well across datasets, and that OCMSE is robust to moderate variations in k and T. The consistent gains across diverse graph domains and scales underscore the generality of the proposed framework, and suggest that the structural imprints capture a fundamental aspect of graph structure that is complementary to existing GNN aggregation schemes.

E.2 Computational Overhead of OCMSE

We evaluate the computational overhead introduced by OCMSE in terms of both inference time and memory consumption. To quantify this, we define two metrics:

- **Increased Reasoning Time (IRT)**: the increase in inference time (in seconds) when OCMSE is applied to a given backbone GNN, compared to the inference time of the backbone alone.
- **Increased Reasoning Peak Memory (IRPM)**: the increase in peak memory consumption (in MB) during inference when OCMSE is applied to a given backbone GNN, compared to the peak memory of the backbone alone.

Table 6 reports the computational overhead of OCMSE in terms of IRT and IRPM across all backbones and datasets. Overall, the overhead introduced by OCMSE is modest: the IRT is at most 0.134 seconds for GCN on ogbn-arxiv, and the IRPM is at most 289.7 MB for GraphSAGE on ogbn-arxiv. In several cases, the overhead is even negative, indicating that OCMSE-augmented models can be more efficient than their vanilla counterparts, likely because the structural imprints allow the backbone to converge to a simpler effective computation.

Table 6: Computational overhead of OCMSE on citation networks.

P-GNN & GCN & GraphSAGE				**GAT & GIN & GFS**			
Method	**Dataset**	**IRT (s)**	**IRPM (MB)**	**Method**	**Dataset**	**IRT (s)**	**IRPM (MB)**
P-GNN	Communities	0.004	1.4	GAT	Cora	0.003	15.9
	Email	0.024	-6.27		Citeseer	0.004	47.7
	ogbn-arxiv	-0.006	-158.9		ogbn-arxiv	0.133	15.9
GCN	Cora	0.004	15.5	GIN	NCI1	0.004	11.9
	Citeseer	0.004	67.7		IMDB-BINARY	0.073	6.1
	ogbn-arxiv	0.134	187.9		IMDB-MULTI	0.103	6.4
GraphSAGE	Citeseer	0.003	-4.0	GFS	amazon-ratings	0.004	19.2
	ogbn-arxiv	0.051	289.7		ogbn-arxiv	0.051	74.2

For small-scale datasets such as Cora, Citeseer, Email, and Communities, the IRT is typically below 0.01 seconds, and the IRPM is on the order of tens of MB. For larger datasets such as ogbn-arxiv, the overhead increases but remains acceptable for practical deployment: the largest IRT is 0.134 seconds for GCN, and the largest IRPM is 289.7 MB for GraphSAGE. These values correspond to a small fraction of the total inference time and memory footprint of the backbones on this large-scale dataset.

Interestingly, OCMSE occasionally reduces both inference time and memory peak (negative IRT and IRPM). For example, P-GNN on ogbn-arxiv shows an IRT of -0.006 seconds and an IRPM of -158.9 MB, and GraphSAGE on Citeseer shows an IRPM of -4.0 MB. This suggests that the structural imprints can guide the backbone toward more efficient inference paths, reducing redundant computation in some cases. Although these negative values are relatively small in magnitude, they highlight an additional benefit of OCMSE: by providing explicit structural information, the backbone may avoid unnecessary message passing iterations, leading to more efficient inference.

In summary, the overhead introduced by OCMSE is consistently small across all datasets and backbones, and the framework is therefore practical for real-world deployment. The combination of consistent accuracy improvements and modest computational overhead makes OCMSE an attractive choice for enhancing existing GNN architectures.