GRAPHLCP: Structure-Aware Localized Conformal Prediction on Graphs

Peyman Baghershahi[1] **Fangxin Wang**[1] **Debmalya Mandal**[2] **Sourav Medya**[1]
[1]Department of Computer Science, University of Illinois Chicago
[2]Department of Computer Science, University of Warwick
[1]{pbaghe2, fwang51, medya}@uic.edu; [2]debmalya.mandal@warwick.ac.uk

Abstract

Conformal prediction (CP) provides a distribution-free approach to uncertainty quantification with finite-sample guarantees. However, applying CP to graph neural networks (GNNs) remains challenging as the combinatorial nature of graphs often leads to insufficiently certain predictions and indiscriminative embeddings. Existing methods primarily rely on embedding-space proximity for localization, which can be unreliable for graphs and yield uneven coverage across different graph regions. We propose GRAPHLCP, a proximity-based localized CP framework that incorporates graph topology and inter-node dependencies into localization and weighting. Our approach introduces a feature-aware densification step to mitigate locality bias in sparse graphs, followed by a Personalized PageRank-based kernel computation to model structural proximity. This enables topology-dependent anchor sampling and calibration weighting that captures both local and long-range dependencies. Extensive experiments on regression and classification datasets demonstrate that GRAPHLCP improves test-conditional coverage without excessive conservativeness, while preserving finite-sample marginal coverage.

1 Introduction

Graph neural networks (GNNs) are increasingly deployed for drug discovery [10], weather prediction [30], and fraud detection in financial networks [7]. In these high-stakes applications, relying solely on point predictions is risky and the practitioners require reliable uncertainty quantification (UQ) of GNNs to assess the trustworthiness of predictions. Conformal Prediction (CP) is an effective distribution-free framework for constructing set-valued predictions that guarantees covering the ground-truth values using a finite sample of data points [38, 1]. Consequently, CP has emerged as a widely adopted approach for post-hoc reliability, applied to graph tasks such as standard regression, link prediction, and out-of-distribution detection [27, 31, 49].

Classical CP guarantees *marginal* coverage on average over the test distribution. However, in many practical settings, uncertainty is heterogeneous, requiring reliability specific to subpopulations or local contexts [36, 6]. Achieving *exact* conditional coverage per sample is known to be impossible in finite samples [11]. This limitation has motivated a growing body of work on *relaxed* notions of conditional coverage, leveraging localization and weighting strategies to improve validity [43, 3]. A central requirement across these approaches is the availability of a meaningful notion of *proximity* between the test sample and the calibration data.

In graphs, predictive reliability varies across regions with different homophily patterns [8]. Besides, applying CP on graphs is challenging because nodes are naturally dependent, violating standard i.i.d assumptions [28, 8]. In practice, CP for GNNs often leverages the embedding space, using learned representations as proxies for instances [26]. However, this proxy can be unreliable as message passing can suffer from a graph-topological bottleneck (over-squashing), restricting global

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

information flow and exacerbating noise under heterophily [15]. Meanwhile, deeper message passing causes over-smoothing, reducing the variance of similarity measures [37]. Also, GNNs generally exhibit under-confidence [45, 19], suggesting insufficient separability in the learned representation. These effects complicate the core requirement of having a stable proximity kernel for localized CP, which reweights calibration data based on their similarity to the test instance [18].

Recent work has extended conformal prediction to GNNs to address structural dependencies, with approaches ranging from handling inductive covariate shift [47, 8] to developing diffusion-based scores [46]. Built on findings which preserve exchangeability under permutation-invariant transformation [25, 28, 29], recent studies have also applied CP to GNNs in transductive settings [20, 48]. Additional related work is provided in the Appendix A.2.

However, several limitations remain. Some methods train auxiliary correction or residual models [20, 48], potentially reducing the samples available for calibration, or additionally train the base predictor [26]. Moreover, these approaches do not fully exploit fine-grained topological proximities to capture local variations in prediction uncertainty. Existing localized methods typically rely on either *embedding-space* weighting [26] or *purely structural* partitioning [48]. Embedding similarities may become indiscriminative due to smoothing, while rigid structural clusters may overlook local heterophily, yielding overly conservative or poorly localized prediction sets.

In this work, we bridge the gap between structural and feature-based localization. We propose GRAPHLCP, a localized CP framework for GNNs that incorporates graph topology and inter-node dependencies into conformal calibration. Our approach replaces embedding-only proximity with a Personalized PageRank (PPR) kernel for topology-dependent anchor sampling and calibration weighting. To mitigate locality bias in sparse graphs, we introduce feature-aware densification using embedding similarity before kernel computation. We further provide theoretical results characterizing how over-smoothing weakens embedding-space separability and how graph connectivity influences structural localization boundaries. Together, these enable the preservation of marginal coverage while improving conditional coverage in a favorable balance with prediction size. Experiments on seven regression and six classification datasets demonstrate that GRAPHLCP achieves the target marginal coverage while improving worst-slab coverage without excessive conservativeness.

2 Preliminaries and Problem Setup

2.1 Graph Neural Network (GNN)

In this work, we focus on the canonical form of GNNs, which use a message-passing mechanism. We denote the input graph as $G = (V, E)$, where V and E are its node and edge sets, with $N = |V|$; $\mathbf{X} \in \mathbb{R}^{N\times d}$ and $\mathbf{A} \in \{0,1\}^{N\times N}$ are its feature and adjacency matrices. According to [13], an L-layer pretrained message-passing model ϕ derives a node's l-th layer state or representation, $\mathbf{x}_u^{(l)} (\forall u \in V,\, l \leq L)$ using three functions – (i) a *message-passing* function M_ϕ that propagates the l-th layer state of every node in the neighborhood $\mathcal{N}_u = \{v : v \in V,\, \mathbf{A}_{uv} = 1\}$, (ii) an *aggregate function* C_ϕ to combine the messages from M_ϕ, and (iii) an *update function* U_ϕ to derive the node's latest state integrating its $(l-1)$-th layer state and the aggregated message. Formally, $\mathbf{x}_u^{(l)} = U_\phi(C_\phi(\{M_\phi(\mathbf{x}_v^{(l)}); v \in \mathcal{N}_u\}), \mathbf{x}_u^{(l-1)})$. Given a downstream task, most GNNs use a simple function (e.g., a linear neural network) to decode the latest embeddings or states to prediction logits as $\hat{y}_u = f(\mathbf{x}_u^{(L)})$. Also, for the ease of notation, we use: $\mathbf{x}_i : \forall i \in [N]$ instead of $\mathbf{x}_u^{(L)} : \forall u \in V$, and also y_i instead of $\hat{y}_u$ for $i \in [N]$.

In contrast to standard conformal prediction, where samples are assumed to be exchangeable, conformal validity on graphs is not trivial because node samples are generally dependent, particularly in inductive settings. However, in the node-level transductive setting with a random train/calibration/test split, calibration and test nodes remain exchangeable as long as their labels are not used during training. Once the GNN is trained and frozen, it induces a deterministic transformation from each node and its topological context to an embedding representation, which is then used for conformal prediction. Since conventional GNNs are permutation equivariant and the resulting nonconformity scores are permutation-invariant with respect to calibration and test node permutations, standard conformal prediction guarantees marginal node-level coverage under this transductive setup [20, 28].

2.2 Conformal Prediction (CP)

Split Conformal Prediction (SCP). Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) : (\mathbf{x}_i, y_i) \sim \mathcal{P} = \mathcal{P}_X \times \mathcal{P}_{Y|X},\ \mathbf{x}_i \in \mathcal{X},\ y_i \in \mathcal{Y}\}_{i=1}^{N}$ (e.g., $\mathbf{x}_i$ are the nodes or their embeddings), we first split the dataset to three disjoint sets: 1) $\mathcal{D}_{tr}$ with labels for supervised training, 2) $\mathcal{D}_{cal}$ with labels as holdout calibration set, and 3) $\mathcal{D}_{test}$ purposefully made without labels as test point. After training the message-passing (ϕ) and decoder (f) modules on $\mathcal{D}_{tr}$, we freeze the model for CP. Having n labeled samples $\{(\mathbf{x}_i, y_i) \in \mathcal{D}_{cal} : i \in [n]\}$, the problem is to construct a prediction interval $\widehat{C}_n(\mathbf{x}_{n+1})$ for an unlabeled sample $\mathbf{x}_{n+1} \in \mathcal{D}_{test}$ that contains or "covers" the groundtruth label y_{n+1} of the test sample with probability at least $1-\alpha$, where α is the miscoverage rate.

In order to achieve the desired coverage, every calibration sample and the test sample are assigned a non-conformity as $S_i = s(\mathbf{x}_i, y_i), \forall i \in [n]$ and $S_{n+1} = s(\mathbf{x}_{n+1}, y) : y \in \mathcal{Y}$, where $s(x, y)$ is a non-conformity score function fixed independently of the calibration and test observations. For example, absolute residual, $s(x, y) = |y - f(x)|$, is a commonly used function. Additionally, let $\hat{q} = \text{Quantile}_{1-\alpha}(\sum_{i=1}^{n} \frac{1}{n+1}\delta_{s(\mathbf{x}_i, y_i)} + \frac{1}{n+1}\delta_{+\infty})$, where δ_x is the probability distribution that places probability 1 on the value x, and 0 everywhere else. If S_{n+1} is exchangeable with $\{S_i\}_{i=1}^{n}$, then by constructing $\widehat{C}_n(\mathbf{x}_{n+1}) = \{y \in \mathcal{Y} : s(\mathbf{x}_{n+1}, y) \leq \hat{q}\}$ we have $\mathbb{P}\{y_{n+1} \in \widehat{C}_n(\mathbf{x}_{n+1})\} \geq 1 - \alpha$ [44]. This, being averaged over all random draws of the calibration and the test samples, is called *marginal coverage*. However, if the coverage is guaranteed for every specific instance of $\mathbf{x}_{n+1}$, it is called *conditional coverage*, requiring $\mathbb{P}\{y_{n+1} \in \widehat{C}_n(\mathbf{x}_{n+1}) \mid \mathbf{x}_{n+1}\} \geq 1 - \alpha$. [11] shows that achieving such test conditional coverage is infeasible. Therefore, test conditional coverage is commonly relaxed and measured for a test sample $\mathbf{x}_{n+1} \in B$ where $B \in \mathcal{B}$ and $\mathcal{B}$ is a collection of subsets of feature space, i.e. $\mathcal{B} \subseteq \mathcal{X}$, and the guarantee requires $\mathbb{P}\{y_{n+1} \in \widehat{C}_n(\mathbf{x}_{n+1}) \mid \mathbf{x}_{n+1} \in B : B \in \mathcal{B}\} \geq 1 - \alpha$.

Weighted Conformal Prediction (WCP). Weighted Conformal Prediction (WCP) is one of the ways to attain the above notion of relaxed test conditional coverage [43]. Suppose we are given a kernel function $H(\mathbf{x}, \mathbf{x}')$, e.g. $H(\mathbf{x}, \mathbf{x}') = \frac{1}{(2\pi h^2)^{d/2}} \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2h^2}\right)$. Then we can apply WCP for local coverage by using the kernel centered at $\mathbf{x} = \mathbf{x}_{n+1}$, and obtain coverage for a nearby (synthetic) sample $\tilde{\mathbf{x}}_{n+1} = \mathbf{x}_{n+1} + h \cdot \mathbf{w}$, where $\mathbf{w}$ is chosen so that $\tilde{\mathbf{x}}_{n+1}$ follows the feature distribution $\mathcal{P}_X$ reweighted by $H(\cdot, \mathbf{x}_{n+1})$. This guarantees coverage for a synthetic test pair $(\tilde{\mathbf{x}}_{n+1}, \tilde{y}_{n+1})$, where $\tilde{y}_{n+1} \mid \tilde{\mathbf{x}}_{n+1} \sim \mathcal{P}_{Y|X}$, i.e. $\mathbb{P}\{\tilde{y}_{n+1} \in \widehat{C}_n(\tilde{\mathbf{x}}_{n+1})\} \geq 1 - \alpha$, but not necessarily for $\mathbf{x}_{n+1}$ itself. For small bandwidth h, this provides a locally-weighted relaxation of test-conditional coverage.

Randomly Localized Conformal Prediction (RLCP). Following WCP, Randomly Localized Conformal Prediction (RLCP) [18] guarantees relaxed test conditional coverage for the test sample $\mathbf{x}_{n+1}$, rather than a point $\tilde{\mathbf{x}}_{n+1}$ in its proximity. Let $H : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a normalized localization kernel satisfying $\int_{\mathcal{X}} H(x, x')\, d\nu(x') = 1$ for all $x \in \mathcal{X}$. Given a test point $\mathbf{x}_{n+1}$, RLCP samples an anchor $\tilde{\mathbf{x}}_{n+1} \mid \mathbf{x}_{n+1} \sim H(\mathbf{x}_{n+1}, \cdot)$ and constructs the prediction set/interval for the original test point as $\widehat{C}_n(\mathbf{x}_{n+1}, \tilde{\mathbf{x}}_{n+1}) = \{y \in \mathcal{Y} : s(\mathbf{x}_{n+1}, y) \leq \hat{q}_{1-\alpha}(\mathbf{x}_{n+1}, \tilde{\mathbf{x}}_{n+1})\}$, where

$$\hat{q}_{1-\alpha}(\mathbf{x}_{n+1}, \tilde{\mathbf{x}}_{n+1}) = \text{Quantile}_{1-\alpha}\left(\sum_{i=1}^{n} \tilde{w}_i \delta_{s(\mathbf{x}_i, y_i)} + \tilde{w}_{n+1}\delta_{+\infty}\right), \tag{1}$$

with $\tilde{w}_i = \frac{H(\mathbf{x}_i, \tilde{\mathbf{x}}_{n+1})}{\sum_{j=1}^{n+1} H(\mathbf{x}_j, \tilde{\mathbf{x}}_{n+1})}$, for $i \in [n+1]$.

Under i.i.d. calibration and test observations, for a pair $(X, \widetilde{X})$ s.t. $X \sim \mathcal{P}_X$ and $\widetilde{X} \mid X \sim H(X, \cdot)$, RLCP's relaxed conditional-coverage guarantee can be characterized by the following theorem.

Theorem 1 (Relaxed conditional coverage of RLCP [18]) *For any measurable $B \subseteq \mathcal{X}$ with $\mathcal{P}_X(B) > 0$, let $B^c = \mathcal{X} \setminus B$. Given a metric d on $\mathcal{X}$, define* $\text{bd}_d(B; r) := \{x \in B : \inf_{x' \in B^c} d(x, x') \leq r\}$. *Then*

$$\mathbb{P}\left\{y_{n+1} \in \widehat{C}_n(\mathbf{x}_{n+1}, \tilde{\mathbf{x}}_{n+1}) \mid \mathbf{x}_{n+1} \in B\right\} \geq 1 - \alpha - \inf_{\delta > 0} \frac{\mathbb{P}\{d(X, \widetilde{X}) > \delta\} + \mathcal{P}_X(\text{bd}_d(B; 2\delta))}{\mathcal{P}_X(B)}. \tag{2}$$

Theorem 1 bounds the excess conditional miscoverage through: 1) normalized $\mathbb{P}\{d(\mathbf{X}, \widetilde{\mathbf{X}}) > \delta\}$, which measures how often the sampled anchor lies outside of its corresponding test point's local

neighborhood, and 2) normalized $\mathcal{P}_X(\mathrm{bd}_d(B; 2\delta))$, which measures the probability mass of points in B that lie close to its complement B^c. Thus, the bound is informative when the localization kernel keeps anchors nearby and the chosen distance preserves separation between B and B^c. Motivated by this decomposition, we first study how over smoothing affects the boundary term in GNN embedding space and then examine how this term can be controlled using graph topology.

2.3 Embedding-space boundary under over-smoothing

We first investigate the boundary term in Theorem 1 when localization is defined by distances between GNN embeddings. In particular, we examine whether over-smoothing causes nodes in a fixed nonempty proper subset $B \subsetneq V$ to become indistinguishable from nodes in its complement B^c. For the boundary analysis, assume that G is finite, connected, simple, undirected, and unweighted. Write $Z = (z_u)_{u \in V} \in \mathbb{R}^{N \times p}$ for its node representations, where $z_u := \mathbf{x}_u$ denotes the frozen GNN embedding of node u. Let $d_Z(u, v) := \|z_u - z_v\|_2$ denote the embedding-induced distance. We write $\mathrm{bd}_Z(B; r) := \mathrm{bd}_{d_Z}(B; r)$, and let $\rho_B = |B|/N$ and $\sigma_1 > 0$ be the smallest nonzero eigenvalue of the Laplacian of G, and $\mathcal{E}_G(Z) := \sum_{\{u,v\} \in E} \|z_u - z_v\|_2^2$ denote the Dirichlet energy of the embedding signal Z on G.

Theorem 2 (Embedding-space boundary saturation) *For every embedding matrix Z and every $\delta > 0$,*

$$\frac{|\,\mathrm{bd}_Z(B; 2\delta)|}{|B|} \geq \left[1 - \frac{\mathcal{E}_G(Z)}{4\delta^2 \rho_B(1 - \rho_B) N \sigma_1}\right]_+, \qquad [a]_+ := \max\{a, 0\}. \tag{3}$$

Moreover, let $Z_K \in \mathbb{R}^{N \times p_K}$ be the node representations produced by a K-layer GNN on the fixed graph G. If $\mathcal{E}_G(Z_K) \to 0$ as $K \to \infty$, then, for every fixed B, $\delta > 0$, and sufficiently large K $\mathrm{bd}_{Z_K}(B; 2\delta) = B$.

Theorem 2 shows that, under uniform node sampling, over-smoothing (vanishing Dirichlet energy) causes the boundary penalty in RLCP's conditional-coverage bound to reach its maximum value at any fixed positive embedding radius, making the corresponding bound uninformative due to the loss of geometric separation. As established in Corollary 7 in Appendix A.1.1, a sufficient condition for this energy decay is that the GNN layers are uniformly contractive and preserve zero, and at finite depth, the theorem also provides a lower bound on the boundary fraction when the energy remains positive, with tighter guarantees as energy decreases. These findings motivate using graph topology as an alternative notion of proximity for localized conformal calibration. The proof of the theorem and further discussion are provided in Appendix A.1.1.

Our Objective. We address the problem of CP for GNNs by explicitly incorporating structural proximity into the weighting step. By decoupling weights from potentially lossy encoders, we create a lower-noise residual pathway that leverages embedding proximity while controlling its influence. *Our goal is to satisfy marginal coverage while achieving strong test-conditional coverage with reasonable prediction sizes.* Theorem 2 identifies a precise loss of embedding-space separation under energy decay. Our empirical analysis (Appendix A.3) examines whether embedding-based localization yields poorly concentrated or nearly uniform weights at the depths and bandwidths used in practice. We use graph topology to supply a complementary notion of proximity.

3 Our Method: GRAPHLCP

Recent studies attempted to incorporate structural covariates into GNN-based conformal prediction (CP) [8, 48], yet they have failed to significantly improve conditional coverage. While graph structures are implicitly encoded in GNN embeddings, lossy encoders or sparse topologies can create bottlenecks, and over-smoothing in denser graphs often diminishes topological dependencies [14, 34, 37]. To mitigate this, we propose GRAPHLCP, a structure-aware CP method that ensures marginal coverage while achieving high relaxed test conditional coverage. Our approach utilizes a localized kernel and a weighting strategy, illustrated in Figure 1, which remains resilient to embedding suboptimality by explicitly incorporating underlying structural dependencies (empirical evidence demonstrating the strong correlation between node homophily and coverage is provided in Appendix A.4).

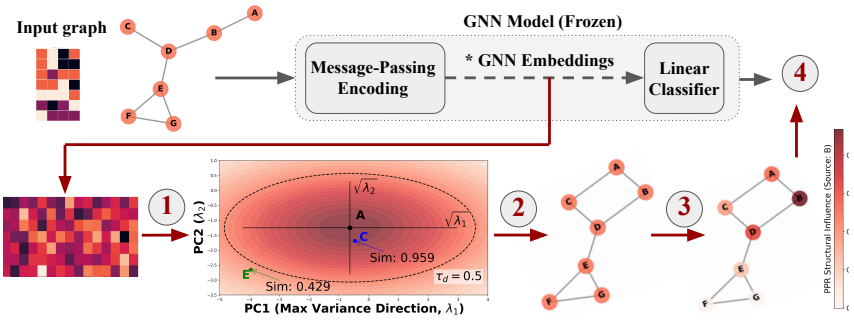


Figure 1: **GRAPHLCP**. Given the embeddings of a frozen GNN for an input graph, GRAPHLCP goes through four stages: 1) **PCA Transformation**: high-dimensional embeddings are decorrelated and largest eigenvalues (corresponding to directions of highest variance) yield an adaptive bandwidth for an anisotropic Gaussian kernel; 2) **Graph Densification**: adding informative edges using a fixed similarity threshold; 3) **PPR Integration**: utilizing the PPR kernel to sample anchor nodes (for each test sample) and weight calibration samples; and 4) **Conformal Prediction**: a weighted quantile is computed from calibration samples to construct the final prediction set/interval.

3.1 Sampling and Weighting with Personalized PageRank (PPR)

To model structural proximity, we adopt Personalized PageRank (PPR) [21] as our localization kernel. PPR captures multi-hop influence while maintaining locality via a restart probability, effectively modeling uncertainty propagation in heterogeneous graphs [33]. Unlike shortest-path or k-hop neighborhoods, PPR provides a soft, probabilistic notion of proximity that naturally aligns with weighted CP. Importantly, PPR yields a valid probability distribution over nodes, allowing it to be directly used as a randomized localization kernel compatible with RLCP.

PPR seeded at node u is formally defined as $\boldsymbol{\pi}_u = \beta \mathbf{e}_u (\mathbf{I} - (1-\beta)\mathbf{M})^{-1}$, where β is the restart/teleportation probability, $\mathbf{e}_u$ is the one-hot seed vector with $[\mathbf{e}_u]_u = 1$ and $[\mathbf{e}_u]_v = 0$ for $v \neq u$, and $\mathbf{M}$ is the row-stochastic transition matrix of the weighted graph. PPR yields a vector representing the stationary distribution of a random walk that, at each step, restarts at u with probability β. The value $[\boldsymbol{\pi}_u]_v$ denotes the long-run probability of the walk being at node v if started at node u. Let $L \sim \text{Geom}(\beta)$ denote a random walk length with $\mathbb{P}(L = k) = \beta(1-\beta)^k$ for $k = 0, 1, \ldots$. Then exact PPR is

$$\boldsymbol{\pi}_u = \mathop{\mathbb{E}}_{L \sim \text{Geom}(\beta)} \left[\mathbf{e}_u \mathbf{M}^L \right] = \sum_{k=0}^{\infty} \beta(1-\beta)^k \mathbf{e}_u \mathbf{M}^k. \tag{4}$$

We approximate exact PPR using K restart iterations [12], initialized at a seed node u with $\boldsymbol{\pi}_u^{(0)} = \mathbf{e}_u$ and updated by $\boldsymbol{\pi}_u^{(t+1)} = (1-\beta)\boldsymbol{\pi}_u^{(t)}\mathbf{M} + \beta \mathbf{e}_u$. The resulting finite kernel is

$$H_K(u, \cdot) = \boldsymbol{\pi}_u^{(K)} = \beta \sum_{k=0}^{K-1} (1-\beta)^k \mathbf{e}_u \mathbf{M}^k + (1-\beta)^K \mathbf{e}_u \mathbf{M}^K. \tag{5}$$

To sample an anchor, we draw L as above and return the endpoint of an $L_K = \min(L, K)$-step weighted random walk from u using $\mathbf{M}$. This yields exactly $H_K(u, \cdot)$, as longer walks are capped rather than discarded, preserving total probability mass. We use the same kernel and transition matrix for anchor sampling and calibration weighting, with $K = 30$ in our experiments.

Anchor Sampling. Analogous to RLCP, in graph space, we construct prediction sets by sampling an anchor node in the neighborhood of the test node and computing conformal scores relative to this anchor. In GRAPHLCP, anchor nodes are sampled from the finite PPR-induced distribution centered at the test node, so the nearby nodes exert greater influence on the quantile estimation. Given a test node u_{n+1}, we sample

$$\tilde{u}_{n+1} \mid u_{n+1} \sim H_K(u_{n+1}, \cdot) = \boldsymbol{\pi}_{n+1}^{(K)}. \tag{6}$$

Structural Weighting. Once the anchor node is sampled, calibration and test nodes are weighted according to their structural proximity to the anchor. Unlike Gaussian kernels, PPR is asymmetric,

meaning structural influence is direction-dependent. Essentially, we have $H_K(u_i, u_j) \neq H_K(u_j, u_i)$, and $\tilde{w}_i \propto H_K(u_i, \tilde{u}_{n+1})$ which requires computing $\boldsymbol{\pi}_i^{(K)} : \forall i \in [n]$. In cases where we have n calibration samples and m test samples s.t. $n < m$, it is efficient to compute $\boldsymbol{\pi}_i^{(K)} : \forall i \in [n]$ directly. However, when $n > m$ (e.g., in a single-sample/pointwise case), this approach is inefficient. To resolve this issue for undirected graphs, we compute weights efficiently by exploiting the degree-normalized symmetry property, which also holds for the finite kernel: $d_i H_K(u_i, u_j) = d_j H_K(u_j, u_i)$, where d_i denotes the weighted degree of u_i in the graph used for PPR. In particular, after densification and isolated-node handling, $d_i = \tilde{\mathbf{D}}_{ii}$. Thus $d_i[\boldsymbol{\pi}_i^{(K)}]_j = d_j[\boldsymbol{\pi}_j^{(K)}]_i$. Letting $\tilde{\boldsymbol{\pi}}_{n+1} = \boldsymbol{\pi}_{\tilde{u}_{n+1}}^{(K)}$ denote the finite PPR distribution seeded at the sampled anchor $\tilde{u}_{n+1}$, the calibration and test nodes are weighted according to the anchor node as follows

$$\tilde{w}_i = \frac{d_i^{-1}[\tilde{\boldsymbol{\pi}}_{n+1}]_i}{\sum_{j=1}^{n+1} d_j^{-1}[\tilde{\boldsymbol{\pi}}_{n+1}]_j}, \quad \forall i \in [n+1] \tag{7}$$

Intuitively, nodes that exert stronger structural influence on the anchor node contribute more to the conformal quantile, adapting the graph context. This topological control of nodes' contributions particularly helps with conditional coverage by filtering out noisy information and focusing on structurally proximal nodes.

Under the transductive random-split setting, permutation-invariant nonconformity scores preserve the exchangeability of calibration and test nodes [20]. Furthermore, conditioning on the sampled anchor reweights the probability of each node being the test node proportionally to its localization kernel value. Since GraphLCP uses these same kernel values for weighted calibration, the weighted conformal argument yields marginal coverage, provided the kernel and scores are fixed independently of calibration/test roles and labels.

Remark. While the PPR-based kernel effectively captures structural similarity between nodes, it does not incorporate node attributes. At the same time, although GNN embeddings are imperfect and can be noisy due to information bottlenecks and over-smoothing, they determine the prediction logits and therefore remain an important source of information for uncertainty quantification. In addition, the PPR-based proximity measure behaves differently depending on graph density. In dense graphs, PPR explores many paths via random walks and captures global structure, whereas in sparse graphs it performs poorly due to strong localization [32], since the random walk probability mass drops rapidly, leading to negligible scores for distant nodes. This causes biased or overly localized weighting.

To address these limitations, we introduce feature-aware graph densification before computing the PPR kernel. Theorem 2 shows that, if Dirichlet energy tends to zero, every node in B eventually becomes close in embedding space to B^c at any fixed positive radius. At finite depth, we hypothesize that relative embedding similarities can still identify useful edges even as absolute distances shrink. We therefore use thresholded similarities to augment the topology, while PPR on the augmented graph determines anchor sampling and calibration weights. We also hypothesize that this multi-hop weighting is less sensitive to individual noisy similarities than directly weighting in continuous embedding-space. Our boundary analysis motivates the use of graph-space localization, and we evaluate the weighting strategy empirically.

3.2 Densification to Address Local Bias

We densify the graph before applying the kernel through three stages: (i) PCA fitted on pooled unlabeled calibration and test embeddings to concentrate informative signals and reduce feature correlations, (ii) an anisotropic Gaussian similarity kernel with adaptive bandwidths, and (iii) informative edge additions using a fixed similarity threshold. These steps expand the effective neighborhood of the structural kernel while limiting noise and over-densification.

For densification, we exploit the embedding similarities in the encoder embedding space using a Gaussian kernel. This kernel has two advantages: (i) unlike other similarity measures (e.g., cosine), it captures the non-linearity of the embeddings; and ii) the bandwidth controls the range of effect to adjust for local or global densification. To densify the graph, we construct a new adjacency matrix $\tilde{\mathbf{A}}$:

$$[\tilde{\mathbf{A}}]_{ij} = \begin{cases} 1 & \text{if } [\mathbf{A}]_{ij} = 1 \\ k_h(\mathbf{x}_i, \mathbf{x}_j) & \text{if } i \neq j,\ [\mathbf{A}]_{ij} = 0 \text{ and } k_h(\mathbf{x}_i, \mathbf{x}_j) \geq \tau_d \\ 0 & \text{o.w.} \end{cases} \tag{8}$$

where $k_h(\mathbf{x}_i, \mathbf{x}_j)$ is the anisotropic Gaussian similarity on PCA-projected embeddings defined below, and τ_d is a fixed similarity threshold set before calibration and evaluation. For nodes with zero weighted degree, we add a self-loop before computing the PPR transition matrix. With $\tilde{\mathbf{D}}_{ii} = \sum_{j=1}^{N} [\tilde{\mathbf{A}}]_{ij}$ recomputed after this step, the transition matrix is $\mathbf{M} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}$, which is row-stochastic.

The densification threshold τ_d controls the number of added edges. Small thresholds may produce nearly complete graphs, increasing computational cost and introducing noisy edges when message-passing embeddings are insufficiently discriminative. Conversely, large thresholds may miss informative connections needed to overcome graph sparsity. We fix τ_d, and use the resulting adjacency for both anchor sampling and calibration weighting. PCA is fitted on pooled unlabeled calibration and test embeddings, preserving calibration/test role symmetry. Find more details on the practical design principles of PCA in Appendix A.7.

Dimensionality Reduction. Two pivotal issues with GNN embeddings are as follows: (i) dimensional information redundancy [40], i.e., feature dimensions are highly correlated, and (ii) the lack of discriminative power due to over-smoothing [37]. These accumulate noise and errors in similarity measures derived from the raw, high-dimensional, ill-posed space.

We use PCA [41] to decouple feature correlations and condense the embedding space. Let $V_{\text{pool}} = V_{\text{cal}} \cup V_{\text{test}}$ and $N_{\text{pool}} = |V_{\text{pool}}|$. Using their unlabeled embeddings, let $\bar{\mathbf{x}}_{\text{pool}} = N_{\text{pool}}^{-1} \sum_{i \in V_{\text{pool}}} \mathbf{x}_i$, and let $\mathbf{Z}_{\text{pool}} \in \mathbb{R}^{N_{\text{pool}} \times p}$ contain the centered rows. Set $\mathbf{C}_Z = N_{\text{pool}}^{-1} \mathbf{Z}_{\text{pool}}^T \mathbf{Z}_{\text{pool}}$. We choose c leading orthonormal eigenvectors $\mathbf{V}_c \in \mathbb{R}^{p \times c}$ of $\mathbf{C}_Z$, with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_c > 0$, and write $\mathbf{\Lambda}_c = \text{diag}(\lambda_1, \ldots, \lambda_c)$. The projected embedding of any node i is $\mathbf{h}_i = \mathbf{V}_c^T(\mathbf{x}_i - \bar{\mathbf{x}}_{\text{pool}})$. These embeddings are used to compute similarities for densification. Retaining high-variance directions reduces redundancy but cannot recover distinctions lost to complete over-smoothing.

Selecting Informative Edges by Adaptive Bandwidth. Even in the reduced space, samples are not uniformly distributed: topological bottlenecks often make embeddings dense in some directions and sparse in others. Proximity in such non-uniform distributions cannot be effectively captured by an isotropic Gaussian kernel, which assumes equal variance across all dimensions. In contrast, an anisotropic kernel accounts for varying scales across dimensions and captures geometric correlations. Therefore, we construct an anisotropic Gaussian kernel with adaptive bandwidth.

The PCA projection offers a specific advantage: the variance of the projected data along each principal component equals its corresponding eigenvalue. This motivates us to adaptively set the bandwidths of the anisotropic Gaussian kernel based on the PCA eigenvalues. Formally, our Gaussian kernel in the PCA projection space is defined as:

$$k_h(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2} D_M^2(\mathbf{h}_i, \mathbf{h}_j; h^2\mathbf{\Lambda}_c)\right) \tag{9}$$

where $D_M^2(a, b; h^2\mathbf{\Lambda}_c) = (a-b)^T(h^2\mathbf{\Lambda}_c)^{-1}(a-b)$ is the squared Mahalanobis distance on the projected space and $h > 0$ is the base bandwidth. The retained eigenvalues are positive, ensuring a well-defined inverse. Without PCA, Gaussian similarities use the original embeddings.

Weighted Quantile Computation and Prediction. After graph densification, anchor point sampling, we apply the structure-aware weighting as in Eq. 7 and use the weights for computing the weighted quantile $\hat{q}_{1-\alpha}(\mathbf{x}_{n+1}, \tilde{\mathbf{x}}_{n+1})$ following Eq.1. Finally we derive the prediction set or interval as $\widehat{C}_n^{\text{GRAPHLCP}}(\mathbf{x}_{n+1}, \tilde{\mathbf{x}}_{n+1}) = \{y \in \mathcal{Y} : s(\mathbf{x}_{n+1}, y) \leq \hat{q}_{1-\alpha}(\mathbf{x}_{n+1}, \tilde{\mathbf{x}}_{n+1})\}$.

3.3 Graph-space boundary control

The preceding embedding-space result motivates studying the boundary term in RLCP's conditional-coverage bound using graph topology instead of GNN embedding distances. For vertices $u, v \in V$, let $d_G(u, v)$ denote their shortest-path distance in G, and let $d_G(u, S) := \min_{v \in S} d_G(u, v)$ for nonempty $S \subseteq V$. Following Theorem 1, we define $\text{bd}_G(B; r) := \text{bd}_{d_G}(B; r)$ for integer $r \geq 1$. Let $G[B] = (B, E[B])$ be the subgraph induced by B, where $E[B] = \{\{u, v\} \in E : u, v \in B\}$.

Distances between disconnected vertices in $G[B]$ are taken as $+\infty$. For $t \geq 0$, define

$$\Gamma_B(t) := \max_{\varnothing \neq S \subseteq B} \frac{|\{u \in B : d_{G[B]}(u, S) \leq t\}|}{|S|},$$

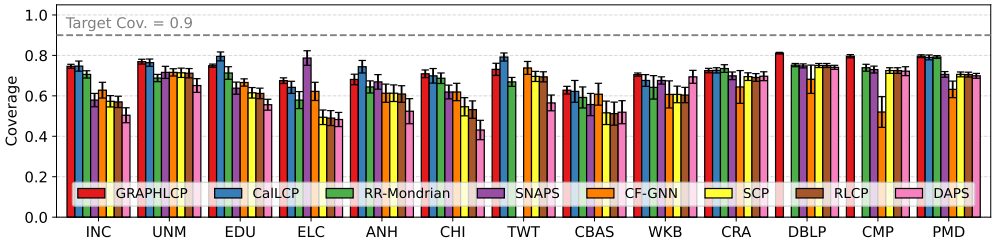


Figure 2: Results for WSC on regression and classification datasets. Miscoverage rate $\alpha = 0.1$. Overall, GRAPHLCP improves WSC over SCP and RLCP across all datasets.

which measures the maximum t-step neighborhood growth within $G[B]$. Let $E(B, B^c) = \big\{\{u, v\} \in E : u \in B,\ v \in B^c\big\}$ denote the set of edges crossing between B and B^c.

Theorem 3 (Graph-space boundary control) *For every integer $r \geq 1$,*

$$\frac{|\operatorname{bd}_G(B; r)|}{|B|} \leq \min\left\{1,\ \frac{\Gamma_B(r-1)|E(B, B^c)|}{|B|}\right\}. \tag{10}$$

Unlike Theorem 2, which shows boundary saturation under over-smoothing, Theorem 3 upper-bounds the boundary penalty in RLCP's conditional-coverage analysis. Under uniform node sampling, a small crossing cut and moderate neighborhood growth keep this penalty small. The localization-tail penalty is also controlled by PPR: since an anchor $\tilde{u} \mid u \sim H_K(u, \cdot)$ is sampled through a random walk of length $L_K = \min(L, K)$ with $L \sim \operatorname{Geom}(\beta)$, $\mathbb{P}\{d_{\tilde{G}}(u, \tilde{u}) > t \mid u\} \leq (1-\beta)^{t+1}$ for any integer $t \geq 0$. Together, these results bound both penalties in Eq. 2, motivating structural localization. But, they do not independently establish conditional-coverage guarantees in the transductive setting.

For GRAPHLCP, both results are applied to the unweighted graph of the densified adjacency $\tilde{\mathbf{A}}$ in Eq. 8, excluding self-loops. Proofs and an explicit maximum-degree bound are in Appendix A.1.2. Algorithm 1 in Appendix A.5 presents the pseudocode of our proposed method GRAPHLCP.

4 Experiments

Datasets. We show performance across *seven node regression* datasets, and *six node classification* datasets, and provide their details and statistics in the Appendix A.11.
Metrics. We show marginal coverage as a conventional metric. However, as discussed in Sec. 3, our localization-based approach is more advantageous for efficient relaxed conditional coverage. Therefore, for conditional CP, we use two settings from the recent comprehensive work on conditional CP diagnostics [5]. 1) Worst-case diagnostics: we evaluate worst-slab coverage (WSC) [36, 6]. 2) Group-based diagnostics: we show *four* feature-based groupings (partitioning).
Baselines. For validity, we compare with various baselines: vanilla split CP (SCP), two localized CP methods, namely RLCP [18] and CalLCP [16], and five graph-specific methods, specifically NAPS [8], DAPS [46], CF-GNN[20], and SNAPS [42], and RR-Mondrian [48]. For a fair comparison, we use the standard nonconformity scoring mechanism of vanilla split CP across methods, replacing the APS-based mechanism [36] in NAPS, DAPS, SNAPS, and CF-GNN. Alternative approaches such as APS for classification and CQR [35] for regression can independently affect coverage and prediction size, making it difficult to isolate the contributions of the underlying calibration mechanisms. CalLCP's high computational cost on large graphs exceeded our 3-day time limit, so it is occasionally excluded from our results. We fix the miscoverage rate to $\alpha = 0.1$ across all experiments. Additional details on other settings are provided in Appendix A.7. Code is available at https://github.com/pbaghershahi/GraphLCP.git.

4.1 Results on Conditional and Marginal CP

4.1.1 Conditional Coverage

We evaluate the conditional coverage under worst-case and group-based settings. Note that the worst-case setting using *WSC is the most challenging evaluation*, as it is an optimization-based

evaluation that adversarially seeks a bounded interval along a projection direction where a sufficient number of test points exhibit the minimum coverage. Therefore, searching across a large number of directions could reveal the method's lowest conditional coverage.

Worst-case Setting. We take 100 random direction vectors sampled uniformly from the unit sphere. The results are presented in Figure 2 for both classification and regression datasets. GRAPHLCP improves WSC over SCP and RLCP across all datasets and remains competitive with the strong baseline CalLCP, outperforming it on six of the eleven datasets with available results (Table 4). It also performs favorably against other graph-specific methods. NAPS attains high WSC on classification datasets but often with much larger sets and marginal overcoverage. These results highlight the benefit of structure-aware localization and feature-aware densification.

Next, we compare WSC and prediction length across datasets in Figure 3, along with each method's centroid of performance. Prediction lengths (sizes) are normalized by the maximum length across methods (denoted by NPL). GRAPHLCP achieves the highest average WSC, with slightly larger normalized prediction lengths than CalLCP. This demonstrates the improvement in local coverage despite a modest efficiency cost. An additional analysis in Appendix A.8 shows that increasing structural localization consistently improves WSC with a gradual increase in prediction size.

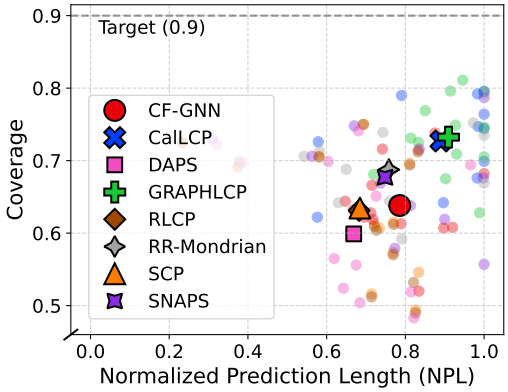


Figure 3: Results for WSC and efficiency on all datasets. GRAPHLCP achieves the highest average WSC among the methods shown, with slightly larger NPL than CalLCP.

Group-based Setting. Next, we evaluate group-based conditional coverage by partitioning test samples based on message-passing embeddings (*Feature Clustering*) and structural properties (*Clustering Coefficient*, *Homophily*, and *Node Degree*) [5]. For each property, we divide samples into three equal quantiles: "Low", "Mid", and "High". Figure 4 reports the worst coverage across partitions on four selected datasets. The results consistently favor GRAPHLCP, aligning with the WSC findings.

The effect of GRAPHLCP's two major components is evident in the group-based conditional coverage results, where we consistently outperform. By leveraging structural information, motivated by the importance of homophily in graph-based CP (Appendix A.4), GRAPHLCP improves coverage on homophily-based partitions while also outperforming baselines on feature-based partitions.

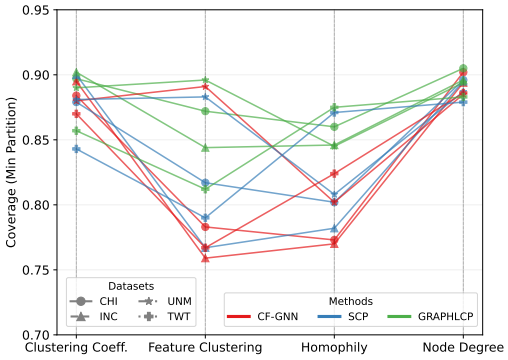


Figure 4: Group-based conditional coverage results ($\alpha = 0.1$), showing minimum coverage across semantic and structural partitions. GRAPHLCP outperforms baselines, particularly on homophily and feature-clustering partitions.

4.1.2 Marginal Coverage

Marginal coverage serves as a sanity check for CP methods. Table 1 shows that most methods meet or approach the target, although DAPS undercovers on several regression datasets. Since marginal coverage can mask local undercoverage, WSC remains important for assessing conditional performance. Marginal prediction lengths are reported in Appendix A.9.

4.2 Ablation Studies

Our method achieves robust conditional coverage by replacing embedding-space proximity with a structure-aware PPR kernel. To isolate the effect of structural localization from embedding-space reconditioning during densification, we replace both PPR-based anchor sampling and calibration

Table 1: Marginal coverage results. The target max miscoverage of $\alpha = 0.1$ is almost met by the baselines (except for DAPS). "-" means infinite-length prediction interval. OOT (out-of-time) takes an unreasonably long time.

Method	ANH	CHI	EDU	ELC	INC	UNM	TWT	CMP	CRA	DBLP	CBAS	WKB	PMD
SCP	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.91^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.00}$	$0.91^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.01}$	$0.90^{\pm.02}$	$0.90^{\pm.00}$
NAPS	-	-	-	-	-	-	-	$0.93^{\pm.00}$	$0.99^{\pm.00}$	$0.99^{\pm.00}$	$1.00^{\pm.00}$	$1.00^{\pm.00}$	$0.99^{\pm.00}$
DAPS	$0.87^{\pm.02}$	$0.86^{\pm.01}$	$0.88^{\pm.01}$	$0.90^{\pm.01}$	$0.87^{\pm.01}$	$0.87^{\pm.01}$	$0.82^{\pm.01}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.00}$
RLCP	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.00}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.01}$	$0.90^{\pm.02}$	$0.90^{\pm.00}$
CF-GNN	$0.92^{\pm.01}$	$0.91^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.91^{\pm.01}$	$0.90^{\pm.00}$	$0.91^{\pm.01}$	$0.90^{\pm.00}$	$0.92^{\pm.02}$	$0.90^{\pm.02}$	$0.90^{\pm.00}$
SNAPS	$0.92^{\pm.01}$	$0.93^{\pm.01}$	$0.92^{\pm.01}$	$0.97^{\pm.01}$	$0.91^{\pm.01}$	$0.91^{\pm.01}$	-	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.00}$
RR-Mondrian	$0.92^{\pm.01}$	$0.91^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.91^{\pm.01}$	$0.90^{\pm.00}$	$0.91^{\pm.01}$	$0.89^{\pm.00}$	$0.92^{\pm.01}$	$0.92^{\pm.02}$	$0.90^{\pm.00}$
CalLCP	$0.94^{\pm.01}$	$0.94^{\pm.01}$	$0.94^{\pm.00}$	$0.94^{\pm.01}$	$0.94^{\pm.00}$	$0.94^{\pm.00}$	$0.94^{\pm.00}$	OOT	$0.91^{\pm.00}$	OOT	$0.95^{\pm.01}$	$0.93^{\pm.01}$	$0.92^{\pm.00}$
GRAPHLCP	$0.90^{\pm.01}$	$0.92^{\pm.01}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.01}$	$0.90^{\pm.01}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$	$0.90^{\pm.01}$	$0.90^{\pm.00}$	$0.90^{\pm.00}$

Table 2: Isolated kernel type effect. A PPR kernel provides better coverage but a larger prediction interval/set than a Gaussian (GSS) kernel.

Measure	Variant	ANH	CHI	EDU	ELC	INC	UNM	TWT	CRA	CBAS	WKB
Coverage	GSS	$0.63^{\pm.04}$	$0.62^{\pm.02}$	$0.70^{\pm.01}$	$0.59^{\pm.02}$	$0.69^{\pm.02}$	$0.72^{\pm.02}$	$0.71^{\pm.02}$	$0.69^{\pm.02}$	$0.52^{\pm.06}$	$0.66^{\pm.02}$
	PPR	$0.68^{\pm.03}$	$0.71^{\pm.02}$	$0.75^{\pm.01}$	$0.68^{\pm.01}$	$0.75^{\pm.01}$	$0.77^{\pm.01}$	$0.73^{\pm.03}$	$0.72^{\pm.01}$	$0.63^{\pm.02}$	$0.70^{\pm.01}$
Prediction Length	GSS	$1.60^{\pm.09}$	$2.00^{\pm.09}$	$2.21^{\pm.15}$	$1.06^{\pm.04}$	$2.14^{\pm.11}$	$1.79^{\pm.06}$	$2.02^{\pm.11}$	$0.90^{\pm.05}$	$2.04^{\pm.10}$	$3.60^{\pm.14}$
	PPR	$1.66^{\pm.12}$	$2.17^{\pm.14}$	$2.44^{\pm.24}$	$1.11^{\pm.06}$	$2.52^{\pm.21}$	$1.85^{\pm.05}$	$2.10^{\pm.17}$	$2.00^{\pm.04}$	$2.75^{\pm.09}$	$4.21^{\pm.06}$

weighting with Gaussian localization (GSS). We sample a continuous anchor from a Gaussian centered at the test-node representation and weight calibration and test nodes by their similarities to this anchor. We preserve the same densification and preprocessing settings, including PCA and eigenvalue-based bandwidths when enabled. Table 2 shows that PPR consistently improves WSC at the cost of larger prediction sets, highlighting the benefit of multi-hop structural localization over direct embedding-space similarities.

Table 3: Ablation study on the main components across regression and classification with $\alpha = 0.1$. Results show that the most significant component is densification particularly for regression.

Variant	ANH		CHI		TWT		EDU		ELC		INC		UNM		CBAS		WKB	
	Cov	Size	Cov	Size	Cov	Size	Cov	Size	Cov	Size	Cov	Size	Cov	Size	Cov	Size	Cov	Size
GRAPHLCP	$0.68^{\pm.03}$	$1.66^{\pm.12}$	$0.71^{\pm.02}$	$2.17^{\pm.14}$	$0.73^{\pm.03}$	$2.10^{\pm.17}$	$0.75^{\pm.01}$	$2.44^{\pm.24}$	$0.68^{\pm.01}$	$1.11^{\pm.06}$	$0.75^{\pm.01}$	$2.52^{\pm.21}$	$0.77^{\pm.01}$	$1.85^{\pm.05}$	$0.63^{\pm.02}$	$2.75^{\pm.09}$	$0.70^{\pm.01}$	$4.21^{\pm.06}$
w/o Densification	$0.46^{\pm.01}$	$1.14^{\pm.09}$	$0.52^{\pm.02}$	$1.80^{\pm.08}$	$0.63^{\pm.03}$	$1.96^{\pm.11}$	$0.72^{\pm.01}$	$2.61^{\pm.38}$	$0.70^{\pm.02}$	$1.32^{\pm.17}$	$0.70^{\pm.01}$	$2.67^{\pm.54}$	$0.74^{\pm.01}$	$2.00^{\pm.20}$	$0.62^{\pm.02}$	$2.88^{\pm.07}$	$0.69^{\pm.02}$	$4.14^{\pm.04}$
w/o PCA	$0.67^{\pm.02}$	$1.64^{\pm.13}$	$0.62^{\pm.02}$	$1.94^{\pm.04}$	$0.73^{\pm.02}$	$2.19^{\pm.13}$	$0.74^{\pm.01}$	$2.44^{\pm.19}$	$0.65^{\pm.01}$	$1.05^{\pm.04}$	$0.72^{\pm.02}$	$2.35^{\pm.14}$	$0.76^{\pm.01}$	$1.80^{\pm.05}$	$0.63^{\pm.02}$	$2.75^{\pm.09}$	$0.67^{\pm.01}$	$4.11^{\pm.05}$

Add-on Effects. While GRAPHLCP relies heavily on the structure due to the PPR-based kernel, the densification step, which incorporates useful information from the embedding space, plays an important role. Table 3 shows that densification often has the largest effect, particularly on regression datasets (ANH, CHI, and TWT), although the gains are not uniform. PCA offers a more modest benefit by reducing redundant embedding dimensions. In practice, we apply PCA when the low-dimensional space is more useful for densification (see Appendix A.7).

5 Conclusion

We have introduced GRAPHLCP, a structure-aware localized conformal prediction framework for GNNs that explicitly incorporates graph topology into conformal localization and weighting. By replacing embedding-only proximity with a PPR-based kernel, our method achieves reliable marginal coverage while improving worst-slab coverage. Experiments on regression and classification datasets demonstrate the benefits of structure-aware localization, with a prediction-length trade-off. We believe this work highlights the importance of structure-aware localization for UQ on graphs. An important future direction is to extend localized conformal prediction to dynamic graphs.

Limitations. PPR-based localization can be less effective on sparse graphs. While densification mitigates this issue, poor embeddings may introduce noisy edges or miss informative connections depending on the similarity threshold. Longer PPR walks may weaken localization, while small calibration sets can lead to unstable weights. Moreover, improved conditional coverage may require larger prediction sets, and densification and PPR computations introduce additional runtime costs, particularly on large graphs.

References

[1] Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

[2] Álvaro Arroyo, Alessio Gravina, Benjamin Gutteridge, Federico Barbero, Claudio Gallicchio, Xiaowen Dong, Michael Bronstein, and Pierre Vandergheynst. On vanishing gradients, over-smoothing, and over-squashing in gnns: Bridging recurrent and graph learning. *Advances in Neural Information Processing Systems*, 38:74356–74393, 2026.

[3] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 2022.

[4] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, 2018.

[5] Sacha Braun, David Holzmüller, Michael I. Jordan, and Francis Bach. Conditional coverage diagnostics for conformal prediction, 2025.

[6] Maxime Cauchois, Suyash Gupta, and John C. Duchi. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *J. Mach. Learn. Res.*, 22(1), January 2021.

[7] Dawei Cheng, Yao Zou, Sheng Xiang, and Changjun Jiang. Graph neural networks for financial fraud detection: a review. *Frontiers of Computer Science*, 19(9):199609, 2025.

[8] Jase Clarkson. Distribution free prediction sets for node classification. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[9] Chung Fan. Discrete isoperimetric inequalities. *Surveys in differential geometry*, 9:53–82, 1996.

[10] Evan N Feinberg, Debnil Sur, Zhenqin Wu, Brooke E Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S Pande. Potentialnet for molecular property prediction. *ACS central science*, 4(11):1520, 2018.

[11] Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 08 2020.

[12] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*, 2019.

[13] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1263–1272. JMLR.org, 2017.

[14] Francesco Di Giovanni, T. Konstantin Rusch, Michael Bronstein, Andreea Deac, Marc Lackenby, Siddhartha Mishra, and Petar Veličković. How does over-squashing affect the power of GNNs? *Transactions on Machine Learning Research*, 2024.

[15] Jhony H. Giraldo, Konstantinos Skianis, Thierry Bouwmans, and Fragkiskos D. Malliaros. On the trade-off between over-smoothing and over-squashing in deep graph neural networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 566–576, New York, NY, USA, 2023. Association for Computing Machinery.

[16] Leying Guan. Localized conformal prediction: a generalized inference framework for conformal prediction. *Biometrika*, 110(1):33–50, 07 2022.

[17] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. GOOD: A graph out-of-distribution benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[18] Rohan Hore and Rina Foygel Barber. Conformal prediction with local weights: randomization enables robust guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(2):549–578, 11 2024.

[19] Hans Hao-Hsun Hsu, Yuesong Shen, Christian Tomani, and Daniel Cremers. What makes graph neural networks miscalibrated? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 13775–13786. Curran Associates, Inc., 2022.

[20] Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. *NeurIPS*, 2023.

[21] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, page 271–279, New York, NY, USA, 2003. Association for Computing Machinery.

[22] Junteng Jia and Austion R. Benson. Residual correlation in graph neural network regression. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 588–598, New York, NY, USA, 2020. Association for Computing Machinery.

[23] Sunay Joshi, Shayan Kiyani, George J. Pappas, Edgar Dobriban, and Hamed Hassani. Conformal inference under high-dimensional covariate shifts via likelihood-ratio regularization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

[25] Arun Kumar Kuchibhotla. Exchangeability, conformal prediction, and rank tests, 2021.

[26] Siddhartha Laghuvarapu, Zhen Lin, and Jimeng Sun. Codrug: conformai drug property prediction with density estimation under covariate shift. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[27] Xixun Lin, Yanan Cao, Nan Sun, Lixin Zou, Chuan Zhou, Peng Zhang, Shuai Zhang, Ge Zhang, and Jia Wu. Conformal graph-level out-of-distribution detection with adaptive data augmentation. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 4755–4765, New York, NY, USA, 2025. Association for Computing Machinery.

[28] Robert Lunde. On the validity of conformal prediction for network data under non-uniform sampling, 2023.

[29] Robert Lunde, Elizaveta Levina, and Ji Zhu. Conformal prediction for network-assisted regression. *Journal of the American Statistical Association*, 120(551):1633–1644, 2025.

[30] Minbo Ma, Peng Xie, Fei Teng, Bin Wang, Shenggong Ji, Junbo Zhang, and Tianrui Li. Histgnn: Hierarchical spatio-temporal graph neural network for weather forecasting. *Information Sciences*, 648:119580, 2023.

[31] Ariane Marandon. Conformal link prediction for false discovery rate control. *TEST*, 33:1062 – 1083, 2023.

[32] Huda Nassar, Kyle Kloster, and David F. Gleich. Strong localization in personalized pagerank vectors. In *Proceedings of the 12th International Workshop on Algorithms and Models for the Web Graph - Volume 9479*, WAW 2015, page 190–202, Berlin, Heidelberg, 2015. Springer-Verlag.

[33] Athanasios N. Nikolakopoulos, Xia Ning, Christian Desrosiers, and George Karypis. Trust your neighbors: A comprehensive survey of neighborhood-based methods for recommender systems. *ArXiv*, abs/2109.04584, 2021.

[34] Oyebade K. Oyedotun, Kassem Al Ismaeil, and Djamila Aouada. Why is everyone training very deep neural network with skip connections? *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):5961–5975, 2023.

[35] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[36] Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[37] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks, 2023.

[38] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[39] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation, 2019.

[40] Wei Shen, Mang Ye, and Wenke Huang. Resisting over-smoothing in graph neural networks via dual-dimensional decoupling. In *ACM Multimedia 2024*, 2024.

[41] Jonathon Shlens. A tutorial on principal component analysis, 2014.

[42] Jianqing Song, Jianguo Huang, Wenyu Jiang, Baoming Zhang, Shuangjie Li, and Chongjun Wang. Similarity-navigated conformal prediction for graph neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[43] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[44] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005. Springer, New York.

[45] Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[46] Soroush H Zargarbashi, Simone Antonelli, and Aleksandar Bojchevski. Conformal prediction sets for graph neural networks. In *International Conference on Machine Learning*, pages 12292–12318. PMLR, 2023.

[47] Soroush H. Zargarbashi and Aleksandar Bojchevski. Conformal inductive graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[48] Zheng Zhang, Jie Bao, Zhixin Zhou, Nicolo Colombo, Lixin Cheng, and Rui Luo. Residual reweighted conformal prediction for graph neural networks. In *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, UAI '25. JMLR.org, 2025.

[49] Tianyi Zhao, Jian Kang, and Lu Cheng. Conformalized link prediction on graph neural networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 4490–4499, New York, NY, USA, 2024. Association for Computing Machinery.

A Appendix

A.1 Proofs and further boundary results

The graph throughout these proofs is finite, connected, simple, undirected, and unweighted, with $\varnothing \neq B \subsetneq V$. All distances in a disconnected induced subgraph are taken to be $+\infty$.

A.1.1 Embedding-space boundary result

Lemma 4 (Boundary reduction to cross-set distances) *Let U and W be independent and uniformly distributed on B and B^c, respectively, and define*

$$Q_B(Z) := \mathbb{E}\big[\|z_U - z_W\|_2^2\big] = \frac{1}{|B||B^c|} \sum_{u\in B} \sum_{v\in B^c} \|z_u - z_v\|_2^2.$$

For every $\delta > 0$,

$$\frac{|\operatorname{bd}_Z(B; 2\delta)|}{|B|} \geq \left[1 - \frac{Q_B(Z)}{4\delta^2}\right]_+. \tag{11}$$

Proof.

Fix $u \in B$. By definition, $u \in \operatorname{bd}_Z(B; 2\delta)$ if and only if there is at least one $v \in B^c$ such that $\|z_u - z_v\|_2 \leq 2\delta$. Hence,

$$\mathbf{1}\{u \in \operatorname{bd}_Z(B; 2\delta)\} \geq \frac{1}{|B^c|} \sum_{v\in B^c} \mathbf{1}\{\|z_u - z_v\|_2 \leq 2\delta\}.$$

Indeed, if no such v exists, both sides are zero. Otherwise, the left-hand side is one and the right-hand side is a fraction between zero and one. Therefore we have:

$$\frac{1}{|B|} \sum_{u\in B} \mathbf{1}\{u \in \operatorname{bd}_Z(B; 2\delta)\} \geq \frac{1}{|B||B^c|} \sum_{u\in B} \sum_{v\in B^c} \mathbf{1}\{\|z_u - z_v\|_2 \leq 2\delta\}.$$

The left-hand side counts the vertices in the boundary and equals $|\operatorname{bd}_Z(B; 2\delta)|/|B|$. Because U and W are independent and uniform on B and B^c, respectively, each pair $(u, v) \in B \times B^c$ has probability $1/(|B||B^c|)$. Thus the right-hand side is $\mathbb{P}(\|z_U - z_W\|_2 \leq 2\delta)$, and

$$\frac{|\operatorname{bd}_Z(B; 2\delta)|}{|B|} \geq \mathbb{P}(\|z_U - z_W\|_2 \leq 2\delta) = 1 - \mathbb{P}(\|z_U - z_W\|_2 > 2\delta).$$

By Markov's inequality,

$$\mathbb{P}(\|z_U - z_W\|_2 > 2\delta) \leq \frac{\mathbb{E}[\|z_U - z_W\|_2^2]}{4\delta^2} = \frac{Q_B(Z)}{4\delta^2}.$$

The boundary fraction is also nonnegative. Combining these facts proves (11).

□

Lemma 5 (Vector-valued Poincaré inequality) *Let $Z = (z_u)_{u\in V} \in \mathbb{R}^{N\times p}$ and $\bar{z} = N^{-1}\sum_{u\in V} z_u$. Then*

$$\sigma_1 \sum_{u\in V} \|z_u - \bar{z}\|_2^2 \leq \mathcal{E}_G(Z). \tag{12}$$

Proof. The variational characterization of the smallest nonzero eigenvalue of the combinatorial Laplacian $L_G = D - A$ gives

$$\sigma_1 = \min_{\substack{f:V\to\mathbb{R},\ f\neq 0 \\ \sum_{u\in V} f(u)=0}} \frac{\sum_{\{u,v\}\in E} (f(u) - f(v))^2}{\sum_{u\in V} f(u)^2} \tag{13}$$

This is the Rayleigh–Ritz formula for L_G given by Chung [9]. For each coordinate j, set $f_j(u) = z_{u,j} - \bar{z}_j$. Then $\sum_{u\in V} f_j(u) = 0$, so (13) implies

$$\sigma_1 \sum_{u\in V} (z_{u,j} - \bar{z}_j)^2 \leq \sum_{\{u,v\}\in E} (z_{u,j} - z_{v,j})^2.$$

The inequality also holds if f_j is identically zero. Summing over $j = 1, \ldots, p$ proves the claim. □

Lemma 6 (Spectral control of cross-set dispersion) *For a connected graph G and $\rho_B = |B|/N$,*

$$Q_B(Z) \le \frac{N}{|B||B^c|\sigma_1}\mathcal{E}_G(Z) = \frac{\mathcal{E}_G(Z)}{\rho_B(1-\rho_B)N\sigma_1}. \tag{14}$$

Proof.

Let $\bar{z} = N^{-1}\sum_{u\in V} z_u$. Expanding squared distances gives

$$\begin{aligned}\sum_{\{u,v\}\subseteq V}\|z_u - z_v\|_2^2 &= \frac{1}{2}\sum_{u,v\in V}\left(\|z_u\|_2^2 + \|z_v\|_2^2 - 2z_u^\top z_v\right)\\ &= N\sum_{u\in V}\|z_u\|_2^2 - \left\|\sum_{u\in V} z_u\right\|_2^2\\ &= N\sum_{u\in V}\|z_u - \bar{z}\|_2^2\\ &\le \frac{N}{\sigma_1}\mathcal{E}_G(Z),\end{aligned}$$

where the last step uses Lemma 5. The pairs with $u \in B$ and $v \in B^c$ are a subset of all unordered vertex pairs. Therefore,

$$\sum_{u\in B}\sum_{v\in B^c}\|z_u - z_v\|_2^2 \le \frac{N}{\sigma_1}\mathcal{E}_G(Z).$$

Dividing both sides by $|B||B^c|$ proves (14). □

Proof of Theorem 2. We obtain (3) by substituting (14) into (11).

For the final statement, the proof of Lemma 6 gives

$$\sum_{\{u,v\}\subseteq V}\|z_u^{(K)} - z_v^{(K)}\|_2^2 \le \frac{N}{\sigma_1}\mathcal{E}_G(Z_K) \longrightarrow 0.$$

The sum contains finitely many nonnegative terms. Hence, for all sufficiently large K, every pair of vertices satisfies $\|z_u^{(K)} - z_v^{(K)}\|_2 < 2\delta$. Since $B^c \neq \varnothing$, every $u \in B$ then belongs to $\mathrm{bd}_{Z_K}(B; 2\delta)$. □

The contractivity analysis of [2] shows that contractive GNN layers lead to vanishing Dirichlet energy. Combining this result with Theorem 2, we obtain the following finite-depth bound on the embedding-space boundary.

Corollary 7 (Finite-depth bound for contractive layers) *Let $Z_0 = \mathbf{X} \in \mathbb{R}^{N\times p_0}$ be the input node-feature matrix, and let $Z_k = F_k(Z_{k-1}) \in \mathbb{R}^{N\times p_k}$ for $k = 1, \dots, K$, where F_k is the kth GNN layer. Suppose F_k is Lipschitz with respect to the Frobenius norm with constant ℓ_k and $F_k(0) = 0$. Set $\Lambda_K = \prod_{k=1}^K \ell_k^2$ and $\bar{d} = 2|E|/N$. Then, for every nonempty $B \subsetneq V$ and every $\delta > 0$,*

$$\frac{|\,\mathrm{bd}_{Z_K}(B; 2\delta)|}{|B|} \ge \left[1 - \frac{\bar{d}\,\Lambda_K\|Z_0\|_F^2}{4\delta^2\rho_B(1-\rho_B)\sigma_1}\right]_+. \tag{15}$$

If $\ell_k \le q$ for every k, where $0 \le q < 1$, then, on a fixed graph, $\mathrm{bd}_{Z_K}(B; 2\delta) = B$ for all sufficiently large K.

Proof. Since $F_k(0) = 0$, the Lipschitz assumption gives

$$\|Z_k\|_F = \|F_k(Z_{k-1}) - F_k(0)\|_F \le \ell_k\|Z_{k-1}\|_F.$$

Applying this inequality successively yields

$$\|Z_K\|_F^2 \le \left(\prod_{k=1}^K \ell_k^2\right)\|Z_0\|_F^2 = \Lambda_K\|Z_0\|_F^2. \tag{16}$$

For any edge $\{u, v\} \in E$,

$$\|z_u^{(K)} - z_v^{(K)}\|_2^2 \le 2\big(\|z_u^{(K)}\|_2^2 + \|z_v^{(K)}\|_2^2\big) \le 2\|Z_K\|_F^2.$$

The last inequality holds because u and v are distinct rows of Z_K. Summing over the $|E|$ edges and using (16), we obtain

$$\begin{aligned}\mathcal{E}_G(Z_K) &= \sum_{\{u,v\}\in E} \|z_u^{(K)} - z_v^{(K)}\|_2^2 \\ &\le 2|E|\|Z_K\|_F^2 \\ &\le 2|E|\Lambda_K\|Z_0\|_F^2 \\ &= N\bar{d}\,\Lambda_K\|Z_0\|_F^2.\end{aligned}$$

Substituting this bound into (3) proves (15).

Finally, if $\ell_k \le q < 1$ for all k, then $\Lambda_K \le q^{2K} \to 0$. Thus the energy bound above implies $\mathcal{E}_G(Z_K) \to 0$, and the eventual equality follows from Theorem 2. □

Remark. For intermediate depths with $\mathcal{E}_G(Z_K) > 0$, Eq. 3 equivalently gives

$$\frac{|B \setminus \mathrm{bd}_{Z_K}(B; 2\delta)|}{|B|} \le \min\left\{1,\ \frac{\mathcal{E}_G(Z_K)}{4\delta^2\rho_B(1-\rho_B)N\sigma_1}\right\}. \tag{17}$$

Thus, whenever the normalized energy term is below one, the bound limits the fraction of nodes in B that remain farther than 2δ from B^c. Lower energy yields a tighter bound, although it does not necessarily imply a monotonic increase in the actual boundary fraction across GNN depths.

A.1.2 Graph-space boundary result

Define the one-hop inner boundary of B by

$$\partial_{\mathrm{in}} B := \{u \in B : \exists v \in B^c \text{ s.t. } \{u, v\} \in E\}.$$

Lemma 8 (First-exit identity) *For every $u \in B$,*

$$d_G(u, B^c) = 1 + d_{G[B]}(u, \partial_{\mathrm{in}} B).$$

Consequently, for every integer $r \ge 1$,

$$\mathrm{bd}_G(B; r) = \{u \in B : d_{G[B]}(u, \partial_{\mathrm{in}} B) \le r - 1\}. \tag{18}$$

Proof. Let $z \in \partial_{\mathrm{in}} B$ be reachable from u by a shortest path in $G[B]$. Since $z \in \partial_{\mathrm{in}} B$ has a neighbor in B^c, a path from u to z in $G[B]$ extends to B^c, therefore

$$d_G(u, B^c) \le 1 + d_{G[B]}(u, \partial_{\mathrm{in}} B).$$

Conversely, consider a shortest path from u to B^c, of length $k = d_G(u, B^c)$. All vertices before its final vertex lie in B; otherwise the path would have reached B^c sooner. Its second-to-last vertex on the path belongs to $\partial_{\mathrm{in}} B$, and the preceding $k - 1$ edges form a path in $G[B]$. Hence

$$d_{G[B]}(u, \partial_{\mathrm{in}} B) \le k - 1,$$

which proves the reverse inequality. The graph is connected and B^c is nonempty, so these distances are finite. The representation follows by comparing both sides with r. □

Proof of Theorem 3. By the definition of $\Gamma_B(r-1)$, every nonempty $S \subseteq B$ satisfies

$$|\{u \in B : d_{G[B]}(u, S) \le r - 1\}| \le \Gamma_B(r-1)|S|.$$

Since G is connected and B is nonempty and proper, $\partial_{\mathrm{in}} B$ is nonempty. Taking $S = \partial_{\mathrm{in}} B$ and using (18) gives

$$\begin{aligned}|\,\mathrm{bd}_G(B; r)| &= |\{u \in B : d_{G[B]}(u, \partial_{\mathrm{in}} B) \le r - 1\}| \\ &\le \Gamma_B(r-1)|\partial_{\mathrm{in}} B|.\end{aligned}$$

Every vertex in $\partial_{\text{in}}B$ is incident to at least one edge in $E(B, B^c)$. Counting crossing edges by their endpoint in B therefore $|\partial_{\text{in}}B| \le |E(B, B^c)|$. Moreover, $\operatorname{bd}_G(B; r) \subseteq B$, so $|\operatorname{bd}_G(B; r)| \le |B|$. Together with the preceding bounds, this gives

$$\begin{aligned}|\operatorname{bd}_G(B; r)| &\le \Gamma_B(r-1)|\partial_{\text{in}}B| \\ &\le \Gamma_B(r-1)|E(B, B^c)|.\end{aligned}$$

And then

$$|\operatorname{bd}_G(B; r)| \le \min\left\{|B|,\, \Gamma_B(r-1)|E(B, B^c)|\right\}.$$

Dividing by $|B|$ yields

$$\frac{|\operatorname{bd}_G(B; r)|}{|B|} \le \min\left\{1,\, \frac{\Gamma_B(r-1)|E(B, B^c)|}{|B|}\right\}$$

which complete the proof. □

To obtain a bound without the growth function Γ_B, let $\deg_{\max}(B) := \max_{u\in B} \deg_{G[B]}(u)$ be the maximum internal degree, where $\deg_{G[B]}(u) := |\{v \in B : \{u, v\} \in E\}|$. For each integer $t \ge 0$, define

$$M_B(t) := \begin{cases} 1, & \deg_{\max}(B) = 0, \\ 1 + \deg_{\max}(B) \displaystyle\sum_{j=0}^{t-1} \big(\deg_{\max}(B) - 1\big)^j, & \deg_{\max}(B) \ge 1, \end{cases}$$

where the sum is empty when $t = 0$ and $0^0 = 1$.

Lemma 9 (Growth under bounded internal degree) *For every nonempty $S \subseteq B$ and integer $t \ge 0$,*

$$|\{u \in B : d_{G[B]}(u, S) \le t\}| \le M_B(t)|S|.$$

In particular, $\Gamma_B(t) \le M_B(t)$.

Proof. If $\deg_{\max}(B) = 0$, every vertex of $G[B]$ is isolated. The t-neighborhood of S is then S itself, and the result follows from $M_B(t) = 1$. Suppose $\deg_{\max}(B) \ge 1$. For $j \ge 0$, let

$$L_j := \{u \in B : d_{G[B]}(u, S) = j\}.$$

In particular, $L_0 = S$. Every vertex in L_1 has an edge to L_0, while each vertex of L_0 has at most $\deg_{\max}(B)$ neighbors. Hence $|L_1| \le \deg_{\max}(B)|L_0| = \deg_{\max}(B)|S|$. Now for $j \ge 1$ we derive two bounds based on the same count $|E(L_j, L_{j+1})|$, i.e. the number of edges crossing L_j to L_{j+1}.

First, every vertex in L_{j+1} has at least one neighbor in L_j, so $|L_{j+1}| \le |E(L_j, L_{j+1})|$. Next, for each vertex in L_j, deducting at least one neighbor which lies in L_{j-1} along a shortest path to S, it has at most $\deg_{\max}(B) - 1$ neighbors in L_{j+1} which gives $|E(L_j, L_{j+1})| \le (\deg_{\max}(B) - 1)|L_j|$. Combining these inequalities yields

$$|L_{j+1}| \le (\deg_{\max}(B) - 1)|L_j|.$$

Starting from the bound on $|L_1|$ and applying this recurrence $j - 1$ times, we obtain, for every $j \ge 1$,

$$|L_j| \le (\deg_{\max}(B) - 1)^{j-1}|L_1| \le \deg_{\max}(B)(\deg_{\max}(B) - 1)^{j-1}|S|.$$

The layers $L_0, \dots, L_t$ are disjoint, and their union is the set of vertices within distance t of S in $G[B]$. Therefore

$$\begin{aligned}|\{u \in B : d_{G[B]}(u, S) \le t\}| &= \sum_{j=0}^{t} |L_j| \\ &= |S| + \sum_{j=1}^{t} |L_j| \\ &\le |S| + \sum_{j=1}^{t} \deg_{\max}(B)\big(\deg_{\max}(B) - 1\big)^{j-1}|S| \\ &= |S| \left(1 + \deg_{\max}(B) \sum_{i=0}^{t-1} \big(\deg_{\max}(B) - 1\big)^i\right) \\ &= M_B(t)|S|.\end{aligned}$$

Finally, since the bound holds for every nonempty $S \subseteq B$,

$$\begin{aligned}\Gamma_B(t) &= \max_{\varnothing \neq S \subseteq B} \frac{|\{u \in B : d_{G[B]}(u, S) \leq t\}|}{|S|} \\ &\leq \max_{\varnothing \neq S \subseteq B} M_B(t) = M_B(t).\end{aligned}$$

□

Alternatively to connect explicitly with maximum-degree, we write $\mathrm{vol}_G(S) := \sum_{u \in S} \deg_G(u)$. The conductance of B in G is

$$\Phi_G(B) := \frac{|E(B, B^c)|}{\min\{\mathrm{vol}_G(B), \mathrm{vol}_G(B^c)\}}.$$

Corollary 10 (Explicit maximum-degree bound) *For every integer $r \geq 1$,*

$$\frac{|\,\mathrm{bd}_G(B; r)|}{|B|} \leq \min\left\{1,\ \frac{M_B(r-1)|E(B, B^c)|}{|B|}\right\} \leq \min\left\{1,\ M_B(r-1)\Phi_G(B)\overline{d}_G(B)\right\},$$

where $\overline{d}_G(B) = \mathrm{vol}_G(B)/|B|$.

Proof. Applying Lemma 9 in Theorem 3 proves the first inequality in the corollary. For the second inequality, the definition of $\Phi_G(B)$ gives

$$\begin{aligned}|E(B, B^c)| &= \Phi_G(B) \min\{\mathrm{vol}_G(B), \mathrm{vol}_G(B^c)\} \\ &\leq \Phi_G(B)\,\mathrm{vol}_G(B) \\ &= \Phi_G(B)|B|\overline{d}_G(B).\end{aligned}$$

Consequently,

$$\frac{M_B(r-1)|E(B, B^c)|}{|B|} \leq M_B(r-1)\Phi_G(B)\overline{d}_G(B).$$

Taking the minimum of each side with 1 proves the second inequality. □

A.2 Additional Related Work

Conformal Prediction (CP), Weighted CP (WCP), and Localized CP (LCP) Unlike single-point predictors that do not quantify uncertainty [38], CP provides a simple distribution-free framework for constructing prediction sets in classification and prediction intervals in regression with finite-sample guarantees [38, 1, 44]. CP's validity comes from the assumption of exchangeability, and as a distribution-free method, it does not rely on strong parametric assumptions or strict distributions. [25, 44]. The high variance in marginal coverage across different regions of the data distribution has increased interest in conditional CP although exact conditional coverage cannot generally be achieved in finite samples [11].

Many recent works study approximate or relaxed conditional coverage to improve coverage in heterogeneous settings by adapting prediction set sizes to the difficulty of individual instances [6, 36]. Also, several weighted CP methods have proposed relaxing standard exchangeability assumptions [43, 3]. These approaches typically estimate distribution ratios between calibration and test samples and use them to compute weighted quantiles [18, 16].

Similarly, localized CP (LCP) gives larger weights to calibration points that are more similar to the test instance to improve local coverage behavior [43, 18]. From this category, Randomized LCP (RLCP) further introduces randomization and anchor sampling strategies to improve coverage around the test point itself rather than only its surrounding neighborhood [18]. Still, most localized methods depend heavily on the quality of the feature representation used to measure similarity [23, 18]. In high-dimensional settings or when embeddings are poorly structured, defining reliable local neighborhoods becomes difficult and can lead to inefficient prediction intervals [3, 23].

Uncertainty Quantification in Graph Neural Networks (GNNs) Calibration remains a challenge in graph neural networks (GNNs) since 1) their learned representations are often not sufficiently separable or reliable [45, 19], and 2) predictions are not uniformly reliable across different regions of

the graph due to structural effects [8, 19]. This is particularly important in high-stakes applications such as drug discovery or fraud detection, where inaccurate confidence estimates may lead to unreliable decisions [10, 7, 19]. In addition, under heterophily, GNNs can suffer from over-squashing, while deeper architectures are also prone to over-smoothing [15, 37]. As a result, node embeddings may become noisy or less informative for measuring similarity, which limits the application of existing LCP methods to graph data as they rely heavily on stable embedding-space distances [15, 18].

Conformal Prediction on Graphs Applying CP to graph data is challenging because node dependencies violate the standard i.i.d. assumption [28, 8]. Still, prior work has shown that exchangeability can hold in transductive settings for conventional GNNs due to permutation invariance, which supports the use of finite-sample CP guarantees on graph data [25, 20, 28]. This has motivated CP for graph tasks such as link prediction and out-of-distribution detection [27, 31, 49].

Recent graph CP methods often incorporate additional structural or contextual information during calibration [8, 20, 48]. Some approaches use embedding-space density estimates for weighting [26], while others rely on structural partitioning [48], graph diffusion scores [46], or combinations of node features and graph topology to adapt nonconformity scores [42]. Many of these methods remain sensitive to embedding quality or rigid graph partitions. Our approach instead uses a structure-aware kernel to improve localization when feature-space similarity is unreliable.

A.3 Limitations of RLCP

While RLCP excels in low-dimensional settings, its application to GNNs is hindered by poor embedding discriminability. GNN embeddings often exhibit high feature correlation, non-uniformity, and over-smoothing. Consequently, applying a Gaussian kernel here tends to collapse the distribution toward a Dirac delta or a uniform weight. We verify this behavior by measuring the weights of the calibration and test samples across varying bandwidths.

Figure 5 shows the empirical cumulative distribution function (ECDF) for calibration (dashed) and test (solid) weights according to sampled anchor points for a Gaussian kernel applied in the embedding space for different bandwidths. Small bandwidth assigns almost all the weight to the test points (i.e., $w_{n+1} \simeq 1.0$), leaving the calibration nodes with identical weights close to zero, i.e., $w_i \simeq 0.0 : \forall i \in [n]$. This pushes the estimated quantile $\hat{q}_{1-\alpha}(\mathbf{x}_{n+1}, \tilde{\mathbf{x}}_{n+1}) \simeq +\infty$ to infinity according to Eq. 1. Whereas, larger bandwidths yield identical weights for all points due to indiscriminative, over-smoothed embeddings. This reduces RLCP to standard SCP, ignoring local proximity for better test conditional coverage. We address this by introducing GRAPHLCP, which explicitly incorporates graph-structural proximity.

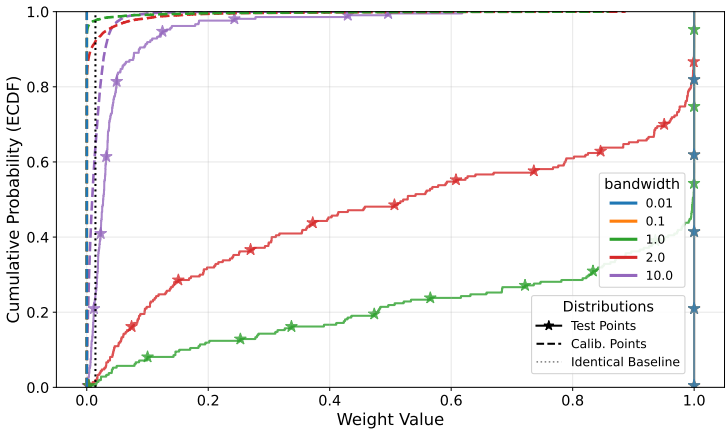


Figure 5: GOODCBAS dataset: Impact of Gaussian kernel bandwidth h on RLCP weight distribution. Low bandwidths concentrate probability mass on test points, leaving calibration points with near-zero weights. Conversely, large bandwidths cause weights to jump toward a uniform distribution where all points receive near-zero values.

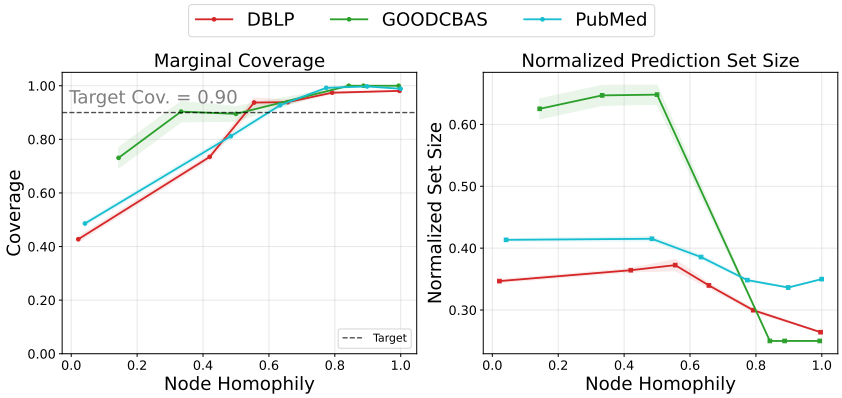


Figure 6: Impact of node homophily on the marginal coverage and prediction length/size of SCP. Satisfying coverage for more homophilous nodes is easier at the same or lower prediction length, underlining the significant contribution of graph structure in CP application for GNNs.

A.4 Empirical Insights and Design Principle

Our assumption is that weighted CP can highly benefit from nodes' structural dependencies. We confirm this assumption by an experiment that shows the effect of node homophily on the coverage and prediction length of vanilla SCP (defined in Sec. 2.2). For classification datasets, a node's homophily ratio $\eta_u = \frac{|\{v \in \mathcal{N}_u : y_v = y_u\}|}{|\mathcal{N}_u|}$ and for regression datasets, we have $\eta_u = 1 - \frac{1}{|\mathcal{N}_u|} \sum_{v \in \mathcal{N}_u} \frac{|y_v - y_u|}{\max_{i,j:A_{ij}=1} |y_i - y_j|}$, where y_u is target value (e.g. label) of u. Figure 6 elaborates on a common pattern: nodes with low homophily suffer from significant under-coverage, while high-homophily nodes are well covered, exhibiting an easier task. Also, the prediction lengths are generally stable or decrease with higher homophily ratios. This demonstrates that SCP fails to guarantee coverage for heterophilous nodes, even with a higher uncertainty budget.

Because uncertainty and graph topology are strongly correlated, we explicitly incorporate the underlying structure into our quantile weighting. GRAPHLCP replaces embedding-based kernels with a structure-aware kernel that governs both anchor sampling and weighting. By defining proximity through topology, we enable localized CP that respects structural dependencies and bypasses the suboptimality of message-passing embeddings [34]. GRAPHLCP comprises three components: (1) a Personalized PageRank-based localization kernel, (2) randomized anchor sampling and weighting compatible with RLCP, and (3) a feature-aware densification step to mitigate sparsity. These are detailed below.

A.5 Pseudocode for Our Method

Algorithm 1 presents the Pseudocode of the three main stages of our proposed method GRAPHLCP: 1) *graph densification* with a PCA projection and adaptive bandwidths, 2) *anchor point sampling* according to a geometric distribution of random walk lengths, and 3) *weighting and quantile computation* using our structure-aware kernel. The algorithm shows the PCA-enabled variant; otherwise, Gaussian similarities are computed directly on the original embeddings.

A.6 Fixed Densification Threshold.

The similarity threshold τ_d is fixed before calibration and evaluation and is not selected using calibration or test labels. When enabled, PCA is fitted using the pooled unlabeled calibration and test embeddings, and the same transformation is applied to all nodes. The threshold and transformation are therefore unchanged by swapping calibration/test roles within this pooled set.

A.7 Experimental Settings

For all datasets, we split each dataset into 4 sets: training, validation, calibration, and test. We train the models on the training set and use the validation set to tune the hyperparameters. Once the models

Algorithm 1 GRAPHLCP

Input: Graph $G = (V, E)$ with adjacency $\mathbf{A}$ and features $\mathbf{X}$, frozen GNN embeddings, Calibration $\mathcal{D}_{cal}$, test nodes V_{test} (including u_{n+1}), Params α, β, K, c, h, fixed τ_d
Output: Prediction set $\widehat{C}_n(\mathbf{x}_{n+1})$

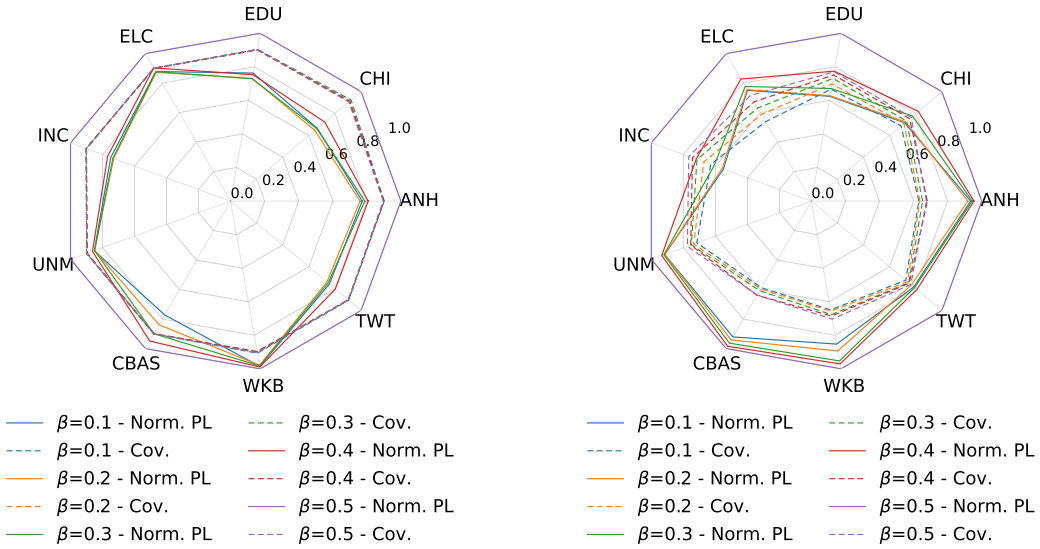
Stage 1: Feature-Aware Densification
$V_{\text{pool}} \leftarrow V_{\text{cal}} \cup V_{\text{test}}$
$\mathbf{V}_c, \mathbf{\Lambda}_c, \bar{\mathbf{x}}_{\text{pool}} \leftarrow \text{PCA}(\{\mathbf{x}_i : i \in V_{\text{pool}}\}, \text{top-}c)$
$\tilde{\mathbf{A}} \leftarrow \mathbf{A}$
for each pair (i, j) with $i \neq j$ and $\mathbf{A}_{ij} = 0$ **do**
  $\mathbf{h}_i = \mathbf{V}_c^T(\mathbf{x}_i - \bar{\mathbf{x}}_{\text{pool}})$; $\mathbf{h}_j = \mathbf{V}_c^T(\mathbf{x}_j - \bar{\mathbf{x}}_{\text{pool}})$
  $s_{ij} \leftarrow \exp\left(-\frac{1}{2}(\mathbf{h}_i - \mathbf{h}_j)^T(h^2\mathbf{\Lambda}_c)^{-1}(\mathbf{h}_i - \mathbf{h}_j)\right)$
  if $s_{ij} \geq \tau_d$ **then**
    $\tilde{\mathbf{A}}_{ij} \leftarrow s_{ij}$
  end if
end for

Stage 2: Anchor Sampling
For any i with $\sum_j \tilde{\mathbf{A}}_{ij} = 0$, set $\tilde{\mathbf{A}}_{ii} \leftarrow 1$
$\tilde{\mathbf{D}}_{ii} \leftarrow \sum_j \tilde{\mathbf{A}}_{ij}$; $\mathbf{M} \leftarrow \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}$
Draw $L \sim \text{Geom}(\beta)$ with $\mathbb{P}(L = k) = \beta(1 - \beta)^k$, $k = 0, 1, \ldots$
Set $L_K \leftarrow \min(L, K)$ and $X_0 \leftarrow u_{n+1}$
for $t = 0, \ldots, L_K - 1$ **do**
  Sample $X_{t+1} \sim \mathbf{M}_{X_t,\cdot}$
end for
$\tilde{u}_{n+1} \leftarrow X_{L_K}$

Stage 3: Weighting & Quantile Computation
Initialize $\boldsymbol{\pi}^{(0)} \leftarrow \mathbf{e}_{\tilde{u}_{n+1}}$
for $t = 0, \ldots, K - 1$ **do**
  $\boldsymbol{\pi}^{(t+1)} \leftarrow (1 - \beta)\boldsymbol{\pi}^{(t)}\mathbf{M} + \beta\mathbf{e}_{\tilde{u}_{n+1}}$
end for
$\tilde{\boldsymbol{\pi}}_{n+1} \leftarrow \boldsymbol{\pi}^{(K)}$
for $i \in \mathcal{D}_{cal} \cup \{u_{n+1}\}$ **do**
  $\tilde{w}_i \leftarrow [\tilde{\boldsymbol{\pi}}_{n+1}]_i / d_i \quad \{d_i = \tilde{\mathbf{D}}_{ii}\}$
end for
Normalize weights $\tilde{w}_i \leftarrow \tilde{w}_i / \sum \tilde{w}_j$
$\hat{q} \leftarrow \text{Quantile}_{1-\alpha}\left(\sum \tilde{w}_i \delta_{s(\mathbf{x}_i, y_i)} + \tilde{w}_{n+1}\delta_\infty\right)$
return $\{y \in \mathcal{Y} \mid s(\mathbf{x}_{n+1}, y) \leq \hat{q}\}$

are trained, we fix the pretrained models. The selection fractions (train, validation, calibration, test) are as follows: ANH, CHI, TWT: (0.4, 0.1, 0.1, 0.4) - CMP, CRA, DBLP, PMD, : (0.2, 0.1, 0.2, 0.5) - EDU, ELC, INC, UNM: (0.4, 0.1, 0.1, 0.4) - CBAS, WKB: (0.5, 0.1, 0.1, 0.3). We use GCN [24], as our pretrained model. Across all the experiments, we fix: miscoverage $\alpha = 0.1$, the number os APPNP steps, $K = 30$, the base badnwith $h = 2.0$, and $c = 5$ wherever PCA is used. We choose $\tau_d \in \{0.2, 0.5, 0.8\}$, $\beta \in \{0.2, 0.4\}$. We run the experiments on a single GPU core NVIDIA GeForce RTX 3090 and x86_64 CPUs.

Design Principle of PCA Application PCA is enabled on eight datasets: all seven regression datasets and one classification dataset (WKB). We apply PCA when reducing redundancy can improve similarity estimation in high dimensional embeddings. However, when the original graph provides sufficient local connectivity, PPR can capture informative structural dependencies without aggressive densification. In such cases, dimensionality reduction may increase the apparent similarity of distinct nodes, potentially introducing noisy edges. We therefore retain the original embeddings to favor more conservative densification.



(a) Marginal Cov. and PL

(b) WSC Cov. and PL

Figure 7: Sensitivity analysis of the PPR restart probability β. Higher β values prioritize local neighborhood information, leading to improved marginal and WSC coverage at the cost of a moderate increase in WSC PL.

A.8 Additional Ablation Studies

The previous ablation studies clarified the significant impact of the PPR kernel and the densification step. Therefore, here analyze the effect of the two main hyperparameters within the setup of the PPR kernel and the densification.

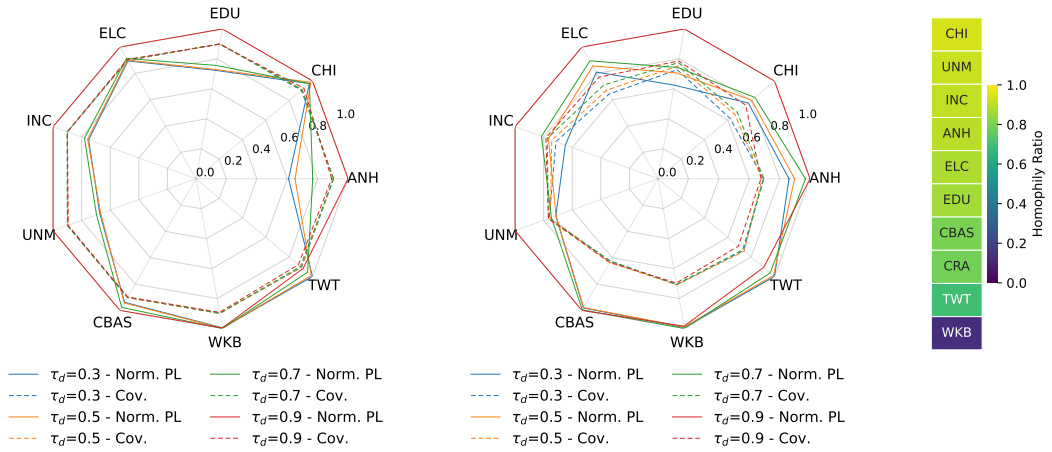
PPR restart probability (β) First, we conduct a sensitivity analysis over the PPR restart probability $\beta \in [0.1, 0.5]$. This hyperparameter controls the kernel's locality extent: lower values allow longer random walks, allowing more nodes to contribute to the weighted CP and vice versa. The results in Figure 7 show that increasing β improves both marginal and WSC (coverage) while moderately increasing PL (prediction length) of the worst-slab scenario. In other words, smaller β yields overly global kernels, whereas larger values emphasize locality. Overall, performance is stable across this range, with intermediate values providing the best balance.

Fixed densification threshold (τ_d) We examine the sensitivity to the fixed threshold $\tau_d \in \{0.3, 0.5, 0.7, 0.9\}$ across 10 datasets. Lower thresholds add more similarity-based edges, while higher thresholds restrict additions to more similar nodes, reducing potentially noisy connections. Figure 8 shows that the effect varies with graph homophily: lower thresholds favor coverage on less homophilous graphs (e.g., WKB), whereas higher thresholds can improve coverage on more homophilous graphs (e.g., ELC). Overall, $\tau_d = 0.5$ provides a favorable balance between coverage and prediction length. Homophily is used only to interpret these results, not to select the threshold.

WSC–prediction-size trade-off. To examine the trade-off between structural localization and prediction size, we interpolate between uniform and PPR kernels:

$$H_\lambda(u, v) = (1 - \lambda)\frac{1}{N} + \lambda H_K(u, v), \qquad \lambda \in [0, 1]. \tag{19}$$

We use the same mixed kernel for anchor sampling and calibration weighting. Here, $\lambda = 0$ recovers SCP and $\lambda = 1$ recovers GRAPHLCP, while intermediate values serve only as a diagnostic. Figure 9 shows that stronger structural localization consistently improves mean WSC, with a gradual increase in prediction size across nine datasets (ANH, CHI, EDU, ELC, INC, UNM, TWT, CBAS, WKB). This highlights the coverage–efficiency trade-off, where the preference for stronger localization depends on the cost of local undercoverage.



(a) Marginal Cov. and PL (b) WSC Cov. and PL

Figure 8: Sensitivity analysis of the fixed densification threshold τ_d. Coverage and prediction length vary with the extent of similarity-based edge additions and graph homophily.

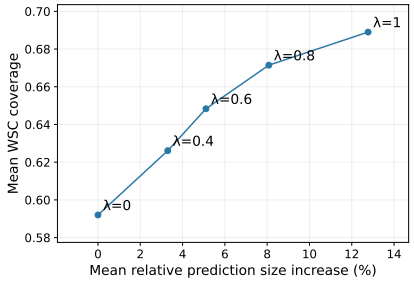


Figure 9: WSC–prediction-size trade-off under mixed uniform–PPR localization across nine datasets. Prediction size is normalized by its value at $\lambda = 0$ within each dataset before averaging.

A.9 More Conditional and Marginal Results

WSC results To have a more detailed picture of how we perform in comparison to the baselines, we show the full results of WSC in Tables 4 and 5. As previously discussed, GRAPHLCP attains a desirable trade-off between prediction length and coverage in comparison with other baselines, and this trade-off is clearer on the regression tasks.

Group-based conditional results We provide additional conditional coverage results for all four types of group-based conditional coverage in Figure 10.

Marginal Prediction Length Table 6 shows that SCP generally yields short prediction sets, while CalLCP is often more conservative. GRAPHLCP produces shorter intervals than CalLCP across the regression datasets, alongside improved WSC over SCP and RLCP (Table 4). Considering Table 1, these comparisons should also account for CalLCP's frequent marginal overcoverage.

A.10 Empirical runtime

We evaluate the wall-clock runtime of GRAPHLCP against SCP and RLCP on larger datasets. Results are in Table 7. As expected, GRAPHLCP incurs noticeable overhead due to densification and PPR-based weighting, especially on DBLP. The fixed similarity threshold can limit added edges, but the additional cost remains a trade-off for improved localization. The results use the same hyperparameters as the main experiments.

Table 4: WSC results. GRAPHLCP achieves strong WSC across regression and classification datasets. "-" means infinite-length prediction interval. OOT (out-of-time) shows results attained after an unreasonably long time.

Method	ANH	CHI	EDU	ELC	INC	UNM	TWT	CMP	CRA	DBLP	CBAS	WKB	PMD
SCP	$0.61^{\pm.04}$	$0.55^{\pm.04}$	$0.61^{\pm.03}$	$0.49^{\pm.04}$	$0.57^{\pm.03}$	$0.71^{\pm.02}$	$0.70^{\pm.03}$	$0.72^{\pm.01}$	$0.69^{\pm.02}$	$0.75^{\pm.01}$	$0.52^{\pm.06}$	$0.61^{\pm.04}$	$0.71^{\pm.01}$
NAPS	-	-	-	-	-	-	-	$0.82^{\pm.01}$	$0.94^{\pm.01}$	$0.96^{\pm.00}$	$1.00^{\pm.00}$	$1.00^{\pm.00}$	$0.97^{\pm.00}$
DAPS	$0.52^{\pm.06}$	$0.43^{\pm.05}$	$0.56^{\pm.03}$	$0.48^{\pm.04}$	$0.50^{\pm.04}$	$0.65^{\pm.03}$	$0.56^{\pm.04}$	$0.72^{\pm.02}$	$0.70^{\pm.02}$	$0.74^{\pm.01}$	$0.52^{\pm.06}$	$0.69^{\pm.03}$	$0.70^{\pm.01}$
RLCP	$0.61^{\pm.04}$	$0.53^{\pm.04}$	$0.61^{\pm.03}$	$0.49^{\pm.04}$	$0.57^{\pm.03}$	$0.71^{\pm.02}$	$0.69^{\pm.03}$	$0.72^{\pm.01}$	$0.69^{\pm.02}$	$0.75^{\pm.01}$	$0.51^{\pm.06}$	$0.60^{\pm.04}$	$0.70^{\pm.01}$
CF-GNN	$0.61^{\pm.04}$	$0.62^{\pm.04}$	$0.67^{\pm.02}$	$0.62^{\pm.04}$	$0.63^{\pm.04}$	$0.72^{\pm.02}$	$0.74^{\pm.03}$	$0.52^{\pm.08}$	$0.64^{\pm.08}$	$0.68^{\pm.07}$	$0.61^{\pm.05}$	$0.61^{\pm.07}$	$0.63^{\pm.04}$
SNAPS	$0.67^{\pm.04}$	$0.62^{\pm.03}$	$0.64^{\pm.03}$	$0.79^{\pm.04}$	$0.58^{\pm.03}$	$0.72^{\pm.03}$	-	$0.73^{\pm.02}$	$0.70^{\pm.02}$	$0.75^{\pm.01}$	$0.56^{\pm.06}$	$0.68^{\pm.02}$	$0.71^{\pm.01}$
RR-Mondrian	$0.64^{\pm.03}$	$0.69^{\pm.03}$	$0.71^{\pm.03}$	$0.58^{\pm.04}$	$0.71^{\pm.02}$	$0.69^{\pm.02}$	$0.67^{\pm.02}$	$0.74^{\pm.02}$	$0.73^{\pm.02}$	$0.75^{\pm.01}$	$0.59^{\pm.05}$	$0.64^{\pm.06}$	$0.79^{\pm.01}$
CalLCP	$0.74^{\pm.03}$	$0.70^{\pm.04}$	$0.80^{\pm.02}$	$0.64^{\pm.03}$	$0.75^{\pm.03}$	$0.76^{\pm.02}$	$0.79^{\pm.02}$	OOT	$0.73^{\pm.01}$	OOT	$0.62^{\pm.05}$	$0.68^{\pm.03}$	$0.79^{\pm.01}$
GRAPHLCP	$0.68^{\pm.03}$	$0.71^{\pm.02}$	$0.75^{\pm.01}$	$0.68^{\pm.01}$	$0.75^{\pm.01}$	$0.77^{\pm.01}$	$0.73^{\pm.03}$	$0.80^{\pm.01}$	$0.72^{\pm.01}$	$0.81^{\pm.00}$	$0.63^{\pm.02}$	$0.70^{\pm.01}$	$0.80^{\pm.01}$

Table 5: WSC prediction length results. GRAPHLCP yields shorter relative prediction lengths on regression tasks than on classification tasks. OOT (out-of-time) shows results attained after an unreasonably long time.

Method	ANH	CHI	EDU	ELC	INC	UNM	TWT	CMP	CRA	DBLP	CBAS	WKB	PMD
SCP	$1.66^{\pm.09}$	$1.94^{\pm.07}$	$2.02^{\pm.10}$	$1.01^{\pm.03}$	$1.94^{\pm.07}$	$1.81^{\pm.04}$	$2.10^{\pm.07}$	$1.03^{\pm.04}$	$0.91^{\pm.04}$	$1.40^{\pm.06}$	$1.99^{\pm.10}$	$3.09^{\pm.34}$	$1.22^{\pm.02}$
NAPS	-	-	-	-	-	-	-	$2.72^{\pm.23}$	$5.18^{\pm.20}$	$3.00^{\pm.10}$	$3.78^{\pm.04}$	$5.00^{\pm.00}$	$2.12^{\pm.10}$
DAPS	$1.48^{\pm.08}$	$1.75^{\pm.06}$	$1.77^{\pm.09}$	$1.01^{\pm.03}$	$1.73^{\pm.06}$	$1.58^{\pm.04}$	$1.60^{\pm.05}$	$1.41^{\pm.10}$	$0.93^{\pm.04}$	$1.37^{\pm.05}$	$2.26^{\pm.08}$	$4.07^{\pm.10}$	$1.27^{\pm.02}$
RLCP	$1.66^{\pm.09}$	$1.91^{\pm.07}$	$2.02^{\pm.10}$	$1.01^{\pm.03}$	$1.94^{\pm.07}$	$1.81^{\pm.04}$	$2.10^{\pm.07}$	$1.03^{\pm.04}$	$0.90^{\pm.04}$	$1.39^{\pm.06}$	$1.99^{\pm.10}$	$3.06^{\pm.33}$	$1.22^{\pm.02}$
CF-GNN	$1.82^{\pm.22}$	$1.67^{\pm.14}$	$1.88^{\pm.12}$	$0.85^{\pm.02}$	$1.80^{\pm.07}$	$1.61^{\pm.05}$	$2.26^{\pm.07}$	$3.68^{\pm.58}$	$1.55^{\pm.30}$	$2.01^{\pm.26}$	$2.56^{\pm.29}$	$3.78^{\pm.24}$	$1.41^{\pm.21}$
SNAPS	$1.70^{\pm.09}$	$2.10^{\pm.04}$	$2.13^{\pm.03}$	$1.23^{\pm.03}$	$1.95^{\pm.04}$	$1.81^{\pm.05}$	-	$1.37^{\pm.12}$	$0.91^{\pm.04}$	$1.35^{\pm.04}$	$2.78^{\pm.16}$	$4.20^{\pm.05}$	$1.19^{\pm.01}$
RR-Mondrian	$1.75^{\pm.23}$	$1.78^{\pm.08}$	$1.53^{\pm.24}$	$0.51^{\pm.08}$	$1.37^{\pm.20}$	$1.41^{\pm.26}$	$2.58^{\pm.58}$	$3.27^{\pm.40}$	$2.40^{\pm.34}$	$1.96^{\pm.07}$	$2.20^{\pm.25}$	$2.65^{\pm.33}$	$2.11^{\pm.05}$
CalLCP	$2.31^{\pm.26}$	$2.33^{\pm.09}$	$2.62^{\pm.14}$	$1.10^{\pm.03}$	$2.49^{\pm.08}$	$2.17^{\pm.07}$	$2.54^{\pm.14}$	OOT	$1.39^{\pm.07}$	OOT	$1.60^{\pm.22}$	$4.14^{\pm.08}$	$1.67^{\pm.03}$
GRAPHLCP	$1.66^{\pm.12}$	$2.17^{\pm.14}$	$2.44^{\pm.24}$	$1.09^{\pm.06}$	$2.52^{\pm.21}$	$1.85^{\pm.05}$	$2.10^{\pm.17}$	$4.41^{\pm.06}$	$2.00^{\pm.04}$	$1.90^{\pm.08}$	$2.75^{\pm.09}$	$4.21^{\pm.06}$	$1.93^{\pm.03}$

A.11 Datasets

We show the detailed statistics of all datasets in Table 8. We evaluate our proposed method across a diverse set of benchmark datasets across multiple domains and task types. For citation networks, we include Cora_ML (CRA), DBLP, and PubMed (PMD), while the Amazon co-purchase domain is represented by Computers (CMP). We utilize a set of geographic and transportation datasets for regression tasks, including Education (EDU), Election (ELC), Income (INC), Unemployment (UNM), Anaheim (ANH), and Chicago (CHI). Finally, we incorporate the Twitch (TWT) social network and finally GOOD-WebKB (WKB) and GOOD-CBAS (CBAS).

Table 6: Marginal prediction length. OOT (out-of-time) shows results attained after an unreasonably long time.

Method	ANH	CHI	EDU	ELC	INC	UNM	TWT	CMP	CRA	DBLP	CBAS	WKB	PMD
SCP	$1.66^{\pm.09}$	$1.94^{\pm.07}$	$2.02^{\pm.10}$	$1.01^{\pm.03}$	$1.94^{\pm.07}$	$1.81^{\pm.04}$	$2.10^{\pm.07}$	$1.08^{\pm.01}$	$0.98^{\pm.01}$	$1.16^{\pm.01}$	$1.92^{\pm.03}$	$4.26^{\pm.10}$	$1.09^{\pm.00}$
NAPS	-	-	-	-	-	-	$5.73^{\pm1.18}$	$2.19^{\pm.06}$	$5.54^{\pm.06}$	$3.26^{\pm.01}$	$3.79^{\pm.03}$	$5.00^{\pm.00}$	$2.65^{\pm.01}$
DAPS	$1.48^{\pm.08}$	$1.75^{\pm.06}$	$1.77^{\pm.09}$	$1.01^{\pm.03}$	$1.73^{\pm.06}$	$1.58^{\pm.04}$	$1.60^{\pm.05}$	$1.26^{\pm.02}$	$0.99^{\pm.01}$	$1.16^{\pm.01}$	$2.05^{\pm.04}$	$4.41^{\pm.06}$	$1.12^{\pm.01}$
RLCP	$1.66^{\pm.09}$	$1.91^{\pm.07}$	$2.02^{\pm.10}$	$1.01^{\pm.03}$	$1.94^{\pm.07}$	$1.81^{\pm.04}$	$2.10^{\pm.07}$	$1.08^{\pm.01}$	$0.98^{\pm.01}$	$1.16^{\pm.01}$	$1.92^{\pm.03}$	$4.24^{\pm.10}$	$1.09^{\pm.00}$
CF-GNN	$1.82^{\pm.22}$	$1.67^{\pm.14}$	$1.88^{\pm.12}$	$0.85^{\pm.02}$	$1.80^{\pm.07}$	$1.61^{\pm.05}$	$2.26^{\pm.07}$	$3.56^{\pm.63}$	$1.55^{\pm.26}$	$1.82^{\pm.25}$	$2.40^{\pm.29}$	$4.41^{\pm.15}$	$1.30^{\pm.13}$
SNAPS	$1.70^{\pm.09}$	$2.10^{\pm.04}$	$2.13^{\pm.03}$	$1.23^{\pm.03}$	$1.95^{\pm.04}$	$1.81^{\pm.05}$	-	$1.22^{\pm.05}$	$0.98^{\pm.01}$	$1.16^{\pm.01}$	$2.40^{\pm.03}$	$4.32^{\pm.04}$	$1.09^{\pm.00}$
RR-Mondrian	$2.12^{\pm.29}$	$1.70^{\pm.07}$	$2.18^{\pm.16}$	$1.21^{\pm.07}$	$2.05^{\pm.11}$	$2.30^{\pm.19}$	$4.61^{\pm.99}$	$2.62^{\pm.13}$	$2.36^{\pm.29}$	$1.71^{\pm.02}$	$2.73^{\pm.21}$	$4.41^{\pm.11}$	$1.43^{\pm.01}$
CalLCP	$2.24^{\pm.28}$	$2.15^{\pm.06}$	$2.24^{\pm.14}$	$1.06^{\pm.02}$	$2.12^{\pm.08}$	$2.13^{\pm.06}$	$2.54^{\pm.13}$	OOT	$1.54^{\pm.03}$	OOT	$2.71^{\pm.02}$	$4.59^{\pm.03}$	$1.15^{\pm.01}$
GRAPHLCP	$1.70^{\pm.11}$	$1.92^{\pm.08}$	$2.01^{\pm.10}$	$1.03^{\pm.03}$	$1.97^{\pm.06}$	$1.84^{\pm.05}$	$2.09^{\pm.15}$	$4.33^{\pm.01}$	$2.10^{\pm.02}$	$1.81^{\pm.01}$	$2.81^{\pm.04}$	$4.45^{\pm.02}$	$1.82^{\pm.01}$

Table 7: Wall-Clock Runtime on Larger Datasets (seconds)

Dataset	SCP	RLCP	GRAPHLCP
CMP	4.4	29.7	111.1
DBLP	7.5	28.3	508.8
PMD	9.2	30.7	145.2

Table 8: Dataset Statistics

Dataset	Domain	#Nodes	#Edges	#Features	#Classes
Cora_ML [4]	Citation	2,995	16,316	2,879	7
DBLP [4]	Citation	17,716	105,734	1,639	4
PubMed [4]	Citation	19,717	88,648	500	3
Computers [39]	Amazon	13,752	491,722	767	10
Education [22]	Geography	3,234	12,717	6	Reg.
Election [22]	Geography	3,234	12,717	6	Reg.
Income [22]	Geography	3,234	12,717	6	Reg.
Unemployment [22]	Geography	3,234	12,717	6	Reg.
Anaheim [22]	Transportation	914	3,881	4	Reg.
Chicago [22]	Transportation	2,176	15,104	4	Reg.
Twitch [22]	Social	1,912	31,299	3,170	Reg.
GOOD-WebKB [17]	OOD Benchmark	617	1138	1703	5
GOOD-CBAS [17]	OOD Benchmark	700	3962	4	4

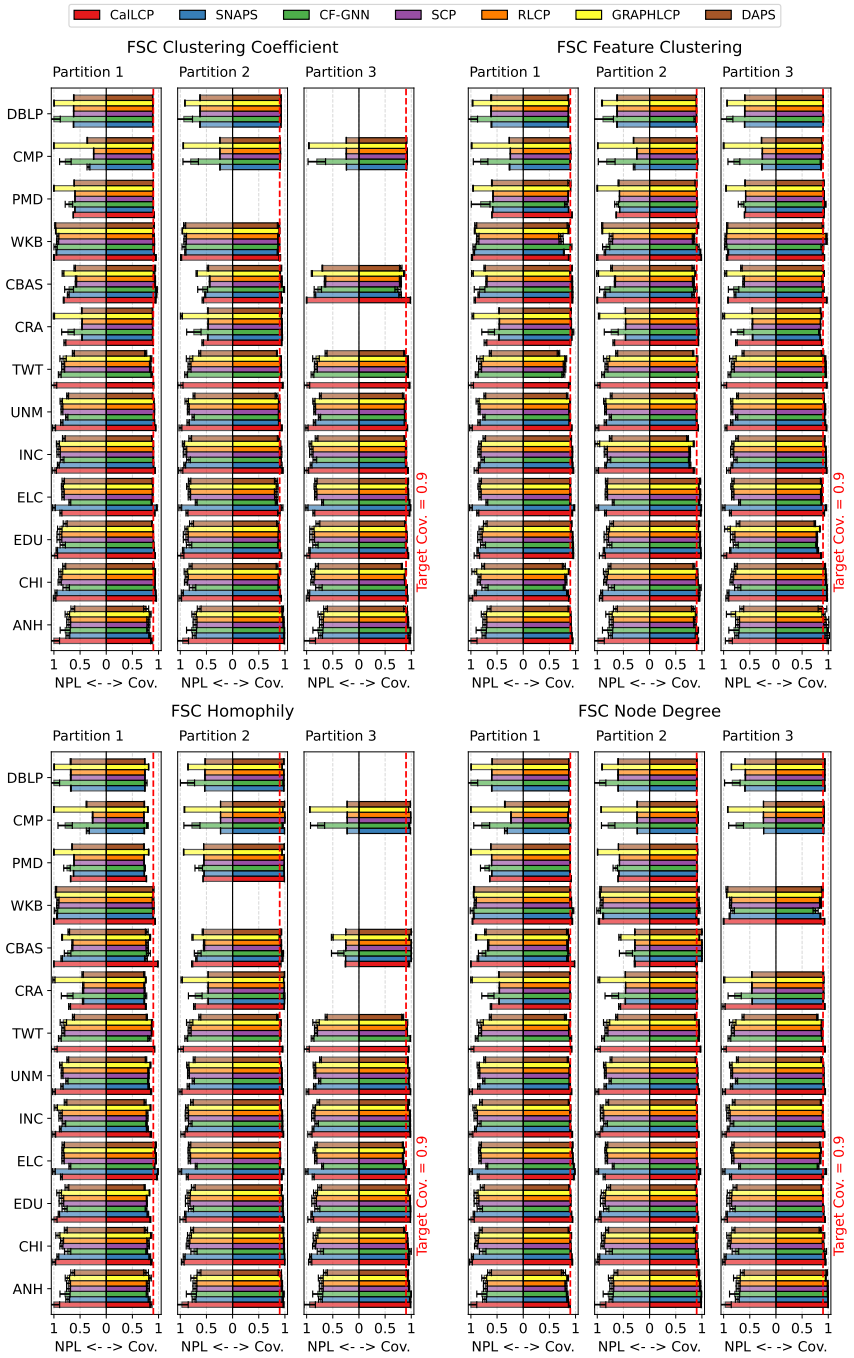


Figure 10: Results for group-based conditional coverage. It shows the minimum coverage across partitions for semantic and structural features where $\alpha = 0.1$. GRAPHLCP demonstrates superior coverage compared to baselines and more specifically in homophily and feature clustering.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction outline the paper's main contributions and scope. We present our contributions more specifically in Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Our work is grounded in previous theoretical results of weighted conformal prediction and the permutation invariance/equivariant assumption, which are clearly explained throughout the paper (More specifically, Sections 2.1, 2.2, 3, and 3.1). Also, all necessary references are cited. We also provide new theoretical results to motivate our method, and proofs are in Appendix A.1.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided the code at https://github.com/pbaghershahi/GraphLCP.git. Besides, the experimental setup for reproducibility has been discussed in Section 4 and Appendix A.7.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have provided the code at https://github.com/pbaghershahi/GraphLCP.git (lined in the main body as well). Besides, the datasets have been discussed in Section 4 and Appendix A.7.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: the details have been provided in Section 4 and Appendix A.7.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All the results presented in Section 4 and Appendix A.7 have reported appropriate information accordingly. All major results report mean and standard deviation over multiple seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide computation machine details in Section A.7 and show empirical wall-clock runtime in Section A.10.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: We perform fundamental research on uncertainty quantification of GNN models in this paper. We do not foresee any specific negative societal impact from this work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use publicly available datasets and model implementations, and properly credit them by citing the corresponding papers.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We do not release any new assets. We use publicly available datasets and machine learning models for evaluation, which are not new assets. Although we have included the code link in the paper, it is not public yet. We will release the code publicly upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.