big data and cognitive computing

MDPI

Article

SETTA: Parameter-Free Test-Time Adaptation for Graph Neural Networks via Spectral-Energy-Guided Semantic Refinement

Dongyang Yu [1], Xia Cui [2,*] and Rong Xiao [3,*]

[1] Manchester Metropolitan Joint Institute, Hubei University, Wuhan 430062, China; yudongyang@stu.hubu.edu.cn
[2] School of Computing and Mathematics, Manchester Metropolitan University, Manchester M15 6BH, UK
[3] School of Computer and Information Engineering, Hubei University, Wuhan 430062, China
* Correspondence: x.cui@mmu.ac.uk (X.C.); x_rong@hubu.edu.cn (R.X.)

Abstract

Node classification is a central graph data mining task, yet repeated message passing can over-smooth representations and degrade frozen graph neural network (GNN) predictions after deployment. We present SETTA (Spectral-Energy Test-Time Adaptation), a prediction-level graph test-time adaptation framework that refines frozen outputs without test labels, gradients, parameter updates, or learnable adaptation parameters. SETTA denoises features for semantic-neighbor construction, adds complementary semantic routes while preserving observed edges, monitors a smoothness-energy proxy during diffusion, and accepts refinements through entropy-based gating. Configurations are fixed by a dataset-level protocol or selected using validation data only. Across six mostly homophilic benchmarks with 2708–19,717 nodes, SETTA improved a frozen two-layer GCN on every dataset and achieved the highest mean accuracy among the evaluated methods on five, with gains of 4.61, 3.08, and 2.01 percentage points on Cora, CiteSeer, and PubMed, respectively. Positive mean gains were also observed across all 30 dataset–backbone settings. Ablations and transition analyses indicate that semantic injection is most beneficial on sparse citation graphs and that selective refinement limits harmful changes. The current dense implementation supports benchmark-scale, amortized refinement; scalability and robustness on heterophilic graphs remain open.

Keywords: graph data mining; graph neural networks; node classification; over-smoothing; semantic graph refinement; test-time adaptation

Check for updates

Academic Editors: Vikas Palakonda and Samira Ghorbanpour

Received: 1 June 2026
Revised: 29 July 2026
Accepted: 2 August 2026
Published: 4 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In graph data mining, node classification extracts label information from relational structures such as citation, co-purchase, coauthor, social, and knowledge graphs. Graph neural networks (GNNs) are a standard tool for this setting because they mine relational patterns through message passing [1–3]. In recommender systems [4], molecular graphs [5,6], and social networks [2], each node updates its representation by aggregating information from neighboring nodes. This mechanism exploits relational structures and creates a persistent tension between useful information exchange and representation collapse.

This collapse is commonly described as over-smoothing. Early analyses connected GCN propagation to Laplacian smoothing [7]; subsequent theory showed that repeated propagation can reduce expressive power and drive node representations toward low-variation subspaces [8,9]. Figure 1 illustrates this low-variation effect on the Karate Club

graph: repeated normalized graph diffusion contracts a fixed random node-signal matrix in a shared projection space, while its relative Dirichlet energy decreases from 1.000 to 0.0085 after 16 propagation steps. Similar degradation can appear even in shallow practical GNNs when diffusion is strong or topology is poorly aligned with class structure [10,11].

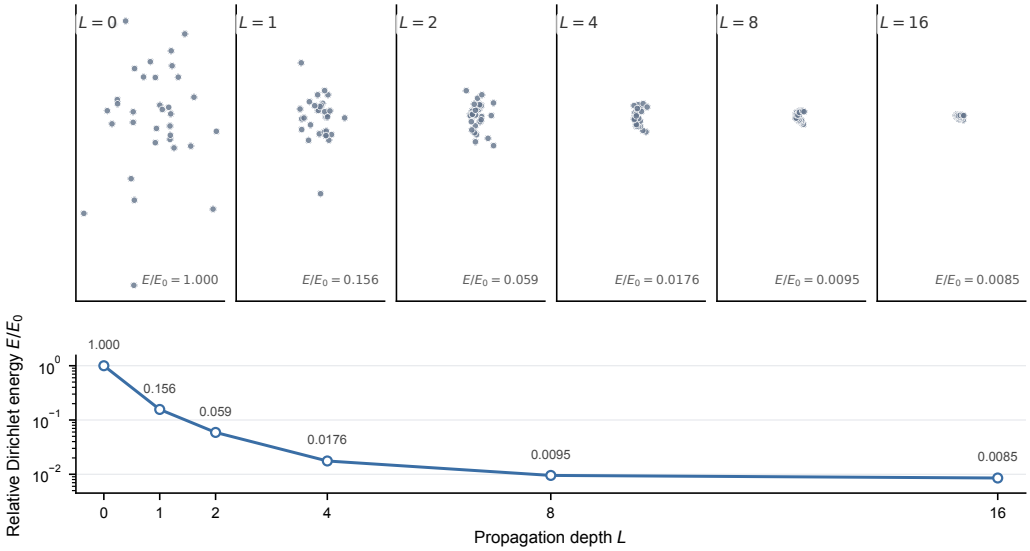


Figure 1. Fixed Gaussian node signals contract under repeated normalized graph diffusion on the Karate Club graph. All projections share one basis and identical axis limits; the lower panel reports relative Dirichlet energy E/E_0.

Many anti-over-smoothing methods act during training. Residual, initial, and jumping connections preserve earlier representations [11,12]; normalization and stochastic edge removal maintain representation diversity [13,14]; decoupled propagation separates prediction from diffusion [15–17]; label-propagation-based refinement corrects and smooths predictions [18]; and attention or rewiring methods modify information flow [19,20]. These approaches improve robustness, but often assume that the model can be redesigned, retrained, optimized at test time, corrected with labels, or coupled to a new propagation operator.

Those assumptions are not always available after deployment. A trained GNN may be embedded in an operational data-mining workflow, shared as a frozen model, or evaluated on unlabeled graph data where labels, gradients, and parameter updates are inaccessible. Existing anti-over-smoothing, rewiring, post-hoc correction, and graph test-time adaptation methods do not jointly address this restricted interface: they generally depend on model redesign or retraining, label-assisted correction, test-time optimization, or a replacement propagation operator. The resulting research gap is prediction-level adaptation under a frozen-model constraint, where only predictions, node features, and the observed topology are available.

This gap motivates the central question of this paper: can frozen GNN predictions be refined without test labels, gradients, parameter updates, or learnable adaptation parameters while controlling the risk of harmful diffusion? To study this question, SETTA combines topology-preserving semantic route construction with graph-level propagation control and node-level selective acceptance. Figure 2 provides a high-level overview; the construction rules, propagation update, stopping criterion, gating rule, configuration protocol, and computational assumptions are specified in Section 3.

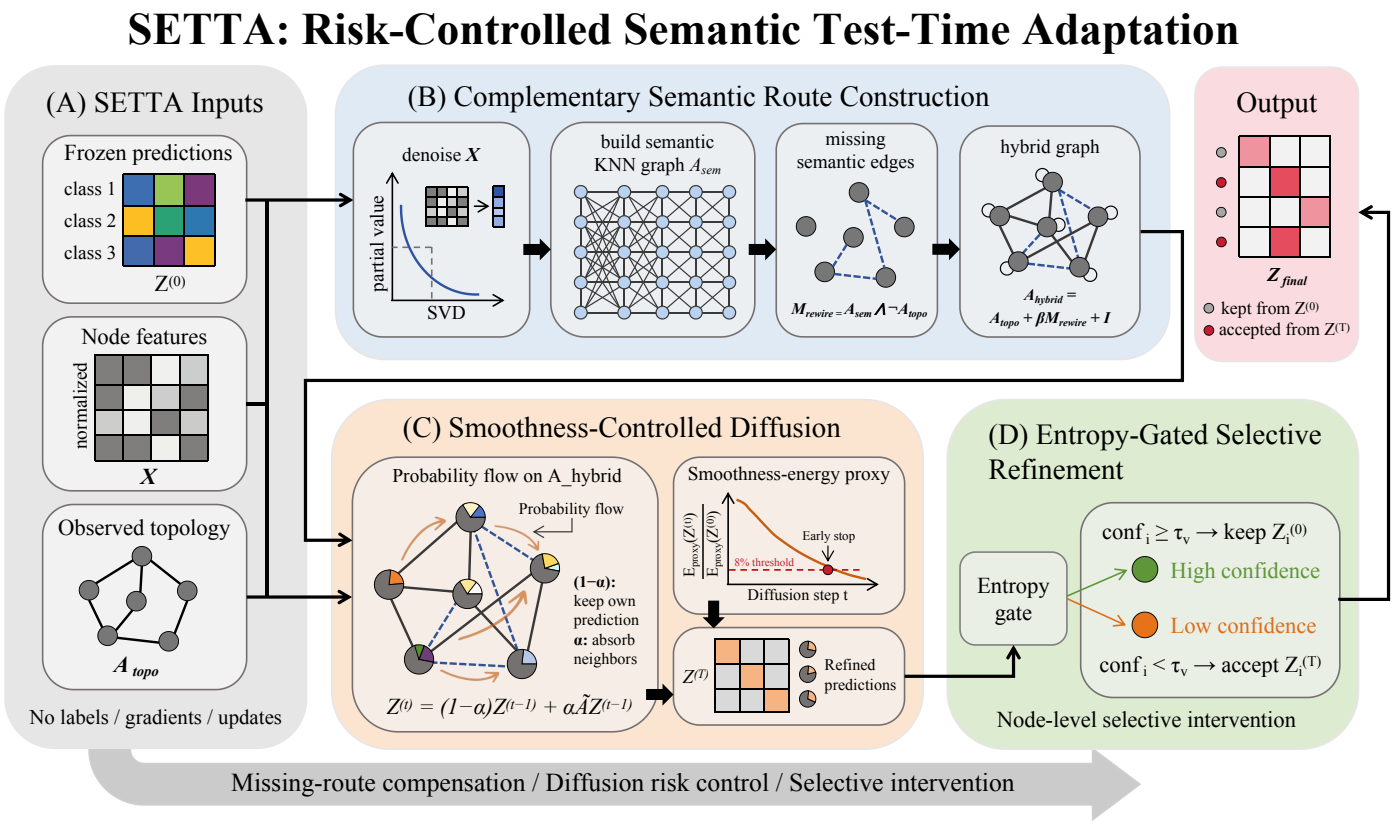


Figure 2. The SETTA framework. SETTA performs prediction-level graph test-time adaptation over frozen GNN predictions without learnable adaptation parameters during refinement. It couples complementary semantic route construction, smoothness-controlled diffusion, and entropy-gated selective refinement. The final output $\mathbf{Z}_{\text{final}}$ is selected node-wise from the original prediction $\mathbf{Z}^{(0)}$ and the refined prediction $\mathbf{Z}^{(T)}$.

The novelty lies in the frozen-deployment protocol and in the integration of complementary route construction with two levels of risk control, rather than in claiming semantic graphs, diffusion, or entropy estimates as isolated new primitives.

The main contributions of this work are as follows:

- Frozen GNN prediction refinement is formulated as graph test-time adaptation without test labels, gradients, parameter updates, or learnable adaptation parameters during refinement.
- A risk-controlled semantic adaptation principle is introduced in which complementary semantic routes are constructed, frozen predictions are propagated through the hybrid graph, and refinements are accepted selectively.
- Feature-denoised semantic edge injection, smoothness-energy-monitored propagation, and entropy-based node-level gating are integrated to strengthen semantic diffusion while limiting harmful intervention.
- The framework is evaluated on six node-classification benchmarks using trained-GNN and decoupled-propagation comparisons, ablations, node-level transition analysis, backbone transfer across five frozen GNN predictors, and paired significance tests over 10 seeds.
- Validation-based configuration selection, sensitivity, one-time graph construction cost, and dense-memory limitations are reported to define the present benchmark-scale scope.

2. Related Work

2.1. Over-Smoothing and Anti-over-Smoothing GNNs

Over-smoothing is a central failure mode of deep message passing. Li et al. [7] interpreted GCN layers as Laplacian smoothing, Oono and Suzuki [8] showed that repeated propagation can reduce expressive power, and Cai and Wang [9] linked propagation collapse to Dirichlet-energy decay. Later work refined this view and questioned whether Dirichlet energy alone diagnoses all forms of representational collapse [10,21].

Training-stage anti-over-smoothing methods typically modify the model or training process. GCNII uses initial residual and identity mapping mechanisms [11]; Jumping Knowledge Networks aggregate representations from different depths [12]; PairNorm controls representation variance [13]; and DropEdge reduces excessive smoothing by randomly removing edges during training [14]. These methods are effective when the practitioner can change the backbone architecture or training recipe, but they do not provide a model-agnostic correction interface for already frozen predictions.

2.2. Decoupled Propagation and Post-Hoc Refinement

Decoupled propagation methods show that prediction and diffusion need not be fused into a single trainable message-passing stack. Approximate Personalized Propagation of Neural Predictions (APPNP) propagates neural predictions with a personalized PageRank-style operator [15]. Simplified Graph Convolution (SGC) collapses repeated graph convolution into a linear propagation step [17], the Scalable Inception Graph Neural Network (SIGN) precomputes multi-scale diffusion features for scalable learning [22], and Graph Diffusion Convolution (GDC) constructs an alternative diffusion operator before learning or inference [16]. Correct and Smooth (C&S) further shows that simple predictors can be improved by residual correction and label smoothing [18].

These methods motivate prediction-level refinement, but they do not fully match the frozen-deployment setting considered here. APPNP, SGC, SIGN, and GDC are coupled to a chosen propagation operator, predictor, or feature-precomputation pipeline, whereas C&S uses labeled residuals during correction. Such assumptions are difficult to satisfy when only frozen predictions, node features, and topology are available at test time.

2.3. Graph Rewiring and Semantic Graph Construction

Graph rewiring methods modify topology to improve information flow or reduce harmful propagation [20]. Diffusion-based rewiring constructs alternative graph operators from the observed topology [16]; graph structure learning (GSL) infers or refines edges from data [23,24]; and curvature- or spectrum-based rewiring targets bottlenecks and over-squashing [25,26]. Recent spectral and community-guided approaches have explored edge deletion, edge addition, and feature-aware rewiring for over-squashing or over-smoothing [27–29]. More generally, semantic k-nearest-neighbor graphs use feature similarity to create propagation routes absent from the observed graph.

Topology modification can be risky when it replaces, deletes, or globally rewires the original graph. Feature similarity may also be unreliable in noisy feature spaces or heterophilic graphs, where feature-near nodes are not necessarily label-compatible. In addition, most learned structure methods require joint training or optimization and therefore cannot be inserted after a predictor has been frozen.

2.4. Graph Test-Time Adaptation

Graph test-time adaptation (TTA) studies how graph models adjust at deployment time under distribution shift or changing graph conditions. In broader TTA, entropy minimization is a common mechanism for updating normalization or model parameters on

unlabeled test data [30]. Graph-specific work has explored contrastive adaptation [31], test-time graph transformation [32], test-time training (TTT) for graph neural networks [33], and homophily-aware or structure-shift objectives [34–36]. These studies support deployment-stage adjustment, but most optimize an objective, update model components, or maintain adapted state at test time. They therefore address a broader interface than prediction-only refinement of a fully frozen model.

2.5. Uncertainty-Aware Selective Refinement

Selective prediction and uncertainty-aware decision control recognize that not every prediction should be treated identically [37]. Calibration studies show that neural-network confidence can be misaligned with empirical correctness [38]. In graph learning, Bayesian and uncertainty-aware models estimate predictive uncertainty to improve robustness or calibration [39]. The common principle is that confidence information can help decide when a model should act, abstain, or request correction.

These approaches often rely on learned confidence models, Monte Carlo sampling, ensembles, or post-hoc calibration. Such machinery is not always available through a frozen-prediction interface, and confidence alone cannot certify that a proposed correction is accurate.

2.6. Deployment Constraints and Research Gap

The reviewed method families leave a specific deployment gap. Training-stage defenses require architecture or optimization access; decoupled diffusion is normally coupled to a predefined predictor or operator; label-based post-processing requires supervision during correction; rewiring and GSL may replace topology or require learning; and existing graph TTA or TTT generally performs test-time optimization. No representative method in this comparison jointly accepts arbitrary frozen predictions, avoids labels and gradients during refinement, preserves the observed topology while adding semantic routes, and applies both graph-level and node-level risk control. The present study targets this combination of constraints. Table 1 reports the comparison at the method level rather than grouping all approaches into broad families.

Table 1. Method-level comparison under frozen-deployment constraints. "Frozen pred." indicates whether a method can directly refine an arbitrary prediction matrix without retraining its source model. "Adapt. state" indicates whether parameters, model state, or test-time variables are learned or updated for adaptation.

Method	Labels in Refinement	Grad./Model Update	Frozen Pred.	Topology	Adapt. State	Semantic Routes	Risk Control
APPNP [15]	No	No	Partial	Preserve	None	No	Restart
SGC/SIGN [17,22]	No	Training required	No	Precompute	Predictor	No	Fixed depth
C&S [18]	Yes	No	Yes	Preserve	None	No	Label correction
GDC [16]	No	Training required	No	Replace operator	None	No	Sparsification
GSL/rewiring [20,23,24]	Varies	Usually	Usually no	Modify/replace	Often	Varies	Structural objective
TENT [30]	No	Yes	No	Preserve	Updated	No	Entropy objective
GraphTTA [31]	No	Yes	No	Augment	Updated	No	Contrastive objective
GTrans [32]	No	Yes	No	Transform	Optimized	Varies	Transform objective
GT3 [33]	No	Yes	No	Preserve	Updated	No	Self-supervision
SETTA	No	No	Yes	Preserve + add	None	Yes	Energy + entropy

Notes: "Partial" for APPNP means that prediction-like scores are propagated, but the propagation operator is normally specified as part of the predictive architecture rather than used as a model-agnostic adaptation interface. "Training required" indicates that a new predictor is trained with the specified precomputation or operator. "Varies", "usually", and "often" reflect implementation-dependent choices within the GSL and rewiring family.

3. Materials and Methods

3.1. Problem Setting and Notation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with N nodes, adjacency matrix $\mathbf{A}_{\text{topo}}$, node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, and C classes. A backbone GNN f_θ is trained on labeled training nodes using the standard supervised objective. After training, SETTA receives the frozen prediction matrix $\mathbf{Z}^{(0)}$, the node features $\mathbf{X}$, and the topology $\mathbf{A}_{\text{topo}}$. For a fixed configuration, the adaptation step is strictly post-hoc in implementation and test-time in use: it does not access test labels, compute gradients, or update model parameters.

For entropy and gating, $\mathbf{Z}^{(0)}$ is treated as a class-probability matrix. If the backbone returns log-probabilities, they should be converted to probabilities before entropy is computed. This convention keeps the propagation, entropy, and gating steps mathematically aligned.

3.2. Design Rationale: From Missing Semantic Routes to Risk-Controlled Adaptation

SETTA is designed around the premise that some frozen GNN errors arise not only from the backbone architecture, but also from incomplete propagation routes in the observed graph. This is consistent with graph structure learning work showing that learned or feature-informed edges can change message-passing behavior [23,24]. If feature-aligned nodes are not connected, a trained GNN may never pass useful class evidence between them. Reliable semantic-neighbor construction is therefore required, and feature denoising suppresses noisy high-dimensional directions before nearest-neighbor search.

Reliable semantic neighbors allow SETTA to construct complementary routes absent from the observed topology. These routes strengthen diffusion over frozen predictions, but stronger diffusion also increases over-smoothing and error-propagation risk. SETTA therefore couples semantic edge injection with smoothness-energy-proxy monitoring as graph-level safety control.

Graph-level safety is not sufficient because nodes need different degrees of intervention. Even when a propagation step is globally acceptable, a confident node may be harmed by unnecessary refinement, whereas an uncertain node may benefit. Entropy gating therefore performs node-level selective intervention. The resulting design is a risk-controlled adaptation loop that constructs semantic routes, propagates through them, and selectively accepts the refined prediction.

3.3. Adaptive Feature Denoising for Reliable Semantic Neighbors

Semantic graph construction can be distorted by noisy high-dimensional features. SETTA therefore applies truncated SVD to continuous feature matrices before nearest-neighbor search [40]. SVD is not an independent classifier enhancer; it is used here as a deterministic dimensionality-reduction heuristic for stabilizing semantic-neighbor construction. The retained dimension is computed as

$$d_{\text{svd}} = \min\left(D,\ \max\left(128,\ \left\lfloor \frac{D}{8} \right\rfloor\right)\right). \tag{1}$$

For Coauthor-CS [41], where $D = 6805$, this gives $d_{\text{svd}} = 850$. For continuous features with $D > 128$ but $\lfloor D/8 \rfloor < 128$, 128 dimensions are retained. When $D \leq 128$, the outer minimum gives $d_{\text{svd}} = D$, and the implementation bypasses SVD rather than expanding the feature dimension. For binary bag-of-words datasets such as Cora and CiteSeer [42], SETTA also bypasses SVD and uses Jaccard similarity [43] directly. The denoising step is distinct from semantic graph construction: in the current implementation, dense nearest-neighbor construction dominates memory use and preprocessing cost as graph size increases.

The constants 128 and 8 were set during method design as simple capacity and compression anchors; they were not obtained from a theoretical derivation or a dataset-specific parameter search. The value 128 acts as a target dimensionality floor only when dimensionality reduction is activated, while the outer minimum ensures that the retained dimension never exceeds D. The divisor 8 substantially compresses high-dimensional features while allowing the retained dimension to grow with D. This same deterministic rule is applied to all continuous-feature datasets and is not selected using test labels. The ablation in Section 4 compares the resulting representation with raw features; it does not establish that 128 and 8 are universally optimal, and sensitivity to these two anchors remains to be studied.

3.4. Complementary Semantic Route Construction

The observed topology may omit feature-aligned propagation routes, especially in sparse citation graphs. From the denoised features, SETTA constructs a semantic k-nearest-neighbor graph $\mathbf{A}_{\text{sem}}$, a nonparametric counterpart to feature-aware graph construction in graph structure learning [23,24]. Cosine similarity is used for continuous features and Jaccard similarity [43] for binary features. The semantic graph does not replace the observed topology; instead, SETTA preserves original edges and identifies only complementary semantic edges:

$$\mathbf{M}_{\text{rewire}} = \mathbf{A}_{\text{sem}} \wedge \neg \mathbf{A}_{\text{topo}}. \tag{2}$$

Here, $\neg \mathbf{A}_{\text{topo}}$ denotes the element-wise Boolean complement of the binary topological adjacency matrix, excluding the diagonal. Thus, $\mathbf{M}_{\text{rewire},ij} = 1$ only if (i, j) is selected by the binary semantic k-nearest-neighbor graph, $\mathbf{A}_{\text{topo},ij} = 0$, and $i \neq j$; otherwise, $\mathbf{M}_{\text{rewire},ij} = 0$. Self-loops are excluded from $\mathbf{M}_{\text{rewire}}$ and are added separately through $\mathbf{I}$.

The hybrid propagation graph is then defined as

$$\mathbf{A}_{\text{hybrid}} = \mathbf{A}_{\text{topo}} + \beta \mathbf{M}_{\text{rewire}} + \mathbf{I}, \tag{3}$$

where β controls the weight of injected semantic edges and $\mathbf{I}$ adds self-loops. The hybrid adjacency depends on β, but not on the propagation coefficient α. The hybrid adjacency is row-normalized before prediction propagation. Figure 3 illustrates this topology-preserving semantic injection procedure.

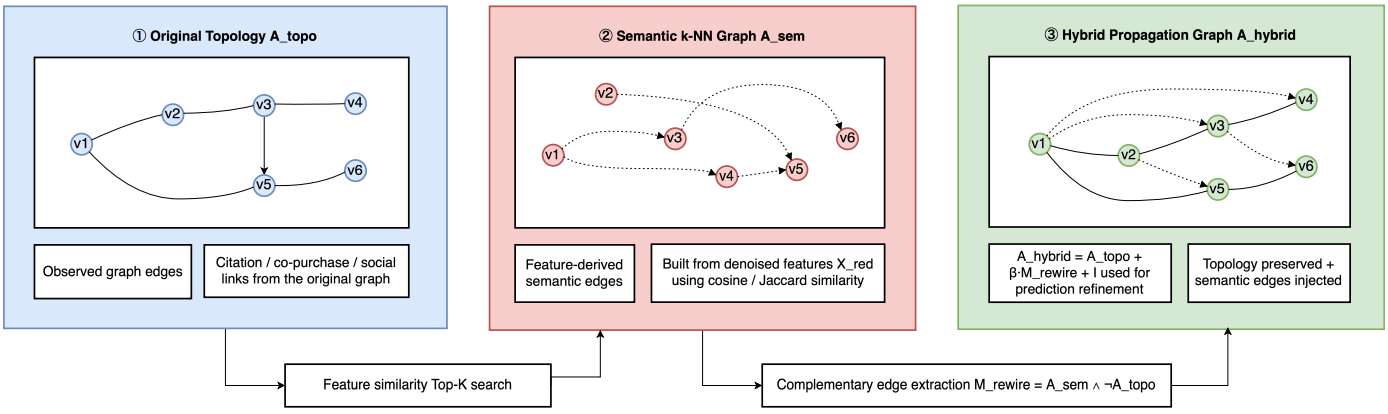


Figure 3. Topology-semantic complementary edge injection. Original topological edges are preserved, feature-aligned semantic neighbors are identified, and only semantic edges absent from the original topology are injected into the hybrid propagation graph.

This controlled perturbation lets feature-similar nodes that are not topologically adjacent exchange prediction information while retaining the original topology as the structural backbone. In graph-operator terms, injected semantic edges alter the propagation basis and

can reconnect feature-aligned regions, but they do not guarantee that every injected edge preserves class-discriminative signal.

3.5. Smoothness-Energy-Monitored Controlled Propagation

Semantic edge injection expands diffusion routes, but stronger diffusion can also increase over-smoothing and error-propagation risk [7–9]. SETTA therefore refines predictions with an energy-monitored lazy prediction diffusion update:

$$\mathbf{Z}^{(t)} = (1-\alpha)\mathbf{Z}^{(t-1)} + \alpha\tilde{\mathbf{A}}\mathbf{Z}^{(t-1)}. \tag{4}$$

Unlike β, which changes the weight assigned to semantic edges in Equation (3), α does not alter the graph. It is a lazy-diffusion mixing coefficient in Equation (4): $1-\alpha$ retains the previous prediction mass, whereas α controls the contribution absorbed from the propagated prediction at each step.

After each step, SETTA computes a row-normalized Dirichlet-energy proxy of the propagated prediction signal on the hybrid graph:

$$\mathcal{E}_{\text{proxy}}(\mathbf{Z}) = \text{Tr}\Big(\mathbf{Z}^T(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{Z}\Big), \tag{5}$$

where $\tilde{\mathbf{A}}$ is the row-normalized hybrid propagation matrix. This quantity is used as a practical collapse-monitoring signal rather than as a complete spectral characterization of over-smoothing. Equation (6) is an algorithmic stopping rule rather than an additional objective. To match the implementation exactly, let $s = 0, 1, \ldots, T_{\max}-1$ denote the zero-based loop index. At iteration s, Equation (4) generates a candidate $\widehat{\mathbf{Z}}^{(s+1)}$ from the most recently accepted iterate $\mathbf{Z}^{(s)}$. The candidate is then checked before assignment:

$$\text{stop if } s \geq 3 \ \wedge \ \frac{\mathcal{E}_{\text{proxy}}(\widehat{\mathbf{Z}}^{(s+1)})}{\mathcal{E}_{\text{proxy}}(\mathbf{Z}^{(0)})} < 0.08. \tag{6}$$

Because s is zero-based, the first three candidates, $\mathbf{Z}^{(1)}$, $\mathbf{Z}^{(2)}$, and $\mathbf{Z}^{(3)}$, are accepted before the condition can trigger; the first checked candidate is $\widehat{\mathbf{Z}}^{(4)}$ at $s = 3$. If Equation (6) is satisfied, the implementation terminates before assigning the candidate, and the most recently accepted iterate $\mathbf{Z}^{(s)}$ is passed to entropy gating. Otherwise, $\mathbf{Z}^{(s+1)}$ is set to $\widehat{\mathbf{Z}}^{(s+1)}$. If the condition is never satisfied, propagation ends after the configured maximum number of steps. The index of the final accepted iterate is denoted by T in Equation (11). This rule is intended as graph-level safety control for deep or uncertain propagation regimes. In shallow standard configurations, the energy monitor may not trigger, which means that no energy-collapse event was detected under that configuration. It should not be interpreted as a formal guarantee against all accuracy degradation. Figure 4 illustrates the corresponding energy-decay behavior across the benchmark datasets.

Operator-Level Interpretation of Semantic Injection

The hybrid graph can be interpreted as a risk-controlled perturbation of the original propagation operator. In $\mathbf{A}_{\text{hybrid}} = \mathbf{A}_{\text{topo}} + \beta\mathbf{M}_{\text{rewire}} + \mathbf{I}$, β controls how strongly complementary semantic routes enter the row-normalized operator, while α controls how much neighbor prediction mass is absorbed at each diffusion step. Under the assumption that injected semantic edges have higher label compatibility than random non-edges, prediction diffusion over the hybrid graph increases the contribution of feature-aligned neighborhoods to low-confidence nodes. This is a conditional interpretation rather than a guarantee of improvement: if semantic edges are unreliable, the smoothness-energy monitor limits its graph-level collapse risk, and the entropy gate restricts node-level acceptance. Thus,

SETTA should be read as risk-controlled operator perturbation for frozen predictions, not as unconstrained graph rewiring.

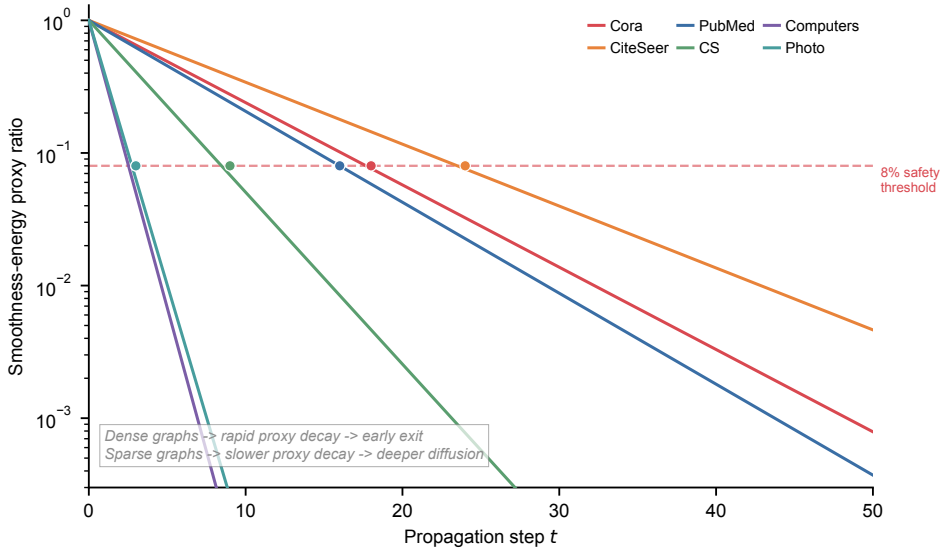


Figure 4. Row-normalized smoothness-energy proxy decay curves across six benchmark datasets. Densely connected graphs dissipate energy rapidly, whereas sparse citation graphs decay more gradually. The 8% threshold provides an adaptive stopping signal during deep propagation.

3.6. Entropy-Based Node-Level Selective Refinement

SETTA treats test-time adaptation as selective intervention rather than uniform smoothing. Low-confidence nodes may benefit from semantic diffusion, whereas high-confidence nodes should be protected from unnecessary perturbation. For node i, normalized entropy is computed from the original prediction:

$$H_{\text{norm}}[i] = \frac{-\sum_{c=1}^{C} \mathbf{Z}^{(0)}[i,c] \log(\mathbf{Z}^{(0)}[i,c] + \varepsilon)}{\log(C)}. \tag{7}$$

The scale-adaptive maximum confidence threshold is

$$\tau_{\max} = \max\big(0.85,\ 0.99 - 0.128 \max(0, \log_{10}(N) - 3.43)\big). \tag{8}$$

The node-level confidence threshold interpolates between $\tau_{\max}$ and

$$\tau_{\min} = \max(0.70, \tau_{\max} - 0.20) \tag{9}$$

according to node entropy, as follows:

$$\tau_v[i] = \tau_{\max} - (\tau_{\max} - \tau_{\min}) H_{\text{norm}}[i]. \tag{10}$$

The constants in Equations (8) and (9) were set during method design as fixed anchors for a monotone threshold schedule; they were not produced by a theoretical derivation, learned fit, or dataset-specific test-label search. In Equation (8), 0.99 defines a near-unity upper threshold for smaller graphs; 3.43 places the log-scale breakpoint at approximately 2.7×10^3 nodes, near the smallest graph scale considered; 0.128 determines the gradual decrease beyond this breakpoint; and 0.85 lower-bounds $\tau_{\max}$. In Equation (9), 0.20 limits the entropy-dependent threshold span, while 0.70 prevents the node-level threshold from becoming excessively low. For a fixed original confidence, decreasing either $\tau_{\max}$ or $\tau_v[i]$ makes the acceptance condition in Equation (11) harder to satisfy. Consequently, the

schedule becomes more selective as graph scale or normalized entropy increases. These constants are applied uniformly across datasets and are not selected using test labels. They should be regarded as fixed design heuristics rather than universally optimal values, and their sensitivity remains a topic for future work.

A node accepts the refined prediction when its original maximum class probability is below the node-level threshold; otherwise, it retains the original prediction:

$$\mathbf{Z}_{\text{final}}[i] = \begin{cases} \mathbf{Z}^{(T)}[i], & \text{if } \max_c \mathbf{Z}^{(0)}[i,c] < \tau_v[i], \\ \mathbf{Z}^{(0)}[i], & \text{otherwise.} \end{cases} \tag{11}$$

Figures 5 and 6 visualize the scale-adaptive threshold and the resulting node-level gating boundary.

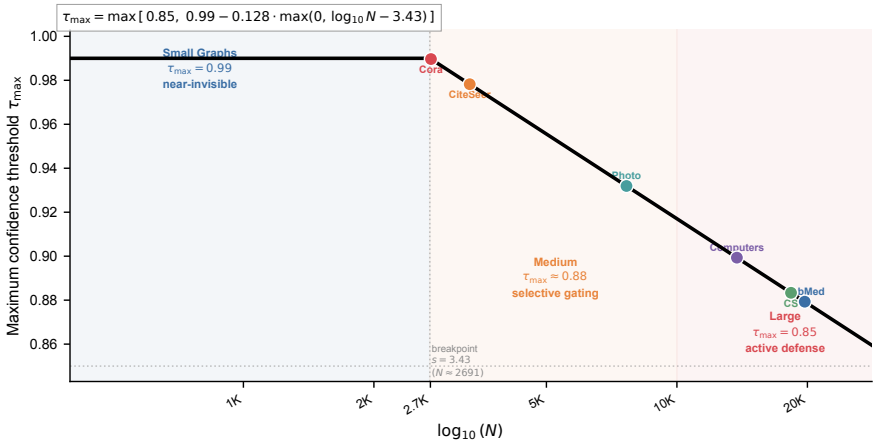


Figure 5. Scale-adaptive maximum confidence threshold. The maximum threshold decreases with graph size until reaching a floor of 0.85, making refinement acceptance more selective on larger graphs.

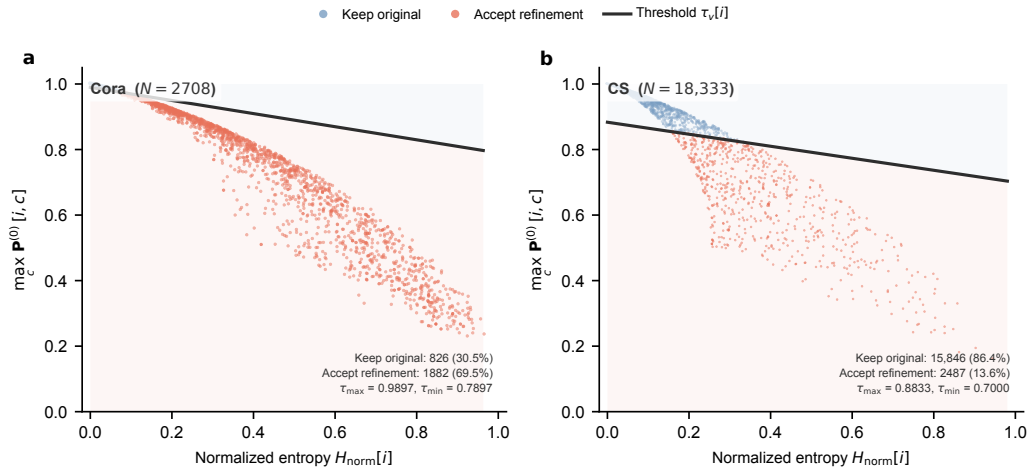


Figure 6. Per-node entropy-gating decision boundaries: (**a**) Cora, with refinement acceptance dominant; (**b**) Coauthor-CS, with retention of the original prediction dominant. In both panels, nodes below the threshold curve accept the refined prediction, whereas nodes above it retain the original GCN prediction.

This gate is a node-level risk-control heuristic inspired by selective prediction and confidence-aware decision control [37,38]. It protects high-confidence predictions from unnecessary propagation while allowing low-confidence predictions to benefit from re-

finement. Because confidence and correctness can diverge, the gate cannot certify that a retained or refined prediction is correct.

3.7. Computational Profile and Assumptions

SETTA has two cost components: one-time preprocessing for semantic graph construction and dense adjacency preparation, and per-run refinement on frozen predictions. In the current implementation, dense similarity and adjacency operations make the method appropriate for the benchmark scale evaluated here. Scaling SETTA to substantially larger graphs will require approximate nearest-neighbor search, sparse graph storage, and sparse propagation, for example through HNSW-style indexing or GPU similarity-search libraries [44,45].

The main modeling assumption is that feature similarity is informative for label-compatible propagation. This assumption is reasonable for many homophilic citation, co-purchase, and coauthor networks, but it may fail on heterophilic graphs where useful neighborhoods may not be homophilous [46]. In that case, semantic injection can connect cross-class nodes even when their features are close. Neither the global energy monitor nor the entropy gate directly verifies the label compatibility of an injected edge, so the current risk controls cannot guarantee robustness under heterophily.

3.8. Validation-Based Configuration Selection

SETTA performs refinement without learnable adaptation parameters, but it is not configuration-free. In the main experiments, the semantic neighbor count K and propagation depth are fixed by a dataset-level protocol and reported in Table 2, whereas the propagation coefficient α and semantic-edge coefficient β are selected using validation data only. K and propagation depth are fixed before α–β validation selection, are not optimized against the test split, and are examined through ablation and sensitivity analyses. The test split is never used for configuration selection.

Table 2. SETTA configuration per dataset. K, propagation depth, and similarity metric are fixed by the dataset-level protocol. α and β are selected using validation data only. For one-step refinement datasets, β is fixed to 1.0 and α is selected from the active shallow-refinement range $\alpha > 0.3$. The test split is used only for final evaluation.

Dataset	α	β	K	Steps	Metric	d_{svd}	$\tau_{\max}$
Cora	0.9	0.1	18	10	Jaccard	–	0.990
CiteSeer	0.6	0.1	12	10	Jaccard	–	0.978
PubMed	0.5	0.2	5	10	Cosine	128	0.879
CS	0.1	1.0	20	10	Cosine	850	0.883
Computers	0.4	1.0	25	1	Cosine	128	0.899
Photo	0.6	1.0	30	1	Cosine	128	0.932

For multi-step refinement datasets, α is selected from $\{0.1, 0.2, \ldots, 0.9\}$ and β from $\{0.0, 0.1, \ldots, 1.0\}$. For one-step refinement datasets, we use a constrained shallow-refinement search space: β is fixed to 1.0 and α is selected from $\{0.4, 0.5, \ldots, 0.9\}$. This constraint avoids near-identity propagation in the one-step setting, where semantic-route injection has only one opportunity to affect prediction refinement. This constraint is defined by propagation depth rather than by dataset identity or test-set performance.

The α–β grids reported in the sensitivity analysis are used only to characterize configuration sensitivity and are not used as a test-set selection protocol.

4. Results

4.1. Experimental Setup

Experiments were conducted on six node-classification datasets: Cora, CiteSeer, PubMed, Coauthor-CS, Amazon-Computers, and Amazon-Photo [41,42,47]. Cora, CiteSeer, and PubMed use fixed public splits. For Coauthor-CS and Amazon datasets, we use RandomNodeSplit with a fixed split seed, and all models are trained over seeds 42–51; means and standard deviations are reported over these 10 runs.

The trained-model baselines are MLP, GCN, GraphSAGE, GIN, GAT, GraphTransformer, and GradGateGNN [1,2,6,19,48,49]. We additionally compare APPNP and Correct and Smooth as decoupled or post-hoc baselines [15,18]. GraphTransformer is a two-layer TransformerConv baseline implemented with PyTorch 2.11.0 and PyTorch Geometric 2.7.0. SETTA is applied to frozen predictions from a two-layer GCN. During SETTA adaptation, no test labels, gradients, or parameter updates are used.

For SETTA configuration, K and propagation depth are pre-specified structural settings fixed before validation selection, based on feature type, graph scale, and whether the dataset uses multi-step or one-step refinement (Table 2). The fixed K and propagation-depth values are treated as part of this structural protocol rather than as validation- or test-selected hyperparameters. The propagation coefficient α and semantic-edge coefficient β are selected using the validation split only. The test split is evaluated only after the configuration has been fixed and is never used for selecting α or β. The full α–β heatmaps are sensitivity analyses, not deployment-time or test-label-based selection protocols.

Two additional analyses evaluate architectural transfer and statistical reliability. For backbone transfer, the same dataset-level SETTA refinement protocol is applied after training to frozen predictions from five backbones: GCN, GraphSAGE, GAT, GraphTransformer, and GradGateGNN. No backbone-specific SETTA tuning is performed. We report frozen accuracy, SETTA accuracy, accuracy change, and prediction-change rate. For paired significance testing, frozen GCN predictions and SETTA-refined predictions are compared over the same 10 random seeds using a two-sided paired t-test, with a Wilcoxon signed-rank test as a nonparametric robustness check. Test labels are used only for final evaluation and statistical analysis, not for adaptation or parameter selection.

As a reproducibility note, different experimental tables are generated from their corresponding reruns under the same configuration protocol and the same seed list. For Coauthor-CS and Amazon datasets, the data split is fixed before repeated training runs. Small numerical differences can still occur because model initialization, stochastic training, early stopping, and backend operations are not perfectly identical across independent executions. This effect is most visible on Amazon-Computers and Amazon-Photo, where the absolute SETTA gains are small.

4.2. Overall Performance on Frozen GNN Refinement

The first hypothesis is that frozen GCN predictions can be improved by test-time refinement without learnable adaptation parameters. On sparse citation networks, SETTA improved over the GCN backbone and all evaluated baselines on Cora, CiteSeer, and PubMed (Figure 7; Table 3).

Using validation-selected configurations, SETTA improves over the frozen GCN backbone on all three citation networks. The gains are largest on Cora, CiteSeer, and PubMed, with improvements of 4.61, 3.08, and 2.01 percentage points, respectively. The semantic-injection ablation indicates that these gains mainly reflect topology augmentation rather than aggressive prediction replacement, consistent with sparse citation links leaving many feature-aligned relationships unconnected.

On the larger benchmark graphs, SETTA remained useful beyond the small citation-network setting. It achieved the highest mean accuracy among the evaluated methods on Coauthor-CS and Amazon-Computers. On Amazon-Photo, GraphTransformer achieved the highest mean accuracy, while SETTA still produced a smaller but positive gain over the GCN backbone (Figure 8; Table 4).

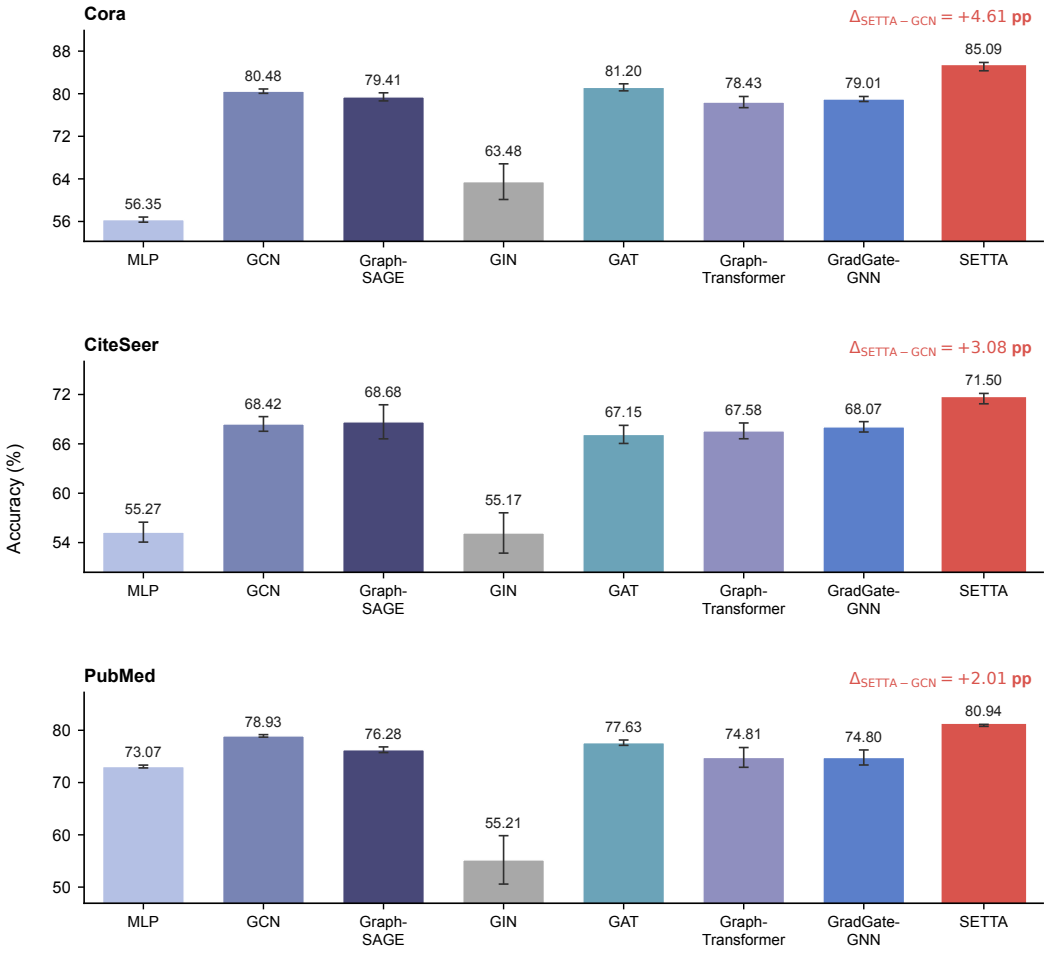


Figure 7. Main results on citation networks under validation-selected SETTA configurations. SETTA achieves the highest mean accuracy among the evaluated methods on Cora, CiteSeer, and PubMed.

Table 3. Node classification accuracy (%) on citation networks. Mean ± standard deviation over 10 random seeds. Best result is represented in bold.

Method	Cora	CiteSeer	PubMed
MLP	56.35 ± 0.48	55.27 ± 1.21	73.07 ± 0.27
GCN	80.48 ± 0.41	68.42 ± 0.89	78.93 ± 0.24
GraphSAGE	79.41 ± 0.76	68.68 ± 2.07	76.28 ± 0.54
GIN	63.48 ± 3.35	55.17 ± 2.45	55.21 ± 4.63
GAT	81.20 ± 0.66	67.15 ± 1.10	77.63 ± 0.51
GraphTransformer	78.43 ± 1.05	67.58 ± 0.96	74.81 ± 1.89
GradGateGNN	79.01 ± 0.47	68.07 ± 0.63	74.80 ± 1.44
SETTA (Ours)	**85.09 ± 0.78**	**71.50 ± 0.63**	**80.94 ± 0.19**
Δ_{GCN}	+4.61	+3.08	+2.01

Using validation-selected configurations, SETTA improves Coauthor-CS by 1.37 percentage points, whereas the gains on Amazon-Computers (+0.23 points) and Amazon-Photo (+0.46 points) are marginal. These effect sizes are substantially smaller than those on the

citation datasets. They may reflect stronger frozen baselines, ceiling effects, and the low prediction-change rates observed on the co-purchase graphs, but the present experiments do not isolate these explanations. The product-graph results should therefore be interpreted as modest benchmark-specific improvements rather than as evidence of large gains on dense or larger graphs in general.

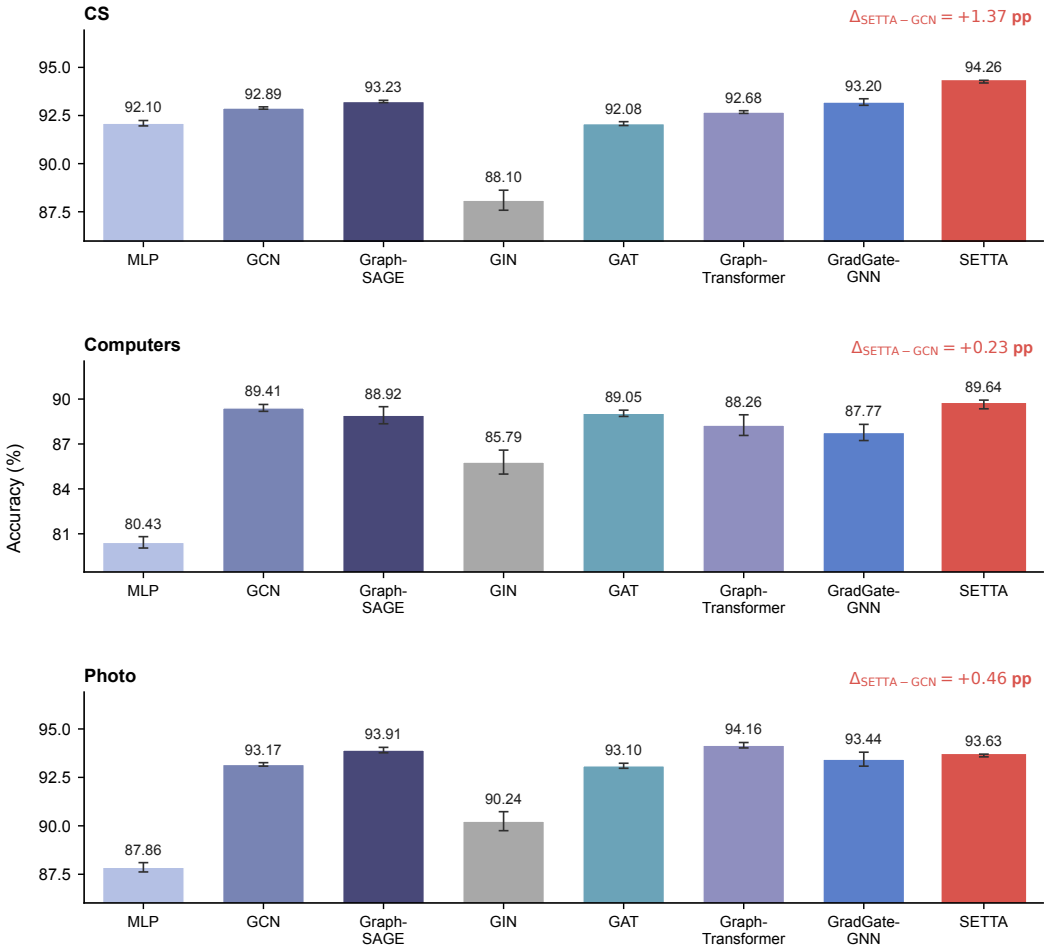


Figure 8. Results on larger benchmark graphs under validation-selected SETTA configurations. SETTA ranks first among evaluated methods on Coauthor-CS and Amazon-Computers, while GraphTransformer obtains the highest mean accuracy on Amazon-Photo.

Table 4. Node classification accuracy (%) on larger benchmark datasets. Mean ± standard deviation over 10 random seeds. Best result is represented in bold.

Method	CS	Computers	Photo
MLP	92.10 ± 0.14	80.43 ± 0.38	87.86 ± 0.24
GCN	92.89 ± 0.06	89.41 ± 0.23	93.17 ± 0.09
GraphSAGE	93.23 ± 0.06	88.92 ± 0.57	93.91 ± 0.14
GIN	88.10 ± 0.52	85.79 ± 0.80	90.24 ± 0.49
GAT	92.08 ± 0.10	89.05 ± 0.21	93.10 ± 0.13
GraphTransformer	92.68 ± 0.07	88.26 ± 0.69	94.16 ± 0.14
GradGateGNN	93.20 ± 0.17	87.77 ± 0.54	93.44 ± 0.36
SETTA (Ours)	**94.26 ± 0.07**	**89.64 ± 0.29**	93.63 ± 0.07
Δ_{GCN}	+1.37	+0.23	+0.46

4.3. Backbone-Transfer Analysis Across Frozen GNN Predictors

The main benchmark uses a frozen two-layer GCN as the prediction source. To examine whether SETTA is tied to a specific GNN architecture, we apply the same refinement protocol to frozen predictions from GCN, GraphSAGE, GAT, GraphTransformer, and GradGateGNN. Without backbone-specific tuning, SETTA produces positive mean accuracy changes in all 30 dataset–backbone settings (Figure 9; Table 5). For all non-GCN backbones, we reuse the dataset-level SETTA configurations in Table 2 without reselecting α or β for each backbone. The gains are largest on sparse citation datasets and remain positive on larger coauthor and co-purchase graphs. On Amazon-Computers, however, some backbones show only very small gains. These frozen-versus-refined comparisons show that SETTA acts as post-hoc refinement of each backbone rather than as a replacement for the backbone predictor.

Table 5. Backbone-transfer results after applying SETTA to frozen predictions. Frozen and SETTA accuracies are mean ± standard deviation over 10 seeds. Δ denotes SETTA-refined accuracy minus frozen-backbone accuracy in percentage points, and Changed denotes the test-set prediction-change rate.

Dataset	Backbone	Frozen Acc.	SETTA Acc.	Δ Acc.	Changed
Cora	GCN	80.48 ± 0.41	85.09 ± 0.78	+4.61	12.82%
	GraphSAGE	79.28 ± 0.94	83.79 ± 1.27	+4.51	14.96%
	GAT	81.20 ± 0.66	85.00 ± 0.67	+3.80	13.06%
	GraphTransformer	78.54 ± 0.59	84.24 ± 0.38	+5.70	14.24%
	GradGateGNN	79.14 ± 0.81	84.39 ± 0.54	+5.25	13.58%
CiteSeer	GCN	68.42 ± 0.89	71.50 ± 0.63	+3.08	16.92%
	GraphSAGE	69.21 ± 1.27	72.15 ± 0.41	+2.94	17.09%
	GAT	67.15 ± 1.10	71.10 ± 0.80	+3.95	17.32%
	GraphTransformer	67.79 ± 1.14	72.08 ± 0.76	+4.29	18.47%
	GradGateGNN	68.66 ± 0.91	72.53 ± 0.76	+3.87	16.88%
PubMed	GCN	78.93 ± 0.24	80.94 ± 0.19	+2.01	8.57%
	GraphSAGE	76.16 ± 0.51	78.38 ± 0.96	+2.22	10.30%
	GAT	77.63 ± 0.51	78.87 ± 0.78	+1.24	9.87%
	GraphTransformer	73.89 ± 1.20	76.45 ± 1.73	+2.56	9.54%
	GradGateGNN	75.83 ± 1.36	78.27 ± 1.04	+2.44	8.61%
CS	GCN	92.89 ± 0.06	94.26 ± 0.07	+1.37	3.63%
	GraphSAGE	93.24 ± 0.07	94.10 ± 0.06	+0.87	2.48%
	GAT	92.08 ± 0.10	93.67 ± 0.09	+1.58	3.86%
	GraphTransformer	92.69 ± 0.11	93.50 ± 0.09	+0.81	2.40%
	GradGateGNN	93.22 ± 0.16	93.58 ± 0.14	+0.37	1.97%
Computers	GCN	89.42 ± 0.18	89.62 ± 0.21	+0.20	1.87%
	GraphSAGE	88.79 ± 0.31	88.92 ± 0.41	+0.13	2.03%
	GAT	89.05 ± 0.19	89.13 ± 0.27	+0.08	1.57%
	GraphTransformer	88.78 ± 0.41	89.42 ± 0.26	+0.65	1.59%
	GradGateGNN	88.07 ± 0.51	88.55 ± 0.50	+0.48	1.67%
Photo	GCN	93.17 ± 0.09	93.63 ± 0.09	+0.46	2.56%
	GraphSAGE	94.04 ± 0.20	94.36 ± 0.16	+0.32	2.15%
	GAT	93.13 ± 0.13	93.62 ± 0.23	+0.49	2.22%
	GraphTransformer	94.14 ± 0.14	94.48 ± 0.08	+0.34	1.73%
	GradGateGNN	93.48 ± 0.19	93.93 ± 0.17	+0.45	1.85%

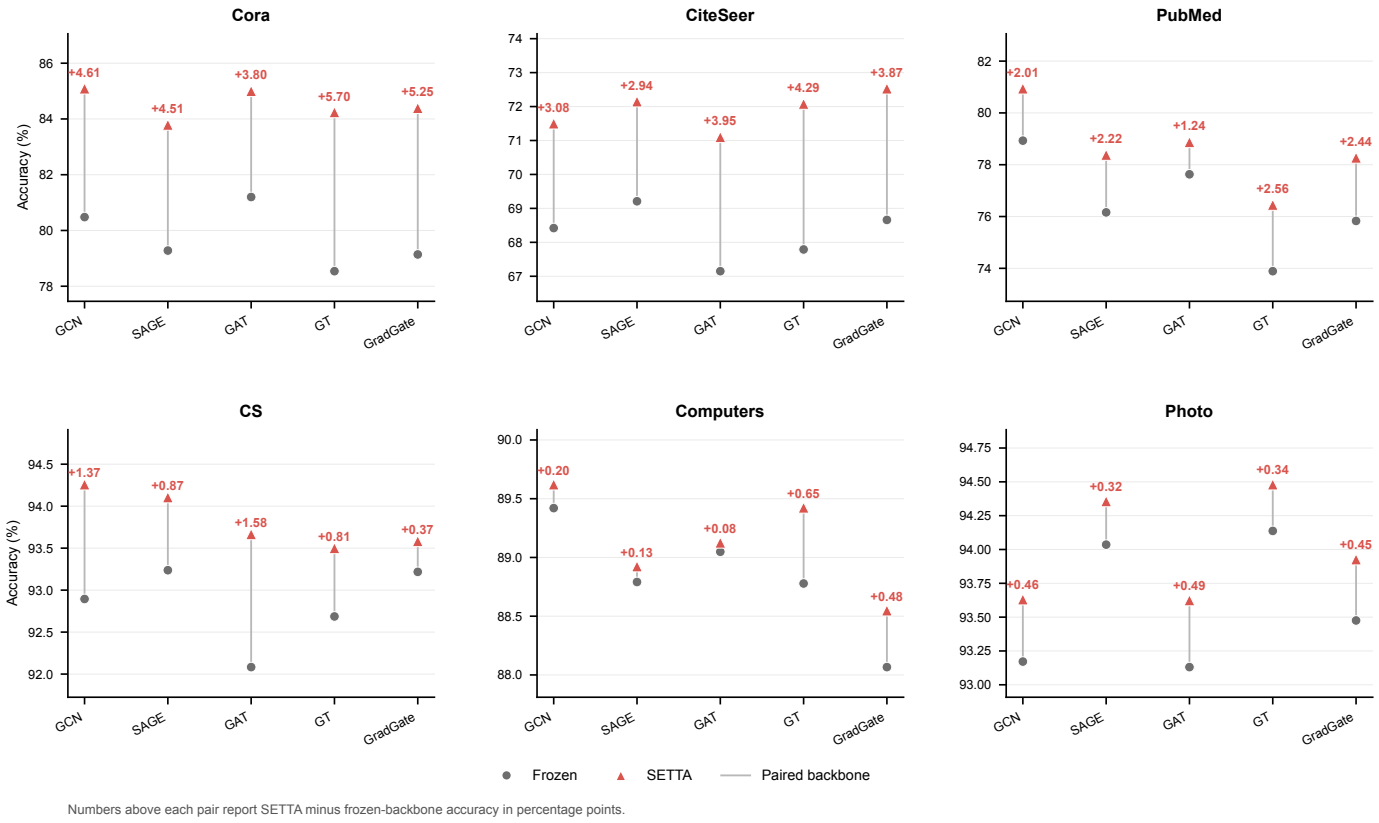


Figure 9. Backbone-agnostic SETTA refinement on frozen predictions. Each panel compares frozen and SETTA-refined accuracy within one dataset across five frozen GNN backbones. The same dataset-level refinement protocol is used across all backbones without backbone-specific tuning. Numbers above paired points report SETTA accuracy minus frozen-backbone accuracy in percentage points.

4.4. Paired Significance Analysis for the Primary GCN-Backed Setting

We further perform paired significance tests for the primary GCN-backed setting over the same 10 random seeds, comparing frozen GCN predictions with SETTA-refined predictions (Table 6). SETTA yields statistically significant improvements on all six datasets under a two-sided paired t-test, and the Wilcoxon signed-rank test leads to the same conclusion. The conclusions remain unchanged under a simple Bonferroni correction over the six GCN-backed dataset-level comparisons. Although the gain on Amazon-Computers is numerically small, its confidence interval remains positive, suggesting a marginal but statistically reliable improvement. The Amazon-Photo gain is also modest but consistent across seeds.

Table 6. Paired significance tests comparing frozen GCN predictions with SETTA-refined predictions over the same 10 random seeds. Δ denotes SETTA accuracy minus frozen-GCN accuracy in percentage points, reported as the mean $\pm$ standard deviation of paired seed-level differences.

Dataset	Δ Acc.	95% CI	Paired t-Test p	Wilcoxon p
Cora	$+4.61 \pm 0.63$	[+4.16, +5.06]	2.37×10^{-9}	0.0020
CiteSeer	$+3.08 \pm 0.92$	[+2.42, +3.74]	2.29×10^{-6}	0.0020
PubMed	$+2.01 \pm 0.39$	[+1.73, +2.29]	5.84×10^{-8}	0.0020
CS	$+1.37 \pm 0.05$	[+1.33, +1.40]	2.99×10^{-14}	0.0020
Computers	$+0.20 \pm 0.14$	[+0.10, +0.30]	0.0013	0.0059
Photo	$+0.46 \pm 0.10$	[+0.39, +0.53]	1.74×10^{-7}	0.0020

4.5. Does Semantic Injection Repair Missing Propagation Routes?

We next examine whether feature-aligned neighbors absent from the observed topology can provide useful propagation routes for frozen prediction refinement. Semantic edge injection improved over topology-only propagation on all six datasets (Figure 10; Table 7). The gains were largest on Cora and CiteSeer, moderate on PubMed and Coauthor-CS, and smaller on Amazon-Computers and Amazon-Photo.

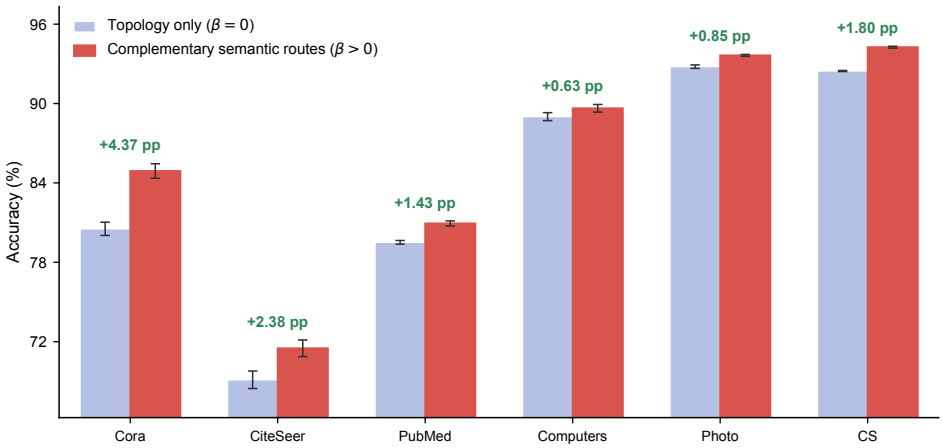


Figure 10. Semantic edge-injection ablation. Complementary semantic route injection improves over topology-only propagation across all evaluated datasets.

Table 7. Semantic edge-injection ablation. Mean accuracy (%) over 10 seeds. Δ denotes semantic injection minus topology-only diffusion. Values are taken from independent ablation reruns and may differ slightly from the main benchmark table due to rerunning stochastic training.

Dataset	β	Topology-Only	Semantic Injection	Δ
Cora	0.1	80.53 ± 0.50	84.90 ± 0.55	+4.37
CiteSeer	0.1	69.12 ± 0.66	71.50 ± 0.63	+2.38
PubMed	0.2	79.51 ± 0.14	80.94 ± 0.19	+1.43
Computers	1.0	89.01 ± 0.30	89.64 ± 0.29	+0.63
Photo	1.0	92.79 ± 0.13	93.64 ± 0.07	+0.85
CS	1.0	92.46 ± 0.04	94.26 ± 0.07	+1.80

This pattern supports the missing-route interpretation: citation graphs benefit most from complementary semantic edges, whereas denser product graphs show smaller but still positive gains. For continuous-feature datasets, Table 8 further shows that SVD-based feature denoising provides modest but positive gains before semantic-neighbor construction.

Table 8. SVD feature denoising ablation. Mean accuracy (%) over 10 seeds. Δ denotes SVD-enhanced minus raw features. Values are taken from independent ablation reruns and may differ slightly from the main benchmark table due to rerunning stochastic training.

Dataset	D	Raw Features	SVD-Enhanced	Δ
PubMed	500	80.54 ± 0.36	80.94 ± 0.19	+0.40
Computers	767	89.44 ± 0.31	89.64 ± 0.29	+0.20
Photo	745	93.25 ± 0.11	93.64 ± 0.07	+0.39
CS	6805	94.24 ± 0.05	94.26 ± 0.07	+0.02

4.6. Does Risk Control Prevent Harmful Propagation?

We then test whether stronger semantic diffusion must be risk-controlled. In this stress test, blind diffusion degraded as the semantic neighbor count increased, especially

on Coauthor-CS (Figure 11; Table 9). Energy monitoring reduced part of this degradation, whereas entropy gating and the full SETTA pipeline preserved accuracy more effectively.

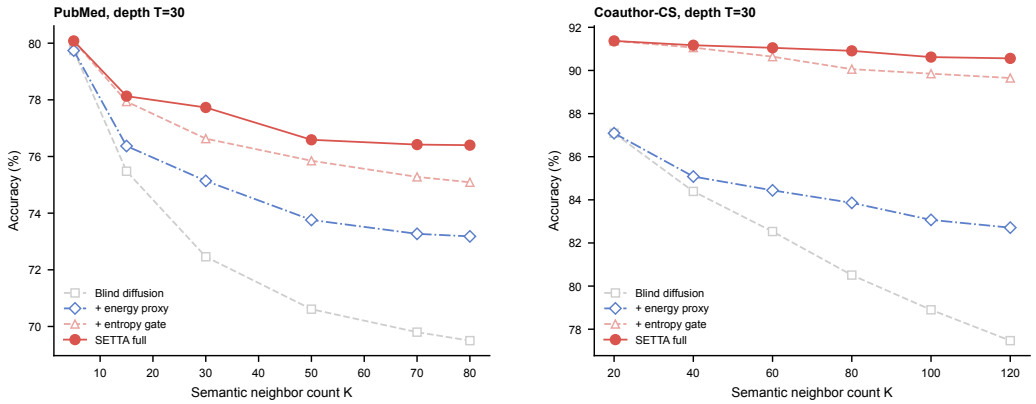


Figure 11. Dual-defense ablation under deep propagation. Points show the mean accuracies reported in Table 9. Blind diffusion degrades as the semantic neighbor count increases, while entropy gating and the full SETTA pipeline preserve accuracy more effectively.

Table 9. Dual-defense ablation on PubMed and Coauthor-CS ($T = 30$), using the 8% energy stopping rule in Equation (6). Mean accuracy (%) over 10 seeds.

Dataset	K	Blind	+Energy	+Gating	Full	Full − Blind
PubMed	5	79.74	79.74	80.08	80.08	+0.34
	15	75.48	76.37	77.94	78.13	+2.65
	30	72.46	75.14	76.63	77.73	+5.27
	50	70.61	73.76	75.85	76.59	+5.98
	70	69.80	73.27	75.28	76.42	+6.62
	80	69.50	73.18	75.09	76.40	+6.90
CS	20	87.09	87.09	91.37	91.37	+4.28
	40	84.39	85.08	91.06	91.17	+6.78
	60	82.53	84.44	90.64	91.05	+8.52
	80	80.51	83.86	90.06	90.91	+10.40
	100	78.90	83.07	89.85	90.62	+11.72
	120	77.47	82.71	89.65	90.56	+13.09

Entropy gating is the dominant node-level safeguard in these deep-propagation stress tests, with smoothness-energy monitoring acting as a graph-level safety mechanism.

4.7. Node-Level Transition Analysis of Entropy-Gated Refinement

To examine whether SETTA corrects frozen-GCN errors rather than perturbing correct predictions, we conduct a node-level transition analysis on the test split (Figure 12; Table 10). For each seed, test nodes are grouped as Wrong→Correct, Correct→Wrong, Wrong→Wrong, or Correct→Correct. The net correction rate is Wrong→Correct minus Correct→Wrong. A positive value means that SETTA corrects more frozen-GCN errors than it introduces, but it remains empirical evidence for selective risk control rather than a guarantee of node-level correctness.

Across all six datasets, Wrong→Correct exceeds Correct→Wrong. The effect is strongest on citation networks, consistent with their larger overall gains. On Coauthor-CS, only 14.72% of test nodes accept refinement, but accepted nodes improve by 9.27 percentage points on average, indicating selective intervention. On Amazon-Computers, the net correction rate is modest but positive, suggesting that selective refinement is most useful when the frozen backbone leaves semantically recoverable errors.

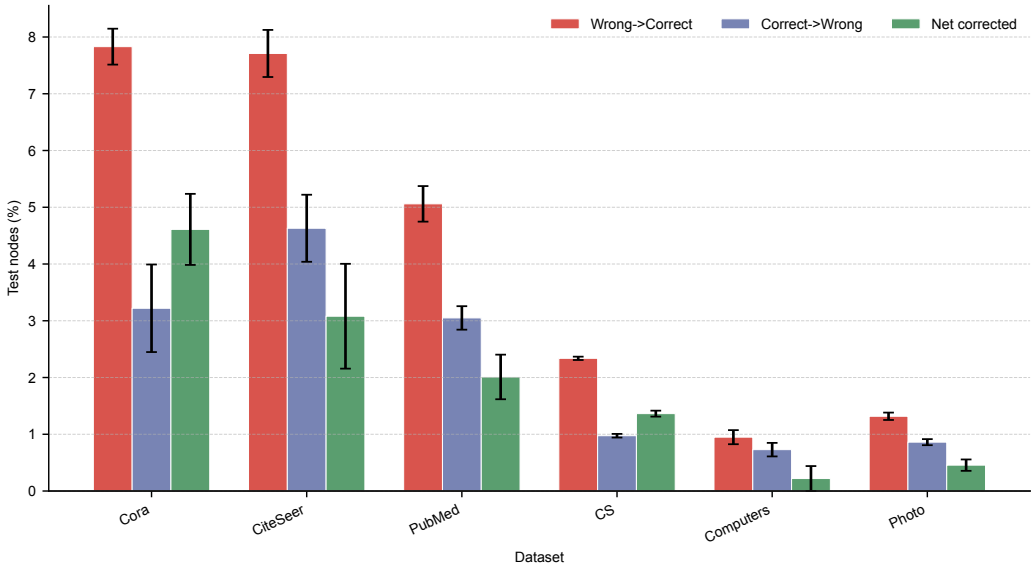


Figure 12. Node-level transition rates before and after SETTA refinement. Wrong→Correct denotes test nodes incorrectly classified by the frozen GCN but correctly classified after SETTA. Correct→Wrong denotes originally correct predictions changed to incorrect ones. Net correction is computed as Wrong→Correct minus Correct→Wrong. Bars show means over 10 random seeds, and error bars indicate ±1 sample standard deviation; they are not confidence intervals. SETTA produces positive net correction on all six datasets.

Table 10. Node-level transition analysis of SETTA refinement on the test split. Values are mean ± standard deviation over 10 seeds. Values are computed from independent transition-analysis runs and may differ slightly from the main benchmark table because they come from an independent stochastic training rerun, a known source of variation in GNN evaluation [41]. Accepted denotes the percentage of test nodes that accepted the refined prediction under entropy gating; Accepted ΔAcc reports the accuracy gain among accepted nodes.

Dataset	GCN Acc. (%)	SETTA Acc. (%)	Δ Acc. (%)	Accepted (%)	Accepted ΔAcc (%)	W→C (%)	C→W (%)	Net (%)
Cora	80.48 ± 0.43	85.09 ± 0.82	4.61 ± 0.63	73.62 ± 4.92	6.33 ± 1.14	7.83 ± 0.32	3.22 ± 0.77	4.61 ± 0.63
CiteSeer	68.42 ± 0.94	71.50 ± 0.67	3.08 ± 0.92	89.08 ± 4.71	3.47 ± 1.06	7.71 ± 0.41	4.63 ± 0.59	3.08 ± 0.92
PubMed	78.93 ± 0.25	80.94 ± 0.20	2.01 ± 0.39	47.54 ± 1.36	4.23 ± 0.83	5.06 ± 0.31	3.05 ± 0.21	2.01 ± 0.39
CS	92.89 ± 0.06	94.26 ± 0.07	1.37 ± 0.05	14.72 ± 0.42	9.27 ± 0.47	2.34 ± 0.03	0.97 ± 0.03	1.37 ± 0.05
Computers	89.44 ± 0.22	89.66 ± 0.35	0.22 ± 0.22	29.90 ± 1.57	0.75 ± 0.73	0.95 ± 0.12	0.73 ± 0.12	0.22 ± 0.22
Photo	93.18 ± 0.08	93.63 ± 0.09	0.46 ± 0.10	20.45 ± 0.92	2.24 ± 0.53	1.32 ± 0.07	0.86 ± 0.05	0.46 ± 0.10

4.8. Efficiency, Configuration Sensitivity Under Validation-Based Selection, and Scope

SETTA was compared with APPNP and Correct and Smooth under a one-time preprocessing protocol. Accuracy and runtime comparisons are shown in Figures 13 and 14. Because the semantic graph depends only on node features, topology, and fixed configuration values, its construction cost is reported separately from per-run refinement.

In this post-hoc comparison, SETTA achieved the highest accuracy among the evaluated prediction-level refinement baselines on all six datasets. Its per-run runtime remained close to the GCN runtime after excluding one-time graph construction. This accounting is appropriate when the same graph is reused across multiple evaluations, but it is not cold-start end-to-end runtime. The current dense implementation therefore supports benchmark-scale graph mining with amortized per-run refinement, not production-scale deployment.

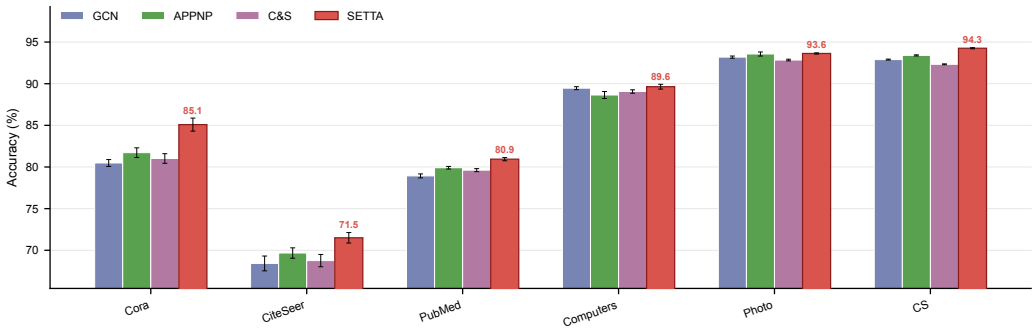


Figure 13. Accuracy comparison with decoupled propagation and post-hoc refinement baselines across six datasets. SETTA obtains the highest accuracy among the evaluated prediction-level refinement baselines in this comparison on all six datasets.

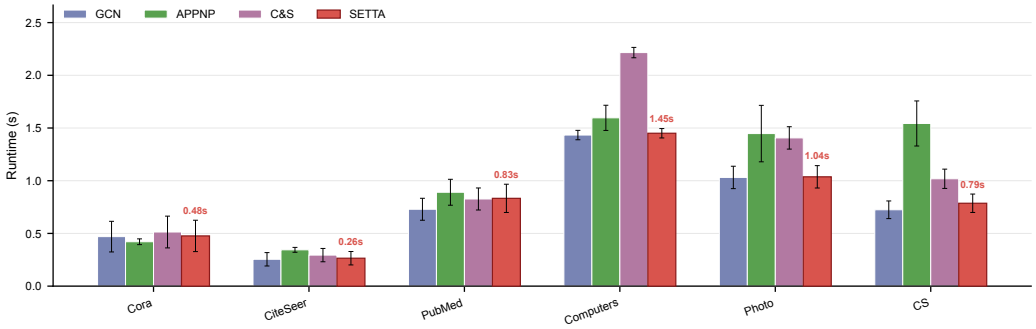


Figure 14. Runtime comparison under the one-time preprocessing protocol. Runtime excludes dataset-level semantic graph construction for SETTA, which is reported separately in Table 11.

Table 11. Accuracy (%) and runtime (seconds) for decoupled propagation and post-hoc refinement baselines. SETTA runtime is measured as shared GCN training plus refinement on frozen predictions; one-time graph construction is reported separately.

Method	Cora	CiteSeer	PubMed	Computers	Photo	CS
			Accuracy (%)			
GCN (Base)	80.48	68.42	78.93	89.46	93.19	92.89
APPNP	81.72	69.67	79.91	88.64	93.57	93.40
C&S	81.02	68.76	79.62	89.06	92.84	92.33
SETTA	**85.09**	**71.50**	**80.94**	**89.66**	**93.63**	**94.26**
			Runtime excluding one-time preprocessing (s)			
GCN (Base)	0.470	0.255	0.729	1.433	1.031	0.724
APPNP	0.421	0.344	0.890	1.597	1.447	1.543
C&S	0.513	0.294	0.827	2.216	1.406	1.018
SETTA	0.476	0.265	0.833	1.450	1.037	0.786
			One-time Graph Construction (s)			
SETTA	2.54	14.71	3.80	2.07	0.71	11.95

Note: Boldface indicates the highest accuracy in each dataset; italicized labels identify the three metric blocks.

Because SETTA is a test-time refinement method, configuration selection must not depend on test labels. We therefore report sensitivity under the validation-based protocol defined in Section 3.8: K and propagation depth are fixed by the dataset-level protocol, while α and β are selected using validation data only. Multi-step datasets use the full validation grid, whereas one-step datasets use the constrained shallow-refinement range. The heatmaps below characterize validation sensitivity; they are not test-set selection grids.

Figure 15 shows validation α–β sensitivity heatmaps for the multi-step datasets. The marked points denote the validation-selected configurations used for final test evaluation. High-performing validation regions are generally broad rather than isolated, indicating that SETTA is not dependent on a single fragile configuration.

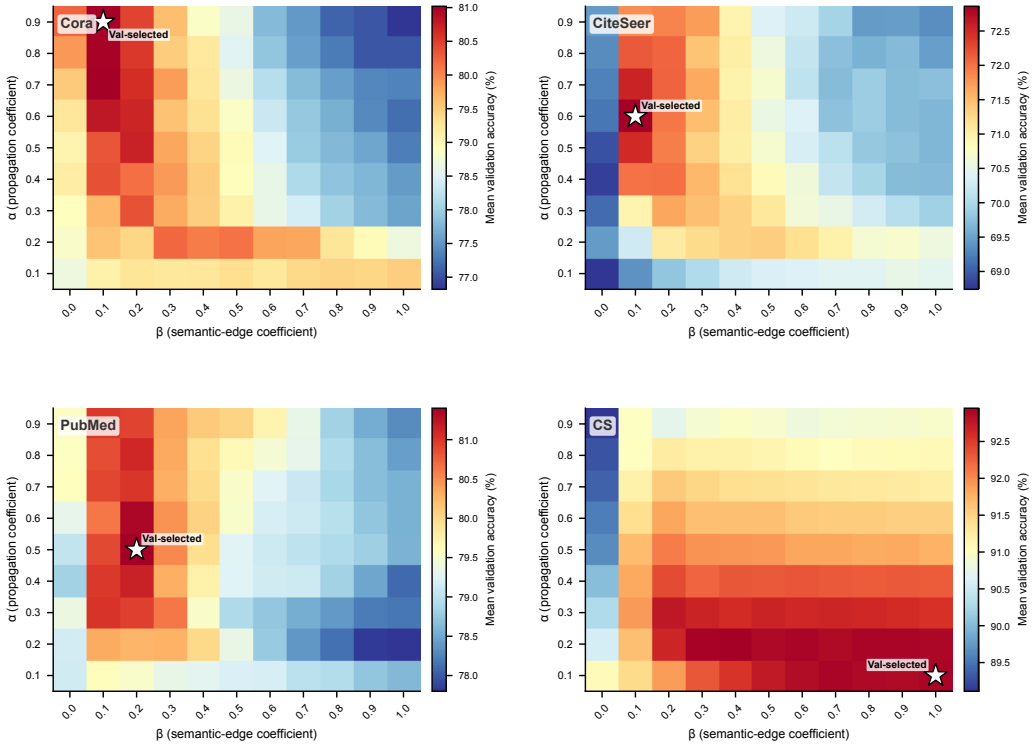


Figure 15. Validation-based α–β sensitivity heatmaps on multi-step refinement datasets. Colors indicate mean validation accuracy over 10 seeds. The marked points denote the validation-selected configuration used for final test evaluation. These heatmaps are reported to characterize configuration sensitivity and are not used as test-set selection grids.

Figure 16 shows the constrained one-step validation search for Amazon-Computers and Amazon-Photo. Because β is fixed to 1.0, the curves report mean validation accuracy as α varies in the active propagation range. The selected α values are fixed before test evaluation, and the figures are reported only as validation sensitivity evidence.

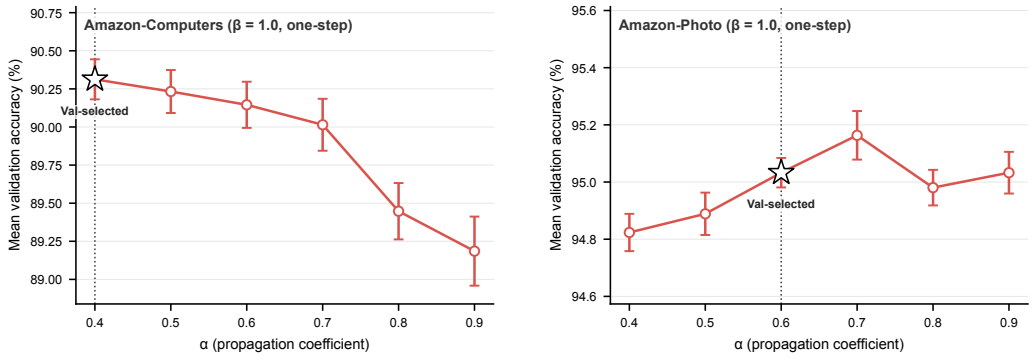


Figure 16. Validation-based sensitivity under the constrained one-step refinement protocol. For Amazon-Computers and Amazon-Photo, β is fixed to 1.0 and α is selected from the active shallow-refinement range. The star and vertical dotted line in each panel mark the validation-selected α used for final test evaluation.

5. Discussion

The results support the central deployment premise of this paper: anti-over-smoothing refinement can be partly separated from GNN training and performed over frozen predictions. From a graph data mining perspective, SETTA studies how frozen GNN predictors can refine unlabeled graph data without retraining. This setting is complementary to scalable GNN training algorithms: SETTA addresses robust post-deployment refinement of GNN-based mining outputs under frozen-model constraints. This section interprets the evidence through three design principles rather than repeating the experimental tables.

5.1. Semantic Routes as Missing Topology Compensation

The semantic-route experiments point to a broader deployment principle: the observed topology is not always a complete map of task-relevant similarity. When a backbone is frozen, missing propagation routes cannot be repaired by changing learned filters or retraining message-passing layers. A refinement method without learned adaptation parameters can instead expose frozen predictions to complementary semantic paths along which existing class evidence can travel.

This principle is most plausible when feature similarity and label compatibility are aligned, as in many homophilic citation, coauthor, or co-purchase graphs. It is less reliable when features are noisy, weakly related to labels, or heterophilic. For that reason, SETTA preserves observed edges and uses semantic construction only as complementary missing-topology compensation for frozen prediction refinement, rather than as graph structure learning before or during model training [23,24].

Backbone-transfer results suggest that SETTA acts primarily as a prediction-level semantic refinement method rather than a GCN-specific correction. It can refine frozen outputs from heterogeneous architectures under the same dataset-level protocol. Nevertheless, gain magnitude remains dataset- and backbone-dependent. On dense co-purchase graphs, where frozen predictors may already be strong and prediction-change rates are low, SETTA produces modest improvements rather than large gains.

5.2. Selective Adaptation as Risk Control

SETTA does not claim that more propagation is always better. Once semantic routes are added, the central question becomes when additional diffusion should be trusted. The method therefore treats test-time refinement as a controlled intervention: smoothness-energy monitoring limits graph-level collapse, and entropy gating limits node-level perturbation.

The resulting principle is selective adaptation rather than uniform smoothing. Uncertain nodes may benefit from semantic diffusion, whereas confident nodes should often remain unchanged. This is still heuristic risk control, because confidence can be miscalibrated and entropy is not a calibrated Bayesian uncertainty estimate [38]. The evidence therefore supports SETTA as a practical selective-refinement rule, not as a guarantee that every accepted refinement is correct.

5.3. Scope and Limitations

Several limitations constrain the interpretation. First, SETTA is training-free during refinement and introduces no learnable adaptation parameters, but it is not configuration-free. Here, K and propagation depth are fixed by a dataset-level protocol, while α and β are selected using validation data only. This avoids test-label-based configuration selection in the main evaluation, but validation labels may not always be available in fully unlabeled deployment scenarios. Future work should develop unsupervised configuration-selection criteria based on confidence, smoothness-energy behavior, calibration, or graph statistics.

Second, the current implementation relies on dense semantic graph construction and dense adjacency operations. The efficiency results therefore support benchmark-scale evaluation and amortized per-run refinement, not production-scale deployment on substantially larger graphs. Larger public benchmarks, including Open Graph Benchmark-scale settings, should be considered in future work together with approximate nearest-neighbor search, sparse storage, and sparse propagation [44,45,50].

Third, SETTA assumes that feature similarity is informative for label-compatible propagation. This assumption is plausible for the mostly homophilic citation, co-purchase, and coauthor benchmarks evaluated here, but may fail on heterophilic graphs, where useful neighborhoods can contain many cross-class edges [46]. The energy monitor detects excessive global smoothing but does not validate individual semantic edges, and the entropy gate controls whether a node changes without guaranteeing that the accepted label is correct. Robustness on heterophilic graphs is therefore outside the evidence established by the present experiments. Future extensions could estimate local homophily before injection and use reliability-weighted, signed, or heterophily-aware semantic routes.

6. Conclusions

SETTA shows that anti-over-smoothing refinement can be partially decoupled from GNN training and formulated as test-time adaptation over frozen predictions without learnable adaptation parameters during refinement. Using only frozen GNN outputs, node features, and graph topology during refinement, it constructs complementary semantic routes, propagates predictions under graph-level energy control, and applies node-level selective refinement without test labels, gradients, or parameter updates.

Across six mostly homophilic node-classification benchmarks, SETTA consistently improves the frozen two-layer GCN backbone under a validation-based configuration-selection protocol and achieves the highest mean accuracy among the evaluated methods on five datasets. The gains are largest on the sparse citation networks, moderate on Coauthor-CS, and marginal on Amazon-Computers and Amazon-Photo. The same refinement protocol also produces positive mean gains across five frozen GNN backbones, while paired tests over 10 seeds support the consistency of the primary GCN-backed comparisons. No test labels or test-set performance are used for configuration selection; all reported test analyses use configurations fixed in advance through the validation-based protocol. Ablations indicate that semantic routes can compensate for missing topology, while risk-controlled propagation helps prevent harmful over-smoothing under deep-propagation stress. Node-level transition analysis further shows more error-correcting than error-inducing prediction changes on all six datasets. The current results support benchmark-scale graph mining and amortized per-run refinement; larger-scale deployment and robustness on heterophilic graphs remain open issues.

Author Contributions: Conceptualization, D.Y.; methodology, D.Y.; software, D.Y.; validation, D.Y.; formal analysis, D.Y.; investigation, D.Y.; data curation, D.Y.; writing—original draft preparation, D.Y.; writing—review and editing, D.Y., X.C. and R.X.; visualization, D.Y.; supervision, X.C. and R.X.; project administration, X.C. and R.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The article processing charge was funded by D.Y.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available benchmark datasets, including Cora, CiteSeer, PubMed, Coauthor-CS, Amazon-Computers, and Amazon-Photo. The implementation code, configuration files, and experimental scripts are available at https://github.com/hubuydy/SETTA (accessed on 1 August 2026).

Acknowledgments: The authors acknowledge the use of artificial intelligence tools for language editing, manuscript polishing, and formatting assistance. The authors reviewed and edited all tool-generated output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

APPNP	Approximate Personalized Propagation of Neural Predictions
C&S	Correct and Smooth
CI	Confidence Interval
CS	Computer Science (Coauthor-CS Dataset)
GAT	Graph Attention Network
GCN	Graph Convolutional Network
GCNII	Graph Convolutional Network with Initial Residual and Identity Mapping
GDC	Graph Diffusion Convolution
GIN	Graph Isomorphism Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
GraphTTA	Graph Test-Time Adaptation
GSL	Graph Structure Learning
GT3	Graph Test-Time Training
GTrans	Test-Time Graph Transformation
HNSW	Hierarchical Navigable Small World
KNN	*K*-Nearest Neighbors
MLP	Multilayer Perceptron
pp	Percentage Points
SETTA	Spectral-Energy Test-Time Adaptation
SGC	Simplified Graph Convolution
SIGN	Scalable Inception Graph Neural Network
SVD	Singular Value Decomposition
TENT	Test Entropy Minimization
TTA	Test-Time Adaptation
TTT	Test-Time Training

References

1. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
2. Hamilton, W.L.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
3. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018*; Springer: Cham, Switzerland, 2018; pp. 593–607. [CrossRef]
4. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. LightGCN: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 639–648. [CrossRef]
5. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, NSW, Australia, 6–11 August 2017; pp. 1263–1272.
6. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? In Proceedings of the 7th International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.

7. Li, Q.; Han, Z.; Wu, X.M. Deeper insights into graph convolutional networks for semi-supervised learning. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018; pp. 3538–3545.
8. Oono, K.; Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. In Proceedings of the 8th International Conference on Learning Representations (ICLR), Virtual, 26–30 April 2020.
9. Cai, C.; Wang, Y. A note on over-smoothing for graph neural networks. In Proceedings of the 37th International Conference on Machine Learning (ICML), Virtual, 13–18 July 2020.
10. Rusch, T.K.; Bronstein, M.M.; Mishra, S. A survey on oversmoothing in graph neural networks. *arXiv* **2023**, arXiv:2303.10993.
11. Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; Li, Y. Simple and deep graph convolutional networks. In Proceedings of the 37th International Conference on Machine Learning (ICML), Virtual, 13–18 July 2020.
12. Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.; Jegelka, S. Representation learning on graphs with jumping knowledge networks. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 5453–5462.
13. Zhao, L.; Akoglu, L. PairNorm: Tackling oversmoothing in GNNs. In Proceedings of the 8th International Conference on Learning Representations (ICLR), Virtual, 26–30 April 2020.
14. Rong, Y.; Huang, W.; Xu, T.; Huang, J. DropEdge: Towards deep graph convolutional networks on node classification. In Proceedings of the 8th International Conference on Learning Representations (ICLR), Virtual, 26–30 April 2020.
15. Klicpera, J.; Bojchevski, A.; Günnemann, S. Predict then propagate: Graph neural networks meet personalized PageRank. In Proceedings of the 7th International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
16. Gasteiger, J.; Weißenberger, S.; Günnemann, S. Diffusion improves graph learning. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; pp. 13333–13345.
17. Wu, F.; Souza, A.H., Jr.; Zhang, T.; Fifty, C.; Yu, T.; Weinberger, K.Q. Simplifying graph convolutional networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6861–6871.
18. Huang, Q.; He, H.; Singh, A.; Lim, S.N.; Benson, A.R. Combining label propagation and simple models out-performs graph neural networks. In Proceedings of the 9th International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
19. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
20. Attali, H.; Buscaldi, D.; Pernelle, N. Rewiring techniques to mitigate oversquashing and oversmoothing in GNNs: A survey. *arXiv* **2024**, arXiv:2411.17429.
21. Zhang, Y.; Deidda, R.; Higham, D.J.; Tudisco, F. Rethinking oversmoothing in graph neural networks: A rank-based perspective. *arXiv* **2025**, arXiv:2502.04591.
22. Frasca, F.; Rossi, E.; Eynard, D.; Chamberlain, B.; Bronstein, M.; Monti, F. SIGN: Scalable inception graph neural networks. *arXiv* **2020**, arXiv:2004.11198.
23. Franceschi, L.; Niepert, M.; Pontil, M.; He, X. Learning discrete structures for graph neural networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 1972–1982.
24. Chen, Y.; Wu, L.; Zaki, M.J. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS), Virtual, 6–12 December 2020; pp. 19314–19326.
25. Topping, J.; Di Giovanni, F.; Chamberlain, B.P.; Dong, X.; Bronstein, M.M. Understanding over-squashing and bottlenecks on graphs via curvature. In Proceedings of the 10th International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
26. Karhadkar, K.; Banerjee, P.K.; Montúfar, G. FoSR: First-order spectral rewiring for addressing oversquashing in GNNs. In Proceedings of the 11th International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
27. Jamadandi, A.; Rubio-Madrigal, C.; Burkholz, R. Spectral graph pruning against over-squashing and over-smoothing. In Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 10–15 December 2024.
28. Linkerhägner, J.; Shi, C.; Dokmanić, I. Joint graph rewiring and feature denoising via spectral resonance. In Proceedings of the 13th International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
29. Rubio-Madrigal, C.; Jamadandi, A.; Burkholz, R. GNNs getting ComFy: Community and feature similarity guided rewiring. In Proceedings of the 13th International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
30. Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In Proceedings of the 9th International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
31. Chen, G.; Zhang, J.; Xiao, X.; Li, Y. GraphTTA: Test time adaptation on graph neural networks. *arXiv* **2022**, arXiv:2208.09126.
32. Jin, W.; Zhao, T.; Ding, J.; Liu, Y.; Tang, J.; Shah, N. Empowering graph representation learning with test-time graph transformation. In Proceedings of the 11th International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
33. Wang, Y.; Li, C.; Jin, W.; Li, R.; Zhao, J.; Tang, J.; Xie, X. Test-time training for graph neural networks. *arXiv* **2022**, arXiv:2210.08813.

34. Zhang, J.; Wang, Y.; Yang, X.; Zhu, E. A fully test-time training framework for semi-supervised node classification on out-of-distribution graphs. *ACM Trans. Knowl. Discov. Data* **2024**, *18*, 172. [CrossRef]
35. Bao, W.; Zeng, Z.; Liu, Z.; Tong, H.; He, J. Matcha: Mitigating graph structure shifts with test-time adaptation. In Proceedings of the 13th International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
36. Jiang, Y.; Qiu, R.; Huang, Z. Does homophily help in robust test-time node classification? *arXiv* **2025**, arXiv:2510.22289.
37. Geifman, Y.; El-Yaniv, R. Selective classification for deep neural networks. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 4885–4894.
38. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, NSW, Australia, 6–11 August 2017; pp. 1321–1330.
39. Zhang, Y.; Pal, S.; Coates, M.; Üstebay, D. Bayesian graph convolutional neural networks for semi-supervised classification. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI), Honolulu, HI, USA, 27 January–1 February 2019.
40. Halko, N.; Martinsson, P.G.; Tropp, J.A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **2011**, *53*, 217–288. [CrossRef]
41. Shchur, O.; Mumme, M.; Bojchevski, A.; Günnemann, S. Pitfalls of graph neural network evaluation. In Proceedings of the NeurIPS Relational Representation Learning Workshop, Montreal, QC, Canada, 8 December 2018 .
42. Yang, Z.; Cohen, W.; Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016.
43. Jaccard, P. Étude comparative de la distribution florale dans une portion des Alpes et du Jura. *Bull. Soc. Vaudoise Sci. Nat.* **1901**, *37*, 547–579.
44. Malkov, Y.A.; Yashunin, D.A. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 824–836. [CrossRef] [PubMed]
45. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* **2021**, *7*, 535–547. [CrossRef]
46. Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS), Virtual, 6–12 December 2020; pp. 7793–7804.
47. McAuley, J.; Targett, C.; Shi, Q.; van den Hengel, A. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference, Santiago, Chile, 9–13 August 2015.
48. Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; Sun, Y. Masked label prediction: Unified message passing model for semi-supervised classification. In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, 19–27 August 2021.
49. Rusch, T.K.; Chamberlain, B.P.; Mahoney, M.W.; Bronstein, M.M.; Mishra, S. Gradient gating for deep multi-rate learning on graphs. In Proceedings of the 11th International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
50. Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. Open Graph Benchmark: Datasets for machine learning on graphs. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS), Virtual, 6–12 December 2020; pp. 22118–22133.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.