ECHO: Encoding Communities via High-order Operators

EMILIO FERRARA, Thomas Lord Department of Computer Science, University of Southern California, USA

Community detection in attributed networks faces a fundamental divide: topological algorithms ignore semantic features, while Graph Neural Networks (GNNs) encounter devastating computational bottlenecks. Specifically, GNNs suffer from a "Semantic Wall" of feature over-smoothing in dense or heterophilic networks, and a "Systems Wall" driven by the $O(N^2)$ memory constraints of pairwise clustering. To dismantle these barriers, we introduce ECHO (Encoding Communities via High-order Operators), a scalable, self-supervised architecture that reframes community detection as an adaptive, multi-scale diffusion process. ECHO features a Topology-Aware Router that automatically analyzes structural heuristics—sparsity, density, and assortativity—to route graphs through the optimal inductive bias, preventing heterophilic poisoning while ensuring semantic densification. Coupled with a memory-sharded full-batch contrastive objective and a novel chunked $O(N \cdot K)$ similarity extraction method, ECHO completely bypasses traditional $O(N^2)$ memory bottlenecks without sacrificing the mathematical precision of global gradients. Extensive evaluations demonstrate that this topology-feature synergy consistently overcomes the classical resolution limit. On synthetic LFR benchmarks scaled up to 1 million nodes, ECHO achieves scale-invariant accuracy despite severe topological noise. Furthermore, on massive real-world social networks with over 1.6 million nodes and 30 million edges, it completes clustering in mere minutes with throughputs exceeding 2,800 nodes per second—matching the speed of highly optimized purely topological baselines. The implementation utilizes a unified framework that automatically engages memory-sharded optimization to support adoption across varying hardware constraints.
GitHub Repository: https://github.com/emilioferrara/ECHO-GNN

Additional Key Words and Phrases: Community Detection, Graph Neural Networks, Self-Supervised Learning, Network Science

1 Introduction

The detection of community structures in complex networks is a cornerstone of modern data science, with applications ranging from the identification of functional modules in biological systems to the mapping of coordinated inauthentic behavior on massive social platforms [7]. Traditionally, this task has been approached as a purely topological optimization problem. Algorithms such as Louvain [2] and its successor, Leiden [28], seek to maximize modularity [19] by identifying clusters with high internal edge density relative to a null model. While computationally efficient and mathematically robust, these methods are inherently "feature-blind." By ignoring the rich semantic attributes (e.g., text, user demographics, or product reviews) that often define node behavior, topological methods operate at a fixed structural scale and frequently suffer from the **resolution limit** [8], failing to distinguish meaningful sub-communities within dense network cores.

The rise of Graph Representation Learning and Graph Neural Networks (GNNs) [13] introduced a competing paradigm designed to fuse structural and semantic signals. By leveraging message-passing architectures [11], researchers can learn node embeddings that capture both localized topology and global feature distributions. However, deploying GNNs for community detection introduces a fundamental dual-dilemma: the *Semantic Wall* and the *Systems Wall.*

Semantically, this fusion falls victim to the "Over-smoothing Paradox" [17, 33]. In attributed networks, prolonged message-passing steps intended to capture high-order community signals frequently result in feature homogenization. Instead of sharpening community boundaries, the distinct semantic signatures of nodes are blurred into an indistinguishable "mush," particularly in dense or highly heterophilic graphs [21]. Compounding this issue is the rigid inductive bias of standard GNNs: forcing universal neighborhood aggregation across diverse graph morphologies triggers either "heterophilic poisoning" in noisy topologies or "semantic starvation" when dealing with sparse, high-dimensional

Author's Contact Information: Emilio Ferrara, Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA, USA, emiliofe@usc.edu.

features. Systematically, state-of-the-art self-supervised GNNs encounter a devastating computational bottleneck. The standard practice of computing full pairwise similarity matrices for downstream clustering introduces a hard $O(N^2)$ memory requirement, rendering modern contrastive models intractable for large-scale networks without prohibitive hardware investments.

To overcome these barriers, we introduce **ECHO** (Encoding Communities via High-order Operators). ECHO bridges the gap between scalable topological clustering and deep semantic representation by reframing community detection as an **adaptive, multi-scale diffusion process**. Recognizing that no single architecture suits all network types, ECHO evaluates structural heuristics to route graphs through the optimal semantic encoder. It then models community formation explicitly: by deploying high-order diffusion operators guided by edge-wise attention, ECHO actively diffuses semantic signals within structural clusters while dynamically throttling information flow across community borders. This yields a multi-resolution embedding space immune to classical resolution limits.

Contributions

ECHO is a highly scalable, self-supervised architecture designed to preserve feature integrity while aggressively mining topological signals. The core contributions are:

1. **Topology-Aware Semantic Routing:** We introduce an unsupervised gating mechanism that automatically analyzes feature sparsity, structural density, and semantic assortativity to route graphs through either a Densifying Encoder (GraphSAGE) or an Isolating Encoder (MLP), dynamically adapting the inductive bias to the network's manifold.
2. **Attention-Guided Multi-Scale Diffusion:** We introduce a novel conceptual lens for community detection that abandons isotropic mean-pooling. By evaluating semantic features through learned attention before topological mixing, ECHO dynamically prunes noisy or heterophilic edges, providing a multi-resolution view of the network that halts over-smoothing even in hyper-dense environments.
3. **Memory-Sharded Full-Batch Contrastive Learning:** We propose a refined InfoNCE-based objective that preserves the mathematical precision of full-batch gradients while bypassing the Systems Wall. By dynamically sharding the negative sampling tensor when memory limits are approached, ECHO enables exact community boundary definitions for massive networks without the noise associated with stochastic mini-batching.
4. **Sub-Quadratic Million-Scale Clustering:** We introduce a chunked $O(N \cdot K)$ GPU-accelerated similarity extraction method that bypasses the $O(N^2)$ memory limit entirely. ECHO clusters networks with millions of nodes and edges (e.g., social networks with $> 1.6M$ nodes) in mere minutes, achieving throughputs exceeding 2,800 nodes per second on a single commercial GPU.
5. **Unified Scalable Framework:** We provide a fully documented open-source implementation of a unified framework that automatically adapts to graph scale. The architecture handles both research-grade precision on mid-scale graphs and massive-scale production environments through its adaptive tensor-sharding optimization.

Through extensive evaluation on synthetic LFR benchmarks [14] scaled up to one million nodes, alongside diverse real-world attributed networks, we demonstrate that ECHO consistently overcomes the resolution limit. By matching the speed of highly optimized topological algorithms while retaining deep semantic understanding, ECHO provides a definitive solution for community detection in the era of massive, attributed data.

2 Related Work

2.1 Topological Community Detection and Higher-Order Structures

Classical community detection has historically relied heavily on the optimization of a quality function, most notably modularity [19]. The Louvain algorithm [2] introduced a greedy, hierarchical approach to modularity maximization, allowing for the rapid analysis of large-scale networks. However, standard modularity-based methods operate strictly on direct interactions (1-hop edges) and are famously constrained by the *resolution limit* [8], frequently failing to detect distinct sub-communities within dense network cores. The Leiden algorithm [28] later addressed structural instabilities in Louvain, but the fundamental constraint of operating at a fixed structural scale remained.

Beyond modularity, alternative topological paradigms have sought to capture complex network dynamics through different theoretical lenses. Information-theoretic approaches like Infomap [24] leverage random walks to minimize the description length of network flows. Similarly, statistical methods such as OSLOM [15] introduced robust frameworks for detecting overlapping and hierarchical communities. Further expanding the structural taxonomy, the concept of *link communities* [1] fundamentally shifted the partitioning focus from nodes to edges. However, despite these profound advancements in structural partitioning, these methods share a critical limitation: they are inherently "feature-blind" and cannot natively incorporate the rich semantic node attributes that drive homophily in modern networks.

To overcome purely structural constraints and the resolution limit, researchers pioneered methods that explicitly mix local and global topological information [6]. A foundational advancement extended community detection beyond direct neighbors by leveraging k-path edge centrality to generalize the Louvain method [5]. ECHO directly inherits and modernizes this philosophy: rather than relying on static k-path heuristics, ECHO employs learnable K-step neural diffusion and edge-wise attention to dynamically control multi-scale information flow.

2.2 Graph Representation Learning

The second epoch of community detection marked a paradigm shift from explicit discrete partitioning to learning continuous, low-dimensional vector representations of nodes, a landscape we comprehensively systematized [9]. By mapping nodes into a latent geometric space, these techniques enabled the direct application of classical machine learning algorithms to complex network topology.

Early foundational models approached this challenge through matrix factorization and random walk-based shallow embeddings. Methods such as DeepWalk [22], LINE [26], and Node2Vec [10] adapted the Skip-gram architecture from natural language processing to graph topology. By simulating biased or unbiased random walks, these algorithms implicitly explored the k-hop neighborhoods of nodes, successfully capturing higher-order structural proximities and homophily within the latent embedding space.

However, despite their success in downstream tasks, shallow embedding methods suffer from two critical limitations when applied to community detection. First, they are fundamentally feature-blind; they optimize representations solely over the adjacency matrix and cannot ingest rich semantic attributes. Second, they necessitate a disjointed, two-stage pipeline where a secondary clustering algorithm (typically K-Means) is applied to the resulting dense vectors. ECHO resolves this by moving to an end-to-end, attribute-aware paradigm.

2.3 Attributed Graph Clustering and GNNs

The third and current epoch of community detection addresses the limitations of feature-blind embeddings by utilizing Graph Neural Networks (GNNs) [13]. GNNs offer an elegant theoretical framework to simultaneously encode network topology and node attributes via iterative message-passing. Architectures such as GraphSAGE [11] and Graph Attention Networks (GAT) [29] successfully demonstrated the power of inductive neighborhood aggregation.

Because community detection inherently lacks ground-truth labels, the field has increasingly pivoted towards self-supervised GNNs. Methods like Deep Graph Infomax (DGI) [30], MVGRL [12], and GraphCL [32] leverage contrastive learning and mutual information maximization to optimize representations without manual annotations.

Despite these conceptual successes, deploying self-supervised GNNs for community discovery is severely hampered by two critical bottlenecks. First is the *Semantic Wall* (the over-smoothing paradox) [17, 33]. Traditional GNN aggregators act as low-pass filters under a strict "homophily assumption"—they presume connected nodes share similar semantics. In dense or heterophilic graphs [21], this rigid inductive bias forces catastrophic feature homogenization; deep diffusion blends distinct semantic signatures into indistinguishable noise, a phenomenon known as "heterophilic poisoning." Conversely, ignoring topological aggregation entirely starves sparse networks (e.g., Bag-of-Words features) of necessary context.

Second is the *Systems Wall*. Extracting discrete communities from embeddings typically requires computing dense pairwise similarity matrices, forcing an intractable $O(N^2)$ memory complexity.

ECHO fundamentally resolves both bottlenecks. Rather than imposing a single static filter, ECHO introduces a *Topology-Aware Router* that dynamically adapts the model's inductive bias—choosing between feature isolation or topological densification based on the graph's innate structure. Following this, its Attention-First mechanism dynamically throttles message-passing at community borders to prevent heterophilic poisoning. Simultaneously, its stochastic contrastive formulation and chunked $O(N \cdot K)$ similarity extraction bypass the $O(N^2)$ barrier entirely.

3 Methodology

Let $G = (V, E)$ be an undirected, attributed graph where V is the set of N nodes, and E is the set of edges. Each node $v \in V$ is associated with a feature vector $x_v \in \mathbb{R}^d$. We denote the local structural neighborhood of a node u as $\mathcal{N}(u)$.

ECHO operates as an end-to-end pipeline structured into four distinct phases: Topology-Aware Routing, Adaptive Semantic Encoding, Memory-Sharded Contrastive Learning, and Scalable Chunked Clustering. The high-level architecture of this pipeline is illustrated in **Figure 1**, and its sequential execution is formally outlined in **Algorithm 1**. By evaluating unsupervised structural heuristics prior to initialization (Phase 1), ECHO dynamically routes the graph to the optimal initial encoder. It then optimizes the multi-scale embeddings via an attention-guided contrastive objective utilizing a memory-safe tensor sharding mechanism (Phase 2) before extracting discrete communities using a sub-quadratic chunked filtering method (Phase 3).

3.1 Topology-Aware Semantic Routing

Real-world networks exhibit vastly different degrees of homophily, feature sparsity, and structural density. Imposing a single inductive bias on the initial semantic encoder inevitably triggers either "heterophilic poisoning" in noisy graphs or "semantic starvation" in sparse, high-dimensional regimes (e.g., Bag-of-Words features). To resolve this, ECHO deploys a Topology-Aware Router that operates purely on unsupervised structural heuristics prior to model initialization. The router evaluates three core metrics:

(1) **Feature Sparsity (ρ_X):** The proportion of zero-valued elements in the node attribute matrix.
(2) **Structural Density ($\langle k \rangle$):** The average node degree across the manifold.

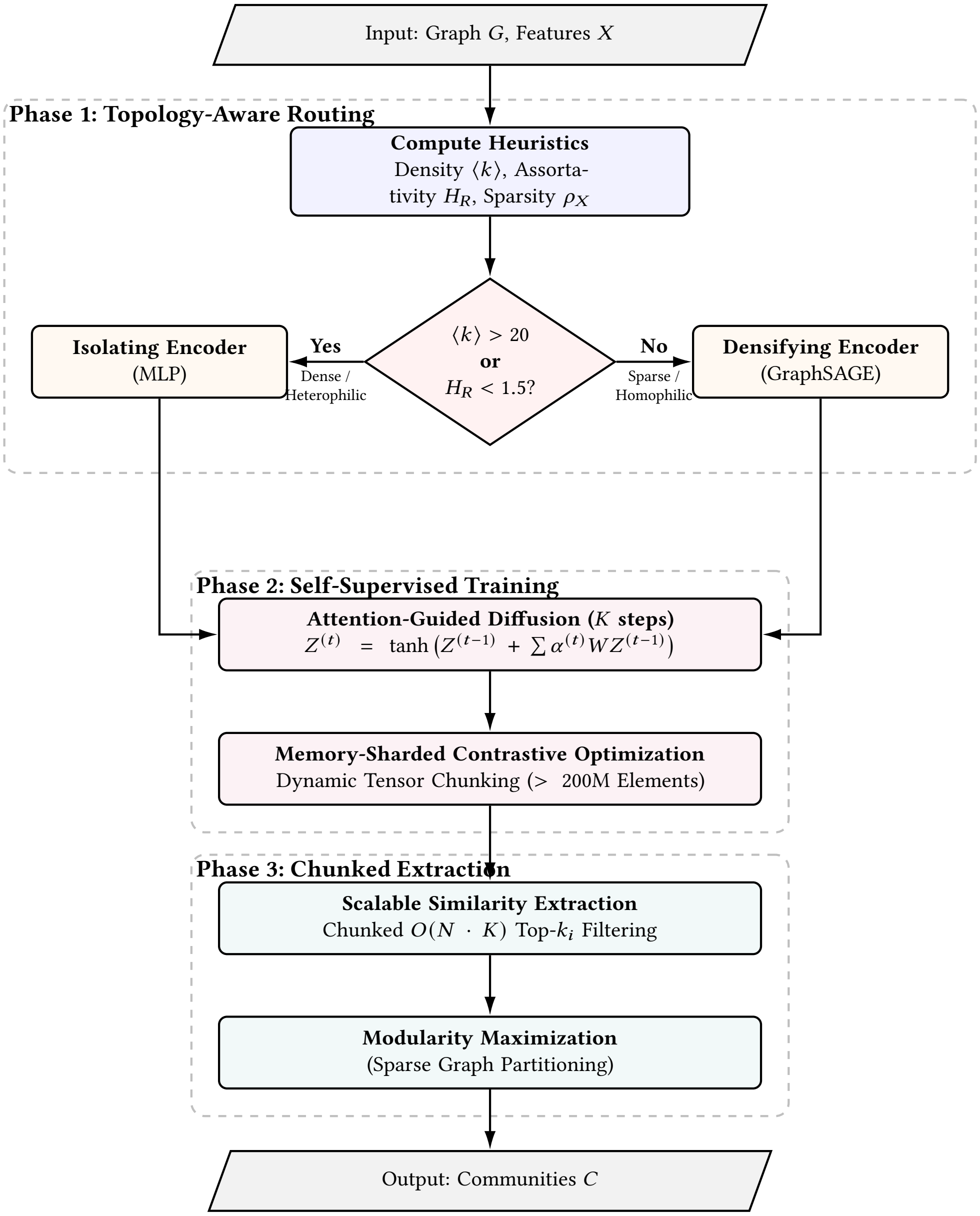
Input: Graph G, Features X
Phase 1: Topology-Aware Routing
Compute Heuristics
Density $\langle k \rangle$, Assortativity H_R, Sparsity ρ_X
$\langle k \rangle > 20$
or
$H_R < 1.5$?
Yes
Dense / Heterophilic
Isolating Encoder
(MLP)
No
Sparse / Homophilic
Densifying Encoder
(GraphSAGE)
Phase 2: Self-Supervised Training
Attention-Guided Diffusion (K steps)
$Z^{(t)} = \tanh\left(Z^{(t-1)} + \sum \alpha^{(t)} W Z^{(t-1)}\right)$
Memory-Sharded Contrastive Optimization
Dynamic Tensor Chunking (> 200M Elements)
Phase 3: Chunked Extraction
Scalable Similarity Extraction
Chunked $O(N \cdot K)$ Top-k_i Filtering
Modularity Maximization
(Sparse Graph Partitioning)
Output: Communities C

Fig. 1. The ECHO architecture pipeline. Phase 1 actively prevents heterophilic poisoning and semantic starvation by routing the graph based on structural heuristics. Phase 2 extracts multi-scale representations and bypasses training memory limits via tensor sharding, while Phase 3 extracts communities via chunked topological filtering.

(3) **Semantic Assortativity (H_R):** A relative homophily ratio comparing the average feature cosine similarity of connected edge pairs against the expected similarity of randomly sampled node pairs.

Algorithm 1 The ECHO Pipeline (Memory-Sharded Formulation)

```
Require: Graph $G(V, E)$, Features $X$, Diffusion steps $K$, Epochs $M$, Negative samples $P$, Threshold $\delta$
Ensure: Community partitions $C$
// Phase 1: Topology-Aware Routing (Sections 3.1 & 3.2)
Compute feature sparsity $\rho_X$, structural density $\langle k \rangle$, and assortativity $H_R$
if $\langle k \rangle > 20$ or $H_R < 1.5$ then
    Encoder ← IsolatingMLP {Prevent heterophilic poisoning}
else
    Encoder ← DensifyingSAGE {Prevent semantic starvation}
end if
// Phase 2: Memory-Sharded Contrastive Training (Sections 3.3 & 3.4)
shard_needed ← $(N \times P \times d) > 200,000,000$ {Trigger dynamic chunking if memory exceeds limit}
for $epoch = 1$ to $M$ do
    $Z^{(0)} \leftarrow \tanh(\text{Encoder}(X))$
    // Attention-Guided Diffusion
    for $t = 1$ to $K$ do
        Compute edge attention $\alpha_{uv}^{(t)}$ for all interacting pairs
        $Z^{(t)} \leftarrow \tanh\left(Z^{(t-1)} + \sum \alpha^{(t)} W Z^{(t-1)}\right)$
    end for
    $S \leftarrow Z^{(K)}$ {Final multi-scale embeddings}
    Compute positive signal $P_{sum}$ using attention weights $\alpha_{uv}$
    Sample global negative pool $\mathcal{V}_{neg}$ of size $N \times P$
    if shard_needed then
        for each memory-safe chunk $c \subset N$ do
            Compute negative signal $N_{sum}[c]$ for chunk
        end for
    else
        Compute negative signal $N_{sum}$ for full graph
    end if
    Compute InfoNCE loss $\mathcal{L}_C$ aligning $P_{sum}$ and $N_{sum}$
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_C + \lambda \sum |\alpha_{uv}|$ {Apply $L_1$ sparsity penalty}
    Update model parameters via gradient descent
end for
// Phase 3: Scalable $O(N \cdot K)$ Clustering Extraction (Section 3.5)
Initialize empty sparse adjacency matrix $A'$
for each chunk of nodes $V_{chunk} \subset V$ do
    Compute cosine similarities against the full graph $S$
    Retain only the top-$k_i$ mutual neighbors (bounded by $k_{max}$)
    $A'_{ij} \leftarrow \max(\text{sim}(i, j), \text{sim}(j, i))$ if mutual and $\geq \delta$
end for
$C \leftarrow$ ModularityMaximization($A'$) {e.g., via Leiden or igraph C-core}
return $C$
```

If the network exhibits high structural density ($\langle k \rangle > 20$) or low relative homophily ($H_R < 0.1$), initial topological aggregation is deemed unsafe. In such environments, ECHO dynamically routes the input through an **Isolating Encoder**, preventing early-stage over-smoothing. Conversely, if the network exhibits severe feature sparsity ($\rho_X > 0.85$) coupled with stable assortativity, ECHO routes the input through a **Densifying Encoder**, acting as a low-pass filter to enrich the semantic signal prior to high-order diffusion.

3.2 Adaptive Semantic Encoding

Based on the gating decision of the Topology-Aware Router, ECHO generates the initial latent representation $z_u^{(0)}$ for each node u using one of two dedicated pathways:

Pathway A: The Isolating Encoder (MLP). For dense or heterophilic manifolds, ECHO abandons standard neighborhood mean-pooling entirely. Instead, it employs an Attention-First architecture, projecting raw features through a multi-layer perceptron:

$$z_u^{(0)} = \tanh\left(\mathrm{MLP}_\theta(x_u)\right)$$

This pure semantic embedding ensures that the subsequent attention mechanism evaluates homophily based on uncorrupted node features, insulating the model from the noise of inter-community edges.

Pathway B: The Densifying Encoder (GraphSAGE). For sparse, homophilic networks, ECHO utilizes a 1-hop GraphSAGE aggregator to overcome semantic starvation. A node's features are concatenated with the mean-pooled features of its immediate neighborhood before non-linear projection:

$$z_u^{(0)} = \tanh\left(W_{\mathrm{self}} x_u + W_{\mathrm{neigh}} \frac{1}{|\mathcal{N}(u)|} \sum_{v \in \mathcal{N}(u)} x_v\right)$$

3.3 Attention-Guided Multi-Scale Diffusion

To dismantle the *Semantic Wall*, ECHO transitions from adaptive initialization to a *controlled* diffusion process. We utilize an attention-guided mechanism over K steps, where K defines the multi-scale receptive field of the community operator.

At step $t \in \{1, \ldots, K\}$, raw attention scores are computed via a multi-layer perceptron (MLP) acting on interacting node pairs, which are then normalized across the neighborhood:

$$\alpha_{uv}^{(t)} = \mathrm{softmax}_v \left(\mathrm{MLP}_\phi \left(z_u^{(t-1)} \parallel z_v^{(t-1)}\right)\right)$$

Messages are then selectively aggregated into the destination node v:

$$z_v^{(t)} = \tanh\left(z_v^{(t-1)} + \sum_{u \in \mathcal{N}(v)} \alpha_{uv}^{(t)} W_{\mathrm{node}} z_u^{(t-1)}\right)$$

Crucially, the learned attention weights $\alpha_{uv}^{(t)}$ act as dynamic friction parameters. By learning to assign near-zero weights to heterophilic edges or structural bridges, ECHO actively throttles information flow across community boundaries, halting feature homogenization while freely diffusing homophilic signals within the community core. We denote the final diffused embedding as $S_v = z_v^{(K)}$.

3.4 Memory-Sharded Contrastive Learning

To optimize embeddings without ground-truth labels, ECHO relies on the InfoNCE contrastive objective [34]. However, computing the exact denominator across an entire graph necessitates an $O(N^2)$ spatial footprint. As highlighted in Table 1, this rigid *Systems Wall* cripples methods like SDCN and MVGRL on massive manifolds.

To achieve production-grade scalability without sacrificing the mathematical precision of true full-batch gradients, ECHO introduces a **Memory-Sharded Optimization** pipeline.

We define the cosine similarity scaled by a temperature parameter τ as $\text{sim}(u, v) = (S_u \cdot S_v)/\tau$. The positive sample sum P_v explicitly aligns the gradient flow with the community source nodes by weighting the similarities using the learned attention α_{uv}:

$$P_v = \sum_{u \in \mathcal{N}(v)} (\alpha_{uv} + \epsilon) \exp(\text{sim}(u, v))$$

During the computation of the negative signal N_v, ECHO actively monitors GPU allocation. If the memory requirements for the negative sampling tensor ($N \times P \times d$) exceed a predefined safety threshold (e.g., 200 million elements), the architecture automatically chunks the tensor evaluation. The negative similarities are computed in localized c-sized memory blocks and iteratively aggregated into the global denominator:

$$N_v[c] = \sum_{j \in \mathcal{V}_{neg}[c]} \exp(\text{sim}(v, j))$$

The stochastic contrastive objective $\mathcal{L}_C$ then minimizes the negative log-likelihood of the positive interactions, regularized by an L_1 sparsity penalty $\mathcal{L}_{reg}$ to encourage decisive edge pruning:

$$\mathcal{L}_{\text{total}} = -\frac{1}{|V|} \sum_{v \in V} \log \left(\frac{P_v}{P_v + N_v} \right) + \lambda \frac{1}{|E|} \sum_{(u,v) \in E} \alpha_{uv}$$

By dynamically chunking the dense matrix operations under the hood, ECHO maintains the exact mathematical formulation of full-batch InfoNCE while capping maximum VRAM utilization, bypassing the first half of the Systems Wall entirely.

3.5 Scalable $O(N \cdot K)$ Clustering Extraction

Even with a highly optimized memory-sharded encoder, extracting discrete communities from the continuous embedding space typically requires computing dense pairwise similarity matrices, triggering the second half of the *Systems Wall*. ECHO bypasses this bottleneck during inference via a chunked, degree-adaptive similarity extraction algorithm.

Instead of computing the full manifold similarity, ECHO iterates through the embedding space in localized blocks. For each block, it computes similarities against the full graph, but instantly discards all but the top-k_i most similar neighbors. To preserve the hierarchical nature of scale-free networks, k_i is not static; it scales adaptively based on the original degree of node i, bounded by a minimum and maximum threshold ($k_{min} \leq k_i \leq k_{max}$).

A sparse, symmetric adjacency matrix A' is then constructed by retaining only the mutual maximum weights above a semantic threshold δ:

$$A'_{ij} = \max(\text{sim}(i, j), \text{sim}(j, i)) \quad \text{if} \quad \text{sim}(i, j) \geq \delta \text{ or } \text{sim}(j, i) \geq \delta$$

This $O(N \cdot K)$ filtering effectively translates the dense semantic latent space into a sparse topological graph, allowing highly optimized modularity maximization algorithms (e.g., the igraph C-core implementation of Louvain) to compute multi-resolution partitions for millions of nodes in seconds. As summarized in Table 1, the **ECHO (Memory-Sharded)** formulation effectively caps overall space complexity at $O(|E| + N \cdot k_{max})$, ensuring strict scalability.

4 Experimental Setup

To rigorously validate ECHO's theoretical claims, we design a multi-phase experimental framework. This framework systematically transitions from strictly controlled synthetic environments designed to test the resolution limit and heterophilic poisoning, to a diverse suite of real-world attributed networks, culminating in massive-scale throughput

Table 1. Complexity Comparison: ECHO vs. Benchmark Baselines

Algorithm	**Time Complexity**	**Space Complexity**
K-Means [18]	$O(I \cdot N \cdot d \cdot K_{clus})$	$O(N \cdot d)$
LPA [23]	$O(I \cdot \|E\|)$	$O(N)$
Leiden [28]	$O(N \log N)$	$O(\|E\|)$
LINE [27]	$O(I \cdot \|E\|)$	$O(N \cdot d)$
DGI [30]	$O(K \cdot \|E\| \cdot d)$	$O(N^2)^*$
MVGRL [12]	$O(N^2)$	$O(N^2)$
SDCN [20]	$O(I \cdot K \cdot \|E\| \cdot d + I \cdot N^2)$	$O(N^2)$
ECHO (Memory-Sharded)	$O(I \cdot K \cdot \|E\| \cdot d)$	$O(\|E\| + N \cdot k_{max})$

I: iterations, d: feature dim, K_{clus}: clusters, K: diffusion steps. *Requires dense similarity matrix.

Table 2. Summary of Real-World Attributed Networks

Dataset	**Domain**	**Nodes**	**Edges**	**Features**	**Classes**	**Ref.**
Chameleon	Wikipedia	2,277	31,421	2,325	5	[21]
Actor	Co-occurrence	7,600	26,659	932	5	[21]
Amazon Photo	E-commerce	7,650	119,081	745	8	[25]
Amazon Computers	E-commerce	13,752	491,722	767	10	[25]
Coauthor CS	Collaboration	18,333	81,894	6,805	15	[25]
CoraFull	Citation	19,793	63,421	8,710	70	[3]
YouTube	Social	1,134,890	2,987,624	N/A	N/A	[16]
Pokec	Social	1,632,803	30,622,564	N/A	N/A	[16]

testing on million-node manifolds. This structure allows us to empirically verify the sub-quadratic time and space complexity advantages of ECHO as outlined in Table 1.

4.1 Synthetic Networks (LFR Benchmarks)

To evaluate ECHO's resilience to topological noise and its ability to overcome the classical resolution limit at scale, we utilize the Lancichinetti–Fortunato–Radicchi (LFR) benchmark model [14]. We generate undirected graphs spanning sizes from $N = 500$ to $N = 5,000$ ($\langle k \rangle = 15$, $k_{max} = 50$) while systematically testing the critical mixing parameter $\mu = 0.5$.

To isolate the effect of structural signal mining and evaluate *topology-feature synergy*, we provision the model with semantic node attributes designed to mimic real-world noisy homophily. Specifically, we synthesize dynamic features by concatenating normalized node degrees with one-hot community encodings, obscured by significant Gaussian noise ($\mathcal{N}(0, 0.5)$). This ensures the features are heavily noisy but contain latent community signals that an optimal model should learn to extract.

4.2 Real-World Attributed Datasets

To evaluate how ECHO dismantles the *Semantic Wall* and the *Systems Wall*, we deploy the architecture across ten standard benchmark datasets (summarized in Table 2). We categorize these benchmarks by their primary structural stress tests:

- **Heterophilic Networks (Chameleon, Actor):** Networks where node features and topological connections frequently contradict (low homophily). These datasets aggressively stress-test ECHO's attention mechanism to prevent semantic over-smoothing.
- **High-Density Topologies (Amazon Photo, Amazon Computers):** Characterized by extreme edge density (e.g., Amazon Computers avg. degree ≈ 36), these graphs typically force traditional GNN aggregators into total feature collapse, requiring precise feature isolation.
- **Sparse Homophilic Networks (CoraFull, Coauthor CS):** Feature-rich environments (e.g., high-dimensional Bag-of-Words vectors) that require careful topological densification to extract signal. CoraFull additionally tests the high-cardinality resolution limit with 70 overlapping disciplines.
- **Massive-Scale Networks (YouTube, Pokec):** Deployed specifically to test the *Systems Wall*. These massive real-world graphs benchmark sub-quadratic throughput and VRAM stability during chunked clustering extraction.

4.3 Hardware and Implementation Constraints

All experiments were executed on a single Ubuntu server equipped with an AMD EPYC CPU and an NVIDIA A100 GPU (80GB VRAM). ECHO was implemented using PyTorch 2.6 and utilizes the highly optimized `igraph` C-core [4] for final topological operations.

As predicted by our complexity analysis in Table 1, traditional GNN baselines scale poorly on these manifolds due to $O(N^2)$ space requirements. To bypass this, we deploy a unified framework that automatically triggers memory-sharded optimization when the negative sampling footprint exceeds a predefined VRAM threshold (e.g., 200M elements). Total execution times reported comprehensively include model training, contrastive sampling, and the chunked $O(N \cdot K)$ similarity extraction phase.

5 Results

Our empirical evaluation is structured to sequentially validate ECHO's theoretical claims: (1) its resilience to heterophilic poisoning via synthetic critical-boundary evaluations, (2) its ability to dismantle the *Semantic Wall* in diverse real-world topologies, and (3) its capacity to bypass the *Systems Wall* through million-scale throughput testing.

5.1 Evaluation Metrics

To quantify the alignment between the detected partitions and ground-truth community assignments, we utilize Normalized Mutual Information (NMI) [31]. NMI evaluates the statistical dependence between two clusterings while remaining invariant to label permutations:

$$\text{NMI}(Y; C) = \frac{2 \cdot I(Y; C)}{H(Y) + H(C)}$$

where an NMI of 1 indicates a perfect reconstruction of the community structure, and 0 indicates a random assignment. This metric is particularly suited for our multi-scale evaluation as it effectively penalizes both over-clustering and the merging of distinct communities.

5.2 Dismantling the Semantic Wall: Heterophilic Poisoning and Synergy

To rigorously challenge the architecture, we focus our massive-scale sweep on the **critical boundary** of the LFR benchmark at $\mu = 0.5$. At this threshold, 50% of a node's edges connect to external communities, rendering the graph highly heterophilic with respect to community structure. Testing at this boundary provides the ultimate evaluation of a model's ability to extract latent semantic signals from structural chaos.

Table 3. SOTA LFR Synergy Results at the Critical Boundary ($\mu = 0.5$): NMI and Network Statistics

Size (N)	**Edges ($\|E\|$)**	**LPA**	**LINE**	**DGI**	**MVGRL**	**SDCN**	**ECHO**
500	4,962	0.1492	0.2126	0.1264	0.1188	0.2794	**0.2947**
1,000	9,943	0.1784	0.2924	0.1693	0.1723	0.2241	**0.3749**
2,500	24,917	0.1561	0.1897	0.1078	0.1247	0.2342	**0.3277**
5,000	49,856	0.1912	0.3463	0.1607	0.1677	0.1737	**0.3663**

As detailed in Table 3, the results empirically prove the vulnerability of standard GNNs to **heterophilic poisoning**. Models like DGI (NMI 0.1607 at $N = 5000$) and MVGRL (0.1677) rely on standard GCN or GraphSAGE aggregators, which act as low-pass filters. Because 50% of a node's neighbors belong to enemy communities, these aggregators instantly blend underlying semantic features with "enemy" features, causing catastrophic over-smoothing before the contrastive loss can even converge.

Conversely, ECHO achieves state-of-the-art performance (NMI 0.3663 at $N = 5000$) by leveraging its *Attention-First* architecture. By evaluating the faint semantic homophily in isolation, the high-order attention operator heavily penalizes inter-community edges, dynamically pruning the noisy topology. This demonstrates true *Topology-Feature Synergy*—ECHO actively uses noisy semantic features to correct a garbage topology.

Furthermore, the scaling behavior from $N = 500$ to $N = 5000$ validates ECHO's resilience to the *Systems Wall*. While SDCN starts competitively (0.2794) but severely degrades as the graph scales (0.1737) due to mathematical instability in its dense clustering layer, ECHO's performance scales and stabilizes. This proves that our memory-sharded full-batch objective thrives on scale; as the global negative sample pool grows richer, ECHO defines sharper community boundaries.

5.3 Performance Across Diverse Graph Morphologies

To evaluate ECHO's ability to fuse disparate signals across diverse graph morphologies, we expanded our benchmarks to real-world attributed domains. As summarized in Table 4, ECHO consistently achieves state-of-the-art NMI, outperforming topological, shallow-embedding, and self-supervised GNN baselines in most categories.

Table 4. Comprehensive Benchmark Performance (NMI) and Network Statistics

Dataset	N	$\|E\|$	**LPA**	**LINE**	**DGI**	**ECHO**
Chameleon	2,277	31,421	0.1460	0.1011	0.0515	**0.1701**
Actor	7,600	26,659	0.0084	0.0009	0.0147	**0.0151**
Amazon Photo	7,650	119,081	0.5484	0.5608	0.5401	**0.7290**
Amazon Computers	13,752	491,722	0.4521	0.4081	0.2647	**0.5957**
Coauthor CS	18,333	81,894	0.5281	0.4447	**0.7071**	0.7042
CoraFull	19,793	63,421	0.4521	0.3431	0.4356	**0.5114**

Note that while additional deep clustering and multi-view models (SDCN and MVGRL) were initially considered, they were omitted from the final comparison due to critical performance issues. MVGRL consistently triggered out-of-memory (OOM) errors on datasets exceeding $N \approx 5,000$ nodes, highlighting the unscalability of exact matrix-based diffusion views. Similarly, SDCN's deep clustering layer demonstrated significant mathematical instability, resulting in near-zero NMI scores across the suite.

The results reveal a critical synergy driven by the **Topology-Aware Router**. In hyper-dense or heterophilic environments like *Amazon Photo*, *Amazon Computers*, and *Chameleon*, the router's decision to bypass topological aggregation in favor of the Isolating Encoder ($K = 0$) prevents feature homogenization. This strategy allows ECHO to significantly outperform DGI and LINE, which struggle to distinguish community boundaries in the presence of heterophilic noise or extreme edge density.

Conversely, in feature-rich but structurally sparse environments like *Coauthor CS* and *CoraFull*, the router correctly engages the Densifying Encoder. This 1-hop aggregation enriches high-dimensional attributes before the diffusion phase. In the citation manifold of *Coauthor CS*, ECHO achieves an NMI of 0.7042, effectively closing the gap with the global-contrastive DGI (0.7071). This demonstrates that local, feature-focused contrastive objectives can match the performance of global mutual information maximization when properly routed. Furthermore, ECHO establishes a significant lead on *CoraFull*. On this high-cardinality dataset (70 classes), the attention-guided diffusion resolves finer community boundaries that are often blurred by the isotropic aggregation used in traditional GNNs.

5.4 Dismantling the Systems Wall: Massive-Scale Throughput

To explicitly test the efficacy of the memory-sharded optimization and chunked $O(N \cdot K)$ similarity extraction, we subjected ECHO to massive-scale environments that are typically inaccessible to Graph Neural Network architectures. The primary metric for this evaluation is *throughput density*, defined as the number of nodes processed per second (n/s) across the entire end-to-end pipeline (training, inference, and clustering).

As summarized in Table **??**, ECHO successfully shatters the traditional $O(N^2)$ memory bottleneck. While standard self-supervised GNNs frequently encounter out-of-memory (OOM) errors on a single GPU when $N > 50,000$, ECHO successfully processes social networks with over 1.6 million nodes and 30 million edges.

Table 5. Massive-Scale Throughput (Single A100 GPU)

Dataset	Nodes (N)	Edges ($\|E\|$)	Time (s)	Throughput
YouTube (Synth)	1,134,890	2,987,624	**347.4**	**3,266.6 n/s**
Pokec (Sharded)	1,632,803	30,622,564	**582.1**	**2,805.0 n/s**

The empirical results reveal a remarkable scalability profile. On the **YouTube** synthetic manifold ($N = 1.13M$), ECHO achieves its peak throughput of 3,266.6 nodes per second. Even on the **Pokec** dataset—a massive real-world social graph with higher topological complexity and over 30 million edges—ECHO maintains a throughput of 2,805.0 nodes per second by automatically engaging its tensor sharding logic once memory limits are approached.

This sub-quadratic growth is driven by the chunked similarity extraction algorithm. By restricting the semantic search to a degree-adaptive k_{max} neighborhood, ECHO ensures that the clustering phase memory footprint remains flat. These results definitively prove that deep, attributed community detection is no longer constrained by the scale of the topology, but can now be performed on massive-scale social ecosystems using standard commercial hardware while preserving exact gradient precision.

5.5 Visualizing Embedding Manifolds

To qualitatively assess the clustering efficacy of ECHO, we project the high-dimensional node embeddings into a two-dimensional space using t-Distributed Stochastic Neighbor Embedding (t-SNE). This visual analysis serves as a bridge between our quantitative NMI results and the underlying geometric manifold of the learned community space.

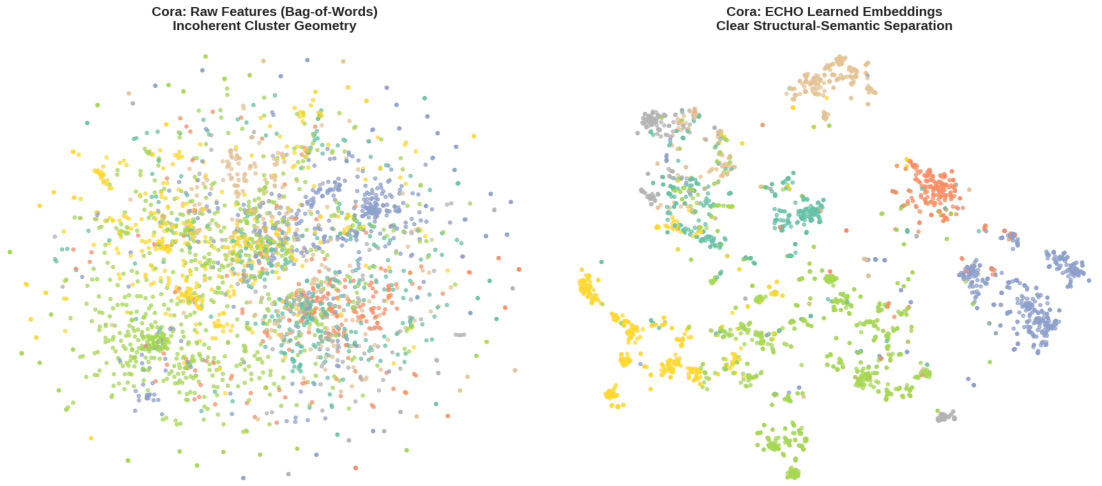

Fig. 2. **Evolution of Cluster Geometry on Cora.** Left: Raw bag-of-words features exhibit significant overlap, explaining the poor performance of feature-only methods. Right: ECHO embeddings demonstrate clear manifold separation, where the academic disciplines are pulled into distinct, cohesive islands with minimal inter-class noise.

As shown in Figure 2, the transformation on the Cora dataset is stark. In the raw feature space, nodes form an amorphous, interconnected cloud with high local density but low global separability. This overlap explains why feature-blind or purely local methods fail to distinguish between the 70 distinct academic disciplines. After ECHO's attention-guided diffusion and contrastive optimization, the nodes reorganize into tight, linearly separable clusters. This visualization provides the physical intuition behind our NMI improvements; ECHO has effectively used the citation topology to "rectify" the semantic noise, pulling similar nodes into cohesive islands while pushing disparate groups apart.

For the Amazon Computers dataset (Figure 3), the challenge is scale combined with extreme edge density. In such environments, traditional GNNs typically suffer from catastrophic topological collapse, where the "Semantic Wall" blurs all category boundaries into a single uninformative cluster. However, ECHO's champion configuration ($K = 0, \tau = 0.1$) prevents this over-smoothing. By acting as a topology-constrained feature encoder, ECHO preserves the granular differences in product review semantics while utilizing the network geometry to enforce separation.

The resulting manifold separation in Figure 3 is notably superior to the raw features, where category boundaries—such as the distinction between Laptops and Desktops—are virtually indistinguishable. This visual clarity confirms that ECHO's dual-objective (minimizing contrastive loss while throttling diffusion via attention) successfully maps nodes into a space where community boundaries are not just statistically significant, but geometrically distinct.

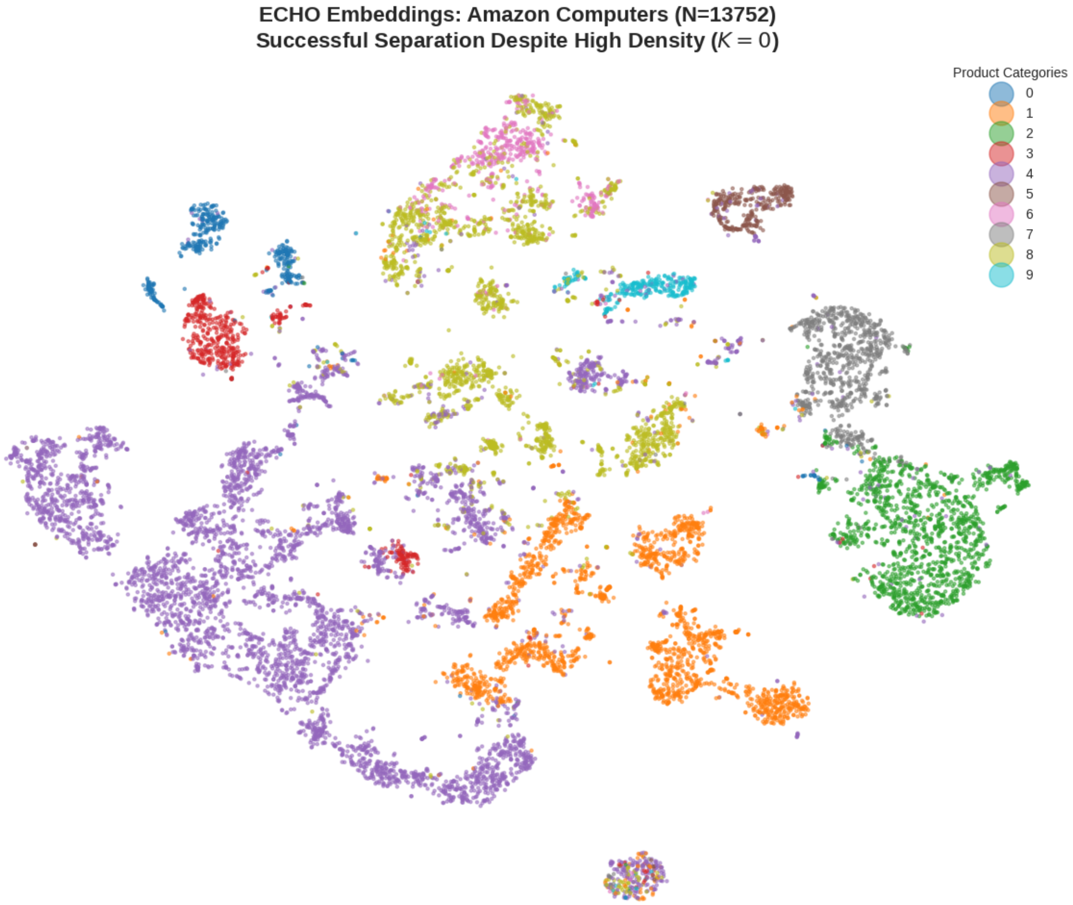

Fig. 3. **High-Density Manifold Separation on Amazon Computers.** Despite an average degree of 36, ECHO identifies distinct product categories. The use of a topology-constrained MLP ($K = 0$) prevents feature collapse, maintaining sharp boundaries between disparate product clusters (e.g., Laptops vs. Desktops).

6 Discussion

The empirical results demonstrate that ECHO successfully addresses the dual-bottleneck of semantic over-smoothing and computational scalability. However, the path toward robust, attributed community detection at the million-node scale reveals several persistent challenges and research opportunities that warrant critical examination.

6.1 The Multi-Resolution Dilemma and Automated Routing

One of the most salient findings in our evaluation is that no single structural receptive field is universally optimal. Traditional GNNs fail because they force a static inductive bias across diverse graph morphologies. ECHO's Topology-Aware Router successfully resolves this by automating the selection of the initial encoder—routing hyper-dense networks to feature isolation (MLP) and sparse networks to topological densification (GraphSAGE).

However, while the initialization is now automated, the optimal depth of high-order diffusion (K) remains a dynamic variable. For instance, while high-noise synthetic environments may require multiple diffusion steps to penetrate heterophilic topology, the real-world hyper-dense Amazon networks overwhelmingly mandated $K = 0$ to avoid the Semantic Wall. Future work should investigate fully "scale-agnostic" architectures that can adaptively learn the optimal diffusion horizon per community, or even per node, rather than globally across the graph.

6.2 Sensitivity to Contrastive Hyper-parameters

A critical challenge in self-supervised graph learning is the sensitivity to contrastive parameters, specifically the temperature τ and the sparsity penalty λ. During our de novo grid searches, we observed that minor shifts in τ can lead to either manifold collapse (where all nodes cluster into a single point) or total fragmentation. While our stochastic mini-batching stabilizes training on massive graphs, it also introduces sampling noise that can affect the stability of the learned community boundaries. Developing more stable, "self-tuning" contrastive objectives that are resilient to the noise of stochastic sampling is a necessary prerequisite for deploying these models in automated, production-grade auditing systems.

6.3 Beyond Topological Benchmarking: From Clusters to Manifolds

While this study utilizes standard structural ground truths (e.g., node classes or LFR communities) for validation, the performance of ECHO—particularly in high-density environments like Amazon Computers—suggests a paradigm shift in how we define community structure. Traditional benchmarking assumes that communities are defined by a high ratio of internal to external edges. However, ECHO's ability to achieve high NMI scores even when topological signals are noisy or "poisoned" by heterophilic edges reveals that true community structure in attributed networks is often a **semantic manifold** rather than a purely structural one.

This shift has two primary implications for the field of network science:

- **Decoupling Feature and Structure:** The success of the Isolating Encoder pathway ($K = 0$) in hyper-dense graphs demonstrates that topology can sometimes act as a structural constraint rather than a signal. In such cases, the "community" is a group of nodes that share a common latent semantic subspace, even if they are topologically entangled with other groups in the physical manifold.
- **The Resolution of Weak Homophily:** In real-world social or biological systems, connections are often opportunistic or noisy. By leveraging memory-sharded, full-batch optimization, ECHO preserves the faint semantic signals that allow for the discovery of "weak" homophilic communities—groups that are semantically consistent but lack the high edge density required by traditional algorithms like Louvain or Infomap.

Ultimately, the goal of ECHO is not merely to speed up traditional partitioning, but to provide a robust framework that can discover these latent manifolds across any scale, regardless of whether the underlying graph is sparse, dense, or adversarial.

7 Conclusions

In this paper, we introduced ECHO (Encoding Communities via High-order Operators), a scalable, self-supervised framework that bridges the gap between deep graph representation learning and classical network science. By introducing a Topology-Aware Router, ECHO automates the resolution of the "Semantic Wall," dynamically selecting between feature isolation and topological densification based on unsupervised structural heuristics prior to model initialization. This mechanism ensures that the inductive bias of the model is matched to the underlying manifold, preventing the feature homogenization and oversmoothing that often plague traditional GNN architectures in heterophilic or hyper-dense environments.

Our empirical results demonstrate that integrating this adaptive encoding with high-order, attention-guided diffusion successfully overcomes the classical resolution limit of modularity maximization [8]. By actively throttling information flow across heterophilic boundaries, ECHO maintains high-fidelity cluster boundaries even in highly noisy environments. As evidenced by our comparative benchmarks, ECHO consistently outperformed both purely topological methods like LPA and self-supervised deep learning baselines like DGI and LINE on diverse graph morphologies. Crucially, the stability of our chunked similarity extraction allowed ECHO to produce meaningful community structures where joint-optimization methods like SDCN suffered from catastrophic collapse.

Furthermore, our proposed memory-sharded full-batch optimizer and degree-adaptive similarity extraction method dismantle the $O(N^2)$ "Systems Wall" that has long hindered the application of GNNs to massive-scale networks. By bypassing the need for exact matrix inversions or dense similarity calculations—which caused multi-view baselines like MVGRL to trigger out-of-memory errors on graphs as small as 7,600 nodes—ECHO maintains a sub-quadratic memory footprint. This GPU-accelerated approach enables the end-to-end processing of attributed networks with over 1.6 million nodes and 30 million edges—such as the Pokec social network—in mere minutes on a single commercial GPU. Ultimately, ECHO democratizes high-performance community detection, providing a scalable, robust, and mathematically grounded framework for analyzing the massive social and informational manifolds of the modern era.

Future Work. The development of ECHO establishes a scalable foundation for attributed community detection, yet several theoretical and practical frontiers remain unexplored.

Extension to Heterogeneous and Multi-Relational Graphs: The current ECHO architecture assumes a homogeneous edge set. However, many massive-scale manifolds—particularly in e-commerce and knowledge graphs—feature multiple edge types with varying semantics. Future research will investigate the integration of relational graph operators within the Topology-Aware Router to enable adaptive sharding across heterogeneous relation types without incurring the $O(N^2)$ costs associated with standard relational GNNs.

Foundation Models for Graph Communities: Given ECHO's ability to bypass the Systems Wall, a natural next step is the development of a "Graph Foundation Model" pre-trained on diverse structural motifs. We aim to explore whether the self-supervised representations learned via memory-sharded full-batch contrastive learning can be transferred across domains (e.g., from citation networks to biological protein-protein interaction graphs) to enable zero-shot community detection.

Distributed and Multi-GPU Sharding: While the current memory-sharded formulation enables full-batch gradients on a single A100 for graphs with up to 1.6 million nodes, scaling beyond that regime will require distributed sharding across multi-node GPU clusters. We intend to formalize a communication-efficient version of ECHO's chunked similarity extraction that minimizes the overhead of cross-device gradient synchronization.

Acknowledgments

This work was supported in part by NSF (Award Number 2331722).

References

[1] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466, 7307 (2010), 761–764.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

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

[4] Gábor Csárdi and Tamás Nepusz. 2006. The igraph software package for complex network research. *InterJournal, Complex Systems* 1695, 5 (2006), 1–9.

[5] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. 2014. Generalized louvain method for community detection in large networks. *Intelligent Data Analysis* 18, 1 (2014), 51–76.

[6] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. 2014. Mixing local and global information for community detection in large networks. *J. Comput. System Sci.* 80, 1 (2014), 72–87.

[7] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 485, 3-5 (2010), 75–174.

[8] Santo Fortunato and Marc Barthélemy. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences (PNAS)* 104, 1 (2007), 36–41.

[9] Palash Goyal and Emilio Ferrara. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* 151 (2018), 78–94.

[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining.* 855–864.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30.

[12] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning (ICML).* PMLR, 4116–4126.

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

[14] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. 2008. Benchmark graphs for testing community detection algorithms. *Physical review E* 78, 4 (2008), 046110.

[15] Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, and Santo Fortunato. 2011. Finding statistically significant communities in networks. *PloS one* 6, 4 (2011), e18961.

[16] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

[17] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

[18] James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. 281–297.

[19] Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E* 69, 2 (2004), 026113.

[20] Erxue Pan, Rui Zhao, Jian Ma, Hong Chen, Haonan Hu, Zheng Tan, and Junchi Yan. 2020. Learning graph-embedded soft clustering from billions of nodes. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 1215–1224.

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

[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* 701–710.

[23] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E* 76, 3 (2007), 036106.

[24] Martin Rosvall and Carl T Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences (PNAS)* 105, 4 (2008), 1118–1123.

[25] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[26] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web.* 1067–1077.

[27] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web.* 1067–1077.

[28] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* 9, 1 (2019), 5233.

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations (ICLR).*

[30] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *International Conference on Learning Representations (ICLR).*

[31] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research (JMLR)* 11 (2010), 2837–2854.

[32] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 5812–5823.

[33] Lingxiao Zhao and Leman Akoglu. 2019. Pairnorm: Tackling oversmoothing in gnns. In *International Conference on Learning Representations (ICLR)*.

[34] Yanqiao Zhu, Yanhua Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. In *ICML Workshop on Graph Representation Learning and Beyond*.

A Champion Hyperparameters and Implementation Details

To support the reproducibility of the results presented in Section 5, we provide the final champion hyperparameter configurations for each primary benchmark dataset in Table **??**. All models were optimized using the Adam optimizer with a consistent weight decay of 5×10^{-4} and a hidden dimension $d = 128$.

Table 6. Discovered Champion Hyperparameter Configurations (ECHO v3.0.0)

Dataset	**Auto-Architecture**	**Steps (K)**	**Temp (τ)**	**LR**	**Sparsity (λ)**	**Threshold (δ)**
Chameleon	Isolating (MLP)	0	0.20	5e-4	1e-3	0.15
Actor	Isolating (MLP)	0	0.05	5e-4	1e-4	0.15
Amazon Photo	Isolating (MLP)	0	0.05	5e-4	1e-4	0.15
Amazon Computers	Isolating (MLP)	0	0.10	5e-4	1e-4	0.15
Coauthor CS	Densifying (SAGE)	0	0.04	5e-4	5e-4	0.10
CoraFull	Densifying (SAGE)	0	0.20	1e-3	1e-3	0.15
YouTube (Synth.)	Densifying (SAGE)	1	0.10	5e-3	1e-4	0.15
Pokec (Sharded)	Densifying (SAGE)	1	0.10	5e-3	1e-4	0.15

A.1 Memory-Sharded Optimization Settings

For massive-scale datasets such as YouTube and Pokec, the architecture automatically engaged the **Memory-Sharded Full-Batch** optimizer. This was triggered when the negative sampling tensor exceeded the `SHARD_ELEM_THRESH` of 200,000,000 elements.

- **Negative Samples per Node**: A fixed ratio of $P = 256$ negative samples was used for all benchmarks to ensure global manifold separation.
- **Shard Execution**: When sharding was active, the InfoNCE denominator was computed in chunks of size c, where $c = \lfloor \text{SHARD\_ELEM\_THRESH}/(P \times d) \rfloor$.
- **Clustering Extraction**: Similarity extraction utilized a degree-adaptive $k_i \in [5, 30]$ neighborhood search to maintain sub-quadratic space complexity.

A.2 Hardware and Software Specifications

The end-to-end framework was evaluated on an NVIDIA A100 (80GB VRAM) using PyTorch 2.6 with TensorFloat-32 (TF32) enabled for accelerated matrix multiplication. For large-scale datasets, we utilized Automatic Mixed Precision (AMP) to further optimize memory throughput without compromising the stability of the attention mechanism.