# Theoretical Justification for a Hierarchical Embedded Graph Approach to Shortest Path Abstraction and Knowledge Validation

## 1 Introduction

Graphs provide a powerful and intuitive formalism for representing structured knowledge. With the rise of large-scale knowledge graphs in domains such as citation networks, biological systems, and commonsense reasoning, the need for fast, scalable, and semantically aware algorithms to extract meaningful paths and insights has become increasingly critical. Traditional shortest path algorithms, while exact, do not scale well with graph size due to their inherent lack of parallelism and computational complexity.

We present a hybrid approach that integrates the benefits of hierarchical graph summarization and community-aware node embedding to accelerate and abstract shortest path queries. Our system is also designed to evolve into a fact-checking and knowledge validation engine for dynamically changing graphs. This document provides a theoretical justification for this approach grounded in recent literature on graph embeddings and summarization.

## 2 Graph Embeddings: Structure and Semantics

Graph embedding techniques aim to map nodes, edges, or substructures into a low-dimensional continuous space such that structural properties are preserved. Structural embeddings like Node2Vec [1] use random walks to encode node proximity, capturing community and role-based features via biased sampling schemes. These embeddings have been successful for tasks such as link prediction, node classification, and clustering.

However, traditional node embeddings face several limitations in the context of pathfinding:

- They lack explicit encoding of higher-order graph topology (e.g., community hierarchies).
- They are not robust to changes in graph structure, requiring retraining for dynamic graphs.
- They cannot represent routing preferences or abstracted semantic paths.

By embedding nodes within communities independently, we preserve local structure while reducing the scale of each embedding problem. This also permits adaptive embeddings under dynamic changes localized to subgraphs.

## 3 Graph Summarization: Compression and Efficiency

Graph summarization seeks to compress graphs by reducing their size while retaining task-relevant information. Summarization methods include:

- Community aggregation [2]
- Router-based compression [3]

- Node pruning [4]
- Minimum description length (MDL) [5]

While these approaches reduce graph complexity, they often compromise fine-grained connectivity information necessary for accurate path queries. Our method retains the skeleton of inter-community connections and augments it with boundary node embedding, enabling efficient yet precise routing through abstracted topologies.

# 4 Justification for the Hybrid Approach

Our algorithm leverages the strengths of both embeddings and summarization:

- **Multi-level Community Detection:** Louvain or METIS allows recursive abstraction of the graph into successively smaller graphs. This aligns with HARP [6], which proposes hierarchical embedding for robustness and scalability.
- **Per-Community Embedding:** Ensures local structure preservation and allows routing heuristics to remain meaningful.
- **Boundary Node Selection:** Inspired by influence-based summarization [7], selecting nodes using degree, PageRank, and embedding deviation.
- **Fast Bitmask Indexing:** Enables constant-time rejection in A* planning.
- **Hierarchical Path Planning:** Mimics real-world network routing using intra-community A*, inter-community skeleton, and descent phases.

# 5 Theoretical Advantages

- **Sublinear Query Time:** Uses coarse abstraction and fast rejection for scalability.
- **Localized Adaptability:** Supports re-embedding subgraphs independently.
- **Semantic Abstraction:** Communities approximate latent concepts in knowledge domains.
- **Support for Distributed Learning:** Queries become experiments that train the graph's representation.

# 6 Conclusion

This hybrid model positions itself as a scalable, dynamic alternative to monolithic graph embedding or pure summarization techniques. By structurally and semantically encoding both intra- and inter-community relations, our approach allows abstract reasoning, fast path queries, and online knowledge validation. It opens a path toward self-programming systems where queries not only retrieve information but generate the training signals to improve the graph's semantic and structural representation over time.

# 7 Empirical Results and Limitations

We benchmarked this system on the Citeseer citation graph. Our parallel A* implementation was compared with NetworkX Dijkstra over queries constrained to long paths.

### Average Time Per Path

| Method | Avg Time (ms) |
|---|---|
| NetworkX Dijkstra | 0.69 |
| Parallel A* | 1.82 |

### Observed Weaknesses

- Overhead of thread scheduling exceeds gains for small query sets.
- L2 distance in high dimensions is expensive.
- No caching or batching; subpaths are redundantly explored.

## Appendix A: Sample Path Outputs

- 407 $\rightarrow$ 430: [407, ..., 430] (10 hops)
- 1218 $\rightarrow$ 3197: [1218, ..., 3197] (4 hops)
- 1252 $\rightarrow$ 1865: [1252, ..., 1865] (10 hops)

## References


[1] Aditya Grover and Jure Leskovec. *node2vec: Scalable Feature Learning for Networks*. KDD, 2016.

[2] V. Kumar and G. Karypis. *Analysis of Multilevel Graph Partitioning*. SC, 1998.

[3] A. Maccioni and D. Abadi. *Scaling Knowledge Graphs with ROAR*. SIGMOD, 2016.

[4] Z. Lin et al. *Node Removal for Pathfinding in Large Graphs*. TKDE, 2013.

[5] D. Chakrabarti et al. *Graph Mining Using MDL*. ICDM, 2004.

[6] H. Chen et al. *HARP: Hierarchical Representation Learning*. AAAI, 2018.

[7] T. Safavi and D. Koutra. *Personalized Summarization of Influence Graphs*. WWW, 2019.