COLLEGE OF ENGINEERING
COMPUTER SCIENCE
VIRGINIA TECH

NeurIPS 2023:

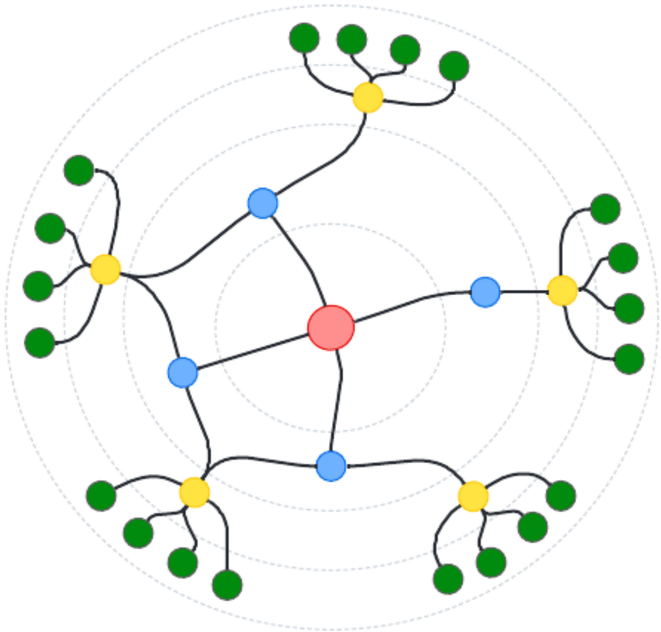
Hyperbolic Graph Neural Networks at Scale: A Meta Learning Approach

Authors: Nurendra Choudhary, Nikhil Rao, Chandan K. Reddy

Introduction

Scalable hyperbolic models

- In Euclidean Graphs, we depend on local subgraph encodings to scale over large graph datasets.
- In Hyperbolic Graphs, we are not able to directly apply this because the representations are relative to an origin.

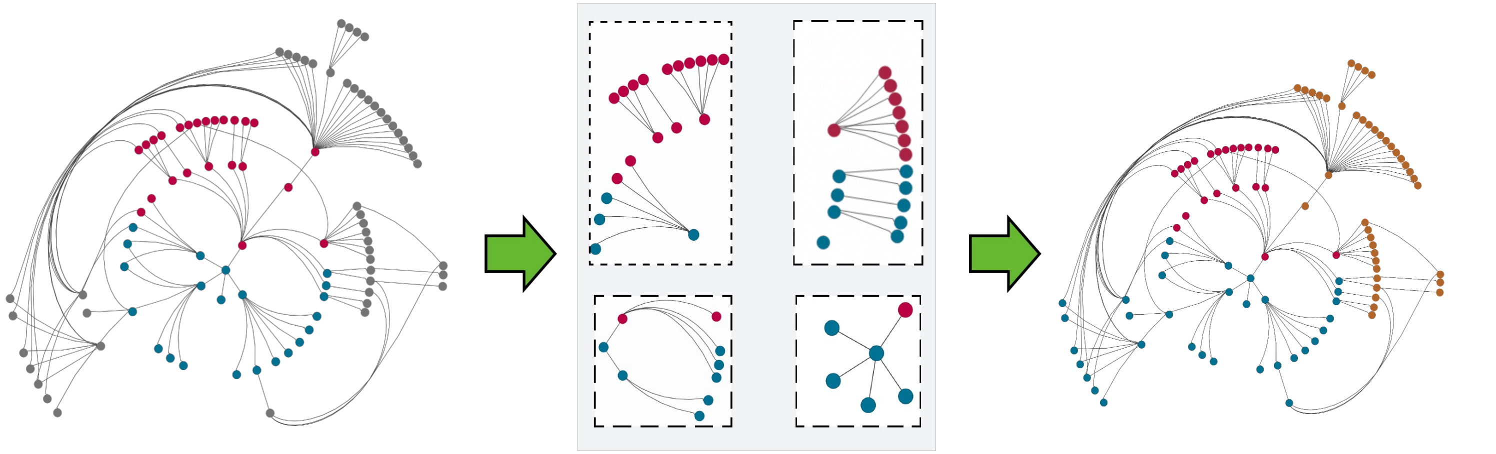


Hyperbolic Embeddings

Our Solution: H-GRAM

Key Ideas

- It can be theoretically shown that one can move the origin to local subgraphs with a bounded information loss.

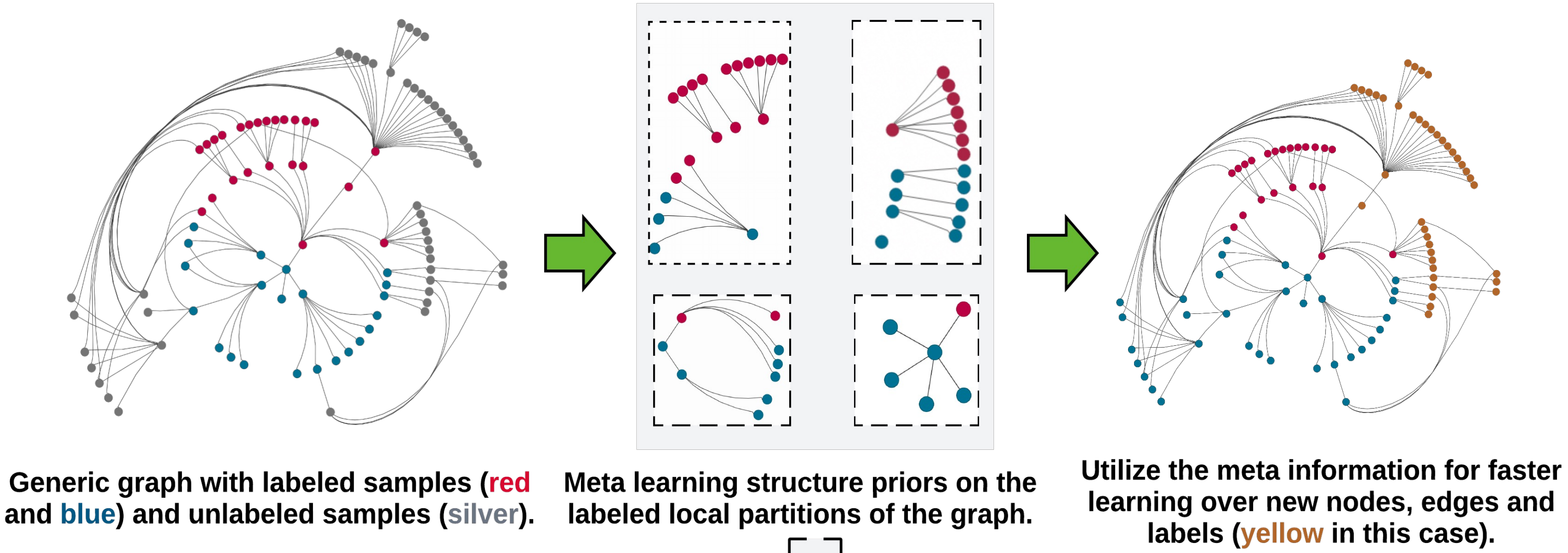


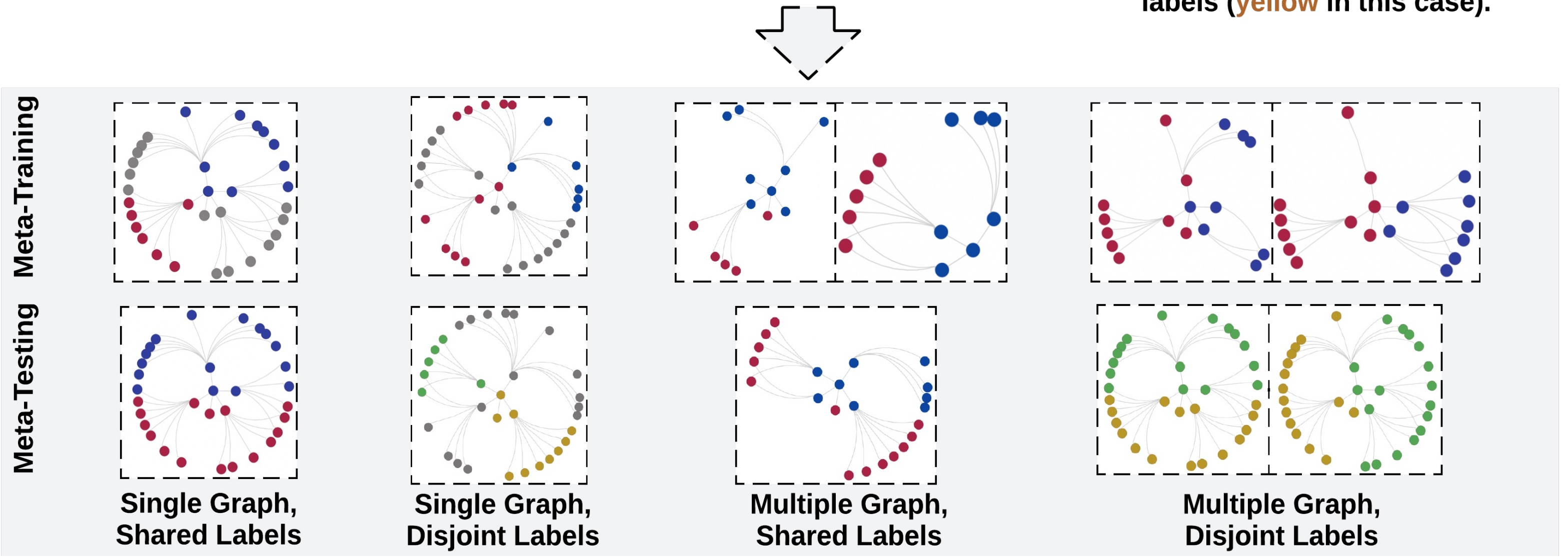
Generic graph with labeled samples (red and blue) and unlabeled samples (silver). Meta learning structure priors on the labeled local partitions of the graph. Utilize the meta information for faster learning over new nodes, edges and labels (yellow in this case).

Our Solution: H-GRAM

Graph Sections

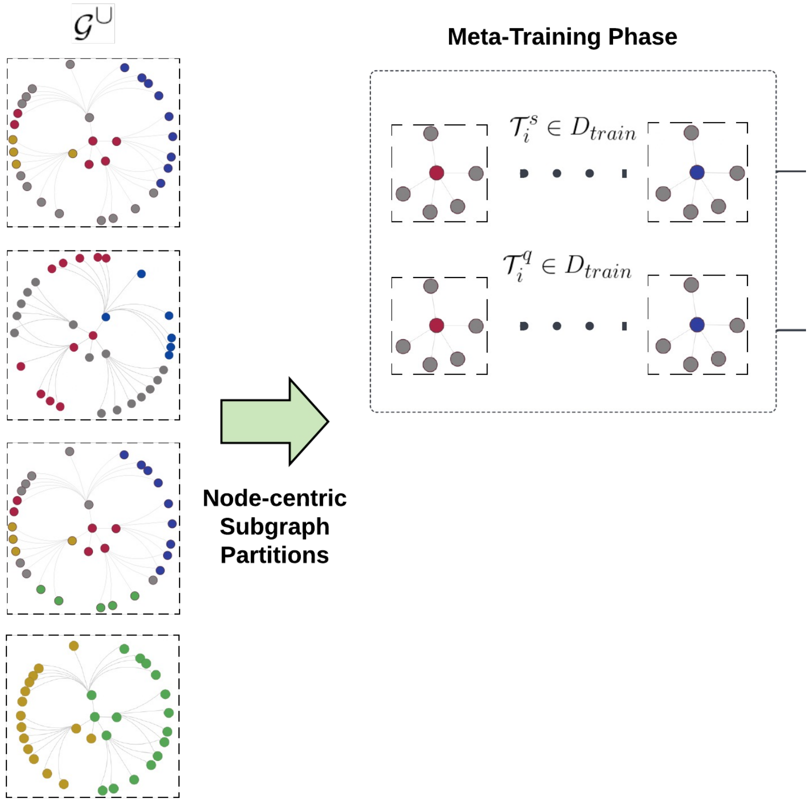
- It can be theoretically shown that one can move the origin to local subgraphs with a bounded information loss.
- Divide the graph into subgraphs and note four possible scenarios:
  - Single Graph, Shared Labels
  - Single Graph, Disjoint Labels
  - Multiple Graph, Shared Labels
  - Multiple Graph, Disjoint Labels





Our Solution: H-GRAM

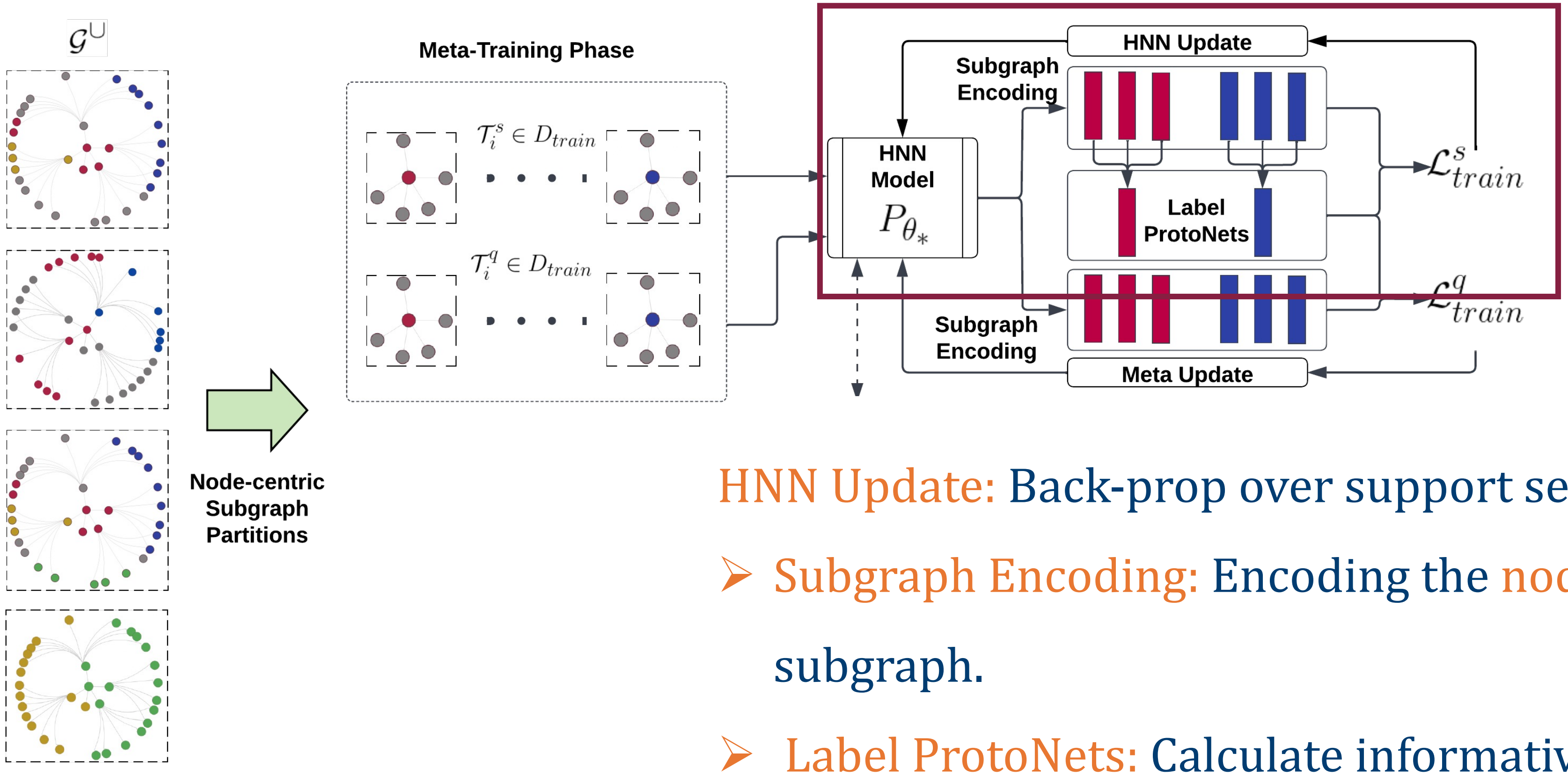
Meta-learning: Handling the Graph Sections



- In the case of Multiple Graphs or Disjoint Labels, we need to rely on Meta-learning for knowledge transfer between different subgraphs.
- In Meta-learning, we partition the problem into;
  - Meta-training: only training samples
  - Meta-testing: few training samples

Our Solution: H-GRAM

Model Architecture: Local HNN update

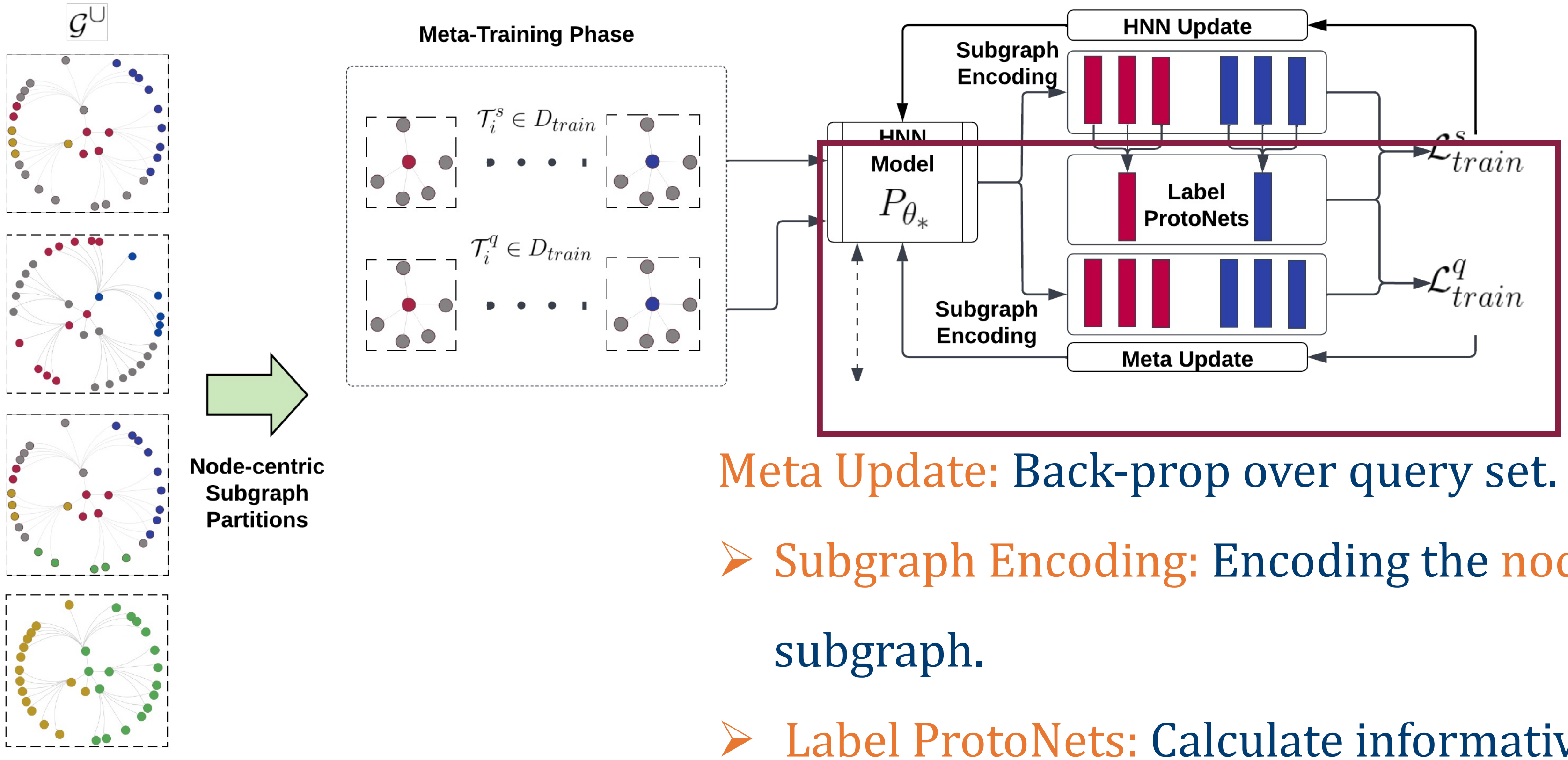


HNN Update: Back-prop over support set.

- Subgraph Encoding: Encoding the node-centric subgraph.
- Label ProtoNets: Calculate informative continuous label prototypes.

Our Solution: H-GRAM

Model Architecture: Meta-Update

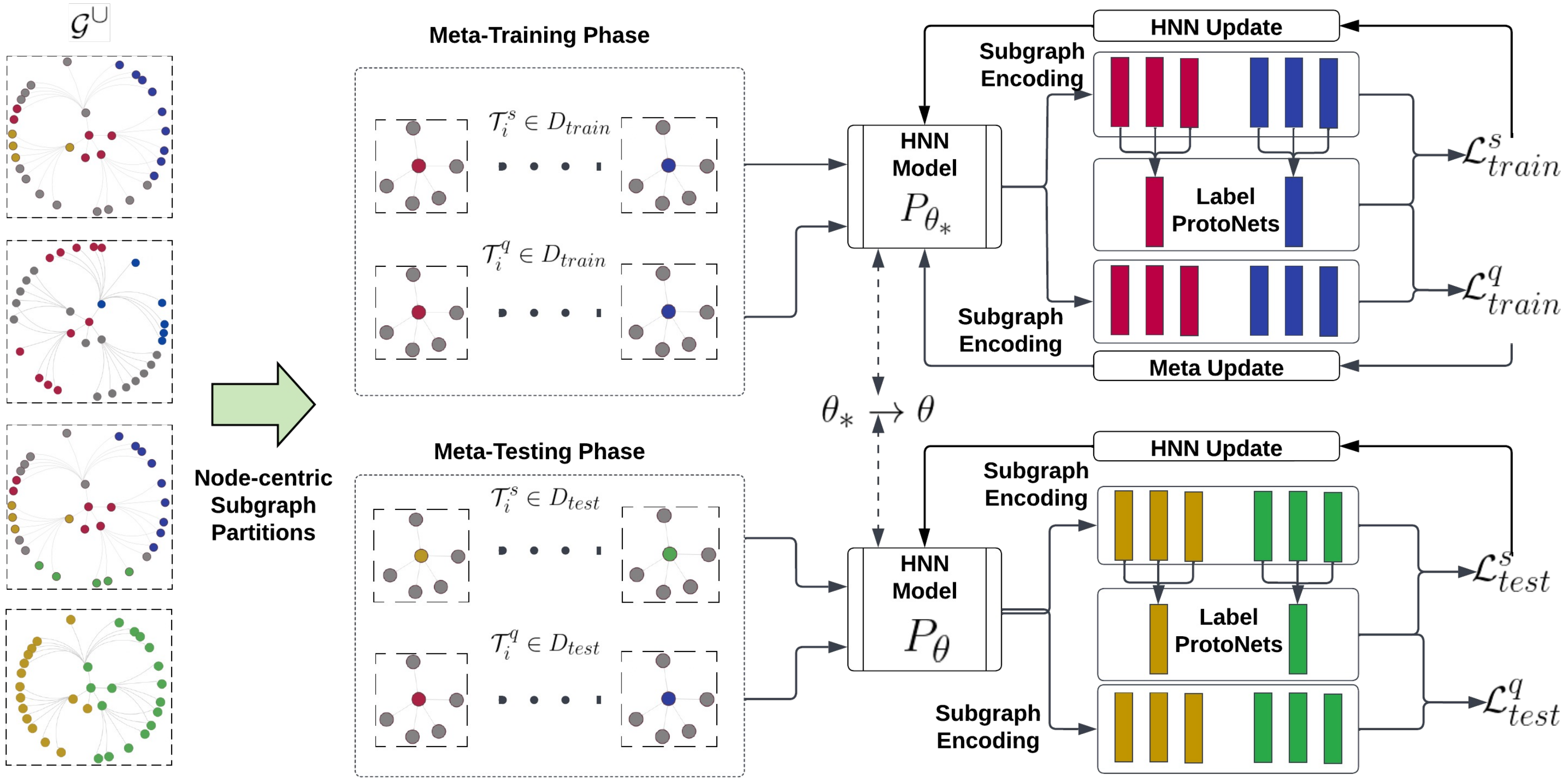


Meta Update: Back-prop over query set.

- Subgraph Encoding: Encoding the node-centric subgraph.
- Label ProtoNets: Calculate informative continuous label prototypes.
- Aggregate over a task and meta update.

Our Solution: H-GRAM

Model Architecture: Meta-learning



Meta Testing:

- HNN Updates: Few-shot over support set of test data.

Prediction over query set of test data for final evaluation.

Our Solution: H-GRAM

Evaluation: Experiments

1. Performance of H-GRAM
2. Challenging Few-shot Settings
3. Time Comparison and Ablation Study

Our Solution: H-GRAM

Dataset and Baselines

1. Datasets: Synthetic Cycle graph and Synthetic Barabási-Albert graph, ogbn-arxiv, Tissue-PPI, FirstMM-DB, Fold-PPI, Tree-of-Life, Cora, PubMed, and Citeseer.
2. Baselines: Meta-Graph, Meta-GNN, FS-GIN, FS-SGC, ProtoNet, MAML, HMLP, HGCN, and HAT.
3. Evaluation: Accuracy of Node Classification and Link Prediction

Our Solution: H-GRAM

Performance on Graph Tasks

Task	Node Classification		Node Classification		Node Classification		Node Classification			Link Prediction	
Setup	SG,DL		MG,SL		MG,DL		SG,DL	MG,SL	MG,DL	MG,SL	MG,SL
Dataset	Syn. Cycle	Syn. BA	Syn. Cycle	Syn. BA	Syn. Cycle	Syn. BA	ogbn-arxiv	Tissue-PPI	Fold-PPI	FirstMM-DB	Tree-of-Life
Meta-Graph	-	-	-	-	-	-	-	-	-	0.719	0.705
Meta-GNN	0.72	0.694	-	-	-	-	0.273	-	-	-	-
FS-GIN	0.684	0.749	-	-	-	-	0.336	-	-	-	-
FS-SGC	0.574	0.715	-	-	-	-	0.347	-	-	-	-
ProtoNet	0.821	0.858	0.282	0.657	0.749	0.866	0.372	0.546	0.382	0.779	0.697
MAML	0.842	0.848	0.511	0.726	0.653	0.844	0.389	0.745	0.482	0.758	0.719
G-META	0.872	0.867	0.542	0.734	0.767	0.867	0.451	0.768	0.561	0.784	0.722
H-GRAM	0.883	0.873	0.555	0.746	0.779	0.888	0.472	0.786	0.584	0.804	0.742

Accuracy of H-GRAM compared to Euclidean baselines on Node Classification and Link Prediction

Our Solution: H-GRAM

Summary

- Meta-learning helps in learning meta-information from local subgraphs and generalizing it over the global graph structure.
- H-GRAM shows improved performance on different graph tasks compared to both scalable Euclidean methods and non-scalable hyperbolic methods.
- H-GRAM parallelizes well in a multi-GPU setup, thus providing a scalable formulation of HNN models.

Thanks!

Any questions?

Find me at:

https://nurendra.com

nurendra@vt.edu