Graph Neural Networks for Adaptive Coarsening of Graphs

Marek Dědič[1][2],

[1]Czech Technical University in Prague

[2]Cisco Systems, Inc.

Machine learning seminar 2022, Thursday 24th November, 2022

marek@dedic.eu

Problem introduction

- Problem introduction
- Prior art - the HARP method
- The performance-complexity trade-off problem
- Adaptive prolongation
- Alternative approaches to coarsening
- Conclusion
- Direct adaptive coarsening
- Theoretical properties of graph coarsening
- Direct coarsening experiments
- Conclusion II

Motivation

- Network communication $\rightarrow$ cybersecurity

Motivation

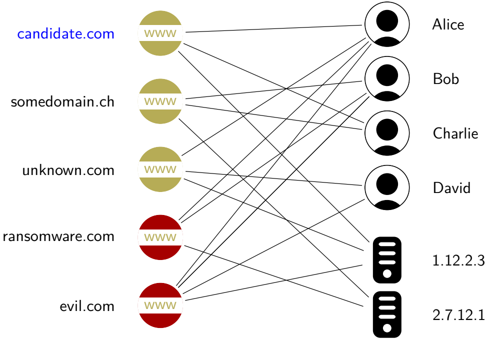
- Network communication → cybersecurity

time	*user*	*domain*	*ip*
t_1	alice	candidate.com	1.12.2.3
t_2	bob	somedomain.ch	2.7.12.1
t_3	bob	ransomware.com	2.7.12.1
t_4	david	unknown.com	1.12.2.3
t_5	bob	evil.com	1.12.2.3
t_6	charlie	candidate.com	1.12.2.3
t_7	bob	somedomain.ch	2.7.12.1
t_8	david	unknown.com	1.12.2.3
⋮	⋮	⋮	⋮

Motivation

- Network communication → cybersecurity

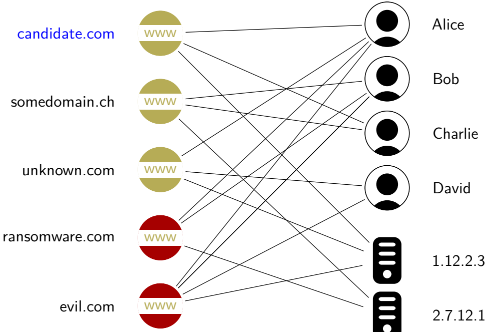
time	*user*	*domain*	*ip*
t_1	alice	candidate.com	1.12.2.3
t_2	bob	somedomain.ch	2.7.12.1
t_3	bob	ransomware.com	2.7.12.1
t_4	david	unknown.com	1.12.2.3
t_5	bob	evil.com	1.12.2.3
t_6	charlie	candidate.com	1.12.2.3
t_7	bob	somedomain.ch	2.7.12.1
t_8	david	unknown.com	1.12.2.3
⋮	⋮	⋮	⋮



Motivation

- Network communication → cybersecurity

time	*user*	*domain*	*ip*
t_1	alice	candidate.com	1.12.2.3
t_2	bob	somedomain.ch	2.7.12.1
t_3	bob	ransomware.com	2.7.12.1
t_4	david	unknown.com	1.12.2.3
t_5	bob	evil.com	1.12.2.3
t_6	charlie	candidate.com	1.12.2.3
t_7	bob	somedomain.ch	2.7.12.1
t_8	david	unknown.com	1.12.2.3
⋮	⋮	⋮	⋮

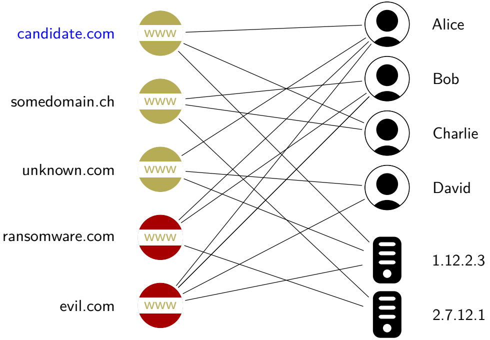


- Goal: identification of malicious domains

Motivation

- Network communication → cybersecurity

time	*user*	*domain*	*ip*
t_1	alice	candidate.com	1.12.2.3
t_2	bob	somedomain.ch	2.7.12.1
t_3	bob	ransomware.com	2.7.12.1
t_4	david	unknown.com	1.12.2.3
t_5	bob	evil.com	1.12.2.3
t_6	charlie	candidate.com	1.12.2.3
t_7	bob	somedomain.ch	2.7.12.1
t_8	david	unknown.com	1.12.2.3
⋮	⋮	⋮	⋮

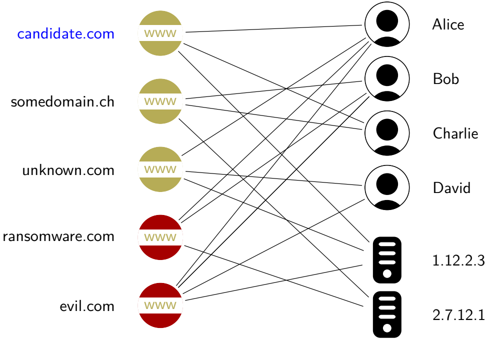


- Goal: identification of malicious domains
- Currently simple, scalable models
- State of the art: GNNs

Motivation

- Network communication → cybersecurity

time	*user*	*domain*	*ip*
t_1	alice	candidate.com	1.12.2.3
t_2	bob	somedomain.ch	2.7.12.1
t_3	bob	ransomware.com	2.7.12.1
t_4	david	unknown.com	1.12.2.3
t_5	bob	evil.com	1.12.2.3
t_6	charlie	candidate.com	1.12.2.3
t_7	bob	somedomain.ch	2.7.12.1
t_8	david	unknown.com	1.12.2.3
⋮	⋮	⋮	⋮

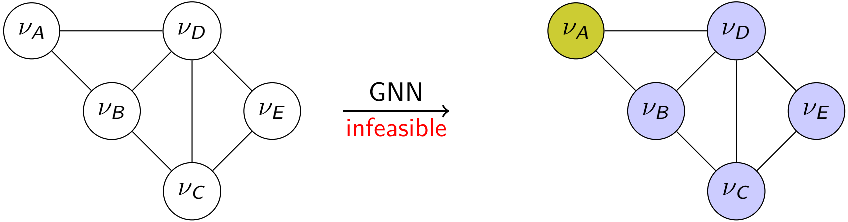


- Goal: identification of malicious domains
- Currently simple, scalable models
- State of the art: GNNs
- Problem – complexity required by GNNs on large graphs

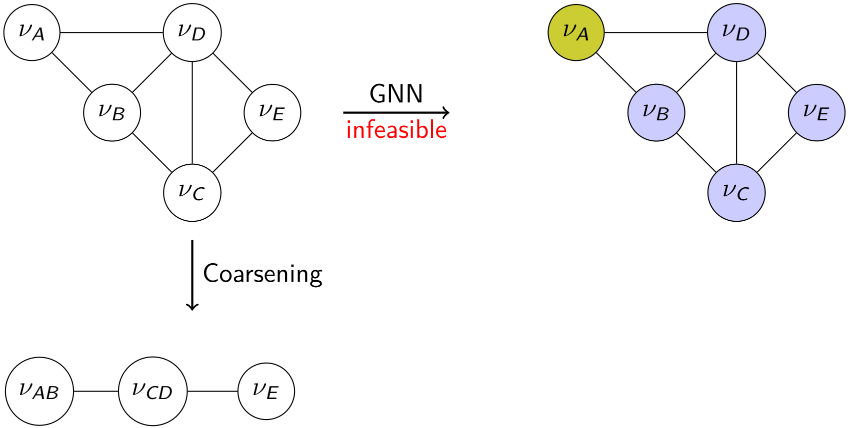
Problem introduction

- GNNS on big graphs – computationally costly and sometimes even unfeasible.
- Suggested solution: compressing the input graph
- How much can a graph be coarsened?
- Sometimes compressed data $\implies$ better performace

Main idea

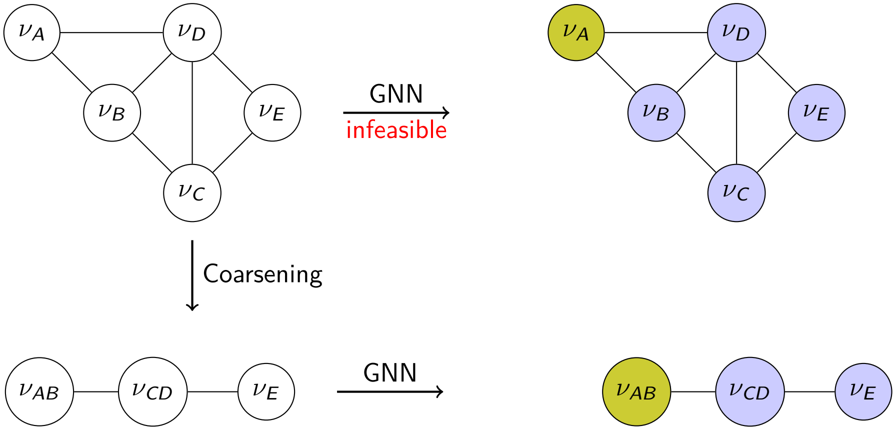


Main idea



ν_A
ν_D
ν_B
ν_E
ν_C
GNN
infeasible
ν_A
ν_D
ν_B
ν_E
ν_C
Coarsening
ν_{AB}
ν_{CD}
ν_E

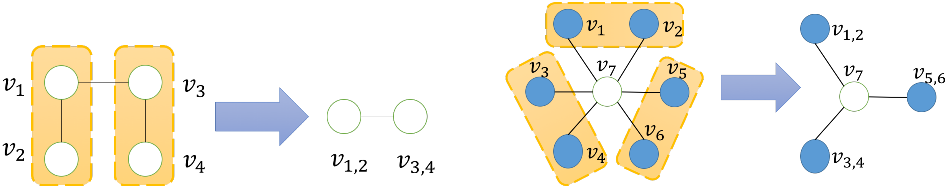
Main idea


ν_A
ν_D
ν_B
ν_E
ν_C
GNN
infeasible
Coarsening
ν_{AB}
ν_{CD}
ν_E
GNN

Prior art - the HARP method

HARP - learning on coarser graphs

- HARP - a method for pretraining on simplified graphs
- Simplified graphs as depicted bellow
- Embedding trained first on the coarser graph

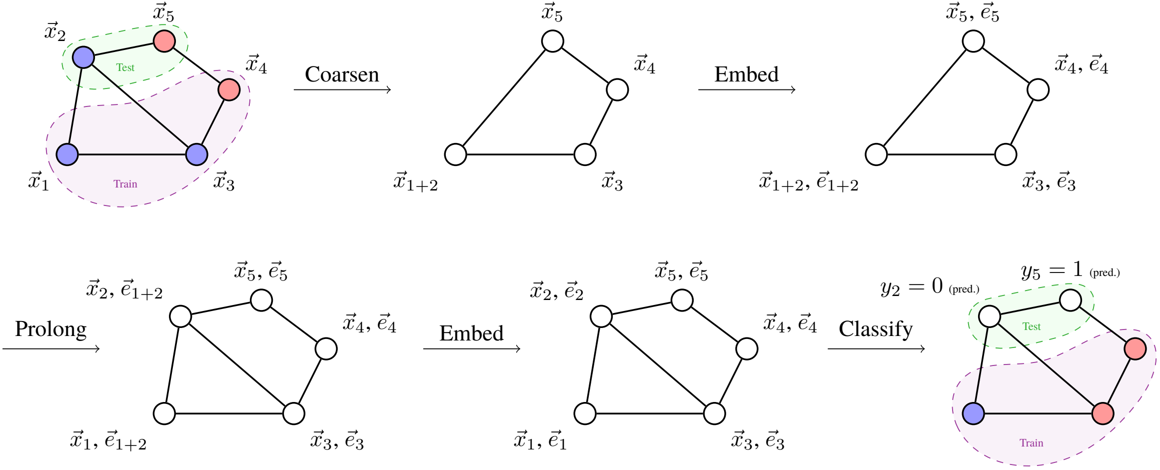


(a) Edge collapsing (b) Star collapsing

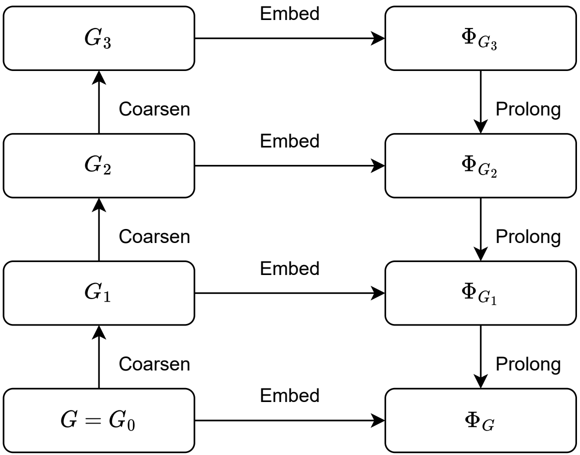
Figure: HARP coarsening algorithm. [1]

[1] Images from Chen et al., 2018.

HARP pipeline overview



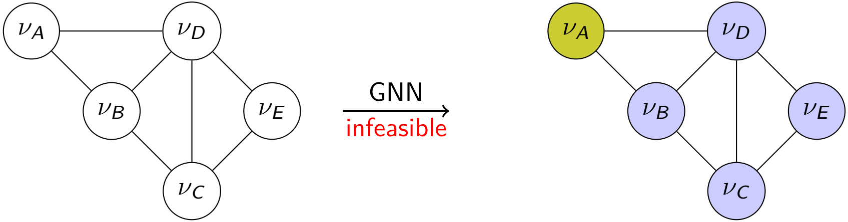
Deep HARP



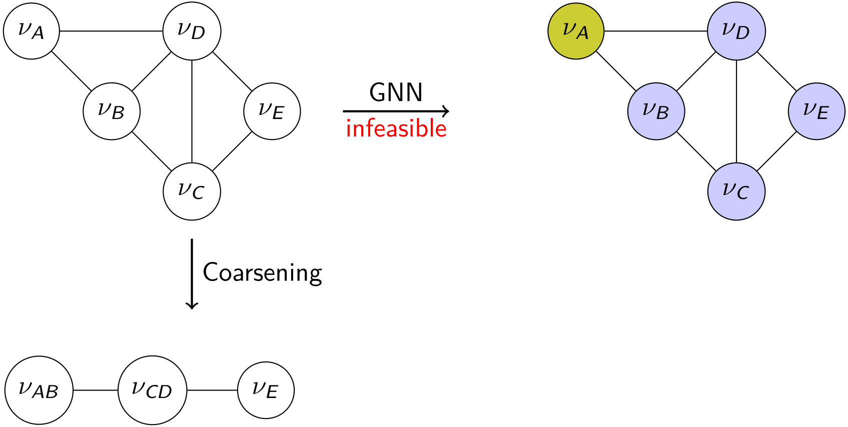
G_3
Embed
Φ_{G_3}
Coarsen
Prolong
G_2
Embed
Φ_{G_2}
Coarsen
Prolong
G_1
Embed
Φ_{G_1}
Coarsen
Prolong
$G = G_0$
Embed
Φ_G

The performance-complexity trade-off problem

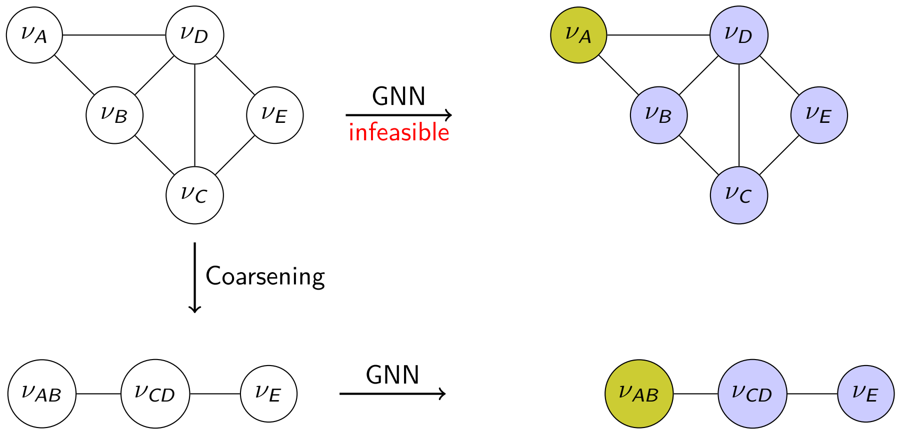
Main idea



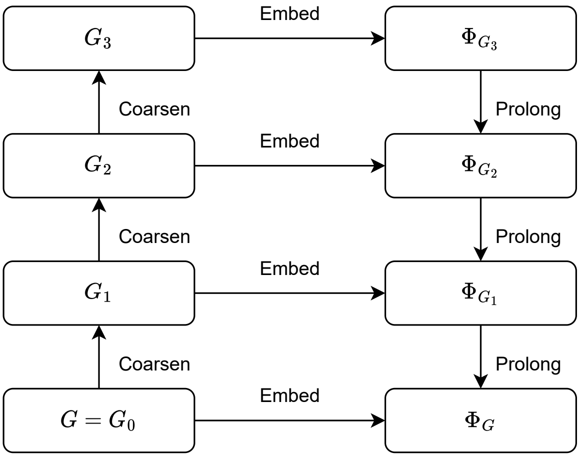
Main idea


ν_A
ν_D
ν_B
ν_E
ν_C
GNN
infeasible
ν_A
ν_D
ν_B
ν_E
ν_C
Coarsening
ν_{AB}
ν_{CD}
ν_E

Main idea

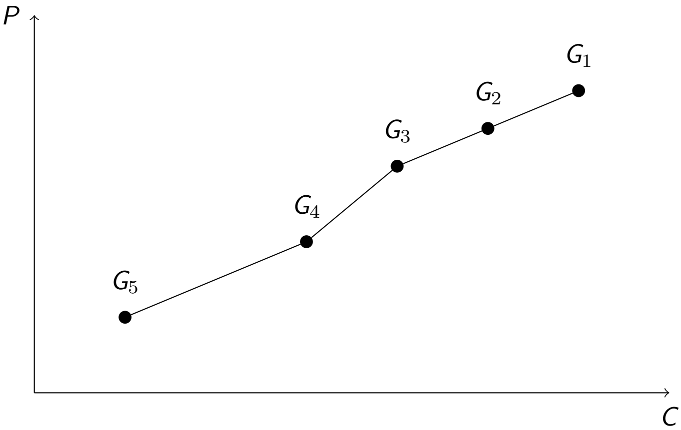

ν_A
ν_D
ν_B
ν_E
ν_C
GNN
infeasible
ν_A
ν_D
ν_B
ν_E
ν_C
Coarsening
ν_{AB}
ν_{CD}
ν_E
GNN
ν_{AB}
ν_{CD}
ν_E

Deep HARP

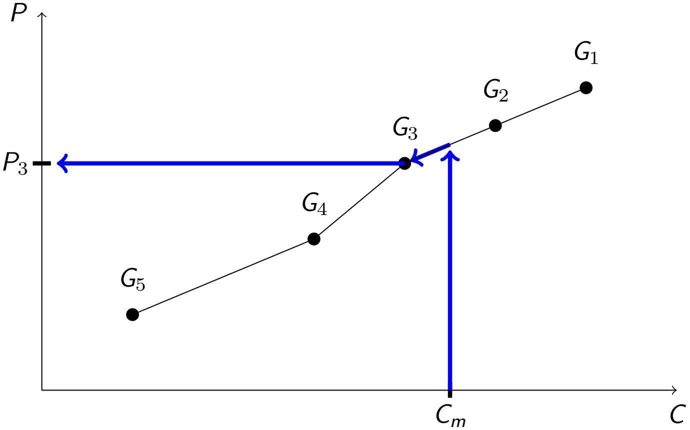


G_3
Embed
Φ_{G_3}
Coarsen
Prolong
G_2
Embed
Φ_{G_2}
Coarsen
Prolong
G_1
Embed
Φ_{G_1}
Coarsen
Prolong
$G = G_0$
Embed
Φ_G

Complexity performance trade-off

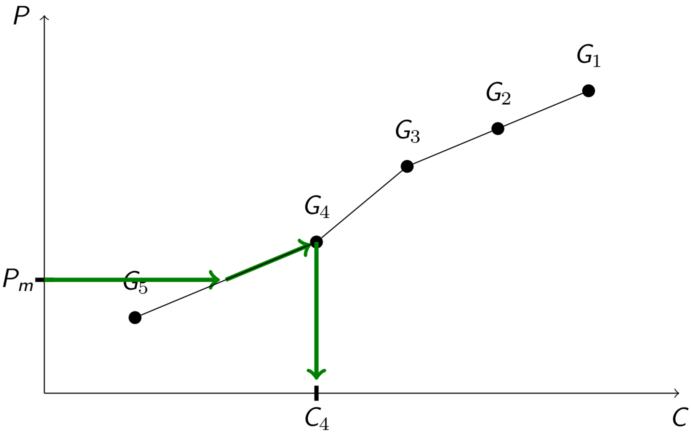


Complexity performance trade-off



Achieving best performance P_3 for complexity C_m.

Complexity performance trade-off



The least complexity (C_4) with performance at least P_m.

Performance & Complexity

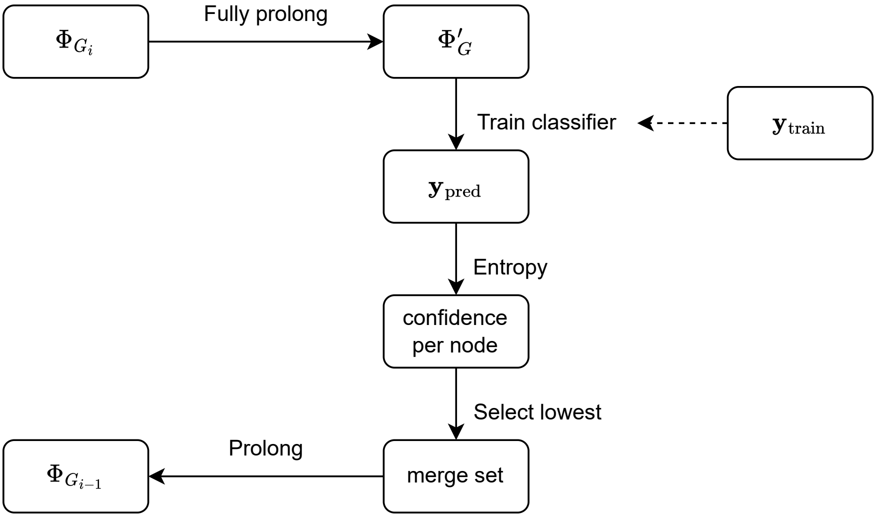
- Performance = classification accuracy
- Usually time & memory complexity
- In business, time translates to cost
  - More interpretable
  - Monthly budget \$100, AWS EKS costs \$0.33/hour
  - Therefore daily run can take 10 hours → select graph size accordingly
- For simplicity, complexity = node count

Adaptive prolongation

Adaptive prolongation

- Graph prolongation (de-coarsening) micro-stepping
- Guiding the prolongation based on local properties
- Possibly results in different "model resolution" in different parts of the graph
- Evaluation – Training a classifier in each (micro) step

Adaptive prolongation



Adaptive prolongation - experimental verification

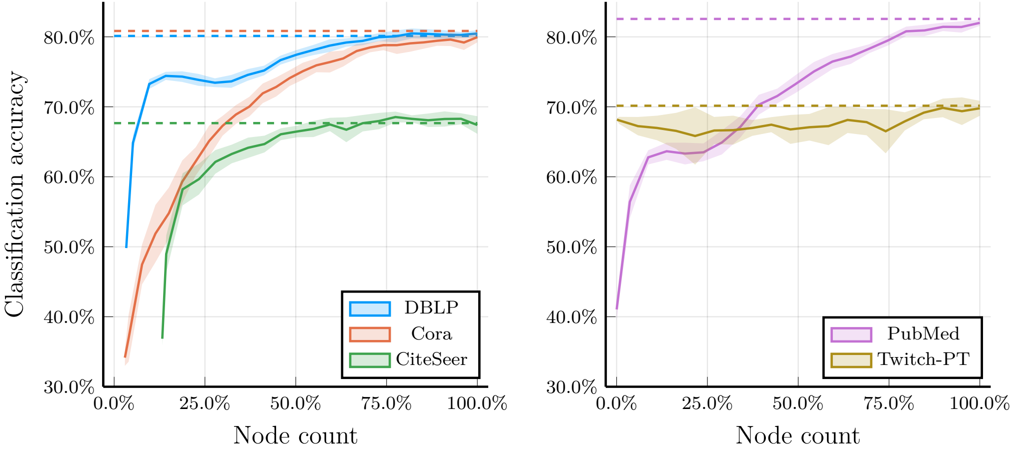
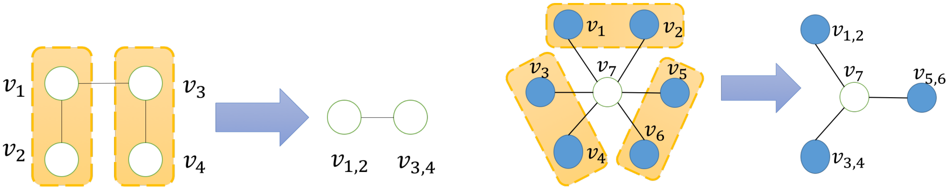


Figure: Prediction accuracy with guided prolongation.

Alternative approaches to coarsening

- Problem introduction
- Prior art - the HARP method
- The performance-complexity trade-off problem
- Adaptive prolongation
- **Alternative approaches to coarsening**
- Conclusion
- Direct adaptive coarsening
- Theoretical properties of graph coarsening
- Direct coarsening experiments
- Conclusion II



(a) Edge collapsing (b) Star collapsing

Figure: HARP coarsening algorithm. [2]

[2] Images from Chen et al., 2018.

Diffusion-based coarsening

- Creating a new edge matrix $\hat{\mathbf{S}}$
- Contracting edges $E\left(G_{\hat{\mathbf{S}}}\right) \cap E\left(G\right)$
- $\hat{\mathbf{S}} = \sum_{k=1}^{\infty} \theta_k \mathbf{T}^k$

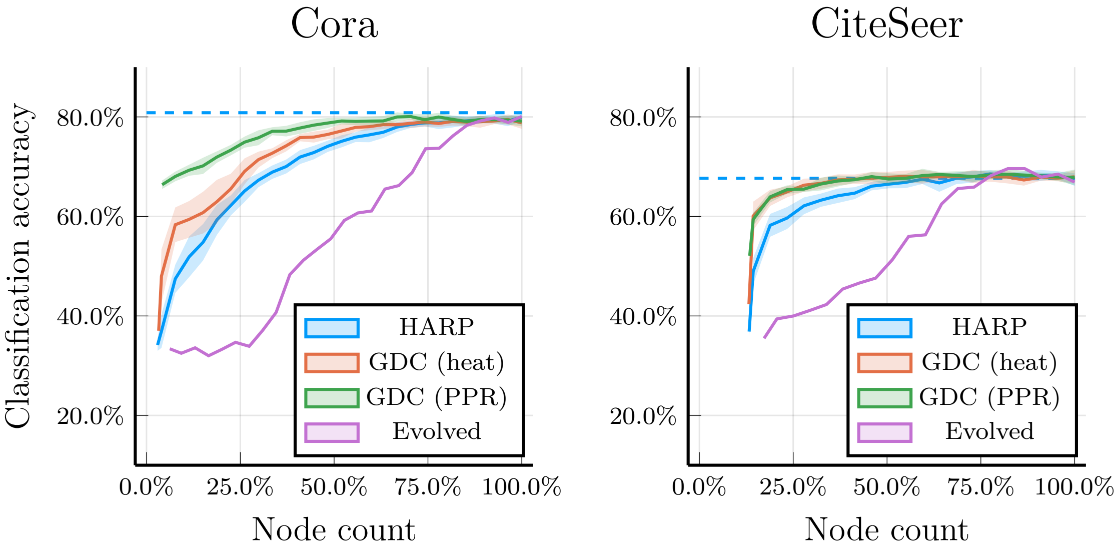
Diffusion-based coarsening

- $\hat{\mathbf{S}} = \sum_{k=1}^{\infty} \theta_k \mathbf{T}^k$
- Personalized PageRank
  - $\mathbf{T} = \mathbf{A}\mathbf{D}^{-1}$
  - $\theta_k = \alpha \left(1 - \alpha\right)^k$
- Heat kernel
  - $\mathbf{T} = \mathbf{A}\mathbf{D}^{-1}$
  - $\theta_k = e^{-t} \frac{t^k}{k!}$

Evolved coarsening

- A set of atomic coarsenings
- A coarsening candidate = sequence of atomic coarsenings
- Evolved using a simple genetic algorithm

Experimental comparison of coarsening approaches



Conclusion

Conclusion

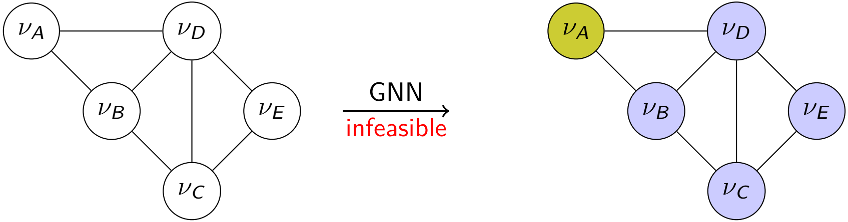
- Studying GNN behaviour while coarsening a graph
- The adaptive prolongation approach – a way of studying such behaviour in detail.
- 3 alternative coarsening strategies
- A new re-formalization of HARP, allowing for a more general approach to graph coarsening
- HARP as a framework for graph reduction

Future work

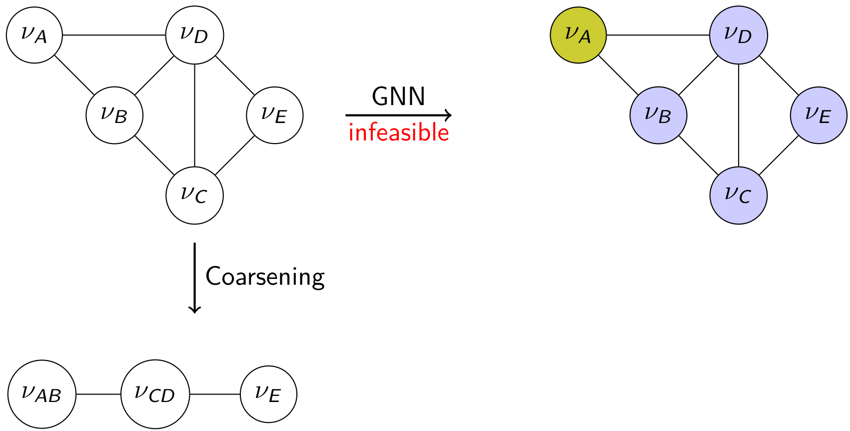
- Investigate more datasets
- Surrogate metric for adaptive prolongation
- More thorough exploration of evolved coarsenings

Direct adaptive coarsening

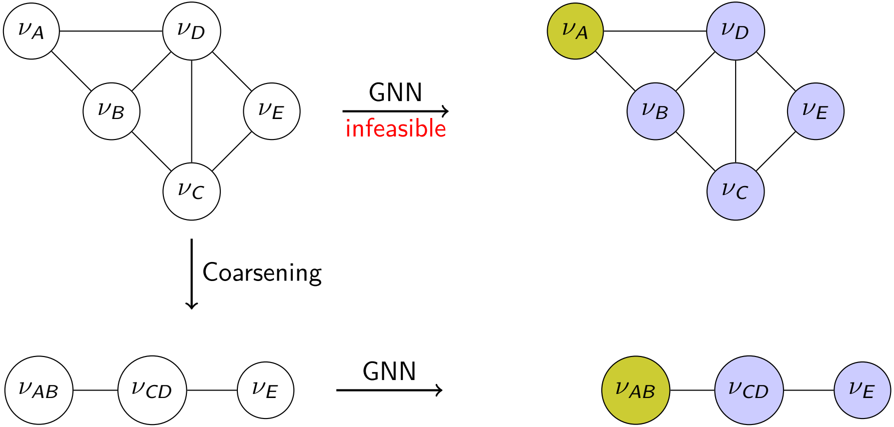
Main idea



Main idea

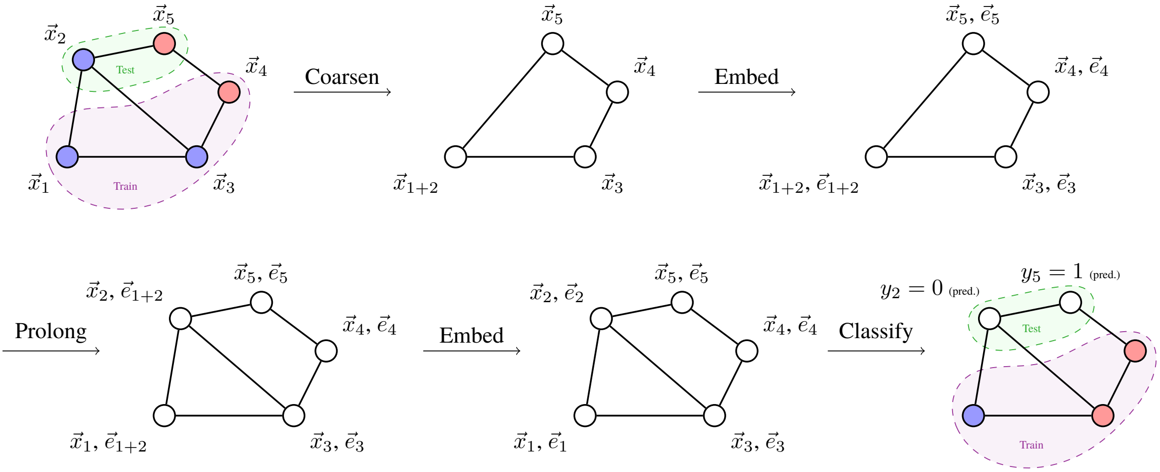


Main idea

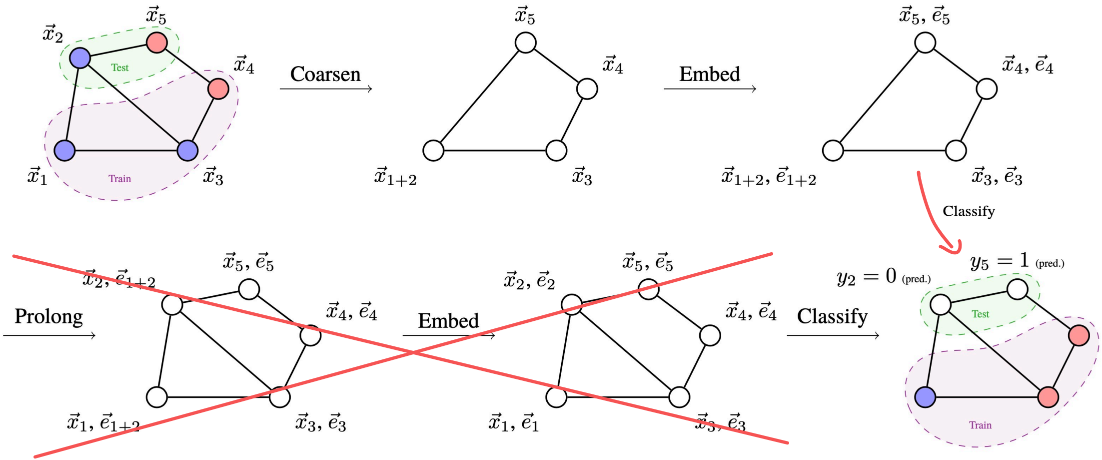


ν_A
ν_D
ν_B
ν_E
ν_C
GNN
infeasible
ν_A
ν_D
ν_B
ν_E
ν_C
Coarsening
ν_{AB}
ν_{CD}
ν_E
GNN
ν_{AB}
ν_{CD}
ν_E

HARP pipeline overview

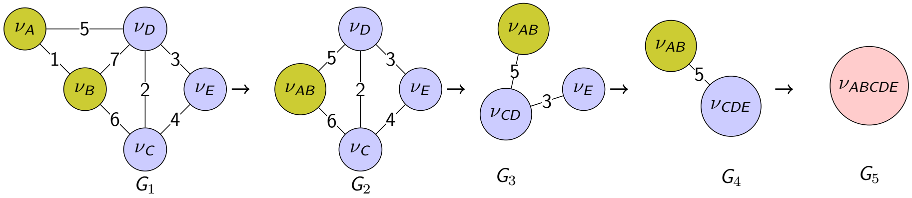


Direct pipeline



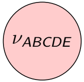
Details

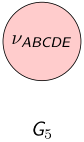
- Base model → posterior for all nodes
- Similarity measure
- Edge contraction → graph sequence
- GNN on each graph with label propagation



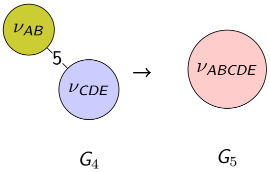
Theoretical properties of graph coarsening

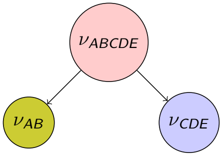
Edge contraction induced hierarchical tree



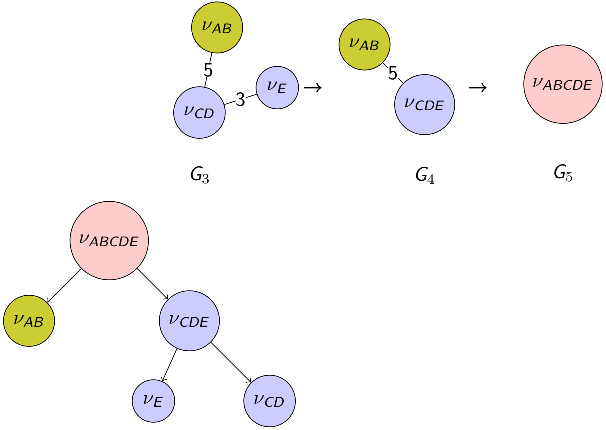


Edge contraction induced hierarchical tree

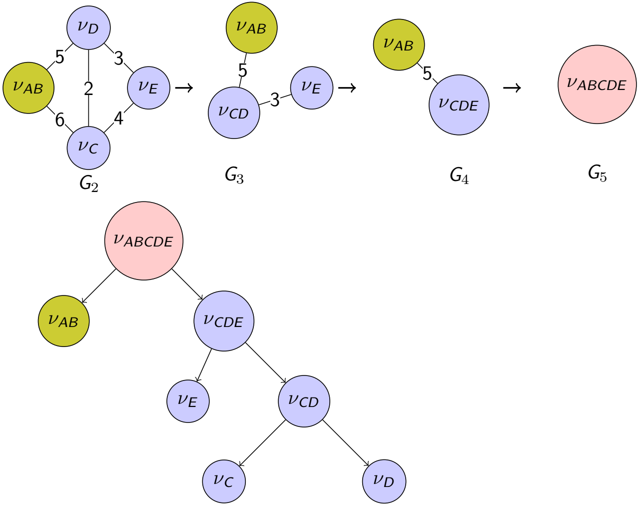




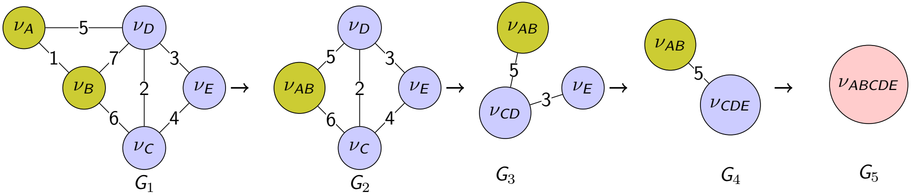
Edge contraction induced hierarchical tree

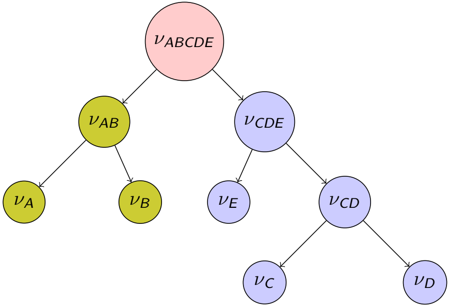


Edge contraction induced hierarchical tree



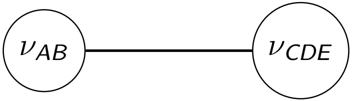
Edge contraction induced hierarchical tree





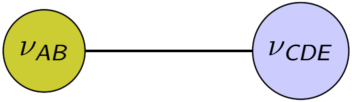
Shared prediction induced performance upper bound

Graph G_4 – the fourth step of the coarsening procedure



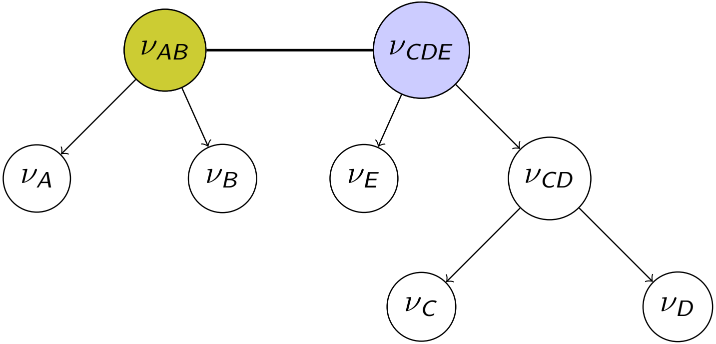
Shared prediction induced performance upper bound

A model prediction for nodes in G_4



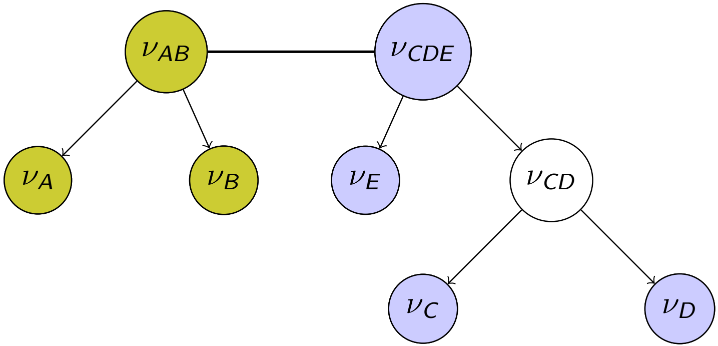
Shared prediction induced performance upper bound

Hierarchical tree corresponding to nodes in G_4



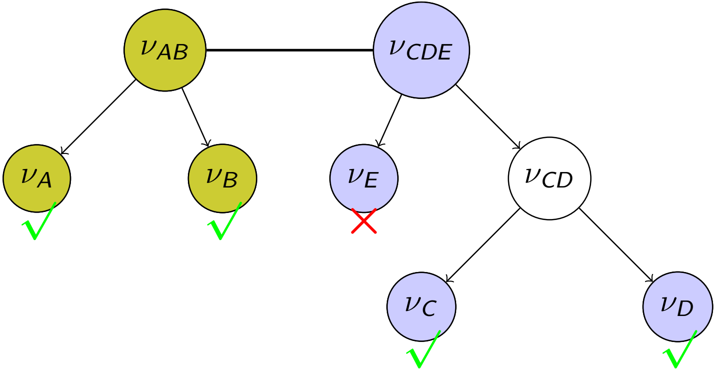
Shared prediction induced performance upper bound

Label refinement into nodes from the original graph G_1



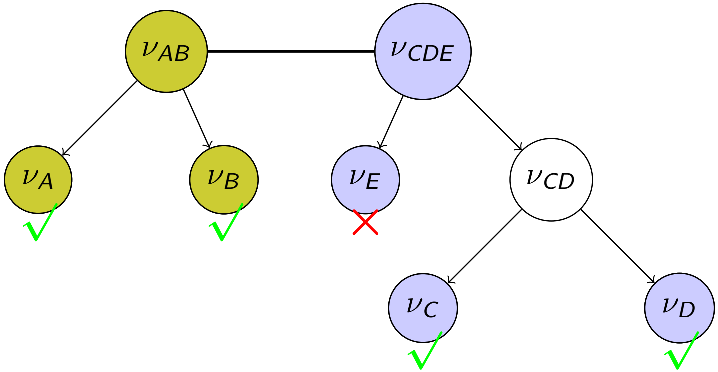
Shared prediction induced performance upper bound

Comparing refined prediction with true labels on $G_1 \rightarrow Acc = 0.8$.



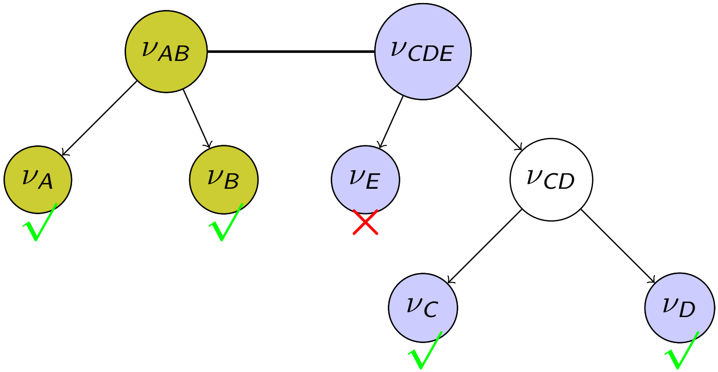
Shared prediction induced performance upper bound

Accuracy upper-bound for $G_4 \rightarrow$ Acc=0.8



Shared prediction induced performance upper bound

UB=1 until merging nodes with the same label

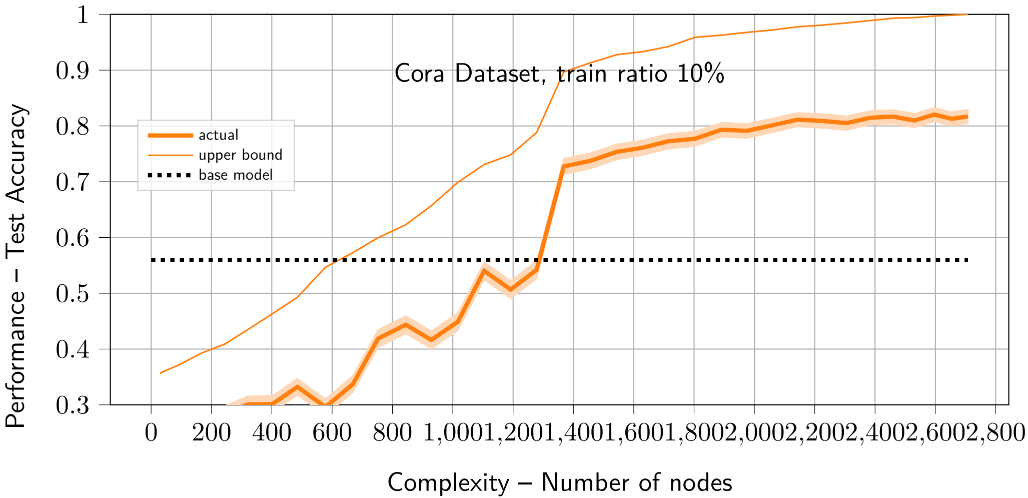


Direct coarsening experiments

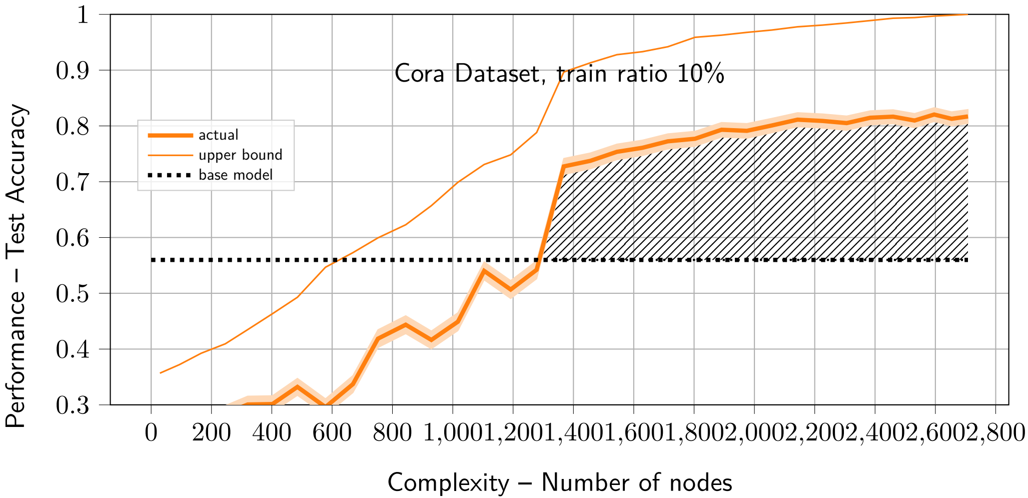
Experiment Setup

- Base model: logistic regression
- Similarity measure: KL divergence
- GNN: 2-layer GCN
- Dataset: Cora

Comparison of Performance Given by Probability Similarities



Comparison of Performance Given by Probability Similarities



Conclusion II

Conclusion & Future work

- Conclusions
  - Hierarchical tree as a byproduct
  - Computationally cheap upper bound
- Future Work
  - Run on more datasets
  - Run on Cisco datasets
  - Investigate the validity of many ad-hoc choices in the algorithm

Graph Neural Networks for Adaptive Coarsening of Graphs

Marek Dědič[1,2],

[1]Czech Technical University in Prague

[2]Cisco Systems, Inc.

Machine learning seminar 2022, Thursday 24th November, 2022