PRODIGY: Enabling In-context Learning Over Graphs

CS224W: Machine Learning with Graphs
Qian Huang, Stanford University
11/28

Input:

- hello
- hello
- hello

Training:

- 5
- h
- bonjour

Output:

- hello
- hello
- hello
Different Machine Learning Paradigms

Output
5  h  bonjour

Neural Networks

hello  hello  hello
Training

5  h  bonjour

hello
Finetuning
Different Machine Learning Paradigms

Input: hello, hello, hello

Training Neural Networks:
- Output: 5, h, bonjour

Finetuning:
- Output: bonjour

Few-shot prompting:
- Thanks -> merci
- Wall -> mur
- hello ->
In-context Learning

Performing a new task by “learning” from the input context w/o gradient update. Very powerful — we only need one foundation model directly answering all tasks now!
In-context Learning Over Graphs

Thanks -> merci
Wall -> mur

What is in-context learning over graphs?

Few-shot prompting over text
Today’s Plan

We formulate and enable **in-context learning over graphs**

- **Formulation**: An in-context learner for graphs should be able to solve novel tasks on novel graphs.
- **PRODIGY**:
  - **Prompt Graph representation**: represent the few-shot prompt for different graph tasks in the same input format, so that it can be consumed by one shared model
  - **Prompt Graph Inference**: in-context prediction GNN
  - **Pretraining**: generate diverse pretraining tasks in the format of PromptGraph
    - Neighbor Matching
    - MultiTask
Formulation
Graph Learning Tasks

What are the tasks on graphs?

Node classification

Link Prediction

Graph classification
In-context Learning Over Graphs

Thanks -> merci
Wall -> mur

What is in-context learning over graphs?

Few-shot prompting over text
In-context Learning Over Graphs: Link Prediction Example

An in-context learner for graphs should be able to solve novel tasks on novel graphs without gradient updates.

Graph $G$:

Prompt Examples $S$:
- Input $x$ to Label $y$
  - $(\circ, \triangle) \rightarrow \lozenge$
  - $(\circ, \triangle) \rightarrow \lozenge$
  - $(\circ, \triangle) \rightarrow \lozenge$
  - $(\circ, \triangle) \rightarrow \lozenge$

Queries $Q$:
- $(\circ, \triangle) \rightarrow \lozenge$ or $\lozenge$?

Thanks -> merci
Wall -> mur

Different tasks

Few-shot prompting over text

Different graphs

Different tasks

Few-shot prompting over graph (for link classification)
But, how to achieve this? Two Challenges:

1. How to represent the few-shot prompt for different graph tasks in the same input format, so that it can be consumed by one shared model?

2. How to pretrain a model that can solve any task in this format?
But, how to achieve this? Two Challenges:

1. How to represent the few-shot prompt for different graph tasks in the same input format, so that it can be consumed by one shared model?
   
   => **Prompt Graph**: represent each few-shot prompt over graph as a meta hierarchical graph

2. How to **pretrain** a model that can solve any task in this format?
   
   => **PGPretraining**: Pretrain a message passing model over self-supervised tasks in PromptGraph format with diverse underlying structures
Prompt Graph
Prompt Graph

**Prompt Graph** is a unified representation of few-shot prompts over graph for diverse tasks.
Step 1: Data Graph – Link Prediction

Data Graph contextualizes each input $x$ in the graph $G$ (e.g. by subgraph extraction)

- Task Graph
- Data Graphs
- Prompt examples $\mathcal{S}$
- Queries $Q$
Step 1: Data Graph – Link Prediction

Data Graph contextualizes each input $x$ in the graph $G$ (e.g. by subgraph extraction).

Graph $G$: Link Prediction

Prompt Examples $S$: Input $x$ Label $y$

Queries $Q$: $(\bullet, \triangle) \rightarrow \Diamond \text{ OR } ?$

Data Graphs

Prompt examples $S$

Queries $Q$
Step 1: Data Graph – Node Classification

Data Graph contextualizes each input $x$ in the graph $G$ (e.g., by subgraph extraction)
Step 1: Data Graph – Graph Classification

Data Graph contextualizes each input $x$ in the graph $G$ (e.g., by subgraph extraction).
Step 1: DataGraph Construction

DataGraph unifies input format:

- Use text feature to unify features over different datasets
- Use different input node set for different classification over different levels (nodes vs edge vs graph)
Step 2: Task Graph

*Task Graph* interconnects inputs and labels across examples to form context for queries.
Step 2: Task Graph – Link Prediction

The Task Graph interconnects inputs and labels across examples to form context for queries.
Step 2: Task Graph – Link Prediction

Task Graph interconnects inputs and labels across examples to form context for queries.
Step2: Task Graph – Link Prediction

**Task Graph** interconnects inputs and labels across examples to form context for queries.

- **Prompt examples:** bidirectional edges between data nodes and all label nodes.
Step 2: Task Graph – Link Prediction

Task Graph interconnects inputs and labels across examples to form context for queries.

- Prompt examples: bidirectional edges between data nodes and all label nodes.
Step 2: Task Graph – Link Prediction

**Task Graph** interconnects inputs and labels across examples to form context for queries

- **Prompt examples**: bidirectional edges between data nodes and all label nodes
- **Queries**: single directed edges from each label to each data node
Step 2: Task Graph – Link Prediction

**Task Graph** interconnects inputs and labels across examples to form context for queries

- **Prompt examples**: bidirectional edges between data nodes and all label nodes
- **Queries**: single directed edges from each label to each data node
Step2: Task Graph – Link Prediction

Task Graph interconnects inputs and labels across examples to form context for queries.
Step 2: Task Graph – Node Classification

Task Graph interconnects inputs and labels across examples to form context for queries
Step 2: Task Graph – Graph Classification

Task Graph interconnects inputs and labels across examples to form context for queries.
Flexibility of Task Graph

Task Graph unifies classification task format:

- Different number of classes are represented as different number of label nodes
- Different number of prompt examples (i.e. shots) and queries are represented as different number of data nodes as well as how they connect with label nodes
Prompt Graph for in-context learning

How to use PromptGraph for in-context learning?

Reflects what is the task using examples (commonalities and differences)

Captures all relevant information about the input

In-context learning over graph

inductive link prediction over hierarchical graph
PrompGraph Inference
In context prediction GNN
In context prediction GNN
In context prediction GNN

Hierarchical Message passing over PromptGraph
In context prediction GNN

Hierarchical Message passing over PromptGraph
- Data Graph Encoder
In context prediction GNN

Hierarchical Message passing over PromptGraph

- Data Graph Encoder
- Message Passing over Task Graph
In context prediction GNN

Hierarchical Message passing over PromptGraph

- Data Graph Encoder
- Message Passing over Task Graph
- Compute Logits
PRODIGY Pretraining
In-context Pretraining

We want to generate pretraining tasks in the format of **PromptGraph** and pretrain our model over them!
Pretraining Data Generation

Pretraining Data/Graph → Data Gen → Model → Optimize Loss
Pretraining Data Generation

Two stages:

- few-shot prompt generation
- convert to PromptGraph
Pretraining Data Generation

Two stages:

- **few-shot prompt generation**
- **convert to PromptGraph**

**Pretraining**

**Data/Graph**

- **Neighbor Matching**
- **MultiTask**

**Graph \( \mathcal{G} \):**

**Prompt Examples \( \mathcal{S} \):**
- Input \( x \)
- Label \( y \)
- Queries \( Q \):
  - \((\bullet, \Delta) \rightarrow \bigdiamond\)
  - \((\bullet, \Delta) \rightarrow \bigdiamond\)
  - \((\circ, \Delta) \rightarrow ?\)
  - \((\circ, \Delta) \rightarrow \bigdiamond\)

**PromptGraph with Augmentation**

Simply select multiple tasks for pretraining

- \( \rightarrow \) Supervised pretraining
Pretraining Data Generation

Two stages:

- **few-shot prompt generation**
  - self-supervised pretraining

**Pretraining**

- Data/Graph
  - Neighbor Matching
  - MultiTask

**Graph $G$:**

**Prompt Examples $\mathcal{S}$:**

- Input $x$
- Label $y$

- Queries $Q$:
  - $(\otimes, \Delta) 
  - (\otimes, \Delta) 
  - (\otimes, \Delta) 
  - (\otimes, \Delta) 
  - (\otimes, \Delta) 

**convert to PromptGraph**

**PromptGraph with Augmentation**

**Task Graph**

**Data Graphs**

**Prompt examples $\mathcal{S}$**

**Queries $Q$**
Self-supervised Task Example: Neighbor Matching

Idea: the task is to classify which neighborhood a node is in, where each neighborhood is defined by other nodes in it.
Pretraining Data Generation

Two stages:

- few-shot prompt generation
- convert to PromptGraph

Mixing different data generation pipeline gives more diverse pretraining data and better results!
In-context Pretraining Objectives

- Classification loss over generated tasks
- Attribute prediction loss: reconstruct corrupted features during DataGraph augmentation
Conclusion

PRODIGY enables **in-context learning over graphs** with a novel in-context task representation, Prompt Graph, and corresponding pretraining architecture and objectives.

- **Prompt Graph** provides a unified representation of few-shot prompts over graph for diverse tasks
- **Pretrain** a hierarchical message passing model over Prompt Graph enables in-context learning over unseen tasks on unseen graphs
- **Hard and diverse pretraining tasks** allow the model to keep scaling with more training data
Reliable Graph Learning with Guaranteed Uncertainty Estimates

Guest Lecture at Stanford CS 224W: Machine Learning with Graphs

Kexin Huang
PhD student at Stanford CS

@KexinHuang5
kexinhuang.com
kexinh@cs.stanford.edu
GNNs are powerful

Patient Network

Financial Network

Drug-discovery Network
Errors in critical applications

In high stake settings, errors have huge costs.

GNN on patient network:
- Wrong diagnosis

GNN on drug discovery network:
- Wasted experiment

GNN on financial network:
- Undetected Fraud
As the errors accumulate…

People stop trusting GNNs!

What should we do?

Can we know when will the model break down?

In another word, can we produce measure of uncertainty for model prediction?
If we know when the model will break down, ...

GNN on patient graph:

Low uncertainty → Right diagnosis

High uncertainty → Don’t use model prediction

How to produce a good uncertainty estimation?
After this lecture, you will learn...

- How to evaluate if an uncertainty estimation method is good?
  - What is reliability mathematically?

- How to produce uncertainty estimates with reliability guarantees?
  - Introduction to conformal prediction

- How to produce reliable uncertainty estimates for graphs?
  - State-of-the-art: conformalized GNNs
Plan for Today

- How to evaluate if an uncertainty estimation method is good?
- How to produce uncertainty estimates with reliability guarantees?
- How to produce reliable uncertainty estimates for graphs?
When will the model break down? quantifying uncertainty

<table>
<thead>
<tr>
<th>Test node</th>
<th>Test node</th>
<th>Test node</th>
<th>Test node</th>
<th>Test node</th>
<th>Test node</th>
<th>Test node</th>
<th>Test node</th>
<th>Test node</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
<td>$X_{n+1} \in \mathcal{D}_{\text{test}}$</td>
</tr>
</tbody>
</table>

**Point prediction**

$\hat{Y}_{n+1}$

**A range of plausible predictions**

$C(X_{n+1})$

**Classification**

1 2 3

**Prediction set**

\{2, 1\}

**Regression**

14.5

**Prediction interval**

14.1 [ ] 14.8
Key benefits of prediction
set/interval

A range of plausible predictions

\( C(X_{n+1}) \)

\{Disease A, Disease B, Disease C\}

- Offers a meaningful range of values for more informed decision-making.
- The size of set/interval measures level of uncertainty.
- Size is large - the model will break down
- Enable a rigorous notion of reliability
How to define if a prediction set/interval is good? - Coverage

Coverage := \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i \in \mathcal{D}_{\text{test}}} \mathbb{I}(Y_i \in C(X_i))

i.e. % of testing points $X_i$ where ground truth $Y_i$ falls into the prediction set/interval $C$
Can we control coverage?

\[
\text{Coverage} := \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i \in \mathcal{D}_{\text{test}}} \mathbb{I}(Y_i \in C(X_i))
\]

- A truly trustworthy ML model have 100% coverage.
- But prediction set/interval is predicted from the model, so 100% is not possible.
- Is it possible to control it with guarantees?
What if we can control the coverage with guarantees?

99% of predicted diagnosis set \textbf{provably} includes ground truth

Given a pre-defined target coverage level $1 - \alpha$, we reach this coverage level with guarantees.

Reliability & trust = coverage guarantees
Ablation features. We observe that it is also compatible with mini-batching techniques. See Appendix D.5. We conduct ablations in Table 4. While CF-GNN achieves marginal coverage, it is practically useful for practitioners that aim for a specified coverage in settings such as planning and selection.

Previous methods produce heuristic uncertainties but have no guarantees!
Plan for Today

- How to measure if an uncertainty estimation method is good?
- How to produce uncertainty estimates with reliability guarantees?
- How to produce reliable uncertainty estimates for graphs?
The promises of conformal prediction

- Theoretical guarantees on finite sample coverage validity
- Distribution-free
- Model-agnostic
- Post-hoc wrapper: no training modifications
Input of the data

- Training (+val)
- Calibration
- Testing

Pre-defined coverage level

\[ 1 - \alpha \]

ML model

Train set

Any test point

...
Step 1/4: define heuristics uncertainty

- Training (+val)
- Calibration
- Testing

Pre-defined coverage level $1 - \alpha$

Classification: softmax scores

$\hat{\mu}(x)(3)$
Model “confidence” about this instance having label 3

$\hat{\mu}(x)$
Step 2/4: non-conformity scores

\[ V : (\mathcal{X} \times \mathcal{Y}) \rightarrow \mathbb{R} \quad \text{Non-conformity score function} \]

\[ V(X, Y) \quad \text{How much label Y “conforms” to the prediction at X} \]

Larger \( V \): higher non-conformity, lower confidence
Smaller \( V \): lower non-conformity, more confidence

Let’s construct one for softmax score:

\[ V(X = x, Y = 3) = 1 - \hat{\mu}(x)_{(3)} \]

Low non-conformity score when softmax is high i.e. model is confident
Step 3/4: quantile computation

- Training (+val)
- Calibration
- Testing

Pre-defined coverage level

$1 - \alpha$

First calculate non-conformity score for each calibration data point

$\{V(X_i, Y_i)\}_{i=1}^{n}$
Step 3/4: quantile computation

\[ \hat{\eta} = \text{quantile}\left(\{V(X_i, Y_i)\}_{i=1}^n, (1 - \alpha)(1 + \frac{1}{n})\right) \]

- Takes the quantile of all non-conformity scores (with small correction)

1 - \alpha
Step 3/4: quantile computation

\[ \hat{\eta} = \text{quantile}\left\{V(X_i, Y_i)\right\}_{i=1}^{n}, (1 - \alpha)(1 + \frac{1}{n}) \]

- Takes the quantile of all non-conformity scores (with small correction)

- e.g. 95% of calibration truth label have non-conformity scores below \( \hat{\eta} \)
Step 4/4: prediction set/interval construction

\[ C(X_{n+1}) = \{ y \in \mathcal{Y} : V(X_{n+1}, y) \leq \hat{\eta} \} \]

The set of labels where non-conformity scores are smaller than \( \hat{\eta} \)

\[ \begin{align*}
V(X_{n+1}, Y = 1) \\
V(X_{n+1}, Y = 2) \\
V(X_{n+1}, Y = 3) \\
V(X_{n+1}, Y = 4) \\
V(X_{n+1}, Y = 5)
\end{align*} \]
Step 4/4: prediction set/interval construction

The set of labels where non-conformity scores are smaller than $\hat{\eta}$

$$C(X_{n+1}) = \{y \in \mathcal{Y} : V(X_{n+1}, y) \leq \hat{\eta}\}$$

$1 - \alpha$

$X_{n+1}$

$V(X_{n+1}, Y = 1) > \hat{\eta}$
$V(X_{n+1}, Y = 2) < \hat{\eta}$
$V(X_{n+1}, Y = 3) > \hat{\eta}$
$V(X_{n+1}, Y = 4) < \hat{\eta}$
$V(X_{n+1}, Y = 5) > \hat{\eta}$
In Summary

- Step 1/4: define heuristics for uncertainty
  - e.g. softmax class probabilities \( \hat{\mu}(x) \)
- Step 2/4: compute non-conformity scores for calibration sets
  - e.g. 1-softmax scores
- Step 3/4: compute quantile \( \hat{\eta} \) of non-conformity scores
- Step 4/4: Construct prediction set
  - The set of labels below \( \hat{\eta} \)
Similarly, for regression

**Step 1:** heuristic uncertainty

\[ V(X = x, Y = y) = \max(\hat{\mu}_{\alpha/2}(x) - y, y - \hat{\mu}_{1-\alpha/2}(x)) \]

**Step 2:** non-conformity score

- Training (+val)
- Calibration
- Testing

Pre-defined coverage level

\[ 1 - \alpha \]
Conformalized Quantile Regression

**Step 3:** computes the quantile over calibration data (standard and not shown)

**Step 4:** prediction interval construction

\[ C(X_{n+1}) = \{ y \in Y : V(X_{n+1}, y) \leq \hat{\eta} \} \]

\[ C(X_{n+1}) = [\hat{\mu}_{\alpha/2}(X_{n+1}) - \hat{\eta}, \hat{\mu}_{1-\alpha/2}(X_{n+1}) + \hat{\eta}] \]
Coverage guarantees

Theorem 1 [Vovk, Gammerman, and Saunders, 1999]
Given exchangeability between calibration set \(\{(X_i, Y_i)\}_{i=1}^n\) and \((X_{n+1}, Y_{n+1})\)

\[
P\{Y_{n+1} \in C(X_{n+1})\} \geq 1 - \alpha
\]

The probability of the ground truth falls into the prediction set is \(1 - \alpha\)

What is exchangeability?
Assumption: Exchangeability

Exchangeability between calibration set and test point

\[ (X_1, Y_1), \cdots, (X_n, Y_n), (X_{n+1}, Y_{n+1}) \]

\begin{align*}
\text{calibration set} & \quad \text{Any test point} \\
\end{align*}

Denote \( Z_i = (X_i, Y_i) \)

Given any permutation \( \pi \) of \( \{1, \cdots, n + 1\} \), it holds that

\[ P((Z_{\pi(1)}, \ldots, Z_{\pi(n+1)}) = (z_1, \ldots, z_{n+1})) = P((Z_1, \ldots, Z_{n+1}) = (z_1, \ldots, z_{n+1})) \]

Draw 3 colored balls from a bag,

\[ P( \color{blue}{\bullet} \color{red}{\bullet} \color{orange}{\bullet} ) = P( \color{orange}{\bullet} \color{blue}{\bullet} \color{red}{\bullet} ) \]
Conformal Prediction have been applied to vision, NLP, etc.

In computer vision, NLP problems, the **calibration** and **test points** are **i.i.d** (independent, and identically distributed).

Note: i.i.d implies exchangeability whereas the reverse is not true.
Plan for Today

- How to measure if an uncertainty estimation method is good?
- How to produce uncertainty estimates with reliability guarantees?
- How to produce reliable uncertainty estimates for graphs?
However, does exchangeability hold for graph structured data?

- Dependencies between testing and calibration nodes. (i.e., not i.i.d).
- Message-passing during training includes calibration and test nodes.

Previously, \( V(X, Y; \{Z_v\}_{v \in \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}}}) \)

Now, \( V(X, Y; \{Z_v\}_{v \in \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}}}, \{X_v\}_{v \in \mathcal{D}_{\text{calib}} \cup \mathcal{D}_{\text{test}}}, V, E) \)

Calib & test seen in training

Graph dependencies
Theorem 1: in transductive node-level prediction problem under random data split, graph exchangeability holds given permutation invariance.

Most popular GNNs are permutation invariant!

Validity of coverage guarantees for GNNs.

Kexin Huang, Ying Jin, Emmanuel Candès, Jure Leskovec. Uncertainty Quantification over Graph with Conformalized Graph Neural Network. NeurIPS 2023, spotlight.
Okay, we have coverage, is that enough?

- No! An arbitrarily large prediction set ensures coverage validity but is practically useless

If there are 5 classes:

\[ \{1, 2, 3, 4, 5\} \]

100% coverage!

-10,000

+10,000

Both are not usable!
How to define if a prediction set/interval is good? - Efficiency

- Inefficiency calculates the length of the prediction set size

\[
\text{Inefficiency} := \frac{1}{|D_{\text{test}}|} \sum_{i \in D_{\text{test}}} |C(X_i)|
\]

\[
\{2, 1\} \quad \text{vs} \quad \{2, 1, 8, 3\}
\]

\[
14.1 \quad \text{vs} \quad 13.1
\]

\[
14.8 \quad \text{vs} \quad 16.8
\]

- Of course, while ensuring coverage guarantees!
How to improve efficiency?

Key observation: GNNs prediction scores are not optimized for conformal efficiency.
GNN prediction scores are shown to be biased for uncertainty quantification.

GNN prediction scores are under-confident.

Can we learn to update the scores to make it conformal aware?

- A post-hoc step (not affect training!)

\[ \hat{\mu}(X_i) \]

or

\[ \tilde{\mu}(X_i) \]
How to adjust the prediction scores for GNNs?

- Inefficiency has a topological root?
- Could we use network structure to improve efficiency?
Key idea: use another GNN over prediction scores

- Note: the input node embedding is prediction scores from the base GNN.
- Asking around the neighbors on how to update the prediction scores to maximize efficiency.
How do we update the correction GNN?

**Key insight:** design a loss to approximate efficiency and directly learn to optimize over it.

This requires us to make the downstream conformal step differentiable.

\[ \hat{\mu}(X_i) \]
Differentiable conformal proxy

Step 1: define heuristic uncertainty
Step 2: non-conformity scores
Step 3: quantile computation
Step 4: prediction set/interval construction

\[ V(X, Y) = 1 - \tilde{\mu}(X)(Y) \]

Differentiable!

\[ V(X, Y) = \max(\tilde{\mu}_{\alpha/2}(X) - Y, Y - \tilde{\mu}_{1-\alpha/2}(X)) \]
Differentiable conformal proxy

Step 1: define heuristic uncertainty
Step 2: non-conformity scores

\[ \mathcal{V}_{\text{cor-calib}} \rightarrow \text{GNN}_\vartheta \rightarrow \begin{array}{c} V(X_1, Y_1) \\ V(X_2, Y_2) \\ \cdots \\ V(X_M, Y_M) \end{array} \xrightarrow{\text{DiffQuantile} \left( \{V(X_i, Y_i)\}, \hat{\alpha} \right)} \hat{\eta} \]

Step 3: quantile computation
Step 4: prediction set/interval construction

\[ \hat{q} \]

Smooth quantile operator
(e.g. torch.quantile)
Differentiable conformal proxy

Step 1: define heuristic uncertainty
Step 2: non-conformity scores
Step 3: quantile computation
Step 4: prediction set/interval construction

\[ \nabla_\theta \mathcal{L}_{\text{Ineff}} \]

Not differentiable

Inefficiency := \[ \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i \in \mathcal{D}_{\text{test}}} |C(X_i)| \]

\[ |C(X_{n+1})| = |\{ y \in \mathcal{Y} : V(X_{n+1}, y) \leq \hat{\eta} \}| \]
Differentiable conformal proxy

Step 1: define heuristic uncertainty
Step 2: non-conformity scores
Step 3: quantile computation
Step 4: prediction set/interval construction

Prediction Set Size Proxy

\[ |C(X_{n+1})| = |\{y \in \mathcal{Y} : V(X_{n+1}, y) \leq \hat{\eta}\}| \]

\[ V(X_i, k) \leq \hat{\eta} \rightarrow 1 \]
\[ V(X_i, k) > \hat{\eta} \rightarrow 0 \]

Use smooth indicator function:

\[ \sigma((\hat{\eta} - V(X_i, k))/\tau) \]
Differentiable conformal proxy

Step 1: define heuristic uncertainty
Step 2: non-conformity scores
Step 3: quantile computation
Step 4: prediction set/interval construction

Prediction Set Size Proxy

\[ \mathcal{L}_{\text{Ineff}} = \frac{1}{N} \sum_{i \in \mathcal{V}_{ct}} \sum_{k \in \mathcal{Y}} \sigma((\hat{\eta} - V(X_i, k))/\tau) \]

\[
\begin{align*}
V(X_i, k) > \hat{\eta} & \quad \text{and} \quad V(X_i, k) \leq \hat{\eta} \\
\hat{\eta} - V(X_i, k) & \quad \text{and} \quad \hat{\eta} - V(X_i, k)
\end{align*}
\]
Differentiable conformal proxy

Step 1: define heuristic uncertainty
Step 2: non-conformity scores
Step 3: quantile computation
Step 4: prediction set/interval construction

**Prediction Interval Length Proxy**

\[
L_{\text{Ineff}} = \frac{1}{N} \sum_{i \in V_{ct}} (\tilde{\mu}_{1-\alpha/2}(X_i) + \hat{\eta}) - (\tilde{\mu}_{\alpha/2}(X_i) - \hat{\eta})
\]

\[
\tilde{\mu}_{1-\alpha/2}(X_i) \quad \tilde{\mu}_{\alpha/2}(X_i)
\]

\[
\hat{\eta} \quad \hat{\eta}
\]

Prediction Interval Length
Will the correction step affect coverage guarantees?
  - Because the update step is also a permutation invariant GNN.
  - Based on theorem 1, it still achieves coverage guarantees.
Table 1: CF-GNN achieves empirical coverage guarantee.

<table>
<thead>
<tr>
<th>Task</th>
<th>UQ Model</th>
<th>Cora</th>
<th>DBLP</th>
<th>CiteSeer</th>
<th>PubMed</th>
<th>Computers</th>
<th>Covered?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node classif.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temp. Scale.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vector Scale.</td>
<td>0.946±.003</td>
<td>0.920±.009</td>
<td>0.952±.004</td>
<td>0.899±.002</td>
<td>0.929±.002</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Ensemble TS</td>
<td>0.944±.004</td>
<td>0.921±.009</td>
<td>0.951±.004</td>
<td>0.899±.003</td>
<td>0.932±.002</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>CaGCN</td>
<td>0.947±.003</td>
<td>0.920±.008</td>
<td>0.953±.003</td>
<td>0.899±.002</td>
<td>0.930±.002</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>GATS</td>
<td>0.939±.005</td>
<td>0.922±.004</td>
<td>0.949±.005</td>
<td>0.898±.003</td>
<td>0.926±.003</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>CF-GNN</td>
<td>0.952±.001</td>
<td>0.952±.001</td>
<td>0.953±.001</td>
<td>0.953±.001</td>
<td>0.952±.001</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>UQ Model</th>
<th>Anaheim</th>
<th>Chicago</th>
<th>Education</th>
<th>Election</th>
<th>Twitch</th>
<th>Covered?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node regress.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MC dropout</td>
<td>0.553±.022</td>
<td>0.427±.015</td>
<td>0.423±.013</td>
<td>0.417±.008</td>
<td>0.448±.017</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>BayesianNN</td>
<td>0.967±.001</td>
<td>0.955±.003</td>
<td>0.957±.002</td>
<td>0.958±.009</td>
<td>0.923±.006</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>CF-GNN</td>
<td>0.957±.003</td>
<td>0.954±.002</td>
<td>0.951±.001</td>
<td>0.950±.001</td>
<td>0.954±.001</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
CF-GNN enables drastic efficiency gain

<table>
<thead>
<tr>
<th>Task</th>
<th>Dataset</th>
<th>CP $\rightarrow$ CF-GNN</th>
<th>Node regress.</th>
<th>Dataset</th>
<th>CP $\rightarrow$ CF-GNN</th>
<th>Average Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node clasif.</td>
<td>Cora</td>
<td>3.80±2.8 $\rightarrow$ -53.61% $\rightarrow$ 1.76±.27</td>
<td>Anaheim</td>
<td>2.89±.39 $\rightarrow$ -25.00% $\rightarrow$ 2.17±.11</td>
<td>-53.75%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DBLP</td>
<td>2.43±.03 $\rightarrow$ -49.13% $\rightarrow$ 1.23±.01</td>
<td>Chicago</td>
<td>2.05±.07 $\rightarrow$ -0.48% $\rightarrow$ 2.04±.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CiteSeer</td>
<td>3.86±.11 $\rightarrow$ -74.27% $\rightarrow$ 0.99±.02</td>
<td>Education</td>
<td>2.56±.02 $\rightarrow$ -5.07% $\rightarrow$ 2.43±.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PubMed</td>
<td>1.60±.02 $\rightarrow$ -19.05% $\rightarrow$ 1.29±.03</td>
<td>Election</td>
<td>0.90±.01 $\rightarrow$ +0.21% $\rightarrow$ 0.90±.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computers</td>
<td>3.56±.13 $\rightarrow$ -49.05% $\rightarrow$ 1.81±.12</td>
<td>Income</td>
<td>2.51±.12 $\rightarrow$ -4.58% $\rightarrow$ 2.40±.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Photo</td>
<td>3.79±.13 $\rightarrow$ -56.28% $\rightarrow$ 1.66±.21</td>
<td>Unemploy</td>
<td>2.72±.03 $\rightarrow$ -10.83% $\rightarrow$ 2.43±.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS</td>
<td>7.79±.29 $\rightarrow$ -62.16% $\rightarrow$ 2.95±.49</td>
<td>Twitch</td>
<td>2.43±.10 $\rightarrow$ -1.36% $\rightarrow$ 2.39±.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physics</td>
<td>3.11±.07 $\rightarrow$ -62.81% $\rightarrow$ 1.16±.13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average Improvement: -6.73%

More results on conditional coverage, sensitivity analysis, other GNNs, etc. in the paper!
Lots of exciting follow-ups…

- Within the last 6 months
  - Extension to link prediction setting
  - Extension to non-uniform split
  - Extension to inductive setting
  - Extension to edge exchangeability
  - ……
Thank you!

- How to measure if an uncertainty estimation method is good?
- How to produce uncertainty estimates with reliability guarantees?
- How to produce reliable uncertainty estimates for graphs?