Graph Agreement Models for Semi-Supervised Learning

Carnegie Mellon University

Google

Otilia Stretcu
Carnegie Mellon University

Krishnamurthy Viswanathan
Google Research

Dana Movshovitz-Attias
Google Research

Emmanouil Antonios Platanios
Carnegie Mellon University

Andrew Tomkins
Google Research

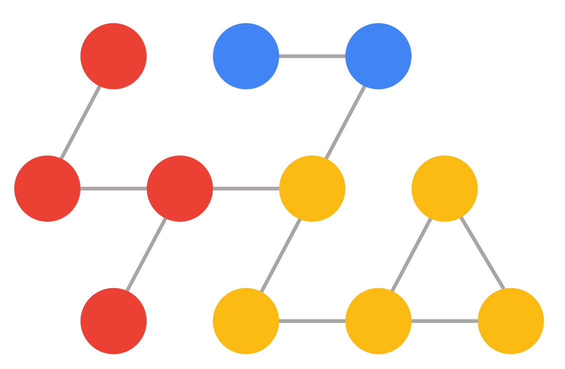
Sujith Ravi
Google Research

Problem

Consider the task of classifying a set of samples based on their features.

In some cases, we also have a graph connecting the samples.

Our goal is to learn a function that classifies these samples.



APPLICATIONS

- Document Classification
- Image Classification
- Intent Classification (e.g., for Smart Reply)
- User Classification (e.g., in Social Networks)

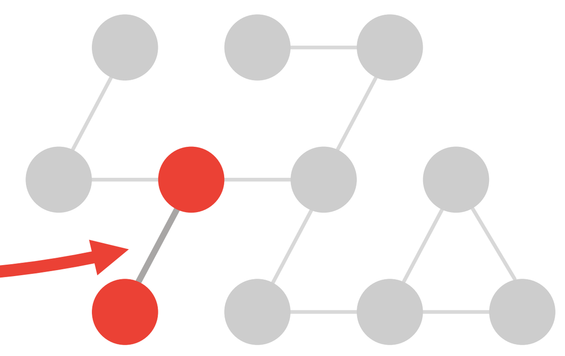
WHERE DO GRAPHS COME FROM?

Natural	Constructed
Wikipedia hyperlinks	Embedding similarity between documents, images, etc.
Publication citations	
Social network relationships	Structural similarity (e.g., dependency parse similarity)

ASSUMPTION

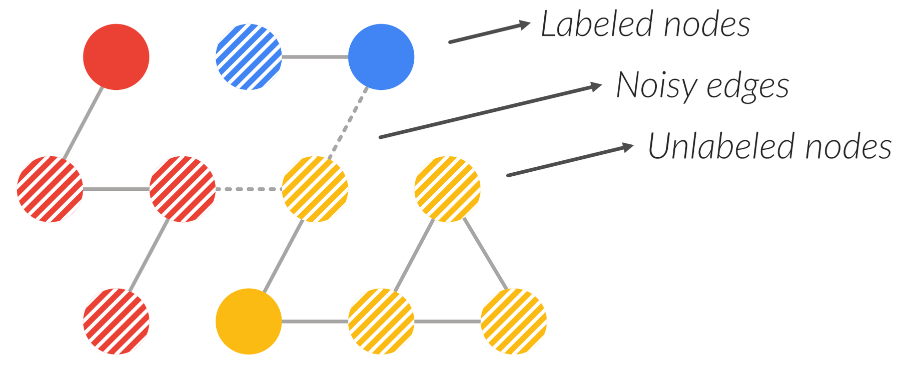
Many node classification methods rely on the following key assumption:

Connected nodes likely belong to the same class!



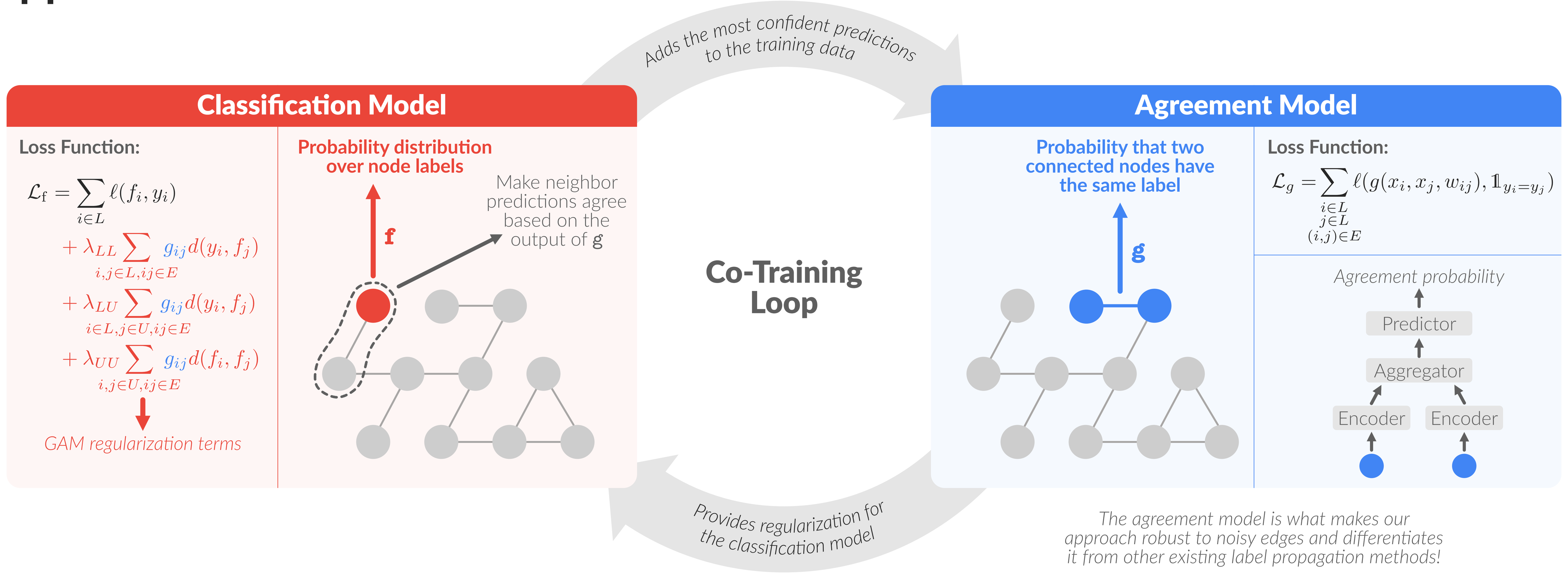
However, in practice, graphs are noisy!

CHALLENGES



1. Limited Supervision
   Only a small number of nodes are labeled.
2. Noisy Edges
   Some edges may violate our key assumption.
3. Scale
   Practical problems often involve massive graphs.

Approach

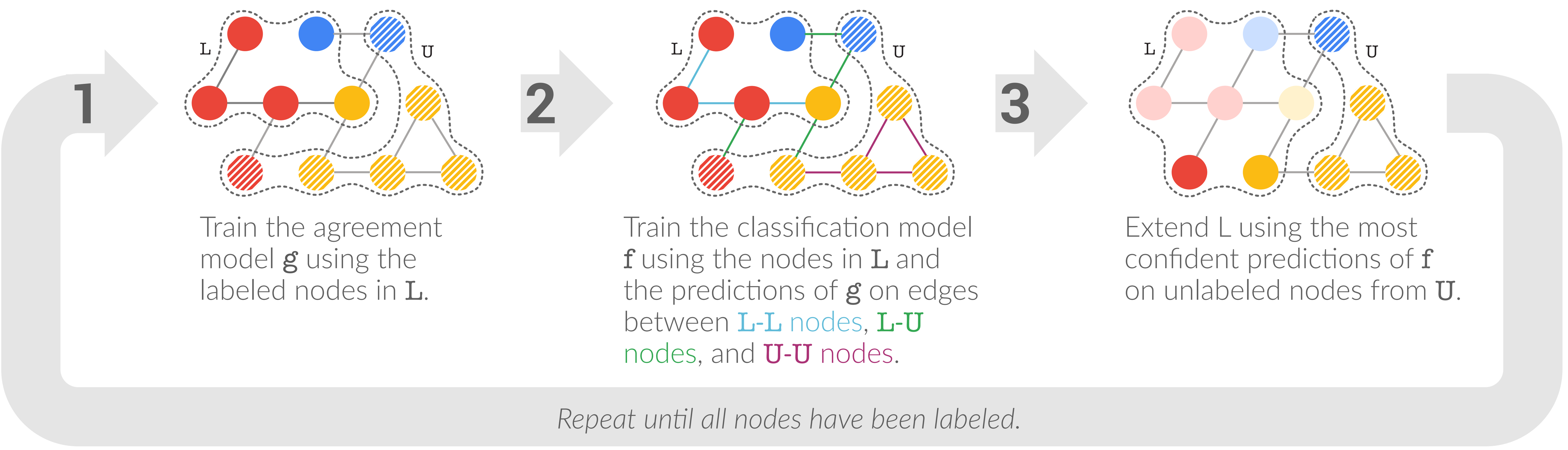


The agreement model is what makes our approach robust to noisy edges and differentiates it from other existing label propagation methods!

PROPERTIES

1. General
   Can be applied on top of any classification model to improve its performance.
2. State-of-the-Art
   Outperforms previous methods on several established datasets.
3. Efficient
   Does not incur any additional performance cost at inference.
4. Robust
   Up to 18% accuracy boost even when only 5% of the edges are "correct".
5. Extends Beyond Graphs
   Even works without a graph, being applicable to broader SSL settings.

TRAINING ALGORITHM

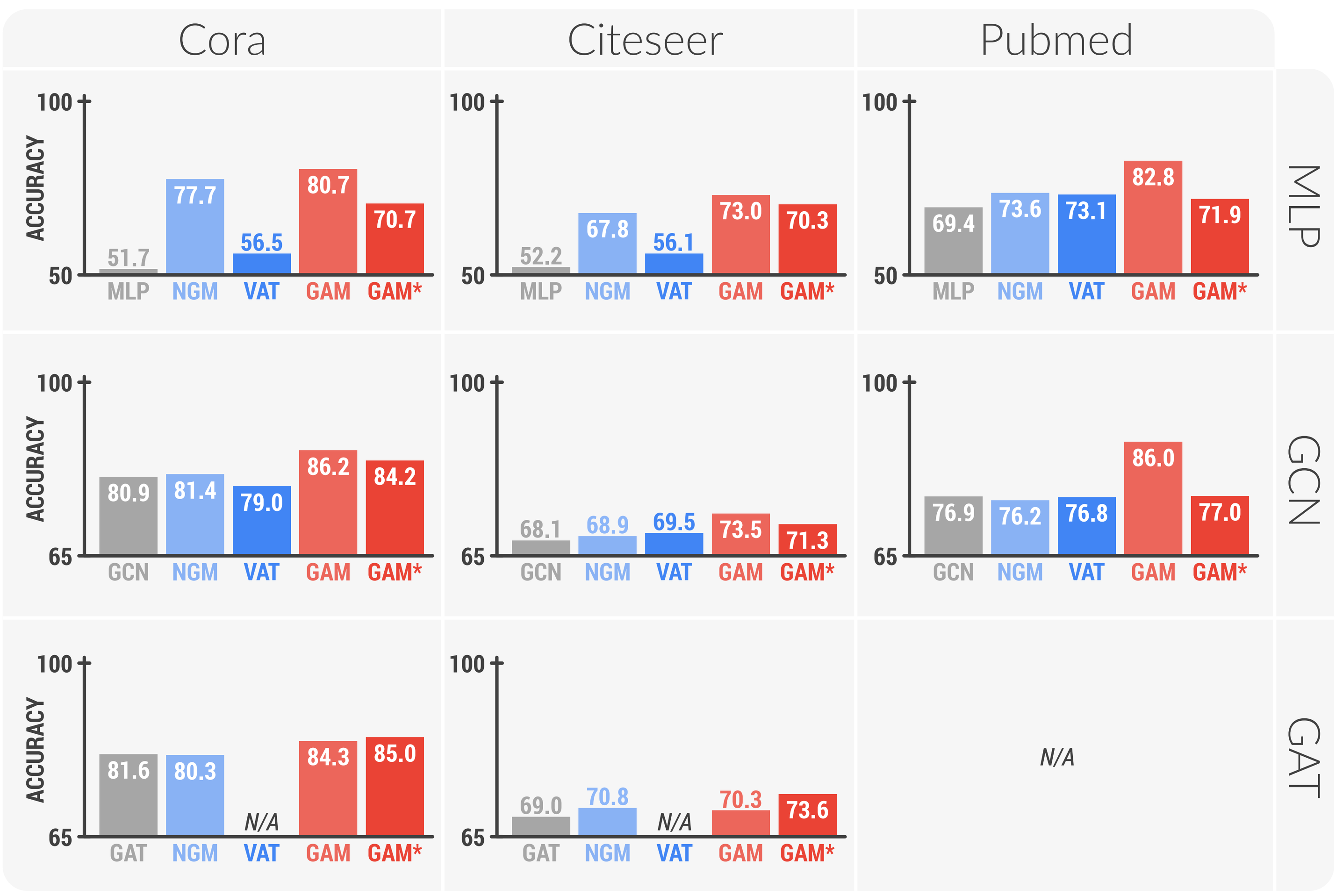


1. Train the agreement model $\mathbf{g}$ using the labeled nodes in $\mathbf{L}$.
2. Train the classification model $\mathbf{f}$ using the nodes in $\mathbf{L}$ and the predictions of $\mathbf{g}$ on edges between L-L nodes, L-U nodes, and U-U nodes.
3. Extend L using the most confident predictions of $\mathbf{f}$ on unlabeled nodes from $\mathbf{U}$.

Repeat until all nodes have been labeled.

Experiments

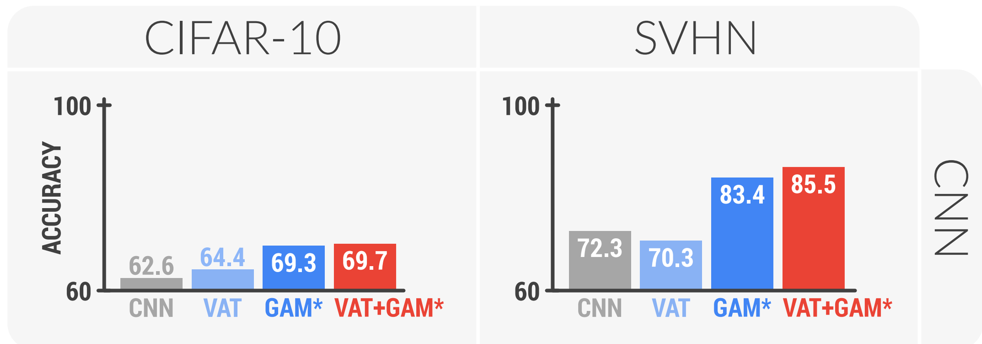
We first evaluate on benchmark graph node classification datasets, comparing to multiple baseline methods:



GAM = our method when a graph is provided
GAM* = our method when a graph is *not* provided

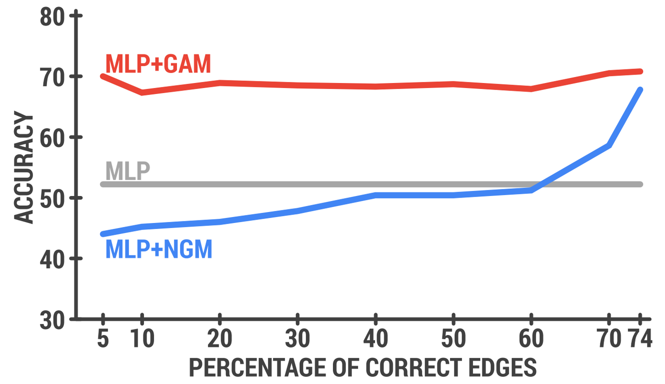
WHAT IF THERE IS NO GRAPH?

GAM can also be applied in settings where no graph is provided, by assuming a fully-connected graph. This is because it can handle noisy edges. We evaluate on two popular semi-supervised learning datasets.



WHAT IF THE GRAPH IS NOISY?

To test how well GAM can handle noisy edges, we performed a robustness analysis by artificially introducing wrong edges to the Citeseer dataset.



Contact: ostretcu@cs.cmu.edu
Code: https://github.com/tensorflow/neural-structured-learning