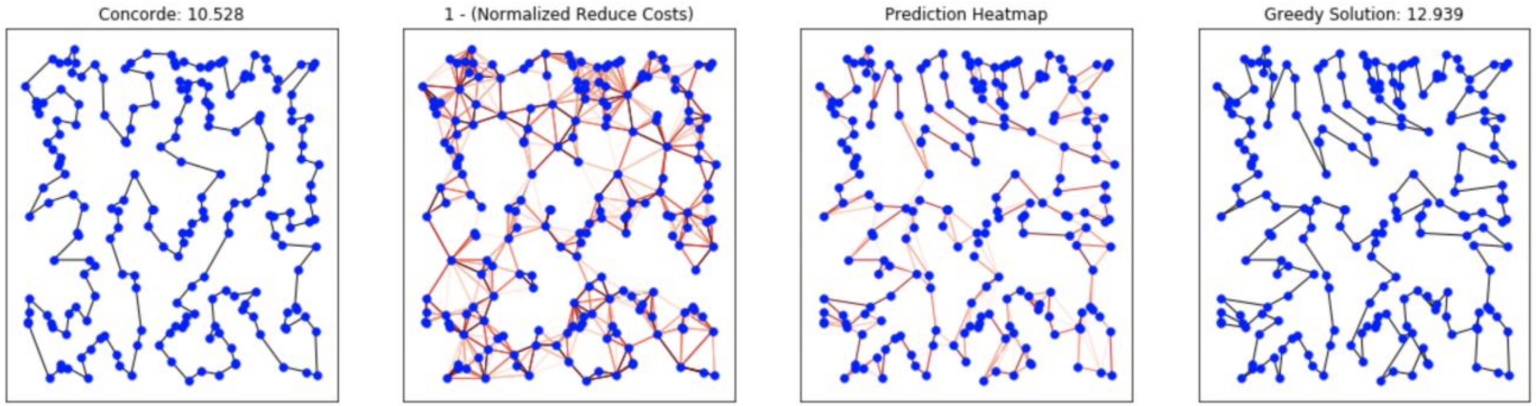


Learning TSP Requires Rethinking Generalization

Chaitanya K. Joshi

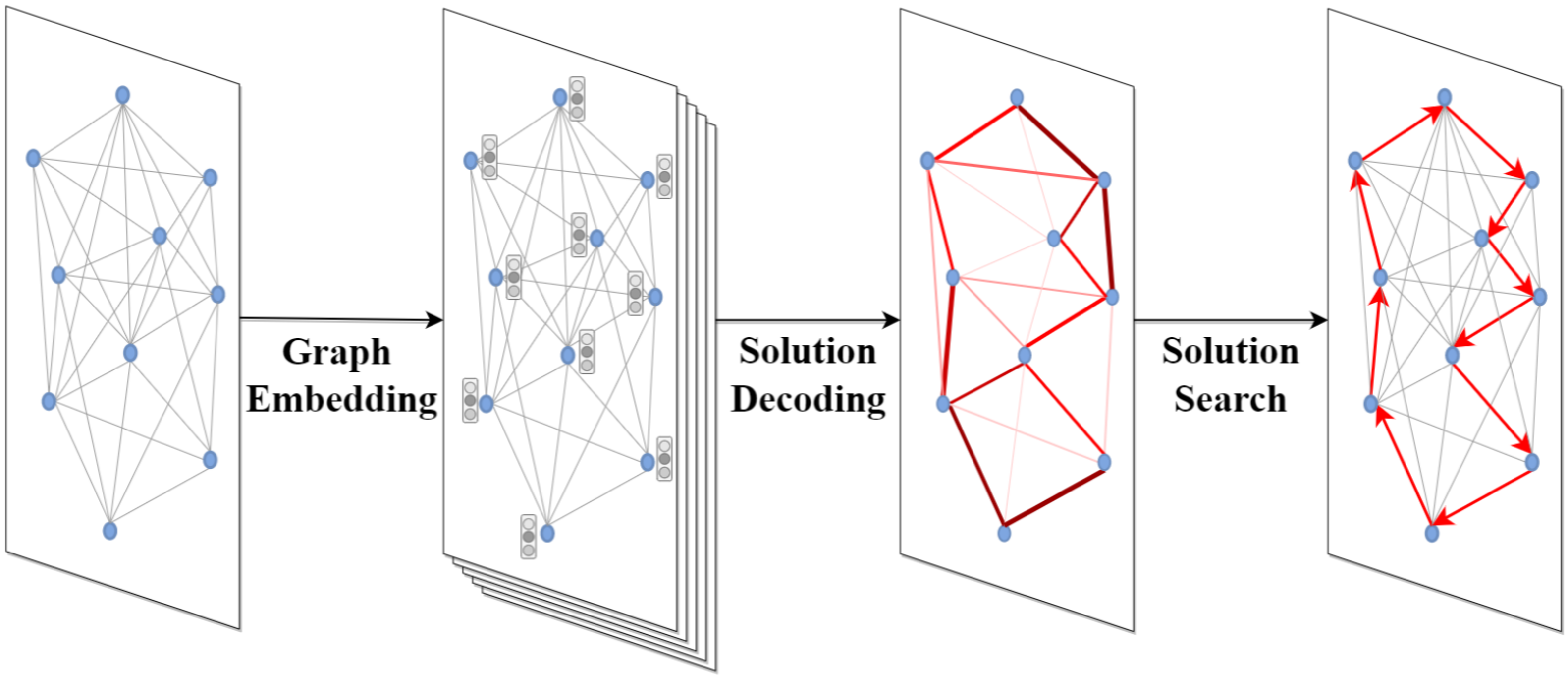
Institute for Infocomm Research, A*STAR, Singapore

Joint work with **Q. Cappart**, **L. M. Rousseau** (Polytechnique Montreal), **T. Laurent** (LMU), **X. Bresson** (NTU)

Travelling Salesperson Problem (TSP)

- Most extensively studied NP-hard problem with wide practical applications.

- Engine of discovery for advances in applied mathematics...and deep learning?



Summary

- Experimental review of SotA **deep learning-based combinatorial optimization solvers**, with TSP as a benchmark.

- We can learn to solve trivially small instances close to optimality, but **extrapolating** to larger and realistic problem instances is a challenging and **open problem**.

Inspired by

UNDERSTANDING DEEP LEARNING REQUIRES RE-THINKING GENERALIZATION

Chiyuan Zhang*
Massachusetts Institute of Technology
chiyuan@mit.edu

Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht†
University of California, Berkeley
brecht@berkeley.edu

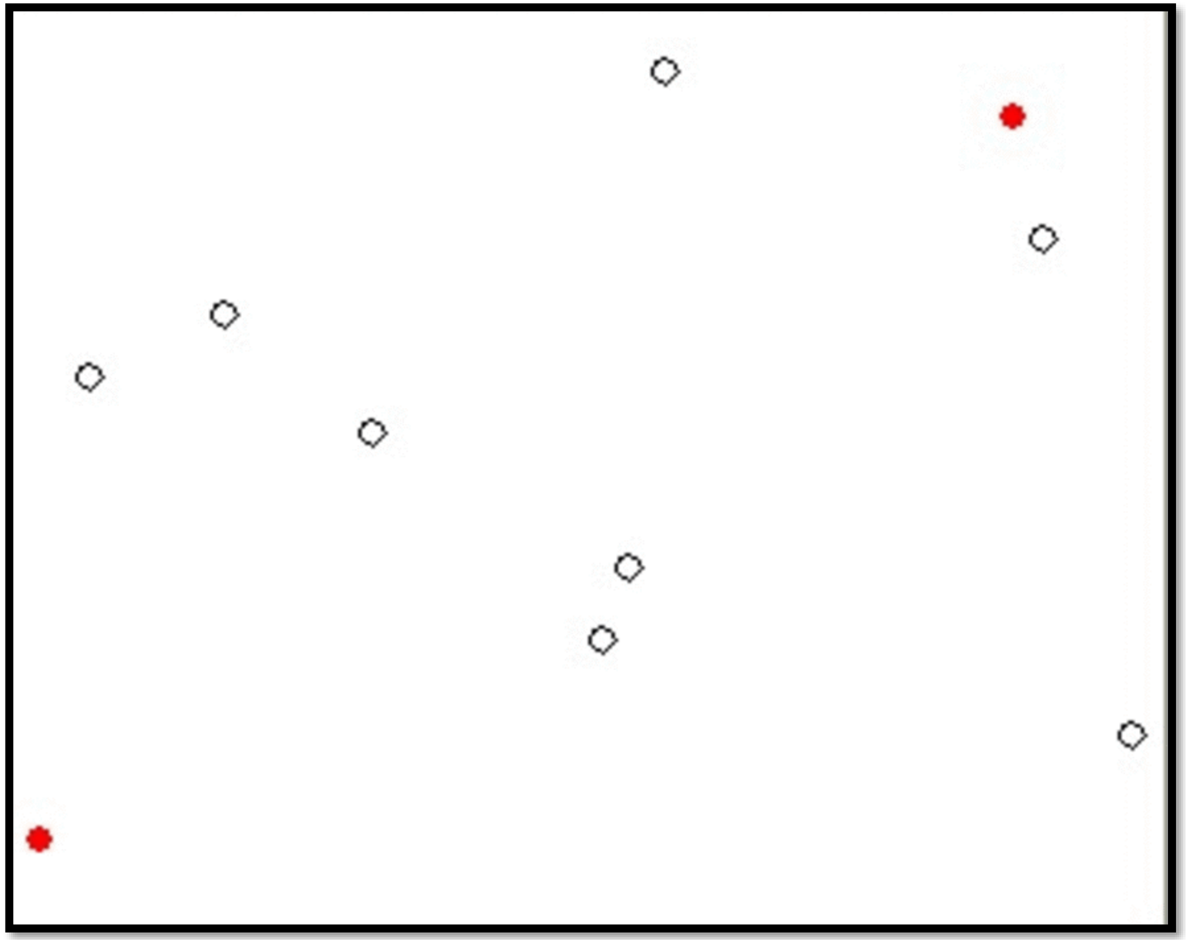
Oriol Vinyals
Google DeepMind
vinyals@google.com

- Deep Neural Networks can **perfectly memorize** very complex and **random training data**.
- Inspired the community to study **generalization** and **extrapolation** performance **beyond training** data.

Motivation

Non-learnt Heuristics for TSP

- Cheap approximate solutions.
- No/few theoretical guarantees.
- Handcrafted.

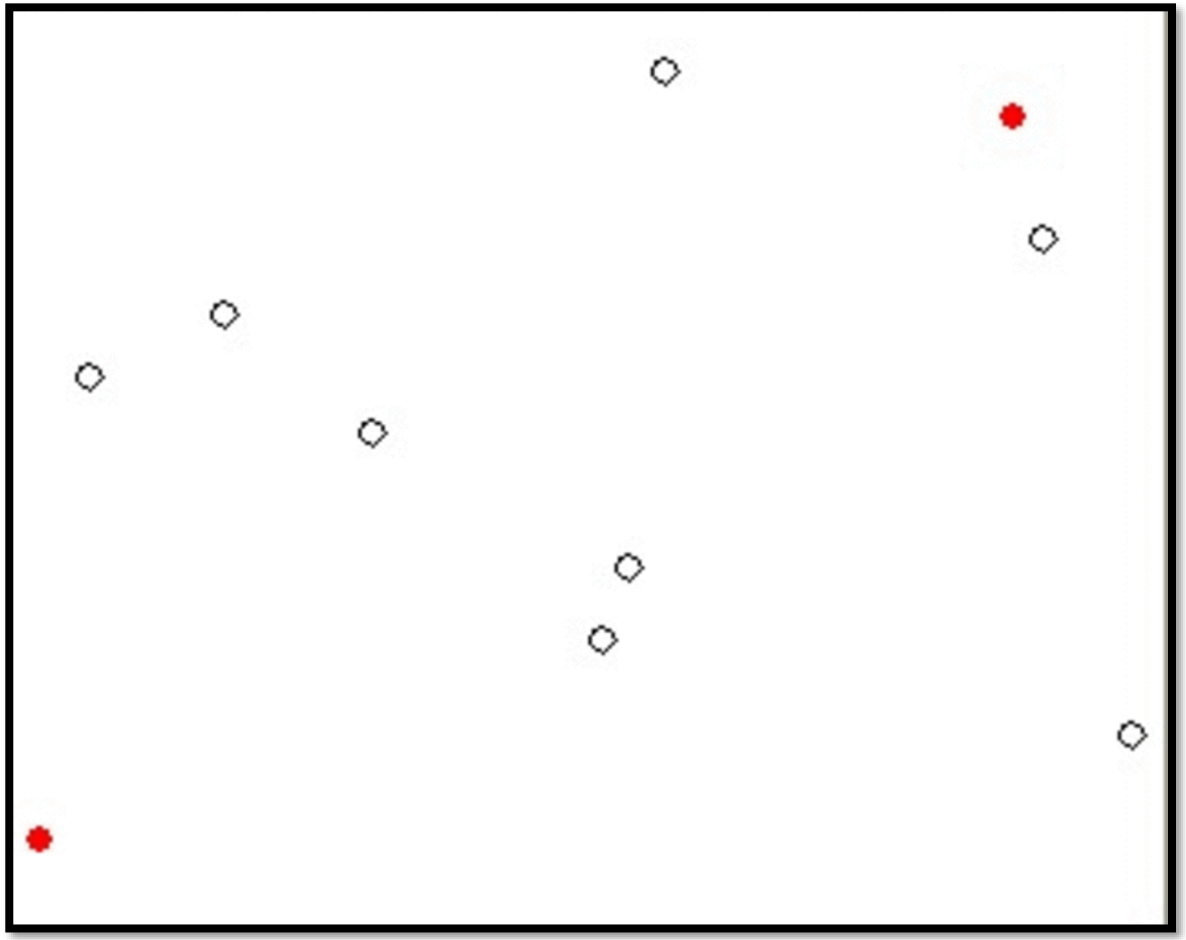


Furthest Insertion Heuristic (GIF)

Motivation

Learnt Heuristics for ~~TSP~~ Novel NP-hard Problems

- Cheap approximate solutions.
- No/few theoretical guarantees.
- ~~Handcrafted.~~
  - **Learnt from problem instances via deep neural networks.**



Furthest Insertion Heuristic (GIF)

Motivation

Learnt Heuristics for ~~TSP~~ Novel NP-hard Problems

Recent works showing this is possible for trivially small TSP instances…

Under review as a conference paper at ICLR 2017

NEURAL COMBINATORIAL OPTIMIZATION WITH REINFORCEMENT LEARNING

Irwan Bello*, **Hieu Pham***, **Quoc V. Le**, **Mohammad Norouzi**, **Samy Bengio**
Google Brain
{ibello,hyhieu,qvl,mnorouzi,bengio}@google.com

Learning Combinatorial Optimization Algorithms over Graphs

Hanjun Dai[†*], **Elias B. Khalil**[†*], **Yuyu Zhang**[†], **Bistra Dilkina**[†], **Le Song**[†§]
[†] College of Computing, Georgia Institute of Technology
[§] Ant Financial
{hanjun.dai, elias.khalil, yuyu.zhang, bdilkina, lsong}@cc.gatech.edu

Published as a conference paper at ICLR 2019

ATTENTION, LEARN TO SOLVE ROUTING PROBLEMS!

Wouter Kool
University of Amsterdam
ORTEC
w.w.m.kool@uva.nl

Herke van Hoof
University of Amsterdam
h.c.vanhoof@uva.nl

Max Welling
University of Amsterdam
CIFAR
m.welling@uva.nl

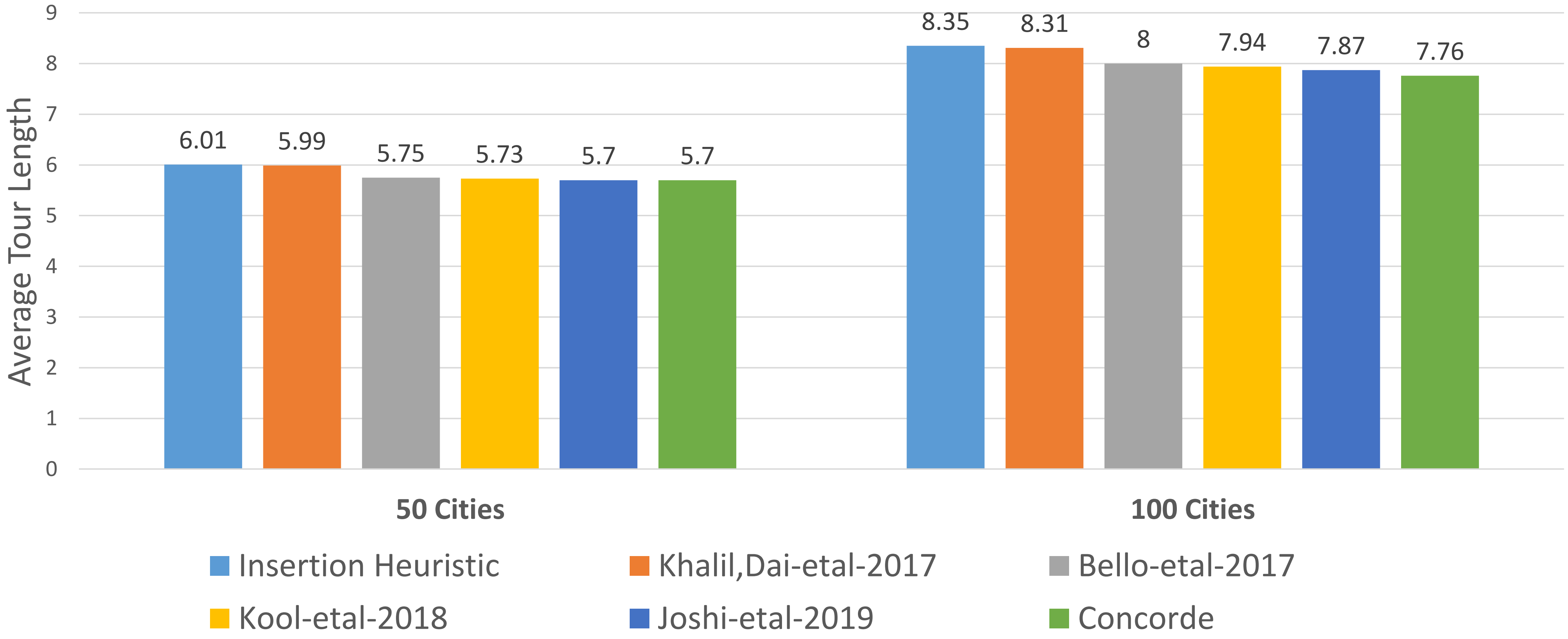
An Efficient Graph Convolutional Network Technique for the Travelling Salesman Problem

Chaitanya K. Joshi[1], **Thomas Laurent**[2], **and Xavier Bresson**[1]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]Department of Mathematics, Loyola Marymount University
{chaitanya.joshi, xbresson}@ntu.edu.sg, tlaurent@lmu.edu

Motivation

Learnt Heuristics for ~~TSP~~ Novel NP-hard Problems

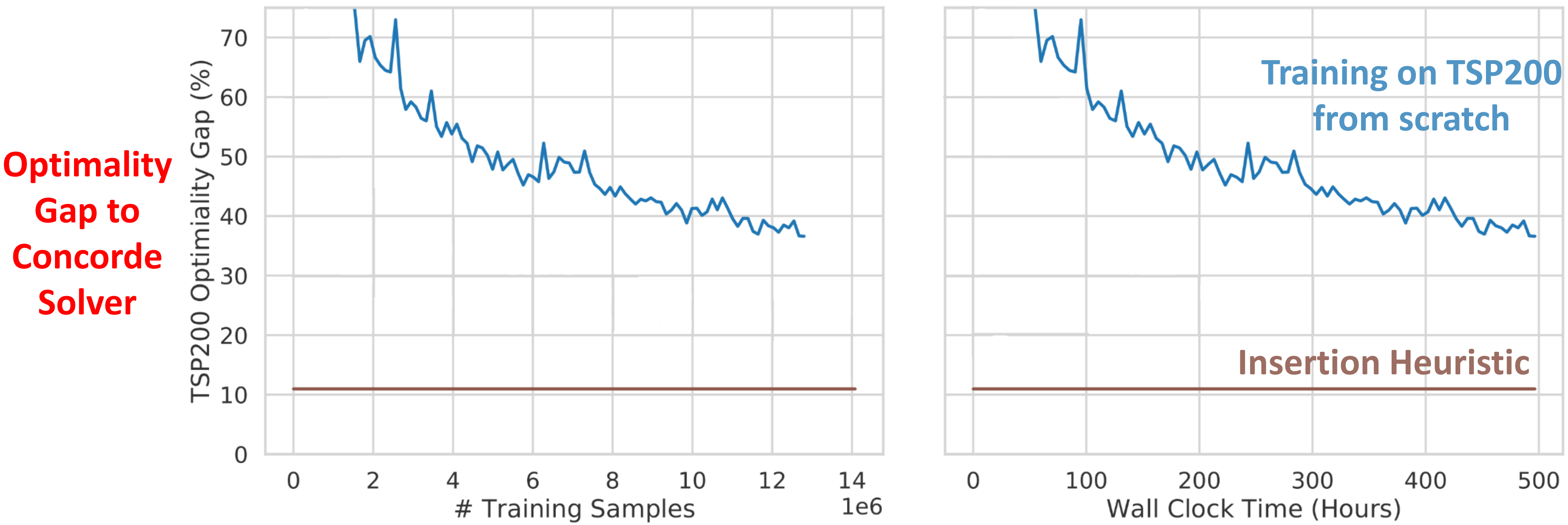
Recent works showing this is possible for trivially small TSP instances…



Research Question

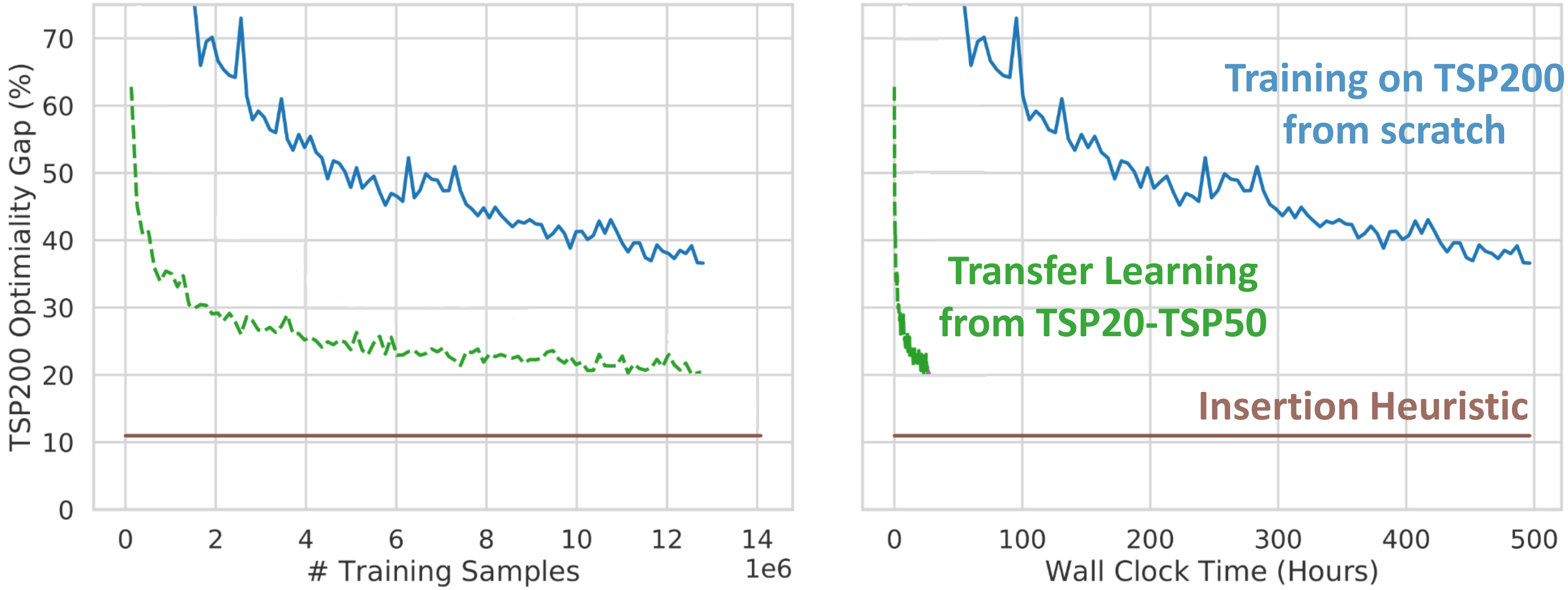
How do we scale to practical sizes beyond few hundreds of nodes?

Option 1: Just Scale SotA Approaches



We were unable to outperform the simple insertion heuristic when directly training on **10+ Million TSP200 samples** for **500 hours** on **university-scale hardware**…

Option 2: Transfer Learning from small instances



Alternative: learn **efficiently** from **trivially small TSPs** + transfer the learnt policy to larger graphs in a **zero-shot fashion** or via fast finetuning.

Which

Architectures,

Learning Paradigms and

Inductive Biases

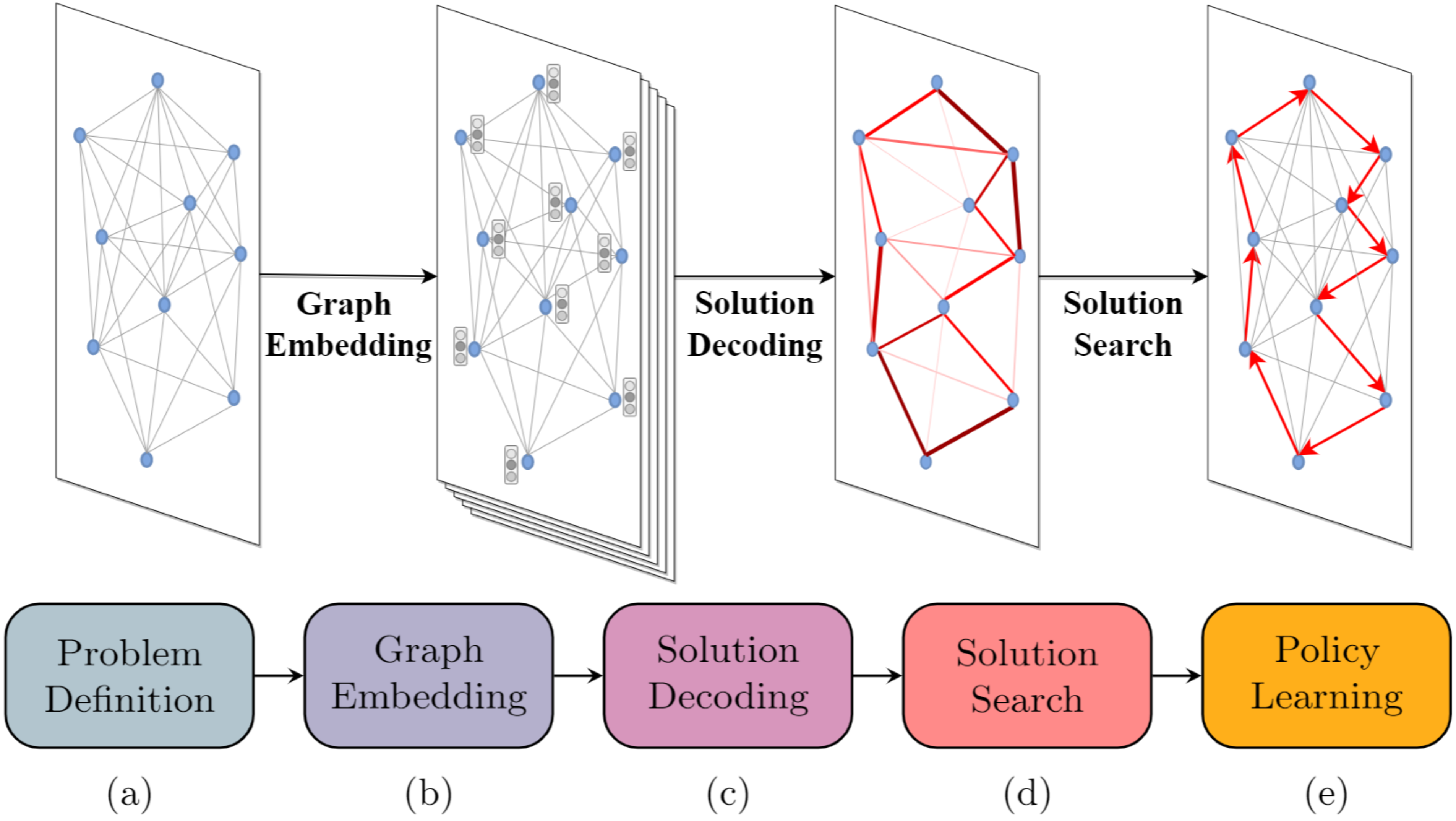
enable strong

Zero-shot Generalization

to large TSP instances?

Step 1: A unified view of recent advances

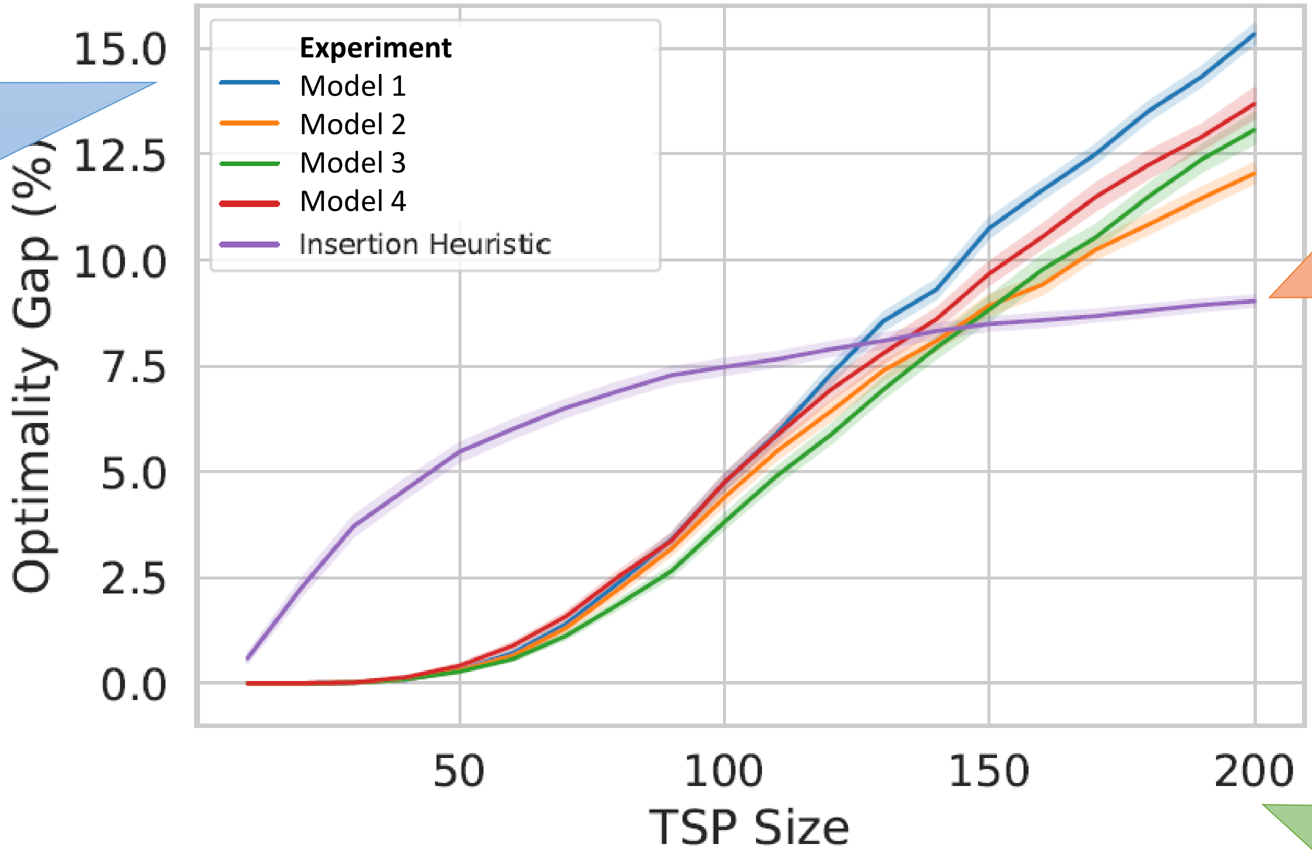
End-to-end Neural Combinatorial Optimization Pipeline



Step 2: A fair and controlled experimental setup

Measuring generalization across TSP sizes

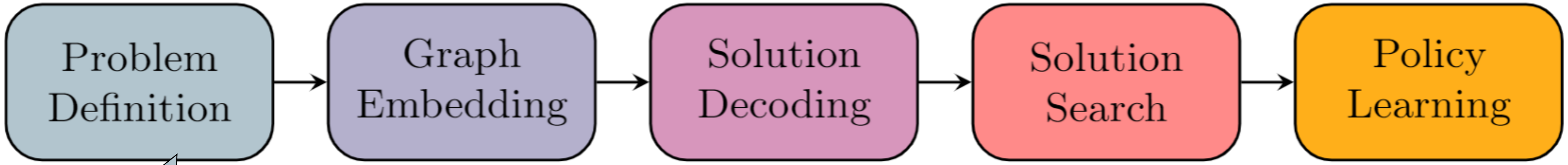
(1) Compare models fairly: fixed #params, epochs, computation.



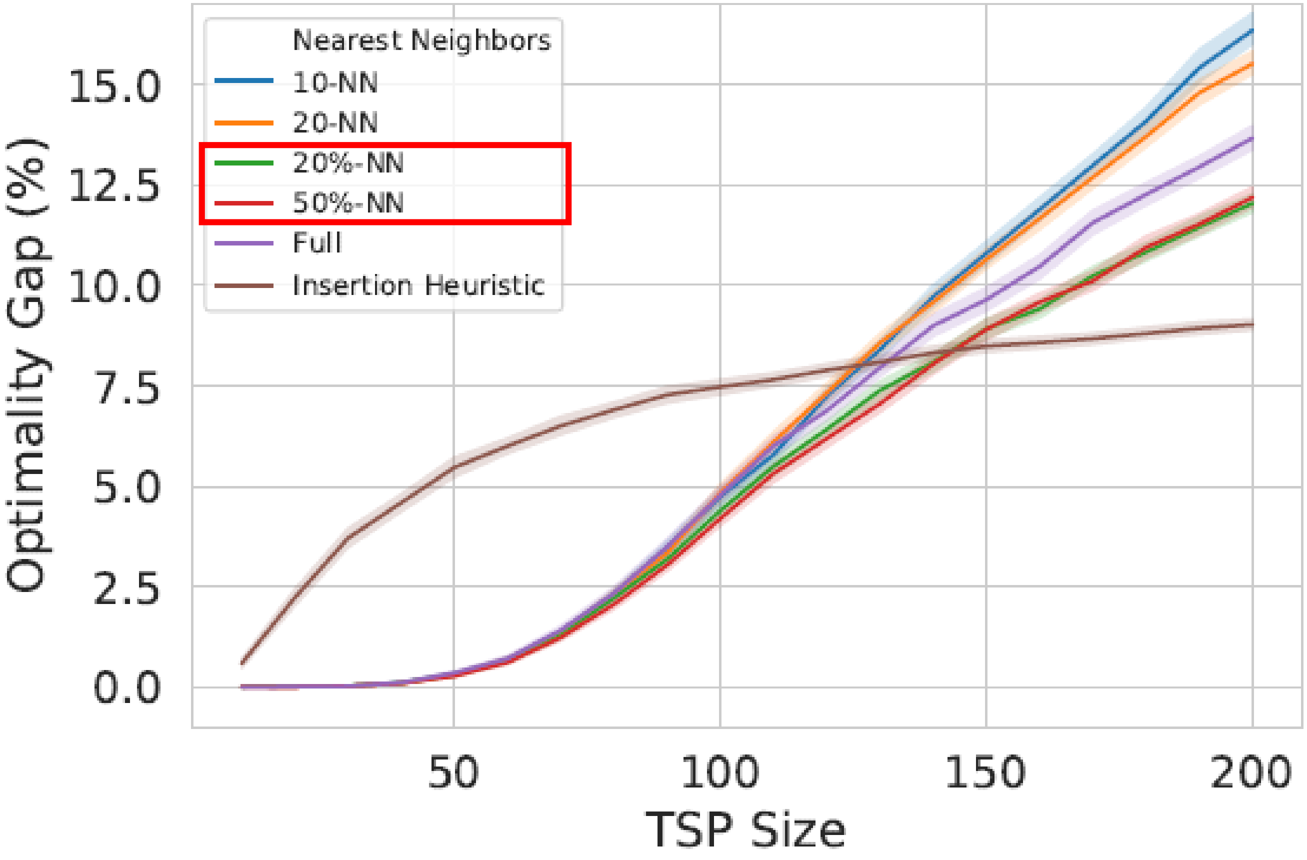
(3) Insertion heuristic baseline: quantify 'good' generalization.

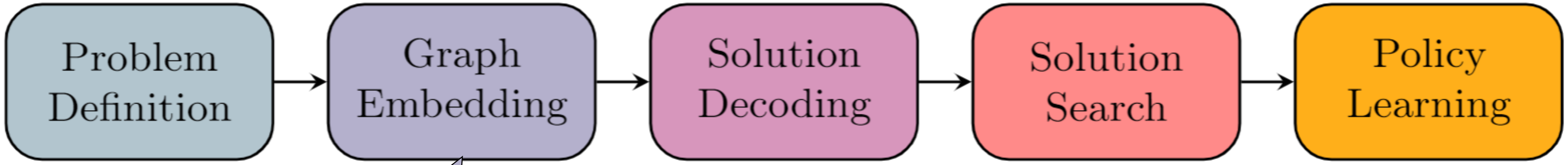
(2) Generalization performance: test TSPs beyond training range.

Our Findings

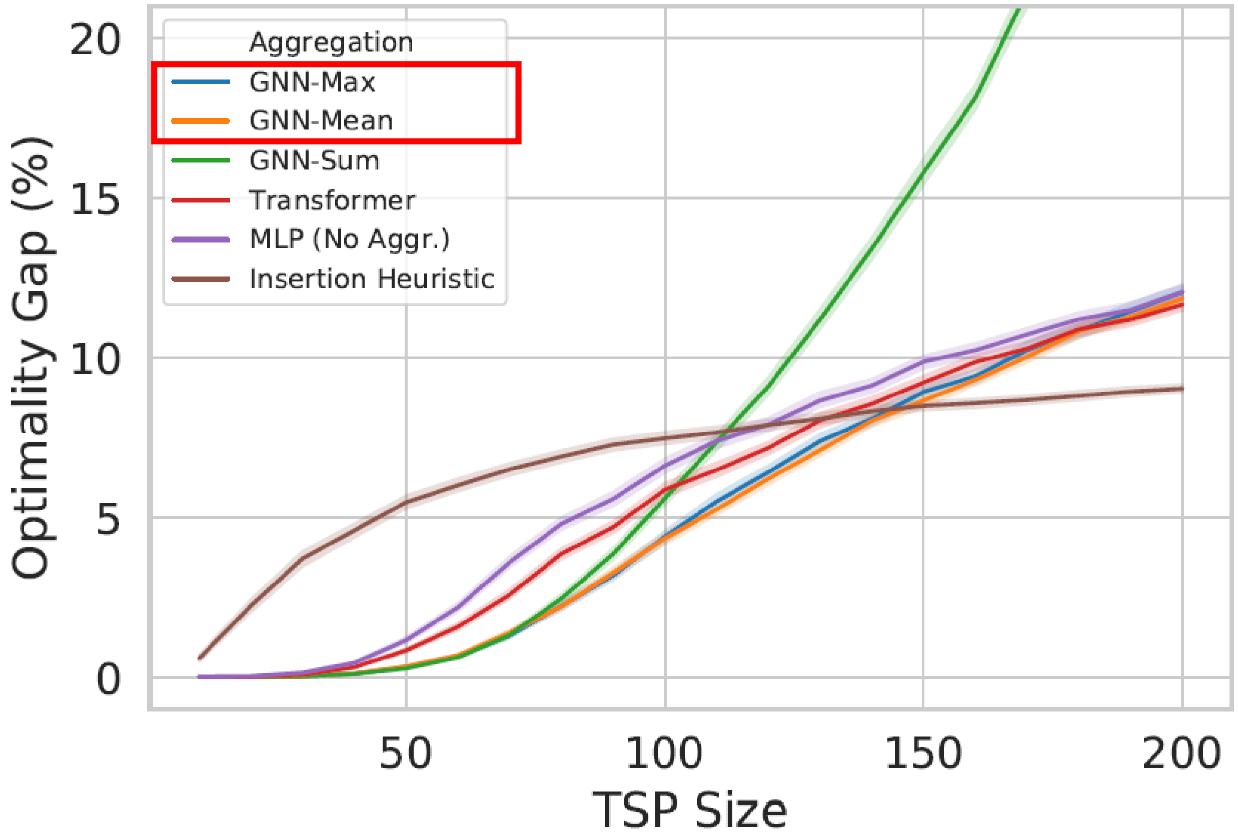


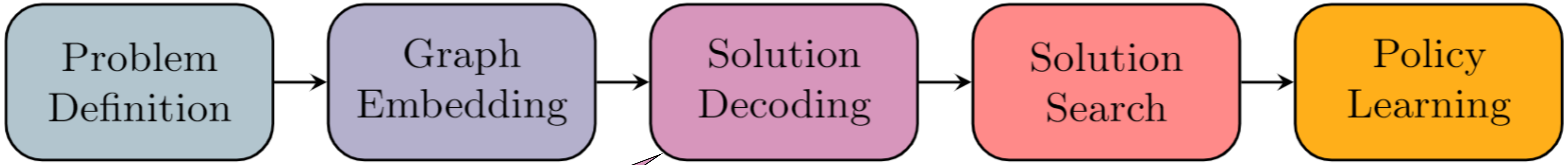
- **Sparse k-NN graphs** > Fully connected graphs.
- Maintain **consistent graph diameter** across TSP sizes.



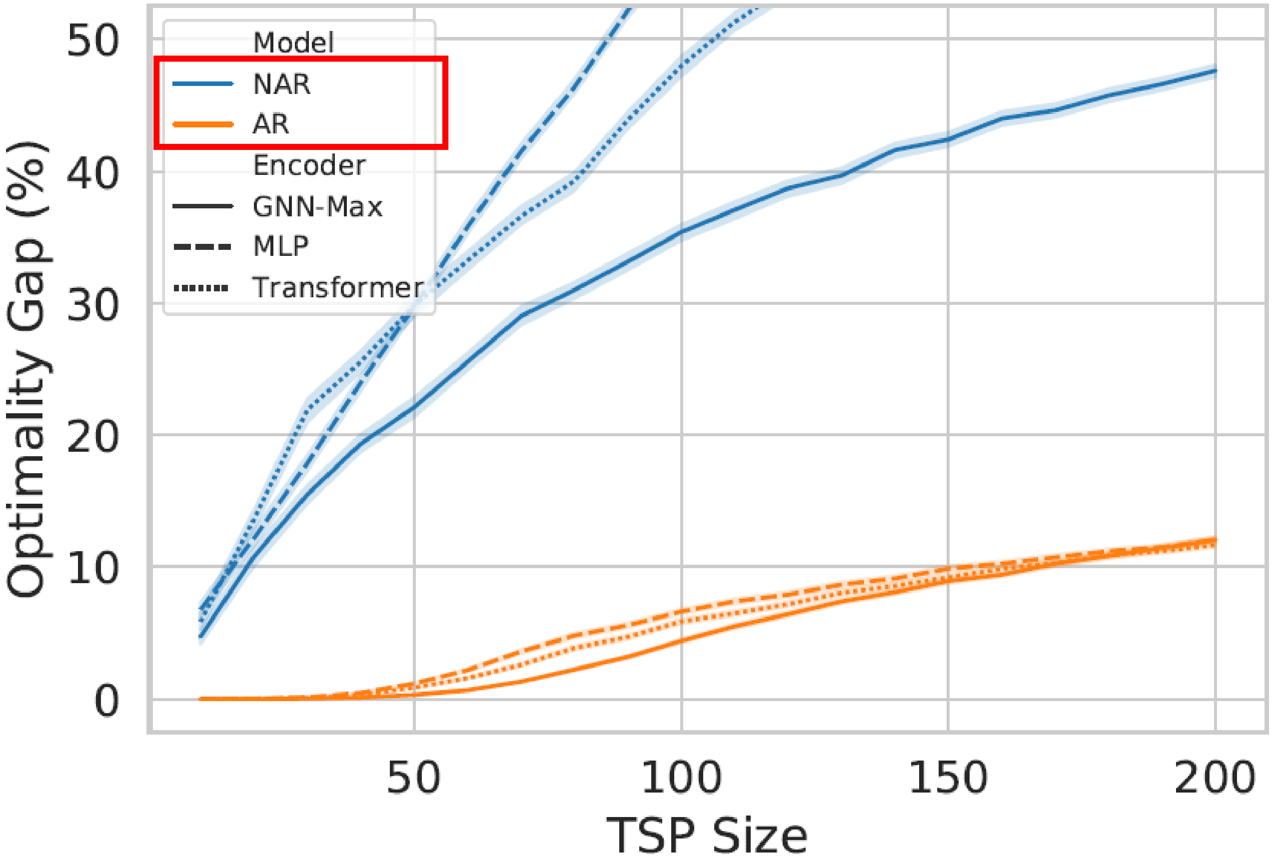


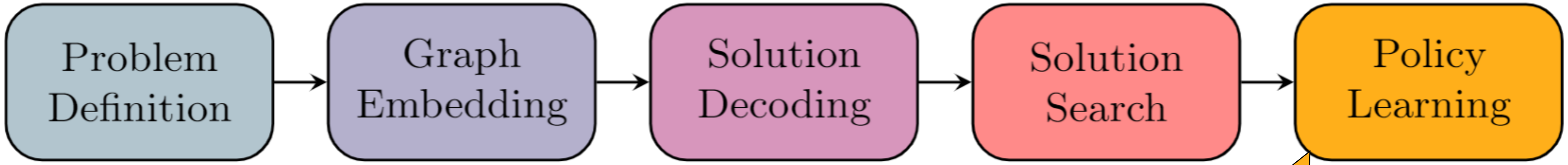
- GNN-Sum, Transformers (**GAT**): most expressive.
- **GNN-Max/Mean**: agnostic to node degree => **better generalization**.

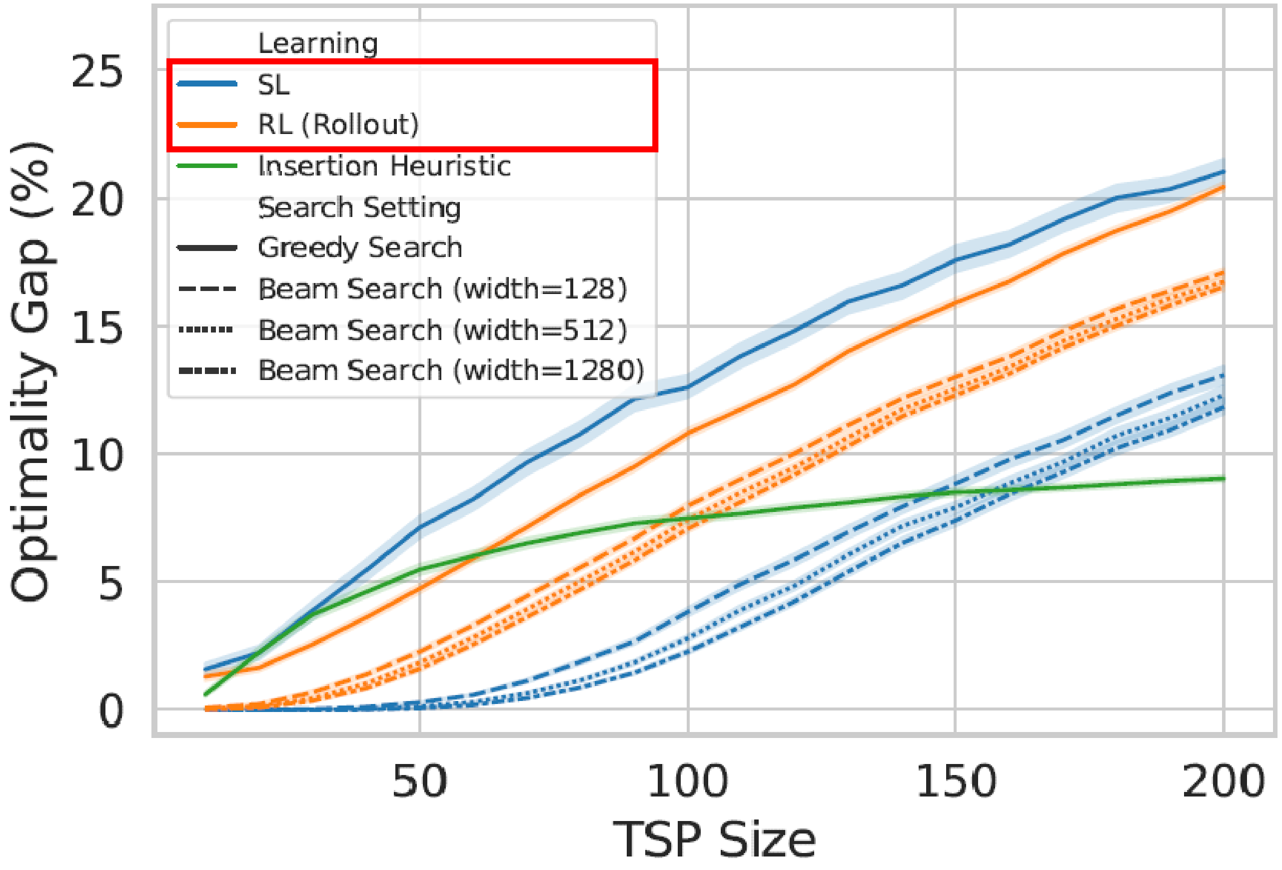




- **Autoregressive decoding (AR):** sequential inductive bias => strong generalization!
- NAR: **fast**, but poor extrapolation.

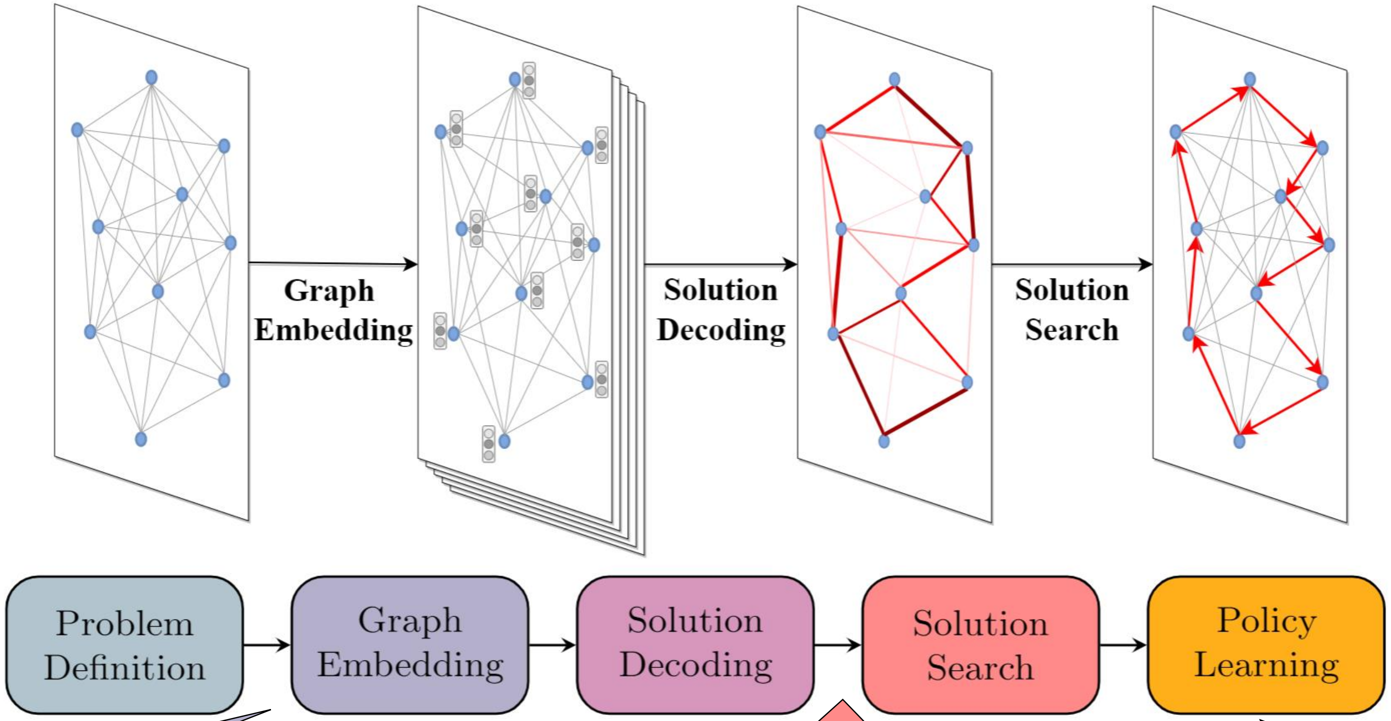






- **Reinforcement Learning (RL):** better under greedy decoding.
- **Supervised Learning (SL):** more amenable to beam search decoding.

What’s next?

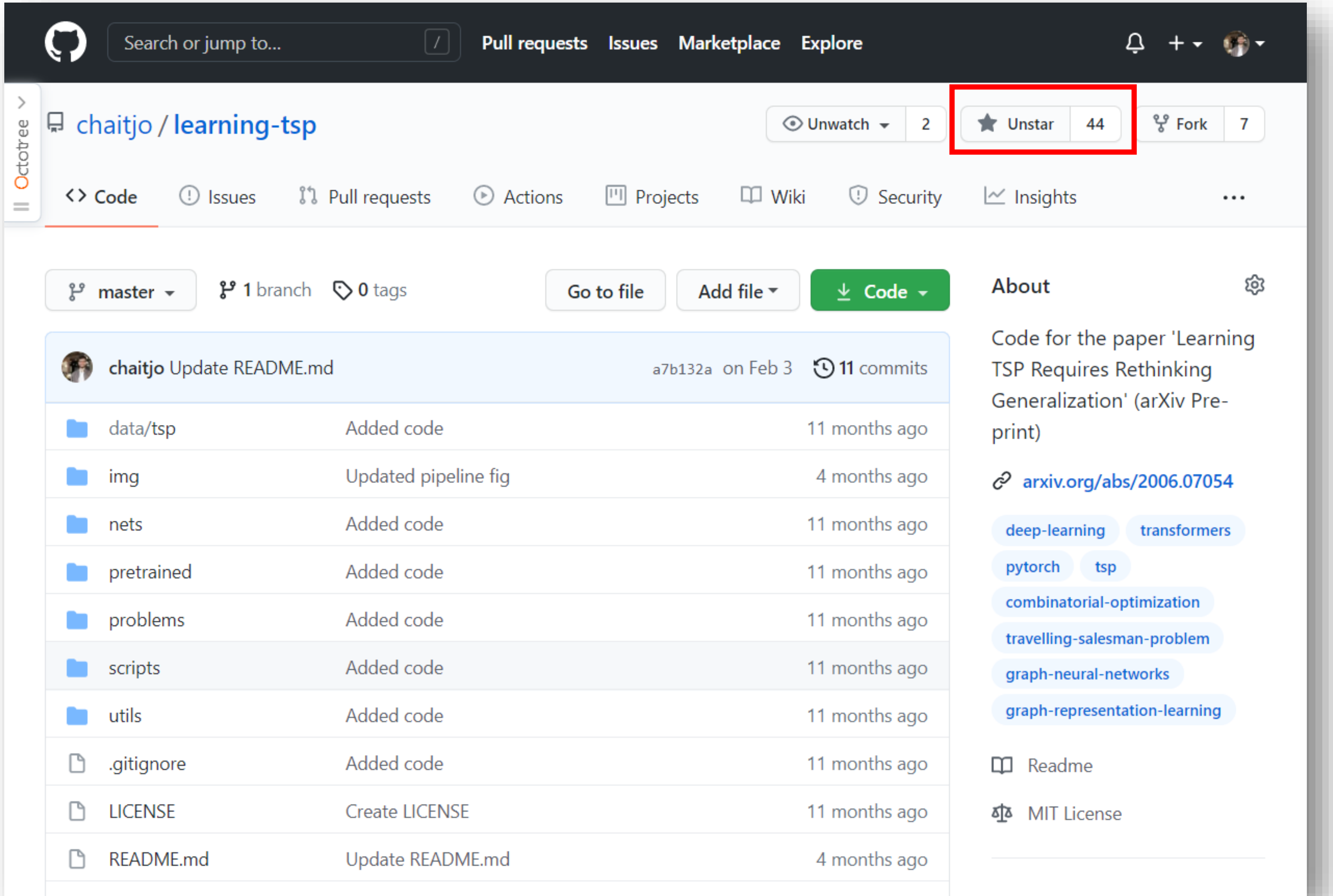


Expressive + **scale invariant GNN** architectures for COPs.

More powerful **classical search techniques** s.a. MC Tree Search, post-hoc Local Search.

Novel **transfer learning** and **meta-learning** techniques for extrapolation.

Pre-print and Code are online



- **ArXiv:** arxiv.org/abs/2006.07054
- **GitHub:** github.com/chaitjo/learning-tsp
- **Blog:** chaitjo.com/neural-combinatorial-optimization/

Thank you!