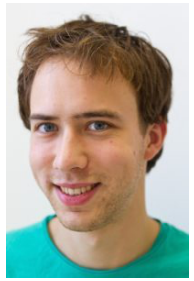
Learning to Solve Routing Problems

Wouter Kool

University of Amsterdam

ORTEC

AMLAB Amsterdam Machine Learning Lab



Herke van Hoof

Joaquim Gromicho

Max Welling

Wouter

Master of Business Analytics &
Master of Econometrics & OR

1+6 years at ORTEC , OR engineer

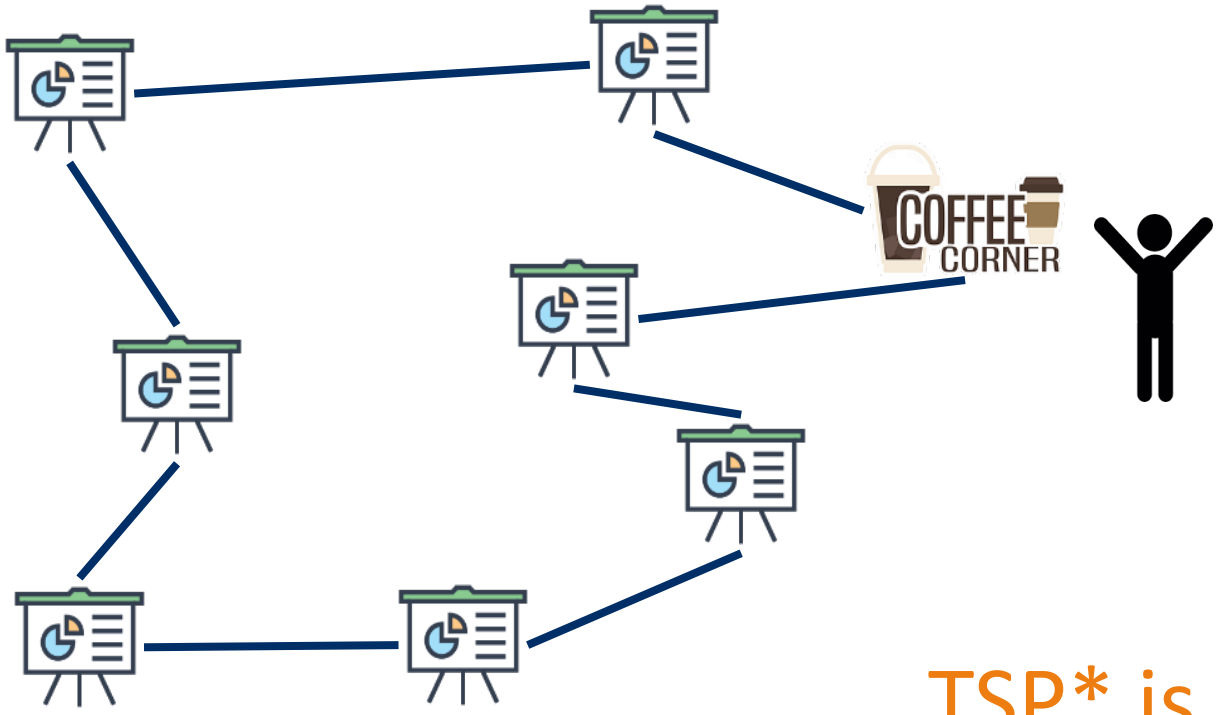
September 2017, PhD @

AMLAB
Amsterdam
Machine Learning Lab

Working on ML & OR

UNIVERSITY OF AMSTERDAM ORTEC

Travelling Scientist Problem (TSP)



TSP* is (NP-)hard!

Kool et al., 2019

UNIVERSITY OF AMSTERDAM ORTEC

What does it mean?

Finding *optimal* solutions for *all* problem instances

MISSION: IMPOSSIBLE

* unless P = NP

Finding *acceptable* solutions for *relevant* problem instances

MISSION: ~~IM~~POSSIBLE

We use HEURISTICS
Can be seen as ‘rules of thumb’

‘next location should be nearby’

Designing of heuristics is like feature engineering

HARD WORK

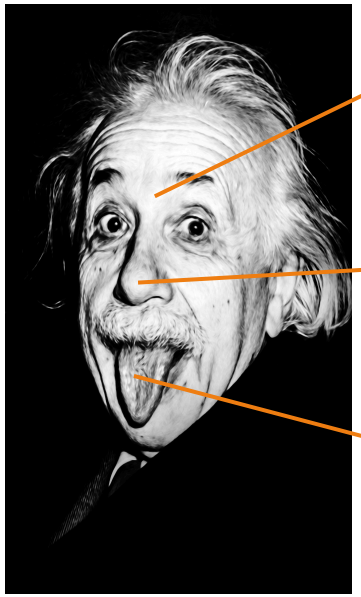
Computer Vision Features (SIFT, etc.)

Feature engineering

- Needs expert knowledge
- Time consuming hand-tuning

So what do we do?

Designing of heuristics is like feature engineering

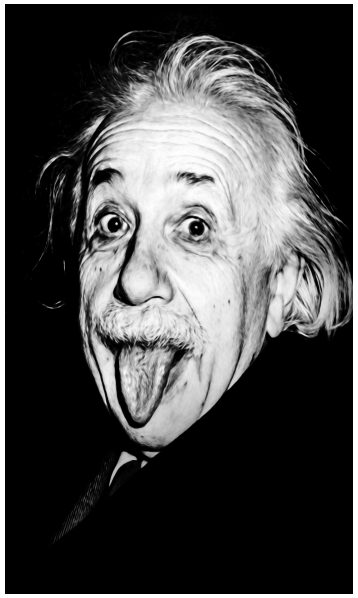


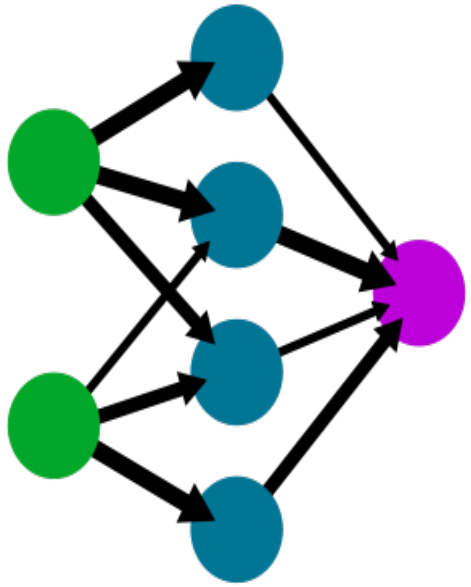
Two eyes?

Nose?

Mouth?

It's a face!





It's a face!

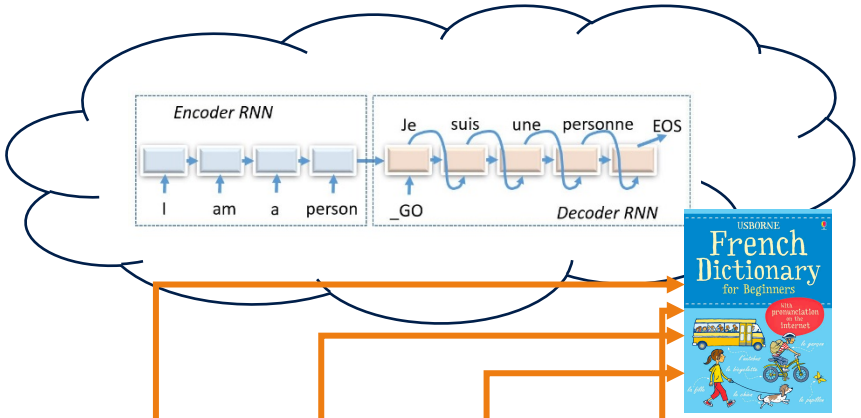
Traditional approach

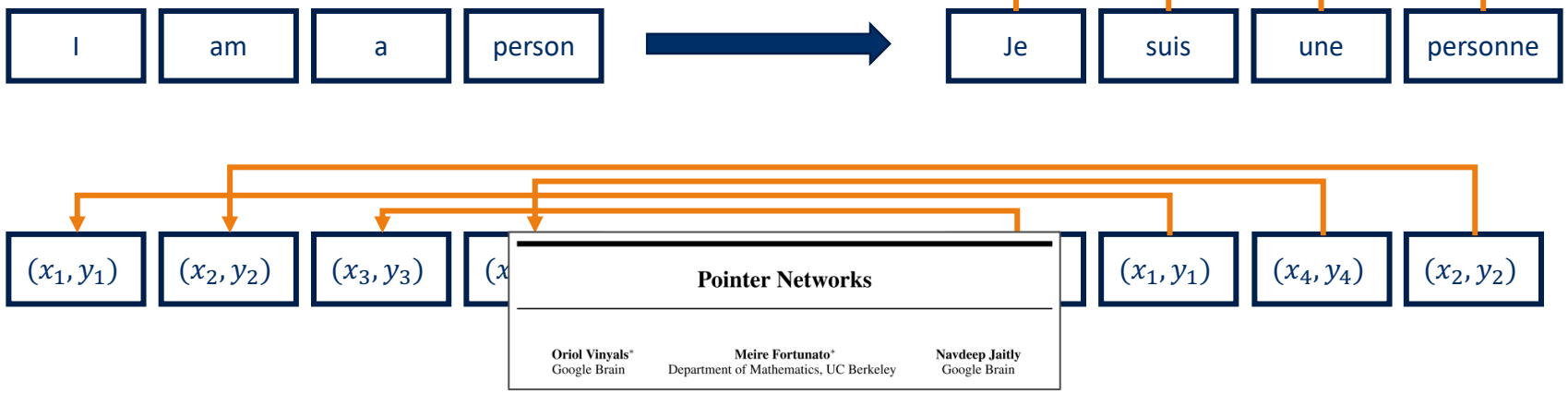
Feature engineering

Deep Learning

No feature engineering

‘Translate’ problem into solution





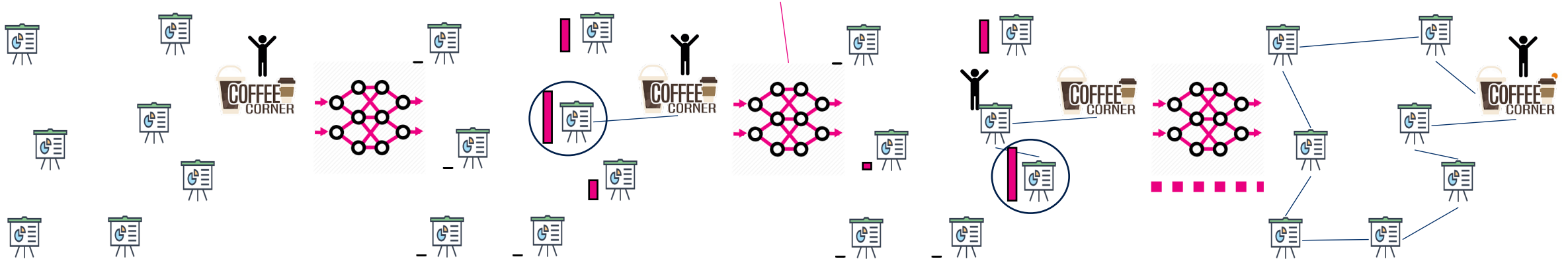
UNIVERSITY OF AMSTERDAM ORTEC

How does that work?

Instance $s =$
$((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$

Model $p_{\boldsymbol{\theta}}(\pi_t | s, \pi_{<t})$=
$p_{\boldsymbol{\theta}}$(next node | partial tour)

Solution $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots)$
with length $L(\boldsymbol{\pi})$



Sample $\pi_1 \sim p_{\boldsymbol{\theta}}(\pi_1 | s)$

Sample $\pi_2 \sim p_{\boldsymbol{\theta}}(\pi_2 | s, \pi_1)$

Sample $\pi_t \sim p_{\boldsymbol{\theta}}(\pi_t | s, \pi_{<t})$

Randomized algorithm
with expected cost:

$$E_{p_{\boldsymbol{\theta}}(\boldsymbol{\pi}|s)}[L(\boldsymbol{\pi})]$$

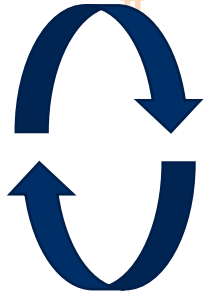
How to optimize $\boldsymbol{\theta}$?

NEURAL COMBINATORIAL OPTIMIZATION WITH REINFORCEMENT LEARNING

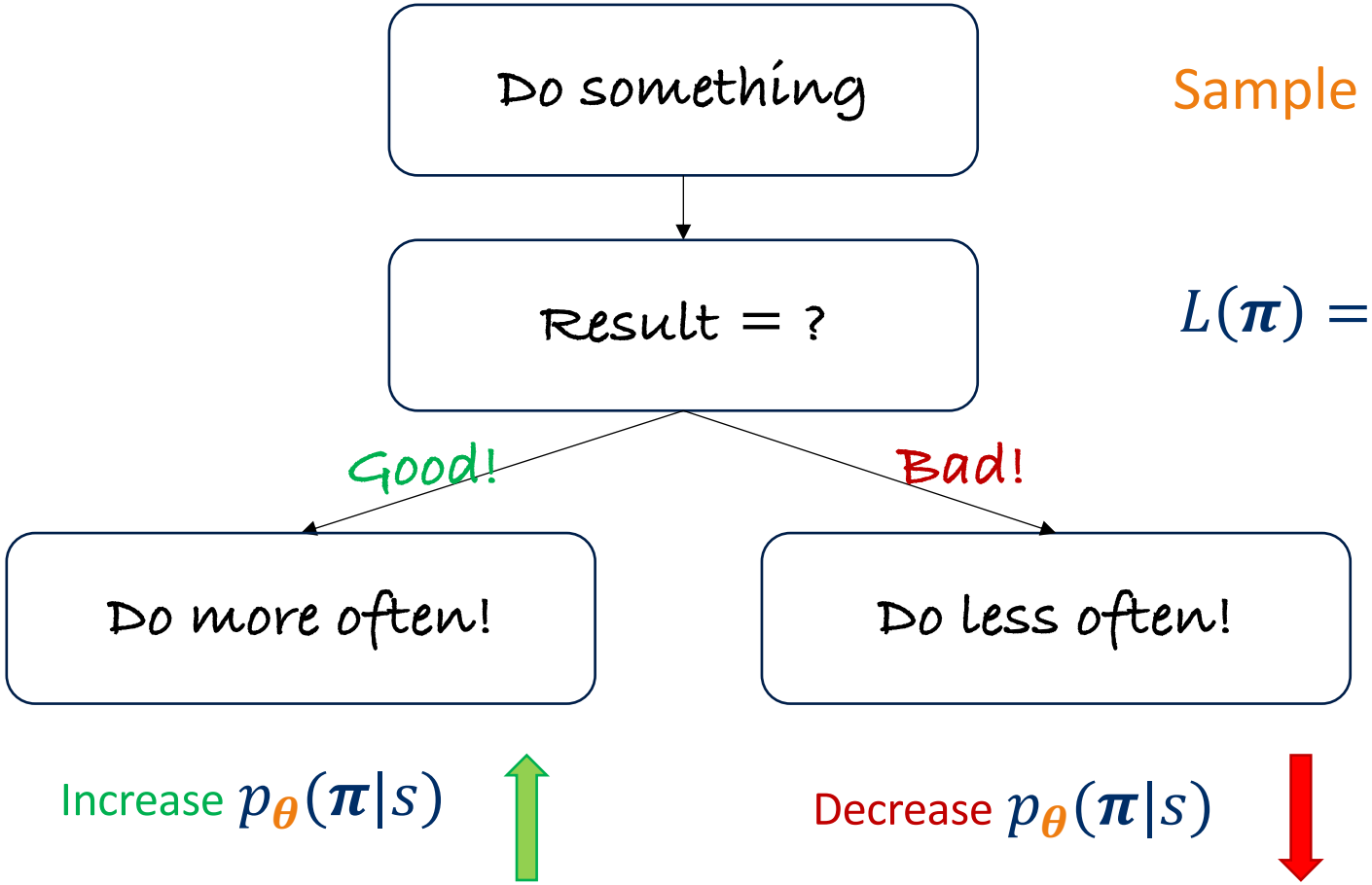
Irwan Bello*, Hieu Pham*, Quoc V. Le, Mohammad Norouzi, Samy Bengio
Google Brain
{ibello,hyhieu,qvl,mnorouzi,bengio}@google.com

UNIVERSITY OF AMSTERDAM ORTEC

REINFORCE (for dummies)



Repeat



Sample $\pi \sim p_\theta(\cdot|s)$

$L(\pi) = 7.43$

We need a *baseline* to compare against: *rollout* earlier model

$\pi^{bl} \sim p_{\theta^{bl}}(\cdot|s)$ **(greedy!)**

$L(\pi^{bl}) = 6.89$

UNIVERSITY OF AMSTERDAM ORTEC

The rollout baseline

Use (rollout) the model but *greedy* instead of sampling!

Sample $\boldsymbol{\pi} \sim p_{\boldsymbol{\theta}}(\cdot\,|s)$

Rollout $\boldsymbol{\pi}^{bl} \sim p_{\boldsymbol{\theta}^{bl}}(\cdot\,|s)$ **(greedy!)**

$L(\boldsymbol{\pi}) < L(\boldsymbol{\pi}^{bl})$ Good!

$L(\boldsymbol{\pi}) > L(\boldsymbol{\pi}^{bl})$ Bad!

Adjust $p_{\boldsymbol{\theta}}(\boldsymbol{\pi}|s)$ proportional to $L(\boldsymbol{\pi}) - L(\boldsymbol{\pi}^{bl})$

Learning Algorithm (roughly)

Init $\boldsymbol{\theta}, \boldsymbol{\theta}^{bl} \leftarrow \boldsymbol{\theta}$

For ever(y epoch):

For iteration:

Sample s

Sample $\boldsymbol{\pi} \sim p_{\boldsymbol{\theta}}(\cdot \,|s)$

Rollout $\boldsymbol{\pi}^{bl} \sim p_{\boldsymbol{\theta}^{bl}}(\cdot \,|s)$ **(greedy!)**

Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \, \nabla \log p_{\boldsymbol{\theta}}(\boldsymbol{\pi}|s) \left(L(\boldsymbol{\pi}) - L\left(\boldsymbol{\pi}^{bl}\right) \right)$

If $\boldsymbol{\theta}$ better* than $\boldsymbol{\theta}^{bl}$

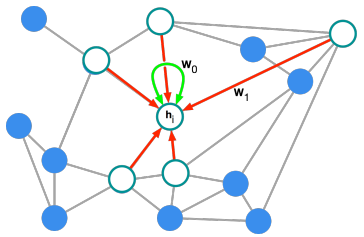
Update $\boldsymbol{\theta}^{bl} \leftarrow \boldsymbol{\theta}$

* Paired *t*-test on solution of 10 000 instances with greedy rollout

What's the model architecture?

$$p_{\boldsymbol{\theta}}(\pi_t | s, \pi_{<t})$$

Graph convolutions



+

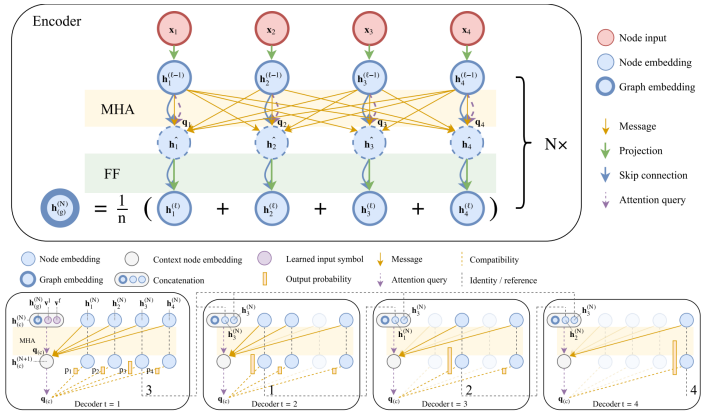
Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Read the paper…

ATTENTION, LEARN TO SOLVE ROUTING PROBLEMS!

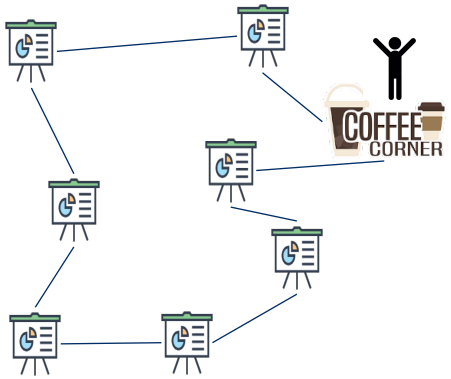
Wouter Kool University of Amsterdam w.w.m.kool@uva.nl
Herke van Hoof University of Amsterdam h.c.vanhoof@uva.nl
Max Welling University of Amsterdam m.welling@uva.nl



UNIVERSITY OF AMSTERDAM ORTEC

Experiments

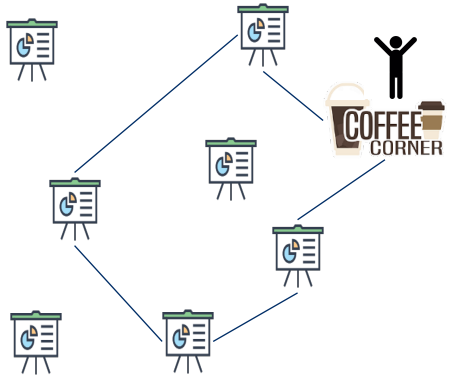
Travelling Salesman Problem (TSP)



Minimize length

Visit all nodes

Orienteering Problem (OP)



Maximize total prize

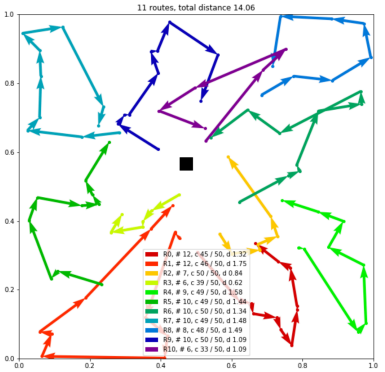
Max length constraint

(Stochastic) Prize Collecting TSP ((S)PCTSP)

Minimize length + penalties of unvisited nodes

Collect minimum total prize

Vehicle Routing Problem (VRP)



Minimize length

Visit all nodes

Total route demand must fit vehicle capacity

Train for each problem, *same hyperparameters*!

UNIVERSITY OF AMSTERDAM ORTEC

Results Attention Model + Rollout Baseline

- Improves over classical heuristics!
- Improves over prior learned heuristics!
  - Attention Model improves
  - Rollout helps significantly
- Gets close to single-purpose SOTA (20 to 100 nodes)!
  - TSP 0.34% to 4.53% (greedy)
  - TSP 0.08% to 2.26% (best of 1280 samples)

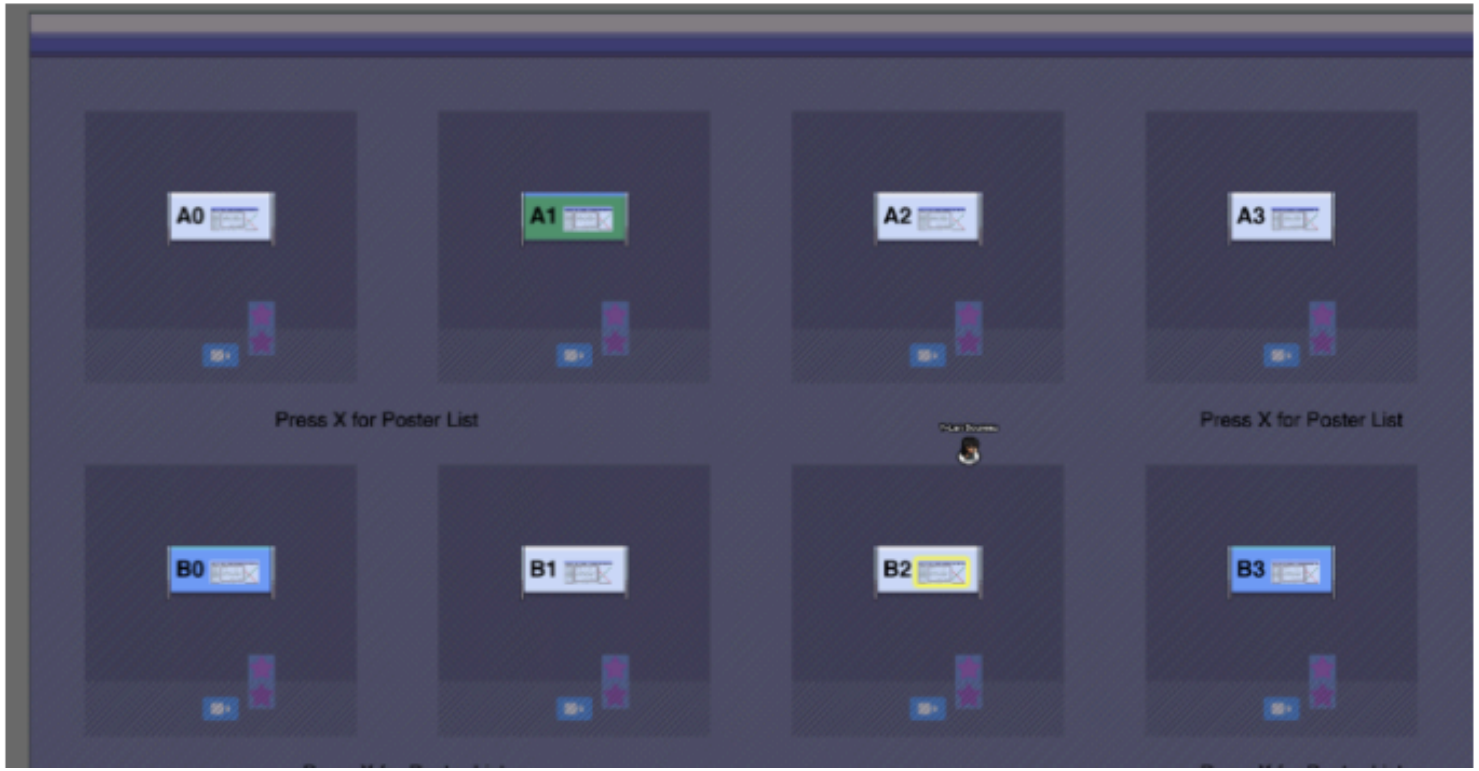
Neural Information Processing Systems Conference

Tweets sent to this account are not actively monitored. To contact us please go to http://neurips.cc/Help/Contact

Follow

41

simply wandering; each poster had a clearly marked presenter spot to easily spot the presenter; people could teleport directly to the poster of their choice from the NeurIPS website, and a coordinate systems allowed people to locate a poster of interest once they were in a room.



https://neuripsconf.medium.com/neurips-2020-online-experiments-gather-town-poster-sessions-and-mementor-ac1573d61c8a

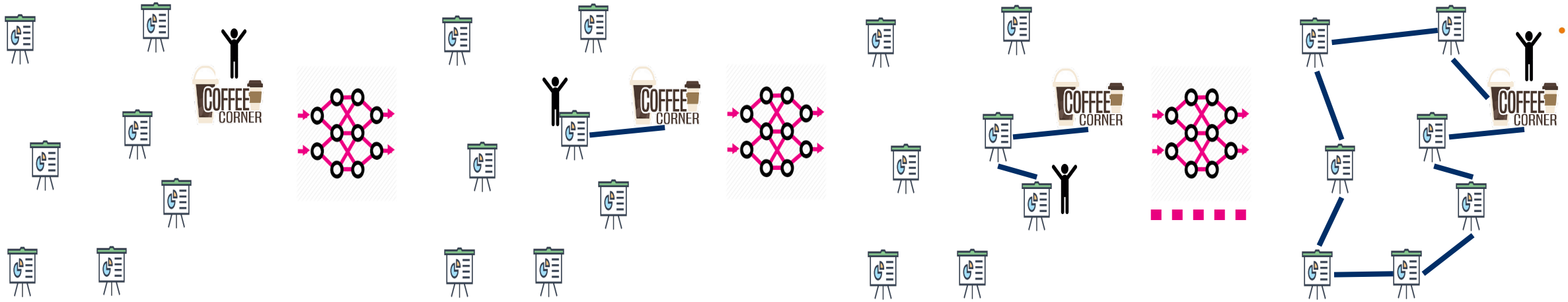
UNIVERSITY OF AMSTERDAM ORTEC

Something else!

Deep Policy Dynamic Programming

ORTEC

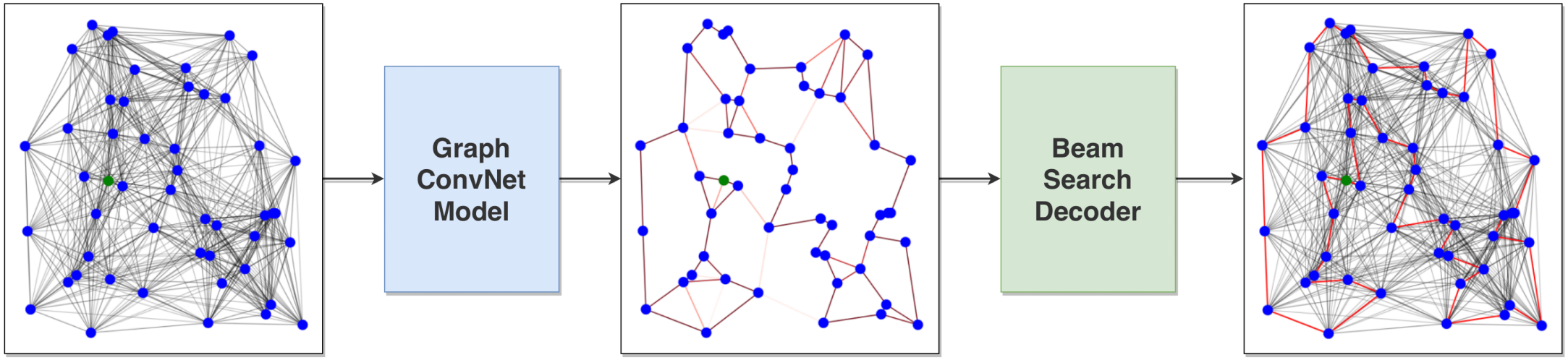
Autoregressive approach



Vinyals et al., 2015
Bello et al., 2016
Kool et al., 2019

UNIVERSITY OF AMSTERDAM ORTEC

Non-autoregressive approach (Joshi et al., 2019)



Picture by Joshi et al., 2019

Note: trained using supervised learning!

Dynamic Programming for TSP (Held & Karp, 1962; Bellman, 1962)

$$C(S, i) = \min_{j \in S \setminus \{i\}} C(S \setminus \{i\}, j) + c_{ji}$$

Minimum cost to go from 0 to i visiting all nodes in S

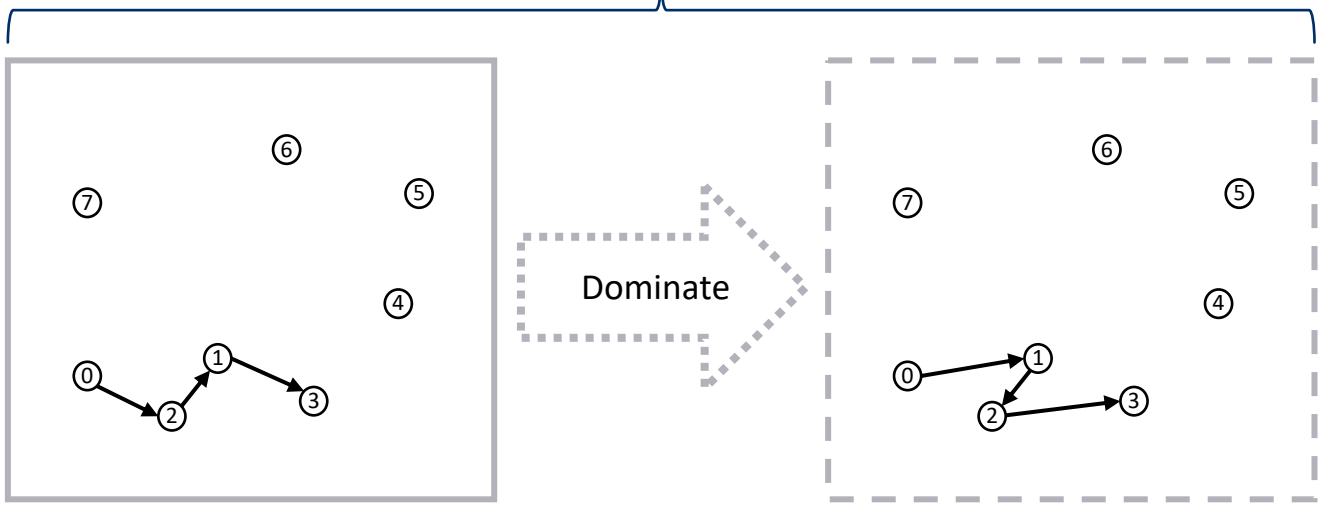
Set of visited nodes

Current node

DP state

Cost/distance from j to i

Find best solution for each DP state



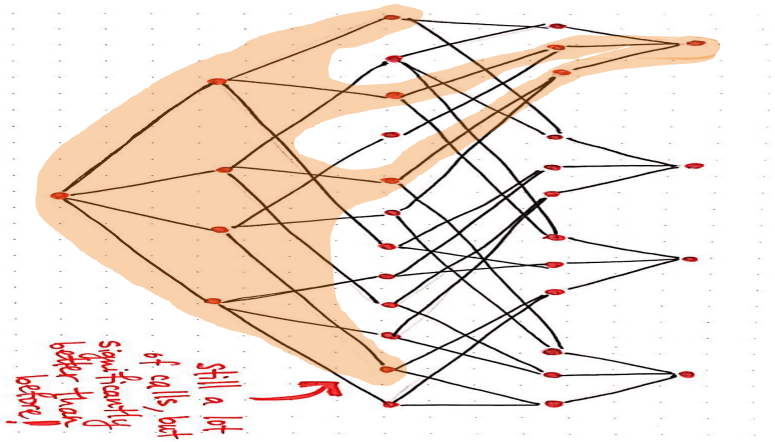
UNIVERSITY OF AMSTERDAM ORTEC

Deep Policy Dynamic Programming (DPDP)

https://arxiv.org/abs/2102.11756

DPDP

$O(Bn)$ or *linear*



For each iteration
- Expand solutions
- Remove dominated solutions
- Select top B according to *policy*
- Repeat

Policy: select top B solutions that maximize the score.

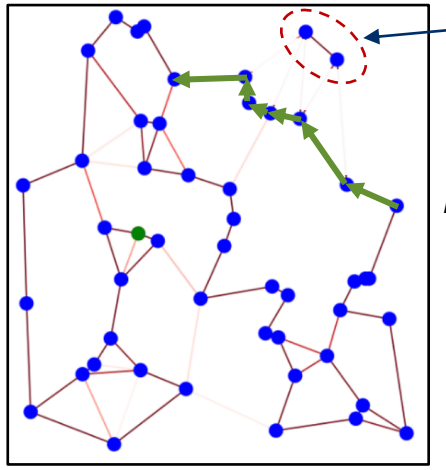
$$\text{SCORE} = \text{HEAT} + \text{POTENTIAL}$$

Heat of edges in solution

Estimate for unvisited nodes based on remaining edges

Forward view

Potential avoids 'skipped nodes'



Heat $h_{ij} \in (0,1)$

Picture by Joshi et al., 2019

UNIVERSITY OF AMSTERDAM ORTEC

Deep Policy Dynamic Programming (DPDP)

https://arxiv.org/abs/2102.11756

- DPDP is a *beam search* over the DP state space, guided by a neural network
- DP is flexible framework for many VRP variants e.g. time windows
- Suitable for GPU implementation
- Natural trade-off compute vs. performance -> asymptotically optimal
- Supervised training based on example solutions
- Test time: only evaluate NN once!

Results

Travelling Salesman Problem

Table 1. Main results for TSP with 100 nodes.

METHOD	COST	GAP	TIME
CONCORDE	7.765	0.000 %	6M
LKH	7.765	0.000 %	42M
GUROBI	7.776	0.15 %	31M
KOOL ET AL. (2019)	7.94	2.26 %	1H
JOSHI ET AL. (2019A)	7.87	1.39 %	40M
DA COSTA ET AL. (2020)	7.83	0.87 %	41M
FU ET AL. (2020)	7.764*	0.04 %	4M + 11M
DPDP 10K	7.765	0.009 %	10M + 1H06M
DPDP 100K	7.765	0.004 %	10M + 2H35M

Vehicle Routing Problem

Table 2. Main results for VRP with 100 nodes.

METHOD	COST	GAP	TIME
LKH	15.647	0.000 %	12H59M
XIN ET AL. (2020)	16.49	4.99 %	39S
KOOL ET AL. (2019)	16.23	3.72 %	2H
CHEN & TIAN (2019)	16.10	2.90 %	1H
PENG ET AL. (2019)	16.27	3.96 %	6H
WU ET AL. (2019)	16.03	2.47 %	5H
HOTTUNG & TIERNEY (2019)	15.99	1.02 %	1H
LU ET AL. (2020)	15.57*	-	4000H
DPDP 10K	15.832	1.183 %	10M + 2H24M
DPDP 100K	15.694	0.305 %	10M + 5H48M
DPDP 1M	15.627	- 0.127 %	10M + 48H27M

The end

Questions?