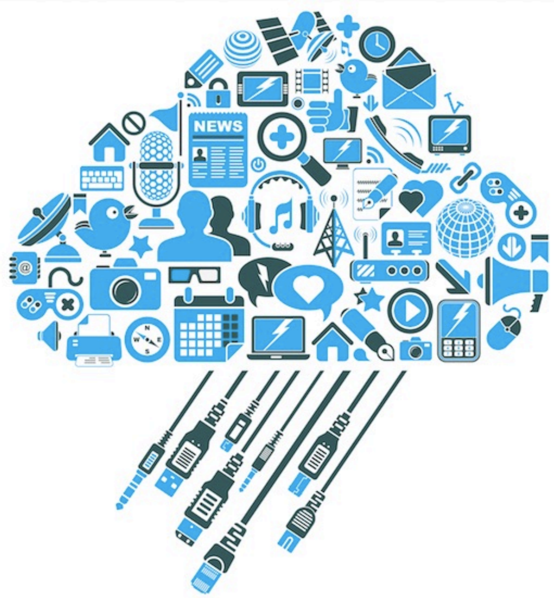


An ML Perspective on WAN TE

Michael Schapira

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Purpose of this Talk

- Many more questions than answers.
  A call to arms!

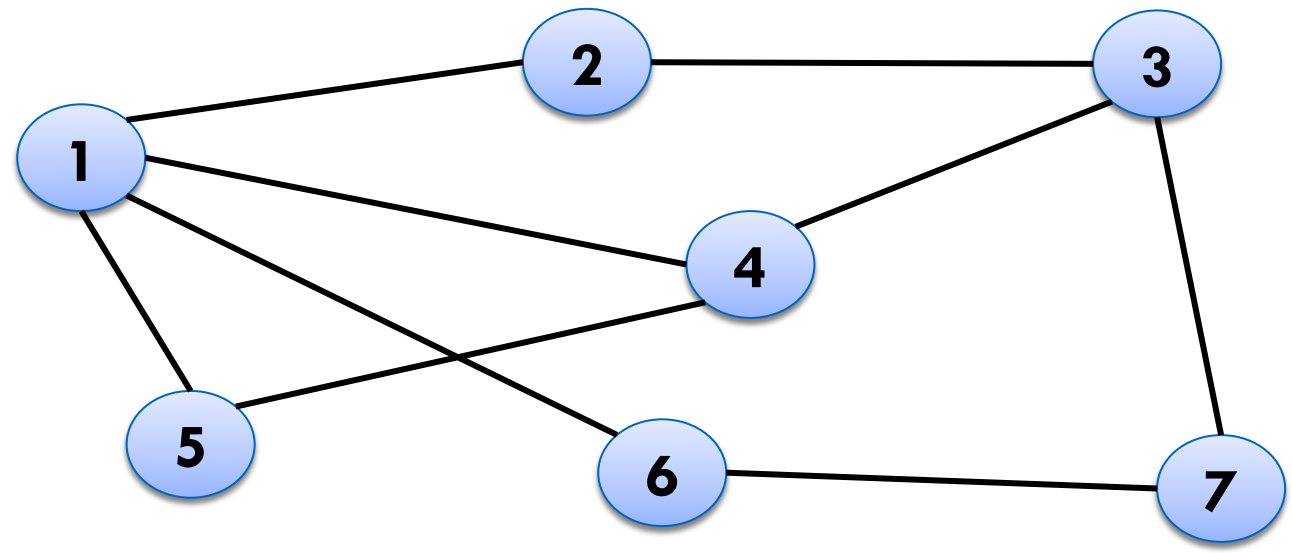
- (Very) high-level overview of challenges and promising approaches

Classical Algorithmic Approach to Routing

	1	2	3	…	N
1	0	100	3	…	13
2	142	0	5	…	0
3	20	0	0	…	32
…	…	…	…	0	…
N	12	0	50	…	0

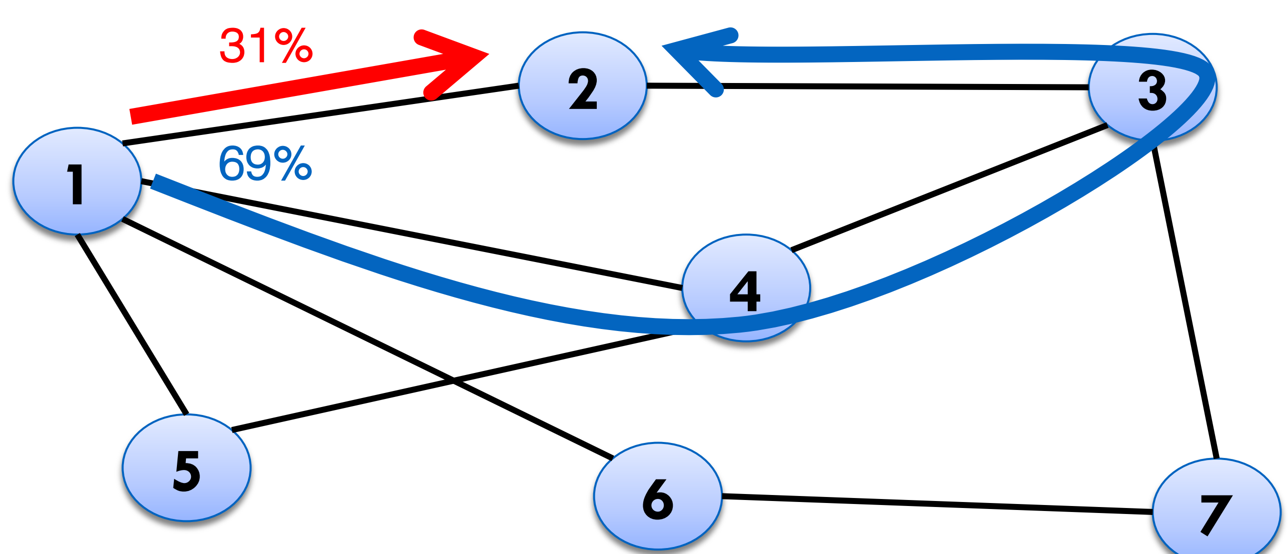
demand matrix

+



network topology

flow optimization
(= linear programming)



routing configuration

Two Sources of Uncertainty

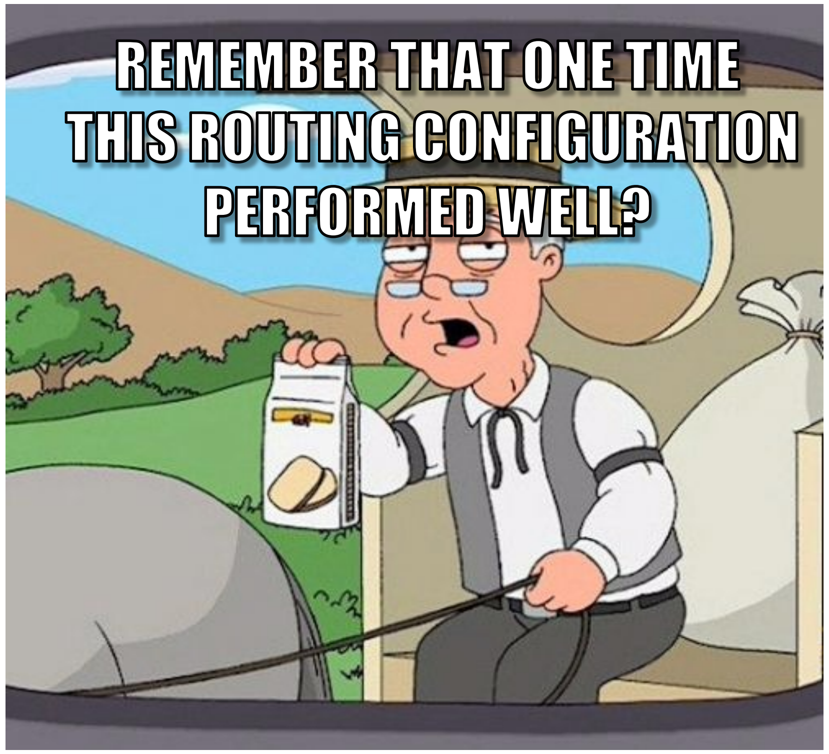
the key focus on this talk

- What if you don't (exactly) know in advance the next demand matrix?

- What if you don't (exactly) in advance how the network topology might change?

to be revisited later

What to do if you don't know the next DM?



This is where ML comes in
[Valadarsky et al., HotNets 2017]

Optimize routing with respect to previously observed traffic patterns

Oblivious Routing

Suboptimal for real-world traffic [Applegate-Cohen, SIGCOMM 2003]

Various combinations of the two have also been proposed
[COPE, SIGCOMM 2006] [SMORE, NSDI 2018]

ML-Guided Routing

Premise

The history of traffic demands contains useful information regarding the future

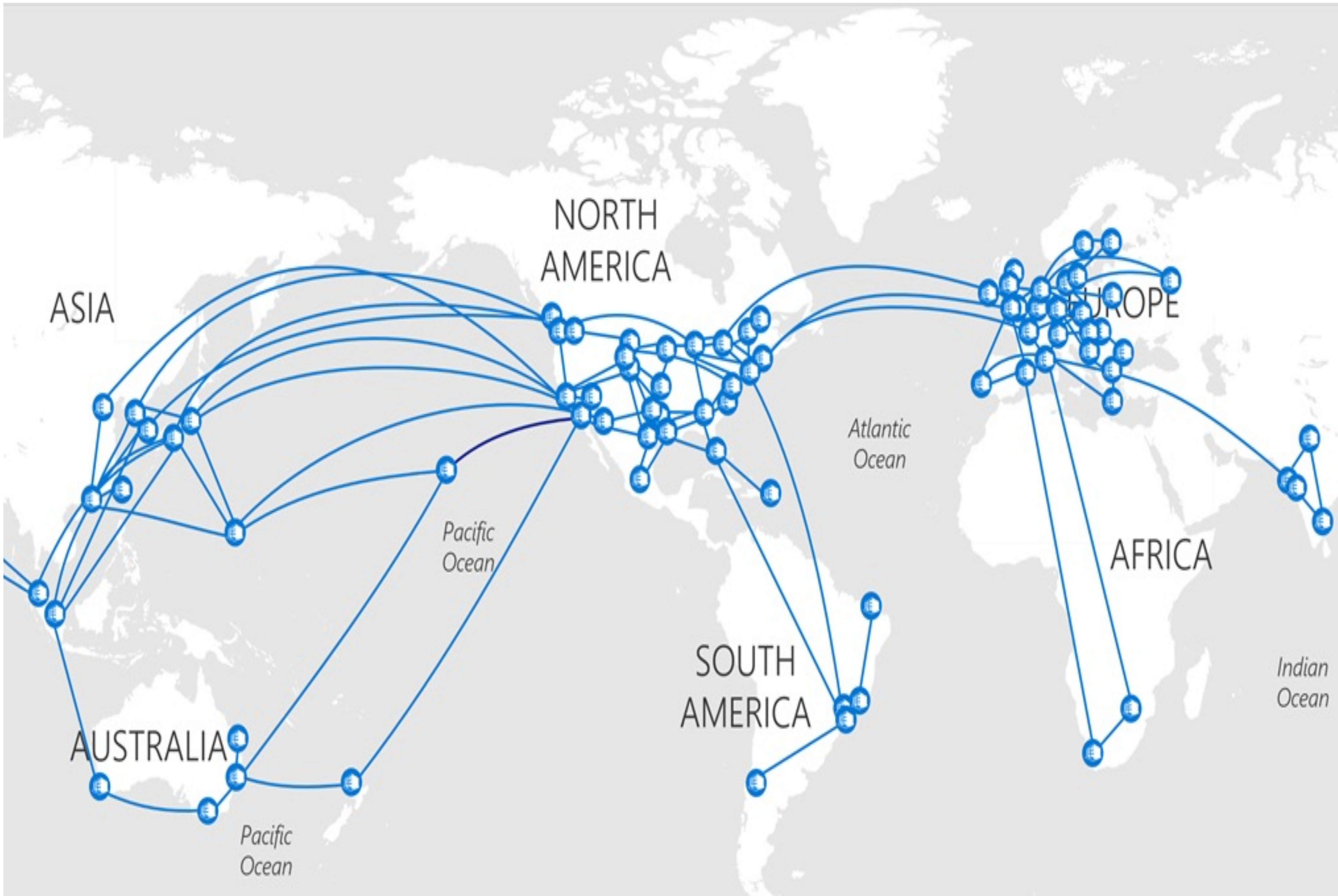
(e.g., temporal patterns, skewness, highly-communicating pairs, …)

ML-Guided Routing

- Continuously observe traffic demands
- Adapt routing with respect to (implicit or explicit) predictions

Wide-Area Networks (WANs)

[SWAN, SIGCOMM 2013] [B4, SIGCOMM 2013]



Modeling ML-Based WAN TE
[Valadarsky et al., HotNets 2017]

Network:

Directed, capacitated graph, G=(V,E).

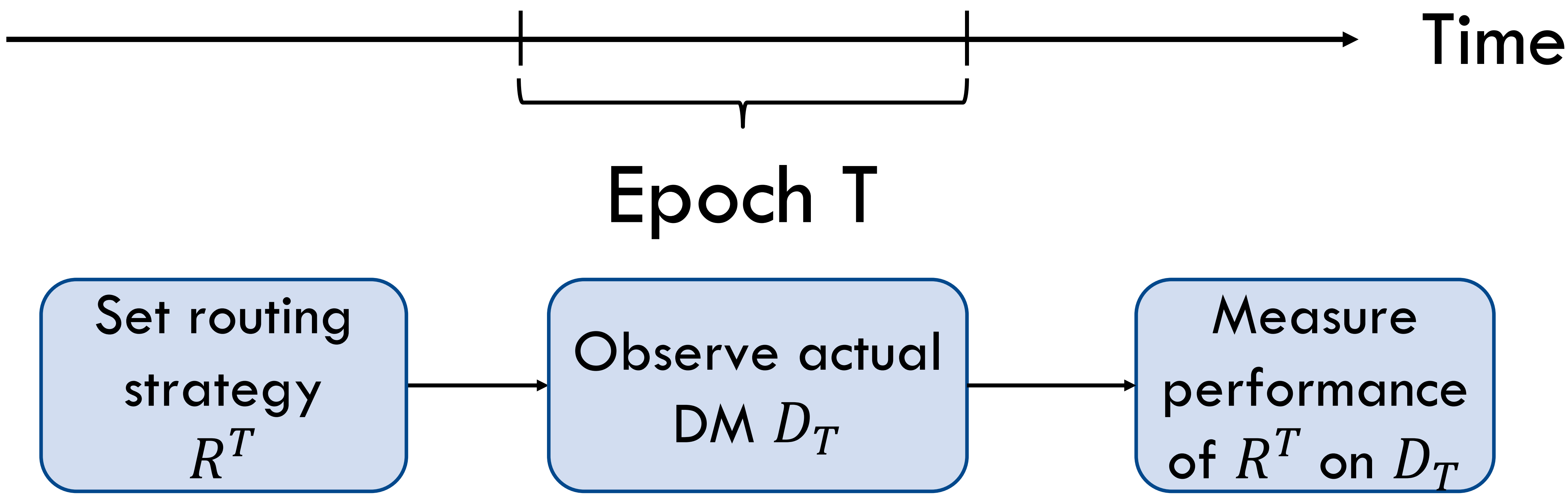
Fixed set of routes ("tunnels") between each pair of nodes.

Routing strategy = the splitting ratios across tunnels for each pair of nodes

Quntifying performance:

maximize the total flow, minimize the maximum link utilization, etc.

Consecutive Time Epochs



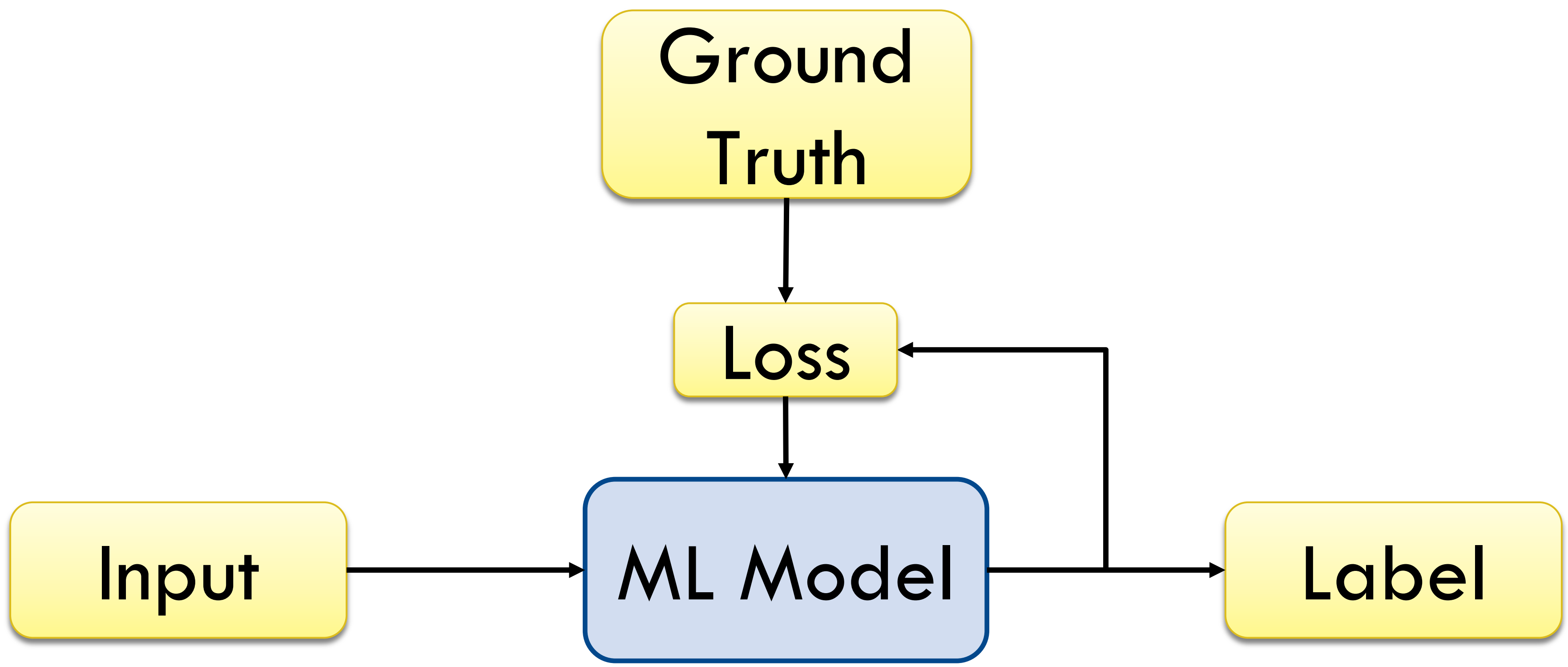
R^T can only depend on the history of *past* DMs

What Should We Learn?

Learn the next DM? → **Supervised Learning**

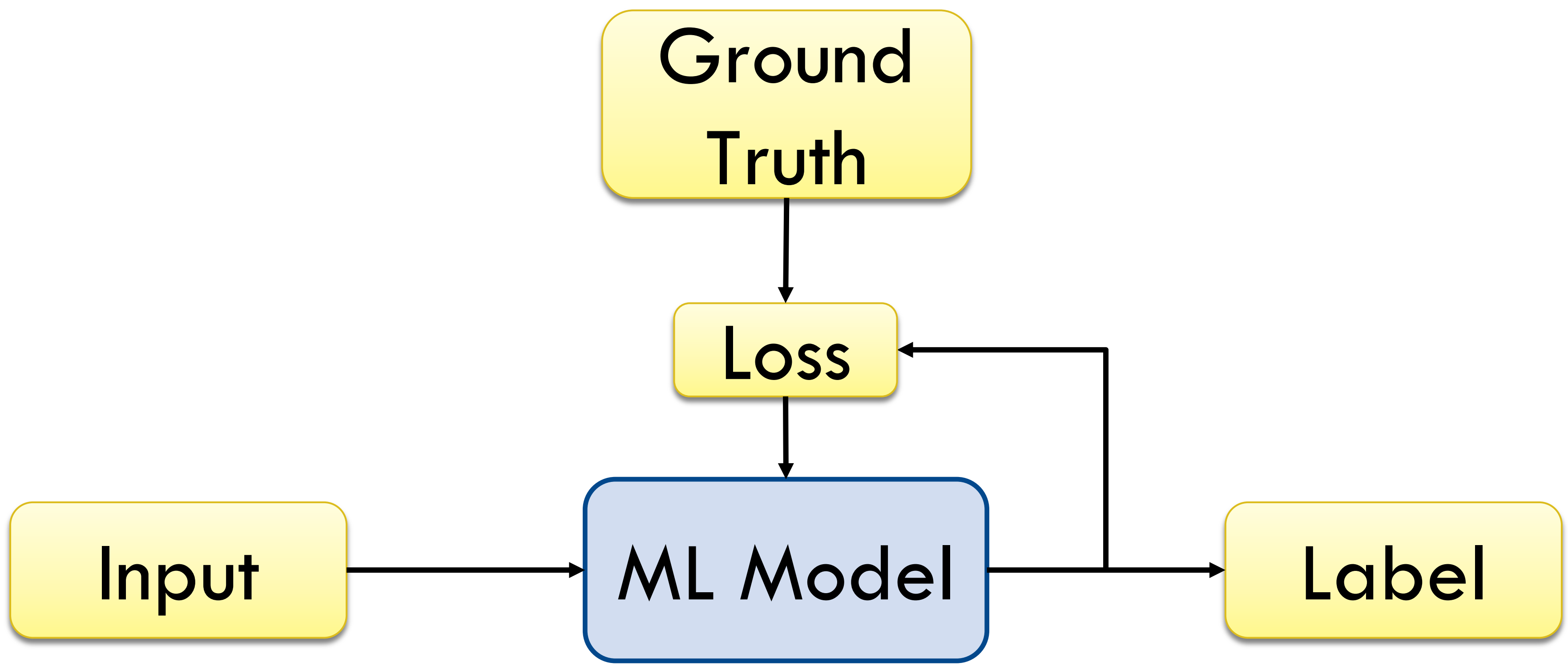
Learn the next routing configuration directly! → **Reinforcement Learning**

Supervised Learning (SL)



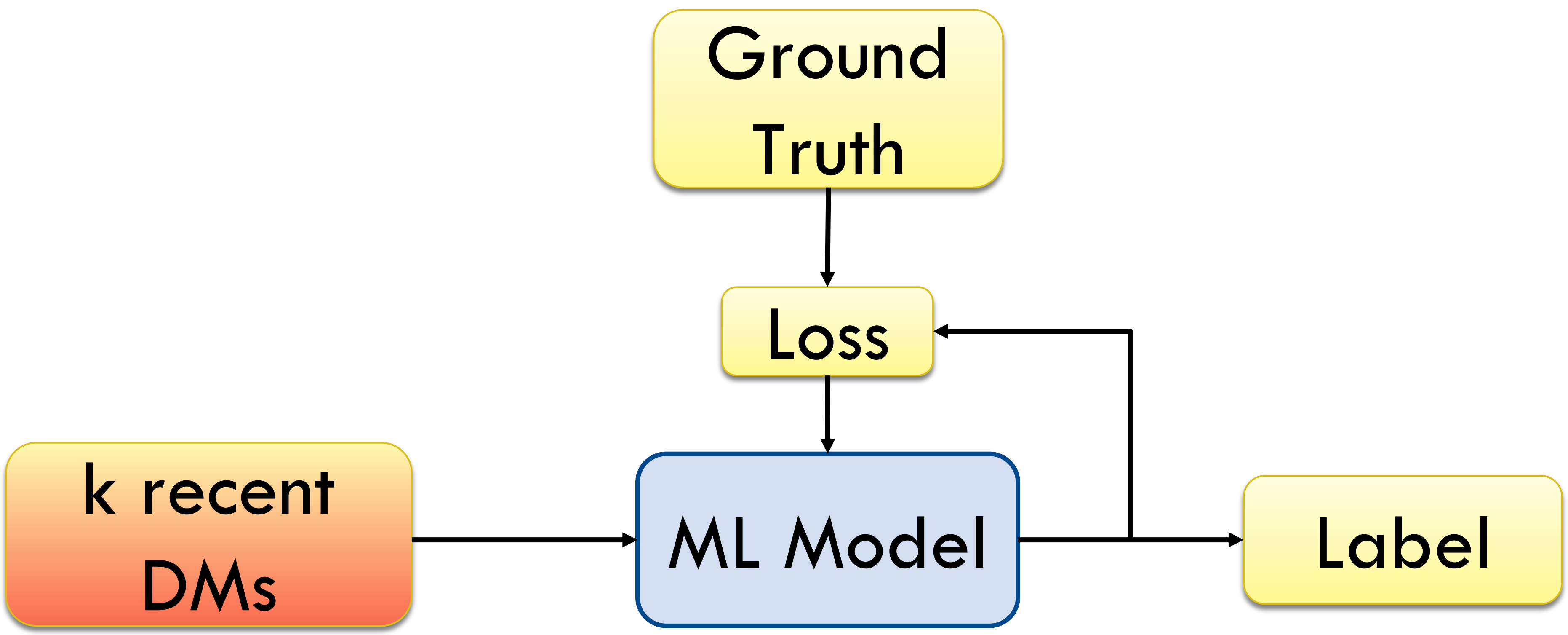
Goal: Minimize the loss wrt to the actual labels

WAN TE as an SL Task



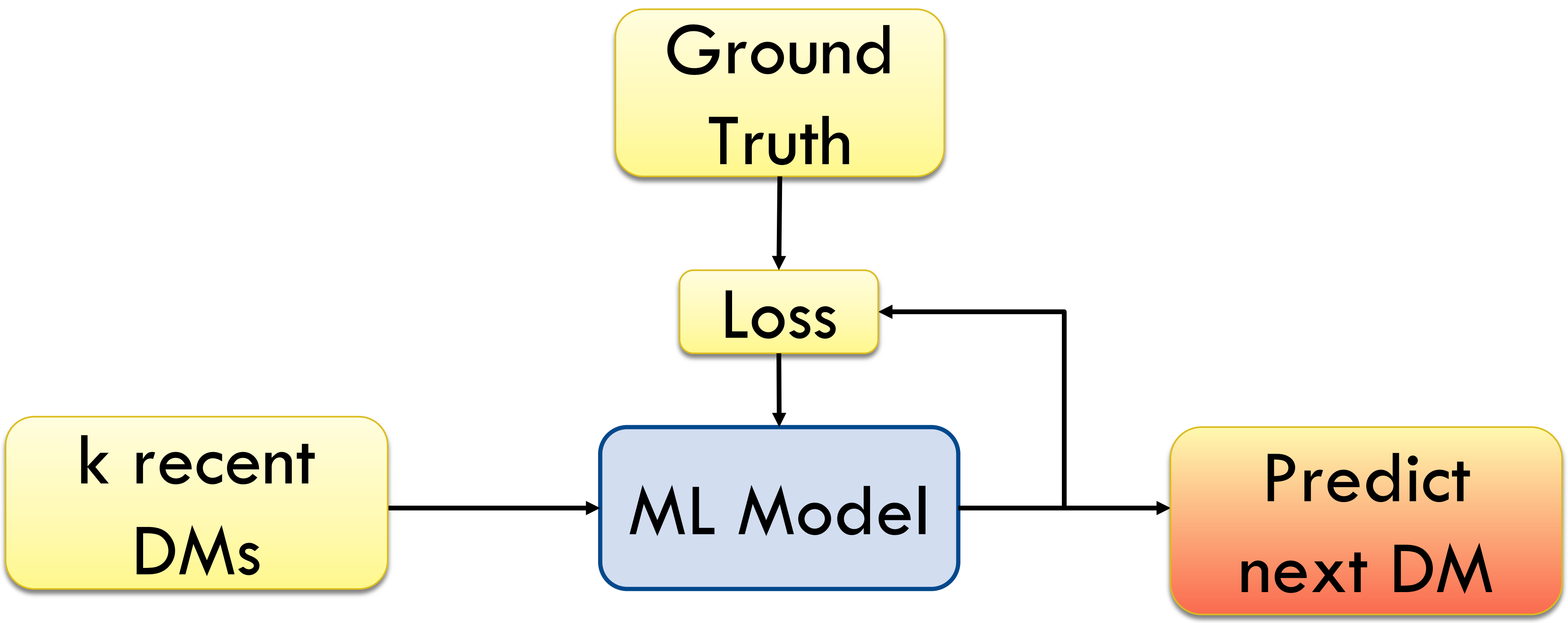
Goal: Minimize the loss wrt to the actual labels

WAN TE as an SL Task



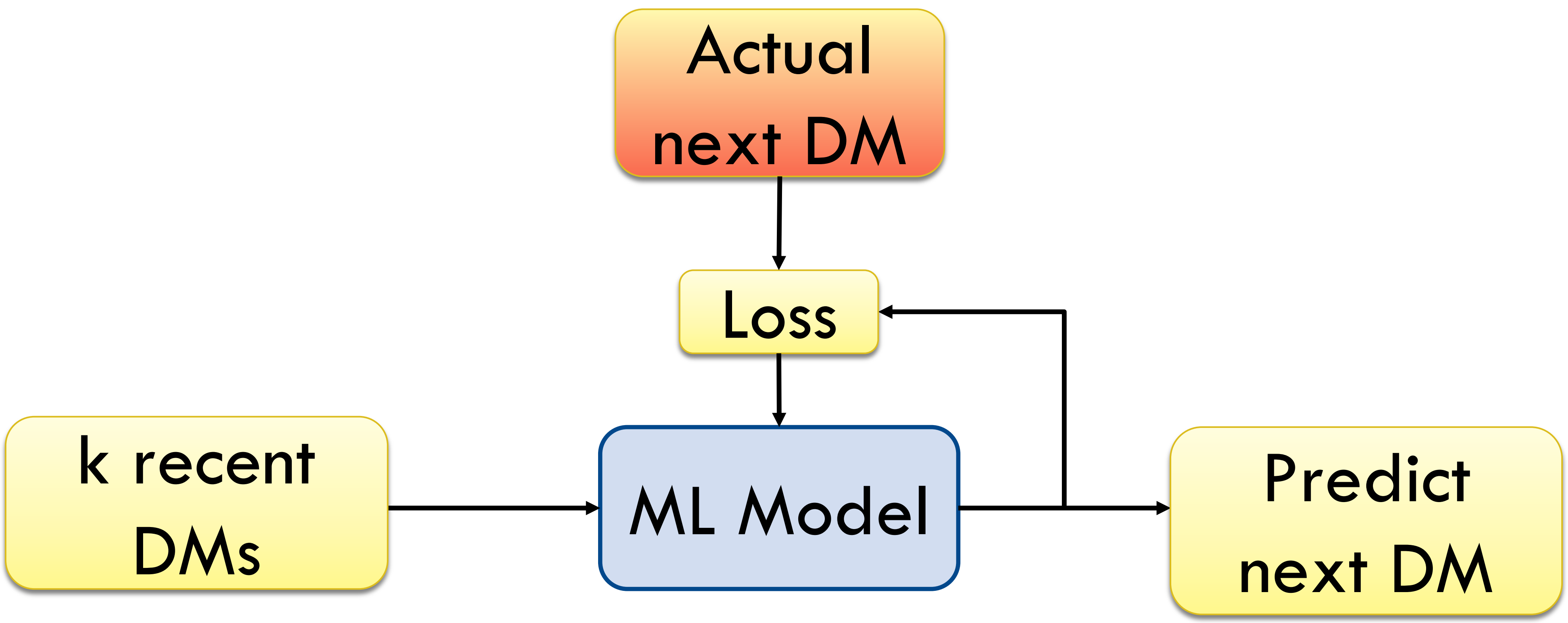
Goal: Minimize the loss wrt to the actual labels

WAN TE as an SL Task



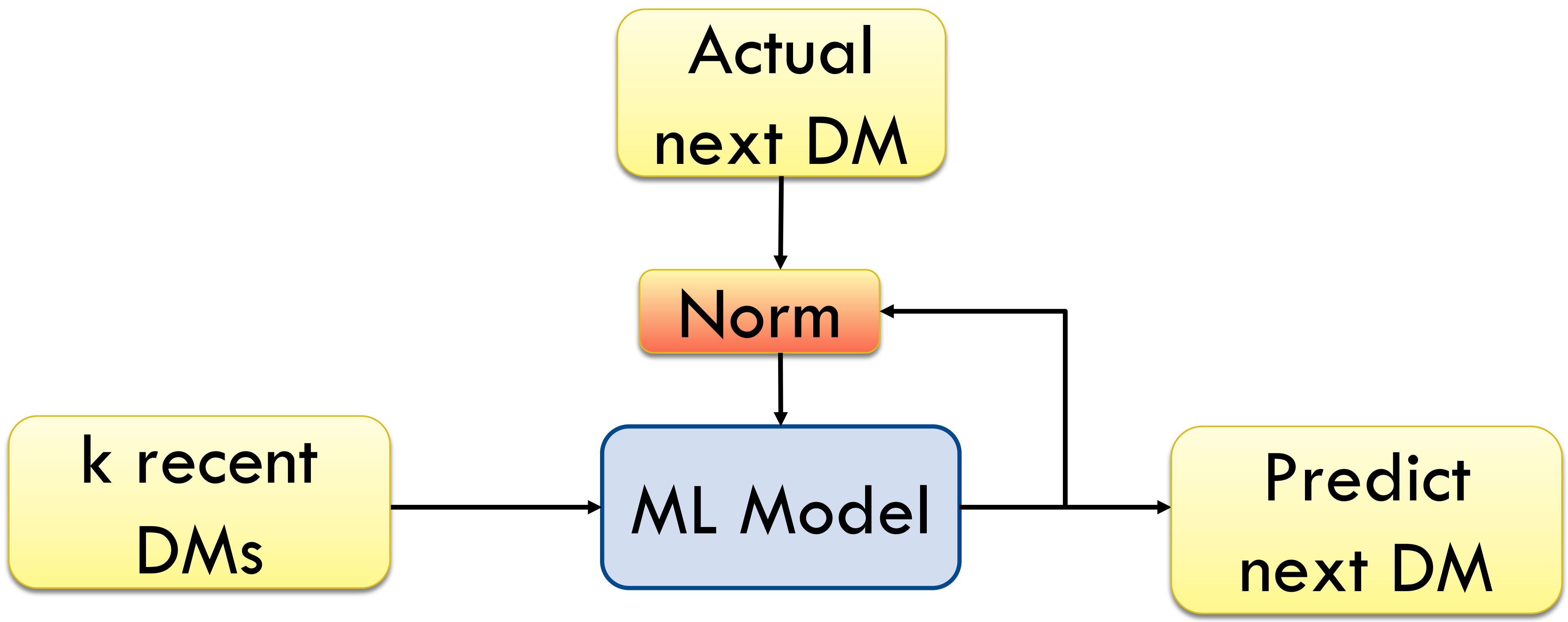
Goal: Minimize the loss wrt to the actual labels

WAN TE as an SL Task



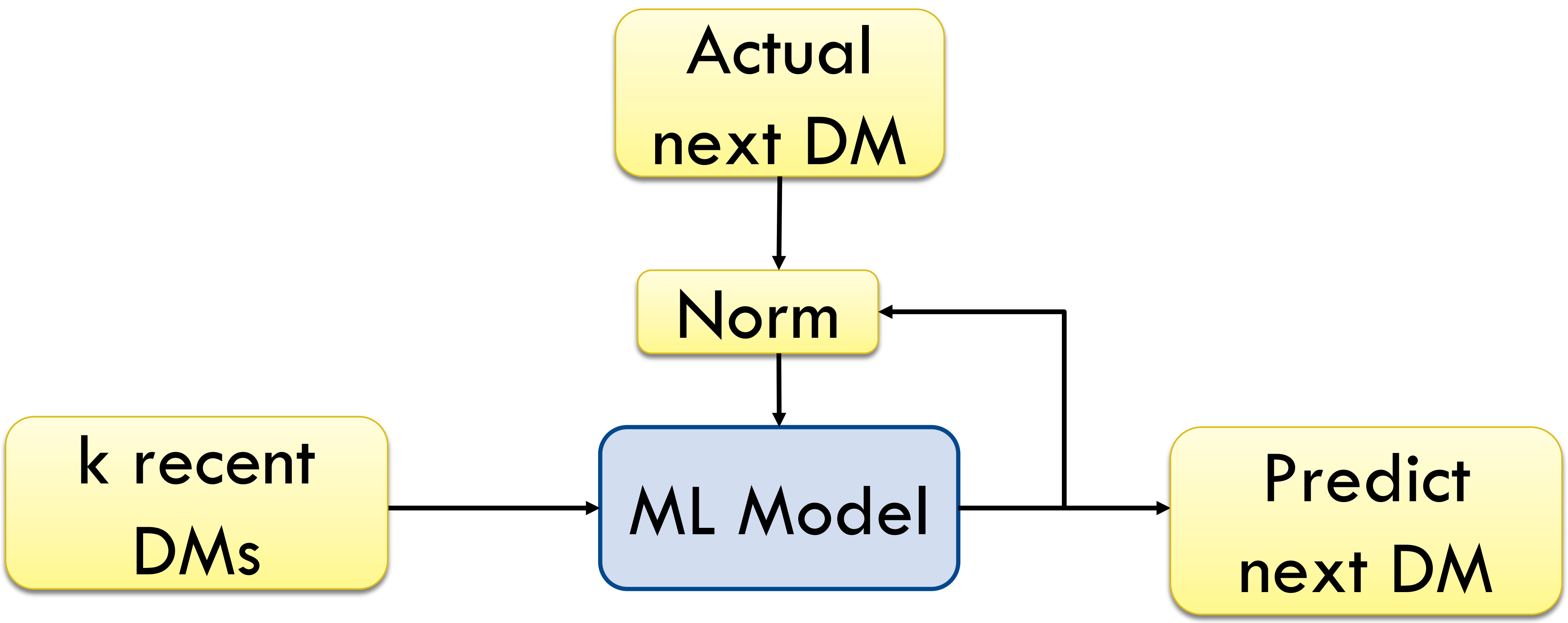
Goal: Minimize the loss wrt to the actual labels

WAN TE as an SL Task



Goal: Minimize the loss wrt to the actual labels

WAN TE as an SL Task



Goal: Minimize the loss wrt to the actual DM

Many Subtleties and Open Questions

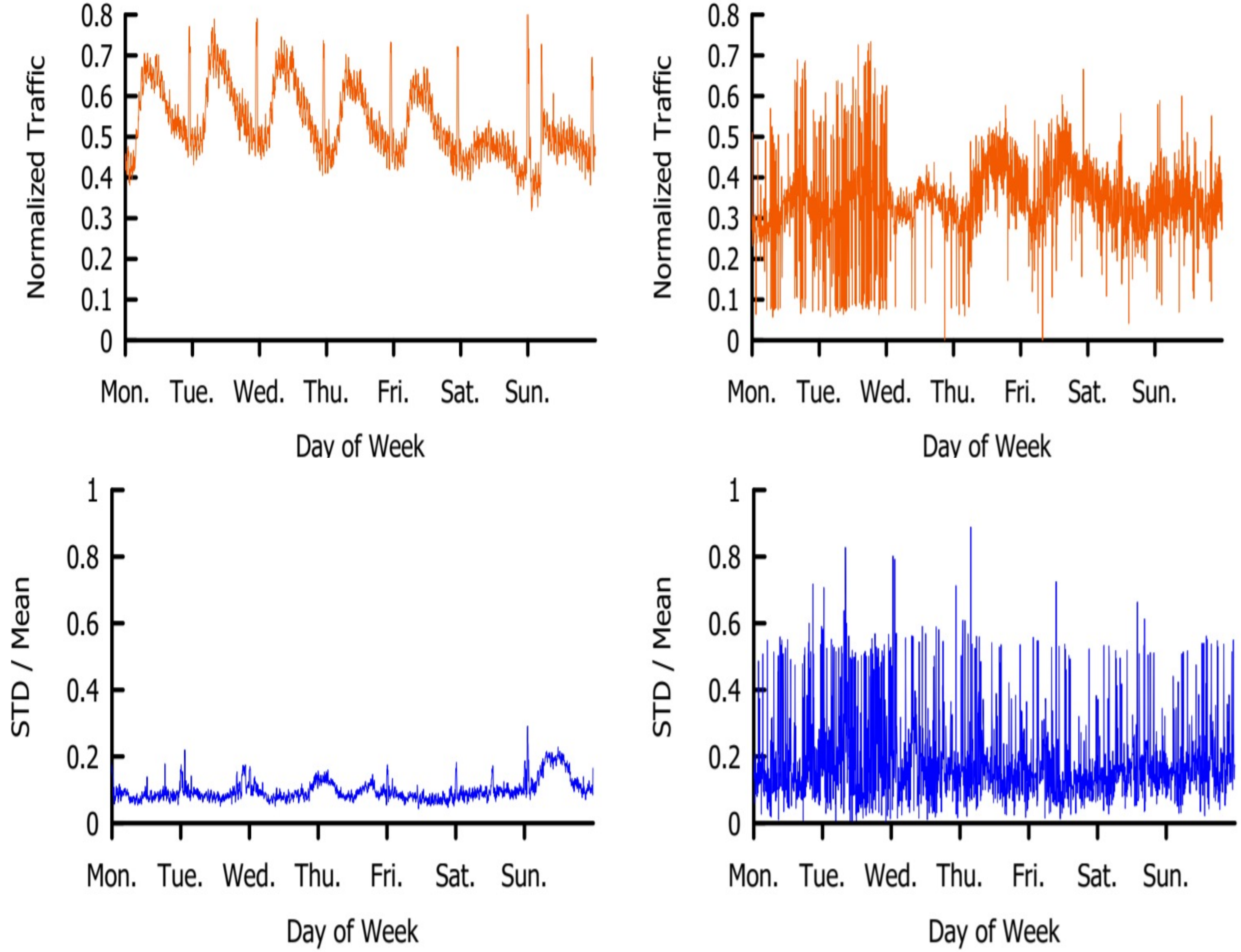
- **<u>The input</u>**: Are past DMs enough? What else is needed (time-of-day? day-of-week? other?)

- **<u>The output</u>**:
  - Should we learn a single DM? A distribution over DMs?
  - Can each DM entry be learned independently? Or other there meaningful correlations?

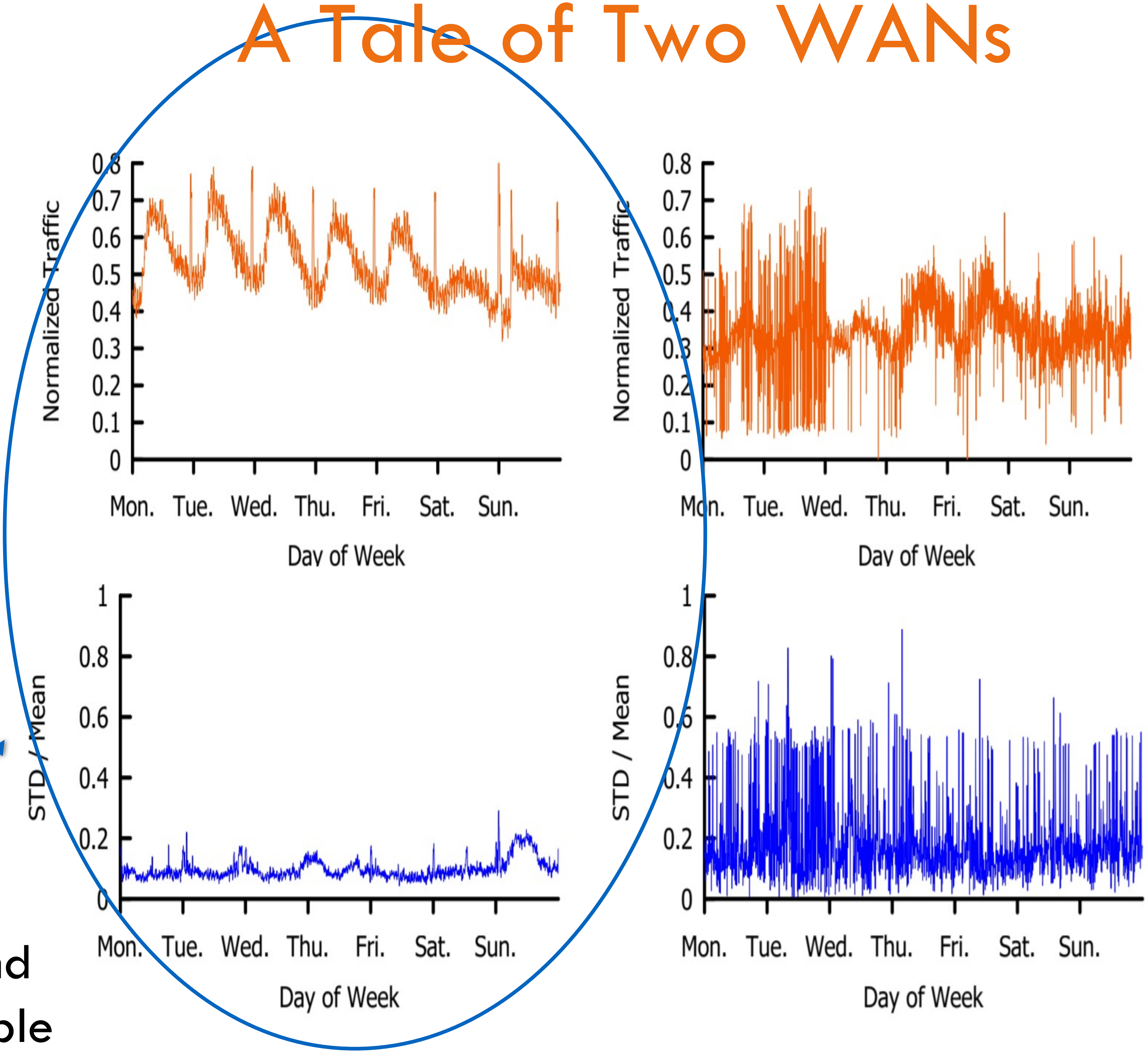
- **<u>The ML model</u>**: Linear regression? Deep Neural Network? Other?

Intriguing theoretical and empirical questions.
Investigation requires extensive data.

A Tale of Two WANs

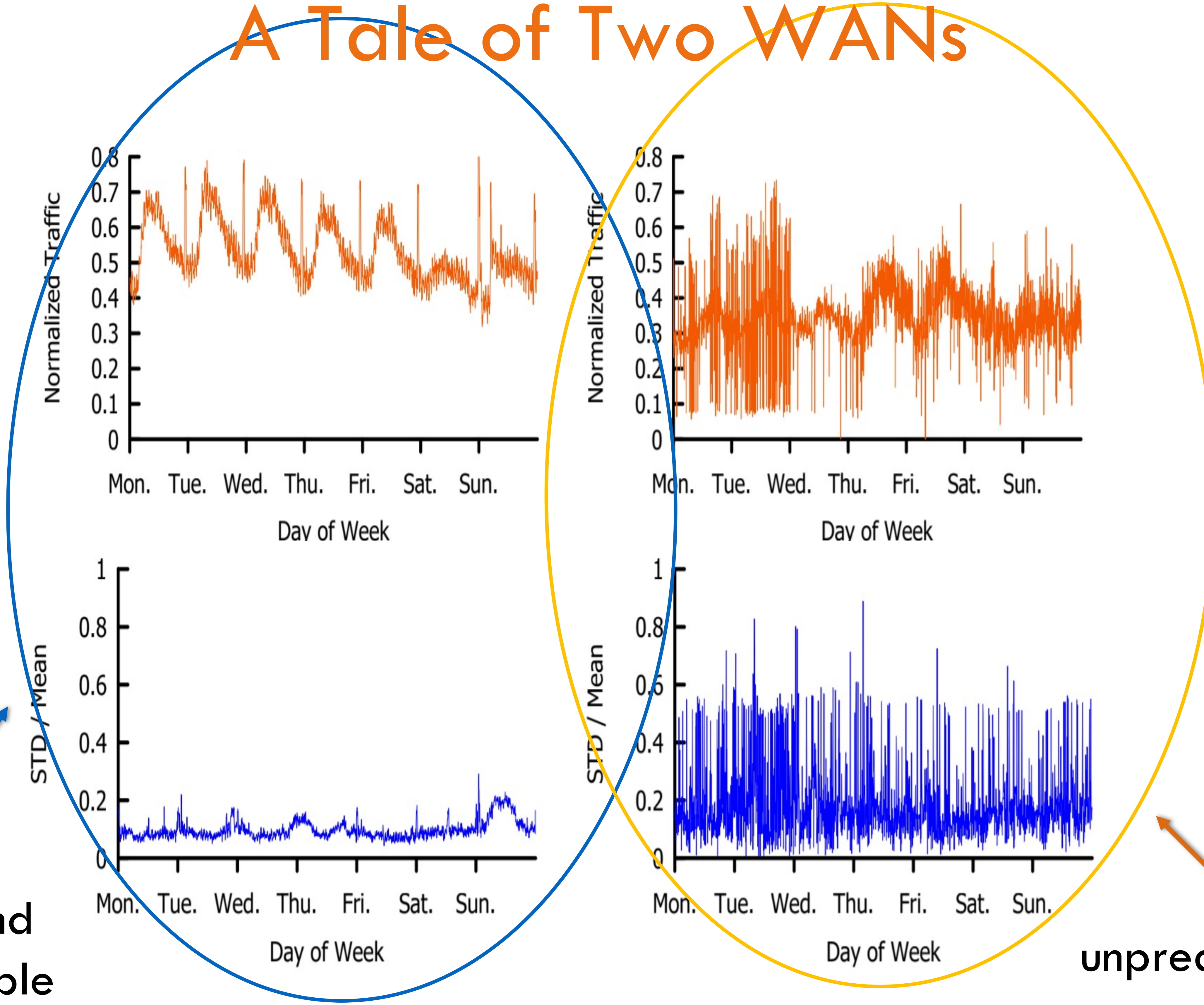


A Tale of Two WANs



small and predictable

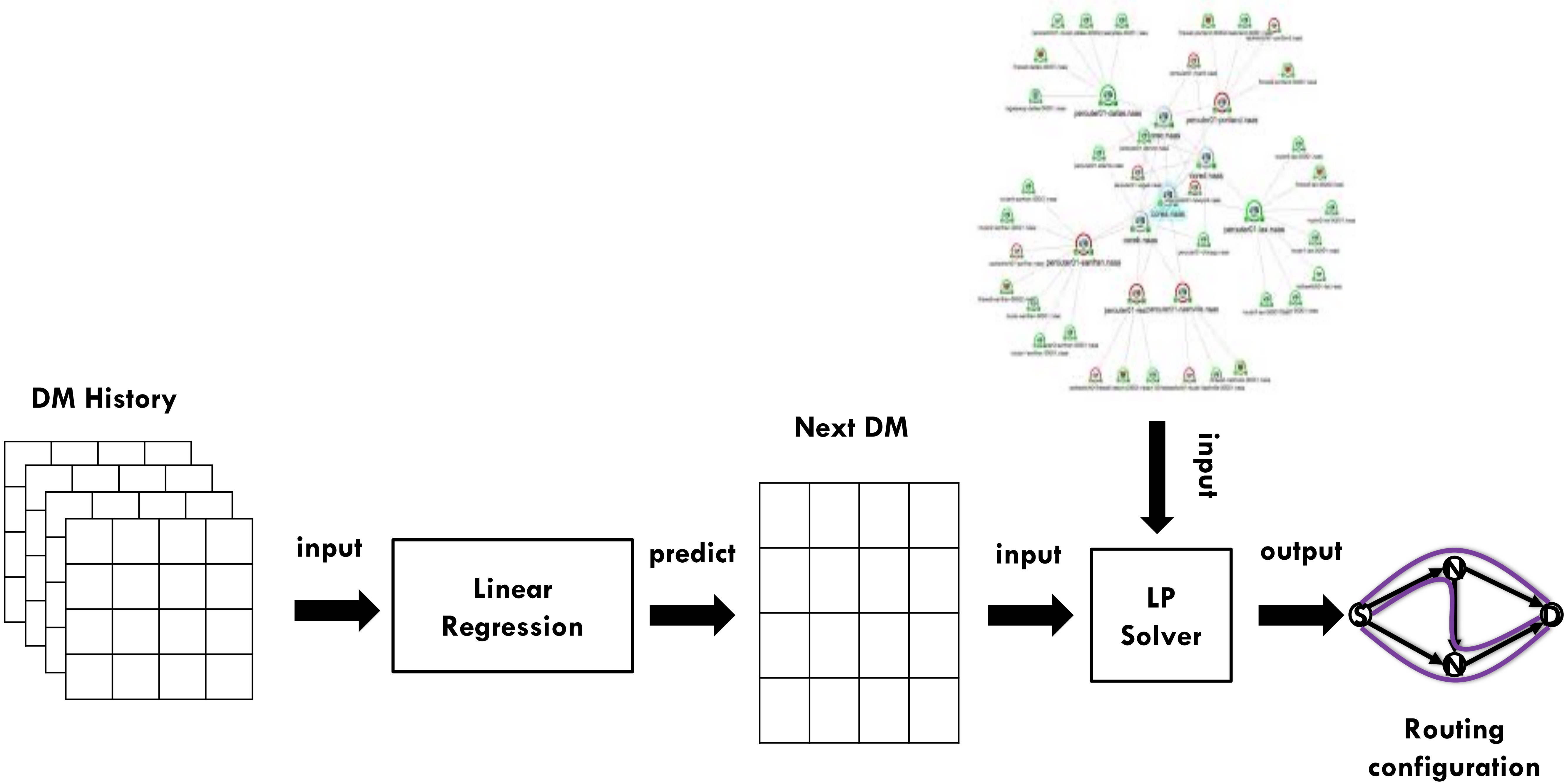
A Tale of Two WANs



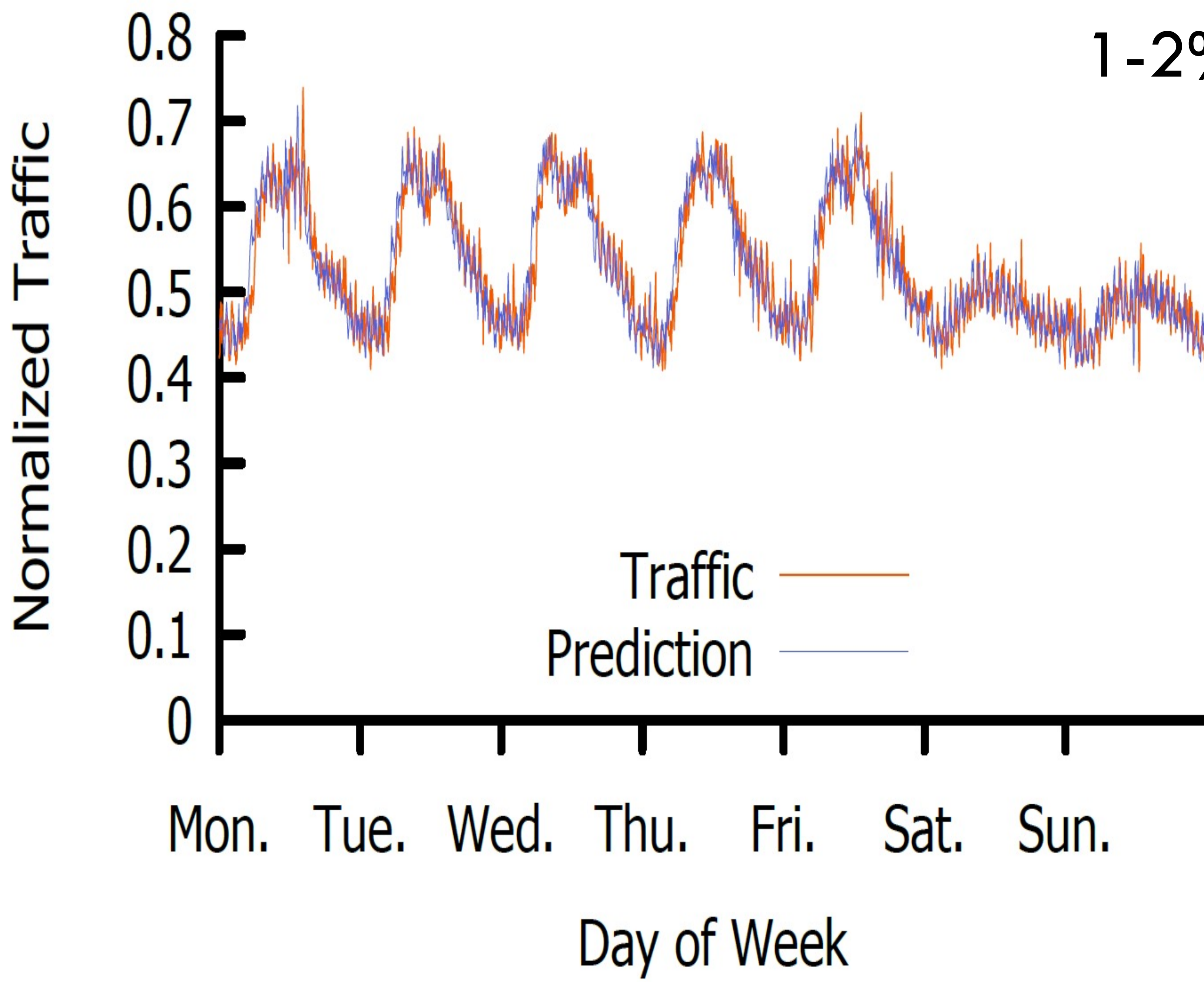
small and predictable

unpredictable

Conventional Approach: Predict (=SL) and Optimize

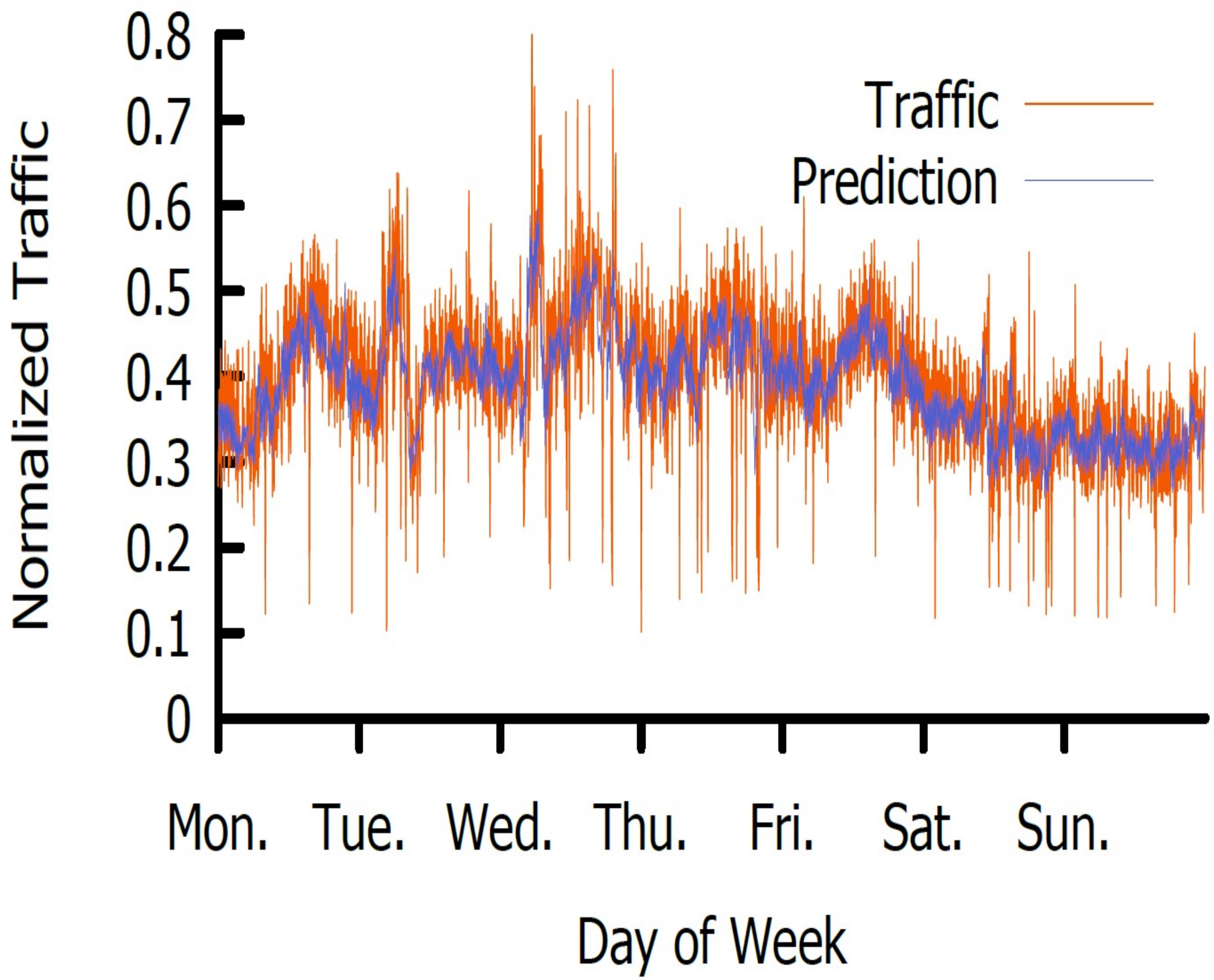


Appropriate when traffic is predictable <u>and</u> the network is small (10s of nodes)

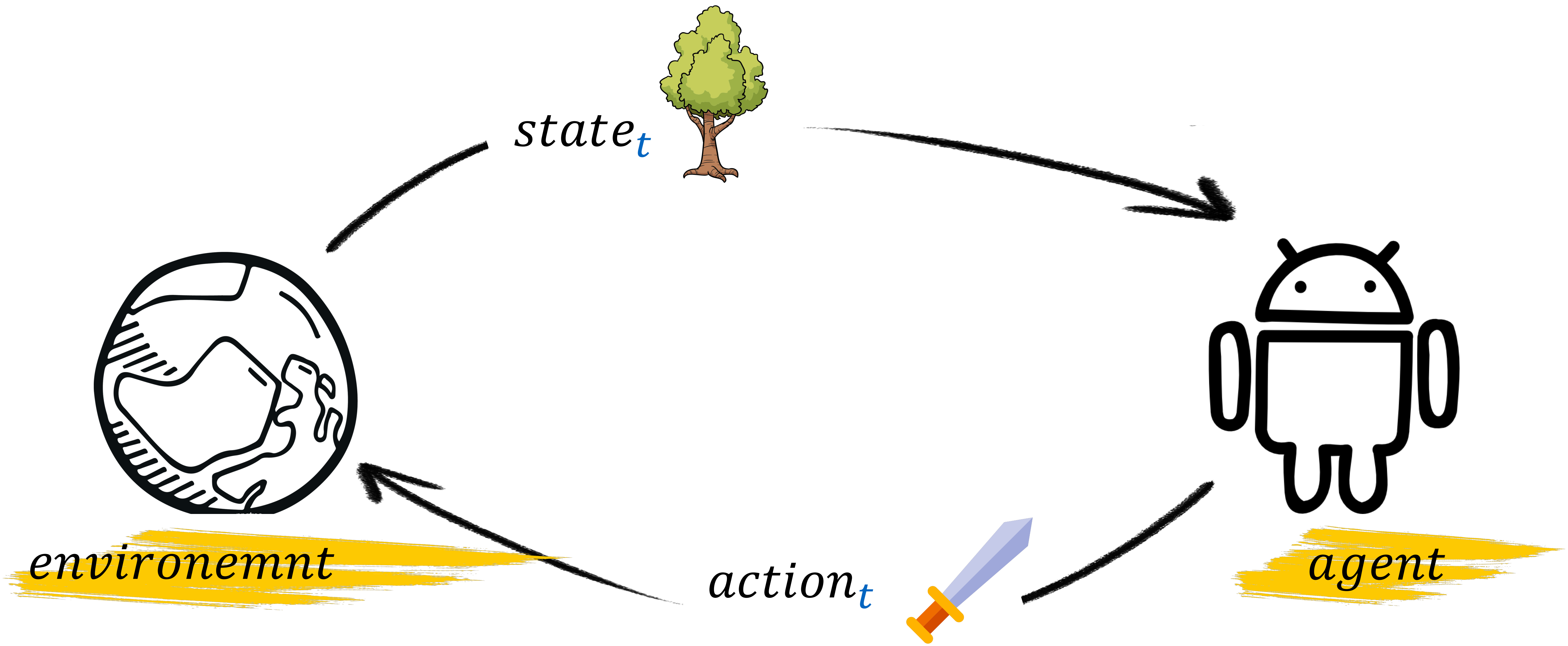


1-2% away from optimum!

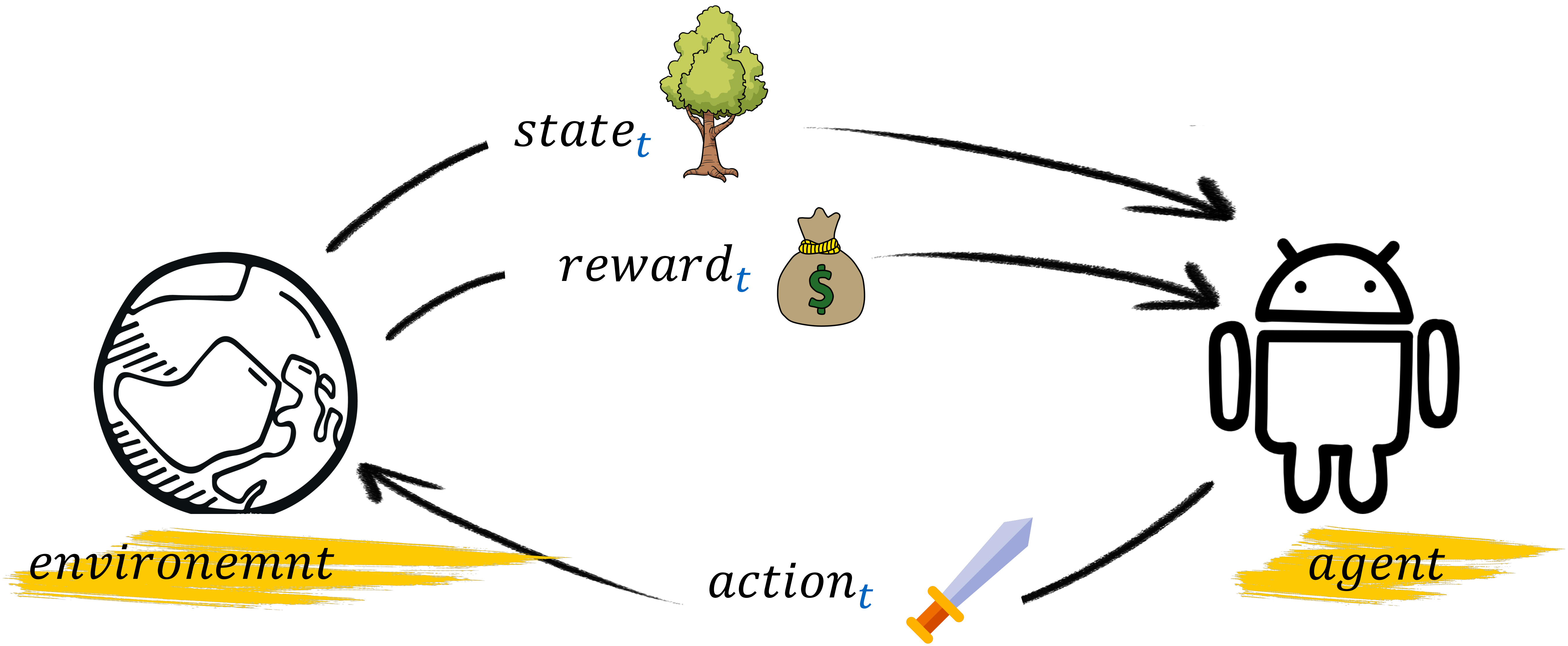
… maybe not when traffic is unpredictable <u>or</u> the network is large (100s of nodes)



Reinforcement Learning (RL)



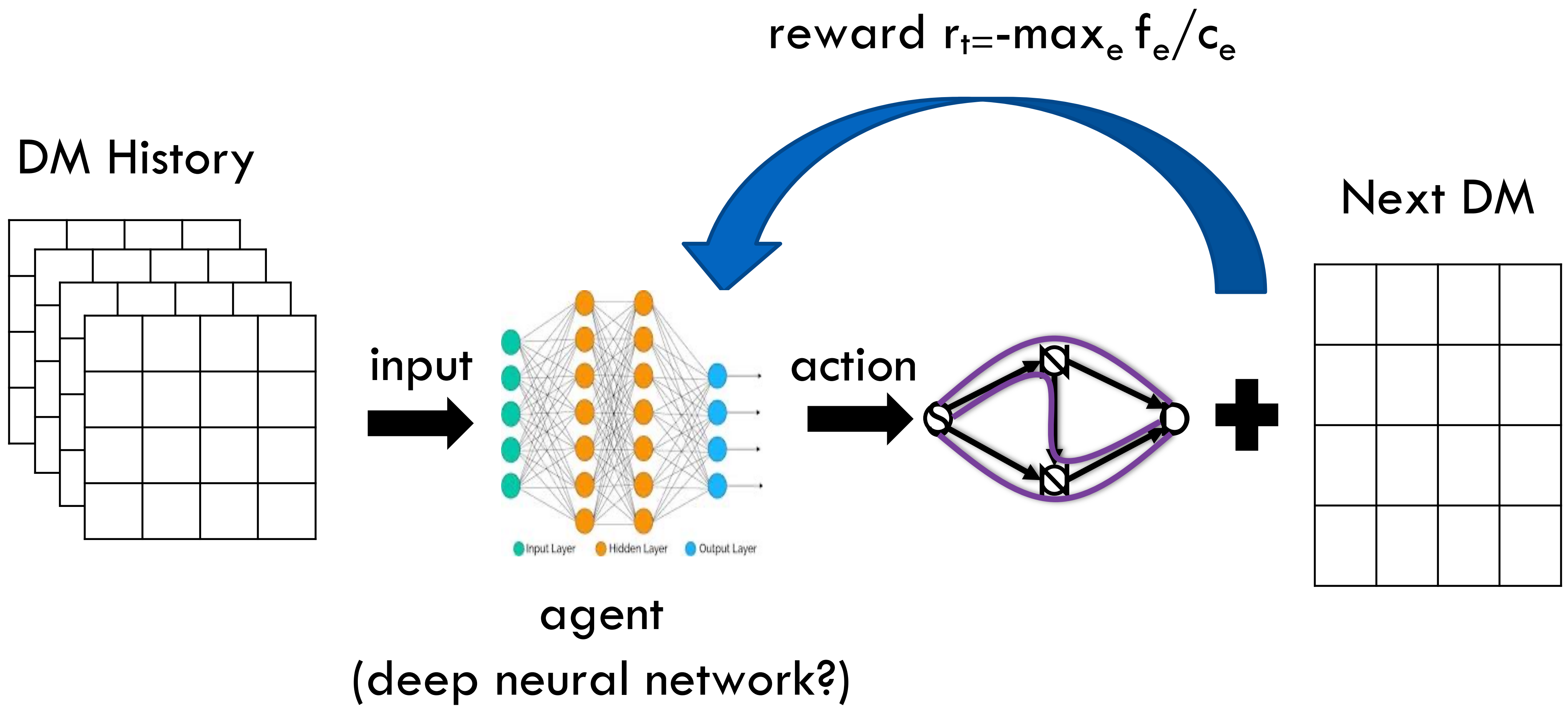
Reinforcement Learning (RL)



Reinforcement Learning (RL)

$$max(E[\sum_t \gamma^t r^t])$$

WAN TE as an RL Talk



Remark: end-to-end computation time in this manner can decrease from O(seconds/minutes) to O(ms)

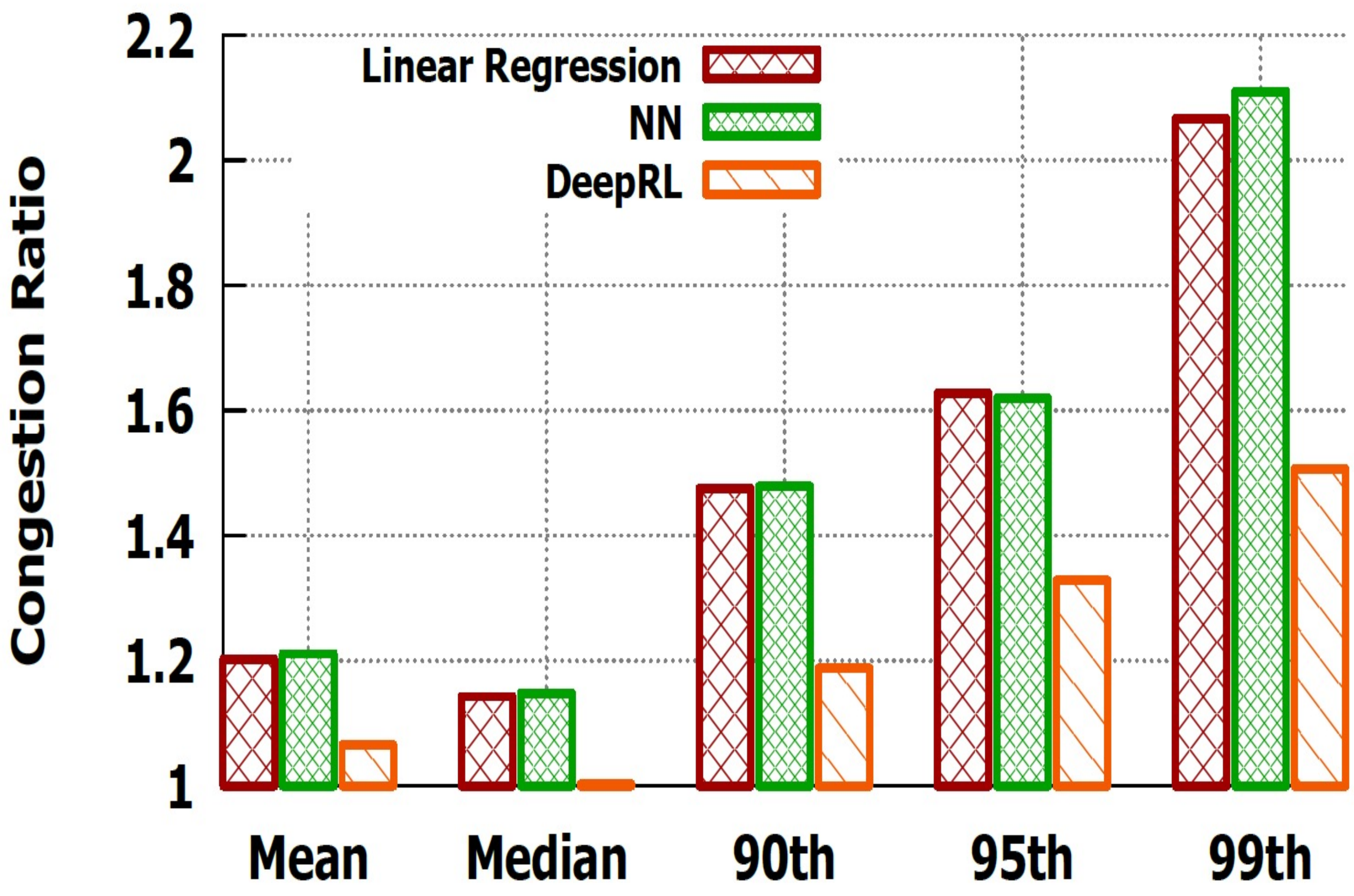
But, this is far from straightforward …

- Scalability! There are two many possible actions
  - |V|x|V|x|E|
  - Instead, learn edge weights that induce routing configurations? [Valadarsky et al., HotNets 2017]

- Deep RL is notorious for high sample complexity need for tuning many parameters

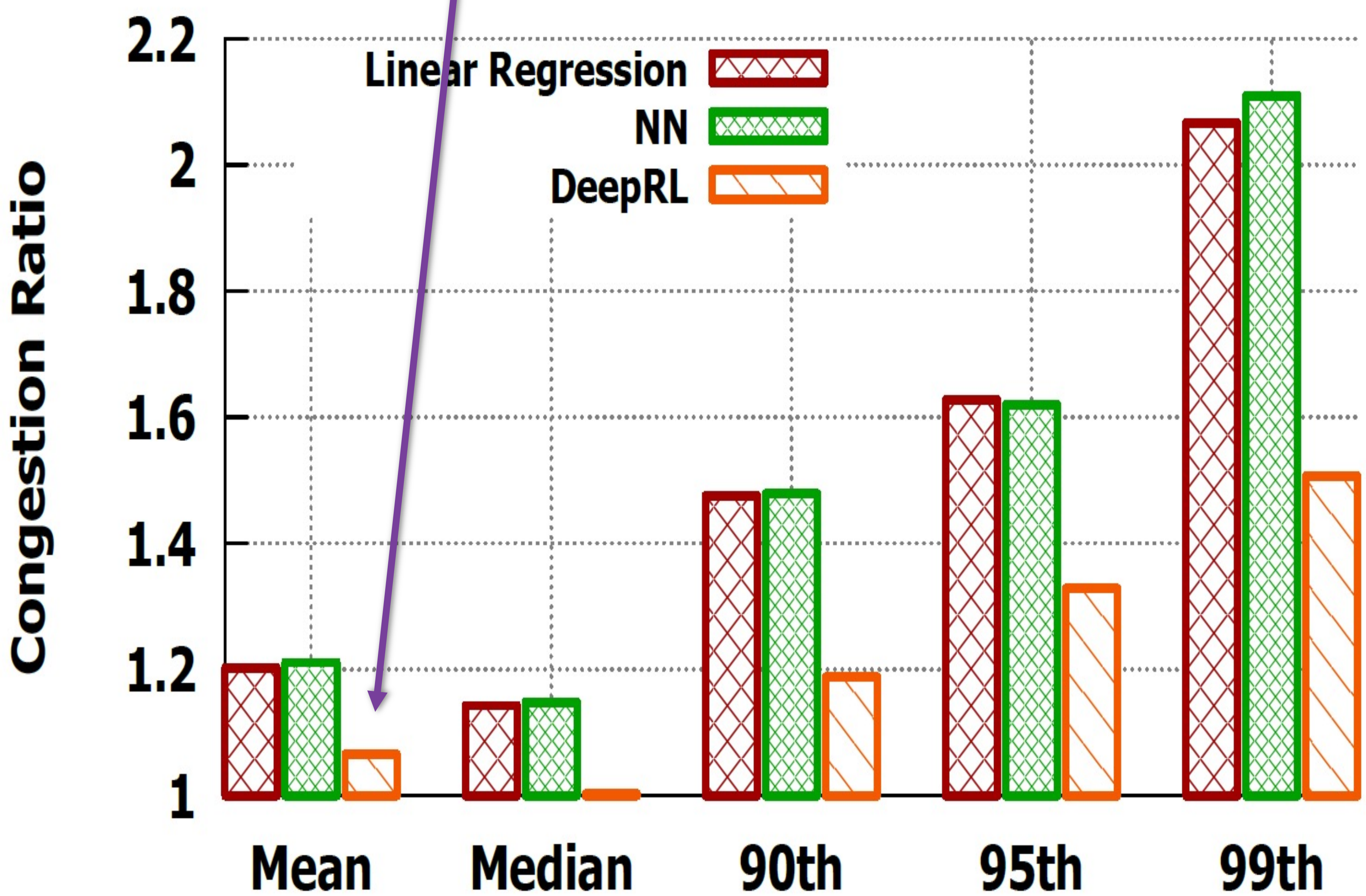
- Are simpler RL approaches feasible? Other approaches?

Promising results for the "unpredictable WAN"



Promising results for the "unpredictable WAN"

6% from optimum!



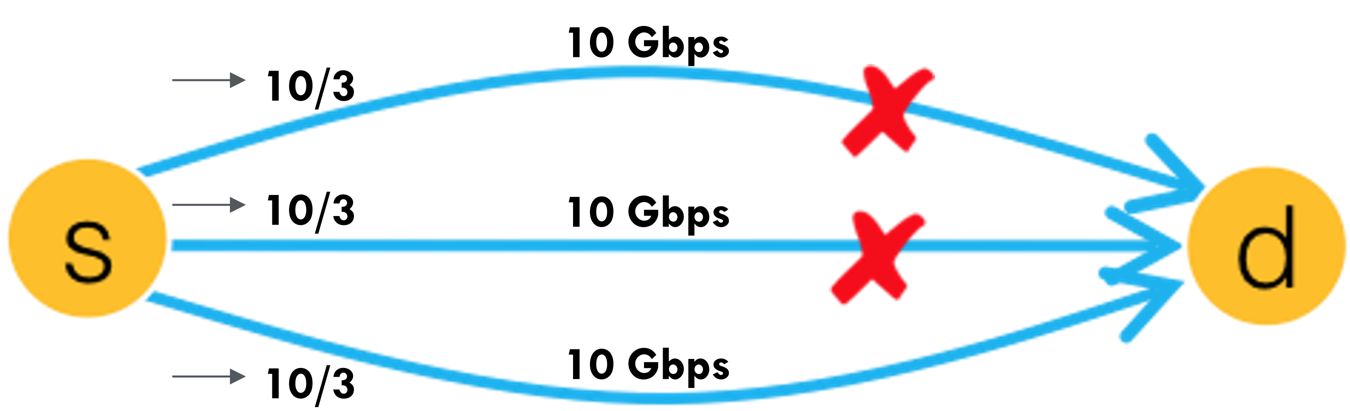
Computation time decreased from seconds/minutes to ±15ms!

More Open Questions!

- Learn the tunnels?
- What lies between "Predict and Optimize" and Deep RL? Is there a sweet spot?

What about network failures?

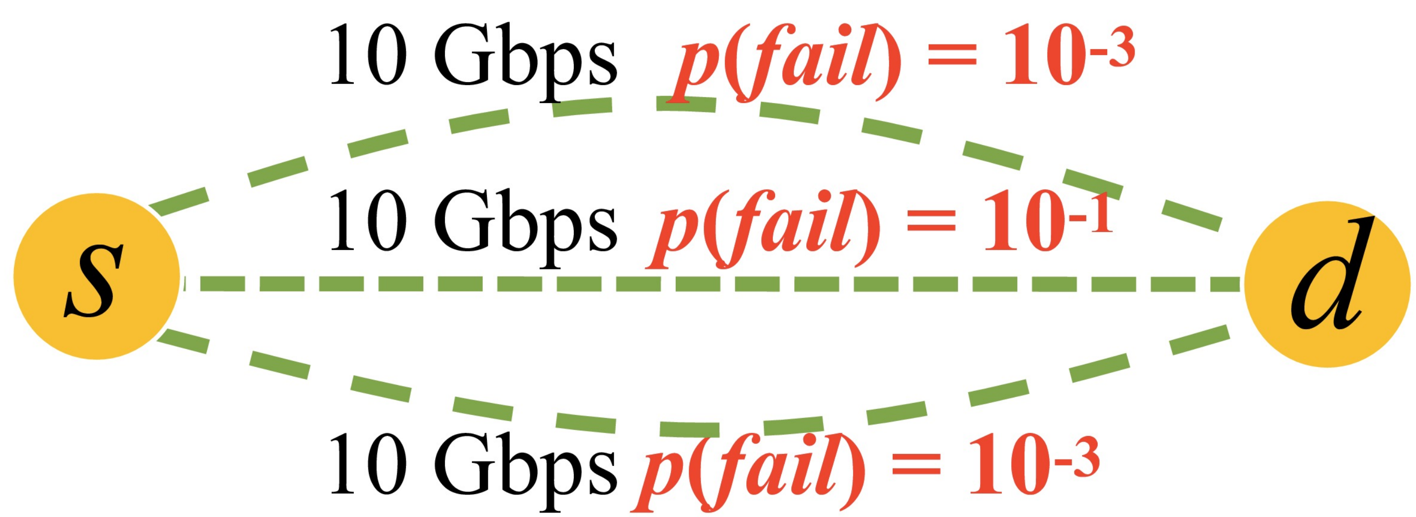
Efficient network utilization and high resiliency to failures can be at odds!



One approach [FFC, SIGCOMM 2014]:
Operate the WAN at lower utilization so as to be capable of withstanding 1-2 concurrent link failures.

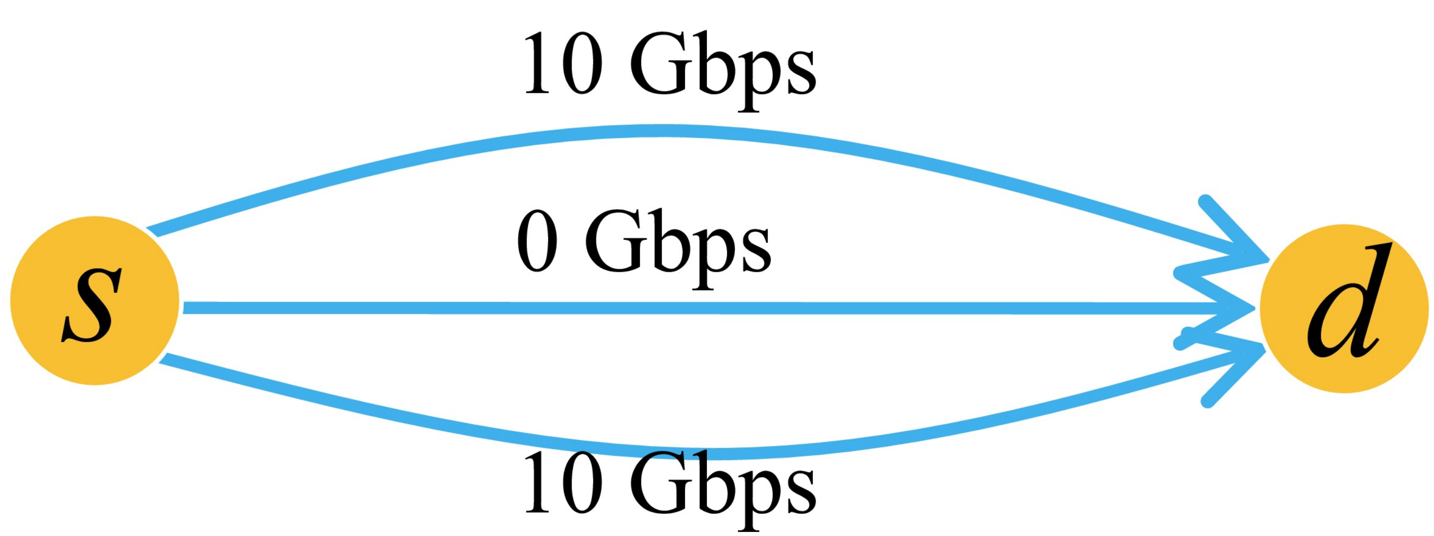
A Data-Driven Approach:

[TeaVaR, SIGCOMM 2019]



Assign failure probabilities to links/nodes.

Optimize with respect to these and target availability goals.



But where do these probabilities come from?

- Operators have visibility into failure patterns and dynamics [Ghobadi-Mahajan, ICM 2016]
  - failures are more probable during working hours
  - failures can be predicted by sudden drops in optical signal quality, (50% chance of outage within 1 hour, 70% within 1 day)

- Can this wealth of empirical data be exploited to explicitly reason about the probability of different failure scenarios?

- A supervised learning challenge!

- Or maybe optimize end-to-end instead?

Dual Problem: Capacity Planning

Check out Zhu et al.'s upcoming SIGCOMM paper on "Network Planning with Deep Reinforcement Learning"

Conclusion

- ML-based WAN TE is still young
- Many fascinating open questions
- Looking forward to seeing progress on this important real-world challenge soon!

Thank You