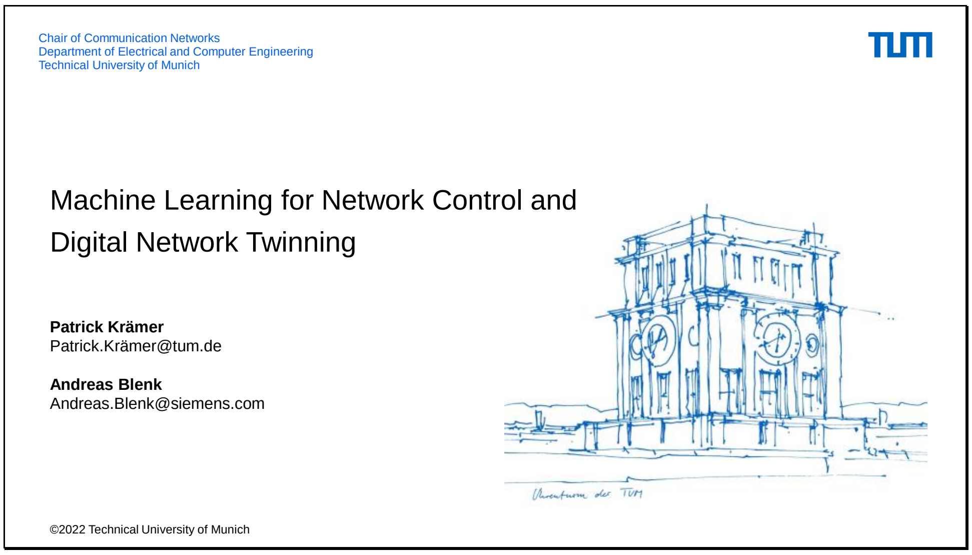
Slide 1

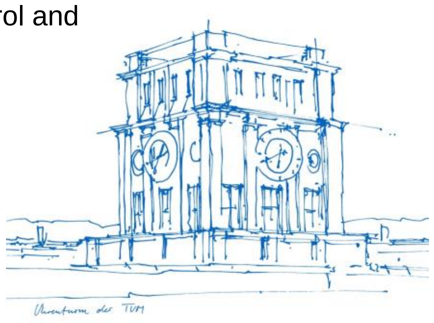


Chair of Communication Networks
Department of Electrical and Computer Engineering
Technical University of Munich

Machine Learning for Network Control and Digital Network Twinning

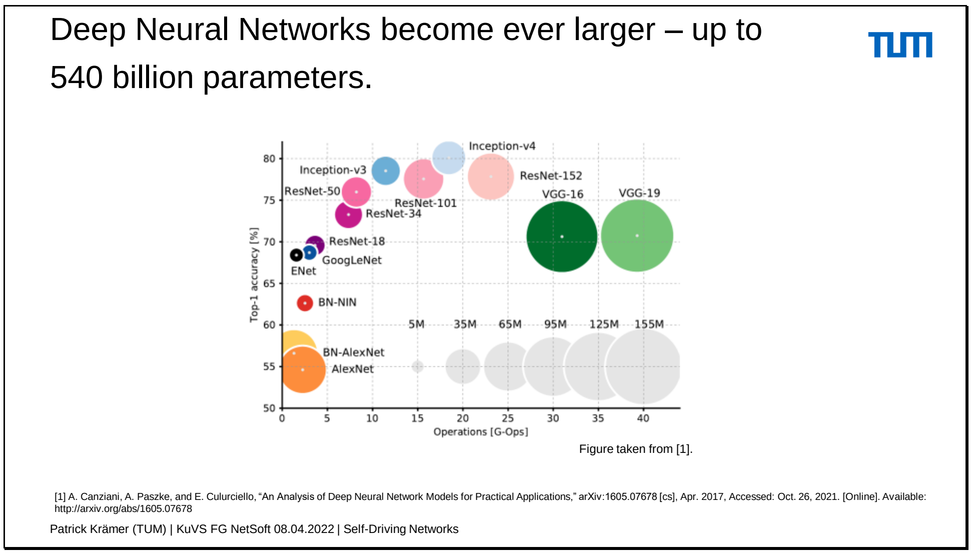
Patrick Krämer
Patrick.Krämer@tum.de

Andreas Blenk
Andreas.Blenk@siemens.com



- Disclaimer: Ongoing work.

Slide 2



Deep Neural Networks become ever larger – up to 540 billion parameters.

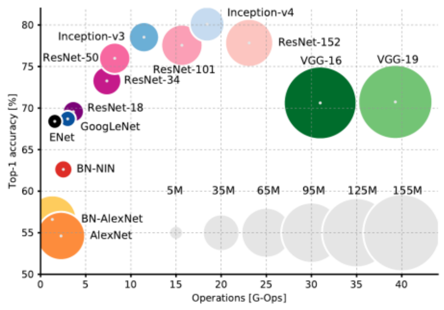


Figure taken from [1].

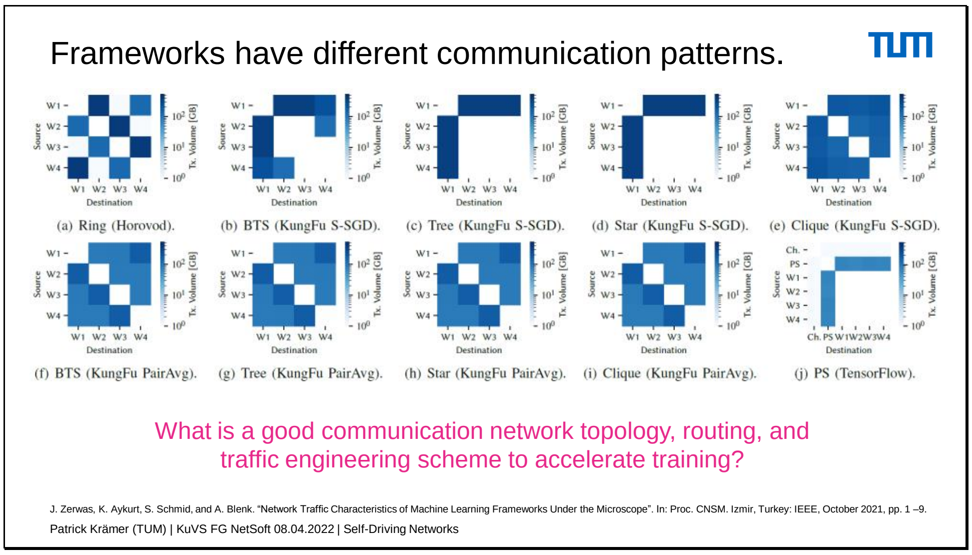
[1] A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," arXiv:1605.07678 [cs], Apr. 2017, Accessed: Oct. 26, 2021. [Online]. Available: http://arxiv.org/abs/1605.07678

- New NN from Google (PALM) with 540 Billion Parameters (or 2 TB Float32).
- Training on thousands of GPUS/TPUs
- OpenAI has a Kubernets Cluster with 7500 Nodes.
- Models and data does not fit into a single machine -> Distributed ML necessary.
- More and more companies with ML and Big-Data use-cases exists -> Distributed ML and data analysis.
- Networking important, reduces cost of operation, reduces training and turn-around times.

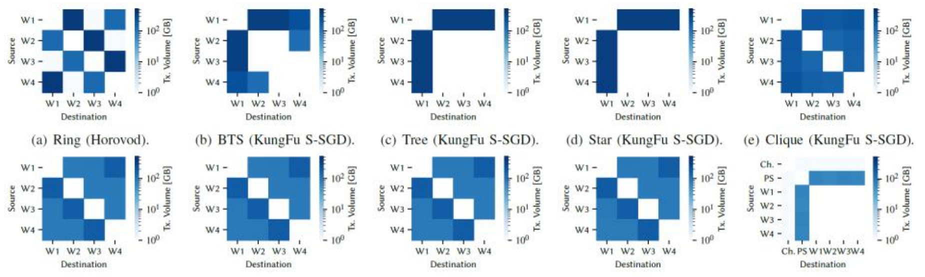
- Different ML Tools available: Horovod, Ray, Tensorflow, Pytorch, KungFu, etc.

https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html
https://openai.com/blog/scaling-kubernetes-to-7500-nodes/

Slide 3



Frameworks have different communication patterns.



(a) Ring (Horovod). (b) BTS (KungFu S-SGD). (c) Tree (KungFu S-SGD). (d) Star (KungFu S-SGD). (e) Clique (KungFu S-SGD).

(f) BTS (KungFu PairAvg). (g) Tree (KungFu PairAvg). (h) Star (KungFu PairAvg). (i) Clique (KungFu PairAvg). (j) PS (TensorFlow).

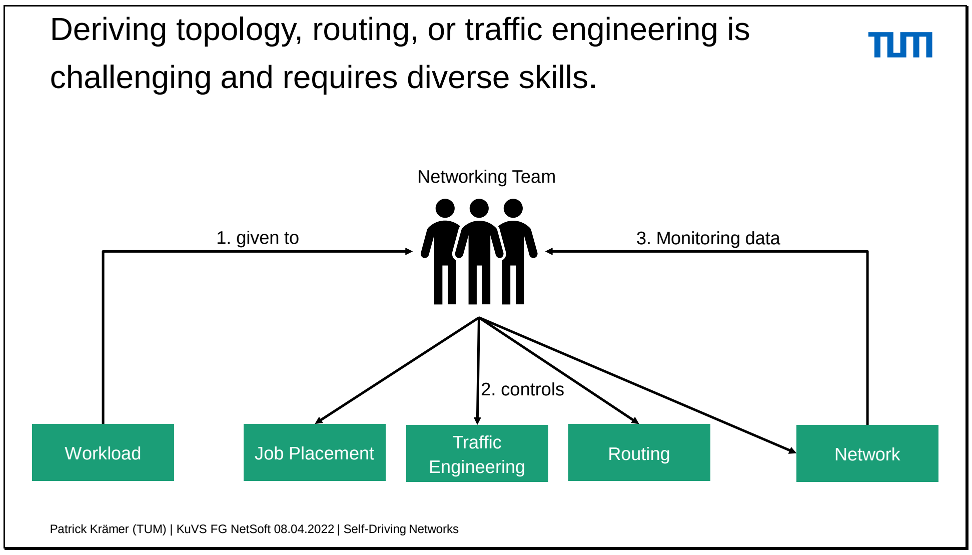
What is a good communication network topology, routing, and traffic engineering scheme to accelerate training?

J. Zerwas, K. Aykurt, S. Schmid, and A. Blenk. "Network Traffic Characteristics of Machine Learning Frameworks Under the Microscope". In: Proc. CNSM. Izmir, Turkey: IEEE, October 2021, pp. 1 –9.

Patrick Krämer (TUM) | KuVS FG NetSoft 08.04.2022 | Self-Driving Networks

- ML tools have varying connection patterns.
- What is the best network topology, routing and TE scheme for these tools?
- We can adapt the topology.
- We can adapt routing.
- We can adapt TE.
- We can adapt the placement of individual machines, data, and jobs.

Slide 4



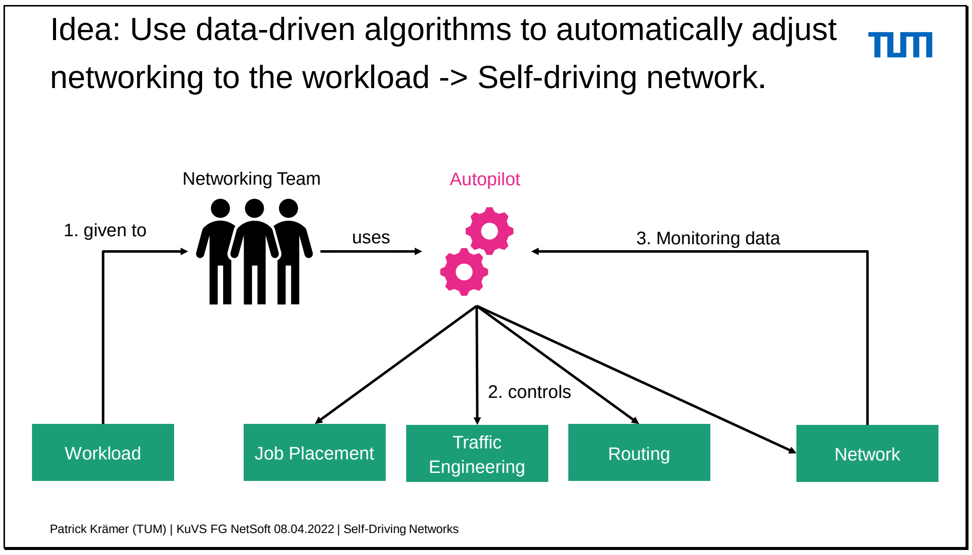
Deriving topology, routing, or traffic engineering is challenging and requires diverse skills.

Patrick Krämer (TUM) | KuVS FG NetSoft 08.04.2022 | Self-Driving Networks

- Some workload is dumped to the networking team. Their job is to improve KPIs for that workload, e.g., the training time.
- The workload might not be known initially, and the team must find out what the application or task looks like on the network.
- Based on the effect in the network, the team can discern the placement of individual application components to hosts, the routing scheme, traffic engineering settings, and maybe even the physical network topology.
- Each of those aspects influence the others, i.e., changing the placement might change how to configure the TE or routing, and network, and vice verca.
- After specifying each component, the team gets monitoring data back from the network. The team can use the monitoring data to adjust the components.
- Challenging and time consuming.
- Requires highly trained specialists → Are rare and hard to find, especially for start-ups.

What happens if a new workload is dumped to the team?
https://www.zeit.de/wirtschaft/2022-04/fachkraeftemangel-startups-studie-deutschland?utm_referrer=https%3A%2F%2Fwww.google.de%2F#comment-form

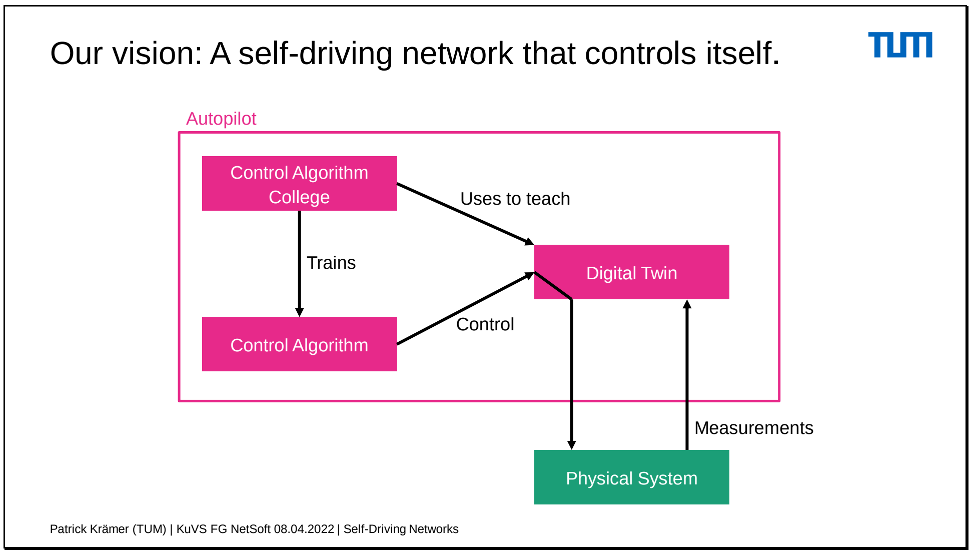
Slide 5



Idea: Use data-driven algorithms to automatically adjust networking to the workload -> Self-driving network.

- Make the network self-driving (in certain bounds). One could also say: Network control as a service.
- The networking team uses an Autopilot that automates the optimization process of the network.
- The networking team operates at a higher level, low level configurations and finding solutions to optimization problems does the autopilot.
- This would enable smaller teams to tune their network to their specific workload.

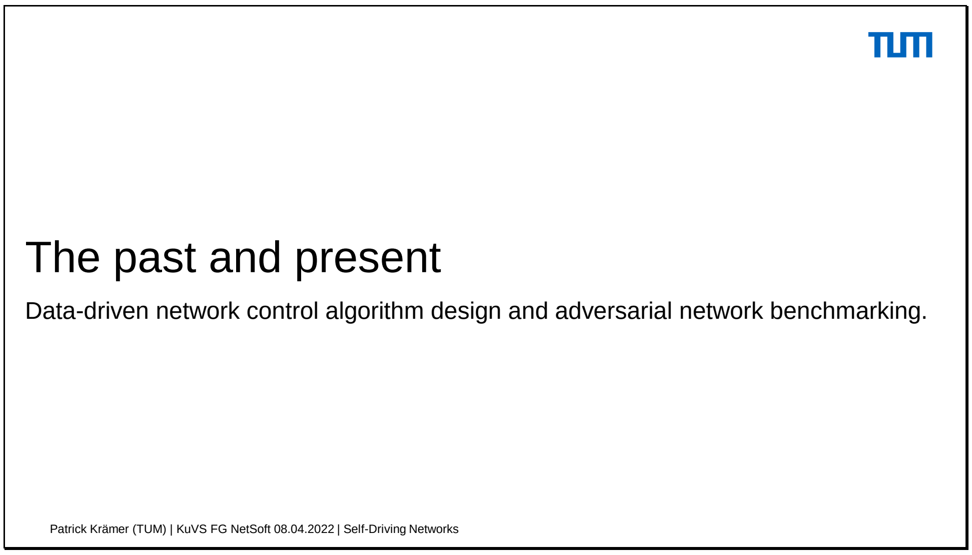
Slide 6



Our vision: A self-driving network that controls itself.

Patrick Krämer (TUM) | KuVS FG NetSoft 08.04.2022 | Self-Driving Networks

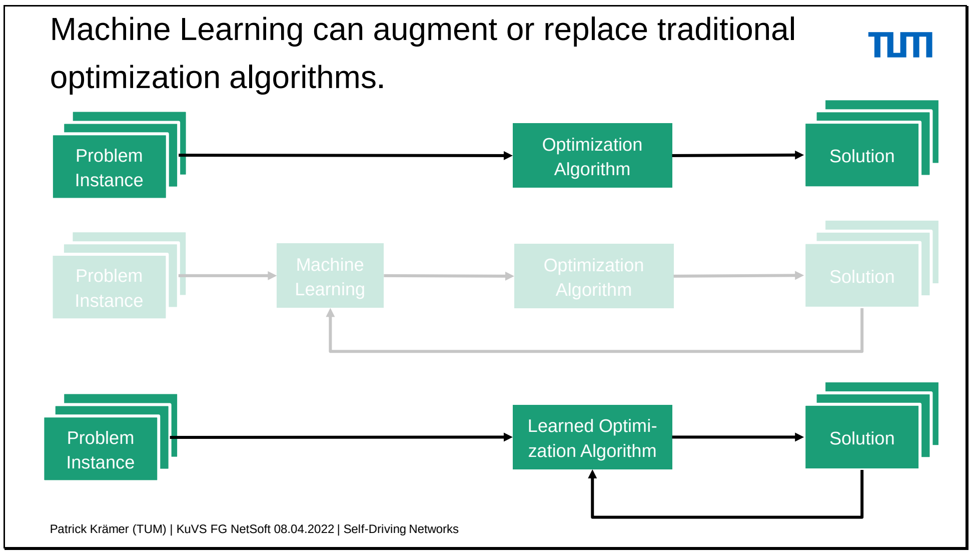
- Digital Twin of the network is the cornerstone of a self-driving network. The DT serves is interface for other components to the network.
- The network twin can:
    - Display and hold the current state of the network.
    - Exert control decisions, i.e., serves as a middle-ware that translates (abstract) control decisions to concrete network configurations/updates.
    - Perform what-if analysis.
- The Control Algorithm College uses the DT and ist what-if-analysis capabilities to train Control Algorithms.
- The Control Algorithms use the DT as a data source and use the DT to exert control decisions based on the returned information.
- A continuous monitoring allows the DT to gather data, and effects of certain control decisions.
- With the gathered data, the DT can update ist own capabilities, i.e., improving the quality of what-if-analysis, which in turn allows the Control Algorithm College to produce better Control Algorithms.



The past and present

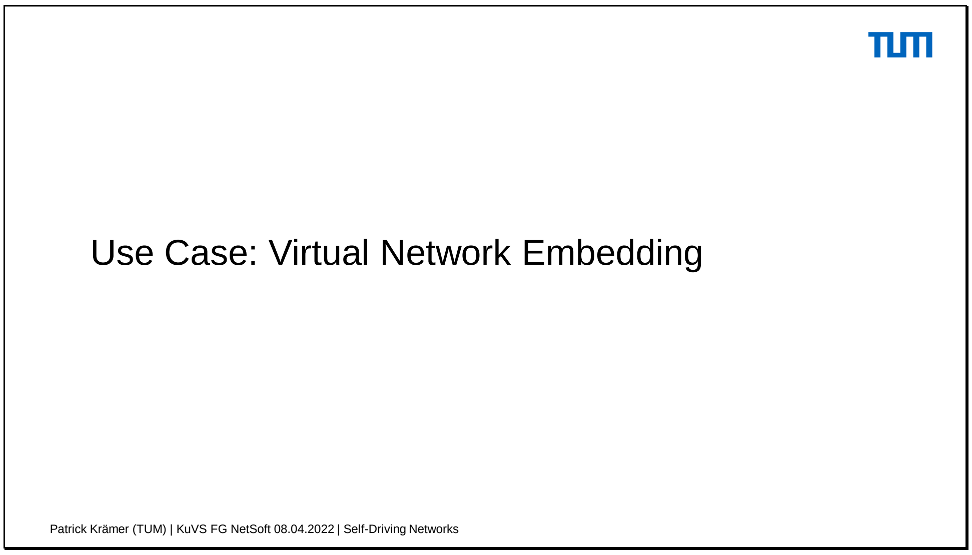
Data-driven network control algorithm design and adversarial network benchmarking.

Slide 8



Machine Learning can augment or replace traditional optimization algorithms.

Problem Instance
Optimization Algorithm
Solution
Problem Instance
Machine Learning
Optimization Algorithm
Solution
Problem Instance
Learned Optimi-zation Algorithm
Solution

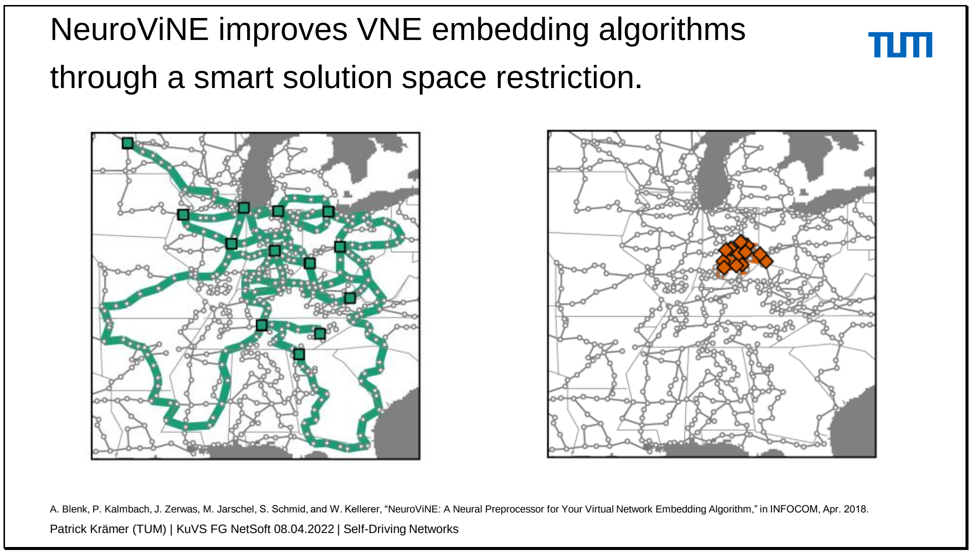


Use Case: Virtual Network Embedding

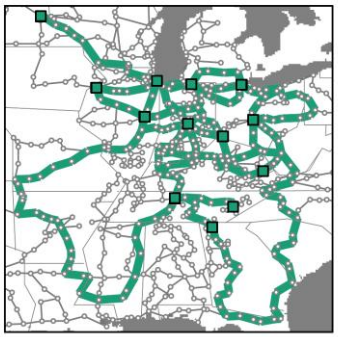
- Physical substrate network.
- Virtual Networks should be embedded onto this physical susbstrate network.
- Problem is a hard combinatorial optimization problem and the community developed a large number of algorithms.

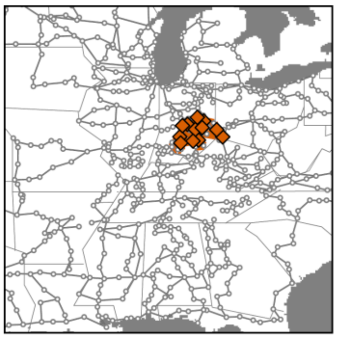
- We improve an existing algorithm through reducing the solution space using Hopfield Neural Networks.
- The solution space reduction eliminates bad local optima which results in better performance.
- Applied this principle on multiple use-cases.

Slide 10



NeuroViNE improves VNE embedding algorithms through a smart solution space restriction.

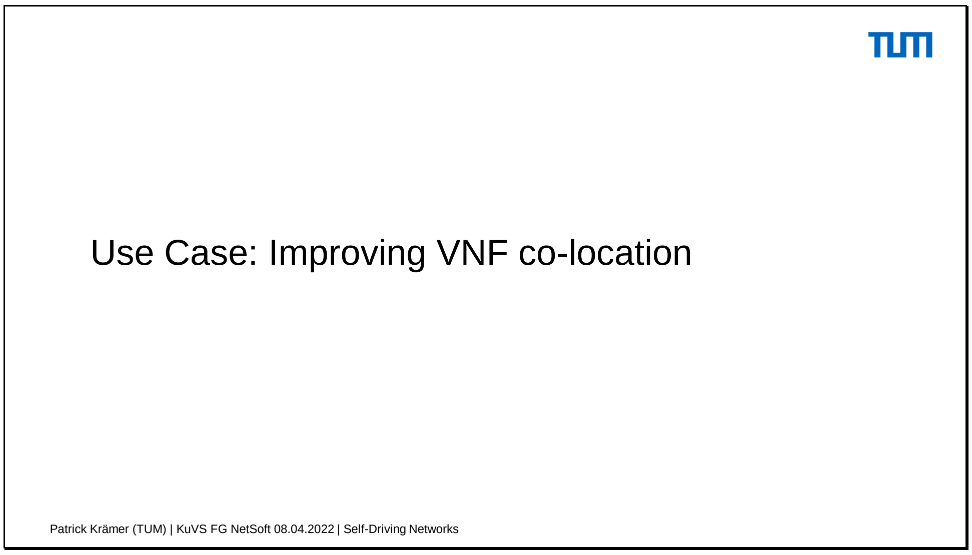




A. Blenk, P. Kalmbach, J. Zerwas, M. Jarschel, S. Schmid, and W. Kellerer, "NeuroViNE: A Neural Preprocessor for Your Virtual Network Embedding Algorithm," in INFOCOM, Apr. 2018.

- Left image shows the embedding from the optimization algorithm.
- Right shows the embedding after the solution space reduction based on a Hopfield neural network.
- Nodes are closer together, Paths are shorter → more resources are left.

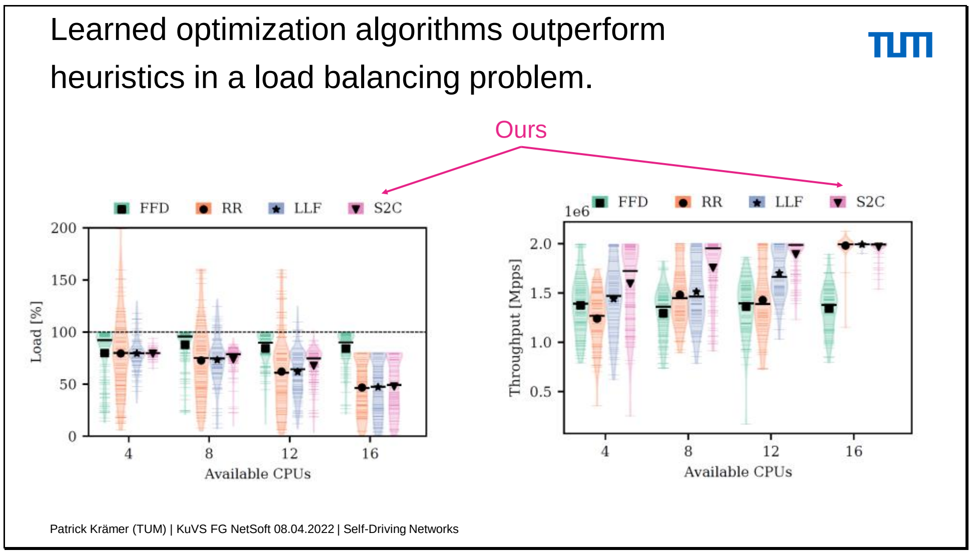
Slide 11



Use Case: Improving VNF co-location

- NF platform with cheap VNFs that do not fully utilize CPU cores (e.g., due to micro service implementation).
- VNFs are co-located on CPU cores.
- Goal: Assign VNFs to CPU cores in such a way, that CPU cores are not overloaded.

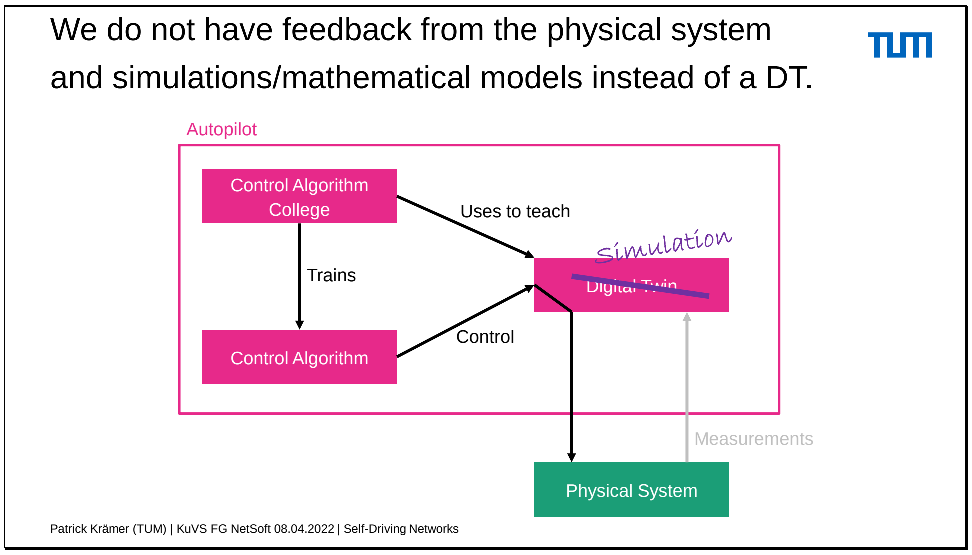
Slide 12



Learned optimization algorithms outperform heuristics in a load balancing problem.

- S2C does not overload CPU cores.
- S2C achieves higher throughput.
- S2C has lower latency.
- Still: plots show room for improvements → see later slides.
- For previous work: We use simple abstractions of the physical systems that do not reflect all properties of the network.
- This introduces artifacts into the control algorithms, which can lower performance.
- For example FFD never overlaod a CPU core, still it has low throughput and high latencies.

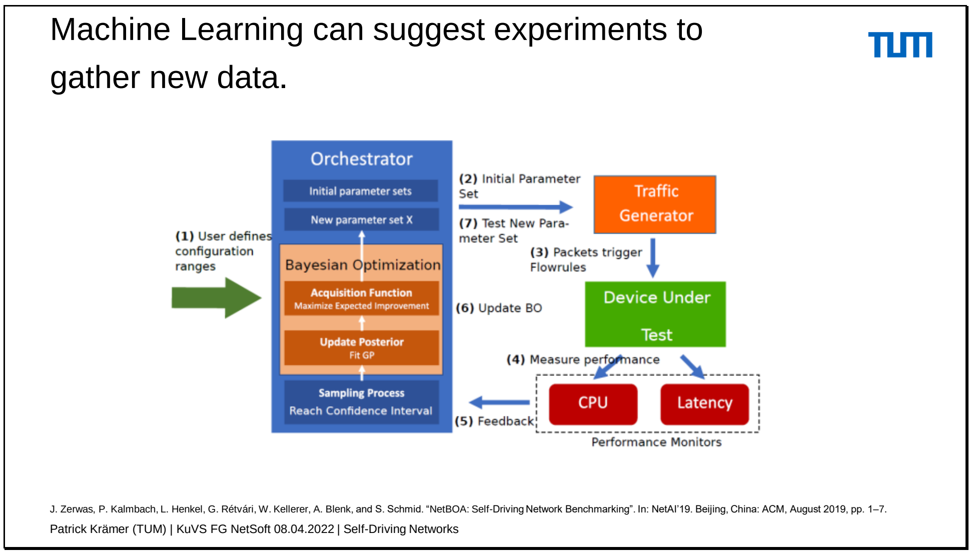
Slide 13



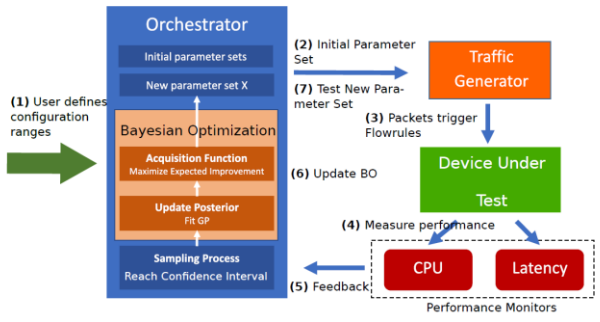
We do not have feedback from the physical system and simulations/mathematical models instead of a DT.

- We lack a feedback loop.
- We have abstract and idealized models instead of a DT.
- How to get a better idea of how the physical system behaves?
- For learning/evaluating new algorithms a wide range of system configurations must be evaluated.
- Thorough understanding of the system necessary → get as many system bevahiors with as few measurements as possible

Slide 14



Machine Learning can suggest experiments to gather new data.

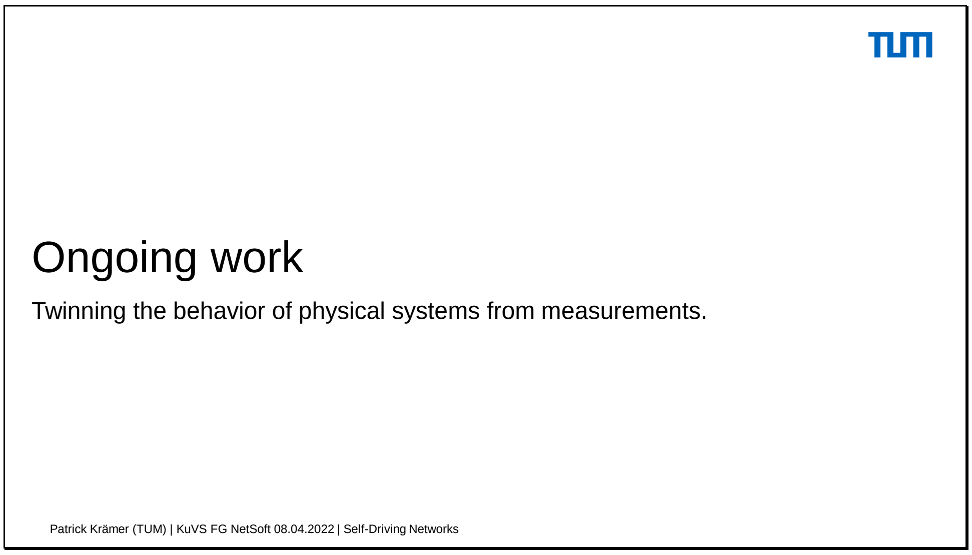


J. Zerwas, P. Kalmbach, L. Henkel, G. Rétvári, W. Kellerer, A. Blenk, and S. Schmid. “NetBOA: Self-Driving Network Benchmarking”. In: NetAI’19. Beijing, China: ACM, August 2019, pp. 1–7.

Patrick Krämer (TUM) | KuVS FG NetSoft 08.04.2022 | Self-Driving Networks

- Use ML to schedule experiments.
- User gives configuration ranges.
- ML picks parameter sets, system runs the configurations and monitors KPIs.
- KPIs flow back to ML, which chooses new configuration parameters.
- ML picks configurations for which the result is most uncertain given the previously obtained data.
- Domain Knowledge important to design the input, KPIs and the overall measurement setup.

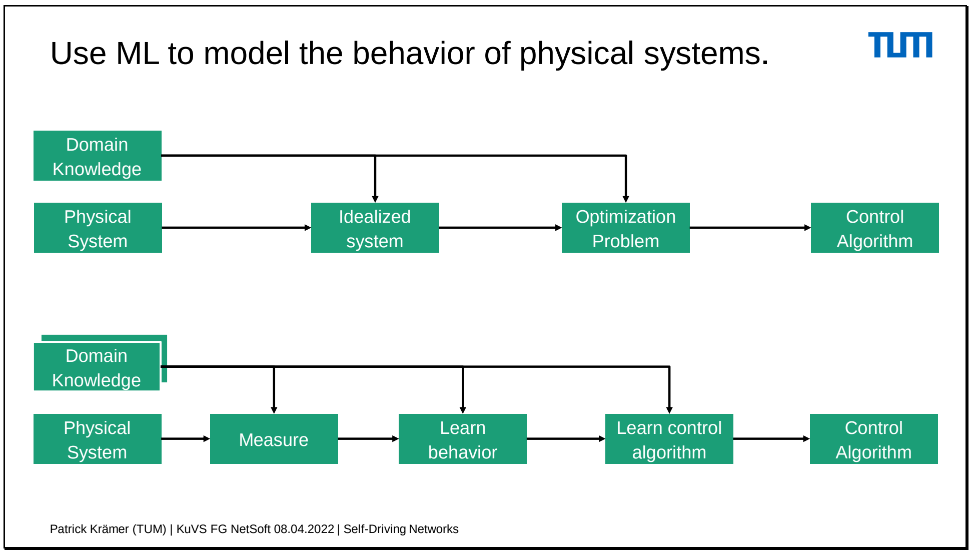
Slide 15



Ongoing work

Twinning the behavior of physical systems from measurements.

Slide 16



Use ML to model the behavior of physical systems.

$$\begin{tikzcd}
\text{Domain Knowledge} \arrow[rr] \arrow[rrrr] & & & & \\
\text{Physical System} \arrow[rr] & & \text{Idealized system} \arrow[rr] & & \text{Optimization Problem} \arrow[rr] & & \text{Control Algorithm}
\end{tikzcd}$$

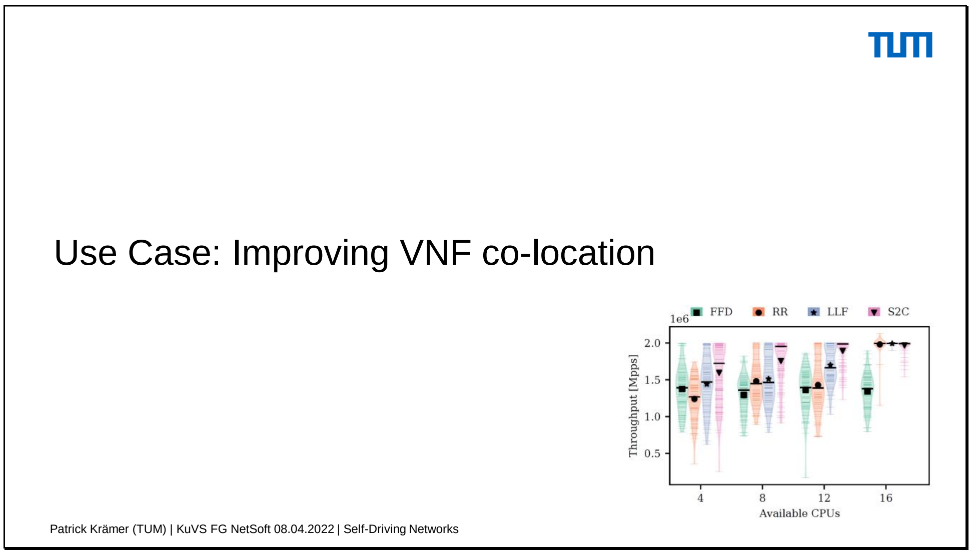
$$\begin{tikzcd}
\text{Domain Knowledge} \arrow[r] \arrow[rr] \arrow[rrr] & & & \\
\text{Physical System} \arrow[r] & \text{Measure} \arrow[r] & \text{Learn behavior} \arrow[r] & \text{Learn control algorithm} \arrow[r] & \text{Control Algorithm}
\end{tikzcd}$$

- So far: Idealized System behavior is defined from domain knowledge and the physical system.
    - Behaviour of concurrent flows on a link.
    - Behavior of VNFs on a system
- For example: Two flows take up some bandwidth on a link, without regard to burstiness or transport protocol behavior.
- Based on the domain knowledge define an optimization problem to solve, maybe with problem specific properties. Derive a control algorithm from the optimization problem, potentially translating the solution to the optimization problem to an actually deployable system, e.g., the algorithm has to run in the data-plane.

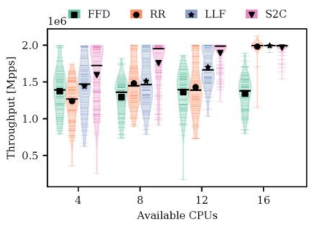
- Instead, measure the physical system.
- Learn the behavior of the system, most likely individual components, i.e., a bottom up approach.
- Requires even more domain knowledge, i.e., from the ML side, as well as the networking side.
- Use the learned behavior to learn control algorithms.
- Potentially, control algorithms must again be translated into something that fits certain requirements.
- Overall process a black-to-white-to-gray box approach:
    - Initial blackbox must be understood thoroughly, i.e., become a white box. This is important to design ML models, plan which data to acquire, and how to do that.
    - The resulting models can then again be treated as a black box, i.e., input is mapped to output, no need to know what is in-between.

- Goal is to determine important aspects once, then derive a procedure how to obtain necessary data and learn models. This process can be automated and brought to individual networks.

Slide 17

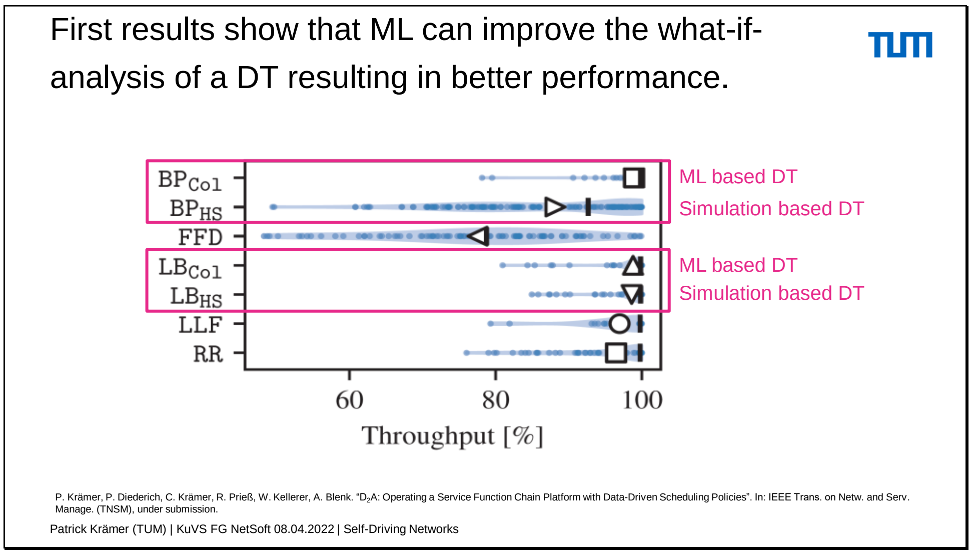


Use Case: Improving VNF co-location

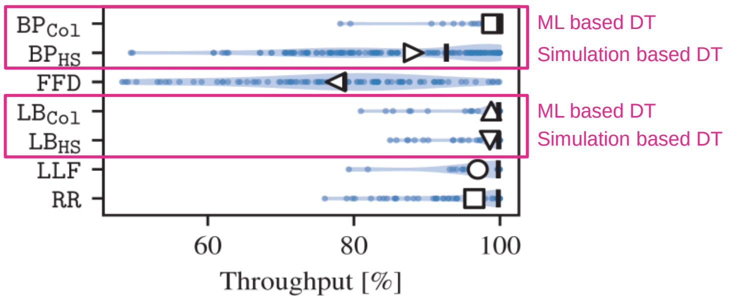


- Throughput often is very low.
- Used model does not include interference of VNFs on the system.
- From measurement data obtained during the experiments, learn those interferences.
- Include the interferences in the learning process.

Slide 18



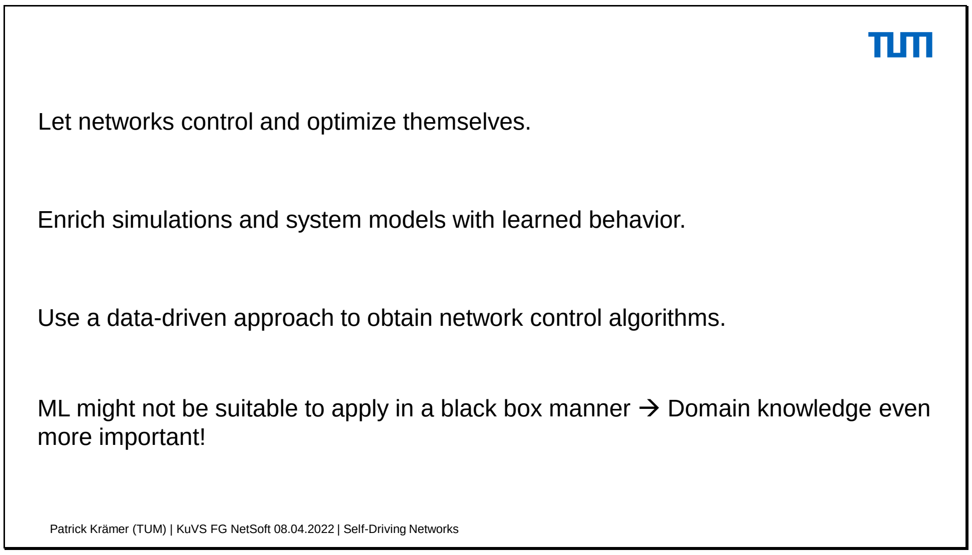
First results show that ML can improve the what-if-analysis of a DT resulting in better performance.



P. Krämer, P. Diederich, C. Krämer, R. Prieß, W. Kellerer, A. Blenk. "D$_2$A: Operating a Service Function Chain Platform with Data-Driven Scheduling Policies". In: IEEE Trans. on Netw. and Serv. Manage. (TNSM), under submission.

- HS corresponds to simulation based control algorithms.
- Col corresponds to control algorithms that are obtained using the learned interferenecs.
- BP corresponds to Bin Packing.
- LB corresponds to Load Balancing
- Throughput drastically improved, learned Bin Packing algorithm better than Load Balancing heuristics, although BP algo uses fewer cores.

Slide 19



Let networks control and optimize themselves.

Enrich simulations and system models with learned behavior.

Use a data-driven approach to obtain network control algorithms.

ML might not be suitable to apply in a black box manner → Domain knowledge even more important!