NN4SysBench: Characterizing Neural Network Verification for Computer Systems

Shuyi Lin, Haoyu He, Tianhao Wei, Kaidi Xu, Huan Zhang, Gagandeep Singh, Changliu Liu, Cheng Tan

Northeastern University

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

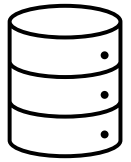
Tübingen AI Center

Carnegie Mellon University

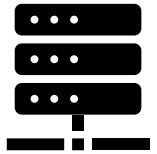
Neural Networks for computer systems (NN4Sys)

Neural Networks

Operating Systems



Databases



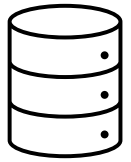
Networked Systems

For example, NN for

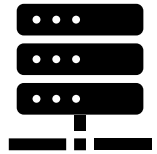
Neural Networks for computer systems (NN4Sys)

Neural Networks

Operating Systems



Databases



Networked Systems

For example, NN for

- OS scheduler => better job completion time
- database index => smaller memory footprint, faster lookup
- network routing => more efficient packet routing

However, NN4Sys…

- …, despite improving the average performance, …
- …may do much worse in the worst-case scenarios.

However, NN4Sys…

- …, despite improving the average performance, …
- …may do much worse in the worst-case scenarios.

For example, NN for

- OS scheduler => job starvation
- database index => pointing to wrong positions
- network routing => blackholes (loops in the chosen path)

The problem is fundamental

- NNs are complicated **black boxes**.
  - Hard to understand what a NN has learned.

The problem is fundamental

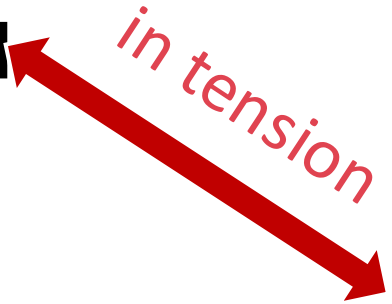
- NNs are complicated **black boxes**.
  - Hard to understand what a NN has learned.

- NN's behaviors are **not well-defined.**
  - NNs may produce unexpected results.

The problem is fundamental

- NNs are complicated **black boxes**.
  - Hard to understand what a NN has learned.

- NN's behaviors are **not well-defined**
  - NNs may produce unexpected results.



- Meanwhile, computer systems require **safety properties**.

NN-Verification to the rescue

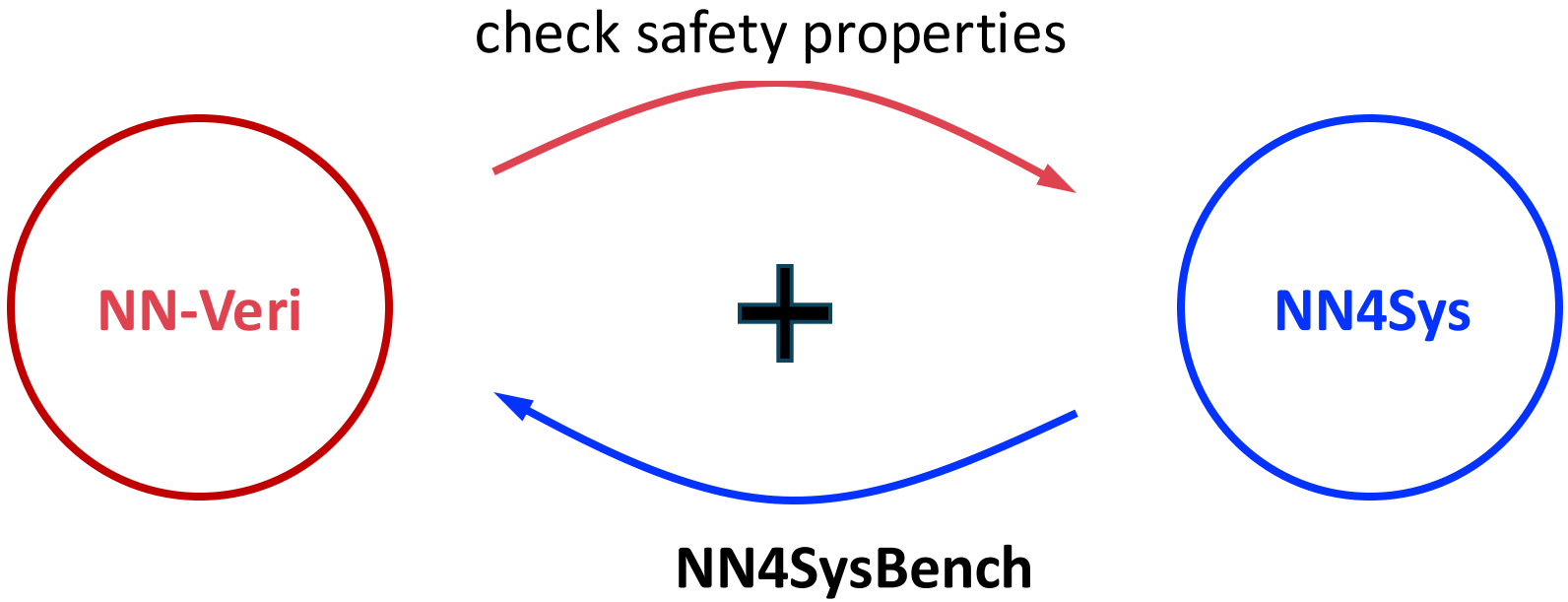
NN-Verification can provide a “lower bound” for NN4Sys.

NN-Verification to the rescue

NN-Verification can provide a “lower bound” for NN4Sys.

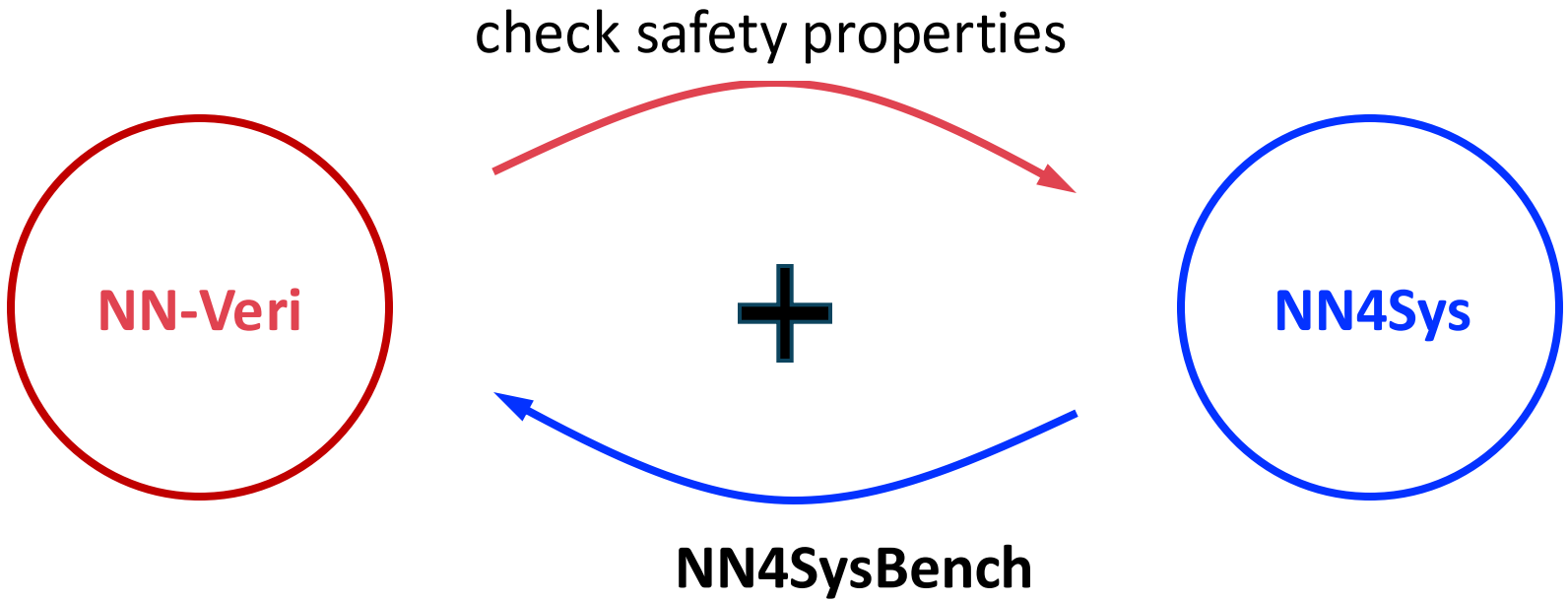
- Crucially, NN-Verification works well for NN4Sys because
  - NN4Sys has simple models;
  - NN4Sys has clear semantics.

Our vision: NN4Sys + NN-Verification



- We argue: NN4Sys + NN-verification whenever possible

Our vision: NN4Sys + NN-Verification



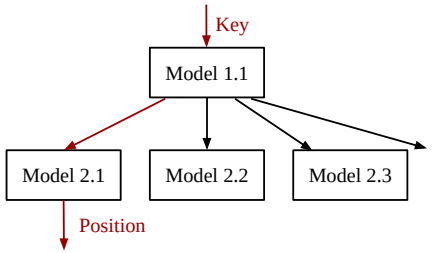
- We argue: NN4Sys + NN-verification whenever possible
- However, NN4Sys have characteristics…

  …that today's verifiers do not support well.

NN4SysBench: a benchmark suite for NN-Verification whose benchmarks are from impactful NN4Sys

NN4SysBench: a benchmark suite for NN-Verification whose benchmarks are from impactful NN4Sys

learned index

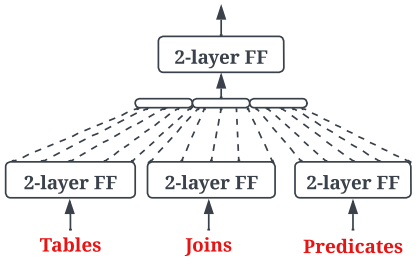


specifications

$\forall k \in [\mathcal{K}[i], \mathcal{K}[i+1]]$,

$F(k) \in [DB(\mathcal{K}[i]) - \epsilon, DB(\mathcal{K}[i+1]) + \epsilon]$

learned cardinality estimation



specifications

1. $F(q) \leq |R|$.
2. $F(q) \geq 0$.
3. $F(q) \geq F(q \wedge \theta_i), i \in [1, d]$.

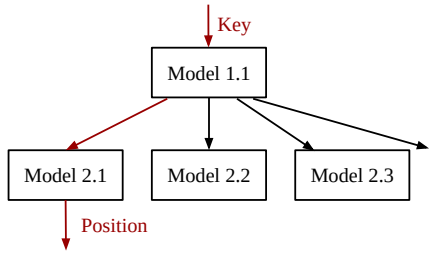
...

learned bloom filter

supervised learning

NN4SysBench: a benchmark suite for NN-Verification whose benchmarks are from impactful NN4Sys

learned index

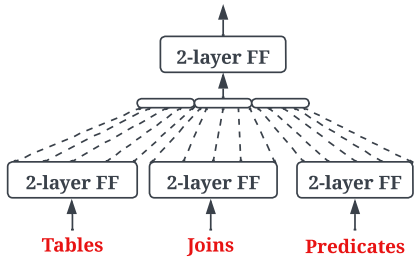


specifications

$\forall k \in [\mathcal{K}[i], \mathcal{K}[i+1]]$,

$F(k) \in [DB(\mathcal{K}[i])-\epsilon, DB(\mathcal{K}[i+1])+\epsilon]$

learned cardinality estimation



specifications

1. $F(q) \leq |R|$.
2. $F(q) \geq 0$.
3. $F(q) \geq F(q \wedge \theta_i), i \in [1, d]$.

...

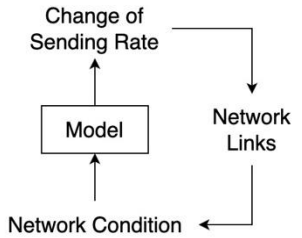
supervised learning

learned bloom filter

learned adaptive bitrate

learned distributed system scheduler

learned Internet congestion control



specifications

- Basic properties: in good (or bad) network condition, model does not decrease(increase) packet sending rates.

...

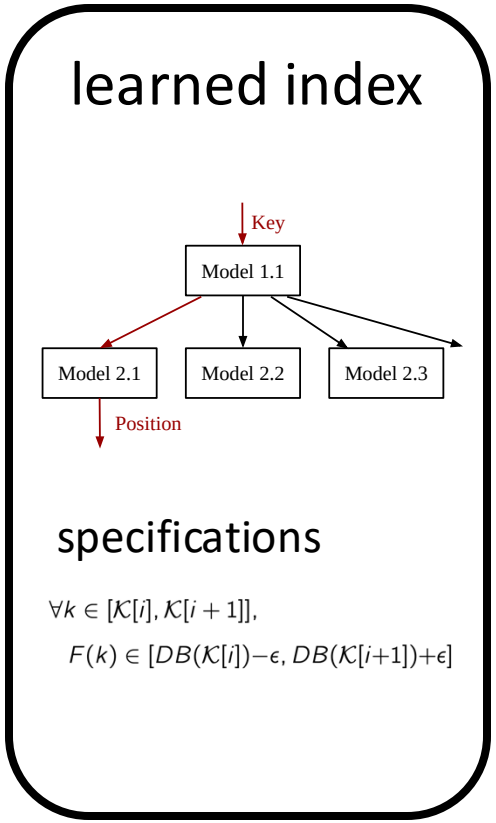
reinforcement learning

An (incomplete) list of NN4Sys characteristics

1. small number of input dimensions
2. large number of specification entries
3. hierarchical models
4. temporal specification
5. monotonicity specification

An (incomplete) list of NN4Sys characteristics

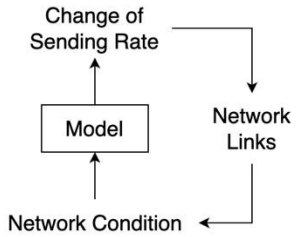
1. small number of input dimensions
2. large number of specification entries
3. hierarchical models
4. temporal specification
5. monotonicity specification



An (incomplete) list of NN4Sys characteristics

1. small number of input dimensions
2. large number of specification entries
3. hierarchical models
4. temporal specification
5. monotonicity specification

learned Internet congestion control

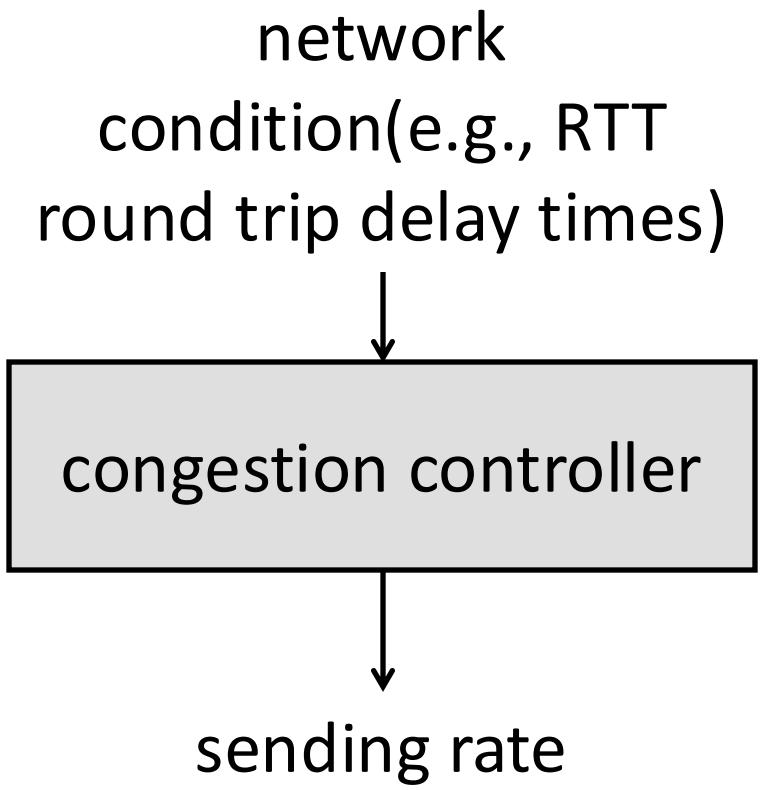


specifications

• Basic properties: in good (or bad) network condition, model cannot choose the worst (or best) resolution.

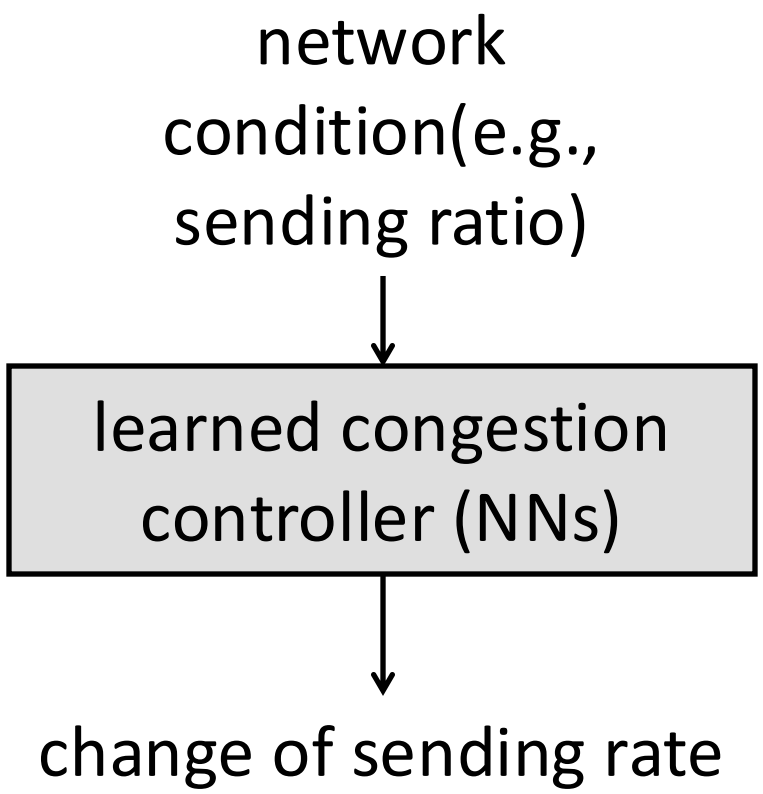
...

Traditional congestion control



- Classic congestion controller: Cubic, BBR

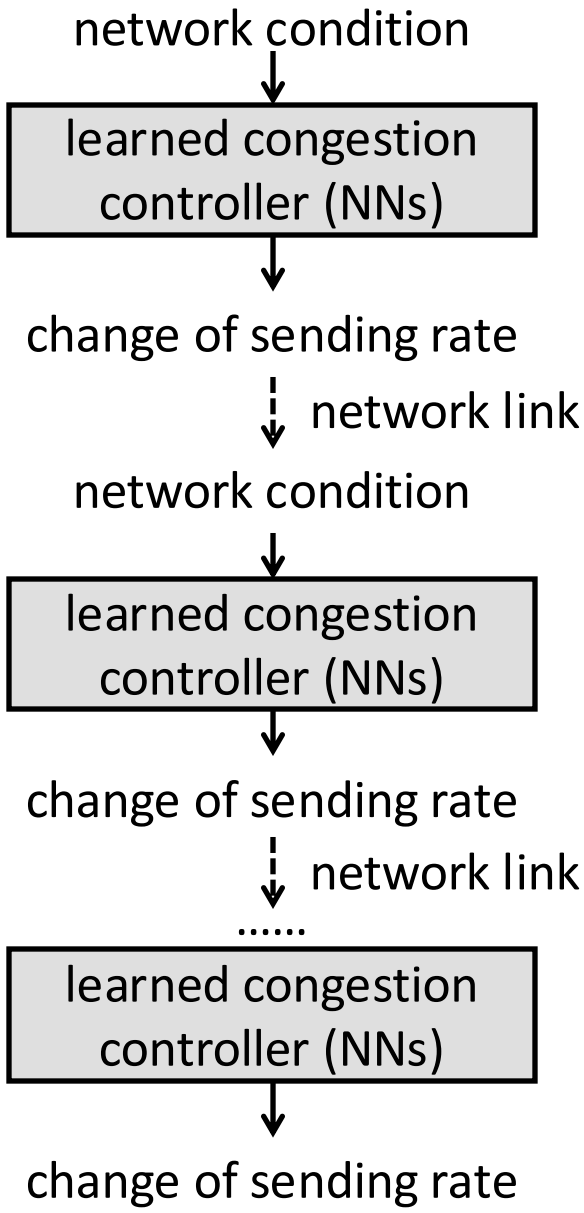
Learned congestion control



- Classic congestion controller: Cubic, BBR
- Learned congestion control: use NNs to replace the congestion control algorithm

Nathan Jay, Noga Rotman, Brighten Godfrey, Michael Schapira, and Aviv Tamar. A deep reinforcement learning perspective on internet congestion control. In International Conference on Machine Learning. PMLR, 2019

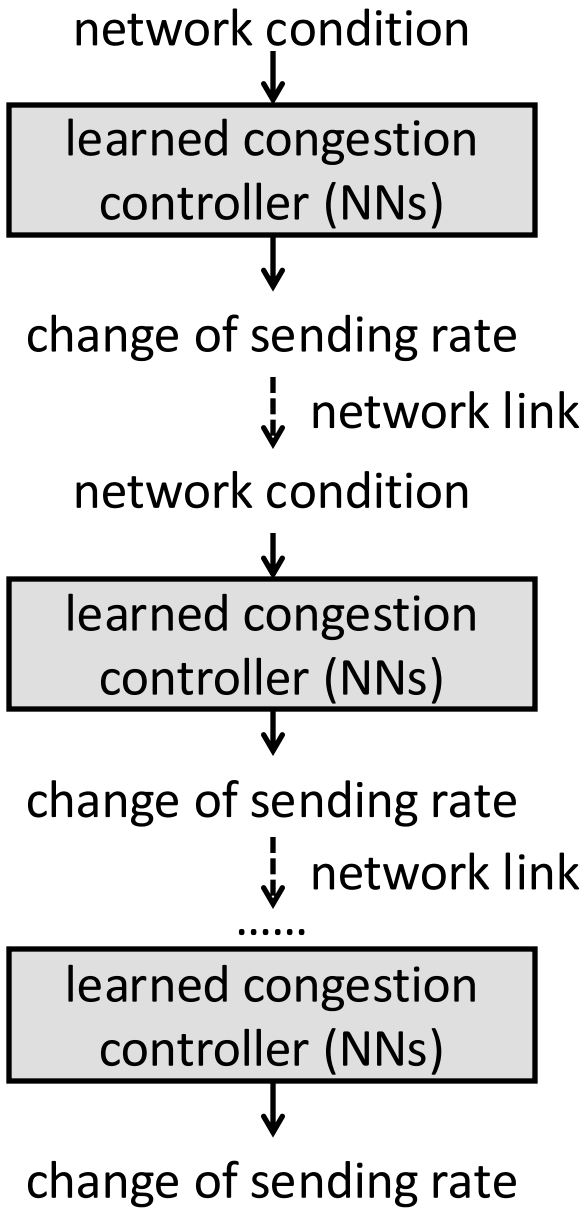
Specifications of learned congestion control



- When the network condition changes from bad to good, the sender eventually increases packet sending rates.

Nathan Jay, Noga Rotman, Brighten Godfrey, Michael Schapira, and Aviv Tamar. A deep reinforcement learning perspective on internet congestion control. In International Conference on Machine Learning. PMLR, 2019

Specifications of learned congestion control



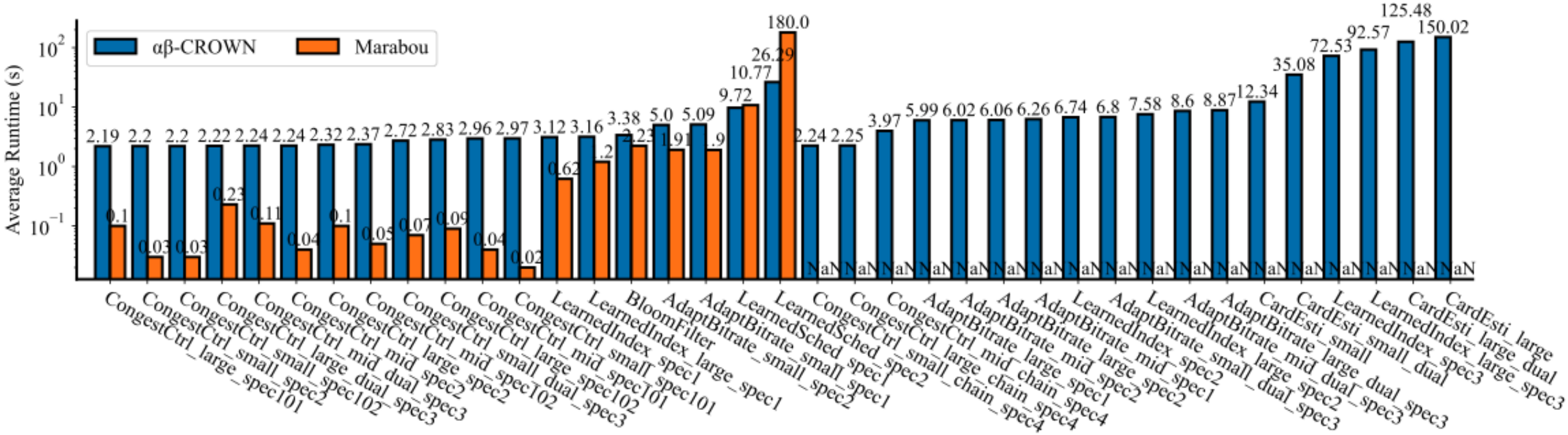
multiple steps

- When the network condition changes from bad to good, the sender eventually increases packet sending rates.

characteristic 4: temporal specification

Nathan Jay, Noga Rotman, Brighten Godfrey, Michael Schapira, and Aviv Tamar. A deep reinforcement learning perspective on internet congestion control. In International Conference on Machine Learning. PMLR, 2019

Evaluation on 2 verifiers: αβ-crown and marabou



Verification runtime for different benchmarks

NN4SysBench: Characterizing Neural Network Verification for Computer Systems

Contribution: Previous version adopted by the Verification of Neural Networks Competition (VNN-Comp) for 3 consecutive years

Code, paper and contact info at website: https://shuyilinn.github.io/BenchmarkWeb/