CausalSim: A Causal Framework for Unbiased Trace-Driven Simulation

Arash Nasr-Esfahany*

Joint with: Abdullah Alomar*, Pouya Hamadanian*

Anish Agarwal, Mohammad Alizadeh, and Devavrat Shah

*Equal Contribution

Massachusetts
Institute of
Technology

Trace-driven simulation

✓ Use traces to capture **real** system behavior

✓ **Less complex** than full-system simulation

Biased outcomes

Trace-driven simulation

- ✓ Use traces to capture **real** system behavior
- ✓ **Less complex** than full-system simulation
- **Biased** outcomes

Promise

- ➢ Key source of bias in trace-driven simulation

Trace-driven simulation

✓ Use traces to capture **real** system behavior

✓ **Less complex** than full-system simulation

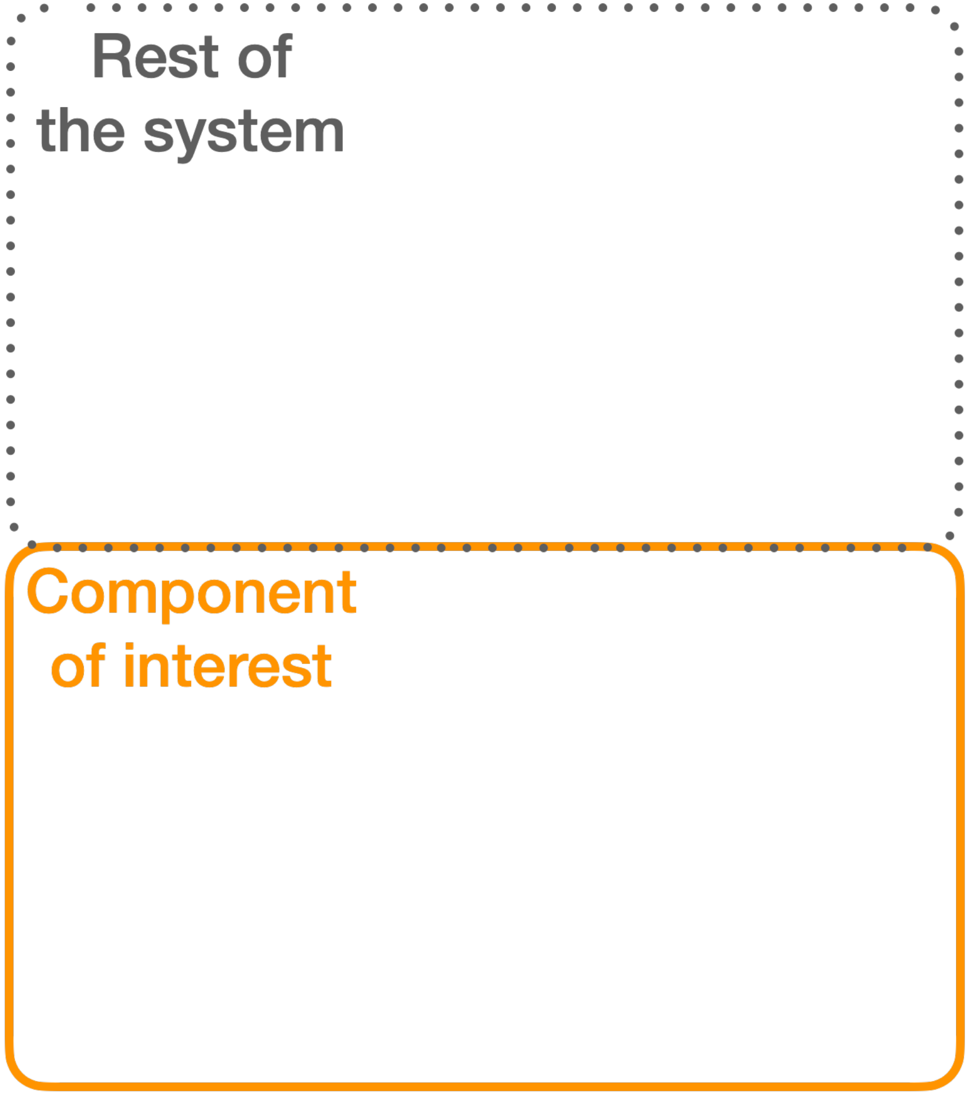
Biased outcomes

Promise

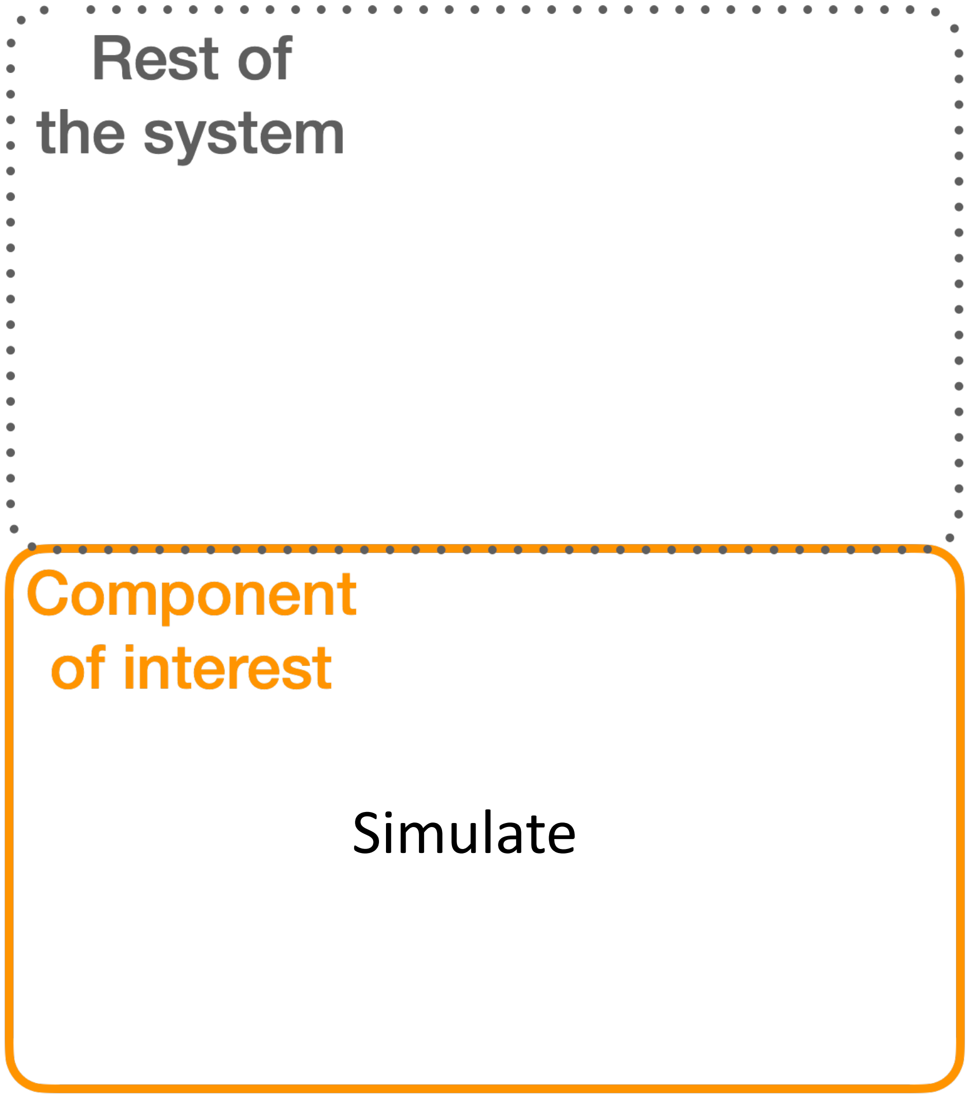
➢ Key source of bias in trace-driven simulation

➢ How to do unbiased trace-driven simulation?

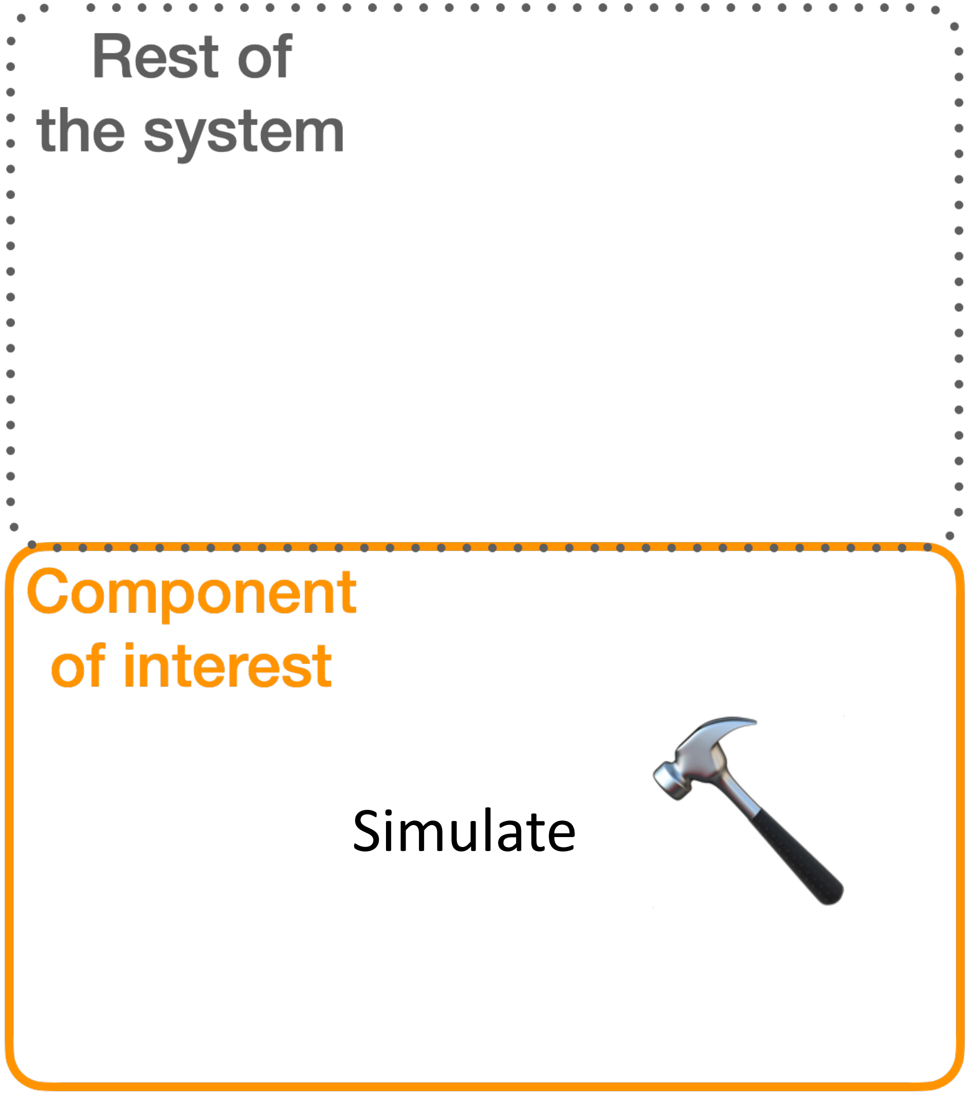
Exogenous trace assumption in trace-driven simulation



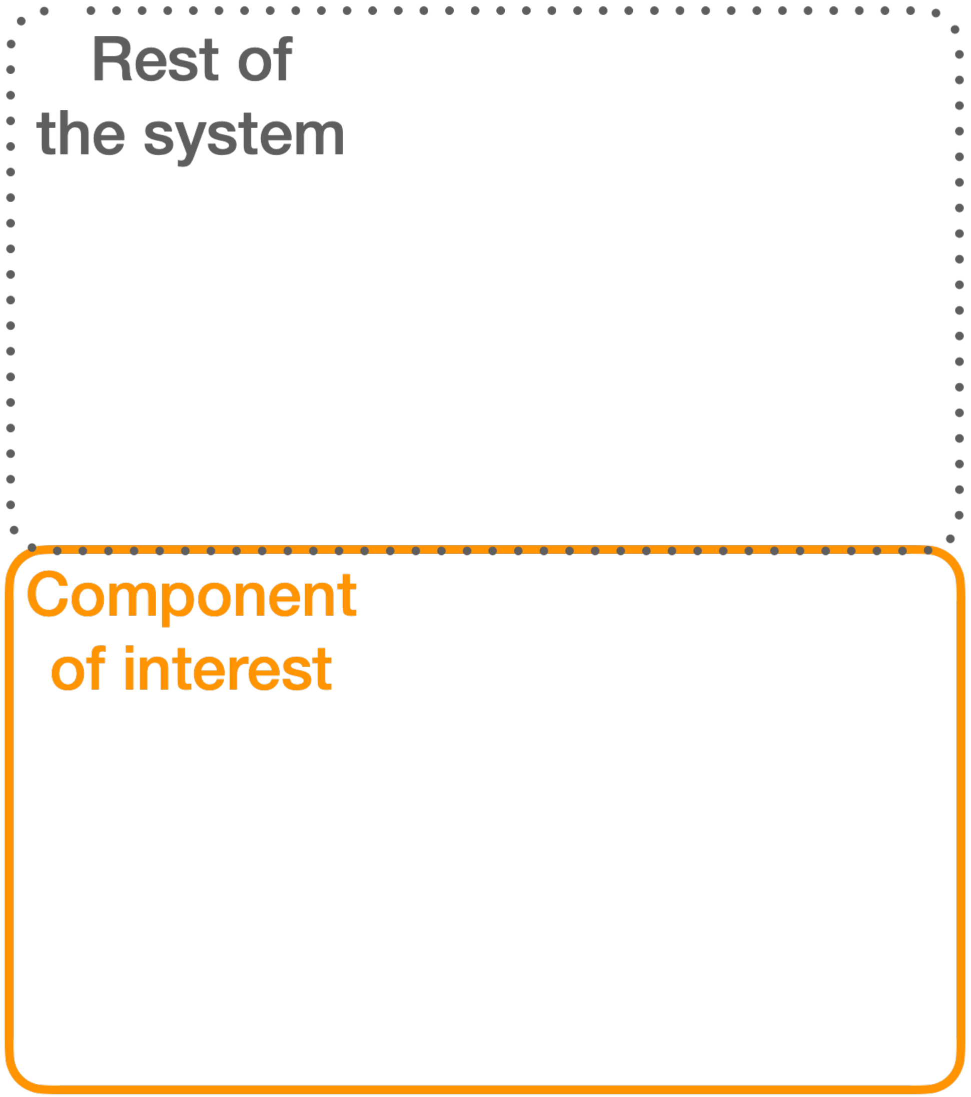
Exogenous trace assumption in trace-driven simulation



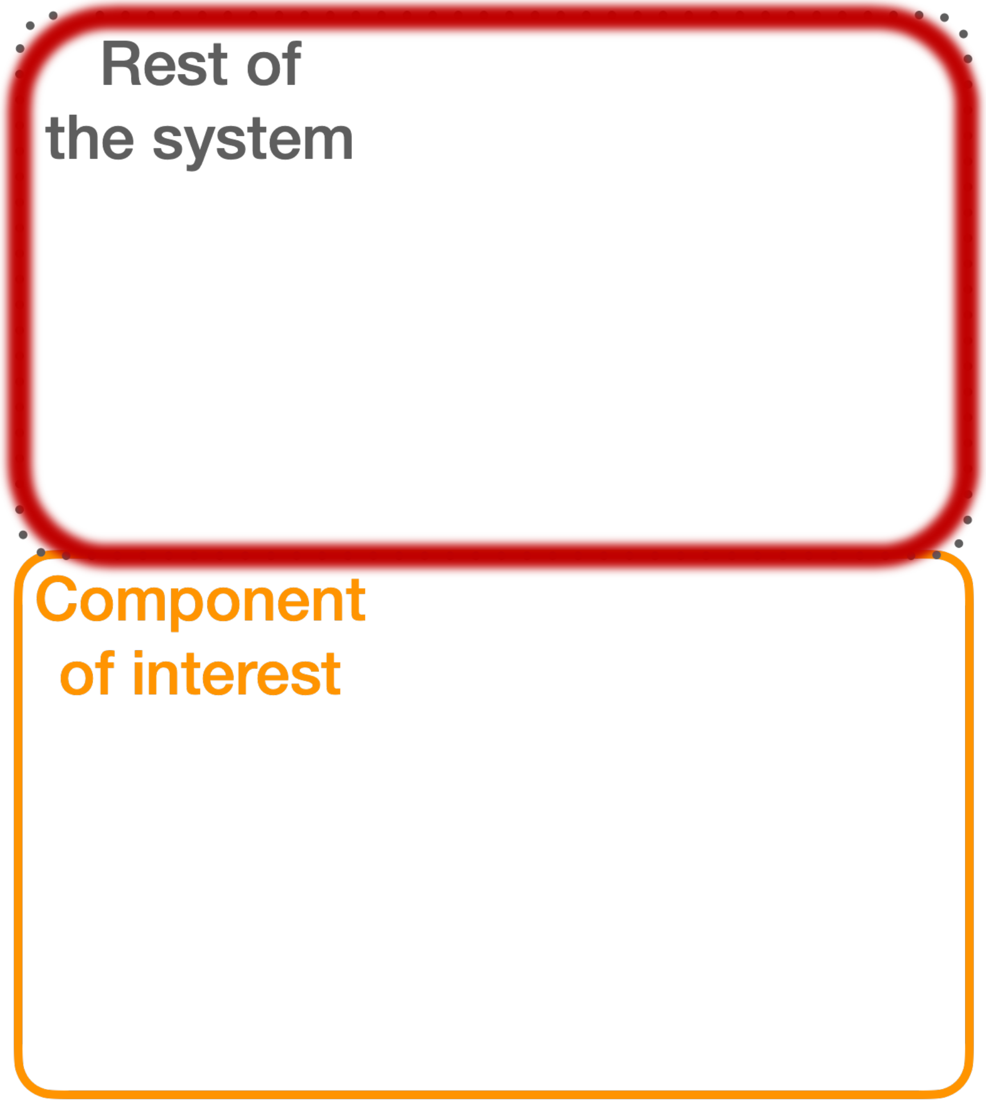
Exogenous trace assumption in trace-driven simulation



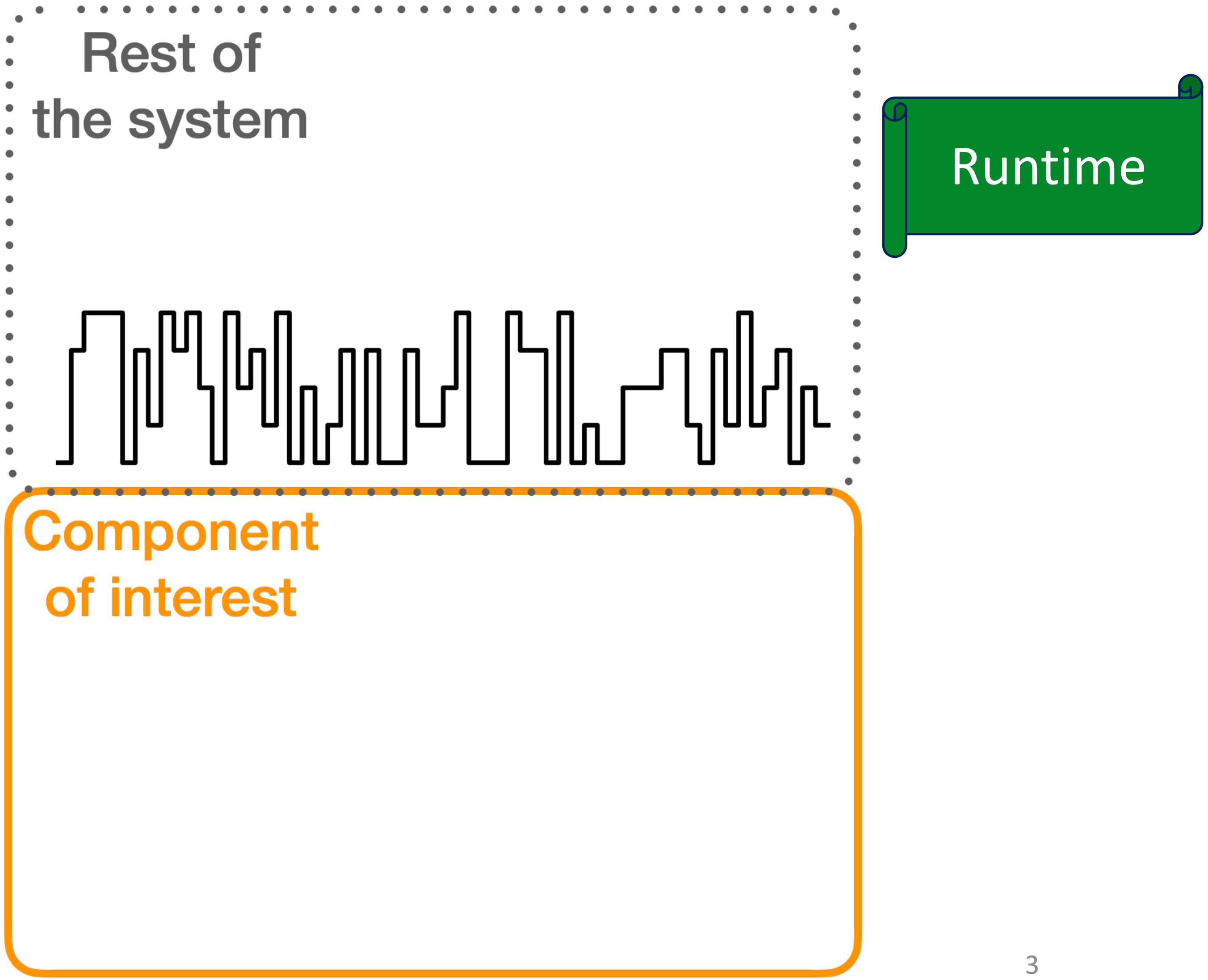
Exogenous trace assumption in trace-driven simulation



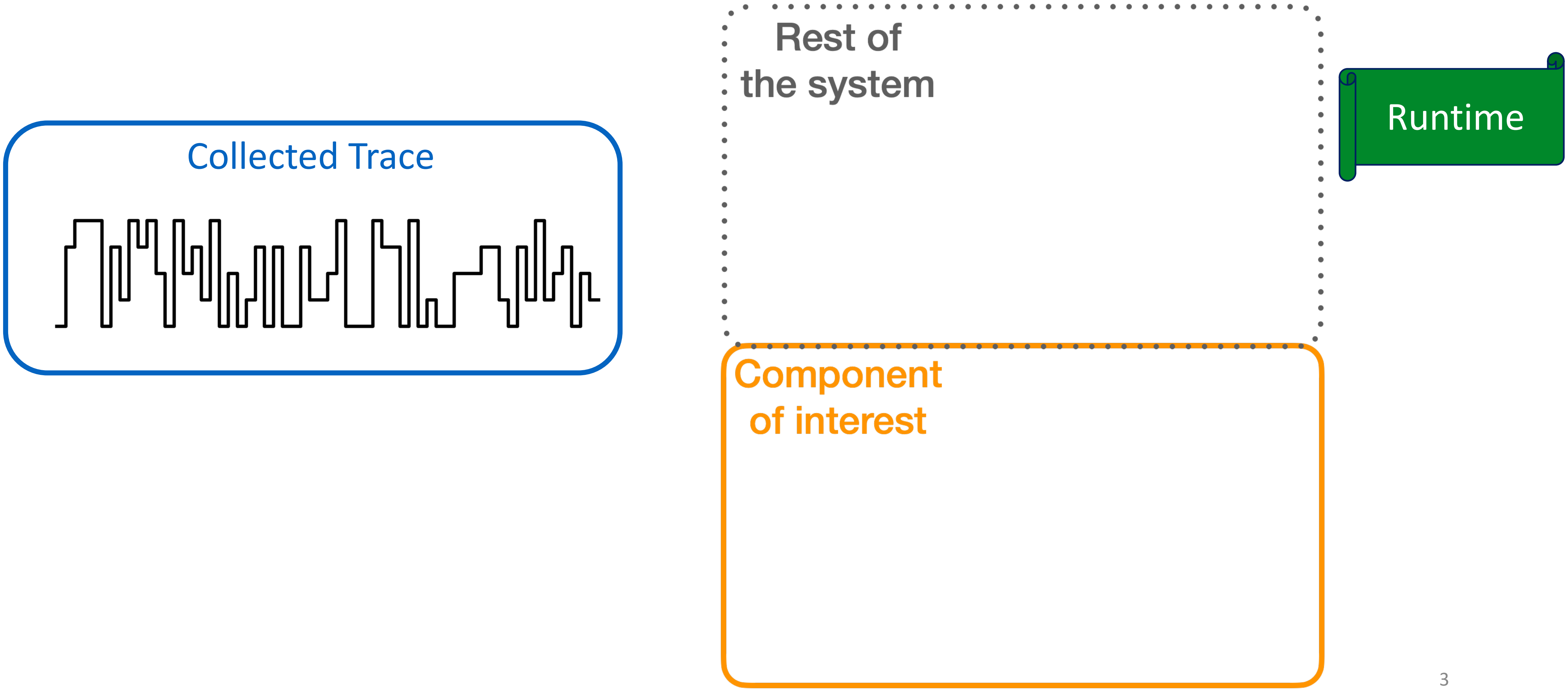
Exogenous trace assumption in trace-driven simulation



Exogenous trace assumption in trace-driven simulation



Exogenous trace assumption in trace-driven simulation



Collected Trace

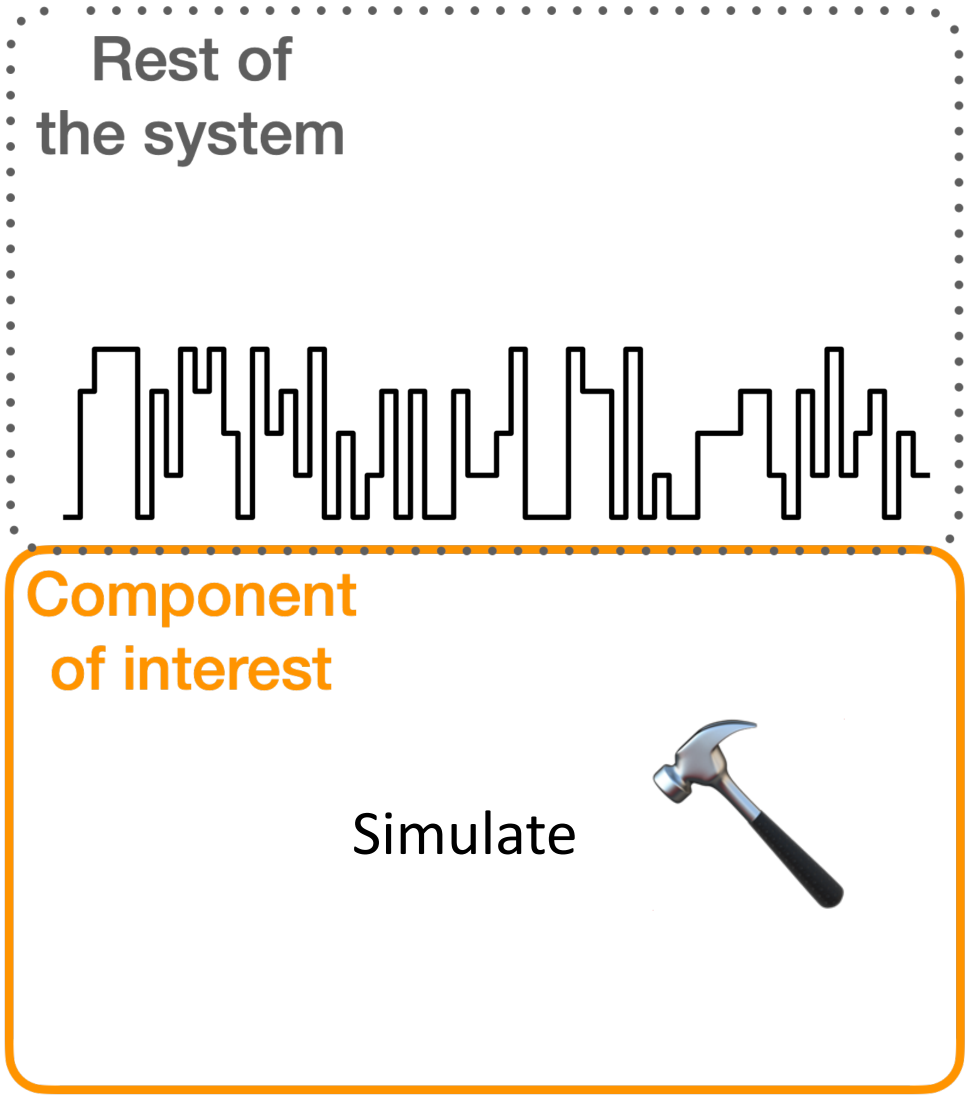
Rest of the system

Runtime

Component of interest

Exogenous trace assumption in trace-driven simulation

Collected Trace



Rest of the system

Component of interest

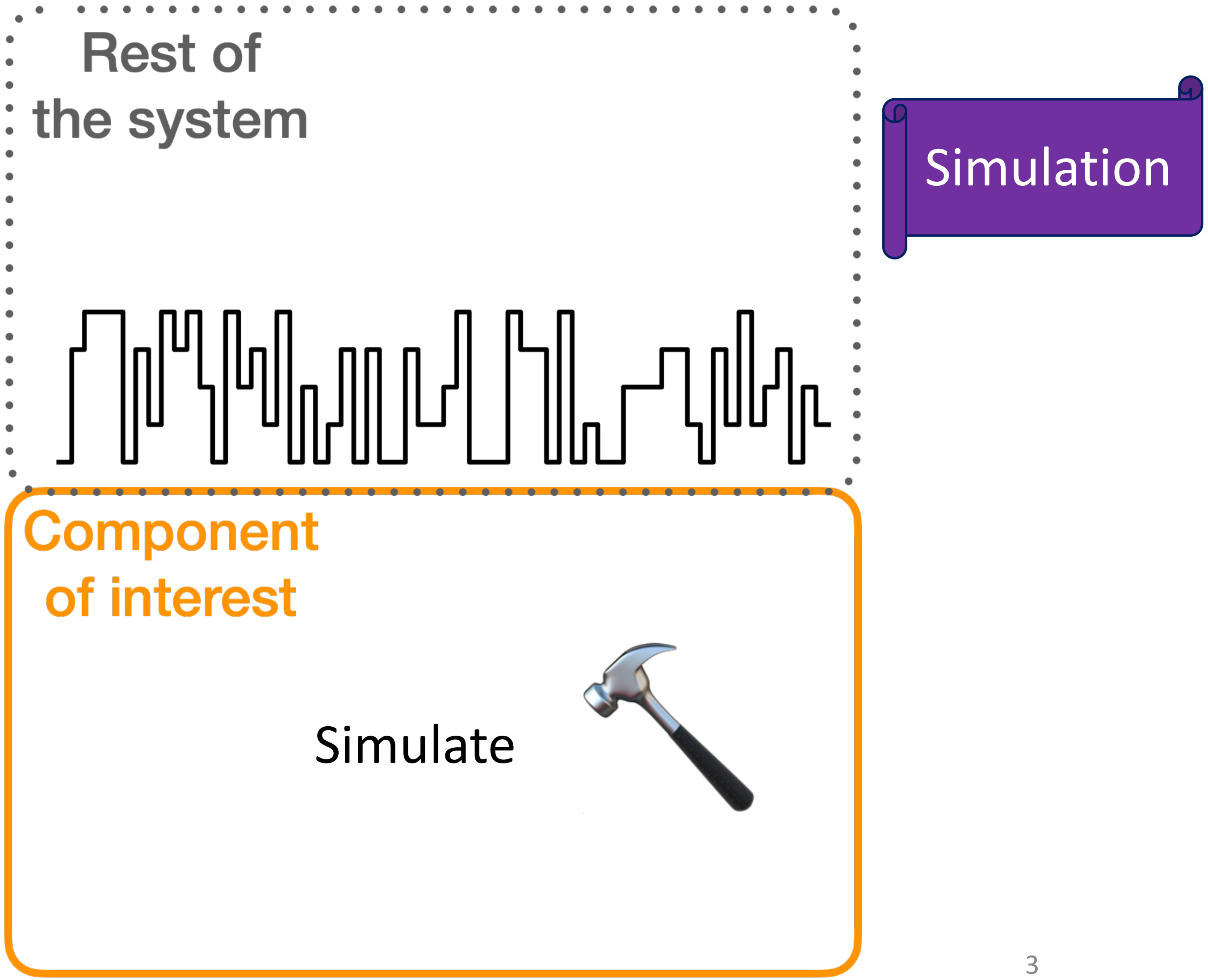
Simulate

Simulation

Exogenous trace assumption in trace-driven simulation

Collected Trace

- ***Exogenous trace*** assumption: Simulated interventions would not affect the replayed trace.



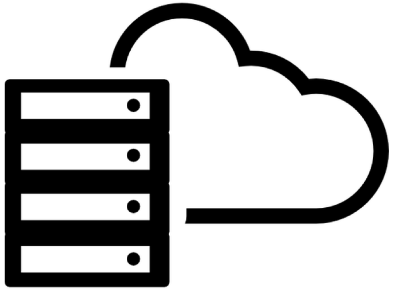
Exogenous trace assumption does not hold for many real-world traces

Exogenous trace assumption does not hold for many real-world traces

… hurts accuracy, can lead to completely wrong conclusions

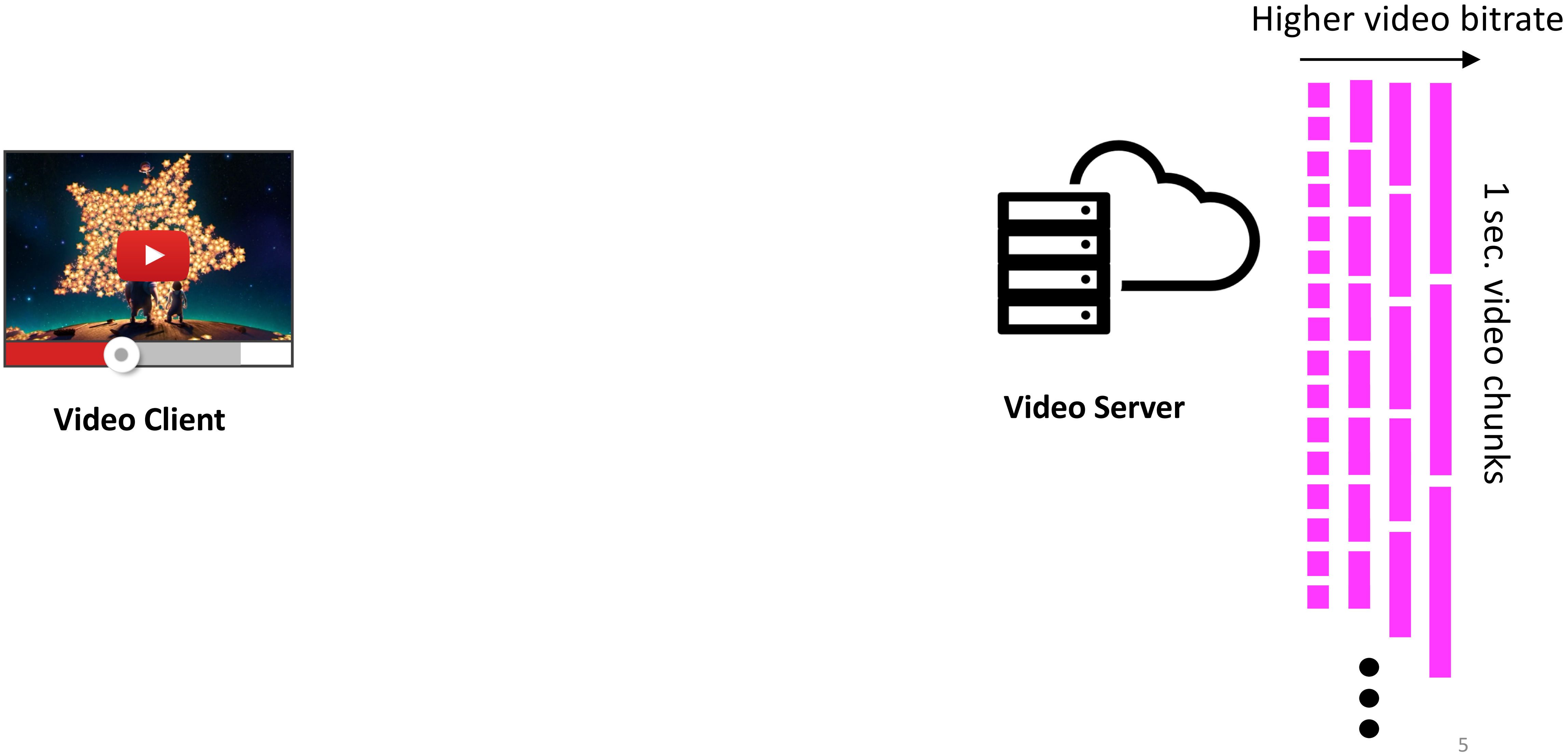
Example: Adaptive Bitrate Video Streaming (ABR)

Video Client

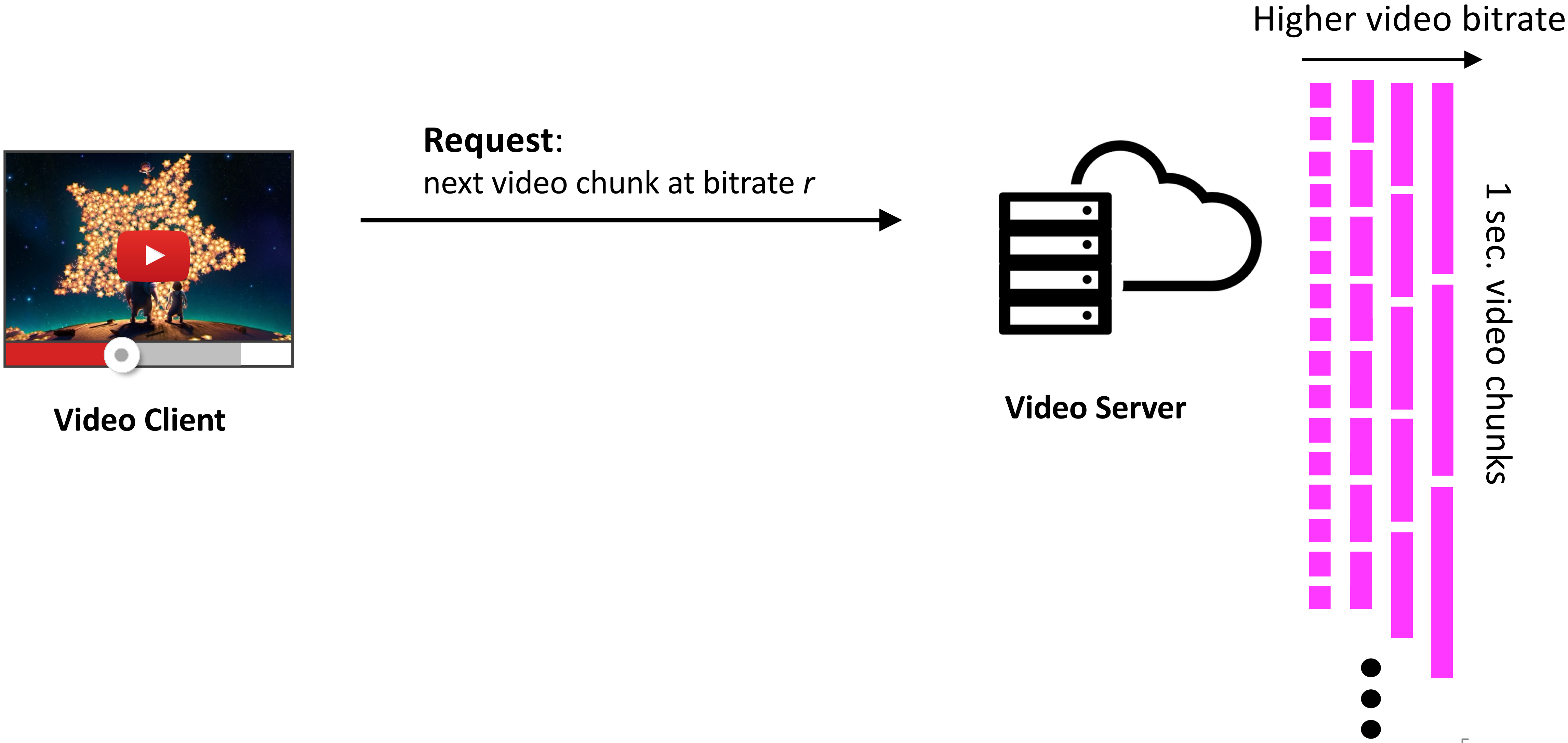


Video Server

Example: Adaptive Bitrate Video Streaming (ABR)

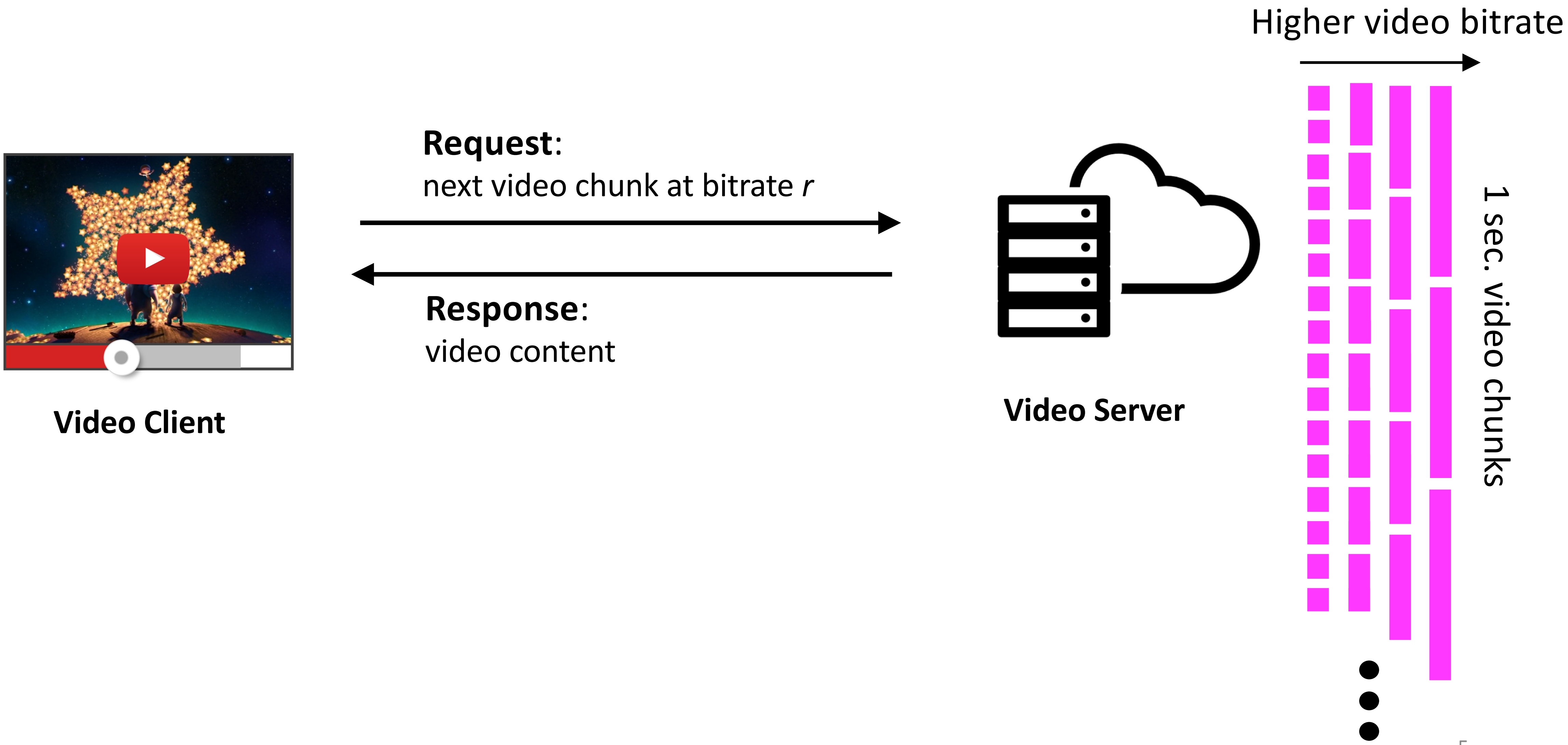


Example: Adaptive Bitrate Video Streaming (ABR)



Request:
next video chunk at bitrate *r*

Example: Adaptive Bitrate Video Streaming (ABR)



Request:
next video chunk at bitrate *r*

Response:
video content

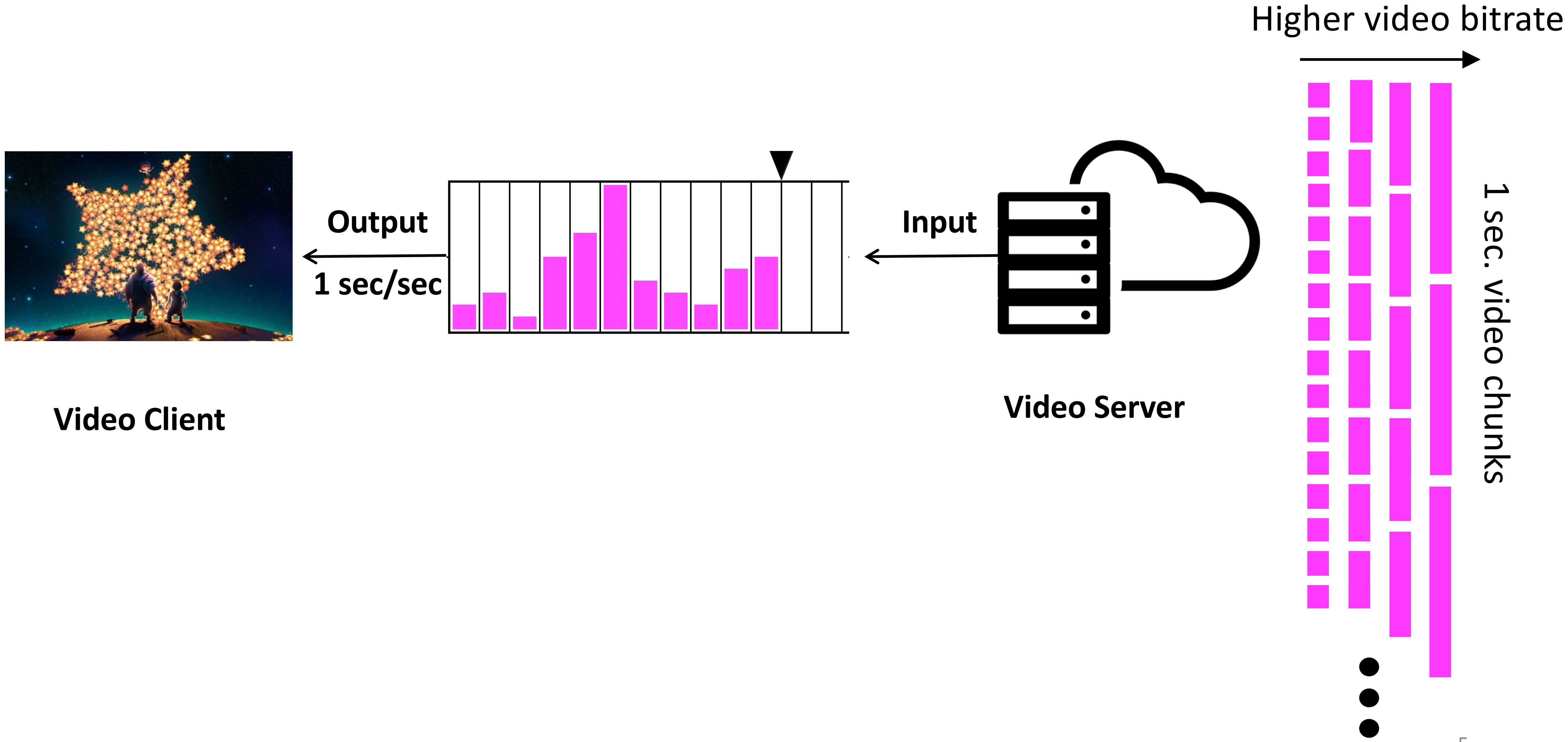
Video Client

Video Server

Higher video bitrate

1 sec. video chunks

Example: Adaptive Bitrate Video Streaming (ABR)



Higher video bitrate

Output

1 sec/sec

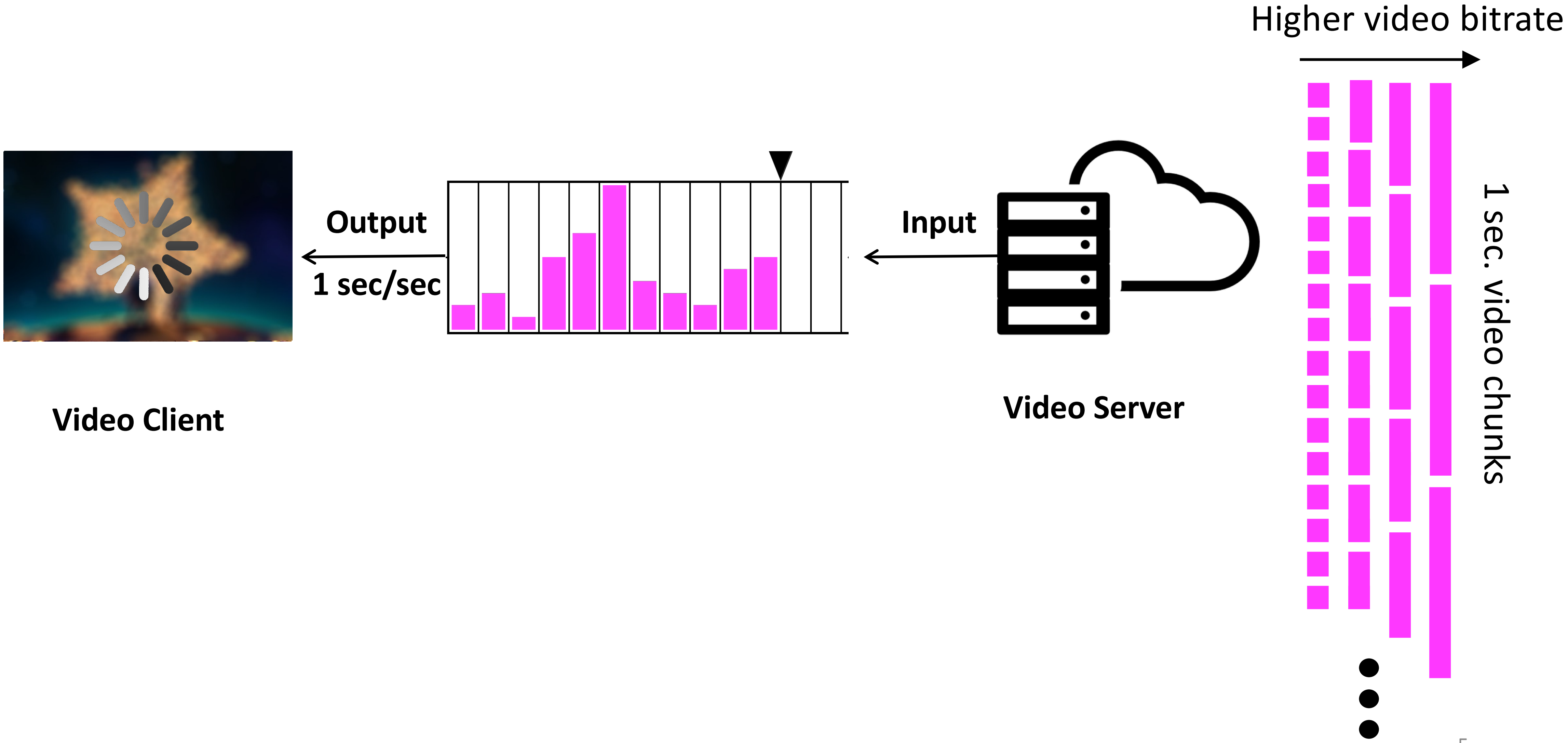
Input

1 sec. video chunks

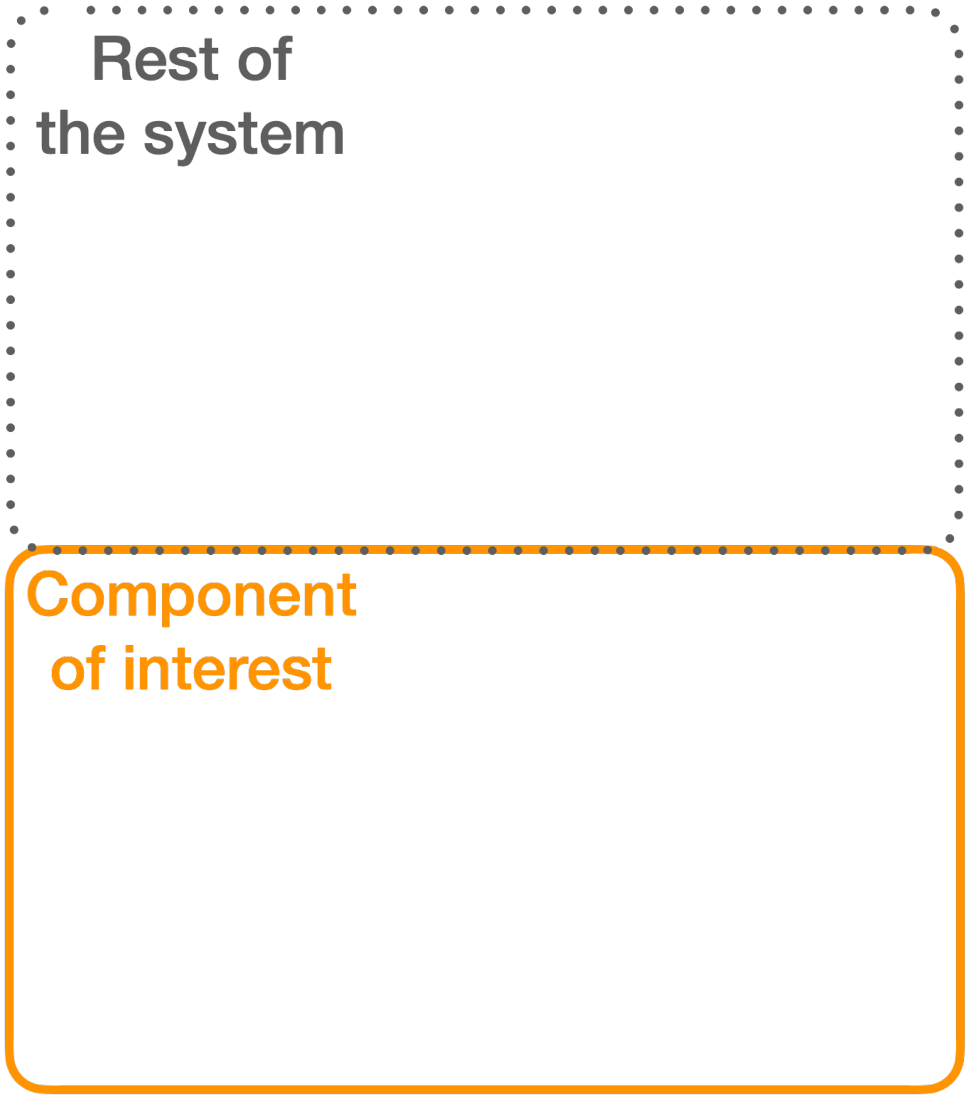
Video Client

Video Server

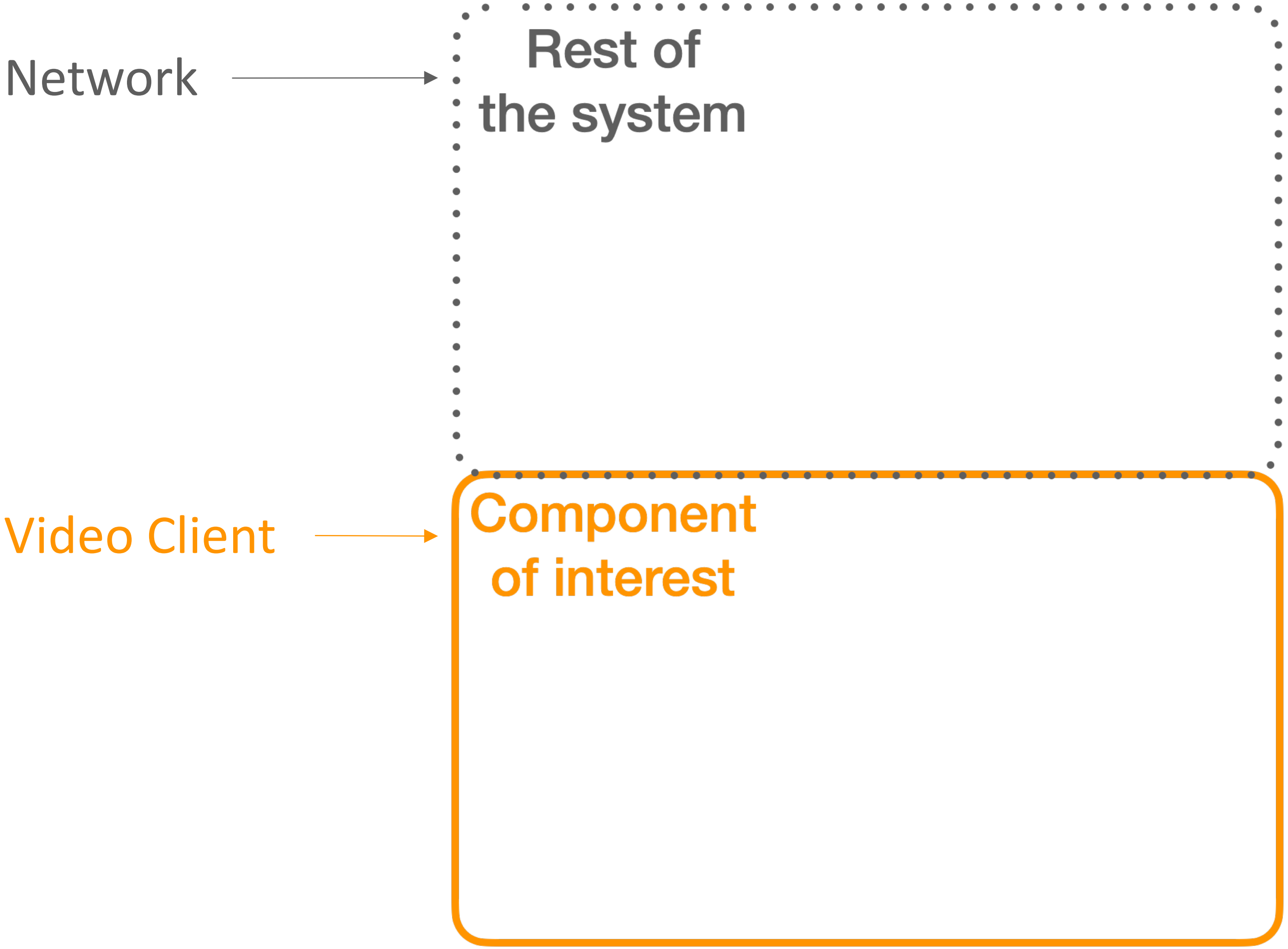
Example: Adaptive Bitrate Video Streaming (ABR)



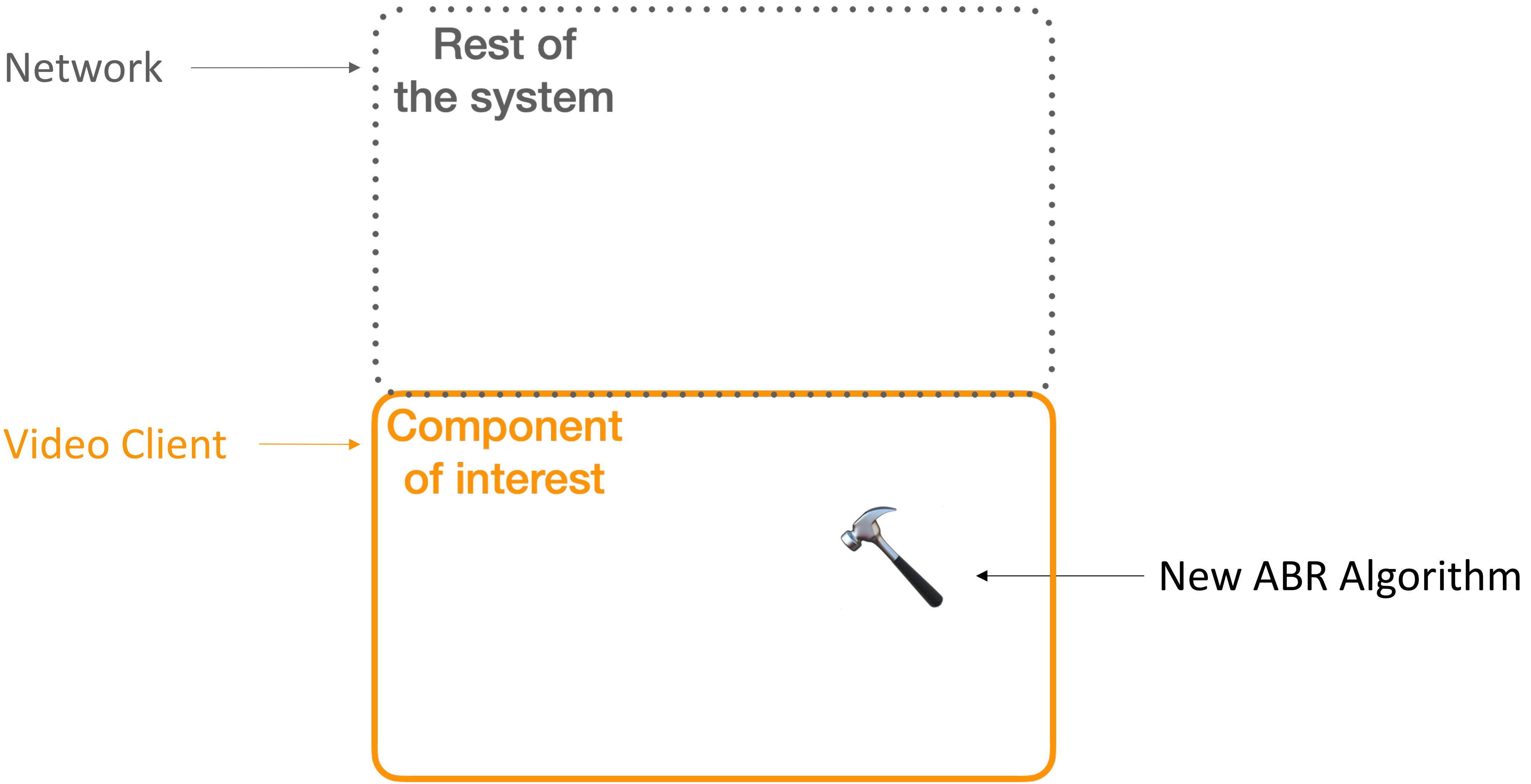
Trace-driven simulation for ABR



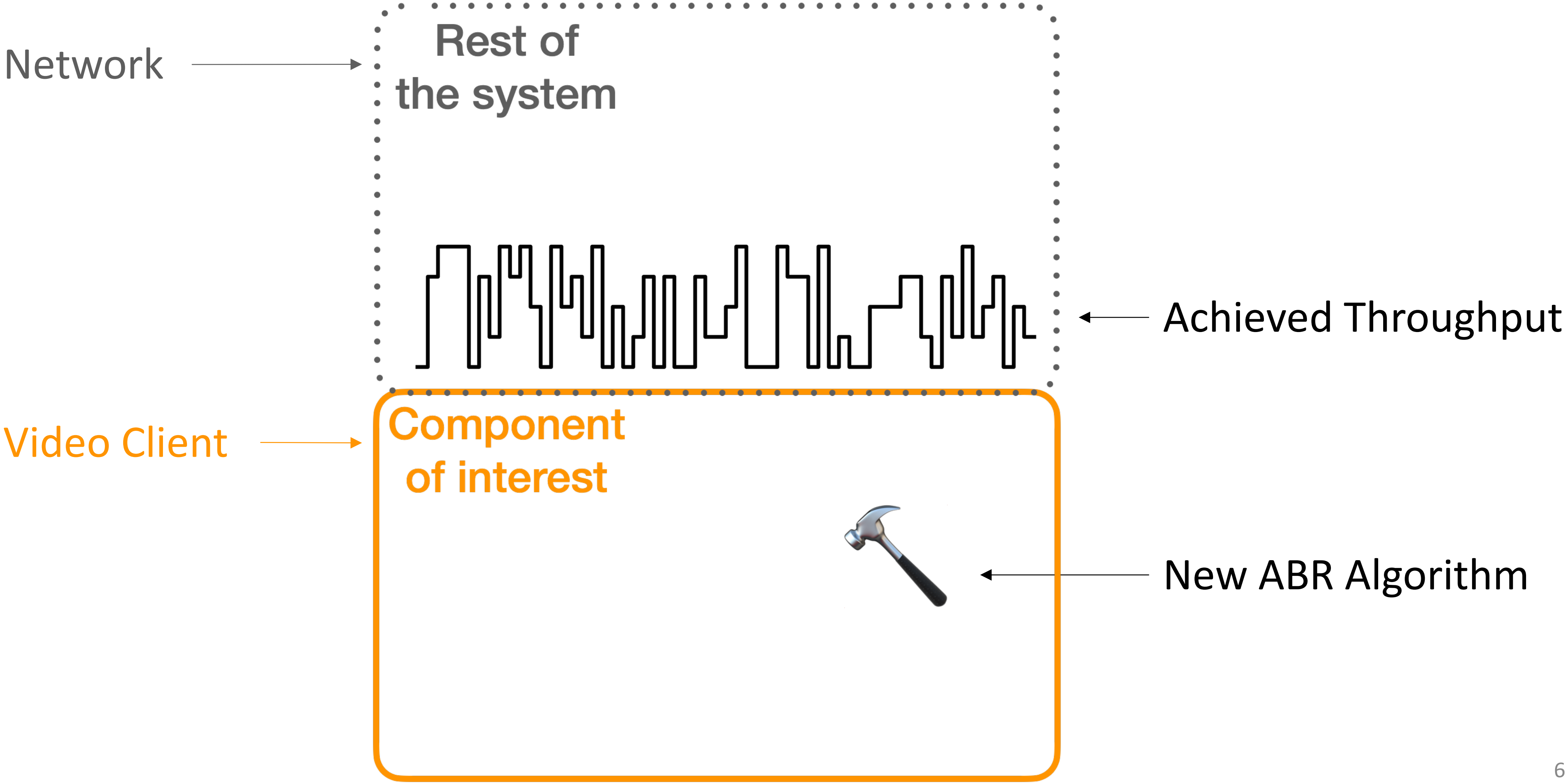
Trace-driven simulation for ABR



Trace-driven simulation for ABR



Trace-driven simulation for ABR



Trace-driven simulation for ABR

- ExpertSim

Trace-driven simulation for ABR

- ExpertSim

download time

playback buffer
(in sec)

$$b_{t+1} = \max\left(b_t - \ d_t \ , 0\right) + T$$

Trace-driven simulation for ABR

- ExpertSim

chunk size (The new algorithm’s choice)

playback buffer (in sec)

$$b_{t+1} = \max\left(b_t - \frac{a_t}{m_t}, 0\right) + T$$

achieved throughput (from the replayed trace)

Trace-driven simulation for ABR

- ExpertSim
  - [Yin et al. SIGCOMM’15][Mao et al. SIGCOMM’17]

chunk size (The new algorithm’s choice)

playback buffer (in sec)

$$b_{t+1} = \max\left(b_t - \frac{a_t}{m_t}, 0\right) + T$$

achieved throughput (from the replayed trace)

The Puffer dataset

- The Puffer Randomized Control Trial [Yan et al. NSDI 2020]
  - July 2020 – June 2021
  - 5 ABR algorithms
  - 56M downloaded chunks over 230K streaming sessions
  - 3.5 years worth of streamed video

The Puffer dataset

- The Puffer Randomized Control Trial [Yan et al. NSDI 2020]
  - July 2020 – June 2021
  - 5 ABR algorithms
  - 56M downloaded chunks over 230K streaming sessions
  - 3.5 years worth of streamed video

Task: Given the traces for all except one ABR algorithm, simulate the held out algorithm on the same paths

A typical trace

Chunk #	1	2	3	
Bitrate	360p	480p	480p	
Achieved Throughput	1Mbps	0.8Mbps	1.2Mbps	• • •
Playback Buffer	5s	3s	7s	

<u>Algorithm</u>

BBA

(“**source**”)

A typical trace

Chunk #	1	2	3	
Bitrate	360p	480p	480p	
Achieved Throughput	1Mbps	0.8Mbps	1.2Mbps	…
Playback Buffer	5s	3s	7s	

Algorithm
BBA
("**source**")

Algorithm
BOLA1
("**target**")

A typical trace

Chunk #	1	2	3	
Bitrate	360p	480p	480p	
Achieved Throughput	1Mbps	0.8Mbps	1.2Mbps	• • •
Playback Buffer	5s	3s	7s	

<u>Algorithm</u>
BBA
("**source**")

A simulated trajectory

Chunk #	1
Bitrate	**720p**
Achieved Throughput	**?**
Playback Buffer	**?**

<u>Algorithm</u>
BOLA1
("**target**")

A typical trace

Chunk #	1	2	3	
Bitrate	360p	480p	480p	
Achieved Throughput	1Mbps	0.8Mbps	1.2Mbps	…
Playback Buffer	5s	3s	7s	

Algorithm
BBA
(“**source**”)

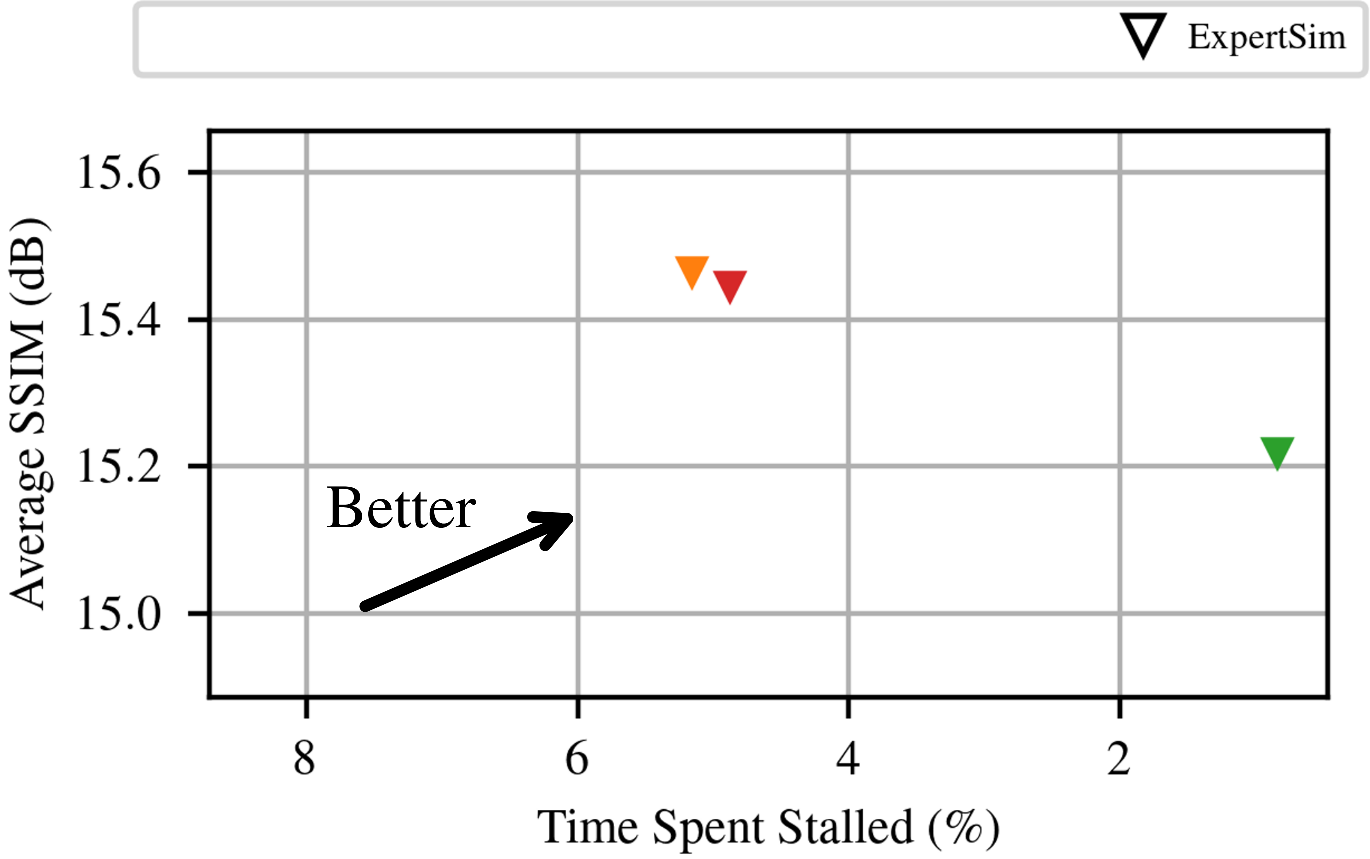
A simulated trajectory

Chunk #	1	2	3	
Bitrate	**720p**	**?**	**?**	
Achieved Throughput	**?**	**?**	**?**	…
Playback Buffer	**?**	**?**	**?**	

Algorithm
BOLA1
(“**target**”)

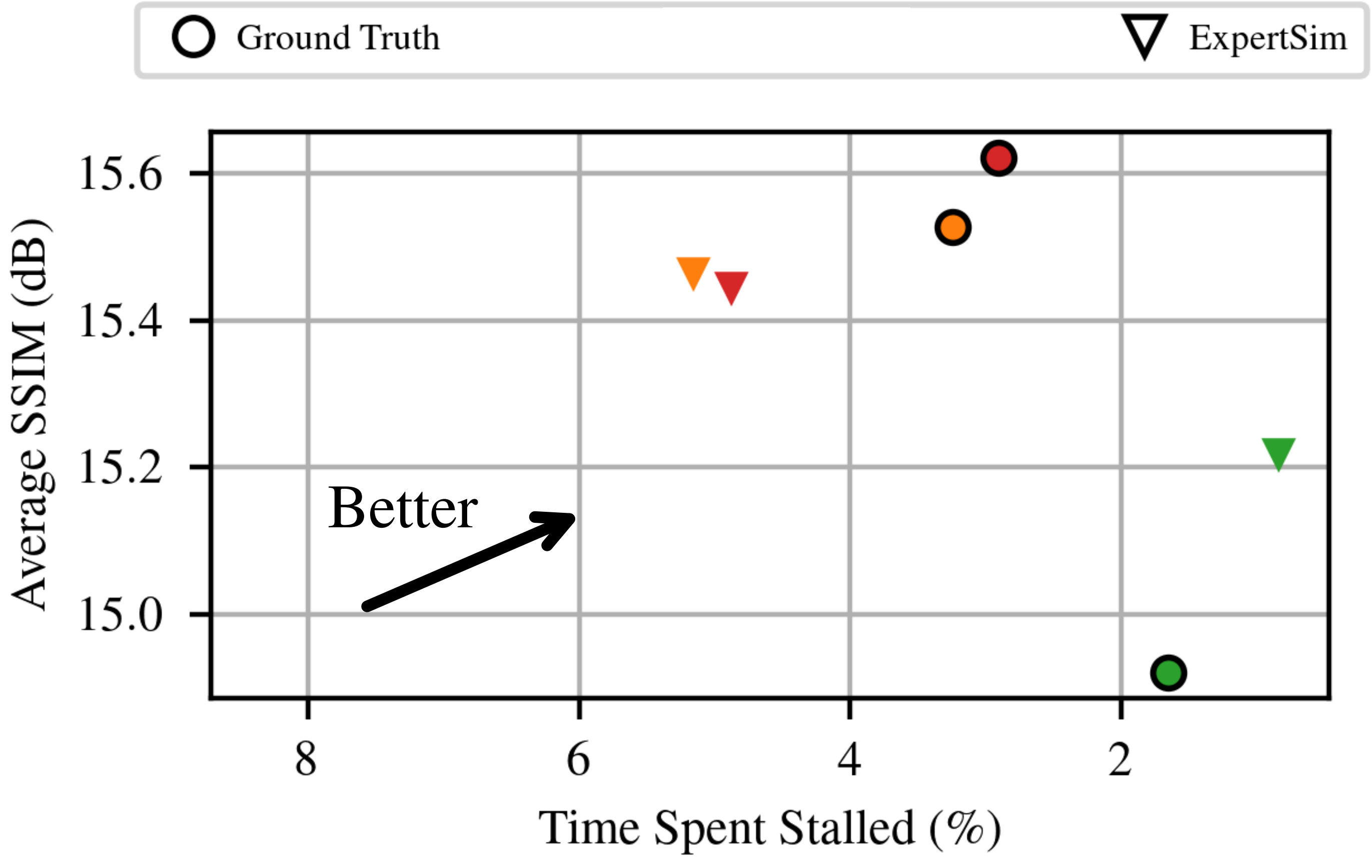
How accurate is trace-driven simulation?

- Source data: The Puffer Dataset with 5 algorithms
- Target algorithm (unseen): BBA, BOLA1, BOLA2



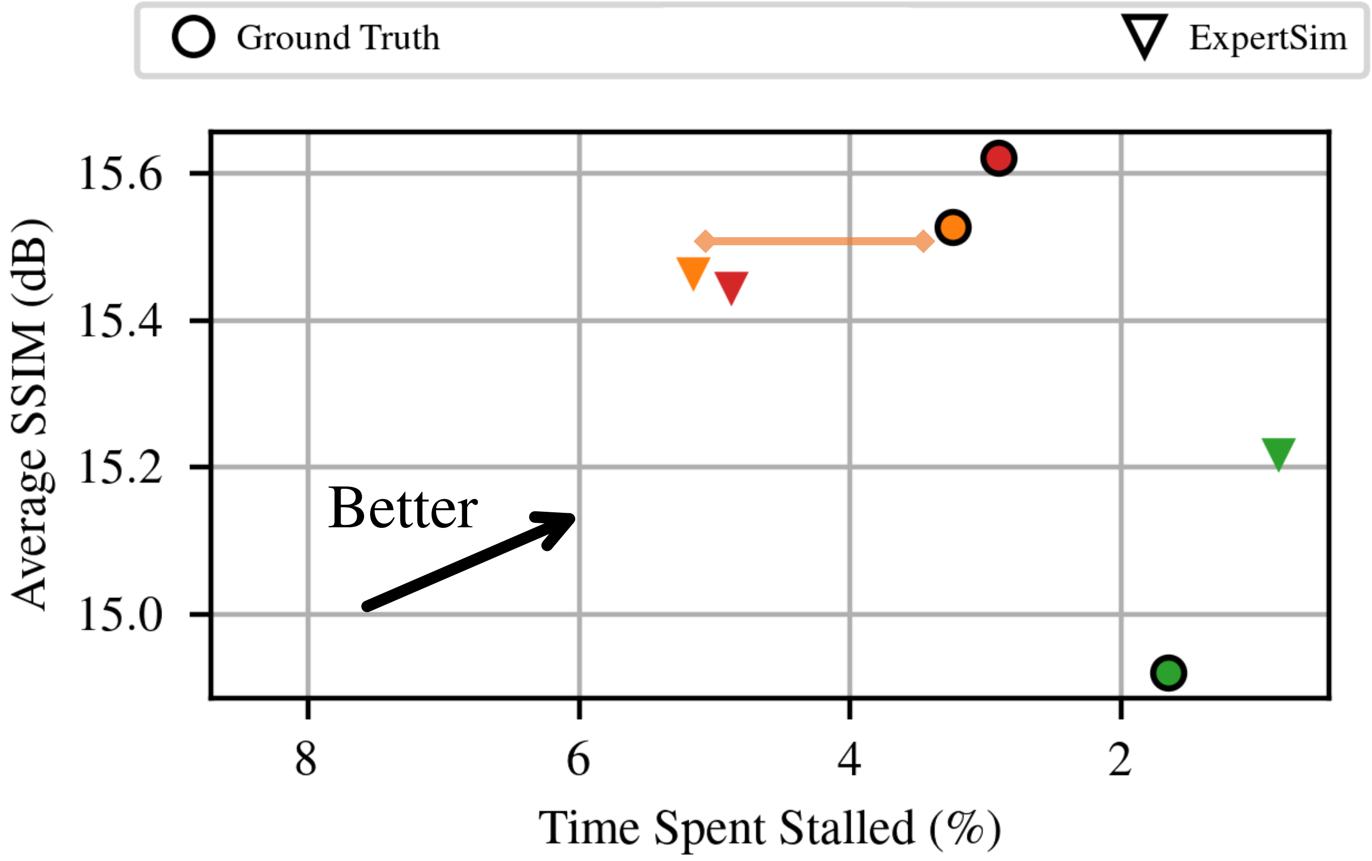
How accurate is trace-driven simulation?

- Source data: The Puffer Dataset with 5 algorithms
- Target algorithm (unseen): BBA, BOLA1, BOLA2



How accurate is trace-driven simulation?

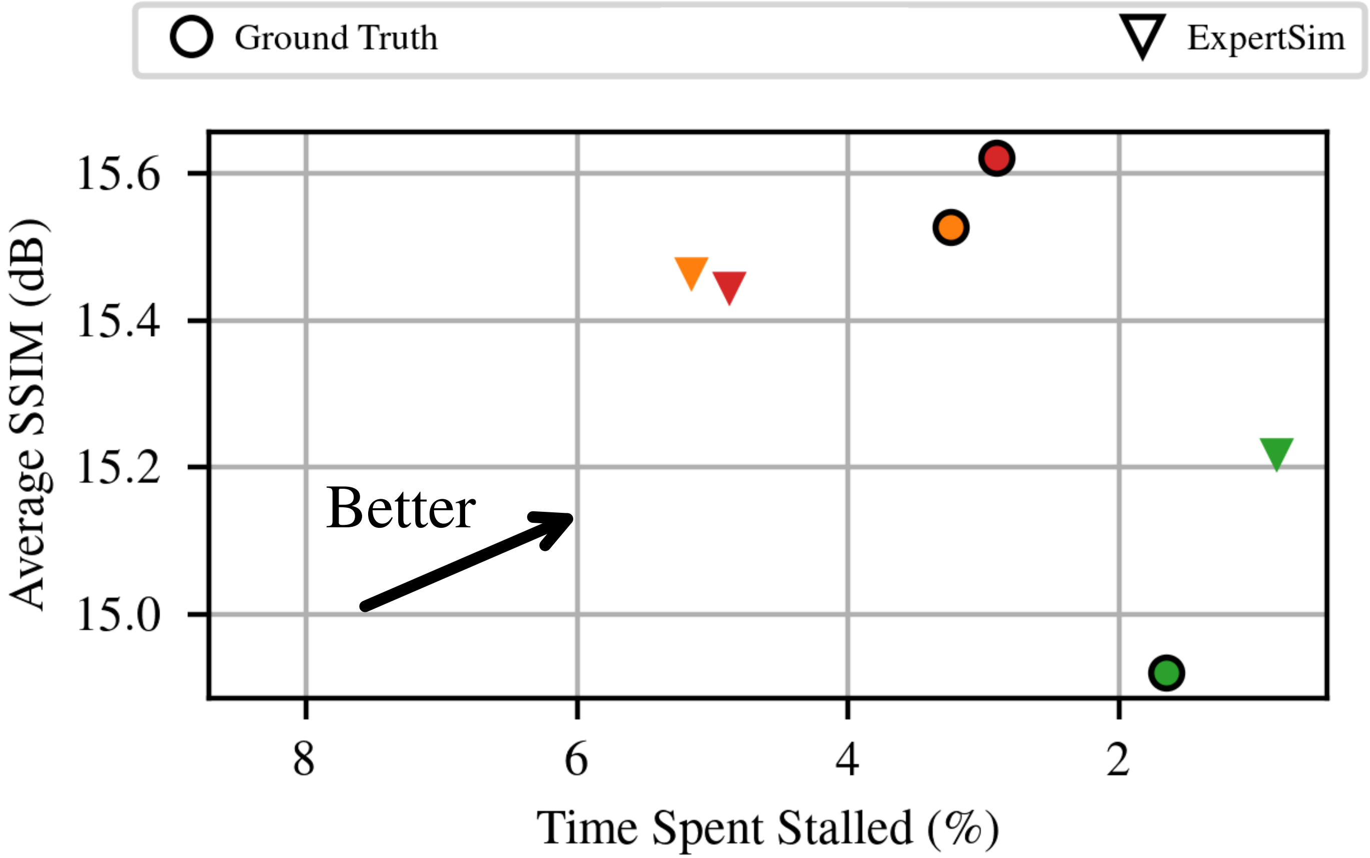
- Source data: The Puffer Dataset with 5 algorithms
- Target algorithm (unseen): BBA, BOLA1, BOLA2



How accurate is trace-driven simulation?

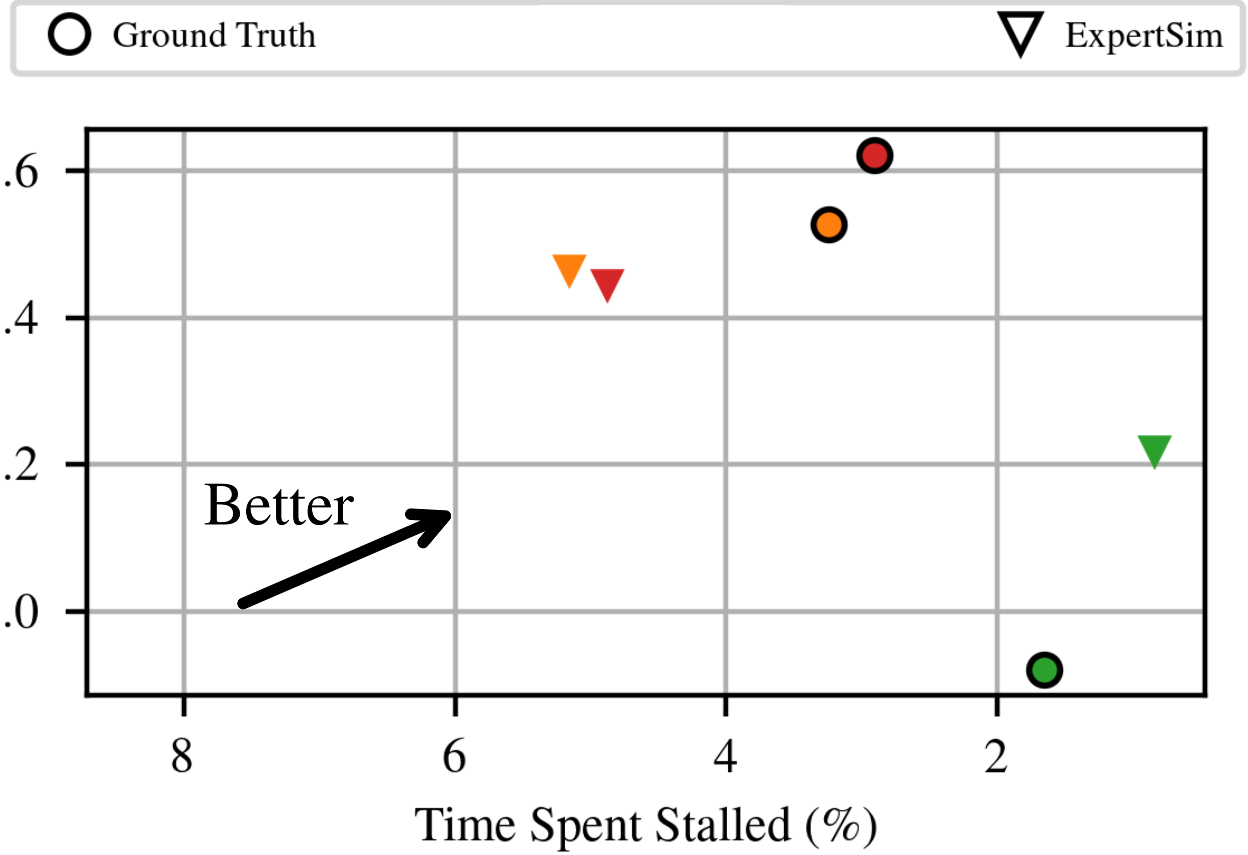
- Source data: The Puffer Dataset with 5 algorithms
- Target algorithm (unseen): BBA, BOLA1, BOLA2

Exogenous trace assumption



How accurate is trace-driven simulation?

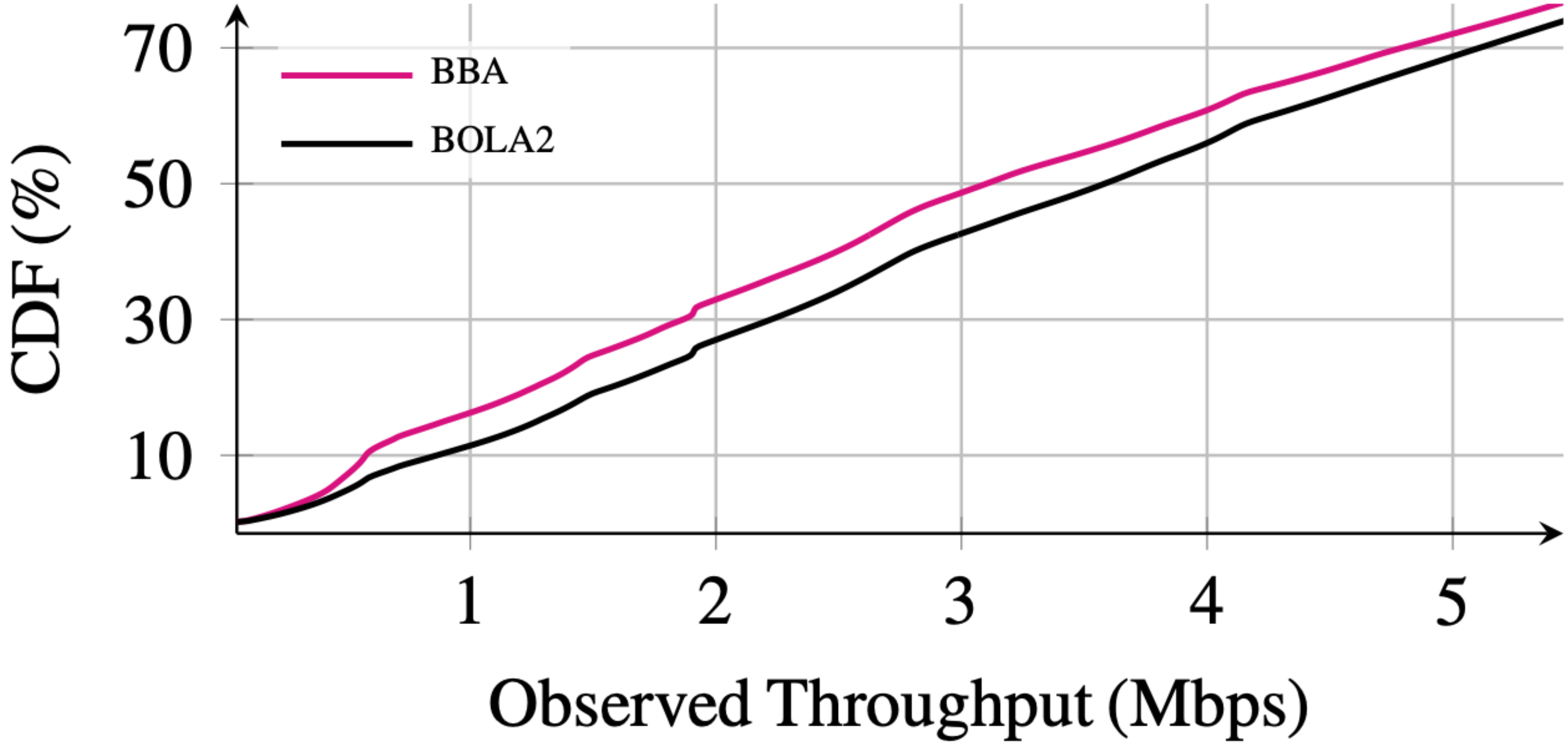
- Source data: The Puffer Dataset with 5 algorithms
- Target algorithm (unseen): BBA, BOLA1, BOLA2



Exogenous trace assumption

Bitrates chosen by the ABR algorithm affect the achieved throughput.

ABR algorithms affect throughput



Can we relax the exogenous trace assumption?

Can we relax the exogenous trace assumption?

$$\overset{\substack{\text{Achieved Throughput}\\ \nwarrow}}{m_t}$$

Can we relax the exogenous trace assumption?

Achieved Throughput

$$m_t = f(\qquad\qquad)$$

Can we relax the exogenous trace assumption?

$$\overset{\text{Achieved Throughput}}{\uparrow}\; m_t = f(\overset{\text{Bitrate}}{\uparrow}\; a_t \qquad)$$

Can we relax the exogenous trace assumption?

$$\overset{\text{Achieved Throughput}}{\overset{\nwarrow}{m_t}} = f\left(\overset{\text{Bitrate}}{\overset{\uparrow}{a_t}}, \overset{\text{Latent Network Conditions}}{\overset{\nearrow}{u_t}}\right)$$

Can we relax the exogenous trace assumption?

Achieved Throughput Bitrate Latent Network Conditions

$$m_t = f(a_t, u_t)$$

Both u_t and $f(\cdot)$ are unknown

Towards a solution

$$m_t = f(a_t, u_t)$$

Towards a solution

- If u and $f(\cdot)$ were known…

$$m_t = f(a_t, u_t)$$

Towards a solution

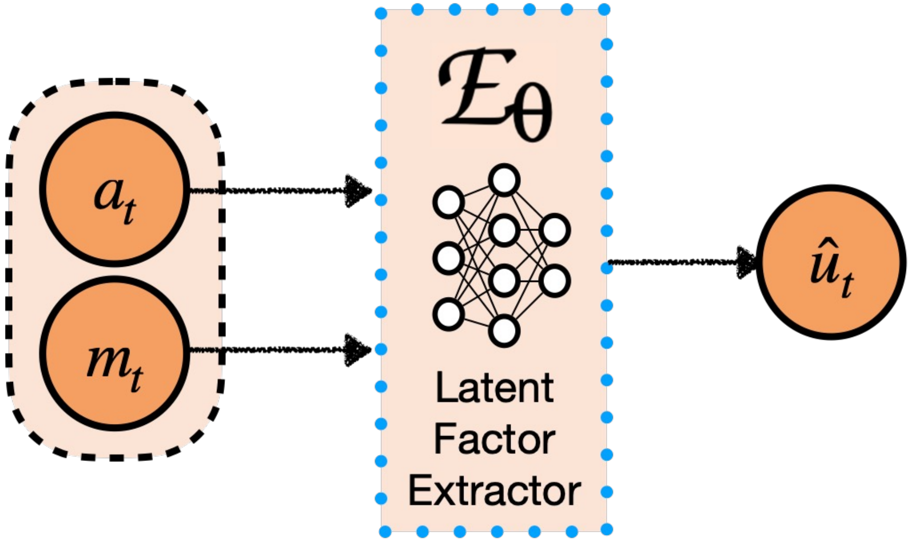
- If u and $f(\cdot)$ were known…

$$\widetilde{m}_t = f(\tilde{a}_t, u_t)$$

simulated bitrate
(counterfactual)

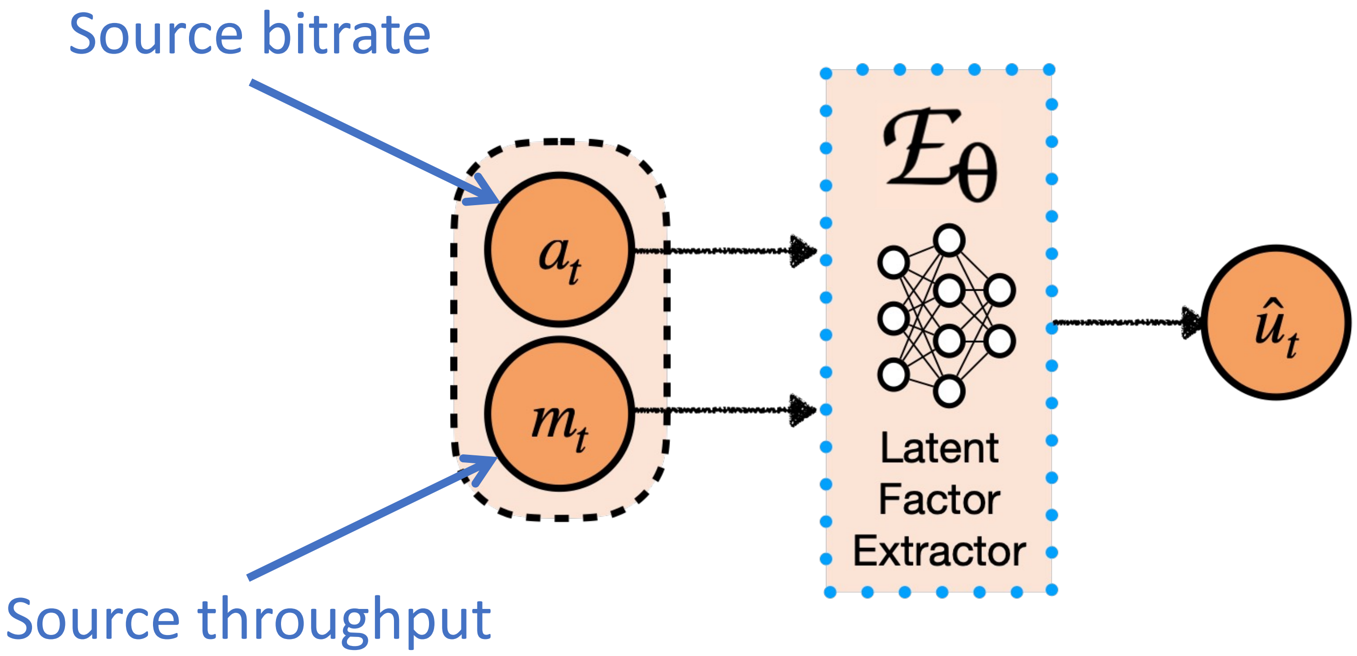
A learning approach

$m_t = f(a_t, u_t)$



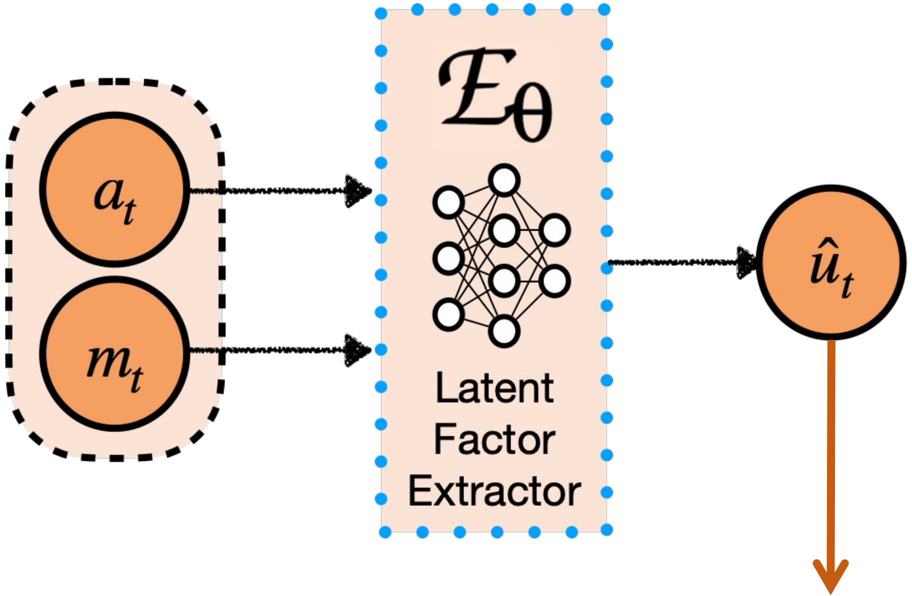
A learning approach

$$m_t = f(a_t, u_t)$$



A learning approach

$m_t = f(a_t, u_t)$

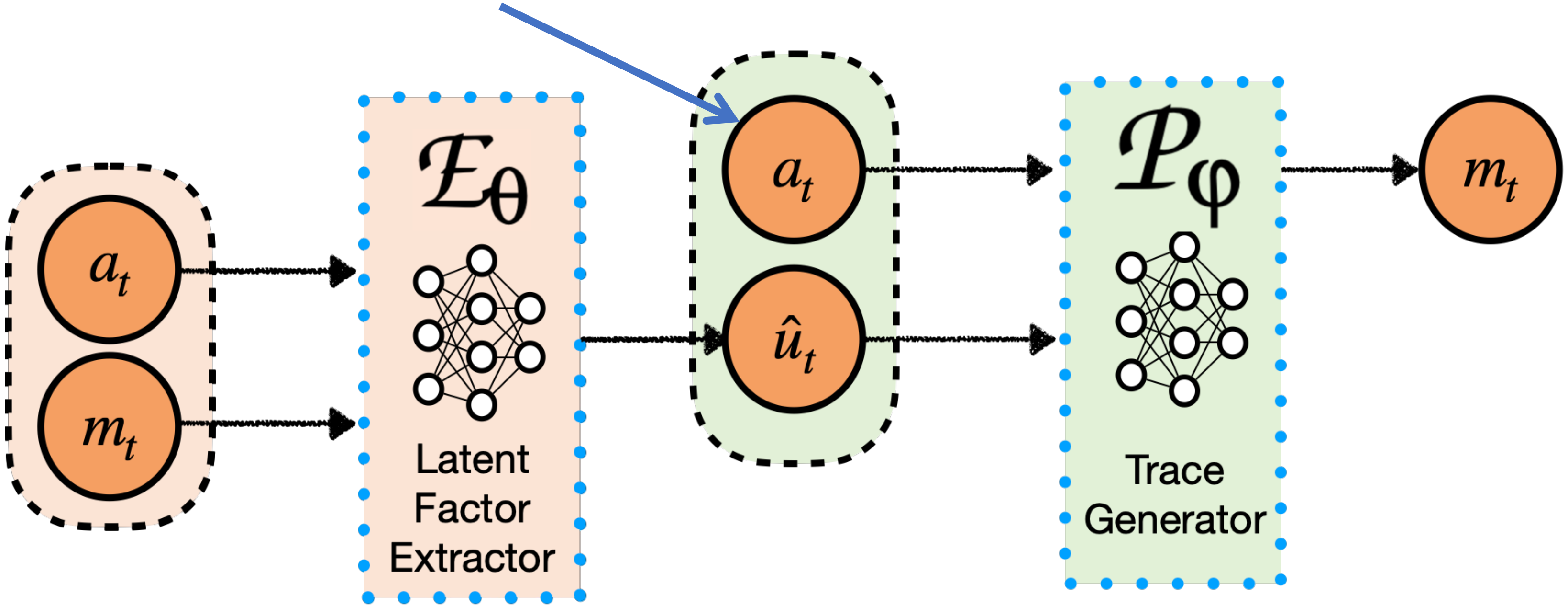


Informative for throughput reconstruction

A learning approach

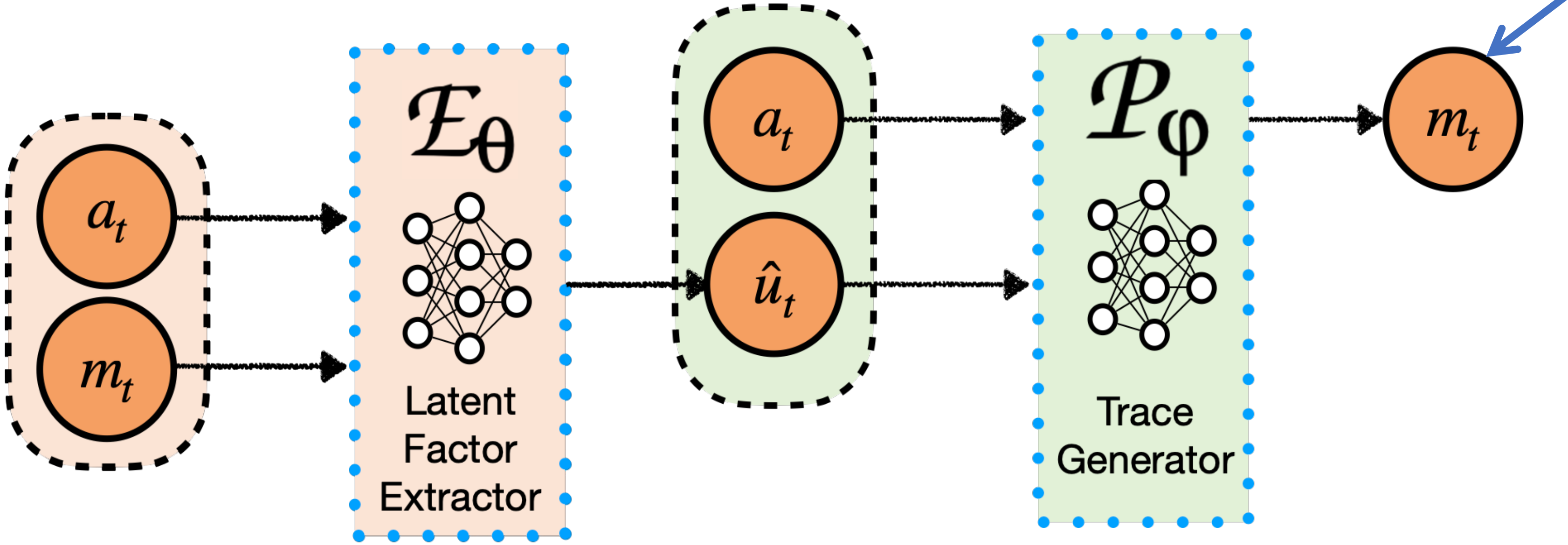
$m_t = f(a_t, u_t)$

Source bitrate



A learning approach

$m_t = f(a_t, u_t)$



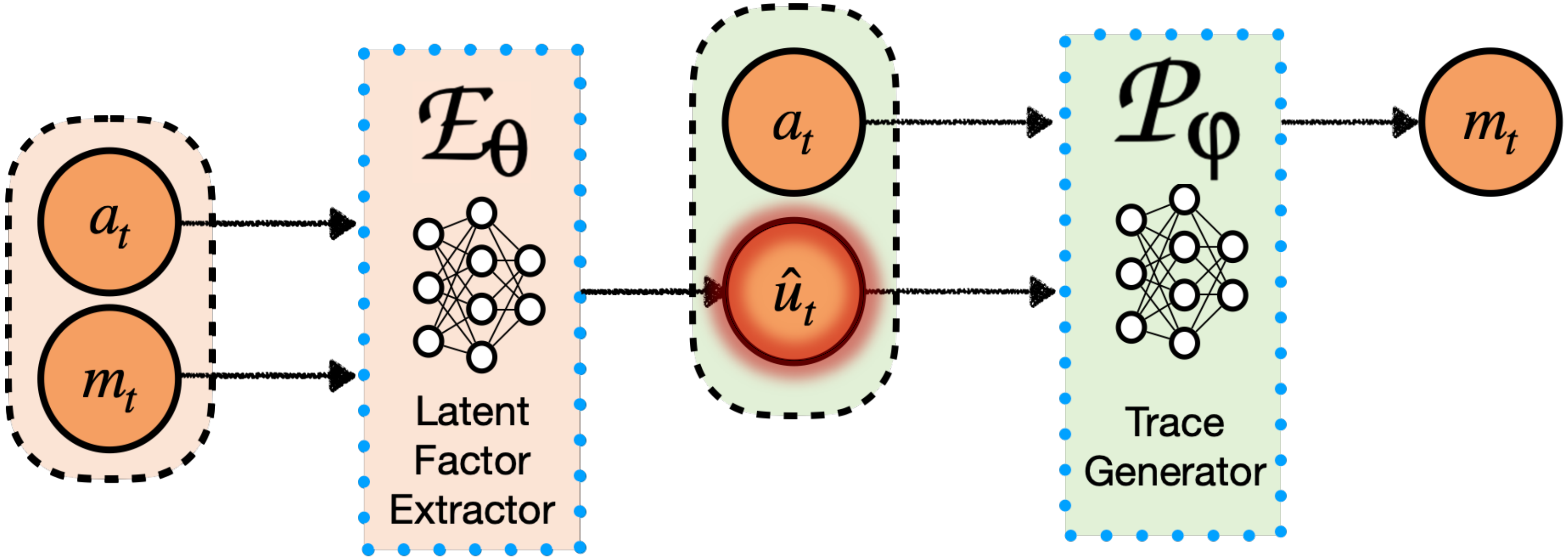
Source throughput

Learning goal

- Fit observed data

A learning approach

$$m_t = f(a_t, u_t)$$

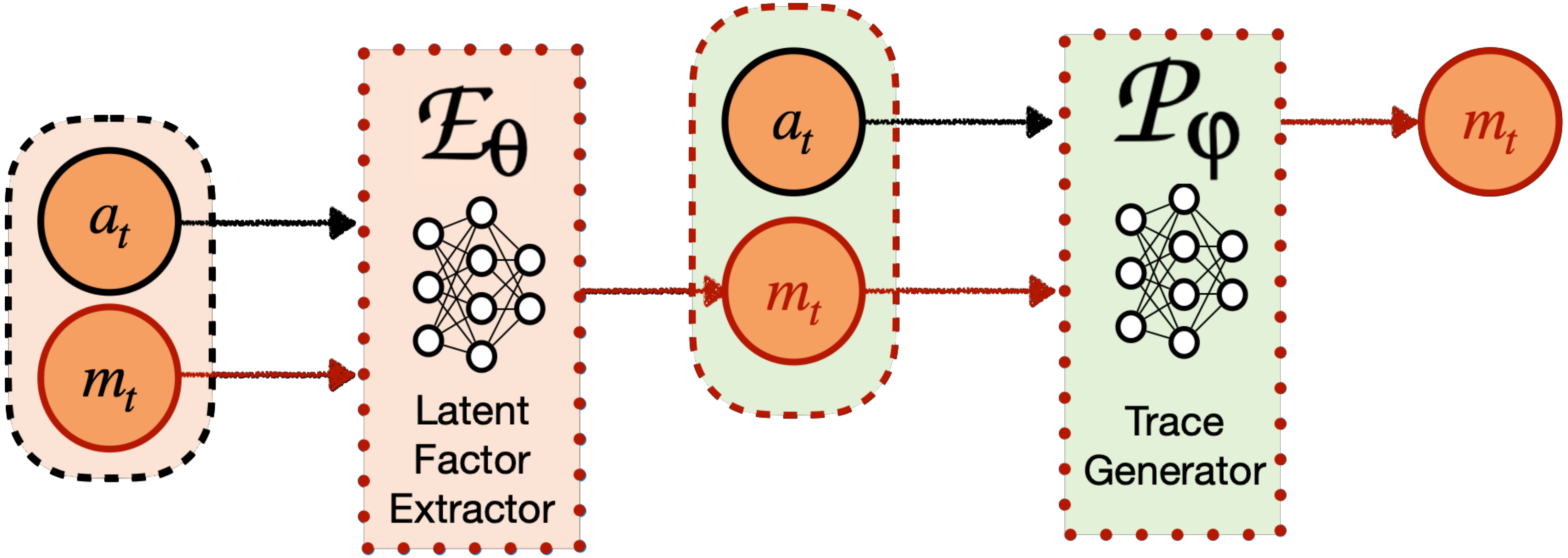


Learning goal

- Fit observed data

A learning approach

$m_t = f(a_t, u_t)$



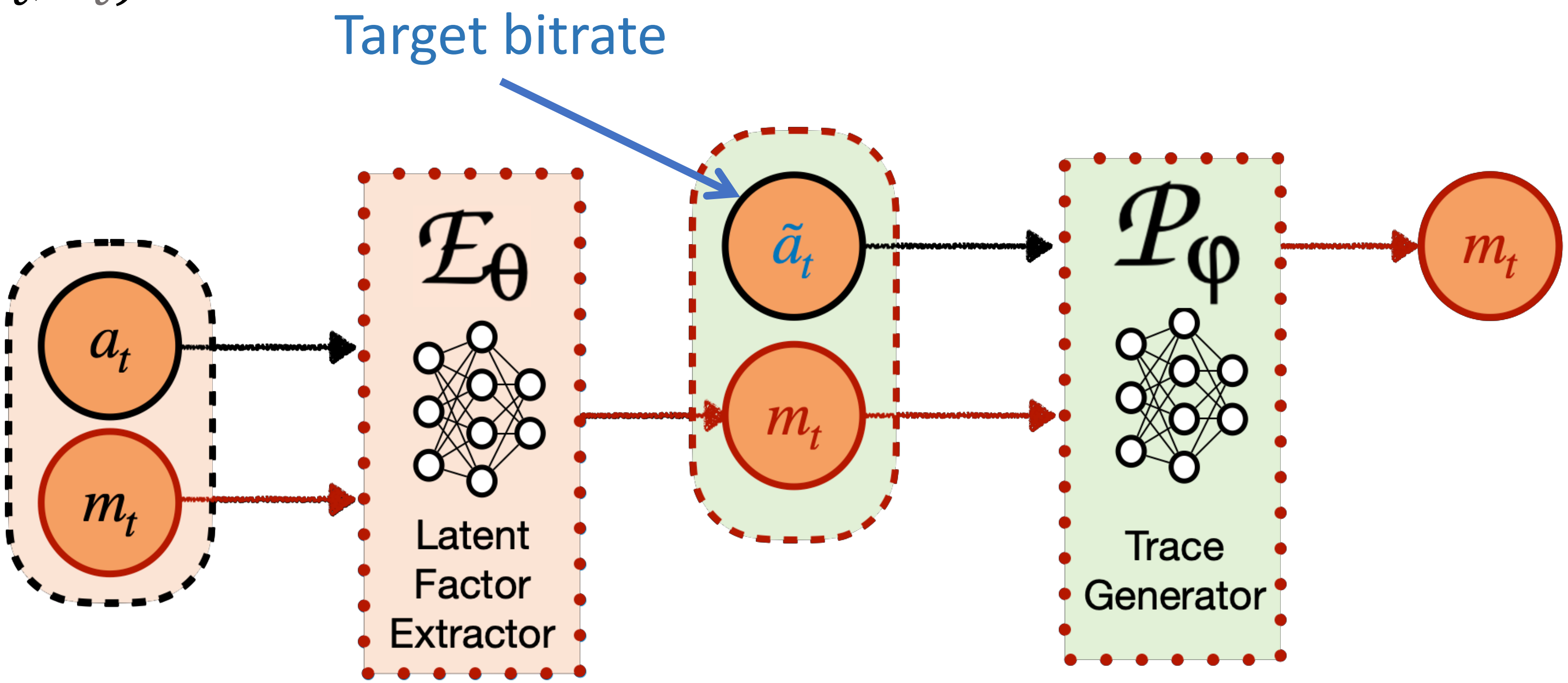
Learning goal

- Fit observed data

A degenerate solution

A learning approach

$m_t = f(a_t, u_t)$



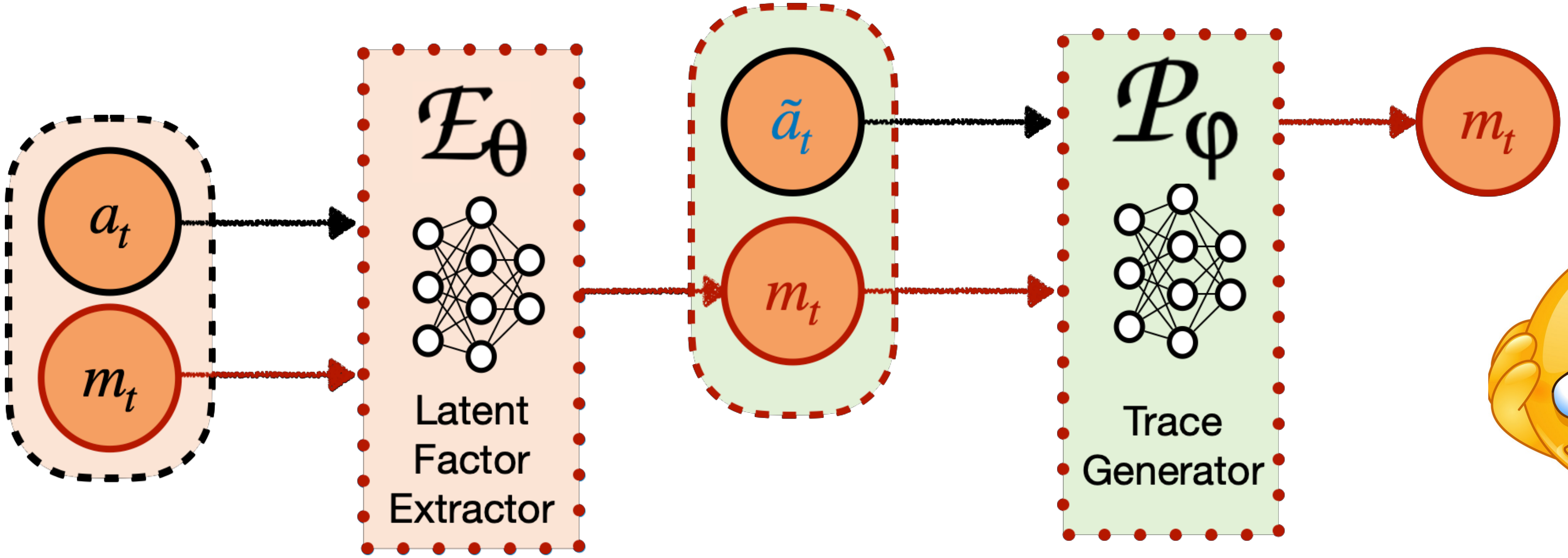
Learning goal

- Fit observed data

A degenerate solution

A learning approach

$m_t = f(a_t, u_t)$



Learning goal

- Fit observed data

A degenerate solution

RCT to the rescue!

💡 ***RCT property:*** Distribution of latent network conditions is the same in trajectories assigned to different algorithms.

RCT to the rescue!

RCT property: Distribution of latent network conditions is the same in trajectories assigned to different algorithms.

↳ Latent network condition is independent of the source algorithm (used for trace collection).

RCT to the rescue!

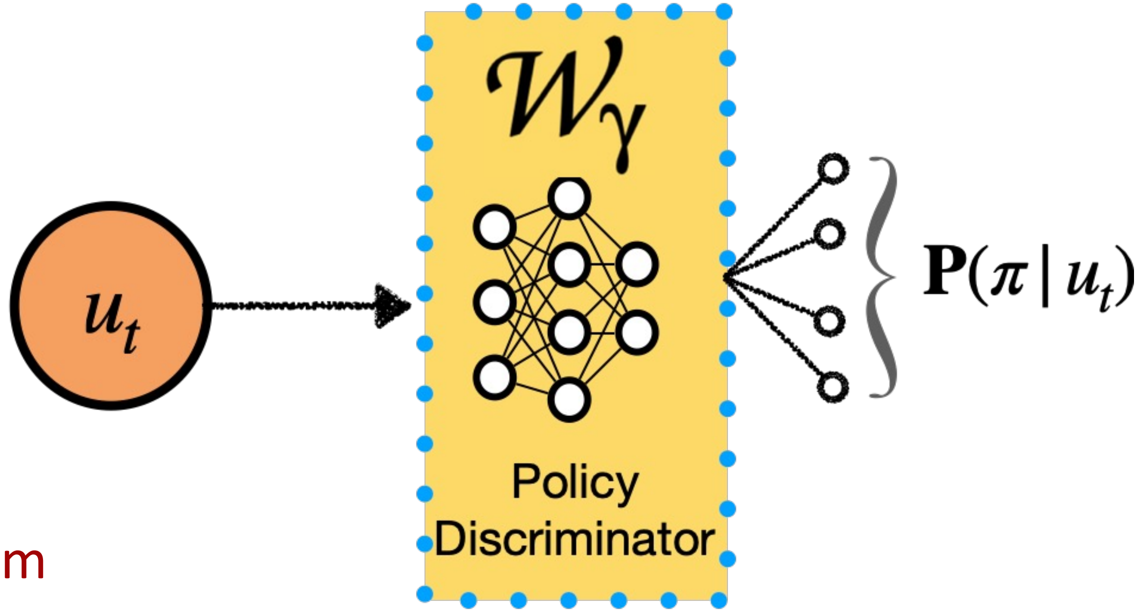
💡 ***RCT property:*** Distribution of latent network conditions is the same in trajectories assigned to different algorithms.

↳ Latent network condition is independent of the source algorithm (used for trace collection).

↳ Latent network condition does not give any information about the source algorithm.

RCT to the rescue!

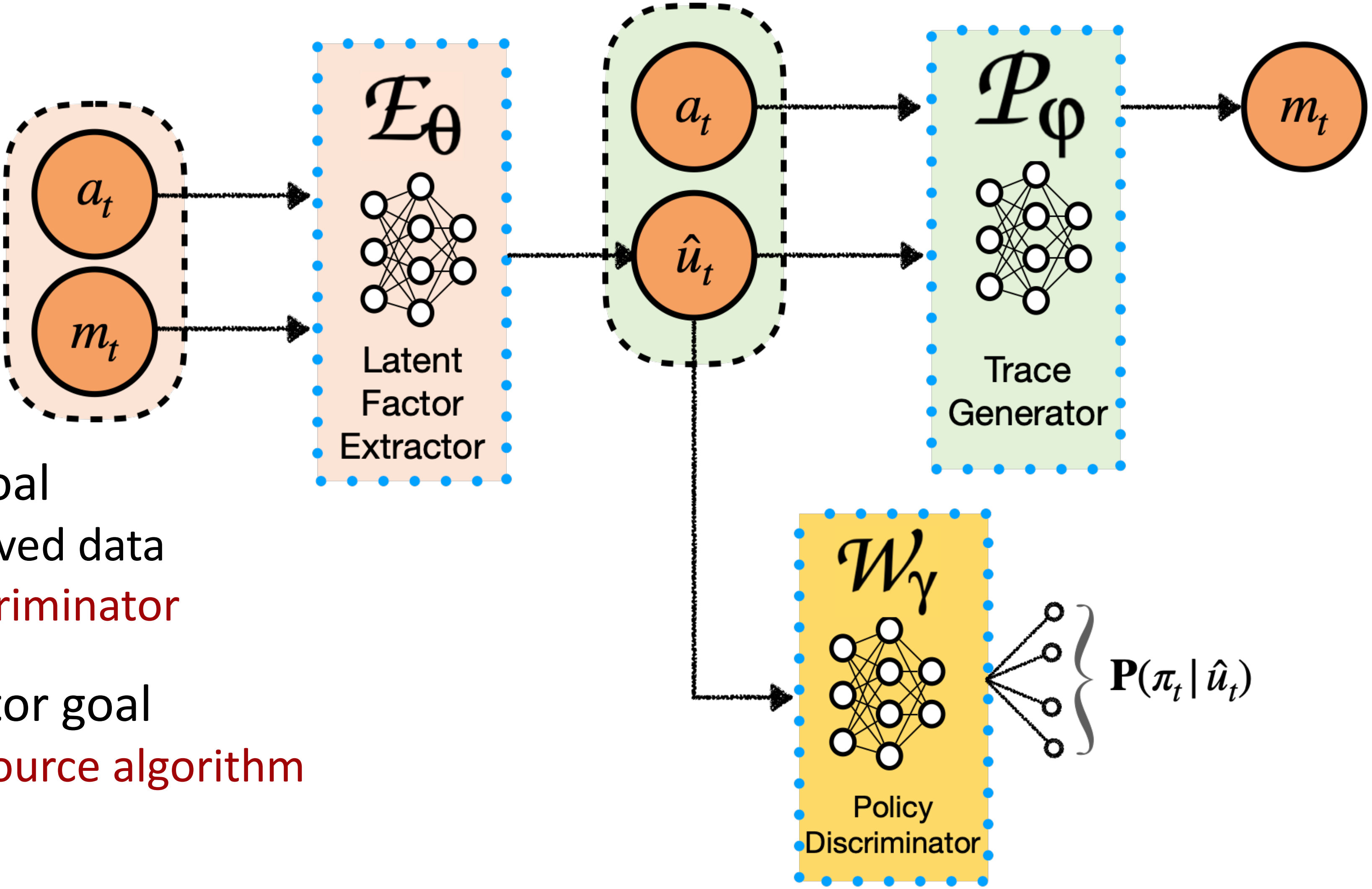
RCT property: Distribution of latent network conditions is the same in trajectories assigned to different algorithms.



Discriminator goal

- Predict source algorithm

Exploiting the RCT property



Learning goal

- Fit observed data
- Fool discriminator

Discriminator goal

- Predict source algorithm

RCT property is sufficient for unbiased simulation

Theorem: The RCT property (independence of u and the algorithm) is sufficient for estimating the counterfactual trace if

RCT property is sufficient for unbiased simulation

Theorem: The RCT property (independence of u and the algorithm) is sufficient for estimating the counterfactual trace if

1. (**Invertibility**) $\forall \mathrm{a}:\ \mathrm{m} = \mathrm{f}(\mathrm{a}, \mathrm{u})$ is invertible.

RCT property is sufficient for unbiased simulation

Theorem: The RCT property (independence of u and the algorithm) is sufficient for estimating the counterfactual trace if

1. (**Invertibility**) $\forall \mathrm{a}$: $\mathrm{m} = \mathrm{f}(\mathrm{a}, \mathrm{u})$ is invertible.
2. (**Low-rank factorization**) Matrix representation of f has rank r, and $\mathrm{r} \leq \dim(\mathrm{trace})$.

RCT property is sufficient for unbiased simulation

Theorem: The RCT property (independence of u and the algorithm) is sufficient for estimating the counterfactual trace if

1. (**Invertibility**) $\forall \mathrm{a}:\ \mathrm{m} = \mathrm{f}(\mathrm{a}, \mathrm{u})$ is invertible.
2. (**Low-rank factorization**) Matrix representation of f has rank r, and $\mathrm{r} \leq \dim(\mathrm{trace})$.
3. Traces are collected using sufficient number of **diverse** algorithms (See the paper for the precise statement).

Fulfilling the initial promise

Fulfilling the initial promise

- Key source of bias
  - Exogenous Trace Assumption

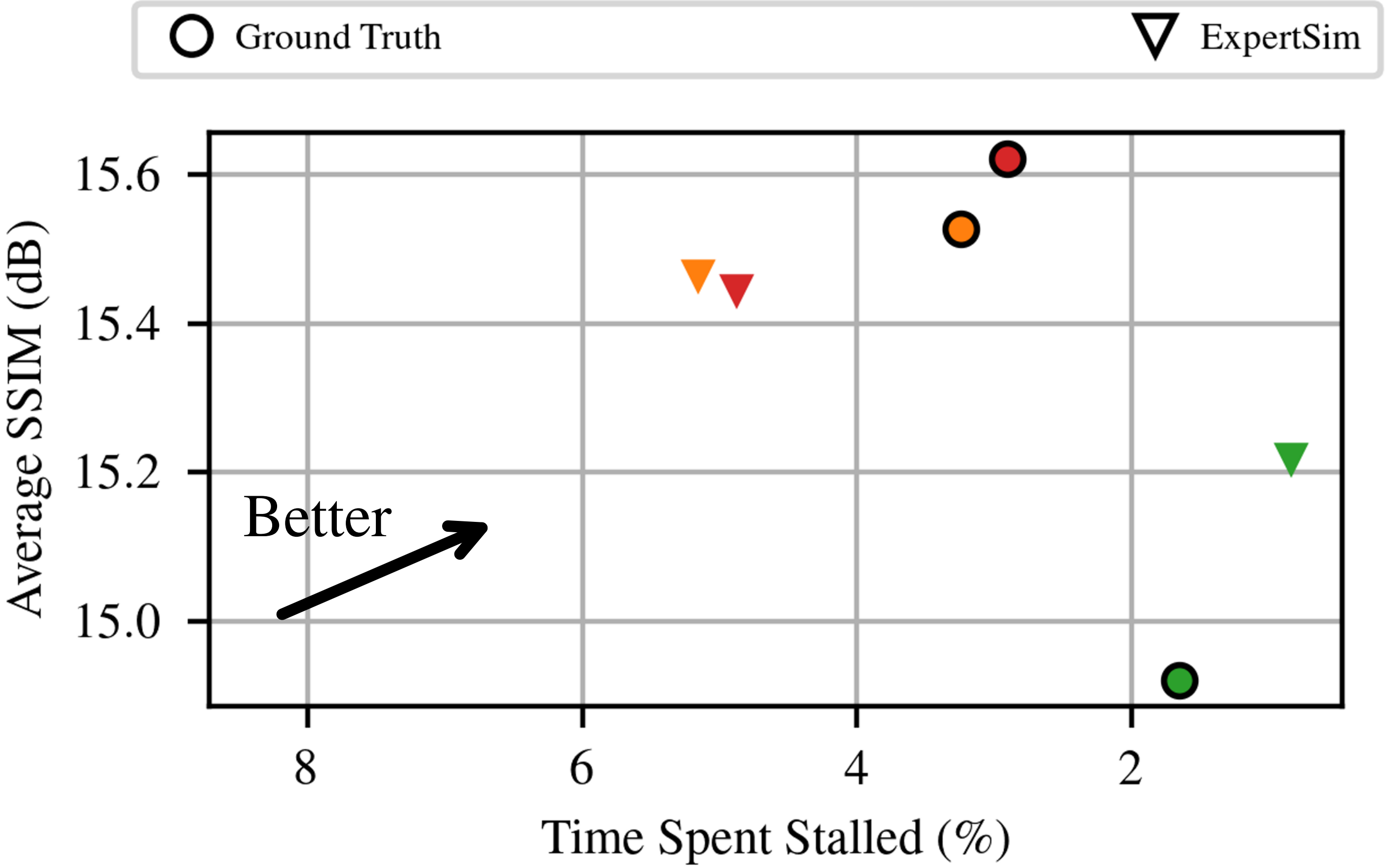
Fulfilling the initial promise

- Key source of bias
  - Exogenous Trace Assumption

- How to do unbiased trace-driven simulation?
  - CausalSim

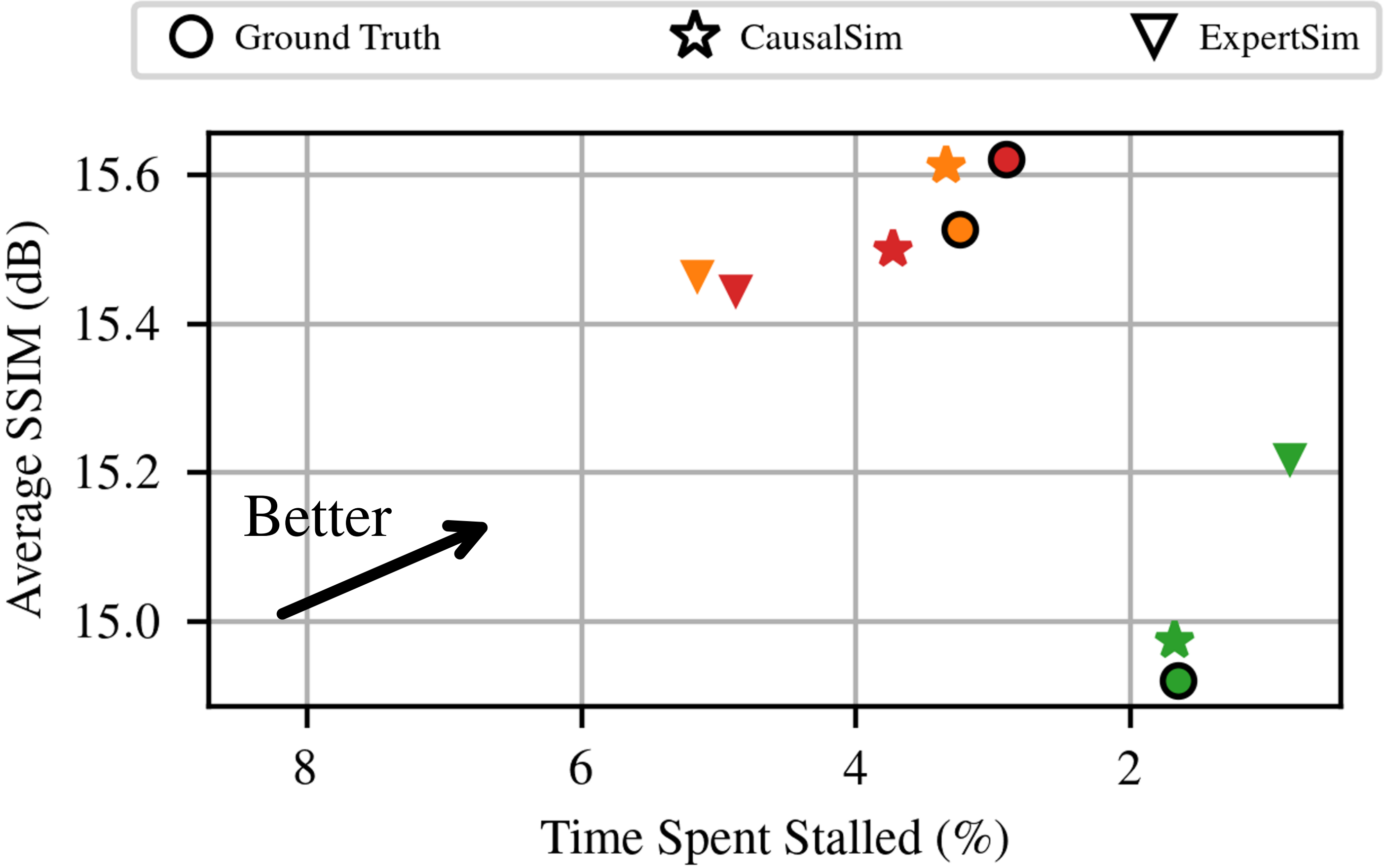
Simulation accuracy

- Source data: The Puffer Dataset with 5 algorithms
- Target algorithm (unseen in training): BBA, BOLA1, BOLA2



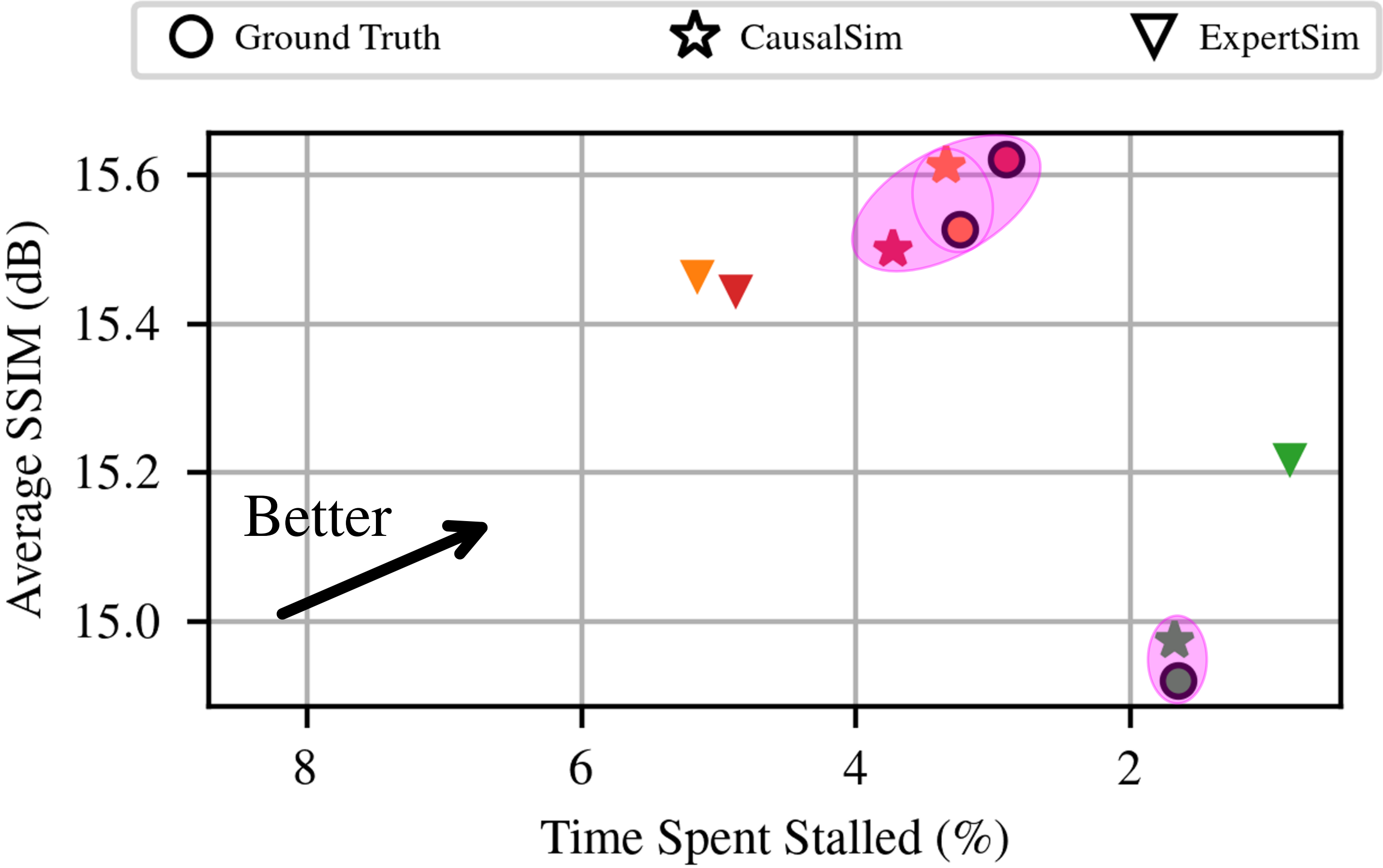
Simulation accuracy

- Source data: The Puffer Dataset with 5 algorithms
- Target algorithm (unseen in training): BBA, BOLA1, BOLA2



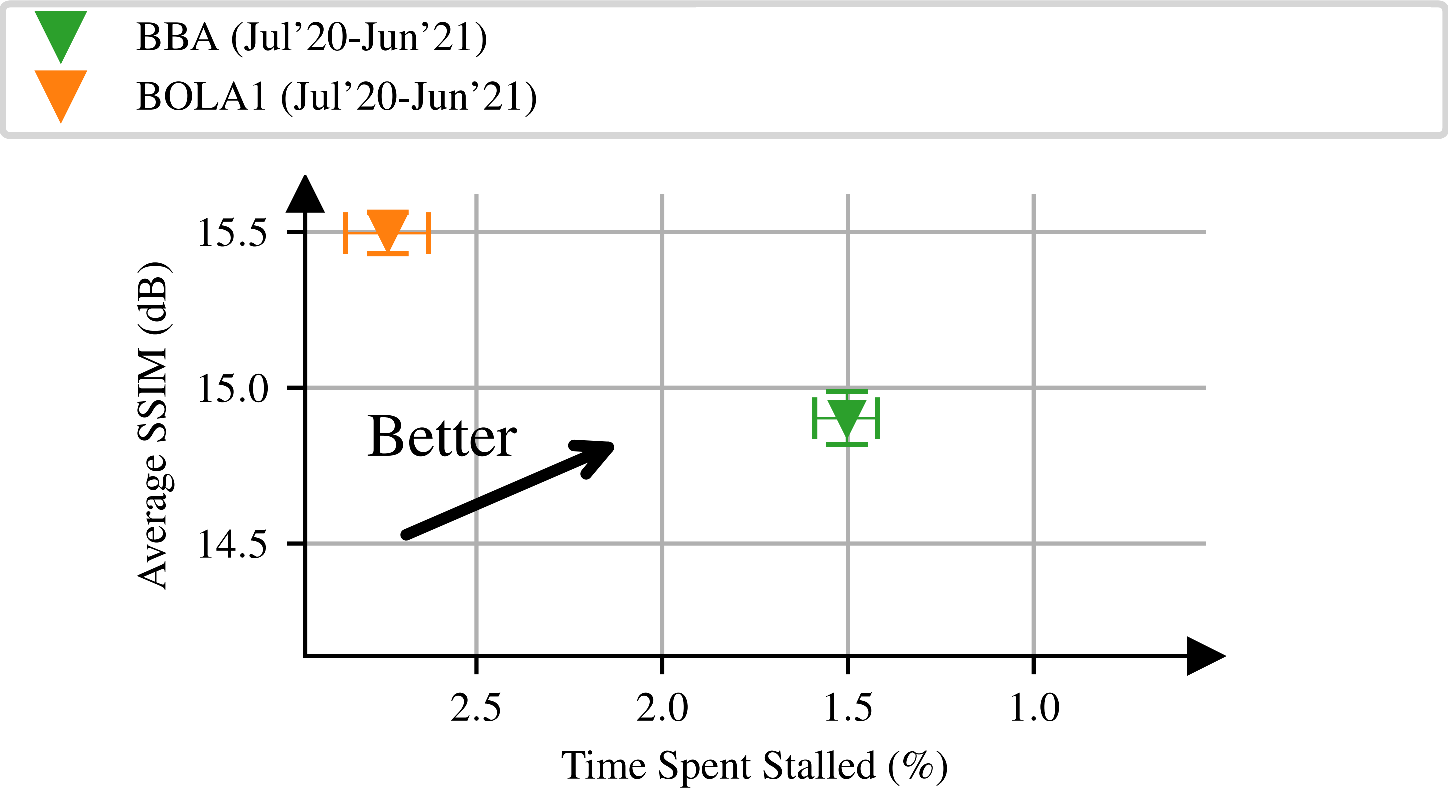
Simulation accuracy

- Source data: The Puffer Dataset with 5 algorithms
- Target algorithm (unseen in training): BBA, BOLA1, BOLA2



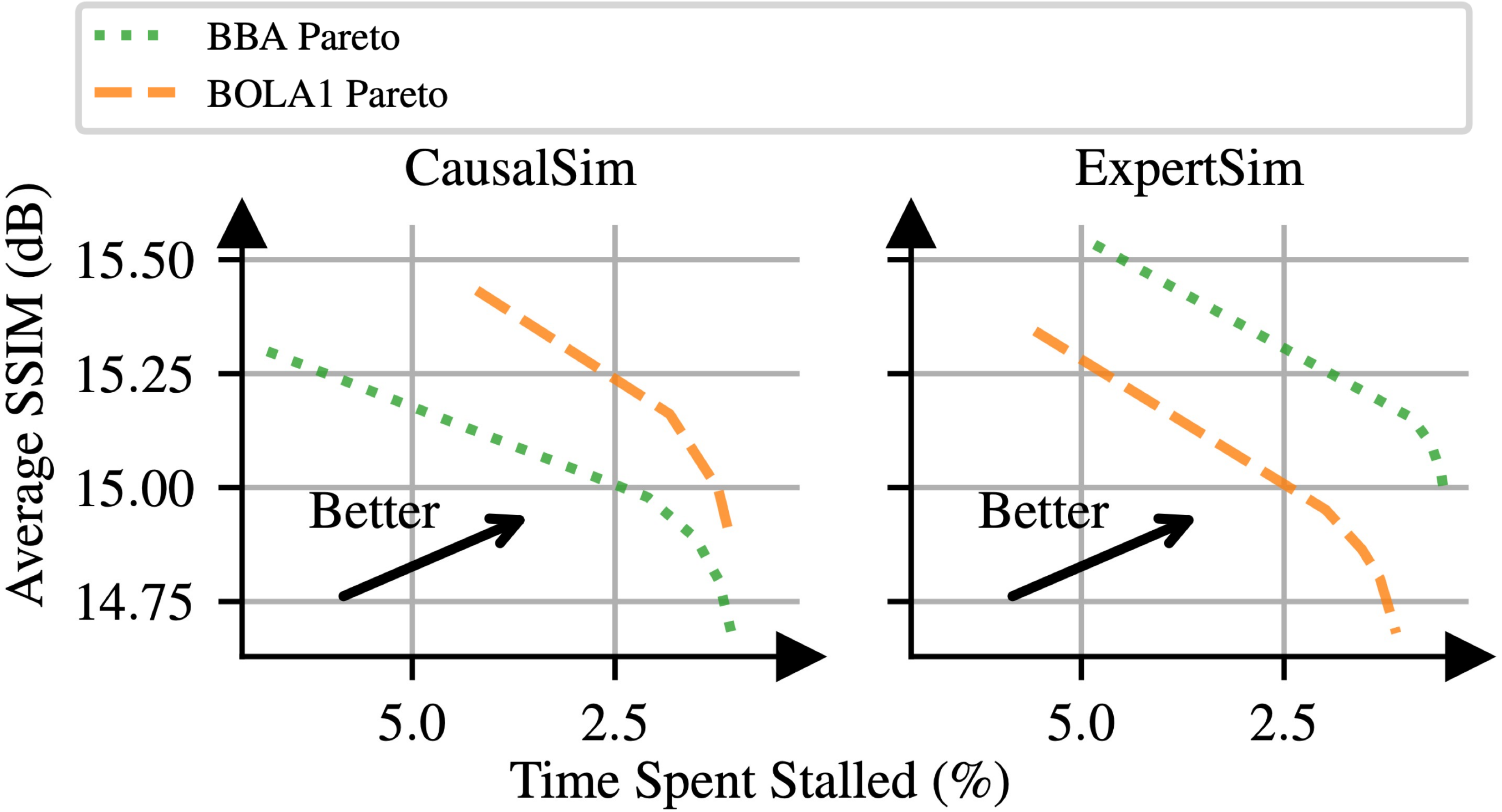
Case study: CausalSim in the wild

Case study: CausalSim in the wild

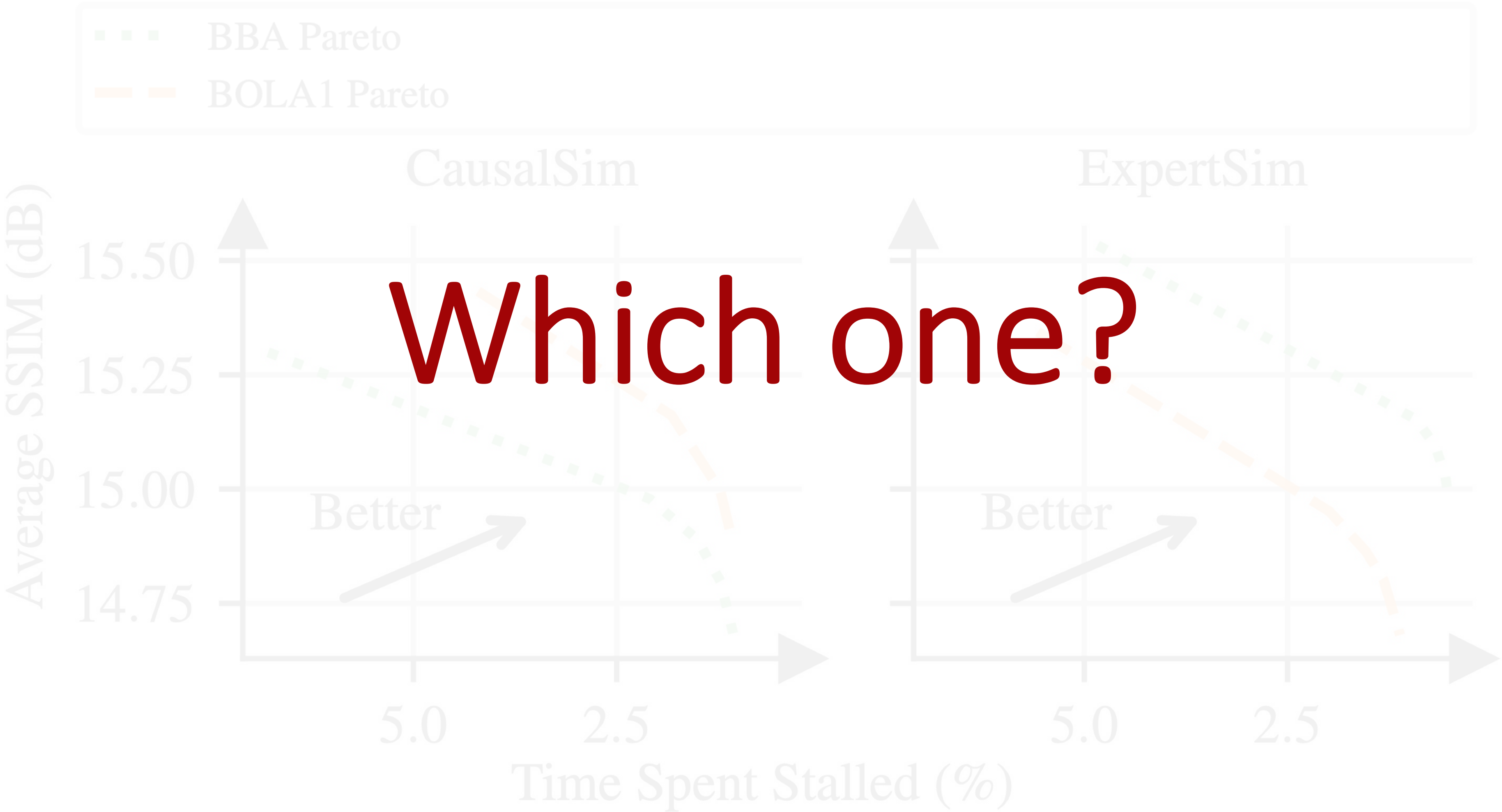


- Can we use CausalSim to improve algorithms?

Case study: CausalSim in the wild

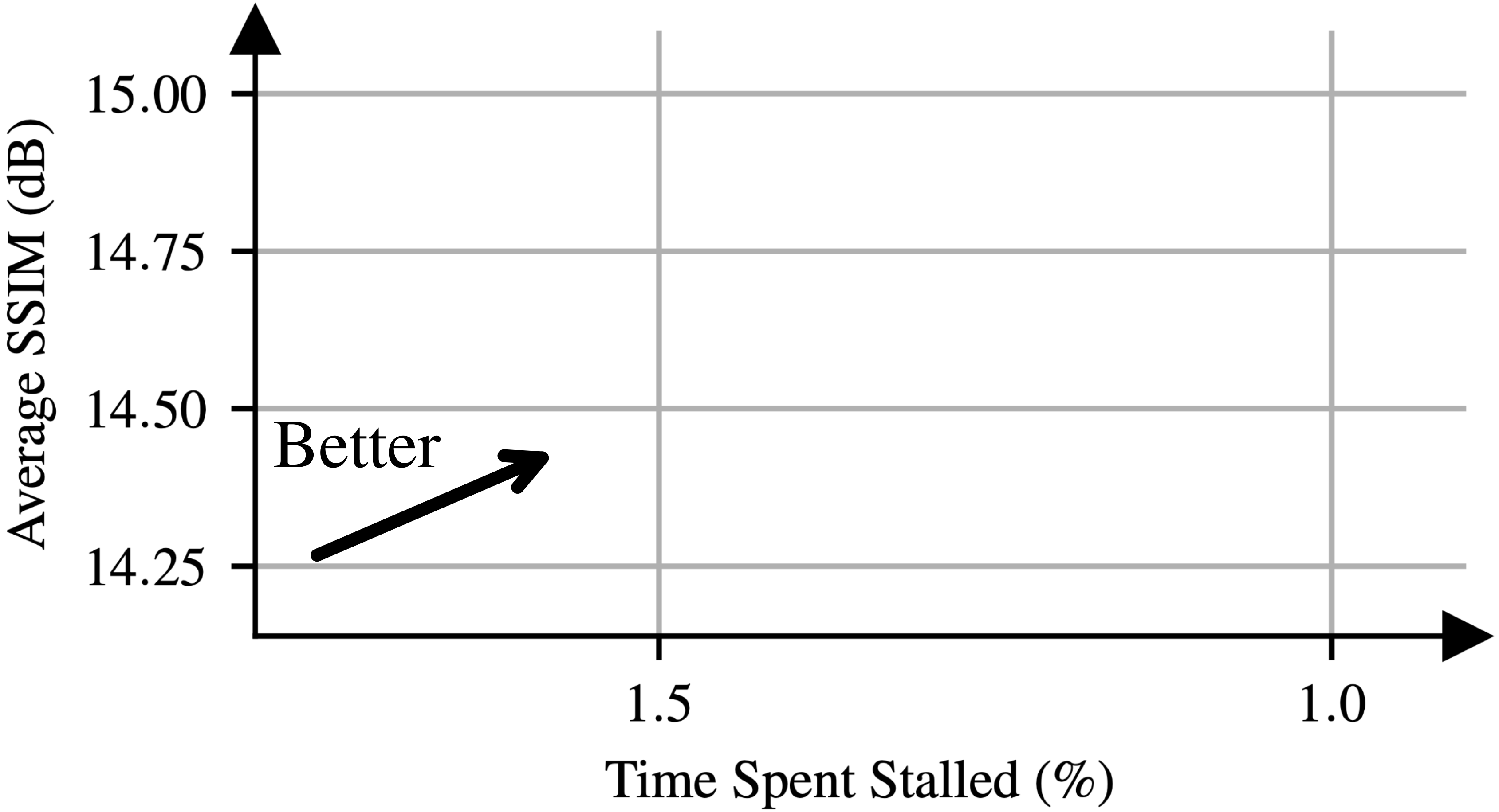


Case study: CausalSim in the wild

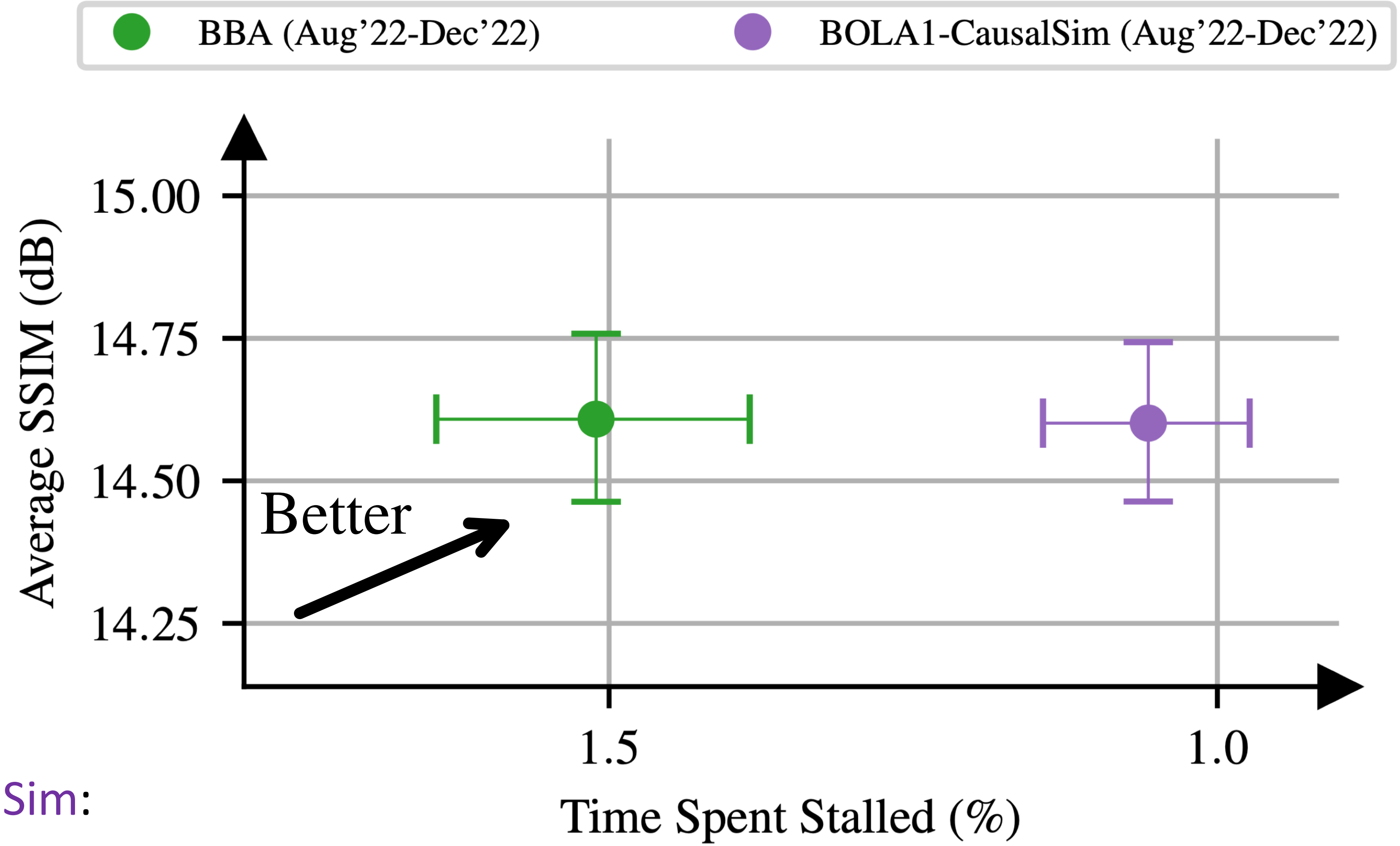


Which one?

Case study: CausalSim in the wild



Case study: CausalSim in the wild



BOLA1-CausalSim:

- Deployment: 1.4x less stalling than BBA

Contributions

❑ Identified *Exogenous Trace Assumption* as a key source of bias in trace-driven simulation.

❑ Proposed ***CausalSim*** for eliminating bias, by modeling the effect of interventions on the trace.

❑ Demonstrated ***CausalSim***'s impact by a real-world ABR algorithm design and deployment.

🔗 **causalsim**.csail.mit.edu