Model Simulation Using Offline Observations with Low-Rank Model

Devavrat Shah

EECS, IDSS/SDSC, LIDS, ORC

Massachusetts Institute of Technology

Agenda

Setup

An Example: Video Bit Rate Adaptation

Causal Sim: A System Design

Findings

Discussion

Setup

Setup: Markov Decision Process

We consider an infinite-horizon discounted MDP $(\mathcal{S}, \mathcal{A}, p, R, \gamma)$

- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $p(s'|s,a)$: transition kernel
- $R(s,a)$: reward
- $\gamma \in (0,1)$: discount factor

Value function for policy $\pi : \mathcal{S} \to \mathcal{A}$

$$V^{\pi}(s) = \mathbb{E}_p\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \;\middle|\; s_0 = s\right]$$

Optimal value function

$$V^*(s) = \max_{\pi} V^{\pi}(s), \quad \forall s \in \mathcal{S}$$

achieved by optimal policy π^*

Setup: Markov Decision Process

The optimal Q-function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ satisfies the Bellman equation

$$Q^*(s,a) = R(s,a) + \gamma \mathbb{E}_{s' \in p(\cdot|s,a)}\left[V^*(s')\right], \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$$

achieved by optimal policy π^* is given by

$$\pi^*(s) \in \arg\max_{a \in \mathcal{A}} Q^*(s,a)$$

Therefore, quantity of interest is Q^*

Precisely learn $\widehat{Q}$ so that $\|\widehat{Q} - Q^*\|_\infty$ is small

Question of interest

Typical learning question

Develop a method for learning $\widehat{Q}$ so that $\|\widehat{Q} - Q^*\|_\infty < \varepsilon$

in a sample-efficient manner through access to simulate the underlying model

However

Data available from prior (biased) observations, no access to "simulator"

With single trajectory per related, heterogenous environment

Some more setup, refining question

In addition to $\mathcal{S} = [0,1]^{d_1}$ and $\mathcal{A} = [0,1]^{d_2}$

We might have "heterogenous" environment

Environment of "agent" or "trajectory" is parameterized by $\theta \in \Theta$

For simplicity: $\Theta = [0,1]^{d_3}$

Model or transition kernel: agent parameterized by $\theta \in \Theta$

$$d\mathbb{P}\big(s' \mid s, a; \theta\big)$$

$$\mathbb{E}\big[s' \mid s, a; \theta\big] = f(s, a, \theta)$$

nice function

latent param
(captures confounding)

Question of interest

Typical learning question

Develop a method for learning $\widehat{Q}$ so that $\|\widehat{Q} - Q^*\|_\infty < \varepsilon$

in a sample-efficient manner through access to simulate the underlying model

However

Data available from prior (biased) observations, no access to "simulator"

With single trajectory per related, heterogenous environment

Tasks of interest

Learn Q with minimal number of samples from a "simulator"

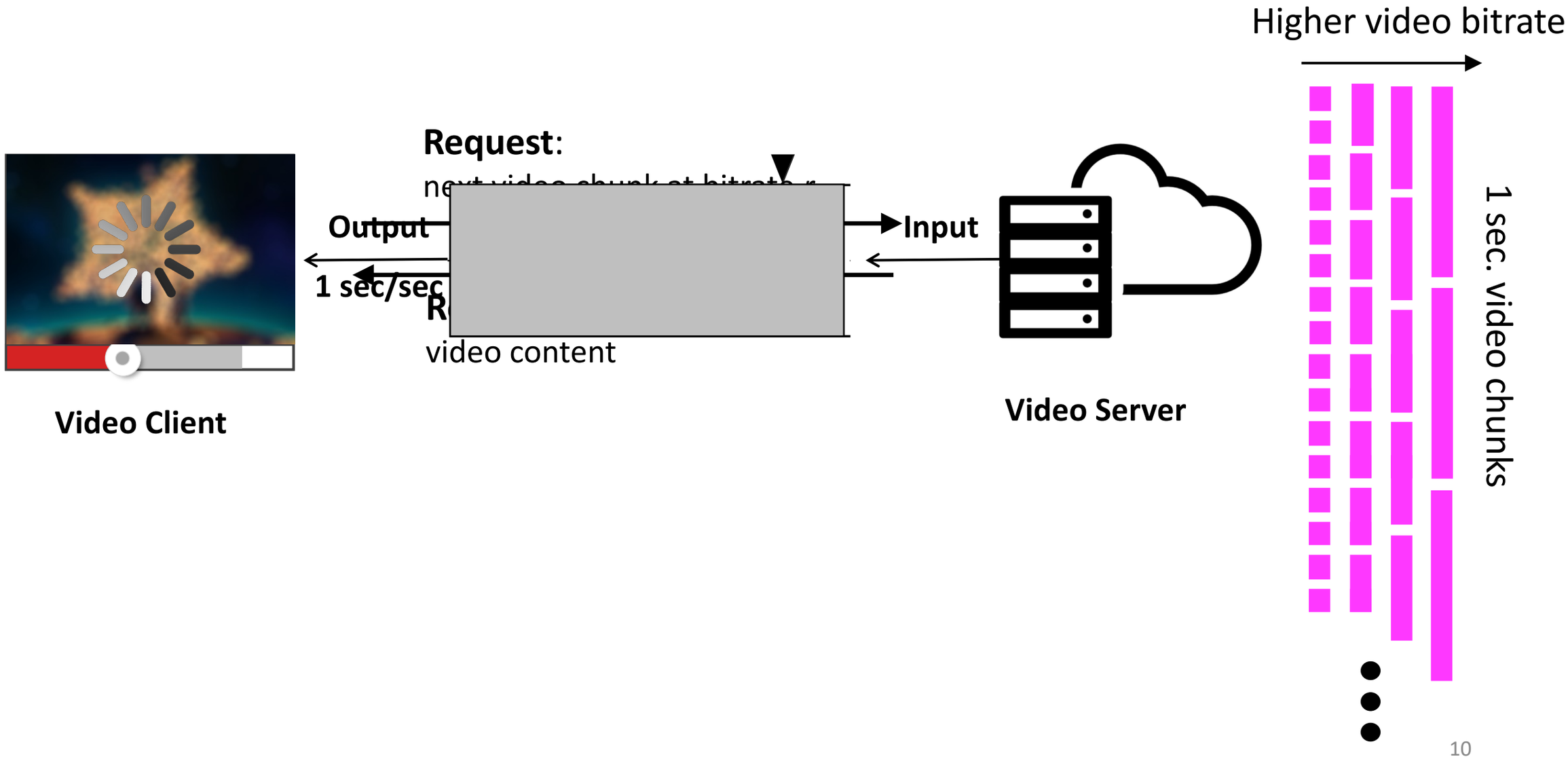
Learn "simulator" from prior (biased) observational data accurately

The talk

Factorization can help learn "simulator" in from observational data

An Example: Video Bit Rate Adaptation

Adaptive Bit Rate (ABR) Video Streaming



cf. see [Mao, Netravali, Alizadeh ’17]

Task of Interest

How does a new ABR protocol perform

compared to existing ones?

Two “Standard” Protocols: BBA and BOLA1

which is better?

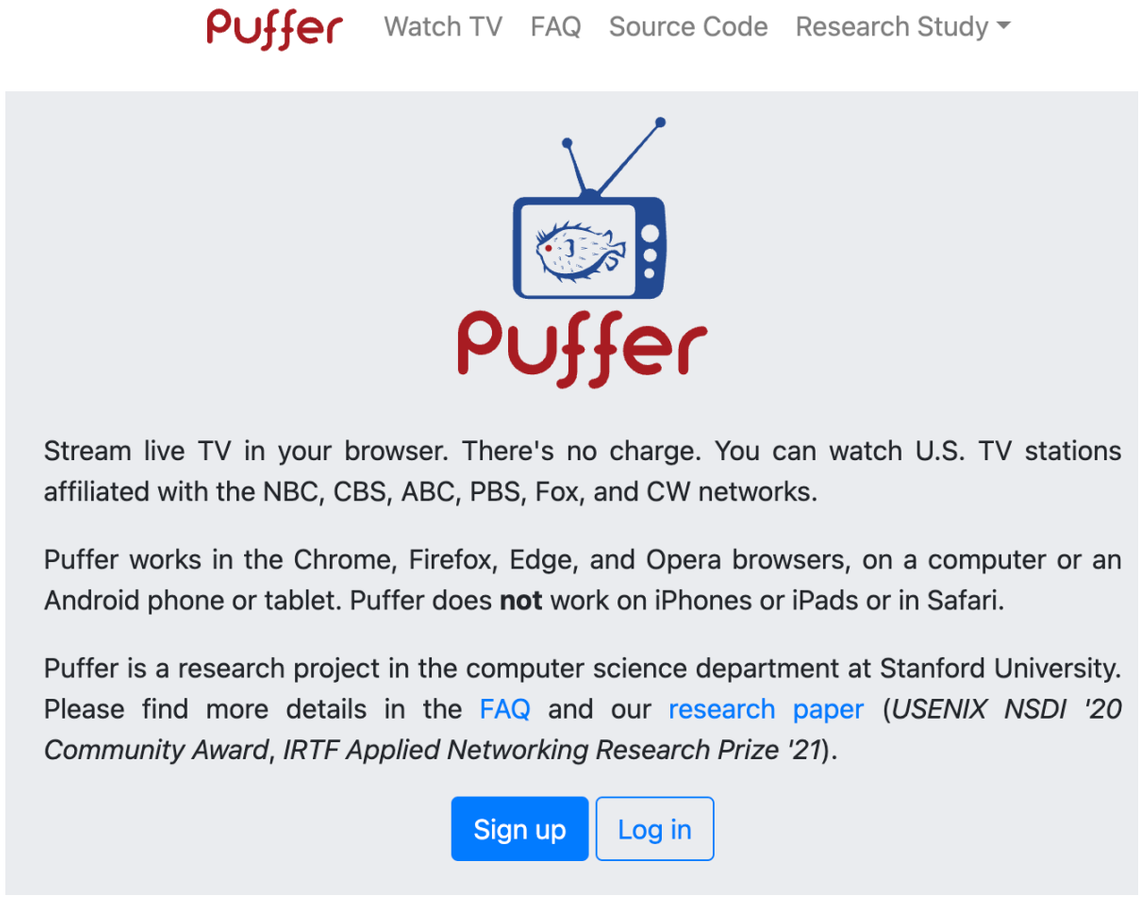
Approach #1: Randomized Experiment

Puffer Randomized Trial

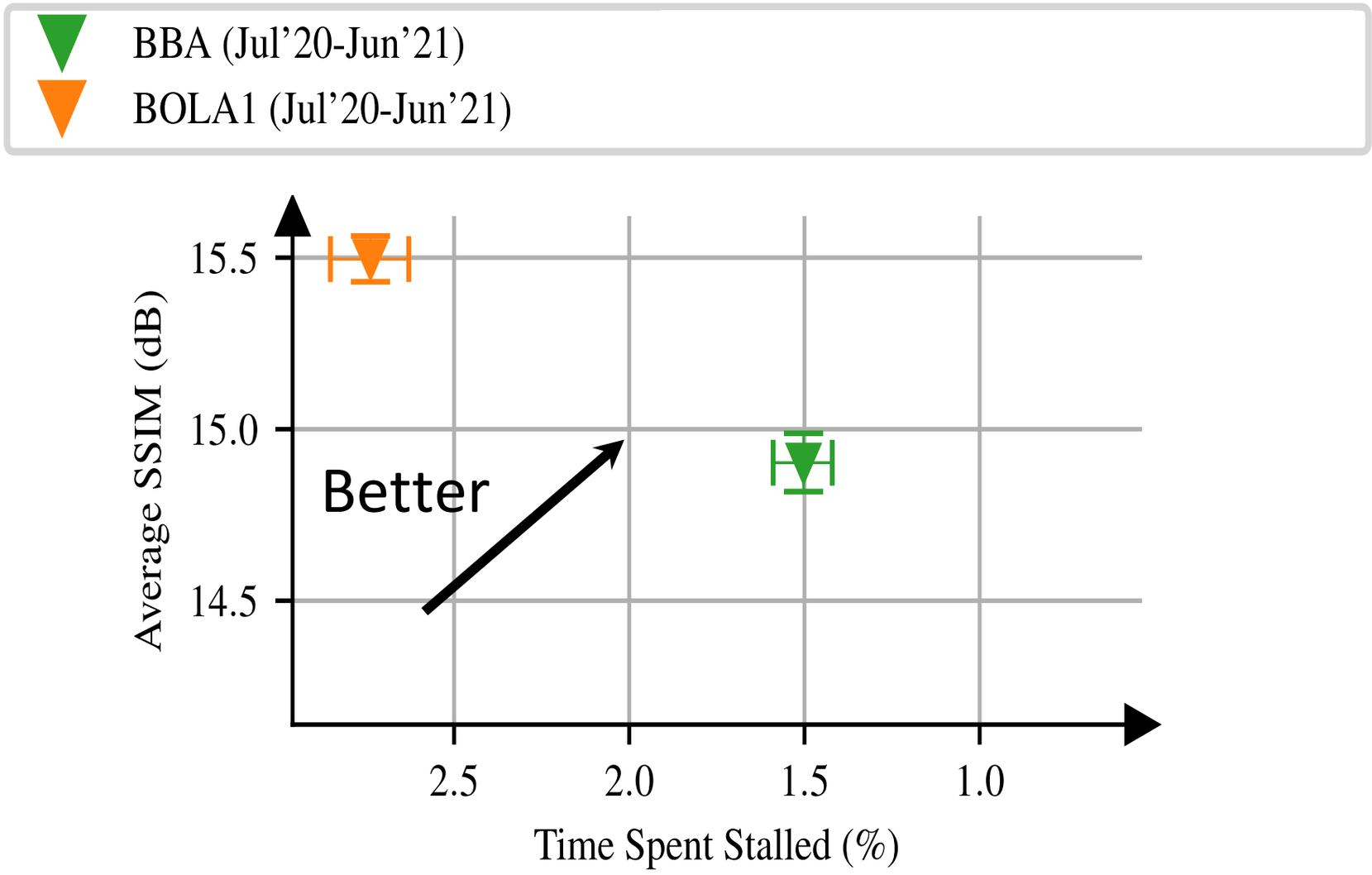
- 5 different Bit Rate Adaptation (ABR) protocols

- 62M downloaded chunks over 32k sessions
  - roughly 1k chunks / session
  - more than 4 yrs worth of video

- Measurements
  - buffer occupancy, download time, chosen chunk size, …



Randomized Experiment: Puffer



No one dominates other
(for given choice of protocol parameters)

Can we optimize choice of parameters? Too many experiments!

Approach #2: Simulation Using Observed Traces

Trace 1 implemented ABR 1

What if

Trace 1 implemented ABR 2?
…
Trace 1 implemented ABR 5?

Trace 100 implemented ABR 2

What if

Trace 100 implemented ABR 1?
…
Trace 100 implemented ABR 5?

Approach #2: Simulation Using Observed Traces

Chunk #	1	2	3		Algorithm
Bitrate	360p	480p	480p		BBA
Achieved Throughput	1Mbps	0.8Mbps	1.2Mbps	• • •	(“source”)
Playback Buffer	5s	3s	7s		

Chunk #	1	2	3		Algorithm
Bitrate	**720p**	**?**	**?**		BOLA1
Achieved Throughput	**?**	**?**	**?**	• • •	(“target”)
Playback Buffer	**?**	**?**	**?**		

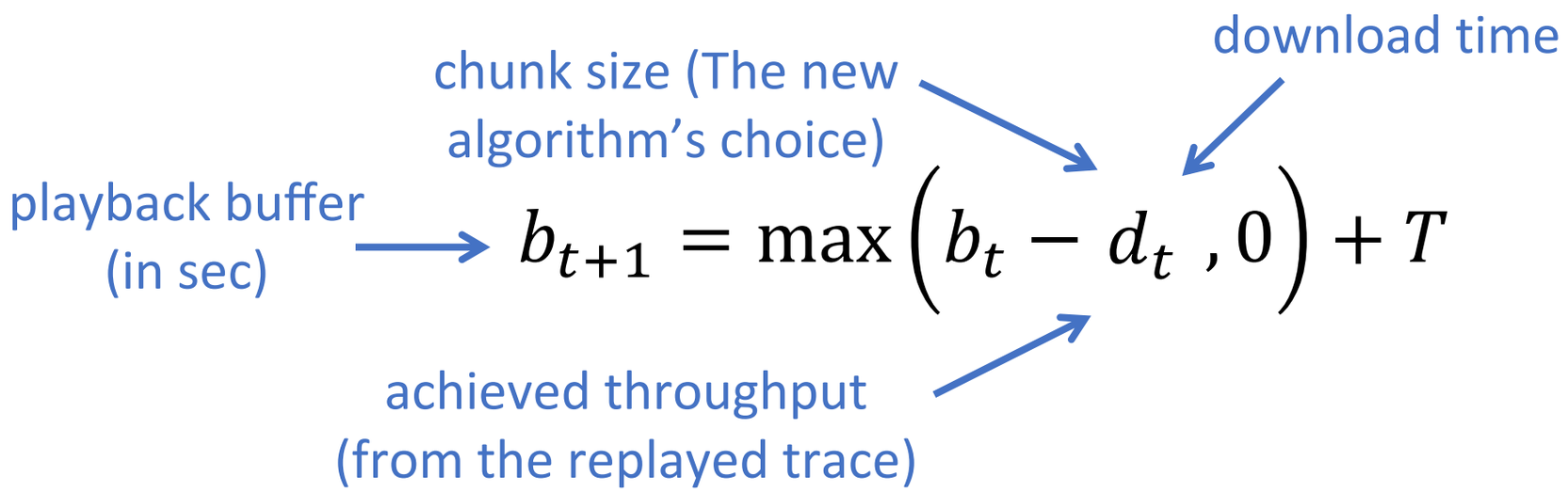
ABR Dynamics

- Adaptive bitrate (ABR) protocol
  - decide the rate at which to "stream" the video

- Dynamics: at decision step t
  - video buffer (playback time): b_t
  - ABR chooses rate
    - which translates into "chunk" of video of size: a_t
  - realized network throughput (or transmission rate): m_t
  - playback time (buffer) corresponding to a decision step: T

$$b_{t+1} = \min(b_t - a_t/m_t, 0) + T$$

Naïve Simulator: ExpertSim

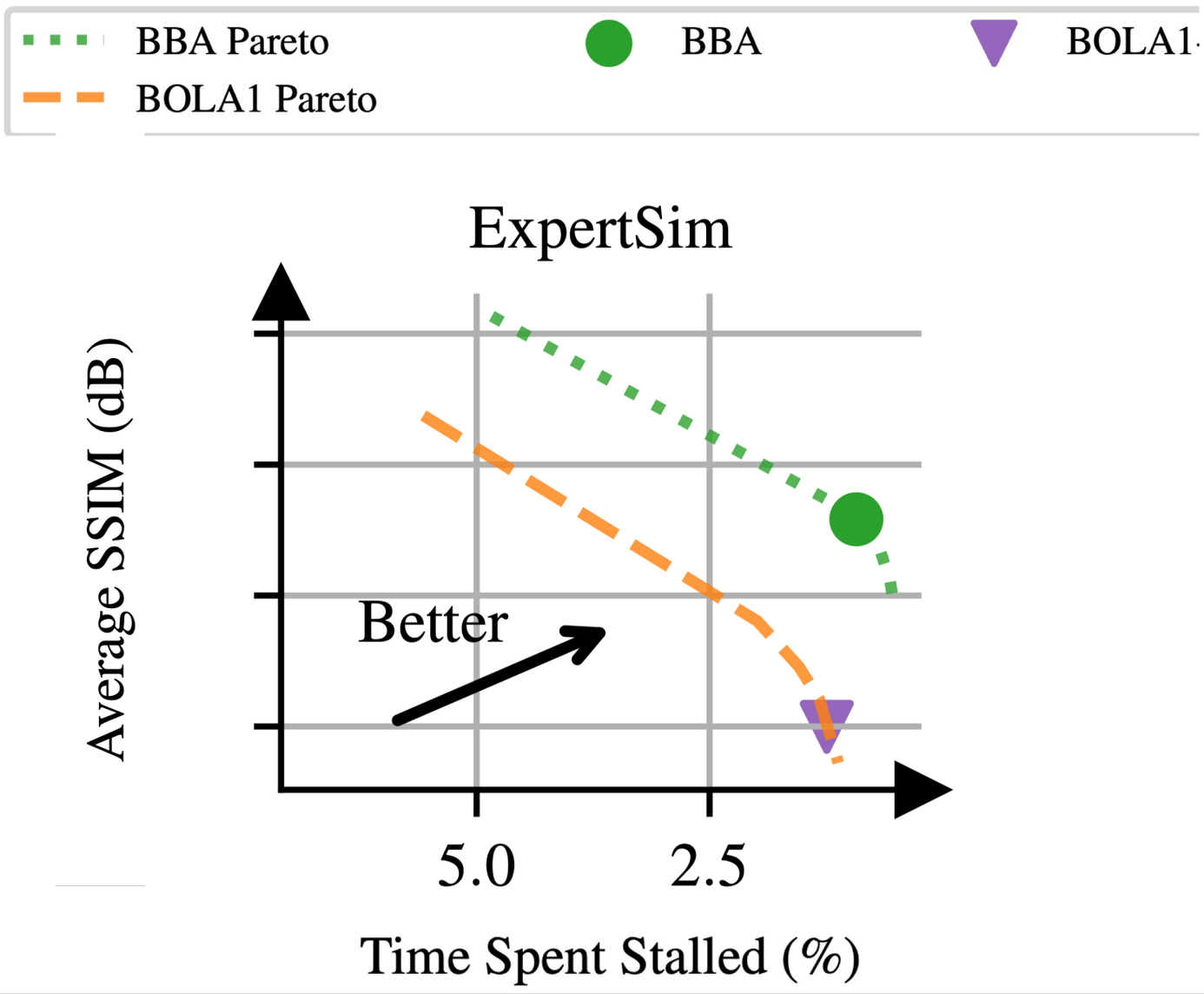


Key assumption

achieved throughput is NOT impacted by choice of algorithm

[Yin et al. SIGCOMM'15][Mao et al. SIGCOMM'17]

Naïve Simulator: ExpertSim

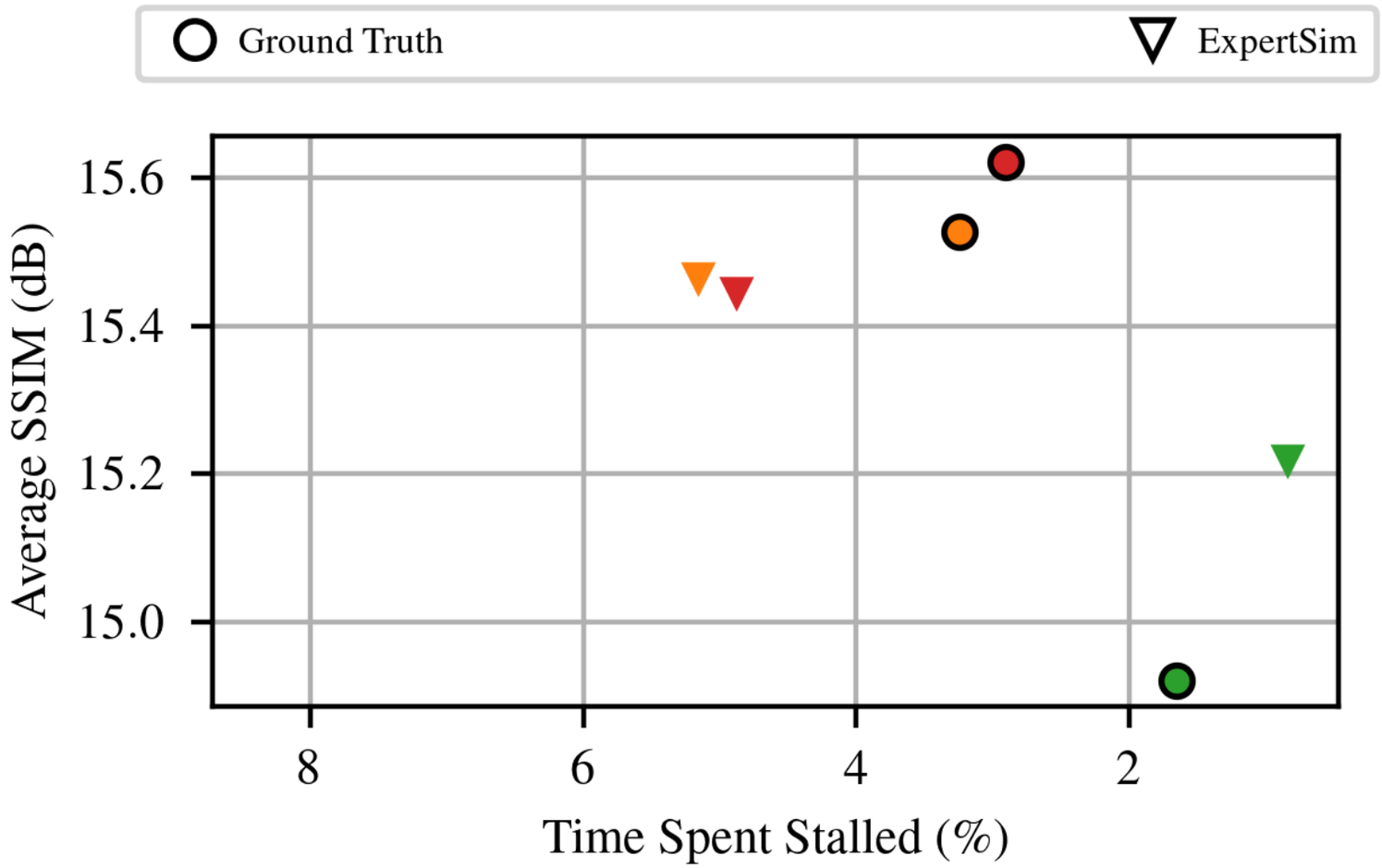


BBA seem to dominate BOLA1

Should we trust this?

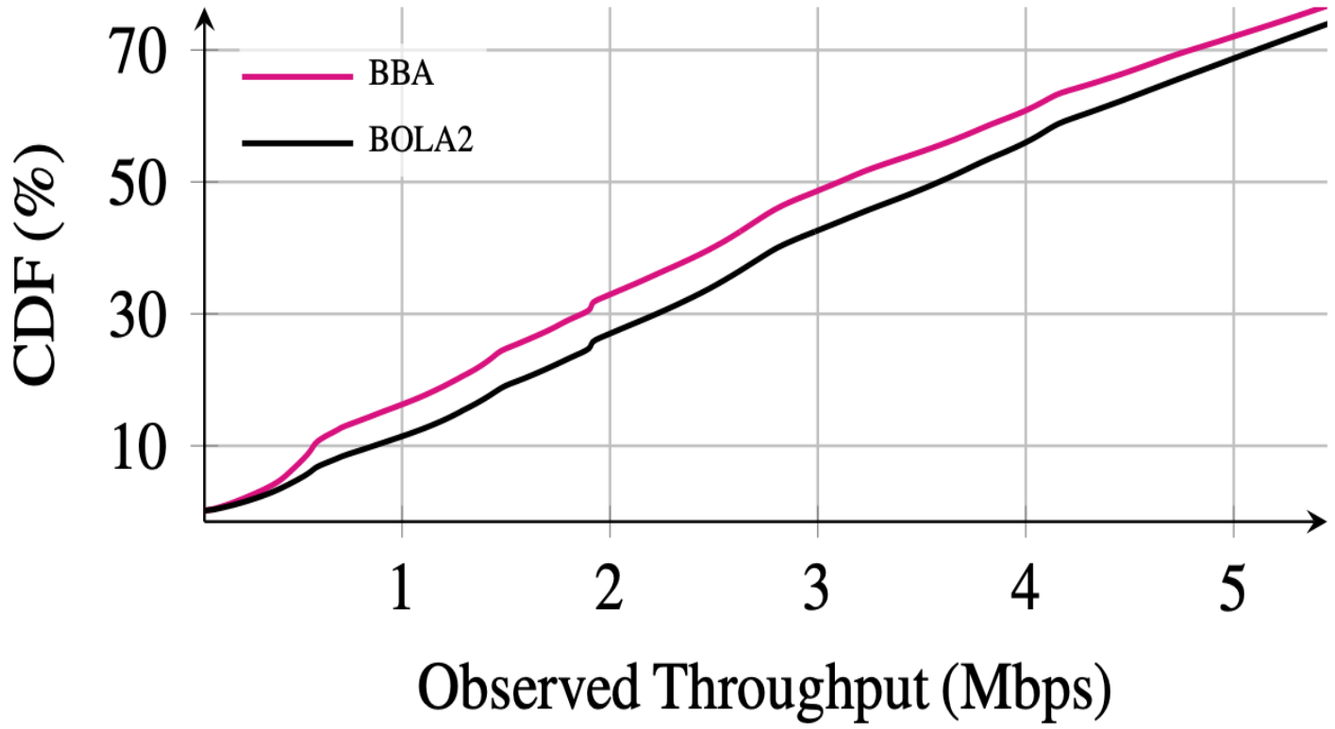
Naïve Simulator: ExpertSim

Puffer Dataset (w 5 ABR protocols)
Simulation target: BBA, BOLA1, BOLA2



accuracy is *low*

ExpertSim: Verifying Assumption



Observed Throughput Depends on Choice of Protocol

Under Randomized experimentation

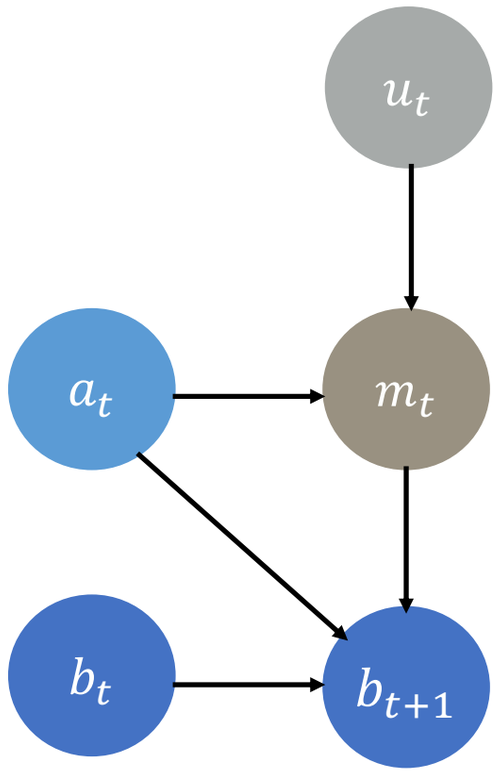
That is, assumption is not true $\rightarrow$ Need a *better* simulator

Causal Sim

ABR Dynamics

- Dynamics
  - system state at time t
    - *latent exogenous state:* $\mathrm{u_t}$
      - *latent network condition*
    - *observed mediation state:* $\mathrm{m_t}$
      - *observed throughput*
    - *observed measurement:* $\mathrm{o_t}$
      - *observed buffer size*
  - *action at time t:* $\mathrm{a_t}$
    - *chosen bitrate (or chunk size)*

$$b_{t+1} = \min(b_t - a_t/m_t, 0) + T$$



- *system evolves as per*

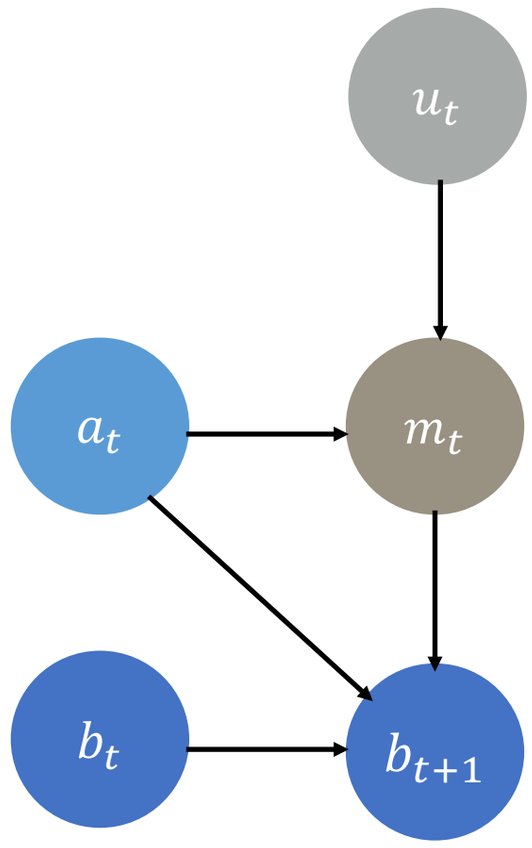
$$m_t = F(a_t, u_t) \quad b_{t+1} = G(b_t, m_t, a_t)$$

ABR Dynamics

- Question
  - N trajectories collected under K protocols (in RCT)

  - *Trajectory* $i \in [N]$:
    - H_i *be the length*
    - *observe* (m_t^i, a_t^i, b_t^i): $t \in [H_i]$
  - *latent* u_t^i: $t \in [H_i]$

  - *simulated actions* $\tilde{a}_t^i$: $t \in [H_i]$
  - *estimate* $\tilde{b}_t^i$: $t \in [H_i]$ *where*



$$\tilde{m}_t = F(\tilde{a}_t, u_t) \qquad \tilde{b}_{t+1} = G(\tilde{b}_t, \tilde{m}_t, \tilde{a}_t)$$

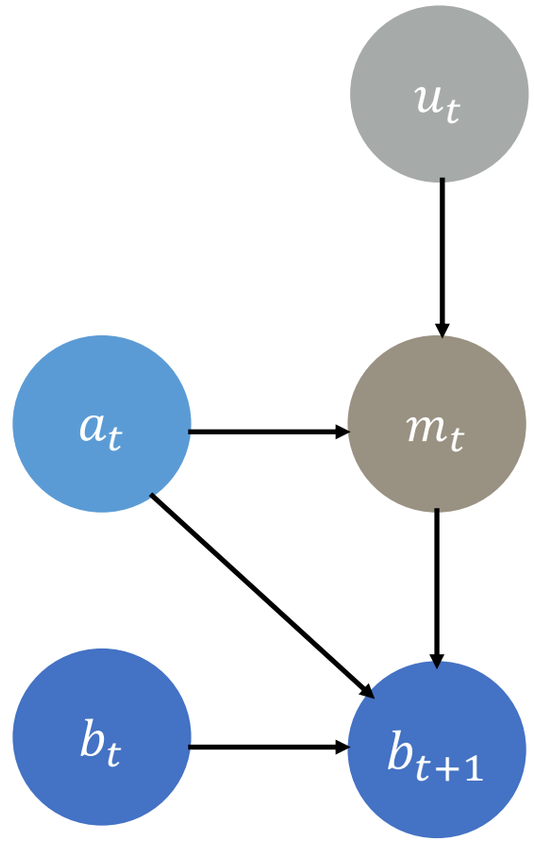
Towards a Solution

- Given system dynamics, a path towards a solution

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

$$b_{t+1} = G(b_t, m_t, a_t)$$

  - Learn G
    - using observed $(b_{t+1}; (\, b_t, m_t, a_t))$
    - across trajectories



  - *For trajectory* $i \in [N]$
    - *given simulated actions* $\tilde{a}_t^i : t \in [H_i]$
    - *if we could estimate* $\tilde{m}_t^i : t \in [H_i]$ *then we can estimate outcome of interest*

  - *therefore, key challenge: how to estimate*

$$\tilde{m}_t^i : t \in [H_i]$$

Estimating "Mediator"

- The mediator has functional form

$$m = F(a, u)$$

- Let
  - F is a "nice" function
  - And, actions, Confounders belong to "nice" domains
  - Then

$$m = F(a, u) \approx \sum_{\ell=1}^{r} \alpha_\ell(a)\beta_\ell(u)$$

Because

Singular Value Decomposition:

- $\mathrm{h} \in L^2([0,1]^2)$
- h is L-Lipschitz
- Then there exists
  - $\sigma_i, i \in N$ such that $\sum_i {\sigma_i}^2 < \infty$
  - orthonormal functions $\alpha_i \in L^2([0,1])$
  - orthonormal functions $\beta_i \in L^2([0,1])$
  - such that

$$\mathrm{h} = \sum_i \sigma_i \, \alpha_i \otimes \beta_i \qquad \text{i.e.} \qquad \mathrm{h}(x,y) = \sum_i \sigma_i \alpha_i(x) \beta_i(y)$$

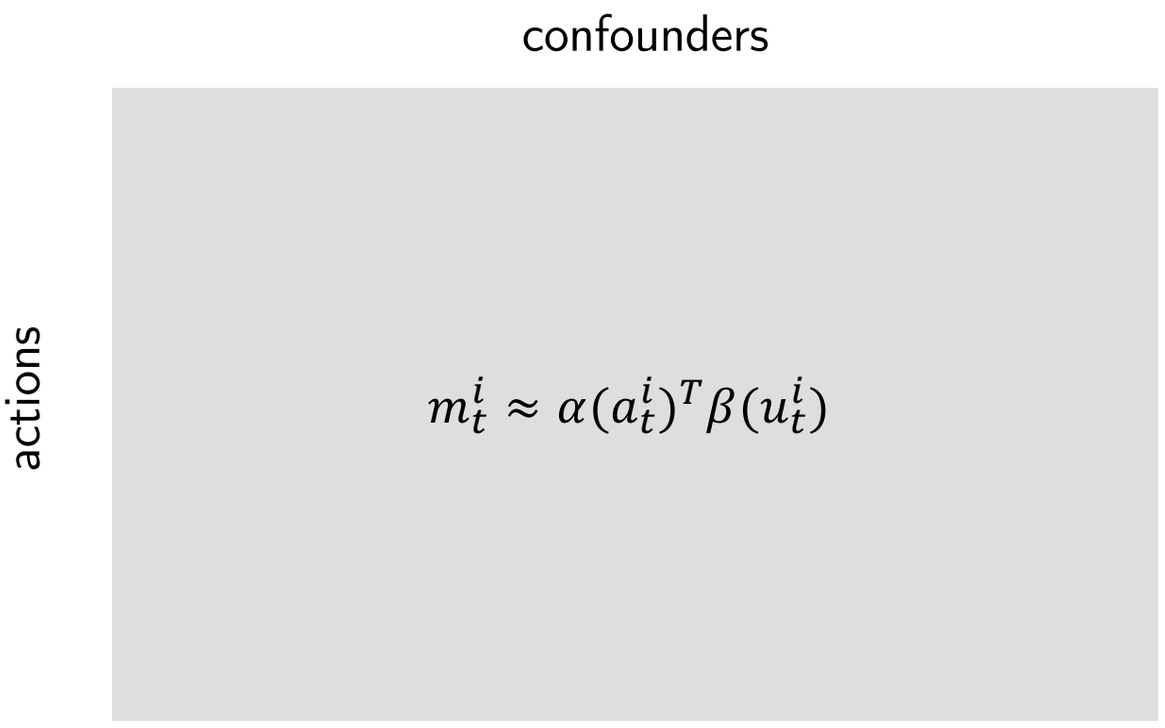
- That is, for any $\eta > 0$ there is $r = r(\eta)$ s.t.

$$|\mathrm{h} - \sum_{i \le r} \sigma_i \alpha_i \otimes \beta_i \,|^2 \; \le \eta$$

cf. see [Shah, D., Song, D., Xu, Z. and Yang, Y., 2020]

Back To Estimating Mediators: Consider A Matrix

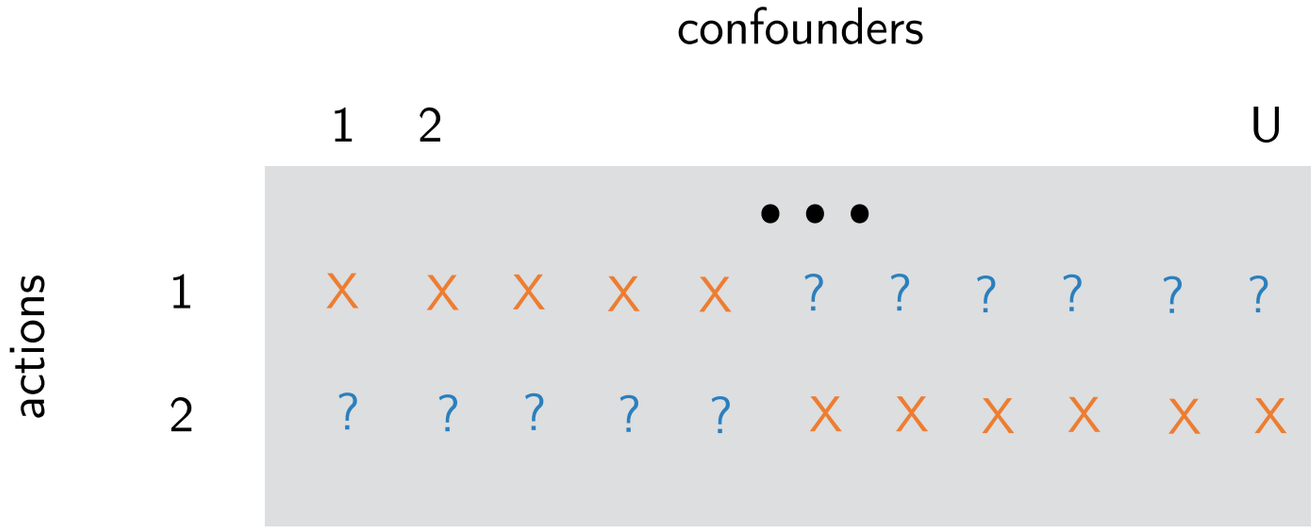
- Consider the matrix of mediators



- It's a low-rank matrix
  - but *extremely* sparse: one entry per column!

Back To Estimating Mediators: Consider A Matrix

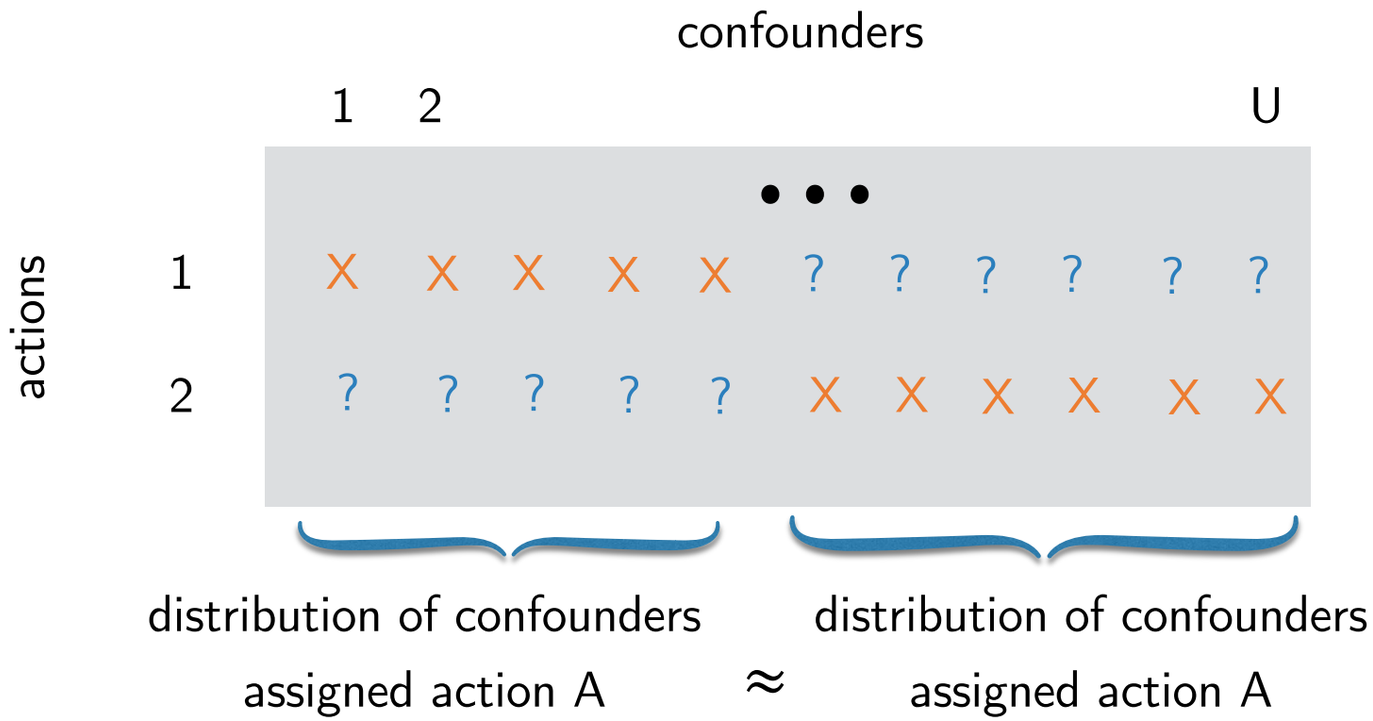
- Consider the matrix of mediators: two actions, rank 1



- Entry in row i, column j: $m(i,j) = \alpha_i \beta_j$
  - But difficult to estimate α_i, β_j due to one observation per column

Estimating Mediators: RCT to Rescue

- Consider the matrix of mediators: two actions, rank 1: $m(i,j) = \alpha_i \beta_j$



- This suggests:

$$\frac{\alpha_1}{\alpha_2} \approx \frac{\text{average of obs in row } 1}{\text{average of obs in row } 2}$$

Estimating Mediators: RCT to Rescue

- More generally, it is a three order tensor
  - Actions, Confounders (trace) and Measurements (of mediator)

$$d\mathbb{P}\left(s' \mid s, a; \theta\right)$$

$$\mathbb{E}\left[s' \mid s, a; \theta\right] = f(s, a, \theta)$$

$$\approx \sum_{\ell=1}^{r} u_\ell(s) v_\ell(a) w_\ell(\theta)$$

max-norm error $< \delta$

number of factors r $= O\left(\delta^{2d}\right), \quad d = \max(d_1, d_2, d_3)$

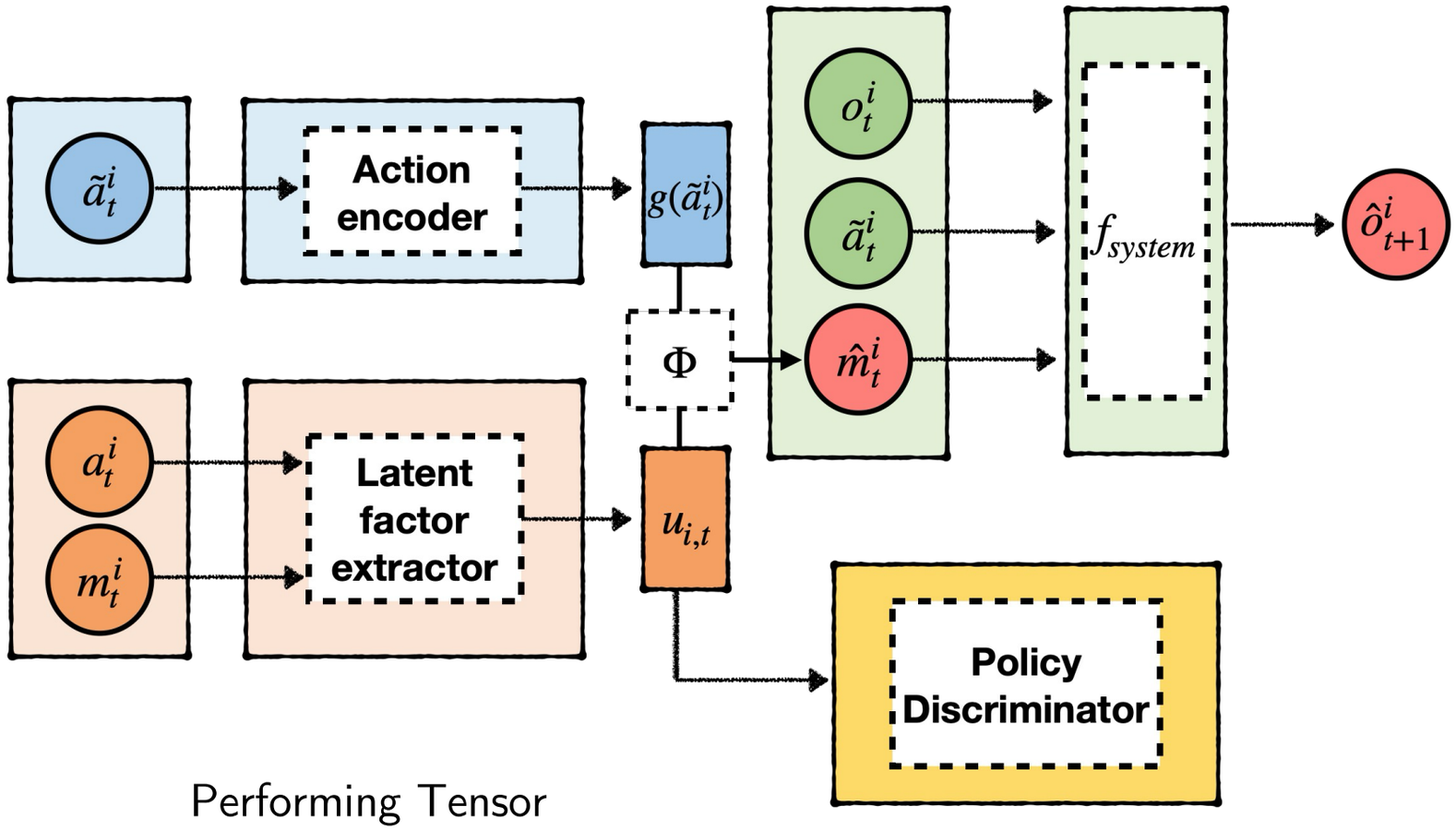
cf. see [Agarwal, A., Alomar, A., Alumootil, V., Shah, D., Shen, D., Xu, Z. and Yang, C., 2021]

Estimating Mediators: RCT to Rescue

- More generally, it is a three order tensor
  - Actions, Confounders and Measurements (of mediator)
  - Under appropriate conditions
    - Low CP-rank Tensor
    - Randomized assignment of actions
    - Measurement dimension $\geq$ CP-rank of Tensor
    - "Coverage" (enables solving certain linear equations)

- The tensor estimation is feasible, even when
  - For each confounder, measurements are observed
    - under only one action

System Implementation

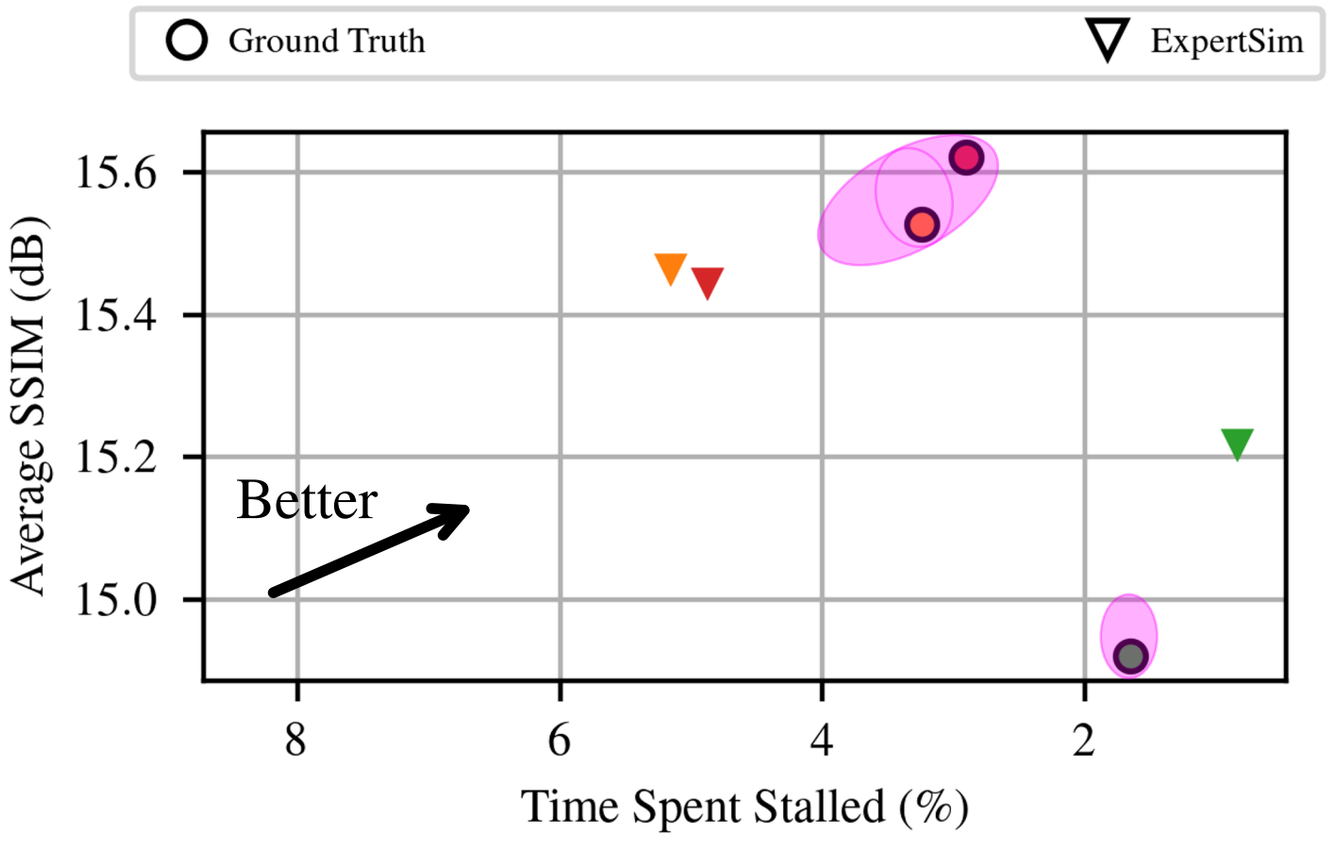


Performing Tensor Estimation

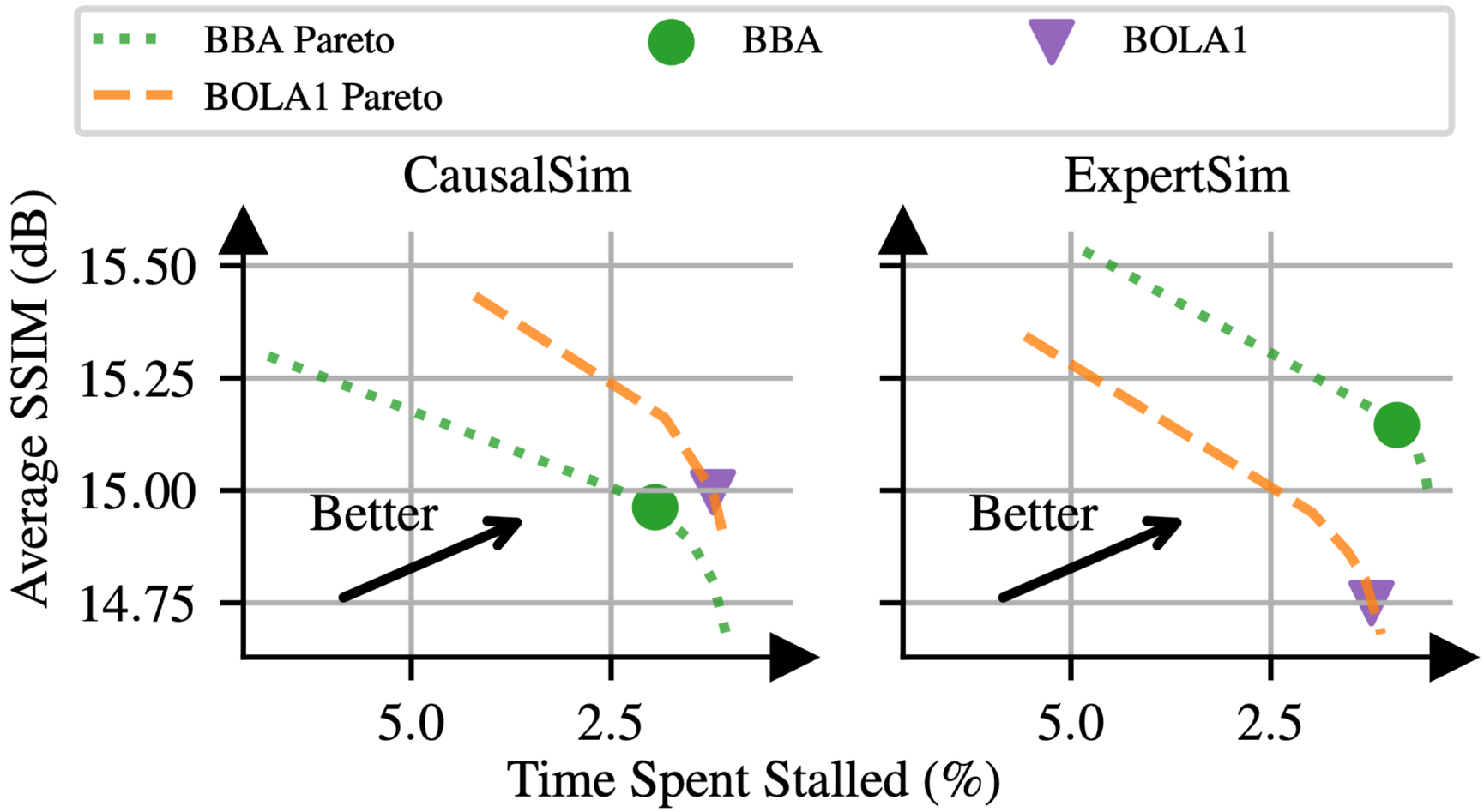
Distributional invariance due to RCT

Findings

CausalSim vs ExpertSim



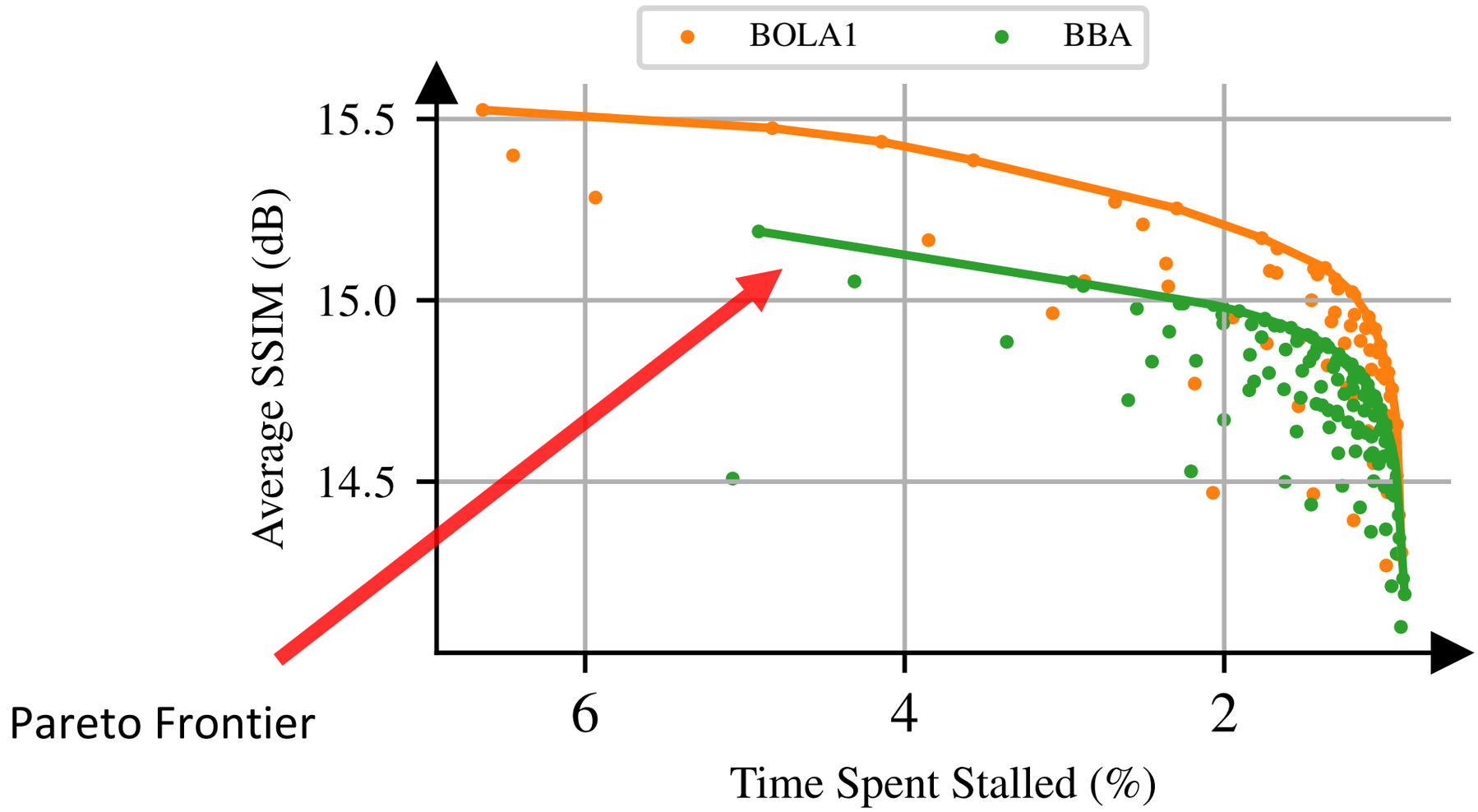
CausalSim vs ExpertSim



ExpertSim: BBA seem to dominate BOLA1 vs. CausalSim: BOLA1 seem to dominate BBA

which one is right?

Case-Study: BBA and BOLA1 in Puffer



ExpertSim: BBA seem to dominate BOLA1 vs. CausalSim: BOLA1 seem to dominate BBA

which one is right? **CausalSim!**

Discussion

Beyond ABR: Load Balancing

- N servers
  - Heterogenous service rate, each with own queue

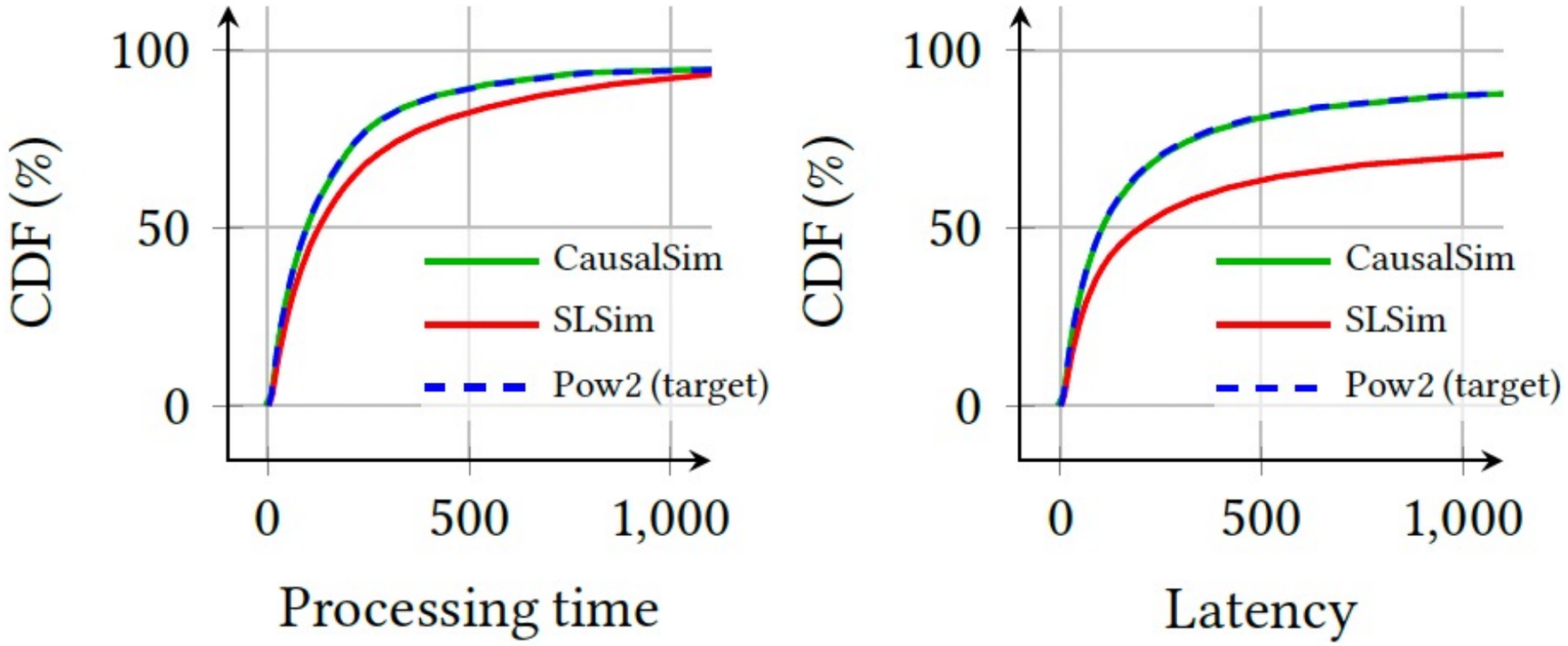
- Arriving jobs need to be served by one of the server
  - Jobs have different *latent* sizes
  - Server $i \in [N]$ serves any job at rate r_i
    - job of size S takes S/r_i time to be served by server i

- Ideal goal
  - Assign incoming job to an available server
  - So as to minimize overall job completion time (many variants)

- Simulation
  - Completion time of a job on all N servers

Beyond ABR: Load Balancing

- Distribution of measurement under CausalSim
  - matches that of "target" policy



SLSim: Supervised Learning (aka Imitation Learning?!)

Formal Guarantees

- Work in progress [Wan, Shah, Wainwright 2026]

- Finite sample guarantees
  - estimating transition kernel
  - across heterogenous, but related trajectories generated under unknown policies
  - with low-rank tensor structure

- In a nutshell
  - matrix or tensor estimation (state, action, trace)
  - with observations sampled as per a Markov chain over space of (state, action)

Formal Guarantees

- Work in progress [Wan, Shah, Wainwright 2026]

- Structure of Q function
  - additionally, if Reward function has low-rank structure
  - then Q function has low-rank structure as well

- In a nutshell
  - low-rank transition kernel and reward function across heterogenous trajectories
  - enables (sample-)efficient learning of transition kernel
  - which in turn can enable (sample-)efficient estimation of Q function cf. [Shah, Song, Xu, Yang 2020]

References

Shah, D., Song, D., Xu, Z. and Yang, Y., 2020. Sample efficient reinforcement learning via low-rank matrix estimation. *Advances in Neural Information Processing Systems*, *33*, pp.12092-12103.

Agarwal, A., Alomar, A., Alumootil, V., Shah, D., Shen, D., Xu, Z. and Yang, C., 2021. Persim: Data-efficient offline reinforcement learning with heterogeneous agents via personalized simulators. Advances in Neural Information Processing Systems, 34, pp.18564-18576.

Alomar, A., Hamadanian, P., Nasr-Esfahany, A., Agarwal, A., Alizadeh, M. and Shah, D., 2023. {CausalSim}: A causal framework for unbiased {Trace-Driven} simulation. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)* (pp. 1115-1147).

Wan, J. Shah, D., and Wainwright, M., 2026. Estimation transition kernel of MDP by pooling information across related heterogenous systems using offline data. *Under Preparation*.

Questions