By: **Sagar Patel**, Sangeetha Abdu Jyothi,
Nina Narodytska

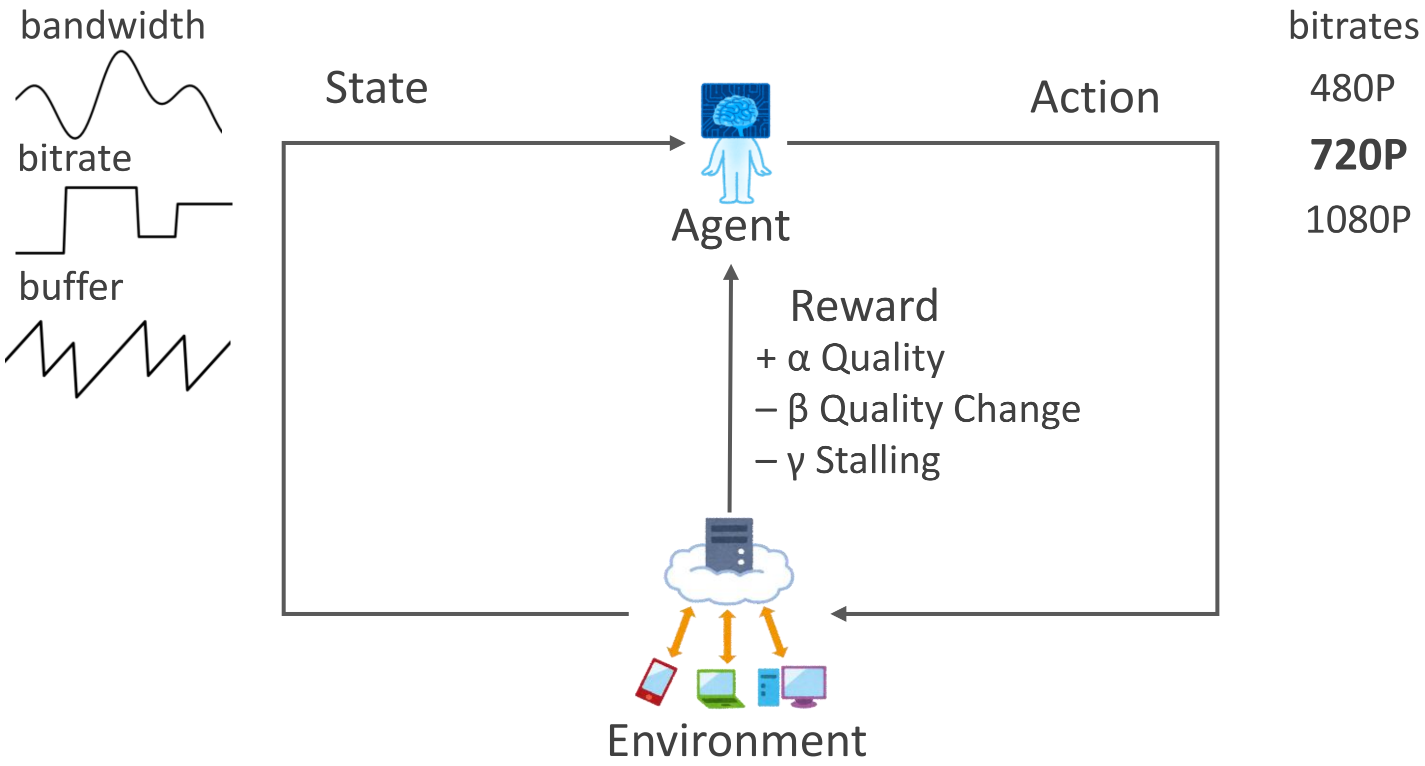
Towards Future-Based Explanations for DRL Network Controllers

University of California, Irvine
vmware

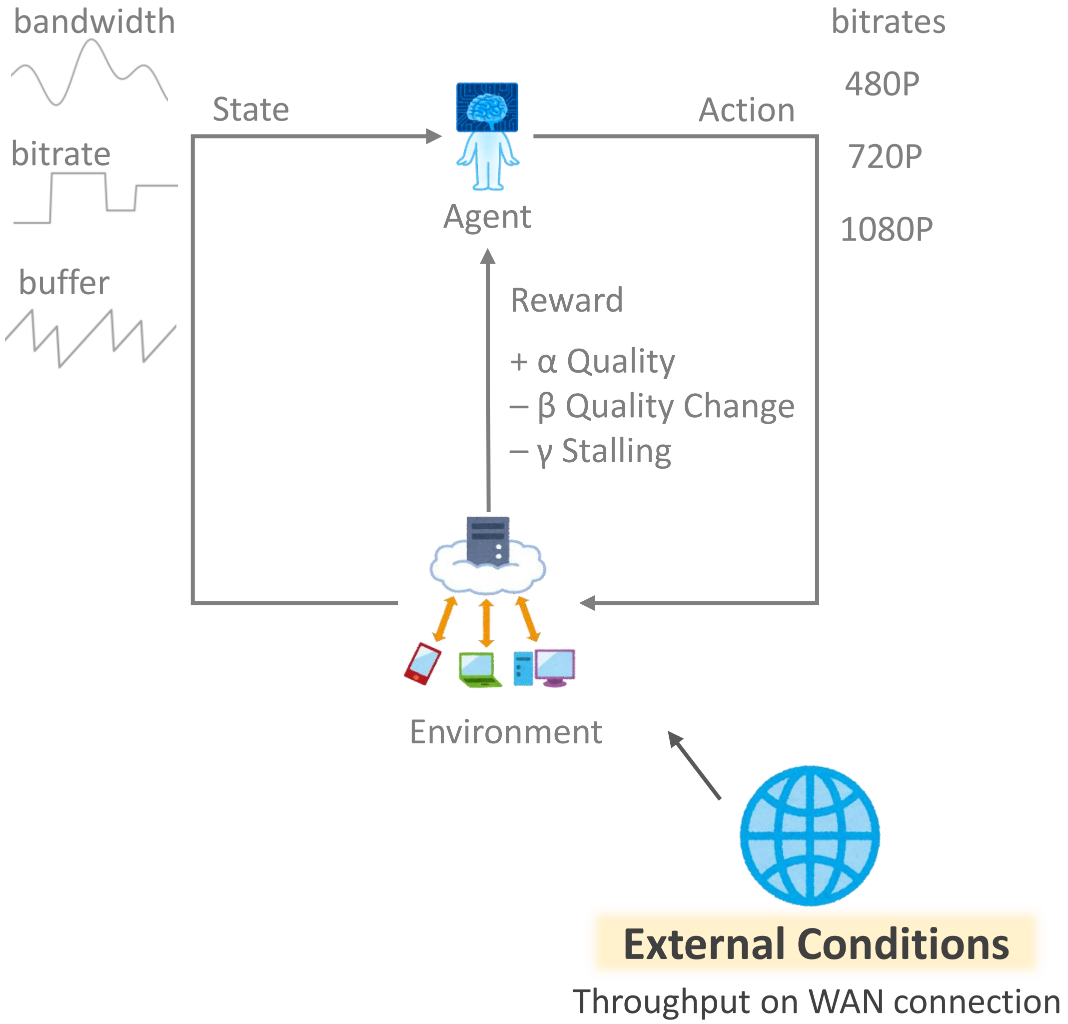
The Need for Explainability

- Deep Reinforcement Learning (DRL) offers high performance in increasingly more domains
- However, DRL lacks wide-scale deployment because operators cannot
  - Understand the blackbox neural network
  - Debug the controller when it misbehaves
  - Adjust to fix problems

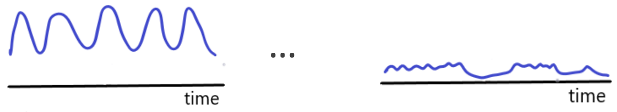
Reinforcement Learning: ABR Perspective



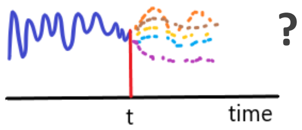
Reinforcement Learning: ABR Perspective



- DRL network controllers are deployed in highly variable network environments
  - E.g., WAN
  - Cannot be faithfully modeled yet
  - During training
    - Replayed as external conditions by a dataset of traces

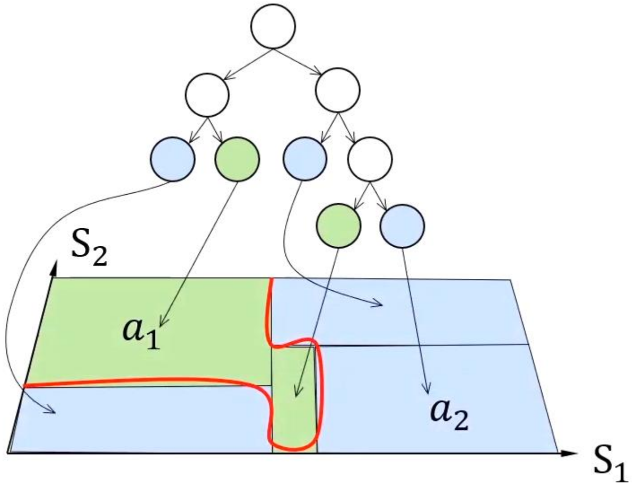


  - At deployment
    - Future is unknown



Prior Explainers for DRL Network Controllers

- Inputs
  - A black-box controller
- Procedure
  1. Sample states from the training environment
  2. Get actions for the states
  3. Train an explainable model using all the state-action pairs
     - Decision Tree [Metis (SIGCOMM ‘20), Trustee (CCS ‘22)]



The Need for Future Based Explanations

Feature Based

+ Identify key features
+ Simplify the model (e.g. decision trees)

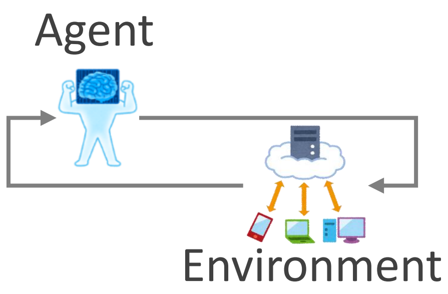
- Do not capture time-dependent nature of DRL
- Do not give insights into impact of actions

Future Based

+ Reveal future impact of actions
+ Allow contrasting actions and states
+ Enable network observability

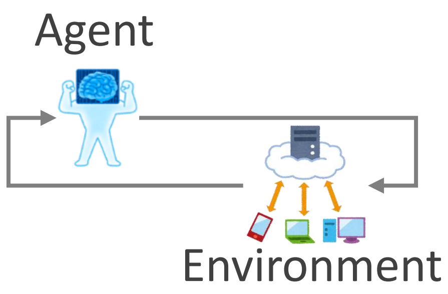
- Cannot simplify the model

Explaining the future: What can we use?



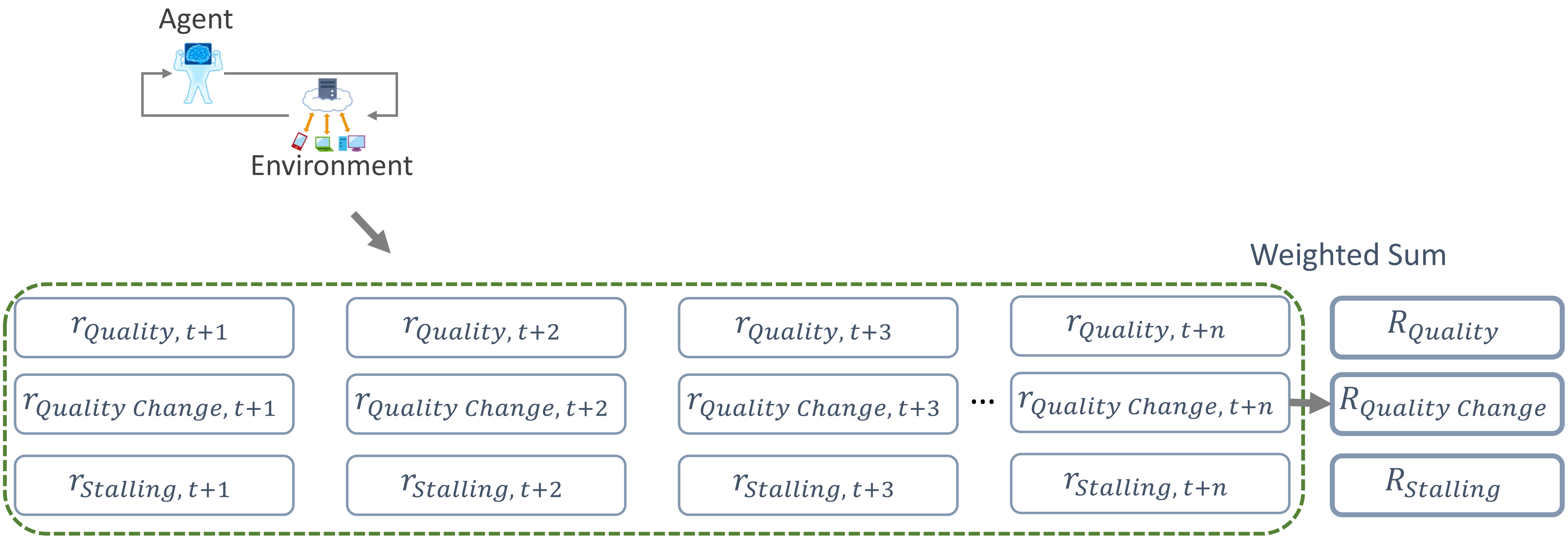
r_{t+1} r_{t+2} r_{t+3} r_{t+n}

Decomposed Future Rewards as Explanations



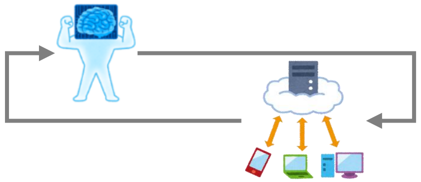
$r_{Quality,\,t+1}$	$r_{Quality,\,t+2}$	$r_{Quality,\,t+3}$		$r_{Quality,\,t+n}$
$r_{Quality\ Change,\,t+1}$	$r_{Quality\ Change,\,t+2}$	$r_{Quality\ Change,\,t+3}$	…	$r_{Quality\ Change,\,t+n}$
$r_{Stalling,\,t+1}$	$r_{Stalling,\,t+2}$	$r_{Stalling,\,t+3}$		$r_{Stalling,\,t+n}$

Decomposed Future Returns as Explanations



Decomposed Future Returns as Explanations

Agent



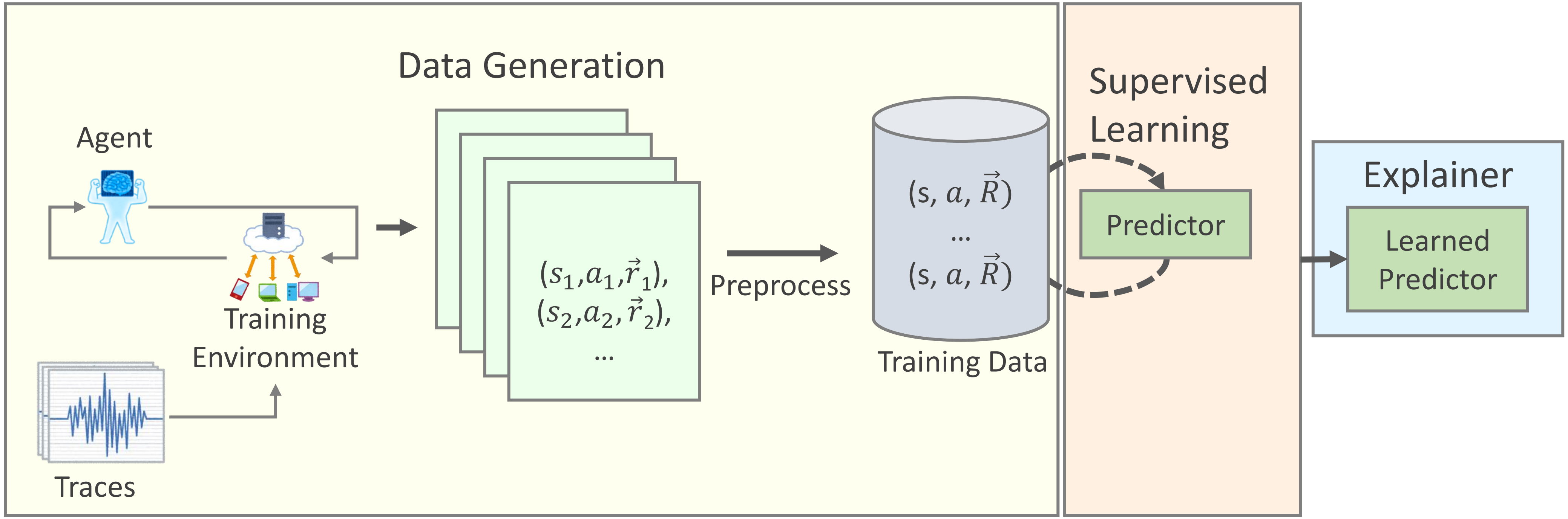
Environment

- We do not yet know how to model Network environments (e.g., WANs)
  - Future is unknown
  - Rewards cannot be calculated

Weighted Sum

$r_{Quality,t+1}$	$r_{Quality,t+2}$	$r_{Quality,t+3}$	$r_{Quality,t+n}$	$R_{Quality}$
$r_{Quality\ Change,t+1}$	$r_{Quality\ Change,t+2}$	$r_{Quality\ Change,t+3}$	$r_{Quality\ Change,t+n}$	$R_{Quality\ Change}$
$r_{Stalling,t+1}$	$r_{Stalling,t+2}$	$r_{Stalling,t+3}$	$r_{Stalling,t+n}$	$R_{Stalling}$

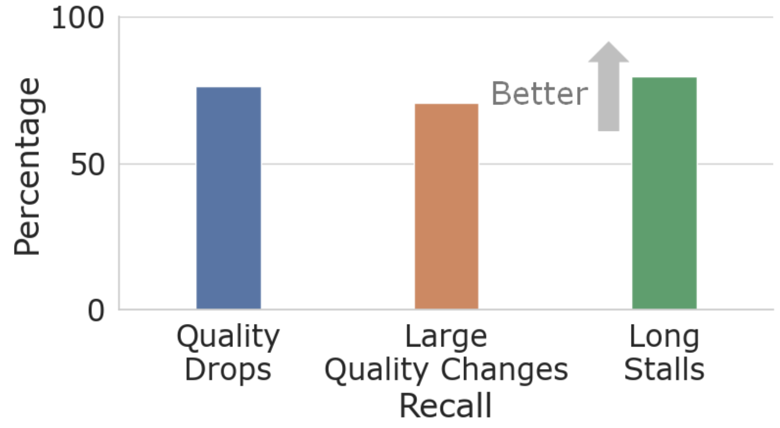
CrystalBox

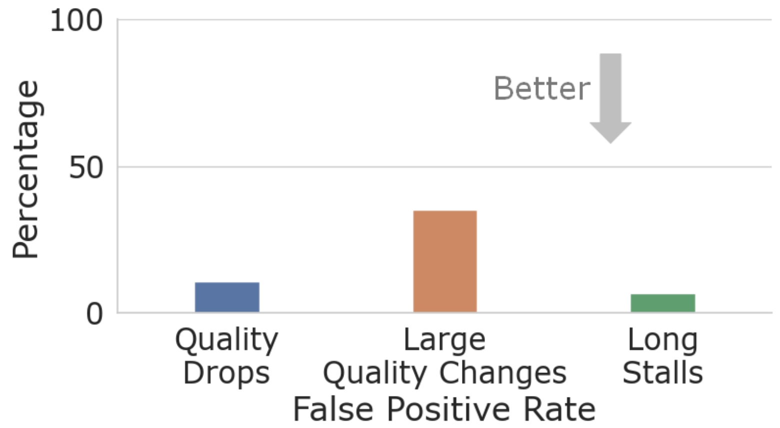


Data Generation
Agent
Training
Environment
Traces
$(s_1, a_1, \vec{r}_1)$,
$(s_2, a_2, \vec{r}_2)$,
...
Preprocess
$(s, a, \vec{R})$
...
$(s, a, \vec{R})$
Training Data
Supervised
Learning
Predictor
Explainer
Learned
Predictor

Use Case 1: Network Observability

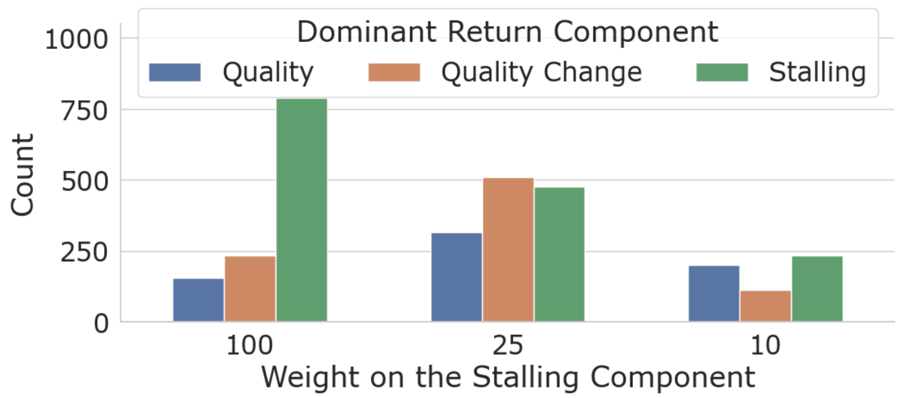
- Controllers experience a wide variety of network conditions
- Alerting for future performance drops before happen is important
- Future-based explanations can help



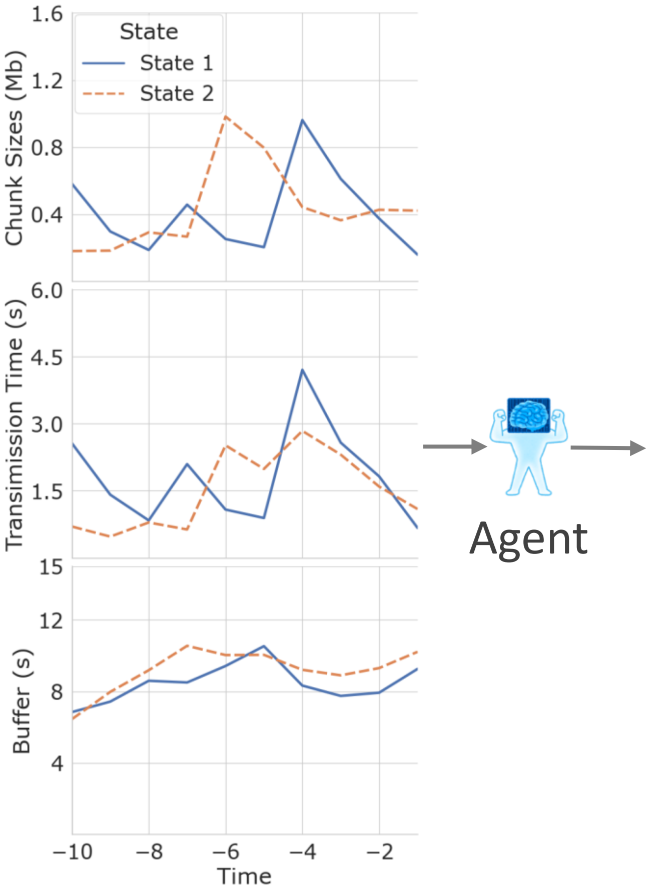


Use Case 2: Guided Reward Design

- Fine tuning reward weights is tedious and resource inefficient
- Future-based explanations can help
- For example, keeping everything else constant, we change stall weight



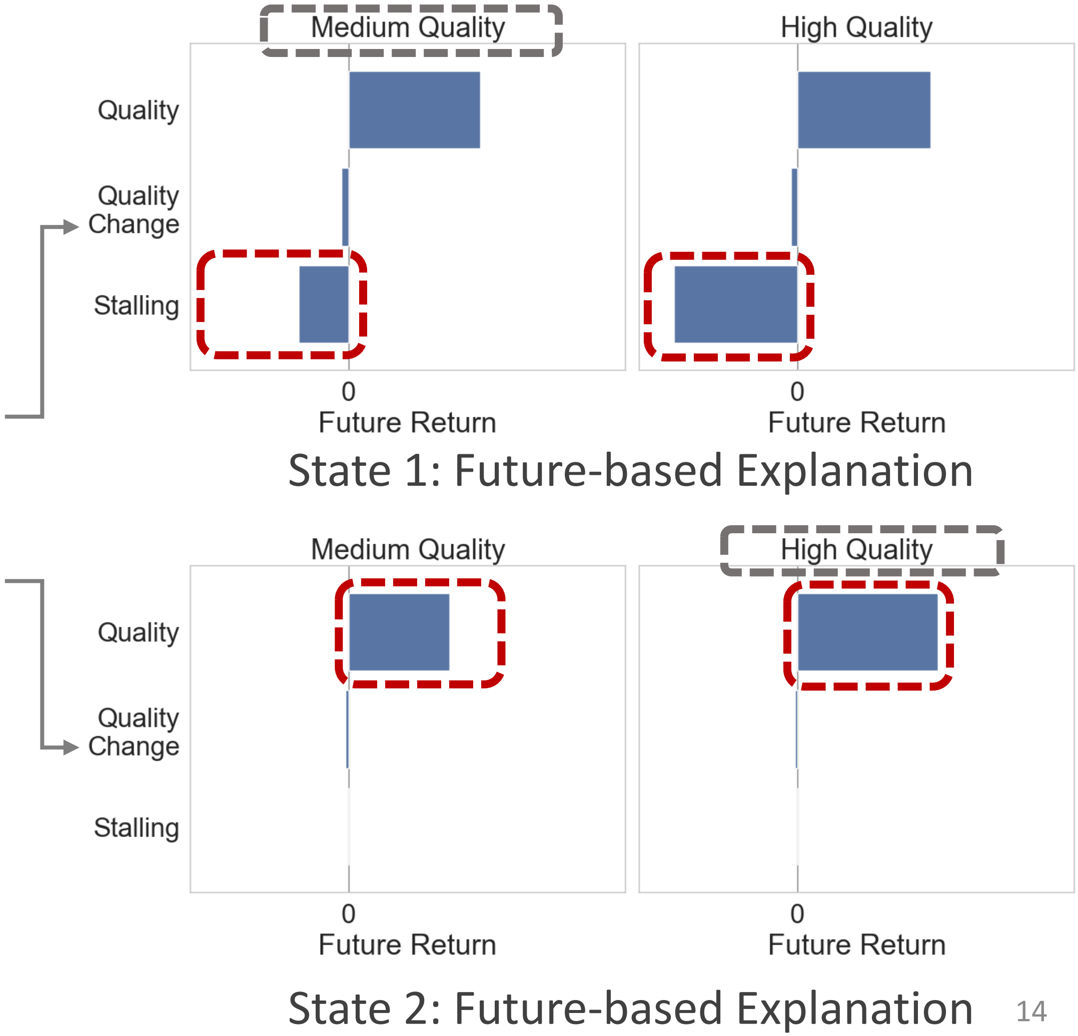
Use Case 3: Cross-State Explainability



State 1: Medium Quality

vs

State 2: High Quality



Open Questions

- Using future states as explanations
- Online safety assurance
- Combining Feature + Future based explainers

Thank You