Rediscovering heuristics: A litmus test for RL in networked systems

Seoyul Oh Yikai Zheng Eric Xiang Gao Deepak Vasisht Francis Y. Yan

1. RL feels like a black box, Can we fix this?

Underperforming RL policy

- Learning got stuck?
- Wrong behavior rewarded?

⋮

Our Intuition
Many networked systems already have good, simple heuristics

- **ABR** → Buffer-Based Algorithm (BBA)
  - Buffer↑ → video bitrate↑
- **CC** → delay feedback (e.g., Vegas)
  - Delay↑ → sending rate↓

IF
A compact RL policy class can represent the heuristic
THEN
Training should reliably rediscover the heuristic

⬇

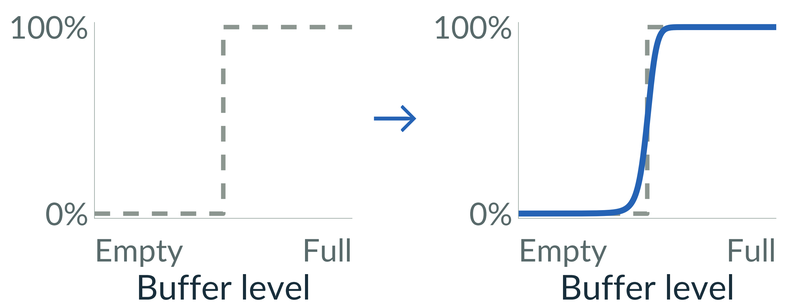
A minimum capability check for RL training

⬇

Our litmus test

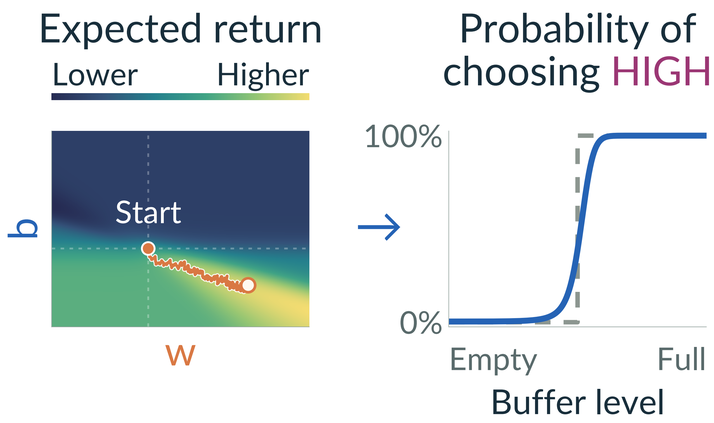
2. Enabling diagnosis and visualization

Choose your RL formulation to represent the heuristic



- ONE Input: Buffer level x
- TWO Actions: HIGH or LOW bitrate
- Training: Policy gradient
- Policy: $\sigma(w\,x + b)$

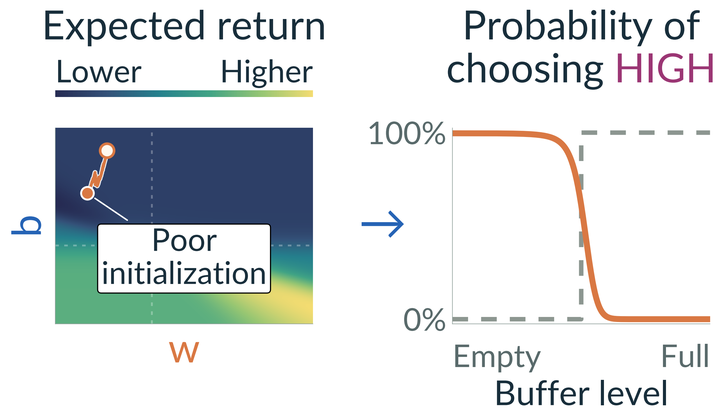
Now we can visualize the training and policy



✅ More buffer → Choose HIGH

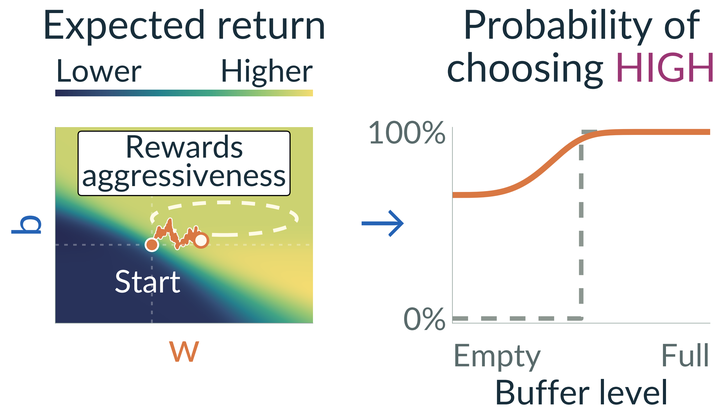
3. Case studies on common failures

Case 1: Learning gets stuck



❌ Empty buffer → Choose HIGH
❌ More buffer → Choose LOW

Case 2: Reward function favors risky choices



⚠️ Favors HIGH at any buffer level

paper

If your trained RL policy can't match the heuristic's behavior and performance, time to recheck your training pipeline!

demo