# Non-divergent Imitation for Verification of Complex Learned Controllers

Vahdat Abdelzad*, **Jaeyoung Lee***, Sean Sedwards*, Soheil Soltani*, Krzysztof Czarnecki

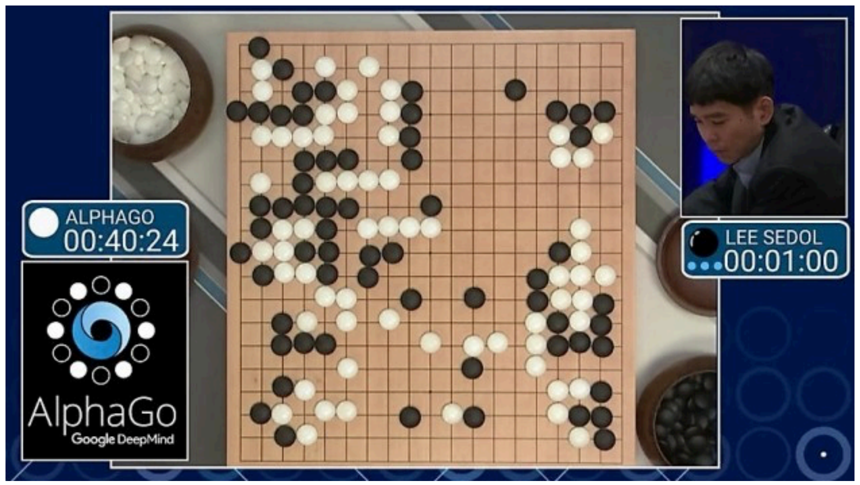
University of Waterloo, Canada

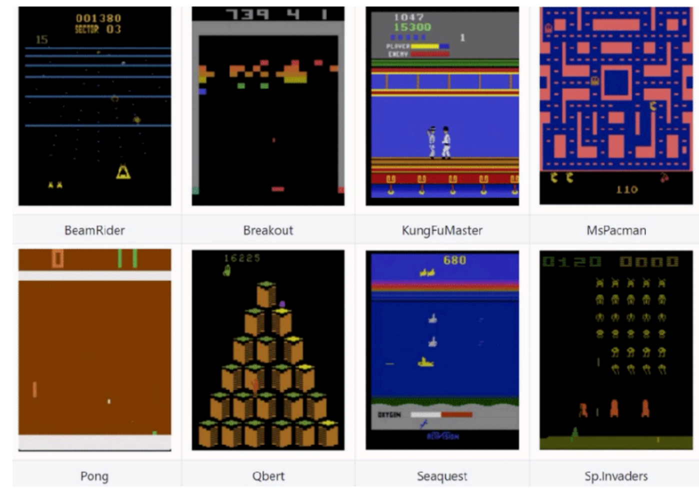
* contributed equally

Presented at IJCNN, 2021

# Introduction

- Machine learning solves complex sequential decision-making tasks





but solutions are often *opaque / difficult to formally verify*

- Our primary focus:

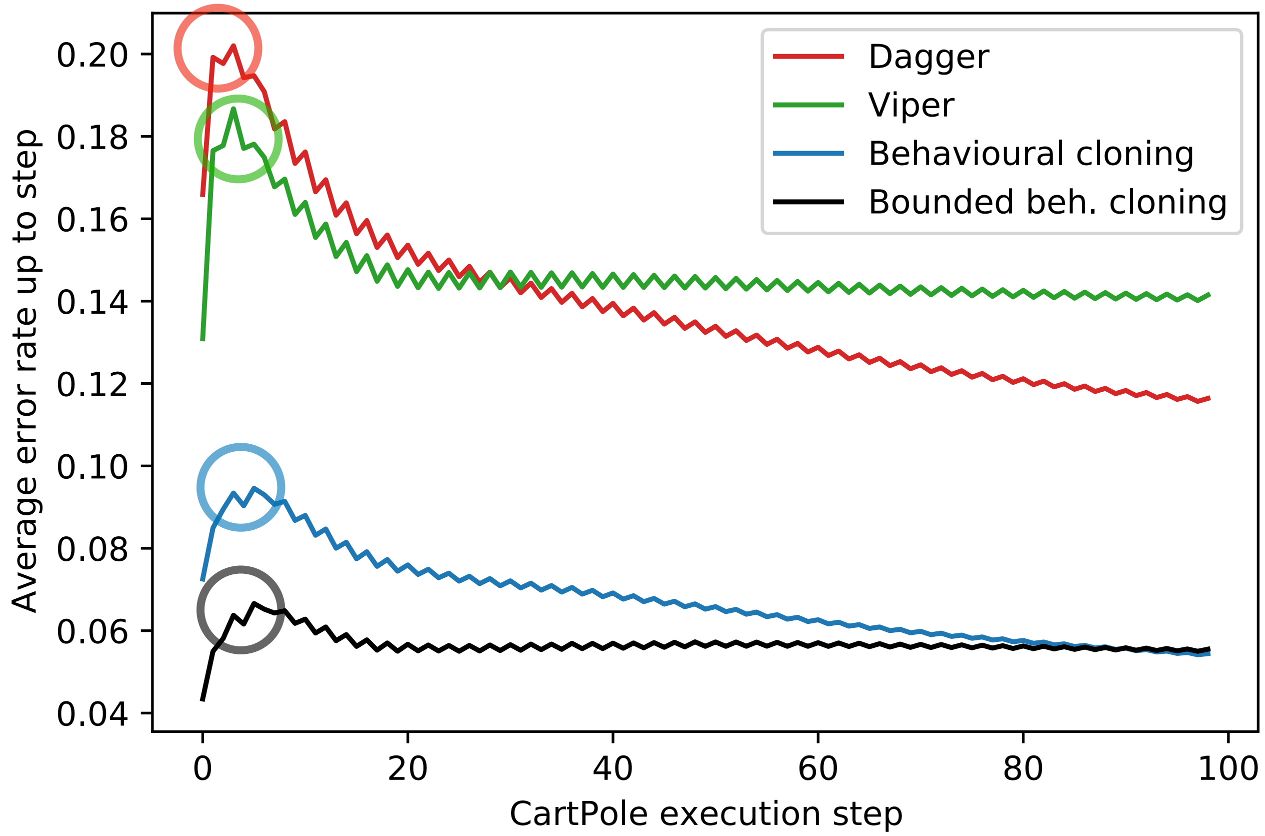
*Distill a learned controller into a verifiable structure such as a decision tree*

(*a learned controller* = *oracle* $\pi^*$; *a verifiable structure* = *solution* $\hat{\pi}$)

- Bounded model checker verifies temporal properties of system controlled by $\hat{\pi}$

E.g. in CartPole — *"pole angle always* $\leq 10^\circ$ *within* 100 *execution steps from any initial state"*

# Motivating Example: CartPole with a DQN Oracle $\pi^*$




Averages over 10 distillations
10000 rollouts used to estimate errors

Average error rate (1−accuracy) of the solution $\hat{\pi}$ w.r.t. the oracle $\pi^*$, up to a given execution step

- Accuracy of behavioural cloning $\gg$ accuracy of Dagger / Viper (especially in early execution steps)
- For bounded verification, accuracy in the early execution steps is critical (e.g. $\leq 40$)
- Bounded behavioural cloning

  trains the solution $\hat{\pi}$ on states from executions only up to 40 steps

  performs the best in early steps ➡ Idea: limit training data up to reasonable execution steps

# Markov Decision Process

- $(\mathcal{S}, \mathcal{A}, d, \mathcal{P})$ where

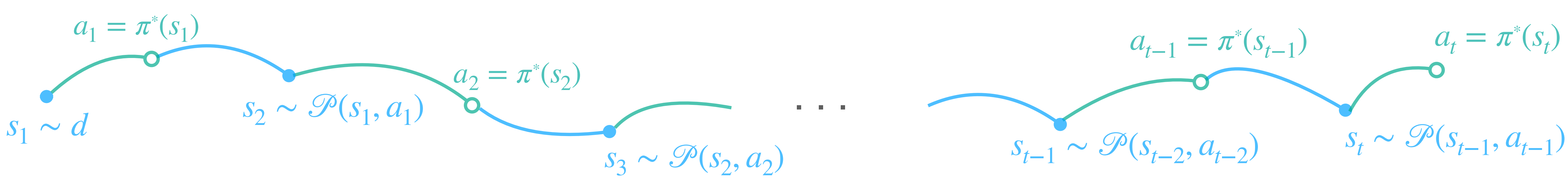
  state space $\mathcal{S}$

  finite action space $\mathcal{A}$

  initial state distribution $d$

  next-state distribution $\mathcal{P}(s, a)$, given current state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$

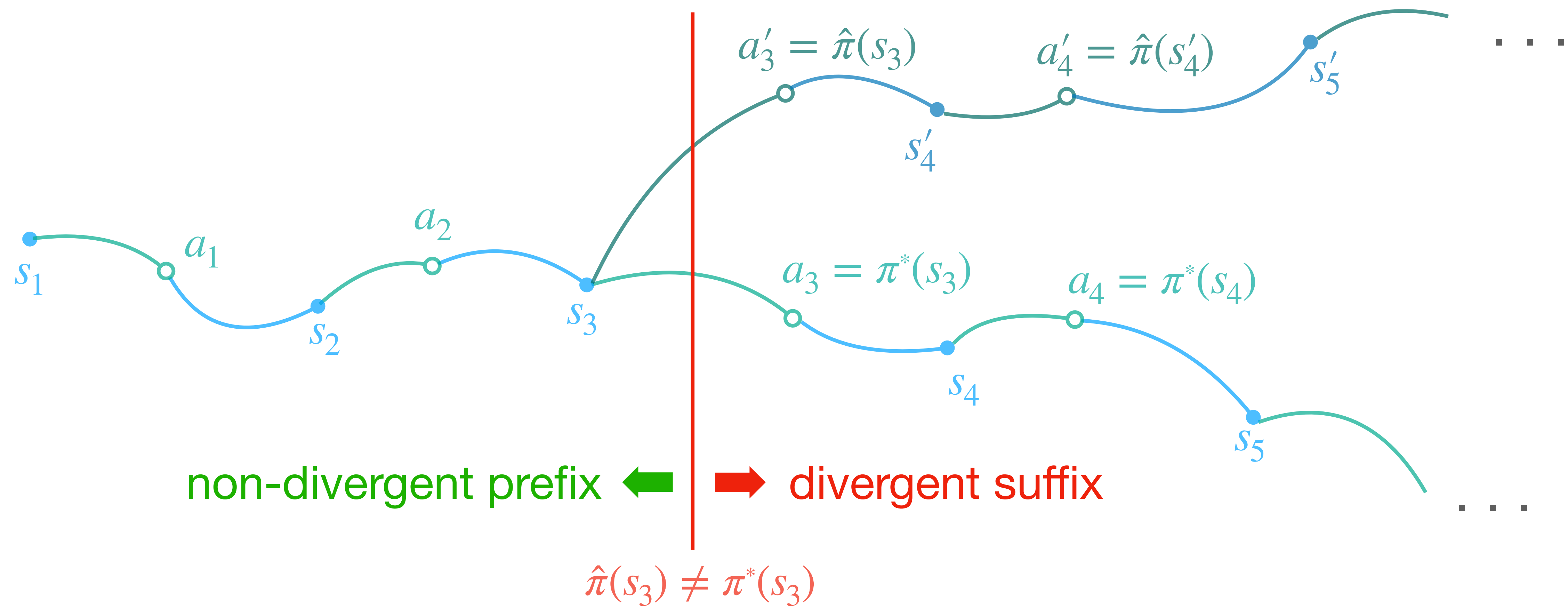
- *Finite path* $\tau \equiv \tau^{\pi^*} = s_1 a_1 s_2 a_2 \cdots s_t a_t$ generated by oracle $\pi^* : \mathcal{S} \to \mathcal{A}$




where the path length $|\tau| := t \in \mathbb{N}$

# Fidelity Issue

- Errors in the early execution steps can generate totally different paths thereafter



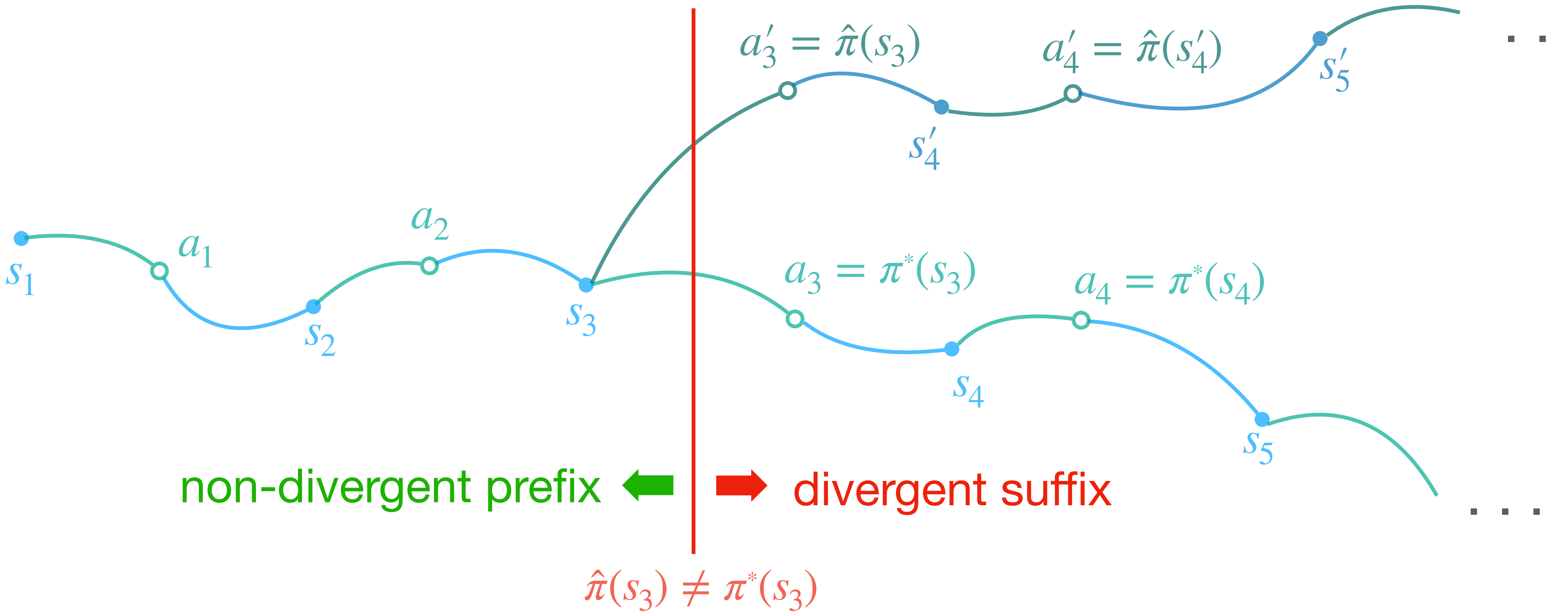

In this case, $\begin{cases} \langle\text{verification of distilled solution } \hat{\pi}\rangle \neq \langle\text{verification of oracle } \pi^*\rangle \\ \text{accuracy at states } s_4, s_5, s_6 \cdots \text{ is meaningless} \end{cases}$

➡ *Accuracy is NOT a sufficient metric for verification*

# Non-divergent Path Length (NPL)

- Definition:

$$\ell(\pi\,|\,\tau) := \max \left\{ t \in \{0, 1, 2, \cdots, |\tau|\} \,\middle|\, t = 0 \text{ or } \pi(s_n) = a_n \ \forall 1 \leq n \leq t \right\}$$




In this example, $\ell(\hat{\pi}\,|\,\tau) = 2$ ➡ the higher, the better

- Statistics of $\ell(\hat{\pi}\,|\,\tau)$ are suitable metrics to judge behavioural fidelity of $\hat{\pi}$ w.r.t. $\pi^*$

# NPL Maximization

$\Pi :=$ ⟨class of verifiable policies, to be optimized⟩

- Find a solution $\hat{\pi} \in \Pi$ maximizing the expected NPL over $\Pi$:

$$\hat{\pi} \in \arg\max_{\pi \in \Pi} \mathbb{E}\left[ ł(\pi \mid \tau) \right]$$

Let

$$\begin{cases} \ell(\pi \mid \tau) := \sum_{t=1}^{|\tau|} \mathbf{1}\left[ a_t = \pi(s_t) \right] & \text{(pathwise similarity)} \\ \tau_{1:t} := s_1 a_1 s_2 a_2 \cdots s_t a_t & \text{(path } \tau \text{ up to } t \text{ execution steps)} \end{cases}$$

**Lemma** $ł(\pi \mid \tau) = \ell(\pi \mid \tau_{1:ł(\pi|\tau)})$

- The NPL maximization is equivalent to

$$\hat{\pi} \in \arg\max_{\pi \in \Pi} \mathbb{E}\left[ \ell(\pi \mid \tau_{1:ł(\pi|\tau)}) \right]$$

# NPL Maximization

- The NPL maximization is equivalent to

$$\hat{\pi} \in \arg\max_{\pi \in \Pi} \mathbb{E}\left[\ell(\pi \mid \tau_{1:\ell(\pi|\tau)})\right] = \arg\max_{\pi \in \Pi} \mathbb{E}\left[\ell(\pi \mid \tau_{1:\ell(\pi|\tau)+1})\right]$$

**Lemma** $\ell(\pi \mid \tau_{1:\ell(\pi|\tau)}) = \ell(\pi \mid \tau_{1:\ell(\pi|\tau)+1})$

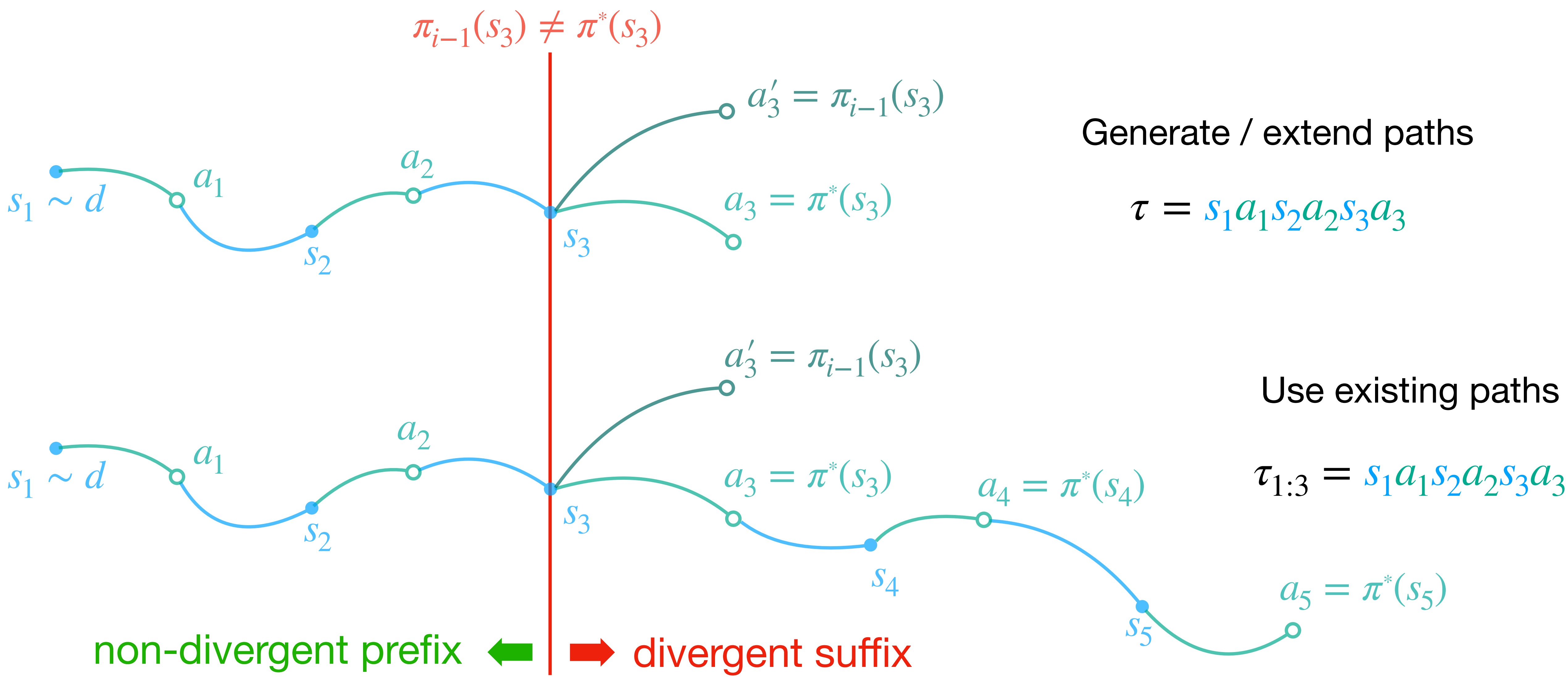
- Our proposal, Non-Divergent Imitation (NDI), is designed in a way that its fixed point $\pi_\bullet$ (if it exists) approximately satisfies

$$\pi_\bullet \in \arg\max_{\pi \in \Pi} \mathbb{E}\left[\ell(\pi \mid \tau_{1:\ell(\pi|\tau)+1})\right]$$

# Non-Divergent Imitation (NDI)

- An iterative algorithm: for each iteration $i = 1, 2, 3, \cdots$
- Key idea:

  *Consider paths up to "non-divergent prefixes + 1" w.r.t. $\pi_{i-1}$ (previous policy)*



# Non-Divergent Imitation (NDI)

- Procedure at each iteration $i = 1, 2, 3, \cdots$

  *(1) Consider paths up to "non-divergent prefixes + 1" w.r.t.* $\pi_{i-1}$ *(previous policy)*

  E.g. $\tau_{1:3} = s_1 a_1 s_2 a_2 s_3 a_3,\ \tau' = s'_1 a'_1 s'_2 a'_2,\ \cdots$

  *(2) Construct dataset* $D$ *from all those paths*

  E.g. $D = \{(s_1, a_1), (s_2, a_2), (s_3, a_3)\} \cup \{(s'_1, a'_1), (s'_2, a'_2)\} \cup \cdots$

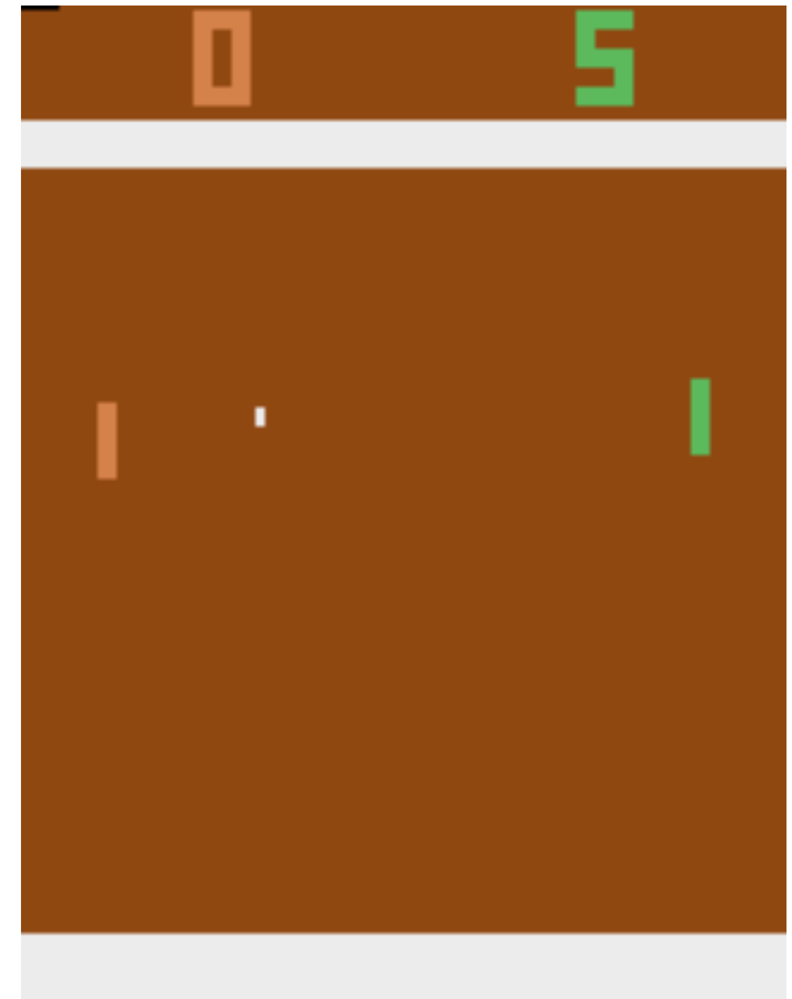
  **Input** ($s_2$) — **Output** ($a_2$)

  *(3) Train the next verifiable policy* $\pi_i$ *on* $D$

  ➡ $\pi_i \in \arg\max_{\pi \in \Pi} \mathbb{E}\left[\ell(\pi \mid \tau_{1:\not{\ell}(\pi_{i-1}|\tau)+1})\right]$ (approximately)

  *where* $\pi_0$ *is the oracle* $\pi^*$

# Experiments — Pong




- Pixel-based traditional computer game
- Used in related literature, *with state abstraction*
- Treated as a dynamical system, with

  state space $\mathscr{S} = \mathbb{Z}^7$ (abstracted / extracted from pixel images)

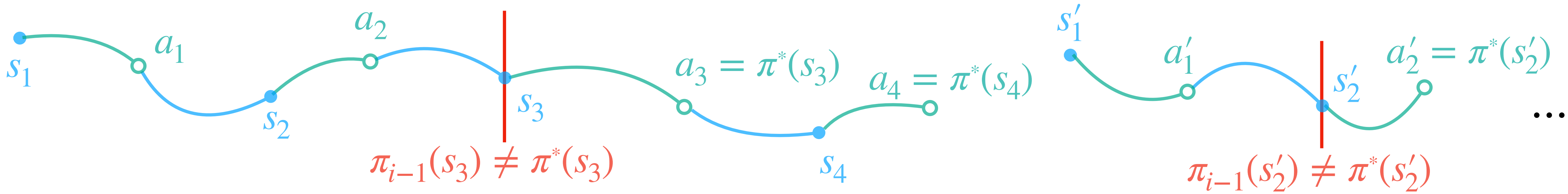
  action space $\mathscr{A} = \{\texttt{NoOp}, \texttt{Fire}, \texttt{Right}, \texttt{Left}, \texttt{RightFire}, \texttt{LeftFire}\}$

- Train decision tree policies $\pi_1, \pi_2, \cdots, \pi_{40}$ (with tree depth $\leq 12$)

# NDI vs BC

- Non-Divergent Imitation (NDI)

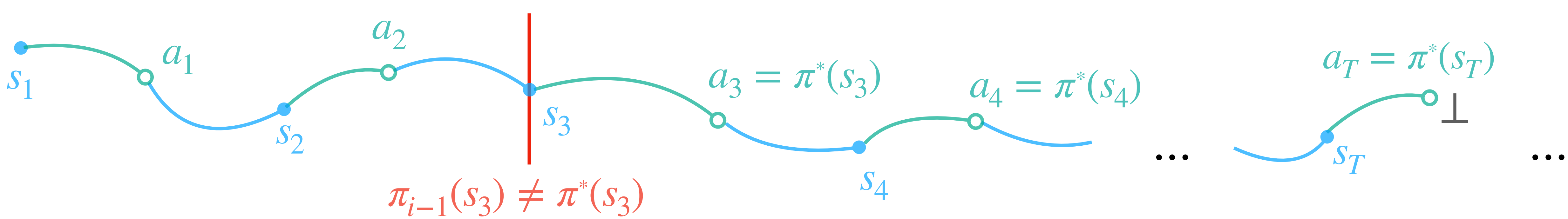
  Trained on “non-divergent prefixes + 1”




➡ $D = \{(s_1, a_1), (s_2, a_2), (s_3, a_3)\} \cup \{(s'_1, a'_1), (s'_2, a'_2)\} \cup \cdots$

- Behavioural Cloning (BC)

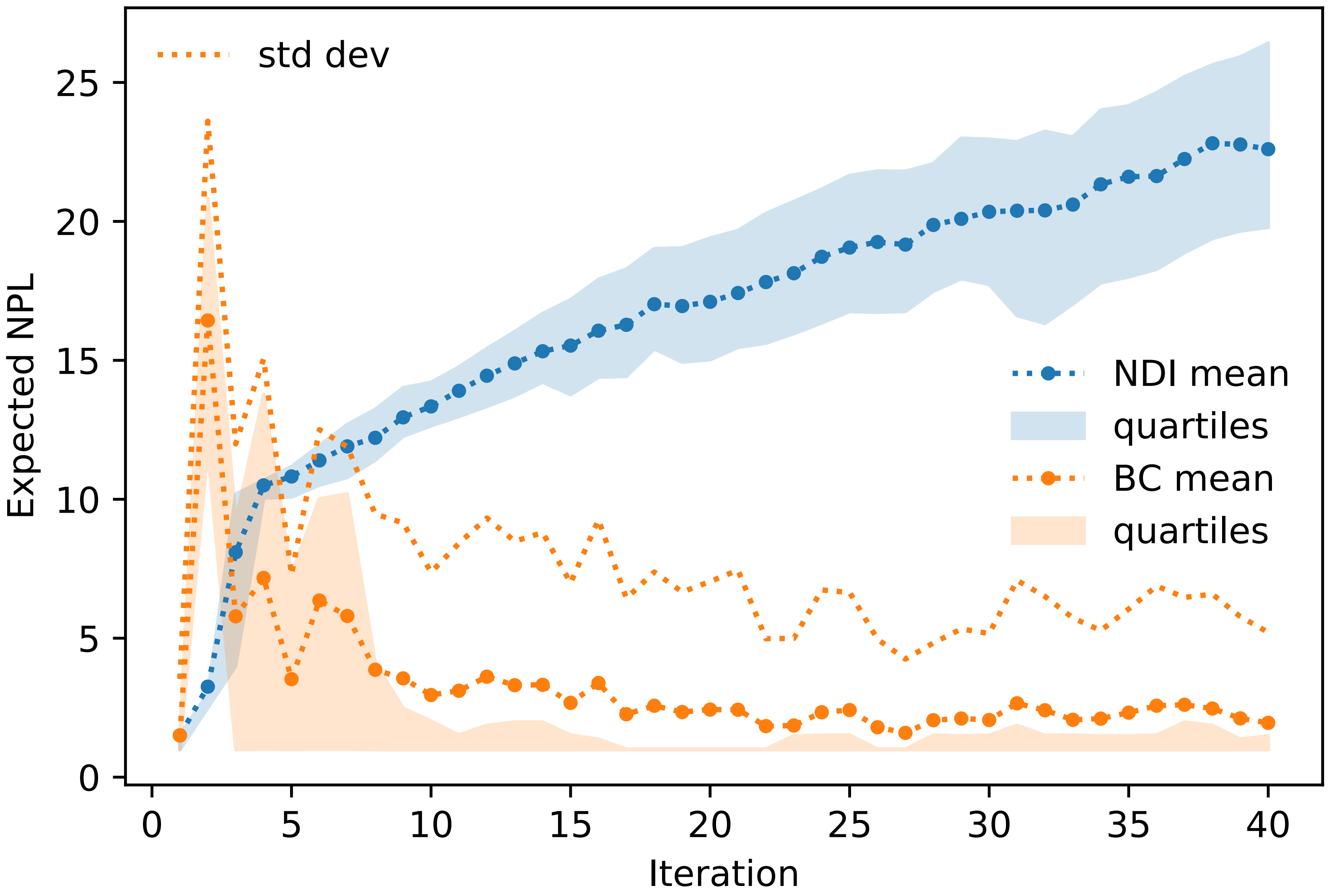
  Trained on the same # of data in $D$, but obtained from entire rollouts




➡ $D = \{(s_1, a_1), (s_2, a_2), \cdots, (s_T, a_T)\} \cup \{(s'_1, a'_1), \cdots, (s'_T, a'_T)\} \cup \cdots$

# Expected NPL vs Iteration

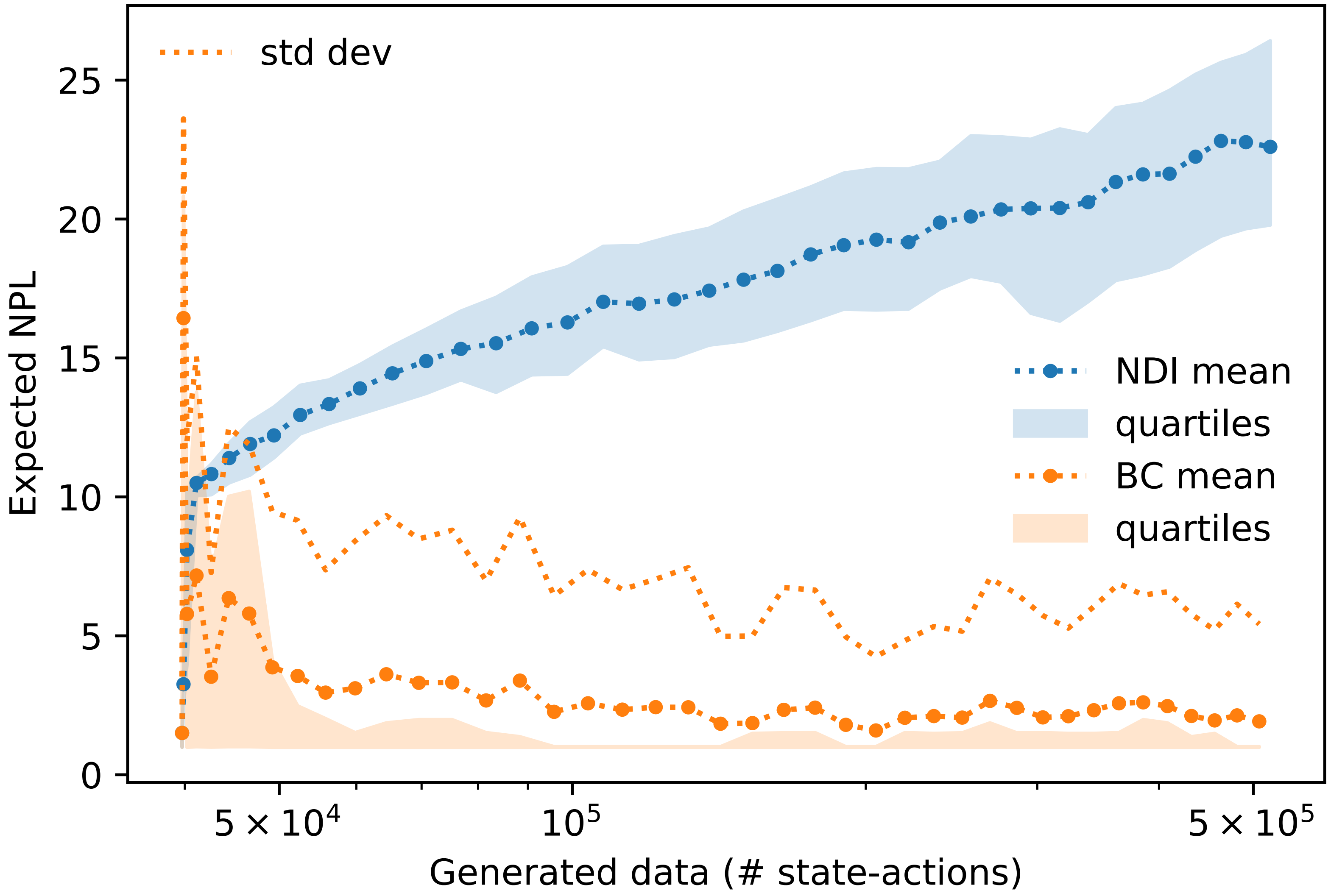
- Statistics are w.r.t. 50 repetitions
- Expected NPL estimated with 20000 rollouts




➡ *NDI keeps increasing expected NPL!*

# Expected NPL vs Generated Data

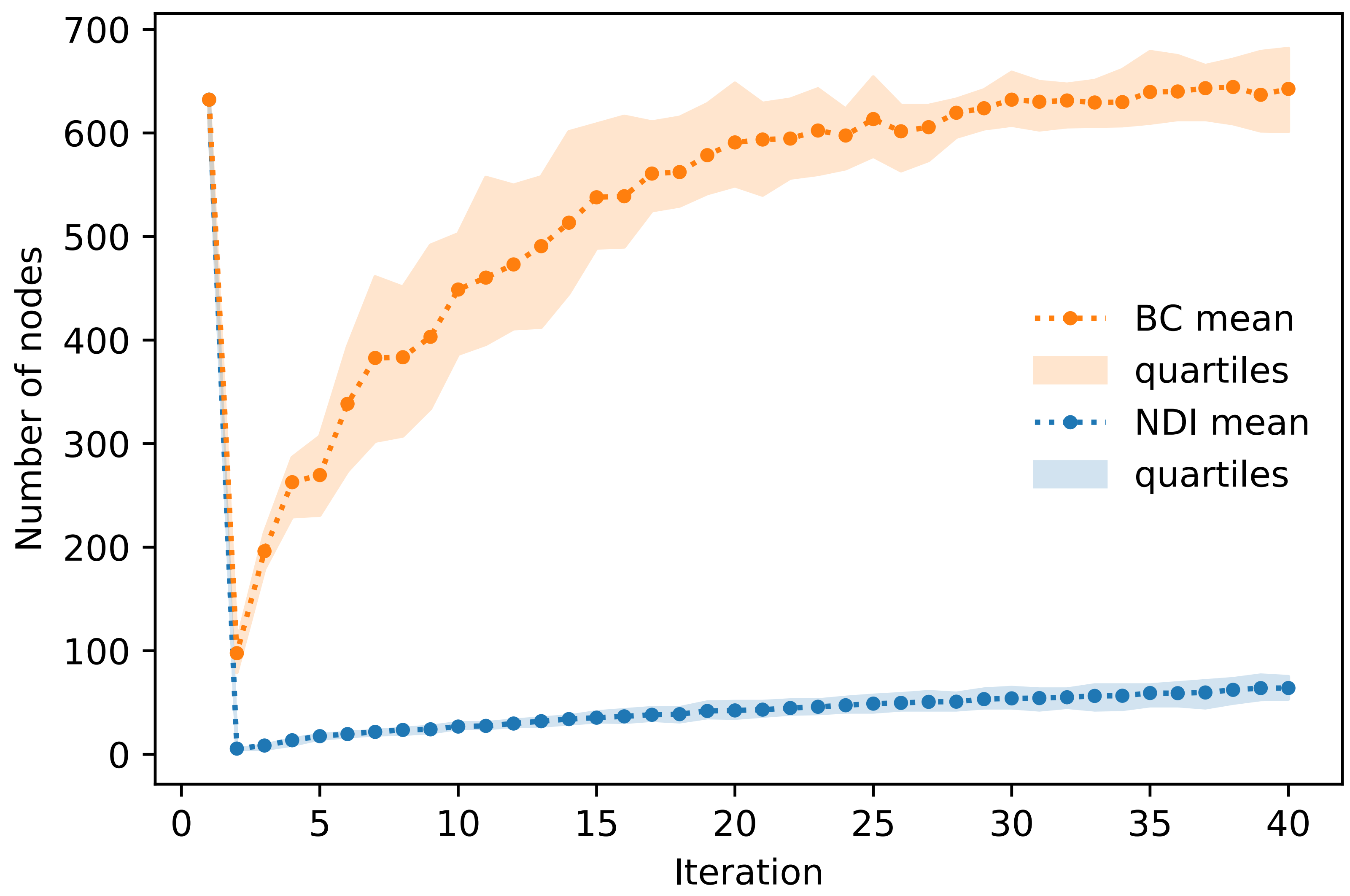
- Statistics are w.r.t. 50 repetitions
- Expected NPL estimated with 20000 rollouts




➡ *NDI is more data-efficient!*

# Number of Nodes vs Iteration

- Statistics are w.r.t. 50 repetitions




➡ *NDI produces more compact models!*

# Conclusion

- Contributions

1. New Concept: Non-divergent Path Length (NPL)

   ✓ A metric for behavioural fidelity of distilled models for verification

2. Algorithm: Non-Divergent Imitation (NDI)

   ✓ Achieves higher expected NPL than the state-of-the-art

- Challenges

1. Divergent dynamics

2. Aleatoric uncertainty (e.g. in Pong)