Introduction to Richens et al., 2025: "General Agents Need World Models"

Kevin Denamganaï

October 14th, 2025

Outline

1 Background & Overview

2 Environment & Assumption

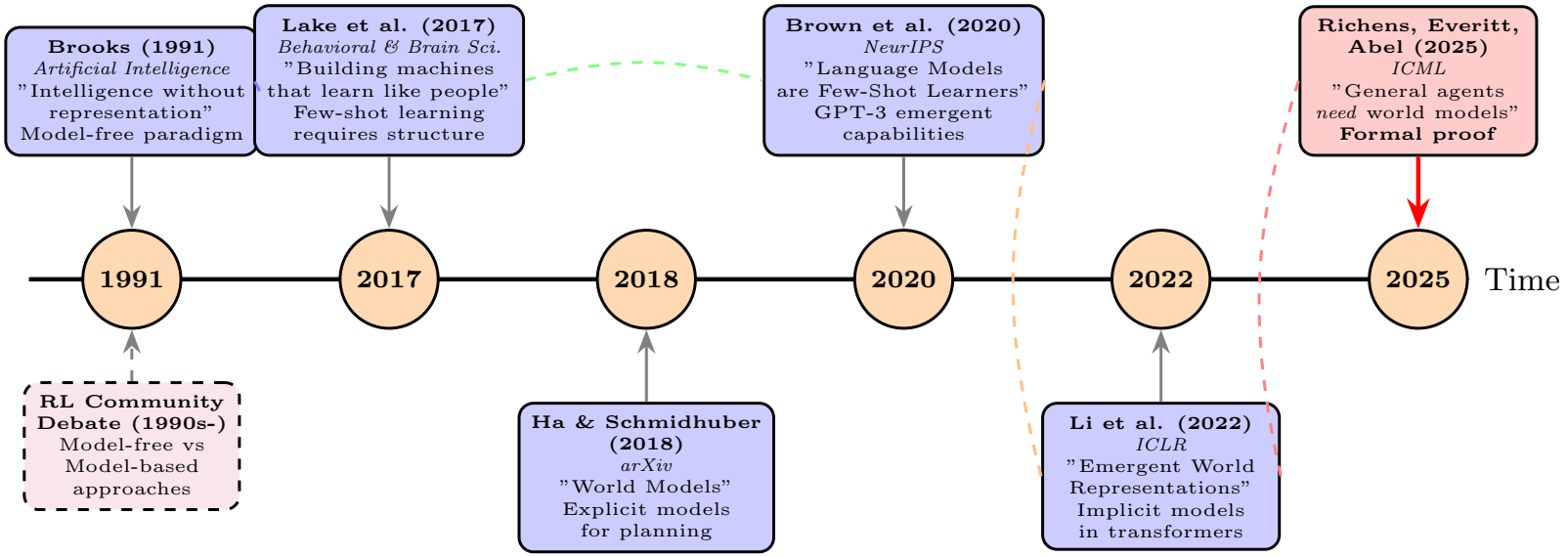
3 Agents & Goals

4 Building Intuition

5 Theorem 1 & Algorithm 1

6 Limitations

Historical Context: The World Model Debate



Evolution of the Debate

1990s-2010s: RL community debates model-free (sample efficient, no world knowledge) vs model-based (requires accurate models, hard to learn)
Brooks (1991): "The world is its own best model"—representations unnecessary
Lake et al. (2017): Few-shot learning requires compositional structure & world knowledge
2020-2022: Evidence of implicit world models in model-free agents

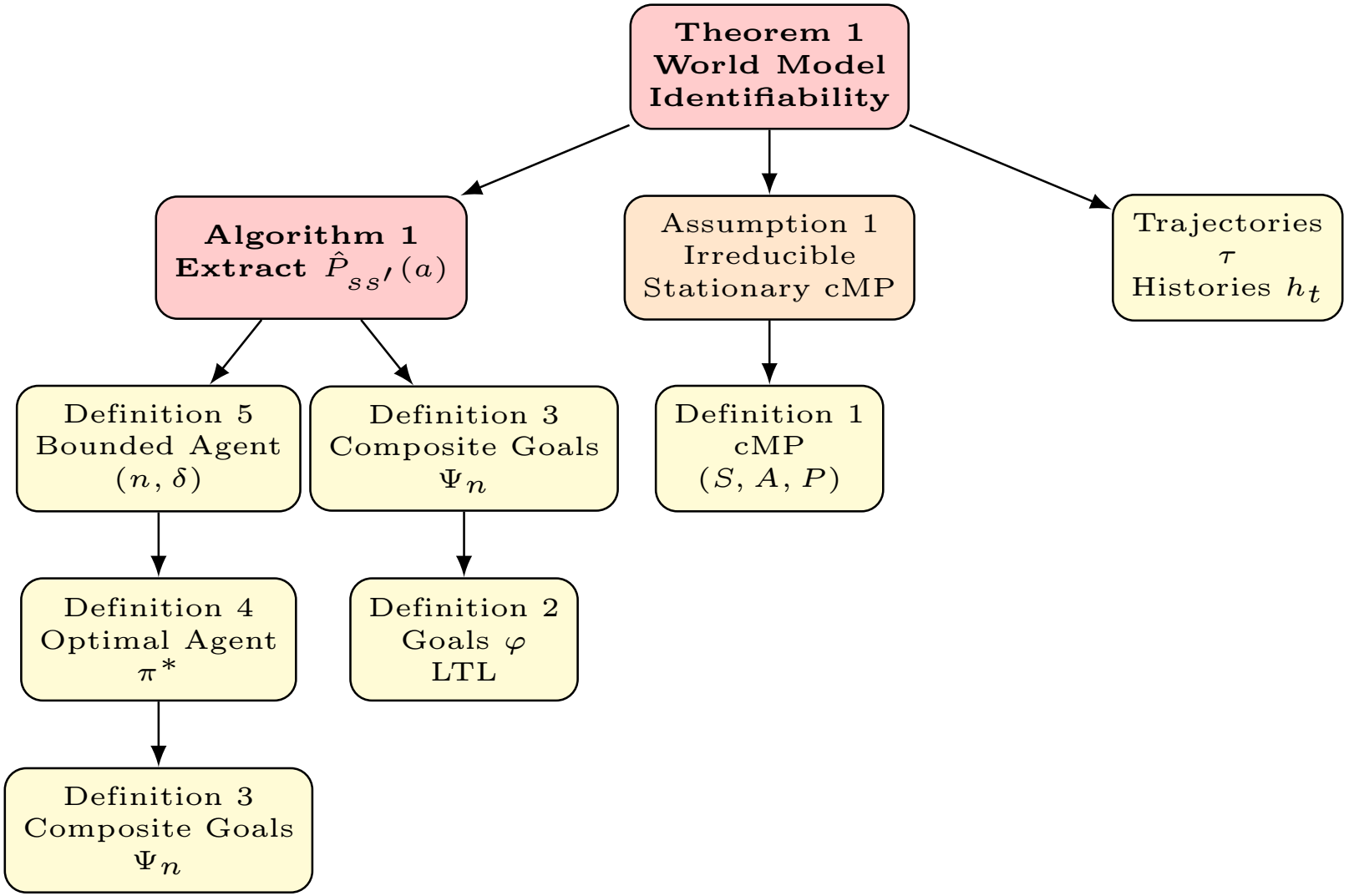
Research Question & Contributions

"Are world models a necessary ingredient for flexible, goal-directed behaviour, or is model-free learning sufficient?"

Contributions :

- multi-step goal-directed task generalization REQUIRES learning a **predictive model** of the environment
- Agent's **internal world model** can be extracted from agent's policy alone (with Algo.1).
- improving the performance of an agent over increasingly more complex goal-directed tasks REQUIRES improving the accuracy of the **internal world model** of said agent.

Concept Dependency Tree



Figure

Outline

1 Background & Overview

2 Environment & Assumption

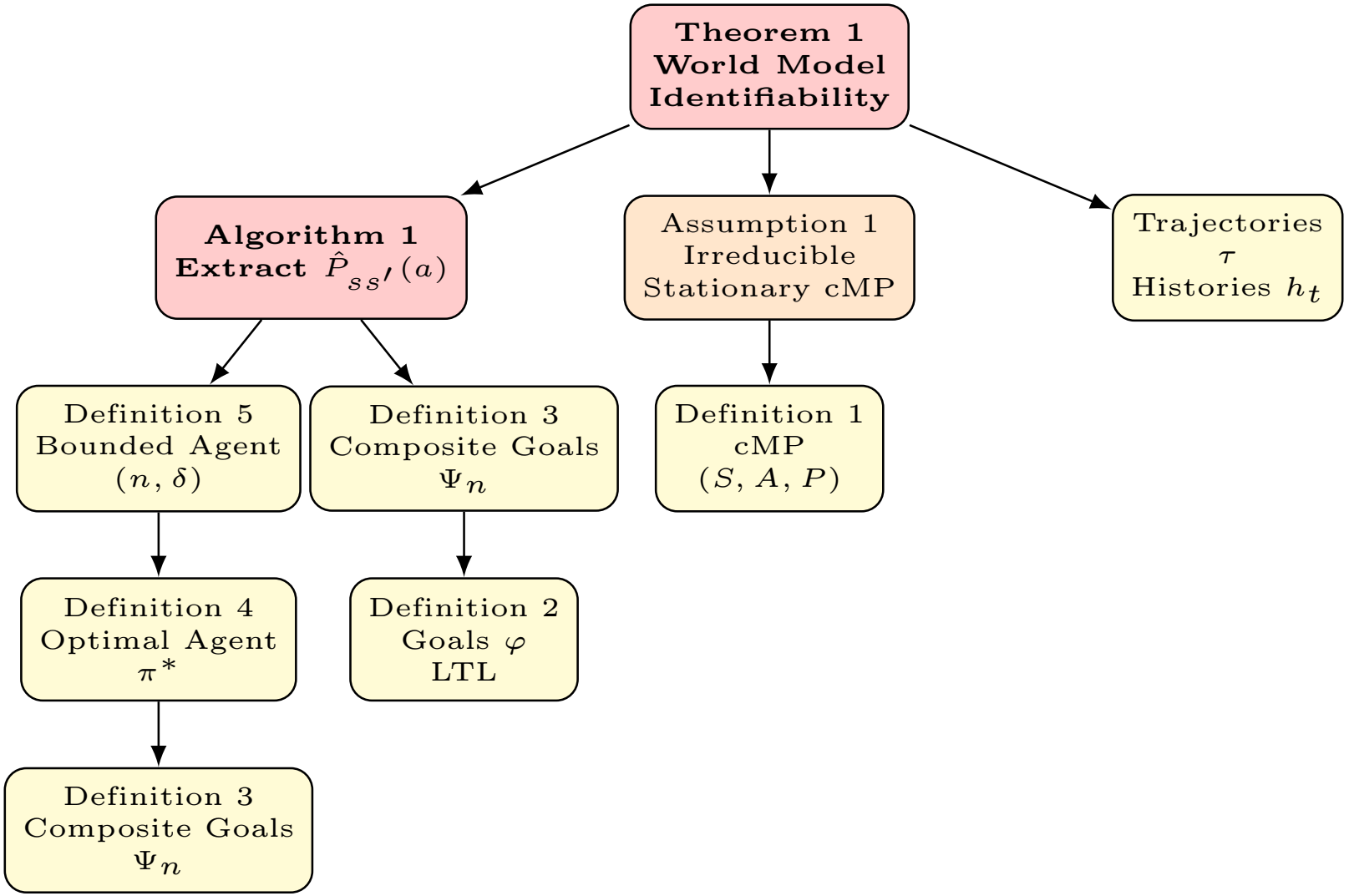
3 Agents & Goals

4 Building Intuition

5 Theorem 1 & Algorithm 1

6 Limitations

Concept Dependency Tree



Figure

Definition 1: Controlled Markov Process

A **controlled Markov process (cMP)** is a Markov decision process (MDP) without a specified reward function or discount factor. It is defined by the tuple $(S, A, P_{ss'}(a))$ where:

- S is the state space
- A is the action space
- $P_{ss'}(a) = P(S = s' \mid A = a, S = s)$ is the transition function

Trajectories and History

- A **trajectory** is a sequence of state-action pairs over time:

$$\tau = (s_0, a_0, s_1, a_1, \ldots)$$

- A **history** is a finite prefix of τ:

$$h_t = (s_0, a_0, \ldots, s_t)$$

Assumption 1

We assume the environment is described by an **irreducible, stationary, finite, controlled Markov process** (Def. 1) with at least two actions.

This means:

- Every state is reachable from every other state under some finite sequence of actions
- Transition probabilities do not change over time
- The state and action spaces are finite
- $|A| \geq 2$

Outline

1 Background & Overview

2 Environment & Assumption

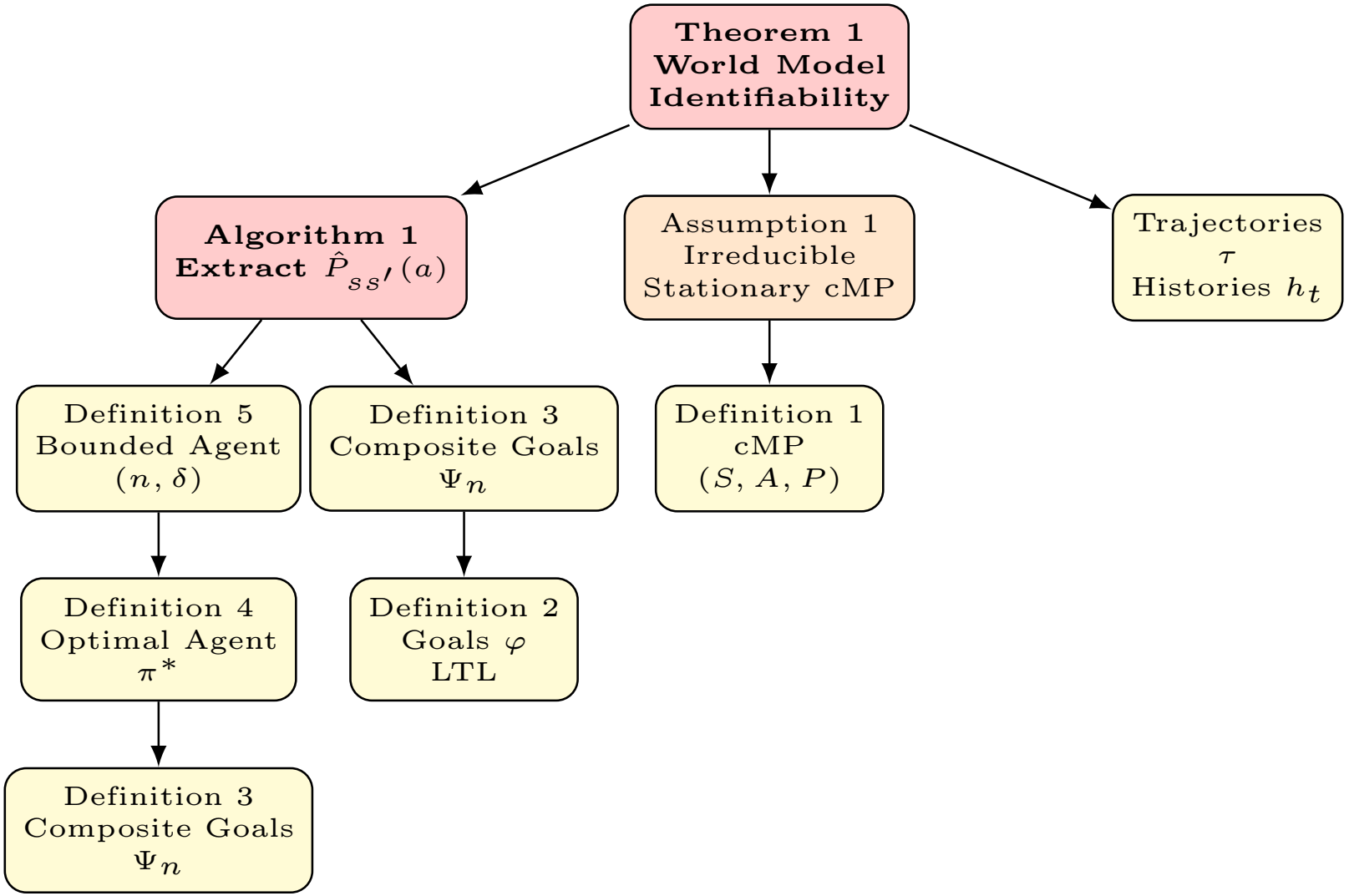
3 Agents & Goals

4 Building Intuition

5 Theorem 1 & Algorithm 1

6 Limitations

Concept Dependency Tree



Figure

Goal-Conditioned Agents

Goal-conditioned agents are policies π that map histories and goals to actions:

$$\pi : h_t, \psi \mapsto a_t$$

- The policy takes as input the history h_t and a goal ψ
- It outputs an action a_t
- We assume the environment is fully observed by the agent

Definition 2: Goals

A **goal** φ is an Linear Temporal Logic (LTL) expression of the form $\varphi = \mathcal{O}([(s, a) \in g])$ where:

- g is a set of goal-states, a subset of the joint states of the environment-agent system $(s, a) \in S \times A$
- $\mathcal{O}$ is a temporal operator specifying the time horizon for reaching g. We restrict to $\mathcal{O} \in \{\bigcirc, \Diamond, \top\}$ where:
  - $\bigcirc$ = Next
  - $\Diamond$ = Eventually
  - $\top$ = Now

Goals & Trajectory Satisfaction

A trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots)$ **satisfies** a goal φ, denoted $\tau \models \varphi$, iff the sequence of states and actions meets the goal specification.

Examples:

- $\varphi = \Diamond([S = s_{\text{goal}}])$: $\tau \models \varphi$ iff $\exists t \geq 0$ such that $s_t = s_{\text{goal}}$
- $\varphi = \bigcirc([S = s'])$: $\tau \models \varphi$ iff $s_1 = s'$

Definition 3: Sequential & Composite Goals

- A **sequential goal** ψ is an ordered sequence of sub-goals (Def. 2):

$$\psi = \langle \varphi_1, \ldots, \varphi_n \rangle$$

  where the agent must achieve sub-goal φ_i before φ_{i+1}.
- The **depth** of a sequential goal is the number of sub-goals: $\text{depth}(\psi) = n$.

Definition 3: Sequential & Composite Goals

- A **composite goal** is a disjunction of one or more sequential goals:

$$\psi = \bigvee_{i=1}^{m} \psi_i$$

- The depth of a composite goal is: $\text{depth}(\psi) = \max_{\psi_i} \text{depth}(\psi_i)$.
- Ψ_n is the set of all composite goals ψ with $\text{depth}(\psi) \leq n$.

Definition 4: Optimal Goal-Conditioned Agent

For a given set of goals Ψ (Def. 3), an **optimal agent** is a goal-conditioned policy $\pi^*(a_t \mid h_t; \psi)$ where π^* is deterministic and satisfies:

$$\pi^* = \arg\max_{\pi} P(\tau \models \psi \mid \pi, s_0)$$

$\forall s_0$ s.t. $P(s_0) > 0$, where s_0 is the initial state of the environment at $t = 0$, and $\forall \psi \in \Psi$.

Definition 5: Bounded Goal-Conditioned Agent

A **bounded goal-conditioned agent** is a goal-conditioned policy $\pi(a_t \mid h_t; \psi)$ satisfying:

$$P(\tau \models \psi \mid \pi, s_0) \geq \max_{\pi} P(\tau \models \psi \mid \pi, s_0)(1 - \delta)$$

$\forall \psi \in \Psi_n$ where:

- n is the maximum goal depth
- s_0 is the initial state of the environment at $t = 0$
- $\delta \in [0, 1]$ is the failure rate (regret bound)

Outline

1 Background & Overview

2 Environment & Assumption

3 Agents & Goals

4 Building Intuition

5 Theorem 1 & Algorithm 1

6 Limitations

Toy Experiment

The Scenario:

- You're a basketball player with a teammate
- Your teammate is either Stephen Curry (70%) or Shaqille O'Neal (20%)
- You've practiced—you know who it is
- Your goal: performing a given pass and complete a challenge where your teammates shoot 3-pointers

Toy Experiment

The Scenario:

- You're a basketball player with a teammate
- Your teammate is either Stephen Curry (70%) or Shaqille O'Neal (20%)
- You've practiced—you know who it is
- Your goal: performing a given pass and complete a challenge where your teammates shoot 3-pointers

The Experimenter (Me):

- I watch your behaviour from outside
- I **don't know** who your teammate is
- I can only see your *first action*
- Goal: figure out your teammate's shooting % from your choices

Toy Experiment

The Scenario:

- You're a basketball player with a teammate
- Your teammate is either Stephen Curry (70%) or Shaqille O'Neal (20%)
- You've practiced—you know who it is
- Your goal: performing a given pass and complete a challenge where your teammates shoot 3-pointers

The Experimenter (Me):

- I watch your behaviour from outside
- I **don't know** who your teammate is
- I can only see your *first action*
- Goal: figure out your teammate's shooting % from your choices

Key Question

Can I deduce whether you're playing with Curry or Shaq just by watching whether you **bounce pass the ball** (φ_1) or **lob pass the ball** (φ_1')?

The Challenge Format

Your Decision:
At $t = 0$, you choose ONE action:

- Bounce pass ball (φ_1): Accept Challenge A
- Lob pass ball (φ_1'): Accept Challenge B

After your choice:

- Your teammate shoots $n = 10$ times
- You count successes
- Win or lose based on challenge

Challenge A (after a bounce pass):
"Teammate makes $\leq k$ out of 10 shots"

Challenge B (after a lob pass):
"Teammate makes $> k$ out of 10 shots"

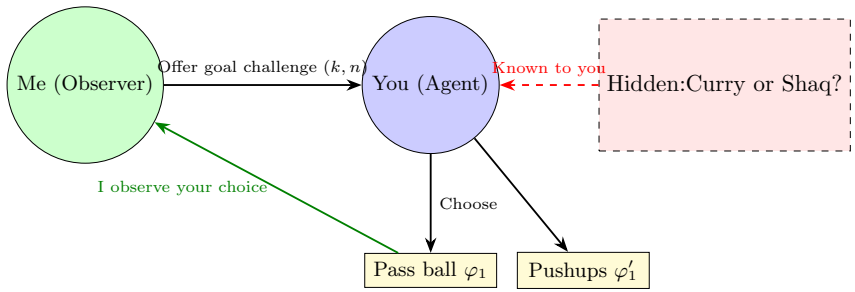
Your Strategy (as Agent)

You pick whichever challenge gives higher success probability—you know your teammate's skill!

My View (Experimenter): What Do I See?

I observe:

1. The challenge parameters $(k, n = 10)$
2. Your initial action:
   - Bounce pass the ball $(\varphi_1) \rightarrow$ Challenge A
   - Lob pass the ball $(\varphi_1') \rightarrow$ Challenge B
3. (I don't see the shooting after!)



My strategy:
Vary k from 0 to 10, watch when you switch from "lob pass" to "bounce pass"

Example: You're Partnered with Curry (70% Accuracy)

Your reasoning:

Challenge $k = 2$:

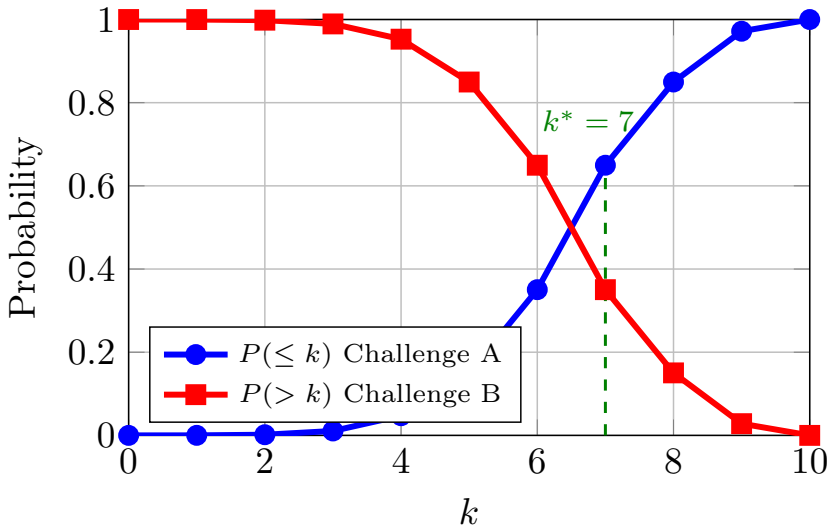
- A: Make $\leq 2 \rightarrow 0.2\%$
- B: Make $> 2 \rightarrow 99.8\%$
- You: Lob pass the ball (B)

Challenge $k = 5$:

- A: Make $\leq 5 \rightarrow 15\%$
- B: Make $> 5 \rightarrow 85\%$
- You: Lob pass the ball (B)

Challenge $k = 7$:

- A: Make $\leq 7 \rightarrow 65\%$
- B: Make $> 7 \rightarrow 35\%$
- You: Bounce pass the ball (A)



Switch at $k^* = 7$

My observation:
You switch from "lob pass" to "bounce pass" at $k = 7$

My estimate:
$\hat{P}_{\text{make}} = 7/10 = 0.70$

Example: You're Partnered with Shaq (20% Accuracy)

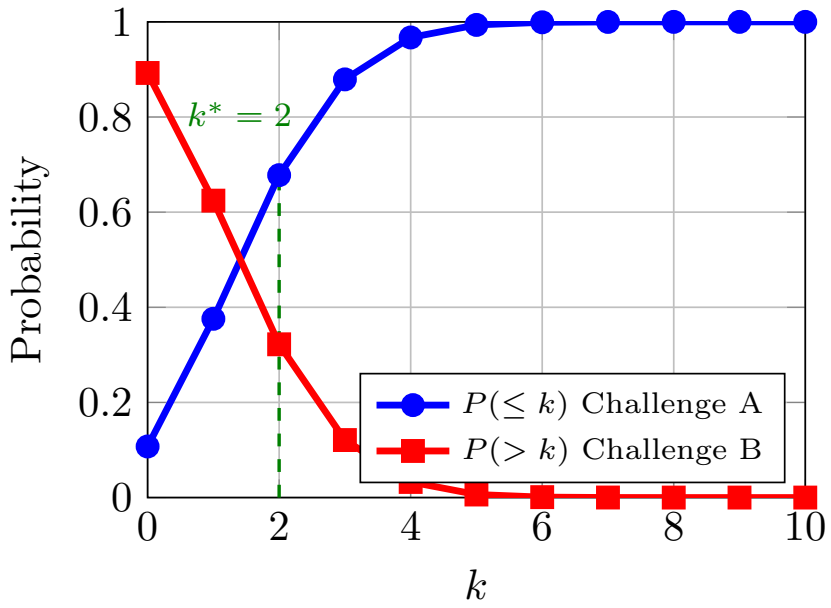
Your reasoning:

Challenge $k = 2$:

- A: Make $\leq 2 \rightarrow 68\%$
- B: Make $> 2 \rightarrow 32\%$
- You: Bounce pass the ball (A)

You immediately start doing bounce passing the ball to your teammate!

Key difference:
With Shaq, you switch *very early* ($k^* = 2$)

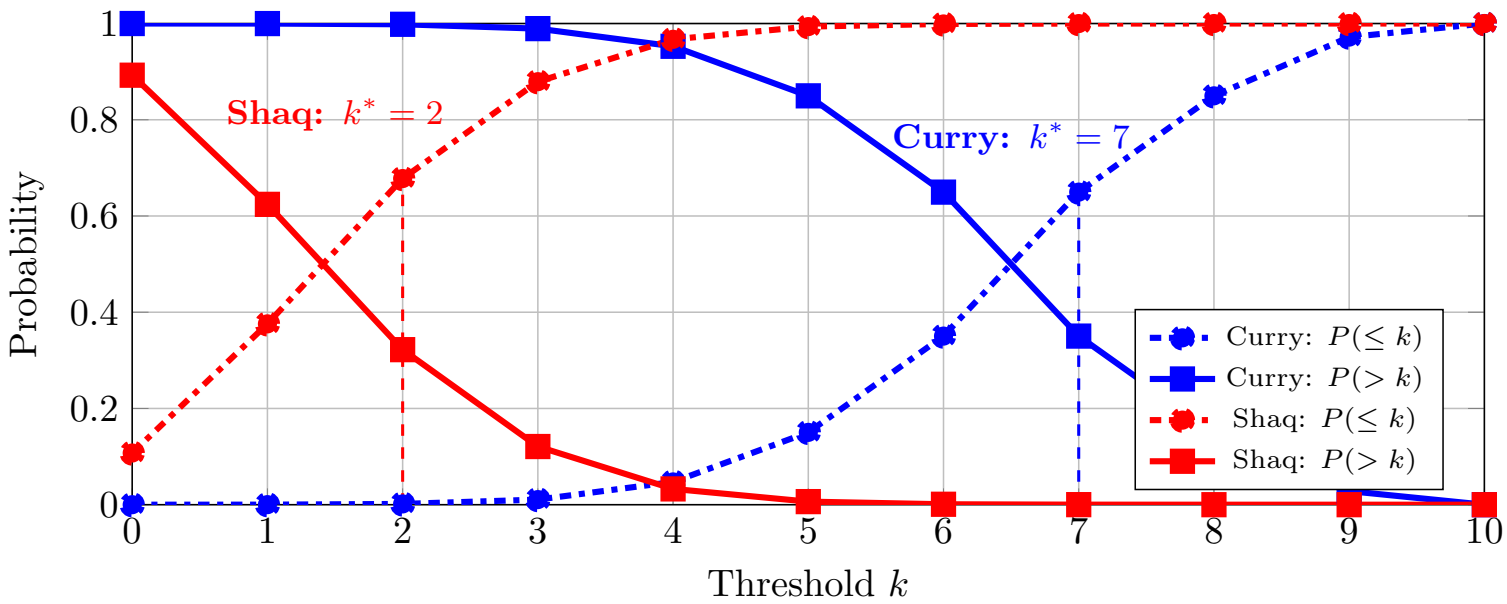


Switch at $k^* = 2$

My observation:
You switch to "bounce pass" at $k = 2$

My estimate:
$\hat{P}_{\text{make}} = 2/10 = 0.20$

Curry vs Shaq: Dramatic Difference



What I Learn from Your Behavior

- **Curry teammate**: You're confident—lob pass the ball for high thresholds ($k^* = 7$)
- **Shaq teammate**: You're cautious—do bounce pass even for low thresholds ($k^* = 2$)
- **Your switch point reveals your hidden knowledge about the environment (teammate skill)!**

Why It Works?

Your Decision Rule:
You switch when:

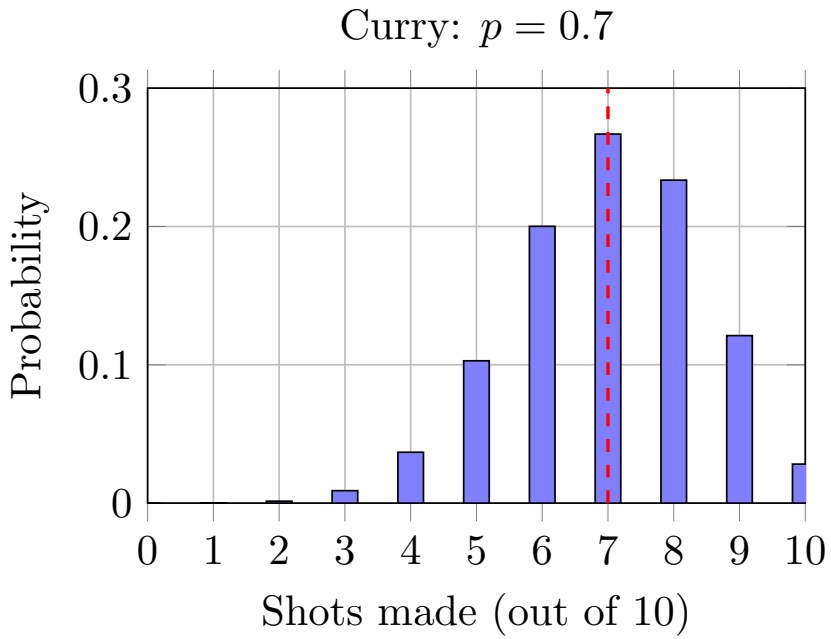
$$P(\leq k) \geq P(> k)$$

This happens at the **binomial median**:

$$k^* \approx \lfloor n \cdot p \rfloor$$

Therefore:

$$\hat{p} = \frac{k^*}{n} \approx p$$



Curry's distribution peaks at 7—the switch point!

Core Insight

Your switch point finds the median of the binomial distribution, which approximates the true probability. I can extract your hidden knowledge by observing your action choices!

Summary

The Setup

- **You** = Agent, trained in environment (and learned a world model)
- **Your teammate** = Hidden env. param. (Curry 70% or Shaq 20%)
- **Your action** = Bounce pass the ball (φ_1) or lob pass the ball (φ_1')
- **Me** = Experimenter only observes your actions

Summary

The Setup

- **You** = Agent, trained in environment (and learned a world model)
- **Your teammate** = Hidden env. param. (Curry 70% or Shaq 20%)
- **Your action** = Bounce pass the ball (φ_1) or lob pass the ball (φ_1')
- **Me** = Experimenter only observes your actions

The Procedure

1. I offer challenges: "teammate makes $\leq k$ vs $> k$ shots"
2. You choose: bounce pass (if $> k$ is most likely) or lob pass (if $\leq k$ is most likely)
3. I vary k and watch your switch point k^*
4. I estimate: $\hat{P}_{\text{make}} = k^*/n$

Summary

The Setup

- **You** = Agent, trained in environment (and learned a world model)
- **Your teammate** = Hidden env. param. (Curry 70% or Shaq 20%)
- **Your action** = Bounce pass the ball (φ_1) or lob pass the ball (φ_1')
- **Me** = Experimenter only observes your actions

The Procedure

1. I offer challenges: "teammate makes $\leq k$ vs $> k$ shots"
2. You choose: bounce pass (if $> k$ is most likely) or lob pass (if $\leq k$ is most likely)
3. I vary k and watch your switch point k^*
4. I estimate: $\hat{P}_{\text{make}} = k^*/n$

The Insight

Your behaviour reveals your world knowledge.
I can extract it by observing your action choices over multi-step goals!

Outline

1 Background & Overview

2 Environment & Assumption

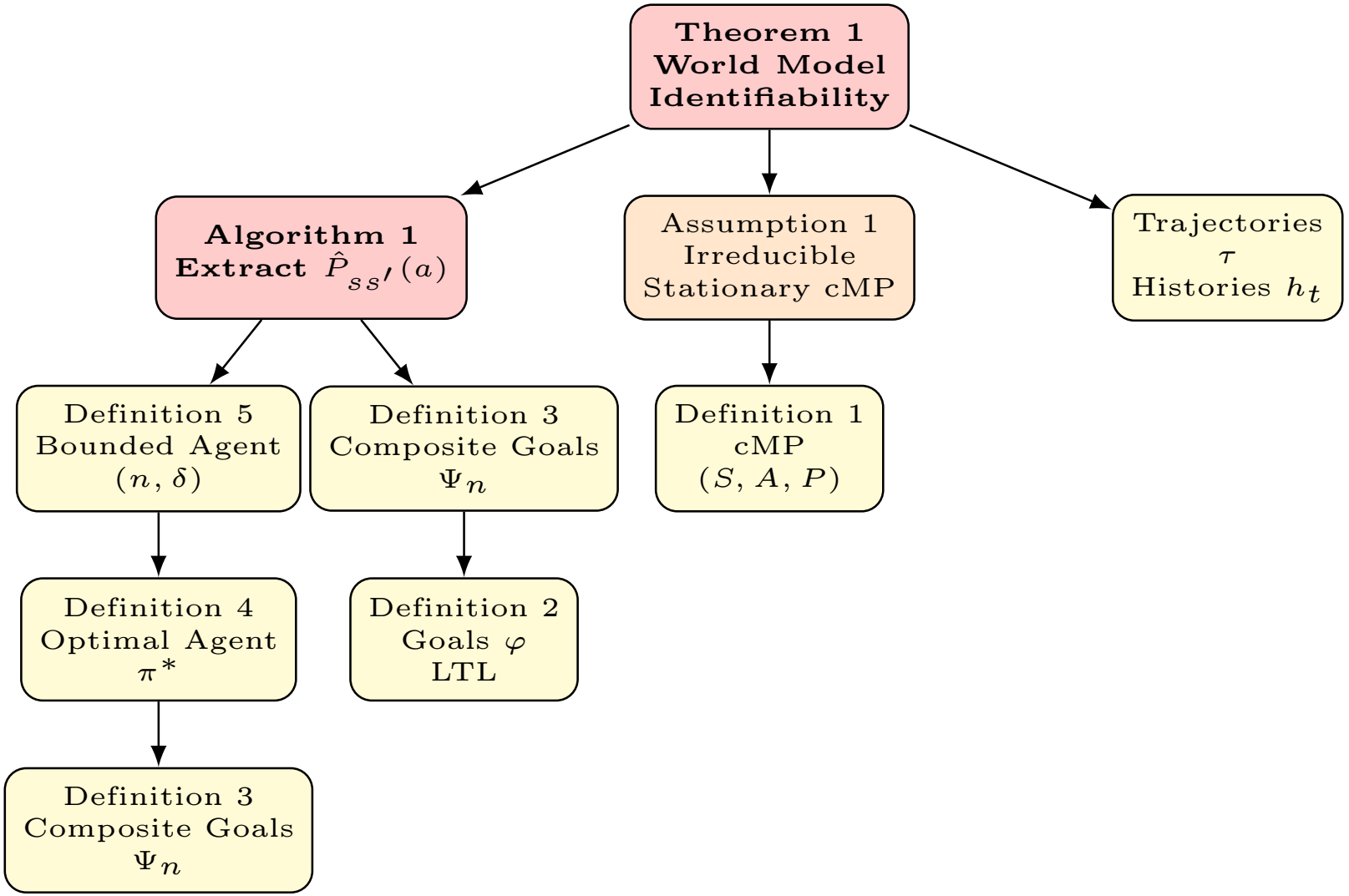
3 Agents & Goals

4 Building Intuition

5 Theorem 1 & Algorithm 1

6 Limitations

Concept Dependency Tree



Figure

Theorem 1 (Part 1)

Let $P_{ss'}(a) = P(S_{t+1} = s' \mid A_t = a, S_t = s)$ be the transition probabilities of an environment satisfying Assumption 1.

Let π be a goal-conditioned agent (Def. 5) with a maximum failure rate δ for all goals $\psi \in \Psi_n$ where Ψ_n is the set of all composite goals with maximum goal depth $n > 1$.

Then π fully determines a model for the environment transition probabilities $\hat{P}_{ss'}(a)$, with errors satisfying:

$$\left|\hat{P}_{ss'}(a) - P_{ss'}(a)\right| \leq \sqrt{\frac{2P_{ss'}(a)(1 - P_{ss'}(a))}{(n-1)(1-\delta)}}$$

Theorem 1 (Part 2)

For any n, δ, and for $\delta \ll 1, n \gg 1$, the error scales as:

$$\left|\hat{P}_{ss'}(a) - P_{ss'}(a)\right| \sim \mathcal{O}\left(\sqrt{\delta/n}\right) + \mathcal{O}(1/n)$$

Key insight: Any agent capable of generalizing to multi-step goal-directed tasks must have learned a predictive model of its environment, which can be extracted from the agent's policy.

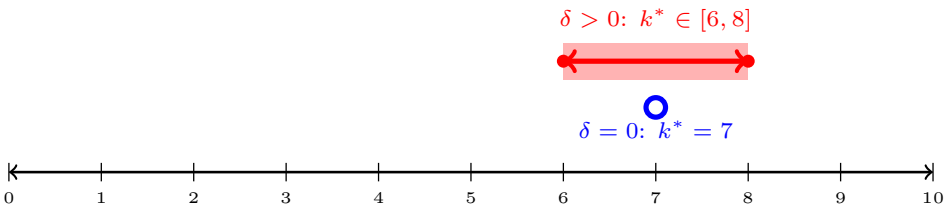
Intuition Building: Optimal (Def.4) VS Imperfect Agents

Reality: You are not an optimal policy and might make mistakes

- Tired, distracted, miscalculated
- Pick suboptimal challenge sometimes

Definition 5: You achieve $\geq (1-\delta)$ of optimal probability

Effect: Switch point gets "blurry"



Theorem 1 Guarantee:

$$\left|\hat{P} - P\right| \leq \sqrt{\frac{2P(1-P)}{(n-1)(1-\delta)}}$$

Implications:

- More trials ($n \uparrow$) $\rightarrow$ less error
- Better agent ($\delta \downarrow$) $\rightarrow$ less error
- Error scales as $O(\sqrt{\delta/n})$

Example:
$\delta = 0.2$, $n = 100$:
Error $\leq 7.9\%$

Mathematical Thinking: Core Proof Strategy

Key Techniques:

1. **Constructive Identifiability**
   - Algorithm 1 constructs world model
   - Algorithm existence = proof of encoding
2. **Adversarial Goal Design**
   - Create either-or decisions
   - Agent as binomial oracle
   - Mutual exclusivity reveals info
3. **Switch Point = Median**
   - Agent switches where $P(\leq k) \approx P(> k)$
   - Locates binomial median

Mathematical Thinking: Core Proof Strategy

Key Techniques:

1 Constructive Identifiability

- Algorithm 1 constructs world model
- Algorithm existence = proof of encoding

2 Adversarial Goal Design

- Create either-or decisions
- Agent as binomial oracle
- Mutual exclusivity reveals info

3 Switch Point = Median

- Agent switches where $P(\leq k) \approx P(> k)$
- Locates binomial median

The Proof Flow:

1 Goal Factorization

- Decompose sequential goals (Lemmas 3-5)
- Exploit Markov property
- Reduce to binomial trials

2 Action-Space Extension

- Add "teleport" action (Lemma 2)
- Simplifies optimal policy analysis
- Results transfer to original space

Mathematical Thinking: Core Proof Strategy

Key Techniques:

1 Constructive Identifiability

- Algorithm 1 constructs world model
- Algorithm existence = proof of encoding

2 Adversarial Goal Design

- Create either-or decisions
- Agent as binomial oracle
- Mutual exclusivity reveals info

3 Switch Point = Median

- Agent switches where $P(\leq k) \approx P(> k)$
- Locates binomial median

The Proof Flow:

1 Goal Factorization

- Decompose sequential goals (Lemmas 3-5)
- Exploit Markov property
- Reduce to binomial trials

2 Action-Space Extension

- Add "teleport" action (Lemma 2)
- Simplifies optimal policy analysis
- Results transfer to original space

Core Innovation: Constructive Necessity Proof of Identifiability

Assume agent has capability X (bounded regret δ for depth-n goals)
$\rightarrow$ Design Algorithm 1 that constructs world model $\hat{P}_{ss'}(a)$ from policy
$\rightarrow$ Therefore: achieving capab. X *necessarily requires* learning a world model.
The algorithm's existence constructively proves the world model is identifiable from the agent's policy.

Algorithm 1: Overview

EstimateTransitionProbability(π, s, a, s', n, b)

Inputs:

- Goal-conditioned policy $\pi(a_t|h_t;\psi)$
- State s, action a, outcome s'
- Precision parameter $n \in \mathbb{N}$ (related to max goal depth $2n+1$)
- Alternative action $b \neq a$

Output: Estimate $\hat{P}_{ss'}(a)$ of transition probability

Method: Query the agent with composite goals $\psi_{a,b}(k, n)$ for increasing k, observing when the agent switches from preferring action a to action b.

Mapping Intuition to Algorithm 1

Algorithm 1:

1. For $k = 0, 1, \ldots, n$:
   - Construct $\psi_a(k, n)$: "$\leq k$ successes"
   - Construct $\psi_b(k, n)$: "$> k$ successes"
   - Query: $a_0 = \pi(a_0 | s_0; \psi_a \vee \psi_b)$
2. Find switch: k^* where a_0 changes from b to a
3. Estimate: $\hat{P}_{ss'}(a) = (k^* - 1/2)/n$

Key: Agent's initial action reveals its world model!

Mapping Intuition to Algorithm 1

Basketball Analogy:

- You = Agent policy π
- Teammate skill = $P_{ss'}(a)$
- Your training = Agent learning
- One shot = State transition
- Bounce pass ball = Action a (φ_1)
- Lob pass ball = Action b (φ_1')
- n shots = Goal depth
- Me = Experimentor extracting your internal world model
- Switch point k^* = Reveals $P_{ss'}(a)$

Algorithm 1:

1. For $k = 0, 1, \ldots, n$:
   - Construct $\psi_a(k, n)$: "$\leq k$ successes"
   - Construct $\psi_b(k, n)$: "$> k$ successes"
   - Query: $a_0 = \pi(a_0 | s_0; \psi_a \vee \psi_b)$
2. Find switch: k^* where a_0 changes from b to a
3. Estimate: $\hat{P}_{ss'}(a) = (k^* - 1/2)/n$

Key: Agent's initial action reveals its world model!

Algorithm 1: Pseudocode (Part 1) I

function EstimateTransitionProbability(π, s, a, s', n, b)
    Initialize $k^* \leftarrow n$
    for $k = 1$ to n **do**
        Define base LTL components:
        $\varphi_0 \leftarrow [A_0 = a]$ ▷ Take action a
        $\varphi_0' \leftarrow [A_0 = b]$ ▷ Take action b
        $\varphi_1 \leftarrow \Diamond[A = a, S = s]$ ▷ Eventually to state s, action a
        $\varphi_2 \leftarrow \bigcirc[S = s']$ ▷ Next to state s'
        $\varphi_2' \leftarrow \bigcirc[S \neq s']$ ▷ Next to state $\neq s'$

        Define composite goal:
        $\psi_0 \leftarrow \langle \varphi_1, \varphi_2' \rangle$ ▷ Sequential goal (Fail)
        $\psi_1 \leftarrow \langle \varphi_1, \varphi_2 \rangle$ ▷ Sequential goal (Success)

Algorithm 1: Pseudocode (Part 2) I

$\psi_a(k,n) \leftarrow \bigvee_{\text{sequences with } r \leq k \text{ successes}} \langle \varphi_0, (\psi_0 \text{ or } \psi_1) \times n \rangle$

$\psi_b(k,n) \leftarrow \bigvee_{\text{sequences with } r > k \text{ successes}} \langle \varphi'_0, (\psi_0 \text{ or } \psi_1) \times n \rangle$

$\psi_{a,b}(k,n) \leftarrow \psi_a(k,n) \vee \psi_b(k,n)$

$a_0 \leftarrow \pi(a_0 | s_0; \psi_{a,b}(k,n))$ ▷ Query the policy

if $a_0 = a$ **then**

  $k^* \leftarrow k$

  break ▷ Found smallest k where agent prefers $\leq k$ successes

end if

end for

Estimate $\hat{P}_{ss'}(a) \leftarrow (k^* - 1/2)/n$

return $\hat{P}_{ss'}(a)$

end function

Summary

Basketball Analogy $\rightarrow$ General AI Agents

- **Shooting accuracy** = Environment transition probabilities $P_{ss'}(a)$, where:
  - s is the state where your teammate is at the 3pt line ;
  - a is the action of shooting the ball to the hoop ;
  - s' is the state where the ball goes through the hoop ;
- **Challenge choices** = Goal-conditioned policy queries $\pi(a_t|h_t;\psi)$
- **Switch point reveals skill** = Agent behavior reveals world model

Theorem 1: Main Result

Any agent capable of achieving multi-step goals (bounded regret for depth-n goals) must have learned a world model accurate enough to support this behavior.
The world model can be extracted via Algorithm 1 (the challenge procedure).

Philosophical point: There is no "model-free" shortcut: Achieving complex goals *requires* learning how the world works.

Outline

1 Background & Overview

2 Environment & Assumption

3 Agents & Goals

4 Building Intuition

5 Theorem 1 & Algorithm 1

6 Limitations

Limitations and Future Directions

What the paper does NOT cover:

1. **Partial observability**
   - Only fully observed states
   - POMDPs remain open
2. **Infinite/continuous spaces**
   - Finite S, A assumed
   - Real robotics has $\mathbb{R}^n$ states
3. **Non-Markovian dynamics**
   - History-dependent transitions
   - Memory/context effects
4. **Agent's internal use**
   - Proves model exists
   - Not how agent uses it (n-step planning?)

Practical considerations:

- **Query Complexity:**
  - $O(n \cdot |S| \cdot |A|)$ queries
  - Expensive for large spaces
- **Real Agents:**
  - May fail on some goals ($\delta = 1$)
  - Average regret more realistic
- **Goal Specification:**
  - LTL formulation required
  - Agent must parse goals

Future work:

- Extend to POMDPs (latent states)
- Continuous state/action spaces
- Causal world models
- Scalable extraction algorithms

Thank you for your attention!
Any Questions?