# Deep Learning

Randall Balestriero

Lecture 25: Model-Based RL

# Recap!

# Origins of Model-Based RL: The Key Lineages

OPTIMAL CONTROL (1950s)
Richard Bellman

$f(s, a)$ = Perfect Physics

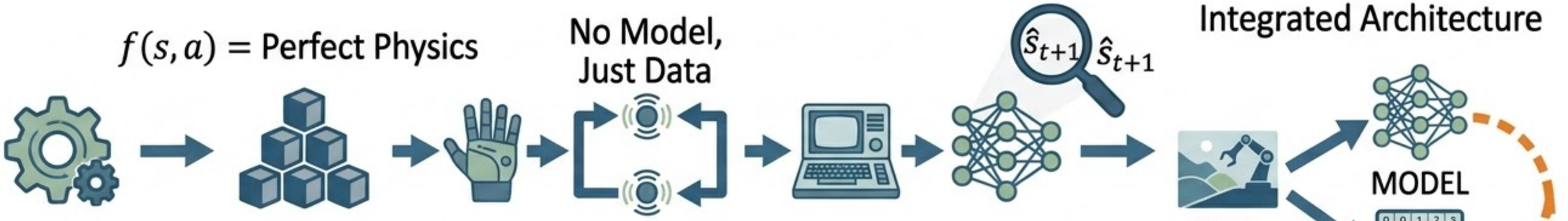
TRIAL-AND-ERROR
AI Pioneers

No Model,
Just Data

STeLLA (1963)
John Andreae

DYNA-Q (1990)
Richard Sutton

Integrated Architecture




OPTIMAL CONTROL (1950s)
Richard Bellman

DYNAMICAL MODELS & DP
$f(s, a)$ = Perfect
$V(s)$ = DP

MODEL-FREE LEARNING
No Model, Just Data
Policy Gradient

LEARNING AN INTERNAL MODEL
To handle hidden states and "think ahead".

IMAGINARY EXPERIENCE
Simulated Rollouts

Key Driver: SAMPLE EFFICIENCY. Use models to "practice" and learn from imaginary data when real data is expensive or dangerous.

# Notations

# Notations

- On-policy and off-policy

# Notations

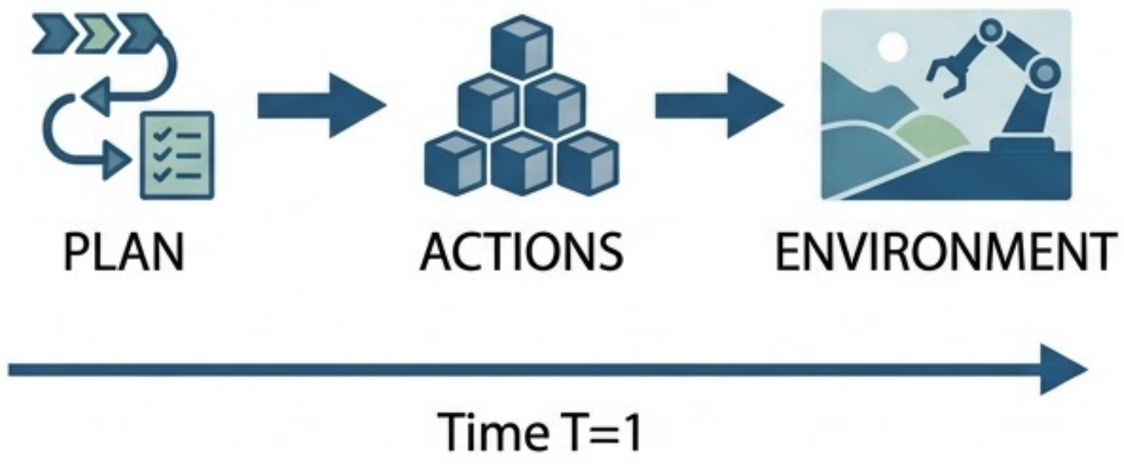
- On-policy and off-policy
- Model-free and model-based

# Notations

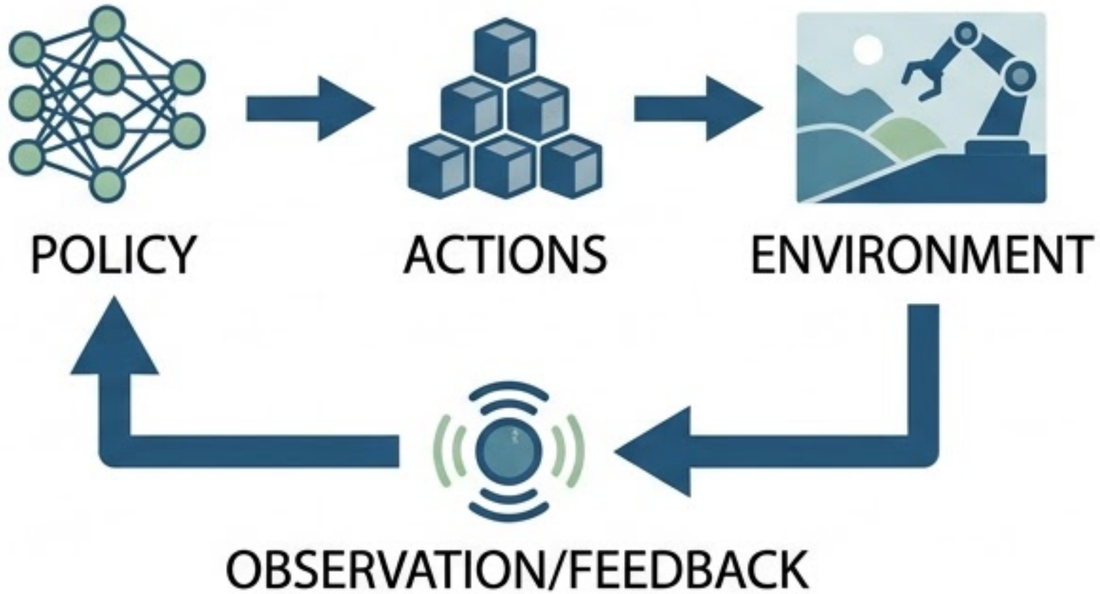
- On-policy and off-policy
- Model-free and model-based
- Open-loop and closed-loop

# Open-Loop vs. Closed-Loop Control

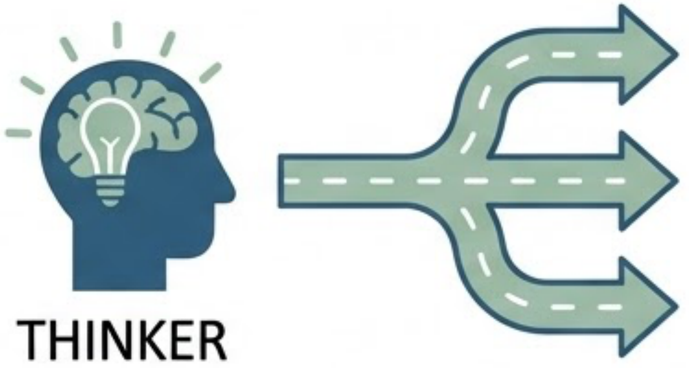
## Open-Loop Control




## Closed-Loop Control



# Why Use Open-Loop Planning?




## Computational Efficiency

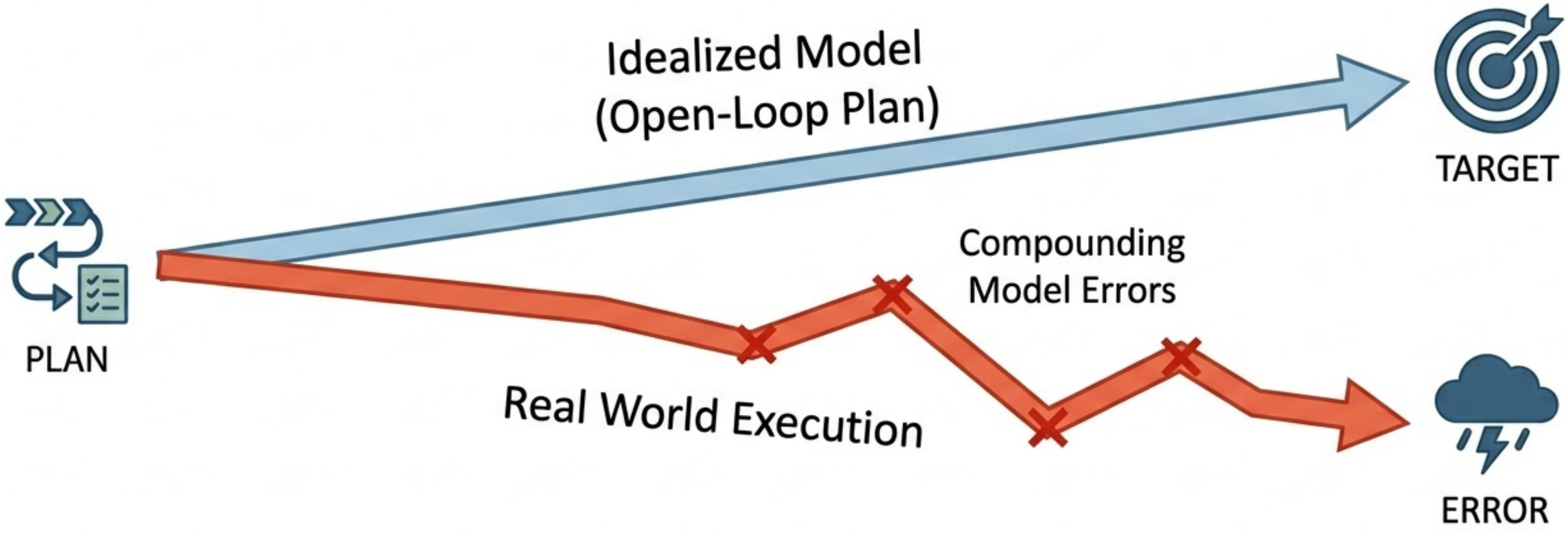
Faster and cheaper to optimize a single trajectory of actions than a complex policy.

## Foundation for Closed-Loop


Essential component for Model Predictive Control (MPC): We plan open-loop but execute closed-loop.

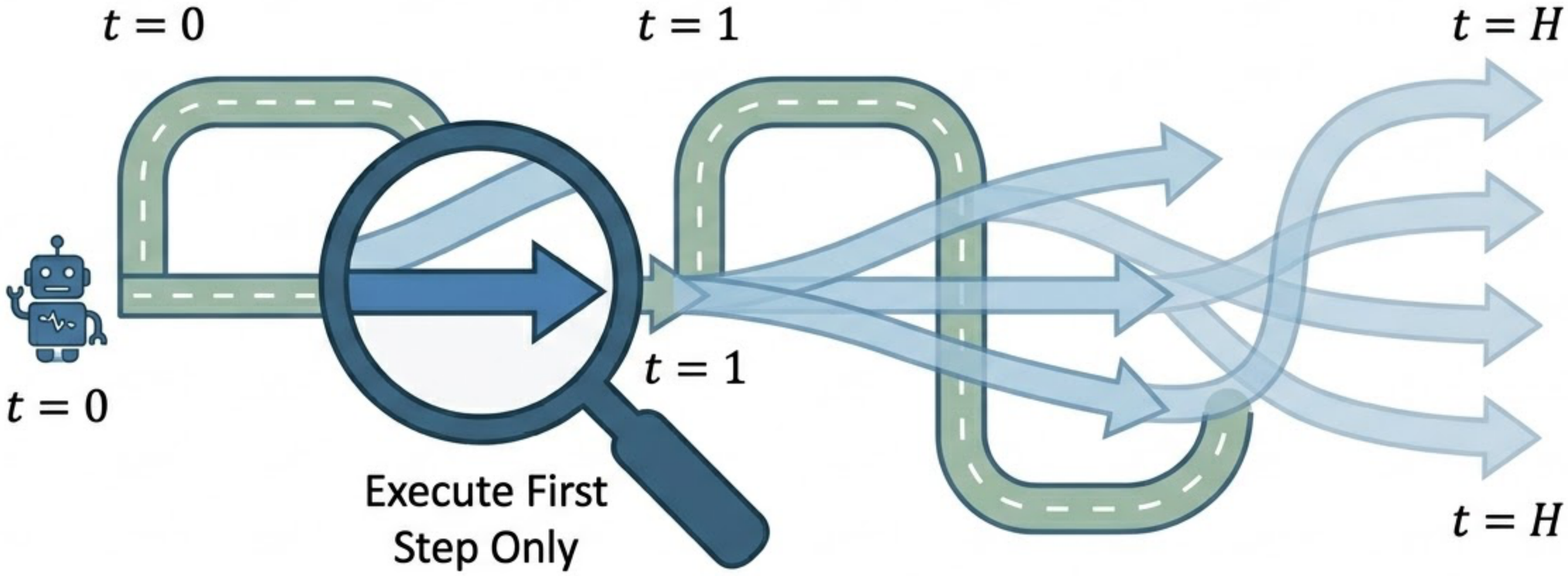
# But still… why not open-loop?

# The Problem: Model Drift & The Reality Gap




Small prediction errors at the start lead to massive failures over long horizons without feedback.

# Model Predictive Control (MPC): The Receding Horizon

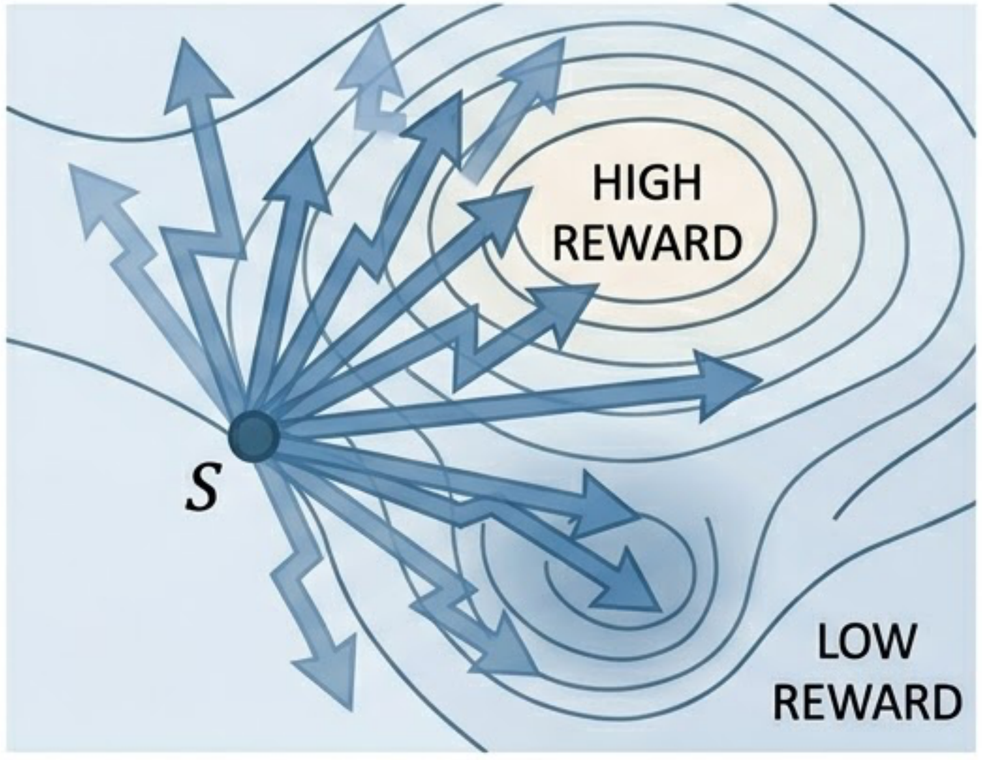



[Plan $t = 0 \ldots H$] → [Execute $a_0$] → [Observe $s_1$] → [Re-Plan $t = 1 \ldots H + 1$]

# How to generate candidate actions?

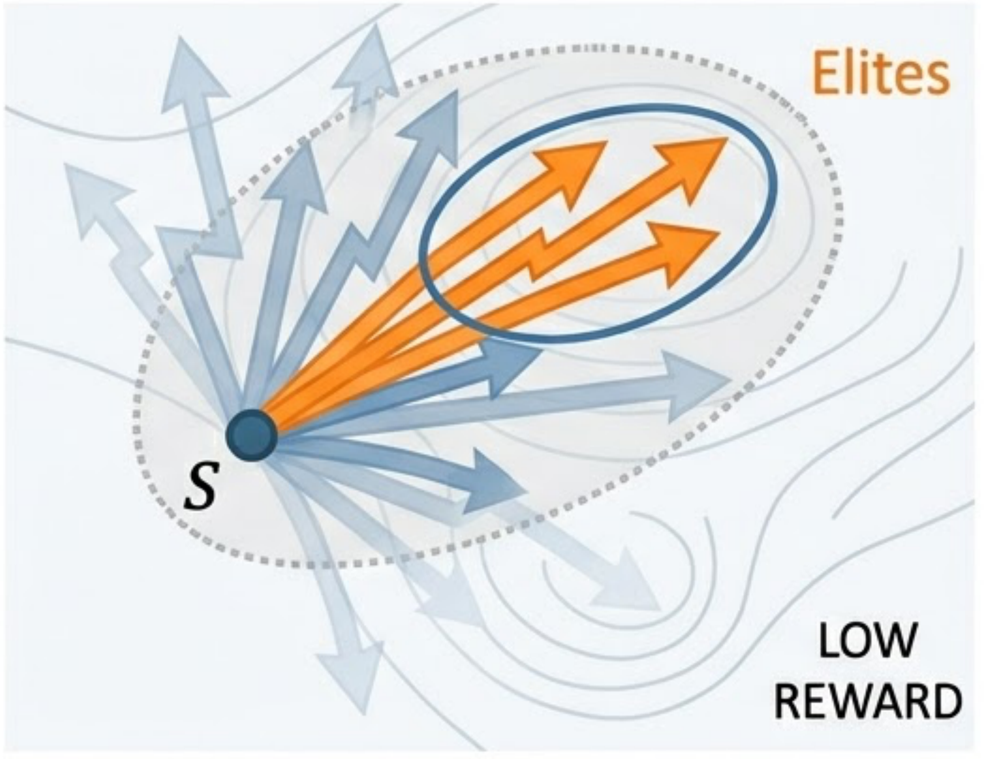
# Optimization for Planning: Shooting Methods

## Random Shooting




Sample $N$ sequences, pick the best.

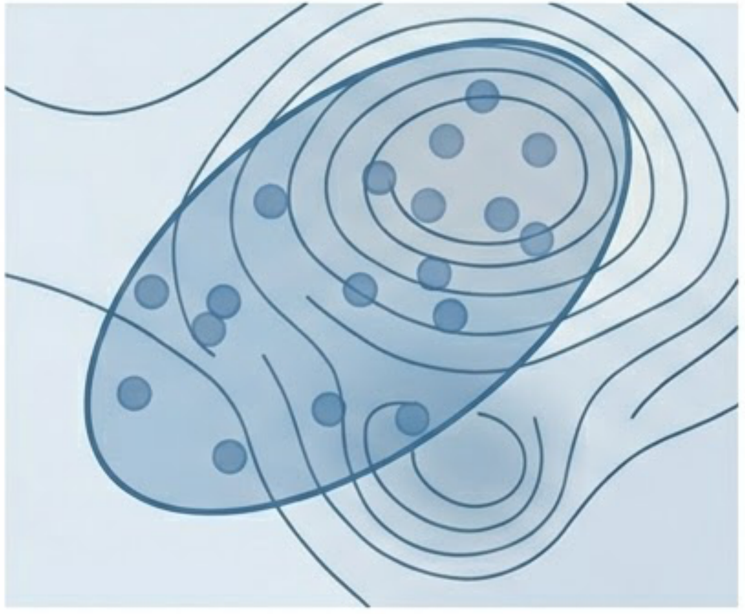
## Cross-Entropy Method (CEM)




Iteratively refit distribution to elite samples.

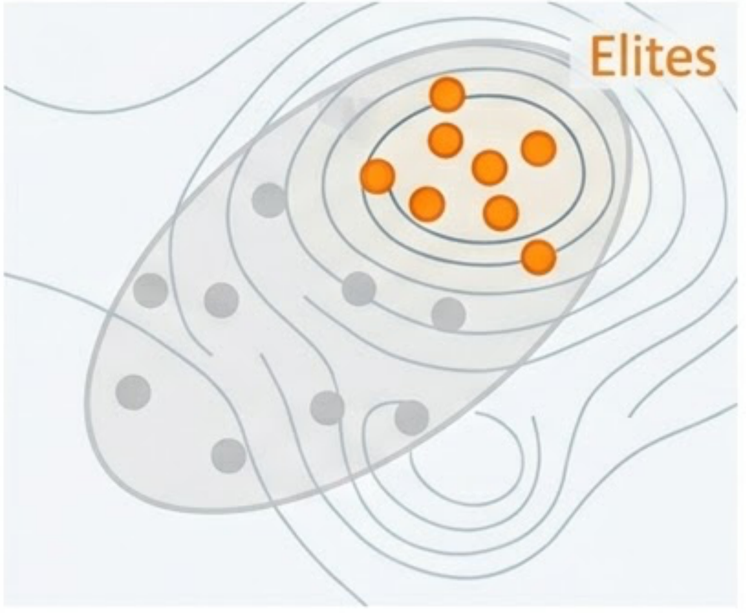
# The Cross-Entropy Method (CEM) Algorithm

## 1. Sample and Evaluate



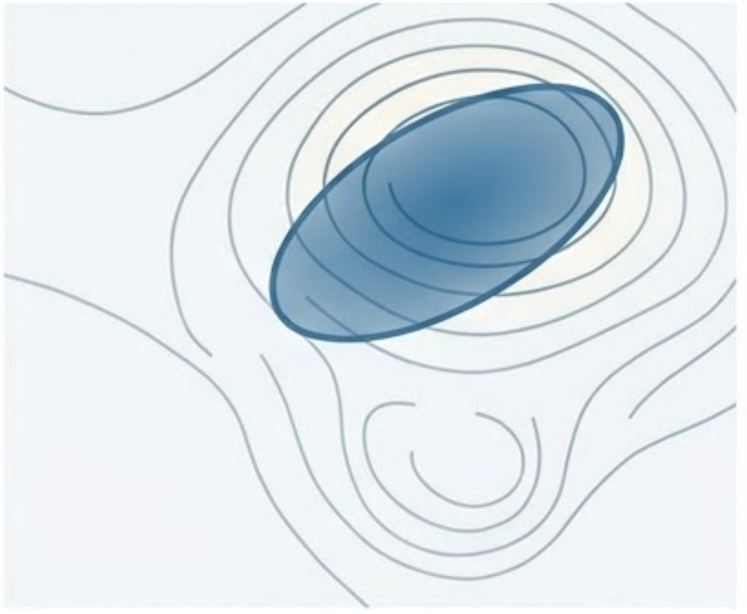
Sample $N$ trajectories from initial distribution $P(A)$.

## 2. Select Elites



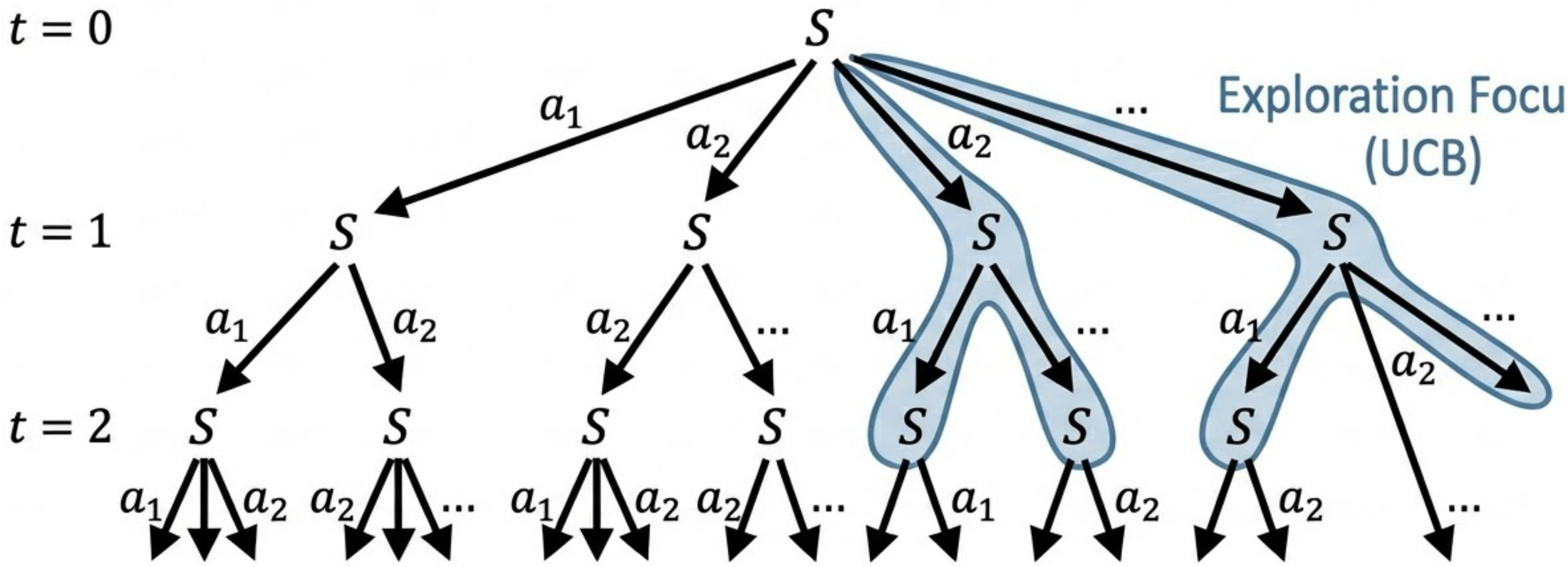

Keep top $k$ (e.g., 10%) highest-reward samples.

## 3. Refit and Resample



Re-calculate parameters of $P(A)$ (e.g., mean and variance) to fit the Elites. Repeat.

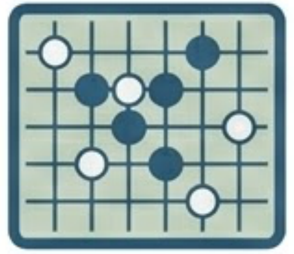
# Monte Carlo Tree Search (MCTS): Planning with Discrete Actions



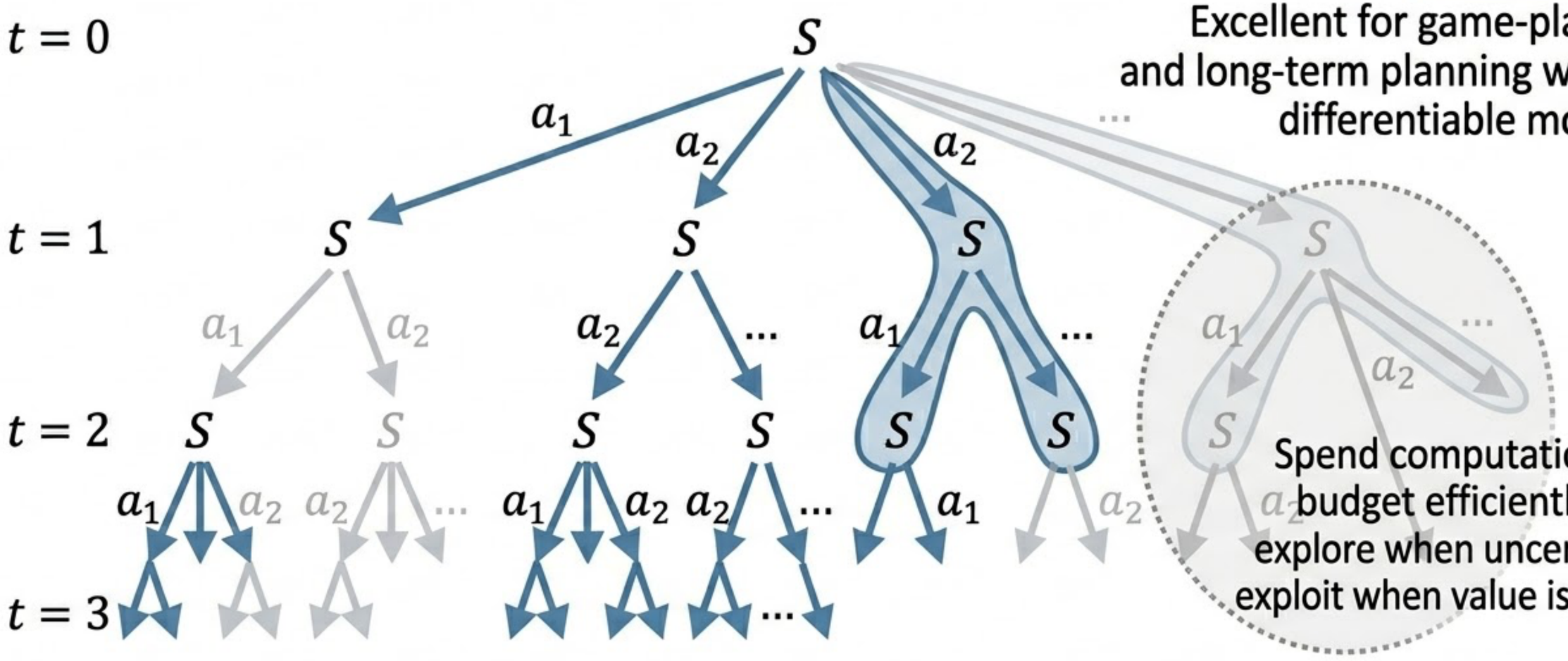

Best for discrete actions and long horizons.

Dynamically builds an asymmetric search tree using a simulation model.

# Why Monte Carlo Tree Search (MCTS)?



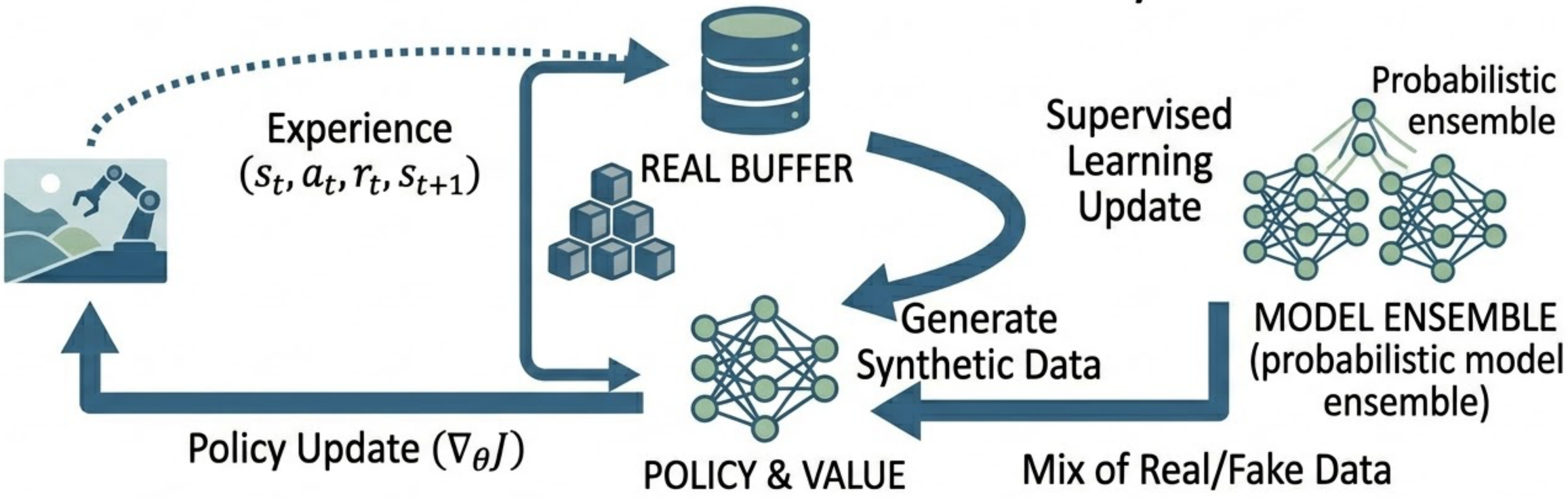
Excellent for game-playing and long-term planning without differentiable models.




Spend computational budget efficiently: explore when uncertain, exploit when value is high.

Asymmetric growth: focus search on promising paths.

# The MBPO Algorithm Loop




Hybrid approach: Learn a model, hallucinate short rollouts from real states, and train policy on augmented data.

# Fundamental question: What is s_t?

See you on Monday!