

Quasimetric Reinforcement Learning

Tongzhou Wang @ DLCT

ICLR 2022

NeurReps Workshop @ NeurIPS 2022

ICML 2023

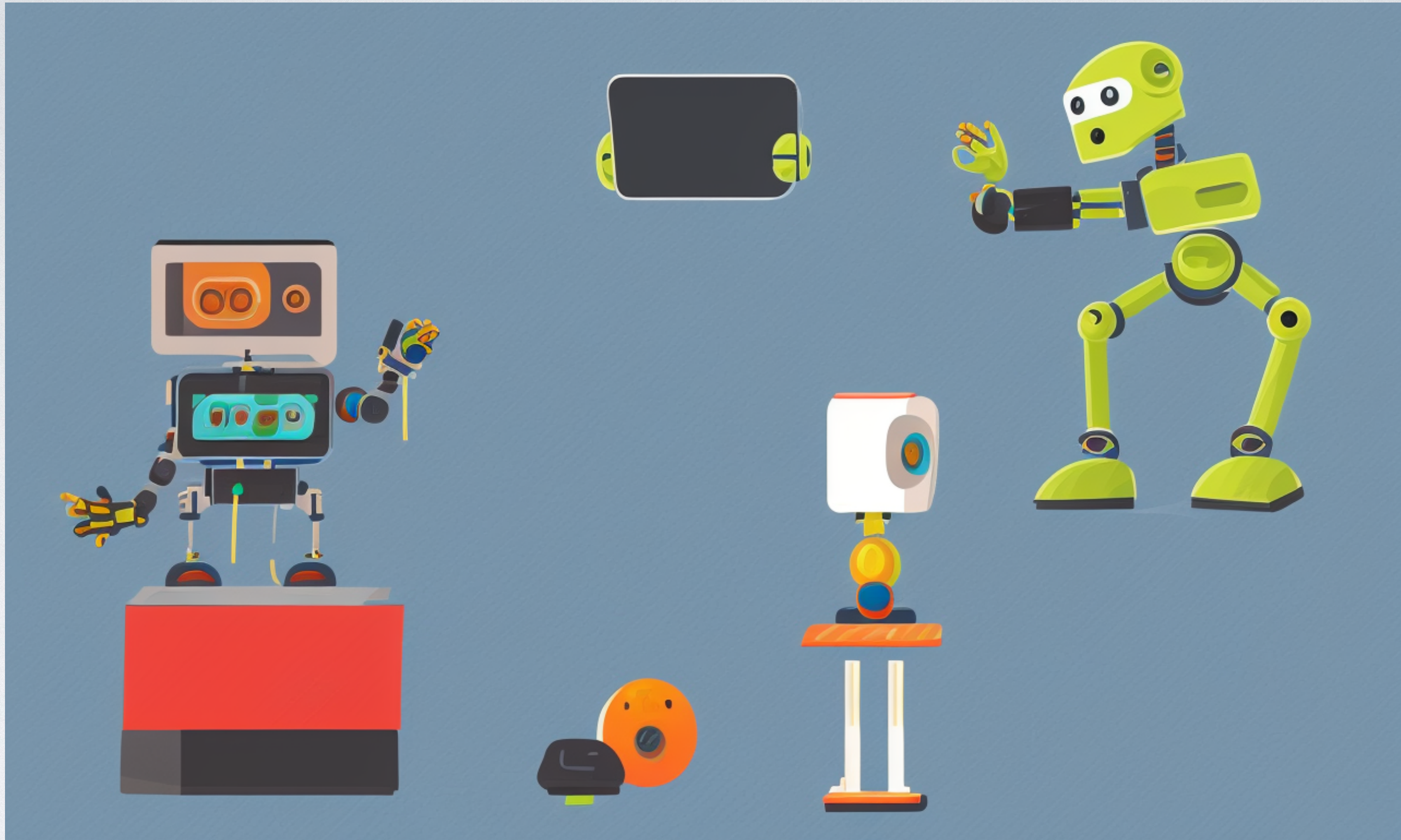
Agenda

Motivation Generalist (Goal-Reaching) Agents

???

???

???



Generalist Agents

Image by Stable Diffusion 2.1 (Prompt = "a robot playing video games and a robot doing housework; high quality flat design")

Sparks of AGI?

General (Pre)training → **Generalist Agents** for General Task-Solving

(Multi-Task, Goal-Reaching, Instruction-Following)

Structure

Vision

View-based SSL → **Encoder** for feature extraction/FT

Language

Sequence Modeling → **LLM** for prompting/FT

Decision-Making

??? Training → **Agent that can do ???**

Sparks of AGI?

General (Pre)training → **Generalist Agents** for General Task-Solving
(Multi-Task, Goal-Reaching, Instruction-Following)

Structure

Vision

View-based SSL → **Encoder** for feature extraction/FT

Language

Sequence Modeling → **LLM** for prompting/FT

Decision-Making

Learning how to reach many goals → **Agent that can reach any goal**

Sparks of AGI?

General (Pre)training → **Generalist Agents** for General Task-Solving
(Multi-Task, Goal-Reaching, Instruction-Following)

Structure

Vision

View-based SSL → **Encoder** for feature extraction/FT

Language

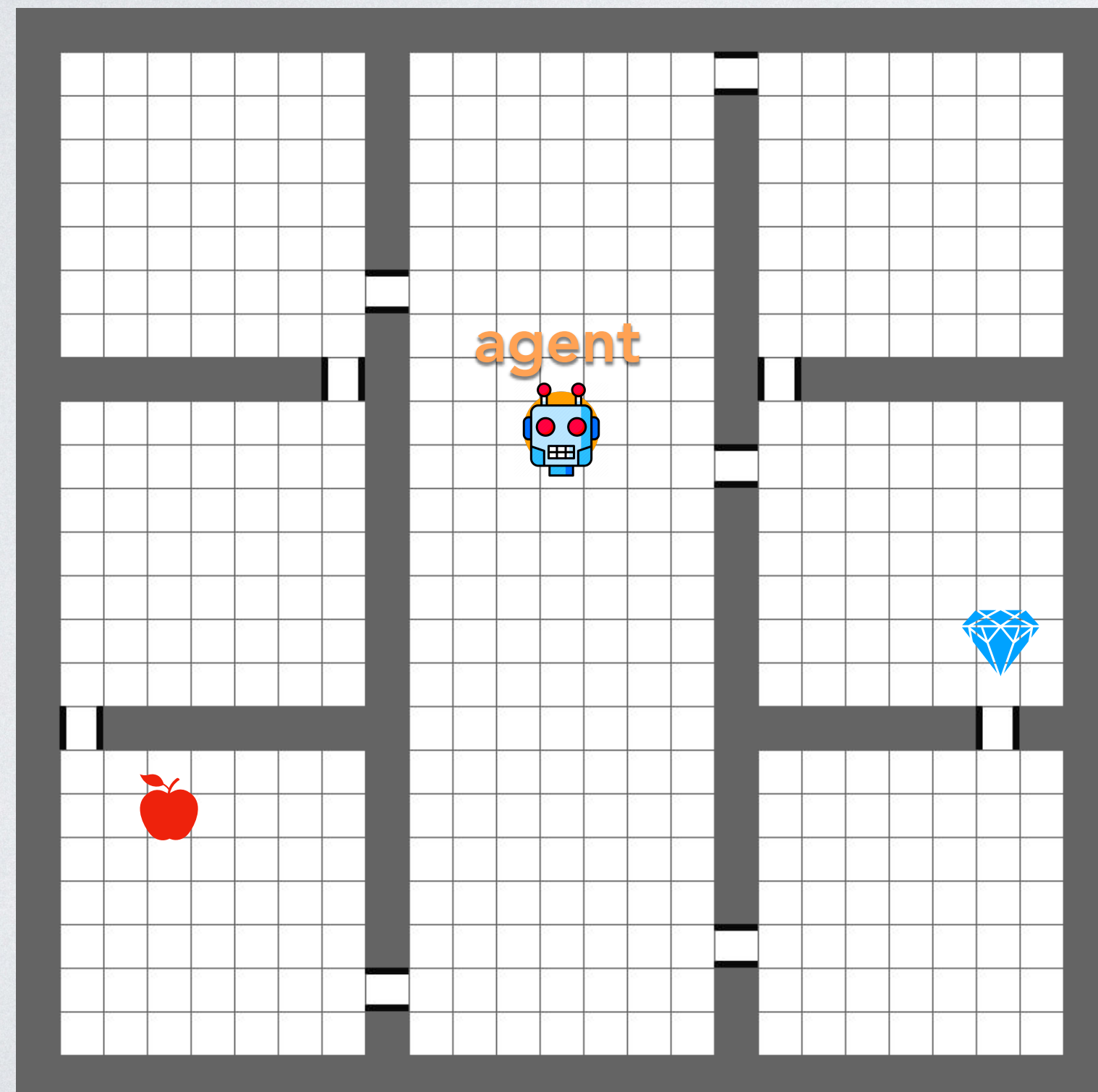
Sequence Modeling → **LLM** for prompting/FT

Decision-Making

Quasimetric → **Agent** that can reach any goal
core structure in multi-goal

Agents for Sequential Decision Making

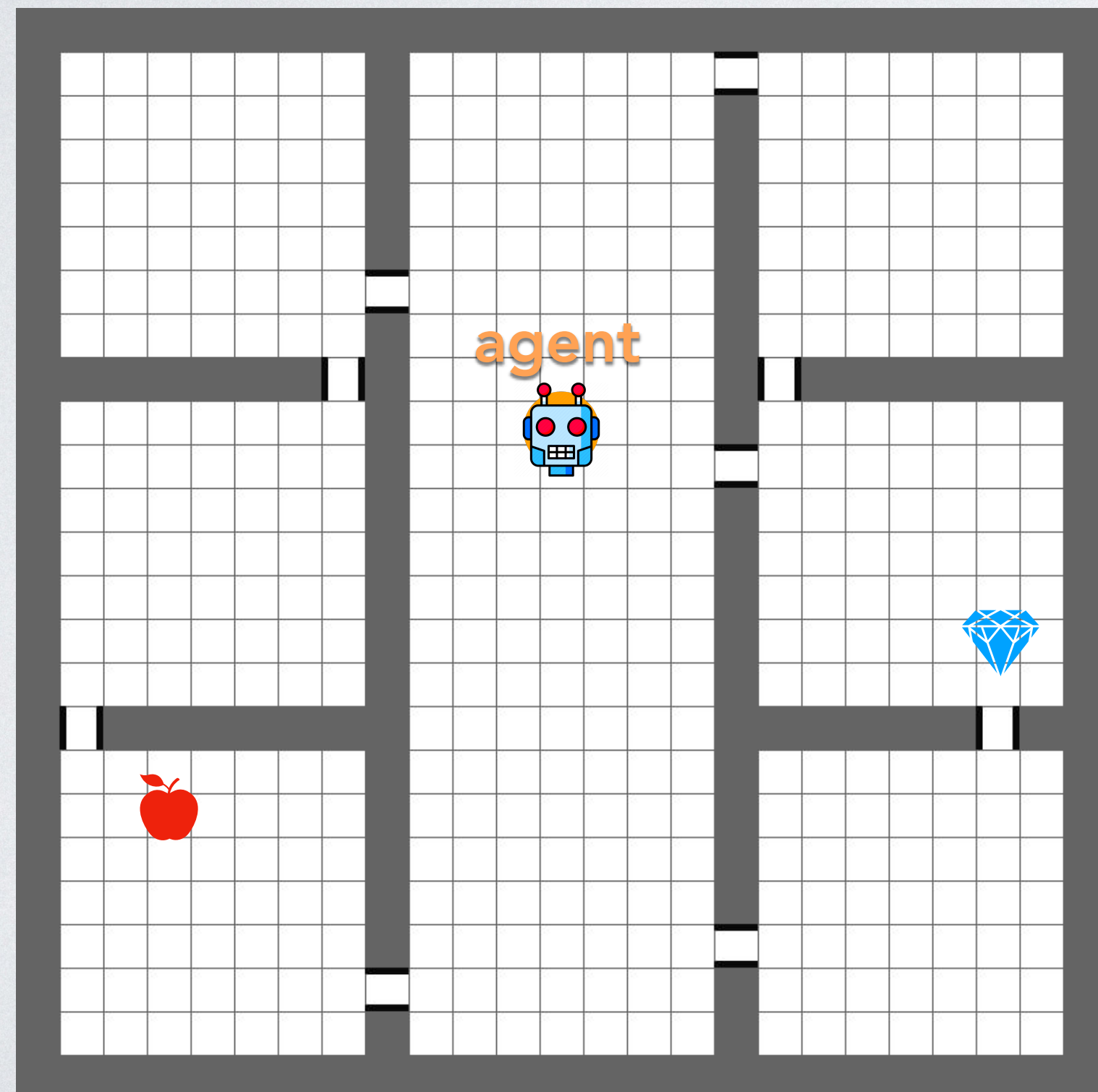
Control / Sequential decision making = Act at each **state** at each **timestep**



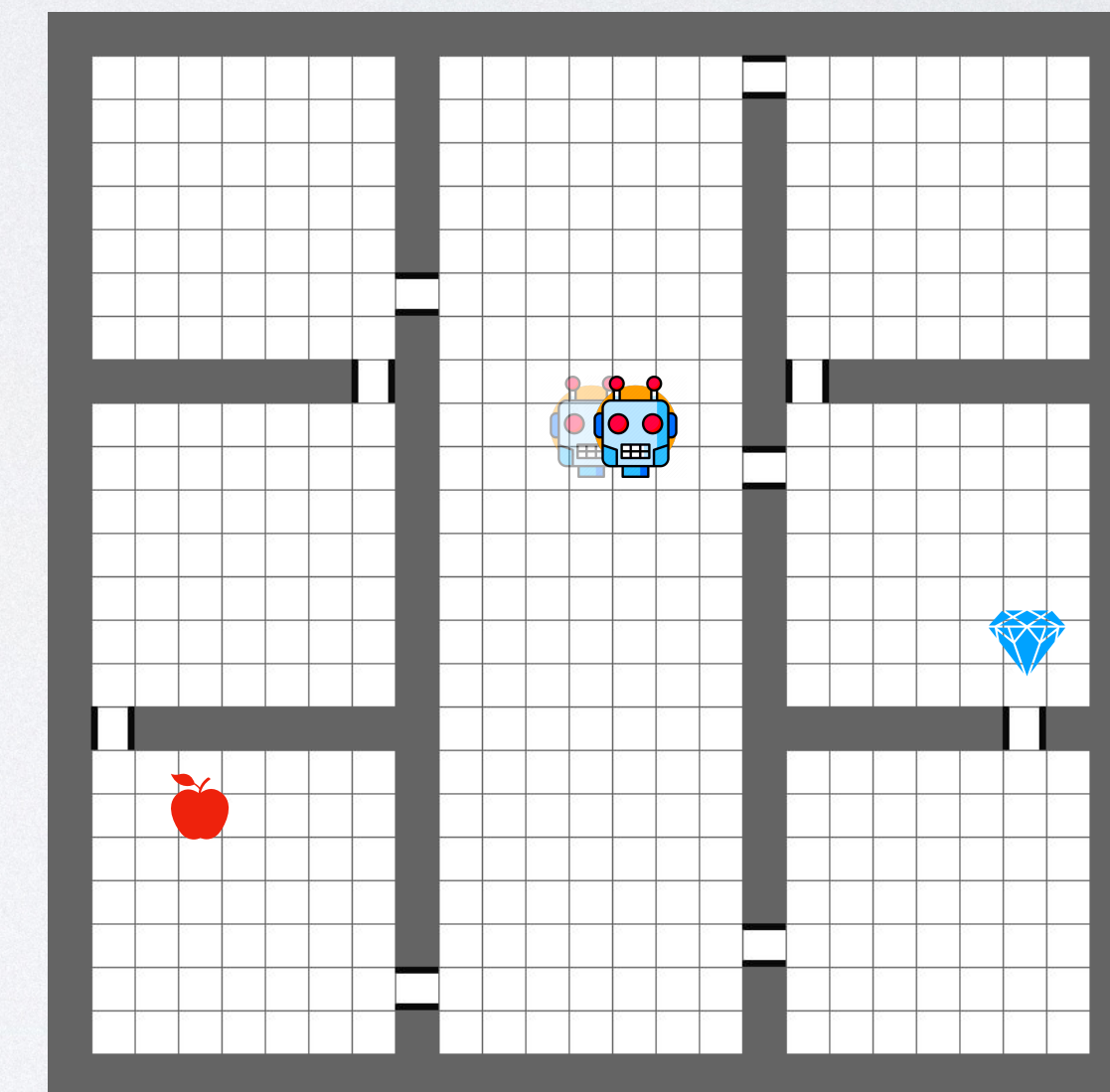
state = [12, 9]
action $\in \{ \leftarrow, \uparrow, \rightarrow, \downarrow \}$

Agents for Sequential Decision Making

Control / Sequential decision making = Act at each **state** at each **timestep**



state = [12, 9]
action $\in \{ \leftarrow, \uparrow, \rightarrow, \downarrow \}$

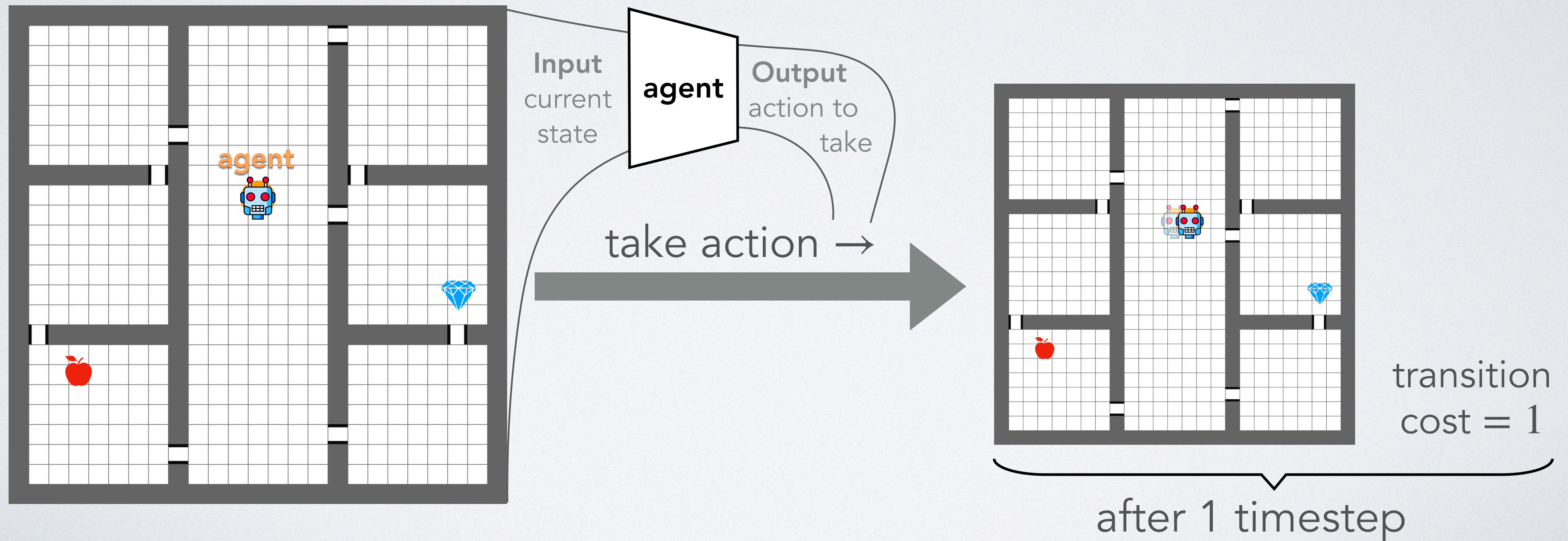


transition
cost = 1

after 1 timestep

Agents for Sequential Decision Making

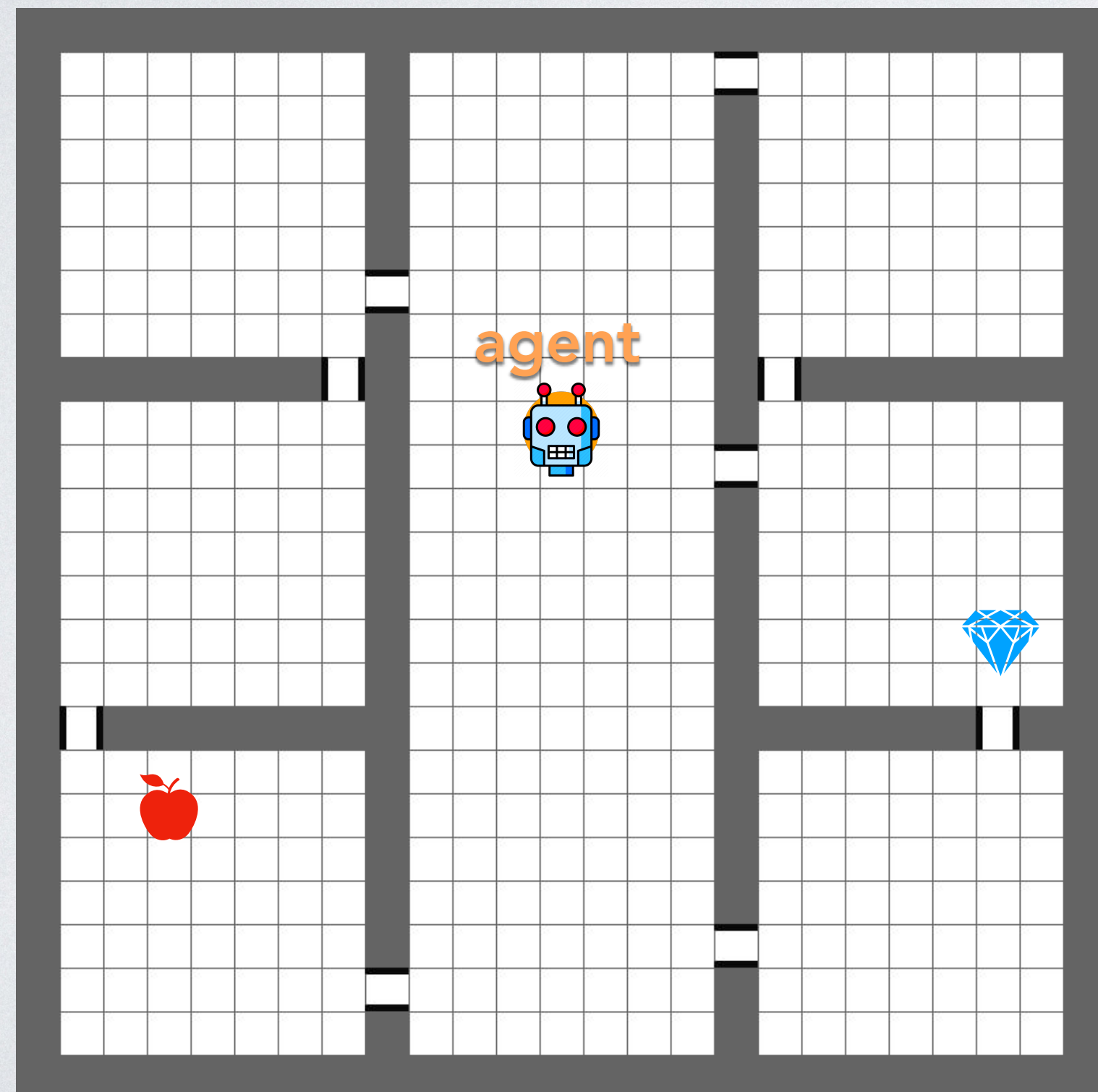
Control / Sequential decision making = Act at each **state** at each **timestep**



Generalist Goal-Reaching Agents

Control / Sequential decision making = **Act** at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)



Given ANY goal, e.g., 

From any starting state s_0 , optimal agent should give a "shortest path" $s_0 \longrightarrow$ 

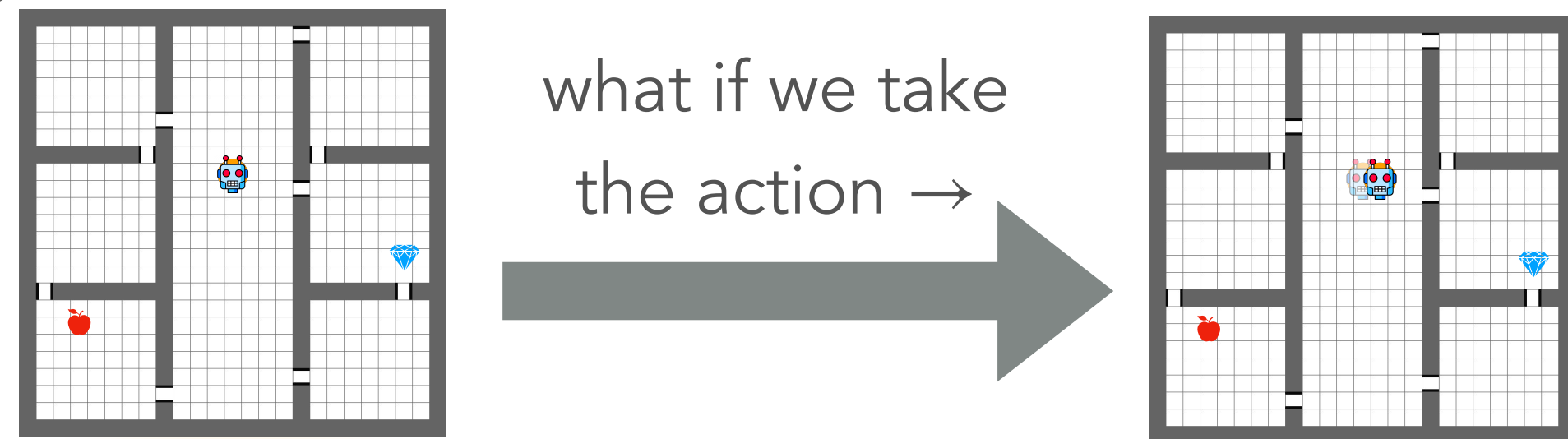
How to make (sequence of) decisions?

Generalist Goal-Reaching Agents

Control / Sequential decision making = **Act** at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)

Model-Based Agent



Learn a model of the world
Plan w.r.t. world model

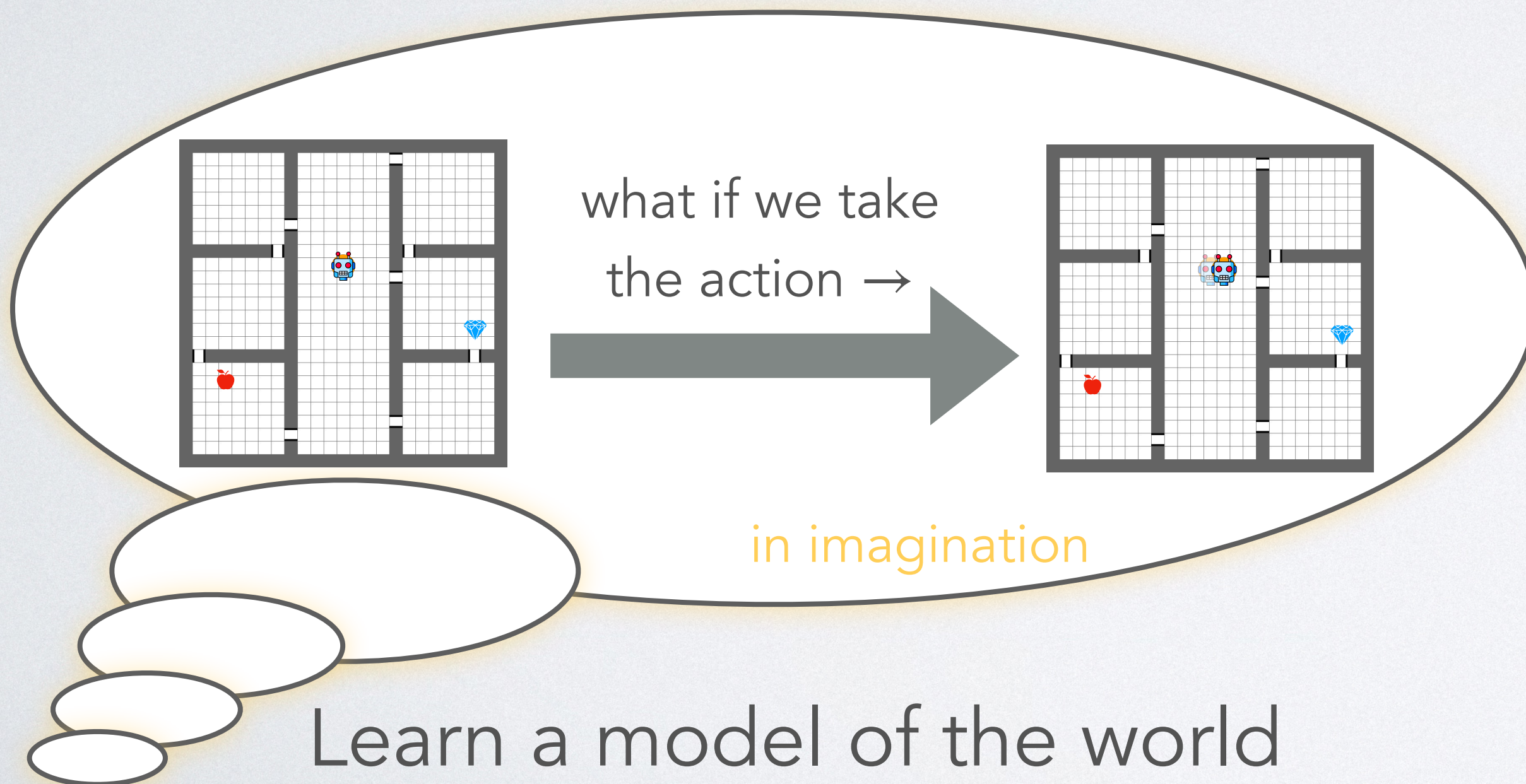
Value-Based Agent

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

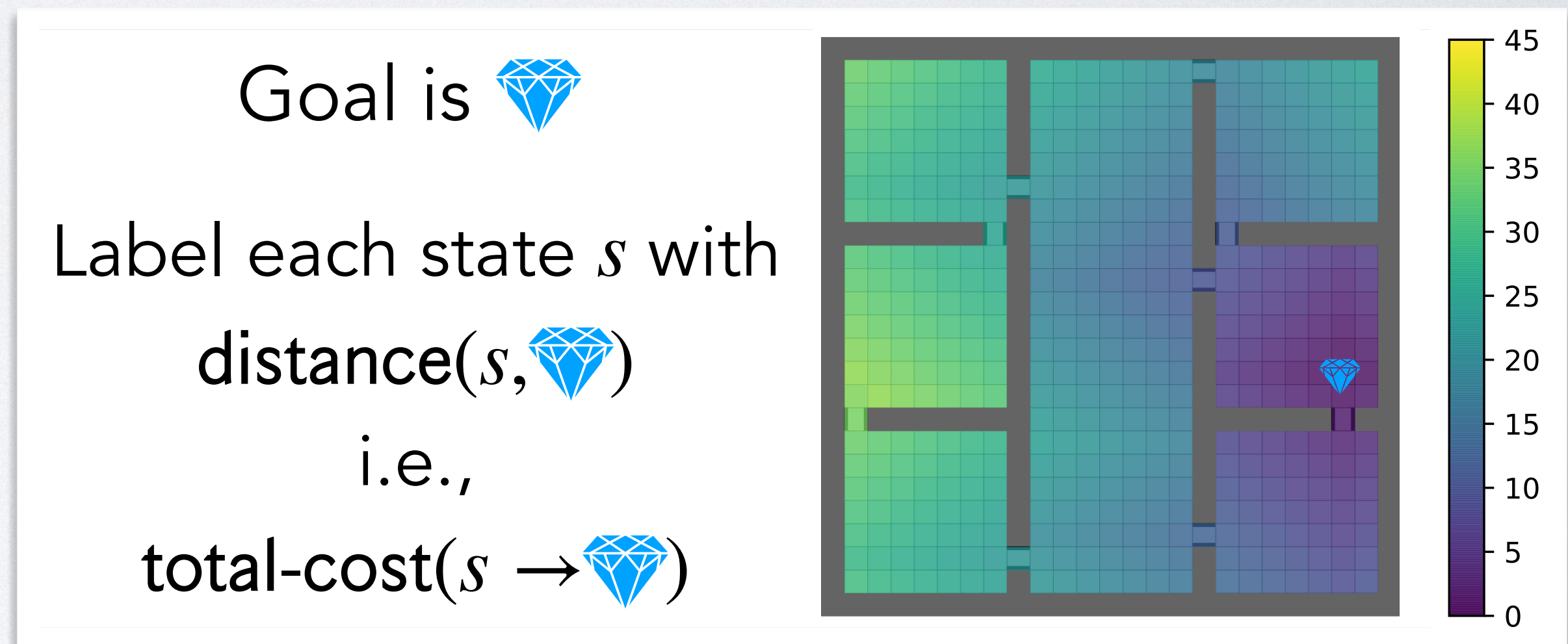
Generalist agents can do **everything** in an environment (e.g., entire house/city)

Model-Based Agent



Learn a model of the world
Plan w.r.t. world model

Value-Based Agent



Pick action that leads to smaller

$\text{distance}(\text{next state}, \text{goal})$

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)

Model-Based Agent

- + Easy to optimize, learn from local transitions
- + Easy to add known structure (e.g., +objects)
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation

Learn a model of the world
Plan w.r.t. world model

Value-Based Agent

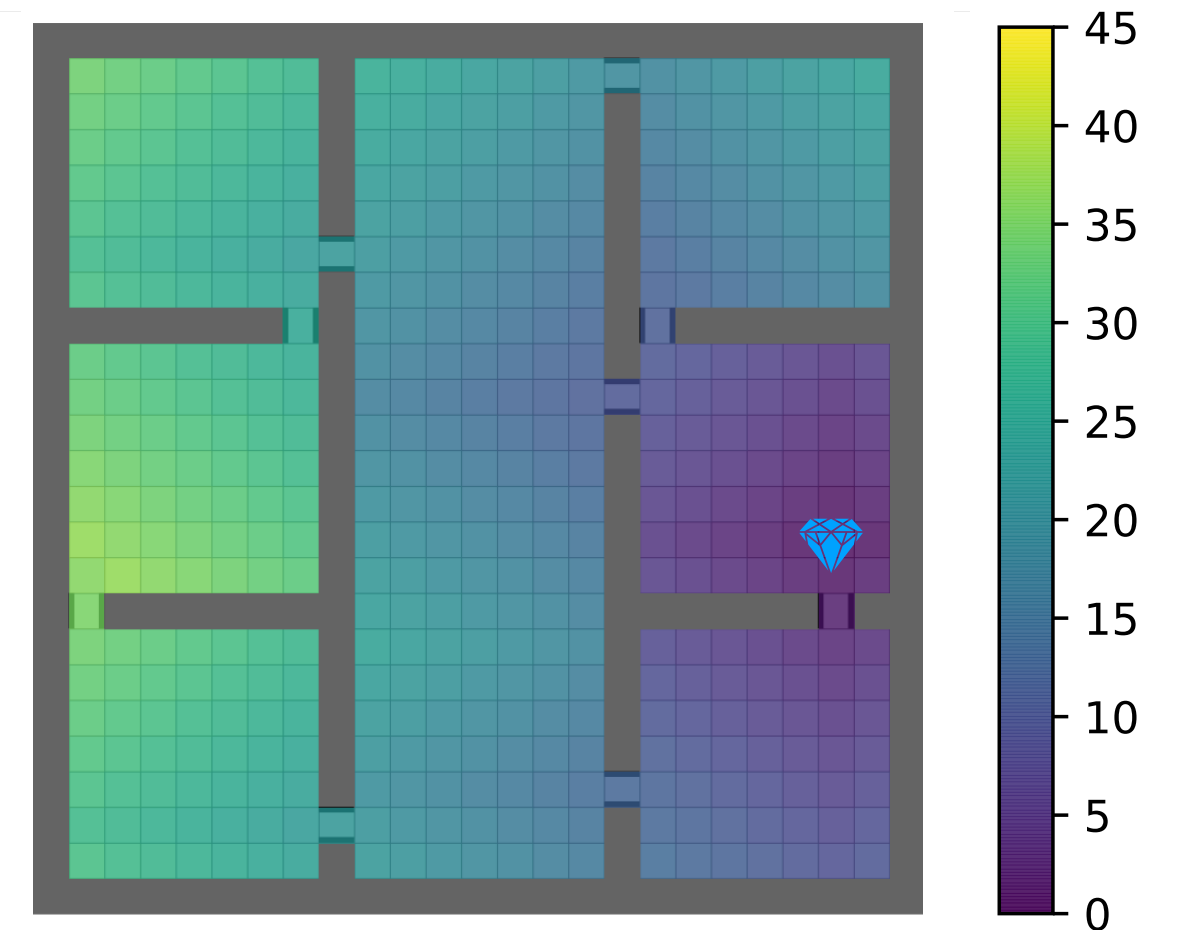
Goal is 

Label each state s with

$\text{distance}(s, \text{diamond})$

i.e.,

$\text{total-cost}(s \rightarrow \text{diamond})$



Pick action that leads to smaller
 $\text{distance}(\text{next state}, \text{diamond})$

Generalist agents

Control / Sequence
Generalist agent

Model-Based

- + Easy to optimize, learn
- + Easy to add known skills
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation

Learn a model of the world
Plan w.r.t. world model

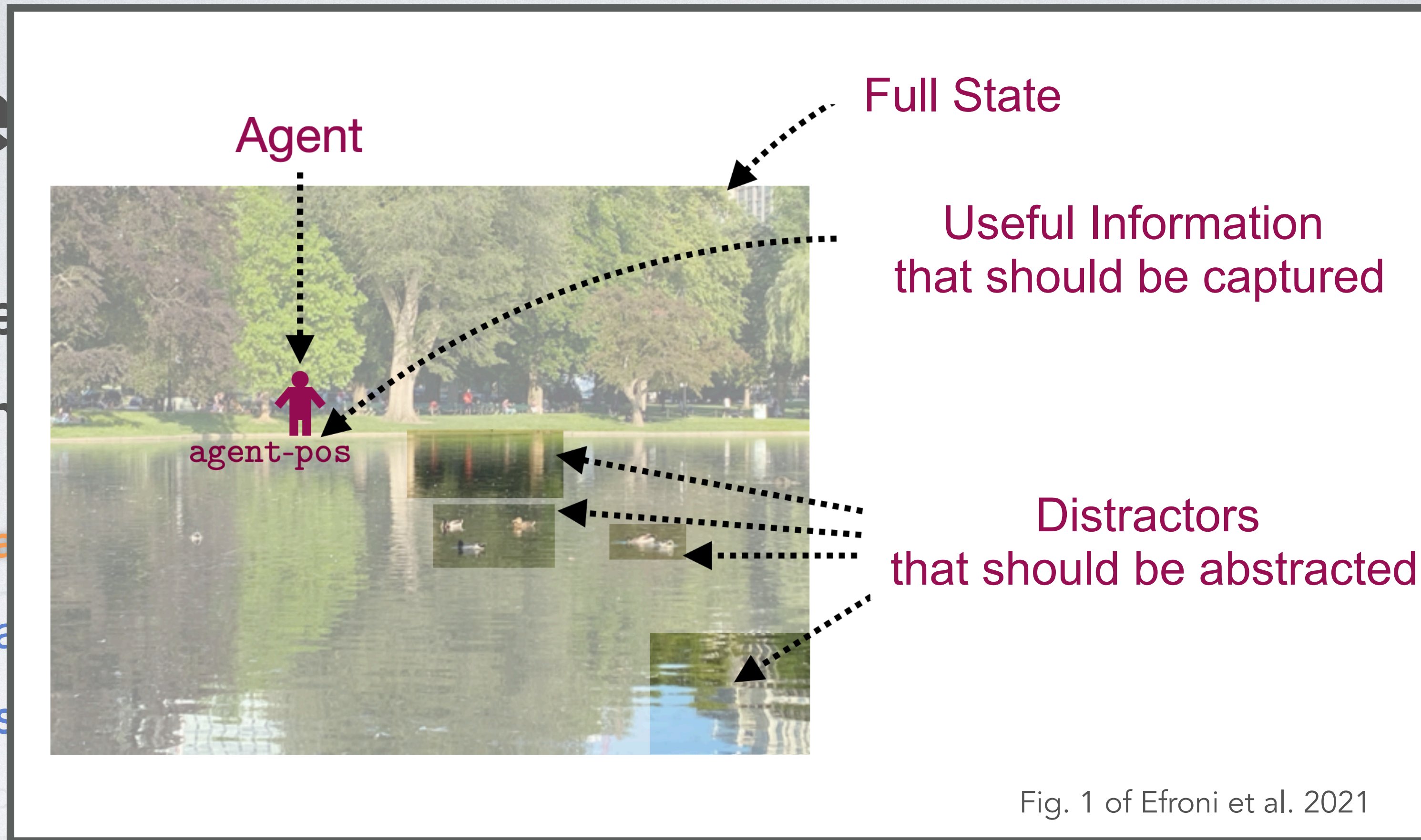
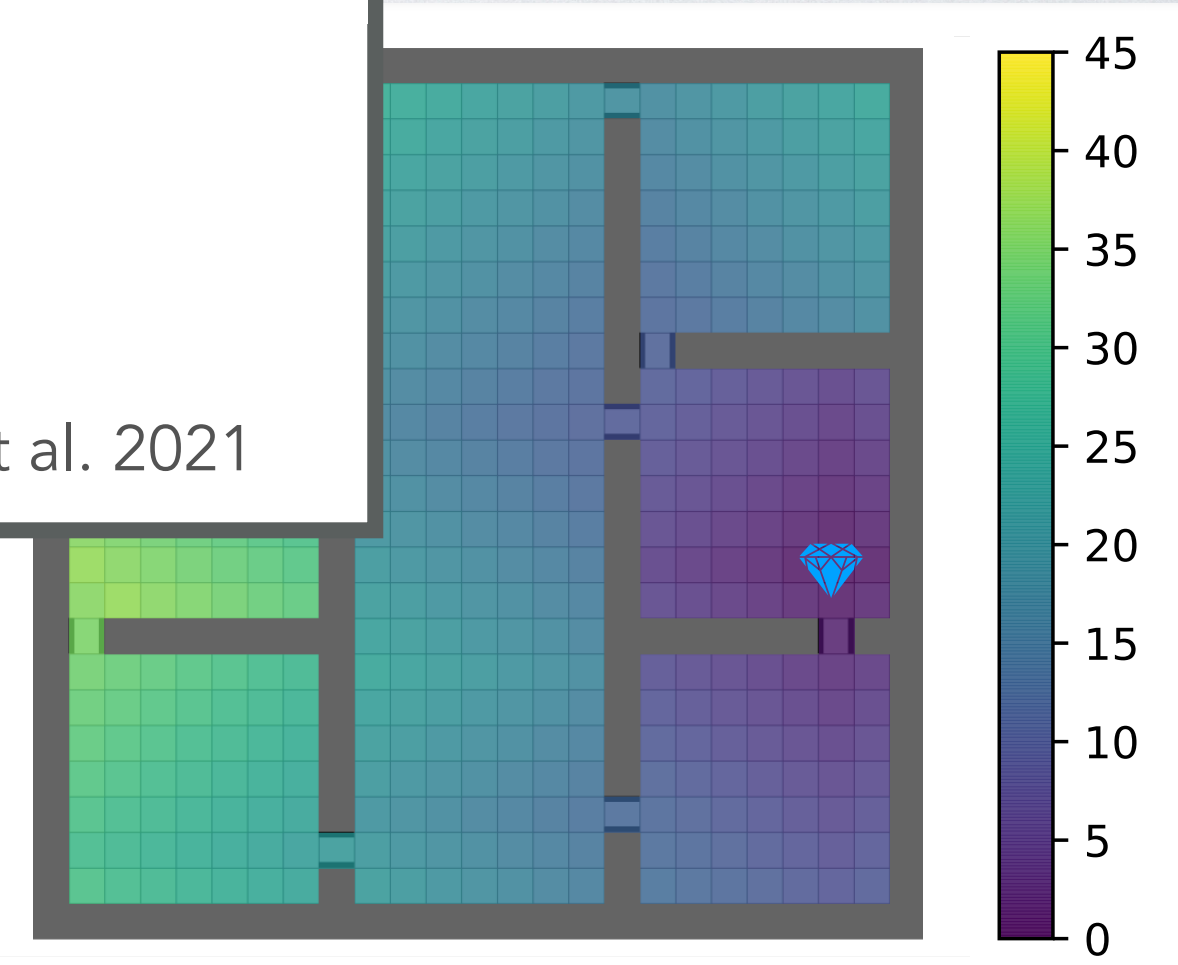


Fig. 1 of Efroni et al. 2021

$\text{distance}(s, \diamond)$
i.e.,
 $\text{total-cost}(s \rightarrow \diamond)$



Pick action that leads to smaller
 $\text{distance}(\text{next state}, \diamond)$

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)

Model-Based Agent

- + Easy to optimize, learn from local transitions
- + Easy to add known structure (e.g., +objects)
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation

Learn a model of the world
Plan w.r.t. world model

Value-Based Agent

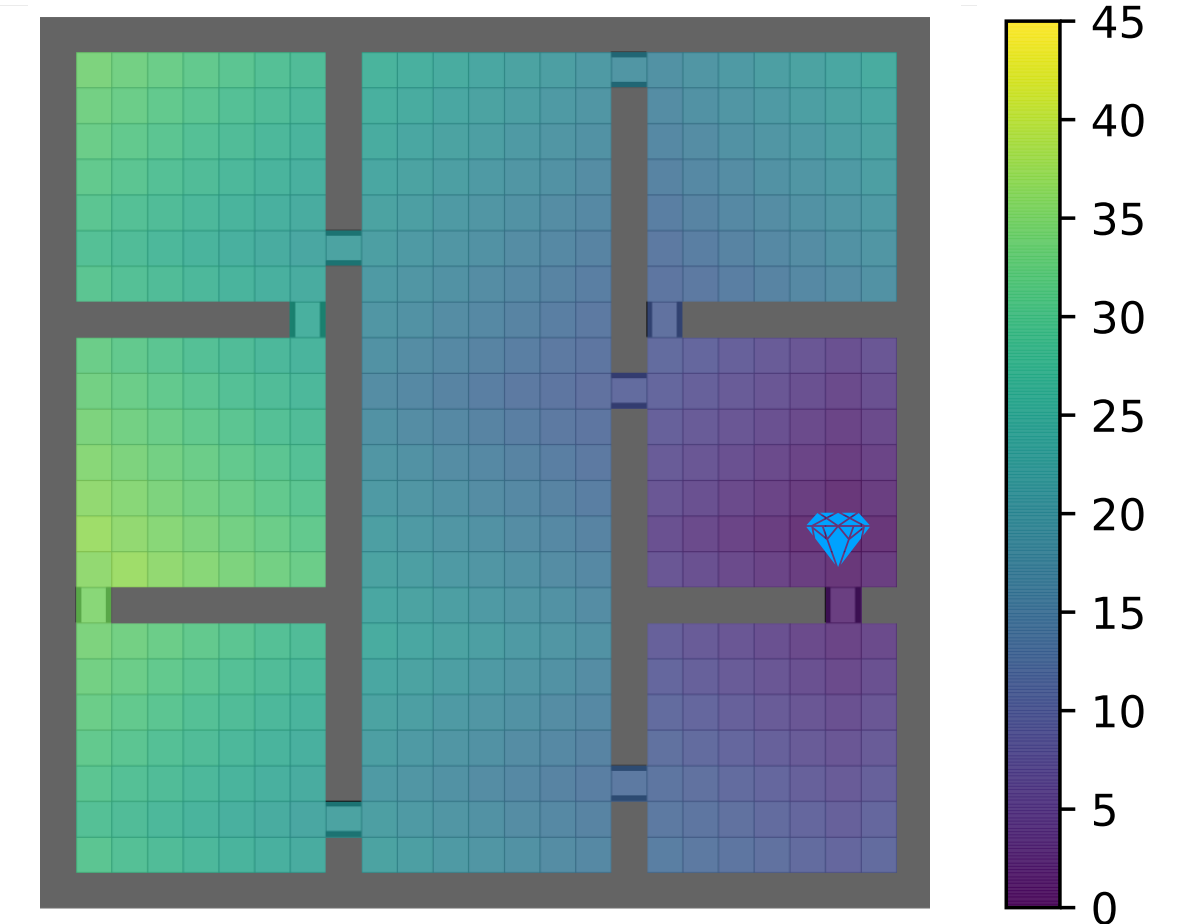
Goal is 

Label each state s with

$\text{distance}(s, \text{diamond})$

i.e.,

$\text{total-cost}(s \rightarrow \text{diamond})$



Pick action that leads to smaller
 $\text{distance}(\text{next state}, \text{diamond})$

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)


Model-Based Agent

- + Easy to optimize, learn from local transitions
- + Easy to add known structure (e.g., +objects)
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation

Learn a model of the world
Plan w.r.t. world model

Value-Based Agent

- + Learn **exactly** what is needed to do well
- + Long-range global quantity (distance)
- Hard to optimize (b/c bootstrapping alg.)
- Any structure in distance/value functions?
- Multi-Goal?

Pick action that leads to smaller
distance(next state, )

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)


Model-Based Agent

- + Easy to optimize, learn from local transitions
- + Easy to add known structure (e.g., +objects)
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation
- In practice still learns value...

Learn a model of the world
Plan w.r.t. world model

Value-Based Agent

- + Learn **exactly** what is needed to do well
- + Long-range global quantity (distance)
- Hard to optimize (b/c bootstrapping alg.)
- Any structure in distance/value functions?
- Multi-Goal?

Pick action that leads to smaller
distance(next state, )

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)


Model-Based Agent

- + Easy to optimize, learn from local transitions
- + Easy to add known structure (e.g., +objects)
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation
- In practice still learns value...

Learn a model of the world
Plan w.r.t. world model

Value-Based Agent

- + Learn **exactly** what is needed to do well
- + Long-range global quantity (distance)
- Hard to optimize (b/c bootstrapping alg.)
- Any structure in distance/value functions?
- Multi-Goal?

Pick action that leads to smaller
distance(next state, )

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)


Model-Based Agent

- + Easy to optimize, learn from local transitions
- + Easy to add known structure (e.g., +objects)
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation
- In practice still learns value...

Learn a model of the world
Plan w.r.t. world model

Value-Based Agent

- + Learn **exactly** what is needed to do well
- + Long-range global quantity (distance)
- Hard to optimize (b/c bootstrapping alg.)
- Any structure in distance/value functions?
- Multi-Goal?

Pick action that leads to smaller
distance(next state, )

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)


Model-Based Agent

- + Easy to optimize, learn from local transitions
- + Easy to add known structure (e.g., +objects)
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation
- In practice still learns value...

Learn a model of the world
Plan w.r.t. world model

Value-Based Agent

- + Learn **exactly** what is needed to do well
- + Long-range global quantity (distance)
- Hard to optimize (b/c bootstrapping alg.)
- Any structure in distance/value functions?
- Multi-Goal?

Pick action that leads to smaller
distance(next state, )

Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)

Model-Based Agent

+ Easy to optimize, learn from local transitions

+ Easy to

+ Multi-Goal

- Abstract

- Error acc

- In practice


Learn a model of the world
Plan w.r.t. world model

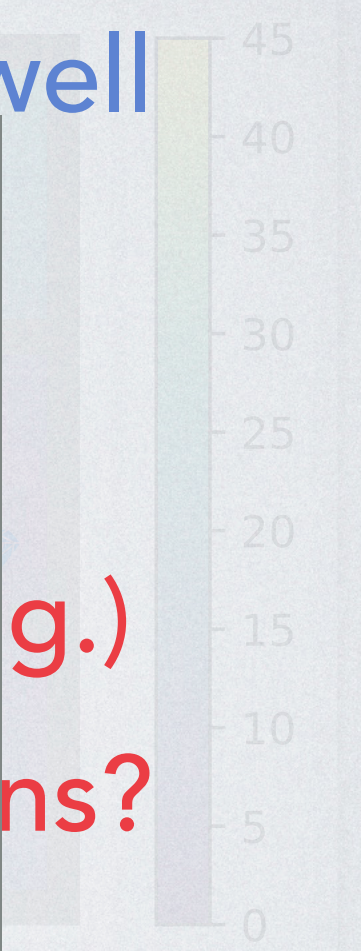
Value-Based Agent

+ Learn **exactly** what is needed to do well

✗ learn a model of how actions affect the full world

✓ learn a **model** of how actions affect the **distance/value** to goal

Pick action that leads to smaller
distance(next state, )



Generalist Goal-Reaching Agents

Control / Sequential decision making = Act at each **state** at each **timestep**

Generalist agents can do everything in an environment (e.g., entire house/city)

Model-Based Agent

+ Easy to optimize, learn from local transitions

+ Easy to

+ Multi-Goal

- Abstract

- Error acc

- In practice

✗ learn a model of how actions affect the full world

✓ learn a **model** of how actions affect the **distance/value** to ALL goals (e.g., diamonds?)

Learn a model of the world

Plan w.r.t. world model

Value-Based Agent

+ Learn **exactly** what is needed to do well

Pick action that leads to smaller

distance(next state, )

✓ learn a **model** of how actions affect the **distance/value** to ALL goals

- + Learn *exactly* what is needed to do well
- + Local training
- + Long-range global quantity
- + Value estimates can be imperfect (can use model)
- + No bootstrapping alg.
- + Inherently designed for multi-goal


Model-Based Agent

- + Easy to optimize, learn from local transitions
- + Easy to add known structure (e.g., +objects)
- + Multi-Goal
- Abstraction? Learn without reconstruction?
- Error accumulation
- In practice still learns value...

Learn a model of the world
Plan w.r.t. world model

Value-Based Agent

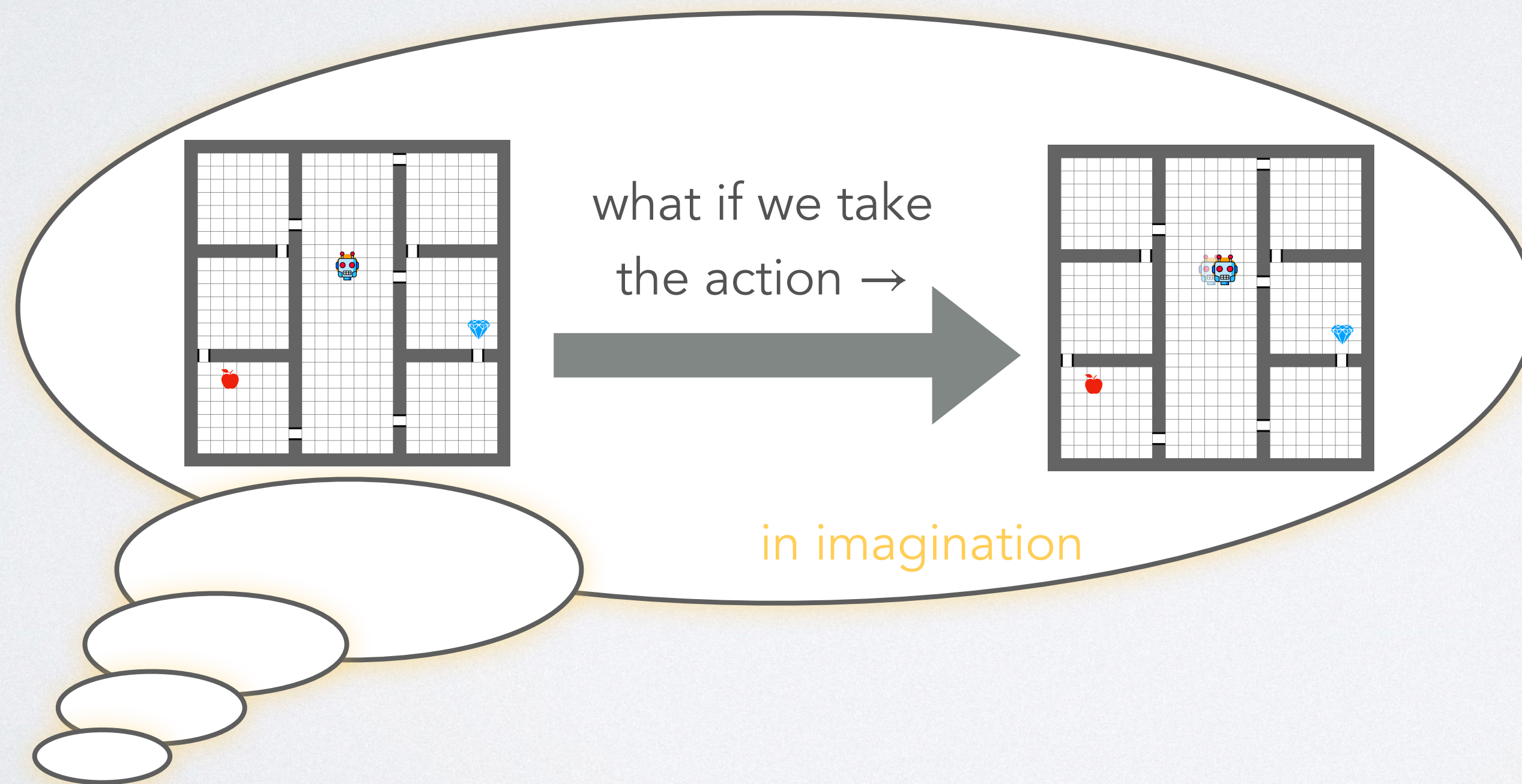
- + Learn *exactly* what is needed to do well
- + Long-range global quantity (distance)
- Hard to optimize (b/c bootstrapping alg.)
- Any structure in distance/value functions?
- Multi-Goal?

Pick action that leads to smaller
distance(next state, )

Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

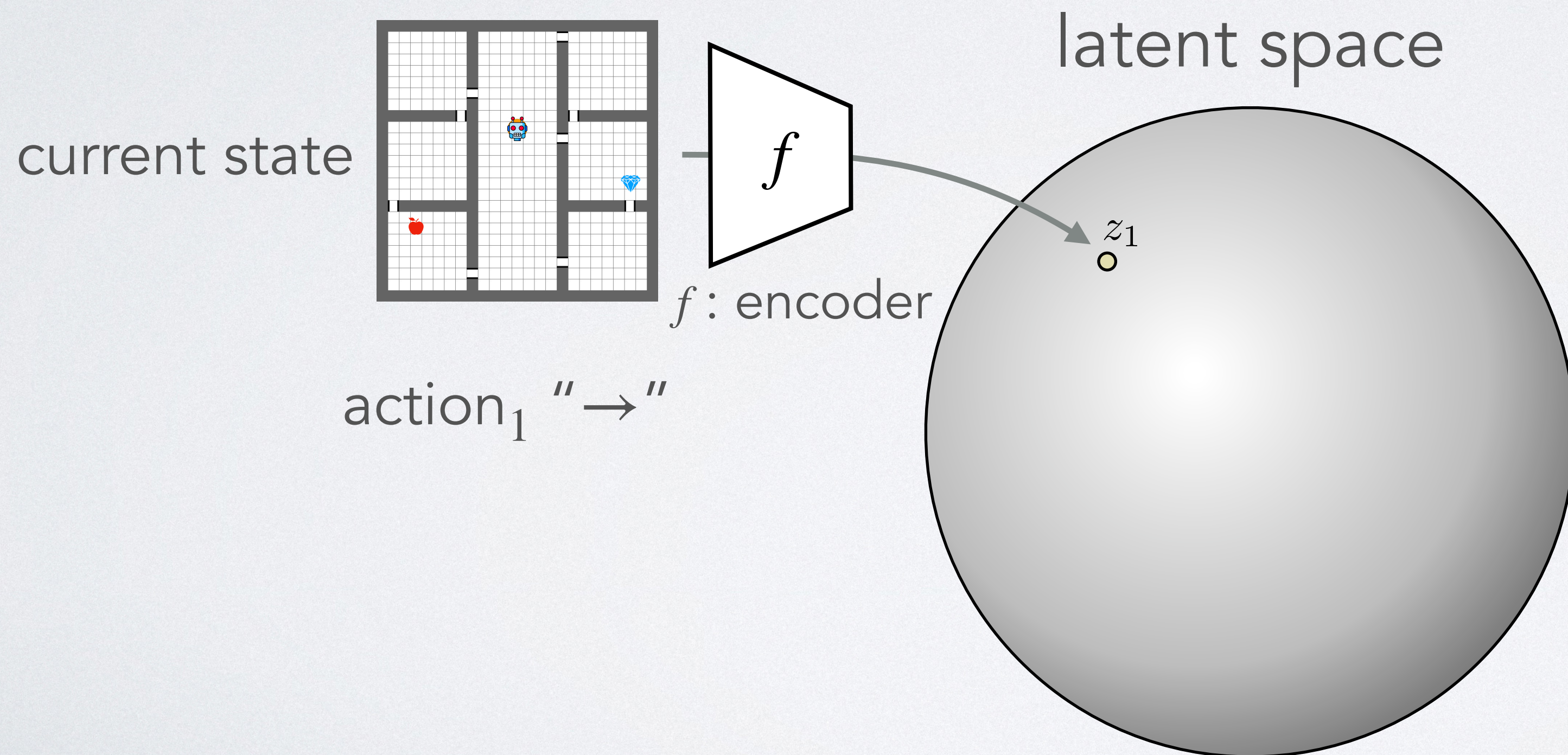
Background (X): Model-Based Agents learn a model of how actions affect the full world



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

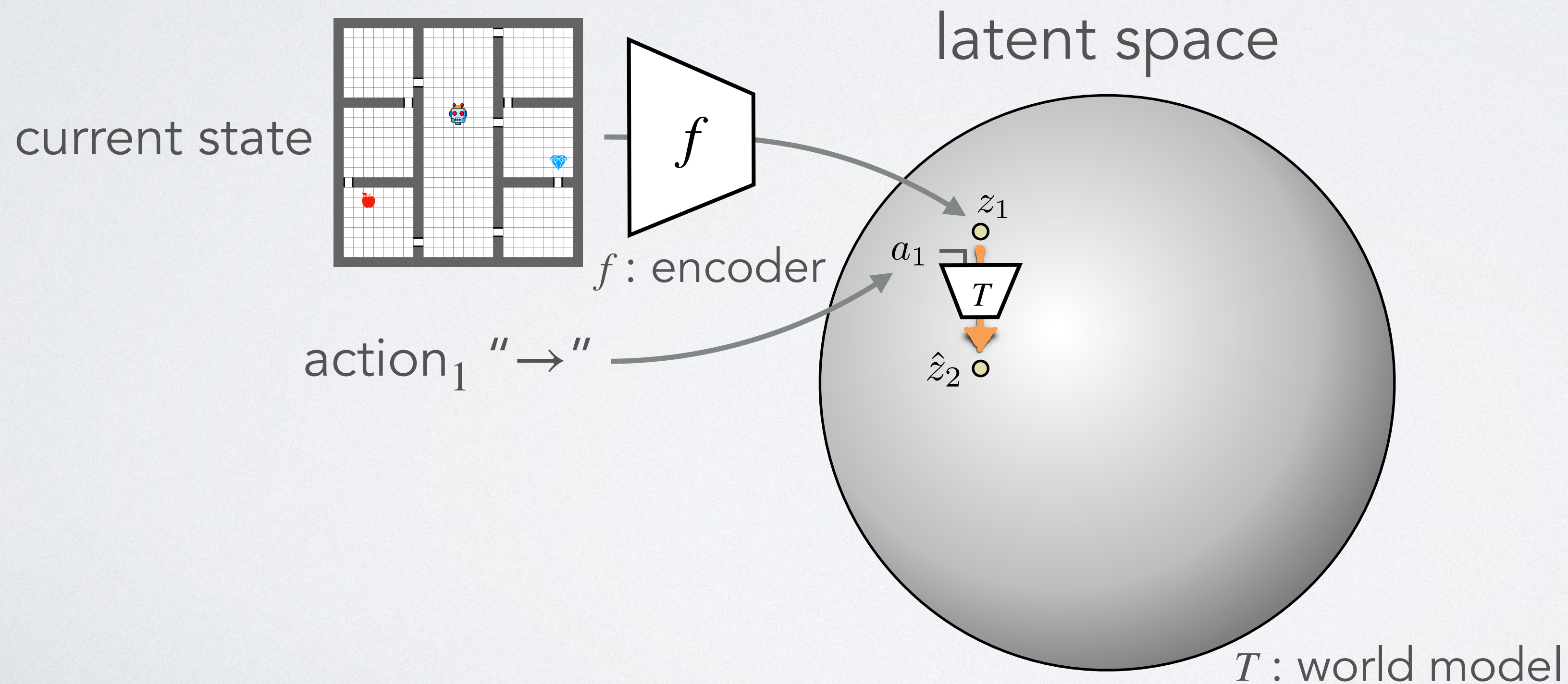
Background (✗): Model-Based Agents learn a model of how actions affect the full world



Generalist Agent via Value-Aware World Models

- ✓ learn a model of how actions affect the distance/value to **ALL** goals

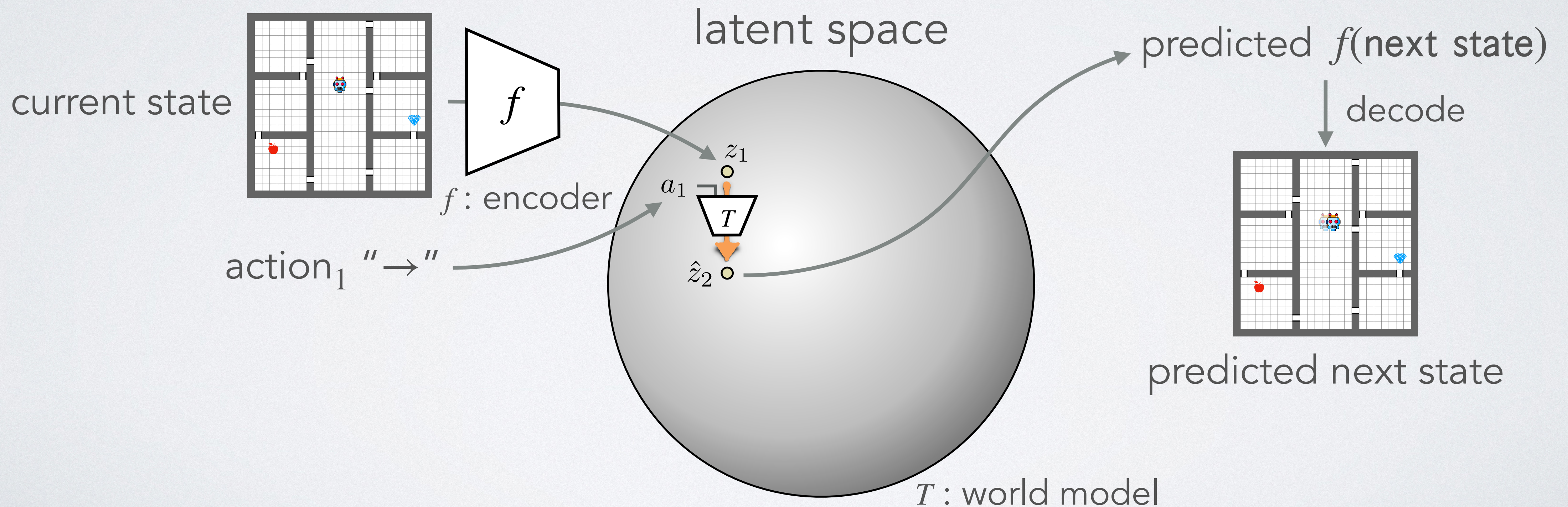
Background (X): Model-Based Agents learn a model of how actions affect the full world



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

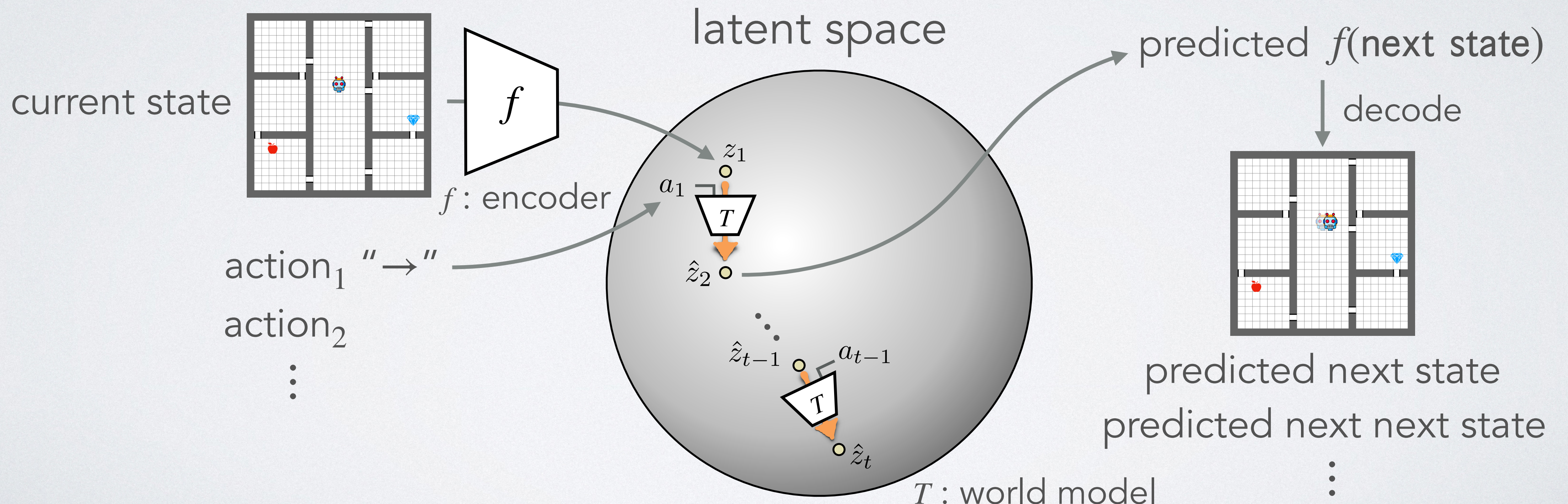
Background (X): Model-Based Agents learn a model of how actions affect the full world



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

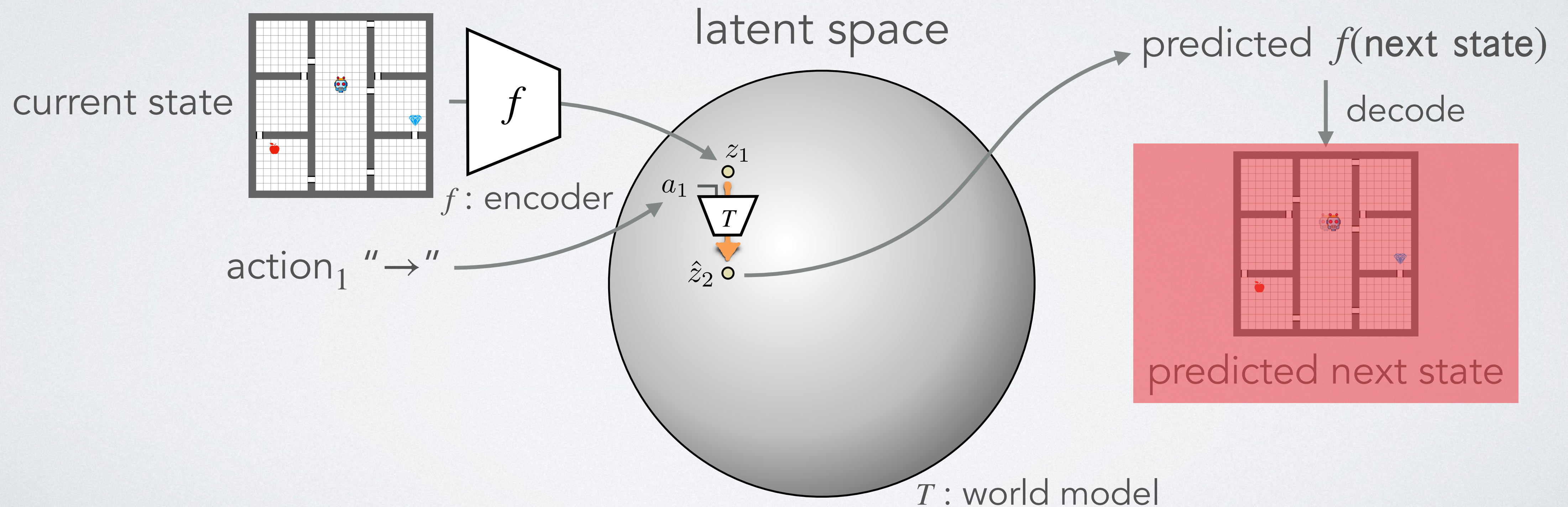
Background (✗): Model-Based Agents learn a model of how actions affect the full world



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

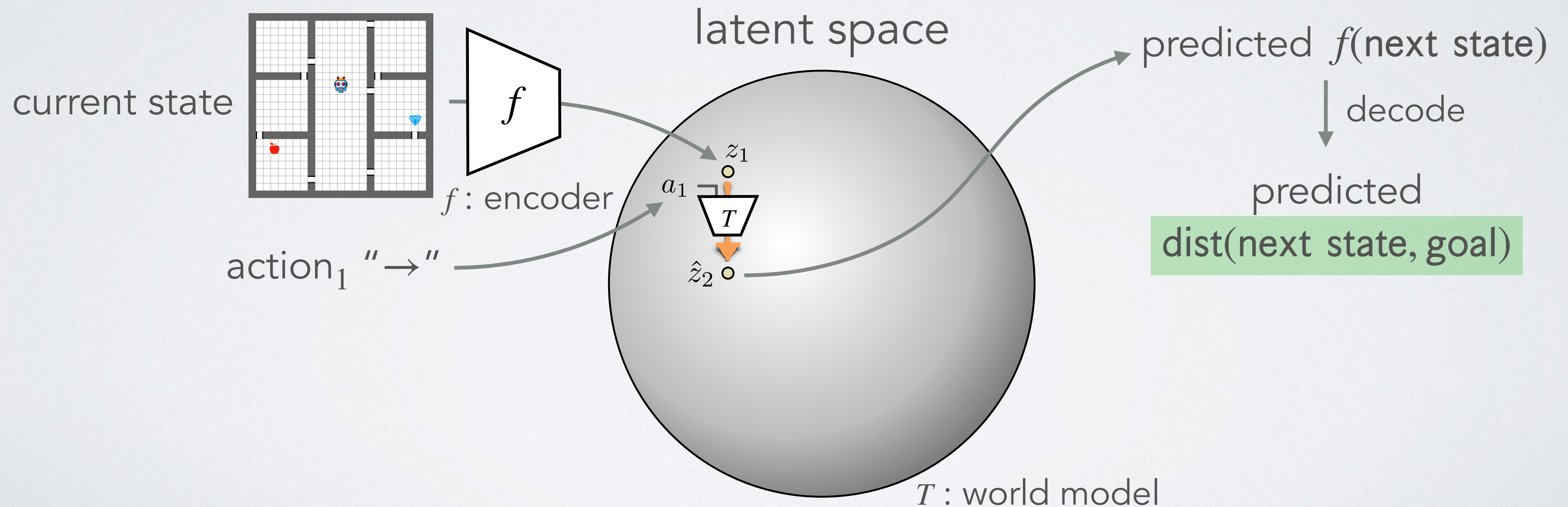
✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

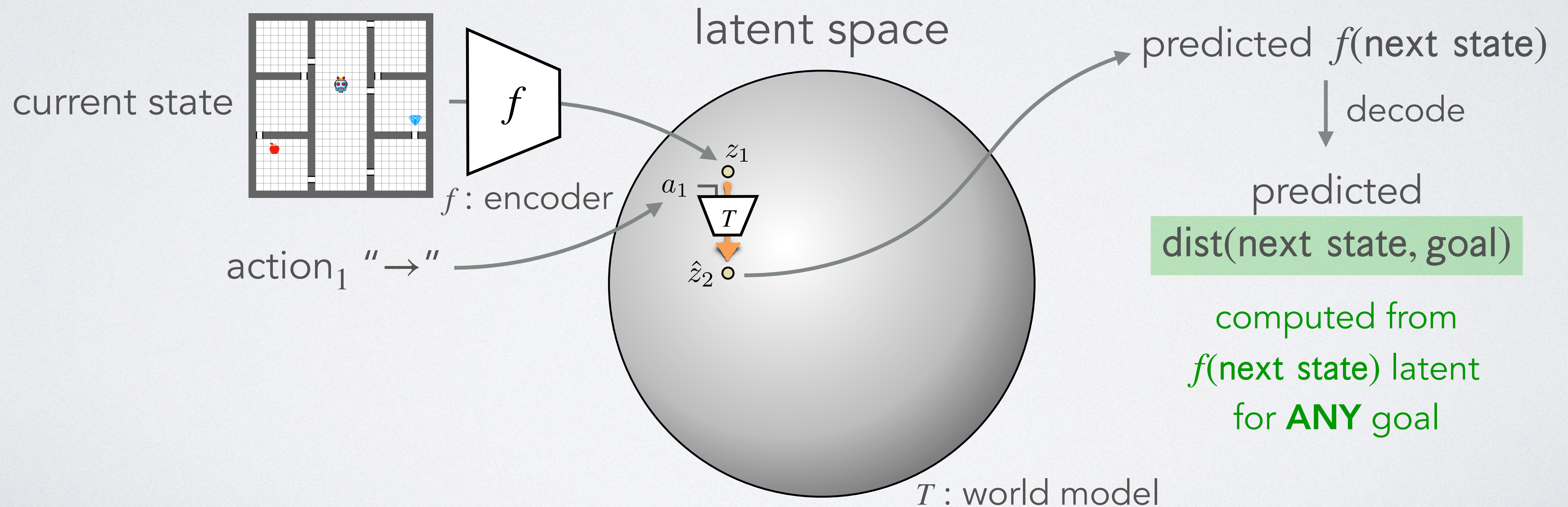
✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

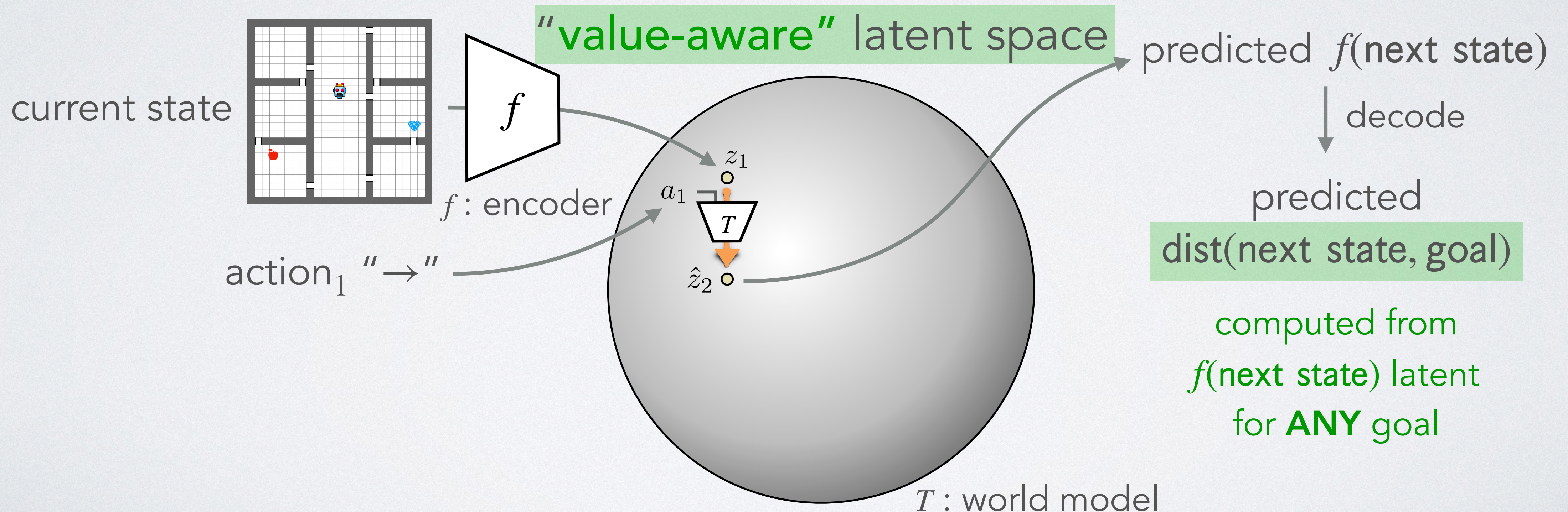
✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

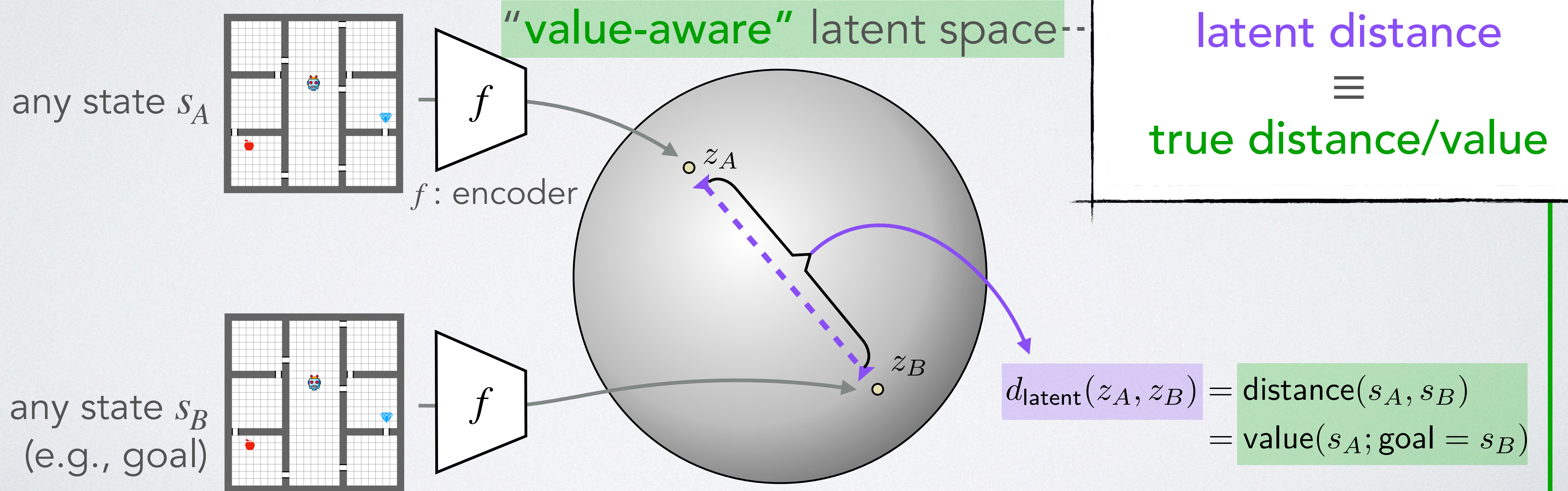
✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

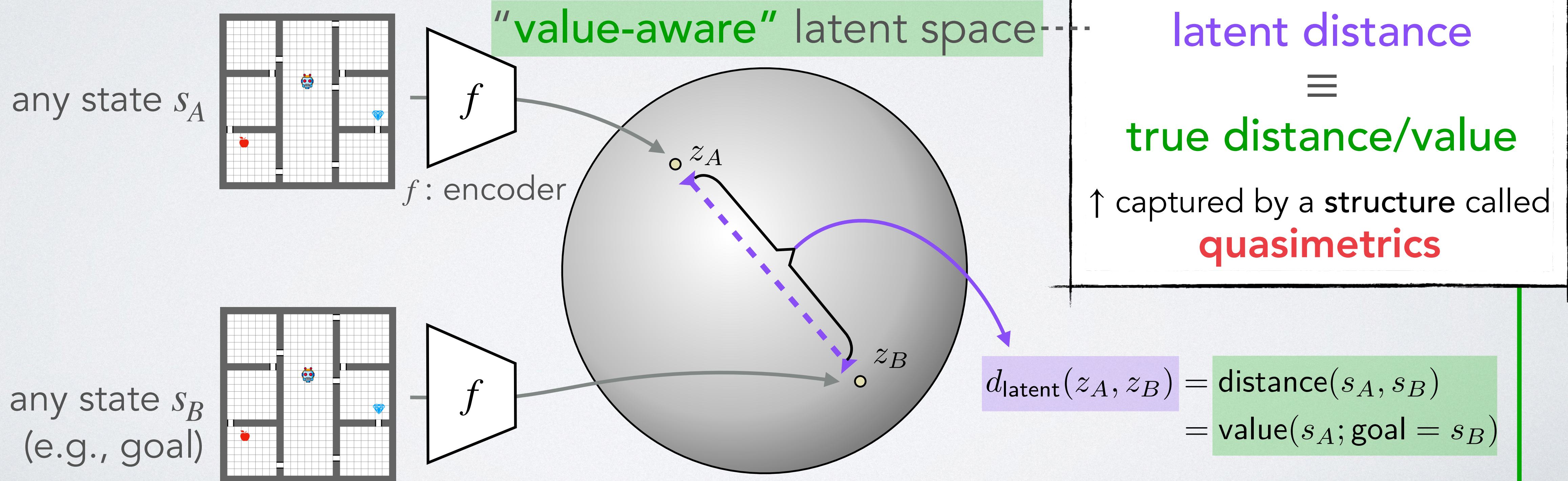
✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

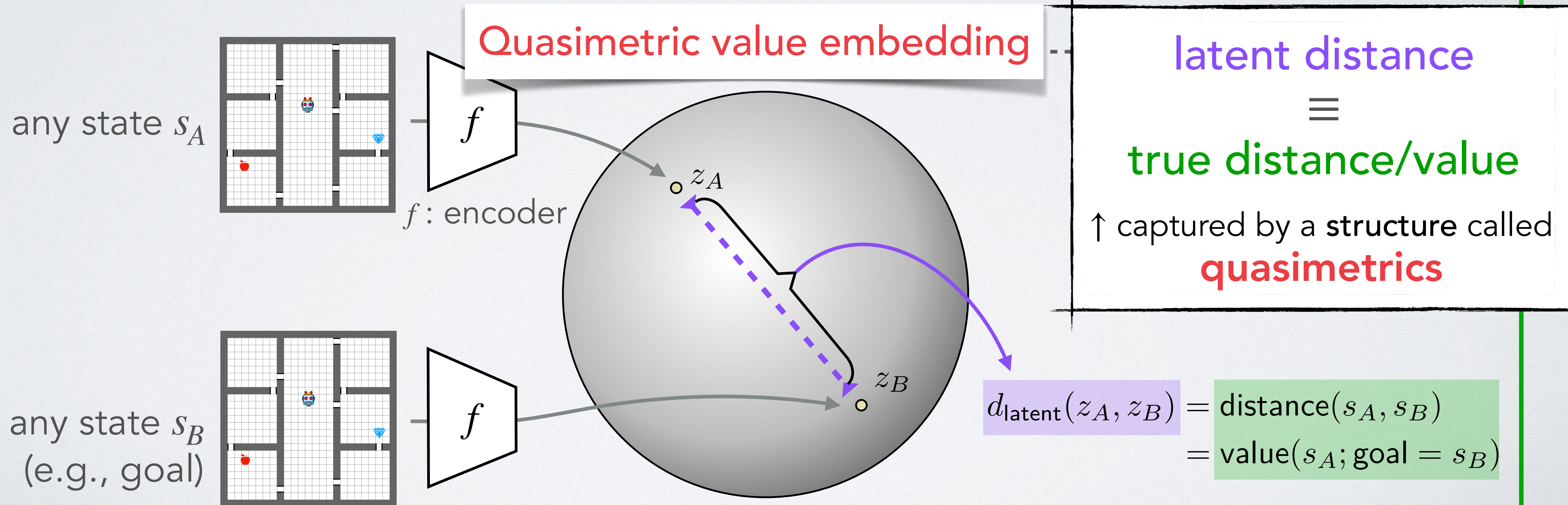
✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via Value-Aware World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

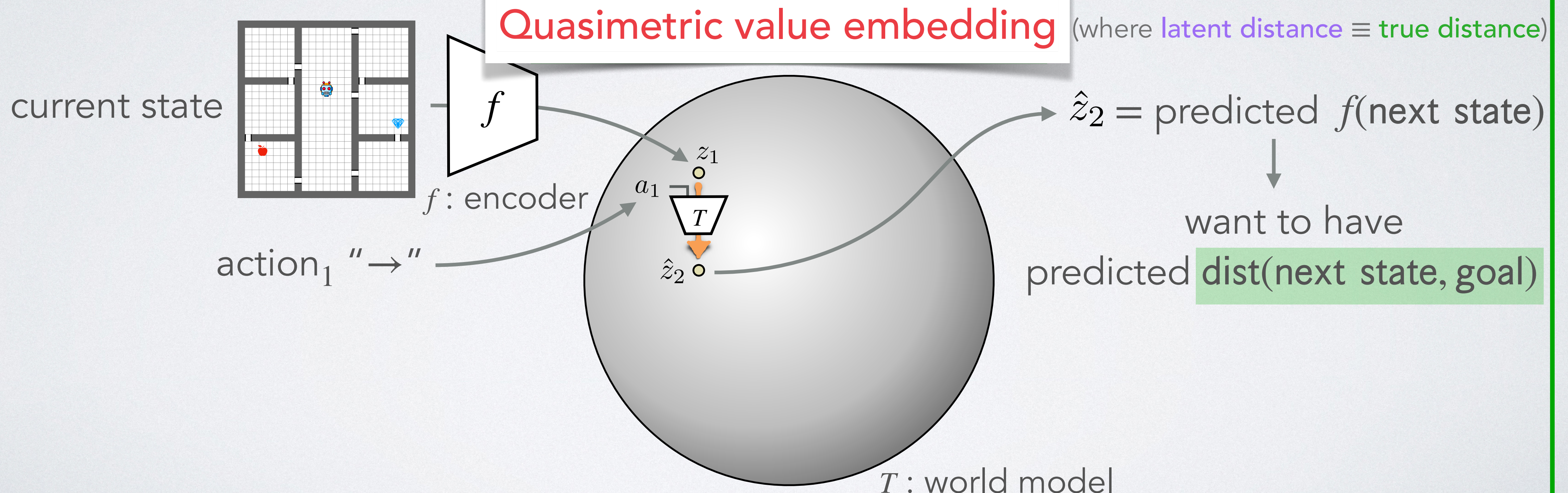
✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via **Quasimetric** World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

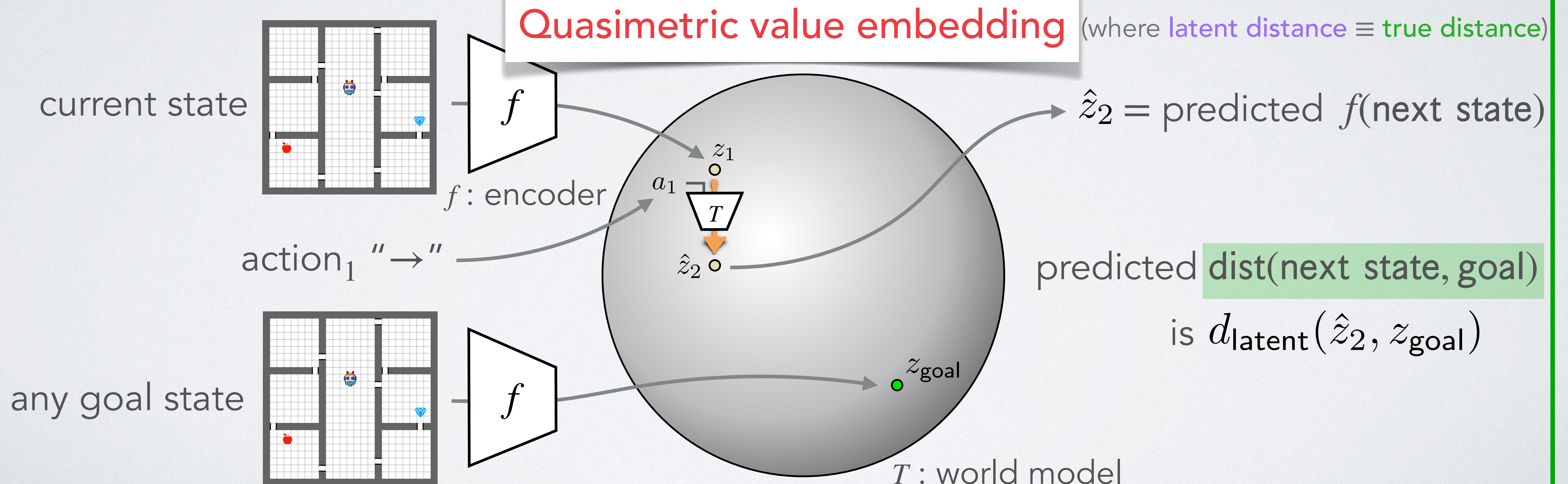
✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via **Quasimetric** World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

✗ predict full state ✓ predict distance to goal — what planning really needs



Generalist Agent via **Quasimetric** World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

Quasimetric value embedding is all you need

1. get a **Quasimetric value embedding**
where **latent distance** \equiv **true distance**
2. train a latent world model in it
3. agent plan/trained w.r.t. world model
4. profit!

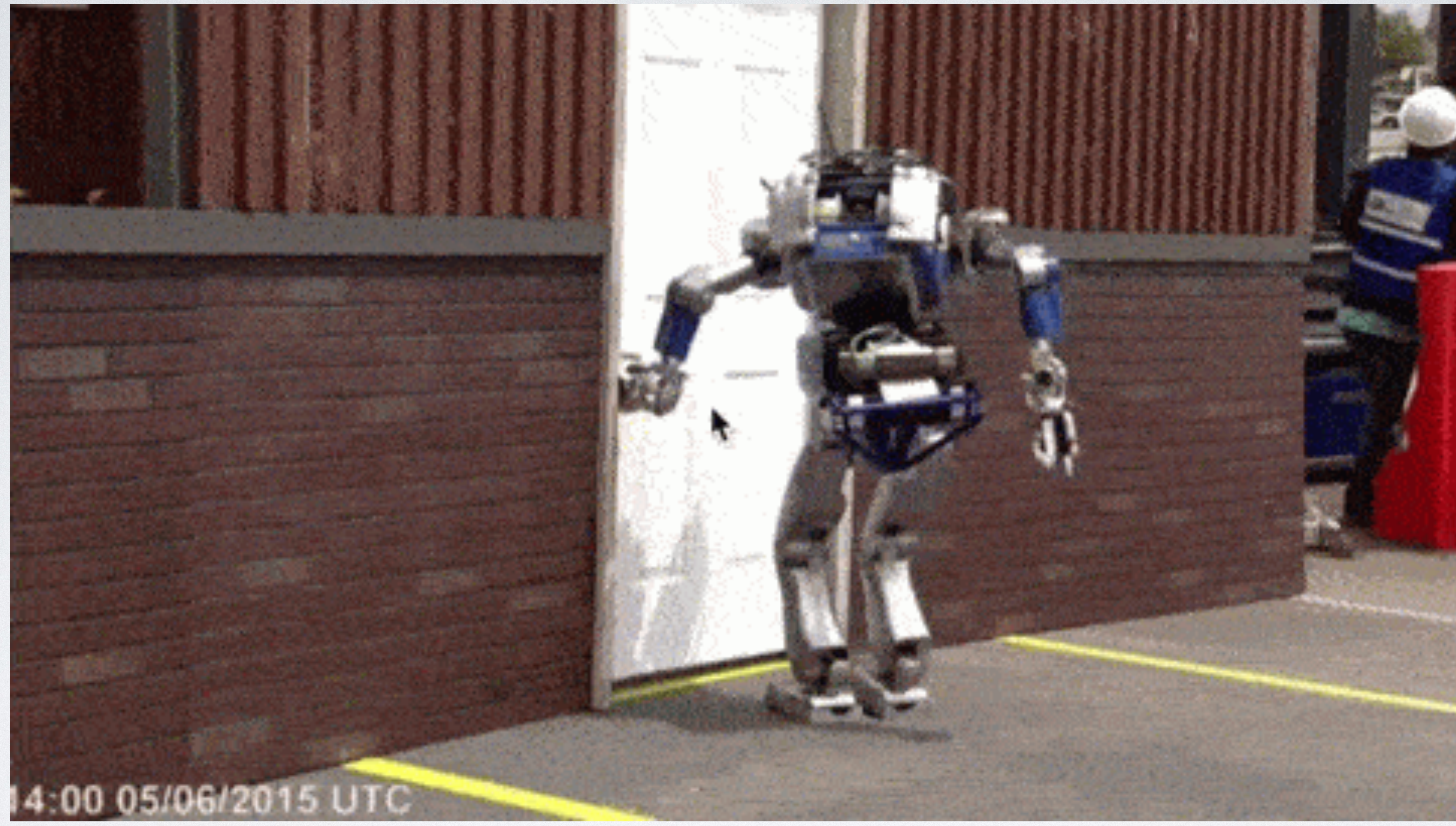
Agenda

Motivation Generalist (Goal-Reaching) Agents
via a **quasimetric** world models

Value-Aware Latent Spaces are Quasimetrics

Quasimetrics with Neural Nets Quasimetric Embedding

Learn **Quasimetric** World Models for Agents Quasimetric Reinforcement Learning



Structures for Generalist Agents

Value Function Is a “Distance”

Definition (value function is a “distance” between states)

– $V^*(s_0; \text{goal} = s_1)$ measures closeness of $s_0 \rightarrow s_1$ by optimal total **cost** to go $s_0 \rightarrow s_1$.

V^* is known as the optimal goal-conditioned value function in RL.

- Shortest-path distance in the finite case

Value Function Is a “Distance”

Definition (value function is a “distance” between states)

– $V^*(s_0; \text{goal} = s_1)$ measures closeness of $s_0 \rightarrow s_1$ by optimal total **cost** to go $s_0 \rightarrow s_1$.
 V^* is known as the optimal goal-conditioned value function in RL.

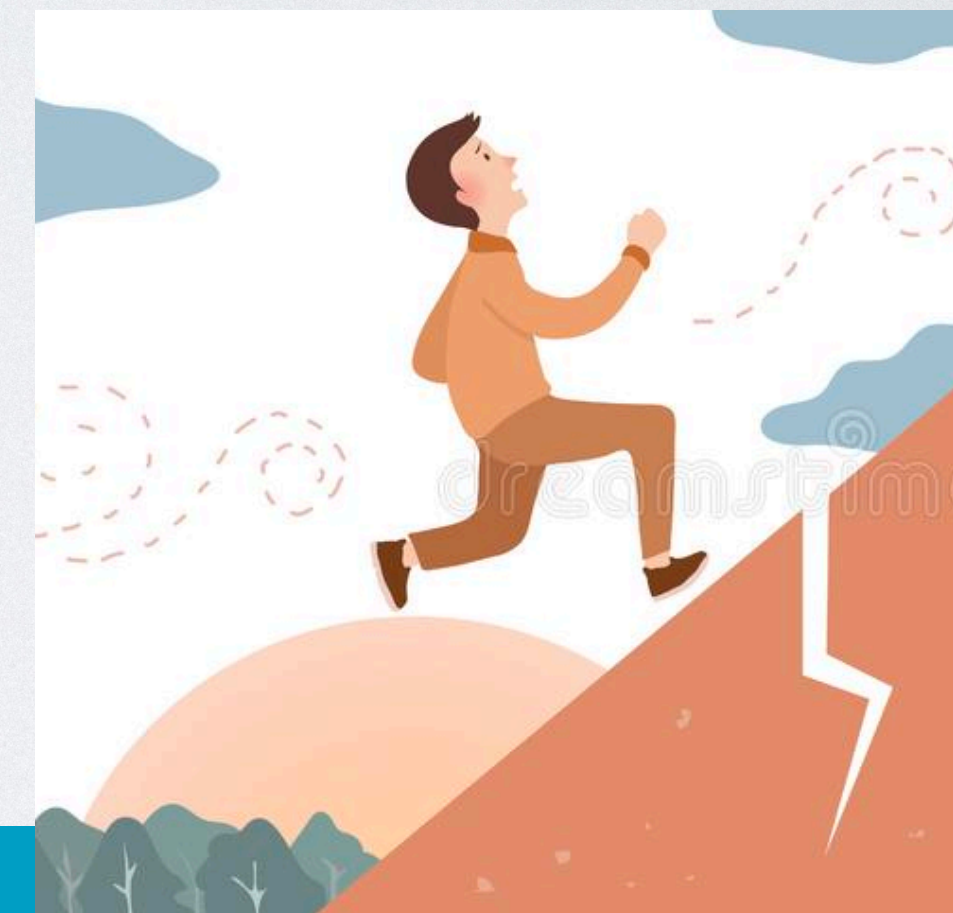
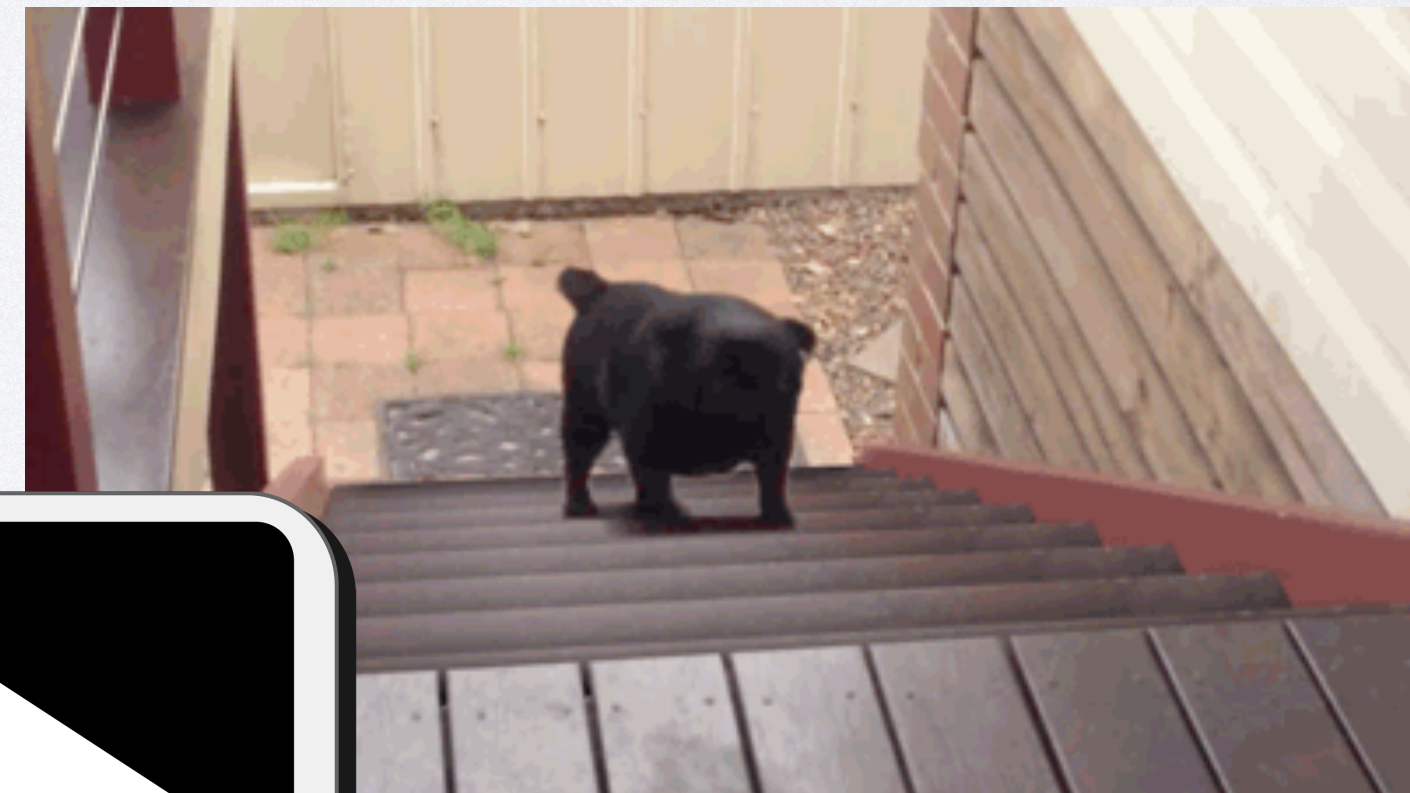
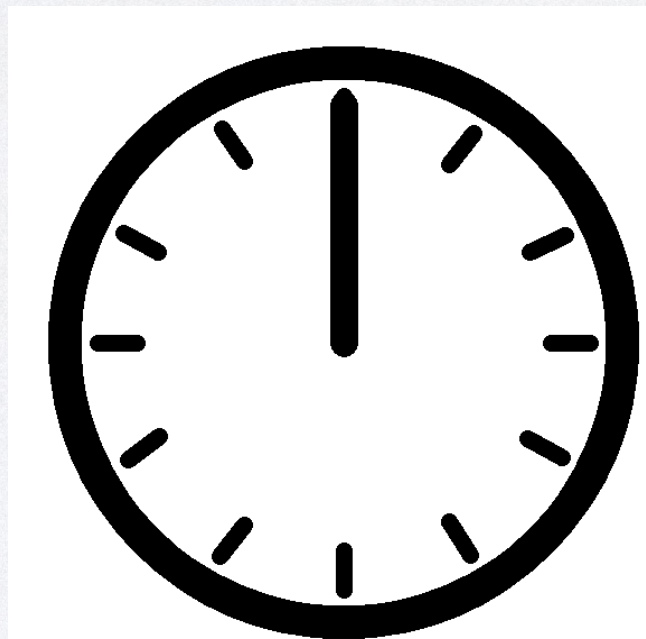
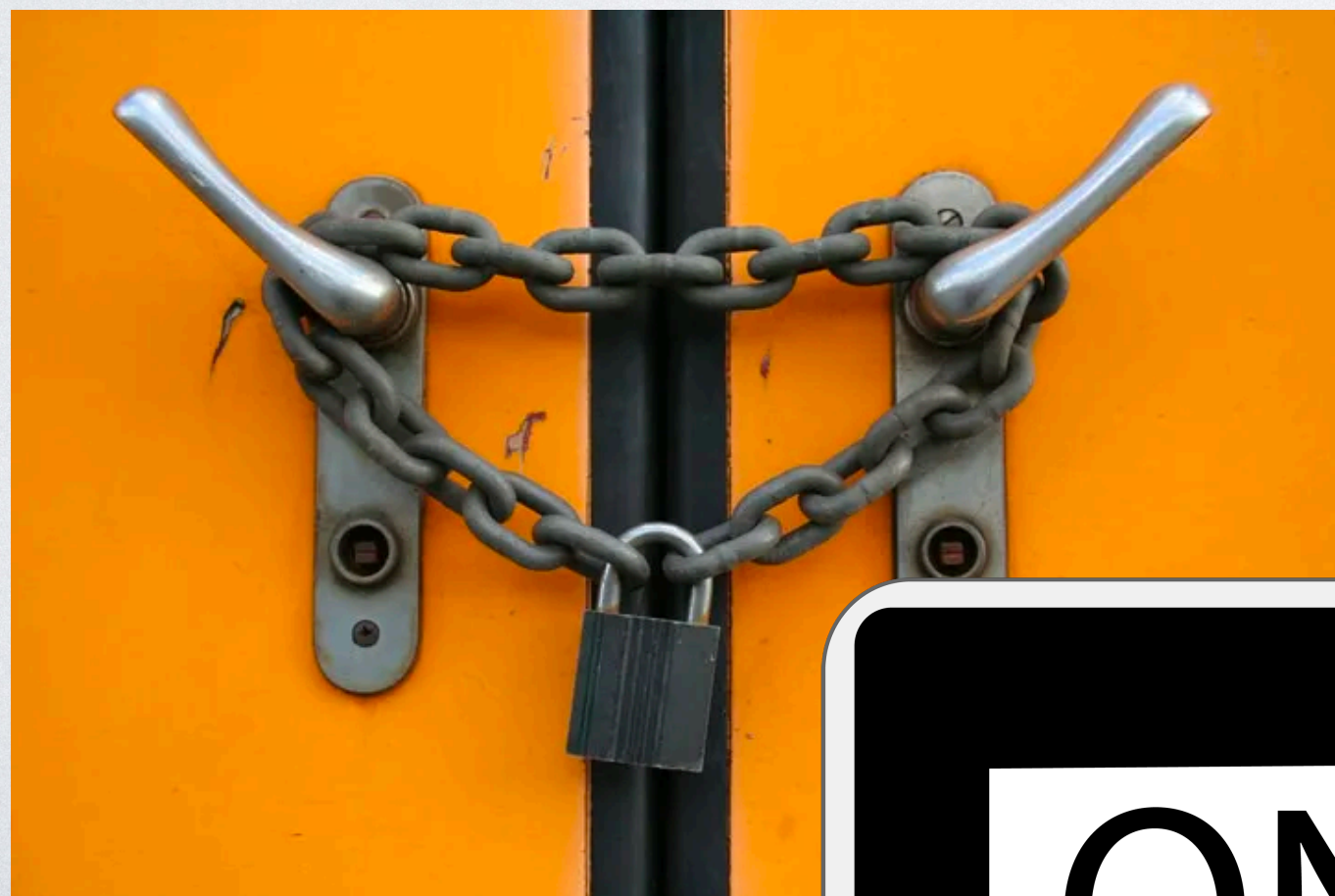
- Shortest-path distance in the finite case
- $V^*(s_0; \text{goal} = s_1)$ is the cost of reaching s_1 from s_0 from an optimal agent
- V^* is what a generalist decision-making agent should (approx.) learn
(value-based agents, model-based agent, “value-aware” latent space)

Value Function Is a “Distance”

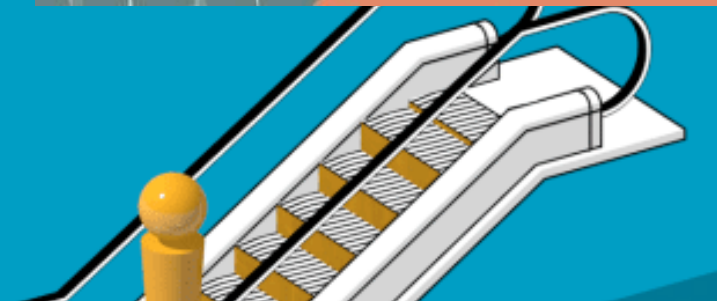
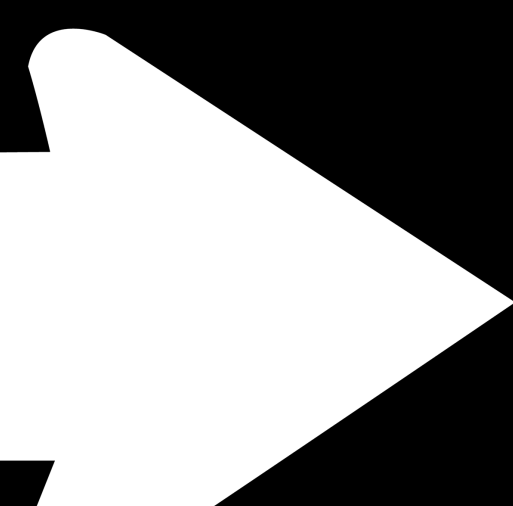
Definition (value function is a “distance” between states)

- $V^*(s_0; \text{goal} = s_1)$ measures closeness of $s_0 \rightarrow s_1$ by optimal total **cost** to go $s_0 \rightarrow s_1$.
- V^* is known as the optimal goal-conditioned value function in RL.

- **Asymmetry:** $V^*(s_0; s_1) \neq V^*(s_1; s_0)$ in general. So not actually a distance



ONE WAY

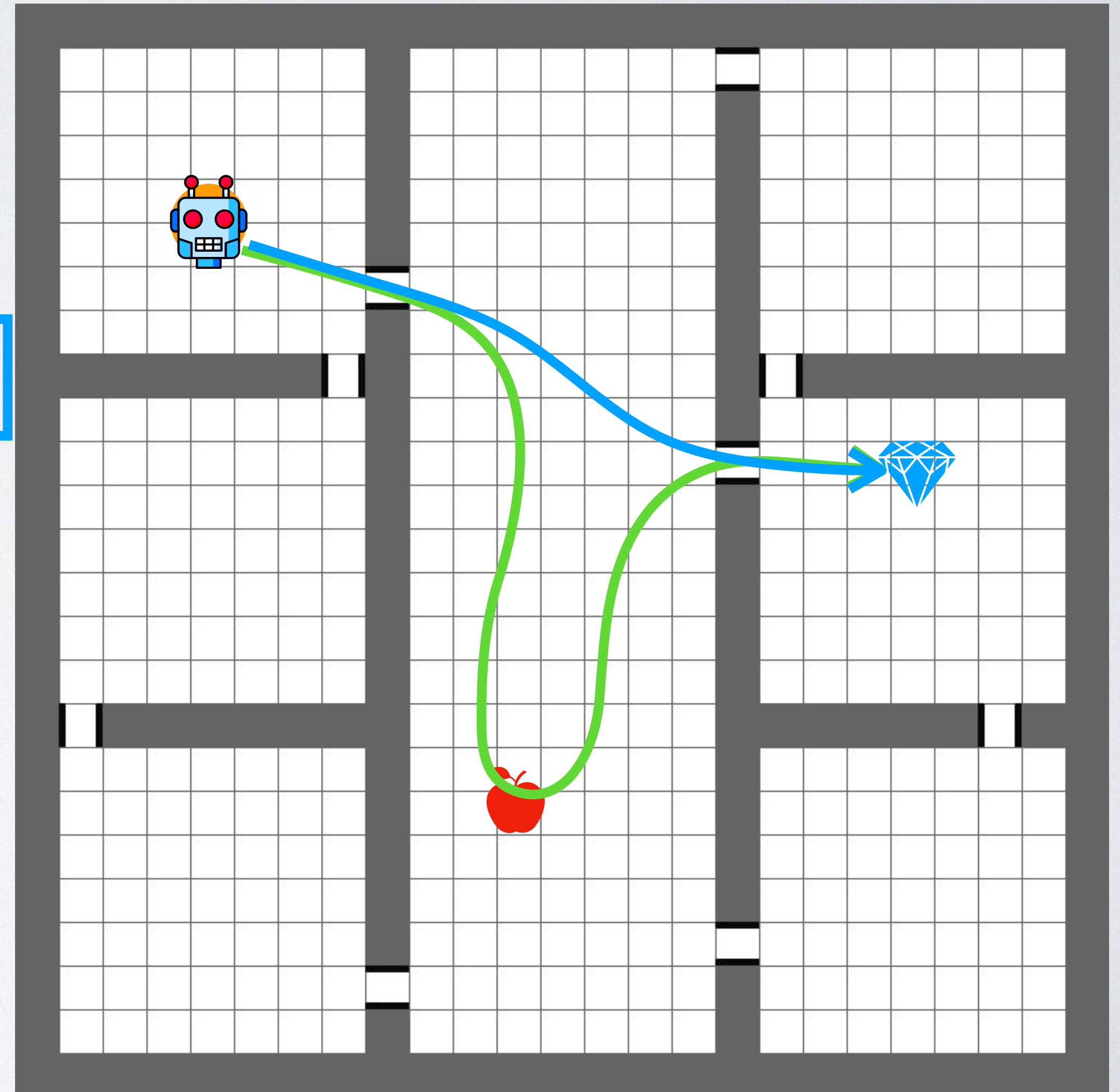


Optimality \implies Triangle Inequality

– $V^*(s, g)$ always satisfies triangle inequality:

$$- V^*(s_A; s_B) - V^*(s_B; s_C) \geq - V^*(s_A; s_C)$$

Why? Because optimal plan from A to C *could be* for $A \rightarrow B \rightarrow C$, if that's the best, otherwise is better.



Value Function Is a “Distance”

Definition (value function is a “distance” between states)

– $V^*(s_0; \text{goal} = s_1)$ measures closeness of $s_0 \rightarrow s_1$ by optimal total **cost** to go $s_0 \rightarrow s_1$.
 V^* is known as the optimal goal-conditioned value function in RL.

- **Asymmetry:** $V^*(s_0; s_1) \neq V^*(s_1; s_0)$ in general. So not actually a distance
- Optimal cost \implies **Triangle inequality**

Value Function Is a “Distance”

Definition (value function is a “distance” between states)

– $V^*(s_0; \text{goal} = s_1)$ measures closeness of $s_0 \rightarrow s_1$ by optimal total **cost** to go $s_0 \rightarrow s_1$.
 V^* is known as the optimal goal-conditioned value function in RL.

- **Asymmetry:** $V^*(s_0; s_1) \neq V^*(s_1; s_0)$ in general. So not actually a distance
- Optimal cost \implies **Triangle inequality**
- – V^* is almost a distance between states, **except for the symmetry constraint**
- ... such relaxed metrics are called **quasimetrics**

Value Function Is a Quasimetric

Definition (value function – $V^* \in$ quasimetrics on states)

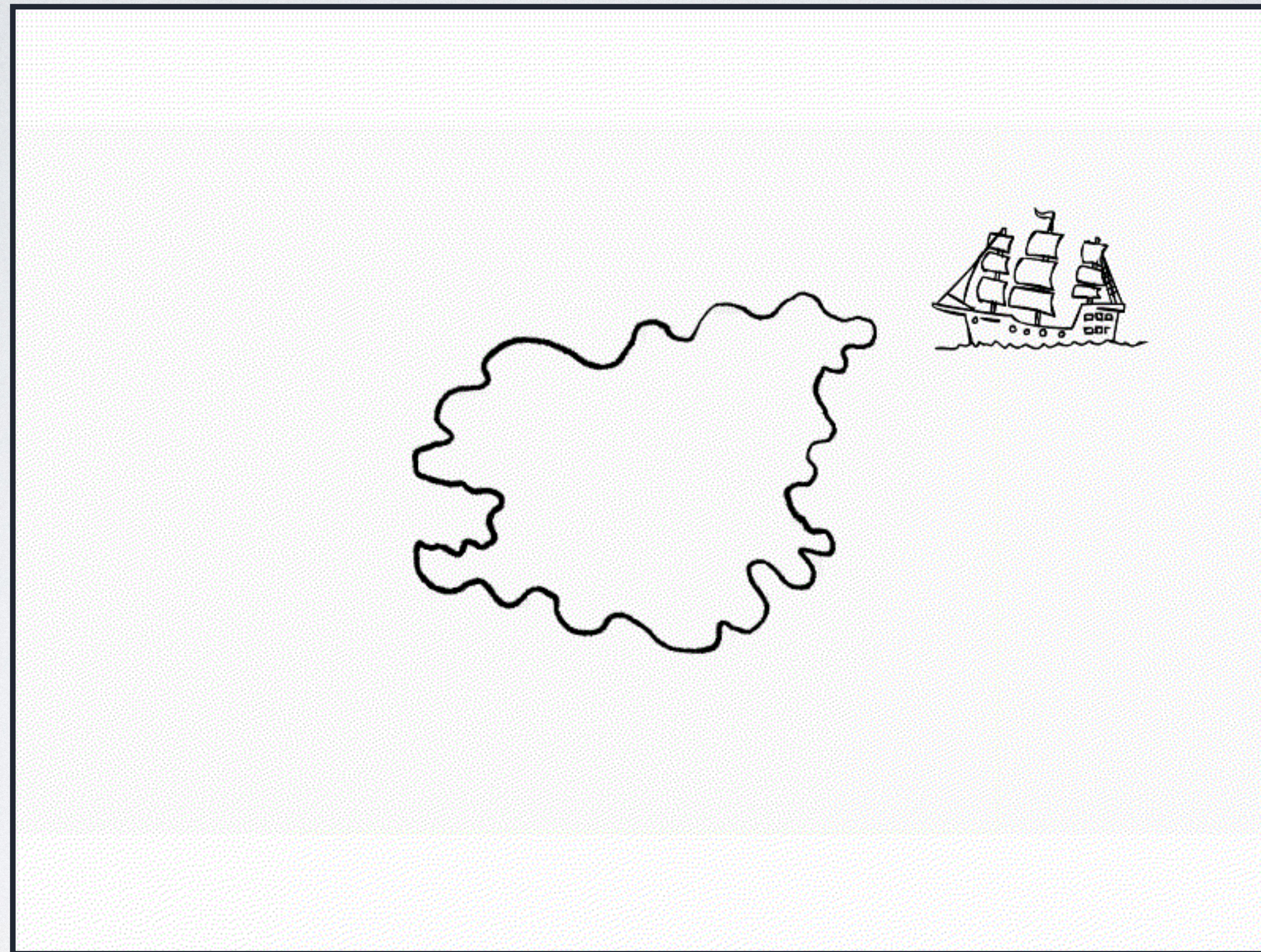
– $V^*(s_0; \text{goal} = s_1)$ measures closeness of $s_0 \rightarrow s_1$ by optimal total **cost** to go $s_0 \rightarrow s_1$.

V^* is known as the optimal goal-conditioned value function in RL.

- **Asymmetry:** $V^*(s_0; s_1) \neq V^*(s_1; s_0)$ in general. So not actually a distance
- Optimal cost \implies **Triangle inequality**
- – V^* is almost a distance between states, **except for the symmetry constraint**
- ... such relaxed metrics are called quasimetrics

Value Functions \equiv Quasimetrics

- A generalist decision-making agent should learn V^*
 - For any environment, $-V^*$ and $d \in$ quasimetrics on states
 - Reverse also hold...
- [ICML 23, Thm. 1; Value-Quasimetric **Equivalence**]
 {all quasimetrics on states} \equiv { $-V^*$ for all MDPs}
- Proof by construction.
- Quasimetrics is the exact and only structure for V^*
 - To learn V^* , quasimetrics is the exact function class with correct inductive bias



(Deep) Learning **Quasimetrics**

Learning **Quasimetrics**

Symmetrical Relation ← Metric Embeddings

Asymmetrical Relation ← **????**

Learning **Quasimetrics**

Symmetrical Relation ← Metric Embeddings

Asymmetrical Relation ← **Just a regular neural network $f_{\text{NN}}(x, y)$?**

- Indeed, many goal-reaching RL papers use this!
- But....

- [ICLR 22, Thm. 4.6 & Experiments]

Such NN formulations can arbitrarily badly violate quasimetric properties.

Proof by construction + NN as NTK dot-product kernel + extremal combinatorics (probabilistic method).

Learning **Quasimetrics**

Symmetrical Relation ← Metric Embeddings

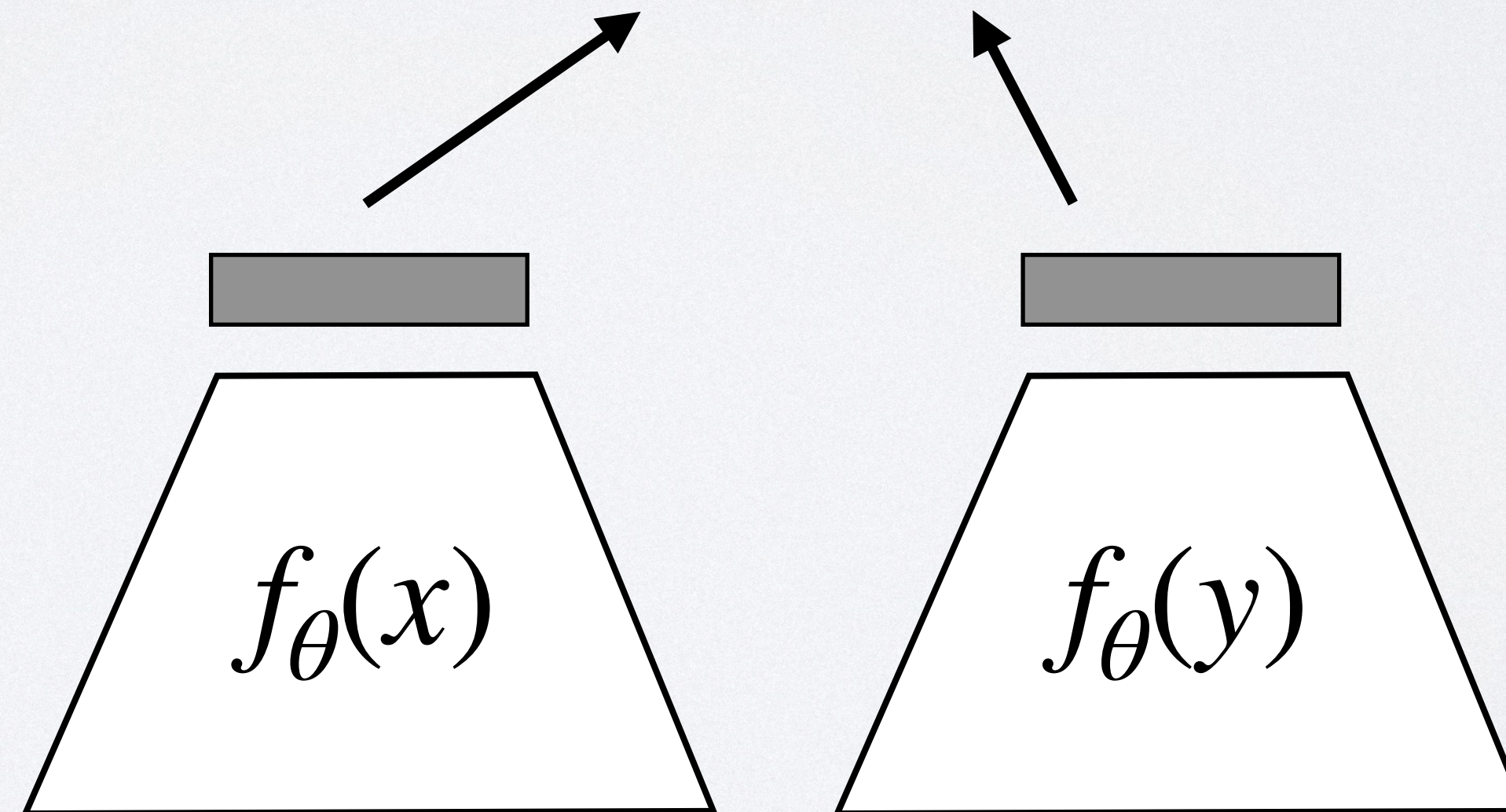
Asymmetrical Relation ← **Quasimetric Embeddings** [ICLR 22, NeurIPS 22 workshop]

Quasimetric Embeddings

Components of A **Quasimetric Embedding** on \mathcal{X} (e.g., states, images)

1. $f_\theta: \mathcal{X} \rightarrow \mathbb{R}^d$ generic unconstrained neural network encoder
2. $d_{\text{latent}}: \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, \infty]$ latent space quasimetric function

$$d_\theta(x, y) \triangleq d_{\text{latent}}(f_\theta(x), f_\theta(y)) \quad (\text{Quasimetric Embedding})$$



Aside: How to choose latent quasimetric d_{latent} ?

An Inductive Bias for Distances: Neural Nets that Respect the Triangle Inequality

Silviu Pitis, Harris Chan, Kiarash Jamali, Jimmy Ba, ICLR 2020

On the Learning and Learnability of Quasimetrics

Tongzhou Wang, Phillip Isola, ICLR 2022

Metric Residual Networks for Sample Efficient Goal-Conditioned Reinforcement Learning

Bo Liu, Yihao Feng, Qiang Liu, Peter Stone, arXiv 2022

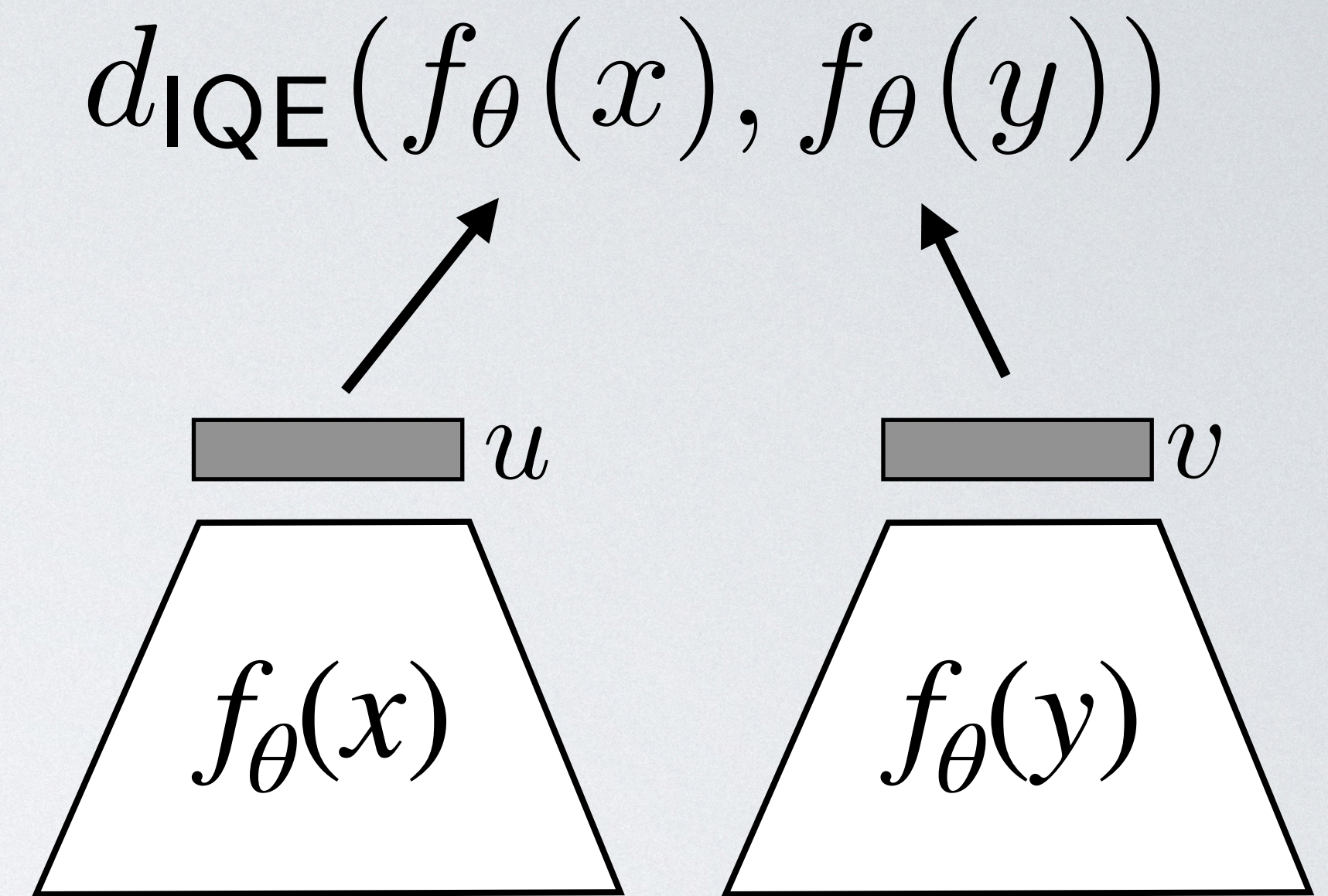
Improved Representation of Asymmetrical Distances with Interval Quasimetric Embeddings

Tongzhou Wang, Phillip Isola, NeurReps Workshop 2022

Interval Quasimetric Embeddings

$$d_{\theta}(x, y) \triangleq \boxed{d_{\text{IQE}}(f_{\theta}(x), f_{\theta}(y))}$$

detail skipped in this talk









Properties:


- * (Quasimetric constraints) All functions in this family are quasimetrics
- * (Universal approximation) All quasimetrics are close to one in this family

UA proof by approximating any quasimetric as cvx combination of many binary ones.

master 1 branch 0 tags

Go to file Add file Code

 ssnl update_bib	ff13149 last week	🕒 3 commits
 torchqmet	Add code	5 months ago
 .gitignore	Add code	5 months ago
 LICENSE	Add code	5 months ago
 README.md	update_bib	last week
 setup.py	Add code	5 months ago

☰ README.md 

torchqmet: PyTorch Package for Quasimetric Learning

[Tongzhou Wang](#)

This repository provides a PyTorch package for quasimetric learning --- Learning a **quasimetric** function from data.

It implements many recent quasimetric learning methods (in reverse chronological order):

- [1] Interval Quasimetric Embeddings (IQEs) ([paper](#)) ([website](#))
Wang & Isola. NeurIPS 2022 NeurReps Workshop Proceedings Track.
- [2] Metric Residual Networks (MRNs) ([paper](#))
Liu et al. arXiv 2022.

About

PyTorch Package For Quasimetric Learning

-  Readme
-  BSD-3-Clause license
-  20 stars
-  3 watching
-  0 forks

[Report repository](#)

Releases

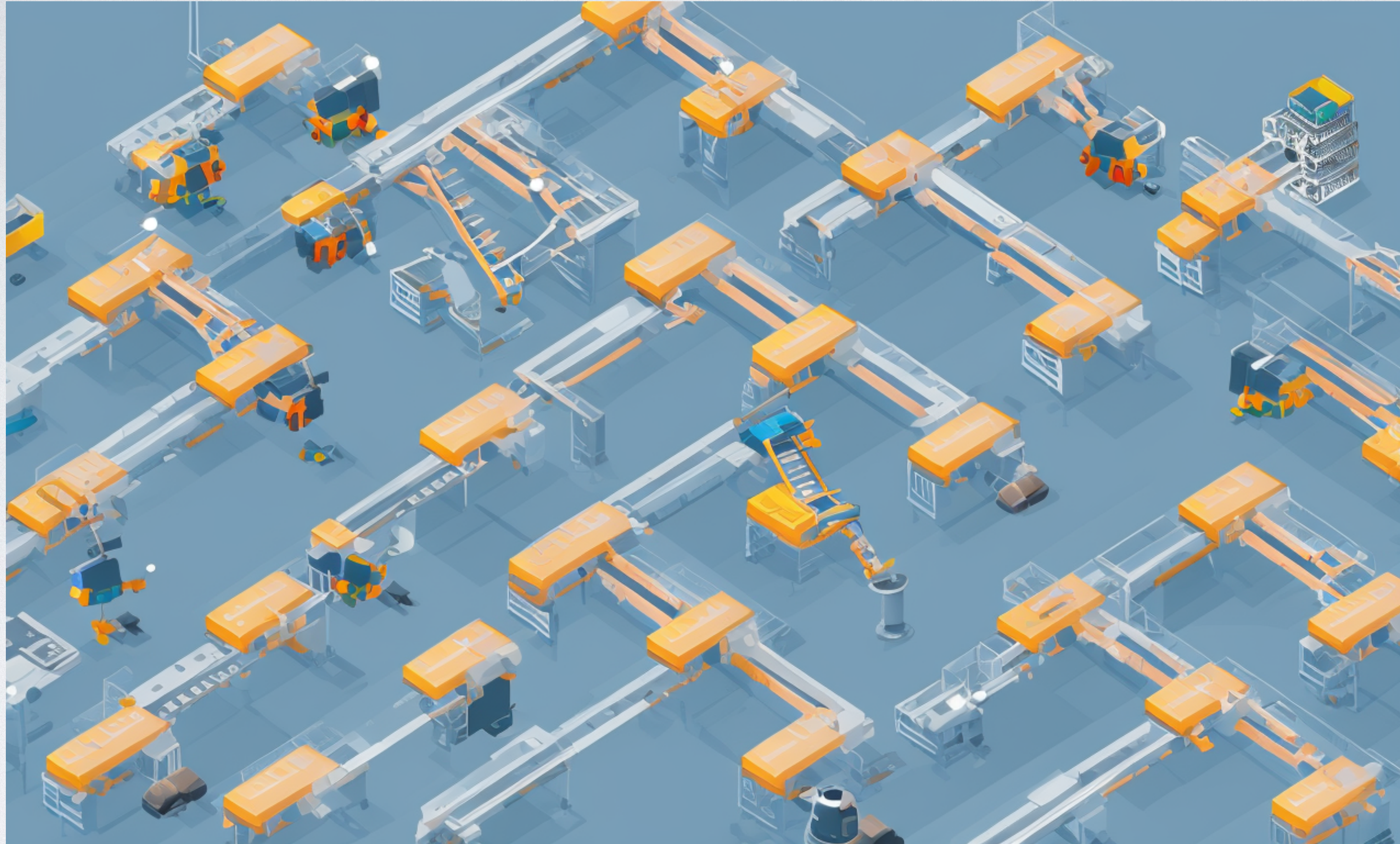
No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages





Decision-Making via **Quasimetrics**

Image by Stable Diffusion 2.1 (Prompt = "a robot factory with many complex conveyor belts; high quality flat design")

Decision-Making $V^* \in$ **Quasimetrics**

Quasimetric Embedding for Modeling
Quasimetrics

Generalist Agent via **Quasimetric** World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

Quasimetric value embedding is all you need

1. get a **Quasimetric value embedding**
where **latent distance** \equiv **true distance**
2. train a latent world model in it
3. agent plan/trained w.r.t. world model
4. profit!

Generalist Agent via **Quasimetric** World Models

✓ learn a model of how actions affect the distance/value to **ALL** goals

Quasimetric value embedding is all you need

1. get a **Quasimetric Embedding** that models V^* for all (state, goal) pairs
2. train a latent world model in it
3. agent plan/trained w.r.t. world model
4. profit!

Learn a **Quasimetric Value Embedding** V^*

- **Quasimetric Embedding** to parametrize $V(\cdot, \text{goal} = \cdot)$ in existing RL algs. (e.g., DDPG)
[ICLR 22, NeurIPS 22 workshop, papers by others at Austin & Toronto]
✗ straightforward, but **doesn't always work well**

Learn a **Quasimetric Value Embedding** V^*

- **Quasimetric Embedding** to parametrize $V(\cdot, \text{goal} = \cdot)$ in existing RL algs. (e.g., DDPG)
[ICLR 22, NeurIPS 22 workshop, papers by others at Austin & Toronto]
 - ✗ straightforward, but **doesn't always work well**
- Existing RL algorithms
 - 😓 need accurate representation of intermediate results \notin Quasimetrics
 - \implies convergence fail $\implies V^*$ estimates may not improve \implies bad-quality agents
[Sutton & Barto §11.5; Xie and Jiang 2020; Xiao et al. 2022]
 - 😓 many optimization issues b/c bootstrapping
[Fujimoto 2022; Lyle 2022]

Learn a **Quasimetric Value Embedding** V^*

- **Quasimetric Embedding** to parametrize $V(\cdot, \text{goal} = \cdot)$ in existing RL algs. (e.g., DDPG)
[ICLR 22, NeurIPS 22 workshop, papers by others at Austin & Toronto]
 - ✗ straightforward, but **doesn't always work well**
- Existing RL algorithms
 - 😓 need accurate representation of intermediate results \notin Quasimetrics
 - \implies convergence fail $\implies V^*$ estimates may not improve \implies bad-quality agents
[Sutton & Barto §11.5; Xie and Jiang 2020; Xiao et al. 2022]
 - 😓 many optimization issues b/c bootstrapping
[Fujimoto 2022; Lyle 2022]
- New algorithm for learning V^* as a quasimetric embedding?

Decision-Making wants V^*

Quasimetric Embedding parametrizes V^*

RL Algorithm Designed for Quasimetrics

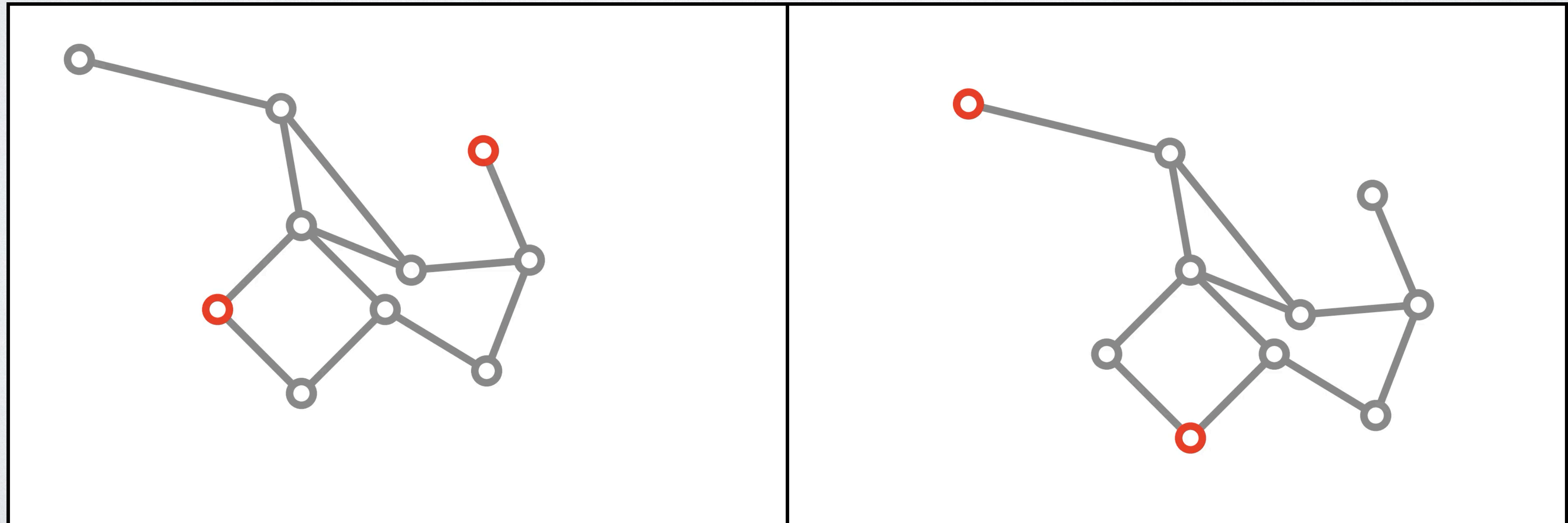
learns V^*



Quasimetric RL (QRL)

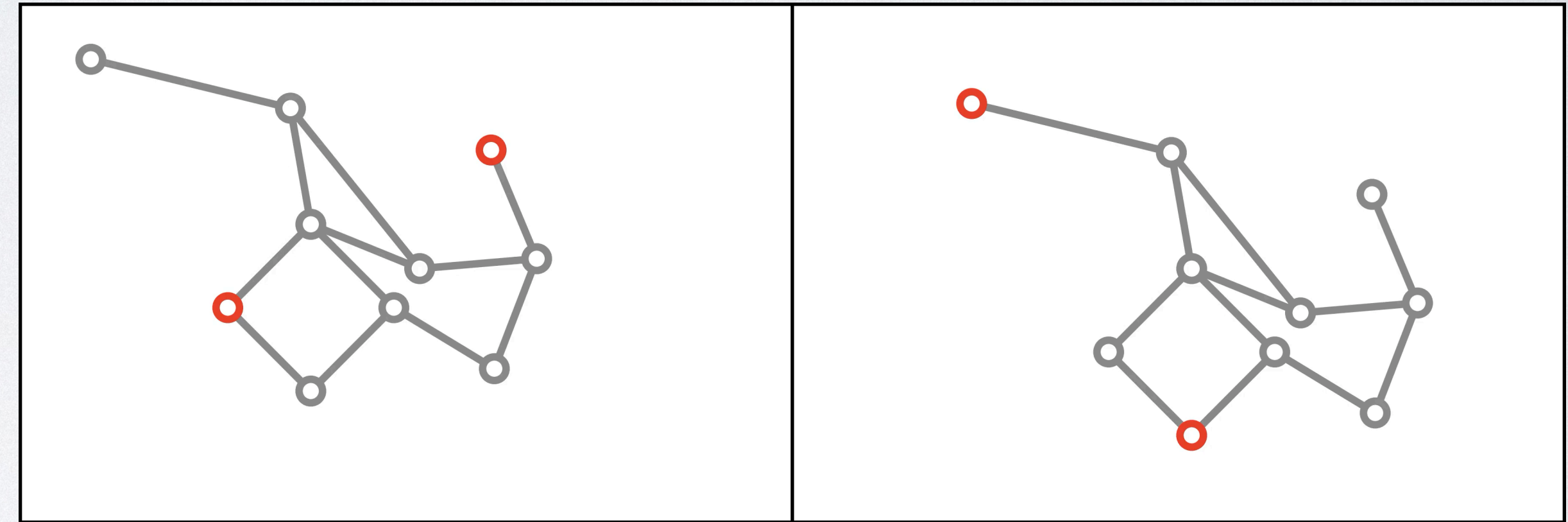
How To Learn Optimal $V^* \in \text{Quasimetrics}$

- **Q:** Two objects connected by multiple chains. How to find length of the **shortest** chain connecting them?



How To Learn Optimal $V^* \in \text{Quasimetrics}$

- **Q:** Two objects connected by multiple chains. How to find length of the **shortest** chain connecting them?
- **A:** Pull them apart. Then measure!



How To Learn Optimal $V^* \in \text{Quasimetrics}$

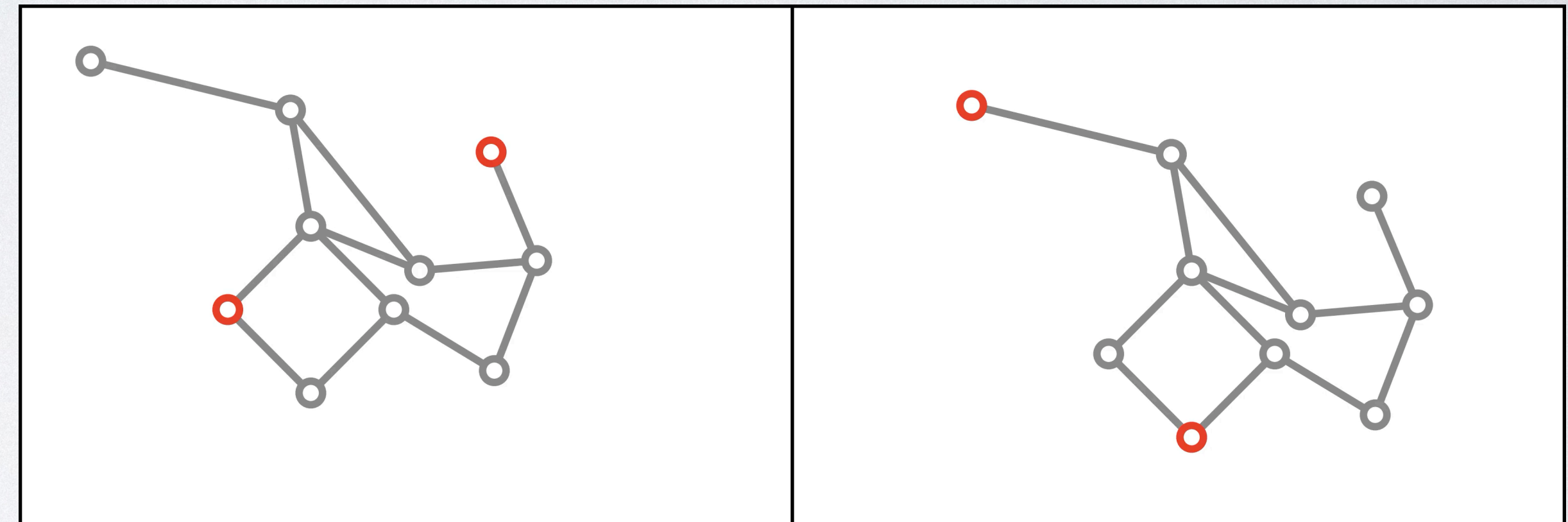
- **Q:** Two objects connected by multiple chains. How to find length of the **shortest** chain connecting them?

- **A:** Pull them apart. Then measure!

- This relies on

1. **Triangle inequality** of our physical Euclidean space

2. Each link of chains has **fixed length** unaffected by our actions



How To Learn Optimal $V^* \in \text{Quasimetrics}$

- **Q:** Two objects connected by multiple chains. How to find length of the **shortest** chain connecting them?

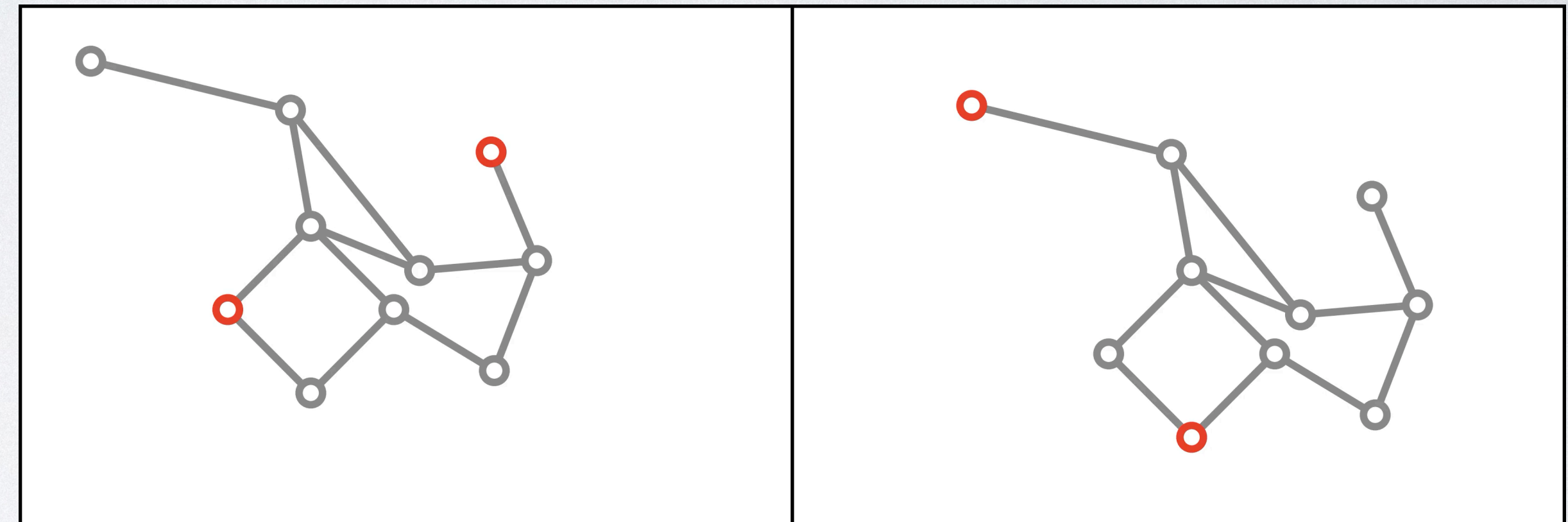
- **A:** Pull them apart. Then measure!

- This relies on

1. **Triangle inequality** of our physical Euclidean space

2. Each link of chains has **fixed length** unaffected by our actions

- **QRL:** Do the same with a **quasimetric embedding** that can approx. any environment!



How To Learn Optimal $V^* \in \text{Quasimetrics}$

- **Q:** Two objects connected by multiple chains. How to find length of the **shortest** chain connecting them?

- **A:** Pull them apart. Then measure!

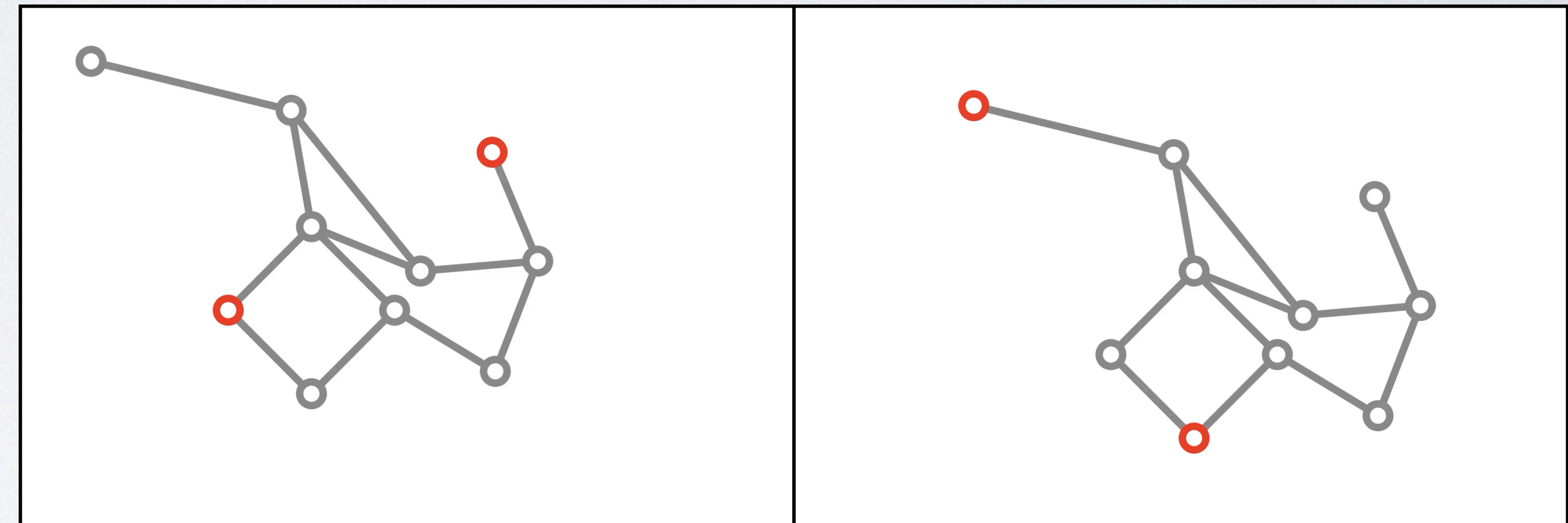
Optimize over all (state, goal)

- This relies on

1. **Triangle inequality** of our physical Euclidean space

2. Each link of chains has **fixed length** unaffected by our actions

- **QRL:** Do the same with a **quasimetric embedding** that **can approx. any environment!**



How To Learn Optimal $V^* \in \text{Quasimetrics}$

- **Q:** Two objects connected by multiple chains. How to find length of the **shortest** chain connecting them?

- **A:** Pull them apart. Then measure!

..... Optimize over all (state, goal)

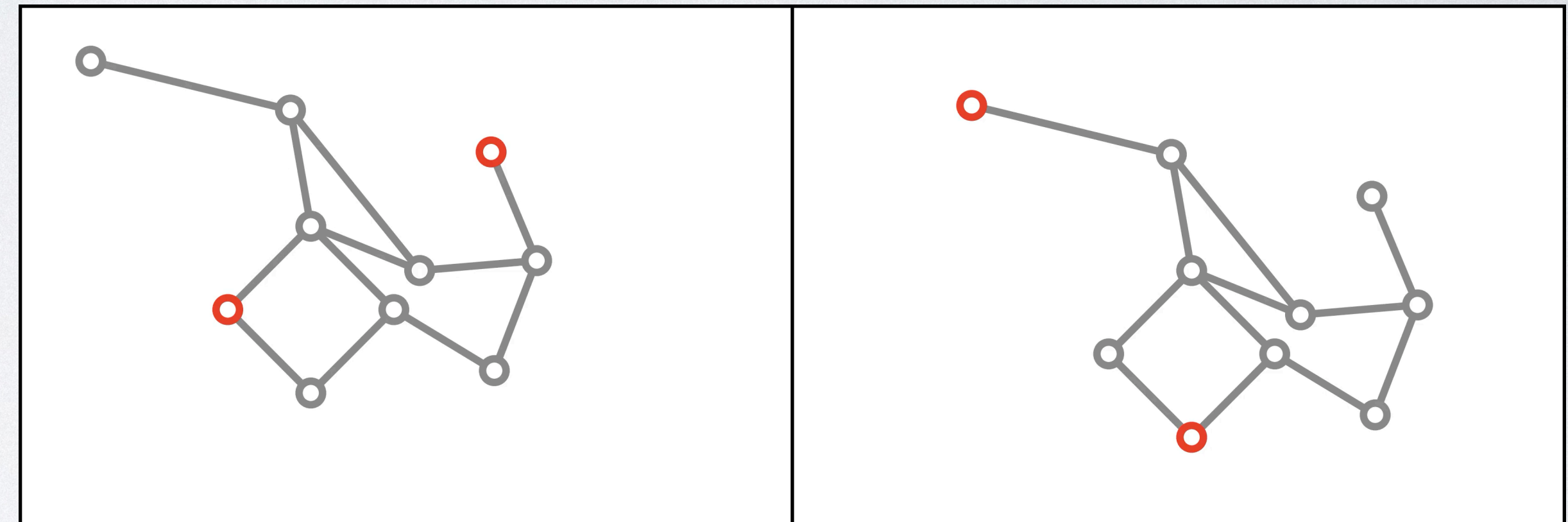
- This relies on

1. Triangle inequality of our physical Euclidean space

..... Always satisfied via **quasimetric emb.**

2. Each link of chains has **fixed length** unaffected by our actions

- **QRL:** Do the same with a **quasimetric embedding** that **can approx. any environment!**



How To Learn Optimal $V^* \in \text{Quasimetrics}$

- **Q:** Two objects connected by multiple chains. How to find length of the **shortest** chain connecting them?

- **A:** Pull them apart. Then measure!

..... Optimize over all (state, goal)

- This relies on

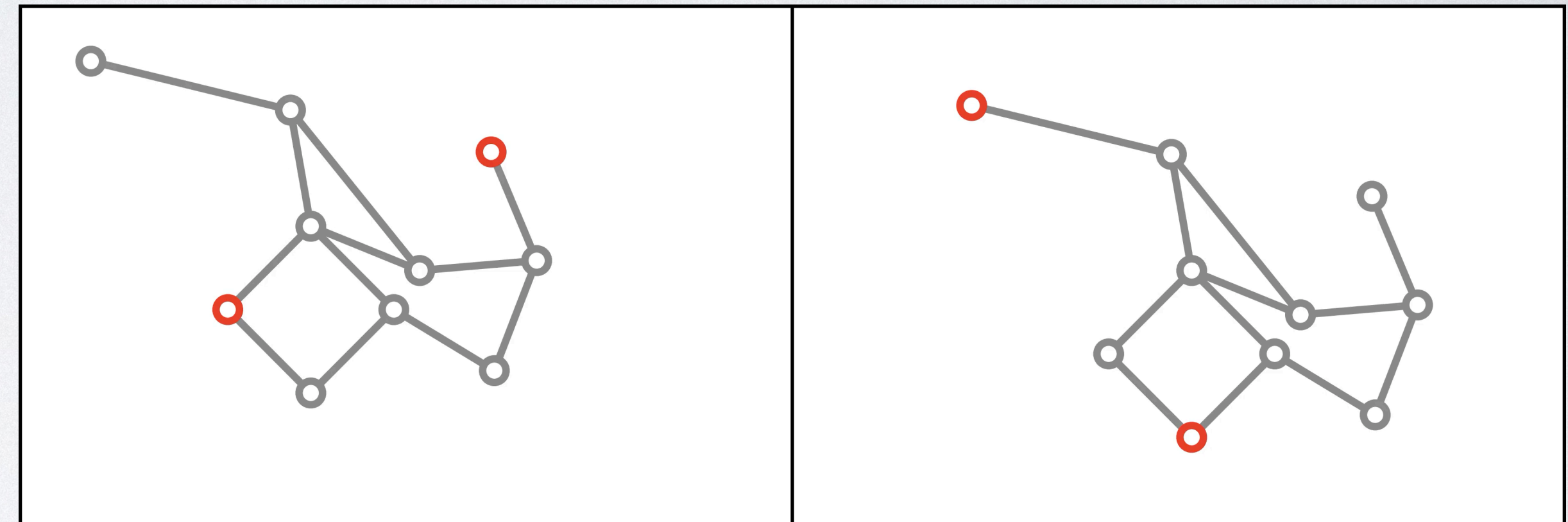
1. Triangle inequality of our physical Euclidean space

..... Always satisfied via **quasimetric emb.**

2. Each link of chains has fixed length unaffected by our actions

..... Ensure as a constraint

- **QRL:** Do the same with a **quasimetric embedding** that **can approx. any environment!**



Quasimetric RL $\implies V^*$ Quasimetric Value Embedding

Given ways to sample (e.g., from a dataset / replay buffer)

$$(s, a, s', \text{cost}) \sim p_{\text{transition}}$$

$$s \sim p_{\text{state}}$$

$$s_{\text{goal}} \sim p_{\text{goal}},$$

(transitions)

(random state)

(random goal)

Quasimetric RL $\implies V^*$ Quasimetric Value Embedding

Given ways to sample (e.g., from a dataset / replay buffer)

$$(s, a, s', \text{cost}) \sim p_{\text{transition}}$$

$$s \sim p_{\text{state}}$$

$$s_{\text{goal}} \sim p_{\text{goal}},$$

(transitions)

(random state)

(random goal)

Quasimetric RL (**QRL**) optimizes a Quasimetric emb. d_θ as (negated) value function

$d(s_1, s_2) = \text{optimal cost } s_1 \rightarrow s_2$

Quasimetric RL $\implies V^*$ Quasimetric Value Embedding

Given ways to sample (e.g., from a dataset / replay buffer)

$$\begin{aligned}(s, a, s', \text{cost}) &\sim p_{\text{transition}} && \text{(transitions)} \\ s &\sim p_{\text{state}} && \text{(random state)} \\ s_{\text{goal}} &\sim p_{\text{goal}}, && \text{(random goal)}\end{aligned}$$

Quasimetric RL (QRL) optimizes a Quasimetric emb. d_θ as (negated) value function:

$$\begin{aligned}\max_{\theta} \mathbb{E}_{\substack{s \sim p_{\text{state}} \\ g \sim p_{\text{goal}}}} [d_\theta(s, g)] &&& \text{(maximize over all pairs)} \\ \text{subject to } \mathbb{E}_{(s, a, s', \text{cost}) \sim p_{\text{transition}}} [\text{relu}(d_\theta(s, s') - \text{cost})^2] \leq \epsilon^2 &&& \text{(not overestimate local cost)} \\ &&& \sqcup \\ &&& \epsilon > 0 \text{ small}\end{aligned}$$

Quasimetric RL $\implies V^*$ Quasimetric Value Embedding

Given ways to sample (e.g., from a dataset / replay buffer)

$$\begin{aligned}(s, a, s', \text{cost}) &\sim p_{\text{transition}} && \text{(transitions)} \\ s &\sim p_{\text{state}} && \text{(random state)} \\ s_{\text{goal}} &\sim p_{\text{goal}}, && \text{(random goal)}\end{aligned}$$

Quasimetric RL (QRL) optimizes a Quasimetric emb. d_θ as (negated) value function:

$$\begin{aligned}\max_{\theta} \mathbb{E}_{\substack{s \sim p_{\text{state}} \\ g \sim p_{\text{goal}}}} [d_\theta(s, g)] &&& \text{(maximize over all pairs)} \\ \text{subject to } \mathbb{E}_{(s, a, s', \text{cost}) \sim p_{\text{transition}}} [\text{relu}(d_\theta(s, s') - \text{cost})^2] &\leq \epsilon^2 && \text{(not overestimate local cost)} \\ &\sqcup && \epsilon > 0 \text{ small}\end{aligned}$$

[Thms. 2 & 3] With sufficient data & model capacity, QRL recovers V^* .

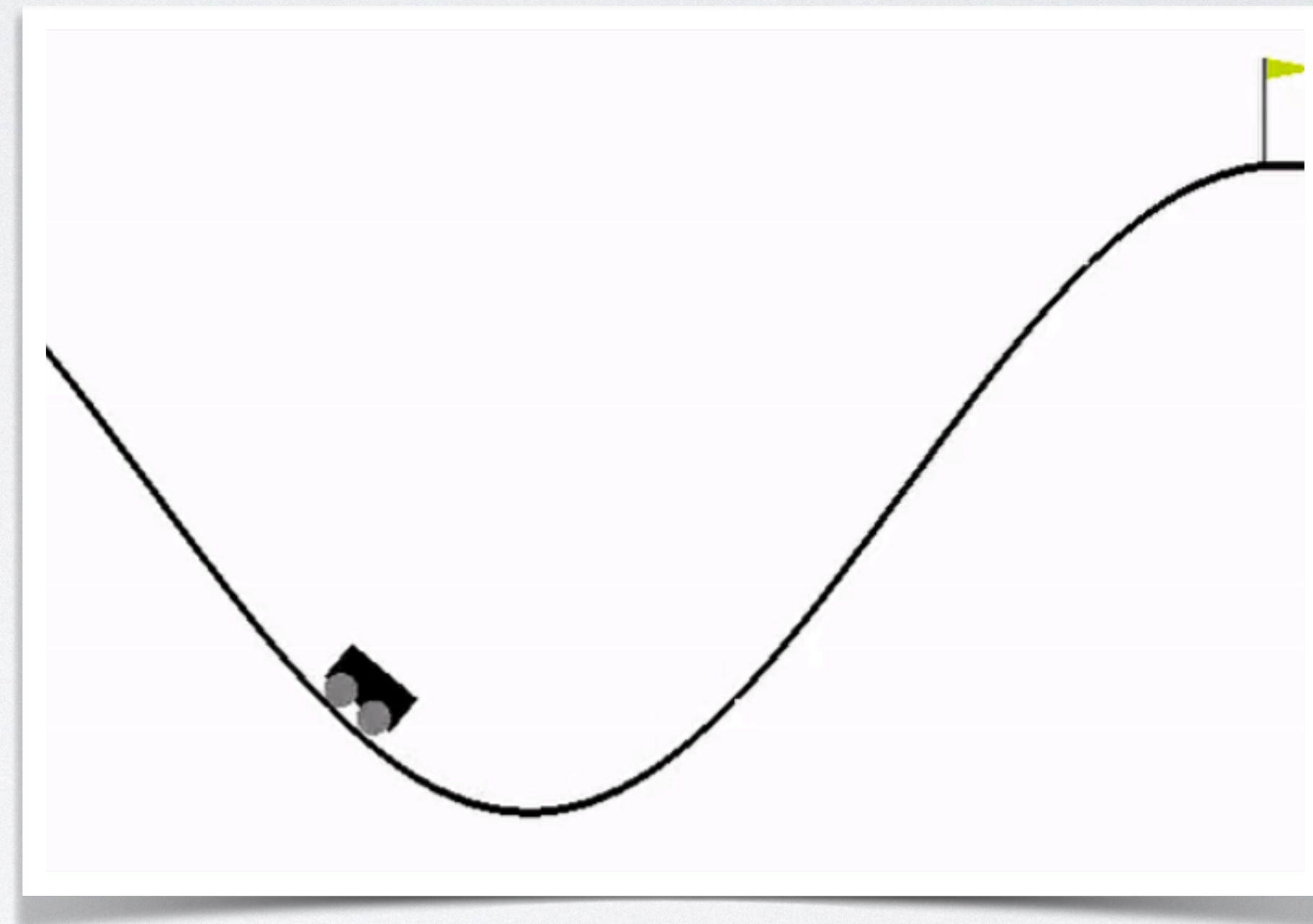
Proof by (1) $\mathbb{E}[d] > C \implies d$ recovers V^* with high prob. (2) constructing $\mathbb{E}[d_\theta] = C$

QRL on Discrete Mountain Car

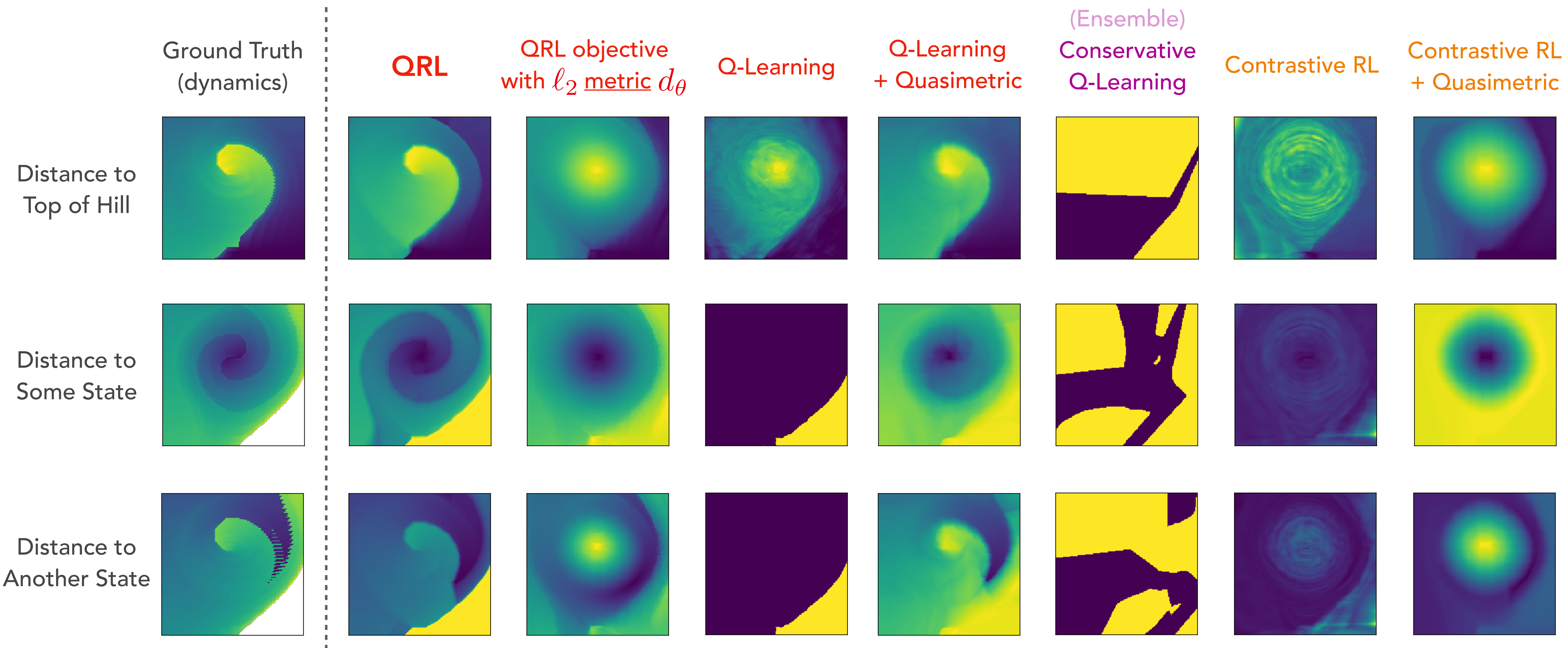
State: [position, velocity],
discretized into 160×160 grid.

Action: Discrete $\{-1, 0, 1\}$.
Horizontal acceleration.

Dataset: **offline** trajectory
dataset w/ **random** actor.



QRL Recovers V^* on Discrete MountainCar



From V^* to Quasimetric World Model and Policy

(Assume Deterministic Dynamics)

- A learned latent transition T jointly trained with value function d_θ w.r.t.

$$\mathcal{L}_{\text{transition}}(s, a, s') \triangleq \frac{d_{\text{latent}}(T(z_s, a), z_{s'})^2 + d_{\text{latent}}(z_{s'}, T(z_s, a))^2}{}$$

⋮
Value-Aware Model Training

Optimizes quasimetric distance (value)

NOT reconstruction

From V^* to Quasimetric World Model and Policy

(Assume Deterministic Dynamics)

- A learned latent transition T jointly trained with value function d_θ w.r.t.

$$\mathcal{L}_{\text{transition}}(s, a, s') \triangleq \frac{d_{\text{latent}}(T(z_s, a), z_{s'})^2 + d_{\text{latent}}(z_{s'}, T(z_s, a))^2}{\quad}$$

⋮
Value-Aware Model Training

Optimizes quasimetric distance (value)

NOT reconstruction

Local loss (on transitions) + Quasimetric \implies Error guarantee on global pairs

From V^* to Quasimetric World Model and Policy

(Assume Deterministic Dynamics)

- A learned latent transition T jointly trained with value function d_θ w.r.t.

$$\mathcal{L}_{\text{transition}}(s, a, s') \triangleq \frac{d_{\text{latent}}(T(z_s, a), z_{s'})^2 + d_{\text{latent}}(z_{s'}, T(z_s, a))^2}{2}$$

⋮
Value-Aware Model Training

Optimizes quasimetric distance (value)

NOT reconstruction

Local loss (on transitions) + Quasimetric \implies Error guarantee on global pairs

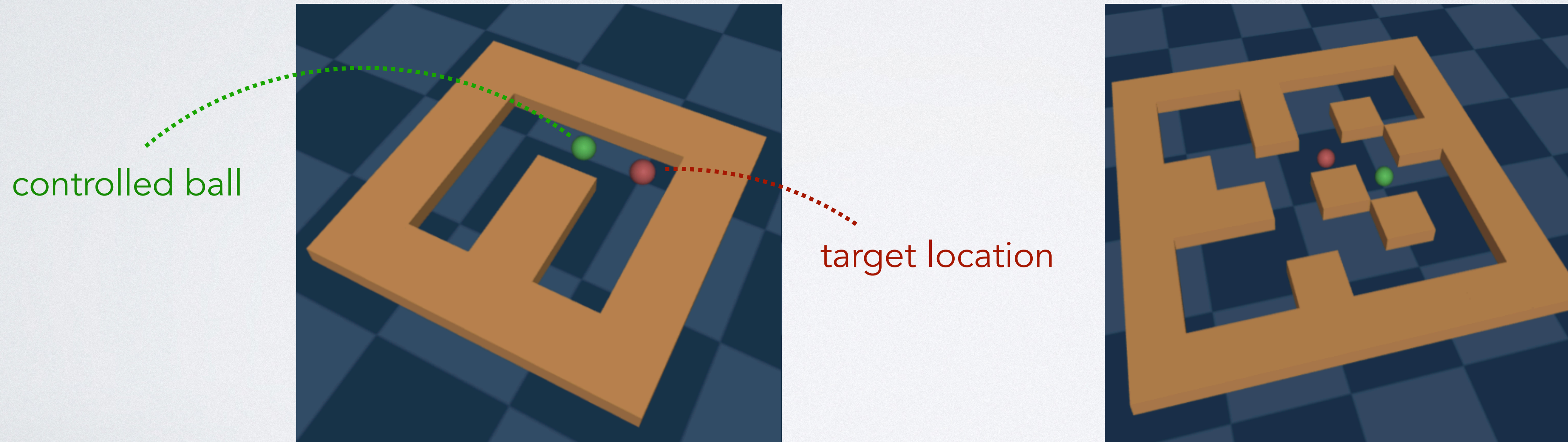
- Optimize policy via **Quasimetric World Model**

What Makes **QRL** Special?

	<u>Optimality</u>	<u>Goal Reaching</u>	Quasimetric structure	<u>TD-Free</u>	Stochastic Dynamics
Value-Based RL (e.g., IQL, MSG, SAC, CQL)	(with bootstrapping)	(possible via relabelling)			
Quasimetric + Value-Based RL					
<u>QRL</u>					(empirically works well)
Contrastive RL	(with bootstrapping)				
RL via Supervised/Traj. Learning (e.g., DT, GCBC, Diffuser)					

Benchmarking **QRL** (Offline)

- Offline RL. Maze2D: Guide a ball through a maze toward target location.



Benchmarking QRL (Offline)

- Offline RL. Maze2D: Guide a ball through a maze toward target location.

Environment	QRL	Contrastive RL	Ensemble Q-Learning		Planning	Trajectory Modelling		
			MSG (#critic = 64)	MSG + HER (#critic = 64)	MPPI with GT Dynamics	Diffuser	Diffuser with Handcoded Controller	
Single-Goal	large	185.26 ± 28.46	81.65 ± 43.79	159.30 ± 49.40	59.26 ± 46.70	5.1	7.98 ± 1.54	128.13 ± 2.59
	medium	148.48 ± 46.75	10.11 ± 0.99	57.00 ± 17.20	75.77 ± 9.02	10.2	9.48 ± 2.21	127.64 ± 1.47
	umaze	47.40 ± 23.72	95.11 ± 46.23	101.10 ± 26.30	55.64 ± 31.82	33.2	44.03 ± 2.25	113.91 ± 3.27
	Average	127.05	62.29	105.80	63.56	16.17	20.50	123.23
Multi-Goal	large	199.19 ± 4.07	172.64 ± 5.13	—	44.57 ± 25.30	8	13.09 ± 1.00	146.94 ± 2.50
	medium	161.91 ± 8.10	137.01 ± 6.26	—	99.76 ± 9.83	15.4	19.21 ± 3.56	119.97 ± 1.22
	umaze	134.11 ± 12.56	142.43 ± 11.99	—	27.90 ± 10.39	41.2	56.22 ± 3.90	128.53 ± 1.00
	Average	165.07	150.69	—	57.41	21.53	29.51	131.81

Benchmarking QRL (Offline)

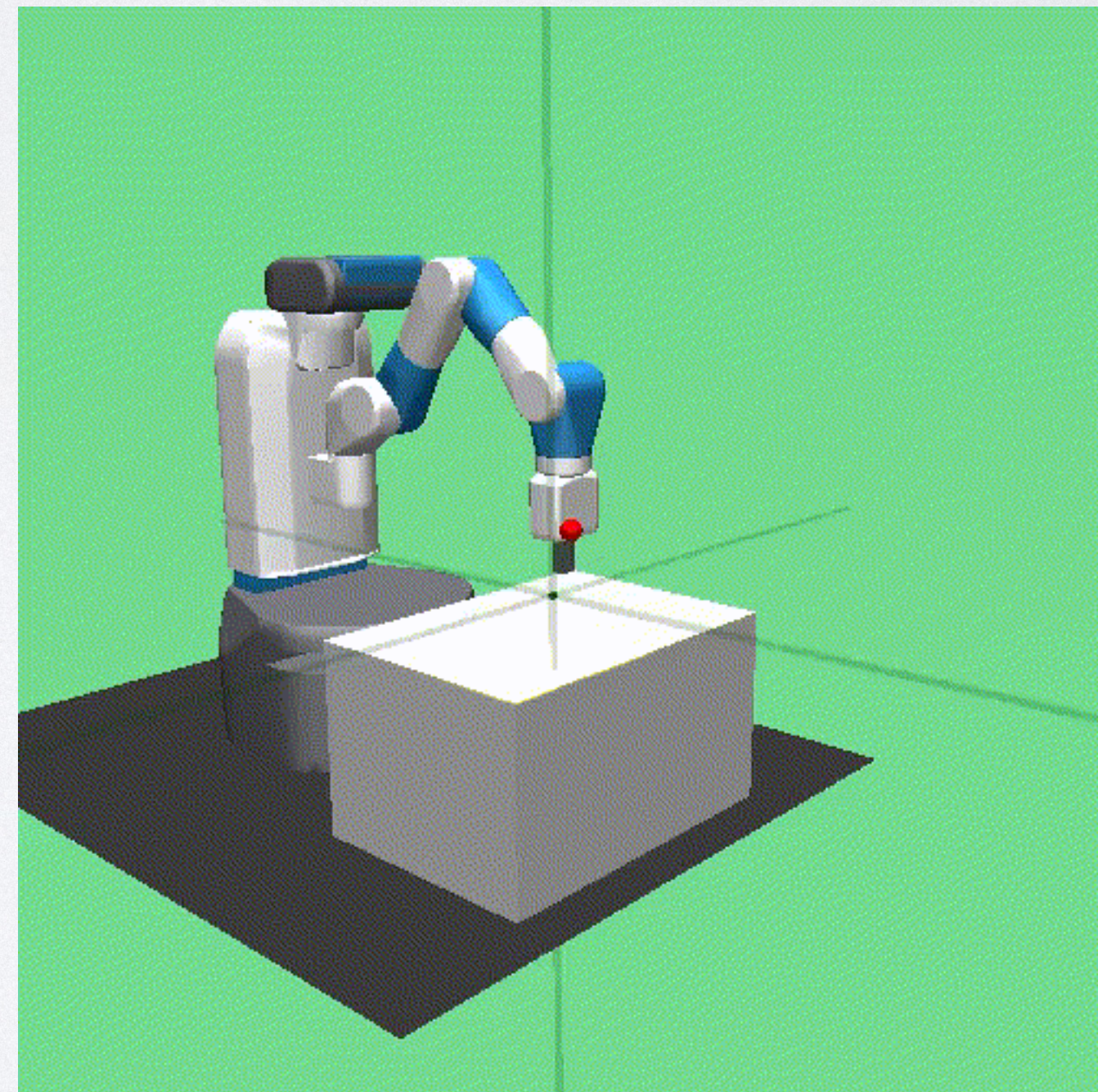
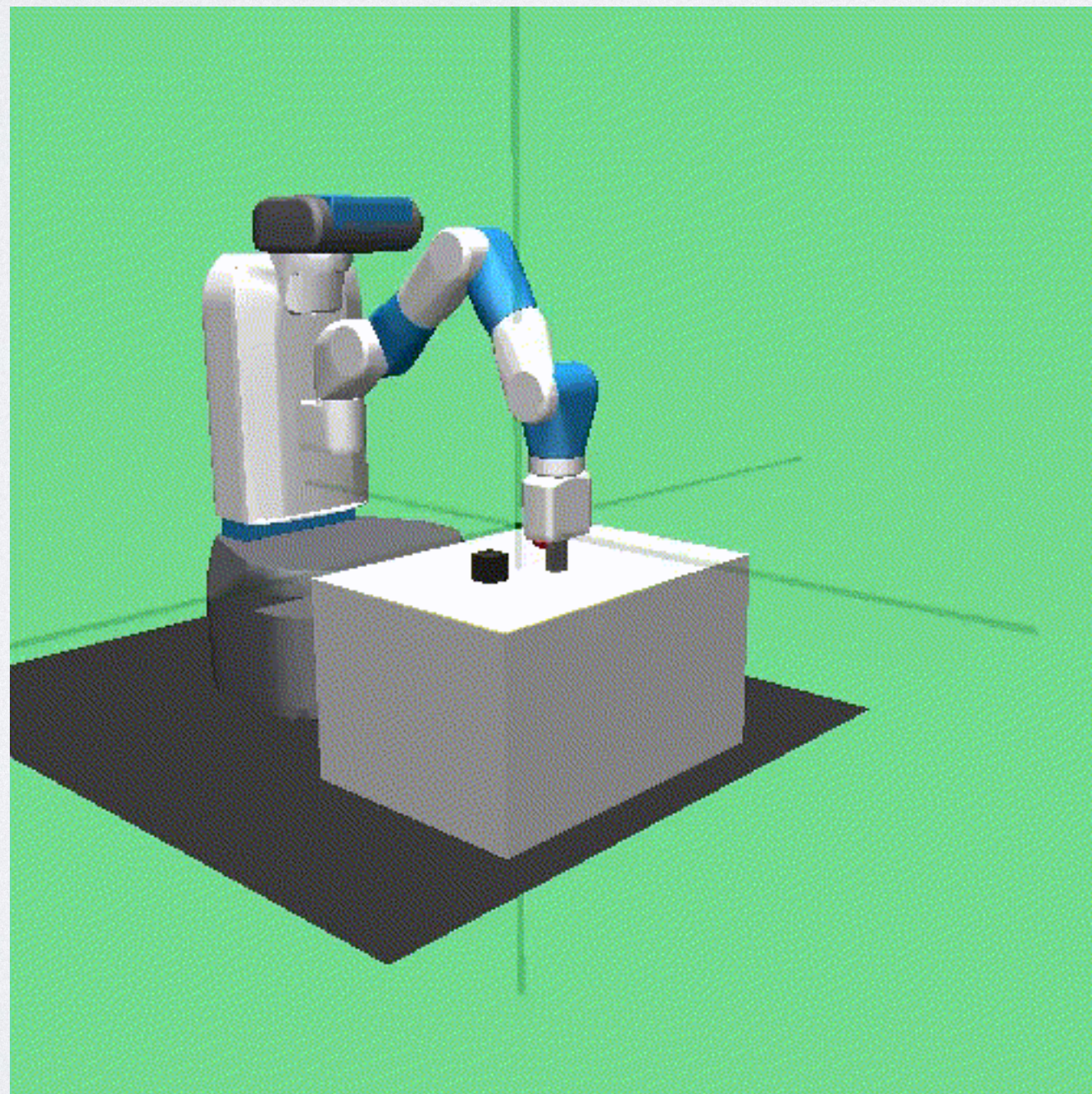
- Offline RL. Maze2D: Guide a ball through a maze toward target location.



Environment	QRL	Contrastive RL	MSG (#critic = 64)	MSG + HER (#critic = 64)	MPPI with GT Dynamics	MPPI with QRL Value	Diffuser	Diffuser with QRL Value Guidance	Diffuser with Handcoded Controller	
Single-Goal	large	185.26 ± 28.46	81.65 ± 43.79	159.30 ± 49.40	59.26 ± 46.70	5.1	4.67 ± 5.31	7.98 ± 1.54	11.33 ± 1.48	128.13 ± 2.59
	medium	148.48 ± 46.75	10.11 ± 0.99	57.00 ± 17.20	75.77 ± 9.02	10.2	60.89 ± 40.38	9.48 ± 2.21	10.52 ± 3.26	127.64 ± 1.47
	umaze	47.40 ± 23.72	95.11 ± 46.23	101.10 ± 26.30	55.64 ± 31.82	33.2	45.88 ± 9.32	44.03 ± 2.25	42.19 ± 4.23	113.91 ± 3.27
	Average	127.05	62.29	105.80	63.56	16.17	37.15	20.50	21.35	123.23
Multi-Goal	large	199.19 ± 4.07	172.64 ± 5.13	—	44.57 ± 25.30	8	54.04 ± 7.47	13.09 ± 1.00	21.78 ± 2.86	146.94 ± 2.50
	medium	161.91 ± 8.10	137.01 ± 6.26	—	99.76 ± 9.83	15.4	71.24 ± 6.69	19.21 ± 3.56	33.68 ± 2.82	119.97 ± 1.22
	umaze	134.11 ± 12.56	142.43 ± 11.99	—	27.90 ± 10.39	41.2	84.72 ± 7.69	56.22 ± 3.90	69.49 ± 3.85	128.53 ± 1.00
	Average	165.07	150.69	—	57.41	21.53	70.00	29.51	41.65	131.81

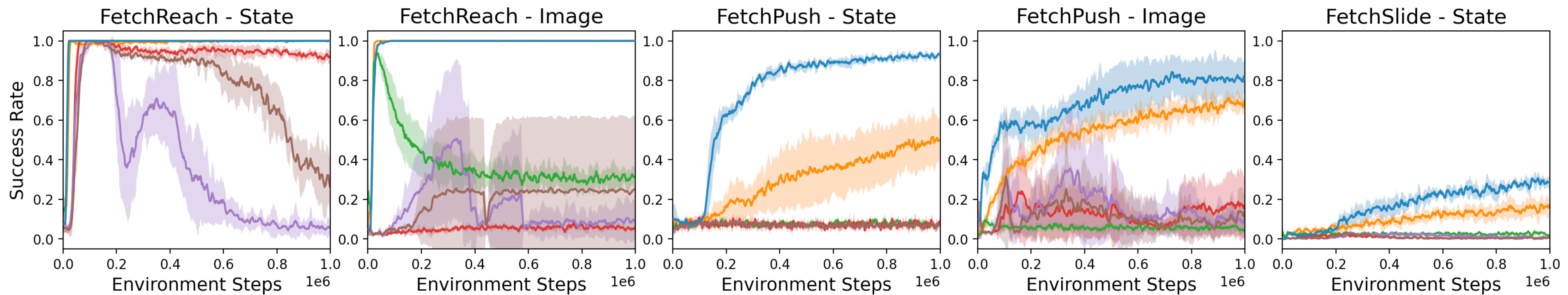
Benchmarking **QRL** (Online)

- Online RL. GCRL: Control a robot to perform tasks, e.g., pushing a block.
- More complex environments. Continuous actions.



Benchmarking QRL (Online)

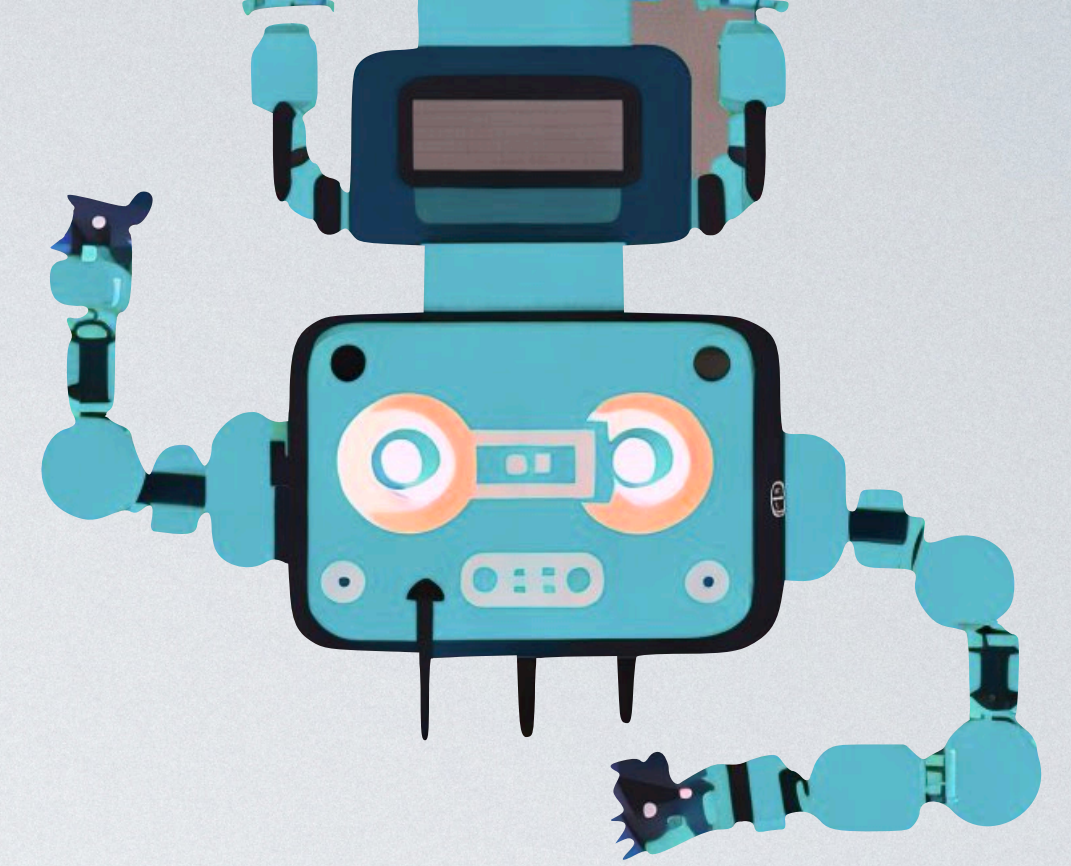
- Online RL. GCRL: Control a robot to perform tasks, e.g., pushing a block.
- More complex environments. Continuous actions.



— Quasimetric RL — Contrastive RL — Goal-Conditioned Behavior Cloning (GCBC) — DDPG + HER — DDPG + HER + Quasimetric (MRN) (Method by Liu et al. (2022)) — DDPG + HER + Quasimetric (IQE) (Method by Liu et al. (2022))

Continuous actions
Quasimetric with TD fails

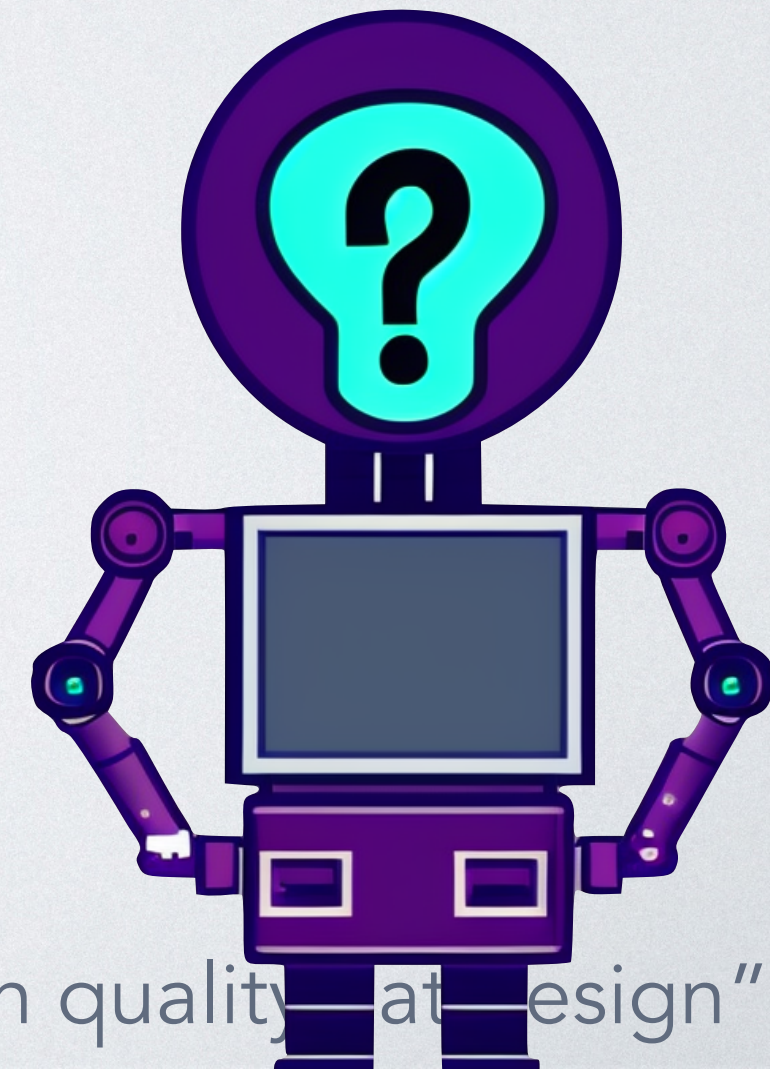
What's Next?



- **Aspects of QRL.** QRL is a representation learning method (encoder), and learns a world model (latent transition). Analyses these aspects!
- **Multi-Environment QRL**
- **Quasimetrics in RL.** Quasimetric-aware actor/planner? Exploration?
- **Exploration for structure learning**
- **RL algorithms designed for quasimetric/other structures**

Papers `\setminusminus` This Talk

- **Building Quasimetric Embeddings: Desiderata, Constructions, Analyses**
[ICLR 22; NeurReps 22]
- **Evaluating Quasimetric Embeddings on Learning General Quasimetric Spaces**
[ICLR 22; NeurReps 22]
- **Details on World Model (and thus Q-function) Training** [ICML 23]
- **Differences with Contrastive Approaches** [ICML 23]
- **Improved Learning Dynamics Over Regular Value-based Learning**
[ICML 23]



Thank You!

- **On the Learning and Learnability of Quasimetrics**
Tongzhou Wang, Phillip Isola. ICLR 2022
- **Improved Representation of Asymmetrical Distances with Interval Quasimetric Embeddings**
Tongzhou Wang, Phillip Isola. NeurIPS 2022 NeurReps Workshop
- **Optimal Goal-Reaching Reinforcement Learning via Quasimetric Learning**
Tongzhou Wang, Antonio Torralba, Phillip Isola, Amy Zhang. ICML 2023



Tongzhou Wang



Antonio Torralba



Phillip Isola



Amy Zhang