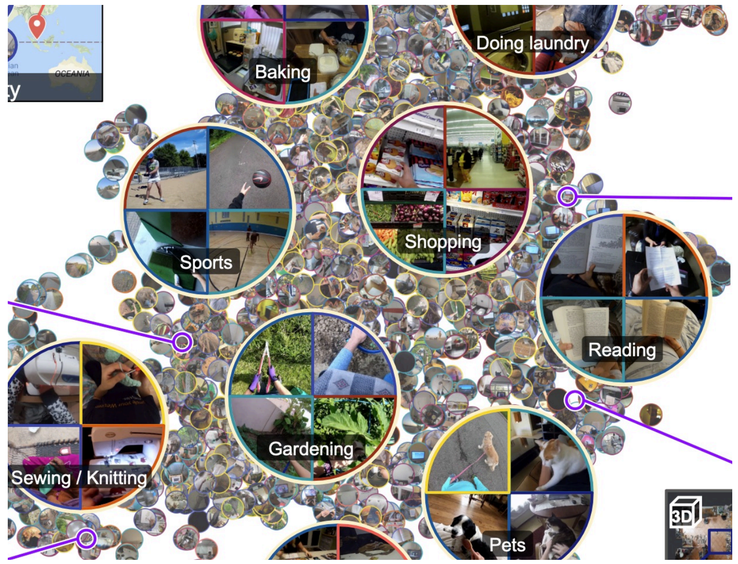
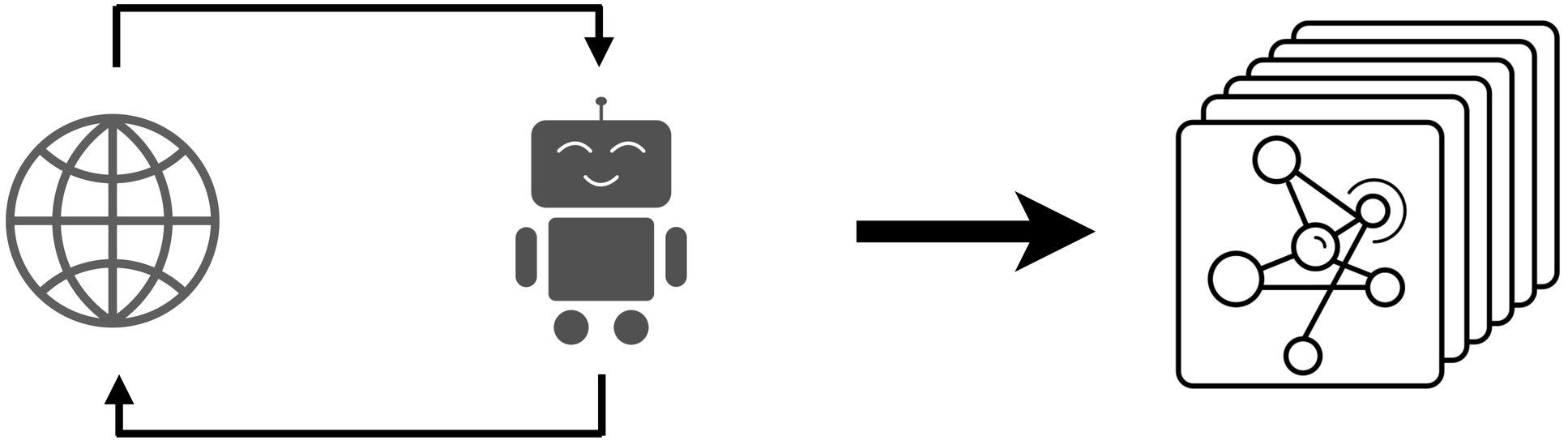
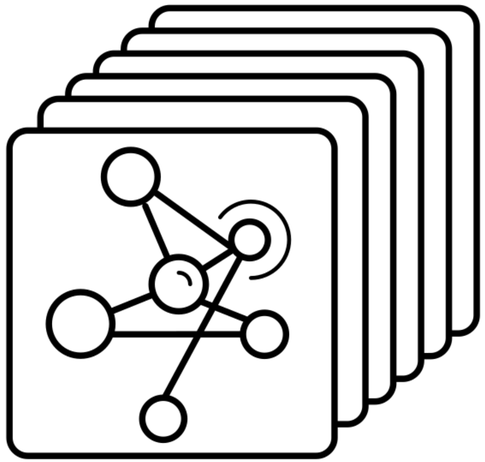
Offline RL Policies Should be Trained to be Adaptive

Dibya Ghosh, Anurag Ajay, Pulkit Agrawal, Sergey Levine

Learning to make decisions from large datasets

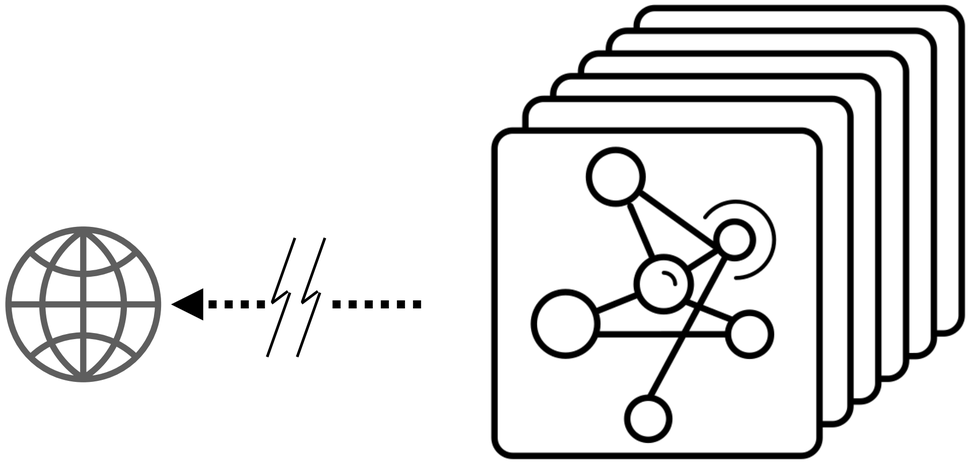






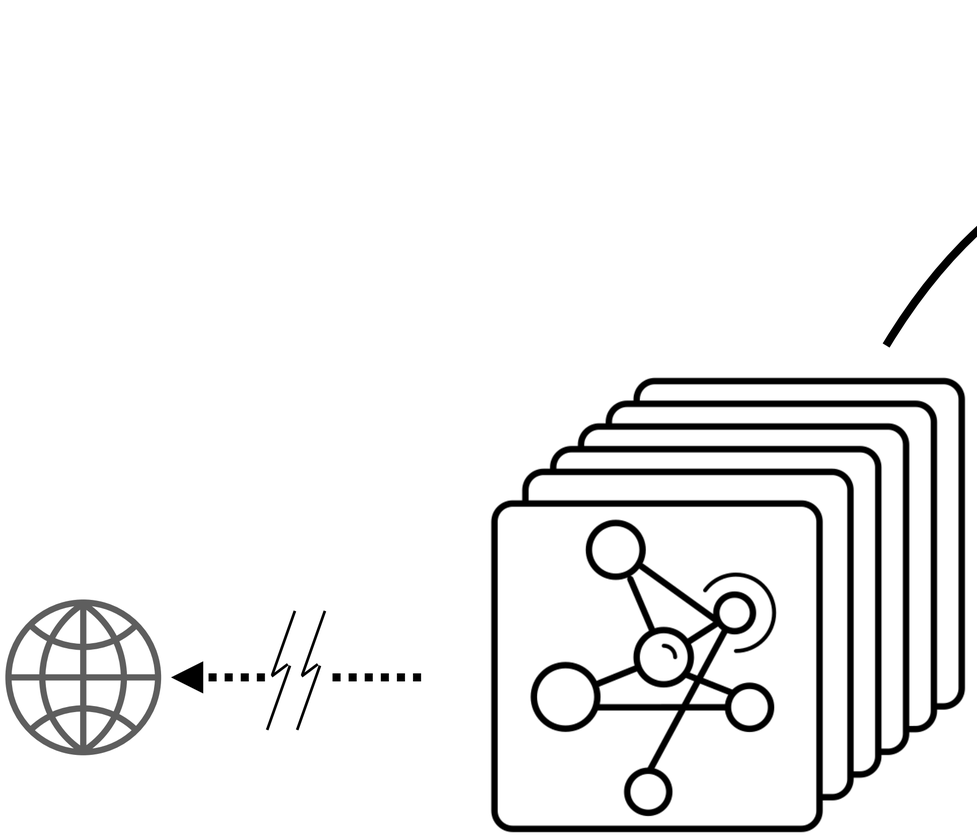
Learn a policy π from dataset $\mathscr{D}$ that maximizes

Return(🌐 , π)



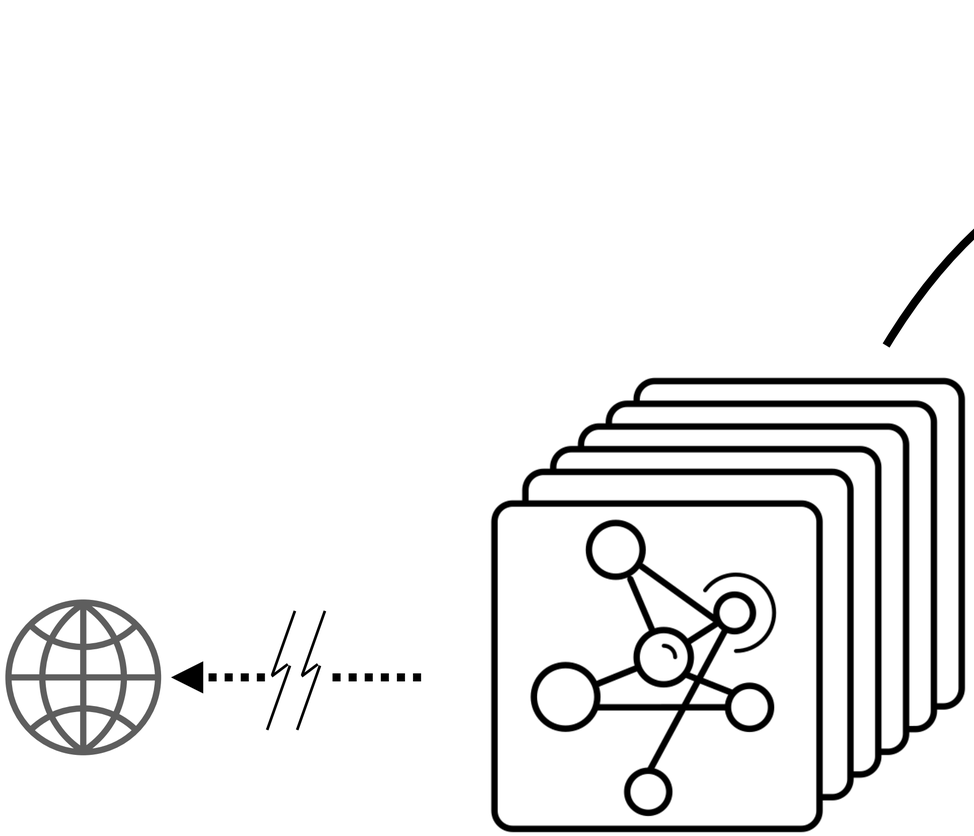
Learn a policy π from dataset $\mathscr{D}$ that maximizes

Return(, π)



Learn a policy π from dataset $\mathscr{D}$ that maximizes

Return(, π)



$$\max_{\pi(a|s)} \text{ConservativeReturn}(\pi)$$

Conservatism: Do the thing you know best

Conservatism: Do the thing you know best

Is this optimal?

$$\max_{\pi(a|s)} \text{ConservativeReturn}(\pi)$$

Conservatism: Do the thing you know best

Is this optimal?

$$\max_{\pi(a|s)} \text{ConservativeReturn}(\pi)$$

State-based policies are not enough.
We need adaptation!

$$\max_{\pi(a|s)} \text{ConservativeReturn}(\pi)$$

Offline RL agents should be adaptive under uncertainty

$$\max_{\pi(a|s)} \text{ConservativeReturn}(\pi)$$

Offline RL agents should be adaptive under uncertainty

$$\max_{\pi} \text{ConservativeReturn}(\pi)$$

~~$\pi(a|s)$~~

$\pi(a \mid h)$

Policies that have memory

Offline RL agents should be adaptive under uncertainty

Objectives that
promote adaptation

AdaptiveReturn

$\max$ ~~ConservativeReturn~~(π)

~~$\pi(a|s)$~~

$\pi(a \mid h)$

Policies that have
memory

Offline RL agents should be adaptive under uncertainty

Objectives that
promote adaptation

AdaptiveReturn

$$\max \text{ConservativeReturn}(\pi)$$

$\pi(a|s)$

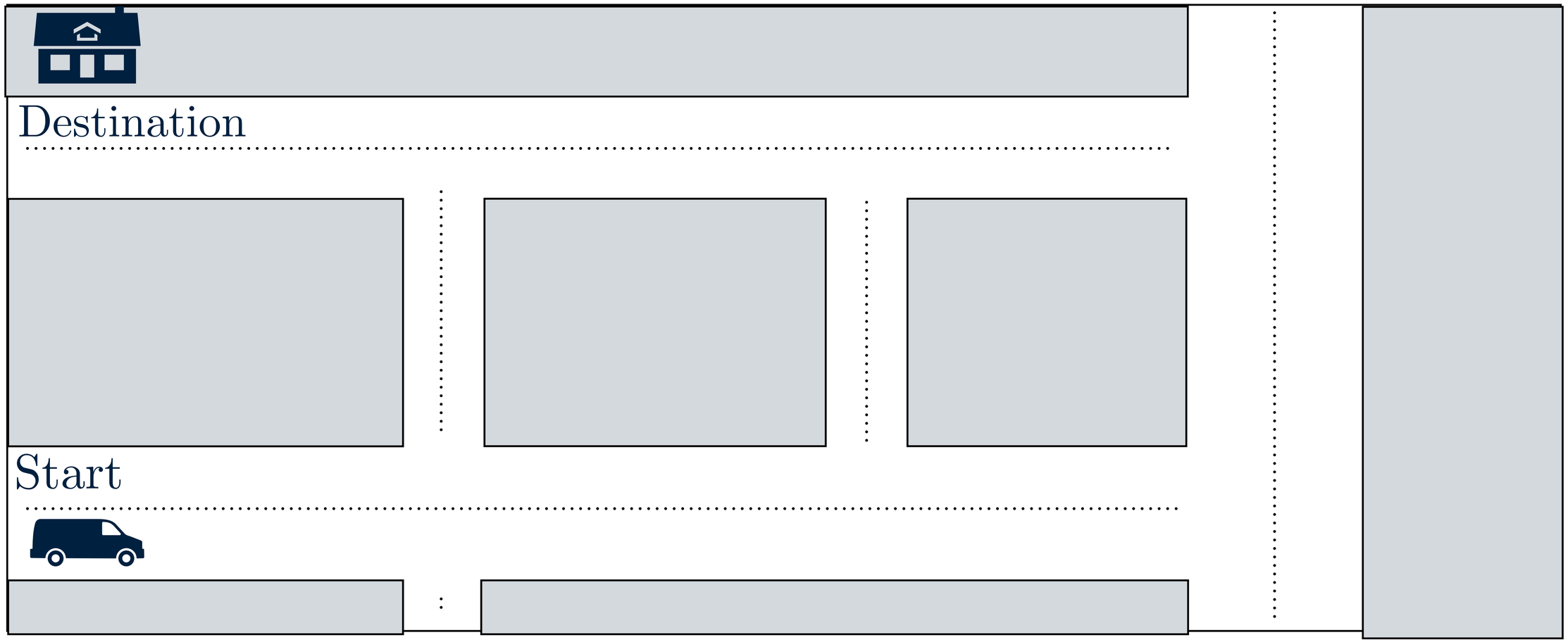
$\pi(a \mid h)$

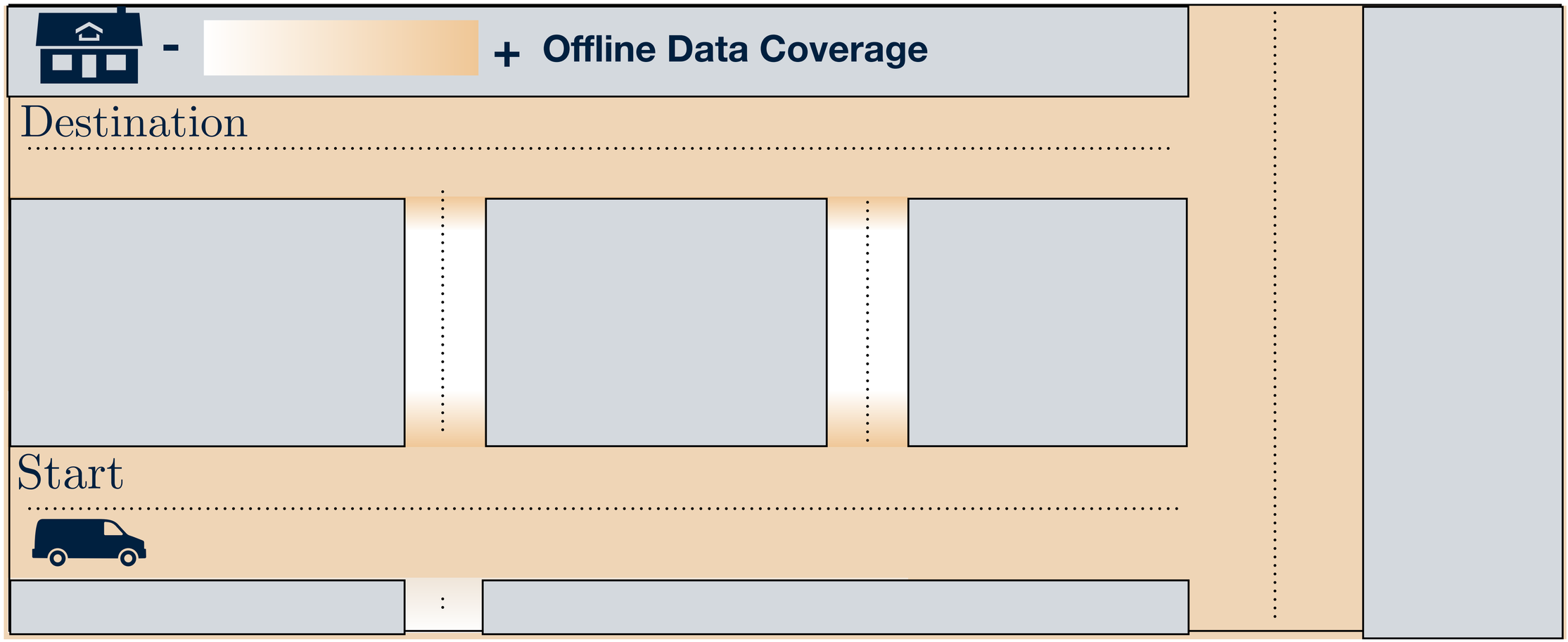
Policies that have
memory

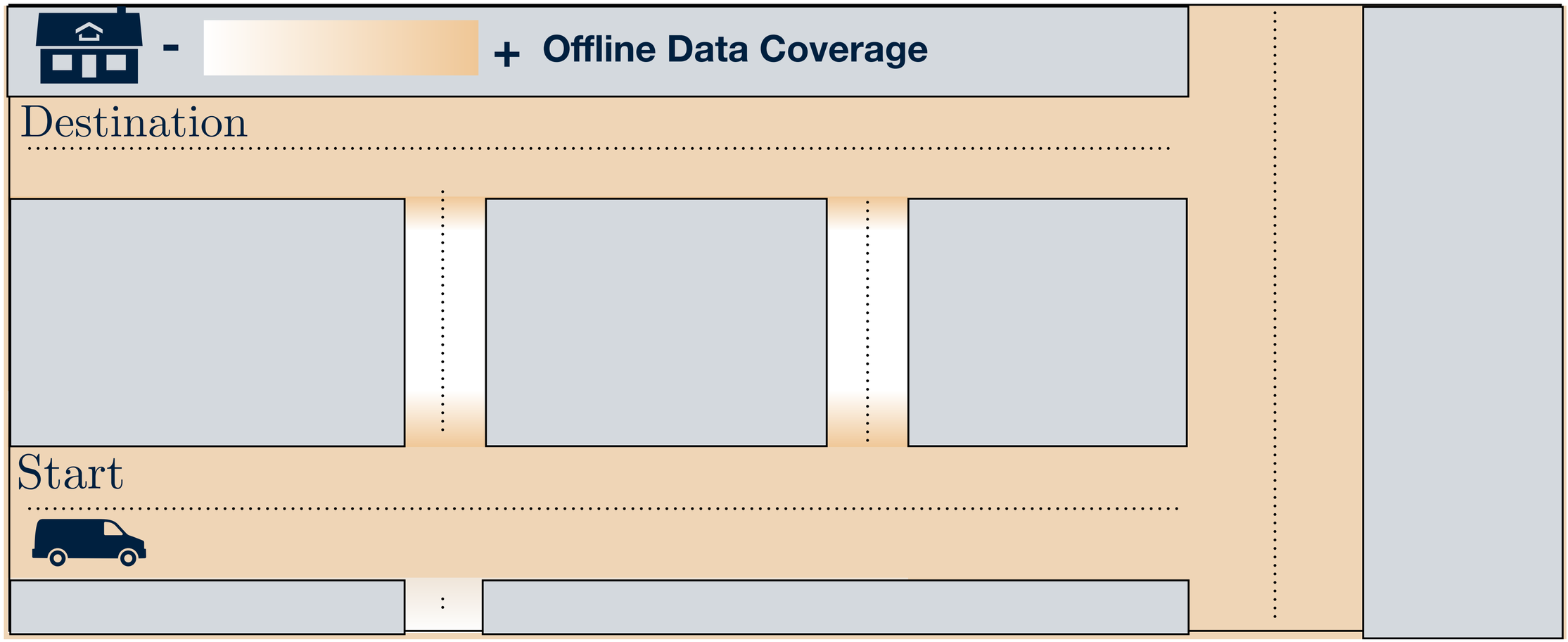
Why is adaptation necessary?

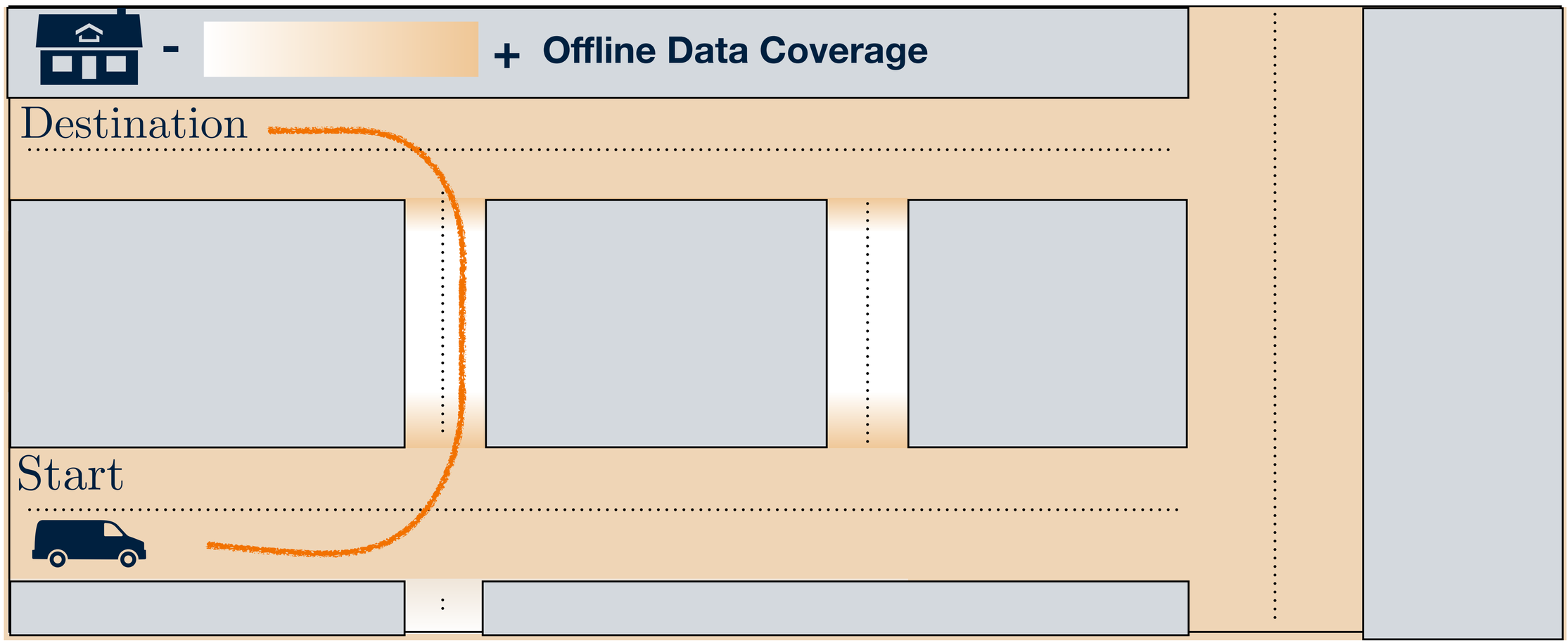
How should we train to adapt?

Why is adaptation necessary in Offline RL?


Destination
Start

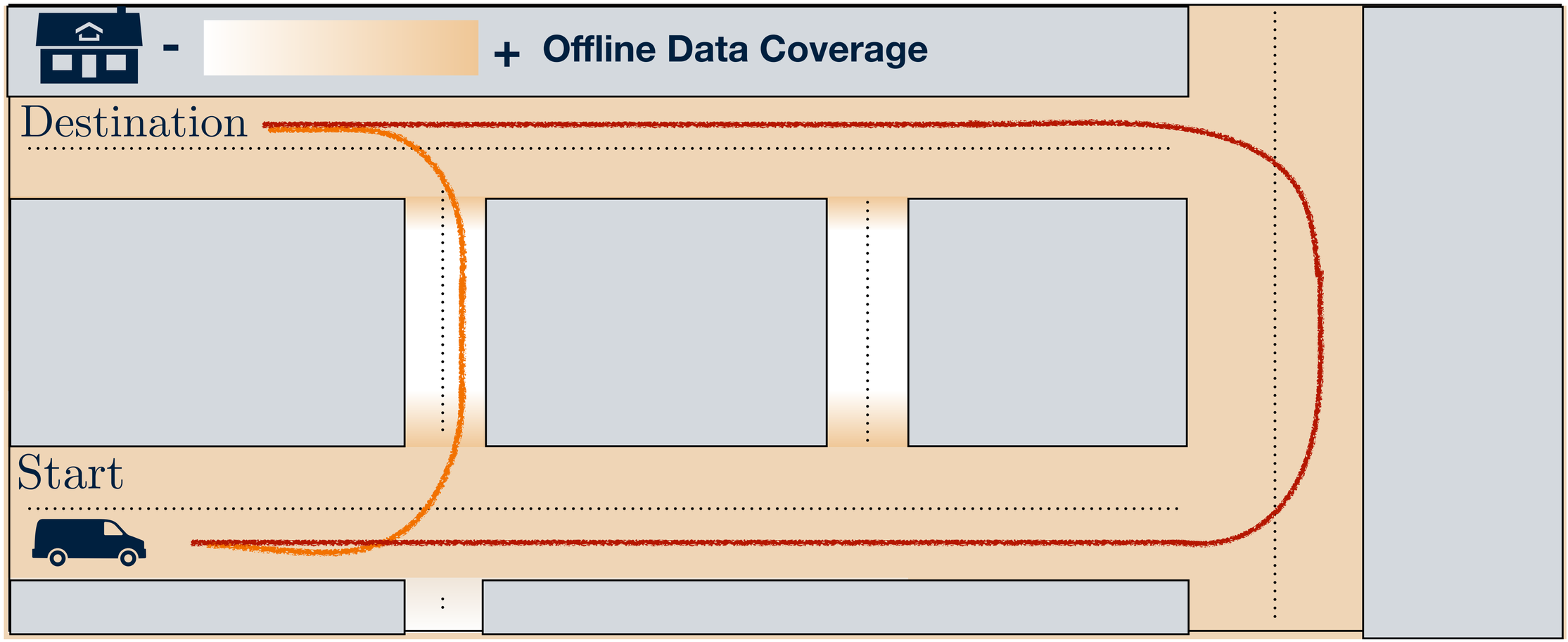

-
+ Offline Data Coverage
Destination
Start


-
+ Offline Data Coverage
Destination
Start



Non-conservative solution

Risky (but fast if succeeds)

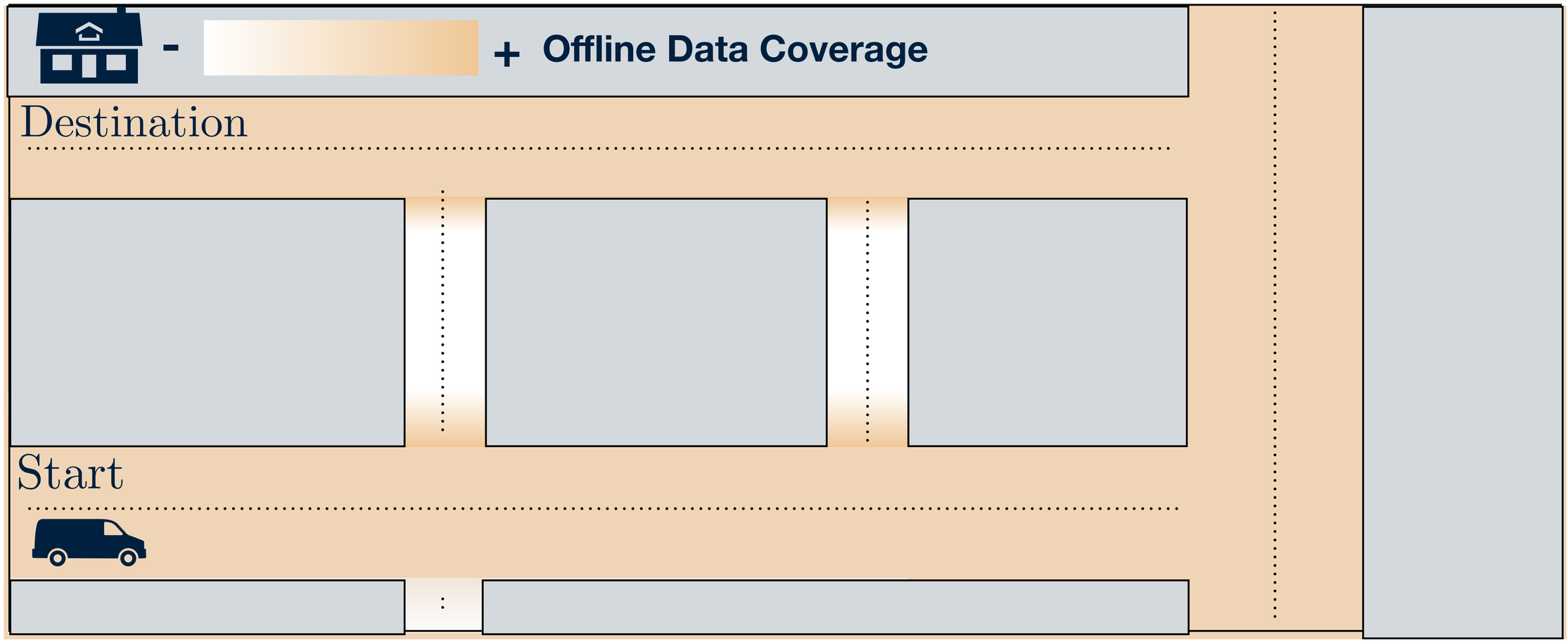


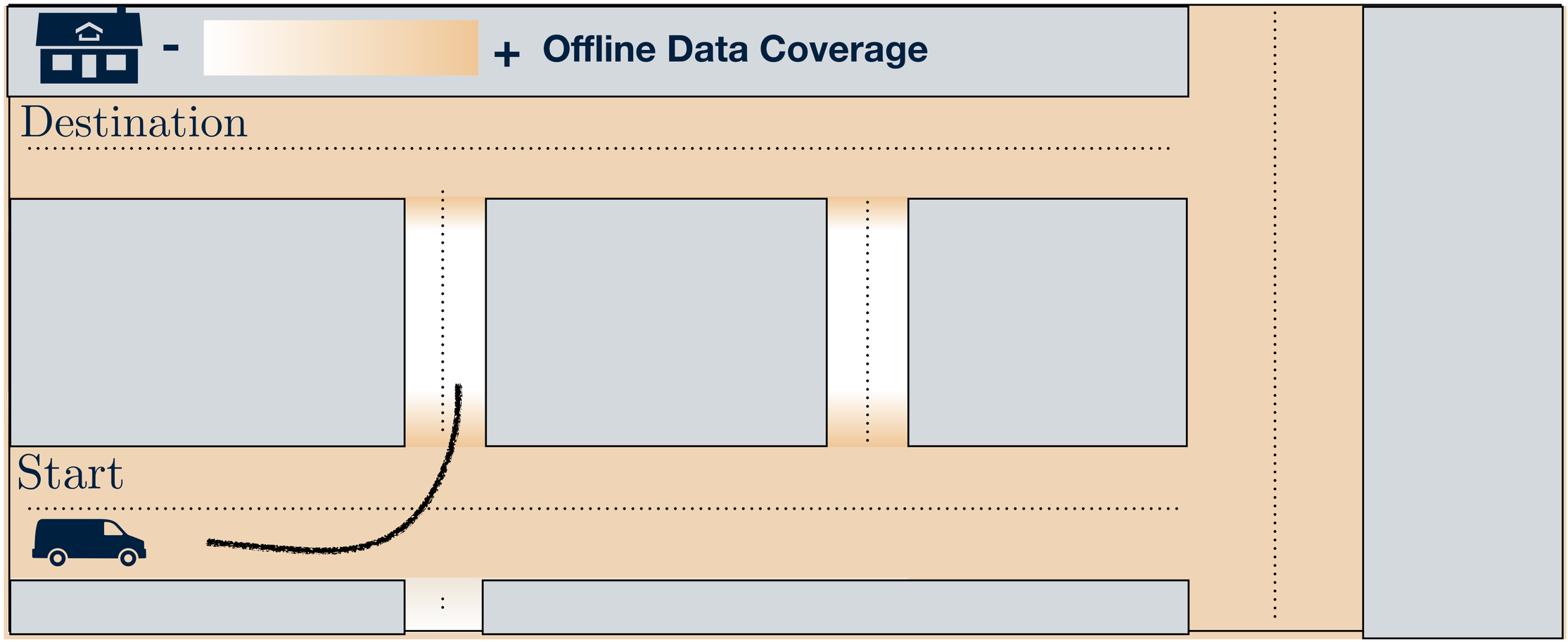
Non-conservative solution

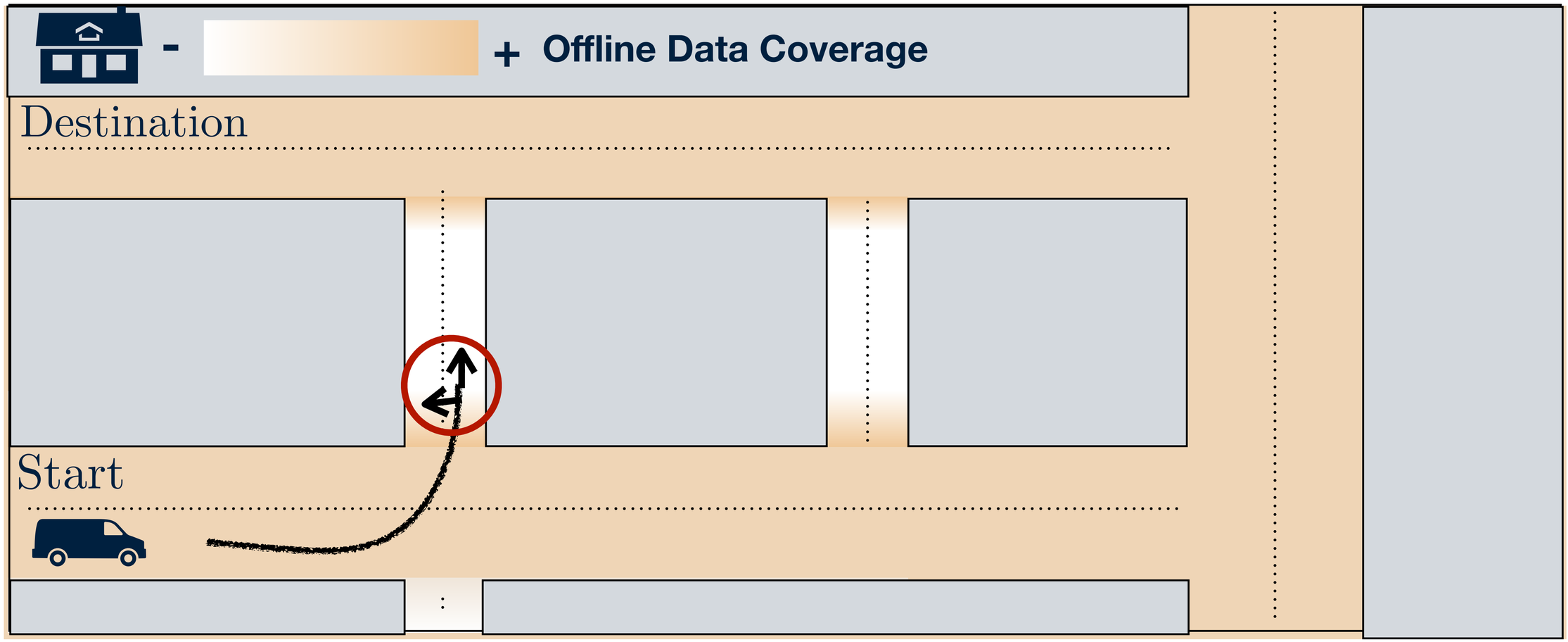
Risky (but fast if succeeds)

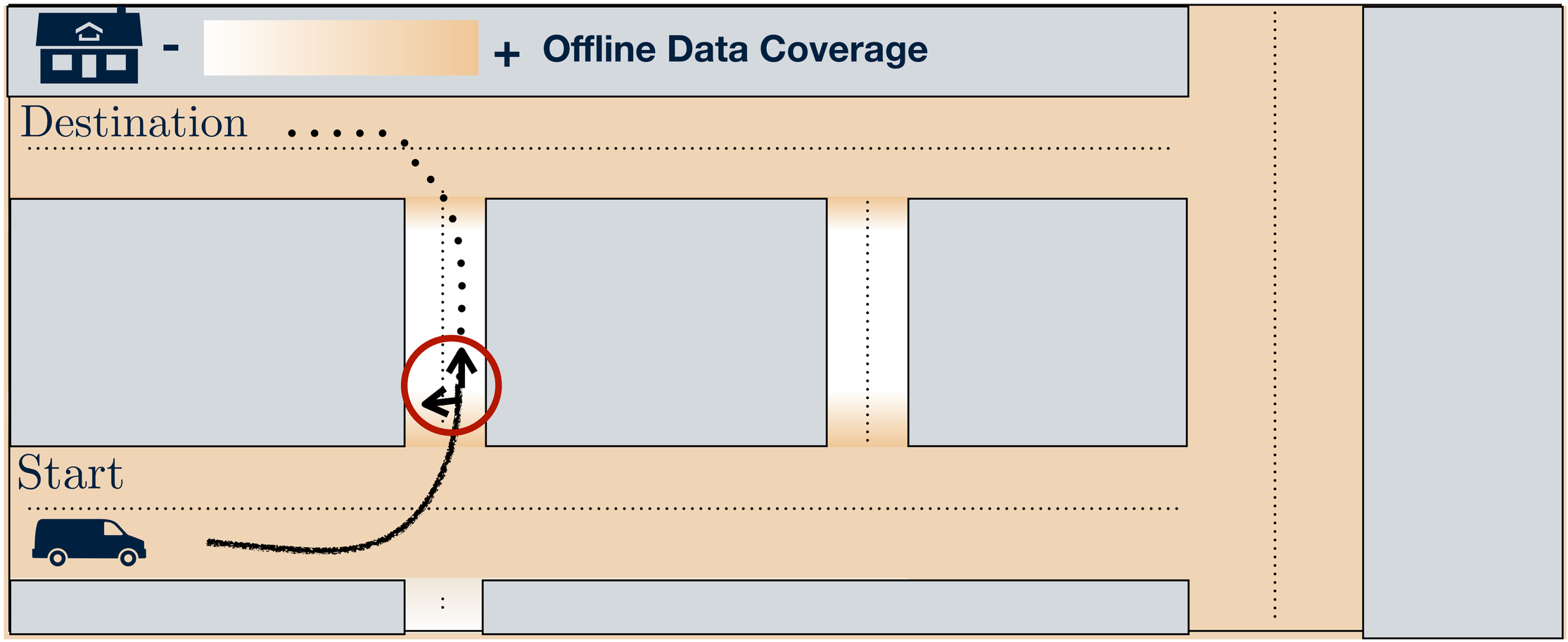
Conservative solution

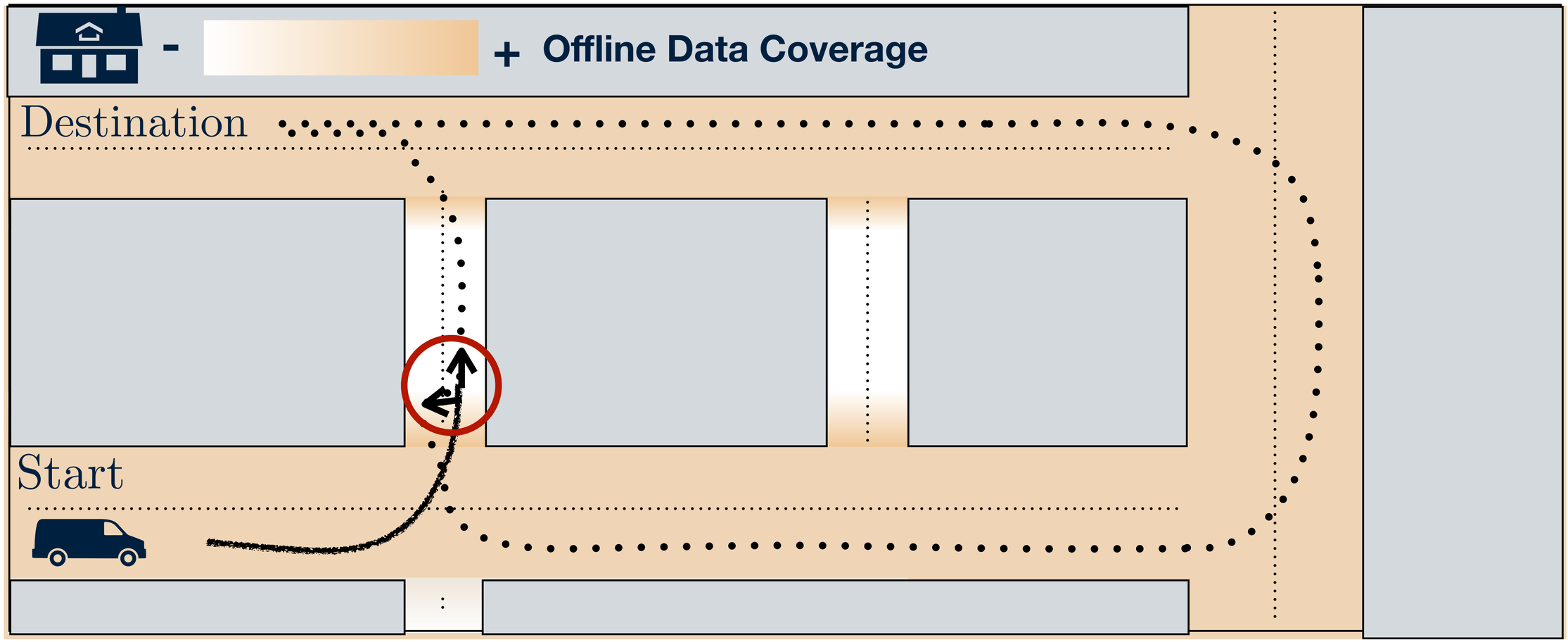
Stable but always slow

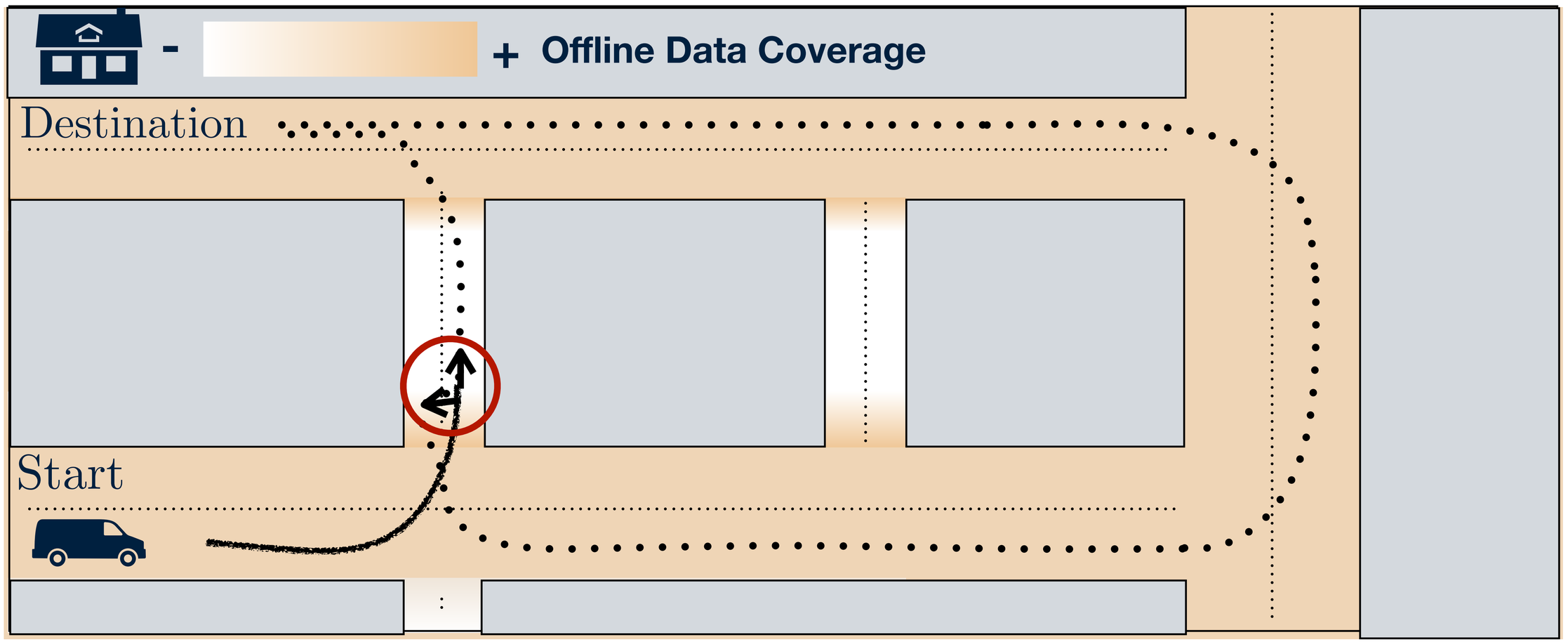

-
+ Offline Data Coverage
Destination
Start


-
+ Offline Data Coverage
Destination
Start


-
+ Offline Data Coverage
Destination
Start


-
+ Offline Data Coverage
Destination
Start


-
+ Offline Data Coverage
Destination
Start



During evaluation, agent *changes* its uncertainty about environment

The agent's epistemic uncertainty is not static

During evaluation, transitions provided by environment *changes the agent's uncertainty*

The agent's epistemic uncertainty is not static

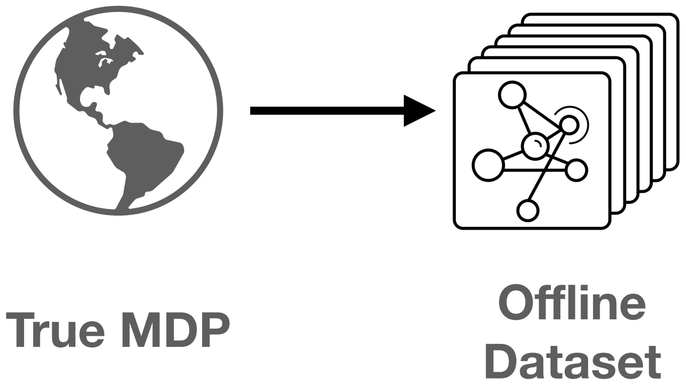
During evaluation, transitions provided by environment *changes the agent's uncertainty*

The policy can increase performance by changing

Offline RL in a Bayesian Perspective

True MDP

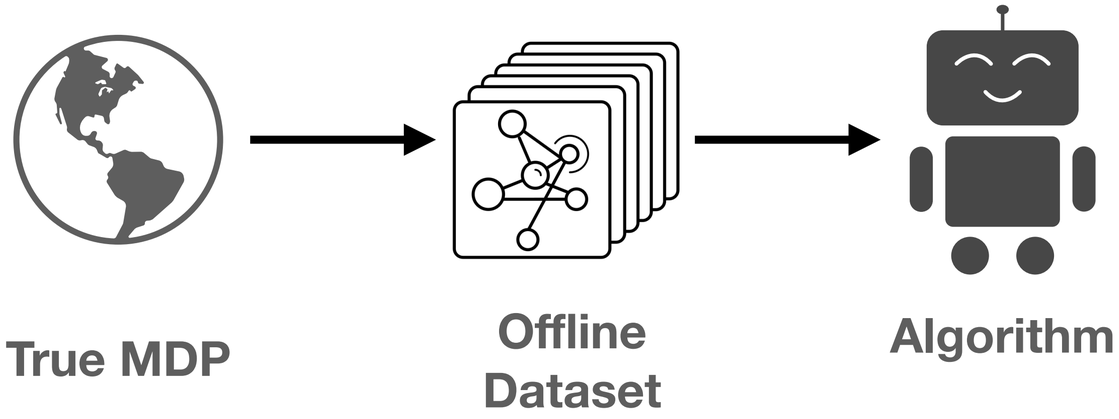
Offline RL in a Bayesian Perspective



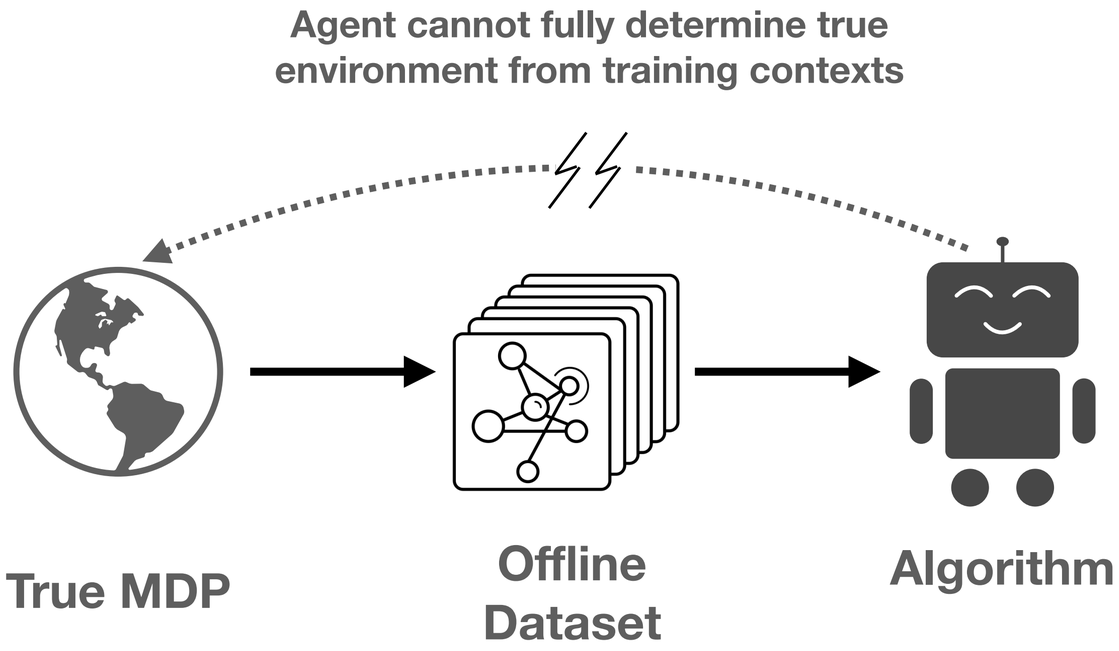
True MDP

Offline
Dataset

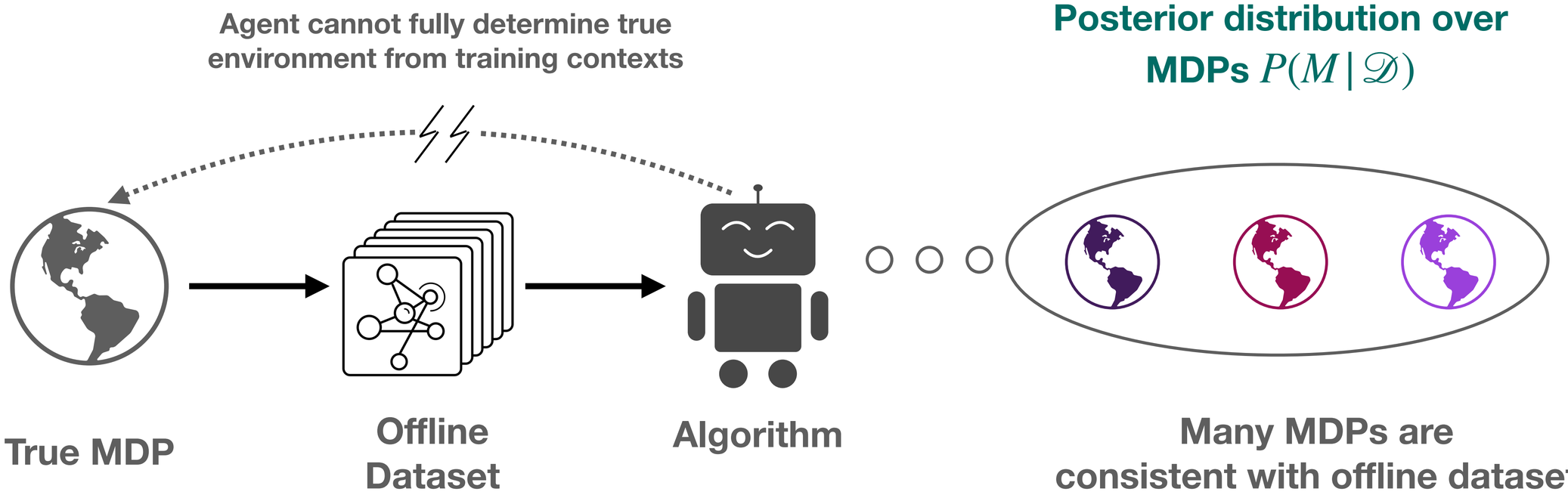
Offline RL in a Bayesian Perspective



Offline RL in a Bayesian Perspective



Offline RL in a Bayesian Perspective



To be (Bayes)-optimal in offline RL…

Maximize return on average from MDPs from the posterior distribution

To be (Bayes)-optimal in offline RL…

Maximize return on average from MDPs from the posterior distribution

$$\max_{\pi} J_{Bayes}(\pi) = \mathbb{E}_{M \sim P(M|D)}[J_M(\pi)]$$

Average over feasible MDPs given offline dataset

Return of policy in MDP

To be (Bayes)-optimal in offline RL…

Maximize return on average from MDPs from the posterior distribution

$$\max_{\pi} J_{Bayes}(\pi) = \mathbb{E}_{M \sim P(M|D)}[J_M(\pi)]$$

Average over feasible MDPs given offline dataset

Return of policy in MDP

This turns out to be a POMDP objective! [Duff et al, 2002]

Theorem (informal): The Bayes-optimal offline RL policy is memory-based.

Intuition: Test-time return objective is a POMDP, so optimal policy is adaptive

Proposition A.1 (Sub-optimality of Markovian policies and optimality of adaptiveness)**.** *Let* $n \in \mathbb{N}$*. There are offline RL problem instances* $(\mathcal{D}, p(\mathcal{M}))$ *with* n*-state MDPs where the adaptive Bayes-optimal policy achieves* $J_{Bayes}(\pi^*_{adaptive}) = -2n$ *but the highest performing Markovian policy achieves return of a magnitude worse:* $J_{Bayes}(\pi^*_{markov}) \leq -\frac{1}{2}n^2$.

How can we learn to adapt in Offline RL?

Approach

$$\max_{\pi} J_{Bayes}(\pi) = \mathbb{E}_{M \sim P(M|D)}[J_M(\pi)]$$

Average over likely MDPs given offline dataset

Return of policy in MDP

Follow the policy gradient of the Bayesian offline RL objective

The Important Components

- **The policy needs to be adaptive to changes in uncertainty**
- **Value functions must understand how uncertainty can change**
- **The policy should learn to focus on value functions consistent with the current trajectory**

Choosing the right policy class

State-based policies $\pi_\theta(a \mid s)$ are suboptimal in offline RL because they don't understand how agent's uncertainty has changed during an episode.

Definition: The **relative MDP weighting** $\mathbf{b}(h)$ measures which MDP in the posterior distribution is most likely to have produced the history h

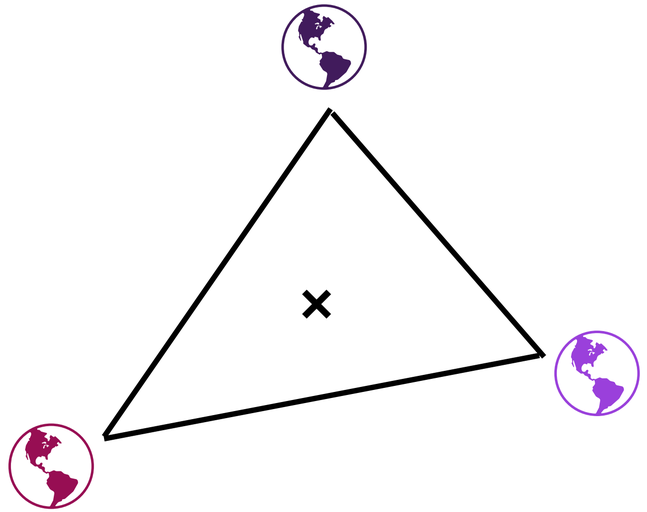
$$\mathbf{b}(h)(M) \propto \exp(-\sum_{i=1}^{T} \textbf{Surprise}(M, (s_t, a_t, r_t, s_{t+1}))$$

State-based Policies ➡ Uncertainty-Adaptive Policies

Definition: The **relative MDP weighting** $\mathbf{b}(h)$ measures which MDP in the posterior distribution is most likely to have produced the history h

$$\mathbf{b}(h)(M) \propto \exp(-\sum_{i=1}^{T} \textbf{Surprise}(M, (s_t, a_t, r_t, s_{t+1}))$$

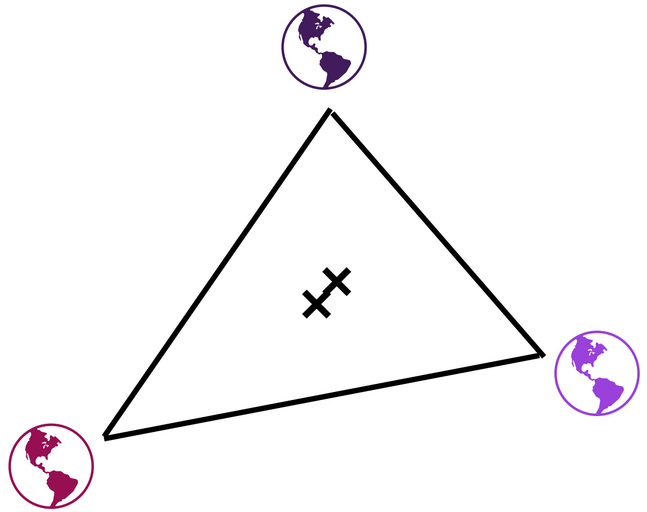
State-based Policies ➡ Uncertainty-Adaptive Policies



Definition: The **relative MDP weighting** $\mathbf{b}(h)$ measures which MDP in the posterior distribution is most likely to have produced the history h

$$\mathbf{b}(h)(M) \propto \exp(-\sum_{i=1}^{T} \textbf{Surprise}(M, (s_t, a_t, r_t, s_{t+1}))$$

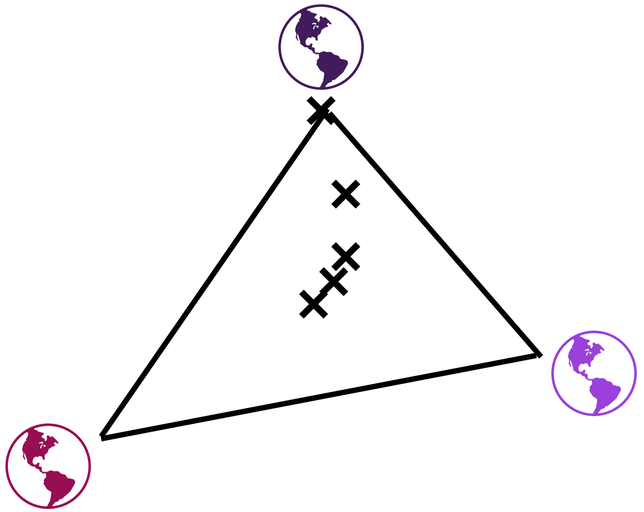
State-based Policies ➡ Uncertainty-Adaptive Policies



Definition: The **relative MDP weighting** $\mathbf{b}(h)$ measures which MDP in the posterior distribution is most likely to have produced the history h

$$\mathbf{b}(h)(M) \propto \exp(-\sum_{i=1}^{T} \textbf{Surprise}(M, (s_t, a_t, r_t, s_{t+1}))$$

State-based Policies ➡ Uncertainty-Adaptive Policies

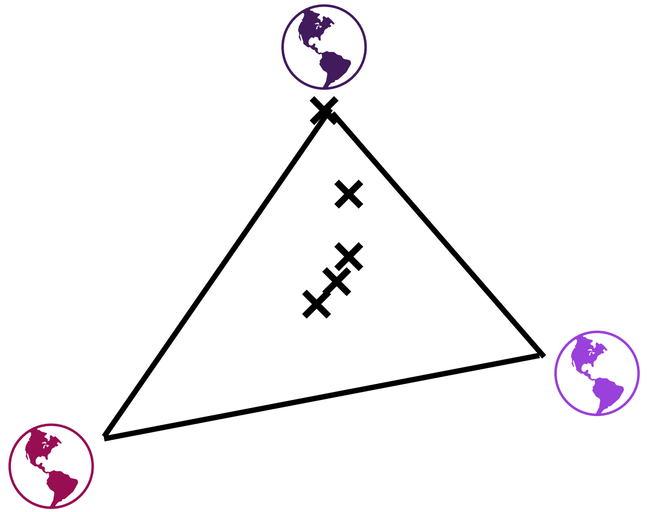


Definition: The **relative MDP weighting** $\mathbf{b}(h)$ measures which MDP in the posterior distribution is most likely to have produced the history h

$$\mathbf{b}(h)(M) \propto \exp(-\sum_{i=1}^{T} \textbf{Surprise}(M, (s_t, a_t, r_t, s_{t+1}))$$

State-based Policies ➡ Uncertainty-Adaptive Policies

$$\pi(a \mid s)$$

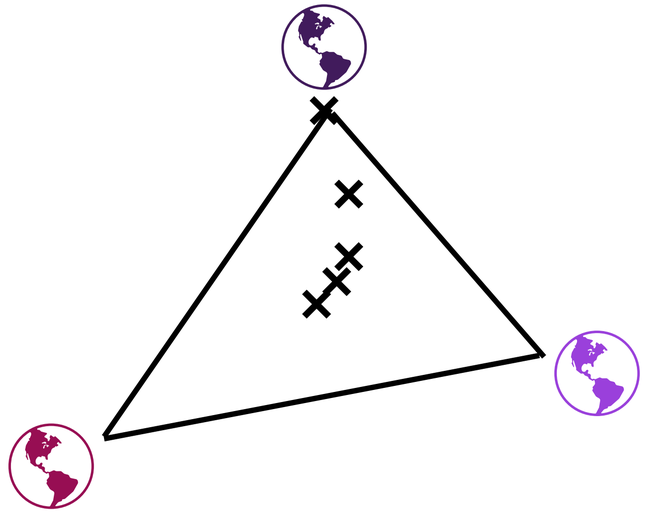


Definition: The **relative MDP weighting** $\mathbf{b}(h)$ measures which MDP in the posterior distribution is most likely to have produced the history h

$$\mathbf{b}(h)(M) \propto \exp(-\sum_{i=1}^{T} \textbf{Surprise}(M, (s_t, a_t, r_t, s_{t+1}))$$

State-based Policies ➡ Uncertainty-Adaptive Policies

$$\pi(a \mid s, \mathbf{b}(h))$$



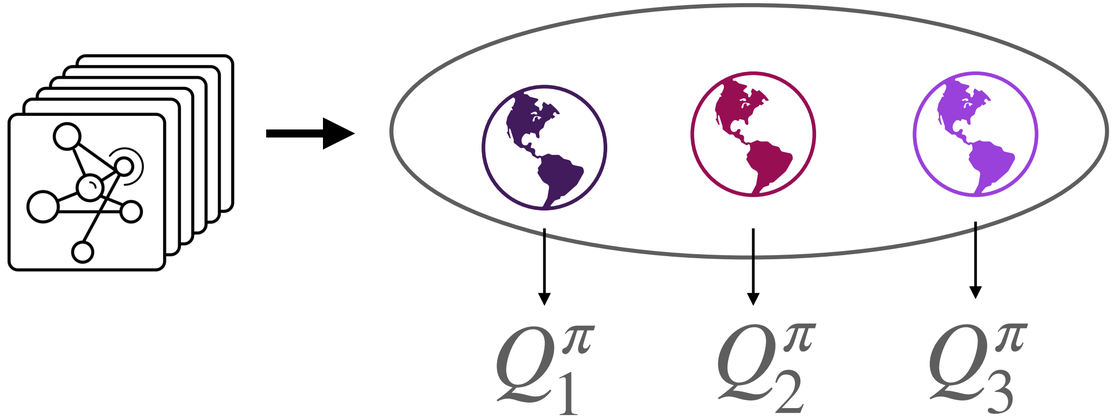
Definition: The **relative MDP weighting** $\mathbf{b}(h)$ measures which MDP in the posterior distribution is most likely to have produced the history h

$$\mathbf{b}(h)(M) \propto \exp(-\sum_{i=1}^{T} \textbf{Surprise}(M, (s_t, a_t, r_t, s_{t+1}))$$

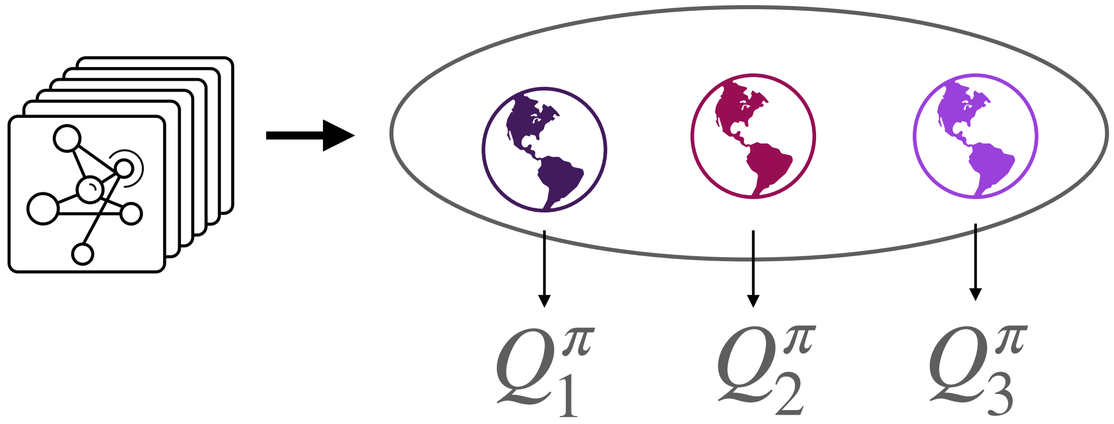
An Abridged Algorithm

- The policy needs to be adaptive to changes in uncertainty
- **Value functions must understand how uncertainty can change**
- The policy should learn to focus on value functions consistent with the current trajectory

Learning value functions



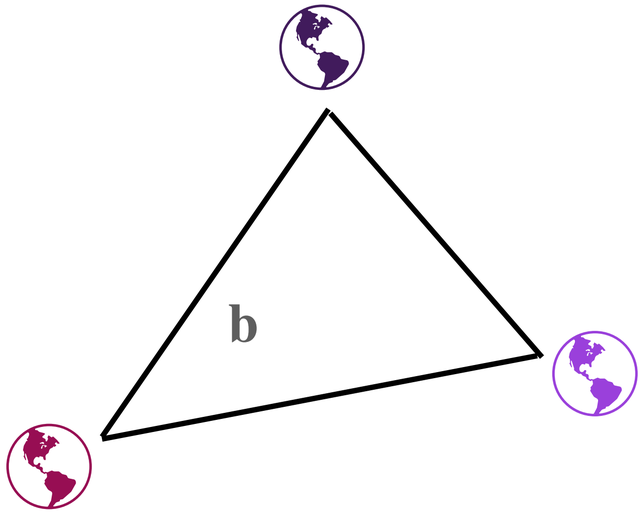
Learning value functions



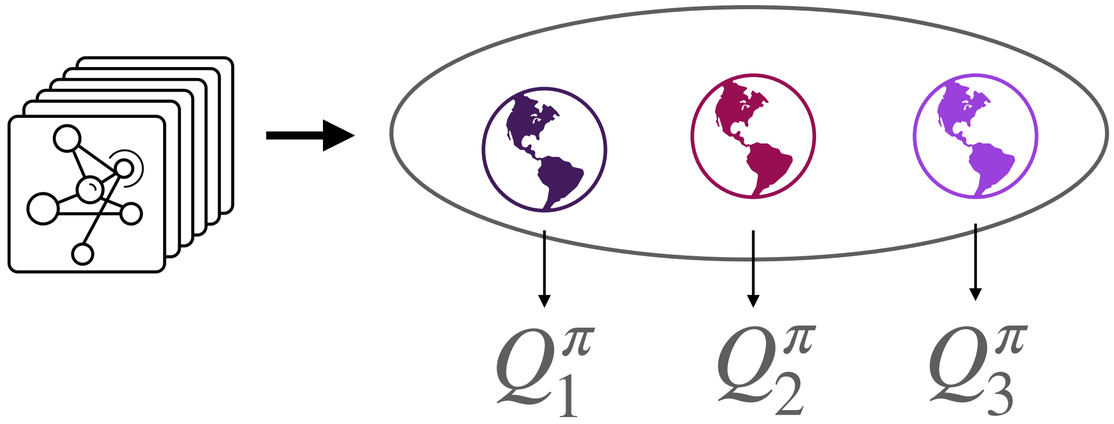
$$Q_1^\pi(s, a, \mathbf{b}) =$$

$$r(s, a) + \gamma \mathbb{E}_{s' \sim \text{🌎}} [\mathbb{E}_{a' \sim \pi}[Q_1^\pi(s', a', \mathbf{b}')]$$

where $\mathbf{b}'$ is the new MDP weighting after witnessing (s, a, r, s')



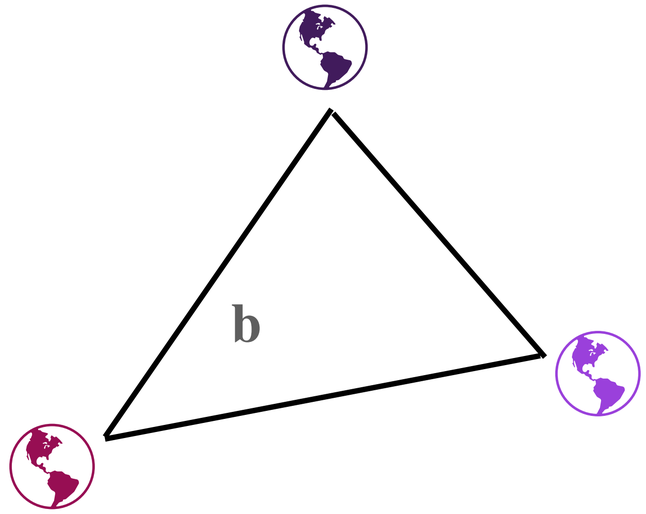
Learning value functions



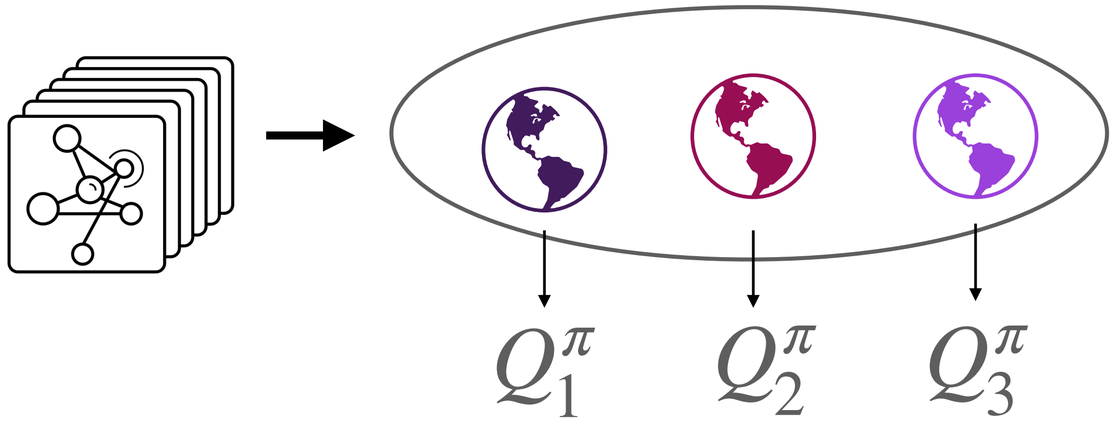
$$Q_1^\pi(s, a, \mathbf{b}) =$$

$$r(s, a) + \gamma \mathbb{E}_{s' \sim} [\mathbb{E}_{a' \sim \pi}[Q_1^\pi(s', a', \mathbf{b}')]$$

where $\mathbf{b}'$ is the new MDP weighting after witnessing (s, a, r, s')



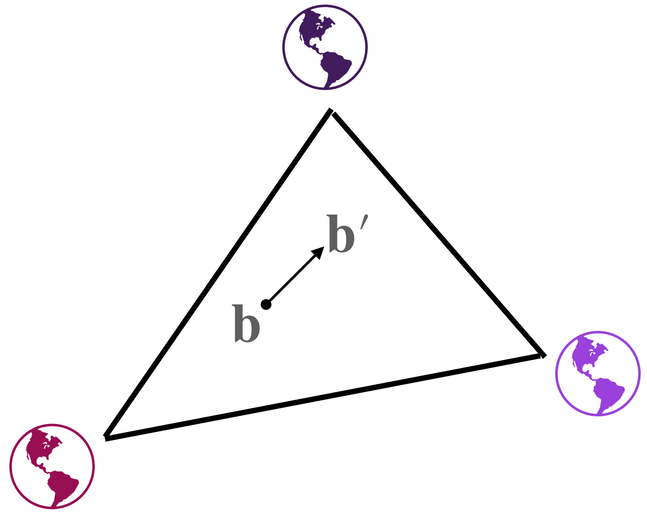
Learning value functions



$$Q_1^\pi(s, a, \mathbf{b}) =$$

$$r(s, a) + \gamma \mathbb{E}_{s' \sim \text{🌎}} [\mathbb{E}_{a' \sim \pi}[Q_1^\pi(s', a', \mathbf{b}')]$$

where $\mathbf{b}'$ is the new MDP weighting after witnessing (s, a, r, s')



The Important Components

- The policy needs to be adaptive to changes in uncertainty
- Value functions must understand how uncertainty can change
- **The policy should learn to focus on value functions consistent with the current trajectory**

$$\max_{\pi(a|s,\mathbf{b})} \mathbb{E}_{a\sim\pi(\cdot|s,\mathbf{b})}[\mathbb{E}_{M\sim P(M|\mathcal{D})}[\mathbf{b}(M)Q_M^\pi(s,a,\mathbf{b})]]$$

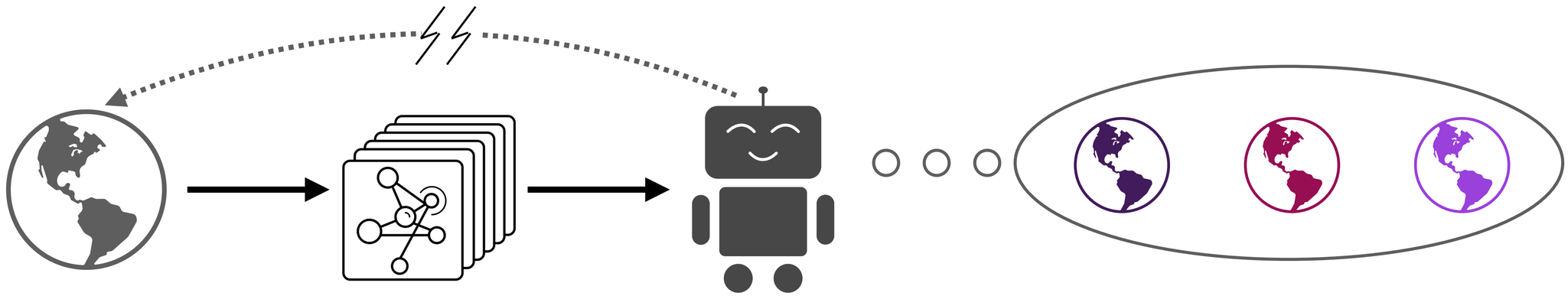
$$\max_{\pi(a|s,\mathbf{b})} \mathbb{E}_{a\sim\pi(\cdot|s,\mathbf{b})}[\, Q^{\pi}_{M}(s, a, \mathbf{b})]$$

For a single MDP M

$$\max_{\pi(a|s,\mathbf{b})} \mathbb{E}_{a\sim\pi(\cdot|s,\mathbf{b})}[\, Q^{\pi}_{M}(s, a, \mathbf{b})]$$

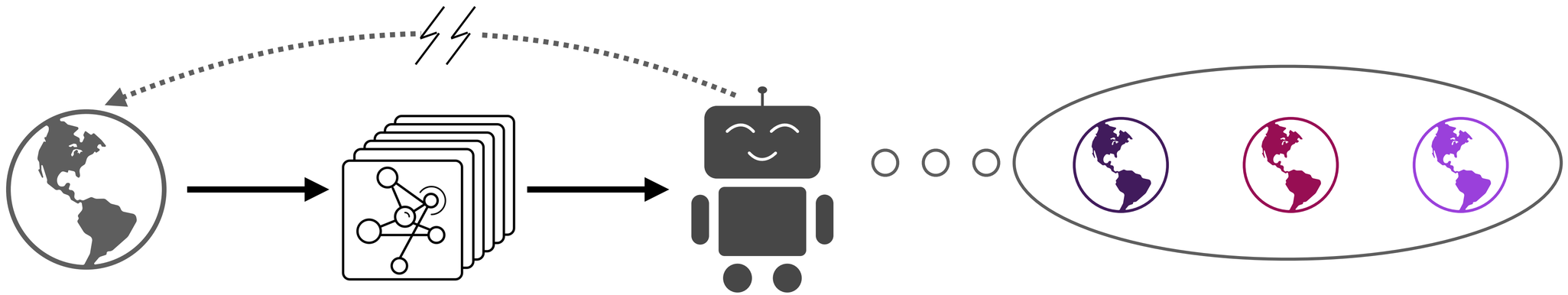
For a single MDP M

The dataset induces a distribution over MDPs $P(M \mid \mathcal{D})$



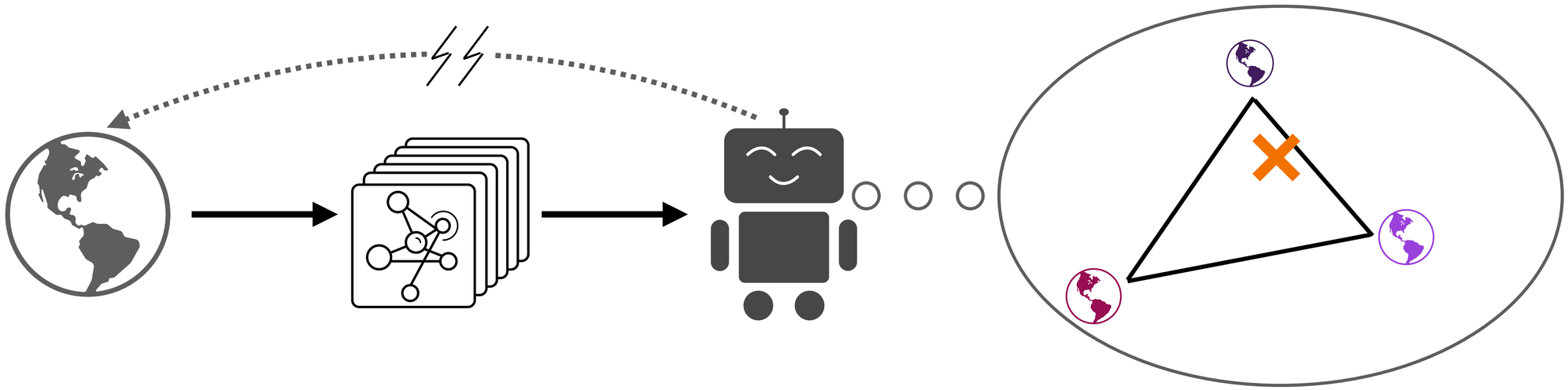
$$\max_{\pi(a|s,\mathbf{b})} \mathbb{E}_{a\sim\pi(\cdot|s,\mathbf{b})}[\mathbb{E}_{M\sim P(M|\mathcal{D})}[\, Q^{\pi}_{M}(s, a, \mathbf{b})]]$$

The dataset induces a distribution over MDPs $P(M \mid \mathcal{D})$



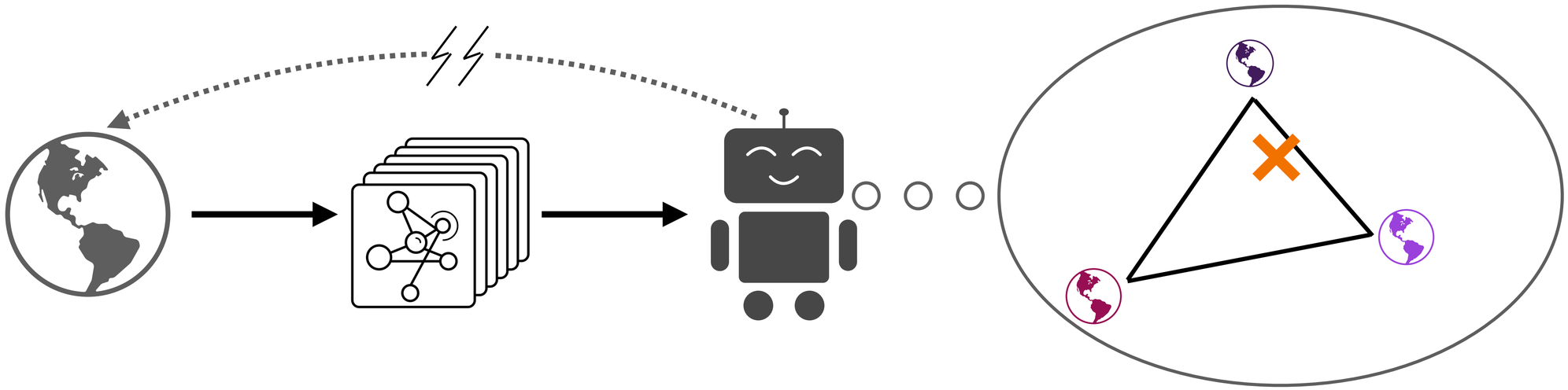
$$\max_{\pi(a|s,\mathbf{b})} \mathbb{E}_{a\sim\pi(\cdot|s,\mathbf{b})}[\, \mathbb{E}_{M\sim P(M|\mathcal{D})}[\, Q_M^{\pi}(s,a,\mathbf{b})]]$$

Furthermore, this distribution has changed within the episode (relative MDP weighting)

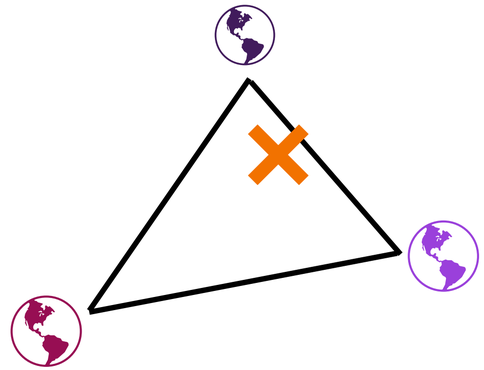


$$\max_{\pi(a|s,\mathbf{b})} \mathbb{E}_{a\sim\pi(\cdot|s,\mathbf{b})}[\ \mathbb{E}_{M\sim P(M|\mathcal{D})}[\mathbf{b}(M)Q_M^\pi(s,a,\mathbf{b})]]$$

Furthermore, this distribution has changed within the episode (relative MDP weighting)



$$\max_{\pi(a|s,\mathbf{b})} \mathbb{E}_{a\sim\pi(\cdot|s,\mathbf{b})}[\mathbb{E}_{M\sim P(M|\mathcal{D})}[\mathbf{b}(M)Q^{\pi}_{M}(s,a,\mathbf{b})]]$$



Interpretation: Take actions with high value averaged across MDPs in the posterior that are consistent with the trajectory seen so far

The Important Components

- The policy needs to be adaptive to changes in uncertainty
- Value functions must understand how uncertainty can change
- The policy should learn to focus on value functions consistent with the current trajectory

APE-V (Adaptive Policies with Ensembles of Value Functions)

APE-V (Adaptive Policies with Ensembles of Value Functions)

Ensemble of Value Functions $\{Q_1^\pi, Q_2^\pi, \dots Q_n^\pi\}$

Trained to represent posterior over value functions $P(Q_M^\pi \mid \mathcal{D})$

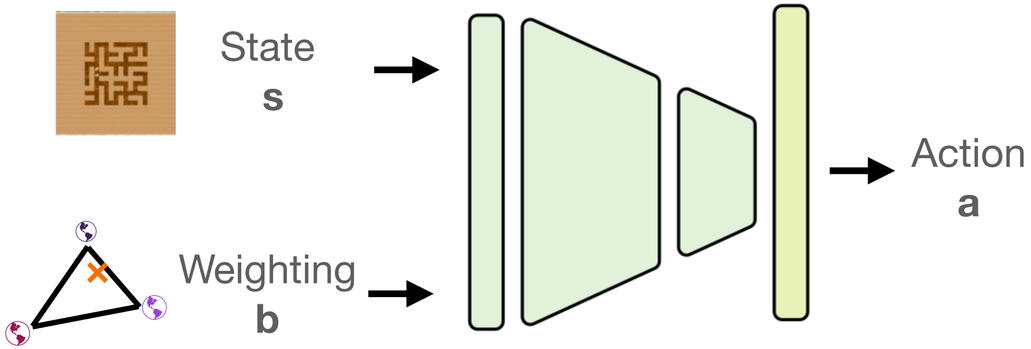
APE-V (Adaptive Policies with Ensembles of Value Functions)

Ensemble of Value Functions

$$\{Q_1^\pi, Q_2^\pi, \dots Q_n^\pi\}$$

Trained to represent posterior over value functions $P(Q_M^\pi \mid \mathcal{D})$

Uncertainty-Adaptive Policy



Trained to maximize weighted average of value functions

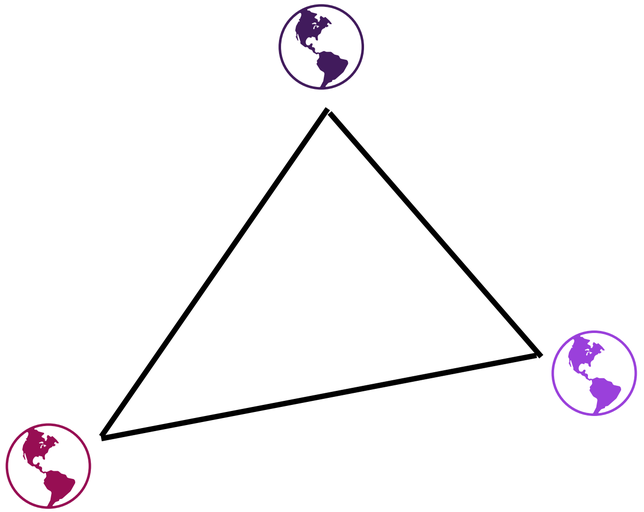
$$\max \mathbb{E}_{a\sim\pi}[\sum_k \mathbf{b}_k Q_k]$$

APE-V (Adaptive Policies with Ensembles of Value Functions)

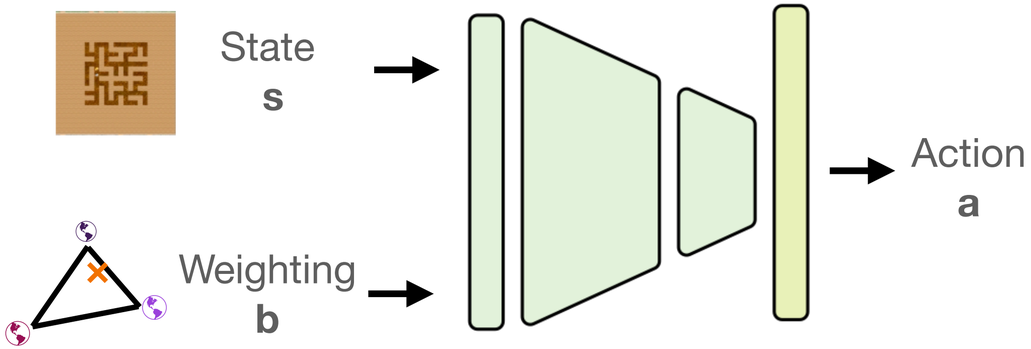
Ensemble of Value Functions

$$\{Q_1^\pi, Q_2^\pi, \dots Q_n^\pi\}$$

Trained to represent posterior over value functions $P(Q_M^\pi \mid \mathcal{D})$



Uncertainty-Adaptive Policy



Trained to maximize weighted average of value functions

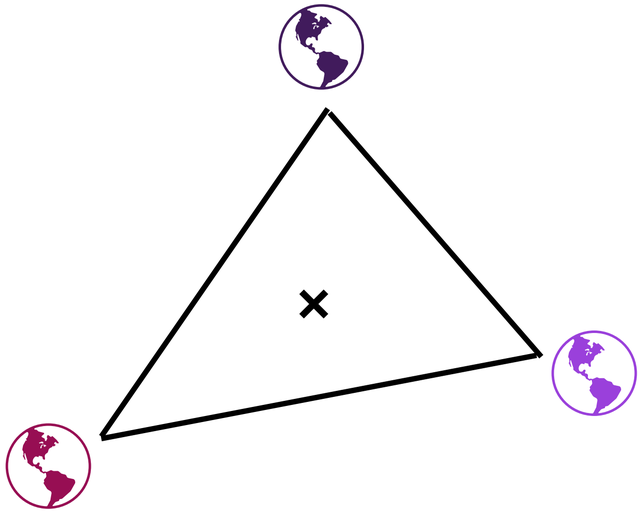
$$\max \mathbb{E}_{a\sim\pi}[\sum_k \mathbf{b}_k Q_k]$$

APE-V (Adaptive Policies with Ensembles of Value Functions)

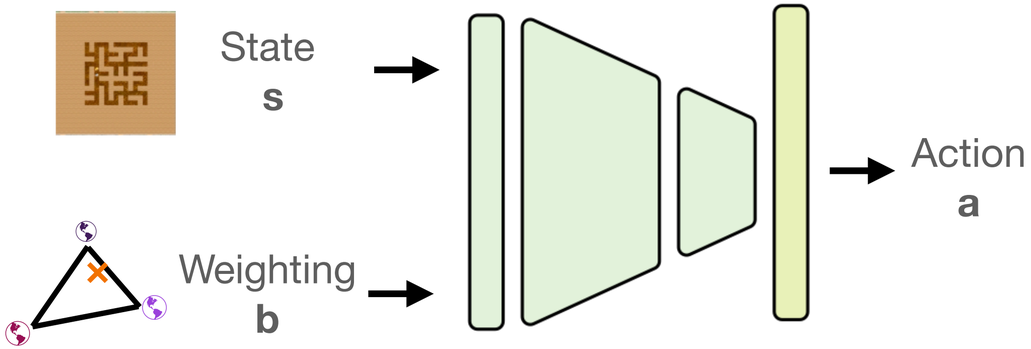
Ensemble of Value Functions

$$\{Q_1^\pi, Q_2^\pi, \dots Q_n^\pi\}$$

Trained to represent posterior over value functions $P(Q_M^\pi \mid \mathcal{D})$



Uncertainty-Adaptive Policy



Trained to maximize weighted average of value functions

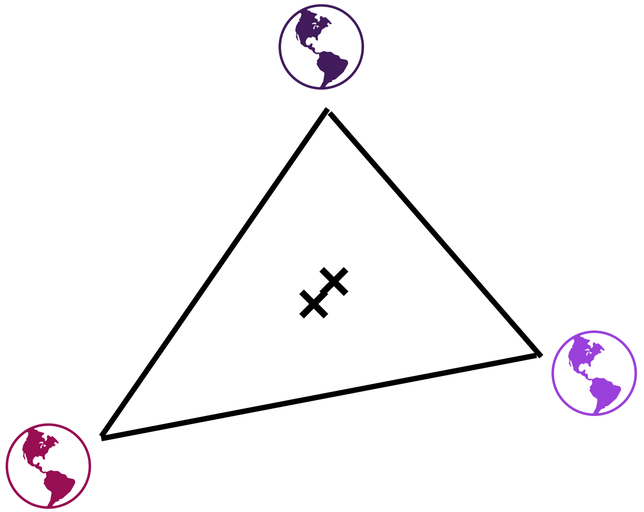
$$\max \mathbb{E}_{a\sim\pi}[\sum_k \mathbf{b}_k Q_k]$$

APE-V (Adaptive Policies with Ensembles of Value Functions)

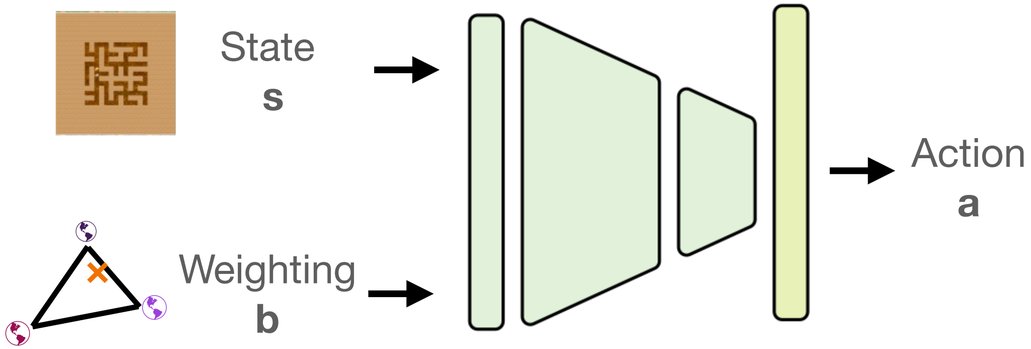
Ensemble of Value Functions

$$\{Q_1^\pi, Q_2^\pi, \dots Q_n^\pi\}$$

Trained to represent posterior over value functions $P(Q_M^\pi \mid \mathcal{D})$



Uncertainty-Adaptive Policy



Trained to maximize weighted average of value functions

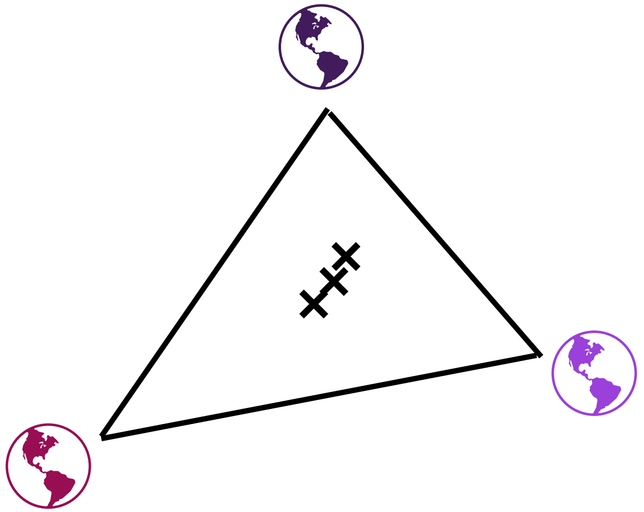
$$\max \mathbb{E}_{a\sim\pi}[\sum_k \mathbf{b}_k Q_k]$$

APE-V (Adaptive Policies with Ensembles of Value Functions)

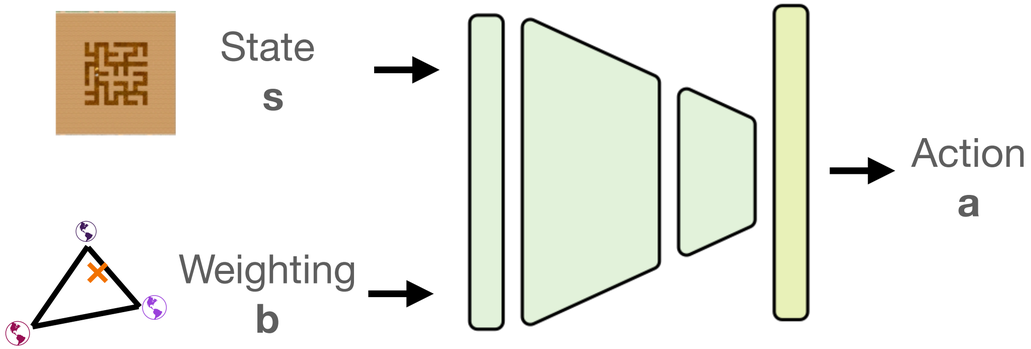
Ensemble of Value Functions

$$\{Q_1^\pi, Q_2^\pi, \dots Q_n^\pi\}$$

Trained to represent posterior over value functions $P(Q_M^\pi \mid \mathcal{D})$



Uncertainty-Adaptive Policy



Trained to maximize weighted average of value functions

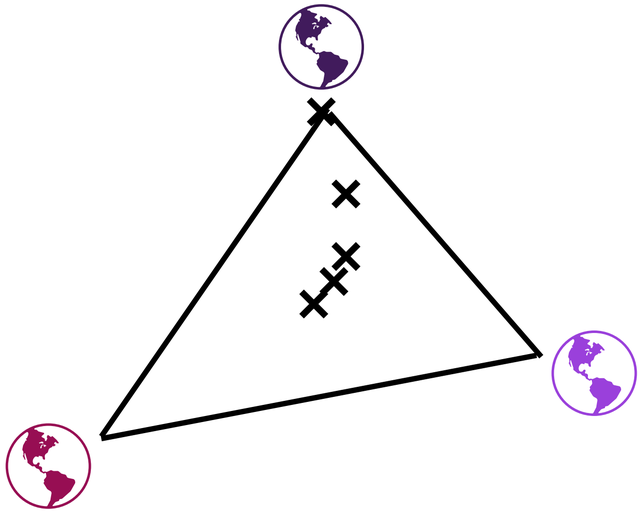
$$\max \mathbb{E}_{a\sim\pi}[\sum_k \mathbf{b}_k Q_k]$$

APE-V (Adaptive Policies with Ensembles of Value Functions)

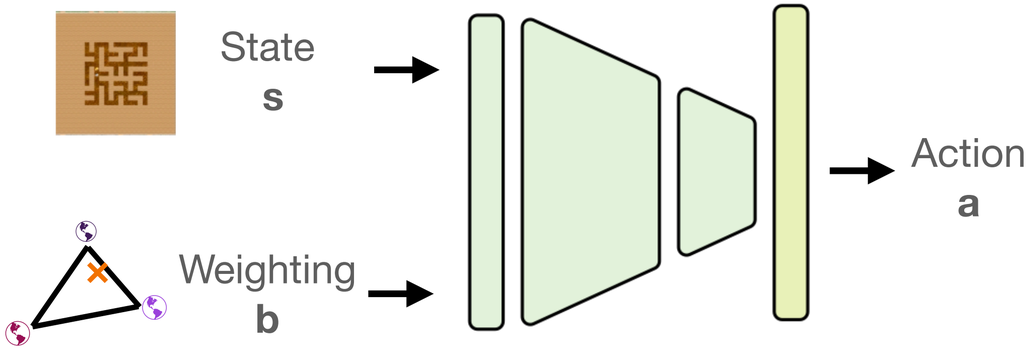
Ensemble of Value Functions

$$\{Q_1^\pi, Q_2^\pi, \dots Q_n^\pi\}$$

Trained to represent posterior over value functions $P(Q_M^\pi \mid \mathcal{D})$



Uncertainty-Adaptive Policy



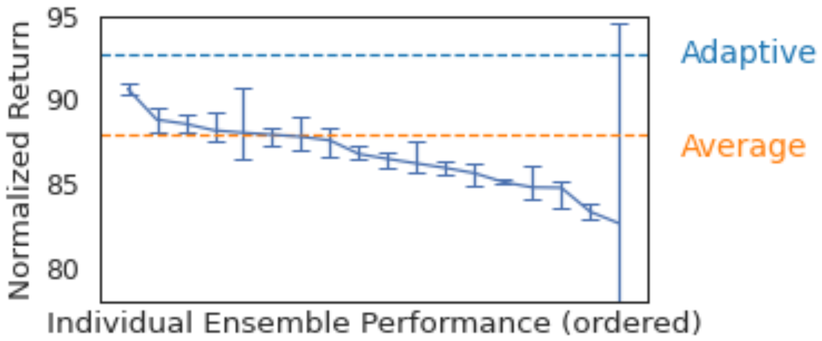
Trained to maximize weighted average of value functions

$$\max \mathbb{E}_{a\sim\pi}[\sum_k \mathbf{b}_k Q_k]$$

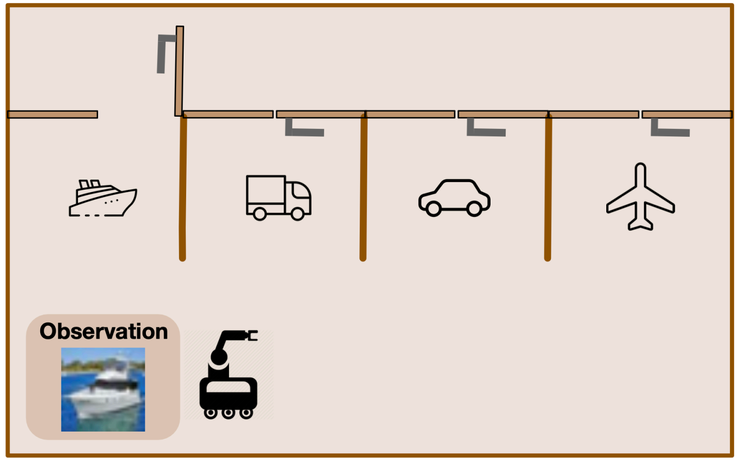
Experiments

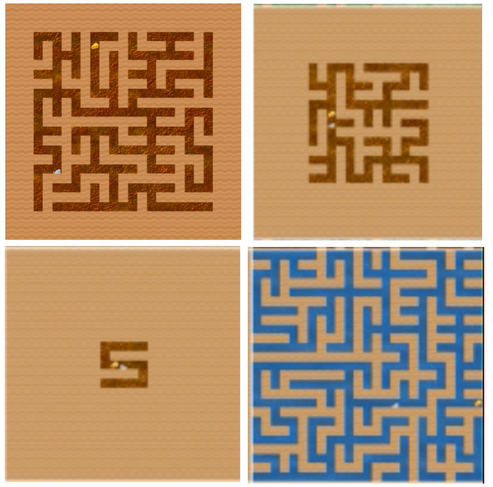
D4RL Offline RL Benchmark

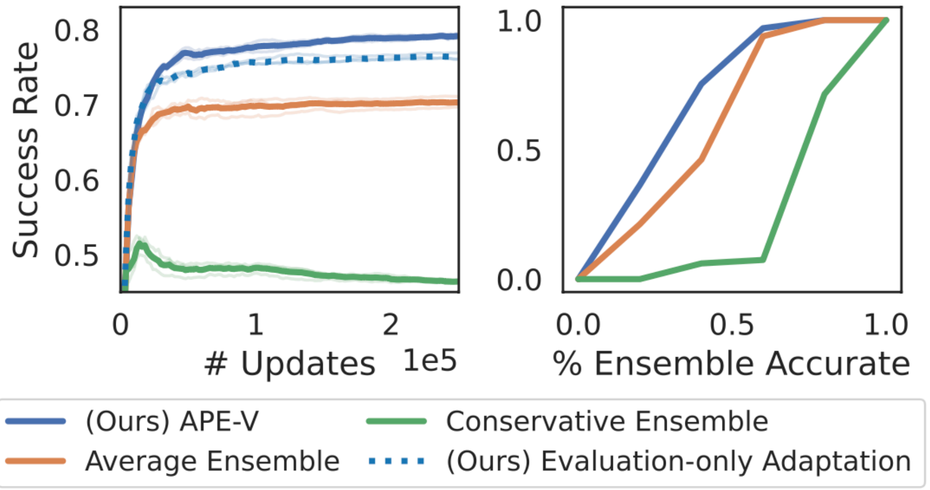
Task Name	CQL (Kumar et al., 2020)	IQL (Kostrikov et al., 2021b)	SAC-N (An et al., 2021)	APE-V
halfcheetah-random	**35.4**	31.3±3.5	29.8±1.6	29.9±1.1
halfcheetah-medium	44.4	47.4±0.2	67.5±1.2	**69.1 ± 0.4**
halfcheetah-medium-expert	62.4	95.0±1.4	**102.7±1.5**	**101.4 ± 1.4**
halfcheetah-medium-replay	46.2	44.2±1.2	**63.9±0.8**	**64.6 ± 0.9**
hopper-random	10.8	5.3±0.6	**31.3±0.0**	**31.3±0.2x**
hopper-medium-expert	**111.0**	96.9±15.1	110.1±0.3	105.72 ± 3.7
hopper-medium-replay	48.6	94.7±8.6	**101.8±0.5**	98.5 ± 0.5
walker2d-random	7.0	5.4±1.7	**16.3±9.4**	**15.5±8.5**
walker2d-medium	74.5	78.3±8.7	87.9±0.2	**90.3 ± 1.6**
walker2d-medium-expert	98.7	109.1±0.2	**116.0±6.3**	110.0 ± 1.5
walker2d-medium-replay	32.6	73.8±7.1	78.7±0.7	**82.9 ± 0.4**

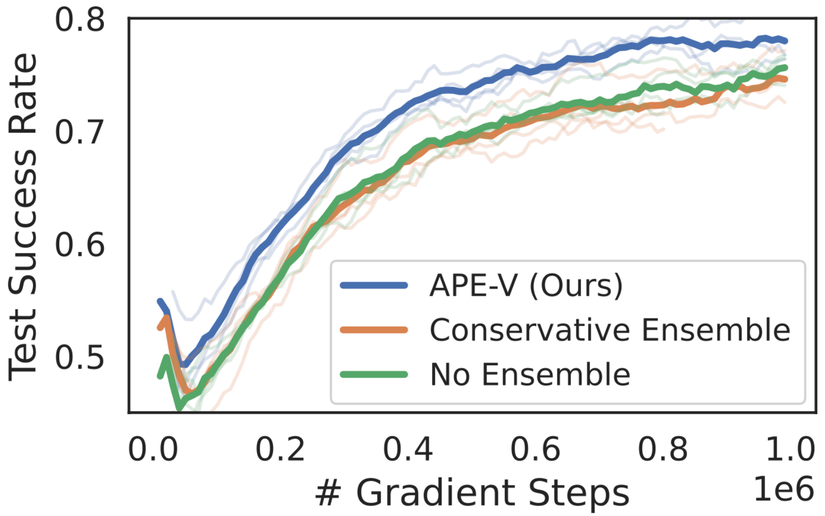


Adaptation Excels in Diverse Environments









Summary

Offline RL policies need the ability to adapt, and to be taught how to adapt