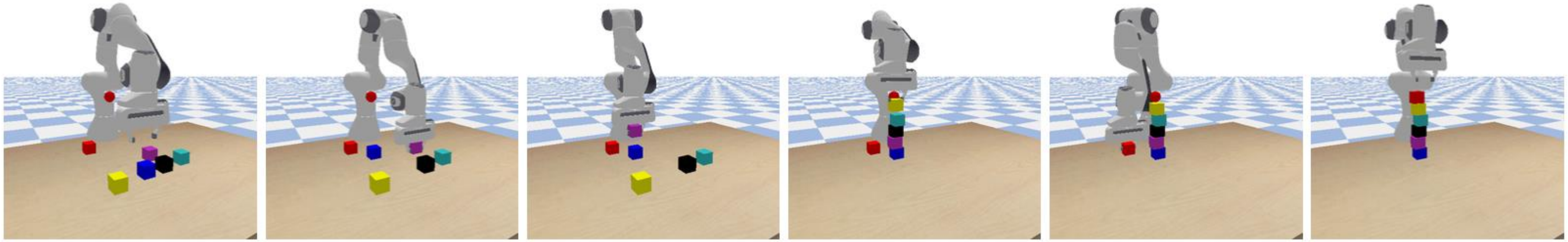
Phasic Self-Imitative Reduction for Sparse-Reward Goal-Conditioned Reinforcement Learning

Yunfei Li[1*], Tian Gao[1*], Jiaqi Yang[2], Huazhe Xu[3], Yi Wu[1,4]

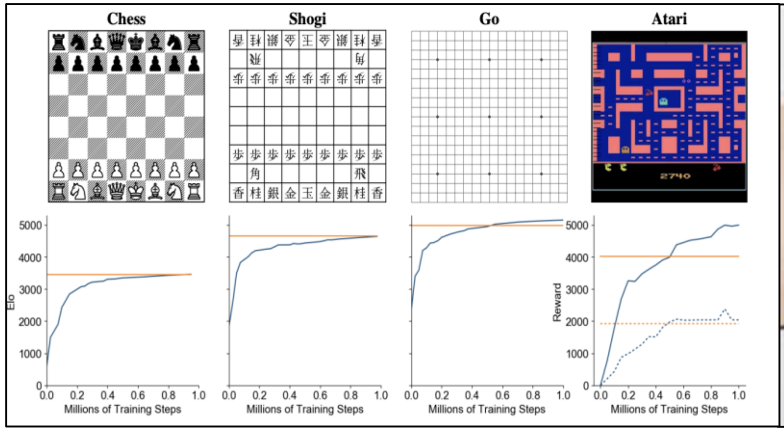
[1]IIIS Tsinghua University [2]UC Berkeley [3]Stanford

[4]Shanghai Qi Zhi Institute

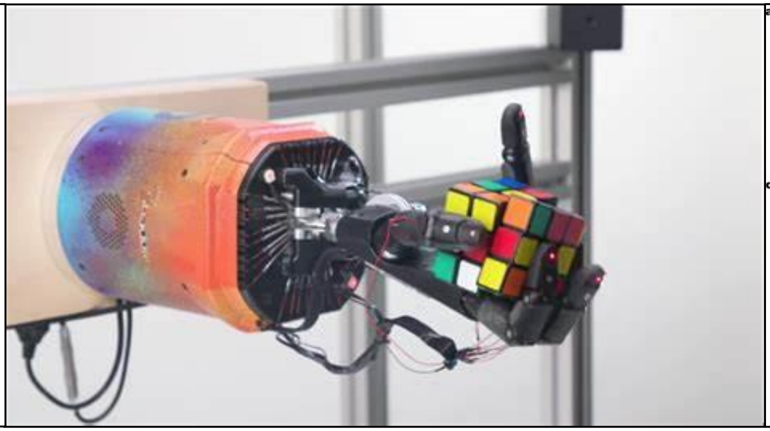


Reinforcement learning

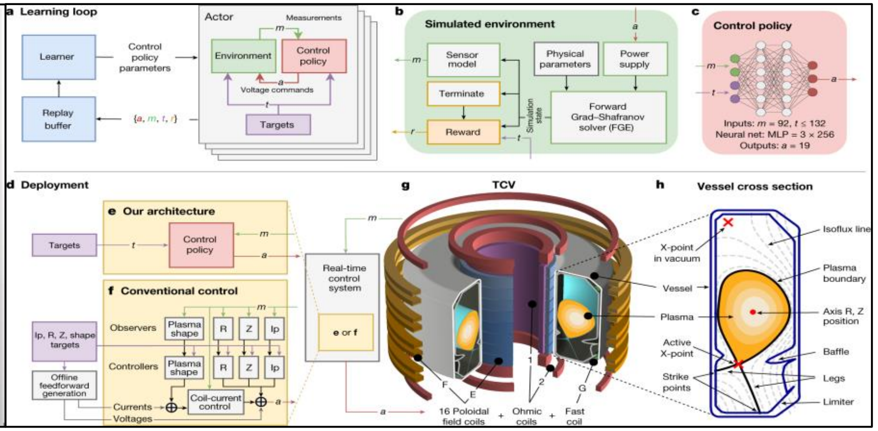
Reinforcement learning



MuZero
Schrittwieser, J. et al. Nature

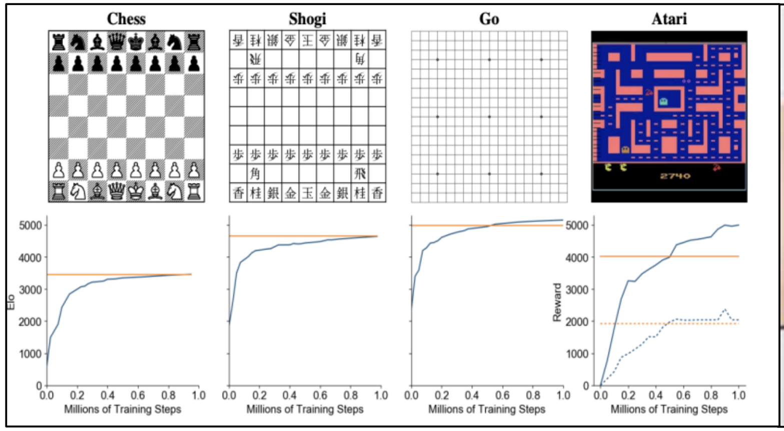


OpenAI's robot hand
OpenAI

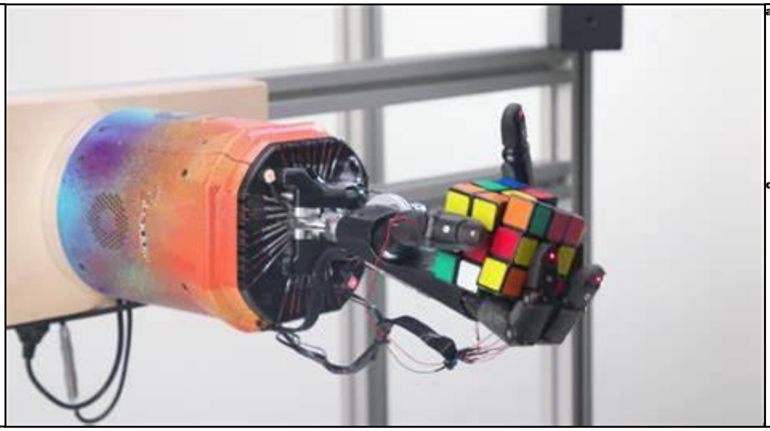


Magnetic control of tokamak plasmas
Degrave, J. et al. Nature

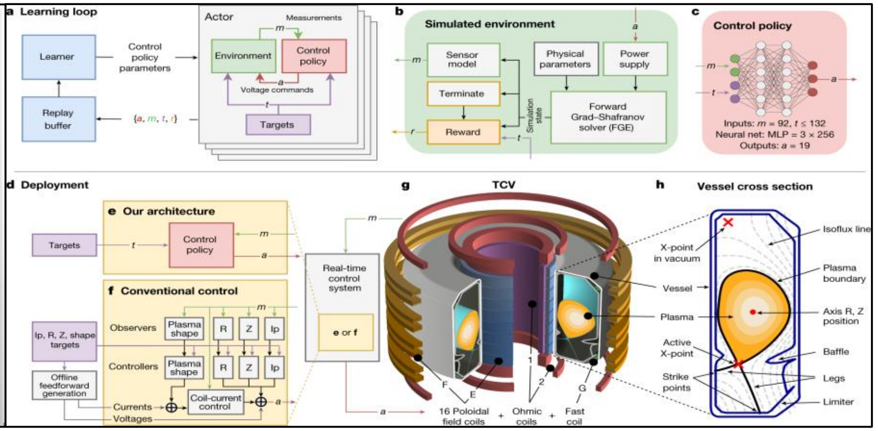
Reinforcement learning



MuZero
Schrittwieser, J. et al. Nature



OpenAI’s robot hand
OpenAI



Magnetic control of tokamak plasmas
Degrave, J. et al. Nature

Deep reinforcement learning (RL)

- Great performance in many domains
- Brittle to tune

Combining with Supervised learning

Deep reinforcement learning (RL)

- Great performance in many domains
- Brittle to tune

Combining with Supervised learning

Deep reinforcement learning (RL)

- Great performance in many domains
- Brittle to tune

Supervised learning (SL)

- Steady optimization
- Require dataset and rely on its quality

Combining with Supervised learning

Deep reinforcement learning (RL)
- Great performance in many domains
- ~~Brittle to tune~~

Supervised learning (SL)
- Steady optimization
- ~~Require dataset and rely on its quality~~

Combine SL and RL
- Great performance
- Steady optimization
- Not require dataset

How to combine RL and SL

How to combine RL and SL

- Self-imitation learning (SIL)

How to combine RL and SL

- Self-imitation learning (SIL)
  - Optimize RL and SL objectives jointly (non-phasic)

How to combine RL and SL

- Self-imitation learning (SIL)
  - Optimize RL and SL objectives jointly (non-phasic)
  - Even more brittle to optimize the mixed objective

How to combine RL and SL

- Self-imitation learning (SIL)
  - Optimize RL and SL objectives jointly (non-phasic)
  - Even more brittle to optimize the mixed objective
- Our method

How to combine RL and SL

- Self-imitation learning (SIL)
  - Optimize RL and SL objectives jointly (non-phasic)
  - Even more brittle to optimize the mixed objective
- Our method
  - **Phasic** optimization process
    - Alternate between RL phase and SL phase
    - Optimize RL and SL objectives in two separate phases

How to combine RL and SL

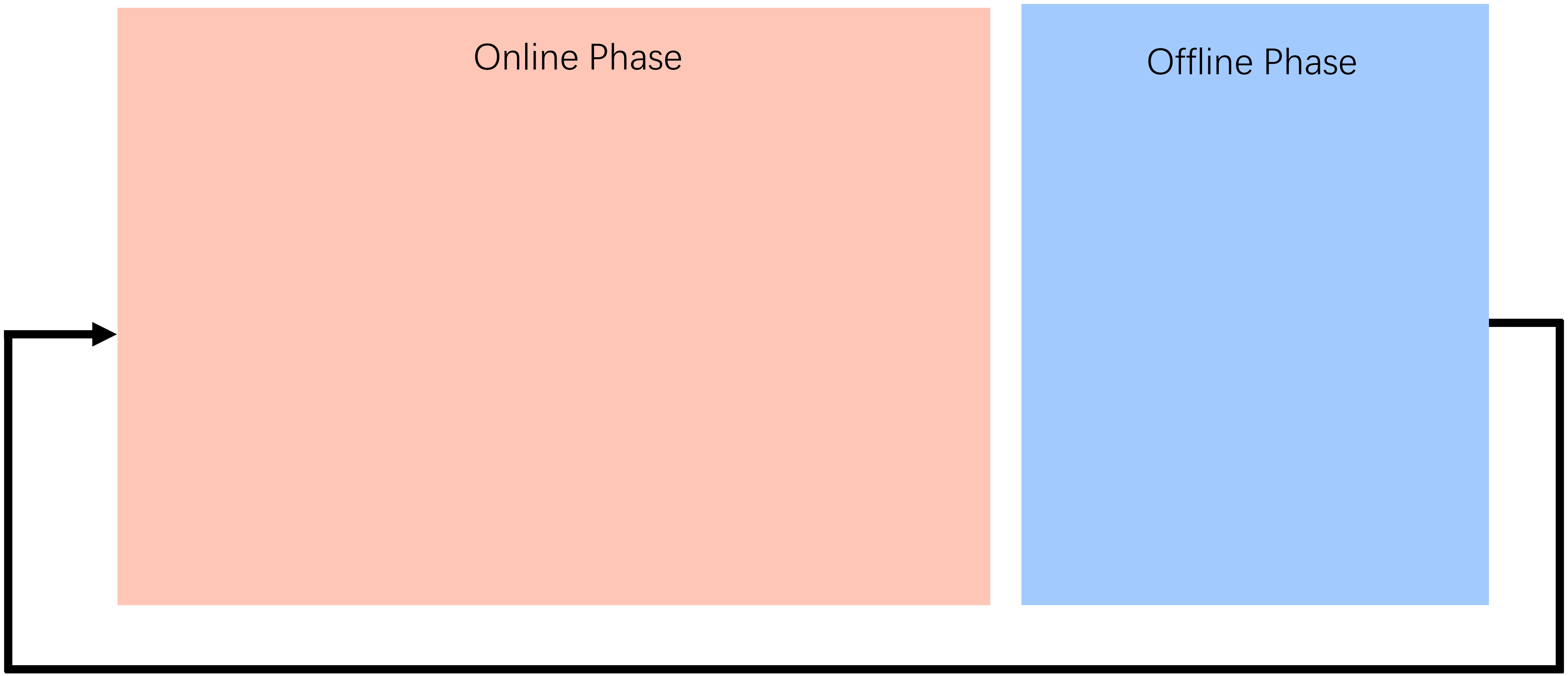
- Self-imitation learning (SIL)
  - Optimize RL and SL objectives jointly (non-phasic)
  - Even more brittle to optimize the mixed objective
- Our method
  - **Phasic** optimization process
    - Alternate between RL phase and SL phase
    - Optimize RL and SL objectives in two separate phases
  - Tackle sparse-reward problems effectively

How to combine RL and SL

- Self-imitation learning (SIL)
  - Optimize RL and SL objectives jointly (non-phasic)
  - Even more brittle to optimize the mixed objective
- Our method
  - **Phasic** optimization process
    - Alternate between RL phase and SL phase
    - Optimize RL and SL objectives in two separate phases
  - Tackle sparse-reward problems effectively
  - No need for datasets

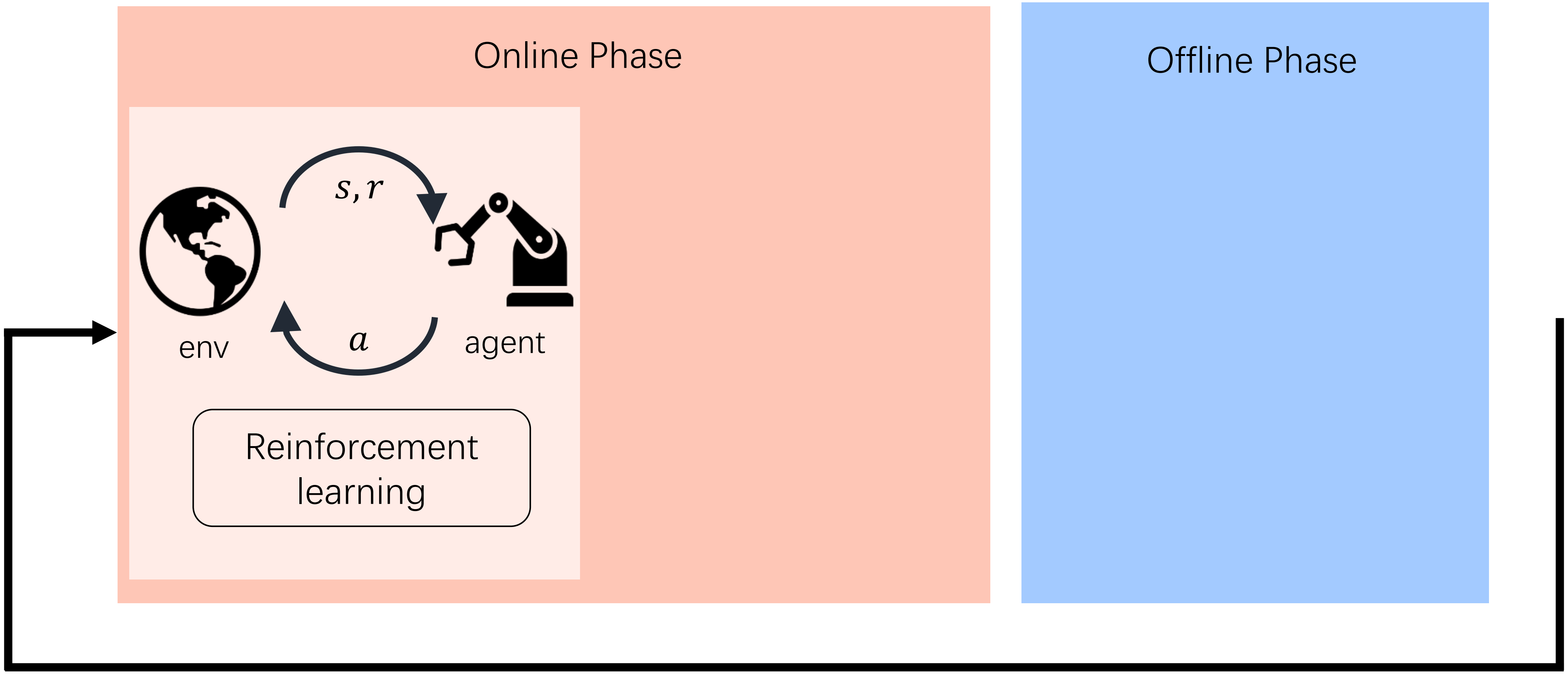
Our framework

Ph**A**sic self-**I**mitative **R**eduction (**PAIR**)



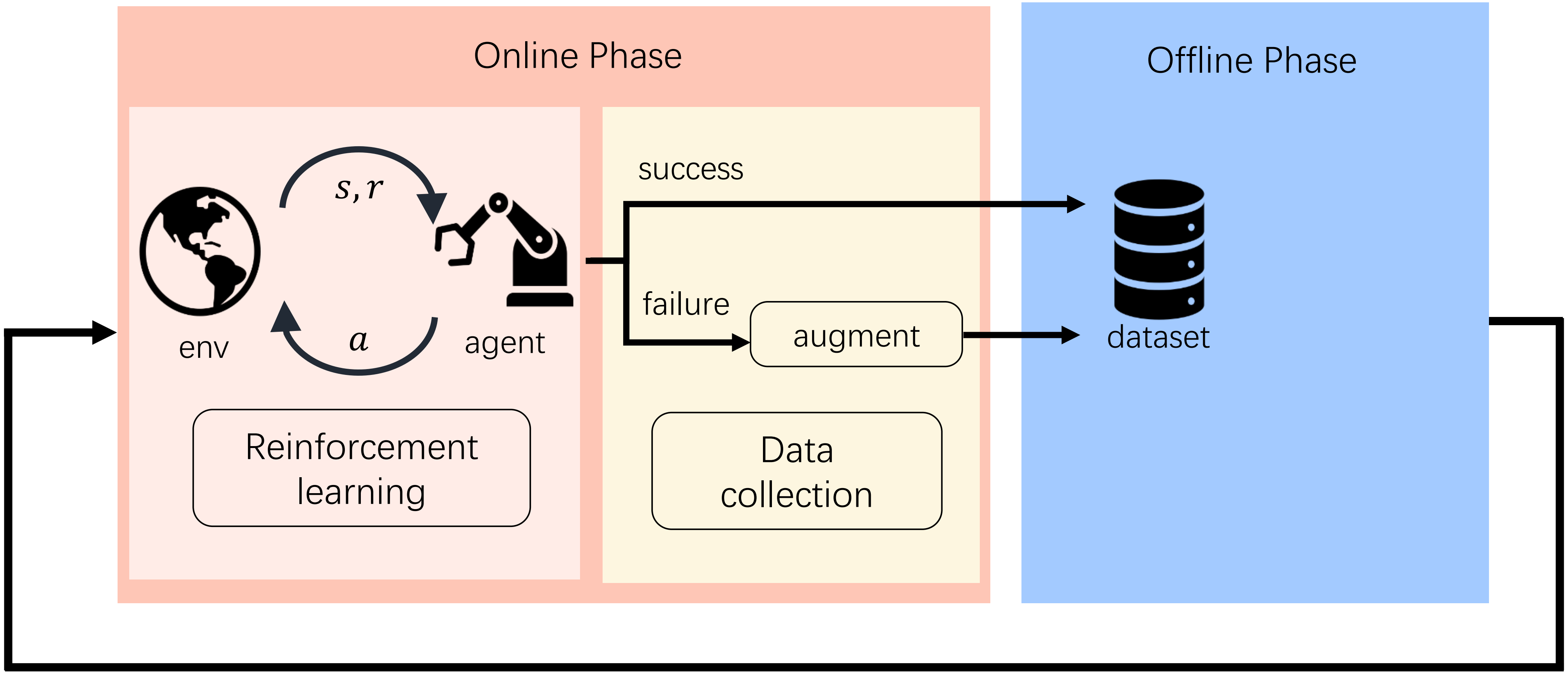
Our framework

PhAsic self-Imitative Reduction (PAIR)



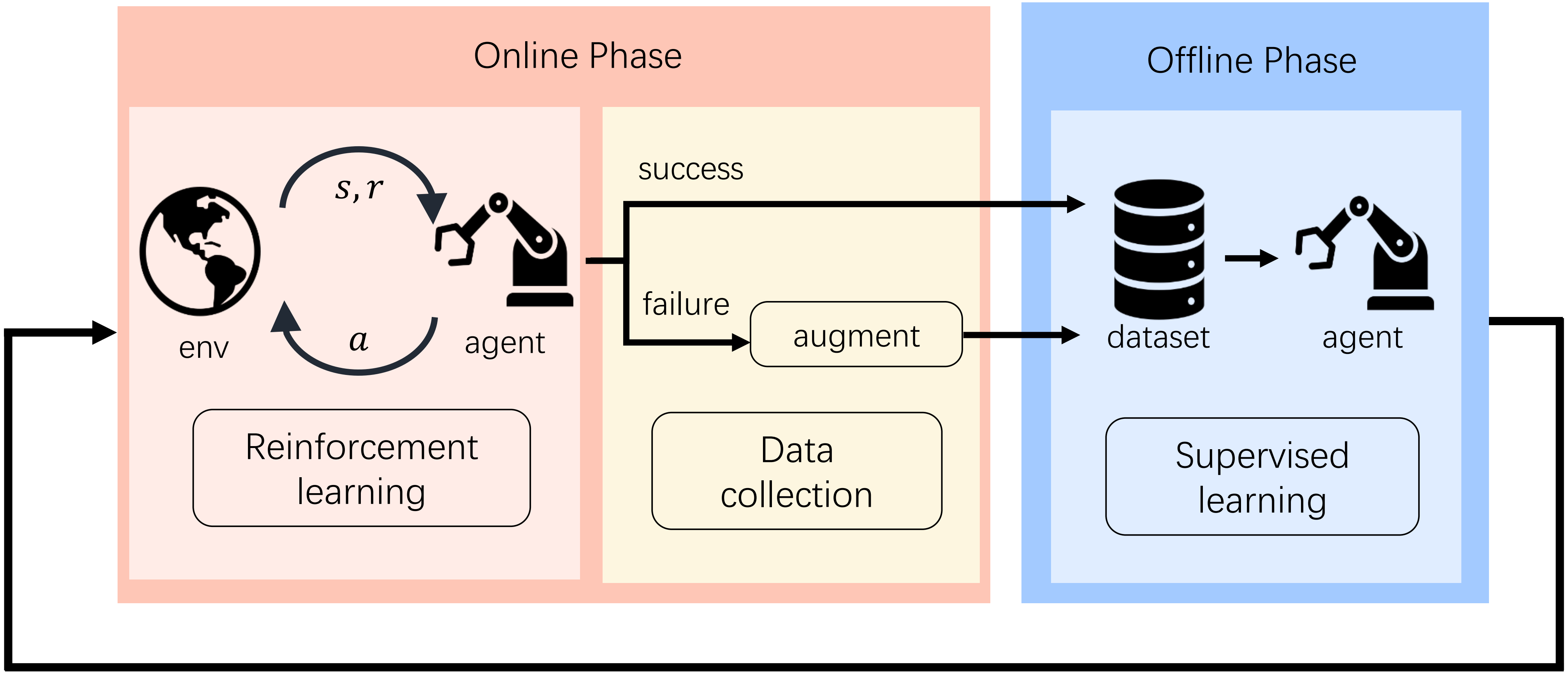
Our framework

PhAsic self-Imitative Reduction (PAIR)



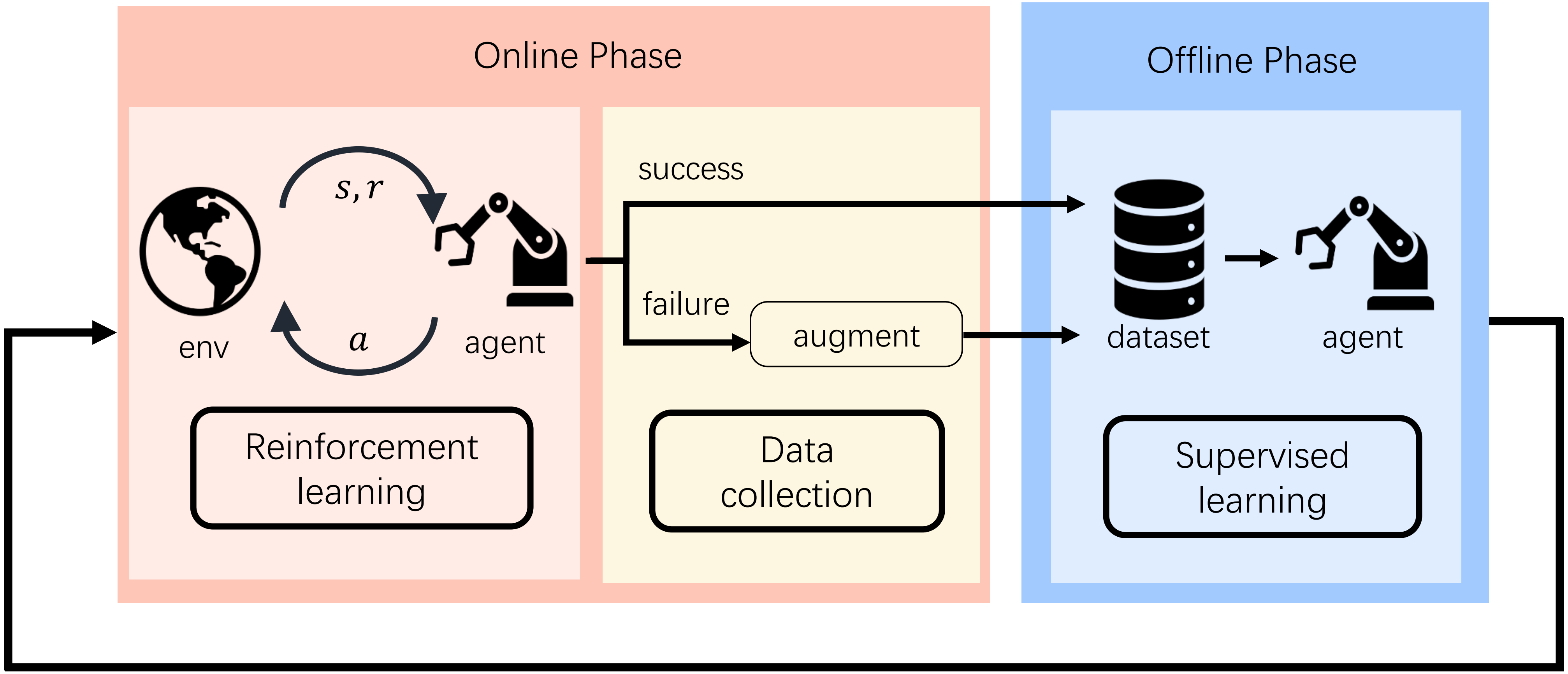
Our framework

PhAsic self-Imitative Reduction (PAIR)

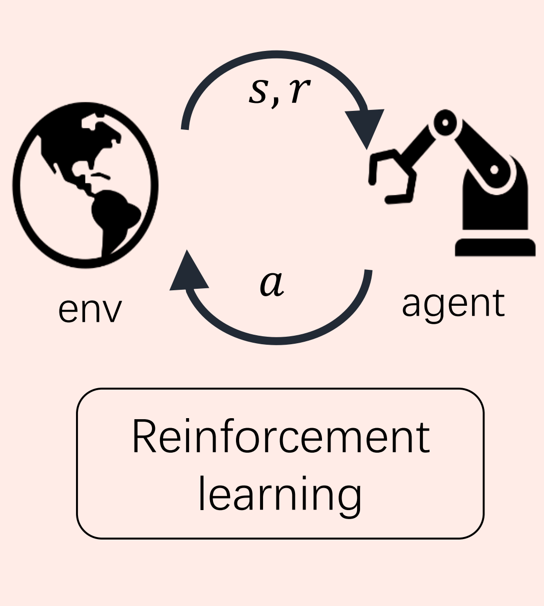


Our framework

PhAsic self-Imitative Reduction (PAIR)

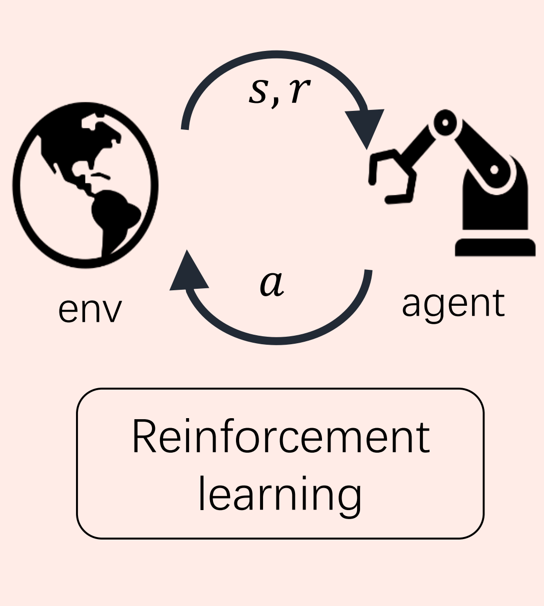


Online phase: RL with intrinsic reward

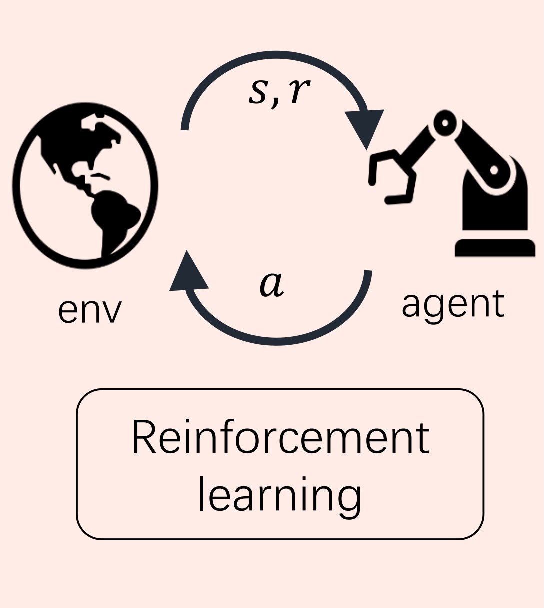


Online phase: RL with intrinsic reward

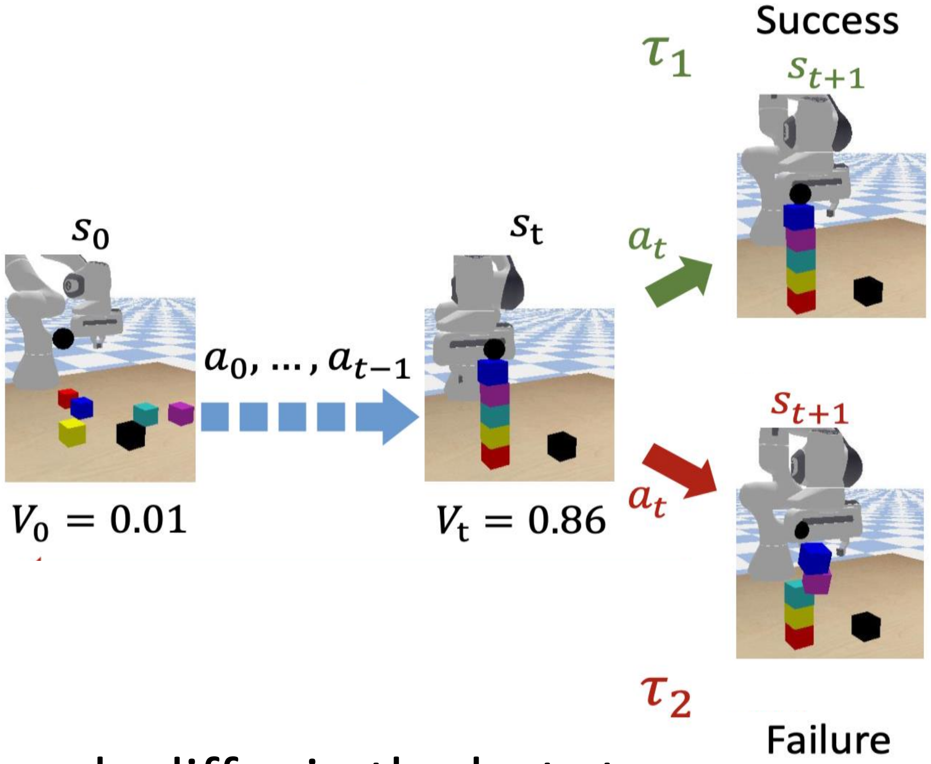
Sparse reward issue for online RL



Online phase: RL with intrinsic reward

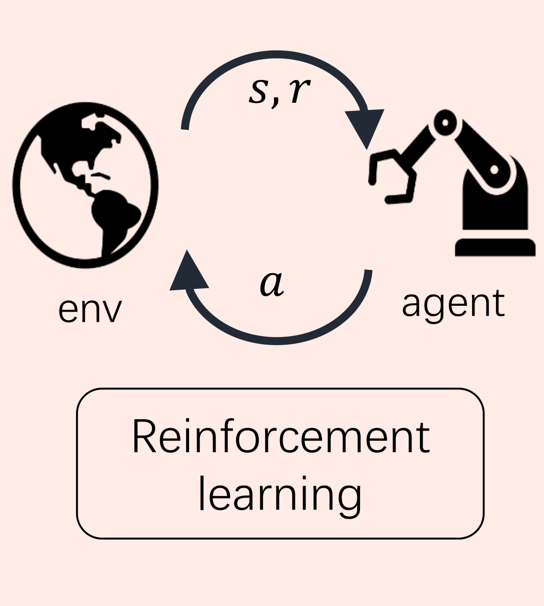


Sparse reward issue for online RL

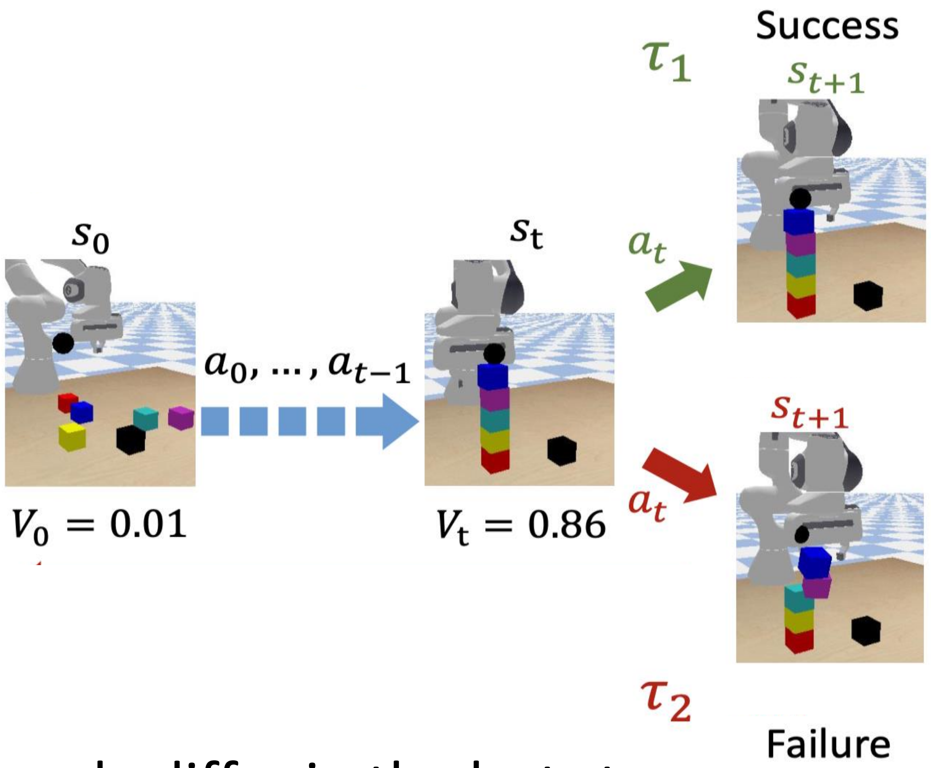


- Successful τ_1 and failed τ_2 only differ in the last step

Online phase: RL with intrinsic reward

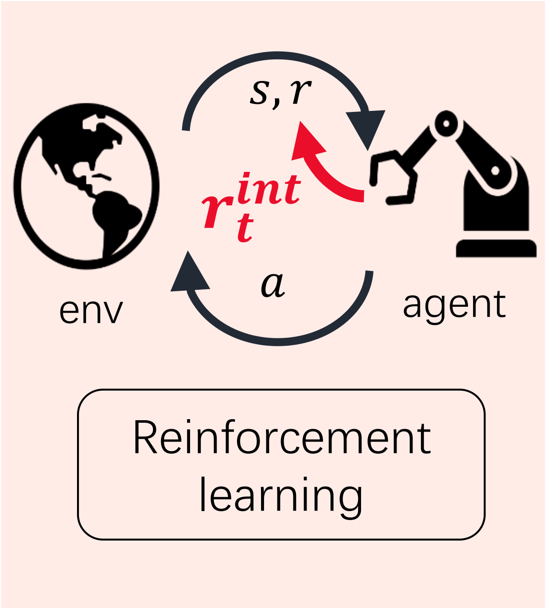


Sparse reward issue for online RL

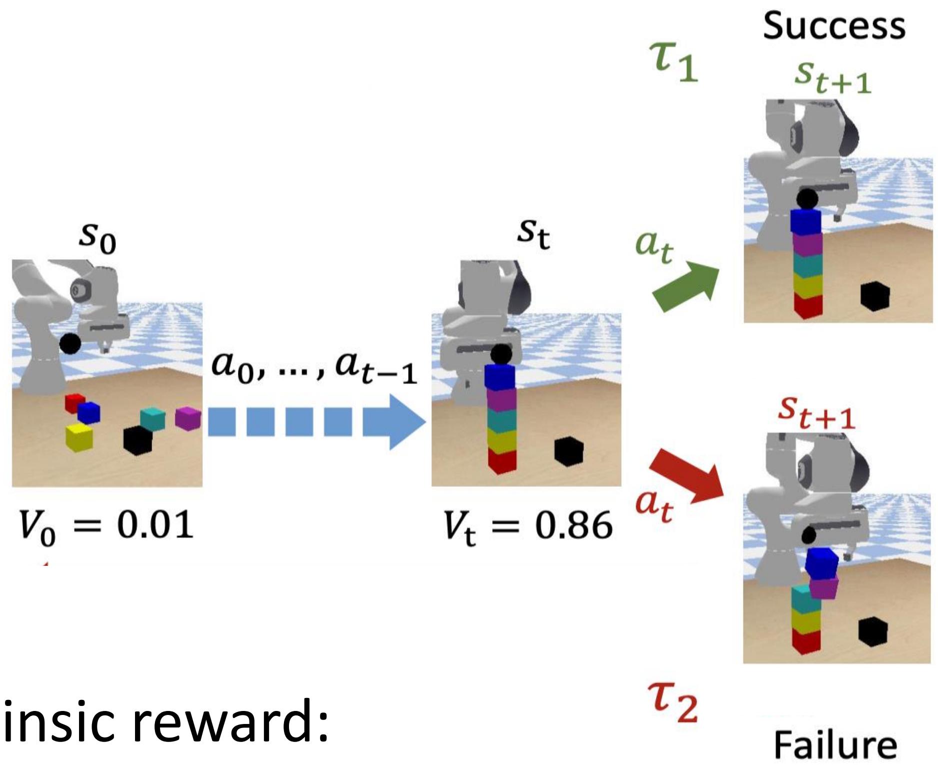


- Successful τ_1 and failed τ_2 only differ in the last step
- All actions in τ_2 are considered bad based on the final reward 0

Online phase: RL with intrinsic reward



Sparse reward issue for online RL

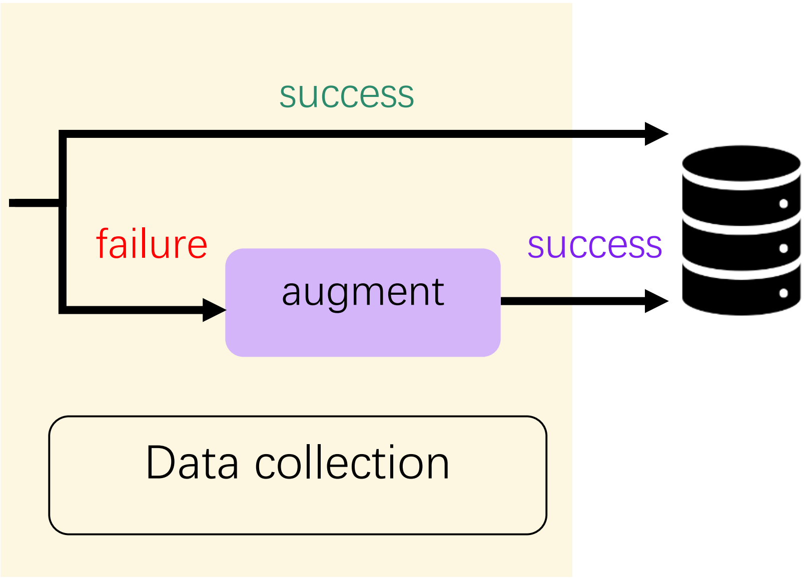


Add a **value-difference** intrinsic reward:

$$r^{\text{int}}(s_t, a_t, g) \coloneqq V_\psi(s_{t+1}, g) - V_\psi(s_t, g)$$

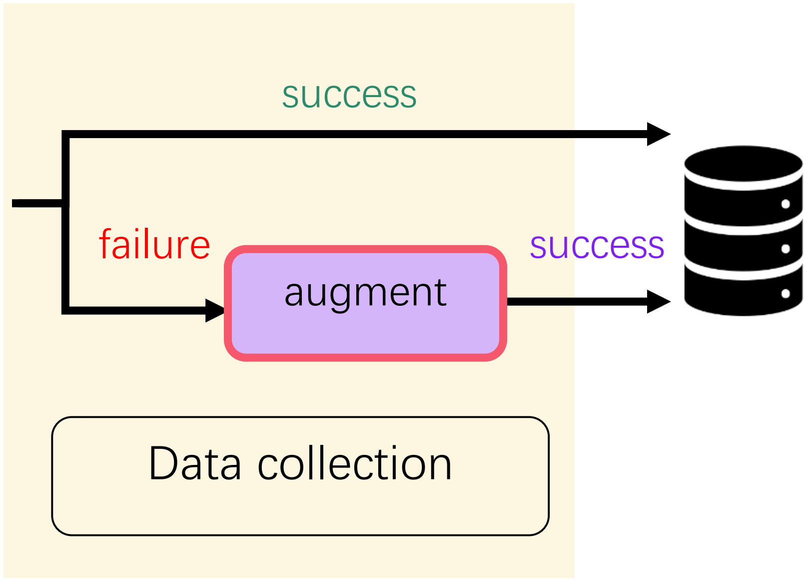
- Difference of V can capture whether a transition is approaching the goal

Online phase: data collection



Successful rollouts + augmented data

Online phase: data collection

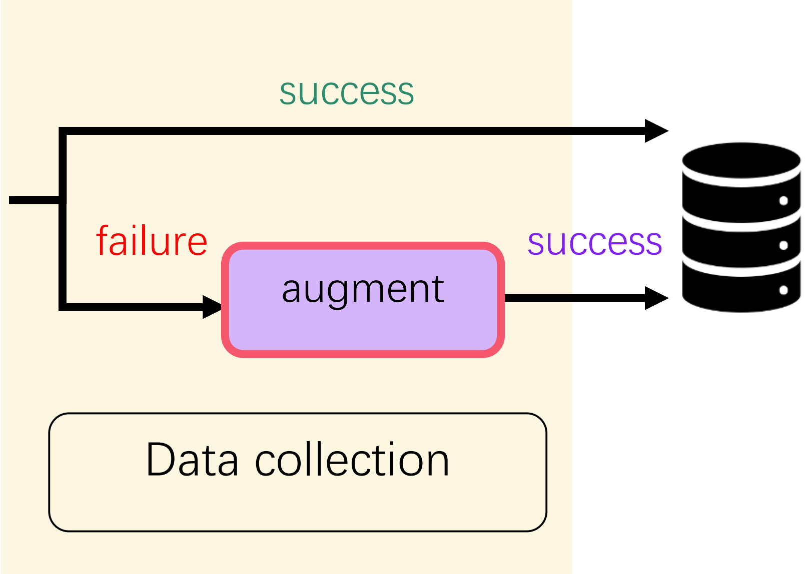


Successful rollouts + augmented data

Augmentation

- Goal relabeling
- Task reduction

Online phase: data collection



Successful rollouts + augmented data

Augmentation

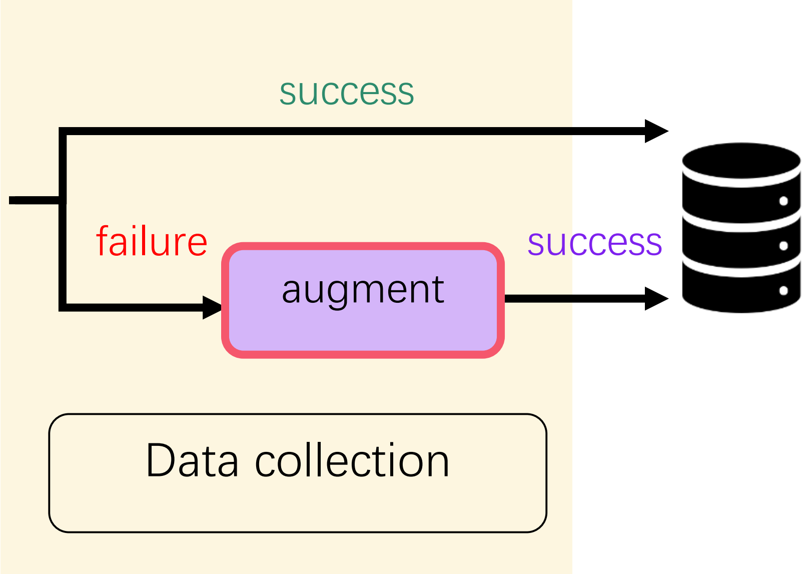
- Goal relabeling
- Task reduction

Goal relabeling

$(s_0, \ldots, s_T; g \neq s_T) \rightarrow (s_0, \ldots, s_T; g' = s_T)$

Failure → Success

Online phase: data collection

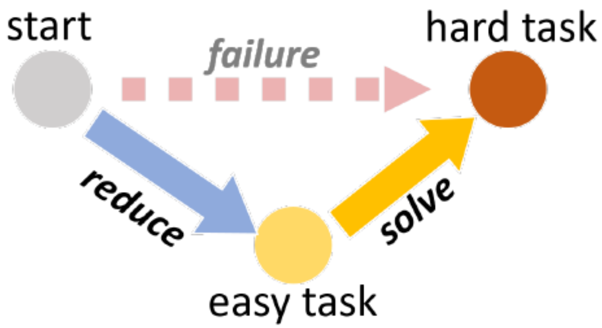


Successful rollouts + augmented data

Augmentation

- Goal relabeling
- Task reduction

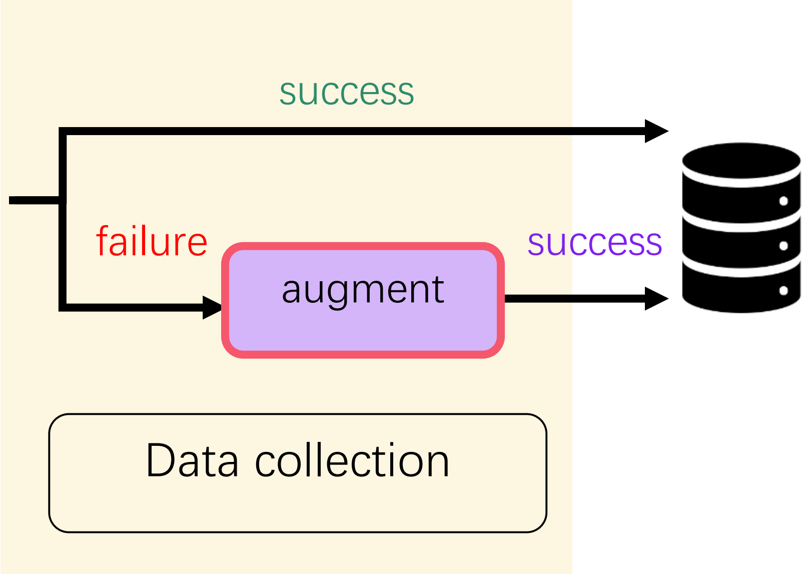
Task reduction



- Decompose a challenging task into two simpler sub-tasks

$$(s_0, g) \rightarrow (s_0, s_B) + (s_B, g)$$

Online phase: data collection

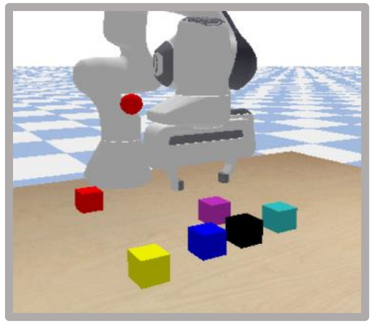


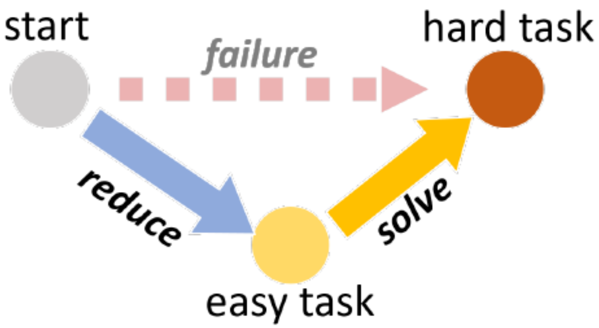
Successful rollouts + augmented data

Augmentation

- Goal relabeling
- Task reduction

Task reduction

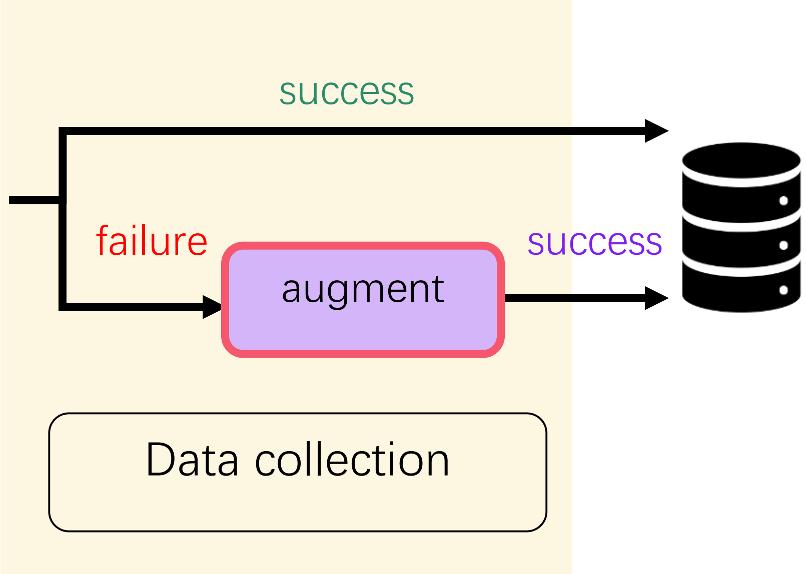




- Decompose a challenging task into two simpler sub-tasks

$$(s_0, g) \rightarrow (s_0, s_B) + (s_B, g)$$

Online phase: data collection

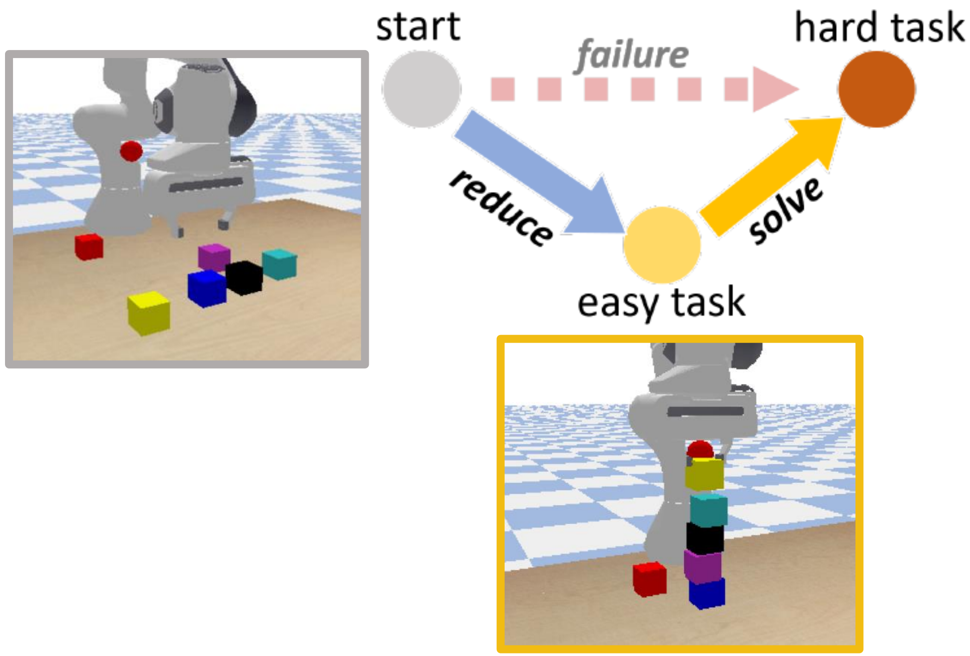


Successful rollouts + augmented data

Augmentation

- Goal relabeling
- Task reduction

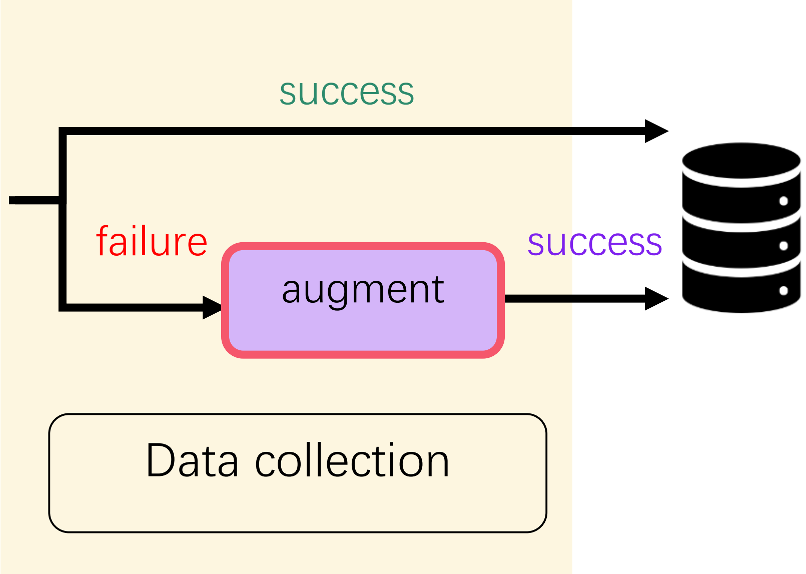
Task reduction



- Decompose a challenging task into two simpler sub-tasks

$$(s_0, g) \rightarrow (s_0, s_B) + (s_B, g)$$

Online phase: data collection

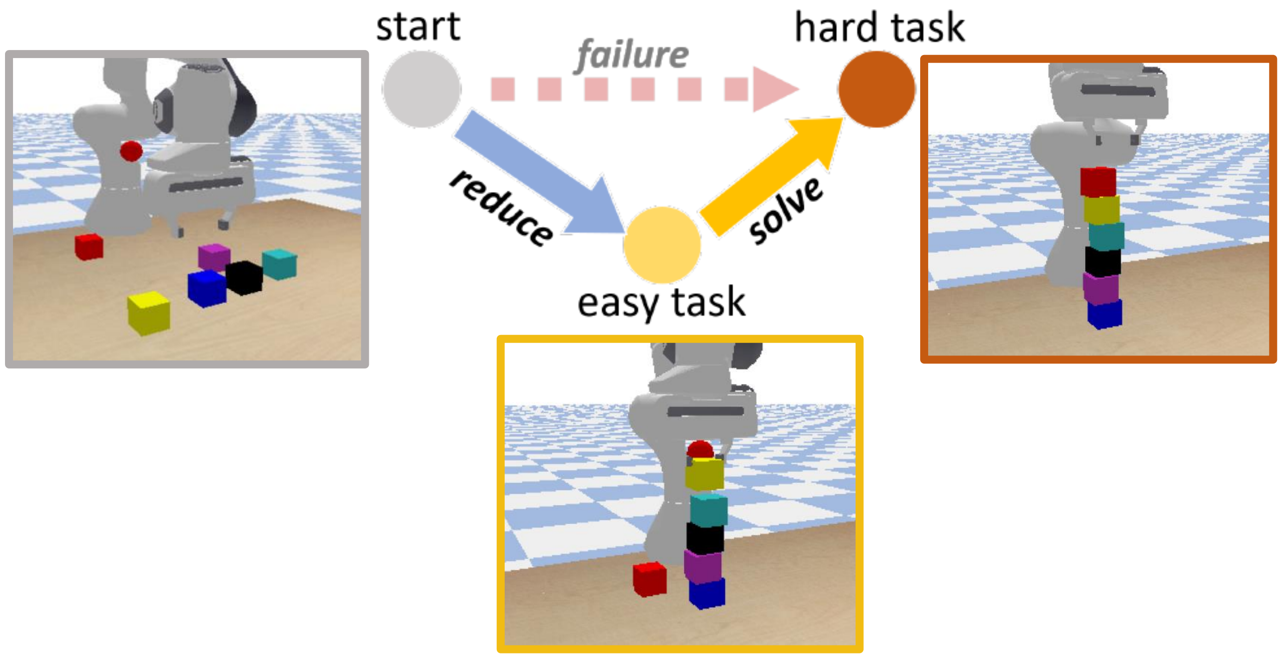


Successful rollouts + augmented data

Augmentation

- Goal relabeling
- Task reduction

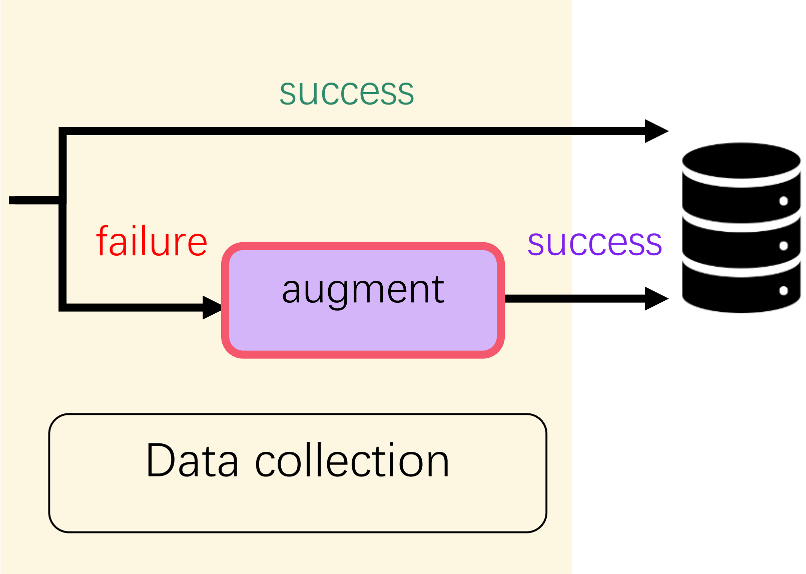
Task reduction



- Decompose a challenging task into two simpler sub-tasks

$$(s_0, g) \rightarrow (s_0, s_B) + (s_B, g)$$

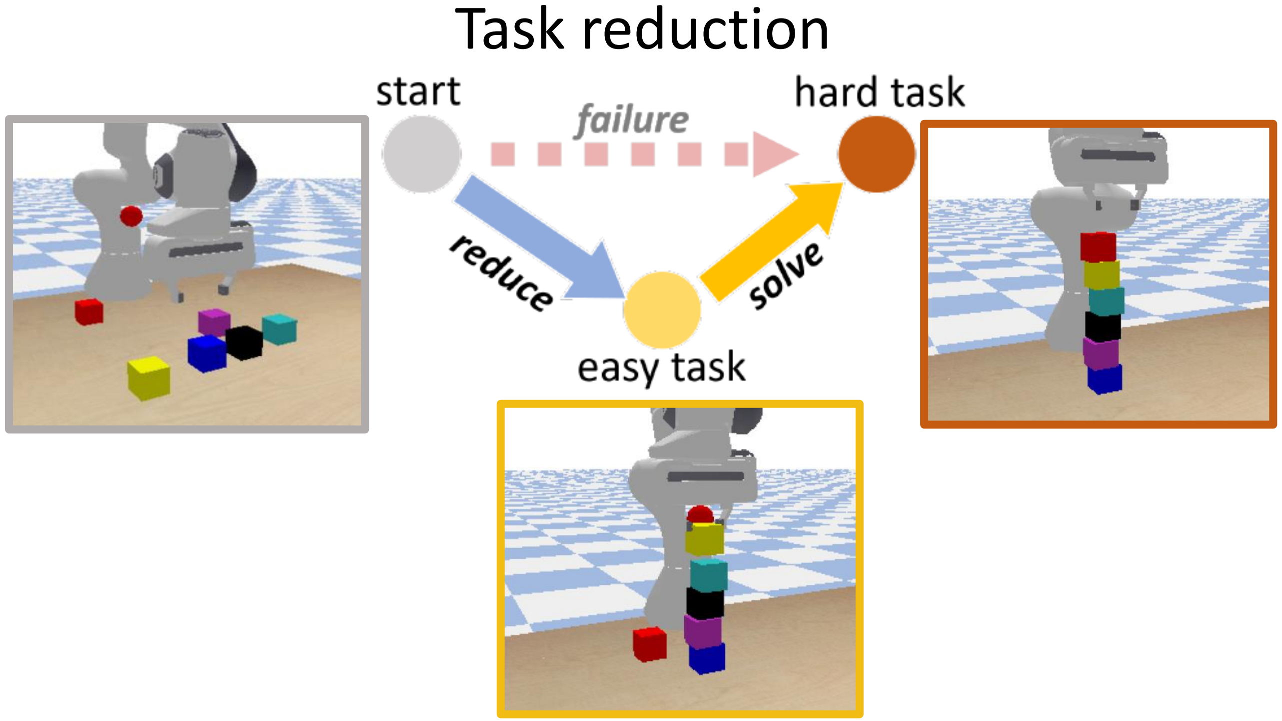
Online phase: data collection



Successful rollouts + augmented data

Augmentation

- Goal relabeling
- Task reduction

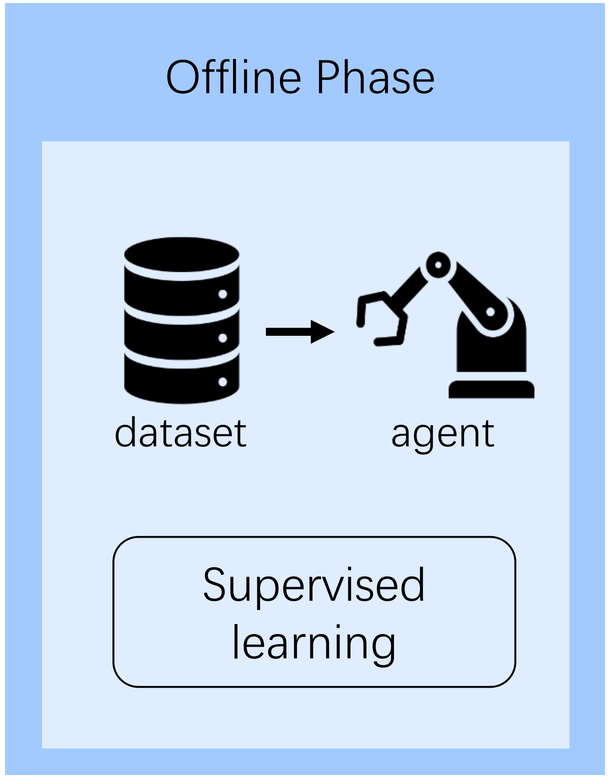


- Decompose a challenging task into two simpler sub-tasks

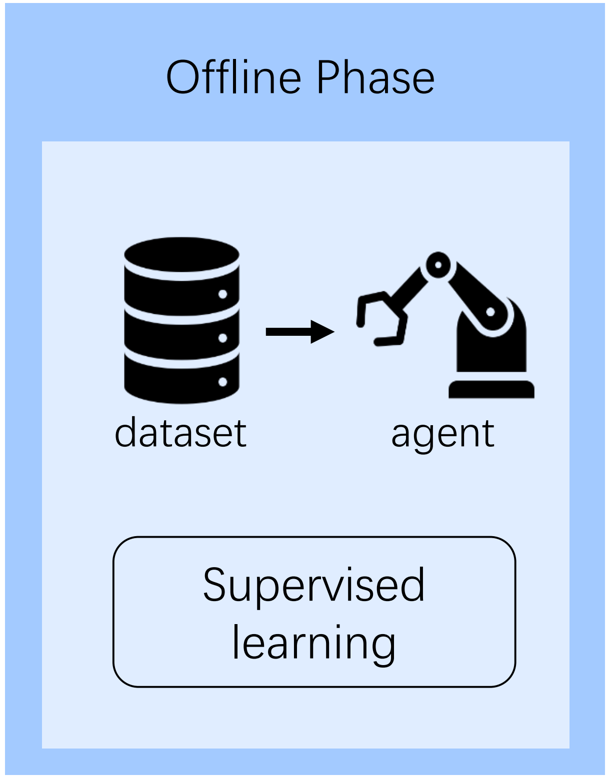
$$(s_0, g) \rightarrow (s_0, s_B) + (s_B, g)$$

- $s_B^\star = \arg\max_{s_B} V_\psi(s_0, s_B) \oplus V_\psi(s_B, g)$

Offline phase

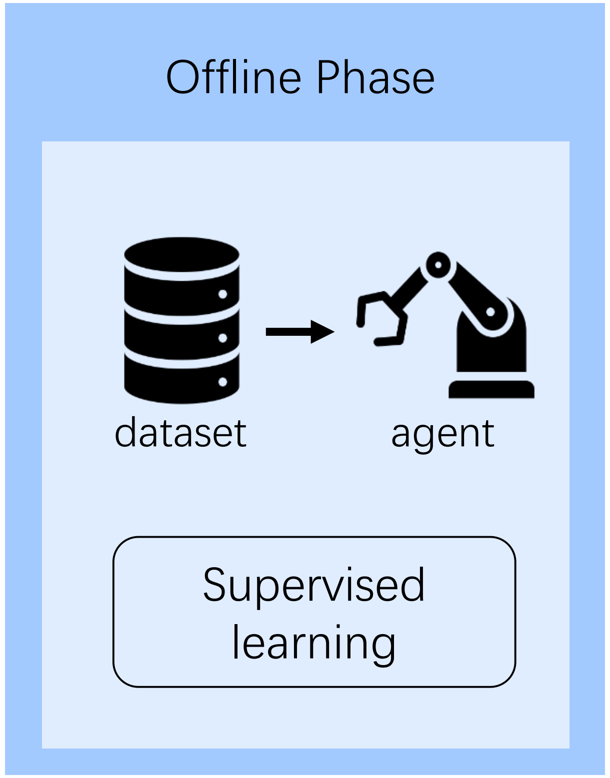


Offline phase



- Advantage weighted behavior cloning
  - $L(\theta) = -\mathbb{E}_{(g;s,a)\in\mathcal{D}}[w(s,a,g)\log\pi(a|s,g)]$,
  - $w(s,a,g) = \exp\left(\frac{1}{\beta}\left(R - V_{\phi}(s,g)\right)\right)$

Offline phase



- Advantage weighted behavior cloning
  - $L(\theta) = -\mathbb{E}_{(g;s,a)\in\mathcal{D}}[w(s,a,g)\log\pi(a|s,g)],$
  - $w(s,a,g) = \exp\left(\frac{1}{\beta}\left(R - V_\phi(s,g)\right)\right)$
- It is feasible to adopt advanced offline RL methods

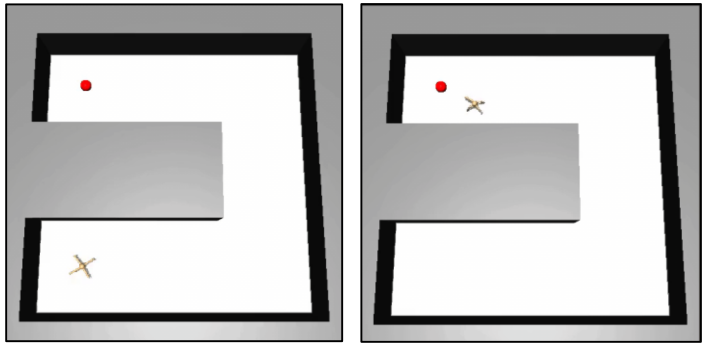
Experimental tasks

Goal-conditioned control tasks with ONLY 0-1 sparse reward

Experimental tasks

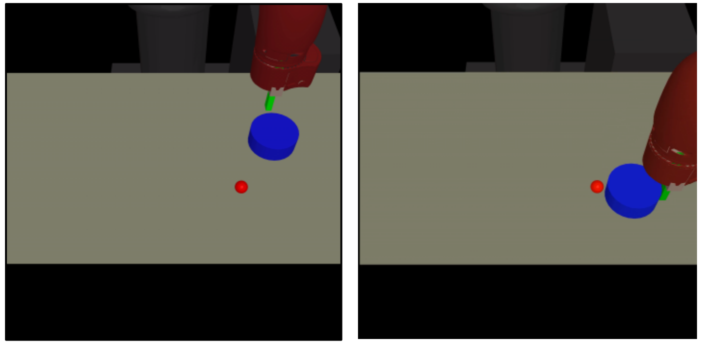
Goal-conditioned control tasks with ONLY 0-1 sparse reward

Initial state Goal state



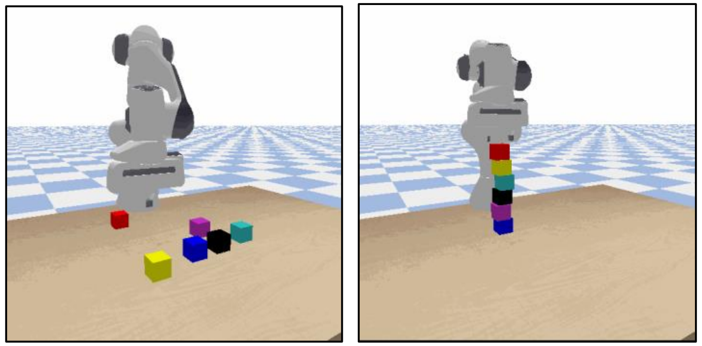
Ant Maze

Initial state Goal state



Sawyer Push

Initial state Goal state

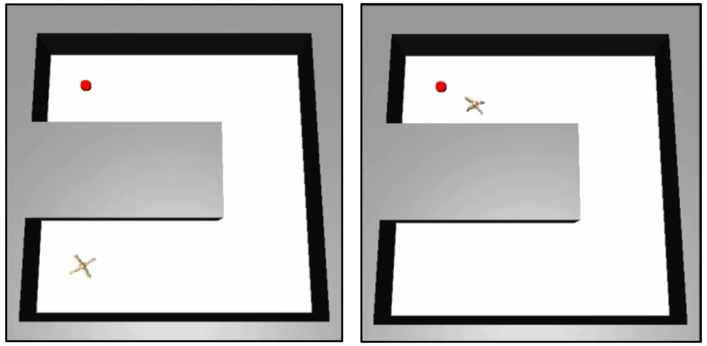


Cube Stacking

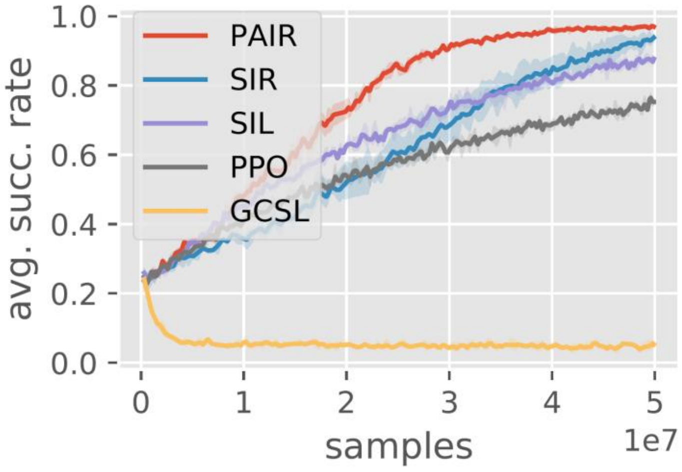
Experimental tasks

Goal-conditioned control tasks with ONLY 0-1 sparse reward

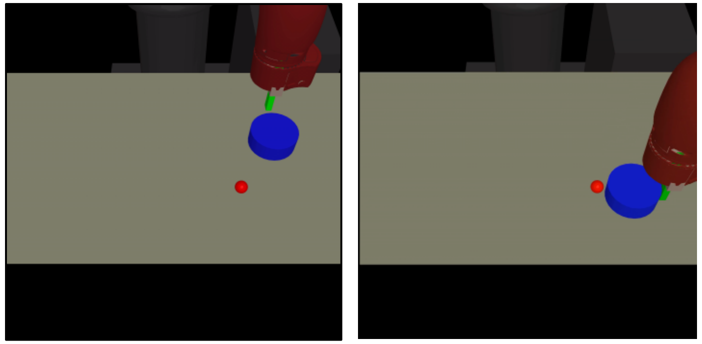
Initial state | Goal state



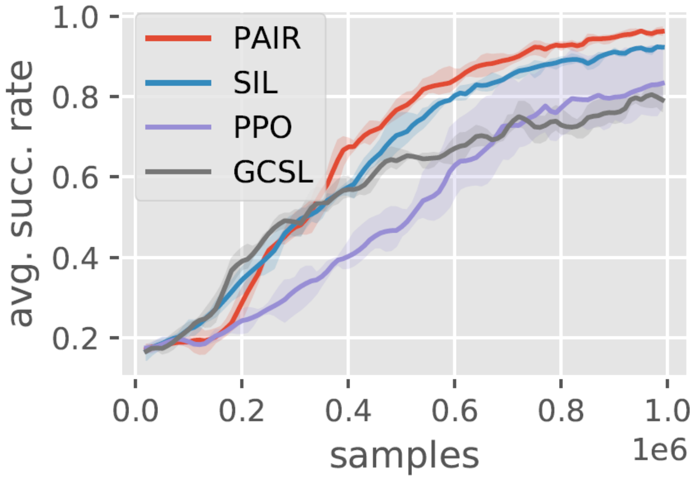
Ant Maze



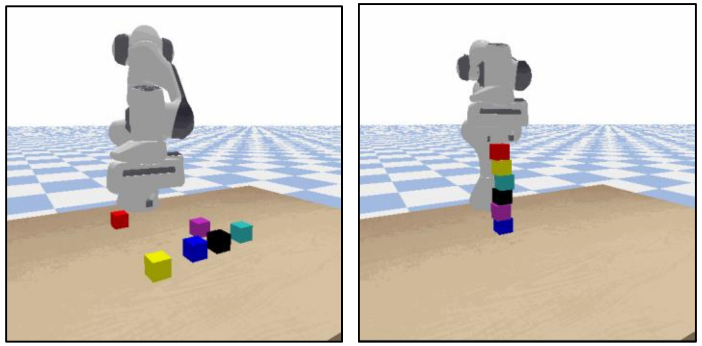
Initial state | Goal state



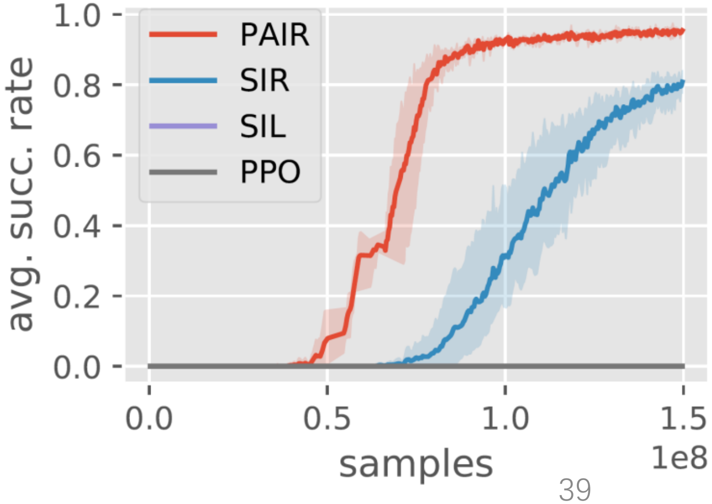
Sawyer Push



Initial state | Goal state



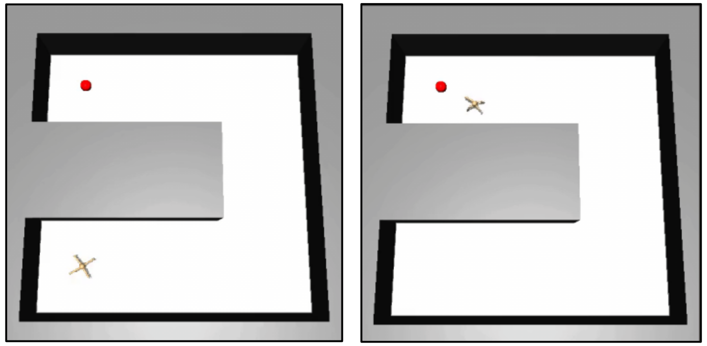
Cube Stacking



Experimental tasks

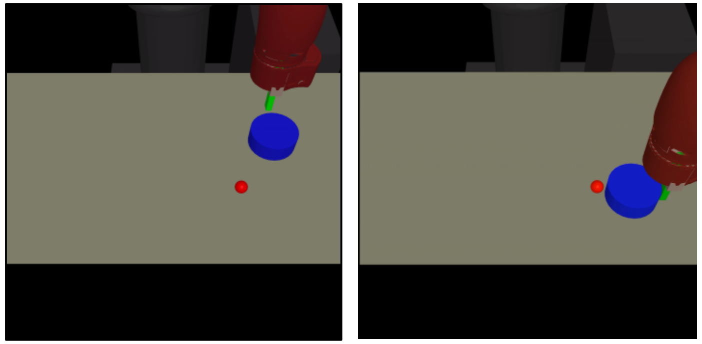
Goal-conditioned control tasks with ONLY 0-1 sparse reward

Initial state Goal state



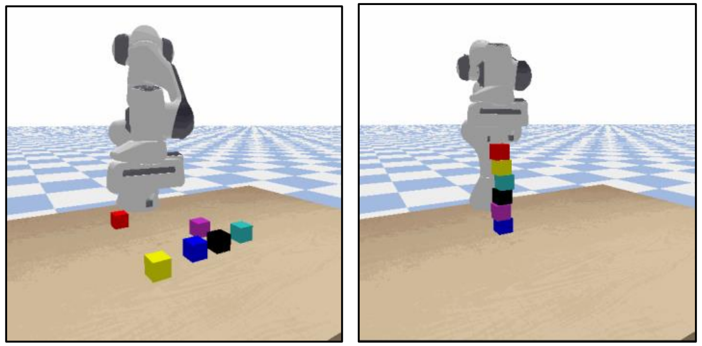
Ant Maze

Initial state Goal state

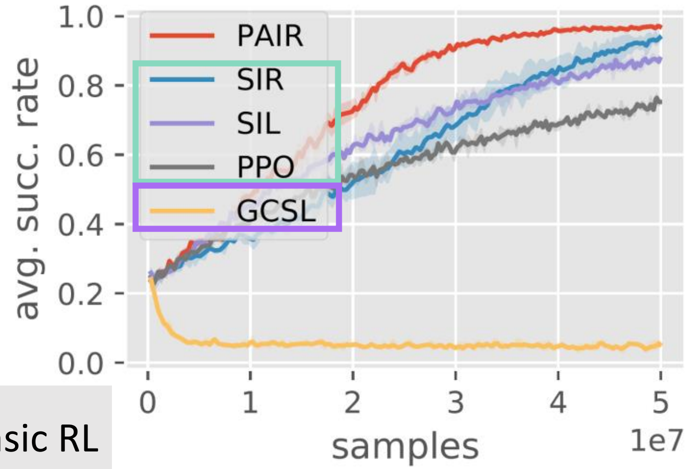


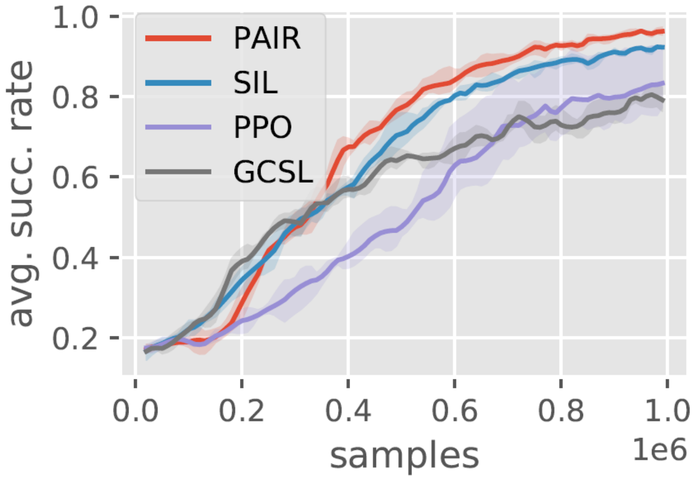
Sawyer Push

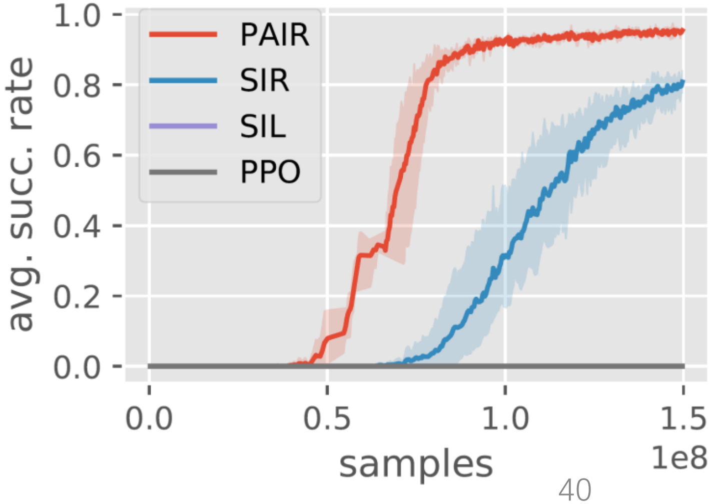
Initial state Goal state



Cube Stacking





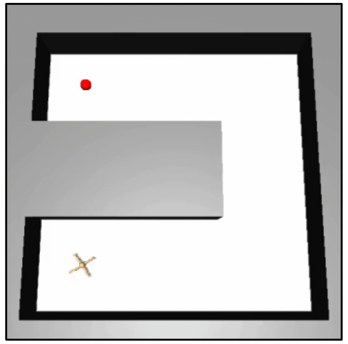


Non-phasic RL
Phasic SL

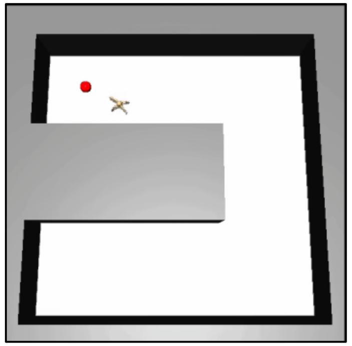
Experimental tasks

Goal-conditioned control tasks with ONLY 0-1 sparse reward

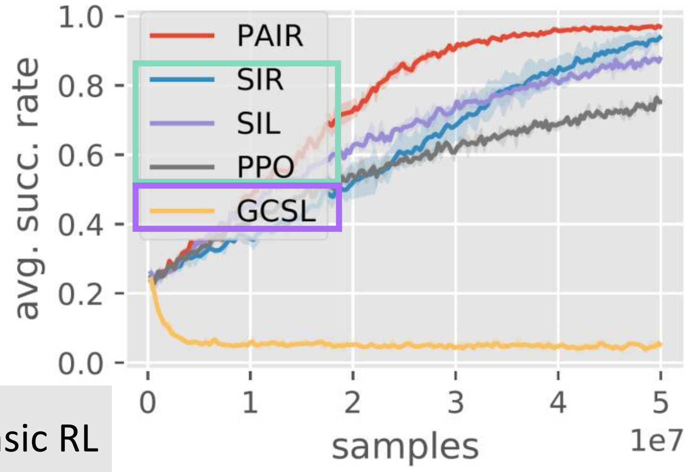
Initial state



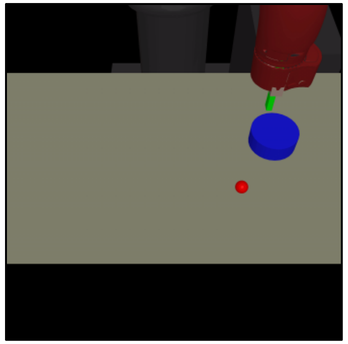
Goal state



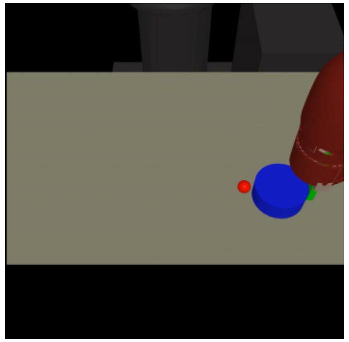
Ant Maze



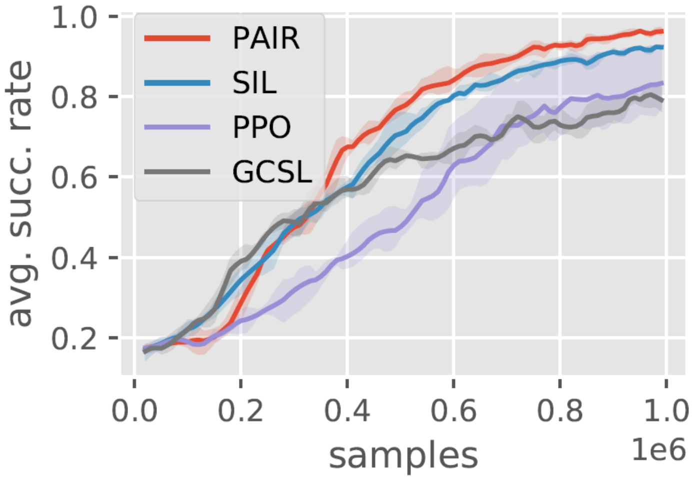
Initial state



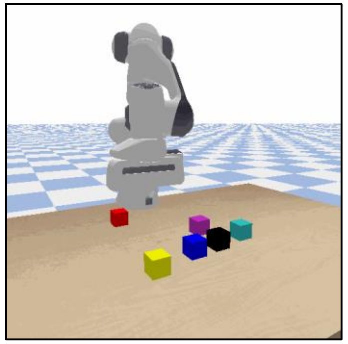
Goal state



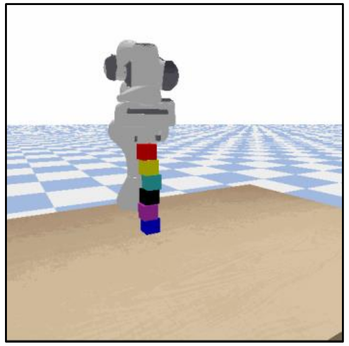
Sawyer Push



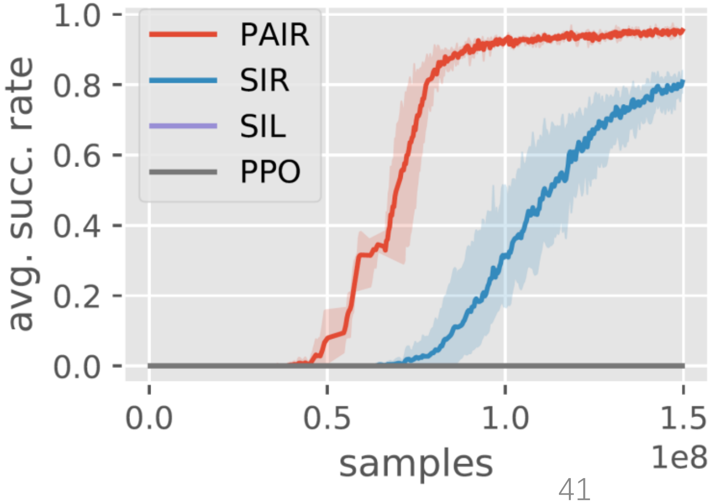
Initial state



Goal state



Cube Stacking



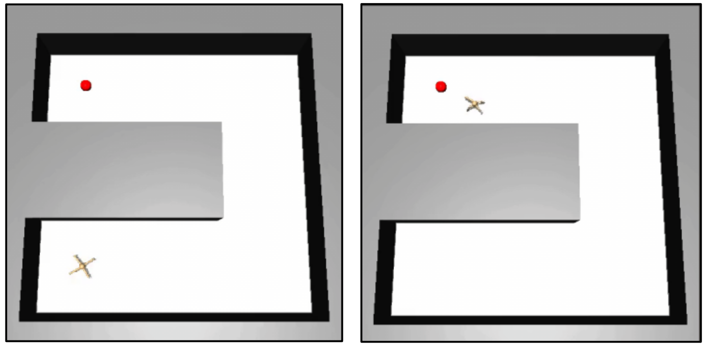
Non-phasic RL

Phasic SL

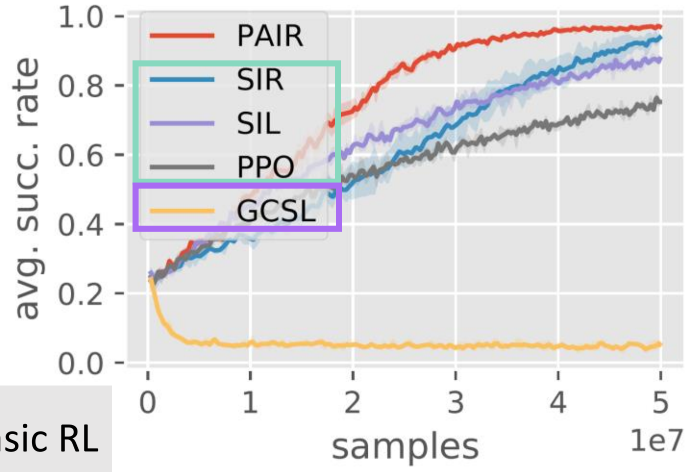
Experimental tasks

Goal-conditioned control tasks with ONLY 0-1 sparse reward

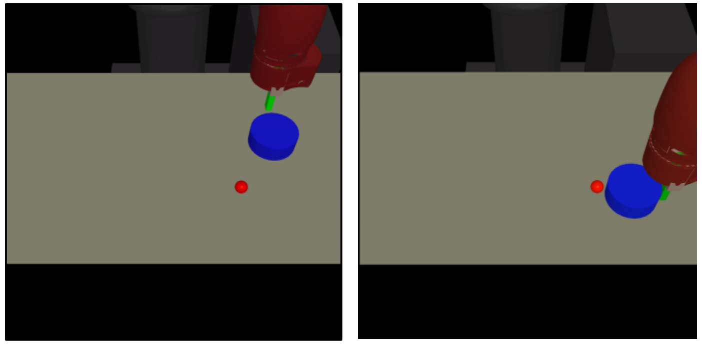
Initial state Goal state



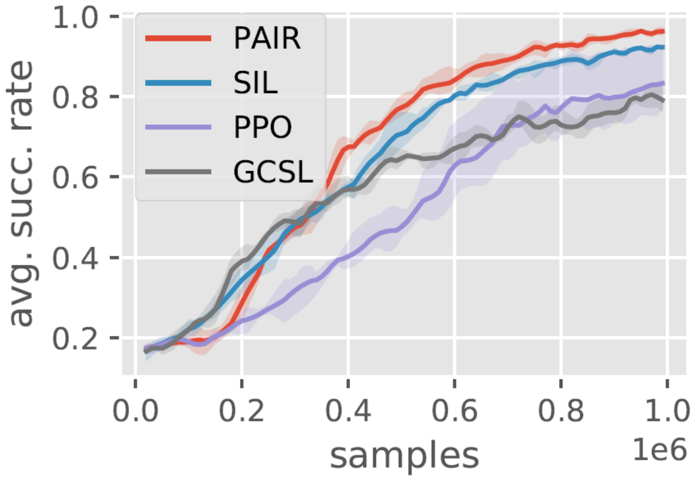
Ant Maze



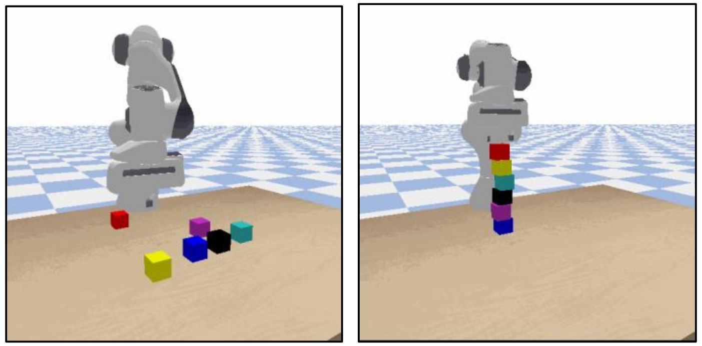
Initial state Goal state



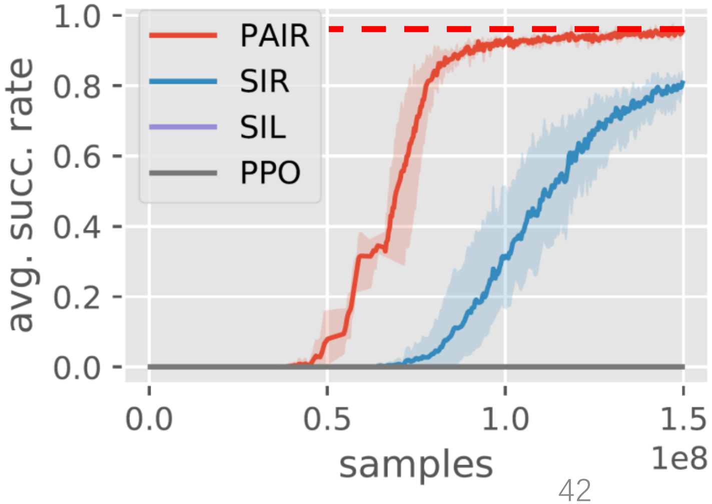
Sawyer Push



Initial state Goal state

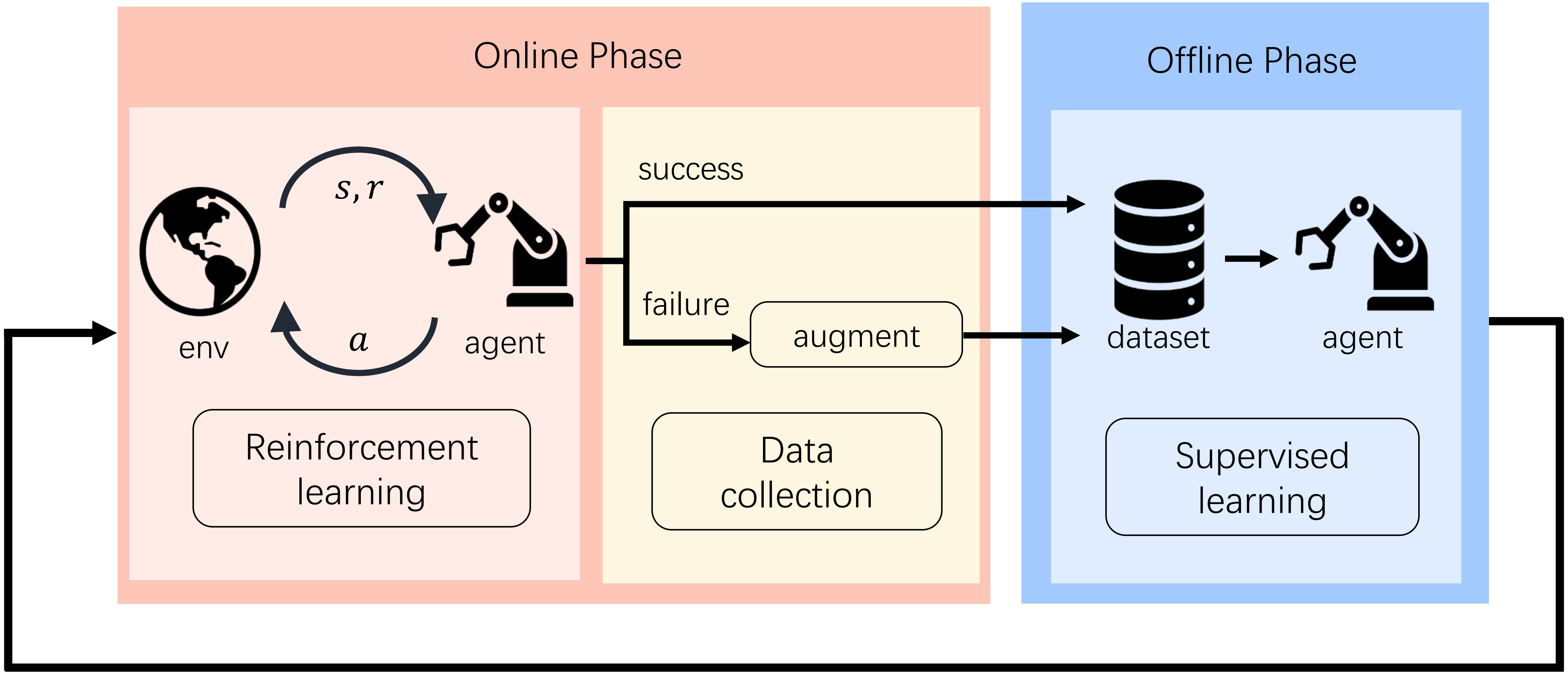


Cube Stacking



Non-phasic RL
Phasic SL

PhAsic self-Imitative Reduction (PAIR)



https://sites.google.com/view/pair-gcrl