

Frontiers and Open Challenges

CS330

Logistics

Poster session on Weds

Details on Ed.

Final project report

Due next Monday.

This is our last lecture!

From high-resolution feedback

- If you are remote & need TA mentor input, email them to set-up a zoom meeting.
- Will briefly discuss the connection between topics/lectures

Plan for Today

Meta reinforcement learning

The meta-RL problem set-up

Black-box meta-RL

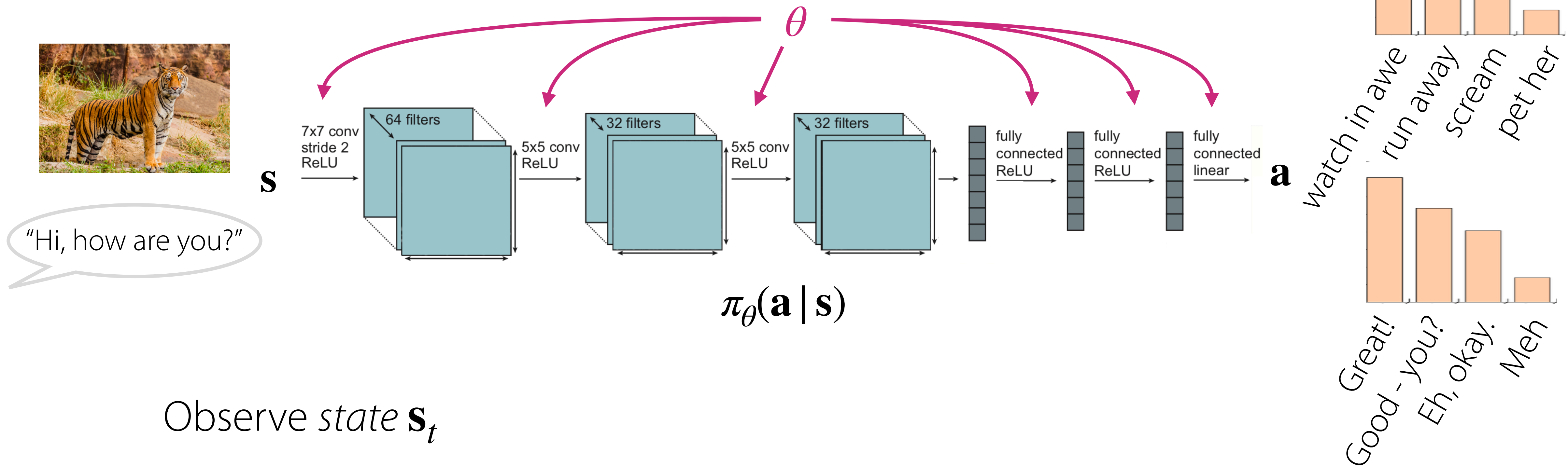
Meta-learning efficient exploration

Open challenges

Other frontiers of research

How to develop generalists?

A formalization of behavior



Observe state \mathbf{s}_t

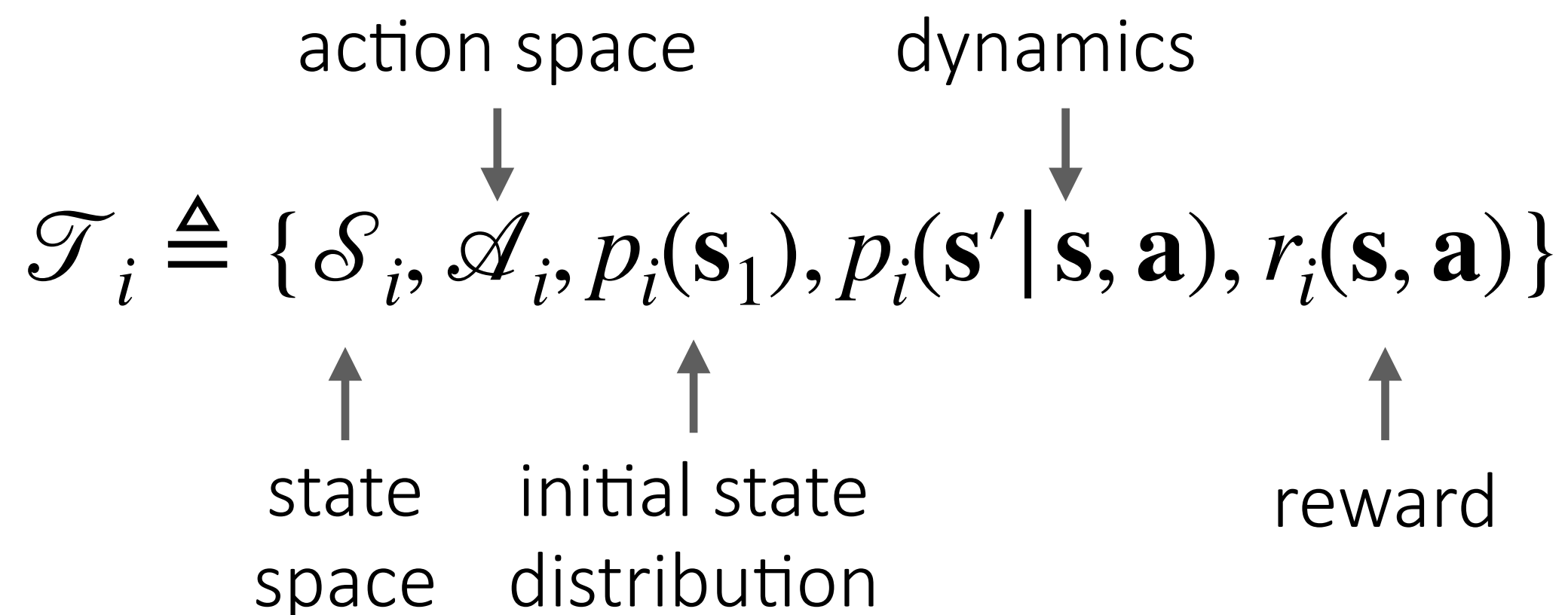
Take action \mathbf{a}_t (e.g. by sampling from policy $\pi_{\theta}(\cdot | \mathbf{s}_t)$)

Observe next state \mathbf{s}_{t+1} sampled from unknown world dynamics $p(\cdot | \mathbf{s}_t, \mathbf{a}_t)$

Result: a trajectory $\mathbf{s}_1, \mathbf{a}_1, \dots, \mathbf{s}_T$ also called a policy roll-out

Reinforcement learning is reward maximization

A reinforcement learning **task**:
(an MDP)



Meta-reinforcement learning = **meta-learning** with RL tasks

Rewards $r(\mathbf{s}, \mathbf{a})$: tell us which states & actions are better



high reward



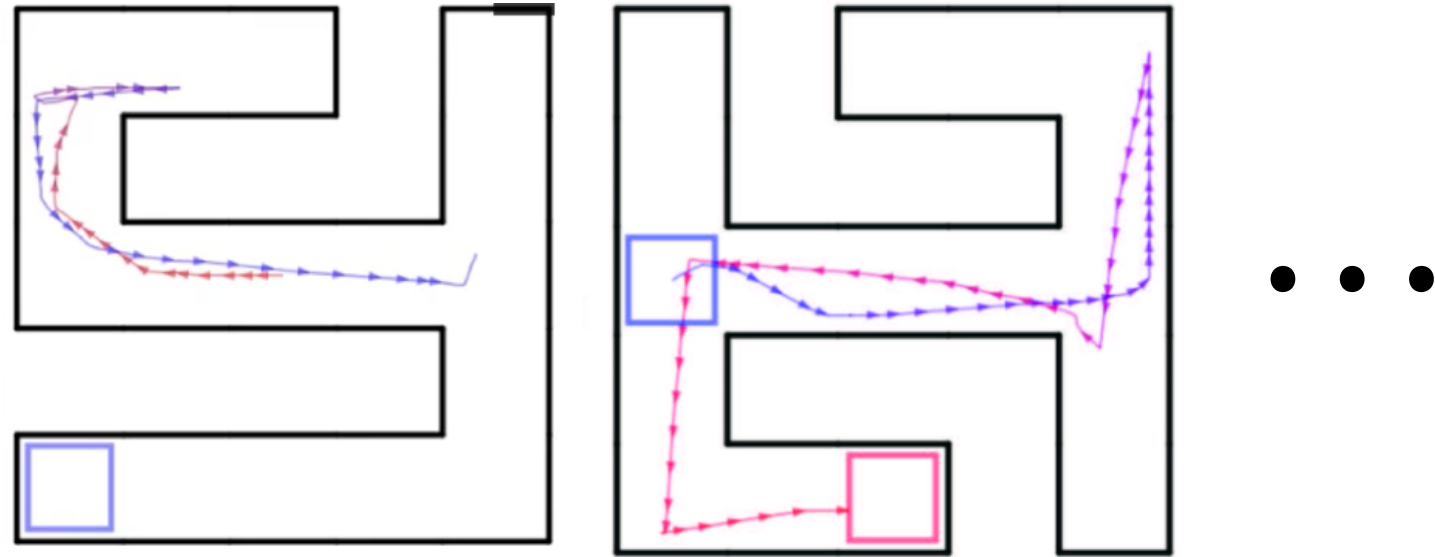
low reward

$$\text{Goal of RL: } \max_{\theta} \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{(\mathbf{s}_t, \mathbf{a}_t) \in \tau} r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

Actions affect future states (& thus future rewards)

Examples of meta-reinforcement learning problems

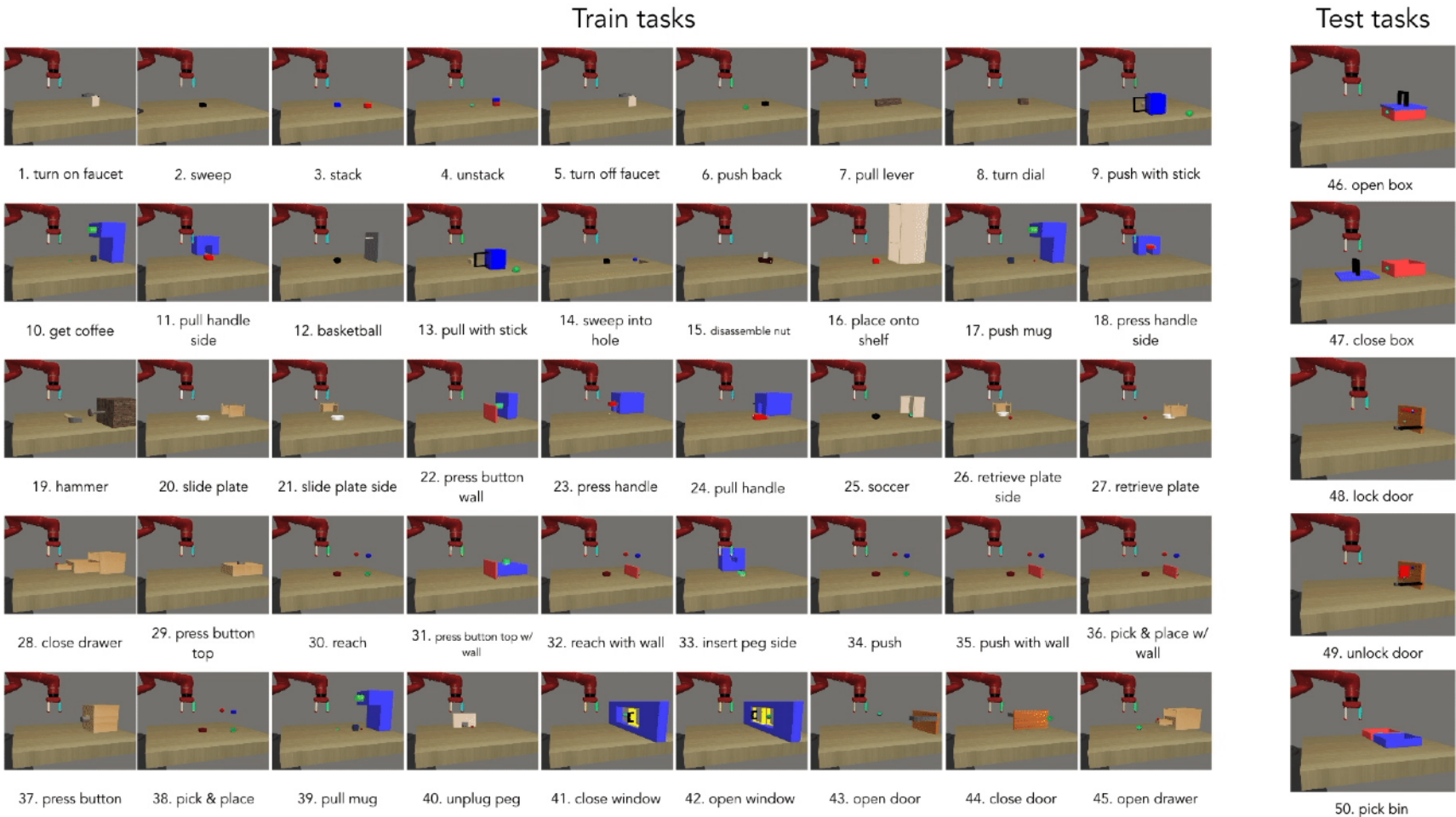
Navigation through different mazes



Locomotion on different terrains, slopes



Object manipulation with different objects, goals

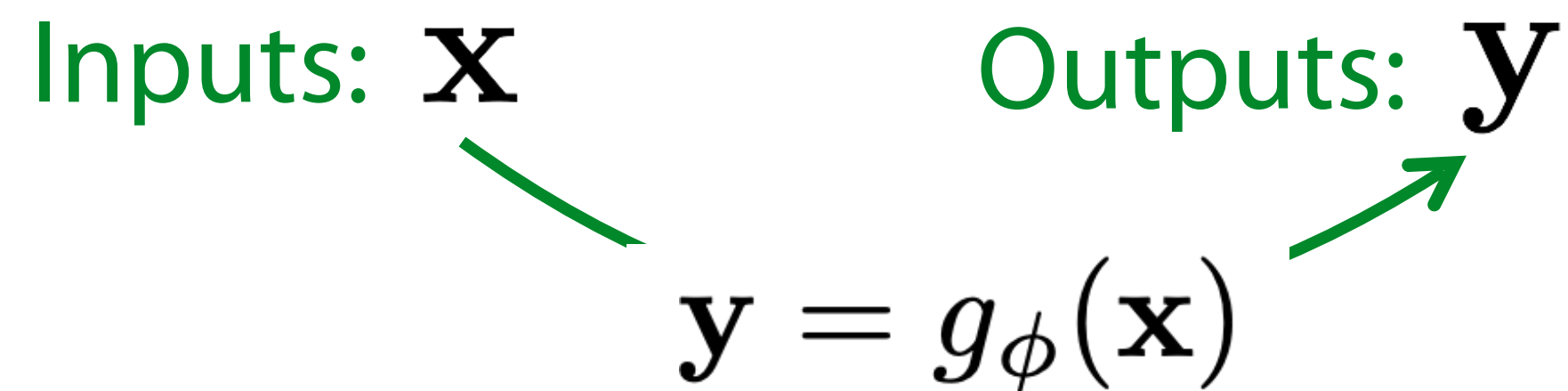


Dialog with different users w/ different preferences



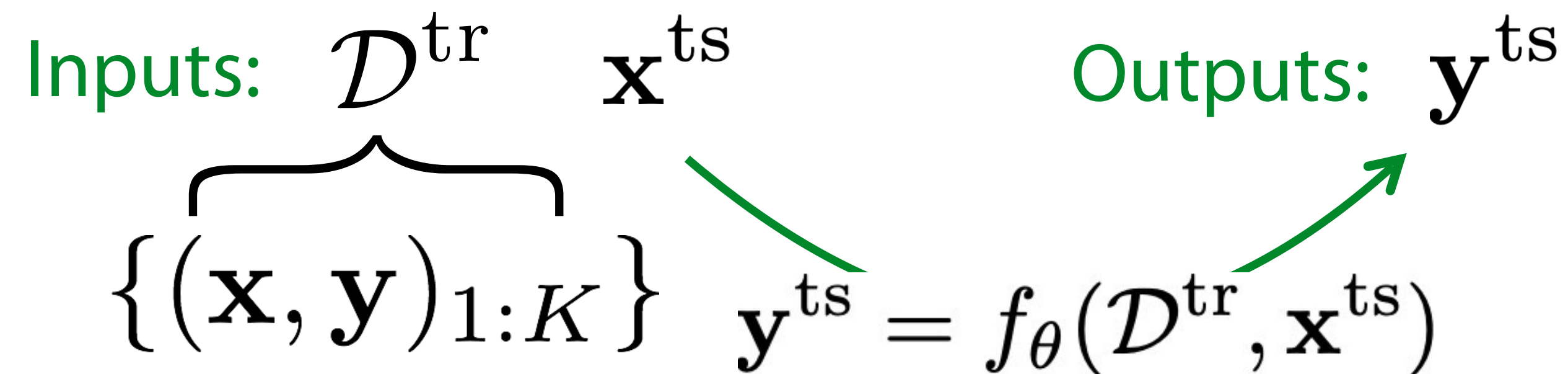
Recall: The Meta-Learning Problem

Supervised Learning:



Data: $\{(\mathbf{x}, \mathbf{y})_i\}$

Meta Supervised Learning:



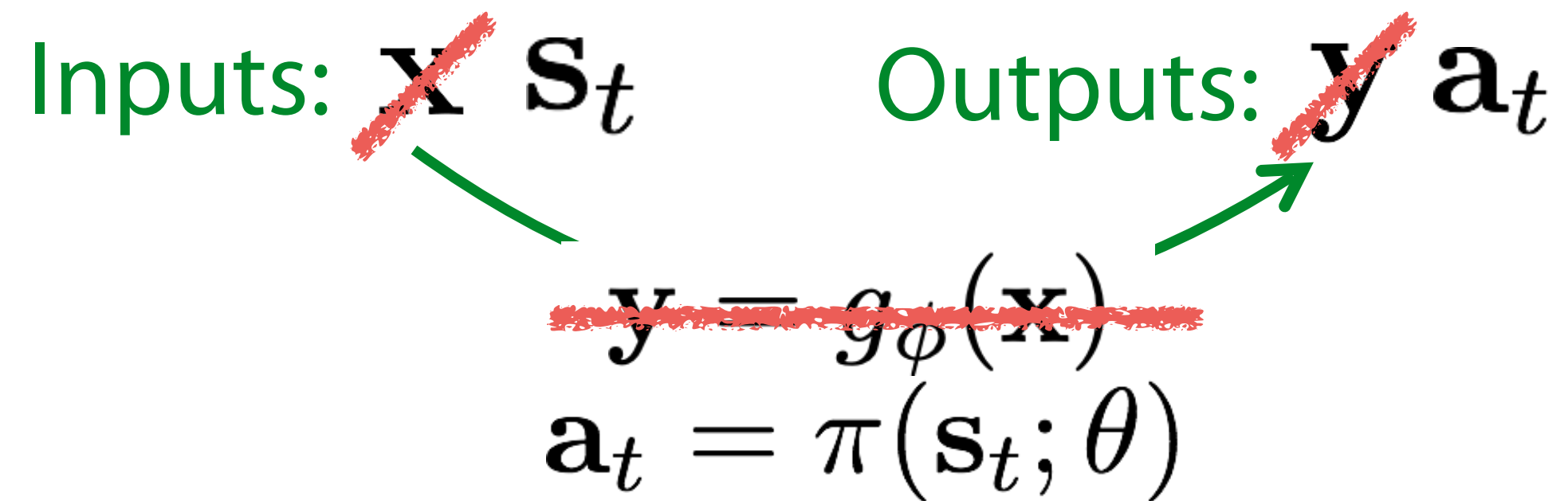
Data: $\{\mathcal{D}_i\}$
 $\mathcal{D}_i : \{(\mathbf{x}, \mathbf{y})_j\}$

Why is this view useful?

Reduces the meta-learning problem to the design & optimization of f .

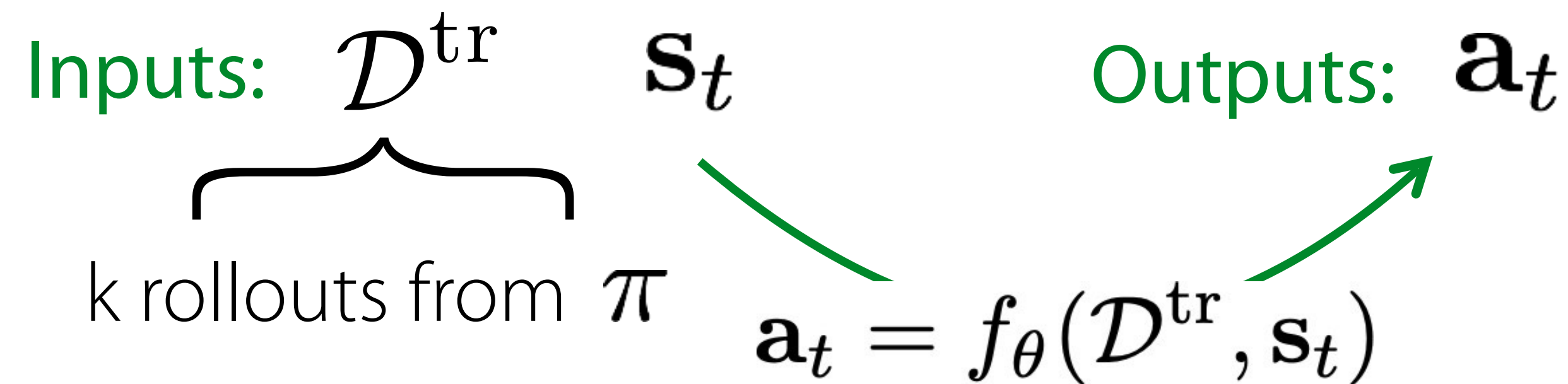
The Meta Reinforcement Learning Problem

Reinforcement Learning:



Data: ~~$\{(\mathbf{x}, \mathbf{y})_i\}$~~
 $\{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})\}$

Meta Reinforcement Learning:



Data: $\{\mathcal{D}_i\}$
dataset of datasets
collected for each task

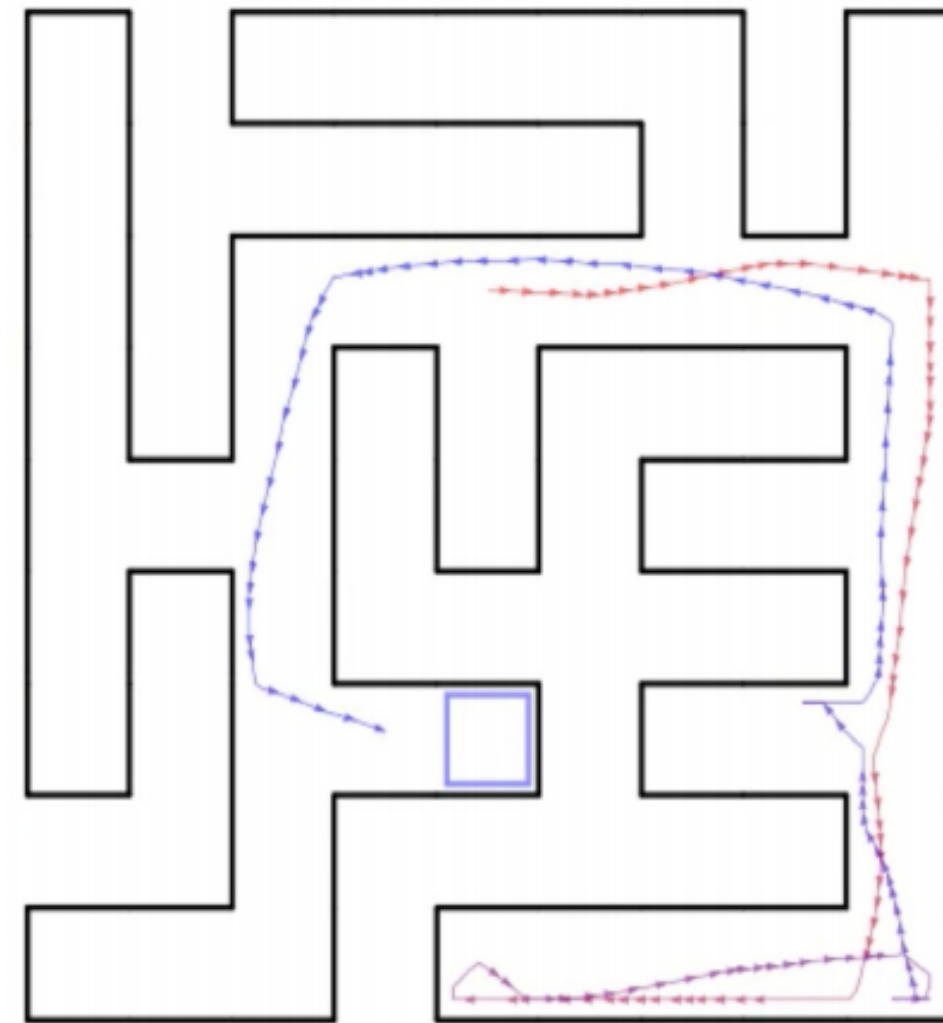
Design & optimization of f *and* collecting appropriate data
(learning to explore)

Meta-RL Example: Maze Navigation

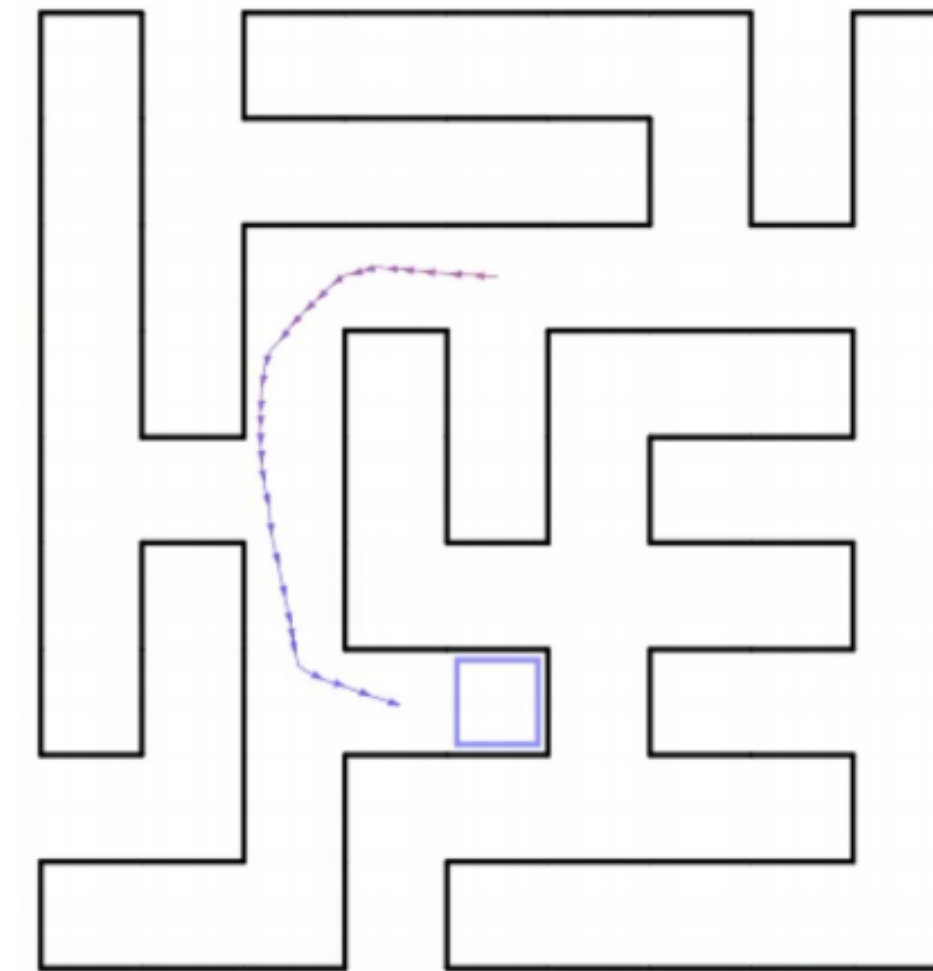
Given a small amount of experience

Learn to solve the task

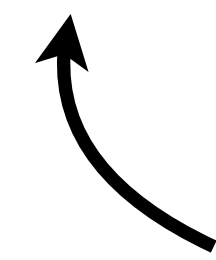
[meta] test time



$\mathcal{D}_{\text{train}}$



$\mathbf{s}_t \rightarrow \mathbf{a}_t$

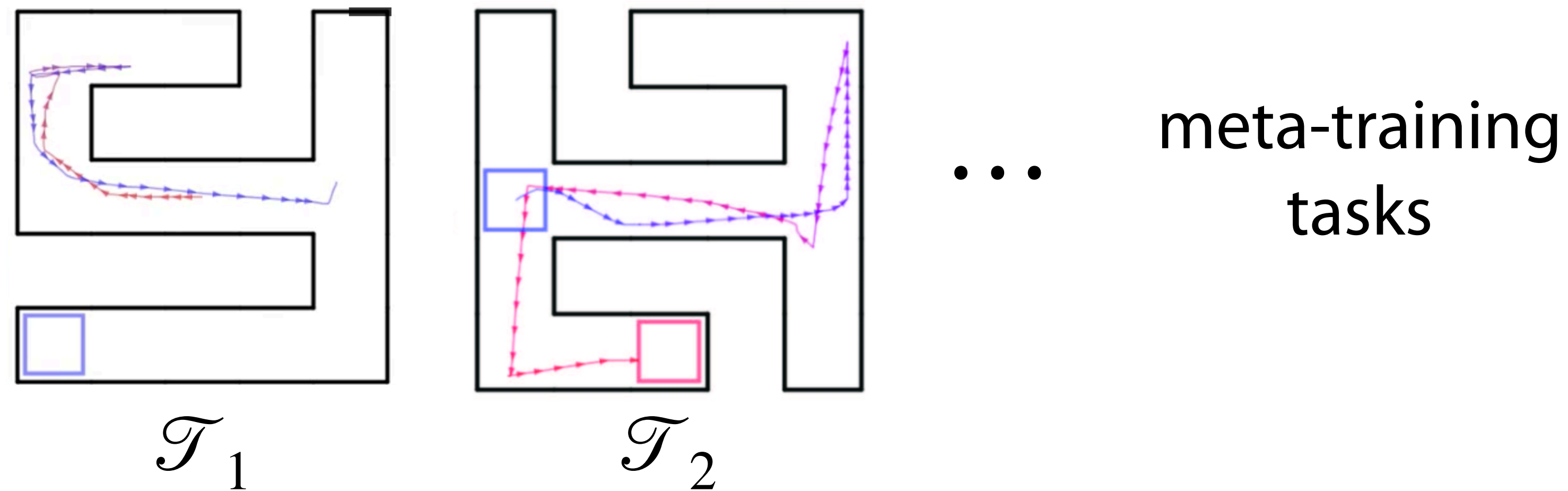


We need to figure out how to collect this experience too!

Meta-RL Example: Maze Navigation

By learning how to learn many other tasks:

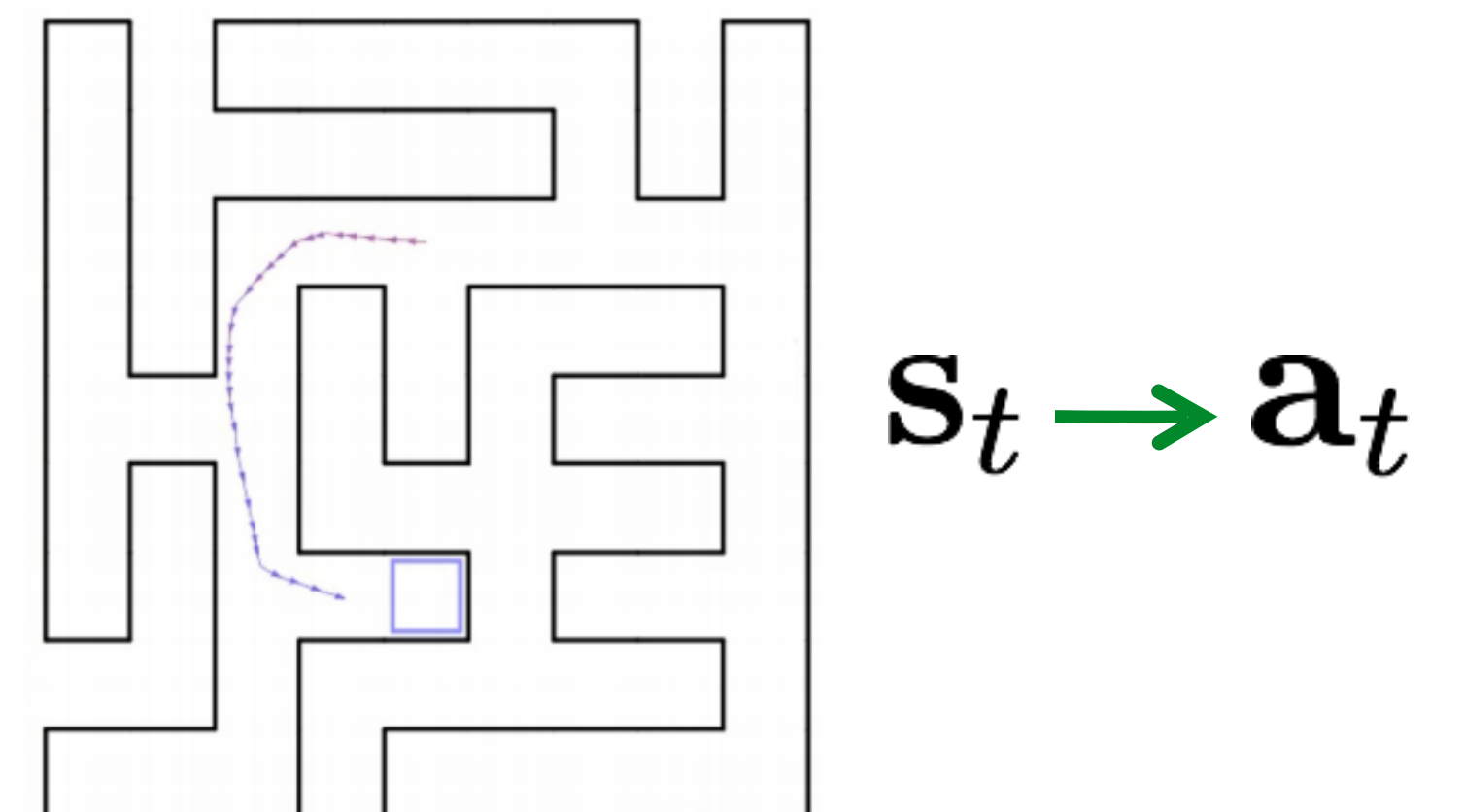
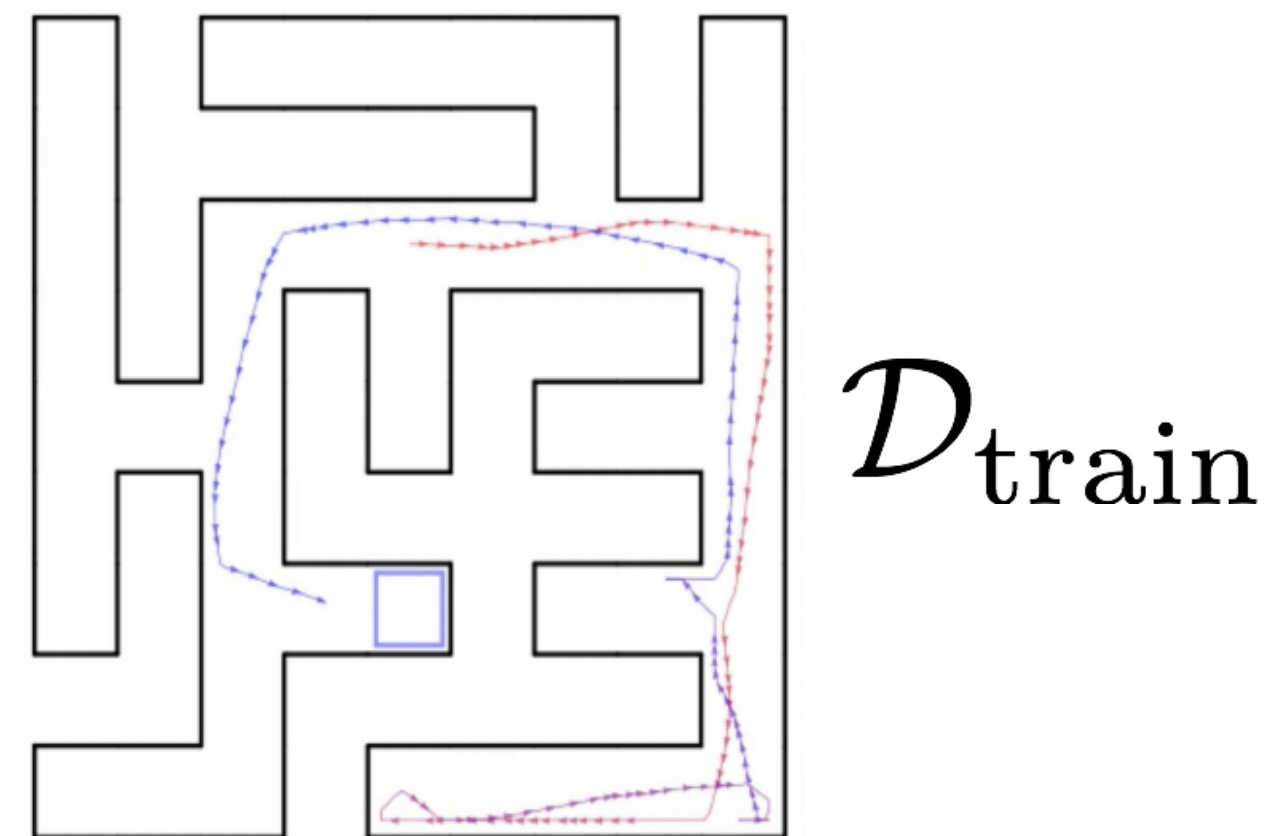
[meta] train time



Given a small amount of experience

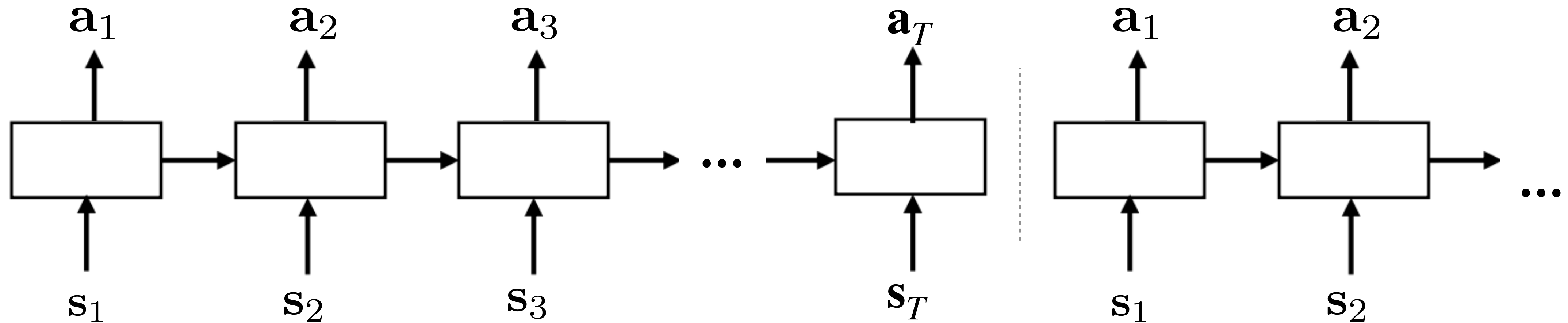
Learn to solve the task

[meta] test time



How to extend black-box meta-learning to meta-RL?

Policy with memory
(LSTM, Transformer, Conv, ...)

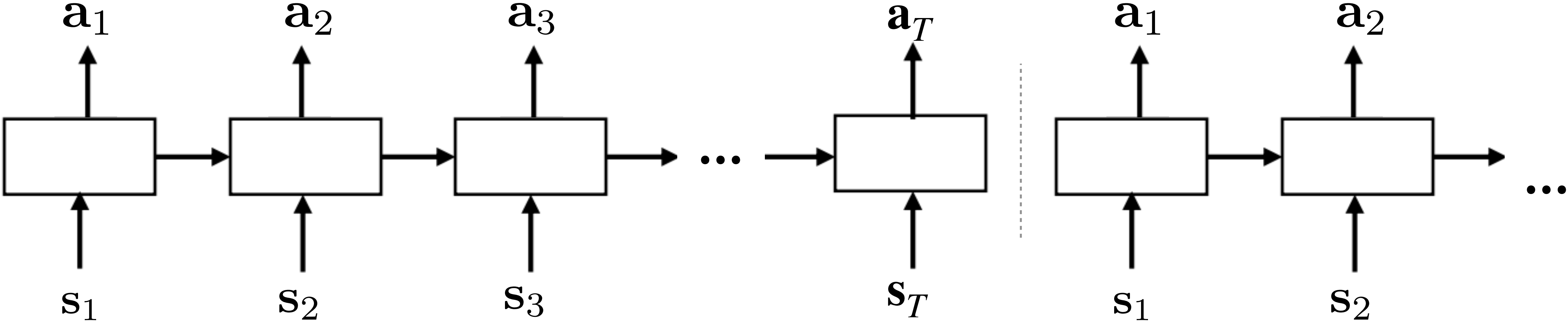


Train across tasks: $\mathcal{T}_i \triangleq \{ \mathcal{S}_i, \mathcal{A}_i, p_i(\mathbf{s}_1), p_i(\mathbf{s}' | \mathbf{s}, \mathbf{a}), r_i(\mathbf{s}, \mathbf{a}) \}$

Question: What is one change to make this suitable for meta-learning?

How to extend black-box meta-learning to meta-RL?

Policy with memory (LSTM, Transformer, Conv, ...)



Train across tasks: $\mathcal{T}_i \triangleq \{ \mathcal{S}_i, \mathcal{A}_i, p_i(s_1), p_i(s' | s, a), r_i(s, a) \}$

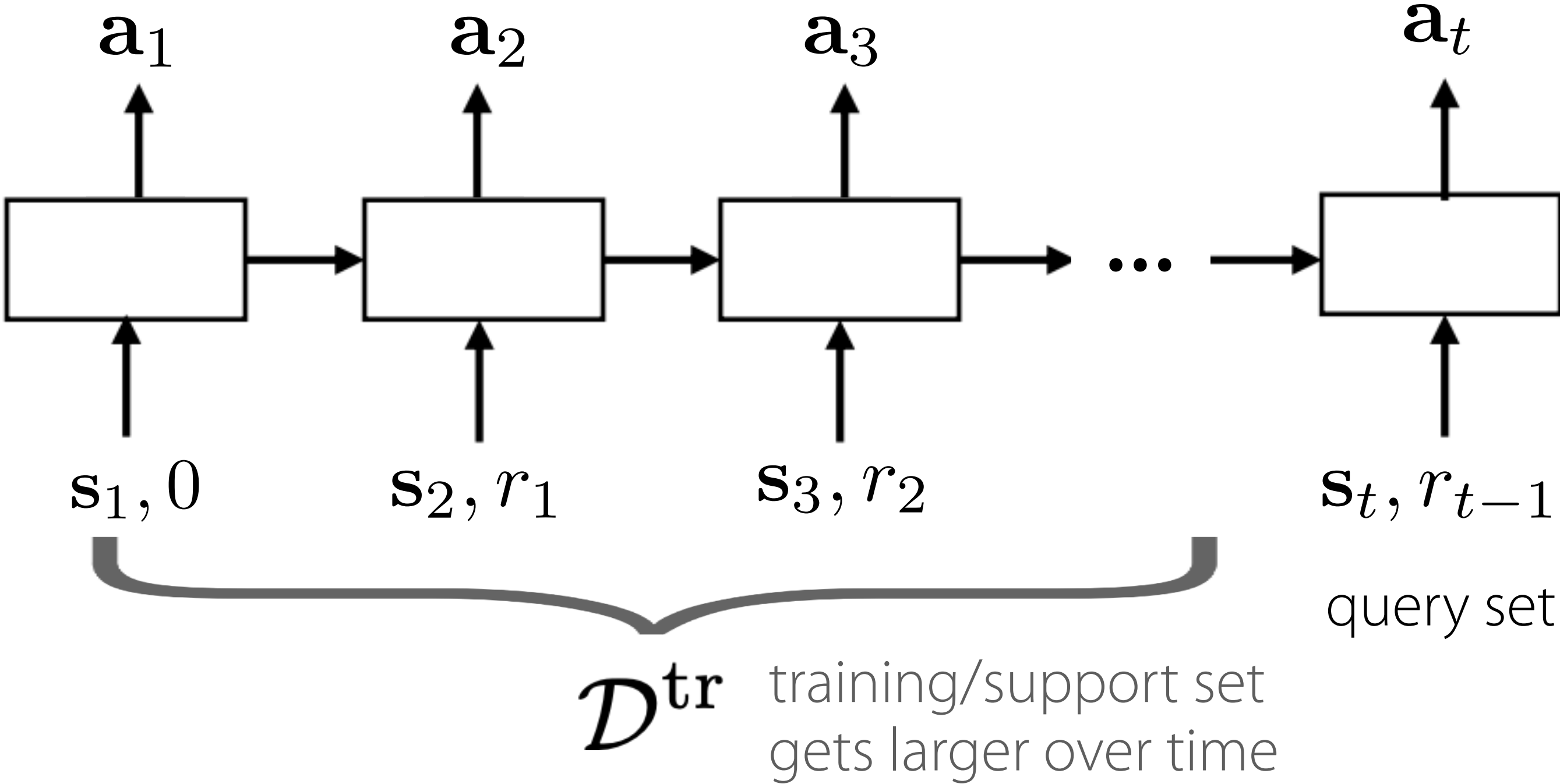
Question: What is one change to make this suitable for meta-learning?

Pass in reward as input Maintain hidden state **across episodes** within a task!

Black-Box Meta-RL: Overview

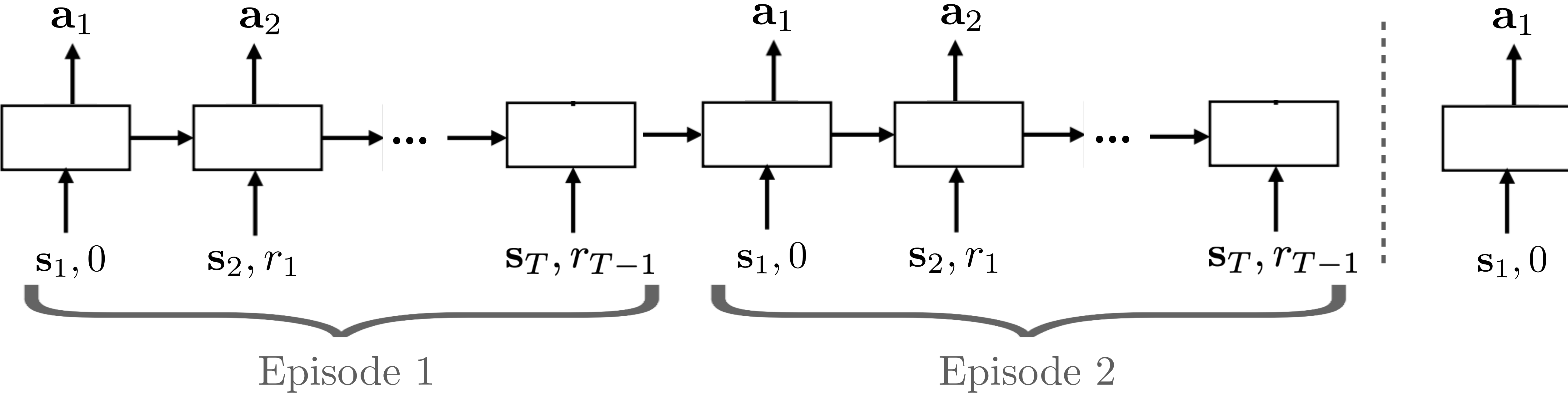
Black-box network
(LSTM, Transformer, Conv, ...)

$$\mathbf{a}_t = f_{\theta}(\mathcal{D}^{\text{tr}}, \mathbf{s}_t)$$



Question: Why don't we need to pass in the actions \mathbf{a}_{t-1} with the support set?


Black-Box Meta-RL: Algorithm



1. Sample task \mathcal{T}_i
2. Roll-out policy $\pi(a | s, \mathcal{D}_i^{\text{tr}})$ for N episodes (under dynamics $p_i(s' | s, a)$ and reward $r_i(s, a)$)
3. Store sequence in replay buffer for task \mathcal{T}_i .
4. Update policy to maximize discounted return for all tasks.

Black-Box Meta-RL: Algorithm

Meta-Training

1. Sample task \mathcal{T}_i
 2. Roll-out policy $\pi(a | s, \mathcal{D}_i^{\text{tr}})$ for N episodes (under dynamics $p_i(s' | s, a)$ and reward $r_i(s, a)$)
 3. Store sequence in replay buffer for task \mathcal{T}_i .
 4. Update policy to maximize discounted return for all tasks.
- 

Meta-Test Time

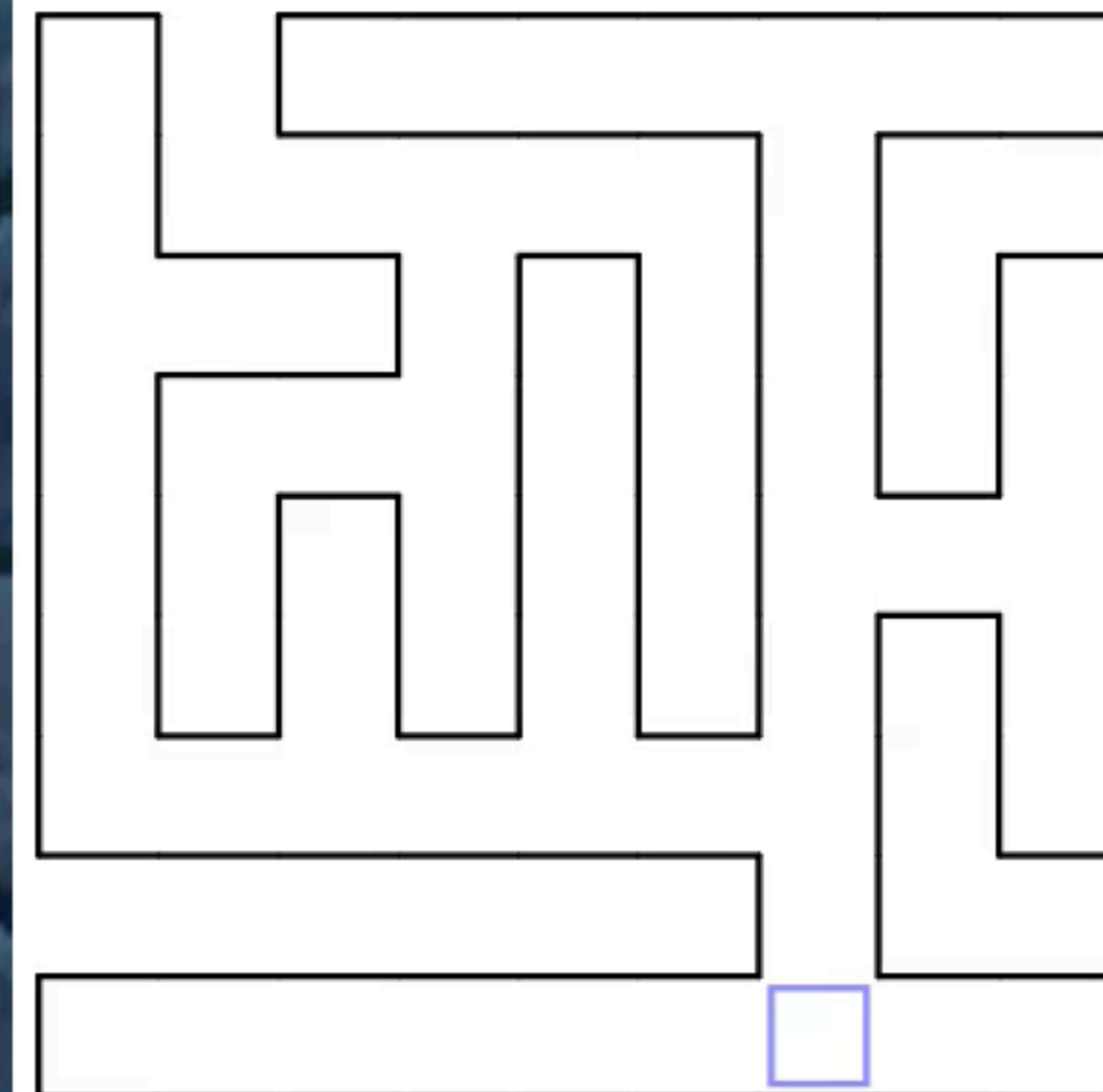
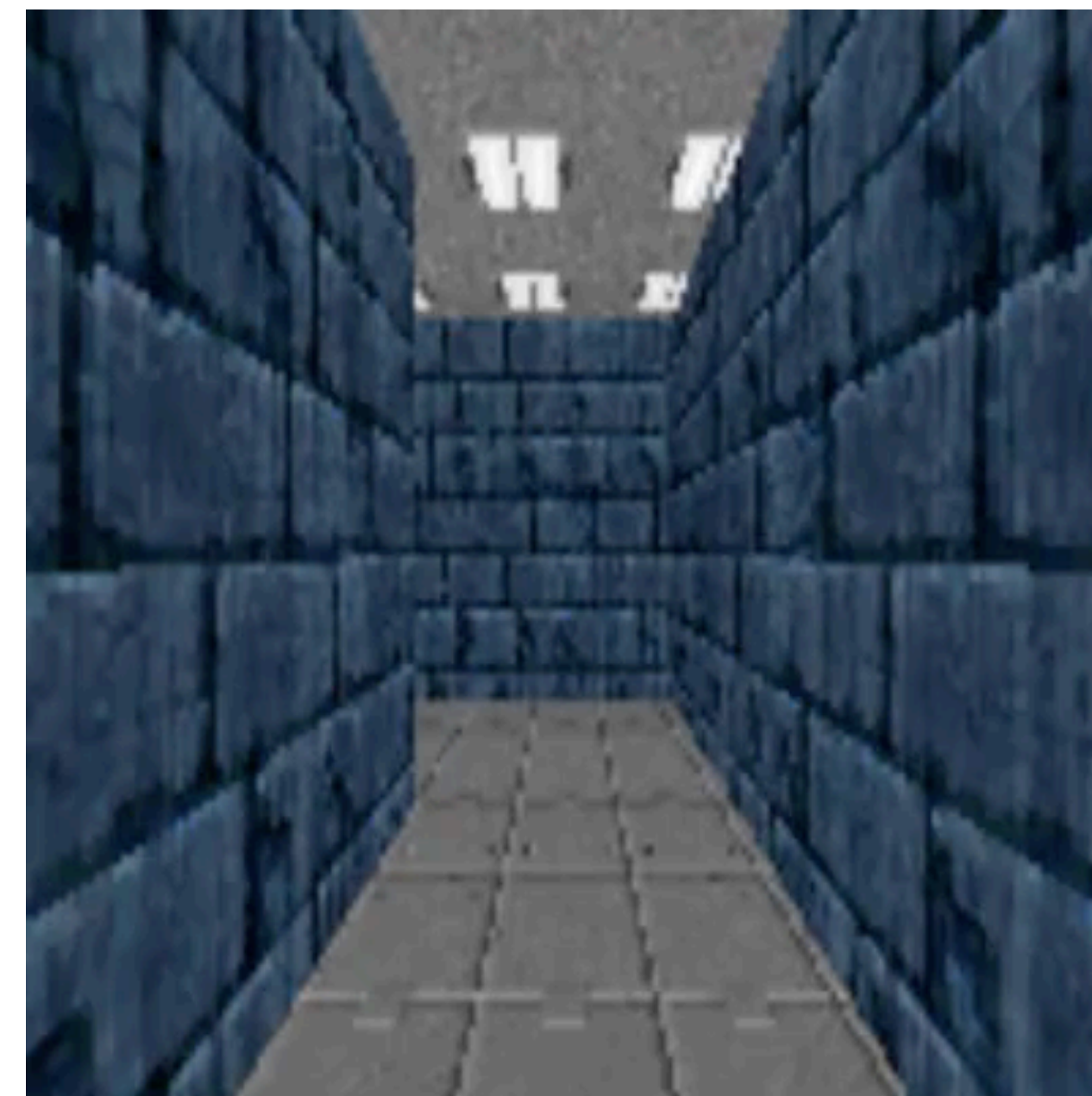
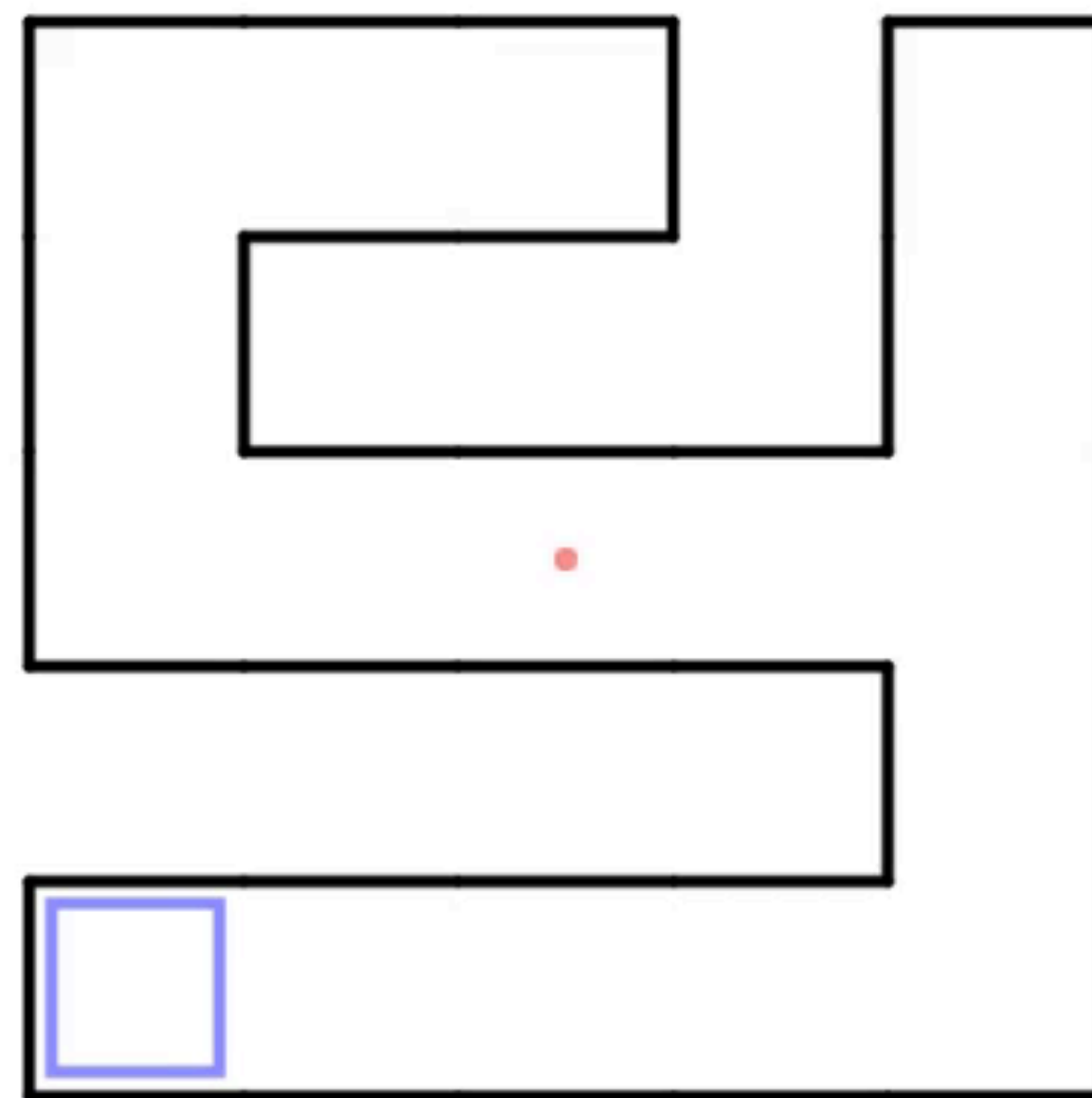
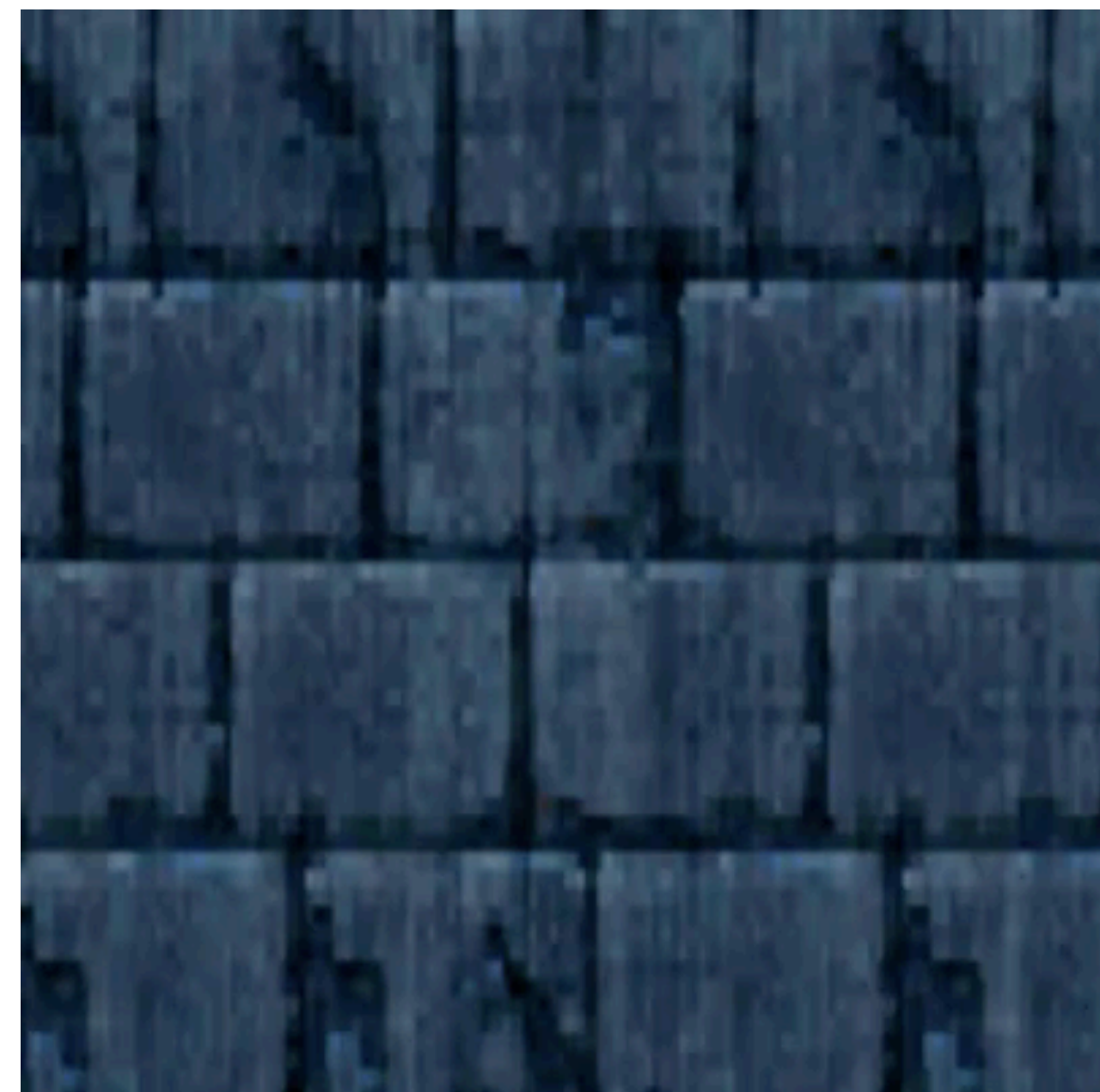
1. Sample *new* task \mathcal{T}_j
2. Roll-out policy $\pi(a | s, \mathcal{D}_j^{\text{tr}})$ for up to N episodes

Meta-RL Example

From: Mishra, Rohaninejad, Chen, Abbeel. *A Simple Neural Attentive Meta-Learner*. ICLR 2018

Experiment: Learning to visually navigate a maze

- train on 1000 small mazes
- test on held-out small mazes and large mazes



Meta-RL Example

From: Mishra, Rohaninejad, Chen, Abbeel. *A Simple Neural Attentive Meta-Learner*. ICLR 2018

Experiment: Learning to visually navigate a maze

- train on 1000 small mazes
- test on held-out small mazes and large mazes

Method	Small Maze		Large Maze	
	Episode 1	Episode 2	Episode 1	Episode 2
Random	188.6 \pm 3.5	187.7 \pm 3.5	420.2 \pm 1.2	420.8 \pm 1.2
LSTM	52.4 \pm 1.3	39.1 \pm 0.9	180.1 \pm 6.0	150.6 \pm 5.9
SNAIL (ours)	50.3 \pm 0.3	34.8 \pm 0.2	140.5 \pm 4.2	105.9 \pm 2.4

Table 5: Average time to find the goal on each episode

Plan for Today

Meta reinforcement learning

The meta-RL problem set-up

Black-box meta-RL

Meta-learning efficient exploration

Open challenges

Other frontiers of research

How to develop more generalists?

How Do We Learn to Explore?

Solution #1: Optimize for Exploration & Exploitation *End-to-End* w.r.t. Reward

(Duan et al., 2016, Wang et al., 2016, Mishra et al., 2017, Stadie et al., 2018, Zintgraf et al., 2019, Kamienny et al., 2020)

- + simple
- + leads to optimal strategy in principle
- challenging optimization when exploration is hard

A simple, running example

Hallway 1

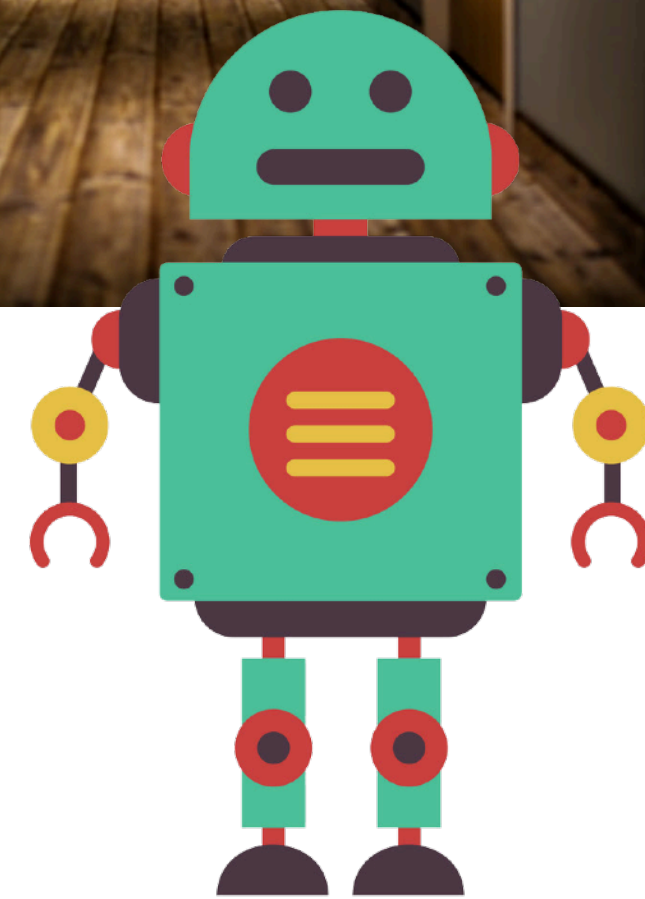


Hallway 2



...

Hallway N



agent



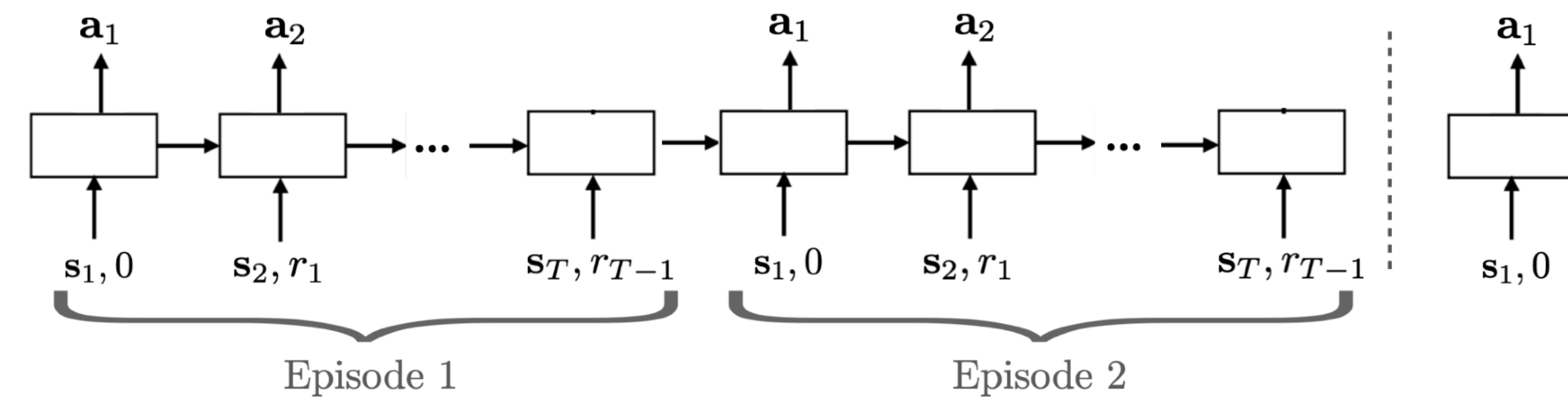
information on
where to go

Different tasks: navigating to
the ends of different hallways

How Do We Learn to Explore?

Solution #1: Optimize for Exploration & Exploitation *End-to-End* w.r.t. Task Reward

(Duan et al., 2016, Wang et al., 2016, Mishra et al., 2017, Stadie et al., 2018, Zintgraf et al., 2019, Kamienny et al., 2020)



Example episodes during meta-training:

agent goes to the end of the correct hallway

agent goes to wrong hallway then correct hallway

agent looks at the instructions

- gets positive reward for current task,
but $\mathcal{D}_i^{\text{tr}}$ won't be different than for any other task

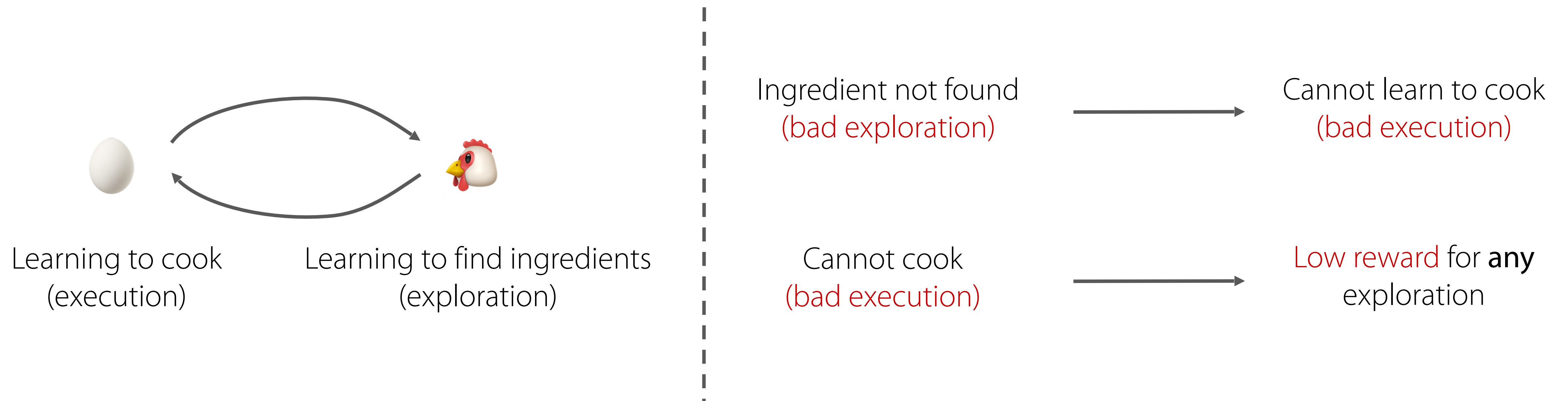
+/- provides signal on a **suboptimal**
exploration + exploitation strategy

- good exploratory behavior, but won't
get any reward for this behavior

It's hard to learn exploration & exploitation at the same time!

Why is End-to-End Training Hard in This Example?

End-to-end approach: optimize exploration and execution episode behaviors end-to-end to maximize reward of execution



Coupling problem: learning exploration and execution depend on each other

—> can lead to poor local optima, poor sample efficiency

Solution #2

Idea 2.0: Label each training task with a unique ID μ

Meta training

Exploration policy: train policy $\pi^{\text{exp}}(\mathbf{a} | \mathbf{s})$ and task identification model $q(\mu | \mathcal{D}_{\text{tr}})$
such that $\mathcal{D}_{\text{tr}} \sim \pi^{\text{exp}}$ allows accurate task prediction from f

Execution policy: train ID-conditioned policy $\pi^{\text{exec}}(\mathbf{a} | \mathbf{s}, \mu_i)$

Meta testing

Explore: $\mathcal{D}_{\text{tr}} \sim \pi^{\text{exp}}(\mathbf{a} | \mathbf{s})$ Infer task: $\hat{\mu} \sim q(\mu | \mathcal{D}_{\text{tr}})$ Perform task: $\pi^{\text{exec}}(\mathbf{a} | \mathbf{s}, \hat{\mu})$

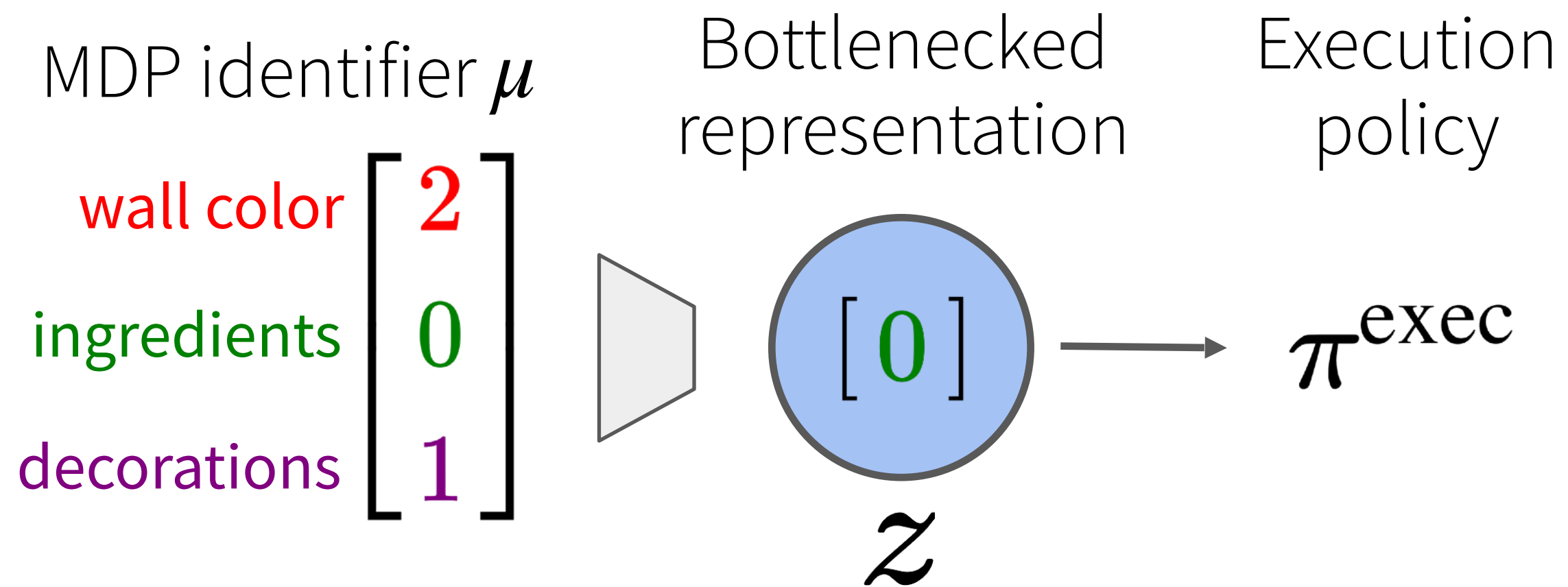
+ decoupled exploration and exploitation

— may not generalize well for one-hot μ

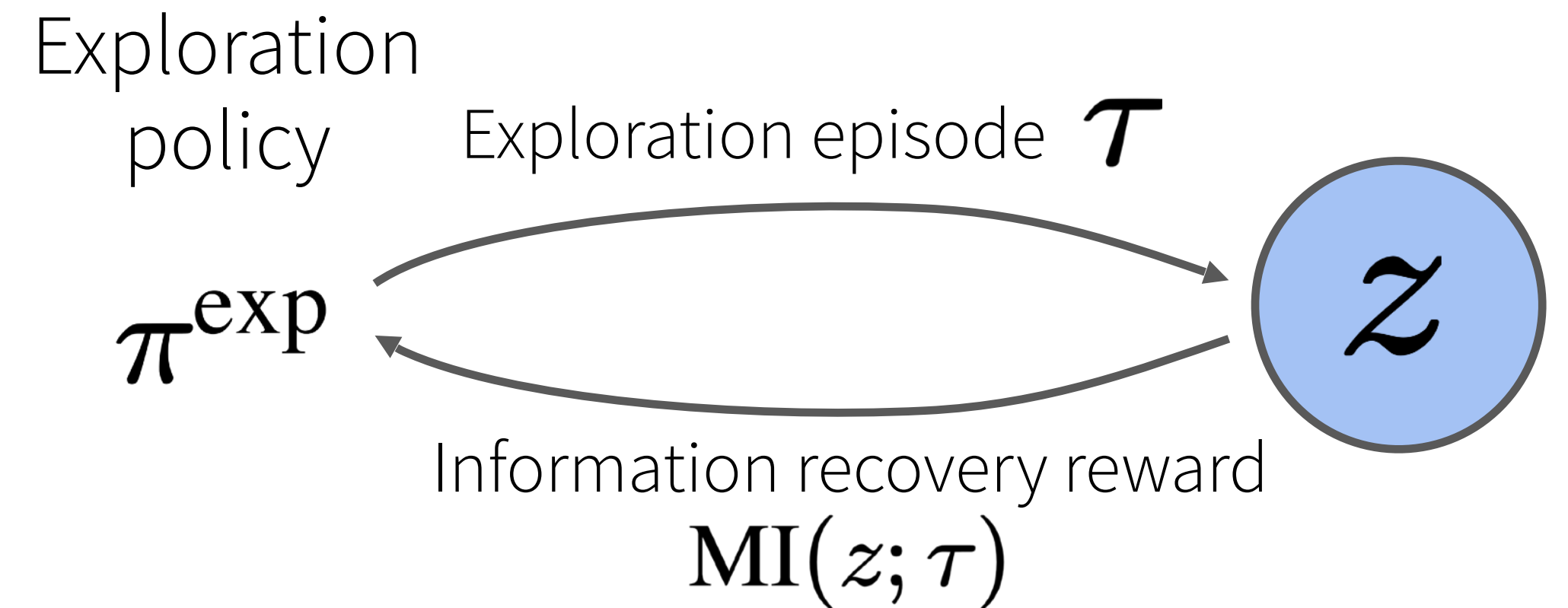
Solution #3: **Decouple** by acquiring representation of task relevant information

Meta-training

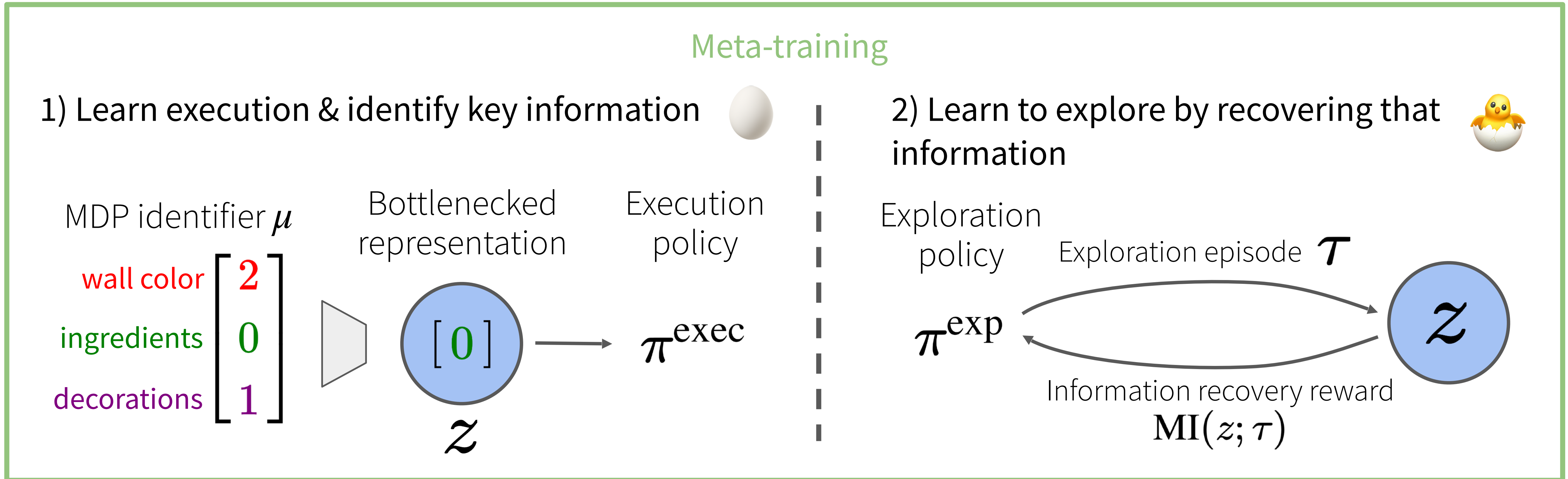
1) Learn execution & identify key information 



2) Learn to explore by recovering that information 



Solution #3: **Decouple** by acquiring representation of task relevant information



Train $\pi^{\text{exec}}(\mathbf{a} | \mathbf{s}, z_i)$ and encoder $F(z_i | \mu_i)$ to:

$$\max \sum_i \mathbb{E}_{\pi^{\text{exec}}} [r_i] - D_{\text{KL}} (F(z_i | \mu_i) || \mathcal{N}(0, 1))$$

Train π^{exp} such that collected \mathcal{D}_{tr} is predictive of z_i .

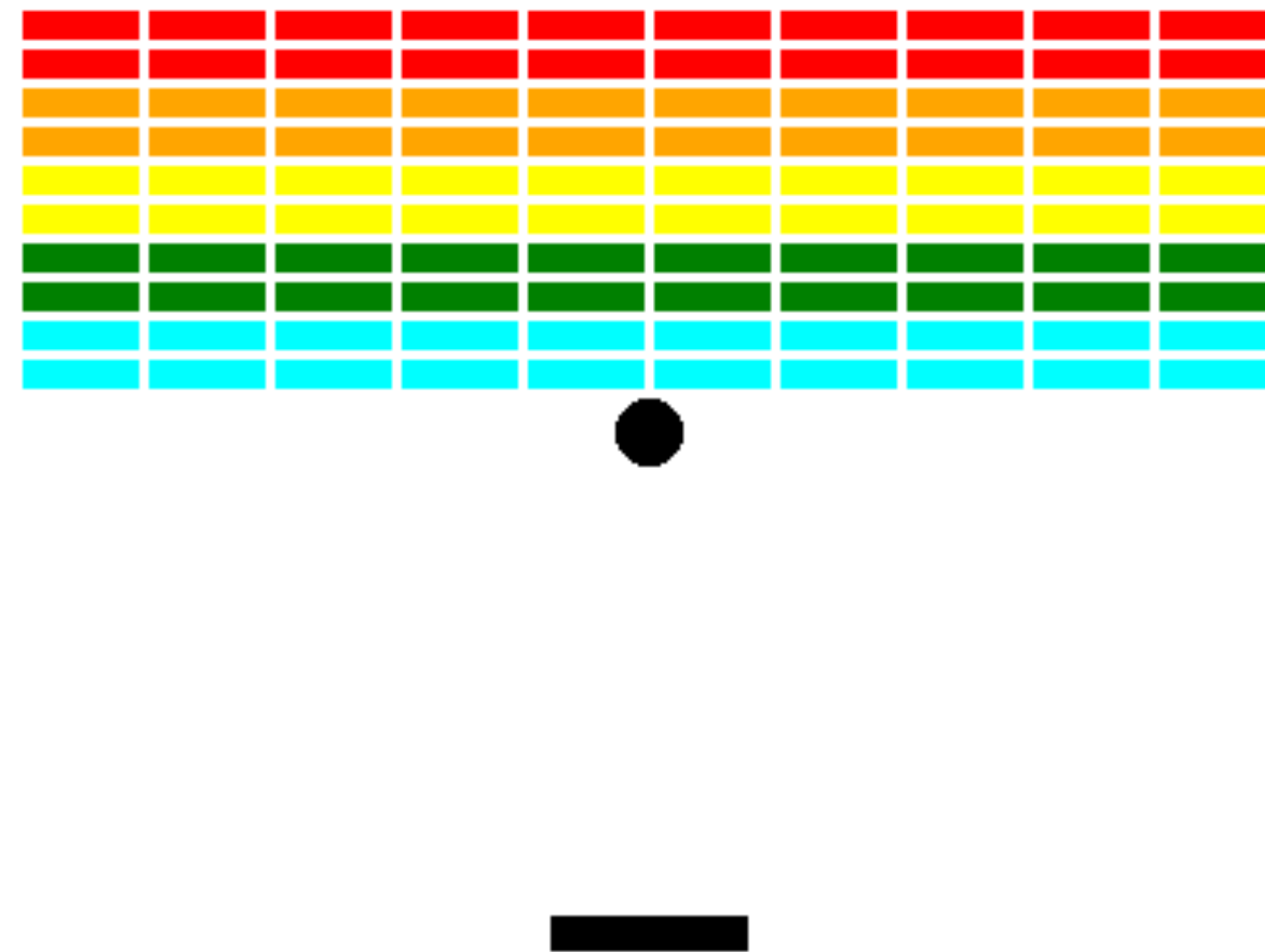
In practice: (1) and (2) can be trained simultaneously.

Example application: finding bugs & providing feedback in student programs

Bounce programming assignment
(Code.org)

```
Underlying env ID: 7340
Env ID: 1
Label: [1 1 0 0 0 0 1 0 0 1 0 1 0 1 1]
Binary label: whenGoal-noBallLaunch
Action: None
Reward: 0
Timestep: 0
Exploration reward: 0.020
Prob: 0.456
```

Breakout assignment
(CS106A)



Time-consuming for instructors/TAs to give feedback, grades.
Use meta-RL to learn exploration!

Experiments: Learned Exploration Behavior on Bounce

```
Underlying env ID: 7340  
Env ID: 1  
Label: [1 1 0 0 0 0 1 0 0 1 0 1 0 1 1]  
Binary label: whenGoal-noBallLaunch  
Action: None  
Reward: 0  
Timestep: 0  
Exploration reward: 0.020  
Prob: 0.456
```

```
Underlying env ID: 4843  
Env ID: 0  
Label: [0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]  
Binary label: whenMiss-noBallLaunch  
Action: None  
Reward: 0  
Timestep: 0  
Exploration reward: 0.005  
Prob: 0.507
```

```
Underlying env ID: 2732  
Env ID: 1  
Label: [0 1 0 1 1 0 0 1 0 0 1 1 0 0 1]  
Binary label: whenWall-illegal-moveRight  
Action: None  
Reward: 0  
Timestep: 0  
Exploration reward: 0.079  
Prob: 0.331
```

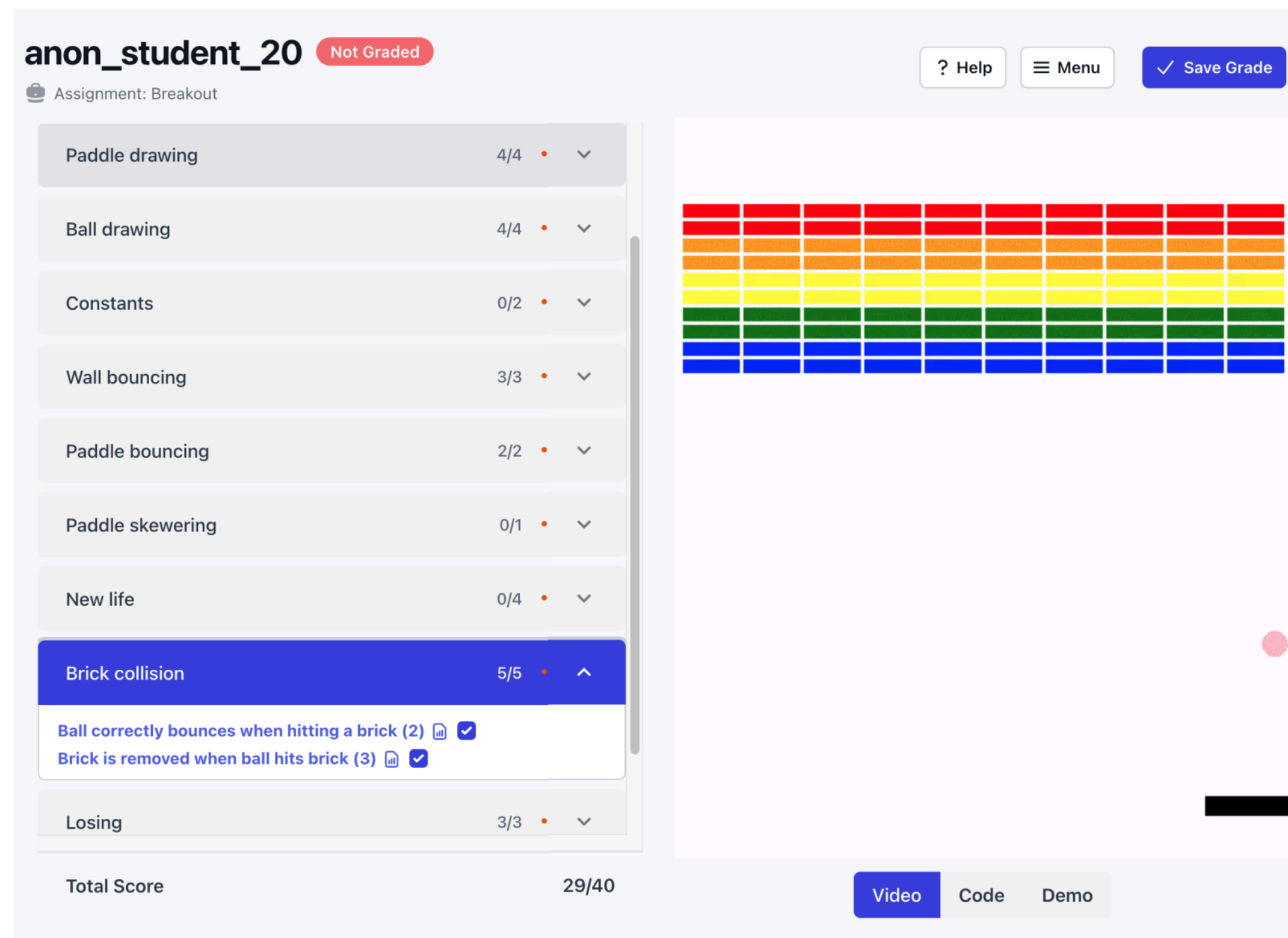
What happens when...

the ball hits the goal?

the ball hits the floor?

the ball hits the wall?

Experiments: AI-Assisted Grading in CS106A (Spring 2023)



Autograder prepopulates rubric & shows videos.

Leads to 44% faster & 6% more accurate grading.

Grading Scheme	Human Grading Time	Grading Accuracy
Manual	8 min 35s ± 6 min 47s	86.4% ± 8.9%
Autograder with human	4 min 49s ± 2 min 5s	92.3% ± 7.6%
Autograder only	—	90.1% ± 11.0%

Stanford TAs like using it.

Likert Scale (Strongly Disagree = 1, Disagree = 2, Neutral = 3, Agree = 4, Strongly Agree = 5)	
Statement	Avg. Score
Using the autograder is easier than manually grading.	4.5
Using the autograder is faster than manually grading.	4.5
Using the autograder is more accurate than manually grading.	3.9
The autograder's grades were useful to me.	4.4
I enjoyed using the autograder.	4.6
Net Promoter Score (0 - 10 inclusive)	
How much would you recommend using the autograder over manually grading in the future?	9.0

How Do We Learn to Explore?

End-to-End

- + leads to optimal strategy in principle
- challenging optimization when exploration is hard

Decoupled Exploration & Execution

- + leads to optimal strategy in principle
- + easy to optimize in practice
- requires task identifier

Plan for Today

Meta reinforcement learning

The meta-RL problem set-up

Black-box meta-RL

Meta-learning efficient exploration

Open challenges

Other frontiers of research

How to develop more generalists?

Putting Some of the Pieces Together

Multi-Task Learning

Learning many tasks in one model

Transfer Learning

Only care about one target task

Meta Learning

Transfer from multiple tasks to a new one

Optimize for transfer to new tasks

Lifelong Learning

Apply these ideas to a sequence of tasks

Supervised tasks:

Vary in terms of $p(x)$, $p(y|x)$, L

Domains (special case)

Vary only in terms of $p(x)$

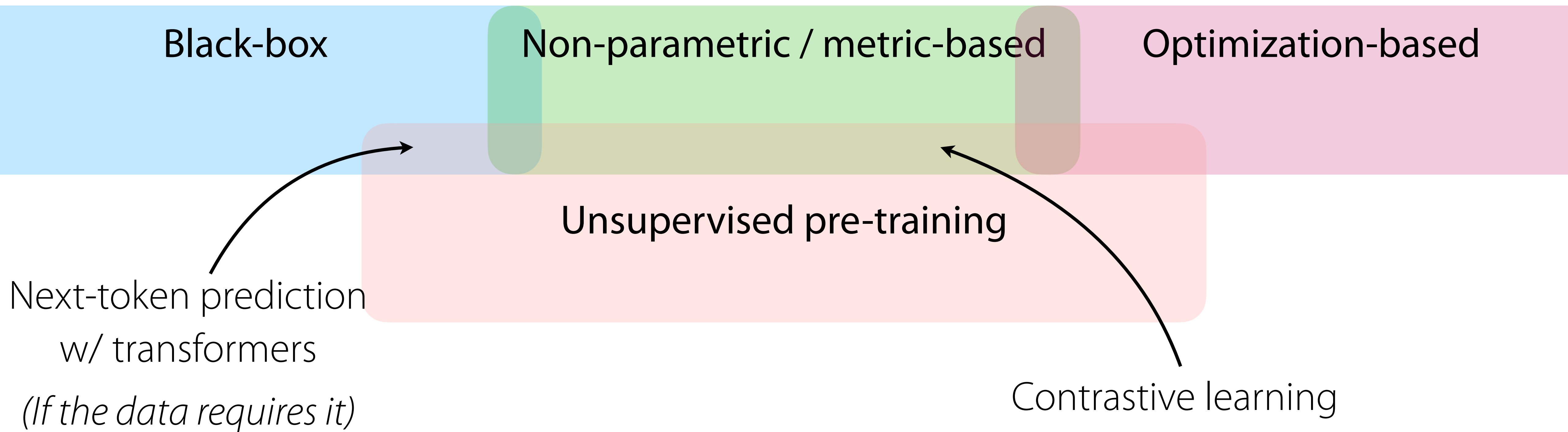
RL tasks:

Vary in terms of S , A , $p(s'|s,a)$, $r(s,a)$

Putting Some of the Pieces Together

Meta Learning

Transfer from multiple tasks to a new one
Optimize for transfer to new tasks



Open Challenges in Multi-Task and Meta Learning

(that we haven't previously covered)

Open Challenges in Multi-Task and Meta Learning

Improving scalability

- How best to use meta-learning algorithms in conjunction with foundation models?

Can we combine meta-learning with modern foundation models?

Some models are already a form of black-box meta-learning (enables in-context learning)

Can we improve the adaptability of foundation models?

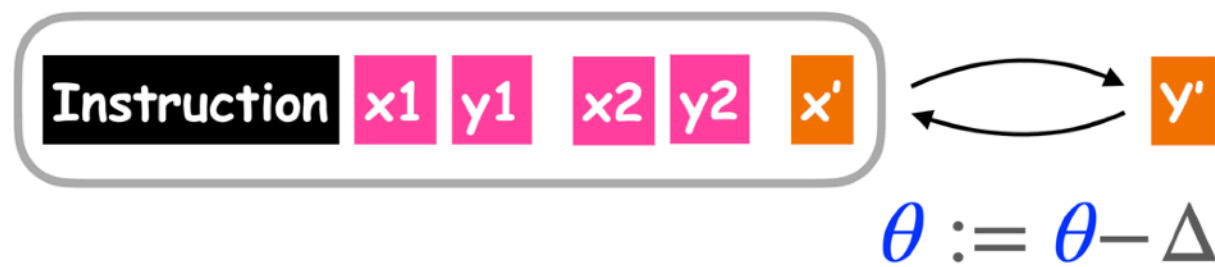
Enable LLM editing

Improve in-context learning

Instruction: "Is the comment positive?"

x1: "Good movie!" y1: "yes"

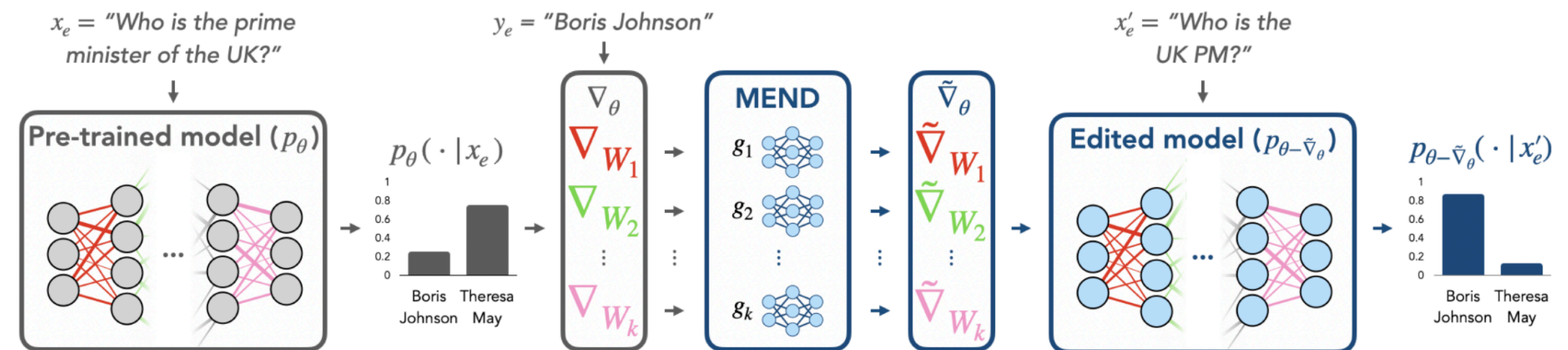
x2: "Bad movie!" y2: "no"



Few-shot Adaptation via In-context Learning

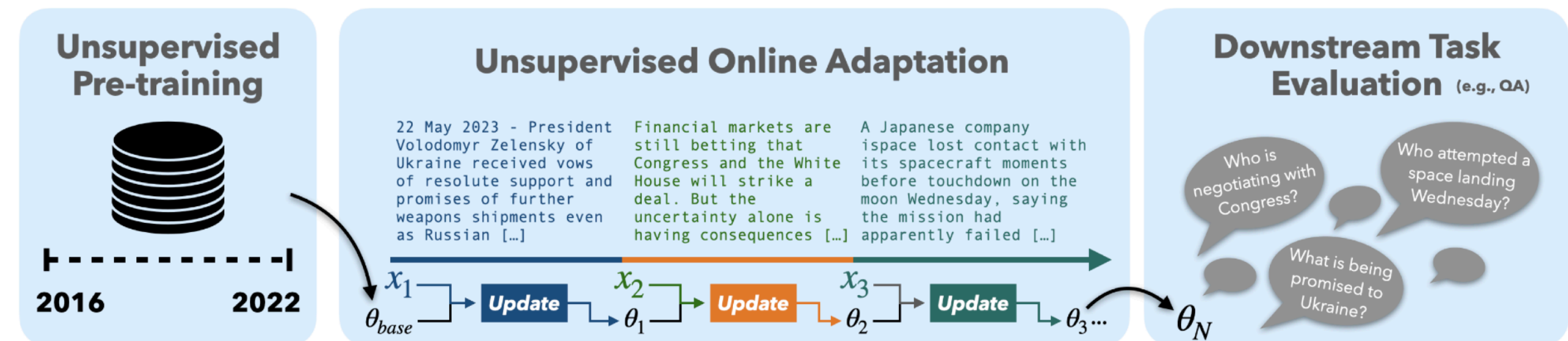
Meta-Update via Gradient Descent

Chen, Zhong, Zha, Karypis, He. ACL '22



Mitchell, Lin, Bosselut, Finn, Manning. ICLR '22 Tan, Zhang, Fu. '23

More effective fine-tuning



Hu, Mitchell, Manning, Finn. EMNLP '23

Open Challenges in Multi-Task and Meta Learning

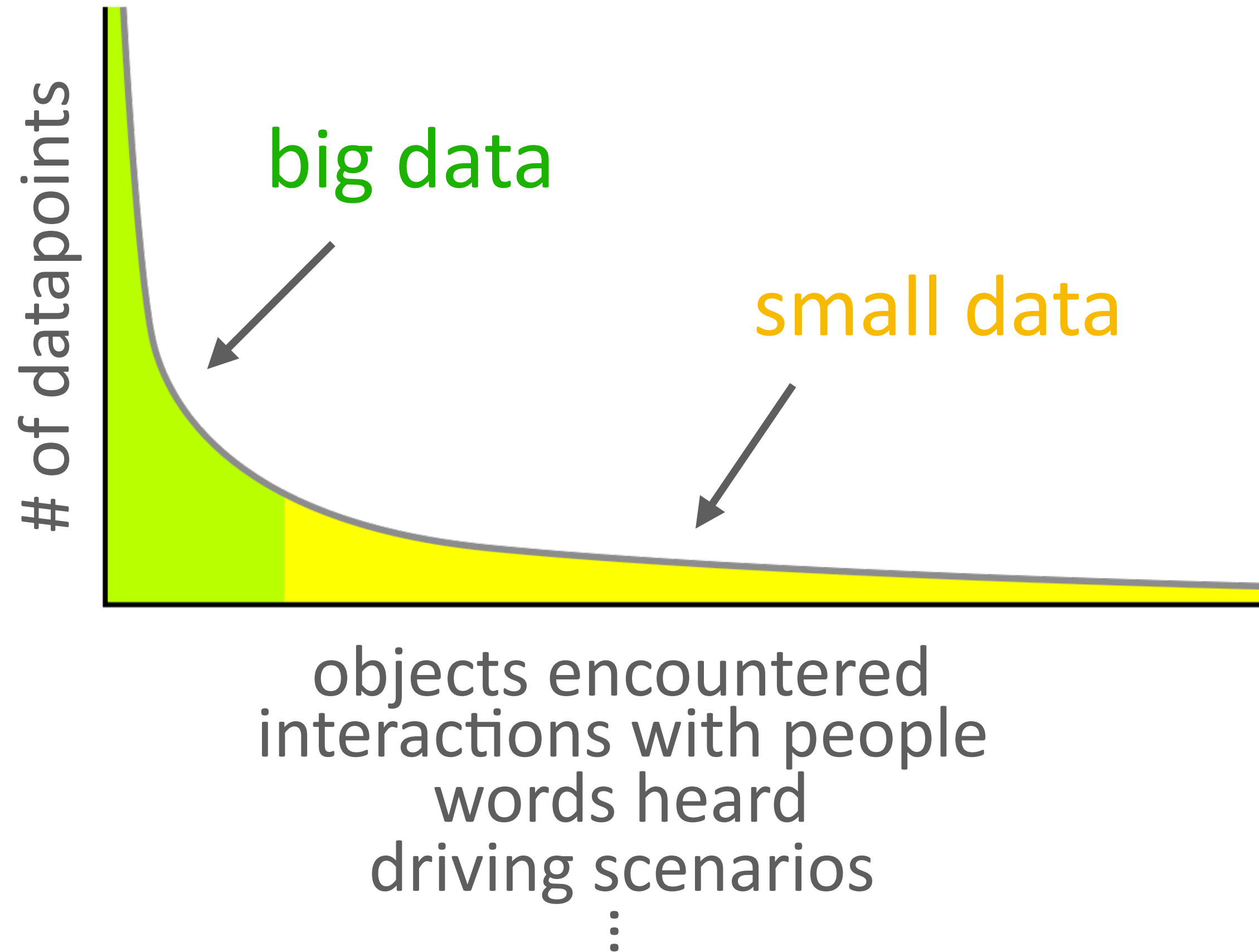
Improving scalability

- How best to use meta-learning algorithms in conjunction with foundation models?
- Can we make large-scale bi-level optimization more practical?

Addressing problem assumptions

- Generalization: Out-of-distribution tasks, long-tailed task distributions

The challenge of long-tailed distributions.



Few-shot generalization to the tail:

- prototypical clustering networks for dermatological diseases (Prabhu et al. 2018)
- adaptive risk minimization (Zhang et al. 2021)

Further hints might come from domain adaptation, robustness literature.

We learned how to do few-shot learning

...but these few-shot tasks may be from
a **different distribution**

Open Challenges in Multi-Task and Meta Learning

Improving scalability

- How best to use meta-learning algorithms in conjunction with foundation models?
- Can we make large-scale bi-level optimization more practical?

Addressing problem assumptions

- Generalization: Out-of-distribution tasks, long-tailed task distributions
- Multimodality: Can you learn priors from multiple modalities of data?

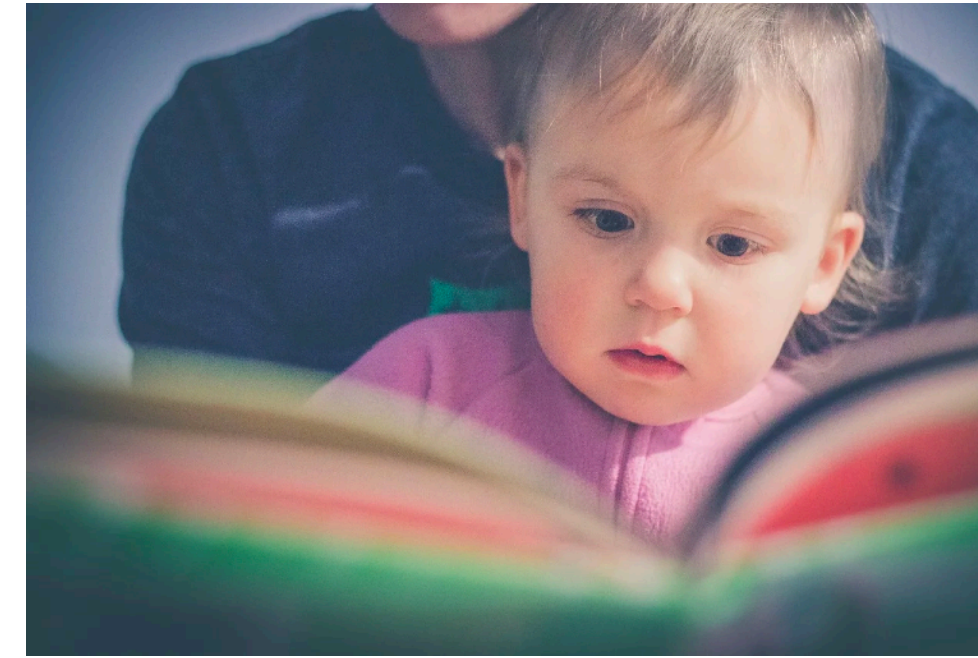
Rich sources of prior experiences.



visual imagery



tactile feedback



language



social cues

Can we learn priors across multiple data modalities?

Varying dimensionalities, units

Carry different, complementary forms of information

Some hints might come from some recent works.

Liang et al. Cross-Modal Generalization: Learning in Low Resource Modalities via Meta-Alignment. MM 2021.

Reed, Zolna, Parisotto et al. Gato: A Generalist Agent. TMLR 2022

Alayrac, Donahue, Luc, Miech et al. Flamingo: a Visual Language Model for Few-Shot Learning

OpenAI. GPT-4V(ision) System Card. 2023.

**But these are mostly
vision+language!**

Open Challenges in Multi-Task and Meta Learning

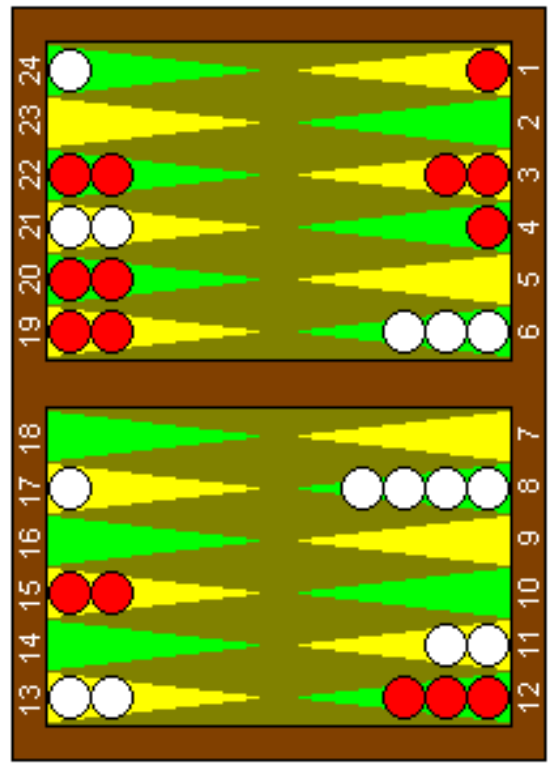
Improving scalability

- How best to use meta-learning algorithms in conjunction with foundation models?
- Can we make large-scale bi-level optimization more practical?

Addressing problem assumptions

- Generalization: Out-of-distribution tasks, long-tailed task distributions
- Multimodality: Can you learn priors from multiple modalities of data?
- Algorithm, Model Selection: When will multi-task learning help you?

Developing generalists



TD Gammon



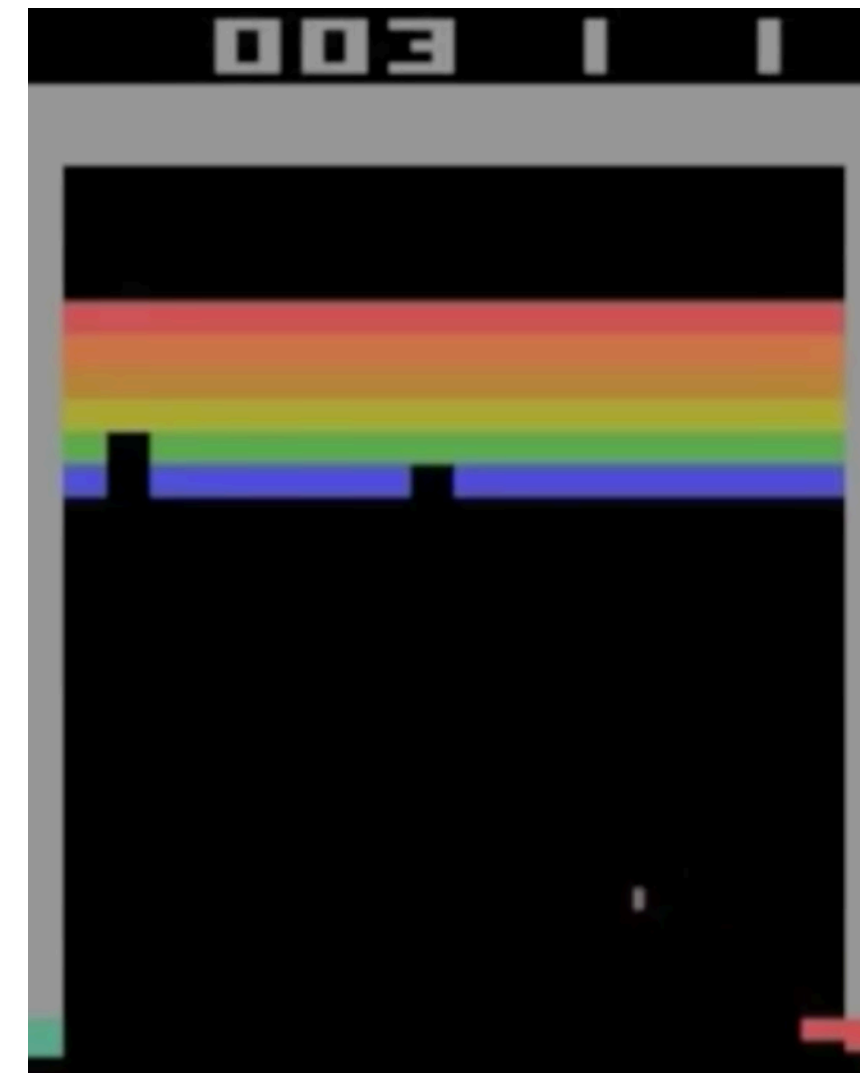
Watson



helicopter acrobatics



machine translation



DQN



Many machines are *specialists*.



Humans are *generalists*.

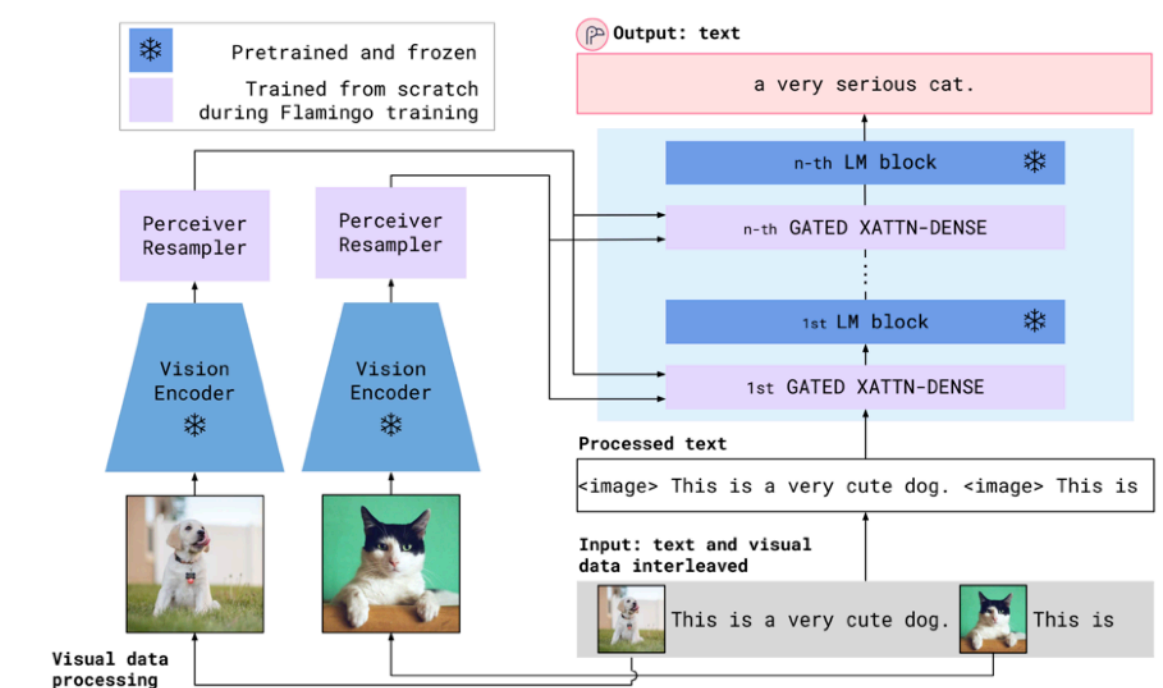
Some generalist models

In language, vision domains



ChatGPT

Flamingo



How do we build other generalists, extend existing ones?

e.g. generalist robot; generalist web agent; extending GPT to handle video; ...

Some of what we covered in CS330:

- learn multiple tasks in a single model (multi-task learning)
- leverage prior experience when learning new things (pre-training, meta-learning)
- leveraging *unlabeled* prior data (contrastive, generative pre-training)
- leveraging data from different domains (domain adaptation & generalization)
- learn continuously (lifelong learning)

What's missing?

Open Challenges in Multi-Task and Meta Learning

Improving scalability

- How best to use meta-learning algorithms in conjunction with foundation models?
- Can we make large-scale bi-level optimization more practical?

Addressing problem assumptions

- Generalization: Out-of-distribution tasks, long-tailed task distributions
- Multimodality: Can you learn priors from multiple modalities of data?
- Algorithm, Model Selection: When will multi-task learning help you?

Developing generalists

- Can we build generalists in domains beyond language? (e.g. robotics, web nav, scientists)
- What is needed to extend the capabilities of existing generalist models?

+ the challenges you discovered in your homework & final projects!

Logistics

Poster session on Weds

Details on Ed.

Final project report

Due next Monday.

This is our last lecture!

Thank you all for a great quarter!

(and see you at the poster session on Weds!)