Model-Based Reinforcement Learning via Meta-Policy Optimization

Ignasi Clavera*, Jonas Rothfuss*,
John Schulman, Yasuhiro Fujita, Tamim Asfour, Pieter Abbeel

UC Berkeley **Karlsruhe Institute of Technology**

Motivation: model-free vs. model-based RL

Model-free RL

+ Good asymptotic performance
+ Effective for learning complex policies

- High sample complexity

Model-based RL through control

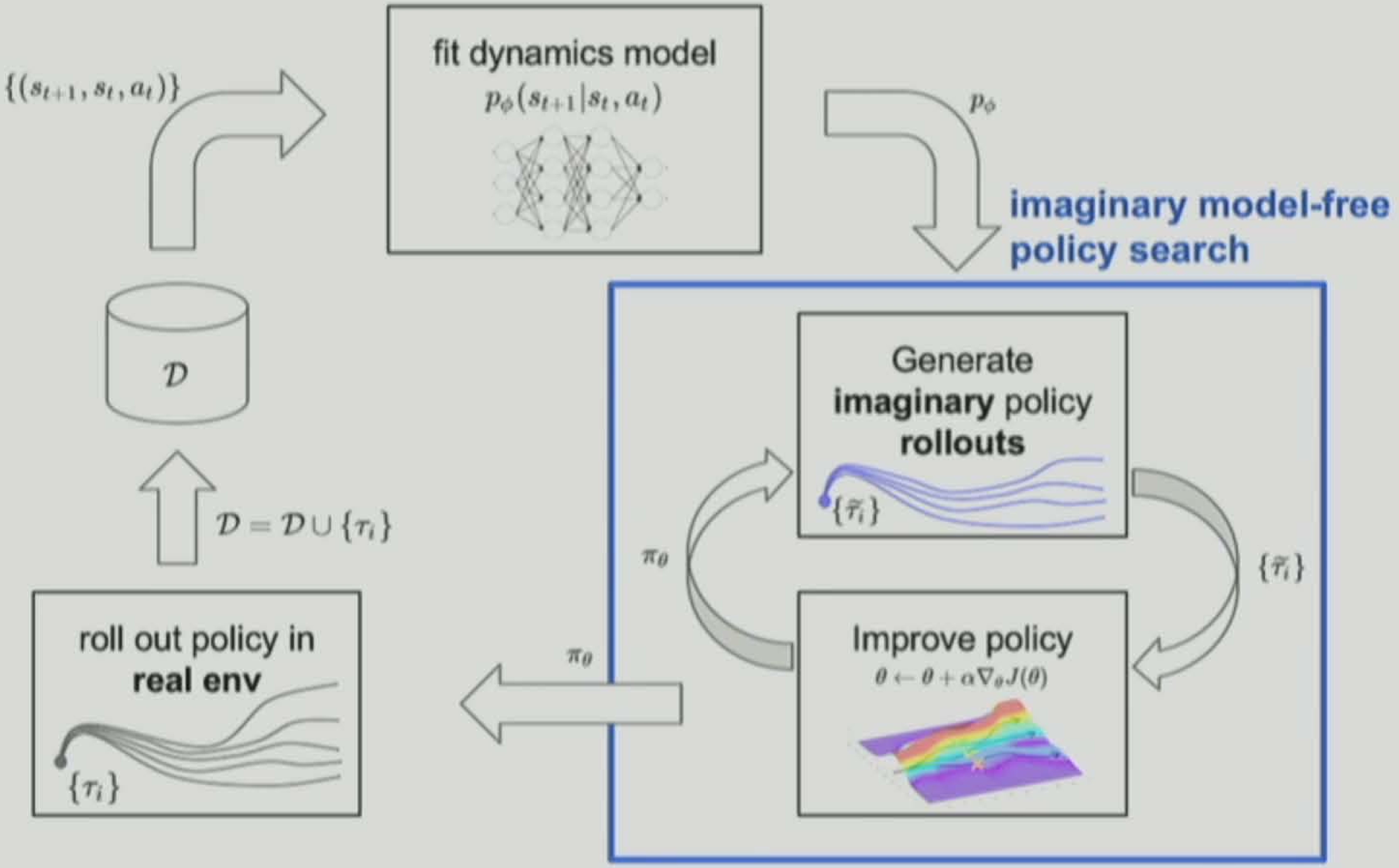
+ Low sample complexity
+ Possibility to transfer across tasks

- Restricted to short horizon planning or simple dynamics

Can we combine the advantages of both approaches?
→ **Model-based-model-free approaches**

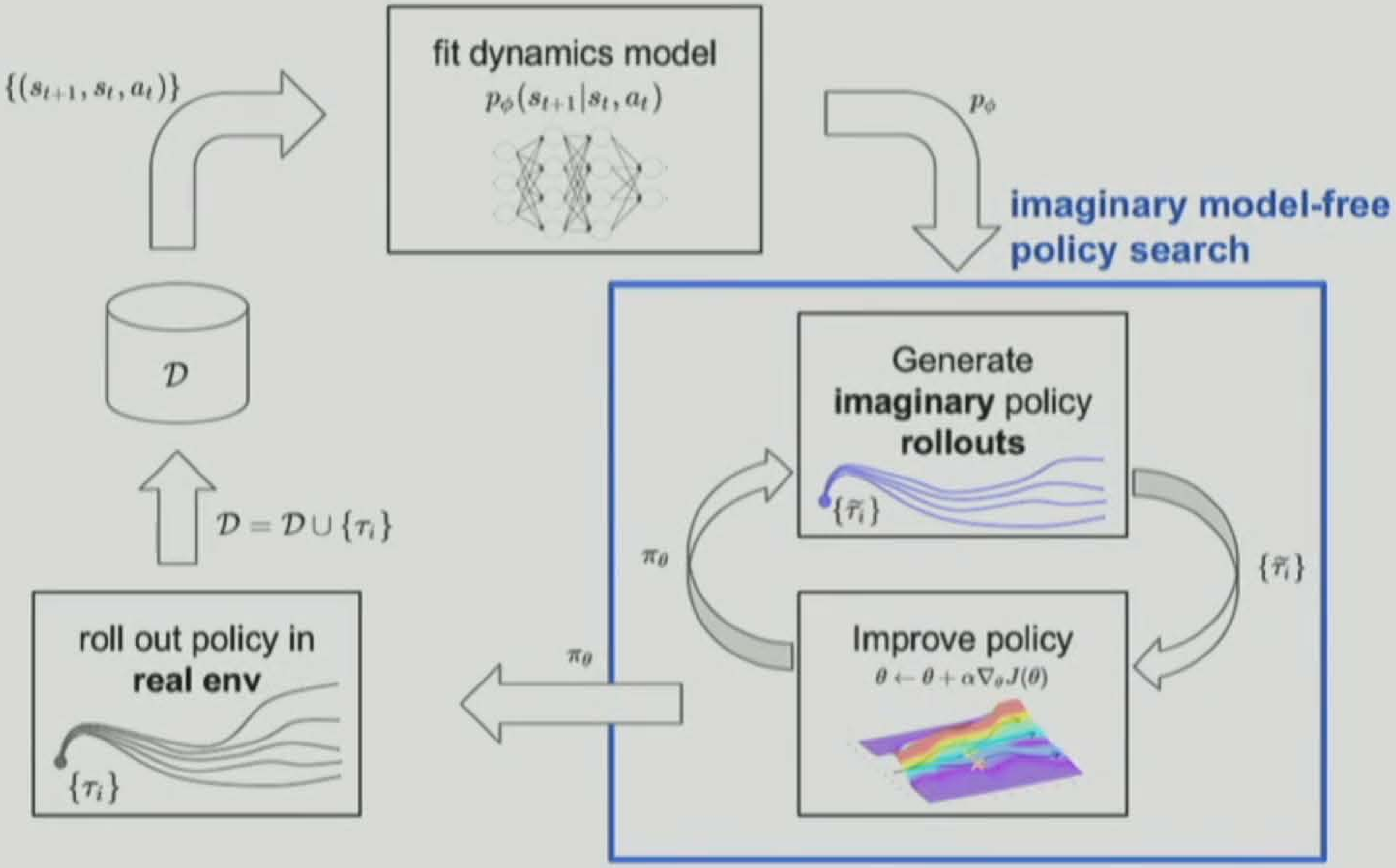
Naive idea to combine model-based and model-free

- **Learn a simulator** from transition data
- **Model-free policy search** with simulated / **imaginary environment** interactions



What is the problem?

- **Compounding errors** lead to unrealistic trajectories
- Policy **overfits** to deficiencies of dynamics model → **model bias**
- Policy behavior does **not successfully transfer** to the real environment



Model-based RL as a meta-learning problem

Key idea 1:

Learn an ensemble of dynamics models

Key idea 2:

Consider each model as a (meta-learning) task/MDP

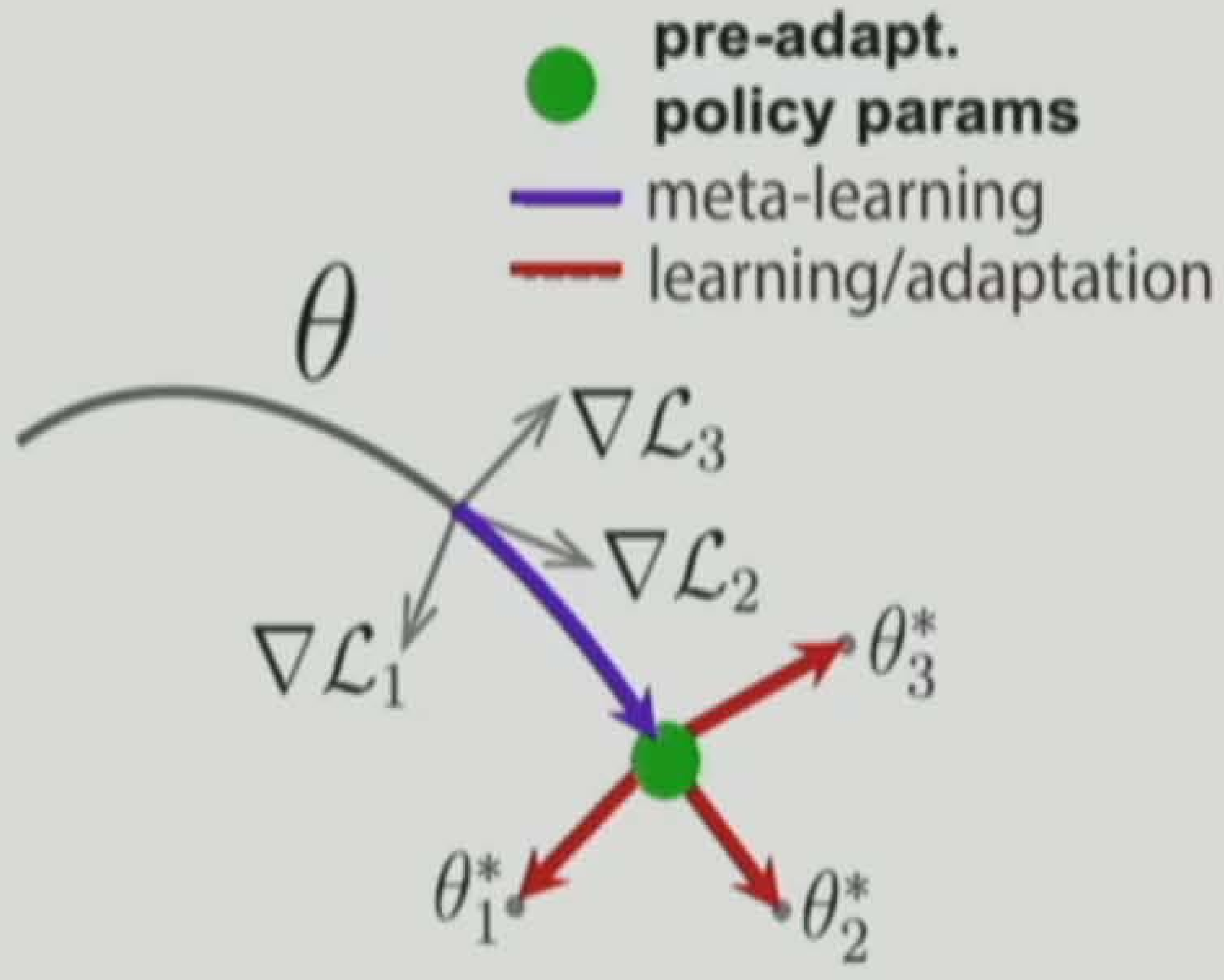
⇨ phrase model-based RL as meta-learning problem

Solution: Meta-Reinforcement Learning

- "Meta-Learning = Learning to learn"
- Learn over a distribution of tasks (MDPs)

$$\mathcal{T} \sim \rho(\mathcal{T})$$

- Goal: Adapt fast to a new task / MDP
- **Gradient-based Meta-Learning (e.g. MAML):**
    - **Learn** very good **initial parameters**
    - Perform one or few policy gradient **adaptation step(s)**
    - Maximize performance after the gradient update(s)

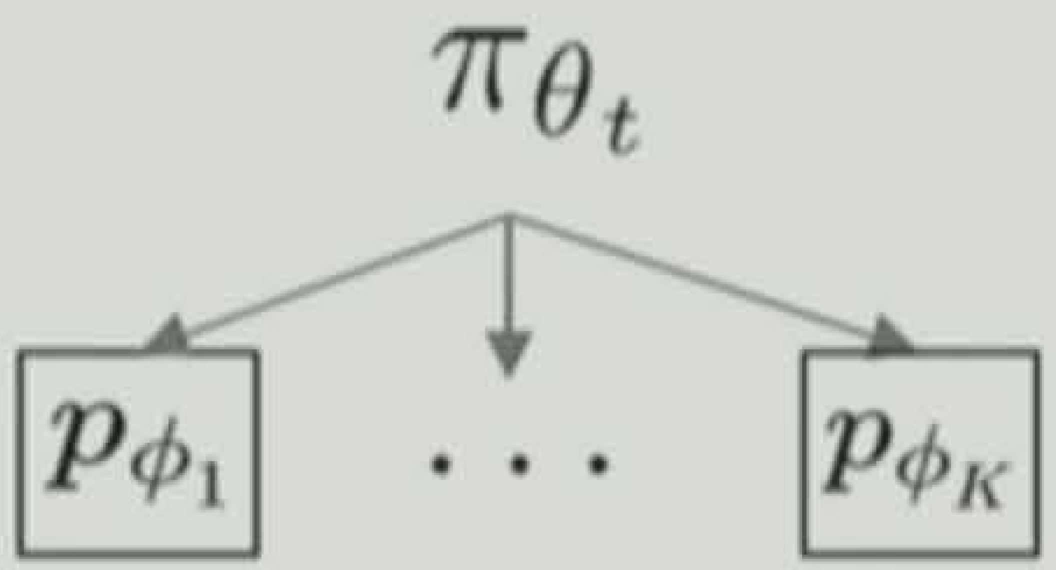


Finn et al. (2017), *"MAML: Model Agnostic Meta Learning"*

Model-based RL as a meta-learning problem

π_{θ_t} **pre-update** policy

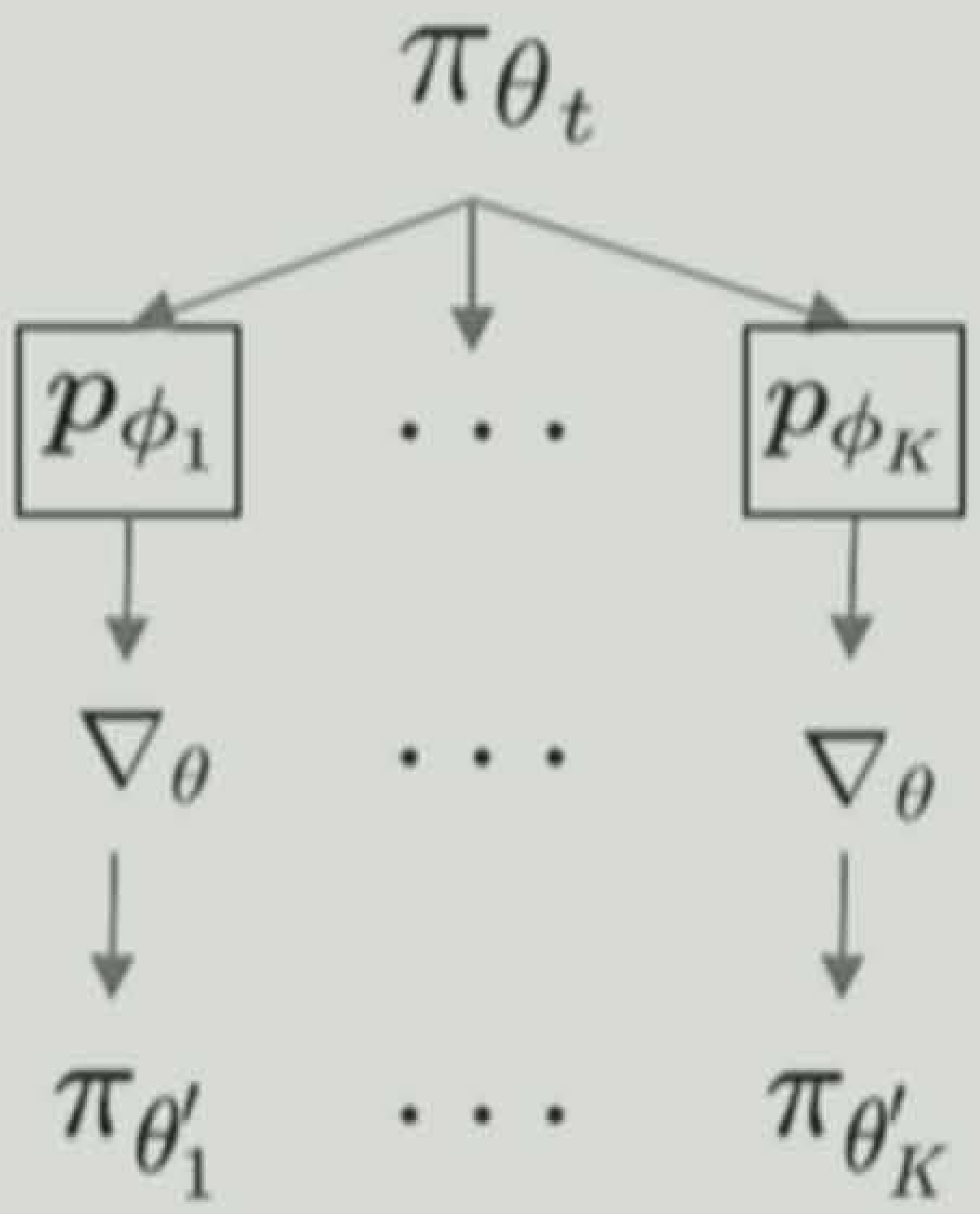
Model-based RL as a meta-learning problem



pre-update policy

generate trajectories with the K dynamic models

Model-based RL as a meta-learning problem



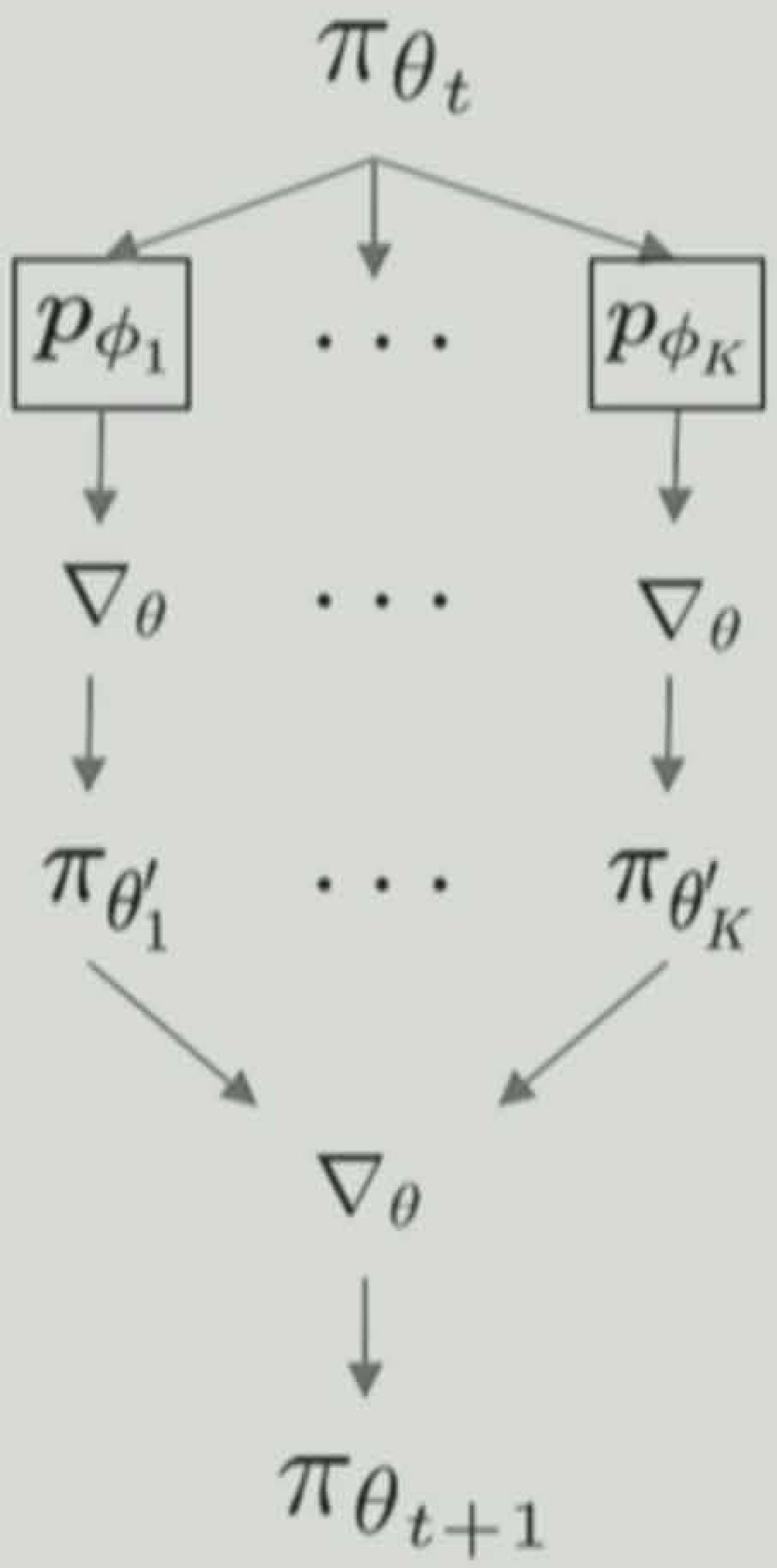
pre-update policy

generate trajectories with the K dynamic models

adaptation step: policy gradient update

post-update policy

Model-based RL as a meta-learning problem



pre-update policy

generate trajectories with the K dynamic models

adaptation step: policy gradient update

post-update policy

meta-gradient step

Model-based RL as a meta-learning problem

π_{θ_t} **pre-update** policy

$p_{\phi_1} \quad \cdots \quad p_{\phi_K}$ **generate trajectories** with the K dynamic models

$\nabla_\theta \quad \cdots \quad \nabla_\theta$ **adaptation step**: policy gradient update

$\pi_{\theta'_1} \quad \cdots \quad \pi_{\theta'_K}$ **post-update** policy

∇_θ **meta-gradient** step

$\pi_{\theta_{t+1}}$

repeat

Model-based RL as a meta-learning problem

Use gradient-based Meta-RL to learn to adapt fast to dynamics models

Meta-RL objective

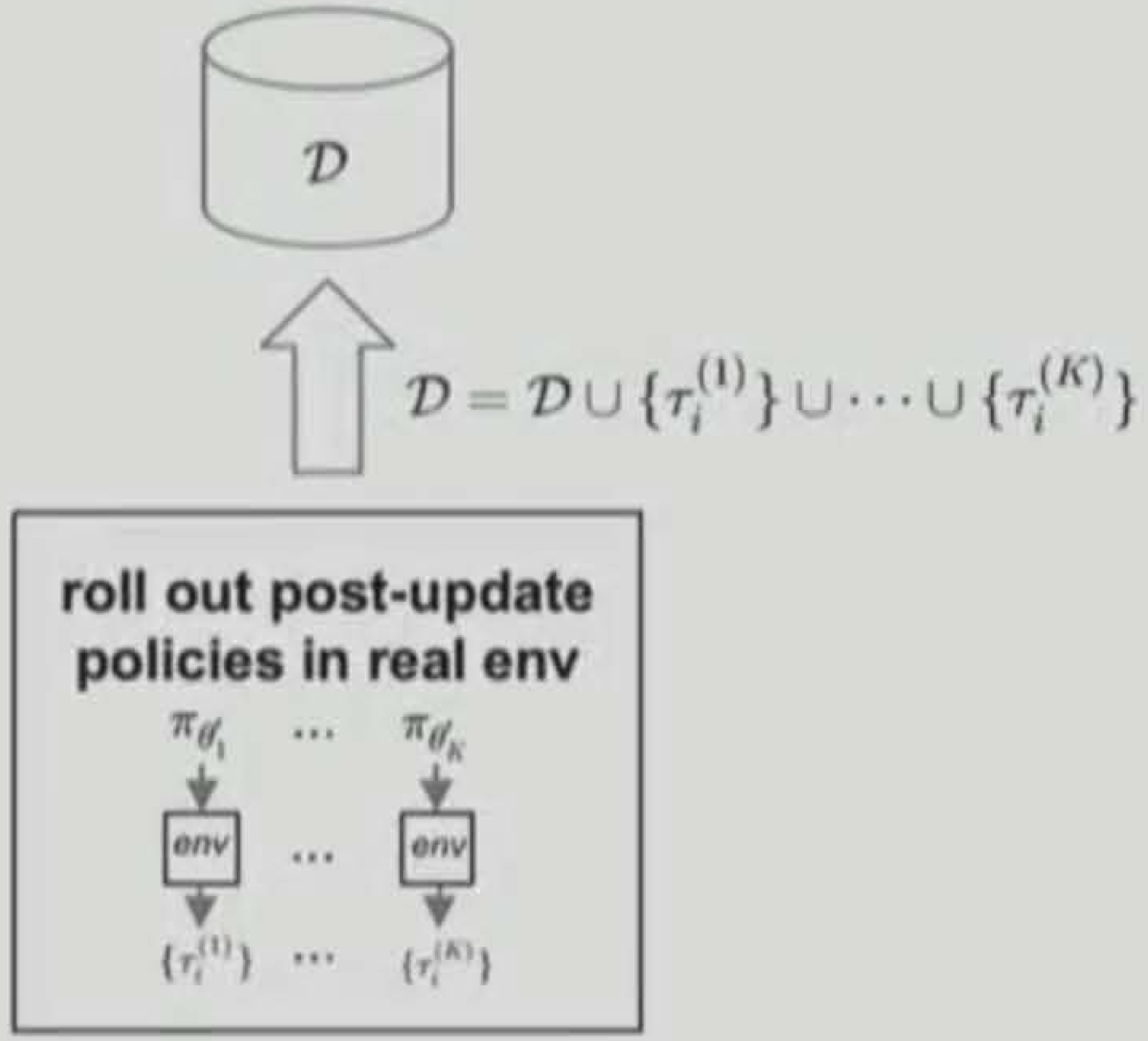
$$\max_\theta \quad \frac{1}{K}\sum_{k=0}^{K} J_k(\theta'_k) \qquad \text{s.t.:} \quad \underbrace{\theta'_k = \theta + \alpha \nabla_\theta J_k(\theta)}_{\text{adaptation step}}$$

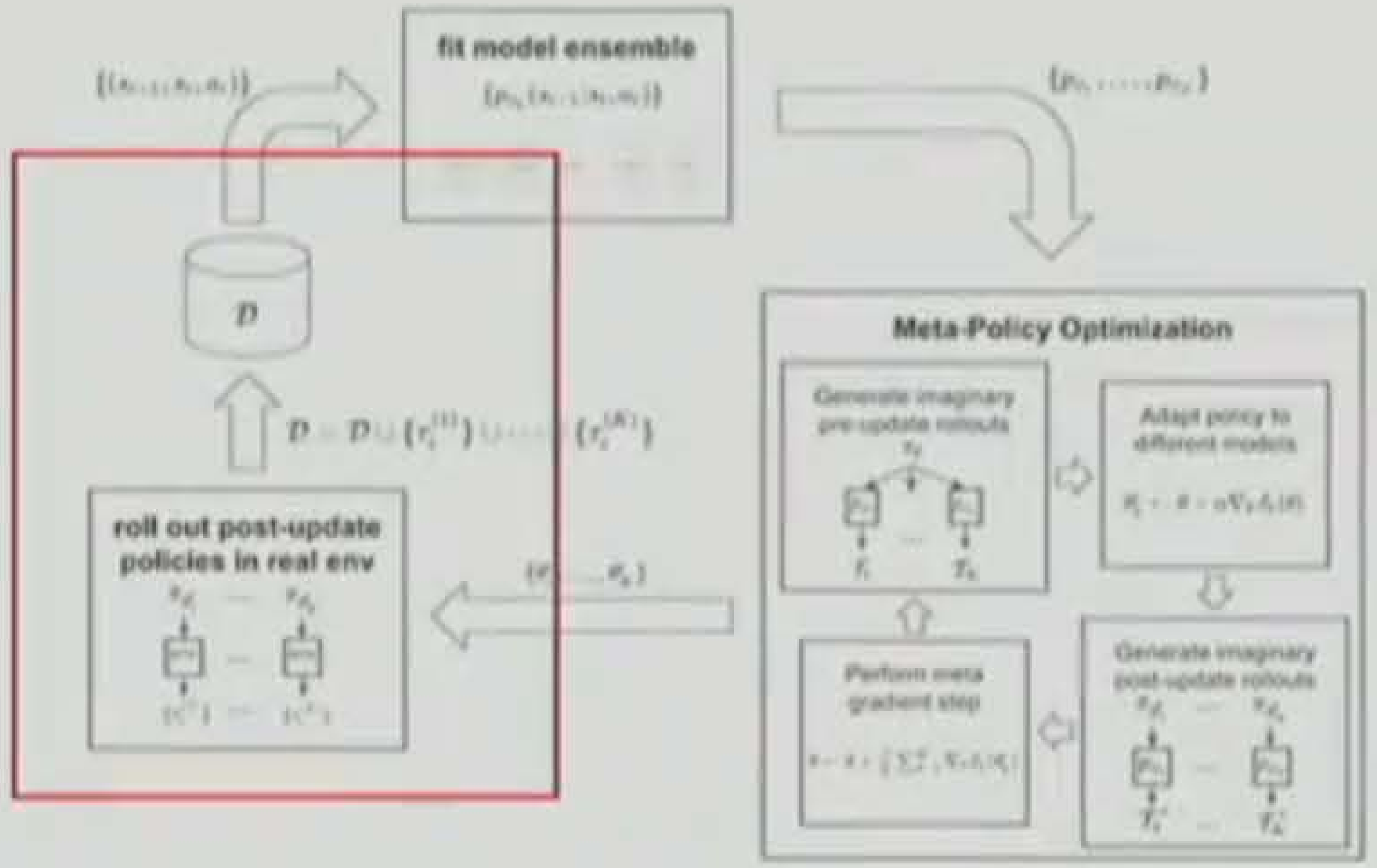
RL objective w.r.t learned model

$$J_k(\theta) = \mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)}\left[\sum_{t=0}^{H-1} r(s_t, a_t) \middle| \underbrace{s_{t+1} = \hat{f}_{\phi_k}(s_t, a_t)}_{\text{model predictions}}\right]$$

Model-Based Meta-Policy Optimization (MB-MPO)

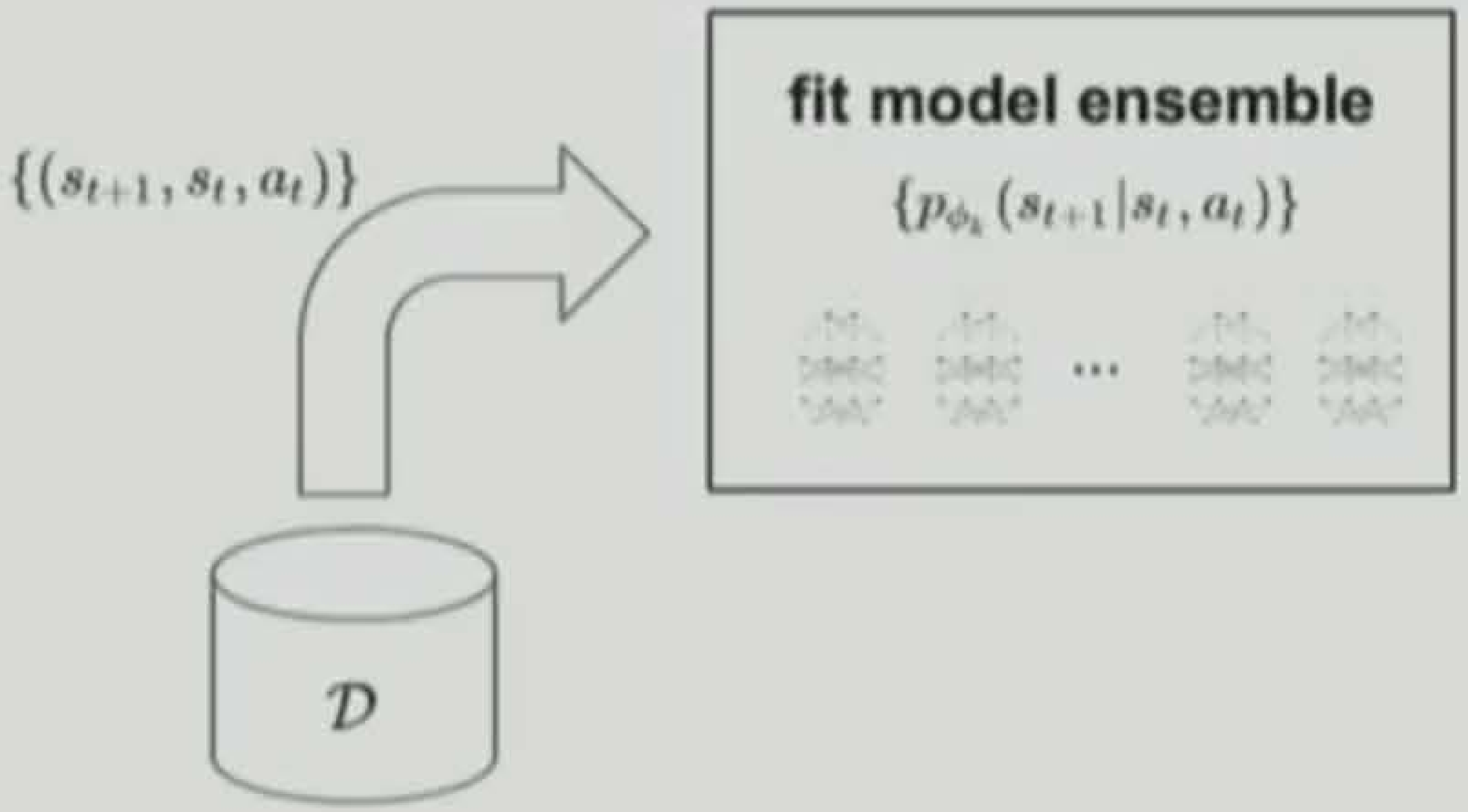
1. Collect data with the post-update policies in the real environment
2. Fit the ensemble of models
3. Meta-learn a policy on the ensemble

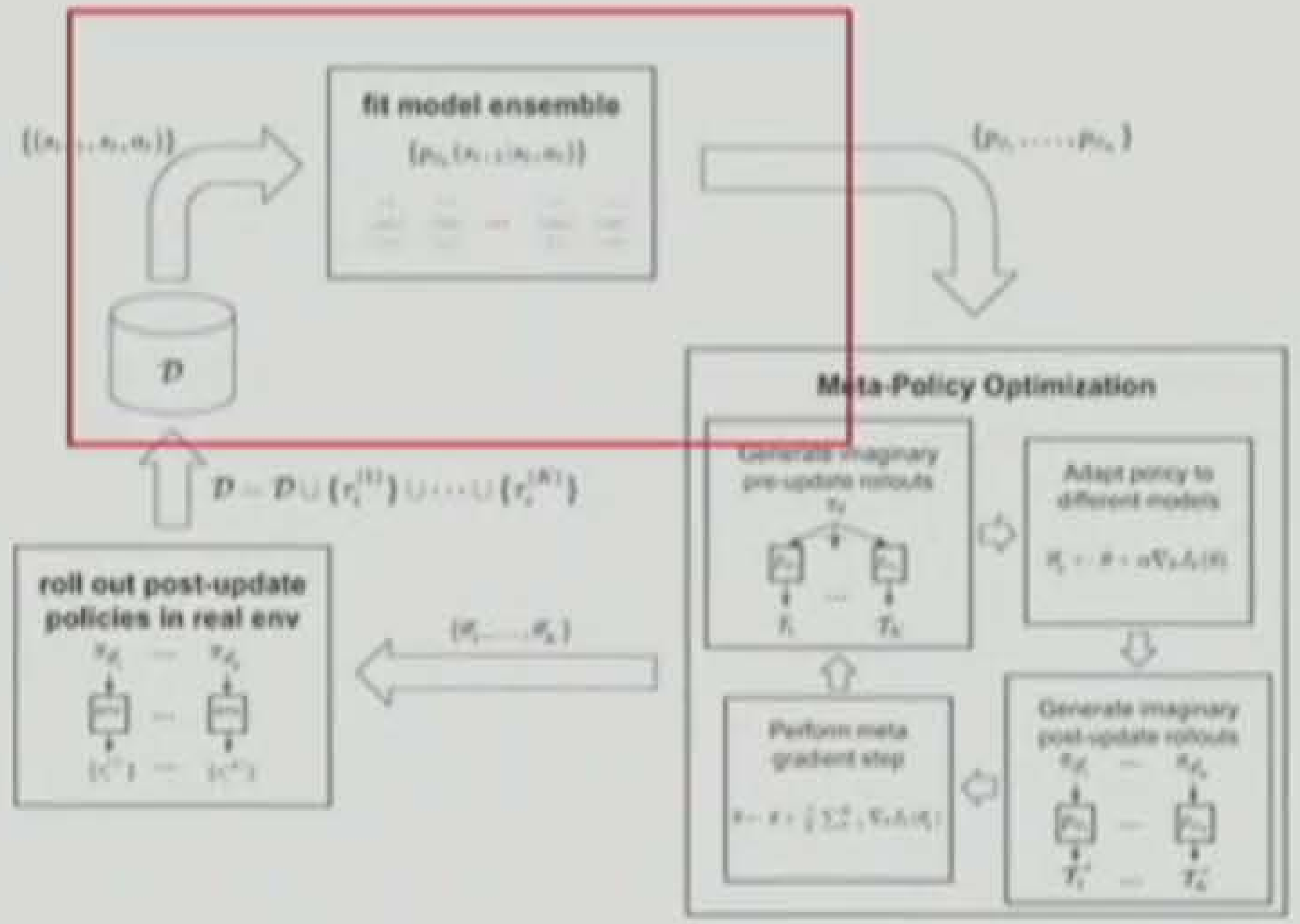




Model-Based Meta-Policy Optimization (MB-MPO)

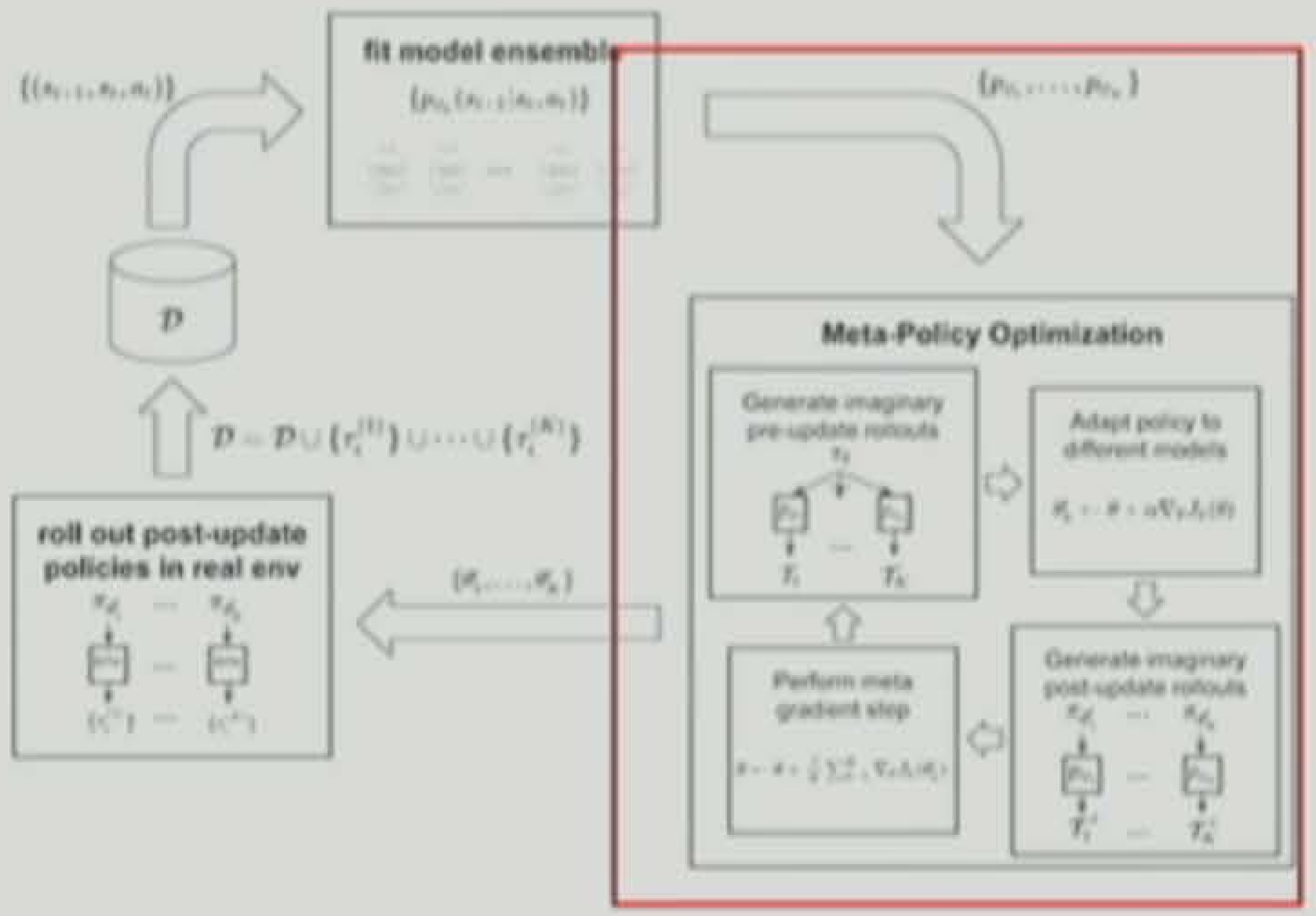
1. Collect data with the post-update policies in the real environment
2. Fit the ensemble of models
3. Meta-learn a policy on the ensemble

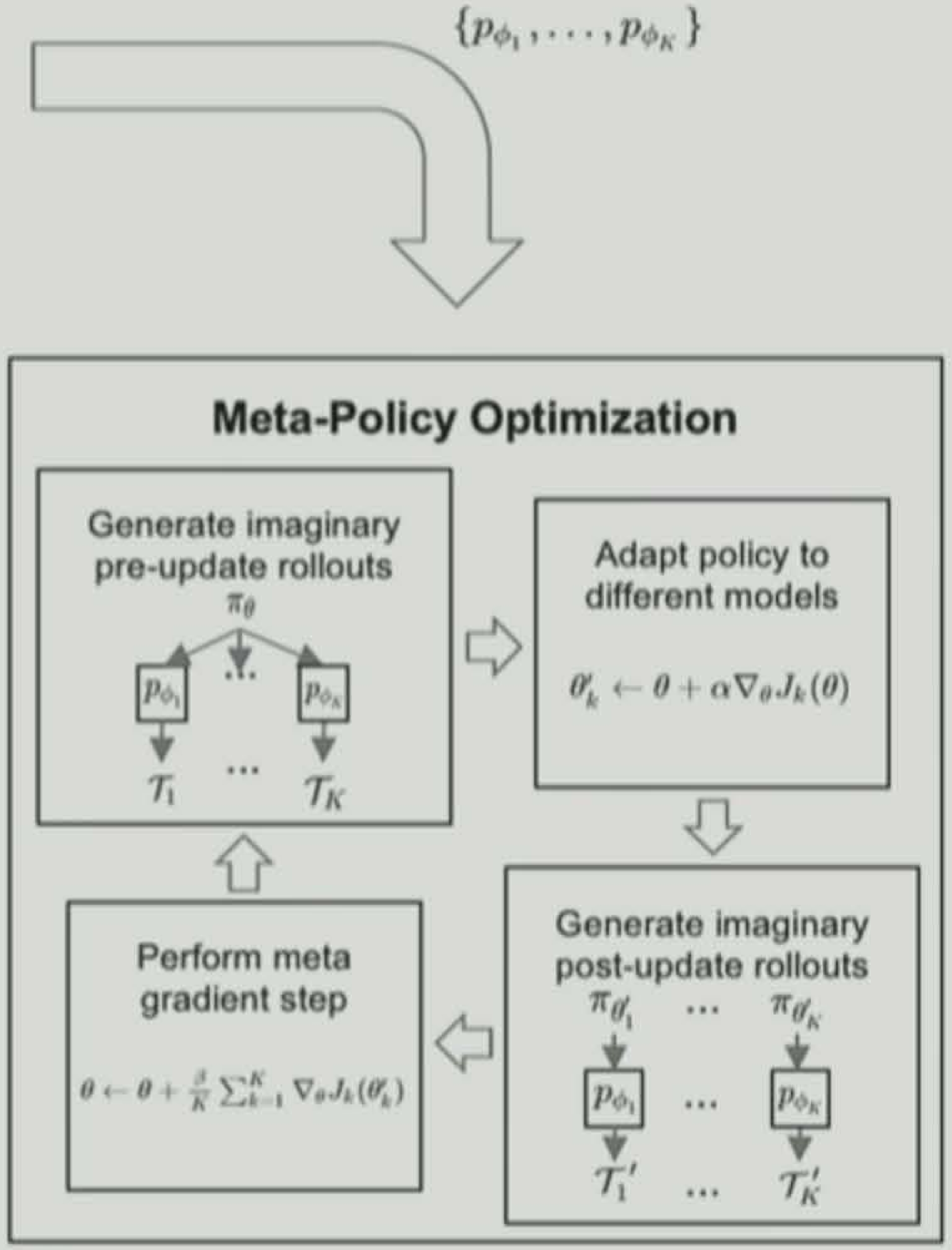




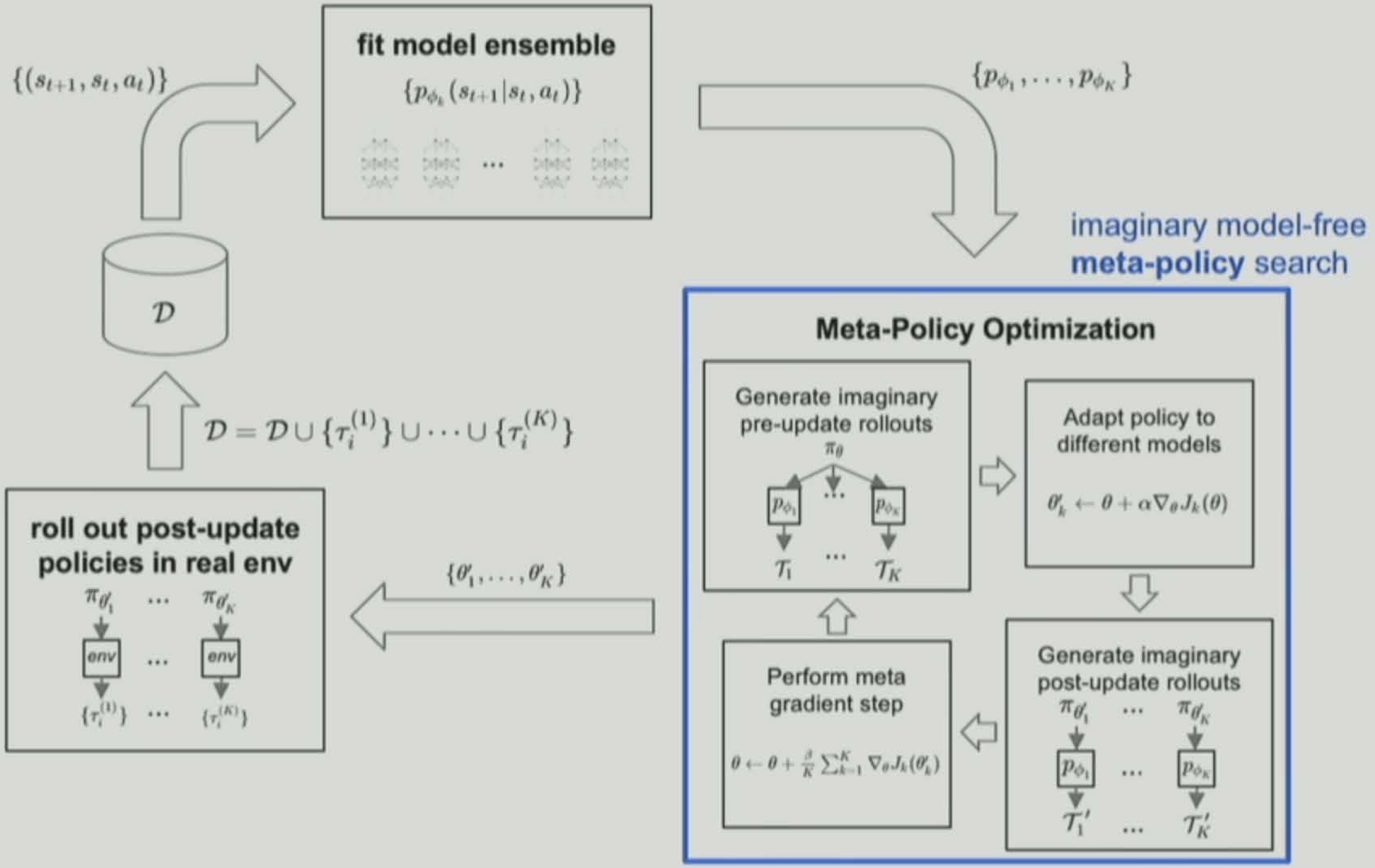
Model-Based Meta-Policy Optimization (MB-MPO)

1. Collect data with the post-update policies in the real environment
2. Fit the ensemble of models
3. Meta-learn a policy on the ensemble





Model-Based Meta-Policy Optimization (MB-MPO)



Results

Mujoco Locomotion Benchmarks

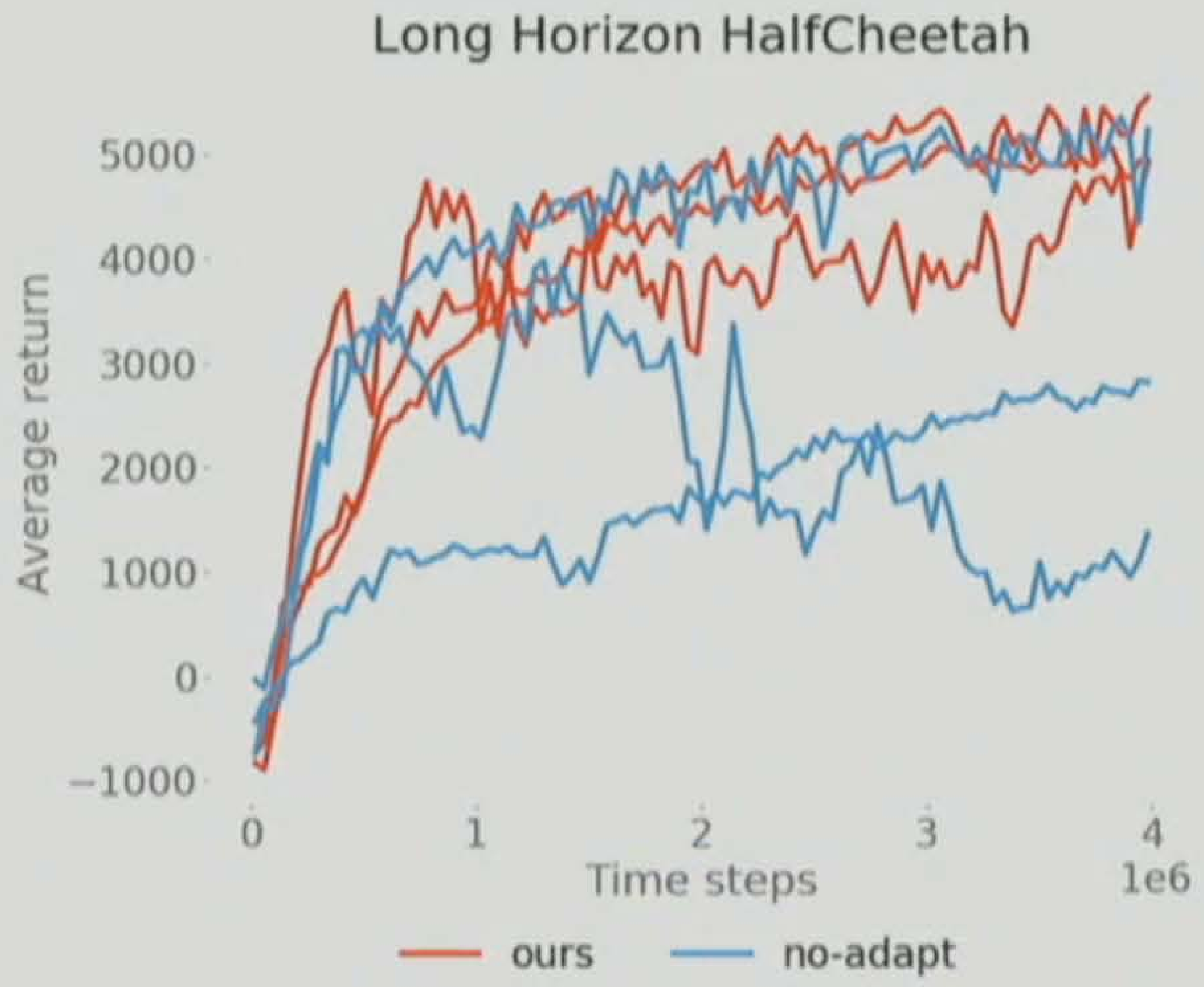
Robustness to long horizons tasks

- Open loop prediction for 1000 time-steps

  → Highly susceptible to compounting errors

- Multiple random seeds

<u>MB-MPO successfully learns across seeds</u>



Why does MB-MPO works so well?

1. Regularization effect during training

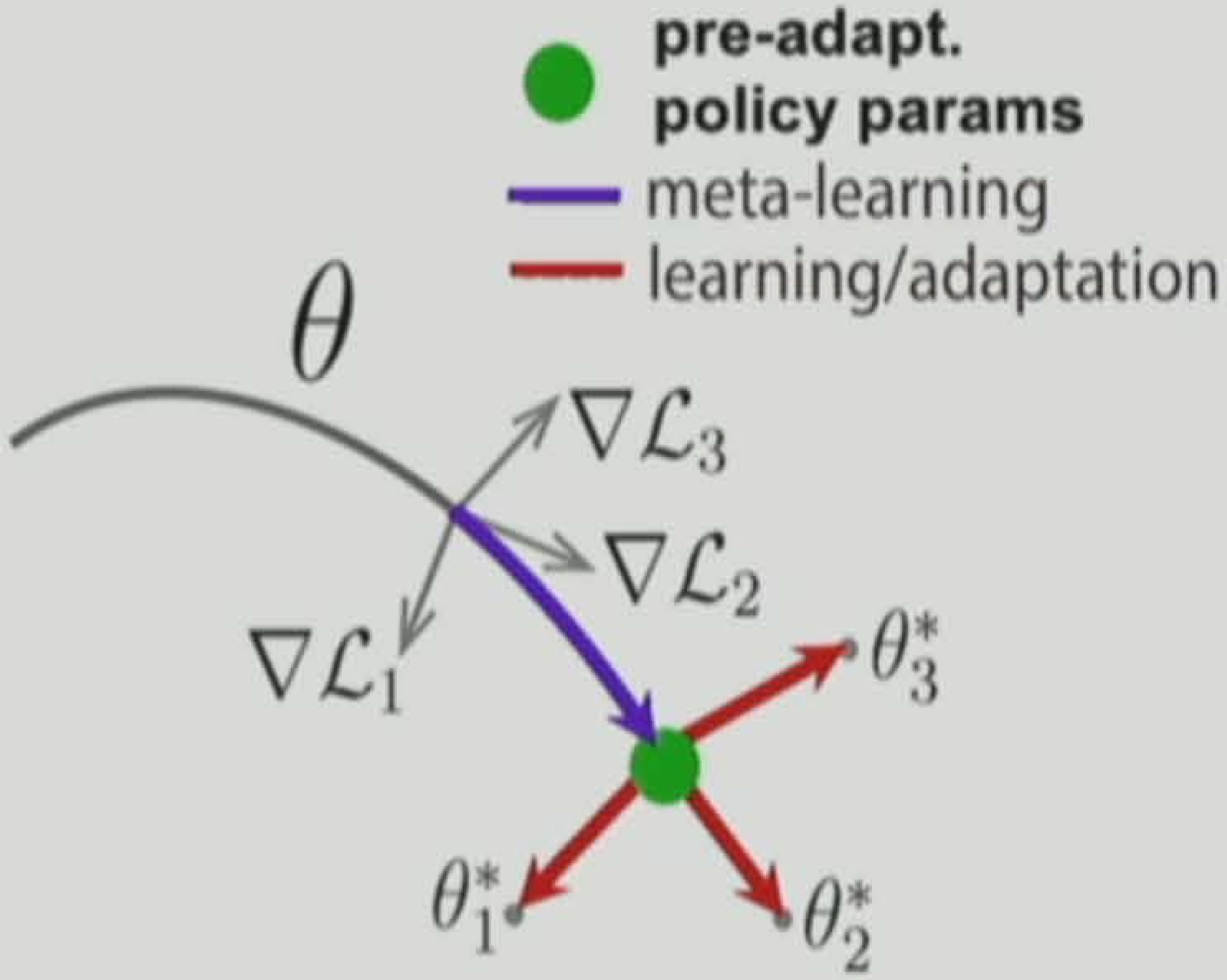
2. Tailored data collection

Regularization effect during training

- **"Relaxed policy search"**
  - Discrepancies between models

    → Adaptation step
  - Consistent dynamics prediction among the ensemble

    → Internalized in the pre-update policy



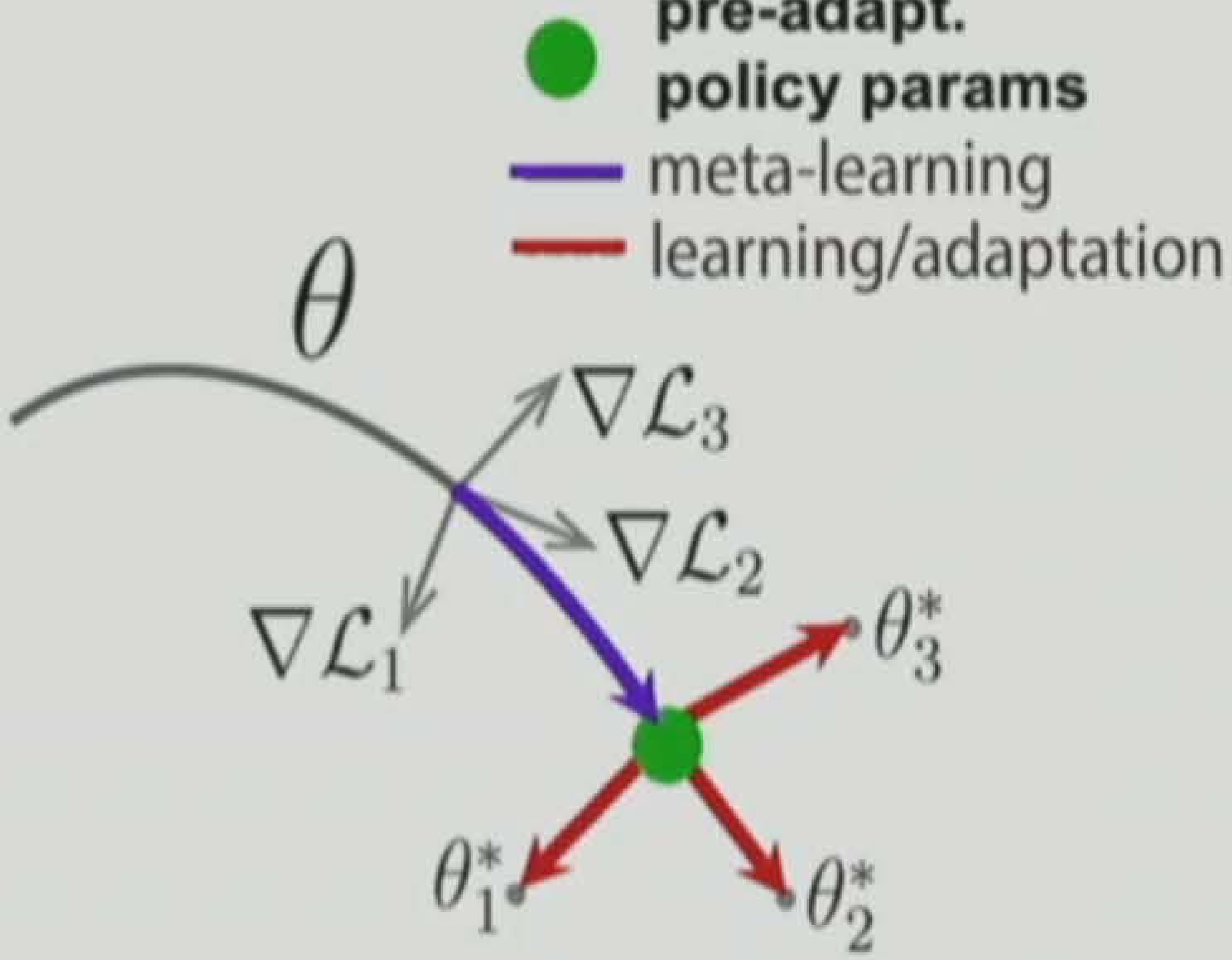
Finn et al. (2017), *"MAML: Model Agnostic Meta Learning"*

Regularization effect during training

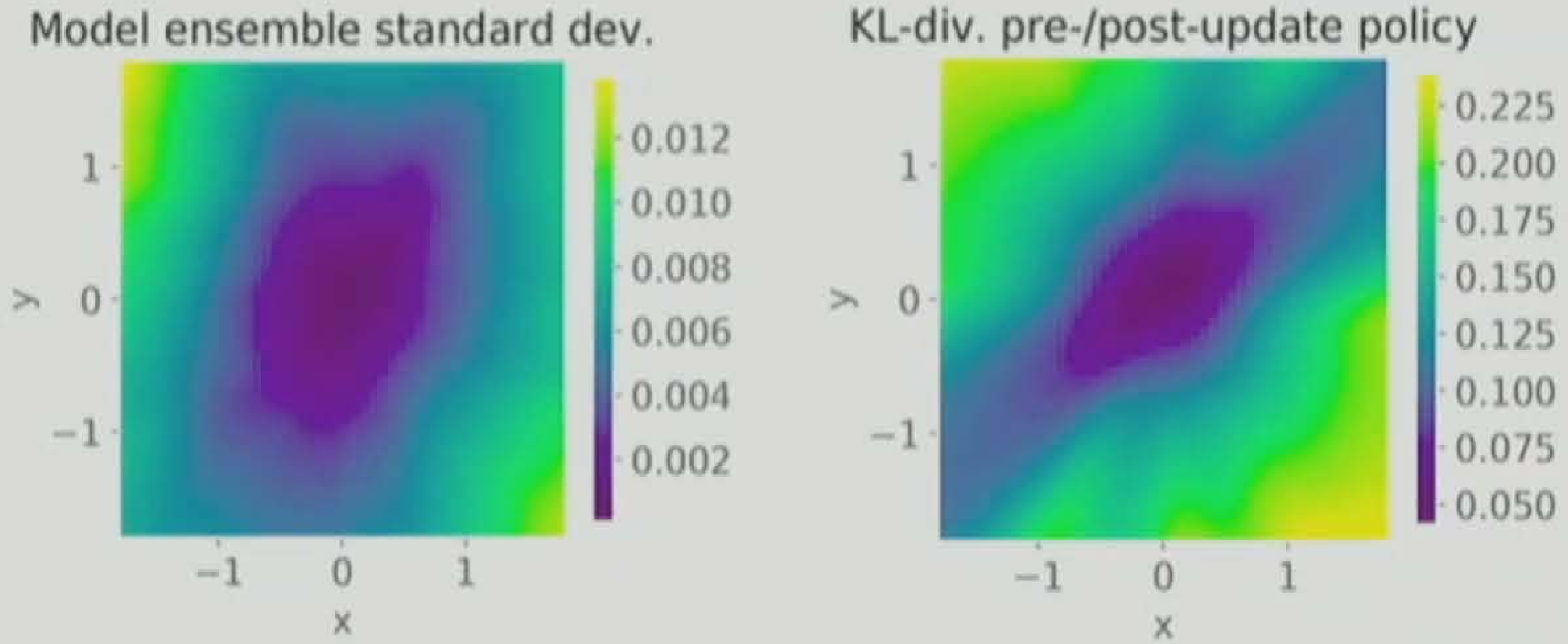
- **"Relaxed policy search"**
  - Discrepancies between models

    → Adaptation step
  - Consistent dynamics prediction among the ensemble

    → Internalized in the pre-update policy



Finn et al. (2017), *"MAML: Model Agnostic Meta Learning"*



⇒ High correlation between model uncertainty and policy adaptability

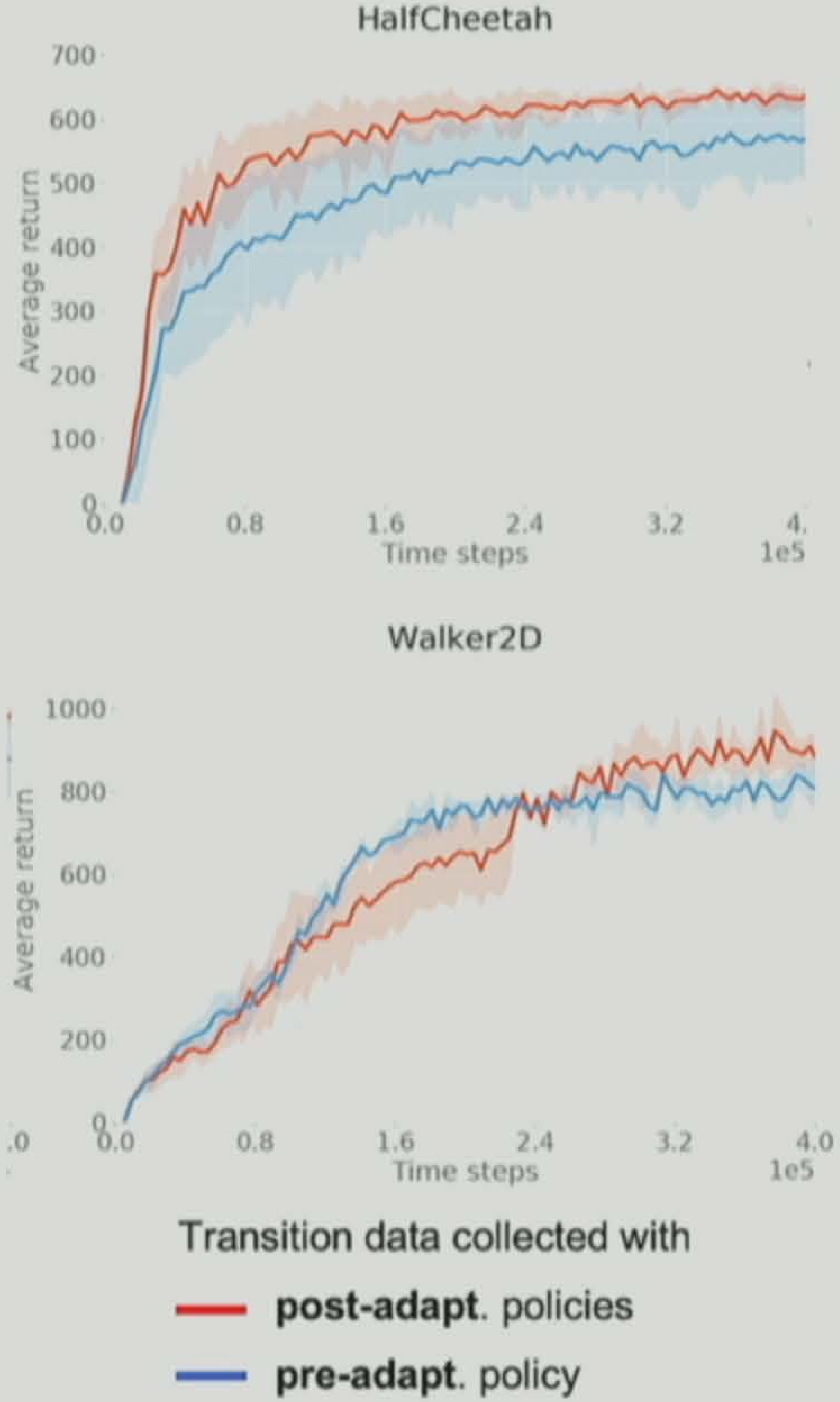
⇒ Prevents overfitting to deficiencies of models

Tailored data collection

- Post-adapt. policies are overfitted to their dynamics model
- Exploits characteristic deficiencies of the models

→ Data collection in regions where models are bad

→ More diverse transition data



Summary

Key ideas:

1) Use the models as simulators
2) Learn an ensemble of dynamics models
3) Phrase model-based RL as meta-learning problem

Summary

Key ideas:

1) Use the models as simulators
2) Learn an ensemble of dynamics models
3) Phrase model-based RL as meta-learning problem

Results:

1) MF performance in high-dimensional control environments
2) 10 - 100 times more data efficient