Learning Policy Committee for Personalized MDPs

ICML 2025

Luise Ge, Michael Lanier, Anindya Sarkar, Bengisu Guresti, Chongjie Zhang, Yevgeniy Vorobeychik
Washington University in St Louis

Why Personalized MDPs?

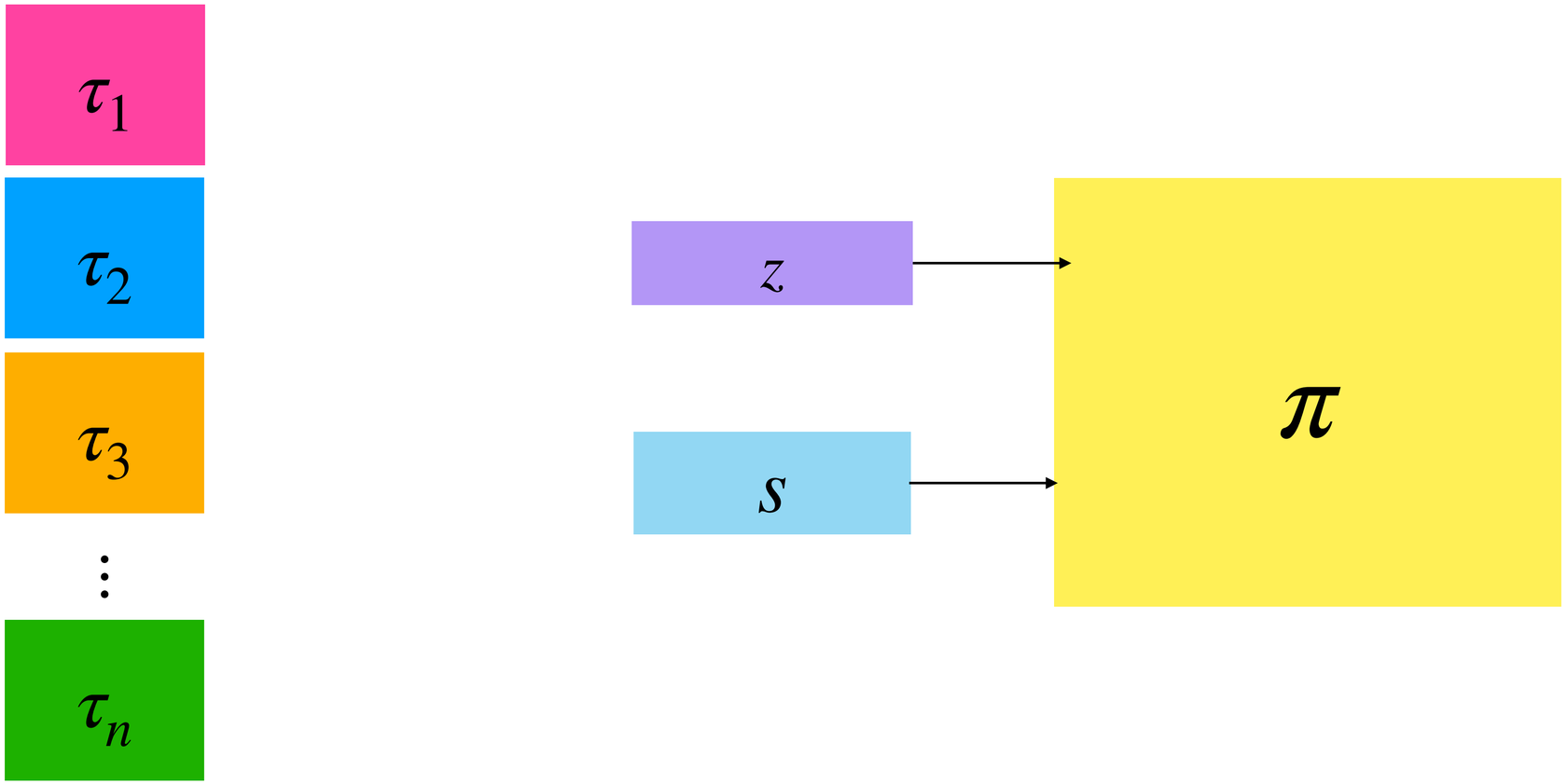
Real-world RL applications don't involve a single reward function or dynamics model — they involve ***a population***.
(eg. traffic signal control; healthcare; education; recommendation engine)

But it is impractical to allocate lots of computational resources to solve each task individually.

How do we serve *individuals*, not averages, under fixed computational constraints?

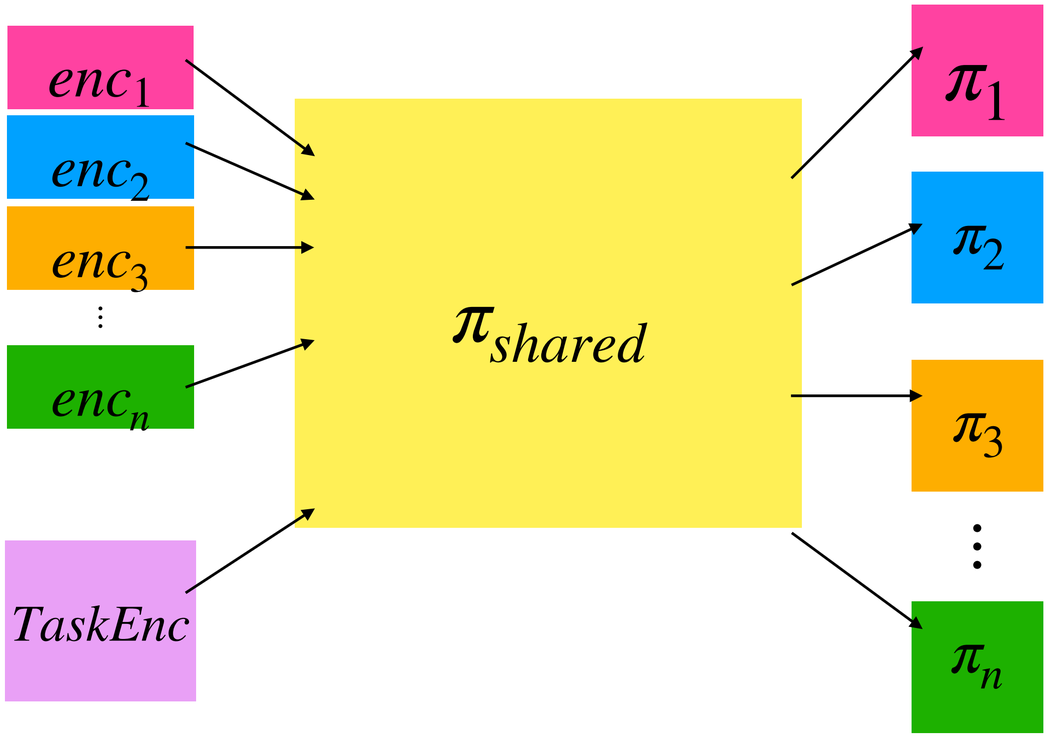
Related Work

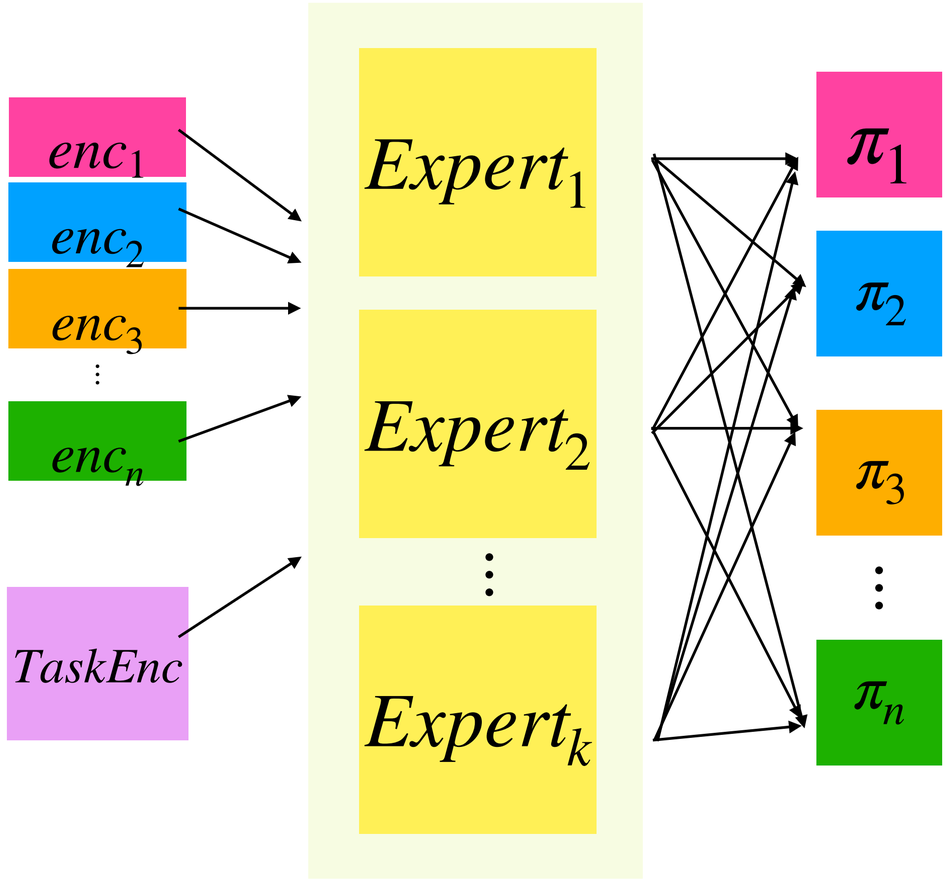
Multi-task RL: solve a given set of tasks
eg. with contextual MDPs



Related Work

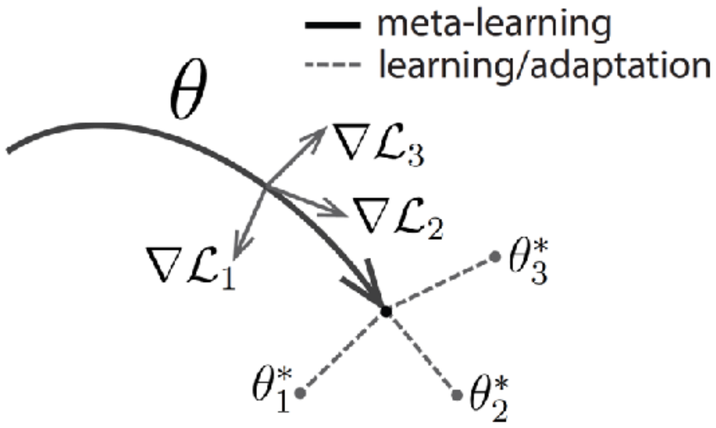
MT Learner with various architectures…

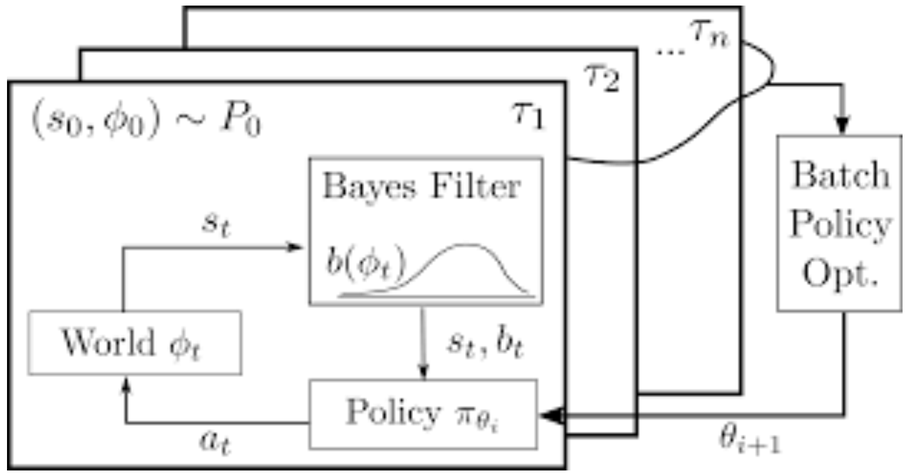




Related Work

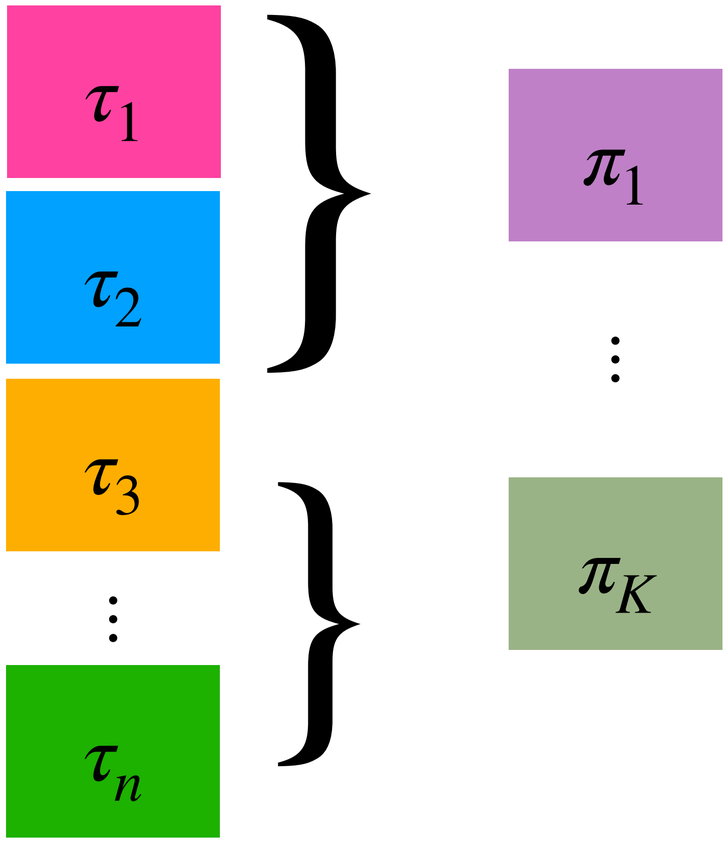
Meta RL: find a good initialization for adaptation





Related Work

Personalized/Clustered RL: train a set of K policies (K<<n) to maximize $\mathbb{E}_{\tau\sim\Gamma}[\sum_{\pi\in\Pi}\alpha_\pi(\tau)\mathscr{V}_\tau^\pi]$



[(Ivanov et Ben-Porat, 2024)]

Algorithm 1 EM-like meta-algorithm

Input: r-MDP $\mathcal{M}_r$

Arbitrarily initialize $(\alpha^i)_{i\in N}$ s.t. $\forall j : \exists i, \alpha^i(j) > 0$
while assignments or policies change **do**
    Perform M-step to update policies
    Perform E-step to update assignments
end while

Set-Up

Consider a **dynamic environment** $\mathscr{E} = (\mathcal{S}, \mathcal{A}, h, \gamma, \rho)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, h the decision horizon, γ the discount factor, and ρ the initial state distribution.

Let a **task** be $\tau = (\mathcal{T}, r)$ where $\mathcal{T}$ is the transition model (distribution over next state given current state and action) and r the reward function.

Let Γ be the (unknown) distribution over tasks.
What we want is to solve a distribution of MDPs $(\mathscr{E}, \Gamma)$ in the following sense…

New Objective: $(\epsilon, 1-\delta)$ Cover

We wish to find a policy committee Π with fixed size K such that with probability at least $1-\delta$, for $\tau \sim \Gamma$, $\max_{\pi \in \Pi} V_\tau^\pi \geq V_\tau^* - \epsilon$.

(in other words, with high probability, there is a policy in the committee that is near-optimal for the drawn task)

Then we call Π a $(\epsilon, 1-\delta)$ cover for the task distribution Γ.

This is to be contrasted with traditional objectives like $max_\pi \mathbb{E}_{\tau \sim \Gamma}[\mathscr{V}_\tau^\pi]$ or $max_\Pi \mathbb{E}_{\tau \sim \Gamma}[\sum_{\pi \in \Pi} \alpha_\pi(\tau) \mathscr{V}_\tau^\pi]$!

Algorithmic Framework

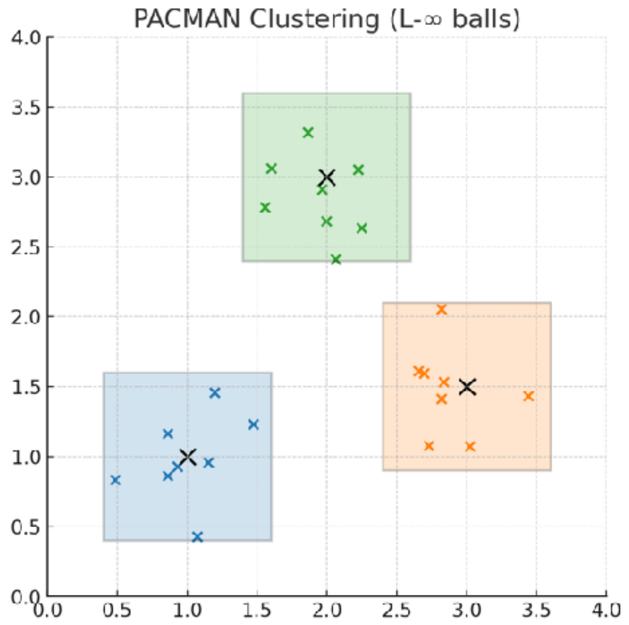
Suppose that tasks are parametric. Given a finite sample of tasks T drawn i.i.d. from Γ,

our framework proceeds in two steps:

1. Cluster the tasks in T via seeking K prototypes embeddings that maximize the empricial coverage

$$\max_{\{\theta_1,\ldots,\theta_K\}} \sum_{\theta \in T} \mathbf{1}(\min_{k \in [K]} \|\theta_k - \theta\|_\infty \leq \epsilon);$$

2. Train the committee via identifying a task for each cluster prototype and training a policy to optimality on it.



Clustering: Hardness and Approximation

Let's begin with K=1. Consider $\Theta = \{\theta_1, \ldots, \theta_n\} \subseteq \mathbb{R}^d$.

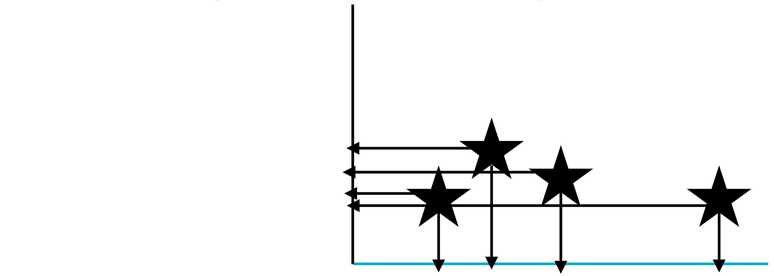
Find $\hat{\theta} \in \mathbb{R}^d$ which maximizes the coverage
$|S| = |\{\theta \in \Theta \; s.t. \|\theta - \hat{\theta}\|_\infty \leq \epsilon\}|$.

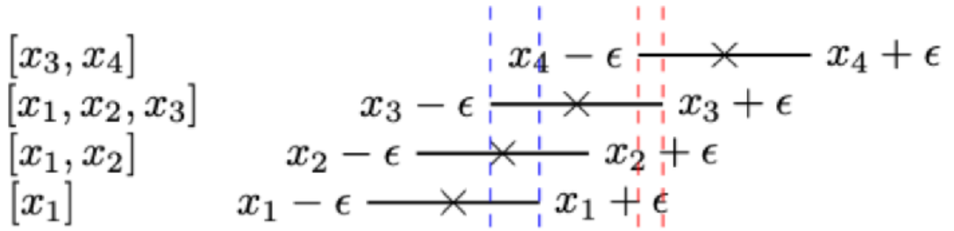
Theorem (informal): No $n^{1-\epsilon}$ approximation unless P=NP.
Theorem (informal): With a fixed dimension d, we can find a constant-factor $(1 - \frac{1}{e})$ approximation of optimal achievable cover within polynomial time.

Greedy Intersection Algorithm

1. $\|\theta_i - \theta_j\|_\infty \leq \epsilon$ iff for each dimension m, $\|\theta_i^m - \theta_j^m\|_\infty \leq \epsilon$

2. $\|\theta_i^m - \theta_j^m\|_\infty \leq \epsilon$ iff their potential representatives have intersection





For each cluster 1…K, we pick the maximum among all intersections of d lists, one from each dimension. **Drawback: running time is exponential in d.**

Gradient-Descent

$$\max_{\{\theta_1,\ldots,\theta_K\}} \sum_{\theta \in T} \mathbf{1}\left(\min_{k \in [K]} \|\theta_k - \theta\|_\infty \leq \epsilon \right)$$

find a set such that as many θs as possible are covered

$$\min_{\{\theta_1,\ldots,\theta_K\};\, w \in \mathbb{R}^{nK}} \sum_i \mathbf{ReLU}\left(\left\{ \sum_{k=1}^{K} \sigma(w_{ik}) \|\theta_k - \theta_i\|_\infty \right\} - \epsilon \right)$$

softmax

find a set as well as an assignment matrix w; so that each θ is (softly) assigned to a member that covers it

Theorem (informal): If the cover exists, the sets of optimal solutions to (1) and (2) are the same.

Training: Zero-Shot Guarantee

Theorem (informal): If tasks have shared dynamics and their reward functions are Lipschitz w.r.t. their embedding parameters,
with small sample complexity our framework guarantees a $(\epsilon, 1-\delta)$ cover for Γ.

e.g. robotic control with different target speed/directions; individualized glucose control levels in metabolic models…

Training: Few-Shot Selection

Theorem (informal): Given a new, unknown task from the distribution Γ, with sample complexity depending only on K, not the dimension/size of the state or action space, we could identify the policy in the committee that is near-optimal for it.

Comparisons

	Objectives	Limitations
Multi-Task RL	Learn policies (one neural network) for all seen tasks	Suffers **negative transfer** when task diversity is high
Meta RL	Learn to adapt to unseen tasks	Requires **high sample complexity** for training
Personalized RL	Cluster and assign policies	Assumes shared dynamics; lacks performance guarantees; no meta/multi-task reuse
PACMAN (Ours)	***Learn a policy committee with generalization guarantees***	***Requires the tasks to be parametric; GIA's computational complexity is exponential in d***

Practice

- *Is there really no hope if the tasks are not obviously parametric?*

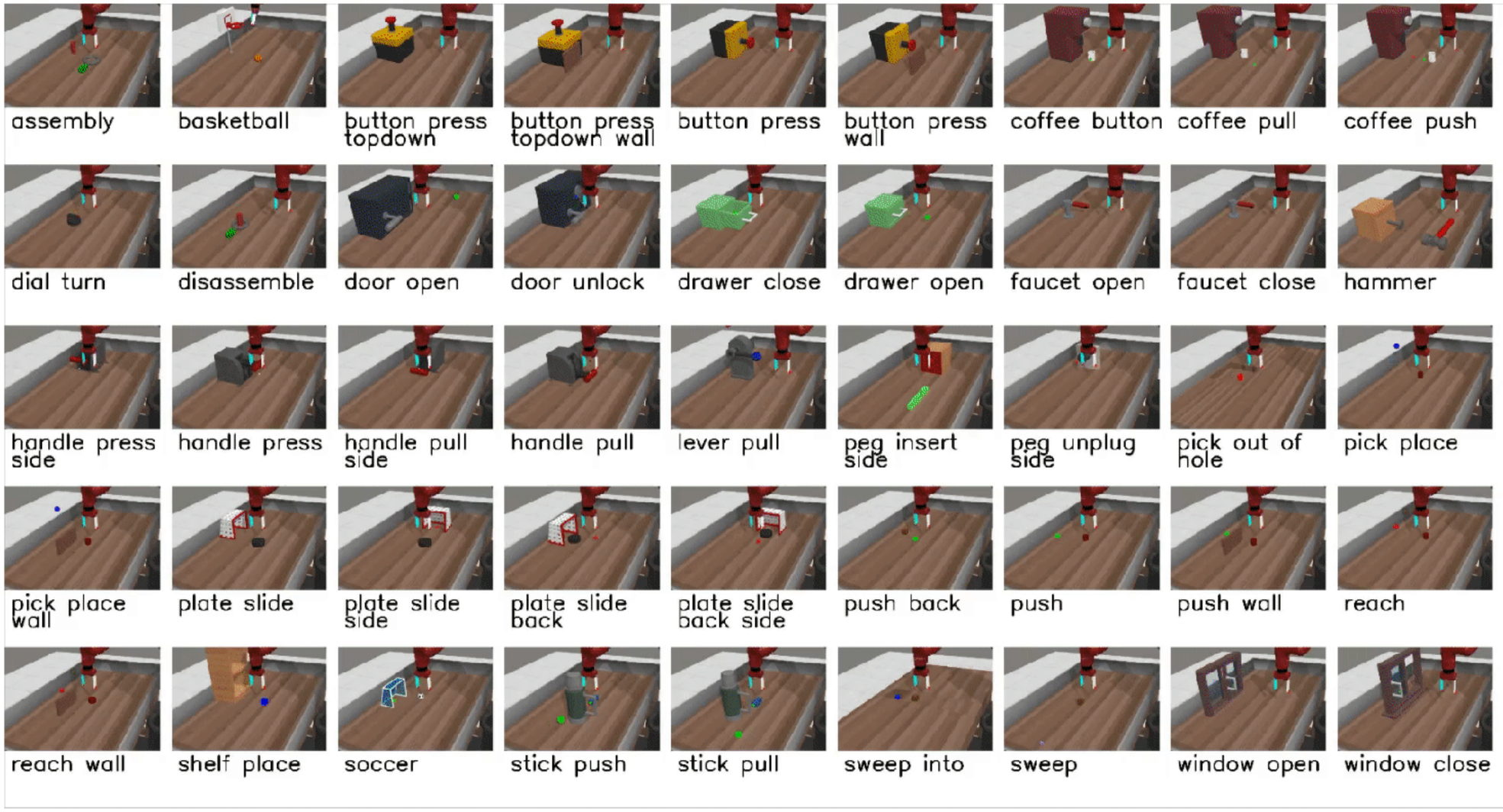
Let’s try generating descriptions of the tasks using LLMs and taking the features of penultimate layer as a parameterization for each task.

- Can we leverage the state-of-art Multi-task/Meta RL algorithms for better generalization?

Why not?

Experiments: Meta-World

Train


assembly
basketball
button press topdown
button press topdown wall
button press
button press wall
coffee button
coffee pull
coffee push
dial turn
disassemble
door open
door unlock
drawer close
drawer open
faucet open
faucet close
hammer
handle press side
handle press
handle pull side
handle pull
lever pull
peg insert side
peg unplug side
pick out of hole
pick place
pick place wall
plate slide
plate slide side
plate slide back
plate slide back side
push back
push
push wall
reach
reach wall
shelf place
soccer
stick push
stick pull
sweep into
sweep
window open
window close

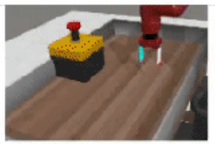
Test

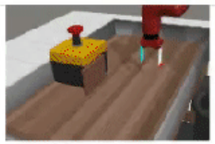

bin picking
box close
door lock
door unlock
hand insert

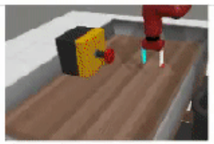
Train

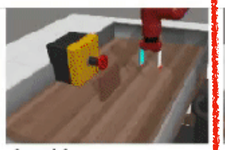

assembly


basketball

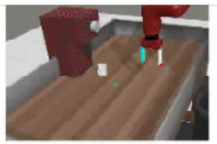

button press topdown

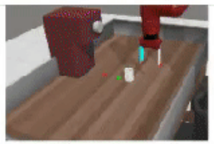

button press topdown wall


button press


button press wall

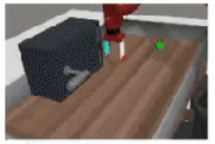

coffee button

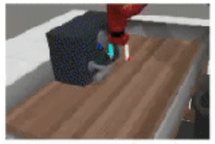

coffee pull

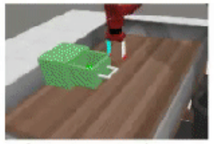

coffee push

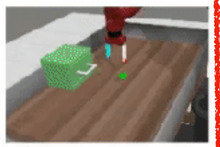

dial turn

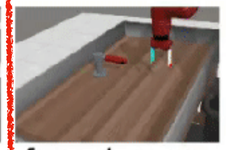

disassemble

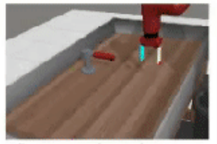

door open

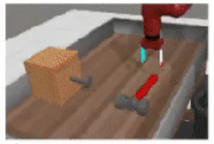

door unlock

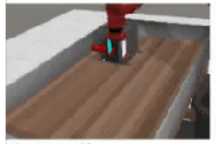

drawer close

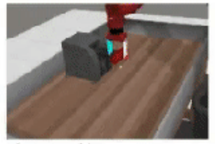

drawer open


faucet open


faucet close

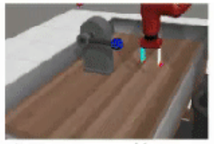

hammer

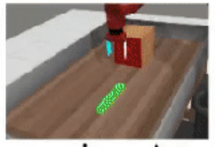

handle press side


handle press

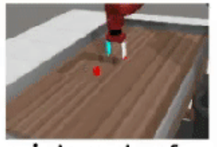

handle pull side

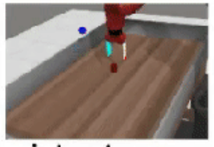

handle pull

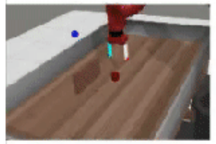

lever pull


peg insert side


peg unplug side


pick out of hole


pick place


pick place wall

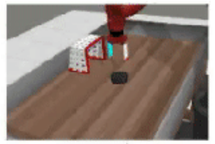


plate slide

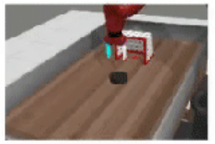


plate slide side

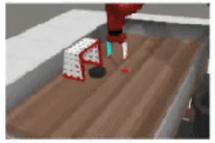


plate slide back

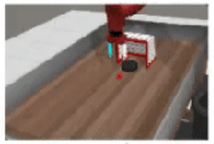
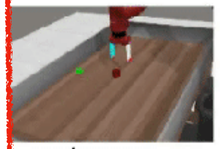
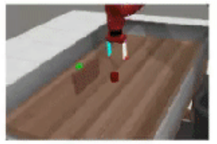
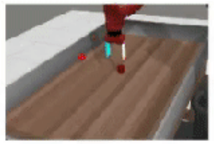


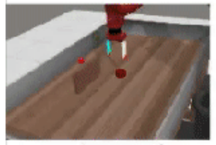
plate slide back side


push back

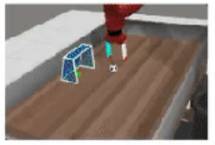

push


push wall


reach


reach wall


shelf place


soccer


stick push

stick pull

sweep into

sweep

window open

window close

Test

bin picking

box close

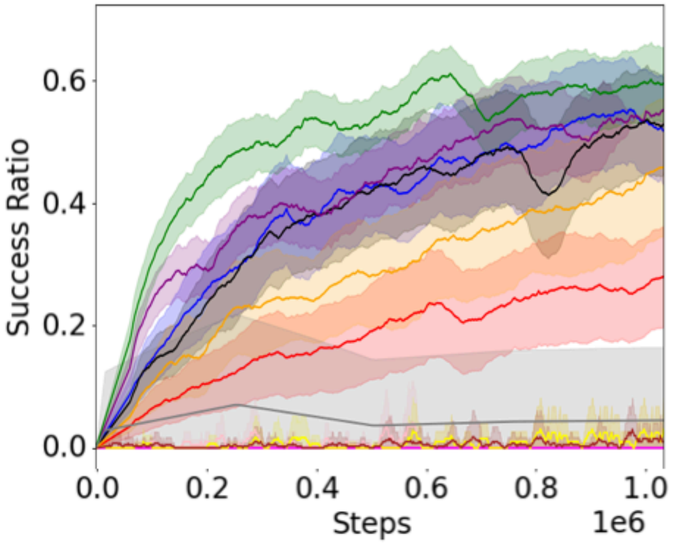
door lock

door unlock

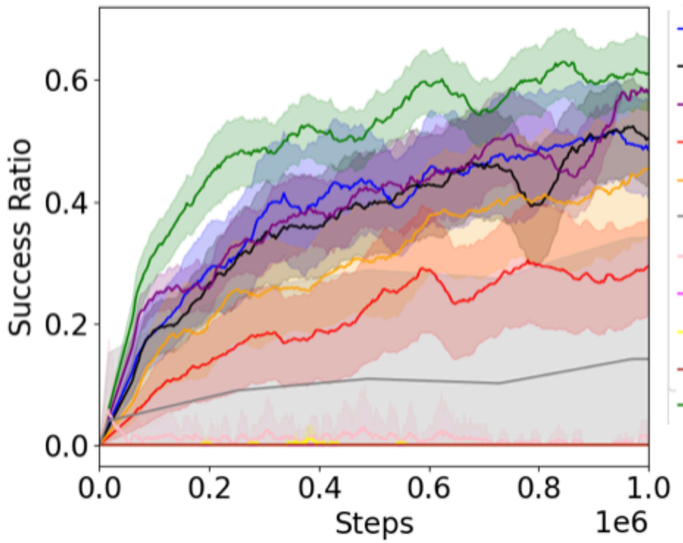
hand insert

30 vs. 20

PACMAN achieves best zero-shot and few-shot performance, with up to 36% gain over best baseline (Meta-World, 12K updates).



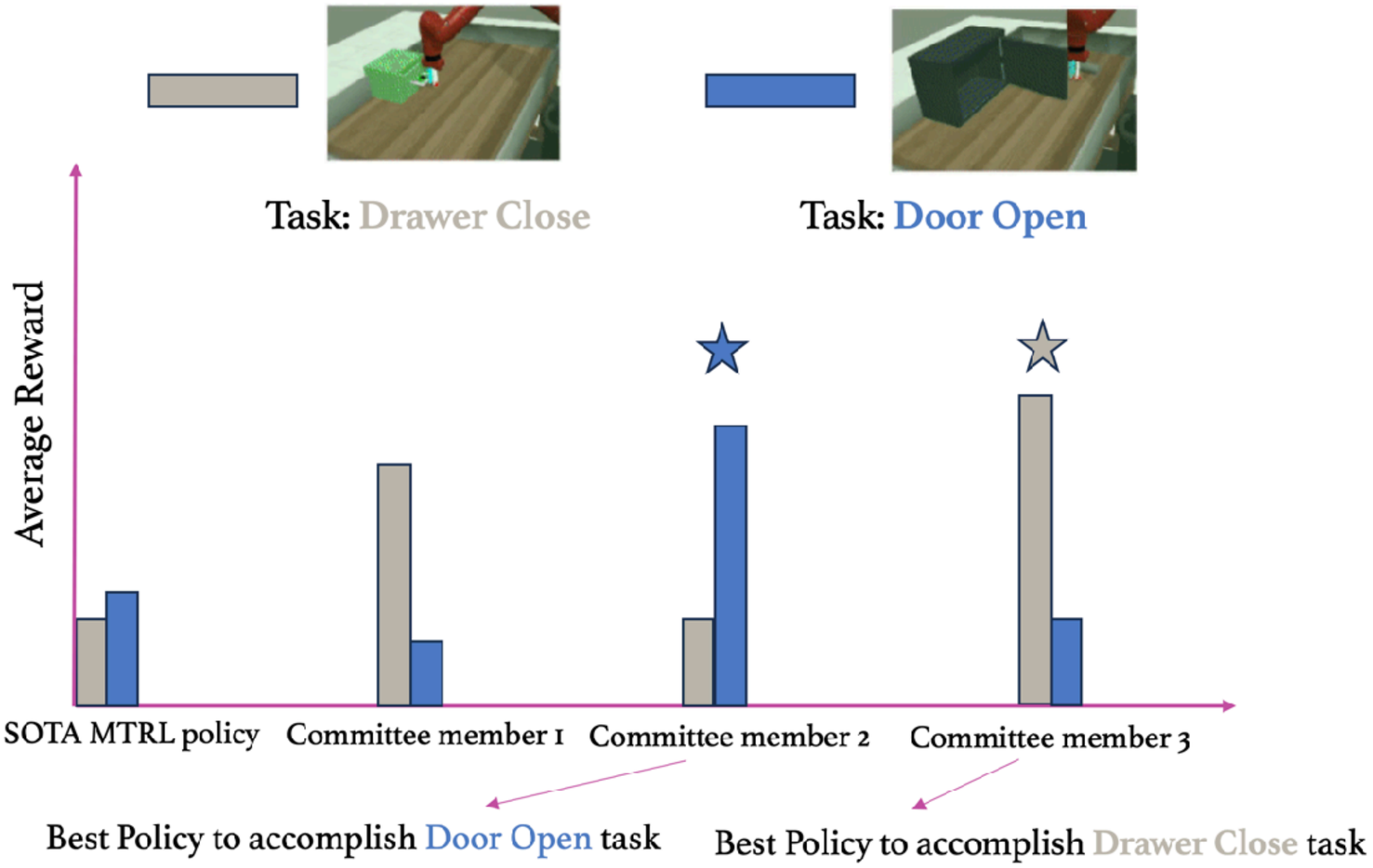
Train



Zero Shot

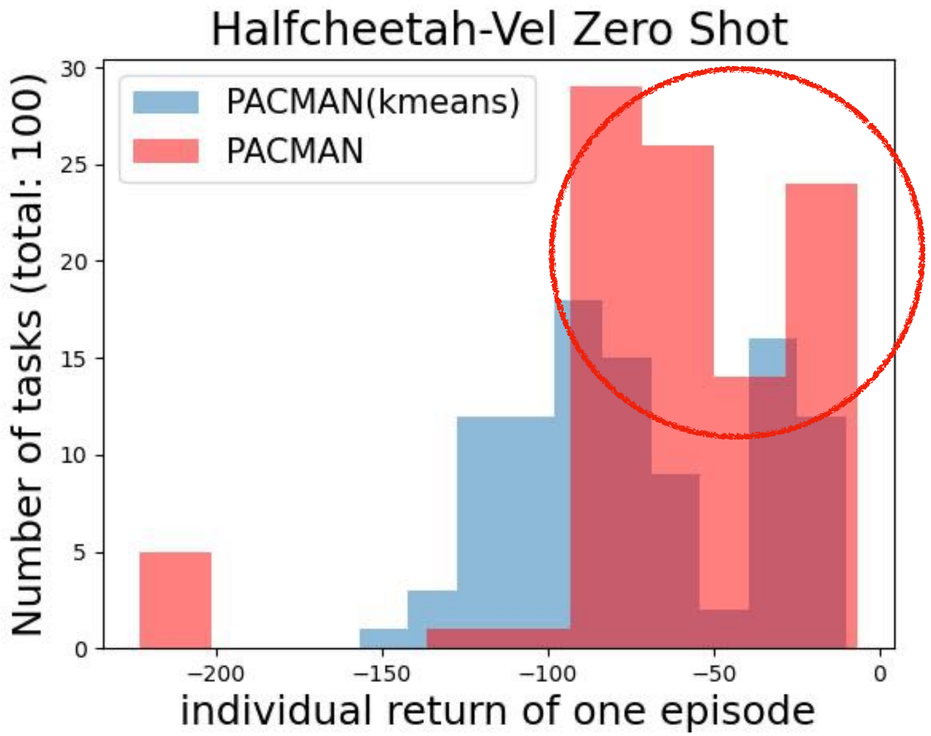
Method	6K Updates	12K Updates
MAML	0.0025 ± 0.006	0.01 ± 0.03
PEARL	0.03 ± 0.03	0.27 ± 0.07
RL2	0.007 ± 0.01	0.02 ± 0.02
VariBAD	0.025 ± 0.06	0.027 ± 0.07
AMAGO	0.08 ± 0.09	$.093 \pm 0.09$
Soft	0.27 ± 0.07	0.26 ± 0.08
MTTE	0.37 ± 0.08	0.40 ± 0.10
CARE	0.39 ± 0.05	0.40 ± 0.06
CMTA	0.45 ± 0.07	0.34 ± 0.08
MOORE	0.41 ± 0.08	0.44 ± 0.11
PACMAN	$\mathbf{0.53 \pm 0.02}$	$\mathbf{0.60 \pm 0.02}$

Few Shot



Each commitee member specializes

Ablation 1: Why you shouldn't use off-the-shelf clustering methods



Meta-World, Zero-Shot

Method	125K Steps	250K Steps
KMeans++	0.22 ± 0.06	0.30 ± 0.07
DBScan	0.19 ± 0.04	0.28 ± 0.06
GMM	0.29 ± 0.07	0.33 ± 0.08
Random	0.22 ± 0.04	0.25 ± 0.04
PACMAN (Ours)	$\mathbf{0.36 \pm 0.07}$	$\mathbf{0.48 \pm 0.09}$

HalfCheetah-Velocity, Zero-Shot

Method	6 Million Frames	12 Million Frames
KMeans++	-105.91 ± 3.44	-89.50 ± 3.10
DBScan	-234.05 ± 9.56	-213.42 ± 4.09
GMM	-239.32 ± 11.27	-199.86 ± 9.54
Random	-274.13 ± 16.76	-258.07 ± 13.62
PACMAN	$\mathbf{-97.42 \pm 3.70}$	$\mathbf{-74.20 \pm 6.76}$

Failure to promote near-optimality/isolate outlier tasks

Ablation 2: Different Ks and Different Epsilons

Increasing K isn't always good;
PACMAN is robust to ε.

Table 5. Few-shot in Meta-World, varying K.

Method	**6K Updates**	**12K Updates**
MOORE	0.42 ± 0.06	0.43 ± 0.05
PACMAN ($K = 1$)	0.32 ± 0.05	0.31 ± 0.04
PACMAN ($K = 2$)	0.50 ± 0.05	0.50 ± 0.05
PACMAN ($K = 3$)	0.61 ± 0.04	0.62 ± 0.05
PACMAN ($K = 4$)	0.32 ± 0.05	0.35 ± 0.05

	$\epsilon = .5$	$\epsilon = .6$	$\epsilon = .7$	$\epsilon = .8$
500K Steps	0.23 ± 0.08	0.54 ± 0.08	0.56 ± 0.09	0.60 ± 0.07
1M Steps	0.25 ± 0.07	0.60 ± 0.10	0.58 ± 0.14	0.58 ± 0.07

Table 7. Success rate for $K = 3$.

PACMAN: A New Principle for Personalization

Gave a better interpretation of personalization: near-optimality guarantee for the individuals

Developed principled framework with generalization guarantees with respect to a task distribution

Simple training mechanism that has been shown to work

- 1s clustering vs 40h training (Meta-World); parallel training; leveraging SOTA algorithms
- One tunable hyperparameter (ε).
- Works with LLM embeddings for non-parametric tasks.

Outlook:

- Extending the framework to other personalization domains
- Adaptive ε
- …

Thank you!