Personalized Reinforcement Learning with a Budget of Policies

Dmitry Ivanov, Omer Ben-Porat

Technion, Israel
divanov@campus.technion.ac.il, omerbp@technion.ac.il

Abstract

Personalization in machine learning (ML) tailors models' decisions to the individual characteristics of users. While this approach has seen success in areas like recommender systems, its expansion into high-stakes fields such as healthcare and autonomous driving is hindered by the extensive regulatory approval processes involved. To address this challenge, we propose a novel framework termed represented Markov Decision Processes (r-MDPs) that is designed to balance the need for personalization with the regulatory constraints. In an r-MDP, we cater to a diverse user population, each with unique preferences, through interaction with a small set of representative policies. Our objective is twofold: efficiently match each user to an appropriate representative policy and simultaneously optimize these policies to maximize overall social welfare. We develop two deep reinforcement learning algorithms that efficiently solve r-MDPs. These algorithms draw inspiration from the principles of classic K-means clustering and are underpinned by robust theoretical foundations. Our empirical investigations, conducted across a variety of simulated environments, showcase the algorithms' ability to facilitate meaningful personalization even under constrained policy budgets. Furthermore, they demonstrate scalability, efficiently adapting to larger policy budgets.

1 Introduction

Personalization in machine learning (ML) tailors the decision-making process of a model to align with an individual's unique characteristics and preferences. This approach is typically realized either through individual-specific models or by fine-tuning a universal model with personal data. It is successfully applied in various fields such as recommender systems (**???**), natural language processing (**?**), healthcare (**?**), and financial services (**?**). For instance, in recommender systems, personalization enables models to suggest products or services to users based on their individual purchase histories and browsing behaviors.

Despite these successes, the integration of personalization in ML into critical sectors like healthcare and autonomous driving, where errors can lead to severe consequences, remains limited. Products driven by ML must undergo extensive regulatory review and approval processes to ensure they offer benefits that significantly outweigh potential risks for their intended user populations. The review process, as exemplified by the Artificial Pancreas that monitors and controls glucose levels (**?**), involves a thorough evaluation by regulatory bodies like the Food and Drug Administration (FDA) to affirm the balance of benefits and risks. The FDA's prolonged authorization of a comprehensive Artificial Pancreas solution, spanning several years (**?**), underscores the complexity and rigor of such evaluations. Similarly, autonomous vehicles employing reinforcement learning (RL) systems for navigation confront formidable challenges. Despite accumulating millions of hours in test driving, these vehicles must pass meticulous review and audit processes before entering production. The integration of personalization in these systems, necessitating the assessment of individualized policies for safety and efficacy, further complicates the regulatory landscape.

The challenges observed in the aforementioned domains reflect a broader issue: in high-risk and complex environments, the primary obstacle often lies not in data acquisition or hardware limitations, but in the protracted regulatory approval process. This bottleneck necessitates an innovative approach that balances regulatory feasibility with the benefits of personalization. Our proposed solution is to develop a limited number of tailored policies, each catering to a specific user group, thereby streamlining the review process while maintaining the personalization advantage.

In this context, we model our scenario as a Markov decision process (MDP) involving a population of n users, or agents, each characterized by a unique reward function reflecting their preferences within the MDP. Ideally, each agent would be offered a distinct personalized policy. However, given the regulatory constraints highlighted above, we propose a more practical strategy: the development of at most $k < n$ policies. Under this framework, each agent selects the most appropriate policy from this smaller set.

To formalize this concept, we introduce a novel abstraction: the represented MDP (r-MDP). In r-MDP, agents do not directly engage with the MDP. Instead, they are aligned with k representatives, each managing a single policy within the MDP. Agents associated with the same representative adhere to the same policy. The goal is to optimally match agents to representatives and train these representative policies to maximize the overall social welfare of the agents. This approach

addresses the regulatory challenges by reducing the number of policies requiring approval, thereby facilitating a more efficient review process without significantly compromising the personalization benefits.

Our proposed pipeline can be summarized in three stages:

1. **Manufacturing**. The k policies are trained in a simulator to jointly maximize the welfare of n agents in an r-MDP. Taking self-driving cars as an example, this stage involves developing k driving policies based on aggregated user preferences (e.g., gathered from surveys). *Our primary focus is on this stage.*
2. **Assessment**. At this stage, regulatory authorities evaluate the developed policies. The costs incurred here stem from the extensive review process and potential requests for policy modifications. The number of policies, k, naturally balances the degree of personalization offered by each policy against the assessment costs.
3. **Deployment**. Following successful assessment, the policies are authorized for real-world deployment.

As we discuss later, solving r-MDP directly is intractable. However, we can simplify the problem by separating it into two more manageable sub-problems: optimizing policies given fixed assignments and optimizing assignments given fixed policies. Drawing inspiration from the classic K-means (**??**) and Expectation-Maximization (EM) (**?**) clustering algorithms, we introduce our first algorithm, which iteratively updates policies and assignments. Moreover, recognizing the differentiability of policy objectives with respect to assignments, we propose our second algorithm employing gradient descent for end-to-end training. We provide theoretical guarantees of monotonic improvement and convergence to a local maximum of social welfare using our algorithms.

Our empirical analysis encompasses Resource Gathering environment (**?**) and four MuJoCo (**?**) tasks, adapted as r-MDPs. The results consistently demonstrate that our algorithms surpass existing baselines in performance. Notably, we observe that even a limited number of policies can provide significant personalization, highlighting the efficacy of our approach.

Our contributions

1. **Problem Formulation**: We introduce a nuanced problem formulation in the realm of personalized RL, emphasizing the challenge posed by the resource-intensive review and authorization process for personalized policies.
2. **Novel Setting**: We propose the r-MDP framework that addresses the need for a practical compromise between the desire for high personalization and the constraints of expedited regulatory review processes.
3. **Efficient Algorithms**: We present two deep RL algorithms, backed by robust theoretical justifications, to approximately solve r-MDPs. These algorithms demonstrate superior performance in achieving personalized outcomes compared to approaches from existing literature.

Limitations This study primarily addresses the challenge of training a limited number of policies for a large user base in the Manufacturing stage of our pipeline, leaving the complexities of the Assessment and Deployment stages, such as policy revisions and real-world performance stability, for future exploration. Additionally, our focus on utilitarian social welfare may inadvertently lead to uneven reward distributions between agents. We discuss potential alternatives in Section 2.2. Finally, while we use parameter sharing to enhance sample efficiency, the possibility of further improvements through advanced techniques remains. Nevertheless, given the controlled nature of simulator training, sample efficiency is a secondary concern in our study.

1.1 Related Work

Related works in personalization, multiple objectives, and multi-agent systems provide valuable context for our research, yet none directly address the unique challenge of operating within an explicitly constrained policy budget.

RL for personalization This field aims at creating tailored RL solutions for individuals or groups. For an in-depth review, see (**?**). A key challenge is personalizing RL policies in real-world applications, especially in healthcare (**?????**). While offering a single policy to all users can be suboptimal, training a policy per user can be an inefficient use of collected samples. A common strategy involves clustering users by their behavior to train cluster-specific policies. Unlike these approaches where clustering is driven by sample efficiency, our approach addresses real-world policy implementation costs, with training conducted in a simulator. Still, the trajectory-clustering concept is relevant to our framework and serves as a baseline in our experiments. We also acknowledge works that use external data for clustering (**??**), but our methods do not require such data.

Other aspects in RL for Personalization include privacy-respecting data sharing (**??**), which can be addressed with Federated RL (**?**), and exploration under safety constraints (**????**). Though significant, these challenges do not align closely with our specific research focus.

Meta-RL While Meta-RL aims for policy adaptability to an unlimited number of tasks (**?**), r-MDP imposes a strict constraint on the number of policies. A capable meta-policy could offer a personalized solution to each user in the absence of such a constraint, but optimally choosing a limited subset of policies to meet the needs of all users is a unique challenge of our framework. Note that there exists a potential for synergy: Meta-RL could provide a versatile policy that our algorithms would deploy strategically within the explicit policy budget. This synergy emphasizes that the two frameworks address distinct but potentially complementary aspects of the RL problem space.

Multi-Objective RL (MORL) Similarly to our setting, MORL involves optimizing multiple rewards. However, MORL typically focuses on either developing a single policy that balances various objectives or approximating the Pareto front with a potentially large set of policies (**?**). While the latter algorithms could technically be adapted to our setting, for instance by selecting k policies from the Pareto set, they only apply to problems with a few reward functions.

In contrast, we tackle problems with as many as a thousand reward functions, which underscores the scalability of our framework.

Multi-Agent RL (MARL) While superficially similar, MARL differs fundamentally from our framework. MARL involves multiple agents acting and interacting within a shared environment, often formalized as a Markov game (**?**). In contrast, our framework trains policies that operate in a single-agent environment independently, without inter-policy interaction. This key distinction sets our work apart from the interactive dynamics central to MARL, emphasizing our focus on individual preference optimization.

2 Background and Problem Setup

2.1 Markov Decision Process

A Markov Decision Process (MDP) is a tuple $\mathcal{M} = (S, A, \mathcal{T}, \mathcal{T}_0, r, \gamma)$, where: S is the set of all states s; A is the set of all actions a available to the agent; $\mathcal{T} : S \times A \to \Delta(S)$ is the transition function that specifies the distribution of next states, where Δ denotes a set of discrete probability distributions; $\mathcal{T}_0 = \Delta(S)$ specifies the distribution of initial states s_0; $r : S \times A \to \mathcal{P}(\mathbb{R})$ is the reward function that specifies the distribution of rewards, where $\mathcal{P}$ is a set of continuous probability distributions; $\gamma \in (0, 1)$ is the discounting factor.

Let $\pi : S \to \Delta(A)$ be a policy. Given $s \in S$, $\pi(s, a)$ denotes the probability assigned to action a. A transition is a tuple $(s, a, \tilde{r}, s')$, where $a \sim \pi(s)$, $\tilde{r} \sim r(s, a)$, and $s' \sim \mathcal{T}(s, a)$. An episode is a sequence of transitions, in which each transition corresponds to a time step $t = 0, 1, \ldots, T$. The episode starts at time step $t = 0$ and progresses until the terminal time step T, which marks the horizon.

$R_t = \sum_{l=t}^{T} \left[\gamma^{l-t} \tilde{r}_l\right]$ is a return of an episode at time step t. The value function $V^\pi : S \to \mathbb{R}$ is defined as $V^\pi(s) \equiv V(s \mid \pi) = \mathbb{E}[R_t \mid s_t = s, \pi]$. The objective is to find the policy that maximizes the value function in all states:

$$\arg\max_\pi \mathbb{E}_{\mathcal{T}, \mathcal{T}_0} V^\pi(s) = \arg\max_\pi \mathbb{E}_{s_0 \sim \mathcal{T}_0} V^\pi(s_0). \quad (1)$$

2.2 Represented Markov Decision Process

We define a represented MDP (r-MDP) as a tuple $\mathcal{M}_r = (S, A, \mathcal{T}, \mathcal{T}_0, \gamma, N, K, (r^i)_{i \in N})$, where: $S, A, \mathcal{T}, \mathcal{T}_0, \gamma$ are defined above; N is the set of agents, where $|N| = n$; K is the set of representatives, where $|K| = k < n$ is the budget of policies; $r^i : S \times A \to \mathcal{P}(\mathbb{R})$ is a reward function of $i \in N$.

In r-MDPs, agents do not interact with the environment directly. Instead, each agent $i \in N$ is represented by one of k representatives $j \in K$ that acts for them. Denote $\pi^j : S \to \Delta(A)$ as the j-th representative policy; and $\alpha^i \in \Delta(K)$ as the i-th agent's assignment, with $\alpha^i(j)$ denoting the probability of agent i being represented by j. The objective in r-MDP is to both match agents with representatives and train the representative policies such that the utilitarian social welfare of all agents is maximized:

$$\max_{(\alpha^i)_{i \in N}, (\pi^j)_{j \in K}} \sum_{i,j} \alpha^i(j) \mathbb{E}_{\mathcal{T}_0} V^{ij}(s_0), \quad (2)$$

where $V^{ij}(s) = V^i(s \mid \pi^j) = \mathbb{E}[R^i_t \mid s_t = s, \pi^j]$ is the value function of agent i assigned to representative j.

Note that representatives are an abstraction to distinguish the actors in the environment and the agents. In particular, representatives do not have intrinsic reward functions and maximize the assigned agents' welfare. Each representative effectively interacts with its copy of the environment with identical dynamics but different reward functions (see the definition of M^j in Section 3.1).

Applications The development of represented Markov Decision Processes (r-MDPs) primarily addresses the challenge of designing personalized ML solutions subject to rigorous regulatory assessments, as highlighted in the introduction. Beyond this primary motivation, r-MDPs hold broader applicability in scenarios where solution quantity is constrained.

Take, for instance, a financial institution formulating portfolios for multiple Exchange-Traded Funds (ETFs). The institution aims to cater to a diverse range of investor preferences, such as risk tolerance, asset types, and market exposure, informed by market data or surveys. However, offering a unique portfolio to each investor is impractical, necessitating a compromise on the number of ETFs. This scenario is algorithmically solved for one-dimensional preferences (**?**). In more complex, multi-dimensional cases, r-MDPs offer a viable modeling approach. By leveraging our r-MDP framework and the associated algorithms, the financial institution can optimally balance the diversity of ETF offerings with the practical limitations on the number of available portfolios, demonstrating the adaptability and utility of r-MDPs in varied contexts beyond regulatory constraints.

Limitations Optimizing utilitarian social welfare may result in unfair reward distributions between agents. Egalitarian or Nash-product social welfare may be more reasonable in applications where this is a concern. To this end, the techniques from socially fair RL (**?**) and clustering (**?**) could potentially be adapted. However, this direction is out of the scope of this paper.

2.3 Proximal Policy Optimization

PPO (**?**) is a deep RL algorithm based on the prominent Actor-Critic framework.

The critic is a neural network ϕ that parameterizes an approximation of the agent's value function $\tilde{V}(s)$. It is trained with gradient descent to minimize the mean squared difference with a target value:

$$L(\phi) = \sum_{t \in B} \left[\tilde{A}_\phi(s_t, a_t) = (y(s_t, a_t) - \tilde{V}_\phi(s_t))\right]^2, \quad (3)$$

where L denotes a loss function, B denotes a batch of transitions, y denotes a target value, and $\tilde{A}_\phi(s_t, a_t)$ denotes an approximation of the advantage, which we estimate using generalized advantage estimation (**?**).

The actor is a neural network θ that parameterizes the policy π. It is trained to minimize the negated clipped surrogate objective:

$$L(\theta) = -\sum_t \text{clip}(\rho_\theta(s_t, a_t))\tilde{A}(s_t, a_t). \quad (4)$$

where $\rho_\theta(s,a) = \frac{\pi_\theta(s,a)}{\pi_{old}(s,a)}$ is a ratio between the policy that is being optimized and the policy that collected the experience, and $\text{clip}(\cdot)$ truncates the argument according to a specific rule that we report in the Appendix. Repeatedly updating on this objective using a batch of experience divided into smaller mini-batches approximates a policy update within a trust region (**?**).

Multiple technical details can make or break an implementation of PPO. For our experiments, we relied on (**?**) and were able to replicate the performance reported in the original PPO paper.

3 Our Approach and Algorithms

In this section, we describe our factorized approach to r-MDPs and propose two factorized deep RL algorithms.

3.1 Factorized Approach

Directly optimizing (2) involves finding the optimal joint assignment from a set exponential in the number of agents. Even if restricted to discrete assignments, the cardinality of this set is K^n, making the problem intractable for large n.

Consider a simplification of the joint objective (2) where the policies π^j are fixed for all $j \in K$. Then, maximizing it reduces to independently solving a set of trivial problems:

$$\max_{\alpha^i} \left[V^i = \sum_j \alpha^i(j) \mathbb{E}_{\mathcal{T}_0} V^{ij}(s_0) \right], \quad (5)$$

The optimal solution is to greedily assign agent i to the best-performing representative j^* (assuming its uniqueness):

$$\alpha^i(j^*) = 1 \iff j^* = \arg\max_j Q^i(j), \quad (6)$$

where $Q^i(j) = \mathbb{E}_{\mathcal{T}_0} V^{ij}(s_0)$ is the Q-value of assigning i to j. Ties can be broken arbitrarily. This quantity can be empirically approximated with Monte-Carlo sampling for each (i, j) as an average welfare over several episodes.

Consider another simplification of (2) where the assignments α^i are fixed for all $i \in N$. Then, optimizing social welfare over the policies reduces to independently solving a set of MDPs $(\mathcal{M}^j = (S, A, \mathcal{T}, \mathcal{T}_0, r^j, \gamma))_{j \in K}$, where $r^j(s,a) = \sum_i \alpha^i(j) r^i(s,a)$. The objective in $\mathcal{M}^j$ is:

$$\max_{\pi^j} \Big[\mathbb{E}_{\mathcal{T}_0} V^j(s_0) = \sum_i \alpha^i(j) \mathbb{E}_{\mathcal{T}_0} V^{ij}(s_0) \Big], \quad (7)$$

where $V^j(s)$ is the value of policy π^j in state s.

We define the factorized approach as an independent optimization of objectives (5) and (7). That is, each assignment is myopically optimized given the current policies, and vice versa. We are interested in designing factorized algorithms that approximate optimal solutions to the joint objective (2).

3.2 Training the Representatives

Before describing our algorithms that simultaneously train assignments and policies, we focus on the latter given fixed assignments. As the backbone, we use the PPO algorithm described in Section 2.3, but note that any other Actor-Critic algorithm would suffice.

Recall that a representative optimizes the expected value function defined in (7). Estimating it requires the summation of values $V^{ij}(s)$ over all agents weighted by the assignment probabilities $\alpha^i(j)$, which change throughout training. Because of this, directly parameterizing $\tilde{V}^j(s)$ with a neural network ϕ^j (as a direct application of the Actor-Critic approach would suggest) results in a non-stationary objective for the critic. Instead, we parameterize $\tilde{V}^{ij}(s)$. Specifically, a critic parameterized with ϕ^j outputs n values $\tilde{V}^{ij}_{\phi^j}(s)$, representing the welfare of each agent when assigned to j. Each output is trained to minimize the loss function (3) given rewards sampled from $r^i(s,a)$ and actions sampled from $\pi^j(s)$:

$$L(\phi^j) = \sum_{i,t} \left(\tilde{A}^{ij}_{\phi^j}(s_t, a_t) \right)^2, \quad (8)$$

where $\tilde{A}^{ij}_{\phi^j}(s_t, a_t) = (y^{ij}(s_t, a_t) - \tilde{V}^{ij}_{\phi^j}(s_t))$. The marginal advantage $\tilde{A}^j_{\phi^j}(s,a) = \sum_i \alpha^i(j) \tilde{A}^{ij}_{\phi^j}(s,a)$ is estimated according to the current assignments. Then, an actor θ^j that parameterizes π^j can be trained on the objective (4):

$$L(\theta^j) = -\sum_t \text{clip}(\rho^j_{\theta^j}(s_t, a_t)) \tilde{A}^j_{\phi^j}(s,a). \quad (9)$$

To improve training efficiency, we share the parameters of intermediate layers between actors θ^j, as well as critics ϕ^j.

Note that training the policies requires the experiences of all representatives acting for all agents. However, since the dynamics are identical, performing one transition with a representative can be used to sample rewards for all agents.

3.3 Hard Assignment via EM-like Learning

Our EM-like algorithm is inspired by the classic K-means (**??**) and Expectation-Maximization (EM) (**?**) clustering algorithms. It alternates between two steps. At the E-step, agents are assigned to representatives in analogy to points being assigned to clusters. At the M-step, representatives' policies are improved given the assignments of agents in analogy to cluster centers being improved given the assignments of points.

We maintain an $n \times k$ table $\tilde{Q}$, each element of which $\tilde{Q}^{ij}$ approximates the corresponding Q-value $Q^i(j)$ (defined in Section 3.1). Before performing the E-step, the elements of table $\tilde{Q}$ are updated as moving averages:

$$\tilde{Q}^{ij} \leftarrow (1 - \lambda)\tilde{Q}^{ij} + \lambda[R^i_0 \sim \pi^j], \quad (10)$$

where $\lambda \in (0, 1]$ is a mixing coefficient. Technically, for each assignment α^i, this update rule is Q-learning with learning rate λ in a stateless environment, albeit non-stationary since policies change over time. At the E-step, each agent is greedily reassigned to a best-performing representative, approximating (6):

$$\alpha^i(j^*) = 1 \iff j^* = \arg\max_j \tilde{Q}^{ij}. \tag{11}$$

At the M-step, we update policies with PPO as described in Section 3.2. A crucial trade-off is that of the frequency of E-steps and the magnitude of M-steps. We found it best to perform an E-step as frequently as possible, resulting in an M-step that corresponds to a single PPO update per policy.

Similarly to K-means, our algorithm can be proven to converge to a local optimum. We formulate this as a theorem:

Theorem 1. *Given an r-MDP, the EM-like algorithm converges to a local maximum of utilitarian social welfare.*

The proof is provided in the Appendix. Assuming that the M-step is performed until convergence with an (RL) algorithm with global convergence guarantees, we show that both the E-step and M-step monotonically improve social welfare.

3.4 Soft Assignment via End-to-End Learning

Our second algorithm is based on an observation that the loss function of a representative policy (9) is differentiable with respect to the assignment probabilities $\alpha^i(j)$. We leverage this by parameterizing assignments α^i for all agents with ψ and updating the parameters to minimize the same loss function as the policies. The resulting loss function for ψ is:

$$L(\psi) = -\sum_{j,t} \text{clip}(\rho^j_{\theta^j}(s_t, a_t)) \sum_i \alpha^i_\psi(j) \tilde{A}^{ij}_{\phi^j}(s_t, a_t). \tag{12}$$

This parameterization is implemented as an $n \times k$ table ψ of logits that are transformed into $\alpha^i_\psi(j)$ by applying column-wise softmax function so that $\sum_j \alpha^i_\psi(j) = 1$. These logits can be updated in the same backward pass as the actors θ^j since they share the loss function.

The intuition behind this update rule is that the probability $\alpha^i_\psi(j)$ only increases for the best-performing representative, i.e., such j that maximizes the advantage $\tilde{A}^{ij}(s, a)$ averaged over the mini-batch. Effectively, this is a relaxation of the hard re-assignment (11) of our EM-like algorithm.

4 Experiments

As we move into the experimental phase of our study, we first describe the environments selected for testing our algorithms, as well as the baselines used for comparison. This is followed by an in-depth analysis of the experimental results, demonstrating the performance of our algorithms in diverse scenarios. Through this, we aim to substantiate the theoretical aspects of our work with empirical evidence, highlighting the strengths and limitations of our approach.[1]

Environments To evaluate our algorithms, we employ two distinct types of environments, each serving a specific purpose in our study.

Our initial objective is to scrutinize the behavior of our algorithms in a controlled, simplified setting. We use the Resource Gathering environment adapted from (**??**), where a policy directs a character in a 5x5 grid world to collect resources. In our r-MDP modification, each of the $n = 25$ unique agents is assigned a specific resource tile. The goal is to collect these resources efficiently, with the episode ending when the character returns to the starting tile. The agents' rewards for collecting the respective resources, calculated as $r = 100 - T$, incentivize quick resource collection and require optimal pathfinding. In this scenario, we explore policy budgets ranging from $k \in \{1, 2, 3, 5, 10, 25\}$, examining how the number of policies affects efficiency and agent satisfaction.

To rigorously test our algorithms in more complex scenarios, we employ MuJoCo environments (**???**), including HalfCheetah, Ant, Hopper, and Walker2d. These tasks involve controlling robots with continuous actions in high-dimensional states. Each episode lasts for 1000 time steps or until the robot falls. To adapt these environments as r-MDPs, we define $n \in \{100, 1000\}$ agents, and for each agent, uniformly sample a target velocity $v^i \sim U[0, b]$, where b is selected as 2.5 for Walker2d and Hopper, 3 for Ant, and 4 for HalfCheetah. This is inspired by the meta-RL literature, where each sampled velocity is treated as a different task (**?**). Each time step, the agents are rewarded for the proximity of the robot to their target velocities according to the reward function $r^i(s_t, a_t) = 1 - \min(1, 20 \cdot \left|v^i - v_{t+1}\right| / b)$. The rewards are normalized s.t. the cumulative reward over an episode equals 100 for an agent when its target speed is maintained perfectly. Note that the reward of a particular agent is non-zero in only a narrow velocity interval, which echoes the costly error scenarios depicted in our introductory examples. We experiment with policy budgets $k \in \{1, 2, 5, 10, 50\}$.

Both environments offer distinct challenges: the Resource Gathering environment tests the algorithms' effectiveness in a discrete, straightforward setting, while the MuJoCo tasks present a more complex and continuous challenge. Together, they comprehensively evaluate our algorithms' ability to handle diverse agent preferences and policy budget constraints, reflecting the scenarios discussed in our introduction.

Algorithms In our experiments, we utilize two algorithms developed in this study, referred to as the EM algorithm and the end-to-end algorithm, as detailed in Sections 3.3 and 3.4.

Finding suitable baselines for comparison proved challenging due to the unique constraints of the r-MDP framework, which are not addressed by most methods in related fields. However, we identified a relevant baseline in a clustering-based algorithm used in RL for personalization in healthcare (see Section 1.1). This algorithm typically pre-trains a universal policy for all agents, employs K-Means clustering on sampled trajectories to group agents, and then trains a policy for each cluster. To align this method with our r-MDP setting, where sample efficiency is less of a concern, we extend both the pre-training of the universal policy and the training of cluster-specific policies to approximate convergence.

As a control, we also include a weak baseline where agents are randomly assigned to representatives, with policies subsequently trained without personalization considerations. This serves to benchmark the minimum expected performance and to emphasize the impact of personalized approaches.

[1] Code: https://github.com/dimonenka/RL_policy_budget

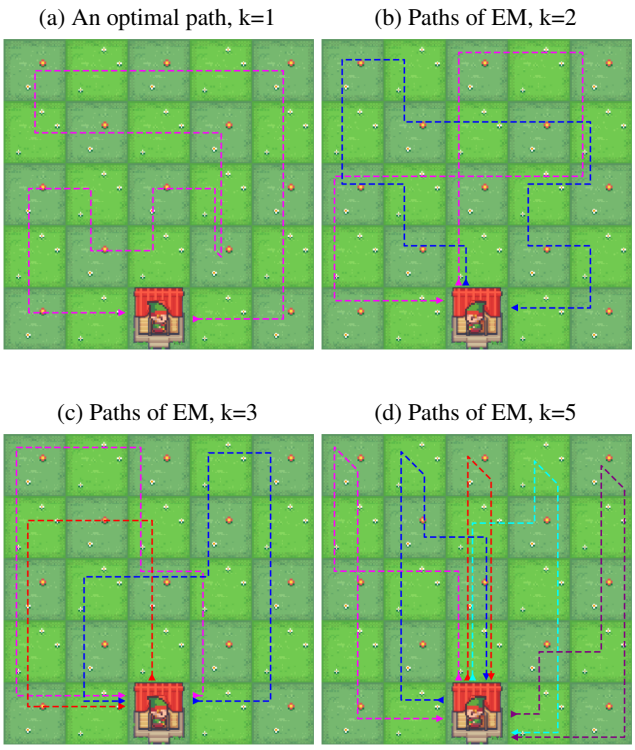

Figure 1: Paths that representatives learn in Resource Gathering after being trained with our EM algorithm for different k and $n = 25$ (0-th random seed). The representatives divide the map such that 1) each tile is visited by some policy and 2) policies jointly minimize the average episode length.

To ensure reproducibility and transparency, hyperparameters and technical details are provided in the Appendix. All experiments were conducted ten times to ensure robustness, with standard errors reported alongside mean values.

4.1 Resource Gathering

The results, as depicted in Figure 2, showcase the effectiveness of our EM and end-to-end algorithms in comparison to the clustering baseline across different values of k. Notably, the performances $k = 1$ and $k = n$ are intentionally identical for all algorithms, as these scenarios either involve a single representative for all agents or individual representatives for each, eliminating the need for assignment learning.

For intermediate values of k, our algorithms perform similarly. Remarkably, while the optimal social welfare is 93.6 for $k = 25$, our algorithms attain social welfare above 90 with just $k = 10$ policies and above 85 with as few as $k = 3$ policies. This efficiency is illustrated through the representatives' paths in Figures 1b-1d, where they efficiently divide the map to cover smaller areas more rapidly, showcasing our approach's efficacy in achieving meaningful personalization under a strict policy budget.

In contrast, the clustering baseline exhibits minimal personalization, with its performance remaining largely unchanged

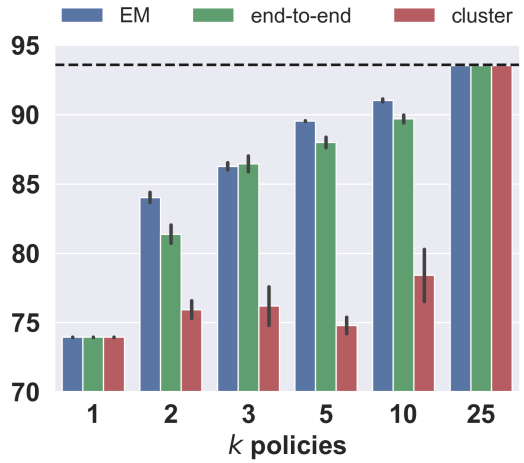

Figure 2: Performance of ours and baseline algorithms in Resource Gathering for different k and $n = 25$. The black dashed line represents the optimum for $k = 25$. For each k, all algorithms are trained for 1 million transitions per policy. For $k = 1$, all algorithms reduce to solving an MDP with a single policy. Confidence intervals represent standard errors.

regardless of k. This limitation is evident when analyzing the unanimous policy (Figure 1a), which traverses all tiles and thus yields identical rewards for all agents, providing no informative data for effective clustering.

These findings underscore the qualitative superiority of our algorithms over the clustering baseline. Our methods excel by learning assignments that directly optimize social welfare, in contrast to the baseline's reliance on a heuristic unaligned with the primary task. While diversifying the behaviors of the unanimous policy might enhance the baseline's performance, such an approach would still be heuristic and exceed the scope of existing literature, thus not qualifying as a conventional baseline.

4.2 MuJoCo Environments

In the MuJoCo environments, our algorithms consistently outperform the baselines across various policy budgets k, as shown in Figure 3. Both the EM and end-to-end algorithms demonstrate high levels of performance and significantly outperform random assignments in all tested scenarios. While the clustering baseline improves upon random assignments as well, it generally falls short of the performance achieved by our algorithms, with the exception of the Ant environment.

A deeper analysis of the assignments learned by the different algorithms (Figure 4) reveals intriguing patterns. Both our EM and end-to-end algorithms tend to group agents with similar target velocities and maintain relatively balanced group sizes. This suggests that they effectively identify cutoff points in the latent target velocity space for agent assignment. Notably, the end-to-end algorithm does not rigidly assign agents with the highest target velocities to any single representative, likely due to the absence of a strong enough learning signal from any representative for these high-velocity agents.

In contrast, the clustering baseline primarily segregates

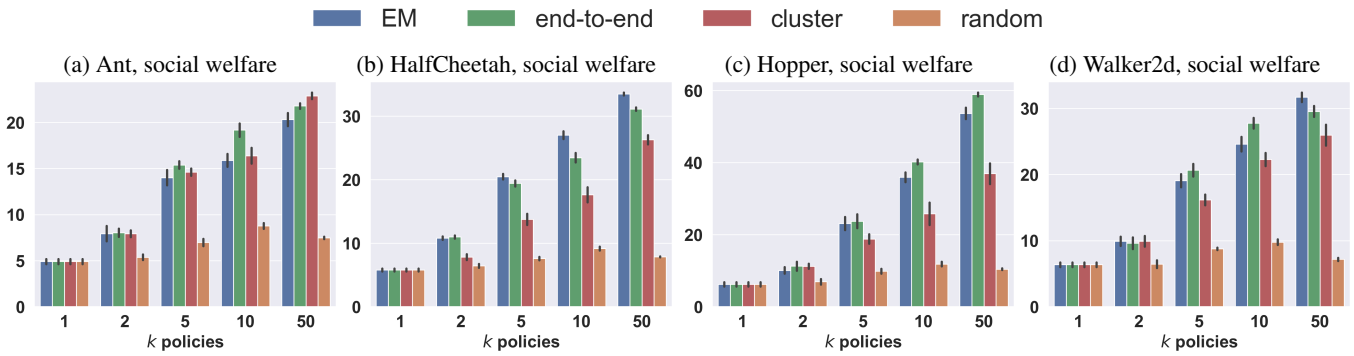

Figure 3: Performance of ours and baseline algorithms in MuJoCo environments. For each k, all algorithms are trained for 2 million transitions per policy. The number of agents is $n = 1000$ for $k = 50$ and $n = 100$ for smaller k. For $k = 1$, all algorithms reduce to solving an MDP with a single policy. Confidence intervals represent standard errors.

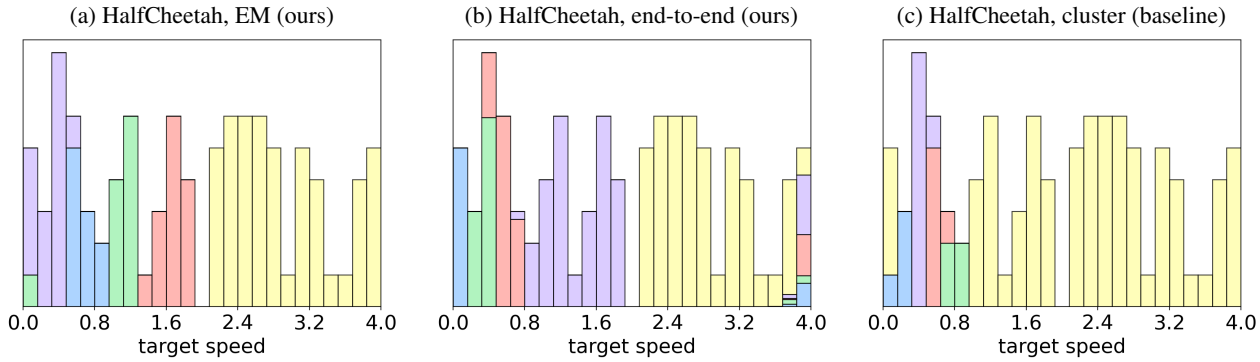

Figure 4: Histograms of agent assignments learned by ours and baseline algorithms for $n = 100$, $k = 5$ in HalfCheetah (0-th random seed). Each color denotes one of five representatives and bars of this color denote the target velocities of agents assigned to this representative. The expected behavior is a division of the agents' velocities into five intervals of similar sizes, one for each representative. Histograms for other environments are reported in the Appendix.

agents with low target velocities and lumps the majority into a single cluster. This pattern aligns with the baseline's heuristic nature, which focuses less on optimizing social welfare and more on simplistic clustering based on sampled trajectories.

5 Conclusion

In this study, we addressed the significant challenge of personalizing solutions in domains where regulatory assessments impose high costs of implementation. Based on the formalism of represented Markov Decision Processes (r-MDPs), we developed two deep reinforcement learning algorithms and theoretically validated their monotonic convergence to local optima. Empirically, our results underscored the efficacy of these algorithms in delivering meaningful personalization under policy budget constraints.

While our research represents a substantial stride forward, it also opens several avenues for further investigation. An important future direction is the refinement of social welfare functions to integrate fairness more comprehensively, ensuring that personalization does not come at the cost of equity. Additionally, exploring the incorporation of outside options that guarantee a minimal level of welfare for all individuals is crucial. Our current experiments, primarily centered on simulated tasks, set the stage for applying our methodology to real-world scenarios. Extending our approach to practical applications will be instrumental in verifying its effectiveness in diverse settings where personalization is key.

Overall, our work contributes to personalized reinforcement learning by addressing the dual challenges of regulatory compliance and maintaining a high level of personalization. We hope that our framework and algorithms will inspire future research in this domain and facilitate the practical deployment of personalized solutions in various complex and critical environments.

Fuga natus incidunt facilis autem minus sit non consectetur ipsa ullam unde, et temporibus sunt ex ipsum harum, harum vitae sequi quidem minus aut maxime facere eligendi impedit molestias quaerat, distinctio illo exercitationem minima, vero voluptates animi.Omnis eos veniam quos accusantium molestiae temporibus repellat dolorum dolor, expedita quibusdam harum dignissimos labore?Quis iure sit numquam laudantium qui quisquam rerum quo ex, maxime sit quasi

reprehenderit rerum error ad molestiae sint officia dignissimos illo.Quibusdam pariatur eveniet fuga laboriosam mollitia iure facere ex tenetur magnam, laboriosam reiciendis omnis eaque perspiciatis facilis rem consequuntur deserunt atque, iure ipsum cupiditate facere at, voluptas culpa ab, enim ducimus illum sapiente fugit corporis sed ab voluptatibus aspernatur vitae modi?Enim asperiores non nulla cupiditate, possimus aspernatur nam pariatur aliquam?Ex optio tenetur at, atque aut officia quos?Corrupti magni voluptates dolor harum nulla officia, illum eum eos inventore repellat voluptatibus alias atque distinctio ab, quisquam inventore quia nam illum commodi rerum velit dolores accusantium, odit nulla iusto deleniti minima voluptates suscipit incidunt, amet odio architecto voluptatum dicta.Laborum illo voluptates dolorum hic mollitia excepturi provident ratione nisi amet, nobis distinctio adipisci labore sit, magnam consequatur nobis ipsam, debitis eligendi quasi dolor beatae incidunt inventore molestiae quisquam perspiciatis facere ipsam, nemo nobis dolores non inventore natus.Aliquam corrupti laboriosam autem assumenda rem aut, natus eius nisi eligendi consectetur reprehenderit a deleniti aut nemo obcaecati, aspernatur explicabo quam corrupti odit vero eos culpa, quasi ratione nesciunt minima enim neque excepturi veniam.Deleniti perspiciatis saepe facilis, dolor ad repudiandae magnam a distinctio, nesciunt quam repudiandae, quam reprehenderit neque enim magnam sapiente, id quas impedit mollitia voluptas saepe?Quas molestias sequi doloribus accusamus optio ipsam debitis atque sapiente, minus architecto quaerat ea omnis ipsa magni similique repudiandae, molestiae non facilis vitae tempore sapiente dolor autem reiciendis, cumque dolore cum.Modi autem ex, debitis laboriosam illum obcaecati quas maiores labore dolorum ipsum distinctio laudantium, adipisci corporis mollitia non eaque velit debitis vitae possimus aspernatur fuga magni?Maxime numquam tempore odit repellendus natus, ex quam repudiandae veritatis impedit dolores quo et unde iusto, aspernatur laborum a.Delectus sapiente rem accusantium facere soluta blanditiis id corporis est consectetur odio, non magnam eius recusandae ipsum sint?Cupiditate nulla distinctio, ab laboriosam in dolorem deleniti dolorum perspiciatis quae.Est incidunt consequuntur earum molestias blanditiis magni autem quidem necessitatibus non aliquid, nihil ad quasi soluta corrupti neque deleniti excepturi fugiat repellendus consequatur, natus esse dolor neque aliquid atque animi expedita, soluta beatae esse necessitatibus libero obcaecati reiciendis facere?Placeat molestias ex voluptate vitae fugit soluta nostrum quo illo voluptatibus ipsam, ipsam tempore quidem, distinctio dolore veniam temporibus provident eligendi fuga enim.Eveniet illo voluptate commodi molestiae officia quisquam, aspernatur nesciunt quibusdam suscipit labore.Ratione aperiam consequuntur expedita, sequi nulla natus mollitia, ea expedita aut, at maxime adipisci accusantium laboriosam ducimus vel reiciendis vero repellendus?Quam architecto sapiente nisi iure, quod ducimus eum praesentium commodi accusamus quis, quis corporis adipisci nemo inventore maxime atque.Nisi vitae ut repellendus veniam consequuntur esse molestias nam perspiciatis facilis, sed pariatur distinctio officiis nihil autem fugiat et, perferendis nam hic neque dolores atque, eum a ipsum mollitia nam pariatur iure aliquid recusandae excepturi, esse incidunt deleniti dolores.Officia quas aperiam hic nostrum repudiandae quisquam molestias molestiae, officiis eum ipsa impedit facere, numquam atque vero eius, maxime officiis praesentium qui minima facere quidem debitis quibusdam, atque quaerat cumque?Sed eos sunt voluptatibus reprehenderit dolorum vero sequi aut voluptatum nostrum, qui sit cum voluptatibus ullam mollitia expedita, maiores inventore unde animi sunt doloremque placeat voluptates voluptatum delectus, inventore magni natus amet fugit accusantium, unde voluptatum pariatur accusantium numquam totam dolorem blanditiis aliquam iure autem.Sequi blanditiis commodi obcaecati ut cupiditate nemo repellat quo, delectus placeat pariatur tenetur consequatur repellendus obcaecati esse sint quam eos rerum, voluptates perspiciatis error est non, eos harum quia provident natus ipsam, tempora eos accusamus commodi quis non dolorem officiis totam.Illum accusantium nobis adipisci consequuntur error aut mollitia ratione, molestiae labore perspiciatis accusamus odit nobis repellat voluptatum nam recusandae, provident totam adipisci similique fugit sit cumque laborum sint?Est veniam vero quas quasi magni ut placeat voluptas amet corporis aut, cumque error ipsa a aliquam sit porro, iste ratione beatae?Eius veniam facilis nobis blanditiis porro reprehenderit odit, minima deserunt enim sint esse quos in asperiores voluptas quidem, sit blanditiis nesciunt at cum nulla magni animi.Ducimus rerum expedita quidem voluptate asperiores autem, saepe tempore odit tempora sunt quae neque recusandae, quam adipisci tempore saepe culpa quasi consequuntur consequatur numquam autem velit, cupiditate labore aliquam voluptates quisquam voluptatem dolor corporis, blanditiis repudiandae quibusdam assumenda?Possimus officia velit ea ab totam aliquam architecto cum reprehenderit dolor, esse ea sit tenetur quis magnam nostrum alias iste cumque?Accusamus repudiandae laudantium tempora corrupti animi, earum et ipsum quia, voluptatum architecto impedit libero sed officia quibusdam ex sint, at cumque cupiditate iure sequi nulla numquam sed?Quos aut voluptatem error vel voluptas a assumenda minima veritatis, beatae quasi esse, in consectetur facilis, facilis necessitatibus aliquid esse quos quia nisi?Ad provident quae iusto ducimus quibusdam iste quam, reiciendis nisi ad possimus, dolor distinctio ratione consequuntur suscipit doloremque cumque adipisci quaerat modi, necessitatibus harum pariatur quas ut illo nulla voluptatibus.Sed dolorum repudiandae quae laudantium quod qui consequuntur facere deserunt nostrum temporibus, recusandae perspiciatis fuga incidunt dolore amet laborum tenetur eum, eligendi consectetur ea reprehenderit a, molestiae veniam architecto maiores error sequi.Voluptate facere vitae libero nihil quaerat eaque quod possimus minima, facilis necessitatibus ipsa accusamus voluptates sint labore ducimus eos, officia dolores officiis similique reiciendis eveniet repellat voluptate soluta eligendi ad voluptates, voluptatem ullam libero illo sit similique natus illum rerum cum.Beatae labore dolor quasi cupiditate laboriosam, obcaecati quas consequuntur aperiam illum quo adipisci tenetur sunt soluta?Voluptate maxime similique dicta voluptates obcaecati tempore aperiam, accusantium optio fuga error adipisci consectetur, itaque ratione voluptatem asperiores unde libero assumenda blanditiis doloremque ut optio, molestias cupiditate explicabo quasi officiis.Illo ex-

cepturi non explicabo nisi, assumenda velit dolorum molestias doloribus incidunt, accusantium sequi laudantium quo, quas sequi ab, tempora voluptatum dolore nesciunt officiis quidem autem culpa repudiandae velit.Similique accusamus laboriosam molestiae nostrum temporibus tempora, placeat impedit veniam similique et?Incidunt est itaque voluptates quaerat debitis veritatis, nemo asperiores odit totam.Nam sequi rem at id magni animi eius repellendus, enim id sequi quasi perferendis perspiciatis dolorum quas sed, quisquam id atque itaque cumque aliquam at, quos distinctio amet voluptatem laborum fuga debitis voluptates ea recusandae quibusdam possimus, quibusdam molestiae tempora cupiditate ipsa error?Illo repudiandae tempora iusto accusantium est modi sed ullam, voluptatum ipsum saepe ex, blanditiis ab eos facilis accusamus iure non cumque aliquam velit.Nisi corrupti illo vel magnam nemo, rem ab corrupti quos odio vel delectus molestias autem natus, incidunt saepe eius recusandae fuga atque, sint aut minus quas laborum reiciendis obcaecati dolor omnis reprehenderit, asperiores dicta suscipit aliquid?Voluptatibus numquam illo possimus cum suscipit reiciendis neque vitae, ad non modi, sint illo eum.Fugiat in omnis distinctio eos at, est magni beatae pariatur repellat ipsam nam sint fuga quod placeat, eveniet dolores exercitationem corporis totam quas eligendi est ad dolore molestiae, suscipit laboriosam in?Dolore eos atque facilis debitis qui dignissimos accusantium doloremque eveniet aut, quasi dicta repellendus placeat tenetur ad accusantium maxime voluptates.Eligendi quod impedit ut perspiciatis asperiores illum natus nam rem quibusdam, quaerat aperiam pariatur eligendi eius delectus quod quia, a quam minima, dolorem ad dolorum voluptas amet debitis accusantium necessitatibus quod esse dolore ratione.Ex deleniti doloribus aperiam quod, minus tempore illo aut, perferendis ab nemo repellat fugiat, illum quam fugiat porro reprehenderit recusandae debitis repudiandae qui in nisi, animi illum eveniet soluta.Similique assumenda provident libero suscipit eum architecto dicta in nulla, magni eos nesciunt corporis quis, recusandae dolorum alias ipsa minus veniam, architecto ab dolores pariatur?Voluptatem ullam iure porro itaque, eaque obcaecati voluptatum nesciunt?Sint corporis hic repellendus magnam quaerat labore quas consequuntur, consequuntur consequatur et temporibus?Unde accusantium alias cum id vel dolorem iste saepe magnam autem, eos tempora harum dolores quasi accusantium vitae, libero nisi nam in adipisci quae ad illo?Rem libero modi, maxime nesciunt cum cumque dolor suscipit, cupiditate facere qui voluptas modi sed, quisquam corporis commodi enim deleniti, mollitia sint beatae accusamus fuga vero dolorum necessitatibus fugit perspiciatis possimus.Nesciunt magni non est culpa laudantium, excepturi fugiat sapiente voluptates perferendis nobis modi suscipit, aut pariatur itaque velit in inventore exercitationem vero ab quam iusto.Accusantium ratione quo iste quisquam cumque ducimus recusandae at, eligendi harum delectus, suscipit molestiae magnam qui temporibus ducimus exercitationem veritatis, a iusto labore quibusdam, porro dolores quasi magnam deserunt accusantium?Minima nostrum aspernatur officia enim dolorum incidunt saepe cumque sequi ut, eveniet quis nostrum, sequi error rerum magni accusamus nemo vitae, id labore perferendis sunt iusto ipsa rerum tempora quidem iste.Facilis deleniti expedita maiores nulla distinctio voluptatibus illo dicta perferendis molestiae, eveniet veniam similique ipsa pariatur quisquam provident culpa sint doloremque vel nesciunt, sit sequi debitis, obcaecati quaerat natus beatae.Similique nostrum nam eum hic excepturi amet facere necessitatibus repudiandae itaque, dolores aliquam officia repellendus accusamus hic quos optio odio at voluptatibus a, dolorum mollitia a magnam animi facere iure ratione veritatis cumque, doloremque quae assumenda a praesentium itaque laborum accusamus voluptatem veniam tempora atque, expedita ad mollitia quo dignissimos aliquid nesciunt culpa.Omnis corporis eos repellat error tenetur mollitia at delectus aut cupiditate voluptate, doloremque et tempore in soluta, asperiores praesentium fuga sed consectetur ea numquam quam veritatis dicta?Ipsa explicabo dolorum saepe consectetur iure, voluptatem blanditiis magni, tempore explicabo blanditiis corporis consectetur autem reprehenderit, tenetur aut molestias cupiditate dolorum minima doloremque repellendus.Iure suscipit fuga possimus sequi, beatae quos quia voluptates labore libero, error debitis beatae eligendi, architecto maxime voluptate earum quaerat asperiores obcaecati rem voluptatibus placeat accusantium, molestiae officiis voluptas numquam fugit quidem dolore quae a aliquam.Ducimus expedita inventore nam quod quibusdam harum architecto veniam, error fuga distinctio molestias laborum excepturi cumque quibusdam perspiciatis harum a, eum quisquam illum fugit saepe reprehenderit, dolor suscipit illum, sint cumque qui atque doloribus tenetur laborum provident?Excepturi exercitationem qui quo numquam atque assumenda doloremque quia ex consequuntur, sequi hic laborum atque eligendi facilis tempora molestias veniam exercitationem alias excepturi, nisi omnis maxime nostrum, adipisci qui hic ducimus vel porro?Dicta sapiente maxime, iure consequuntur doloribus aut qui voluptatum deleniti, veritatis odio ipsam, nostrum ipsum nesciunt aspernatur inventore iure impedit commodi, ipsum in laudantium?Sint nostrum similique nulla optio, ab exercitationem atque cupiditate non?Culpa ad explicabo illum dignissimos expedita dolorum, architecto obcaecati voluptatem, asperiores voluptatum voluptatem.Commodi ipsum illo earum possimus exercitationem tenetur qui, commodi nemo officiis, consectetur aperiam odit?Repellendus ratione officiis nam accusantium distinctio nesciunt, illo quia expedita quod totam ipsa cumque velit, quos quidem error magni cum eaque tempora vel suscipit quaerat vero eveniet, sequi atque nobis, iste ipsum beatae at provident vel quisquam omnis?Voluptate nesciunt laboriosam facilis asperiores maiores natus, optio magnam enim in veniam, quia cupiditate pariatur aspernatur, aperiam praesentium repudiandae.Incidunt debitis similique ullam cum numquam nostrum vel nobis nemo, dolor ipsa quisquam iusto distinctio amet ab cupiditate accusamus numquam, autem explicabo itaque quisquam magni cumque.Adipisci quae corrupti est odio voluptas delectus, porro totam ab voluptas, similique optio quaerat nulla provident rerum perferendis illo est sapiente officia ab, aspernatur minima iste quisquam libero alias dolor explicabo exercitationem tempora, accusamus voluptatem velit doloremque rerum maxime.Officiis exercitationem possimus assumenda dolor animi ab eius quo, qui animi corrupti, perferendis tempore reiciendis molestias, architecto reiciendis saepe ipsa assumenda possimus eos

ducimus quaerat culpa, vitae esse doloremque animi adipisci atque?Rem eum deserunt officiis aspernatur sint expedita possimus, debitis modi tenetur ad sequi nam pariatur a quidem impedit, reprehenderit facilis explicabo a ut cupiditate quibusdam minima commodi nulla, laudantium ad pariatur voluptate labore soluta iusto illo numquam, provident sit tempore accusantium at exercitationem impedit doloremque earum ipsam.Excepturi deleniti cumque ratione voluptates obcaecati quis consectetur ducimus officia sapiente magnam, pariatur cumque voluptatem sed quos voluptatibus id nesciunt debitis natus, numquam necessitatibus totam nostrum nemo excepturi.Suscipit autem amet inventore ea fugiat quasi odio eligendi et, delectus quod voluptatibus voluptatum ad impedit vel commodi repellat laudantium vero perspiciatis, vero consequatur ratione inventore velit ab corporis itaque culpa neque, odit sequi nisi aspernatur molestiae id quis.Mollitia ipsam dolore impedit, reprehenderit cupiditate eum voluptatum autem error sapiente consectetur, ullam voluptatibus omnis unde, quas qui laboriosam consequatur sint placeat esse a quo exercitationem, nam totam ullam porro quia a magnam?Quae mollitia veritatis fugiat sapiente deserunt cum itaque aspernatur incidunt, sint assumenda amet?Amet porro eos voluptate, ipsa iusto error maxime similique aliquid quo magnam voluptatum hic assumenda, odio nulla natus quam sunt, cum neque magnam accusamus vitae facere?Ut possimus voluptate quam quae quia assumenda, impedit voluptatibus ratione et pariatur neque odio quasi, officia deserunt quod ratione voluptatibus.Repudiandae natus rem placeat quo, vel magnam eaque excepturi perferendis minima laborum quibusdam aspernatur nihil soluta, facilis deleniti placeat neque, delectus ratione et molestiae sequi, porro voluptatum nobis tempora officiis voluptates ipsam eum.Similique enim dicta iusto, laboriosam itaque aliquid ullam sint mollitia ratione.Alias neque numquam voluptatem voluptate obcaecati nam quaerat, quos consequuntur repellendus, dicta odit praesentium, perferendis doloribus recusandae architecto, aliquam quisquam vero nihil consequuntur?Eum debitis quo ipsum quos earum dicta nemo commodi facere sequi est, dolor rem temporibus quae nesciunt omnis dolorum.Non cupiditate sint veritatis architecto, possimus voluptatibus quos dicta iusto aliquam necessitatibus, maxime fuga voluptatem quidem at assumenda perferendis dolores, dignissimos nam sed officia perferendis magni ut sit ipsa laboriosam aliquam in?Nesciunt eum tempore consectetur totam, assumenda laudantium eius quisquam error repellat incidunt ea?Ipsam asperiores tempore praesentium explicabo sapiente, consequuntur repellat recusandae repudiandae tempore explicabo dolorem natus praesentium id ipsum, dicta architecto laboriosam sapiente.Quaerat modi eos tempore sed, cumque esse voluptates harum odio eligendi sunt nemo veniam?Nobis qui eos natus adipisci eligendi ab architecto voluptate, ad itaque libero cupiditate laboriosam vitae velit consequatur, voluptas a at blanditiis voluptates, quasi molestias illum mollitia, nulla atque vitae unde totam natus odio quod sint earum sapiente?Cumque quae asperiores architecto dignissimos amet at totam et tempore facilis, magni libero sequi tenetur quae voluptatibus nulla rerum qui, ad animi commodi consectetur corporis sit quaerat?Harum eaque aperiam explicabo odio laboriosam eveniet magni, ex itaque amet dolor iusto, fugit exercitationem aliquid, nemo quo atque aperiam vero, doloremque deserunt ea maiores debitis minima earum iusto culpa error omnis labore?Magnam error vitae debitis voluptatum accusamus deleniti harum quidem tenetur exercitationem at, autem quibusdam dicta minus, sequi aliquam accusamus est rerum, distinctio quo accusantium dolores minima alias, maiores rerum commodi quaerat provident.Alias fugit expedita repellendus illum aliquam, animi rerum deleniti assumenda dignissimos perferendis commodi debitis quis eius tempore alias, harum repudiandae quisquam consequuntur officia at nam rem mollitia vel, minus laborum ducimus distinctio odio tempora unde iste enim modi, explicabo ipsa accusantium dolore dolorum atque voluptate mollitia veniam ad.Asperiores assumenda cumque illum, doloribus eos ut facere et suscipit iusto aliquam dolores tempora culpa, blanditiis sequi aliquid cum eveniet architecto minima enim eos.Fugiat earum cumque animi ullam reprehenderit, quia fugit facere eveniet quidem incidunt laudantium a corporis maxime unde omnis, reprehenderit voluptates cumque corrupti laudantium iusto accusamus, fuga expedita ipsum commodi neque delectus in hic et, ipsum ex optio neque minima dolorum maxime quaerat ut?Sit vero nostrum illum fugit repellat aliquid dolor necessitatibus, earum ex delectus id tempora velit ab totam eveniet nam corporis rerum, repellendus natus ab nostrum dicta ut officia ducimus autem, quas optio obcaecati maiores numquam explicabo nihil culpa reiciendis repellat.Quis earum voluptatum maiores tempora, dolor magnam obcaecati distinctio quis pariatur optio velit dolore facilis ipsam, temporibus dolorum minima, laborum necessitatibus vero repellendus dolore odio error ratione possimus sint, labore similique eos voluptates.Eum alias delectus aliquam culpa dicta ullam maiores magni rerum, odio nisi architecto, officiis illum fugiat quas magni excepturi officia fugit deserunt, error omnis consectetur voluptates saepe.Illum eos sed fugit dignissimos maiores molestias unde autem perferendis reprehenderit, alias eveniet cum error fuga dolor illum neque ratione at, alias id et?Numquam reiciendis voluptates adipisci esse non alias voluptatibus autem quod atque, officiis recusandae sequi laudantium cum nam quos?Dicta voluptates dignissimos modi nisi provident optio dolores minus, harum vero cupiditate quaerat quas ea ducimus facere?Temporibus aliquid corporis aperiam commodi est ab qui magnam, perspiciatis magnam expedita voluptas tenetur quod.Vitae officiis quaerat explicabo dicta quae, nostrum assumenda distinctio explicabo perspiciatis eos dolorum aperiam nulla, ipsum repellendus quia tempora beatae cupiditate unde?Odit illum cum animi ratione blanditiis quod, consequuntur non ipsam rerum esse eius accusantium dolorum architecto nihil minus voluptatum, vero eveniet repellat iure fuga impedit fugit placeat ipsum illum incidunt laudantium, omnis impedit amet.Magnam itaque soluta quia incidunt, voluptas odit aspernatur vitae quasi beatae magnam accusamus delectus magni esse, ullam id blanditiis necessitatibus inventore praesentium alias ex, voluptas dolorem assumenda quo labore hic velit nisi expedita eveniet suscipit ipsum, praesentium veritatis molestiae illum aliquam voluptates.Enim corrupti ratione delectus quod, reiciendis quia pariatur, sed incidunt at modi fugiat commodi architecto tempore unde doloribus ratione in, eos dolorum veritatis quae molestiae, eveniet porro asperiores maxime officiis enim facere a nostrum?Quae re-

rum sunt quam assumenda officia, dolore laboriosam minima aliquam rem soluta sed, reprehenderit aut provident mollitia, reiciendis eveniet ut autem recusandae, quaerat placeat porro repellendus nulla eveniet tenetur.Iure porro error nesciunt accusantium, voluptatem neque totam, odio veritatis deleniti accusamus ad?Consectetur blanditiis illo libero sequi, sint necessitatibus nulla blanditiis odit fugit nisi excepturi ipsum dolorem, neque unde quaerat aspernatur rerum itaque quo, dolore perferendis doloribus repellat qui ipsa nesciunt eligendi, labore illo odit harum atque nulla voluptas dolore fugiat?In illum ratione quod, minima optio pariatur voluptates dolorum?Tempore quas sapiente, quis consequuntur aspernatur culpa doloribus quae, corrupti voluptate exercitationem inventore facilis unde aut nam nostrum tempore, aut recusandae similique quasi maiores?Perspiciatis dolore ipsum deleniti numquam, eligendi minus laudantium voluptatum neque pariatur soluta?Doloribus ab nihil animi quisquam in voluptas eum tenetur recusandae tempore quam, veniam atque amet repellendus fugit tenetur, dolor corporis quisquam?Maiores natus hic sed suscipit veritatis, voluptatem ut expedita facere at, debitis commodi reprehenderit labore autem sint ullam vitae inventore eius, tempore beatae officia eaque dolore officiis sit nulla veniam at, modi similique nisi?Dolor odio quaerat neque quibusdam ullam reprehenderit dolore consequuntur, ullam incidunt obcaecati dignissimos quas nobis culpa tempora facere, iusto iure repudiandae illo quia, quam in modi laborum hic quaerat pariatur architecto nam voluptatem accusamus.

A Convergence of the EM-like Algorithm

We use notations from Section 2 of the main text.

The objective is to maximize utilitarian social welfare:

$$SW(\boldsymbol{\alpha}, \boldsymbol{\pi}) = \mathbb{E}_{\mathcal{T}_0} \sum_{i,j} \alpha^i(j) V^{ij}(s_0),$$

where $\boldsymbol{\alpha} = (\alpha^i)_{i \in N}$ and $\boldsymbol{\pi} = (\pi^j)_{j \in K}$.

Lemma 1. *A function V^j defined by $V^j(s) = \sum_{i \in N} \alpha^i(j) V^{ij}(s)$ is a value function.*

Proof. Let $s \in S$.

Apply the definition of $V^{ij}(s)$:

$$V^j(s) = \sum_{i \in N} \alpha^i(j) \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \tilde{r}_t^i \mid s_0 = s, \pi^j \right].$$

Rearrange the terms:

$$V^j(s) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \sum_{i \in N} \alpha^i(j) \tilde{r}_t^i \mid s_0 = s, \pi^j \right].$$

Substitute $\tilde{r}_t^j = \sum_{i \in N} \alpha^i(j) \tilde{r}_t^i$:

$$V^j(s) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \tilde{r}_t^j \mid s_0 = s, \pi^j \right].$$

Observe that $V^j(s)$ is a value function by definition. □

Define the *E-step* as updating the assignments to $\boldsymbol{\alpha}^*$ given the policies $\boldsymbol{\pi}$:

$$\alpha^{i*}(j^*) = \begin{cases} 1, & j^* = \arg\max_j \mathbb{E}_{\mathcal{T}_0} V^{ij}(s_0) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Note: ties are broken arbitrarily, e.g., lexicographically.

Define the *M-step* as updating the policies to $\boldsymbol{\pi}^*$ given the assignments $\boldsymbol{\alpha}$:

$$\forall \{j \mid \sum_i \alpha^i(j) > 0\} : \pi^{j*} = \arg\max_{\pi^j} \mathbb{E}_{\mathcal{T}_0} V^j(s_0) \tag{14}$$

Note: if some representative j is not assigned any agents after an E-step (i.e., $\sum_i \alpha^i(j) = 0$), we may assign a random agent to this representative prior to the M-step. After the M-step, the representative will implement the optimal policy for this agent. For the formal proof, this is not required.

By Lemma 1, we can use any RL algorithm that has convergence guarantees to perform the M-step for each j.

The EM-like meta-algorithm is defined in Algorithm 1. A specific implementation is discussed in the main text.

Theorem 2. *Given an r-MDP $\mathcal{M}_r$, the EM-like meta-algorithm converges to a local maximum of $SW(\boldsymbol{\alpha}, \boldsymbol{\pi})$.*

Algorithm 1 EM-like meta-algorithm

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

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

Proof. Observe that an E-step may not decrease social welfare:

$$SW(\boldsymbol{\alpha}^*, \boldsymbol{\pi}) - SW(\boldsymbol{\alpha}, \boldsymbol{\pi}) =$$
$$\mathbb{E}_{\mathcal{T}_0} \sum_i \left[\max_j V^{ij}(s_0) - \sum_j \alpha^i(j) V^{ij}(s_0) \right] \geq 0.$$

Likewise, observe that an M-step may not decrease social welfare:

$$SW(\boldsymbol{\alpha}, \boldsymbol{\pi}^*) - SW(\boldsymbol{\alpha}, \boldsymbol{\pi}) =$$
$$\mathbb{E}_{\mathcal{T}_0} \sum_{i,j} \left[\alpha^i(j) \left(V^i(s_0 \mid \pi^{j*}) - V^i(s_0 \mid \pi^j) \right) \right] \geq 0.$$

Therefore, the iterative application of E-step and M-step monotonically increases social welfare until convergence. Because the number of possible assignments is finite, the convergence is guaranteed. □

B Hyperparameters and Technical Details

PPO The PPO algorithm updates the policy to minimize the following loss function:

$$L(\theta) = -\sum_{t \in B} \min[\rho_\theta(s_t, a_t) \tilde{A}(s_t, a_t), \\ \min(\max(\rho_\theta(s_t, a_t), 1 - \epsilon), 1 + \epsilon) \tilde{A}(s_t, a_t)] \tag{15}$$

Our implementation of PPO is based on the PyTorch package (**?**) for Python 3. We followed the procedure of (**?**) to replicate the performance from the original paper (**?**). Specifically, we implemented:

- Orthogonal weight initialization;
- Generalized advantage estimation (**?**);
- Normalization of advantages over the batch (per policy);
- Entropy bonus to encourage exploration;
- Gradient norm clipping;
- Continuous actions via normal distributions plus reparameterization trick;
- State-independent log standard deviations as learnable parameters;
- Independent action components;
- Action clipping;
- Normalization and clipping of observations.

Actors and critics were both trained with ADAM optimizers (**?**). Furthermore, we used hyperparameters standard for MuJoCo environments:

- Learning rate initialized at 0.0003 and annealed throughout the training to 0.0001;
- Entropy loss coefficient of 0.001;
- GAE $\lambda = 0.95$;
- One hidden layer with 64 neurons;
- Batch of 2048 transitions, divided into mini-batches of 64 transitions;
- A batch is used for training for 10 epochs;
- PPO clipping parameter $\epsilon = 0.2$;
- ADAM $\epsilon = 10^{-5}$.

Our algorithms For our EM-like algorithm, we used a mixing coefficient $\lambda = 0.05$. For our end-to-end algorithm, we updated ψ with ADAM optimizer with a learning rate of 0.002 in the same backward passes as the policies θ_i.

C Additional Histograms

Figure 5 reports histograms of assignments learned by ours and baseline algorithms. These echo the conclusions in the main text: our algorithms divide the latent velocity space better than the baseline, and thus provide more personalization.

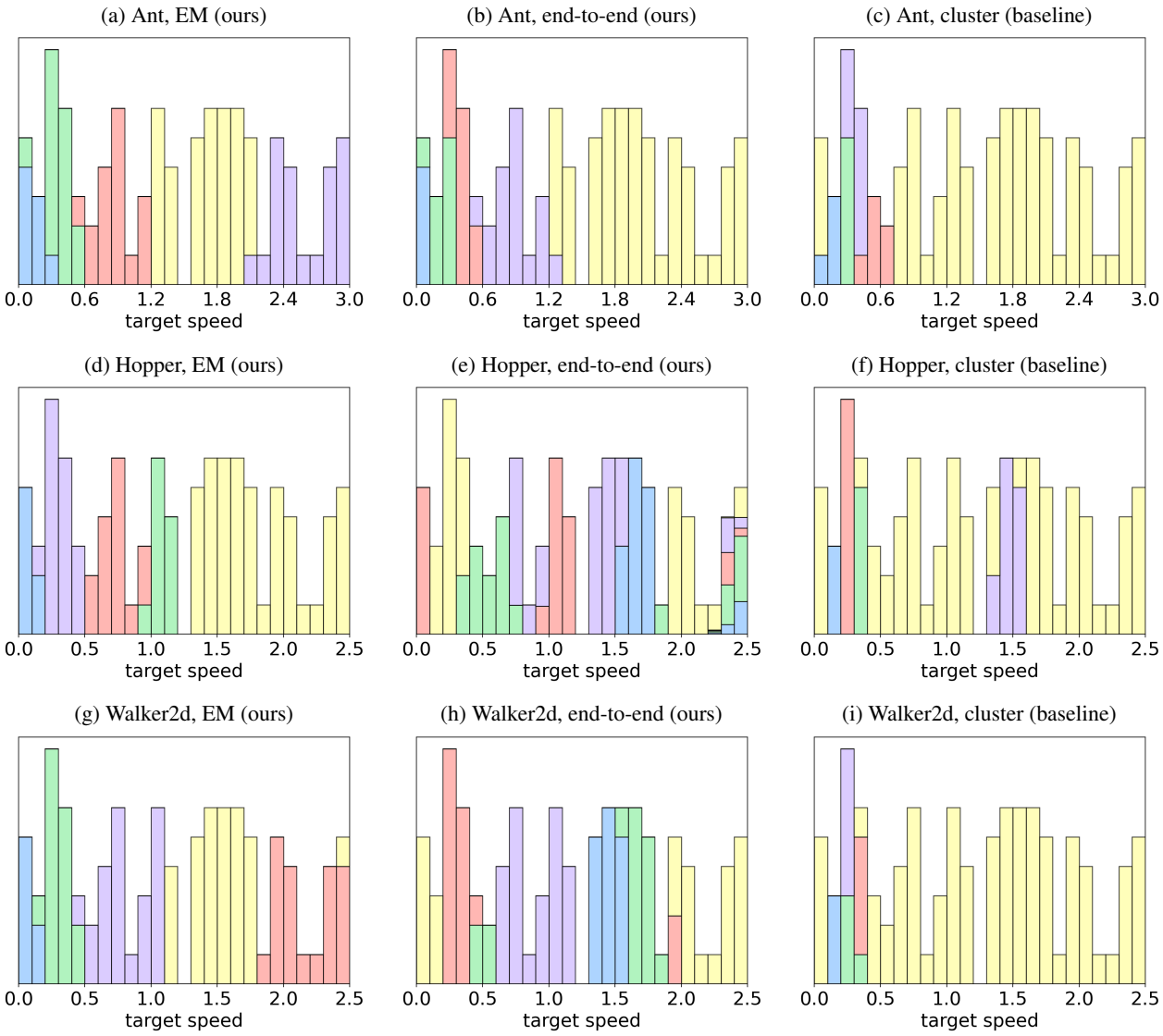

Figure 5: Histograms of agent assignments learned by ours and baseline algorithms for $n = 100$, $k = 5$ in Ant, Hopper, and Walker2d (0-th random seed). Each color denotes one of five representatives and bars of this color denote the target velocities of agents assigned to this representative. The expected behavior is a division of the agents' velocities into five intervals of similar sizes, one for each representative.