F-DRL: Federated Dynamics Representation Learning for Robust Multi-Task Reinforcement Learning

Anurag Upadhyay , Yashar Baradaranshokouhi , Xin Lu * , Jun Li , Yanguo Jing

Posted Date: 29 January 2026

doi: 10.20944/preprints202601.2257.v1

Keywords: robotics priors; action-conditioned system dynamics; state-action encode; deep reinforcement learning; robotic manipulation; dynamics aware embeddings

Article

F-DRL: Federated Dynamics Representation Learning for Robust Multi-Task Reinforcement Learning

Anurag Upadhyay [1], Yashar Baradaranshokouhi [1], Xin Lu [1,*], Jun Li [2] and Yanguo Jing [3]

[1] School of Computing and Creative Industries, Leeds Trinity University, Brownberrie Lane Horsforth, Leeds LS18 5HD, UK
[2] Centre for Computational Engineering Sciences, School of Aerospace, Transport and Manufacturing (SATM) Cranfield University, Cranfield, Bedfordshire MK43 0AL, UK
[3] Institute of Business, industry and leadership, University of Cumbria,Cumbria, CA1 2HH, United Kingdom
* Correspondence: X.Lu@leedstrinity.ac.uk

Abstract

Reinforcement learning for robotic manipulation is often limited by poor sample efficiency and unstable training dynamics, challenges that are further amplified in federated settings due to data privacy constraints and task heterogeneity. To address these issues, we propose *F-DRL*, a federated dynamics-aware representation learning framework that enables multiple robotic tasks to collaboratively learn structured latent representations without sharing raw trajectories or policy parameters. The framework combines robotics priors with an action-conditioned latent dynamics model to learn low-dimensional state and state–action embeddings that explicitly capture task-relevant geometric and transition structure. Representation learning is performed locally at each client, while a central server aggregates encoder parameters using a similarity-weighted scheme based on second-order latent geometry. The learned representations are then used as frozen auxiliary inputs for downstream model-free reinforcement learning.We evaluate F-DRL on seven heterogeneous robotic manipulation tasks from the MetaWorld benchmark. While achieving performance comparable to centralized training and standard federated baseline, F-DRL substantially improves training stability relative to FedAvg on heterogeneous manipulation tasks with partially shared dynamics (e.g., Drawer-Open and Window-Open),reducing the mean across-seed standard deviation and the AUC of this deviation by over **60%**. The method remains neutral on simple tasks and performs less consistently on contact-rich manipulation tasks with task-specific dynamics, indicating both the benefits and the practical limits of representation-level knowledge sharing in federated robotic learning.

Keywords: robotics priors; action-conditioned system dynamics; state-action encode; deep reinforcement learning; robotic manipulation; dynamics aware embeddings

1. Introduction

The combination of deep learning and reinforcement learning (RL), following the seminal work of [1], has accelerated the adoption of Deep Reinforcement Learning (DRL) across domains such as robotics, gaming, autonomous driving, and recommendation systems [2]. Although DRL enables end-to-end learning directly from high-dimensional raw observations without explicit feature engineering, these methods remain notoriously sample inefficient, often requiring tens of millions of interactions to solve even simple tasks. This inefficiency arises from several inherent challenges in RL, including the need for extensive exploration, long-horizon credit assignment, and instability in value-based and policy-gradient optimization. These issues are amplified in real-world robotic settings, where data collection is slow, expensive, and physically risky, and where even high-fidelity simulators suffer from sim-to-real gaps [3]. As a result, training instability and poor reproducibility remain fundamental barriers to deploying DRL systems in practice.

A promising direction for mitigating these challenges is representation learning [4], where compact, low-dimensional embeddings are learned explicitly rather than implicitly within the policy

 doi:10.20944/preprints202601.2257.v1

network. Such representations constrain the hypothesis space of downstream policy learning, improving sample efficiency and stability. Recent work on unsupervised representation learning [5] has demonstrated that structured latent spaces can capture meaningful factors of variation useful for robotic control. While much of the existing literature focuses on high-dimensional visual observations [6], representation learning is equally critical for proprioceptive inputs, since the resulting observation stream corresponds to a partially observable Markov decision process (POMDP). In this setting, representations must filter sensor noise while preserving task-relevant physical structure.

However, learning representations solely from interaction data is inherently challenging: unconstrained encoders can capture spurious correlations or task-specific artefacts that fail to generalize. Robotics Priors [7] address this issue by introducing physics-inspired inductive biases—such as temporal coherence, causality, repeatability, and proportionality—that encourage physically meaningful latent spaces. By constraining representation learning with these priors, learned embeddings better reflect the underlying structure of robotic dynamics and improve robustness in downstream control.

Building on this foundation, an action-conditioned representation is introduced to explicitly capture system dynamics within a robotics-prior-constrained latent space. Unlike state-only embeddings, the proposed encoder models how actions transform states, enabling the latent space to encode transition structure rather than static observations alone. The resulting representation, $z(s, a)$, is therefore well suited for transfer across tasks with partially shared dynamics. However, exploiting this property in collaborative multi-task settings requires a mechanism for selective and stable knowledge sharing. In realistic deployments, robots often learn multiple tasks under privacy, communication, or operational constraints that preclude centralized training. In such settings, naive parameter sharing can be harmful: tasks with incompatible dynamics can interfere, leading to unstable learning and negative transfer. This motivates a formulation in which knowledge is shared only to the extent that tasks are dynamically related, and where the unit of sharing is the representation rather than the policy or value function.

Federated learning (FL) provides a natural framework for such collaboration, as it enables decentralized knowledge sharing without exposing raw data or requiring centralized optimization. However, existing federated reinforcement learning (FRL) approaches are poorly suited to heterogeneous robotic settings: most aggregate policy or value-function parameters directly, tightly coupling federation with unstable RL updates and making these methods highly sensitive to task diversity.

To address this limitation, a representation-centric federated learning framework is introduced in which federation is applied exclusively to representation learning and fully decoupled from downstream policy optimization. Multiple tasks performed by the same robot are treated as federated clients, reflecting realistic scenarios in manufacturing, service robotics, and household automation. Clients are aligned based on learned dynamics similarity rather than task identity or uniform averaging, enabling selective, stability-oriented knowledge transfer while avoiding direct synchronization of unstable policy updates. This design shifts the role of federation from policy coordination to representation alignment, which is more appropriate for heterogeneous robotic systems.

The core hypothesis is that tasks sharing the same state and action spaces, and exhibiting partially overlapping transition dynamics, should share similar latent representations. Federated aggregation guided by dynamics-based similarity encourages alignment across such tasks, improving generalization while mitigating negative transfer and remaining neutral when heterogeneity is limited. To the best of the authors' knowledge, this work is the first to integrate dynamics-aware action-conditioned representations with representation-level federated aggregation within a unified FRL framework for robotic manipulation.

Contributions.

The main contributions of this work are:

 doi:10.20944/preprints202601.2257.v1

1. A federated dynamics-aware representation learning framework that learns shared state and action-conditioned embeddings across multiple robotic manipulation tasks without sharing raw data or policy parameters, explicitly targeting stability under task heterogeneity.
2. An extension of robotics-prior-constrained representation learning through the incorporation of action-conditioned system dynamics, enabling embeddings that capture task-relevant transition structure rather than static state features.
3. A similarity-weighted federated aggregation strategy based on second-order latent geometry, which selectively aligns clients according to dynamics similarity and mitigates negative transfer induced by uniform averaging.
4. An extensive empirical evaluation on heterogeneous MetaWorld manipulation tasks demonstrating that the learned representations act as a stabilizing auxiliary signal for downstream model-free reinforcement learning, substantially reducing variance across random seeds while achieving performance comparable to centralized training.

The remainder of this paper is organized as follows. Section 2 reviews related work on deep reinforcement learning, representation learning for control, and federated reinforcement learning. Section 3 presents the proposed federated dynamics-aware representation learning framework. Section 4 details the experimental setup and empirical evaluation. Section 5 concludes and outlines directions for future work.

2. Related Work

This section reviews prior work across three complementary research directions that motivate the proposed framework. We first discuss advances in deep reinforcement learning, highlighting persistent challenges such as sample inefficiency, training instability, and limited generalisation, which are particularly pronounced in robotic manipulation and motivate the need for structured and transferable representations. We then review representation learning methods for reinforcement learning, with a focus on robotics-prior-based and dynamics-aware approaches, which demonstrate that incorporating physical inductive biases and transition structure into latent representations can improve learning efficiency and robustness. Finally, we survey FRL methods that address privacy and data-sharing constraints, noting that most existing approaches rely on policy- or value-level aggregation and remain highly sensitive to task heterogeneity, while representation-level knowledge sharing is comparatively under-explored. Together, these works expose a gap at the intersection of dynamics-aware representation learning and federated aggregation, which our framework addresses by decoupling representation learning from control and enabling selective, stability-oriented knowledge sharing across heterogeneous robotic tasks.

2.1. Deep Reinforcement Learning and Sample Inefficiency

DRL achieved widespread attention following the seminal work of [1], which demonstrated that deep neural networks can learn control policies directly from raw observations. While this removed the need for hand-engineered features, DRL methods remain fundamentally limited by unstable optimization dynamics, high variance, and severe sample inefficiency. A key innovation introduced in the DQN framework is the *experience replay buffer*, which decorrelates training data and stabilizes off-policy learning. However, uniform sampling treats all transitions as equally informative, constraining learning efficiency. Several extensions addressed these limitations by improving representational and learning capacity. The duelling network architecture [8] decomposes the Q-function into state-value and advantage components for faster generalization, while Double DQN [9] reduces overestimation bias through decoupled action selection and evaluation. Prioritized Experience Replay (PER) [10] further enhances sample efficiency by sampling transitions according to temporal-difference error, with bias correction via importance sampling. Policy-gradient approaches take a complementary perspective by directly parametrizing the policy $\pi_\theta(a \mid s)$ and optimizing it via gradient ascent. These methods provide expressive control policies but are prone to high variance and unstable updates. Trust

 doi:10.20944/preprints202601.2257.v1

Region Policy Optimization (TRPO) [11] addressed this by enforcing a trust-region constraint on policy updates, ensuring monotonic improvement. Proximal Policy Optimization (PPO) [12] introduced a clipped surrogate objective that prevents destructive policy shifts while remaining computationally efficient. Generalized Advantage Estimation (GAE) [13] reduced variance in advantage estimation by combining multi-step returns through an exponentially weighted scheme, offering a controllable bias-variance trade-off. Modern off-policy actor-critic methods unify value-based and policy-based principles. DDPG [14] extended deterministic policy gradients to continuous control using replay buffers, target networks, and exploration noise. TD3 [15] further improved stability and sample efficiency by introducing twin critics to mitigate overestimation, target policy smoothing, and delayed policy updates.

Although these advances substantially improve optimization stability and reduce estimation variance, they operate primarily at the level of value approximation and policy updates. They do not directly address the challenge of learning compact, structured, and transferable representations—a limitation that becomes critical in robotic manipulation, where data collection is expensive and task variations induce significant state–action distribution shifts. This motivates the study of explicit representation learning techniques, which we discuss next.

2.2. Representation Learning for Reinforcement Learning

2.2.1. Robotics Priors and Physical Constraints

A long-standing view in representation learning is that useful latent spaces should reflect general priors about how the physical world behaves [4]. In robotics, this idea was first introduced by [16], who proposed *Robotics Priors*: a set of physics-inspired constraints—temporal coherence, proportionality, causality, and repeatability—that impose local structure on the latent state space. These priors encourage representations that evolve smoothly over time, respond predictably to actions, and remain consistent with underlying physical dynamics. Several extensions have been proposed to enhance this framework. Position–Velocity Encoders (PVEs) [17] incorporate additional priors to induce an explicit position–velocity decomposition reflecting canonical quantities in robot motion. Other works broadened the applicability of Robotics Priors to multi-task reinforcement learning via task-gating mechanisms and task-coherence priors that cluster representations within an episode [18]. To improve cross-episode consistency, [19] introduced a reference point prior that stabilizes latent representations across trajectories. Extensions to partially observable settings were explored by [20], who incorporated recurrent architectures to maintain temporally coherent latent states. Further, reward-shaped variants of Robotics Priors were proposed in [21], and adaptations to continuous-action spaces were introduced in [22]. Most prior work in this line of research focuses on imposing structural constraints on state-only representations and is studied in centralized or single-task settings. In contrast, our work considers action-conditioned, dynamics-aware representations and investigates how such structured embeddings can be selectively shared across tasks in a federated setting under privacy constraints.

2.2.2. Dynamics Modelling via Self-Prediction

Dynamics-modelling-based representation learning methods learn low-dimensional embeddings by approximating transition and reward distributions $p(s' \mid s, a)$ and $p(r \mid s, a)$. By jointly training the representation and predictive models, the resulting latent space is encouraged to encode task-relevant system dynamics. These approaches are typically categorised according to the downstream reinforcement learning paradigm in which the learned representations are used. Model-free methods employing deterministic encoders include [23–27]. In contrast, approaches using stochastic encoders learn parametrized distributions over latent states rather than point embeddings, as in [28–30]. Although often described as model-free, many of these methods still learn approximate transition or reward models during representation learning. In this context, "model-free" refers to the absence of model-based planning in the control pipeline rather than the absence of predictive modelling altogether. Model-based methods with deterministic encoders include [31–35]. Approaches with stochastic latent dynamics models, such as [36–38], employ probabilistic representations to capture uncertainty and

 doi:10.20944/preprints202601.2257.v1

support planning-based control. While these approaches demonstrate the benefits of dynamics-aware latent spaces for reinforcement learning, they are predominantly studied in centralized settings and do not address how dynamics representations can be aligned, aggregated, or selectively shared across heterogeneous agents, which is the focus of our framework.

2.3. Federated Reinforcement Learning

Federated reinforcement learning (FRL) has been widely studied as a means of addressing sequential decision-making problems in privacy-preserving and decentralized settings [39]. In most existing FRL approaches, federation is performed at the policy or value-function level, where the parameters of the actor, critic, or Q-network are periodically synchronized across clients using federated averaging or proximal variants [40–44]. While effective in settings with homogeneous tasks or aligned objectives, such approaches tightly couple federated synchronization with the inherently unstable optimization dynamics of reinforcement learning. As a result, aggregation noise, non-stationarity, and policy interference are amplified under task diversity, often leading to degraded performance or unstable training.

FRL methods have also been applied to robotic systems, motivated by scenarios in which robots must collaboratively learn while preserving data locality and privacy [45–51]. These works demonstrate the feasibility of federated training across robotic platforms or tasks and explore extensions such as lifelong learning, preference-based aggregation, and selective communication. A subset of studies explicitly considers task heterogeneity in robotic manipulation and control [52–54]. However, despite addressing heterogeneity at the algorithmic or optimization level, these methods remain fundamentally rooted in policy-level or value-function aggregation. Consequently, federated updates are directly entangled with unstable reinforcement learning gradients, rendering such approaches sensitive to inter-task mismatch and prone to negative transfer when task dynamics differ.

A key limitation shared by existing FRL methods lies in the choice of aggregation unit. Policies and value functions are task-specific, highly non-linear, and tightly coupled to reward structure and exploration dynamics. Aggregating these parameters across heterogeneous tasks implicitly assumes alignment at the control level, an assumption that rarely holds in realistic multi-task robotic settings. Even when tasks share identical state and action spaces, differences in transition dynamics or reward geometry can cause federated averaging to introduce destructive interference, undermining both convergence and training stability.

In contrast, representation-centric federated learning frameworks shift the unit of federation from control parameters to learned representations. When federation operates on dynamics-aware embeddings rather than policies, aggregation is decoupled from unstable policy optimization and can be guided by structural similarity in latent space. Aligning clients based on latent geometric similarity enables selective knowledge sharing under task heterogeneity, while avoiding direct synchronization of unstable gradients. This architectural distinction underpins the stability improvements observed in recent representation-level FRL approaches and motivates the framework introduced in this work.

3. Framework and Methodology

3.1. Design Rationale and Framework Overview

The proposed framework follows a *representation-centric federated learning paradigm*, in which representation learning is decoupled from downstream control and federated synchronization is applied exclusively in latent space. This design is motivated by the instability of policy- and value-level aggregation under task heterogeneity, which can amplify variance and induce negative transfer in reinforcement learning. By federating only representation parameters and keeping control optimization local, the framework enables selective knowledge sharing while preserving training stability. An overview of the modular framework is shown in Figure 1, highlighting the separation between federated representation learning and local policy optimization.

 doi:10.20944/preprints202601.2257.v1

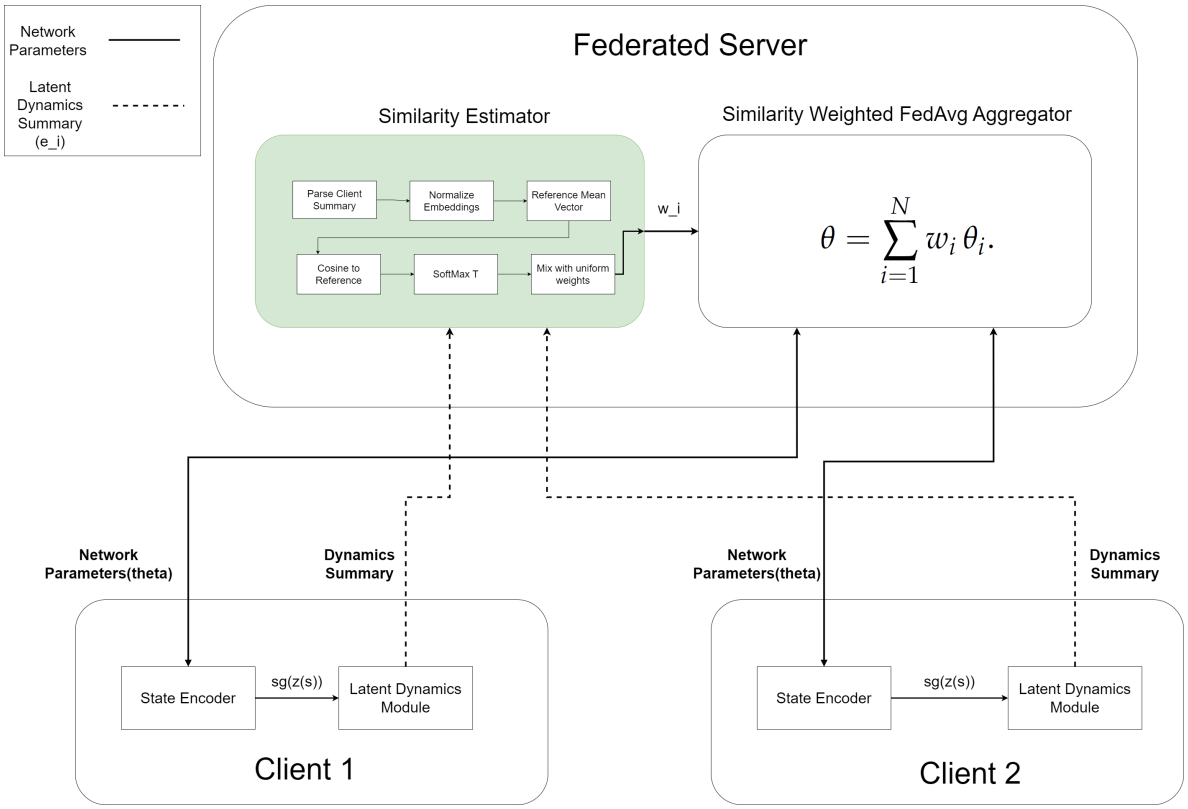


Figure 1. Overview of the proposed F-DRL framework.

3.2. Federated Learning Setting and Assumptions

We consider a federated reinforcement learning setting in which each client is modelled as a Markov Decision Process (MDP) [55], defined by a state space $\mathcal{S}_i$, action space $\mathcal{A}_i$, transition dynamics P_i, reward function R_i, and discount factor γ. In our experiments, each client corresponds to a distinct robotic manipulation task from the MetaWorld benchmark [56]. All federated clients share identical observation and action spaces and operate on robots with comparable morphology. This assumption reflects realistic deployment scenarios such as multi-task industrial manipulators, service robots, or household robots, where a single physical platform performs multiple heterogeneous tasks under a fixed embodiment. Offline trajectories are required for representation learning, and no interaction data is shared across clients, preserving data privacy. Under these conditions, the framework is most effective when tasks share common dynamical structure but differ in task-specific objectives. This setting enable beneficial representation-level knowledge sharing without direct policy synchronisation.

We adopt a client–server federated learning architecture [57,58], where clients perform local representation learning and periodically communicate model updates to a central server. The server aggregates these updates to produce a global representation model, which is then broadcast back to clients for subsequent local training rounds. *Only representation parameters are shared between clients and the server*. Policy and value networks, replay buffers, and environment interactions remain strictly local to each client throughout training. This design preserves data privacy while decoupling federated synchronization from unstable policy optimization.

Table 1 summarizes the key architectural differences between the proposed framework and existing federated reinforcement learning approaches.

 doi:10.20944/preprints202601.2257.v1

Table 1. Comparison between existing federated reinforcement learning frameworks and the proposed F-DRL framework.

Aspect	Existing FRL	F-DRL (Ours)
Federated object	Policy / value networks	Representation encoder
Shared signal	Control parameters	Latent dynamics embeddings
Aggregation strategy	Uniform / proximal averaging	Similarity-weighted aggregation
Sensitivity to task heterogeneity	High	Reduced
Training stability focus	Implicit	Explicit
Local control optimisation	Federated	Fully local
Privacy exposure	Policy gradients	Latent summaries only
Extensibility	Limited	Modular

This comparison highlights how shifting federation from policy parameters to latent representations fundamentally changes the stability and applicability of federated reinforcement learning under task heterogeneity.

3.3. Overview of the Proposed Framework

Our framework is composed of two tightly coordinated yet decoupled stages: (i) *federated dynamics-aware representation learning* (see Figure 1) and (ii) *local downstream policy learning* (see Figure 3)

In the federated training stage, each client privately learns a structured state encoder $z(s)$ and an action-conditioned latent dynamics module $z(s, a)$ from offline trajectories using robotics priors and auxiliary dynamics objectives. Instead of sharing raw data or policy parameters, each client periodically transmits only a compact latent dynamics summary e_i together with its local encoder parameters to a central federated server. The server computes inter-client similarity through a dedicated similarity estimator, which normalizes client summaries, constructs a reference mean embedding, and derives similarity scores via cosine alignment. These scores are transformed into aggregation weights and used by a similarity-weighted FedAvg aggregator to produce a global encoder that selectively integrates knowledge from dynamically related tasks. The aggregated encoder is then broadcast back to all clients for the next local training round.

In the downstream control stage, the learned encoders are frozen and used as auxiliary inputs for model-free reinforcement learning. Each client independently trains its own actor–critic policy using local environment interactions, replay buffers, and rewards, without any further federated synchronization. This strict decoupling ensures that unstable policy optimization remains fully local, while federation is applied exclusively to representation learning, where it improves stability, transferability, and robustness under task heterogeneity.

3.4. Federated Representation Learning

3.4.1. Local Client Architecture

Figure 2 illustrates the local client architecture. Each client trains its representation model using offline trajectories consisting of a mixture of random and expert demonstrations, ensuring sufficient coverage of the state–action space while preserving task-relevant structure.

 doi:10.20944/preprints202601.2257.v1

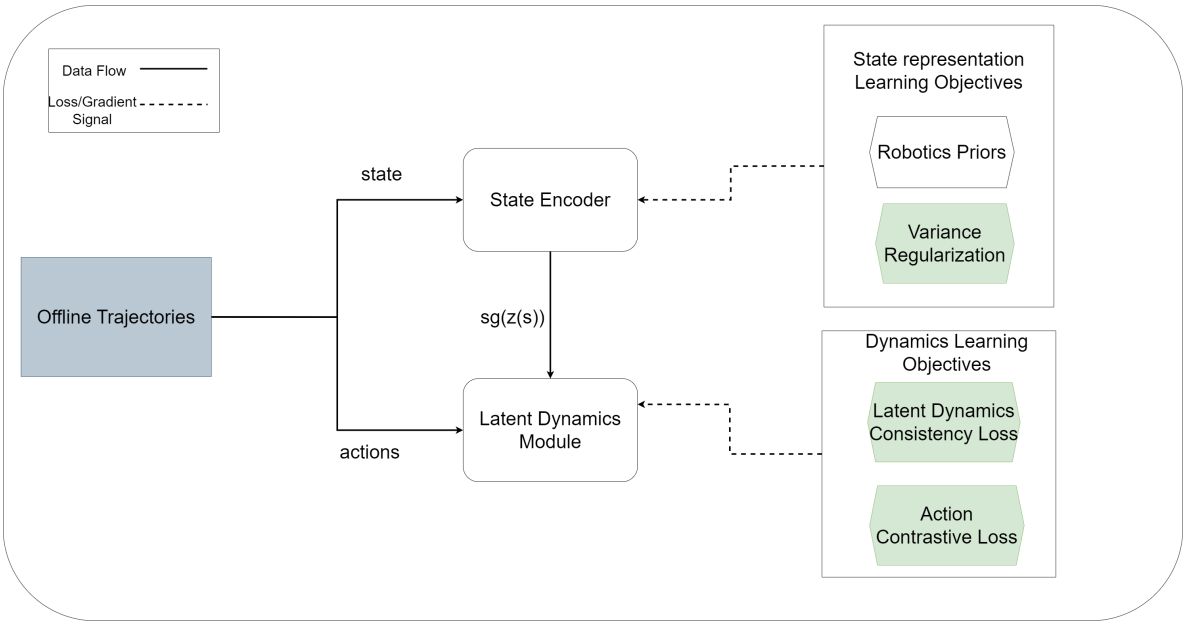


Figure 2. Local Client Architecture

We learn deterministic state and state–action embeddings using parametric neural encoders. To impose physics-inspired structure on the latent space, we incorporate Robotics Priors [22], which encode inductive biases such as temporal coherence, proportionality, repeatability, and causality. These priors constitute the primary geometric training objectives for the state encoder during representation learning.

The state encoder outputs a latent state representation $z(s)$ and is trained using the following objectives.

Temporal Coherence Prior:

$$\mathcal{L}_{\text{temp}} = \mathbb{E}\Big[\big(\|\Delta z_t\|_1 \, \exp(-\alpha\|a_t\|_1)\big)^2\Big]. \tag{1}$$

Proportionality Prior:

$$\mathcal{L}_{\text{prop}} = \mathbb{E}\Big[\big(\|\Delta z_{t_2}\|_2 - \|\Delta z_{t_1}\|_2\big)^2 \exp\Big(-\beta\|a_{t_1} - a_{t_2}\|_2^2\Big)\Big]. \tag{2}$$

Repeatability Prior:

$$\mathcal{L}_{\text{rep}} = \mathbb{E}\Big[\|\Delta z_{t_2} - \Delta z_{t_1}\|_2^2 \exp\Big(-\|z_{t_2} - z_{t_1}\|_2^2\Big) \exp\Big(-\beta\|a_{t_1} - a_{t_2}\|_2^2\Big)\Big]. \tag{3}$$

Causality Prior:

$$\mathcal{L}_{\text{caus}} = \mathbb{E}\Big[\exp\Big(-\|z_{t_2} - z_{t_1}\|_2^2\Big) \exp\Big(-\beta\|a_{t_1} - a_{t_2}\|_2^2\Big)\Big]. \tag{4}$$

To prevent representational collapse and encourage sufficient dispersion across latent dimensions, we additionally incorporate a variance regularization term inspired by VICReg [59]. The overall training objective for the state encoder is defined as:

$$\mathcal{L}_{\text{state}} = w_{\text{temp}}\mathcal{L}_{\text{temp}} + w_{\text{caus}}\mathcal{L}_{\text{caus}} + w_{\text{prop}}\mathcal{L}_{\text{prop}} + w_{\text{rep}}\mathcal{L}_{\text{rep}} + w_{\text{var}}\mathcal{L}_{\text{var}}. \tag{5}$$

The state–action encoder takes the state embedding $z(s)$ together with the executed action a and produces a joint embedding $z(s, a)$. A stop-gradient operation is applied to $z(s)$ to prevent dynamics learning objectives from distorting the geometric structure induced by the robotics priors.

 doi:10.20944/preprints202601.2257.v1

The dynamics module is trained using a latent dynamics consistency loss that aligns the encoded next state representation $z(s')$ with the predicted next latent state $\hat{z}(s,a)$ produced by the action-conditioned dynamics module for the current state and action:

$$\mathcal{L}_{\text{dyn}} = \mathbb{E}_{(s,a,s')}\left[\left\|z(s') - \hat{z}(s,a)\right\|^2\right]. \tag{6}$$

To enforce explicit action conditioning in the dynamics head, a mismatched ("wrong") action is constructed by randomly permuting actions within a minibatch. Let $\tilde{a}$ denote the permuted action corresponding to a within the minibatch, and let $\cos(\cdot,\cdot)$ denote cosine similarity. The wrong-action loss is defined as:

$$\mathcal{L}_{\text{wa}} = \mathbb{E}_{(s,a,s')}\left[-\log\left(\sigma\left(\frac{\cos\big(\hat{z}(s,a),\, z(s')\big) - \cos\big(\hat{z}(s,\tilde{a}),\, z(s')\big)}{T}\right) + \epsilon\right)\right], \tag{7}$$

where $\sigma(\cdot)$ is the sigmoid function, $T > 0$ is a temperature parameter, and ϵ is a small constant for numerical stability.

The full objective for the state–action encoder is:

$$\mathcal{L}_{\text{SA}} = w_{\text{dyn}}\mathcal{L}_{\text{dyn}} + w_{\text{wa}}\mathcal{L}_{\text{wa}}. \tag{8}$$

3.4.2. Similarity-Weighted Federated Aggregation

The federated server module in Figure 1 illustrates the federated aggregation procedure. At each federated round, each client computes a compact summary vector from a held-out validation set, instantiated as a second-order Gram fingerprint of its state–action embeddings. These summaries capture latent geometric structure while remaining robust to sign ambiguities and representation noise.

Let $e_i \in \mathbb{R}^m$ denote the summary vector reported by client i, and let N denote the number of participating clients. Each summary is normalized:

$$\tilde{e}_i = \frac{e_i}{\|e_i\|_2}. \tag{9}$$

A reference vector is constructed as the normalized mean of the client summaries:

$$\tilde{r} = \frac{\frac{1}{N}\sum_{i=1}^{N}\tilde{e}_i}{\left\|\frac{1}{N}\sum_{i=1}^{N}\tilde{e}_i\right\|_2}. \tag{10}$$

Each client receives a similarity score:

$$\text{sim}_i = \langle \tilde{e}_i, \tilde{r}\rangle. \tag{11}$$

Aggregation weights are obtained via a temperature-scaled softmax and interpolated with a uniform prior:

$$w_i = \alpha\left(\frac{\exp(\text{sim}_i/T)}{\sum_{k=1}^{N}\exp(\text{sim}_k/T)}\right) + (1-\alpha)\frac{1}{N}. \tag{12}$$

The uniform interpolation mitigates overly sharp weighting and improves robustness during early training stages when latent representations are still evolving.

The global encoder parameters θ are computed via similarity-weighted federated averaging:

$$\theta = \sum_{i=1}^{N} w_i\,\theta_i. \tag{13}$$

Algorithm 1 describes the federated training procedure for learning dynamics-aware representations. At each federated round, every client samples offline trajectories and updates its state encoder

 doi:10.20944/preprints202601.2257.v1

$z(s)$ using robotics priors and variance regularization (Section 5). The action-conditioned dynamics module $z(s,a)$ is then updated using the latent dynamics consistency and action-contrastive losses (Section 8). Each client computes a latent dynamics summary by evaluating the $z(s,a)$ embeddings on a held-out validation set and transmits this summary together with its local encoder parameters to the server. The server estimates inter-client similarity from these summaries and uses the resulting weights to perform similarity-weighted aggregation of the encoder parameters. The aggregated global encoder is then broadcast back to all clients and used to initialize the next round of local training.

Algorithm 1 Federated Dynamics-Aware Representation Learning

Require: Clients $\{\mathcal{C}_i\}_{i=1}^{N}$, initial encoder parameters θ^0
for federated round $r = 1, \ldots, R$ **do**
  for all clients i **in parallel do**
    Sample offline trajectories $\mathcal{D}_i$
    Update state encoder $z(s;\theta_i)$ using Robotics Priors and variance regularization
    Update action-conditioned module $z(s,a)$ using latent dynamics consistency and action-contrastive losses
    Compute latent dynamics summary e_i
    Send (θ_i, e_i) to the server
  end for
  Compute similarity weights $\{w_i\}$ from $\{e_i\}$
  Aggregate global encoder $\theta^r \leftarrow \sum_{i=1}^{N} w_i \theta_i$
  Broadcast θ^r to all clients
end for

Upon receiving the aggregated global encoder, each client initializes the next local training round by updating its encoder parameters with the broadcast model, using a convex interpolation to retain task-specific structure defined as follows

$$\theta_i^{(r+1)} \leftarrow \alpha\, \theta^{(r)} + (1-\alpha)\, \theta_i^{(r)}, \tag{14}$$

where $\theta^{(r)}$ denotes the aggregated global encoder parameters at federated round r, $\theta_i^{(r)}$ denotes the local encoder parameters of client i prior to synchronization, and $\theta_i^{(r+1)}$ denotes the updated local encoder used to initialize the next round of local training. The scalar $\alpha \in [0,1]$ controls the interpolation between global and local parameters, with $\alpha = 1$ corresponding to a full overwrite by the global model and $\alpha = 0$ corresponding to retaining the previous local parameters.

3.5. Local Policy Learning

Figure 3 illustrates the downstream reinforcement learning stage. This component is decoupled from representation learning, with encoder parameters kept fixed. Each client trains an actor–critic agent using locally collected trajectories.

In our experiments, we employ TD3 as the downstream reinforcement learning algorithm. The policy and value networks receive raw states together with frozen state and state–action embeddings as auxiliary inputs. All remaining components of policy optimization, including target networks, replay buffers, and exploration strategies, follow standard TD3 practice.

The pseudocode for local policy learning with frozen representations is defined in Algorithm 2. Each client independently optimizes an actor–critic policy using locally collected environment interactions, while keeping the representation encoders fixed. This stage is fully decoupled from federated training: no parameters are exchanged between clients, and all replay buffers and updates remain local. By isolating policy optimization from federation, the framework avoids synchronizing unstable policy gradients across heterogeneous tasks while still exploiting the shared, dynamics-aware representations learned in the federated stage.

 doi:10.20944/preprints202601.2257.v1

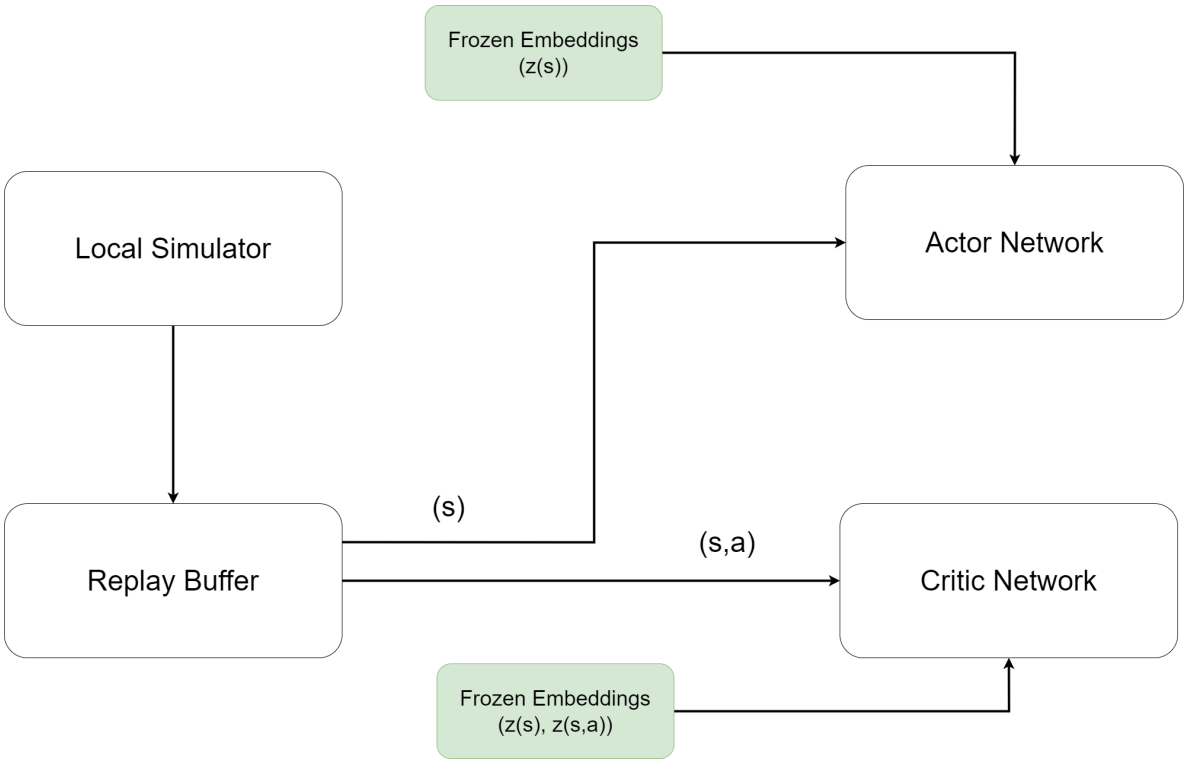


Figure 3. Local Policy Learning

Algorithm 2 TD3 Policy Learning with Frozen Dynamics-Aware Representations

Require: Frozen encoders $(z(\cdot), z(\cdot,\cdot))$ with parameters θ, actor π_ϕ, critics Q_{ψ_1}, Q_{ψ_2}

for all clients i **do**
  Initialize actor and twin critics; initialize target networks $\pi_{\phi'}, Q_{\psi_1'}, Q_{\psi_2'}$
  Initialize replay buffer $\mathcal{B}_i$
  for each training step **do**
    Observe state s_t
    Compute state embedding $z_t = z(s_t)$
    Select action with exploration noise:
      $a_t = \pi_\phi(s_t, z_t) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$
    Execute a_t, observe reward r_t and next state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}_i$
    if update step **then**
      Sample minibatch (s, a, r, s') from $\mathcal{B}_i$
      Compute embeddings $z = z(s)$ and $z^a = z(s, a)$
      Compute target action with smoothing:
        $\tilde{a} = \pi_{\phi'}(s', z(s')) + \text{clip}(\epsilon', -c, c)$
      Compute target value:
        $y = r + \gamma \min(Q_{\psi_1'}(s', \tilde{a}, z(s', \tilde{a}), z(s')), Q_{\psi_2'}(s', \tilde{a}, z(s', \tilde{a}), z(s')))$
      Update both critics Q_{ψ_1} and Q_{ψ_2} by minimizing the TD error between $Q_{\psi_j}(s, a, z^a, z)$ and target y, for $j \in \{1, 2\}$
      if delayed policy update **then**
        Actor update using $a_\phi = \pi_\phi(s, z)$ and $z_\phi^a = z(s, a_\phi)$
        Update target networks with Polyak averaging
      end if
    end if
  end for
end for

 doi:10.20944/preprints202601.2257.v1

3.6. Scope and Extensions

The proposed framework is designed for federated reinforcement learning settings in which all clients share identical observation and action spaces and operate on robots with comparable morphology, while exhibiting partially overlapping transition dynamics. This setting reflects many practical multi-task robotic scenarios, such as a single manipulator performing diverse manipulation behaviours in a shared workspace, where task heterogeneity arises primarily from differences in contact dynamics, constraints, and reward functions rather than embodiment.

Within this scope, the framework assumes access to offline trajectories for representation learning at each client. These trajectories remain private and are never shared across clients, preserving data privacy while enabling federated alignment of latent dynamics representations. The approach is therefore most effective when sufficient offline data is available to learn stable dynamics-aware embeddings before downstream policy optimization.

Several natural extensions of the proposed framework are possible. First, extending F-DRL to heterogeneous embodiments (e.g., different robot arms, grippers, or kinematic structures) would require task-specific front-end encoders followed by a shared latent projection space, enabling alignment at the representation level while preserving embodiment-specific features. Second, the current formulation assumes proprioceptive state inputs; incorporating visual observations would require combining the proposed dynamics-aware objectives with perceptual encoders and cross-modal alignment mechanisms.

Third, the similarity-based aggregation mechanism could be made adaptive over training time. In early federated rounds, when representations are still evolving and similarity estimates are noisy, aggregation naturally approaches uniform averaging. As representations stabilize, increasing the influence of similarity-weighted aggregation may further improve selectivity and reduce negative transfer. Finally, extending the framework to continual and lifelong federated learning settings, where new tasks arrive over time, would allow investigation of how latent dynamics geometry can support incremental knowledge accumulation without catastrophic interference.

These extensions are conceptually orthogonal to the core framework and can be incorporated without modifying the decoupled representation–control architecture, making F-DRL a flexible foundation for scalable and robust federated robotic learning.

4. Experimental Evaluation and Discussion

4.1. Experimental Setup

We evaluate the proposed framework on a suite of robotic manipulation tasks from the MetaWorld benchmark [56]. The tasks considered include *button-press-topdown*, *door-open*, *drawer-open*, *drawer-close*, *reach*, *window-open*, and *window-close*. These tasks span a range of contact-rich and articulated manipulation behaviours, introducing heterogeneity in transition dynamics while sharing identical state and action spaces.

We consider a federated learning setting with $N = 7$ clients, where each client corresponds to a distinct task instance. Each episode is capped at 500 environment steps. Clients perform 20 local representation learning epochs per federated round, and representation models are aggregated over 100 federated rounds. Offline datasets for representation learning consist of a mixture of random and expert trajectories and remain local to each client throughout training.

For downstream control, each task is trained for 200,000 environment interaction steps using three independent random seeds. Performance is evaluated using task success rate, which directly reflects successful task completion.

Representation networks.

The state encoder E_θ is implemented as a multilayer perceptron with two hidden layers of 64 units and ReLU activations, mapping the raw state to a latent embedding of dimensionality $D = 16$. The action-conditioned module F_ϕ is also a multilayer perceptron with two hidden layers of 64 units.

 doi:10.20944/preprints202601.2257.v1

It takes as input the concatenation of the state embedding and the executed action, i.e., $[z(s), a]$, and outputs a predicted next-latent embedding $\hat{z}(s, a) \in \mathbb{R}^{16}$. Both modules are trained using the Adam optimizer with a learning rate of 10^{-4}. During local representation learning, a batch size of 128 is used. Client summary representations for federated aggregation are computed by averaging Gram-based fingerprints over 50 validation batches of size 64.

Downstream reinforcement learning networks.

For policy learning, we employ a TD3 style actor–critic architecture augmented with frozen representation inputs. The actor network is implemented as a multilayer perceptron with two hidden layers of 256 units. It takes as input the raw state together with the frozen state embedding $z(s)$ and outputs a continuous action via a tanh activation. The critic consists of twin Q-networks following the TD3 formulation. Each critic receives the raw state–action pair together with frozen state and state–action embeddings, i.e., $(s, a, z(s), z(s, a))$, and outputs a scalar Q-value.

We use standard TD3 hyperparameters: discount factor $\gamma = 0.99$, target smoothing coefficient $\tau = 0.005$, policy delay of 2, batch size 256, and Gaussian exploration noise with standard deviation 0.2. Encoder parameters remain fixed during downstream reinforcement learning.

4.2. Baselines and Evaluation Protocol

We compare the proposed method against the following baselines. All baselines use the same offline datasets, downstream reinforcement learning algorithm, network architectures, optimization hyperparameters, and random seeds; only the representation learning and aggregation mechanisms differ.

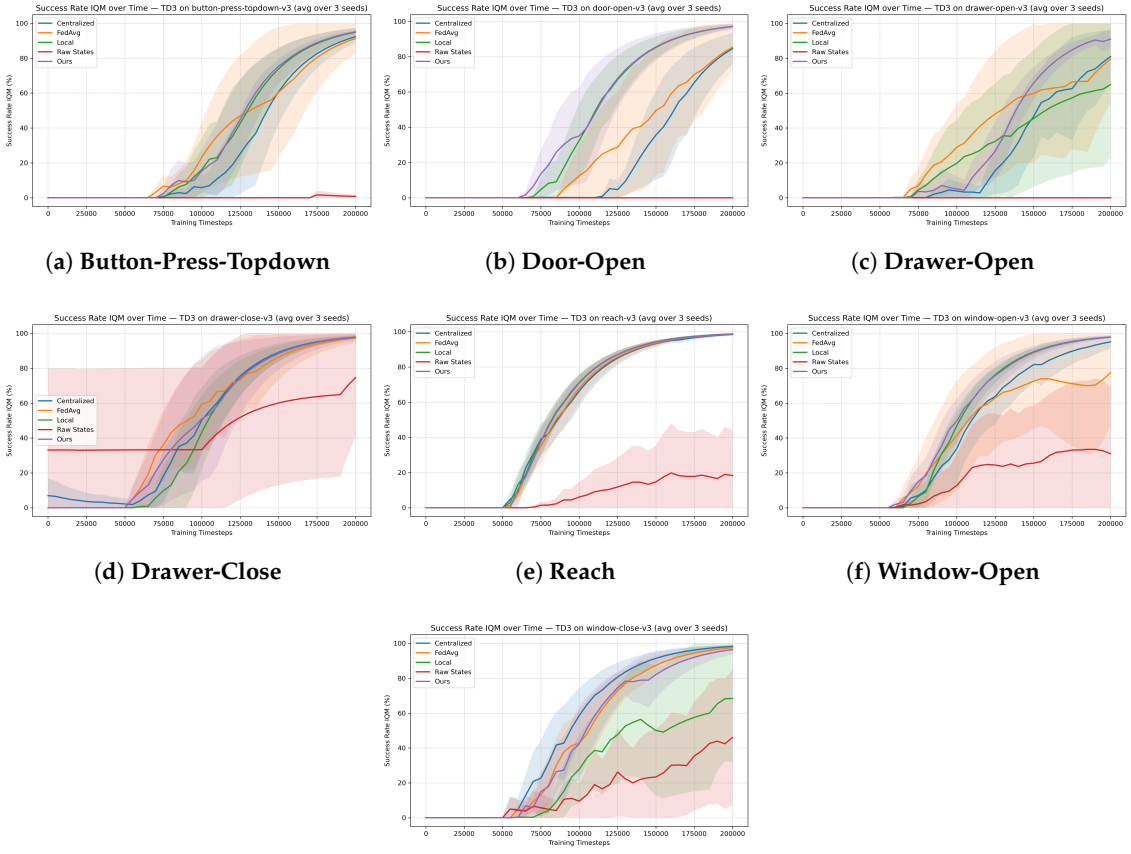
1. **Centralized:** Offline trajectories from all tasks are pooled to train a single state encoder and state–action encoder without privacy constraints. Training proceeds for 20 epochs per round over 100 rounds and serves as an upper bound on representation sharing.
2. **FedAvg:** Client state encoder parameters are aggregated using equal weights for each client at each federated round following the standard FedAvg procedure. All other training components are identical to the proposed method.
3. **Local End-to-End Training:** Representation learning and policy optimization are trained jointly for each task in an end-to-end manner, without federated aggregation.
4. **Raw States Only:** Policies are trained directly on raw state observations without any learned representation or auxiliary embedding signals.

4.3. Downstream Reinforcement Learning Performance

Figure 4 reports downstream reinforcement learning performance across the seven MetaWorld manipulation tasks. Although representation learning methods are often evaluated using latent-space proxy metrics, such measures do not necessarily correlate with control performance. We therefore assess learned representations solely through their impact on downstream reinforcement learning.

Across all tasks, the proposed method achieves performance that is competitive with centralized training and standard FedAvg, while consistently outperforming local end-to-end training and raw-state baselines. In tasks with relatively simple dynamics and limited heterogeneity, such as *reach*, performance differences between federated methods are minimal, indicating that uniform aggregation is sufficient when client representations are naturally aligned. In contrast, tasks involving articulated objects and constrained motion, including *drawer-open*, *drawer-close*, and *window-close*, exhibit clearer benefits from representation-aligned aggregation.

 doi:10.20944/preprints202601.2257.v1



(**a**) **Button-Press-Topdown** (**b**) **Door-Open** (**c**) **Drawer-Open**

(**d**) **Drawer-Close** (**e**) **Reach** (**f**) **Window-Open**

(**g**) **Window-Close**

Figure 4. Downstream reinforcement learning performance across seven MetaWorld manipulation tasks. Panels show task success rate as a function of environment steps for (**a**) Button-Press-Topdown, (**b**) Door-Open, (**c**) Drawer-Open, (**d**) Drawer-Close, (**e**) Reach, (**f**) Window-Open, and (**g**) Window-Close. Curves are averaged over three random seeds and shaded regions denote 95% confidence intervals.

4.4. Stability and Effect of Task Heterogeneity

A consistent trend across tasks is that the proposed method exhibits more stable learning dynamics than FedAvg and, in most cases, than local end-to-end training.

While final success rates are often similar, the proposed approach produces smoother learning curves with reduced variance across random seeds. This effect is particularly pronounced in longer-horizon manipulation tasks, where heterogeneous dynamics can induce conflicting gradient updates under naive aggregation. Table 2 quantitatively summarizes stability metrics for representative heterogeneous tasks.Stability metrics are computed over the first 70% of evaluation checkpoints, corresponding to the primary learning phase prior to performance saturation.

Stability metrics are reported only for tasks exhibiting non-trivial inter-task heterogeneity; for tasks that saturate early across all methods, variance-based measures are not informative and are therefore omitted.

 doi:10.20944/preprints202601.2257.v1

Table 2. Stability metrics for heterogeneous MetaWorld tasks.

Task	Method	Final Success (%)	Mean Seed Std ↓	AUC Std ↓
Drawer-Open	Centralized	100.00	7.57	211.83
	FedAvg	99.79	20.11	562.94
	Local	79.43	18.93	530.10
	Ours	98.65	7.72	216.12
Window-Open	Centralized	100.00	10.11	283.04
	FedAvg	100.00	19.06	533.67
	Local	100.00	3.97	111.10
	Ours	100.00	6.49	181.79

Mean Seed Standard Deviation.

To quantify training stability across random initializations, we report the *mean seed standard deviation*, defined as the average standard deviation of task success rates across random seeds, computed at each evaluation checkpoint and averaged over a fixed training window. Lower values indicate more consistent and reproducible learning dynamics.

AUC Standard Deviation.

We additionally report the standard deviation of the area under the learning curve (AUC) across random seeds. This metric captures variability in learning speed and trajectory shape over time, rather than only pointwise disagreement. Lower values indicate more temporally stable and consistent learning behaviour.

Importantly, the proposed method does not uniformly outperform FedAvg across all tasks. In settings with low inter-task heterogeneity, uniform aggregation performs competitively, suggesting that selective weighting is not always necessary. However, as task dynamics diverge, similarity-weighted aggregation becomes increasingly beneficial, mitigating negative transfer between incompatible tasks. This adaptive behaviour is desirable in federated robotic settings, where task similarity cannot be assumed *a priori*.

The observed performance trends are consistent with the behaviour of the representation-aligned aggregation mechanism. Clients whose latent spaces exhibit similar geometric structure naturally receive comparable aggregation weights, while clients with divergent dynamics are softly down-weighted. By relying on second-order latent geometry rather than first-order statistics, the aggregation process remains robust to representation noise and sign ambiguities.

4.5. Analysis of Similarity-Weighted Federated Aggregation

To interpret the behaviour of the proposed similarity-weighted aggregation strategy, we analyse the evolution of the server-side client weights across federated rounds. During training, the server logs the aggregation weights $\{w_i^{(r)}\}_{i=1}^N$ at each round r (see Section 3 for the weight computation). For analysis, we load the per-round weight vectors saved by the server (one file per round) and stack them into a matrix $W \in \mathbb{R}^{R\times N}$, where $W_{r,i} = w_i^{(r)}$ denotes the contribution of client i at federated round r.

For interpretability, we normalise weights within each round such that $\sum_{i=1}^N W_{r,i} = 1$ and sort clients by their mean weight across rounds, i.e., in descending order of $\frac{1}{R}\sum_{r=1}^R W_{r,i}$. Figure 5 visualises the resulting matrix as a heat map, where rows correspond to tasks (clients) and columns correspond to federated rounds.

The heat map shows that aggregation is not uniform and exhibits structured, smooth evolution over training. In particular, a subset of tasks consistently receives higher weights, indicating that the similarity estimator assigns greater influence to clients whose latent dynamics summaries are more aligned with the global reference. This provides direct evidence that F-DRL performs selective

 doi:10.20944/preprints202601.2257.v1

representation sharing rather than uniform averaging, supporting the claim that similarity-weighted aggregation mitigates negative transfer under task heterogeneity.

Figure 5. Evolution of task-wise aggregation weights across federated rounds.

4.6. Discussion

Overall, the experimental results indicate that representation-aligned aggregation enables stable and effective knowledge sharing in federated reinforcement learning, particularly under task heterogeneity. Importantly, the primary benefit of the proposed approach does not lie in uniformly improving peak task performance, but rather in shaping the *learning dynamics* themselves. Across heterogeneous manipulation tasks, similarity-weighted aggregation consistently reduces variance across random seeds and produces smoother, more predictable learning curves, indicating improved optimization stability and reproducibility.

This behaviour highlights a key distinction between representation-level and policy-level federation. While uniform aggregation (FedAvg) can be effective when client tasks are naturally aligned, it becomes fragile when transition dynamics diverge, as conflicting gradients are averaged indiscriminately. By contrast, the proposed method selectively aligns clients based on latent dynamics similarity, allowing shared structure to be exploited while softly down-weighting incompatible tasks. The resulting effect is not aggressive transfer, but *controlled sharing*, which mitigates negative transfer without suppressing task-specific specialization.

The task-wise analysis further reveals that the method is neutral when heterogeneity is low and beneficial when heterogeneity is high. In simple tasks such as *reach*, where representations are naturally similar, all aggregation strategies perform comparably. However, in articulated and contact-rich tasks such as *drawer-open*, *drawer-close*, and *window-close*, where dynamics differ more substantially, representation-aligned aggregation leads to markedly more stable learning. This adaptive behaviour is desirable in realistic robotic deployments, where task similarity is rarely known *a priori* and may evolve over time.

The aggregation weight analysis provides direct evidence that the server does not converge to uniform averaging, but instead learns a structured weighting scheme that remains stable across rounds. The persistence of higher weights for a subset of tasks suggests that the learned latent geometry captures meaningful dynamical overlap, supporting the central hypothesis that dynamics-aware representations form a suitable substrate for federated knowledge sharing. Crucially, this selectivity emerges without explicit task labels, reward information, or policy synchronization, reinforcing the value of representation-centric federation.

 doi:10.20944/preprints202601.2257.v1

From a broader perspective, these results suggest that stability, rather than peak performance, should be treated as a first-class objective in federated reinforcement learning. In safety-critical and long-horizon robotic settings, predictable and reproducible learning dynamics are often more valuable than marginal gains in final success rate. The proposed framework offers a practical mechanism to achieve this by decoupling representation alignment from control optimization and grounding aggregation decisions in latent dynamics geometry rather than parameter similarity.

5. Conclusion and Future Work

In this work, we introduced a dynamics-aware federated representation learning framework for robotic control that combines robotics priors, action-conditioned embeddings, and similarity-based aggregation. By aligning representations based on latent geometric structure rather than uniform parameter averaging, our approach enables stable and selective knowledge sharing across heterogeneous tasks while preserving a model-free downstream reinforcement learning setup. Empirical results on a suite of MetaWorld manipulation tasks demonstrate that the proposed method achieves competitive performance with improved training stability under task heterogeneity.

Despite these improvements, our approach has limitations. The computation of client summary representations introduces additional communication overhead, and similarity estimates can be noisy during early training when representations are still evolving. In such cases, aggregation weights tend to be close to uniform, effectively recovering FedAvg behaviour.

Future work may explore adaptive aggregation schedules that gradually increase the influence of similarity-based weighting as representations stabilize, as well as uncertainty-aware weighting strategies that explicitly account for confidence in client summaries. Extending the framework to more diverse robotic platforms and real-world deployments, and investigating tighter connections between representation geometry and theoretical generalization guarantees, are also promising directions.

Author Contributions: Conceptualization, AU; methodology, AU; software,AU; validation, AU; formal analysis, AU and XL; investigation, AU; resources, AU; data curation, AU; writing—original draft preparation, AU; writing—review and editing, AU, XL, YJ, JL, YB. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Informed Consent Statement: Not applicable

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning, 2013, [arXiv:cs.LG/1312.5602].
2. Wang, X.; Wang, S.; Liang, X.; Zhao, D.; Huang, J.; Xu, X.; Dai, B.; Miao, Q. Deep Reinforcement Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* **2024**, *35*, 5064–5078. https://doi.org/10.1109/TNNLS.2022.3207346.
3. Pitkevich, A.; Makarov, I. A Survey on Sim-to-Real Transfer Methods for Robotic Manipulation. In Proceedings of the 2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY), 2024, pp. 000259–000266. https://doi.org/10.1109/SISY62279.2024.10737545.
4. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **2013**, *35*, 1798–1828.
5. Botteghi, N.; Poel, M.; Brune, C. Unsupervised Representation Learning in Deep Reinforcement Learning: A Review. *IEEE Control Systems* **2025**, *45*, 26–68. https://doi.org/10.1109/MCS.2025.3534477.
6. Echchahed, A.; Castro, P.S. A Survey of State Representation Learning for Deep Reinforcement Learning. *Transactions on Machine Learning Research* **2025**. Survey Certification.
7. Jonschkowski, R.; Brock, O. Learning state representations with robotic priors. *Auton. Robots* **2015**, *39*, 407–428. https://doi.org/10.1007/s10514-015-9459-7.

 doi:10.20944/preprints202601.2257.v1

8. Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; de Freitas, N. Dueling Network Architectures for Deep Reinforcement Learning, 2016, [arXiv:cs.LG/1511.06581].
9. van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-learning, 2015, [arXiv:cs.LG/1509.06461].
10. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay, 2016, [arXiv:cs.LG/1511.05952].
11. Schulman, J.; Levine, S.; Moritz, P.; Jordan, M.I.; Abbeel, P. Trust Region Policy Optimization, 2017, [arXiv:cs.LG/1502.05477].
12. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms, 2017, [arXiv:cs.LG/1707.06347].
13. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation, 2018, [arXiv:cs.LG/1506.02438].
14. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning, 2019, [arXiv:cs.LG/1509.02971].
15. Fujimoto, S.; van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods, 2018, [arXiv:cs.AI/1802.09477].
16. Jonschkowski, R.; Brock, O. State Representation Learning in Robotics: Using Prior Knowledge about Physical Interaction. In Proceedings of the Robotics: Science and systems, 2014.
17. Jonschkowski, R.; Hafner, R.; Scholz, J.; Riedmiller, M. Pves: Position-velocity encoders for unsupervised learning of structured state representations. *arXiv preprint arXiv:1705.09805* **2017**.
18. Raffin, A.; Höfer, S.; Jonschkowski, R.; Brock, O.; Stulp, F. Unsupervised learning of state representations for multiple tasks **2016**.
19. Lesort, T.; Seurin, M.; Li, X.; Díaz-Rodríguez, N.; Filliat, D. Deep unsupervised state representation learning with robotic priors: a robustness analysis. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp. 1–8.
20. Morik, M.; Rastogi, D.; Jonschkowski, R.; Brock, O. State representation learning with robotic priors for partially observable environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019, pp. 6693–6699.
21. Botteghi, N.; Obbink, R.; Geijs, D.; Poel, M.; Sirmacek, B.; Brune, C.; Mersha, A.; Stramigioli, S. Low dimensional state representation learning with reward-shaped priors. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 3736–3743.
22. Botteghi, N.; Alaa, K.; Poel, M.; Sirmacek, B.; Brune, C.; Mersha, A.; Stramigioli, S. Low Dimensional State Representation Learning with Robotics Priors in Continuous Action Spaces. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 190–197. https://doi.org/10.1109/IROS51168.2021.9635936.
23. Munk, J.; Kober, J.; Babuška, R. Learning state representation for deep actor-critic control. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), 2016, pp. 4667–4673. https://doi.org/10.1109/CDC.2016.7798980.
24. Zhang, A.; Satija, H.; Pineau, J. Decoupling Dynamics and Reward for Transfer Learning, 2018.
25. Gelada, C.; Kumar, S.; Buckman, J.; Nachum, O.; Bellemare, M.G. DeepMDP: Learning Continuous Latent Space Models for Representation Learning. *ArXiv* **2019**, *abs/1906.02736*.
26. Fujimoto, S.; Chang, W.D.; Smith, E.J.; Gu, S.S.; Precup, D.; Meger, D. For SALE: state-action representation learning for deep reinforcement learning. In Proceedings of the Proceedings of the 37th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 2023; NIPS '23.
27. Fujimoto, S.; D'Oro, P.; Zhang, A.; Tian, Y.; Rabbat, M. Towards General-Purpose Model-Free Reinforcement Learning. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
28. van Hoof, H.; Chen, N.; Karl, M.; van der Smagt, P.; Peters, J. Stable reinforcement learning with autoencoders for tactile and visual data. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 3928–3934. https://doi.org/10.1109/IROS.2016.7759578.
29. Lee, A.X.; Nagabandi, A.; Abbeel, P.; Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems* **2020**, *33*, 741–752.
30. Zintgraf, L.; Schulze, S.; Lu, C.; Feng, L.; Igl, M.; Shiarlis, K.; Gal, Y.; Hofmann, K.; Whiteson, S. Varibad: Variational bayes-adaptive deep rl via meta-learning. *Journal of Machine Learning Research* **2021**, *22*, 1–39.
31. Finn, C.; Tan, X.Y.; Duan, Y.; Darrell, T.; Levine, S.; Abbeel, P. Deep spatial autoencoders for visuomotor learning. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016, pp. 512–519.

 doi:10.20944/preprints202601.2257.v1

32. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* **2020**, *588*, 604–609.
33. Schrittwieser, J.; Hubert, T.; Mandhane, A.; Barekatain, M.; Antonoglou, I.; Silver, D. Online and offline reinforcement learning by planning with a learned model. *Advances in Neural Information Processing Systems* **2021**, *34*, 27580–27591.
34. Ye, W.; Liu, S.; Kurutach, T.; Abbeel, P.; Gao, Y. Mastering atari games with limited data. *Advances in neural information processing systems* **2021**, *34*, 25476–25488.
35. Hansen, N.; Su, H.; Wang, X. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828* **2023**.
36. Watter, M.; Springenberg, J.; Boedecker, J.; Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems* **2015**, *28*.
37. Karl, M.; Soelch, M.; Bayer, J.; Van der Smagt, P. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432* **2016**.
38. Hafner, D.; Pasukonis, J.; Ba, J.; Lillicrap, T.P. Mastering Diverse Domains through World Models. *ArXiv* **2023**, *abs/2301.04104*.
39. Qi, J.; Zhou, Q.; Lei, L.; Zheng, K. Federated reinforcement learning: techniques, applications, and open challenges. *Intelligence & Robotics* **2021**, *1*. https://doi.org/10.20517/ir.2021.02.
40. Jin, H.; Peng, Y.; Yang, W.; Wang, S.; Zhang, Z. Federated Reinforcement Learning with Environment Heterogeneity. In Proceedings of the Proceedings of The 25th International Conference on Artificial Intelligence and Statistics; Camps-Valls, G.; Ruiz, F.J.R.; Valera, I., Eds. PMLR, 28–30 Mar 2022, Vol. 151, *Proceedings of Machine Learning Research*, pp. 18–37.
41. Tehrani, P.; Restuccia, F.; Levorato, M. Federated Deep Reinforcement Learning for the Distributed Control of NextG Wireless Networks. *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)* **2021**, pp. 248–253.
42. Wong, Y.J.; Tham, M.L.; Kwan, B.H.; Owada, Y. FedDdrl: Federated Double Deep Reinforcement Learning for Heterogeneous IoT with Adaptive Early Client Termination and Local Epoch Adjustment. *Sensors* **2023**, *23*. https://doi.org/10.3390/s23052494.
43. Rengarajan, D.; Ragothaman, N.; Kalathil, D.; Shakkottai, S. Federated Ensemble-Directed Offline Reinforcement Learning, 2024.
44. Woo, J.; Shi, L.; Joshi, G.; Chi, Y. Federated offline reinforcement learning: Collaborative single-policy coverage suffices. *arXiv preprint arXiv:2402.05876* **2024**.
45. Na, S.; Rouček, T.; Ulrich, J.; Pikman, J.; Krajník, T.; Lennox, B.; Arvin, F. Federated Reinforcement Learning for Collective Navigation of Robotic Swarms. *IEEE Transactions on Cognitive and Developmental Systems* **2023**, *15*, 2122–2131. https://doi.org/10.1109/TCDS.2023.3239815.
46. Yuan, Z.; Xu, S.; Zhu, M. Federated reinforcement learning for robot motion planning with zero-shot generalization. *Automatica* **2024**, *166*, 111709. https://doi.org/https://doi.org/10.1016/j.automatica.2024.111709.
47. Liu, B.; Wang, L.; Liu, M. Lifelong federated reinforcement learning: A learning architecture for navigation in cloud robotic systems. *IEEE Robotics and Automation Letters* **2019**, *4*, 4555–4562.
48. An, X.; Lin, Y.; Lin, M.; Wu, C.; Murase, T.; Ji, Y. Federated Reinforcement Learning Framework for Mobile Robot Navigation Using ROS and Gazebo. *IEEE Internet of Things Magazine* **2025**, *8*, 45–51. https://doi.org/10.1109/MIOT.2025.3575929.
49. Lu, S.; Cai, Y.; Liu, Z.; Lian, Y.; Chen, L.; Wang, H. A Preference-Based Multi-Agent Federated Reinforcement Learning Algorithm Framework for Trustworthy Interactive Urban Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems* **2025**, *26*, 10131–10145. https://doi.org/10.1109/TITS.2025.3543810.
50. Fu, Y.; Li, C.; Yu, F.R.; Luan, T.H.; Zhang, Y. A Selective Federated Reinforcement Learning Strategy for Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 1655–1668. https://doi.org/10.1109/TITS.2022.3219644.
51. Hafid, A.; Hocine, R.; Guezouli, L.; Abdessemed, M.R. Centralized and Decentralized Federated Learning in Autonomous Swarm Robots: Approaches, Algorithms, Optimization Criteria and Challenges : The Sixth Edition of International Conference on Pattern Analysis and Intelligent Systems (PAIS'24). In Proceedings of the 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), 2024, pp. 1–8. https://doi.org/10.1109/PAIS62114.2024.10541145.

 doi:10.20944/preprints202601.2257.v1

52. Kong, X.; Peng, L.; Rahim, S. Dynamic Service Function Chain (SFC) Deployment for Autonomous Intelligent Systems. *IEEE Transactions on Consumer Electronics* **2025**, *71*, 10776–10785. https://doi.org/10.1109/TCE.2025.3612966.
53. Wang, Y.; Zhong, S.; Yuan, T. Grasp Control Method for Robotic Manipulator Based on Federated Reinforcement Learning. In Proceedings of the 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE), 2024, pp. 1513–1519. https://doi.org/10.1109/ICAACE61206.2024.10549724.
54. Yue, S.; Hua, X.; Deng, Y.; Chen, L.; Ren, J.; Zhang, Y. Momentum-Based Contextual Federated Reinforcement Learning. *IEEE Transactions on Networking* **2025**, *33*, 865–880. https://doi.org/10.1109/TNET.2024.3510352.
55. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, second ed.; The MIT Press, 2018.
56. McLean, R.; Chatzaroulas, E.; McCutcheon, L.; Röder, F.; Yu, T.; He, Z.; Zentner, K.; Julian, R.; Terry, J.K.; Woungang, I.; et al. Meta-World+: An Improved, Standardized, RL Benchmark. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2025.
57. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Nitin Bhagoji, A.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. https://doi.org/10.1561/2200000083.
58. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data, 2023, [arXiv:cs.LG/1602.05629].
59. Bardes, A.; Ponce, J.; LeCun, Y. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In Proceedings of the International Conference on Learning Representations, 2022.