Grouped Adaptive Head Mixing for Personalized Multi-Task Federated Reinforcement Learning

Yiran Pang **Zhen Ni** **Dimitris Pados** **Xiangnan Zhong***

Florida Atlantic University
{ypang2022, zhenni, dpados, xzhong}@fau.edu

Abstract

Multi-task reinforcement learning (MT-RL) trains a single agent to solve multiple tasks by leveraging shared knowledge across task domains. However, most MT-RL methods assume centralized access to task data, making them impractical for privacy-sensitive or large-scale distributed settings. Multi-task federated reinforcement learning (MT-FRL) mitigates this issue by enabling agents to collaborate through shared models rather than raw trajectories, but often suffers from unstable performance under task heterogeneity, negative transfer, and imbalanced learning across tasks. A fundamental question in MT-FRL is how each agent should decide with whom to share skills and how strongly to rely on shared knowledge. To answer this question, we propose GAdapHeadFRL, a personalized MT-FRL framework based on geometry-aware grouping and adaptive decision-layer task-head mixing. First, we decompose the local Bellman residual into representation and task-head estimation errors, and derive a closed-form optimal sharing strength governed by task mismatch and estimation uncertainty. Second, we instantiate this insight by identifying compatible client groups from local updated geometry, constructing group-specific task-head prototypes, and dynamically mixing them with local heads using lightweight practical proxies. Experiments on heterogeneous benchmarks, including MiniGrid (up to 12 tasks) and MetaWorld (up to 50 tasks), show that GAdapHeadFRL consistently outperforms state-of-the-art personalized MT-FRL baselines. It achieves up to 24.5% relative improvement on MiniGrid, increases the converged-task ratio from 50.0% to 72.9% over the personalized baseline. In addition, the proposed method reduces the average performance standard deviation by over 50% compared with centralized MT-RL references. These results demonstrate that geometry-aware grouping and adaptive task-head mixing provide an effective and scalable principle for heterogeneous MT-FRL.

https://yiranpang.github.io/GAdapHeadFRL.github.io/

1 Introduction

Multi-task reinforcement learning (MT-RL) is a powerful paradigm for enabling knowledge sharing and generalization across related tasks. However, most existing MT-RL methods assume centralized access to task data, trajectories, or training environments, which can be impractical in privacy-sensitive or large-scale distributed control systems. Federated reinforcement learning (FedRL) offers a natural alternative by allowing multiple agents to collaboratively improve their decision policies without exchanging raw trajectories. In many realistic deployments, however, agents rarely solve identical tasks. For example, autonomous vehicles may operate under different regional traffic

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

laws, driving objectives, and road geometries (Lu et al., 2025; Zhang et al., 2025; Ma et al., 2026). Similarly, in power system orchestration, independent utility operators must manage regional grids with distinct topological constraints and baseline demand profiles (Moazzeni et al., 2025; Kwon et al., 2025; Shankar et al., 2026). These scenarios require agents to collaborate across heterogeneous but related Markov decision processes (MDPs), where useful knowledge may be shared across tasks but local dynamics and objectives remain distinct (Xiong et al., 2024; Pang et al., 2026). This creates a central tension: federated collaboration can reduce estimation uncertainty and transfer useful decision knowledge across agents, but indiscriminately sharing parameters across incompatible tasks can aggregate conflicting update directions and induce negative transfer (McMahan et al., 2017; Jin et al., 2022).

A prevalent strategy for addressing this heterogeneity in FedRL is to decouple the policy or value network into a shared representation backbone and task-specific decision heads (Collins et al., 2021; Oh et al., 2022; Xiong et al., 2025). The shared backbone captures structural knowledge that can transfer across related tasks, while the decision head absorbs task-specific preferences induced by local dynamics and action requirements (Arivazhagan et al., 2019; D'Eramo et al., 2020; Sun et al., 2022). This architecture mitigates heterogeneity, but leaves open how decision heads themselves should be shared (Cao et al., 2026). Existing methods largely fall into two categories. Several works rely on fully shared decision layers or global parameter averaging to maximize collaboration, though this risks severe negative transfer across diverse local objectives (McMahan et al., 2017; Jin et al., 2022; Yu et al., 2020a; Liu et al., 2021). Conversely, another line of work strictly localizes heads to protect client-specific decision rules (Arivazhagan et al., 2019; Collins et al., 2021; Oh et al., 2022; Xiong et al., 2025). While this avoids incompatible signals, it also prevents decision-level knowledge transfer (Xu et al., 2023; Wang et al., 2026). Moving beyond this binary paradigm requires an adaptive mechanism that regulates how strongly each client should rely on shared decision heads according to client-specific task mismatch and estimation uncertainty.

The success of such adaptive sharing, however, is inherently bottlenecked by the quality of the shared prior. In existing multi-task FedRL settings, tasks are not uniformly related. Instead, they exhibit structured heterogeneity, often forming latent groups with distinct decision patterns and updating geometries (Sattler et al., 2021; Ghosh et al., 2020; Duan et al., 2020; Bo et al., 2025). Consequently, directly averaging decision layer (task-head) parameters across all agents may introduce conflicting signals, producing a compromised global task-head that serves as a weak prior for everyone (McMahan et al., 2017; Jin et al., 2022; Yu et al., 2020a; Liu et al., 2021). When faced with such an incompatible prior, even an optimal adaptive mixing strategy will be forced to reject the shared knowledge, degenerating back to fully localized training. This highlights a key challenge: effective decision-level collaboration cannot rely on indiscriminate global sharing, but should instead be selectively constructed among sufficiently compatible clients (Zhang et al., 2021; Huang et al., 2021; Sattler et al., 2021; Ghosh et al., 2020).

To address this challenge, we reformulate task-head sharing in multi-task FedRL from both theoretical and algorithmic perspectives. Our main contributions are summarized as follows:

- We provide, to the best of our knowledge, the first theoretical characterization of adaptive task-head sharing in MT-FRL. Existing methods typically rely on fully shared heads, fully local heads, or heuristic sharing rules, without clarifying how task-head sharing affects local RL error under heterogeneity. We address this gap by deriving an exact orthogonal decomposition of the local Bellman residual, which separates representation error from the task-head error governed by the mixing coefficient. This decomposition provides a direct theoretical basis for deciding how much decision-level knowledge each client should borrow from others, turning adaptive task-head sharing from a heuristic design choice into a principled client-specific optimization problem.
- We derive the optimal client-specific sharing strength for adaptive task-head mixing. By modeling local and shared heads as stochastic estimators, we prove that the expected task-head error is a strictly convex quadratic function of the mixing coefficient and obtain its closed-form optimum. This result unifies fully shared and fully localized heads as two boundary cases of the same adaptive principle, showing that the optimal mixing strength is governed by task mismatch, estimator bias, estimation noise, and noise coupling.
- We propose GAdapHeadFRL, an efficient algorithm that integrates geometry-aware grouping with practical adaptive task-head mixing. In heterogeneous MT-FRL, effective decision-layer

sharing requires not only deciding how much each client should rely on shared knowledge, but also identifying whom it should share with. GAdapHeadFRL addresses the "whom to share with" problem by clustering agents according to their updating directions and automatically selecting the number of groups using a cosine-silhouette criterion, thereby constructing group-specific head prototypes that better align shared decision knowledge with local task geometry. It then addresses the "how much to share" problem using lightweight practical proxies based on Mahalanobis mismatch and estimation instability, enabling each client to adaptively mix its local head with the assigned group prototype.

- Across MiniGrid and MetaWorld case studies, GAdapHeadFRL improves over state-of-the-art personalized MT-FRL baselines by up to 24.5% and reduces the average performance standard deviation by over 50% compared with centralized MT-RL references. These results demonstrate that compatibility-aware and client-adaptive task-head sharing improves both performance and stability under heterogeneous tasks.

2 Preliminaries

MDPs and multi-task RL. An MDP is defined as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, P is the transition kernel, R is the reward function, and $\gamma \in (0,1)$ is the discount factor. Given a policy π, the action-value function is $Q^\pi(s,a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s, a_0 = a, \pi]$. The optimal action-value function $Q^\star$ is the fixed point of the optimal Bellman operator $(\mathcal{T}^\star Q)(s,a) = \int_{\mathcal{S}} P(s' \mid s,a)[R(s,a,s') + \gamma \max_{a'} Q(s',a')]ds'$ (Sutton et al., 2018). We consider a collection of N related tasks $\{\mathcal{M}_i\}_{i=1}^N$, where each task $\mathcal{M}_i = \langle \mathcal{S}_i, \mathcal{A}_i, P_i, R_i, \gamma \rangle$ has its own dynamics and rewards. Let $\mathcal{T}_i^\star$ and $Q_i^\star$ denote the corresponding optimal Bellman operator and optimal action-value function. The multi-task objective is to learn task-specific policies that perform well on average across tasks while allowing task-dependent decision rules (Teh et al., 2017).

Multi-task AVI bound. Following the multi-task approximate value iteration (AVI) framework of D'Eramo et al. (2020), let $(Q_{i,k})_{k=0}^K$ be any bounded value-function sequence with $\|Q_{i,k}\|_\infty \le Q_{\max}$ and $Q_{\max} \le R_{\max}/(1-\gamma)$. Define the local Bellman residual as $\varepsilon_{i,k} := \|Q_{i,k+1} - \mathcal{T}_i^\star Q_{i,k}\|_{\nu_i}^2$ and the average residual as $\varepsilon_{\text{avg},k} := \frac{1}{N}\sum_{i=1}^N \varepsilon_{i,k}$. Then the average performance gap of the induced greedy policies $\{\pi_{i,K}\}_{i=1}^N$ satisfies

$$\frac{1}{N}\sum_{i=1}^{N} \left\| Q_i^\star - Q_i^{\pi_{i,K}} \right\|_{1,\rho_i} \le \frac{2\gamma}{(1-\gamma)^2} \left[\inf_{r \in [0,1]} C_{\text{VI}}^{1/2}(K;r) \sum_{k=0}^{K-1} \beta_k \varepsilon_{\text{avg},k}^{1/2} + 2\gamma^K Q_{\max} \right]. \quad (1)$$

Here C_{VI} is the worst-case concentrability coefficient across tasks, β_k are discount-dependent weights, and ν_i and ρ_i are the local sampling and evaluation distributions, respectively (Farahmand, 2011; D'Eramo et al., 2020). Equation 1 motivates our subsequent analysis: for fixed concentrability and discount-dependent constants, reducing $\varepsilon_{\text{avg},k}$ tightens the average multi-task performance bound.

Multi-task federated RL. We study the same multi-task RL objective under a federated training protocol, where each task $\mathcal{M}_i$ is assigned to a client that collects trajectories and performs RL updates locally, while a central server aggregates model information rather than raw trajectories (Qi et al., 2021; Jin et al., 2022). Under the standard conditions used in FedRL analysis (Jin et al., 2022), this protocol still induces a bounded task-specific value-function sequence $\{Q_{i,k}\}_{k=0}^K$ for each client. Thus, Eq. 1 remains applicable by evaluating the local Bellman residual $\varepsilon_{i,k}$ on each client and averaging across tasks. Our subsequent analysis focuses on how federated task-head sharing affects these residuals under task heterogeneity.

3 Federated Task-Head Sharing: A New Problem Formulation and Theory

In this section, we analyze how mixing local and shared task heads affects the local Bellman residual in Eq. 1 and derive the theoretically optimal dynamic mixing weight for each client. Let $\varphi_\theta(s,a) \in \mathbb{R}^d$ denote the state-action representation produced by the globally synchronized backbone θ. Client i maintains a local head $\hat{\phi}_{i,k}$, while the server aggregates client heads to form a shared head $\hat{\phi}_{g,k}$. At round k, client i evaluates actions using the mixed value function $Q_{i,k}(s,a;\theta,\alpha) = \langle \varphi_\theta(s,a), \tilde{\phi}_{i,k}(\alpha) \rangle$, where $\tilde{\phi}_{i,k}(\alpha) = (1-\alpha)\hat{\phi}_{i,k} + \alpha\hat{\phi}_{g,k}$ and $\alpha \in [0,1]$ controls the strength of

decision-layer sharing. The following analysis characterizes how this mixing coefficient affects, and can be chosen to reduce, the local Bellman residual.

Isolating the Impact of Head Mixing via Bellman Decomposition. For a given global representation θ in each round, the local Bellman residual can be decomposed into a representation error and a task-head estimation error, which isolates the effect of α in the latter term. Let $\phi^\star_{i,k}(\theta) = \arg\min_\phi \mathbb{E}_{\nu_i}[(\langle\varphi_\theta, \phi\rangle - \mathcal{T}_i^\star Q_{i,k})^2]$ be the optimal local projection head, and let $\Sigma_i(\theta) = \mathbb{E}_{\nu_i}[\varphi_\theta \varphi_\theta^\top]$ be the feature covariance matrix.

Lemma 1 (Orthogonal Decomposition)**.** *For any $\alpha \in [0,1]$ and fixed θ, the Bellman residual satisfies $\varepsilon_{i,k}(\alpha) = \varepsilon_{\mathrm{rep},i,k}(\theta) + \big\|\tilde{\phi}_{i,k}(\alpha) - \phi^\star_{i,k}(\theta)\big\|^2_{\Sigma_i(\theta)}$, where $\varepsilon_{\mathrm{rep},i,k}(\theta)$ is independent of α.*

Characterizing Task Mismatch and Estimation Uncertainty. We analyze the head estimation error by decomposing it into task mismatch and estimation uncertainty. For readability, we suppress the dependence on (k, θ). Following Jin et al. (2022), define the optimal global head as $\phi^\star_g = \sum_{j=1}^N p_j \phi^\star_j$, where $p_j \geq 0$ and $\sum_j p_j = 1$. The oracle mismatch between the global and local heads is $b_{i,g} := \phi^\star_g - \phi^\star_i$. In practice, only stochastic estimates are available: $\hat{\phi}_i = \phi^\star_i + m_i + \xi_i$ and $\hat{\phi}_g = \phi^\star_g + m_g + \xi_g$, where m_i, m_g are deterministic biases and ξ_i, ξ_g are zero-mean noises. We define $V_i := \mathbb{E}\|\xi_i\|^2_{\Sigma_i}$, $V_g := \mathbb{E}\|\xi_g\|^2_{\Sigma_i}$, and $C_{ig} := \mathbb{E}[\xi_i^\top \Sigma_i \xi_g]$.

Assumption 1 (Imperfect Noise Coupling)**.** *Assume $V_i > 0$, $V_g > 0$, and there exists $\rho \in [0,1)$ such that $|C_{ig}| \leq \rho\sqrt{V_i V_g}$.*

This assumption rules out the degenerate case of perfectly coupled estimation noise. In MT-FRL, the global head aggregates stochastic updates from multiple clients with different tasks, making its noise unlikely to be an exact copy of any single client's local noise.

Optimal Task-head Mixing. Under Assumption 1, we characterize the expected head error as a function of the mixing coefficient. Conditioned on θ, define $\mathcal{E}_{i,k}(\alpha) := \mathbb{E}[\|\tilde{\phi}_{i,k}(\alpha) - \phi^\star_{i,k}(\theta)\|^2_{\Sigma_i(\theta)}]$, and suppress the round index k in the right side of the equation below.

Theorem 1 (Optimal Adaptive Mixing)**.** *Under Assumption 1, $\mathcal{E}_{i,k}(\alpha)$ is a strictly convex quadratic in α. Its unconstrained minimizer is*

$$\alpha_{\mathrm{unc},i,k} = \frac{V_i - C_{ig} - \langle m_i,\ m_g + b_{i,g} - m_i\rangle_{\Sigma_i}}{\|m_g + b_{i,g} - m_i\|^2_{\Sigma_i} + V_i + V_g - 2C_{ig}},$$

and the constrained optimum is $\alpha^\star_{i,k} = \Pi_{[0,1]}(\alpha_{\mathrm{unc},i,k})$.

The proof is provided in Appendix 7. Theorem 1 shows that the task-head mixing weight is not a free hyperparameter, but the solution to a client-specific bias–variance trade-off.

Boundary regimes. The adaptive head $\tilde{\phi}_i(\alpha) = (1-\alpha)\hat{\phi}_i + \alpha\hat{\phi}_g$ unifies two common FedRL regimes. When task heterogeneity $b_{i,g}$ is small and local estimation noise V_i is large, the variance-reduction benefit of sharing dominates, pushing $\alpha^\star_i$ toward 1 and recovering shared-head, FedAvg-style learning. Conversely, when task mismatch is large, the global head induces a large bias, pushing $\alpha^\star_i$ toward 0 and recovering the personalized local-head regime. Thus, fully shared and fully local heads are two boundary cases of the same adaptive mixing principle in MT-FRL. This perspective also explains why a fixed global α is generally suboptimal under heterogeneous tasks.

4 Proposed Method: Grouped Adaptive Head Mixing

We translate the theory in Section 3 into a practical MT-FRL algorithm. The key challenge is that adaptive mixing requires a compatible shared prior: a single global head may average conflicting task directions, increasing the mismatch between local and shared heads and pushing the optimal mixing weight toward the local extreme. GAdapHeadFRL addresses this by first grouping clients using warmup update signatures to construct group-specific shared heads, and then computing a client-specific mixing coefficient from empirical proxies of mismatch and uncertainty.

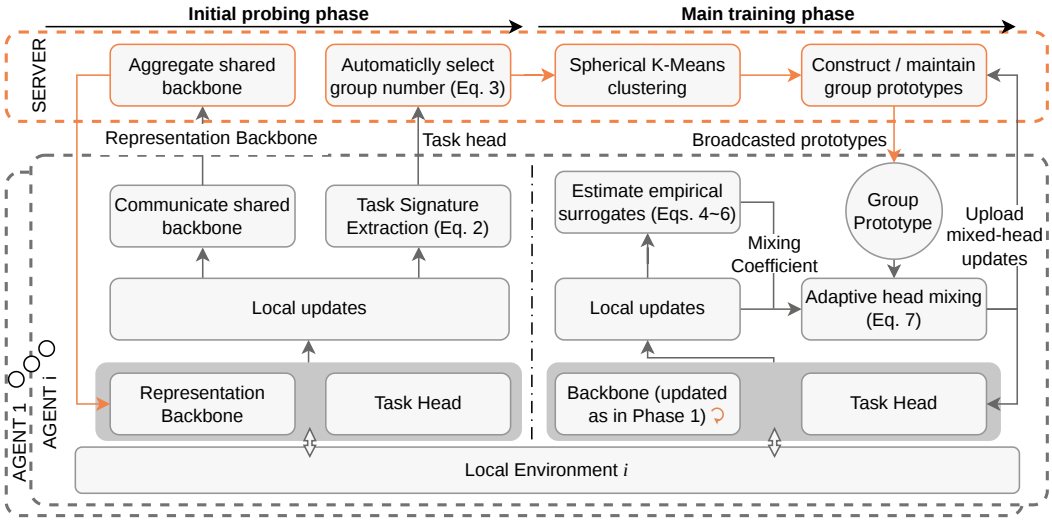


Figure 1: System overview of GAdapHeadFRL. During initial phase, clients perform local updates and send task signatures to the server for automatic grouping. In main training loop, the server maintains group-specific head prototypes and broadcasts them together with the shared backbone. Each client locally computes the adaptive mixing coefficient $\alpha_{i,k}$, updates under the mixed head, and returns post-update backbone and head parameters. The design separates group assignment from sharing intensity: the server selects the prototype, while the client controls how strongly to use it.

4.1 Mitigating Task Conflict via Automatic Grouping

Task signature extraction. During the first W initial rounds, clients train with fully local heads ($\alpha_{i,k} = 0$). Each client forms an update-direction signature by applying an EMA to the normalized concatenation of its actor and critic head updates:

$$u_{i,k} = \begin{bmatrix} \Delta\phi_{i,k}^{\text{actor}} \\ \Delta\phi_{i,k}^{\text{critic}} \end{bmatrix}, \quad v_{i,k} = \frac{u_{i,k}}{\|u_{i,k}\|_2}, \quad \bar{v}_{i,k} \leftarrow \frac{\gamma_v \bar{v}_{i,k-1} + (1-\gamma_v) v_{i,k}}{\|\gamma_v \bar{v}_{i,k-1} + (1-\gamma_v) v_{i,k}\|_2}. \tag{2}$$

Automatic grouping. At round W, the server clusters $\{\bar{v}_{i,W}\}_{i=1}^{N}$ by spherical k-means, since the signatures encode update directions. The group number is selected by maximizing the cosine-silhouette score over $M \in [M_{\min}, M_{\max}]$:

$$\text{Sil}(M) = \frac{1}{N} \sum_{i=1}^{N} \frac{b_i(M) - a_i(M)}{\max\{a_i(M), b_i(M)\} + \epsilon}, \qquad M^\star = \underset{M}{\text{argmax}}\, \text{Sil}(M). \tag{3}$$

Group Assignment and Prototype Initialization. This clustering assigns each client i to a specific group, denoted by $G_i \in \{1, \ldots, M^\star\}$. The centroids are used only to determine these hard assignments in the directional signature space. The server then initializes and independently maintains one shared head for each group, defining these as group-specific head prototypes. Client i subsequently uses the prototype of its assigned group, $\hat{\phi}_{G_i,k}$, as its tailored shared prior for the adaptive mixing phase.

4.2 Practical Implementation of Adaptive Task-Head Mixing

With group prototypes established, each client must decide how strongly to rely on its assigned prototype. Theorem 1 suggests that the optimal weight balances local uncertainty against prototype mismatch $\|\Delta_m\|_{\Sigma_i}^2$ and shared uncertainty V_g. Since the exact bias, variance, and covariance terms are unobservable, we use the first-order surrogate

$$\alpha_i^\star \approx \frac{V_i}{\|\Delta_m\|_{\Sigma_i}^2 + V_i + V_g}, \qquad \Delta_m = b_{i,g} + m_g - m_i, \tag{4}$$

which preserves the predicted monotonic behavior: sharing increases with local uncertainty and decreases with prototype mismatch or shared uncertainty.

We instantiate the surrogate using observable proxies. First, client i estimates a diagonal feature geometry from a local mini-batch $\mathcal{B}_{i,k}$: $\hat{\Sigma}_{i,k} = \mathrm{diag}(|\mathcal{B}_{i,k}|^{-1} \sum_{(s,a)\in\mathcal{B}_{i,k}} \varphi_\theta(s,a) \odot \varphi_\theta(s,a)) + \eta I$, regularized by $\eta > 0$. With smoothed local and group heads $\bar{\phi}_{i,k}$ and $\bar{\phi}_{G_i,k}$, we define

$$\begin{aligned} \hat{H}_{i,k} = \|\bar{\phi}_{G_i,k} - \bar{\phi}_{i,k}\|^2_{\hat{\Sigma}_{i,k}}, \quad \hat{V}_{i,k} = (1-\lambda_V)\hat{V}_{i,k-1} + \lambda_V \|\hat{\phi}_{i,k} - \bar{\phi}_{i,k-1}\|^2_{\hat{\Sigma}_{i,k}}, \\ \hat{V}_{G_i,k} = (1-\lambda_G)\hat{V}_{G_i,k-1} + \lambda_G \|\hat{\phi}_{G_i,k} - \bar{\phi}_{G_i,k-1}\|^2_{\hat{\Sigma}_{i,k}} \end{aligned} \tag{5}$$

where $\hat{H}_{i,k}$ serves as an observable proxy for how incompatible the current shared prototype is with the local task, $\hat{V}_{i,k}$ approximates the local estimation uncertainty (V_i), and $\hat{V}_{G_i,k}$ approximates the shared-head uncertainty (V_g). λ_V and λ_G are exponential moving averages rates. Substituting these proxies into Equation 4 yields the practical adaptive mixing rule:

$$\alpha_{i,k} = \frac{\hat{V}_{i,k}}{\hat{H}_{i,k} + \hat{V}_{i,k} + \hat{V}_{G_i,k} + \epsilon}. \tag{6}$$

This rule preserves the same bias–variance trade-off as Theorem 1: $\alpha_{i,k}$ increases when local estimation becomes unstable, encouraging stronger reliance on the shared prototype, and decreases when either the prototype mismatch or the shared-head instability becomes large, thereby preventing negative transfer. Finally, client i updates its Q-values using the dynamically mixed head

$$\tilde{\phi}_{i,k} = (1-\alpha_{i,k})\hat{\phi}_{i,k} + \alpha_{i,k}\hat{\phi}_{G_i,k}. \tag{7}$$

During the subsequent local RL update phase, client i optimizes its parameters using trajectories collected under the mixed head $\tilde{\phi}_{i,k}$. This produces the post-update local head $\hat{\phi}^+_{i,k}$ and the updated representation backbone $\theta^+_{i,k}$, which are then transmitted to the server.

4.3 Overall Training Pipeline and Server-Side Aggregation

Figure 1 summarizes the two-phase pipeline. During initial phase, clients use local heads and send update signatures for grouping. During main training, each client receives the global backbone and its assigned group prototype, computes $\alpha_{i,k}$ locally, updates under the mixed head, and returns the post-update backbone and head to the server. The group assignment G_i determines which prototype a client receives, while the sharing intensity is controlled only by $\alpha_{i,k}$. Let $\mathcal{C}_{r,k} = \{i \in \mathcal{S}_k \mid G_i = r\}$ denote the selected clients $\mathcal{S}_k$ in group r at round k. The server updates group prototypes by averaging post-update heads within each group and updates the backbone by federated averaging:

$$\hat{\phi}_{r,k+1} = \frac{1}{|\mathcal{C}_{r,k}|} \sum_{i\in\mathcal{C}_{r,k}} \hat{\phi}^+_{i,k}, \qquad \theta_{k+1} = \frac{1}{|\mathcal{S}_k|} \sum_{i\in\mathcal{S}_k} \theta^+_{i,k}. \tag{8}$$

For clarity, we denote the full framework with both automatic grouping and adaptive head mixing as GAdapHeadFRL, and the variant without grouping as AdapHeadFRL.

5 Experiments

5.1 Benchmark Results on MiniGrid

Benchmarks. For MiniGrid (Chevalier-Boisvert et al., 2023), we construct a unified multi-task benchmark from a 12-task pool, denoted as **MT12**, selected from the original MiniGrid environment suite. We retain environments that are compatible with a shared visual MTRL setting and exclude instruction-centric tasks whose successful completion depends heavily on textual mission understanding. This yields a broader task pool covering navigation, obstacle avoidance, memory, door-key interaction, and multi-room exploration under a common partially observable visual interface. To preserve direct comparability with prior work, we keep the original MiniGrid task choices used by Hendawy et al. (2024), denoted as **MT5** and **MT7**. Based on the same MT12 pool, we further

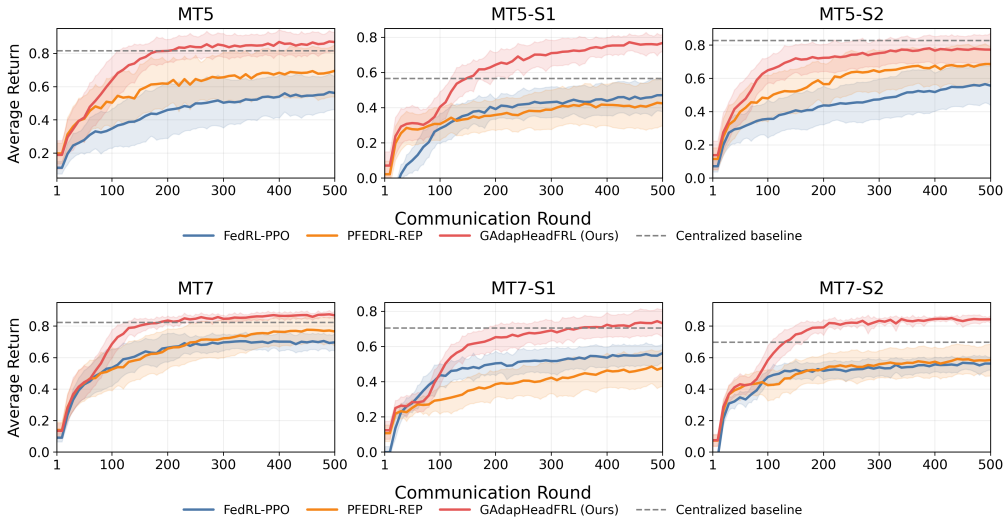


Figure 2: Average return on MiniGrid under two benchmark scales. Top: MT5, including the original split and two additional random sampling splits (MT5-S1/S2). Bottom: MT7, including the original split and two additional random sampling splits (MT7-S1/S2). We report the mean and the 95% confidence interval over 30 independent runs. The dashed horizontal line denotes the performance of the centralized baseline MOORE (Hendawy et al., 2024).

construct two additional random sampling splits for each scale, yielding **MT5-S1/S2** and **MT7-S1/S2**. As a result, each benchmark scale is evaluated on three scenarios with the same task cardinality but different task compositions and overlaps, allowing us to test not only performance on standard settings but also robustness under alternative task combinations. The detailed task identities of all splits are provided in Appendix 10.1.

Baselines. We compare against both centralized multi-task RL methods and federated RL baselines. For centralized training, we include MOORE (Hendawy et al., 2024), a representative modular MTRL method, and the recent centralized MTRL architecture of McLean et al. (2025). For the federated setting, we compare with FedRL-PPO (Pang et al., 2025), which performs regular federated policy learning based on PPO, and PFEDRL-REP (Xiong et al., 2025), a personalized federated RL method that shares the representation backbone while keeping task-specific heads local. This set of baselines enables comparison against strong centralized MTRL references as well as privacy-preserving federated alternatives under the same MiniGrid benchmark.

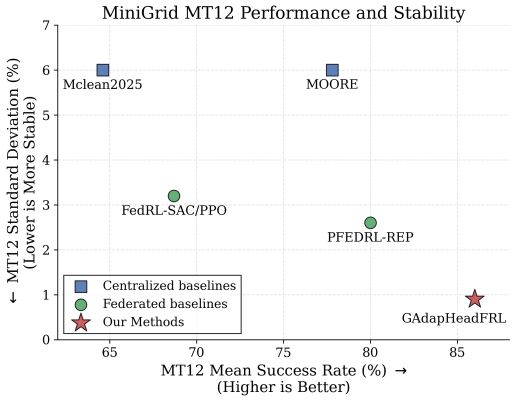


Figure 3: Each point shows the mean success rate and standard deviation of a method. The bottom-right corresponds to the desirable regime of higher mean performance and lower variability.

MiniGrid Results. We evaluate all methods on MT12 and on the three-set versions of MT5 and MT7, where MT5 and MT7 average over the original split and two additional randomly sampled splits. Figure 2 reports the average return curves with 95% confidence intervals across 30 runs, and Table 1 summarizes the final average performance. These results evaluate both learning efficiency under a fixed local environment-interaction budget and final policy quality across different MiniGrid benchmark scales.

Table 1 and Figures 2 and 3 show that our methods achieve the best performance in every MiniGrid scenario and lower variance under heterogeneous task compositions. GAdapHeadFRL outperforms the strongest personalized MT-FRL baseline in each setting, with the largest gain on MT7, where

Table 1: Main results on MetaWorld and MiniGrid. All values are reported as percentages (%). For MT10, MT5, and MT7, we report the average over three sets. Entries are shown as mean ± standard deviation. The upper and lower blocks correspond to centralized training and federated learning methods, respectively.

		MetaWorld		MiniGrid		
Method	Venue	MT10 (3Set)	MT50	MT5 (3Set)	MT7 (3Set)	MT12
MOORE (Hendawy et al., 2024)	ICLR	79.2 ± 5.2	72.3 ± 3.3	73.7 ± 8.0	74.3 ± 7.3	77.8 ± 6.0
McLean et al. (2025)	RLJ	80.9 ± 5.4	72.5 ± 3.5	68.0 ± 9.4	66.4 ± 6.8	64.6 ± 6.0
FedRL-SAC/PPO (Pang et al., 2025)	DCS	78.1 ± 7.4	65.6 ± 3.1	57.6 ± 5.9	65.3 ± 4.5	68.7 ± 3.2
PFEDRL-REP (Xiong et al., 2025)	ICLR	78.1 ± 9.5	67.3 ± 1.0	66.2 ± 1.1	66.1 ± 8.2	80.0 ± 2.6
AdapHeadFRL(Ours)	-	80.7 ± 2.2	68.9 ± 0.9	79.1 ± 8.3	81.8 ± 2.7	85.9 ± 0.8
GAdapHeadFRL(Ours)	-	**84.9 ± 3.6**	**72.0 ± 0.6**	**80.8 ± 6.1**	**82.3 ± 3.2**	**86.0 ± 0.9**

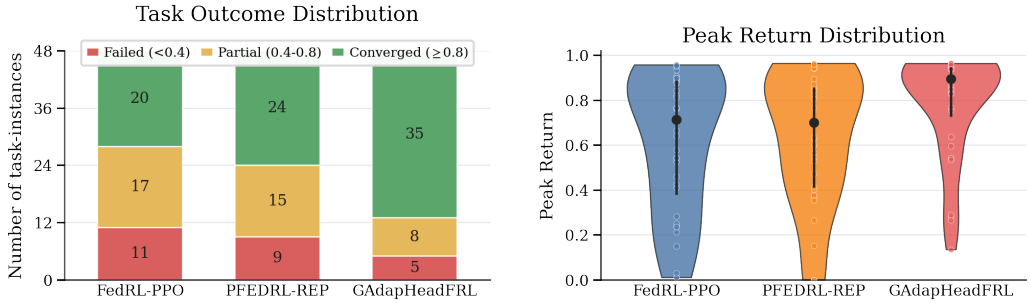


Figure 4: Distributional analysis of task-level outcomes on MiniGrid. Left: Task instances grouped by peak normalized return. GAdapHeadFRL produces fewer failed cases and more converged cases than the baselines, indicating stronger reliability under heterogeneous task compositions. Right: Violin plots of peak returns across all task instances. Our methods shift the distribution upward, reduce the low-return tail, and concentrate more density in the high-performance region, helping explain the lower variability observed in the main results.

it improves over PFEDRL-REP (Xiong et al., 2025) from 66.1% to 82.3%, a 24.5% relative improvement. On MT12, the largest and most heterogeneous MiniGrid setting, our two methods obtain the two highest mean returns and the two lowest standard deviations. GAdapHeadFRL reaches the best mean return with a standard deviation of 0.9, 85.0% lower than that of the strongest centralized MT-RL reference. Compatibility-aware and client-adaptive sharing thus improves both final policy quality and training reliability as task diversity increases.

5.2 Benchmark Results on MetaWorld

Benchmarks. For MetaWorld (Yu et al., 2020b), we use the standard 50-task manipulation suite, denoted as **MT50**. To maintain comparability with prior work, we retain the original 10-task split in Hendawy et al. (2024) as **MT10**. We further construct two additional 10-task splits, **MT10-S1** and **MT10-S2**, by sampling from the MT50 tasks. These additional splits allow us to evaluate robustness under alternative task compositions with the same task cardinality. The task composition of all splits is given in Appendix 10.2.

Baselines. We use the same centralized baselines as in MiniGrid. For the federated setting, we implement FEDRL-SAC by replacing the local PPO learner in FEDRL-PPO (Pang et al., 2025) with SAC, while keeping the same federated training and communication structure. This better matches the off-policy continuous-control setting of MetaWorld.

MetaWorld Results. As shown in Table 1, both proposed methods outperform prior federated baselines on both MetaWorld scales. GAdapHeadFRL obtains the best result on MT10 (3Set), exceeding the strongest personalized MT-FRL baseline and both centralized MT-RL references. It raises the average success rate of the MT-FRL baselines from 78.1% to 84.9%, an 8.7% relative

improvement. On MT50, GAdapHeadFRL again improves over the strongest federated baseline and is competitive with the centralized MT-RL methods.

The stability advantage is largest on MT50, the largest and most diverse MetaWorld setting. GAdapHeadFRL achieves a standard deviation of 0.6, 82.9% lower than that of the strongest centralized MT-RL reference, at a comparable mean success rate. Together with the gain of GAdapHeadFRL over AdapHeadFRL, these results indicate that adaptive head mixing is effective for continuous-control multi-task FedRL, and that automatic grouping improves the shared priors by reducing interference among less related manipulation tasks.

5.3 Robustness to Task Heterogeneity

Beyond average learning curves, Fig. 4 examines task-level outcome distributions across 48 MiniGrid task instances, including 15 from MT5, 21 from MT7, and 12 from MT12. This view tests whether the gains are robust across heterogeneous task compositions rather than being driven by a small number of favorable tasks.

GAdapHeadFRL improves convergence reliability. It reaches the converged regime on 35/48 task instances (72.9%), compared with 20/48 (41.7%) for FedRL-PPO and 24/48 (50.0%) for PFEDRL-REP, corresponding to 15 and 11 additional converged cases, respectively. At the same time, the number of failed cases drops to 5/48 (10.4%), down from 11/48 (22.9%) and 9/48 (18.8%) for the two baselines. The violin plots further show that GAdapHeadFRL reduces the low-return tail and shifts the distribution toward the high-performance region, with a median peak return of 0.90 versus 0.74 and 0.80. Adaptive grouping and mixing therefore improve task-level reliability under heterogeneous compositions, not just average performance.

5.4 Ablation Study

Dynamic vs. Fixed Mixing Weights. We compare the proposed adaptive α with several fixed mixing coefficients. As shown in Figure 5, adaptive α achieves the highest average return, outperforming all static choices. While a small fixed sharing ratio ($\alpha = 0.2$) is beneficial, larger fixed values lead to noticeable degradation, indicating that excessive reliance on the shared head can hurt performance under task heterogeneity. This result is consistent with our theory: rather than being any fixed value, the optimal mixing strength should vary with the degree of task mismatch and estimation uncertainty during training. Appendix 9 further analyzes how grouping affects the mismatch proxy $\hat{H}$ and the resulting α.

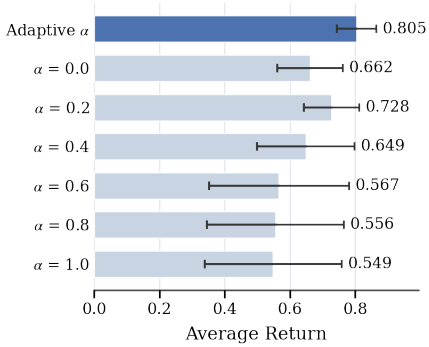


Figure 5: Dynamic vs. fixed mixing weights.

Empirical Coefficient Validation. We compare the practical coefficient with a checkpoint-specific empirical oracle on MiniGrid MT5, freezing the representation, local critic head, group prototype, and Bellman targets and sweeping $\alpha \in [0, 1]$. The oracle minimizes critic loss over this sweep on an independent selection split. Table 2 reports convex profiles in 90–94% of cases and coefficient MAEs of 0.074 and 0.068 at rounds 250 and 500, showing close agreement between the practical coefficient and the checkpoint-specific empirical loss minimizer at both evaluated checkpoints.

Table 2: Practical coefficient vs. empirical oracle on MiniGrid MT5.

Round	Convex (%)	Mean α_{oracle}	Mean $\alpha_{\text{practical}}$	MAE
250	90	0.236	0.219	0.074
500	94	0.287	0.271	0.068

5.5 When Does Selective Sharing Help?

The benefit of selective head sharing should depend on the structure of task heterogeneity. If all tasks were equally related, a single shared prior would suffice and grouping would add nothing. If tasks were entirely unrelated, no amount of sharing would help. The informative regime lies

in between, and we test whether the two components respond to it as the analysis predicts. We therefore construct three same-cardinality MetaWorld MT10 sets with different levels of task-update alignment, which avoids confounding by task count. From a common initialization, each task is trained independently for 20K environment steps without federation or head sharing. Following the update-similarity criterion of Sattler et al. (2021), heterogeneity is measured as the average pairwise cosine dissimilarity between the resulting local updates. The ordering of the three sets is stable across measurements at 5K–20K steps, so the ranking does not depend on the probe length. For each set, we then train PFEDRL-REP, AdapHeadFRL, and GAdapHeadFRL under the same budget and attribute the improvement over PFEDRL-REP to its two sources.

Figure 6 decomposes the improvement over PFEDRL-REP into the gain from adaptive mixing and the additional gain from geometry-based grouping. At low heterogeneity, selective sharing provides a modest total gain of 2.9 percentage points, primarily through geometry-based grouping. At medium heterogeneity, the grouping contribution increases to 6.5 points, showing that partner selection is especially valuable when the tasks form compatible subsets. At high heterogeneity, adaptive mixing becomes the larger contributor, providing 6.0 points compared with 3.6 from grouping. This indicates that when cross-task compatibility is weak overall, client-specific control of the sharing strength becomes increasingly important. Together, these results show that the two components address complementary aspects of heterogeneity: adaptive mixing determines *how much* to share, while directional grouping determines *with whom* to share.

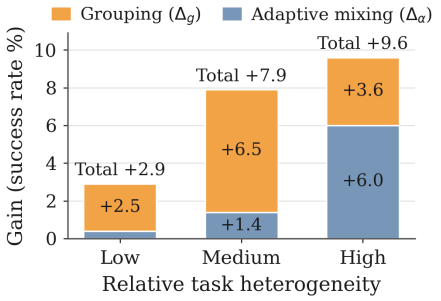


Figure 6: Gain decomposition over PFEDRL-REP on three MetaWorld MT10 sets ordered by update heterogeneity: Δ_α from adaptive mixing, Δ_g additionally from grouping.

5.6 Deployment Robustness

Beyond static full participation, we evaluate new-client assignment, partial participation, and temporarily missing groups. Table 3 summarizes both evaluations.

Table 3: Deployment robustness: new-client assignment on MetaWorld MT10 (left) and partial participation on MiniGrid MT12 (right). RMST is measured in communication rounds.

Integration strategy	Early AUC ↑	Steps to 0.8 ↓	Final return ↑
Local only	0.5050	300k	0.9101
Global prototype	0.6041	275k	0.8761
Refreshed centroid	**0.6272**	**175k**	**0.9968**

Rate (%)	Final performance	Retained performance	RMST to 90% baseline
50	79.0 ± 0.7	91.9%	160
75	84.7 ± 0.5	98.5%	156
100	86.0 ± 0.9	100%	66

New-client assignment. We hold out `drawer-open-v3` and `drawer-close-v3` in MetaWorld MT10. After training, group centroids are recomputed from updated signatures with memberships fixed. Each new client uses a short local probe and joins the most compatible centroid. Local-only adaptation performs well but learns slowly, whereas a single global prototype improves early transfer but lowers final return. Refreshed-centroid assignment performs best both early and finally, reaching 0.8 in 175k steps, 41.7% fewer than local adaptation.

Partial participation and missing groups. Across 10 matched seeds per participation level on MiniGrid MT12, the previous prototype is retained whenever a group is absent. At 75% participation, GAdapHeadFRL retains 98.5% of full-participation performance, with slower convergence, and at 50% it retains 91.9%. Together, refreshed centroids and retained prototypes support these deployment shifts with limited degradation.

6 Conclusion

We studied MT-FRL under structured task heterogeneity and proposed GAdapHeadFRL, which combines geometry-aware grouping with adaptive head mixing. By interpolating between local and shared task heads according to task mismatch and estimation uncertainty, it builds more compatible shared priors while avoiding negative transfer. On MiniGrid and MetaWorld, it consistently improves over federated baselines and remains competitive with strong centralized MT-RL methods, highlighting adaptive intermediate sharing as an effective principle for heterogeneous federated RL.

Two directions follow from this work. The analysis is exact for linear task heads and extends to nonlinear heads only locally, so a global characterization under nonlinear heads remains open. Group assignments are also fixed after warmup, whereas task relationships may drift as the shared backbone evolves, and periodic regrouping would extend the framework to non-stationary deployments.

Acknowledgments

This work was supported by the National Science Foundation, United States under Grants 2047010, 2047064, 2205205, and 2612474. Ni was partially supported by the Department of Transportation, United States under Grant 69A3552348304. Pados' work was supported by AFOSR under grant FA9550-26-1-B225 and AFRL under grant FA8750-25-1-1000 (C-26-263).

References

Arivazhagan, M. G.; Aggarwal, V.; Singh, A. K.; and Choudhary, S. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*.

Bertsekas, D. P.; and Tsitsiklis, J. N. 1996. *Neuro-Dynamic Programming*. Athena Scientific.

Bo, W.; Sun, Y.; Wang, Y.; Zhang, X.; and Li, Z. 2025. FedMGP: Personalized Federated Learning with Multi-Group Text-Visual Prompts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Cao, Q.; Zhu, Z.; Lian, Z.; Zhang, R.; Li, B.; Xiong, Y.; and Zhou, X. 2026. PPFL: A Parameter Behavior-Driven Plug-in Personalization Engine for Federated Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 19898–19906.

Chen, Y.; Geng, Y.; Zhong, F.; Ji, J.; Jiang, J.; Lu, Z.; Dong, H.; and Yang, Y. 2024. Bi-DexHands: Towards Human-Level Bimanual Dexterous Manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5): 2804–2818.

Chevalier-Boisvert, M.; Dai, B.; Towers, M.; de Lazcano, R.; Willems, L.; Lahlou, S.; Pal, S.; Castro, P. S.; and Terry, J. 2023. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. *CoRR*, abs/2306.13831.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting Shared Representations for Personalized Federated Learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2086–2096. PMLR.

D'Eramo, C.; Tateo, D.; Bonarini, A.; Restelli, M.; Peters, J.; et al. 2020. Sharing Knowledge in Multi-Task Deep Reinforcement Learning. In *8th International Conference on Learning Representations,{ICLR} 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 1–11. International Conference on Learning Representations, ICLR.

Duan, M.; Liu, D.; Ji, X.; Liu, R.; Liang, L.; Chen, X.; and Tan, Y. 2020. FedGroup: Efficient Clustered Federated Learning via Decomposed Data-Driven Measure. *arXiv preprint arXiv:2010.06870*.

Farahmand, A.-m. 2011. Regularization in reinforcement learning.

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2020. An Efficient Framework for Clustered Federated Learning. In *Advances in Neural Information Processing Systems*, volume 33.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition.

Hendawy, A.; Peters, J.; and D'Eramo, C. 2024. Multi-Task Reinforcement Learning with Mixture of Orthogonal Experts. In *The Twelfth International Conference on Learning Representations*.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized Cross-Silo Federated Learning on Non-IID Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7865–7873.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31.

Jin, H.; Peng, Y.; Yang, W.; Wang, S.; and Zhang, Z. 2022. Federated Reinforcement Learning with Environment Heterogeneity. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, 18–37. PMLR.

Kwon, D.; Hong, S.; Hong, S.; et al. 2025. Federated Model-Based Offline Multi-Agent Reinforcement Learning for Wireless Networks. In *NeurIPS 2025 Workshop: AI and ML for Next-Generation Wireless Communications and Networking*.

Lee, J.; Xiao, L.; Schoenholz, S. S.; Bahri, Y.; Novak, R.; Sohl-Dickstein, J.; and Pennington, J. 2019. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. In *Advances in Neural Information Processing Systems*, volume 32.

Liu, B.; Liu, X.; Jin, X.; Stone, P.; and Liu, Q. 2021. Conflict-Averse Gradient Descent for Multi-task Learning. In *Advances in Neural Information Processing Systems*, volume 34, 18878–18890.

Lu, S.; Cai, Y.; Liu, Z.; Lian, Y.; Chen, L.; and Wang, H. 2025. A Preference-Based Multi-Agent Federated Reinforcement Learning Algorithm Framework for Trustworthy Interactive Urban Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, 26(7): 10131–10145.

Ma, Y.; Hu, S.; Fang, Z.; Ji, Y.; Deng, Y.; and Fang, Y. 2026. Sense4FL: Vehicular crowdsensing enhanced federated learning for object detection in autonomous driving. *IEEE Transactions on Mobile Computing*.

McLean, R.; Chatzaroulas, E.; Terry, J. K.; Woungang, I.; Farsad, N.; and Castro, P. S. 2025. Multi-Task Reinforcement Learning Enables Parameter Scaling. *Reinforcement Learning Journal*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, 1273–1282. PMLR.

Moazzeni, S.; Huang, Z.; Zeb, S.; Zhang, X.; Parra Ullauri, J.; Bravalheri, A.; Hussain, R.; Wu, Y.; Vasilakos, X.; and Simeonidou, D. 2025. Federated Intelligent Service Function Chain Orchestration in Future 6G Networks. *IEEE Transactions on Network and Service Management*, 22(6): 5690–5704.

Munos, R.; and Szepesvári, C. 2008. Finite-Time Bounds for Fitted Value Iteration. *Journal of Machine Learning Research*, 9: 815–857.

Oh, J.; Kim, S.; and Yun, S.-Y. 2022. FedBABU: Toward Enhanced Representation for Federated Image Classification. In *International Conference on Learning Representations*.

Pang, Y.; Ni, Z.; and Zhong, X. 2025. Integration of a new layer normalization process into federated reinforcement learning for environments with heterogeneous attribute spaces. In Schwartz, P. J.; and Hohil, M. E., eds., *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications VII*, volume 13473, 134730O. International Society for Optics and Photonics, SPIE.

Pang, Y.; Ni, Z.; and Zhong, X. 2026. Decoupling Shared and Personalized Knowledge: A Dual-Branch Federated Learning Framework for Multi-Domain with Non-IID Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 24745–24754.

Qi, J.; Zhou, Q.; Lei, L.; and Zheng, K. 2021. Federated reinforcement learning: techniques, applications, and open challenges. *Intelligence & Robotics*, 1(1): 18–57.

Sattler, F.; Müller, K.-R.; and Samek, W. 2021. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization Under Privacy Constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8): 3710–3722.

Shankar, A.; Mumtaz, S.; Rodrigues, J. J.; Karthikeyan, P.; and Sarveshwaran, V. 2026. Energy-Efficient Task Orchestration in the Edge-Cloud Continuum Using Deep Reinforcement and Federated Learning for Sustainable IOT. *IEEE Transactions on Industrial Informatics*.

Sun, L.; Zhang, H.; Xu, W.; and Tomizuka, M. 2022. Paco: Parameter-compositional multi-task reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 21495–21507.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition.

Teh, Y. W.; Bapst, V.; Czarnecki, W. M.; Quan, J.; Kirkpatrick, J.; Hadsell, R.; Heess, N.; and Pascanu, R. 2017. Distral: Robust Multitask Reinforcement Learning. In *Advances in Neural Information Processing Systems*.

Tsitsiklis, J. N.; and Van Roy, B. 1997. An Analysis of Temporal-Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control*, 42(5): 674–690.

Wang, H.; Xu, W.; Wang, J.; Fan, Y.; Yi, X.; and Zhang, R. 2026. pFedDKS: Detached Knowledge Sharing for Personalized Federated Learning. In *Proceedings of the ACM Web Conference 2026*, 5953–5964.

Xiong, G.; Wang, S.; Jiang, D.; and Li, J. 2025. On the Linear Speedup of Personalized Federated Reinforcement Learning with Shared Representations. In *International Conference on Learning Representations*.

Xiong, W.; Liu, Q.; Li, F.; Wang, B.; and Zhu, F. 2024. Personalized Federated Reinforcement Learning: Balancing Personalization and Experience Sharing via Distance Constraint. *Expert Systems with Applications*, 238: 122290.

Xu, J.; Tong, X.; and Huang, S.-L. 2023. Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867*.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020a. Gradient Surgery for Multi-Task Learning. In *Advances in Neural Information Processing Systems*, volume 33, 5824–5836.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2020b. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, 1094–1100. PMLR.

Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; and Alvarez, J. M. 2021. Personalized Federated Learning with First Order Model Optimization. In *International Conference on Learning Representations*.

Zhang, Y.; Chen, H.; Wang, Z.; Kumar, R.; and Wang, C. 2025. FedVC: Virtual Clients for Federated Autonomous Driving With Imbalanced Label Distribution. *IEEE Transactions on Consumer Electronics*, 71(2): 6253–6262.

7 Proofs of Theoretical Results

7.1 Justification of Assumption 1.

By the Cauchy–Schwarz inequality, $|C_{ig}| \leq \sqrt{V_i V_g}$ always holds. Assumption 1 strengthens this bound only by requiring the inequality to be strict up to a constant $\rho < 1$, thereby excluding the degenerate case where local and global estimation noises are perfectly collinear. This is natural in MT-FRL because the global head aggregates stochastic updates from multiple clients with different environments, trajectories, and mini-batches, so its estimation noise is generally not an exact deterministic copy of any single client's local noise.

7.2 Proof of Lemma 1 (Orthogonal Decomposition)

Orthogonal projection arguments are commonly used to separate approximation and estimation errors in least-squares learning and value-function approximation (Hastie et al., 2009; Munos et al., 2008; Tsitsiklis et al., 1997; Bertsekas et al., 1996). Here, we use this principle to separate the representation-induced error from the task-head error.

Proof. By definition, the local Bellman residual for client i at iteration k, using the mixed projection head $\tilde{\phi}_i(\alpha)$ and a fixed representation θ, is given by the expected squared TD error:

$$\varepsilon_{i,k}(\alpha) = \mathbb{E}_{\nu_i}\left[\left(\langle \varphi_\theta, \tilde{\phi}_{i,k}(\alpha)\rangle - \mathcal{T}_i^\star Q_{i,k}\right)^2\right] \tag{9}$$

To decouple this error, we add and subtract the optimal local projection term $\langle \varphi_\theta, \phi_{i,k}^\star(\theta)\rangle$ inside the squared objective:

$$\varepsilon_{i,k}(\alpha) = \mathbb{E}_{\nu_i}\left[\left(\left\langle \varphi_\theta, \tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\right\rangle + \left\langle \varphi_\theta, \phi_{i,k}^\star(\theta)\right\rangle - \mathcal{T}_i^\star Q_{i,k}\right)^2\right] \tag{10}$$

Expanding the square yields three distinct terms:

$$\begin{aligned}\varepsilon_{i,k}(\alpha) = \mathbb{E}_{\nu_i}&\left[\left\langle \varphi_\theta, \tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\right\rangle^2\right] \\ &+ \mathbb{E}_{\nu_i}\left[\left(\langle \varphi_\theta, \phi_{i,k}^\star(\theta)\rangle - \mathcal{T}_i^\star Q_{i,k}\right)^2\right] \\ &+ 2\mathbb{E}_{\nu_i}\left[\left\langle \varphi_\theta, \tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\right\rangle\left(\langle \varphi_\theta, \phi_{i,k}^\star(\theta)\rangle - \mathcal{T}_i^\star Q_{i,k}\right)\right]\end{aligned} \tag{11}$$

We analyze each term in Equation 11 separately:

Term 1 (Head Estimation Error): We can rewrite the expected squared inner product as a quadratic form using the local feature covariance matrix $\Sigma_i(\theta) = \mathbb{E}_{\nu_i}[\varphi_\theta \varphi_\theta^\top]$:

$$\begin{aligned}\mathbb{E}_{\nu_i}\left[\left((\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta))^\top \varphi_\theta\right)^2\right] &= \mathbb{E}_{\nu_i}\left[(\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta))^\top \varphi_\theta \varphi_\theta^\top (\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta))\right] \\ &= (\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta))^\top \mathbb{E}_{\nu_i}[\varphi_\theta \varphi_\theta^\top](\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)) \\ &= \left\|\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\right\|_{\Sigma_i(\theta)}^2\end{aligned} \tag{12}$$

Term 2 (Representation Error): By definition, this term is the minimum representational error achieved by the optimal head $\phi_{i,k}^\star(\theta)$. This term is independent of the mixing parameter α:

$$\varepsilon_{\mathrm{rep},i,k}(\theta) = \mathbb{E}_{\nu_i}\left[\left(\langle \varphi_\theta, \phi_{i,k}^\star(\theta)\rangle - \mathcal{T}_i^\star Q_{i,k}\right)^2\right] \tag{13}$$

Term 3 (Cross-Term): Recall that the optimal local head $\phi_{i,k}^\star(\theta)$ is defined as the minimizer of the local squared projected Bellman error. Let the objective function be defined as:

$$J(\phi) = \mathbb{E}_{\nu_i}\left[\left(\langle \varphi_\theta, \phi\rangle - \mathcal{T}_i^\star Q_{i,k}\right)^2\right] \tag{14}$$

To find the minimum, we take the gradient of $J(\phi)$ with respect to the vector ϕ by applying the chain rule:

$$\nabla_\phi J(\phi) = 2\,\mathbb{E}_{\nu_i}\left[\varphi_\theta\big(\langle\varphi_\theta, \phi\rangle - \mathcal{T}_i^\star Q_{i,k}\big)\right] \tag{15}$$

Since $\phi_{i,k}^\star(\theta)$ is the optimal solution, the first-order optimality condition requires that the gradient evaluated at $\phi_{i,k}^\star(\theta)$ must be strictly zero. Setting $\nabla_\phi J(\phi_{i,k}^\star(\theta)) = \mathbf{0}$ and dividing by 2 yields the normal equations (which act as the orthogonality principle in this space):

$$\mathbb{E}_{\nu_i}\left[\varphi_\theta\big(\langle\varphi_\theta, \phi_{i,k}^\star(\theta)\rangle - \mathcal{T}_i^\star Q_{i,k}\big)\right] = \mathbf{0} \tag{16}$$

Return to the cross-term, we can rewrite the first inner product as a matrix multiplication: $\langle\varphi_\theta, \tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\rangle = \big(\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\big)^\top \varphi_\theta$. Because the vector difference $\big(\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\big)$ is fixed for a given iteration and deterministic with respect to the expectation over the state-action distribution ν_i, we can factor it out:

$$\begin{aligned}
&\mathbb{E}_{\nu_i}\left[\big(\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\big)^\top \varphi_\theta\big(\langle\varphi_\theta, \phi_{i,k}^\star(\theta)\rangle - \mathcal{T}_i^\star Q_{i,k}\big)\right] \\
&= \big(\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\big)^\top \mathbb{E}_{\nu_i}\left[\varphi_\theta\big(\langle\varphi_\theta, \phi_{i,k}^\star(\theta)\rangle - \mathcal{T}_i^\star Q_{i,k}\big)\right] \\
&= \big(\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\big)^\top \mathbf{0} = 0
\end{aligned} \tag{17}$$

By substituting the normal equations (Equation 16) established above, the expectation exactly matches our zero-gradient condition, collapsing the entire term.

Substituting the simplified Terms 1, 2, and 3 back into Equation 11, we obtain the final exact decomposition:

$$\varepsilon_{i,k}(\alpha) = \big\|\tilde{\phi}_{i,k}(\alpha) - \phi_{i,k}^\star(\theta)\big\|_{\Sigma_i(\theta)}^2 + \varepsilon_{\mathrm{rep},i,k}(\theta). \tag{18}$$

□

7.3 Proof of Theorem 1 (Strict Convexity and Analytic Minimum)

Proof. By definition, the expected head estimation error is given by

$$\mathcal{E}_i(\alpha) := \mathbb{E}\big[\|\tilde{\phi}_i(\alpha) - \phi_{i,k}^\star(\theta)\|_{\Sigma_i(\theta)}^2\big].$$

For notational simplicity, we drop the iteration subscript k on the optimal local head.

Step 1: Bias-Variance Decomposition of the Head Error

We first substitute the stochastic estimates $\hat{\phi}_i = \phi_i^\star + m_i + \xi_i$ and $\hat{\phi}_g = \phi_g^\star + m_g + \xi_g$ into the definition of the adaptively mixed head $\tilde{\phi}_i(\alpha) = (1-\alpha)\hat{\phi}_i + \alpha\hat{\phi}_g$:

$$\begin{aligned}
\tilde{\phi}_i(\alpha) - \phi_i^\star &= (1-\alpha)(\phi_i^\star + m_i + \xi_i) + \alpha(\phi_g^\star + m_g + \xi_g) - \phi_i^\star \\
&= \alpha(\phi_g^\star - \phi_i^\star) + m_i + \alpha(m_g - m_i) + (1-\alpha)\xi_i + \alpha\xi_g
\end{aligned} \tag{19}$$

By substituting the task heterogeneity definition $b_{i,g} = \phi_g^\star - \phi_i^\star$ and grouping the deterministic and stochastic terms, we obtain:

$$\tilde{\phi}_i(\alpha) - \phi_i^\star = \underbrace{m_i + \alpha(b_{i,g} + m_g - m_i)}_{\text{Deterministic Bias } D(\alpha)} + \underbrace{(1-\alpha)\xi_i + \alpha\xi_g}_{\text{Stochastic Noise } S(\alpha)} \tag{20}$$

When we compute the expected squared Σ_i-norm, $\mathcal{E}_i(\alpha) = \mathbb{E}\big[\|D(\alpha) + S(\alpha)\|_{\Sigma_i}^2\big]$, the cross-term is $2\mathbb{E}[\langle D(\alpha), S(\alpha)\rangle_{\Sigma_i}]$. Because the bias $D(\alpha)$ is deterministic and the estimation noise ξ has strictly zero mean ($\mathbb{E}[\xi_i] = \mathbb{E}[\xi_g] = \mathbf{0}$), this cross-term evaluates to exactly zero. Thus, the error exactly decouples into a bias-variance trade-off:

$$\mathcal{E}_i(\alpha) = \|D(\alpha)\|_{\Sigma_i}^2 + \mathbb{E}\big[\|S(\alpha)\|_{\Sigma_i}^2\big] \tag{21}$$

Step 2: Expanding the Deterministic Bias Term

Let $\Delta_m \triangleq m_g + b_{i,g} - m_i$ represent the effective deterministic shift. We can expand the squared Σ_i-norm of the bias $D(\alpha)$:

$$\begin{aligned}
\|D(\alpha)\|_{\Sigma_i}^2 &= \|m_i + \alpha\Delta_m\|_{\Sigma_i}^2 \\
&= (m_i + \alpha\Delta_m)^\top \Sigma_i (m_i + \alpha\Delta_m) \\
&= \|m_i\|_{\Sigma_i}^2 + 2\alpha\langle m_i, \Delta_m\rangle_{\Sigma_i} + \alpha^2\|\Delta_m\|_{\Sigma_i}^2
\end{aligned} \tag{22}$$

Step 3: Expanding the Stochastic Variance Term
Using the linearity of expectation and the definitions $V_i = \mathbb{E}\|\xi_i\|^2_{\Sigma_i}$, $V_g = \mathbb{E}\|\xi_g\|^2_{\Sigma_i}$, and $C_{ig} = \mathbb{E}[\xi_i^\top \Sigma_i \xi_g]$, we expand the variance term $S(\alpha)$:

$$\begin{aligned}\mathbb{E}\big[\|S(\alpha)\|^2_{\Sigma_i}\big] &= \mathbb{E}\left[\big((1-\alpha)\xi_i + \alpha\xi_g\big)^\top \Sigma_i \big((1-\alpha)\xi_i + \alpha\xi_g\big)\right] \\ &= (1-\alpha)^2\mathbb{E}\|\xi_i\|^2_{\Sigma_i} + \alpha^2\mathbb{E}\|\xi_g\|^2_{\Sigma_i} + 2\alpha(1-\alpha)\mathbb{E}[\xi_i^\top\Sigma_i\xi_g] \\ &= (1-2\alpha+\alpha^2)V_i + \alpha^2 V_g + (2\alpha - 2\alpha^2)C_{ig} \\ &= \alpha^2(V_i + V_g - 2C_{ig}) + 2\alpha(C_{ig} - V_i) + V_i\end{aligned} \tag{23}$$

Step 4: Proving Convexity
By combining Equation 22 and Equation 23 back into Equation 21, the total expected error $\mathcal{E}_i(\alpha)$ is a quadratic function of α:

$$\mathcal{E}_i(\alpha) = \alpha^2\Big(\|\Delta_m\|^2_{\Sigma_i} + V_i + V_g - 2C_{ig}\Big) + 2\alpha\Big(\langle m_i, \Delta_m\rangle_{\Sigma_i} - V_i + C_{ig}\Big) + \Big(\|m_i\|^2_{\Sigma_i} + V_i\Big) \tag{24}$$

To determine the convexity of this function, we compute its second derivative with respect to α:

$$\frac{\partial^2 \mathcal{E}_i(\alpha)}{\partial \alpha^2} = 2\Big(\|\Delta_m\|^2_{\Sigma_i} + V_i + V_g - 2C_{ig}\Big) \tag{25}$$

Since Σ_i is a positive semi-definite covariance matrix, $\|\Delta_m\|^2_{\Sigma_i} \geq 0$. For the variance components, Assumption 1 states that there exists a $\rho \in [0, 1)$ such that $|C_{ig}| \leq \rho\sqrt{V_i V_g}$. By the AM-GM inequality, we know $\sqrt{V_i V_g} \leq \frac{1}{2}(V_i + V_g)$. Therefore:

$$2C_{ig} \leq 2|C_{ig}| \leq 2\rho\sqrt{V_i V_g} < 2\sqrt{V_i V_g} \leq V_i + V_g \tag{26}$$

This inequality guarantees that $V_i + V_g - 2C_{ig} > 0$. Consequently, the second derivative is positive ($\frac{\partial^2 \mathcal{E}_i}{\partial \alpha^2} > 0$), proving that $\mathcal{E}_i(\alpha)$ is a convex function of α.

Step 5: Finding the Analytic Minimum
Because $\mathcal{E}_i(\alpha)$ is convex, its unique global minimum occurs where the first derivative is zero. Taking the first derivative of Equation 24 and setting it to zero:

$$\frac{\partial \mathcal{E}_i(\alpha)}{\partial \alpha} = 2\alpha\Big(\|\Delta_m\|^2_{\Sigma_i} + V_i + V_g - 2C_{ig}\Big) + 2\Big(\langle m_i, \Delta_m\rangle_{\Sigma_i} - V_i + C_{ig}\Big) = 0 \tag{27}$$

Solving for α, we obtain the unconstrained optimal mixing coefficient:

$$\alpha_{\text{unc}} = \frac{V_i - C_{ig} - \langle m_i, \Delta_m\rangle_{\Sigma_i}}{\|\Delta_m\|^2_{\Sigma_i} + V_i + V_g - 2C_{ig}} \tag{28}$$

Substituting back $\Delta_m = b_{i,g} + m_g - m_i$ yields the exact expression presented in Theorem 1.

Finally, since the mixing weight in practice must be a valid convex combination $\alpha \in [0, 1]$, and the objective function is a convex parabola, the optimal constrained weight is simply the projection of the unconstrained minimum onto the valid interval: $\alpha_i^\star = \Pi_{[0,1]}(\alpha_{\text{unc}})$. □

8 Extension to Nonlinear Task Heads

Lemma 1 and Theorem 1 are exact for linear task heads under a fixed representation. For a smooth nonlinear head, the same analysis admits a local first-order extension, which we state here to clarify the scope of the main results. Let $Q_{\theta,\phi}(s, a) = h_\phi(\varphi_\theta(s, a))$ with head parameters $\phi \in \mathbb{R}^p$, and fix the representation θ, the sampling distribution ν_i, and the Bellman target within a round. Around a reference head ϕ_0 (for example the local head at the start of the round),

$$Q_{\theta,\phi}(s, a) \approx Q_{\theta,\phi_0}(s, a) + J_{\phi_0}(s, a)\,(\phi - \phi_0), \qquad J_{\phi_0}(s, a) := \left.\frac{\partial Q_{\theta,\phi}(s, a)}{\partial \phi^\top}\right|_{\phi=\phi_0}, \tag{29}$$

where J_{ϕ_0} is the Jacobian of the head output with respect to its parameters. This converts the nonlinear head into a local linear model in ϕ, in the spirit of first-order neural-network analyses (Jacot et al.,

2018; Lee et al., 2019), but applied only within one round and around the current head rather than globally over training.

Under this linearization, the feature covariance $\Sigma_i(\theta) = \mathbb{E}_{\nu_i}[\varphi_\theta \varphi_\theta^\top]$ of the linear analysis is replaced by the Jacobian-induced local metric

$$\Gamma_i := \mathbb{E}_{\nu_i}\left[J_{\phi_0}(s,a)^\top J_{\phi_0}(s,a)\right], \tag{30}$$

so that $\|\hat{\phi}_{g,k} - \hat{\phi}_{i,k}\|^2_{\Gamma_i}$ measures how a difference between the shared and local head parameters translates into an output difference on task i. With Σ_i replaced by Γ_i and the optimal local head replaced by the least-squares minimizer of the linearized model, the normal-equation argument of Lemma 1 goes through unchanged, so the head-error term retains the same quadratic form, and the proof of Theorem 1 applies verbatim to the linearized error since it uses only that the metric is positive semidefinite and that $\tilde{\phi}_{i,k}(\alpha)$ is affine in α. The bias–variance reading therefore remains locally valid: sharing reduces local estimation uncertainty, while excessive sharing introduces bias when the shared and local heads are mismatched.

For a linear head, $J_{\phi_0} = \varphi_\theta^\top$, so $\Gamma_i = \Sigma_i$ and the linearization is exact. The main results are thus exact for linear heads and a local first-order approximation for nonlinear heads. The approximation error is the Taylor remainder of Eq. (29), which under a locally bounded head Hessian is of order $\|\tilde{\phi}_{i,k}(\alpha) - \phi_0\|_2^2$, that is, quadratic in the local–shared displacement along the mixing path. Because the shared backbone also evolves during training, this extension should be interpreted locally at the current representation and head, not as a global guarantee over the optimization trajectory.

8.1 Empirical Evaluation with Multilayer Nonlinear Heads

To complement the local analysis above, we evaluate the sharing principle on Bi-DexHands MT4 (Chen et al., 2024) using PPO with multilayer nonlinear actor–critic heads. Table 4 reports the four task returns and their average. GAdapHeadFRL improves all four tasks over the compared methods and raises the average return from 5.4 for the centralized reference to 6.8, providing direct evidence that mismatch–uncertainty-based sharing remains effective with multilayer nonlinear heads.

Table 4: Returns on Bi-DexHands MT4 with multilayer nonlinear actor–critic heads. Higher is better. Averages are rounded to one decimal place.

Method	`hand_over`	`catch_underarm`	`two_catch_underarm`	`catch_abreast`	Average
Centralized (Chen et al., 2024)	16.5	2.0	0.6	2.5	5.4
FedRL-PPO (Pang et al., 2025)	7.3	1.5	0.6	0.5	2.5
PFEDRL-REP (Xiong et al., 2025)	4.4	1.2	1.0	1.0	1.9
GAdapHeadFRL (Ours)	**17.5**	**4.2**	**2.2**	**3.1**	**6.8**

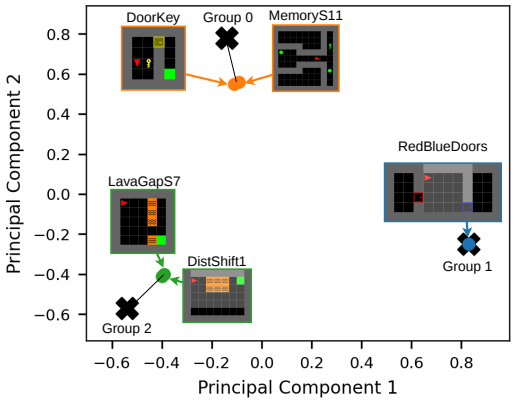
9 Analysis of the Grouping Mechanism

This section complements the main-text ablation with three questions: whether the discovered groups reflect real structure in task updates, whether grouping alone is sufficient without adaptive mixing, and how the choice of update signature affects both clustering quality and performance.

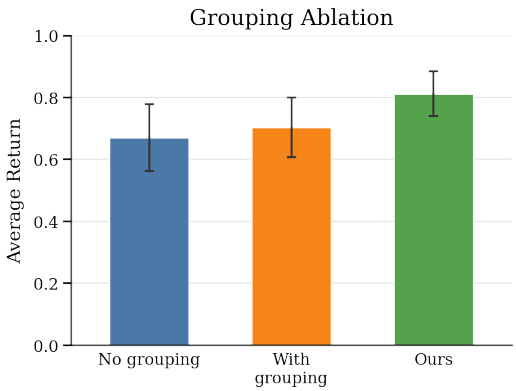
9.1 Is Grouping Alone Sufficient?

Fig. 7(a) projects the task update signatures collected during the warmup stage on MiniGrid MT5 into a two-dimensional PCA space. Tasks assigned to the same group form coherent local clusters even in this low-dimensional view, indicating that the automatic grouping captures genuine similarity in update directions rather than arbitrary partitions.

Using these same assignments, Fig. 7(b) compares FedRL-PPO, FedRL-PPO with grouping, and the full GAdapHeadFRL. Grouping alone improves over the single global head, but the gain is modest and smaller than that of the full method. This is consistent with the heterogeneity-regime study in Section 5.5: the contribution of grouping peaks when tasks form compatible subsets and diminishes as heterogeneity increases, where adaptive mixing becomes the dominant source of improvement.



(a) PCA projection of task update signatures on MiniGrid MT5. Tasks with similar update directions are assigned to the same group.



(b) Ablation on MiniGrid MT5 using the same grouping as in (a). The three bars correspond to FedRL-PPO, FedRL-PPO with grouping, and GAdapHeadFRL.

Figure 7: Additional analysis of the grouping mechanism on MiniGrid MT5. (a) PCA visualization of task update signatures, showing coherent task clusters. (b) Grouping ablation with the same assignments as in (a). Grouping alone changes transfer behavior but does not guarantee optimal sharing, while combining grouping with adaptive head mixing achieves the best performance.

Table 5: Ablation of grouping and signature construction on MetaWorld MT10 (3Set) under adaptive mixing.

Variant	Success rate (%)	Δ
No grouping	80.7	–
Direction + magnitude	80.9	+0.2
Importance-weighted	81.7	+1.0
Raw update	82.8	+2.1
Geometry-based (Ours)	**84.9**	**+4.2**

The reason is that grouping only decides *with whom* a task shares, not *how much*. When cross-task compatibility is weak, even within-group heads retain a non-negligible mismatch with each task, and a grouping-only variant, which applies a fixed transfer intensity, cannot adjust its reliance on the shared head to this residual mismatch.

MetaWorld MT10 shows the same pattern. Relative to the no-grouping baseline, random, manual, and automatic grouping with a fixed α reach 80.1%, 81.2%, and 83.1%, and the full method with adaptive mixing reaches 84.9%. Three observations follow. Even random grouping helps, so restricting sharing to subsets already alleviates cross-task interference of a single global head. Automatic grouping outperforms manual grouping, so the useful sharing structure is better reflected by update-direction geometry than by human task categories. Adaptive mixing adds a further gain on top of the best grouping.

9.2 Grouping and Signature Design

Table 5 compares update-signature choices under adaptive mixing on MetaWorld MT10 (3Set), with improvements measured relative to no grouping. The direction-only signature used by GAdapHeadFRL gives both the best performance (84.9%) and the most reproducible clustering, a stable $K = 4$ partition with ARI/NMI of $0.96/0.97$. Raw updates produce a coarse two-group partition (ARI/NMI $0.81/0.80$) and reach 82.8%, while importance weighting and direction with magnitude yield less reproducible clusterings with varying group counts and drop to 81.7% and 80.9%, respectively.

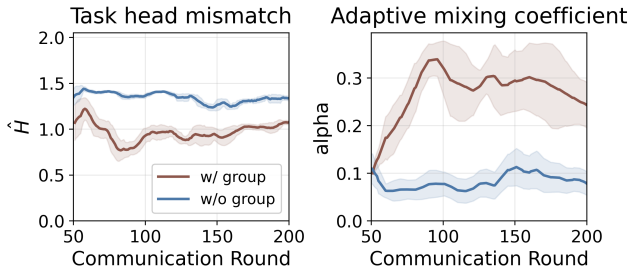


Figure 8: Evolution of the task-head mismatch proxy $\hat{H}$ (left) and the adaptive mixing coefficient α (right) with and without grouping. Grouping lowers the estimated mismatch and correspondingly raises the selected sharing strength.

9.3 Effect of Grouping on Task-Head Mismatch and Adaptive Mixing

Figure 8 tracks the two coupled quantities of the adaptive mixing rule during training. Grouping consistently lowers the estimated mismatch $\hat{H}$ between the local head and the shared prior, and the rule responds by selecting a larger mixing coefficient α. The curves thus make the intended mechanism visible: $\hat{H}$ measures task-prior incompatibility, and α adjusts the degree of transfer accordingly.

10 Environment Details and Proposed Benchmark Variants

10.1 MiniGrid: Original Splits and Our Additional Variants

Following the setup described in the main text, all MiniGrid experiments use the visual observation setting with a $7 \times 7 \times 3$ partially observable input. Our MiniGrid benchmark is built from a 12-task pool (**MT12**) under a unified visual MTRL setting. In addition to retaining the original splits used in Hendawy et al. (2024) as MT5 and MT7, we further introduce two additional sampled variants for each scale, namely **MT5-S1/S2** and **MT7-S1/S2**. Moreover, we also evaluate on the full 12-task benchmark **MT12**, which is used as a broader heterogeneity setting.

In our notation, SimpleCrossing, Pickup, and DynObs correspond to the official MiniGrid environments `Crossing`, `UnlockPickup`, and `DynamicObstacles`, respectively. The detailed task composition of all splits is reported in Table 6.

10.2 MetaWorld: Original Split and Our Additional Variants

Following the setup described in the main text, all MetaWorld experiments are conducted under the same unified multi-task continuous-control setting. Our MetaWorld benchmark is built from the standard 50-task pool (MT50). We retain the original split used in Hendawy et al. (2024) as MT10, and further introduce two additional sampled variants, namely **MT10-S1** and **MT10-S2**.

For reproducibility, the two additional MT10 splits are sampled from the MT50 task pool after removing the original MT10 tasks. Specifically, we follow the canonical MT50 task order, shuffle the remaining 40 tasks using a fixed random seed, and assign the first 10 and the next 10 tasks to MT10-S1 and MT10-S2, respectively. The detailed task composition of all splits is reported in Table 7. Since MetaWorld contains a large number of manipulation tasks, we do not provide per-task descriptions here.

11 Detailed Results on the MetaWorld MT10 Variants

In the main text, we report only the aggregated performance over the three MT10 variants to keep the presentation concise. Table 8 provides the corresponding split-wise results on the original MT10 benchmark and the two additional sampled variants introduced in this work. These detailed results offer a more fine-grained view of how each method behaves under different task compositions with the same task cardinality.

Table 6: Task composition of the MiniGrid benchmarks. MT5 (ori) and MT7 (ori) follow Hendawy et al. (2024). MT5-S1/S2 and MT7-S1/S2 are newly sampled subsets from MT12. Pickup denotes `UnlockPickup`, DynObs denotes `DynamicObstacles`, and SimpleCrossing denotes `Crossing`.

MT \Task	RedBlue Door	DoorKey	DistShift	LavaGap	Memory	Simple Crossing	MultiRoom	Unlock	Empty	Pickup	DynObs	Four Rooms
MT5	✓	✓	✓	✓	✓							
MT5-S1			✓				✓			✓	✓	✓
MT5-S2	✓		✓	✓				✓				✓
MT7	✓	✓	✓	✓	✓	✓	✓					
MT7-S1	✓			✓	✓	✓			✓	✓	✓	
MT7-S2		✓		✓			✓		✓	✓	✓	✓
MT12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 7: Task composition of the MetaWorld benchmarks. MT10 follows Hendawy et al. (2024). MT10-S1 and MT10-S2 are newly sampled 10-task subsets from the MT50 task pool after removing the original MT10 tasks. The sampling uses the canonical MT50 task order.

MT	Task composition
MT10	`reach-v3`, `push-v3`, `pick-place-v3`, `door-open-v3`, `drawer-open-v3`, `drawer-close-v3`, `button-press-topdown-v3`, `peg-insert-side-v3`, `window-open-v3`, `window-close-v3`
MT10-S1	`faucet-close-v3`, `hammer-v3`, `plate-slide-back-v3`, `assembly-v3`, `disassemble-v3`, `handle-pull-side-v3`, `door-lock-v3`, `button-press-topdown-wall-v3`, `coffee-pull-v3`, `coffee-push-v3`
MT10-S2	`door-unlock-v3`, `button-press-v3`, `handle-press-side-v3`, `plate-slide-v3`, `door-close-v3`, `dial-turn-v3`, `pick-place-wall-v3`, `handle-press-v3`, `plate-slide-back-side-v3`, `lever-pull-v3`
MT50	All 50 official MetaWorld tasks.

Overall, the results in Table 8 are consistent with the conclusions drawn in the main text. Our adaptive-head methods remain competitive across all three MT10 sets, and the full method, GAdapHeadFRL, achieves the strongest overall performance when averaged across the variants. This supports our central claim that combining adaptive mixing with grouping yields a more robust federated transfer mechanism under heterogeneous multi-task continuous-control settings.

12 Baseline Architecture Results

This appendix compares our implementations of selected baseline MTRL architectures with the results reported in prior work on MiniGrid and Meta-World, using the training budget of the corresponding original papers. The reproduced numbers are within a few points of the published ones. For the MiniGrid results, the public MOORE numbers are approximated from the performance curves reported in McLean et al. (2025). For the Meta-World results, the public MOORE numbers are drawn from both McLean et al. (2025) and (Hendawy et al., 2024). When two public numbers are shown in the same cell, the first one corresponds to source (1) and the second to source (2). Entries prefixed with $\sim$ denote values that were not explicitly reported in a table and were instead read approximately from the published curves.

13 Implementation Details

This appendix summarizes the implementation details of our federated training pipeline and the additional hyperparameters introduced by adaptive mixing and automatic grouping. We intentionally omit architecture- or method-specific components that are not used in our framework (e.g., mixture-of-experts task encoders), and focus only on the hyperparameters required to reproduce AdapHeadFRL and GAdapHeadFRL.

Common Federated and Adaptive-Mixing Hyperparameters. Table 11 reports the common federated training settings and the method-specific hyperparameters used by AdapHeadFRL and GAdapHeadFRL. The same adaptive-mixing rule is applied to both actor and critic heads. For the warmup length W, we are motivated by Pang et al. (2026) to decouple structure discovery from the early unstable optimization phase, and therefore trigger grouping only after training has entered a relatively stable regime. We set W to approximately 10% of the total communication rounds.

Table 8: Detailed results on the three MT10 sets of MetaWorld. All values are reported as percentages (%) and rounded to one decimal place. The last column reports the average over the three MT10 sets. Entries are shown as mean ± standard deviation when both are available.

Method	Set 0	Set 1	Set 2	MT10 (3Set)
MOORE	84.9 ± 5.6	69.7 ± 4.0	82.9 ± 5.8	79.2 ± 5.2
McLean et al.	86.0 ± 3.5	71.2 ± 7.8	85.5 ± 4.9	80.9 ± 5.4
FedRL-SAC	76.3 ± 1.8	73.5 ± 1.6	84.5 ± 3.6	78.1 ± 7.4
PFEDRL-REP	80.3 ± 1.2	68.2 ± 1.6	85.8 ± 0.3	78.1 ± 9.5
AdapHeadFRL	81.7 ± 3.9	74.2 ± 1.5	86.2 ± 1.0	80.7 ± 2.2
GAdapHeadFRL	$\mathbf{88.2} \pm \mathbf{1.3}$	$\mathbf{77.8} \pm 8.0$	$\mathbf{88.7} \pm 1.5$	$\mathbf{84.9} \pm \mathbf{3.6}$

Table 9: Replication of selected MiniGrid baseline results. All values are success rates (%). Entries prefixed with ∼ are approximate values read from published curves.

Baseline Architecture	MT5 (Ours)	MT5 (Pub)	MT7 (Ours)	MT7 (Pub)
MOORE (Hendawy et al., 2024)	78.4	∼78	79.0	∼77

Empirically, this choice matches the observed learning dynamics: in Fig. 2, the MiniGrid curves already begin to flatten around this stage, indicating that the task update signatures have become more reliable for clustering.

MiniGrid. Following the main experimental setting, all MiniGrid experiments use the visual observation space. To better observe the full training dynamics, we increase the training budget of *all* MiniGrid methods to 500 communication rounds. This change is applied uniformly across baselines and our methods, so the comparison remains fair. We use PPO as the local RL algorithm on MiniGrid. The network architecture consists of a shared representation block and task-specific output heads. The representation block contains three convolutional layers with channels $[16, 32, 64]$, kernel sizes $[(2, 2), (2, 2), (2, 2)]$, and activation functions $[\mathrm{ReLU}, \mathrm{ReLU}, \mathrm{Tanh}]$. The output head contains two linear layers for each task, with hidden/output dimensions $[128, |\mathcal{A}|]$ for the actor and $[128, 1]$ for the critic, using activation functions $[\mathrm{Tanh}, \mathrm{Linear}]$.

MetaWorld. For MetaWorld, we use SAC as the local RL algorithm on MetaWorld. The representation block is implemented as a 3-layer MLP with hidden dimension 400 in each layer and activation functions $[\mathrm{ReLU}, \mathrm{ReLU}, \mathrm{Linear}]$. We match all methods by the *total local environment interaction budget*, rather than by communication rounds alone. Specifically, each client is trained with the same total budget of 20M local environment steps, evenly distributed across tasks within the benchmark. Thus, although our method uses 200 communication rounds, the per-round rollout budget is reduced accordingly so that the total local interaction budget remains unchanged.

For a benchmark with $\mathbb{T}$ tasks, the per-task rollout budget per round is $\frac{20{,}000{,}000}{200 \times \mathbb{T}}$. Therefore, this yields 10,000 local steps per task per round on MT10, and 2,000 local steps per task per round on MT50. Our method therefore does not use more local interaction data than the baselines. It only redistributes the same total budget over more communication rounds.

Communication and computation overhead. We briefly compare the additional overhead of GAdapHeadFRL against the federated baselines. Let the communication cost of synchronizing the full shared model (backbone + task head) in FedRL-PPO/SAC be normalized to 100%. If we denote the backbone size by P_b and the task-head size by P_h, then PFEDRL-REP, which does not synchronize the task head, incurs a normalized communication cost of $P_b/(P_b + P_h)$ per round. In contrast, GAdapHeadFRL behaves like PFEDRL-REP during the warmup phase and like FedRL-PPO/SAC after grouping. Therefore, over T total rounds with a warmup length W, its average normalized communication cost is $\frac{WP_b+(T-W)(P_b+P_h)}{T(P_b+P_h)} = 1 - \frac{W}{T}\frac{P_h}{P_b+P_h}$, which lies strictly between PFEDRL-REP and FedRL-PPO/SAC. Since the synchronized task head is small relative to the full model in our implementation, the absolute communication difference among the three methods is correspondingly small. Table 14 reports the exact cumulative FP32 payload under the evaluated architectures, taking full-model synchronization (FedAvg) as the reference. GAdapHeadFRL transmits 90.90% of the

Table 10: Replication of selected Meta-World baseline results. All values are success rates (%). Entries prefixed with $\sim$ are approximate values read from published curves.

Baseline Architecture	MT10 (Ours)	MT10 (Pub)	MT50 (Ours)	MT50 (Pub)
MOORE (Hendawy et al., 2024)	84.9	$83.2^{(1)}$/$88.7^{(2)}$	72.3	$72.0^{(1)}$/$72.9^{(2)}$
McLean et al. (2025)	86.0	$\sim$86	72.5	$\sim$72

Note. For MOORE, source (1) refers to McLean et al. (2025) and source (2) refers to (Hendawy et al., 2024).

Table 11: Common federated and adaptive-mixing hyperparameters.

Hyperparameter	MiniGrid	MetaWorld
Client fraction	1.0	1.0
Local epochs per round	1	1
Warmup rounds W	50	20
Minimum number of groups $M_{\min}$	2	2
Maximum number of groups $M_{\max}$	4	8
EMA rate for local instability λ_V	0.10	0.10
EMA rate for shared instability λ_G	0.10	0.10
Diagonal geometry regularizer η	10^{-3}	10^{-3}
Numerical stabilizer ϵ	10^{-8}	10^{-8}

FedAvg total on MiniGrid MT5 over 500 rounds and 99.95% on MetaWorld MT10 over 200 rounds, confirming that it never exceeds the communication cost of full synchronization.

In terms of computation, the extra cost of GAdapHeadFRL comes from three lightweight components: warmup-stage signature accumulation, one-time spherical k-means clustering, and per-round prototype maintenance and scalar mixing. These operations are low-dimensional and occur outside the main RL optimization loop, so their cost is not the dominant component compared with rollout collection and local actor–critic training.

Hardware Configuration We report the hardware configuration used for the two benchmark suites for completeness. MiniGrid experiments were conducted on a workstation equipped with an AMD Ryzen Threadripper PRO 3995WX CPU, 1.0 TiB system memory, and 2 NVIDIA GeForce RTX 3080 GPUs. MetaWorld experiments were conducted on a server equipped with dual Intel Xeon Platinum 8480C CPUs, 2.0 TiB system memory, and 8 NVIDIA H200 GPUs.

14 Limitations and Broader Impacts

Our method has several limitations. First, although Theorem 1 provides an analytic characterization of the optimal mixing coefficient, the exact optimum depends on unobservable quantities such as task-specific bias and noise cross-covariance. As a result, the practical implementation relies on a first-order surrogate rather than an exact realization of the theoretical optimum. Second, the proposed grouping mechanism assumes that the warmup-stage update signatures provide a sufficiently stable summary of task relationships. In highly unstable or non-stationary settings, the grouping quality may become sensitive to the warmup dynamics, which could reduce the reliability of the resulting shared prototypes. There is room for exploring better grouping methods.

From a broader-impact perspective, our framework may have positive value in privacy-preserving collaborative learning settings, since it enables knowledge sharing without requiring clients to exchange raw local trajectories. At the same time, like other reinforcement learning methods, it may be misused or deployed in safety-critical settings without sufficient safeguards. Improved federated optimization does not by itself guarantee safe, fair, or socially beneficial deployment.

Table 12: MiniGrid training hyperparameters.

Hyperparameter	Value
Communication rounds	500
Head patterns	linear_layer
Discount factor γ	0.99
Batch size	128

Table 13: MetaWorld training hyperparameters.

Hyperparameter	Value
Communication rounds	200
Total local environment steps per client	20M
Budget allocation across tasks	even split
Per-task rollout steps per round (MT10)	10000
Per-task rollout steps per round (MT50)	2000
Head patterns	linear_layer
Discount factor γ	0.99
Batch size	1280
Replay size	10^6

Table 14: Exact cumulative FP32 communication payload under the evaluated architectures.

Setting	FedAvg total	GAdapHeadFRL total	Relative to FedAvg
MiniGrid MT5, 500 rounds	5.6904 GB	5.1741 GB	90.90%
MetaWorld MT10, 200 rounds	16.4738 GB	16.4657 GB	99.95%

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's scope and main contributions. These claims are supported by the theoretical analysis in Section 3, the method in Section 4, and the experimental and appendix results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limited by length, we discuss the limitations in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

Table 15: Hardware configuration used in our experiments.

Component	MiniGrid	MetaWorld
CPU model	AMD Ryzen Threadripper PRO 3995WX	Intel Xeon Platinum 8480C
CPU configuration	1 socket, 64 cores, 128 threads	2 sockets, 56 cores/socket, 224 threads
System memory	1.0 TiB	2.0 TiB
GPU model	NVIDIA GeForce RTX 3080	NVIDIA H200
Number of GPUs	2	8
GPU memory (per GPU)	10240 MiB	143771 MiB
NVIDIA driver version	570.211.01	550.144.03

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper states the assumptions needed for its main theoretical results in Section 3, most notably Assumption 1 for Theorem 1. The formal statements of Lemma 1 and Theorem 1 are numbered and cross-referenced in the main text, and complete proofs are provided in Appendix 7, while the main paper also includes brief remarks explaining the intuition and role of each result.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose the key elements of the repeatable main conclusions, including baseline construction, task partitioning, evaluation summary, and the overall training process.

The appendix also provides additional results by partition, baseline replication instructions, and the specific values of key method hyperparameters and optimization configurations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code will be submitted with the manuscript and will be open-sourced after the paper is accepted.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experimental table contains the standard deviation region, and the table also includes standard deviation values.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This article does not involve human subjects. We comply with all requirements.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work does not release a high-risk generative model, a scraped Internet-scale dataset, or other artifacts with an obvious dual-use or misuse profile of the kind targeted by this question. The paper focuses on a methodological contribution for multi-task federated reinforcement learning and evaluates it on standard benchmark environments, so this safeguard question is not directly applicable.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We explicitly cited all datasets and code repositories used. Their license requirements are met.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces new benchmark variants rather than new raw datasets or pretrained models, including additional sampled task splits for MiniGrid and MetaWorld. These new assets are documented in the appendix through a description of how the splits are constructed and tables listing their detailed task compositions (Appendix 10).

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments are conducted on standard reinforcement learning benchmark environments, so participant instructions, screenshots, and compensation details are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. Therefore, participant risk disclosure and IRB approval are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as an important, original, or non-standard component. The paper focuses on a multi-task federated reinforcement learning method, its theoretical analysis, and benchmark evaluations, so LLM declaration is not applicable.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.