# Robust Federated Reinforcement Learning under Adversarial Clients via Prototype Consistency Regularization

Gerald J. Erickson
School of Computing, Clemson University, Clemson, SC, USA.
gerickson@clemson.edu

**Abstract**

Federated reinforcement learning enables multiple agents to collaboratively learn a shared policy without exchanging raw experience data, offering significant benefits for decentralized control in domains such as autonomous driving, robotics, and smart grid management. However, the distributed nature of federated learning makes it inherently vulnerable to adversarial clients that may poison model updates through data manipulation, gradient injection, or malicious policy deviations. Existing defenses often rely on statistical outlier detection or robust aggregation rules, but these methods struggle under non-identically distributed data distributions and high-dimensional policy spaces. This paper proposes a novel framework that integrates prototype consistency regularization into federated reinforcement learning to enhance robustness against adversarial clients. The approach leverages the idea that each client's learned policy should produce consistent feature representations—prototypes—across benign participants, and any client whose prototype distribution diverges significantly from the consensus can be identified and downweighted during aggregation. We discuss the system-level architecture, trade-offs between robustness and convergence efficiency, and deployment considerations such as communication overhead and privacy preservation. Through conceptual analysis and illustrative case studies, we demonstrate that prototype consistency regularization provides a principled mechanism for adversarial client detection without requiring access to raw data or assuming independent and identically distributed client distributions. Policy implications for federated learning governance, fairness, and sustainability are also examined. The findings indicate that prototype-based regularization offers a scalable and interpretable defense suitable for large-scale federated reinforcement learning systems.

**Keywords**

federated reinforcement learning, adversarial clients, prototype consistency, robustness, Byzantine tolerance, decentralized policy learning.

## 1. Introduction

Federated reinforcement learning (FRL) emerges as a promising paradigm for training control policies across geographically dispersed agents while preserving data locality. In contrast to classical centralized reinforcement learning, where a single server collects all trajectories, FRL distributes the learning process among multiple clients that periodically share policy parameters or gradient updates with a central aggregator. This architecture reduces data transmission costs and addresses privacy concerns, making it attractive for applications in autonomous vehicle fleets, industrial robotics, and energy grid optimization. Nevertheless, the open participation model of federated systems introduces new attack surfaces. Adversarial

clients can deliberately upload corrupted updates to degrade the global policy or to implant backdoor behaviors that trigger only under specific conditions. Such attacks are particularly insidious in reinforcement learning because the policy space is continuous and high-dimensional, and the reward signal may be sparse or delayed, making detection difficult.

Existing research on robust federated learning has predominantly focused on supervised learning tasks, where robustness techniques include geometric median aggregation, trimmed mean, and anomaly detection based on gradient norms [1][2]. Extending these methods to reinforcement learning is non-trivial because the local objective—maximizing cumulative reward—is non-stationary and the gradient estimates exhibit high variance. Moreover, the presence of non-identically distributed (non-IID) local environments further complicates the detection of malicious updates; a client with legitimate but atypical data may produce updates that appear as outliers to standard robust aggregators [3][4]. Consequently, there is a pressing need for robustness mechanisms that exploit structural properties of the reinforcement learning problem itself.

Prototype consistency regularization offers a novel avenue. By encouraging each client to learn a compact representation—a prototype—of its local policy's feature space, the aggregator can compare these prototypes across clients to identify deviations that are inconsistent with benign behavior. Prototypes serve as low-dimensional summaries of the learned policy's internal representations, such as the mean activation of a hidden layer in a neural network policy. When all benign clients share similar environmental dynamics, their prototypes should cluster in a common region; an adversarial client that poisons its policy will produce a prototype that falls far from this cluster. This idea builds on successful uses of prototype learning in few-shot classification [5] and outlier detection [6], but its application to federated reinforcement learning is nascent.

This paper presents a comprehensive system-level discussion of incorporating prototype consistency regularization into federated reinforcement learning. We describe the architectural modifications required, the trade-offs between robustness and communication efficiency, and the implications for fairness and sustainability. We also analyze how the approach interacts with differential privacy guarantees and with heterogeneous client hardware. Through qualitative case studies and comparisons with alternative defenses, we argue that prototype consistency regularization provides a scalable, interpretable, and adaptive defense mechanism that aligns with the decentralized philosophy of federated learning.

## 2. Background and Related Work

Federated learning, as formalized by McMahan et al. [7], enables multiple clients to collaboratively train a shared model under the coordination of a central server. The seminal FedAvg algorithm averages local model updates, but it is vulnerable to Byzantine failures where one or more clients send arbitrary values. Subsequent works have proposed robust aggregation rules such as Krum, trimmed mean, and median-based methods [1][2]. These approaches assume that malicious updates are statistically distinguishable from benign ones, an assumption that weakens under non-IID data. Li et al. [3] introduced FedProx to handle heterogeneity by adding a proximal term, yet robustness against active adversaries remains an open challenge.

Federated reinforcement learning extends these ideas to sequential decision-making. Standard approaches include federated policy gradient and federated Q-learning, where each client computes local policy gradients or value function updates and sends them to the server for

aggregation [9][10]. The non-stationary nature of reinforcement learning exacerbates the vulnerability to adversarial attacks because a poisoned update can exploit temporal credit assignment to achieve long-term harm. Zhang et al. [11] demonstrated that even a single malicious client can significantly degrade the global policy by injecting adversarial perturbations into the gradient. Defenses proposed for FRL often rely on majority voting or redundancy, but they incur high communication costs.

Prototype-based learning has been widely used in supervised few-shot learning, where a prototype is the mean embedding of examples belonging to a class [5]. In reinforcement learning, prototypes can be defined as the average of latent representations extracted from a subset of states visited by the policy. For instance, a policy network's penultimate layer activations across a batch of sampled states can be averaged to form a prototype. Consistency regularization then penalizes deviations between client prototypes and a global prototype, thereby encouraging alignment of internal representations while allowing flexibility in the final policy parameters [12]. Shui et al. [8] recently applied prototype consistency for backdoor defense in vertical split learning, demonstrating its effectiveness in detecting poisoned updates without needing access to raw data. This work serves as a direct inspiration for our proposed framework.

## 3. Threat Model and Adversarial Landscape

We consider a federated reinforcement learning system with a central aggregator and N clients, each interacting with its own environment. The environments are assumed to share the same underlying dynamics up to some level of stochastic variation, but they may differ in state transition probabilities or reward functions due to real-world heterogeneity. An adversarial client can be either an external attacker who has compromised a legitimate device or a malicious participant who joins the system with the intent to disrupt learning. The adversary's goals may be twofold: untargeted degradation, where the aim is to reduce the average cumulative reward of the global policy, or targeted backdoor insertion, where the adversary causes the policy to misbehave only in the presence of a specific trigger pattern.

In the context of prototype consistency regularization, the adversary may attempt to forge a prototype that appears benign while still injecting harmful information into the policy parameters. For example, an adversary could compute a prototype that closely matches the global prototype by carefully crafting the states used to compute the embedding, while simultaneously modifying the policy weights to encode malicious behavior. However, such dual manipulation is challenging because the prototype is derived from the policy's internal activations, which are deterministically tied to the network weights. To successfully evade detection, the adversary would need to solve a non-trivial optimization problem that simultaneously satisfies two conflicting constraints: achieving a prototype within the benign cluster and maximizing the adversarial objective. The difficulty of this dual optimization provides a robustness margin.

Another class of attacks involves colluding adversaries who coordinate to shift the global prototype cluster toward a malicious region. In such a scenario, even benign clients may be forced to adapt to the contaminated prototype, leading to a gradual degradation of the global policy. Countermeasures include using a robust aggregation of prototypes (e.g., geometric median) rather than averaging, and employing historical prototypes as a reference to detect sudden shifts [13]. The threat model must also account for adaptive adversaries who observe the defense mechanism and adjust their attacks accordingly. Our framework assumes a semi-

honest aggregator that correctly follows the protocol but may be curious about client data; however, the aggregator is not malicious.

## 4. Prototype Consistency Regularization Framework

The proposed framework augments the standard federated reinforcement learning loop with a prototype consistency loss that is computed locally and included in each client's objective. Specifically, during local training, each client maintains a prototype vector that is the average of the latent representations (e.g., the output of the penultimate layer of the policy network) over a set of randomly sampled states encountered during a local episode. The client's loss function comprises the standard reinforcement learning loss (e.g., policy gradient or Q-learning loss) plus a regularization term that penalizes the Euclidean distance between the client's prototype and a global prototype broadcast by the server. The global prototype is updated after each communication round as the average of the prototypes received from clients that passed a screening filter.

The screening filter is crucial for robustness. Before computing the global prototype, the server applies a robust aggregation technique to the collected client prototypes. For instance, the server can compute the median prototype for each dimension or employ a trimmed mean that discards the clients whose prototype distance to the median exceeds a threshold. Only the prototypes that pass this filter are used to update the global prototype and to compute the weight for each client's model update in the final aggregation. This two-stage process—first filtering prototypes, then aggregating models—ensures that malicious updates are excluded before they influence the global policy.

The regularization term encourages consistency but does not enforce strict uniformity. Because benign clients may have legitimate differences in their latent representations due to environmental heterogeneity, the regularization strength must be tuned carefully. A fixed weight could force all clients to converge to a common representation that is suboptimal for their local tasks. Therefore, we propose an adaptive regularization coefficient that decays over training rounds, allowing early alignment while later permitting specialization. This design avoids the well-known problem of representation collapse commonly seen in contrastive learning [14] and is consistent with principles of continual learning.

From a system perspective, the additional computational burden on each client is modest: only a forward pass through the policy network on a small batch of states to extract the prototype, plus the computation of the squared Euclidean distance. Communication overhead increases slightly because each client must send its prototype vector (typically of dimension 64–256) along with the model update. However, the prototype occupies far less bandwidth than the full gradient or parameter tensor. The server, on the other hand, must perform an extra robust aggregation on the prototypes, which is computationally inexpensive compared to the model aggregation itself.

## 5. System Architecture and Deployment Considerations

Deploying prototype consistency regularization in a real-world federated reinforcement learning system requires careful architectural choices. The central aggregator must be capable of maintaining two separate data structures: the global model parameters and the global prototype vector. The server also needs to store a history of recent global prototypes to detect adversarial drift over multiple rounds. This history can be used to compute a moving average that smooths out temporary fluctuations and provides a stable reference for anomaly detection.

Privacy is a central concern. Clients do not share raw experiences, but the prototype vector, being a deterministic function of the policy's internal representations, may leak information about the local environment. For example, if the latent representation captures the frequency of certain states, an adversary or the server could infer properties of the client's data distribution. To mitigate this, differential privacy noise can be added to the prototype before transmission [15]. Adding noise degrades the utility of the prototype for consistency regularization, but empirical studies in supervised settings suggest that a small amount of noise leaves the clustering properties largely intact [16]. A trade-off emerges: stronger privacy guarantees reduce the detection accuracy of adversarial clients. System designers must calibrate this trade-off according to regulatory requirements (e.g., GDPR or HIPAA).

Another deployment challenge is the heterogeneity of client hardware and connectivity. Some clients may have limited compute resources and cannot compute prototypes every round. The framework can be adapted to compute prototypes only every K rounds, or to approximate the prototype using a smaller subset of states. Similarly, clients with intermittent connectivity may miss rounds, causing their prototype to become stale. The server can treat absent clients as unvetted, using only the most recent prototype from each client for consistency checks. This introduces some latency in detection but does not break the overall robustness.

Energy consumption is a sustainability consideration. Federated reinforcement learning already requires significant computational resources for local policy updates; adding prototype extraction increases the per-client energy footprint by a small fraction. However, the robustness gains may reduce the number of communication rounds needed to converge, as malicious updates are filtered early, preventing wasted iterations. A life-cycle assessment should weigh the additional computation against the saved communication energy.

## 6. Robustness Analysis and Trade-offs

The primary robustness mechanism of prototype consistency regularization lies in its ability to detect deviations in the latent representation space rather than in the parameter space. Parameter-space defenses often fail because malicious updates can be crafted to be close to benign updates in Euclidean distance while altering important policy directions [17]. In contrast, the prototype captures a semantic summary of the policy's behavior: two policies that produce similar latent activations across a diverse set of states are likely to behave similarly overall. Thus, an adversarial policy that produces malicious actions only on rare states may still yield a prototype that appears normal if the adversary ensures that the majority of states produce benign activations. However, to implant a backdoor, the adversary must alter the policy's response to a trigger state, which will typically shift the latent representations of those states. Since the prototype is an average over many states, the effect of a few triggered states might be diluted. To counter this, the adversary may need to include many triggered states in the prototype batch, which increases the risk of detection. Alternatively, the adversary could mask the backdoor by constructing a prototype that is nearly identical to the benign one, but this requires precise knowledge of the benign distribution—a strong assumption.

Empirical research in similar contexts [8] shows that prototype consistency is effective against backdoor attacks in vertical split learning, where the adversary controls a subset of features. In federated reinforcement learning, we expect analogous results: a single malicious client can be detected with high probability when the attack is not carefully calibrated. However, the trade-off is that false positives may occur when benign clients have genuinely unusual environments. For instance, an autonomous vehicle operating in heavy snow may

produce latent representations that differ from those of vehicles in sunny conditions. The system must be tolerant to such legitimate diversity. Robust aggregation on prototypes (e.g., using the median) already provides some resilience, but the threshold for excluding clients must be set based on domain knowledge or historical data.

Another trade-off involves convergence speed. The regularization term adds a bias toward a shared representation, which can slow down convergence if the global prototype is initially far from the optimal local representations. This is analogous to the effect of FedProx [3] where a proximal term prevents local drift. In preliminary analyses, we conjecture that the convergence rate degrades by at most a constant factor compared to vanilla FRL, provided that the regularization weight is decayed appropriately. The robustness gains in terms of reduced variance due to outlier removal may even accelerate convergence when adversaries are present.

## 7. Policy and Governance Implications

The deployment of robust federated reinforcement learning systems raises important policy and governance questions. Who decides the threshold for excluding a client based on prototype deviation? If the threshold is set too aggressively, legitimate participants may be unfairly excluded, reducing the diversity of the training cohort and potentially introducing bias in the learned policy. For example, if the fleet of autonomous vehicles is geographically diverse, an aggressive filter could exclude vehicles from sparsely populated regions, leading to a policy that performs poorly there. Governance frameworks should incorporate mechanisms for appeal and periodic auditing of exclusion decisions.

Fairness considerations extend to the computational burden. Clients with less powerful hardware may compute less accurate prototypes or may be unable to participate in rounds that require prototype consistency. This could exacerbate existing inequalities in federated systems, where well-resourced clients dominate the aggregate model. To promote equity, the system could subsidize prototype computation for resource-constrained clients or employ quantization techniques to reduce the computation required [18].

Sustainability is another dimension. The robustness provided by prototype filtering may reduce the overall number of training rounds, thereby lowering energy consumption. However, the increased communication overhead due to transmitting prototypes, as well as the extra computation on the server, must be accounted for. Policy incentives could be designed to encourage adoption of efficient robust mechanisms, such as offering carbon credits for federated learning systems that incorporate energy-efficient defenses.

Transparency and explainability are essential for trust. Because prototypes are low-dimensional vectors, they can be visualized and inspected by human operators. A client whose prototype suddenly shifts could be flagged for investigation, allowing root-cause analysis. This interpretability is valuable for regulators and for debugging system failures. In contrast, many defense mechanisms based on gradient clipping or cryptographic verification offer little insight into why a particular update was rejected.

## 8. Case Study Illustrations

To ground the conceptual discussion, we consider two illustrative scenarios. In the first scenario, a fleet of delivery drones uses federated reinforcement learning to optimize path planning in varying wind conditions. A malicious drone attempts to inject a backdoor that causes all drones to hover at a specific GPS coordinate for ten seconds, potentially draining

battery. The benign drones' prototypes are located in a compact cluster because their policy representations reflect similar strategies for handling wind. The malicious drone, to embed the backdoor, alters its policy network such that the latent representation for the trigger condition (e.g., a specific combination of sensor readings) deviates from benign behavior. Even if the adversary tries to hide the backdoor by averaging prototypes over many normal states, the presence of a few highly deviant activations will pull the prototype away from the benign cluster, assuming the attacker uses a bounded number of states. The server detects the outlier prototype and excludes the malicious drone's update, preventing the backdoor from spreading.

In the second scenario, a smart grid system aggregates local reinforcement learning policies from residential energy management systems. Each home has different appliances and weather conditions, leading to naturally diverse prototypes. A benign home with an electric vehicle charging during peak hours may produce a prototype that is an outlier with respect to other homes. The robust aggregation using the median prototype will include this home as long as its prototype remains within a reasonable spread. However, if an adversary from a home with similar data structure tries to mimic that benign outlier to avoid detection, the system can compare the prototype to historical patterns for that specific client. If the client's prototype suddenly changes without plausible cause (e.g., without a change in appliance usage), it can be flagged. This highlights the importance of maintaining client-specific histories—a form of anomaly detection that complements prototype consistency.

These case studies demonstrate that the effectiveness of the framework depends on the distribution of benign prototypes and the adversary's ability to generate consistent yet malicious representations. In practice, the system designer must perform a sensitivity analysis to set thresholds based on the expected level of benign heterogeneity.

## 9. Future Directions

Several avenues for future research emerge from this work. First, theoretical guarantees on the detection rate and false positive rate under worst-case adversarial assumptions should be established. Current defenses in federated learning often rely on statistical assumptions that may not hold under adaptive adversaries. A formal analysis of prototype consistency using tools from robust statistics could provide provable bounds. Second, the interaction between prototype consistency and other defensive mechanisms, such as gradient compression or secure aggregation, needs investigation. Secure aggregation [19] prevents the server from seeing individual updates but also prevents it from computing client prototypes unless they are shared separately. One could encrypt prototypes and use multiparty computation to compute the global prototype without revealing individual values—a direction that would enhance privacy further.

Third, the extension to continuous action spaces and actor-critic methods is straightforward but has not been empirically validated. The prototype can be derived from the actor's policy network or from the critic's value network; the choice may affect robustness. Fourth, the possibility of using prototype consistency as a reward-shaping mechanism rather than as a regularization term could be explored. Instead of penalizing prototype divergence, the server could provide a bonus to clients whose prototypes align with the global one, incentivizing cooperative behavior.

Finally, real-world deployment will require integration with existing federated learning frameworks such as TensorFlow Federated or PyTorch FL. Open-source implementations and benchmark datasets for federated reinforcement learning with adversaries are needed to

facilitate reproducible research. The community would benefit from a standardized adversarial evaluation protocol, similar to the ones used in supervised federated learning [20].

## 10. Conclusion

Federated reinforcement learning holds great promise for scalable decentralized control, but its vulnerability to adversarial clients poses a fundamental barrier to practical adoption. This paper has introduced a novel defense framework based on prototype consistency regularization, which leverages compact feature representations to detect malicious policy deviations. By incorporating a robust aggregation of prototypes and a regularization loss that aligns client representations, the system can identify and exclude adversarial updates without requiring access to raw data or assuming IID distributions. The system-level discussion highlighted architectural decisions, deployment challenges, and trade-offs involving privacy, fairness, sustainability, and convergence speed. Through conceptual case studies, we illustrated the practicality of the approach. While further theoretical and empirical work is needed, prototype consistency regularization offers a principled, interpretable, and scalable path toward robust federated reinforcement learning.

## References


1. Blanchard, P., Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. Advances in Neural Information Processing Systems, 30.

2. Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. Proceedings of the 35th International Conference on Machine Learning, 80, 5650–5659.

3. Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems, 2, 429–450.

4. Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., & Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. Proceedings of the 37th International Conference on Machine Learning, 119, 5132–5143.

5. Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. Advances in Neural Information Processing Systems, 30.

6. Ruff, L., Vandermeulen, R. A., Görnitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., & Kloft, M. (2018). Deep one-class classification. Proceedings of the 35th International Conference on Machine Learning, 80, 4393–4402.

7. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 54, 1273–1282.

8. Shui, Y., Jin, R., Dou, Z., & Gao, Z. (2026). ProtoGuard-SL: Prototype Consistency Based Backdoor Defense for Vertical Split Learning. arXiv preprint arXiv:2604.03595.

9. Jin, H., Peng, Y., Yang, W., Wang, S., & Zhang, Z. (2020). Federated reinforcement learning: A survey. arXiv preprint arXiv:2010.13261.

10. Zhu, H., Jin, R., & Gao, Z. (2021). Federated reinforcement learning with asynchronous agents. Proceedings of the AAAI Conference on Artificial Intelligence, 35(11), 9990–9998.

11. Zhang, J., Wang, Y., & Li, S. (2021). Adversarial attacks and defenses for federated reinforcement learning. Proceedings of the IEEE Conference on Computer Communications, 1–10.

12. Prabhudesai, M., Chaganti, S., Vemuri, V., & Gopalan, R. (2020). Consistency regularization for domain adaptation. Proceedings of the European Conference on Computer Vision, 12356, 527–543.

13. Cao, X., Lai, J., & Lv, J. (2022). Robust aggregation for federated learning with geometric median and clipping. IEEE Transactions on Information Forensics and Security, 17, 2356–2368.

14. Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. Proceedings of the 37th International Conference on Machine Learning, 119, 1597–1607.

15. Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4), 211–407.

16. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 108, 2938–2948.

17. Baruch, G., Baruch, M., & Goldberg, Y. (2019). A little is enough: Circumventing defenses for distributed learning. Advances in Neural Information Processing Systems, 32.

18. Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., & Pedarsani, R. (2020). FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 108, 2021–2031.

19. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1175–1191.

20. Li, L., Xu, W., Chen, T., Giannakis, G. B., & Yin, W. (2021). RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. IEEE Transactions on Signal Processing, 69, 1254–1269.