COORDINATING COEXISTING LEARNING AGENTS IN SHARED SPECTRUM VIA PARAMETER SPACE COMPLEMENTARITY

Md Ashikul Haque[1] **Abusayeed Saifullah**[1] **Haibo Zhang**[2]
[1]Department of Computer Science, University of Texas at Dallas
[2]School of Computer Science and Engineering, University of New South Wales

ABSTRACT

Independently managed Low Power Wide Area Networks (LPWANs) such as LoRa and SigFox increasingly share crowded unlicensed spectrum. Recent systems push adaptation into the LoRa Network Server (LNS) and use reinforcement learning to tune transmission parameters such as channel, spreading factor, and power. When multiple deployments operate learning agents at their LNS and overlap in time and frequency, independent agents exposed to similar conditions can drift toward the same resource choices, creating persistent collisions and non stationary dynamics that slow or destabilize learning. We propose a lightweight coordination layer that exchanges only model level information. Each LNS periodically shares policy parameters and an optional compact performance summary, and a Collaborative Coordination Service synthesizes individualized updates that encourage complementary operating regions rather than global consensus. We describe a hypernetwork style coordination mechanism and a composite objective that balances local utility preservation with cross network diversity. Experiments on a small physical setup and representative NS-3 simulations show that parameter space coordination can improve reliability and reduce transmissions per delivered packet relative to uncoordinated deep RL agents, while requiring no raw traffic sharing and only periodic backhaul side updates. We also discuss practical safety, privacy, and robustness considerations for deploying such coordination infrastructure in open environments.

1 INTRODUCTION

Low Power Wide Area Network (LPWAN) technologies are a key enabler of the Internet of Things, providing long range connectivity for battery powered devices across applications such as smart cities, environmental monitoring, industrial automation, and smart agriculture (Fahmida & et al., 2020; 2022). The rapid growth of IoT has accelerated deployments of LoRa, SigFox, and NB IoT, many of which operate in congested unlicensed ISM bands (Saifullah & et al., 2018). In dense urban and industrial environments, independently managed deployments overlap in time and frequency, and coexistence becomes increasingly difficult: cross network interference degrades packet reception rate (PRR), increases latency, and raises energy per delivered packet (Mekkia & et al., 2018). Prior studies highlight the severity of this problem, reporting large throughput drops with only a few coexisting LoRa networks (Voigt & et al., 2017) and collision rates that approach saturation in high density settings (Krupka & et al., 2016; Georgiou & et al., 2017).

Agents in the wild view. When learning based controllers are deployed at the network server, each LNS behaves as an autonomous agent that senses partial feedback and acts through a shared environment. The wireless medium couples agents through interference over long horizons, and emergent synchronization can yield undesirable outcomes such as persistent collisions and unstable oscillations.

Why coexistence is hard in LPWANs. LPWAN links are long range and often operate under severe hidden terminal conditions. LoRaWAN Class A uplinks are effectively ALOHA like, and collisions are particularly costly because airtime can be long at high spreading factors. Duty cycle regulations and downlink scarcity further limit the use of reactive coordination, and end devices are constrained in compute, memory, and energy. These factors make it difficult to apply classical collision avoidance or fine grained scheduling across independently managed deployments.

Existing mitigation options and their limits. On the PHY side, a line of work improves collision resolution (for example by decoding overlaps or exploiting structure) (Xu et al.; Xia & et al., 2020; Shahid & et al.; Wang & et al.; Tong & et al., 2020). These approaches can improve robustness, but they may require gateway modifications, tighter synchronization assumptions, or expensive processing, and they do not eliminate repeated conflicts when many independent deployments continually select similar operating points. On the MAC and control side, static or heuristic policies (for example fixed channel plans or spreading factor assignments) can be brittle under dynamic traffic and heterogeneous link budgets.

Reinforcement learning (RL) offers a practical path to adaptation by tuning transmission parameters in response to changing conditions. In LPWANs, sustained on device training is often infeasible, so recent systems shift learning to the LoRa Network Server (LNS), which can run deep RL and apply network wide control without increasing end device burden (Haque et al., 2025). This model provides a strong foundation for optimizing a single adaptive deployment.

A new failure mode: coexisting learning agents. Our focus is an emergent multi network failure mode. When multiple learning agents overlap, their independently trained policies can converge toward similar "best" channels and spreading factor mixes, creating repeated collisions and a moving target for each agent. The environment becomes non stationary because other agents also adapt. This can prolong convergence, induce oscillations, and reduce the gains that learning was intended to provide. In shared unlicensed spectrum, locally rational learning can be globally harmful and can also degrade performance for nearby legacy devices that cannot adapt.

Our approach. We introduce a coordination layer that is periodic rather than continuous, model based rather than trace based, and diversity seeking rather than consensus seeking. Each network retains autonomy and continues its local learning loop, while the coordination service provides a slow moving signal that nudges policies away from persistent conflicts. The key idea is complementarity: when networks face similar local conditions, coordination encourages their learning agents to occupy distinct regions of the resource space (for example, different channel preferences or spreading factor mixtures), reducing repeated collisions without requiring strict orthogonal partitioning.

Contributions.

- We identify and formalize the *coexisting learning agents* failure mode for LNS centric RL in LPWANs, and we articulate why naive extensions of single network learning and static coexistence mechanisms are insufficient.
- We introduce a parameter space coordination setting in which networks exchange only model parameters and optional compact KPIs with a Collaborative Coordination Service (CCS), avoiding raw traffic sharing and high overhead coupling.
- We present a generative complementarity mechanism that synthesizes distinct policy updates for each network, optimizing a composite objective that balances local utility preservation with cross network diversity.
- We provide evidence from a small physical setup and representative NS-3 simulations (ns3; Magrin & et al.) that coordinated updates can improve reliability and reduce transmissions per delivered packet relative to uncoordinated deep RL agents, and we discuss practical safety, privacy, and robustness considerations.

2 Related work and limitations of existing approaches

This section reviews prior work and uses it to motivate the need for a coordination layer that is both lightweight and explicitly multi network.

2.1 Coexistence mechanisms in LPWANs

Traditional coexistence approaches for WiFi or Bluetooth rely on CSMA or TDMA style coordination (Yang & et al., 2010; 2011). These approaches transfer poorly to LPWANs due to long range hidden terminals, long airtime, strict duty cycle constraints, and the need to keep end device logic simple. For LoRa in particular, several works improve collision resolution at the PHY layer (Xu et al.; Xia & et al., 2020; Shahid & et al.; Wang & et al.; Tong & et al., 2020), for example by decoding overlapping transmissions or exploiting signal structure. At the MAC layer, protocol designs such as Burst-MAC target bursty traffic within a single LoRa deployment (Jain et al., 2024). Related anti-jamming systems improve robustness to malicious interference and collaborative jamming (Haque & Saifullah, 2023; 2024; 2025; 2026).

Limitations. PHY collision resolution improves robustness but has deployment friction. Many techniques require gateway modifications, tighter synchronization assumptions, or expensive processing. In addition, collision resolution does not address repeated conflicts that arise when many independent deployments continuously select similar operating points, especially under adaptive control. Further, when interference dynamics become non stationary due to other adaptive deployments, improvements in decoding alone may not stabilize network wide behavior.

2.2 Learning based control for LPWANs

RL has been used for wireless resource management across cognitive radio, routing, and general scheduling (Yau & et al., 2012; Zhang & et al.). In LPWANs, Q learning has been explored for parameter selection (Yu & et al., 2020), and some approaches place learning on end devices (Fahmida & et al., 2023). More recent work shifts learning to the LNS and uses deep RL to adapt a single network under interference (Haque et al., 2025).

Limitations. Most learning based approaches optimize an individual network in the presence of exogenous interference. When the interference source is itself adaptive, the learning dynamics change. With multiple concurrently learning LNS entities acting as learning agents, independent optimization can create non stationary dynamics that are not addressed by single agent RL methods. In practice, this can appear as repeated policy oscillations, extended exploration phases, or convergence to unstable local equilibria that are sensitive to minor traffic changes.

2.3 Coordination in shared spectrum

There is extensive work on spectrum etiquette and coexistence in unlicensed bands, including listen before talk, channel hopping, and fairness mechanisms. However, LPWAN constraints make many of these mechanisms difficult to apply at fine time scales: downlinks are scarce, uplinks are sporadic, and end devices are intentionally simple. In addition, independently managed deployments have limited incentives to reveal internal telemetry or accept an externally imposed schedule. These constraints motivate periodic coordination that can be implemented at the control plane (backhaul side) rather than at the radio plane. Orthogonal work on cross-technology communication shows that lightweight signaling between heterogeneous LPWAN technologies is feasible (Haque et al., 2026), which could complement but does not replace model-based coordination among learning agents.

2.4 Why not solve this with standard MARL or full centralization

Multi agent RL explicitly studies multiple learners, but directly applying standard MARL to coexisting LPWAN deployments is challenging. It often assumes centralized training with shared experience, dense communication among agents, or a shared reward with careful credit assignment. These assumptions conflict with independently managed networks, privacy constraints, and the desire to avoid sharing raw traffic traces or detailed channel measurements.

At the other extreme, a fully centralized scheduler that assigns resources across networks would require real time ingestion of fine grained telemetry and would raise privacy and scalability concerns. Simple static partitioning of orthogonal resources can be brittle under dynamic traffic and heterogeneous coverage, and it can become inefficient when resource demand is unbalanced.

Takeaway. Existing solutions either do not address the coexisting learning agents failure mode, or they rely on coordination assumptions that are hard to satisfy in practice. This motivates periodic parameter level coordination that preserves autonomy and seeks diversity only when it reduces persistent conflicts.

3 Problem definition and coordination setting

We consider K coexisting LPWAN deployments in a shared region. Each deployment is managed by an LNS_k that controls a set of end devices and runs a local RL agent. In this paper, we refer to the LNS side RL controller as the deployment's learning agent.

3.1 Control plane and decision variables

In LoRaWAN style LPWANs, many impactful decisions are taken at the network server and then applied as network wide control. Typical controllable knobs include channel selection and frequency plans, spreading factor and data rate selection, transmit power, and (where supported) retransmission or repetition policies. These knobs interact with airtime, capture effects, and collision probability. In congested regions, small shifts in channel preference mixtures can have outsized effects on contention and can change the effective reward landscape experienced by learning agents.

We focus on LNS level control because it (i) avoids on device training, (ii) can aggregate feedback across many end devices, and (iii) can implement consistent policies across a deployment. Importantly, LNS level control also defines a natural boundary for cooperation between independently managed deployments: coordination can occur between servers without requiring changes to end devices or gateway RF chains.

Local learning model. Each LNS_k maintains a policy π_k parameterized by θ_k. The policy maps a local state s to an action a. The state can include recent acknowledgment outcomes, observed PRR trends, queue information, and coarse interference indicators available to the LNS. The action can include channel selection, spreading factor selection, power level, and other controllable parameters. The reward signal is derived from a local utility function that reflects reliability, energy, and latency objectives.

The coexisting learning agents failure mode. When LNS learning agents optimize independently, they often experience similar state distributions. If the reward function values similar operating points, the agents can drift toward similar actions. This increases collision probability and creates a feedback loop: collisions change reward and observed state, which causes further adaptation. From the perspective of each learner, the environment becomes non stationary because the transition dynamics depend on the other networks' evolving policies. A key challenge is that this non stationarity is structured: it is driven by policy similarity across deployments, not only by stochastic channel effects.

Coordination constraints. We aim to address this failure mode under practical constraints.

- **Limited sharing.** Networks should not be required to share raw packet traces, detailed device metadata, or gateway IQ samples.
- **Low overhead.** Coordination should be feasible over commodity backhaul, with updates on the order of minutes rather than per packet.
- **Independence.** Each network retains autonomy and continues its own learning and control loop.
- **Heterogeneity.** Different networks may use different RL agent architectures or different local reward designs.

4 Approach overview: parameter space coordination

We explore a coordination mechanism that exchanges only model level information.

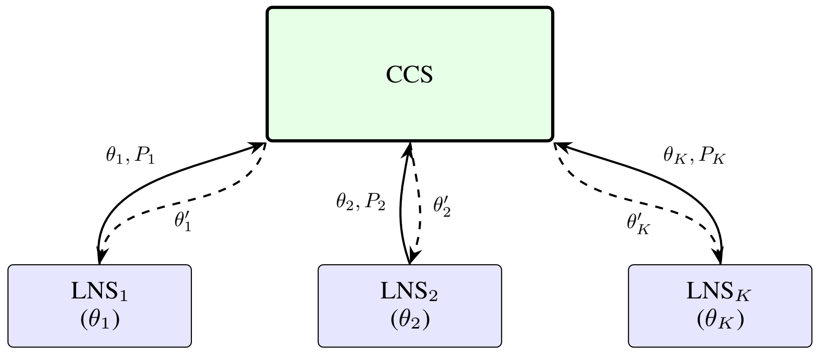

Figure 1: System overview: Each LNS$_k$ periodically shares its policy parameters θ_k along with a compact KPI report P_k (e.g., PRR, EPN, ATP), and the CCS synthesizes coordinated parameters θ'_k that encourage complementary (non identical) resource preferences to reduce repeated conflicts in shared spectrum.

Collaborative Coordination Service (CCS). Each LNS periodically sends a compact snapshot of its current policy parameters and optionally a small scalar performance summary to a CCS. The CCS then produces individualized parameter updates for each network and returns them to the corresponding LNS.

Design goal: complementarity rather than consensus. Instead of aggregating all networks into one shared policy, the CCS encourages different networks to prefer different operating regions when their local states are similar. This provides a coarse coordination signal that can reduce repeated conflicts while preserving privacy and avoiding high overhead sharing.

This parameter centric exchange resembles federated style communication in that raw traffic remains local, but it differs from federated aggregation because the objective is to synthesize a set of coordinated yet distinct policies rather than a single global model.

Coordination cadence. Coordination rounds occur periodically, for example every few minutes. Between rounds, each LNS continues its local control loop and can still adapt at its own pace. This separation of time scales is important: the CCS provides a slow moving coordination signal that nudges policies away from repeated conflicts, while local learning reacts to faster dynamics. A practical benefit of this design is that the CCS interface can be implemented as a simple backhaul API, leaving the radio protocol unchanged.

Worked example. Consider two coexisting deployments whose independent learning agents both identify the same channel and spreading factor pair as locally optimal during early exploration. If both converge to that operating point, collisions persist and both networks experience degraded PRR, higher ATP, and a noisier reward signal. In our setting, each LNS reports only its current parameters and a compact KPI summary. The CCS can respond by nudging one policy toward a neighboring operating region, for example by shifting the channel preference or adjusting the spreading factor mix, while keeping both policies close to their locally learned configurations. The goal is not to enforce strict orthogonality, but to break symmetry so that similarly situated learning agents do not repeatedly choose the same resources.

5 Generative complementarity mechanism

The CCS uses a generative mapping G to transform an input set of potentially conflicting policies $\{\theta_k\}_{k=1}^{K}$ into a coordinated set $\{\theta'_k\}_{k=1}^{K}$. Alongside policy parameters, each LNS can send a compact KPI summary over the last reporting window,

$$P_k \triangleq \big[\mathrm{PRR}_k,\ \mathrm{EPN}_k,\ \mathrm{ATP}_k,\ \bar{R}_k\big], \tag{1}$$

where PRR_k is packet reception rate, EPN_k is average energy per delivered packet, ATP_k is average attempts per delivered packet, and $\bar{R}_k$ is the average RL return or a scalar utility proxy.

Algorithm 1 One coordination round (high level)

for $k = 1$ to K in parallel **do**
  LNS$_k$ exports θ_k and computes P_k.
  LNS$_k$ $\rightarrow$ CCS: send (θ_k, P_k).
end for
CCS forms contexts $\{C_k\}$ and samples latent codes $\{z_k\}$.
for $k = 1$ to K **do**
  CCS computes $\theta'_k = \theta_k + G_k(\theta_k, C_k, z_k)$.
  CCS $\rightarrow$ LNS$_k$: send θ'_k.
end for
Each LNS applies θ'_k and continues local control until the next round.

5.1 HYPERNETWORK STYLE INDIVIDUALIZED UPDATES

In our instantiation, G is a hypernetwork that produces a residual update around each local policy,

$$\theta'_k = \theta_k + \Delta\theta_k, \qquad \Delta\theta_k = G_k\big(\theta_k,\ C_k,\ z_k\big), \tag{2}$$

where z_k is a policy specific latent code that prevents deterministic collapse and C_k is a coordination context that summarizes the other networks.

A simple permutation invariant context uses pooled policy embeddings,

$$\bar{\theta}_{-k} \triangleq \frac{1}{K-1}\sum_{j\neq k}\tilde{\theta}_j, \qquad C_k \triangleq \mathrm{MLP}_{\mathrm{ctx}}\big([\tilde{\theta}_k \parallel \bar{\theta}_{-k}]\big), \tag{3}$$

where $\tilde{\theta}_k$ is an embedding of θ_k produced by an encoder for the local agent type. This embedding step is where heterogeneity is handled: different RL agent architectures can be mapped into a shared representation space.

5.2 COORDINATION OBJECTIVE

The CCS aims to preserve each network's local utility while explicitly encouraging diversity in behavior across networks. We express this as a composite objective,

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{task}} + \beta\mathcal{L}_{\text{div}} + \gamma\mathcal{L}_{\text{adv}}, \tag{4}$$

where $\mathcal{L}_{\text{task}}$ penalizes degraded network utility (either via return or via a KPI weighted proxy derived from P_k), and $\mathcal{L}_{\text{div}}$ encourages separation.

For discrete action spaces, one practical choice is to maximize pairwise divergence between action distributions,

$$\mathcal{L}_{\text{div}} \triangleq -\frac{2}{K(K-1)}\sum_{1\leq i<j\leq K}\mathbb{E}_{s\sim\mathcal{D}}\Big[\mathrm{JS}\big(\pi'_i(\cdot|s)\parallel\pi'_j(\cdot|s)\big)\Big], \tag{5}$$

which discourages overlapping resource preferences under similar states. The optional term $\mathcal{L}_{\text{adv}}$ reinforces functional distinguishability through an auxiliary discriminator that predicts which network produced a given state action pair.

5.3 COORDINATION ROUND PROTOCOL

A typical coordination cycle is lightweight.

5.4 TRAINING AND DEPLOYMENT

In our prototype, G is trained offline using simulated rollouts that provide diverse policy snapshots, and then deployed as a fixed inference model during online operation. Each LNS continues its own local learning and exploration, while the CCS provides periodic coordinated updates. A deeper treatment of training stability, dataset construction, and sensitivity to coordination interval and objective weights is deferred to future work.

6 Practical considerations

Overhead and update frequency. Policy parameter snapshots are modest compared to typical gateway backhaul. For the deep RL agents we use, snapshots are on the order of hundreds of kilobytes. With coordination intervals on the order of minutes, the resulting control overhead is well below commodity backhaul capacity.

Safety, privacy, and robustness. Exchanging only parameters reduces exposure compared to sharing raw traffic traces, but parameters and KPIs can still leak information in some settings. Coordination also introduces an attack surface: a faulty or strategic participant can send outlier parameters or manipulated KPIs that steer updates toward harmful regions. Because coordination is periodic, a practical safeguard is conservative update application at each LNS, for example clipping the residual $\Delta\theta_k$ and rejecting updates that degrade a short horizon local KPI proxy. Robust coordination should also account for staleness and partial participation by reducing update magnitude when snapshots are old, and deployments may require fairness safeguards so that diversity pressure does not push smaller networks into inefficient extremes.

Heterogeneity. Different networks may adopt different RL algorithms over time. The encoder based design allows the CCS to map different parameter structures into a shared embedding space, enabling coordinated updates across heterogeneous learners without requiring all networks to standardize on one RL stack.

Incentives and fairness. A coordination layer is only useful if networks choose to participate. In shared unlicensed spectrum, participation can be framed as mutually beneficial: complementarity reduces repeated conflicts and improves reliability for everyone. Practical deployments may therefore require fairness constraints or adaptive diversity weights that depend on congestion and relative traffic load.

Asynchrony and partial participation. Perfect synchronization is not required. In practice, each LNS can report on its own schedule and the CCS can operate on the most recent snapshot from each participant, generating updates for all participants or for the subset that reported within a window. Non participating deployments are treated as external interference, and participating networks can still benefit if the coordination signal reduces repeated conflicts among the learning enabled subset.

Compatibility with LoRaWAN operational constraints. LoRaWAN uplinks are often the dominant traffic source, while downlinks are scarce and constrained by duty cycle. A practical coordination mechanism should therefore primarily influence uplink decisions, such as channel and spreading factor distributions, and should avoid requiring per packet coordination or frequent downlink control. The periodic parameter update model aligns with these constraints, and it can be implemented as a backhaul side service rather than a radio side protocol change.

7 Evaluation

We evaluate parameter space coordination using a small indoor physical setup and representative NS-3 simulations (ns3; Magrin & et al.). We focus on coexistence regimes where independent learning agents tend to concentrate on similar operating points and create repeated collisions.

7.1 Experimental setup and baselines

Physical deployment model. Our physical setup instantiates multiple coexisting LoRa networks in the same indoor environment. Each network comprises a gateway and a set of end devices managed by an LNS instance. Networks are configured to overlap in time and frequency, and traffic is generated to emulate common periodic IoT reporting patterns alongside bursty episodes (for example due to alarms). To isolate coexistence effects, each LNS observes only its own acknowledgments and network side statistics, and it does not receive privileged information about other deployments beyond what is provided through the coordination interface.

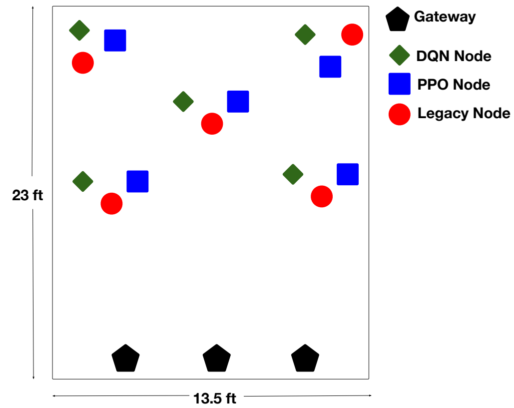

Figure 2: Physical experiment layout (illustrative). Multiple coexisting LoRa networks operate in the same indoor environment while LNS learning agents adapt and optionally coordinate through the CCS.

Learning agents and action space. Each LNS runs a deep RL agent that selects transmission configurations from a restricted action set that reflects practical LPWAN operations. Actions include channel selection and a small set of data rate or spreading factor options, and may include transmit power choices when permitted by the regional rules. The reward trades off delivery success and airtime or energy cost proxies, so that policies are incentivized to reduce retransmissions while maintaining reliability.

Coordination interval. Coordination occurs periodically. Between coordination rounds, each LNS continues local learning and uses the current policy to make per packet decisions. This isolates the effect of coordination updates and makes it clear whether periodic parameter nudges are sufficient to avoid persistent policy collapse.

Simulation setup. Simulations vary the number of networks, managed nodes, and external interferers within a multi kilometer region. The objective is to stress test the coexisting learning agents failure mode under controlled scaling and traffic dynamics. We vary traffic intensity across networks to emulate heterogeneous deployments (for example a large industrial deployment coexisting with smaller building scale systems) and introduce time varying external interference to test stability.

Metrics. We report packet reception rate (PRR), energy per delivered packet (EPN), and attempts per delivered packet (ATP). ATP captures retransmission pressure and directly reflects airtime contention. EPN summarizes the combined cost of higher transmission counts and higher transmit power choices. When evaluating mixed settings with legacy nodes, we also report spillover impact on the legacy group.

Baselines. We compare against uncoordinated independent deep RL at each LNS (Haque et al., 2025). We also include a simple centralized coordinator that statically partitions orthogonal resources in a greedy manner to illustrate the limits of rigid assignment under traffic and interference dynamics. This baseline is informative because it shows the gap between strict partitioning and adaptive complementarity, especially when traffic load is imbalanced.

7.2 Representative results

Summary table. Table 1 summarizes representative improvements for coordinated updates relative to independent deep RL in several coexistence regimes.

Dynamic interference slice. Figure 3 shows one simulation slice under dynamic external interference, where complementarity yields higher PRR and fewer transmissions per delivered packet.

Heterogeneous environment. We also consider a mixed architecture environment where one network uses a DQN style agent, one uses a policy gradient style agent, and one group behaves

Scenario	PRR gain	EPN gain	ATP gain
2 coexisting learning networks	$\geq 20\%$	$\geq 10\%$	$\geq 15\%$
4 coexisting learning networks	$\geq 10\%$	$\geq 8\%$	$\geq 10\%$
Heterogeneous learners plus legacy	$\geq 5\%$	$\geq 5\%$	$\geq 5\%$

Table 1: Representative improvements from parameter space coordination relative to independent learning.

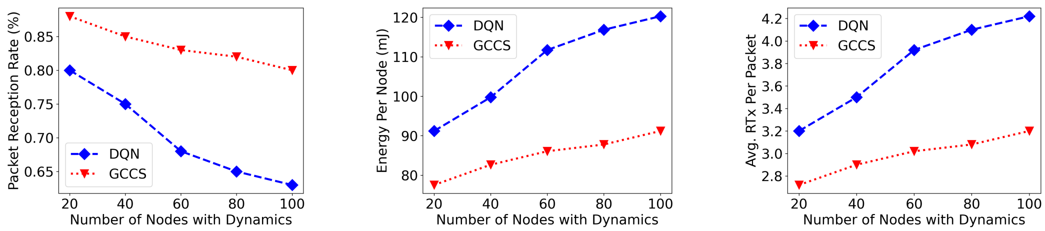

Figure 3: Simulation slice under dynamic external interference. Coordinated policies maintain higher PRR while reducing energy per delivered packet and transmission attempts per delivered packet.

as legacy LoRaWAN nodes. Figure 4 illustrates that coordination can improve reliability in this heterogeneous setting while also reducing interference spillover to the legacy group.

7.3 Qualitative observations

Symmetry breaking and sustained separation. Across scenarios where independent learners repeatedly collide, coordinated updates push policies into distinct regions of the resource space and keep them separated for long windows. Importantly, this is not strict orthogonality. Rather, each network develops a distinct preference mixture, for example different channel probabilities combined with different spreading factor tradeoffs, which reduces simultaneous collisions while still allowing local learning to adapt within its own region. This behavior is valuable in practice because it tolerates shifts in demand: if one network becomes temporarily quiet, others can expand without requiring a rigid long term partition.

Role of latent codes. The latent codes $\{z_k\}$ act as a low cost source of stochasticity that helps avoid deterministic collapse when two networks report nearly identical parameter vectors. In qualitative ablations where the latent path is removed, the hypernetwork can sometimes produce similar updates for multiple participants, especially when the KPI summaries are also similar. With the latent path retained, updates tend to be more individualized even when the input snapshots are highly symmetric.

When coordination can hurt. Complementarity is not always beneficial. If the channel is already lightly loaded, enforcing diversity can introduce unnecessary exploration and can temporarily reduce local utility. Likewise, if one network has a much larger traffic volume than the others, naive diversity may shift smaller networks into inefficient configurations to accommodate the larger network. These observations motivate fairness constraints and adaptive diversity weights that depend on congestion and relative traffic load, as well as safeguards that reduce the strength of diversity when collision indicators are low.

Compute footprint. A coordination round consists of encoding parameter snapshots, pooling contexts, and producing residual updates. For the model sizes we consider, inference is lightweight compared to local training and can be amortized across participants. Because updates occur on the order of minutes, a cloud hosted CCS can support many concurrent networks, with engineering attention to staleness, asynchronous arrivals, and admission control under bursts.

8 Discussion, limitations, and open questions

Stability under asynchronous coordination. Real deployments may not synchronize coordination rounds, and policies may arrive at the CCS with different delays. Understanding stability and

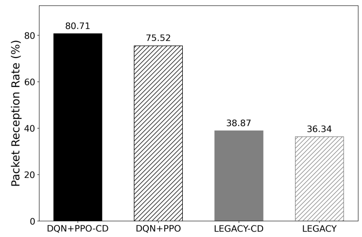

Figure 4: Mixed learner setting. Coordinated updates improve PRR relative to uncoordinated learning in a heterogeneous environment.

convergence in asynchronous and partially participating environments remains open. A promising direction is to couple coordination strength to an estimate of staleness or uncertainty, so that older snapshots lead to smaller residual updates.

Strategic behavior and incentives. Networks may have incentives to misreport KPIs or to avoid coordination if it appears locally costly. Incentive compatible coordination and mechanisms that discourage free riding are important for practical adoption. In unlicensed bands, participation may be most natural in environments where multiple networks share long term infrastructure or where coordinated operation yields measurable mutual gains.

Robustness to adversaries. A malicious participant could attempt to poison the coordination process by sending adversarial parameters or by inducing collapse toward a harmful configuration. Defenses could include anomaly detection on parameter updates, robustness regularizers, and trust scoring. Because coordination is periodic, one can also add conservative acceptance tests at each LNS, for example applying an update only if it improves short horizon KPIs.

Privacy leakage. Even without raw traces, model parameters can encode information about traffic patterns or local conditions. A careful privacy analysis is required, along with possible mitigations such as parameter compression, differentially private noise, or secure aggregation variants. An important practical question is the threat model: who operates the CCS and what guarantees participants require.

Choosing diversity. Diversity is not an end in itself. The diversity objective must be aligned with coexistence metrics and fairness, and it must avoid pushing networks toward extreme configurations that harm local utility. Understanding how to choose and tune β in (4) across heterogeneous networks is a central open question, and it likely depends on congestion regimes, traffic imbalance, and regulatory constraints.

Broader agentic takeaway. When multiple deployed learning agents interact through a shared environment, independent optimization can produce emergent synchronization that is locally stable but globally harmful. Periodic parameter space coordination is a lightweight way to break symmetry under limited sharing constraints.

9 Conclusion

Model parameter coordination offers a pragmatic path toward more stable coexistence among learning enabled LPWAN deployments. By synthesizing complementary policy updates rather than aggregating toward consensus, a coordination service can reduce repeated conflicts while keeping data local. Our results indicate that parameter space coordination can improve reliability and reduce transmissions per delivered packet, motivating deeper evaluation and theoretical characterization in future work.

REFERENCES

Ns-3. https://www.nsnam.org/.

Sezana Fahmida and et al. Long-lived lora: Prolonging the lifetime of a lora network. In *2020 ICNP*, pp. 1–12. IEEE, 2020.

Sezana Fahmida and et al. Real-time communication over lora networks. In *2022 IoTDI*, pp. 14–27. IEEE, 2022.

Sezana Fahmida and et al. Handling coexistence of lora with other networks through embedded reinforcement learning. In *IoTDI*, pp. 410–423, 2023. ISBN 9798400700378.

O. Georgiou and et al. Low power wide area network analysis: Can lora scale? *WCL*, 2017.

Md Ashikul Haque and Abusayeed Saifullah. A game-theoretic approach for mitigating jamming attacks in lpwan. In *EWSN*, pp. 274–284, 2023.

Md Ashikul Haque and Abusayeed Saifullah. Handling jamming attacks in a lora network. In *2024 IEEE/ACM Ninth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 146–157. IEEE, 2024.

Md Ashikul Haque and Abusayeed Saifullah. Mitigating jamming attacks in lora networks: A defense strategy against lora-based jammers. In *MobiHoc*, pp. 51–60, 2025.

Md Ashikul Haque and Abusayeed Saifullah. Decoding lora packets under collaborative jamming attacks. *EWSN*, pp. 1–12, 2026.

Md Ashikul Haque, Abusayeed Saifullah, and Haibo Zhang. Deep reinforcement learning based coexistence management in lpwan. In *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2025.

Md Ashikul Haque, Aakriti Jain, Venkata Prashant Modekurthy, and Abusayeed Saifullah. Enabling cross technology communication from lr-fhss to lora. *SenSys*, pp. 1–13, 2026.

Aakriti Jain, Md Ashikul Haque, Abusayeed Saifullah, and Haibo Zhang. Burst-mac: A mac protocol for handling burst traffic in lora network. In *2024 IEEE Real-Time Systems Symposium (RTSS)*, pp. 148–160, 2024. doi: 10.1109/RTSS62706.2024.00022.

L. Krupka and et al. The issue of lpwan technology coexistence in IoT environment. In *ME*, pp. 1–8, 2016.

Davide Magrin and et al. Performance evaluation of lora networks in a smart city scenario. In *ICC (2017)*.

Kais Mekkia and et al. A comparative study of lpwan technologies for large-scale iot deployment. In *ICT Express*, 2018. doi: https://doi.org/10.1016/j.icte.2017.12.005.

Abusayeed Saifullah and et al. Low-power wide-area networks over white spaces. *ToN*, 26(4): 1893–1906, 2018. doi: 10.1109/TNET.2018.2856197.

Muhammad Osama Shahid and et al. Concurrent interference cancellation: Decoding multi-packet collisions in lora. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*.

Shuai Tong and et al. Colora: Enabling multi-packet reception in lora. In *INFOCOM'20*, 2020.

Thiemo Voigt and et al. Mitigating inter-network interference in lora networks. EWSN '17, pp. 323–328, 2017.

Xiong Wang and et al. mlora: A multi-packet reception protocol in lora networks. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*.

Xianjin Xia and et al. Ftrack: Parallel decoding for lora transmissions. *IEEE/ACM Transactions on Networking*, 2020.

Zhenqiang Xu, Pengjin Xie, and Jiliang Wang. Pyramid: Real-time lora collision decoding with peak tracking. In *INFOCOM'21*.

D. Yang and et al. Coexistence of ieee802.15.4 based networks: A survey. In *IECON*, 2010.

Dong Yang and et al. Wireless coexistence between ieee 802.11- and ieee 802.15.4-based networks: A survey. *IJDSN*, 7(1):912152, 2011.

Kok-Lim Alvin Yau and et al. Review: Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues. *J. Netw. Comput. Appl.*, 35(1): 253–267, January 2012. ISSN 1084-8045.

Yi Yu and et al. Multi-agent q-learning algorithm for dynamic power and rate allocation in lora networks. In *2020 PIMRC*, 2020.

Hailu Zhang and et al. Reinforcement learning-based interference control for ultra-dense small cells. In *GLOBECOM'18*.