CENTROID-PRESERVING DYNAMICS: COMPOSABLE ASYNCHRONOUS COORDINATION FOR MULTI-LEARNER REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

While multi-learner parallelism scales training throughput in deep reinforcement learning (DRL), asynchronous coordination often fails to translate raw throughput into policy improvements. We show that coordination mechanisms can shift the population parameter centroid and induce structural drift, destabilizing the coupled parameter–policy–data feedback loop. To address this, we model multi-learner training as a driven–dissipative dynamical system. Through an orthogonal decomposition into parameter centroid and consensus disagreement, we identify two essential operator-theoretic criteria for stable coordination: *centroid preservation* and *disagreement non-expansion*. We prove both properties are closed under finite asynchronous compositions, establishing asymptotic disagreement bounds under window contraction and closed-loop feedback. Guided by these principles, we develop a centralized $1/N$-normalized incremental commit mechanism and a decentralized symmetric zero-sum pairwise protocol, accompanied by an in-flight mass model capturing physical delays. Across the TianJi benchmark and four Atari environments, scaling to three nodes demonstrates that our decentralized protocol achieves a $1.66\times$–$2.05\times$ wall-clock speedup over synchronous baselines with higher tail returns and AUC. Furthermore, it cuts time-to-threshold by 10.9%–46.1% over non-centroid-preserving baselines and maintains robust stability under dynamic network latencies, showing that principled coordination reliably converts barrier-free throughput into end-to-end efficiency.

1 INTRODUCTION

Distributed deep reinforcement learning (DRL) has transitioned from single-learner multi-actor systems Babaeizadeh et al. (2017); Horgan et al. (2018) to multi-learner clusters Huang et al. (2024); He et al. (2025). In multi-learner architectures, the parameter coordination protocol dictates whether increased throughput yields genuine wall-clock speedups. Synchronous coordination relies on `AllReduce` primitives or barriers that are vulnerable to stragglers and network jitter Dutta et al. (2018); Chen et al. (2016). Conversely, asynchronous schemes eliminate lockstep barriers, yet parameter staleness and elevated update variance often degrade sample efficiency, forcing contemporary frameworks to retain version locks, importance sampling, or periodic barriers Wijmans et al. (2020); Huang et al. (2024); He et al. (2025); Sun et al. (2023).

Prior work often conflates update staleness with coordination mechanics under monolithic asynchronous models, obscuring structural biases induced by the coordination operators themselves. DRL features a closed "parameter–policy–data" loop: parameter updates shift the behavior policy and state visitation distribution, which directly govern subsequent gradient evaluations Schulman et al. (2015); Kakade & Langford (2002); Sutton & Barto (2018). Uncontrolled coordination biases compound along this loop, degrading asymptotic performance and inflating sample complexity.

To formalize these dynamics, we model multi-learner training as a driven–dissipative dynamical system in the parameter space. We project the stacked parameter vector x onto two orthogonal subspaces: the centroid subspace $\mathcal{S}_{\parallel}$ (replica average) and the disagreement subspace $\mathcal{S}_{\perp}$ (replica deviations). Local optimization drives centroid motion while injecting disagreement; conversely, coordination must dissipate disagreement without inducing spurious centroid drift. We focus on

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

these dissipative dynamics, modeling closed-loop policy–data feedback via an affine perturbation bound within a locally convex region supporting stable single-learner updates. This decomposition shifts the focus from version staleness to the balance between disagreement injection and dissipation.

Our analysis reveals that structural bias arises when coordination operators leak disagreement into the centroid subspace. Leveraging consensus optimization tools Olfati-Saber et al. (2007); Lian et al. (2017); Koloskova et al. (2020), we establish stability criteria and prove that positive semidefinite, doubly stochastic coordination atoms are closed under arbitrary asynchronous composition (*asynchronous composability*), providing a theoretical foundation for barrier-free execution. We then implement compliant centralized and decentralized coordination mechanisms within TianJi He et al. (2025), alongside an *in-flight mass model* that bounds physical communication latency as transient perturbations.

Our main contributions are:

- **Parameter Dynamics and Stability Criteria.** We formalize centroid shift as disagreement leakage via orthogonal decomposition, conditionally characterizing drift-induced degradation. We establish two operator criteria—*centroid preservation* and *disagreement non-expansion*—prove their closure under asynchronous composition, and derive long-term disagreement and learning-rate bounds under window contraction and closed-loop feedback.
- **Barrier-Free Coordination Mechanisms.** We construct compliant coordination atoms for centralized ($1/N$-normalized incremental commits) and decentralized (symmetric zero-sum pairwise mixing) topologies. Via our in-flight mass model, we formalize physical latency as capacity-bounded transient perturbations that incur no cumulative structural drift.
- **System Implementation and Empirical Evaluation.** Evaluated on TianJi across homogeneous clusters, three-node scaling, and dynamic-latency networks, our decentralized protocol (DB) achieves a $1.66\times$–$2.05\times$ speedup over synchronous execution (CS), reduces time-to-threshold by 40%–51%, and outperforms non-compliant unidirectional rings (DU) by 10.9%–46.1% in scaled environments.

2 Related Work

Multi-Learner Distributed DRL. Parallel DRL has advanced from decoupled sampling Nair et al. (2015); Horgan et al. (2018); Espeholt et al. (2020) to multi-learner architectures Huang et al. (2024); He et al. (2025). However, parameter coordination still largely depends on `AllReduce` or periodic barriers Huang et al. (2024); He et al. (2025); Mei et al. (2024), which throttle throughput when stragglers arise Dutta et al. (2018); Chen et al. (2016). While A3C demonstrated barrier-free updates Mnih et al. (2016), synchronous batched methods (A2C) outperformed it because policy staleness and gradient variance undermined sample efficiency Stooke & Abbeel (2018); Espeholt et al. (2018). Gossip-based frameworks such as GALA and PALA relaxed lockstep synchronization, yet their row-stochastic implementations frequently require mutex locks or epoch barriers across multi-node clusters Assran et al. (2019); Sun et al. (2023). We instead investigate whether consensus operators preserve the population centroid and contract disagreement under barrier-free asynchronous compositions.

Consensus Optimization and Parameter Dynamics. Average consensus and randomized gossip establish the mean-preserving and contractive properties of doubly stochastic matrices Olfati-Saber et al. (2007); Boyd et al. (2006); decentralized SGD extends this to interleaved optimization and communication Lian et al. (2017); Koloskova et al. (2020). In directed topologies, push-sum algorithms correct row-stochastic mixing bias with auxiliary weights **??**, while EASGD introduces elastic worker–center coupling **?**. In DRL, policy updates shift state visitation distributions, tightly coupling parameter dynamics to data generation Kakade & Langford (2002); Schulman et al. (2015); Sutton & Barto (2018). Lyu et al. (2025) analyzed single-learner stability via conformal Hamiltonian dynamics; we generalize dissipative trajectory perspectives to multi-replica systems via an orthogonal centroid–disagreement decomposition.

Asynchronous Execution and Perturbed Trajectories. Parameter servers and Stale Synchronous Parallel (SSP) models bound staleness via explicit versioning **??**. In lock-free execution,

Hogwild! and perturbed iterate frameworks model gaps between idealized sequences and physical operations Niu et al. (2011); **?**. Decoupling logical coordination from physical delay allows us to treat communication latency as an additive perturbation over reference consensus trajectories, yielding joint analytical disagreement bounds.

3 THEORETICAL FRAMEWORK

In barrier-free multi-learner DRL, local gradient updates, inter-replica parameter transfers, and physical network delays are tightly intertwined. This concurrency makes it challenging to decouple the adverse effects of parameter staleness from the structural biases introduced by the coordination mechanism itself. In this section, we first formalize the local conditions under which multiple learners can be treated as an ensemble evolving coherently around a shared reference policy. We then decompose the joint parameter dynamics into centroid driving, disagreement injection, and coordination-induced dissipation. This formulation frames parameter coordination as a consensus operator problem while explicitly retaining the closed-loop policy–data distribution feedback inherent to DRL.

3.1 DRIVEN–DISSIPATIVE MODELING: FROM SINGLE-REPLICA TO POPULATION DYNAMICS

Local Compatibility from Single Learner to Population. Let $\Omega \subseteq \mathbb{R}^d$ denote a locally convex, stable training region for an isolated learner, and suppose the current parameter replicas $\theta_1, \dots, \theta_N$ reside in Ω. If coordination proceeds via a non-negative, row-stochastic convex combination,

$$\theta_i^+ = \sum_{j=1}^{N} W_{ij}\theta_j, \qquad W_{ij} \geq 0, \qquad \sum_{j=1}^{N} W_{ij} = 1,$$

the convexity of Ω guarantees that $\theta_i^+ \in \mathrm{conv}\{\theta_1, \dots, \theta_N\} \subseteq \Omega$. Consequently, convex coordination operators cannot push parameters outside this locally stable basin. Defining the ensemble centroid and the root-mean-square consensus disagreement as

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N}\theta_i, \qquad R(\Theta) = \left(\frac{1}{N}\sum_{i=1}^{N}\|\theta_i - \bar{\theta}\|_2^2\right)^{1/2},$$

and assuming the policy parameterization is locally Lipschitz continuous on Ω Pirotta et al. (2013), i.e., $D_{\mathrm{TV}}(\pi_\theta, \pi_{\theta'}) \leq L_\pi\|\theta - \theta'\|_2$, we obtain

$$\frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{TV}}(\pi_{\theta_i}, \pi_{\bar{\theta}}) \leq L_\pi R(\Theta). \tag{1}$$

A complete derivation of Eq. equation 1 is provided in Appendix B.1. This bound confirms that within Ω, controlling parameter disagreement directly bounds the average policy deviation from the ensemble centroid. Under this regime, the centroid captures global task progress, while local replica dispersion provides exploratory diversity. Throughout this work, we take the local progress of isolated single-learner optimization as a given prerequisite and focus our analysis on the coupled multi-replica coordination dynamics.

Population Objective. Let $\bar{\theta} = \frac{1}{N}\sum_i \theta_i$ denote the population centroid parameter. We define the objective of the multi-learner ensemble as the expected discounted return under the centroid policy:

$$J(\bar{\theta}) := \mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}\Big[\sum_{t=0}^{\infty}\gamma_{\mathrm{rl}}^t r_t\Big], \tag{2}$$

where $\gamma_{\mathrm{rl}} \in (0, 1)$ is the discount factor. When replica i samples transitions under its local policy π_{θ_i}, the collected trajectories exhibit an off-policy discrepancy relative to the centroid policy $\pi_{\bar{\theta}}$. This discrepancy is locally bounded by the parameter distance:

$$D_{\mathrm{TV}}(\pi_{\theta_i}, \pi_{\bar{\theta}}) \leq L_\pi\|\theta_i - \bar{\theta}\|_2. \tag{3}$$

When disagreement is small, samples remain approximately on-policy; as disagreement grows, off-policy corrections such as V-trace Espeholt et al. (2018) are typically required. Here, we assume that consensus mechanisms maintain $R(x)$ within an admissible operating regime.

Driven–Dissipative Decomposition. Stacking the parameters into $x = \mathrm{col}(\theta_1, \ldots, \theta_N) \in \mathbb{R}^{Nd}$, we define the centroid projection matrix $\mathbf{J}$ and the disagreement projection matrix P as

$$\mathbf{J} = \frac{\mathbf{1}\mathbf{1}^\top}{N} \otimes I_d, \qquad P = I_{Nd} - \mathbf{J}.$$

These operators induce the orthogonal decomposition

$$\mathbb{R}^{Nd} = \mathcal{S}_\parallel \oplus \mathcal{S}_\perp, \qquad \mathcal{S}_\parallel = \mathrm{range}(\mathbf{J}), \qquad \mathcal{S}_\perp = \mathrm{range}(P),$$

where $\mathbf{J}x$ isolates the consensus state defined by the population centroid, Px captures local replica deviations from the centroid, and $R(x) = \|Px\|_2/\sqrt{N}$.

A round of local updates generates a stacked displacement $u \in \mathbb{R}^{Nd}$. Its consensus component $\mathbf{J}u$ drives the overall centroid, while its orthogonal component Pu perturbs replica dispersion. We quantify this disagreement injection as

$$\mathcal{D}(u) := \frac{1}{\sqrt{N}}\|Pu\|_2, \qquad R(x+u) \le R(x) + \mathcal{D}(u). \tag{4}$$

3.2 Stable Coordination Operators: Mitigating Structural Drift

Disagreement Injection in Closed-Loop DRL. Unlike distributed supervised optimization with static data distributions, parameter divergence in DRL alters behavioral policies and perturbs environmental state visitation distributions, establishing a closed feedback loop. In an MDP, the total variation distance between stationary distributions induced by different policies satisfies $\|d_{\pi_i} - d_{\bar{\pi}}\|_1 \le C_{\mathrm{MDP}} L_\pi R$, where C_{MDP} depends strictly on the discount factor and transition mixing times. Assuming locally Lipschitz policy mappings and occupancy measures, a first-order expansion yields an affine upper bound on disagreement injection:

$$\mathcal{D}(u) \le \beta_0^{\mathrm{evt}} + \kappa^{\mathrm{evt}} R(x) = \eta\sigma + \eta L_{\mathrm{rl}} R(x), \tag{5}$$

where $\beta_0^{\mathrm{evt}} = \eta\sigma$ captures local trajectory sampling variance, and $\kappa^{\mathrm{evt}} = \eta L_{\mathrm{rl}}$ characterizes the feedback gain of policy divergence propagated through shifted state visitations. Here, L_{rl} aggregates policy smoothness, occupancy perturbation bounds, and value bounds (see Appendix 19 for details). Equation equation 5 highlights the risk of self-reinforcing disagreement divergence, imposing an explicit lower bound on the dissipative rate required of coordination operators.

Conditional Characterization of Detrimental Drift. Our stability criteria constitute sufficient, rather than necessary, conditions. To characterize their boundary of necessity, consider the one-step centroid drift introduced by an arbitrary coordination operator: $\Delta_c := \frac{1}{\sqrt{N}}\|\mathbf{J}\mathcal{C}(x) - \mathbf{J}x\|_2 = \|\bar{\theta}^+ - \bar{\theta}\|_2$.

Proposition 1 (Sufficient Conditions for Drift-Induced Degradation)**.** *Let the ensemble objective be $J(\bar{\theta})$ from Eq. equation 2. Assume the policy parameterization satisfies $D_{\mathrm{TV}}(\pi_\theta, \pi_{\theta'}) \le L_\pi\|\theta - \theta'\|_2$ within the stable region Ω, and that J is locally Lipschitz. Then:*

$$\left|J(\pi_{\bar{\theta}'}) - J(\pi_{\bar{\theta}})\right| \le L_J L_\pi \Delta_c + O(\Delta_c^2), \tag{6}$$

where L_J is the local Lipschitz constant of the performance objective.

Proposition 1 indicates that the empirical severity of parameter drift depends on the magnitude of $L_J L_\pi \Delta_c$ relative to the effective single-step policy update. When coordination steps τ are small, replica update directions are homogeneous, and activation frequencies are balanced, Δ_c can be absorbed by local optimization steps without discernible degradation. Conversely, under large coordination intervals τ, heterogeneous gradients, or sustained activation imbalances, Δ_c can compound to the scale of nominal gradient updates, causing substantial performance loss. We therefore empirically evaluate these dynamics across heterogeneous and large-scale cluster configurations.

Criteria for Stable Coordination. Under the injection dynamics of Eq. equation 5, we impose the following sufficient operator conditions to guarantee coordination stability.

Definition 1 (Stable Coordination Operator). *A coordination operator $\mathcal{C} : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$ is stable if, for all $x \in \mathbb{R}^{Nd}$,*

$$\mathbf{J}\mathcal{C}(x) = \mathbf{J}x, \qquad \|P\mathcal{C}(x)\|_2 \le \|Px\|_2. \tag{7}$$

The first identity enforces centroid preservation, while the second guarantees disagreement non-expansion.

For a linear coordination operator $\mathcal{C}(x) = Wx$, centroid preservation is equivalent to $\mathbf{J}W = \mathbf{J}$; in particular, $\mathbf{J}WP = 0$ strictly prevents the leakage of disagreement components into the centroid trajectory. When this condition is violated, coordination perturbs the centroid trajectory away from the intended RL updates, introducing what we term *structural bias*. Casting these criteria into projection and algebraic matrix constraints enables the direct application of classical consensus theory to analyze operational composition and contraction rates.

3.3 Barrier-Free Asynchrony: Operator Composability and Long-Term Bounds

Asynchronous Composability. The stability criteria in Definition 1 are closed under finite function composition. If elementary coordination atoms $\mathcal{C}_1, \dots, \mathcal{C}_K$ each satisfy Definition 1, their arbitrary composition $\mathcal{C}_{K:1} = \mathcal{C}_K \circ \cdots \circ \mathcal{C}_1$ likewise preserves the centroid and ensures disagreement non-expansion. Consequently, the coordination layer requires no global synchronization barriers: arbitrarily interleaved sequences of valid coordination events introduce neither cumulative structural drift nor divergence in parameter disagreement. Physical non-idealities—such as concurrent read/write races and message transit delays—can thus be modeled modularly as bounded, exogenous additive perturbations to the underlying logical trajectory.

Lemma 1 (Linear Mixing Atoms). *Let $W \in \mathbb{R}^{N\times N}$ be a non-negative, symmetric, doubly stochastic, and positive semidefinite matrix. Then:*

$$\mathbf{J}W = W\mathbf{J} = \mathbf{J}, \qquad \|WP\|_2 \le 1.$$

Consequently, the operator $W \otimes I_d$ satisfies Definition 1.

The proof is given in Appendix D.1. This formulation mirrors established conditions in distributed consensus and decentralized optimization Nedić et al. (2009); Boyd et al. (2006); Koloskova et al. (2020).

Window Contraction and Long-Term Disagreement Bounds. Because an isolated pairwise coordination atom contracts disagreement along only a subset of spectral modes, strict contraction must be evaluated over finite logical event windows. Let Φ_t denote the ordered product of mixing matrices across the t-th window of length B, where B represents the number of atomic events (see Appendix A for event definitions). For the pairwise and centralized mixing families considered here, if the union graph of active communication edges within each window remains connected and effective mixing weights admit a uniform positive lower bound, standard consensus results ensure that

$$\gamma := \sup_t \|\Phi_t P\|_2 < 1. \tag{8}$$

We emphasize that this window B serves purely as an analytical tool and does not impose a physical synchronization barrier in the implementation.

Aggregating local policy updates and implementation errors into an additive disturbance ξ_s, Eq. equation 5 implies that the cumulative disagreement injected over a window satisfies

$$\sum_{s=tB}^{(t+1)B-1} \mathcal{D}(\xi_s) \le \beta_0 + \kappa R_t,$$

where $\beta_0 = \eta B\sigma$ and $\kappa = \eta B L_{\mathrm{rl}}$ denote window-level constants (see Eq. equation 5 for their event-level counterparts $\beta_0^{\mathrm{evt}}, \kappa^{\mathrm{evt}}$). Applying standard perturbed consensus recursions Tsitsiklis et al. (1986); Agarwal & Duchi (2011); Koloskova et al. (2020), we obtain

$$R_{t+1} \le (\gamma + \kappa)R_t + \beta_0. \tag{9}$$

Algorithm 1 Centralized Asynchronous Mean Commit with Delayed Pulls

Init: Coordinator sets θ_c; each learner i sets $b_i \leftarrow \theta_c$
for all learners i asynchronously in parallel **do**
  loop
    Evolve τ local steps from b_i to get θ_i^{loc}; set $p_i \leftarrow \theta_i^{\mathrm{loc}} - b_i$
    Send (i, p_i) to center; Center sets $\theta_c \leftarrow \theta_c + p_i/N$
    Learner refreshes $b_i \leftarrow \theta_c$, $\theta_i \leftarrow b_i$
  end loop
end for

When $\gamma + \kappa < 1$, unrolling recursion equation 9 yields

$$R_t \le (\gamma+\kappa)^t R_0 + \frac{1-(\gamma+\kappa)^t}{1-\gamma-\kappa}\beta_0, \qquad \limsup_{t\to\infty} R_t \le \frac{\beta_0}{1-\gamma-\kappa}. \tag{10}$$

The formal proof and detailed derivations relating β_0 and κ to local learning rates, update frequencies, and hardware delays appear in Appendices C.2 and F.3.

Equation equation 10 demonstrates that in closed-loop DRL, asymptotic replica dispersion is jointly governed by gradient noise β_0, feedback sensitivity κ, and operator contraction rate $1-\gamma$. Combining this with Eq. equation 1, the asymptotic policy deviation is bounded by

$$\limsup_{t\to\infty} \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{TV}}(\pi_{\theta_i^t}, \pi_{\bar{\theta}^t}) \le \frac{L_\pi \beta_0}{1-\gamma-\kappa}.$$

Therefore, while stable single-learner updates provide the underlying driving force, the coordination protocol must satisfy centroid preservation and disagreement non-expansion. Furthermore, ensuring $\gamma + \kappa < 1$ is vital to suppress both the closed-loop policy–data feedback and the physical communication perturbations introduced by asynchronous execution. Under the first-order approximations $\kappa = \eta B L_{\mathrm{rl}}$ and $\beta_0 = \eta B \sigma$ from Appendix C.2, the strict contraction condition $\gamma + \kappa < 1$ translates directly into an upper bound on the learning rate:

$$\eta < \frac{1-\gamma}{B L_{\mathrm{rl}}}.$$

This highlights how long-term temporal dependencies and closed-loop distribution shifts in DRL impose fundamental constraints on admissible local step sizes and coordination window lengths.

4 Algorithmic Implementations

We realize stable coordination in centralized and decentralized topologies using a two-layer abstraction:

- **Logical Commit Layer:** Commits are instantaneous atomic operations that satisfy Definition 1.
- **Physical Execution Layer:** Finite transit delays are captured via an **in-transit mass** model **?**. Under reliable, bounded-latency transmission, unabsorbed updates remain capacity-bounded transient perturbations rather than compounding drift.

4.1 Centralized Implementation: Normalized Incremental Commits

Logical Commit. The coordinator maintains global mean θ_c. Learner i computes increment $p_i = \theta_i^{\mathrm{loc}} - b_i$ over τ local steps from baseline b_i, and commits (i, p_i). The center updates $\theta_c \leftarrow \theta_c + p_i/N$. For stacked state y, this commit applies $\mathcal{C}(y) = \mathbf{J}y$, satisfying $\mathbf{J}\mathcal{C}(y) = \mathbf{J}y$ and $P\mathcal{C}(y) = 0$ (Algorithm 1).

In-Transit Mass Bound. If the coordinator absorbs $K_i \le B_c$ commits from other learners while learner i drifts by q_i, the unpulled centralized mass is:

$$m_i^c := \frac{1}{N}\sum_{s=1}^{K_i} p^{(s)}, \qquad \widehat{\theta}_i - \theta_c = q_i - m_i^c. \tag{11}$$

Algorithm 2 Decentralized Pairwise Asynchronous Consensus

1: **for all** activated edges (i, j) initiated by i **do**
2: Node i transmits snapshot $s_i(k)$; node j receives it at $k + t$
3: Node j computes $\delta_{ij} \leftarrow \frac{\alpha}{2}\big(s_j(k + t) - s_i(k)\big)$, applies $-\delta_{ij}$, and returns $+\delta_{ij}$ to node i
4: Node i applies $+\delta_{ij}$ upon receipt at $k + 2t$
5: **end for**

Assuming $\|p^{(s)}\|, \|q_i\| \le G_p$, the physical deviation is bounded by:

$$\|\widehat{\theta}_i - \theta_c\| \le \Big(1 + \frac{B_c}{N}\Big) G_p \triangleq \varepsilon_c, \qquad R^2(\widehat{x}) \le \varepsilon_c^2. \tag{12}$$

Periodic pulls reset m_i^c to zero, bounding physical disagreement without altering the logical centroid.

4.2 Decentralized Implementation: Symmetric Pairwise Contraction

Logical Commit. Along active edge (i, j), learners apply symmetric mixing:

$$\theta_i^+ = \theta_i + \frac{\alpha}{2}(\theta_j - \theta_i), \qquad \theta_j^+ = \theta_j - \frac{\alpha}{2}(\theta_j - \theta_i), \qquad \alpha \in (0, 1]. \tag{13}$$

The matrix $W_{ij} = I - \frac{\alpha}{2}(e_i - e_j)(e_i - e_j)^\top$ is doubly stochastic and PSD, ensuring $\|W_{ij}P\|_2 \le 1$ and driving window contraction: $R_{t+1} \le \gamma R_t + \beta_t$ (Algorithm 2).

Channel In-Transit Mass. During physical transit, the in-flight correction forms a mass vector:

$$\mu^d(t) := \sum_{(i \leftarrow j) \in \mathcal{M}_{\text{flight}}(t)} \delta_{ij} \otimes \mathbf{e}_i. \tag{14}$$

The physical state satisfies $\widehat{x} = x^\star - \mu^d + e_{\text{stale}}$, preserving the augmented centroid: $\frac{1}{N}\sum_i \widehat{\theta}_i + \frac{1}{N}\sum \delta_{ij} = \bar{\theta}^\star$. If $|\mathcal{M}_{\text{flight}}| \le M_{\text{p}}$ and $\|\delta_{ij}\| \le D_{\max}$, physical disagreement satisfies:

$$R(x^\star - \mu^d) \le R(x^\star) + \eta_{\text{p}}, \qquad \eta_{\text{p}} := M_{\text{p}} \frac{\sqrt{N-1}}{N} D_{\max}. \tag{15}$$

At window boundaries:

$$R(\widehat{x}^{tB}) \le (\gamma + \kappa)^t R(z^0) + \frac{1 - (\gamma + \kappa)^t}{1 - \gamma - \kappa} \beta_0^{\text{impl}} + \eta_{\text{p}}. \tag{16}$$

The instantaneous physical centroid deviation is $\frac{M_{\text{p}} D_{\max}}{N} = \mathcal{O}(1/N)$, precluding permanent drift.

4.3 Baselines Violating Coordination Criteria

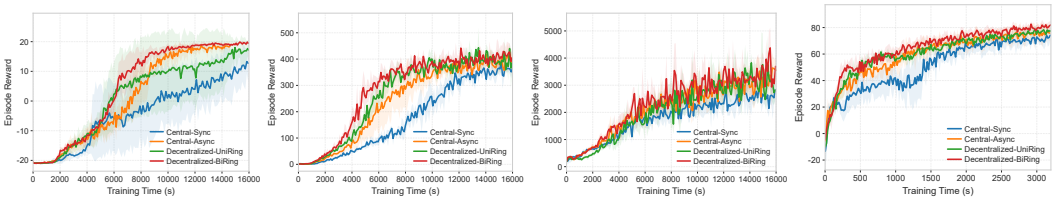
Unnormalized Commits (A3C-style). Using $\theta_c^+ = \theta_c + p_i$ introduces a per-commit shift of $(1 - \frac{1}{N})p_i$. Over K asynchronous commits, drift accumulates to $\mathcal{O}\big(\frac{N-1}{N} K G_p\big)$, diverging with iteration count.

Unidirectional Ring Propagation. Setting $\theta_j^+ = (1-\alpha)\theta_j + \alpha\theta_i$ with static θ_i shifts the centroid by $\frac{\alpha}{N}(\theta_i - \theta_j)$. If edge discrepancies have collinear components, drift compounds to $\mathcal{O}\big(\frac{\alpha K_T}{N}\overline{\Delta}\big)$.

5 Empirical Evaluation

We investigate three empirical questions:

1. **RQ1 (Speedup):** Does barrier-free coordination convert raw concurrency into wall-clock speedups while maintaining sample efficiency?
2. **RQ2 (Stability Criteria):** Does violating stability criteria under dynamic latency cause empirical degradation as predicted by theory?
3. **RQ3 (Straggler Robustness):** How robustly do centralized vs. decentralized topologies preserve the centroid under persistent stragglers?



(a) `Pong` ($N = 15$) (b) `Breakout` ($N = 15$) (c) `Assault` ($N = 9$) (d) `Boxing` ($N = 21$)

Figure 1: **Wall-clock training curves on the three-node cluster** (rolling mean ± 1 std over 4 runs).

Table 1: **Training efficiency across base and three-node scaled clusters**. T_{tgt}: wall-clock time to target; Speedup is relative to CS; bold indicates best performance.

		Base Scale				Scaled Cluster (3 Nodes)			
Task	**Method**	T_{tgt} (s) ↓	**Speedup** ↑	Tail Return ↑	AUC ↑	T_{tgt} (s) ↓	**Speedup** ↑	Tail Return ↑	AUC ↑
`Pong`	CS (Sync Baseline)	15238.1	1.00×	17.27 ± 2.51	-1.61	15838.1	1.00×	9.93 ± 8.24	-4.23
	CA (Ours, Central)	12566.5	1.21×	18.41 ± 1.81	**3.89**	9172.4	1.73×	18.06 ± 1.98	0.55
	DU (Ablation Control)	**11800.3**	**1.29**×	**18.96 ± 1.28**	2.14	14362.7	1.10×	15.52 ± 5.60	-0.01
	DB (Ours, Decentral)	13561.6	1.12×	18.42 ± 1.52	2.50	**7744.0**	**2.05**×	**19.31 ± 1.15**	**4.77**
`Breakout`	CS (Sync Baseline)	15060.1	1.00×	370.17 ± 63.97	176.17	15153.9	1.00×	357.17 ± 54.65	174.99
	CA (Ours, Central)	**8091.2**	**1.86**×	**413.06 ± 69.64**	**278.03**	13219.1	1.15×	382.81 ± 84.30	235.98
	DU (Ablation Control)	10815.7	1.39×	381.19 ± 60.67	249.58	9697.1	1.56×	395.50 ± 74.39	264.17
	DB (Ours, Decentral)	10276.6	1.47×	399.17 ± 61.91	243.57	**8636.5**	**1.76**×	**412.69 ± 74.31**	**282.84**
`Assault`	CS (Sync Baseline)	15114.8	1.00×	2798.64 ± 955.10	1871.72	13531.0	1.00×	2526.57 ± 554.08	1773.40
	CA (Ours, Central)	**8745.1**	**1.73**×	**3680.22 ± 1310.0**	**2724.22**	9789.8	1.38×	3067.51 ± 892.58	2134.37
	DU (Ablation Control)	11303.4	1.34×	2982.80 ± 1030.7	2393.24	9580.8	1.41×	2973.94 ± 895.05	2080.26
	DB (Ours, Decentral)	8905.8	1.70×	3077.10 ± 890.71	2512.38	**7590.4**	**1.78**×	**3545.86 ± 1149.9**	**2391.90**
`Boxing`	CS (Sync Baseline)	3176.9	1.00×	75.44 ± 6.00	62.38	3191.3	1.00×	71.43 ± 7.51	52.04
	CA (Ours, Central)	2474.6	1.28×	**80.22 ± 5.14**	**66.12**	2345.7	1.36×	76.20 ± 7.41	60.07
	DU (Ablation Control)	**2278.9**	**1.39**×	79.11 ± 6.24	64.32	2337.1	1.37×	76.73 ± 4.86	60.15
	DB (Ours, Decentral)	2664.4	1.19×	77.98 ± 5.86	62.73	**1919.5**	**1.66**×	**79.35 ± 5.57**	**65.14**

5.1 Experimental Setup

Baselines and Configurations. We evaluate DQN on TianJi He et al. (2025) across four Atari games (`Pong`, `Breakout`, `Assault`, and `Boxing`): (1) **CS** (Central-Sync): Synchronous baseline with bounded centralized averaging; (2) **CA** (Central-Async): Centralized $1/N$-incremental commit mechanism; (3) **DB** (Decentral-BiRing): Decentralized symmetric zero-sum pairwise protocol ($\alpha = 1/2$); (4) **DU** (Decentral-UniRing): Unidirectional asymmetric mixing ablation baseline. Base-scale setups evaluate $N \in [3, 7]$; scaled clusters evaluate $N \in [9, 21]$ across three nodes. Metrics report mean $\pm$ standard deviation over four seeds. Target thresholds are: `Pong`: 18, `Breakout`: 400, `Assault`: 3000, and `Boxing`: 80.

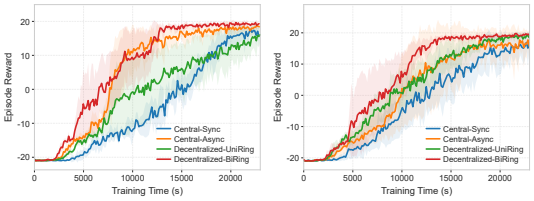
5.2 RQ1: End-to-End Efficiency and Cluster Scalability

Speedup via Barrier-Free Execution. As shown in Figure 1 and Table 1, synchronous barriers bottleneck CS under straggler tail latency. Eliminating barriers enables DB to achieve a $1.66\times$–$2.05\times$ speedup over CS on three nodes (40%–51% time-to-target reduction) while attaining top tail return and AUC. While CA excels at base scale, aggregation contention diminishes its scaling efficiency on larger clusters (`Breakout`, `Assault`).

Preserving Sample Efficiency. Throughput gains do not sacrifice sample efficiency. On three-node `Pong`, DB requires $(1.01 \pm 0.19) \times 10^6$ environment steps versus $(1.68 \pm 0.67) \times 10^6$ for CS; in `Assault`, CA and DB cut sample requirements by nearly 40%. Stable centroid preservation ensures that replica dispersion acts as implicit exploration without introducing divergence.

5.3 RQ2: Validation of Stability Criteria under Dynamic Latency (DB vs. DU)

Scaling Accentuates Drift. At base scale, DB and DU exhibit comparable convergence (Table 1). However, on scaled clusters, DB consistently outperforms DU, reducing time-to-target by 46.1%, 10.9%, 20.8%, and 17.9% on `Pong`, `Breakout`, `Assault`, and `Boxing`. Unidirectional drift is benign in small homogeneous groups, but centroid preservation becomes critical as concurrency scales.



(a) Seed = 2025 (b) Seed = 2026

Figure 3: **Convergence under dynamic latency on `Pong`** (mean ± 1 std, 4 runs). DB exhibits consistent stability, while DU degrades markedly.

Table 2: **DB vs. DU under three-node dynamic latency**. T_{mean}: average time-to-threshold; N/A: budget exceeded.

Task (Threshold)	Method	T_{mean} (ks)↓	AUC ↑	Tail Return ↑
`Pong`	DU	N/A	-1.60	13.66
(18)	DB	**12.68**	**6.58**	**19.15**
`Breakout`	DU	13.91	156.10	338.45
(400)	DB	**10.85**	**184.69**	**358.68**
`Assault`	DU	**6.82**	2452.48	3341.03
(3000)	DB	8.39	**2541.93**	**3676.36**
`Boxing`	DU	3.75	44.29	**65.85**
(80)	DB	**3.41**	**46.93**	61.09

Robustness Under Network Jitter. Under dynamic latency traces (Table 2), DB attains higher AUC across all tasks. On `Pong`, DU fails to reach the target threshold within the compute budget across multiple seeds (Figures 2a and 2b), whereas DB converges stably in $12.68\,\text{ks}$. This confirms that asymmetric mixing leaves policies vulnerable to latency jitter due to compounded structural drift.

5.4 RQ3: Centroid Robustness under Persistent Stragglers

In stress tests with persistent stragglers (Appendix J), centralized CA demonstrates the highest robustness. In ring topologies (DB and DU), a persistent straggler creates a localized bottleneck that degrades effective spectral connectivity γ. On `Boxing`, larger learner counts and smoother RAM-state dynamics mitigate straggler impact across all methods.

6 Conclusion

We modeled multi-learner DRL as a driven–dissipative dynamical system, decomposing parameter evolution into centroid driving, disagreement injection, and coordination-induced dissipation. We established two operator-theoretic stability criteria—centroid preservation and disagreement non-expansion—proving their closure under asynchronous composition and deriving long-term consensus bounds under policy–data feedback. Implementing these principles via centralized normalized commits and decentralized zero-sum mixing, we demonstrated that physical delays act as bounded transient mass without inducing cumulative drift. Evaluations in TianJi confirm that stable barrier-free coordination achieves a $1.66\times$–$2.05\times$ speedup over synchronous baselines and prevents the degradation observed in non-centroid-preserving protocols.

Limitations and Future Work. Our theoretical guarantees rely on analytical working assumptions common in DRL analysis, such as local parameter convexity and Lipschitz continuity, which lack rigorous general guarantees in deep networks. Furthermore, while treating the ensemble centroid as an arithmetic mean aligns with standard distributed optimization, it leaves open the investigation of non-uniform or adaptive policy-space aggregation schemes. In addition, while we frame distributed DRL as a driven–dissipative system, a rigorous characterization of the disagreement injection operator itself remains an open problem. At the systems level, our implementations adopt minimal topologies; exploring topology-aware asynchronous primitives presents a promising direction. Finally, our experiments focus on CPU clusters, DQN, and Atari tasks; verifying the generalizability of these findings across larger actor-critic architectures and continuous control domains remains an objective for future work.

AI use statement

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools (large language models) for the following tasks with required disclosure:

1. **Code development assistance.** AI tools assisted in drafting, debugging, and refactoring parts of the implementation code for the coordination mechanisms and the experimental pipeline.
2. **Proof cross-checking.** While deriving the theoretical results, we used AI tools in an iterative, back-and-forth process to cross-check individual derivation steps, flag potential gaps or typos, and verify algebraic manipulations.

We have *not* used generative AI tools as a source of the scientific contributions of this work: the research ideas, problem formulation, theoretical framework, algorithm design, system architecture, and the design and execution of all experiments were conceived and carried out entirely by the authors. Generative AI tools were not used to autonomously generate scientific text, figures, data, or experimental results in this paper.

Additionally, we used generative AI tools for the following tasks with recommended disclosure:

1. **Literature search and discovery.** AI tools were used to help locate and surface relevant prior work; all retrieved references were then read, verified, and curated manually by the authors.
2. **Ideation discussions.** During early-stage conceptualization, we discussed preliminary ideas with AI tools as a sounding board; these discussions did not originate the central ideas, novelty claims, or technical approach of this paper.
3. **Language polishing.** AI tools assisted in improving the grammar, phrasing, and fluency of the English text.

We have reviewed all AI-assisted work. Specifically, all AI-suggested references were manually verified against the original publications; AI-assisted code was reviewed line by line, tested, and validated through the experiments reported in this paper; every proof step flagged during AI-assisted cross-checking was independently re-derived and confirmed by the authors; and AI-polished text was checked to ensure that it faithfully conveys our intended meaning without introducing claims beyond our own results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Reproducibility statement

It is important that the work published in ICLR is reproducible. We provide detailed descriptions of the experimental setup, including hardware configuration, resource orchestration, hyperparameter settings, and evaluation metrics in Appendix H. The delay injection protocol for dynamic network perturbation experiments is described in Appendix I. Complete proofs of all theoretical results are provided in the appendices. We plan to release the source code and detailed configuration files as supplementary materials to facilitate reproduction of all experimental results.

References

Alekh Agarwal and John C. Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 24, pp. 873–881. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/hash/f0e52b27a7a5d6a1a87373dffa53dbe5-Abstract.html.

Mahmoud Assran, Joshua Romoff, Nicolas Ballas, Joelle Pineau, and Michael Rabbat. Gossip-based actor-learner architectures for deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 13299–13309, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/a1a527267c0d33a86382a03c4c721cd2-Paper.pdf.

Mohammad Babaeizadeh, Iuri Frosio, Stephen Tyree, Jason Clemons, and Jan Kautz. Reinforcement learning through asynchronous advantage actor-critic on a gpu. In *International Conference*

on Learning Representations (ICLR), 2017. URL https://mlanthology.org/iclr/2017/babaeizadeh2017iclr-reinforcement/.

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006. doi: 10.1109/TIT.2006.874516. URL https://doi.org/10.1109/TIT.2006.874516.

Jianmin Chen, Rajat Monga, Samy Bengio, and Rafal Jozefowicz. Revisiting distributed synchronous SGD. In *International Conference on Learning Representations Workshop Track (ICLR Workshop)*, 2016. URL https://arxiv.org/abs/1604.00981.

Sanghamitra Dutta, Gauri Joshi, Soumyadip Ghosh, Parijat Dube, and Priya Nagpurkar. Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84 of *Proceedings of Machine Learning Research*, pp. 803–812. PMLR, 2018. URL https://proceedings.mlr.press/v84/dutta18a.html.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1407–1416. PMLR, 2018. URL https://proceedings.mlr.press/v80/espeholt18a.html.

Lasse Espeholt, Raphaël Marinier, Piotr Stanczyk, Ke Wang, and Marcin Michalski. SEED RL: Scalable and efficient deep-RL with accelerated central inference. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1910.06591.

Zhouyu He, Peng Qiao, Rongchun Li, Yong Dou, and Yusong Tan. Highly parallelized reinforcement learning training with relaxed assignment dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pp. 17159–17167, 2025. doi: 10.1609/aaai.v39i16.33886. URL https://ojs.aaai.org/index.php/AAAI/article/view/33886.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://mlanthology.org/iclr/2018/horgan2018iclr-distributed/.

Shengyi Huang, Jiayi Weng, Rujikorn Charakorn, Min Lin, Zhongwen Xu, and Santiago Ontañón. Cleanba: A reproducible and efficient distributed reinforcement learning platform. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://mlanthology.org/iclr/2024/huang2024iclr-cleanba/.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pp. 267–274, 2002. URL https://mlanthology.org/icml/2002/kakade2002icml-approximately/.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized SGD with changing topology and local updates. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5381–5393. PMLR, 2020. URL https://proceedings.mlr.press/v119/koloskova20a.html.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 5330–5340, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/f75526659f31040afeb61cb7133e4e6d-Abstract.html.

Yao Lyu, Xiangteng Zhang, Shengbo Eben Li, Jingliang Duan, Letian Tao, Qing Xu, Lei He, and Keqiang Li. Conformal symplectic optimization for stable reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):10791–10804, 2025. doi: 10.1109/TNNLS.2024.3507015.

Zhiyu Mei, Wei Fu, Guangju Wang, Huanchen Zhang, and Yi Wu. Scaling distributed reinforcement learning to over ten thousand cores. In *Proceedings of the 19th European Conference on Computer Systems (EuroSys)*, pp. 837–852, 2024.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937. PMLR, 2016. URL https://proceedings.mlr.press/v48/mniha16.html.

Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, Shane Legg, Volodymyr Mnih, Koray Kavukcuoglu, and David Silver. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*, 2015. URL https://arxiv.org/abs/1507.04296. Presented at the Deep Learning Workshop, International Conference on Machine Learning (ICML).

Angelia Nedić, Alex Olshevsky, Asuman Ozdaglar, and John N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009. doi: 10.1109/TAC.2009.2031203. URL https://doi.org/10.1109/TAC.2009.2031203.

Feng Niu, Benjamin Recht, Christopher Ré, and Stephen J. Wright. HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*, volume 24, pp. 693–701. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/hash/218a0aefd1d1a4be65601cc6ddc1520e-Paper.pdf.

Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007. doi: 10.1109/JPROC.2006.887293. URL https://doi.org/10.1109/JPROC.2006.887293.

Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Adaptive step-size for policy gradient methods. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

Adam Stooke and Pieter Abbeel. Accelerated methods for deep reinforcement learning. *arXiv preprint arXiv:1803.02811*, 2018. URL https://arxiv.org/abs/1803.02811.

Zhenglun Sun, Peng Qiao, Yong Dou, Qingqing Li, and Rongchun Li. PALA: Parallel actor-learner architecture for distributed deep reinforcement learning. *Chinese Journal of Computers*, 46(2):229–243, 2023. doi: 10.11897/SP.J.1016.2023.00229. in Chinese.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts, second edition, 2018. ISBN 978-0-262-03924-6. URL http://incompleteideas.net/book/the-book-2nd.html.

John N. Tsitsiklis, Dimitri P. Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986. doi: 10.1109/TAC.1986.1104412. URL https://doi.org/10.1109/TAC.1986.1104412.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=H1gX8C4YPr.

A Notation, Basic Tools, and Accounting Conventions

Notational conventions. Throughout this paper, $J(\cdot)$ refers exclusively to the performance functional $J(\theta) = J(\pi_\theta)$. The averaging projection operator is denoted $\mathbf{J}$ (parameter space) or $\bar{J}$ (node space). Let $N \geq 2$ parameter replicas $\theta_i \in \mathbb{R}^d$ be given, with stacked state $x = \mathrm{col}(\theta_1, \dots, \theta_N) \in \mathbb{R}^{Nd}$. Let $a_i \in \mathbb{R}^N$ denote the standard basis vectors, $\mathbf{1} = \sum_i a_i$, and define

$$\bar{J} = \frac{1}{N}\mathbf{1}\mathbf{1}^\top, \qquad \mathbf{J} = \bar{J} \otimes I_d, \qquad P = I_{Nd} - \mathbf{J} = (I_N - \bar{J}) \otimes I_d.$$

Since $\bar{J}$ is symmetric and idempotent, $\mathbf{J}$ and P are symmetric idempotent, P is a nonzero orthogonal projection, and $\|P\|_2 = 1$. For any $v = \mathrm{col}(v_1, \dots, v_N)$,

$$(\mathbf{J}v)_i = \bar{v}, \qquad (Pv)_i = v_i - \bar{v}, \qquad \|\mathbf{J}v\|_2^2 = N\|\bar{v}\|_2^2. \tag{17}$$

The centroid, variance, and disagreement are defined as

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^N \theta_i, \qquad V(\Theta) = \frac{1}{N}\sum_{i=1}^N \|\theta_i - \bar{\theta}\|_2^2, \qquad R(x) = \sqrt{V(\Theta)}.$$

From equation 17,

$$R(x) = \frac{1}{\sqrt{N}}\|Px\|_2. \tag{18}$$

Lemma 2 (Minimality of the mean)**.** *For any $v_1, \dots, v_N \in \mathbb{R}^d$ and $c \in \mathbb{R}^d$,*

$$\frac{1}{N}\sum_i \|v_i - \bar{v}\|_2^2 = \frac{1}{N}\sum_i \|v_i - c\|_2^2 - \|\bar{v} - c\|_2^2 \leq \frac{1}{N}\sum_i \|v_i - c\|_2^2.$$

Proof. Expand $\|v_i - c\|_2^2$; the cross terms sum to zero over i. □

Two clocks. The atomic event clock s indexes single local updates, single commits, or one half of a pairwise transaction taking effect; the round clock t indexes complete rounds where all N replicas each execute one update. The window length is B events. Physical execution and in-transit mass are stated on the atomic event clock; RL injection and window recursion are stated on the round clock, with conversions made explicit subsequently.

Policy TV distance. We use the state-uniform total variation

$$D_{\mathrm{TV}}(\pi_\theta, \pi_{\theta'}) = \sup_s D_{\mathrm{TV}}\big(\pi_\theta(\cdot|s), \pi_{\theta'}(\cdot|s)\big), \qquad D_{\mathrm{TV}}(p, q) = \tfrac{1}{2}\sum_a |p(a) - q(a)|.$$

Disagreement injection operator.

Definition 2 (Disagreement injection)**.** *For a displacement $u = \mathrm{col}(u_1, \dots, u_N)$, define*

$$\mathcal{D}(u) = \frac{1}{\sqrt{N}}\|Pu\|_2 = \Big(\frac{1}{N}\sum_i \|u_i - \bar{u}\|_2^2\Big)^{1/2}.$$

By linearity of P and the triangle inequality,

$$R(x + u) \leq R(x) + \mathcal{D}(u). \tag{19}$$

If only node i undergoes displacement p, i.e., $u = a_i \otimes p$, then $\bar{u} = p/N$, and direct computation yields

$$\mathcal{D}(u) = \frac{\sqrt{N-1}}{N}\|p\|_2. \tag{20}$$

B Policy Disagreement Bounds and Drift Harm

The following holds pointwise within the locally stable region $\Omega \subset \mathbb{R}^d$, where Ω is convex. We assume:

(B1) $|r_t| \le R_{\max}$;

(B2) $\psi_\theta(s,a) = \nabla_\theta \log \pi_\theta(a|s)$ satisfies $\sup_{s,a} \|\psi_\theta\|_2 \le G_{\max}$, and $\|\psi_\theta - \psi_{\theta'}\|_2 \le L_\psi \|\theta - \theta'\|_2$;

(B3) $D_{\mathrm{TV}}(\pi_\theta, \pi_{\theta'}) \le L_\pi \|\theta - \theta'\|_2$;

(B4) $g_i = \nabla J(\theta_i) + \varepsilon_i$, $\|\varepsilon_i\|_2 \le \sigma$.

B.1 From Policy Disagreement to Average TV Deviation

If all $\theta_i \in \Omega$, then $\bar\theta \in \Omega$. By (B3) and Cauchy–Schwarz,

$$\frac{1}{N}\sum_i D_{\mathrm{TV}}(\pi_{\theta_i}, \pi_{\bar\theta}) \le L_\pi \frac{1}{N}\sum_i \|\theta_i - \bar\theta\|_2 \le L_\pi R(\Theta),$$

which is Eq. equation 1 in the main text.

B.2 Occupancy Measure and Q-Value Perturbation

Let P_π be the policy-induced state kernel, and d_π the normalized discounted occupancy measure.

Lemma 3 (Occupancy measure deviation)**.** *For any π, π',*

$$\|d_\pi - d_{\pi'}\|_1 \le C_{\mathrm{MDP}} D_{\mathrm{TV}}(\pi, \pi'), \qquad C_{\mathrm{MDP}} = \frac{2\gamma_{\mathrm{rl}}}{1-\gamma_{\mathrm{rl}}}.$$

Proof. The single-step kernel difference satisfies $\|(P_\pi - P_{\pi'})\nu\|_1 \le 2D_{\mathrm{TV}}(\pi, \pi')$; telescoping and using the L_1 non-expansiveness of stochastic kernels gives $\|P_\pi^t \mu - P_{\pi'}^t \mu\|_1 \le 2t D_{\mathrm{TV}}(\pi, \pi')$; summing over the discounted series $\sum_{t\ge 0} t\gamma_{\mathrm{rl}}^t$ yields the result. □

Lemma 4 (Q-value perturbation)**.** *Under (B1),*

$$\|Q^\theta - Q^{\theta'}\|_\infty \le \frac{2R_{\max}}{(1-\gamma_{\mathrm{rl}})^2} D_{\mathrm{TV}}(\pi_\theta, \pi_{\theta'}). \tag{21}$$

Proof. From the Bellman equation, $Q^\theta - Q^{\theta'}$ is controlled by the discounted expectation of $V^\theta - V^{\theta'}$; the V difference is controlled by the standard performance difference lemma and (B1). □

B.3 Policy Gradient Lipschitz Constant

Denote $G(\theta) = \nabla J(\theta)$. By the policy gradient theorem,

$$G(\theta) = \frac{1}{1-\gamma_{\mathrm{rl}}} \sum_{s,a} \rho_\theta(s,a)\psi_\theta(s,a)Q^\theta(s,a).$$

Inserting $\pm$ terms:

$$\begin{aligned}(1-\gamma_{\mathrm{rl}})\big(G(\theta) - G(\theta')\big) = &\sum_{s,a}\big(\rho_\theta - \rho_{\theta'}\big)\psi_\theta Q^\theta + \sum_{s,a}\rho_{\theta'}\big(\psi_\theta - \psi_{\theta'}\big)Q^\theta \\ &+ \sum_{s,a}\rho_{\theta'}\psi_{\theta'}\big(Q^\theta - Q^{\theta'}\big).\end{aligned}$$

The three terms are controlled by occupancy measure deviation, score function Lipschitz continuity, and Q-value perturbation, respectively:

$$\|\rho_\theta - \rho_{\theta'}\|_1 \le \|d_\theta - d_{\theta'}\|_1 + 2D_{\mathrm{TV}}(\pi_\theta, \pi_{\theta'}),$$

hence

$$\|G(\theta) - G(\theta')\|_2 \le L_{\mathrm{rl}} \|\theta - \theta'\|_2,$$

where

$$L_{\mathrm{rl}} = \frac{4 G_{\max} R_{\max} L_\pi}{(1-\gamma_{\mathrm{rl}})^3} + \frac{R_{\max} L_\psi}{(1-\gamma_{\mathrm{rl}})^2}. \tag{22}$$

The L_{rl} referenced in the main text is given by the above expression.

B.4 Proof of Proposition 1

The standard performance difference lemma gives

$$J(\pi) - J(\pi') = \frac{1}{1-\gamma_{\mathrm{rl}}} \sum_s d_\pi(s) \sum_a (\pi(a|s) - \pi'(a|s)) Q^{\pi'}(s,a). \tag{23}$$

Under (B1),

$$|J(\pi) - J(\pi')| \le L_J D_{\mathrm{TV}}(\pi, \pi'), \qquad L_J = \frac{2 R_{\max}}{(1-\gamma_{\mathrm{rl}})^2}. \tag{24}$$

Let the coordination map $\mathcal{C}$ send centroid $\bar\theta$ to $\bar\theta'$, and define $\Delta_c = \frac{1}{\sqrt{N}} \|\mathbf{J}\mathcal{C}(x) - \mathbf{J}x\|_2$. From equation 17,

$$\Delta_c = \|\bar\theta' - \bar\theta\|_2. \tag{25}$$

Proof of Proposition 1. First-order (Lipschitz) bound. By Eq. equation 24 and (B3), when $\bar\theta, \bar\theta' \in \Omega$,

$$|J(\pi_{\bar\theta'}) - J(\pi_{\bar\theta})| \le L_J \, D_{\mathrm{TV}}(\pi_{\bar\theta'}, \pi_{\bar\theta}) \le L_J L_\pi \|\bar\theta' - \bar\theta\|_2 = L_J L_\pi \Delta_c.$$

Second-order (Taylor) bound. If J is twice continuously differentiable in Ω with $\sup_{\theta\in\Omega} \|\nabla^2 J(\theta)\|_2 \le H_{\max}$, then

$$|J(\pi_{\bar\theta'}) - J(\pi_{\bar\theta})| \le G_\nabla \Delta_c + \tfrac{1}{2} H_{\max} \Delta_c^2, \qquad G_\nabla = \sup_\Omega \|\nabla J\|_2.$$

The final drift harm bound takes the minimum of the two:

$$|J(\pi_{\bar\theta'}) - J(\pi_{\bar\theta})| \le \min\Big\{ L_J L_\pi \Delta_c,\ G_\nabla \Delta_c + \tfrac{1}{2} H_{\max} \Delta_c^2 \Big\}. \tag{26}$$

□

C RL Closed-Loop Injection and Long-Term Disagreement Bounds

Accounting conventions. The main text uses fully parallel round accounting: one round consists of all replicas each executing one step with step size η; a window contains B injection events, of which $H \le B$ are RL injection events.

For each *event class*, only one accounting scheme is used:

- RL gradient injection events use aligned accounting:

$$\mathcal{D}(\xi_r) \le \beta_0^{\mathrm{evt}} + \kappa^{\mathrm{evt}} R(x_r), \qquad \beta_0^{\mathrm{evt}} = \eta\sigma, \quad \kappa^{\mathrm{evt}} = \eta L_{\mathrm{rl}}.$$

- Non-RL implementation-layer events (e.g., completion of stale transactions, non-RL local updates) use a conservative additive budget, denoted β^{impl} for their total additive injection within a window, without R-dependent terms.

These act on different event classes and are not double-counted; they may appear simultaneously in the window partial sum but do not mix accounting schemes for the same event.

C.1 AFFINE INJECTION

Consider a fully parallel update round $u_i = -\eta g_i$, $g_i = G(\theta_i) + \varepsilon_i$. Then

$$\mathcal{D}(u) = \eta\Big(\frac{1}{N}\sum_i \|g_i - \bar{g}\|_2^2\Big)^{1/2}.$$

Write $g_i - \bar{g} = a_i' + b_i$, where

$$a_i' = G(\theta_i) - \frac{1}{N}\sum_j G(\theta_j), \qquad b_i = \varepsilon_i - \bar{\varepsilon}.$$

By Minkowski's inequality,

$$\Big(\frac{1}{N}\sum_i \|a_i' + b_i\|_2^2\Big)^{1/2} \le \Big(\frac{1}{N}\sum_i \|a_i'\|_2^2\Big)^{1/2} + \Big(\frac{1}{N}\sum_i \|b_i\|_2^2\Big)^{1/2}.$$

For the structural term, using Lemma 2 and equation 22:

$$\Big(\frac{1}{N}\sum_i \|a_i'\|_2^2\Big)^{1/2} \le L_{\mathrm{rl}} R(x).$$

For the noise term, using (B4):

$$\Big(\frac{1}{N}\sum_i \|b_i\|_2^2\Big)^{1/2} \le \sigma.$$

Thus the single-event disagreement injection is

$$\mathcal{D}(u) \le \beta_0^{\mathrm{evt}} + \kappa^{\mathrm{evt}} R(x), \qquad \beta_0^{\mathrm{evt}} = \eta\sigma, \quad \kappa^{\mathrm{evt}} = \eta L_{\mathrm{rl}}. \tag{27}$$

This is Eq. equation 5 in the main text.

C.2 WINDOW RECURSION AND LONG-TERM BOUNDS: DIRECT EVENT-LEVEL RECURSION

Let the window $[tB, (t+1)B)$ contain H RL injection events, indexed by occurrence order $r = 0, \dots, H-1$, with injections $\xi_{(r)}$ and pre-injection states $x^{(r)}$. Define

$$A_m = \sum_{r<m} \mathcal{D}(\xi_{(r)}), \qquad A_0 = 0, \qquad S = \max_{0\le m\le H} A_m.$$

By the event-level chain bound equation **??**,

$$R(x^{(r)}) \le R(X_t) + A_r.$$

For each RL injection event, applying equation 27:

$$\mathcal{D}(\xi_{(r)}) \le \beta_0^{\mathrm{evt}} + \kappa^{\mathrm{evt}} R(x^{(r)}) \le \beta_0^{\mathrm{evt}} + \kappa^{\mathrm{evt}}\big(R(X_t) + A_r\big).$$

Let the total additive budget for non-RL implementation-layer events within the window be β^{impl}. Summing over $r < m$,

$$A_m \le m\beta_0^{\mathrm{evt}} + m\kappa^{\mathrm{evt}} R(X_t) + \kappa^{\mathrm{evt}} \sum_{r<m} A_r + \beta^{\mathrm{impl}}.$$

Since $m \le H$ and $A_r \le S$, letting

$$\beta_0 = \eta H\sigma, \qquad \kappa = \eta H L_{\mathrm{rl}},$$

we obtain

$$S \le \beta_0 + \beta^{\mathrm{impl}} + \kappa R(X_t) + \kappa S.$$

If $\kappa < 1$, then

$$\beta_t = A_H \le S \le \frac{\beta_0 + \beta^{\mathrm{impl}} + \kappa R(X_t)}{1-\kappa}. \tag{28}$$

Substituting into the window scalar recursion equation 36,

$$R(X_{t+1}) \le \Big(\gamma + \frac{\kappa}{1-\kappa}\Big) R(X_t) + \frac{\beta_0 + \beta^{\mathrm{impl}}}{1-\kappa}. \tag{29}$$

The strict contraction condition is

$$\gamma + \frac{\kappa}{1-\kappa} < 1 \iff \gamma + \kappa(2-\gamma) < 1.$$

If this condition holds, then

$$\limsup_{t\to\infty} R(X_t) \le \frac{\beta_0 + \beta^{\mathrm{impl}}}{1-\gamma-\kappa(2-\gamma)}. \tag{30}$$

When $H = B$, the equivalent step-size constraint is

$$\eta < \frac{1-\gamma}{BL_{\mathrm{rl}}(2-\gamma)}. \tag{31}$$

Here β^{impl} and β_0 enter the same additive budget but must pass through the $1/(1-\kappa)$ amplification of event-level partial sums; one cannot directly add implementation-layer injections such as staleness errors to the first-order long-term bound numerator while ignoring the κ coupling.

Combining with §B.1,

$$\limsup_{t\to\infty} \frac{1}{N}\sum_i D_{\mathrm{TV}}(\pi_{\theta_i^t}, \pi_{\bar\theta^t}) \le \frac{L_\pi(\beta_0 + \beta^{\mathrm{impl}})}{1-\gamma-\kappa(2-\gamma)}.$$

D Window Contraction and Perturbation Recursion

D.1 Non-Expansiveness of Linear Mixing Compositions

Lemma 5. *Let $W \in \mathbb{R}^{N\times N}$ be non-negative, symmetric, doubly stochastic, and positive semidefinite. Then*

$$\mathbf{J}W = W\mathbf{J} = \mathbf{J}, \qquad WP = PW, \qquad \|WP\|_2 \le 1,$$

and for any u,

$$\mathcal{D}(Wu) \le \mathcal{D}(u). \tag{32}$$

Proof. Doubly stochasticity gives $W\bar J = \bar J W = \bar J$. Positive semidefiniteness with spectrum in $[0,1]$ implies $\|W\|_2 \le 1$, hence $\|WP\|_2 \le 1$. Finally,

$$\mathcal{D}(Wu) = \frac{1}{\sqrt{N}}\|PWu\|_2 = \frac{1}{\sqrt{N}}\|WP(Pu)\|_2 \le \mathcal{D}(u).$$

□

Full averaging $W_{\mathrm{full}} = \bar J$ satisfies $W_{\mathrm{full}}P = 0$. Pairwise mixing

$$W_{ij}(\alpha) = I_N - \frac{\alpha}{2}(a_i - a_j)(a_i - a_j)^\top, \quad \alpha \in (0,1], \tag{33}$$

is non-negative, symmetric, doubly stochastic, and positive semidefinite. Its spectrum is

$$\mathrm{spec}(W_{ij}(\alpha)) = \{1 \text{ (multiplicity } N-1),\ 1-\alpha\} \subset [0,1], \tag{34}$$

because $\alpha \in (0,1]$ implies $1-\alpha \in [0,1)$, but since the spectrum contains eigenvalue 1, one cannot write $\subset [0,1)$. Its action in parameter space is

$$\theta_i^+ = \theta_i - \frac{\alpha}{2}(\theta_i - \theta_j), \qquad \theta_j^+ = \theta_j + \frac{\alpha}{2}(\theta_i - \theta_j), \qquad \theta_k^+ = \theta_k \ (k \ne i,j),$$

corresponding to the logical atom in Eq. equation 13 of the main text.

D.2 Strict Contraction over Fixed Windows

For the window product $\Phi = W_{B-1} \cdots W_0$:

- If some $W_{s_0} = \bar{J}$, then $\Phi = \bar{J}$, so $\|\Phi P\|_2 = 0$.
- If the window contains only identity and pairwise mixing, and the union graph of activated edges $\mathcal{G}_t$ is connected, then $\|\Phi P\|_2 < 1$.

The proof of the second statement is as follows. Suppose, for contradiction, that there exists a zero-mean unit vector y with $\|\Phi y\|_2 = 1$. By the non-expansiveness of each W_s, equality must hold at every step in the chain, so $W_s y = y$ for all W_s in the window. For $W_{ij}(\alpha)$, this is equivalent to $y_i = y_j$. Propagating along the connected graph gives $y = \mathbf{1} \otimes v$; the zero-mean condition gives $v = 0$, a contradiction. The converse follows by constructing constant zero-mean vectors on disconnected components. Thus we may take a window-index-independent

$$\gamma := \max_{\sigma \text{ connected}} \|\Phi_\sigma P\|_2 < 1, \qquad \|\Phi_t P\|_2 \le \gamma. \tag{35}$$

If the underlying graph is connected with maximum degree $d_{\max}$, any consecutive M_f events involve each node initiating at least one interaction with round-robin neighbor coverage, and transaction completion delay is at most Q events, then the connected window length can be taken as

$$B = d_{\max} M_f + Q + 1.$$

D.3 Perturbation Recursion

The event recursion is $x^{s+1} = W_s x^s + \xi_s$. Expanding over the window and left-multiplying by P, using $\|P\Phi_t X_t\|_2 \le \|\Phi_t P\|_2 \|P X_t\|_2$ and equation 32, we obtain

$$R(X_{t+1}) \le \gamma R(X_t) + \beta_t, \qquad \beta_t = \sum_{s=tB}^{tB+B-1} \mathcal{D}(\xi_s). \tag{36}$$

If $\beta_t \le \beta$, then

$$R(X_t) \le \gamma^t R(X_0) + \frac{1-\gamma^t}{1-\gamma}\beta, \qquad \limsup_t R(X_t) \le \frac{\beta}{1-\gamma}. \tag{37}$$

E Centralized Implementation

Before submission, the logical state is $y = \mathbf{1} \otimes \theta_c$. After node i injects p_i, applying full averaging:

$$\mathbf{J}(y + a_i \otimes p_i) = \mathbf{1} \otimes \Big(\theta_c + \frac{p_i}{N}\Big).$$

Thus the coordination map $\mathcal{C}(y) = \mathbf{J}y$ preserves the centroid with zero logical disagreement. By induction over submission order,

$$\theta_c^M = \theta_c^0 + \frac{1}{N}\sum_{s=1}^{M} p^{(s)}.$$

Node i's base point version is $r_i \le M$, local accumulated displacement is q_i, and the physical parameter is $\widehat{\theta}_i^M = \theta_c^{r_i} + q_i$. Let $d_i^M = \widehat{\theta}_i^M - \theta_c^M$. If $\|q_i\|_2 \le G_p$, $\|p^{(s)}\|_2 \le G_p$, and $M - r_i \le B_c$, then

$$\|d_i^M\|_2 \le \varepsilon_c := G_p + \frac{B_c}{N} G_p. \tag{38}$$

By Lemma 2,

$$V(\widehat{\Theta}^M) \le \frac{1}{N}\sum_i \|d_i^M\|_2^2 \le \varepsilon_c^2, \qquad R(\widehat{\Theta}^M) \le \varepsilon_c, \tag{39}$$

which is Eq. equation 12 in the main text. When refreshing the base point, $r_i \leftarrow M$ and $q_i \leftarrow 0$; the in-transit error does not accumulate.

F Decentralized Implementation

F.1 Pairwise Zero-Sum and Variance Decrease

The pairwise zero-sum logical atom is the action of $W_{ij}(\alpha)$, which strictly preserves the centroid. Let $d = \theta_i - \theta_j$. By the two-point identity,

$$V(\Theta^+) - V(\Theta) = -\frac{\alpha(2-\alpha)}{2N}\|\theta_i - \theta_j\|_2^2 \le 0. \tag{40}$$

F.2 Physical-Logical Ledger and Staleness Error

The physical state is denoted $\widehat{x}^s$. Each transaction ℓ has fixed endpoints (i_ℓ, j_ℓ), correction $\delta_\ell = \frac{\alpha_\ell}{2}(s_{j_\ell} - s_{i_\ell})$, applying $+\delta_\ell$ and $-\delta_\ell$ to the two endpoints respectively. Define the logical state z^s: local updates are recorded synchronously; the first half of a transaction taking effect is not recorded; when the second half takes effect, the complete zero-sum correction is recorded at once. Let h^s be the sum of half-corrections for transactions where one half has taken effect but the transaction is incomplete.

Lemma 6 (Physical-logical identity)**.** *For all s,*

$$\widehat{x}^s = z^s + h^s. \tag{41}$$

Proof. Induction over event types: local updates change both $\widehat{x}$ and z simultaneously; the first half changes only $\widehat{x}$ and h; in the second half, $\widehat{x}$ increases by the second half, and the net change in $z + h$ is also the second half. □

Stale snapshot error. Upon transaction completion, let the logical parameters be θ_i, θ_j, snapshots be s_i, s_j, and errors $e_i = \theta_i - s_i$, $e_j = \theta_j - s_j$. Then

$$\delta_{ij} = \frac{\alpha}{2}(\theta_j - \theta_i) + \frac{\alpha}{2}(e_i - e_j).$$

The first term is the non-stale correction; the second term produces a zero-sum displacement $u_i = \frac{\alpha}{2}(e_i - e_j)$, $u_j = -u_i$. Thus

$$\mathcal{D}(u) = \frac{\alpha}{\sqrt{2N}}\|e_i - e_j\|_2 \le \frac{\sqrt{2}\,\alpha E_d}{\sqrt{N}}, \qquad \|e_i\|_2, \|e_j\|_2 \le E_d.$$

That is,

$$\mathcal{D}(u) \le \frac{\sqrt{2}\,\alpha E_d}{\sqrt{N}}. \tag{42}$$

If each local update displacement is $\le G_d$, each transaction half-correction is $\le D_{\max}$, within the interval there are at most L_{loc} local updates and L_{corr} other complete corrections, and at any moment at most M_{p} half-completed transactions, then

$$E_d = L_{\mathrm{loc}} G_d + (L_{\mathrm{corr}} + M_{\mathrm{p}}) D_{\max}. \tag{43}$$

F.3 In-Transit Mass and Final Physical Bound

Suppose at any moment there are at most M_{p} half-effective transactions, with $\|\delta_\ell\|_2 \le D_{\max}$. From equation 20 and the triangle inequality,

$$R(\widehat{x}^s) \le R(z^s) + \eta_{\mathrm{p}}, \qquad \eta_{\mathrm{p}} = M_{\mathrm{p}}\frac{\sqrt{N-1}}{N} D_{\max}. \tag{44}$$

The logical state z satisfies the window recursion with contraction factor γ. Let the number of RL injection events within a window be H, and the total additive budget for non-RL implementation-layer events be β^{stale}, e.g., from stale transactions:

$$\beta^{\mathrm{stale}} \le \sum_{\text{stale transaction completions in window}} \frac{\sqrt{2}\,\alpha_{\max} E_d}{\sqrt{N}}.$$

Let

$$\beta_0 = \eta H \sigma, \qquad \kappa = \eta H L_{\mathrm{rl}}, \qquad \beta^{\mathrm{add}} = \beta_0 + \beta^{\mathrm{stale}}.$$

By the direct event-level recursion from the previous section, the window budget satisfies

$$\beta_t \le \frac{\beta^{\mathrm{add}} + \kappa R(z^{tB})}{1-\kappa}.$$

Thus

$$R(z^{(t+1)B}) \le \Big(\gamma + \frac{\kappa}{1-\kappa}\Big) R(z^{tB}) + \frac{\beta^{\mathrm{add}}}{1-\kappa}. \tag{45}$$

If

$$\gamma + \kappa(2-\gamma) < 1,$$

then

$$\limsup_{t\to\infty} R(\widehat{x}^{tB}) \le \frac{\beta^{\mathrm{add}}}{1-\gamma-\kappa(2-\gamma)} + \eta_{\mathrm{p}}. \tag{46}$$

When $\kappa = 0$, this degenerates to

$$\limsup_{t\to\infty} R(\widehat{x}^{tB}) \le \frac{\beta^{\mathrm{stale}}}{1-\gamma} + \eta_{\mathrm{p}}. \tag{47}$$

Here β^{stale} is not directly added to β_0 as two independent final additive terms; they first enter the unified additive budget β^{add}, then pass through the $1/(1-\kappa)$ amplification of event-level partial sums. This avoids double-counting and coupling omission of stale errors and closed-loop injection.

G Drift under Criterion Violation

Central commit coefficient mismatch. The stable rule is $\theta_c^+ = \theta_c + p_i/N$. If instead $\theta_c^+ = \theta_c + p_i$ is used, each submission absorbs an excess of $(1-\frac{1}{N})p^{(s)}$, with cumulative drift

$$\Delta_c^{(K)} = \Big(1-\frac{1}{N}\Big)\sum_{s=1}^{K} p^{(s)}, \qquad \|\Delta_c^{(K)}\|_2 \le \frac{N-1}{N} K G_p.$$

If there exists a fixed direction w and $c_0 > 0$ such that $\langle p^{(s)}, w\rangle \ge c_0$ for all s, then $\|\Delta_c^{(K)}\|_2 = \Theta(\frac{N-1}{N}K)$, which diverges linearly with training steps. If increments cancel, the actual drift may be smaller; under the stable rule, this quantity is identically zero.

Unidirectional pairwise propagation. The unidirectional update $\theta_j^+ = (1-\alpha)\theta_j + \alpha\theta_i$ changes only node j, so

$$\bar{\theta}^+ - \bar{\theta} = \frac{\alpha}{N}(\theta_i - \theta_j). \tag{48}$$

After K_T activations,

$$\|\bar{\theta}^{K_T} - \bar{\theta}^0\|_2 \le \frac{\alpha K_T}{N}\overline{\Delta}, \qquad \overline{\Delta} = \frac{1}{K_T}\sum_{\ell=1}^{K_T} \|\theta_{i_\ell}^{(\ell^-)} - \theta_{j_\ell}^{(\ell^-)}\|_2.$$

Drift reaches linear order only when persistent edge differences have a non-vanishing component in a fixed direction; without directional consistency, one can only assert that the centroid is no longer conserved per event, not that drift is necessarily unbounded. Under the pairwise zero-sum rule, the centroid is exactly preserved per event, and the physical centroid deviation is controlled by equation 44 with an $\mathcal{O}(1/N)$ bound.

H Experimental Setup and Evaluation Details

This section supplements the hardware environment, resource orchestration, hyperparameter settings, and evaluation metrics omitted from the main text.

H.1 Hardware and Execution Environment

All experiments are conducted on a CPU cluster, with the experimental environment consistent with TianJi's benchmark settings (He et al., 2025). Each compute node is equipped with two Intel Xeon Gold 6240 CPUs (36 physical cores total, 2.60 GHz base frequency) and 384 GB DDR4-2933 memory, interconnected via 100 Gbps Mellanox ConnectX-6 high-speed networking. The software environment includes Ubuntu 20.04 LTS, PyTorch, OpenMPI 4.x, and the asynchronous communication library `hiprlcomm`.

To reduce performance noise from CPU core contention and ensure reliable wall-clock time measurements, all components are bound to exclusive physical cores via `taskset`. For image tasks (Pong, Breakout, and Assault), each Learner uses 4 physical cores, and each Replay Buffer and Actor uses 1 physical core; for RAM-state tasks (Boxing-ram), Learner, Replay Buffer, and Actor each use 1 physical core.

H.2 Cluster Scale and Topology Orchestration

Each consensus group consists of 1 Learner, 1 Replay Buffer, and several Actors, with consensus communication occurring only among the N Learners. The specific resource configurations for single-node baseline scale (Scale-1x) and three-node scaling (Scale-3x) are shown in Table 3.

Table 3: Parallel resource orchestration configurations under different experimental scales.

Task	Nodes	Groups (N)	Actors/Group	Total Actors	CPU Cores
`Boxing-ram-v4`	1	7	3	21	35
	3	21	3	63	105
`PongNoFrameskip-v4`	1	5	2	10	35
	3	15	2	30	105
`BreakoutNoFrameskip-v4`	1	5	2	10	35
	3	15	2	30	105
`AssaultNoFrameskip-v4`	1	3	6	18	33
	3	9	6	54	99

H.3 Evaluation Metrics and Data Processing Pipeline

Evaluation metrics. We adopt the following four metrics: (1) **Time-to-Target (T_{tgt})**: wall-clock time for the training curve to first reach the target return threshold R_{tgt}. Target returns for Boxing, Pong, Breakout, and Assault are 80, 18, 300, and 2500, respectively, with a maximum training budget of 3.2×10^5 steps; (2) **Steps-to-Target (TTT)**: cumulative environment interaction steps across all Actors when first reaching R_{tgt}; (3) **Tail Return** (Tail): average return over the last 10% of the training budget (`TAIL_RATIO`$= 0.1$); (4) **Normalized AUC**: numerical integration of the return curve with respect to cumulative environment steps, normalized by the total step budget. All quantitative results in tables are computed directly from unsmoothed raw logs.

Logging, binning, and smoothing. Each Actor process directly logs episode return, cumulative environment steps, and wall-clock timestamps. Each experimental configuration is run 4 times independently (`RUN_COUNT`$= 4$). Due to asynchronous execution, log points from different runs are not naturally aligned in physical time and cumulative steps. Therefore, we uniformly divide the cumulative environment step or wall-clock time axis into 200 discrete bins (`TIME_BINS`$= 200$) and aggregate returns within each bin to obtain cross-run aligned statistics. Display curves use a sliding average with window size 10 (`SMOOTH_WINDOW`$= 10$); shaded regions represent ± 1 standard deviation across 4 independent runs.

Measurement scope. Although the theoretical analysis is built on parameter-space quantities such as the virtual parameter centroid $\bar{\theta}_t$, experimental evaluation focuses on task returns in policy space and end-to-end training efficiency. In fully asynchronous, barrier-free systems, precisely collecting

parameter snapshots of all Learners at the same logical moment requires additional synchronization or locking mechanisms, which would interfere with the system's natural parallel execution and affect wall-clock measurements. Moreover, parameter variance and training convergence are endogenously coupled. For example, as the algorithm approaches convergence, gradient and parameter update magnitudes naturally decrease, and cross-node parameter variance may decrease accordingly even without improvement in the consensus mechanism. Therefore, static parameter distance is not a reliable proxy for policy performance or consensus quality.

I Dynamic Delay Injection Protocol

This appendix details the generation protocol for the dynamic delay sequences in Section 5.3. For group g in window w, the delay ratio $r_{w,g}$ is obtained by independently multiplying a long-term baseline heterogeneity b_g, short-term random jitter $j_{w,g}$, and sporadic slow-node spikes $s_{w,g}$:

$$r_{w,g} = b_g \cdot j_{w,g} \cdot s_{w,g}, \tag{49}$$

where the components are defined and sampled as follows:

- **Baseline heterogeneity** $b_g \sim \mathcal{U}(1.0, 1.1)$: fixed per group, characterizing inherent hardware/link performance differences;
- **Random jitter** $j_{w,g}$: with probability $p_j = 0.30$, sampled from $\mathcal{U}(1.0, 1.2)$; otherwise 1;
- **Slow-node spikes** $s_{w,g}$: with probability $p_s = 0.20$, sampled from $\mathcal{U}(1.8, 3.0)$; otherwise 1 (when $N = 15$, approximately 3 groups per window are expected to spike).

Table 4: Hyperparameters for dynamic delay generation

Parameter	Value	Parameter	Value	
`total_steps`	320,000	`base_range`	$(1.0, 1.1)$	
`window_size`	100	`jitter_prob/range`	0.30 / $(1.0, 1.2)$	
`num_groups`	15	`slow_prob/range`	0.20 / $(1.8, 3.0)$	
`seeds`	$\{2025, 2026\}$	–	–	–

In the experiments, we independently generate per-group temporal delay ratios under seeds 2025 and 2026, pre-stored for evaluation.

J Supplementary Experiments under Fixed Slow Nodes

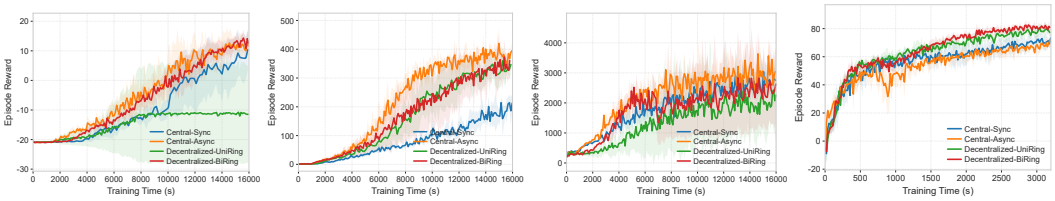
To evaluate the performance of coordination mechanisms under extreme asymmetric delay and near-failure scenarios, we inject a fixed slow-node configuration (i.e., a designated node experiences persistent high-delay suppression). This section reports training curves and quantitative metrics for the four mechanisms on four Atari tasks.

J.1 Experimental Results and Quantitative Analysis

Wall-clock return curves for each method on the four tasks are shown in Figure 4, with detailed quantitative summaries in Table 5.

J.2 Main Observations and Discussion

Topology-induced heterogeneous adaptability differences. In vision-input `Breakout` and `Assault` tasks, the centralized asynchronous mechanism (CA) exhibits clear advantages, achieving $2.373\times$ and $1.988\times$ speedup over CS respectively, with significantly leading tail means. This is because CA employs a hub-and-spoke topology where parameter aggregation of each active worker is completed directly through the center, avoiding cumulative crosstalk at intermediate network nodes caused by a single slow point. In contrast, decentralized methods (DB and DU) are constrained by the ring communication topology, where the long-term presence of a slow node prolongs the global information diffusion cycle and reduces effective communication generations.



(a) `Pong` (b) `Breakout` (c) `Assault` (d) `Boxing`

Figure 4: **Return vs. wall-clock time under fixed slow-node conditions** (mean and ± 1 standard deviation over 4 independent trials).

Table 5: **Summary of performance and efficiency metrics under fixed slow-node configuration**. Time-to-threshold is benchmarked against the wall-clock time for CS to reach CS Peak Return; TailMean and TailStd are computed over the last 10% of steps.

Task	Method	CS Peak Target	Time (s) ↓	Speedup (vs. CS) ↑	TailMean ± TailStd ↑	AUC ↑
`Pong`	Central-Sync (CS)	10.43	15917.15	1.000×	7.08 ± 9.19	-9.62
	Central-Async (CA)	10.43	13112.17	1.214×	10.69 ± 8.06	**-4.37**
	Decentral-UniRing (DU)	10.43	N/A	N/A	-11.21 ± 16.89	-14.64
	Decentral-BiRing (DB)	10.43	13805.53	1.153×	**11.66 ± 4.23**	-6.24
`Breakout`	Central-Sync (CS)	214.31	15769.98	1.000×	193.54 ± 71.90	80.72
	Central-Async (CA)	214.31	**6646.67**	**2.373**×	**370.47 ± 59.24**	**220.20**
	Decentral-UniRing (DU)	214.31	9069.57	1.739×	331.44 ± 54.25	161.51
	Decentral-BiRing (DB)	214.31	8859.45	1.780×	338.36 ± 69.65	166.55
`Assault`	Central-Sync (CS)	3038.72	15037.33	1.000×	2521.17 ± 669.71	1784.30
	Central-Async (CA)	3038.72	**7564.29**	**1.988**×	**3148.11 ± 979.70**	**2151.69**
	Decentral-UniRing (DU)	3038.72	N/A	N/A	1856.00 ± 1007.09	1287.91
	Decentral-BiRing (DB)	3038.72	N/A	N/A	2507.00 ± 1449.78	1724.80
`Boxing`	Central-Sync (CS)	73.23	3197.31	1.000×	71.69 ± 7.98	57.34
	Central-Async (CA)	73.23	N/A	N/A	68.85 ± 7.05	54.03
	Decentral-UniRing (DU)	73.23	2046.70	1.562×	77.36 ± 6.26	62.46
	Decentral-BiRing (DB)	73.23	**1793.36**	**1.783**×	**80.75 ± 5.24**	**63.66**

Protective effect of the stable coordination criteria under strong asymmetric perturbations. When fixed slow nodes introduce extremely asymmetric update frequencies, DU, which violates the stable coordination criteria, exposes severe centroid drift problems. The most typical evidence is the `Pong` task: due to asymmetric bias continuously accumulating from unidirectional contraction, DU ultimately falls into a suboptimal plateau (tail mean of only -11.21, failing to reach the threshold within the time limit). In contrast, DB, which strictly satisfies the stable coordination criteria, maintains the symmetric zero-sum contraction property, achieving a tail return of 11.66 ± 4.23 and final performance comparable to CA.

Dilution effect of node scale and input representation on sensitivity. In the `Boxing` task, decentralized methods do not degrade due to fixed slow nodes; DB actually achieves the best speedup ($1.783\times$) and tail return (80.75), while DU also achieves second-best performance. This is primarily because the task configures more learners ($N = 21$), significantly diluting the proportion of a single slow node in the overall topology. Combined with the more stable policy exploration and gradient variance under RAM state representation, the negative impact of local ring communication delay on centroid update stability is relatively limited.