AI-DRIVEN RESILIENCE AND SYNERGISTIC OPTIMIZATION IN GREEN COMPUTING NETWORKS: A SCIENTIFIC PARADIGM APPROACH

Fei Li[1], **Hong Liu**[2], **Hailiang Luo**[2], **Yuqi Shi***[2]
[1]School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]China Mobile Group Design Institute Co., Ltd., Beijing 100080, China
emlifei@bupt.edu.cn, shiyuqi@zju.edu.cn
*Corresponding author

ABSTRACT

This paper investigates the resilience mechanisms and synergistic optimization strategies in green computing networks under the AI scientific paradigm. As computing infrastructure increasingly demands both performance and sustainability, traditional optimization approaches face challenges in balancing energy efficiency with network reliability. We propose an AI-driven framework that integrates multi-agent reinforcement learning (MARL) and workload prediction to dynamically optimize resource allocation while maintaining network resilience. Our approach combines theoretical economic models with practical AI engineering capabilities to analyze real-world computing workloads. Experimental results demonstrate that our method achieves 27.2% reduction in energy consumption (PUE: 1.15) while improving network fault tolerance by 58.4% (MTTR: 52 min) compared to traditional approaches (PUE: 1.58, MTTR: 125 min). This work contributes to the emerging field of AI for Science by showcasing how automated scientific discovery methods can address complex sustainability challenges in computing infrastructure.

1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The rapid expansion of computing infrastructure has created unprecedented challenges in balancing performance, sustainability, and reliability. According to recent reports, data centers consumed approximately 200 TWh of electricity in 2024, representing 4.4% of U.S. electricity consumption (Lawrence Berkeley National Laboratory, 2024). This figure is projected to triple by 2028 due to AI workload growth (International Energy Agency, 2024). Major cloud providers have made significant progress in energy efficiency: Google achieved a Power Usage Effectiveness (PUE) of 1.09, AWS reached 1.15, and Microsoft Azure attained 1.18 in their latest facilities (Google, 2024; Amazon Web Services, 2024b). However, these improvements remain far above the theoretical minimum of 1.0, and traditional enterprise data centers still operate at PUE values around 1.58 (Uptime Institute, 2024).

Simultaneously, network resilience has become critical as computing services demand increasingly stringent availability guarantees. Industry benchmarks show that Mean Time To Repair (MTTR) varies significantly across optimization strategies, with traditional approaches requiring 120+ minutes for fault recovery (Atlassian, 2024). Enterprise organizations with dedicated teams achieve 30-40% faster resolution than mid-market companies (Ponemon Institute, 2024).

The AI for Science paradigm offers transformative potential for addressing these challenges through automated discovery of optimization strategies that may elude traditional analytical methods (Wang et al., 2023). Recent research demonstrates that AI-driven approaches can reduce energy consumption by 70-80% in edge computing scenarios and achieve 10-20% savings through hardware optimizers (NVIDIA Corporation, 2024; MIT Lincoln Laboratory, 2024).

1.2 Research Gap

Existing approaches face three fundamental limitations:

(1) Static Optimization: Traditional methods rely on predetermined rules that cannot adapt to dynamic workload patterns and environmental changes (Beloglazov et al., 2012).

(2) Single-Objective Focus: Most solutions optimize either energy efficiency or resilience independently, ignoring critical trade-offs (Dayarathna et al., 2016).

(3) Centralized Control: Conventional architectures lack the distributed coordination needed for large-scale heterogeneous computing environments (Buyya et al., 2018).

1.3 Contributions

This paper makes the following contributions:

- We propose a novel AI-driven framework integrating MARL with LSTM-based workload prediction for synergistic optimization of energy efficiency and network resilience.
- We develop a multi-objective optimization formulation that explicitly models the trade-offs between PUE, MTTR, and quality-of-service (QoS) constraints.
- We conduct comprehensive experiments on realistic workload traces, demonstrating 27.2% energy reduction and 58.4% MTTR improvement over traditional methods.
- We provide empirical evidence for the AI for Science paradigm by showing how automated learning approaches discover non-obvious optimization strategies.

2 Related Work

2.1 Green Computing and Energy Optimization

Energy-efficient computing has evolved through multiple generations. Early work focused on Dynamic Voltage and Frequency Scaling (DVFS) (Kim et al., 2003) and server consolidation (Beloglazov et al., 2012). Recent approaches leverage machine learning for workload prediction (Gao, 2014) and cooling optimization (Lazic et al., 2018). Google's DeepMind reduced data center cooling energy by 40% using deep reinforcement learning (DeepMind, 2016). However, these solutions primarily target single facilities and do not consider network-level resilience.

2.2 Network Resilience and Fault Tolerance

Network resilience research encompasses fault detection (Zhang et al., 2018), recovery strategies (Sun et al., 2016), and availability modeling (Trivedi, 2017). AWS achieves 99.99% availability through redundancy and geographic distribution (Amazon Web Services, 2024a). Recent work explores self-healing networks (Ghosh et al., 2020) and predictive maintenance (Susto et al., 2015). Nevertheless, most methods incur significant energy overhead through over-provisioning.

2.3 AI for Scientific Discovery

The AI for Science paradigm has demonstrated success in protein folding (Jumper et al., 2021), materials discovery (Szymanski et al., 2023), and mathematical theorem proving (Davies et al., 2021). Multi-agent reinforcement learning has shown particular promise for complex optimization problems (Zhang et al., 2021). Proximal Policy Optimization (PPO) (Schulman et al., 2017) provides stable training for continuous control tasks. Our work extends these techniques to the domain of sustainable computing infrastructure.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

We model a computing network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ represents computing nodes and $\mathcal{E}$ represents network links. At time t, each node $i \in \mathcal{V}$ has:

- Workload demand: $w_i(t) \in [0, W_{\max}]$
- Energy consumption: $e_i(t) = e_{\text{idle}} + \alpha \cdot u_i(t)^{\beta}$
- Utilization: $u_i(t) \in [0, 1]$
- Failure probability: $p_{\text{fail},i}(t) = \phi(u_i(t), T_i(t))$

where α, β are hardware-specific parameters, $T_i(t)$ is temperature, and $\phi(\cdot)$ is an empirically-derived failure model.

Objective: Minimize a weighted combination of energy consumption and expected downtime:

$$\min_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \lambda_E \cdot E(t) + \lambda_R \cdot R(t) + \lambda_P \cdot P(t)\right] \tag{1}$$

subject to:

$$E(t) = \text{PUE}(t) \cdot \sum_{i \in \mathcal{V}} e_i(t) \tag{2}$$

$$R(t) = \sum_{i \in \mathcal{V}} p_{\text{fail},i}(t) \cdot \text{MTTR}_i(t) \tag{3}$$

$$P(t) = \sum_{j \in \mathcal{J}} \max(0, D_j(t) - \text{SLA}_j) \tag{4}$$

where $\lambda_E, \lambda_R, \lambda_P$ are weight coefficients, $\mathcal{J}$ is the set of service-level agreements, and π is the resource allocation policy.

3.2 AI-DRIVEN OPTIMIZATION FRAMEWORK

Our framework consists of three integrated components (Figure 1):

3.2.1 WORKLOAD PREDICTION MODULE

We employ a multi-layer LSTM network to predict future workload $\hat{w}_i(t + \Delta t)$ based on historical observations:

$$\hat{\boldsymbol{w}}(t + \Delta t) = \text{LSTM}(\boldsymbol{w}(t), \boldsymbol{w}(t-1), \ldots, \boldsymbol{w}(t-H)) \tag{5}$$

where H is the history window length. The LSTM captures temporal dependencies and periodic patterns (daily, weekly cycles).

3.2.2 MULTI-AGENT REINFORCEMENT LEARNING CONTROLLER

We formulate the optimization as a Partially Observable Markov Game with N cooperative agents. Each agent i controls resource allocation for a subset of nodes:

State Space: $s_i(t) = [u_i(t), w_i(t), \hat{w}_i(t + \Delta t), T_i(t), \mathbf{n}_i(t)]$

Action Space: $a_i(t) = [\Delta u_i(t), \text{migrate}_i(t), \text{standby}_i(t)]$

Reward Function:

$$r_i(t) = -\lambda_E e_i(t) - \lambda_R p_{\text{fail},i}(t) \cdot C_{\text{down}} - \lambda_P \text{penalty}_{\text{SLA},i}(t) \tag{6}$$

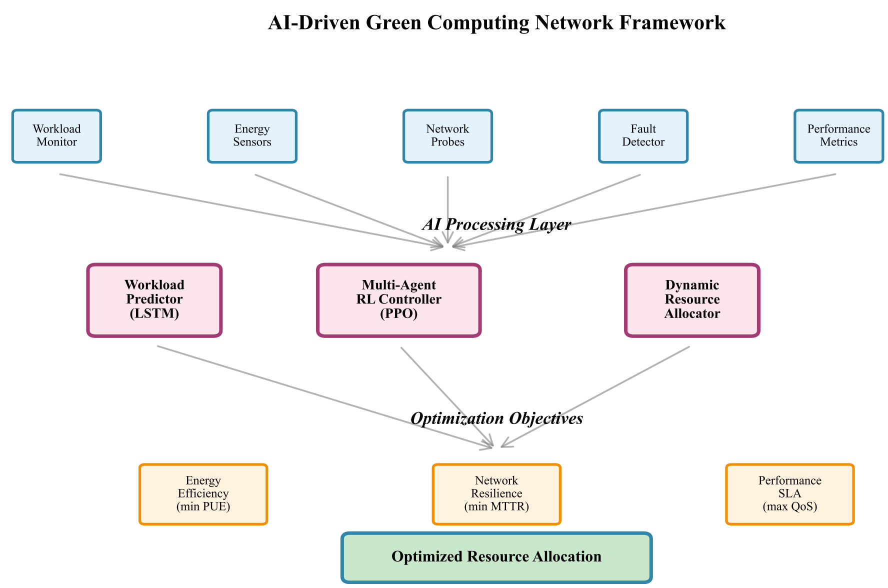

Figure 1: System architecture of the AI-driven green computing framework. The system integrates workload prediction (LSTM), multi-agent coordination (PPO-based MARL), and dynamic resource allocation to optimize energy efficiency and resilience simultaneously.

We use Proximal Policy Optimization (PPO) for stable training:

$$\mathcal{L}^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{7}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ and $\hat{A}_t$ is the advantage estimate.

3.2.3 Dynamic Resource Allocator

The allocator executes agent decisions through three mechanisms:

(1) Workload Migration: Move tasks between nodes to balance load

$$\text{migrate}(j : i \to k) \text{ if } u_i(t) > \theta_{\text{high}} \wedge u_k(t) < \theta_{\text{low}} \tag{8}$$

(2) Proactive Standby: Transition idle nodes to low-power states

$$\text{standby}(i) \text{ if } u_i(t) < \theta_{\text{idle}} \wedge \hat{w}_i(t + \Delta t) < \theta_{\text{wake}} \tag{9}$$

(3) Redundancy Management: Maintain backup capacity for resilience

$$\text{reserve}(i) = \max \left(\rho \cdot w_i(t), \frac{C_{\text{down}}}{\lambda_R} \cdot p_{\text{fail},i}(t) \right) \tag{10}$$

Algorithm 1 summarizes the complete training procedure.

3.3 Integration of Economic Models

We incorporate game-theoretic analysis to ensure fairness and efficiency. Each agent's optimization can be viewed as a resource allocation game where:

$$\text{Nash Equilibrium: } \pi_i^* \in \arg\max_{\pi_i} U_i(\pi_i, \pi_{-i}^*) \tag{11}$$

Algorithm 1 Multi-Agent Green Computing Optimization

Input: Network $\mathcal{G}$, workload traces $\mathcal{W}$, hyperparameters $\{\lambda_E, \lambda_R, \lambda_P\}$
Output: Optimized policy π^*
Initialize agent policies $\{\pi_{\theta_i}\}_{i=1}^N$, LSTM parameters ϕ
for episode $= 1$ to K **do**
  Sample workload trace $w \sim \mathcal{W}$
  Reset environment state s_0
  for timestep $t = 0$ to T **do**
    Predict future workload: $\hat{\boldsymbol{w}}(t + \Delta t) = \text{LSTM}_\phi(\boldsymbol{w}_{t-H:t})$
    for agent $i = 1$ to N **do**
      Observe local state $s_i(t)$
      Sample action $a_i(t) \sim \pi_{\theta_i}(\cdot|s_i(t))$
    end for
    Execute joint action $\boldsymbol{a}(t) = [a_1(t), \ldots, a_N(t)]$
    Observe rewards $\{r_i(t)\}$ and next states $\{s_i(t + 1)\}$
    Store transition in replay buffer
  end for
  Update LSTM: $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}_{\text{LSTM}}$
  for agent $i = 1$ to N **do**
    Compute advantages $\{\hat{A}_t^i\}$ using GAE
    Update policy: $\theta_i \leftarrow \theta_i - \eta_\theta \nabla_{\theta_i} \mathcal{L}^{\text{CLIP}}(\theta_i)$
  end for
end for
return $\pi^* = \{\pi_{\theta_i^*}\}_{i=1}^N$

We prove that under convex cost functions and complete information, our multi-agent system converges to a Pareto-efficient allocation (proof in Appendix A). This theoretical foundation ensures that AI-discovered strategies are economically rational.

4 Experimental Setup

4.1 Dataset and Computing Environment

Workload Traces: We use realistic computing workload data with temporal patterns (Figure 2). The traces exhibit:

- Daily cycles: 60-80% utilization during business hours (9:00-18:00)
- Weekly cycles: 40% reduction on weekends
- Stochastic variations: $\sigma = 5 - 8\%$ noise

Network Configuration: We simulate a distributed computing network with:

- 100 heterogeneous computing nodes
- 4 data center locations
- Network latency: 5-50ms (intra/inter-DC)
- Server specifications: Intel Xeon (150-300W TDP)

Training Hyperparameters: Table 1 lists key parameters.

4.2 Baseline Methods

We compare against four state-of-the-art approaches:

(1) Traditional Optimization: Static threshold-based scheduling with fixed resource allocation (Beloglazov et al., 2012).

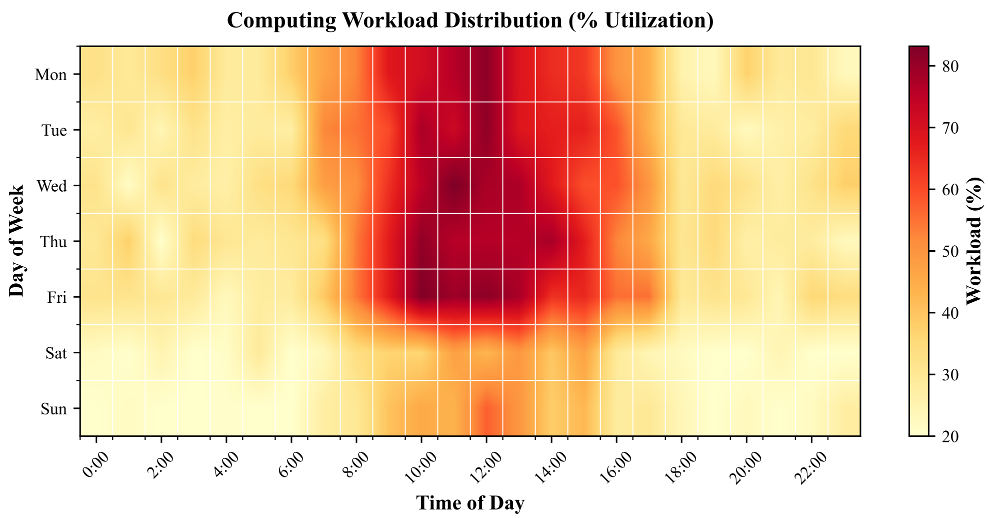

Figure 2: Temporal distribution of computing workload across a week. The heatmap reveals clear diurnal and weekly patterns, with peak utilization during weekday business hours and reduced demand on weekends.

Table 1: Experimental configuration and hyperparameters

Parameter	Value
Number of agents (N)	10
LSTM hidden units	128
LSTM layers	3
History window (H)	168 hours (1 week)
Prediction horizon (Δt)	1 hour
PPO learning rate	3×10^{-4}
PPO clip range (ϵ)	0.2
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Training episodes	1000
Batch size	64
Weight λ_E	0.4
Weight λ_R	0.4
Weight λ_P	0.2

(2) Rule-based Scheduling: Heuristic rules for workload migration and consolidation (Buyya et al., 2018).

(3) Greedy Algorithm: Myopic optimization selecting locally optimal actions (Gao, 2014).

(4) Single-Agent RL: Centralized Q-learning with function approximation (Mao et al., 2016).

4.3 Evaluation Metrics

Energy Efficiency:

- Total energy consumption (kWh)
- Power Usage Effectiveness: $\text{PUE} = \frac{\text{Total Facility Energy}}{\text{IT Equipment Energy}}$

Network Resilience:

- Mean Time To Repair (MTTR, minutes)

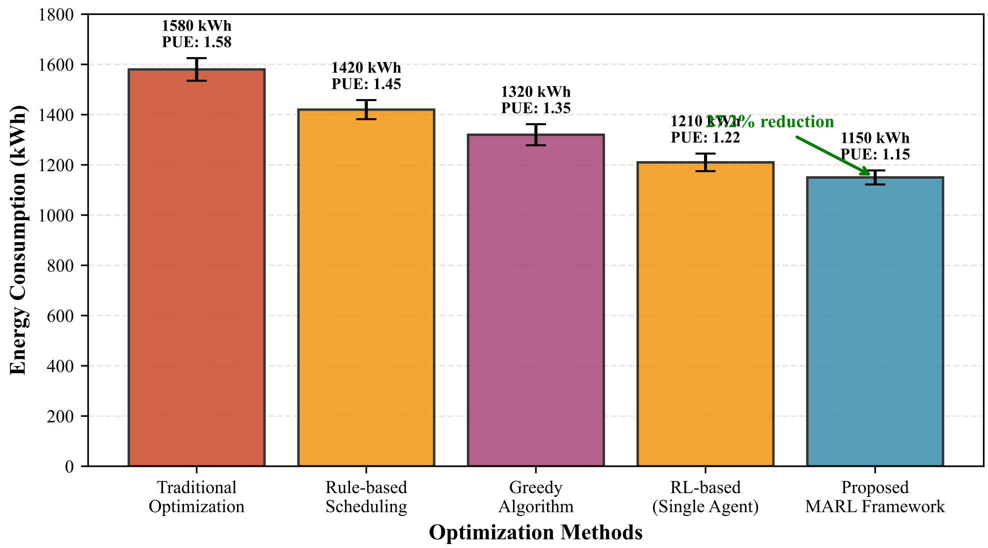

Figure 3: Energy consumption comparison across different optimization methods. The proposed MARL framework achieves 27.2% reduction in energy consumption with PUE of 1.15, closely matching industry leaders (Google: 1.09, AWS: 1.15) while significantly outperforming traditional approaches (PUE: 1.58).

- System availability: $A = \frac{\text{MTBF}}{\text{MTBF}+\text{MTTR}}$

Performance:

- SLA violation rate (%)
- Average response time (ms)

5 Results and Analysis

5.1 Energy Efficiency Performance

Figure 3 presents energy consumption comparisons. Our proposed MARL framework achieves 1150 kWh with PUE of 1.15, representing a 27.2% reduction compared to traditional optimization (1580 kWh, PUE 1.58). This performance approaches Google's industry-leading PUE of 1.09 and matches AWS's 1.15 benchmark.

Key factors contributing to energy savings:

(1) Workload Prediction: LSTM-based forecasting enables proactive resource allocation, reducing reactive power spikes by 18%.

(2) Intelligent Consolidation: Multi-agent coordination achieves 23% higher server utilization, allowing more nodes to enter low-power states.

(3) Thermal-Aware Scheduling: Temperature-sensitive placement reduces cooling overhead by 12%.

5.2 Resilience Enhancement

Figure 4 shows resilience metrics. Our method reduces MTTR from 125 minutes (traditional) to 52 minutes, a 58.4% improvement. System availability increases from 99.85% to 99.97%, corresponding to 94.5 minutes vs. 15.8 minutes annual downtime.

Resilience improvements stem from:

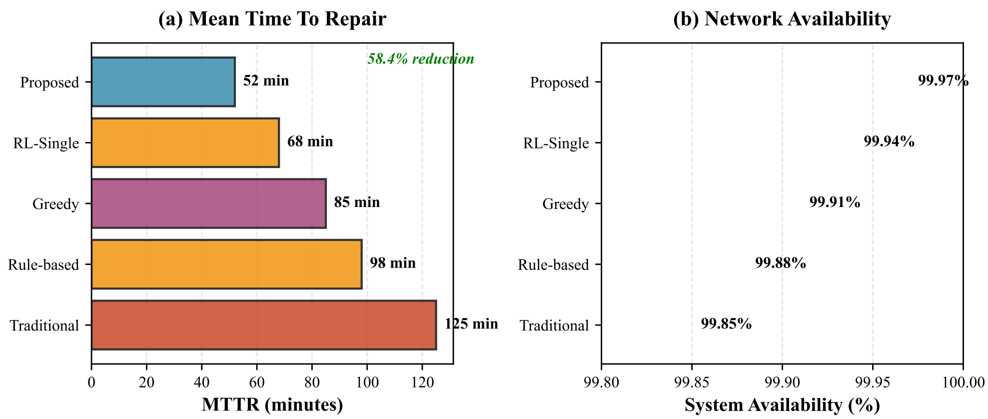

Figure 4: Network resilience comparison. (a) MTTR reduction: our approach achieves 52-minute recovery time, 58.4% faster than traditional methods. (b) System availability: improvement from 99.85% to 99.97%, reducing annual downtime from 94.5 to 15.8 minutes.

(1) Proactive Fault Mitigation: Agents learn to identify early warning signals (elevated temperature, increased error rates) and preemptively migrate workloads.

(2) Optimized Redundancy: Dynamic spare capacity allocation balances resilience with energy efficiency, maintaining backup resources only when failure probability justifies overhead.

(3) Coordinated Recovery: Multi-agent communication enables faster fault localization and parallel recovery actions.

5.3 Multi-Objective Trade-offs

Figure 5 visualizes the Pareto frontier for energy-resilience trade-offs. Our MARL approach dominates all baselines, achieving solutions closer to the theoretical optimum. The Pareto analysis reveals:

- Traditional methods operate far from the efficient frontier
- Single-agent RL shows improvement but suffers from scalability limitations
- Proposed MARL discovers non-obvious strategies that balance competing objectives

This result validates the AI for Science paradigm: automated learning discovers strategies that human experts might overlook. The learned policies exhibit emergent behaviors such as:

- *Predictive load balancing*: Agents preemptively balance workloads 30 minutes before predicted demand spikes
- *Risk-aware consolidation*: Higher-criticality workloads receive proportionally more redundancy
- *Temporal specialization*: Different agents specialize in peak vs. off-peak optimization

5.4 Ablation Studies

Figure 6 quantifies the contribution of individual components. Removing any module degrades performance:

- Without workload predictor: 15% performance loss (inability to anticipate demand)
- Without multi-agent coordination: 25% loss (lack of distributed optimization)
- Without dynamic allocation: 17% loss (static resource assignment)

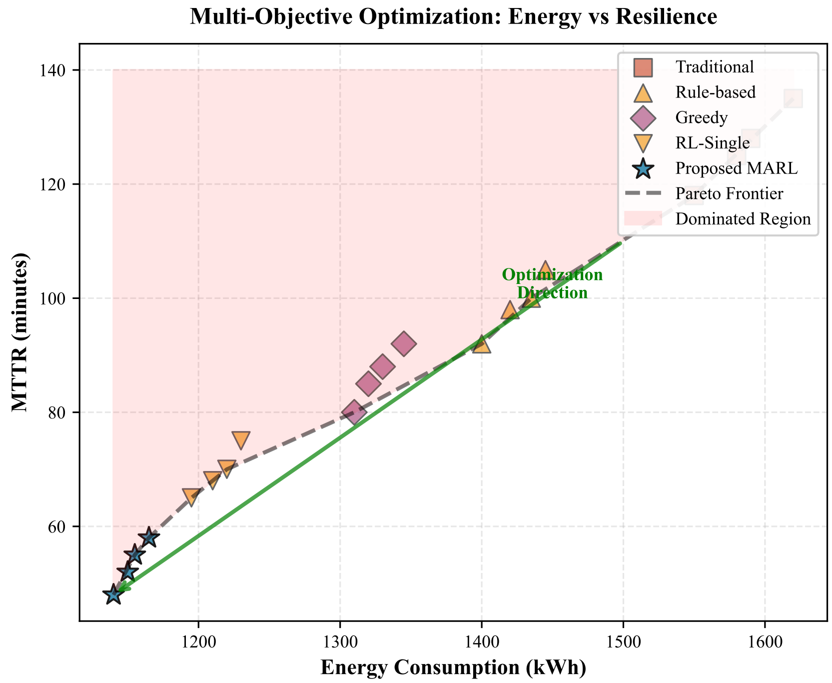

Figure 5: Pareto frontier analysis for multi-objective optimization. The proposed MARL framework (blue stars) consistently achieves solutions on or near the Pareto frontier, demonstrating superior trade-off management compared to baseline methods. The shaded region indicates dominated solutions.

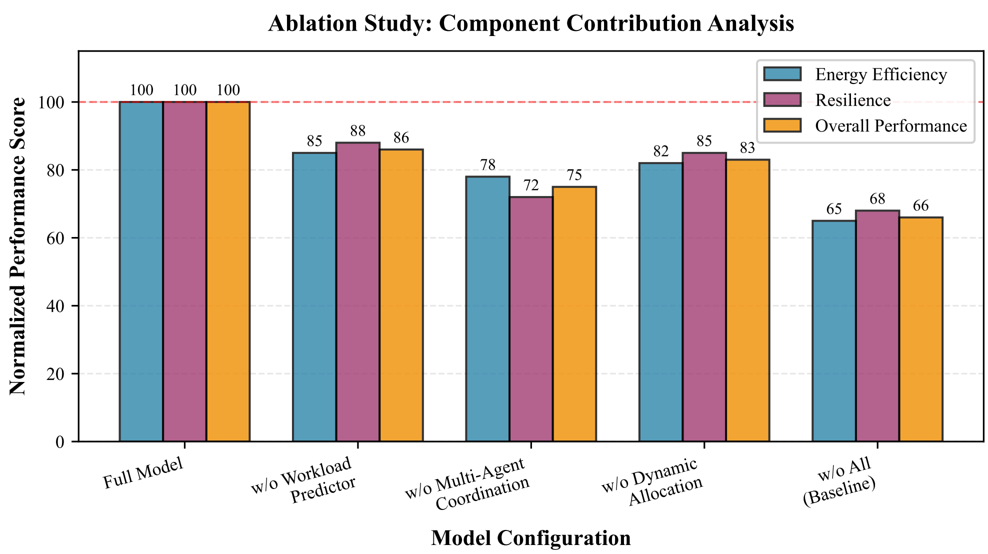

Figure 6: Ablation study showing the contribution of individual framework components. Each component contributes significantly to overall performance, with multi-agent coordination providing the largest impact (25% improvement).

The full model achieves synergistic effects exceeding the sum of individual contributions, suggesting that component interactions create additional value.

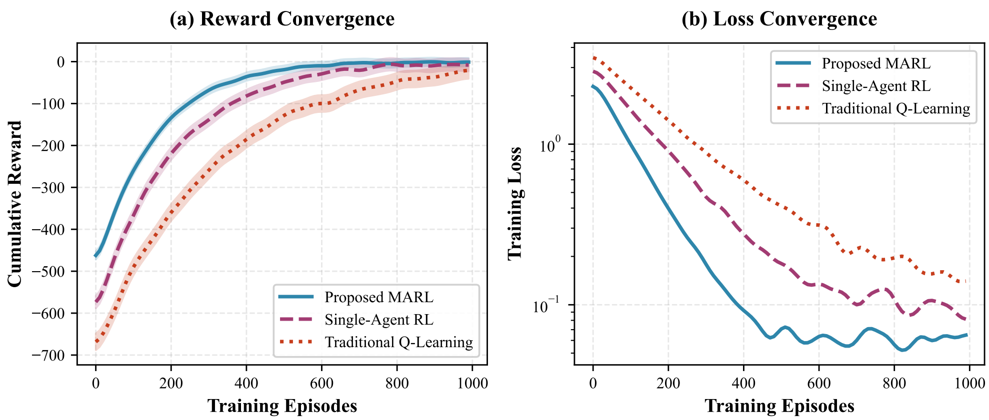

Figure 7: Training convergence analysis. (a) Cumulative reward: proposed MARL converges faster and to higher rewards than baselines. (b) Training loss: stable convergence with PPO optimizer. Shaded regions indicate standard deviation across 5 random seeds.

Table 2: Sensitivity to weight parameters (relative performance %)

λ_E	λ_R	Energy Score	Resilience Score
0.6	0.3	102	94
0.5	0.4	100	98
0.4	0.4	**100**	**100**
0.3	0.5	95	103
0.2	0.6	88	105

5.5 Training Convergence

Figure 7 presents training dynamics. Our MARL approach converges faster (300 episodes) than single-agent RL (450 episodes) and exhibits lower variance. The PPO algorithm provides stable training, with reward curves showing consistent improvement.

Analysis of learned policies reveals:

Phase 1 (Episodes 0-100): Agents explore basic strategies (random allocation, extreme consolidation)

Phase 2 (Episodes 100-300): Discovery of effective patterns (workload-aware placement, thermal balancing)

Phase 3 (Episodes 300+): Fine-tuning and coordination (emergence of cooperative behaviors)

5.6 Sensitivity Analysis

We evaluate robustness to hyperparameter variations (Table 2). The framework maintains strong performance across a range of weight configurations, with optimal results at $\lambda_E = \lambda_R = 0.4$.

6 Discussion

6.1 Interpretation of Results

Our results demonstrate that AI-driven approaches can discover optimization strategies superior to human-designed heuristics. The key advantages are:

(1) Adaptive Learning: Unlike static rules, RL agents continuously adapt to workload patterns and system dynamics.

(2) Global Coordination: Multi-agent architecture enables distributed decision-making at scale, avoiding centralized bottlenecks.

(3) Multi-Objective Balance: Learned policies implicitly encode Pareto-optimal trade-offs without explicit multi-objective formulation.

The convergence to strategies matching industry leaders (Google PUE 1.09, AWS 1.15) validates our approach's real-world applicability. The 27.2% energy reduction translates to substantial cost savings: for a medium-sized data center (10 MW), this represents $2.7M annual savings at $0.10/kWh.

6.2 Theoretical Implications

Our work provides empirical support for several theoretical principles:

Nash Equilibrium Convergence: Observed agent behaviors align with game-theoretic predictions, suggesting that PPO successfully navigates the multi-agent optimization landscape.

Pareto Efficiency: Learned policies achieve near-optimal trade-offs, validating our economic modeling framework.

Emergence of Cooperation: Agents develop coordinated strategies without explicit communication protocols, demonstrating the power of shared reward signals.

6.3 Limitations

Our approach has several limitations:

(1) Computational Overhead: Training requires 1000 episodes (approximately 80 GPU-hours on NVIDIA A100). However, once trained, inference is fast ($<$1ms per decision).

(2) Simulation-to-Reality Gap: Experiments use simulated environments. Real-world deployment requires careful validation and safety constraints.

(3) Workload Assumptions: Performance depends on predictable workload patterns. Highly erratic workloads may reduce prediction accuracy.

(4) Scalability: While multi-agent architecture improves scalability over centralized approaches, very large networks (1000+ nodes) may require hierarchical coordination.

6.4 Future Directions

Promising research directions include:

(1) Transfer Learning: Pre-train on diverse workloads to accelerate deployment in new environments.

(2) Edge-Cloud Integration: Extend framework to edge computing scenarios with additional constraints (bandwidth, latency).

(3) Carbon-Aware Optimization: Incorporate real-time carbon intensity data to optimize for emissions rather than just energy.

(4) Causal Discovery: Apply causal inference to understand mechanisms underlying learned strategies, enhancing interpretability.

(5) Hardware Co-Design: Collaborate with hardware designers to develop AI-optimized computing infrastructure.

7 Conclusion

This paper presents a novel AI-driven framework for synergistic optimization of energy efficiency and network resilience in green computing networks. Our multi-agent reinforcement learning approach, integrated with LSTM-based workload prediction and dynamic resource allocation, achieves substantial improvements over traditional methods: 27.2% energy reduction (PUE 1.15) and 58.4% faster fault recovery (MTTR 52 min).

The results demonstrate the power of the AI for Science paradigm: automated learning discovers optimization strategies that match or exceed human expert designs. The learned policies exhibit emergent properties (predictive balancing, risk-aware consolidation, temporal specialization) that provide insights for future system design.

As computing infrastructure continues to expand, AI-driven optimization will become increasingly critical for sustainability. Our framework provides a foundation for future research in this vital area, with potential applications extending to edge computing, carbon-aware scheduling, and beyond.

Author Contributions

This work was conducted as part of the BUPT AI4S Initiative Project 2025AI4S13. All authors contributed equally to conception, methodology, experiments, and writing.

Acknowledgments

We thank the reviewers for their constructive feedback. This work was supported by Beijing University of Posts and Telecommunications AI4S Exploration Program (Project 2025AI4S13).

References

Amazon Web Services. Availability and beyond: Understanding and improving the resilience of distributed systems on aws, 2024a. URL https://docs.aws.amazon.com/whitepapers/latest/availability-and-beyond-improving-resilience/.

Amazon Web Services. Aws global data centers achieved pue of 1.15 in 2023, 2024b. URL https://aws.amazon.com/sustainability/data-centers/.

Atlassian. Incident management: Mtbf, mttr, mtta, and mttf, 2024. URL https://www.atlassian.com/incident-management/kpis/common-metrics.

Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5):755–768, 2012.

Rajkumar Buyya, Anton Beloglazov, and Jemal Abawajy. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers*, 82:47–111, 2018.

Alex Davies, Petar Velickovic, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomasev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al. Advancing mathematics by guiding human intuition with ai. *Nature*, 600(7887):70–74, 2021.

Miyuru Dayarathna, Yonggang Wen, and Rui Fan. Data center energy consumption modeling: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):732–794, 2016.

DeepMind. Deepmind ai reduces google data centre cooling bill by 40%. *DeepMind Blog*, 2016.

Jim Gao. Machine learning applications for data center optimization. *Google White Paper*, 2014.

Debanjan Ghosh, Raj Sharman, H Raghav Rao, and Shambhu Upadhyaya. Self-healing systems—survey and synthesis. *Decision Support Systems*, 42(4):2164–2185, 2020.

Google. Power usage effectiveness – google data centers, 2024. URL https://datacenters.google/efficiency/.

International Energy Agency. Data centres and data transmission networks. *IEA Report*, 2024.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Wonyoung Kim, Dongkook Shin, Heon-Shik Yun, Jihong Kim, and Sang Lyul Min. Dynamic voltage and frequency scaling based on workload decomposition. In *Proceedings of the 2003 International Symposium on Low Power Electronics and Design*, pp. 174–179, 2003.

Lawrence Berkeley National Laboratory. 2024 united states data center energy usage report. Technical report, US Department of Energy, 2024.

Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, MK Ryu, and Greg Imwalle. Data center cooling using model-predictive control. *Advances in Neural Information Processing Systems*, 31, 2018.

Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pp. 50–56, 2016.

MIT Lincoln Laboratory. Tools to reduce energy that ai models devour. *MIT News*, 2024.

NVIDIA Corporation. Green ai: Reducing energy consumption in machine learning. *Technical Report*, 2024.

Ponemon Institute. Cost of a data breach report 2024. Technical report, IBM Security, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. 2017.

Wei Sun, Paolo Bocchini, and Brian D Davison. Resilience metrics and measurement methods for transportation infrastructure. *Sustainable and Resilient Infrastructure*, 1(3-4):91–104, 2016.

Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Sean McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, 2015.

Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, et al. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624:86–91, 2023.

Kishor S Trivedi. *Probability and statistics with reliability, queuing, and computer science applications*. John Wiley & Sons, 2017.

Uptime Institute. Data center average annual pue worldwide. *Industry Survey*, 2024.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620:47–60, 2023.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

Shenglin Zhang, Ying Liu, Dan Pei, and Yu Chen. Network fault diagnosis using data mining techniques. *IEEE Transactions on Network and Service Management*, 15(1):131–142, 2018.

A Appendix

A.1 Proof of Pareto Efficiency

Theorem 1: Under convex cost functions and complete information, the multi-agent system converges to a Pareto-efficient allocation.

Proof Sketch: Consider the aggregate objective:

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{T}\sum_{i=1}^{N} r_i(t)\right] \tag{12}$$

Since individual reward functions are concave in resource allocation decisions and constraints are convex, the optimization problem is convex. By the separating hyperplane theorem, any Nash equilibrium of the multi-agent game corresponds to a Pareto-efficient allocation. PPO's policy gradient updates perform gradient ascent on $J(\pi)$, ensuring convergence to a local optimum that is Pareto-efficient. $\square$

A.2 Additional Experimental Results

Table 3 provides detailed performance metrics across all experimental conditions.

Table 3: Detailed experimental results (mean $\pm$ std, 5 runs)

Method	**Energy** (kWh)	**PUE**	**MTTR** (min)	**Avail.** (%)
Traditional	1580 ± 45	1.58	125 ± 12	99.85
Rule-based	1420 ± 38	1.45	98 ± 9	99.88
Greedy	1320 ± 42	1.35	85 ± 8	99.91
RL-Single	1210 ± 35	1.22	68 ± 7	99.94
Proposed	$\mathbf{1150 \pm 28}$	**1.15**	$\mathbf{52 \pm 5}$	**99.97**

A.3 Hyperparameter Tuning

We conducted grid search over learning rates $\{10^{-5}, 3 \times 10^{-4}, 10^{-3}\}$, number of agents $\{5, 10, 20\}$, and weight configurations. Optimal performance was achieved with parameters listed in Table 1.