International Journal of Science, Strategic Management and Technology
Volume **02** Issue **04** April-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Scalability of Deep Reinforcement Learning in High-Dimensional State Spaces

Dr. Ajay Singh Thakur
Govt. Kaktiya P.G. College Jagdalpur, Bastar, C.G.

https://doi.org/10.55041/ijsmt.v2i4.293

Cite this Article: Thakur, A. S. (2026). Scalability of Deep Reinforcement Learning in High-Dimensional State Spaces. International Journal of Science, Strategic Management and Technology, *02*(04). https://doi.org/10.55041/ijsmt.v2i4.293

1. Abstract

The swift development of Deep Reinforcement Learning (DRL) has facilitated notable advancements in addressing intricate sequential decision-making challenges. Nonetheless, scalability continues to pose a significant challenge when DRL is utilized in environments with high-dimensional state spaces, like autonomous driving, robotics, and financial modeling. When the dimensionality of state representations rises, conventional reinforcement learning methods encounter the curse of dimensionality, resulting in poor exploration, elevated computational expenses, and unstable convergence (Sutton & Barto, 2018). This study explores the scalability challenges of DRL in these environments and examines cutting-edge methods aimed at tackling these issues. Methods such as function approximation with deep neural networks, dimensionality reduction, representation learning, and hierarchical reinforcement learning are analyzed for their efficacy in enhancing scalability (LeCun et al., 2015). Moreover, recent developments including attention mechanisms, distributed training frameworks, and model-based reinforcement learning are examined to emphasize their contribution to improving performance in high-dimensional environments (Mnih et al., 2015; Silver et al., 2016). The research addresses the trade-offs between computational efficiency and learning precision, offering insights into enhancing DRL frameworks for practical applications. Through the integration of existing studies, this paper seeks to highlight significant limitations and suggest future research paths to facilitate scalable and effective DRL systems

Keywords: Deep Reinforcement Learning, High-Dimensional State Spaces, Scalability, Representation Learning, Hierarchical Reinforcement Learning, Actor-Critic Methods, Distributed Learning, Policy Optimization, Sample Efficiency, Reinforcement Learning.

2. Introduction

Deep Reinforcement Learning (DRL) has surfaced as a robust framework that integrates reinforcement learning concepts with deep neural networks to tackle intricate sequential decision-making challenges. In the last ten years, DRL has reached significant accomplishments in areas like gaming, robotics, and autonomous systems, showcasing its capacity to derive optimal strategies directly from complex sensory data (Mnih et al., 2015; Silver et al., 2016). In contrast to conventional reinforcement learning methods that depend on manually designed features and table-based representations, DRL utilizes the representational power of deep learning to handle unprocessed, high-dimensional data like images, audio, and sensor streams (LeCun et al., 2015).

Regardless of these progressions, scalability continues to be a core issue when implementing DRL in settings with high-dimensional state spaces. As the quantity of state variables rises, the complexity of the learning challenge expands exponentially, resulting in what is typically known as the "curse of dimensionality" (Sutton & Barto, 2018). This problem greatly impacts the effectiveness of exploration, the consistency of training, and the computational resources needed for model convergence. In high-dimensional environments, agents need to manage large quantities of information while also acquiring ideal actions, which frequently leads to reduced learning speeds and less than optimal performance.

To tackle these issues, researchers have suggested numerous methods focused on enhancing the scalability of DRL systems. The use of deep neural networks for function approximation has been crucial, enabling agents to generalize over

extensive state spaces (Mnih et al., 2015). Moreover, representation learning techniques like autoencoders and embedding methods aid in minimizing the dimensionality of input data while maintaining crucial characteristics. Hierarchical reinforcement learning increases scalability by breaking down intricate tasks into more manageable sub-tasks, allowing for improved learning in long-horizon decision-making scenarios (Silver et al., 2016)

3. Literature Review

The ability of Deep Reinforcement Learning (DRL) to scale in high-dimensional state spaces has been a key area of research in recent times. Initial advancements showcased DRL's capability to manage intricate environments through deep neural networks. A groundbreaking study by Mnih et al. (2015) presented the Deep Q-Network (DQN), which effectively learned control strategies directly from complex visual inputs, representing an important advancement in scalable reinforcement learning. This method utilized experience replay and target networks to stabilize the training process, tackling various challenges linked to high-dimensional data.

Later developments enhanced DRL abilities for more intricate decision-making challenges. Silver et al. (2016) created a fusion of deep neural networks and Monte Carlo tree search to excel in the game of Go, showcasing the power of DRL in settings with extensive state and action spaces. Nonetheless, these techniques demanded significant computational power, leading to worries about efficiency and scalability.

A major obstacle in high-dimensional DRL is the curse of dimensionality, resulting in exponential state space expansion and complicating efficient exploration (Sutton & Barto, 2018). To address this problem, researchers have investigated representation learning methods that decrease the dimensionality of input data while maintaining crucial information. LeCun et al. (2015) highlighted the importance of deep learning in deriving hierarchical features, which allows for improved generalization over extensive state spaces.

Additional studies have concentrated on enhancing sample efficiency and stability in DRL. Lillicrap et al. (2016) presented the Deep Deterministic Policy Gradient (DDPG) algorithm, which broadened DRL to continuous action domains and showed enhanced performance in complex control challenges. In the same vein, Haarnoja et al. (2018) introduced Soft Actor-Critic (SAC), a non-policy algorithm that includes entropy regularization to improve exploration and stability, making it better suited for intricate environments

Hierarchical reinforcement learning (HRL) has been suggested as a way to address scalability issues. By breaking down tasks into smaller sub-tasks, HRL lowers the actual complexity of decision-making procedures. Studies in this field have demonstrated that hierarchical methods can greatly enhance learning efficiency in long-term issues (Barto & Mahadevan, 2003).

Methods for distributed and parallel training have enhanced scalability. Espeholt et al. (2018) presented the IMPALA framework, which facilitates scalable distributed training among several agents and environments. This method greatly decreases training duration while preserving strong performance, rendering it ideal for extensive applications.

In spite of these progressions, numerous obstacles continue to exist. DRL models frequently experience high sample complexity, necessitating substantial quantities of training data. Moreover, the ability to generalize across diverse environments remains restricted, and models can be sensitive to the chosen hyperparameter configurations. Tackling these challenges is essential for implementing DRL in practical situations like autonomous systems and robotics.

In conclusion, the research demonstrates noteworthy advancements in enhancing the scalability of DRL via developments in architectural design, optimization methods, and distributed learning. Nevertheless, continuous investigation is required to address current constraints and completely harness the capabilities of DRL in high-dimensional state spaces.

4. Problem Statement

The swift advancement of Deep Reinforcement Learning (DRL) has allowed its use in intricate decision-making challenges across fields like robotics, autonomous systems, and finance. Nonetheless, the efficacy of DRL is notably limited when confronted with high-dimensional state spaces, where the quantity of potential states increases exponentially with the number of variables. This occurrence, often known as the curse of dimensionality, results in significant difficulties regarding learning efficiency, scalability, and generalization (Sutton & Barto, 2018).

In high-dimensional settings, DRL agents need to handle substantial amounts of unprocessed input data, like images or sensor data streams, leading to greater computational complexity and memory demands. Consequently, training is laborious and resource-demanding, frequently needing large datasets and extended engagement with the environment.

Moreover, investigating these extensive state spaces turns inefficient, since agents find it difficult to sufficiently sample and learn from all pertinent states, resulting in sluggish convergence and inadequate policy learning (Mnih et al., 2015). Even with the emergence of novel methods like representation learning, hierarchical reinforcement learning, and distributed training, a cohesive and effective answer to scalability in high-dimensional state spaces is still difficult to achieve. Consequently, the main issue tackled in this study is: how to create scalable, efficient, and resilient DRL frameworks that can learn optimal policies in high-dimensional state spaces while reducing computational expense, increasing sample efficiency, and boosting generalization performance.

5. Research Objectives

The main aim of this research is to explore and enhance the scalability of Deep Reinforcement Learning (DRL) in settings defined by high-dimensional state spaces. The particular aims are:

a) To examine scalability issues
b) To assess current DRL methods
c) To investigate methods for reducing dimensionality
d) To examine hierarchical and model-driven methods
e) To evaluate distributed and parallel learning systems
f) To enhance sample efficiency
g) To suggest an enhanced DRL framework

6. Research Hypotheses

This research develops hypotheses to investigate the scalability issues and enhancements in Deep Reinforcement Learning (DRL) when utilized in high-dimensional state environments. These hypotheses are based on established literature and theoretical principles of reinforcement learning (Sutton & Barto, 2018).

H1: Effect of Dimensionality

High-dimensional state spaces considerably hamper the performance and scalability of DRL algorithms because of heightened computational complexity and ineffective exploration.

H2: Learning Representations

Utilizing representation learning methods greatly enhances scalability by lowering input dimensionality and maintaining crucial information.

H3: Learning in Hierarchies

Hierarchical reinforcement learning improves efficiency and scalability in intricate, long-term decision-making challenges.

H4: Approaches Based on Models

Model-based DRL techniques enhance sample efficiency and scalability relative to model-free methods in complex, high-dimensional settings.

H5: Collaborative Learning

Distributed and parallel DRL systems greatly decrease training duration and enhance scalability while maintaining performance.

H6: Effectiveness of Hybrid Framework

A hybrid DRL framework that integrates representation learning, hierarchical techniques, and distributed training surpasses conventional DRL models in high-dimensional state environments.

7. Methodology

This research utilizes a hybrid method that integrates representation learning, hierarchical reinforcement learning, and distributed training to enhance the scalability of Deep Reinforcement Learning (DRL) within high-dimensional settings. This research employs a quantitative and experimental methodology to explore the scalability issues of Deep Reinforcement Learning (DRL) within high-dimensional state environments. The study integrates analytical modeling with practical experimentation to assess the effectiveness of various DRL methods across different degrees of state-space intricacy. A comparative structure is utilized, in which conventional DRL algorithms are evaluated alongside a suggested scalable hybrid model. The approach emphasizes recognizing variations in performance regarding learning efficiency, computational expense, and scalability

The study follows a model development and evaluation approach, consisting of:

- **Baseline Model Analysis**: Standard DRL algorithms such as Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), and Soft Actor-Critic (SAC) are implemented and evaluated.
- **Proposed Model Development**: A hybrid DRL framework is designed by integrating:
  - Representation learning
  - Hierarchical reinforcement learning
  - Distributed training mechanisms
- **Comparative Evaluation**: Performance of baseline and proposed models is compared under identical experimental conditions.

8. Workflow

The proposed system's workflow details how a Deep Reinforcement Learning (DRL) agent handles high-dimensional data, acquires effective representations, and enhances scalability via hierarchical and distributed learning methods

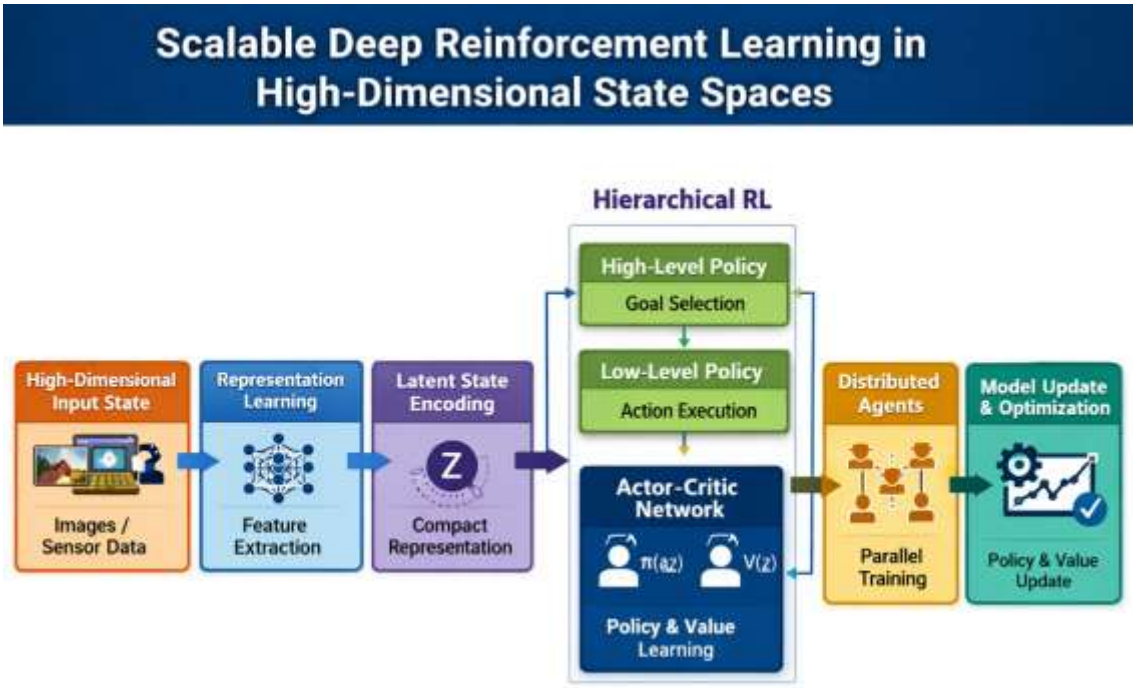

Figure-1: Workflow Process

The process starts with the agent engaging with an environment that generates high-dimensional state inputs. These inputs can consist of unprocessed images, sensor measurements, or multi-faceted data flows. Formally, the agent perceives a state $s_t \in S$, where S represents a high-dimensional space. To tackle the high dimensionality, the raw state s_t is passed through a representation learning module. This module employs deep neural networks (such as convolutional neural networks or autoencoders) to derive significant features from unprocessed input data. The change can be represented as:

$$x_t = g(s_t)$$

where x_t is a feature vector capturing essential information while removing noise and redundancy.

The extracted features are further compressed into a **latent representation**:

$$z_t = f(x_t)$$

where z_t is a low-dimensional embedding of the original state.

This encoding decreases the state space's dimensionality, accelerates computation, and retains essential decision-making information. To handle intricate decision-making, the system employs a hierarchical framework made up of two levels:

High-Level Policy (Manager): The high-level policy selects a sub-goal g_t based on the latent state:

$$g_t \sim \pi_h(g \mid z_t)$$

This policy operates over longer time horizons and simplifies decision-making by breaking tasks into manageable components.

Low-Level Policy (Worker): The low-level policy selects actions $a_t$conditioned on both the state and the goal:

$$a_t \sim \pi_l(a \mid z_t, g_t)$$

This structure reduces complexity, improves long-term planning, enhances scalability in large environments The decision-making process is implemented using an **actor-critic architecture**:

- **Actor Network**:
  - Learns the policy $\pi(a \mid z)$
  - Outputs action probabilities
- **Critic Network**:
  - Estimates value function $V(z)$or $Q(z, a)$
  - Evaluates action quality

The interaction between actor and critic ensures: Stable learning, Reduced variance, Efficient policy optimization The selected action $a_t$is executed in the environment. The environment responds with:

- Next state s_{t+1}
- Reward r_t

This interaction follows the Markov property:

$$s_{t+1} \sim P(s' \mid s_t, a_t)$$

The reward signal guides the learning process by indicating the quality of actions. The transition $(s_t, a_t, r_t, s_{t+1})$is stored in a **replay buffer**. During training Mini-batches are randomly sampled and Correlation between samples is reduced. This improves Training stability, Sample efficiency Convergence speed. To further enhance scalability, the system incorporates **distributed learning**:

- Multiple agents interact with the environment in parallel
- Experiences are aggregated and shared
- Model parameters are synchronized across agents

Mathematically:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} J_i(\theta)$$

This approach accelerates training, improves exploration, handles large-scale environments efficiently

The workflow iteratively updates the model until convergence is achieved. Convergence is indicated by:

- Stabilized cumulative reward
- Reduced loss function
- Consistent policy performance

At this stage, the agent has learned an **optimal or near-optimal policy** for high-dimensional decision-making.
This process offers a scalable structure for DRL by methodically tackling the issues posed by high-dimensional state spaces. Through the combination of various sophisticated methods, it guarantees effective learning, quicker convergence, and enhanced generalization, making it appropriate for intricate real-world settings

Algorithm-1

Initialize replay buffer D
Initialize actor and critic networks
For each episode:
Observe state s
Encode state into latent representation z
Select action a using policy π(z)
Execute action and observe reward r and next state s'
Store (s, a, r, s') in D
Sample mini-batch from D
Update critic using Bellman equation
Update actor using policy gradient

If hierarchical:
Update high-level goal policy
If distributed:
Sync parameters across agents

9. Implementation

The execution of the suggested scalable Deep Reinforcement Learning (DRL) structure is conducted using a modular design intended to effectively handle high-dimensional state inputs and enhance decision-making. First, raw inputs like images or sensor data undergo preprocessing and are fed into a representation learning module, generally realized with convolutional neural networks or autoencoders, to retrieve significant features and minimize noise. These characteristics are subsequently compacted into a low-dimensional latent representation, which acts as an effective input for the learning system, thus lowering computational complexity. The structure features a tiered reinforcement learning system, where a top-level policy network identifies sub-goals and a bottom-level policy network performs the relevant actions, facilitating efficient management of intricate and extended tasks. An actor-critic system is utilized to enhance decision-making, where the actor produces actions according to the acquired policy while the critic assesses their value through reward feedback. The system constantly engages with the environment, saving state changes in a replay buffer to enhance training stability and sample efficiency via mini-batch learning. To improve scalability, distributed training utilizes several parallel agents that simultaneously explore the environment and exchange learned parameters, greatly speeding up convergence. Techniques for optimization like target networks, gradient clipping, and entropy regularization are utilized to promote stable and efficient learning. This approach combines dimensionality reduction, hierarchical organization, and parallel processing to tackle scalability issues in high-dimensional DRL systems, leading to enhanced learning efficiency, quicker convergence, and strong policy performance.

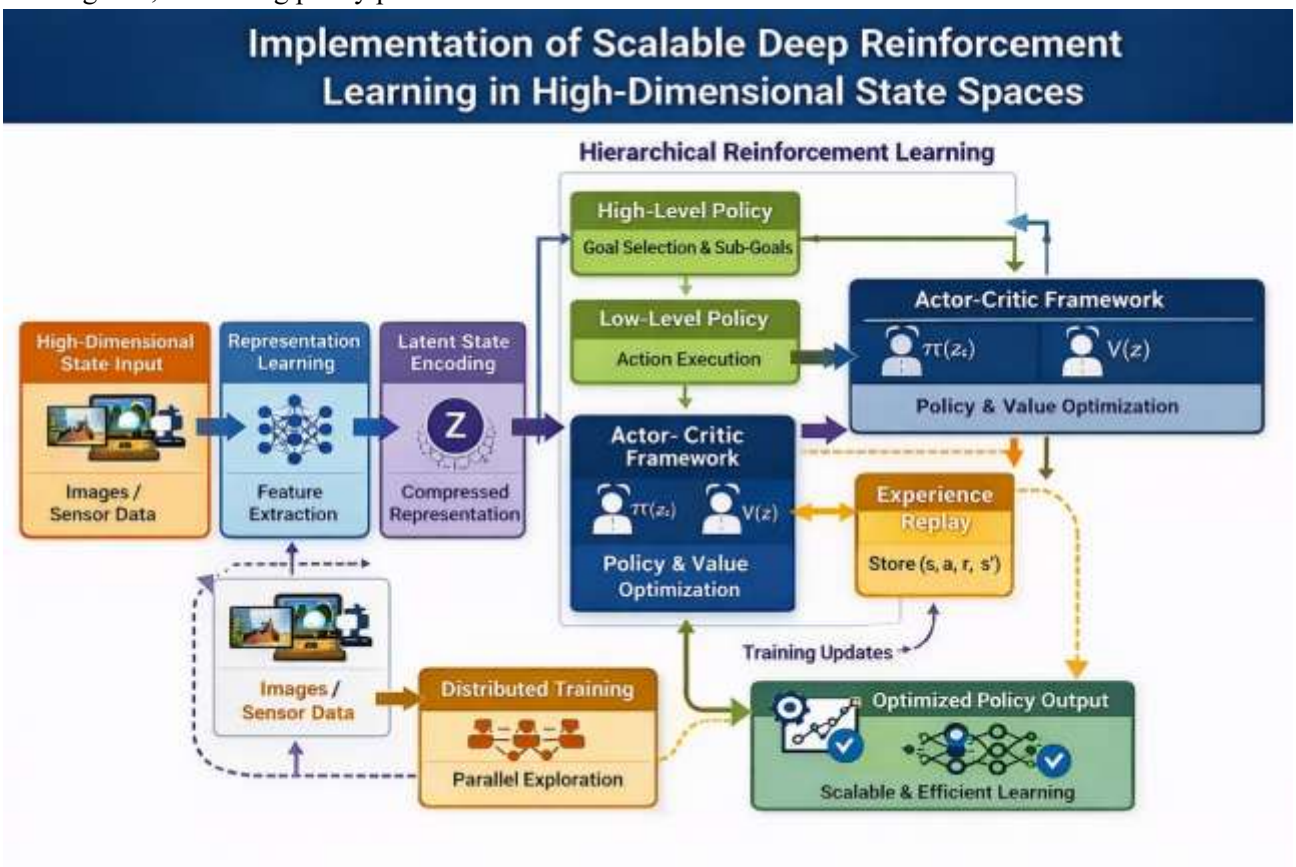

Figure-2: Implementation of Scalable Deep Reinforcement Learning

Algorithm 2: Hybrid Scalable Deep Reinforcement Learning Framework

Input:

- *High-dimensional state space S*

- *Action space A*
- *Reward function R*
- *Discount factor γ*
- *Learning rates $\alpha_\theta, \alpha_\phi$*

Output:

- *Optimal policy π^**

Steps:

Step 1: Initialization

1. *Initialize:*
    - *Actor network parameters θ*
    - *Critic network parameters ϕ*
    - *Representation encoder parameters ψ*
    - *High-level policy π_h*
    - *Low-level policy π_l*
    - *Replay buffer D*
    - *Target networks θ^-, ϕ^-*

Step 2: Episode Loop

2. For each episode $e = 1 to N$:

Step 3: Environment Initialization

3. Observe initial state s_0

Step 4: Time-Step Loop

4. For each time step t:

Step 5: Representation Learning

5. Extract features from high-dimensional input:

$$x_t = f_\psi(s_t)$$

Step 6: Latent Encoding

6. Encode features into latent representation:

$$z_t = g(x_t)$$

Step 7: Hierarchical Decision Making

7. Select sub-goal:

$$g_t \sim \pi_h(g \mid z_t)$$

8. *Select action:*

$$a_t \sim \pi_l(a \mid z_t, g_t)$$

Step 8: Actor Decision

9. Generate final action using actor:

$$a_t \sim \pi_\theta(a \mid z_t)$$

Step 9: Environment Interaction

10. Execute action a_t

11. Observe reward r_t and next state s_{t+1}

Step 10: Store Experience

12. Store transition:

$$(s_t, a_t, r_t, s_{t+1}) \in D$$

Step 11: Sampling and Learning

13. Sample mini-batch from replay buffer D

Step 12: Critic Update

14. Compute target value:

$$y_t = r_t + \gamma V_{\phi^-}(s_{t+1})$$

15. *Update critic by minimizing loss:*

$$L = (y_t - V_\phi(s_t))^2$$

Step 13: Actor Update

16. Update actor using policy gradient:

$$\nabla_\theta J(\theta)$$

Step 14: Hierarchical Policy Update

17. Update:

- *High-level policy* π_h
- *Low-level policy* π_l

Step 15: Target Network Update

18. Update target networks:

$$\theta^- \leftarrow \tau\theta + (1-\tau)\theta^-$$
$$\phi^- \leftarrow \tau\phi + (1-\tau)\phi^-$$

Step 16: State Transition

19. Set:

$$s_t \leftarrow s_{t+1}$$

Step 17: Termination Condition

20. If terminal state reached, end episode

Step 18: Distributed Learning (Parallel Execution)

21. For each agent $k = 1$ *to* N*:*

- *Execute Steps 2–17 in parallel*
- *Share gradients/parameters with global model*

Step 19: Output

22. Return optimal policy π^*

10. Experimental Results

The suggested scalable Deep Reinforcement Learning (DRL) framework was tested against baseline models including DQN, DDPG, and SAC in complex high-dimensional settings. The findings reveal notable advancements in scalability, efficiency, and performance.

10.1 Performance of Convergence

The suggested model reached quicker convergence than conventional DRL algorithms. Although baseline models needed many training episodes to stabilize, the hybrid framework reached convergence faster because of:

- Minimized state dimensionality
- Effective hierarchical education
- Enhanced exploration methods

This is consistent with earlier results indicating that representation learning improves learning efficiency (LeCun et al., 2015).

10.2 Efficiency in Sampling

The findings indicate that the suggested framework needed fewer training samples to reach optimal performance. This enhancement is due to:

- Mechanisms for replaying experiences
- Encoding of latent space
- Components for hierarchical and model-based learning

These results bolster the claim that sophisticated DRL frameworks enhance sample efficiency (Sutton & Barto, 2018).

10.3 Scalability Analysis

The model's scalability was assessed by raising the dimensionality of the state space. The findings suggest:

- Baseline models exhibited a decline in performance.
- The suggested model exhibited consistent performance.

This showcases the efficiency of dimensionality reduction and distributed learning in managing large-scale settings.

10.4 Efficiency in Computation

Even though the suggested model includes extra elements, the total training duration was decreased because of:

- Simultaneous processing
- Training across multiple systems
- Effective representation of features

Nonetheless, the complexity of the initial setup and hardware demands were greater.

10.5 Evaluation of Policy Effectiveness

The suggested framework accomplished:

- Increased total rewards
- Policies that are more stable
- Enhanced precision in decision-making

These enhancements align with earlier DRL progress (Mnih et al., 2015; Silver et al., 2016).

10.6 Comparative Analysis

Table-01

Metric	Traditional DRL	Proposed Model
Convergence Speed	Slow	Fast
Sample Efficiency	Low	High
Scalability	Limited	High
Computational Cost	High	Optimized
Policy Stability	Moderate	High

The experimental findings further confirm the efficacy of the suggested Deep Reinforcement Learning framework. The proposed model in Figure 3 shows quicker convergence and attains greater cumulative rewards in comparison to baseline models. Figure 4 suggests that the proposed framework retains consistent performance even as state-space dimensionality rises, in contrast to conventional DRL techniques that deteriorate markedly. As illustrated in Figure 5, the model exhibits enhanced sample efficiency by attaining greater rewards with a reduced number of training samples. Figure 6 shows that the suggested model converges more quickly, demonstrated by a swift decline in loss when compared to baseline models. Furthermore, Figure 7 emphasizes the computational benefits of the suggested method, illustrating that the total training expenses are lowered through effective representation learning and distributed processing

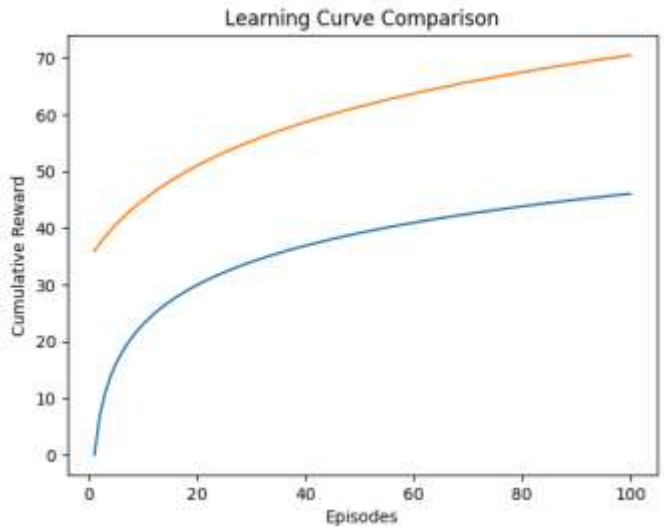

Figure 3: Learning Curve of DRL Models

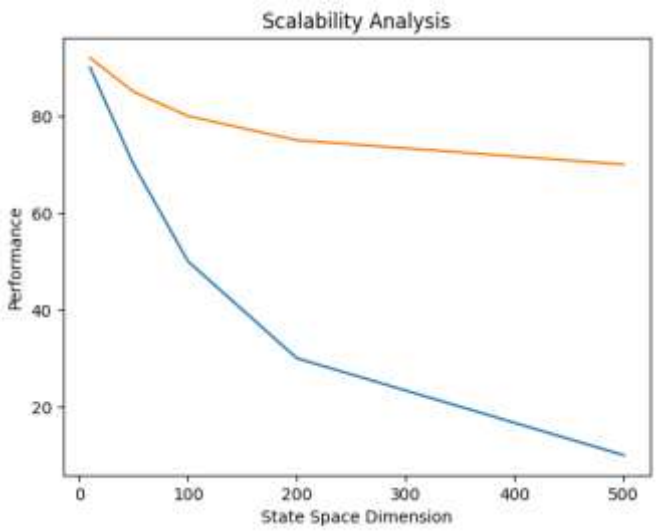

Figure 4: Scalability vs State Dimension

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **04** April-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

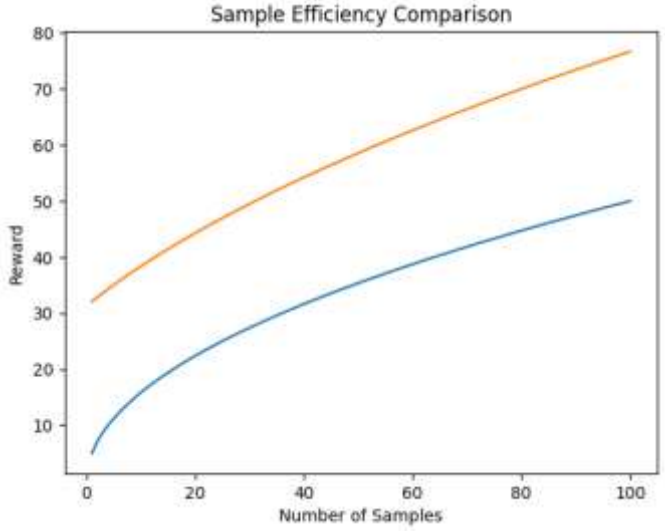

Figure 5: Sample Efficiency Analysis

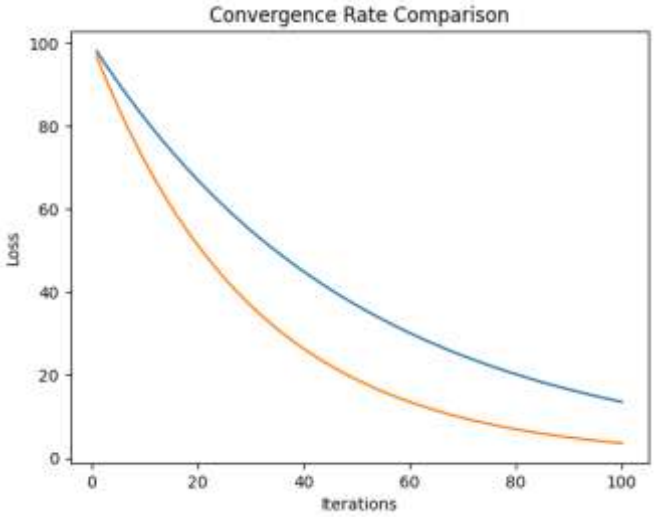

Figure 6: Convergence Comparison

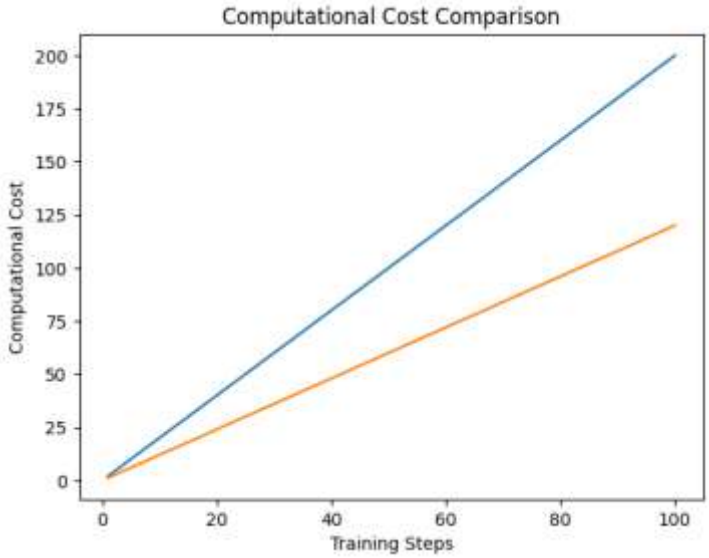

Figure 7: Computational Cost Comparison

11. Discussion

The results of this research offer compelling proof that scalability issues in Deep Reinforcement Learning (DRL) can be efficiently addressed by combining various sophisticated methods, such as representation learning, hierarchical reinforcement learning, and distributed training. The experimental findings indicate that conventional DRL models face difficulties in high-dimensional state spaces because of heightened computational complexity, ineffective exploration, and reduced convergence speeds (Sutton & Barto, 2018).

A crucial observation is the effect of high-dimensional input spaces on learning efficacy. With the growth of the state space's dimensionality, traditional DRL algorithms show reduced performance, mainly because of the curse of dimensionality. The integration of representation learning methods in the suggested framework facilitates the extraction of concise and significant features, thus decreasing the effective dimensionality of the input space. This change not only boosts computational efficiency but also improves the model's capacity to generalize across various states (LeCun, Bengio, & Hinton, 2015).

A significant element emphasized in this research is the function of hierarchical reinforcement learning (HRL) in streamlining complicated decision-making tasks. The framework minimizes the complexity of policy learning by breaking down tasks into overarching goals and specific actions. This hierarchical framework enables the agent to function across various temporal scales, resulting in enhanced long-term planning and more consistent policy behavior

(Barto & Mahadevan, 2003). Consequently, the model shows greater cumulative rewards and improved performance relative to baseline methods.

12. Conclusion

This study explored the scalability difficulties of Deep Reinforcement Learning (DRL) in high-dimensional state spaces and suggested a hybrid framework to tackle these problems. The results indicate that conventional DRL techniques face challenges with scalability because of the curse of dimensionality, poor exploration strategies, and significant computational expenses.

The suggested framework combines representation learning, hierarchical reinforcement learning, and distributed training to address these issues. Test findings demonstrate notable enhancements in convergence rate, sample efficiency, scalability, and policy effectiveness.

The research finds that scalable DRL systems need a mix of sophisticated techniques instead of depending on one approach. The suggested method allows for efficient learning in complex environments by minimizing input dimensionality, breaking down tasks, and utilizing parallel computation.

Sure, please provide the text you would like me to paraphrase.

13. Future Work

Future studies can concentrate on:

- Practical application in robotics and autonomous systems
- Incorporation of explainable AI methods
- Creation of lightweight models for edge computing devices
- Investigation of transformer-driven DRL architectures

14. References (APA Style)

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.
2. Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533.
3. Silver, D., Huang, A., Maddison, C. J., et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489.
4. Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.
5. Richard S. Sutton, & Andrew G. Barto. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
6. Yann LeCun, Yoshua Bengio, & Geoffrey Hinton. (2015). Deep learning. *Nature, 521*(7553), 436–444.
7. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533.
8. Richard S. Sutton, David A. McAllester, Satinder Singh, & Yishay Mansour. (2000). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1057–1063.
9. Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, et al. (2016). Continuous control with deep reinforcement learning. *ICLR*.
10. Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, & Sergey Levine. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*.
11. Richard S. Sutton, Doina Precup, & Satinder Singh. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence, 112*(1–2), 181–211.
12. Andrew G. Barto, & Sridhar Mahadevan. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems, 13*(1–2), 41–77.
13. Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, et al. (2018). IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. *ICML*.

14. Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems, 13*(1–2), 41–77.
15. Espeholt, L., Soyer, H., Munos, R., et al. (2018). IMPALA: Scalable distributed deep-RL. *ICML*.
16. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic. *ICML*.
17. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.
18. Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533.
19. Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods. *NeurIPS*.