SPRIG: Stackelberg Perception-Reinforcement Learning with Internal Game Dynamics

Fernando Martinez-Lopez[1], Juntao Chen[1], Yingdong Lu[2]
[1]Fordham University
[2]IBM Research
{fmartinezlopez, jchen504}@fordham.edu, yingdong@us.ibm.com

Abstract

Deep reinforcement learning agents often face challenges to effectively coordinate perception and decision-making components, particularly in environments with high-dimensional sensory inputs where feature relevance varies. This work introduces SPRIG (Stackelberg Perception-Reinforcement learning with Internal Game dynamics), a framework that models the internal perception-policy interaction within a single agent as a cooperative Stackelberg game. In SPRIG, the perception module acts as a leader, strategically processing raw sensory states, while the policy module follows, making decisions based on extracted features. SPRIG provides theoretical guarantees through a modified Bellman operator while preserving the benefits of modern policy optimization. Experimental results on the Atari BeamRider environment demonstrate SPRIG's effectiveness, achieving around 30% higher returns than standard PPO through its game-theoretical balance of feature extraction and decision-making.

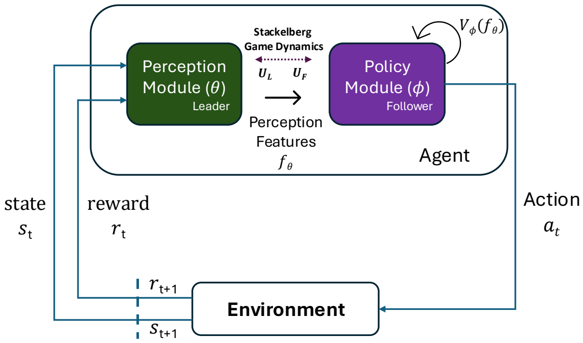

Figure 1: SPRIG architecture overview

Introduction

Deep Reinforcement Learning (RL) has successfully solved complex tasks across various domains, from game playing to robotic control (Mnih et al. 2015; Berner et al. 2019; Ibarz et al. 2021). However, a fundamental challenge persists: the effective coordination between perception and decision-making components, particularly in environments with high-dimensional sensory inputs where the relevance of features varies across tasks or time (Mao et al. 2024).

While traditional approaches treat perception and decision-making as a unified process, this integration overlooks fundamental insights from cognitive science, particularly the two-stream hypothesis of visual processing (Goodale and Milner 1992). In biological systems, visual information flows through distinct pathways: a "what" stream for object recognition and a "how" stream for action guidance. This natural division suggests an inherent hierarchy where perceptual processing precedes and informs action selection as separate subsystems. Despite this biological inspiration, current RL approaches lack the fundamentals of this natural cooperative design.

We address these challenges by introducing SPRIG (Stackelberg Perception-Reinforcement learning with Internal Game dynamics), a framework that models the perception-policy interaction as a cooperative Stackelberg game. Our *perception module*, implemented as a hierarchical spatio-temporal attention mechanism, acts as a leader that strategically processes raw sensory inputs, while the *policy module* follows by making decisions based on the extracted features. Our game-theoretical formulation provides: (1) a principled mathematical framework for perception-policy interaction through a modified Bellman operator, (2) thorough theoretical analysis with provable convergence properties while maintaining the advantages of modern policy optimization, and (3) the creation of a natural balance between feature extraction and policy optimization through our cooperative formulation. Our approach extends the Proximal Policy Optimization algorithm (PPO; (Schulman et al. 2017)) to incorporate this game-theoretical dynamic, introducing a two-stage optimization process with advantage normalization. Through our formulation of perception cost and utility functions, we ensure that our method converges to a unique fixed point, providing both theoretical soundness and practical applicability.

We demonstrate the effectiveness of SPRIG on the Atari BeamRider environment, where the perception module must identify and track relevant visual features for successful policy learning. Our empirical results show that SPRIG achieves higher than standard PPO, with returns reaching approximately 850 versus 650 for the baseline.

Related Work

Integrating perception and decision-making in RL has become an interesting subdomain, especially in environments with high-dimensional sensory inputs. The Perception and Decision-making Interleaving Transformer (PDiT; (Mao et al. 2024)) uses separate transformers for perception and decision-making, leading to enhanced performance in complex tasks. Incorporating game-theoretic principles, (Zheng et al. 2022) proposed the Stackelberg Actor-Critic framework, modeling the actor-critic interaction as a Stackelberg game to improve learning stability. Extending this approach, (Huang et al. 2022) addressed robustness in uncertain environments by formulating robust RL as a Stackelberg game, demonstrating the adaptability of leader-follower structures in RL. Attention mechanisms have also been explored for adaptive feature extraction in RL. For instance, (Manchin, Abbasnejad, and Van Den Hengel 2019) introduced a self-supervised attention model that significantly improved performance in the Arcade Learning Environment, highlighting the potential of attention mechanisms in RL.

Our work advances these approaches by introducing a framework with theoretical guarantees through a modified Bellman operator that explicitly accounts for perception-policy interaction, while maintaining the advantages of modern policy optimization. Our cooperative game formulation creates a natural balance between feature extraction and decision-making, complementing previous approaches by adding provable convergence properties for the entire system and demonstrating empirical improvements.

Background and Preliminaries

Markov Decision Processes and Reinforcement Learning

A Markov Decision Process (MDP) provides the fundamental model for sequential decision-making under uncertainty (Sutton and Barto 2018). Formally, an MDP is defined as a tuple $\mathcal{M} = (S, A, P, R, \gamma)$, where S represents the state space, A the action space, $P : S \times A \times S \to [0, 1]$ the transition probability function, $R : S \times A \to \mathbb{R}$ the reward function, and $\gamma \in [0, 1)$ the discount factor.

Here an agent interacts with the environment by selecting actions according to a policy $\pi : S \to \Delta(A)$, where $\Delta(A)$ denotes the probability simplex over actions. The objective is to find an optimal policy π^* that maximizes the expected discounted return:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s \right]. \qquad (1)$$

The optimal policy π^* satisfies the Bellman optimality equation:

$$V^*(s) = \max_{a \in A} \left[R(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^*(s')] \right]. \qquad (2)$$

Stackelberg and Cooperative Games

Stackelberg games model sequential decision-making scenarios through a hierarchical structure. Let $\mathcal{G} = (N, \Theta, \Phi, u_L, u_F)$ be a two-player game where $N = \{L, F\}$ denotes the leader and follower, with strategy spaces Θ and Φ respectively. The utility functions $u_L : \Theta \times \Phi \to \mathbb{R}$ and $u_F : \Theta \times \Phi \to \mathbb{R}$ define the payoffs for each player, though their usage differs due to the sequential nature of the game.

In this interplay, the leader commits to a strategy $\theta \in \Theta$, after which the follower observes this commitment and responds with $\phi \in \Phi$. This creates a subgame perfect equilibrium where the follower's best response function is:

$$BR_F(\theta) = \{\phi \in \Phi : u_F(\theta, \phi) \geq u_F(\theta, \phi') \text{ for all } \phi' \in \Phi\}. \qquad (3)$$

The leader, anticipating this response, solves:

$$\theta^* = \arg\max_{\theta \in \Theta} u_L(\theta, BR_F(\theta)). \qquad (4)$$

While Stackelberg games capture hierarchical interaction, cooperative game theory provides tools for analyzing scenarios where players coordinate for mutual benefit. A cooperative game is defined by (N, v), where N is the player set and $v : 2^N \to \mathbb{R}$ is the characteristic function assigning values to coalitions.

In our two-player setting, the cooperative value emerges through a weighted combination of individual utilities:

$$v(\{L, F\}) = \alpha u_L(\theta, \phi) + (1 - \alpha) u_F(\theta, \phi), \qquad (5)$$

where $\alpha \in [0, 1]$ represents the cooperation weight. The solution concept focuses on finding allocations that maximize this joint value while ensuring individual rationality: $v(\{i\}) \leq u_i$ for $i \in \{L, F\}$.

Perception-Policy Learning: Motivation and Need

Current approaches to perception-policy learning typically fall into two categories. The first approach treats perception and policy as a single end-to-end system, while the second attempts to optimize these components independently. Nevertheless, recent work has shown that perception and decision models separately can lead to reduce robustness since mismatched state extraction and control decision-making become asynchronous (Zhu et al. 2022). Standard RL models often treat these processes as a unified pipeline, optimizing perception and policy jointly in an end-to-end fashion. While this approach simplifies implementation, it struggles to generalize in high-dimensional environments where irrelevant features dominate or feature relevance varies over time, as evidenced in complex visual navigation tasks (Zhu et al. 2017). Such limitations derives from the inability to effectively balance the demands of feature extraction with those of action selection.

Inspired by the two-stream hypothesis of visual processing (Goodale and Milner 1992), we argue for a principled separation of perception and policy into distinct, hierarchically organized modules, aligning with approaches in hierarchical RL that decompose complex tasks (Diuk et al. 2013). This biological insight suggests that perception should focus on extracting meaningful, task-relevant features while policy concentrates on optimal action selection based on these features. This separation enables better modularity and adaptability in complex environments, akin to how biological systems achieve robust and efficient decision-making.

However, decoupling perception from policy introduces coordination challenges. Misalignment between the extracted features and the policy's decision-making objectives can degrade performance, necessitating a structured design to guide this interaction. Game theory, particularly Stackelberg games, provides a natural solution. By modeling the perception module as a leader and the policy module as a follower, we establish a hierarchical interaction where the perception module optimizes its feature extraction strategy while anticipating the policy module's response.

Through this hierarchical formulation, we address the shortcomings of traditional RL methods, introducing a modular, biologically inspired framework capable of robust generalization in complex tasks. This structured interaction also facilitates theoretical analysis and practical improvements, setting the foundation for the novel game-theoretical approach introduced in this paper.

Our Approach

We propose SPRIG (Stackelberg Perception-Reinforcement Learning with Internal Game Dynamics), a cooperative Stackelberg game framework for perception-policy learning in RL. Our approach builds upon PPO, extending it to incorporate game-theoretical dynamics between modules. As shown in Figure. 1, the SPRIG architecture comprises two key components: the *perception module*, which acts as the leader, and the *policy module*, which serves as the follower.

Perception-Policy Game Formulation

In our SPRIG agent, the perception module θ implements a hierarchical spatio-temporal attention mechanism consisting of three convolutional layers combined with self-attention, mapping raw states S to features $\mathcal{F}$. This way, the agent can process raw visual inputs by progressively refining spatial relationships while maintaining temporal consistency across frames. On the other hand, the policy module ϕ consists of a Multi-Layer Perception that takes the feature representation coming from the perception module and outputs action probabilities. The policy module is optimized iteratively using PPO, alternating between policy updates and value function updates.

The interaction between these modules is formulated as a cooperative Stackelberg game where:

$$\theta^* = \arg\max_{\theta\in\Theta} u_L(\theta, \phi^*(\theta)), \tag{6}$$

$$\phi^*(\theta) = \arg\max_{\phi\in\Phi} \mathbb{E}_{\pi_\phi}[R(s,a)]. \tag{7}$$

The leader's utility function u_L balances both the policy's performance and the perception computational efficiency:

$$u_L(\theta,\phi) = \alpha_{coop}\mathbb{E}_{\pi_\phi}[R(s,a)] - (1-\alpha_{coop})C_\theta(s). \tag{8}$$

where α_{coop} is the cooperation weight and $C_\theta(s)$ represents the perception cost. The cost function penalizes excessive attention across all layers:

$$C_\theta(s) = \lambda_c \sum_{k=1}^{K} \mathbb{E}_{s\sim\mathcal{D}}[\|A_k(s)\|_1], \tag{9}$$

where $A_k(s)$ represents the attention weights at layer k, λ is the cost weight, $\mathcal{D}$ is the distribution of states encountered during training, and K is the total number of layers.

Stackelberg Equilibrium Computation

The Stackelberg equilibrium is computed through a two-stage optimization process. In the first stage, the perception module optimizes its utility while anticipating the policy module's response (Eq. (8)):

$$\mathcal{L}_\theta = -u_L(\theta,\phi). \tag{10}$$

This optimization uses Generalized Advantage Estimation (GAE; (Schulman et al. 2015)) to compute advantages, which are normalized for training stability. The perception cost directly influences this stage by penalizing excessive attention allocation.

In the second stage, the policy module optimizes its objective given the features provided by the perception module:

$$\mathcal{L}_\phi = -\mathbb{E}_{\pi_\phi}[R(s,a)] + \beta H(\pi_\phi), \tag{11}$$

where $H(\pi_\phi)$ is the policy entropy and β is the entropy coefficient. The policy optimization includes both value function and policy updates, with the perception cost indirectly affecting this stage through the quality of extracted features, as outlined in Algorithm 1.

The perception cost influences both optimization stages: directly in the leader's utility computation and indirectly in the follower's optimization through feature quality. This dual influence creates a balanced cooperation between modules, where the perception module must provide useful features while maintaining computational efficiency, and the policy module must effectively utilize these features for decision-making.

Theoretical Formulation and Convergence Properties

Stackelberg-MDP Formulation We extend the traditional MDP framework to incorporate the perception-policy interaction through a cooperative Stackelberg game. Our augmented MDP is defined as $\mathcal{M}_S = (S, A, P, R, \gamma, \Theta, \Phi, C)$, where Θ is the perception parameter space, Φ is the policy parameter space, and $C : S \times \Theta \to [0,1]$ is the perception cost function implemented through attention mechanisms.

Bellman Operator and Properties For our Stackelberg-MDP, we first define the standard Bellman operator $\mathcal{T}$ for MDPs:

$$(\mathcal{T}f)(s,a) = R(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[\max_{a'} f(s',a')]. \tag{12}$$

Building upon this, we define our Stackelberg-Bellman operator $\mathcal{T}_S$ that incorporates the perception-policy interaction:

$$\begin{aligned}(\mathcal{T}_S f)(s,a) = \max_{\theta\in\Theta}\min_{\phi\in\Phi} \Big[& R(s,a) - \lambda C_\theta(s) \\ & + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\big[f(s',a';\phi\big]\Big],\end{aligned} \tag{13}$$

where $C_\theta(s) = \sum_{k=1}^{K} \|A_k(s)\|_1$ represents our implemented attention-based perception cost (Equation (9)).

Algorithm 1: SPRIG Agent: Cooperative Stackelberg Game Training for Perception-Policy Learning

```
Require: Initial parameters $\theta$ for perception, $\phi$ for policy module.
Require: Cooperation weight $\alpha$, discount factor $\gamma$, GAE parameter $\lambda$
for each iteration do
    Collect trajectories $\mathcal{D}$ using current policy
    Compute returns and normalize GAE values $\hat{A}_t$
    for each PPO epoch do
        for each mini-batch $\mathcal{B}$ do
            // Leader (Perception) Stage
            $f_\theta \leftarrow$ perception features for states in $\mathcal{B}$
            $C_\theta$                ▷ Attention cost, Equation (9)
            $\pi_\phi \leftarrow$ policy distribution from $f_\theta$
            $u_{\text{policy}} \leftarrow (\log \pi_\phi(a) \cdot \hat{A}_t)_{\text{mean}}$
            $u_L \leftarrow \alpha_{coop}(-C_\theta) + (1 - \alpha_{coop})u_{\text{policy}}$
            Update $\theta$ by maximizing $u_L$ with gradient clipping
            // Follower (Policy) Stage
            Compute PPO ratio $r_t(\phi)$
            $\mathcal{L}_{\text{CLIP}} \leftarrow \min(r_t(\phi)\hat{A}_t, \text{clip}(r_t(\phi), 1\pm\epsilon)\hat{A}_t)$
            $\mathcal{L}_V \leftarrow (V_\phi(s_t) - R_t)^2$
            $\mathcal{L}_\phi \leftarrow -\mathcal{L}_{\text{CLIP}} + 0.5\mathcal{L}_V - 0.01H(\pi_\phi) + C_\theta$
            Update $\phi$ by minimizing $\mathcal{L}_\phi$ with gradient clipping
        end for
    end for
end for
```

Contraction Properties The Stackelberg-Bellman operator $\mathcal{T}_S$ maintains the contraction property under the following conditions: 1) *bounded rewards*: $|R(s,a)| \leq R_{max}$; 2) *bounded perception cost*: $0 \leq C_\theta(s) \leq 1$ (guaranteed by our L_1-norm attention cost); and 3) *discount factor*: $\gamma \in [0, 1)$. For any two value functions f_1 and f_2:

$$\|\mathcal{T}_S f_1 - \mathcal{T}_S f_2\|_\infty \leq \gamma \|f_1 - f_2\|_\infty. \tag{14}$$

The contraction property of $\mathcal{T}_S$ ensures the existence of a unique fixed point f^* satisfying $f^* = \mathcal{T}_S f^*$, guarantees convergence of value iteration: $\|\mathcal{T}_S^n f - f^*\|_\infty \leq \gamma^n \|f - f^*\|_\infty$, and that the optimal policy derived from f^* represents the Stackelberg equilibrium between perception and policy modules.

Numerical Experiments

We evaluate our SPRIG agent on the BeamRider Atari environment, conducting experiments across five different random seeds over 10 million environment interactions. We considered BeamRider as an interesting challenge for our agent since temporal element and visual focus are important in this game. We utilize identical hyperparameters with both the baseline (PPO) and SPRIG except for the perception module configuration. The detailed perception module architecture specifications are provided in Appendix, Fig. 3. SPRIG achieves superior performance compared to the baseline PPO implementation as presented in Fig. 2, reaching approximately 850 points compared to PPO's 650, showing a clear advantage in the final performance. The learning process exhibits interesting dynamics: SPRIG demonstrates faster initial learning in the first 2 million steps, followed by a period of exploration and adjustment between 4-6 million steps, before stabilizing at a higher performance level. While the learning trajectory shows higher variance during the middle phase (as indicated by the purple-shaded region), this exploration appears beneficial for discovering better policies, ultimately leading to more robust performance. The baseline PPO, in contrast, shows more stable but conservative learning, with a steady but slower improvement curve and lower final performance. These results suggest that our game-theoretical framework effectively balances the exploration-exploitation trade-off while maintaining learning stability.

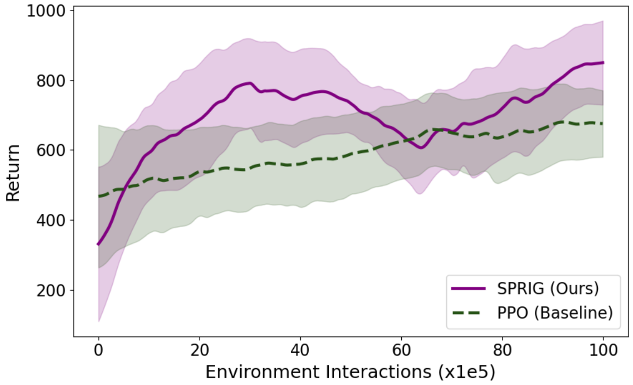

Figure 2: Return curves for SPRIG and baseline PPO on BeamRider. Results averaged across 5 seeds with shaded regions showing standard deviation.

Conclusions

In this paper, we presented SPRIG, a novel framework that formalizes perception-policy interaction in reinforcement learning through cooperative Stackelberg games. Our approach provides theoretical guarantees through a modified Bellman operator while demonstrating practical improvements in learning efficiency and stability. The preliminary results suggest that explicitly modeling module interaction through game theory could be a promising direction for improving single-agent reinforcement learning systems.

References

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Diuk, C.; Schapiro, A.; Córdova, N.; Ribas-Fernandes, J.; Niv, Y.; and Botvinick, M. 2013. Divide and conquer: hierarchical reinforcement learning and task decomposition in humans. *Computational and Robotic Models of the Hierarchical Organization of Behavior*, 271–291.

Goodale, M. A.; and Milner, A. D. 1992. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1): 20–25.

Huang, P.; Xu, M.; Fang, F.; and Zhao, D. 2022. Robust Reinforcement Learning as a Stackelberg Game via Adaptively-Regularized Adversarial Training. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 3099–3106. Main Track.

Ibarz, J.; Tan, J.; Finn, C.; Kalakrishnan, M.; Pastor, P.; and Levine, S. 2021. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40: 698 – 721.

Manchin, A.; Abbasnejad, E.; and Van Den Hengel, A. 2019. Reinforcement learning with attention that works: A self-supervised approach. In *International Conference on Neural Information Processing*, 223–230.

Mao, H.; Zhao, R.; Li, Z.; Xu, Z.; Chen, H.; Chen, Y.; Zhang, B.; Xiao, Z.; Zhang, J.; and Yin, J. 2024. PDiT: Interleaving Perception and Decision-making Transformers for Deep Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 1363–1371.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M. I.; and Abbeel, P. 2015. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *CoRR*, abs/1506.02438.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Sutton, R. S.; and Barto, A. G. 2018. Reinforcement learning: An introduction. *A Bradford Book*.

Zheng, L.; Fiez, T.; Alumbaugh, Z.; Chasnov, B.; and Ratliff, L. J. 2022. Stackelberg actor-critic: Game-theoretic reinforcement learning algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9217–9224.

Zhu, P.; Liu, S.; Jiang, T.; Liu, Y.; Zhuang, X.; and Zhang, Z. 2022. Autonomous Reinforcement Control of Visual Underwater Vehicles: Real-Time Experiments Using Computer Vision. *IEEE Transactions on Vehicular Technology*, 71(8): 8237–8250.

Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J. J.; Gupta, A.; Fei-Fei, L.; and Farhadi, A. 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3357–3364.

Appendix

Rollout Length	2048
Batch Size	64
Discount Factor (γ)	0.99
GAE Parameter (λ)	0.95
Learning Rate	1e-4
PPO Epochs	4
PPO Clip Range (ϵ)	0.2
Value Coefficient	0.5
Entropy Coefficient	0.01
Max Grad Norm (gradient clipping)	0.5
Perception Cost Weight (λ_c)	1e-4
Cooperation Weight (α_{coop})	0.7
Total Timesteps	1e7
Max Episode Length	10000

Table 1: Hyperparameters for SPRIG & PPO

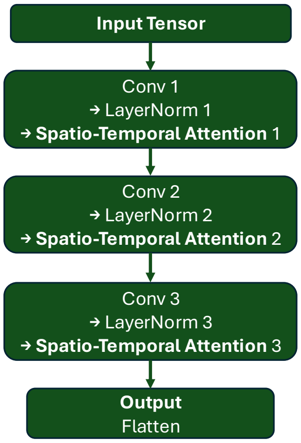
Input Tensor
Conv 1
→ LayerNorm 1
→ Spatio-Temporal Attention 1
Conv 2
→ LayerNorm 2
→ Spatio-Temporal Attention 2
Conv 3
→ LayerNorm 3
→ Spatio-Temporal Attention 3
Output
Flatten

Figure 3: Perception Module (θ)

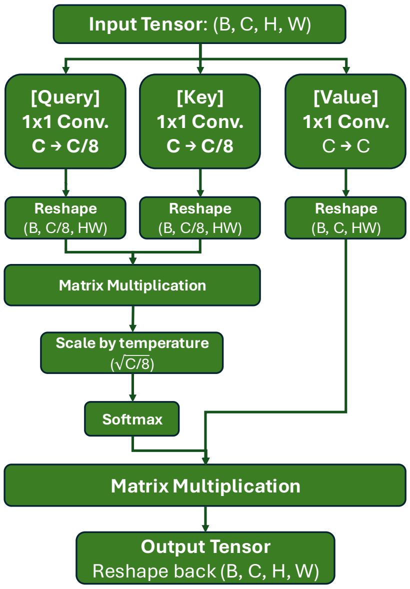

Figure 4: Spatio-Temporal Attention Block