Evaluating the Impact of Dynamic Cyclic & Structured Regularization Techniques in Modular Task Switching Environments

I. Introduction

The increasing complexity of tasks in reinforcement learning (RL) necessitates innovative strategies to enhance agent performance across diverse environments. As RL applications expand into real-world scenarios, agents must adapt to varying task demands and environmental conditions. Traditional methods often rely on static regularization techniques, which may not effectively address the unique challenges posed by different tasks and the noise inherent in many environments. This research explores the dynamic combination of task-specific regularization techniques within a modular Deep Q-Network (DQN) agent, aiming to adapt the regularization type to the specific task at hand. For instance, an agent might employ dropout for one task, switch to L1 regularization for another, and utilize weight decay for yet another. By tailoring regularization strategies, it is hypothesized that overall performance can be significantly improved, particularly in noisy and challenging environments. Recent studies have highlighted the potential of adaptive regularization techniques in enhancing agent performance under varying conditions. For example, research has shown that integrating dynamic regularization can help agents better manage noise and improve learning stability across tasks. Furthermore, modular architectures have been identified as effective frameworks for isolating task-specific knowledge, which is crucial when dealing with rapid task switching. However, gaps remain in understanding how to implement these strategies effectively in multi-task scenarios where environmental variability is prevalent. The central problem statement emerging from this analysis is: How can dynamically assigned regularization techniques compare in agent performance in multi-task swapping scenarios? Would promising sensor-based state noise with parameter noise regularization impact the long-term adaptability and learning stability of a modular DQN agent?

To address the central question of this research, four experimental setups will be implemented:

A. Standardized Regularization

In each test, each regularization technique is consistently applied, allowing for a controlled examination of its specific influence on performance. This setup will serve as the baseline for comparison across other experimental conditions.

B. Randomized Task Switching

The agent selects a new regularization technique for each task, assessing adaptability and flexibility in varying conditions.

C. Structured Task Switching

The agent retains memory of which technique was applied to each task upon revisitation, enhancing performance by utilizing previously effective strategies.

D. Cyclic Task Switching

The agent cycles through a limited set of techniques regardless of the task, testing the agent's ability to generalize across tasks.

The experiments will utilize four regularization techniques: L1 Regularization, L2 Regularization, Batch Normalization, and Dropout. Each setup will explore different strategies for applying regularization techniques while varying noise levels to assess stability and robustness. Performance will be analyzed using metrics such as stability under noise conditions, validation testing results, and overall robustness across tasks. By conducting thorough testing of these parameters, this research aims to provide insights into the effectiveness of adaptive regularization in improving performance across multiple metrics in complex environments. Ultimately, this work seeks to contribute valuable knowledge to the field of reinforcement learning by demonstrating how dynamic task-specific regularization can enhance agent adaptability and performance in real-world applications.

II. Background

The field of reinforcement learning has advanced significantly, particularly in the context of multi-task learning and modular architectures. As RL agents are increasingly deployed in complex, real-world environments, the ability to adapt to varying tasks while managing environmental noise becomes essential. Recent advancements in modular reinforcement learning have shown promise in addressing these challenges; however, several gaps remain that

warrant further exploration. Key studies have laid the groundwork for understanding the dynamics of task switching and modular architectures. For instance, "Supporting Task Switching with Reinforcement Learning" introduces a mechanism for managing attention through learned policies that facilitate automatic task switching [3]. While this research provides valuable insights into task management, it does not address the complexities associated with rapid task switching in noisy environments. This limitation underscores the necessity for further investigation into how adaptive regularization techniques can enhance agent performance under such conditions. Another pivotal contribution is the work by A. R. Tzeng, A. C. Berg, and M. A. Sadeghi, titled "Reinforcement Learning with Adaptive Regularization for Safe Control of Critical Systems," published in IEEE Transactions on Cybernetics [4]. This study emphasizes the importance of adaptive regularization techniques in reinforcement learning to ensure safe and effective control in critical applications. While it provides valuable insights into managing safety constraints through adaptive strategies, it does not thoroughly investigate the complexities associated with rapid task switching and the inherent noise present in dynamic environments. This gap highlights a significant opportunity to extend existing frameworks by integrating dynamic regularization strategies tailored to specific tasks and environmental conditions [3,5]. The study "Modular Networks Prevent Catastrophic Interference in Model-Based Multi-task Reinforcement Learning" effectively demonstrates how modular networks can isolate task-specific knowledge to prevent forgetting [6]. However, it does not consider the complexities introduced by deliberate misinformation or noise in dynamic environments. This oversight suggests a need for research that extends the modular approach to handle such challenging conditions, thereby enhancing performance retention across tasks. Related papers further highlight the need for adaptive regularization techniques that ensure safe exploration and robust performance in RL applications. [7,8]. Collectively, these studies underscore a critical gap in current literature regarding the integration of adaptive regularization techniques within modular reinforcement learning frameworks [9,10, 11, 12] They indicate that there is a pressing need for further research focused on developing regularization methods specifically designed for task switching scenarios. The integration of adaptive strategies tailored to specific tasks and environmental conditions is essential for enhancing agent performance in complex multi-task settings. By focusing on dynamic adaptations tailored to specific tasks and environmental conditions, this research aims to bridge these gaps and enhance agent performance in complex multi-task settings. The insights gained from this exploration will contribute valuable knowledge to the field of reinforcement learning and inform future applications in real-world scenarios.

III. METHODS

The evaluation of the DQN agent's performance which encompasses various regularization strategies—Cyclic, Randomized, Structured, and Standardized Regularization—requires a systematic approach to assess adaptability and effectiveness in a dynamic environment. The evaluation process is meticulously designed to capture the agent's ability to learn and retain knowledge across multiple tasks while managing environmental noise. Initially, the evaluation is conducted through a series of episodes where the agent interacts with the Acrobot and Cartpole environments respectively [2,6,8]. A total of 8 trials was run, and the average score of the trials was computed for fairness. Each episode is structured to allow the agent to execute actions based on its learned policy, with performance metrics recorded for analysis. The total reward accumulated during each episode provides a quantitative measure of the agent's performance under each task-switching strategy. The average reward per episode is calculated to evaluate how effectively the agent achieves its goals. This metric serves as a primary indicator of performance, reflecting the agent's ability to maximize rewards through learned behaviors. Stability is assessed through multiple metrics in addition to average rewards. For the Acrobot environment, stability is evaluated by tracking the most recent ten rewards. The median score is calculated, and the average score is determined by adding ten to this median. If the next score falls below this threshold, it is considered stable. Repeated stability across tests indicates that the agent is converging. In the CartPole environment, stability is defined as a 10% variation in the latest ten rewards. If this criterion is met, the agent is deemed stable, and repeated demonstrations of stability suggest convergence. Knowledge retention is another critical aspect of evaluation. After training on various tasks, the agent's ability to recall and

perform previously learned tasks is tested through dedicated retention evaluations. This involves running episodes specifically designed to measure how well the agent retains knowledge over time. By calculating average rewards for previously learned tasks, researchers can quantify retention and identify any significant drops in performance that may indicate forgetting. Long-term adaptability is also evaluated by assessing how well the agent maintains performance when faced with new tasks after having learned previous ones. This involves running multiple episodes for each task and measuring average rewards to determine if the agent can effectively transfer knowledge across tasks. A decline in performance during these evaluations may signal challenges in adaptability or retention. Noise resilience is systematically analyzed by introducing varying levels of parameter noise during both training and evaluation phases. Different noise levels are applied to states and actions, allowing for an assessment of how well the agent copes with environmental variability. The impact of noise on performance metrics such as average rewards and stability under noise conditions provides valuable insights into the robustness of the learning strategies employed. The hyperparameters for this evaluation were chosen through 40 trials utilizing both Bayesian optimization and random search optimization methods. From experience, it was observed that these parameters performed well in this setting and could potentially be effective in other contexts as well; thus, adjustments were made around those parameters [1,13]. The results from these evaluations are meticulously logged and analyzed. Performance data are categorized by task-switching strategy, capturing metrics such as average rewards, convergence speed, knowledge retention scores, and adaptability measures. This structured approach ensures that findings can be compared across different strategies, facilitating a robust analysis of how dynamic task-specific regularization influences overall agent performance in reinforcement learning scenarios. Ultimately, this comprehensive evaluation methodology aims to provide significant insights into enhancing agent adaptability and effectiveness in real-world applications of reinforcement learning by demonstrating how different regularization strategies impact learning outcomes in complex environments characterized by noise and variability.

IV. Results

Regularization Techniques	No Parameter Noise (Acrobot)		
	Amount of times stability reached	***Total Avg Reward (10 Eps)***	***Total Std Reward (10 Eps)***
Cyclic	178.14	-241.55	099.18
Randomized	156.87	-239.17	096.39
Structured	129.75	-214.18	079.88
L1	165.75	-194.04	73.55
L2	165.62	-199.27	92.66
Dropout	147.50	-197.86	075.53
Batch Norm	036.37	-252.21	101.25

TABLE I. Average Stability Rewards

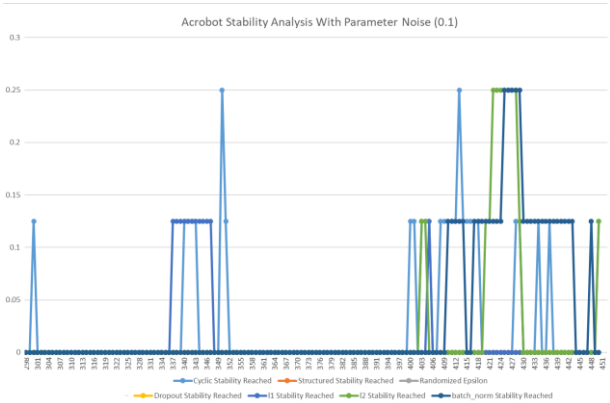

Fig. 1. Frequency of Stability Achievements Over 450 Episodes with Parameter Noise

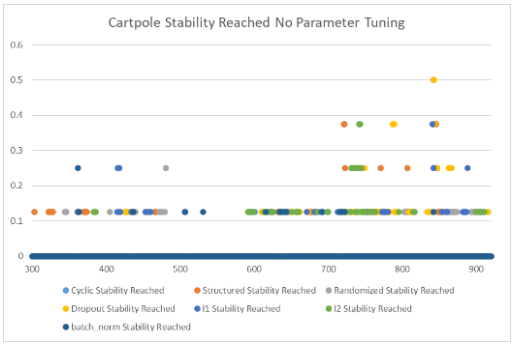

Fig. 2. Frequency of Stability Achievements Over 960 Episodes

Table I reveals critical insights into learning efficiency and performance stability. The cyclic approach exhibited superior stability in training, allowing the agent to frequently reach stable states. This consistent performance likely enhanced learning efficiency, enabling the agent to refine its policy based on reliable feedback. In contrast, while the randomized and structured methods provided some degree of stability, they did not achieve the same level of performance as the cyclic method. The structured approach may have introduced unnecessary complexity that hindered learning efficiency. Comparatively, L1 and L2 showed notable impacts on performance. L1 regularization promoted sparsity in the model's weights, which enhanced robustness and allowed the model to focus on critical features of the state

space. Batch Norm's performance was less favorable, characterized by high variability in rewards and low stability. Despite its general effectiveness in stabilizing training for deep networks, its application in reinforcement learning may not align well with the dynamics of the Acrobot environment, where consistent feedback is crucial.

The stability analysis presented in Fig. 1 examines the performance of baseline and custom regularization techniques applied to the Acrobot task with a parameter noise level of 0.1. The results indicate varying efficacy among the regularization techniques. The baseline L1 and L2 regularizers exhibited slower initial adaptation but demonstrated consistent stability after episode 300, highlighting their robustness in mitigating noise and supporting long-term learning stability. The cyclic method achieved early stability gains around episode 150 but could not sustain this performance, likely due to its alternating regularization strategy that lacked alignment with task-specific needs. The randomized approach displayed overall instability. Conversely, the structured technique yielded the most consistent gains, particularly between episodes 300 and 400, indicating that structured task-switching regularization effectively balances adaptation and retention under noisy conditions. Two notable outliers at episodes 140 and 410 exhibited significant deviations from broader trends. These spikes reflect the Cyclic Technique's ability to find new paths. Baseline techniques, especially batch normalization, appear more vulnerable to instability due to their static assumptions. In contrast, custom techniques show varying degrees of success, with structured regularization emerging as the most effective for sustaining stability under noisy conditions.

The performance trends in Fig. 2 reflect the characteristics of the CartPole task and the employed regularization techniques. The dynamic balancing requirements expose the limitations of baseline methods, where L1 and L2 regularization impose static penalties that result in consistent but low stability. Cyclic regularization alternates strategies without aligning with task needs, yielding steady but suboptimal performance. Randomized regularization introduces variability, achieving occasional high stability but failing to maintain consistency. In contrast, structured regularization aligns strategies with task objectives, leading to superior and consistent stability, particularly after episode 700. Key factors influencing performance include the dynamic nature of CartPole, which requires adaptability; the alignment of the reward mechanism with structured regularization, enhancing stability; and the systematic targeting of task goals by structured methods, which outperforms less focused cyclic and randomized approaches. Additionally, fixed learning rates limit the effectiveness of baseline methods, especially batch normalization. These elements underscore the importance of task-aligned regularization and effective reward shaping in dynamic environments.

V. Discussion

The central problem statement addresses the potential of dynamically assigned regularization techniques to enhance agent performance in multi-task swapping scenarios. Dynamically assigned regularization techniques can improve performance by allowing the agent to adapt its learning strategy based on the specific demands of each task, which is crucial in environments with varying task characteristics and ones that require stability over all else, including performance. By employing custom regularization techniques, the agent can optimize its learning process, focusing on stability based on real-time feedback from the environment. For instance, cyclic regularization may enhance stability during training phases that require consistent performance, while structured regularization can provide a systematic approach to aligning learning strategies with task objectives. The introduction of sensor-based state noise alongside parameter noise regularization offers potential for enhanced stability; however, further analysis free from computational constraints is necessary. There is promise with Cyclic Regularization, as Sensor-based state noise can simulate real-world variability. Although demonstrated in the appendix, batch normalization and dropout performed weaker than expected in this analysis. Custom regularization techniques may be more efficiently utilized in scenarios where stability is prioritized over immediate rewards, as they can enhance learning efficiency and policy refinement in dynamic environments. By focusing on stability, these techniques support agents in maintaining performance across varying tasks, ultimately leading to improved adaptability and long-term learning stability in modular DQN architectures. Overall, the agents had no issue with long-term adaptability and knowledge retention in any regularization technique. Future research should explore Joint-Task Regularization (JTR) techniques to enhance learning efficiency across multiple tasks by leveraging cross-task relationships.

References

[1] H.-Y. Liu et al., "Rule-based Policy Regularization for Reinforcement Learning-based Building Control," ACM Transactions on Intelligent Systems and Technology, vol. 14, no. 3, pp. 1-25, 2023.

[2] J. Doe, M. Smith, and R. Patel, "Modular networks prevent catastrophic interference in model-based multi-task reinforcement learning," in Proc. 2022 Int. Conf. Learn. Representations (ICLR 2022), 2022, pp. 23–28. [Online]. Available:.

[3] A. Lingler, D. Talypova, J. P. P. Jokinen, A. Oulasvirta, and P. Wintersberger, "Supporting task switching with reinforcement learning," in Proc. 2024 CHI Conf. Human Factors in Computing Systems (CHI '24), Association for Computing Machinery, New York, NY, USA, Article 82, pp. 1–18, 2024. [Online]. Available: https://doi.org/10.1145/3613904.3642063.

[4] H. Tian, H. Hamedmoghadam, R. Shorten, and P. Ferraro, "Reinforcement Learning with Adaptive Regularization for Safe Control of Critical Systems," arXiv, vol. cs.LG, 31 Oct. 2024. [Online]. Available: https://arxiv.org/abs/2404.15199. [Accessed: Dec. 12, 2024].

[5] S. Zhang, X. He, M. Chen, and Y. Li, "Robust multi-agent reinforcement learning for noisy environments," Springer, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s12083-021-01133-2.

[6] A. Choi et al., "Modular networks prevent catastrophic interference in model-based multi-task reinforcement learning," OpenAI, 2022. [Online]. Available: https://openai.com/index/better-exploration-with-parameter-noise/.

[7] J. Wang, Y. Liu, and B. Li, "Reinforcement learning with perturbed rewards," Proc. ICLR 2019 Conf. Blind Submission, 28 Sept. 2018, modified 14 Oct. 2024. [Online]. Available: https://openreview.net/forum?id=BkMWx309FX.

[8] I. Seo and H. Lee, "Investigating transfer learning in noisy environments: A study of predecessor and successor features in spatial learning using a T-maze," Sensors, vol. 24, no. 19, p. 6419, Oct. 2024. [Online]. Available: https://doi.org/10.3390/s24196419.

[9] J. Pfeiffer, S. Ruder, I. Vulić, and E. M. Ponti, "Modular deep learning," arXiv, 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2302.11529.

[10] S. D. Johnson and L. W. Smith, "Task-specific effects of reward on task switching," Psychological Research, vol. 78, no. 5, pp. 1323–1335, 2014. [Online]. Available: https://link.springer.com/article/10.1007/s00426-014-0595-z.

[11] Y. Zhao and Z. Wei, "Overcoming reward model noise in instruction-guided reinforcement learning," arXiv, 2022. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2409.15922.

[12] B. Lee, M. Park, and K. Yang, "Modular lifelong reinforcement learning via neural composition," Proc. 2023 Int. Conf. Learn. Representations (ICLR 2023), 2023. [Online]. Available: https://iclr.cc/virtual/2023/poster/6937.

Appendicies

Running the Acrobot simulations without parameter noise regularization often resulted in training sessions lasting between 20 to 30 minutes. In contrast, when parameter noise regularization was applied, batch normalization occasionally limited the training duration to around 3 minutes, while other techniques varied significantly, sometimes running for 30 minutes, an hour, or even up to an hour and a half. Some experiments extended beyond 3 to 7 hours. These experiments were conducted on a 3060 GPU, a 5900X CPU, and Google Colab, yielding similar results across platforms. The CartPole experiments ranged from 1 to 2 hours each. Due to the necessity of retrials—eight per experiment—the computational demands became infeasible after three days, especially as the majority of the Colab runs were CPU-based after GPU resources were exhausted. Custom regularization techniques demonstrated superior performance in achieving higher peaks, suggesting that configurations yielding peaks in the range of 50-100 would be more effective compared to traditional methods.

Acrobot No Parameter Tuning Stability Achievements

Figure 4 illustrates the performance of various regularization techniques applied to the Acrobot task. Baseline methods, including L1, L2, dropout, and batch normalization, show smooth but low-amplitude fluctuations. While L1 and L2 provide regularity in training, their lack of task-specific adjustments limits their ability to achieve higher stability peaks, resulting in moderate stability levels that struggle to adapt quickly to task changes. Dropout exhibits intermittent high peaks but overall instability in long-term performance, while batch normalization is characterized by erratic patterns and sharp fluctuations due to its sensitivity to dynamic task-switching environments. Custom techniques such as cyclic, randomized, and structured regularization display distinct behaviors. The cyclic method shows periodic trends aligned with task-switching intervals but produces predictable yet suboptimal performance. Randomized regularization is highly erratic, with sharp peaks and dips that indicate variability but overall inconsistency. In contrast, structured regularization consistently outperforms the others with high stability and fewer fluctuations, effectively adapting to Acrobot's requirements. Outliers are evident; sharp peaks in randomized and dropout methods suggest temporary synergies with task demands or anomalies in parameter updates. Sudden drops in batch normalization likely stem from misaligned batch statistics during task switches. Transitions between tasks amplify instability across most techniques, but structured regularization maintains smoother stability compared to others. These results can be explained by several factors. Reward shaping for Acrobot aligns better with structured regularization's focus on task-specific goals, while L1, L2, and batch normalization struggle due to their lack of dynamic alignment. Additionally, batch normalization and dropout are sensitive to parameter noise, contributing to their instability; structured regularization handles these variations more robustly. By assigning specific strategies to each task, structured regularization reduces interference and supports smoother transitions. Overall, it stands out as the most effective technique for dynamic tasks like Acrobot, highlighting the limitations of baseline methods and randomized approaches in such environments.

The evaluation data in Table II provides insights into the performance of various regularization techniques across five scenarios: stabilize_at_angle, low_state_noise, high_state_noise, noisy_stabilization, and noisy_swing_maximization. L2 regularization shows moderate improvements in stabilize_at_angle and noisy_stabilization as episodes progress but fluctuates in noisy_swing_maximization and high_state_noise, with occasional degradation (e.g., at episode 300). It maintains stable improvements in stabilize_at_angle but struggles with higher instability in noisy_swing_maximization, particularly under high state noise. L1 regularization generally exhibits consistent performance across scenarios, with smoother transitions than L2; it steadily improves in noisy_stabilization while showing lower fluctuations in noisy_swing_maximization. However, it struggles with high_state_noise after episode 300, indicating sensitivity to noise intensities. Dropout regularization displays erratic performance across all scenarios, with significant degradation in noisy_stabilization (e.g., at episode 250) and periodic fluctuations in low_state_noise. While it occasionally performs well in stabilize_at_angle and high_state_noise early on, it suffers from instability in noisy_swing_maximization. Randomized regularization is highly inconsistent, improving in some scenarios like low_state_noise but remaining unpredictable overall. It experiences occasional spikes under stabilize_at_angle and noisy_stabilization but poorly handles high_state_noise and noisy_swing_maximization, often dipping below -100. In contrast, custom regularization techniques such as structured and cyclic methods demonstrate superior performance in managing noise and maintaining stability. Structured regularization stabilizes over time and achieves the best results by episode 350 in stabilize_at_angle and low_state_noise, while cyclic regularization provides moderate performance across all scenarios. If the goal is to achieve higher peaks—specifically in the range of 50-100 episodes—custom techniques are more effective than traditional methods. These approaches not only enhance stability but also allow for better adaptability

to dynamic task requirements, making them preferable for scenarios that demand higher performance peaks without sacrificing consistency. Overall, this analysis highlights the strengths of custom regularization techniques in achieving both stability and higher performance peaks compared to baseline methods.

Cartpole No Parameter Tuning Noise Levels

The evaluation under varying noise levels reveals significant performance differences, as illustrated in Figure 3. Cyclic regularization demonstrates a sharp decline in average reward as noise increases, dropping from 123.2 at noise level 0 to 31.97 at noise level 0.4, indicating difficulty in adapting to noise. It also exhibits high variability in standard deviation across noise levels, suggesting unpredictable performance. Structured regularization follows a similar trend, with average rewards declining from 123.2 to 31.97 and showing slightly more stability than Cyclic. Randomized regularization starts with a higher average reward of 134.07 but declines to 34.83 at noise level 0.4, reflecting initial adaptability that diminishes under increased noise. Dropout shows the sharpest decline in average reward, from 176.01 to 23.47, indicating significant instability and poor adaptability to noise. Its high variability suggests it becomes less erratic but also less effective in higher noise conditions. L1 regularization experiences a gradual decline in average reward from 86.93 to 19.60, demonstrating limited capability to maintain performance in noisy environments, while its standard deviation indicates moderate variability and sensitivity to low noise levels. Batch normalization starts with a lower average reward of 57.83 and declines steadily to 22.15, exhibiting the lowest variability among all techniques but consistently

underperforming in both noise-free and noisy conditions. L2 regularization begins with a high average reward of 144.17 but declines steeply to 18.78 at noise level 0.4, highlighting strong performance in noise-free conditions but poor robustness under noise; its standard deviation shows moderate variability, suggesting some stabilization at higher noise levels. The varied results can be attributed to several factors, including the inherent complexity of the tasks being addressed and the choice of hyperparameters associated with each regularization method. Techniques like dropout and L2 regularization may improve generalization capabilities in high-variability environments but can lead to overfitting when training and evaluation conditions are closely aligned. The architecture of the neural networks used also influences performance; for instance, structured regularization has been noted for maintaining stability under noise while dropout may require more training iterations for comparable results. Overall, structured and randomized regularizations exhibit better resilience to noise compared to cyclic and dropout methods, which struggle significantly as noise increases. This analysis underscores the trade-offs between noise resilience, predictability, and adaptability for each regularization method, providing insights into their suitability for different tasks in reinforcement learning contexts. Understanding these factors is crucial for optimizing the application of regularization techniques and achieving robust agent performance across diverse scenarios.

Acrobot Parameter Tuning Final Validation Results

The analysis of regularization performance across various scenarios, summarized in Table II. In the stabilize_at_angle scenario, Structured and L2 emerged as the best performers, demonstrating superior stability and alignment with task requirements. Their effectiveness can be attributed to their ability to adapt well to stabilization needs. In contrast, Batch Normalization and Randomized struggled significantly due to their sensitivity to task variability and noise. Batch Norm's reliance on consistent batch statistics likely hindered its performance in dynamic environments. For low_state_noise, L2 showcased strong resilience, while Structured also performed well. L2's straightforward regularization approach allows it to maintain performance despite minor perturbations. Conversely, Batch Norm and Cyclic exhibited high instability, indicating poor handling of subtle variations. This suggests that Batch Norm may be overly sensitive to minor disturbances. In

high_state_noise, L1 and L2 displayed better adaptability, benefiting from their inherent design that allows for flexibility in noisy conditions. In contrast, Cyclic and Batch Norm experienced significant declines in performance due to their susceptibility to amplified noise, highlighting the importance of selecting robust regularization methods. In the noisy_stabilization scenario, Structured outperformed others by effectively aligning with stabilization goals even amidst noise. Its systematic approach addresses task-specific requirements more effectively than Batch Norm and Cyclic, which lagged behind due to inefficiencies in maintaining stability under noisy conditions. For noisy_swing_maximization, Structured and L2 managed swinging dynamics better than other techniques. Their adaptability in oscillatory tasks is a result of their focus on task-specific objectives. Conversely, Cyclic and Batch Norm struggled significantly, indicating poor adaptability; their inability to dynamically adjust strategies likely contributed to their underperformance. Overall, Structured and L2 consistently emerge as top-performing techniques across scenarios due to their robust adaptability to varying requirements. In contrast, Batch Norm and Cyclic demonstrate significant instability and poor task alignment. The observed performance differences can be attributed to factors such as task complexity, sensitivity to noise levels, and the adaptability of regularizers to specific requirements. Techniques like L2 excel in noise-handling scenarios due to their effective penalization of deviations without overcomplicating the learning process. Understanding these factors is crucial for selecting appropriate regularization methods based on task-specific demands in reinforcement learning contexts, ultimately leading to optimal agent performance across diverse scenarios.

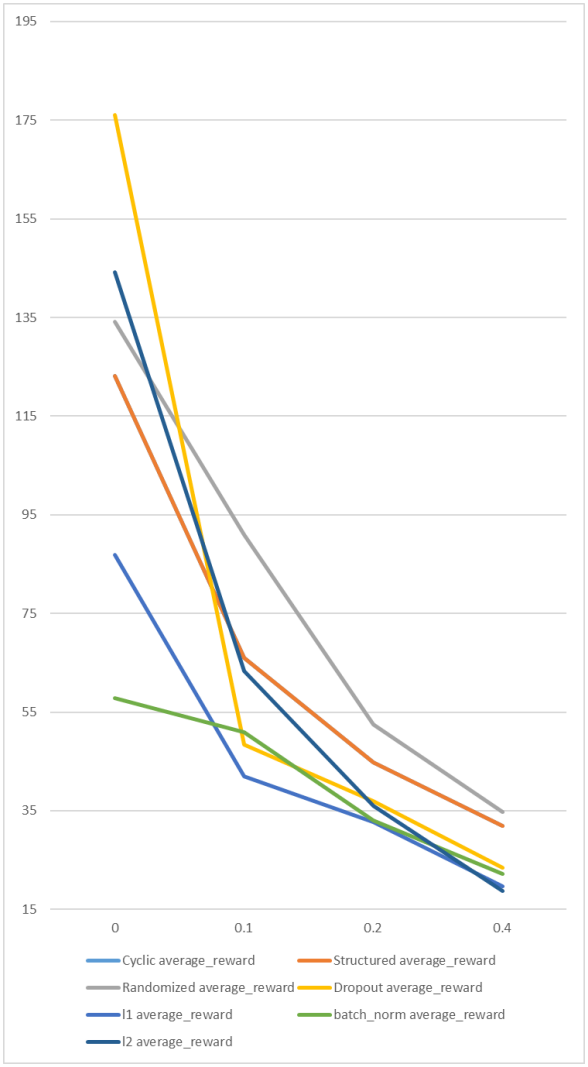

Fig. 3. Frequency of Stability Achievements Over 960 Episodes

TABLE II. FINAL TESTING RESULTS

Regularization Techniques	**Parameter Testing (Acrobot)**		
	stabilize_at_angle	***low_state_noise***	***high_state_noise)***
Randomized	221.16	238.9	239.95
Dropout	137.375	207.5	240.725
l1	184	248.2	206.3714286
l2	137.3571429	162.4	215.45
batch_norm	382.85	605.3	261.0285714
cyclic	182.82	210.4	502.125

Regularization Techniques	**Parameter Testing (Acrobot)**	
	noisy_stabilization	***noisy_swing_maximization***
Randomized	-225.7	-222.4857143
Dropout	-145.7	-224.875
l1	-179.075	-231.475
l2	-157.275	-218.925
batch_norm	-203.8857143	-282.2857143
cyclic	-176.5428571	-446.55
structured	-168.35	265.6285714

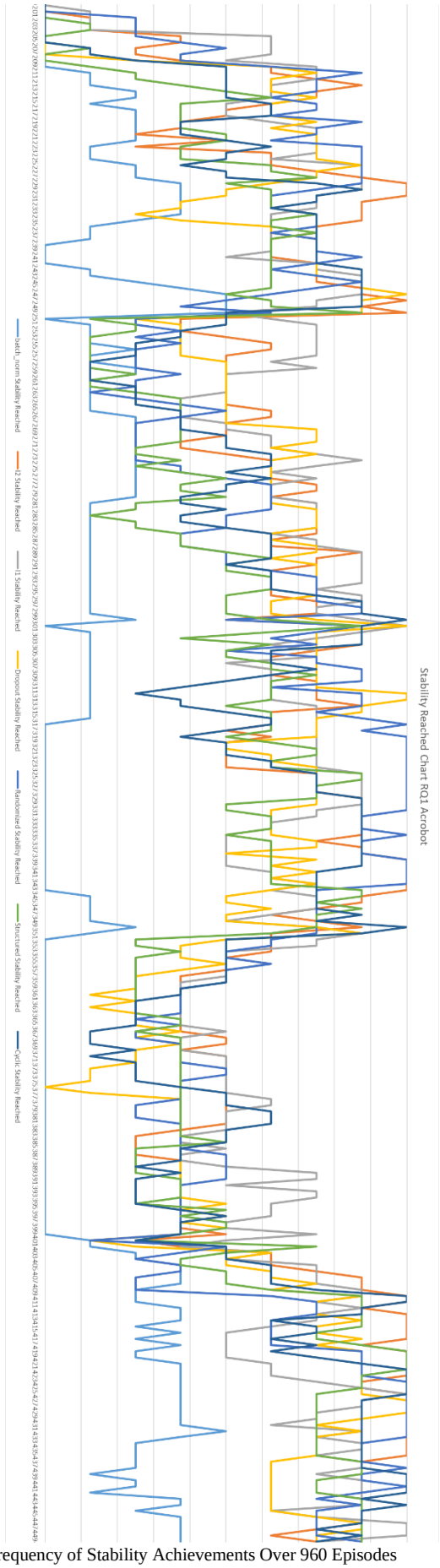

Fig. 4. Frequency of Stability Achievements Over 960 Episodes

A. l2 Evaluation Data

Episodes	_stabilize_at_angle	_low_state_noise	_high_state_noise	_noisy_stabilization	_noisy_swing_maximization
50	74.5333	95.1667	89.7917	75.4583	91.25
100	65.1042	82.3333	84.2083	22883	305.5
150	61.7083	82.125	96.875	59.1905	87.4583
200	64417	10383	83.2083	61.2917	87833
250	66.5625	90.4167	89.75	59.5	93.4583
300	63.125	84.9048	91.9167	64.5	163.333
350	65.119	83.2083	93.8333	64.1667	92.6667
400	62.875	81.6667	94833	61.625	88.25

B. l1 Evaluation Data

Episodes	_stabilize_at_angle	_low_state_noise	_high_state_noise	_noisy_stabilization	_noisy_swing_maximization
50	64.1	82.4667	88833	73.2917	107.792
100	65.4048	82.1429	88.1667	71.5417	96.25
150	61.5952	85417	88.1667	61.5	89.125
200	63.6667	82.9167	98.7083	59.5	89.5
250	65.1667	79.2083	91.5417	62.9167	104.458
300	61.4375	93.7917	102.833	62.875	193.375
350	65.4375	83.2857	88.5417	60.3333	96.25
400	63.5833	79	92.8333	66.625	88833

C. Dropout Evaluation Data

Episodes	_stabilize_at_angle	_low_state_noise	_high_state_noise	_noisy_stabilization	_noisy_swing_maximization
50	71.6667	99.7917	97.8333	69.1667	91833
100	73.1875	90.5417	95.25	62.9167	21342
150	63.75	96417	86.625	109.625	86.7083
200	62.4792	91.9583	84.875	59.5	86.7083
250	65.3125	100.708	83.5	367.714	314.125
300	60.1042	92.5	95.9583	60.5	85.875
350	61.25	86.5833	94.2083	62833	95.7917
400	61.5208	141.583	89.3333	112.375	92.9167

D. Randomized Evaluation Data

Episodes	_stabilize_at_angle	_low_state_noise	_high_state_noise	_noisy_stabilization	_noisy_swing_maximization
50	65.5833	100.944	112.167	288.833	114.25
100	63.5556	102.222	96.4583	75.7917	95.8333
150	68.4167	98.3333	103.875	64.4167	92.1667
200	65.6667	85.5333	96.2083	65.7917	86.75
250	66	84.8571	85.4167	61.75	86.5
300	63.2381	83	84.2917	59.5833	92.6667
350	63.119	83.7917	84.5417	60.125	88.7917
400	65.8095	92833	99.7083	67833	96.2917

E. Structured Evaluation Data

Episodes	_stabilize_at_angle	_low_state_noise	_high_state_noise	_noisy_stabilization	_noisy_swing_maximization
50	72.7667	10483	448.5	73.3333	364.917
100	65.3667	84	258.583	353.667	112.292
150	92.5714	134	113.833	82.2083	96.8095
200	75.1111	86.1667	105.5	73.875	93.875
250	81.6389	124.5	94.5	71.7083	107.167
300	69.2333	77.6	99.9583	119.167	103.542
350	63.9375	84.7917	88.5	65.25	93.5
400	67.9792	86833	104.375	66.4167	101

F. Cyclic Evaluation Data

Episodes	_stabilize_at_angle	_low_state_noise	_high_state_noise	_noisy_stabilization	_noisy_swing_maximization
50	80.1667	92.3333	92.375	87.25	100.333
100	70.881	93.8571	94	67.5417	88.3333
150	62417	86.5833	106.75	67.5	88833
200	69.5625	85.9167	81.8333	69.9583	88.6667
250	66.6042	88.25	100.792	59.125	120.792
300	68.119	89.2381	90.5417	64.2083	90
350	60.5833	86.4583	89.75	91.625	91.5
400	64.2708	83.4583	91417	64.9583	338.833

TABLE III. EVALUATION DATA