Uncovering RL Integration in SSL Loss: Objective-Specific Implications for Data-Efficient RL

Ömer Veysel Çağatan
Department of Computer Engineering
Koç University
Sarıyer, İstanbul 34450
ocagatan19@ku.edu.tr

Barış Akgün
Department of Computer Engineering
Koç University
Sarıyer, İstanbul 34450
baakgun@ku.edu.tr

Abstract

In this study, we examine the impact of different SSL objectives within the Self Predictive Representations (SPR) [35] framework. Specifically, we explore SSL modifications like terminal state masking and prioritized replay weighting, which were not explicitly discussed in the original framework. These modifications are RL-specific but are not applicable to all RL algorithms. As such, it is of interest to gauge their impact on performance and look at other SSL objectives. We evaluate six SPR variants on the Atari 100k benchmark, including versions without these modifications, as well as others incorporating feature decorrelation methods like Barlow Twins and VICReg, which cannot accommodate these specific adjustments. Additionally, we assess the performance of these objectives on the DeepMind Control Suite, where the environment does not feature these modifications. Our findings show that the SSL modifications within SPR significantly influence performance, underscoring the critical importance of both the SSL objective selection and its accompanying modifications in data-efficient and self-predictive reinforcement learning.

1 Introduction

Self-supervised learning (SSL) has become increasingly popular in data-efficient RL due to its benefits in enhancing both efficiency and performance [37, 48, 16, 41, 43, 25, 5]. However, the application of SSL methods are often problem/domain specific (see in Appx. C for a more detailed discussion) to maximize the performance of RL agents. Although this approach is rational given the nature of these methods, it raises questions about generalization and transferability.

One of the main challenges in Deep RL is understanding what drives performance improvements, whether through hyperparameter tuning or new algorithmic methods [31]. Although algorithmic innovations are often well-documented, the lack of clarity around hyperparameter selection can be problematic. In our study of different SSL objectives within the Self Predictive Representations (SPR) [35] framework, we noticed that the SSL loss used in SPR differs from what is described in the original literature. Unlike typical SSL methods in RL, which follow vision pretraining approaches [9] and directly combine SSL and RL losses [41], SPR modifies the SSL loss before merging it with the RL objective. This raises a critical question: How do these modifications affect the performance of SSL objectives, and can they be effectively applied to other SSL techniques in the RL domain?

In this spirit, we use SPR as the main agent that we thoroughly describe in Appx. A.3. SPR uses the BYOL/SimSiam [14, 10] auxiliary objective and includes two algorithm-specific adjustments to the SSL objective; (i) masking SSL loss with boolean non-terminal state matrix and (ii) applying prioritized replay weighting to the batch loss. These modifications are feasible in the context of SPR due to the sample-wise nature of the objective it uses and the target domain.

NeurIPS 2024 Workshop: Self-Supervised Learning - Theory and Practice.

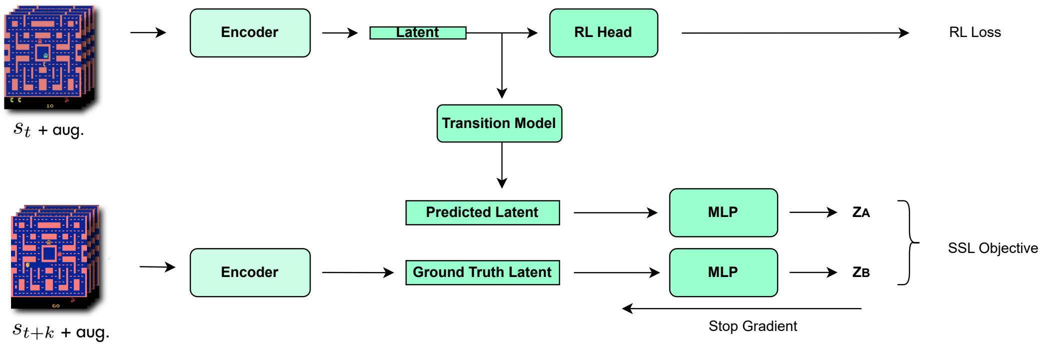

Figure 1: General flow diagram of SPR based methods. An encoder is used to create representations used for reinforcement learning and predicting future representations via a transition model and ground truth representations are created by the same encoder. MLPs differ when the predictor layer is used as in the case of BYOL/SimSiam. While we show the k^{th} step here, the actual loss computation covers steps 1 to K. The SSL objective and RL loss changes between specific methods.

On the other hand, a multitude of novel self-supervised representation learning objectives [50, 4, 32, 6] has been proposed which has been shown to excel beyond image pretraining [24, 13, 52, 53]. Since these objectives are based on feature decorrelation, they do not inherently support the modifications used in SPR, as described in Appx. B. Consequently, another important question is how other objectives perform relative to the original SPR without SSL modifications. This is significant because the information required to modify SSL objectives may not always be available in the environment.

Therefore, we investigate an additional six SPR models, along with the original SPR: (i) SPR-Naked, featuring no modifications, (ii) SPR-Naked+Non, incorporating terminal masking, (iii) SPR-Naked+Prio, integrating prioritized replay weighting, (iv) SPR-Barlow, (v) SPR-VICReg-High, characterized by a high covariance weight, and (vi) SPR-VICReg-Low, characterized by a low covariance weight.

Even though there are newly proposed SSL objectives [39, 51, 46], it is impractical to include all objectives in experiments due to limited computational resources and the need to prioritize rigorous evaluation to draw precise conclusions however, we attempt to cover the two main families of SSL methods within SPR. The first is self-distillation, represented by BYOL [14] or SimSiam [10], which are already incorporated into SPR. The second family includes canonical correlation methods, such as VICReg and Barlow. Another category is Deep Metric Learning, which includes contrastive learning variants [3]. However, we do not separately test contrastive objectives, as they have already been shown to be ineffective in SPR [35].

With a central focus on data efficiency, our primary evaluation of these models is conducted on Atari 100k [20]. Our results show that the RL specific modifications to the SPR's SSL objective have significant impact on performance and using a pure feature decorrelation method like Barlow Twins perform on par. We further evaluate these SSL objectives in the DeepMind Control Suite 100k [42] with appropriate modifications to handle continuous actions. Notably, VICReg emerges as the top performer, surpassing even PlayVirtual [49], which features a more complex transition model than SPR. Overall, our findings underscore the importance of SSL objectives in data-efficient RL, revealing variations in performance depending on the chosen objective and the environment, and suggest that pure SSL objectives may mitigate the need for problem specific modifications.

2 Analysis

2.1 Atari 100k

Figure 2 shows the performance of the seven agents in the Atari 100k benchmark, calculated using the rliable framework [1]. The individual game performances are given in Appx. E and we describe evaluation setup in Appx. D.

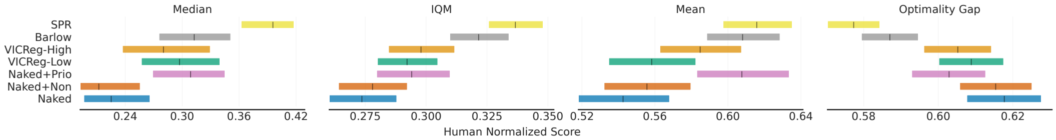

Figure 2: Mean, median, interquartile mean human normalized scores and optimality gap (lower is better) computed with stratified bootstrap confidence intervals in Atari 100k. 50 runs for SPR-Barlow, SPR-VICReg-High, SPR-VICReg-Low, SPR-Naked+Prio, SPR-Naked+Non,SPR-Naked, 100 runs for SPR from [1].

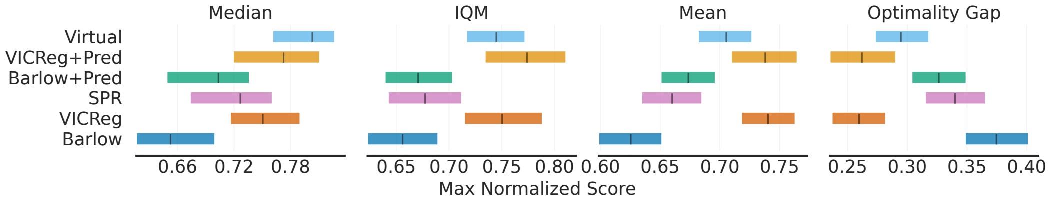

Figure 3: Mean, median, interquartile mean max normalized scores and optimality gap (lower is better) computed with stratified bootstrap confidence intervals in Deep Mind Control Suite 100k, 10 runs for all agents.

SPR and SSL Modifications. The original SPR-agent performs the best (top row of Fig. 2). The modifications to the SPR's SSL objective (see Appx. A.3) have significant impact on the performance but they are not mentioned in the relevant papers (SPR [35], SR-SPR [11], or BBF [37]). The no modifications version, SPR-Naked, performs the worst with a nearly 20% performance drop based on the IQM score (last row of Fig. 2). This is crucial because such modifications may not be suitable for all problem domains, which limits their transferability and generalizability. On the other hand, the role of terminal masking and prioritized replay weighting in SPR is especially interesting, as they help boost performance in situations where pure representation learning struggles.

Incorporating prioritized replay weights has a positive effect on SPR (5^{th} row of Fig. 2). These weights act as markers for Bellman errors that mirror the agent's Q-value approximation performance on particular transitions. Introducing these weights into the representation loss intensifies the emphasis on refining representations that the agent struggles with.

Empirically, terminal state masking shows negligible positive effects, unlike replay weighting, (6^{th} row of Fig. 2). The limited impact of masking might be attributed to the episode lengths, where the agent encounters many regular states but only a single terminal state. The SSL loss may be primarily influenced by intermediate states, which could reduce the effectiveness of masking in these scenarios.

On the other hand, there is a clear synergy between these modifications within SPR. Masking terminal states might be advantageous when agents encounter frequent failures during the initial stages of training or due to the nature of the games. In such cases, terminal states may dominate the replay buffer, which could introduce biased representations that become challenging to correct later on and make it harder for the agent to adapt and improve as it progresses

SPR-Barlow. The performance of the Barlow Twins agent is close to the SPR's (2^{nd} row of Fig. 2), with only a 5% difference, where as SPR-Naked has a 20% gap. As described in Appx. B, modifications related to SSL do not directly apply to Barlow Twins, VICReg, or any other method regularization in the feature dimension. As such, performing similar to a method with RL specific modifications suggests that Barlow Twins has the potential to serve as a substitute, indicating its promise as a versatile SSL objective for data-efficient RL.

The performance gap between SPR-naked and the feature decorrelation methods (Barlow and VICReg) in this context is somewhat surprising since BYOL or Simsiam outperform them in image classification. In vision pretraining, the goal is to obtain embeddings with well-defined clusters based on the training corpora, enhancing classification performance, where feature decorrelation may be of hindrance. In RL, it is important to differentiate between states (good, bad, or promising if they have not been explored yet) which may not be too different in the image space. As such, methods that emphasize the use of the entire embedding space potentially have a better chance of state separation.

To test this, we evaluate the rank [22] of the advantage and value heads, as well as the output of the convolution head, which is shared by both the RL and SSL objectives. We evaluated multiple methods like Barlow Twins and VICReg, in addition to a variant without SSL loss. We found that the rank converges similarly across different games and even if they don't, this does not correlate with performance. We also measured dormant neurons [40] and observed that the results were consistent with the rank findings. These evaluations are detailed in Appx. G.

SPR-VICRegs. Initially, we used the default VICReg hyperparameters given in the original paper [4]. Surprisingly, VICReg exhibits a 13% lower performance (4^{th} row of Fig. 2) compared to SPR although it surpasses SPR-Naked. It also falls short of Barlow Twins. This outcome is not immediately evident given that it has a high similarity to the Barlow Twins' objective. One plausible explanation could be the presence of multiple loss components, possibly undermining covariance. To address this, we explore alternative hyperparameters, selecting the set with the highest covariance hyperparameter that avoids collapse and denote it as SPR-VICReg-High, while the previous one is referred to as SPR-VICReg-Low. However, the performance only marginally increases by 2% (3^{rd} row of Fig. 2), lacking behind Barlow Twins once again. The underlying reasons for this performance gap remain subject to further exploration. Nonetheless, it still showcases the effectiveness of feature decorrelation based objectives since both types outperform SPR-Naked.

2.2 DMControl

We further evaluate the SSL objectives with the DMControl suite, described in Appx. D) since this domain can provide additional insights into the efficacy of SSL objectives in RL. However, since there is no terminal state in this environment and a uniform replay buffer is used, modifications to the SPR loss are not feasible. As such, this evaluation will focus on the generalization of used objectives across domains without targeted optimization for specific problems.

Moreover, SPR is not explicitly designed for continuous control. As such, we use a different set of agents modified for continuous control as described in Appx. B but keep the same SSL hyperparameters from the Atari benchmark. We pick SPR-VICReg-High due to its better performance over the lower covariance version. We additionally evaluate SPR-Barlow and SPR-Vicreg with an MLP layer as an additional predictor, reflecting Bardes et al. [4]'s findings on the enhanced performance of BYOL with variance regularization. We build upon the PlayVirtual [49] methodolody, which is an SPR equipped with an improved transition model, and use it as our baseline.

We observe from Fig. 3 that the Barlow Twins objective exhibits the lowest performance, although it closely aligns with SPR, with IQM scores of 0.656, and 0.677 respectively. An interesting observation is that VICReg with an IQM of 0.75 is as good as PlayVirtual [49] with 0.744. This underscores the potential of SSL objectives in continuous control. While their impact is vital in discrete control as well, the overall effect, especially when considering the maximum score (representing human performance), is relatively modest. Nevertheless, a substantial improvement is evident in continuous control, even when compared to the highest achievable score. We also see that adding a predictor network has a minimal but positive impact on the IQM performances of both Barlow and VICReg.

3 Conclusion

Our study investigates the impact of RL specific and off-the-shelf SSL objectives on SPR. We find that RL-based modifications are essential in discrete control but may not be always applicable, especially in continuous control. Despite SSL coupled with RL being most effective in Atari, Barlow Twins performs well without any modifications, indicating that SSL objectives can yield strong results without problem-specific modifications. However, its success in Atari doesn't transfer to DMControl, where VICReg excels. VICReg stands out with consistent performance in both domains, highlighting its effectiveness as an SSL objective in self-predictive RL, while being problem-agnostic.

4 Acknowledgements

This work was supported by KUIS AI Center computational resources.

References

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. 2021. Deep Reinforcement Learning at the Edge of the Statistical Precipice. In *Neural Information Processing Systems*.

[2] Benjamin J. Ayton and Masataro Asai. 2021. Width-Based Planning and Active Learning for Atari. In *International Conference on Automated Planning and Scheduling*.

[3] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. 2023. A Cookbook of Self-Supervised Learning.

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. 2021. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. *ArXiv*, abs/2105.04906.

[5] Omer Veysel Cagatan and Baris Akgun. 2023. BarlowRL: Barlow Twins for Data-Efficient Reinforcement Learning.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2021. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments.

[7] Edoardo Cetin, Philip J. Ball, Steve Roberts, and Oya Çeliktutan. 2022. Stabilizing Off-Policy Deep Reinforcement Learning from Pixels. In *International Conference on Machine Learning*.

[8] Edoardo Cetin, Benjamin Paul Chamberlain, Michael M. Bronstein, and Jonathan J. Hunt. 2022. Hyperbolic Deep Reinforcement Learning. *ArXiv*, abs/2210.01542.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *ArXiv*, abs/2002.05709.

[10] Xinlei Chen and Kaiming He. 2020. Exploring Simple Siamese Representation Learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15745–15753.

[11] Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G. Bellemare, and Aaron C. Courville. 2023. Sample-Efficient Reinforcement Learning by Breaking the Replay Ratio Barrier. In *International Conference on Learning Representations*.

[12] Manuel Goulão and Arlindo L. Oliveira. 2022. Pretraining the Vision Transformer using self-supervised methods for vision based Deep Reinforcement Learning. *ArXiv*, abs/2209.10901.

[13] Manuel Goulão and Arlindo L. Oliveira. 2023. Pretraining the Vision Transformer using self-supervised methods for vision based Deep Reinforcement Learning.

[14] Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. *ArXiv*, abs/2006.07733.

[15] Tuomas Haarnoja, Aurick Zhou, P. Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *ArXiv*, abs/1801.01290.

[16] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2023. Mastering Diverse Domains through World Models. *arXiv preprint arXiv:2301.04104*.

[17] H. V. Hasselt, Matteo Hessel, and John Aslanides. 2019. When to use parametric models in reinforcement learning? *ArXiv*, abs/1906.05243.

[18] Matteo Hessel, Joseph Modayil, H. V. Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. 2017. Rainbow: Combining Improvements in Deep Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.

[19] Tao Huang, Jiacheng Wang, and Xiao Chen. 2022. Accelerating Representation Learning with View-Consistent Dynamics in Data-Efficient Reinforcement Learning. *ArXiv*, abs/2201.07016.

[20] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, K. Czechowski, D. Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, G. Tucker, and Henryk Michalewski. 2019. Model-Based Reinforcement Learning for Atari. *ArXiv*, abs/1903.00374.

[21] Ilya Kostrikov, Denis Yarats, and Rob Fergus. 2020. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. *ArXiv*, abs/2004.13649.

[22] Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. 2021. Implicit Under-Parameterization Inhibits Data-Efficient Deep Reinforcement Learning.

[23] Hojoon Lee, Hanseul Cho, Hyunseung Kim, Daehoon Gwak, Joonkee Kim, Jaegul Choo, Se-Young Yun, and Chulhee Yun. 2023. Enhancing Generalization and Plasticity for Sample Efficient Reinforcement Learning. *ArXiv*, abs/2306.10711.

[24] Hojoon Lee, Koanho Lee, Dongyoon Hwang, Hyunho Lee, Byungkun Lee, and Jaegul Choo. 2023. On the Importance of Feature Decorrelation for Unsupervised Representation Learning in Reinforcement Learning.

[25] Xiang Li, Jinghuan Shang, Srijan Das, and Michael S. Ryoo. 2023. Does Self-supervised Learning Really Improve Reinforcement Learning from Pixels?

[26] Litian Liang, Yaosheng Xu, Stephen McAleer, Dailin Hu, Alexander T. Ihler, P. Abbeel, and Roy Fox. 2022. Reducing Variance in Temporal-Difference Value Estimation via Ensemble of Deep Networks. *ArXiv*, abs/2209.07670.

[27] Hao Liu and P. Abbeel. 2021. APS: Active Pretraining with Successor Features. In *International Conference on Machine Learning*.

[28] Vincent Micheli, Eloi Alonso, and Franccois Fleuret. 2022. Transformers are Sample Efficient World Models. *ArXiv*, abs/2209.00588.

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Kirkeby Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

[30] Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron C. Courville. 2022. The Primacy Bias in Deep Reinforcement Learning. In *International Conference on Machine Learning*.

[31] Johan Obando-Ceron, João G. M. Araújo, Aaron Courville, and Pablo Samuel Castro. 2024. On the consistency of hyper-parameter selection in value-based deep reinforcement learning.

[32] Serdar Ozsoy, Shadi S. Hamdan, Sercan Ö. Arik, Deniz Yuret, and Alper Tunga Erdogan. 2022. Self-Supervised Learning with an Information Maximization Criterion. *ArXiv*, abs/2209.07999.

[33] Jan Robine, Marc Hoftmann, Tobias Uelwer, and Stefan Harmeling. 2023. Transformer-based World Models Are Happy With 100k Interactions. *ArXiv*, abs/2303.07109.

[34] Jan Robine, Tobias Uelwer, and Stefan Harmeling. 2021. Smaller World Models for Reinforcement Learning.

[35] Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron C. Courville, and Philip Bachman. 2020. Data-Efficient Reinforcement Learning with Self-Predictive Representations. In *International Conference on Learning Representations*.

[36] Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron C. Courville, and Philip Bachman. 2021. Repository published by the SPR. `https://github.com/mila-iqia/spr`.

[37] Max Schwarzer, Johan S. Obando-Ceron, Aaron C. Courville, Marc G. Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. 2023. Bigger, Better, Faster: Human-level Atari with human-level efficiency. *ArXiv*, abs/2305.19452.

[38] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, Devon Hjelm, Philip Bachman, and Aaron C. Courville. 2021. Pretraining Representations for Data-Efficient Reinforcement Learning. In *Neural Information Processing Systems*.

[39] Thalles Silva, Helio Pedrini, and Adín Ramírez Rivera. 2024. Learning from Memory: Non-Parametric Memory Augmented Self-Supervised Learning of Visual Features. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 45451–45467. PMLR.

[40] Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. 2023. The Dormant Neuron Phenomenon in Deep Reinforcement Learning.

[41] A. Srinivas, Michael Laskin, and P. Abbeel. 2020. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. *ArXiv*, abs/2004.04136.

[42] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. 2018. DeepMind Control Suite.

[43] Manan Tomar, Utkarsh A. Mishra, Amy Zhang, and Matthew E. Taylor. 2021. Learning Representations for Pixel-based Control: What Matters and Why?

[44] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation Learning with Contrastive Predictive Coding.

[45] Ziyun Wang, Tom Schaul, Matteo Hessel, H. V. Hasselt, Marc Lanctot, and Nando de Freitas. 2015. Dueling Network Architectures for Deep Reinforcement Learning. In *International Conference on Machine Learning*.

[46] Xi Weng, Yunhao Ni, Tengwei Song, Jie Luo, Rao Muhammad Anwer, Salman Khan, Fahad Shahbaz Khan, and Lei Huang. 2024. Modulate Your Spectrum in Self-Supervised Learning.

[47] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. 2020. Improving Sample Efficiency in Model-Free Reinforcement Learning from Images.

[48] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. 2021. Mastering Atari Games with Limited Data.

[49] Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang, and Zhibo Chen. 2021. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *Advances in Neural Information Processing Systems*, 34.

[50] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. *ArXiv*, abs/2103.03230.

[51] Yifan Zhang, Zhiquan Tan, Jingqin Yang, Weiran Huang, and Yang Yuan. 2024. Matrix Information Theory for Self-Supervised Learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 59897–59918. PMLR.

[52] Jinghao Zhou, Li Dong, Zhe Gan, Lijuan Wang, and Furu Wei. 2022. Non-Contrastive Learning Meets Language-Image Pre-Training.

[53] Ömer Veysel Çağatan. 2024. UNSEE: Unsupervised Non-contrastive Sentence Embeddings.

A Background

A.1 Barlow Twins

The Barlow Twins [50] employs a symmetric network with twin branches, each processing a different augmented perspective of input data. It aims to minimize off-diagonal components and align diagonal elements of a cross-covariance matrix derived from the representations of these branches. The process involves generating two altered views (Y^A and Y^B) using data augmentations, inputting them into a function f_θ to produce embeddings (Z^A and Z^B).

The Barlow Twins loss is defined as:

$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} {\mathcal{C}_{ij}}^2}_{\text{redundancy reduction term}} \tag{1}$$

where $\lambda > 0$ balances the invariance (diagonal elements) and redundancy reduction (off-diagonal) in the loss function. $\mathcal{C}$ is the cross-correlation matrix from embedding outputs of identical networks in the batch. A matrix element is defined as:

$$\mathcal{C}_{ij} \triangleq \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b \left(z^A_{b,i}\right)^2} \sqrt{\sum_b \left(z^B_{b,j}\right)^2}} \tag{2}$$

where b represents the samples in the batch, and i and j represent dimension indices of the networks' output. Each dimension of the square covariance matrix, $\mathcal{C}$, is the same as the embedding dimension (output dimensionality of the networks). Its values range between -1 (indicating complete anti-correlation) and 1 (representing perfect correlation).

A.2 VICReg

VICReg [4] is a method designed to tackle the challenge of collapse directly. It achieves this by introducing a straightforward regularization term that specifically targets the variance of the embeddings along each dimension independently. In addition to addressing the variance, VICReg includes a mechanism to diminish redundancy and ensure decorrelation among the embeddings, accomplished through covariance regularization.

The variance regularization term is a hinge function on the standard deviation of the embeddings along the batch dimension:

$$v(Z) = \frac{1}{d} \sum_{j=1}^{d} \max(0, \gamma - S(z^j, \epsilon)) \tag{3}$$

where S is the regularized standard deviation defined by:

$$S(x; \epsilon) = \sqrt{\mathrm{Var}(x) + \epsilon} \tag{4}$$

Covariance matrix of Z is defined as:

$$C(Z) = \frac{1}{n-1} \sum_{i=1}^{n} (z_i - \bar{z})(z_i - \bar{z})^T \tag{5}$$

where $\bar{z} = \frac{1}{n} \sum_{i=1}^{n} z_i$. Covariance regularization is defined as:

$$c(Z) = \frac{1}{d} \sum_i \sum_{j \neq i} {\mathcal{C}_{ij}}^2 \tag{6}$$

where d is the feature dimension. The invariance criterion between Z and Z' is the mean-squared Euclidean distance between each pair of vectors, without any normalization.

$$s(Z, Z') = \frac{1}{n} \sum_{i=1}^{n} ||z_i - z'_i||^2 \tag{7}$$

The overall loss function is a weighted average of the invariance, variance, and covariance terms:

$$l(Z, Z') = \alpha v(Z) + \beta c(Z) + \gamma s(Z, Z') \tag{8}$$

where α, λ, and γ hyper-parameters control the importance of each term in the loss.

VICReg is quite similar to Barlow Twins in terms of its loss formulation. However, instead of decorrelating the cross-correlation matrix directly, it regularizes the variance along each dimension of the representation, reduces correlation and minimizes the difference of embeddings. This prevents dimension collapse and also forces the two views to be encoded similarly. Additionally, reducing covariance encourages different dimensions of the representation to capture distinct features.

A.3 SPR

Self Predictive Representations (SPR) [35] is a performant data-efficient agent and a baseline of many other performant agents [37, 30, 11, 49] and its general architecture is depicted in Figure 1. The approach trains an agent by having it predict the latent state based on the current state. It encodes the present state, forecasts the latent representation of the next state using a transition model, and calculates loss by measuring the mean squared error between normalized embeddings. Additionally, SPR adjusts its loss through terminal masking and prioritized replay weighting. These two modifications inject RL-specific information into the auxiliary self-supervised learning task. While the utilization of these ideas is not explicitly mentioned by Schwarzer et al. [35], it is possible that these techniques were considered self-evident and consequently were included in their implementation [36]. We mention them here so as to be able to better differentiate between SPR and other SPR variants.

SSL loss matrix in SPR denoted as L, encompasses negative cosine similarities between predicted latent representations and ground truth latent representations, with dimensions of $B \times (K+1)$, where B is the batch size, and K is the prediction horizon with 1 coming from the current observation. The batch of interactions is drawn from the replay buffer, and their terminal status is known. The terminal mask matrix, M, is composed of 0s and 1s denoting terminal and non-terminal states. The process involves updating L through a Hadamard product with M, denoted as $L \circ M$, effectively modifying the loss matrix.

The loss matrix is divided into two components: SPR loss and Model SPR loss. SPR loss is between the latent representations of the augmented views of the present state. Model SPR loss is between the latent representations of the augmented views of the future states and the predicted future latent representations, generated by the transition model. Model SPR is averaged across the temporal dimension and as a result, both components have $N \times 1$ dimensionality.

The loss of each transition is multiplied by the prioritized replay weighting, as determined by the temporal difference errors. Then the final loss is computed as the weighted sum of the average SPR loss and half the average of the Model SPR loss across a batch, defined as follows:

$$\mathcal{L}_{SPR} = \frac{1}{N} \sum_{i=1}^{N} \omega_i (\lambda \text{SPR}_i + \gamma \text{Model SPR}_i) \tag{9}$$

where N is the batch size, ω_i is the priority weight ($\sum_i \omega_i = 1$), and i indexes individual transitions, where λ,γ are hyperparameters.

B SPR-*

Despite variations in SSL objectives and RL algorithms across different benchmarks, the architecture remains largely consistent, as depicted in Figure 1. SPR employs a BYOL [14] objective with a momentum of 1, essentially adopting the SimSiam [10] approach. The primary architectural distinction lies in the inclusion of an extra predictor layer in the online MLP of BYOL or SimSiam to prevent collapse, a feature omitted in the original Barlow Twins and VICReg formulations as their objectives inherently mitigate the risk of collapse.

SPR-Nakeds While SPR demonstrates considerable efficacy, the fundamental question remains unanswered—what is the impact of pure self-supervised learning and potential adaptations leading to SPR? Consequently, we introduce SPR-Naked, representing pure SSL. To assess the effects

	Median	IQM	Mean	Opt. Gap
Stop-Gradient	**0.271**	**0.303**	**0.615**	**0.577**
No Stop-Gradient	0.266	0.282	0.595	0.611

Table 1: Human-normalized aggregate metrics in Atari 100k by VICReg-High. Scores, collected from 10 random runs to assess the efficacy of including stop-gradient.

	Median	IQM	Mean	Opt. Gap
Barlow	**0.324**	**0.320**	**0.605**	**0.5930**
VICReg	0.281	0.289	0.600	0.610
VICReg + Non	0.221	0.279	0.554	0.617
Barlow + Non	- 0.009	- 0.011	-0.171	1.171
ZeroJump	0.270	0.262	0.528	0.636

Table 2: Human-normalized aggregate metrics in Atari 100k. Scores were collected from 10 random runs.

of prioritized replay weighting and terminal masking, we further establish SPR-Naked+Prio and SPR-Naked+Non, respectively.

SPR-Barlow To extend the Barlow Twins to future predictions, we compute individual cross-correlation matrices for both the current and predicted latent representations at each time step. This results in a total of $K + 1$ matrices, each with dimensions $d \times d$, where d denotes the embedding dimension within a single batch. Subsequently, we calculate the loss for each matrix and average the results. To make it easier to compare, we can define $\overline{\text{SPR}}$ Loss and $\overline{\text{Model SPR}}$ Loss analogously to their SPR counterparts, where the first is about the current state and the latter is about the future states. The final loss is then;

$$\mathcal{L}_{SPR-Barlow} = \overline{\text{SPR}} + \frac{1}{K}\sum_{k=1}^{K}\overline{\text{Model SPR}}_k \tag{10}$$

where K is the number of predicted future observations.

SPR-VICRegs We employ a parallel procedure as in Barlow Twins for VICReg. We introduce two variations of VICReg-High and VICReg-Low, featuring high or low covariance weights in the VICReg loss (Equation 8), while maintaining consistency in other hyperparameters. The primary objective is to observe the impact of feature decorrelation without inducing model collapse.

Why not employ replay weighting and terminal state masking in Barlow/VICReg? The key limitation preventing the use of replay weighting or terminal masking in feature decorrelation-based methods lies in their reliance on covariance regularization. These methods employ either a cross-correlation matrix or a covariance matrix, both with dimensions matching the feature dimension. This structure prohibits applying the weighting of a feature dimension matrix using a batch dimension matrix. Consequently, these methods produce a unified loss for the entire batch, unlike approaches such as BYOL or SimSiam, which generate losses on a per-sample basis.

Why use stop-gradient in Barlow/VICReg? Barlow Twins and VICReg effectively prevent collapse without resorting to symmetry-breaking architectural techniques such as predictor layers or stop-gradient mechanisms. While not strictly necessary in this scenario, we choose to include a stop-gradient due to its empirically observed performance improvement, as depicted in Table 1. A more grounded reason stems from the architectural asymmetry introduced by the transition model. In the absence of a stop-gradient, gradients from the encoder's upper branch flow through the transition model, whereas gradients from the lower branch directly influence the encoder. This asymmetry can potentially lead to suboptimal encoder updates. Despite collapse avoidance in both cases, the inclusion of a stop-gradient is maintained for its superior performance outcomes.

Removing Features with Masking We discussed why post-loss-calculation modifications cannot be applied to objectives that involve components in the feature dimension rather than the batch dimension. However, non-terminal masking can be employed to exclude samples from the batch before calculating the SSL loss. Thus, we masked features during the training of the SPR-VICReg and SPR-Barlow agents, leading to unexpected results. As shown in Table 2, the SPR-Barlow agent performed even worse than the random agent. A likely explanation is that the Barlow Twins' objective relies on batch normalization to compute the cross-covariance matrix. Since masking causes the batch size to vary dynamically, the batch statistics become inconsistent, adversely affecting the batch normalization process.However, this degradation is not observed to the same extent in the SPR-VICReg agent, as the VICReg objective does not rely on batch normalization.

Continuous Control Formulation Although SPR is created specifically for discrete control, delving into the impact of SSL objectives solely within discrete control domains doesn't provide a comprehensive understanding. This is why we adopt a parallel setup to that of PlayVirtual [49], where they establish an SPR-like scheme referred to as SPR† as a baseline for continuous control. They utilize the soft actor-critic algorithm [15], instead of q-learning due to the continuous nature of the actions. They do not use terminal state masking (since terminal states for control problems are target states) and prioritized replay weighting (since they use a uniform buffer). This shows the importance of generally applicable auxiliary tasks for data-efficient RL.

We evaluate PlayVirtual and SPR† from scratch since we were not able to replicate Yu et al. [49]'s results, potentially due to different benchmark versions. Furthermore, we assess the performance of VICReg-High and Barlow Twins within the SPR† configuration. We exclude VICReg-Low in this setting due to the minimal performance difference observed in Atari.

Finally, we explore the potential impact of incorporating the predictor network into Barlow Twins and VICReg, even though they inherently do not need it to prevent dimension collapse. Although the addition of a predictor network is novel in Barlow Twins, VICReg becomes similar to the SPR with this addition like SPR with variance-covariance regularization. The decision to refrain from conducting similar experiments in Atari stems from the substantially higher experimental costs, which are at least 10 times greater than those in the control setting.

C Related Work

Tomar et al. [43] tackles a more challenging setting with background distractors, using a simple baseline approach that avoids metric-based learning, data augmentations, world-model learning, and contrastive learning. They analyze why previous methods may fail or perform similarly to the baseline in this tougher scenario and stress the importance of detailed benchmarks based on reward density, planning horizon, and task-irrelevant components. They propose new metrics for evaluating algorithms and advocate for a data-centric approach to better apply RL to real-world tasks.

Li et al. [25] explore whether SSL can enhance online RL from pixel data. By extending the contrastive reinforcement learning framework [41] to jointly optimize SSL and RL losses, and experimenting with various SSL losses, they find that the current SSL approaches offer no significant improvement over baselines that use image augmentation alone, given the same data and augmentation. Even after evolutionary searches for optimal SSL loss combinations, these methods do not outperform carefully designed image augmentations. Their evaluation across various environments, including real-world robots, reveals that no single SSL loss or augmentation method consistently excels.

C.1 Data Efficient RL in Atari 100k

The introduction of the Atari 100k benchmark [20] has expedited the advancement of sample-efficient reinforcement learning algorithms. Model-based approach, SimPLe [20], outperformed Rainbow DQN [18], showcasing superior performance. Building on Rainbow's framework, Hasselt et al. [17] enhanced its efficacy through minor hyperparameter adjustments, resulting in Data-Efficient Rainbow (DER), which achieved a higher score compared to SimPLe.

DrQ [21] employs a multi-augmentation strategy to regularize the value function during training of both Soft Actor-Critic [15] and Deep Q-Network [29]. This approach effectively reduces overfitting and enhances training efficiency, leading to performance improvements for both algorithm

Several prevalent methods adopt the Atari 100k dataset, and these can be classified as follows: Model-Based [16, 33, 28, 2, 34], Pretraining [12, 38, 23, 27], Model-Free [37, 19, 30, 7, 23, 26]

C.2 Representation Learning in Atari 100k

Cetin et al. [8] presents a deep reinforcement learning method using hyperbolic space for latent representations. Their innovative approach tackles optimization challenges in existing hyperbolic deep learning, ensuring stable end-to-end learning through deep hyperbolic representations.

Huang et al. [19] proposes a Multiview Markov Decision Process (MMDP) with View-Consistent Dynamics (VCD), a method that enhances traditional MDPs by considering multiple state perspectives. VCD trains a latent space dynamics model for consistent state representations, achieved through data augmentation.

Srinivas et al. [41] incorporate the InfoNCE [44] as an auxiliary component within DER. Cagatan and Akgun [5] uses Barlow Twins [50] instead of a contrastive objective to further improve results. This integration serves to enhance the learning process. SPR [35] outperforms all previous model-free approaches by predicting its latent state representations multiple steps into the future with BYOL [14].

PlayVirtual [49] introduces a novel transition model as an alternative to the simplistic module in SPR. The methodology enriches actual trajectories by incorporating a multitude of cycle-consistent virtual trajectories. These virtual trajectories, generated using both forward and backward dynamics models, collectively form a closed 'trajectory cycle.' The crucial aspect is ensuring the consistency of this cycle, validating the projected states against real states and actions. This innovative approach significantly improves data efficiency, empowering reinforcement learning algorithms to acquire robust feature representations with reduced reliance on real-world experiences. This method proves particularly advantageous for tasks where obtaining real-world data is costly or challenging.

D Evaluation Setup

D.1 Atari 100k

We assess the SPR framework in a reduced-sample Atari setting, called the Atari 100k benchmark [20]. In this setting, the training dataset comprises 100,000 environment steps, which is equivalent to about 400,000 frames or slightly under two hours of equivalent human experience. This contrasts with the conventional benchmark of 50,000,000 environment steps, corresponding to approximately 39 days of accumulated experience.

The main metric for this setting, widely acknowledged for assessing performance in the Atari 100k context, is the human-normalized score. This measure is mathematically defined as in equation 11, where random score pertains to outcomes achieved through a random policy and the human score is derived from human players [45].

$$\frac{score_{\text{agent}} - score_{\text{random}}}{score_{\text{human}} - score_{\text{random}}} \tag{11}$$

D.2 Deep Mind Control Suite

In the Deep Mind Control Suite [42], the agent is configured to function solely based on pixel inputs. This choice is justified by several reasons: the environments involved offer a reasonably challenging and diverse array of tasks, the sample efficiency of model-free reinforcement learning algorithms is notably low when operating directly from pixels in these benchmarks and the performance on the DM control suite is comparable to the context of robot learning in real-world benchmarks.

We use the following six environments [47] for benchmarking: ball-in-cup, finger-spin, reacher-easy, cheetah-run, walker-walk and cartpole-swingup, for 100k steps each.

The main metric for this setting is the normalized score with respect to the maximum score. Note that human scores are not suitable for such a continuous control setting.

D.3 Benchmarking: Rliable Framework

Agarwal et al. [1] discusses the limitations of using mean and median scores as singular estimates in RL benchmarks and highlights the disparities between conventional single-point estimates and the broader interval estimates, emphasizing the potential ramifications for benchmark dependability and interpretation. In alignment with their suggestions, we provide a succinct overview of human-normalized scores, furnished with stratified bootstrap confidence intervals, in Figures 2 and 3.

E Full Results on Atari 100k

Table 3: Returns on the 26 games of Atari 100k after 2 hours of real-time experience, and human-normalized aggregate metrics. (VR: VICReg, results with 5 integral digits are rounded to the first integer to fit the table)

Game	Rand.	Human	Naked	Non	Prio	VR-L	VR-H	Barlow	SPR
Alien	227.8	7127.7	868.9	881.7	872.7	902.9	922.4	891.8	841.9
Amidar	5.8	1719.5	165.6	179.1	164.2	181.1	176.4	177.1	179.7
Assault	222.4	742.0	544.5	564.6	589.2	536.4	575.7	581.4	565.6
Asterix	210	8503.3	972.0	951.0	977.8	955.4	1021.7	981.2	962.5
BankHeist	14.2	753.1	61.6	70.1	60.2	79.9	82.9	73.5	345.4
BattleZone	2360	37188	7552.4	9424.2	13102	12557	14892	14954	14834
Boxing	0.1	12.1	27.3	30.4	36.4	31.3	33.9	35.1	35.7
Breakout	1.7	30.5	16.7	18.0	18.2	16.9	16.3	17.0	19.6
ChopComm	811	7387.8	906.8	949.8	901.0	832.9	929.9	938.9	946.3
CrzyClmbr	10781	35829	30056	32667	35829	27035	29023	29229	36701
DemonAtt	152.1	1971.0	514.7	511.0	522.9	461.2	547.2	519.2	517.6
Freeway	0.0	29.6	17.4	13.71	16.3	28.0	27.7	29.5	19.3
Frostbite	65.2	4334.7	1137.2	1010.9	1014.2	1353.0	1181.4	1191.3	1170.7
Gopher	257.6	2412.5	585.0	660.1	548.4	737.9	713.5	691.2	660.6
Hero	1027	30826	6937.8	6497.8	5686.6	5495.1	5559.6	5746.8	5858.6
Jamesbond	29	302.8	327.2	359.9	349.1	357.6	384.3	404.2	366.5
Kangaroo	52	3035.0	2970.9	2812.1	3016.5	2290.6	1998.3	1771.2	3617.4
Krull	1598	2665.5	3980.4	4061.8	4213.1	4166.6	4513.9	4363.2	3681.6
KFMaster	258.5	22736	13126	14595	15757	1488.4	15548	15998	14783
MsPacman	307.3	6951.6	1262.1	1162.6	1324.6	1366.8	1588.2	1388.2	1318.4
Pong	-20.7	14.6	-1.8	-6.0	-7.2	-6.3	-10.1	-6.7	-5.4
PrivateEye	24.9	69571	85.6	77.0	88.0	100.9	96.6	99.6	86.0
Qbert	163.9	13455	847.2	758.6	759.8	796.9	687.6	765.8	866.3
RoadRunner	11.5	7845.0	12595	12713	11211	10683	9531.5	12412	12213
Seaquest	68.4	42055	524.0	524.2	523.2	576.3	651.0	669.1	558.1
UpNDown	533.4	11693	9569.3	8130.6	10331	7952.7	9415.3	10818	10859
#Sprhmn(↑)	0	N/A	4	3	3	4	4	4	6
Mean (↑)	0.00	1.000	0.542	0.555	0.608	0.558	0.585	0.608	0.616
Median (↑)	0.00	1.000	0.225	0.221	0.308	0.297	0.280	0.312	0.396
IQM (↑)	0.00	1.000	0.273	0.278	0.298	0.292	0.298	0.321	0.337
Opt. Gap (↓)	1.00	0.000	0.617	0.615	0.603	0.609	0.605	0.587	0.577

F Full Results on DMControl 100k

Table 4: Returns on the of DMControl 100k, and Max-normalized aggregate metrics.

Environment	Virtual	VICReg+Pred	Barlow+Pred	SPR	VICReg	Barlow
FINGER, SPIN	896.2	760.6	781.0	755.9	730.0	861.8
CARTPOLE, SWINGUP	815.1	791.6	784.0	826.0	780.1	778.6
REACHER, EASY	827.0	790.7	589.6	671.5	736.1	526.5
CHEETAH, RUN	489.6	504.3	461.6	435.2	493.5	478.6
WALKER, WALK	404.7	622.8	521.7	404.7	765.	182.2
BALL IN CUP, CATCH	835.4	891.6	622.8	835.4	937.5	924.9
Mean (↑)	0.705	0.738	0.673	0.660	0.740	0.625
Median (↑)	0.803	0.772	0.703	0.726	0.750	0.652
IQM (↑)	0.744	0.773	0.670	0.677	0.750	0.656
Optimality Gap (↓)	0.294	0.260	0.326	0.339	0.259	0.374

G Rank and Dormant Neuron

Kumar et al. [22] introduced the concept of *effective rank* for representations, represented as $srank_\delta(\phi)$, with δ being a threshold parameter, set to 0.01 as per their study. They proposed that effective rank is linked to the expressivity of a network, where a decrease in effective rank implies an implicit under-parameterization. The study provides evidence indicating that bootstrapping is the primary factor contributing to the collapse of effective rank, which in turn degrades performance.

To investigate how SSL objectives might mitigate rank collapse, we computed the rank of the convolution output and the outputs of the penultimate layers from the advantage and value heads of three different agents: SPR-VICReg, SPR-Barlow, and ZeroJump (SPR without a transition model), scores in 2. Our observations indicate that, although there are some rank differences among the agents, they often converge to the same rank, and these differences do not correlate with the performance scores. Figure 4, 6 and 5 include ranks across all games.

To explore this further, we examined the proportion of dormant neurons, which are neurons that have near-zero activations. Sokar et al. [40] showed that deep reinforcement learning agents experience a rise in the number of dormant neurons within their networks. Additionally, a higher prevalence of dormant neurons is associated with poorer performance.

We also do not observe a clear pattern in the fractions of dormant neurons, in Figure 7 that could account for the disparities in performance scores, similar to what was seen in the case of neuron ranks. Unlike rank-based observations, where patterns may emerge, the distribution of dormant neurons does not offer an explanation for the differences in the scores across models. This suggests that the relationship between neuron activity and performance metrics might be more complex and not directly attributable to the proportion of inactive neurons.

H Experimental Details

We retain all hyperparameters of SPR, SR-SPR, and BBF, except for SPR-Barlow and SPR-VICReg, where we adjust the SPR loss weight and increase the batch size from 32 to 64. The official repositories of the models are used, and all experiments are conducted on a Tesla T4 GPU.

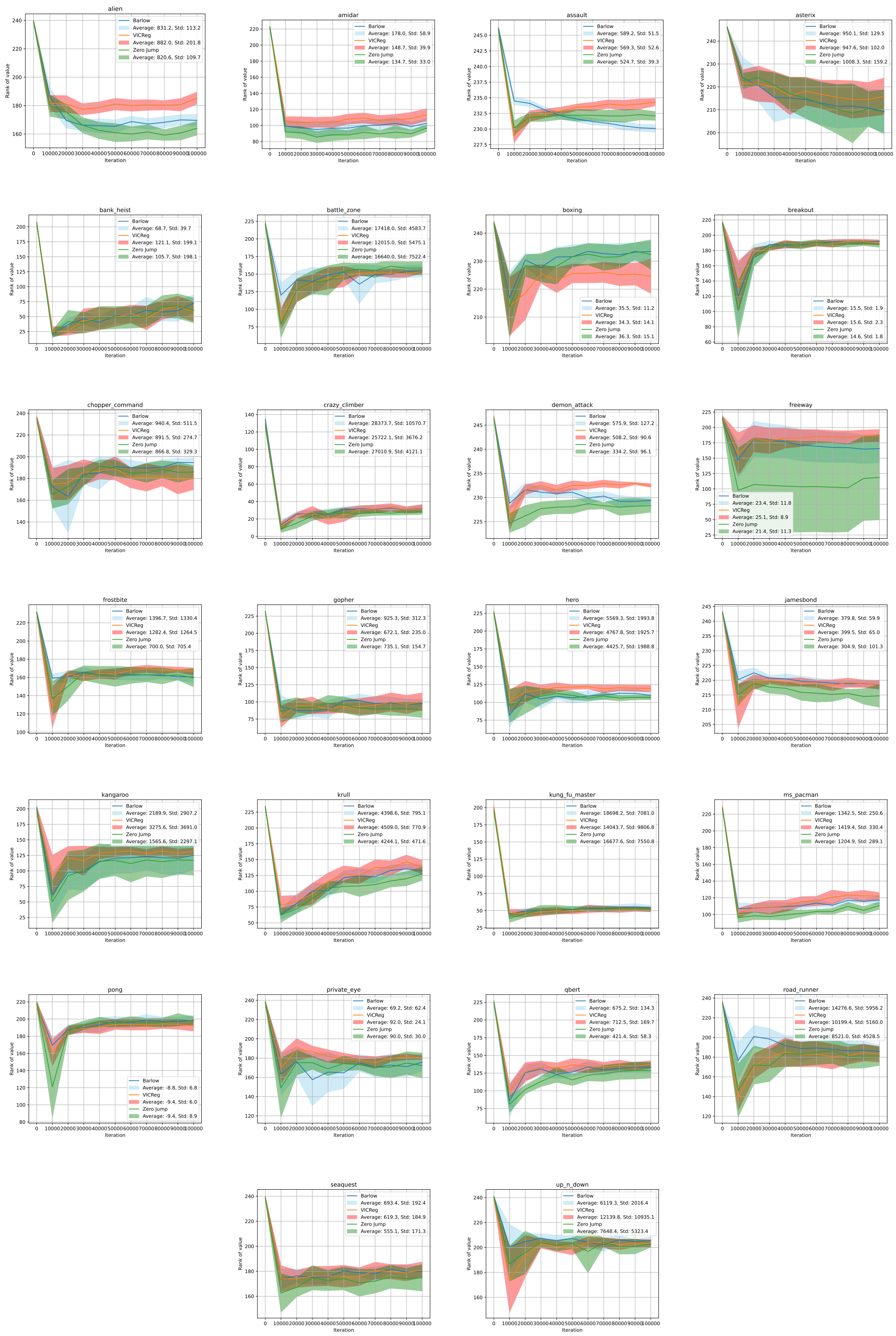

Figure 4: Rank of the output from the penultimate layer of the value head, measured every 10,000 steps and averaged across 10 different runs for every game.

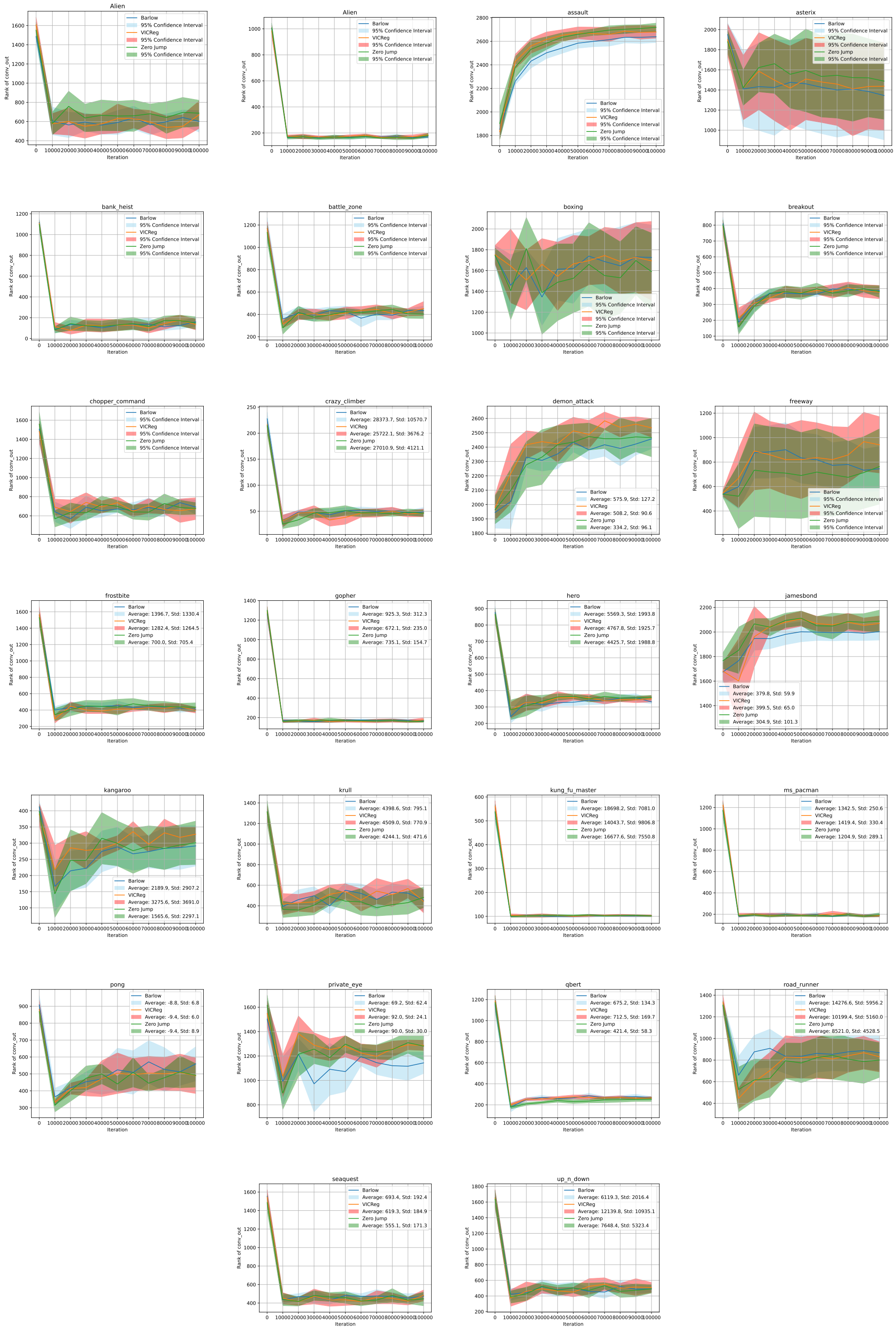

Figure 5: Rank of the output from the convolution encoder, measured every 10,000 steps and averaged across 10 different runs for every game.

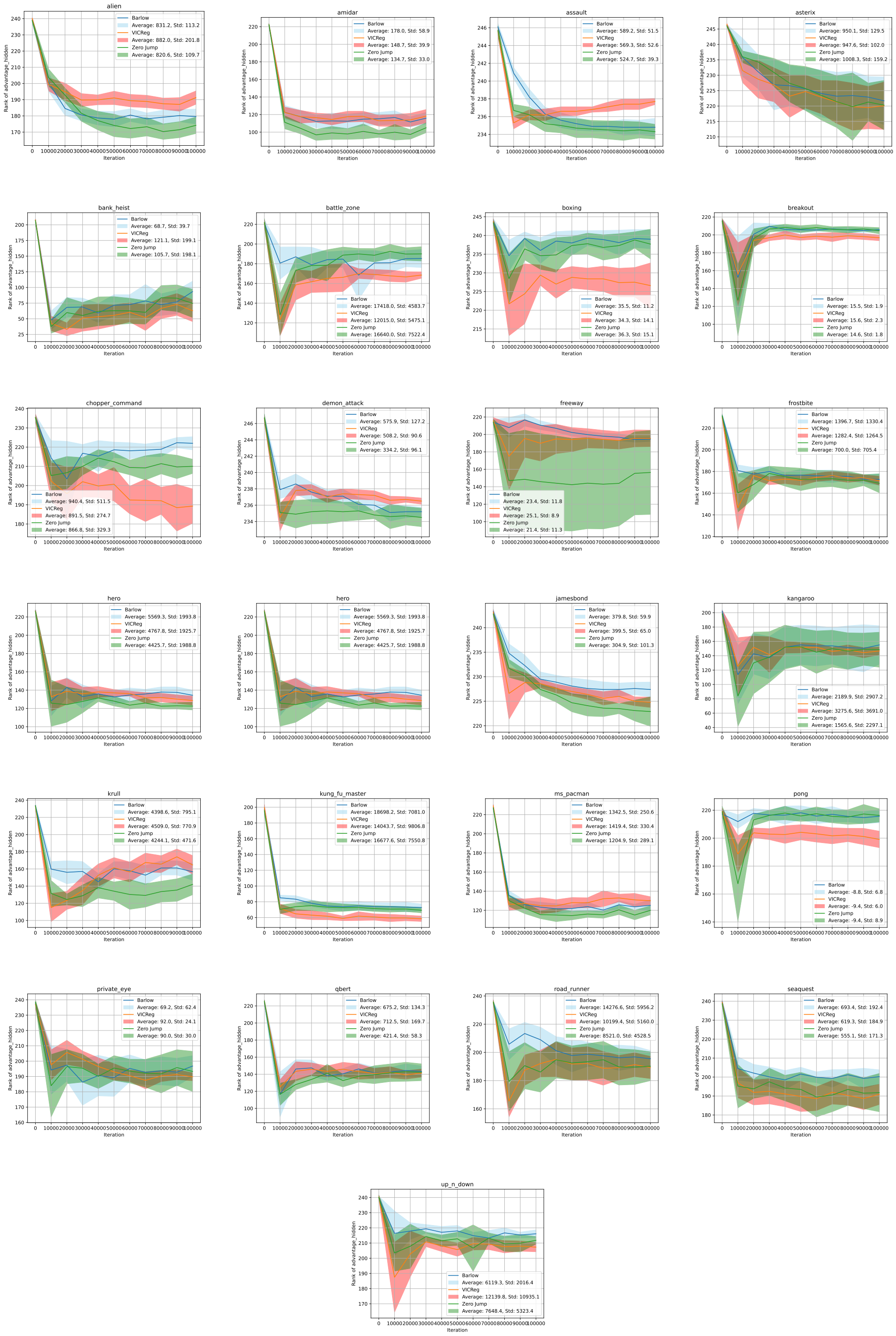

Figure 6: Rank of the output from the penultimate layer of the advantage head, measured every 10,000 steps and averaged across 10 different runs for every game.

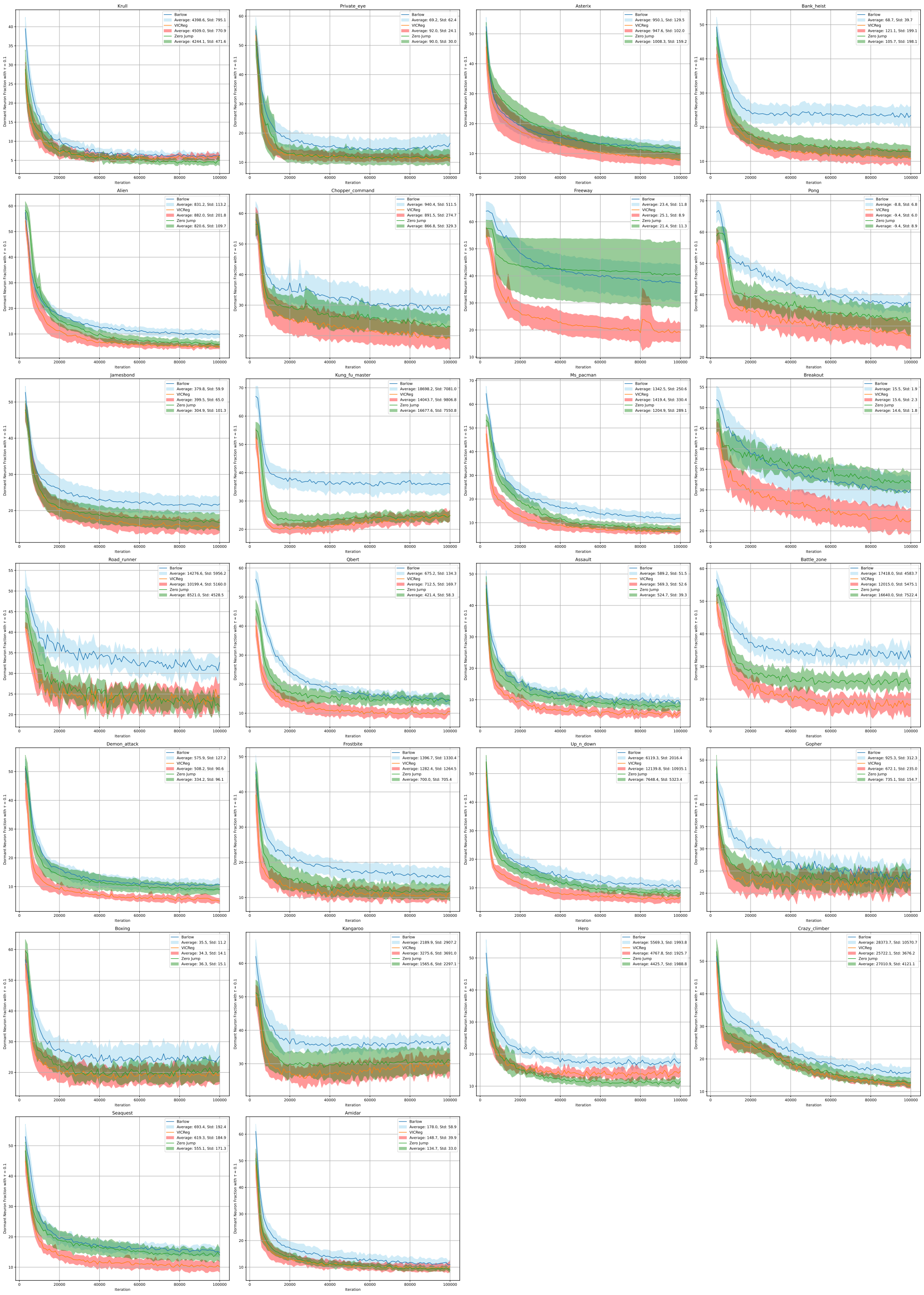

Figure 7: Fraction of dormant neurons averaged across 10 different runs for every game.