Understanding Learned Representations and Action Collapse in Visual Reinforcement Learning

Anonymous authors
Paper under double-blind review

Keywords: Visual reinforcement learning, representation understanding.

Summary

In contrast to deep learning models trained with supervised data, visual reinforcement learning (VRL) models learn to represent their environment implicitly via the process of seeking higher rewards. However, there has been little research on the specific representations VRL models learn. Using linear probing, we study the extent to which VRL models learn to linearly represent the ground truth vectorized state of an environment, on which layers these representations are most accessible, and how this relates to the reward achieved by the final model. We observe that poorly performing agents differ substantially from well-performing ones in the representation learned in their later MLP layers, but not their earlier CNN layers. When an agent is initialized by reusing the later layers of a poorly performing agent, the result is always poor. These poorly performing agents end up with no entropy in their actor network output, a phenomenon we call action collapse. Based on these observations, we propose a simple rule to prevent action collapse during training, leading to better performance on tasks with image observations with no additional computational cost.

Contribution(s)

1. We analyze how VRL models learn linear representations of the ground truth vectorized environment states using Orthogonal Matching Pursuit (OMP), i.e., linear probing with a sparsity constraint.
   Context: Linear probing has been widely used to study representations in other domains, but not in VRL.
2. The results of linear probing show that well- and poorly performing agents differ primarily in their later MLP layers, but not their earlier CNN layers.
   Context: This is counter to intuition that the CNN layers are primarily responsible for representation learning.
3. We show that the linear probing results are predictive of agent quality after retraining.
   Context: Linear probing has been questioned because it assumes the features are linearly accessible from learned representations.
4. We identify that the MLP layers of a poorly performing agent suffer from action collapse, a failure mode where all inputs to a network produce the same output.
   Context: Dormant and dead neurons have previously been observed in VRL (Xu et al., 2023), but action collapse is a more precise understanding of this failure mode.
5. By studying the metrics associated with action collapse, we show that it can be avoided with a simple rule.
   Context: None

Understanding Learned Representations and Action Collapse in Visual Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

In contrast to deep learning models trained with supervised data, visual reinforcement learning (VRL) models learn to represent their environment implicitly via the process of seeking higher rewards. However, there has been little research on the specific representations VRL models learn. Using linear probing, we study the extent to which VRL models learn to linearly represent the ground truth vectorized state of an environment, on which layers these representations are most accessible, and how this relates to the reward achieved by the final model. We observe that poorly performing agents differ substantially from well-performing ones in the representation learned in their later MLP layers, but not their earlier CNN layers. When an agent is initialized by reusing the later layers of a poorly performing agent, the result is always poor. These poorly performing agents end up with no entropy in their actor network output, a phenomenon we call *action collapse*. Based on these observations, we propose a simple rule to prevent action collapse during training, leading to better performance on tasks with image observations with no additional computational cost.

1 Introduction

Visual reinforcement learning (VRL) trains agents to learn effective policies directly from raw image observations in domains such as Atari games and continuous control tasks (Schrittwieser et al., 2020; Badia et al., 2020; Yarats et al., 2021a; Zheng et al., 2023). Despite these impressive advances, our understanding of how these agents internally represent their environments remains limited. Examining the learned representations is vital for two reasons: first, interpretable features explain why an agent makes a specific decision, helping trust and debugging; second, understanding how representations evolve during training can reveal possible failure modes caused by misrepresentations of key information about the environment.

While earlier works have investigated internal representations in RL (Greydanus et al., 2018; Wijmans et al., 2023; Dabney et al., 2021; Wang et al., 2022; Zahavy et al., 2016), many of them focus on task-specific features that require careful designs. For example, Greydanus et al. (2018) finds that agents learn where-to-look in Atari games and Wijmans et al. (2023) demonstrates that map-like structures emerge in learned representations in navigation tasks. However, in VRL we can directly compare what the agent learns to the vectorized environment state to analyze how agents learn from raw image observations.

Our work starts by investigating how well a model learns vectorized environment states from image observations using linear probing with a sparsity constraint, specifically, Orthogonal Matching Pursuit (OMP) (Pati et al., 1993). In addition to confirming models learn vectorized states and where they are learned, our results show an interesting insight: later MLP layers of poorly performing agents do not learn representations as well as well-performing agents, while the learned representations in earlier CNN layers are almost identical for all agents. This suggests that poorly performing agents struggle in their MLP, where learned representations in CNN fail to propagate. However, since OMP assumes that the features are linearly accessible from learned representations, it might not tell us how well the vectorized environment states are captured by the model. Based on the idea that better representations should lead to better agent performance, we retrain the agents from different initializations to confirm that, indeed, the MLP is the source of the issue and the CNNs are interchangeable.

After identifying MLP as the source of failure, we analyze what occurs in these MLPs. According to the videos, poorly performing agents repeatedly take the same actions regardless of the input observation, a phenomenon we call *action collapse*. With the hypothesis that MLP representations also collapse to a single point, we measure numerical rank, gradient norm, and other metrics that can reflect the extent to which the MLP representations collapse. In addition to confirming that hypothesis, we observe that if an agent does not escape action collapse before exploration noise decays to its minimum, it is unlikely to recover. Equipped with this insight, we propose a simple rule to prevent action collapse. We validate its effectiveness across multiple tasks.

To summarize, our workflow serves as an example of how analyzing learned representations can help identify issues and develop solutions based on observed patterns. The key contributions of this work can be summarized below:

- We are the first to use linear probing to analyze how well RL models learn vectorized environment states from image observations.
- We confirm that linear probing is effective for studying whether RL models learn vectorized environment states despite its strict assumptions.
- By analyzing model metric patterns, we derive a simple rule to help agents escape action collapse.

In Section 2 and 3, we review related work and introduce necessary preliminaries. In Section 4, we first conduct a representation analysis using OMP, identifying the key difference between well-performing and poorly performing agents in their MLP layers, and then verify that the observations obtained from linear probing are reasonable through control experiments. Section 5 further examines action collapse by checking various related metrics, revealing that poorly performing agents lose diversity in their actions due to collapsed learned representations, and based on these findings, we propose a simple rule to help agents escape action collapse and validate its effectiveness empirically.

2 Related Work

Understanding Representations Learned in RL. Understanding the representations learned by RL agents is crucial for interpreting decision-making processes and improving performance (Greydanus et al., 2018; Wijmans et al., 2023; Dabney et al., 2021; Wang et al., 2022; Zahavy et al., 2016). Greydanus et al. (2018) proposes a method for generating saliency maps to show which specific regions Atari agents focus on, revealing the evolution of agent attention throughout training. Wijmans et al. (2023) finds the emergence of spatial representations in blind navigation agents, demonstrating that the agents develop map-like structures in memory to support navigation tasks even without direct visual information. Studying representations in VRL is a natural idea because we can compare what is learned by the agent to the vectorized environment states, a near-ground-truth representation of the environment. Our work is the first to explore how agents learn those states from image observations.

Representation Studies Using Linear Probing. Linear probing has been widely applied in different domains to investigate what models learn (Alain & Bengio, 2016; Belinkov, 2022). For example, some use linear probing to see if the hidden layer activations of models capture class-specific features in classification tasks (Chen et al., 2023; Morcos et al., 2018), while others employ it to uncover how and where linguistic or conceptual variables are encoded in (large) language models (Hewitt & Manning, 2019; Peters et al., 2018; Adi et al., 2016; Nanda et al., 2023; Zhao et al., 2024; Singh et al., 2024). However, the application of linear probing assumes that features are linearly accessible from hidden vectorized activations, which may overlook more complex, nonlinear relationships (Shen & Younes, 2024; White et al., 2021). Our work is the first to study the learning of vectorized environment states during RL training from image observations using linear probing in Section 4.1, and we prove the effectiveness of linear probing by retraining the agents from different initializations in Section 4.2.

Visual Reinforcement Learning. There are many powerful algorithms to train RL agents that receive image observations including CURL (Laskin et al., 2020), DrQ (Yarats et al., 2021b), DrQ-v2 (Yarats et al., 2021a), TACO (Zheng et al., 2023), and DrM (Xu et al., 2023). Our studies focus on DrQ-v2 because of its better performance compared with CURL and DrQ and its simpler design compared with TACO and DrM, making it the best place for us to study representations learned by the agents. Xu et al. (2023) observe the consequences of action collapse when they remark that agents take repeated identical actions. They prevent agents from getting stuck by proposing a metric called dormant ratio thatmeasures the fraction of inactive neurons, and they tune the exploration in refenence to that metric. We provide a more precise analysis of the action collapse phenomena. We discuss further differences in Supplementary Materials A.

3 Preliminaries

Markov Decision Problem. We consider a Markov Decision Process (MDP) (Puterman, 2014) defined by $\langle O, S, A, R, P, \beta, \gamma \rangle$, where O is the observation space (a three-stack of images), S is the vectorized environment state space, A is the action space, $R : S \times A \to \mathbb{R}$ is the reward function, $P : S \times A \to \Delta S$ is the transition function, $\beta \in \Delta S$ is the initial state distribution, and $\gamma \in [0, 1]$ is the discount factor. In VRL, the agent only observes O, but S is used to compute the reward, and we assume that S can be inferred from O. At timestep t, we denote the observation as $\boldsymbol{o}_t$, the state as $\boldsymbol{s}_t$, the action as $\boldsymbol{a}_t$, and the reward as $r_t = R(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The objective is to find a policy $\pi : O \to \Delta A$ that maximizes the expected discounted return, i.e., $\pi^* \in \arg\max_\pi \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(\boldsymbol{s}_t, \boldsymbol{a}_t)]$.

Orthogonal Matching Pursuit. Orthogonal Matching Pursuit (OMP) (Pati et al., 1993) is an algorithm that reconstructs signals by iteratively selecting the feature most correlated with the current residual and then updating the residual to remove its influence. By enforcing a sparsity constraint, OMP effectively eliminates irrelevant dimensions, reducing overfitting when handling high-dimensional inputs, where it typically outperforms unconstrained linear probing by maintaining more robust and interpretable results.

DrQ-v2. DrQ-v2 uses a convolutional encoder to transform augmented image observations into a low-dimensional latent space that feeds both an actor network and two critic networks. Its training follows an off-policy DDPG-based scheme. We analyze activations from the hidden layers, specifically the four CNN layers in the encoder and the three MLP layers in the actor, in addition to considering the raw image inputs.

4 Representation Studies

In this section, we study if the VRL model learns the representations of vectorized environment states using linear probing and validate the observations obtained from linear probing through controlled experiments. In Section 4.1, by probing vectorized environment states with hidden layer activations, we find high agent performance correlates with better representations in MLP while CNNs are very similar between well-performing and poorly performing agents. In Section 4.2, we retrain agents starting from different initializations and show that better representations, as measured by linear probing, lead to better performance after retraining.

4.1 Study 1: do VRL agents learn the linear representations of vectorized environment states and where are they learned?

Main Question. In VRL environments, the reward function is usually defined in reference to the vectorized environment state. Therefore, it is natural to think that an agent trained with image observations should learn to represent the vectorized environment state components in order to learn a good policy that achieves high rewards. Furthermore, we hypothesized that these representations are mainly learned in the CNN layers, and MLP is responsible for transforming the learned representations into actions in the actor, which has been a thoroughly studied question in the domain of classification problems (Zeiler & Fergus, 2014; Bau et al., 2017). Because learning good representations is important, we hypothesized that agents' differing performances are reflected in their ability to linearly represent the vectorized environment state.

Setup. We use DeepMind Control Suite (DMC) (Tassa et al., 2018) that provides the environments with image observations. Among the benchmark methods working with DMC, we select DrQ-v2 (Yarats et al., 2021a) because of its high performance and simple designs discussed in Section 2 so this work focuses on analyzing the behaviors of DrQ-v2 in DMC. In this section, we study walker_walk as an instance, which is to train a bipedal agent to walk forward fast and stably. We choose this task because its reward is clearly defined regarding some components of vectorized environment states, which makes it easy for us to target the most critical state components. Trained agents can be classified into three classes referring to their performance: most of them are either well-performing agents achieving more than 900 rewards or poorly performing agents achieving less than 30 rewards, and only a few are middle agents between them. However, we note that all agents were trained under the exact same conditions, except for randomness in initialization and exploration.

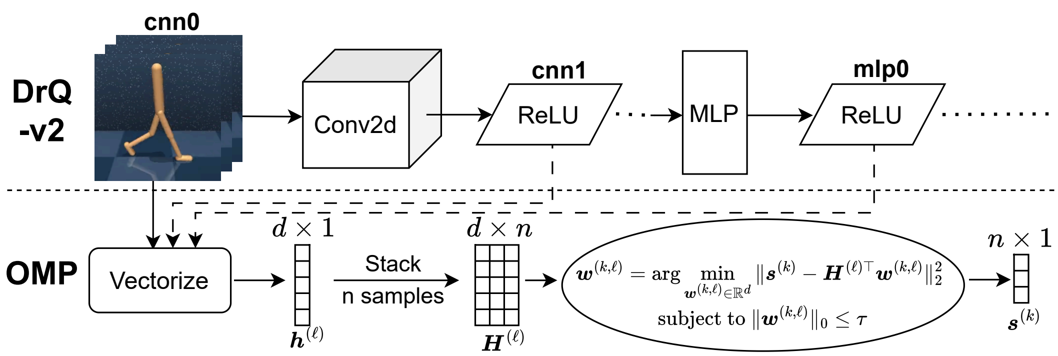

Figure 1: We apply OMP to probe the vectorized environment states using the hidden layer activations. In the top half, the input to DrQ-v2 is three consecutive frames, followed by four 2D convolution layers and three MLP layers. OMP tries to fit $\boldsymbol{w}^{(k,\ell)}$ to linearly probe a vectorized environment state component $\boldsymbol{s}^{(k)}$ using a given layer's activations as input, subject to a sparsity constraint

Method. As we discussed in Section 2, linear probing is a widely used tool for studying representations. In our case, we use the ℓ-th layer activations, denoted by $\boldsymbol{h}^{(\ell)} \in \mathbb{R}^d$, to probe each vectorized environment state $s^{(k)}$ where k refers to the k-th component. Following the linear representation hypothesis (Elhage et al., 2022), where features are represented by direction in the activation space if the vectorized environment states $s^{(k)}$ are learned in the ℓ-th layer, we may assume $\boldsymbol{h}^{(\ell)} \approx \sum_k s^{(k)} \boldsymbol{w}^{(k,\ell)} + \boldsymbol{e}$, where $\boldsymbol{w}^{(k,\ell)} \in \mathbb{R}^d$ represents the direction for $s^{(k)}$ and $\boldsymbol{e} \in \mathbb{R}^d$ denotes the residual containing other features. Further assuming that the features are represented as orthogonal directions gives $s^{(k)} \approx \boldsymbol{w}^{(k,\ell)\top} \boldsymbol{h}^{(\ell)}$. In practice, without knowing $\boldsymbol{w}^{(k,\ell)}$, we first estimate it from training samples. Specifically, given n training samples, we collect all the ℓ-th layer activation vectors $\boldsymbol{H}^{(\ell)} = \begin{bmatrix} \boldsymbol{h}_1^{(\ell)} & \cdots & \boldsymbol{h}_n^{(\ell)} \end{bmatrix} \in \mathbb{R}^{d \times n}$, the corresponding k-th state $\boldsymbol{s}^{(k)} = \begin{bmatrix} s_1^{(k)} & \cdots & s_n^{(k)} \end{bmatrix}^\top$, and then learn $\boldsymbol{w}^{(k,\ell)}$ by solving the minimizing the following mean squared error (MSE)

$$\widehat{\boldsymbol{w}}^{(k,\ell)} = \underset{\boldsymbol{w}^{(k,\ell)} \in \mathbb{R}^d}{\arg\min} \|\boldsymbol{s}^{(k)} - \boldsymbol{H}^{(\ell)\top} \boldsymbol{w}^{(k,\ell)}\|_2^2, \quad \|\boldsymbol{w}^{(k,\ell)}\|_0 \leq r, \tag{1}$$

where we have included a sparsity constraint to avoid overfitting since d is often very large. We use orthogonal matching pursuit (OMP) (Pati et al., 1993; Tropp & Gilbert, 2007) to solve the above problem, with its pipeline shown in Figure 1, and use the validation approach to select the best sparsity τ for each layer and state component. We then evaluate the probing MSE $(s'^{(k)} - \widehat{\boldsymbol{w}}^{(k,\ell)\top}\boldsymbol{h}'^{(\ell)})^2$ over testing samples to measure how well the learned activation vector $\boldsymbol{h}'^{(\ell)}$ in the ℓ-th layer for reconstructing the k-th state component $s'^{(k)}$. All the samples in three sets are collected from well-performing agents.

In Figure 2, we plot the probing MSE for three environment state components: torso velocity, joint orientation, and torso height. We choose these three components because all of them are the input to the reward function. We let cnnℓ refer to the layer activations after the ℓ-th CNN layer (with cnn0 being the raw image input), and mlpℓ refer to the layer activations after the ℓ-th MLP layer. We compute the average MSE for well-performing agents (first row) and poorly performing agents (second row). In all figures, the probing MSE using raw images (cnn0) serves as a baseline to measure the relative difficulty of probing that state component. The total number of training frames is $1.1e6$ and there is an evaluation every $1e4$ frames, resulting in $x \in [0, 109]$ on the x-axes for all the figures regarding VRL training.

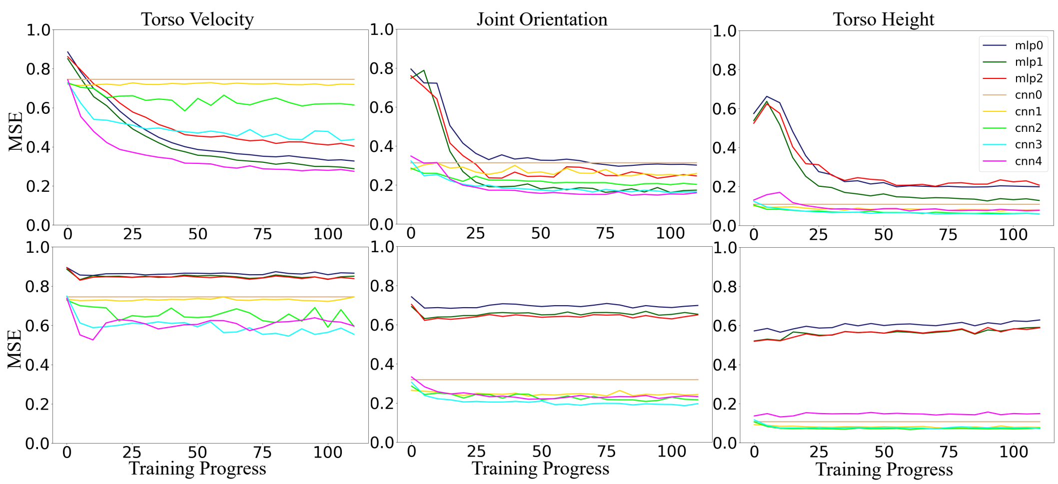

Figure 2: The MSE of probing different vectorized state components in different layers in evaluation. The first row is the 13/20 well-performing agents that achieve more than 900 episode reward. The second row is the 6/20 poorly performing agents achieving less than 30 episode reward.

Observation 1: Well-performing agents gradually learn the vectorized environment states during training. If we study the overall trend of MSE of well-performing agents (first row in Figure 2), we see that the MSE in all layers decreases over training. This indicates that well-performing agents are learning to linearly represent the vectorized environment states over training.

Observation 2: The last CNN layer (cnn4) learns vectorized environment states the best. If we focus on the order of MSE among different layers of well-performing agents (first row in Figure 2), there is a decrease from the earlier CNN layers to the later CNN layers. Deeper CNN layers are more expressive and represent more complex features. The MSE differences among CNN layers are very small when probing the torso height (third column), which seems to be because this component is easy to linearly extract, even from the raw image. Joint orientation is harder to extract and velocity is the hardest.

The MSE in later MLP layers is higher than the one in cnn4. Our interpretation to this phenomenon is that the responsibility of MLP layers is to generate actions compared with the previous layers because they are deeper in the actor. Therefore, the features in later MLP layers have experienced non-linear transformation from the representations learned in the previous layers to generate actions so we cannot probe the vectorized environment states as well as cnn4.

Observation 3: The MSE in MLP of poorly performing agents does not appear to decrease during training, but the MSE in the CNNs resembles that of well-performing agents. Comparing the first and second rows, we see a clear difference in the probing results, but that difference primarily exists in MLP, which is surprising. The only exception is the cnn4 for torso velocity—perhaps the most difficult features are harder to learn with a poor agent.

The results are consistent with our hypotheses in Observations 1 and 2, but Observation 3 is unexpected. It appears that the CNN of poorly performing agents learns about the environment, but the representations fail to transfer to MLP. We conduct an additional study to understand this better.

4.2 Study 2: Is the representation quality measured by OMP is predictive of agent performance after retraining?

Main Question. Using linear probing to understand learned representations assumes features can be *linearly* extracted from them. If linear probing MSE really tells us the representation quality, then we might conjecture that a model trained from initialized CNN/MLP with a smaller MSE should perform better than one with a larger MSE because better representation quality of the initialized model should lead to better agent performance after retraining. From Study 1, we hypothesize that the initialized MLP has more influence than the initialized CNN on performance after retraining because of the larger difference in the MSE in MLP between well-performing and poorly performing agents.

Setup. We extract the CNN and MLP from two random well-performing agents, which we regard as "good" CNN and MLP. We also extract "bad" CNN and MLP from two poorly performing agents. Adding blank CNN and MLP, we arrive at nine combinations of CNN × MLP ({blank, good, bad} CNN × {blank, good, bad} MLP) for initialization. For each combination, we show the number of successful agents (out of 20 runs) and their evaluation curves in each setting in Figure 3.

(a)

	MLP		
CNN	blank	good	bad
blank	13	16	0
good	16	20	0
bad	11	20	0

(b)

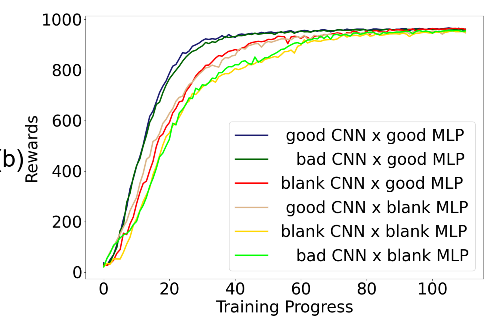

Figure 3: (a) The number of successful (rewards >= 900) agents out of 20 runs trained from {blank, good, bad} CNN × {blank, good, bad} MLP; (b) The evaluation performance of the successful agents in each setting during the training.

Observation 1: The initialized MLP, no matter if it is good or bad, has a much stronger influence on the performance after retraining than the initialized CNN. According to Figure 3(a), it is striking that no matter what the initialized CNN is, no agents starting from bad MLP (third column) succeed. Also, the agents initialized with good MLP (second column) achieve a high successful ratio. In comparison, the influence of the initialized CNN is less although we can still tell an advantage of good CNN (second row) over blank CNN (first row). This observation not only confirms that the MSE of OMP in Study 1 is a meaningful metric that indicates the representation quality but validates the issue of poorly performing agents is indeed in their MLP.

Observation 2: The training speed of successful agents aligns with the fraction of successful agents. In line with Figure 3(a), the settings resulting in large successful ratios (e.g., good CNN × good MLP) also lead to a high convergence speed in Figure 3(b). This observation further strengthens the conclusions we make at the end of Observation 1 from the perspective of the convergence speed.

In this study, we prove the usefulness of OMP in studying representations in RL and confirm the problem of poorly performing agents originates in their MLP. However, why the bad MLP is not recoverable (third column in Figure 3(a)) and why the MLP of poorly performing agents loses the ability to probe the vectorized environment states we observe in Section 4.1 are unanswered. In the next section, we study these questions and propose a resolution to prevent these training failures.

5 Action Collapse Analysis

We visually inspected the videos of poorly performing agents, whose frames are in Supplementary Materials C. They appear to take the same action no matter what the image observations, which we verify by checking the actions generated by the actor. We call this phenomenon *action collapse*. To further understand it, we check various metrics of MLP that might explain the same action generation in Section 5.1. In Section 5.2, we propose a simple rule based on our analysis, improving performance by helping agents escape action collapse.

5.1 What happens in MLP

Hypotheses. Action collapse appears similar to the neuron collapse encountered in supervised classification problems, where the representations in the last MLP layer collapse to a single point for the samples of the same class (Zhu et al., 2021; Papyan et al., 2020; Mixon et al., 2022; Lu & Steinerberger, 2022; Ji et al., 2021). Although the contexts differ, same class labels in classification correspond to the same actions in RL. Therefore, it is reasonable to hypothesize the issue in the MLP of poorly performing agents is the collapsed representations. To validate that hypothesis, we compute various potentially related metrics of MLP layers, which are followed by the micro-hypothesis for each metric below.

- **Numerical Rank.** To measure the dimensionality of the hidden-layer representations, we compute the numerical rank (Zhou et al., 2022) of the matrix of MLP activiations $\boldsymbol{H}^{(\ell)}$, as defined in Figure 1, using $\widetilde{\text{rank}}(\boldsymbol{H}^{(\ell)}) = \frac{(\sum_i \sigma_i)^2}{\sum_i \sigma_i^2}$, where $\{\sigma_i\}$ denote the singular values of $\boldsymbol{H}^{(\ell)}$. This numerical rank benefits from discounting small singular values when the matrix is close to low-rank but has small singular values, serving as a stable measure for the rank of a matrix. We expect to see low numerical ranks in MLP layers of a poorly performing agent.
- **Correlation & Zeros.** Other two metrics that can reflect the similarity of MLP activations are the average correlation among every two non-zero activations and the number of zero activations, also called dead neurons (Lu et al., 2019). As DrQ-v2 uses ReLU (Nair & Hinton, 2010) as the activation function, we consider the possible contribution of dead neurons to similar features.[1] We hypothesize that they are both high for a poorly performing agent.
- **Gradient Norm.** As the poorly performing agent suffers from action collapse continuously, we think the model might not get updated, so we check the gradient norm of MLP layers. It is expected that the gradient norm is always around zero when an agent gets stuck in action collapse.

We check the metrics studied in Figure 4 in other environments in Supplementary Materials D. Note that the dimension of mlp0 is 50, which is not on the same scale as mlp1 and mlp2 with 1024 dimensions. For clarity, we only visualize the metrics of mlp1 and mlp2 in Figure 4 although the trend of those metrics of mlp0 is similar to the ones of mlp1 and mlp2.

Observation 1: All the metrics of a poorly performing agent (third row) agree with our hypotheses, and the metrics are correlated with each other. The features learned in the MLP of poorly performing agents also collapse to a single point and their MLP does not update as we dis-

[1] We tried substituting all the ReLU with Leaky-ReLU (Maas et al., 2013) where the neurons cannot die, but it did not resolve action collapse.

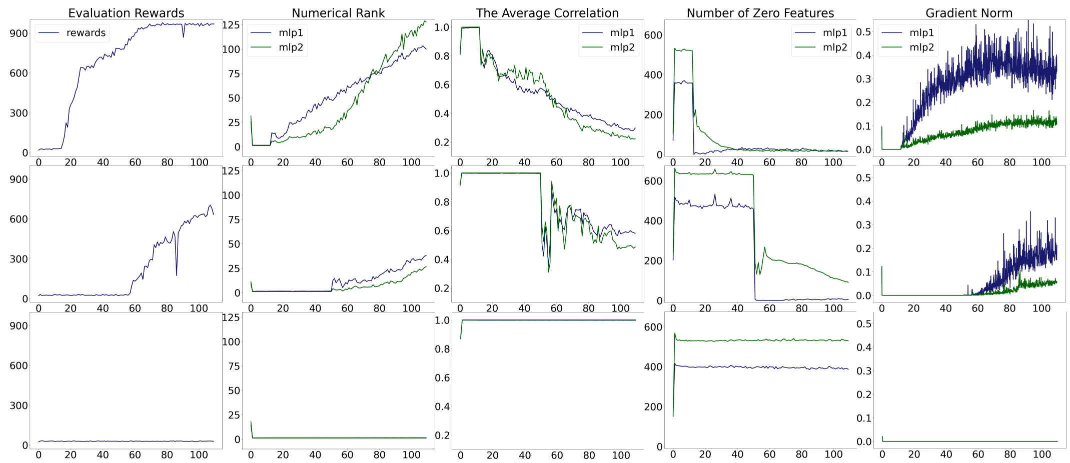

Figure 4: The metrics of three agents, where the x-axis represents the training progress. The first row is a randomly picked well-performing agent achieving larger than 900 rewards. The second row is a middle agent between goodness and badness. The third row is a randomly picked poorly performing agent achieving lower than 30 rewards.

cuss above. We also tell the correlations among those metrics in the first and second rows because they experience abrupt changes at the same time ($x = 13$ and $x = 51$ respectively), which can be used as the signal determines the model has escaped action collapse in the next part.

Observation 2: Even a well-performing agent (first row) initially experiences action collapse. Surprisingly, the metrics of a well-performing agent (first row) are almost the same as a poorly performing one before $x = 13$. It appears that all agents must experience this period in this setup. What distinguishes a well-performing agent from a poorly performing one is that it escapes from action collapse successfully.

Observation 3: It is rare to encounter a middle agent (second row) and middle agents are those that escape action collapse slowly. The middle agent is the only one between well- and poorly performing agents among 20 runs. When action collapse hits, exploration is the only factor that can bring valuable samples with gradients to the actor update. DrQ-v2 uses an exploration noise that decreases with training steps, which becomes the smallest (but not zero) when $x = 20$. We observe that it is rare to escape action collapse after this exploration period has ended, which informs our escape rule in the next section.

5.2 A Simple Rule for Escaping Action Collapse

There are two insights we gain from Section 5.1 inspiring the design of our proposed rule. First, according to Observation 1, several metrics (second to fifth columns in Figure 4) can be used to tell if an agent escapes action collapse or not. Second, it is almost impossible for a poorly performing agent to recover from action collapse after $x = 20$ ($2e5$ frames) in the light of Observation 3.

Inspired by the two insights above, we choose to use the gradient norm of mlp1 to determine whether an agent suffers from action collapse and we would reset the training from scratch every $2e5$ frames if an agent is continuously stuck in action collapse. To develop our simple rule for preventing the agent from sticking in action collapse by combining the two choices together, if we do not see a gradient norm larger than 0.001 of consecutive five computations every $2e5$ frames, the training would be reset; once we see that, the training continues until the end ($1.1e6$ frames). Based on the rule we describe above, we formulate the following Proposition 1 to illustrate the effectiveness of our proposed rule, which is intuitive.

Proposition 1 *Supposing the number of training frames is $n \cdot m$ (m is the frames where the exploration decreases to the smallest) and the probability of getting stuck in action collapse is p, then the probability of escaping action collapse when applying our proposed rule is $1 - p^n$.*

As we mention in Observation 3 in Section 5.1, DrQ-v2 sets the exploration noise for each task. In a task whose $n \cdot m$ equals $1.1e6$, m is set to $2e5$. Therefore, in such environments, the probability that an agent escapes action collapse is larger than $1 - p^5$. Referring to Figure 3, we determine what needs to be reset to escape action collapse. Whenever a reset is needed, we treat the current model as a bad CNN $\times$ bad MLP. Thus, there are four possible CNN $\times$ MLP combinations available to us: $\{$blank, bad$\}$CNN $\times$ $\{$blank, bad$\}$MLP. To have any chance of escaping action collapse, we must reset the bad MLP (third column in Figure 3) to a blank one (first column). By comparing the successful rates of blank CNN (first row) and bad CNN (third row) in the first column, we see that the bad CNN should also be reset to a blank one. We show the results of experiments in different environments in the next paragraph.

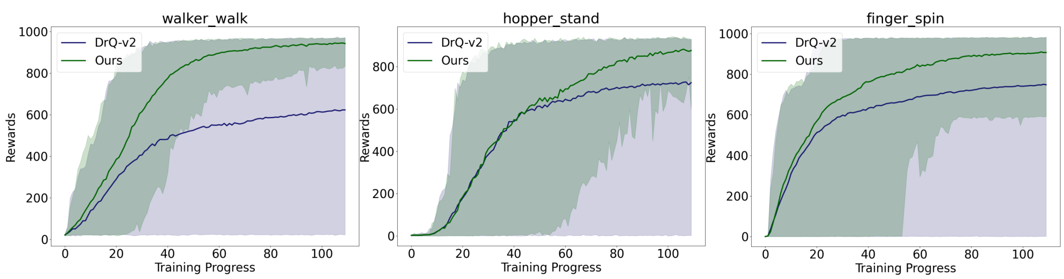

Figure 5: The performance comparison of 40 runs between DrQ-v2 and DrQ-v2 plus our rule in different environment_task. Shaded areas represent the gap between the 5th percentile and the 95th percentile.

Results. To cover broad tasks to validate the wide existence of action collapse and the effectiveness of our rule, we randomly select three kinds of tasks including locomotion (walker_walk), living (hopper_stand), and manipulation (finger_spin) tasks. In Figure 5, we can see a clear performance improvement while applying the rule. The rule does not hurt the performance in the 95th percentile, but boosts it in the 5th percentile significantly according to the shaded areas in Figure 5. Moreover, the average standard deviation over the last ten evaluations drops a lot, from 418.2 to 38.2, from 359.5 to 93.8, and from 362.7 to 116.2 for the three tasks, respectively. In addition, we count the number of runs stuck in action collapse continuously. There are 13, 8, and 7 runs out of 40 respectively for DrQ-v2. In comparison, all 40 runs of ours escape action collapse, showing the success of the simple rule.

6 Conclusion

In this work, we investigated how VRL agents learn the vectorized environment states, a unique advantage of VRL environments that has not been utilized in prior work. Using the linear probing technique, we identify that a problem occurs in the MLP layers of poorly performing agents, which can not be inferred from agent performance alone. This highlights the potential of studying learned representations to diagnose failure modes in RL. A separate stream of evidence based on retraining agents using supports the linear representation hypothesis in the context of VRL. After localizing the failure mode to the MLP, we show that it is due to collapsed representations and relate it to the parallel phenomenon of neural collapse that has been observed in supervised classification tasks. Moreover, we show that action collapse can be detected during training, and we develop a simple rule to prevent it, which works well despite its simplicity. That provides strong evidence that studying representations can offer insights for improving agent performance.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*, 2016.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *International conference on machine learning*, pp. 507–517. PMLR, 2020.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. DOI: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Yiting Chen, Zhanpeng Zhou, and Junchi Yan. Going beyond neural network feature similarity: The network feature complexity and its interpretation using category theory. *arXiv preprint arXiv:2310.06756*, 2023.

Will Dabney, Mark Rowland, Marc Bellemare, and Remi Munos. Representation learning in reinforcement learning: Empirical methods and a taxonomy. *AAAI Conference on Artificial Intelligence*, 35(13):11577–11585, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1787–1796. PMLR, 2018.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. DOI: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419/.

Wenlong Ji, Yiping Lu, Yiliang Zhang, Zhun Deng, and Weijie J Su. An unconstrained layer-peeled perspective on neural collapse. *arXiv preprint arXiv:2110.02796*, 2021.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5639–5650. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/laskin20a.html.

Jianfeng Lu and Stefan Steinerberger. Neural collapse under cross-entropy loss. *Applied and Computational Harmonic Analysis*, 59:224–241, 2022.

Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, pp. 3. Atlanta, GA, 2013.

Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):11, 2022.

Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*, 2023.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44. IEEE, 1993.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. DOI: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202/.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Sheng Shen and Rabih Younes. Reimagining linear probing: Kolmogorov-arnold networks in transfer learning. *arXiv preprint arXiv:2409.07763*, 2024.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

Qiyang Wang, Xuhong Xu, Yang Yu, Hao Qian, and Wenming Huang. Understanding representation learning in deep reinforcement learning through features and transferable knowledge. *arXiv preprint arXiv:2203.15955*, 2022.

Jennifer C. White, Tiago Pimentel, Naomi Saphra, and Ryan Cotterell. A non-linear structural probe. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 132–138, Online, June 2021. Association for Computational Linguistics. DOI: 10.18653/v1/2021.naacl-main.12. URL https://aclanthology.org/2021.naacl-main.12/.

Erik Wijmans, Manolis Savva, Irfan Essa, Stefan Lee, Ari S Morcos, and Dhruv Batra. Emergence of maps in the memories of blind navigation agents. *AI Matters*, 9(2):8–14, 2023.

Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. *arXiv preprint arXiv:2310.19668*, 2023.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Moritz Zahavy, Nir Ben-Zrihem, and Shie Mannor. Graying the black box: Understanding dqns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2), February 2024. ISSN 2157-6904. DOI: 10.1145/3639372. URL https://doi.org/10.1145/3639372.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. $\texttt{TACO}$: Temporal latent action-driven contrastive loss for visual reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=ezCsMOy1w9.

Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. In *International Conference on Machine Learning*, pp. 27179–27202. PMLR, 2022.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Two key insights identified in this work

There are mainly two key messages learned by us but are not known to DrM (Xu et al., 2023).

First, by analyzing representations learned by RL models, we find the issue of the agents stuck in action collapse exists in their MLP in Section 4.1. In Section 5.1, we further realize the representations learned in MLP also collapse based on relevant metrics. Without representation studies, DrM does not uncover what is behind inactive neurons.

Second, DrM proposes the dormant ratio to measure inactive neurons and rely on that metric to tune exploration and exploitation, but that metric might not work as expected. In our representation study specific to MLP in Section 5.1, we learn that even active neurons in a stuck agent collapse to a single point (third column in Figure 4). If inactive neurons become active but the average correlation stays high, the dormant ratio increases, yet the agent remains stuck. This suggests proposing a method based only on phenomena, without further analysis of the underlying causes, might not be unreliable.

B Detailed analysis of probing MSE orders of MLP layers when probing vectorized environment states using OMP

When we analyze the orders of MLP layers in Figure 2, we observe several patterns between two consecutive layers:

- **From cnn4 to mlp0, MSE increases.** The number of activations decreases from 39200 to 50 from cnn4 to mlp0. This design likely aims to concatenate low-dimensional activations with low-dimensional actions. However, this reduction makes the activations in mlp0 more abstract, making it harder to linearly probe vectorized environment states from them.
- **From mlp0 to mlp1, MSE decreases.** The number of activations expands from 50 to 1024. We suspect that this increase improves interpretability, making the activations more suitable for linear probing.
- **From mlp1 to mlp2, MSE increases again.** Mlp2 is the last MLP layer before action generation. We hypothesize that its activations become more complex as the model non-linearly transforms the representations (e.g., velocity) interpretable to humans into a form suitable for action prediction. This transformation may be necessary for the actor to produce actions using only a final linear layer on top of mlp2.

C Video frames of well-performing agents and poorly performing agents sticking in action collapse

The second and third rows in Figure 6 show that the poorly performing agent repeatedly takes the same action, regardless of the image observations. Similarly, in Figure 7, the poorly performing agent (the finger on the left) quickly gets stuck after the environment resets.

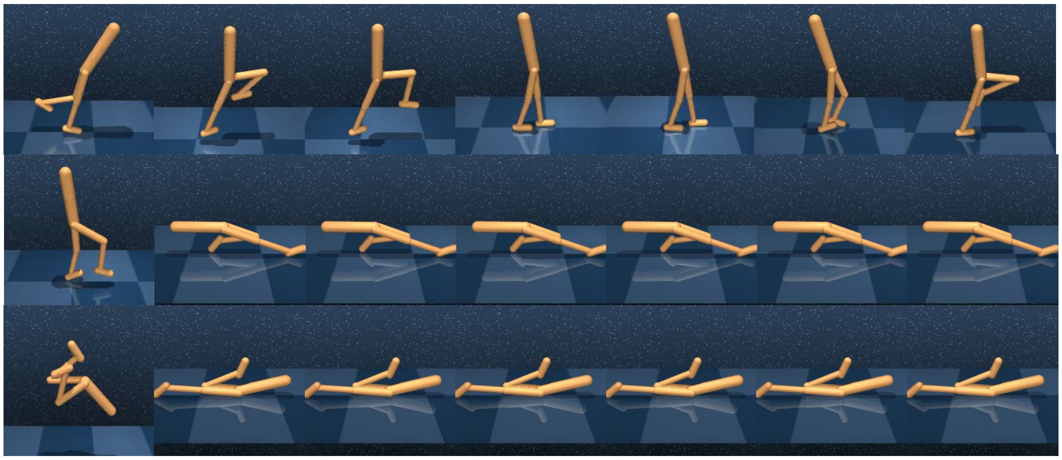

Figure 6: The video frames from 0 to 300 frames collected from a well-performing agent (first row) and a bad one (second and third rows with different environment initialization after reset) in walker_walk.

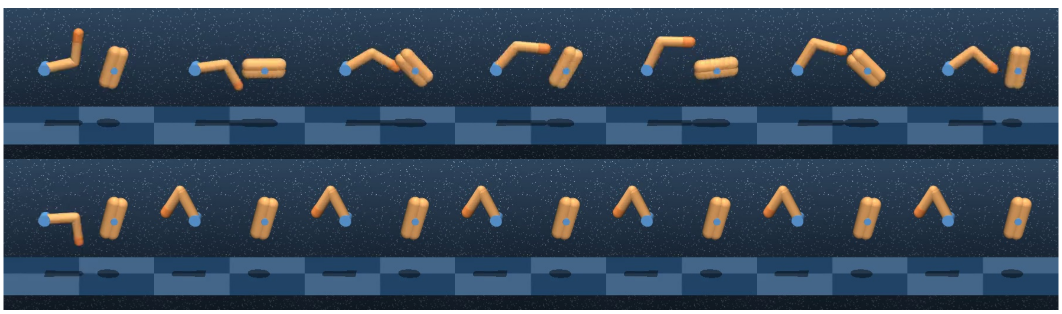

Figure 7: The video frames from 0 to 300 frames collected from a well-performing agent (first row) and a bad one (second row) in finger_spin.

D The metrics discussed in the main paper in other environments

Figures 8 and 9 analyze the same metrics as in Section 5.1 for two other tasks, hopper_stand and finger_spin. These results, along with Figure 4, support our hypotheses from Section 5.1.

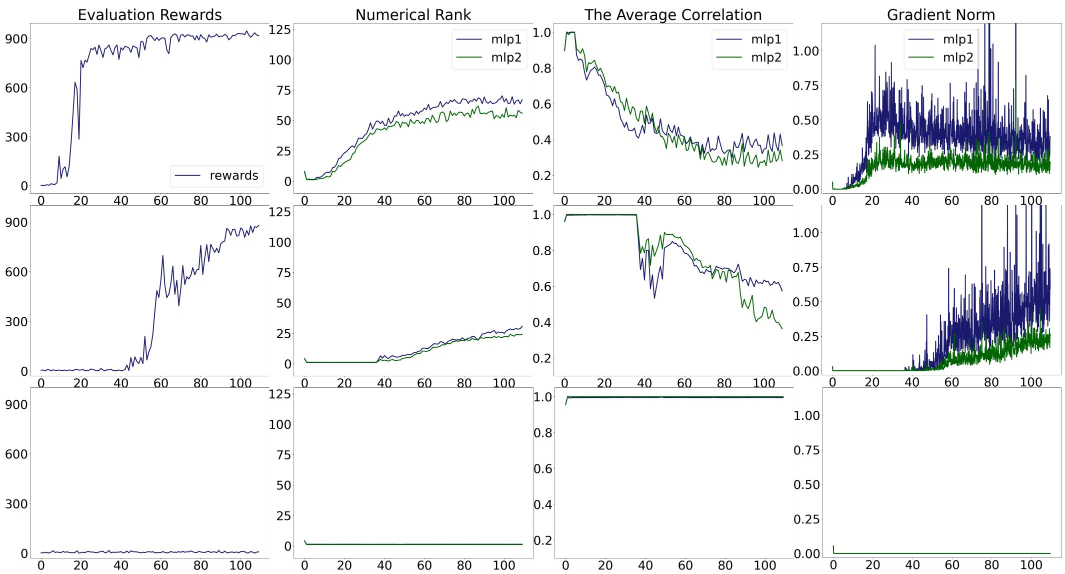

Figure 8: The metrics of three agents in hopper_stand, where the x-axis represents the training progress. The first row is a randomly picked well-performing agent achieving larger than 900 rewards. The second row is a middle agent between goodness and badness. The third row is a randomly picked poorly performing agent achieving lower than 30 rewards.

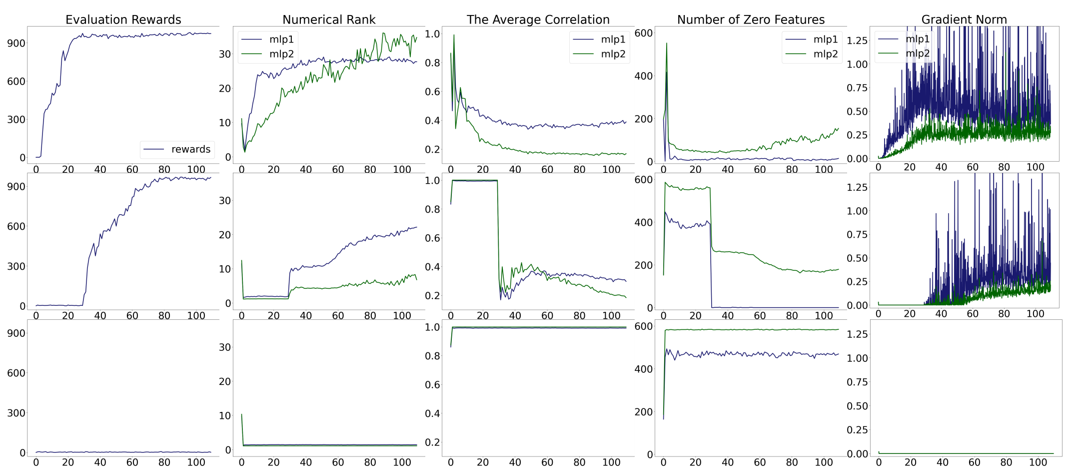

Figure 9: The metrics of three agents in finger_spin, where the x-axis represents the training progress. The first row is a randomly picked well-performing agent achieving larger than 900 rewards. The second row is a middle agent between goodness and badness. The third row is a randomly picked poorly performing agent achieving lower than 30 rewards.