OBSERVATIONAL OVERFITTING IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A major component of overfitting in model-free reinforcement learning (RL) involves the case where the agent may mistakenly correlate reward with certain spurious features from the observations generated by the Markov Decision Process (MDP). We provide a general framework for analyzing this scenario, which we use to design multiple synthetic benchmarks from only modifying the observation space of an MDP. When an agent overfits to different observation spaces even if the underlying MDP dynamics is fixed, we term this *observational overfitting*. Our experiments expose intriguing properties especially with regards to *implicit regularization*, and also corroborate results from previous works in RL generalization and supervised learning (SL).

1 INTRODUCTION

Generalization for RL has recently grown to be an important topic for agents to perform well in unseen environments. Complication arises when the dynamics of the environments entangle with the observation, which is often a high-dimensional projection of the true latent state. One particular frame work, which we denote the *zero-shot supervised framework* (Zhang et al., 2018a;c; Nichol et al., 2018; Justesen et al., 2018) used to study RL generalization is to treat it analogous to a classical supervised learning (SL) problem – i.e. assume there exists a distribution of MDP's, train jointly on a finite "training set" sampled from this distribution, and check expected performance on the entire distribution, with the fixed trained policy. In this framework, there is a spectrum of analysis, ranging from almost purely theoretical analysis (Wang et al., 2019; Asadi et al., 2018) to full empirical results on diverse environments (Zhang et al., 2018c; Packer et al., 2018).

However, there is a lack of analysis in the middle of this spectrum - previous theoretical works lack analysis for the case when the underlying MDP is relatively complex and requires the policy to be a non-linear function approximator such as neural networks, while there also is no common ground between recently proposed empirical benchmarks. This is partially caused by multiple confounding factors for RL generalization that can be hard to identify and separate. For instance, an agent can overfit to the MDP dynamics of the training set, such as for control in Mujoco (Pinto et al., 2017; Rajeswaran et al., 2017b). In other cases, an RNN-based policy can overfit to maze-like tasks in exploration (Zhang et al., 2018c), or even exploit determinism and avoid using observations (Bellemare et al., 2012; Machado et al., 2018). Furthermore, various hyperparameters such as the batch-size in SGD (Smith et al., 2018), choice of optimizer (Kingma & Ba, 2014), discount factor γ (Jiang et al., 2015) and regularizations such as entropy (Ahmed et al., 2018) and weight norms (Cobbe et al., 2018) can also affect generalization.

Due to these confounding factors, it can be unclear what parts of the MDP or policy are actually contributing to overfitting or generalization in a principled manner, especially in empirical works with newly proposed benchmarks. In order to isolate these factors, we study one broad factor affecting generalization that is most correlated with themes in SL, specifically *observational overfitting*, where an agent overfits due to properties of the observation which are irrelevant to the latent dynamics of the MDP family. To study this factor, we fix a single underlying MDP's dynamics and generate a distribution of MDP's by only modifying the observational outputs.

Our contributions in this paper are the following:

1. We discuss realistic instances where observational overfitting may occur and its difference from other confounding factors, and design a parametric theoretical framework to induce observational overfitting that can be applied to *any* underlying MDP.
2. We study observational overfitting with linear quadratic regulators (LQR) in a synthetic environment and neural networks such as multi-layer perceptrons (MLPs) and convolutions in classic Gym environments. A primary novel result we demonstrate for all cases is that *implicit regularization* occurs in this setting in RL. We further test the implicit regularization hypothesis on the benchmark CoinRun from using MLPs, even when the underlying MDP dynamics are changing per level.

2 Motivation and Related Works

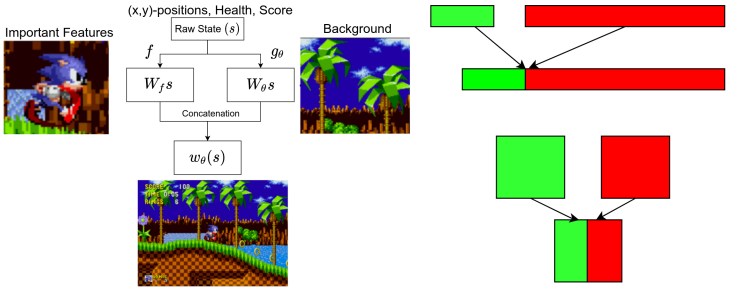
Currently most architectures used in model-free RL are simple (with fewer than one million parameters) compared to the much larger and more complex ImageNet architectures used for classification. This is due to the fact that most RL environments the community studies either have relatively simple and highly structured images (e.g. Atari) compared to real world images, or conveniently do not directly force the agent to observe highly detailed images. For instance in large scale RL such as DOTA2 (OpenAI, 2018) or Starcraft 2 (Vinyals et al., 2017), the agent observations are internal minimaps pertaining to object xy-locations, rather than human-rendered observations.

Figure 1: Example of observational overfitting in Gym Retro (Nichol et al., 2018) for Sonic. Saliency maps highlight (in red) the top-left timer and background objects because they are correlated with progress. The agent could memorize optimal actions for training levels if its observation was *only from the timer*, and "blacking-out" the timer consistently improved generalization performance (see Appendix A.2.3)

Figure 1 highlights the issues surrounding MDP's with rich, textured observations - specifically, the agent can use any features that are correlated with progress, even those which may not generalize across levels. Several artificial benchmarks (Zhang et al., 2018b; Gamrian & Goldberg, 2019) have been proposed before to portray this notion of overfitting, where an agent must deal with a changing background - however, a key difference in our work is that we explicitly require the "background" to be **correlated with the progress** rather than loosely correlated (e.g. through determinism between the background and the game avatar) or not at all. This makes a more explicit connection to *causal inference* (Arjovsky et al., 2019; Heinze-Deml & Meinshausen, 2019; Heinze-Deml et al., 2019) where *spurious correlations* between ungeneralizable features and progress may make training easy, but are detrimental to test performance because they induce false attributions.

Previously, many works interpret the decision-making of an agent through saliency and other network visualizations (Greydanus et al., 2018; Such et al., 2018) on common benchmarks such as Atari. However, our work is motivated by learning theoretic frameworks to capture this phenomena, as there is vast literature on understanding the generalization properties of SL classifiers (Neyshabur et al., 2017; Vu, 2007; Novak et al., 2018; Neyshabur et al., 2018b). For an RL policy with high-dimensional observations, we hypothesize its overfitting can come from more theoretically principled reasons, as opposed to purely good inductive biases on game images.

As an example of what may happen in high dimensional observation space, consider a generic loss function $\ell(x)$ acting on a low dimensional input space $x \in \mathbb{R}^d$, with a projected loss function $\ell_{projected}(y) = \ell(Zy)$ acting on a high dimensional space $y \in \mathbb{R}^h$. If x^* is a minima of $\ell(x)$,



(a) (b)

Figure 2: (a) Visual Analogy of the Observation Function. (b) Our combinations for 1-D (top) and 2-D (bottom) images for synthetic tasks.

$\ell_{projected}$ can posses many more minima as rank of the solution space y^* can be as high as $h - d$ from solving $Zy^* = x^*$ where Z is $d \times h$ and $h \gg d$. Thus a high dimensional observation space with a low dimensional state space can induce multiple solutions, some of which are not generalizable to other functions or MDP's.

2.1 Notation

To formalize the zero-shot framework for RL generalization, let Θ be a distribution over parameters θ that parametrize an MDP family $\{\mathcal{M}_\theta : \theta \in \Theta\}$. Each θ parametrizes $\mathcal{M}_\theta = (\mathcal{S}_\theta, \mathcal{A}_\theta, r_\theta, \mathcal{T}_\theta, S_{0,\theta}, w_\theta)$ which are respectively, state space, action space, reward, transitions, initial state, and observation function. An appropriate train and test set can then be created by randomly sampling $\theta \sim \Theta$ and training or evaluating within $\mathcal{M}_\theta$, and thus expected episodic reward will also be parametrized as $R_\theta(\pi)$. In this work, we focus on modifying the observation function $w_\theta : \mathcal{S} \to \mathcal{W}$, where the agent receives input from the high dimensional observation space $\mathcal{W}$ while keeping the rest of the MDP family parts fixed.

Let $\widehat{\Theta}_{train} = \{\theta_i, \ldots, \theta_n\}$ be a set of n i.i.d. samples from Θ, and suppose we train π to optimize reward against $\{\mathcal{M}_\theta : \theta \sim \widehat{\Theta}_{train}\}$. The objective $J(\pi)$ maximized is the average reward over this empirical sample, $J_{\widehat{\Theta}}(\pi) = \frac{1}{|\widehat{\Theta}_{train}|} \sum_{\theta_i \in \widehat{\Theta}_{train}} R_{\theta_i}(\pi)$. We want to generalize to all $\theta \in \Theta$, which can be expressed as the average episode reward R over the full distribution, $J_\Theta(\pi) = \mathbb{E}_{\theta \sim \Theta}[R_\theta(\pi)]$. Thus it follows to define the generalization gap in RL as $J_{\widehat{\Theta}}(\pi) - J_\Theta(\pi)$.

2.2 Setup

We can model the effects of Figure 1 more generally, not specific to sidescroller games. We assume that there is an underlying *state* s (e.g. xy-locations of objects in a game), whose features may be very well structured, but that this state has been projected to a high dimensional observation space by w_θ. To abstract the notion of generalizable and non-generalizable features, we construct a simple and natural candidate class of functions, where $w_\theta(s) = h(f(s), g_\theta(s))$.

$$w_\theta(s) = h(f(s), g_\theta(s)) \tag{1}$$

In this setup, $f(\cdot)$ is a function invariant for the entire MDP population Θ, while $g_\theta(\cdot)$ is a function dependent on the sampled parameter θ. h is a "combination" function which combines the two outputs of f and g to produce a final observation. While f projects this latent data into salient and important, *invariant* features such as the avatar, monsters, and items, g_θ projects the latent data to unimportant features that do not contribute to extra generalizable information, and can cause overfitting, such as the changing background or textures. A visual representation is shown in Figure 2.2. This is a simplified but still insightful model relevant in more realistic settings. For instance, in settings where g does matter, learning this separation and task-identification (Yu et al., 2017; Peng et al., 2018) could potentially help fast adaptation in meta-learning (Finn et al., 2017). From now on, we denote this setup as the *fg-scheme*.

This setting also leads to more interpretable generalization bounds - Lemma 2 of (Wang et al., 2019) provides a high probability $(1 - \delta)$ bound for the "intrinsic" generalization gap

when m levels are sampled: $gap \leq Rad_m(R_\pi) + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)$, where $Rad_m(R_\pi) = \mathbb{E}_{(\theta_1,\ldots,\theta_m)\sim\Theta^m}\left[\mathbb{E}_{\sigma\in\{-1,+1\}}\left[\sup_\pi \frac{1}{m}\sum_{i=1}^m \sigma_i R_{\theta_i}(\pi)\right]\right]$ is the Rademacher complexity under the MDP, where θ_i are the ζ_i parameters used in the original work, and the transition $\mathcal{T}$ and initialization $\mathcal{I}$ are fixed, therefore omitted, to accommodate our setting. For most RL benchmarks, this is not interpretable due to multiple confounding factors such as the varying level dynamics. However, in our case, because the environment parameters θ are only from g_θ, the Rademacher complexity is directly based on how much the policy "looks at" g_θ. A generalized optimal policy π^* therefore should not be affected by changes in g_θ; i.e. $\nabla_\theta \pi^*(w_\theta(s)) = 0 \ \forall s$ and $R_\theta(\pi^*) = R_{const} \ \forall \theta$, which implies that the environment parameter θ has no effect on the reward; hence $Rad_m(R_\pi) = \mathbb{E}_{\sigma\in\{-1,+1\}}\left[\sup_\pi \frac{1}{m}\sum_{i=1}^m \sigma_i R_{const}\right] = 0$.

2.3 Architecture and Implicit Regularization

Normally in a MDP such as a game, the concatenation operation may be dependent on time (e.g. textures move around in the frame). In the scope of this work, we simplify the concatenation effect and assume $h(\cdot)$ is a static concatenation, but still are able to demonstrate insightful properties. [1]

This setting is naturally attractive to analyzing architectural differences, as it is more closely related in spirit to image classifiers and SL. One particular line of work to explain the effects of certain architectural modifications in SL such as overparametrization and residual connections is *implicit regularization* (Neyshabur, 2017; Neyshabur et al., 2018b), as overparametrization through more layer depth and wider layers has proven to have no ℓ_p-regularization equivalent (Arora et al., 2019), but rather precondition the dynamics during training. Thus, in order to fairly experimentally measure this effect, we always use fixed hyperparameters and only vary based on architecture. [2]

3 Experiments

3.1 WarmUp - LQR

We start with a a principled example in the deterministic classic control setting, by using the linear quadratic regulator (LQR) as a basis for the underlying MDP. We use full gradient descent through the loss, ignoring confounding aspects of RL (exploration, entropy, γ, noise, stochastic gradients, etc.). Furthermore, all minima are *global minima*, and hence asymptotic training performance is always the same. For a given θ, we let $f(s) = W_{constant} \cdot s$, while $g_\theta(s) = W_\theta \cdot s$ where $W_{constant}, W_\theta$ are semi-orthogonal matrices, to prevent information loss relevant to outputting the optimal action, as the state is transformed into observation. We sample W_θ randomly, using a scalar integer θ as the seed for random generation. In terms of dimensions, if s is of shape d_{state}, then f also projects to a shape of d_{state}, while g_θ projects to a much larger shape d_{noise}, implying that the observation to the agent is of dimension $d_{signal} + d_{noise}$. In our experiments, we set as default $(d_{signal}, d_{noise}) = (100, 1000)$.

A key insight is that that a policy in high dimensional policy K acting on observation Ws is equivalent to a low-dimensional policy $K_{state} = KW$ acting on state s. We begin with a theorem which implies that a high dimensional observational space directly contributes to overfitting:

Theorem 3.1 *For LQR's whose observation consists of (d_{signal}, d_{noise})-dimensional vectors constructed with the fg-scheme and fixed number of training levels m, the generalization gap upper bound scales with $\mathcal{O}(\sqrt{d_{noise}})$ with high probability.*

We empirically verify that this bound is tight in Figure 3 and defer the detailed proof to Appendix A.4.3. Denote $\|\cdot\|$, $\|\cdot\|_1$, $\|\cdot\|_F$ as the spectral, ℓ_1, and Frobenius norms respectively of a matrix. Furthermore, the continuity of the cost function (Fazel et al., 2018) states that

[1] We note that this inductive bias on h allows *explicit regularization* to trivially solve this problem, by penalizing a policy's first layer that is used to "view" $g_\theta(s)$ (Appendix A3), hence we only focus on implicit regularizations.

[2] In this work, we only refer to *architectural* implicit regularization techniques, which do not have a explicit regularization equivalent. Some techniques e.g. coordinate descent (Bradley et al., 2011) are equivalent to explicit ℓ_1-regularization.

$C(K_{state}) - C(K'_{state}) \leq \mathcal{O}(\|K_{state} - K'_{state}\|^3)$, which implies that $gap \leq \sqrt{\frac{\mathcal{O}(\|K_{state}\|^3)}{m}}$. By semi-orthogonality $\|W\| \leq 1$, hence $\|KW\| \leq \|K\|$. Since the spectral norm $\|K\|$ of a random matrix scales with its dimension d by $d^{1/3}$, thus the generalization gap also scales with the size of $\sqrt{(d_{noise}^{1/3})^3} = \sqrt{d_{noise}}$.

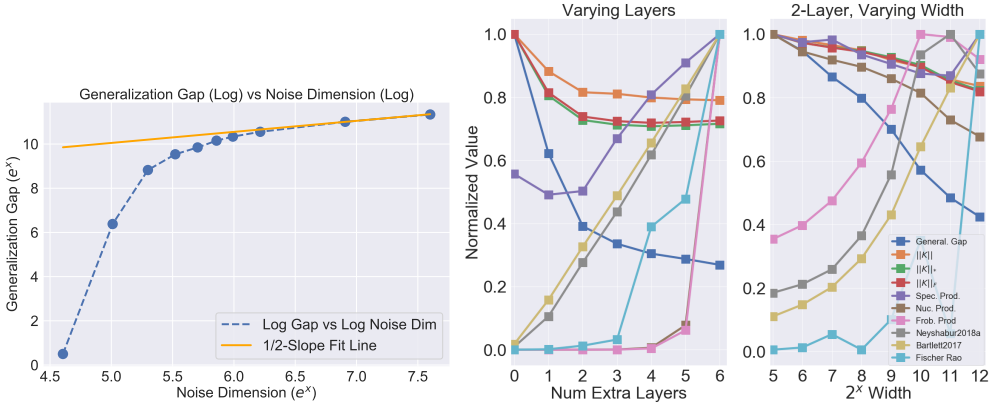
Note the significant difference from SL bounds: $C(K) - C(K') \leq \mathcal{O}(\|K - K'\|^d)$, where $d = 3$ for our LQR case, whereas classification only uses $d = 1$ bounds and are mainly concerned with the Lipschitz constant. We conjecture that this difference is key to why SL bounds do not properly bound the RL generalization gap, especially shown in our overparametrization results shown below. To the best of our knowledge, our problem is not equivalent to any of the well-studied problems in overparametrization.

Experimentally, we added more (100×100) linear layers $K = K_0 K_1, ..., K_j$ and increased widths for a 2-layer case (Figure 3), and observe that both settings reduce the generalization gap, and also reduce the norms (spectral, nuclear, Frobenius) of the final end-to-end policy K, without changing its expressiveness. This suggests that gradient descent under overparametrization implicitly biases the policy towards a "simpler" model in the LQR case. However, how do we quantify this intuition by bounding the generalization performance of the final policy K in terms of norm functions of the layers $K_0, ..., K_j$? For instance, from examining the distribution of singular values on K (Appendix A1), we find that more layers does not bias the policy towards a low rank solution in the nonconvex LQR case, unlike (Arora et al., 2018) which shows this does occur for matrix completion, and in general, convex losses.

Since our setup is similar to SL in that "LQR levels" which may be interpreted as a dataset, we use bounds of the form $\Delta \cdot \Phi$, where Δ is a "macro" product term $\Delta = \prod_{i=0}^{j} \|K_i\| \geq \left\| \prod_{i=0}^{j} K_i \right\|$ derivable from the fact that $\|AB\| \leq \|A\| \|B\|$ in the linear case, and Φ is a *weight-counting* term which deals with the overparametrized case, such as $\Phi = \sum_{i=0}^{j} \frac{\|K_i\|_F^2}{\|K_i\|^2}$ (Neyshabur et al., 2018a) or $\Phi = \left(\sum_{i=0}^{j} \left(\frac{\|K_i\|_1}{\|K_i\|} \right)^{2/3} \right)^3$ (Bartlett et al., 2017). While $gap \leq \sqrt{\mathcal{O}(\|K_{raw}\|^3)} \leq \mathcal{O}(\prod_{i=0}^{j} \|K_i\|^{\frac{3}{2}}) = \Delta^{3/2}$, we may replace any SL perturbation bounds $|f_w(x) - f_{w'}(x)|$ with $C(K) - C(K') \leq \mathcal{O}(\|K - K'\|^3)$, which can grant us expressions similar to Φ, but with different exponents. However, the Φ terms increase too rapidly as shown in Figure 3.

Terms such as Frobenius product (Golowich et al., 2018) and Fischer-Rao (Liang et al., 2019) are effective for the SL depth case, but are both ineffective in the LQR depth case. For width, the only product which is effective is the nuclear norm product.

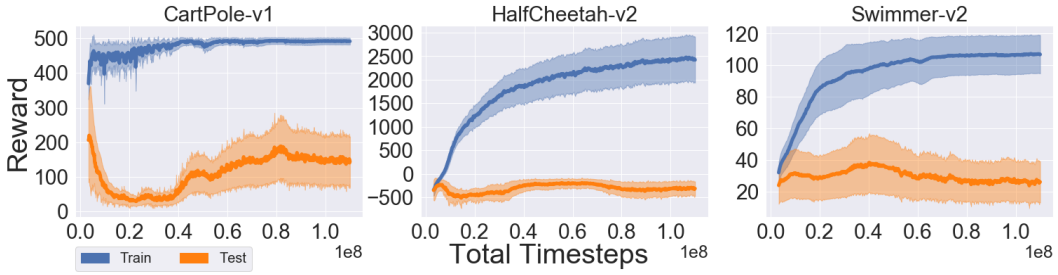
Figure 3: (**Left**) We show that the generalization gap vs noise dimension is tight as the noise dimension increases, showing that this bound is accurate. (**Middle** and **Right**) LQR Generalization Gap vs Number of Intermediate Layers. We plotted different $\Phi = \sum_{i=0}^{j} \frac{\|A\|_*}{\|A\|}$ terms without exponents, as powers of those terms are monotonic transforms since $\frac{\|A\|_*}{\|A\|} \geq 1 \ \forall A$ and $\|A\|_* = \|A\|_F, \|A\|_1$. We see that the naive spectral bound diverges at 2 layers, and the weight-counting sums are too loose.



3.2 PROJECTED GYM ENVIRONMENTS

In Section 3.1, we find that observational overfitting exists and overparametrization potentially helps in the linear setting. We perform the fg-scheme again on nonlinear, common Gym Environments, and use Proximal Policy Gradient (Schulman et al., 2017) for optimization. We observe empirically that the underlying state dynamics has a significant effect on generalization performance as the policy nontrivially increased test performance such as in CartPole-v1 and Swimmer-v2, while it could not for others. This suggests that the Rademacher complexity and the weight-perturbation bound for rewards vary highly for different environments.

Figure 4: Observational overfitting occurring for Gym. Full Plots in A2.



Switching between ReLU and Tanh activations produces different results during overparametrization. For instance, increasing Tanh layers improves generalization on CartPole-v1, and width increase with ReLU helps on Swimmer-v2. Tanh is noted to consistently improve generalization performance. However, stacking Tanh layers comes at a cost of also producing vanishing gradients which can produce subpar training performance, for e.g. HalfCheetah. To allow larger depths, we use ReLU *residual* layers, which also improves generalization and stabilizes training.

Previous work (Zhang et al., 2018c) did not find such an architectural pattern for GridWorld environments, suggesting that this effect may exist primarily for observational overfitting cases. While there have been numerous works which avoid overparametrization on simplifying policies (Rajeswaran et al., 2017a; Mania et al., 2018) or compactifying networks (Choromanski et al., 2018; Gaier & Ha, 2019), we instead find that there are generalization benefits to overparametrization even in the nonlinear control case.

Figure 5: Effects of Depth.

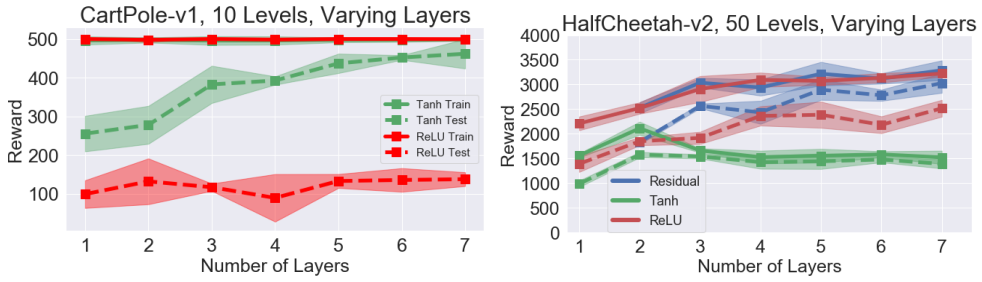
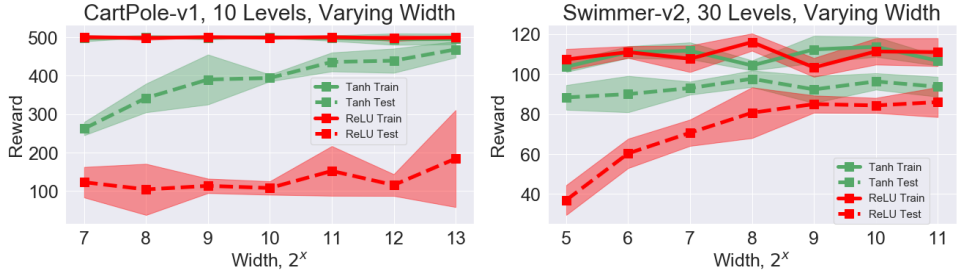


Figure 6: Effects of Width.



3.3 DECONVOLUTIONAL PROJECTIONS

From the above results with MLP's, one may wonder if similar results may carry to convolutional networks, as they are mainly used for vision-based RL tasks. As a ground truth reference for our

experiment, we the canonical networks proven to generalize well in the dataset CoinRun, which are from worst to best, NatureCNN Mnih et al. (2013), IMPALA Espeholt et al. (2018), and IMPALA-LARGE (IMPALA with more residual blocks and higher convolution depths), which have respective parameter numbers (600K, 622K, 823K).

We setup a similar fg-scheme appropriate for the inductive bias of convolutions, by projecting the state latent to a fixed length, reshaping it into a square, and replacing f and g_θ both with the same orthogonally-initialized *deconvolution* architecture to each produce a 84×84 image (but g_θ's network weights are still generated by $\theta_1, ..., \theta_m$ similar to before). We combine the two outputs by using one half of the "image" from f, and one half from g_θ, as shown back in Figure 2.2.

Figure 7: (**Top**) Performance of architectures in the synthetic Gym-Deconv dataset. To cleanly depict test performance, training curves are replaced with horizontal (max env. reward) and vertical black lines (avg. timestep when all networks reach max reward). (**Bottom**) We only show the observation from $g_\theta(s)$, which tests memorization capacity on Swimmer-v2.

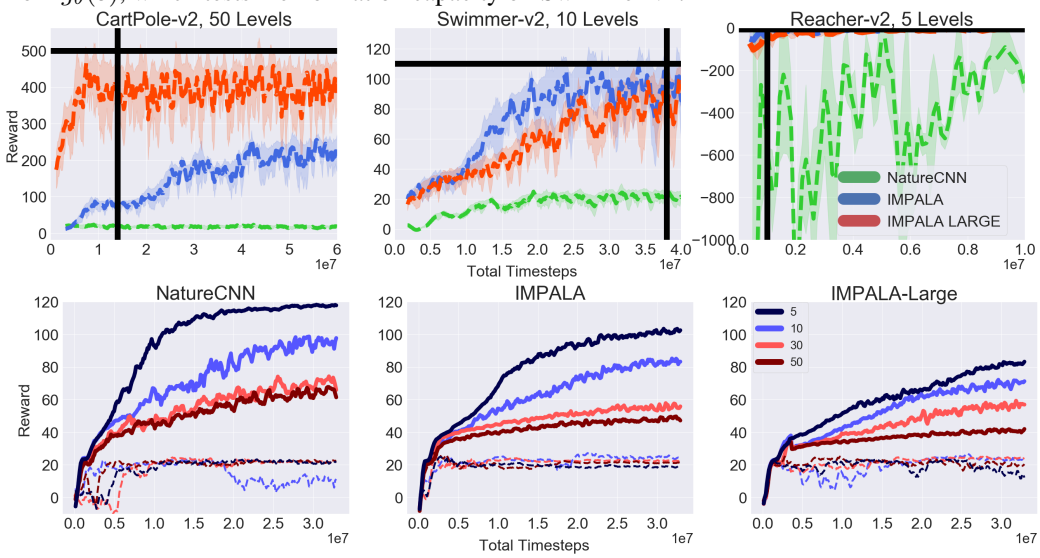


Figure 7 shows that the same ranking between the three architectures exists as well on the Gym-Deconv dataset. This consistency in results suggests that the observational overfitting framework is correlated with the generalization issues found in CoinRun. Furthermore, this also suggests that the RL generalization quality of a convolutional architecture is not limited to *real world data*, as our test purely uses numeric observations - i.e. the observation is simply unintelligible static if represented in human format.

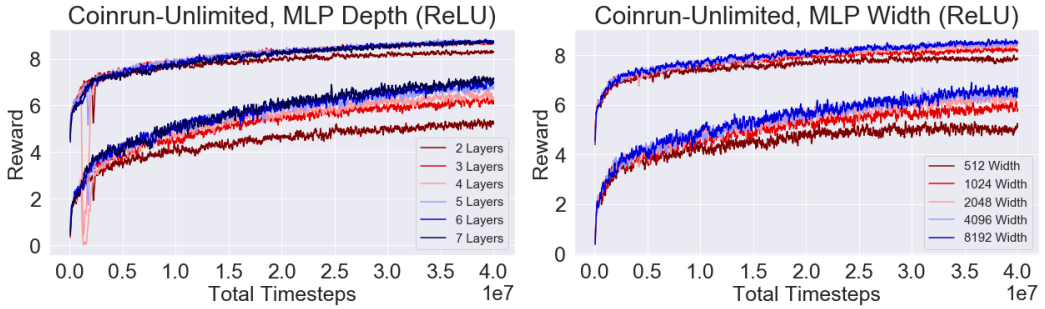
We also perform a memorization test by only showing g_θ's output to the policy. This makes the dataset impossible to generalize to, as the policy network cannot invert every single observation function $g_{\theta_1}(\cdot), g_{\theta_2}(\cdot), ...$ simultaneously. (Zhang et al., 2018c) also constructs a memorization test for mazes and grid-worlds, and showed that more parameters increased the memorization ability of the policy. We show in the bottom figure that this is perhaps not a complete picture when implicit regularization becomes involved.

Using the underlying MDP as a Swimmer-v2 environment, we see that NatureCNN, IMPALA, IMPALA-LARGE have reduced memorization performances. IMPALA-LARGE, which has more depth parameters and more residual layers (and thus technically has more capacity), memorizes *less* than IMPALA due its inherent inductive bias. Another memorization test where an LQR is used as underlying MDP is shown in Appendix A.1.2, with similar results. We hypothesize that these extra residual blocks may be implicitly regularizing the network. This is corroborated by the fact that residual layers are also explained as an *implicit regularization* technique (Neyshabur, 2017) for SL.

4 OVERPARAMETRIZATION IN COINRUN

We test our hypothesis from the above to the CoinRun benchmark, using unlimited levels for training. For MLP networks, we downsized CoinRun from native 64×64 to 32×32, and flattened the $32 \times 32 \times 3$ image for input to an MLP. Two significant differences from previous cases are that 1. inherent dynamics are changing per level in CoinRun, and 2. the relevant and irrelevant CoinRun features change locations across the 1-D input vector. Regardless, we show that overparametrization can still improve generalization in this more realistic RL benchmark, much akin to Neyshabur et al. (2018b) which showed that overparametrization for MLP's improved generalization on $32 \times 32 \times 3$ CIFAR-10.

Figure 8: Overparametrization improves generalization for CoinRun.



While we also extend the case of large-parameter convolutional networks using ImageNet networks in Appendix A.2.1, an important question is how to predict the generalization gap only from the training phase. A particular set of metrics, popular in the SL community are *margin distributions* (Jiang et al., 2018; Bartlett et al., 2017), as they deal with the case for softmax outputs which do not explicitly penalize the weight norm of a network, by normalizing the "confidence" margin of the logit outputs. While using margins on state-action pairs (from an on-policy replay buffer) is not technically rigorous, one may be curious to see if they have predictive power, especially as MLP's are relatively simple to norm-bound. We plotted these margin distributions in Appendix A.2.2, but found that the weight norm bounds used in SL are simply too dominant for this RL case. This, with the bound results found earlier for the LQR case, suggests that current norm bounds are simply too loose for the RL case even though we have shown overparametrization helps generalization in RL, and hopefully this motivates more of the study of such theory.

5 CONCLUSION

We have identified and isolated a key component of overfitting in RL as the particular case of "observational overfitting", which is particularly attractive for studying architectural implicit regularizations. We have analyzed this setting extensively, by examining 3 main components:

1. The analytical case of LQR and linear policies under exact gradient descent, which lays the foundation for understanding theoretical properties of networks in RL generalization.
2. The empirical but principled Projected-Gym case for both MLP and convolutional networks which demonstrates the effects of neural network policies under nonlinear environments.
3. The large scale case for CoinRun, which can be interpreted as a case where relevant features are moving across the input, where empirically, MLP overparametrization also improves generalization.

We noted that current network policy bounds using ideas from SL are unable to explain overparametrization effects in RL, which is an important further direction. In some sense, this area of RL generalization is an extension of static SL classification from adding extra RL components. For instance, adding a nontrivial "combination function" between f and g_θ that is dependent on time (to simulate how object pixels move in a real game) is both an RL generalization issue and potentially video classification issue, and extending results to the memory-based RNN case will also be highly beneficial. Extending the analysis to off-policy methods such as Q-learning and also ES-based methods is also important. We believe that this work provides an important initial step towards solving these future problems.

REFERENCES

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. *CoRR*, abs/1811.11214, 2018.

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019. URL http://arxiv.org/abs/1907.02893.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 244–253, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *CoRR*, abs/1905.13655, 2019. URL http://arxiv.org/abs/1905.13655.

Kavosh Asadi, Dipendra Misra, and Michael L. Littman. Lipschitz continuity in model-based reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 264–273, 2018.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6240–6249. Curran Associates, Inc., 2017.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *CoRR*, abs/1207.4708, 2012.

Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for l1-regularized loss minimization. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 321–328, 2011. URL https://icml.cc/2011/papers/231_icmlpaper.pdf.

Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard E. Turner, and Adrian Weller. Structured evolution with compact architectures for scalable policy optimization. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 969–977, 2018. URL http://proceedings.mlr.press/v80/choromanski18a.html.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. *CoRR*, abs/1812.02341, 2018.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 1406–1415, 2018.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 1466–1475, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1126–1135, 2017. URL http://proceedings.mlr.press/v70/finn17a.html.

Adam Gaier and David Ha. Weight agnostic neural networks. *CoRR*, abs/1906.04358, 2019. URL http://arxiv.org/abs/1906.04358.

Shani Gamrian and Yoav Goldberg. Transfer learning for related reinforcement learning tasks via image-to-image translation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 2063–2072, 2019. URL http://proceedings.mlr.press/v97/gamrian19a.html.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018.*, pp. 297–299, 2018. URL http://proceedings.mlr.press/v75/golowich18a.html.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 1787–1796, 2018.

Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *CoRR*, abs/1710.11469, 2019. URL https://arxiv.org/abs/1710.11469.

Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. *CoRR*, abs/1706.08576, 2019. URL https://arxiv.org/abs/1706.08576.

Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pp. 1181–1189, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-3413-6.

Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *CoRR*, abs/1810.00113, 2018. URL http://arxiv.org/abs/1810.00113.

Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. Procedural level generation improves generality of deep reinforcement learning. *CoRR*, abs/1806.10729, 2018. URL http://arxiv.org/abs/1806.10729.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Tengyuan Liang, Tomaso A. Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pp. 888–896, 2019. URL http://proceedings.mlr.press/v89/liang19a.html.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *J. Artif. Intell. Res.*, 61:523–562, 2018. doi: 10.1613/jair.5699.

Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning. *CoRR*, abs/1803.07055, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

Behnam Neyshabur. Implicit regularization in deep learning. *CoRR*, abs/1709.01953, 2017. URL http://arxiv.org/abs/1709.01953.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 5949–5958, 2017.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018a. URL https://openreview.net/forum?id=Skz_WfbCZ.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *CoRR*, abs/1805.12076, 2018b. URL http://arxiv.org/abs/1805.12076.

Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. Gotta learn fast: A new benchmark for generalization in RL. *CoRR*, abs/1804.03720, 2018.

Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *CoRR*, abs/1802.08760, 2018.

OpenAI. Openai five. 2018. URL https://openai.com/blog/openai-five/.

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *CoRR*, abs/1810.12282, 2018. URL http://arxiv.org/abs/1810.12282.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pp. 1–8, 2018. doi: 10.1109/ICRA.2018.8460528. URL https://doi.org/10.1109/ICRA.2018.8460528.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 2817–2826, 2017.

Aravind Rajeswaran, Kendall Lowrey, Emanuel Todorov, and Sham M. Kakade. Towards generalization and simplicity in continuous control. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 6553–6564, 2017a.

Aravind Rajeswaran, Kendall Lowrey, Emanuel Todorov, and Sham M. Kakade. Towards generalization and simplicity in continuous control. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 6553–6564, 2017b.

Adam Santoro, Ryan Faulkner, David Raposo, Jack W. Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy P. Lillicrap. Relational recurrent neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pp. 7310–7321, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=B1Yy1BxCZ.

Felipe Petroski Such, Vashisht Madhavan, Rosanne Liu, Rui Wang, Pablo Samuel Castro, Yulun Li, Ludwig Schubert, Marc G. Bellemare, Jeff Clune, and Joel Lehman. An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents. *CoRR*, abs/1812.07069, 2018.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy P. Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. Starcraft II: A new challenge for reinforcement learning. *CoRR*, abs/1708.04782, 2017. URL http://arxiv.org/abs/1708.04782.

Van H. Vu. Spectral norm of random matrices. *Combinatorica*, 27(6):721–736, 2007. doi: 10.1007/s00493-007-2190-z.

Huan Wang, Stephan Zheng, Caiming Xiong, and Richard Socher. On the generalization gap in reparameterizable reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 6648–6658, 2019. URL http://proceedings.mlr.press/v97/wang19o.html.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017. doi: 10.15607/RSS.2017.XIII.048. URL http://www.roboticsproceedings.org/rss13/p48.html.

Amy Zhang, Nicolas Ballas, and Joelle Pineau. A dissection of overfitting and generalization in continuous reinforcement learning. *CoRR*, abs/1806.07937, 2018a.

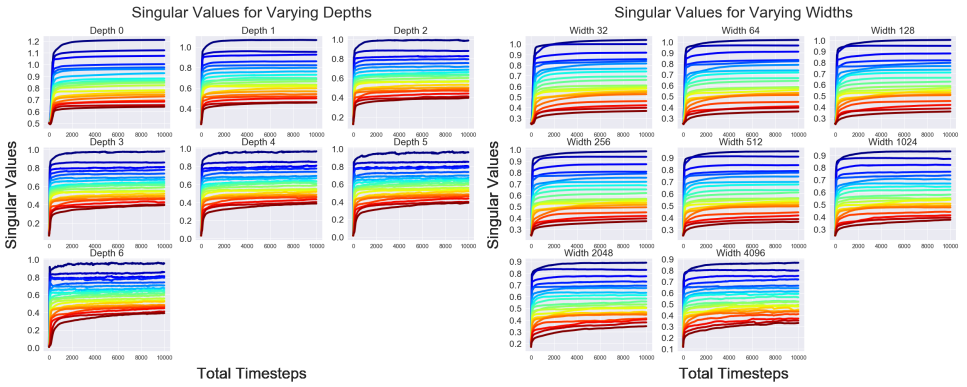
Amy Zhang, Yuxin Wu, and Joelle Pineau. Natural environment benchmarks for reinforcement learning. *CoRR*, abs/1811.06032, 2018b.

Chiyuan Zhang, Oriol Vinyals, Rémi Munos, and Samy Bengio. A study on overfitting in deep reinforcement learning. *CoRR*, abs/1804.06893, 2018c. URL http://arxiv.org/abs/1804.06893.

A.1 FULL PLOTS FOR LQR, FG-GYM-MLP, FG-GYM-DECONV

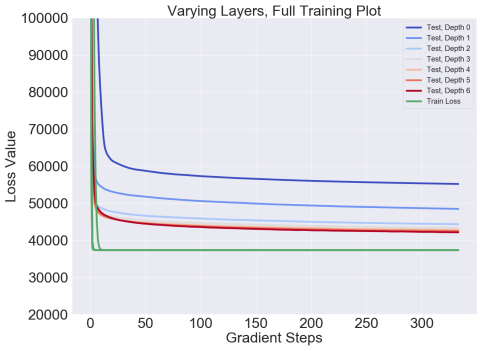
A.1.1 LQR

Figure A1: (a,b): Singular Values for varying depths and widths. (c,d): Train and Test Loss for varying widths and depths. (e): Train and Test Loss for varying Noise Dimensions.

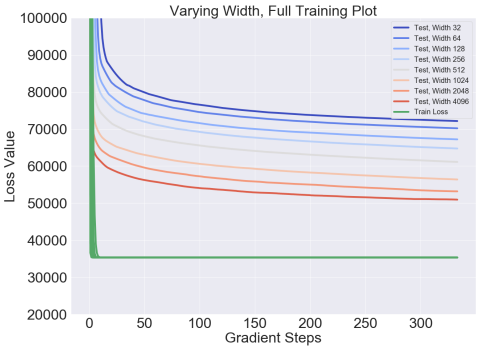


(a)

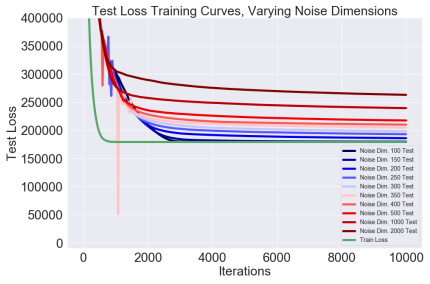
(b)



(c)



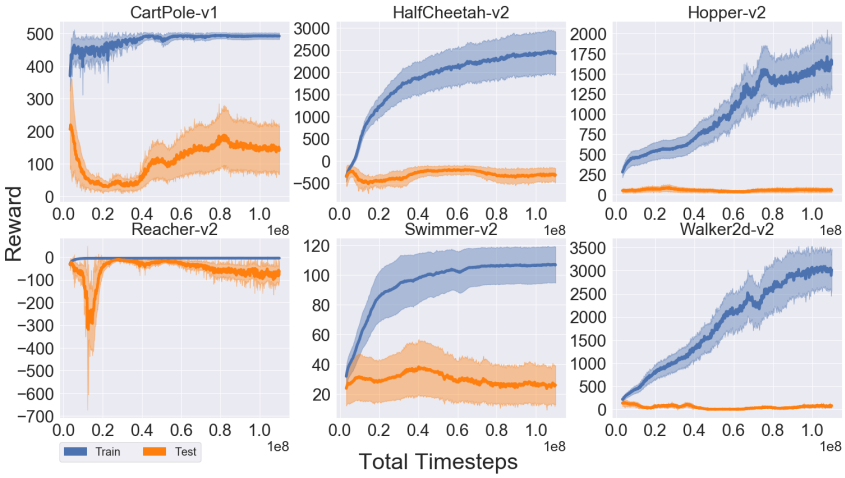
(d)



(e)

A.1.2 EXTENDED fg-SCHEME RESULTS

Figure A2: Each Mujoco task is given 10 training levels (randomly sampling g_θ parameters). We used a 2-layer Tanh policy, with 128 hidden units each. Dimensions of outputs of (f, g) were $(30, 100)$ respectively.



(a)

We further verify that *explicit regularization* (norm based penalties) also reduces generalization gaps. However, explicit regularization may be explained due to the bias of the synthetic tasks, since the first layer's matrix may be regularized to only "view" the output of f, especially as regularizing the first layer's weights substantially improves generalization.

Figure A3: Explicit Regularization on layer norms.

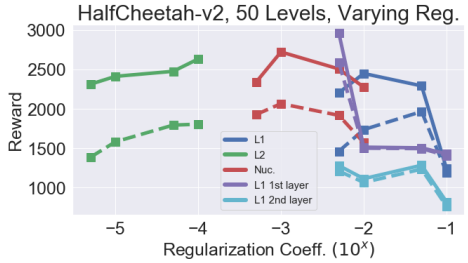
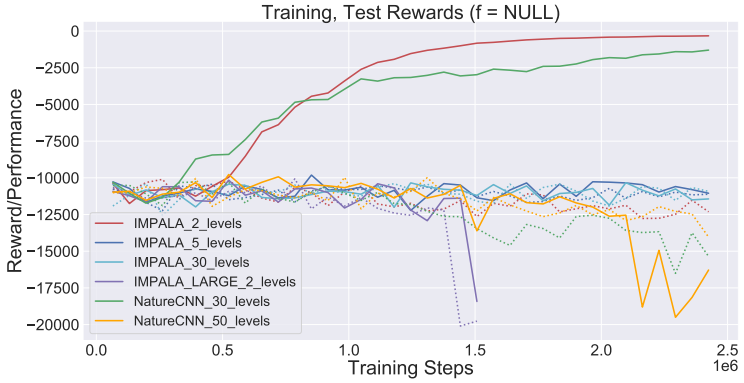


Figure A4: Another deconvolution memorization test, using an LQR as the underlying MDP. While fg-Gym-Deconv shows that memorization performance is dampened, this test shows that there can exist specific hard limits to memorization. Specifically, NatureCNN can memorize 30 levels, but not 50; IMPALA can memorize 2 levels but not 5; IMPALA-LARGE cannot memorize 2 levels at all.



A.2 EXTENDED LARGE RL RESULTS

A.2.1 LARGE IMAGENET MODELS FOR COINRUN

We experimentally verify in table (1) that large ImageNet models perform very differently in RL than during SL. We note that default network with the highest test reward was IMPALA-LARGE-BN (IMPALA-LARGE, with Batchnorm) at ≈ 5.5 test score.

In order to verify that this is inherently a *feature learning* problem rather than a *combinatorial problem* involving objects, such as in (Santoro et al., 2018), we show that state-of-the-art attention mechanisms for RL such as Relational Memory Core (RMC) using pure attention on raw 32×32 pixels does not perform well here, showing that a large portion of generalization and transfer must be based on correct convolutional setups.

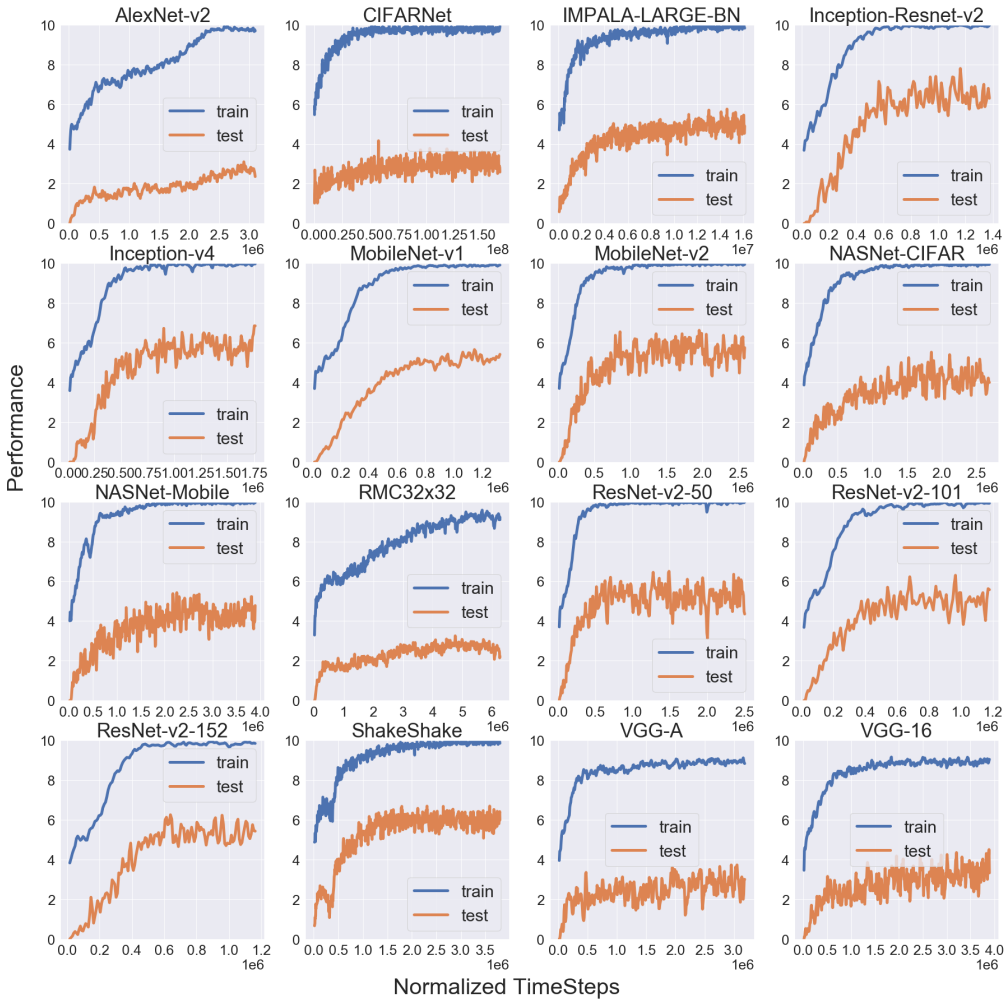
Architecture	Coinrun-100 (Train, Test)
AlexNet-v2	(10.0, 3.0)
CifarNet	(10.0, 3.0)
IMPALA-LARGE-BN	(10.0, 5.5)
Inception-ResNet-v2	(10.0, 6.5)
Inception-v4	(10.0, 6.0)
MobileNet-v1	(10.0, 5.5)
MobileNet-v2	(10.0, 5.5)
NASNet-CIFAR	(10.0, 4.0)
NASNet-Mobile	(10.0, 4.5)
ResNet-v2-50	(10.0, 5.5)
ResNet-v2-101	(10.0, 5.0)
ResNet-v2-152	(10.0, 5.5)
RMC32x32	(9.0, 2.5)
ShakeShake	(10.0, 6.0)
VGG-A	(9.0, 3.0)
VGG-16	(9.0, 3.0)

Table 1: Raw Network Performance (rounded to nearest 0.5) on CoinRun, 100 levels. Images scaled to default image sizes (32×32 or 224×224) depending on network input requirement. See Appendix A5 for training curves.

We provide the training/testing curves for the ImageNet/large convolutional models used. Note the following:

1. RMC32x32 projects the native image from CoinRun from 64×64 to 32×32, and uses all pixels as components for attention, after adding the coordinate embedding found in (Santoro et al., 2018). Optimal parameters were (mem_slots = 4, head_size = 32, num_heads = 4, num_blocks = 2, gate_style = 'memory').
2. Auxiliary Loss in ShakeShake was not used during training, only the pure network.
3. VGG-A is a similar but slightly smaller version of VGG-16.

Figure A5: Large Architecture Training/Testing Curves (Smoothed).
Train and Test Performances of Special Architectures on CoinRun-100



A.2.2 Do State-Action Margin Distributions Predict Generalization in RL?

A conceptual difference between CoinRun and our other tasks is due to the discrete action space of CoinRun. We verify in Figure A6, that indeed, simply measuring the raw norms of the policy network is a poor way to predict generalization, as it generally increases even as training begins to plateau. This is inherently because the softmax on the logit output does not penalize arbitrarily high logit values, and hence proper normalization is needed.

We are curious in measuring the *margin distribution* of action logits, as this has been used extensively to empirically predict the generalization properties of classifiers (). For a policy, the the margin distribution will be defined as $(s, a) \rightarrow \frac{F_\pi(s)_a - \max_{i \neq y} F_\pi(x)_i}{\mathcal{R}_\pi \|S\|_2 / n}$, where $F_\pi(s)_a$ is the logit value of action a given input s, before the softmax, and S is the matrix of states in the replay buffer, and $\mathcal{R}_\pi$ is the norm-based Lipschitz bound on the policy network logits. We used the Spectral, Sharpness and Bartlett bounds, for $\mathcal{R}_\pi$, and we replace the classical supervised learning pair $(x, y) = (s, a)$ with the state action pairs found on-policy.

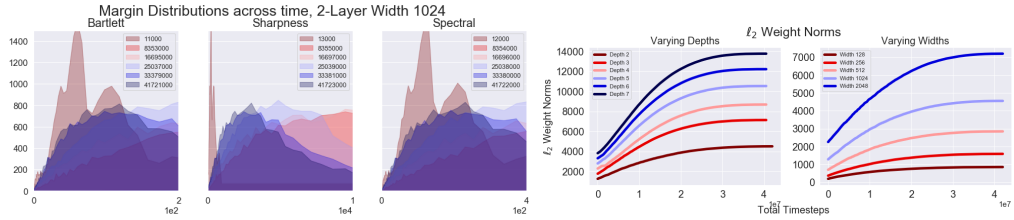
We used the following metrics $\mathcal{R}_\pi$ (after removing irrelevant constants)

1. Bartlett Bound: $\left(\prod_{i=1}^{d} \|W_i\|\right) \left(\sum_{i=1}^{d} \frac{\|W_i\|_1^{2/3}}{\|W_i\|^{2/3}}\right)^{3/2}$
2. Sharpness Bound: $\sqrt{\frac{\sum_{i=1}^{d} \left\|W_i - W_i^0\right\|_F^2 + \ln(2m/\delta)}{m}}$

3. Spectral Bound: $\sqrt{\frac{\ln(d)\prod_{i=1}^{d}\|W_i\|_2^2\sum_{j=1}^{d}\frac{\|W_j-W_j^0\|_F^2}{\|W_j\|_2^2}+\ln(\frac{6m}{\delta})}{m}}$

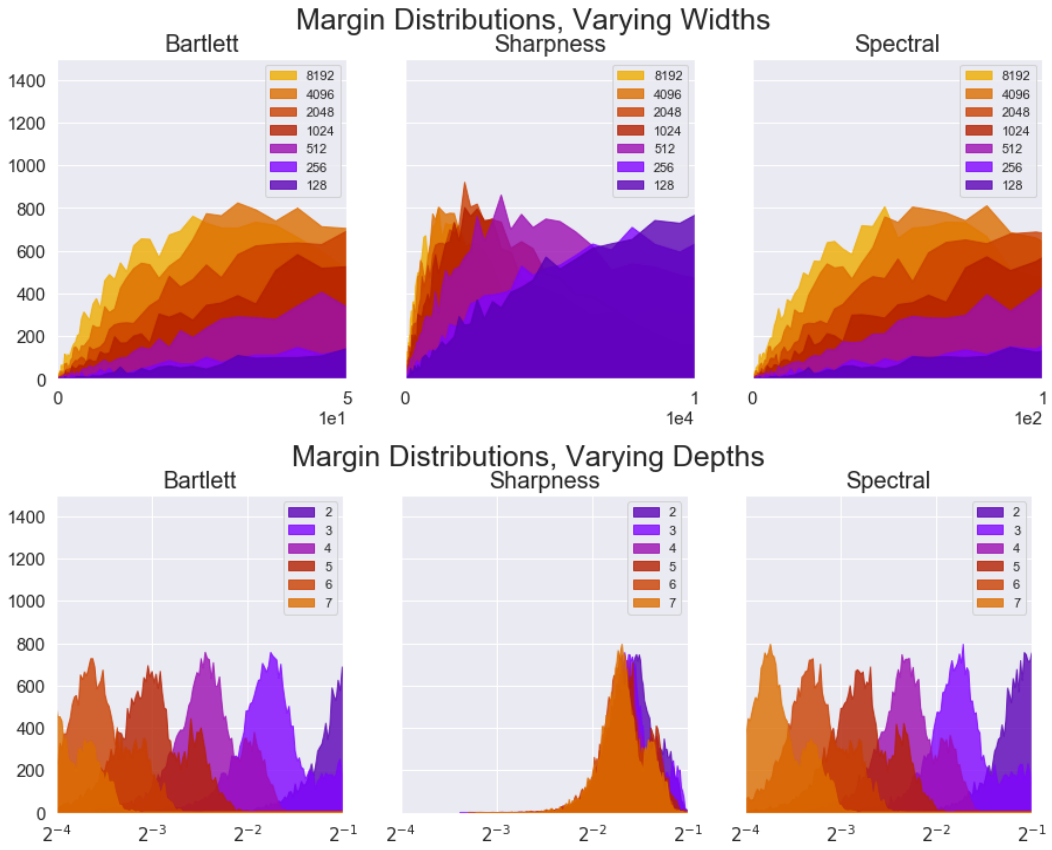
Unlike the other metrics mentioned, the margin distribution converges to a fixed distribution even long after training has plateaued. However, unlike SL, the margin distribution is conceptually not fully correlated with RL generalization on the total reward, as a policy overconfident in some state-action pairs does not imply bad testing performance. This correlation is stronger if there are Lipschitz assumptions on state-action transitions, as noted in Wang et al. (2019). For empirical datasets such as CoinRun, a metric-distance between transitioned states is ill-defined however. Nevertheless, the distribution over the on-policy replay buffer at each policy gradient iteration is a rough measure of overall confidence.

Figure A6: Margin Distributions at the end of training.



We note that there are two forms of modifications, *network dependent* (explicit modifications to the policy - norm regularization, dropout, etc.) and *data dependent* (modifications only to the data in the replay buffer - action stochasticity, data augmentation, etc.). Ultimately however, we find that current norm bounds R_π become too dominant in the fraction, leading to the monotonic decreases in the means of the distributions as we increase parametrization.

Figure A7: Margin Distributions at the end of training.



A.2.3 GYM-RETRO (SONIC)

In the Gym-Retro benchmark (Sonic), the agent is given 47 training levels with rewards corresponding to increases in horizontal location. The policy is trained until 5k reward. At test time, 11 unseen levels are partitioned into starting positions, and the rewards are measured and averaged.

We briefly mention that the agent strongly overfits to the scoreboard (i.e. an artifact correlated with progress in the level), which may be interpreted as part of the output of $g_\theta(\cdot)$. In fact, the agent is still able to train to 5k reward from purely observing the timer as the observation. By blacking out this scoreboard with a black rectangle, we see an increase in test performance.

Settings	IMPALA	NatureCNN
Blackout	1250 ± 40	1141 ± 40
NoBlackout	1130 ± 40	1052 ± 40

Table 2: IMPALA vs NatureCNN test rewards, with and without Blackout.

A.3 HYPERPARAMETERS AND EXACT SETUPS

A.3.1 EXACT INFINITE LQR

For infinite horizon case, see (Fazel et al., 2018) for the the full solution and notations. Using the same notation (A, B, Q, R), denote $C(K) = \sum_{x_0 \sim \mathcal{D}} x_0^T P_K x_0$ as the cost and $u_t = -Kx_t$ as the policy, where P_K satisifies the infinite Algebraic-Ricatti equation:

$$P_K = Q + K^T R K + (A - BK)^T P_K (A - BK) \tag{2}$$

We may calculate the precise LQR cost by vectorizing (i.e. flattening) both sides' matrices and using the Kroncker product $\otimes$, which leads to a linear regression problem on P_K, which has a precise solution, implementable in TensorFlow:

$$\text{vec}(P_K) = \text{vec}(Q) + \text{vec}(K^T R K) + \left[(A - BK)^T \otimes (A - BK)^T)\right] \text{vec}(P_K) \tag{3}$$

$$\left[I_{n^2} - (A - BK)^T \otimes (A - BK)^T\right] \text{vec}(P_K) = \text{vec}(Q) + \text{vec}(K^T R K) \tag{4}$$

Parameter	Generation
A	Uniform Random from set of orthogonal matrices on $n \times n$, scaled 0.99
B	I_n
Q	I_n
R	I_n
n	10

Table 3: Hyperparameters for LQR

A.3.2 PROJECTION METHOD

The basis for producing f, g_θ outputs is due to using batch matrix multiplication operations, or "BMV", where the same network architecture uses different network weights for each batch dimension, and thus each entry in a batchsize of B will be processed by different network weights. This is to simulate the effect of g_{θ_i} - The numeric ID i of the environment is used as an index to collect a specficic set of network weights θ_i from a global memory of network weights (e.g. using `tensorflow.gather`). We did not use nonlinear activations for the BMV architectures, as they did not change the outcome of the results.

Architecture	Setup
BMV-Deconv	(filtersize = 2, stride = 1, outchannel = 8, padding = "VALID") (filtersize = 4, stride = 2, outchannel = 4, padding = "VALID") (filtersize = 8, stride = 2, outchannel = 4, padding = "VALID") (filtersize = 8, stride = 3, outchannel = 3, padding = "VALID")
BMV-Dense	f : Dense 30, g : Dense 100

A.3.3 IMAGENET MODELS

For the networks used in the supervised learning tasks, we direct the reader to the following repository: `https://github.com/tensorflow/models/blob/master/research/slim/nets/nets_factory.py`. We also used the RMC: `deepmind/sonnet/blob/master/sonnet/python/modules/relational_memory.py`

A.3.4 PPO PARAMETERS

For the projected gym tasks, we used for PPO2 Hyperparameters:

PPO2 Hyperparameters	Values
nsteps	2048
nenvs	16
nminibatches	64
λ	0.95
γ	0.99
noptepochs	10
entropy	0.0
learning rate	$3 \cdot 10^{-4}$
vf coeffiicent	0.5
max-grad-norm	0.5
total time steps	Varying

See (Cobbe et al., 2018) for the default parameters used for CoinRun. We only varied nminibatches in order to fit memory onto GPU. We also did not use RNN additions, in order to measure performance only from the feedforward network - the framestacking/temporal aspect is replaced by the option to present the agent velocity in the image.

A.4 THEORETICAL RESULTS (LQR)

See (Fazel et al., 2018) for more extensive LQR notation and statements that we will use. Below proofs of certain overfitting properties in the LQR case, which give more rigorous bounds.

A.4.1 NOTATION AND SETTING

Let $\|\cdot\|$ be the spectral norm of a matrix (i.e. largest singular value). Suppose $C(K)$ was the infinite horizon cost for an (A, B, Q, R)-LQR where action $a_t = -K \cdot x_t$, x_t is the state at time t, state transition is $x_{t+1} = A \cdot x_t + B \cdot a_t$, and timestep cost is $x_t^T Q x_t + a_t^T R a_t$.

$C(K)$ for an infinite horizon LQR, while known to be non-convex, still possess the property that when $\nabla C(K^*) = 0$, K is a global minimizer, or the problem statement is rank deficient. By varying the observation projections, θ generates a population of $C_\theta(K)$ cost functions with all of the population having the same minimizer K^*.

A.4.1.1 SMOOTHNESS BOUNDS

As described in Lemma 16 of (Fazel et al., 2018), we define

$$T_K(X) = \sum_{t=0}^{\infty} (A - BK)^t X [(A - BK)^T]^t \tag{5}$$

and $\|T_K\| = \sup_X \frac{T_K(X)}{\|X\|}$ over all non-zero symmetric matrices X.

Lemma 27 of (Fazel et al., 2018) provides a bound on the difference $C(K') - C(K)$ for two different policies K, K' when LQR parameters A, B, Q, R are fixed. During the derivation, it states that when $\|K - K'\| \leq \min\left(\frac{\sigma_{min}(Q)\mu}{4C(K)\|B\|(\|A-BK\|+1)}, \|K\|\right)$, then

$$C(K') - C(K) \leq \mathbb{E}\left\|x_0\right\|^2 \left\|P_{K'} - P_K\right\| \tag{6}$$

where $P_K = \left\|T_K(Q + K^T R K)\right\|$, and $\|P_{K'} - P_K\|$ is bounded by the sum of two terms, which are $2\left\|T_K\right\|\left(3\left\|K\right\|\left\|R\right\|\left\|K' - K\right\|\right)$ and $2\left\|T_K\right\|^2 2\left\|B\right\|\left(\left\|A - BK\right\| + 1\right)\left\|K - K'\right\|\left\|K\right\|^2\left\|R\right\|$.

Thus we have the bound

$$\begin{aligned} C(K') - C(K) \leq 2\left\|T_K\right\|\left(2\left\|K\right\|\left\|R\right\|\left\|K' - K\right\| + \left\|R\right\|\left\|K' - K\right\|^2\right) + \\ 2\left\|T_K\right\|^2 2\left\|B\right\|\left(\left\|A - BK\right\| + 1\right)\left\|K - K'\right\|\left\|K\right\|^2\left\|R\right\| \end{aligned} \tag{7}$$

Lemma 17 also states that:

$$\left\|T_K\right\| \leq \frac{C(K)}{\mu \sigma_{min}(Q)} \tag{8}$$

where

$$\mu = \sigma_{min}(\mathbb{E}_{x_0 \sim D}[x_0 x_0^T]) \tag{9}$$

Assuming that in our problem setup, x_0, Q, R, A, B were fixed, this means many of the parameters in the bounds are constant, and thus we conclude:

$$C(K') - C(K) \leq \mathcal{O}\left(C(K)^2\left[\left\|K\right\|^2\left\|K - K'\right\|\left(\left\|A - BK\right\| + \left\|B\right\| + 1\right) + \left\|K\right\|\left\|K - K'\right\|^2\right]\right) \tag{10}$$

Since $\|A - BK\| \leq 1$ or else $T_K(X)$ is infinite and in this scheme $\mathcal{O}(\|K' - K\|) = \mathcal{O}(\|K\|)$, we thus finally collect the terms to get the bound we will use in the next sections:

$$C(K') - C(K) \leq \mathcal{O}\left(C(K)^2\left\|K\right\|^3\right) = \mathcal{O}(C(K)^2\left\|K' - K\right\|^3) \tag{11}$$

A.4.2 Observational Projections

Let $C(\cdot)$ be the cost function $\mathcal{S} \to \mathbb{R}$ for a policy acting on the state space, and without loss of generality, normalize the constants in (11) so that $C(K') - C(K) \leq \left\|K' - K\right\|^3$. In the observational projection case, we note that an observation of Ws with policy K is exactly the same as the case when the observation is s and the policy is KW. In our experiments, a semi-orthogonal W is sampled from combining two samples from f and g and normalizing. The experiments allowed W_{obs} to be a matrix of size (d_{obs}, d_{state}), and K size (d_{action}, d_{obs}). Using (11), we see that $C(KW_{obs}^{(1)}) - C(KW_{obs}^{(2)}) \leq \left\|KW_1 - KW_2\right\|^3 \leq \left\|K\right\|^3\left\|W_1 - W_2\right\|^3$.

The dominant term in this expression is $\|K\|$. We can examine how much it scales as a function of the dimension d_{obs}, since for any random K since for any K, we can form an LQR with K as optimal policy, which means random matrix theory can provide insights into the distribution of $\|KW\|$. It is established (Vu, 2007) that for random matrices of such dimensions $d_{obs} \gg d_{action}$, $\|K\|$ scales with the dimension, with growth bound of $\mathcal{O}(d_{obs}^{1/3})$.

A.4.3 Generalization Bounds for LQR linear case

We can think of the semi-orthogonal samples $W_1, W_2, ...$ as the "randomly sampled datapoints" from a distribution $\mathcal{D}_{obs}$ analogous to supervised learning. If we fix A, B, Q, R, then we may write our cost function in LQR as $C_i(K) = C(KW)$. Note that $C(\cdot) \in [0, M]$ for some maximal value M based on A, B, Q, R, x_0.

Then, for a fixed K, and drawing infinite samples of W_i, we define the following:

$$C_{\mathcal{D}_{obs}}(K) = \mathbb{E}_{W \sim \mathcal{D}_{obs}}\left[C(KW)\right] \tag{12}$$

Optimizing this infinite sample case is optimizing the "true cost function". However, if we have finite samples $S_m = \{W_1, ..., W_m\}$, we can define the average sample cost as

$$\widehat{C}_{S_m}(K) = \frac{1}{|S|}\sum_{W_i \in S} C(KW_i) = \frac{1}{|S|}\sum_i C_i(K) \tag{13}$$

We apply the standard proof of generalization gap: From definition of supremum, for a fixed $K \in \mathcal{K}$ denote

$$gap = C_{\mathcal{D}_{obs}}(K) - \widehat{C}_{S_m}(K) \leq \sup_{K_{max} \in \mathcal{K}} \left(C_{\mathcal{D}_{obs}}(K_{max}) - \widehat{C}_{S_m}(K_{max}) \right) \tag{14}$$

Denote random variable $\psi(S) = \sup_{K_{max} \in \mathcal{K}} \left(C_{\mathcal{D}_{obs}}(K_{max}) - \widehat{C}_{S_m}(K_{max}) \right)$. We need to understand how much ψ changes as a result of changing the samples W_i.

A common approach to forming generalization gap bounds comes from the McDiarmid inequality:

Suppose K is fixed. If $\psi(S)$ satisfies:

$$\sup_{W_1,...,W_m} |\psi(W_1, ...W_i, ..., W_m) - \psi(W_1, ...W_{i'}, ..., W_m)| \leq c_i \tag{15}$$

Then

$$Pr[\psi(S) - \mathbb{E}_S[\psi(S)] \geq \epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^m c_i^2} \tag{16}$$

Suppose that we only changed one of the samples W_i. Then we can use our main equation (11) above, to get

$$c_i = \frac{1}{m} \mathcal{O}\left(C(KW_i)^2 \left\| K \right\|^3 \right) \tag{17}$$

Plugging this in, we then get with probability at least $1 - \delta$,

$$\psi(S) \leq \mathbb{E}[\psi(S)] + \sqrt{\left(\frac{\ln(1/\delta) C(K)^2 \left\| K \right\|^3}{m} \right)} \tag{18}$$

Note that this equation (A.4.3) essentially is the main term found in Theorem 3.1.

To bound $\mathbb{E}_S[\psi(S)]$ from equation A.4.3, we use the standard definition of Rademacher complexity:

$$\mathbb{E}_S[\psi(S)] \leq 2\mathcal{R}_m(\mathcal{K}) \tag{19}$$

where Rademacher complexity is defined for our case as:

$$\mathcal{R}_m(\mathcal{K}) = \frac{1}{m} \mathbb{E}_\sigma \left[\sup_{K \in \mathcal{K}} \sum_{i=1}^m \sigma_i C_i(K) \right] \tag{20}$$

To ease on notation, assume $\sup_K = \sup_{K \in \mathcal{K}}$. We use the following technique:

$$\mathbb{E}_{\sigma_1,...,\sigma_{m-1}} \left[\sup_K \sum_{i=1}^{m-1} \sigma_i C_i(K) \right] \tag{21}$$

$$= \mathbb{E}_{\sigma_1,...,\sigma_m} \left[\frac{1}{2} \left(\sup_K \sum_{i=1}^{m-1} \sigma_i C_i(K) + \sigma_m C_m(K) + \sup_{K'} \sum_{i=1}^{m-1} \sigma_i C_i(K') - \sigma_m C_m(K') \right) \right] \tag{22}$$

$$= \mathbb{E}_\sigma \left[\sup_{K,K'} \frac{1}{2} \left(\sum_{i=1}^{m-1} \sigma_i C_i(K) + \sigma_i C_i(K') + \sigma_m C_m(K) - \sigma_m C_m(K') \right) \right] \tag{23}$$

$$\leq \mathbb{E}_\sigma \left[\sup_{K,K'} \frac{1}{2} \left(\sum_{i=1}^{m-1} \sigma_i C_i(K) + \sigma_i C_i(K') + \sigma_m \left\| (K - K')W_m \right\|^3 \right) \right] \tag{24}$$

which implies that after unrolling the induction step m times,

$$\mathbb{E}_{\sigma_1,...,\sigma_m} \left[\sup_K \sum_{i=1}^m \sigma_i C_i(K) \right] \leq \mathbb{E}_\sigma \left[\sup_{K,K'} \frac{1}{2} \sum_{i=1}^m \sigma_i \left\| (K - K')W_i \right\|^3 \right] \tag{25}$$

$$= \mathbb{E}_\sigma \left[\sup_K \frac{1}{2} \sum_{i=1}^{m} \sigma_i \left\| K W_i \right\|^3 \right] \tag{26}$$

where the previous equality holds since $\mathcal{K}$ is a convex set, and thus the set of all possible differences $\mathcal{K} - \mathcal{K} = \mathcal{K}$.

This is upper bounded by:

$$\mathbb{E}_\sigma \left[\sup_K \frac{1}{2} \sum_{i=1}^{m} \sigma_i \left\| K \right\|^3 \left\| W_i \right\|^3 \right] = \mathbb{E}_\sigma \left[\sup_K \frac{1}{2} \left\| K \right\|^3 \sum_{i=1}^{m} \sigma_i \right] \tag{27}$$

where the last equality follows since $\|W_i\| = 1$.

We note that if $\sum_{i=1}^{m} \sigma_i < 0$, then the optimum K satisfies $\|K\| = 0$, and otherwise $\|K\|^3$ is maximized - abusing notation slightly, let $\sup_K \|K\|^3 = \|\mathcal{K}\|^3$. Hence the previous term from (27) is upper bounded by:

$$\leq \mathbb{E}_\sigma \left\| \mathcal{K} \right\|^3 \left[\frac{1}{2} \left| \sum_{i=1}^{m} \sigma_i \right| \right] = \mathcal{O}(\sqrt{m}) \tag{28}$$

where the last equation follows from a well known property of Rademacher variables, which then follows that $\mathcal{R}_m(\mathcal{K}) \leq \frac{\mathcal{O}(\sqrt{m})}{m} = \mathcal{O}(\frac{1}{\sqrt{m}})$.

Hence it follows that gathering all terms, we have finally:

$$gap \leq 2\mathcal{R}_m(\mathcal{K}) + \sqrt{\left(\frac{\ln(1/\delta) C(K)^2 \left\| K \right\|^3}{m} \right)} \leq \mathcal{O}\left(\frac{1}{\sqrt{m}} \left\| K \right\|^{3/2} \right) \tag{29}$$

Since from A.4.2, $\|K\| \sim \mathcal{O}(d_{obs}^{1/3})$, Theorem 3.1 presented from the main section follows.