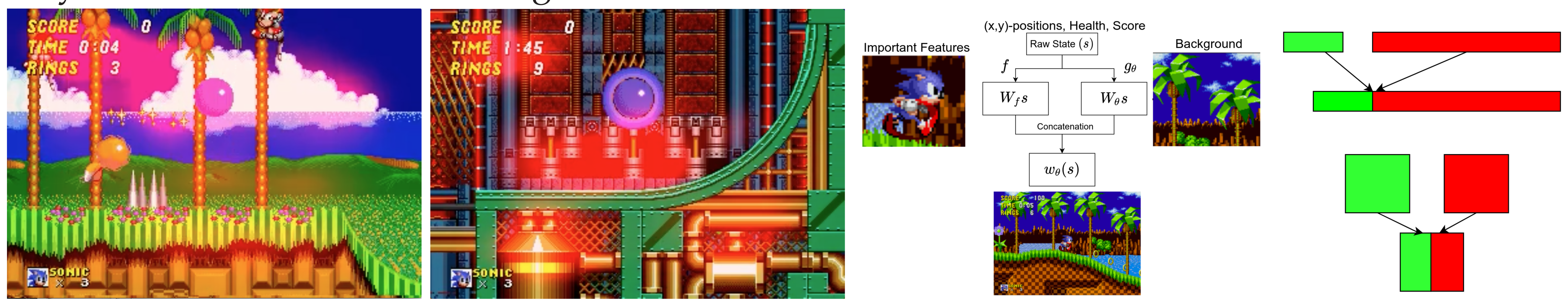
Observational Overfitting in Reinforcement Learning

Xingyou Song[1], Yiding Jiang[1], Stephen Tu[1], Yilun Du[2], Behnam Neyshabur[1],
[1] Google Research [2] MIT

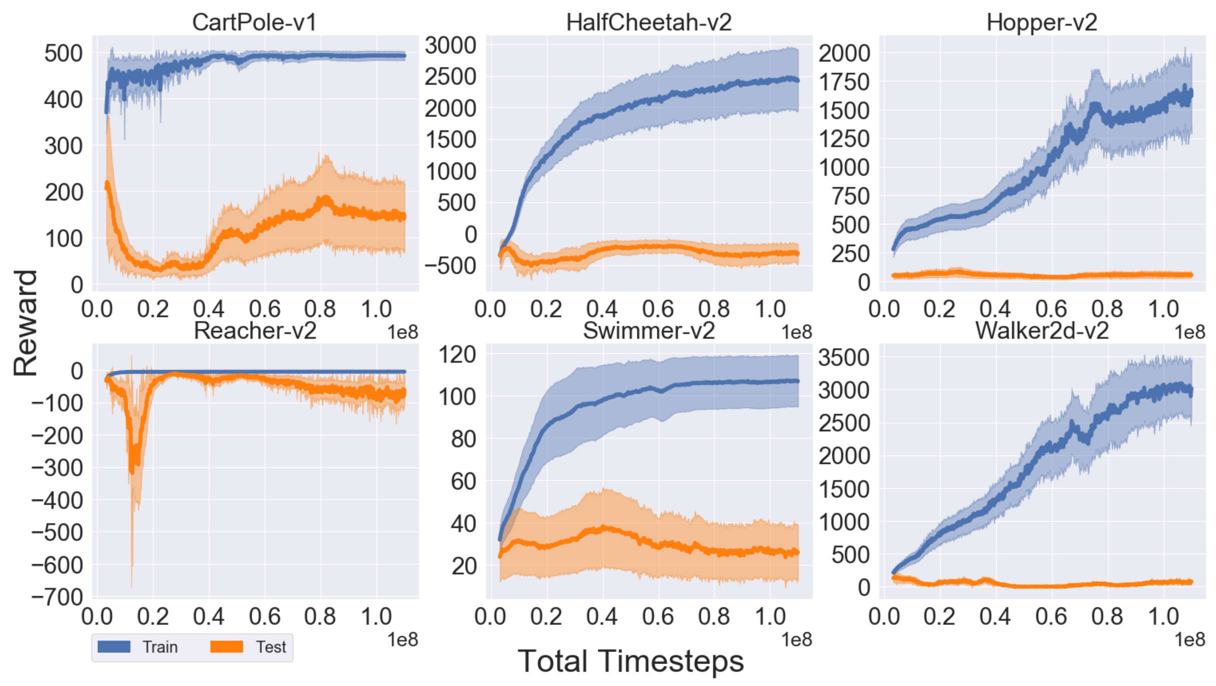
Observational Overfitting

In visually rich environments, the agent can overfit to *anything correlated with progress*. In Sonic the HedgeHog [1], saliency (red) shows agent has overfitted to the clock and background objects because they move backward while the agent moves forward.



We simplify this setting by only considering an underlying MDP, but generate multiple levels by varying the "observation" function $w_\theta(s)$. Observation function projects underlying MDP's state $s \rightarrow w_\theta(s)$ where $w_\theta(s) = h(f(s), g_\theta(s))$. $f(s)$ outputs generalizable features, $g_\theta(s)$ outputs non-generalizable features, $h(\cdot)$ is a concatenation function.

Examples: 1. $f(s)$ is Sonic, $g_\theta(s)$ is the background, $h(\cdot)$ is image rendering. 2. $f(s) = W_f s$, $g_\theta(s) = W_\theta s$, h is 1D concatentation. 3. $f(s), g_\theta(s)$ both deconvolutions but g_θ uses varying weights; $h(\cdot)$ is half-half image concatenation.



This setup causes 1D case to overfit, and is **not limited to 2D image background** (e.g. changing shapes and colors). This suggests something more *principled* is happening, unrelated to "real world images". Using our setup, we can transform *any* objective $C(P) \rightarrow C\left(K \begin{bmatrix} W_f \\ W_\theta \end{bmatrix}\right)$. If $P_\star$ is the unique minimizer of the original cost function $C(P)$, there can exist multiple solutions for high dimensional case, e.g. $\begin{bmatrix} \alpha W_f P_\star^\intercal \\ (1-\alpha) W_\theta P_\star^\intercal \end{bmatrix}^\intercal$ $\forall \alpha$. This extra bottom component $W_\theta P_\star^\intercal$ causes overfitting.

Theory

Simple case: one-step LQR convex objective $C(K; W_\theta) = \frac{1}{2} \left\| I + K \begin{bmatrix} W_f \\ W_\theta \end{bmatrix} \right\|_F^2 \implies \nabla C(K; W_\theta) = \left(I + K \begin{bmatrix} W_f \\ W_\theta \end{bmatrix} \right) \begin{bmatrix} W_f \\ W_\theta \end{bmatrix}^\intercal \implies \nabla^2 C(K; W_\theta) = \begin{bmatrix} W_f \\ W_\theta \end{bmatrix} \begin{bmatrix} W_f \\ W_\theta \end{bmatrix}^\intercal$. Hessian $\nabla^2 C(K; W_\theta)$ is degenerate due to extra observation dimension, which means non-degenerate part of initialized policy (e.g. if using Gaussian initialization) cannot reach to generalized minimizer using only gradient descent. The generalization gap must exist in this setting, establishing a lower bound.

References

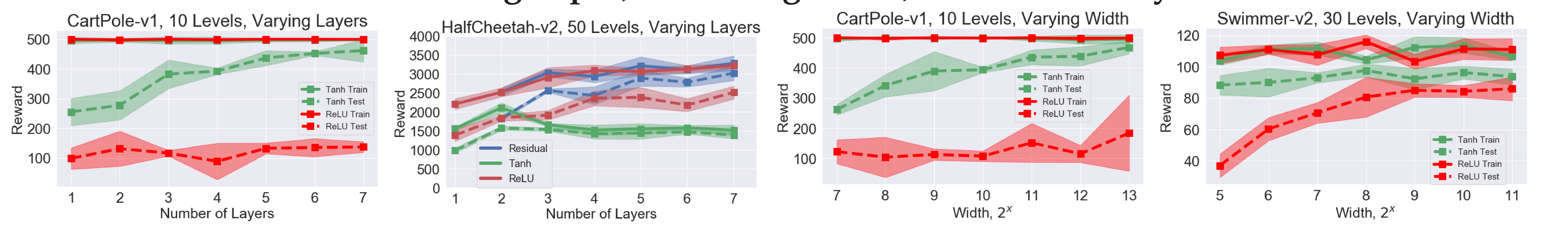
[1] Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. Gotta learn fast: A new benchmark for generalization in RL. *CoRR*, abs/1804.03720, 2018.

[2] Behnam Neyshabur. Implicit regularization in deep learning. *CoRR*, abs/1709.01953, 2017.

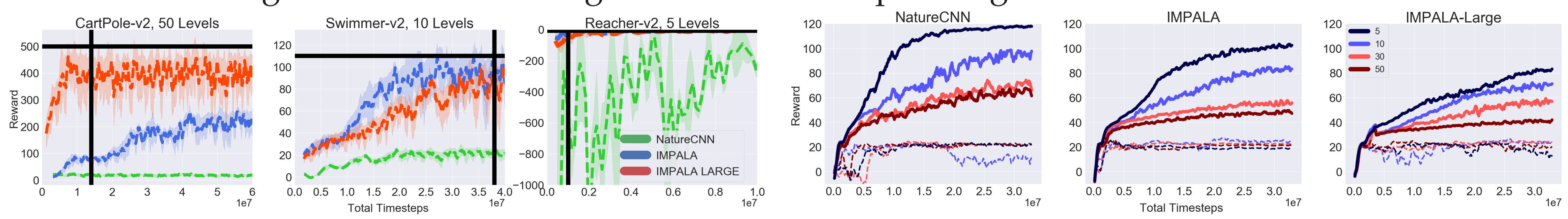
[3] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. *CoRR*, abs/1812.02341, 2018.

Implicit (Architectural) Regularization

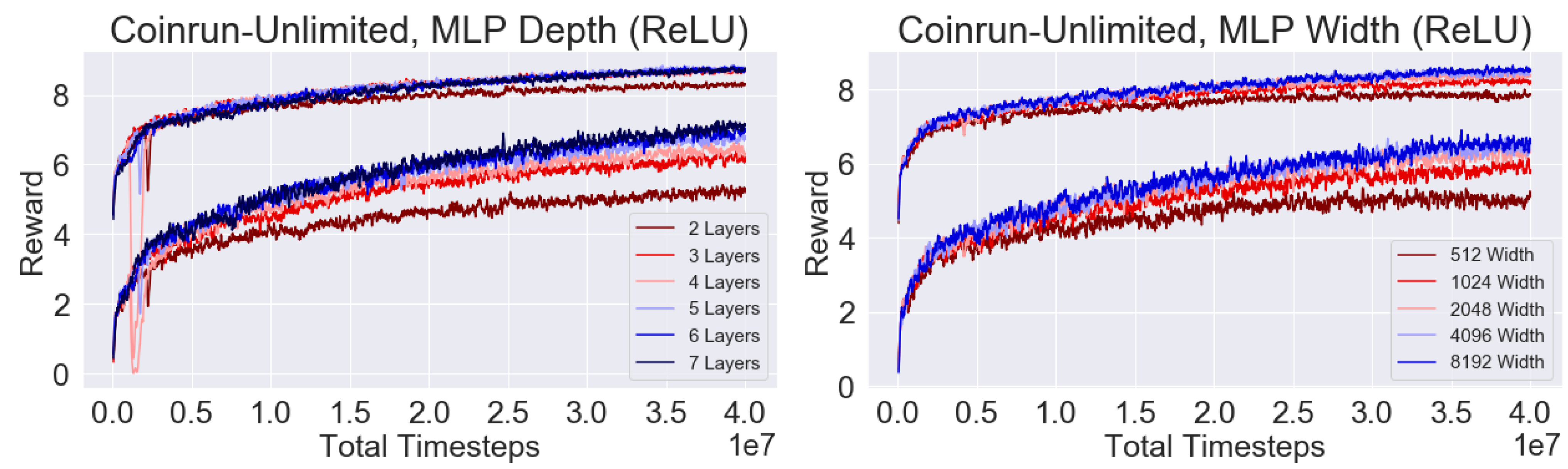
Implicit Regularization [2] is an important part of improving generalization for both MLP's and Convolutions. These include: **increasing depth, increasing width, and residual layers.**



(Left) Using example (3), the same ranking in generalization from [3] (NatureCNN, IMPALA, IMPALA-LARGE) exists. **(Right)** If observation only comes from "background" $g_\theta(s)$, memorization ranking is the *reverse* of the generalization ranking → evidence of implicit regularization.



Similar case when using MLP's on CoinRun when increasing depth and width.



Metrics to Study

Key observations: **(1)** If using nonconvex LQR with an observational setup, increasing observation dimension increases gap. **(2)** SL techniques are poor ways to predict RL generalization gaps. If policy $K = K_0 K_1, ..., K_j$ is overparametrized (**more layers, more width**), well-known SL bounds are poor predictors of generalization gap. **(3)** Similarly, SL margin distributions are poor ways if checking policy's discrete action margins; the norm of weights dominates everything.

Conclusion: our **theoretical understanding of deep RL generalization is poor**.