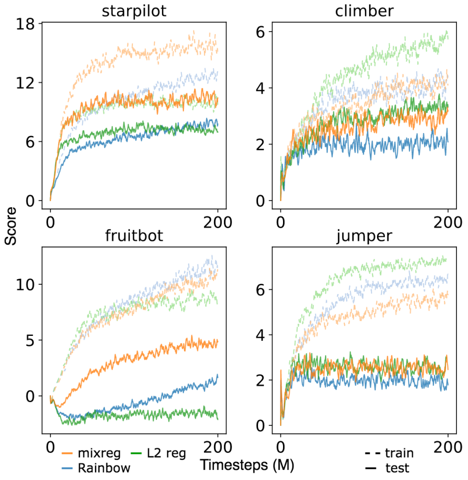
To R1: ▸ Originally we selected `crop`, `cutout-color` and `random conv` from RAD paper based on their Table 2 and Figure 3b. However, we found their released code of `crop` is confusing (cropping 64x64 window out of 64x64 observation; with unsolved github issue). Thus, we exclude `crop` from experiments. ▸ We agree with “it *usually* does not match”. But mixing-obs baseline is similar to data augmentation where input s is perturbed and supervision is from actual s. We thus include it for completeness. ▸ The entropy term in plain PPO is calculated using current policy instead of π_{old}; in mixreg we replace s with the mixed $\tilde{s}$ for entropy calculation. We will correct Eqn (14), (18) and cite the suggested paper. ▸ We agree that mixing “exposes static parts” but we found mixreg favors games with dynamic backgrounds (see Fig. 15). This is worth future investigation. ▸ In the finetuning stage, mixreg is not applied. We use plain PPO to finetune policies trained with different methods (plain PPO, mixreg). ▸ From Fig. 12, when $\alpha = 1$, the performance drop is noticeable in some games (e.g. `starpilot`, `climber`, `fruitbot`). ▸ Mixreg is more like data augmentation methods, so we combine it with regularization techniques (e.g. L2 regularization) instead of other augmentations.

To R2: ▸ Thanks for recognizing our contribution. Although our method is simple, the empirical results are surprisingly good. Besides, it also gives some intriguing observations: (1) it seems to make little sense to mix rewards or Q-values from different environments but the performance is good; (2) despite being discussed in the original mixup paper, applying mixup to RL has been overlooked since, even in three recent papers [1, 2, 3] about using data augmentation in RL. Therefore, we believe our work can inspire new insights into the important topic of generalization in RL.

To R3: ▸ Though it is straightforward, applying mixup to RL has been overlooked, even in three recent papers [1, 2, 3] about using data augmentation in RL. Besides, we do not think our findings are trivial. Other reviewers also agree that our work “hits the spot of intuitively a good idea, yet somehow wasn’t done in the literature yet” (R1), and “would be a valuable contribution and of interest to the community” (R2 & R4). ▸ We disagree with the reviewer’s remark that policy / value based methods are simply examples of regression / classification problems. It diminishes the great progress in the whole RL field. ▸ We will add two prior works mentioned. But the second one is not publicly available before submission deadline. ▸ We thank the reviewer for providing additional feedback on better understanding how mixreg works. We will definitely further pursue along this direction. But lacking some theoretical analysis does not diminish the value of our work. Characterizing the increased training diversity (or the increased number of MDP) is a good direction for future investigation. To analyze changes in the network’s representation, we have tried to visualize the hidden features using t-SNE but did not observe meaningful explanation, so we only include the quantitative results of finetuning experiments. We will do further representation analysis. Regarding failure cases of our method, we observe that mixreg struggles in maze-like environments (e.g. `maze`, `miner`) and environments where object color contains important information (e.g. `plunder` where the agent controls a ship to destroy enemy ships marked by different colors while avoid hitting friendly ships marked by same colors).

To R4: ▸ As demonstrated in Appendix B, we find mixreg helps decrease the Lipschitz constant of the learned network, which coincides with analysis on mixup in supervised learning context [4]. Smaller Lipschitz constant may lead to smoother policy and better generalization, though deeper reason on why mixing improving generalization is still unclear and worth further exploration. ▸ We conduct additional experiments for Rainbow, but due to limited time for rebuttal, we only manage to finish evaluating Rainbow with L2 regularization on 4 environments (see right figure). Our mixreg is on par with or outperforms L2 regularization for Rainbow. We will include the complete results in the final version. ▸ For evaluating the scalability to different model sizes, we choose multiplying the number of convolutional channels by 2 and 4 for a fair comparison with the baseline in Procgen benchmark.



Reference

[1] Kostrikov, Ilya, Denis Yarats, and Rob Fergus. "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels." *arXiv preprint arXiv:2004.13649* (2020).

[2] Laskin, Michael, et al. "Reinforcement Learning with Augmented Data." *arXiv preprint arXiv:2004.14990* (2020).

[3] Raileanu, Roberta, et al. "Automatic Data Augmentation for Generalization in Deep Reinforcement Learning." *arXiv preprint arXiv:2006.12862* (2020).

[4] Carratino, Luigi, et al. "On Mixup Regularization." *arXiv preprint arXiv:2006.06049* (2020).