On the Feasibility of Cross-Task Transfer with Model-Based Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract: Reinforcement Learning (RL) algorithms can solve challenging control problems directly from image observations, but they often require millions of environment interactions to do so. Recently, model-based RL algorithms have greatly improved sample-efficiency by concurrently learning an internal model of the world, and supplementing real environment interactions with imagined rollouts for policy improvement. However, learning an effective model of the world from scratch is challenging, and in stark contrast to humans that rely heavily on world understanding and visual cues for learning new skills. In this work, we investigate whether internal models learned by modern model-based RL algorithms can be leveraged to solve new, distinctly different tasks faster. We propose Model-Based **Cross**-Task **Tra**nsfer (**XTRA**), a framework for sample-efficient online RL with scalable pretraining and finetuning of learned world models. By proper pretraining and concurrent cross-task online fine-tuning, we achieve substantial improvements over a baseline trained from scratch; we improve mean performance of model-based algorithm EfficientZero by 23%, and by as much as 71% in some instances.

Keywords: Model-Based Reinforcement Learning, Pretraining

1 Introduction

Reinforcement Learning (RL) has achieved great feats across a wide range of areas, most notably game-playing [1, 2, 3, 4]. However, traditional RL algorithms often suffer from poor sample-efficiency and require millions (or even billions) of environment interactions to solve tasks – especially when learning from high-dimensional observations such as images. This is in stark contrast to humans that have a remarkable ability to quickly learn new skills despite very limited exposure [5]. In an effort to reliably benchmark and improve the sample-efficiency of image-based RL across a variety of problems, the Arcade Learning Environment (ALE; [6]) has become a long-standing challenge for RL. This task suite has given rise to numerous successful and increasingly sample-efficient algorithms [1, 7, 8, 9, 10, 11, 12], notably most of which are model-based, *i.e.*, they learn a *model* of the environment.

Most recently, EfficientZero Ye et al. [12] – a model-based RL algorithm – has demonstrated impressive sample-efficiency, surpassing human-level performance with as little as 2 hours of real-time game play in select Atari 2600 games from the ALE. This achievement is attributed – in part – to the algorithm concurrently learning an internal *model* of the environment from interaction, and using the learned model to *imagine* (simulate) further interactions for planning and policy improvement, thus reducing reliance on real environment interactions for skill acquisition. However, current RL algorithms – including EfficientZero – are still predominantly assumed to learn both perception, model, and skills *tabula rasa* (from scratch) for each new task. On the contrary, humans rely heavily on prior knowledge and visual cues when learning new skills. For example, a study found that human players easily pick up on visual cues about game mechanics and objectives when exposed to a

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

video game for the very first time, and that human performance is severely degraded if such cues are removed or conflict with prior experiences [5].

In this work, we explore whether such positive transfer can be induced with current model-based RL algorithms in an *online* RL setting, and across *markedly distinct* tasks. Based on our findings, we propose Model-Based **Cross**-Task **Tra**nsfer (**XTRA**), a framework for sample-efficient online RL with scalable pretraining and finetuning of learned world models using extra, auxiliary data from other tasks. Concretely, our framework consists of two stages: *(i) offline multi-task pretraining* of a world model on an offline dataset from m diverse tasks, a *(ii) finetuning* stage where the world model is jointly finetuned on a *target task* in addition to the m offline tasks. By leveraging offline data both in pretraining and finetuning, XTRA overcomes the challenges of catastrophical forgetting. To prevent harmful interference from certain offline tasks, we adaptively re-weight gradient contributions in an unsupervised manner based on similarity to the target task.

2 Model-Based Cross-Task Transfer

2.1 Offline Multi-Task Pretraining

In this stage, we aim to learn a single world model with general perceptive and dynamics priors across a diverse set of offline tasks. We emphasize, however, that the goal of pretraining is not to obtain a truly generalist agent, but rather to learn a good initialization for finetuning to unseen tasks. Learning a single RL agent for a diverse set of tasks is however a difficult in practice, which is only exacerbated by extrapolation errors due to the offline RL setting [13]. To address the challenge of multi-task learning, we propose to pretrain the model following a *student-teacher* training setup, where *teacher* models are trained separately by offline RL for each task, and then distilled into a single multi-task model using a novel instantiation of the MuZero Reanalyze [14] algorithm.

For each pretraining task we assume access to a fixed dataset $\{\hat{\mathcal{D}}^i \mid 1 \leq i \leq m\}$ that consists of trajectories from an unknown (and potentially sub-optimal) behavior policy. Importantly, we do *not* make any assumptions about the quality or the source of trajectories in the dataset, *i.e.*, we do not assume datasets to consist of expert trajectories. We first train individual EfficientZero *teacher* models on each dataset for a fixed number of iterations in a single-task (offline) RL setting, resulting in m *teacher* models $\{\hat{\pi}^i_\psi \mid 1 \leq i \leq m\}$. After training, each *teacher* model $\hat{\pi}^i_\psi$ has learned to produce task-specific quantities ($\hat{\pi}$, $\hat{u}$, $\hat{z}$) for a given game $\hat{\mathcal{M}}^i$. Next, we learn a *multi-task student* model $\hat{\pi}_\theta$ by distilling the task-specific teachers into a single model. Specifically, we optimize the student policy by sampling data uniformly from all pretraining tasks, and generate value/policy targets using the respective teacher models rather than bootstrapping from student predictions as commonly done in the (single-task) MuZero Reanalyze algorithm. This step can be seen as learning multiple tasks simultaneously with direct supervision by distilling predictions from multiple teachers' into a single model. Empirically, we find this to be a key component in scaling up the number of pretraining tasks. Although teacher models may not be optimal depending on the provided offline datasets, we find that they provide stable (due to fixed parameters during distillation) targets of sufficiently good quality. The simpler alternative – training a multi-task model on all m pretraining tasks simultaneously using RL is found to not scale beyond a couple of tasks in practice. After distilling teacher models into the multi-task student model, we now have a single set of pretrained parameters that can be used for finetuning to a variety of tasks via online interaction, which we introduce in the following section.

2.2 Online Finetuning on a Target Task

In this stage, we iteratively interact with a target task (environment) to collect interaction data, and finetune the pretrained model on data from the target task. However, we empirically observe that directly finetuning the pretrained model often leads to catastrophical forgetting, and consequently poor performance on the target task. To overcome this challenge, we retain offline data from the pretraining stage, and concurrently finetune the model on both data from the target task, as well as data from the pretraining tasks. While this procedure addresses catastrophical forgetting, interference

between the target task and certain pretraining tasks can be harmful for the sample-efficiency during online RL. As a solution, gradient contributions from offline tasks are periodically re-weighted in an unsupervised manner based on their similarity to the target task.

At each training step t, we jointly optimize the target online task $\mathcal{M}$ and m offline (auxiliary) tasks $\{\hat{\mathcal{M}}^i \mid \hat{\mathcal{M}}^i \neq \mathcal{M}, 1 \leq i \leq m\}$ that were used during the *offline multi-task pretraining* stage. Our online finetuning objective is defined as $\mathcal{L}_t^{\text{adapt}}(\theta) = \mathcal{L}_t^{\text{ez}}(\mathcal{M}) + \sum^i \eta^i \mathcal{L}_t^{\text{ez}}(\hat{\mathcal{M}}^i)$ where $\mathcal{L}^{\text{ez}}$ is the ordinary (single-task) EfficientZero objective, and η^i are dynamically (and independently) updated task weights for each of the m pretraining tasks. The target task loss term maintains a constant task weight of 1.

In order to dynamically re-weight task weights η^i throughout the training process, we break down the total number of environment steps (*i.e.*, 100k in our experiments) into even T-step cycles (intervals). Within each cycle, we spend first N-steps to compute an updated η^i corresponding to each offline task $\hat{\mathcal{M}}^i$. The new η^i will then be fixed during the remaining $T - N$ steps in the current cycle and the first N steps in the next cycle. We dynamically assign the task weights by measuring the "relevance" between each offline task $\hat{\mathcal{M}}^i$ and the (online) target task $\mathcal{M}$ by gradient cosine similarity. While re-weighting task weights at every gradient update would result in the least amount of conflicting gradients, it is prohibitively costly to do so in practice. However, we empirically find the cosine similarity of task gradients to be strongly correlated in time, *i.e.*, the cosine similarity does not change much between consecutive gradient steps. By instead updating task weights every N steps, our proposed technique mitigates gradient conflicts at a negligible computational cost in contrast to the compute-intensive gradient modification method proposed in [15].

3 Experiments

Experimental setup. We base our architecture and backbone learning algorithm on EfficientZero [12] and focus our efforts on the pretraining and finetuning aspects of our problem setting. We consider EfficientZero with two different network sizes to better position our results: *(i)* the same network architecture as in the original EfficientZero implementation which we simply refer to as **EfficientZero**, and *(ii)* a larger variant with 4 times more parameters in the representation network (denoted **EfficientZero-L**). We use the EfficientZero-L variant as the default network for our framework through our experiments, unless stated otherwise. However, we find that our EfficientZero baseline generally does not benefit from a larger architecture, and we thus include both variants for a fair comparison. We experiment with cross-task transfer on three subsets of tasks: tasks that share *similar* game mechanics (for which we consider two ***Shooter*** and ***Maze*** categories), and tasks that have no discernible properties in common (referred to as ***Diverse***). We measure performance on individual Atari games by absolute scores, and also provide aggregate results as measured by mean and median scores across games, normalized by either human performance or EfficientZero performance at 100k environment steps. All of our results are averaged across 5 random seeds to ensure reliability.

Baselines. We compare our method against **7** prior methods for online RL that represent the state-of-the-art on the Atari100k benchmark (including EfficientZero), as well as a multi-task behavior cloning policy trained on the full pretraining dataset, and a set of ablations that include EfficientZero with several different model sizes and pretraining/finetuning schemes. The former baselines serve to position our results with respect to the state-of-the-art, and the latter baselines and ablations serve to shed light on the key ingredients for successful multi-task pretraining and finetuning.

3.1 Results & Discussion

Tasks that share *similar* game mechanics. We first investigate the feasibility of finetuning models that are pretrained on games with *similar* mechanics. We select 5 shooter games and 5 maze games for this experiment. Results for our method, baselines, and a set of ablations on the Atari100k benchmark are shown in Table 1. We find that pretraining improves sample-efficiency substantially across most tasks, improving mean and median performance of EfficientZero by $\mathbf{23}\%$ and $\mathbf{25}\%$, respectively,

Table 1: **Scores on the Atari 100k benchmark (*similar* pretraining tasks).** Methods are evaluated after 100k environment steps. For each game, XTRA is first pretrained on all other 4 games from the same category. We include three main ablation results by removing cross-task optimization in finetuning (only online RL), the pretraining stage (random initialization), or task weights assignment (constant weights). We also include zero-shot performance of our method for target tasks in comparison to a behavioral cloning baseline. All numbers are means of 5 seeds with 32 evaluation episodes.

Category	Game	Efficient Zero	Efficient Zero-L	XTRA (Ours)	Ablations (XTRA) w.o. cross-task	w.o. pretraining	w.o. task weights	Zero-Shot BC	XTRA (Ours)
Shooter	Assault	1027.1	1041.6	**1294.6**	1246.4	1257.5	1164.2	0.0	92.8
	Carnival	3022.1	2784.3	**3860.9**	3544.4	2370.0	3071.6	93.75	719.3
	Centipede	3322.7	2750.7	5681.4	3833.2	**6322.7**	5484.1	162.2	1206.8
	Demon Attack	11523.0	4691.0	14140.9	6381.5	9486.8	**51045.9**	73.8	113.6
	Phoenix	10954.9	3071.0	14579.8	10797.3	9010.6	**22873.9**	0.0	8073.4
	Mean Improvement	1.00	0.69	1.36	1.02	1.11	**2.06**	0.02	0.29
	Median Improvement	1.00	0.83	1.28	1.15	0.82	**1.65**	0.01	0.24
Maze	Alien	695.0	641.5	**954.8**	722.8	703.6	633.6	108.1	294.1
	Amidar	109.7	84.2	90.2	**121.8**	70.8	69.7	0.0	5.2
	Bank Heist	246.1	244.5	**304.9**	280.1	225.1	261.4	0.0	7.3
	Ms Pacman	1281.4	1172.8	**1459.7**	1011.1	1122.6	809.2	147.6	448.9
	Wizard Of Wor	1033.1	928.8	985.0	**1246.1**	654.4	263.5	100.0	9.4
	Mean Improvement	1.00	0.90	**1.11**	1.06	0.82	0.70	0.07	0.17
	Median Improvement	1.00	0.92	**1.14**	1.11	0.88	0.64	0.10	0.05
Overall	Mean Improvement	1.00	0.79	1.23	1.04	0.96	**1.38**	0.05	0.23
	Median Improvement	1.00	0.91	**1.25**	1.12	0.85	1.04	0.02	0.16

Table 2: **Scores on the Atari 100k benchmark (diverse tasks)** The reported 5 XTRA results are from finetuning the same set of pretrained model parameters with the same 8 pretrained offline tasks. All numbers are computed for 5 seeds each with 32 evaluation episodes. All other results are adopted from EfficientZero[12].

Game	XTRA (Ours)	EfficientZero	Random	Human	SimPLe	OTRainbow	CURL	DrQ	SPR	MuZero
Assault	**1742.2**	1263.1	222.4	742.0	527.2	351.9	600.6	452.4	571.0	500.1
BattleZone	14631.25	13871.2	2360.0	37187.5	5184.4	4060.6	14870.0	12954.0	**16651.0**	7687.5
Hero	**10631.8**	9315.9	1027.0	30826.4	2656.6	6458.8	6279.3	3736.3	7019.2	3095.0
Krull	**7735.8**	5663.3	1598.0	2665.5	4539.9	3277.9	4229.6	4018.1	3688.9	4890.8
Seaquest	749.5	1100.2	68.4	42054.7	683.3	286.9	384.5	301.2	583.1	208.0
Normed Mean	**1.87**	1.29	0.0	1.0	0.70	0.41	0.75	0.62	0.65	0.77
Normed Median	0.35	0.33	0.0	1.0	0.08	0.18	0.36	0.30	**0.41**	0.15

overall. Interestingly, XTRA also had a notable zero-shot ability compared to a multi-game behavior cloning baseline that is trained on the same offline dataset. We also consider three ablations: *(1)* **XTRA without cross-task:** a variant of our method that naively finetunes the pretrained model without any additional offline data from pretraining tasks during finetuning, *(2)* **XTRA without pretraining:** a variant that uses our concurrent cross-task learning (*i.e., leverages offline data during finetuning*) but is initialized with random parameters (no pretraining), and finally *(3)* **XTRA without task weights:** a variant that uses constant weights of 1 for all task loss terms during finetuning. We find that XTRA achieves extremely high performance on 2 games (DemonAttack and Phoenix) without dynamic task weights, improving over EfficientZero by as much as $\mathbf{343}\%$ on DemonAttack. However, its median performance is overall low compared to our default variant that uses dynamic weights. We conjecture that this is because some (combinations of) games are more susceptible to gradient conflicts than others.

Tasks with *diverse* game mechanics. We now consider a more diverse set of pretraining and target games that have no discernible properties in common. Specifically, we use the following tasks for pretraining: Carnival, Centipede, Phoenix, Pooyan, Riverraid, VideoPinball, WizardOfWor, and YarsRevenge, and evaluate our method on 5 tasks from Atari100k. Results are shown in Table 2. We find that XTRA advances the state-of-the-art in a majority of tasks on the Atari100k benchmark, and achieve a mean human-normalized score of $\mathbf{187}\%$ vs. $\mathbf{129}\%$ for the previous SOTA, EfficientZero. This suggests that, while task similarity may play a role in the success of XTRA, the algorithmic advances of our proposed frames are a bigger factor in the strong empirical performance.

4 Conclusion

In this paper, we propose Model-Based **Cross**-Task **Tra**nsfer (**XTRA**), a framework for sample-efficient online RL with scalable pretraining and finetuning of learned world models using extra, auxiliary data from other tasks.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[2] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016. doi:10.1038/nature16961.

[3] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. W. Pachocki, M. Petrov, H. P. de Oliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019.

[4] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. Leveraging procedural generation to benchmark reinforcement learning. *ArXiv*, abs/1912.01588, 2020.

[5] R. Dubey, P. Agrawal, D. Pathak, T. L. Griffiths, and A. A. Efros. Investigating human priors for playing video games. In *ICML*, 2018.

[6] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents (extended abstract). In *IJCAI*, 2013.

[7] A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, D. Guo, and C. Blundell. Agent57: Outperforming the atari human benchmark. *ArXiv*, abs/2003.13350, 2020.

[8] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, R. Sepassi, G. Tucker, and H. Michalewski. Model-based reinforcement learning for atari. *ArXiv*, abs/1903.00374, 2020.

[9] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. P. Lillicrap, and D. Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 2020.

[10] I. Kostrikov, D. Yarats, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *ArXiv*, abs/2004.13649, 2021.

[11] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. *ArXiv*, abs/2010.02193, 2021.

[12] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao. Mastering atari games with limited data. In *NeurIPS*, 2021.

[13] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *ArXiv*, abs/2006.04779, 2020.

[14] J. Schrittwieser, T. K. Hubert, A. Mandhane, M. Barekatain, I. Antonoglou, and D. Silver. Online and offline reinforcement learning by planning with a learned model. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=HKtsGW-lNbw`.

[15] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.