다중 작업 강화 학습에서의 심플리셜 정규화 평가

조건우[1], 이수비[1], 이재문[2]
광주과학기술원[1], 서울대학교[2]
{joungju257, soobi_lee}@gm.gist.ac.kr, dlwoans0001@snu.ac.kr

Evaluating Simplicial Normalization in Multi-Task Reinforcement Learning

Geon woo Cho[1], Subi Lee[1], Jae moon Lee[2]
Gwangju Institute of Science and Technology[1], Seoul National University[2]
{joungju257, soobi_lee}@gm.gist.ac.kr, dlwoans0001@snu.ac.kr

Abstract

Multi-task reinforcement learning (MTRL) addresses key limitations of existing reinforcement learning (RL) methods, notably in generalization and sample efficiency. However, managing multiple tasks simultaneously remains a significant challenge. Various approaches, including curriculum learning, the mixture of experts, and parameter-sharing strategies, have been explored to improve MTRL performance. On the other hand, one of the recent research suggests that Simplicial Normalization (SimNorm), rather than ReLU, is an effective activation function for modeling the objective function on single task RL. In this paper, we investigate whether this claim extends to MTRL. We conducted experiments on two types of agents—one using ReLU and the other using SimNorm—within the Meta-world environments, comparing their total return and success rates. Our findings show that SimNorm appears to underperform compared to ReLU in the MTRL environments.

1 Introduction

RL has made significant progress in recent years, largely due to the integration of deep learning. Deep RL has enabled RL to handle complex, high-dimensional tasks by leveraging deep neural networks (DNNs) as function approximators. Despite these successes, deep RL typically requires a substantial amount of data and environment interaction, making it inefficient and impractical for real-world applications where data collection is expensive or time-consuming [1].

One of the key limitations of deep RL is its predominant focus on single-task learning. Most existing methods train separate policies for individual tasks, which not only limits the reusability of learned knowledge but also results in suboptimal sample efficiency when tackling a large number of tasks. To address these issues, MTRL has been proposed as a solution. MTRL aims to develop a single policy that can effectively handle multiple tasks by sharing representations and parameters across related tasks, improving both sample efficiency and generalization [2]. However, despite its potential, MTRL faces significant challenges. As the number and diversity of tasks increase, issues such as negative transfer and gradient conflicts arise, where learning one task adversely affects the performance of others [3]. Additionally, it is challenging to determine what knowledge

should be shared between tasks and how to share it efficiently.

Recent approaches to MTRL have attempted to address these problems through various kinds of techniques: curriculum learning which learns tasks in the appropriate order to efficiently extract information, mixture of experts which selects or mixes expert agents, learning multiple tasks by adding losses from different tasks, or partitioning parameters which are shared and not. Other approaches are using generative models, such as diffusion model [4, 5], or transformer model [6].

Recently, some studies have proposed that using SimNorm [7] instead of ReLU may offer advantages. They argue that a regularized model can help prevent suboptimal regions formed by biased gradients, and SimNorm contributes to this regularization [8]. We aim to evaluate whether SimNorm's smoothing of the objective can mitigate issues arising from gradient conflict in MTRL. In our work, we apply this solution to the current State-Of-The-Art (SOTA) MTRL model HarmoDT [6] in Meta-World [9] environments, focusing on evaluating suboptimality and success rates across different models. By systematically benchmarking these models, we aim to either validate or refute the efficacy of SimNorm in MTRL. Our analysis will contribute to a deeper understanding of its potential impact on improving model performance and generalization.

2 Preliminary

2.1 Multi-task Reinforcement Learning

In RL, the goal aims to learn a policy$\pi_\theta(a|s)$ That maximizes the expected cumulative discounted rewards in a Markov Decision Process (MDP), defined by the tuple (S, A, P, R, γ, ρ). Here, S is the state space, A is the action space, $P(s'|\, s, a)$ denotes the environment dynamics, $R\colon S \times A \to \mathbb{R}$ is the reward function, $\gamma \in [0,1]$ is the discount factor, and ρ is the initial state distribution. In an offline environment, a static dataset $D = \{(s, a, s', r)\}$ is provided. Offline RL algorithms learn policies exclusively from static datasets without the need for online interactions. The cumulative discounted reward (Eq. 1) measures the total expected reward that an agent can accumulate from a given state, considering future rewards with a discount factor γ [8].

$$\max J(\theta) = \max \mathbb{E}_{\substack{1\sim\rho(\cdot) \\ a_t\sim\pi_\theta(\cdot|s_t)}}\left[\nabla_\theta\left(\sum_{t=1}^{\infty}\gamma^t r(s_t, a_t)\right)\right] \quad (1)$$

In the MTRL setting, each task may have distinct reward functions, state spaces, and transition dynamics. Given a specific task $\mathrm{T} \sim p(\mathrm{T})$, the task specific MDP is represented as (S^{T}, A^{T}, P^{T}, R^{T}, γ^{T}, ρ^{T}).

2.2 Simplical Normalization

$$SimNorm(z) = [g_1, \cdots, g_L],\ g_i = Softmax(z_{i:i+V}) \quad (2)$$

SimNorm is the activation function dividing the latent space into continuous vector partitions and applies the softmax function to each partition, resulting in a probability distribution over the latent factors within that partition (Eq. 2). It introduces an approach to normalize latent representations by projecting them onto fixed-dimensional simplices. This continuous relaxation enables a smoother representation while inducing a degree of sparsity [7]. The PWM results indicate that the SimNorm activation function is an effective tool for enhancing the efficient learning of environmental dynamics in the world model [8].

3 Experiment

3.1 SimNorm in Multi-Objective Toy Environment

$$g(\theta) = x_0 + v\cos(\theta)t + \frac{1}{2}gt^2 \quad (3)$$

$$x = f(\theta) = \begin{cases} g(\theta) & if\ y_{contact} > h \\ w & else \end{cases} \quad (4)$$

$$(x_1, x_2) = f(\theta_1, \theta_2) = \begin{cases} \big(g(\theta_1), g(\theta_2)\big) & if\ y_{contact} > h \\ \big(w, g(\theta_2)\big) & else \end{cases} \quad (5)$$

$$(x_1, x_2) = f(\theta_1, \theta_2) = \begin{cases} \big(g(\theta_1), g(\theta_2)\big) & if\ y_{contact} > h \\ (w, w) & else \end{cases} \quad (6)$$

In this section, we experimentally demonstrate that using SimNorm as an activation function may not yield optimal performance in multi-objective optimization. To illustrate this, we design a toy environment inspired by PWM. In PWM, a toy environment is constructed as Eq. 4. They use a two-layer Multi-Layer Perceptron (MLP) to approximate the objective function, and then determine the launch angle θ, which minimizes the objective as approximated by the MLP. In this environment $g = 9.81$ is the gravitational acceleration, h and w represent the height of the wall and the distance from the launch point to the wall, respectively, (x_0, y_0) denotes the launch point, $v = 10$ is the launch velocity, and $t = 2$ is the time elapsed after the projectile is fired. We extend the toy environment from PWM by setting new objectives in Eq. 5 and Eq. 6. Eq. 5 represents an environment where tasks with and without walls coexist, while Eq. 6 represents an environment where both tasks involve walls.

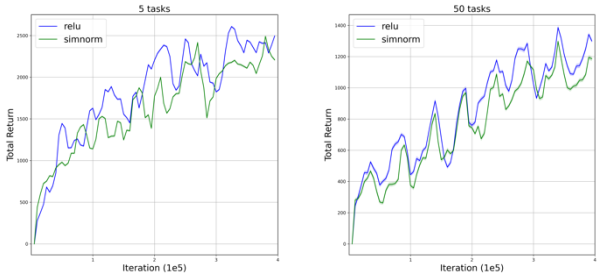
Table 1. Comparison of optimality gap between ReLU and SimNorm. Average optimality gap across 10 seeds.

Activation function	Opt. gap for Eq. 5	Opt. gap for Eq. 6
ReLU	0.43±0.20	1.52±0.79
SimNorm	1.71±0.41	4.13±0.67

To evaluate the smoothing effect of SimNorm in a multi-objective setting, we train the MLP under the same conditions as in PWM, using a model with two hidden layers and 32 neurons. The MLP is trained with the Adam optimizer with learning rate $\alpha = 2 \times 10^{-3}$, utilizing 1,000 uniform samples, a batch size of 50, and 100 epochs. SimNorm V is set to 8, as in PWM. Contrary to the single-objective case, in the multi-objective scenario, models using SimNorm exhibited a larger optimality gap compared to those using ReLU, as shown in Table 1. This suggests that SimNorm may not be as effective in optimization tasks within a multi-objective framework.

3.2 SimNorm in Multi-Task Reinforcement Learning

In this section, we observed that using SimNorm does not improve the performance of MTRL models. We conducted experiments on the Meta-World benchmark. To investigate the effect of SimNorm based on the number of tasks, MT50 (50 selected manipulation tasks) and MT5 (5 selected tasks) datasets are utilized. As a baseline model, we used HarmoDT-F, the SOTA MTRL model in Meta-World. The models were trained using a near-optimal dataset collected via SAC replay [10]. Performance comparisons between the models using ReLU and SimNorm were made by evaluating the averaged success rate across tasks.



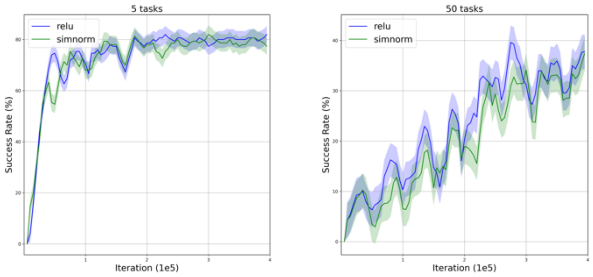


Fig1. Displayed from left and right are the results (total return and success rate) for 5 tasks and 50 tasks. Each result is averaged over 3 different seeds. Each task is evaluated for 50 episodes.

Table2. Comparison of final success rate between ReLU and SimNorm. Averaged over 3 different seeds.

Activation function	5 tasks	50 tasks
ReLU	81.33 ± 3.48	30.33 ± 3.47
SimNorm	78.67 ± 3.28	24.53 ± 3.61

As shown in Table 2, applying ReLU to HarmoDT-F yields a higher final success rate compared to SimNorm, with improvements of 2.66% in MT5 and 5.8% in MT50. Additionally, as illustrated in Fig. 1, the total return and success rate throughout the training process are overall higher with ReLU than with SimNorm. These findings empirically demonstrate that smoothing the objective with SimNorm does not ensure higher performance or faster convergence in MTRL.

4 Conclusion

In this study, we examined the effects of using ReLU versus SimNorm as activation functions in the MTRL model. Through evaluations conducted in both toy environments inspired by PWM and in the Meta-World benchmark, we found that SimNorm does not outperform ReLU in optimizing multi-objective tasks. Our results indicate that objective smoothing, as achieved through SimNorm, does not significantly enhance the training efficiency or performance of the MTRL model. In future research, we plan to explore alternative approaches to resolve gradient conflicts as a potential pathway to improve the performance of MTRL models.

5 References

[1] Levine, Sergey, et al., "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," in *arXiv*

preprint arXiv:2005.01643, 2020.
[2] Caruana, R., “Multitask Learning,” in *Machine Learning,* vol. 28, p. 41–75, 1997.
[3] Ruder, S., "An Overview of Multi-Task Learning in Deep Neural Networks," in *arXiv preprint arXiv:1706.05098*, 2017.
[4] Sohl-Dickstein, Jascha, et al., "Deep unsupervised learning using nonequilibrium thermodynamics,” in *International Conference on Machine Learning*, PMLR, p. 2256-2265, 2015.
[5] Zhu, Zhengbang, et al., "Diffusion models for reinforcement learning: A survey," in *arXiv preprint arXiv:2311.01223*, 2023.
[6] Hu, Shengchao, et al., "HarmoDT: Harmony Multi-Task Decision Transformer for Offline Reinforcement Learning," in *arXiv preprint arXiv:2405.18080*, 2024.
[7] Nicklas Hansen, Hao Su, and Xiaolong Wang., “Td-mpc2: Scalable, robust world models for continuous control”, in I*nternational Conference on Learning Representation,* 2024.
[8] Georgiev, Ignat, et al. "PWM: Policy Learning with Large World Models," in *arXiv preprint arXiv:2407.02466*, 2024.
[9] Yu, Tianhe, et al., "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning*, PMLR, 2020.
[10] Haarnoja, Tuomas, et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International Conference on Machine Learning*, PMLR, 2018.