META-WORLD CONDITIONAL NEURAL PROCESSES

Suzan Ece Ada, Emre Uğur
Department of Computer Science
Bogazici University
Istanbul, Turkey
{ece.ada,emre.ugur}@boun.edu.tr

ABSTRACT

We propose Meta-World Conditional Neural Processes (MW-CNP), a conditional world model generator that leverages sample efficiency and scalability of Conditional Neural Processes to allow an agent to sample from the generated world model. We intend to reduce the agent's interaction with the target environment as much as possible. Thus, MW-CNP meta-learns world models that use prior experience. Using the world model generated from MW-CNP the RL agent can be conditioned on significantly fewer samples collected from the target environment to imagine the unseen environment. We emphasize that the agent does not have access to the task parameters throughout training and testing.

1 INTRODUCTION

A diverse set of skills can be acquired through incremental learning in environments with emergent complexity. In line with this, world model generation has been an integral part of open-ended learning (Wang et al. (2019)). Although creating a variety of challenging simulation environments is a promising direction, it is bounded by the limitations of the physics engine and the predetermined set of environment parameters. We propose generating world models from the agent's own experience to overcome this problem.

Research on lifelong learning focus on learning incrementally and adapting rapidly to unknown tasks from incoming streams of data (Parisi et al. (2019)). Challenges associated with lifelong learning include catastrophic forgetting, negative transfer, and resource limitations. In addition, data to be processed after deployment and optimal model hyperparameters are not available a priori.

Meta reinforcement learning is an enduring area of interest that enables fast adaptation in test tasks. Recent works have integrated meta-learning to offline RL (Mitchell et al. (2020)) and lifelong learning (Finn et al. (2019); Nagabandi et al. (2018); Berseth et al. (2021)). In open-ended learning, a policy with high representational capacity can be utilized for increasingly challenging environments. In this work, we are interested in few-shot learning in settings where the transition dynamics of the environment change across tasks. More specifically, the agent is trained a the distribution of tasks with varying transition dynamics and expected to adapt to an unseen task with few samples from the target environment. Sampling in the real world is expensive; hence reducing the number of samples used for fast adaptation is an important research direction for sim-to-real RL.

In this work we are interested in meta-reinforcement learning settings where the reward function stays the same but the transition function changes across multiple tasks. In particular, each transition function depends on the environment parameters that are hidden during training and testing. More concretely, imagine a setting where an agent moves to a goal location in different environments parametrized by different force fields. Different force fields push the agent in different directions. Through trial and error, the agent learns to move robustly in environments with different transition dynamics. The agent is required to adapt quickly to a new target environment with hidden environment parameters and reward signals at test time.

We propose Meta-World CNP to generate world models from a few samples collected from the target environment. Our contributions include (1) generating and learning world models with no access to target environment parameters, (2) sample efficient, fast adaptation to the unseen test environment. Requires significantly less rollouts from the unseen target environment at test time compared to No

Reward Meta-Learning (NORML) Yang et al. (2019) and (3) utilizing offline datasets of Markov Decision Process tuples for training.

2 Preliminaries

2.1 Conditional Neural Processes

Conditional Neural Processes are proposed as a novel neural architecture by (Garnelo et al. (2018) to predict the parameters of a probability distribution conditioned on a permutation invariant prior data and target input. CNPs attempt to represent a family of functions using Bayesian Inference and the high representational capacity of neural networks. The architecture of CNPs consists of parameter sharing encoders and a decoder named the query network. In particular, first a random number of input and true output pairs $(x_{f^i}, y_{f^i}^{true})$ are sampled from the function $f^i \in F$. Each pair is encoded into a latent representation via the encoder networks. Then, these representations are averaged along the dimensions to account for the invariance to permutation and the number of inputs. The resulting representation is concatenated with the target input query and fed to the query network. The query network outputs the predicted mean and standard deviation for the queried input $(x_{f^i}^q)$.

2.2 No-Reward Meta Learning

NORML, proposed by (Yang et al. (2019)), is an extension of MAML-RL framework (Finn et al. (2017)) for settings where the environment dynamics change across tasks instead of the reward function. The goal of NORML is to utilize past experience to quickly adapt to tasks with unknown parameters from a few samples with missing reward signals. Provided that the change in dynamics can be represented in $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$, NORML learns an pseudo-advantage function $A_\psi\left(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}\right)$. It is important to note that the aim of A_ψ is to guide the meta-policy adaptation instead of fitting to the advantage function. A_ψ is used to compute task specific parameters in the MAML inner loop from a set of state transitions of task i D_i^{train} that does not contain reward signal. The learned advantage function is optimized in the MAML outer loop using the reward information present in rollouts D_i^{test} obtained from updated task-specific policy parameters.

3 Proposed Method: Meta-World Conditional Neural Process (MW-CNP)

In this section, we present our method, named Meta-World Conditional Neural Processes (MW-CNP), for learning world models using prior experience. We first describe the problem setup and then explain MW-CNP's structure in detail.

In few-shot learning, the goal is to quickly adapt to an unseen target task using a few labeled data in the target environment. Meta-World Conditional Neural Processes (MW-CNP) can reduce the number of samples required from the target environment by generating world models from fewer samples from the target environment with no access to the target environment parameter. These models can then be used to obtain inexpensive rollouts for finetuning at test time.

Online Meta-Learning We denote the initial state distribution as $\rho(s_0) : \mathcal{S} \to \mathbb{R}$, state transition distribution of $task_i$ as $\rho_i(s_{t+1}|s_t, a_t) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, and the reward function as $r_t : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. During meta-training we store transitions for each $task_i$ as a set of observations $B_i = \{(s_t, a_t, s_{t+1})\}_{t=0}^{n} \subset S \times A \times S$ without the task parameter. Hence during both training and testing the parameters of the state transition distribution are hidden.

After we obtain the meta-policy and learn the pseudo-advantage function $A_\psi\left(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}\right)$ using NORML, we train our MW-CNP model using the replay buffer of transitions $B = \{B_i\}_{i=1}^{n}$ where n denotes the number of environments.

Meta-World Conditional Neural Processes (MW-CNP) MW-CNP is trained in an offline fashion using unlabeled batches of Markov Decision Process (MDP) tuples collected during online meta-learning. Figure 1 illustrates the training procedure for MW-CNP. It is worth noting that the environment parameter is hidden during training and testing. In each training iteration, an unlabeled batch

$B_i = \{(s_t, a_t, s_{t+1})\}_{t=0}^{n}$ is randomly sampled from the offline dataset. Then, a set of task-specific MDP tuples $\{(s_k, a_k, s_{k+1})\}_k$ and a single MDP (s_q, a_q, s'_q) are randomly sampled from the chosen batch B_i. (s_q, a_q, s'_q) is used for target state-action query $[s_q, a_q]$ and true target next state label $[s'_q]$. Each MDP tuple (s_k, a_k, s_{k+1}) is encoded into a fixed size representation using a parameter sharing encoder network. These representations are then passed through an averaging module A to obtain a latent representation r of the hidden environment transition function used in batch B_i. The resulting latent representation r is concatenated with the $[s_q, a_q]$ to predict the distribution parameters μ_q, σ_q of the next state s'_q given the latent representation r and the target query $[s_q, a_q]$.

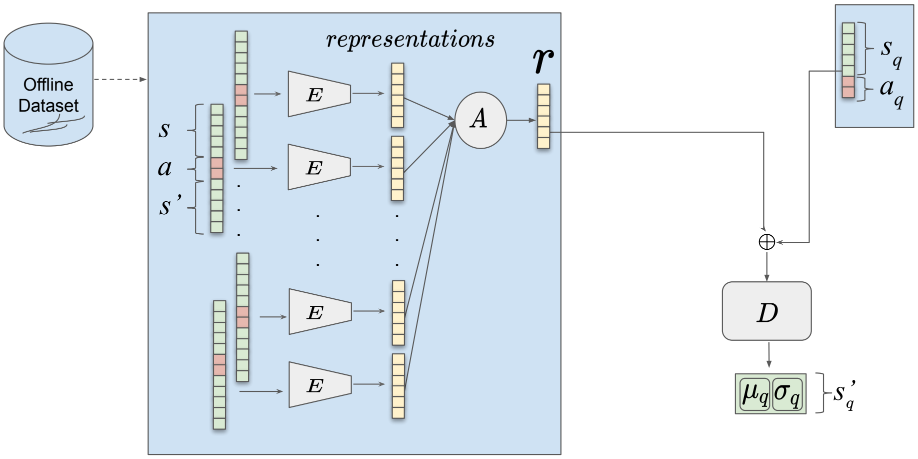

Figure 1: Structure and the training procedure of the Meta World-CNP

The loss function of MW-CNP can be expressed as

$$\mu_q, \sigma_q = f_{\theta_D}\left([s_q, a_q] \oplus \frac{1}{n}\sum_k^n g_{\theta_E}(s_k, a_k, s'_k)\right)$$

$$\mathcal{L}(\theta_E, \theta_D) = -\log P\left(s_q^{\prime true} \mid \mu_q, softplus(\sigma_q)\right)$$

where f_{θ_D}, g_{θ_E} are the decoder the encoder networks, $[s_q, a_q]$ is the target state action query, (s_k, a_k, s'_k) are the randomly sampled transitions from the set of observations B_i.

Figure 2 illustrates the test procedure of MW-CNP. At test-time, the agent is allowed to collect a few samples from the unseen target environment. These samples are encoded into representations of fixed size by an encoder network with shared weights. A shared representation of the target environment is obtained using an averaging module shown in Figure 2(I). Once the latent representation is obtained for the target environment with a hidden task parameter rollouts can be generated inexpensively. This representation is used to predict the parameters of the next-state distribution for the state-action query $[s_q, a_q]$.

For each MW-CNP generated rollout, same true initial state, sampled from the real target environment, is used. The inital state is fed to the stochastic meta-policy to obtain the action query a_q Figure 2(II). Then, the predicted next-state is used to sample the next action query until the episode is terminated. The rollouts ”hallucinated” from the MW-CNP are combined with the true rollout sampled from the real target environment. The resulting set of rollouts are fed to the pseudo-advantage network. The learned pseudo-advantage network learned during online meta-learning uses (s, a, s') tuple as input and outputs an advantage estimation value. Hence, once experience from the generated world model is collected, these experiences are fed to the learned advantage function in the form of (s, a, s'). Finally, meta policy is finetuned for fast adaptation to the target task using the

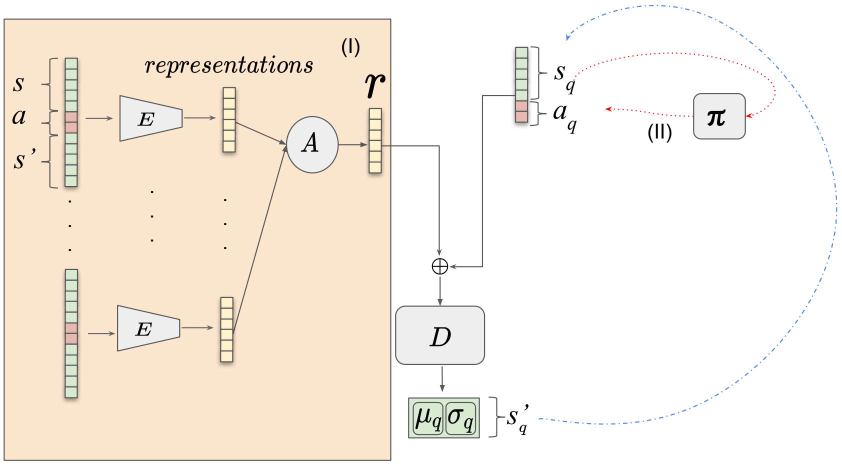

Figure 2: Structure and the test procedure of the MW-CNP

estimated advantage values and combined set of MW-CNP generated rollouts and a single target environment rollout.

4 Experiments

In this section, we analyze the performance of our method and compare it with NORML and the oracle in 2D point agent environment. The goal of the point agent, initialized at [x=0,y=0], is to move to the position [x=0,y=1], where x,y are the positions on the 2D plane. We are interested in a meta-RL setting where the reward function is identical across multiple tasks. Different tasks are created by generating different artificial force fields that push the agent in different directions (ϕ). We use the same reward function, the negative Euclidean distance from the goal position, and hyperparameters used in NORML for comparative analysis. In the Point Agent environment 5000 tasks are defined over the $[-\pi, \pi]$ interval.

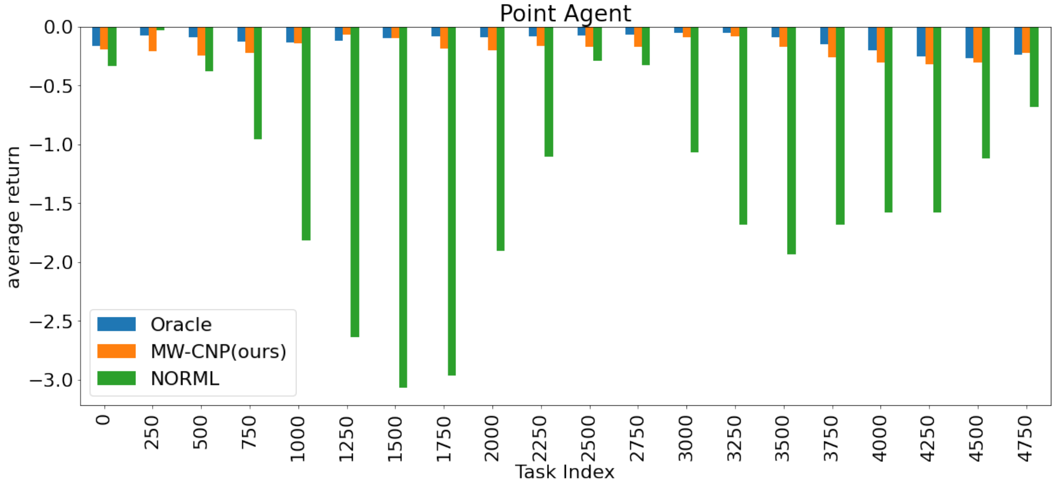

Figure 3: Expected reward (y-axis) of finetuning over generated data and ground truth data in target tasks. X-axis represents the target task indices.

The agent is initially trained across a distribution of 5000 tasks, i.e. in environment with 5000 different force fields. Then, it is tested in an unseen target task. The rollouts obtained from the target environment will be referred to as actual rollouts, whereas the rollouts generated from the MW-CNP will be named imagined rollouts. At test time, Oracle agent uses 25 actual rollouts from the unseen target environment. 25 rollouts were used in the original NORML experiment hence we use the same number of rollouts for comparison. The NORML agent and the MW-CNP agent are allowed to use only a single rollout for finetuning the meta-policy. By limiting the number of actual rollouts that can be used for finetuning we aim to compare sample efficiency of MW-CNP to the baseline NORML. The average return obtained from the target environment after finetuning the meta policy of NORML with 1 rollout are shown in Fig. 3 with green bars. As shown, when MW-CNP and NORML used the same amount of actual rollouts (1 actual rollout) from the target environment, MW-CNP outperforms NORML dramatically. Even though the oracle agent trained with NORML used a significantly higher number of actual rollouts than MW-CNP (25 to 1), their performances are similar, as shown in illustrations in Figures 4, 5, 6 and the bar plots in 3.

Figure 3 shows the average return obtained in the target environment over tasks. We compare results for NORML finetuning over 1 rollout, NORML finetuning over 25 rollouts (Oracle) and finetuning over a combination of 24 imagined rollouts from the MW-CNP model, and 1 actual rollout from the target environment. At test-time MW-CNP uses a total of 25 combined rollouts similar to the oracle NORML that uses 25 actual rollouts.

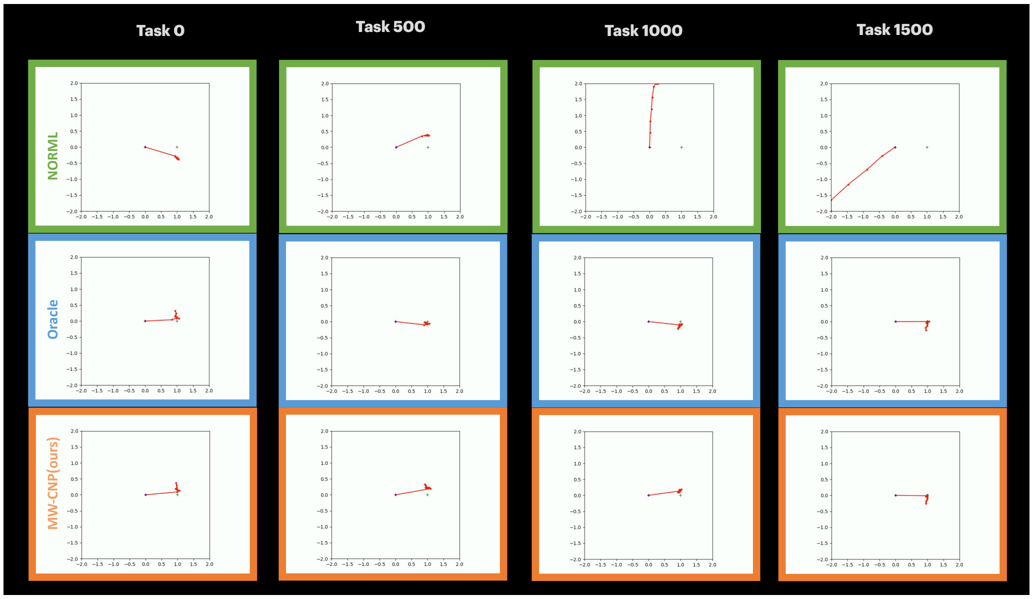

Figure 4: Illustrations of trajectories for Tasks 0, 500, 1000 and 1500

We condition the MW-CNP model on the same rollout used for 1 rollout-finetunig. The results show that the samples generated from the agent's imagination created by the MW-CNP generated sophisticated samples that can be used for finetuning the meta-policy for fast domain adaptation. All in all, MW-CNP requires significantly less interaction with the target environment compared to NORML, for fast adaptation to the unseen task.

5 Discussion

We showed that meaningful hallucinated rollouts can be collected using the MW-CNP framework to guide the meta-policy adaptation. We compared the expected return obtained from finetuning

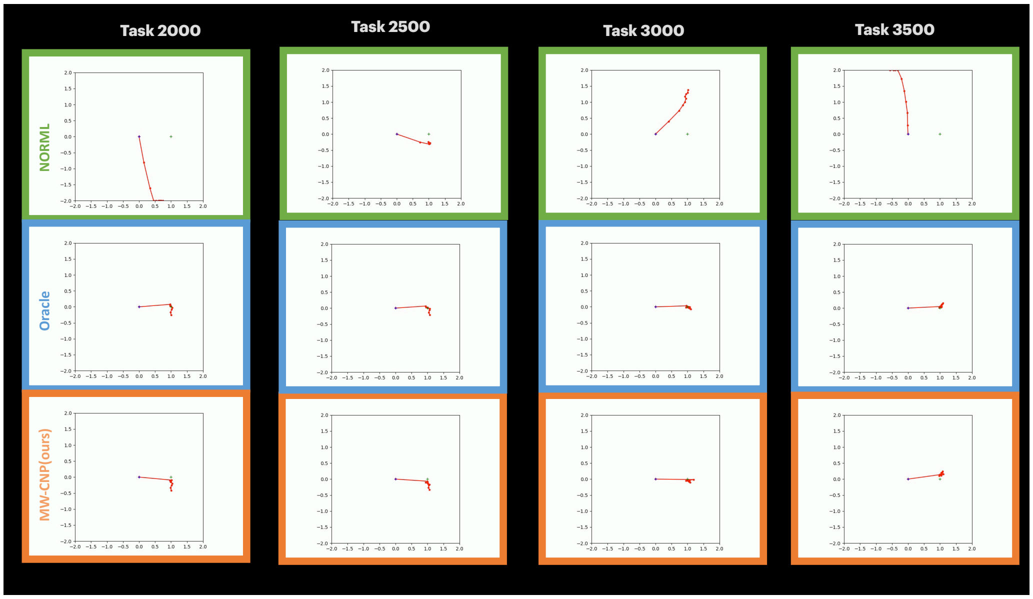

Figure 5: Illustrations of trajectories for Tasks 2000, 2500, 3000 and 3500

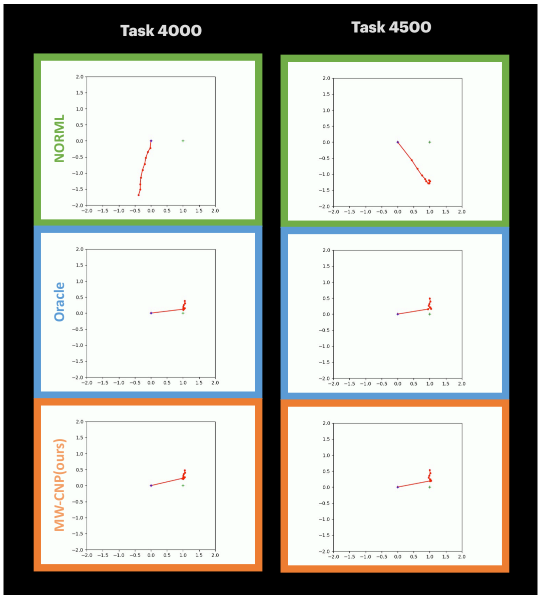

Figure 6: Illustrations of trajectories for Tasks 4000 and 4500

over generated data with MW-CNP and the ground truth data. While our initial experiments are conducted in low-dimensional state and actions spaces such as the point agent, CNPs were shown to perform well with high dimensional inputs like image data (Seker et al. (2019)). Therefore, we plan to extend our work and apply it to high-dimensional sensorimotor spaces, such as manipulator robots with RGB-D cameras.

REFERENCES

Glen Berseth, Zhiwei Zhang, Grace Zhang, Chelsea Finn, and Sergey Levine. Comps: Continual meta policy search. *CoRR*, abs/2112.04467, 2021. URL https://arxiv.org/abs/2112.04467.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1920–1930. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/finn19a.html.

Marta Garnelo, Dan Rosenbaum, Chris J. Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J. Rezende, and S. M. Ali Eslami. Conditional neural processes. *CoRR*, abs/1807.01613, 2018. URL http://arxiv.org/abs/1807.01613.

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. Offline meta-reinforcement learning with advantage weighting. *CoRR*, abs/2008.06043, 2020. URL https://arxiv.org/abs/2008.06043.

Anusha Nagabandi, Chelsea Finn, and Sergey Levine. Deep online learning via meta-learning: Continual adaptation for model-based RL. *CoRR*, abs/1812.07671, 2018. URL http://arxiv.org/abs/1812.07671.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.01.012. URL https://www.sciencedirect.com/science/article/pii/S0893608019300231.

Muhammet Yunus Seker, Mert Imre, Justus H Piater, and Emre Ugur. Conditional neural movement primitives. In *Robotics: Science and Systems*, 2019.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (POET): endlessly generating increasingly complex and diverse learning environments and their solutions. *CoRR*, abs/1901.01753, 2019. URL http://arxiv.org/abs/1901.01753.

Yuxiang Yang, Ken Caluwaerts, Atil Iscen, Jie Tan, and Chelsea Finn. Norml: No-reward meta learning. *arXiv preprint arXiv:1903.01063*, 2019.