NEVER FORGET - ALLEVIATE CATASTROPHIC FORGETTING FOR DRL MULTI-TASK GENERALIZATION

Benny Jiang, Sophia Yan, Gary Yang
{jiangxinhao, bunnyyan, lzyang}@berkeley.edu

ABSTRACT

Deep reinforcement learning policies, when trained with multi-task generalization in mind as in curriculum learning, will inevitably lead to the problem of Catastrophic Forgetting French (1999), in which the model trained on the latest task leads to performance degradation on earlier tasks.

One recent way to alleviate this problem in the domain of computer vision is parameter superposition (PSP). The intuition behind the method is that with the over-parameterization nature of contemporary neural networks, only a small subset of the parameters is most relevant to one task, and all these small subsets can be transformed by specific linear transformations into orthogonal alignments, such that when storing they would not affect each other.Cheung et al. (2019) We applied this approach in the domain of deep reinforcement learning, using a similar method to store the weights of the decision network in order to improve the multi-task generalization ability of the model. We may also combine this method with transfer learning between tasks to further enhance the training process.

We propose to use the Meta-World simulator Yu et al. (2020) built on top of MuJoCo, mainly because it offers a range of tasks that we can train and evaluate on. Of the 50 tasks offered, we plan on working with MT10, the multi-task benchmark offered with the least number of tasks (10) for easier coding, debugging, and evaluation. Figure 1 illustrates the specific tasks.

As the name of our topic suggests, our proposed research is directly related to Deep Reinforcement Learning.

We conducted an experiment with Policy-Gradients deep reinforcement learning with a simple feed-forward network on two tasks in sequence as a baseline. We applied PSP to the policy network, and it significantly alleviate catastrophic forgetting, while different tasks still interfere in the training process obviously. We performed a parameter sweep and used the best candidate parameters in the following experiments. After that, we applied PSP to a much deeper and larger MLP policy network. The interference between tasks in training process is reduced, and the advantage of PSP became more palpable, which demonstrates that PSP can make better use of sparsity in parameter space. We also did experiments on both Binary superposition and Complex superposition to see different outcome based on different kinds of settings. Finally, we applied the same techniques on different combinations of environment. We explored the benefit of superposition in combination with similar context (such as window or door), similar movement (such as closing the door and closing the window), and a series of movement (such as slide side, slide back, and slide side back). We draw conclusions based on the space sparsity of different environment.

In conclusion, by applying superposition to policy network with appropriate parameters and networks, our modified version of superposition outperforms the origin setting in regards of catastrophic forgetting. PSP on policy network significantly improves the catastrophic forgetting problem under the MetaWorld environment.

1 INTRODUCTION

1.1 MOTIVATION

The ability to learn in a sequential order is essential in the development of artificial intelligence. However, neural networks are not capable enough for it currently, and catastrophic forgetting is an inevitable phenomenon in this area. We are interested in adopting superposition to help with the capability of sequentially learning without catastrophic forgetting.

1.2 RELATED WORK

In this section, we will briefly review the main problem and some previously proposed methods and knowledge that are related to our work. Catastrophic forgetting is the main problem to deal with in our work, and we have a brief overview of it at the beginning. After that, we go over the Multi-task learning which is a cornerstone of our work. Finally, since parameter superposition is the core method that we used for solving multitasking problem, we review some related work and methods in the field of superposition.

Catastrophic Forgetting In this project, catastrophic forgetting problem is the main problem upon which we want to make improvement. Catastrophic forgetting is a challenge to many machine learning models and algorithm. When train on two sequential tasks, many machine learning models ”catastrophically forgot” the way to perform the first task. Catastrophic forgetting is widely believed to be a serious problem for neural networks.

We start with French (1999), a milestone paper in this area, in which the causes, consequences and numerous solutions to the problem of catastrophic forgetting in neural networks are examined. After that, we go over an investigation via a latest work Cheung et al. (2019) They investigate the extent to which the catastrophic forgetting problem occurs for modern neural networks by comparing both established and recent gradient-based training algorithms and activation functions. We also review James et al. (2016), whose approach remembers old tasks by selectively slowing down learning on the weights important for those tasks, so that it is possible to overcome the catastrophic forgetting and maintain expertise after a long time. In the next step, we review Joan et al. (2018), which propose a task-based hard attention mechanism that preserves previous information of the task without affecting the learning of the current task. Finally, we look at Craig et al. (2018), which propose a model that overcomes catastrophic forgetting in sequential reinforcement learning by combining ideas from continual learning in both the image classification domain and the reinforcement learning domain.

Multi-task learning Multi-task learning is the cornerstone towards generalized intelligence. Currently, deep neural networks perform relatively well in tasks such as image classification like Dosovitskiy et al. (2020), object detection like Wang et al. (2020) and text generation like Brown et al. (2020). However these models are good at their own tasks, but when the context change just a bit they fail spectacularly. Indeed a model trained to find dogs cannot find humans, and a model trained on the English language cannot fathom Chinese correctly. Thus multi-task learning has always been a trend in research today. Multi-task comes in two flavors - multi-task on generalization and multi-task in tasks. The first flavor comes into play is multi-task on generalization, which studies the ”one ring that rules all” method of approach - by creating methods that can learn from multiple domains as in Albuquerque et al. (2020); Li et al. (2020); Mahajan et al. (2020); Piratla et al. (2020), while the second flavor focuses on the swiss knife strategy of creating a model that can accomplish different user-defined tasks such as Brown et al. (2020) and Yan et al. (2019). The problem of multi-task learning also exists in reinforcement learning, and is addressed in Hessel et al. (2019), Oh et al. (2017) and a number of papers.

One obstacle in multi-task learning is catastrophic forgetting French (1999), where the model trained on the latest task leads to performance degradation on earlier tasks. Some people address it by co-training the tasks as in Yan et al. (2019), however this does not really apply in reinforcement learning since it is difficult for the simulator to perform 2 tasks at once. Thus the approach of learning one task at a time needs to be used and thus parameter superposition is needed.

Parameter Superposition The method using Parameter Superposition (PSP) to store multiple models simultaneously into one set of parameters actually stems from the fundamental operation existing in all neural networks: to multiply the inputs X by a weight matrix to compute features (y = Wx). In fact, over-parameterization of a network implies that only a small subspace spanned by rows of W in R are relevant to the task.

Based on different context, there are Rotational Superposition, Complex Superposition, and Binary Superposition. Rotational Superposition is the most general way to choose the context is to sample rotations uniformly from orthogonal group; Complex Superposition is where we can chose element to be a vector of complex numbers; Binary Superposition is when constraining the phase to two possible values, which is a special case of complex superposition.

Here, we first consider an investigation in the Cheung et al. (2019), they present a method for storing multiple models within a single set of parameters. Models can coexist in superposition and still be retrieved individually. This approach may be viewed as the online complement of compression: rather than reducing the size of a network after training, They make use of the unrealized capacity of a network during training. After that, we went over the Song et al. (2016) to get a full spectrum of the problem. To address the limitation of computationally intensity and memory intensity of neural network for hardware resources, they introduce ”deep compression”, a three stage pipeline: pruning, trained quantization and Huffman coding, which work together to reduce the storage requirement of neural networks without affecting their accuracy. Lastly, we review the abstract of the latest work Mitchell et al. (2020), they present the Supermasks in Superposition (SupSup) model, which is capable of learning thousands of tasks sequentially without catastrophic forgetting. Their approach is based on a randomly initialized fixed-base network and to find a supermask that achieves good performance in each task.

2 Method

In general, the method we are using is deep reinforcement learning, combined with PSP to increase its capacity for multi-task learning. We divide the description of our method into 2 subsections The first subsection briefly introduces the background method of deep reinforcement learning, and the second subsection explains the method we take to alleviate catastrophic forgetting.

Parameter space: The parameter space deep neural network is the space of parameters of deep neural networks. In deep RL that we are using, the parameter space is the vector space the parameter weights, W and bias, b in each layer of the deep neural network.

Multitask Learning: K is the number of tasks in multi-task learning. and W_K or b_K is the sets of parameters for k different tasks.

2.1 Preliminary: Deep Reinforcement Learning, and Policy-Gradient

Deep reinforcement learning is reinforcement learning empowered by deep neural networks. Deep neural networks can improve the agents’ approximation in value function(Q learning), environment states(Model based RL), and stochastic policies(Policy Gradients). Although used in totally different ways and have different meanings, deep neural networks play an important rule in all types of deep RL. The parameters of the networks decides the policy the agents take with different observations, and therefore determines the performance. The parameters of the networks update in the training from experience of finishing a task. Therefore it is intuitive to understand the parameters as “skills” being learned in deep RL agents.

In multi-task learning, the deep RL agents tend to learn multiple skills in dynamic environments. The most naive way of doing so is trying to learn different skills in the same set of parameters, namely to learn a generic skill. However, it is common that some parameters updated in more recent tasks will compromise the skill learned in earlier ones. We will use parameter superposition to address this issue as follows.

Parameter Superposition in Deep RL The intuition behind the application of Parameter Superposition is to store many models into one set of parameters. Let W1, W2, ..., WK be the sets of parameters required for each of the K tasks. If the network is large and only a small subspace in $< N$ is required by each Wk, it should be possible to transform each Wk using a task specific linear transformation C_k^{-1} (that we call as context), such that rows of each $W_k C_k^{-1}$ occupy mutually orthogonal subspace in $< N$. Because each $W_k C_k^{-1}$ occupies a different subspace, these parameters can be summed together without interfering when stored in superposition:

$$W = \sum_{i=1}^{K} W_i C_i^{-1}$$

When we want to extract he policy parameter specific to task k:

$$\hat{W}_k = W C_K = \sum_{i=1}^{K} W_i (C_i^{-1} C_k)$$

We apply this formulations of parameter superposition in each layer l of neural network:

$$x^{(l+1)} = g(W^{(l)}(c(k)^{(l)} \odot x^{(l)}))$$

where g is the non-linear activation function.

A premise for parameter superposition is over-parameterization. Therefore, ideally we need a large enough neural network with the agent, so that each skill only takes a small portion of superpositioned parameters. We will also test the influence of different level of parameterization in experiments.

2.2 Policy Gradients and Complex Superposition

In our experiments and evaluation. We focused on the policy model and its parameter superposition. Therefore we used the on Policy-Gradients method. We used fully connected layer as the policy model, and update its parameter in training iterations. We used a task identifier in order to choose the context for superposition in both training and evaluation for multi-task learning.

There are multiple types of context selection matrix Ck. In our paper, we focused the Complex Superposition: we can chose C_k to be a vector of complex numbers, where each component $C_k^{(j)}$ is given by

$$C_k^{(j)} = e^{i\phi_j k}$$

Each of the $C_k^{(j)}$ lies on the complex unit circle. The phase $\phi_j(k) \in [\pi, \pi]$ for all j is sampled with uniform probability density $p(\phi) = \frac{1}{2\pi}$. Such a choice of ck results in an orthogonal matrix. A special case of complex superposition is binary superposition, where $C_k^{(j)} \in \{1, 1\}$. The low-precision of the context vectors in this form of superposition has both computational and memory advantages.

The procedure of the algorithm is as follows:

Algorithm 1: Parameter Superposition

```
Tasks: T different tasks;
Policy network: P($s|\theta_T^P$);
Initialize replay buffer R;
N ← $traningbatchsize$ for Task TS in [1,T] do
    select context $\theta_{TS}^P$ for P;
    for path in [1, N do
        for t in [1,episode length] do
            $a_t$ = P($s_t|\theta_{ts}^P$ ;
            execute $a_t$;
            store $(s_t, o_t, r_t, s_t)$ in R ;
        end
    end
    sample random minibatch of M transitions for R;
    gradient ←$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta \left(a_{it} \mid s_{it}\right) \left(\sum_{t'=t}^{T-1} r\left(s_{it'}, a_{it'}\right)\right)$;
    update $P(s_t|\theta_{ts}^P)$ with gradient;
end
```

3 Experiments

For all experiments and figures, please note that we switch the task and environment in training every 150 iterations.

3.1 Experiment 1: Baseline and Simple Parameter Superposition

Here we construct our baseline using the box-close-v1 and button-press-v1 environments. We trained one Policy-Gradient agent on these two tasks one by one, and observed the performance of it as our baseline. The policy network of this agent is a middle-sized fully functional neural network, with no optimization for multi-task learning. We then used the same network plus the parameter superposition(SPS) on each layer. We tested whether it alleviates catastrophic forgetting. Results are shown as follows and we can clearly see that parameter superposition has partially overcome catastrophic learning while the second task continues to learn, as shown in Figure 1.

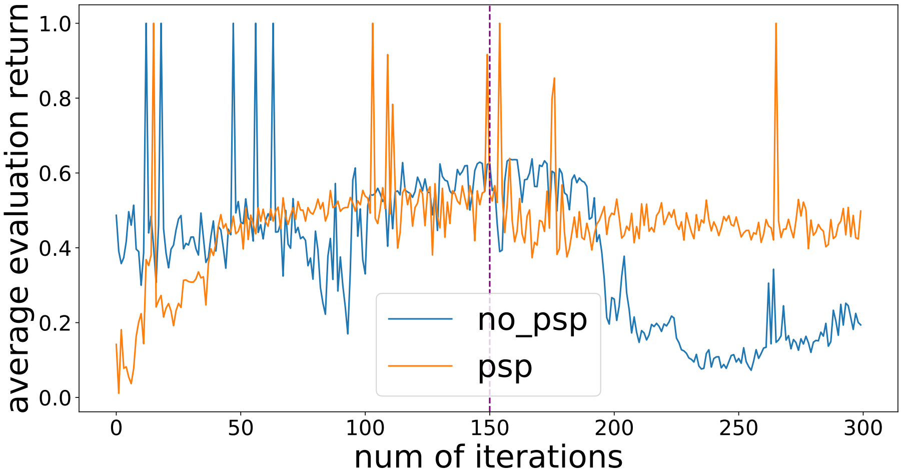

Figure 1: PSP vs non-PSP in the first task, the value we are comparing the normalized reward.

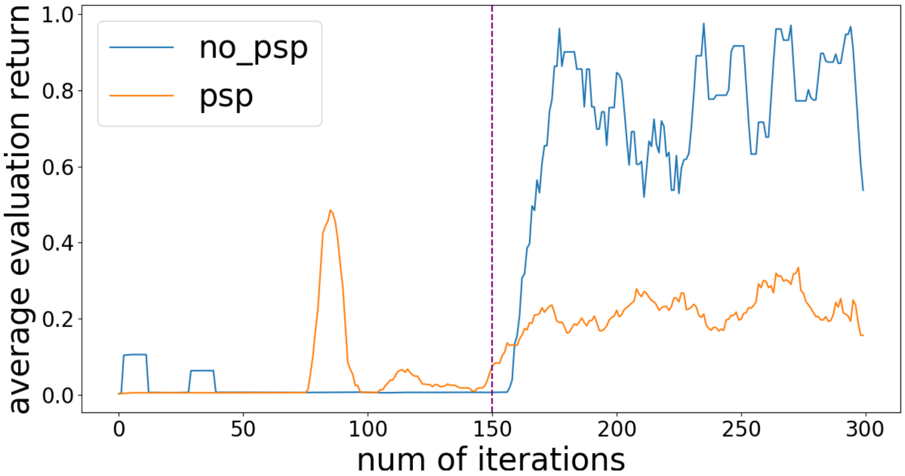

Figure 2: PSP vs non-PSP in the second task, the value we are comparing the normalized reward.

Based on the implementation of baseline and binary superposition, we did a grid search on related parameters – size of training set(batch size), size of network(size and depth), and learning rate. It turns out that larger networks combined with more training steps result in better performance in both learning and keeping the skill of task 1. Results are displayed in Figure 5, 6 and 7.

Besides whether they forget the first skill, closing box, we also care about whether they are able to pick up the new skill, pressing button. In Figure 2, we can see that gradually forgetting the first task, Mlp Policy managed to learn the second task and ran with high average rewards in the second environment. For Psp policy, it gets improvements through the training process, but they are insignificant compared to those of Mlp Policy. Therefore, although PSP can oversome forgetting to some extent, it does not eliminate interference between training of different tasks and is limited by the capacity of superposition(binary PSP) and the capacity of neural networks. We explored these two properties in the following experiments.

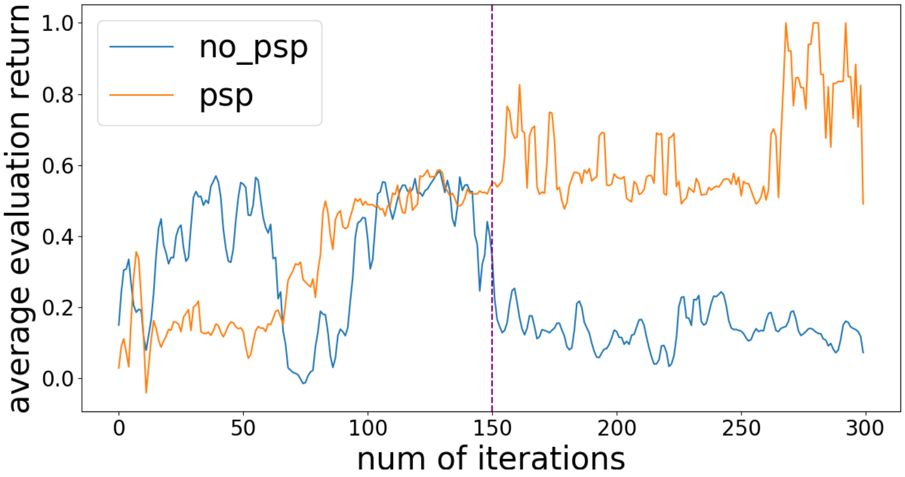

Figure 3: Performances of the first task of MLP policy and PSP Policy with larger policy networks.

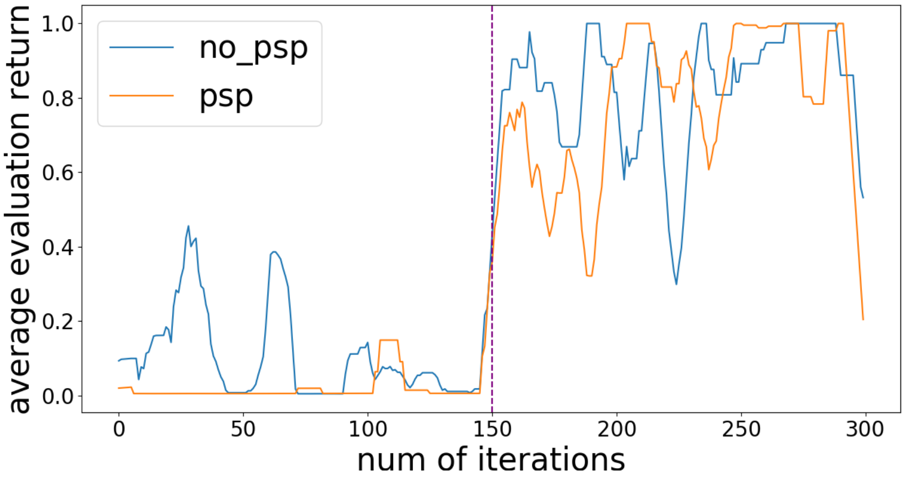

Figure 4: Performances of the first task of MLP policy and PSP Policy with larger policy networks.

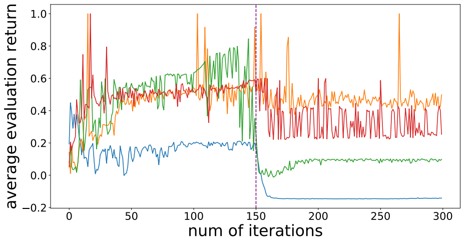

Figure 5: Effects of batch size: Blue(1000), Orange(3000), Green(5000), Red(7000)

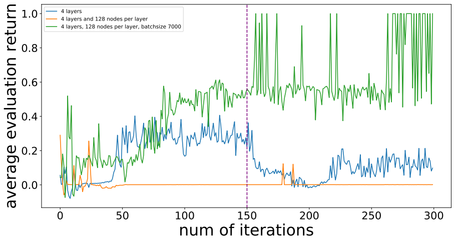

Figure 6: Effect of size of the network: Blue(4 layers), Orange(4 layers, 1280 nodes per layer), Green(4 layers, 1280 nodes per layer, batch size 7000)

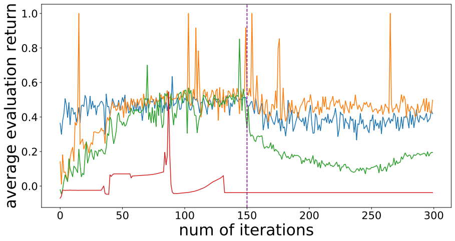

Figure 7: Effect of learning rate: Blue(1e-3), Orange(5e-3), Green(5e-2), Red(5e-2)

3.2 Experiment 2: Improvements on Parameter Superposition

Intuitively, a larger policy neural network has more parameters, and will accordingly have larger capacity for a single task and multiple tasks. We then conducted experiments to explore how larger neural networks would help both MLP policy and PSP policy overcome catastrophic forgetting. Aftering trying with different combinations of parameters, which we will elaborate in the next section, we found a large neural network with 4 layers and size 128 a good candidate for both MLP policy and PSP policy.

In Figure 3, MLP Policy has better performance than in the last experiment. It has the same learning growth in the first 150 iterations on the first task, while has a more smooth curve forgetting it in the second 150 iterations. PSP policy keeps its high performance in task 1, and even better in task 1 while learning task 2. We would still classify this as an interference in that interference would happen in both directions. For picking up task 2, take a look at Figure 4. MLP policy boosted even more in performing task 2. At the same time, PSP policy has a much more significant improvement. Compared to small-network PSP policy, this policy performs twice better, although it's still slightly worse than that of MLP Policy. For this set of experiments, we observe, aligning to what we expected, larger policy networks bring along better capacity of learning and less forgetting, and especially enhances networks with parameter superposition.

3.3 Experiment 3: Superposition on Different Combination of Environments

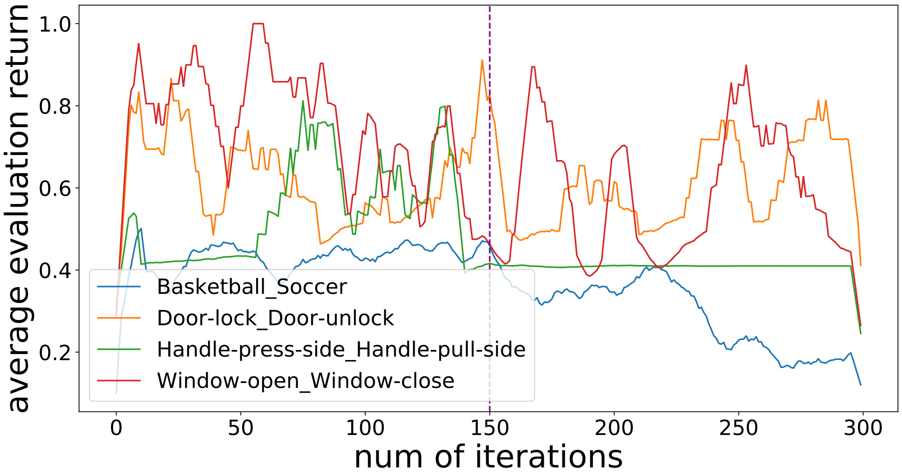

Figure 8: Context: performing different actions on a same object

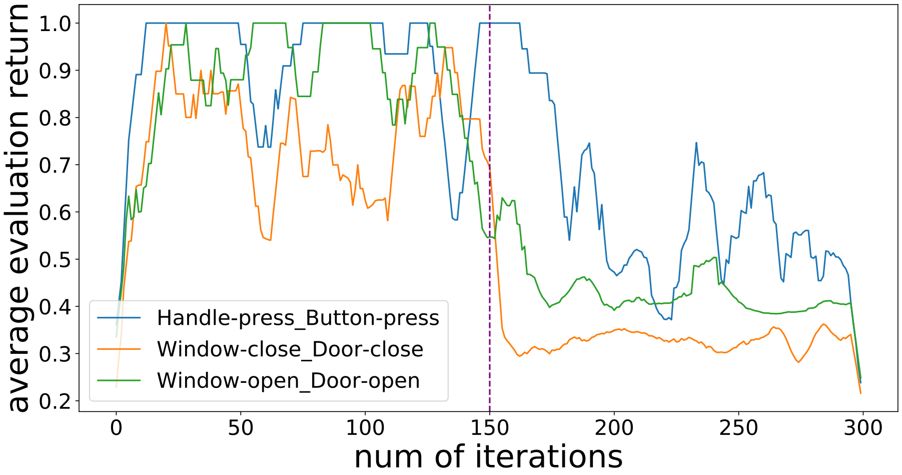

Figure 9: Movement: performing the same actions on different objects

When we are playing the multi-tasking scheme among different combinations of environment, we should expect different performance. Intuitively, similar context (such as the door or the window) and similar action (such as push, pull, and press) should outperform those combination without inherent similarity. Therefore, we conducted three sets of experiment which includes similar context, similar actions, and serial movements in a specific context. The context stands for we are performing

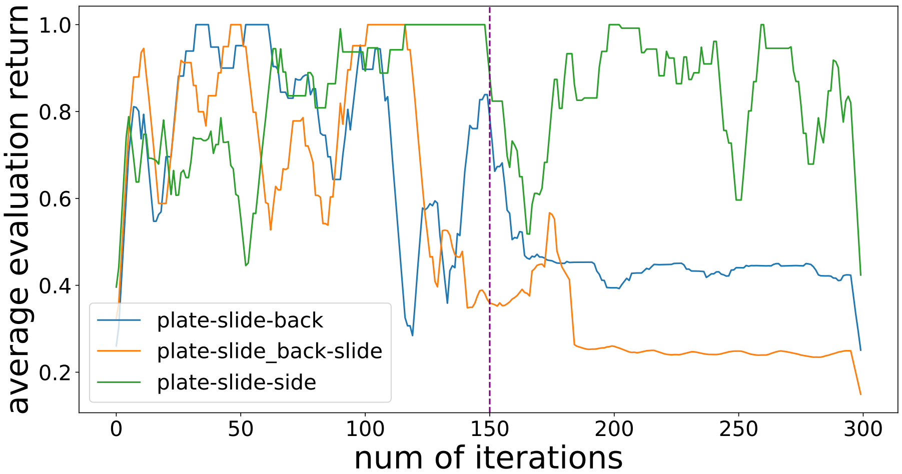

Figure 10: Series: playing in an environment with a series of relevant actions

different actions on a same object, such as a window; the movement means that we are performing the same actions on different objects, such as opening the door and opening the window; the series means that we are playing in an environment with a series of relevant actions, such s slide back, slide side, and slide back side. We need to notice that the first environment in Serial group is plate-slide.

We can see that our experiment fits our intuition. From our experiment on Context, the combination with the best performance is the one of the window, and the one with poorest performance is the one with soccer and basketball. Intuitively, we know that the inherent similarity of soccer and basketball is lower than that of opening the window and closing the window. Also, the soccer and basketball environment is more complex than that of the window or the door. Same as the previous, the best-performing environment in Series is the slide-side, which has the greatest inherent similarity with plate-slide itself, and the worst-performing environment is the plate-slide-back-slide one, which is the most complex one with least inherent similarity with plate-slide. The above performance and evidence shows that our implementation of superposition is successful, and it helps to overcoming the catastrophic forgetting under multi-tasking environment.It is also noticeable that the performance of different groups varies, and it heavily depends on the inherent similarity between the two environments. This is because of the space sparsity of different environments.

4 Conclusion and Future Works

In this paper, we proposed a novel framework to adopt the method of superposition in tackling the problem of catastrophic forgetting under multi-task environment. We model the learning problem as a reinforcement learning problem aided by superposition models.

By conducting experiments in different settings, which includes simple parameter superposition, different policy networks, different combination of environments, and different superposition models, we show that our application of superposition can significantly enhance the capability of sequential learning and improve catastrophic forgetting problem.

We made the following contributions throughout the project.

By adopting our superposition method, the ability to learn in a sequential order of our Deep Q-learning model is improved significantly. We demonstrate the improvement on catastrophic forgetting phenomenon in many different settings.

By observing effectiveness of parameter superposition in different network settings and environments, we explored the correlation of the PSP and sparsity of parameter space and observation space.

We are planning to further dive into this direction in the future, by pursuing the following avenues:

First, we plan to comprehensively investigate the relationship between the parameter sparsity and the effectiveness of parameter superposition. Intuitively, the more sparse the network is, the better parameter superposition will overcome catastrophic forgetting.

Second, we plan to try out more combination of different environments. We can explore the relationship between different actions and their corresponding space sparsity. Furthermore, we can also try to multi-tasking on three or more different environment at the same time. Currently, we are only doing multi-tasking between two environments.

Finally, parameter superposition can potentially be applied to Deep Q-learning, model based learning and many other mechanisms of Deep Reinforcement Learning, and we can study on more extensive applications in the future.

REFERENCES

Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H. Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2020.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Brian Cheung, Alexander Terekhov, Yubei Chen, Pulkit Agrawal, and Bruno Olshausen. Superposition of many models into one. In *Advances in Neural Information Processing Systems*, pp. 10868–10877, 2019.

Atkinson Craig, McCane Brendan, Szymanski Lech, and Robins Anthony. Pseudo-rehearsal: Achieving deep reinforcement learning without catastrophic forgetting. *arXiv preprint arXiv:1812.02464*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3796–3803, 2019.

Kirkpatrick James, Pascanu Razvan, Rabinowitz Neil, Veness Joel, Desjardins Guillaume, Rusu Andrei A, Milan Kieran, Quan John, Ramalho Tiago, Grabska-Barwinska Agnieszka, Hassabis Demis, Clopath Claudia, Kumaran Dharshan, and Hadsell Raia. Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:1612.00796*, 2016.

Serrà Joan, Surís Dídac, Miron Marius, and Karatzoglou Alexandros. Overcoming catastrophic forgetting with hard attention to the task. *arXiv preprint arXiv:1801.01423*, 2018.

Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q Weinberger. On feature normalization and data augmentation. *arXiv preprint arXiv:2002.11102*, 2020.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv preprint arXiv:2006.07500*, 2020.

Wortsman Mitchell, Ramanujan Vivek, Liu Rosanne, Kembhavi Aniruddh, Rastegari Mohammad, Yosinski Jason, and Farhadi Ali. Supermasks in superposition. In *Advances in Neural Information Processing Systems*, 2020.

Junhyuk Oh, Satinder Singh, Honglak Lee, and Pushmeet Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. *arXiv preprint arXiv:1706.05064*, 2017.

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. *arXiv preprint arXiv:2003.12815*, 2020.

Han Song, Mao Huizi, and J. Dally William. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2016.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. *arXiv preprint arXiv:2011.08036*, 2020.

Ke Yan, Youbao Tang, Yifan Peng, Veit Sandfort, Mohammadhadi Bagheri, Zhiyong Lu, and Ronald M Summers. Mulan: Multitask universal lesion analysis network for joint lesion detection, tagging, and segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 194–202. Springer, 2019.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020.