Lightweight Prompt Learning with General Representation for Rehearsal-free Continual Learning

Anonymous Authors

Abstract

Recently, the prompt-based continual learning has become a new state-of-the-art by using small prompts to induce a large pre-trained model toward each target task. However, we figure out that they still suffer from memory problem as the number of prompts should increase if the model learns very many tasks. To improve this limit, inspired by the human hippocampus, we propose Lightweight Prompt Learning with General Representation (LPG), a novel rehearsal-free continual learning method. Throughout the study, we experimentally show our LPG's promising performances and corresponding analyses. We expect our proposition to spotlight a novel continual learning paradigm that utilizes a single prompt to hedge memory problems as well as sustain precise performance.

1 Introduction

Background and Motivation Humans have a sophisticated ability to acquire knowledge from the past, and they continuously learn to solve multiple tasks throughout their lifetime. Although deep neural networks accomplished enormous successes in various tasks [1, 2, 3, 4, 5, 6], they are still insufficient in continual learning, which aims to train a single model under non-stationary data distribution where streams of tasks are sequentially given [7, 8, 9, 10]. One primary challenge to this continual learning is catastrophic forgetting [11, 12], which harshly decreases the model's performance on previously-trained task. But then, why do deep neural networks experience catastrophic forgetting while humans do not suffer? Our study aims to scrutinize this reason and bridge this gap by proposing an effective continual learning model.

The early regularization-based methods [13, 14, 15, 16] isolate the plasticity of the model or dynamically expand the number of neurons to solve novel tasks [17, 18]. From a different viewpoint, another stream of work leveraged a rehearsal buffer that stores several past data and re-trains the model to sustain inductive biases on the past tasks [9, 19, 20]. While they accomplish precise continual learning performances, they were not adequate in the real world. These approaches cannot be established when storing past samples is not allowed (i.e., privacy concerns), and the performance drastically decreases under complex datasets or limited memory size. Several works recently suggested prompt-based approaches as rehearsal-free continual learning, which became novel state-of-the-art[21, 8]. A prompt is a small learnable model that instructs a pre-trained model to effectively reuse its representations on the novel task rather than fine-tuning. Instead of storing past data in the rehearsal buffer, these methods store these prompts in a prompt pool. Under the frozen pre-trained model, these methods train the prompts to dynamically instruct the pre-trained model to solve novel tasks. Following the Complementary Learning System theory [22, 23, 24], we regard this prompt-based paradigm as precisely imitating the human brain. A large pre-trained model behaves like a neocortex, which acquires long-term memories and generalization ability, and prompts act similar to the hippocampus that fastly learns the specific experience. However, we hereby discover one major drawback of prior

Submitted to the 6th Workshop on Meta-Learning at NeurIPS 2022, New Orleans. Do not distribute.

prompt-based approaches. As they store many prompts (i.e., more than 10) in the prompt pool, the size of the prompt pool should also become larger if the number of tasks increases or the task becomes more complex. We presume this poses a computation overhead in the real world.

Main Idea and Its Novelty Inspired by the human hippocampus, an organ that consists of many neurons, we seek to improve this memory issue. The human hippocampus is a single organ located in each left and right temporal lobe. Therefore, if we aim to imitate the human brain, we hypothesize that one well-designed prompt (which acts like the hippocampus) can sufficiently alternate many prompts in the prior approaches. Furthermore, in the hippocampus, only a small number of neurons are activated for a specific experience. Therefore, we presume not every neuron is necessary for solving novel tasks; instead, some neurons can be re-used to solve novel tasks if given share similar representation to the past task. To this end, we propose Lightweight Prompt learning with General representation (LPG), a novel, lightweight prompt-based continual learning method. Our main idea is re-using particular neurons at a single prompt if a given task shares a similar representation with the task seen a priori. We experimentally show our idea works well with various task streams consisting of public benchmarks and provide corresponding analyses. We hereby highlight our proposition is novel, because this is the first attempt to utilize a single prompt instead of many prompts. Moreover, our work firstly establishes prompt-based continual learning by re-using particular neurons.

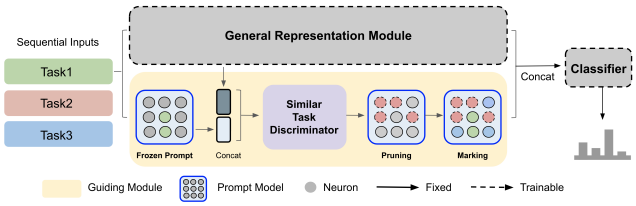

Figure 1: Overview of the proposed LPG

2 Our Approach: Lightweight Prompt learning with General representation

Our LPG consists of two modules: 1) General Representation Module and 2) Guiding Module. Given a task stream, LPG passes the data into each module and concatenates the output representations to make a final prediction. The LPG trains both global representation module and guiding module simultaneously at each task. We hereby highlight that our LPG requires a fixed small memory for serving the general representation module and guiding module, and it barely increases along with the task number. Please refer to the supplementary materials for training details.

2.1 General Representation Module

The global representation module is a pre-trained model that already acquires a general understanding of the task stream. This module aims to provide a general representation of a given data. We presume this representation cannot perfectly describe discriminative cues for a given task but provide a sufficient baseline understanding of it. This representation finally becomes effective when it is concatenated with task-specific representations yielded by the guiding module. In our LPG, we utilize an ImageNet pre-trained ResNet-50[25] for this module as ImageNet-trained weight contributed enormous successes in various computer vision tasks [26]. While previously-proposed prompt-based method requires transformer-architecture for encoding representations on a given task, we hereby highlight our LPG's benefit; Any deep neural network models (i.e., Convolutional Neural Networks or Vision Transformers [27]) can be widely utilized as our method does require particular architecture.

2.2 Guiding Module

Prompt Model The prompt model aims to induce the pre-trained model to understand task-specific knowledge. When the first task is given (we say it is *Task 1*), it simply fine-tunes the prompt with the given data. Then, we randomly select some portion of the neurons, prune the other neurons, and

trains the selected neurons with the given data again. At this stage, our guiding module records which neurons were activated for *Task 1*. When the next task (*Task 2*) is provided, the prompt considers the result of a similar task discriminator (and this discriminator's details are illustrated below section). If the discriminator says the *Task 2* looks similar to *Task 1*, then the prompt retrieves selected neurons at *Task 1*, prunes the others and trains the *Task 2*'s data on the selected neurons. On the other hand, if the discriminator decides the *Task 2* does not share any representation with *Task 2*, the prompt selects the other neurons except for the ones chosen for *Task 1*, prunes the others, and fine-tunes the selected neurons with *Task 2*'s data. Note that neurons selected for *Task 2* are also recorded in small memory space. For the upcoming tasks (*Task* $N > 3$), it iterates the aforementioned procedures. To this end, we alternate past continual learning approaches' multiple prompts or rehearsal buffers to reduce memory consumption. Instead, we let particular neurons be activated for each task, and new neurons are only activated when unseen representations is provided as a novel task.

Similar Task Discriminator Given a novel task, the similar task discriminator aims to classify whether a given data can share similar representation with the past task. The discriminator sends given data to extract representations from the frozen pre-trained model and frozen prompt. Then, it concatenates yielded representations and calculates variance among them. As the guiding module records which neurons are activated for which task, the discriminator examines whether each past task shares a similar representation with the given one. Suppose *Task 1, 2* are already recorded. In this case, the discriminator selects corresponding neurons for *Task 1, 2*, prunes the other neurons, and concatenates the representation with the one yielded from the general representation module. We presume the variance becomes small when these models particularly understand the given data, and this understanding would have stemmed from the past tasks. If the variance from every task exceeds the threshold, we regard the given task as a novel one. Conversely, we regard the given task as a similar one if its variance is lower than the threshold and the other past tasks' variances. Note that we empirically set this preset threshold as 10.05.

3 Experiment

Experimental Setups We implement two scenarios: 1) coarse-grained scenarios and 2) fine-grained scenarios. The coarse-grained scenario is a continual learning task where labels at each task do not share many similar characteristics. We configure the coarse-grained scenario with four public benchmark datasets sequentially: CIFAR-10, CIFAR-100 [28], STL-10 [29], and SVHN [30]. The fine-grained scenario is another continual learning task where discriminative cues among each task are particularly similar. We split labels at the Stanford-Cars dataset [31] (which is conventionally utilized for fine-grained classification) into four tasks. From *Task 1* to *Task 3*, it has 50 labels, and *Task 4* has 46 labels. For evaluation metrics, we measure the classification accuracy on each task after we finish training every task sequentially. For a comparative study, we compare our LPG's performance with task-specific supervised classification, which is a conventional fine-tuning. To implement this supervised classification, we use ImageNet-trained ResNet-50 as a model. Moreover, we additionally compare the results with two classical continual learning methods: PackNet [17] and LWF [13]. Note that we acknowledge the LPG should be compared with state-of-the-art prompt-based methods, but we could not perform it due to computational limitations. We leave this as an improvement avenue.

LPG's Effectiveness and Ablations We then examine whether our LPG accomplishes promising continual learning performances compared to the baselines. We perform experiments on coarse-grained and fine-grained scenarios and report the results in Table 1. Following the result, we discover that LPG successfully outperforms the baselines in most tasks. Based on these results, we urge that using a single prompt is conceptually sufficient to solve continual learning problems; thus, we can utilize much lightweight prompt-based continual learning in the real world.

Furthermore, to scrutinize which module contributes to this precise performance, we perform ablations. First, we implement the LPG without training the general representation module to examine whether the pre-trained model had better stay in a frozen status (denoted as *w/o Update*). Second, we also implement the LPG without updating pre-trained model as well as deactivated similar task

discriminator (denoted as *w/o Update and Disc*. Without a similar task discriminator, a different set of neurons are independently used for each task. Following the results shown in Table 1, we figure out that both updating the pre-trained model and similar task discriminator is essential, but its impact differs along with how much shared representation exists across each task. In the coarse-grained scenario, where STL-10 shows similar representations to CIFAR-10, these two components become significant as ablated LPG achieves decreased performance. We analyze this phenomenon happens because two modules support utilizing shared representations among given tasks; thus, under the coarse-grained scenario where shared knowledge exists, these components become much more effective rather than the other.

Table 1: Experiment results on the effectiveness of our LPG and its ablations

Scenario		PackNet	LWF	Supervised	OURS	w/o Update	w/o Update&Disc	Predicted Similar Task
Coarse-grained	CIFAR-10	89.82	91.29	94.80	**97.56**	96.61	96.56	-
	CIFAR-100	83.09	82.25	89.23	**82.97**	82.43	78.32	N/A
	STL-10	92.81	92.78	96.30	**94.86**	93.56	90.82	CIFAR-10
	SVHN	89.45	88.28	94.05	**95.52**	93.50	94.23	N/A
Fine-grained	Stanford Cars1	42.89	**69.74**	74.89	58.22	60.98	58.93	-
	Stanford Cars2	20.10	28.65	75.22	**44.21**	44.11	43.32	N/A
	Stanford Cars3	41.90	26.86	76.44	**42.75**	41.01	42.08	N/A
	Stanford Cars4	63.45	35.07	75.83	**65.23**	46.05	54.90	N/A

Effectiveness of Similar Task Discriminator Lastly, we aim to scrutinize whether our similar task discriminator indeed contributes to selecting adequate neurons at the prompt module. First, we examine how the variance differs between the data which shares similar representations with the past task and the one that does not share. We set a frozen CIFAR-10 trained model and provide test samples at CIFAR-100, STL, and SVHN. Note that we perform uniform sampling at each label of the corresponding dataset. We visualize the concatenated representations (utilized at the discriminator) with t-sne [32] to analyze how the CIFAR-10 model interprets these unseen data. Following the results in Figure 2, we discover the samples in STL-10 (which was classified to have similar representations with CIFAR-10) show less variance, and the CIFAR-10 model was able to establish decision boundaries between STL-10's labels. Conversely, on the other datasets, the CIFAR-10 model fails to understand their labels and yields high variances. Therefore, we conclude that the similar task discriminator indeed poses low variances to the task where it can re-use the representations learned a priori.

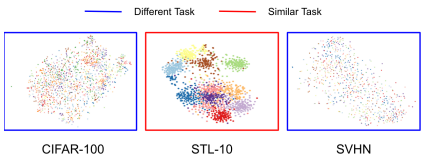

Figure 2: Visualized t-sne results

Table 2: Experiment results for our neuron selection strategy's effectiveness

Scenario	Coarse-grained				Fine-grained			
	CIFAR-10	CIFAR-100	STL-10	SVHN	Stanford Cars1	Stanford Cars2	Stanford Cars3	Stanford Cars4
Random Marking	96.56	70.59	94.29	90.67	58.22	41.93	7.99	47.45
LPG (OURS)	96.56	82.27	94.86	95.52	58.22	44.21	42.75	65.23

To take one step further, we compare the original LPG with the one that randomly select target neurons to train the prompt at each task. Following the results shown in Table 2, we figure out that our neuron selecting strategy has a tangible impact on continual learning performances. Especially in a fine-grained scenario, our neuron selection strategy seems to prevent catastrophic forgetting to many extents. We interpret particular neurons to encode essential information regarding the task, and careless training of whole neurons definitely causes catastrophic forgetting.

4 Discussions and Conclusion

Throughout the work, we propose LPG, a novel rehearsal-free continual learning method, and perform a series of analyses that supports its effectiveness. While we show the LPG's promising performances compared to the baselines, still, it requires more in-depth analyses. First, the LPG should be compared with recent state-of-the-art methods [21] and regularization-based or rehearsal-based approaches under various public benchmarks. Moreover, more experimental questions still exist as follows. What if we change the size of the pre-trained model into various architectures? What if we change the size of the prompt model? What if we change the threshold for a similar task discriminator? By resolving these questions, we highly expect the machine learning community can design an effective continual learner that precisely imitates the human brain.

References

[1] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[2] Douglas L Medin, Mark W Altom, Stephen M Edelson, and Deborah Freko. Correlated symptoms and simulated medical classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 8(1):37, 1982.

[3] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

[4] Juan Du. Understanding of object detection based on cnn family and yolo. In *Journal of Physics: Conference Series*, volume 1004, page 012029. IOP Publishing, 2018.

[5] Wangzhe Du, Hongyao Shen, Jianzhong Fu, Ge Zhang, and Quan He. Approaches for improvement of the x-ray image defect detection of automobile casting aluminum parts based on deep learning. *NDT & E International*, 107:102144, 2019.

[6] Manu S Pillai, Gopal Chaudhary, Manju Khari, and Rubén González Crespo. Real-time image enhancement for an automatic automobile accident detection through cctv using deep learning. *Soft Computing*, 25(18):11929–11940, 2021.

[7] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020.

[8] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2204.04799*, 2022.

[9] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[10] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

[11] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[12] Prakhar Kaushik, Alex Gain, Adam Kortylewski, and Alan Yuille. Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping. *arXiv preprint arXiv:2102.11343*, 2021.

[13] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[14] Benjamin Maschler, Thi Thu Huong Pham, and Michael Weyrich. Regularization-based continual learning for anomaly detection in discrete manufacturing. *Procedia CIRP*, 104:452–457, 2021.

[15] Timothée Lesort, Andrei Stoian, and David Filliat. Regularization shortcomings for continual learning. *arXiv preprint arXiv:1912.03049*, 2019.

[16] Hyo-Eun Kim, Seungwook Kim, and Jaehwan Lee. Keep and learn: Continual learning by constraining the latent space for knowledge preservation in neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 520–528. Springer, 2018.

[17] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[18] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82, 2018.

[19] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018.

[20] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1131–1140, 2020.

[21] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[22] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

[23] Henry WP Dalgleish, Lloyd E Russell, Adam M Packer, Arnd Roth, Oliver M Gauld, Francesca Greenstreet, Emmett J Thompson, and Michael Häusser. How many neurons are sufficient for perception of cortical activity? *Elife*, 9:e58889, 2020.

[24] Diek W Wheeler, Charise M White, Christopher L Rees, Alexander O Komendantov, David J Hamilton, and Giorgio A Ascoli. Hippocampome. org: a knowledge base of neuron types in the rodent hippocampus. *Elife*, 4, 2015.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

[27] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.

[28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[29] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[31] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[32] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[33] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[34] Zijun Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–2. Ieee, 2018.

[35] Sebastian Bock, Josef Goppold, and Martin Weiß. An improvement of the convergence proof of the adam-optimizer. *arXiv preprint arXiv:1804.10587*, 2018.

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Do Pretrained Global Representation Well Describes Stream of Tasks?

Setup First, we aim to examine whether the pre-trained model (which is the ImageNet-trained model) already understands particular knowledge given tasks. We expect our LPG to become meaningful only if the general representation already understands a given task; thus, we first perform this analysis. To discover an answer, we measure the representation similarity between the pre-trained model and the one fine-tuned on the target task. Suppose the representation does not change a lot after fine-tuning; we regard the pre-trained model as already embracing particular knowledge as it shows additional training on the given tasks does not convey significant knowledge to the model. We utilize Centered Kernel Alignment (CKA)[33] to measure representation similarity between two models. The CKA yields a similarity score between 0 and 1, where 1 means high representation similarity at the particular layers of a different model. We measure representation similarities among residual blocks of ResNet-50 between the ImageNet-trained model and four fine-tuned models (independently trained with CIFAR-10, CIFAR-100, STL-10, and SVHN).

Result Looking at the diagonal area (0,0 to 50,50) of visualized representation similarities, we discover that pre-trained and fine-tuned models do not have many representation differences. To interpret, layer *M* at the pre-trained model embraces similar representations at the layer *M* of the fine-tuned model. Consequentially, we discover that pre-trained general representation already knows the novel task's discriminative cues. Furthermore, one thing to improve the task performance in a continual learning scenario is just guiding this general representation to solve the target task, not just simply training it for every target task (as it causes catastrophic forgetting).

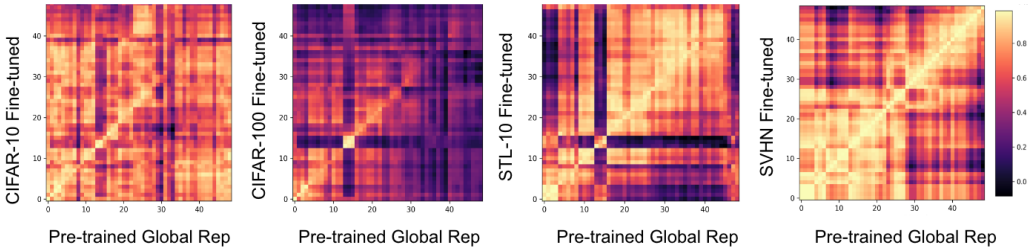

Figure 3: Representation similarities between pre-trained global representation and fine-tuned models. Note that both x, y axis implies layer at ResNet-50.

B Formalized Description on Similar Task Discriminator

We formalize this similar task discriminator in equation 1. Note that Z is the concatenated vector of global and task-specific representation, and n_c is the number of samples in the c label at a given task. The total number of the label is c, and t_c is a threshold.

$$\frac{1}{cn_c}\sum_{i=1}^{c}\sum_{j=1}^{n_c}(Z_{ij} - \frac{1}{n_c}(Z_i))^2 < t_c, (C = 1, 2, .., c) \tag{1}$$

C Training Details

For every experiment in our work, we train the model with cross-entropy loss with Adam Optimizer[34, 35]. We also normalized the given image with the one used at ImageNet training. We set the learning rate with 1e-5 and scheduled with cosine annealing[36], and the batch size was 128. The prompt model is a simple two-layered convolutional neural networks, where each layer has a size of 64. For model training, we used the NVIDIA 2 TESLA P100 provided by the Google Cloud Platform. We publicize the codes in `BlindURL` for reproducibility of our work.