OPT2022: 14th Annual Workshop on Optimization for Machine Learning

Adaptive Methods for Nonconvex Continual Learning

Seungyub Han SEUNGYUBHAN@SNU.AC.KR
Yeongmo Kim MEARK234@SNU.AC.KR
Taehyun Cho TALIUM@SNU.AC.KR
Jungwoo Lee JUNGLEE@SNU.AC.KR
Seoul National University, Republic of Korea

Abstract

One of the objectives of continual learning is to prevent catastrophic forgetting in learning multiple tasks sequentially, and the existing solutions have been driven by the conceptualization of the plasticity-stability dilemma. However, the convergence of continual learning for each sequential task is less studied so far. In this paper, we provide a convergence analysis of memory-based continual learning with stochastic gradient descent and empirical evidence that training current tasks causes the cumulative degradation of previous tasks. We propose an adaptive method for nonconvex continual learning (NCCL), which adjusts step sizes of both previous and current tasks with the gradients. The proposed method can achieve the same convergence rate as the SGD method when the catastrophic forgetting term which we define in the paper is suppressed at each iteration. Further, we demonstrate that the proposed algorithm improves the performance of continual learning over existing methods for several image classification tasks.

1. Introduction

Learning new tasks without forgetting previously learned tasks is a key aspect of artificial intelligence to be as versatile as humans. Unlike the conventional deep learning that observes tasks from an i.i.d. distribution, continual learning train sequentially a model on a non-stationary stream of data [28, 31]. The continual learning AI systems struggle with catastrophic forgetting when the data access of previously learned tasks is restricted [7]. Although novel continual learning methods successfully learn the non-stationary stream sequentially, studies on the theoretical convergence analysis of both previous tasks and a current task have not yet been addressed. In this line of research, nonconvex stochastic optimization problems have been well studied on a single task to train deep neural networks and prove theoretical guarantees of good convergence.

Previous continual learning algorithms have introduced novel methods such as a replay memory to store and replay the previously learned examples [1, 2, 19], regularization methods that penalize neural networks [11, 36], Bayesian methods that utilize the uncertainty of parameters or data points [6, 22], and other recent approaches [15, 34]. The study of continual learning in Bayesian frameworks formulate a trained model for previous tasks parameter into an approximate posterior to learn a probabilistic model which have empirically good performance on entire tasks. However, Bayesian approaches can fail in practice and it can be hard to analyze the rigorous convergence due to the approximation. The memory-based methods are more straightforward approaches, where the learner stores a small subset of the data for previous tasks into a memory and utilizes the memory by replaying samples to keep a model staying in a feasible region without losing the performance on the previous tasks. Gradient episodic memory (GEM) [19] first formulated the replay based continual learning as a constrained optimization problem. This formulation allows us to rephrase the constraints on objectives for previous tasks as inequalities based on the inner product of loss gradient vectors for previous tasks and a current task. However, the gradient update by GEM variants cannot guarantee both theoretical and empirical convergence of its constrained optimization problem. The modified gradient updates do not always satisfy the loss constraint theoretically, and we can also observe the forgetting phenomenon occurs empirically. It also implies that this intuitive reformulation violates the constrained optimization problem and cannot provide theoretical guarantee to prevent catastrophic forgetting without a rigorous convergence analysis.

In this work, we explain the cause of catastrophic forgetting by describing continual learning with a smooth nonconvex finite-sum optimization problem. In the standard single task case, SGD [8], ADAM [26], YOGI [35], SVRG [24], and SCSG [17] are the algorithms for solving nonconvex problems that arise in deep learning. To analyze the convergence of those algorithms, previous works study the following nonconvex finite-sum problem

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x), \tag{1}$$

where we assume that **each objective** $f_i(x)$ **with a model** x **and a data point index** $i \in [n]$ for a dataset with size n (by the convention for notations in nonconvex optimization literature [24]) is nonconvex with L-smoothness assumption. In general, we denote $f_i(x)$ as $f(x; d_i)$ where d_i is a datapoint tuple (INPUT, OUTPUT) with index i. We expect that a stochastic gradient descent based algorithm reaches a stationary point instead of the global minimum in nonconvex optimization. Unlike the convex case, the convergence is generally measured by the expectation of the squared norm of a gradient $\mathbb{E}\|\nabla f(x)\|^2$. The theoretical computational complexity is derived from the ϵ-accurate solution, which is also known as a stationary point with $\mathbb{E}\|\nabla f(x)\|^2 \leq \epsilon$. The general nonconvex finite-sum problems assume that all data points can be sampled during training iterations. This fact is an obstacle to directly apply (1) for continual learning problem.

We provide a solution of the above issue by leveraging memory-based methods, which allow models to access a partial access to the dataset of previous tasks. In this setting, we can analyze nonconvex stochastic optimization problems on the convergence of previous tasks with limited access. Similar with adaptive methods for noncovex optimization, we apply adaptive step sizes during optimization to minimize forgetting with theoretical guarantee.

2. Preliminaries

Suppose that we observe the learning procedure on a data stream of continual learning at some arbitrary observation point. Let us consider time step $t = 0$ as given observation point. We define the previous task $\mathcal{P}$ for $t < 0$ as all visited data points and the current task $\mathcal{C}$ for $t \geq 0$ as all data points which will face in the future. Then, P and C can be defined as the sets of data points in $\mathcal{P}$ and $\mathcal{C}$ at time step $t = 0$, respectively. Note that the above task description is based on not a sequence of multiple tasks, but two separate sets to analyze the convergence of each of P and C when starting to update the given batch at the current task $\mathcal{C}$ at some arbitrary observation point. For clarity, we use $f(x) = h(x)|_P$ and $g(x) = h(x)|_C$ for the restriction of h to each dataset P and C, respectively. $f_i(x)$ and $g_j(x)$ also denotes the objective terms induced from data where each index is $i \in P$ and $j \in C$, respectively. We consider a continual learning problem as a smooth nonconvex finite-sum

Algorithm 1: Nonconvex Continual Learning (NCCL)

Data: previous task set P, current task set C, initial model x^0
Sample a initial memory $M_0 \subset P$ // By replay schemes, the selection dist. of M_0 are different;
for $t = 0$ *to* $T-1$ **do**
  sample a mini-batch $I_t \subset M_t$;
  sample a mini-batch $J_t \subset C$;
  compute step sizes $\alpha_{H_t}, \beta_{H_t}$ by $\nabla f_{I_t}(x^t)$, $\nabla g_{J_t}(x^t)$;
  $x^{t+1} \leftarrow x^t - \alpha_{H_t}\nabla f_{I_t}(x^t) - \beta_{H_t}\nabla g_{J_t}(x^t)$;
  update M_{t+1} by the rule of replay scheme with J_t;
end

optimization problem with two decomposed objectives

$$\min_{x\in\mathbb{R}^d} h(x) = \frac{1}{n_f+n_g}\sum_{i\in P\cup C} h_i(x) = \frac{n_f}{n_f+n_g}\left(\frac{1}{n_f}\sum_{i\in P} f_i(x)\right) + \frac{n_g}{n_f+n_g}\left(\frac{1}{n_g}\sum_{j\in C} g_j(x)\right) \tag{2}$$

where n_f and n_g are the numbers of elements for P and C.

Suppose that the replay memories M_t for time step $\in [0, T]$ are random variables which are the subsets of $P \cup C$ to cover prior memory-based approaches [2, 3]. To formulate an algorithm for memory-based approaches, we define mini-batches I_t which are sampled from a memory M_t at step t. We now define the stochastic update of memory-based method

$$x^{t+1} = x^t - \alpha_{H_t}\nabla f_{I_t}(x^t) - \beta_{H_t}\nabla g_{J_t}(x^t), \tag{3}$$

where $I_t \subset M_t$ and $J_t \subset C$ denote the mini-batches from the replay memory and the current data stream, respectively. Here, H_t is the union of I_t and J_t. In addition, for a given set S, $\nabla f_S(x^t), \nabla g_S(x^t)$ denote the loss gradient of a model x^t with the mini-batch S at time step t The adaptive step sizes (learning rates) of $\nabla f_{I_t}(x^t)$ and $\nabla g_{J_t}(x^t)$ are denoted by α_{H_t} and β_{H_t} which are the functions of H_t. It should be noted the mini-batch I_t from M_t might contain a datapoint $j \in C$ for some cases, such as ER-Reservoir. In appendix, we provide assumptions used in the proof.

3. Continual Learning as Nonconvex Optimization

We present a theoretical convergence analysis of memory-based continual learning in nonconvex setting. We aim to understand why catastrophic forgetting occurs in terms of the convergence rate, and reformulate the optimization problem of continual learning into a nonconvex setting with theoretical guarantee. For completeness we present all proofs in Appendix D. The amount of effect on convergence by a single update can be measured by using Assumption 4. By Equation 7, we have

$$\begin{aligned} f(x^{t+1}) &\leq f(x^t) + \langle \nabla f(x^t), x^{t+1} - x^t\rangle + \frac{L}{2}\|x^{t+1}-x^t\|^2 \\ &= f(x^t) - \langle \nabla f(x^t), \alpha_{H_t}\nabla f_{I_t}(x^t) + \beta_{H_t}\nabla g_{J_t}(x^t)\rangle + \frac{L}{2}\|\alpha_{H_t}\nabla f_{I_t}(x^t) + \beta_{H_t}\nabla g_{J_t}(x^t)\|^2 \end{aligned}$$

by letting $x \leftarrow x^{t+1}$ and $y \leftarrow x^t$. We now propose two terms of interest in a gradient update of nonconvex continual learning (NCCL). We define the overfitting term B_t and the catastrophic forgetting term Γ_t as follows:

$$B_t = (L\alpha_{H_t}^2 - \alpha_{H_t})\langle \nabla f(x^t), e_t \rangle + \beta_{H_t}\langle \nabla g_{J_t}(x^t), e_t \rangle,$$
$$\Gamma_t = \frac{\beta_{H_t}^2 L}{2}\|\nabla g_{J_t}(x^t)\|^2 - \beta_{H_t}(1 - \alpha_{H_t}L)\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle.$$

These two terms essentially measure the performance degradation in NCCL with respect to time. It should be noted that Γ_t has $\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle$, which is a key factor to determine interference and transfer [27]. On the other hand, B_t includes e_t, which is an error gradient between the batch from M_t and the entire dataset P. We now describe a convergence analysis of Algorithm 1. We telescope over training iterations for the current task, which leads to obtain the following theorem.

Theorem 1 *Let $\alpha_{H_t} = \alpha = \frac{c}{\sqrt{T}}$ for some $0 < c \leq \frac{2\sqrt{T}}{L}$ and $t \in \{0, \cdots, T-1\}$. By Lemma 7, the iterates of NCCL satisfy*

$$\min_t \mathbb{E}\|\nabla f(x^t)\|^2 \leq \frac{A}{\sqrt{T}}\left(\frac{1}{c}\left(\Delta_f + \sum_{t=0}^{T-1}\mathbb{E}\left[\Gamma_t\right]\right) + \frac{Lc}{2}\sigma_f^2\right), \tag{4}$$

where $A = 1/(1 - L\alpha/2)$.

One key observation is that $\mathbb{E}[\Gamma_t]$ are cumulatively added on the upper bound of $\mathbb{E}\|\nabla f(x)\|^2$. The loss gap Δ_f and the variance of gradients σ_f are fixed values. In practice, tightening $\sum_t \mathbb{E}[\Gamma_t]$ appears to be critical for the performance of NCCL. However, $\sum_{t=0}^{T-1}\mathbb{E}[\Gamma_t]/\sqrt{T}$ is not guaranteed to converge to 0. This fact gives rise to forgetting in terms of a nondecreasing upper bound.

Lemma 2 *Let an upper bound $\beta > \beta_{H_t} > 0$. Consider two cases, $\beta < \alpha$ and $\beta \geq \alpha$ for α in Theorem 1. We have the following bound*

$$\sum_{t=0}^{T-1}\frac{\mathbb{E}[\Gamma_t]}{\sqrt{T}} < O\left(1/T^{3/2} + 1/T\right) \text{ when } \beta < \alpha, \quad \sum_{t=0}^{T-1}\frac{\mathbb{E}[\Gamma_t]}{\sqrt{T}} < O\left(\sqrt{T} + 1/\sqrt{T}\right), \text{ when } \beta \geq \alpha.$$

Corollary 3 *For $\beta_{H_t} < \alpha = \frac{c}{\sqrt{T}}$ for all t, the IFO complexity of Algorithm 1 for $f(x)$ to obtain an ϵ-accurate solution is as follows:*

$$\textit{IFO calls} = O(1/\epsilon^2). \tag{5}$$

To prevent forgetting, β_{H_t} should be lower than the step size of $f(x)$, α_{H_t}. We provide more details in Appendix. It should also be noted that $\mathbb{E}_{M_{[1:t]}}[B_t|M_0]$ is not always 0 for any M_0. This implies that, from time step 0, each trial with different given M_0 also has the non-zero cumulative sum $\sum \mathbb{E}_{M_{[1:T]}}[B_t|M_0]$, which occurs overestimating bias. We reformulate the constraint optimization problem for CL [19] into the problem on $\mathbb{E}[\Gamma_t]$ in Appendix D.3. We note that $\mathbb{E}[\Gamma_t]$ is a quadratic polynomial of β_{H_t} where $\beta_{H_t} > 0$. The analytic minimum on β_{H_t} can be obtained when

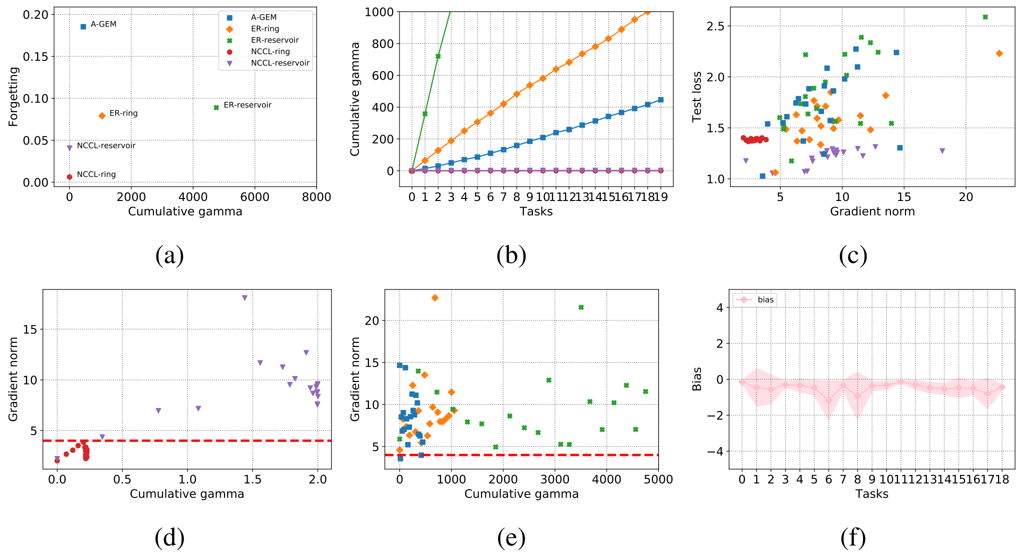

Figure 1: Metrics on split-CIFAR100 with 5 seeds. (a) Forgetting versus $\sum \mathbb{E}[\Gamma_t]$ at the end of training. (b) Evolution of $\sum \mathbb{E}[\Gamma_t]$. (c) Empirical verification of the relation between $\|\nabla f(x)\|$ for the first task and test loss of the first task. (d)-(e) are the empirical verification of $\sum \mathbb{E}[\Gamma_t]$ versus $\|\nabla f(x)\|$ for the first task in CL algorithms. The red horizontal line indicates the empirical $\|\nabla f(x)\|$ right after training the first task. (f) Illustration of empirical B_t at the end of each task.

$\Lambda_{H_t} = \langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle > 0$ by differentiating on β_{H_t}. Then the minimum $\mathbb{E}[\Gamma_t^*]$ and the optimal $\beta_{H_t}^*$ can be obtained as

$$\beta_{H_t}^* = \frac{(1-\alpha_{H_t}L)\Lambda_{H_t}}{L\|\nabla g_{J_t}(x^t)\|^2}, \quad \mathbb{E}[\Gamma_t^*] = -\frac{(1-\alpha_{H_t}L)\Lambda_{H_t}}{2L\|\nabla g_{J_t}(x^t)\|^2}.$$

To satisfy the constraints of (9) in Appendix, we should update $\nabla f_{I_t}(x^t)$ with non-zero step size and $\beta_{H_t} < \alpha_{H_t}$ for all t. Then the proposed adaptive method is given by

$$\alpha_{H_t} = \begin{cases} \alpha(1 - \frac{\Lambda_{H_t}}{\|\nabla f_{I_t}(x^t)\|^2}), & \Lambda_{H_t} \leq 0 \\ \alpha, & \Lambda_{H_t} > 0, \end{cases} \quad \beta_{H_t} = \begin{cases} \alpha, & \Lambda_{H_t} \leq 0 \\ \min\left(\alpha(1-\delta), \frac{(1-\alpha L)\Lambda_{H_t}}{L\|\nabla g_{J_t}(x^t)\|^2}\right), & \Lambda_{H_t} > 0 \end{cases}$$

where $\alpha = c/\sqrt{T}$ and δ is some constant $0 < \delta \ll 1$. For $\Lambda_{H_t} \leq 0$, see details in Appendix D.4.

4. Experiments

We use two following metrics to evaluate algorithms. **(1) Average accuracy** is defined as $\frac{1}{T}\sum_{j=1}^{T} a_{T,j}$, where $a_{i,j}$ denotes the test accuracy on task j after training on task i. **(2) Forgetting** is the average maximum forgetting is defined as $\frac{1}{T-1}\sum_{j=1}^{T-1} \max\limits_{l\in[T-1]}(a_{l,j} - a_{T,j})$. Due to limited space, we report the training details and missing results with additional datasets in Appendix. The following table show our main experimental results, which is averaged over 5 runs. We denote the number of example per class per task at the top of each column. Overall, NCCL + memory schemes outperform baseline methods especially in the forgetting metric. Our goal is to demonstrate the usefulness of the adaptive methods to reduce the catastrophic forgetting, and to show empirical evidence for our convergence analysis. We remark that NCCL successfully suppress **forgetting by a large margin** compared to baselines. It is noted that NCCL also outperforms A-GEM, which does not maximize transfer when $\Lambda_{H_t} > 0$ and violates the proposed constraints in (9).

Table 1: Multi-headed split-CIFAR100, reduced size Resnet-18 $n_f = 20$.

Method	**memory size**	**1**		**5**	
	memory	accuracy	forgetting	accuracy	forgetting
EWC	✗	42.7 (1.89)	0.28 (0.03)	42.7 (1.89)	0.28 (0.03)
Fintune	✗	40.4 (2.83)	0.31 (0.02)	40.4 (2.83)	0.31 (0.02)
Stable SGD	✗	59.9 (1.81)	0.08 (0.01)	59.9 (1.81)	0.08 (0.01)
MC-SGD	✗	63.3 (2.21)	0.06 (0.03)	63.3 (2.21)	0.06 (0.03)
A-GEM	✓	50.7 (2.32)	0.19 (0.04)	59.9 (2.64)	0.10 (0.02)
ER-Ring	✓	56.2 (1.93)	0.13 (0.01)	62.6 (1.77)	0.08 (0.02)
ER-Reservoir	✓	46.9 (0.76)	0.21 (0.03)	65.5 (1.99)	0.09 (0.02)
ORTHOG-subspace	✓	58.81 (1.88)	0.12 (0.02)	64.38 (0.95)	0.055 (0.007)
NCCL + Ring	✓	54.63 (0.65)	**0.059 (0.01)**	61.09 (1.47)	**0.02 (0.01)**
NCCL + Reservoir	✓	52.18 (0.48)	0.118 (0.01)	63.68 (0.18)	0.028 (0.009)

We now investigate the proposed terms with regard to memory-based continual learning, $\sum \mathbb{E}[\Gamma_t]$ and B_t. To verify our theoretical analysis, in Figure 1 we show the cumulative catastrophic forgetting term $\sum_t \mathbb{E}[\Gamma_t]$ is the key factor of the convergence of the first task in split-CIFAR100. During continual learning, $\sum_t \mathbb{E}[\Gamma_t]$ increases in all methods of Figure 1b. Figrure 1a, 1d, 1e show that the larger $\sum_t \mathbb{E}[\Gamma_t]$ causes the larger forgetting and $\|\nabla f(x)\|$ for the first task. We can observe that $\|\nabla f(x)\|$ gets larger than 4, which is for the red line, when $\sum_t \mathbb{E}[\Gamma_t]$ becomes larger than 2. We also verify that the theoretical result $\mathbb{E}_t[B_t] = 0$ is valid in Figure 1f. It implies that the empirical results of Lemma 5, which show the effect of B_t on Equation 14. Furthermore, the memory bias helps to tighten the convergence rate of P by having negative values in practice. Even with tiny memory, the estimated B_t has much smaller value than $\mathbb{E}[\Gamma_t]$ as we can observe in Figure 1. For experience replay, we need not to worry about the degradation by memory bias and would like to emphasize that tiny memory can slightly help to keep the convergence on P empirically. We conclude that the overfitting bias term might not be a major factor in degrading the performance of continual learning agent when it is compared to the catastrophic forgetting term Γ_t.

5. Conclusion

In this paper, we have presented a theoretical convergence analysis of continual learning. Our proof shows that a training model can circumvent catastrophic forgetting by suppressing catastrophic forgetting term in terms of the convergence on previous task. We demonstrate theoretically and empirically that adaptive methods with memory schemes show the better performance in terms of forgetting. It is also noted that there exist two factors on the convergence of previous task: catastrophic forgetting and overfitting to memory. Finally, it is expected the proposed nonconvex framework is helpful to analyze the convergence rate of other continual learning algorithms.

Acknowledgement

This work is in part supported by National Research Foundation of Korea (NRF, 2021R1A2C2014504), Artificial Intelligence Innovation Hub (2021-0-02068) grant funded by the Ministry of Science and ICT (MSIT), Center for Applied Research in Artificial Intelligence(CARAI, UD190031RD) grant Funded by Defense Acquisition Program Administration(DAPA), Agency for Defense Development(ADD), INMAC Seoul National University, and BK21-plus.

References

[1] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, pages 11816–11825, 2019.

[2] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[3] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[4] Arslan Chaudhry, Albert Gordo, Puneet K Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. *arXiv preprint arXiv:2002.08165*, 3, 2020.

[5] Arslan Chaudhry, Naeemullah Khan, Puneet Dokania, and Philip Torr. Continual learning in low-rank orthogonal subspaces. *Advances in Neural Information Processing Systems*, 33, 2020.

[6] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[7] Robert M. French and Nick Chater. Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting. *Neural Computation*, 14(7): 1755–1769, 2002.

[8] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[9] 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.

[10] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. *Advances in Neural Information Processing Systems*, 34:29193–29205, 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] Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard. *arXiv preprint arXiv:2006.05188*, 2020.

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

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 312–321, 2019.

[16] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[17] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pages 2348–2358, 2017.

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

[19] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

[20] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020.

[21] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021.

[22] Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[23] 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.

[24] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323, 2016.

[25] Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Stochastic variance reduction for nonconvex optimization. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 314–323, 2016. URL http://proceedings.mlr.press/v48/reddi16.html.

[26] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[27] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2018.

[28] Mark B. Ring. *Continual learning in reinforcement environments*. PhD thesis, University of Texas at Austin, TX, USA, 1995. URL http://d-nb.info/945690320.

[29] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019.

[30] Umut Simsekli, Lingjiong Zhu, Yee Whye Teh, and Mert Gurbuzbalaban. Fractional underdamped langevin dynamics: Retargeting sgd with momentum under heavy-tailed gradient noise. In *International Conference on Machine Learning*, pages 8970–8980. PMLR, 2020.

[31] Sebastian Thrun. A lifelong learning perspective for mobile robot control. In *Intelligent Robots and Systems, Selections of the International Conference on Intelligent Robots and Systems 1994, IROS 94, Munich, Germany, 12-16 September 1994*, pages 201–214, 1994. doi: 10.1016/b978-044482250-5/50015-3. URL https://doi.org/10.1016/b978-044482250-5/50015-3.

[32] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.

[33] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020.

[34] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[35] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in neural information processing systems*, pages 9793–9803, 2018.

[36] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987, 2017.

[37] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33, 2020.

Appendix A. Review of terminology

(Restriction of f) If $f : A \to B$ and if A_0 is a subset of A, then the **restriction of f to A_0** is the function

$$f|_{A_0} : A_0 \to B$$

given by $f|_{A_0}(x) = f(x)$ for $x \in A_0$.

A.1. Summary of notations

Notations	Definitions	Notations	Definitions
x	model parameter	H_t	the union of I_t and J_t
$\mathcal{P}$	previous task	n_f	the number of data points in P
$\mathcal{C}$	current task	n_g	the number of data points in C
P	dataset of $\mathcal{P}$	$\langle \cdot, \cdot \rangle$	inner product
C	dataset of $\mathcal{C}$	L	L-smoothness constant
$h(x)$	mean loss of x on entire datasets	α_{H_t}	adaptive step size for f with H_t
$f(x)$	mean loss of x on P	β_{H_t}	adaptive step size for g with H_t
$g(x)$	mean loss of x on C	M_t	memory at time t
$f_i(x)$	loss of x on a data point $i \in P$	e_t	error of estimate f at time t
$g_j(x)$	loss of x on a data point $j \in C$	e_{M_t}	error of estimate f with M_t
$f_{I_t}(x)$	mini-batch loss of x on a batch I_t	f_{M_t}	mean loss of x with M_t
$g_{J_t}(x)$	mini-batch loss of x on a batch J_t	$M_{[t1:t2]}$	the history of memory from $t1$ to $t2$
I_t	minibatch sampled from P	B_t	memory bias term at t
J_t	minibatch sampled from C	Γ_t	forgetting term at t
$\mathbb{E}_t$	total expectation from 0 to time t	Λ_{H_t}	inner product between ∇f_{I_t} and ∇g_{J_t}

Appendix B. Related Work

Memory-based methods. Early memory-based methods utilize memory by the distillation [18, 23] or the optimization constraint [2, 19]. Especially, A-GEM [2] simplifies the approach for constraint violated update steps as the projected gradient on a reference gradient which ensures that the average memory loss over previous tasks does not increase. Recent works [3, 4, 27] have shown that updating the gradients on memory directly, which is called experience replay, is a light and prominent approach. We focus on convergence of continual learning, but the above methods focus on increasing the empirical performance without theoretical guarantee. Our analysis provides a legitimate theoretical convergence analysis under the standard smooth nonconvex finite-sum optimization problem setting. Further, [12] shows the perfect memory for optimal continual learning is NP-hard by using set-theory, but the quantitative analysis of performance degradation is less studied.

Adaptive step sizes in nonconvex setting. Adaptive step sizes under smooth nonconvex finite-sum optimization problem have been studied on general single task cases [26, 35, 37] recently. [29, 30, 37] have revealed that there exists a heavy-tailed noise in some optimization problems for neural networks, such as attention models, and [37] shows that adaptive methods are helpful to achieve the faster convergence under the heavy-tailed distribution where stochastic gradients are poorly concentrated around the mean. In this work, we treat the continual learning problem where stochastic gradients of previous tasks are considered as the out-of-distribution samples in regard to a current task, and develop adaptive methods which are well-performed in continual learning.

Regularization based methods. EWC has an additional penalization loss that prevent the update of parameters from losing the information of previous tasks. When we update a model with EWC, we have two gradient components from the current task and the penalization loss.

task-specific model components. SupSup learns a separate subnetwork for each task to predict a given data by superimposing all supermasks. It is a novel method to solve catastrophic forgetting with taking advantage of neural networks.

SGD methods without expereince replay. stable SGD [20] and MC-SGD [10] show overall higher performance in terms of average accuracy than the proposed algorithm. For average forgetting, our method has the lowest value, which means that NCCL prevents catastrophic forgetting successfully with achieving the reasonable performance on the current task. We think that our method is focused on reducing catastrophic forgetting as we defined in the reformulated continual learning problem (12), so our method shows the better performance on average forgetting. Otherwise, MC-SGD finds a low-loss paths with mode-connectivity by updating with the proposed regularization loss. This procedure implies that a continual learning model might find a better local minimum point for the new (current) task than NCCL.

For non-memory based methods, the theoretical measure to observe forgetting and convergence during training does not exist. Our theoretical results are the first attempt to analyze the convergence of previous tasks during continual learning procedure. In future work, we can approximate the value of with fisher information for EWC and introduce Bayesian deep learning to analyze the convergence of each subnetworks for each task in the case of SupSup [33].

Appendix C. Additional Experimental Results and Implementation Details

We implement the baselines and the proposed method on Tensorflow 1. For evaluation, we use an NVIDIA 2080ti GPU along with 3.60 GHz Intel i9-9900K CPU and 64 GB RAM.

C.1. Experimental setup

Datasets. We demonstrate the experimental results on standard continual learning benckmarks: **Permuted-MNIST** [11] is a MNIST [14] based dataset, where each task has a fixed permutation of pixels and transform data points by the permutation to make each task distribution unrelated. **Split-MNIST** [36] splits MNIST dataset into five tasks. Each task consists of two classes, for example (1, 7), (3, 4), and has approximately 12K images. **Split-CIFAR10, 100, and MiniImagenet** also split versions of CIFAR-10, 100 [13], and MiniImagenet [32] into five tasks and 20 tasks.

Baselines. We report the experimental evaluation on the online continual setting which implies a model is trained with a single epoch. We compare with the following continual learning baselines. **Fine-tune** is a simple method that a model trains observed data naively without any support, such as replay memory. **Elastic weight consolidation (EWC)** is a regularization based method by Fisher Information [11]. **ER-Reservoir** chooses samples to store from a data stream with a probability proportional to the number of observed data points. The replay memory returns a random subset of samples at each iteration for experience replay. ER-Reservoir [3] shows a powerful performance in continual learning scenario. **GEM and A-GEM** [2, 19] use gradient episodic memory to overcome forgetting. The key idea of GEM is gradient projection with quadratic programming and A-GEM simplifies this procedure. We also compare with iCarl, MER, ORTHOG-SUBSPACE [5], stable SGD [20], and MC-SGD [21].

C.2. Architecture and Training detail

For fair comparison, we follow the commonly used model architecture and hyperparameters of [5, 16]. For Permuted-MNIST and Split-MNIST, we use fully-connected neural networks with two hidden layers of $[400, 400]$ or $[256, 256]$ and ReLU activation. ResNet-18 with the number of filters $n_f = 64, 20$ [9] is applied for Split CIFAR-10 and 100. All experiments conduct a single-pass over the data stream. It is also called 1 epoch or 0.2 epoch (in the case of split tasks). We deal both cases with and without the task identifiers in the results of split-tasks to compare fairly with baselines. Batch sizes of data stream and memory are both 10. All reported values are the average values of 5 runs with diffrent seeds, and we also provide standard deviation. Other miscellaneous settings are the same as in [5].

C.3. Hyperparameter grids

We report the hyper-paramters grid we used in our experiments below. Except for the proposed algorithm, we adopted the hyper-paramters that are reported in the original papers. We used grid search to find the optimal parameters for each model.

- finetune - learning rate [0.003, 0.01, 0.03 (CIFAR), 0.1 (MNIST), 0.3, 1.0]
- EWC - learning rate: [0.003, 0.01, 0.03 (CIFAR), 0.1 (MNIST), 0.3, 1.0] - regularization: [0.1, 1, 10 (MNIST,CIFAR), 100, 1000]
- A-GEM - learning rate: [0.003, 0.01, 0.03 (CIFAR), 0.1 (MNIST), 0.3, 1.0]
- ER-Ring - learning rate: [0.003, 0.01, 0.03 (CIFAR), 0.1 (MNIST), 0.3, 1.0]
- ORTHOG-SUBSPACE - learning rate: [0.003, 0.01, 0.03, 0.1 (MNIST), 0.2, 0.4 (CIFAR), 1.0]
- MER - learning rate: [0.003, 0.01, 0.03 (MNIST, CIFAR), 0.1, 0.3, 1.0] - within batch meta-learning rate: [0.01, 0.03, 0.1 (MNIST, CIFAR), 0.3, 1.0] - current batch learning rate multiplier: [1, 2, 5 (CIFAR), 10 (MNIST)]
- iid-offline and iid-online - learning rate [0.003, 0.01, 0.03 (CIFAR), 0.1 (MNIST), 0.3, 1.0]
- ER-Reservoir - learning rate: [0.003, 0.01, 0.03, 0.1 (MNIST, CIFAR), 0.3, 1.0]
- NCCL-Ring (default) - learning rate α: [0.003, 0.001(CIFAR), 0.01, 0.03, 0.1, 0.3, 1.0]
- NCCL-Reservoir - learning rate α: [0.003(CIFAR), 0.001, 0.01, 0.03, 0.1, 0.3, 1.0]

C.4. Hyperparameter Search on β_{max} and Training Time

We modify the clipping bound of β_{H_t} in Section 3 to resolve the lower performance in terms of average accuracy. In Table 4, 1, NCCL+Ring does not have the best average accuracy score, even though it has the lowest value of $\sum \mathbb{E}[\Gamma_t]$. As we discussed earlier, it is because the convergence rate of C is slower than vanilla ER-Ring with the fixed step sizes. Now, we remove the restriction of β_{H_t}, $\min\left(\alpha(1-\delta), \frac{(1-\alpha L)\Lambda_{H_t}}{L\|\nabla g_{J_t}(x^t)\|^2}\right)$ for $\Lambda_{H_t} > 0$, and instead apply the maximum clipping bound β_{max} to maximize the transfer effect, which occurs if $\Lambda_{H_t} > 0$, by getting $\mathbb{E}[\Gamma_t^*]$. In the original version, we force $\beta_{H_t} < \alpha$ to reduce theoretical catastrophic forgetting term completely. However, replacing with β_{max} is helpful in terms of average accuracy as shown in Table 2. It means that β_{max} is a hyperparameter to increase the average accuracy by balancing between forgetting on P and learning on C.

Table 2: Permuted-MNIST (23 tasks 10000 examples per task), FC-[256,256] and Multi-headed split-CIFAR100, full size Resnet-18. Accuracies with different clipping rate on NCCL + Ring.

β_{max}	**Permuted-MNIST**	**Split-CIFAR100**
0.001	72.52(0.59)	49.43(0.65)
0.01	72.93(1.38)	56.95(1.02)
0.05	72.18(0.77)	56.35(1.42)
0.1	72.29(1.34)	58.20(0.155)
0.2	74.38(0.89)	57.60(0.36)
0.5	72.95(0.50)	59.06(1.02)
1	72.92(1.07)	57.43(1.33)
5	72.31(1.79)	57.75(0.24)

Table 3: Permuted-MNIST (23 tasks 10000 examples per task), FC-[256,256] and Multi-headed split-CIFAR100, full size Resnet-18. Training time.

Methods	**Training time [s]**	
	Permuted-MNIST	**Split-CIFAR100**
fine-tune	91	92
EWC	95	159
A-GEM	180	760
ER-Ring	109	129
ER-Reservoir	95	113
ORTHOG-SUBSPACE	90	581
NCCL+Ring	167	248
NCCL+Reservoir	168	242

C.5. Additional Experiment Results

We add more results with larger sizes of memory, which shows that NCCL outperforms in terms of average accuracy. It means that estimating transfer and interference in terms of Λ_{H_t} to alleviate forgetting by the small memory for NCCL is less effective.

Table 4: Permuted-MNIST (23 tasks 60000 examples per task), FC-[256,256].

Method	**memory size**	1		5	
	memory	accuracy	forgetting	accuracy	forgetting
multi-task	✗	83	-	83	-
Fine-tune	✗	53.5 (1.46)	0.29 (0.01)	47.9	0.29 (0.01)
EWC	✗	63.1 (1.40)	0.18 (0.01)	63.1 (1.40)	0.18 (0.01)
stable SGD	✗	80.1 (0.51)	0.09 (0.01)	80.1 (0.51)	0.09 (0.01)
MC-SGD	✗	85.3 (0.61)	0.06 (0.01)	85.3 (0.61)	0.06 (0.01)
MER	✓	69.9 (0.40)	0.14 (0.01)	78.3 (0.19)	0.06 (0.01)
A-GEM	✓	62.1 (1.39)	0.21 (0.01)	64.1 (0.74)	0.19 (0.01)
ER-Ring	✓	70.2 (0.56)	0.12 (0.01)	75.8 (0.24)	0.07 (0.01)
ER-Reservoir	✓	68.9 (0.89)	0.15 (0.01)	76.2 (0.38)	0.07 (0.01)
ORHOG-subspace	✓	84.32 (1.10)	0.12 (0.01)	84.32 (1.1)	0.11 (0.01)
NCCL + Ring	✓	74.22 (0.75)	0.13 (0.007)	84.41 (0.32)	0.053 (0.002)
NCCL+Reservoir	✓	79.36 (0.73)	**0.12 (0.007)**	**88.22 (0.26)**	**0.028 (0.003)**

Table 5: Multi-headed split-MiniImagenet, full size Resnet-18 $n_f = 64$. Accuracy and forgetting results.

Method	memory size	1	
	memory	accuracy	forgetting
Fintune	✗	36.1(1.31)	0.24(0.03)
EWC	✗	34.8(2.34)	0.24(0.04)
A-GEM	✓	42.3(1.42)	0.17(0.01)
MER	✓	45.5(1.49)	0.15(0.01)
ER-Ring	✓	49.8(2.92)	0.12(0.01)
ER-Reservoir	✓	44.4(3.22)	0.17(0.02)
ORTHOG-subspace	✓	51.4(1.44)	0.10(0.01)
NCCL + Ring	✓	45.5(0.245)	**0.041(0.01)**
NCCL + Reservoir	✓	41.0(1.02)	**0.09(0.01)**

Table 6: Multi-headed split-CIFAR100, full size Resnet-18 $n_f = 64$. Accuracy and forgetting results.

Method	memory size	1		5	
	memory	accuracy	forgetting	accuracy	forgetting
Fintune	✗	42.6 (2.72)	0.27 (0.02)	42.6 (2.72)	0.27 (0.02)
EWC	✗	43.2 (2.77)	0.26 (0.02)	43.2 (2.77)	0.26 (0.02)
ICRAL	✓	46.4 (1.21)	0.16 (0.01)	-	-
A-GEM	✓	51.3 (3.49)	0.18 (0.03)	60.9 (2.5)	0.11 (0.01)
MER	✓	49.7 (2.97)	0.19 (0.03)	-	-
ER-Ring	✓	59.6 (1.19)	0.14 (0.01)	67.2 (1.72)	0.06 (0.01)
ER-Reservoir	✓	51.5 (2.15)	0.14 (0.09)	62.68 (0.91)	0.06 (0.01)
ORTHOG-subspace	✓	64.3 (0.59)	0.07 (0.01)	67.3 (0.98)	0.05 (0.01)
NCCL + Ring	✓	59.06 (1.02)	0.03 (0.02)	66.58 (0.12)	0.004 (0.003)
NCCL + Reservoir	✓	54.7 (0.91)	0.083 (0.01)	66.37 (0.19)	0.004 (0.001)

Table 7: permuted-MNIST (23 tasks 10000 examples per task), FC-[256,256]. Accuracy and forgetting results.

Method	memory size	1		5	
	memory	accuracy	forgetting	accuracy	forgetting
multi-task	✗	91.3	-	83	-
Fine-tune	✗	50.6 (2.57)	0.29 (0.01)	47.9	0.29 (0.01)
EWC	✗	68.4 (0.76)	0.18 (0.01)	63.1 (1.40)	0.18 (0.01)
MER	✓	78.6 (0.84)	0.15 (0.01)	88.34 (0.26)	0.049 (0.003)
A-GEM	✓	78.3 (0.42)	0.21 (0.01)	64.1 (0.74)	0.19 (0.01)
ER-Ring	✓	79.5 (0.31)	0.12 (0.01)	75.8 (0.24)	0.07 (0.01)
ER-Reservoir	✓	68.9 (0.89)	0.15 (0.01)	76.2 (0.38)	0.07 (0.01)
ORHOG-subspace	✓	86.6 (0.91)	0.04 (0.01)	87.04 (0.43)	0.04 (0.003)
NCCL + Ring	✓	74.38 (0.89)	0.05 (0.009)	83.76 (0.21)	0.014 (0.001)
NCCL+Reservoir	✓	76.48 (0.29)	0.1 (0.002)	86.02 (0.06)	0.013 (0.002)

Table 8: Single-headed split-MNIST, FC-[256,256]. Accuracy and forgetting results.

Method	memory size	1		5		50	
	memory	accuracy	forgetting	accuracy	forgetting	accuracy	forgetting
multi-task	✗	95.2	-	-	-	-	-
Fine-tune	✗	52.52 (5.24)	0.41 (0.06)	-	-	-	-
EWC	✗	56.48 (6.46)	0.31 (0.05)	-	-	-	-
A-GEM	✓	34.04 (7.10)	0.23 (0.11)	33.57 (6.32)	0.18 (0.03)	33.35 (4.52)	0.12 (0.04)
ER-Reservoir	✓	34.63 (6.03)	0.79 (0.07)	63.60 (3.11)	0.42 (0.05)	86.17 (0.99)	0.13 (0.016)
NCCL + Ring	✓	34.64 (3.27)	0.55 (0.03)	61.02 (6.21)	0.207 (0.07)	81.35 (8.24)	-0.03 (0.1)
NCCL+Reservoir	✓	37.02 (0.34)	0.509 (0.009)	65.4 (0.7)	0.16 (0.006)	88.9 (0.28)	-0.125 (0.004)

Table 9: Single-headed split-MNIST, FC-[400,400] and mem. size=500(50 / cls.). Accuracy and forgetting results.

Method	**accuracy**
multi-task	96.18
Fine-tune	50.9 (5.53)
EWC	55.40 (6.29)
A-GEM	26.49 (5.62)
ER-Reservoir	85.1 (1.02)
CN-DPM	93.23
Gdumb	91.9 (0.5)
NCCL + Reservoir	95.15 (0.91)

Table 10: Single-headed split-CIFAR10, full size Resnet-18 and mem. size=500(50 / cls.). Accuracy and forgetting results.

Method	**accuracy**
iid-offline	93.17
iid-online	36.65
Fine-tune	12.68
EWC	53.49 (0.72)
A-GEM	54.28 (3.48)
GSS	33.56
Reservoir Sampling	37.09
CN-DPM	41.78
NCCL + Ring	54.63 (0.76)
NCCL + Reservoir	55.43 (0.32)

Table 11: Single-headed split-CIFAR100, Resnet18 with $n_f = 20$. Memory size = 10,000. We conduct the experiment with the same setting of GMED [10].

Methods	**accuracy**
Finetune	3.06(0.2)
iid online	18.13(0.8)
iid offline	42.00(0.9)
A-GEM	2.40(0.2)
GSS-Greedy	19.53(1.3)
BGD	3.11(0.2)
ER-Reservoir	20.11(1.2)
ER-Reservoir + GMED	20.93(1.6)
MIR	20.02(1.7)
MIR + GMED	21.22(1.0)
NCCL-Reservoir	**21.95(0.3)**

Appendix D. Main Theoretical Results

D.1. Assumptions

Throughout the paper, we assume L-smoothness and the following statements.

Assumption 4 *f_i is L-smooth that there exists a constant $L > 0$ such that for any $x, y \in \mathbb{R}^d$,*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\| \tag{6}$$

where $\|\cdot\|$ denotes the Euclidean norm. Then the following inequality directly holds that

$$-\frac{L}{2}\|x - y\|^2 \leq f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y\rangle \leq \frac{L}{2}\|x - y\|^2. \tag{7}$$

We derive Equation 7 in Appendix D. Assumption 4 is a well-known and useful statement in nonconvex finite-sum optimization problem [24, 26, 35, 37], and also helps us to describe the convergence of continual learning. We also assume the supremum of loss gap between an initial point x^0 and a global optimum x^* as Δ_f, and the upper bound on the variance of the stochastic gradients as σ_f in the following.

$$\Delta_f = \sup_{x^0} f(x^0) - f(x^*),\ \sigma_f^2 = \sup_x \frac{1}{n_f}\sum_{i=1}^{n_f}\|\nabla f_i(x) - \nabla f(x)\|^2.$$

It should be noted that $g_j(x), \nabla g_j(x)$, which denote the loss and the gradient for a current task, also satisfy all three above assumptions and the following statement.

To measure the efficiency of a stochastic gradient algorithm, we define the Incremental First-order Oracle (IFO) framework [8]. IFO call is defined as a unit of computational cost by taking an index i which gets the pair $(\nabla f_i(x), f_i(x))$, and IFO complexity of an algorithm is defined as the summation of IFO calls during optimization. For example, a vanilla stochastic gradient descent (SGD) algorithm requires computational cost as much as the batch size b_t at each step, and the IFO complexity is the sum of batch sizes $\sum_{t=1}^{T} b_t$. Let $T(\epsilon)$ be the minimum number of iterations to guarantee ϵ-accurate solutions. The average bound of IFO complexity is less than or equal to $\sum_{t=1}^{T(\epsilon)} b_t = O(1/\epsilon^2)$ [24].

D.2. Memory-based Nonconvex Continual Learning

Unlike conventional smooth nonconvex finite-sum optimization problems where each mini-batch is i.i.d-sampled from the whole dataset $P \cup C$, the replay memory based continual learning encounters a non-i.i.d stream of data C with access to a small sized memory M_t. Algorithm 1 provides the pseudocode for memory-based approach with the iterative update rule 3. Now, we can analyze the convergence on P and C during a learning procedure on an arbitrary data stream from two consecutive sets P and C for continual learning [2, 3, 5].

By limited access to P, the expectation of gradient update $\mathbb{E}_{I_t \subset M_t}[\nabla f_{I_t}(x^t)]$ in Equation 3 for $f(x)$ is a biased estimate of the gradient $\nabla f(x^t)$. At the timestep t, we have

$$\nabla f_{M_t}(x^t) = \mathbb{E}_{I_t}\left[\nabla f_{I_t}(x^t)|M_t\right] = \mathbb{E}_{I_t}\left[\nabla f(x^t) + e_t|M_t\right] = \nabla f(x^t) + e_{M_t},$$

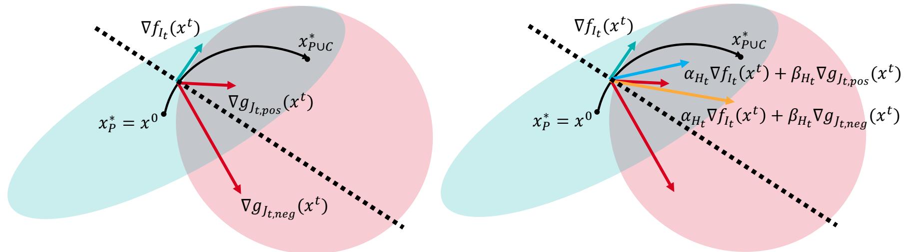

Figure 2: Geometric illustration of Non-Convex Continual Learning (NCCL). In continual learning, a model parameter x^t starts from a local optimal point for the previously learned tasks x_P^*. Over T iterations, we expect to reach a new optimal point $x_{P\cup C}^*$ which has a good performance on both P and C. In the t-th iteration, x^t encounters either $\nabla g_{J_t,pos}(x^t)$ or $\nabla g_{J_t,neg}(x^t)$. These two cases indicate whether $\langle f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle$ is positive or not. To prevent x^t from escaping the feasible region, i.e., catastrophic forgetting, we impose a theoretical condition on learning rates for f and g.

where e_t and e_{M_t} denote the error terms, $\nabla f_{I_t}(x^t) - \nabla f(x^t)$ and the expectation over I_t given M_t, respectively. It should be noted that a given replay memory M_t with small size at timestep t introduces an inevitable overfitting bias.

For example, there exist two popular memory schemes, episodic memory and ER-reservoir. The episodic memory $M_t = M_0$ for all t is uniformly sampled once from a random sequence of P, and ER-reservoir iteratively samples the replay memory M_t by the selection rule $M_t \subset M_{t-1} \cup J_t$. Here, we denote the history of M_t as $M_{[0:t]} = (M_0, \cdots, M_t)$. To compute the expectation over all stochasticities of NCCL, we need to derive the expectation of $\nabla f_{M_t}(x^t)$ over the randomness of M_t. We formalize the expectation over all learning trials with the selection randomness as follows.

Lemma 5 *If M_0 is uniformly sampled from P, then both episodic memory and ER-reservoir satisfies*

$$\mathbb{E}_{M_{[0:t]}}\left[\nabla f_{M_t}(x^t)\right] = \nabla f(x^t) \quad \textit{and} \quad \mathbb{E}_{M_{[0:t]}}\left[e_{M_t}\right] = 0. \tag{8}$$

Note that taking expectation iteratively with respect to the history $M_{[0:t]}$ is needed to compute the expected value of gradients for M_t. Surprisingly, taking the expectation of overfitting error over memory selection gets zero. However, it does not imply $e_t = 0$ for each learning trial with some $M_{[0:t]}$.

D.3. Reformulated Problem of Continual Learning.

The previous section showed that the essential terms in continual learning to observe the theoretical convergence rate. We now reformulate the continual learning problem 2 as follows.

$$\begin{aligned} &\underset{\alpha_{H_t},\, \beta_{H_t}}{\text{minimize}} \quad \sum_{t=0}^{T-1} \mathbb{E}[\Gamma_t] \\ &\text{subject to} \quad 0 < \beta_{H_t} < \alpha_{H_t} \leq 2/L \text{ for all } t < T \end{aligned} \tag{9}$$

It is noted that the above reformulation presents a theoretically guaranteed continual learning framework for memory-based approaches in nonconvex setting.

D.4. Proposed Adaptive Learning Rates

In this section, we derive the proposed adaptive learning rates for both cases $\Lambda_{H_t} \leq 0$ and $\Lambda_{H_t} > 0$.

D.4.1. $\Lambda_{H_t} > 0$

Unlike the exisitnig algorithms, we can leverage the transfer case $\Lambda_{H_t} > 0$ with theoretical guarantee. Then, we propose the optimal learning rates which tighten the upper bound of stationary of $f(x)$ as follows.

Derivation of optimal Γ_t^* and $\beta_{H_t}^*$ For a fixed learning rate α, we have

$$
\begin{aligned}
0 = \frac{\partial \mathbb{E}[\Gamma_t]}{\partial \beta_{H_t}} &= \mathbb{E}\left[\frac{\partial \Gamma_t}{\partial \beta_{H_t}}\right] \\
&= \mathbb{E}\left[\beta_{H_t} L \|\nabla g_{J_t}(x^t)\| - (1-\alpha L)\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle\right].
\end{aligned}
$$

Thus, we obtain

$$
\begin{aligned}
\beta_{H_t}^* &= \frac{(1-\alpha_{H_t} L)\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle}{L\|\nabla g_{J_t}(x^t)\|^2} = \frac{(1-\alpha_{H_t} L)\Lambda_{H_t}}{L\|\nabla g_{J_t}(x^t)\|^2}, \\
\Gamma_t^* &= -\frac{(1-\alpha_{H_t} L)\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle}{2L\|\nabla g_{J_t}(x^t)\|^2} = -\frac{(1-\alpha_{H_t} L)\Lambda_{H_t}}{2L\|\nabla g_{J_t}(x^t)\|^2}.
\end{aligned}
$$

D.4.2. $\Lambda_{H_t} \leq 0$

We adopt the algorithm of A-GEM partially when $\Lambda_{H_t} \leq 0$. In this section, we explain how A-GEM take advantages in terms of our theoretical analysis. A-GEM propose a surrogate of $\nabla g_{J_t}(x^t)$ as the following equation to avoid violating the constraint when $\Lambda_{H_t} \leq 0$ [2]:

$$
\nabla g_{J_t}(x^t) - \left\langle \frac{\nabla f_{I_t}(x^t)}{\|\nabla f_{I_t}(x^t)\|}, \nabla g_{J_t}(x^t)\right\rangle \frac{\nabla f_{I_t}(x^t)}{\|\nabla f_{I_t}(x^t)\|}.
$$

Let β be the step size for $g(x)$ when the constraint is not violated. Then we can interpret the surrogate as adaptive learning rate α_{H_t}, which is either $\alpha(1 - \frac{\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle}{\|\nabla f_{I_t}(x^t)\|^2})$ to cancel out the negative component of $\nabla f_{I_t}(x^t)$ on $\nabla g_{J_t}(x^t)$ or $\alpha_{H_t} = 0$ for $\Lambda_{H_t} > 0$. After applying the surrogate, $\mathbb{E}[\Gamma_t]$ is reduced as shown in the below. It is noted that A-GEM theoretically violates the constraints of (9) to prevent catastrophic forgetting by letting $\alpha_{H_t} = 0$. That is to say, A-GEM is an adaptive method without theoretical guarantee.

Derivation for A-GEM Let the surrogate $\nabla \tilde{g}_{J_t}(x^t)$ as

$$
\nabla \tilde{g}_{J_t}(x^t) = \nabla g_{J_t}(x^t) - \left\langle \frac{\nabla f_{I_t}(x^t)}{\|\nabla f_{I_t}(x^t)\|}, \nabla g_{J_t}(x^t)\right\rangle \frac{\nabla f_{I_t}(x^t)}{\|\nabla f_{I_t}(x^t)\|}, \tag{10}
$$

where $\alpha_{H_t} = \alpha(1 - \frac{\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle}{\|\nabla f_{I_t}(x^t)\|^2})$ and $\beta_{H_t} = \alpha$ for Equation 3.

Then, we have

$$\begin{aligned}
\mathbb{E}[\Gamma_t] &= \mathbb{E}\left[\frac{\beta_{H_t}^2 L}{2}\|\nabla \tilde{g}_{J_t}(x^t)\|^2 - \beta_{H_t}\langle \nabla f_{I_t}(x^t), \nabla \tilde{g}_{J_t}(x^t)\rangle\right] \\
&= \mathbb{E}\left[\frac{\beta_{H_t}^2 L}{2}\left(\|\nabla g_{J_t}(x^t)\|^2 - 2\frac{\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle^2}{\|\nabla f_{I_t}(x^t)\|^2} + \frac{\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle^2}{\|\nabla f_{I_t}(x^t)\|^2}\right) - \beta_{H_t}\langle \nabla f_{I_t}(x^t), \nabla \tilde{g}_{J_t}(x^t)\rangle\right] \\
&= \mathbb{E}\left[\frac{\beta_{H_t}^2 L}{2}\left(\|\nabla g_{J_t}(x^t)\|^2 - \frac{\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle^2}{\|\nabla f_{I_t}(x^t)\|^2}\right) - \beta_{H_t}\left(\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle - \langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle\right)\right] \\
&= \mathbb{E}\left[\frac{\beta_{H_t}^2 L}{2}\left(\|\nabla g_{J_t}(x^t)\|^2 - \frac{\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle^2}{\|\nabla f_{I_t}(x^t)\|^2}\right)\right]. \qquad (11)
\end{aligned}$$

Now, we compare the catastrophic forgetting term between the original value with $\nabla g_{J_t}(x^t)$ and the above surrogate.

$$\mathbb{E}\left[\frac{\beta_{H_t}^2 L}{2}\left(\|\nabla g_{J_t}(x^t)\|^2 - \frac{\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle^2}{\|\nabla f_{I_t}(x^t)\|^2}\right)\right] < \mathbb{E}\left[\frac{\beta_{H_t}^2 L}{2}\|\nabla g_{J_t}(x^t)\|^2 - \beta_{H_t}\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle\right].$$

Then, we can conclude that $\mathbb{E}[\Gamma_t]$ with the surrogate of A-GEM is smaller than the original $\mathbb{E}[\Gamma_t]$.

D.5. Technical Lemma

In the following, we provide the proofs of the results for nonconvex continual learning. We first start with the derivation of Equation 7 in Assumption 4.

Proof [**Derivation of Equation 7**] Recall that

$$|f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y\rangle| \le \frac{L}{2}\|x - y\|^2. \qquad (12)$$

Note that f_i is differentiable and nonconvex. We define a function $g(t) = f_i(y + t(x - y))$ for $t \in [0, 1]$ and an objective function f_i. By the fundamental theorem of calculus,

$$\int_0^1 g'(t)dt = f(x) - f(y). \qquad (13)$$

By the property, we have

$$\begin{aligned}
&|f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y\rangle| \\
&= \left|\int_0^1 \langle \nabla f_i(y + t(x - y)), x - y\rangle dt - \langle \nabla f_i(y), x - y\rangle\right| \\
&= \left|\int_0^1 \langle \nabla f_i(y + t(x - y)) - \nabla f_i(y), x - y\rangle dt\right|.
\end{aligned}$$

Using the Cauchy-Schwartz inequality,

$$\begin{aligned}
&\left|\int_0^1 \langle \nabla f_i(y + t(x - y)) - \nabla f_i(y), x - y\rangle dt\right| \\
&\le \left|\int_0^1 \|\nabla f_i(y + t(x - y)) - \nabla f_i(y)\| \cdot \|x - y\| dt\right|.
\end{aligned}$$

Since f_i satisfies Equation 6, then we have

$$\begin{aligned}
&|f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y \rangle| \\
&\leq \left| \int_0^1 L\|y + t(x - y) - y\| \cdot \|x - y\| dt \right| \\
&= L\|x - y\|^2 \left| \int_0^1 t dt \right| \\
&= \frac{L}{2}\|x - y\|^2.
\end{aligned}$$

■

Lemma 6 *Let $p = [p_1, \cdots p_D],\ q = [q_1, \cdots, q_D]$ be two statistically independent random vectors with dimension D. Then the expectation of the inner product of two random vectors $\mathbb{E}[\langle p, q \rangle]$ is $\sum_{d=1}^{D} \mathbb{E}[p_d]\mathbb{E}[q_d]$.*

Proof By the property of expectation,

$$\begin{aligned}
\mathbb{E}[\langle p, q \rangle] &= \mathbb{E}[\sum_{d=1}^{D} p_d q_d] \\
&= \sum_{d=1}^{D} \mathbb{E}[p_d q_d] \\
&= \sum_{d=1}^{D} \mathbb{E}[p_d]\mathbb{E}[q_d].
\end{aligned}$$

■

D.6. Proof of Main Results

We now show the main results of our work. We first state the stepwise change of upper bound.

Since taking the expectation over all stochasticities of NCCL implies the total expectation, we define the operator of total expectation with respect to $0 \leq t < T$ for ease of exposition as follows:

$$\mathbb{E}_t = \mathbb{E}_{M_{[0:t]}} \left[\mathbb{E}_{I_t} \left[\mathbb{E}_{J_t} \left[\, \cdot \, | I_t \right] \right] | M_{[0:t]} \right].$$

In addition, we denote $\mathbb{E}_{T-1} = \mathbb{E}$.

Lemma 7 *Suppose that Assumption 4 holds and $0 < \alpha_{H_t} \leq \frac{2}{L}$. For x^t updated by Algorithm 1, we have*

$$\mathbb{E}_t \|\nabla f(x^t)\|^2 \leq \mathbb{E}_t \left[\frac{f(x^t) - f(x^{t+1}) + B_t + \Gamma_t}{\alpha_{H_t}(1 - \frac{L}{2}\alpha_{H_t})} \right] + \mathbb{E}_t \left[\frac{\alpha_{H_t} L}{2(1 - \frac{L}{2}\alpha_{H_t})} \sigma_f^2 \right]. \tag{14}$$

Proof [**Proof of Lemma 5**] To clarify the issue of $\mathbb{E}_{M_t}\left[\mathbb{E}_{I_t}\left[e_t|M_t\right]\right]=0$, let us explain the details of constructing replay-memory as follows. We have considered episodic memory and reservoir sampling in the paper. We will first show the case of episodic memory by describing the sampling method for replay memory. We can also derive the case of reservoir sampling by simply applying the result of episodic memory.

Episodic memory (ring buffer). We divide the entire dataset of continual learning into the previous task P and the current task C on the time step $t=0$. For the previous task P, the data stream of P is i.i.d., and its sequence is random on every trial (episode). The trial (episode) implies that a continual learning agent learns from an online data stream with two consecutive data sequences of P and C. Episodic memory takes the last data points of the given memory size m by the First In First Out (FIFO) rule, and holds the entire data points until learning on C is finished. Then, we note that $M_t=M_0$ for all $t\geq 0$ and M_0 is uniformly sampled from the i.i.d. sequence of P. By the law of total expectation, we derive $\mathbb{E}_{M_0\subset P}\left[\mathbb{E}_{I_t}\left[\nabla f_{I_t}(x^t)|M_0\right]\right]$ for any $x^t,\ \forall t\geq 0$.

$$\mathbb{E}_{M_0\subset P}\left[\mathbb{E}_{I_t}\left[\nabla f_{I_t}(x^t)|M_0\right]\right]=\mathbb{E}_{M_0\subset P}\left[\nabla f_{M_0}(x^t)\right].$$

It is known that M_0 was uniformly sampled from P on each trial before training on the current task C. Then, we take expectation with respect to every trial that implies the expected value over the memory distribution M_0. We have

$$\mathbb{E}_{M_0\subset P}\left[\nabla f_{M_0}(x^t)\right]=\nabla f(x^t)$$

for any $x^t,\ \forall t$. We can consider $\nabla f_{M_t}(x^t)$ as a sample mean of P on every trial for any $x^t,\ \forall t\geq 0$. Although x^t is constructed iteratively, the expected value of the sample mean for any x^t, $\mathbb{E}_{M_0\subset P}\left[\nabla f_{M_0}(x^t)\right]$ is also derived as $\nabla f(x^t)$.

Reservoir sampling. To clarify the notation for reservoir sampling first, we denote the expectation with respect to the history of replay memory $M_{[0:t]}=(M_0,\cdots,M_t)$ as $\mathbb{E}_{M_{[0:t]}}$. This is the revised version of $\mathbb{E}_{M_t}$. Reservoir sampling is a trickier case than episodic memory, but $\mathbb{E}_{M_{[0:t]}}\left[\mathbb{E}_{I_t}\left[e_t|M_t\right]\right]=0$ still holds. Suppose that M_0 is full of the data points from P as the episodic memory is sampled and the mini-batch size from C is 1 for simplicity. The reservoir sampling algorithm drops a data point in M_{t-1} and replaces the dropped data point with a data point in the current mini-batch from C with probability $p=m/n$, where m is the memory size and n is the number of visited data points so far. The exact pseudo-code for reservoir sampling is described in [1]. The replacement procedure uniformly chooses the data point which will be dropped. We can also consider the replacement procedure as follows. The memory M_t for P is reduced in size 1 from M_{t-1}, and the replaced data point d_C from C contributes in terms of $\nabla g_{d_C}(x^t)$ if d_C is sampled from the replay memory. Let $M_{t-1}=[d_1,\cdots,d_{|M_{t-1}|}]$ where $|\cdot|$ denotes the cardinality of the memory. The sample mean of M_{t-1} is given as

$$\nabla f_{M_{t-1}}(x^{t-1})=\frac{1}{|M_{t-1}|}\sum_{d_i}\nabla f_{d_i}(x^{t-1}). \tag{15}$$

By the rule of reservoir sampling, we assume that the replacement procedure reduces the memory from M_{t-1} to M_t with size $|M_{t-1}|-1$ and the set of remained upcoming data points $C_t\in C$ from the current data stream for online continual learning is reformulated into $C_{t-1}\cup[d_C]$. Then, d_C can be resampled from $C_{t-1}\cup[d_C]$ to be composed of the minibatch of reservoir sampling with the dfferent probability. However, we ignore the probability issue now to focus on the effect of replay-memory on ∇f. Now, we sample M_t from M_{t-1}, then we get the random vector $\nabla f_{M_t}(x^t)$ as

$$\nabla f_{M_t}(x^t) = \frac{1}{|M_t|} \sum_{j=1}^{|M_{t-1}|} W_{ij} \nabla f_{d_j}(x^t), \tag{16}$$

where the index i is uniformly sampled from $i \sim [1, \cdots, |M_{t-1}|]$, and W_{ij} is the indicator function that W_{ij} is 0 if $i = j$ else 1.

The above description implies the dropping rule, and M_t can be considered as an uniformly sampled set with size $|M_t|$ from M_{t-1}. There could also be $M_t = M_{t-1}$ with probability $1 - p = 1 - m/n$. Then the expectation of $\nabla f_{M_t}(x^t)$ given M_{t-1} is derived as

$$\begin{aligned} \mathbb{E}_{M_t}[\nabla f_{M_t}(x^t)|M_{t-1}] &= p \left(\frac{1}{|M_{t-1}|} \sum_{i}^{|M_{t-1}|} \frac{1}{|M_t|} \sum_{j=1}^{|M_{t-1}|} W_{ij} \nabla f_{d_j}(x^t) \right) + (1-p) \left(\nabla f_{M_{t-1}}(x^t) \right) \\ &= \nabla f_{M_{t-1}}(x^t). \end{aligned}$$

When we consider the mini-batch sampling, we can formally reformulate the above equation as

$$\mathbb{E}_{M_t \sim p(M_t|M_{t-1})} \left[\mathbb{E}_{I_t \subset M_t} \left[\nabla f_{I_t}(x^t) | M_t \right] | M_{t-1} \right] = \nabla f_{M_{t-1}}(x^t). \tag{17}$$

Now, we apply the above equation recursively. Then,

$$\mathbb{E}_{M_1 \sim p(M_1|M_0)} \left[\cdots \mathbb{E}_{M_t \sim p(M_t|M_{t-1})} \left[\mathbb{E}_{I_t \subset M_t} \left[\nabla f_{I_t}(x^t) | M_t \right] | M_{t-1} \right] \cdots | M_0 \right] = \nabla f_{M_0}(x^t). \tag{18}$$

Similar to episodic memory, M_0 is uniformly sampled from P. Therefore, we conclude that

$$\mathbb{E}_{M_0, \cdots, M_t}[\nabla f_{M_t}(x^t)] = \nabla f(x^t) \tag{19}$$

by taking expectation over the history $M_{[0:t]} = (M_1, M_2, \cdots, M_t)$.

Note that taking expectation iteratively with respect to the history $M_{[t]}$ is needed to compute the expected value of gradients for M_t. However, the result $\mathbb{E}_{M_0, \cdots, M_t}[\mathbb{E}_{I_t}[e_t|M_t]] = 0$ still holds in terms of expectation.

Furthermore, we also discuss that the effect of reservoir sampling on the convergence of C. Unlike we simply update $g(x)$ by the stochastic gradient descent on C, the datapoints $d \in M \cap C$ have a little larger sampling probability than other datapoints $d_{C-M} \in C - M$. The expectation of gradient norm on the averaged loss $\mathbb{E}\|\nabla g(x^t)\|^2$ is based on the uniform and equiprobable sampling over C, but the nature of reservoir sampling distort this measure slightly. In this paper, we focus on the convergence of the previous task C while training on the current task C with several existing memory-based methods. Therefore, analyzing the convergence of reservoir sampling method will be a future work.

■

Proof [**Proof of Lemma 7**] We analyze the convergence of nonconvex continual learning with replay memory here. Recall that the gradient update is the following

$$x^{t+1} = x^t - \alpha_{H_t} \nabla f_{I_t}(x^t) - \beta_{H_t} \nabla g_{J_t}(x^t)$$

for all $t \in \{1, 2, \cdots, T\}$. Let $e_t = \nabla f_{I_t}(x^t) - \nabla f(x^t)$. Since we assume that $f,\ g$ is L-smooth, we have the following inequality by applying Equation 7:

$$\begin{aligned}
f(x^{t+1}) &\leq f(x^t) + \langle \nabla f(x^t), x^{t+1} - x^t \rangle + \frac{L}{2} \| x^{t+1} - x^t \|^2 \\
&= f(x^t) - \langle \nabla f(x^t), \alpha_{H_t} \nabla f_{I_t}(x^t) + \beta_{H_t} \nabla g_{J_t}(x^t) \rangle + \frac{L}{2} \| \alpha_{H_t} \nabla f_{I_t}(x^t) + \beta_{H_t} \nabla g_{J_t}(x^t) \|^2 \\
&= f(x^t) - \alpha_{H_t} \langle \nabla f(x^t), \nabla f_{I_t}(x^t) \rangle - \beta_{H_t} \langle \nabla f(x^t), \nabla g_{J_t}(x^t) \rangle \\
&\quad + \frac{L}{2} \alpha_{H_t}^2 \| \nabla f_{I_t}(x^t) \|^2 + \frac{L}{2} \beta_{H_t}^2 \| \nabla g_{J_t}(x^t) \|^2 + L \alpha_{H_t} \beta_{H_t} \langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t) \rangle \\
&= f(x^t) - \alpha_{H_t} \langle \nabla f(x^t), \nabla f(x^t) \rangle - \alpha_{H_t} \langle \nabla f(x^t), e_t \rangle - \beta_{H_t} \langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t) \rangle + \beta_{H_t} \langle \nabla g_{J_t}(x^t), e_t \rangle \\
&\quad + \frac{L\alpha_{H_t}^2}{2} \| \nabla f(x^t) \|^2 + L\alpha_{H_t}^2 \langle \nabla f(x^t), e_t \rangle + \frac{L\alpha_{H_t}^2}{2} \| e_t \|^2 + \frac{L\beta_{H_t}^2}{2} \| \nabla g_{J_t}(x^t) \|^2 + L \alpha_{H_t} \beta_{H_t} \langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t) \rangle \\
&= f(x^t) - \left(\alpha_{H_t} - \frac{L}{2} \alpha_{H_t}^2 \right) \| \nabla f(x^t) \|^2 + \frac{L}{2} \beta_{H_t}^2 \| \nabla g_{J_t}(x^t) \|^2 - \beta_{H_t} (1 - \alpha_{H_t} L) \langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t) \rangle \\
&\quad + \left(L\alpha_{H_t}^2 - \alpha_{H_t} \right) \langle \nabla f(x^t), e_t \rangle + \beta_{H_t} \langle \nabla g_{J_t}(x^t), e_t \rangle + \frac{L}{2} \alpha_{H_t}^2 \| e_t \|^2.
\end{aligned} \tag{20}$$

To show the proposed theoretical convergence analysis of nonconvex continual learning, we define the catastrophic forgetting term Γ_t and the overfitting term B_t as follows:

$$\begin{aligned}
B_t &= (L\alpha_{H_t}^2 - \alpha_{H_t}) \langle \nabla f(x^t), e_t \rangle + \beta_{H_t} \langle \nabla g_{J_t}(x^t), e_t \rangle, \\
\Gamma_t &= \frac{\beta_{H_t}^2 L}{2} \| \nabla g_{J_t}(x^t) \|^2 - \beta_{H_t} (1 - \alpha_{H_t} L) \langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t) \rangle.
\end{aligned}$$

Then, we can rewrite Equation 20 as

$$f(x^{t+1}) \leq f(x^t) - \left(\alpha_{H_t} - \frac{L}{2} \alpha_{H_t}^2 \right) \| \nabla f(x^t) \|^2 + \Gamma_t + B_t + \frac{L}{2} \alpha_{H_t}^2 \| e_t \|^2. \tag{21}$$

We first note that B_t is dependent of the error term e_t with the batch I_t. In the continual learning step, an training agent cannot access $\nabla f(x^t)$, then we cannot get the exact value of e_t. Furthermore, Γ_t is dependent of the gradients $\nabla f_{I_t}(x^t), \nabla g_{I_t}(x^t)$ and the learning rates $\alpha_{H_t}, \beta_{H_t}$.

Taking expectations with respect to I_t on both sides given J_t, we have

$$\begin{aligned}
\mathbb{E}_{I_t} \left[f(x^{t+1}) \right] &\leq \mathbb{E}_{I_t} \left[f(x^t) - \left(\alpha_{H_t} - \frac{L}{2} \alpha_{H_t}^2 \right) \| \nabla f(x^t) \|^2 + \Gamma_t + B_t + \frac{L}{2} \alpha_{H_t}^2 \| e_t \|^2 \Big| J_t \right] \\
&\leq \mathbb{E}_{I_t} \left[f(x^t) - \left(\alpha_{H_t} - \frac{L}{2} \alpha_{H_t}^2 \right) \| \nabla f(x^t) \|^2 + \frac{L}{2} \alpha_{H_t}^2 \| e_t \|^2 \right] + \mathbb{E}_{I_t} \left[\Gamma_t + B_t \Big| J_t \right].
\end{aligned}$$

Now, taking expectations over the whole stochasticity we obtain

$$\mathbb{E} \left[f(x^{t+1}) \right] \leq \mathbb{E} \left[f(x^t) - \left(\alpha_{H_t} - \frac{L}{2} \alpha_{H_t}^2 \right) \| \nabla f(x^t) \|^2 + \Gamma_t + B_t + \frac{L}{2} \alpha_{H_t}^2 \| e_t \|^2 \right].$$

Rearranging the terms and assume that $\frac{1}{1 - L\alpha_{H_t}/2} > 0$, we have

$$\left(\alpha_{H_t} - \frac{L}{2}\alpha_{H_t}^2\right) \mathbb{E}\|\nabla f(x^t)\|^2 \le \mathbb{E}\left[f(x^t) - f(x^{t+1}) + \Gamma_t + B_t + \frac{L}{2}\alpha_{H_t}^2\|e_t\|^2\right]$$

and

$$\begin{aligned}\mathbb{E}\|\nabla f(x^t)\|^2 &\le \mathbb{E}\left[\frac{1}{\alpha_{H_t}(1-\frac{L}{2}\alpha_{H_t})}\left(f(x^t) - f(x^{t+1}) + \Gamma_t + B_t\right) + \frac{\alpha_{H_t}L}{2(1-\frac{L}{2}\alpha_{H_t})}\|e_t\|^2\right] \\ &\le \mathbb{E}\left[\frac{1}{\alpha_{H_t}(1-\frac{L}{2}\alpha_{H_t})}\left(f(x^t) - f(x^{t+1}) + \Gamma_t + B_t\right) + \frac{\alpha_{H_t}L}{2(1-\frac{L}{2}\alpha_{H_t})}\sigma_f^2\right].\end{aligned}$$

■

Proof [**Proof of Theorem 1**] Suppose that the learning rate α_{H_t} is a constant $\alpha = c/\sqrt{T}$, for $c > 0$, $1 - \frac{L}{2}\alpha = \frac{1}{A} > 0$. Then, by summing Equation 14 from $t = 0$ to $T - 1$, we have

$$\begin{aligned}\min_t \mathbb{E}\|\nabla f(x^t)\|^2 &\le \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x^t)\|^2 \\ &\le \frac{1}{1-\frac{L}{2}\alpha}\left(\frac{1}{\alpha T}\left(f(x^0) - f(x^T) + \sum_{t=0}^{T-1}\left(\mathbb{E}\left[B_t + \Gamma_t\right]\right)\right) + \frac{L}{2}\alpha\sigma_f^2\right) \\ &= \frac{1}{1-\frac{L}{2}\alpha}\left(\frac{1}{c\sqrt{T}}\left(\Delta_f + \sum_{t=0}^{T-1}\left(\mathbb{E}\left[B_t + \Gamma_t\right]\right)\right) + \frac{Lc}{2\sqrt{T}}\sigma_f^2\right) \\ &= \frac{A}{\sqrt{T}}\left(\frac{1}{c}\left(\Delta_f + \sum_{t=0}^{T-1}\mathbb{E}\left[B_t + \Gamma_t\right]\right) + \frac{Lc}{2}\sigma_f^2\right). \end{aligned} \tag{22}$$

We note that a batch I_t is sampled from a memory $M_t \subset M$ which is a random vector whose element is a datapoint $d \in P \cup C$. Then, taking expectation over $I_t \subset M_t \subset P \cup C$ implies that $\mathbb{E}[B_t] = 0$. Therefore, we get the minimum of expected square of the norm of gradients

$$\min_t \mathbb{E}\|\nabla f(x^t)\|^2 \le \frac{A}{\sqrt{T}}\left(\frac{1}{c}\left(\Delta_f + \sum_{t=0}^{T-1}\mathbb{E}[\Gamma_t]\right) + \frac{Lc}{2}\sigma_f^2\right).$$

■

Lemma 8 *Suppose that $I_t \cap J_t = \emptyset$, Taking expectation over $I_t \subset M_t$ and $J_t \subset C$, we have*

$$\min_t \mathbb{E}\|\nabla h|_{M\cup C}(x^t)\|^2 \le \sqrt{\frac{2\Delta_{h|_{M\cup C}}L}{T}}\sigma_{h|_{M\cup C}}, \tag{23}$$

where $\Delta_{h|_{M\cup C}}$ and $\sigma_{h|_{M\cup C}}$ is the version of loss gap and the variance for h on $M \cup C$, respectively.

Proof [**Proof of Lemma 8**] To simplify the proof, we assume that learning rates $\alpha_{H_t}, \beta_{H_t}$ are a same fixed value $\beta = c'/\sqrt{T}$. The assumption is reasonable, because it is observed that the RHS of Equation 14 is not perturbed drastically by small learning rates in $0 < \alpha_{H_t}, \beta_{H_t} \leq 2/L \ll 1$. Let us denote the union of M_t over time $0 \leq t \leq T-1$ as $M = \bigcup_t M_t$. By the assumption, it is equivalent to update on $M \cup C$. Then, the non-convex finite sum optimization is given as

$$\min_{x \in \mathbb{R}^d} h|_{M \cup C}(x) = \frac{1}{n_g + |M|} \sum_{i \in M \cup C} h_i(x), \tag{24}$$

where $|M|$ is the number of elements in M. This problem can be solved by a simple SGD algorithm [25]. Thus, we have

$$\min_t \mathbb{E}\|\nabla h|_{M \cup C}(x^t)\|^2 \leq \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\|\nabla h|_{M \cup C}(x^t)\|^2 \leq \sqrt{\frac{2\Delta_{h|_{M \cup C}} L}{T}} \sigma_{h|_{M \cup C}}. \tag{25}$$

■

Lemma 9 *For any $C \subset D \subset M \cup C$, define $\omega^2_{h|_D}$ as*

$$\omega^2_{h|_D} = \sup_x \mathbb{E}_{j \in D}\|\nabla h_j(x^t) - \nabla h|_{M \cup C}(x^t)\|^2].$$

Then, we have

$$\mathbb{E}\|\nabla g_{J_t}(x^t)\|^2 \leq \mathbb{E}\|\nabla h|_{M \cup C}(x^t)\|^2 + \sup_{C \subset D \subset M \cup C} \omega^2_{h|_D}. \tag{26}$$

Proof [**Proof of Lemma 9**] We arrive at the following result by Jensen's inequality

$$\begin{aligned}
\sup_x \mathbb{E}_{J_t \subset C}\|\nabla g_{J_t}(x^t) - \nabla h|_{M \cup C}(x^t)\|^2 &= \sup_x \mathbb{E}_{J_t \subset C}\left[\|\mathbb{E}_{j \in J_t}[\nabla h_j(x^t)] - \nabla h|_{M \cup C}(x^t)\|^2\right] &(27)\\
&\leq \sup_{C \subset D \subset M \cup C} \sup_x \mathbb{E}_{J_t \subset D}\left[\|\mathbb{E}_{j \in J_t}[\nabla h_j(x^t)] - \nabla h|_{M \cup C}(x^t)\|^2\right] &(28)\\
&\leq \sup_{C \subset D \subset M \cup C}\left[\sup_x \mathbb{E}_{j \in D}[\ \|\nabla h_j(x^t) - \nabla h|_{M \cup C}(x^t)\|^2]\right] &(29)\\
&= \sup_{C \subset D \subset M \cup C} \omega^2_{h|_D}. &(30)
\end{aligned}$$

By the triangular inequality, we get

$$\begin{aligned}
\mathbb{E}\|\nabla g_{J_t}(x^t)\|^2 &\leq \mathbb{E}\|\nabla g_{J_t}(x^t) - \nabla h|_{M \cup C}(x^t)\|^2 + \mathbb{E}\|\nabla h|_{M \cup C}(x^t)\|^2 &(31)\\
&\leq \mathbb{E}\|\nabla h|_{M \cup C}(x^t)\|^2 + \sup_{C \subset D \subset M \cup C} \omega^2_{h|_D}. &(32)
\end{aligned}$$

■

For continual learning, the model x^0 reaches to an ϵ-stationary point of $f(x)$ when we have finished to learn P and start to learn C. Now, we discuss the frequency of transfer and interference during continual learning before showing Lemma 2. It is well known that the frequencies between interference and transfer have similar values (the frequency of constraint violation is approximately 0.5 for AGEM) as shown in Appendix D of [2]. Even if memory-based continual learning has a small memory buffer which contains a subset of P, random sampling from the buffer allows to have similar frequencies between interference and transfer.

In this paper, we consider two cases for the upper bound of $\mathbb{E}[\Gamma_t]$, the moderate case and the worst case. For **the moderate case**, which covers most continual learning scenarios, we assume that the inner product term $\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle$ has the same probabilities of being positive (transfer) and negative (interference). Then, we can approximate $\mathbb{E}[\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle] \approx 0$ over all randomness. For **the worst case**, we assume that all $\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle$ has negative values.

Proof [**Proof of Lemma 2**] For the moderate case, we derive the rough upper bound of $\mathbb{E}[\Gamma_t]$:

$$\mathbb{E}\left[\Gamma_t\right] = \mathbb{E}\left[\frac{\beta_{H_t}^2 L}{2}\|\nabla g_{J_t}(x^t)\|^2 - \beta_{H_t}(1-\alpha_{H_t}L)\langle \nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle\right] \tag{33}$$

$$\approx \mathbb{E}\left[\frac{\beta_{H_t}^2 L}{2}\|\nabla g_{J_t}(x^t)\|^2\right] \tag{34}$$

$$= O\left(\mathbb{E}\left[\frac{\beta^2 L}{2}\|\nabla g_{J_t}(x^t)\|^2\right]\right) \tag{35}$$

By plugging Lemma 9 into $\mathbb{E}[\Gamma_t]$, we obtain that

$$\mathbb{E}[\Gamma_t] \leq O\left(\mathbb{E}\left[\frac{\beta^2 L}{2}\|\nabla g_{J_t}(x^t)\|^2\right]\right) \tag{36}$$

$$= O\left(\mathbb{E}\left[\frac{\beta^2 L}{2}\|\nabla h|_{M\cup C}(x^t)\|^2 + \frac{\beta^2 L}{2}\sup_{C\subset D\subset M\cup C}\omega_{h|_D}^2\right]\right). \tag{37}$$

We use the technique for summing up in the proof of Theorem 1, then the cumulative sum of catastrophic forgetting term is derived as

$$\sum_{t=0}^{T-1}\mathbb{E}[\Gamma_t] \leq \sum_{t=0}^{T-1}\frac{\beta^2 L}{2} O\left(\mathbb{E}\left[\|h|_{M\cup C}(x^t)\|^2\right] + \sup_{C\subset D\subset M\cup C}\omega_{h|_D}^2\right) \tag{38}$$

$$\leq \frac{\beta^2 L}{2}\sum_{t=0}^{T-1} O\left(\frac{1}{\beta}\left[h|_{M\cup C}(x^t) - h|_{M\cup C}(x^{t+1})\right] + \frac{L\beta}{2}\sigma_{h|_{M\cup C}}^2 + \sup_{C\subset D\subset M\cup C}\omega_{h|_D}^2\right) \tag{39}$$

$$\leq \frac{\beta^2 L}{2} O\left(\frac{1}{\beta}\Delta_{h|_{M\cup C}} + \frac{TL\beta}{2}\sigma_{h|_{M\cup C}}^2 + T\sup_{C\subset D\subset M\cup C}\omega_{h|_D}^2\right) \tag{40}$$

$$= O\left(\beta\Delta_{h|_{M\cup C}} + \frac{TL\beta^3}{2}\sigma_{h|_{M\cup C}}^2 + T\beta^2\sup_{C\subset D\subset M\cup C}\omega_{h|_D}^2\right). \tag{41}$$

Now, we consider the randomness of memory choice. Let D^* be as follows:

$$D^* = \operatorname*{arg\,max}_{C\subset D\subset P\cup C} \beta\Delta_{h|_D} + \frac{TL\beta^3}{2}\sigma_{h|_D}^2. \tag{42}$$

Then, we obtain the following inequality,

$$\sum_{t=0}^{T-1} \mathbb{E}[\Gamma_t] \leq O\left(\beta\Delta_{h|_{D^*}} + \frac{TL\beta^3}{2}\sigma^2_{h|_{D^*}} + T\beta^2 \sup_{C\subset D\subset M\cup C} \omega^2_{h|_D}\right) \tag{43}$$

$$\leq O\left(\beta\Delta_{h|_{D^*}} + \frac{TL\beta^3}{2}\sigma^2_{h|_{D^*}} + T\beta^2 \sup_{C\subset D\subset P\cup C} \omega^2_{h|_D}\right). \tag{44}$$

Rearranging the above equation, we get

$$\sum_{t=0}^{T-1} \mathbb{E}[\Gamma_t] \leq O\left(T\left(\frac{L\beta^3}{2}\sigma^2_{h|_{D^*}} + \beta^2 \sup_{C\subset D\subset P\cup C} \omega^2_{h|_D}\right) + \beta\Delta_{h|_{D^*}}\right). \tag{45}$$

For the moderate case, we provide the derivations of the convergence rate for two cases of β as follows.

When $\beta < \alpha = c/\sqrt{T}$, the upper bound always satisfies

$$\sum_{t=0}^{T-1} \frac{\mathbb{E}[\Gamma_t]}{\sqrt{T}} \leq \frac{1}{\sqrt{T}} O\left(\frac{1}{T}\left(\frac{L\beta}{2}\sigma^2_{h|_{D^*}} + \frac{1}{\sqrt{T}} \sup_{C\subset D\subset P\cup C} \omega^2_{h|_D}\right) + \frac{1}{\sqrt{T}}\Delta_{h|_{D^*}}\right) < O\left(\frac{1}{T^{3/2}} + \frac{1}{T}\right).$$

For $\beta \geq \alpha = c/\sqrt{T}$, we cannot derive a tighter bound, so we still have

$$\sum_{t=0}^{T-1} \frac{\mathbb{E}[\Gamma_t]}{\sqrt{T}} \leq \frac{1}{\sqrt{T}} O\left(T\left(\frac{L\beta^3}{2}\sigma^2_{h|_{D^*}} + \beta^2 \sup_{C\subset D\subset P\cup C} \omega^2_{h|_D}\right) + \beta\Delta_{h|_{D^*}}\right) = O\left(\sqrt{T} + \frac{1}{\sqrt{T}}\right).$$

For the worst case, we assume that there exists a constant $c_{f,g}$ which satisfies $c_{f,g}\|\nabla g_{J_t}(x^t)\| \geq \|\nabla f_{I_t}(x^t)\|$.

$$\mathbb{E}\left[\Gamma_t\right] = \mathbb{E}\left[\frac{\beta^2_{H_t}L}{2}\|\nabla g_{J_t}(x^t)\|^2 - \beta_{H_t}(1-\alpha_{H_t}L)\langle\nabla f_{I_t}(x^t), \nabla g_{J_t}(x^t)\rangle\right] \tag{46}$$

$$\leq \mathbb{E}\left[\frac{\beta^2_{H_t}L}{2}\|\nabla g_{J_t}(x^t)\|^2 + \beta_{H_t}(1-\alpha_{H_t}L)\|\nabla f_{I_t}(x^t)\|\|\nabla g_{J_t}(x^t)\|\right] \tag{47}$$

$$\leq \mathbb{E}\left[\frac{\beta^2 L}{2}\|\nabla g_{J_t}(x^t)\|^2 + \beta c_{f,g}\|\nabla g_{J_t}(x^t)\|^2\right] \tag{48}$$

$$= O\left(\mathbb{E}\left[\left(\beta^2+\beta\right)\|\nabla g_{J_t}(x^t)\|^2\right]\right). \tag{49}$$

By plugging Lemma 9 into $\mathbb{E}[\Gamma_t]$, we obtain that

$$\mathbb{E}[\Gamma_t] \leq O\left(\mathbb{E}\left[\left(\beta^2+\beta\right)\|\nabla g_{J_t}(x^t)\|^2\right]\right) \tag{50}$$

$$= O\left(\left(\beta^2+\beta\right)\mathbb{E}\left[\|\nabla h|_{M\cup C}(x^t)\|^2 + \sup_{C\subset D\subset M\cup C} \omega^2_{h|_D}\right]\right). \tag{51}$$

We use the technique for summing up in the proof of Theorem 1, then the cumulative sum of catastrophic forgetting term is derived as

$$\sum_{t=0}^{T-1} \mathbb{E}[\Gamma_t] \leq \sum_{t=0}^{T-1} \left(\beta^2+\beta\right) O\left(\mathbb{E}\left[\|h|_{M\cup C}(x^t)\|^2\right] + \sup_{C\subset D\subset M\cup C} \omega_{h|_D}^2\right) \tag{52}$$

$$\leq \left(\beta^2+\beta\right) \sum_{t=0}^{T-1} O\left(\frac{1}{\beta}\left[h|_{M\cup C}(x^t) - h|_{M\cup C}(x^{t+1})\right] + \frac{L\beta}{2}\sigma_{h|_{M\cup C}}^2 + \sup_{C\subset D\subset M\cup C} \omega_{h|_D}^2\right) \tag{53}$$

$$\leq \left(\beta^2+\beta\right) O\left(\frac{1}{\beta}\Delta_{h|_{M\cup C}} + \frac{TL\beta}{2}\sigma_{h|_{M\cup C}}^2 + T\sup_{C\subset D\subset M\cup C} \omega_{h|_D}^2\right) \tag{54}$$

$$= O\left((\beta+1)\Delta_{h|_{M\cup C}} + \frac{TL\beta^2(\beta+1)}{2}\sigma_{h|_{M\cup C}}^2 + T\beta(\beta+1)\sup_{C\subset D\subset M\cup C} \omega_{h|_D}^2\right). \tag{55}$$

For the worst case, we provide the derivations of the convergence rate for two cases of β as follows.

When $\beta < \alpha = c/\sqrt{T}$, the upper bound always satisfies

$$\sum_{t=0}^{T-1} \frac{\mathbb{E}[\Gamma_t]}{\sqrt{T}} \leq \frac{1}{\sqrt{T}} O\left(\frac{Lc+\sqrt{T}}{\sqrt{T}}\sigma_{h|_{D^*}}^2 + (\sqrt{T}+c)\sup_{C\subset D\subset P\cup C} \omega_{h|_D}^2 + \frac{\sqrt{T}+c}{\sqrt{T}}\Delta_{h|_{D^*}}\right) < O\left(\frac{1}{T} + \frac{1}{\sqrt{T}} + 1\right).$$

For $\beta \geq \alpha = c/\sqrt{T}$, we cannot derive a tighter bound, so we still have

$$\sum_{t=0}^{T-1} \frac{\mathbb{E}[\Gamma_t]}{\sqrt{T}} \leq \frac{1}{\sqrt{T}} O\left(T\left(\frac{L\beta^2(\beta+1)}{2}\sigma_{h|_{D^*}}^2 + \beta(\beta+1)\sup_{C\subset D\subset P\cup C} \omega_{h|_D}^2\right) + (\beta+1)\Delta_{h|_{D^*}}\right) = O\left(\sqrt{T} + \frac{1}{\sqrt{T}}\right).$$

■

Even if we consider the worst case, we still have $O(1)$ for the cumulative forgetting $\mathbb{E}[\Gamma_t]$ when $\beta < \alpha$. This implies that we have the theoretical condition for control the forgetting on $f(x)$ while evolving on C. In the main text, we only discuss the moderate case to emphasize $f(x)$ can be converged by the effect of transfer during continual learning, but we have also considered the worst case can be well treated by our theoretical condition by keeping the convergence of $f(x)$ over time as follows.

Corollary 10 *Let $\beta_{H_t} < \alpha = \frac{c}{\sqrt{T}}$ for all t. Then we have the convergence rate*

$$\min_t \mathbb{E}\|\nabla f(x^t)\|^2 \leq O\left(\frac{1}{\sqrt{T}}\right). \tag{56}$$

Otherwise, $f(x)$ is not guaranteed to converge when $\beta \geq \alpha$ and might diverge at the rate $O(\sqrt{T})$.

Proof [**Proof of Corollary 10**]

By Lemma 2, we have

$$\sum_{t=0}^{T-1} \frac{\mathbb{E}[\Gamma_t]}{\sqrt{T}} < O\left(\frac{1}{T^{3/2}} + \frac{1}{T}\right)$$

for $\beta < \alpha$ for **the moderate case**. Then, we can apply the result into RHS of Equation 4 in Theorem 1 as follows.

$$\begin{aligned}
\min_t \mathbb{E}\|\nabla f(x^t)\|^2 &\leq \frac{A}{\sqrt{T}}\left(\frac{1}{c}\left(\Delta_f + \sum_{t=0}^{T-1}\mathbb{E}\left[\Gamma_t\right]\right) + \frac{Lc}{2}\sigma_f^2\right) \\
&= \frac{A/c}{\sqrt{T}}\left(\Delta_f + \frac{Lc^2}{2}\sigma_f^2\right) + \frac{A/c}{\sqrt{T}}\sum_{t=0}^{T-1}\mathbb{E}[\Gamma_t] \\
&= O\left(\frac{1}{T^{3/2}} + \frac{1}{T} + \frac{1}{T^{1/2}}\right) = O\left(\frac{1}{\sqrt{T}}\right).
\end{aligned}$$

In addition, we have the convergence rate of $f(x)$ for **the worst case** as follows:

$$\min_t \mathbb{E}\|\nabla f(x^t)\|^2 = O(1), \tag{57}$$

which implies that $f(x)$ can keep the convergence while evolving on C.

■

Proof [**Proof of Corollary 3**] To formulate the IFO calls, Recall that $T(\epsilon)$

$$T(\epsilon) = \min\left\{T:\ \min\ \mathbb{E}\|\nabla f(x^t)\|^2 \leq \epsilon\right\}.$$

A single IFO call is invested in calculating each step, and we now compute IFO calls to reach an ϵ-accurate solution.

$$\frac{A}{\sqrt{T}}\left(\frac{1}{c}\left(\Delta_f + \sum_{t=0}^{T-1}\mathbb{E}\left[\Gamma_t\right]\right) + \frac{Lc}{2}\sigma_f^2\right) \to \epsilon.$$

When $\beta < \alpha$, we get

$$\text{IFO calls} = O\left(\frac{1}{\epsilon^2}\right).$$

Otherwise, when $\beta \geq \alpha$, we cannot guarantee the upper bound of stationary decreases over time. Then, we cannot compute IFO calls for this case.

■

D.7. Remarks

Surprisingly, we observe $\mathbb{E}_t[B_t] = 0$ by Lemma 5. It should be also noted that the individual trial with a randomly given M_0 cannot cancel the effect of B_t. We discuss more details of overfitting to memory in Appendix E.

The convergence rate with respect to the marginalization on M_0 in Corollary 10 exactly match the usual nonconvex SGD rates. As empirically shown in stable A-GEM and stable ER-Reservoir [20], the condition of $\beta_{H_t} < \alpha$ theoretically implies that decaying step size is a key solution to continual learning considering we can pick any arbitrary observation points. In addition, it should be also noted that the selection rules for M_0 with various memory schemes are important to perturb the convergence rate based on the mean value in Equation 56 for each trial. This is why memory schemes matters in continual learning in terms of variance. Please see more details in Appendix E.

Appendix E. Overfitting to replay Memory

In Lemma 7, we show the expectation of stepwise change of upper bound. Now, we discuss the distribution of the upper bound by analyzing the random variable B_t. As B_t is computed by getting

$$B_t = (L\alpha_{H_t}^2 - \alpha_{H_t})\langle \nabla f(x^t), e_t \rangle + \beta_{H_t}\langle \nabla g_{J_t}(x^t), e_t \rangle.$$

The purpose of our convergence analysis is to compute the upper bound of Equation 14, then we compute the upper bound of B_t.

$$B_t \leq (L\alpha_{H_t}^2 - \alpha_{H_t})\|\nabla f(x^t)\|\|e_t\| + \beta_{H_t}\|\nabla g_{J_t}(x^t)\|\|e_t\|.$$

It is noted that the upper bound is related to the distribution of the norm of e_t. We have already know that $\mathbb{E}[e_t] = 0$, so we consider its variance, Var($\|e_t\|$) in this section. Let us denote the number of data points of P in a memory M_0 as m_P. We assume that M_0 is uniformly sampled from P. Then the sample variance, Var($\|e_t\|$) is computed as

$$\text{Var}(\|e_t\|) = \frac{n_f - m_P}{(n_f - 1)m_P}\sigma_f^2$$

by the similar derivation with Equation 27. The above result directly can be applied to the variance of B_t. This implies m_t is a key feature which has an effect on the convergence rate. It is noted that the larger m_P has the smaller variance by applying schemes, such as larger memory. In addition, the distributions of e_t and $\nabla f_{I_t}(x^t)$ are different with various memory schemes. Therefore, we can observe that memory schemes differ the performance even if we apply same step sizes.