Meta-analysis of Continual Learning

Cuong V. Nguyen[†], **Alessandro Achille**[†], **Michael Lam**[†], **Tal Hassner**[‡]*,
Vijay Mahadevan[†], **Stefano Soatto**[†]
[†]Amazon Web Services
{nguycuo,aachille,michlam,vmahad,soattos}@amazon.com
[‡]Facebook Inc.
{talhassner}@gmail.com

Abstract

We propose a novel meta-analysis to study the relationship between properties of task sequences and the performance of continual learning algorithms. Our analysis makes use of recent developments in task space modeling as well as correlation analysis to specify and analyze the properties we are interested in. As a case study, we apply our meta-analysis to study two properties of a task sequence: *total complexity* and *sequential heterogeneity*. The findings from our analysis suggest directions for improving continual learning benchmarks and methods.[2]

1 Introduction

Continual learning (CL) [19, 23] is the ability of a model to continuously learn from a stream of data, which could possibly be non-iid or come from different but related tasks. In recent years, interest in CL has risen [1, 9, 18, 25, 28, 29] due to its potential to reduce training time and training set sizes, both of which are critical for training modern deep networks. However, CL by deep models has proven to be challenging due to catastrophic forgetting [10, 17], the tendency of a model to forget previously learned tasks if not trained properly on a new task. Recent work attempted to tackle this problem either by better training algorithms [4, 13, 14, 20, 34], structure sharing [22, 26, 32], episodic memory [6, 16, 18], machine-generated pseudo-data [12, 15, 27], or a combination of those approaches [18, 24]. Benchmarks to compare these methods typically constructed a sequence of tasks and measured the algorithms' performance when transferring from one task to another. Two popular benchmarks are permuted MNIST [10] and split MNIST [34].

In this paper, we seek to understand CL at a more fundamental level. Specifically, we investigate the following question: *Given a sequence of tasks, which properties of the tasks influence the hardness of the entire sequence?*

An answer to this question is useful in several ways. First, it helps us estimate the hardness of a benchmark based on its individual tasks, thereby potentially assisting the development of new and better benchmarks for CL. Additionally, knowing the hardness of a task sequence allows us to estimate a priori the cost and limits of running CL algorithms on it. Crucially, by gaining a better understanding at a more fundamental level, we have more insights to develop better CL methods.

This work is the first attempt to answer the above question. To this end, we propose a novel meta-analysis to study the relationship between properties of task sequences and their hardness. Our analysis makes use of recent task space modeling methods, such as Task2Vec [2], to specify the interested properties and then applies correlation analysis to study the relationship between the specified properties and the actual measures of task sequence hardness.

[*] Work done at Amazon.

[2] A long version of this paper is available at: `https://arxiv.org/abs/1908.01091`.

3rd Workshop on Meta-Learning at NeurIPS 2019, Vancouver, Canada.

As a case study, we apply our meta-analysis to study two properties of a task sequence—*total complexity* and *sequential heterogeneity*. Total complexity measures the total hardness of individual tasks in the sequence, while sequential heterogeneity measures the total dissimilarity between pairs of consecutive tasks. We show how these two properties can be estimated using the Task2Vec framework [2], which maps datasets (equivalently, tasks) to vectors on a vector space. We choose these two properties for our case study because of their intuitive relationships to the hardness of task sequences: since CL algorithms attempt to transfer knowledge from one task to another, both the hardness of each individual task and the dissimilarity between them should play a role in determining the effectiveness of the transfer.

The findings from our analysis are summarized as follows: (1) total complexity has a *strong correlation* with task sequence hardness, while (2) sequential heterogeneity has *little or no correlation* with task sequence hardness. When factoring out the task complexity, we even find negative correlations between sequential heterogeneity and task sequence hardness in some cases.

The first finding, although expected, emphasizes that we should explicitly take into account the complexity of each task when designing new algorithms or benchmarks, which is currently lacking in CL research. Besides, the community is currently divided on the issue whether task similarity helps or hurts CL. Some authors showed that task similarity helps improve performance in the context of transfer learning [2, 3, 21], while some others conjectured that task dissimilarity could help improve CL [8]. Our second finding gives evidence that supports the latter view. Our findings also suggest that (a) task complexities should be explicitly considered when designing CL algorithms and benchmarks, and (b) CL algorithms should be customized for specific task pairs to improve their effectiveness.

2 Meta-analysis of continual learning algorithms

Our meta-analysis is conceptually simple and consists of the following steps:

(1) *Specify the properties* of a task sequence that we are interested in and *estimate these properties* using a suitable task space modeling method.

(2) *Estimate actual measures of task sequence hardness* from real experiments. In our case, task sequence hardness is measured as the final error rate of a model trained sequentially on the sequence.

(3) Use *correlation analysis* to study the correlations between the estimated properties in Step 1 and the actual measures in Step 2.

This meta-analysis can be used even in other cases, such as transfer or multi-task learning, to study properties of new algorithms.

3 Case study: total complexity and sequential heterogeneity

As a case study, we apply our meta-analysis to two properties of task sequences: *total complexity* and *sequential heterogeneity*. We shall define these properties and detail the methodology to estimate them using Task2Vec [2], a recently developed framework for embedding visual classification tasks as vectors in a real vector space. The embeddings have many desirable properties that allow reasoning about the semantic and taxonomic relations between different visual tasks. Alternatives to Task2Vec, such as [7, 31, 33], can also be used in our analysis.

Total complexity (TC). The total complexity of a task sequence is the sum of the complexities of its individual tasks. Formally, let $T = (t_1, t_2, \ldots, t_k)$ be a sequence of k *distinct* tasks and $c(t)$ be a function measuring the complexity of a task t. The total complexity of the task sequence T is $C(T) = \sum_{i=1}^{k} c(t_i)$.

We can estimate $c(t)$ from the Task2Vec embedding of task t. Specifically, we measure the complexity of task t by its distance to the trivial task in the embedding space. That is, $c(t) = d(e_t, e_0)$, where e_t and e_0 are the embeddings of task t and the trivial task respectively, and d is a symmetric distance between two tasks in the embedding space. Following [2], we choose d to be the normalized cosine distance, $d(e_t, e_{t'}) = \cos\left(\frac{e_t}{e_t + e_{t'}}, \frac{e_{t'}}{e_t + e_{t'}}\right)$, since it was shown to be well correlated with natural distances between tasks.

Sequential heterogeneity (SH). The sequential heterogeneity of a task sequence is the sum of the dissimilarities between all pairs of consecutive tasks in the sequence. Formally, for a task sequence $T = (t_1, t_2, \ldots, t_k)$ of distinct tasks, its sequential heterogeneity is $F(T) = \sum_{i=1}^{k-1} f(t_i, t_{i+1})$, where $f(t, t')$ is a function measuring the dissimilarity between tasks t and t' and can be estimated from the Task2Vec embeddings. For our purpose, it is clear that we can use the distance d above as an estimate for f. That is, $f(t, t') = d(e_t, e_{t'})$.

Correlation analysis. Having defined total complexity and sequential heterogeneity, we now discuss how we can study their relationships to task sequence hardness. Given a task sequence $T = (t_1, t_2, \ldots, t_k)$, we measure its actual hardness w.r.t. a CL algorithm A by the final test error rate obtained after running A on the tasks $t_1, t_2, \ldots, t_k$ sequentially. That is, the hardness of T with respect to A is $H_A(T) = \mathrm{err}_A(T)$. We use final error rate because it is an important metric commonly used to evaluate CL algorithms.

To analyze the relationships between task sequence hardness and total complexity or sequential heterogeneity, we employ correlation analysis as the main statistical tool. In particular, we sample M task sequences $T_1, T_2, \ldots, T_M$ and compute their hardness measures $(H_A(T_i))_{i=1}^M$ as well as their total complexity $(C(T_i))_{i=1}^M$ and sequential heterogeneity $(F(T_i))_{i=1}^M$ measures. From these measures, we compute the Pearson correlation coefficients between $(H_A(T_i))_{i=1}^M$ and $(C(T_i))_{i=1}^M$ or between $(H_A(T_i))_{i=1}^M$ and $(F(T_i))_{i=1}^M$, which tell us how correlated these quantities are.

When computing the correlations between H_A and C, we constrain the task sequences T_i to have the same length. The reason is that longer sequences tend to have larger complexities, thus the correlation may be biased toward sequence lengths rather than reflecting the complexity of individual tasks.

Normalized sequential heterogeneity (normSH). Since the complexity of individual tasks in a sequence may influence the heterogeneity between the tasks (e.g., an easy task may be more similar to another easy task than to a hard task), the complexity may indirectly affect the sequential heterogeneity. Thus, when computing the correlations between H_A and F, we can also constrain the total complexity of the task sequences T_i to be the same, so that individual tasks' complexities would not affect the correlations. This can be achieve by using the same set of individual tasks for all the sequences (i.e., the sequences are permutations of each other).

4 Experiments

Task construction. We conduct experiments on MNIST and CIFAR10, which are the most common datasets used to evaluate CL algorithms. For each dataset, we construct a more general *split* version as follows. First, we consider all pairs of different labels as a unit binary classification task, resulting in a total of 45 unit tasks. From these unit tasks, we then create 120 task sequences of length five by randomly drawing, for each sequence, five unit tasks without replacement. We also construct 120 *split* task sequences which are permutations of a fixed task set containing five random unit tasks to compute the normSH. For each unit task, we train its Task2Vec embedding using a ResNet18 [11] probe network pre-trained on a combined dataset containing both MNIST and CIFAR10.

Algorithms and network architectures. We analyze two CL algorithms: synaptic intelligence (SI) [34] and variational continual learning (VCL) [18, 30]. On MNIST, we also consider the coreset version of VCL (CVCL). These algorithms are among the state-of-the-arts on the considered datasets, with SI representing the regularization-based methods, VCL representing the Bayesian methods, and CVCL combining Bayesian and rehearsal methods.

On CIFAR10, we run SI with the same network architecture considered in [34]: a CNN with 4 convolutional layers, followed by 2 dense layers with dropout. Since VCL was not developed with convolutional layers, we flatten the input images and train with a fully connected network containing 4 hidden layers, each of which has 256 hidden units. On MNIST, we run both SI and VCL with a fully connected network containing 2 hidden layers, each of which has 256 hidden units. We denote this setting by MNIST-256^2.

Since MNIST is a relatively easy dataset, we may not observe meaningful results if all errors obtained from different sequences are low and not very different. To make the dataset harder for the learning algorithms, we also consider smaller, fully connected networks with a *single* hidden layer, containing either 50 hidden units (for MNIST-50) or 20 hidden units (for MNIST-20). Following [18, 34], we

Table 1: **Correlation coefficients (p-values) between error rate and (a) total complexity, (b) sequential heterogeneity, and (c) normalized sequential heterogeneity** of SI, VCL, and CVCL algorithms on 4 different tests conducted on CIFAR10 and MNIST. Results with statistical significance ($p < 0.05$) are shown in bold. Full scatter plots of the correlations are in Appendix A.

	Property	Algorithm	MNIST-256^2	MNIST-50	MNIST-20	CIFAR-10
(a)	Total Complexity	SI	**0.24** ($p < 0.01$)	**0.22** ($p < 0.05$)	**0.36** ($p < 0.01$)	**0.86** ($p < 0.01$)
		VCL	0.05 ($p = 0.59$)	0.17 ($p = 0.07$)	**0.21** ($p < 0.05$)	**0.69** ($p < 0.01$)
		CVCL	**0.28** ($p < 0.01$)	**0.41** ($p < 0.01$)	**0.37** ($p < 0.01$)	-
(b)	Sequential Heterogeneity	SI	-0.01 ($p = 0.86$)	0.05 ($p = 0.55$)	0.07 ($p = 0.48$)	**0.30** ($p < 0.01$)
		VCL	0.04 ($p = 0.69$)	0.01 ($p = 0.88$)	0.05 ($p = 0.58$)	**0.21** ($p < 0.05$)
		CVCL	0.09 ($p = 0.31$)	0.12 ($p = 0.18$)	0.18 ($p = 0.05$)	-
(c)	Normalized Sequential Heterogeneity	SI	-0.07 ($p = 0.43$)	-0.04 ($p = 0.65$)	0.05 ($p = 0.58$)	**-0.25** ($p < 0.01$)
		VCL	0.03 ($p = 0.76$)	**-0.20** ($p < 0.05$)	**-0.21** ($p < 0.05$)	-0.17 ($p = 0.06$)
		CVCL	-0.08 ($p = 0.37$)	**-0.26** ($p < 0.01$)	-0.16 ($p = 0.07$)	-

use the multi-head version of the models where a separate last layer is trained for each different task and the other weights are shared among tasks. For CVCL, we use random coresets with sizes 40, 40, 20 for MNIST-256^2, MNIST-50 and MNIST-20 respectively.

Results. Table 1(a) shows strong positive correlations between error rate and TC for both SI and VCL on CIFAR10, with a correlation coefficient of 0.86 for SI and 0.69 for VCL. These correlations are both statistically significant with $p < 0.01$. On the MNIST-256^2 settings, SI and CVCL have weak positive correlations with TC, where the correlation coefficients are 0.24 and 0.28 respectively. When we reduce network capacity and make the problem relatively harder (i.e., in MNIST-50 and MNIST-20), we observe stronger correlations for all three algorithms. With the smallest network in MNIST-20, all the algorithms have statistically significant positive correlation with TC.

In terms of SH, Table 1(b) shows it has a weak positive correlation with error rate on CIFAR10, with SI and VCL having correlation coefficients of 0.30 and 0.21 (both statistically significant) respectively. Interestingly, we find no significant correlation between error rate and SH in all the MNIST settings, which suggests that heterogeneity may not be a significant factor determining the performance of CL algorithms on this dataset.

We also look at the normSH in Table 1(c) where the set of tasks is fixed and thus task complexity has been factored out. Surprisingly, Table 1(c) reports some negative correlations between error rate and normSH. For example, the correlation coefficient for SI on CIFAR10 is -0.25 with a $p < 0.01$, while there is no significant correlation for this algorithm on the MNIST dataset. VCL, on the other hand, has negative correlations with coefficients -0.20 and -0.21 respectively on MNIST-50 and MNIST-20, with $p < 0.05$. CVCL also has negative correlation between its error rate and normSH on MNIST-50, with coefficient -0.26 and $p < 0.01$. These unexpected results suggest that in some cases, dissimilarity between tasks may even help CL algorithms, a fact contrary to the common assumption that the performance of CL algorithms would degrade if the tasks they need to solve are very different [3, 21].

5 Discussions

We now summarize some discussions regarding our analysis. More detailed discussions are given in Appendix B.

On total complexity. Although the correlation between error rate and task complexity seems trivial, we are still not very clear which definition of task sequence complexity would be best to explain the error rate (i.e., to give the best correlations). Our paper proposes the total complexity as a measure for this purpose. The strong correlations between error rate and total complexity found in our analysis show that task complexity is important for CL algorithms. However, it was usually not considered when designing CL algorithms or benchmarks. We suggest that task complexity is explicitly considered to improve algorithm and benchmark design. For example, different transfer methods can be used depending on whether one transfers from an easy task to a hard one or vice versa, rather than using a single transfer technique across all task complexities, as currently done in the literature. Similarly, when designing new benchmarks for CL, it is also useful to provide different complexity structures to test the effectiveness of CL algorithms on a broader range of scenarios and difficulty levels.

On sequential heterogeneity. The weak or negative correlations between error rate and sequential heterogeneity found in our analysis show an interesting contradiction to our intuition on the relationship between CL algorithms' performance and task dissimilarity. We emphasize that in our context, the weak and negative correlations are not a negative result, but actually a *positive* result. In fact, some previous work showed that task similarity helps improve performance in transfer learning [2, 3, 21], while some others claimed that task dissimilarity could help CL [8] although their discussion was more related to the permuted MNIST setting. Our finding gives evidence that supports the latter view in the split MNIST and split CIFAR10 settings. Deeper analysis into VCL and SI (see Appendix B) also suggests that dissimilarities between consecutive tasks may not be enough to explain CL algorithms' performance. Thus, a more expressive definition of heterogeneity in task sequence would be needed.

A Appendix: Plots from experiments

We show in Fig. 1, 2, and 3 below the scatter plots of the errors versus total complexity, sequential heterogeneity, and normalized sequential heterogeneity respectively, together with the linear regression fits and 95% confidence intervals for the meta-analysis in our experiments.

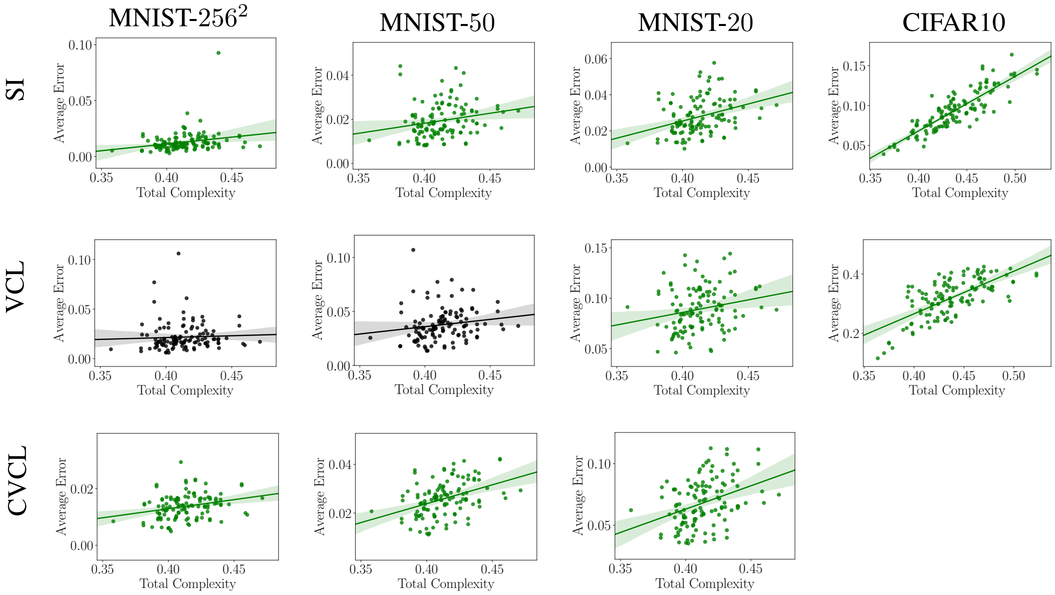

Figure 1: **Total complexity vs. average error**, together with the linear regression fit and 95% confidence interval, for each algorithm and test in Table 1(a). Green color indicates statistically significant positive correlations. Black color indicates negligible correlations.

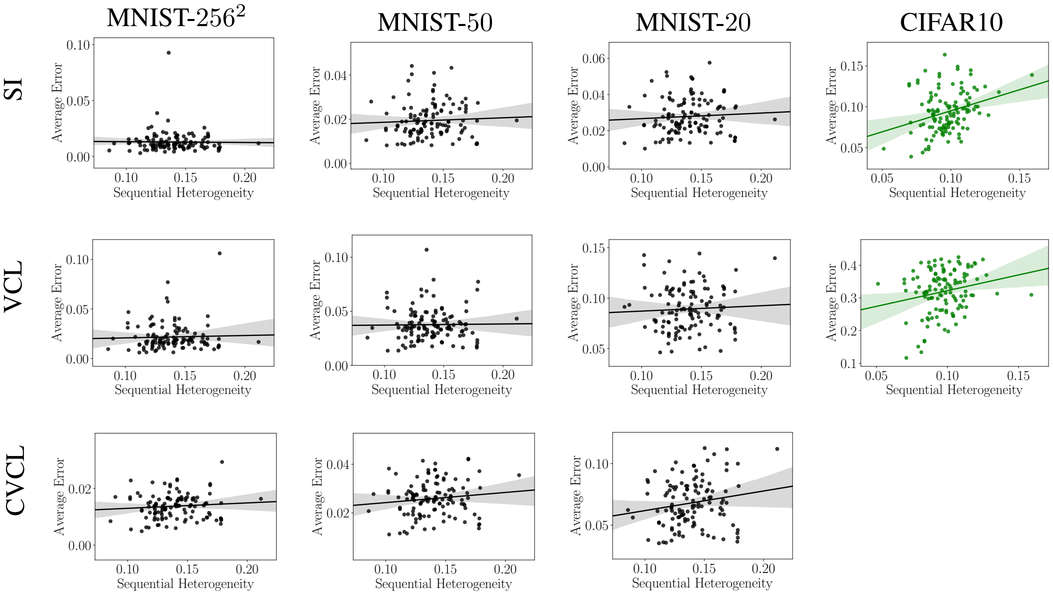

Figure 2: **Sequential heterogeneity vs. average error**, together with the linear regression fit and 95% confidence interval, for each algorithm and test in Table 1(b). Green color indicates statistically significant positive correlations. Black color indicates negligible correlations.

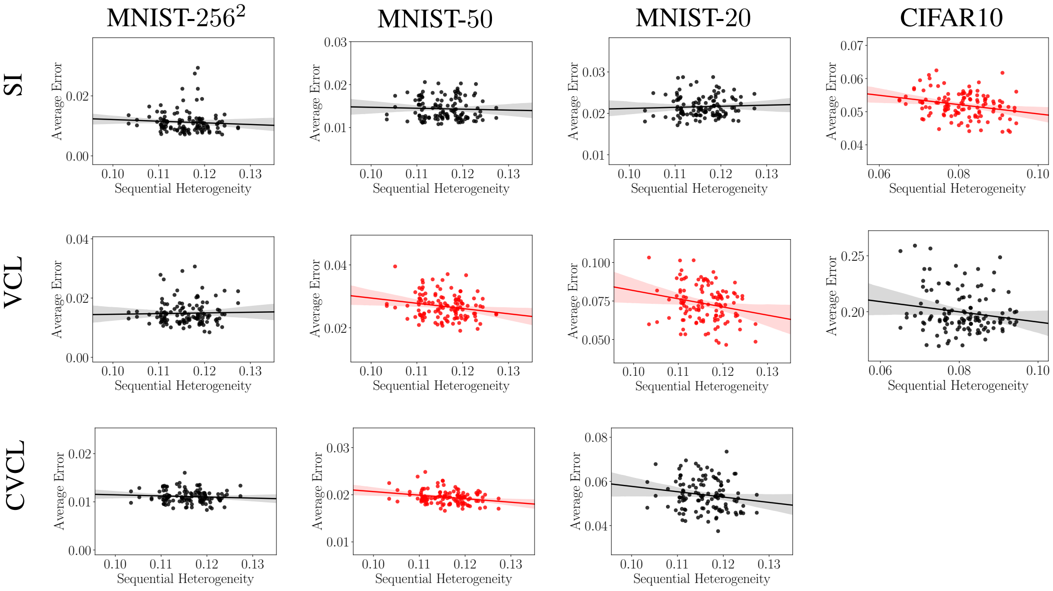

Figure 3: **Normalized sequential heterogeneity vs. average error**, together with the linear regression fit and 95% confidence interval, for each algorithm and test in Table 1(c). Red color indicates statistically significant negative correlations. Black color indicates negligible correlations.

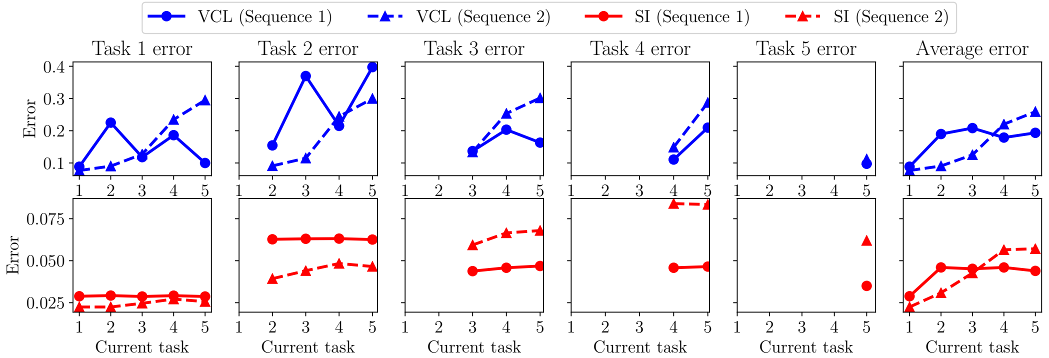

Figure 4: **Details of the error rates** of VCL and SI on two typical task sequences from CIFAR10. Each column shows the errors on a particular task when subsequent tasks are continuously observed. Sequence 1 contains the binary tasks 2/9, 0/4, 3/9, 4/8, 1/2 with sequential heterogeneity 0.091, while sequence 2 contains the tasks 1/2, 2/9, 3/9, 0/4, 4/8 with sequential heterogeneity 0.068 (the labels are encoded to 0, 1, . . . , 9 as usually done for this dataset). For both algorithms, the final average errors (the last points in the right-most plots) on sequence 2 are higher than those on sequence 1, despite sequence 1's higher sequential heterogeneity.

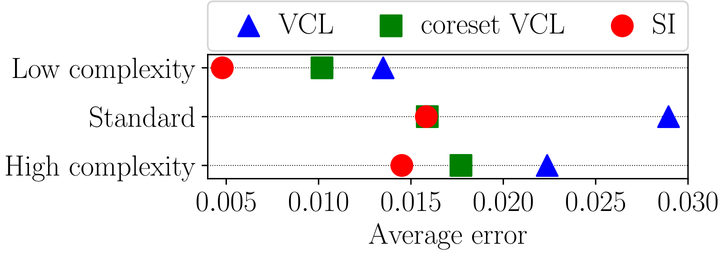

Figure 5: **Average error rates of VCL, CVCL and SI** on 3 task sequences from MNIST with different complexity levels. The high complexity sequence contains the binary tasks 0/1, 2/5, 3/5, 2/3, 2/6 with total complexity 0.48, while the low complexity sequence contains the tasks 0/1, 1/8, 1/3, 1/5, 7/8 with total complexity 0.35. The standard sequence contains the common split 0/1, 2/3, 4/5, 6/7, 8/9 with total complexity 0.41.

B Appendix: More detailed discussions

On total complexity. The strong positive correlations between error rate and total complexity found in our analysis show that task complexity is an important factor in determining the effectiveness of continual learning algorithms. However, this factor is usually not taken into consideration when designing new algorithms or benchmarks. We suggest that task complexity is explicitly considered to improve algorithm and benchmark design. For example, different transfer methods can be used depending on whether one transfers from an easy task to a hard one or vice versa, rather than using a single transfer technique across all task complexities, as currently done in the literature. Similarly, when designing new benchmarks for continual learning, it is also useful to provide different complexity structures to test the effectiveness of continual learning algorithms on a broader range of scenarios and difficulty levels.

To illustrate the usefulness of comparing on various benchmarks, we construct two split MNIST sequences, one of which has high total complexity while the other has low total complexity. The sequences are constructed by starting with the binary classification task 0/1 and greedily adding tasks that have the highest (or lowest) complexity $C(t)$. Fig. 5 shows these sequences and the error rates of VCL, CVCL and SI when evaluated on them. We also show the error rates of the algorithms on the standard split MNIST sequence for comparison. From the figure, if we only compare on the standard sequence, we may conclude that CVCL and SI have the same performance. However, if we consider the other two sequences, we can see that SI is in fact slightly better than CVCL. This small experiment suggests that we should use various benchmarks, ideally with different levels of complexity, for better comparison of continual learning algorithms.

It is also worth noting that although the correlation between error rate and task complexity seems trivial, we are still not very clear which definition of task sequence complexity would be best to explain catastrophic forgetting (i.e., to give the best correlations). In this paper, we propose the first measure for this purpose, the total complexity.

On sequential heterogeneity. The weak or negative correlations between error rate and sequential heterogeneity found in our analysis show an interesting contradiction to our intuition on the relationship between catastrophic forgetting and task dissimilarity. We emphasize that in our context, the weak and negative correlations are not a negative result, but actually a *positive* result. In fact, some previous work showed that task similarity helps improve performance in the context of transfer learning [2, 3, 21], while some others claimed that task dissimilarity could help continual learning [8] although their discussion was more related to the permuted MNIST setting. Our finding gives evidence that supports the latter view in the split MNIST and split CIFAR10 settings.

To identify possible causes of this phenomenon, we carefully analyze the changes in error rates of VCL and SI on CIFAR10 and observe some issues that may cause the negative correlations. For illustration, we show in Fig. 4 the detailed error rates of these algorithms on two typical task sequences where the final average error rates do not conform with the sequential heterogeneity. Both of these sequences have the same total complexity, with the first sequence having higher sequential heterogeneity.

From the changes in error rates of VCL in Fig. 4, we observe that for the first sequence, learning a new task would cause forgetting of its immediate predecessor task but could also help a task learned before that. For instance, learning task 3 and task 5 increases the errors on task 2 and task 4 respectively, but helps reduce errors on task 1 (i.e., backward transferring to task 1). This observation suggests that the dissimilarities between only consecutive tasks may not be enough to explain catastrophic forgetting, and thus we should take into account the dissimilarities between a task and all the previously learned tasks.

From the error rates of SI in Fig. 4, we observe a different situation. In this case, catastrophic forgetting is not severe, but the algorithm tends not to transfer very well on the second sequence. This inability to transfer leads to higher error rates on tasks 3, 4, and 5 even when the algorithm learns them for the first time. One possible cause of this problem could be that a fixed regularization strength $\lambda = 1$ is used for all tasks, making the algorithm unable to adapt to new tasks well. This explanation suggests that we should customize the algorithm (e.g., by tuning the λ values or the optimizer) for effectively transferring between different pairs of tasks in the sequence.

Future directions. The analysis offered by our paper provides a general and novel methodology to study the relationship between catastrophic forgetting and properties of task sequences. Although the two measures considered in our paper, total complexity and sequential heterogeneity, can explain some aspects of catastrophic forgetting, the correlations in Table 1 are not very strong (i.e., their coefficients are not near 1 or -1). Thus, they can still be improved to provide better explanations for the phenomenon. Besides these two measures, we can also design other measures for properties such as intransigence [5].

References

[1] Alessandro Achille, Tom Eccles, Loic Matthey, Chris Burgess, Nicholas Watters, Alexander Lerchner, and Irina Higgins. Life-long disentangled representation learning with cross-domain latent homologies. In *Advances in Neural Information Processing Systems*, pages 9895–9905, 2018.

[2] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. Task2Vec: Task embedding for meta-learning. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[3] Haitham Bou Ammar, Eric Eaton, Matthew E Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. An automated measure of MDP similarity for transfer in reinforcement learning. In *AAAI Conference on Artificial Intelligence Workshops*, 2014.

[4] Thang D Bui, Cuong V Nguyen, Siddharth Swaroop, and Richard E Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. *arXiv:1811.11206*, 2018.

[5] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision*, 2018.

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019.

[7] Harrison Edwards and Amos Storkey. Towards a neural statistician. In *International Conference on Learning Representations*, 2017.

[8] Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv:1805.09733*, 2018.

[9] Siavash Golkar, Michael Kagan, and Kyunghyun Cho. Continual learning via neural pruning. *arXiv:1903.04476*, 2019.

[10] Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *International Conference on Learning Representations*, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *AAAI Conference on Artificial Intelligence*, 2018.

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017.

[14] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4652–4662, 2017.

[15] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

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

[17] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

[18] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

[19] Mark B Ring. CHILD: A first step towards continual learning. *Machine Learning*, 28(1):77–104, 1997.

[20] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured Laplace approximations for overcoming catastrophic forgetting. *arXiv:1805.07810*, 2018.

[21] Sebastian Ruder and Barbara Plank. Learning to select data for transfer learning with Bayesian optimization. In *Conference on Empirical Methods in Natural Language Processing*, pages 372–382, 2017.

[22] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv:1606.04671*, 2016.

[23] Jeffrey C Schlimmer and Douglas Fisher. A case study of incremental concept induction. In *AAAI Conference on Artificial Intelligence*, volume 86, pages 496–501, 1986.

[24] Jonathan Schwarz, Jelena Luketina, Wojciech M Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, 2018.

[25] Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, 2018.

[26] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[27] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

[28] Shagun Sodhani, Sarath Chandar, and Yoshua Bengio. On training recurrent neural networks for lifelong learning. *arXiv:1811.07017*, 2018.

[29] Stefan Stojanov, Samarth Mishra, Ngoc Anh Thai, Nikhil Dhanda, Ahmad Humayun, Chen Yu, Linda B Smith, and James M Rehg. Incremental object learning from contiguous views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8777–8786, 2019.

[30] Siddharth Swaroop, Cuong V Nguyen, Thang D Bui, and Richard E Turner. Improving and understanding variational continual learning. In *Continual Learning Workshop @ NeurIPS*, 2018.

[31] Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[32] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

[33] Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[34] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995, 2017.