Proceedings of Machine Learning Research 157, 2021 ACML 2021

Lifelong Learning with Sketched Structural Regularization

Haoran Li HLI143@JHU.EDU
Aditya Krishnan AKRISH23@JHU.EDU
Jingfeng Wu UUUJF@JHU.EDU
Johns Hopkins University, Baltimore, MD 21218, USA

Soheil Kolouri SOHEIL.KOLOURI@VANDERBILT.EDU
Vanderbilt University, Nashville, TN 37235, USA

Praveen K. Pilly PKPILLY@HRL.COM
HRL Laboratories, LLC, Malibu, CA 90265, USA

Vladimir Braverman VOVA@CS.JHU.EDU
Johns Hopkins University, Baltimore, MD 21218, USA

Editors: Vineeth N Balasubramanian and Ivor Tsang

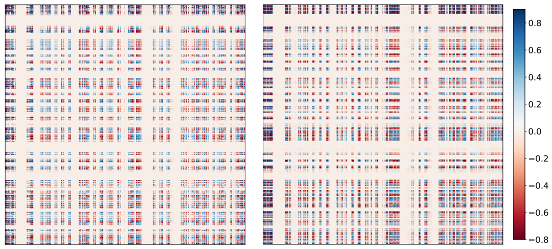
Abstract

Preventing catastrophic forgetting while continually learning new tasks is an essential problem in lifelong learning. Structural regularization (SR) refers to a family of algorithms that mitigate catastrophic forgetting by penalizing the network for changing its "critical parameters" from previous tasks while learning a new one. The penalty is often induced via a quadratic regularizer defined by an *importance matrix*, e.g., the (empirical) Fisher information matrix in the Elastic Weight Consolidation framework. In practice and due to computational constraints, most SR methods crudely approximate the importance matrix by its diagonal. In this paper, we propose *Sketched Structural Regularization* (Sketched SR) as an alternative approach to compress the importance matrices used for regularizing in SR methods. Specifically, we apply *linear sketching methods* to better approximate the importance matrices in SR algorithms. We show that sketched SR: (i) is computationally efficient and straightforward to implement, (ii) provides an approximation error that is justified in theory, and (iii) is method oblivious by construction and can be adapted to any method that belongs to the SR class. We show that our proposed approach consistently improves various SR algorithms' performance on both synthetic experiments and benchmark continual learning tasks, including permuted-MNIST and CIFAR-100.

Keywords: Lifelong learning, sketching, online learning, machine learning

1. Introduction

Lifelong learning, also termed as *continuous learning* or *incremental learning*, is the ability to continually learn in a varying environment through integrating the newly acquired knowledge while maintaining the previously learned experiences (Parisi et al., 2019). A central issue that prevents the state-of-the-art machine learning models (e.g. deep neural networks) from achieving lifelong learning is *catastrophic forgetting*, i.e. learning a new task may severely modify the model parameters, including those that are important to the previous tasks (Parisi et al., 2019).



(a) Full EF (b) Sketched EF

Frobenius-Norm Approx. Error $\|\widetilde{\Omega} - \Omega\|_F^2 / \|\Omega\|_F^2$	
Diagonal $94.7\% \pm 0.6\%$	Block-Diagonal $82.8\% \pm 1.3\%$
Sketched $\mathbf{8.1\% \pm 10.2\%}$	Low-Rank $0.0\% \pm 0.0\%$

Figure 1: Illustration of the (sketched) empirical Fisher on a synthetic 2D binary classification task from Pan et al. (2020). The figures show the heat map of the full empirical Fisher and the sketched empirical Fisher with $t = 50$; the table shows the approximation error of each methods to the full empirical Fisher. The plots and the table suggest that: (i) the full empirical Fisher cannot be well-approximated by its diagonal or block-diagonal, (ii) sketching method can utilize the off-diagonal entries to obtain a better approximation, and (iii) though low-rank method (with 50 ranks) also leads to a good approximation, the computational cost is not affordable in practice. See Section 5.1 for more details.

Structural regularization (SR), or *selective synaptic plasticity*, is a general and widely-adopted paradigm to mitigate catastrophic forgetting in lifelong learning (Kolouri et al., 2020; Aljundi et al., 2018; Kirkpatrick et al., 2017; Chaudhry et al., 2018; Zenke et al., 2017). From a geometric perspective (Kolouri et al., 2020; Chaudhry et al., 2018), SR methods construct an (positive semi-definite) *importance matrix* (IM) that measures the relative importance of the model parameters to the old tasks (which are aimed be preserved in lifelong learning) and add a quadratic regularizer defined by the IM when training on new tasks. The intuition behind SR is clear: the quadratic regularizer adaptively penalizes parameters from changing according to their criticality measured by the IM. As a result, SR encourages the model to fit to the new task using non-critical parameters so that it is able to preserve important information from old tasks. For example, Kirkpatrick et al. (2017) choose the (diagonal) *empirical Fisher information matrix*[1] (EF) as the IM in their seminal algorithm, *Elastic Weight Consolidation* (EWC) (Kirkpatrick et al., 2017; Kolouri et al., 2020; Chaudhry et al., 2018). However, a full IM (e.g. EF) scales as $\mathcal{O}(m^2)$ for a model with m parameters and can be prohibitively big to use in large-scale lifelong learning models. Often in practice, the diagonal, which scales as $\mathcal{O}(m)$, is used as a crude approximation to the full IM (Kirkpatrick et al., 2017; Kolouri et al., 2020; Aljundi et al., 2018). We refer to SR with an IM approximated by the diagonal by *diagonal SR*.

While exploring new and effective importance matrices has been a hot direction for SR (Kolouri et al., 2020; Aljundi et al., 2018; Kirkpatrick et al., 2017; Chaudhry et al., 2018; Zenke et al., 2017), little effort has been spent on examining the effectiveness of the crude

1. In their original paper (Kirkpatrick et al., 2017) (and follow-up papers, e.g., (Kolouri et al., 2020)), the importance matrix in EWC is referred as the "Fisher Information Matrix", but precisely, it should be called the "Empirical Fisher" — the two terms are often used interchangeable in the community, though they are not identical. See (Kunstner et al., 2019) for a detailed clarification.

diagonal approximation (a few exceptions, e.g. (Liu et al., 2018; Ritter et al., 2018), are discussed later in Section 2). Intuitively speaking, a diagonal IM assumes independence between parameters, which is far from reality (Liu et al., 2018; Ritter et al., 2018). In mathematics, the diagonal is generally not a good approximator to a positive semi-definite matrix — the only non-trivial exception to our knowledge is when the matrix is diagonally dominant (Horn and Johnson, 2012). Unfortunately, for the importance matrices considered in SR methods this might not be the case, especially when training using neural networks. As an illustration, we examine the empirical Fisher (EF) as the IM (Kirkpatrick et al., 2017) of a synthetic experiment from Pan et al. (2020); the full EF is shown in Figure 1. The plot shows that the full EF is far from its diagonal; in fact the diagonal only contributes to less than 5.3% of the Frobenius norm of the EF matrix (see table in Figure 1). Hence, approximating IM with its diagonal might be problematic. A natural question then is:

Is there a computational and memory efficient method to approximate the importance matrix without losing critical information in the matrix?

Our Contributions. In this paper, we answer the above question by providing a *linear sketching method* (Charikar et al., 2002) as a *provable*, *ubiquitous*, *efficient* and *effective* approach to approximate the importance matrix in SR methods. Specifically, in one pass of the data (which is also required for diagonal approximation), a $\mathcal{O}(tm)$ size sketched matrix can be produced that approximately recovers the quadratic regularizer defined by the $\mathcal{O}(m^2)$ size importance matrix. Here, $t \ll m$ is a tuneable parameter that balances the computation cost and matrix size with the quality of approximation, and can be chosen as a small number in practice. Our method, called *sketched SR*, has the following notable advantages:

1. Has a *theoretically guaranteed* small approximation error, provided the importance matrix has a well-behaved spectrum, e.g. has low effective rank. Fortunately, for deep neural network and commonly used SR methods, the importance matrix (e.g. EF) does have low (effective) rank (Sagun et al., 2017; Chaudhari and Soatto, 2018), but is far from being diagonal (see Figure 1).

2. Is *algorithm oblivious* by construction, i.e. for any algorithm that belongs to the SR framework (defined in Section 3), a sketched version can be readily established without additional, algorithm specific considerations.

3. Is *computationally efficient* and *easy to implement*. Both sketched SR and diagonal SR make only one pass of the data (of the old task) to obtain the approximation. Though sketched SR saves $\mathcal{O}(tm)$ parameters, which is slightly larger than the $\mathcal{O}(m)$ parameters in diagonal SR. This additional cost is easily affordable as setting $t \leq 50$ is sufficient for sketched SR to outperform diagonal SR in our experiments.

4. *Consistently outperforms* its diagonal counterpart on overcoming catastrophic forgetting, in both synthetic experiments and benchmark lifelong-learning tasks, including permuted-MNIST and CIFAR-100.

2. Related Works

Functional Regularization. Besides SR, another popular category of approaches to overcome catastrophic forgetting is *functional regularization* (FR) (Jung et al., 2016; Li and Hoiem, 2017; Rannen et al., 2017; Shin et al., 2017; Hu et al., 2018; Rozantsev et al., 2018; Wu et al., 2018; Li et al., 2019). Similar to SR, FR also adds regularizer (when training new tasks) to penalize the forgetting of useful old knowledge; however, FR may use very general (hence, functional) regularizers, in addition to quadratic ones. For example, Jung et al. (2016); Li and Hoiem (2017) snapshot a teacher model learned from old tasks, and use it to regularize a student model that fits new tasks. Moreover, generative models are applied to generate pseudo-data (*memory*) from old tasks, and the pseudo-data is mixed to the new data distribution as a regularization (*replay*) for learning new tasks (Rannen et al., 2017; Rostami et al., 2019; Shin et al., 2017; Wu et al., 2018; Hu et al., 2018). This is also known as *memory replay*. Finally, we remark that FR can be used together with SR (Shin et al., 2017; Rozantsev et al., 2018). Our focus of this paper is to use linear sketching methods to improve SR methods; an interesting future work is to apply similar ideas (e.g., coresets (Feldman and Langberg, 2011; Har-Peled and Mazumdar, 2004)) to improve FR methods, especially for those based on memory replay.

Non-Diagonal Importance Matrix. Diagonal approximation is a crude, but de facto approach to compress the full IM in most of existing SR algorithms (Kolouri et al., 2020; Aljundi et al., 2018; Kirkpatrick et al., 2017; Chaudhry et al., 2018; Zenke et al., 2017). Before this paper, there are a few works that study SR with non-diagonal IM (Liu et al., 2018; Ritter et al., 2018), which we discuss in sequence. Ritter et al. (2018) adopt the layer-wise block-diagonal approximation as a better replacement to the commonly used diagonal version for the IM: even so, the cross-layer weight dependence is being ignored; moreover, in our experiments, block-diagonal EF is not a good approximation to EF, either (see Figure 1). Liu et al. (2018) propose layer-wise rotation of the EF such that the new matrix can be more diagonal-alike; this procedure not only assumes cross-layer independence (of weights), but even assumes independence between layer inputs and layer gradients (see Eq. (7) in Liu et al. (2018)). In comparison, the sketching methods adopted in this paper only require a very weak assumption, i.e., the IM has low effective rank.

Linear Sketching. Linear sketching is a widely studied technique for dimensionality reduction. We rely on the popular sketching method *CountSketch* (Charikar et al., 2002) that has its roots in the Johnson-Lindenstrauss transform. Randomized linear sketching methods, such as CountSketch, draw a random matrix $S \in \mathbb{R}^{t \times m}$ and embed the columns of the input matrix $W \in \mathbb{R}^{n \times m}$ into a smaller dimension $t \ll n$ by outputting SW. By carefully constructing the random distribution, it can be shown that the sketch SW *preserves the norms of the vectors in the subspace spanned by the columns of* W up to some error. Such sketching techniques are known as oblivious subspace embeddings (OSEs). This property of OSEs makes them a natural tool for approximating the quadratic regularizer in SR methods.

Sparse OSE methods (Nelson and Nguyên, 2013; Cohen, 2016) such as CountSketch have a two-fold advantage: i) they're *oblivious*, which means that the random distribution is defined independent of the input matrix W and ii) the sketch SW can be computed in time that is linear in the input size (e.g. proportional to the number of non-zero entries in W). These methods have been widely used, giving fast algorithms for various problems such as low-rank approximation, linear regression (Sarlos, 2006; Clarkson and Woodruff, 2017; Meng and Mahoney, 2013), k-means clustering (Cohen et al., 2015), leverage score estimation (Drineas et al., 2012) and numerous other problems (Lee et al., 2019; Ahle et al., 2020; van den Brand et al., 2021).

3. Preliminaries

We use $(x, y) \in \mathbb{R}^s \times \mathbb{R}^k$ to denote a feature-label pair, and $\theta \in \mathbb{R}^m$ to denote the model parameter. A parametric model is denoted by $\phi(\cdot\,; \theta) : \mathbb{R}^s \to \mathbb{R}^k$. Given a distance measure of two distributions, $d(\cdot, \cdot)$, the individual loss over data point (x, y) can be formulated as

$$\ell(x, y; \theta) := d(\phi(x; \theta), y).$$

For example, in deep neural networks, $\phi(\cdot\,; \theta)$ is the network output, and $d(\cdot, \cdot)$ is usually chosen to be the cross entropy loss (Goodfellow et al., 2016).

Structural Regularization. Let task A with data distribution $(x, y) \sim \mathcal{D}_A$ be an already well-learned task on network ϕ with learnt parameters θ^*_A. In order to overcome catastrophic forgetting when learning a new task B, with data distribution $(x, y) \sim \mathcal{D}_B$, structural regularization (SR) algorithms apply an extra regularizer $\mathcal{R}(\theta)$ to the main loss and optimize the following objective:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_B} \left[\ell(x, y; \theta)\right] + \lambda \cdot \mathcal{R}(\theta).$$

Here, the expectation should be understood as the empirical expectation over the training set. As for the regularization term, λ is a hyper-parameter, and $\mathcal{R}(\theta)$ is a quadratic regularizer that penalizes the weight for being deviated from θ^*_A, the learnt weight from the previous task A:

$$\mathcal{R}(\theta) := \frac{1}{2}(\theta - \theta^*_A)^\top \Omega (\theta - \theta^*_A),$$

where $\Omega \in \mathbb{R}^{m\times m}$ is an importance matrix and is PSD. As we will see shortly, the PSD matrix Ω considered in SR usually has a natural decomposition as (Kirkpatrick et al., 2017; Aljundi et al., 2018):

$$\Omega = \frac{1}{n} W^\top W, \tag{1}$$

where each row of $W \in \mathbb{R}^{n\times m}$ is a Jacobian matrix of a certain individual loss (which might not be the one used for the main loss) of data x from task A, and n is the number of training data in task A. Then, the structural regularizer $\mathcal{R}(\theta)$ can be written as

$$\mathcal{R}(\theta) = \frac{1}{2n}\|W \cdot (\theta - \theta^*_A)\|_2^2, \quad W \in \mathbb{R}^{n\times m}. \tag{2}$$

Examples. Table 1 summarizes three examples for the importance matrices in: *Elastic Weight Consolidation* (EWC) (Kirkpatrick et al., 2017), *Memory Aware Synapses* (MAS) (Aljundi et al., 2018), and *Sliced Cramer Plasticity* (Kolouri et al., 2020). It is worth noting that the importance matrix used in EWC is the *Empirical Fisher* (EF) evaluated at the optimal weight for task A.

Table 1: The construction of the importance matrices in EWC (Kirkpatrick et al., 2017), MAS (Aljundi et al., 2018), and SCP (Kolouri et al., 2020). In EWC, the loss function $\ell(x, y; \theta) := d(\phi(x; \theta), y)$ is defined with $d(\cdot, \cdot)$ being the cross entropy loss. In SCP, $\{\xi_l\}$ are i.i.d. random unit vectors.

	Matrix Ω	Row-Vector $(W)_x$
EWC	$\mathbb{E}_{(x,y)\sim\mathcal{D}_A} \nabla_\theta \ell(x, y; \theta_A^*) \cdot \nabla_\theta \ell(x, y; \theta_A^*)^\top$	$\nabla_\theta \ell(x, y; \theta_A^*)$
MAS	$\mathbb{E}_{x\sim\mathcal{D}_A} \big(\nabla_\theta \Vert\phi(x; \theta_A^*)\Vert_2^2\big) \cdot \big(\nabla_\theta \Vert\phi(x; \theta_A^*)\Vert_2^2\big)^\top$	$\nabla_\theta \Vert\phi(x; \theta_A^*)\Vert_2^2$
SCP	$\frac{1}{L} \sum_{l=1}^{L} \mathbb{E}_{x\sim\mathcal{D}_A} \big(\nabla_\theta \xi_l \phi(x; \theta_A^*)\big) \cdot \big(\nabla_\theta \xi_l \phi(x; \theta_A^*)\big)^\top$	$\{\nabla_\theta \xi_l \phi(x; \theta_A^*)\}_{l=1}^{L}$

Diagonal Approximation. Unfortunately, both matrices Ω and W have m^2 and mn entries respectively, which makes them prohibitively large to compute and store for big models like deep neural networks. As a compromise, practitioners often take the diagonal of Ω as an approximation. This leads to the presented version of EWC, MAS, and SCP in their original paper. These are called *diagonal EWC* (respectively, *diagonal* MAS, SCP) in this paper to be distinguishing with our variants. However, as have discussed and demonstrated on a synthetic dataset, such a treatment ignores the dependence between weights and exacerbates performance degeneration for overcoming catastrophic forgetting. In the following we present our sketched version of the above algorithms, which can make use of the off-diagonal entries of Ω to improve the diagonal approximated version.

4. Sketched Structural Regularization

In this section we propose our framework of sketching the regularizer from (2) and describe the specific sketch construction along with some theoretical guarantees. We describe our construction in terms of the general framework of structural regularization for lifelong learning from Section 3. Then we contrast our approximation method with other compression methods like PCA. Finally we describe how we go from the two-task settings to an online version of the algorithm in a way that is standard in works on structural regularization (Kolouri et al., 2020; Aljundi et al., 2018; Kirkpatrick et al., 2017).

4.1. Sketched Regularizer

We propose a method to sketch the matrix Ω from (1) by reducing the dimensionality of each of the matrix W from n dimensions to t dimensions for a $t \ll \min\{n, m\}$. Specifically, we draw a random matrix $S \in \mathbb{R}^{t\times n}$ from a carefully chosen distribution and approximate the regularizer (2) in SR methods with

$$\widetilde{\mathcal{R}}(\theta) = \frac{1}{2n} \Vert \widetilde{W} \cdot (\theta - \theta_A^*)\Vert_2^2, \quad \widetilde{W} := SW \in \mathbb{R}^{t\times m}. \tag{3}$$

We use *CountSketch* (Charikar et al., 2002) to construct the sketched matrix $\widetilde{W} = SW$, which is formally presented in Algorithm 1. CountSketch reduces the number of rows (aka, the dimension of the columns) of W by the following: first the rows of W are randomly

Algorithm 1: Sketch Construction in Sketched SR

1: **Input:** Data from task A and optimized neural network $\phi(\cdot\,; \theta_A^*)$ for task A
2: **Parameters:** Size of sketch $t \in \mathbb{N}^+$
3: Initialize 2-wise and 4-wise independent hash functions $h : [n] \to [t]$ and $\sigma : [n] \to \{-1, 1\}$ respectively
4: **for** $k = 1, \ldots, t$ **do**
5: Group data $G_k := \{x \in A : h(x) = k\}$
6: Compute $\sum_{x \in G_k} \sigma(x)(W)_x$ as per Table 1 by auto-differentiation
7: Set $(\widetilde{W})_k \leftarrow \sum_{x \in G_k} \sigma(x)(W)_x$
8: **end for**
9: **return** $\widetilde{W} \in \mathbb{R}^{t \times m}$

partitioned into t groups (Algorithm 1, line 5), then rows in each group are randomly, linearly combined (with random signs as weights) into a single new row (Algorithm 1, line 7).

Two remarks are in order for the practical implementation of Algorithm 1: (i) note that Algorithm 1 only makes one pass of the data from task A, which is as required for computing diagonal approximation; (ii) note that Algorithm 1 requires $\mathcal{O}(t)$ times auto-differentiation, but since t is small and the sketch construction only needs to done once per new task, the cost is affordable in practice (see more in Section 5).

Comparison with Low-Rank Approximation Methods. The main advantage of using CountSketch over more complicated low-rank approximation methods (e.g. PCA) to compress the importance matrix in SR methods, is that it can be computed with only a small amount of additional computation and only a modest blow-up in memory compared to the diagonal approximation. However PCA is usually computational intractable for big models such as deep neural networks. Moreover, in below, we show CountSketch achieves provable small approximation error (for matrix with low stable-rank), as can be guaranteed by PCA.

4.2. Theoretical Properties

The following theorem from Cohen et al. (2016) builds on several results on CountSketch matrices, giving theoretical guarantees for sketching quadratic forms of matrices. The theorem is re-phrased for our purposes, showing the quality of approximation by the sketch in preserving ℓ_2-norms of vectors in the subspace spanned by the columns of W, the matrix that is being sketched. There is a trade-off in the quality of approximation by the sketch and its size, given by the dimension of the columns t. In particular, the error in preserving the ℓ_2-norm of any $W\theta$ depends on the spectrum of W; when $t \geq \|W\|_F^4/(\epsilon^2\|W\|_2^4)$ the error is *additive* and scales with $\epsilon\|W\|_2^2\|\theta\|_2^2$, which we detail in the following theorem. We state a full-version of the theorem showing the exact trade-off between the the number of buckets t and the quality of approximation in Appendix A.1.

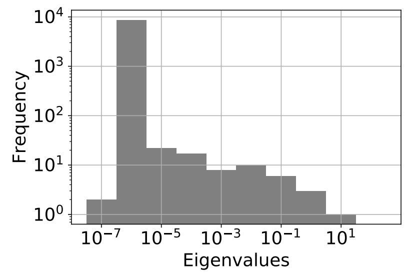


Figure 2: The spectrum of the EF studied in Figure 1. The EF is $8,770 \times 8,770$ and its stable rank is 1.26 ± 0.13 (over 5 random seeds). Both the plotted spectrum and the computed stable rank show that this EF has very small effective rank.

Theorem 1. *For a matrix $W \in \mathbb{R}^{n \times m}$ with stable rank[2] r, a CountSketch matrix $S \in \mathbb{R}^{t \times n}$ with $t = \mathcal{O}(r^2/\epsilon^2)$ has the property that for all $\theta \in \mathbb{R}^m$,*

$$\left| \|SW\theta\|_2^2 - \|W\theta\|_2^2 \right| \leq \epsilon \cdot \|W\|_2^2 \cdot \|\theta\|_2^2$$

with probability at least 0.99.

Notice that stable rank never exceeds the usual rank, and can be significantly smaller when the matrix has a decaying spectrum. The importance matrix considered in SR methods usually have fast decaying spectrum (see Figure 2), i.e., small stable rank, making it effective to use CountSketch to approximate quadratic forms with the matrices. For instance, in the synthetic experiment we considered, the stable rank of the EF shown in Figure 1 is 1.26 with standard deviation 0.13, measured over 5 trials. Note that the EF is $8,770 \times 8,770$.

4.3. Online Extension of Sketched SR.

Lifelong learning often requires learning more than two tasks sequentially. One method of extending the Sketched SR method to learn on multiple tasks to maintain separate sketches for each task and compute the regularizer $\widetilde{\mathcal{R}}(\theta)$ in (3) from each of the previous tasks when learning the current one. This approach would cause the memory requirement to grow linearly in the number of tasks and can become a bottleneck in scaling the method. A standard way to tackle this in works on structural regularization is to apply the *moving average* method to aggregate the histories (Chaudhry et al., 2018; Schwarz et al., 2018). Specifically, let $\widetilde{\Omega}_{\tau-1}$ be the importance matrix maintained after training on the $(\tau-1)$-th task, then, given the (approximate) importance matrix $\widetilde{\Omega}$ outputted on the data from task τ, the histories are updated as

$$\widetilde{\Omega}_\tau \leftarrow \alpha\widetilde{\Omega} + (1-\alpha)\widetilde{\Omega}_{\tau-1} \tag{4}$$

where $\alpha \in (0, 1]$ is a hyperparameter.

Since the matrix $\widetilde{\Omega}$ is a diagonal matrix for each task in the aforementioned methods, computing the sum from (4) is straightforward. Sketched SR, however, doesn't explicitly

2. The stable rank of a matrix W is $\|W\|_F^2/\|W\|_2^2$.

compute the matrix $\widetilde{\Omega} = \widetilde{W}^\top \widetilde{W}$, hence we cannot hope to compute the matrix $\widetilde{\Omega}_\tau$ defined by the sum in (4). We propose the following method: let $\widetilde{W}_{\tau-1}$ be the maintained sketch after training on the $(\tau-1)$-th task, then, given the weight θ^* and the sketch $\widetilde{W}$ outputted on the data from task τ, we update the importance matrix as

$$\widetilde{W}_\tau \leftarrow \sqrt{\alpha}\widetilde{W} + \sqrt{1-\alpha}\widetilde{W}_{\tau-1}. \tag{5}$$

When learning on task $\tau+1$ we use the regularizer

$$\widetilde{\mathcal{R}}_\tau(\theta) := \frac{1}{2n}\|\widetilde{W}_\tau(\theta-\theta^*)\|_2^2. \tag{6}$$

A priori, it is not clear why the regularizer from (6) is a good approximation to that induced by the importance matrix from (4). We give a theorem, along with a proof in Appendix A.2, that shows that for *all* $\theta \in \mathbb{R}^m$ *simultaneously*, the regularizer given by (6) is close to that induced by the importance matrix $\widetilde{\Omega}_\tau$ from (4).

Theorem 2. *Let $W_1, \ldots, W_\tau \in \mathbb{R}^{n\times m}$ be a sequence of matrices, $\alpha_1, \ldots, \alpha_\tau \geq 0$ be a sequence of weights , and $S_1, \ldots, S_\tau \in \mathbb{R}^{t\times n}$ be a sequence of independent CountSketch matrices with sketch size $t \cdot (\tau/\epsilon)^2$ for constants $t \in \mathbb{N}^+$ and $\epsilon \in (0, 1/2)$. There exists a constant $C > 0$ such that for all vectors $\theta \in \mathbb{R}^m$ simultaneously, with probability at least* 0.99*,*

$$\widetilde{\mathcal{R}}\tau(\theta) = \mathcal{R}_\tau(\theta) \pm \epsilon \cdot \|\theta-\theta^*\|_2^2 \cdot \max_{i\in[\tau]}\left(\|W_i\|_2^2 + \frac{C\|W_i\|_F^2}{\sqrt{t}}\right).$$

Here $\widetilde{\mathcal{R}}_\tau(\theta)$ is as defined in (6) and $\mathcal{R}_\tau(\theta)$ is the exact online regularizer $\sum_{i=1}^{\tau}(\theta-\theta^*)^\top\Omega_i(\theta-\theta^*)$ given by the importance matrices $\{\Omega_i\}$ for each task.

5. Experiments

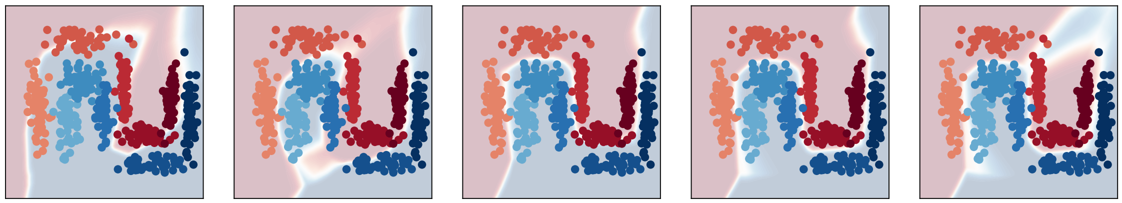
In this section, we present empirical evidence that verifies the effectiveness of our proposed Sketched SR methods. All the reported numerical results are averaged over 5 runs with different random seeds.

5.1. Synthetic Experiments

We start with a series of synthetic experiments.

Experimental Setup. We first consider a synthetic 2D binary classification task from Pan et al. (2020). The experiment consists of 5 classification tasks learnt sequentially using the regularization induced by each of EWC with a small multi-layer perceptron. The network has 8,770 parameters. For the regularization matrix induced by EWC, we compare the performance of various approaches to approximating the matrix including:

(i) a diagonal approximation;
(ii) a block-diagonal approximation, with a sequence of 50×50 non-zero blocks along the diagonal;
(iii) sketched approximation with sketch size $t = 50$;
(iv) a rank-50 SVD;



(a) Diagonal (b) Block-Diag. (c) Sketched (d) Low-Rank (e) Full EWC

Average Acc.	Diagonal	Block Diagonal	Sketched	Low-Rank	Full IM
EWC	$88.0 \pm 7.4\%$	$79.3 \pm 9.8\%$	$92.1 \pm 6.8\%$	$93.9 \pm 3.9\%$	$95.7 \pm 5.4\%$

Figure 3: Variants of EWC (Kirkpatrick et al., 2017) on a synthetic 2D binary classification task from (Pan et al., 2020). The figures show the decision boundaries found by the compared algorithms between the two classes represented by red and blue. The table shows the average accuracy across all tasks (after learning the final task) for the compared algorithms. The plots and the table suggest that: sketched SR has a higher average accuracy than both diagonal SR and block-diagonal SR by overcoming catastrophic forgetting; while average accuracy of low-rank SR and full SR is higher, they requires significantly more computation which is not affordable in practice.

(v) and the full importance matrix.

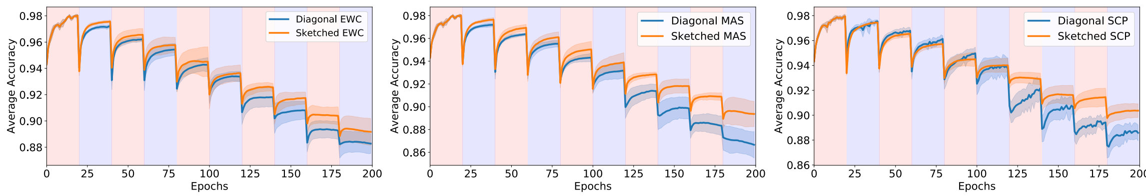
For all algorithms, we use ADAM as the optimizer with learning rate 10^{-3}, and use the moving average parameter $\alpha = 0.5$. For more details please see Appendix B.1.

Approximate vs. Full Matrix Comparison. We first plot the EF (the importance matrix in EWC methods) and the sketched EF in Figure 1. The EF is obtained with the optimal weight that fits the first four tasks and the sketched EF uses sketch size $t = 50$. From the figure we observe that the EF cannot be well-approximated by its diagonal or block-diagonal; moreover, the sketched EF can utilize the off-diagonal entries to generate a better approximation. This is further supported by the numerical approximation error shown in the table within Figure 1. Note that while the low-rank method can offer a better approximation, it is not computationally efficient in practice.

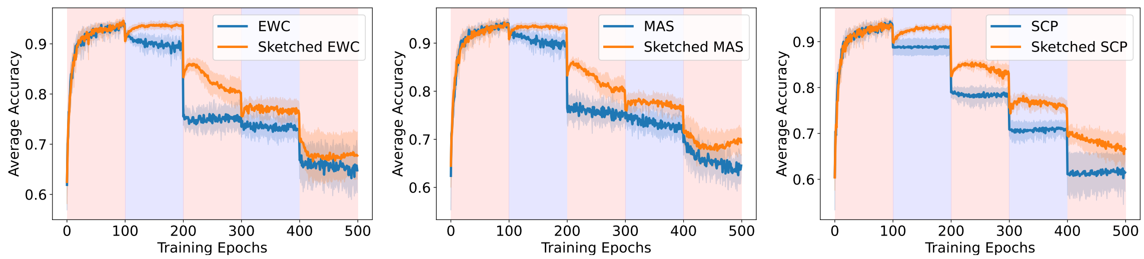
Performance of the Compared Algorithms. We then compare the performance of each algorithms in Figure 3. The plots indicates that sketched methods are more effective than diagonal methods for overcoming catastrophic forgetting. Additionally, while low-rank and full EWC perform better than sketched EWC, they are not computationally feasible in practical settings with large models.

5.2. Sketched versus Diagonal Regularization

In this section we demonstrate the effectiveness of our sketched methods on various SR algorithms. Three representative SR algorithms are examined: EWC (Kirkpatrick et al., 2017), MAS (Aljundi et al., 2018), and SCP (Kolouri et al., 2020).



(a) Permuted-MNIST



(b) CIFAR-100

Figure 4: The averaged accuracy (over all tasks) of sketched EWC (resp. MAS, SCP) vs. diagonal EWC (resp. MAS, SCP) on permuted-MNIST and CIFAR-100. For the sketched methods, we set $t = 50$. The plots suggest that sketched methods consistently outperform their diagonal counterparts for overcoming catastrophic forgetting.

Datasets. We perform our experiments on two continual learning benchmarks: Permuted-MNIST and CIFAR-100 Distribution Shift. The permuted-MNIST experiment is a basic benchmark for lifelong learning (Kirkpatrick et al., 2017; Zenke et al., 2017; Rostami et al., 2019; Ritter et al., 2018; Ramasesh et al., 2021). Each of the 10 permuted-MNIST tasks is a 10-classes classification task based on a permutation of MNIST dataset, where the pixels in each figure are permuted according to certain rule (to be more specific, the permutation rule is same within a task but random across different tasks). The 5-task *CIFAR-100 Distribution Shift* dataset introduced in Ramasesh et al. (2021) is a more challenging dataset. The main difference from the split CIFAR experiment commonly used in the literature (see, e.g., (Zenke et al., 2017)) is that the *CIFAR-100 Distribution Shift* only requires single-head neural network without task-specific information for classifying classes of each task.

Experimental Setup. We conduct our experiments on both datasets in a single-head setting, where the task ID is unknown (Chaudhry et al., 2018), which is considered the more challenging and practical evaluation mode comparing to multi-head evaluation. In our permuted-MNIST experiment, we use a multi-layer perceptron with the architecture $784 \to 1024 \to 512 \to 256 \to 10$ as the classifier, and ADAM as the optimizer. In the CIFAR-100 Distribution Shift experiment, we used a Wide-ResNet (Zagoruyko and Komodakis, 2016) as our backbone. For each compared algorithm, the regularization coefficient λ is chosen to be optimal by grid search. For more details, please see Appendix B.2.2.

Performance of the Compared Algorithms. Figure 4 shows the average accuracy across previously learned tasks after each epoch of training for the compared methods. Ta-

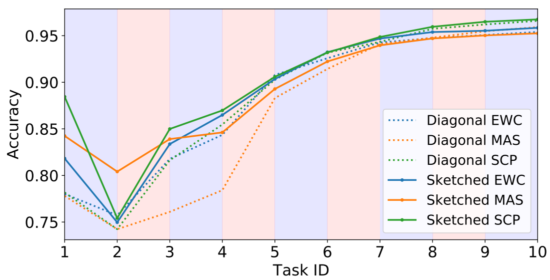


Figure 5: The accuracy of each task (after training on all tasks) of sketched methods vs. diagonal methods on permuted-MNIST.

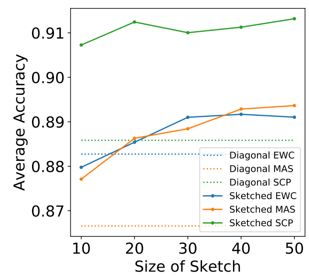


Figure 6: Effect of the sketch size (t) on the average accuracy of sketched methods for learning permuted-MNIST tasks.

ble 2 reports the averaged accuracy (across all tasks) of the compared algorithms. From the figures and the table, we consistently see that sketched SR methods outperform their diagonal counterparts in all three SR regimes, in terms of overcoming catastrophic forgetting.

Table 2: The average accuracy (over all tasks) of sketched SR and diagonal SR methods on Permuted-MNIST and CIFAR-100. For sketched SR methods we set $t = 50$.

Dataset	Regime	Diagonal	Sketched
Permuted-MNIST	EWC	88.3±0.8%	**89.8±0.9%**
	MAS	86.7±1.2%	**90.4±0.8%**
	SCP	88.6±1.2%	**90.4±0.6%**
CIFAR-100	EWC	63.6±4.1%	**67.2±5.0%**
	MAS	63.7±2.9%	**68.3±2.6%**
	SCP	62.2±4.2%	**66.8±3.0%**

5.3. Ablation Analysis

In this section, we further investigate the benefits of sketched SR methods on continual learning algorithms on different aspects.

Overcoming Catastrophic Forgetting. First we consider the effect of overcoming catastrophic forgetting with sketched SR methods. Figure 5 shows the accuracy on each task after training on all the tasks for the compared algorithms. We can see that sketched SR methods forget less about the early tasks, which directly demonstrate its advantage for overcoming catastrophic forgetting. This is consistent to our finding from the synthetic experiments.

Effects of the Sketch Size. We then study the effects of the size of the sketch, i.e. t in (3), on the performance of sketched SR. The results are shown in Figure 6. From the plot we see a clear trade-off between the size of the sketch and the average accuracy, where the average accuracy generally grows as the size of sketches increases — however using more sketches costs more computation resources. Fortunately, even with a very small sketch size, e.g. $t \geq 20$, which is easily affordable in practice, sketched SR methods already significantly outperform diagonal SR methods. This demonstrates the practical effectiveness of the proposed sketched SR framework.

6. Conclusion

In this paper we present sketched structural regularization as a general framework for overcoming catastrophic forgetting in lifelong learning. Compared with the widely-used diagonal version of structural regularization approaches, our methods achieve better performance for overcoming catastrophic forgetting, since an improved approximation to the large importance matrix is adopted. In contrast to the inefficient low-rank approximation methods (e.g., PCA), the proposed sketched structural regularization is computational affordable for practical lifelong learning models. The effectiveness of the proposed methods is verified in multiple benchmark lifelong learning tasks.

References

Thomas D Ahle, Michael Kapralov, Jakob BT Knudsen, Rasmus Pagh, Ameya Velingker, David P Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 141–160. SIAM, 2020.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer, 2002.

Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):1–45, 2017.

Michael B Cohen. Simpler and tighter analysis of sparse oblivious subspace embeddings. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), to appear*, 2016.

Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 163–172, 2015.

Michael B. Cohen, Jelani Nelson, and David P. Woodruff. Optimal Approximate Matrix Product in Terms of Stable Rank. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-013-2. doi: 10.4230/LIPIcs.ICALP.2016.11. URL http://drops.dagstuhl.de/opus/volltexte/2016/6278.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 569–578, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306911. doi: 10.1145/1993636.1993712. URL https://doi.org/10.1145/1993636.1993712.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, page 291–300, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138520. doi: 10.1145/1007352.1007400. URL https://doi.org/10.1145/1007352.1007400.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *International Conference on Learning Representations*, 2018.

Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122*, 2016.

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.

Soheil Kolouri, Nicholas A. Ketz, Andrea Soltoggio, and Praveen K. Pilly. Sliced cramer synaptic consolidation for preserving deeply learned representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJge3TNKwH.

Frederik Kunstner, Lukas Balles, and Philipp Hennig. Limitations of the empirical fisher approximation for natural gradient descent. *arXiv preprint arXiv:1905.12558*, 2019.

Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In *Conference on Learning Theory*, pages 2140–2157. PMLR, 2019.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pages 3925–3934. PMLR, 2019.

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

Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2262–2268. IEEE, 2018.

Xiangrui Meng and Michael W Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 91–100, 2013.

Jelani Nelson and Huy L Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 ieee 54th annual symposium on foundations of computer science*, pages 117–126. IEEE, 2013.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard E Turner, and Mohammad Emtiyaz Khan. Continual deep learning by functional regularisation of memorable past. *arXiv preprint arXiv:2004.14070*, 2020.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LhY8QdUGSuw.

Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1320–1328, 2017.

Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3742–3752, 2018.

Mohammad Rostami, Soheil Kolouri, and Praveen K Pilly. Complementary learning for overcoming catastrophic forgetting using experience replay. *arXiv preprint arXiv:1903.04566*, 2019.

Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41 (4):801–814, 2018.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 143–152. IEEE, 2006.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, 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*, pages 4528–4537. PMLR, 2018.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*, 2017.

Jan van den Brand, Binghui Peng, Zhao Song, and Omri Weinstein. Training (Overparametrized) Neural Networks in Near-Linear Time. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISBN 978-3-95977-177-1. URL https://drops.dagstuhl.de/opus/volltexte/2021/13602.

Chenshen Wu, Luis Herranz, Xialei Liu, Joost van de Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *Advances in Neural Information Processing Systems*, 31:5962–5972, 2018.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

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