Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence (Supplementary Material)

Arslan Chaudhry*, Puneet K. Dokania*, Thalaiyasingam Ajanthan*, Philip H. S. Torr

University of Oxford, United Kingdom
{firstname.lastname}@eng.ox.ac.uk

For the sake of completeness, we first give more details on the KL-divergence approximation using Fisher information matrix (Sec. 2.3). In particular, we give the proof of KL approximation, $D_{KL}(p_\theta \| p_{\theta+\varDelta\theta}) \approx \frac{1}{2}\varDelta\theta^\top F_\theta \varDelta\theta$, discuss the difference between the true Fisher and the empirical Fisher[1], and explain why the Fisher goes to zero at a minimum. Later, in Sec. B.1, we provide a comparison with GEM [4] and show that RWalk significantly outperforms it. Note that, comparison with GEM is not available in the main paper. Additionally, we discuss the sensitivity of different models to the regularization hyperparameter (λ) in Sec. B.2. Finally, we conclude in Sec. B.3 with the details of the architecture and task-based analysis of the network used for CIFAR-100 dataset. *We note that with additional experiments and further analysis in this supplementary the conclusions of the main paper hold.*

A Approximate KL divergence using Fisher Information Matrix

A.1 Proof of Approximate KL divergence

Lemma 1. *Assuming $\varDelta\theta \to 0$, the second-order Taylor approximation of KL-divergence can be written [1,5] as:*

$$D_{KL}(p_\theta \| p_{\theta+\varDelta\theta}) \approx \frac{1}{2}\varDelta\theta^\top F_\theta \varDelta\theta \ , \tag{1}$$

where F_θ is the empirical Fisher at θ.

Proof. The KL divergence is defined as:

$$D_{KL}(p_\theta(\mathbf{z}) \| p_{\theta+\varDelta\theta}(\mathbf{z})) = \mathbb{E}_{\mathbf{z}}\left[\log p_\theta(\mathbf{z}) - \log p_{\theta+\varDelta\theta}(\mathbf{z})\right] \ . \tag{2}$$

Note that we use the shorthands $p_\theta(\mathbf{z}) = p_\theta(\mathbf{y}|\mathbf{x})$ and $\mathbb{E}_{\mathbf{z}}[\cdot] = \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x})}[\cdot]$. We denote partial derivatives as column vectors. Let us first write the second order Taylor series expansion of $\log p_{\theta+\varDelta\theta}(\mathbf{z})$ at θ:

$$\log p_{\theta+\varDelta\theta} \approx \log p_\theta + \varDelta\theta^\top \frac{\partial \log p_\theta}{\partial\theta} + \frac{1}{2}\varDelta\theta^\top \frac{\partial^2 \log p_\theta}{\partial\theta^2}\varDelta\theta \ . \tag{3}$$

[$\star$] Joint first authors

[1] By Fisher, we always mean the empirical Fisher.

Now, by substituting this in Eq. (2), the KL divergence can be approximated as:

$$
\begin{aligned}
D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) &\approx \mathbb{E}_{\mathbf{z}}[\log p_\theta] - \mathbb{E}_{\mathbf{z}}[\log p_\theta] \\
&\quad - \Delta\theta^\top \, \mathbb{E}_{\mathbf{z}}\left[\frac{\partial \log p_\theta}{\partial \theta}\right] - \frac{1}{2}\Delta\theta^\top \, \mathbb{E}_{\mathbf{z}}\left[\frac{\partial^2 \log p_\theta}{\partial \theta^2}\right] \Delta\theta \;, \qquad \text{(4a)} \\
&= \frac{1}{2}\Delta\theta^\top \, \mathbb{E}_{\mathbf{z}}\left[-\frac{\partial^2 \log p_\theta}{\partial \theta^2}\right] \Delta\theta \quad \text{see Eq. (5)} \;, \\
&= \frac{1}{2}\Delta\theta^\top \bar{H} \Delta\theta \quad \text{see Eq. (6b)} \;. \qquad \text{(4b)}
\end{aligned}
$$

In Eq. (4a), since the expectation is taken such that, $\mathbf{x} \sim \mathcal{D}, \mathbf{y} \sim p_\theta(\mathbf{y}|\mathbf{x})$, the first order partial derivatives cancel out, *i.e.*,

$$
\begin{aligned}
\mathbb{E}_{\mathbf{z}}\left[\frac{\partial \log p_\theta(\mathbf{z})}{\partial \theta}\right] &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\sum_{\mathbf{y}} p_\theta(\mathbf{y}|\mathbf{x}) \frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial \theta}\right] \;, \qquad \text{(5)} \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\sum_{\mathbf{y}} p_\theta(\mathbf{y}|\mathbf{x}) \frac{1}{p_\theta(\mathbf{y}|\mathbf{x})} \frac{\partial p_\theta(\mathbf{y}|\mathbf{x})}{\partial \theta}\right] \;, \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\frac{\partial}{\partial \theta} \sum_{\mathbf{y}} p_\theta(\mathbf{y}|\mathbf{x})\right] \;, \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[0] = 0 \;.
\end{aligned}
$$

Note that this holds for the continuous case as well, where assuming sufficient smoothness and the fact that limits of integration are constants (0 to 1), the Leibniz's rule would allow us to interchange the differentiation and integration operators.

Additionally, in Eq. (4b), the expected value of negative of the Hessian can be shown to be equal to the true Fisher matrix ($\tilde{F}$) by using Information Matrix Equality.

$$
\begin{aligned}
\mathbb{E}_{\mathbf{z}}\left[-\frac{\partial^2 \log p_\theta(\mathbf{z})}{\partial \theta^2}\right] &= -\,\mathbb{E}_{\mathbf{z}}\left[\frac{1}{p_\theta(\mathbf{z})} \frac{\partial^2 p_\theta(\mathbf{z})}{\partial \theta^2}\right] \qquad \text{(6a)} \\
&\quad + \mathbb{E}_{\mathbf{z}}\left[\left(\frac{\partial \log p_\theta(\mathbf{z})}{\partial \theta}\right)\left(\frac{\partial \log p_\theta(\mathbf{z})}{\partial \theta}\right)^\top\right] \;, \\
&= -\,\mathbb{E}_{\mathbf{z}}\left[\frac{1}{p_\theta(\mathbf{z})} \frac{\partial^2 p_\theta(\mathbf{z})}{\partial \theta^2}\right] + \tilde{F}_\theta \;. \qquad \text{(6b)}
\end{aligned}
$$

- By the definition of KL-divergence, the expectation in the above equation is taken such that, $\mathbf{x} \sim \mathcal{D}, \mathbf{y} \sim p_\theta(\mathbf{y}|\mathbf{x})$. This cancels out the first term by following a similar argument as in Eq. (5). Hence, in this case, the expected value of negative of the Hessian equals true Fisher matrix ($\tilde{F}$).
- However, if in Eq. (6b), the expectation is taken such that, $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$, the first term does not go to zero, and $\tilde{F}_\theta$ becomes the *empirical Fisher matrix* (F_θ).
- Additionally, at the optimum, since the model distribution approaches the true data distribution, hence even sampling from dataset *i.e.*, $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$ will make the first term to approach zero, and $\bar{H} \approx F_\theta$.

With the approximation that $\bar{H} \approx \tilde{F}_\theta \approx F_\theta$, the proof is complete.

Note that, as we will argue in Section A.2 the true Fisher matrix is expensive to compute as it requires multiple backward passes, hence, instead, as widely used in literature [1,5], we employ empirical Fisher to approximate the KL-divergence.

A.2 Empirical vs True Fisher

Loss gradient Let q be any reference distribution and p (parametrized by θ) be the model distribution obtained after applying softmax on the class scores (s). The cross-entropy loss between q and p can be written as: $\ell(\theta) = -\sum_j q_j \log p_j$. The gradients of the loss with respect to the class scores are:

$$\frac{\partial \ell(\theta)}{\partial s_j} = p_j - q_j \ . \tag{7}$$

By chain rule, the loss gradients w.r.t. the model parameters are $\frac{\partial \ell(\theta)}{\partial \theta} = \frac{\partial \ell(\theta)}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \theta}$.

Empirical Fisher In case of an empirical Fisher, the expectation is taken such that $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$. Since every input $\mathbf{x}$ has only one ground truth label, this makes q a Dirac delta distribution . Then, Eq. (7) becomes:

$$\frac{\partial \ell(\theta)}{\partial s_j} = \begin{cases} p_j - 1, & \text{if `j' is the ground truth label ,} \\ p_j, & \text{otherwise .} \end{cases}$$

Since *at any optimum* the loss-gradient approaches to zero, thus, Fisher being the expected loss-gradient covariance matrix would also approach to a zero matrix.

True Fisher In case of true Fisher, given $\mathbf{x}$, the expectation is taken such that $\mathbf{y}$ is sampled from the model distributions $p_\theta(\mathbf{y}|\mathbf{x})$. In case of multi-class classification, at a given θ, the model distribution learned by a neural network is actually a conditional multinoulli distribution defined as $p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{K} p_{\theta,j}^{[y=j]}$, where $p_{\theta,j}$ is the soft-max probability of the j-th class, K are the total number of classes, $\mathbf{y}$ is the one-hot encoding of length K, and $[\cdot]$ is Iverson bracket. At a good optimum, the model distribution $p_\theta(\mathbf{y}|\mathbf{x})$ becomes peaky around the ground truth label, implying $p_{\theta,t} \gg p_{\theta,j}, \forall j \neq t$ where t is the ground-truth label. Thus, given input $\mathbf{x}$, the model distribution $p_\theta(\mathbf{y}|\mathbf{x})$ approach the ground-truth output distribution. This makes the true and empirical Fisher behave in a very similar manner. Note, in order to compute the expectation over the model distribution, the true Fisher requires multiple backward passes. This makes it prohibitively expensive to compute and the standard solution is to resort to the empirical Fisher instead [1,5].

Table 1: *Following GEM, all the results on ResNets are in the multi-head evaluation setting. Note that, the total number of samples are from all the tasks combined.*

Methods	**Total Number of Samples**	$\boldsymbol{A_k}$ (%)
iCaRL	5120	50.8
GEM	5120	65.4
RWalk (Ours)	5000	**70.1**

B Additional Experiments and Analysis

B.1 Comparison with GEM [4] on ResNets

In this section we show experiments with ResNet18 [2] on CIFAR-100 dataset. In Tab. 1 we report the results where we compare our method with iCaRL [6] and Gradient Episodic Memory (GEM) [4]. Both of these methods use ResNet18 as an underlying architecture. Following GEM-setup, we split the CIFAR-100 dataset in 20 tasks where each task consists of 5 consecutive classes, such that $\cup_{k=1}^{20} \mathbf{y}^k = \{\{0-4\}, \{5-9\}, \dots, \{95-99\}\}$. Note, following GEM, all the algorithms are evaluated in multi-head setting (refer Sec. 2.1 of the main paper). We refer GEM [4] to report the accuracies of iCaRL and GEM. From the Tab. 1, it can be seen that *RWalk outperforms both the methods by a significant margin*.

B.2 Effect of Regularization Hyperparameter (λ)

In Tab. 2 we analyse the sensitivity of different methods to the regularization hyperparameter (λ). As evident, RWalk is less sensitive to λ compared to EWC [2] [3] and PI [7]. This is because of the normalization of the Fisher and Path-based importance scores in RWalk. For example, as we vary λ by a factor of 1×10^5 on MNIST, the *forgetting* and *intransigence* measures changed by -0.06 and 0.14 on EWC [3], and -0.07 and 0.13 on PI [7], respectively. On the other hand, the change in RWalk, as can be seen in the Tab. 2, is 0 for both the measures. On CIFAR-100 a similar trend is observed in Tab. 2.

B.3 CIFAR Architecture and Task-Level Analysis

In Tab. 3 we report the detailed architecture of the convolutional network used in the incremental CIFAR-100 experiments (Sec. 6). Note that, in contrast to PI [7], we use only one fully-connected layer (denoted as 'FC' in the table). For each task k, the weights in the last layer of the network is dynamically added. Additionally, in Fig 1, we present a similar task-level analysis on CIFAR-100 as done for MNIST (Fig. 2 in the main paper). Note that, for all the experiments 'α' in Eq. (6) is set to 0.9 and 'Δt' in Eq. (7) is 10 and 50 for MNIST and CIFAR, respectively.

[2] By EWC we always mean its faster version EWC++.

Table 2: *Comparison of different methods on MNIST and CIFAR-100 as the regularization strength (λ) is varied. With Forgetting and Intransigence we also provide the change (Δ) in the corresponding measures, where the first row in each method is taken as the reference. As discussed in Sec. 4.1 in the main paper, RWalk is less sensitive to λ compared to* EWC *and* PI, *making it more appealing for incremental learning.*

Methods	**MNIST**				**CIFAR**			
	λ	$A_5(\%)$	$F_5(\Delta)$	$I_5(\Delta)$	λ	$A_{10}(\%)$	$F_{10}(\Delta)$	$I_{10}(\Delta)$
EWC	75	80.3	0.19 (0)	0.1 (0)	3.0	28.9	0.38 (0)	-0.17 (0)
	75×10^3	79.2	0.15 (-0.04)	0.21 (0.11)	300	34.1	0.28 (-0.1)	-0.07 (0.1)
	75×10^5	79.1	0.13 (-0.06)	0.24 (0.14)	3×10^5	33.7	0.27 (-0.11)	-0.03 (0.14)
PI	0.1	79.3	0.23 (0)	0.05 (0)	0.1	34.7	0.27 (0)	-0.07 (0)
	100	80.3	0.15 (-0.08)	0.22 (0.17)	10	34.3	0.26 (-0.01)	-0.04 (0.03)
	10000	78.5	0.16 (-0.07)	0.18 (0.13)	1×10^4	33.7	0.27 (0)	-0.06 (0.01)
RWalk (Ours)	0.1	82.6	0.16 (0)	0.12 (0)	0.1	34.5	0.28 (0)	-0.06 (0)
	100	81.6	0.16 (0)	0.14 (0.02)	10	33.2	0.28 (0)	-0.06 (0)
	10000	81.6	0.16 (0)	0.12 (0)	1×10^4	34.2	0.28 (0)	-0.05 (0.01)

Table 3: *CNN architecture for incremental CIFAR-100 used for Vanilla,* EWC, PI, *iCaRl, RWalk in the main paper. Here, 'n' denotes the number of classes in each task.*

Operation	**Kernel**	**Stride**	**Filters**	**Dropout**	**Nonlin.**
3x32x32 input					
Conv	3×3	1×1	32		ReLU
Conv	3×3	1×1	32		ReLU
MaxPool		2×2		0.5	
Conv	3×3	1×1	64		ReLU
Conv	3×3	1×1	64		ReLU
MaxPool		2×2		0.5	
Task 1: FC			n		
$\cdots$: FC			n		
Task k: FC			n		

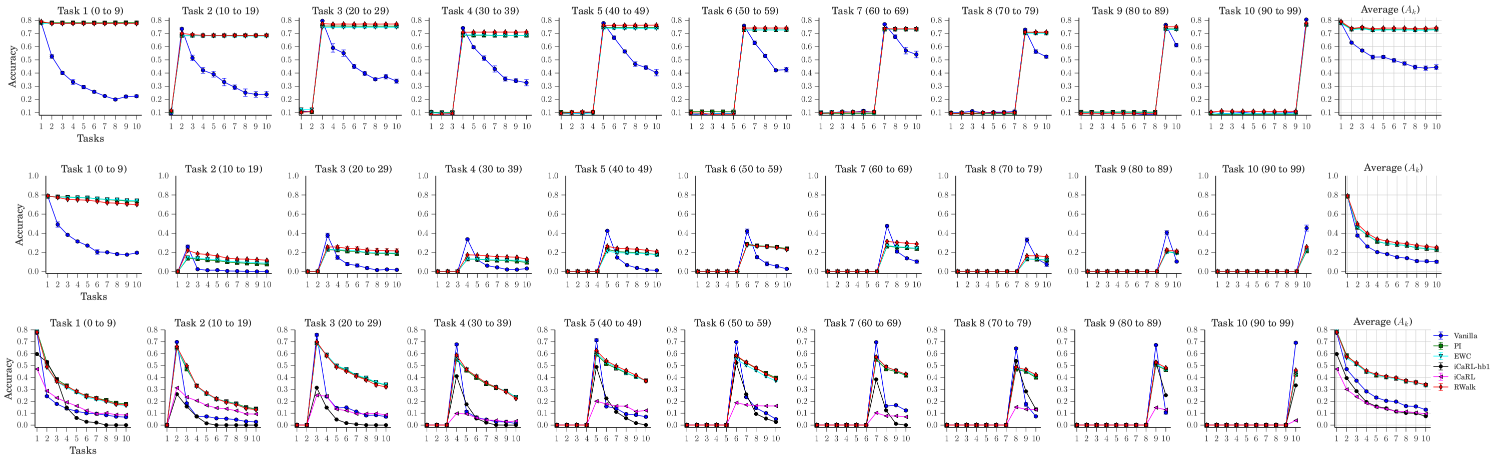

Fig. 1: *Accuracy measure in incremental CIFAR-100 with multi-head evaluation (**top**), and the single-head evaluation without (**middle**) and with samples (**bottom**). The first ten columns show how the performance of different tasks vary as the model is trained for new tasks,* e.g.*, the first plot depicts the variation in performance on Task 1 when the network is sequentially trained for the ten tasks in an incremental manner. The last column shows the average accuracy measure (A_k, by varying k). Mean of features (MoF) sampling is used. (best viewed in color)*

References

1. Amari, S.I.: Natural gradient works efficiently in learning. Neural Computation (1998) 1, 3
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 4
3. Kirkpatrick, J., Pascanu, R., Rabinowitz, N.C., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences of the United States of America (PNAS) (2016) 4
4. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continuum learning. In: NIPS (2017) 1, 4
5. Pascanu, R., Bengio, Y.: Revisiting natural gradient for deep networks. In: ICLR (2014) 1, 3
6. Rebuffi, S.V., Kolesnikov, A., Lampert, C.H.: iCaRL: Incremental classifier and representation learning. In: CVPR (2017) 4
7. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: ICML (2017) 4