On Learning the Geodesic Path for Incremental Learning (Supplementary Material)

Christian Simon[†,§] Piotr Koniusz[§,†] Mehrtash Harandi[♣,§]
[†]The Australian National University, [♣]Monash University, [§]Data61-CSIRO
firstname.lastname@{anu.edu.au,monash.edu,data61.csiro.au}

In this supplementary material, we provide the details of our method and additional results.

1. The details of generating the geodesic flow

Below, we explain the details on how to generate the geodesic flow in § 4 (main submission). Recall the subspaces from the old model $\boldsymbol{P}_{t-1}$, the current model $\boldsymbol{P}_t$, and the orthogonal complement $\boldsymbol{R}$ are used to compute the geodesic flow at ν:

$$\boldsymbol{\Pi}(\nu) = \begin{bmatrix}\boldsymbol{P}_{t-1} & \boldsymbol{R}\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_1\boldsymbol{\Gamma}(\nu) \\ -\boldsymbol{U}_2\boldsymbol{\Sigma}(\nu)\end{bmatrix}. \tag{1}$$

We decompose $\boldsymbol{P}_{t-1}^\top\boldsymbol{P}_t$ and $\boldsymbol{R}^\top\boldsymbol{P}_t$ via the generalized SVD [1] to obtain the orthonormal matrices $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$:

$$\begin{aligned}\boldsymbol{P}_{t-1}^\top\boldsymbol{P}_t &= \boldsymbol{U}_1\boldsymbol{\Gamma}(1)\boldsymbol{V}^\top, \\ \boldsymbol{R}^\top\boldsymbol{P}_t &= -\boldsymbol{U}_2\boldsymbol{\Sigma}(1)\boldsymbol{V}^\top.\end{aligned} \tag{2}$$

All intermediate time steps $\nu \in (0,1)$ on the geodesic path are used for feature projection $\boldsymbol{\Pi}(\nu)^\top\boldsymbol{z}$ for obtaining the similarity in our distillation loss. We note that it is not necessary to compute or store all projection into the intermediate subspace because a closed form solution can be computed as follows:

$$\boldsymbol{Q} = \boldsymbol{\Delta}\begin{bmatrix}\boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_2 & \boldsymbol{\lambda}_3\end{bmatrix}\boldsymbol{\Delta}^\top, \tag{3}$$

where:

$$\boldsymbol{\Delta} = \begin{bmatrix}\boldsymbol{P}_{t-1}\boldsymbol{U}_1 & \boldsymbol{R}\,\boldsymbol{U}_2\end{bmatrix}, \tag{4}$$

$$\begin{aligned}\lambda_{1i} &= 1 + \frac{\sin(2\omega_i)}{2\omega_i}, \\ \lambda_{2i} &= \frac{\cos(2\omega_i) - 1}{2\omega_i}, \\ \lambda_{3i} &= 1 - \frac{\sin(2\omega_i)}{2\omega_i}.\end{aligned} \tag{5}$$

We can calculate $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$, and $\boldsymbol{\lambda}_3$ by using diagonal elements of $\boldsymbol{\Gamma}(1)$ and calculating $\omega_i = \arccos(\gamma_i)$. Note that, the value of γ_i is clamped between -1 and 1 for computational stability.

Algorithm A1 provides details of how we generate the geodesic flow.

Algorithm A1 Generate the Geodesic Flow

Input: The subspaces of the old model $\boldsymbol{P}_{t-1}$ and the current model $\boldsymbol{P}_t$

1: Get the orthogonal complement $\boldsymbol{R}$ of $\boldsymbol{P}_{t-1}$
2: Compute $\boldsymbol{A} = \boldsymbol{P}_{t-1}^\top\boldsymbol{P}_t$ and $\boldsymbol{B} = \boldsymbol{R}^\top\boldsymbol{P}_t$
3: Decompose $\boldsymbol{A}, \boldsymbol{B}$ using gen. SVD to obtain $\boldsymbol{\Sigma}, \boldsymbol{\Gamma}, \boldsymbol{U}_1, \boldsymbol{U}_2$
4: Compute $\boldsymbol{\omega}$ from the diag. elements of $\boldsymbol{\Gamma}(1)$
5: Compute $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$, and $\boldsymbol{\lambda}_3$ using Eq. 5.
6: Compute $\boldsymbol{Q}$ using the closed-form solution in Eq. 3
7: Return the generated geodesic flow $\boldsymbol{Q}$

2. Additional results

We show on ImageNet-subset that our method improves the basic approach for IL without considering additional loss functions or exemplars selection. We follow the setup in § 5.3 in the main paper where only the cross-entropy loss and the *herding* selection mechanism [2] are used for exemplars. The comparison is made between our method (GeoDL) and the prior knowledge distillation approaches proposed in LwF [3] and LUCIR [4] given multiple numbers of tasks and several classifiers. The distillation losses in LwF [3] and LUCIR [4] are referred to as $\mathcal{L}_{\text{LwF}}$ and $\mathcal{L}_{\text{Cos}}$, respectively.

Impact of increasing the number of exemplars in the memory. We investigate the accuracy of IL on ImageNet-subset with 20, 40, 60, 80, 100 exemplars in the memory. The setup is similar to that of where we investigate the impact of different classifiers but we only apply IL with 10 tasks. We also compare our method to the baseline training without any distillation loss $\mathcal{L}_{CE}$.

Table A1 shows that our method outperforms the other knowledge distillation techniques under all memory sizes. Using 20 exemplars in the memory, our method $\mathcal{L}_{\text{GeoDL}}$ outperforms the feature distillation loss $\mathcal{L}_{\text{LwF}}$ and the prediction distillation loss $\mathcal{L}_{\text{Cos}}$ by 1.6% and 7.4%, respectively. Unlike on CIFAR-100, AME obtains the highest accuracy under most memory sizes on ImageNet-subset. We also note that using more exemplars in the memory helps close the performance gap between training the model without and

with a distillation loss.

Method	Classifier	Average accuracy (%) Memory size per class				
		20	40	60	80	100
$\mathcal{L}_{\text{CE}}$	CNN	46.89	56.53	60.84	63.57	66.06
	k-NME	55.22	61.16	64.26	66.03	68.19
	AME	60.65	64.40	66.72	67.81	69.58
$\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{LwF}}$ [3]	CNN	49.39	57.17	61.74	64.66	66.73
	k-NME	59.37	64.96	67.55	68.71	70.40
	AME	64.67	67.85	69.35	70.00	71.32
$\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{Cos}}$ [4]	CNN	65.41	68.38	70.52	71.51	72.77
	k-NME	67.05	68.80	70.77	71.14	72.61
	AME	70.38	70.29	71.78	71.65	73.21
$\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{GeoDL}}$	CNN	59.81	66.09	68.68	70.58	72.04
	k-NME	70.37	72.24	73.32	73.71	**74.28**
	AME	**72.01**	**73.00**	**73.57**	**73.76**	74.25

Table A1: The average accuracy for 10 tasks on ImageNet-subset by varying the number of exemplars in the memory.

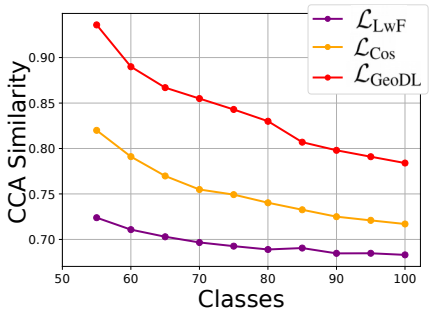

Figure A1: The CCA similarity score between the feature extractor at a specific time $\boldsymbol{\theta}_t$ and the base model $\boldsymbol{\theta}_0$ on CIFAR-100. The score is computed based on the feature outputs in the last layer of $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_0$.

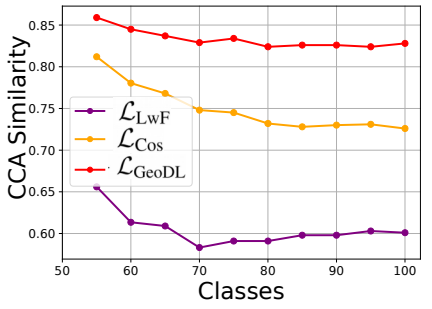

Figure A2: The CCA similarity score between the feature extractor at a specific time $\boldsymbol{\theta}_t$ and the base model $\boldsymbol{\theta}_0$ on ImageNet-subset. The score is computed based on the feature outputs in the last layer of $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_0$.

Analyzing the similarity between models. As we observe that different classifiers may yield different accuracy for IL tasks. In this experiment, we explore the similarity notion between the current model and the old model using Canonical Correlation Analysis (CCA) in [5], as a tool to analyze the representation of deep models. We evaluate the score based on the current model at a specific time $\boldsymbol{\theta}_t$ and the base model $\boldsymbol{\theta}_0$ with the samples coming from the base classes. The high CCA scores show that the model is *less-forgetting*. Fig. A1 shows that the CCA similarity on CIFAR-100 using our approach is the highest compared to training the model with the other distillation losses $\mathcal{L}_{\text{LwF}}$ and $\mathcal{L}_{\text{Cos}}$. We also note that our approach results in the highest CCA similarity between the current feature extractor $\boldsymbol{\theta}_t$ and the base model $\boldsymbol{\theta}_0$, as shown in Fig. A2. The high CCA similarity scores indicate that the current model at time t still highly preserves the representations from the base model (evaluated on the samples of the base classes).

Time and memory consumption. Below, we discuss the time complexity of our approach. The time complexity to obtain a subspace using a standard SVD [1] costs $\mathcal{O}(\text{n}^2\text{d})$. Obtaining the geodesic flow (Eq. 4) costs $\mathcal{O}(\text{nd})$. Our operations are more costly than the less-forget operations [4] which enjoy $\mathcal{O}(\text{d})$ complexity. The time for one iteration using our method is 1.4$\times$ and 1.3$\times$ slower compared to using the distillation losses in LwF [3] and LUCIR [4], respectively. In addition, our approach does not require additional memory to store the exemplars. For the computational memory using NVIDIA GTX Titan X, the whole process of our method requires 2.4GB while training processes with $\mathcal{L}_{\text{Cos}}$, and $\mathcal{L}_{\text{LwF}}$ require 2.1GB and 1.7GB, respectively.

Initialization with less number of classes. The results of our method for 50 classes and 10 classes initialization are 62.8% and 60.5%, respectively, while the results of LUCIR are 60.5% (50 classes) and 57.3% (10 classes).

References

[1] C. F. Van Loan, "Generalizing the singular value decomposition," *SIAM Journal on numerical Analysis*, vol. 13, no. 1, pp. 76–83, 1976. 1, 2

[2] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010. 1

[3] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017. 1, 2

[4] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[5] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems*, 2017, pp. 6076–6085. 2