Supplementary Material for "What Deep CNNs Benefit from Global Covariance Pooling: An Optimization Perspective"

Qilong Wang[1], Li Zhang[1], Banggu Wu[1], Dongwei Ren[1], Peihua Li[2], Wangmeng Zuo[3], Qinghua Hu[1,*]
[1] Tianjin Key Lab of Machine Learning, College of Intelligence and Computing, Tianjin University, China
[2] Dalian University of Technology, China [3] Harbin Institute of Technology, China

A. Implementation Details for Analyzing Smoothing Effect of GCP

In Section 3.2, we analyze smoothing effect of GCP on deep CNNs in terms of the Lipschitzness of optimization loss and the predictiveness of gradients. Specifically, the Lipschitzness of optimization loss is measured by

$$\triangle_l = \mathcal{L}(\mathbf{X} + \eta_l \nabla \mathcal{L}(\mathbf{X})), \eta_l \in [a, b], \tag{A1}$$

and the predictiveness of gradients is measured by

$$\triangle_g = \|\nabla \mathcal{L}(\mathbf{X}) - \nabla \mathcal{L}(\mathbf{X} + \eta_g \nabla \mathcal{L}(\mathbf{X}))\|_2, \eta_g \in [a, b], \tag{A2}$$

where $\mathbf{X}$ is the input; $\nabla \mathcal{L}(\mathbf{X})$ indicates the gradient of loss with respect to the input $\mathbf{X}$; η_l and η_g indicate step sizes of gradient descent.

To assess effect of GCP on the whole CNN models following [8], we employ output of the first convolution layer as $\mathbf{X}$ to compute Eqns. (A1) and (A2). Note that the experiments in Section 4.2 demonstrate that the networks with GCP is more robust to input images with perturbations, comparing with those based on GAP. Accordingly, optimization loss of the networks with GCP also is more stable to input images with perturbations. For clear illustration, we calculate the ranges of $\triangle_l$ and $\triangle_g$ every 1,000 and 500 training steps for MobileNetV2 and ResNet-18, respectively. For calculating the ranges of $\triangle_l$ and $\triangle_g$, we uniformly sample 50 points of η_l (and η_g) from $[0.045,\ 1.5]$ and $[0.1,\ 75]$ for MobileNetV2 and ResNet-18, respectively. Then, we plot the ranges of $\triangle_l$ and $\triangle_g$ determined by the minimum and maximum of the 50 sampled points.

B. Derivations of Eqn. (6) and Eqn. (7)

As described in Section 3.3, the gradient of the loss with respect to the input $\mathbf{X}$ through GCP layer can be calculated as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} = 2\mathbf{J}\mathbf{X}\left[\mathbf{U}\left(\left(\mathbf{K}^T \circ \left(\mathbf{U}^T 2\left(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\right)_{\text{sym}} \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\right)\right) + \left(\frac{1}{2}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T \frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U}\right)_{\text{diag}}\right)\mathbf{U}^T\right]_{\text{sym}}. \tag{A3}$$

Here, we give detailed derivations of Eqn. (A3) as follows. To perform GCP, we compute the square root of sample covariance matrix of features $\mathbf{X} \in \mathbb{R}^{N \times D}$ as

$$\mathbf{Z}_{\text{GCP}} = \mathbf{\Sigma}^{\frac{1}{2}} = (\mathbf{X}^T \mathbf{J} \mathbf{X})^{\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T, \tag{A4}$$

where $\mathbf{U}$ and $\mathbf{\Lambda}$ are the matrix of eigenvectors and the diagonal matrix of eigenvalues of sample covariance $\mathbf{\Sigma}$, respectively. As shown in [4], $\frac{\partial \mathcal{L}}{\partial \mathbf{X}}$ can be calculated as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} = \mathbf{J}\mathbf{X}\left(\frac{\partial \mathcal{L}}{\partial \mathbf{\Sigma}} + \left(\frac{\partial \mathcal{L}}{\partial \mathbf{\Sigma}}\right)^T\right), \tag{A5}$$

*Qinghua Hu is the corresponding author.
Email: {qlwang, li_zhang, huqinghua}@tju.edu.cn. The work was supported by the National Natural Science Foundation of China (No. 61806140, 61971086, 61925602, U19A2073, 61732011). Qilong Wang was supported by National Postdoctoral Program for Innovative Talents.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{\Sigma}} = \mathbf{U}\Big(\Big(\mathbf{K}^T \circ \Big(\mathbf{U}^T \frac{\partial \mathcal{L}}{\partial \mathbf{U}}\Big)\Big) + \Big(\frac{\partial \mathcal{L}}{\partial \mathbf{\Lambda}}\Big)_{\text{diag}}\Big)\mathbf{U}^T, \tag{A6}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = \Big(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}} + \Big(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\Big)^T\Big)\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}, \tag{A7}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{\Lambda}} = \frac{1}{2}\Big(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T \frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U}\Big)_{\text{diag}}. \tag{A8}$$

Let $(\mathbf{A})_{\text{sym}} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$, we can rewrite Eqn. (A5) and Eqn. (A7) as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} = 2\mathbf{J}\mathbf{X}\Big(\frac{\partial \mathcal{L}}{\partial \mathbf{\Sigma}}\Big)_{\text{sym}}, \tag{A9}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = 2\Big(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\Big)_{\text{sym}}\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}. \tag{A10}$$

By substituting Eqns. (A6), (A8) and (A10) into Eqn. (A9), we achieve

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{X}} &= 2\mathbf{J}\mathbf{X}\Big(\frac{\partial \mathcal{L}}{\partial \mathbf{\Sigma}}\Big)_{\text{sym}} \\
&= 2\mathbf{J}\mathbf{X}\Big(\mathbf{U}\Big(\Big(\mathbf{K}^T \circ \Big(\mathbf{U}^T \frac{\partial \mathcal{L}}{\partial \mathbf{U}}\Big)\Big) + \Big(\frac{\partial \mathcal{L}}{\partial \mathbf{\Lambda}}\Big)_{\text{diag}}\Big)\mathbf{U}^T\Big)_{\text{sym}} \\
&= 2\mathbf{J}\mathbf{X}\Big[\mathbf{U}\Big(\Big(\mathbf{K}^T \circ \Big(\mathbf{U}^T 2\Big(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\Big)_{\text{sym}}\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\Big)\Big) + \Big(\frac{1}{2}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T \frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U}\Big)_{\text{diag}}\Big)\mathbf{U}^T\Big]_{\text{sym}}.
\end{aligned} \tag{A11}$$

So far, we obtain Eqn. (A3).

With some assumptions and simplification, Eqn. (A3) can be trimmed as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} \approx 2\mathbf{J}\mathbf{X}\Big(2\mathbf{K}^T \circ \mathbf{\Lambda}^{\frac{1}{2}} + \frac{1}{2}\mathbf{\Lambda}^{-\frac{1}{2}}\Big)\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}, \tag{A12}$$

where $\circ$ denotes matrix Hadamard product. In the following, we explain how we obtain Eqn. (A12). Specifically, we simplify Eqn. (A3) by neglecting $(\centerdot)_{\text{sym}}$ and $(\centerdot)_{\text{diag}}$ operations. Thus, Eqn. (A3) can be approximated by

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} \approx 2\mathbf{J}\mathbf{X}\Big[\mathbf{U}\Big(\mathbf{K}^T \circ \Big(\mathbf{U}^T 2\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\Big) + \frac{1}{2}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T \frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U}\Big)\mathbf{U}^T\Big]. \tag{A13}$$

Then, we assume that matrix multiplications between diagonal matrix $\mathbf{\Lambda}$ and orthogonal matrix $\mathbf{U}$ (or symmetric matrix $\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}$) in Eqn. (A13) satisfy the commutative law of multiplication. So Eqn. (A13) can be trimmed as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} \approx 2\mathbf{J}\mathbf{X}\Big[\mathbf{U}\Big(\mathbf{K}^T \circ \Big(\mathbf{U}^T 2\mathbf{\Lambda}^{\frac{1}{2}}\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U}\Big) + \frac{1}{2}\mathbf{U}^T\mathbf{\Lambda}^{-\frac{1}{2}}\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U}\Big)\mathbf{U}^T\Big]. \tag{A14}$$

Finally, we assume that the mask matrix $\mathbf{K}$ only has effect on the diagonal matrix of eigenvalues $\mathbf{\Lambda}$. So we have

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{X}} &\approx 2\mathbf{J}\mathbf{X}\Big[\mathbf{U}\Big(\mathbf{U}^T 2\mathbf{K}^T \circ \mathbf{\Lambda}^{\frac{1}{2}}\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U} + \frac{1}{2}\mathbf{U}^T\mathbf{\Lambda}^{-\frac{1}{2}}\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\mathbf{U}\Big)\mathbf{U}^T\Big] \\
&= 2\mathbf{J}\mathbf{X}\Big[\mathbf{U}\Big(\mathbf{U}^T\Big(2\mathbf{K}^T \circ \mathbf{\Lambda}^{\frac{1}{2}}\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}} + \frac{1}{2}\mathbf{\Lambda}^{-\frac{1}{2}}\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}\Big)\mathbf{U}\Big)\mathbf{U}^T\Big] \\
&= 2\mathbf{J}\mathbf{X}\Big(2\mathbf{K}^T \circ \mathbf{\Lambda}^{\frac{1}{2}} + \frac{1}{2}\mathbf{\Lambda}^{-\frac{1}{2}}\Big)\frac{\partial \mathcal{L}}{\partial \mathbf{Z}_{\text{GCP}}}.
\end{aligned} \tag{A15}$$

Note that, in practice, Eqn. (A15) is not employed for back-propagation of GCP, but provides a simplified form of Eqn. (A3) for discussion on connection with second-order optimization in context of deep CNNs.

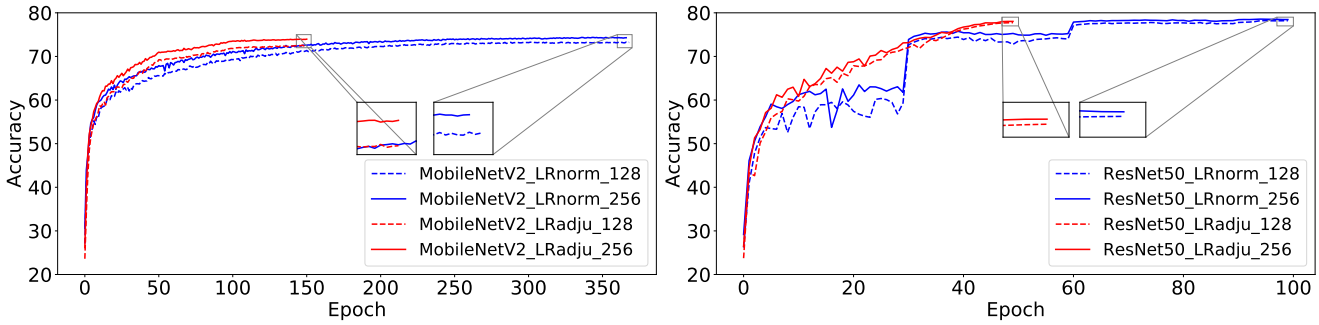

Figure A. Convergence curves of MobileNetV2 and ResNet-50 trained with GCP under different settings of lr and various dimension (D) of input features on ImageNet.

C. Convergence Curves of Networks with GCP under Various Dimensions of Input

In Table 5 of Section 4.1, we gave the results of MobileNetV2 and ResNet-50 with GCP under different settings of lr (i.e., LR_{norm} and LR_{adju}) and various dimension (i.e., $D = 256$ and $D = 128$) of input features on ImageNet. Figure A illustrates their corresponding convergence curves, from which we can see that lower-dimensional covariance representations (COV-Reps) share similar behavior with higher-dimensional COV-Reps, but the lower-dimensional COV-Reps suffer from larger performance degradation in the case of faster convergence (i.e., LR_{adju}).

D. Implementation Details on Applying Pre-trained Networks with GCP to Other Vision Tasks

To apply the pre-trained networks with GCP to object detection and instance segmentation on MS COCO, we adopt the same strategy with the original GAP-based CNN models [3, 7] and make a modification, i.e., increasing resolution of feature maps in the last stage. The detailed steps are described as follows. All detectors are implemented using MMDetection toolkit [1].

S_I: Pre-training the networks with GCP on ImageNet [2] without down-sampling in *conv5_1* as suggested in [4];

S_II: Discarding the GCP layer and the classifier, while introducing Region Proposal Networks (RPN) [7] and Region of Interest (ROI) Pooling [3, 7];

S_III: Increasing resolution of feature maps in the last stage using GCP$_D$ (i.e., use of down-sampling as done in the original ResNet) and GCP$_M$ (i.e., a max-pooling layer with a step size 2 is inserted before *conv5_1*) strategies, while introducing feature pyramid networks (FPN) [5];

S_IV: Fine-tuning the whole networks in S_III on MS COCO [6] using the same hyper-parameters with those of the original GAP-based CNN models.

E. Computational Comparison of GCP and GAP

Here, we compare GCP and GAP in terms of computational cost. The experiments are conducted on large-scale ImageNet using ResNet-18, ResNet-34, ResNet-50 and ResNet-101 as backbone models. The evaluation metrics include network parameters, floating point operations per second (FLOPs), training or inference time per image, and Top-1/Top-5 accuracies. For GCP, size of covariance representations is set to 8k. All models are trained with the same experimental settings and run on a workstation equipped with four Titan Xp GPUs, two Intel(R) Xeon Silver 4112 CPUs @ 2.60GHz, 64G RAM and 480 GB INTEL SSD. From the results in Table 1, we can see that GCP introduces extra $\sim$7M parameters, $\sim$0.4ms training time and $\sim$0.2ms inference time, but increase about 4.6%, 2.6%, 1.5% and 1.8% Top-1 accuracies over GAP-based ResNet-18, ResNet-34, ResNet-50 and ResNet-101, respectively. Besides, GCP achieves matching performance using much lower computational complexity than GAP (e.g., ResNet34+GCP *vs.* ResNet101+GAP and ResNet50+GCP *vs.* ResNet152+GAP). Additionally, GCP with similar computational complexity achieves much better performance than GAP (e.g., ResNet18+GCP *vs.* ResNet34+GAP and ResNet50+GCP *vs.* ResNet101+GAP). Note that we discard down-sampling operation in $conv5\_x$ for GCP with ResNets, which significantly increases FLOPs. When we use this down-sampling operation, GCP shares similar FLOPs with GAP, leading slight performance decrease.

Table 1. Comparison of GCP and GAP using various ResNets in terms of network parameters, floating point operations per second (FLOPs), training or inference time per image, and classification accuracy.

Methods	Parameter	GFLOPs.	Training time (ms)	Inference time (ms)	Top-1 Err. (%)	Top-5 Err. (%)
ResNet18 + GAP	11.69M	1.81	0.77	0.60	70.47	89.59
ResNet18 + GCP	19.60M	3.11	1.21	0.85	75.07	92.14
ResNet34 + GAP	21.80M	3.66	1.17	0.88	74.19	91.60
ResNet34 + GCP	29.71M	5.56	1.61	1.10	76.80	93.11
ResNet50 + GAP	25.56M	3.86	1.85	1.29	76.02	92.97
ResNet50 + GCP	32.32M	6.19	2.22	1.49	78.56	93.72
ResNet101 + GAP	44.55M	7.57	2.79	1.72	77.67	93.83
ResNet101 + GCP	51.31M	9.90	3.14	1.83	79.47	94.30
ResNet152 + GAP	60.19M	11.28	3.54	2.55	78.13	94.04

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017.

[4] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? In *ICCV*, 2017.

[5] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[7] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[8] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *NeurIPS*, 2018.