Supplementary Material: Equivariance-bridged SO(2)-Invariant Representation Learning using Graph Convolutional Network

Sungwon Hwang
shwang.14@kaist.ac.kr
Hyungtae Lim
shapelim@kaist.ac.kr
Hyun Myung
hmyung@kaist.ac.kr

School of Electrical Engineering,
Korea Advanced Institute of
Science and Technology (KAIST),
Daejeon, Korea

A Maintaining Spatial Correspondence over Batch Normalization

If faced with disturbance of layer-wise means and variances distribution introduced by interpolation when images for inference are rotated, batch normalization [1] may not be able to strictly preserve the rotational invariance of spatial correspondence. Note that batch normalization uses running means and variances of intermediate representations retrieved from upright images during training in order to normalize the representations when making inferences. Specifically, given each dimension of a set of vertices representation $\bar{\mathcal{H}}^{(l)} = ({}^0\bar{\mathcal{H}}^{(l)} \dots {}^{c'^{(l)}}\bar{\mathcal{H}}^{(l)}) = W_1^{(l)}\hat{\mathcal{H}}^{(l)}$ of a rotated image R^{θ}, normalization for $BN(\bar{\mathcal{H}}^{(l)})$ during inference is conducted as

$$\frac{{}^k\bar{\mathcal{H}}^{(l)} - \mathbb{E}[{}^k\bar{\mathcal{V}}^{(l)}]}{\sqrt{Var[{}^k\bar{\mathcal{V}}^{(l)}] + \varepsilon}}$$

where $\mathbb{E}[\cdot]$ and $Var[\cdot]$ denote expectation and variance respectively, and ε is a small constant to prevent zero division. Accordingly, $\mathbb{E}[{}^k\bar{\mathcal{H}}^{(l)}]$ and $Var[{}^k\bar{\mathcal{H}}^{(l)}]$ must be equal to $\mathbb{E}[{}^k\bar{\mathcal{V}}^{(l)}]$ and $Var[{}^k\bar{\mathcal{V}}^{(l)}]$ in order to strictly follow the philosophy of batch normalization and to preserve the RISC.

Unfortunately, contrary to models trained with data augmentation, which are provided with ${}^k\hat{\mathcal{H}}^{(l)}$ during training, models trained in our constrained problem definition are not provided with such information.

However, we have $\hat{\mathcal{H}}^{(l)}_{(u,v)} \approx \hat{\mathcal{V}}^{(l)}_{(w,h)}$ from the analysis of SMP, from which we can assume that $\mathbb{E}[{}^k\bar{\mathcal{H}}^{(l)}] \approx \mathbb{E}[{}^k\bar{\mathcal{V}}^{(l)}]$ and $Var[{}^k\bar{\mathcal{H}}^{(l)}] \approx Var[{}^k\bar{\mathcal{V}}^{(l)}]$.

Still, the lingering errors occurring from the approximations can be ameliorated by using means and variances extracted from whole test data, or practically mini-batch statistics during inference. Improvement of classification performance of our model when making inferences with test mini-batch statistics is reported in Table 1.

Table 1: Classification accuracies of SWN-GCN (w/ C.F) when using running statistics(means and variances) versus using test mini-batch statistics for batch normalizations during inference.

	MNIST	CIFAR-10
RUNNING STAT.	90.37 %	50.31%
TEST MINI-BATCH STAT.	91.78 %	50.51%

In fact, improvement in classification accuracy is more noticeable in *R*-MNIST. We may accuse the nature of grayscale, hand-written digit images for such observation. Interpolated values on borderlines of pixels in an MNIST image are likely to be retrieved from two far-ended values, white and black. This is quite an extreme case for interpolation compared to softer difference of pixels on borderlines of the objects in CIFAR-10 images.

Note that the aforementioned problem caused by unseen distribution of data introduced by interpolation applies to all baseline models with batch normalizations. Thus, all the reported performances on the baselines used test mini-batch for batch normalizations when making inference.

B Implementation Details

We can easily implement SMP in spatial domain by letting every vertex receive messages using depth-wise convolution of 3×3 kernels whose adjacent weights in the kernels fixed as $\frac{1}{9}$, leaving the parameter at the center of the kernel to be trainable, which corresponds to $\beta^{(l)}$. SWP is implemented with pointwise convolution, or convolution with kernel size of 1×1. Specifically, we need $c'^{(l)}$ number of $1 \times 1 \times c^{(l)}$ sized convolution kernel to implement first shared-weight propagation by $W_1^{(l)}$, and need $c^{(l+1)}$ number of $1 \times 1 \times c'^{(l)}$ sized convolution kernel to implement second shared-weight propagation by $W_2^{(l)}$. The propagating dimensions of vertices for every layer is summarized in Table 2.

Note that the order of SMP and SWP is switched deliberately to construct layers of WN-GCN model for CIFAR-10, as the first SMP layer over the input images yields an unwanted blurring effect.

C Classification Preservation Rate over Rotations

Apart from direct quantification of rotational invariance via relative L_2 norm of rotational variance ($\delta_{L_2}^{\theta}$) and absolute cosine similarity of rotational invariance ($\delta_{\cos}^{\theta}$), it is also important to observe how consistently the linear classifier can make inferences on an image over rotations. For that matter, we suggest Class Preservation Rate over Rotations(*CPRR*), which measures how much of the correctly classified images that are rotated by θ degrees are still

Table 2: SWN-GCN model configuration

l	$c^{(l)}$	$c'^{(l)}$	$c^{(l+1)}$
0	3	64	64
1	64	64	64
2	64	128	128
3,4	128	128	128
5	128	256	256
6,7,8	256	256	256
9	256	512	512
10,11,12	512	512	512

correctly classified when they are rotated by θ' degrees. That is,

$$CPRR_{\theta}^{\theta'} := \frac{\sum\{correct[(\text{SWN-GCN}(R^{\theta'})] \cap correct[(\text{SWN-GCN}(R^{\theta})]\}}{\sum\{correct[(\text{SWN-GCN}(R^{\theta})]\}}, \tag{1}$$

where *correct()* returns a list of images correctly classified by a trained linear classifier, and $\sum$ counts the total number of images of the list, so that we can measure the proportion of correctly classified images that are also correctly classified when rotated in different angles.

We have measured *CPRR*s over SWN-GCN along the other two baselines, [2] and [3], on CIFAR-10 dataset, and the result is presented in Table 3. Again, the networks as well as classifiers are trained with upright images only.

Table 3: Comparison of *CPRR*s on CIFAR-10

	$CPRR_0^{30} \uparrow$	$CPRR_0^{60} \uparrow$	$CPRR_{30}^{60} \uparrow$
TIGRANET [2]	0.916	0.881	0.901
E(2)-CNN *C-8* [3]	0.607	0.492	0.652
SWN-GCN (OURS)	0.921	0.876	0.891

CPRR values of TIGraNet and SWN-GCN are substantially higher than E(2)-CNN for all angle differences. However, despite the similar *CPRR*s between SWN-GCN and TIGraNet, SWN-GCN outperforms TIGraNet on classification accuracies for all angles by large margins as are reported in the main paper.

Please note that despite *CPRR* being a good measure of *"preservative classifiability"* of invariant representations, *CPRR* cannot be a strict measure of rotational invariance because the classification boundary can still allow rooms for representations that are rotation-varying within the bound.

D Ablating the Components of Deep Network

Ability of SWN-GCN to construct deeper representation over spectral graph convolution-based invariant representation learning [2] is one of the main contributions of the work. We ablated over components of our network that are typically employed to construct deeper networks: a) network size and b) batch normalization, in order to observe the behavior over each of their presence.

We controlled the depth and size of the ablating networks by the number of training parameters in SMP and SWP, while configuring the classifiers to each of the network's output channel accordingly, since the classifier is not the focus of our ablation. The result is displayed in Table 4.

Table 4: Ablations over network size and batch normalization. Values are classifcation accuracies over rotated CIFAR-10. All models are trained with upright images. S refers to the number of parameters corresponding to the configuration in Table 2, B.N is Batch Normalization w/ (with) and w.o/ (without).

NETOWRK SIZE	$0.3S$	S	$1.5S$
W/ B.N	40.2%	50.4%	**50.7**%
W.O/ B.N	36.9%	45.5%	45.6%

With batch normalizations, deeper networks could yield better performance yet the increment of network size yields less increment in performance as the network size increases. However, deeper network yields larger performance discrepancies between those with and without batch normalizations. We conjecture that the effect of internal covariance shift problem is more pervasive for deeper network (a widely known problem targeted by batch-normalizations).

References

[1] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[2] Renata Khasanova and Pascal Frossard. Graph-based isometry invariant representation learning. In *Proceedings of the International Conference on Machine Learning*, pages 1847–1856, 2017.

[3] Maurice Weiler and Gabriele Cesa. General $e(2)$-equivariant steerable cnns. In *Advances in Neural Information Processing Systems*, pages 14334–14345, 2019.