Learning Canonical View Representation for 3D Shape Recognition with Arbitrary Views

Supplementary Material

Xin Wei[1*], Yifei Gong[2*], Fudong Wang[2], Xing Sun[2]✉, Jian Sun[1]✉
[1]Xi'an Jiaotong University, [2] Tencent Youtu Lab
wxmath@stu.xjtu.edu.cn, {yifeigong,winfredsun}@tencent.com
fudong-wang@whu.edu.cn, jiansun@xjtu.edu.cn

In this supplementary material, we provide additional ablation study and the visualization for our approach.

A. Additional Ablation Study

To further evaluate the performance impact of different components in our network, we report additional results on the selections of hyperparameters and architectures. We conduct all these experiments on ModelNet40 [6] under the arbitrary-view setting.

A.1. Backbone network

We first examine the performance of our method with different CNN backbone networks: AlexNet [2], ResNet-18 [1], ResNet-50 [1] and ResNet-101 [1]. As shown in Tab. 1, a more efficient backbone network produces better performance, as variants of the ResNet architecture outperform AlexNet significantly. However, the performance margin among the ResNet backbones are much less noticeable, with the deeper ResNet-101 achieves less than 1% gain in accuracy over ResNet-18. We choose the ResNet-18 network for our implementation since it performs reasonably well while being less computationally expensive.

A.2. Obtaining the global representation

As mentioned in Sect. 4.3, Global Average Pooling (GAP) is performed on the outputs of the Transformer Encoder in canonical view aggregator to obtain a global representation of the 3D shape. Here we compare the GAP to other methods in producing the global representation, including Global Max Pooling (GMP) and concatenating the features directly. As shown in Tab. 2, we can see that GAP performs noticeably better than GMP, while marginally outperforming the direct concatenation of features. One possible explanation for GMP's lower performance is that the gradients are only back-propagated to the maximum elements. For our particular network design, this could potentially be harmful for learning diverse and robust canonical view features.

Table 1. Results with different CNN backbone networks.

Backbone	Per Class Acc.	Per Ins. Acc.
AlexNet [2]	75.94%	78.88%
ResNet-18 [1]	84.01%	86.91%
ResNet-50 [1]	83.64%	87.18%
ResNet-101 [1]	**84.34%**	**87.77%**

Table 2. Comparing methods for obtaining global representation.

	Per Class Acc.	Per Ins. Acc.
Concat.	83.76%	86.42%
GMP	82.77%	85.41%
GAP	**84.01%**	**86.91%**

Table 3. Impact of the weighting factor λ for the Canonical View Feature Separation Loss.

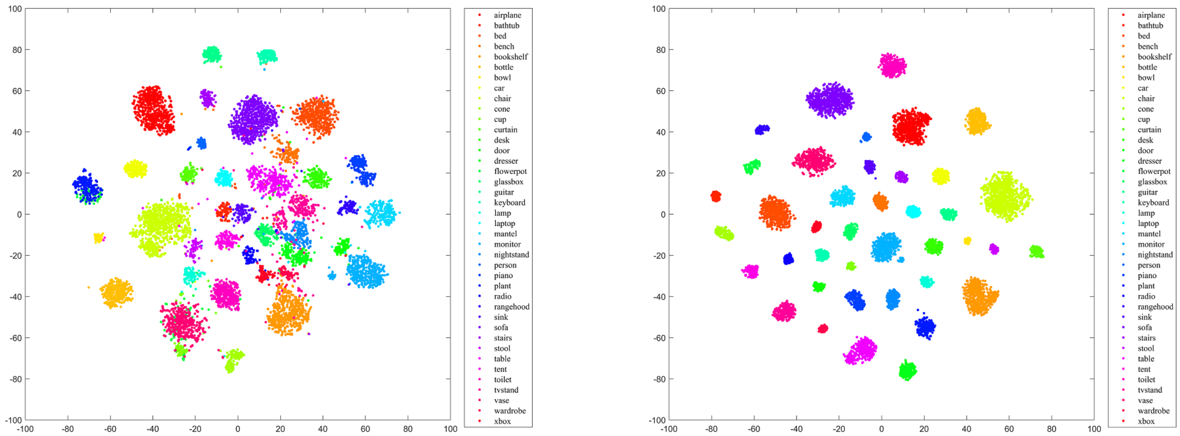
	Per Class Acc.	Per Ins. Acc.
$\lambda = 1.0$	82.97%	85.12%
$\lambda = 0.5$	81.06%	84.02%
$\lambda = 0.1$	**84.01%**	**86.91%**

Table 4. Comparison of learnable spatial embeddings with fixed sinusoidal positional embeddings.

	Per Class Acc.	Per Ins. Acc.
Fixed	82.14%	85.41%
Learned	**84.01%**	**86.91%**

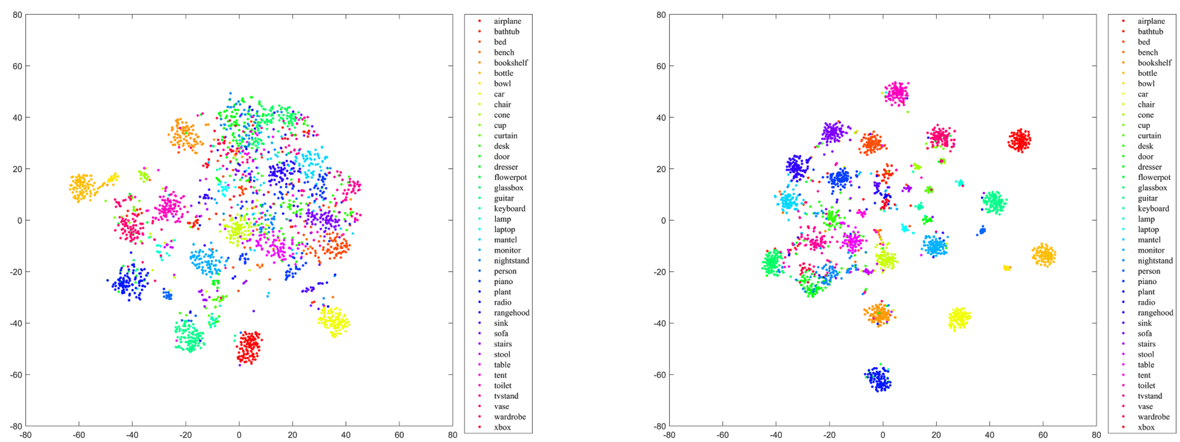
A.3. Loss coefficient λ

As defined in Eq. (11), the overall loss of our network consists of the classification loss L_{cls} and the Canonical View Feature Separation Loss (CVFSL) L_{sep}, where the coefficient λ controls the weighting factor between the two loss functions. We conduct experiments to examine how λ can affect the performance. As seen in Tab. 3, increasing λ



(a) MVCNN-M (b) Ours

Figure 1. Visualization of shape features learned by MVCNN-M (a) and our method (b) via t-SNE on ModelNet40 train set.



(a) MVCNN-M (b) Ours

Figure 2. Visualization of shape features learned by MVCNN-M (a) and our method (b) via t-SNE on ModelNet40 test set.

from 0.1 to 0.5 and 1.0 lowers the classification accuracy. This shows that a good balance between the classification loss and the CVFSL is important for maximizing the performance. We set $\lambda = 0.1$ for our implementation in all experiments.

A.4. Positional embedding

Positional embedding is crucial in Transformer-based architectures to capture sequential information of the inputs. Vaswani et al. [5] originally adopts fixed sinusoidal positional embeddings to represent positions, where the t-th input's sinusoidal positional embedding is defined as

$$PE_{(t,2i)} = \sin\left(t/100000^{2i/d}\right) \tag{1}$$

where d is the feature dimension and $i = 1, 2, ..., d$.

As mentioned in Sect. 4.3, our approach uses learnable spatial embeddings $F^{se} = \Psi(F^s)$ to encode positional information, where F^s is the spatial representation inferred from the canonical view features F^c by a two-layer MLP Ψ and is constrained by Canonical View Feature Separation Loss (CVFSL). To compare the performance impacts of fixed and learned embeddings, we substitute the learned spatial embedding F^{se} with fixed sinusoidal positional embeddings. As shown in Tab. 4, the classification results drop by 1.87% and 1.50% in two accuracies, which demonstrates the effectiveness of learnable spatial embeddings.

B. Visualization

In Fig. 1 and Fig. 2, we visualize the features learned by MVCNN-M [3] and our method on both the train set and the test set of ModelNet40 under the arbitrary-view setting. We perform t-SNE [4] on features of object instances from all classes to visualize the feature discriminability on a macro level. According to Fig. 1 and Fig. 2, in both the train set and the test set, features from our method display much better clustered distributions under t-SNE than those produced by MVCNN-M [3]. Specifically, we can observe both lower intra-class variance and higher inter-class variance in the results of our method compared to MVCNN-M [3], which reflects better overall shape classification performance on ModelNet40 with arbitrary view.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25:1097–1105, 2012. 1

[3] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, pages 945–953, 2015. 3

[4] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 3

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017. 2

[6] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015. 1