Supplementary Material: Attentive Weights Generation for Few Shot Learning via Information Maximization

Yiluan Guo, Ngai-Man Cheung
Singapore University of Technology and Design
yiluan_guo@mymail.sutd.edu.sg, ngaiman_cheung@sutd.edu.sg

1. Effects of Two Self-Attention Networks

We are using two self-attention(SA) networks in contextual and attentive paths. Letting these two SA networks share weights seems a natural choice to reduce computational overhead and avoid over-fitting. However, in Table 1 we can observe that shared SA networks harm the performance. To further investigate the reasons behind, we randomly sample two 5-way 1-shot tasks and visualize the attention weight maps for all the four heads of three attention networks in Figure 1. Comparing first two rows (SA_1 and SA_2 in the two figures), we can see that the two SA networks produce significantly different attention weight maps for the same support samples. This demonstrates that the SA network in contextual path emphasizes on generating the classification weights and its couterpart in attentive path plays the role of providing the context to be attended by different query samples in the following cross attention. Sharing the same self-attention networks might limit the expressiveness of learned representations in both paths.

Another important observation from Figure 1 is that multiple heads in attention networks are indeed learning distinct subspaces when necessary. In particular, query sample in each row of CA in Figure 1 has completely different attention weights for 4 heads. This shows the expressiveness of multi-head attention, validating Table 4 in the paper.

Table 1. Accuracy results on *mini*ImageNet with shared/separate self-attention (SA) networks.

Method	5-way 1-shot	5-way 5-shot
seperate SA	63.12%	78.40%
shared SA	62.46%	76.81%

2. Visualization

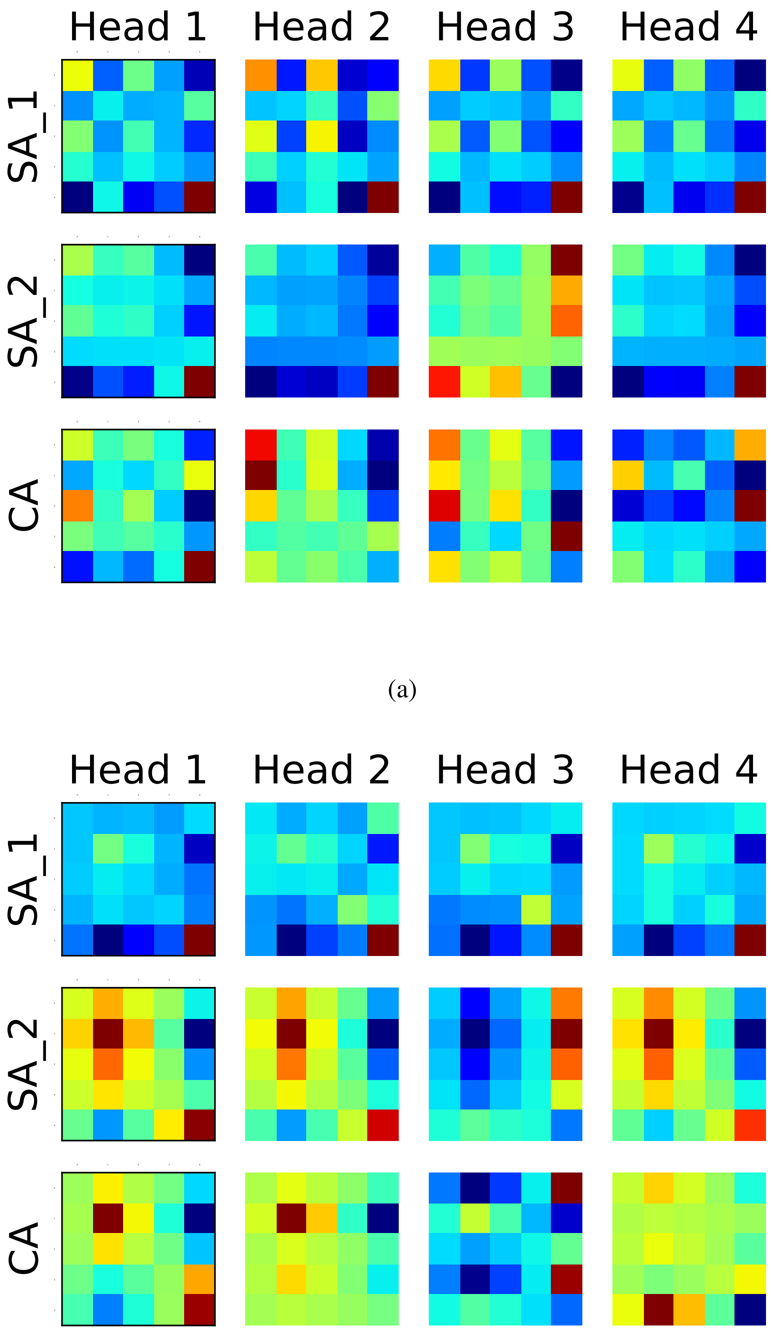
We visualize the generated classification weights by t-SNE [1]. First we sample 400 tasks from meta-validation set of 5-way 1-shot *mini*ImageNet experiment. Each task contains 5 query samples from 5 different classes. Thus in total there are $400 \times 5 \times 5 = 10,000$ weight vectors to visualize. As comparison, inputs to the generator g are also plotted. The visualization results are shown in Figure 2. Two figures in the first row show the inputs to g while those in second row show the generated weights. Both two figures in each row are the same except that in the second column the color is turned off to highlight the weights for different query samples. Different colors indicate different classes.

From the comparison between (a) and (c), we can see the decoded weights for each class in (c) are clustered closer than (a) in general. Red and blue dots in (b, d) denote the classification weights for two query samples from two different classes within one task. It can be observed that g can generate adapted weights for different query samples. This is consistent with Table 3 in the paper, where the results of "random shuffle between classes" suggest that query samples from different class have distinct classification weights.

3. Few Shot Regression

AWGIM can be applied to few shot regression task by slight adaptations. During meta-training, we set the number of classes N equal to 1 and change the cross entropy loss to mean square error. We use the data points (x, y) as inputs to AWGIM and generate weight as well as bias parameters for a three layer MLP with hidden dimension 40. This is consistent with few shot regression experimental setting in LEO [2].

The few shot regression tasks are constructed as either sinusoidal or linear regression tasks. For sinusoidal regression tasks, the amplitude range is $[0.1, 5]$, phase range $[0, 2\pi]$, frequency range $[0.5, 2.0]$. For linear regression tasks, the slope range is $[-1, 1]$, intercept range $[-5, 5]$. Input x is randomly sample from $[-5, 5]$. Gaussian noise with standard deviation 0.3 is added to y during meta-training. We show some qualitative results in Figure 3. (a), (b) and



(a)

(b)

Figure 1. Attention weight maps for all the 4 heads of three attention networks. Each row in SA_1 and SA_2 stands for one support sample while 5 rows mean 5 query samples in CA. Color indicates the intensity of the attention weights. (a) and (b) are two randomly sampled 5-way 1-shot tasks from *mini*ImageNet.

(c) are examples that can be tackled easily. For some non-trivial cases such as (d), where both sinusoidal and linear regression can fit the support data, AWGIM produces predictions slightly mixing with another regression family, de-

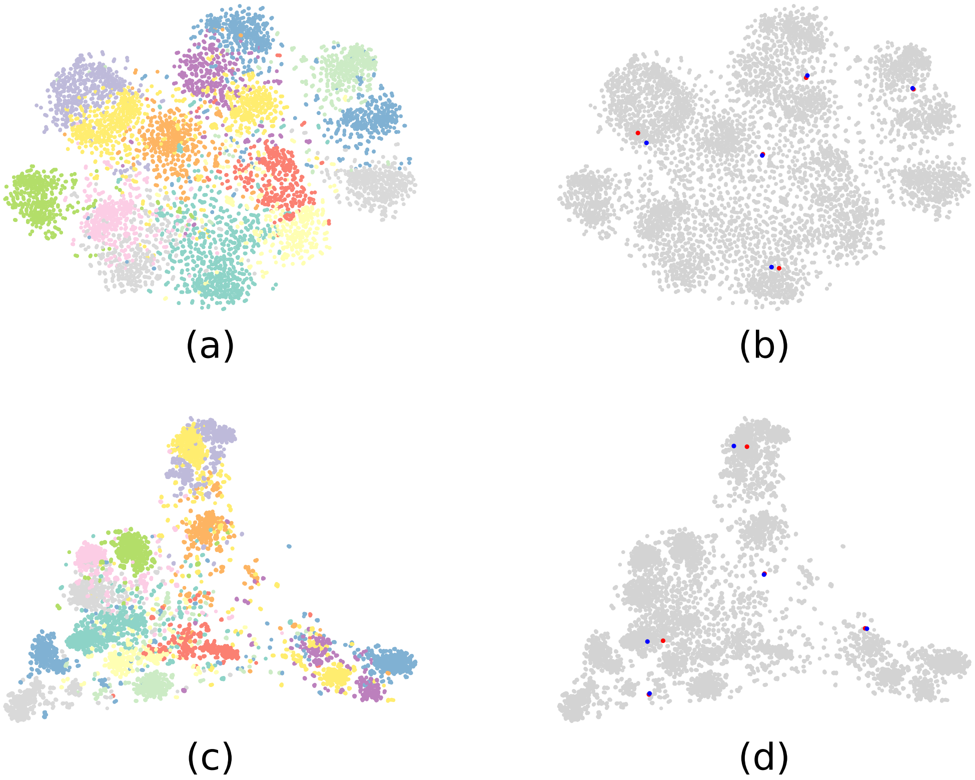


Figure 2. t-SNE visualization of the inputs to g in (a, b) and the generated classification weights in (c, d). Blue and red dots in (b) and (d) are the classification weights for two query samples in the same task. Best view in color.

spite that overall results are still faithful.

References

[1] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 1

[2] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 1

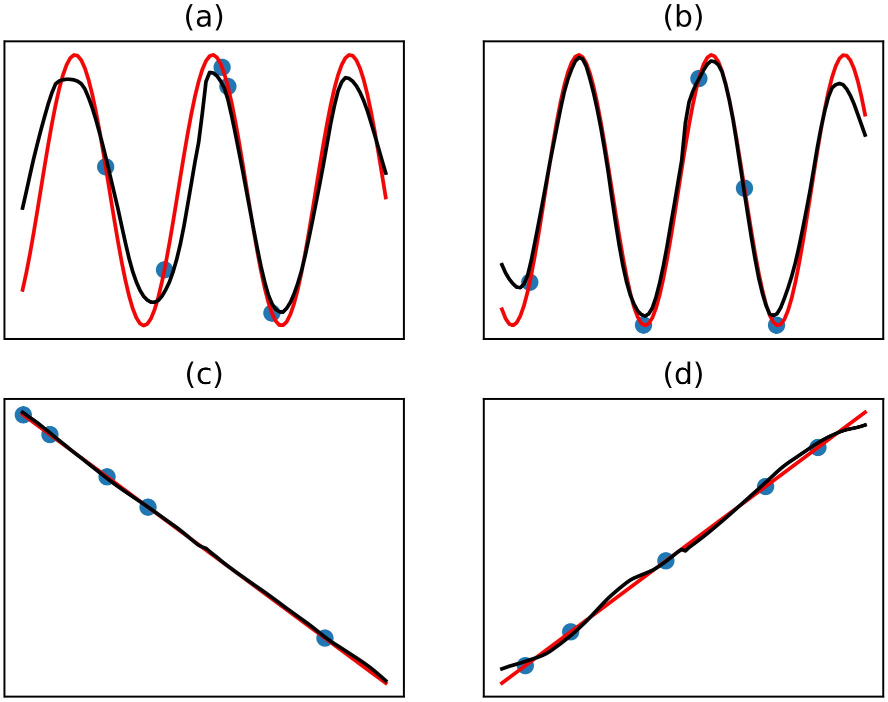


Figure 3. 5-shot regression results for a multi-modal task distribution. Regression targets are plotted in red and prediction in black. 5 training samples per task are plotted with blue solid circles.