Attributable Visual Similarity Learning Supplementary Material

Borui Zhang , Wenzhao Zheng , Jie Zhou , Jiwen Lu*
Department of Automation, Tsinghua University, China
Beijing National Research Center for Information Science and Technology, China
{zhang-br21, zhengwz18}@mails.tsinghua.edu.cn; {jzhou, lujiwen}@tsinghua.edu.cn

A. Implementation Details

A.1. Loss Functions

Loss functions in deep metric learning can be categorized into pair-based methods and proxy-based methods. During training, we apply the proposed AVSL framework to the margin loss [4] and the ProxyAnchor loss [1] as the representative pair-based and proxy-based methods, respectively, to verify the effectiveness.

Margin loss [4] compresses positive pairs while repelling negative pairs in the embedding space as follows:

$$L_{margin} = \frac{1}{|\mathbf{P}|} \sum_{(x,x^+)\in\mathbf{P}} [d(x,x^+) - (\beta_{y_x} - \alpha)]_+ \\ \frac{1}{|\mathbf{N}|} \sum_{(x,x^-)\in\mathbf{N}} [(\beta_{y_x} + \alpha) - d(x,x^-)]_+, \quad (1)$$

where $[\cdot]_+$ is the hinge function (i.e., $[x]_+ = \max\{x, 0\}$) and $d(\cdot,\cdot)$ denotes the Euclidean distance. We use $\mathbf{P}$ and $\mathbf{N}$ to indicate the set of positive pairs and negative pairs and $|\cdot|$ to denote the set of the size. To address the variable intra-class distributions, the margin loss introduce a learnable parameter $\boldsymbol{\beta} \in \mathbb{R}^C$ to adaptively control the range of each class, where C denotes the number of classes. α is a fixed parameter to enforce a large margin between classes.

ProxyAnchor loss [1] instead constrains the relations between proxies and samples as follows:

$$L_{pa} = \frac{1}{|\mathbf{P}^+|} \sum_{p\in\mathbf{P}^+} \log\left(1 + \sum_{x\in\mathbf{X}_p^+} e^{-\alpha(s(x,p)-\delta)}\right) + \\ \frac{1}{|\mathbf{P}|} \sum_{p\in\mathbf{P}} \log\left(1 + \sum_{x\in\mathbf{X}_p^-} e^{\alpha(s(x,p)+\delta)}\right), \quad (2)$$

where α is a scaling factor, δ is the margin, $s(\cdot,\cdot)$ is the cosine similarity function, $\mathbf{P}$ denotes the proxy set, and $\mathbf{P}^+$ denotes the positive proxy set where each proxy has at least one positive samples in the current batch. Also, $\mathbf{X}_p^+$ includes the positive samples for a proxy p and $\mathbf{X}_p^-$ contains the rest negative samples in the batch. However, the original form of ProxyAnchor loss (2) is defined with the cosine similarity, while our proposed AVSL is defined in the context of dissimilarity. To address this, we reformulate the ProxyAnchor loss as follows:

$$L_{pa} = \frac{1}{|\mathbf{P}^+|} \sum_{p\in\mathbf{P}^+} \log\left(1 + \sum_{x\in\mathbf{X}_p^+} e^{\alpha(d(x,p)-(\beta-\tau))}\right) + \\ \frac{1}{|\mathbf{P}|} \sum_{p\in\mathbf{P}} \log\left(1 + \sum_{x\in\mathbf{X}_p^-} e^{-\alpha(d(x,p)-(\beta+\tau))}\right), \quad (3)$$

where $d(\cdot,\cdot)$ indicates the dissimilarity and β and τ control the interclass margin similar to δ in (2).

A.2. Pooling Linearization

To construct the similarity graph, we first employ a CNN to extract the feature map $\mathbf{z} = f(x)$ at each level and then reduce the feature map to a feature vector using pooling operation as $\mathbf{v} = g(\mathbf{z})$. Specifically, we use both max pooling g_{max} and average pooling g_{avg} operations following [1] as follows:

$$v_i = \max_{h,w} z_{ihw} + \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} z_{ihw}. \quad (4)$$

Finally, we adopt a linear layer h to map $\mathbf{v}$ into an embedding space as:

$$\mathbf{e} = h(\mathbf{v}) = (h \circ g)(\mathbf{z}). \quad (5)$$

In addition, we need to compute CAMs [5] as follows:

$$u_{ihw} = h(\mathbf{z}_{\cdot hw}) = \sum_{j=1}^{c} a_{ij} z_{jhw}, \quad (6)$$

*Corresponding author.

Figure 1. Visualization of the similarity inference and attribution. We randomly selected two sample pairs from CUB-200-2011 and Cars196. For each pair of images, we attribute the overall similarity to the specific similarity nodes in an undirected graph. We report the corresponding values of reliabilities, nodes, and sensitivities under each node. Best viewed in color.

where c is the number of channels and a_{ij} indicates the weights of the linear layer h. In order to maintain the spatial information of $\mathbf{e}$, we want to ensure the following property:

$$g(\mathbf{u}) = (g \circ h)(\mathbf{z}) = \mathbf{v}. \tag{7}$$

However, the pooling operation and the linear mapping is not commutative (i.e., $g \circ h \neq h \circ g$) since the pooling operation is a nonlinear function. To address this, we propose a linearization operation $\tilde{g}$ as follows:

$$\tilde{\mathbf{z}}_i = \tilde{g}(\mathbf{z}_i) = \begin{cases} K \cdot z_{ihw}, & \text{if } z_{ihw} = \max\limits_{kl} z_{ikl} \\ 0, & \text{Otherwise} \end{cases} \tag{8}$$

where $K = \frac{HW}{\#\{z_{ihw} | z_{ihw} = \max_{kl} z_{ikl}\}}$. Thus, we can decompose the pooling operation as follows:

$$\begin{aligned} g &= g_{max} + g_{avg} = g_{avg} \circ \tilde{g} + g_{avg} \\ &= g_{avg} \circ (\tilde{g} + \mathbb{I}), \end{aligned} \tag{9}$$

where $\mathbb{I}$ denotes the identity mapping. By employing this linearization trick, we first preprocess the feature map as follows:

$$\tilde{\mathbf{z}} = (\tilde{g} + \mathbb{I})(\mathbf{z}) = \tilde{g}(\mathbf{z}) + \mathbf{z}. \tag{10}$$

We then rewrite (5) and (7) as:

$$\begin{aligned} \mathbf{e} = (h \circ g)(\mathbf{z}) = (h \circ g_{avg})(\tilde{\mathbf{z}}) \\ g_{avg}(\mathbf{u}) = g_{avg}(h(\tilde{\mathbf{z}})) = (g_{avg} \circ h)(\tilde{\mathbf{z}}), \end{aligned}$$

The g_{avg} and h are now commutative (i.e., $g_{avg} \circ h = h \circ g_{avg}$) so that the CAMs can preserve the spatial information of the embeddings.

B. Attribution Property

The proposed AVSL can attribute the overall similarity to specific similarity nodes quantitively as:

$$\begin{aligned} \hat{d} &= \sum_{i=1}^{r} \hat{\delta}_i^L = \mathbf{1}\hat{\boldsymbol{\delta}}^L = \mathbf{1}\mathbf{P}^L\boldsymbol{\delta}^L + \mathbf{1}(\mathbf{I} - \mathbf{P}^L)\tilde{\mathbf{W}}^L\hat{\boldsymbol{\delta}}^{L-1} \\ &= \sum_{l=1}^{L}\sum_{i=1}^{r} \lambda_i^l \delta_i^l, \end{aligned} \tag{11}$$

where $\hat{d}$ is the overall similarity, δ_i^l is the similarity node, and λ_i^l denotes the sensitivity of the corresponding node. λ_i^l represents the influence of the corresponding node on the overall similarity. The sensitivities have the following property:

Property 1. *The sum of λ_i^l of all nodes is a constant.*

Proof. We rewrite (11) as follows:

$$\hat{d} = \sum_{l=1}^{L} \mathbf{1}\boldsymbol{\Lambda}^l\boldsymbol{\delta}^l, \tag{12}$$

where $\boldsymbol{\Lambda}^l = (\mathbf{I} - \mathbf{P}^L)\tilde{W}^L \cdots (\mathbf{I} - \mathbf{P}^{l+1})\tilde{W}^{l+1}\mathbf{P}^l$, and $\boldsymbol{\lambda}^l = [\lambda_1^l \ \lambda_1^l \ \cdots \lambda_r^l] = \mathbf{1}^T\boldsymbol{\Lambda}^l$. Let $\tilde{\boldsymbol{\Lambda}}^l = (\mathbf{I} - \mathbf{P}^L)\tilde{W}^L \cdots (\mathbf{I} -$

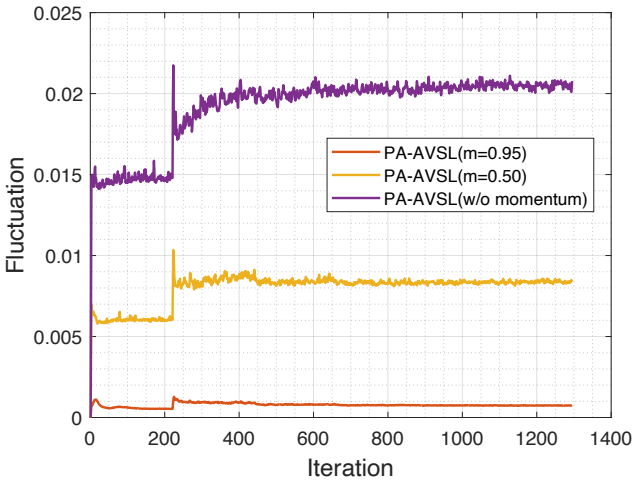


Figure 2. Fluctuation of edges during training.

$\mathbf{P}^{l+1})\tilde{W}^{l+1}$. Since $\tilde{W}^l$ is normalized by row (i.e., $\tilde{W}^l\mathbf{1} = \mathbf{1}$), we can derive that:

$$\begin{aligned}
&(\mathbf{\Lambda}^{l+1} + \tilde{\mathbf{\Lambda}}^l)\mathbf{1} \\
=&(\mathbf{I} - \mathbf{P}^L)\tilde{W}^L \cdots (\mathbf{I} - \mathbf{P}^{l+2})\tilde{W}^{l+2} \\
&\left(\mathbf{P}^{l+1}\mathbf{1} + (\mathbf{I} - \mathbf{P}^{l+1})\tilde{W}^{l+1}\mathbf{1}\right) \\
=&(\mathbf{I} - \mathbf{P}^L)\tilde{W}^L \cdots (\mathbf{I} - \mathbf{P}^{l+2})\tilde{W}^{l+2}\mathbf{1} \\
=&\tilde{\mathbf{\Lambda}}^{l+1}\mathbf{1} \qquad (13)
\end{aligned}$$

Therefore, the sum of λ_i^l is computed as:

$$\begin{aligned}
&\sum_{l=1}^{L}\sum_{i=1}^{r}\lambda_i^l = \sum_{l=1}^{L}\mathbf{1}^T\mathbf{\Lambda}^l\mathbf{1} \\
=&\mathbf{1}^T\left(\sum_{l=2}^{L}\mathbf{\Lambda}^l\mathbf{1} + \tilde{\mathbf{\Lambda}}^1\mathbf{1}\right) \\
=&\mathbf{1}^T\tilde{\mathbf{\Lambda}}^L\mathbf{1} = \mathbf{1}^T\mathbf{1} = r, \qquad (14)
\end{aligned}$$

where r is the dimension of embeddings. □

Property 1 ensures that the absolute value of the sensitive λ_i^l is meaningful across samples and can directly indicate the significance of the corresponding similarity node when inferring the overall similarity.

C. More Experimental Results

C.1. Detailed Visualization

We provide more detailed visualization results of the similarity inferring and attribution. we randomly selected two sample pairs from CUB-200-2011 [3] and Cars196 [2] for similarity attribution, as shown in Figure 1. From top to bottom, we first selected the top-128 reliable nodes with a high p_i^l among all the 512 nodes and further displayed the three most similar nodes framed in green dotted boxes and the three most dissimilar ones framed in red dotted boxes. Subsequently, we decompose one unreliable node to the adjacent related nodes. We quantitatively show the reliabilities p_i^l, similarity nodes δ_i^l, and sensitivities λ_i^l below each box.

We observe that the similarity nodes with higher sensitivity value λ_i^l are more likely positioned in higher layers, which correspond to clearer concepts such as "wing", "head", and "feet" as shown on the left of Figure 1. In addition, patterns of low-level features are relatively difficult to recognize. This demonstrates that high-level features tend to encode object-level concepts while low-level features focus on pixel-level concepts. In addition, we discover that nodes and concepts may not correspond to each other one-to-one. For example, multiple nodes may all focus on the "wheel" part of cars as shown on the right of Figure 1, which indicates that concepts extracted by CNNs are not completely consistent with humans.

Table 1. Ablation study about the edge design.

Method	R@1	R@2	R@4	R@8
PA	87.7	92.9	95.8	97.9
PA-AVSL (w/o MUS)	91.0	94.6	96.7	98.1
PA-AVSL ($\gamma = 0.50$)	91.5	95.0	97.0	**98.4**
PA-AVSL ($\gamma = 0.95$)	**91.6**	**95.2**	**97.2**	**98.4**

Table 2. Ablation study about the reliability design.

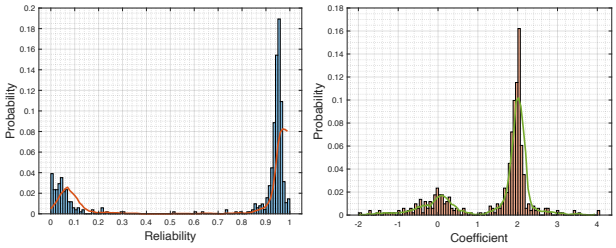
Method	R@1	R@2	R@4	R@8
PA	87.7	92.9	95.8	97.9
PA-AVSL (LR)	90.9	94.6	96.6	98.0
PA-AVSL	**91.5**	**95.0**	**97.0**	**98.4**

C.2. Further Analysis

The strategy of edge construction : Due to the image noise, computing edges only based on a single sample may cause large fluctuation. Therefore we propose that edges should depend on the entire dataset. We adopt the momentum updating strategy to filter the image noise formulated by Equation 4 in the original paper. We plot the fluctuation amplitude curves of edges in Figure 2 and see that the proposed momentum updating strategy (MUS) obtains more stable edges. We also conducted an ablation study to analyze the influence on the performance. Table 1 demonstrates the effectiveness of the proposed strategy.

The design of reliability: We conducted an ablation study about different designs of computing reliability as shown in Table 2, where "PA-AVSL (LR)" represents learning the reliability by a fully-connected layer (i.e., $\eta_i^l = h(\hat{u}_i^l)h(\hat{u}_i'^l)$). The comparison demonstrates that a priori design is more effective than a learning-based one.

The effectiveness of reliability detection: We show the distribution of the reliability in Figure 3a and observe that



(a) Reliability (b) Coefficient

Figure 3. Frequency histogram.

only a few significantly unreliable nodes will trigger the top-down rectification. We further show the distribution of the coefficient of the sigmoid regression as Equation 5 in Figure 3b. The sigmoid regression acts like a filter and assign a small number of inaccurate reliability estimation with small coefficients close to zero.

References

[1] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, pages 3238–3247, 2020. 1

[2] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561, 2013. 3

[3] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 3

[4] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, pages 2840–2848, 2017. 1

[5] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 1