Supplementary Material of Domain Separation Graph Neural Networks for Saliency Object Ranking

Zijian Wu[1] Jun Lu[1] Jing Han[1] Lianfa Bai[1] Yi Zhang[1] Zhuang Zhao[1,†] Siyang Song[2,†]
[1]Nanjing University of Science and Technology [2]University of Leicester
{wuzijian, lujunchenhao, eohj, blf, zhy441}@njust.edu.cn
zhaozhuang3126@gmail.com, ss1535@leicester.ac.uk

1. Training Details

To train our DSGNN, the overall loss function consists of three parts, including task-specific loss $\mathcal{L}_{\text{task}}$, loss $\mathcal{L}_{\text{st}}$ for the STGDS module and loss $\mathcal{L}_{\text{ci}}$ for the CIGDS module, which is formulated as:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{task}} + \beta\mathcal{L}_{\text{st}} + \lambda\mathcal{L}_{\text{ci}} \tag{1}$$

where α, β and λ are weights to describe the relative importance of these three loss parts.

$\mathcal{L}_{\textbf{task}}$: The task-related loss $\mathcal{L}_{\text{task}}$ jointly supervises the model to generate the final saliency predictions, which is a combination of segmentation loss, restoration loss, classification loss and saliency ranking loss as:

$$\mathcal{L}_{\text{task}} = \mathcal{L}_{\text{seg}} + \mathcal{L}_{\text{restor}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{rk}} \tag{2}$$

Within this framework, $\mathcal{L}_{\text{seg}}$ trains the model to predict the shape predictions from the shape head. Specifically, $\mathcal{L}_{\text{seg}}$ follows the same settings in Mask2former [2], consisting of a combination of the binary cross-entropy loss and the Dice loss [4]:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{bce}}(S, \widehat{S}) + \mathcal{L}_{\text{dice}}(S, \widehat{S}) \tag{3}$$

where S represents the shape predictions from the Shape Head, and $\widehat{S}$ denotes the shape ground truth of the saliency objects in images.

$\mathcal{L}_{\text{restor}}$ is applied for the restoration of texture predictions from the texture head using the Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{restor}} = \mathcal{L}_{\text{mse}}(T, \widehat{T}) \tag{4}$$

where T represents the texture predictions from the Texture Head, and $\widehat{S}$ denotes the texture ground truth of the saliency objects in images.

$\mathcal{L}_{\text{cls}}$ serves to discern whether predictions in class head qualifies as a saliency object. Follow the Mask2former, we employ the cross-entropy loss to this process:

$$\mathcal{L}_{\text{cls}} = \mathcal{L}_{\text{ce}}(sc, \widehat{sc}) \tag{5}$$

where sc is the classification predictions of all objects form the Class Head, and $\widehat{sc}$ is the ground truth for classification, including two states (0,1). Here, 0 signifies salient objects, while 1 represents non-salient objects.

$\mathcal{L}_{\text{rk}}$ is guided by the saliency ranking loss [3] for ranking the saliency level of predictions in the rank head. The specific supervision process can be described as:

$$\mathcal{L}_{\text{rk}} = \mathcal{L}_{\text{rk}}(sr, \widehat{sr}) \tag{6}$$

where sr denotes the saliency score predictions of all objects from the Rank Head, and $\widehat{sr}$ represents the ground truth for saliency. Among them, a higher numerical value indicates a higher degree of saliency for the object. For instance, in the ASSR dataset [5], the object with label ”5” represents the most saliency level, whereas in the IRSR dataset [3], the value ”8” signifies the highest saliency level for a object.

$\mathcal{L}_{\textbf{st}}$: $\mathcal{L}_{\text{st}}$ denotes the loss used for the STGDS module, which is a combination of the differnece loss, the similarity loss, and the reconstruction loss:

$$\mathcal{L}_{\text{st}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{sim}} \tag{7}$$

where $\mathcal{L}_{\text{recon}}$ is the MSE loss, $\mathcal{L}_{\text{diff}}$ utilizes the loss from [1], and $\mathcal{L}_{\text{sim}}$ is the Pearson correlation loss, where Pearson similarity can be expressed as:

$$\mathcal{P}_{\text{sim}}(x, y) = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2)(\sum_{i=1}^{n} y_i^2 - \frac{1}{n}(\sum_{i=1}^{n} y_i)^2)}} \tag{8}$$

Here, the Pearson correlation loss is defined as:

$$\mathcal{L}_{\text{pearson}}(x, y) = 1 - \mathcal{P}_{\text{sim}}(x, y) \tag{9}$$

where $x, y \in \mathbb{R}^{1\times n}$ represent two vectors used to measure similarity.

To separate the feature representations as intended, we apply a difference loss between $\widehat{V}_p^s$ and $\widehat{V}_c^s$, as well as between $\widehat{V}_p^t$ and $\widehat{V}_c^t$ to encourage the divergence of saliency degree-related and unrelated representations. The process

[†] Corresponding author.

can be formulated as:

$$\mathcal{L}_{\text{diff}} = \mathcal{L}_{\text{diff}}(\widehat{V}_p^s, \widehat{V}_c^s) + \mathcal{L}_{\text{diff}}(\widehat{V}_p^t, \widehat{V}_c^t) \tag{10}$$

Additionally, a similarity loss is employed between $\widehat{V}_c^s$ and $\widehat{V}_c^t$ for maintaining the similarity of the shared-domain representations:

$$\mathcal{L}_{\text{sim}} = \mathcal{L}_{\text{pearson}}(\widehat{V}_c^s, \widehat{V}_c^t, 1) \tag{11}$$

Finally, to prevent the model from producing trivial solutions, we introduce a reconstruction loss between between $\widehat{V}_r^s$ and $\widehat{V}^s$, as well as between $\widehat{V}_r^t$ and $\widehat{V}^t$:

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{mse}}(\widehat{V}_r^s, \widehat{V}^s) + \mathcal{L}_{\text{mse}}(\widehat{V}_r^t, \widehat{V}^t) \tag{12}$$

$\mathcal{L}_{\textbf{ci}}$: $\mathcal{L}_{\text{ci}}$ is the loss function in the CIGDS module. To ensure the successful separation of the desired shared representations during this process, we also apply a similarity loss in $\{V_i^{\text{task}}\}_i^B$ to facilitate the learning of similar distributions, as:

$$\mathcal{L}_{\text{sim}} = \sum_{i,j=1,i\neq j}^{B} \mathcal{L}_{\text{pearson}}(\widehat{V}_i^{\text{task}}, \widehat{V}_j^{\text{task}}, 1) \tag{13}$$

At the same time, we employ a difference loss in $\{V_i^{\text{noise}}\}_i^B$ to compel distinct features between the saliency degree-related and unrelated domain, ensuring the prevention of trivial solutions, as:

$$\mathcal{L}_{\text{diff}} = \sum_{i,j=1,i\neq j}^{B} \mathcal{L}_{\text{diff}}(\widehat{V}_i^{\text{noise}}, \widehat{V}_j^{\text{noise}}) \tag{14}$$

where $\mathcal{L}_{\text{diff}}$ and $\mathcal{L}_{\text{sim}}$ have equivalent roles to the respective parts in $\mathcal{L}_{\text{st}}$.

Finally, $\mathcal{L}_{\text{ci}}$ is the combination of $\mathcal{L}_{\text{diff}}$ and $\mathcal{L}_{\text{sim}}$:

$$\mathcal{L}_{\text{ci}} = \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{sim}} \tag{15}$$

2. Analysis of significant differences for STGDS and CIGDS

To further validate the effectiveness of STGDS module and CIGDS module, we conduct multiple sets of experiments to analyze the significant differences between STGDS and CIGDS modules. As illustrated in Fig. 1, we sequentially recorded the results: Baseline results utilizing only shape information, Baseline results employing texture information, results of the shape branch after incorporating the STGDS module, results of the texture branch after integrating the STGDS module, results of merging shape and texture branches after integrating the STGDS module, and results after incorporating the CIGDS module. From the figure, it is evident that there are significant differences in the results after the incorporation of the STGDS module compared to the Baselines. This signifies that the enhancements brought about by our module are consistent and reliable. Moreover, upon the inclusion of the CIGDS module, there are also significant differences compared to the results obtained with the STGDS module. These experimental observations indicate the stability of the enhancements brought about by our modules.

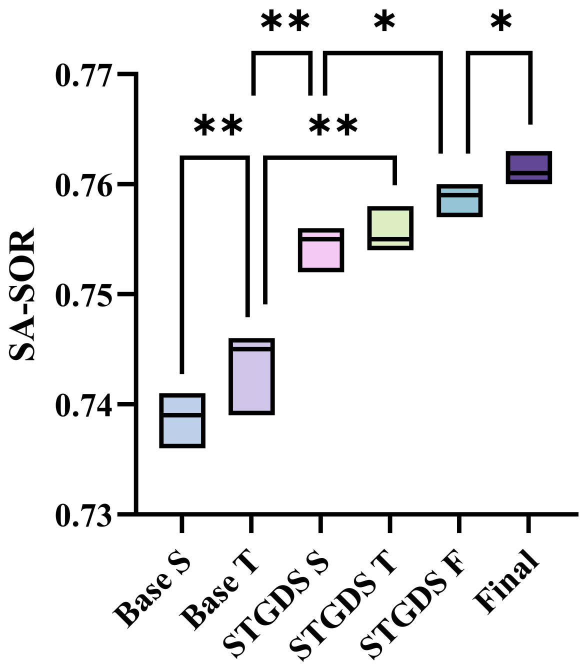


Figure 1. The significant analysis diagrams for STGDS and CIGDS. From left to right, the results are presented for the baseline using the shape information, baseline using texture information, the results of the shape branch after incorporating the STGDS module, the results of texture branch after incorporating the STGDS module, the results of combining shape and texture information, and the final results after adding the CIGDS module. $*$ denotes $p < 0.05$ while $**$ means $p < 0.01$.

3. Analysis of multi-dimensional edges

Our DSGNN starts with a pair of transformer decoders to extract the shape queries and the texture queries of all objects, and then we use the STGDS to first encode these queries as the shape graph and the texture graph, respectively. As shown in Fig. 2, we present a detailed demonstration of the topology of the constructed graphs. Taking the ASSR dataset [5] as an example, we built graphs for each image with 5 nodes and 25 edges. Between any two nodes, there are two reciprocally directed edges, and notably, each node has a self-directed edge for updating its self-relationship. In Fig. 3 and Fig. 4, we respectively visualize the multi-dimensional edges before and after passing through the shared GNN Encoder (GencC) in the STGDS module. Fig. 3 depicts the visualization results of multi-dimensional edges in GencT and GEncS, while Fig. 4 illustrates the display of multi-dimensional edges in GencC

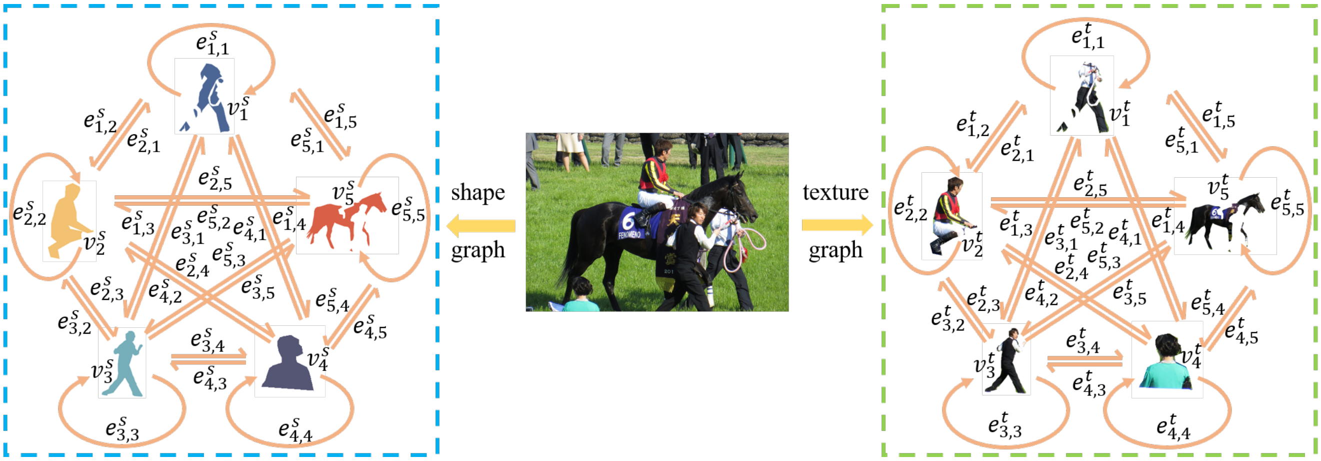


Figure 2. Details of the shape graph topology and the texture graph topology.

utilizing shape and texture information, respectively. It is evident that there are notable differences among the edges connecting different nodes (particularly nodes with different saliency levels), indicating that our multi-dimensional edges can capture diverse relationships between different nodes.

Additionally, we highlight the differences between the edges of the shape graph and the texture graph. Fig. 5a illustrates the absolute value differences of the multi-dimensional edges between the GencT and GencS. The clear differences in edges suggest that the relationships between targets' shapes and textures are different. These differences also indicate that although the shape and texture information of the targets are crucial for determining their saliency, there still exist saliency-irrelevant noises in them. Influenced by these noises, utilizing different information for inferring relationships between targets may yield disparate outcomes. Ultimately, this adversely affects the model's ability to assess targets' saliency degree. Fig. 5b displays the absolute value differences of the edges between the shape and texture graphs after passing through the shared GNN encoder (GencC). It is apparent that after GencC processing, the relationships between the graphs become more similar, indicating that our STGDS module achieved its intended objective by extracting common relationships relevant to saliency from both the shape and texture graphs.

References

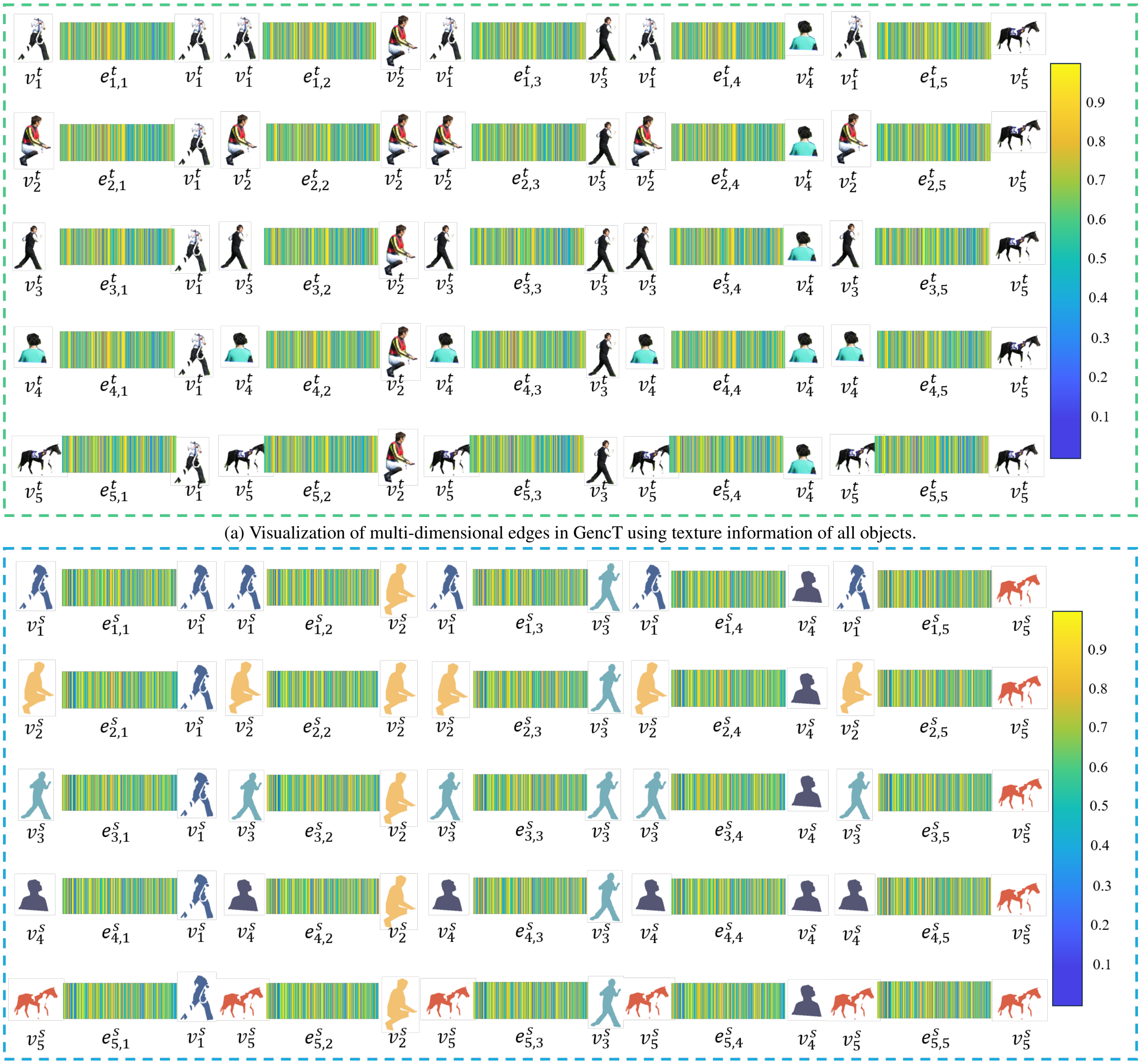
[1] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016. 1

[2] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 1

[3] Nian Liu, Long Li, Wangbo Zhao, Junwei Han, and Ling Shao. Instance-level relative saliency ranking with graph reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8321–8337, 2021. 1

[4] F Milletari, N Navab, SAVN Ahmadi, and V Net. Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. 1

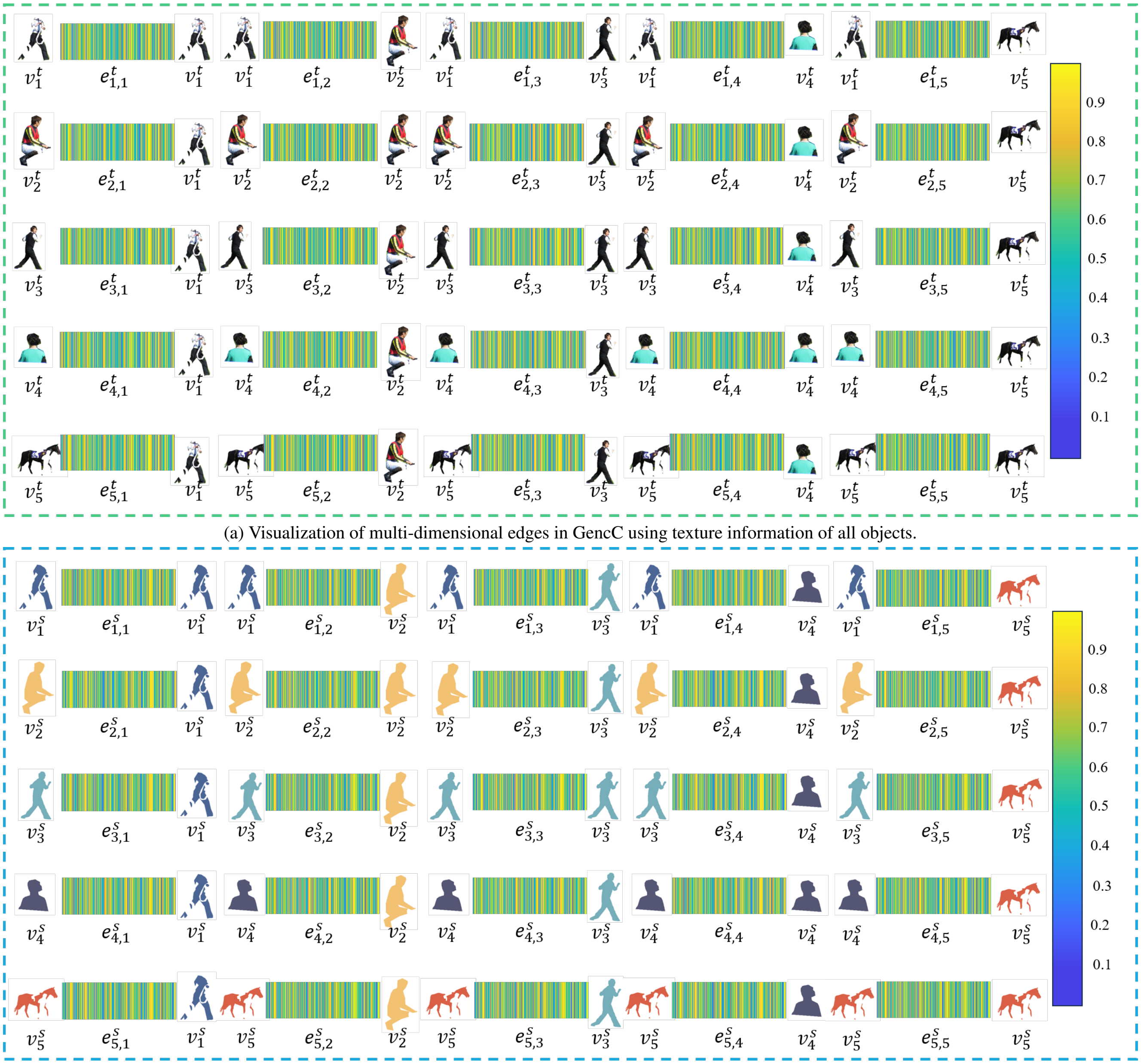
[5] Avishek Siris, Jianbo Jiao, Gary KL Tam, Xianghua Xie, and Rynson WH Lau. Inferring attention shift ranks of objects for image saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12133–12143, 2020. 1, 2



(a) Visualization of multi-dimensional edges in GencT using texture information of all objects.

(b) Visualization of multi-dimensional edges in GEncS using shape information of all objects.

Figure 3. Visualization of multi-dimensional edges in GencT and GEncS.

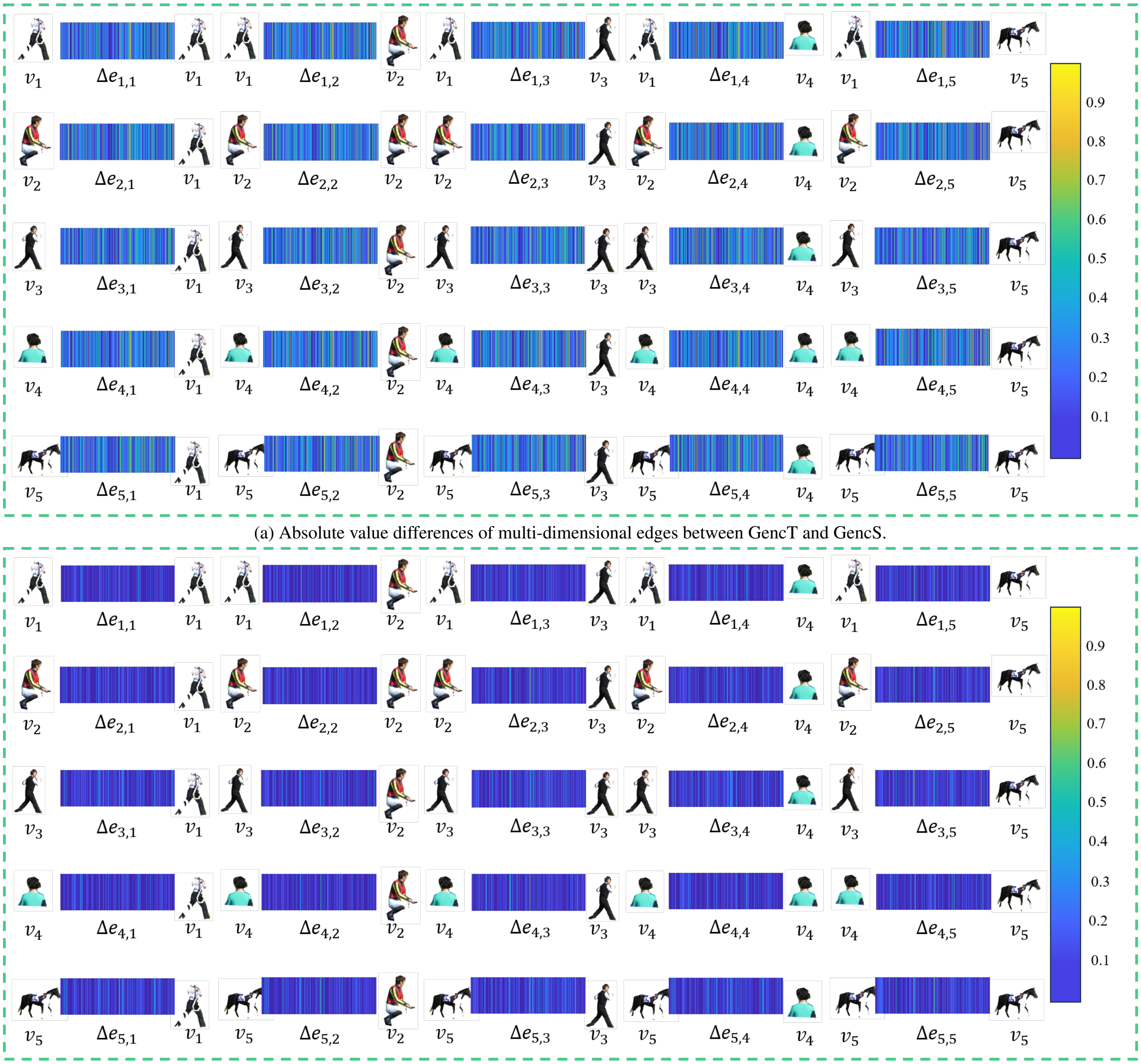

v_1^t $e_{1,1}^t$ v_1^t v_1^t $e_{1,2}^t$ v_2^t v_1^t $e_{1,3}^t$ v_3^t v_1^t $e_{1,4}^t$ v_4^t v_1^t $e_{1,5}^t$ v_5^t
v_2^t $e_{2,1}^t$ v_1^t v_2^t $e_{2,2}^t$ v_2^t v_2^t $e_{2,3}^t$ v_3^t v_2^t $e_{2,4}^t$ v_4^t v_2^t $e_{2,5}^t$ v_5^t
v_3^t $e_{3,1}^t$ v_1^t v_3^t $e_{3,2}^t$ v_2^t v_3^t $e_{3,3}^t$ v_3^t v_3^t $e_{3,4}^t$ v_4^t v_3^t $e_{3,5}^t$ v_5^t
v_4^t $e_{4,1}^t$ v_1^t v_4^t $e_{4,2}^t$ v_2^t v_4^t $e_{4,3}^t$ v_3^t v_4^t $e_{4,4}^t$ v_4^t v_4^t $e_{4,5}^t$ v_5^t
v_5^t $e_{5,1}^t$ v_1^t v_5^t $e_{5,2}^t$ v_2^t v_5^t $e_{5,3}^t$ v_3^t v_5^t $e_{5,4}^t$ v_4^t v_5^t $e_{5,5}^t$ v_5^t
0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1

(a) Visualization of multi-dimensional edges in GencC using texture information of all objects.

v_1^s $e_{1,1}^s$ v_1^s v_1^s $e_{1,2}^s$ v_2^s v_1^s $e_{1,3}^s$ v_3^s v_1^s $e_{1,4}^s$ v_4^s v_1^s $e_{1,5}^s$ v_5^s
v_2^s $e_{2,1}^s$ v_1^s v_2^s $e_{2,2}^s$ v_2^s v_2^s $e_{2,3}^s$ v_3^s v_2^s $e_{2,4}^s$ v_4^s v_2^s $e_{2,5}^s$ v_5^s
v_3^s $e_{3,1}^s$ v_1^s v_3^s $e_{3,2}^s$ v_2^s v_3^s $e_{3,3}^s$ v_3^s v_3^s $e_{3,4}^s$ v_4^s v_3^s $e_{3,5}^s$ v_5^s
v_4^s $e_{4,1}^s$ v_1^s v_4^s $e_{4,2}^s$ v_2^s v_4^s $e_{4,3}^s$ v_3^s v_4^s $e_{4,4}^s$ v_4^s v_4^s $e_{4,5}^s$ v_5^s
v_5^s $e_{5,1}^s$ v_1^s v_5^s $e_{5,2}^s$ v_2^s v_5^s $e_{5,3}^s$ v_3^s v_5^s $e_{5,4}^s$ v_4^s v_5^s $e_{5,5}^s$ v_5^s
0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1

(b) Visualization of multi-dimensional edges in GencC using shape information of all objects.

Figure 4. Visualization of multi-dimensional edges in GencC.



(a) Absolute value differences of multi-dimensional edges between GencT and GencS.

(b) Absolute value differences of multi-dimensional edges in GencC.

Figure 5. Visualization of absolute difference of multi-dimensional edges.