1. MATHEMATICAL DETAILS ON OUR APPROACH

1.1. Likelihood \( L(\tau; \theta, \pi) \)

We define \( L(\tau; \theta, \pi) \) as the likelihood of a standard collage tree \( \tau \) given canvas width \( w \), canvas height \( h \) and image collection \( \{I_i\} \), where \( \pi \) is the nearest neighbor policy (NNP) and \( \theta \) is the parameters of the tree generator. \( L(\tau; \theta, \pi) \) can be calculated using our probability space as

\[
L(\tau; \theta, \pi) = p(\tau|w, h, \{I_i\}; \theta, \pi) = p(\tau|\tau_0, w, h, \{I_i\}; \theta, \pi) p(\tau_0|w, h, \{I_i\}) = p(\tau|\tau_0) = p(\tau|\tau_0)
\]  

where \( \tau_0 \) is the probabilistic collage tree, i.e., PCtree.

1.2. Gradient \( \nabla_\theta \bar{F}_\theta(\tau; \pi) \)

We define \( \mathbb{E}_{\tau \sim L(\tau; \theta, \pi)} \left[ F\left(g\left(\tau\right)\right)\right] \) as \( \bar{F}_\theta(\tau; \pi) \) where \( g \) is the mapping function and \( F \) is a criterion function. We approximate the gradient as

\[
\nabla_\theta \bar{F}_\theta(\tau; \pi) = \nabla_\theta \left( \sum_{\tau \sim L(\tau; \theta, \pi)} F\left(g(\tau)\right)L(\tau; \theta, \pi) \right)
\]

\[
= \sum_{\tau \sim L(\tau; \theta, \pi)} F\left(g(\tau)\right) \nabla_\theta L(\tau; \theta, \pi)
\]

\[
= \sum_{\tau \sim L(\tau; \theta, \pi)} F\left(g(\tau)\right) L(\tau; \theta, \pi) \nabla_\theta \log L(\tau; \theta, \pi)
\]

\[
\approx \frac{1}{M} \sum_{i=1}^{M} F\left(g(\tau_i)\right) \nabla_\theta \log L(\tau; \theta, \pi)
\]

\[
= \nabla_\theta \left( \frac{1}{M} \sum_{i=1}^{M} F\left(g(\tau_i)\right) \log p(\tau|\tau_0) \right)
\]  

where \( M \) is the number of sample \( \tau_i \).

1.3. Inference

At inference stage, the optimal collage tree \( \tau^* \) is determined by the maximum likelihood method as

\[
\tau^* = \arg \max_{\tau} p(\tau|\tau_0)
\]  

Notably the structural parameter inference of each node is independent of each other. Moreover, the nodes in the PCtree and standard collage tree are in one-to-one correspondence. Thus, given a node \( \tilde{n} \) in the PCtree, the corresponding node \( n \) in the standard collage tree is determined once the cut type \( c_n \) is predicted via

\[
c_n^* = \arg \max_{c_n \in \{V, \cdot\}^*} p_{\tilde{n}}(1 | c_n = \cdot) \tilde{n} \)
\]  

After the node inference, given a node \( \tilde{n} \) with sub-nodes \( \tilde{n}_i \) and \( \tilde{n}_j \) in the PCtree, the corresponding nodes \( n_i \) and \( n_j \) in the standard collage tree are determined. Hence, the
edge connection relationship is determined by
\[ l'_n = \arg \max_{l_n \in \{n, n\}} p_c(0)(l'_n, \{\tilde{n}_i, \tilde{n}_j\} \setminus l_n) \quad (5) \]
\[ r^*_n = \{n_i, n_j\} \setminus l'_n \quad (6) \]
where \(l_n\) is the left node of node \(n\), \(r_n\) is the right node
of node \(n\), \(l'_n\) is the PCtree node corresponding to \(l_n\) and
\(\{n_i, n_j\} \setminus x\) denotes the element that is not equal to \(x\) in the
binary set \(\{n_i, n_j\}\).

2. EXPERIMENTAL DETAILS AND MORE RESULTS

2.1. The Image Collection Sampling Framework

The image collection sampling framework is designed to generate AIC from the ICSS. This framework samples train set and test set of AIC respectively from train set and test set of ICSS. Each image collection sampled by the framework satisfies the following constraints:

1. Images in the collection share the identical theme of the ICSS.
2. The collection includes images from at least two categories.
3. There are at least two images in each category in the collection.
4. The category distribution of the collection conforms to uniform distribution and is not biased by the prior category distribution in the ICSS.

where the second and the third constraints are established for acquiring effective \(M_n\) value.

Our framework samples image collections under each collection size and theme for multiple times through the enumeration method. Hence, our framework has two parameters, where one is the list of the collection sizes defined as \(\{s_i\}\) and the other is the maximum number of sampling defined as \(T\). The framework is described in Algo. 1, where \(\binom{1}{C} / 2\) in the seventh line is introduced to prevent sampling identical collections when \(s\) is close to 0 or equivalent to \(|C|\) and the constraints in the tenth line is used to satisfy the second and the third constraint above. To solve the problem of Eq. (7) under the above four constraints, we develop a distribution-balanced sampling algorithm, which is parameterized by \(s\), i.e., collection size, and \(\{c_i\}_{m_i}\), i.e., list of the number of images of each category, where \(m_{th}\) denotes the number of categories in theme \(th\). The algorithm is described in Algo. 2, where notably in the second line \(\min(m_{th}, \lfloor s/2 \rfloor)\) is used to satisfy the third constraint and \(\max(2, \lceil \frac{s}{\min(n \leq s \leq m_{th}, c_i) \rceil} \) is used to satisfy the second constraint and to ensure that there are always at least \(s\) images in any \(m\) categories. Notably \(n\) should not be less than 4 for using this algorithm.

We set \(\{s_i\}\) to \(\{10, 15, 20, 25, 30, 50, 100\}\), and \(T\) is set to 10 and 2 for the train set and the test set respectively. As a result, the train set has 562 image collections including 18535 images and the test set has 62 image collections including 1260 images.

2.2. Implementation Details

We implemented the proposed framework using the PyTorch toolbox \[8\] on one GeForce RTX 3090 GPU. We set the canvas width \(w\) to 900 and height \(h\) to 600. The ResNet-50 \[4\] pre-trained on the ImageNet \[3\] with \(d_{bb}\) equaling 2048 is adopted as the backbone network in our feature extractor. The information embedding hyperparameters \(d_w\) and \(d_h\) are both set to 4, and \(d_{ar}\) and \(d_{in,f}\) are both set to 128. We set \(d_Q\) and \(d_K\) both to 1024, and set \(d_V = d_{bb} + d_{in,f}\). We set \(d_1\) and \(d_2\) to 512 and 2 respectively according to \[6\]. With respect to the loss function, we set \(M\) to 1000 and \(R_e\) to 2. Moreover, \(r_1\), \(r_2\) and \(r_3\) are set to \(10^{-3}\), 0.1 and 0.5 respectively. Notably we set \(s_p\) to 20.0 for \(F_p\). The three criteria are added into \(F\) with \(\lambda_r = \lambda_p = 1.0\) and \(\lambda_a = 0.05\). To train our model, we use the Adam optimizer \[5\] with an initial learning rate \(\alpha\) of \(10^{-4}\) for each image collection and \(T_m\) is set to 100.

2.3. More Ablation Analysis

The self-attentive embedding in the fusion module. To study the necessity of the Eq (10) and Eq (11) in the main paper, we conduct some experiments of the ablated versions of the Eq (10) (i.e. average pooling and one fully connected layer) and the ablated version of the Eq (11) (i.e. identity transform) on the train set of the AIC. The results are presented in Tab. 1.

The distance criterion of the NNP. Since the distance criterion of the NNP has high effects on the model performance, we compare different distance criteria on the train set of the AIC, including Euclidean distance, standardized Euclidean distance and the cosine distance. The results are shown in Tab. 2. Notably we define the cosine distance of the two features \(f_i\) and \(f_j\) as \(D_{cos} = 1 - \frac{\langle f_i, f_j \rangle}{\|f_i\|\|f_j\|}\).

2.4. Human Evaluation

Firstly we carried out the 5-scale evaluation. Fifteen human raters participated in the evaluation and each of them was shown with 16 groups of collages. Each group includes four collages, i.e. one generated by our method and three by the baseline methods. The raters were asked to watch the collages for at least 20 s and rated them from ‘Excellent’ (4) to ‘Bad’ (0). To measure the gain in our method over the baselines, we also conducted the side-by-side evaluation. This comparative task is easier than 5-scale rating task for human and thus can produce more reliable results. Thirty raters participated in the evaluation and they were equally divided into three groups. Each group compares our
Algorithm 1: The image collection sampling framework

Input: ICSS, \{s_i\}, T
Output: AIC

1. \(\text{AIC} \leftarrow \{\}\)
2. for every theme \(th\) in the ICSS do // enumerate theme
3.   Calculate \(m_{th}\), i.e. the number of categories in theme \(th\)
4.   \(C \leftarrow \{c_i\}_{i=1}^{m_{th}} \) // \(c_i\) is the number of images in the \(i\)-th category
5.   \(|C| \leftarrow \sum_{i=1}^{m_{th}} c_i\)
6. for every \(s \in \{s_i\} \cap [0, |C|]\) do // enumerate collection size
7.   Sample \(t \sim \text{DiscreteU}(1, \ldots, \min \left(\binom{|C|}{s}/2, T\right)) \) // \(\text{DiscreteU}\) is discrete uniform distribution and \(\binom{n}{k}\) denotes combinatorial number
8.   \(t_0 \leftarrow 0\)
9. repeat // sample for \(t\) times
10.   Randomly sample out \(\{x_i\}_{i=1}^{m_{th}}\) to generate the image collection \(S_{t_0}^{(th,s)}\) s.t.
11.   \(\sum_{i=1}^{m_{th}} x_i = s,\)
12.   \(x_i \in [0, c_i] \cap \mathbb{Z} \setminus \{1\},\)
13.   \(\exists i \neq j \ x_i x_j > 0\) (7)
14.   Add \(S_{t_0}^{(th,s)}\) into AIC
15.   \(t_0 \leftarrow t_0 + 1\)
16. until \(t_0 = t\)
17. end
18. return AIC

Table 1. Ablation analysis of the self-attentive embedding in the fusion module on the train set of AIC. Here, \(W_s \in \mathbb{R}^{d_2 \times d_V}\) is a learnable parameter.

<table>
<thead>
<tr>
<th>Version of the Eq (10)</th>
<th>Version of the Eq (11)</th>
<th>(M_r)</th>
<th>(M_n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>Original</td>
<td>1.086</td>
<td>0.284</td>
</tr>
<tr>
<td>(A = \text{softmax}(W_s f'_{(i,j)}))</td>
<td>Original ((d_2 = 1))</td>
<td>1.282</td>
<td>0.273</td>
</tr>
<tr>
<td>(A = \text{softmax}(\text{avg.pooling}(f'_{(i,j)})))</td>
<td>Identity transform</td>
<td>1.657</td>
<td>0.259</td>
</tr>
</tbody>
</table>

Table 2. Comparison of different distance criteria of the NNP on the train set of AIC.

<table>
<thead>
<tr>
<th>Distance criterion</th>
<th>(M_r)</th>
<th>(M_n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>1.086</td>
<td>0.284</td>
</tr>
<tr>
<td>Standardized Euclidean</td>
<td>18.459</td>
<td>0.213</td>
</tr>
<tr>
<td>Cosine</td>
<td>1.179</td>
<td>0.291</td>
</tr>
</tbody>
</table>

2.5. Results

Due to the limited space of the main paper, the collages in the main paper are reduced to a small size and only some of the results are presented. To illustrate the superiority of our method more clearly, we here show more results in Figs. 1 to 11.
Algorithm 2: The distribution-balanced sampling algorithm

Input: \( s, \{c_i\}_{mth} \)

Output: \( \{x_i\}_{mth} \)

1. \( \{x_i\}_{mth} \leftarrow \{0\}_{mth} \)

2. Sample \( m \sim \text{DiscreteU} \left( \min(\max(2, \frac{s}{\min_{1 \leq i \leq mth} c_i}), m_{th}, \lfloor s/2 \rfloor), \ldots, \min(m_{th}, \lfloor s/2 \rfloor) \) \) // \( m \) is the number of the sampled categories and \( \text{DiscreteU} \) is discrete uniform distribution

3. Randomly sample out \( \{i_s\}_m \) from \( \{1, \ldots, m_{th}\} \) // \( \{i_s\}_m \) denotes the list of the sampled categories

4. for every \( i_s \in \{i_s\}_m \) do

5. \( x_{i_s} \leftarrow 2 \) // ensure that there are at least two images in each category

6. end

7. \( k \leftarrow 2m \) // \( k \) denotes the number of the sampled images

8. repeat

9. \( \delta \leftarrow \{\} \) // \( \delta \) denotes the list of the categories that have images for sampling

10. for every \( i_s \in \{i_s\}_m \) do

11. if \( x_{i_s} < c_{i_s} \) then // \( c_{i_s} \) is the number of images in the \( i_s \)-th category

12. Add \( i_s \) into \( \delta \)

13. end

14. end

15. Randomly sample \( i_s^* \) from \( \delta \)

16. \( x_{i_s^*} \leftarrow x_{i_s^*} + 1 \)

17. \( k \leftarrow k + 1 \)

18. until \( k = s \)

19. return \( \{x_i\}_{mth} \)

References


IEEE transactions on visualization and computer graphics, 2019. 5, 6, 7, 8, 9, 10

Figure 1. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.

Figure 2. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.
Figure 3. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.

Figure 4. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.
Figure 5. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.

Figure 6. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.
Figure 7. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.

Figure 8. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.
Figure 9. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.

Figure 10. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.
Figure 11. Comparison of the collage results generated by different methods on the AIC. The image content occlusion and severe aspect ratio distortion are both highlighted by red dotted rectangle.