ABSTRACT Self-attention mechanisms are widely used in current encoder-decoder frameworks of image-captioning tasks. These mechanisms use the transformer module as the basic unit to compute the similarity between query and key vectors, and normalize them to obtain the attention distribution matrix, weighting and reconstructing the rich semantic query vectors. However, a single attention distribution matrix cannot express more complex intrinsic relations between query and key vectors. Some very important intrinsic relations can thus become lost. We therefore propose a hybrid attention distribution (HAD) that allows multiple distributions to be reconstructed to express deeper internal relations and avoids a single shallow attention distribution. In addition, to reduce the number of model parameters, we factorize the word-embedding matrix. This effectively increases the training efficiency and prevents the metrics from decreasing. We extensively evaluate our proposals using the MS-COCO image-captioning dataset. Our results outperform existing state-of-the-art methods on some metrics.

INDEX TERMS Self-attention, image captioning, transformer, encoder-decoder.
with the replacement of the original single distribution that makes the image-captioning model relate to each subspace. The FEM can reduce the number of model parameters and effectively reduce the training time by decomposing the large vocabulary-embedding matrix into two small matrices. First, a word is embedded within a lower dimensional space, and another matrix is then used to restore the original dimensional space. The advantage of this operation is that it reduces the number of parameters without reducing the model performance.

The main contribution of this work is the proposal of a HAD and a FEM modules. The former can comprehensively consider multiple different weight distributions in the subspace to obtain a more expressive hybrid distribution. The aim of the FEM is to reduce the number of parameters in a large word-embedding matrix to make the model converge faster and without affecting the model performance.

II. RELATED WORK
A. IMAGE CAPTION
Traditional approaches to describing images are template-based. These involve firstly generating slotted caption templates and filling them using detected objects and attributes [16]–[18]. These methods are relatively efficient in terms of linguistic logic, but they have major drawbacks in terms of diversity. Recently, the neural-based method successfully performed image captioning by applying the encoder-decoder framework, whereby a CNN is first used to encode an image and a RNN is adopted to decode the output words [1]. Soft and hard attentions [10] were introduced subsequently to focus the model on an important area of the image in each time step. Semantic attention [7] was presented to learn to focus selectively on the semantic attributes in an image for sentence generation. An adaptive attention model [9] was proposed to decide whether to attend to image regions at the time of decoding. Furthermore, bottom-up and top-down attention mechanisms [11] can be exploited to focus on objects rather than image regions. A spatial and channel-wise attention model [12] was proposed, and some RL methods [19], [20] were introduced to train the model.

B. TRANSFORMER-BASED METHOD
Since method [15] was introduced, many methods [21]–[24] based on it have been used in natural language processing, including for application [14] to image captioning. The transformer structure has also undergone many improvements [25]–[28]. Owing to the considerable number of model parameters of the transformer, some studies [24], [29], [30] have investigated ways to reduce it while ensuring stable performance. Our work draws on their ideas [24], [28] to investigate how to model the relationships of complex objects in an image-captioning task, and how to effectively reduce the number of model parameters.

We thus propose using HAD and FEM modules as a means to manage complex relationships and reduce large numbers of model parameters. Our experimental results validate our approach, which outperform the best existing models on some metrics.

III. METHOD
We first introduce the FEM to solve the problem of many model parameters. Then, we add the HAD based on the transformer module and apply it to the encoder and decoder for image captioning.

A. FACTORIZED EMBEDDING MATRIX
In the field of natural language processing, tokens are usually expressed in the form of vectors that allow computers to process human language. The conventional approaches [1], [10] involve using the word-embedding matrix to project tokens onto fixed dimensions. This word embedding can be pretrained (e.g., using GloVe or Word2vec [31], [32]) or randomly initialized as part of the model parameters. In NLP, the size of the vocab $|V|$ is much greater than the number of dimensions of the word embedding $E$. This part can thus reach millions or even tens of millions of parameters. We proceed to solve this problem [24] by first projecting the token onto a lower dimension $H$ using a word-embedding matrix, and then restoring the original word-embedding $E$ dimension via projection mapping, as shown in Figure 1 (b). The advantage of doing this can not only reduce the parameters but also increase the relevance between words in the vocabulary. This operation can reduce the original complexity $\mathcal{O}(|V| \times E)$ to $\mathcal{O}(|V| \times H + H \times E)$. The above derivation can be expressed as

$$w_i = \text{Linear}(\text{Emb}(\Pi_i))$$

where $\Pi_i, \text{Emb}, w_i$ and $\text{Linear}$ denote, respectively, the one-hot encoding of the input word at time step $t$, the token-embedding matrix, the word token, and projection mapping.

B. HYBRID ATTENTION DISTRIBUTION
The basic transformer module is widely used in NLP tasks. At its core is the self-attention mechanism:

$$\text{Attention}(Q_i, K_i, V_i) = f_{\text{att}}(Q_i, K_i)V_i$$

where $f_{\text{att}}$ quantifies the similarity between two vectors, and $Q_i, K_i$ and $V_i$ denote some queries, keys, and values,
respectively. While this computation provides a good measure of the correlation between vectors, it is difficult to represent the similarity between vectors that correspond to different modalities, e.g., images and language. Therefore, after the high-dimensional image or language feature vector is divided into several low-dimensional subspaces, these subspaces are independent of each other. In order to connect them to each other to improve the expressive ability of the model, we propose using the HAD based on Eq. 2, as described in Figure 1 (a).

In the HAD, the function \( f_{att} \) superimposes multiple intrinsic inter-vector similarities to express complex intrinsic relationships. This computation requires only very few additional parameters to allow the model to capture complex multimodal information. For given \( Q_i, K_i, V_i \), \( f_{att} \) models as:

\[
f_{att}(Q_i, K_i) = \text{softmax}(\sum_{j=1}^{h} \Lambda_{ij}Z_j), \quad i = 1, 2, \ldots, h
\]

(3)

where \( h \) is the number of heads and \( \Lambda \in \mathbb{R}^{h \times h} \) are model parameters that can automatically measure the correlation between each head during training. \( Z = [Z_1, Z_2, \ldots, Z_h] \) quantifies the similarity between vectors in multiple heads as:

\[
Z_i = \frac{Q_iK_i}{\sqrt{d_k}}, \quad i = 1, 2, \ldots, h
\]

(4)

where \( d_k \) is the subspace dimension. Our experiments revealed that, when using only Eq.4 to calculate the similarity between vectors, the inherent variance fluctuates greatly, resulting in a very steep rise in the softmax function, which prevents model convergence. Thus, before calculating \( Z_i \), we first normalized \( Q_i \) and \( K_i \) (e.g., by instance normalization or layer normalization) to ensure numerical stability. For simplicity, this was omitted in Eq.4.

Then we specify the expression for each head \( \text{head}_i \in \mathbb{R}^{L \times d_k} \), where \( L \) is the length of the input tokens:

\[
\text{head}_i = \text{Attention}(Q_i, K_i, V_i), \quad i = 1, 2, \ldots, h
\]

(5)

Finally, the final expression for our HAD is obtained as:

\[
\text{HAD}(Q, K, V) = \text{concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)
\]

(6)

where the function concat is the operation of splicing the individual heads in the last dimension.

\section*{C. ENCODER WITH HAD}

We build our network model according to the classic encoder-decoder structure and embed HAD within it, as shown in Figure 2. Image captioning involves describing a given image \( I \) using a human language \( Y_{1:T} \). The sentence \( Y_{1:T} = \{w_1, w_2, \ldots, w_T\} \) consists of a sequence of tokens. The image \( I \) is represented as a set of regional image features \( V = \{v_1, v_2, \ldots, v_L\} \) by applying CNN or Faster R-CNN [33], where \( v_i \in \mathbb{R}^D \).

The output \( \hat{V}^{(i)} \) for i-th layer depends on the \( Q, K, V \) of \((i-1)\)-th layer:

\[
\hat{V}^{(i)} = \text{LayerNorm}(Q^{(i-1)} + \text{HAD}(W_Q^{(i-1)}Q^{(i-1)}, W_K^{(i-1)}K^{(i-1)}, W_V^{(i-1)}V^{(i-1)}))
\]

(7)

where \( W_Q^{(i-1)}, W_K^{(i-1)}, W_V^{(i-1)} \in \mathbb{R}^{D \times D} \) are model parameters about \( Q, K, V \) in \((i-1)\)-th layer. The purpose of stacking such layers in the encoder like [15] is to enable the model to express the deep inter-relations between regions for a given image. Notably, Eq.7 omits the GLU activation [34] module for simplicity.

\section*{D. DECODER WITH HAD AND FEM}

The role of the decoder in the image captioning is to generate matching tokens for given image-region features \( \hat{V}^{(N)} \), as depicted in Figure 3. The LSTM is first based on the input that the embedding of the word at current time step and a visual vector \((\bar{v} + c_{t-1})\), where \( \bar{v} = \frac{1}{L} \sum_{i=1}^{L} \hat{V}^{(N)}_i \) denotes global image features and \( c_{t-1} \) denotes the context information in the previous time step, to generate current hidden state \( h_t \):

\[
x_t = [w_t, \bar{v} + c_{t-1}]
\]

(8)

\[
h_t, m_t = \text{LSTM}(x_t, h_{t-1}, m_{t-1})
\]

(9)

where \( w_t \) is derived from Eq.1 and \( h_t, m_t \in \mathbb{R}^D \).
The gradients can be approximated using the cross entropy loss (XE) as follows:

\[ L_{XE} = - \sum_{t=1}^{T} \log(p_{\theta}(Y_t|Y_{1:t-1})) \]  

(13)

where \( Y_{1:t-1} \) denotes the target ground truth sequence.

Training with RL. After training with XE is complete, we use the parameters to initialize RL’s training strategy. Following Self-Critical Sequence Training (SCST):

\[ L_{RL}(\theta) = -E_{Y_{1:T} \sim p_{\theta}}[r(y_{1:T})] \]  

(14)

where the reward \( r(\cdot) \) uses the score of some metrics (e.g., CIDEr-D). The gradients can be approximated as:

\[ \nabla_\theta L_{RL}(\theta) \approx -(r(y_{1:T}^i) - r(\hat{y}_{1:T})) \nabla_\theta \log p_{\theta}(y_{1:T}^i) \]  

(15)

IV. EXPERIMENTS

A. DATASET AND IMPLEMENTATION DETAILS

All the experiments are conducted on the most popular image captioning benchmark MS COCO [37] which contains 123287 images labeled with at least 5 captions for each by 5 different people. Following many previous works [11], [14], [20], we use the standard Karpathy data split for the offline performance comparisons, in which 5000 images are used for validation, 5000 are used for testing and 113287 are used for training. MS COCO provides 40,775 images as test set for online evaluation as well. We convert all sentences to lower case and drop the words that occur less than 6 times, leading to the final vocabulary with 9488 unique words. For evaluation metrics, we use BLEU [38], METEOR [39], ROUGE-L [40], CIDEr [41] and SPICE [42] are simultaneously utilized to evaluate our model.

We employ a pre-trained Faster-RCNN [33] model on ImageNet and Visual Genome [43], [44] to extract bottom up feature vectors [11] of images. We take the output of RoI pooling layer and perform non-maximum suppression for each object class using an IoU threshold. We then select all regions where any class detection probability exceeds a confidence threshold, thus each image has a number of regions ranging from 10 to 100. The dimension of the original vectors is 2048 and we project them to a new space with the dimension of \( D = 1024 \), which is also the hidden size of the LSTM in the decoder. We set \( H=128 \) and \( E=1024 \) in our strong model. As for the training process, we train our model under XE loss for 30 epochs with a mini batch size of 10, and ADAM optimizer is used with a learning rate initialized by 2e-4 and annealed by 0.8 every 3 epochs. We increase the scheduled sampling [45] probability by 0.05 every 5 epochs. We optimize the CIDEr-D score with SCST for another 15 epochs with an initial learning rate of 2e-5 and annealed by 0.5 when the score on the validation split does not improve for some training steps.

B. QUANTITATIVE ANALYSIS

We report the performance of our model compare it with the current state-other-art in Tables 1 and 2. These models include NIC [1], which uses a vanilla CNN-LSTM encoder-decoder framework but lacking of deep feature extraction and fusion; SCST [20], which applies a standard attention-based model and is first to use SCST to directly optimize non-differentiable metrics but it is also insufficient for multi-modal information fusion; Up-Down [11] which uses an attention LSTM to attend over image features extracted from a Faster R-CNN model. However, it lacks further extraction of image regions features; RFNet [35], which fuses encoded features from multiple CNN networks; GCN-LSTM [36], which predicts visual relationships between every two entities in the image and encodes the relationship information into feature vectors; and finally AoANet [14] uses an information vector and an attention gate to obtain attended information based on transformer language model. However, in the internal calculation of the transformer, the relations between the sub-heads are not considered.

For the cross-entropy loss training stage in Table 1, we first decompose the word embedding matrix on the basis of [14], keeping the rest of the model unchanged and the indicators kept. Except for a slight decrease in CIDEr, the rest of the metrics remained the same or slightly improved and the number of model parameters was reduced. These results
TABLE 1. Performance of our single model and other state-of-the-art methods on MS-COCO “Karpathy” test split under cross-entropy training, where B{N}, M, R, C and S are short for BLEU{N}, METEOR, ROUGE-L, CIDEr-D and SPICE scores. All values are reported as percentage %. - indicates that the results are not provided.

| Model                  | B1   | B2   | B3   | B4   | M    | R    | C    | S    |
|------------------------|------|------|------|------|------|------|------|------|
| NIC [1]                | -    | -    | -    | -    | 29.6 | 25.2 | 52.6 | 94.0 |
| SCST [20]              | -    | -    | -    | -    | 30.0 | 25.9 | 53.4 | 99.4 |
| LSTM-A [6]             | -    | 75.4 | 56.5 | 42.9 | 35.2 | 26.9 | 55.8 | 108.8|
| Up-Down [11]           | -    | 77.2 | -    | -    | 36.2 | 27.0 | 56.4 | 113.5|
| RNet [35]              | -    | 76.4 | 60.4 | 46.6 | 35.8 | 27.4 | 56.8 | 112.5|
| GCN-LSTM [36]          | -    | 77.3 | -    | -    | 36.8 | 27.9 | 57.0 | 116.3|
| AoANet [5]             | 85.6M| 77.4 | -    | -    | 37.2 | 28.4 | 57.5 | 119.8|
| AoANet+FEM             | 77.3M| 77.5 | 61.8 | 48.1 | 37.2 | 28.4 | 57.5 | 118.4|
| HAD+FEM                | 77.2M| 78.0 | 62.4 | 48.6 | 37.5 | 28.2 | 57.5 | 118.9|

TABLE 2. Performance of our single model and other state-of-the-art models on MS-COCO Karpathy test split under CIDEr-D score optimization. All values are reported as percentage %. - indicates that the results are not provided.

| Model                  | #params | B1   | B2   | B3   | B4   | M    | R    | C    | S    |
|------------------------|---------|------|------|------|------|------|------|------|------|
| NIC [1]                | -       | -    | -    | -    | -    | 31.9 | 25.5 | 54.3 | 106.3|
| SCST [20]              | -       | -    | -    | -    | -    | 34.2 | 26.7 | 55.7 | 114.0|
| LSTM-A [6]             | -       | 78.6 | -    | -    | -    | 35.5 | 27.3 | 56.8 | 118.3|
| Up-Down [11]           | -       | 79.8 | -    | -    | -    | 36.3 | 27.7 | 56.9 | 110.1|
| RNet [35]              | -       | 79.1 | 63.1 | 48.4 | 36.5 | 27.7 | 57.3 | 121.9|
| GCN-LSTM [36]          | -       | 80.5 | -    | -    | -    | 38.2 | 28.5 | 58.3 | 127.6|
| AoANet [5]             | 85.6M   | 80.2 | -    | -    | -    | 38.9 | 29.2 | 58.8 | 129.8|
| HAD+FEM                | 77.2M   | 80.5 | 65.1 | 50.8 | 38.9 | 28.9 | 58.7 | 128.8|

TABLE 3. Leaderboard of various methods on the online MS-COCO test server where c5 and c40 represents that each image has 5 or 40 reference captions. All values are reported as percentage %.

| Model                  | Bleu1 | Bleu2 | Bleu3 | Bleu4 | METEOR | ROUGE-L | CIDEr-D |
|------------------------|-------|-------|-------|-------|--------|---------|---------|
| NIC [1]                | 71.3  | 89.5  | 54.2  | 80.2  | 40.7   | 69.4    | 30.9    | 58.7   | 25.4    | 34.6   | 53.0   | 68.2   | 94.3   | 94.6   |
| SCST [20]              | 78.1  | 93.7  | 61.9  | 86.0  | 47.0   | 75.9    | 35.2    | 64.5   | 27.0   | 55.5   | 56.3   | 70.7   | 114.7  | 116.0  |
| LSTM-A [6]             | 78.7  | 93.7  | 62.7  | 86.7  | 47.6   | 76.5    | 35.6    | 65.2   | 27.0   | 55.4   | 56.4   | 70.5   | 116.0  | 118.0  |
| Up-Down [11]           | 80.2  | 95.2  | 64.1  | 88.8  | 49.1   | 79.4    | 36.9    | 68.5   | 27.6   | 57.1   | 71.2   | 117.9  | 120.5  |
| RNet [35]              | 80.4  | 95.0  | 64.9  | 89.3  | 50.1   | 80.1    | 38.0    | 69.2   | 28.2   | 57.2   | 58.2   | 73.1   | 122.9  | 125.1  |
| GCN-LSTM [36]          | -     | 65.5  | 89.3  | 50.8  | 80.3   | 38.7    | 69.7    | 28.5   | 37.6   | 58.5   | 73.4   | 125.3  | 126.5  |
| Ours                   | 80.0  | 94.4  | 64.6  | 88.6  | 50.1   | 79.9    | 38.2    | 69.5   | 28.5   | 37.9   | 58.1   | 73.5   | 123.3  | 125.6  |

TABLE 3. Leaderboard of various methods on the online MS-COCO test server where c5 and c40 represents that each image has 5 or 40 reference captions. All values are reported as percentage %.

C. QUALITATIVE ANALYSIS

Figure 5 shows some results generated by our HAD model. It displays a strong baseline as well as the human-annotated ground truth. We derive the baseline model by re-implementing [14]. These examples suggest that the baseline is syntactically logical, but there are problems with the description of specific objects. Furthermore, (1) our
demonstrate the effectiveness of FEM at reducing the model size while maintaining stable performance.

Then we use our model(HAD+FEM) to validate the performance of HAD, our single model achieves higher score for all the metrics than AoANet+FEM, with the exception METEOR. Our model is higher than [14] on BLEU1~4 and SPICE indicators, but the METEOR and CIDEr scores was marginally lower than [14].

For the sequence-level optimization stage in Table 2, our model also achieves the highest scores for Bleu1~Bleu4 and SPICE. Because of the reduced size of our model (from 85.6M to 77.2M), the performance is slightly less for METEOR, ROUGE-L, and CIDEr, but nonetheless approaches the highest score.

We also evaluate our single model on the online COCO test server in Table 3. The results show that our single model performance is close to [35] and [36], slightly improved on ROUGE and METEOR indicators.

FIGURE 5. Examples of captions generated by our model and a baseline model as well as the corresponding ground truths. The red field indicates the correct description, while the blue field is the opposite.
FIGURE 6. The performance of Bleu1–Bleu4, CIDEr and SPICE metrics on the validation set during training under XE loss, we decoded the words at each time step using beam size 1.

TABLE 4. Changing the value of $H$, the performance of the model on the test set under XE loss.

| #params | B1  | B4  | M   | R   | C   | S   |
|---------|-----|-----|-----|-----|-----|-----|
| H=128  | 77.2 M | 78.0 | 37.5 | 28.2 | 57.5 | 118.9 | 21.5 |
| H=64   | 76.5 M | 77.5 | 36.5 | 28.1 | 57.0 | 117.3 | 21.4 |
| H=32   | 75.2 M | 77.0 | 36.3 | 27.9 | 56.8 | 116.2 | 21.2 |
| H=16   | 75.0 M | 76.8 | 36.4 | 28.0 | 56.8 | 116.2 | 21.1 |
| H=8    | 74.9 M | 77.1 | 36.2 | 27.9 | 56.9 | 116.0 | 21.3 |
| H=1    | 73.8 M | 77.2 | 36.5 | 28.0 | 57.0 | 116.3 | 21.2 |

model correctly describes the specific objects in the image. For example, in the first example, the baseline describes couch as table, and in the second example (from left to right), the baseline mistakes the hot dog for a sandwich. In contrast, our model yields accurate descriptions. (2) Our model correctly describes the number of specific objects in the image. In the third and fourth examples, our model accurately recognizes that there are three cupcakes and a black and white cat looking out the window in the picture, while the baseline model depicts them as a group of cupcakes and two black and white cats. Our model also focuses on a wooden bench in the picture of the fifth example, and not on two wooden benches. The above advantages of our model stem precisely from the HAD module, which is able to focus on multiple layers of specific complex objects in an image to understand their relationships and to accurately generate corresponding words in the decoding phase. Thus, our model performs better than the baseline in terms of accuracy and fluency.

In the sixth example, however, our model depicts a truck parked in the forest and does not represent the snow scene. The baseline model describes the snow but fails to describe the forest. This example highlights the remaining challenges of image captioning and scope for future research.

D. ABLATIVE ANALYSIS

Table 4 reports the effect of using different values of $H$ in the FEM module on the model performance. The best performance is achieved for $H = 128$, using the highest values for each metric. As $H$ decreases to 32, the indicators appear significantly reduced, but the values of Bleu1, METEOR and SPICE are greater than the situation at $H = 16$ and $H = 8$. We perform this analysis because, for smaller values of $H$, the number of model parameters changes little, resulting in little change in the model’s performance. In the extreme case $H = 1$, the model performance is better than $H = 8, 16, 32$ for some metrics.

Figure 6 also analyses the model convergence rate at the training stage for different values of $H$. The results are compared with the baseline model [14]. Figure 6 clearly shows that the model converges much better than the baseline in Bleu1–Bleu4, with the higher values of $H$ achieving faster convergence. For example, our model converges after 200k iterations while the baseline converges after approximately 300k iterations. This advantage is less pronounced for the CIDEr and SPICE metrics, but is nonetheless apparent, relative to the baseline criterion. We can thus conclude that our model converges faster than the baseline model. In Figure 4, we observe that the attention map in HAD is lighter than the baseline and focuses on different positions at each time step which indicates that our model is able to handle the intrinsic relationships of complex objects to generate the corresponding correct word, e.g., couch instead of table.

V. CONCLUSION

We have proposed a method that uses the hybrid attention distribution and the factorized embedding matrix to address problems above in our paper. The former solves the problem wherein a single distribution cannot effectively express the complex interconnections between multimodal states. The latter effectively reduces the model parameters and makes the model converge faster. Experiments performed on the MS COCO dataset demonstrate that our approach achieves state-of-the-art performance across some image captioning metrics.
JIAN WANG was born in Nantong, Jiangsu, China, in 1996. He received the bachelor’s degree from the Nanjing University of Information Science and Technology, Nanjing, China, in 2018. He is currently pursuing the master’s degree with Zhejiang Sci-Tech University, Hangzhou, China.

JIE FENG was born in Jinzhou, Liaoning, China, in 1980. He received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2003 and 2009, respectively. Since 2009, he has been with Zhejiang Sci-Tech University, Hangzhou. His major research interests include video analysis, 3D video conversion, and video processing.