Contrastive Grouping with Transformer for Referring Image Segmentation

Jiajin Tang¹ Ge Zheng¹ Cheng Shi¹ Sibei Yang¹,2†
¹School of Information Science and Technology, ShanghaiTech University
²Shanghai Engineering Research Center of Intelligent Vision and Imaging
{tangjj, zhengge, shicheng2022, yangsb}@shanghaitech.edu.cn

Abstract

Referring image segmentation aims to segment the target referent in an image conditioning on a natural language expression. Existing one-stage methods employ per-pixel classification frameworks, which attempt straightforwardly to align vision and language at the pixel level, thus failing to capture critical object-level information. In this paper, we propose a mask classification framework, Contrastive Grouping with Transformer network (CGFormer), which explicitly captures object-level information via token-based querying and grouping strategy. Specifically, CGFormer first introduces learnable query tokens to represent objects and then alternately queries linguistic features and groups visual features into the query tokens for object-aware cross-modal reasoning. In addition, CGFormer achieves cross-level interaction by jointly updating the query tokens and decoding masks in every two consecutive layers. Finally, CGFormer cooperates contrastive learning to the grouping strategy to identify the token and its mask corresponding to the referent. Experimental results demonstrate that CGFormer outperforms state-of-the-art methods in both segmentation and generalization settings consistently and significantly. Code is available at https://github.com/Toneyaya/CGFormer.

1. Introduction

Referring Image Segmentation (RIS) aims to segment the target referent in an image given a natural language expression [12, 17, 64]. It attracts increasing attention in the research community and is expected to show its potential in real applications, such as human-robot interaction via natural language [50] and image editing [3]. Compared to classical image segmentation that classifies pixels or masks into a closed set of fixed categories, RIS requires locating referent at pixel level according to the free-form natural languages with open-world vocabularies. It faces challenges of comprehensively understanding vision and language modalities and aligning them at pixel level.

Existing works mainly follow the segmentation framework of per-pixel classification [17, 34] integrated with multi-modal fusion to address the challenges. They introduce various fusion methods [19, 20, 26, 32, 46, 63] to obtain vision-language feature maps and predict the segmentation results based on the feature maps. Recently, the improvement of vision-language fusion for RIS mainly lies in utilizing the Transformer [12, 25, 51, 62]. LAVT [62] integrates fusion module into the Transformer-based visual encoder. CRIS [51] and ReSTR [25] fuse linguistic features into each feature of the visual feature maps, as shown in Figure 1a. In contrast, VLT [12] integrates the relevant visual features into language-conditional query vectors via transformer decoder, as shown in Figure 1b.

Although these methods have improved the segmenta-
tion accuracy, they still face several intrinsic limitations. First, the works [25, 51, 62] based on pixel-level fusion only model the pixel-level dependencies for each visual feature, which fails to capture the crucial object/region-level information. Therefore, they cannot accurately ground expressions that require efficient cross-modal reasoning on objects. Second, although VLT [12]’s query vectors contain object-level information after querying, it directly weights and reshapes different tokens into one multi-modal feature map for decoding the final segmentation mask. Therefore, it loses the image’s crucial spatial priors (relative spatial arrangement among pixels) in the reshaping process. More importantly, it does not model the inherent differences between query vectors, resulting in that even though different query vectors comprehend expressions in their own way, they still focus on similar regions, but fail to focus on different regions and model their relations.

In this paper, we aim to propose a simple and effective framework to address these limitations. Instead of using per-pixel classification framework, we adopt an end-to-end mask classification framework [1, 16] (see Figure 1c) to explicitly capture object-level information and decode segmentation masks for both the referent and other disturbing objects/stuffs. Therefore, we can simplify RIS task by finding the corresponding mask for the expression. Note that our framework differs from two-stage RIS methods [53, 64] which require explicitly detecting the objects first and then predicting the mask in the detected bounding boxes.

Specifically, we propose a Contrastive Grouping with Transformer (CGFormer) network consisting of the Group Transformer and Consecutive Decoder modules. The Group Transformer aims to capture object-level information and achieve object-aware cross-modal reasoning. The success of applying query tokens in object detection [1, 69] and instance segmentation [8, 16, 54] could be a potential solution. However, it is non-trivial to apply them to RIS. Without the annotation supervision of other mentioned objects other than the referent, it is hard to make tokens pay attention to different objects and distinguish the token corresponding to the referent from other tokens. Therefore, although we also specify query tokens as object-level information representations, we explicitly group the visual feature map’s visual features into query tokens to ensure that different tokens focus on different visual regions without overlaps. Besides, we can further cooperate contrastive learning with the grouping strategy to make the referent token attend to the referent-relevant information while forcing other tokens to focus on different objects and background regions, as shown in Figure 1c. In addition, we alternately query the linguistic features and group the visual features into the query tokens for cross-modal reasoning.

Furthermore, integrating and utilizing multi-level feature maps are crucial for accurate segmentation. Previous works [32, 40, 63] fuse visual and linguistic features at multiple levels in parallel and late integrate them via ConvLSTM [47] or FPNs [30]. However, their fusion modules are solely responsible for cross-modal alignment at each level, which fails to perform joint reasoning for multiple levels. Therefore, we propose a Consecutive Decoder that jointly updates query tokens and decodes masks in every two consecutive layers to achieve cross-level reasoning.

To evaluate the effectiveness of CGFormer, we conduct experiments on three standard benchmarks, i.e., RefCOCO series datasets [39, 41, 65]. In addition, unlike semantic segmentation, RIS is not limited by the close-set classification but to open-vocabulary alignment. It is necessary to evaluate the generalization ability of RIS models. Therefore, we introduce new subsets of training sets on the three datasets to ensure the categories of referents in the test set are not seen in the training stage, inspired by the zero-shot visual grounding [44] and open-set object detection [67].

In summary, our main contributions are as follows,

- We propose a Group Transformer cooperated with contrastive learning to achieve object-aware cross-modal reasoning by explicitly grouping visual features into different regions and modeling their dependencies conditioning on linguistic features.
- We propose a Consecutive Decoder to achieve cross-level reasoning and segmentation by jointly performing the cross-modal inference and mask decoding in every two consecutive layers in the decoder.
- We are the first to introduce an end-to-end mask classification framework, the Contrastive Grouping with Transformer (CGFormer), for referring image segmentation. Experimental results demonstrate that our CGFormer outperforms all state-of-the-art methods on all three benchmarks consistently.
- We introduce new splits on datasets for evaluating generalization for referring image segmentation models. CGFormer shows stronger generalizability compared to state-of-the-art methods thanks to object-aware cross-modal reasoning via contrastive learning.

2. Related Work

Referring Image Segmentation (RIS) aims to segment objects from images according to natural language expressions. The pioneering work [17] uses the concatenation operation to fuse the linguistic and visual features. Some following works [7, 17, 18, 22, 23, 26, 37] extract textual features for the expressions at the sentence level, while other works [2, 14, 32, 40] employ word vectors as textual representations. Considering that natural language naturally contains structured information [45, 59] that can be exploited to align with visual constituents, some methods explicitly decompose expressions into different components [20, 53,
The framework of our proposed CGFormer is shown in Figure 2. First, we adopt the visual encoder and language encoder to encode images and referring expressions (see Section 3.1). Second, we achieve object-aware cross-modal reasoning via the proposed Group Transformer (see Section 3.2). Next, we implement the cross-level reasoning and segmentation via the proposed Consecutive Decoder (see Section 3.3).
Section 3.3). Finally, we apply contrastive learning to distinguish the referent token from other tokens and obtain the mask corresponding to the referent token as the segmentation result (see Section 3.4).

3.1. Visual Encoder and Language Encoder

Visual Encoder. Following the previous work [62], we employ Swin Transformer [33] as the visual encoder for fair comparison. For an input image \( I \in \mathbb{R}^{H \times W \times 3} \) with the size of \( H \times W \), we extract its visual feature maps at four stages \( i \in \{1, 2, 3, 4\} \), which we denote it as \( V = \{V_i\}_{i=1}^{4}, V_i \in \mathbb{R}^{H_i \times W_i \times C_i} \). Here, each stage is corresponding to an encoding block of Swin Transformer, and \( H_i, W_i \) and \( C_i \) denote the height, width and channel dimension of \( V_i \).

Language Encoder. We adopt BERT [11] as the language encoder following the previous work [62]. Given an expression contains \( L \) words, we extract its linguistic feature and denote it as \( e \in \mathbb{R}^{C_L} \), where \( C_L \) is the channel dimension. In addition, we obtain the word representations by removing the last pooling layer, which are denoted as \( F \in \mathbb{R}^{L \times C_L} \).

3.2. Group Transformer

We propose the Group Transformer to achieve object-aware cross-modal inference. Group Transformer uses query tokens to represent object-level information and updates query tokens by alternately querying the linguistic features and grouping visual features. Specifically, we first initialize a set of learnable tokens representing the different objects/regions (see Section 3.2.1). Then we query the linguistic and visual features for tokens via our novel Group Transformer layer (see Section 3.2.2). After alternative reasoning, tokens captures the rich object characteristics relevant to the referring expressions. Note that we use multiple Group Transformer layers in different stages of the decoder, which will be introduced in Section 3.3.1.

3.2.1 Definition of Query Token

Inspired by the apply of query tokens [1, 9, 54] in object detection and semantic segmentation, we randomly initialize \( N \) learnable tokens \( T \in \mathbb{R}^{N \times C_L} \) where \( C_L \) is the channel dimension of tokens, to represent referent and other disturbing objects/stuffs.

Next, we feed these tokens \( T \) to Group Transformer layers to capture the object-level information conditioning on the expression and update the features for tokens. For simplicity of demonstration, we use \( T_{i-1} \in \mathbb{R}^{N \times C_L} \) to represent the features of tokens output from the \((i-1)\)-th layer and input them to the \(i\)-th layer of the Group Transformer.

3.2.2 Group Transformer Layer

The right part of Figure 2 illustrates the architecture of a single Group Transformer layer. The Load block and Group block are two core blocks to achieve object-aware cross-modal reasoning. In addition, we follow standard transformer [49] to employ LayerNorm for feature normalization and MLP with activation function for nonlinear mapping. Specifically, the Load block preloads the linguistic information each token should focus on at the current layer. The Group block, the critical component of Group Transformer, performs the cross-modal interaction and groups the visual features into query tokens to ensure that different tokens focus on different visual regions without overlap.

Load Block is expected to preload what linguistic information the query tokens should focus on at the current layer. The load layer is implemented by a classical cross-attention block [49], accepting the input tokens \( T_{i-1} \in \mathbb{R}^{N \times C_L} \) as the query and the word vectors \( F \in \mathbb{R}^{L \times C_L} \) extracted by the text encoder (see Section 3.1) as the key and value. Concretely, the computation of the Load block is as follows:

\[
T_i^q = T_{i-1}W_q, F^k = FW_k, F^v = FW_v,
\]

\[
T_i^l = (\text{softmax}(\frac{T_i^q(F_k)^T}{\sqrt{C_L}})F^v)W_c,
\]

where \( W_q, W_c \in \mathbb{R}^{C_L \times C'}, W_k, W_v \in \mathbb{R}^{C \times C'} \) are learnable projection matrices. And \( T_i^q, F^k \) and \( F^v \) are query, key and value in the cross attention, respectively. We end up with linguistic-enhanced representations for tokens, \( T_i^l \in \mathbb{R}^{N \times C'} \), and feed them into the Group block to query and group the relevant visual features for tokens.

Group Block interacts between vision and language and groups visual features from the feature map into linguistic-enhanced query tokens \( T_i' \). We denote the feature map as \( D_i \in \mathbb{R}^{H_i \times W_i \times C_i} \), which is fused from the feature maps in two consecutive layers in Consecutive Decoder (refer to Section 3.3 for details). Firstly, we project \( T_i' \) and \( D_i \) into a common feature space:

\[
T_i' = T_i'W_i, D_i' = \text{flatten}(D_i)W_d,
\]

where \( W_i \in \mathbb{R}^{C_i \times C'} \) and \( W_d \in \mathbb{R}^{C_i' \times C'} \) are learnable projection matrices, and flatten operation flattens the feature map \( D_i \) into the visual feature with \( H_i, W_i \) vectors. Then, we calculate the similarities \( S_{\text{pixel}} \in \mathbb{R}^{N \times H_i \times W_i} \) between every pairwise features of tokens \( T_i' \) and features \( D_i' \):

\[
S_{\text{pixel}} = \text{norm}(T_i')\text{norm}(D_i')^T,
\]

where \( \text{norm} \) means L2 normalization for vectors.

Next, based on the similarities \( S_{\text{pixel}} \), we group the features in \( D_i' \) and corresponds the groups to tokens \( T_i' \). However, the grouping operation with straightforward hard assignment is non-differentiable. Therefore, we adopt a learnable Gumbel-softmax [21, 38] to hard assign the features in \( D_i' \) to the tokens \( T_i' \) and generate the mask \( S_{\text{mask}} \).
\[ S_{\text{gumbel}} = \text{softmax}(\frac{(S_{\text{pixel}} + G)}{\tau}), \]
\[ S_{\text{onehot}} = \text{onehot}(\text{argmax}_N(S_{\text{gumbel}})), \]
\[ S_{\text{mask}} = (S_{\text{onehot}})^T - \text{sg}(S_{\text{gumbel}}) + S_{\text{gumbel}}, \]

where \( G \in \mathbb{R}^{N \times H_i \times W_i} \) samples from the Gumbel(0, 1) distribution, \( \tau \) is the learnable significance coefficient to assist in finding a more suitable assign boundary, \( \text{sg} \) is the stop gradient operator. Here, \( \text{argmax}_N \) means selecting the corresponding token of \( T'_i \) with the highest similarity for each feature in \( D'_i \), and the onehot operation transforms the token indexes into \( H_i \times W_i \) one-hot vectors \( S_{\text{onehot}} \in \mathbb{R}^{H_i \times W_i \times N} \). The mask \( S_{\text{mask}} \in \mathbb{R}^{N \times H_i \times W_i} \) indicates the grouping from the features \( D'_i \) to the tokens \( T'_i \).

Finally, we integrate the features \( D'_i \) to update tokens \( T'_i \) based on the mask \( S_{\text{mask}} \), which is computed as follows:

\[ T_i = \text{MLP}(S_{\text{mask}}D'_i) + T'_i, \]

where MLP is the multilayer perceptron. And \( T_i \) are the updated features of tokens via the Group block, which capture the rich object/region characteristics relevant to the linguistic features.

### 3.3. Consecutive Decoder

We further perform cross-level reasoning via the proposed Consecutive Decoder. Figure 3 shows the architecture of our Consecutive Decoder and its comparison to the parallel cross-modal fusion. Previous works [2, 18–20, 32, 37, 40, 63] model the vision-language interaction at multiple levels in parallel and late integrate multi-level results. The sole interaction at a single level fails to perform joint interaction across various levels. In contrast, the Consecutive Decoder achieves cross-level reasoning by jointly updating the query tokens in every two consecutive decoder layers, and the two-level cross-modal information will be consecutively propagated in multiple levels from bottom to up.

Specifically, the Consecutive Decoder contains three stages. At each decoding stage, it first fuses feature maps at two levels as the input of the Group Transformer layer to update tokens (see Section 3.3.1) and then decodes the corresponding mask of each token through the Mask Head (see Section 3.3.2).

#### 3.3.1 Consecutive Decoding

We intersperse multi-scale and cross-modal reasoning at each decoding layer. Specifically, for the \( i \)-th decoder layer, we first adopt a convolutional module to fuse the visual feature map \( V_i \) at the current layer \( i \) and the multi-modal feature map \( D_{i-1} \) output from the previous layer \( i - 1 \) of the Consecutive Decoder to generate the feature map \( D_i \in \mathbb{R}^{H_i \times W_i \times C'_i} \). Then, we update the query tokens \( T_{i-1} \) output from the previous Consecutive Decoder layer by querying them on the feature map \( D_i \) via the Group Transformer layer. The calculation is as follows:

\[ D_i = \text{Conv}([V_i; \text{Up}(D_{i-1})]), i \in \{2, 3, 4\} \]
\[ T_i = \text{GroupTransformerLayer}(T_{i-1}, D_i), \]

where \( \text{Conv} \) is the convolution layer, \( \text{Up} \) refers to upsampling \( D_{i-1} \) to the scale of \( V_i \), \( [\cdot] \) denotes concatenation along the channel dimension.

Particularly, for the first decoder layer \( i = 1 \), we skip the fusion and cross-modal interaction and let \( D_1 = V_1 \) and \( T_1 = T \), where \( T \) are the initialized tokens defined in Section 3.2.1.

#### 3.3.2 Mask Head

For \( i \)-th decoder layer, our mask head takes updated tokens \( T_i \) and visual feature map \( D_i \) as inputs and output the segmentation probabilities \( Z_i \in \mathbb{R}^{N \times H_i \times W_i} \) for tokens via dynamic convolutions [6]. For \( n \)-th token with feature \( T_i^{(n)} \), we first project it to convolution kernels \( W_i^{(n)} \) and then predict the segmentation probabilities \( Z_i^{(n)} \in \mathbb{R}^{H_i \times W_i} \) based on the kernels, which is computed as follows:

\[ W_i^{(n)} = \text{MLP}(T_i^{(n)}), \]
\[ Z_i^{(n)} = \text{Sigmoid}(\text{Conv}_{W_i^{(n)}}(D_i)), \]

where the superscript \( (n) \) denotes the features, kernels, and predicted probabilities corresponding to the \( n \)-th token, and the \( \text{Conv}_{W_i^{(n)}} \) means the convolution layer with the convolution kernels \( W_i^{(n)} \).

### 3.4. Contrastive Learning

We use contrastive learning to distinguish the referent token from other tokens by maximizing the similarity between the referent token and the expression and minimizing the similarities between negative pairs. For simplicity

Figure 3. Comparison of (a) parallel cross-modal fusion and (b) our cross-level reasoning via the Consecutive Decoder.
of demonstration, we suppose the first token represents the referent and other tokens represent non-target objects/stuff. The contrastive loss between the tokens with features $T_A \in \mathbb{R}^{N \times C'}$ output from last decoder layer and the expression $e \in \mathbb{R}^{C'}$ is computed as follows,

$$L_{cl} = - \log\left(\frac{\exp(s(T_A^{1}, e))}{\sum_{n=0}^{N} \exp(s(T_A^{n}, e))}\right),$$

where $s(\cdot, \cdot)$ is used to compute the similarity.

In addition, we combine the dice loss [27] and binary cross-entropy loss as the segmentation loss, $L_{seg}$. And we use the segmentation loss on multiple levels to supervise the learning of masks $Z_t$. The total loss $L$ is the sum of contrastive loss $L_{cl}$ and segmentation loss $L_{seg}$:

$$L = L_{cl} + L_{seg}.$$ (9)

During the inference, we predict the mask based on the referent token’s segmentation probabilities of the last decoder layer, i.e., $Z_A^{1}$.

### 4. Experiments

#### 4.1. Datasets and Implementation Details

**Datasets.** We conduct experiments on four common benchmark datasets, RefCOCO [65], RefCOCO+ [65], G-Ref [39, 41], and ReferIt [24]. The images of the first three datasets are all based on MSCOCO [31], but are annotated with different settings. RefCOCO has a short average description length of 3.5 words, RefCOCO+ is limited to not describing absolute locations of referents, and G-Ref has a longer word count per expression (8.4 words). We follow previous works [32, 61] to split the RefCOCO and RefCOCO+ into training, validation, testA and testB. For the G-Ref, we apply both partitions of UMD and Google for the evaluation. In addition, the ReferIt dataset, which is also the main benchmark for referring image segmentation, includes 19,894 images sourced from the IAPR TC-12 [13].

**Implementation Details.** Following [62], our visual encoder is pre-trained on ImageNet22K [10], text encoder is initialized with the weights from HuggingFace [52], and image size is 480 × 480. The hyperparameters, $C^v$, $C'$ and $C''$ are 1024/2^{n-1}, 768 and 512, respectively. We adopt AdamW [35] as the optimizer with initialized learning rate 1e-4 and train the model for 50 epochs with batch size 64. All experiments are conducted on NVIDIA Tesla A40 GPUs. Following [62], we adopt overall IoU (oloU), mean IoU (mIoU), and precision at the 0.5, 0.7, and 0.9 thresholds of IoU as our main evaluation metrics.

**Implementation for Generalization.** We introduce new splits on RefCOCO series datasets to validate the generalization, inspired by [44]. Specifically, we split them according to the splits of seen and unseen classes on MSCOCO of open-vocabulary detection [67]. Image-text pairs in the original training sets whose referent categories belong to the seen classes are selected as the new training sets. Likewise, the image-text pairs of test sets are also split into seen and unseen subsets according to whether the categories of the referents belong to seen classes. And the categories of referents in the unseen splits are not seen in the training stage. We consider that CRIS [51] employs CLIP [43] as the encoder network for transferring the knowledge of CLIP to achieve text-to-pixel alignment. Therefore, for the generalization experiments, we also take the text encoder of CLIP as the language encoder for our CGFormer and LVAT [62] for a fair comparison. For all methods, we train 50 epochs using the official code and select best-performing models.
Comparison on Generalization. We compare CGFormer with LAVT [62] and CRIS [51] as LAVT is the current best-performing model in referring image segmentation and CRIS transfers the knowledge of the strong generalizable CLIP model [43]. As is shown in Table 2, CGFormer outperforms LAVT and CRIS for both seen and unseen splits on all three datasets consistently. On the RefCOCO+ dataset that relies on understanding object-level attributes rather than location information, our performance exceeds LAVT by 5.8% and 9.6% in terms of average mIoU on seen and unseen splits, respectively. In addition, our CGFormer performs significantly better than other methods on the G-ref dataset with more complex languages. The mIoU of our CGFormer outperforms CRIS by 20.49% and 12.21% on seen and unseen splits of G-Ref(G), respectively.

### 4.3. Ablation Study

The results of the ablation study are shown in Table 3. Baseline and Grouping Strategy. (1) The baseline extracts the visual feature map $V_4$ of the visual encoder and predicts the segmentation result from the map via dynamic convolutions with the learned kernels from linguistic feature $e$. (2) We improve the baseline by generating a linguistic-conditioned query token and querying relevant visual features on $V_4$ to capture object-level information, which slightly improves the baseline by 0.69%. (3) We further extend one token to multiple tokens, but the two models have similar performance. The results suggest that simply adding tokens cannot boost performance, as these tokens
are likely to focus on similar information rather than distinct regions. (4) Our grouping strategy cooperated with contrastive loss to make tokens can focus on different regions and let them distinguishable, which delivers a 4.35% improvement. (5) We further use the hard assignment with learnable Gumbel Softmax to obtain a more refined grouping that achieves an improvement of 1.63%.

**Multi-Scale Decoding.** We extend the single-scale model (row 5) to multi-scale one (row 6) by first parallel updating tokens at multiple levels and then integrating these tokens to predict the segmentation mask over the multi-scale visual feature map fused by FPNs [30]. The 0.84% improvement of oIoU shows the effectiveness of multi-scale features. We further connect grouping layers by consecutive decoding, which is applied in our final model (row 7). The 1.47% improved oIoU of the model using a consecutive decoder (row 7) over the model using parallel querying (row 6) suggests that our joint querying and decoding in the decoder is a more desirable solution than aggregating different levels of information in parallel.

**Others.** (1) We further validate the necessity of the proposed contrastive grouping by comparing our CGFormer (row 7) with VLT [12] (row 8) using the same visual backbone and text encoder. We significantly outperforms VLT by 3.99%, 5.88%, 14.13% and 3.86% in terms of P@0.5, P@0.7, P@0.9 and oIoU, respectively. (2) We replace the cosine similarity with dot produce (row 9) or fix the parameter $\tau$ in the Gumble softmax to 0.1 (row 10), which results in a reduction of about 1.3% in oIoU.

### 4.4. Visualization

Figure 4 visualizes segmentation results. The expression in (a) refers to a girl in a complex scenario with a crowd of several dozen people. For (b), CGFormer accurately recognizes the challenging visual concept “sliced banana” and distinguishes it from a similar object with the same visual concept. The (c) demonstrates that our joint grouping and decoding comprehensively understand the expression and image rather than only focusing on local information “hands” and “man”. The (d) illustrates that CGFormer entirely segments the “bear” even though it is shaded by “pole” thanks to our object-aware reasoning.

**Variant Results** of our models without the Contrastive Grouping and Consecutive Decoder are shown in the two rightmost columns of Figure 4, respectively. Contrastive Grouping captures object-level information to distinguish between similar objects, and the Consecutive Decoder help obtain more precise segmentation results.

**Qualitative results of grouping** are shown in Figure 5 (a) and (b), which demonstrates: (1) Both the referent token and others represent certain meaningful objects/regions. (2) Tokens can partition the objects of the same categories (e.g., the different trucks) and different categories (e.g., the cake and candles). (3) The grouping can be more precise in multiple stages (see results for referent token in b).

### 5. Conclusion

This paper proposes a novel Contrastive Grouping with Transformer network (CGFormer) for referring image segmentation, which achieves object-aware cross-modal and cross-level reasoning. The experimental results demonstrate the superiority and the generalization ability of the proposed CGFormer compared to state-of-the-art methods. **Acknowledgment:** this work was supported by the National Natural Science Foundation of China (No.62206174), the Shanghai Pujiang Program (No.21PJ1410900), and the Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI).
References


[25] Namyup Kim, Dongwon Kim, Cuiling Lan, Wenzun Zeng, and Suha Kwak. Restr: Convolution-free referring im-
age segmentation using transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18145–18154, 2022. 1, 2, 3, 6, 7


