PolyFormer: Referring Image Segmentation as Sequential Polygon Generation

Jiang Liu[1,†,*]   Hui Ding[2,†]   Zhaowei Cai[2]   Yuting Zhang[2]

Ravi Kumar Satzoda[2]   Vijay Mahadevan[2]   R. Manmatha[2]

Johns Hopkins University[1]   AWS AI Labs[2]

https://polyformer.github.io/

Abstract

In this work, instead of directly predicting the pixel-level segmentation masks, the problem of referring image segmentation is formulated as sequential polygon generation, and the predicted polygons can be later converted into segmentation masks. This is enabled by a new sequence-to-sequence framework, Polygon Transformer (PolyFormer), which takes a sequence of image patches and text query tokens as input, and outputs a sequence of polygon vertices autoregressively. For more accurate geometric localization, we propose a regression-based decoder, which predicts the precise floating-point coordinates directly, without any coordinate quantization error. In the experiments, PolyFormer outperforms the prior art by a clear margin, e.g., 5.40% and 4.52% absolute improvements on the challenging RefCOCO+ and RefCOCOg datasets. It also shows strong generalization ability when evaluated on the referring video segmentation task without fine-tuning, e.g., achieving competitive 61.5% $\mathcal{J}\&\mathcal{F}$ on the Ref-DAVIS17 dataset.

1. Introduction

Referring image segmentation (RIS) [7, 19, 21, 29–32, 39, 45, 50, 61, 73, 78, 86, 87] combines vision-language understanding [42, 55, 57, 75, 92] and instance segmentation [2, 8, 16, 25, 52], and aims to localize the segmentation mask of an object given a natural language query. It generalizes traditional object segmentation from a fixed number of predefined categories to any concept described by free-form language, which requires a deeper understanding of the image and language semantics. The conventional pipeline [7, 19, 21, 29–32, 45, 50, 61, 73, 87] first extracts features from the image and text inputs, and then fuses the multi-modal features together to predict the mask.

A segmentation mask encodes the spatial layout of an object, and most instance segmentation models [8, 16, 25, 52] rely on a dense binary classification network to deter-

mine whether each pixel belongs to the object. This pixel-to-pixel prediction is preferred by a convolutional operation, but it neglects the structure among the output predictions. For example, each pixel is predicted independently of other pixels. In contrast, a segmentation mask can also be represented by a sparse set of structured polygon vertices delineating the contour of the object [1, 6, 41, 47, 49, 81]. This structured sparse representation is cheaper than a dense mask representation and is the preferred annotation format for most instance segmentation datasets [15, 49, 70]. Thus, it is also tempting to predict structured polygons directly. However, how to effectively predict this type of structured outputs is challenging, especially for convolutional neural networks (CNNs), and previous efforts have not shown much success yet [41, 47, 81].

We address this challenge by resorting to a sequence-to-sequence (seq2seq) framework [3, 14, 65, 66, 74], and propose **Poly**gon trans**Former** (PolyFormer) for referring image segmentation. As illustrated in Fig. 1, it takes a sequence of image patches and text query tokens as input, and autoregressively outputs a sequence of polygon vertices. Since each vertex prediction is conditioned on all preceding predicted vertices, the output predictions are no longer independent of each other. The seq2seq framework is flexible on its input and output format, as long as both of them can be formulated as sequences of variable length. Thus, it is natural to concatenate the visual and language features together as a long sequence, avoiding complicated multi-modal feature fusion as in prior work [7,30,73,86,87]. In the meantime, the output can also be a long sequence of multiple polygons separated by separator tokens, covering the scenario where the segmentation masks are not connected, *e.g.*, by occlusion, as shown in Fig. 1. Furthermore, since a bounding box can be represented as a sequence of two corner points (*i.e.*, top left and bottom right), they can also be output by PolyFormer along with the polygon vertices. Thus, referring image segmentation (polygon) and referring expression comprehension (bounding box) can be unified in our simple PolyFormer framework.

Localization is important for polygon and bounding box

---

*Work done during internship at AWS AI. † Equal contribution.
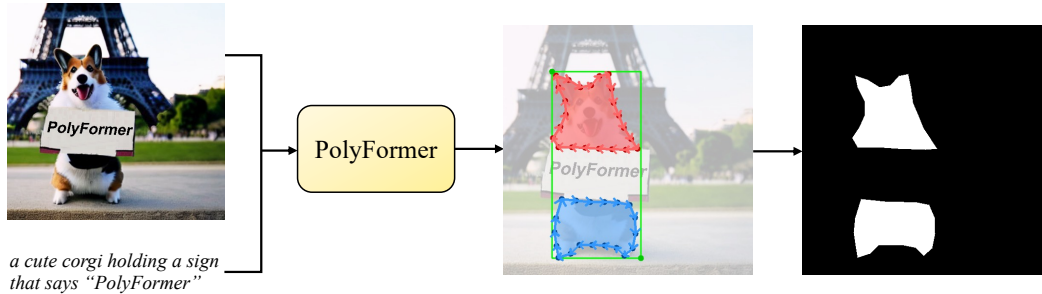
Figure 1. The illustration of PolyFormer pipeline for referring image segmentation (polygon vertex sequence) and referring expression comprehension (bounding box corner points). The polygons are converted to segmentation masks in the end.

generation, as a single coordinate prediction mistake may result in substantial errors in the mask or bounding box prediction. However, in recent seq2seq models for the visual domain [9, 10, 56, 77], in order to accommodate all tasks in a unified seq2seq framework, the coordinates are quantized into discrete bins and the prediction is formulated as a classification task. This is not ideal since geometric coordinates lie in a continuous space instead of a discrete one, and classification is thus usually suboptimal for localization task [23, 43, 93]. Instead, we formulate localization as a regression task, due to its success in object detection [5, 24, 25, 68], where floating-point coordinates are directly predicted without any quantization error. Motivated by [25], the feature embedding for any floating-point coordinate in PolyFormer is obtained by bilinear interpolation [33] of its neighboring indexed embeddings. This is in contrast with the common practice [9, 56, 77] in which the coordinate feature is indexed from a dictionary with a fixed number of discrete coordinate bins. These changes enable our Poly-Former to make accurate polygon and bounding box predictions.

We evaluate PolyFormer on three major referring image segmentation benchmarks. It achieves 76.94%, 72.15%, and 71.15% mIoU on the validation sets of RefCOCO [89], RefCOCO+ [89] and RefCOCOg [60], outperforming the state of the art by absolute margins of **2.48**%, **5.40**%, and **4.52**%, respectively. PolyFormer also shows strong generalization ability when directly applied to the referring video segmentation task without finetuning. It achieves 61.5% $\mathcal{J}\&\mathcal{F}$ on the Ref-DAVIS17 dataset [38], comparable with [80] which is specifically designed for that task.

Our main contributions are summarized as follows:

- We introduce a novel framework for RIS and REC, called PolyFormer, which formulates them as a sequence-to-sequence prediction problem. Due to its flexibility, it can naturally fuse multi-modal features together as input and generate a sequence of polygon vertices and bounding box corner points.

- We propose a regression-based decoder for accurate coordinate prediction in this seq2seq framework,

which outputs continuous 2D coordinates directly without quantization error. To the best of our knowledge, this is the first work formulating geometric localization as a regression task in seq2seq framework instead of classification as in [9, 10, 56, 77].

- For the first time, we show that the polygon-based method surpasses mask-based ones across all three main referring image segmentation benchmarks, and it can also generalize well to unseen scenarios, including video and synthetic data.

2. Related Work

Referring Image Segmentation (RIS) [29] aims to provide pixel-level localization of a target object in an image described by a referring expression. The previous works mainly focus on two aspects: (1) vision and language feature extraction, and (2) multi-modal feature fusion. For feature extraction, there has been a rich line of work, including the use of CNNs [7, 12, 29–32, 50, 87, 88], recurrent neural networks [12, 13, 21, 59, 71, 88], and transformer models [39, 44, 86]. The efforts on feature fusion have explored feature concatenation [29, 50], attention mechanisms [7, 30, 73, 87], and multi-modal transformers [19, 39, 44, 78]. The method most related to ours is SeqTR [94], which also adopts a transformer model for generating the polygon vertices sequentially. However, SeqTR can only produce a single polygon of 18 vertices with coarse segmentation masks, failing to outline objects with complex shapes and occlusion.

Referring Expression Comprehension (REC) predicts a bounding box that tightly encompasses the target object in an image corresponding to a referring expression. Existing works include two-staged methods [27, 28, 91, 95] that are based on region proposal ranking, and one-stage methods [4, 35, 44, 48, 85, 94] that directly predict the target bounding box. Several papers [44, 59, 94] explore multi-task learning of REC and RIS since they are two closely related tasks. However, MCN [59] and RefTR [44] require task-specific heads. Although SeqTR [94] casts the two
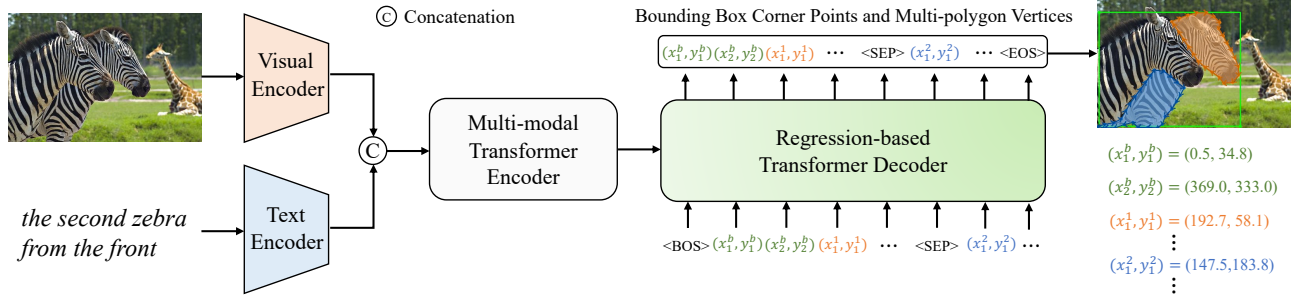
Figure 2. Overview of our PolyFormer architecture. The model takes an image and its corresponding language expression as input, and outputs the floating-point 2D coordinates of the bounding box and polygons in an autoregressive way.

tasks as a point prediction problem in a unified framework, it shows that multi-task supervision degenerates the performance compared with the single-task variant. In contrast, our PolyFormer achieves improved performance via multi-task learning of RIS and REC.

Sequence-to-Sequence (seq2seq) Modeling has achieved a lot of successes in natural language processing (NLP) [3, 14, 65, 66, 74]. Sutskever *et al.* [74] propose a pioneering seq2seq model based on LSTM [71] for machine translation. Raffel *et al.* [66] develop the T5 model to unify various tasks including translation, question answering and classification in a text-to-text framework. [3] further shows that scaling up language models significantly improves few-shot performance. Inspired by these successes in NLP, recent endeavors in computer vision and vision-language also start to explore seq2seq modeling for various tasks [9, 10, 56, 77, 84, 94]. However, they cast geometric localization tasks as a classification problem, *i.e.*, quantizing coordinates into discrete bins and predicting the coordinates as one of the bins. This enables them to unify all tasks into a simple unified seq2seq framework, but neglects the differences among tasks. In PolyFormer, geometric localization is formulated as a more suitable regression task that predicts continuous coordinates without quantization.

Contour-based Instance Segmentation aims to segment instances by predicting the contour. [6] labels object instances with polygons via a recurrent neural network. Polar-Mask [81] models instance masks in polar coordinates, and converts instance segmentation to instance center classification and dense distance regression tasks. Deep Snake [63] extends the classic snake algorithm [82] and uses a neural network to iteratively deform an initial contour to match the object boundary. PolyTransform [47] exploits a segmentation network to first generate instance masks to initialize polygons, which are then fed into a deformation network to better fit the object boundary. BoundaryFormer [41] introduces a point-based transformer with mask supervision via a differentiable rasterizer. However, these papers are limited in how they handle fragmented objects.

3. PolyFormer

3.1. Architecture Overview

Fig. 2 gives an overview of PolyFormer architecture. Instead of predicting dense segmentation masks, PolyFormer sequentially produces the corner points of the bounding box and vertices of polygons outlining the object. Specifically, we first use a visual encoder and a text encoder to extract image and text features, respectively, which are then projected into a shared embedding space. Next, we concatenate the image and text features, and feed them into a multi-modal transformer encoder. Finally, a regression-based transformer decoder takes the encoded features and outputs the continuous floating-point bounding box corner points and polygon vertices in an autoregressive way. The segmentation mask is generated as the region encompassed by the polygons.

3.2. Target Sequence Construction

We first describe how to represent ground-truth bounding box and polygon sequences.

Polygon Representation. A segmentation mask is described using one or more polygons outlining the referred object. We parameterize a polygon as a sequence of 2D vertices $\{(x_i, y_i)\}_{i=1}^{K}$, $(x_i, y_i) \in \mathbb{R}^2$ in the clock-wise order. We choose the vertex that is closest to the top left corner of the image as the starting point of the sequence (see Fig. 3).

Vertex and Special Tokens. For each vertex coordinate x or y, previous works [9, 10, 56, 77, 84, 94] uniformly quantize it into an integer between $[1, B]$, where $B \in \mathbb{N}$ is the number of bins of the coordinate codebook. In contrast, we maintain the continuous floating-point value of the original x or y coordinate without any quantization. To represent multiple polygons, we introduce a separator token <SEP> between two polygons. Polygons from the same object are ordered based on the distance between their starting points and the image origin. Finally, we use <BOS> and <EOS> tokens to indicate the beginning and end of the sequence.
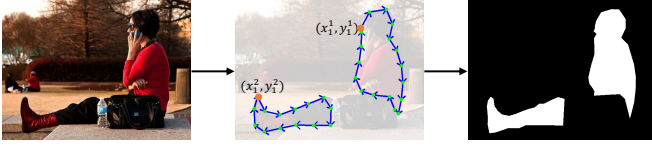
Figure 3. Illustration of polygon sequence representation. The vertices in a polygon are sorted in clockwise order, where the starting points (orange dots) are the vertices that are closest to the image origin. The segmentation mask is generated as the region encompassed by the polygons.

Unified Sequence with Bounding Box. A bounding box is represented by two corner points, *i.e.*, top-left (x_1^b, y_1^b) and bottom-right (x_2^b, y_2^b). The coordinates of the bounding box and multiple polygons can be concatenated together into a single long sequence as follows:

$$[\texttt{<BOS>}, (x_1^b, y_1^b), (x_2^b, y_2^b), (x_1^1, y_1^1),$$
$$(x_2^1, y_2^1), ..., \texttt{<SEP>}, (x_1^n, y_1^n), ..., \texttt{<EOS>}],$$

where (x_1^n, y_1^n) is the starting vertex of the nth polygon. In general, the bounding box corner points and polygon vertices are regarded as the coordinate tokens $\texttt{<COO>}$.

3.3. Image and Text Feature Extraction

As illustrated in Fig. 2, the input of our framework consists of an image I and a referring expression T.

Image Encoder. For an input image $I \in \mathbb{R}^{H \times W \times 3}$, we use a Swin transformer [53] to extract the feature from the 4-th stage as visual representation $F_v \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_v}$.

Text Encoder. Given the language description $T \in \mathbb{R}^L$ with L words, we use the language embedding model from BERT [18] to extract the word feature $F_l \in \mathbb{R}^{L \times C_l}$.

Multi-modal Transformer Encoder. To fuse the image and textual features, we flatten F_v into a sequence of visual features $F_v' \in \mathbb{R}^{(\frac{H}{32} \cdot \frac{W}{32}) \times C_v}$ and project F_v' and F_l into the same embedding space with a fully-connected layer:

$$F_v' = F_v' W_v + b_v, \ F_l' = F_l W_l + b_l, \quad (1)$$

where W_v and W_l are learnable matrices to transform the visual and textual representations into the same feature dimension, b_v and b_l are the bias vectors. The projected image and text features are then concatenated: $F_M = [F_v', F_l']$.

The multi-modal encoder is composed of N transformer layers, where each layer consists of a multi-head self-attention layer, a layer normalization and a feed-forward network. It takes the concatenated feature F_M and generates the multi-modal feature F_M^N progressively.

To preserve position information, absolute positional encodings [37] are added to the image and text features. In addition, we add 1D [66] and 2D [17,79] relative position bias to image and text features, respectively.
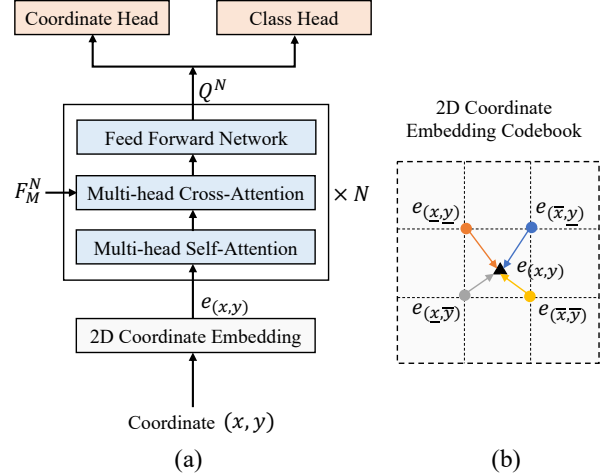

(a)   (b)

Figure 4. The architecture of the regression-based transformer decoder (a). The 2D coordinate embedding is obtained by bilinear interpolation from the nearby grid points, as illustrated in (b).

3.4. Regression-based Transformer Decoder

Previous visual seq2seq methods [9, 10, 56, 84, 94] quantize a continuous coordinate x into a discrete bin $[x]$, introducing an inevitable quantization error $|x - [x]|$. They formulate coordinate localization as a classification problem to predict one of $[x]$s, which is suboptimal for geometric localization. To address this issue, we propose a regression-based decoder that does not use quantization, and instead predicts the *continuous* coordinate values directly (*i.e.*, we use x instead of $[x]$), as shown in Fig. 4.

2D Coordinate Embedding. In [9, 10, 56, 84, 94], the coordinate codebook is in 1D space, $\mathcal{D} \in \mathbb{R}^{B \times C_e}$, where B is the number of bins, C_e is the embedding dimension. The embedding of x is obtained by indexing the codebook, *i.e.*, $\mathcal{D}([x])$. To better capture the geometric relationship between x and y and have a more accurate coordinate representation, we build a 2D coordinate codebook, $\mathcal{D} \in \mathbb{R}^{B_H \times B_W \times C_e}$, where B_H and B_W are the numbers of bins along height and width dimensions. With this 2D coordinate codebook, we can obtain the precise coordinate embedding for any floating-point coordinate $(x, y) \in \mathbb{R}^2$. First, the floor and ceiling operations are applied on (x, y) to generate four discrete bins: $(\underline{x}, \underline{y}), (\bar{x}, \underline{y}), (\underline{x}, \bar{y}), (\bar{x}, \bar{y}) \in \mathbb{N}^2$, and the corresponding embeddings can be indexed from the 2D codebook, *e.g.*, $e_{(\underline{x},\underline{y})} = \mathcal{D}(\underline{x}, \underline{y})$. Finally, we get the accurate coordinate embedding $e_{(x,y)}$ by bilinear interpolation, as in [25]:

$$e_{(x,y)} = (\bar{x} - x)(\bar{y} - y) \cdot e_{(\underline{x},\underline{y})} + (x - \underline{x})(\bar{y} - y) \cdot e_{(\bar{x},\underline{y})} +$$
$$(\bar{x} - x)(y - \underline{y}) \cdot e_{(\underline{x},\bar{y})} + (x - \underline{x})(y - \underline{y}) \cdot e_{(\bar{x},\bar{y})}. \quad (2)$$

Transformer Decoder Layers. To capture the relations between multi-modal feature F_M^N and 2D coordinate embedding $e_{(x,y)}$, we introduce N transformer decoder lay-

ers. Each transformer layer consists of a multi-head self-attention layer, a multi-head cross-attention layer and a feed-forward network.

Prediction Heads. Two lightweight heads are built on top of the last decoder layer output Q^N to generate final predictions. The class head is a linear layer that outputs the token types, indicating whether the current output is a coordinate token (<COO>), separator token (<SEP>) or an end-of-sequence token (<EOS>):

$$\hat{p} = W_c Q^N + b_c, \tag{3}$$

where W_c and b_c are parameters of the linear layer.

The coordinate head is a 3-layer feed-forward network (FFN) with ReLU activation except for the last layer. It predicts the 2D coordinates of the referred object bounding box corner points and polygon vertices:

$$(\hat{x}, \hat{y}) = Sigmoid(FFN(Q^N)). \tag{4}$$

3.5. Training

Polygon Augmentation. A polygon is a sparse representation of the dense object contour. Given a dense contour, the generation of sparse polygons is usually not unique. Given this property, we introduce a simple yet effective augmentation technique to increase polygon diversity. As illustrated in Fig. 5, the dense contour is first interpolated from the original polygon. Then, uniform down-sampling is applied with an interval randomly sampled from a fixed range to generate sparse polygons. This creates diverse polygons at different levels of granularity, and prevents the model from being overfitted to a fixed polygon representation.

Objective. Given an image, a referring expression and preceding tokens, the model is trained to predict the next token and its type:

$$L_t = \lambda_t L_{coo}((x_t, y_t), (\hat{x}_t, \hat{y}_t)|I, T, (x_i, y_i)_{i=1:t-1})$$
$$\cdot \mathbb{I}[p_t == \text{<COO>}] + \lambda_{cls}L_{cls}(p_t, \hat{p}_t|I, T, p_{1:t-1}), \tag{5}$$

where

$$\lambda_t = \begin{cases} \lambda_{box}, & t \leq 2, \\ \lambda_{poly}, & otherwise, \end{cases}$$

L_{coo} is the L_1 regression loss, L_{cls} is the label smoothed cross-entropy loss, and $\mathbb{I}[\cdot]$ is the indicator function. The regression loss is only computed for the coordinate tokens, where λ_{box} and λ_{poly} are the corresponding token weights. The total loss is the sum of L_t over all tokens in a sequence.

Inference. During inference, we start the generation by inputting the <BOS> token. First, we get the token type from the class head. If it is a coordinate token, we will obtain the 2D coordinate prediction from the coordinate head conditioned on the preceding predictions; if it is a separator token, it indicates the end of the preceding polygon, so the separator token will be added to the output sequence. This
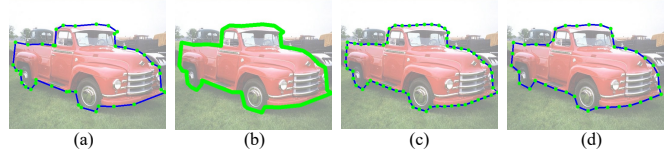

Figure 5. Illustration of polygon augmentation. Polygons at different levels of granularity (c) - (d) are sampled from dense contour (b) that is interpolated from the original polygon (a).

sequential prediction will stop once <EOS> is output. In the generated sequence, the first two tokens are bounding box coordinates and the rest are polygon vertices. The final segmentation mask is obtained from the polygon predictions.

4. Experimental Results

4.1. Datasets and Metrics

Datasets. The experiments are conducted on four major benchmarks for RIS and REC: RefCOCO [89], RefCOCO+ [89], RefCOCOg [60, 62], and ReferIt [36]. RefCOCO has 142,209 annotated expressions for 50,000 objects in 19,994 images, and RefCOCO+ consists of 141,564 expressions for 49,856 objects in 19,992 images. Compared with RefCOCO, location words are absent from the referring expressions in RefCOCO+, which makes it more challenging. RefCOCOg consists of 85,474 referring expressions for 54,822 objects in 26,711 images. The referring expressions are collected on Amazon Mechanical Turk, and therefore the descriptions are longer and more complex (8.4 words on average *vs.* 3.5 words of RefCOCO and RefCOCO+). ReferIt contains 130,364 expressions for 99,296 objects in 19,997 images collected from the SAIAPR-12 dataset [20]. We use the UMD split for RefCOCOg [62] and Berkeley split for ReferIt.

Evaluation Metrics. We use mean Intersection-over-Union (mIoU) as the evaluation metric for RIS. For a fair comparison, we also use overall Intersection-over-Union (oIoU) when comparing with papers that only report oIoU results. Additionally, we also evaluate PolyFormer on the REC task as it is a unified framework for both RIS and REC tasks. We adopt the standard metric Precision@0.5 [77,94], where a prediction is considered correct if its Intersection-over-Union (IoU) with the ground-truth box is higher than 0.5.

4.2. Implementation Details

Model Settings. In PolyFormer-B, we use Swin-B [53] as the visual encoder and BERT-base [18] as the text encoder. For the transformer encoder and decoder, we adopt 6 encoder layers and 6 decoder layers. To investigate the impact of different model scales, we develop a larger model, PolyFormer-L, that adopts Swin-L as the visual backbone with 12 transformer encoder and decoder layers.

Table 1:

	Method	Visual Backbone	Text Encoder	RefCOCO			RefCOCO+			RefCOCOg		ReferIt
				val	test A	test B	val	test A	test B	val	test	test
oIoU	STEP [7]	RN101	Bi-LSTM	60.04	63.46	57.97	48.19	52.33	40.41	-	-	64.13
	BRINet [30]	RN101	LSTM	60.98	62.99	59.21	48.17	52.32	42.11	-	-	63.11
	CMPC [31]	RN101	LSTM	61.36	64.53	59.64	49.56	53.44	43.23	-	-	65.53
	LSCM [32]	RN101	LSTM	61.47	64.99	59.55	49.34	53.12	43.50	-	-	66.57
	CMPC+ [51]	RN101	LSTM	62.47	65.08	60.82	50.25	54.04	43.47	-	-	65.58
	MCN [59]	DN53	Bi-GRU	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
	EFN [21]	WRN101	Bi-GRU	62.76	65.69	59.67	51.50	55.24	43.01	-	-	66.70
	BUSNet [83]	RN101	Self-Att	63.27	66.41	61.39	51.76	56.87	44.13	-	-	-
	CGAN [58]	DN53	Bi-GRU	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	-
	LTS [34]	DN53	Bi-GRU	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
	ReSTR [39]	ViT-B	Transformer	67.22	69.30	64.45	55.78	60.44	48.27	-	-	70.18
	PolyFormer-B	Swin-B	BERT-base	74.82	76.64	71.06	67.64	72.89	59.33	67.76	69.05	71.91
	PolyFormer-L	Swin-L	BERT-base	**75.96**	**78.29**	**73.25**	**69.33**	**74.56**	**61.87**	**69.20**	**70.19**	**72.60**
mIoU	VLT [19]	DN53	Bi-GRU	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65	-
	CRIS [78]	RN101	GPT-2	70.47	73.18	66.10	62.27	68.06	53.68	59.87	60.36	-
	SeqTR [94]	DN53	Bi-GRU	71.70	73.31	69.82	63.04	66.73	58.97	64.69	65.74	-
	RefTr [44]	RN101	BERT-base	74.34	76.77	70.87	66.75	70.58	59.40	66.63	67.39	-
	LAVT [86]	Swin-B	BERT-base	74.46	76.89	70.94	65.81	70.97	59.23	63.34	63.62	-
	PolyFormer-B	Swin-B	BERT-base	75.96	77.09	73.22	70.65	74.51	64.64	69.36	69.88	65.98
	PolyFormer-L	Swin-L	BERT-base	**76.94**	**78.49**	**74.83**	**72.15**	**75.71**	**66.73**	**71.15**	**71.17**	**67.22**

Table 1. Comparison with the state-of-the-art methods on three referring image segmentation benchmarks. RN101 denotes ResNet-101 [26], WRN101 refers to Wide ResNet-101 [90], and DN53 denotes Darknet-53 [67].

Method	Visual Backbone	Text Encoder	RefCOCO			RefCOCO+			RefCOCOg		ReferIt
			val	test A	test B	val	test A	test B	val	test	test
UNTIER-L [11]	RN101	BERT	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77	-
VILLA-L [22]	RN101	BERT	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71	-
RefTr [44]	RN101	BERT-base	85.65	88.73	81.16	77.55	82.26	68.99	79.25	80.01	76.18
SeqTR [94]	DN53	Bi-GRU	87.00	90.15	83.59	78.69	84.51	71.87	82.69	83.37	69.66
MDETR [35]	ENB3	RoBERTa-base	87.51	90.40	82.67	81.13	85.52	72.96	83.35	83.31	-
OFA-B [77]	RN101	Embedding layer	88.48	90.67	83.30	81.39	87.15	74.29	82.29	82.31	-
UniTAB [84]	RN101	RoBERT-base	88.59	91.06	83.75	80.97	85.36	71.55	84.58	84.70	-
OFA-L [77]	RN152	Embedding layer	90.05	**92.93**	85.26	**85.80**	**89.87**	**79.22**	**85.89**	**86.55**	-
PolyFormer-B	Swin-B	BERT-base	89.73	91.73	86.03	83.73	88.60	76.38	84.46	84.96	80.90
PolyFormer-L	Swin-L	BERT-base	**90.38**	92.89	**87.16**	84.98	89.77	77.97	85.83	85.91	**81.50**

Table 2. Comparison with the state-of-the-art methods on three referring expression comprehension benchmarks. ENB3 denotes EfficientNet-B3 [76].

Training Details. To leverage large-scale image-text dataset with grounded box annotations, we first pre-train PolyFormer on the REC task with the combination of Visual Genome [40], RefCOCO [89], RefCOCO+ [89], RefCOCOg [60, 62], and Flickr30k-entities [64]. During the multi-task fine-tuning stage, for RefCOCO, RefCOCO+, and RefCOCOg datasets, we train the model for both RIS and REC on a combined training dataset [4] with all validation and testing images removed; for ReferIt [36] dataset, only the ReferIt training set is used. We use the AdamW optimizer [54] with $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 1 \times 10^{-8}$. The initial learning rate is 5×10^{-5} with polynomial learning rate decay. We train the model for 20 epochs during pre-

training and 100 epochs for fine-tuning with a batch size of 160 and 128, respectively. The coefficients for losses are set as $\lambda_{box} = 0.1$, $\lambda_{poly} = 1$ and $\lambda_{cls} = 5 \times 10^{-4}$. Images are resized to 512×512, and polygon augmentation is applied with a probability of 50%. The 2D coordinate embedding codebook is constructed with 64×64 bins.

4.3. Main Results

Referring Image Segmentation. We compare PolyFormer with the state-of-the-art methods in Table 1. It can be observed that PolyFormer models outperform previous methods on each split of the three datasets under all metrics by a clear margin. First, on the RefCOCO dataset, compared with the recent LAVT [86], PolyFormer-B achieves

Method	Visual Backbone	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
CMSA+RNN [87]	ResNet-50	40.2	36.9	43.5
URVOS [72]	ResNet-50	51.5	47.3	56.0
CITD [46]	ResNet-101	56.4	54.8	58.1
ReferFormer [80]	Swin-L	60.5	57.6	63.4
ReferFormer [80]	Video-Swin-B	61.1	**58.1**	64.1
PolyFormer-B†	Swin-B	60.9	56.6	65.2
PolyFormer-L†	Swin-L	**61.5**	57.2	**65.8**

Table 3. Comparison with the state-of-the-art methods on Ref-DAVIS17. †means our model is trained on image datasets only. ReferFormer is trained on both image and video datasets.

better results with absolute mIoU gains of 1.5%, 0.2%, and 2.28% on the three splits. Second, on the more challenging RefCOCO+ dataset, PolyFormer-B significantly outperforms the previous state-of-the-art RefTr [44] by absolute mIoU margins of 3.9%, 3.93%, 5.24% on the validation, test A and test B sets, respectively. Third, on the most challenging RefCOCOg dataset where language expressions are longer and more complex, PolyFormer-B achieves notable performance improvements of 2.73% and 2.49% mIoU points on the validation and test sets compared with the second-best method RefTr [44]. Using the stronger Swin-L backbone and a larger encoder and decoder, PolyFormer-L achieves consistent performance gains of around 1∼2 absolute points over PolyFormer-B across all datasets. These results demonstrate the superiority of our polygon-based method PolyFormer over the previous mask-based methods.

Referring Expression Comprehension. We further evaluate PolyFormer on the REC datasets and compare the performance with other state-of-the-art methods in Table 2. OFA [77] is a foundation model which provides a unified interface for a wide range of tasks from different modalities. Compared with OFA-L, PolyFormer-L achieves better results on RefCOCO and comparable results on the remaining two datasets. Note that OFA utilizes 19M image-text data for pre-training, while PolyFormer only uses 6M image-text pairs. Another recent paper SeqTR [94] also proposes a seq2seq model for multi-task learning of RIS and REC, but they observe performance degradation for each task when trained jointly. PolyFormer-B significantly outperforms SeqTR by absolute margins of 2.73%, 5.04% and 1.77% on the three validation sets. These results illustrate the effectiveness of our unified framework.

Zero-shot Transfer to Referring Video Object Segmentation. To further test its generalization ability, we evaluate PolyFormer on the referring video object segmentation dataset Ref-DAVIS17 [38] in a zero-shot manner. We simply view the video data as a sequence of images, and apply PolyFormer to the video frame-by-frame. As shown in Table 3, PolyFormer-L achieves 61.5% $\mathcal{J}\&\mathcal{F}$ without any

	Decoder	RefCOCO	RefCOCO+	RefCOCOg
RIS	Classification	74.11	68.79	67.69
	Regression	**75.96** (+1.85)	**70.65** (+1.86)	**69.36** (+1.67)
REC	Classification	87.03	81.35	82.21
	Regression	**89.73** (+2.70)	**83.73** (+2.38)	**84.46** (+2.25)

Table 4. Ablation study on regression-based decoder.

Order	Aug	Multi-task	\<SEP\>	RefCOCO	RefCOCO+	RefCOCOg
✗	✗	✗	✗	55.92	51.64	50.65
✓	✗	✗	✗	68.35	63.28	62.18
✓	✓	✗	✗	72.07	66.68	65.20
✓	✓	✓	✗	75.14	69.86	68.70
✓	✓	✓	✓	75.96	70.65	69.36

Table 5. Ablation study on target sequence construction.

finetuning on the video data. It is even better than the state-of-the-art ReferFormer [80], which is fully trained on referring video segmentation data.

4.4. Ablation Studies

In this section, we perform extensive ablation studies on the RefCOCO, RefCOCO+ and RefCOCOg validation sets to study the effects of core components of PolyFormer. All ablation experiments are performed on PolyFormer-B.

Coordinate Classification *vs.* Regression. We implement a classification-based model for coordinate prediction following [9, 84, 94]. Specifically, the coordinates are quantized into discrete bins and a classifier is adopted to output the discrete tokens. As shown in Table 4, the regression-based model consistently outperforms the classification-based model on all datasets, *e.g.*, +1.86 for RIS and +2.38 for REC on the RefCOCO+ dataset. This shows that the regression-based model is a better choice for geometric localization tasks than the classification-based counterpart.

Component Analysis of Target Sequence Construction. We study the effects of several components in constructing the target sequence, including (1) polygon ordering, where the polygon vertices are ordered clock-wise and the starting point is set as the vertex closest to the image origin; (2) data augmentation, where we generate polygons at different levels of granularity on-the-fly during training; (3) multi-task learning of RIS and REC; and (4) separator token, where we add \<SEP\> token to handle multi-polygon cases. The results are shown in Table 5. Using randomly ordered polygons only achieves 55.92% mIoU on RefCOCO. Polygon ordering is essential and leads to a substantial improvement of 12.43%. Both polygon augmentation and multi-task learning are beneficial, with gains of 3.72% and 3.07%, respectively. The separator token \<SEP\> brings a 0.82% increase of mIoU. This small gain is reasonable considering

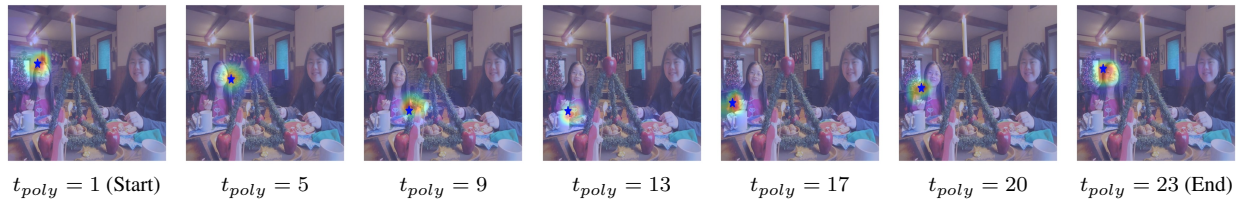Expression: "an Asian girl with a pink shirt eating at the table"

| $t_{poly} = 1$ (Start) | $t_{poly} = 5$ | $t_{poly} = 9$ | $t_{poly} = 13$ | $t_{poly} = 17$ | $t_{poly} = 20$ | $t_{poly} = 23$ (End) |

Figure 6. The cross-attention maps of the decoder when generating the polygon. $\star$ is the 2D vertex prediction at inference step t_{poly}.

(a) "a dark grey dog on a light grey round bed wearing a red collar"
(b) "girl in purple"
(c) "zebra eating grass with a goose in front of it"
(d) "a black car parked at a transportation terminal"
(e) "A gentleman otter in a 19th century portrait"
(f) "A pikachu fine-dining with a view to the Eiffel Tower"
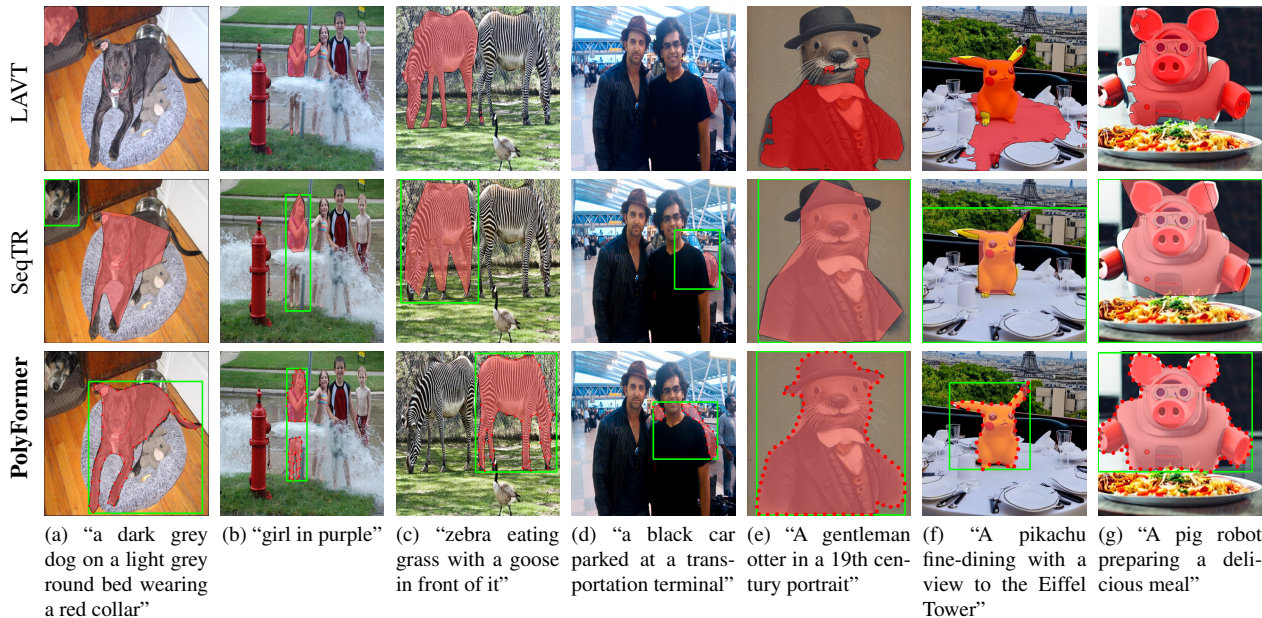(g) "A pig robot preparing a delicious meal"

Figure 7. The results of LAVT [86] (top), SeqTR [94] (middle), and PolyFormer (bottom) on RefCOCOg test set (a-d) and images generated by [69] (e-g). LAVT is for referring image segmentation only. For SeqTR, we generate the bounding boxes and segmentation masks from the task-specific models as they perform better than the multi-task variant.

only a small number of samples have multiple polygons. We observe similar trends in the remaining two datasets.

4.5. Visualization Results

Cross-attention Map. When generating the vertex tokens, the regression-based decoder computes the self-attention over the preceding tokens and cross-attention over the multi-modal feature. Here we visualize the cross-attention map (averaged over all layers and heads) when the model predicts a new token. Fig. 6 shows the cross-attention maps at different steps of the polygon generation. We observe that the cross-attention map concentrates on the object referred to by the sentence, and moves around the object boundary during the polygon generation process.

Prediction Visualization. We show the visualization results of PolyFormer on the RefCOCOg test set in Fig. 7 (a)-(d). It can be seen that PolyFormer is able to segment the referred object in challenging scenarios, e.g., instances with occlusion and complex shapes, and instances that are partially displayed or require complex language understanding.

We also show the results on images generated by Stable Diffusion [69] in Fig. 7 (e)-(g). PolyFormer demonstrates good generalization ability on synthetic images and text descriptions that have never been seen during training. In contrast, the state-of-the-art LAVT [86] and SeqTR [94] fail to generate satisfactory results. More visualization results are provided in the supplementary material.

5. Conclusion

In this work, we propose PolyFormer, a simple and unified framework for referring image segmentation and referring expression comprehension. It is a sequence-to-sequence framework that can naturally fuse multi-modal features as the input sequence and multi-task predictions as the output sequence. Moreover, we design a novel regression-based decoder to generate continuous 2D coordinates without quantization errors. PolyFormer achieves competitive results for RIS and REC and shows good generalization to unseen scenarios. We believe this simple framework can be extended to other tasks beyond RIS and REC.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with Polygon-RNN++. In *CVPR*, pages 859–868, 2018. 1

[2] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: Real-time instance segmentation. In *ICCV*, pages 9157–9166, 2019. 1

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 1, 3

[4] Zhaowei Cai, Gukyeong Kwon, Avinash Ravichandran, Erhan Bas, Zhuowen Tu, Rahul Bhotika, and Stefano Soatto. X-DETR: A Versatile Architecture for Instance-wise Vision-Language Tasks. In *ECCV*, pages 290–308, 2022. 2, 6

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 2

[6] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a Polygon-RNN. In *CVPR*, pages 5230–5238, 2017. 1, 3

[7] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *ICCV*, pages 7454–7463, 2019. 1, 2, 6

[8] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, pages 4974–4983, 2019. 1

[9] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *ICLR*, 2022. 2, 3, 4, 7

[10] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J. Fleet, and Geoffrey Hinton. A unified sequence interface for vision tasks. In *NeurIPS*, 2022. 2, 3, 4

[11] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, pages 104–120, 2020. 6

[12] Yi-Wen Chen, Yi-Hsuan Tsai, Tiantian Wang, Yen-Yu Lin, and Ming-Hsuan Yang. Referring expression object segmentation with caption-aware consistency. In *BMVC*, 2019. 2

[13] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *SSST@EMNLP*, pages 103–111, 2014. 2

[14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014. 1, 3

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, June 2016. 1

[16] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, pages 3150–3158, 2016. 1

[17] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In *NeurIPS*, 2021. 4

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 4, 5

[19] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *ICCV*, pages 16321–16330, 2021. 1, 2, 6

[20] Hugo Jair Escalante, Carlos A. Hernández, Jesus A. Gonzalez, A. López-López, Manuel Montes, Eduardo F. Morales, L. Enrique Sucar, Luis Villaseñor, and Michael Grubinger. The segmented and annotated IAPR TC-12 benchmark. *CVIU*, 114(4):419–428, 2010. 5

[21] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder fusion network with co-attention embedding for referring image segmentation. In *CVPR*, pages 15506–15515, 2021. 1, 2, 6

[22] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. In *NeurIPS*, 2020. 6

[23] Spyros Gidaris and Nikos Komodakis. LocNet: Improving localization accuracy for object detection. In *CVPR*, pages 789–798, 2016. 2

[24] Ross Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015. 2

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pages 2961–2969, 2017. 1, 2, 4

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[27] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. *PAMI*, 44(2):684–696, 2019. 2

[28] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, pages 1115–1124, 2017. 2

[29] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, pages 108–124, 2016. 1, 2

[30] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In *CVPR*, pages 4424–4433, 2020. 1, 2, 6

[31] Shaofei Huang, Tianrui Hui, Si Liu, Guanbin Li, Yunchao Wei, Jizhong Han, Luoqi Liu, and Bo Li. Referring image segmentation via cross-modal progressive comprehension. In *CVPR*, pages 10488–10497, 2020. 1, 2, 6

[32] Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In *ECCV*, pages 59–75, 2020. 1, 2, 6

[33] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeurIPS*, 2015. 2

[34] Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu Tan. Locate then segment: A strong pipeline for referring image segmentation. In *CVPR*, pages 9858–9867, 2021. 6

[35] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR - modulated detection for end-to-end multi-modal understanding. In *ICCV*, pages 1780–1790, 2021. 2, 6

[36] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014. 5, 6

[37] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. In *ICLR*, 2021. 4

[38] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV*, pages 123–141, 2018. 2, 7

[39] Namyup Kim, Dongwon Kim, Cuiling Lan, Wenjun Zeng, and Suha Kwak. Restr: Convolution-free referring image segmentation using transformers. In *CVPR*, pages 18145–18154, 2022. 1, 2, 6

[40] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 6

[41] Justin Lazarow, Weijian Xu, and Zhuowen Tu. Instance segmentation with mask-supervised polygonal boundary transformers. In *CVPR*, pages 4372–4381, 2022. 1, 3

[42] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 1

[43] Mengtian Li, Laszlo Jeni, and Deva Ramanan. Brute-force facial landmark analysis with a 140,000-way classifier. In *AAAI*, volume 32, 2018. 2

[44] Muchen Li and Leonid Sigal. Referring transformer: A one-step approach to multi-task visual grounding. In *NeurIPS*, 2021. 2, 6, 7

[45] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *CVPR*, pages 5745–5753, 2018. 1

[46] Chen Liang, Yu Wu, Tianfei Zhou, Wenguan Wang, Zongxin Yang, Yunchao Wei, and Yi Yang. Rethinking cross-modal interaction from a top-down perspective for referring video object segmentation. *arXiv preprint arXiv:2106.01061*, 2021. 7

[47] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. PolyTransform: Deep Polygon Transformer for Instance Segmentation. In *CVPR*, pages 9128–9137, 2020. 1, 3

[48] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *CVPR*, pages 10880–10889, 2020. 2

[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, pages 740–755, 2014. 1

[50] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *ICCV*, pages 1271–1280, 2017. 1, 2

[51] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *PAMI*, 44(9):4761–4775, 2022. 6

[52] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018. 1

[53] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 4, 5

[54] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[55] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 1

[56] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-IO: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 2, 3, 4

[57] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, pages 10437–10446, 2020. 1

[58] Gen Luo, Yiyi Zhou, Rongrong Ji, Xiaoshuai Sun, Jinsong Su, Chia-Wen Lin, and Qi Tian. Cascade grouped attention network for referring expression segmentation. In *ACMMM*, pages 1274–1282, 2020. 6

[59] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *CVPR*, pages 10034–10043, 2020. 2, 6

[60] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016. 2, 5, 6

[61] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *ECCV*, pages 630–645, 2018. 1

[62] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, pages 792–807, 2016. 5, 6

[63] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *CVPR*, pages 8530–8539, 2020. 3

[64] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, pages 2641–2649, 2015. 6

[65] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 3

[66] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020. 1, 3, 4

[67] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 6

[68] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[69] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 8

[70] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. LabelMe: a database and web-based tool for image annotation. *IJCV*, 77(1):157–173, 2008. 1

[71] Jürgen Schmidhuber, Sepp Hochreiter, et al. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997. 2, 3

[72] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, pages 208–223, 2020. 7

[73] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *ECCV*, pages 38–54, 2018. 1, 2

[74] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014. 1, 3

[75] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*, pages 5100–5111, 2019. 1

[76] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 6

[77] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, pages 23318–23340, 2022. 2, 3, 5, 6, 7

[78] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. CRIS: Clip-driven referring image segmentation. In *CVPR*, pages 11686–11695, 2022. 1, 2, 6

[79] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *ICLR*, 2022. 4

[80] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, pages 4974–4984, 2022. 2, 7

[81] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, pages 12193–12202, 2020. 1, 3

[82] Chenyang Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *TIP*, 7(3):359–369, 1998. 3

[83] Sibei Yang, Meng Xia, Guanbin Li, Hong-Yu Zhou, and Yizhou Yu. Bottom-up shift and reasoning for referring image segmentation. In *CVPR*, pages 11266–11275, 2021. 6

[84] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *ECCV*, pages 521–539, 2022. 3, 4, 6, 7

[85] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *ICCV*, pages 4683–4693, 2019. 2

[86] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. LAVT: Language-Aware Vision Transformer for Referring Image Segmentation. In *CVPR*, pages 18155–18165, 2022. 1, 2, 6, 8

[87] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *CVPR*, pages 10502–10511, 2019. 1, 2, 7

[88] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. MAttNet: Modular attention network for referring expression comprehension. In *CVPR*, pages 1307–1315, 2018. 2

[89] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016. 2, 5, 6

[90] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, pages 87.1–87.12, 2016. 6

[91] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *CVPR*, pages 4158–4166, 2018. 2

[92] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, pages 5579–5588, 2021. 1

[93] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[94] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. SeqTR: A Simple Yet Universal Network for Visual Grounding. In *ECCV*, pages 598–615, 2022. 2, 3, 4, 5, 6, 7, 8

[95] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton Van Den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *CVPR*, pages 4252–4261, 2018. 2