Parallel Vertex Diffusion for Unified Visual Grounding

Zesen Cheng[1,2], **Kehan Li**[1,2], **Peng Jin**[1,2], **Siheng Li**[4]
Xiangyang Ji[4], **Li Yuan**[1,2,3], **Chang Liu**[4] †, **Jie Chen**[1,3,2] †

[1] School of Electronic and Computer Engineering, Peking University, Shenzhen, China
[2] AI for Science (AI4S)-Preferred Program, Peking University Shenzhen Graduate School, China
[3] Peng Cheng Laboratory, Shenzhen, China
[4] Tsinghua University, Beijing, China

Abstract

Unified visual grounding (UVG) capitalizes on a wealth of task-related knowledge across various grounding tasks via one-shot training, which curtails retraining costs and task-specific architecture design efforts. Vertex generation-based UVG methods achieve this versatility by unified modeling object box and contour prediction and provide a text-powered interface to vast related multi-modal tasks, *e.g.*, visual question answering, and captioning. However, these methods typically generate vertexes sequentially through autoregression, which is prone to be trapped in error accumulation and heavy computation, especially for high-dimension sequence generation in complex scenarios. In this paper, we develop Parallel Vertex Diffusion (PVD) based on the parallelizability of diffusion models to accurately and efficiently generate vertexes in a parallel and scalable manner. Since the coordinates fluctuate greatly, it typically encounters slow convergence when training diffusion models without geometry constraints. Therefore, we consummate our PVD by two critical components, *i.e.*, center anchor mechanism and angle summation loss, which serve to normalize coordinates and adopt a differentiable geometry descriptor from the point-in-polygon problem of computational geometry to constrain the overall difference of prediction and label vertexes. These innovative designs empower our PVD to demonstrate its superiority with state-of-the-art performance across various grounding tasks.

Introduction

Visual grounding is an essential vision-language task that establishes a fine-grained correspondence between images and texts by grounding a given referring expression on an image (Li et al. 2022). This task is divided into two sub-tasks based on the manner of grounding: Referring Expression Comprehension (REC) based on bounding box (Hu et al. 2016), and Referring Image Segmentation (RIS) based on mask (Hu, Rohrbach, and Darrell 2016). REC and RIS are chronically regarded as separate tasks with different technology routes, requiring complex task-specific design and repeated training. Since they share high similarities, Unified Visual Grounding (UVG) is proposed to simultaneously perform REC and RIS via single-shot training, reducing the retraining cost and task-specific design labor (Zhu et al. 2022).

† Corresponding Author

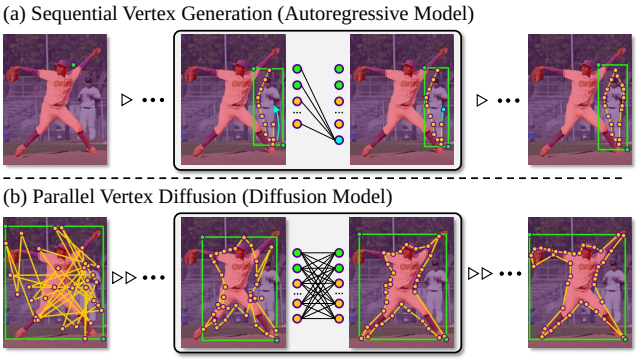


Figure 1: Comparison between (a) Sequential Vertex Generation and (b) our Parallel Vertex Diffusion. The former is easily trapped in error accumulation and heavy computation. The latter parallel refines all vertexes for conquering these issues. The related referring expression is "*center baseman*".

Vertex generation is an advanced UVG scheme because its compatibility with text generation enables unifying visual grounding with other multi-modal tasks (*e.g.*, VQA, Caption), potentially providing an interface to build a multi-modal universal model (Chen et al. 2022b). Previous vertex generation methods represent boxes and masks as vertex sequences of variable length and sequentially generate them via an autoregressive model (Zhu et al. 2022). They easily get stuck in error accumulation when scaling to high-dimension vertex number settings, causing inferior fitting to objects with complex contours. This flaw is mainly attributed to the sequential nature of autoregressive model (Lin et al. 2020) because upstream error vertexes will perturb the prediction of downstream vertexes during sequential generation. In Fig. 1(a), we find that sequential vertex generation methods easily generate error vertexes when upstream vertexes don't hit the right object. Moreover, these methods require heavy computations when scaling to high-dimension data, because the iterations sharply increase with the data dimension (Chen et al. 2023b).

In this work, we design Parallel Vertex Diffusion (PVD) to generate vertexes by adopting the diffusion to model the distribution of object box and contour. Owing to the parallel nature of the diffusion model, PVD can accurately and

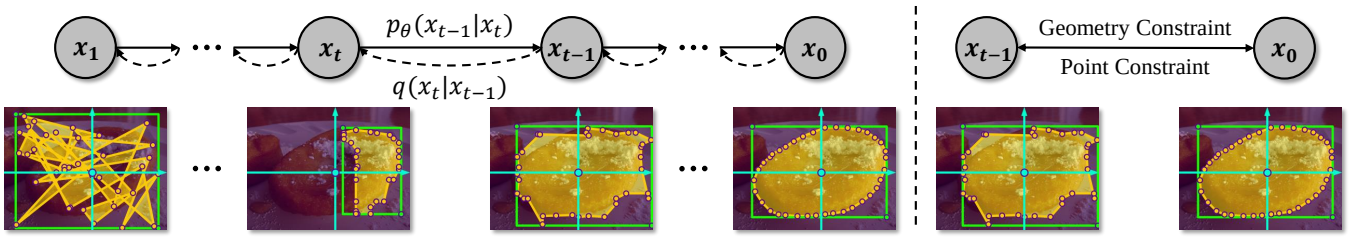


Figure 2: Conceptual mechanism of our PVD, which parallel refines sampled noises to accurate vertex vectors. CAM and ASL are extra designed to stabilize the coordinate fluctuation and provide geometry constraints for better convergence.

efficiently generate high-dimension vertex vectors, avoiding the error accumulation and heavy computation issues of sequential vertex generation. Fig. 1(b) conceptually shows how PVD works. Noisy vertexes are first sampled from Gaussian distribution and then are gradually refined to follow the accurate distribution of the object vertexes. Moreover, our PVD typically encounters slow convergence due to drastic coordinate fluctuation and the lack of geometry constraint. Therefore, we propose to consummate our PVD by Center Anchor Mechanism (CAM) and Angle Summation Loss (ASL). CAM predicts a center point as an anchor to convert coordinates as normalized offset values, stabilizing the coordinate fluctuation. ASL derives the Angle Summation algorithm (Sutherland, Sproull, and Schumacker 1974) from the point-in-polygon problem of computational geometry to provide a differentiable geometry descriptor for constraining the geometry differences between prediction and ground truth vertexes. The overall conceptual mechanism of our PVD is shown in Fig. 2.

In summary, our paper has three contributions: (1) We develop PVD based on the parallelizability of diffusion model to generate vertexes in a more efficient and scalable manner than sequential vertex generation. (2) Based on the Angle Summation algorithm of computational geometry, We design ASL, an effective geometry consistency loss, to improve the convergence of our PVD. (3) The empirical results show that our PVD achieves SOTA on both REC and RIS tasks, and can better handle high-dimension settings with lower computation cost than sequential vertex generation.

Related Work

Unified Visual Grounding

Visual grounding aims at grounding a description on an image. It has two sub-tasks: REC for box-level grounding (Hu et al. 2016) and RIS for mask-level grounding (Hu, Rohrbach, and Darrell 2016). Recent research gradually focuses on UVG because it can leverage data from multiple tasks to explore task relevance for mutual promotion. UVG generally has three types: (1) **Two-stage** (Yu et al. 2018; Liu et al. 2019; Chen et al. 2019) methods use text to retrieve the most confident proposal regions extracted by an extra detector for acquiring results. (2) **Multi-branch** (Luo et al. 2020; Li and Sigal 2021; Su et al. 2023) methods are proposed to assign task-specific branches to different tasks for joint end-to-end optimization. (3) **Sequential Vertex Generation** (SVG) (Zhu et al. 2022) methods model both REC and RIS as a vertex generation problem and adopt an autoregressive model (Chen et al. 2022a) to sequentially generate vertices of objects, which is currently the most popular unified visual grounding scheme (Zhu et al. 2022). Because of the sequential generation nature, SVG is easily trapped in error accumulation and heavy computation. This work proposes parallel vertex diffusion to avoid these issues.

Generative Model for Perception

Since Pix2Seq (Chen et al. 2022a) first claims that sequence generation modeling is a simple and generic framework for object detection, the generative model for perception is gradually gaining more attention. Following Pix2Seq, Pix2Seq v2 (Chen et al. 2022b) is proposed as a general vision interface for multiple location tasks, *e.g.*, object detection, instance segmentation, keypoint detection, and image caption. Although sequence generation modeling shows its potential, its fundamental generative architecture (autoregressive model) is hard to scale to high-dimension data (Chen et al. 2023b). Subsequently, Pix2Seq-D shows that diffusion model (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2021) is a better fundamental generative architecture for processing high-dimension tasks. Then researchers expand diffusion model to other high-dimension tasks, *e.g.*, semantic segmentation (Amit et al. 2021; Wolleb et al. 2022; Graikos et al. 2022) and pose estimation (Holmquist and Wandt 2023; Choi, Shim, and Kim 2023). In this paper, we propose parallel vertex diffusion for leveraging the high-dimension process ability of diffusion model.

Method

Overall Pipeline

To better understand our workflow, we describe the training and inference processes of the overall pipeline (Fig. 3).

Training. An image-text pair $\{I, T\}$ is input into visual and linguistic encoder to extract cross-modal features $\mathcal{F}_c$. The cross-modal features are then used to regress center point c. The center point is used to normalize the bounding box and mask contour vertexes. The final step is to calculate loss:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_p + \mathcal{L}_g, \tag{1}$$

where $\mathcal{L}_c$ is the center point loss (Eq. 4) for optimizing center point prediction, $\mathcal{L}_p$ is the point-to-point reconstruction loss (Eq. 7) for point consistency, and $\mathcal{L}_g$ is the angle summation loss (Eq. 10) for geometry consistency. The below sections will describe how to calculate these losses in detail.

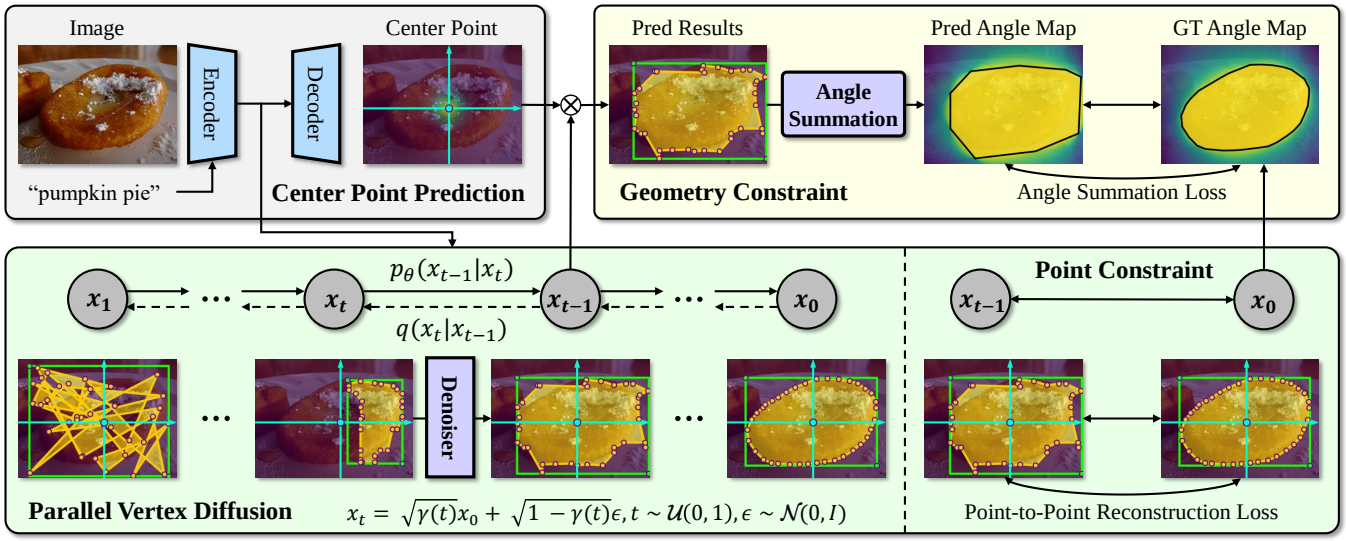


Figure 3: Detailed workflow has three components: (1) *Center Point Prediction* is used to extract cross-modal features and predict center point. (2) *Parallel Vertexes Diffusion* is used to generate normalized vertexes by iteratively applying "Denoiser" to noisy state x_T and provides point level constraint during training phrase. (3) *Geometry Constraint* utilizes angle summation algorithm to ensure the geometry consistency between prediction vertexes and ground truth vertexes for better optimization.

Inference. Same as the training process, we first predict the center point c. Then sampling a noisy state x_t from standard Gaussian distribution $\mathcal{N}(0, I)$. The noise is iteratively denoised to clean vertex vector V_0 by denoiser f_θ. Finally, denormalizing the coordinates of vertexes and convert the denormalized coordinates of vertexes to bounding box and binary mask by toolbox of COCO (Lin et al. 2014).

Cross-modal Feature Extraction

The first step is to prepare cross-modal fusion features. An image-text pair $\{I \in \mathbb{R}^{h\times w\times 3}, T \in \mathbb{N}^n\}$ is first sampled from dataset, where h, w, and n denote image height, image width, and the number of words. To encode image, the image is input in visual backbone (*e.g.*, Darknet53 (Redmon and Farhadi 2018), Swin (Liu et al. 2021)) for acquiring multi-scale visual features $\{\mathbf{F}_{v_1} \in \mathbb{R}^{\frac{h}{8}\times\frac{w}{8}\times d_1}, \mathbf{F}_{v_2} \in \mathbb{R}^{\frac{h}{16}\times\frac{w}{16}\times d_2}, \mathbf{F}_{v_3} \in \mathbb{R}^{\frac{h}{32}\times\frac{w}{32}\times d_3}\}$. To encoder text, the word tokens of text are input in language backbone (*e.g.*, Bert (Devlin et al. 2019)) for obtaining linguistic features $F_l \in \mathbb{R}^{n\times d_l}$. Then we blend visual and linguistic features via element-wise multiplication and multi-scale deformable transformer (MSDeformAttn) (Zhu et al. 2021):

$$\mathbf{F}_{c_i} = \texttt{MSDeformAttn}(\texttt{MLP}(\mathbf{F}_{v_i}) \odot \texttt{MLP}(\mathbf{F}_l)), \quad (2)$$

where $\odot$ denotes element-wise multiplication, $\mathrm{MLP}(\cdot)$ are three fully-connected layers and $\mathbf{F}_{c_i} \in \mathbb{R}^{h_i\times w_i\times d}$ denotes cross-modal features of i-th stage. d is set to 256.

Center Anchor Mechanism

Center Point Prediction. We hope to locate the rough center point $c \in \mathbb{R}^2$ of the object referred by text. Following previous advanced keypoint prediction network (Law and Deng 2018; Tian et al. 2019), the center point is represented as a Gaussian heatmap $Y \in [0,1]^{h\times w}$ by a Gaussian kernel:

$$Y_{ij} = \exp\left(-\frac{(i-c_i)^2 + (j-c_j)^2}{2\sigma^2}\right), \quad (3)$$

where σ^2 is an image size-adaptive standard deviation. Firstly, we use Eq. 3 to generate target Gaussian heatmap $\overline{Y}$ of ground truth center point $\overline{c}$. Then the cross-modal features with largest spatial shape $\mathbf{F}_{v_1}$ are used to generate prediction heatmap Y, *i.e.*, $Y = \mathrm{MLP}(\mathbf{F}_{v_1})$. Finally, a focal loss (Lin et al. 2017) is adopted as the training objective:

$$\mathcal{L}_c = \frac{1}{hw}\sum_{i,j}^{h,w}\begin{cases}(1-Y_{ij})^\alpha \log(Y_{ij}), \overline{Y}_{ij} = 1,\\ (1-Y_{ij})^\beta Y_{ij}^\alpha \log(1-Y_{ij}), \overline{Y}_{ij} < 1,\end{cases} \quad (4)$$

where α and β are set to 2 and 4 according to previous works (Law and Deng 2018). We set peak value point of prediction heatmap as prediction center point $c = \{i_c, j_c\}$.

Coordinate Normalization. To normalize a single point $p = \{i, j\}$, we adopt the scale and bias strategy:

$$\hat{p} = (\hat{i}, \hat{j}) = \left(\frac{i-i_c}{h}, \frac{j-j_c}{w}\right), \quad (5)$$

where $\hat{p}$ denotes the point after normalizing.

Parallel Vertex Diffusion

Our implementation mainly refers to Bit Diffusion (Chen, Zhang, and Hinton 2023). The details are described below:

Training phrase. Given an image-text pair $\{I, T\}$, its related box and mask are $B \in \mathbb{R}^{2\times 2}$ and $M \in \{0,1\}^{h\times w}$. ❶ *Preparation*: The box vertexes are $V_b = \{p_b^1 = (i_b^1, j_b^1), p_b^2 = (i_b^2, j_b^2)\}$. The mask vertexes are sampled from mask contour via a classical contour detection algorithm (Suzuki et al. 1985): $V_m = \{p_m^1 =$

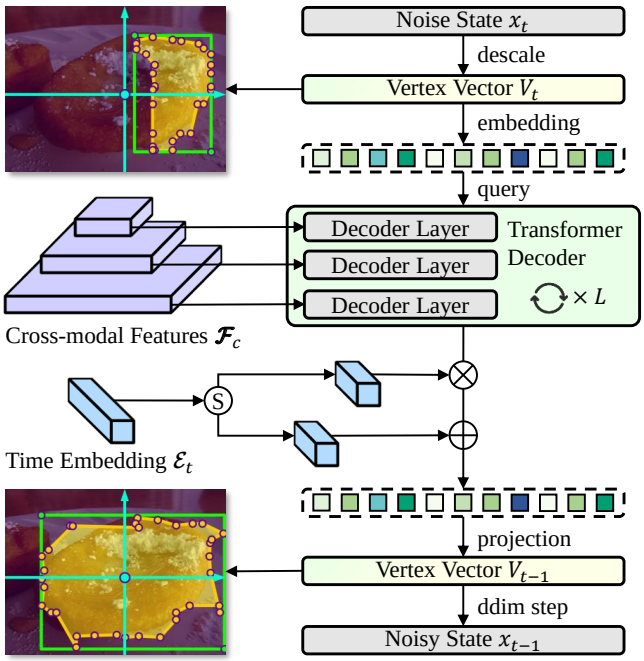


Figure 4: Denoiser. It is the parameterization network of diffusion model reverse process $p_\theta(x_{t-1}|x_t)$. Noisy state x_t is descaled to vertex vector V_t by Eq. 8. The vertex embedding tool is the 2D coordinate embedding (Liu et al. 2023). Refined vertex vector V_{t-1} and next noisy state x_{t-1} are the output of Denoiser. They are respectively used to get prediction results and attend training objective (Eq. 7).

$(i_m^1, j_m^1), \cdots, p_m^N = (i_m^N, j_m^N)\}$, where N is the sampling number. V_m can be seen as a sampling of an unknown distribution decided by the mask contour. Direct fitting of this distribution is difficult. To reduce the difficulty, we associate V_m and V_b to vertex set V for box and mask co-generation. ❷ *Forward transition*: The vertex set is normalized to $\hat{v}$ by Eq. 5. $\hat{V}$ is set as the initial state x_0 of diffusion model and is forward transitioned to noisy state x_t, *i.e.*, $q(x_t|x_0)$:

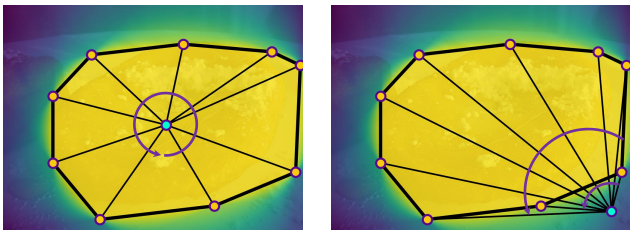
$$x_t = \sqrt{\gamma(t)}x_0 + \sqrt{1-\gamma(t)}\epsilon, \tag{6}$$

where ϵ and t are drawn from the standard normal distribution $\mathcal{N}(0, I)$ and uniform distribution $\mathcal{U}(0, 1)$. $\gamma(t) \in [-1, 1]$ is a scheduling function based on cosine function. ❸ *Point constraint*: For training the denoiser, the noisy state x_t is reversely transitioned to the noisy state x_{t-1}, *i.e.*, $p_\theta(x_{t-1}|x_t)$. x_{t-1} is required to approach initial state x_0:

$$\mathcal{L}_p = \mathbb{E}_{t\sim\mathcal{U}(0,1),\epsilon\sim\mathcal{N}(0,I)}\|f_\theta(x_t, \mathcal{F}_c, t) - x_0\|^2, \tag{7}$$

where $f_\theta(\cdot)$ is the parameterized Denoiser, x_t is derived by Eq. 6, and $\mathcal{F}_c = \{\mathbf{F}_{c_1}\mathbf{F}_{c_2}, \mathbf{F}_{c_3}\}$ denotes all of the cross-modal features. Eq. 7 is the point-level reconstruction loss.

Inference phrase. Generating the vertex vector of boxes and masks requires T-step state transitions $x_1 \to \cdots \to x_t \to \cdots \to x_0$, where the time interval of a single step is set to $1/T$. Specifically, we first sample a noise x_1 from $\mathcal{N}(0, I)$. Denoiser is then iteratively applied to denoise x_1 to implement the state transition chain for generating x_0.



(a) inside the polygon ($= 2\pi$) (b) outside the polygon ($< 2\pi$)

Figure 5: Angle Summation. (a) If a point is inside the polygon, the sum of the angles between this point and each pair of points making up the polygon is 2π. (b) Otherwise, the sum is $< 2\pi$. According to the relative position between pixels and polygon, an angle distribution can be calculated to reflect the geometry information of polygon vertexes.

Denoiser. Denoiser is the reverse transition function ($x_t \to x_{t-1}$) of the diffusion model. In our paper, the Denoiser firstly converts the noisy state x_t to vertex vector V_t by Eq. 8:

$$V_t = x_t/b, \tag{8}$$

where b is set to 2 according to the ablation of previous research (Chen et al. 2023a). Then the vertex vector is embedded to vertex embedding $\mathcal{Q}_t$ by 2D coordinate embedding (Liu et al. 2023). Three scales of cross-modal features are circularly input into transformer decoder (Cheng et al. 2022) L times to refine noise vertex embedding, where L is 3. After the transformer decoder, the embedding is normalized by time embedding $\mathcal{E}_t$ and is projected to refined vertex vector V_{t-1}. Finally, V_{t-1} is converted to the next noisy state x_{t-1} by DDIM step (Song, Meng, and Ermon 2021).

Geometry Constraint

Because the box vertexes are easy to generate, we mainly focus on providing geometry constraints for mask vertexes.

Angle Summation. Given a set of mask polygon vertexes $V = \{p^1, p^2, \cdots, p^N\}$ with clock-wise order, the angle summation value of the point $a = (x, y)$:

$$\mathcal{A}^{(x,y)} = \sum_{k=1}^{N} \arccos \frac{\overrightarrow{\boldsymbol{p^k a}} \cdot \overrightarrow{\boldsymbol{p^{(k+1)\%N} a}}}{\left|\overrightarrow{\boldsymbol{p^k a}}\right| \cdot \left|\overrightarrow{\boldsymbol{p^{(k+1)\%N} a}}\right|}, \tag{9}$$

where $\%$ denotes modulo operator, p^k and $p^{(k+1)\%N}$ denote two endpoints of k-th polygon edge. To better understand Eq. 9, we provide an intuitive and vivid illustration in Fig. 5.

Training Objective. Given prediction vertexes V and ground truth vertexes $\hat{V}$, the geometry information of V and $\hat{V}$ are resolved to prediction angle summation map $\mathcal{A}$ and ground truth angle summation map $\hat{\mathcal{A}}$ by Eq. 9. We hope the prediction angle summation map to approach the ground truth angle summation for achieving geometry consistency:

$$\mathcal{L}_g = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\|\mathcal{A}_t - \hat{\mathcal{A}}\|^2, \tag{10}$$

where $\mathcal{L}_g$ is named Angle Summation Loss (ASL).

Method	Backbone	RefCOCO			RefCOCO+			RefCOCOg		
		val	test A	test B	val	test A	test B	val(U)	test(U)	val(G)
Single-task										
LBYL	Darknet53	79.67	82.91	74.15	68.64	73.38	59.49	-	-	62.70
TransVG	ResNet101	81.02	82.72	78.35	64.82	70.70	56.94	68.67	67.73	67.02
TRAR	Darknet53	-	81.40	78.60	-	69.10	56.10	68.90	68.30	-
QRNet	Swin-base	84.01	85.85	82.34	72.94	76.17	63.81	71.89	73.03	-
Multi-task										
NMTree	MRCNN-Res101	76.41	81.21	70.09	66.46	72.02	57.52	65.87	66.44	-
MCN	Darknet53	80.08	82.29	74.98	67.16	72.86	57.31	66.46	66.01	-
SeqTR	Darknet53	81.23	85.00	76.08	68.82	75.37	58.78	71.35	71.58	-
PVD	Darknet53	82.51	86.19	76.81	69.48	76.83	59.68	68.40	69.57	67.29
PVD	Swin-base	**84.99**	**88.02**	**80.03**	**74.27**	**79.06**	**65.11**	**74.34**	**74.64**	**71.41**

Table 1: Main results on classical REC datasets. Bold denotes the best performance.

Method	Backbone	RefCOCO			RefCOCO+			RefCOCOg		
		val	test A	test B	val	test A	test B	val(U)	test(U)	val(G)
Single-task										
CRIS	CLIP-Resnet50	69.52	72.72	64.70	61.39	67.10	52.48	59.87	60.36	
LAVT	Swin-base	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09	60.50
CoupleAlign	Swin-base	74.70	**77.76**	**70.58**	62.92	68.34	56.69	62.84	62.22	-
SADLR	Swin-base	74.24	76.25	70.06	64.28	69.09	55.19	**63.60**	63.56	61.16
Multi-task										
NMTree	MRCNN-Res101	56.59	63.02	52.06	47.40	53.01	41.56	46.59	47.88	-
MCN	Darknet53	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
SeqTR	Darknet53	67.26	69.79	64.12	54.14	58.93	48.19	55.67	55.64	-
PVD	Darknet53	68.87	70.53	65.83	54.98	60.12	50.23	57.81	57.17	54.15
PVD	Swin-base	**75.07**	77.29	70.13	**64.39**	**69.15**	**57.19**	63.22	**63.89**	**61.74**

Table 2: Main results on classical RIS datasets. Bold denotes the best performance.

Experiments

Experimental Setup

Our model is evaluated on three standard referring image segmentation datasets: RefCOCO, RefCOCO+, and RefCOCOg. The maximum sentence length n is set to 15 for RefCOCO, RefCOCO+, and 20 for RefCOCOg. The images are resized to 640×640. The training sampling number of mask vertexes N is set by default to 36. Other data preprocessing operations are generally in line with the previous methods (Zhu et al. 2022). During inference phrase, T is set to 4 because DDIM step is adopted for accelerating sampling speed. Based on previous works (Tang et al. 2023; Ding et al. 2021; Wang et al. 2022; Huang et al. 2021), mask IoU and det accuracy are adopted to evaluate the performance of methods. AdamW (Loshchilov and Hutter 2019) is adopted as our optimizer, and the learning rate and weight decay are set to 5e-4 and 5e-2. The learning rate is scaled by a decay factor of 0.1 at the 60th step. We train our models for 100 epochs on 4 NVIDIA V100 with a batch size of 64. All of the quantitative analyses are based on val split of RefCOCO.

Main Results

Referring Expression Comprehension. Single-task part of Tab. 1 reports the comparison results between our method and previous referring expression comprehension methods. From Tab. 1, our PVD boosts previous methods by a clear margin. For example, PVD based on Darknet53 (Redmon and Farhadi 2018) respectively surpasses TransVG (Deng et al. 2021) and TRAR (Zhu et al. 2022) with +2.74∼+6.13% and +3.58∼+7.73% absolute improvement on RefCOCO+. The results show that our PVD generally achieves SOTA when compared to previous referring expression comprehension methods. Besides, we construct a stronger network based on Swin Transformer for achieving higher effectiveness, which performs better than previous swin-base methods, *i.e.*, QRNet (Ye et al. 2022).

Referring Image Segmentation. Single-task part of Tab. 2 reports the comparison results between our method and previous methods. In this case, our PVD can outper-

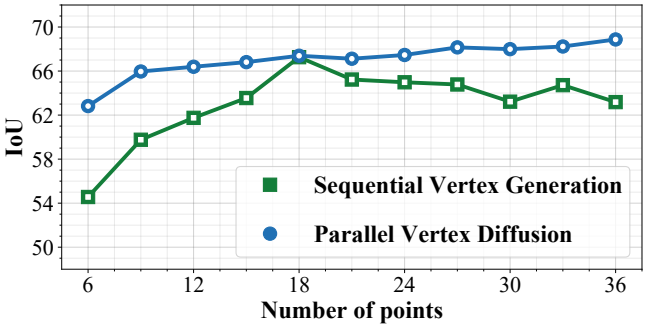
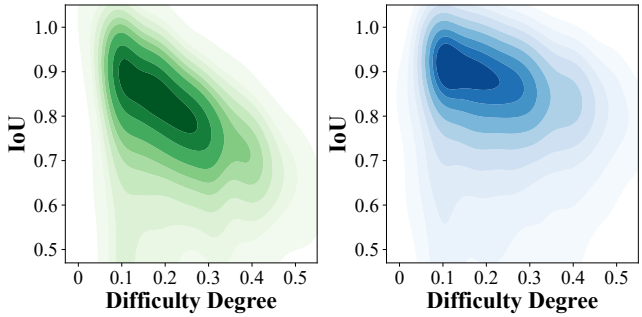


Figure 6: Effectiveness Comparison between sequential vertex generation (SeqTR) and PVD under different point number settings. PVD is only trained once under 36 point settings. SeqTR is retrained under different point settings.

CAM	ASL	Acc (REC)	Acc (RIS)	IoU (RIS)
		79.38	77.67	65.19
✔		81.24 ↑1.86	79.15 ↑1.48	66.78 ↑1.59
	✔	81.17 ↑1.79	78.92 ↑1.25	67.31 ↑2.12
✔	✔	**82.51** ↑**3.13**	**81.03** ↑**3.36**	**68.87** ↑**3.68**

Table 3: Diagnostic Experiments. "Acc" of referring expression comprehension (REC) and referring image segmentation (RIS) denote the precision@0.5, *i.e.*, the rate of samples with IoU > 0.5. "CAM" denotes the center anchor mechanism. "ASL" denotes the angle summation loss.



(a) SVG w/ 18pts (b) PVD w/ 36pts

Figure 7: The density map of samples from (a) SVG and (b) our PVD. Darker area indicates more samples are of the corresponding IoU (%) value and "Difficulty Degree" (the complexity of related object contour).

Number	IoU (SVG)	IoU (PVD)	Speed (SVG)	Speed (PVD)
9pts	60.51	63.14 ↑2.63	101ms	66ms ↓35
18pts	**67.26**	67.39 ↑0.13	192ms	67ms ↓125
27pts	64.78	68.15 ↑3.37	285ms	71ms ↓214
36pts	63.18	**68.87** ↑**5.69**	394ms	75ms ↓319

Table 4: Efficiency Comparison between SVG and PVD. PVD is only trained once under 36 point settings. SVG is retrained under different point settings.

form LAVT (Yang et al. 2022) on all of datasets by +0.77∼+2.34%, CoupleAlign (Zhang et al. 2022) on most of datasets by +0.37∼+1.67%, and SADLR (Yang et al. 2023) on most of datasets by +0.06∼+2.0%, which demonstrates PVD achieves SOTA for referring image segmentation task.

Unified Visual Grounding. Multi-task part of Tab. 1 and Tab. 2 report the comparison results between our method and previous unified visual grounding methods. Compared to SOTA method (SeqTR), our PVD outperforms it by +0.74∼+1.46% for referring expression comprehension and +0.74∼+2.14% for referring image segmentation, which verifies the superiority of our method.

Quantitative Analysis

How does the number of points affect the effectiveness? The main advantage of our PVD compared to sequential vertex generation is easier to scale to high-dimension point number settings. To verify this statement, we check the effectiveness of two methods under different point settings in Fig. 6. The curves contained in this figure provide two justifications: (1) the performance of sequential vertex generation is bottlenecked at 18 points, which substantiates the **dimension dilemma** of sequential vertex generation (the network is perturbed by error accumulation when scaling to a large number of points and is underfitting to complex object when scaling to a small number of points). (2) the performance of our PVD stably increases with the number of points and obviously performs better under high-dimension point number settings , which demonstrates that our method is **more scalable** than sequential vertex generation especially for high-dimension point number settings.

The efficiency of Parallel Vertex Diffusion. Except for effectiveness, efficiency is also an aspect of verifying scalability. To further claim the advantage of our PVD compared to sequential vertex generation, we select several point number settings ("9pts", "18pts", "27pts", "36pts") to benchmark the efficiency of two methods in Tab. 4. This table shows that the sequential vertex generation is heavily impacted by the number of points and needs a large number of extra computation overheads when scaling from a small number of points to a large number of points. For example, the inference speed becomes 4× of previous speed when scaling from "9pts" to "36pts". However, our PVD only requires a little extra overhead to scale to high-dimension point number settings. Specifically, the inference speed only increases +9ms when scaling from "9pts" to "36pts".

The advantage of scalability. The effectiveness and efficiency experiments under different point number settings verify the superior scalability of our PVD, especially for high-dimension settings. The scalability of high-dimension settings enables our PVD to handle complex contours more robustly than sequential vertex generation. To quantitatively analyze the robustness, we define "Difficulty Degree" as the complexity metric and count the "Difficulty-IoU" density

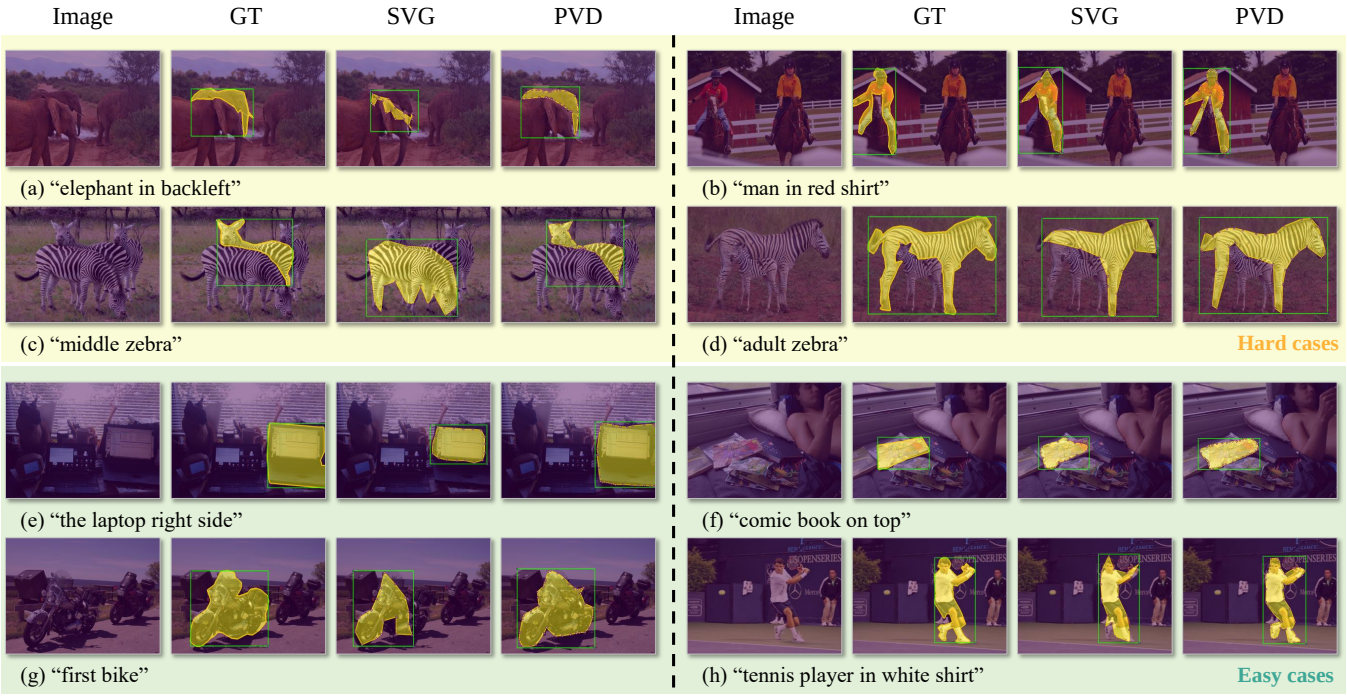


Figure 8: Qualitative results of different cases. “SVG” denotes SeqTR which follows sequential vertex generation. “PVD” is our parallel vertex diffusion method. We select both hard cases whose referred objects have complex contours and easy cases whose referred objects have monotonous contours to show the effectiveness of our PVD.

map of samples in Fig. 7(a) and Fig. 7(b). The calculation details of the metric please refer to appendix. Comparing two figures, we can find that our method still has a high IoU for hard samples (“Difficulty Degree” > 0.2) but the IoU of sequential vertex generation heavily decreases for hard samples, which justifies that our PVD prompts better robustness to hard samples than sequential vertex generation because of the scalability to high-dimension point number settings.

The effectiveness of Center Anchor Mechanism. CAM is proposed to cope with the fluctuations of prediction vertex coordinates. To verify the effectiveness of CAM, we conduct ablation experiments. Tab. 3 shows that PVD w/ CAM boosts vanilla PVD by +1.86% Acc for referring expression comprehension task and +1.59% IoU for referring image segmentation task, which justifies the effectiveness of CAM.

The effectiveness of Angle Summation Loss. Since the original training objective of PVD only achieves point-level constraint between prediction and ground truth vertexes, we propose ASL for geometry constraint. In Tab. 3, PVD w/ ASL improves vanilla PVD by +1.79% box Accuracy for referring expression comprehension task and +2.12% mask IoU for referring image segmentation task. Furthermore, ASL also improves PVD w/ CAM. These results comprehensively verify the effectiveness of ASL.

Qualitative Analysis

As described in Quantitative Analysis, our PVD is more effective and efficient than sequential vertex generation, especially for hard samples. To qualitatively verify this point, we select some easy cases and hard cases to illustrate the grounding difference between two methods. Fig. 8 shows that our PVD generates high-quality vertexes of bounding box and mask contour but the sequential vertex generation easily hits error objects or generates inferior vertexes, which justifies our PVD is more capable of grounding referring expression on the image.

Conclusion

In this paper, we observe that previous UVG methods based on sequential vertex generation easily trap into a dimension dilemma. For high-dimension point number settings, SVG is limited to error accumulation and heavy computation, causing inferior performance and efficiency. For low-dimension point number settings, SVG is hard to handle objects with complex contours. To break this dilemma, we propose PVD to leverage the parallel nature of diffusion model to reduce error accumulation and alleviate inference cost for better scaling to high-dimension point number settings. Considering the poor convergence of directly generating vertex coordinates via a diffusion model, we design CAM and ASL to stabilize greatly fluctuating coordinates and constrain geometry differences between prediction and ground truth. Benchmark experiments show that our PVD achieves SOTA on REC, RIS, and UVG tasks. Comparison experiments verify that our PVD is more scalable and efficient than SVG, especially for high-dimension point number settings. Ablation experiments demonstrate that extra-designed CAM and ASL improve the convergence of our PVD.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (No. 2022ZD0118201), the Natural Science Foundation of China (No. 61972217, 32071459, 62176249, 62006133, 62271465), the Shenzhen Medical Research Funds in China (No. B2302037).

References

Amit, T.; Nachmani, E.; Shaharbany, T.; and Wolf, L. 2021. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*.

Chen, S.; Sun, P.; Song, Y.; and Luo, P. 2023a. Diffusiondet: Diffusion model for object detection. In *Proceedings of the International Conference on Computer Vision*, 19830–19843.

Chen, T.; Li, L.; Saxena, S.; Hinton, G.; and Fleet, D. J. 2023b. A generalist framework for panoptic segmentation of images and videos. In *Proceedings of the International Conference on Computer Vision*, 909–919.

Chen, T.; Saxena, S.; Li, L.; Fleet, D. J.; and Hinton, G. 2022a. Pix2seq: A language modeling framework for object detection. In *Proceddings of the International Conference on Learning Representations*.

Chen, T.; Saxena, S.; Li, L.; Lin, T.-Y.; Fleet, D. J.; and Hinton, G. E. 2022b. A unified sequence interface for vision tasks. *Advances in Neural Information Processing Systems*, 35: 31333–31346.

Chen, T.; Zhang, R.; and Hinton, G. 2023. Analog Bits: Generating Discrete Data using Diffusion Models with Self-Conditioning. In *Proceddings of the International Conference on Learning Representations*.

Chen, Y.-W.; Tsai, Y.-H.; Wang, T.; Lin, Y.-Y.; and Yang, M.-H. 2019. Referring expression object segmentation with caption-aware consistency. In *Proceedings of the British Machine Vision Conference*.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1290–1299.

Choi, J.; Shim, D.; and Kim, H. J. 2023. Diffupose: Monocular 3d human pose estimation via denoising diffusion probabilistic model. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 3773–3780. IEEE.

Deng, J.; Yang, Z.; Chen, T.; Zhou, W.; and Li, H. 2021. Transvg: End-to-end visual grounding with transformers. In *Proceedings of the International Conference on Computer Vision*, 1769–1779.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Ding, H.; Liu, C.; Wang, S.; and Jiang, X. 2021. Vision-language transformer and query generation for referring segmentation. In *Proceedings of the International Conference on Computer Vision*, 16321–16330.

Graikos, A.; Malkin, N.; Jojic, N.; and Samaras, D. 2022. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35: 14715–14728.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Holmquist, K.; and Wandt, B. 2023. Diffpose: Multi-hypothesis human pose estimation using diffusion models. In *Proceedings of the International Conference on Computer Vision*, 15977–15987.

Hu, R.; Rohrbach, M.; and Darrell, T. 2016. Segmentation from natural language expressions. In *Proceedings of the European Conference on Computer Vision*, 108–124. Springer.

Hu, R.; Xu, H.; Rohrbach, M.; Feng, J.; Saenko, K.; and Darrell, T. 2016. Natural language object retrieval. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 4555–4564.

Huang, B.; Lian, D.; Luo, W.; and Gao, S. 2021. Look before you leap: Learning landmark features for one-stage visual grounding. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 16888–16897.

Law, H.; and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision*, 734–750.

Li, L. H.; Zhang, P.; Zhang, H.; Yang, J.; Li, C.; Zhong, Y.; Wang, L.; Yuan, L.; Zhang, L.; Hwang, J.-N.; et al. 2022. Grounded language-image pre-training. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 10965–10975.

Li, M.; and Sigal, L. 2021. Referring transformer: A one-step approach to multi-task visual grounding. *Advances in Neural Information Processing Systems*, 34: 19652–19664.

Lin, C.-C.; Jaech, A.; Li, X.; Gormley, M. R.; and Eisner, J. 2020. Limitations of autoregressive models and their alternatives. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5147–5173.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the International Conference on Computer Vision*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 740–755. Springer.

Liu, D.; Zhang, H.; Wu, F.; and Zha, Z.-J. 2019. Learning to assemble neural module tree networks for visual grounding. In *Proceedings of the International Conference on Computer Vision*, 4673–4682.

Liu, J.; Ding, H.; Cai, Z.; Zhang, Y.; Satzoda, R. K.; Mahadevan, V.; and Manmatha, R. 2023. PolyFormer: Referring Image Segmentation As Sequential Polygon Generation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 18653–18663.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the International Conference on Computer Vision*, 10012–10022.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations*.

Luo, G.; Zhou, Y.; Sun, X.; Cao, L.; Wu, C.; Deng, C.; and Ji, R. 2020. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 10034–10043.

Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *Proceedings of the International Conference on Learning Representations*.

Su, W.; Miao, P.; Dou, H.; Wang, G.; Qiao, L.; Li, Z.; and Li, X. 2023. Language adaptive weight generation for multi-task visual grounding. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 10857–10866.

Sutherland, I. E.; Sproull, R. F.; and Schumacker, R. A. 1974. A characterization of ten hidden-surface algorithms. *ACM Computing Surveys*, 6(1): 1–55.

Suzuki, S.; et al. 1985. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1): 32–46.

Tang, J.; Zheng, G.; Shi, C.; and Yang, S. 2023. Contrastive Grouping with Transformer for Referring Image Segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 23570–23580.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the International Conference on Computer Vision*, 9627–9636.

Wang, Z.; Lu, Y.; Li, Q.; Tao, X.; Guo, Y.; Gong, M.; and Liu, T. 2022. Cris: Clip-driven referring image segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 11686–11695.

Wolleb, J.; Sandkühler, R.; Bieder, F.; Valmaggia, P.; and Cattin, P. C. 2022. Diffusion models for implicit image segmentation ensembles. In *Proceedings of the International Conference on Medical Imaging with Deep Learning*, 1336–1348. PMLR.

Yang, Z.; Wang, J.; Tang, Y.; Chen, K.; Zhao, H.; and Torr, P. H. 2022. LAVT: Language-Aware Vision Transformer for Referring Image Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18155–18165.

Yang, Z.; Wang, J.; Tang, Y.; Chen, K.; Zhao, H.; and Torr, P. H. 2023. Semantics-Aware Dynamic Localization and Refinement for Referring Image Segmentation. *Proceedings of the Conference on Artificial Intelligence*.

Ye, J.; Tian, J.; Yan, M.; Yang, X.; Wang, X.; Zhang, J.; He, L.; and Lin, X. 2022. Shifting more attention to visual backbone: Query-modulated refinement networks for end-to-end visual grounding. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 15502–15512.

Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; and Berg, T. L. 2018. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1307–1315.

Zhang, Z.; Zhu, Y.; Liu, J.; Liang, X.; and Ke, W. 2022. Coupalign: Coupling word-pixel with sentence-mask alignments for referring image segmentation. *Advances in Neural Information Processing Systems*, 35: 14729–14742.

Zhu, C.; Zhou, Y.; Shen, Y.; Luo, G.; Pan, X.; Lin, M.; Chen, C.; Cao, L.; Sun, X.; and Ji, R. 2022. Seqtr: A simple yet universal network for visual grounding. In *Proceedings of the European Conference on Computer Vision*, 598–615. Springer.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable {DETR}: Deformable Transformers for End-to-End Object Detection. In *Proceedings of the International Conference on Learning Representations*.