Supplementary Materials for Language as Queries for Referring Video Object Segmentation

A. Additional Dataset Details

Ref-Youtube-VOS [10] is a large-scale benchmark which covers 3,978 videos with ∼15K language descriptions. There are 3,471 videos with 12,913 expressions in training set and 507 videos with 2,096 expressions in validation set. According to the R-VOS competition, videos in the validation set are further split into 202 and 305 videos for the competition validation and test purpose. Since the test server is currently inaccessible, the results are reported by submitting our predictions to the validation server[1].

Ref-DAVIS17 [4] is built upon DAVIS17 [9] by providing the language description for a specific object in each video. It contains 90 videos with 1,544 expression sentences describing 205 objects in total. The dataset is split into 60 videos and 30 videos for training and validation, respectively. Since there are two annotators and each of them gives the *first-frame* and *full-video* textual description for one referred object, we report the results by averaging the scores using the official evaluation code [2].

B. Additional Implementation Details

Our model is optimized using AdamW [7] optimizer with the weight decay of 5×10^{-4}, initial learning rate of 5×10^{-5} for visual backbone and 10^{-4} for the rest. We first pretrain our model on the image referring segmentation datasets Ref-COCO [14], Ref-COCOg [14] and Ref-COCO+ [8] by setting $T = 1$ with the batch size of 2 on each GPU. The pretrain procedure runs for 12 epochs with the learning rate decays divided by 10 at epoch 8 and 10. Then, on Ref-Youtube-VOS, we finetune the model for 6 epochs with 1 video clip per GPU. The learning rate decays by 10 at the 3-th and 5-th epoch. On Ref-DAVIS17, we directly report the results using the model trained on Ref-Youtube-VOS without finetune.

For A2D-Sentences, we feed the model with the window size of 5. The model is finetuned for 6 epochs with the learning rate decays at the 3-th and 5-th epoch by a factor of 0.1. On JHMDB-Sentences, following the previous works, we evaluate the generality of our method using the model trained on A2D-Sentences without finetune.

Additionally, on the Ref-Youtube-VOS, we also adopt the joint training technique by mixing the dataset with Ref-COCO/+/g. Specifically, for each image in the Ref-COCO dataset, we augment it with $\pm 20^\circ$ to form a 5-frame pseudo video clip. The joint training takes 12 epochs with the learning rate decays at the 8-th and 10-th epoch by a factor of 0.1. We use 32 V100 GPUS for the joint training and each GPU is fed with 2 video clips. It should be noted that the text encoder is froze all the time.

C. Additional Details of Dynamic Convolution

We give the pseudo-code of dynamic convolution in Figure C1, where we take one dynamic kernel for clarification. Specifically, a linear projection is applied to transform the instance embedding into dynamic convolutional weights. Then, the mask features pass through consecutive dynamic convolutional layers with the ReLU activation function. There is no normalization or activation after the last dynamic convolutional layer, and the output channel number of last layer is 1.

```
def dynamic_convolution(mask_feats, dynamic_feats):
    # mask_feats: (B, C, H/4, W/4)
    # dynamic_feats: (B, C)

    # parameters of dynamic convs: (B, num_params)
    dynamic_params = linear(dynamic_features)
    # parse conv parameters
    # weights[l]: (out_channels, in_channels, 1, 1)
    # bias[l]: (out_channels)
    weights, bias = parse_dynamic_params(dynaic_params)

    # dynamic convolution
    n_layer = len(weights)
    x = mask_feats
    for i, (x, b) in enumerate(zip(weights, bias)):
        x = conv2d(x, w, bias=b, stride=1, padding=0)
        if i < n_layer - 1:
            x = relu(x)
    # x: (B, H/4, W/4)
    return x
```

Figure C1. Pseudo-code of dynamic convolution, we take one dynamic kernel for clarification. For multiple dynamic kernels, we use group convolution in `conv2d` for efficient implementation. `linear`: linear projection.

[1] https://competitions.codalab.org/competitions/29139

[2] https://github.com/davisvideochallenge/davis2017-evaluation

Method	Backbone	Precision					IoU		mAP
		P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	Overall	Mean	
Hu *et al*. [3]	VGG-16	63.3	35.0	8.5	0.2	0.0	54.6	52.8	17.8
Gavrilyuk *et al*. [2]	I3D	69.9	46.0	17.3	1.4	0.0	54.1	54.2	23.3
CMSA + CFSA [13]	ResNet-101	76.4	62.5	38.9	9.0	0.1	62.8	58.1	-
ACAN [11]	I3D	75.6	56.4	28.7	3.4	0.0	57.6	58.4	28.9
CMPC-V [6]	I3D	81.3	65.7	37.1	7.0	0.0	61.6	61.7	34.2
ClawCraneNet [5]	ResNet-50/101	88.0	79.6	56.6	14.7	0.2	64.4	65.6	-
MTTR ($\omega = 8$) [1]	Video-Swin-T	91.0	81.5	57.0	14.4	0.1	67.4	67.9	36.6
MTTR ($\omega = 10$) [1]	Video-Swin-T	93.9	85.2	61.6	16.6	0.1	70.1	69.8	39.2
ReferFormer† ($\omega = 6$)	Video-Swin-T	93.3	84.2	61.4	16.4	0.3	70.0	69.3	39.1
ReferFormer ($\omega = 5$)	Video-Swin-T	95.8	89.3	66.8	18.9	0.2	71.9	71.0	42.2
ReferFormer ($\omega = 5$)	Video-Swin-S	95.8	90.1	68.7	20.3	0.2	72.8	71.5	42.4
ReferFormer ($\omega = 5$)	Video-Swin-B	**96.2**	**90.2**	**70.2**	**21.0**	**0.3**	**73.0**	**71.8**	**43.7**

Table C1. Comparison with the state-of-the-art methods on JHMDB-Sentences. † means our model is trained from scratch.

Backbone	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
ResNet-50	55.6	54.8	56.5
ResNet-50*	59.4 (+3.8)	58.1 (+3.3)	60.8 (+4.3)
ResNet-101	57.3	56.1	58.4
ResNet-101*	60.3 (+3.0)	58.8 (+2.7)	61.8 (+3.4)
Swin-T	58.7	57.6	59.9
Swin-T*	61.2 (+2.5)	59.7 (+2.1)	62.6 (+2.7)
Swin-S	59.6	58.1	61.1
Swin-S*	61.3 (+1.7)	59.7 (+1.6)	63.0 (+1.9)
Swin-B	61.8	60.1	63.4
Swin-B*	63.1 (+1.3)	61.4 (+1.3)	64.8 (+1.4)
Swin-L	62.4	60.8	64.0
Swin-L*	63.3 (+0.9)	61.6 (+0.8)	65.1 (+1.1)

Table C2. Ablation study on the visual backbones. * indicates using CFBI [12] as post-process.

D. Additional Experiment Results

D.1. Experiments on JHMDB-Sentences

The experiments on the JHMDB-Sentences use the models trained on A2D-Sentences without further finetuning, the results are shown in Table C1. This is used to validate the generality of methods. ReferFormer significantly outperforms all the existing methods and achieves superior 43.7 mAP using Video-Swin-Base backbone. It is noticeable that all the methods produce low scores on P@0.9. A possible reason is that the ground-truth masks are generated from human puppets, leading to the inaccurate mask annotations.

D.2. Experiments on Visual Backbones

On Ref-Youtube-VOS, we further use a simple post-process technique to refine the object masks. Concretely, we first select a frame with the highest prediction score as the reference frame. Then, we apply the off-the-shelf mask propagation method CFBI [12] to propagate the predicted mask of this frame forward and backward to the entire video. The results with post-process under different visual backbones are shown in Table C2.

As expected, the performance of our model consistently increases by using stronger backbones. And the CFBI [12] post-process can help to further boost the performance under all backbone settings. Interestingly, we observe that the performance improvement by post-process tends to narrow when the backbone gets stronger, *e.g.*, +3.8 for ResNet-50 and +0.9 for Swin-Large when considering the $\mathcal{J}\&\mathcal{F}$ metric. This phenomenon shows that the visual encoder is essential for providing reliable reasoning on which object is described and generating the precise masks.

D.3. Experiments on Class-agnostic Training

By default, our models are trained in the class-agnostic way, *i.e.*, decide whether the object is referred or not. As described in the paper, the class head can be easily modified to predict the referred object category by simply change the class number. In this way, we train our model in a class-discriminative way and show the results in Table D3. We could observe the class-agnostic training method has clear performance gain (+2.1 $\mathcal{J}\&\mathcal{F}$) over the strong class-discriminative training results, since the binary classification is easier to optimize. The selection of training method can flexibly depend on the usage in real applications.

Class Agnostic	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
	53.9	52.8	55.0
✓	56.0	54.8	57.3

Table D3. Ablation study on the class-agnostic training.

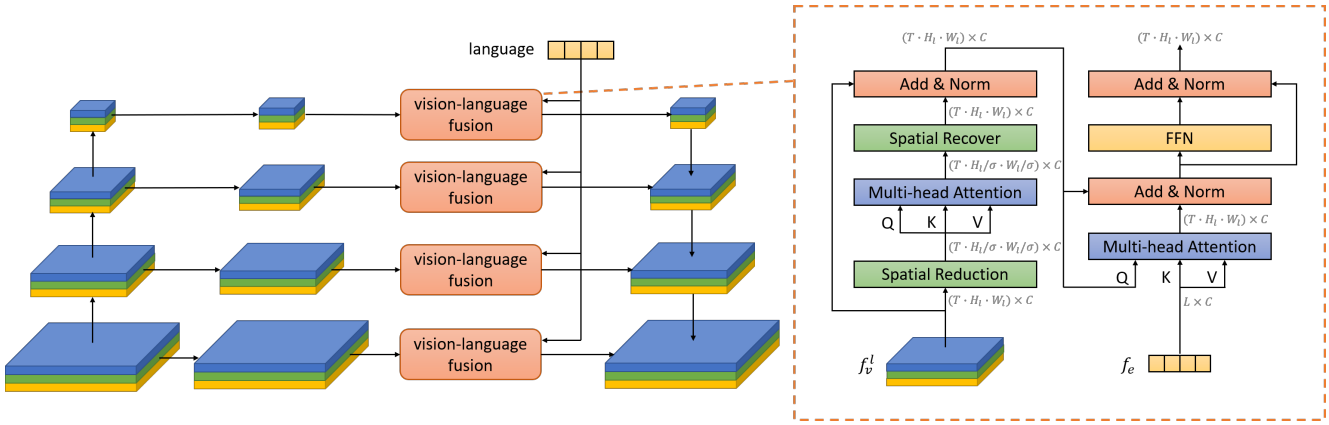

Figure C2. The architecture of cross-modal feature pyramid network (CM-FPN). Note that different colors in the feature maps represent different frames. The visual and textual features are interacted in all the levels of feature maps. The vision-language fusion process is illustrated in the dash box on the right.

E. Architecture Figure of CM-FPN

The standard FPN can already provide a high-resolution feature map with rich visual semantics, however, such feature map lacks the linguistic information and would be sub-optimal for the cross-modal task. So we design a cross-modal feature pyramid network (CM-FPN) to perform multi-scale cross-modal fusion for finer interaction, the architecture in shown in Figure C2.

F. Visualization Results

We show the visualization results of our model in Figure F3. It can be seen that ReferFormer is able to segment and track the referred object in challenging cases, *e.g.*, person pose variations, instances occlusion and instances that are partially displayed or completely disappeared in the camera.

References

[1] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. *arXiv preprint arXiv:2111.14821*, 2021. 2

[2] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5958–5966, 2018. 2

[3] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016. 2

[4] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision*, pages 123–141. Springer, 2018. 1

[5] Chen Liang, Yu Wu, Yawei Luo, and Yi Yang. Clawcranenet: Leveraging object-level relation for text-based video segmentation. *arXiv preprint arXiv:2103.10702*, 2021. 2

[6] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[8] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 1

[9] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 1

[10] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 208–223. Springer, 2020. 1

[11] Hao Wang, Cheng Deng, Junchi Yan, and Dacheng Tao. Asymmetric cross-guided attention network for actor and action video segmentation from natural language query. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3939–3948, 2019. 2

[12] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *European Conference on Computer Vision*, pages 332–348. Springer, 2020. 2

[13] Linwei Ye, Mrigank Rochan, Zhi Liu, Xiaoqin Zhang, and Yang Wang. Referring segmentation in images and videos with cross-modal self-attention network. *arXiv preprint arXiv:2102.04762*, 2021. 2

Figure F3. Visualization results on (a) Ref-DAVIS17 and (b) Ref-Youtube-VOS. Our unified framework is able to detect, segment and track the referred object simultaneously.

[14] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016. 1