Meta Compositional Referring Expression Segmentation

Li Xu[1], Mark He Huang[1,3], Xindi Shang[2], Zehuan Yuan[2], Ying Sun[3], Jun Liu [1*]
[1]Singapore University of Technology and Design, Singapore
[2]ByteDance
[3]Institute for Infocomm Research (I^{2}R) & Centre for Frontier AI Research (CFAR), A*STAR, Singapore
{li_xu, he_huang}@mymail.sutd.edu.sg
{shangxindi, yuanzehuan}@bytedance.com
suny@i2r.a-star.edu.sg, jun_liu@sutd.edu.sg

Abstract

Referring expression segmentation aims to segment an object described by a language expression from an image. Despite the recent progress on this task, existing models tackling this task may not be able to fully capture semantics and visual representations of individual concepts, which limits their generalization capability, especially when handling ***novel compositions of learned concepts****. In this work, through the lens of meta learning, we propose a Meta Compositional Referring Expression Segmentation (****MCRES****) framework to enhance model compositional generalization performance. Specifically, to handle various levels of novel compositions, our framework first uses training data to construct a virtual training set and multiple virtual testing sets, where data samples in each virtual testing set contain a level of novel compositions w.r.t. the virtual training set. Then, following a novel meta optimization scheme to optimize the model to obtain good testing performance on the virtual testing sets after training on the virtual training set, our framework can effectively drive the model to better capture semantics and visual representations of individual concepts, and thus obtain robust generalization performance even when handling novel compositions. Extensive experiments on three benchmark datasets demonstrate the effectiveness of our framework.*

1. Introduction

Referring expression segmentation (RES) [12, 38, 40] aims to segment a visual entity in an image given a linguistic expression. This task has been receiving increasing attention in recent years [5, 18, 34, 37], as it can play an important role in various applications, such as language-based human-robot interaction and interactive image editing. However, despite the recent progress on tackling this task [5, 34, 42], existing methods may struggle with handling the testing samples, of which the expressions contain novel compositions of learned concepts. Here a novel composition means that the composition itself does not exist in the training data, but its individual components (e.g., words, phrases) exist in the training data, as shown in Fig. 1a.

We observe that testing samples containing such novel compositions of learned concepts widely exist in RES datasets [26, 28, 41]. However, existing RES models may not be able to well handle novel compositions during testing. Here we test the generalization capability of multiple state-of-the-art models [25, 37, 42] in terms of handling

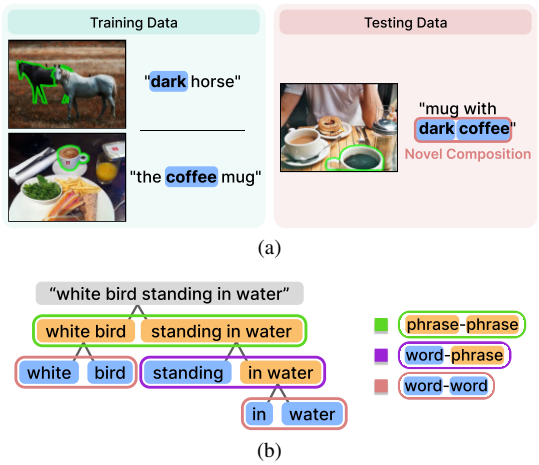


Figure 1. **Illustration of novel compositions and various levels of compositions.** (a) An example of the testing sample containing the novel composition of "dark coffee" in RefCOCO dataset [41]. Such a novel composition itself does not exist in the training data, but its individual components (i.e., "dark" and "coffee") exist in the training data. (b) An example of various levels of compositions in an expression. We perform constituency parsing of the expression using AllenNLP [9]. Based on the obtained parsing tree as shown above, we can then obtain various levels of compositions (e.g., word-word level, word-phrase level) in this expression.

*Corresponding Author

novel compositions, as shown in Table 2. Specifically, we first split each testing set of the RefCOCO dataset. In each testing set, one split subset includes the data samples, in which all the contained compositions are seen in the RefCOCO training set. While another subset includes the data samples containing novel compositions, of which the individual components (e.g., words, phrases) exist in the RefCOCO training set but the composition itself is unseen in the training set, i.e., containing novel compositions of learned concepts. Then we evaluate the models [25, 37, 42] on the two subsets in each testing set, and find that for each model, its testing performance on the subset containing novel compositions drops obviously compared to the performance on the other subset. For these models, the performance gap between the two subsets can reach $14\%-17\%$ measured by the metric of overall IoU. Such a clear performance gap indicates that existing models struggle with generalizing to novel compositions of learned concepts. This might due to that the model does not effectively capture the semantics and visual representations of individual concepts (e.g., "dark", "coffee" in Fig. 1a) during training. Then the trained model may fail to recognize a novel composition (e.g., "dark coffee") at testing time, which though is composed of learned concepts.

Thus to handle this issue, we aim to train the model to effectively capture the semantics and visual representations of individual concepts during training. Despite the conceptual simplicity, how to guide the model's learning behavior towards this goal is a challenging problem. Here from the perspective of meta learning, we propose a *Meta Compositional Referring Expression Segmentation (**MCRES**)* framework, to effectively handle such a challenging problem by only changing the model training scheme.

Meta learning proposes to perform *virtual testing* during model training for better performance [7, 29]. Inspired by this, to improve the generalization capability of RES models, our MCRES framework incorporates a *meta optimization* scheme that consists of three steps: *virtual training*, *virtual testing* and *meta update*. Specifically, we first split the training set to construct a virtual training set for virtual training, and a virtual testing set for virtual testing. The data samples in the virtual testing set contain novel compositions w.r.t. the virtual training set. For example, if the expressions of data samples in the virtual training set contain both words "dark" and "coffee" but do not contain their composition (i.e., "dark coffee"), the virtual testing set can include this novel composition correspondingly.

Based on the constructed virtual training set and virtual testing set, we first train the model using the virtual training set, and then evaluate the trained model on the virtual testing set. During virtual training, the model may learn the compositions of individual concepts as a whole without truly understanding the semantics and visual representations of individual concepts, which though can still improve model training performance. For example, if there are many training samples containing the composition of "yellow banana" in the virtual training set, the model can superficially correlate "banana" with "yellow" and learn this composition as a whole, since using such spurious correlations can facilitate the model learning [1, 10, 39]. However, learning the compositions as a whole over the virtual training set may not improve model performance much on the virtual testing set in virtual testing, since the virtual testing set contains novel compositions w.r.t. the virtual training set. Thus to achieve good testing performance on such a virtual testing set, the model needs to effectively capture semantics and visual representations of individual concepts during virtual training. In this way, the model testing performance on the virtual testing set serves as a generalization feedback to the model virtual training process.

Thus after the virtual training and virtual testing, we can further update the model to obtain better testing performance on the virtual testing set (i.e., meta update), so as to drive the model training on the virtual training set towards the direction of learning to capture semantics and visual representations of individual concepts, i.e., learning to learn. In this manner, our framework is able to optimize the model for robust generalization performance, even tackling the challenging testing samples with novel compositions.

Moreover, given that expressions can often be hierarchically decomposed, there can exist various levels of novel compositions. Specifically, to identify meaningful compositions in an expression, we can parse an expression into a tree structure based on the constituency parsing tool [9] as shown in Fig.1b. In such a parsing tree, under the same parent node, each pair of child nodes (e.g., "white" and "bird") are closely semantically related, and thus can form a meaningful composition. Since each child node can be a word or a phrase as in Fig. 1b, there can naturally exist the following three levels of novel compositions: word-word level (e.g., "white" and "bird"), word-phrase level (e.g., "standing" and "in water") and phrase-phrase level (e.g., "white bird" and "standing in water"), which correspond to different levels of comprehension complexity. To better handle such a range of novel compositions, we construct multiple virtual testing sets in our framework, where each virtual testing set is constructed to handle one level of novel compositions.

Our framework only changes the model training scheme without the need to change the model structure. Thus our framework is general, and can be conveniently applied on various RES models. We test our framework on multiple models, and obtain consistent performance improvement.

The contributions of our work are threefold: 1) We propose a novel framework (MCRES) to effectively improve generalization performance of RES models, especially when handing novel compositions of learned con-

cepts. 2) Via constructing a virtual training set and multiple virtual testing sets w.r.t. various levels of novel compositions, our framework can train the model to well handle various levels of novel compositions. 3) When applied on various models on three RES benchmarks [28, 41], our framework achieves consistent performance improvement.

2. Related Works

Referring Expression Segmentation (RES). RES [12] aims to segment a target object from an image based on an expression. Early works [20, 22, 28, 41] employed convolutional and recurrent networks to extract visual and linguistic features respectively, and then fused the extracted features to predict the segmentation mask. Recently, a series of works [5, 18, 37] employed vision and language transformers to boost performance. Besides, with the development of large-scale pretrained models, Wang et al. [34] proposed to leverage CLIP [30] to improve cross-modal matching.

Some methods have been explored to help model better understand learned concepts in RES. Yu et al. [40] proposed a modular network that uses different modules to process different types of information in the given expression. Yang et al. [36] designed a reasoning module to help align the language concepts with visual regions. Different from all the above-mentioned works, we propose an MCRES framework to drive RES models to better capture semantics and visual representations of individual concepts. Such a framework only changes the model training scheme, and thus can be flexibly applied on various models to improve their generalization performance.

Meta Learning. Meta learning, i.e., the paradigm of learning to learn, has emerged to mainly tackle the few-shot learning problem [7, 29, 31, 32]. MAML [7] and its following works [29, 31] aim to learn a good initialization of network parameters, to achieve fast test-time update to adapt to new few-shot learning tasks. More recently, meta learning has also been explored in other areas [8, 11, 14, 19, 35] to enhance model generalization performance without the need of test-time update. Inspired by these works, we leverage a meta learning-based framework to improve generalization performance of RES models especially when handling novel compositions.

3. Method

Existing RES models may fail to generalize to data samples containing novel compositions of learned concepts. To handle this issue, we aim to encourage the model to better capture the semantics of individual concepts as well as recognize their visual correspondences during training, which however is a non-trivial problem. In this paper, from the perspective of meta learning, we introduce MCRES framework optimizing RES models via a meta optimization scheme to improve their generalization performance.

Specifically, as shown in Fig. 2, in our framework, we first split the original training set ($\mathcal{D}_{train}$) to build a virtual training set ($\mathcal{D}_{v\_tr}$) for virtual training, and a group of K virtual testing sets ($\{\mathcal{D}^k_{v\_te}\}^K_{k=1}$) for virtual testing. Each virtual testing set consists of data samples containing one level of novel compositions w.r.t. the virtual training set. We first use the virtual training set to train the model (virtual training), and then perform model testing on the virtual testing sets (virtual testing). Since the virtual testing sets contain novel compositions of individual concepts from the virtual training set, if the model can still achieve robust testing performance on the virtual testing sets after training on the virtual training set, we posit that the trained model has learned to capture more semantics and visual representations of individual concepts during the virtual training. Guided by this principle, we can optimize the model virtual testing performance, to guide the model virtual training process towards learning more semantics and representations of individual concepts. Below, we first introduce the pipeline of our framework, and then detail the virtual training set and virtual testing sets construction process.

3.1. Framework

We first train the RES model using the virtual training set, i.e., virtual training. Specifically, we denote the model parameters as θ, the loss function (e.g., cross-entropy loss) for training the RES model as $\mathcal{L}$. Thus we can calculate the model virtual training loss ($\mathcal{L}_{v\_tr}$) over the virtual training set ($\mathcal{D}_{v\_tr}$) as:

$$\mathcal{L}_{v\_tr}(\theta) = \mathcal{L}(\theta; \mathcal{D}_{v\_tr}) \tag{1}$$

Based on this loss, we then update the model parameters (θ) as follows:

$$\theta' = \theta - \alpha \nabla_\theta \mathcal{L}_{v\_tr}(\theta) \tag{2}$$

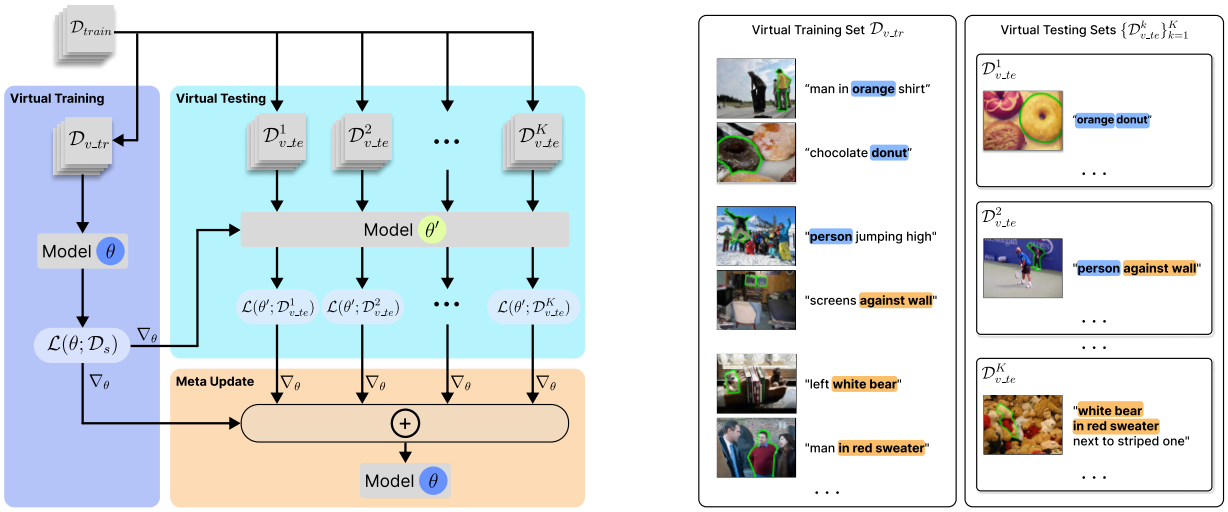
where α denotes the learning rate. It is worth noting that the model update at this step is *virtual*, and the updated parameters θ' only serve as intermediate parameters for the model evaluation at the following virtual testing step.

After the virtual training, we evaluate the generalization performance of the trained model on the virtual testing sets ($\{\mathcal{D}^k_{v\_te}\}^K_{k=1}$), i.e., virtual testing. Specifically, on each virtual testing set $\mathcal{D}^k_{v\_te}$, we compute the model loss $\mathcal{L}^k_{v\_te}$ as:

$$\mathcal{L}^k_{v\_te}(\theta') = \mathcal{L}(\theta'; \mathcal{D}^k_{v\_te}) \tag{3}$$

Such a loss measures how well the model generalizes to the virtual testing set after training on the virtual training set.

Then we perform the meta update step. At this step, we wish to *actually* update the model parameters (θ), so that after the model training on the virtual training set, the trained model can generalize well to the virtual testing sets, i.e., generalizing well to novel compositions. To this end, we



(a) (b)

Figure 2. **Framework overview.** Fig. (a) illustrates our framework pipeline. First, we train the model using the virtual training set ($\mathcal{D}_{v\_tr}$), and then obtain the updated model. We then test the model with updated parameters (θ') on multiple virtual testing sets ($\{\mathcal{D}^k_{v\_te}\}^K_{k=1}$). According to the virtual testing losses, we perform meta update to optimize the model for better generalization capability. Fig. (b) shows that we construct a virtual training set and multiple virtual testing sets to handle various levels of novel compositions. The expressions of data samples in each virtual testing set contain a level of novel compositions w.r.t. the virtual training set.

formulate the optimization objective as:

$$\begin{aligned} &\min_{\theta} \ \mathcal{L}_{v\_tr}(\theta) + \sum_{k=1}^{K} \mathcal{L}^k_{v\_te}(\theta') \\ =&\min_{\theta} \ \mathcal{L}_{v\_tr}(\theta) + \sum_{k=1}^{K} \mathcal{L}^k_{v\_te}\big(\theta - \alpha\nabla_\theta \mathcal{L}_{v\_tr}(\theta)\big) \end{aligned} \tag{4}$$

where the first term indicates the model training performance on the virtual training set, and the second term represents the model testing performance on the virtual testing sets after training on the virtual training set. Based on this objective, we then update the model parameters (θ) as:

$$\theta \leftarrow \theta - \beta\nabla_\theta\Big(\mathcal{L}_{v\_tr}(\theta) + \sum_{k=1}^{K} \mathcal{L}^k_{v\_te}\big(\theta - \alpha\nabla_\theta \mathcal{L}_{v\_tr}(\theta)\big)\Big) \tag{5}$$

where β is the learning rate for meta update. Through the above optimization process, the model is pushed to learn more semantics and visual representations of individual concepts during training.

During the above process, the model is first trained (updated) on the virtual training set. At this step, the model may learn the compositions of individual concepts as a whole without effectively capturing the semantics and visual representations of individual concepts, which though can still improve its training performance on the virtual training set. However, to achieve good testing performance on the virtual testing data which contains novel compositions of individual concepts from the virtual training data, the model is expected to avoid learning compositions as a whole and instead capture more semantics and representations of individual concepts. In this way, the second term of Eqn. 5, which includes the second-order gradients of θ: $\nabla_\theta \mathcal{L}^k_{v\_te}\big(\theta - \alpha\nabla_\theta \mathcal{L}_{v\_tr}(\theta)\big)$, can be regarded as a generalization feedback that can guide the model to capture more semantics and representations of individual concepts.

In our framework, the above three steps (i.e., virtual training, virtual testing and meta update) are performed iteratively until the model training converges.

3.2. Sets Construction

In our framework, to handle various levels of novel compositions, we split the original training set to construct a virtual training set and various virtual testing sets. Each virtual testing set is expected to contain one level of novel compositions w.r.t. the virtual training set. Below we first introduce the details of these sets, and then discuss the strategy for constructing the virtual testing sets.

We randomly sample a subset of the training set ($\mathcal{D}_{train}$) as the virtual training set ($\mathcal{D}_{v\_tr}$), and the remaining training data will be used as the candidate samples to construct a group of (K) virtual testing sets ($\{\mathcal{D}^k_{v\_te}\}^K_{k=1}$). Each virtual testing set is constructed to handle one level of novel compositions. Here considering the hierarchical semantic structure of each expression, we target the following levels of novel compositions: word-word level, word-phrase level and phrase-phrase level. A phrase means a group of words that serve as a grammatical unit in the expression (e.g., "white bird" in Fig. 1b), which can be conveniently identified using off-the-shelf tools (e.g., AllenNLP [9]).

Specifically, to identify various levels of novel compo-

sitions in an expression, we first use the constituency parsing tool [9] to parse an expression into a tree structure as shown in Fig. 1b. In this tree, under the same parent node, each pair of child nodes are closely related, and thus can form a meaningful composition. If a pair of nodes having the same parent node are both words, they form a word-word level composition (e.g., "white" and "bird"). Similarly, if the nodes with the same parent node are a word and a phrase, they form a word-phrase level composition (e.g., "standing" and "in water"). In a similar way, we can obtain phrase-phrase level compositions (e.g., "white bird" and "standing in water"). Thus there can exist three levels of novel compositions, which correspond to different levels of comprehension complexity. Corresponding to these three levels of novel compositions, we will construct a total of three virtual testing sets ($K = 3$).

Virtual testing sets construction. To construct each virtual testing set, we need to select data samples containing the corresponding level of novel compositions w.r.t. the virtual training set, from the candidate samples. A data sample means an expression paired with the corresponding image. Here we design an efficient strategy, which can be leveraged to construct each virtual testing set. Below, we take the process of constructing the virtual testing set for handling *word-word* level novel compositions as an example, to introduce such a strategy. Specifically, this virtual testing set should include all of those candidate samples that contain word-word level novel compositions w.r.t. the virtual training set. To construct such a virtual testing set, our strategy proceeds as follows.

(i) To find the data samples, of which the expressions contain word-word level novel compositions w.r.t. the virtual training set, from the candidate samples, we need to first obtain all the word-word level compositions in the virtual training set and in the candidate samples respectively. Thus by parsing each expression into a parsing tree as shown in Fig. 1b, we can obtain all word-word level compositions (e.g., "white" and "bird", "in" and "water" in Fig. 1b) in each expression. (ii) To identify word-word level novel compositions, we first select the word-word level compositions that exist in the candidate samples but are unseen in the virtual training set. Then for each selected composition, we further check if its individual words exist in the virtual training set. If so, such a composition will be identified as a word-word level novel composition w.r.t. the virtual training set. For example, if the composition "white bird" is unseen in the virtual training set, while its individual words "white" and "bird" both exist in the virtual training set, this composition is a word-word level novel composition w.r.t. the virtual training set. (iii) Finally, we select the candidate samples, of which the expressions contain the identified word-word level novel compositions w.r.t. the virtual training set, to construct the virtual testing set.

To efficiently perform the above steps, we can employ parallel matrix operations. Specifically, to record the word-word level compositions in the virtual training set, we build a matrix $\mathcal{M}_{v\_tr}$ with the shape of $|\mathcal{V}| \times |\mathcal{V}|$, where $|\mathcal{V}|$ is the size of the vocabulary set $\mathcal{V}$ of the original training set. In this matrix, each element at the $i-th$ row and the $j-th$ column ($i, j \in \{1, \ldots, |\mathcal{V}|\}$) is a binary value, and the value 1 indicates that the $i-th$ word and $j-th$ word in the vocabulary set $\mathcal{V}$ form a word-word level composition in the virtual training set, while 0 means such a word-word level composition does not exist in the virtual training set. Similarly, we can also build a matrix $\mathcal{M}_{candi}$ to record the word-word level compositions in the candidate samples. Then to identify word-word level novel compositions w.r.t. the virtual training set ($\mathcal{M}_{v\_tr}$) from the candidate samples ($\mathcal{M}_{candi}$), we can efficiently compute a difference matrix: $\mathcal{M}_{diff} = \mathcal{M}_{candi} - \mathcal{M}_{v\_tr}$. In $\mathcal{M}_{diff}$, any element with the value 1 indicates that the corresponding composition exists in the candidate samples, but is unseen in the virtual training set. Then by further checking whether the individual words in such a composition exist in the virtual training set, we can determine if it is a word-word level novel composition w.r.t. the virtual training set.

Similarly, we can efficiently construct the other two virtual testing sets. Note that to help the model to learn to handle a wide range of possible novel compositions, at the beginning of each training epoch, we randomly sample a subset of the training data to re-construct the virtual training set, and leverage the remaining data to re-construct the virtual testing sets using the above strategy.

As discussed above, to handle various levels of novel compositions, we construct multiple virtual testing sets. A simpler alternative is to construct only one virtual testing set, which includes all the data samples containing any level of novel compositions w.r.t. the virtual training set. However, compared to this alternative, by constructing multiple virtual testing sets, each level of novel compositions in the corresponding virtual testing set can offer an explicit generalization feedback during model training. Thus the model is explicitly encouraged to well handle each level of novel compositions. Moreover, constructing multiple virtual testing sets can facilitate the use of curriculum learning strategy for model training as discussed in Sec. 3.3.

3.3. Training and Testing

To help models generalize to novel compositions in RES, our framework only changes the model training scheme. Thus our framework is general, and can be flexibly applied to train various RES models. During training, at each epoch, we first split the training set to construct a virtual training set and multiple virtual testing sets. Then we iteratively optimize the model over the virtual training set and virtual testing sets via meta optimization. Specifically, for each meta

Table 1. Comparison with the state-of-the-arts on three benchmark datasets using the metric of overall IoU. Moreover, we apply our framework on various models [25, 37, 42], and obtain consistent performance improvement. "-" indicates that the corresponding result is not provided in the original paper. U means the UMD split of RefCOCOg dataset, and G means the Google split.

Method	RefCOCO			RefCOCO+			RefCOCOg		
	val	testA	testB	val	testA	testB	val(U)	test (U)	val (G)
DMN [27]	49.78	54.83	45.13	38.88	44.22	32.29	-	-	36.76
RRN [20]	55.33	57.26	53.93	39.75	42.15	36.11	-	-	36.45
MAttNet [40]	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	-
CAC [4]	58.90	61.77	53.81	-	-	-	46.37	46.95	44.32
CMSA [38]	58.32	60.61	55.09	43.76	47.60	37.89	-	-	39.98
STEP [3]	60.04	63.46	57.97	48.19	52.33	404.1	-	-	46.40
BRINet [13]	60.98	62.99	59.21	48.17	52.32	42.11	-	-	48.04
CMPC [15]	61.36	64.53	59.64	49.56	53.44	43.23	-	-	49.05
LSCM [16]	61.47	64.99	59.55	49.34	53.12	43.50	-	-	48.05
CMPC+ [23]	62.47	65.08	60.82	50.25	54.04	43.47	-	-	49.89
EFN [6]	62.76	65.69	59.67	51.50	55.24	43.01	-	-	51.93
BUSNet [36]	63.27	66.41	61.39	51.76	56.87	44.13	-	-	50.56
CGAN [24]	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	46.54
LTS [17]	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
VLT [5]	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65	49.76
ReSTR [18]	67.22	69.30	64.45	55.78	60.44	48.27	-	-	54.48
CRIS [34]	70.47	73.18	66.10	62.27	68.08	53.68	59.87	60.36	-
MCN [25]	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
MCN + Ours	**64.51**	**66.48**	**62.84**	**52.29**	**56.71**	**46.72**	**51.49**	**51.63**	-
SeqTR [42]	71.70	73.31	69.82	63.04	66.73	58.97	64.69	65.74	-
SeqTR + Ours	**73.23**	**75.01**	**71.95**	**64.71**	**67.85**	**60.85**	**66.77**	**67.48**	-
LAVT [37]	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09	60.50
LAVT + Ours	**74.92**	**76.98**	**70.84**	**64.32**	**69.68**	**56.64**	**63.51**	**64.90**	**61.63**

optimization iteration, we use two batches of data samples: one batch of data for virtual training and the other batch of data for virtual testing. Thus we need to ensure that the batch of data for virtual testing consists of the data samples containing novel compositions w.r.t. the batch of data for virtual training. To this end, for each iteration, we randomly sample some data from each virtual testing set, to form the batch of data for virtual testing. Then we select the virtual training set samples that contain all the individual components of the novel compositions in the batch of data for virtual testing, to form the batch of data for virtual training. Note that the above data preparation procedures can be done before the model training. For model testing, we test the trained model in the conventional way.

Besides, given that the various levels of novel compositions correspond to different levels of comprehension complexity, we adopt a *curriculum learning* strategy [2, 33] for model training, so that the model can progressively learn to handle various levels of novel compositions, i.e., from lower level (word-word) to middle level (word-phrase) to higher level (phrase-phrase), and thus can learn all these levels of novel compositions better. Specifically, in the first $1/3$ of training epochs, we only use the virtual testing set for handling word-word level novel compositions for meta optimization. Then in the middle $1/3 - 2/3$ of training epochs, we add the virtual testing set for handling word-phrase level novel compositions. Finally, in the remaining training epochs, we use all the three virtual testing sets.

Table 2. We evaluate our framework on different testing subsets w.r.t. novel compositions to validate its effectiveness on optimizing the model to better generalize to novel compositions. We use the metric of overall IoU here. The performance gain of our framework compared to the corresponding baseline model is shown in parentheses.

Method	RefCOCO-val		RefCOCO-testA		RefCOCO-testB	
	Novel	Non-novel	Novel	Non-novel	Novel	Non-novel
MCN [23]	53.17	67.41	55.43	70.08	50.64	66.47
MCN + Ours	**57.38 (↑4.21)**	**68.09 (↑0.68)**	**59.66 (↑4.23)**	**70.67 (↑0.59)**	**55.87 (↑5.23)**	**67.04 (↑0.57)**
SeqTR [40]	64.24	78.59	65.71	79.46	60.14	77.21
SeqTR + Ours	**67.31 (↑3.07)**	**79.06 (↑0.47)**	**69.48 (↑3.77)**	**79.84 (↑0.38)**	**64.89 (↑4.75)**	**77.68 (↑0.47)**
LAVT [35]	63.52	78.55	67.17	80.71	58.49	76.23
LAVT + Ours	**67.42 (↑3.90)**	**79.29 (↑0.74)**	**70.05 (↑2.88)**	**81.37 (↑0.66)**	**62.63 (↑4.14)**	**76.92 (↑0.69)**

4. Experiments

We evaluate our method on three commonly used RES benchmarks: RefCOCO [41], RefCOCO+ [41] and RefCOCOg [26, 28]. Images in these three benchmarks are all from the COCO dataset [21]. RefCOCO [41] contains 19994 images, 50000 annotated objects with 142209 referring expressions. RefCOCO+ [41] consists of 19992 images with 49856 annotated objects, and 141564 expressions. Different from RefCOCO, words describing absolute spatial locations (e.g., left, front) are not allowed to be used in the expressions in RefCOCO+. In RefCOCO and RefCOCO+ datasets, following the original split in [41], the visual entities to be segmented in testA subset are people, while the ones in testB subset are objects (i.e., not people). RefCOCOg [26, 28] includes 26711 images, 54822 annotated objects and 104560 expressions. Compared with RefCOCO and RefCOCO+, the expressions in RefCOCOg are more complex, which have an average length of 8.4 words. There exist two different partitions for RefCOCOg dataset: UMD split [28] and Google split [26].

Evaluation metrics. Following [3, 37, 38], we report our results using two kinds of metrics: overall IoU (oIoU) and Precision@X (P@X). The overall IoU measures the ratio of total intersection regions over total union regions of predicted masks and ground truths of all testing samples. Precision@X calculates the percentage of testing samples, of which the model prediction has an IoU score higher than the threshold value X, and X$\in \{0.5, 0.6, 0.7, 0.8, 0.9\}$.

Implementation details. We conduct our experiments on 8 Tesla V100 GPUs. We applied our framework on various RES models [25, 37, 42]. On each dataset, we randomly sample 60% of the training data as the virtual training set and use the remaining training data to construct virtual testing sets at each training epoch. The learning rate (α) for virtual training is 5e-5, and the learning rate (β) for meta update is 2e-5.

4.1. Experimental Results

As shown in Table 1, our framework achieves state-of-the-art performance across all three datasets, demonstrating the superiority of our framework. Moreover, we applied our

Table 3. We test several variants to investigate the impact of each level of novel compositions on model performance.

Method	oIoU	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9
Baseline (LAVT)	72.73	84.46	81.24	75.28	64.71	34.30
w/o word-word novel compositions	73.43	84.71	81.69	76.12	65.23	34.47
w/o word-phrase novel compositions	73.68	84.82	81.76	76.28	65.37	34.56
w/o phrase-phrase novel compositions	73.59	84.73	81.73	76.16	65.35	34.45
Ours	**74.92**	**86.23**	**83.45**	**77.25**	**66.56**	**35.61**

Table 4. We evaluate a variant to test the effectiveness of meta optimization scheme in our framework. For this variant, its optimization objective is to minimize $\mathcal{L}_{v\_tr}(\theta) + \sum_{k=1}^{K} \mathcal{L}_{v\_te}^{k}(\theta)$ (i.e., replacing θ' with θ in Eqn. 4).

Method	oIoU	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9
Training w/o meta	72.76	84.50	81.27	75.33	64.72	34.31
Ours	**74.92**	**86.23**	**83.45**	**77.25**	**66.56**	**35.61**

framework on two transformer-based SOTA models [37,42] and a CNN-based model [25]. Our framework brings consistent performance improvement on all these models and datasets. This shows that our framework can serve as a general approach to enhance model performance

To further validate the effectiveness of our framework for handling novel compositions, we perform the following analysis. Specifically, we split each testing set of RefCOCO dataset to construct two subsets. One subset (*Non-novel*) includes the data samples, in which all the contained compositions are seen in the RefCOCO training set. While another subset (*Novel*) includes the data samples that contain any level of novel compositions w.r.t. the training set. As shown in Table 2, on each testing set, we can see a clear performance gap between the two subsets for each of the baseline models [25, 37, 42], showing that existing models struggle with handling novel compositions. Then by applying our framework on each baseline model, we obtain significant performance improvement on the subset of *Novel*. This validates the general effectiveness of our framework to optimize the model to well generalize to novel compositions. Moreover, our framework also slightly improves the model performance on the subset of *Non-novel*. This can be attributed to that by training the model to better capture semantics and visual representations of individual concepts, our framework can help the model to better understand the given expression and find its visual correspondence, which thus can generally improve model performance. We also show some qualitative results in Fig. 3. As shown, when handling the testing samples containing novel compositions, our framework achieves better performance than the baseline model [37].

4.2. Ablation Studies

Following [3, 37, 38], we conduct ablation experiments to evaluate our framework on RefCOCO validation set.

Impact of various levels of novel compositions. To investigate the impact of each level of novel compositions on model performance, we test multiple variants. One variant (*w/o word-word novel compositions*) ignores the word-word level novel compositions. This means that in this variant, we omit the virtual testing set for handling word-word level novel compositions during meta optimization. Similarly, the other variants (*w/o word-phrase novel compositions* and *w/o phrase-phrase novel compositions*) ignore the corresponding level of novel compositions respectively. As shown in Table 3, ignoring any level of novel compositions leads to performance drop compared to our framework, demonstrating that each level of novel compositions can affect model generalization performance.

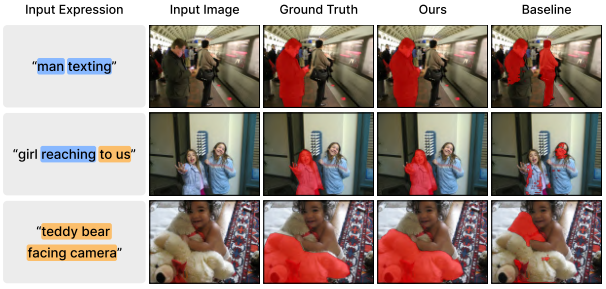


Figure 3. **Qualitative results of our method and the baseline model [37].** The above testing examples contain a word-word novel composition, a word-phrase novel composition, and a phrase-phrase novel composition respectively. As shown, by applying our framework on the baseline model, our method performs better when handling novel compositions of the learned concepts.

Impact of meta optimization. To investigate the effectiveness of the meta optimization scheme in our framework, we evaluate a variant (*training w/o meta*). In this variant, we train the model in the conventional manner on the constructed virtual training set and virtual testing sets, i.e., without meta optimization. Note that for this variant, we still construct the virtual training set and virtual testing sets in the same way as in our framework. As shown in Table 4, our framework outperforms this variant obviously, demonstrating the effectiveness of our meta optimization scheme.

Impact of multiple virtual testing sets and curriculum learning. In our framework, we construct multiple virtual testing sets to handle various levels of novel compositions. Moreover, since the various levels of novel compositions correspond to different levels of comprehension complexity, we adopt a curriculum learning strategy to facilitate model training. To investigate the effectiveness of such design, we evaluate two variants. One variant (*one virtual testing set*) constructs only one virtual testing set to handle all levels of novel compositions. Such a virtual testing set includes all the data samples containing any level of novel compositions w.r.t. the virtual training set. Another variant (*multiple virtual testing sets w/o curriculum learning*) constructs multiple virtual testing sets as in our framework, but does not adopt the curriculum learning strategy. We compare these two variants to our original framework (*multiple virtual testing sets w/ curriculum learning*).

As shown in Table 5, compared to the variant construct-

ing one virtual testing set, the other variant obtains better performance, showing the superiority of using multiple virtual testing sets to handle various levels of novel compositions. Furthermore, by employing the curriculum learning strategy, our framework performs better than the variant constructing multiple virtual testing sets without curriculum learning, demonstrating the effectiveness of our curriculum learning strategy.

Impact of the size of virtual training set. In our framework, we randomly sample 60% of the training data as the virtual training set, and use the remaining 40% of the training data to construct virtual testing sets (*60%:40%*). Here we test two variants. One variant (*50%:50%*) uses 50% of the training data to construct the virtual training set, and the remaining 50% to construct virtual testing sets. While another variant (*70%:30%*) uses 70% of the training data for the virtual training set, and the remaining 30% part for virtual testing sets. As shown in Table 6, our method and these two variants all perform better than the baseline model (i.e., LAVT [37]), showing the robustness of our framework in terms of the varying size of virtual training set.

Impact of virtual testing sets construction strategy. In our framework, after sampling a subset of the training set as the virtual training set, we use the remaining training data to construct multiple virtual testing sets. Each virtual testing set consists of the data samples containing one level of novel compositions w.r.t. the virtual training set. To explore the efficacy of such a strategy, we evaluate a variant (*random virtual testing sets*), in which we totally *randomly* select training data to construct each virtual testing set. As shown in Table 7, our framework obviously outperforms this variant. This shows that our virtual testing sets construct strategy can effectively help our framework to improve model generalization performance.

Training time. As shown in Table 8, we test the training time of our framework that trains the baseline network [37] with meta optimization, and compare it to the training time of the baseline that trains the same network in the conventional manner without meta optimization, on RefCOCO dataset. Though our framework performs better, it brings only relatively little increase (18.18%) in training time.

Impact of additional gradient updates. As discussed above, compared to the baseline model, our framework trains the model for longer time and involves more gradient updates. To explore whether the performance improvement of our framework comes from the additional gradient updates, we test a variant (*baseline w/ additional gradient updates*) in which we train the baseline model (following the original training strategy) for as many iterations as in our framework. As shown in Table 9, the performance of this variant is very close to the baseline model [37], and is obviously worse than our framework. This might be because that the originally trained baseline models have already reached convergence under the original training strategy, and thus additional gradient updates would not bring obvious benefits. Such results further validate the effectiveness of our framework.

Table 5. We evaluate two variants to test the impact of using multiple virtual testing sets and the curriculum learning strategy.

Method	oIoU	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9
Ours (one virtual testing set)	73.64	84.82	82.08	76.24	65.47	34.64
Ours (multiple virtual testing sets w/o curriculum learning)	74.02	85.34	82.63	76.87	66.09	34.95
Ours (multiple virtual testing sets w/ curriculum learning)	**74.92**	**86.23**	**83.45**	**77.25**	**66.56**	**35.61**

Table 6. We test different variants that utilize different proportions of the training data to construct the virtual training set and virtual testing sets.

Method	oIoU	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9
Baseline (LAVT)	72.73	84.46	81.24	75.28	64.71	34.30
Ours (50%:50%)	74.78	86.11	83.32	77.08	66.38	35.54
Ours (60%:40%)	74.92	86.23	83.45	77.25	66.56	35.61
Ours (70%:30%)	74.74	86.04	83.26	77.01	66.29	35.52

Table 7. We evaluate a variant to investigate the efficacy of virtual testing sets construction strategy in our framework.

Method	oIoU	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9
Random virtual testing sets	72.81	84.56	81.28	75.34	64.74	34.32
Ours	**74.92**	**86.23**	**83.45**	**77.25**	**66.56**	**35.61**

Table 8. Comparison of the training time. Note that our method achieves much better performance than the baseline.

Method	Training time	oIoU	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9
Baseline	33 hours	72.73	84.46	81.24	75.28	64.71	34.30
Ours	39 hours	74.92	86.23	83.45	77.25	66.56	35.61

Table 9. We test a variant to investigate the impact of additional gradient updates on model performance.

Method	oIoU	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9
Baseline	72.73	84.46	81.24	75.28	64.71	34.30
Baseline w/ additional gradient updates	72.74	84.43	81.25	75.25	64.73	34.28
Baseline w/ ours	**74.92**	**86.23**	**83.45**	**77.25**	**66.56**	**35.61**

5. Conclusion

In this work, we proposed a meta learning-based framework (MCRES) to improve the generalization performance of RES models, especially when handling novel compositions of learned concepts. By constructing a virtual training set and multiple virtual testing sets w.r.t. various levels of novel compositions and then optimizing the model via meta optimization, our framework can effectively improve model generalization performance. Extensive experiments show that our framework achieves superior performance on widely used benchmarks. Moreover, our framework is flexible, and can be seamlessly applied on various models with different architectures to enhance their performance.

Acknowledgement. This work is supported by MOE AcRF Tier 2 (Proposal ID: T2EP20222-0035), National Research Foundation Singapore under its AI Singapore Programme (AISG-100E-2020-065), and SUTD SKI Project (SKI 2021_02_06).

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 2

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 6

[3] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7454–7463, 2019. 6, 7

[4] Yi-Wen Chen, Yi-Hsuan Tsai, Tiantian Wang, Yen-Yu Lin, and Ming-Hsuan Yang. Referring expression object segmentation with caption-aware consistency. *arXiv preprint arXiv:1910.04748*, 2019. 6

[5] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16321–16330, 2021. 1, 3, 6

[6] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder fusion network with co-attention embedding for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15506–15515, 2021. 6

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 2, 3

[8] Lin Geng Foo, Tianjiao Li, Hossein Rahmani, Qiuhong Ke, and Jun Liu. Era: Expert retrieval and assembly for early action prediction. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 670–688. Springer, 2022. 3

[9] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018. 1, 2, 4, 5

[10] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 2

[11] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6172, 2020. 3

[12] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016. 1, 3

[13] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4424–4433, 2020. 6

[14] Chao Huang, Zhangjie Cao, Yunbo Wang, Jianmin Wang, and Mingsheng Long. Metasets: Meta-learning on point sets for generalizable representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8863–8872, 2021. 3

[15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10488–10497, 2020. 6

[16] 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 *European Conference on Computer Vision*, pages 59–75. Springer, 2020. 6

[17] Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu Tan. Locate then segment: A strong pipeline for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9858–9867, 2021. 6

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

[19] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 3

[20] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2018. 3, 6

[21] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[22] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1271–1280, 2017. 3

[23] 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. 6

[24] 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 *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1274–1282, 2020. 6

[25] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collabora-

tive network for joint referring expression comprehension and segmentation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 10034–10043, 2020. 1, 2, 6, 7

[26] 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, 6

[27] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 630–645, 2018. 6

[28] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*, pages 792–807. Springer, 2016. 1, 3, 6

[29] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 2, 3

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3

[31] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019. 3

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 3

[33] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 6

[34] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11686–11695, 2022. 1, 3, 6

[35] Li Xu, Haoxuan Qu, Jason Kuen, Jiuxiang Gu, and Jun Liu. Meta spatio-temporal debiasing for video scene graph generation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 374–390. Springer, 2022. 3

[36] Sibei Yang, Meng Xia, Guanbin Li, Hong-Yu Zhou, and Yizhou Yu. Bottom-up shift and reasoning for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11266–11275, 2021. 3, 6

[37] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18155–18165, 2022. 1, 2, 3, 6, 7, 8

[38] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10502–10511, 2019. 1, 6, 7

[39] Nanyang Ye, Jingxuan Tang, Huayu Deng, Xiao-Yun Zhou, Qianxiao Li, Zhenguo Li, Guang-Zhong Yang, and Zhanxing Zhu. Adversarial invariant learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12441–12449. IEEE, 2021. 2

[40] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 1, 3, 6

[41] 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, 3, 6

[42] 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 *European Conference on Computer Vision*. Springer, 2022. 1, 2, 6, 7