Query-guided Regression Network with Context Policy for Phrase Grounding

Kan Chen* Rama Kovvuri* Ram Nevatia
University of Southern California, Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089, USA
{kanchen|nkovvuri|nevatia}@usc.edu

Abstract

Given a textual description of an image, phrase grounding localizes objects in the image referred by query phrases in the description. State-of-the-art methods address the problem by ranking a set of proposals based on the relevance to each query, which are limited by the performance of independent proposal generation systems and ignore useful cues from context in the description. In this paper, we adopt a spatial regression method to break the performance limit, and introduce reinforcement learning techniques to further leverage semantic context information. We propose a novel Query-guided Regression network with Context policy (QRC Net) which jointly learns a Proposal Generation Network (PGN), a Query-guided Regression Network (QRN) and a Context Policy Network (CPN). Experiments show QRC Net provides a significant improvement in accuracy on two popular datasets: Flickr30K Entities and Referit Game, with 14.25% and 17.14% increase over the state-of-the-arts respectively.

1. Introduction

Given an image and a related textual description, phrase grounding attempts to localize objects which are mentioned by corresponding phrases in the description. It is an important building block in computer vision with natural language interaction, which can be utilized in high-level tasks, such as image retrieval [3, 26], image captioning [1, 7] and visual question answering [2, 4, 8].

Phrase Grounding is a challenging problem that involves parsing language queries and relating the knowledge to localize objects in visual domain. To address this problem, typically a proposal generation system is first applied to produce a set of proposals as grounding candidates. The main difficulties lie in how to learn the correlation between language (query) and visual (proposals) modalities, and how to localize objects based on multimodal correlation. State-of-the-art methods address the first difficulty by learning a subspace to measure the similarities between proposals and queries. With the learned subspace, they treat the second difficulty as a retrieval problem, where proposals are ranked based on their relevance to the input query. Among these, Phrase-Region CCA [25] and SCRC [14] models learn a multimodal subspace via Canonical Correlation Analysis (CCA) and a Recurrent Neural Network (RNN) respectively. Varun *et al.* [22] learn multimodal correlation aided by context objects in visual content. GroundeR [29] introduces an attention mechanism that learns to attend on related proposals given different queries through phrase reconstruction.

These approaches have two important limitations. First, proposals generated by independent systems may not always cover all mentioned objects given various queries; since retrieval based methods localize objects by choosing one of these proposals, they are bounded by the performance limits from proposal generation systems. Second, even though query phrases are often selected from image descriptions, context from these descriptions is not utilized to reduce semantic ambiguity. Consider example in Fig 1.

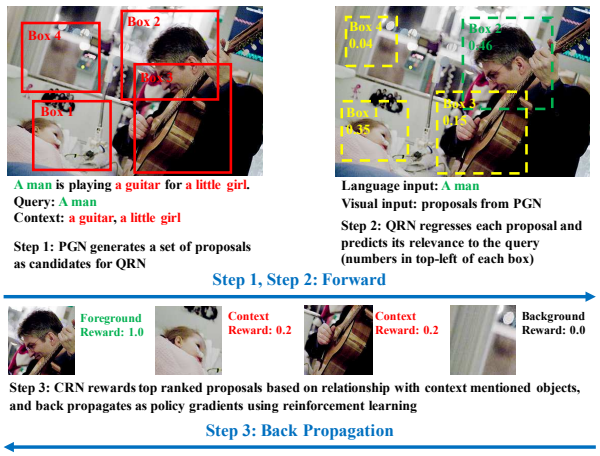

Figure 1. QRC Net first regresses each proposal based on query's semantics and visual features, and then utilizes context information as rewards to refine grounding results.

*Equal contribution. Names are sorted alphabetically.

Given the query "a man", phrases "a guitar" and "a little girl" can be considered to provide context that proposals overlapping with "a guitar" or "a little girl" are less likely to be the ones containing "a man".

To address the aforementioned issues, we propose to predict mentioned object's location rather than selecting candidates from limited proposals. We adopt a regression based method guided by input query's semantics. To reduce semantic ambiguity, we assume that different phrases in one sentence refer to different visual objects. Given one query phrase, we evaluate predicted proposals and down-weight those which cover objects mentioned by other phrases (*i.e.*, context). For example, we assign lower rewards for proposals containing "a guitar" and "a little girl" in Fig 1 to guide system to select more discriminative proposals containing "a man". Since this procedure depends on prediction results and is non-differentiable, we utilize reinforcement learning [31] to adaptively estimate these rewards conditioned on context information and jointly optimize the framework.

In implementation, we propose a novel Query-guided Regression network with Context policy (QRC Net) which consists of a Proposal Generation Network (PGN), a Query-guided Regression Network (QRN) and a Context Policy Network (CPN). PGN is a proposal generator which provides candidate proposals given an input image (red boxes in Fig. 1). To overcome performance limit from PGN, QRN not only estimates each proposal's relevance to the input query, but also predicts its regression parameters to the mentioned object conditioned on the query's intent (yellow and green boxes in Fig. 1). CPN samples QRN's prediction results and evaluates them by leveraging context information as a reward function. The estimated reward is then back propagated as policy gradients (Step 3 in Fig. 1) to assist QRC Net's optimization. In training stage, we jointly optimize PGN, QRN and CPN using an alternating method in [28]. In test stage, we fix CPN and apply trained PGN and QRN to ground objects for different queries.

We evaluate QRC Net on two grounding datasets: Flickr30K Entities [25] and Referit Game [15]. Flickr30K Entities contains more than 30K images and 170K query phrases, while Referit Game has 19K images referred by 130K query phrases. Experiments show QRC Net outperforms state-of-the-art methods by a large margin on both two datasets, with more than 14% increase on Flickr30K Entities and 17% increase on Referit Game in accuracy.

Our contributions are twofold: First, we propose a query-guided regression network to overcome performance limits of independent proposal generation systems. Second, we introduce reinforcement learning to leverage context information to reduce semantic ambiguity. In the following paper, we first discuss related work in Sec. 2. More details of QRC Net are provided in Sec. 3. Finally we analyze and compare QRC Net with other approaches in Sec. 4.

2. Related Work

Phrase grounding requires learning correlation between visual and language modalities. Karpathy *et al.* [16] propose to align sentence fragments and image regions in a subspace, and later replace the dependency tree with a bi-directional RNN in [1]. Hu *et al.* [14] propose a SCRC model which adopts a 2-layer LSTM to rank proposals using encoded query and visual features. Rohrbach *et al.* [29] employ a latent attention network conditioned on query which ranks proposals in unsupervised scenario. Other approaches learn the correlation between visual and language modalities based on Canonical Correlation Analysis (CCA) [11] methods. Plummer *et al.* [24] first propose a CCA model to learn the multimodal correlation. Wang *et al.* [34] employ structured matching and use phrase pairs to boost performance. Recently, Plummer *et al.* [25] augment the CCA model to leverage extensive linguistic cues in the phrases. All of the above approaches are reliant on external object proposal systems and hence, are bounded by their performance limits.

Proposal generation and spatial regression. Proposal generation systems are widely used in object detection and phrase grounding tasks. Two popular methods: Selective Search [33] and EdgeBoxes [38] employ efficient low-level features to produce proposals on possible object locations. Based on proposals, spatial regression method is successfully applied in object detection. Fast R-CNN [9] first employs a regression network to regress proposals generated by Selective Search [33]. Based on this, Ren *et al.* [28] incorporate the proposal generation system by introducing a Region Proposal Network (RPN) which improves both accuracy and speed in object detection. Redmon *et al.* [27] employ regression method in grid level and use non-maximal suppression to improve the detection speed. Liu *et al.* [20] integrate proposal generation into a single network and use outputs discretized over different ratios and scales of feature maps to further increase the performance. Inspired by the success of RPN in object detection, we build a PGN and regress proposals conditioned on the input query.

Reinforcement learning is first introduced to deep neural network in Deep Q-learning (DQN) [21], which teaches an agent to play ATARI games. Lillicrap *et al.* [19] modify DQN by introducing deep deterministic policy gradients, which enables reinforcement learning framework to be optimized in continuous space. Recently, Yu *et al.* [37] adopt a reinforcer to guide speaker-listener network to sample more discriminative expressions in referring tasks. Liang *et al.* [18] introduce reinforcement learning to traverse a directed semantic action graph to learn visual relationship and attributes of objects in images. Inspired by the successful applications of reinforcement learning, we propose a CPN to assign rewards as policy gradients to leverage context information in training stage.

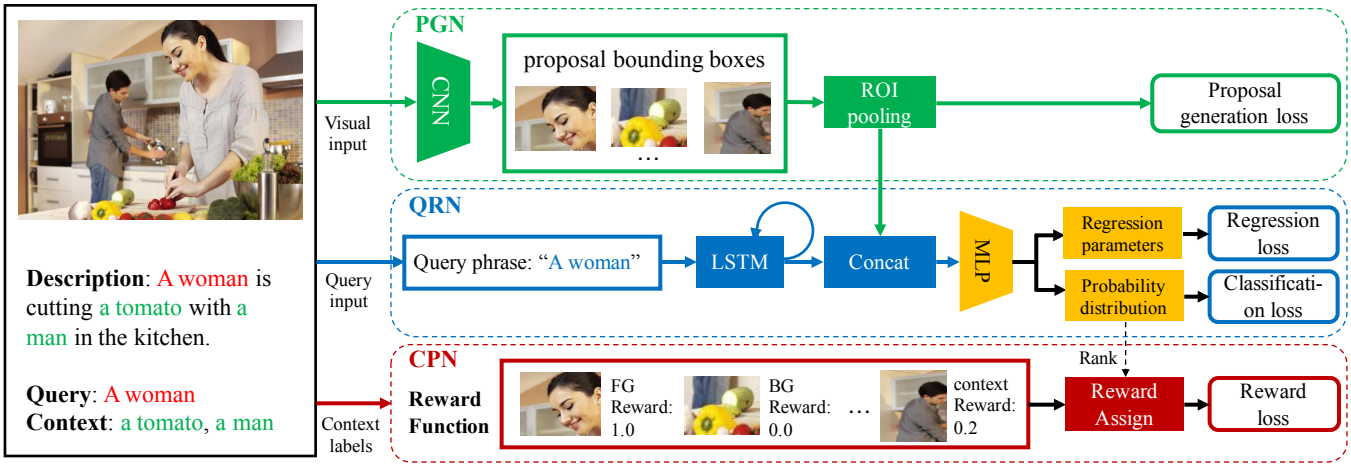

Figure 2. Query-guided Regression network with Context policy (QRC Net) consists of a Proposal Generation Network (PGN), a Query-guided Regression Network (QRN) and a Context Policy Network (CPN). PGN generates proposals and extracts their CNN features via a RoI pooling operation [28]. QRN encodes input query's semantics by an LSTM [13] model and regresses proposals conditioned on the query. CPN samples the top ranked proposals, and assigns rewards considering whether they are foreground (FG), background (BG) or context. These rewards are back propagated as policy gradients to guide QRC Net to select more discriminative proposals.

3. QRC Network

QRC Net is composed of three parts: a Proposal Generation Network (PGN) to generate candidate proposals, a Query-guided Regression Network (QRN) to regress and rank these candidates and a Context Policy Network (CPN) to further leverage context information to refine ranking results. In many instances, an image is described by a sentence that contains multiple noun phrases which are used as grounding queries, one at a time. We consider the phrases that are not in the query to provide context; specifically to infer that they refer to objects not referred to by the query. This helps rank proposals; we use CPN to optimize using a reinforcement learning policy gradient algorithm.

We first present the framework of QRC Net, followed by the details of PGN, QRN and CPN respectively. Finally, we illustrate how to jointly optimize QRC Net and employ QRC Net in phrase grounding task.

3.1. Framework

The goal of QRC Net is to localize the mentioned object's location y given an image x and a query phrase q. To achieve this, PGN generates a set of N proposals $\{r_i\}$ as candidates. Given the query q, QRN predicts their regression parameters $\{\mathbf{t}_i\}$ and probability $\{p_i\}$ of being relevant to the input query. To reduce semantic ambiguity, CPN evaluates prediction results of QRN based on the locations of objects mentioned by context phrases, and adopts a reward function F to adaptively penalize high ranked proposals containing context-mentioned objects. Reward calculation depends on predicted proposals, and this procedure is non-differentiable. To overcome this, we deploy a reinforcement learning procedure in CPN where this reward is back propagated as policy gradients [32] to optimize QRN's parameters, which guides QRN to predict more discriminative proposals. The objective for QRC Net is:

$$\arg\min_{\theta} \sum_{q} [\mathcal{L}_{gen}(\{r_i\}) + \mathcal{L}_{cls}(\{r_i\}, \{p_i\}, y) + \lambda \mathcal{L}_{reg}(\{r_i\}, \{\mathbf{t}_i\}, y) + J(\theta)] \quad (1)$$

where θ denotes the QRC Net's parameters to be optimized and λ is a hyperparameter. $\mathcal{L}_{gen}$ is the loss for generation proposals produced by PGN. $\mathcal{L}_{cls}$ is a multi-class classification loss generated by QRN in predicting the probability p_i of each proposal r_i. $\mathcal{L}_{reg}$ is a regression loss from QRN to regress each proposal r_i to the mentioned object's location y. $J(\theta)$ is the reward expectation calculated by CPN.

3.2. Proposal Generation Network (PGN)

We build PGN with a similar structure as that of RPN in [28]. PGN adopts a fully convolutional neural network (FCN) to encode the input image x as an image feature map $\mathbf{x}$. For each location (*i.e.*, anchor) in image feature map, PGN uses different scales and aspect ratios to generate proposals $\{r_i\}$. Each anchor is fed into a multiple-layer perceptron (MLP) which predicts a probability p_{o_i} estimating the objectness of the anchor, and 4D regression parameters $\mathbf{t}_i = [(x - x_a)/w_a, (y - y_a)/h_a, \log(w/w_a), \log(h/h_a)]$ as defined in [28]. The regression parameters $\mathbf{t}_i$ estimate the offset from anchor to mentioned objects' bounding boxes. Given all mentioned objects' locations $\{y_l\}$, we consider a proposal to be positive when it covers some object y_l with Intersection over Union (IoU) > 0.7, and negative when

IoU < 0.3. The generation loss is:

$$\mathcal{L}_{gen}(\{r_i\}) = -\frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} \delta(i \in S_y \cup S_{\bar{y}}) \log(p_{o_i}) + \frac{\lambda_g}{N_{reg}} \sum_{i=1}^{N_{reg}} \delta(i \in S_y) \sum_{j=0}^{3} f\left(|\mathbf{t}_i^*[j] - \mathbf{t}_i[j]|\right) \tag{2}$$

where $\delta(.)$ is an indicator function, S_y is the set of positive proposals' indexes and $S_{\bar{y}}$ is the set of negative proposals' indexes. N_{reg} is the number of all anchors and N_{cls} is the number of sampled positive and negative anchors as defined in [28]. $\mathbf{t}_i^*$ represents regression parameters of anchor i to corresponding object's location y_l. $f(.)$ is the smooth L1 loss function: $f(x) = 0.5x^2(|x| < 1)$, and $f(x) = |x| - 0.5(|x| \geq 1)$.

We sample the top N anchors based on $\{p_{o_i}\}$ and regress them as proposals $\{r_i\}$ with predicted regression parameters $\mathbf{t}_i$. Through a RoI pooling operation [28], we extract visual feature $\mathbf{v}_i \in \mathbb{R}^{d_v}$ for each proposal r_i. $\{r_i\}$ and $\{\mathbf{v}_i\}$ as fed into QRN as visual inputs.

3.3. Query guided Regression Network (QRN)

For input query q, QRN encodes its semantics as an embedding vector $\mathbf{q} \in \mathbb{R}^{d_q}$ via a Long Short-Term Memory (LSTM) model. Given visual inputs $\{\mathbf{v}_i\}$, QRN concatenates the embedding vector $\mathbf{q}$ with each of the proposal's visual feature $\mathbf{v}_i$. It then applies a fully-connected (fc) layer to generate multimodal features $\{\mathbf{v}_i^q\} \in \mathbb{R}^m$ for each of the $\langle q, r_i \rangle$ pair in an m-dimensional subspace. The multimodal feature $\mathbf{v}_i^q$ is calculated as:

$$\mathbf{v}_i^q = \varphi(\mathbf{W}_m(\mathbf{q}||\mathbf{v}_i) + \mathbf{b}_m) \tag{3}$$

where $\mathbf{W}_m \in \mathbb{R}^{(d_q+d_v)\times m}, \mathbf{b}_m \in \mathbb{R}^m$ are projection parameters. $\varphi(.)$ is a non-linear activation function. "||" denotes a concatenation operator.

Based on the multimodal feature $\mathbf{v}_i^q$, QRN predicts a 5D vector $\mathbf{s}_i^p \in \mathbb{R}^5$ via a fc layer for each proposal r_i (superscript "p" denotes prediction).

$$\mathbf{s}_i^p = \mathbf{W}_s \mathbf{v}_i^q + \mathbf{b}_s \tag{4}$$

where $\mathbf{W}_s \in \mathbb{R}^{m\times 5}$ and $\mathbf{b}_s \in \mathbb{R}^5$ are projection weight and bias to be optimized. The first element in $\mathbf{s}_i^p$ estimates the confidence of r_i being related to input query q's semantics. The next four elements are regression parameters which are in the same form as $\mathbf{t}_i$ defined in Sec. 3.2, where x, y, w, h are replaced by regressed values and x_a, y_a, w_a, h_a are proposal's parameters.

We denote $\{p_i\}$ as the probability distribution of $\{r_i\}$ after we feed $\{\mathbf{s}_i^p[0]\}$ to a softmax function. Same as [29], we consider one proposal as positive which overlaps most with ground truth and with IoU > 0.5. Thus, the classification loss is calculated as:

$$\mathcal{L}_{cls}(\{r_i\}, \{p_i\}, y) = -\log(p_{i^*}) \tag{5}$$

where i^* is positive proposal's index in the proposal set.

Given the object's location y mentioned by query q, each proposal's ground truth regression data $\mathbf{s}_i^q \in \mathbb{R}^4$ is calculated in the same way as the last four elements of $\mathbf{s}_i^p$, by replacing $[x, y, w, h]$ with the ground truth bounding box's location information. The regression loss for QRN is:

$$\mathcal{L}_{reg}(\{\mathbf{t}_i\}, \{r_i\}, y) = \frac{1}{4N} \sum_{i=1}^{N} \sum_{j=0}^{3} f\left(|\mathbf{s}_i^p[j+1] - \mathbf{s}_i^q[j]|\right) \tag{6}$$

where $f(.)$ is the smooth L1 function defined in Sec. 3.2.

3.4. Context Policy Network (CPN)

Besides using QRN to predict and regress proposals, we further apply a CPN to guide QRN to avoid selecting proposals which cover the objects referred by query q's context in the same description. CPN evaluates and assigns rewards for top ranked proposals produced from QRN, and performs a non-differentiable policy gradient [32] to update QRN's parameters.

Specifically, proposals $\{r_i\}$ from QRN are first ranked based on their probability distribution $\{p_i\}$. Given the ranked proposals, CPN selects the top K proposals $\{r_i'\}$ and evaluates them by assigning rewards. This procedure is non-differentiable, since we do not know the proposals' qualities until they are ranked based on QRN's probabilities. Therefore, we use policy gradients reinforcement learning to update the QRN's parameters. The goal is to maximize the expectation of predicted reward $F(\{r_i'\})$ under the distribution of $\{r_i'\}$ parameterized by the QRN, *i.e.*, $J = \mathbb{E}_{\{p_i\}}[F]$. According to the algorithm in [35], the policy gradient is

$$\nabla_{\theta_r} J = \mathbb{E}_{\{p_i\}}[F(\{r_i'\}) \nabla_{\theta_r} \log p_i'(\theta_r)] \tag{7}$$

where θ_r are QRN's parameters and $\nabla_{\theta_r} \log p_i'(\theta_r)$ is the gradient produced by QRN for top ranked proposal r_i.

To predict reward value $F(\{r_i'\})$, CPN averages top ranked proposals' visual features $\{\mathbf{v}_i'\}$ as $\mathbf{v}_c$. The predicted reward is computed as:

$$F(\{r_i'\}) = \sigma(\mathbf{W}_c(\mathbf{v}_c||\mathbf{q}) + \mathbf{b}_c) \tag{8}$$

where "||" denotes concatenation operation and $\sigma(.)$ is a sigmoid function. $\mathbf{W}_c$ and $\mathbf{b}_c$ are projection parameters which produce a scalar value as reward.

To train CPN, we design a reward function to guide CPN's prediction. The reward function performs as feedback from environment and guide CPN to produce meaningful policy gradients. Intuitively, to help QRN select more

discriminative proposals related to query q rather than context, we assign lower reward for some top ranked proposal that overlaps the object mentioned by context and higher reward if it overlaps with the mentioned object by query. Therefore, we design the reward function as:

$$R(\{r'_i\}) = \frac{1}{K}\sum_{i=1}^{K}[\delta(r'_i \in S_q) + \beta\delta(r'_i \notin (S_q \cup S_{bg}))] \quad (9)$$

where S_q is the set of proposals with IoU > 0.5 with mentioned objects by query q, and S_{bg} is the set of background proposals with IoU < 0.5 with objects mentioned by all queries in the description. $\delta(.)$ is an indicator function and $\beta \in (0, 1)$ is the reward for proposals overlapping with objects mentioned by context. The reward prediction loss is:

$$\mathcal{L}_{rwd}(\{r'_i\}) = ||F(\{r'_i\}) - R(\{r'_i\})||^2 \quad (10)$$

During training, $\mathcal{L}_{rwd}$ is backpropagated only to CPN for optimization, while CPN backpropagates policy gradients (Eq. 7) to optimize QRN.

3.5. Training and Inference

We train PGN based on an RPN pre-trained on PASCAL VOC 2007 [6] dataset, and adopt the alternating training method in [28] to optimize PGN. We first train PGN and use proposals to train QRN and CPN, then initialize PGN tuned by QRN and CPN's training, which iterates one time. Same as [29], we select 100 proposals produced by PGN ($N = 100$) and select top 10 proposals ($K = 10$) predicted by QRN to assign reward in Eq. 9. After calculating policy gradient in Eq. 7, we jointly optimize QRC Net's objective (Eq. 1) using Adam algorithm [17]. We choose the rectified linear unit (ReLU) as the non-linear activation function.

During testing stage, CPN is fixed and we stop its reward calculation. Given an image, PGN is first applied to generate proposals and their visual features. QRN regresses these proposals and predicts the relevance of each proposal to the query. The regressed proposal with highest relevance is selected as the prediction result.

4. Experiment

We evaluate QRC Net on Flickr30K Entities [25] and Referit Game datasets [15] for phrase grounding task.

4.1. Datasets

Flickr30K Entities [25]: The numbers of training, validation and testing images are 29783, 1000, 1000 respectively. Each image is associated with 5 captions, with 3.52 query phrases in each caption on average. There are 276K manually annotated bounding boxes referred by 360K query phrases in images. The vocabulary size for all these queries is 17150.

Referit Game [15] consists of 19,894 images of natural scenes. There are 96,654 distinct objects in these images. Each object is referred to by 1-3 query phrases (130,525 in total). There are 8800 unique words among all the phrases, with a maximum length of 19 words.

4.2. Experiment Setup

Proposal generation. We adopt a PGN (Sec. 3.2) to generate proposals. During training, we optimize PGN based on an RPN pre-trained on PASCAL VOC 2007 dataset [6], which does not overlap with Flickr30K Entities [25] or Referit Game [15]. We also evaluate QRC Net based on Selective Search [33] (denoted as "SS") and EdgeBoxes [38] (denoted as "EB"), and an RPN [28] pre-trained on PASCAL VOC 2007 [23] (denoted as "RPN"), which are all independent of QRN and CPN.

Visual feature representation. For QRN, the visual features are directly generated from PGN via a RoI pooling operation. Since PGN contains a VGG Network [30] to process images, we denote these features as "VGG_{pgn}". To predict regression parameters, we need to include spatial information for each proposal. For Flickr30K Entities, we augment each proposal's visual feature with its spatial information $[x_{tl}/W, y_{tl}/H, x_{br}/H, y_{br}/W, wh/WH]$ as defined in [36]. These augmented features are 4101D vectors ($d_v = 4101$). For Referit Game, we augment VGG_{pgn} with each proposal's spatial information $[x_{min}, y_{min}, x_{max}, y_{max}, x_{center}, y_{center}, w_{box}, h_{box}]$ which is same as [29] for fair comparison. We denote these features as "VGG_{pgn}-SPAT", which are 4104D vectors ($d_v = 4104$).

To compare with other approaches, we replace PGN with a Selective Search and an EdgeBoxes proposal generator. Same as [29], we choose a VGG network finetuned using Fast-RCNN [9] on PASCAL VOC 2007 [6] to extract visual features for Flickr30K Entities. We denote these features as "VGG_{det}". Besides, we follow [29] and apply a VGG network pre-trained on ImageNet [5] to extract proposals' features for Flickr30K Entities and Referit Game, which are denoted as "VGG_{cls}". We augment VGG_{det} and VGG_{cls} with spatial information for Flickr30K Entities and Referit Game datasets following the method mentioned above.

Model initialization. Following same settings as in [29], we encode queries via an LSTM model, and choose the last hidden state from LSTM as $\mathbf{q}$ (dimension $d_q = 1000$). All convolutional layers are initialized by MSRA method [12] and all fc layers are initialized by Xavier method [10]. We introduce batch normalization layers after projecting visual and language features (Eq. 3).

During training, the batch size is 40. We set weight λ for regression loss L_{reg} as 1.0 (Eq. 1), and reward value $\beta = 0.2$ (Eq. 9). The dimension of multimodal feature vector $\mathbf{v}_i^q$ is set to $m = 512$ (Eq. 3). Analysis of hyperparameters is provided in Sec. 4.3 and 4.4.

Approach	Accuracy (%)
Compared approaches	
SCRC [14]	27.80
Structured Matching [34]	42.08
SS+GroundeR (VGG_{cls}) [29]	41.56
RPN+GroundeR (VGG_{det}) [29]	39.13
SS+GroundeR (VGG_{det}) [29]	47.81
MCB [8]	48.69
CCA embedding [25]	50.89
Our approaches	
RPN+QRN (VGG_{det})	53.48
SS+QRN (VGG_{det})	55.99
PGN+QRN (VGG_{pgn})	60.21
QRC Net (VGG_{pgn})	**65.14**

Table 1. Different models' performance on Flickr30K Entities. Our framework is evaluated by combining with various proposal generation systems.

Proposal generation	RPN [28]	SS [33]	PGN
UBP (%)	71.25	77.90	89.61
BPG	7.29	3.62	7.53

Table 2. Comparison of different proposal generation systems on Flickr30k Entities

Metric. Same as [29], we adopt accuracy as the evaluation metric, defined to be the ratio of phrases for which the regressed box overlaps with the mentioned object by more than 50% IoU.

Compared approaches. We choose GroundeR [29], CCA embedding [25], MCB [8], Structured Matching [34] and SCRC [14] for comparison, which all achieve leading performances in phrase grounding. For GroundeR [29], we compare with its supervised learning scenario, which achieves the best performance among different scenarios.

4.3. Performance on Flickr30K Entities

Comparison in accuracy. We first evaluate QRN performance based on different independent proposal generation systems. As shown in Table 1, by adopting QRN, RPN+QRN achieves 14.35% increase compared to RPN+GorundeR. We further improve QRN's performance by adopting Selective Search (SS) proposal generator. Compared to SS+GroundeR, we achieve 8.18% increase in accuracy. We then incorporate our own PGN into the framework, which is jointly optimized to generate proposals as well as features (VGG_{pgn}). By adopting PGN, PGN+QRN achieves 4.22% increase compared to independent proposal generation system (SS+QRN) in accuracy. Finally, we include CPN to guide QRN in selecting more discriminative proposals during training. The full model (QRC Net) achieves 4.93% increase compared to PGN+QRN, and 14.25% increase over the state-of-the-art CCA embedding [25] in accuracy.

Weight λ	0.5	1.0	2.0	4.0	10.0
Accuracy (%)	64.15	65.14	64.40	64.29	63.27

Table 3. QRC Net's performances on Flickr30K Entities for different weights λ of L_{reg}.

Dimension m	128	256	512	1024
Accuracy (%)	64.08	64.59	65.14	62.52

Table 4. QRC Net's performances on Flickr30K Entities for different dimensions m of $\mathbf{v}_i^q$.

Reward β	0.1	0.2	0.4	0.8
Accuracy (%)	64.10	65.14	63.88	62.77

Table 5. QRC Net's performances on Flickr30K Entities for different reward values β of CPN.

Detailed comparison. Table 6 provides the detailed phrase localization results based on the phrase type information for each query in Flickr30K Entities. We can observe that QRC Net provides consistently superior results. CCA embedding [25] model is good at localizing "instruments" while GroundeR [29] is strong in localizing "scene". By using QRN, we observe that the regression network achieves consistent increase in accuracy compared to GroundeR model (VGG_{det}) in all phrase types except for class "instruments". Typically, there is a large increase in performance of localizing "animals" (with increase of 11.39%). By using PGN, we observe that PGN+QRN has surpassed state-of-the-art method in all classes, with largest increase in class "instruments". Finally, by applying CPN, QRC Net achieves more than 8.03%, 9.37%, 8.94% increase in accuracy in all categories compared to CCA embedding [25], Structured Matching [34] and GroundeR [29] respectively. QRC Net achieves the maximum increase in performance of 15.73% for CCA embedding [25] ("scene"), 32.90% for Structured Matching [34] ("scene") and 21.46% for GroundeR [29] ("clothing").

Proposal generation comparison. We observe proposals' quality plays an important role in final grounding performance. The influence has two aspects. First is the Upper Bound Performance (UBP) which is defined as the ratio of covered objects by generated proposals in all ground truth objects. Without regression mechanism, UBP directly determines the performance limit of grounding systems. Another aspect is the average number of surrounding Bounding boxes Per Ground truth object (BPG). Generally, when BPG increases, more candidates are considered as positive, which reduces the difficulty for following grounding system. To evaluate UBP and BPG, we consider that a proposal covers the ground truth object when its IoU > 0.5. The statistics for RPN, SS and PGN in these two aspects

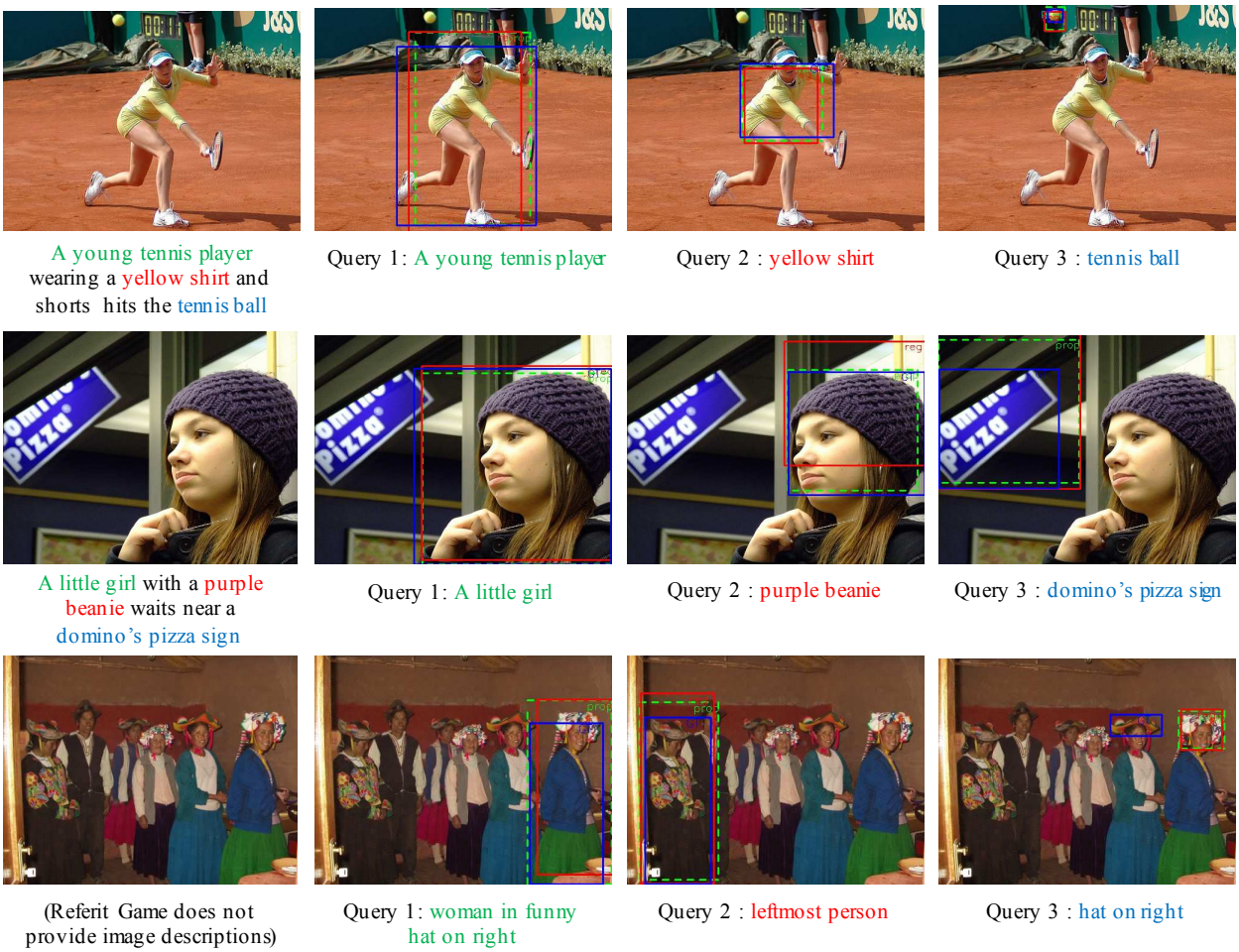

Figure 3. Some phrase grounding results in Flickr30K Entities [25] (first two rows) and Referit Game [15] (third row). We visualize ground truth bounding box, selected proposal box and regressed bounding box in blue, green and red resepctively. When query is not clear without further context information, QRC Net may ground wrong objects (*e.g.*, image in row three, column four).

Phrase Type	people	clothing	body parts	animals	vehicles	instruments	scene	other
GroundeR (VGG_{cls}) [29]	53.80	34.04	7.27	49.23	58.75	22.84	52.07	24.13
GroundeR (VGG_{det}) [29]	61.00	38.12	10.33	62.55	68.75	36.42	58.18	29.08
Structured Matching [34]	57.89	34.61	15.87	55.98	52.25	23.46	34.22	26.23
CCA embedding [25]	64.73	46.88	17.21	65.83	68.75	37.65	51.39	31.77
SS+QRN	68.24	47.98	20.11	73.94	73.66	29.34	66.00	38.32
PGN+QRN	75.08	55.90	20.27	73.36	68.95	45.68	65.27	38.80
QRC Net	**76.32**	**59.58**	**25.24**	**80.50**	**78.25**	**50.62**	**67.12**	**43.60**

Table 6. Phrase grounding performances for different phrase types defined in Flickr30K Entities. Accuracy is in percentage.

are provided in Table 2. We observe that PGN achieves increase in both UBP and PBG, which indicates PGN provides high quality proposals for QRN and CPN. Moreover, since QRN adopts a regression-based method, it can surpass UBP of PGN, which further relieves the influence from UBP of proposal generation systems.

Hyperparameters. We evaluate QRC Net for different sets of hyperparameters. To evaluate one hyperparameter, we fix other hyperparameters to default values in Sec. 4.2.

We first evaluate QRC Net's performance for different regression loss weights λ. The results are shown in Table 3. We observe the performance of QRC Net fluctuates when λ is small and decreases when λ becomes large.

We then evaluate QRC Net's performance for different

Approach	Accuracy (%)
Compared approaches	
SCRC [14]	17.93
EB+GroundeR (VGG_{cls}-SPAT) [29]	26.93
Our approaches	
EB+QRN (VGG_{cls}-SPAT)	32.21
PGN+QRN (VGG_{pgn}-SPAT)	43.57
QRC Net (VGG_{pgn}-SPAT)	**44.07**

Table 7. Different models' performance on Referit Game dataset.

dimensions m for multimodal features in Eq. 3. The performances are presented in Table 4. We observe QRC Net's performance fluctuates when $m < 1000$. When m becomes large, the performance of QRC Net decreases. Basically, these changes are in a small scale, which shows the insensitivity of QRC Net to these hyperparameters.

Finally, we evaluate different reward values β for proposals covering objects mentioned by context. We observe QRC Net's performance fluctuates when $\beta < 0.5$. When β is close to 1.0, the CPN assigns almost same rewards for proposals covering ground truth objects or context mentioned objects, which confuses the QRN. As a result, the performance of QRC Net decreases.

4.4. Performance on Referit Game

Comparison in accuracy. To evaluate QRN's effectiveness, we first adopt an independent EdgeBoxes [38] (EB) as proposal generator, which is same as [29]. As shown in Table 7, by applying QRN, we achieve 5.28% improvement compared to EB+GroundeR model. We further incorporate PGN into the framework. PGN+QRN model brings 11.36% increase in accuracy, which shows the high quality of proposals produced by PGN. Finally, we evaluate the full QRC Net model. Since Referit Game dataset only contains independent query phrases, there is no context information available. In this case, only the first term in Eq. 9 guides the learning. Thus, CPN does not contribute much to performance (0.50% increase in accuracy).

Hyperparameters. We evaluate QRC Net's performances for different hyperparameters on Referit Game dataset. First, we evaluate QRC Net's performance for different weights λ of regression loss L_{reg}. As shown in Table 8, performance of QRC Net fluctuates when λ is small. When λ becomes large, regression loss overweights classification loss, where a wrong seed proposal may be selected which produces wrong grounding results. Thus, the performance decreases.

We then evaluate QRC Net's performance for different multimodal dimensions m of $\mathbf{v}_i^q$ in Eq. 3. In Table 9, we observe performance changes in a small scale when $m < 1000$, and decreases when $m > 1000$.

Weight λ	0.5	1.0	2.0	4.0	10.0
Accuracy (%)	43.71	44.07	43.61	43.60	42.75

Table 8. QRC Net's performances on Referit Game for different weights λ of L_{reg}.

Dimension m	128	256	512	1024
Accuracy (%)	42.95	43.80	44.07	43.51

Table 9. QRC Net's performances on Regerit Game for different dimensions m of $\mathbf{v}_i^q$.

4.5. Qualitative Results

We visualize some phrase grounding results of Flickr30K Entities and Referit Game for qualitative evaluation (Fig. 3). For Flickr30K Entities, we show an image with its associated caption, and highlight the query phrases in it. For each query, we visualize the ground truth box, the selected proposal box by QRN and the regressed bounding box based on the regression parameters predicted by QRN. Since there is no context information in Referit Game, we visualize query and ground truth box, with selected proposal and regressed box predicted by QRN.

As shown in Fig 3, QRC Net is strong in recognizing different people ("A young tennis player" in the first row) and clothes ("purple beanie" in the second row), which is also validated in Table 6. However, when the query is ambiguous without further context description, QRC Net may be confused and produce reasonably incorrect grounding result (*e.g.*, "hat on the right" in the third row of Fig. 3).

5. Conclusion

We proposed a novel deep learning network (QRC Net) to address the phrase grounding task. QRC Net adopts regression mechanism and leverages context information, which achieves 14.25% and 17.14% increase in accuracy on Flickr30K Entities [25] and Referit Game [15] datasets respectively.

Acknowledgements

This paper is based, in part, on research sponsored by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

References

[1] K. Andrej and F.-F. Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1, 2

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Z., and D. Parikh. VQA: Visual question answering. In *ICCV*, 2015. 1

[3] K. Chen, T. Bui, C. Fang, Z. Wang, and R. Nevatia. AMC: Attention guided multi-modal correlation learning for image search. In *CVPR*, 2017. 1

[4] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia. ABC-CNN: An attention based convolutional neural network for visual question answering. *CVPR Workshop*, 2016. 1

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge. In *IJCV*, 2010. 5

[7] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 1

[8] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *EMNLP*, 2016. 1, 6

[9] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 2, 5

[10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, 2010. 5

[11] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 2004. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *CVPR*, 2015. 5

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 3

[14] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016. 1, 2, 6, 8

[15] S. K., V. O., M. M., and T. L. B. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 2, 5, 7, 8

[16] A. Karpathy, A. Joulin, and F.-F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014. 2

[17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] X. Liang, L. Lee, and E. P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. *CVPR*, 2017. 2

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *ICLR*, 2016. 2

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 2

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015. 2

[22] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 1

[23] N. Pham and R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *ACM SIGKDD*, 2013. 5

[24] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 2

[25] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *IJCV*, 2016. 1, 2, 5, 6, 7, 8

[26] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 1

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 3, 4, 5, 6

[29] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016. 1, 2, 4, 5, 6, 7, 8

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. 5

[31] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998. 2

[32] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 1999. 3, 4

[33] J. R. Uijlings, K. E. Van D. S., T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2, 5, 6

[34] M. Wang, M. Azab, N. Kojima, R. Mihalcea, and J. Deng. Structured matching for phrase localization. In *ECCV*, 2016. 2, 6, 7

[35] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992. 4

[36] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *ECCV*, 2016. 5

[37] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. *CVPR*, 2017. 2

[38] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2, 5, 8