Image Captioning using Adversarial Networks and Reinforcement Learning

Shiyang Yan[1,2], Fangyu Wu[1,2], Jeremy S. Smith[1], Wenjin Lu[2] and Bailing Zhang[2]
[1]Department of Electrical Engineering and Electronics, University of Liverpool
[2]Department of Computer Science and Software Engineering,
Xi'an Jiaotong-liverpool University
Email: shiyang.yan@xjtlu.edu.cn

***Abstract*—Image captioning is a significant task in artificial intelligence which connects computer vision and natural language processing. With the rapid development of deep learning, the sequence to sequence model with attention, has become one of the main approaches for the task of image captioning. Nevertheless, a significant issue exists in the current framework: the exposure bias problem of Maximum Likelihood Estimation (MLE) in the sequence model. To address this problem, we use generative adversarial networks (GANs) for image captioning, which compensates for the exposure bias problem of MLE and also can generate more realistic captions. GANs, however, cannot be directly applied to a discrete task, like language processing, due to the discontinuity of the data. Hence, we use a reinforcement learning (RL) technique to estimate the gradients for the network. Also, to obtain the intermediate rewards during the process of language generation, a Monte Carlo roll-out sampling method is utilized. Experimental results on the COCO dataset validate the improved effect from each ingredient of the proposed model. The overall effectiveness is also evaluated.**

I. Introduction

Image captioning, i.e., automatically describing the content of an image, is a fundamental problem in machine learning which connects computer vision and natural language processing. It tries to mimick the human ability to process huge amounts of salient visual information into descriptive language, which is one of the primary goals of artificial intelligence.

In recent years, remarkable progresses have been made towards naturalistic image description generation [1] [2] [3] [4], owning to the development of deep learning [5]. In these works, inspired by the success of the sequence-to-sequence model of neural machine translation [6] [7], most of them represented the image as a single feature vector from the top layer of a pre-trained convolutional neural network (CNN) and cascaded recurrent neural network (RNN) to generate text. Subsequent research [3] introduced the attention mechanism on image locations to discriminate important and relevant image features to facilitate image captioning.

However, most of the previously proposed models trained the RNN using Maximum Likelihood Estimation (MLE) to generate image descriptions. As argued in [8], the MLE approaches suffer from the so-called exposure bias in the inference stage: the model generates a sequence iteratively and and predicts the next token based on the previously predicted ones that may never be observed in the training data. In image captioning, the MLE also suffers from a problem that the generated captions do not correlate well with a human assessment of quality.

Instead of only relying on the MLE, an alternative scheme is under the framework of generative adversarial network (GAN) [9]. GAN was first proposed to generate realistic images. GAN learns generative models without explicitly defining a loss function from the target distribution. Instead, GAN introduces a discriminator network which tries to differentiate real samples from generated samples. The whole network is trained using this adversarial training strategy. One can subsequently build a discriminator to judge how realistic the samples generated by the caption generator are. The caption generator is similar to the generator in conditional GAN [10], which is conditioned on the image features.

There is an inherent problem in GAN when dealing with language problems. Language, unlike images, is essentially a discrete problem. Directly providing these discrete tokens as inputs to the discriminator does not allow the gradients to back propagate through them since they are discontinuous. One solution is to implement a reinforcement learning (RL) [11] framework to estimate the gradients of the discontinuous units. However, the RL framework, when dealing with sequence generation, has the problem of lacking the intermediate reward, as discussed in [12]. The reward signal can only be obtained when the whole sequence is generated. This is not suitable, since what we want is the long-term reward of each intermediately generated token, which is to better optimize the whole sequence.

To tackle the above-mentioned issues, we follow the framework of GAN for image captioning. In the proposed scheme, the discriminator not only considers the similarity between the generated captions and the reference captions but also the consistencies between the captions and image features. Through evaluation of the discriminator, the networks can better compensate the issue where some unrealistic captions might be generated using MLE. Also, to deal with the discreteness of language, we treat the image captioning generator as an agent of RL. The feedbacks from the discriminator are considered as the rewards for the generator. To update the parameters of image captioning generator in this framework, we consider the generator as a stochastic parameterized policy. We train the policy network using Policy Gradient [13], which naturally

solves the differential difficulties in conventional GAN. Also, to solve the problem of the lack of intermediate rewards, we borrow the idea from the famous "AlphaGo" program [14] in which a Monte Carlo roll-out strategy is applied to sample the expected long-term reward for an intermediate move. If we consider the sequence token generation as the the action to be taken in RL, we can apply a similar Monte Carlo roll-out strategy to obtain the intermediate rewards. [12] has successfully applied the Monte Carlo roll-out in sequence generation. In this paper, we use a similar sampling method to deal with intermediate rewards during the process of caption generation.

During implementation, we build our caption generator based on the "show, attend and tell" model [3]. The feature processing and soft attention mechanism are adopted as the same in [3]. We then treat the image captioning model as the generator, and use another RNN network as a discriminator, to automatically evaluate how realistic the generated captions are. The outputs from the discriminator are considered as the rewards in the RL framework. The entire networks are trained using the Policy Gradient algorithm. We evaluated our model on the COCO dataset [15], with improved results over the model based on MLE.

Our contributions can be summarized as follows:

- We propose to use GAN and RL to train a neural model for the image captioning task.
- A Monte Carlo roll-out strategy is applied to obtain intermediate rewards for RL in the sequence generation scenario.
- Experiments prove the effectiveness of adversarial training and RL in the task of image captioning.

II. The proposed method

The proposed scheme is based on generative adversarial networks (GANs), in which a generator and a discriminator are trained using the minimax game in an adversarial way. On one hand, the generator tries to generate realistic samples to fool the discriminator into believing they are real ones. On the other hand, the discriminator is trained to identify the differences between generated samples and real ones.

In the proposed scheme, the image captioning generator is considered as the generator in a GAN framework, which tries to generate naturalistic image descriptions. We build a discriminator to judge whether the generated sequence is realistic. In the vanilla GAN, the gradient from the discriminator can be back propagated directly to the generator, which makes the whole network trainable. However, due to the discrete problem of language, this is not achievable using vanilla GAN. Hence, we treat the model in the framework of RL and apply a Policy Gradient to estimate the gradients of the generator. In the following subsections we will explain the generator, the discriminator, the Policy Gradient algorithm and the training algorithm, respectively. The system diagram can be seen in Fig. 1.

A. Image Captioning Generator

The image caption generator is based on the model in [3]. Specifically, the model consists of an encoder and a decoder. We use a convolutional neural network (Residual Net [16]) pre-trained on the ImageNet dataset [17] in order to extract a set of convolutional features. These features, denoted as $a = \{a_1, ..., a_L\}$, correspond to certain portions of the 2-D image. We extract convolutional features instead of fully connected ones in order to build a soft attention mechanism to discriminate the visual location of the given image.

The Long-short Term Memory (LSTM) network, originally proposed by Hochreiter and Schmidhuber in [18], is applied as the language decoder because of its superior performance in natural language processing.

$$\begin{aligned} i_t &= \sigma(W_{xi} * z_t + W_{hi} * h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * z_t + W_{hf} * h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo} * z_t + W_{ho} * h_{t-1} + b_o) \\ g_t &= \sigma(W_{xc} * z_t + W_{hc} * h_{t-1} + b_c) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\ h_t &= o_t \cdot \phi(c_t) \end{aligned} \tag{1}$$

In Equation 1, i_t, f_t, o_t, c_t and h_t are the input, forget, output, cell memory and hidden state of a LSTM network, respectively. z_t is the context vector, which can be processed by a soft attention mechanism and is able to capture visual information associated with certain input locations. The soft attention mechanism has to automatically allocate adaptive weights, on image locations, to facilitate the task at hand.

$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{2}$$

Equation 2 actually maps the image features from each location, along with information from the hidden state, into an adaptive weight, which indicates the importance of each image location for recognition.

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})} \tag{3}$$

Then, Equation 3 normalizes the adaptive weights into a probability value in the range of 0 to 1 using the softmax function. Once these weights (sum to 1) are computed, we element-wisely multiply the weights vector α_t with the image feature vector a and sum them to generate the context vector z_t, which can be expressed as in Equation 4.

$$z_t = \sum_{i=1}^{L} \alpha_{t,i} a_i \tag{4}$$

Then the context vector z_t is forwarded to the LSTM network to generate captions, as described in Equation 1. This soft attention mechanism is able to adaptively select relevant visual parts of the given image features and thus facilitate recognition.

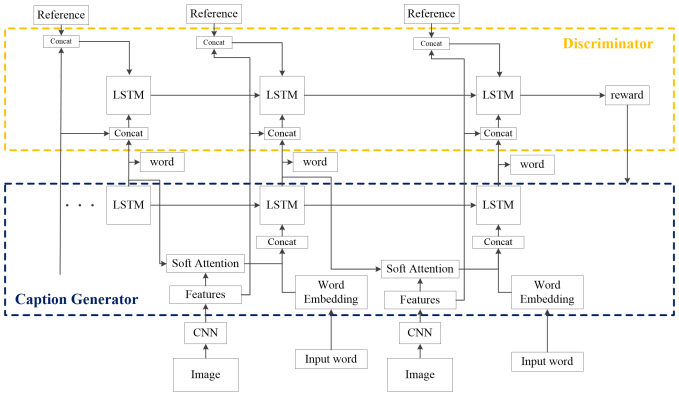


Fig. 1. System Diagram: Our system contains a generator and a discriminator. The generator is an image caption generator with soft attention while the discriminator is a LSTM network which provides rewards to update the parameters of the generator using RL.

B. Discriminator

We feed both the generated sequences and the reference sequences to the discriminator. Before being forwarded to the discriminator, both of the embedding matrices of the generated sequences and the reference sequences are concatenated with the image features, which can be seen in Fig. 1. This operation is to consider the coherence between certain captions (sequences) and the corresponding image features, which is able to make the generated captions more realistic and naturalistic. The reference sequences are labeled as true whilst the generated sequences are labeled as false during the training of the discriminator. The model is also a LSTM network with softmax cross entropy loss. Hence, the discriminator outputs the probabilities of a sample being true. These probabilities, are then considered as the reward signal in the RL framework, to be utilized by the Policy Gradient algorithm for updating the parameters of the image caption generator.

C. Optimization via Policy Gradient

Following [13], the objective of the policy network $G_\theta(y_t|y_{1:t-1})$ (the image caption generator), is to generate a sequence from the start state s_0 to maximize its expected long-term reward as described in Equation 5:

$$J(\theta) = E[R_T|s_0, \theta] = \sum_{y_1 \in Y} G_\theta(y_1|s_0) \cdot Q^{G_\theta}_{D_\theta}(s_0, y_1) \quad (5)$$

where R_T is the reward for a complete sequence. $Q^{G_\theta}_{D_\theta}(s, y)$ is the action-value function of a language sequence, which is defined as the expected accumulative reward starting from state s, taking action a, and then following policy G_θ.

The action-value function is estimated using the REINFORCE algorithm [19] and considers the probability of being real, generated by the discriminator, as a reward, which can be defined as in Equation 6.

$$Q^{G_\theta}_{D_\theta}(a = y_T, s = Y_{1:T-1}) = D_\theta(Y_{1:T}) \quad (6)$$

As can be seen in Equation 6, the discriminator only provides a reward for a complete sequence. We should not only care about the reward for complete tokens but also the long-term reward for the future time-steps since the long-term reward is what we actually want. Similar to the game of Go [14] in which the agent sometimes gives up immediate interest but cares about the final victory, we apply a similar Monte Carlo roll-out strategy for an intermediate state, i.e., an unfinished sequence. We represent an N-time Monte Carlo search as in Equation 7.

$$\begin{aligned} Y^1_{1:T}, ..., Y^N_{1:T} &= MC^{G_\theta}(Y_{1:t}; N) \\ MC &=\sim Multinomial(logits) \end{aligned} \quad (7)$$

where $Y^n_{1:T}$ is the generated sequence tokens and $Y^n_{t+1:T}$ is Monte Carlo sampled based on a roll-out policy, which, in our case, is set the same as the image caption generator. $logits$ is the output of LSTM decoder. MC is defined as a sampling procedure from Multinomial distribution.

If there is no intermediate reward, the Monte Carlo roll-out strategy can sample the future possible tokens N times and average these rewards to achieve the goal of reward estimation, which is described in Equation 8.

$$Q^{G_\theta}_{D_\theta}(a = y_t, s = Y_{1:t-1}) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\theta(Y^n_{1:T}), Y^n_{1:T} \in MC^{G_\theta}(Y_{1:t}; N), & for\ t < T \\ D_\theta(Y_{1:T}), & for\ t = T \end{cases} \quad (8)$$

The Monte Carlo roll-out strategy can be better visualized in Fig. 2.

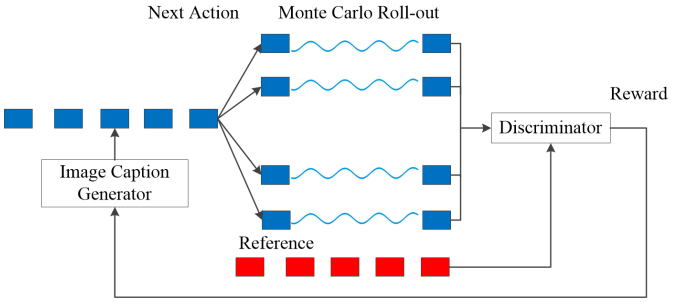


Fig. 2. Monte Carlo roll-out: We use Monte Carlo sampling to sample tokens in the future time steps and average them to obtain the intermediate rewards so as to optimize the token generated at each time step.

Once the reward value from the discriminator is obtained, it is ready to update the generator. We can use the Policy Gradient theorem from [13] and write the gradient of the objective function (reward signal) as in Equation 9.

$$\bigtriangledown_\theta J(\theta) = \sum_{t=1}^{T} E_{Y_{1:t-1} \sim G_\theta}[\sum_{y_t \in Y} \bigtriangledown G_\theta(y_t|Y_{1:t-1}) \cdot Q^{G_\theta}_{D_\theta}(Y_{1:t-1}, y_t)] \quad (9)$$

Since the expectation can be approximated by sampling, we can now update our parameters of the image caption generator using Equation 10.

$$\theta \leftarrow \theta + \alpha_h \bigtriangledown_\theta J(\theta) \quad (10)$$

In practice, we can use advanced gradient algorithms such as RMSprop [20] and Adam [21] in training the caption generator.

D. Adversarial Training

The image caption generator and discriminator are adversarially trained in the GAN framework [9]. In GAN [10], the discriminator can pass the gradient directly to the generator. Due to the discreteness of sequence generation, we apply RL to estimate the gradient for the generator in our model.

Specifically, the training strategy can be described in Algorithm 1. We firstly pre-train the image caption generator using MLE. In practice, this is equivalent to the cross entropy loss [22]. Hence, we can set the pre-training step the same as in [3]. The trained model is used to generate some captions which are set as fake samples, which, along with the reference captions, are fed to the discriminator for training. Similarly, the discriminator is also pre-trained for certain steps. The next step is the adversarial training step, in which the image caption generator and discriminator are trained alternatively until the convergence of the networks.

Algorithm 1 Image Caption Generation by Adversarial Training and Reinforcement Learning

Require: Image Caption Generator G_θ; Discriminator D_θ.
  Pre-training G_θ using MLE by some epoches.
  Generating negative samples using pre-trained G_θ to train D_θ.
  Pre-training D_θ by 2500 iterations.
  repeat
    for update generator for 1 step **do**
      Generate a sequence $Y_{1:T} = (y_1, .., y_T)$.
      for $t = 1$ to T **do**
        Compute the intermediate reward $Q(t)$ by Monte Carlo roll-out.
      end for
      Update the parameters θ using Policy Gradient.
    end for
    for update discriminator for 1 step **do**
      Training discriminator D_θ using reference sequence (True) and generated sequence (Fake) using current generator.
    end for
  until Convergence

III. Experimental Results

A. Experimental protocol

We conducted our experiments using the COCO dataset [15]. To be consistent with [3], we use the COCO 2014 released version, which includes 123,000 images. We used the "Karpathy" splits [1]. The standard evaluation protocol contains BLEU [23] and METEOR [24].

At training time, we set the maximum length of the input sequence to 20. During the alternate testing phase, we set the maximum length of the generated symbols to 30.

TABLE I
COMPARISON OF IMAGE CAPTIONING RESULTS ON THE COCO DATASET WITH DIFFERENT IMAGE ENCODERS

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Soft attention with MLE (VGG-19)	65.7	44.7	30.5	21.1	21.6
Soft attention with GAN and RL (VGG-19)	66.7	45.4	31.0	21.4	21.5
Soft attention with MLE (Residual Net)	70.0	50.3	35.4	25.1	23.6
Soft attention with GAN and RL (Residual Net)	**71.6**	**51.8**	**37.1**	**26.5**	**24.3**

TABLE II
EXPERIMENTAL VALIDATION OF THE IMPROVEMENT BY USING MONTE CARLO ROLL-OUT

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Soft attention with GAN and RL without Monte Carlo roll-out (VGG-19)	66.0	45.0	30.4	21.1	21.3
Soft attention with GAN and RL with Monte Carlo roll-out (VGG-19)	**66.7**	**45.4**	**31.0**	**21.4**	**21.5**
Soft attention with GAN and RL without Monte Carlo roll-out (Residual Net)	71.2	50.9	36.8	26.2	24.0
Soft attention with GAN and RL with Monte Carlo roll-out (Residual Net)	**71.6**	**51.8**	**37.1**	**26.5**	**24.3**

TABLE III
COMPARISON OF IMAGE CAPTIONING RESULTS ON THE COCO DATASET WITH PREVIOUS METHODS

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
CMU/MS Research [28]	-	-	-	-	20.4
MS Research [29]	-	-	-	-	20.7
LRCN [30]	58.7	39.0	25.0	16.5	-
BRNN [1]	64.2	45.1	30.4	20.3	-
Google NIC	66.6	46.1	32.9	24.6	-
Log Bilinear [3]	70.8	48.9	34.4	24.3	20.0
LSTM with Soft attention [3]	70.7	49.2	34.4	24.3	23.9
LSTM with hard attention [3]	**71.8**	50.4	35.7	25.0	23.0
RL with G-GAN [31]	-	-	30.5	29.7	22.4
RL with Embedding Reward [32]	71.3	**53.9**	**40.3**	**30.4**	**25.1**
Soft attention with GAN and RL (VGG-19)	66.7	45.4	31.0	21.4	21.5
Soft attention with GAN and RL (Residual Net)	71.6	51.8	37.1	26.5	24.3

B. Implementation Details

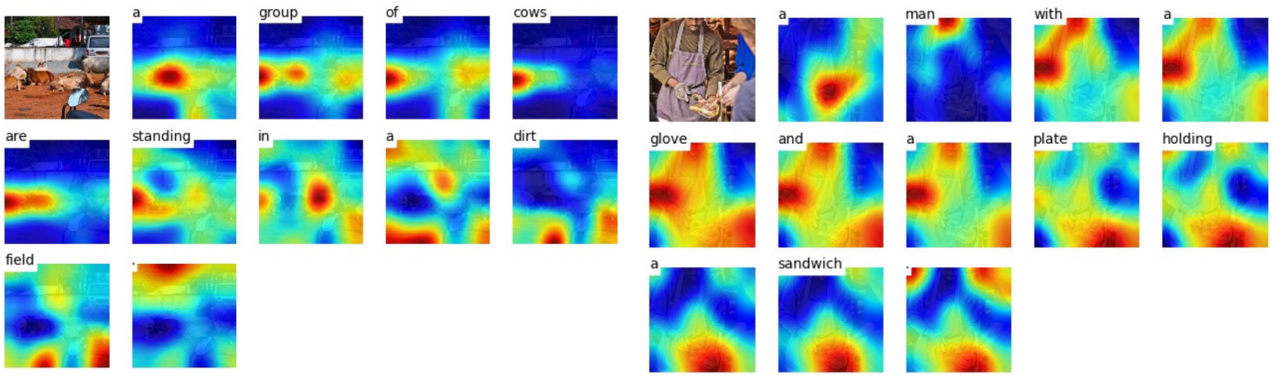
Given raw images, we resize them to 224 $\times$ 224 pixels. Then we extract deep convolutional features (from the layer "res5c") using a pre-trained Residual-152 network [16] under the Caffe platform [25] because of its high efficiency in extracting features. We also extract the features from the first fully connected layer from the VGG16 [26] network to make an experimental comparison on different image encoders. We re-implement the "show, attend and tell" model on the Tensorflow platform [27]. The adversarial networks and Monte Carlo roll-out are also implemented under the same platform.

We set the batch size as 64 and learning rate to 0.0001 for both the MLE pre-training and Adversarial training. The number of Monte Carlo roll-outs is set as 20. During sampling, we fetch the maximum log-likelihoods that the network outputs. Although other techniques, like beam search, are proven to be better than maximum log-likelihoods, we are interested in the improvement of the model itself instead of other greedy techniques. Hence, we all use the maximum log-likelihoods sampling in both the MLE training and adversarial training.

C. Results

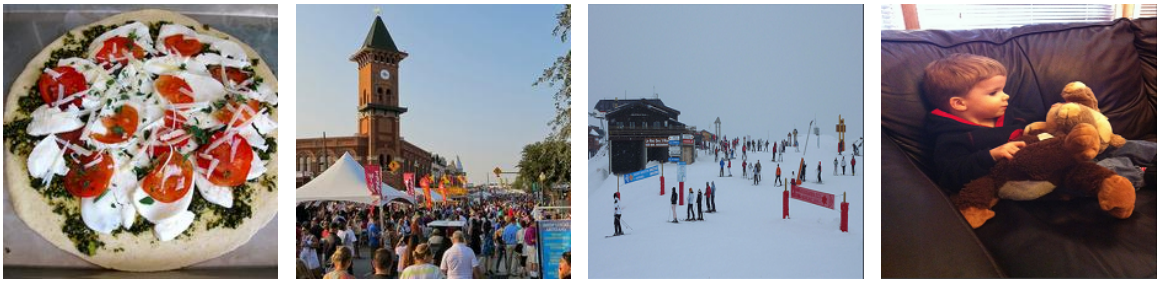
1) Quantitative Evaluation:

- Following [3], we evaluated the generated captions using the metrics of BLEU (1-4) and Meteor and performed certain ablation studies on different settings.
- In addition to using the Residual Net as an image encoder, we also utilized VGG-19 [26] as the image encoder to



(a) (b)

Fig. 3. Visualization of attention maps: the red regions means selected parts while blue regions means unimportant features. All samples are randomly selected.



(a) Caption generated by MLE: A pizza with tomatoes and onions on it.
Caption generated by our model: A pizza with cheese and vegetables on a plate.

(b) Caption generated by MLE: A crowd of people standing in front of a building.
Caption generated by our model: A crowd of people standing around a large clock tower.

(c) Caption generated by MLE: A group of people standing on top of a snow covered slope.
Caption generated by our model: A group of people are skiing on a snowy hill.

(d) Caption generated by MLE: A small child is playing a video game.
Caption generated by our model: A small child sitting on a couch holding a stuffed animal.

Fig. 4. Visualization of generated languages: the red color texts indicate the captions generated by our model, which is more accurate and realistic than the blue text captions generated by MLE model. All samples are randomly selected.

see the important role of advanced image features in the image captioning task. The results can be seen in Table I. The advanced image features from Residual Net bring a significant gain on the overall performance of caption generation. Take the results using MLE for example, for the metric of BLEU (1-4), the average raise is 4.7, which is a very obvious increase for the image captioning task.

- Given the same image features, our method using GAN and RL leads the MLE method in most of the evaluation metrics, under the same image features and the same generator model, which proves the effectiveness of the adversarial training and policy gradient technique, which is shown in Table I.
- To study the effectiveness of Monte Carlo roll-out, we first tested a model without a Monte Carlo roll-out strategy, i.e., the reward can only be obtained after the whole captions are generated. We compared the results of this model with the model using the Monte Carlo roll-out strategy, which can be seen in Table II. As the results reveal, scores from all the evaluation metrics increase by adding an intermediate reward using Monte Carlo roll-out.
- As described in Table III, we compared our best results with other related published researches. As indicated by the results, our method outperforms many related approaches including the attention models [3], which validates the improved effect brought by adversarial training and RL. RL with G-GAN [31] applies conditional GAN and policy gradient to generate image descriptions. Although their results on the evaluation metrics are not improved, they prove that the generated captions are more diverse and natural. Embedding Reward [32] applies a policy network to generate captions and a value network to evaluate the reward. Additionally, they also apply an advanced inference method called lookahead inference and beam search during testing. We also achieved competitive results on this dataset.

2) Qualitive Evaluation:

- The visualization of the attention maps learnt can be seen in Fig. 3. In different time steps, the model adaptively selects relevant parts for the generated word. In the figure, a red region means these parts are selected whilst a blue

region indicates unimportant parts.

- We also randomly select some examples of generated captions for both the MLE model and our model, which are described in Fig. 4. In the figure, the generated captions from our model are more accurate and realistic since our discriminator is able to measure the coherence between captions and image contents.

IV. Conclusion

This research focused on the image captioning task, which is a fundamental problem in artificial intelligence. To address the inherent exposure bias problem of MLE training in sequence problems, an adversarial training method was applied. To estimate the gradients of the network, the feedback from the discriminator was treated as the reward signal in the RL framework. In RL, a long-term reward for each action is needed. In sequence generation, however, the reward can only be obtained when the sequence is generated. To tackle this issue, a Monte Carlo roll-out sampling method was applied to estimate the intermediate reward for each time step. The whole network was trained using the proposed three-step training strategy, which includes pre-training the regenerator, pre-training the discriminator, and adversarial training. Experimental results prove the improved effects of the proposed method. In addition, visualization shows the generated captions from the proposed model are more accurate than the ones from MLE training.

References

[1] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[2] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 652–663, 2017.

[3] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML15*, 2015, pp. 2048–2057.

[4] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4651–4659.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, 2015.

[7] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[8] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[10] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[12] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient." in *AAAI*, 2017, pp. 2852–2858.

[13] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[14] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[20] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[22] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

[23] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[24] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, vol. 29, 2005, pp. 65–72.

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[28] X. Chen and C. L. Zitnick, "Learning a recurrent visual representation for image caption generation," *CoRR*, vol. abs/1411.5654, 2014. [Online]. Available: http://arxiv.org/abs/1411.5654

[29] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1473–1482.

[30] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[31] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional gan," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2970–2979.

[32] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 290–298.