Self-critical n-step Training for Image Captioning

Junlong Gao[1], Shiqi Wang[4], Shanshe Wang*[2,3], Siwei Ma[2,3], and Wen Gao[2,3]

[1]Peking University Shenzhen Graduate School
[2]Institute of Digital Media, Peking University [3]Peng Cheng Laboratory
[4]Department of Computer Science, City University of Hong Kong, Hong Kong

Abstract

Existing methods for image captioning are usually trained by cross entropy loss, which leads to exposure bias and the inconsistency between the optimizing function and evaluation metrics. Recently it has been shown that these two issues can be addressed by incorporating techniques from reinforcement learning, where one of the popular techniques is the advantage actor-critic algorithm that calculates per-token advantage by estimating state value with a parametrized estimator at the cost of introducing estimation bias. In this paper, we estimate state value without using a parametrized value estimator. With the properties of image captioning, namely, the deterministic state transition function and the sparse reward, state value is equivalent to its preceding state-action value, and we reformulate advantage function by simply replacing the former with the latter. Moreover, the reformulated advantage is extended to n-step, which can generally increase the absolute value of the mean of reformulated advantage while lowering variance. Then two kinds of rollout are adopted to estimate state-action value, which we call self-critical n-step training. Empirically we find that our method can obtain better performance compared to the state-of-the-art methods that use the sequence level advantage and parametrized estimator respectively on the widely used MSCOCO benchmark.

1. Introduction

Image captioning aims at generating natural captions automatically for images, which is of great significance in scene understanding. It is a very challenging task, which requires to recognize important objects in the image, as well as their attributes and relationships between each other, such that they can be finally described properly in natural language. The ability of the machine to mimic human in expressing rich information in natural language with correct grammar is important since it can be applied to human-robot interaction and blind users guiding.

Inspired by the recently introduced encoder-decoder framework for machine translation in [6], most recent works in image captioning have adopted this paradigm to generate captions for images [24]. In general, an encoder, *e.g*. convolutional neural network (CNN), encode images to visual features, while a decoder, *e.g*. long short term memory (LSTM) [10], decodes the visual features to generate captions. These methods are trained in an end-to-end manner to minimize cross entropy loss, *i.e*. maximize the likelihood of each ground-truth word given the preceding ground-truth word, which is also known as "Teacher Forcing" [14].

The first problem of cross entropy loss is that it will lead to "exposure bias", since in the training stage, the model is only fed with ground-truth word at each time step, while in the testing stage, the model is fed with the previously predicted word. This discrepancy between training and testing easily results in error accumulation during generation, as the model is not exposed to its predictions during training and difficult to handle the errors which never occur in the training stage. In order to handle exposure bias, Bengio *et al*. [4] feed back the model own predictions as input with scheduled sampling, while Lamb *et al*. [14] proposed the " Professor Forcing" on top of the "Teacher Forcing".

The second problem of cross entropy loss is that the generated sentences are evaluated in the testing stage by non-differentiable metrics, such as BLEU 1-2-3-4 [20], ROUGE [15], METEOR [3], CIDEr [23], SPICE [1], while during training the model is trained to minimize cross entropy loss, which is the inconsistency between the optimizing function and evaluation metrics. The methods proposed in [4, 14] cannot address this inconsistency. Recently, it has shown that policy gradient algorithm in reinforcement learning (RL) can be trained to avoid exposure bias and directly optimize such non-differentiable evaluation metrics

*Corresponding author

[17, 21, 22, 29]. In this way, the model can be exposed to its own predictions during training. However, the algorithms in [22] use sequence level advantage that implicitly makes an invalid assumption that every token makes the same contribution to the whole sequence. Many works [17, 21, 29] have been proposed to model per-token advantage. However, they utilize a parametrized value/baseline estimator at the cost of introducing estimation bias.

In this paper, we improve the advantage actor-critic algorithm to estimate per-token advantage without introducing the biased parametrized value estimator. With the properties of image captioning, namely, the deterministic state transition function and the sparse reward, state value is equivalent to its preceding state-action value, and we reformulate advantage function by simply replacing the former with the latter. Since state-action value cannot be precisely estimated, the model may easily converge to the local maxima trained with the reformulated advantage function. Therefore, we propose n-step reformulated advantage function, which can generally increase the absolute value of the mean of reformulated advantage while lowering variance. In order to estimate state-action value, we use Monte Carlo rollouts inspired by [17, 28] and max-probability rollout inspired by [22], which is termed as self-critical n-step training. According to the empirical results, our model improves the performance of image captioning compared to the methods that use the sequence level advantage and parametrized estimator respectively.

Overall, we make the following contributions in this paper: (1) with the special properties of image captioning, we find the equivalence between state value and its preceding state-action value, and reformulate the original advantage function for each action; (2) on top of the reformulated advantage function, we extend to n-step reformulated advantage function to generally increase the the absolute value of the mean of reformulated advantage while lowering variance; (3) we utilize two kinds of rollout to estimate state-action value function to perform self-critical training.

2. Related Work

Many different models have been developed for image captioning, which can be divided into two categories: template-based methods [8, 13] and neural network-based methods. Since our method adopts neural network architecture, we mainly introduce methods in this vein. Efforts of this line have been devoted to two directions: attention mechanism and reinforcement learning.

2.1. Attention Mechanism

The encoder-decoder framework of machine translation [6] was firstly introduced by [24], which feeds the last fully connected feature of the image into RNN to generate the caption. Xu *et al*. [26] proposed soft and hard attention mechanisms to model the human's eye focusing on different regions in the image when generating different words. This work is further improved in [2, 5, 18, 22]. In [18], they introduced a visual sentinel to allow the attention module to selectively attend to visual and language features. Anderson *et al*. [2] adopted a bottom-up module, that uses object detection to detect objects in the image, and a top-down module that utilizes soft attention to dynamically attend to these object features. Chen *et al*. [5] proposed a spatial and channel-wise attention model to attend to visual features. Rennie *et al*. [22] proposed FC model and Att2in models which achieve good performance.

2.2. Reinforcement Learning

Recently a few works use reinforcement learning-based methods to address the exposure bias and the mismatch between the optimizing function and the non-differentiable evaluation metrics [17,21,22,29] in image captioning. Ranzato *et al*. [21] firstly introduced REINFORCE algorithm [25] to sequence training with RNNs. However, REINFORCE algorithm often results in large variance in gradient estimation. To lower the variance of the policy gradient, many works have introduced different kinds of baseline into REINFORCE algorithm. For example, the reward of the caption generated by the inference algorithm is adopted as the baseline in [22], which uses sequence level advantage while the per-token advantage was not considered. A variety of algorithms proposed in [17, 21, 29] aim at modeling the per-token advantage. Ranzato *et al*. [21] used a baseline reward parametric estimator. In [17], they used FC layers to predict the baseline and used Monte Carlo rollouts to predict the state-action value function. In [29], they combined the advantage actor-critic algorithm and temporal difference learning, and used another RNN to predict the state value function. However, the value/baseline estimator was used in [17,21,29], which introduces estimation bias. In this paper, we utilize the properties of image captioning to reformulate the advantage actor-critic method and use different kinds of rollout to estimate the state-action value function to calculate per-token advantage without introducing bias.

3. Methodology

3.1. Training with cross entropy loss

Given an image I, the goal of image captioning is to generate a token sequence $\mathbb{A} = \{a_1, a_2, ..., a_T\}$, $a_t \in A$, where A is the dictionary. The captioning model predicts a token sequence starting with a_0 and ending with a_T, where a_0 is a special token BOS indicating the start of the sentence, and a_T is also a special token EOS indicating the end of the sentence. In order to simplify the formulas, T is denoted as the total length of a generated sequence, ignoring the fact that generated token sequences have different lengths. We use

the standard encoder-decoder architecture for image captioning, where a CNN as an encoder, encodes an image I to an image feature I_F, and a RNN can be adopted as a decoder to decode I_F to output a token sequence $\mathbb{A}$. In this work, we adopt the Att2in model proposed by [22]. Given a ground-truth sequence $\{a_1^*, a_2^*, ..., a_T^*\}$, the model parameters θ are trained to minimize the cross entropy loss (XENT)

$$L(\theta) = -\sum_{t=1}^{T} \log\left(\pi_\theta\left(a_t^* \left| a_{1:t-1}^*, I_F\right.\right)\right) \tag{1}$$

where $\pi_\theta\left(a_t \left| a_{1:t-1}, I_F\right.\right)$ is a probability distribution of the token a_t given the preceding generated tokens $\{a_1, a_2, ..., a_{t-1}\}$ and the image feature I_F.

3.2. Training using policy gradient

Problem formulation. To address both problems of the cross entropy loss described above, namely, the exposure bias and the inconsistency between the optimizing function and evaluation metrics, we incorporate the reinforcement learning into image captioning. Formally, we consider captioning process as a finite Markov process (MDP). Our captioning model introduced above can be viewed as an agent, which interacts with an environment (words and images). In the MDP setting $\{S, A, P, R, \gamma\}$, S is a state space, A is an action space as well as the dictionary, $P\left(s_{t+1}|s_t, a_t\right)$ is state transition probability, $R(s_t, a_t)$ is reward function and $\gamma \in (0, 1]$ is the discounted factor. The agent selects an action, that corresponds to generating a token, from a conditional probability distribution $\pi\left(a\left|s\right.\right)$ called policy. In policy gradient algorithms, we consider a set of candidate policies $\pi_\theta\left(a\left|s\right.\right)$ parametrized by θ. The state $s_t \in S$ is considered as a list composing of the image feature I_F and the tokens/actions $\{a_0, a_1, a_2, ..., a_{t-1}\}$ generated so far:

$$s_t = \{I_F, a_0, a_1, ..., a_{t-1}\} \tag{2}$$

Here we define the initial state $s_0 = \{I_F\}$. At each time step, the RNN consumes s_t and uses the hidden state of RNN to generate the next token a_t. With the definition of the state, we have the next state $s_{t+1} = \{s_t, a_t\}$: we simply append the token a_t to s_t. According to the process, the state transition function P can be called deterministic state transition function. Formally, we have:

$$P\left(s_{t+1}|s_t, a_t\right) \equiv 1 \tag{3}$$

When the state s_t is transferred to the next state s_{t+1} by selecting action a_t, the agent receives reward r_t issued from the environment. However, in image captioning, we can only obtain a reward $r = R(s_T, a_T) = R\left(a_{1:T}\right)$ when EOS token is generated and $\{I_F, a_0\}$ is not considered in reward calculation. The reward r is computed by evaluating the generated complete sentences compared with corresponding ground-truth sentences under an evaluation metric. Therefore, we define the reward for each action as follows:

$$r_t = \begin{cases} 0, t < T \\ r, t = T \end{cases} \tag{4}$$

In reinforcement learning, a value function is a prediction of the expected, accumulative, γ discounted future reward, measuring how good each state, or state-action pair, is. We define the state-action value function $Q^\pi(s_t, a_t)$ and the state value function $V^\pi(s_t)$ of the policy π as follows:

$$\begin{aligned} Q^\pi(s_t, a_t) &= E_{s_{t+1}, a_{t+1}, ... \sim \pi}\left[\sum_{l=0}^{T} \gamma^l r_{t+l} \left| S_t = s_t, A_t = a_t\right.\right] \\ V^\pi(s_t) &= E_{a_t, s_{t+1}, ... \sim \pi}\left[\sum_{l=0}^{T} \gamma^l r_{t+l} \left| S_t = s_t\right.\right] \end{aligned} \tag{5}$$

where $Q^\pi(s_t, a_t)$ is the expected γ discounted accumulated reward under policy π starting from taking action a_t at state s_t, and $V^\pi(s_t)$ is the expected γ discounted accumulated reward starting from state s_t. To simplify the notation, we denote $E_{a_t, s_{t+1}, ... \sim \pi}\left[\cdot\right]$ and $E_{s_{t+1}, a_{t+1}, ... \sim \pi}\left[\cdot\right]$ with $E_\pi\left[\cdot\right]$ in the rest of paper. It is obvious that the difference between $Q^\pi(s_t, a_t)$ and $V^\pi(s_t)$ lies in whether taking the action a_t or not at state s_t when calculating the accumulated reward. In reinforcement learning, the agent aims to maximize the circumulative reward $L(\theta) = V^\pi(s_0) = E_\pi\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$ by estimating the gradient $\nabla_\theta L(\theta)$ and updating its parameters, instead of minimizing the cross entropy loss as Eq. (1).

In policy gradient methods, the gradient $\nabla_\theta L(\theta)$ can be written as:

$$\nabla_\theta L(\theta) = E_\pi\left[\left(Q^\pi(s_t, a_t) - b(s_t)\right) \nabla_\theta \log \pi_\theta\left(a_t|s_t\right)\right] \tag{6}$$

where the baseline $b(s_t)$ can be any arbitrary function, as long as it does not depend on action a_t. This baseline does not change the expected gradient, but can decrease the variance of the gradient estimate significantly. This algorithm is known as REINFORCE with a Baseline. Using $V^\pi(s_t)$ as the baseline $b(s_t)$, the algorithm is changed to advantage actor-critic (A2C) algorithm as follows:

$$\nabla_\theta L(\theta) = E_\pi\left[A^\pi(s_t, a_t) \nabla_\theta \log \pi_\theta\left(a_t|s_t\right)\right] \tag{7}$$

In Eq. (7), $A^\pi\left(s_t, a_t\right) = Q^\pi(s_t, a_t) - V^\pi(s_t)$ is called advantage function. This equation intuitively guides the agent to an evolution direction that increases the probability of better-than-average actions and decrease the probability of worse-than-average actions [29].

1-step reformulated advantage function. Image captioning is a special case in reinforcement learning, for its state transition is deterministic, while other applications can have different next states with a certain probability, such as Atari Games. Here we use this property to reformulate Eq. (7).

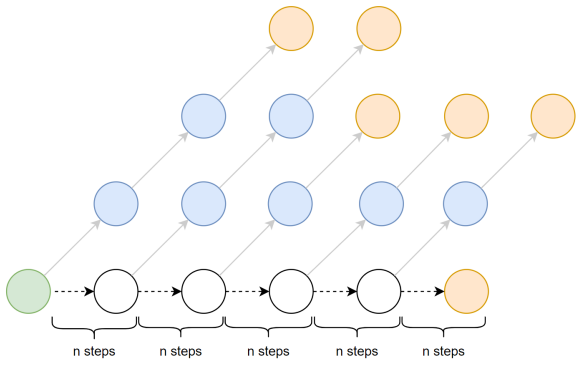


Figure 1. Each state-action value is estimated by the average rewards of K rollouts sequences ($K = 1$) or a reward of a max-probability rollout. The advantage function in our method is estimated by the current state-action value minus the preceding n-step state-action value. The tokens in green and yellow are the special tokens BOS and EOS. The tokens in white are the Monte Carlo trajectory and the tokens in blue are continuation rollout tokens for state-action value estimation. The n steps in the figure means that the model performs rollouts every n steps in n-step reformulated advantage function.

With the definition of $Q^\pi(s_t, a_t)$ and $V^\pi(s_t)$ in Eq. (5), we have

$$Q^\pi\left(s_{t-1}, a_{t-1}\right) = r_{t-1} + \gamma \sum_{s_t \in S} P\left(s_t \left| s_{t-1}, a_{t-1}\right.\right) V^\pi(s_t) \tag{8}$$

Due to the deterministic state transition function described above in Eq. (3), Eq. (8) can be rewritten as

$$Q^\pi\left(s_{t-1}, a_{t-1}\right) = r_{t-1} + \gamma V^\pi(s_t) \tag{9}$$

In this paper, we set discounted factor $\gamma = 1$. According to reward function of Eq. (4), when $t \leq T$, we have $r_{t-1} = 0$. Then $V^\pi(s_t)$ can be written as

$$V^\pi(s_t) = Q^\pi\left(s_{t-1}, a_{t-1}\right) \tag{10}$$

Eq. (10) indicates that given the two properties of image captioning, namely the deterministic state transition function and the reward function, state value is equivalent to its preceding state-action value. Then we can rewrite Eq. (7) by incorporating Eq. (10) into Eq. (7) as follows:

$$\nabla_\theta L\left(\theta\right) = E_\pi\left[A_R^\pi(s_t, a_t) \nabla_\theta \log \pi_\theta\left(a_t | s_t\right)\right] \tag{11}$$

where $A_R^\pi\left(s_t, a_t\right) = Q^\pi(s_t, a_t) - Q^\pi(s_{t-1}, a_{t-1})$ is the reformulated advantage function from $A^\pi\left(s_t, a_t\right)$ in Eq. (7). Therefore, $Q^\pi(s_{t-1}, a_{t-1})$ is a new baseline of $Q^\pi(s_t, a_t)$ instead of $V^\pi(s_t)$. Each state-action value uses its preceding state-action value as baseline, such that it is termed as 1-step reformulated advantage function.

In our approach, the agent aims at maximizing Eq. (11) rather than Eq. (7). Eq. (11) has an intuitive interpretation that it helps the agent to increase the probability of the action which has larger expected accumulated rewards compared to that of preceding action and decrease the probability of the action which has smaller expected accumulated rewards compared to that of preceding action.

The most straightforward way to simulate the environment with the current policy π is to obtain a Monte Carlo trajectory $\{(s_t, a_t, r_t)\}_{t=1}^T$ from the multinomial strategy and estimate the gradient $\nabla_\theta L\left(\theta\right)$:

$$\hat{\nabla}_\theta L\left(\theta\right) = \frac{1}{T} \sum_{t=1}^{T} \hat{A}_R^\pi(s_t, a_t) \nabla_\theta \log \pi_\theta\left(a_t | s_t\right) \tag{12}$$

where $\hat{A}_R^\pi\left(s_t, a_t\right) = \hat{Q}^\pi(s_t, a_t) - \hat{Q}^\pi(s_{t-1}, a_{t-1})$, and $\hat{Q}^\pi(s_t, a_t)$ is an empirical estimate of $Q^\pi(s_t, a_t)$.

n-step reformulated advantage function. According to the property of Eq. (11) described above, the model encourages tokens better than its preceding token in terms of the value, and surpress the worse tokens. Though Eq. (11) is a greedy algorithm, Eq. (11) can guide the evolution direction of the model towards the global maxima only when state-action value is estimated precisely. Image captioning is considered as a model-free reinforcement learning task, which uses rollouts or function approximation to estimate state-action value. However, both methods, where the former suffers from a large variance and the latter introduces estimation bias, cannot predict absolutely precise value that may turn out to be wrong to encourage or suppress a token in this strict greedy strategy. Therefore, we introduce n-step reformulated advantage function. In n-step reformulated advantage function, we view n steps as a large step to perform Eq. (11). Each step within the large step shares the n-step reformulated advantage $\hat{A}_R^\pi(s_t, a_t)$ as follows:

$$\hat{A}_R^\pi\left(s_t, a_t\right) = \hat{Q}^\pi(s_{\tau+n}, a_{\tau+n}) - \hat{Q}^\pi(s_\tau, a_\tau) \tag{13}$$

where $\tau = \lfloor t/n \rfloor n$ and $\lfloor \cdot \rfloor$ is denoted as a round-down function, and n ranges from 1 to T which unifies the two extremes, namely, 1-step and T-step. In the n-step reformulated advantage, n steps show a much clearer evolution trend of the Monte Carlo trajectory from the multinomial strategy than 1 step, and the values of neighboring states in n-step have a more precise margin than that in 1-step, except that state-action value estimation use the same strategy of Monte Carlo trajectory that samples one sequence from multinomial strategy. If they use the same strategy, estimated values of each time step are from the same distribution and thus larger n cannot enlarge the margin of neighboring state values. Therefore, except for that particular case, as n increases, the absolute value of the mean and the variance of reformulated advantage will be increased and reduced respectively.

However, as n increases, per-token advantage is inevitably gradually lost until a sequence level advantage of

$n = T$. Therefore, different methods of estimating state-action value have different distributions and have different most suitable n that performs best in balancing the approximation of per-token reformulated advantage and the improvement on the absolute mean of reformulated advantage. In general, the performance of small n is better than that of $n = T$.

Estimating the state-action value function. According to Eq. (13), we only need to estimate $\hat{Q}^\pi(s_t, a_t)$. Here, we propose two methods to estimate non-parametric $\hat{Q}^\pi(s_t, a_t)$: use K Monte Carlo rollouts inspired by [17, 28] and use inference algorithm (max-probability rollout) inspired by [22]. These processes are illustrated as Fig. 1. Since $Q^\pi(s_t, a_t)$ is an expected accumulated reward, K Monte Carlo is more stable and precise than max-probability rollout to estimate $Q^\pi(s_t, a_t)$ with additional computation cost of $K - 1$ rollouts. In 1-step reformulated advantage function, the model rollouts every steps, while in n-step reformulated advantage function, the model rollouts every n steps. Therefore, the model adopts self-critical training [22], which uses rollouts to estimate value functions as a critic.

In K Monte Carlo rollouts, we sample K continuations of the sequence $\{s_t, a_t\}$ to obtain $\{a_{t+1}, a_{t+2}, ..., a_T\}$, which means that the subsequent tokens are sampled from the multinomial strategy. When $\gamma = 1$, according to Eq. (4) and Eq. (5), the state-action value function can be computed by the average of the K rewards

$$\hat{Q}^\pi(s_t, a_t) = \frac{1}{K} \sum_{k=1}^{K} R\left(a_{1:t}; a^k_{t+1:T}\right) \tag{14}$$

where $R\left(a_{1:t}; a^k_{t+1:T}\right)$ is denoted as the reward of the k'th continuation sampled after $\{s_t, a_t\}$ from the multinomial strategy. In our experiment, we set $K = 5$. A slight difference between our method and [17, 28] is that we need to rollout from $\{s_0, a_0\}$ to estimate $Q^\pi(s_0, a_0)$, and they do not. If $K = 1$, state-action value estimation and Monte Carlo trajectory both sample a sequence from multinomial strategy in each step, and thus larger n cannot enlarge the margin of neighboring state values as discussed above. As K increases, though K rollouts of estimating state-action value are also sampled from multinomial strategy, the mean reward of K can estimate more precise state-action value than $K = 1$ (*i.e*. state-action value estimation and Monte Carlo trajectory use different strategies in $K > 1$) and thus larger n will have larger absolute value of the mean of reformulated advantage with lower variance.

In max-probability rollout, we sample only one continuations of the sequence $\{s_t, a_t\}$ to obtain $\{\hat{a}_{t+1}, \hat{a}_{t+2}, ..., \hat{a}_T\}$, which are tokens of the largest probabilities at every time step. Then we have

$$\hat{Q}^\pi(s_t, a_t) = R\left(a_{1:t}; \hat{a}_{t+1:T}\right) \tag{15}$$

where $R\left(a_{1:t}; \hat{a}_{t+1:T}\right)$ means the reward of the max-probability rollout sequence after $\{s_t, a_t\}$ under the inference algorithm. Interestingly, SCST [22] is equivalent to T-step reformulated advantage function using max-probability rollout, *i.e*. SCST is a variant of ours. Here, state-action value estimation and Monte Carlo trajectory use different strategies, where the former are from max-probability strategy and the latter are from multinomial strategy. Moreover, max-probability strategy can always obtain better sequence than multinomial strategy. Therefore, though the reward of max-probability rollout cannot reflect the real state-action value, larger n can have larger absolute value of the mean of reformulated advantage with lower variance.

It is worth noting that the rollout of preceding step can be used both in preceding token and this token with different effects. Here, we directly optimize CIDEr metric, *i.e*. R is CIDEr score. Moreover, only when calculating the last reformulated advantage of each sequence that includes token EOS, we use CIDEr with EOS as a token. Otherwise, we use CIDEr without EOS as a token. It is because EOS is not a normal token of a sentence like other words but a special token indicating the ending of the sentence, and it is ignored in the standard calculation of evaluation metric scores.

4. Experiments

4.1. Dataset

We evaluate our method on the MSCOCO dataset [16]. For fair comparisons, we use the widely used splits from [11]. The training set contains $113,287$ images with 5 captions for each image and $5K$ images for validation and $5K$ images for offline testing. We follow the standard practice to preprocess all captions, including converting all captions to lower case, tokenizing on white space, truncating captions longer than 16 words, and replacing words that do not occur at least 5 times with UNK token resulting in 9487 words in the dictionary. To evaluate generated caption quality, we use the standard metrics, namely BLEU 1-2-3-4, ROUGE, METEOR, CIDEr, SPICE. We extract image features using Resnet-101 [9] without finetune.

4.2. Implementation Details

The embedding dimensions of the LSTM hidden, image, word and attention are all fixed to 512 for all the models. We pretrain all the models under XENT loss for 30 epochs using ADAM [12] optimizer with default settings and fixed learning rate 4×10^{-4}. During training under XENT loss, our batch size is set to 80. We then run RL training with a fixed learning rate 5×10^{-5}. In RL training, we use the models trained under XENT loss as the pretrained model in order to reduce the search space, and the batch size is set to 32. In the whole training process, we use fixed dropout rate 0.5 to prevent the models from overfitting.

4.3. Experiment Configuration

Here are the configurations of the basic model and several variants of our models. This series of experiments are designed to explore the effects of different n-step, different combinations of n and K Monte Carlo rollouts versus max-probability rollout. Besides, we re-implement two state-of-the-art reinforcement learning-based model SCST [22] and PG-CIDEr [17], and all the hyperparameters are the same as those of our proposed models for fair comparison.

(1) XENT is the basic model trained with cross entropy loss, which is then used as the pretrained model of all reinforcement learning-based models.

(2) For max-probabilty rollout, we conduct n-step-maxpro ($n = 1, 2, 4$) that are trained with n-step reformulated advantage throughout the whole training time. We also conduct models trained with different n-step successively, *e.g.* 1-2-4-T-step-maxpro, T-4-2-1-step-maxpro, 1-2-2-step-maxpro.

(3) For K Monte Carlo rollouts, we conduct 1-step-sample that is trained with 1-step reformulated advantage using K Monte Carlo rollout to estimate the state-action value function. We also conduct 1-2-2-step-sample.

(4) SCST [22] (*i.e.* T-step-maxpro) uses sequence level advantage for every token in a sampled sequence. Here, we compare self-critical per-token advantage with self-critical sequence level advantage.

(5) PG-CIDEr [17] uses K Monte Carlo rollouts with a parametrized estimator. Here, we compare self-critical per-token advantage with parametrized per-token advantage.

4.4. Quantitative Analysis

Performance of the Karpathy test split. In Table 1, we report the performance of our models, SCST [22] and PG-CIDEr [17] on the Karpathy test split, and all the models are single model. In general, we can see that our models have the best performance on all metrics. Comparing our basic model 1-step-maxpro and 1-step-sample with XENT, we obtain a significant improvement on CIDEr score over XENT at a great margin from 102.1% to 115.1% and 115.4% of 1-step-maxpro and 1-step-sample respectively, since our basic models are reinforcement learning-based models and can address the exposure bias and directly optimize the evaluation metric. In particular, the 1-step-sample outperform 1-step-maxpro in terms of almost all metrics, and we can conclude that the average reward of K Monte Carlo rollouts can estimate the more precise state-action value than max-probability rollout, which leads to better performance. However, 1-step-sample need to sample K rollouts with a greater computation cost.

Regarding max-probability rollout, we compare different n-step-maxpro in Table 1. We can see that intermediate settings $n = 2, 4$ attain better overall scores than two extremes 1 and T (SCST [22]). Better performance of intermediate settings originates from the fact that they increase the absolute value of the mean of reformulated advantage while lowering variance in most time steps compared to $n = 1$, which are quantitatively shown in Fig. 3(a) & 3(b). Since rollout-based methods estimate a rough state-action value, when $n = 1$ reformulated advantage is small with large variance and it may turn out to be wrong to encourage or suppress a token in this strict greedy strategy. As n increases, the dilemma will be eased but gradually loses per-token advantage until a sequence level advantage of $n = T$. This implies intermediate n which balances the approximation of per-token advantage and the improvement of the absolute value of the mean of reformulated advantage, is always better in max-probability rollout. Moreover, different n or combining different n has different effects on balancing these two conflicts, *e.g.* the performance of $n = 2$ is better than that of $n = 4$ and close to that of 1-2-2, and 1-2-4-T and T-4-2-1 are both inferior to 1-2-2. We also show the performance curves of the Karpathy validation split during training illustrated in Fig. 2. In Fig. 2(a) & 2(b), our models have an overwhelming advantage over SCST [22] throughout the whole training process, which demonstrates that self-critical per-token advantage is better than self-critical sequence level advantage.

Regarding K Monte Carlo rollouts, 1-step-sample and 1-2-2-step-sample are superior to PG-CIDEr [17], which demonstrates that self-critical per-token advantage is better than parametrized per-token advantage in Table 1 and Fig. 2(c) & 2(d).

Comparing different effects of n-step towards max-probability rollout and K Monte Carlo rollouts, we find that large n can increase the absolute value of the mean of reformulated advantage while lowering the variance using these two kinds of rollouts in Fig. 3. However, 1-2-2-step-maxpro is superior to 1-step-maxpro and 1-2-2-step-sample is close to 1-step-sample in Table 1. Therefore, n-step ($n = 2$) is more effective in max-probability rollout than in K Monte Carlo rollouts. It is possible because degrees of change in the absolute value of the mean and the variance of reformulated advantage across different n are relatively small in K Monte Carlo rollouts and thus possibly cannot offset the lose of per-token advantage, while those are relatively large in max-probability rollout and large n (*e.g.* $n = 2$) can balance better these two conflicts as illustrated in Fig. 3.

Performance on the official MSCOCO testing server. Table 2 shows the result of our single models and 4 ensembled model using beam search with beam size set to 3 on the official MSCOCO evaluation server, and all other results are based on single model. Our single models and ensembled models outperform all of them in terms of most metrics, even the ones which use complex attention mechanisms [18, 27], and other reinforcement learning-based models which all introduce parameterized estimator [17,21,29] and

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
XENT	74.1	57.4	42.8	31.7	25.8	54.1	102.1	19.2
PG-CIDER [17]	77.44	60.66	45.85	34.32	26.35	55.61	113.9	19.25
SCST [22](T-step-maxpro)	76.83	60.65	46.05	34.61	26.65	56.03	112.7	19.99
1-step-sample	77.49	61.19	46.64	**35.08**	26.88	56.11	**115.4**	20.05
1-2-2-step-sample	77.41	61.10	46.46	34.88	26.88	56.10	114.9	20.23
1-step-maxpro	77.24	60.90	46.13	34.46	26.87	56.11	115.1	20.26
2-step-maxpro	77.82	61.30	46.45	34.80	**26.95**	**56.29**	114.6	20.35
4-step-maxpro	77.67	61.01	46.30	34.78	26.91	56.05	114.5	20.20
1-2-4-T-step-maxpro	77.45	61.02	46.25	34.59	26.89	56.26	114.8	20.38
T-4-2-1-step-maxpro	77.30	60.77	46.07	34.48	26.74	56.02	114.0	20.16
1-2-2-step-maxpro	**77.93**	**61.54**	**46.75**	34.96	26.92	56.27	115.2	**20.42**

Table 1. Performance of our proposed models versus state-of-the-art models on the test portion of the Karpathy splits using greedy search.

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Google NIC [24]	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	53.0	68.2	94.3	94.6
Hard-Attention [26]	70.5	88.1	52.8	77.9	38.3	65.8	27.7	53.7	24.1	32.2	51.6	65.4	86.5	86.3
MSRCap [7]	71.5	90.7	54.3	81.9	40.7	71.0	30.8	60.1	24.8	33.9	52.6	68.0	93.1	93.7
mRNN [19]	71.6	89.0	54.5	79.8	40.4	68.7	29.9	57.5	24.2	32.5	52.1	66.6	91.7	93.5
ATT [27]	73.1	90.0	56.5	81.5	42.4	70.9	31.6	59.9	25.0	33.5	53.5	68.2	94.3	95.8
Adaptive [18]	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	55.0	70.5	104.2	105.9
MIXER [21]	74.7	-	57.9	-	43.1	-	31.7	-	25.8	-	54.5	-	99.1	-
PG-SPIDEr [17]	75.1	91.6	59.1	84.2	44.5	73.8	33.1	62.4	25.5	33.9	55.1	69.4	104.2	107.1
AC [29]	**77.8**	92.9	61.2	85.5	45.9	74.5	33.7	62.5	26.4	33.4	55.4	69.1	110.2	112.1
SCST-Att2in(Ens. 4) [22]	-	-	-	-	-	-	34.4	-	26.8	-	55.9	-	112.3	-
1-step-maxpro	77.1	92.5	60.6	85.1	45.8	74.9	34.1	63.5	26.6	35.2	55.6	70.0	111.1	114.0
1-step-sample	77.3	92.5	60.9	85.4	46.2	75.2	34.5	64.0	26.6	35.2	55.6	70.2	111.6	114.5
1-2-2-step-maxpro	77.4	92.9	60.9	85.6	46.0	75.2	34.3	63.7	26.7	35.2	55.8	70.0	111.3	113.5
1-2-2-step-maxpro(Ens. 4)	77.6	**93.1**	**61.3**	**86.1**	**46.5**	**76.0**	**34.8**	**64.6**	**26.9**	**35.4**	**56.1**	**70.4**	**112.6**	**115.3**

Table 2. Leaderboard of published image captioning models on the official MSCOCO evaluation server.

sequence level advantage [22].

4.5. Qualitative Analysis

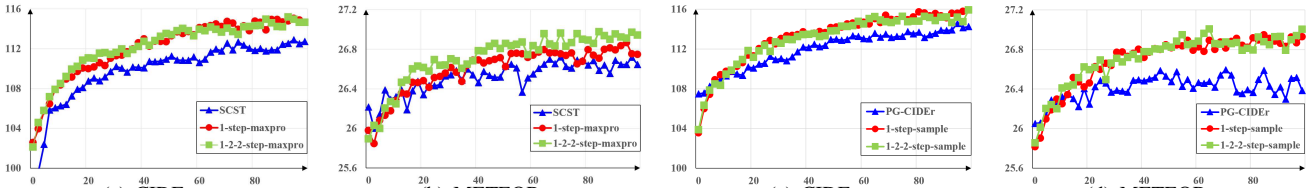
Fig. 4 shows some qualitative results of 1-step-maxpro against Ground Truth and the model trained with XENT loss. Each image has three captions from these sources listed below. In general, the captions predicted by 1-step-maxpro are better compared with the model trained with XENT loss. In Fig. 4(a), we can see that when the image content is common in the dataset and not too complex to describe, XENT and 1-step-maxpro can predict correct captions. Since the reinforcement learning-based model can avoid accumulating errors during generating the caption, the captions in Fig. 4(b)-4(e) generated by 1-step-maxpro can describe more important objects and capture their relationships with more distinctive information of the image, while those generated by XENT are less descriptive or incorrect to some degree. When a variety of human activities that appear rarely in the dataset or different activities with the same objects that are difficult to distinguish by the model, the models easily have the incorrect prediction. For example, in Fig. 4(f), 1-step-maxpro and XENT both predict wrong captions that the player in the base is throwing the ball, who in fact is catching the ball with a glove.

5. Conclusion

We reformulate advantage function to estimate per-token advantage without using parametrized estimator. Moreover, n-step reformulated advantage is proposed to increase the absolute value of the mean of reformulated advantage while lowering variance. Our methods outperform state-of-the-art methods that use the sequence level advantage and parametrized estimator on MSCOCO benchmark.

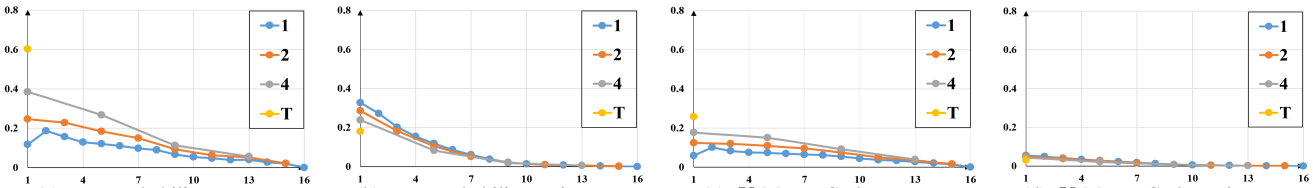
Acknowledgements

This work was supported in part by the National Key R&D Program of China (2017YFC0821005), National Basic Research Program of China (973 Program, 2015CB351800), and High-performance Computing Platform of Peking University, which are gratefully acknowledged.



(a) CIDEr (b) METEOR (c) CIDEr (d) METEOR

Figure 2. (a)(b): Performance of SCST [22], 1-step-maxpro and 1-2-2-step-maxpro; (c)(d): Performance of PG-CIDEr [17], 1-step-sample and 1-2-2-step-sample. The horizotal axes are every $2K$ training steps and the vertical axes are corresponding metrics on validation set.



(a) max-probability-mean (b) max-probability-variance (c) K-Monte-Carlo-mean (d) K-Monte-Carlo-variance

Figure 3. Mean and variance of those n-step reformulated advantage in max-probability rollout (a)(b) and K Monte Carlo rollouts (c)(d), where $n = \{1, 2, 4, T\}$. We do rollout 100 times for each state-action pair after pretraining using cross entropy loss, calculate the mean and variance of reformulated advantage, and finally average all absolute value of the mean and variance of all training data in sequence time step order, where time step $t = \{1, 2, ..., 16\}$ (horizotal axis).

a woman and a kid are standing on skis in the snow.
a woman and a child on skis in the snow.
a woman and a child standing on skis in the snow.

(a)

a box of donuts of different colors and varieties.
a box of donuts and a variety of donuts.
a box of donuts sitting on top of a table.

(b)

a wide variety of vases and chandelier in a window display.
a glass case with many different types of glass.
a display case with a bunch of vases on it.

(c)

many cars and motorcycles are parked in a parking lot.
a motorcycle parked in a parking lot next to a parking lot.
a motorcycle parked in a parking lot with a group of cars.

(d)

a helicopter is flying upwards in the sky.
a black and white photo of a black and white photo.
a black and white photo of a helicopter flying in the sky.

(e)

a man catching a baseball as another slides into the base
a baseball player is throwing a baseball
a baseball player throwing a ball on a field

(f)

Figure 4. Qualitative results of our model compared with Ground Truth and the model trained under XENT loss. Captions in black (first line), red(second line) and blue (third line) are ground truth captions, and those predicted by XENT and 1-step-maxpro respectively.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[3] 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*, pages 65–72, 2005.

[4] 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*, pages 1171–1179, 2015.

[5] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6298–6306. IEEE, 2017.

[6] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Computer Science*, 2014.

[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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015.

[8] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *European conference on computer vision*, pages 15–29. Springer, 2010.

[9] 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*, pages 770–778, 2016.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[11] A. Karpathy and F. F. Li. Deep visual-semantic alignments for generating image descriptions. In *Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

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

[13] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.

[14] A. M. Lamb, A. G. A. P. GOYAL, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609, 2016.

[15] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[16] 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*, pages 740–755. Springer, 2014.

[17] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Improved image captioning via policy gradient optimization of spider. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, page 3, 2017.

[18] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, page 2, 2017.

[19] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[20] 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*, pages 311–318. Association for Computational Linguistics, 2002.

[21] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[22] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3, 2017.

[23] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[24] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[25] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[26] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[27] 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*, pages 4651–4659, 2016.

[28] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[29] L. Zhang, F. Sung, F. Liu, T. Xiang, S. Gong, Y. Yang, and T. M. Hospedales. Actor-critic sequence training for image captioning. *arXiv preprint arXiv:1706.09601*, 2017.