Non-Autoregressive Image Captioning with Counterfactuals-Critical Multi-Agent Learning

Longteng Guo[1,2] , **Jing Liu**[1*] , **Xinxin Zhu**[1] , **Xingjian He**[1,2] , **Jie Jiang**[1,2] and **Hanqing Lu**[1]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
{longteng.guo, jliu, xinxin.zhu, xingjian.he, jie.jiang, luhq}@nlpr.ia.ac.cn

Abstract

Most image captioning models are autoregressive, *i.e.* they generate each word by conditioning on previously generated words, which leads to heavy latency during inference. Recently, non-autoregressive decoding has been proposed in machine translation to speed up the inference time by generating all words in parallel. Typically, these models use the word-level cross-entropy loss to optimize each word independently. However, such a learning process fails to consider the sentence-level consistency, thus resulting in inferior generation quality of these non-autoregressive models. In this paper, we propose a Non-Autoregressive Image Captioning (NAIC) model with a novel training paradigm: Counterfactuals-critical Multi-Agent Learning (CMAL). CMAL formulates NAIC as a multi-agent reinforcement learning system where positions in the target sequence are viewed as agents that learn to cooperatively maximize a sentence-level reward. Besides, we propose to utilize massive unlabeled images to boost captioning performance. Extensive experiments on MSCOCO image captioning benchmark show that our NAIC model achieves a performance comparable to state-of-the-art autoregressive models, while brings 13.9× decoding speedup.

1 Introduction

Image captioning [Vinyals *et al.*, 2017; Guo *et al.*, 2019c] aims at generating a natural language description of an image. Recent methods typically follow the encoder/decoder paradigm where a convolutional neural network (CNN) encodes the input image, and a sequence decoder, *e.g.* recurrent neural networks (RNNs) or Transformer [Vaswani *et al.*, 2017], generates a caption. Most of these models use *autoregressive* decoders that require sequential execution: they generate each word conditioned on the sequence of previously generated words. However, this process is not parallelizable and thus results in high inference latency, which is sometimes unaffordable for real-time industrial applications.

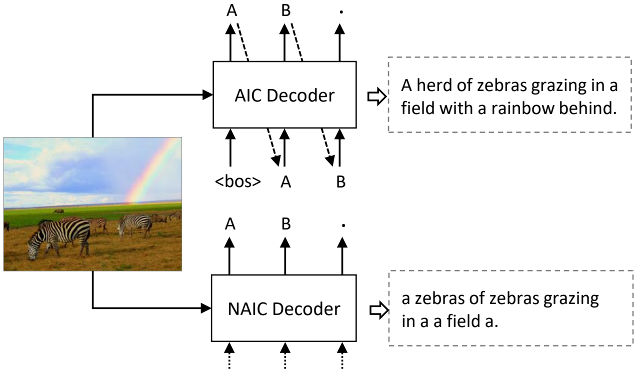

A
B
.
AIC Decoder
A herd of zebras grazing in a field with a rainbow behind.
<bos>
A
B
A
B
.
NAIC Decoder
a zebras of zebras grazing in a a field a.

Figure 1: Given an image, autoregressive image captioning (AIC) model generates a caption word by word, while Non-Autoregressive Image Captioning (NAIC) model outputs all words in parallel.

Recently, *non-autoregressive* decoding was proposed in neural machine translation (NMT) [Gu *et al.*, 2017] to significantly improve the inference speed by predicting all target words in parallel. A non-autoregressive model takes basically the same structure as the autoregressive Transformer network [Vaswani *et al.*, 2017]. But instead of conditioning the decoder on the previously generated words as in autoregressive models, they generate all words independently, as is illustrated in Figure 1. Such models are typically optimized by the cross-entropy (XE) losses of individual words.

However, existing non-autoregressive models still have a large gap in generation quality compared to their autoregressive counterparts, mainly due to their severe decoding inconsistency problem. For example, in Figure 1, the caption generated by the non-autoregressive model has repeated words and incomplete content. A major reason for such performance degradation is that the word-level XE based training objective cannot guarantee the sentence-level consistency. That is, the XE loss encourages the model to generate the golden word in each position without considering the global consistency of the whole sentence.

To simultaneously reduce the inference time and improve the decoding consistency of image captioning, in this paper, we propose a Non-Autoregressive Image Captioning (NAIC) model with a novel training paradigm: Counterfactuals-critical Multi-Agent Learning (CMAL). Specifically, we consider NAIC as a cooperative multi-agent reinforcement learning (MARL) [Buşoniu *et al.*, 2010] system, where position-

[*] Corresponding Author

s in the target sequence are viewed as "agents" that act cooperatively to maximize the quality of the whole sentence. Each agent observes the "environment" (encoded visual context), and communicates with other agents through the self-attention layers in Transformer. After several rounds of environment observation and agent communication, the agents reach an agreement about content of the target sentence and separately take "actions" to predict the words in their corresponding positions. The agents then receive a common sentence-level reward and use policy gradient to update their parameters. A benefit of this training paradigm is that the non-differentiable test metrics of image captioning could be directly optimized. Another benefit is that by optimizing the agents towards a common sentence-level objective, the decoding consistency can be substantially improved.

A crucial challenge in the above MARL training paradigm is multi-agent credit assignment [Chang *et al.*, 2004]: the shared team-reward making it difficult for each agent to deduce its own contribution to the team's success. This could impede multi-agent learning and lead to decoding inconsistency. To address this challenge, we compute an agent-specific advantage function that compares the team-reward for the joint action against an agent-wise *counterfactual baseline* [Foerster *et al.*, 2018; Chen *et al.*, 2019]. The counterfactual baseline of an agent is the expected reward when marginalizing out a single agent's action, while keeping the other agents' actions fixed. As a result, only actions from an agent that outperform the counterfactual baseline are given positive weight, and inferior actions are suppressed. CMAL fully exploits the distinctive features of the multi-agent NAIC system: extremely short episode and large action space.

To further boost captioning performance, we propose to utilize massive unlabeled images as additional data for training, which could be more easily obtained without costly human annotations. We evaluate the proposed method on the challenging MSCOCO [Chen *et al.*, 2015] image captioning benchmark. Experimental results show that our method brings $13.9\times$ decoding speedup relative to the autoregressive counterpart, while achieving comparable performance to state-of-the-art autoregressive models.

To summarize, the main contributions of this paper are three-fold:

- We propose a Non-Autoregressive Image Captioning (NAIC) model with a novel training paradigm: Counterfactuals-Critical Multi-Agent Learning. To the best of our knowledge, we are the first to formulate non-autoregressive sequence generation as a cooperative multi-agent problem.
- We design a counterfactual baseline to disentangle the individual contribution of each agent from the team-reward.
- We propose to utilize massive unlabeled data to boost the performance of non-autoregressive models.
- Our method significantly improves the inference speed of image captioning, while at the same time achieves a performance comparable to state-of-the-art autoregressive image captioning methods.

2 Related Work

Non-Autoregressive Sequence Generation. Non-Autoregressive neural machine Translation (NAT) [Gu *et al.*, 2017] has recently been introduced to speed up the inference process for real-time decoding, but often performs worse than the autoregressive counterparts. Some methods has been proposed to narrow the performance gap between autoregressive and non-autoregressive models, including knowledge distillation [Gu *et al.*, 2017], auxiliary regularization terms [Wang *et al.*, 2019], well-designed decoder input [Guo *et al.*, 2019a], iterative refinement [Lee *et al.*, 2018; Gao *et al.*, 2019] *etc*. Among them, MNIC [Gao *et al.*, 2019] and FNIC [Fei, 2019] are published works on non-autoregressive image captioning. However, these methods are trained with conventional XE loss, which is not sentence-level consistent. Unlike these works, we propose using CMAL to optimize a sentence-level objective.

Multi-Agent Reinforcement Learning (MARL). MARL [Buşoniu *et al.*, 2010] considers a system of agents that interact within a common environment. It is often designed to deal with complex reinforcement learning problems that require decentralised policies, where each agent selects its own action. Compared to well-studied MARL game tasks, our NAIC model has a much larger action space and extremely shorter episode. Our counterfactual baseline gets intuition from [Foerster *et al.*, 2018], which requires training an additional critic network to estimate the Q value for each possible action. Learning such a critic network increases the model complexity and is not practical due to the high-dimensional action space of NAIC. Instead, following [Chen *et al.*, 2019], we turn to the simple yet powerful REINFORCE [Williams, 1992] algorithm that directly uses the actual return to replace Q function.

3 Background

3.1 Autoregressive Decoding

Given an image I as input and a target sentence $y = (y_1, ..., y_T)$, AIC models are based on a chain of conditional probabilities with a left-to-right causal structure:

$$p(y|I;\theta) = \prod_{i=1}^{T} p\left(y_i | y_{<i}, I; \theta\right), \tag{1}$$

where θ is the model's parameters and $y_{<i}$ represents the words before the i-th word of target y. The inference process is not parallelizable under such autoregressive factorization as the sentence is generated word by word sequentially.

3.2 Non-Autoregressive Decoding

Recently, non-autoregressive sequence models were proposed to alleviate the inference latency by removing the sequential dependencies within the target sentence. A NAIC model generates all words independently:

$$p(y|I;\theta) = \prod_{i=1}^{T} p\left(y_i | I; \theta\right). \tag{2}$$

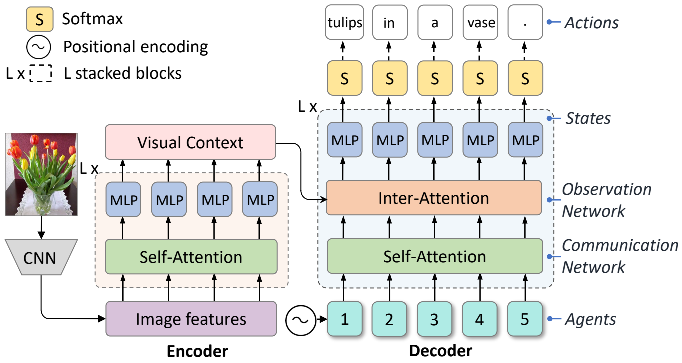


Figure 2: Illustration of our Transformer-based non-autoregressive image captioning model, which composes of an encoder and a decoder. On the rightmost, we cast the non-autoregressive decoder in the multi-agent reinforcement learning terminology.

During inference, all words could be parallelly decoded in one pass, thus the inference speed could be significantly improved.

Maximum Likelihood Training. Typically, a non-autoregressive sequence model straightforwardly adopts maximum likelihood training with a cross-entropy (XE) loss applied at each decoding position i of the sentence:

$$\mathcal{L}_{XE}(\theta) = -\sum_{i=1}^{T} \log \left(p \left(y_i | I; \theta \right) \right) \tag{3}$$

4 Approach

In this section, we first present the architecture of our NAIC model, and then introduce our Counterfactuals-critical Multi-Agent Learning (CMAL) algorithm for model optimization. Finally, we describe how we utilize unlabeled data to boost captioning performance.

4.1 Transformer-Based NAIC Model

Given the image features of an image, NAIC generates a caption about that image in a non-autoregressive manner. The architecture of our NAIC model is based on the well-known Transformer network, which composes of an encoder and decoder, as is shown in Figure 2.

Image features and encoder. Following previous works on image captioning [Anderson *et al.*, 2017], given an image, we first extract vectorial image features from a pre-trained CNN network. The encoder of NAIC is basically the same as the Transformer encoder, which takes the image features as inputs and generates the visual context representation.

Decoder. Since the sequential dependency is removed in the non-autoregressive decoder, previous works often introduce additional components *e.g.* target length predictor, well-designed decoder architecture and decoder inputs *etc.*, which adds on extra inference time. Different from these works, we choose a design that simplifies the decoder to the most degree but proves to work well in our experiment. That is, we keep the decoder architecture almost the same as the Transformer decoder, and simply use a sequence of sinusoidal positional encodings [Vaswani *et al.*, 2017] as the decoder input, each of which represents a position in the target sequence. We remove the autoregressive mask from the self-attention layers of the decoder, allowing each position in the decoder to attend over all positions in the decoder.

4.2 Counterfactuals-Critical Multi-agent Learning

NAIC as a MARL Problem

To address the decoding inconsistency problem caused by word-level XE loss and directly optimize non-differential test metrics, we formulate NAIC model as a fully cooperative Multi-Agent Reinforcement Learning (MARL) system. We now formally cast NAIC in the MARL terminology.

Agent. Each word position in the target sequence is viewed as an agent that interacts with a common "environment" (visual context from the encoder output) and other agents. There are N agents in total, identified by $a \in A \equiv \{1, \ldots, N\}$.

State. The hidden states in our NAIC decoder layers naturally represent the states of the agents, which are updated in each decoder layer. The agents observe the "environment" through the inter-attention layer where they attend to the visual context, and communicate with other agents through the self-attention layer where the messages are passed between every two agents. After L rounds of observation and communication, the final state of each agent is denoted as s_a.

Action. After obtaining s_a, each agent simultaneously chooses an action $u_a \in U$, which is a word from the whole vocabulary U. The actions of all agents form a joint action $\mathbf{u} \in \mathbf{U} \equiv U^N$. To transform the joint action into a sentence, we truncate the word sequence at the first period.

Policy. The parameters of the network, θ, define a stochastic policy π_a for each agent, from which the action is sampled, *i.e.* $u_a \sim \pi_a = \mathit{softmax}(s_a)$. We speed learning and reduce model complexity by sharing parameters among agents.

Reward. After all agents take their actions (words), they receive a shared "team-reward" $R(\mathbf{u})$. The reward is computed with an evaluation metric (*e.g.* CIDEr) by comparing the generated sentence to corresponding ground-truth sequences.

Compared to typical MARL applications, NAIC has a much larger action space (*i.e.* the whole vocabulary, which is near 10,000 words), and extremely shorter episode (*i.e.* the episode length is 1). Actually, agents in NAIC perform a one-step Markov Decision Process (MDP) since all words are generated in one-pass. We denote joint quantities over all agents in bold, *e.g.* $\mathbf{u}$, $\boldsymbol{\pi}$.

Multi-Agent Policy Gradient

The goal of multi-agent learning is to maximize the expected team-reward. With the policy gradient theorem, the expected gradient of the agents can be computed as follows:

$$\nabla_\theta \mathcal{L}(\theta) = -\mathbb{E}_{\boldsymbol{\pi}} \left[\sum_a R(\mathbf{u}) \nabla_\theta \log \pi_a \left(u_a | s_a; \theta \right) \right]. \tag{4}$$

Particularly, using the REINFORCE [Williams, 1992] algorithm, the above equation can be approximated using a single sampling $\mathbf{u} \sim \boldsymbol{\pi}$ from the agents:

$$\nabla_\theta \mathcal{L}(\theta) \approx \sum_a R(\mathbf{u}) \nabla_\theta \log \pi_a \left(u_a | s_a; \theta \right). \tag{5}$$

However, such a gradient estimate suffers from high variance, which leads to unstable and slow learning of the optimal policy. To reduce the variance, a reference reward or *baseline* b can be subtracted from the reward:

$$\nabla_\theta \mathcal{L}(\theta) \approx \sum_a (R(\mathbf{u}) - b) \nabla_\theta \log \pi_a (u_a | s_a; \theta) \,. \tag{6}$$

The baseline still leads to an unbiased estimate, and importantly, it results in lower variance of the gradient estimate [Sutton and Barto, 1998]. The baseline can be an arbitrary function, as long as it does not depend on the action u_a.

Counterfactual Baseline

The above approach, however, fails to address a key multi-agent credit assignment problem. That is, because each agent receives the same team-reward, it is unclear how a specific agent's action contributes to that team-reward. The consequences of this problem are inefficient multi-agent learning and decoding inconsistency. For example, suppose there is a generated sentence (joint action), "a girl girl riding a bike", and it gets a relatively high reward, then the word "girl" taken by the 3rd agent is likely to be pushed up because it receives a positive reward, which, however, should actually be suppressed and replaced with "is".

To address this problem, we decide to compute a separate advantage function $A_a(s_a, \mathbf{u})$ for each agent. It is computed by subtracting an agent-specific *counterfactual baseline* $B_a(s_a, \mathbf{u}_{-a})$ from the common team-reward, *i.e.*:

$$A_a(s_a, \mathbf{u}) = R(\mathbf{u}) - B_a(s_a, \mathbf{u}_{-a}), \tag{7}$$

where $\mathbf{u}_{-a}$ denotes the joint action of all the agents other than agent a. $A_a(s_a, \mathbf{u})$ measures the increase (or decrease) in expected return of a joint action $\mathbf{u}$ due to agent a having chosen action u_a under state s_a. The gradient in Equation 6 then becomes:

$$\nabla_\theta \mathcal{L}(\theta) \approx \sum_a A_a(s_a, \mathbf{u}) \nabla_\theta \log \pi_a (u_a | s_a; \theta) \,. \tag{8}$$

Since $B_a(s_a, \mathbf{u}_{-a})$ does not depend on the action of agent a, as described above, it will not change the expected gradient.

Formally, the counterfactual baseline B_a is calculated by marginalizing the rewards when agent a traverses all possible actions while keeping the other agents' actions $\mathbf{u}_{-a}$ fixed:

$$B_a(s_a, \mathbf{u}_{-a}) = \mathbb{E}_{u'_a \sim \pi_a} \left[R([\mathbf{u}_{-a}, u'_a]) \right] . \tag{9}$$

The key insight of using this counterfactual baseline for NAIC is that: given a sampled sequence/joint-action, if we replace the chosen word/action of a target position/agent with all possible words/actions and see how such counterfactual replacements affect the resulting reward, then the expected reward can act as a baseline to tell the actual influence of the chosen word/action. As a result, for each agent, only actions that outperform its counterfactual baseline would be pushed up, and inferior actions would be suppressed.

Because the action space of each agent is quite large, we approximate the expectation computation in the above equation by only considering k actions with the highest probability:

$$\begin{aligned} B_a(s_a, \mathbf{u}_{-a}) &\approx \sum_{u'_a \in \mathcal{T}_a} \pi'_a (u'_a | s_a; \theta) \, R([\mathbf{u}_{-a}, u'_a]), \\ \pi'_a (u_a | s_a; \theta) &= \frac{\pi_a (u_a | s_a; \theta)}{\sum_{u'_a \in \mathcal{T}_a} \pi_a (u'_a | s_a; \theta)}, \end{aligned} \tag{10}$$

where $\mathcal{T}_a$ is the set of words with top-k probabilities in π_a, and $\pi'_a (u_a | s_a; \theta)$ is the re-normalized probability for action u_a. Experimentally, we found this approximation to be quite accurate even with a relatively small k because the top-ranking words often have dominating probabilities.

Thanks to the one-step MDP nature of our NAIC model, the counterfactual replacements could be effortlessly made by simply choosing new words from $\pi_a (u_a | s_a; \theta)$, without the need for time-consuming Monte-Carlo rollouts as in common multi-step MDP problems.

4.3 Training with Unlabeled Data

We provide a solution to utilize additional unlabeled images to boost captioning performance. Specifically, we use sequence-level knowledge distillation (**KD**) [Kim and Rush, 2016] strategy, where the captions produced by a pre-trained autoregressive Transformer teacher model is considered as pseudo target captions for unlabeled images. Following previous works on NAT [Gu *et al.*, 2017], we also use this KD strategy to generate pseudo target captions for labeled images.

Before starting CMAL training, we first pre-train the NAIC model with the XE loss (Equation 3), during which we use both the labeled and unlabeled images and their corresponding *pseudo* captions as training data. Then during CMAL training (Equation 8), we use the labeled images and their *real* captions from the original dataset. There are two advantages of using real captions instead of pseudo captions for CMAL training: first, the reward computation at training time is consistent with the evaluation metric computation at test time, *i.e.* the generated caption is compared against the real captions; second, unlike previous works on NAT, the performance of our method will not be limited by that of the KD teacher model.

5 Experiments

5.1 Experimental Settings

MSCOCO dataset. MSCOCO [Chen *et al.*, 2015] is the most popular benchmark for image captioning. We use the 'Karpathy' splits [Karpathy and Feifei, 2015] that have been used extensively for reporting results in prior works. This split contains 113,287 training images with 5 captions each, and 5,000 images for validation and test splits, respectively. The vocabulary size is 9,487 words. We use the officially released MSCOCO unlabeled images as unlabeled data. To be consistent with previous works, we pre-extract image features for all the images following [Anderson *et al.*, 2017].

Evaluation metrics. We use standard automatic evaluation metrics to evaluate the quality of captions, including BLEU-1/4, METEOR, ROUGE, SPICE, and CIDEr [Chen *et al.*, 2015], denoted as B1/4, M, R, S, and C, respectively.

Models	BLEU-1	BLEU-4	METEOR	ROUGE	SPICE	CIDEr	Latency	Speedup
Autoregressive models								
NIC-v2 [Vinyals *et al.*, 2017]	/	32.1	25.7	/	/	99.8	/	/
Up-Down [Anderson *et al.*, 2017]	79.8	36.3	27.7	56.9	21.4	120.1	/	/
VSUA [Guo *et al.*, 2019b]	/	38.4	28.5	58.4	22.0	128.6	/	/
ETA† [Li *et al.*, 2019]	**81.5**	**39.3**	28.8	**58.9**	22.7	126.6	/	/
ORT† [Herdade *et al.*, 2019]	80.5	38.6	28.7	58.4	22.6	128.3	/	/
AIC† (bw = 1)	79.8	38.4	29.0	58.7	22.8	126.6	134ms	1.66×
AIC† (bw = 3)	80.3	38.9	**29.1**	**58.9**	**22.9**	**128.8**	222ms	1.00×
Non-autoregressive models								
MNIC† [Gao *et al.*, 2019]	75.4	30.9	27.5	55.6	21.0	108.1	61ms	2.80×
FNIC †[Fei, 2019]	/	36.2	27.1	55.3	20.2	115.7	/	8.15×
Non-autoregressive models (Ours)								
NAIC-base†	60.7	15.9	18.2	45.9	11.9	60.6	**16ms**	**13.90×**
+ weight-init	62.3	17.1	19.0	46.8	12.6	64.6		
+ KD	78.5	35.3	27.3	56.9	20.8	115.5		
+ CMAL	80.3	37.3	28.1	58.0	21.8	124.0		
+ unlabel	**80.5**	**38.0**	**28.3**	**58.2**	**22.0**	**125.5**		

Table 1: Generation quality, latency, and speedup on MSCOCO dataset. "†" indicates the model is based on Transformer architecture. AIC is our implementation of the Transformer-based autoregressive model, which has the same structure as NAIC models and is used as the teacher model for KD. "/" denotes that the results are not reported. "bw" denotes the beam width used for beam search. Latency is the time to decode a single image without minibatching, averaged over the whole test split, and is tested on a GeForce GTX 1080 Ti GPU. The latency and speedup values of MNIC and FNIC are from the paper.

Implementation details. Both our NAIC and AIC models closely follow the same model hyper-parameters as Transformer-*Base* [Vaswani *et al.*, 2017] model. Specifically, the number of stacked blocks L is 6. The AIC model is trained first with XE loss and then with SCST [Rennie *et al.*, 2017]. Beam search with a beam width of 3 is used during decoding of AIC model. Our best NAIC model is trained according to the following process. We first initialize the weights of NAIC model with the pre-trained AIC teacher model. We then pre-train NAIC model with XE loss for 30 epochs. During this stage, we use a warm-up learning rate of $min(t\times10^{-4}; 3\times10^{-4})$, where t is the current epoch number that starts at 1. After 6 epochs, the learning rate is decayed by 0.5 every 3 epochs. After that, we run CMAL training to optimize the CIDEr metric for about 70 epochs. At this training stage, we use an initial learning rate of 7.5×10^{-5} and decay it by 0.8 every 10 epochs. Both training stages use Adam [Kingma and Ba, 2014] optimizer with a batch size of 50. By default, we use $k = 2$ top-ranking words in CMAL, and use $100,000$ unlabeled images for training. We use a fixed number of $N = 16$ agents because most of the captions are no longer than this length.

5.2 Results and Analysis

General comparisons. We first compare the performance of our methods against other non-autoregressive models and state-of-the-art autoregressive models. Among the autoregressive models, ETA, ORT, MNIC, FNIC, and AIC are based on similar Transformer architecture as ours, while others are based on LSTM [Hochreiter and Schmidhuber, 1997]. MNIC and FNIC are published non-autoregressive image captioning models. MNIC adopts an iterative refinement strategy, while FNIC orders words detected in the image with an RNN. As shown in Table 1, our best model (the last row) achieves significant improvements over the previous non-autoregressive models across all metrics, strikingly narrowing their performance gap between AIC from 13.1 CIDEr points down to only 3.3 CIDEr points. Furthermore, we achieve comparable performance with state-of-the-art autoregressive models. Comparing speedups, our method obtains a significant speedup of more than a factor of 10 over the autoregressive counterpart, with latency[1] reduced to about 16ms. We show the results of online MSCOCO evaluation in Table 2.

Ablation study. We conduct an extensive ablation study with the proposed NAIC model. The results are shown in the bottom of Table 1, where "NAIC-base" is the naive NAIC model trained from scratch using XE loss, "KD" represents using knowledge distillation with AIC as the teacher model, "CMAL" denotes further applying our proposed CMAL algorithm for CIDEr optimization, "unlabel" means using additional 100,000 unlabeled data during XE training, and "weight-init" denotes initializing the weights of NAIC with AIC model. We specially consider the case when not using weight-init because it may not be possible to find an autoregressive model that has the same structure as a novelly designed non-autoregressive model. We have the following observations. First, initializing NAIC model's weights with its pre-trained AIC can consistently improve the performance. Second, NAIC-base performs extremely poorly compared to AIC. Third, we see that training on the distillation data during XE training improves the CIDEr score to 115.5. However, there still remains a large performance gap between this model and the AIC teacher. Fourth, applying our CMAL training on top of the above XE trained model significantly improves the performance by 8.5 CIDEr points. Last, using additional unlabeled data for training further boosts the performance by 1.5 CIDEr points.

[1] The time for image feature extraction is not included in latency.

Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down* [Anderson *et al.*, 2017]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
VSUA [Guo *et al.*, 2019b]	79.9	94.7	64.3	88.6	49.5	79.3	37.4	68.3	28.2	37.1	57.9	72.8	123.1	125.5
ETA* [Li *et al.*, 2019]	81.2	95.0	65.5	89.0	50.9	80.4	38.9	70.2	28.6	38.0	58.6	73.9	122.1	124.4
NAIC-CMAL (Ours)	79.8	94.3	63.8	87.2	48.8	77.2	36.8	66.1	27.9	36.4	57.6	72.0	119.3	121.2

Table 2: Results on the online MSCOCO test server. $*$ denotes ensemble model.

Baseline b	B1	B4	M	R	S	C
w/o weight-init:						
XE	77.7	34.8	26.9	56.3	20.3	113.9
None	65.6	19.4	22.7	48.9	15.8	91.4
MA	75.6	28.7	24.4	53.6	17.9	103.3
SC	79.0	34.6	26.9	56.2	20.6	118.1
CF	**79.9**	**36.5**	**27.7**	**57.4**	**21.4**	**122.1**
w/ weight-init:						
XE	78.5	35.3	27.3	56.9	20.8	115.5
None	78.6	33.7	26.5	56.1	20.2	115.2
MA	79.0	34.1	26.6	56.3	20.2	116.1
SC	79.6	36.5	27.6	57.4	21.4	121.2
CF	**80.3**	**37.3**	**28.1**	**58.0**	**21.8**	**124.0**

Table 3: Comparison of using various baselines b in Equation 6. XE: the performance after pre-training with cross-entropy loss.

top-k	B1	B4	M	R	S	C
1	80.1	**37.4**	28.0	57.9	21.7	123.7
2	**80.3**	37.3	**28.1**	**58.0**	**21.8**	**124.0**
5	80.1	37.3	28.0	58.0	21.7	123.7

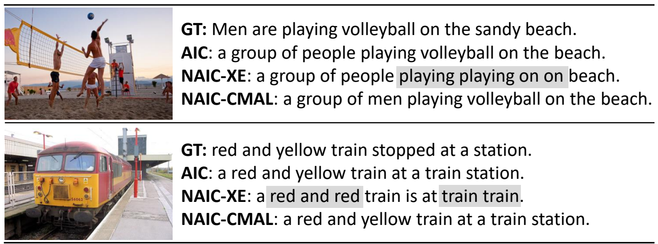
Table 4: Effect of top-k size in CMAL.

Comparison of various reward baselines b. To evaluate the effectiveness of our counterfactual (CF) baseline, we compare it with two widely-used baselines in policy gradient, *i.e.* Moving Average [Weaver and Tao, 2001] (MA) and Self-Critical [Rennie *et al.*, 2017] (SC), and not using a baseline (None), *i.e.* $b = 0$. MA baseline is the accumulated sum of the previous rewards with exponential decay. SC baseline is the received reward when all agents directly take greedy actions. As shown in Table 3, our CF baseline consistently outperforms all the other compared methods. Noteworthy that the performance gaps between our CF baseline and other baselines become larger when trainings start from a poor-performed model (*i.e.* XE model w/o weight-init). That is, our method is less sensitive to model initialization, suggesting its ability to enable more robust and stable reinforcement learning. None and MA severely degrades the performance compared to XE model when not using weight-init, but they perform similar to XE model when using weight-init. While SC considerably outperforms XE model, it is still inferior to CF. The reason is that both MA and SC are agent-agnostic global baselines, which cannot address the multi-agent credit assignment problem, while our CF baseline is agent-specific.

Effect of top-k size. As shown in Table 4, the model is not sensitive to the choice of top-k size. Using a small k of 2 could achieve fairly good performance.

#unlabel	stage	B1	B4	M	R	S	C
0	XE	78.5	35.3	27.3	56.9	20.8	115.5
	CMAL	80.3	37.3	28.1	58.0	21.8	124.0
50k	XE	78.8	36.2	27.6	57.2	21.1	118.1
	CMAL	80.2	37.6	28.1	58.1	21.9	124.8
100k	XE	79.0	36.2	27.7	57.3	21.2	118.3
	CMAL	**80.5**	**38.0**	**28.3**	**58.2**	**22.0**	**125.5**

Table 5: The results after XE and CMAL training when using different numbers of unlabeled images.



GT: Men are playing volleyball on the sandy beach.
AIC: a group of people playing volleyball on the beach.
NAIC-XE: a group of people playing playing on on beach.
NAIC-CMAL: a group of men playing volleyball on the beach.

GT: red and yellow train stopped at a station.
AIC: a red and yellow train at a train station.
NAIC-XE: a red and red train is at train train.
NAIC-CMAL: a red and yellow train at a train station.

Figure 3: Two examples of the generated captions. GT is a ground-truth caption. NAIC-XE and NAIC-CMAL are our NAIC model after XE and CMAL training, respectively. Repeated words are highlighted in gray.

Number of unlabeled images. In Table 5, we show the results after XE and CMAL training when using 0, 50,000 and 100,000 unlabeled images respectively. Generally, using more unlabeled images could lead to better performance. XE training benefits more from the unlabeled images than CMAL training because we directly use the unlabeled images during XE training while not using them for CMAL.

Qualitative analysis. We present two examples of generated image captions in Figure 3. As can be seen, repeated words and incomplete content are most prevalent in the XE trained NAIC model, showing that the word-level XE training often results in decoding inconsistency problem. With our CMAL training, the sentences become far more consistent and fluent.

6 Conclusion

We have proposed a non-autoregressive image captioning model and a novel counterfactuals-critical multi-agent learning algorithm. The decoding inconsistency problem in non-autoregressive models is well addressed by the combined effect of the cooperative agents, sentence-level team-reward, and agent-specific counterfactual baseline. The caption quality is further boosted by using unlabeled images. Results on MSCOCO image captioning benchmark show that our non-autoregressive model can achieve a performance comparable to state-of-the-art autoregressive counterparts, while at the same time enjoy $13.9\times$ inference speedup.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (No.4192059) and National Natural Science Foundation of China (No.61922086, No.61872366, and No.61872364).

References

[Anderson *et al.*, 2017] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[Buşoniu *et al.*, 2010] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*, pages 183–221. Springer, 2010.

[Chang *et al.*, 2004] Yu-Han Chang, Tracey Ho, and Leslie P Kaelbling. All learning is local: Multi-agent learning in global reward games. In *Advances in neural information processing systems*, pages 807–814, 2004.

[Chen *et al.*, 2015] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[Chen *et al.*, 2019] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4613–4623, 2019.

[Fei, 2019] Zheng-cong Fei. Fast image caption generation with position alignment. *arXiv preprint arXiv:1912.06365*, 2019.

[Foerster *et al.*, 2018] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Gao *et al.*, 2019] Junlong Gao, Xi Meng, Shiqi Wang, Xia Li, Shanshe Wang, Siwei Ma, and Wen Gao. Masked non-autoregressive image captioning. *arXiv preprint arXiv:1906.00717*, 2019.

[Gu *et al.*, 2017] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

[Guo *et al.*, 2019a] Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. Non-autoregressive neural machine translation with enhanced decoder input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3723–3730, 2019.

[Guo *et al.*, 2019b] Longteng Guo, Jing Liu, Jinhui Tang, Jiangwei Li, Wei Luo, and Lu Hanqing. Aligning linguistic words and visual semantic units for image captioning. In *ACM MM*, 2019.

[Guo *et al.*, 2019c] Longteng Guo, Jing Liu, Peng Yao, Jiangwei Li, and Hanqing Lu. Mscap: Multi-style image captioning with unpaired stylized text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4204–4213, 2019.

[Herdade *et al.*, 2019] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. *arXiv preprint arXiv:1906.05963*, 2019.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Karpathy and Feifei, 2015] Andrej Karpathy and Li Feifei. Deep visual-semantic alignments for generating image descriptions. *computer vision and pattern recognition*, pages 3128–3137, 2015.

[Kim and Rush, 2016] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*, 2016.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Lee *et al.*, 2018] Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*, 2018.

[Li *et al.*, 2019] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8928–8937, 2019.

[Rennie *et al.*, 2017] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *computer vision and pattern recognition*, 2017.

[Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[Vinyals *et al.*, 2017] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2017.

[Wang *et al.*, 2019] Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. Non-autoregressive machine translation with auxiliary regularization. *arXiv preprint arXiv:1902.10245*, 2019.

[Weaver and Tao, 2001] Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 538–545. Morgan Kaufmann Publishers Inc., 2001.

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