Semi-Autoregressive Image Captioning

Xu Yan[1,2,†], Zhengcong Fei[1,2,†], Zekang Li[1,2], Shuhui Wang[1,*], Qingming Huang[1,2,3], Qi Tian[4]
[1] Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Peng Cheng Laboratory, Shenzhen, China
[4] Cloud BU, Huawei Technologies, Shenzhen, China
{yanxu19s,feizhengcong,lizekang19g,wangshuhui}@ict.ac.cn,qmhuang@ucas.ac.cn,tian.qi1@huawei.com

ABSTRACT

Current state-of-the-art approaches for image captioning typically adopt an autoregressive manner, *i.e.*, generating descriptions word by word, which suffers from slow decoding issue and becomes a bottleneck in real-time applications. Non-autoregressive image captioning with continuous iterative refinement, which eliminates the sequential dependence in a sentence generation, can achieve comparable performance to the autoregressive counterparts with a considerable acceleration. Nevertheless, based on a well-designed experiment, we empirically proved that iteration times can be effectively reduced when providing sufficient prior knowledge for the language decoder. Towards that end, we propose a novel two-stage framework, referred to as Semi-Autoregressive Image Captioning (SAIC), to make a better trade-off between performance and speed. The proposed SAIC model maintains autoregressive property in global but relieves it in local. Specifically, SAIC model first jumpily generates an intermittent sequence in an autoregressive manner, that is, it predicts the first word in every word group in order. Then, with the help of the partially deterministic prior information and image features, SAIC model non-autoregressively fills all the skipped words with one iteration. Experimental results on the MS COCO benchmark demonstrate that our SAIC model outperforms the preceding non-autoregressive image captioning models while obtaining a competitive inference speedup. Code is available at https://github.com/feizc/SAIC.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; *Learning latent representations*; Computer vision tasks.

KEYWORDS

image captioning; semi-autoregressive decoding; Outliner and Filler; training strategies

ACM Reference Format:
Xu Yan[1,2,†], Zhengcong Fei[1,2,†], Zekang Li[1,2], Shuhui Wang[1,*], Qingming Huang[1,2,3], Qi Tian[4]. 2021. Semi-Autoregressive Image Captioning. In *Proceedings of the 29th ACM International Conference on Multimedia (MM ’21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475179

1 INTRODUCTION

Image captioning is one of the fundamental tasks in multimedia analysis and computer vision, which aims to generate a natural description for the visual content of the given image automatically. Most image captioning systems follow an encoder-decoder paradigm [2, 6, 7, 18, 30, 34, 38, 40]. Among these methods, the visual encoder, *e.g.*, convolutional neural network (CNN), first extracts features from the input image. The descriptive sentence is then decoded according to these refined features, one word at each time, using a recurrent neural network (RNN). Despite their remarkable performance, the sequential nature of text generation, *i.e.*, word by word, makes the decoding procedure not parallelizable and results in a high latency, which is challenging to work effectively in some real-time production applications [15]. Inspired from neural machine translation [14], one straight forward solution is the non-autoregressive image captioning (NAIC) [11, 14], which predicts the entire sentence in one shot. However, such a one-pass NAIC model usually lacks dependencies between words and struggles to produce smooth and accurate descriptions.

Recent studies show that extending one-pass NAIC to constant multi-pass, called iterative refinement (IR-NAIC), is promising to break the captioning performance dilemma [10, 12, 19, 39]. Unlike one-pass NAIC, which outputs the total descriptions immediately, IR-NAIC takes the caption hypothesis from the previous iteration as a reference and regularly modifies the new sentence, with a masking strategy or fusion network. The refinement process is terminated until reaching a pre-defined iteration count or no changes appear in the new sentence. Compared with conventional autoregressive image captioning (AIC), IR-NAIC with three iteration steps runs an average ×3 times faster with a comparable captioning quality, as reported by [10].

However, in this paper, we highlight that decoding with multiple iterative refinements in NAIC is unnecessary when providing a good partial prior knowledge. To verify this statement, we carefully design an experiment to understand the relationship between segmental dependency context and iteration times. In practice, we first mask some proportion of words on the caption hypothesis generated by a trained AIC model with different strategies, *i.e.*, *Head, Tail, Random*, and *Group*. Then the residual segments are taken as the

[†]These authors contribute equally to this work. [*]Corresponding author.

MM ’21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475179

Autoregressive image caption	
Input	[BOS]
Step1	A
	A pizza
	…
Step9	A pizza on a pan sitting on a table

Non-autoregressive image caption	
Input	[MASK] [MASK] … [MASK] [MASK]
Step1	A pizza on a pan sitting on a table

Semi-autoregressive image caption	
Input	[BOG]
Step1	A
	A pan
	…
Step3	A pan table
Input	A [MASK] [MASK] [MASK] pan … table [MASK] [MASK] [MASK]
Step4	A pizza on a pan sitting on a table

Figure 1: Illustration of AIC, NAIC and SAIC. AIC models generate the next word conditioned on all preceding subsentence, while NAIC models output all words in parallel in one step. Comparatively, our SAIC model considers a caption as a sequence of word groups. The first word of each group is generated one-by-one while the remaining words are predicted simultaneously conditioned on both the visual representation and jumpily generated words.

input to the language decoder, aiming to measure the quality of the final output caption. Surprisingly, we observed that even masking 70% of AIC hypothesis, the remaining prior knowledge can still help one-shot NAIC model compete with the standard IR-NAIC model. This result confirms a promising improvement direction about the form of iteration.

Inspired by this, we propose a novel two-stage framework, referred to as Semi-Autoregressive Image Captioning (SAIC), which combines the advantage of both AIC and NAIC. After extracting the visual representations with the visual encoder, SAIC first utilizes an autoregressive decoder, named Outliner, to produce several partial discontinuous words in a caption. Then a non-autoregressive decoder, named Filler, predicts previously skipped words with one iteration according to the deterministic prior knowledge. Since both the Outliner and Filler share the same model architecture and parameters, SAIC does not increase extra parameters significantly. To train the SAIC model effectively and efficiently, we further propose three training techniques including group-aware sampling, curriculum learning [4] as well as hybrid knowledge distillation. Experiment results on the MS COCO dataset show that our proposed SAIC brings a consistent decoding speedup relative to the autoregressive counterpart while gains far superior performance than the state-of-the-art IR-NAIC models.

To sum up, our main contributions are as follows:

- Through a well-designed experiment, we demonstrate that the number of IR-NAIC can be significantly reduced when providing a good prior knowledge.
- We propose a new two-stage semi-autoregressive framework to take a better trade-off between caption generation speed and quality performance. To be specific, Outliner first jumpily generates a series of discontinuous words in an autoregressive manner, and Filler then pads all previously skipped words in one non-autoregressive step.
- We introduce three strategies to improve the model training procedure: group-aware sampling, curriculum learning, and hybrid knowledge distillation. Experimentally, SAIC is able to decode faster than the sequential counterpart while strikingly narrowing the performance gap.

2 BACKGROUND

2.1 Autoregressive Image Caption

Autoregressive generation is the mainstream approach in conventional image captioning, which decomposes the distribution of target sentence $P(S|I)$ into a chain of conditional probabilities in a directional manner as:

$$P(S|I) = \prod_{t=1}^{N} P(w_t|I, w_{<t}), \tag{1}$$

where $w_{<t}$ denotes the generated historical sub-sentence before time step t. In particular, beam search [33, 37] is commonly used as a heuristic search technique, because it maintains multiple hypotheses at each decoding step and leads to a satisfactory captioning performance. However, the existence of condition $w_{<t}$ requires that the AIC model have to wait for w_{t-1} to be produced before predicting current w_t, which hinders the possibility of parallel computation along with time step.

2.2 Non-autoregressive Image Caption

Regardless of its effectiveness, the sequential inference methods have two major drawbacks. One is that it cannot generate multiple words simultaneously, leading to inefficient use of parallel computing hardware such as GPUs [28]. The other is that beam search has been found to output low-quality when applied to large search spaces [33]. Non-autoregressive method is first proposed by [12, 14] to address the above issues, allowing the image captioning model to generate all target words simultaneously. NAIC replaces $w_{<t}$ with independent latent variable z to remove the sequential dependencies and rewrite Equation 1 as:

$$P(S|I) = P(N|I) \prod_{t=1}^{N} P(w_t|I, z). \tag{2}$$

Since words are generated independently from each other during the entire decoding process, it usually results in duplicated or missing words in the obtained sentences. Subsequently, researchers developed more advanced methods to enhance the modeling of z, such as reordered latent variable [11] and objective function optimization [15], but there still exists a significant performance gap between AIC and NAIC.

2.3 Iterative Refinement based Non-autoregressive Image Caption

The previous one-pass NAIC can be further boosted by introducing the multi-pass refinement mechanism [10, 12, 13]. Specifically, IR-NAIC applies a fusion function f to deal with the sentence S' produced in the preceding stage and comprehensively predict the new sentence S by:

$$P(S|I) = \prod_{t=1}^{N'} P(w_{m(t)}|I, f(S')), \tag{3}$$

where N' corresponds to the number of newly generated words in S', and $m(t)$ is the real position of t-th refined word in S. In this way, the sentence generation process of IR-NAIC can be concluded as: first, the NAIC model produces a coarse caption as the initial hypothesis and then iteratively refines it until reaches a pre-set criteria with constant steps. In this work, we utilize masked prediction [12, 19] as the representation of IR-NAIC due to its excellent performance and simplicity, where tokens are randomly masked in training but selected with low confidences during inference.

3 IS ITERATIVE REFINEMENT ALL YOU NEED?

Previous work has pointed out that heuristic NAIC with multiple iterations can accelerate the overall speed to a significant extent, however, it slows down severely in some special cases [10, 19]. Correspondingly, this section starts from theoretic computation complexity analysis of IR-NAIC, then a delicate experiment is conducted to verify the assumption that a sufficiently good input context for the language decoder can help reduce the number of iterations. Here we construct the decoder input from the caption hypothesis produced by a well-trained AIC model.

3.1 Computation Complexity Analysis of IR-NAIC

For a conventional NAIC model, we assume that the computation cost of each iteration is proportional to the size of input tensor, denoted as $(B \times M, N, H)$, where B is the batch size, M is the beam size, N is the predicted target length, and H is the network dimension. In this way, the total cost of I iteration, generally $I < N$, is $\propto I \times O(B \times M \times N \times H)$. For convenience, we omit M and H, which simplify to $I \times O(B \times N)$. Likely, the computation cost for AIC model is $N \times O(B \times 1)$. Note that only decoder with multi-head self-attention needs to consider the previous i words. Then we can define the speedup ratio as:

$$ratio = \frac{N}{I} \times \frac{O(B \times 1)}{O(B \times N)}. \tag{4}$$

Therefore, fewer iterations and faster parallel computation are the key points to IR-NAIC.

3.2 Preliminary Experimental Setting

Image Captioning Model. We adopt the official captioning model implementation[1] proposed by [19] with denoising strategy. For convenience, the regional features of images extracted from faster-rcnn [27] on the backone of ResNet-101 [17] are utilized to retrain the image captioner with the standard Transformer-base configuration [31]. Our AIC model is first trained with XE loss and then fine-tuned with SCST [29].

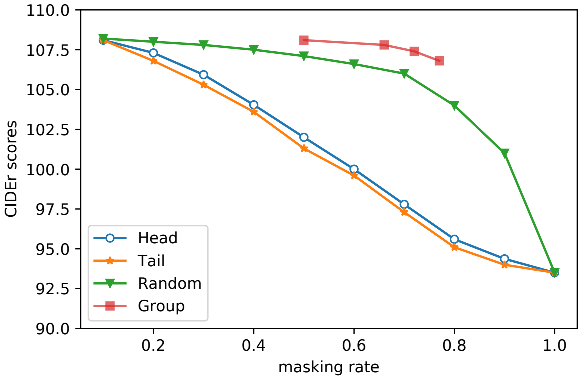


Figure 2: Performance comparison of four masking methods $\{Head, Tail, Random, Group\}$ under different masking rate in refinement experiments on MS COCO test set.

Decoding Strategy. We set the beam size of AIC model to 5 to achieve a good caption hypothesis as input to the decoder. Then we replace a certain percentage of words with `[mask]` symbol and feed the processed sentence to the iterative refinement decoder. Unlike the standard iterative refinement model that iterates several times, we only iterate fixed one step refinement with the setting of beam size being 1 and substitute all input `[mask]` symbol with the prediction of the language decoder to determine the final descriptions.

Different Masking Methods. In this work, we provide four methods to mask caption hypothesis generated from AIC: *Head*, *Tail*, *Random* and *Group*. Given the masking rate p_{mask} and the length of caption N, the number of masked words can be computed as $N_{mask} = \max\,(1, N \times p_{mask})$. Then $Head/Tail$ masking method always masks the first/last N_{mask} words in the sentence, while *Random* masks the position of caption randomly. *Group* is slightly different from the above three strategies. It first divides the sentence into G groups, where $G = \text{ceil}\,(\frac{N}{k})$ and k is the group size. Then in each group, we maintain the first word and mask the remaining k-1 words. Thus, the actual masking rate in *Group* masking can be calculated as:

$$p_{mask} = 1 - \frac{1}{k}. \tag{5}$$

To exclude the experimental randomness, we run *Random* masking method four times with different random seeds and report the average performance result.

3.3 Results Discussion

The experimental results are shown in Figure 2 and Table 1 respectively, where we can conclude that:

- **Uniform partial prior knowledge is critical.** Compared with *Head* and *Tail* masking, it is obvious that both *Random* and *Group* can gain a better captioning performance. We attribute it to the fact that the prior knowledge as input to the language decoder is uniformly distributed in *Random* and *Group*, while the *Head* and *Tail* are only provided with concentrated context on one side, *i.e.*, prefix and suffix. *Group* is superior to *Random*, which indicates

[1] https://github.com/nyu-dl/dl4mt-nonauto

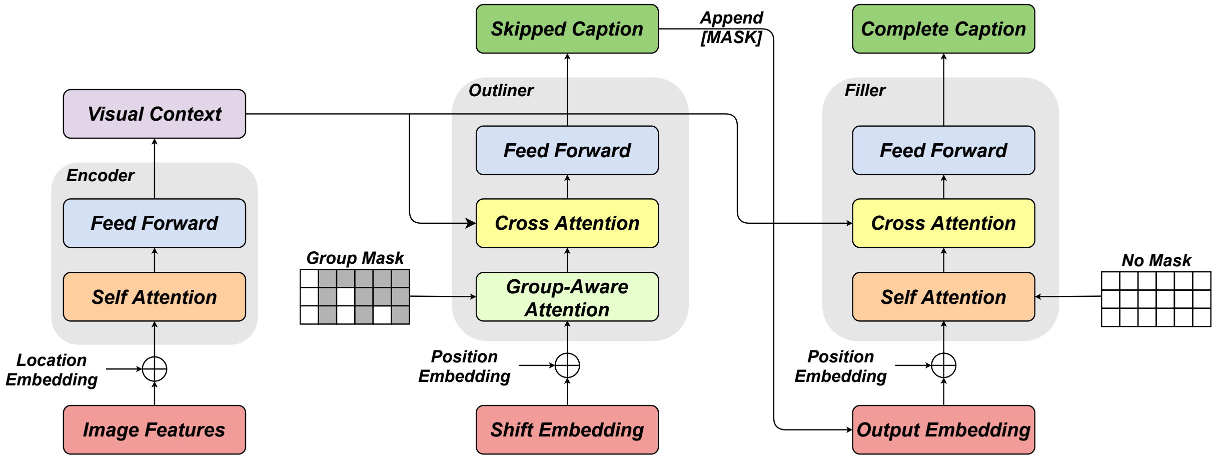


Figure 3: Overview of the proposed two-stage SAIC framework (group size $k = 2$), which consists of a visual encoder for image content, an Outliner for skipped caption and a Filler for complete caption. Those two language decoders, sharing the same architecture and parameters, build the sentence in a paradigm where the first word of each group sequentially but remaining words inside in parallel.

M	I	BLEU-4	CIDEr
1	1	31.0	108.5
5	1	31.2	109.0
1	5	31.0	108.8
5	5	31.3	109.2

Table 1: Evaluation results of different decoding parameters under *Group* masking strategy on MS COCO test set. M denotes the beam size, I is the fixed iteration times and p_{mask} is set to 0.7. The metric value does not fluctuate significantly.

that the balanced distribution of deterministic word hypothesis is necessary. *Group* masking method guarantees that each word without masking can meet at least one deterministic word within the window size of k.

• **Small beam size and one iteration are sufficient.** Compared with the standard IR-NAIC model with the beam size of 5 and multiple iterative refinements, it is interesting to find that even if only 30% of the inputs to the decoder are exposed, the *Group* masking-based decoder with greedy search can achieve quite comparable performance in fixed 1 iteration.

4 APPROACH

Based on the above analysis, we believe that the iterative refinement in IR-NAIC is unnecessary. In other words, we can obtain a high-quality caption in one shot without going through multi-pass modification under group-aware prior knowledge. In this end, we propose a two-stage framework, referred to as Semi-Autoregressive Image Captioning (SAIC). Briefly speaking, SAIC autoregressively generates a discontinuous sequence with group size k (stage I), and then fills the remaining words (stage II) in one neural network via a non-autoregressive manner. Note that the standard AIC can be regarded as the special case of SAIC when group size equals to 1 and without stage II.

4.1 Model Architecture

Overview. Generally, our SAIC model consists of three components, a visual encoder, and two language decoders: Outliner for stage I, and Filler for stage II, as displayed in Figure 3. All components adopt the Transformer architecture [31] containing self-attention sublayers and feedforward sublayers. Particularly, additional cross-attention sublayers are added to the two decoders. Furthermore, all of the sublayers are followed by the residual connection and a layer normalization operation. Note that Outliner and Filler have the same network structure and *share* the parameters, so the number of parameters remains the same as that of standard AIC and NAIC models.

Differences from Previous Works. Our network architecture takes inspiration from the prior work [19] and incorporates three key differences with respect to all previous NAIC algorithms as:

(1) The major difference lies in the *masking strategies* in the self-attention sublayer. The Outliner masks future words discontinuously and causally guarantees a strict left-to-right generation, while the Filler eliminates this limitation to leverage a bi-directional context [9], as the toy examples of group mask matrix and no mask matrix shown in the Figure 3.

(2) Our self-attention layer additionally equips with *relative position representation* (RPR) [8] to enable the language decoders to capture the sequence relationship between words easily and effectively. To be specific, the self-attention layer of the language

decoders with RPR can be represented by:

$$O_i = \mathrm{softmax}(\frac{Q_i(K^T + R_i)}{\sqrt{d_k}})(V + R_i), \quad (6)$$

$$R_i = \mathrm{Emb}(\mathrm{clip}(i, 1), \ldots, \mathrm{clip}(i, N)), \quad (7)$$

$$\mathrm{clip}(i, j) = \max(w, \min(w, j - i)), \quad (8)$$

where Q, K and V corresponds to query, key, and value matrices, d_k is the dimension of the key, R_i denotes the relative position embedding matrix, and w denotes the window size, which can also be regarded as the group size k.

(3) Most previous NAIC methods need to train the captioning model with an extra independent *length predictor* [11, 12, 19]. However, such a length predictor is implicitly modeled in our Outliner-Filler module because the length of captions is a by-product of autoregressive Outliner, *i.e.*, $N_{fill} = k \times N_{out}$, where N_{out} is the length of sequence produced by Outliner and N_{fill} is the final sequence length from Filler. Another bonus is that we can avoid carefully tuning the weighting coefficient between the length loss and the word prediction loss for performance optimization.

4.2 Training Strategies

Directly training our proposed SAIC model is not trivial since a single SAIC model needs to learn to generate captions in both autoregressive and non-autoregressive manners. This section will introduce three training strategies in detail.

Group-Aware Sampling. Compared with conventional AIC, the length of the caption generated by our Outliner shrinks the caption length from N_{fill} to $N_{out} = \frac{N_{fill}}{k}$. Meanwhile, the masking method in Outliner, referred to as group masking, as illustarted in Figure 3, is deterministic. That is, all non-first words are masked in each group. Compared with previous random masking, the training sampling method of our SAIC model is potentially group-aware.

Competence-Aware Curriculum Learning. Jointly training of Outliner and Filler is problematic since the group-aware training sample cannot make full use of the information about all the words in the sentences. In this work, we propose to gradually transfer from join training {AIC, NAIC} to {Outliner, Filler} with competence-aware curriculum learning [26]. That is, the captioning model is trained from group size $N_g = 1$ to group size $N_g = k$. More concretely, given a batch of original image-text pairs $(I_i, S_i)_{i=1}^{B}$, we first let the proportion of group size k in the Batch be $p_g = 0$ and construct the training samples of AIC and NAIC models for all pairs. And then we gradually increase the group rate p_g to introduce more group-aware learning signals for Outliner and Filler until $p_g = 1$. In implementation, we schedule p_g as follows:

$$p_g = (\frac{t}{T})^{\lambda}, \quad (9)$$

where t and T are the current and total training steps, respectively. λ is a hyperparameter indicating the degree of change and we utilize $\lambda = 1$ to increase the curriculum difficulty p_g linearly.

Hybrid Knowledge Distillation. In general, the NAIC model usually train the data generated from a teacher AIC model as knowledge distillation due to the smooth data distribution [16, 21]. However, making full use of distillation data may miss the diversity of the raw data. To combine the advantages of both distillation and raw data, we propose a simple yet effective approach – *hybrid knowledge distillation*. Specifically, hybrid knowledge distillation randomly samples a target sentence from the raw version S with a probability p_{hybr} or its distillation version S^* with a probability $1 - p_{hybr}$ during the entire training stage.

Algorithm 1: Semi-Autoregressive Image Captioning Training Algorithm

```
Input: Training data $D$ including distillation targets,
       pretrained AIC model $P_{AIC}$, group size $k$, hybrid
       distillation rate $p_{hybr}$
Output: Semi-Autoregressive Image Captioning model
        $P_{SAIC}$
1  ▷ Fine-tune on pre-trained AIC model;
2  $P_{SAIC} \leftarrow P_{AIC}$;
3  for each training batch $(I_i, S_i, S_i^*)_{i=1}^{B}$ do
4      ▷ Hybrid distillation;
5      for each sentence $S_i$ do
6          Sampling $\hat{S_i} \sim \{S_i, S_i^*\}$ with probability $p_{hybr}$;
7      end
8      ▷ Curriculum learning;
9      Get the group-aware proportion $p_g$;
10     Split the batch for training into {AIC, NAIC} and
         {Outliner, Filler} proportionally;
11     Joint optimization for $P_{SAIC}$;
12 end
```

Complete Training Algorithm. Algorithm 1 describes the procedure of training a SAIC model. To be specific, the SAIC is first initialized by a pre-trained AIC model (Line 2). Then, the training batch randomly selects a raw sentence S_i or its distilled version S_i^* based on probability p_{hybr} (Line 5-7). Next, according to the Equation 9, we can divide the batch data into two parts: conventional {AIC, NAIC} batch B_c and group-aware {Outliner, Filler} B_g, where $|\frac{B_c}{B_g}| = p_g$ (Line 10). We can construct four kinds of training samples based on corresponding data. Finally, we collect all training samples together and accumulate their gradients to update the SAIC model's parameters, which leads to the double of the batch size of standard training.

4.3 Inference Process

After encoding the image features, the Outliner starts from [bog] (the beginning of a group) to sequentially generate a jumped subsequence $S_{out} = \{w_1, \ldots, w_m\}$ with group size k until meeting [eos] (end of sentence). Then we construct the input of Filler by appending $k - 1$ [mask] symbols after every w_i. The final description is generated by replacing all [mask] with the predicted words from Filler with one iteration. If there are multiple [eos] existing, we truncate the sentence where the first [eos] appeared. Note that the beam size M_{out} in Outliner can be different from the beam size M_{fill} in Filler subject to: $M_{out} \geq M_{fill}$. In particular, if $M_{out} > M_{fill}$, then we only feed the Filler with the top M_{fill} hypothesis. Finally,

Models	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	Latency	SpeedUp
Autoregressive Image Captioning models								
NIC-v2 [34]	-	32.1	25.7	-	99.8	-	-	-
Up-Down [2]	79.8	36.3	27.7	56.9	120.1	21.4	-	-
AoANet† [18]	80.2	38.9	29.2	58.8	129.8	22.4	-	-
M2-T†[7]	80.8	39.1	29.2	58.6	131.2	22.6	-	-
AIC†	80.2	38.7	28.8	58.2	128.7	22.0	185ms	1.00×
Non-Autoregressive Image Captioning models								
MNIC†¶ [12]	75.4	30.9	27.5	55.6	108.1	21.0	-	2.80×
FNIC† [11]	-	36.2	27.1	55.3	115.7	20.2	-	8.15×
MIR†¶ [19]	-	32.5	27.2	55.4	109.5	20.6	-	1.56×
CMAL† [15]	80.3	37.3	28.1	58.0	124.0	21.8	-	13.90×
IBM†¶ [10]	77.2	36.6	27.8	56.2	113.2	20.9	-	3.06×
Semi-Autoregressive Image Captioning models								
SAIC† (M_{out}= 5, M_{fill}= 5)	80.4	38.7	29.4	58.5	128.3	22.2	119ms	1.55×
SAIC† (M_{out}= 5, M_{fill}= 1)	80.4	38.6	29.3	58.3	127.8	22.1	101ms	1.83×
SAIC† (M_{out}= 1, M_{fill}= 1)	80.3	38.4	29.0	58.1	127.1	21.9	54ms	3.42×

Table 2: Performance comparisons of different captioning models using different evaluation metrics on the MS COCO Karpathy test set. All values except Latency and SpeedUp are reported as a percentage (%). "†" denotes the model is based on Transformer architecture. AIC is our implementation of autoregressive teacher model, which has the same structure as SAIC. The SpeedUp values of NAIC models are from the corresponding papers. ¶ represents the NAIC model is iterative refinement-based.

Method	Steps	Computing Cost
AIC	N	$\sum_{i=1}^{N} Q(i)$
NAIC	1	$Q(1)$
IR-NAIC	I	$I \times Q(1)$
SAIC	$\frac{N}{k} + 1$	$\sum_{i=1}^{N/k} Q(i) + Q(1)$

Table 3: Computation complexity analysis for different caption decoding methods. $Q(i)$ denotes the computation cost in autoregressive mode when producing the i-th word. Normally, I = 3 ∼ 6 and k is set to 3.

we select the caption hypothesis with the highest score as:

$$score(\hat{S}) = \underbrace{\sum_{i=1}^{m} p(z_i|I, w_{<i})}_{\text{Outliner}} + \underbrace{\sum_{i=0}^{m-1}\sum_{j=1}^{k-1} p(w_{i\times k+j}|I, S_{out})}_{\text{Filler}}, \tag{10}$$

where $z_i = w_{i\times k}$. Due to the joint training of group size 1 and k simultaneously, SAIC model can also behave like a standard AIC model by forcing decoding with group size fixed to 1. In this way, we can only use the Outliner to generate the entire sequence without the help of Filler. Thus, AIC $k = 1$ can be regarded as a special case.

4.4 Computation Complexity Comparison

We also provide a theoretic complexity comparison with AIC, one-shot NAIC, IR-NAIC, and our proposed SAIC in Table 3. Although both SAIC and AIC models contain a slow generation process, the length of SAIC is k times shorter than conventional AIC. Considering that the computational complexity of self-attention is quadratic with its length, SAIC can save more time during the sequence inference. On the other hand, thanks to the Outliner provides a high-quality semantic context, SAIC does not need to employ a large beam size and multiple iterations like IR-NAIC to maintain a good captioning quality. Experimental results also illustrate that our light Filler can compensate for the extra computation cost in Outliner effectively and can achieve stable acceleration.

5 EXPERIMENTS

5.1 Experimental Preparation

Dataset. MS COCO [5] is a standard estimation benchmark for image captioning tasks. To be consistent with previous works, [7, 18], we adopted the Karpathy split [20] that contains 113,287 training images equipped with 5 human-annotated sentences each and 5,000 images for validation and test splits, respectively. We omit words that occur less than 5 times. The vocabulary size is 10,369 words. Image features are pre-extracted as [2].

Evaluation Metrics. Six metrics are utilized to comprehensively estimate the model performance: BLEU@N [25], METEOR [23], ROUGE-L [24], CIDEr-D [32], SPICE [1] and Latency [11, 15]. Concretely, SPICE focuses on semantic analysis and has a higher correlation with human judgment, and other metrics except Latency favor frequent n-grams and measure the overall sentence fluency. Latency is computed based on the time to decode a single sentence without mini batching, and the values are averaged over the whole off-line test set.

	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
Up-Down* [2]	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
AoANet* [18]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
M2-T* [7]	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
CMAL [15]	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
SAIC	80.0	94.5	64.1	88.2	49.2	78.8	37.2	67.8	28.0	36.8	57.7	72.4	121.4	123.7

Table 4: Leaderboard of different image captioning models on the online MS COCO test server. * denotes the ensemble model.

	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
Raw	37.5	28.1	57.3	123.3	21.3
Seq. Dist.	38.4	28.9	58.1	126.6	21.8
Hybr. Dist.	38.4	29.0	58.1	127.1	21.9

Table 5: Performance against different distillation strategies. The results show that our hybrid distillation is superior to the other two distillation strategies.

k	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
2	38.6	29.1	58.3	128.4	22.0
3	38.4	28.9	58.2	127.4	21.9
4	38.4	29.0	58.1	127.1	21.9

Table 6: Evaluation results of different group size k. The results show that with the increasing group size k, worse quality and faster speed are obtained by our SAIC model.

	SAIC	-FT	-RPR	-CL	-Hybr. Dist.
BLEU-4	38.4	38.3	38.2	38.0	37.5
CIDEr	127.1	126.8	126.5	126.0	123.3

Table 7: Evaluation results of ablation study. The results show that all introduced techniques help to improve captioning performance effectively.

Implementation Details. Our proposed SAIC model closely follows the same network architecture and hyper-parameters setting as Transformer-base model [31]. Specifically, the number of stacked blocks is 6, hidden size is 512, and feed-forward Filler size is 2048. During training, we first initialize the weights of SAIC model with the pre-trained AIC teacher model. We then train the model for 25 epochs with an initial learning rate of 3e-5 and it decays by 0.9 every 5 epochs. Adam [22] optimizer is employed. We use the group size $k = 4$, $\lambda = 1$ and $p_{hybr} = 0.5$ by default. The decoding time is measured on a single NVIDIA GeForce GTX 1080 TI as prior works reported [10, 15]. All speeds are measured by running three times and reporting the average value.

5.2 Overall Performance

Table 2 and 4 summarize the performance of both AIC and NAIC models on off-line and on-line MS COCO testing, respectively. According to the evaluation results, we can find that IR-NAIC models, *e.g.*, IBM [10], outperform those one-pass NAIC models while slow down significantly. However, our small beam variant (M_{out}= 1, M_{fill}= 1, k = 4) can defeat the existing multiple-iteration models. Also, when the beam size increases to 5, SAIC equipped with a standard Transformer achieves +1.2 CIDEr point improvement compared to the previous best results. We can easily trade-off between performance and speed by using M_{out}= 5 and M_{fill}= 1. It is somehow surprising that besides NAIC models, SAIC can even outperform the AIC models trained from scratch on some metrics. We attribute it to two reasons: (1) SAIC is fine-tuned on a well-trained AIC, making the training process easier and smoother, (2) Mixing up AIC and NAIC has a better regularization effect than training alone. Remarkably, SAIC is consistently achieving good performance with various beam search sizes. These results show that our SAIC is more promising than conventional AIC and IR-NAIC.

5.3 Model Analysis

Effect of Hybrid Distillation. To illustrate the success of incorporating hybrid distillation, we compared different distillation strategies, including raw data (Raw), sequence-level knowledge distillation (Seq. Dist.), and hybrid distillation (Hybr. Dist.) for captioning. The results are listed in Table 5. Overall, Hybr. Dist. is superior to the other two methods across the board, which indicates that training with raw data and distillation data is complementary. We also find that the performance of distillation data is higher than the raw data, which is consistent with the previous work [15].

Effect of Group Size. We also test the different group size k, ranging from {2,3,4}, and the results are reported in Table 6. Obviously, we can find that: (1) A larger k has more significant acceleration on decoding speed because fewer autoregressive steps are required; (2) As k increases, the performance of SAIC drops, *e.g.*, CIDEr with $k = 4$ is 1.3 points lower than that of $k = 2$. It illustrates that the learning difficulty of SAIC increases as providing less dependency information.

5.4 Ablation study

We also provide an entire ablation study on the MS COCO testing set. As shown in Table 7, we find that all the proposed help to improve captioning performance to some extent. In particular, the employment of hybrid distillation prevents the SAIC from overfitting and leads to +3.8 CIDEr score improvement compared to the standard distillation (-Hybr. Dist.). In addition, the other three methods, including training the SAIC model from a pre-trained AIC model (FT), using a relative position representation on decoder (RPR), and using curriculum learning (CL), can bring about 0.3 - 1.1 CIDEr score improvement, respectively.

5.5 Case Study

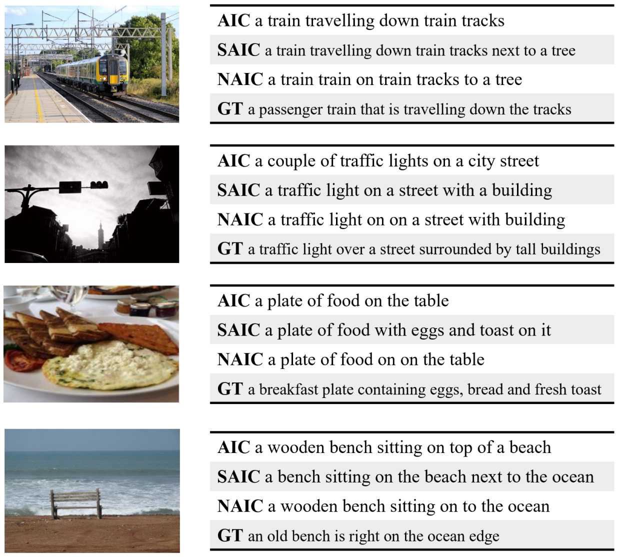
For more intuitive understanding, we present several examples of generated image captions from AIC, NAIC, and our SAIC (K = 4) models, which hold the same model architecture, coupled with human-annotated ground truth sentences (GT) in Figure 4. As we can be seen, in general, all captioning models hold the capability to reflect the content of the given image accurately. Meantime, the incoherent problem, including repeated words and incomplete content, is severe in the sentence generated by pure NAIC, while it can be effectively alleviated by SAIC, *i.e.*, two "train" terms in the first sample. This again confirms that our proposed two-stage framework, including the Outliner and Filler, can guide the captioning model to reduce word prediction errors.

5.6 Human Evaluation

Following previous works [18, 40], we conduct a human study to further compare our SAIC against two patterns, *i.e.*, AIC and NAIC. To be specific, we randomly select 200 samples from the MS COCO testing set and recruit eight workers to compare model performances. Each time, we show only one sentence paired with a corresponding image generated by different approaches or human annotation and ask: can you determine whether the given sentence has been generated by a system or a person? We then calculate the captions that pass the Turing test. The results of Human, SAIC, AIC, and NAIC are 93.6%, 82.6%, 81.1% and 65.0%, separately. It shows the superiority of SAIC in providing human-like captions.

6 RELATED WORK

State-of-the-art image captioning systems are major in autoregressive manner [3], meaning that the model generates captions word by word and is not suitable to modern hardware optimized for parallel execution. The early pioneering work about parallel generation is [41], which generated the words of selected objects first, and the rest of sentences was filled with a two-pass process. Several recent works attempt to accelerate generation by introducing a NAIC framework [14, 36], which produces the entire sentences simultaneously. Although accelerating the decoding process significantly, NAIC models suffer from repetitive and missing problems. Therefore, more efforts are devoted to mitigating those issues in the later image captioning work. Fei *et al.* [11] reorders words detected in the image with a light RNN to form better latent variables before decoding. Cho *et al.* [19] and Gao *et al.* [12] introduce an iterative mask refinement strategy to learn the position matching information. Lu *et al.* [15] addresses the inconsistency problem with a multi-agent learning paradigm and sentence-level optimization. The most relevant to our proposed method is [35]. The biggest difference lies that our captioning model changes one-step left-to-right generation to a hierarchical form, which maintains a considerable speedup and enables the caption decoder to view more abundant local history and future information to avoid errors.



AIC a train travelling down train tracks
SAIC a train travelling down train tracks next to a tree
NAIC a train train on train tracks to a tree
GT a passenger train that is travelling down the tracks

AIC a couple of traffic lights on a city street
SAIC a traffic light on a street with a building
NAIC a traffic light on on a street with building
GT a traffic light over a street surrounded by tall buildings

AIC a plate of food on the table
SAIC a plate of food with eggs and toast on it
NAIC a plate of food on on the table
GT a breakfast plate containing eggs, bread and fresh toast

AIC a wooden bench sitting on top of a beach
SAIC a bench sitting on the beach next to the ocean
NAIC a wooden bench sitting on to the ocean
GT an old bench is right on the ocean edge

Figure 4: Examples of the generated captions from AIC, SAIC, and NAIC models with the same Transformer architecture. Equipped GT represents human-annotated ground-truth captions.

7 CONCLUSION

Through a well-designed experiment, we first point out that provided a sufficient decoding prior knowledge, the number of iterations in NAIC could be dramatically reduced. Inspired by this, we propose a two-stage framework, named Semi-Autoregressive Image Captioning, to combine the advantage of AIC and NAIC. In particular, SAIC keeps the sentence outline via an autoregressive manner, and finishes the remaining filling by way of non-autoregression. To facilitate model training, we further introduce three strategies, including group-aware sampling, curriculum learning, and hybrid knowledge distillation. Compared with previously conventional baselines in autoregressive models, our SAIC model achieves a better balance between captioning quality and decoding speed. Extensive experimental results show that SAIC has an equivalent or even higher performance while maintaining a 50% faster captioning speed. For future works, we are curious about how to improve or correct the prior knowledge from the Outliner which correlates well with final outputs.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102003, in part by National Natural Science Foundation of China: 62022083, 61620106009, 61836002 and 61931008, and in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013.

REFERENCES

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic Propositional Image Caption Evaluation. In *Proc. ECCV*. 382–398.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *Proc. IEEE CVPR*. 6077–6080.

[3] Shuang Bai and Shan An. 2018. A survey on automatic image caption generation. *Neurocomputing* 311 (2018), 291–304.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proc. ICML*. 41–48.

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

[6] Xinlei Chen and C Lawrence Zitnick. 2015. Mind’s eye: A recurrent visual representation for image caption generation. In *Proc. IEEE CVPR*. 2422–2431.

[7] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-Memory Transformer for Image Captioning. In *Proc. IEEE CVPR*. 10578–10587.

[8] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* (2019).

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Zhengcong Fei. 2020. Iterative Back Modification for Faster Image Captioning. In *Proc. ACM MM*. 3182–3190.

[11] Zheng-cong Fei. 2019. Fast Image Caption Generation with Position Alignment. *arXiv preprint arXiv:1912.06365* (2019).

[12] Junlong Gao, Xi Meng, Shiqi Wang, Xia Li, Shanshe Wang, Siwei Ma, and Wen Gao. 2019. Masked Non-Autoregressive Image Captioning. *arXiv preprint arXiv:1906.00717* (2019).

[13] Lianli Gao, Kaixuan Fan, Jingkuan Song, Xianglong Liu, Xing Xu, and Heng Tao Shen. 2019. Deliberate attention networks for image captioning. In *Proc. AAAI*, Vol. 33. 8320–8327.

[14] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-Autoregressive Neural Machine Translation. In *Proc. ICLR*.

[15] Longteng Guo, Jing Liu, Xinxin Zhu, Xingjian He, Jie Jiang, and Hanqing Lu. 2020. Non-Autoregressive Image Captioning with Counterfactuals-Critical Multi-Agent Learning. *arXiv preprint arXiv:2005.04690* (2020).

[16] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. 2018. Dynamic task prioritization for multitask learning. In *Proc. ECCV*. 270–287.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. IEEE CVPR*. 770–778.

[18] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *Proc. IEEE ICCV*. 4634–4643.

[19] Lee Jason, Mansimov Elman, Graham Neubig, and Cho Kyunghyun. 2018. Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. In *Proc. EMNLP*. 1138–1149.

[20] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proc. IEEE CVPR*. 3128–3137.

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

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

[23] Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proc. ACL Workshop*. 228–231.

[24] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of summaries. In *Proc. ACL Workshops*. 74–81.

[25] Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. ACL*. 311–318.

[26] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848* (2019).

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NIPS*. 91–99.

[28] Yi Ren, Jinglin Liu, Xu Tan, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. A Study of Non-autoregressive Model for Sequence Generation. *arXiv preprint arXiv:2004.10454* (2020).

[29] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-Critical Sequence Training for Image Captioning. In *Proc. IEEE CVPR*. 1179–1195.

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NIPS*. 3104–3112.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proc. NIPS*. 5998–6008.

[32] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proc. IEEE CVPR*. 4566–4575.

[33] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424* (2016).

[34] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proc. IEEE CVPR*. 3156–3164.

[35] Chunqi Wang, Ji Zhang, and Haiqing Chen. 2018. Semi-autoregressive neural machine translation. In *Proc. EMNLP*. 479–488.

[36] Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, Jun Xie, and Xu Sun. 2019. Imitation learning for non-autoregressive neural machine translation. *arXiv preprint arXiv:1906.02041* (2019).

[37] Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960* (2016).

[38] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proc. ICML*. 2048–2057.

[39] Bang Yang, Fenglin Liu, Can Zhang, and Yuexian Zou. 2019. Non-Autoregressive Coarse-to-Fine Video Captioning. *arXiv preprint arXiv:1911.12018* (2019).

[40] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring Visual Relationship for Image Captioning. In *Proc. ECCV*. 684–699.

[41] Yue Zheng, Yali Li, and Shengjin Wang. 2019. Intention oriented image captions with guiding objects. In *Proc. IEEE CVPR*. 8395–8404.