# Deep Reinforcement Learning-based Image Captioning with Embedding Reward

Zhou Ren[1] Xiaoyu Wang[1] Ning Zhang[1] Xutao Lv[1] Li-Jia Li[2*]
[1]Snap Inc. [2]Google Inc.
{zhou.ren, xiaoyu.wang, ning.zhang, xutao.lv}@snap.com lijiali@cs.stanford.edu

## Abstract

*Image captioning is a challenging problem owing to the complexity in understanding the image content and diverse ways of describing it in natural language. Recent advances in deep neural networks have substantially improved the performance of this task. Most state-of-the-art approaches follow an encoder-decoder framework, which generates captions using a sequential recurrent prediction model. However, in this paper, we introduce a novel decision-making framework for image captioning. We utilize a "policy network" and a "value network" to collaboratively generate captions. The policy network serves as a local guidance by providing the confidence of predicting the next word according to the current state. Additionally, the value network serves as a global and lookahead guidance by evaluating all possible extensions of the current state. In essence, it adjusts the goal of predicting the correct words towards the goal of generating captions similar to the ground truth captions. We train both networks using an actor-critic reinforcement learning model, with a novel reward defined by visual-semantic embedding. Extensive experiments and analyses on the Microsoft COCO dataset show that the proposed framework outperforms state-of-the-art approaches across different evaluation metrics.*

## 1. Introduction

Image captioning, the task of automatically describing the content of an image with natural language, has attracted increasingly interests in computer vision. It is interesting because it aims at endowing machines with one of the core human intelligence to understand the huge amount of visual information and to express it in natural language.

Recent state-of-the-art approaches [3, 44, 30, 17, 7, 46, 15, 48, 43] follow an encoder-decoder framework to generate captions for the images. They generally employ convolutional neural networks to encode the visual information and utilize recurrent neural networks to decode that information to coherent sentences. During training and inference, they try to maximize the probability of the next word based on recurrent hidden state.

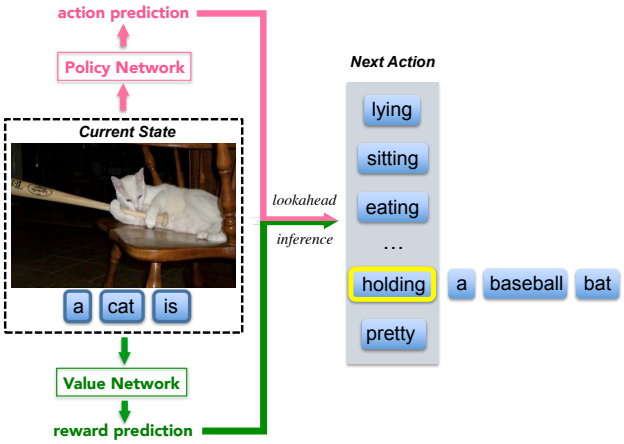


Figure 1. Illustration of the proposed decision-making framework. During our lookahead inference procedure, we utilize a "policy network" and a "value network" to collaboratively predict the word for each time step. The policy network provides an action prediction that locally predicts the next word according to current state. The value network provides a reward prediction that globally evaluates all possible extensions of the current state.

In this paper, we introduce a novel decision-making framework for image captioning. Instead of learning a sequential recurrent model to greedily look for the next correct word, we utilize a "policy network" and a "value network" to jointly determine the next best word at each time step. The policy network, which provides the confidence of predicting the next word according to current state, serves as a *local guidance*. The value network, that evaluates the reward value of all possible extensions of the current state, serves as a *global and lookahead guidance*. Such value network adjusts the goal of predicting the correct words towards the goal of generating captions that are similar to ground truth captions. Our framework is able to include the good words that are with low probability to be drawn by using the policy network alone. Figure 1 shows an example to illustrate the proposed framework. The word *holding* is not among the top choices of our policy network at current

*This work was done when the author was at Snap Inc.

step. But our value network goes forward for one step to the state supposing *holding* is generated and evaluates how good such state is for the goal of generating a good caption in the end. The two networks complement each other and are able to choose the word *holding*.

To learn the policy and value networks, we use deep reinforcement learning with embedding reward. We begin by pretraining a policy network using standard supervised learning with cross entropy loss, and by pretraining a value network with mean squared loss. Then, we improve the policy and value networks by deep reinforcement learning. Reinforcement learning has been widely used in gaming [38], control theory [32], *etc*. The problems in control or gaming have concrete targets to optimize by nature, whereas defining an appropriate optimization goal is nontrivial for image captioning. In this paper, we propose to train using an actor-critic model [21] with reward driven by visual-semantic embedding [11, 19, 36, 37]. Visual-semantic embedding, which provides a measure of similarity between images and sentences, can measure the correctness of generated captions and serve as a reasonable global target to optimize for image captioning in reinforcement learning.

We conduct detailed analyses on our framework to understand its merits and properties. Extensive experiments on the Microsoft COCO dataset [29] show that the proposed method outperforms state-of-the-art approaches consistently across different evaluation metrics, including BLEU [34], Meteor [25], Rouge [28] and CIDEr [42]. The contributions of this paper are summarized as follows:

- We present a novel decision-making framework for image captioning utilizing a policy network and a value network. Our method achieves state-of-the-art performance on the MS COCO dataset. To our best knowledge, this is the first work that applies decision-making framework to image captioning.
- To learn our policy and value networks, we introduce an actor-critic reinforcement learning algorithm driven by visual-semantic embedding. Our experiments suggest that the supervision from embedding generalizes well across different evaluation metrics.

## 2. Related Work

### 2.1. Image captioning

Many image captioning approaches have been proposed in the literature. Early approaches tackled this problem using bottom-up paradigm [10, 23, 27, 47, 24, 8, 26, 9], which first generated descriptive words of an image by object recognition and attribute prediction, and then combined them by language models. Recently, inspired by the successful use of neural networks in machine translation [4], the encoder-decoder framework [3, 44, 30, 17, 7, 46, 15, 48, 43] has been brought to image captioning. Researchers adopted such framework because "translating" an image to a sentence was analogous to the task in machine translation. Approaches following this framework generally encoded an image as a single feature vector by convolutional neural networks [22, 6, 39, 41], and then fed such vector into recurrent neural networks [14, 5] to generate captions. On top of it, various modeling strategies have been developed. Karpathy and Fei-Fei [17], Fang *et al.* [9] presented methods to enhance their models by detecting objects in images. To mimic the visual system of humans [20], spatial attention [46] and semantic attention [48] were proposed to direct the model to attend to the meaningful fine details. Dense captioning [16] was proposed to handle the localization and captioning tasks simultaneously. Ranzato *et al.* [35] proposed a sequence-level training algorithm.

During inference, most state-of-the-art methods employ a common decoder mechanism using greedy search or beam search. Words are sequentially drawn according to local confidence. Since they always predict the words with top local confidence, such mechanism can miss good words at early steps which may lead to bad captions. In contrast, in addition to the local guidance, our method also utilizes a global and lookahead guidance to compensate such errors.

### 2.2. Decision-making

Decision-making is the core problem in computer gaming [38], control theory [32], navigation and path planning [49], *etc*. In those problems, there exist agents that interact with the environment, execute a series of actions, and aim to fulfill some pre-defined goals. Reinforcement learning [45, 21, 40, 31], known as "a machine learning technique concerning how software agent ought to take actions in an environment so as to maximize some notion of cumulative reward", is well suited for the task of decision-making. Recently, professional-level computer Go program was designed by Silver *et al.* [38] using deep neural networks and Monte Carlo Tree Search. Human-level gaming control [32] was achieved through deep Q-learning. A visual navigation system [49] was proposed recently based on actor-critic reinforcement learning model.

Decision-making framework has not been applied to image captioning. One previous work in text generation [35] has used REINFORCE [45] to train its model by directly optimizing a user-specified evaluation metric. Such metric-driven approach [35] is hard to generalize to other metrics. To perform well across different metrics, it needs to be re-trained for each one in isolation. In this paper, we propose a training method using actor-critic reinforcement learning [21] driven by visual-semantic embedding [11, 19], which performs well across different evaluation metrics without re-training. Our approach shows significant performance improvement over [35]. Moreover, we use a decision-making framework to generate captions, while [35] follows the existing encoder-decoder framework.

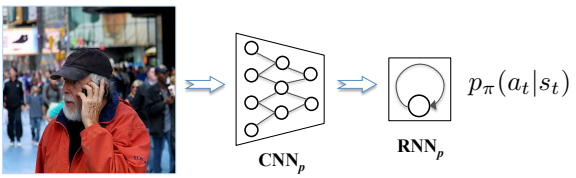


Figure 2. An illustration of our policy network $p_\pi$ that is comprised of a CNN and a RNN. The CNN$_p$ output is fed as the initial input of RNN$_p$. The policy network computes the probability of executing an action $a_t$ at a certain state $s_t$, by $p_\pi(a_t|s_t)$.

## 3. Deep Reinforcement Learning-based Image Captioning

In this section, we first define our formulation for deep reinforcement learning-based image captioning and propose a novel reward function defined by visual-semantic embedding. Then we introduce our training procedure as well as our inference mechanism.

### 3.1. Problem formulation

We formulate image captioning as a decision-making process. In decision-making, there is an *agent* that interacts with the *environment*, and executes a series of *actions*, so as to optimize a *goal*. In image captioning, the goal is, given an image $\mathbf{I}$, to generate a sentence $S = \{w_1, w_2, ..., w_T\}$ which correctly describes the image content, where $w_i$ is a word in sentence $S$ and $T$ is the length. Our model, including the policy network $p_\pi$ and value network $v_\theta$, can be viewed as the agent; the environment is the given image $\mathbf{I}$ and the words predicted so far $\{w_1, ..., w_t\}$; and an action is to predict the next word $w_{t+1}$.

#### 3.1.1 State and action space

A decision-making process consists of a series of actions. After each action $a$, a state $s$ is observed. In our problem, state $s_t$ at time step $t$ consists of the image $\mathbf{I}$ and the words predicted until $t$, $\{w_1, ..., w_t\}$. The action space is the dictionary $\mathcal{Y}$ that the words are drawn from, *i.e.*, $a_t \subset \mathcal{Y}$.

#### 3.1.2 Policy network

The policy network $p_\pi$ provides the probability for the agent to take actions at each state, $p_\pi(a_t|s_t)$, where the current state $s_t = \{\mathbf{I}, w_1, ..., w_t\}$ and action $a_t = w_{t+1}$. In this paper, we use a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) to construct our policy network, denoted as CNN$_p$ and RNN$_p$. It is similar to the basic image captioning model [44] used in the encoder-decoder framework. As shown in Figure 2, firstly we use CNN$_p$ to encode the visual information of image $\mathbf{I}$. The visual information is then fed into the initial input node $\boldsymbol{x}_0 \in \mathbb{R}^n$ of RNN$_p$. As the hidden state $\boldsymbol{h}_t \in \mathbb{R}^m$ of RNN$_p$ evolves over time $t$, the policy at each time step to take an action $a_t$ is provided. The generated word $w_t$ at $t$ will be fed back into RNN$_p$ in the next time step as the network input $\boldsymbol{x}_{t+1}$, which drives the RNN$_p$ state transition from $\boldsymbol{h}_t$ to $\boldsymbol{h}_{t+1}$. Specifically, the main working flow of $p_\pi$ is governed by the following equations:

$$\boldsymbol{x}_0 = \boldsymbol{W}^{x,v}\text{CNN}_p(\mathbf{I}) \quad (1)$$

$$\boldsymbol{h}_t = \text{RNN}_p(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) \quad (2)$$

$$\boldsymbol{x}_t = \phi(w_{t-1}), \quad t > 0 \quad (3)$$

$$p_\pi(a_t|s_t) = \varphi(\boldsymbol{h}_t) \quad (4)$$

where $\boldsymbol{W}^{x,v}$ is the weight of the linear embedding model of visual information, $\phi$ and $\varphi$ denote the input and output models of RNN$_p$.

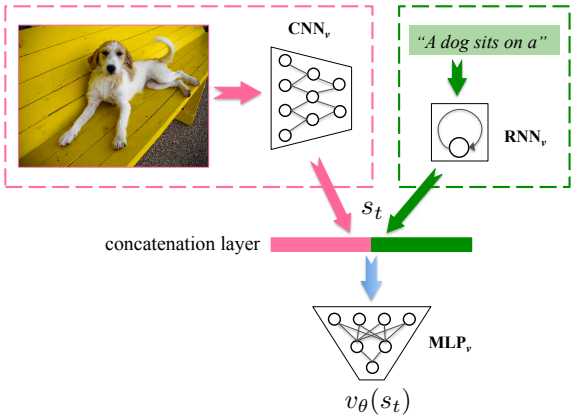


Figure 3. An illustration of our value network $v_\theta$ that is comprised of a CNN, a RNN and a MLP. Given a state $s_t$ which contains raw image input $\mathbf{I}$ and a partially generated raw sentence until $t$, the value network $v_\theta(s_t)$ evaluates its value.

#### 3.1.3 Value network

Before we introduce our value network $v_\theta$, we first define the value function $v^p$ of a policy $p$. $v^p$ is defined as the prediction of the total reward $r$ (will be defined later in Section 3.2) from the observed state $s_t$, assuming the decision-making process is following a policy $p$, *i.e.*,

$$v^p(s) = \mathbb{E}[r|s_t = s, \quad a_{t...T} \sim p] \quad (5)$$

We approximate the value function using a value network, $v_\theta(s) \approx v^p(s)$. It serves as an evaluation of state $s_t = \{\mathbf{I}, w_1, ..., w_t\}$. As shown in Figure 3, our value network is comprised of a CNN, a RNN, and a Multilayer Perceptron (MLP), denoted as CNN$_v$, RNN$_v$ and MLP$_v$. Our value network takes the raw image and sentence inputs. CNN$_v$ is utilized to encode the visual information of $\mathbf{I}$, RNN$_v$ is designed to encode the semantic information of a partially generated sentence $\{w_1, ..., w_t\}$. All the components are trained simultaneously to regress the scalar reward from $s_t$. We investigate our value network architecture in Section 4.4.

### 3.2. Reward defined by visual-semantic embedding

In our decision-making framework, it is important to define a concrete and reasonable optimization goal, *i.e.,* the *reward* for reinforcement learning. We propose to utilize visual-semantic embedding similarities as the reward.

Visual-semantic embedding has been successfully applied to image classification [11, 37], retrieval [19, 36, 33], *etc*. Our embedding model is comprised of a CNN, a RNN and a linear mapping layer, denoted as $\text{CNN}_e$, $\text{RNN}_e$ and $f_e$. By learning the mapping of images and sentences into one semantic embedding space, it provides a measure of similarity between images and sentences. Given a sentence $S$, its embedding feature is represented using the last hidden state of $\text{RNN}_e$, *i.e.*, $\boldsymbol{h}'_T(S)$. Let $\boldsymbol{v}$ denote the feature vector of image $\mathbf{I}$ extracted by $\text{CNN}_e$, and $f_e(\cdot)$ is the mapping function from image features to the embedding space. We train the embedding model using the same image-sentence pairs as in image captioning. We fix the $\text{CNN}_e$ weight, and learn the $\text{RNN}_e$ weights as well as $f_e(\cdot)$ using a bi-directional ranking loss defined as follows:

$$\begin{aligned} L_e = & \sum_{\boldsymbol{v}} \sum_{S^-} \max(0,\ \beta - f_e(\boldsymbol{v}) \cdot \boldsymbol{h}'_T(S) + f_e(\boldsymbol{v}) \cdot \boldsymbol{h}'_T(S^-)) \\ & + \sum_{S} \sum_{\boldsymbol{v}^-} \max(0,\ \beta - \boldsymbol{h}'_T(S) \cdot f_e(\boldsymbol{v}) + \boldsymbol{h}'_T(S) \cdot f_e(\boldsymbol{v}^-)) \end{aligned} \quad (6)$$

where $\beta$ is the margin cross-validated, every $(\boldsymbol{v}, S)$ are a ground truth image-sentence pair, $S^-$ denotes a negative description for the image corresponding to $\boldsymbol{v}$, and vice-versa with $\boldsymbol{v}^-$.

Given an image with feature $\boldsymbol{v}^*$, we define the reward of a generated sentence $\widehat{S}$ to be the embedding similarity between $\widehat{S}$ and $\boldsymbol{v}^*$:

$$r = \frac{f_e(\boldsymbol{v}^*) \cdot \boldsymbol{h}'_T(\widehat{S})}{\|f_e(\boldsymbol{v}^*)\| \|\boldsymbol{h}'_T(\widehat{S})\|} \quad (7)$$

### 3.3. Training using deep reinforcement learning

Following [38], we learn $p_\pi$ and $v_\theta$ in two steps. In the first step, we train the policy network $p_\pi$ using standard supervised learning with cross entropy loss, where the loss function is defined as $L_{p'} = -\log\ p(w_1, ..., w_T | \mathbf{I}; \pi) = -\sum_{t=1}^{T} \log\ p_\pi(a_t | s_t)$. And we train the value network by minimizing the mean squared loss, $||v_\theta(s_i) - r||^2$ where $r$ is the final reward of the generated sentence and $s_i$ denotes a *randomly selected* state in the generating process. For one generated sentence, successive states are strongly correlated, differing by just one word, but the regression target is shared for each entire captioning process. Thus, we randomly sample one single state from each distinct sentence, to prevent overfitting.

In the second step, we train $p_\pi$ and $v_\theta$ jointly using deep reinforcement learning (RL). The parameters of our agent are represented by $\Theta = \{\pi, \theta\}$, and we learn $\Theta$ by maximizing the total reward the agent can expect when interacting with the environment: $J(\Theta) = \mathbb{E}_{s_{1...T} \sim p_\pi}(\sum_{t=1}^{T} r_t)$. As $r_t = 0\ \forall\ 0 < t < T$ and $r_T = r$, $J(\Theta) = \mathbb{E}_{s_{1...T} \sim p_\pi}(r)$.

Maximizing $J$ exactly is non-trivial since it involves an expectation over the high-dimensional interaction sequences which may involve unknown environment dynamics in turn. Viewing the problem as a partially observable Markov decision process, however, allows us to bring techniques from the RL literature to bear: As shown in [45, 40, 31], a sample approximation to the gradient is:

$$\nabla_\pi J \approx \sum_{t=1}^{T} \nabla_\pi \log\ p_\pi(a_t | s_t)\ (r - v_\theta(s_t)) \quad (8)$$

$$\nabla_\theta J = \nabla_\theta v_\theta(s_t)\ (r - v_\theta(s_t)) \quad (9)$$

Here the value network $v_\theta$ serves as a moving baseline. The subtraction with the evaluation of value network leads to a much lower variance estimate of the policy gradient. The quantity $r - v_\theta(s_t)$ used to scale the gradient can be seen as an estimate of the advantage of action $a_t$ in state $s_t$. This approach can be viewed as an actor-critic architecture where the policy $p_\pi$ is the actor and $v_\theta$ is the critic.

However, reinforcement learning in image captioning is hard to train, because of the large action space comparing to other decision-making problems. The action space of image captioning is in the order of $10^3$ which equals the vocabulary size, while that of visual navigation in [49] is only 4, which indicates four directions to go. To handle this problem, we follow [35] to apply curriculum learning [1] to train our actor-critic model. In order to gradually teach the model to produce stable sentences, we provide training samples with gradually more difficulty: iteratively we fix the first $(T - i \times \Delta)$ words with cross entropy loss and let the actor-critic model train with the remaining $i \times \Delta$ words, for $i = 1, 2, ...$, until reinforcement learning is used to train the whole sentence.

### 3.4. Lookahead inference with policy network and value network

One of the key contributions of the proposed decision-making framework over existing framework lies in the inference mechanism. For decision-making problems, the inference is guided by a local guidance and a global guidance, *e.g.*, AlphaGo [38] utilized MCTS to combine both guidances. For image captioning, we propose a novel lookahead inference mechanism that combines the local guidance of policy network and the global guidance of value network. The learned value network provides a lookahead evaluation for each decision, which can complement the policy network and collaboratively generate captions.

Beam Search (BS) is the most prevalent method for decoding in existing image captioning approaches, which

stores the top-$B$ highly scoring candidates at each time step. Here $B$ is the beam width. Let us denote the set of $B$ sequences held by BS at time $t$ as $W_{\lceil t \rceil} = \{\boldsymbol{w}_{1,\lceil t \rceil}, ..., \boldsymbol{w}_{B,\lceil t \rceil}\}$, where each sequence are the generated words until then, $\boldsymbol{w}_{b,\lceil t \rceil} = \{w_{b,1}, ..., w_{b,t}\}$. At each time step $t$, BS considers all possible single word extensions of these beams, given by the set $\mathcal{W}_{t+1} = W_{\lceil t \rceil} \times \mathcal{Y}$, and selects the top-$B$ most scoring extensions as the new beam sequences $W_{\lceil t+1 \rceil}$:

$$W_{\lceil t+1 \rceil} = \underset{\boldsymbol{w}_{b,\lceil t+1 \rceil} \in \mathcal{W}_{t+1}}{\arg topB} \; S(\boldsymbol{w}_{b,\lceil t+1 \rceil}), \;\; s.t. \; \boldsymbol{w}_{i,\lceil t+1 \rceil} \neq \boldsymbol{w}_{j,\lceil t+1 \rceil}$$

where operator arg*topB* denotes the obtaining top-$B$ operation that is implemented by sorting the $B \times |\mathcal{Y}|$ members of $\mathcal{W}_{t+1}$, and $S(\cdot)$ denotes the scoring function of a generated sequence. In existing BS of image captioning, $S(\cdot)$ is the log-probability of the generated sequence. However, such scoring function may miss good captions because it assumes that the log-probability of every word in a good caption must be among top choices. This is not necessarily true. Analogously, in AlphaGo not every move is with top probability. It is beneficial to sometimes allow some actions with low probability to be selected as long as the final reward is optimized.

To this end, we propose a lookahead inference that combines the policy network and value network to consider all options in $\mathcal{W}_{t+1}$. It executes each action by taking both the current policy and the lookahead reward evaluation into consideration, *i.e.*,

$$\begin{aligned} S(\boldsymbol{w}_{b,\lceil t+1 \rceil}) &= S(\{\boldsymbol{w}_{b,\lceil t \rceil}, w_{b,t+1}\}) \\ &= S(\boldsymbol{w}_{b,\lceil t \rceil}) + \lambda \log p_\pi(a_t|s_t) + (1-\lambda)\, v_\theta(\{s_t, w_{b,t+1}\}) \end{aligned} \tag{10}$$

where $S(\boldsymbol{w}_{b,\lceil t+1 \rceil})$ is the score of extending the current sequence $\boldsymbol{w}_{b,\lceil t \rceil}$ with a word $w_{b,t+1}$, $\log p_\pi(a_t|s_t)$ denotes the confidence of policy network to predict $w_{b,t+1}$ as extension, and $v_\theta(\{s_t, w_{b,t+1}\})$ denotes the evaluation of value network for the state supposing $w_{b,t+1}$ is generated. $0 \leq \lambda \leq 1$ is a hyperparameter combining policy and value network that we will analyze experimentally in Section 4.5.

# 4. Experiments

In this section, we perform extensive experiments to evaluate the proposed framework. All the reported results are computed using Microsoft COCO caption evaluation tool [2], including the metrics BLEU, Meteor, Rouge-L and CIDEr, which are commonly used together for fair and thorough performance measure. Firstly, we discuss the dataset and implementation details. Then we compare the proposed method with state-of-the-art approaches on image captioning. Finally, we conduct detailed analyses on our method.

## 4.1. Dataset and implementation details

**Dataset** We evaluate our method on the widely used MS COCO dataset [29] for the image captioning task. For fair comparison, we adopt the commonly used splits proposed in [17], which use 82,783 images for training, 5,000 images for validation, and 5,000 images for testing. Each image is given at least five captions by different AMT workers. We follow [17] to preprocess the captions (*i.e.* building dictionaries, tokenizing the captions).

**Network architecture** As shown in Figure 2 and 3, our policy network, value network both contain a CNN and a RNN. We adopt the same CNN and RNN architectures for them, but train them independently. We use VGG-16 [39] as our CNN architecture and LSTM [14] as our RNN architecture. The input node dimension and the hidden state dimension of LSTM are both set to be 512, *i.e.*, $m = n = 512$. There are many CNN, RNN architectures in the literature, *e.g.*, ResNet [12], GRU [5], *etc*. Some of them have reported better performance than the ones we use. We do not use the latest architecture for fair comparison with existing methods. In our value network, we use a three-layer MLP that regresses to a scalar reward value, with a 1024-dim and a 512-dim hidden layers in between. In Figure 3, a state $s_t$ is represented by concatenating the visual and semantic features. The visual feature is a 512-dim embedded feature, mapped from the 4096-dim $\text{CNN}_v$ output. The semantic feature is the 512-dim hidden state of $\text{RNN}_v$ at the last time step. Thus, the dimension of $s_t$ is 1024.

**Visual-semantic embedding** Visual-semantic embedding can measure the similarity between images and sentences by mapping them to the same space. We followed [19] to use VGG-16 [39] as $\text{CNN}_e$ and GRU [5] as $\text{RNN}_e$. The image feature $\boldsymbol{v}$ in Equation 6 is extracted from the last 4096-dim layer of VGG-16. The input node dimension and the hidden state dimension of GRU are set as 300 and 1024. $f_e(\cdot)$ is a $4096 \times 1024$ linear mapping layer. The margin $\beta$ in Equation 6 is set as 0.2.

**Training details** In training, we use Adam [18] algorithm to do model updating. It is worth noting that, other than using the pretrained VGG-16 model, we only use the images and captions provided in the dataset to train our networks and embedding, without any external data. We set $\Delta$ in curriculum learning as 2. In testing, a caption is formed by drawing words sequentially until a special end token is reached, using the proposed lookahead inference mechanism. We do not use ensemble of models.

## 4.2. Comparing with state-of-the-art methods

In Table 1, we provide a summary of the results of our method and existing methods. We obtain state-of-the-art performance on MS COCO in most evaluation metrics. Note that Semantic ATT [48] utilized rich extra data from social media to train their visual attribute predictor, and

| Methods | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | METEOR | Rouge-L | CIDEr |
|---|---|---|---|---|---|---|---|
| Google NIC [44] | 0.666 | 0.461 | 0.329 | 0.246 | – | – | – |
| m-RNN [30] | 0.67 | 0.49 | 0.35 | 0.25 | – | – | – |
| BRNN [17] | 0.642 | 0.451 | 0.304 | 0.203 | – | – | – |
| LRCN [7] | 0.628 | 0.442 | 0.304 | 0.21 | – | – | – |
| MSR/CMU [3] | – | – | – | 0.19 | 0.204 | – | – |
| Spatial ATT [46] | **0.718** | 0.504 | 0.357 | 0.25 | 0.23 | – | – |
| gLSTM [15] | 0.67 | 0.491 | 0.358 | 0.264 | 0.227 | – | 0.813 |
| MIXER [35] | – | – | – | 0.29 | – | – | – |
| Semantic ATT [48] * | 0.709 | 0.537 | 0.402 | **0.304** | 0.243 | – | – |
| DCC [13] * | 0.644 | – | – | – | 0.21 | – | – |
| Ours | 0.713 | **0.539** | **0.403** | **0.304** | **0.251** | **0.525** | **0.937** |

Table 1. Performance of our method on MS COCO dataset, comparing with state-of-the-art methods. Our beam size is set to 10. For those competing methods, we show the results from their latest version of paper. The numbers in bold face are the best known results and (−) indicates unknown scores. (*) indicates that external data was used for training in these methods.

| Methods | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | METEOR | Rouge-L | CIDEr |
|---|---|---|---|---|---|---|---|
| SL | 0.692 | 0.519 | 0.384 | 0.289 | 0.237 | 0.512 | 0.872 |
| SL-Embed | 0.7 | 0.523 | 0.383 | 0.280 | 0.241 | 0.514 | 0.888 |
| SL-RawVN | 0.706 | 0.533 | 0.395 | 0.298 | 0.243 | 0.52 | 0.916 |
| hid-VN | 0.603 | 0.429 | 0.292 | 0.197 | 0.2 | 0.467 | 0.69 |
| hid-Im-VN | 0.611 | 0.435 | 0.297 | 0.201 | 0.202 | 0.468 | 0.701 |
| Full-model | 0.713 | 0.539 | 0.403 | 0.304 | 0.251 | 0.525 | 0.937 |

Table 2. Performance of the variants of our method on MS COCO dataset, with beam size = 10. **SL**: supervised learning baseline. **SL-Embed**: **SL** with embedding. **SL-RawVN**: **SL** plus pretrained raw value network. **hid-VN**: value network directly utilizes policy hidden state. **hid-Im-VN**: value network utilizes policy hidden state and policy image feature. **Full-model**: our full model.

DCC [13] utilized external data to prove its unique transfer capacity. It makes their results incomparable to other methods that do not use extra training data. Surprisingly, even without external training data, our method outperforms [48, 13]. Comparing to methods other than [48, 13], our approach shows significant improvements in all the metrics except Bleu-1 in which our method ranks the second. Bleu-1 is related to single word accuracy, the performance gap of Bleu-1 between our method and [46] may be due to different preprocessing for word vocabularies. MIXER [35] is a metric-driven trained method. A model trained with Bleu-4 using [35] is hard to generalize to other metrics. Our embedding-driven decision-making approach performs well in all metrics. Especially, considering our policy network shown in Figure 2 is based on a mechanism similar to the very basic image captioning model similar to Google NIC [44], such significant improvement over [44] validates the effectiveness of the proposed decision-making framework that utilizes both policy and value networks. Moreover, the proposed framework is modular w.r.t. the network design. Other powerful mechanisms such as spatial attention, semantic attention can be directly integrated into our policy network and further improve our performance.

Since the proposed embedding-driven decision-making framework is very different from existing methods, we want to perform insightful analyses and answer the following questions: 1) How powerful is embedding? Is the performance gain more because of the framework or embedding alone? 2) How important is lookahead inference? 3) How important is reinforcement learning in the framework? 4) Why the value network is designed as in Figure 3? 5) How sensitive is the method to hyperparameter $\lambda$ and beam size? To answer those questions, we conduct detailed analyses in the following three sections.

### 4.3. How much each component contributes?

In this section, we answer questions 1) 2) 3) above. As discussed in Section 3.3, we train our policy and value networks in two steps: pretraining and then reinforcement learning. We name the initial policy network pretrained with supervised learning as (**SL**). We name the initial value network pretrained with mean squared loss as (**RawVN**). The SL model can be served as our baseline, which does not use value network or lookahead inference. To evaluate the impact of embedding, we incorporate SL with embedding as follows: in the last step of beam search of SL, when a beam of candidate captions are generated, we rank those candidates according to their embedding similarities with the test image other than their log-probabilities, and finally output the one with highest embedding score. This baseline is named as (**SL-Embed**). To validate the contribution of lookahead inference and reinforcement learning, we construct a baseline that use SL and RawVN with the proposed lookahead inference, which is named as (**SL-RawVN**). Finally our full model is named as (**Full-model**).

According to the results of those variants of our method

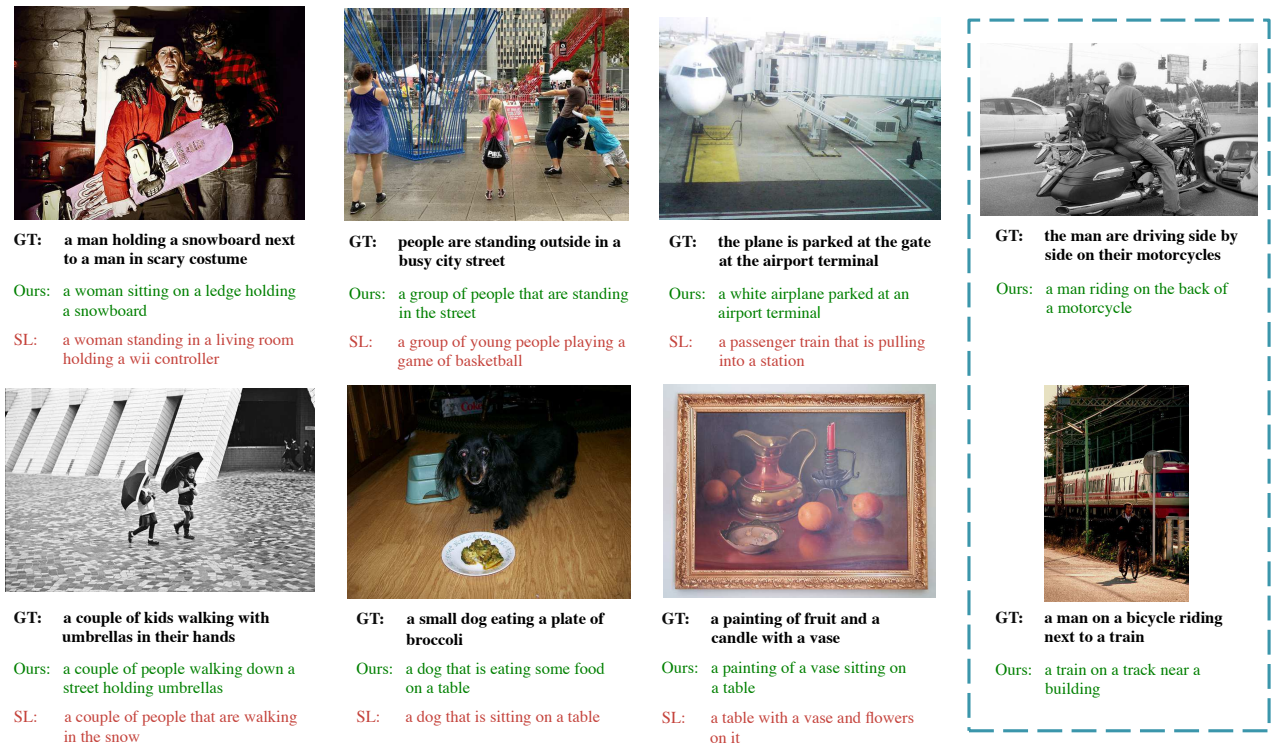


Figure 4. Qualitative results of our method and the supervised learning (SL) baseline. In the first three columns, our method generates better captions than SL. We show two failure cases in the last column. GT stands for ground truth caption.

shown in Table 2, we can answer the questions 1)-3) above:

1. Using embedding alone, **SL-Embed** performs slightly better than the **SL** baseline. However, the gap between **SL-Embed** and **Full-model** is very big. Therefore, we conclude that using embedding alone is not powerful. The proposed embedding-driven decision-making framework is the merit of our method.
2. By using lookahead inference, **SL-RawVN** is much better than the **SL** baseline. This validates the importance of the proposed lookahead inference that utilizes both local and global guidance.
3. After reinforcement learning, our **Full-model** performs better than the **SL-RawVN**. This validates the importance of using embedding-driven actor-critic learning for model training.

We show some qualitative captioning results of our method and the SL baseline in Figure 4. GT stands for ground truth caption. In the first three columns, we compare our method and SL baseline. As we see, our method is better at recognizing key objects that are easily missed by SL, *e.g.*, the *snowboard* and *umbrellas* in the first column images. Besides, our method can reduce the chance of generating incorrect word and accumulating errors, *e.g.*, we generate the word *eating* other than *sitting* for the image in the lower second column. Moreover, thanks to the global guidance, our method is better at generating correct captions at global level, *e.g.*, we can recognize the *airplane* and *painting* for the images in the third column. Finally, we show two failure cases of our method in the last column, in which cases we fail to understand some important visual contents that only take small portions of the images. This may be due to our policy network architecture. Adding more detailed visual modeling techniques such as detection and attention can alleviate such problem in the future.

### 4.4. Value network architecture analysis

In this paper we propose a novel framework that involves value network, whose architecture is worth looking into. As in Figure 3, we use $\text{CNN}_v$ and $\text{RNN}_v$ to extract visual and semantic information from the raw image and sentence inputs. Since the hidden state in policy network at each time step is a representation of each state as well, a natural question is "can we directly utilize the policy hidden state?". To answer this question, we construct two variants of our value network: the first one, named as (**hid-VN**), is comprised of a $\text{MLP}_v$ on top of the policy hidden state of $\text{RNN}_p$; the second variant, (**hid-Im-VN**), is comprised of a $\text{MLP}_v$ on top of the concatenation of the policy hidden state of $\text{RNN}_p$ and the visual input $\boldsymbol{x}_0$ of policy $\text{RNN}_p$. The results are shown in Table 2. As we see, both variants that utilize policy hidden state do not work well, comparing to our **Full-model**. The problem of the policy hidden state is that it compresses and also loses lots of information. Thus, it is reasonable and better to train independent CNN, RNN for value network it-

| $\lambda$ | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | METEOR | Rouge-L | CIDEr |
|---|---|---|---|---|---|---|---|
| 0 | 0.638 | 0.471 | 0.34 | 0.247 | 0.233 | 0.501 | 0.8 |
| 0.1 | 0.683 | 0.51 | 0.373 | 0.274 | 0.248 | 0.516 | 0.894 |
| 0.2 | 0.701 | 0.527 | 0.389 | 0.288 | 0.248 | 0.521 | 0.922 |
| 0.3 | 0.71 | 0.535 | 0.398 | 0.298 | **0.251** | 0.524 | 0.934 |
| 0.4 | **0.713** | **0.539** | **0.403** | **0.304** | 0.247 | **0.525** | **0.937** |
| 0.5 | 0.71 | 0.538 | 0.402 | 0.304 | 0.246 | 0.524 | 0.934 |
| 0.6 | 0.708 | 0.535 | 0.399 | 0.301 | 0.245 | 0.522 | 0.923 |
| 0.7 | 0.704 | 0.531 | 0.395 | 0.297 | 0.243 | 0.52 | 0.912 |
| 0.8 | 0.7 | 0.526 | 0.392 | 0.295 | 0.241 | 0.518 | 0.903 |
| 0.9 | 0.698 | 0.524 | 0.389 | 0.293 | 0.24 | 0.516 | 0.895 |
| 1 | 0.694 | 0.52 | 0.385 | 0.289 | 0.238 | 0.513 | 0.879 |

Table 3. Evaluation of hyperparameter $\lambda$'s impact on our method.

| Method | Beam size | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | METEOR | Rouge-L | CIDEr |
|---|---|---|---|---|---|---|---|---|
| SL | 5 | 0.696 | 0.522 | 0.388 | 0.29 | 0.238 | 0.513 | 0.876 |
| | 10 | 0.692 | 0.519 | 0.384 | 0.289 | 0.237 | 0.512 | 0.872 |
| | 25 | 0.683 | 0.508 | 0.374 | 0.281 | 0.234 | 0.505 | 0.853 |
| | 50 | 0.680 | 0.505 | 0.372 | 0.279 | 0.233 | 0.503 | 0.850 |
| | 100 | 0.679 | 0.504 | 0.372 | 0.279 | 0.233 | 0.503 | 0.849 |
| Ours | 5 | 0.711 | 0.538 | 0.403 | 0.302 | 0.251 | 0.524 | 0.934 |
| | 10 | 0.713 | 0.539 | 0.403 | 0.304 | 0.251 | 0.525 | 0.937 |
| | 25 | 0.709 | 0.534 | 0.398 | 0.299 | 0.248 | 0.522 | 0.928 |
| | 50 | 0.708 | 0.533 | 0.397 | 0.298 | 0.247 | 0.52 | 0.924 |
| | 100 | 0.707 | 0.531 | 0.395 | 0.297 | 0.244 | 0.52 | 0.92 |

Table 4. Evaluation of different beam sizes' impact on SL baseline and our method.

self with raw image and sentence inputs, as in Figure 3.

### 4.5. Parameter sensitivity analysis

There are two major hyperparameters in our method, $\lambda$ in Equation 10 and the beam size. In this section, we analyze their sensitivity to answer question 5) above.

In Table 3, we show the evaluation of $\lambda$'s impact on our method. As in Equation 10, $\lambda$ is a hyperparameter combining policy and value networks in lookahead inference, $0 \leq \lambda \leq 1$. $\lambda = 0$ means we only use value network to guide our lookahead inference; while $\lambda = 1$ means we only use policy network, which is identical to beam search. As shown in Table 3, the best performance is when $\lambda = 0.4$. As $\lambda$ goes down from 0.4 to 0 or goes up from 0.4 to 1, overall the performance drops monotonically. This validates the importance of both networks; we should not emphasize too much on either network in lookahead inference. Besides, $\lambda = 0$ performs much worse than $\lambda = 1$. This is because policy network provides local guidance, which is very important in sequential prediction. Thus, in lookahead inference, it is too weak if we only use global guidance, *i.e.* value network in our approach.

In Table 4, we provide the evaluation of different beam sizes' impact on SL baseline and our full model. As discovered in previous work such as [17], the image captioning performance becomes worse as the beam size gets larger. We validate such discovery for existing encoder-decoder framework. As shown in the upper half of Table 4, we test our SL baseline with 5 different beam sizes from 5 to 100. Note that SL is based on beam search, which follows the encoder-decoder framework as most existing approaches. As we see, the impact of beam size on SL is relatively big. It's mainly because that as we increase the beam size, bad word candidates are more likely to be drawn into the beam, since the confidence provided by the sequential word generator is only consider local information.

On the other hand, as shown in the lower part of Table 4, our method is less sensitive to beam sizes. The performance variations between different beam sizes are fairly small. We argue that this is because of the proposed lookahead inference that considers both policy and value networks. With local and global guidances, our framework is more robust and stable to policy mistakes.

## 5. Conclusion

In this work, we present a novel decision-making framework for image captioning, which achieves state-of-the-art performance on standard benchmark. Different from previous encoder-decoder framework, our method utilizes a policy network and a value network to generate captions. The policy network serves as a local guidance and the value network serves as a global and lookahead guidance. To learn both networks, we use an actor-critic reinforcement learning approach with novel visual-semantic embedding rewards. We conduct detailed analyses on our framework to understand its merits and properties. Our future works include improving network architectures and investigating the reward design by considering other embedding measures.

## References


[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, 2009. 4

[2] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. In *arXiv:1504.00325*, 2015. 5

[3] X. Chen and C. L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, 2015. 1, 2, 6

[4] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014. 2

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *arXiv:1412.3555*, 2014. 2, 5

[6] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: a large-scale hierachical image database. In *CVPR*, 2009. 2

[7] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 1, 2, 6

[8] D. Elliott and F. Keller. Image description using visual dependency representations. In *EMNLP*, 2013. 2

[9] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig. From from captions to visual concepts and back. In *CVPR*, 2015. 2

[10] 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 *ECCV*, 2010. 2

[11] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 2, 4

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[13] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. Deep compositional captioning: Describing novel object categories without paired training data. In *CVPR*, 2016. 6

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997. 2, 5

[15] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars. Guiding the long-short term memory model for image caption generation. In *ICCV*, 2015. 1, 2, 6

[16] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016. 2

[17] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1, 2, 5, 6, 8

[18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[19] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *TACL*, 2015. 2, 4, 5

[20] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Matters of intelligence*, pages 115–141, 1987. 2

[21] V. Konda and J. Tsitsiklis. Actor-critic algorithms. In *NIPS*, 1999. 2

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[23] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating image descriptions. In *CVPR*, 2011. 2

[24] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In *ACL*, 2012. 2

[25] A. Lavie and M. Denkowski. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 2010. 2

[26] R. Lebret, P. O. Pinheiro, and R. Collobert. Simple image description generator via a linear phrase-based approach. In *ICLR*, 2015. 2

[27] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *CoNLL*, 2011. 2

[28] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *WAS*, 2004. 2

[29] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 5

[30] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. Deep captioning with multimodal recurrent neural networks. In *ICLR*, 2015. 1, 2, 6

[31] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016. 2, 4

[32] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. 2

[33] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 4

[34] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 2

[35] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016. 2, 4, 6

[36] Z. Ren, H. Jin, Z. Lin, C. Fang, and A. Yuille. Multi-instance visual-semantic embedding. In *arXiv:1512.06963*, 2015. 2, 4

[37] Z. Ren, H. Jin, Z. Lin, C. Fang, and A. Yuille. Joint image-text representation by gaussian visual-semantic embedding. In *ACM Multimedia*, 2016. 2, 4

[38] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. 2, 4

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 5

[40] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 2000. 2, 4

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[42] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 2

[43] A. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. In *arXiv:1610.02424*, 2016. 1, 2

[44] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 1, 2, 3, 6

[45] R. Williams. simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. 2, 4

[46] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1, 2, 6

[47] Y. Yang, C. L. Teo, H. Daume III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011. 2

[48] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, 2016. 1, 2, 5, 6

[49] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, and A. Gupta. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *arXiv:1609.05143*, 2016. 2, 4