

Exact Adversarial Attack to Image Captioning via Structured Output Learning with Latent Variables

Yan Xu^{††*}, Baoyuan Wu^{††*}, Fumin Shen[‡], Yanbo Fan[†], Yong Zhang[†], Heng Tao Shen[‡], Wei Liu^{†‡}

[†]Tencent AI Lab, [‡]University of Electronic Science and Technology of China

{xuyan5533, wubaoyuan1987, fumin.shen, fanyanbo0124, zhangyong201303}@gmail.com,

shenhengtao@hotmail.com, wl2223@columbia.edu

Abstract

In this work, we study the robustness of a CNN+RNN based image captioning system being subjected to adversarial noises. We propose to fool an image captioning system to generate some targeted partial captions for an image polluted by adversarial noises, even the targeted captions are totally irrelevant to the image content. A partial caption indicates that the words at some locations in this caption are observed, while words at other locations are not restricted. It is the first work to study exact adversarial attacks of targeted partial captions. Due to the sequential dependencies among words in a caption, we formulate the generation of adversarial noises for targeted partial captions as a structured output learning problem with latent variables. Both the generalized expectation maximization algorithm and structural SVMs with latent variables are then adopted to optimize the problem. The proposed methods generate very successful attacks to three popular CNN+RNN based image captioning models. Furthermore, the proposed attack methods are used to understand the inner mechanism of image captioning systems, providing the guidance to further improve automatic image captioning systems towards human captioning.

1. Introduction

It has been shown [29] that deep neural networks (DNNs) [18] are vulnerable to adversarial images, which are visually similar to benign images. Most of these works focus on convolutional neural networks (CNNs) [17] based tasks (e.g., image classification [15, 33, 32, 31], object detection [9], or object tracking [38, 19]), of which the loss functions are factorized to independent (i.e., unstructured) outputs, so that the gradient can be easily computed to generate adversarial noises. However, if the output is structured, it may be

*indicates equal contributions. †indicates corresponding authors. This work was done when Yan Xu was an intern at Tencent AI Lab.

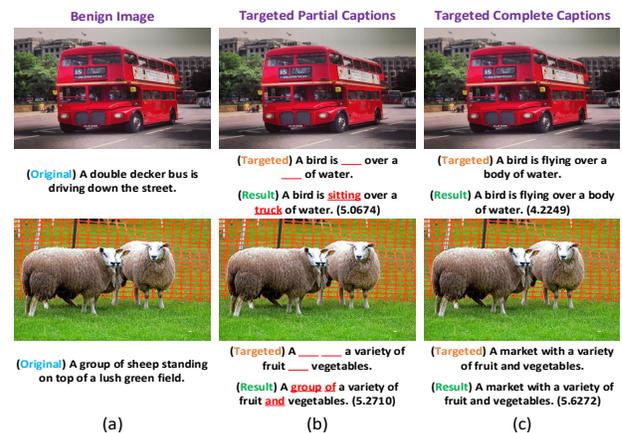


Figure 1. Examples of adversarial attacks to the image captioning model of Show-Attend-and-Tell [35], using the proposed attack methods dubbed GEM (the top row) and latent SSVMs (the bottom row), respectively. In each targeted partial caption (i.e., targeted), the red ‘_’ indicates one latent word. In each predicted caption (i.e., result), the value at the end denotes the norm of adversarial noises $\|\epsilon\|_2$. All targeted captions are successfully attacked, while adversarial noises are invisible to human perception.

difficult to derive the gradient of the corresponding structured loss. One popular deep model with structured outputs is the combination of CNNs and recurrent neural networks (RNNs) [26], where the visual features extracted by CNNs are fed into RNNs to generate a sequential output. We call this combination as a CNN+RNN architecture in this paper. One typical task utilizing the CNN+RNN architecture is image captioning [30], which describes the image content using a sentence. In this work, we present adversarial attacks to image captioning, as an early attempt of the robustness of DNNs with structured outputs.

Given a trained CNN+RNN image captioning model and an benign image, we want to fool the model to produce a targeted partial caption, which may be totally irrelevant to the image content, through adding adversarial noises to that im-

age. This task is called *exact adversarial attack of targeted partial captions*, which has never been studied in previous work. As shown in Fig. 1(b), a targeted partial caption indicates that the words at some locations are observed, while the words at other locations are not specified, *i.e.*, latent. When the words at all locations are observed, it becomes a targeted complete caption (see Fig. 1(c)). To this end, the marginal posterior probability of the targeted partial caption should be maximized, while minimizing the norm of adversarial noises. It could be formulated as a structured output learning problem with latent variables [3, 37]. Specifically, we present two formulations. One is maximizing the log marginal likelihood of the targeted partial caption, which can be optimized by the generalized expectation maximization (GEM) algorithm [4]. The other is maximizing the margin of the log marginal likelihood between the targeted partial caption and all other possible partial captions at the same locations, which can be optimized by the structural support vector machines with latent variables (latent SSVMs) [37]. Note that the proposed formulations are not coupled with any specific CNN+RNN architecture. Thus, we evaluate the proposed methods on three popular image captioning models, including Show-and-Tell [30], Show-Attend-and-Tell [35] and self-critical sequence training (SCST) utilizing reinforcement learning [24]. Experiments on MS-COCO [20] demonstrate that the proposed methods can generate successful adversarial attacks. As shown in Fig. 1(b, c), the targeted captions are successfully attacked, while the adversarial noises are invisible to human perception.

It should be emphasized that the value of this work is not just exploring the robustness the image captioning system, but also understanding its inside mechanism. The analyses about untargeted captions and the style of targeted captions could reveal the differences between automatic captioning and human captioning, as shown in Section 6. Moreover, the proposed formulation based on structured output learning is independent with any specific task. It provides a new perspective for exact attacks to deep neural networks with structured outputs, which has not been well studied.

The main contributions of this work are four-fold. (1) We are the first to study the adversarial attack of targeted partial captions to image captioning systems. (2) We formulate this attack problem as structured output learning with latent variables. (3) Extensive experiments show that state-of-the-art image captioning models can be easily attacked by the proposed methods. (4) We utilize the attack method to understand the inner mechanism of image captioning systems.

2. Related Work

Deep neural networks (DNNs) were firstly shown in [29] to be vulnerable to adversarial examples, and many seminal methods have been developed in this literature. According to the information about the attacked model accessible to

the attacker, existing works can be generally partitioned into three categories, including white-box, gray-box, and black-box attacks. We refer the readers to the survey of adversarial examples in [1] for more details. In this section, we categorize existing works according to the outputs of the attacked model, including independent and structured outputs.

Adversarial attacks to DNNs with independent outputs.

Since DNNs (especially CNNs) show very encouraging results on many visual tasks (*e.g.*, image classification [15], object detection [9], and semantic segmentation [21]), many previous works have also studied the robustness of these DNN-based visual tasks. For example, image classification is a typical successful visual application of CNNs, and it is also widely studied to verify the newly developed adversarial attack methods, such as box-constrained L-BFGS [29], fast-gradient-sign method (FGSM) [10], iterative FGSM [16], momentum iterative FGSM [7], Carlini and Wagner attack [5], DeepFool [23], *etc.* These works demonstrate that image classification based on popular CNN models (*e.g.*, ResNet [11] or Inception-v3 [28]) is very vulnerable to adversarial examples. The robustness of other typical visual tasks, *e.g.*, object detection and semantic segmentation, is also studied in [8, 22, 34] and [34, 8, 2], respectively. A common trait of above works is that they focus on CNNs and their loss functions are factorized to independent outputs. Consequently, the gradients of the loss function with respect to the input image can be easily computed to generate adversarial noises.

Adversarial attacks to DNNs with structured outputs.

However, the outputs of some deep models are structured. One typical model is the CNN+RNN architecture, of which the output is a temporally dependent sequence. It has been the main-stream model in some visual tasks, such as image captioning [30] and visual question answering [27]. Due to the dependencies among words in the sequence, it may be difficult to compute the gradient of the attack loss function with respect to the noise. An early attempt to attack CNN+RNN based tasks was proposed in [36]. However, it can only implement attacking of targeted complete sentences, and treat structured outputs as single outputs. A recent attack to the CNN+RNN based image captioning system is called *Show-and-Fool* [6]. It presents two types of attacks, including targeted captions and targeted keywords. Its attack of targeted captions is a special case of our studied attack of targeted partial captions. Its attack of targeted keywords encourages the predicted sentence to include the targeted keywords, but their locations cannot be specified. In contrast, our attack of targeted partial captions could enforce the targeted keywords to occur at specific locations, which is more restricted. Moreover, the formulations and optimization methods of Show-and-Fool are totally different with ours. Its formulations of targeted captions and keywords are different, while the proposed structured output learning with latent variables provides a systematic formulation for both

attacks of targeted partial and complete captions.

3. Structured Outputs of CNN+RNN based Image Captioning Systems

Given a trained CNN+RNN based captioning model with parameters θ , and an perturbed image $\mathbf{I} = \mathbf{I}_0 + \epsilon \in [0, 1]$, the posterior probability of a caption \mathbf{S} is formulated as

$$P(\mathbf{S}|\mathbf{I}_0, \epsilon; \theta) = \prod_{t=1}^N P(\mathbf{S}_t|\mathbf{S}_{<t}, \mathbf{I}_0, \epsilon; \theta), \quad (1)$$

where \mathbf{I}_0 represents the benign image, and ϵ denotes the adversarial noise. $\mathbf{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_t, \dots, \mathbf{S}_N\}$ indicates a sequence of N variables. \mathbf{S}_t indicates the output variable of t -step, and its state could be one from the candidate set $V = \{1, 2, \dots, |\mathcal{V}|\}$, corresponding to the set of candidate words, *i.e.*, \mathcal{V} . $\mathbf{S}_{<t} = \{\mathbf{S}_1, \dots, \mathbf{S}_{t-1}\}$; when $t = 1$, we define $\mathbf{S}_{<t} = \emptyset$. Note that we do not specify the formulation of $P(\mathbf{S}_t|\mathbf{S}_{<t}, \mathbf{I}_0, \epsilon; \theta)$, and it can be specified as any CNN+RNN model (*e.g.*, Show-and-Tell [30]). For clarity, we ignore the notations \mathbf{I}_0 and θ hereafter.

Besides, a partial caption is denoted as $\mathbf{S}_{\mathcal{O}}$, which means that the variables at the specific places \mathcal{O} are observed, while other variables are unobserved, *i.e.*, latent. Specifically, we define $\mathcal{O} \subset \{1, 2, \dots, N\}$ and $\mathbf{S}_{\mathcal{O}} = \{\mathbf{S}_t | t \in \mathcal{O}\}$, where $\mathbf{S}_t = s_t$ with $s_t \in V$ being the observed state. All observed states are summarized as an ordered set $S_{\mathcal{O}} = \{s_t | t \in \mathcal{O}\}$. The latent variables are defined as $\mathbf{S}_{\mathcal{H}} = \{\mathbf{S}_t | t \in \mathcal{H} \equiv \{1, 2, \dots, N\} \setminus \mathcal{O}\} = \mathbf{S} \setminus \mathbf{S}_{\mathcal{O}}$. Then, the posterior probability of the partial caption $\mathbf{S}_{\mathcal{O}}$ is formulated as:

$$P(\mathbf{S}_{\mathcal{O}}|\epsilon) = \sum_{\mathbf{S}_{\mathcal{H}}} P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}}|\epsilon), \quad (2)$$

where $\sum_{\mathbf{S}_{\mathcal{H}}}$ indicates the summation over all possible configurations of latent variables $\mathbf{S}_{\mathcal{H}}$.

4. Adversarial Attack of Targeted Partial Captions to Image Captioning

Learning ϵ . The goal of the targeted partial caption attack is to enforce the predicted caption \mathbf{S} to be compatible with $\mathbf{S}_{\mathcal{O}}$, meaning that the predicted words at \mathcal{O} are exactly $S_{\mathcal{O}}$. To this end, while minimizing the norm of adversarial noises, either of the following two criterion can be adopted. (1) The log marginal likelihood $\ln P(\mathbf{S}_{\mathcal{O}} = S_{\mathcal{O}}|\epsilon)$ is maximized (see Section 4.1). (2) The margin of the log marginal likelihood between the targeted caption (*i.e.*, $\ln P(\mathbf{S}_{\mathcal{O}} = S_{\mathcal{O}}|\epsilon)$) and all other possible partial captions (*i.e.*, $\ln P(\hat{\mathbf{S}}_{\mathcal{O}} \neq S_{\mathcal{O}}|\epsilon)$) is maximized. It is formulated as structural SVMs with latent variables (see Section 4.2).

Inference. Given the optimized ϵ , the caption of the image perturbed by ϵ is inferred as follows:

$$\mathbf{S}_{\epsilon}^* = \arg \max_{\mathbf{S}} P(\mathbf{S}|\mathbf{I}_0 + \epsilon). \quad (3)$$

4.1. Maximizing Log Marginal Likelihood via Generalized EM Algorithm

According to the first criterion, the adversarial noise ϵ for the targeted partial caption is derived by the maximization of log marginal likelihood, while minimizing $\|\epsilon\|_2^2$, as follows:

$$\begin{aligned} & \arg \max_{\epsilon} \ln P(\mathbf{S}_{\mathcal{O}}|\epsilon) - \lambda \|\epsilon\|_2^2 \quad (4) \\ & \equiv \arg \max_{\epsilon} \ln \sum_{\mathbf{S}_{\mathcal{H}}} P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}}|\epsilon) - \lambda \|\epsilon\|_2^2, \end{aligned}$$

subject to the constraint $\mathbf{I}_0 + \epsilon \in [0, 1]$. This constraint can be easily satisfied by clipping. For clarity, we ignore it hereafter. λ denotes the trade-off parameter. Due to the summation over all possible configurations of $\mathbf{S}_{\mathcal{H}}$, the above problem is difficult. To tackle it, the generalized expectation maximization (GEM) algorithm [4] is adopted. The core idea of GEM is introducing the factorized posterior $\mathbf{q}(\mathbf{S}_{\mathcal{H}}) = \prod_{t \in \mathcal{H}} \mathbf{q}(\mathbf{S}_t)$ to approximate the posterior probability $P(\mathbf{S}_{\mathcal{H}}|\mathbf{S}_{\mathcal{O}}, \epsilon)$. Then, we have the following equation,

$$\ln P(\mathbf{S}_{\mathcal{O}}|\epsilon) = \mathcal{L}(\mathbf{q}, \epsilon) + KL(\mathbf{q} \| P(\mathbf{S}_{\mathcal{H}}|\mathbf{S}_{\mathcal{O}}, \epsilon)), \quad (5)$$

$$\mathcal{L}(\mathbf{q}, \epsilon) = \sum_{\mathbf{S}_{\mathcal{H}}} \mathbf{q}(\mathbf{S}_{\mathcal{H}}) \ln \frac{P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}}|\epsilon)}{\mathbf{q}(\mathbf{S}_{\mathcal{H}})}, \quad (6)$$

$$KL(\mathbf{q}(\mathbf{S}_{\mathcal{H}}) \| P(\mathbf{S}_{\mathcal{H}}|\mathbf{S}_{\mathcal{O}}, \epsilon)) = \sum_{\mathbf{S}_{\mathcal{H}}} \mathbf{q}(\mathbf{S}_{\mathcal{H}}) \ln \frac{\mathbf{q}(\mathbf{S}_{\mathcal{H}})}{P(\mathbf{S}_{\mathcal{H}}|\mathbf{S}_{\mathcal{O}}, \epsilon)}. \quad (7)$$

According to the property of the KL divergence that $KL(\mathbf{q}(\mathbf{S}_{\mathcal{H}}) \| P(\mathbf{S}_{\mathcal{H}}|\mathbf{S}_{\mathcal{O}}, \epsilon)) \geq 0$, we obtain that $\mathcal{L}(\mathbf{q}, \epsilon) \leq \ln P(\mathbf{S}_{\mathcal{O}}|\epsilon)$. Consequently, the maximization problem (4) can be optimized through the following two alternative sub-problems, until convergence.

E step: Given ϵ , $\mathbf{q}(\mathbf{S}_{\mathcal{H}})$ is updated by minimizing the following equation

$$\begin{aligned} & KL(\mathbf{q}(\mathbf{S}_{\mathcal{H}}) \| P(\mathbf{S}_{\mathcal{H}}|\mathbf{S}_{\mathcal{O}}, \epsilon)) = \sum_{\mathbf{S}_{\mathcal{H}}} \mathbf{q}(\mathbf{S}_{\mathcal{H}}) \ln \mathbf{q}(\mathbf{S}_{\mathcal{H}}) \quad (8) \\ & - \sum_{\mathbf{S}_{\mathcal{H}}} \mathbf{q}(\mathbf{S}_{\mathcal{H}}) \left[\ln P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}}|\epsilon) - \ln \sum_{\mathbf{S}_{\mathcal{H}}} P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}}|\epsilon) \right] = \\ & \sum_{t=1}^N \sum_{k=1}^{|\mathcal{V}|} \mathbf{q}(\mathbf{S}_t^k) \left[\ln \mathbf{q}(\mathbf{S}_t^k) - \sum_{\mathbf{S}_{<t, \mathcal{H}}} \mathbf{q}(\mathbf{S}_{<t, \mathcal{H}}) \ln P(\mathbf{S}_t^k|\mathbf{S}_{<t, \mathcal{H}}|\epsilon) \right], \end{aligned}$$

where the constant $\sum_{\mathbf{S}_{\mathcal{H}}} \mathbf{q}(\mathbf{S}_{\mathcal{H}}) \left[\ln \sum_{\mathbf{S}_{\mathcal{H}}} P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}}|\epsilon) \right]$ in the last formula is ignored. $\mathbf{q}(\mathbf{S}_t^k) = \mathbf{q}(\mathbf{S}_t = k)$ indicates the probability of the variable \mathbf{S}_t with the state k , and $\sum_{k \in V} \mathbf{q}(\mathbf{S}_t^k) = 1$. $\mathbf{S}_{<t} = \{\mathbf{S}_1, \dots, \mathbf{S}_{t-1}\}$ and $\mathbf{S}_{<t, \mathcal{H}} = \mathbf{S}_{<t} \cap \mathbf{S}_{\mathcal{H}}$. When $t = 1$ and $t \in \mathcal{H}$, we define $\mathbf{S}_{<t, \mathcal{H}} = \emptyset$. Due to the sequential dependency among \mathbf{S} , the probability $\mathbf{q}(\mathbf{S}_t)$ can be updated in an ascending order (*i.e.*, from 1 to N). Specifically, with fixed $\mathbf{q}(\mathbf{S}_{<t, \mathcal{H}})$, the update of $\mathbf{q}(\mathbf{S}_t^k)$ is derived by setting its gradient to 0, as follows:

$$1 + \ln \mathbf{q}(\mathbf{S}_t^k) - \sum_{\mathbf{S}_{<t, \mathcal{H}}} \mathbf{q}(\mathbf{S}_{<t, \mathcal{H}}) \ln P(\mathbf{S}_t^k|\mathbf{S}_{<t, \mathcal{H}}|\epsilon) = 0$$

$$\begin{aligned} \Rightarrow \mathbf{q}(\mathbf{S}_t^k) &= \exp \left(\sum_{\mathbf{S}_{<t, \mathcal{H}}} \mathbf{q}(\mathbf{S}_{<t, \mathcal{H}}) \ln P(\mathbf{S}_t^k | \mathbf{S}_{<t, \mathcal{H}}, \epsilon) - 1 \right) \\ \Rightarrow \mathbf{q}(\mathbf{S}_t^k) &\leftarrow \mathbf{q}(\mathbf{S}_t^k) / (\sum_k^{|\mathcal{V}|} \mathbf{q}(\mathbf{S}_t^k)) \end{aligned} \quad (9)$$

M step: Given $\mathbf{q}(\mathbf{S}_{\mathcal{H}})$, ϵ is updated as follows:

$$\begin{aligned} &\arg \max_{\epsilon} \mathcal{L}(\mathbf{q}, \epsilon) - \lambda \|\epsilon\|_2^2 \quad (10) \\ &= \text{const} - \lambda \|\epsilon\|_2^2 + \sum_{\mathbf{S}_{\mathcal{H}}} \mathbf{q}(\mathbf{S}_{\mathcal{H}}) \ln P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}} | \epsilon) = \text{const} \\ &+ \sum_{t=1}^N \left[\sum_{\mathbf{S}_{1 \sim t, \mathcal{H}}} \mathbf{q}(\mathbf{S}_{1 \sim t, \mathcal{H}}) \ln P(\mathbf{S}_t | \mathbf{S}_{<t, \mathcal{H}}, \epsilon) \right] - \lambda \|\epsilon\|_2^2, \end{aligned}$$

where $\mathbf{S}_{1 \sim t, \mathcal{H}} = \{\mathbf{S}_1, \dots, \mathbf{S}_t\} \cap \mathbf{S}_{\mathcal{H}}$, and $\text{const} = -\sum_{\mathbf{S}_{\mathcal{H}}} \mathbf{q}(\mathbf{S}_{\mathcal{H}}) \ln \mathbf{q}(\mathbf{S}_{\mathcal{H}})$. It can be easily optimized by any gradient based method for training deep neural networks, such as stochastic gradient descent (SGD) [25] or adaptive moment estimation (ADAM) [13]. It will be specified in our experiments. However, the number of all possible configurations of $\mathbf{S}_{1 \sim t, \mathcal{H}}$ is $|\mathcal{V}|^{|\mathbf{S}_{1 \sim t, \mathcal{H}}|}$. It could be very large even for moderate $|\mathbf{S}_{1 \sim t, \mathcal{H}}|$. Fortunately, since $\mathbf{q}(\mathbf{S}_t^k) \in [0, 1]$ and $\sum_k^{|\mathcal{V}|} \mathbf{q}(\mathbf{S}_t^k) = 1$ for any $t \in \{1, \dots, N\}$, the values of $\mathbf{q}(\mathbf{S}_{1 \sim t, \mathcal{H}})$ for most configurations of $\mathbf{S}_{1 \sim t, \mathcal{H}}$ are so small that they can be numerically ignored. Thus, we only consider the configurations of top-3 probabilities of $\mathbf{q}(\mathbf{S}_t)$ for each latent variable \mathbf{S}_t . Consequently, the number of all configurations is reduced to $3^{|\mathbf{S}_{1 \sim t, \mathcal{H}}|}$, over which the summation becomes tractable.

4.2. Structural SVMs with Latent Variables

According to the second criteria, the adversarial noise ϵ is generated by structural SVMs with latent variables [37],

$$\begin{aligned} &\arg \min_{\epsilon} \lambda \|\epsilon\|_2^2 - \max_{\mathbf{S}_{\mathcal{H}}} \ln P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}} | \epsilon) \quad (11) \\ &+ \max_{\hat{\mathbf{S}}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{H}}} [\ln P(\hat{\mathbf{S}}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{H}} | \epsilon) + \Delta(\mathbf{S}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{O}})], \end{aligned}$$

$$\Delta(\mathbf{S}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{O}}) = \sum_{t \in \mathcal{O}} \Delta(\mathbf{S}_t, \hat{\mathbf{S}}_t), \quad \Delta(\mathbf{S}_t, \hat{\mathbf{S}}_t) = \begin{cases} \zeta, & \mathbf{S}_t \neq \hat{\mathbf{S}}_t \\ 0, & \mathbf{S}_t = \hat{\mathbf{S}}_t, \end{cases} \quad (12)$$

where the scalar $\zeta > 0$ will be specified in experiments. This problem can be optimized by the following two alternative sub-problems, until convergence.

(1) Latent variable completion with fixed ϵ :

$$\mathbf{S}_{\mathcal{H}}^* = \arg \max_{\mathbf{S}_{\mathcal{H}}} \ln P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}} | \epsilon). \quad (13)$$

It is solved by sequential inference in an ascending order.

(2) Optimizing ϵ via Structural SVMs with fixed $\mathbf{S}_{\mathcal{H}}^*$:

$$\begin{aligned} &\arg \min_{\epsilon} \lambda \|\epsilon\|_2^2 + \max_{\hat{\mathbf{S}}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{H}}} [\ln P(\hat{\mathbf{S}}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{H}} | \epsilon) + \Delta(\mathbf{S}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{O}})] \\ &- \ln P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}}^* | \epsilon). \end{aligned} \quad (14)$$

This problem is also optimized by two alternative steps.

(2.1) Loss augmented inference with fixed ϵ :

$$\hat{\mathbf{S}}_{\mathcal{O}}^*, \hat{\mathbf{S}}_{\mathcal{H}}^* = \arg \max_{\hat{\mathbf{S}}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{H}}} \ln P(\hat{\mathbf{S}}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{H}} | \epsilon) + \Delta(\mathbf{S}_{\mathcal{O}}, \hat{\mathbf{S}}_{\mathcal{O}}). \quad (15)$$

This inference problem is also sequentially solved in an ascending order. Specifically, given the inferred configurations $\hat{\mathbf{S}}_{<t}^*$, the inference over $\hat{\mathbf{S}}_t$ is solved as follows:

1. When $t \in \mathcal{O}$, $\hat{\mathbf{S}}_t^* = \arg \max_{\hat{\mathbf{S}}_t} [\ln P(\hat{\mathbf{S}}_t | \hat{\mathbf{S}}_{<t}^*, \epsilon) + \Delta(\mathbf{S}_t, \hat{\mathbf{S}}_t)]$.
2. When $t \in \mathcal{H}$, $\hat{\mathbf{S}}_t^* = \arg \max_{\hat{\mathbf{S}}_t} \ln P(\hat{\mathbf{S}}_t | \hat{\mathbf{S}}_{<t}^*, \epsilon)$.

(2.2) Update ϵ with fixed $\hat{\mathbf{S}}_{\mathcal{O}}^*, \hat{\mathbf{S}}_{\mathcal{H}}^*$:

$$\arg \min_{\epsilon} \lambda \|\epsilon\|_2^2 + \ln P(\hat{\mathbf{S}}_{\mathcal{O}}^*, \hat{\mathbf{S}}_{\mathcal{H}}^* | \epsilon) - \ln P(\mathbf{S}_{\mathcal{O}}, \mathbf{S}_{\mathcal{H}}^* | \epsilon). \quad (16)$$

Similar to M step (see Eq. (10)) in GEM, as the gradients of all three terms in the above objective function with respect to ϵ can be easily computed, any gradient based optimization method for training deep neural networks can be used. It will be specified in our experiments.

Remarks on both Sections 4.1 and 4.2. Unlike the general structured output learning with a repeated inference process (e.g., MRFs [14]), the proposed GEM and latent SSVMs are based on CNN+RNN architecture (i.e., $P(\mathbf{S} | \mathbf{I}_0, \epsilon; \theta)$ in Eq. (1)), which requires only one pass along the prediction sequence of RNNs. Excluding the forward and backward through CNNs, the complexities of GEM and latent SSVMs are $O(T(|\mathcal{V}|N^2 + 3^N d))$ and $O(T_{outer}(|\mathcal{V}|N_{\mathcal{H}} + T_{inner}(|\mathcal{V}|N + 2Nd)))$, respectively, with T being the iteration number, and d being the output dimension of RNNs.

5. Experiments

5.1. Experimental Setup

In this section, we evaluate the attack performance of the proposed two methods on three CNN+RNN based image captioning models, including Show-Attend-and-Tell (SAT) [35], self-critical sequence training (SCST) [24], and Show-and-Tell (ST) [30]. We also compare with the only related method (to the best of our knowledge), called Show-and-Fool [6], that also attacks the Show-and-Tell model.

Database and targeted captions. Our experiments are conducted on the benchmark database for image captioning, i.e., Microsoft COCO 2014 (MSCOCO) [20]. We adopt the split of MSCOCO in [12], including 113, 287 training, 5, 000 validation and 5, 000 test images. Following the setting of [6], we randomly select 1, 000 from 5, 000 validation images as the attacked images. Using each attacked model (i.e., SAT, SCST, or ST), we predict the captions of the remaining

4,000 benign validation images. We randomly choose 5 different targeted complete captions from these 4,000 captions for each attacked image. Based on each targeted complete caption, we also generate 6 targeted partial captions, including the partial captions with 1 to 3 latent words (all other words are observed), and those with 1 to 3 observed words (all other words are latent), respectively. Latent or observed words are randomly chosen from each targeted caption. As the first word in most targeted captions is ‘a’, we keep it as observed, and skip it when choosing latent or observed words. Due to the memory limit of GPUs, observed words are randomly chosen from the second to the 7th location in each targeted caption. The selected 1,000 images and corresponding 5,000 targeted complete captions of each attacked model will be released along with our codes in early future.

Evaluation metrics. Given one targeted caption \mathbf{S}_O for the benign image \mathbf{I}_0 , the adversarial noise ϵ is measured by its ℓ_2 norm, *i.e.*, $\|\epsilon\|_2$; the predicted caption \mathbf{S}_ϵ^* (see Eq. (3)) for $\mathbf{I}_0 + \epsilon$ is evaluated by the following three metrics. First, the success sign is defined as follows:

$$\text{succ-sign} = \begin{cases} 1, & \text{if } \mathbf{S}_{\epsilon, \mathcal{O}}^* \equiv \mathbf{S}_O \\ 0, & \text{if } \mathbf{S}_{\epsilon, \mathcal{O}}^* \neq \mathbf{S}_O \end{cases}, \quad (17)$$

where \equiv exactly compares two sequences, and $\mathbf{S}_{\epsilon, \mathcal{O}}^* \subset \mathbf{S}_\epsilon^*$ denotes the sub-sequence of \mathbf{S}_ϵ^* at observed locations \mathcal{O} . As \mathbf{S}_ϵ^* may be too short to include all observed locations, we know that $|\mathbf{S}_{\epsilon, \mathcal{O}}^*| \leq |\mathbf{S}_O|$, with $|\cdot|$ calculating the length of sequence. However, succ-sign cannot measure how many inconsistent words in \mathbf{S}_ϵ^* with \mathbf{S}_O . Thus, we also define the following two metrics:

$$\text{Precision} = \frac{|\mathbf{S}_{\epsilon, \mathcal{O}}^* \cap \mathbf{S}_O|}{|\mathbf{S}_{\epsilon, \mathcal{O}}^*|}, \quad \text{Recall} = \frac{|\mathbf{S}_{\epsilon, \mathcal{O}}^* \cap \mathbf{S}_O|}{|\mathbf{S}_O|}, \quad (18)$$

where the operator \cap between two sequences returns a sub-sequence including the same words at the same locations. If succ-sign is 1, then both Precision and Recall are 1; if succ-sign is 0, then Precision and Recall may be larger than 0. Besides, considering that $|\mathbf{S}_{\epsilon, \mathcal{O}}^*| \leq |\mathbf{S}_O|$, we obtain that $\text{Precision} \geq \text{Recall} \geq \text{succ-sign}$. We report the average values of above four metrics over all targeted (partial) captions of all images, *i.e.*, 5000 captions. The average value of succ-sign is called as success rate (SR). The lower average norm $\|\epsilon\|_2$, while the higher average values of other three metrics, indicate the better attack performance.

Implementation details. The PyTorch implementations of three target models are downloaded from an open-source GitHub project¹. We train these models based on the training set of MSCOCO. We adopt the ResNet-101 architecture [11] as the CNN part in SAT and SCST. Besides, to fairly compare with the Show-and-Fool algorithm [6], we adopt the Inception-v3 [28] architecture as the CNN part in the ST

method	metric	0 latent	1 latent	2 latent	3 latent	1 obser	2 obser	3 obser
GEM	$\ \epsilon\ _2 \downarrow$	4.2767	4.4976	4.6942	4.858	3.0304	3.5611	3.6583
	SR \uparrow	0.9926	0.9154	0.759	0.5604	0.8908	0.862	0.892
	Prec \uparrow	0.9953	0.9575	0.9092	0.856	0.8908	0.8897	0.9236
	Rec \uparrow	0.9953	0.9528	0.8855	0.8	0.8908	0.8876	0.9234
Latent SSVMs	$\ \epsilon\ _2 \downarrow$	5.1678	5.4558	5.7074	5.8706	5.2509	5.6838	5.8681
	SR \uparrow	0.9806	0.9126	0.8466	0.7526	0.85	0.731	0.708
	Prec \uparrow	0.9892	0.955	0.9197	0.8868	0.85	0.8092	0.8096
	Rec \uparrow	0.9889	0.9524	0.9151	0.8792	0.85	0.7896	0.7917

Table 1. Results of adversarial attack to the Show-Attend-and-Tell model. ‘1 obser’ indicates the targeted partial caption of one observed word. ‘Prec’ indicates Precision, while ‘Rec’ means Recall. \downarrow means that the lower value of that metric is the better attack performance, while \uparrow means that the higher value of that metric is the better attack performance.

model. For the GEM based attack method, the maximum number of iterations between E and M step is set to 50; for the latent SSVM based attack method, the maximum numbers of both outer and inner iterations are set to 10. In the M step (see Eq. (10)) of GEM, and the (2.2) step (see Eq. (16)) of latent SSVMs, we adopt the ADAM optimization algorithm [13] to update the noise ϵ , with the learning rate 0.001, while all other hyper-parameters are set to the default values in the master branch of PyTorch². If without specific illustrations, the trade-off parameters λ in both Eq. (4) and (11) are set to 0.1 in experiments. The scalar ζ of the structured loss (see Eq. (12)) in latent SSVMs is set to 1.

5.2. Attack Results of Three State-of-the-Art Image Captioning Models

Attack results of the Show-Attend-and-Tell model [35] are presented in Table 1. (1) In terms of the attacks of targeted complete captions (*i.e.*, ‘0 latent’ in the third column of Table 1), the SR of GEM is up to 0.9926, while means that only 37 targeted captions out of 5,000 targeted captions are not successfully predicted after generating adversarial noises. And, the corresponding Precision and Recall of GEM are up to 0.9953. It means that even in failed attacks, many words are also successfully predicted. The average noise norm $\|\epsilon\|_2$ of GEM is 4.2767. As shown in Fig. 2, such small noises are invisible to human perception. In contrast, the results of latent SSVMs are slightly worse than those of GEM. (2) In terms of the attacks of targeted partial captions with 1 to 3 latent words, along the increase of the number of latent words, the results of both GEM and latent SSVMs get worse, with decreasing (SR, Precision, Recall) and increasing $\|\epsilon\|_2$. The reason is that more latent words bring in more uncertainties on predictions of these latent locations. Then, the observed words after latent locations will be influenced by these uncertainties. (3) In terms of the attacks of targeted partial captions with 1 to 3 observed words, there is not a clear relationship between the attack performance and the number of observed words. The reason is that there is a trade-off between satisfying observed words and the uncer-

¹<https://github.com/ruotianluo/self-critical.PyTorch>

²<https://github.com/PyTorch/PyTorch/blob/master/torch/optim/adam.py>

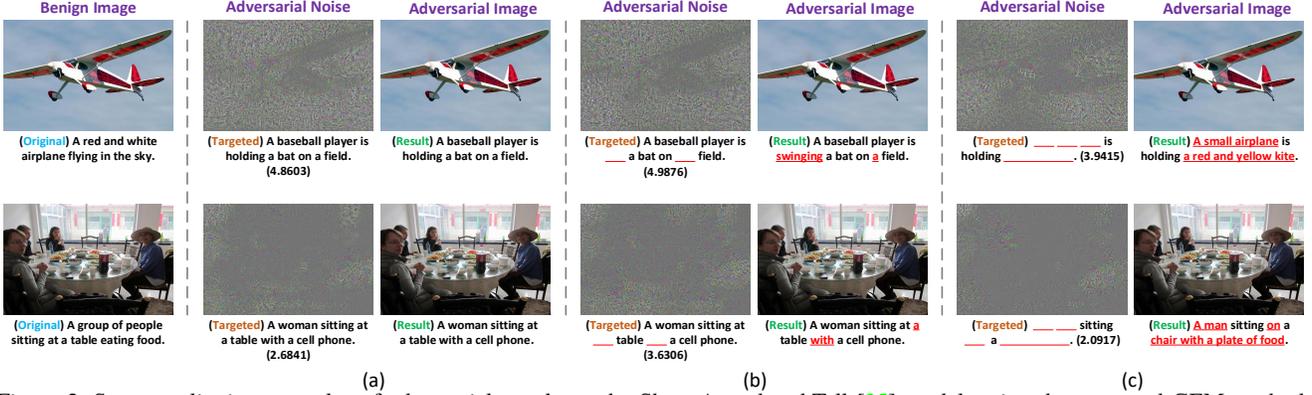


Figure 2. Some qualitative examples of adversarial attacks to the Show-Attend-and-Tell [35] model, using the proposed GEM method. Attacks of (a) targeted complete captions; (a) targeted partial captions with two latent words; (a) targeted partial captions with two observed words. All targeted partial/complete captions are successfully attacked, while the adversarial noises are invisible to human perception.

method	metric	λ					
		0.001	0.01	0.1	1	10	100
GEM	$\ \epsilon\ _2 \downarrow$	8.6353	7.67	4.2767	1.6862	0.7513	0.2701
	SR \uparrow	0.9956	0.9952	0.9926	0.9402	0.4126	0.0118
	Prec \uparrow	0.9973	0.9969	0.9953	0.9595	0.5832	0.2128
	Rec \uparrow	0.9972	0.9969	0.9953	0.9589	0.5754	0.2011
Latent SSVMs	$\ \epsilon\ _2 \downarrow$	9.2682	8.2134	5.1678	2.5074	1.023	0.2939
	SR \uparrow	0.985	0.9818	0.9806	0.9252	0.4144	0.012
SSVMs	Prec \uparrow	0.9919	0.99	0.9892	0.9588	0.6172	0.227
	Rec \uparrow	0.9917	0.9897	0.9889	0.9574	0.6092	0.2161

Table 2. Attack results of targeted complete captions to the Show-Attend-and-Tell model, with different trade-off parameters λ (see Eqs. (4) and (11)).

tainty from the latent words. (4) In comparison of GEM and latent SSVMs, the average norm $\|\epsilon\|_2$ of adversarial noises produced by GEM is always lower than that produced by latent SSVMs at all cases. The attack performance (evaluated by SR, Precision and Recall) of GEM is also better than that of latent SSVMs at most cases, excluding two cases of 2 and 3 latent words. However, based on these results, we cannot simply conclude which method is better for adversarial attacks to image captioning. Because these two methods are influenced by the trade-off parameter λ , and latent SSVMs is also affected by the parameter ζ defined in Eq. (12).

In the above analysis, the trade-off parameters λ in both Eqs. (4) and (11) are fixed at 0.1. In the following, we explore the influence of λ to the attack performance. When λ becomes larger, the norm of adversarial noises is expected to be smaller, while the loss gets larger, leading to weaker attack performance. This point is fully verified by the results in Table 2. When $\lambda = 0.001$, the SR value of GEM is up to 0.9956, and $\|\epsilon\|_2$ is 8.6353; when $\lambda = 100$, the SR value of GEM is up to 0.0118, and $\|\epsilon\|_2$ is 0.2701. With the same λ , GEM performs slightly better than latent SSVMs in most cases, with lower $\|\epsilon\|_2$ and higher SR, Precision, and Recall. However, the performance of latent SSVMs may be also influenced by ζ (see Eq. (12)). Due to the space limit, it will be studied in the **supplementary material**.

Attack results of the SCST model [24] are shown in Table

method	metric	0 latent	1 latent	2 latent	3 latent	1 obser	2 obser	3 obser
		GEM	$\ \epsilon\ _2 \downarrow$	5.1978	5.5643	5.8561	6.1171	4.3749
	SR \uparrow	0.992	0.9168	0.7438	0.5178	0.6344	0.6372	0.7838
	Prec \uparrow	0.9956	0.9549	0.8847	0.7788	0.6344	0.7328	0.8543
	Rec \uparrow	0.9956	0.9528	0.872	0.7503	0.6344	0.7319	0.8543
Latent SSVMs	$\ \epsilon\ _2 \downarrow$	4.7005	5.0926	5.5109	5.8674	5.989	5.7939	5.4646
	SR \uparrow	0.9804	0.916	0.8576	0.7598	0.569	0.7066	0.8294
	Prec \uparrow	0.9926	0.9684	0.934	0.8835	0.6538	0.7835	0.8815
	Rec \uparrow	0.9924	0.967	0.9306	0.8784	0.6502	0.7809	0.8801

Table 3. Results of adversarial attacks to the SCST model.

3. The phenomenon behind these results is similar with that behind the results of the Show-Attend-and-Tell model. The reason is that the model structures of Show-Attend-and-Tell and SCST are similar that the visual features extracted by CNNs are fed into RNNs at each step.

Attack results of the Show-and-Tell model [30] are reported in Table 4. It is found that the attack performance of Show-and-Tell is much worse than that of Show-Attend-and-Tell (see Table 1) and SCST (see Table 3). The main reason is that the model structure of Show-and-Tell is significantly different with the structures of the other two models. Specifically, the visual features extracted by CNN is only fed into the starting step of RNNs, while they are fed into RNNs at every step in Show-Attend-and-Tell and SCST. Consequently, the gradients of observed words in targeted partial captions can be directly back-propagated to the input image in Show-Attend-and-Tell and SCST. In contrast, the gradients of both observed words and latent words are firstly multiplied, then are back-propagated to the input image. Obviously, the influence of observed words becomes much weaker. Thus, it is expected that the observed word closer to the end of one caption is more difficult to be successfully attacked. To verify this point, we summarize the success rate of observed words at each location. As the lengths of targeted captions vary significantly, we only summarize the words at the first 7 locations. As shown in Fig. 3, in both targeted partial captions with one observed word and targeted complete captions, as well as using both GEM and latent SSVMs, the SR value decreases along the increasing of locations.

Due to the space limit, we will present: (1) attack results

method	metric	0 latent	1 latent	2 latent	3 latent	1 obser	2 obser	3 obser
GEM	$\ \epsilon\ _2 \downarrow$	4.5959	3.4488	3.3999	3.3783	2.2588	2.5779	2.7472
	SR \uparrow	0.4404	0.5034	0.4094	0.3408	0.4606	0.4248	0.4962
	Prec \uparrow	0.6758	0.7475	0.691	0.6455	0.4606	0.5468	0.6403
	Rec \uparrow	0.6635	0.7344	0.6763	0.626	0.4606	0.5468	0.6403
Latent SSVMs	$\ \epsilon\ _2 \downarrow$	1.7635	4.5913	4.6584	4.7369	4.5513	4.8617	4.933
	SR \uparrow	0.4924	0.5808	0.4634	0.3978	0.287	0.2118	0.227
	Prec \uparrow	0.7438	0.7982	0.7257	0.6697	0.287	0.3609	0.4065
	Rec \uparrow	0.7318	0.7862	0.7122	0.6545	0.287	0.3459	0.3898

Table 4. Results of adversarial attacks to the Show-and-Tell model.

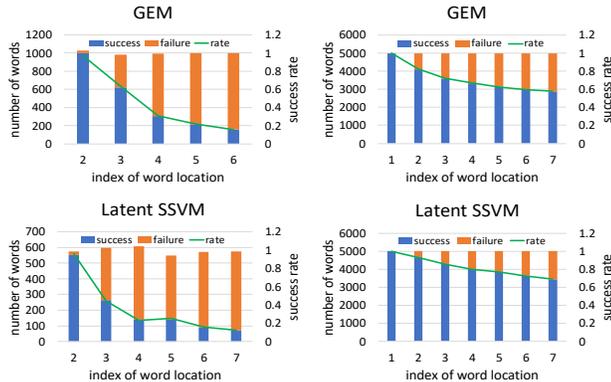


Figure 3. Statistics of success rates of observed words at different locations, on attacking the Show-and-Tell model. (Left): the attack of targeted partial captions with one observed word; (Right): the attack of targeted complete captions.

of the Show-and-Tell model with the ResNet-101 architecture as the CNN part; (2) results of transfer attacks among three captioning models; (3) more qualitative results (like Fig. 2) of attacks of targeted partial captions on above three image captioning models, in **supplementary materials**. We also report the average runtime of attacking one image in the case of targeted complete captions, using the proposed two methods. On the four attacked models, including Show-Attend-and-Tell, SCST, Show-and-Tell with Inception-v3 and Show-and-Tell with ResNet-101, the respective average runtime (seconds) of the GEM method is 95, 81, 36 and 61, and of latent SSVMs is 28, 25, 15 and 24. As GEM requires more back-propagation (see the summation term $\sum_{t=1}^N \sum_{s_1 \sim t, \tau_t}$ in Eq. (10)) than latent SSVMs, its runtime is larger.

5.3. Comparison with Show-and-Fool [6]

In this section, we compare with the only related work called Show-and-Fool [6]. Its attack of targeted captions is a special case of our studied attack of targeted partial captions, *i.e.*, targeted complete captions. The derivations of two methods proposed in Show-and-Fool also start from the joint probability of a caption given a pre-trained CNN+RNN model, which is same with our derivations. However, the derived objective functions of Show-and-Fool are totally different with our objective functions. Specifically,

- *Maximizing logits in Show-and-Fool* (see Eq. (6) in [6]) vs. *our maximizing log likelihood* (see Eq. (4)). Show-and-Fool directly removes the normalization term of the Softmax function of RNNs, as they

thought this term is a constant with respect to the input adversarial noises. Actually, this normalization term depends on adversarial noises. Thus, maximizing logits and maximizing log likelihood are different.

- *Max margin of logits in Show-and-Fool* (see Eq. (7) in [6]) vs. *our max margin of log likelihood* (see Eq. (11)). Show-and-Fool maximizes the logit margin between each observed word and all other possible words at the same location. When inferring the word from all other possible words at one location, the corresponding logit exploits the observed word at its previous location as the condition, rather than the inferred word of its previous location. This objective function is standard SVMs factorized at each location, while our objective function is structural SVMs of the whole caption.

In the following, we present some experimental comparisons between Show-and-Fool and our methods, on the attack of targeted complete captions. Show-and-Fool is implemented using Tensorflow³, and attacks the Show-and-Tell model [30], of which the CNN part is the Inception-v3 model [28]. To fairly compare with Show-and-Fool, we re-implement our methods using Tensorflow, based on the implementation codes of Show-and-Fool, and attack the same checkpoint of the Show-and-Tell model. Besides, Show-and-Fool adopts the arctanh function to transform \mathbf{I}_0 and $\mathbf{I}_0 + \epsilon$ to $y = \text{arctanh}(\mathbf{I}_0)$, $w = \text{arctanh}(\mathbf{I}_0 + \epsilon)$, to satisfy the requirement that $\mathbf{I}_0, \mathbf{I}_0 + \epsilon \in [-1, 1]$. In this section, our method also adopt this setting. However, when computing the norm $\|\epsilon\|_2$ for evaluation, we still transform \mathbf{I}_0 and ϵ into the range $\mathbf{I}_0, \mathbf{I}_0 + \epsilon \in [0, 1]$. The trade-off parameter λ is set to 1 for both Show-and-Fool and our methods. The slack constant ζ (see Eq. (12)) in max margin of Show-and-Fool is set to 10,000 (the default value in the provided code), while 1 in our SSVM method. The experiments are also conducted on the selected 1000 images and 5000 targeted captions (see Section 5.1). The results are shown in Table 5. Our GEM method shows the best attack performance. However, due to the differences on objective functions and optimization methods, these results with similar hyper-parameters may not give a clear conclusion that which method is better. But, we still obtain two observations: (1) Using logits or log probabilities in the loss term can affect the attack performance; (2) The comparison between Table 4 and Table 5 tells that different checkpoints of the same attacked model (*i.e.*, Show-and-tell) will influence the attack performance.

Show-and-Fool [6] also presented the attack of targeted keywords, requiring that the targeted keywords should occur in the predicted caption, but their locations cannot be determined. In contrast, our attack of targeted partial captions can enforce the targeted words to occur at specific locations. Besides, the formulations for attacks of targeted captions and

³ <https://github.com/huanzhang12/ImageCaptioningAttack>

metric	Show-and-Fool [6]		Our methods	
	max logits	max margin of logits	GEM	latent SSVMs
$\ \epsilon\ _2 \downarrow$	1.5202	1.7423	2.3494	4.7854
SR \uparrow	0.5226	0.6586	0.7134	0.4996
Prec \uparrow	0.7239	0.8009	0.8933	0.7335
Rec \uparrow	0.7135	0.7926	0.886	0.7215

Table 5. Comparisons between Show-and-Fool [6] and our methods.

	<p>Passive Sentence (Targeted) A frisbee is played by a young boy on the beach.</p> <p>(1) A man is playing frisbee with a dog (on the beach). (3.5022)</p> <p>(2) A frisbee is played by a young boy on the beach. (4.6760)</p> <p>(3) A frisbee is playing with a frisbee on the beach. (6.3204)</p>	<p>Attributive Clause (Targeted) A young boy is playing with a frisbee which is orange on the beach.</p> <p>(1) A young boy is playing with a frisbee (which is orange) on the beach. (3.2735)</p> <p>(2) A young boy is playing with a frisbee (which is orange) on the beach. (4.7610)</p> <p>(3) A young boy is playing with a frisbee (which is orange) on the beach. (5.3805)</p>
	<p>Passive Sentence (Targeted) A bike is ridden by a man with a dog.</p> <p>(1) A dog is riding a horse on the beach. (3.6312)</p> <p>(2) A bike is ridden by a man with a dog. (5.6850)</p> <p>(3) A bike is driving down a street with a dog. (6.8397)</p>	<p>Attributive Clause (Targeted) A toilet that is white in a bathroom with a table.</p> <p>(1) A small bird sitting on a rock in the middle of a field. (4.2003)</p> <p>(2) A toilet (that) is in the middle of a bathroom (with a table). (5.8707)</p> <p>(3) A toilet (that) is standing in a bathroom with a table. (7.0076)</p>

Figure 4. Two examples of attacks different styles of targeted captions using GEM. (1), (2) and (3) represent the results of Show-Attend-and-Tell, Show-Attend-and-Tell and SCST model, respectively.

Benign Image	Captions	Adversarial Image
	<p>(Original caption) A man is playing tennis on a tennis court.</p> <p>(Untargeted caption) A girl green green green green green green green green green green. (5.8350)</p>	
	<p>(Original caption) A man riding a wave on top of a surfboard.</p> <p>(Untargeted caption) A woman is holding a dog is playing a blue bird in the water. (7.1830)</p>	

Figure 5. Examples of untargeted attacks to Show-Attend-and-Tell.

targeted keywords are different in Show-and-Fool, while the proposed structured output learning with latent variables provides a systematic formulation of both targeted attacks of complete and partial captions.

6. Extended Discussions

What style of targeted captions can be successfully attacked? As demonstrated in Section 5.1, the targeted captions are selected from the captions of 4000 benign validation images. It is found that most of these captions are active sentences, due to that most training captions are active. Can the image captioning system produce other styles of captions through adversarial attacking? To answer it, we run a simple test of using passive sentences and attributive clauses as targeted captions. As shown in Fig. 4 (left), only the Show-Attend-and-Tell model can produce passive sentences, while other two models still keep active. Fig. 4 (right) shows that all three models fail to produce attributive clauses. It demonstrates that current image captioning systems are not flexible enough to produce different styles of captions like humans.

Untargeted caption attack. Until now, we have only pre-

sented targeted caption attacks. In the following, we present a brief analysis about the untargeted caption attack, which can be formulated as follows:

$$\arg \min_{\epsilon} \ln P(\mathbf{S}_0|\epsilon) + \lambda \|\epsilon\|_2^2, \text{ s.t. } \mathbf{I}_0 + \epsilon \in [0, 1], \quad (19)$$

where \mathbf{S}_0 denotes the predicted caption on the benign image \mathbf{I}_0 . This problem can be easily solved by the projected gradient descent algorithm. Two attack results to the Show-Attend-and-Tell model are shown in Fig. 5. The predicted captions after attacking are non-meaningful, *i.e.*, violating the grammar of natural language. It is not difficult to explain this observation. In image classification, the classification space is continuous and closed, and the prediction will jump from one to another label if the image is attacked. However, the distributions of meaningful captions are not continuous in image captioning. There are massive non-meaningful captions around every meaningful caption. Consequently, we think it makes no sense to calculate how many captions are fooled by untargeted attacks. However, this simple analysis reveals an important information that state-of-the-art DNN-based image captioning systems have not learned or understood the grammar of natural language very well.

A brief summary. The above two studies demonstrate that state-of-the-art CNN+RNN image captioning systems are still far from human captioning. The proposed methods can be used as a probe tool to check what grammars have been learned by the automatic image captioning system, thus to guide the improvement towards human captioning.

7. Conclusions

In this paper, we have fooled the CNN+RNN based image captioning system to produce targeted partial captions by generating adversarial noises added onto benign images. We formulate the attack of targeted partial captions as a structured output learning problem. We further present two structured methods, including the generalized expectation maximization and the structural SVMs with latent variables. Extensive experiments demonstrate that state-of-the-art image captioning models can be easily attacked by the proposed methods. Furthermore, the proposed methods have been used to explore the inner mechanism of image captioning systems, revealing that current automatic image captioning systems are far from human captioning. In our future work, we plan to use the proposed methods to guide the improvement of automatic image captioning systems towards human captioning, and enhance the robustness.

Acknowledgement. The involvements of Yan Xu, Fumin Shen and Heng Tao Shen in this work were supported in part by the National Natural Science Foundation of China under Project 61502081, Sichuan Science and Technology Program (No. 2019YFG0003, 2018GZDZX0032).

References

- [1] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018. 2
- [2] A. Arnab, O. Miksik, and P. H. S. Torr. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2
- [3] G. BakIr, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. Vishwanathan. *Predicting Structured Data*. MIT press, 2007. 2
- [4] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 2, 3
- [5] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 2
- [6] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. In *Proceedings of the Association for Computational Linguistics*, volume 1, pages 2587–2597, 2018. 2, 4, 5, 7, 8
- [7] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2
- [8] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramèr, A. Prakash, T. Kohno, and D. Song. Physical adversarial examples for object detectors. *arXiv preprint arXiv:1807.07769*, 2018. 2
- [9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 1, 2
- [10] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*, 2015. 2
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2, 5
- [12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 4
- [13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 5
- [14] D. Koller, N. Friedman, and F. Bach. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009. 4
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1, 2
- [16] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 2
- [17] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995, 1995. 1
- [18] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015. 1
- [19] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang. Target-aware deep tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1
- [20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 2, 4
- [21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2
- [22] J. Lu, H. Sibai, E. Fabry, and D. Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017. 2
- [23] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016. 2
- [24] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 4, 6
- [25] H. Robbins and S. Monro. A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer, 1985. 4
- [26] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. 1
- [27] A. Stanislaw, A. Aishwarya, L. Jiasen, M. Margaret, B. Dhruv, Z. C. Lawrence, and P. Devi. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2
- [28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 2, 5, 7
- [29] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2
- [30] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015. 1, 2, 3, 4, 6, 7
- [31] B. Wu, W. Chen, Y. Fan, Y. Zhang, J. Hou, J. Huang, W. Liu, and T. Zhang. Tencent ml-images: A large-scale multi-label image database for visual representation learning. *arXiv preprint arXiv:1901.01703*, 2019. 1
- [32] B. Wu, W. Chen, P. Sun, W. Liu, B. Ghanem, and S. Lyu. Tagging like humans: Diverse and distinct image annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7967–7975, 2018. 1

- [33] B. Wu, F. Jia, W. Liu, and B. Ghanem. Diverse image annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2567, 2017. [1](#)
- [34] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2017. [2](#)
- [35] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, pages 2048–2057, 2015. [1](#), [2](#), [4](#), [5](#), [6](#)
- [36] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darrell, and D. Song. Fooling vision and language models despite localization and attention mechanism. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. [2](#)
- [37] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *Proceedings of the International Conference on Machine Learning*, pages 1169–1176. ACM, 2009. [2](#), [4](#)
- [38] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2042–2049, 2012. [1](#)