In Proceedings of the Visually Grounded Interaction and Language Workshop at NeurIPS 2019, December 2019.

Hidden State Guidance: Improving Image Captioning Using an Image Conditioned Autoencoder

Jialin Wu and **Raymond J. Mooney**
Department of Computer Science
University of Texas at Austin
{*jialinwu,mooney*}@*cs.utexas.edu*

Abstract

Most RNN-based image captioning models receive supervision on the output words to mimic human captions. Therefore, the hidden states can only receive noisy gradient signals via layers of back-propagation through time, leading to less accurate generated captions. Consequently, we propose a novel framework, Hidden State Guidance (HSG), that matches the hidden states in the caption decoder to those in a teacher decoder trained on an easier task of autoencoding the captions conditioned on the image. During training with the REINFORCE algorithm, the conventional rewards are sentence-based evaluation metrics equally distributed to each generated word, no matter their relevance. HSG provides a word-level reward that helps the model learn better hidden representations. Experimental results demonstrate that HSG clearly outperforms various state-of-the-art caption decoders using either raw images, detected objects, or scene graph features as inputs.

1 Introduction

In recent years, image captioning has been widely studied in both the vision and NLP communities. Most recent research (Xu et al. 2015; Donahue et al. 2015; Karpathy and Fei-Fei 2015; Vinyals et al. 2015; Anderson et al. 2018; Yao et al. 2018; Yang et al. 2018) trains an RNN-based decoder to learn the word probabilities conditioned on the previous hidden state and various visual features. These methods improve results by incorporating richer visual inputs from object detection (Anderson et al. 2018) and relationship detection (Yao et al. 2018; Yang et al. 2018).

By contrast, we focus on improving the hidden state representation learned during training. Most current image captioners are trained using maximum log-likelihood or REINFORCE with CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015) or BLEU (Papineni et al. 2002) rewards, where only the final word probabilities receive supervision. Therefore, the hidden states can only access noisy training signals from layers of backpropagation through time. Especially when training using REINFORCE, rewards are delayed till the end and equally distributed to each word in the caption, regardless of whether or not the words are descriptive, making the training signals even noisier.

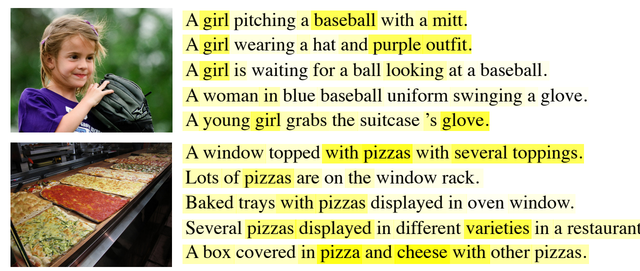

Figure 1: Sample word-level rewards for generated captions. A yellower background indicates the word receives more reward when using REINFORCE.

We present a new framework, called Hidden State Guidance (HSG), that treats the RNN caption decoder as a student network (Romero et al. 2014) and directly guides its hidden-state learning. However, this requires a teacher to provide hidden state supervision. We use a caption autoencoder as the teacher, giving it the same image as input. Its decoder has the same architecture as the caption decoder, allowing matching of the hidden states. Since the teacher has access to all of the human captions *and* visual inputs, its hidden states are expected to encode a richer representation that generates better captions, and therefore, provides useful hidden state supervision.

In order to align the initial states of the teacher and student,[1] we insert a small state transformation network that uses the visual features to estimate the initial state of the teacher decoder. During testing, run-time only slightly increases due to the light-weight state transformation network.

HSG plays a particularly helpful role when training using REINFORCE since it also provides a word-level intermediate reward that highlights the important words. For example, Figure 1 shows generated captions with the rewards indicated by the words' background intensity. These are generated using the model trained with maximum likelihood. HSG recognizes descriptive words like "purple outfit" and "glove," and rewards them more than digressive words like

[1]The initial state of the teacher decoder is the output of the encoder and that of the original student decoder is often initialized to zeros.

"waiting" and "swinging".

Our general framework can be used in almost any RNN-based image captioner. Experimental results show significant improvements over three recent caption decoders, FC (Rennie et al. 2017) using image features, Up-Down (Anderson et al. 2018) using object detection features, and SGAE (Yang et al. 2018) using scene graph features.

2 Related Work

2.1 Image Captioning

Most recent image captioning models use RNNs ($i.e.$ GRUs (Cho et al. 2014), LSTMs (Hochreiter and Schmidhuber 1997)) as caption decoders (Donahue et al. 2015; Karpathy and Fei-Fei 2015; Vinyals et al. 2015; Luo et al. 2018; Liu et al. 2018). The output words' probabilities at each step are trained to maximize the human captions' log-likelihood or some end evaluation metric ($e.g.$ CIDEr) directly using REINFORCE.

However, less research provides hidden-state supervision. *Professor forcing* (Lamb et al. 2016) trains the hidden states of an RNN to be indistinguishable whether the network is trained with its inputs clamped to a training sequence or whether its inputs are self-generated. However, the professor hidden states do not necessarily carry richer information to guide the student hidden states. Our image caption framework uses richer hidden states from an easier task, which generates the human captions given both the visual inputs and all of its human captions, to advise the hidden states in the student caption decoders.

2.2 Autoencoders

Autoencoders (Hinton and Salakhutdinov 2006) learn to compress input data into dimension-reduced features using an encoder and learn a decoder to reconstruct the original input data from the the hidden state. For image captioning, the most relevant research (Yang et al. 2018) uses an autoencoder to construct scene graphs using relations between pairs of objects to represent the complex structure of both image and sentence before generating the caption. In contrast, we utilize autoencoders to learn richer teacher hidden states conditioned on both the image and human captions. Finally, we train the student caption decoder to produce similar hidden states without the human captions as input.

2.3 Teacher-Student Networks

The teacher-student method (Romero et al. 2014) transfers knowledge from a shallower and wider teacher CNN to a deeper and thinner student CNN by minimizing the divergence between the two output probabilities. Kim and Rush (2016) extend this method to train RNNs for machine translation. In our framework, instead of using the same training data, we allow the teacher network (an autoencoder) access to additional information ($i.e.$ human captions), resulting in a richer representation to advise the student caption decoder.

3 Approach

This section presents the details of HSG. We first present the overall architecture in Sec. 3.1, then describe three student caption decoders in Sec. 3.2 to illustrate that HSG can be applied to almost any RNN-based decoder. After that, we explain the teacher autoencoder in Sec. 3.3 and the state transformation network to estimate the initial teacher hidden state only from the visual inputs in Sec. 3.4.

3.1 Overview

The goal of HSG is to provide hidden state guidance to any conventional RNN-based caption decoder, which we regard as a student network, as shown in Figure 2. In order to collect the guidance, we first train a teacher on an easier task that uses images to help autoencode human captions, which shares the same architecture as the student decoder. Then, we utilize a state transformation network to estimate the teacher decoder's initial hidden states (t=0) using only the visual input. These approximations are used to initialize the student decoder's hidden states so that it is capable of directly generating captions from images.

3.2 Student Caption Decoder

In this section, we briefly present three RNN-based student caption decoders.

FC. This model (Vinyals et al. 2015) adopts a single layer LSTM as the caption decoder. For the visual input features, we first feed the full image to a deep CNN ($i.e.$ ResNet-101), and then average-pool the features from the final convolutional layer, resulting in a 2048-d vector for each image. The words are encoded using a trainable embedding matrix. At each time step, the LSTM receives the previous hidden states, generated words, and the visual features to generate the current word. Please refer to (Vinyals et al. 2015) for details.

Up-Down. This model (Anderson et al. 2018) incorporates object detection features into image captioning. The caption decoder operates on features of detected objects extracted using Faster RCNN (Ren et al. 2015) with a ResNet-101 (He et al. 2016) base network. It consists of a two-layer LSTM, where the first LSTM learns to distinguish important objects for generating the current word using an attention mechanism, and the second LSTM encodes the sequential information from the attended features to compute the output word probabilities.

SGAE. This model (Yang et al. 2018) incorporates *scene graphs* for both sentences and images into image captioning. Specifically, they first encode a sentence scene graph to reconstruct human captions and learn an additional memory matrix. Then, they use visual scene graph features with the learned memory to generate captions via a two-layer LSTM with attention similar to Up-Down.

3.3 Teacher Autoencoder

Our teacher autoencoder is trained to generate captions using not only visual input features, but also the set of human captions for the image. The teacher has two components: (1) a caption encoder to learn the joint representation of both the human captions and the visual inputs, and (2) a

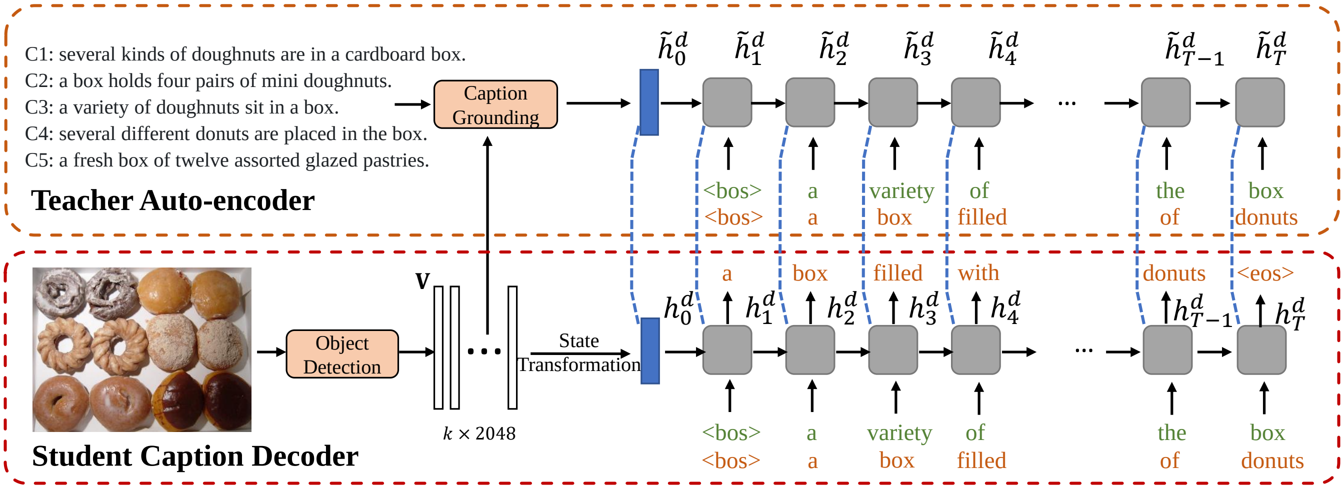

Figure 2: Our framework consists of two parts. First, we train a teacher autoencoder that compresses the captions using the image as context. Second, the student decoder receives hidden state guidance from the teacher. Green captions present the maximum-likelihood training process, where each word from the human captions is fed to the network, and orange captions presents the REINFORCE case, where the previous generated word is fed to the caption decoder. Blue dashed lines indicate the hidden states loss.

decoder to generate captions from the output of the encoder and the visual features.

Caption Encoder

Our caption encoder takes as input the image feature set $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_K\}$ consisting of K vectors for K detected objects, C human captions $\mathbf{W}_i^c = \{w_{i,1}^c, w_{i,2}^c, ..., w_{i,T}^c\}$, where T denotes the length of the captions and $i = 1, ..., C$ are the caption indices.

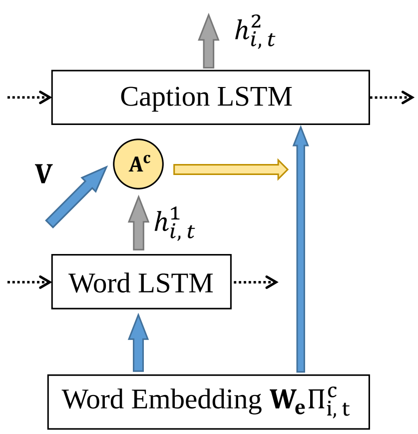

Figure 3: Overview of the caption encoder. The Word LSTM is used to generate attention to identify the key words in each caption, and the Caption LSTM generates the final caption embedding. Blue arrows denote fc layers and yellow arrows attention embedding.

Inspired by (Wu, Hu, and Mooney 2019), we use a two-layer LSTM architecture to encode human captions as illustrated in Figure 3. The first-layer LSTM (called the Word LSTM) sequentially encodes the words in a caption $\mathbf{W}_i^c$ at each time step as $h_{i,t}^{e,1}$.

$$h_{i,t}^{e,1}, c_{i,t}^{e,1} = \text{LSTM}(\mathbf{W}_e \Pi_{i,t}^c,\ h_{i,t-1}^{e,1},\ c_{i,t-1}^{e,1}) \tag{1}$$

where $\mathbf{W}_e$ is the 300-d word embedding matrix, and $\Pi_{i,t}^c$ is the one-hot embedding for the word $w_{i,t}^c$.

Then, we design a caption attention module $\mathbf{A}^c$ which utilizes the image feature set $\mathbf{V}^q$, and $h_{i,t}^{e,1}$ to generate the attention weight on the current word in order to indicate its importance. Specifically, the Word LSTM first encodes the word embedding $\Pi_{i,t}^c$ in Eq. 1. Then we feed the outputs $h_{i,t}^{e,1}$ and $\mathbf{V}$ to the attention module $\mathbf{A}^c$ as shown in Eq. 2.

$$\alpha_{i,j,t}^c = \text{softmax}(h_{i,t}^{e,1} \circ f(\mathbf{v}_j))) \tag{2}$$

where the softmax function is over the K objects in visual feature set $\mathbf{V}$.

Next, the attended word representations w_e in the caption are used to produce the final caption representation in Eq. 4 via the Caption LSTM.

$$w_e = \max_j \{\alpha_{i,j,t}^c\} \mathbf{W}_e \Pi_{i,t}^c \tag{3}$$

$$h_{i,t}^{e,2}, c_{i,t}^{e,2} = \text{LSTM}(w_e,\ h_{i,t-1}^{e,2},\ c_{i,t-1}^{e,2}) \tag{4}$$

where max denotes the element-wise max pooling over the attention weights for the K objects in the image.

Caption Decoder

Since the goal of the teacher caption decoder is to provide hidden state guidance to the student decoder, we require these two decoders to have the same architecture. However, the differences between these two decoders lie in the initial hidden states. The teacher decoder is initialized with the outputs from the encoder while the student caption decoder is initialized with an estimated version as detailed in Sec. 3.4.

For the FC caption decoder, we use the max pooling of the final hidden state from the second LSTM in the caption encoder as the initial state, $i.e.$ $\tilde{h}_0^d = \max_c\{h_{c,T}^{e,2}\}$. Similarly, we max pool the final hidden states from both layers to initialize the hidden state for the two LSTMs in the Up-Down and SGAE caption decoders. Specifically, the teacher initial hidden state is computed as $\tilde{h}_0^d = [\max_c\{h_{c,T}^{e,1}\}; \max_c\{h_{c,T}^{e,2}\}]$. After initialization of the initial states $\tilde{h}_0^d$, the student decoder is trained to match the teacher states $\tilde{h}_t^d$ for each time step t.

3.4 State Transformation Network

The state transformation network uses the visual features to estimate the initial teacher hidden states $\tilde{h}_0^d$ so that the student caption decoder is capable of using the estimated hidden states to start a sentence purely from the visual inputs alone. For efficiency, we simply use a two-layer *fc* network for state transformation. For the FC decoder, we directly apply the two-layer networks to the visual feature vector to estimate the initial hidden states $h_0^d = f(f(\mathbf{v}))$, where the $\mathbf{v}$ is the $2,048$-d features for the image. For the Up-Down and the SGAE decoder, the first *fc* layer adapts the visual features for each object and the second *fc* layer estimates the hidden states using the sum of the first layer's output over all objects or nodes in the scene graph. Specifically the hidden states are computed as $h_0^d = f(\sum_{j=1}^{K} f(\mathbf{v}_j))$.

4 Training

Training involves pre-training the teacher autoencoder and then training the student caption generator using maximum likelihood or REINFORCE. We use θ_α to denote the parameters in the autoencoder ($i.e.$ the caption grounding encoder and the teacher caption decoder), and θ_g to denote the parameters in the state transformation network and the student decoder. We use c to denote the entire caption, c_t to denote the t-th word in the caption, and $c_{\leqslant t}$ to denote the first t words in the caption. We omit the visual features $\mathbf{v}$ in all of probabilities in this section for simplicity. We denote the maximum likelihood loss using parameters θ as $\mathcal{L}_{ll}(\theta)$ defined in Eq. 5:

$$\mathcal{L}_{ll}(\theta) = -\sum_{t=1}^{T} \log(p(c_t|c_{\leqslant t-1};\theta)) \tag{5}$$

4.1 Pretraining the Teacher Autoencoder

We pre-train the teacher autoencoder using cross-entropy loss, minimizing $\mathcal{L}_{ll}(\theta_\alpha)$ defined in Eq. 5. After pre-training, the parameters θ_α are fixed.

Additionally, we pre-train the state transformation network using $\mathcal{L}_{s,t}(\theta_g)$, $t = 0$ as defined in Eq. 6 using L2 distance.

$$\mathcal{L}_{s,t}(\theta_g) = \|h_t^d - \tilde{h}_t^d\|_2^2 \tag{6}$$

In particular, the generated captions from the student decoder are fed to the teacher autoencoder to compute the teacher hidden states at each time (t) as shown in Fig 2. We will omit "(θ_g)" from $\mathcal{L}_{s,t}(\theta_g)$ for simplicity.

4.2 Training the Student Decoder

We tested two different approaches to training the student decoders using either maximum likelihood or REINFORCE (with various evaluation metrics as rewards). The student decoder is initialized with the teacher decoder's parameters.

Maximum Likelihood Training
Maximum likelihood trains the student decoder to maximize the word-level log-likelihood, where human captions are fed into the decoder to compute the next word's probability distribution. We use the joint loss in Eq. 7:

$$\mathcal{L} = \mathcal{L}_{ll}(\theta_g) + \lambda \sum_{t=0}^{T} \mathcal{L}_{s,t} \tag{7}$$

With human captions as input to the teacher autoencoder, we compute its hidden state, which is needed to calculate $\mathcal{L}_{s,t}$. The λ parameter controls the weights of the state loss.

REINFORCE
An alternative to log-likelihood maximization is to fine-tune the model to directly maximize the expected evaluation metric using REINFORCE. Negative rewards, such as BLEU (Papineni et al. 2002) or CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015), are minimized using $\mathcal{L} = -\mathbb{E}_{\hat{c}\sim p_{\theta_g}}[\tilde{r}(\hat{c})]$ where $\tilde{r}(\hat{c}) = r(\hat{c}) - r(c^\star)$ denotes the variance-reduced rewards (Rennie et al. 2017), $\hat{c}$ denotes the sampled captions using the probabilities over the vocabulary, and $c^\star$ denotes greedily sampled captions using the word with the maximum probability. We will omit "θ_g" from p_{θ_g} for simplicity. The parameters in the student caption decoder are updated using the policy gradients $\nabla_{\theta_g}\mathcal{L} = -\mathbb{E}_{\hat{c}\sim p}\Big[\tilde{r}(\hat{c})\nabla_{\theta_g}\log p(\hat{c})\Big]$.

However, one remaining problem with this approach is that the sentence-level reward $\tilde{r}(\hat{c})$ is equally distributed over each word in the sampled captions, no matter how relevant the word is. Therefore, some desired words will not get enough credit because of the presence of some unrelated or inaccurate words in the sentence. To address this issue, we propose to use our hidden state loss as an intermediate reward to encourage the student decoder to produce hidden states that match the hidden states of the high-performing teacher decoder. We add a reward objective $\tilde{\mathcal{R}}$ that is the accumulated expectation of the negative hidden state losses over time (t) as shown in Eq. 8:

$$\tilde{\mathcal{R}} = -\sum_{t=0}^{T} \mathbb{E}_{\hat{c}_{\leqslant t}\sim p}[\mathcal{L}_{s,t}] \tag{8}$$

Therefore, the new policy gradients can be written as Eq. 9, and we provide detailed derivation in the supplementary

	Maximum Likelihood					REINFORCE (CIDEr)				
Model	B-4	M	R-L	C	S	B-4	M	R-L	C	S
LSTM-A (Yao et al. 2017)	35.2	26.9	55.8	108.8	20.0	35.5	27.3	56.8	118.3	20.8
StackCap (Gu et al. 2018)	35.2	26.5	-	109.1	-	36.1	27.4	56.9	120.4	20.9
FC (Vinyals et al. 2015)	32.9	25.0	**54.0**	95.4	17.9	32.8	25.0	54.2	104.0	**18.5**
FC + HSG	**33.2**	**25.5**	53.9	**96.1**	**18.3**	**33.9**	**25.9**	**54.8**	**107.5**	18.4
Up-Down (Anderson et al. 2018)	**36.0**	27.0	56.3	113.1	20.4	36.3	27.5	56.8	120.7	21.4
Up-Down + HSG	35.6	**27.3**	**56.7**	**113.9**	**20.6**	**37.4**	**28.0**	**57.7**	**124.0**	**21.5**
SGAE [2](Yang et al. 2018)	35.8	28.0	56.7	114.0	20.9	37.8	28.2	58.2	125.9	**22.1**
SGAE + HSG	35.9	28.1	56.9	115.2	21.0	**38.5**	**28.4**	**58.5**	**127.7**	21.8

Table 1: Automatic evaluation comparisons with various baseline caption decoders on the Karpathy test set. "HSG" denotes trained with hidden state guidance, B-4, M, R-L, C and S are short hands for BLEU-4, METEOR, ROUGE-L, CIDEr and SPICE. All captions are generated with beam search (beam size=5) except for SGAE decoders that use beam size = 2

materials.

$$\nabla_{\theta_g}\tilde{\mathcal{L}} = \nabla_{\theta_g}(\mathcal{L}+\lambda\tilde{\mathcal{R}}) =$$
$$\underbrace{\mathbb{E}_{\hat{c}\sim p}\Big[\sum_{\tau=0}^{T}\Big(\lambda\sum_{t=\tau}^{T}\mathcal{L}_{s,t}-\tilde{r}(\hat{c})\Big)\nabla_{\theta_g}\log p(\hat{c}_\tau|\hat{c}_{<\tau})\Big]}_{\text{Reward Term}}$$
$$+\,\lambda\,\underbrace{\mathbb{E}_{\hat{c}\sim p}\Big[\sum_{t=0}^{T}\nabla_{\theta_g}\mathcal{L}_{s,t}\Big]}_{\text{Punishing Term}} \quad (9)$$

It is worth noting that unlike the reward $\tilde{r}(\hat{c})$, the hidden state losses $\mathcal{L}_{s,t}$ are differentiable in the parameters θ_g, which is necessary to compute the policy gradients. Intuitively, Eq. 9 can be understood as simultaneously rewarding the student caption decoder when it produces the hidden states that match the teacher hidden states (the second line), and punishing the hidden states that don't match (the third line). In practice, following (Rennie et al. 2017; Luo et al. 2018), we sample 5 sentences (N=5) per image to approximate the expectation.

Word-level Intermediate Rewards. HSG also provides a word-level intermediate reward for more efficient learning. To illustrate this, we differentiate the total loss $\mathcal{L}$ with respect to each output logit[3] s_τ at time τ. As shown in Eq. 10, the reward $\tilde{r}(\hat{c})$ has to be delayed until the end of the caption, but the output logits s_τ are able to collect the rewards for matching the hidden states with the teacher decoders' from time τ.

$$\nabla_{s_\tau}\tilde{\mathcal{L}} = \quad (10)$$
$$\mathbb{E}_{\hat{c}\sim p}\Big[\Big(\lambda\sum_{t=\tau}^{T}\mathcal{L}_{s,t}-\tilde{r}(\hat{c})\Big)\Big(p(\hat{c}_\tau|\hat{c}_{<\tau})-\mathbf{1}_{\hat{c}_\tau}\Big)\Big]$$

where the $\mathbf{1}_{\hat{c}_\tau}$ denotes a vector with the dimension of the vocabulary size where all elements except the $\hat{c}_\tau$-th are 0 and the $\hat{c}_\tau$-th is 1.

Implementation Details

We train our model using the Adam optimizer (Kingma and Ba 2015) with a batch size of 32 for training FC and Up-Down decoders and 16 for training SGAE decoders on a single 12G Titan V card. Following Luo et al. (2018), the learning rate is initialized to 5e-4 and decayed by a factor of 0.8 every five epochs. For the FC decoders, we use the average pooling of the last convolutional layer in the ResNet-101 (He et al. 2016) pre-trained on ImageNet. For the Up-Down decoders, following Anderson et al. (2018), we use at most 100 object detection features for each image. We use a Faster R-CNN head (Ren et al. 2015) in conjunction with a ResNet-101 base network as our detection module. The detection head is first pre-trained on Visual Genome (Krishna et al. 2017). Both the FC and Up-Down decoders are implemented in the same open source framework from Luo et al.[4]. For SGAE decoders, we use the settings and code from (Yang et al. 2018).[5] For captioning models, the dimension of the LSTM hidden state, image feature embedding, attention embedding and word embedding are all set to 512. We also use Glove vectors (Pennington, Socher, and Manning 2014) to initialize the word embedding matrix in the caption encoder. During training, we first pretrain the teacher autoencoder using Eq. 5 for 20 epochs. After that, the student caption decoder is initialized with parameters from the teacher autoencoder, and trained using maximum likelihood ($i.e.$ Eq. 7) for 20 epochs. Finally, we fine-tune the student decoder using REINFORCE ($i.e.$ Eq. 9) for 30 epochs. During testing, we use beam search to sample captions using a beam size of 5 when using FC and Up-Down decoders and 2 when using SGAE decoders.

5 Experimental Evaluation

In this section, we verify the effectiveness of HSG using both standard automatic metrics and human evaluation.

[2]We use a smaller batch size than the original implementation, leading to the performance drop (16 vs 100)

[3]The input to the softmax function

[4]`https://github.com/ruotianluo/self-critical.pytorch`

[5]`https://github.com/yangxuntu/SGAE`

	REINFORCE					REINFORCE + HSG				
Model	B-4	M	R-L	C	S	B-4	M	R-L	C	S
Up-Down (B-4)	**37.5**	26.9	57.0	111.0	20.3	**38.5**	27.3	57.6	112.4	20.5
Up-Down (M)	31.4	**28.7**	56.1	105.0	22.0	33.0	**29.5**	57.2	110.4	**22.3**
Up-Down (R-L)	36.5	26.6	**57.9**	114.2	19.8	36.9	27.2	**58.7**	115.7	20.2
Up-Down (C)	36.3	27.5	56.8	**120.7**	21.4	37.4	28.0	57.7	**124.0**	21.5

Table 2: Evaluation on Up-Down decoder's performance on Karpathy test set with various evaluation metrics as rewards when training using REINFORCE. B-4, M, R-L, C and S are short hands for BLEU-4, METEOR, ROUGE-L, CIDEr and SPICE.

5.1 Dataset

We use the MSCOCO 2015 dataset (Chen et al. 2015) for image captioning. In particular, we use the Karpathy configuration that includes 110K images for training and 5K images each for validation and test. Each image has 5 human caption annotations. Similar to Anderson et al. (2018), we first convert all sentences to lower case, tokenizd on white spaces, and filter words that occur less than 5 times.

5.2 Results on Image Captioning

Comparison with the Base Decoders. In Table 1, we present the standard automatic evaluation for FC, Up-Down and SGAE decoders trained using either Maximum Likelihood alone or using REINFORCE with CIDEr rewards and compare them with the state-of-the-art image captioner. Metrics included are BLEU-4 (Papineni et al. 2002), METEOR (Banerjee and Lavie 2005), ROUGE-L (Lin 2004), CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015) and SPICE (Anderson et al. 2016). The left part reports the results trained with maximum likelihood and the right part shows the results using REINFORCE. We also include an ensemble model, where each member is trained using REINFORCE with a different random seed. We observe that when training with maximum likelihood, our model outperforms both of the baseline decoders for all of the metrics, except ROUGE-L when using FC, and BLEU-4 when using Up-Down, demonstrating the effectiveness of introducing our hidden state guidance. More importantly, we observe a significant improvement on the CIDEr scores over all of the baseline models using REINFORCE (i.e. 107.5 v.s. 104.0 using FC Model, 124.0 v.s. 120.7 using FC Model and 127.7 v.s. 125.9 using SGAE Model). We attribute the improvements to both HSG and the word-level intermediate rewards.

Teacher Autoencoder Performance In this section, we report automatic evaluation results for FC, Up-Down and SGAE teacher autoencoders. As shown in Table 3, all of the

	Maximum Likelihood				
	B	M	R-L	C	S
FC	47.1	30.0	65.0	137.4	23.9
Up-Down	52.1	35.7	68.5	146.7	25.1
SGAE	57.6	35.5	68.6	157.6	29.3

Table 3: Teacher autoencoder performance.

FC, Up-Down and SGAE teacher autoencoders are able to achieve significantly better performance compared to the corresponding student caption decoders, even when trained with REINFORCE, indicating that the teacher hidden states are able to provide suitable guidance.

Results on Various Metrics as Rewards In Table 2, we report the results using Up-Down decoders with various evaluation metrics as the reward for REINFORCE. Specifically, we initialize the REINFORCE training process using the model trained with maximum likelihood with and without hidden states guidance, and train for another 30 epochs with a batch size of 32. The left and right parts of Table 2 report the performance on all of the automatic evaluation metrics with and without HSG, respectively. We observe that the models trained with HSG are consistently able to better optimize the corresponding reward metrics.

	REINFORCE (CIDEr)				
	B	M	R-L	C	S
$\lambda = 0.0$	36.3	27.5	56.8	120.7	21.4
$\lambda = 0.05$	37.4	28.0	57.7	124.0	21.4
$\lambda = 0.2$	37.1	27.8	57.5	123.5	21.5
$\lambda = 1.0$	37.0	27.5	57.4	123.2	21.3
$\lambda = 2.0$	36.8	27.5	57.5	122.9	21.0

Table 4: Results on Karpathy test set with various hidden state loss weights λ.

Ablation Study on Hidden State Weight Table 4 shows results when we vary the hidden state loss weight λ from 0.0 to 2.0 using Up-Down decoders. Our model consistently outperforms the baseline model (λ = 0) on all evaluation metrics when using CIDEr rewards. We also observe that the CIDEr scores are fairly robust to the exact setting of λ.

Human Evaluation We conducted an Amazon Mechanical Turk (AMT) evaluation by asking human judges to directly compare captions from HSG and the baseline models. Following (Wu and Mooney 2018; Park et al. 2018; Venugopalan et al. 2017), we use a ranking-based approach with majority voting. In particular, we randomly chose 1,000 images and generated captions with and without HSG trained with CIDEr rewards using Up-Down with a beam size of 5. The two captions for each image were randomly

Up-Down: A church with a large cathedral in front of a building. (CIDEr: 0.884)

Up-Down + HSG: A church with a bunch of pew in front of it. (CIDEr: 1.449)

Up-Down: A sitting at a table eating a sandwich. (CIDEr: 0.152)

Up-Down + HSG: A man holding a plate of food on a table. (CIDEr: 1.784)

Up-Down: A couple of people sitting on a boat. (CIDEr:1.106)

Up-Down + HSG: Two little girls sitting on a boat in the water. (CIDEr:3.115)

Up-Down: A fighter jet flying in the blue sky. (CIDEr:1.166)

Up-Down + HSG: A fighter jet flying through a blue sky. (CIDEr:1.686)

Up-Down: A plate of food with meat and vegetables. (CIDEr:1.132)

Up-Down + HSG: A plate of food with meat and carrots on a table.(CIDEr:1.522)

Up-Down: A flock of birds sitting on top of a tree. (CIDEr:0.028)

Up-Down + HSG: A teddy bear sitting on top of a tree.(CIDEr:1.256)

Figure 4: Sampled generated captions using Up-Down decoders with and without HSG. The captions are sampled using beam search with a beam size of 5.

ordered and we asked workers to rank them in terms of descriptiveness, allowing for ties. We test each image with two captions with 3 different judges and the final rankings are determined as follows. For each human judgement, we assign $+1$ to the better caption, 0 if they are tied and -1 to the worse one. Then, we compute the average scores for the two captions to decide their final rankings. In Table 5, we report the percentage of our captions that are better than, equivalent to, or worse than the captions without HSG. We observe that our captions are better more than 50% of the time and worse less than 30%, indicating that HSG is effective at improving captioning.

	Better	Equivalent	Worse
Up-Down	54.3 %	16.1 %	29.6%

Table 5: Percentage of our captions that are better than, equivalent to, or worse than those without HSG.

Figure 4 shows the qualitative comparison between some sampled captions using the Up-Down student decoders trained using CIDEr as rewards with or without hidden state guidance. We empirically find that the captions generated with HSG are able to capture more subtle visual concepts and relationships between objects, resulting in more descriptive captions.

Comparison with Professor Forcing In this section, we compare our approach with Professor Forcing (Lamb et al. 2016). The professor decoder receives one of the human captions as input and shares parameters with the student decoder, which uses the generated previous word as input. We use a single-layer GRU with 512 hidden units as the discriminator.

Table 6 reports the performance of Up-Down decoder using maximum likehood and REINFORCE with CIDEr rewards. We observe obvious improvements over Professor Forcing in both the maximum likelihood and REINFORCE cases. Unlike Professor Forcing, our teacher has access to more information (*i.e.* the initial hidden state that encodes the set of human captions) than the student, and therefore possesses a richer hidden state representation to guide the student.

	Maximum Likelihood				
	B	M	R-L	C	S
Professor	35.8	27.0	56.1	113.3	20.3
HSG	35.6	27.3	56.7	113.9	20.6
	REINFORCE (CIDEr)				
	B	M	R-L	C	S
Professor	36.4	27.4	57.0	120.9	21.4
HSG	37.4	28.0	57.7	124.0	21.5

Table 6: Comparison with Professor Forcing using Up-Down decoders on Karpathy test set.

6 Conclusion and Future Work

We have presented a novel image captioning framework that uses an image-conditioned caption autoencoder. This teacher autoencoder, which has the same architecture as the original RNN-based decoder, is trained using an easier task that learns to generate image captions with gold standard human captions as inputs in addition to visual features. This austoencoder is used to guide the hidden state representation learned by the student caption decoder. We integrated this hidden state guidance into both maximum likelihood and REINFORCE using three state-of-the-art image captioners. We observe that especially in the REINFORCE case, the word-level hidden state guidance assigns an intermediate reward that emphasizes the most relevant words. Extensive experimental results demonstrate the effectiveness of our approach. In the future, we would like to explore metrics that can measure and minimize the semantic difference between the teacher and student hidden states better than L2, this could potentially further improve the learned hidden-state representation.

References

[Anderson et al. 2016] Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. Spice: Semantic Propositional image caption evaluation. In *ECCV*, 382–398.

[Anderson et al. 2018] Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-Up and Top-Down Attention for Image Captioning and VQA. In *CVPR*, volume 3, 6.

[Banerjee and Lavie 2005] Banerjee, S., and Lavie, A. 2005. Meteor: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72.

[Chen et al. 2015] Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*.

[Cho et al. 2014] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*.

[Donahue et al. 2015] Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *CVPR*, 2625–2634.

[Gu et al. 2018] Gu, J.; Cai, J.; Wang, G.; and Chen, T. 2018. Stack-captioning: Coarse-to-fine learning for image captioning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

[Hinton and Salakhutdinov 2006] Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science* 313(5786):504–507.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural computation* 9(8):1735–1780.

[Karpathy and Fei-Fei 2015] Karpathy, A., and Fei-Fei, L. 2015. Deep Visual-semantic Alignments for Generating Image Descriptions. In *CVPR*, 3128–3137.

[Kim and Rush 2016] Kim, Y., and Rush, A. M. 2016. Sequence-level knowledge distillation. In *ACL*.

[Kingma and Ba 2015] Kingma, D. P., and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[Krishna et al. 2017] Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision* 123(1):32–73.

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

[Lin 2004] Lin, C.-Y. 2004. Rouge: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*.

[Liu et al. 2018] Liu, X.; Li, H.; Shao, J.; Chen, D.; and Wang, X. 2018. Show, Tell and Discriminate: Image Captioning by Self-retrieval with Partially Labeled Data. *ECCV*.

[Luo et al. 2018] Luo, R.; Price, B.; Cohen, S.; and Shakhnarovich, G. 2018. Discriminability Objective for Training Descriptive Captions. In *CVPR*.

[Papineni et al. 2002] Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In *ACL*, ACL '02.

[Park et al. 2018] Park, D. H.; Hendricks, L. A.; Akata, Z.; Rohrbach, A.; Schiele, B.; Darrell, T.; and Rohrbach, M. 2018. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *CVPR*.

[Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

[Ren et al. 2015] Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *NIPS*, 91–99.

[Rennie et al. 2017] Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical Sequence Training for Image Captioning. In *CVPR*, volume 1, 3.

[Romero et al. 2014] Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

[Vedantam, Lawrence Zitnick, and Parikh 2015] Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-Based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4566–4575.

[Venugopalan et al. 2017] Venugopalan, S.; Anne Hendricks, L.; Rohrbach, M.; Mooney, R.; Darrell, T.; and Saenko, K. 2017. Captioning images with diverse objects. In *CVPR*, 5753–5761.

[Vinyals et al. 2015] Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and Tell: A Neural Image Caption Generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 3156–3164. IEEE.

[Wu and Mooney 2018] Wu, J., and Mooney, R. J. 2018. Faithful Multimodal Explanation for Visual Question Answering. *arXiv preprint arXiv:1809.02805*.

[Wu, Hu, and Mooney 2019] Wu, J.; Hu, Z.; and Mooney, R. J. 2019. Generating Question Relevant Captions to Aid Visual Question Answering. *arXiv preprint arXiv:1906.00513*.

[Xu et al. 2015] Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, 2048–2057.

[Yang et al. 2018] Yang, X.; Tang, K.; Zhang, H.; and Cai, J. 2018. Auto-encoding graphical inductive bias for descriptive image captioning. *arXiv preprint arXiv:1812.02378*.

[Yao et al. 2017] Yao, T.; Pan, Y.; Li, Y.; Qiu, Z.; and Mei, T. 2017. Boosting image captioning with attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, 4894–4902.

[Yao et al. 2018] Yao, T.; Pan, Y.; Li, Y.; and Mei, T. 2018. Exploring visual relationship for image captioning. In *ECCV*, 684–699.