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

Jialin Wu

Department of Computer Science
University of Texas at Austin
jialinwu@utexas.edu

Raymond J. Mooney

Department of Computer Science
University of Texas at Austin
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 or detected objects as inputs.

Most captioning research [18, 5, 8, 15, 2, 21, 19] trains an RNN-based decoder to learn the output word probabilities conditioned on the previous hidden state and various visual features. Recent methods improve results by incorporating richer visual inputs from object detection [2] and relationship detection [21, 19].

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 [14] or BLEU [10] 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 until 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.

We present a new framework, called Hidden State Guidance (HSG), that treats the RNN caption decoder as a student network [13] 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 additional 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. HSG plays a particularly helpful role when training using REINFORCE since it also provides a word-level intermediate reward that highlights the important words. Our general framework can be used in almost any RNN-based image captioner. Experimental results show significant improvements over two recent caption decoders, FC [12] using image features and Up-Down [2] using object-detection features.

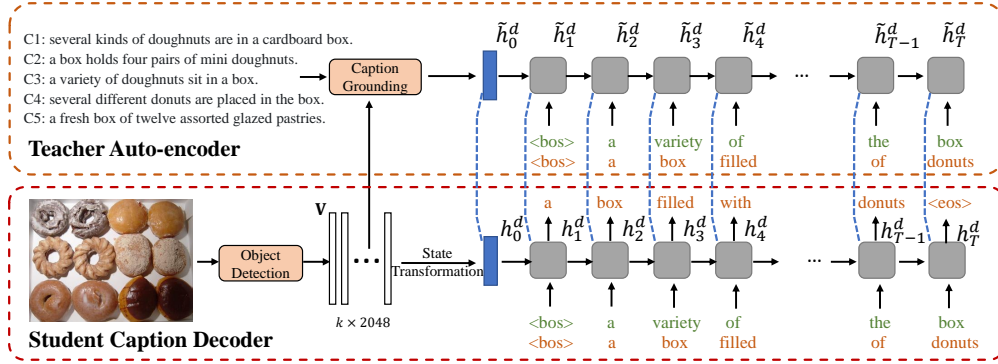


Figure 1: Our framework consists of two parts. First, we train a teacher autoencoder that compresses the captions using the image as context. Second, the student decoder receives hidden state guidance from the teacher. Green captions present the maximum-likelihood training process, where each word from 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.

1 Approach

We first present the overall architecture, then describe two student caption decoders, illustrating that HSG can be applied to almost any RNN-based decoder. After that, we explain the teacher autoencoder and the state transformation network that estimates the initial teacher hidden state from the visual inputs.

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 1. 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.

Student Caption Decoder We briefly present two RNN-based student caption decoders.

FC. This model [15] adopts a single layer LSTM as the caption decoder. We first feed the full image to a deep CNN, and then average-pool the features from the final layer as visual features. 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.

Up-Down. This model [2] incorporates object detection features and has been widely adopted by recent research [2, 16, 21, 19]. The caption decoder operates on features of detected objects extracted using Faster RCNN [11] with a ResNet-101 [7] 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 sequentially encodes the attended features to compute the output word probabilities.

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.

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

Inspired by [17], we use a two-layer LSTM architecture to encode human captions as illustrated in Figure 2. The first-layer LSTM (called the Word LSTM) sequentially encodes the words in a caption \mathbf{W}_i^c at each time step as $h_{i,t}^{e,1}$:

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

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

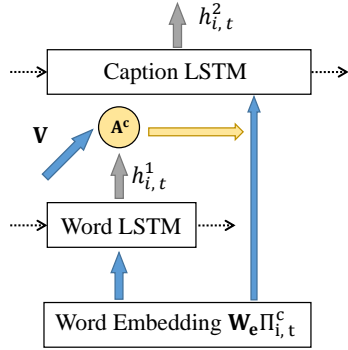


Figure 2: Overview of the caption encoder. The Word LSTM generates attention to identify the key words in each caption, and the Caption LSTM generates the final caption embedding. Blue arrows are fully-connected layers and yellow arrows are attention.

Similarly, we max pool the final hidden states from both layers to initialize the hidden states for the LSTMs in the Up-Down decoders.

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 $h_0^d = f(f(\mathbf{v}))$.

2 Training

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

Pretraining the Teacher Autoencoder. We use cross-entropy loss, minimizing $\mathcal{L}_{ll}(\theta_\alpha)$. After pre-training, the parameters θ_α are fixed. Additionally, we pre-train the state transformation network using $\mathcal{L}_{s,t}(\theta_g) = \|h_t^d - \tilde{h}_t^d\|_2^2$, $t = 0$. 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 1. We will omit “ (θ_g) ” from $\mathcal{L}_{s,t}(\theta_g)$ for simplicity.

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 $\mathcal{L} = \mathcal{L}_{ll}(\theta_g) + \lambda \sum_{t=0}^T \mathcal{L}_{s,t}$. 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 weight of the state loss.

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 . In particular, $\alpha_{i,j,t}^c = \text{softmax}(h_{i,t}^{e,1} \circ f(\mathbf{v}_j))$, 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. 3 via the Caption LSTM.

$$w_e = \max_j \{\alpha_{i,j,t}^c\} \mathbf{W}_e \Pi_{i,t}^c \quad (2)$$

$$h_{i,t}^{e,2}, c_{i,t}^{e,2} = \text{LSTM}(w_e, h_{i,t-1}^{e,2}, c_{i,t-1}^{e,2}) \quad (3)$$

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

Caption Decoder. We require these two decoders to have the same architecture to provide hidden state guidance. The differences between these two decoders are the initial hidden states. The teacher decoder is initialized with the encoders’ output while the student caption decoder is initialized with an estimated version. For the FC decoder, we use the max pooling of the final hidden state from the second LSTM in the caption encoder as the initial state.

Model	Maximum Likelihood					REINFORCE (CIDEr)				
	B-4	M	R-L	C	S	B-4	M	R-L	C	S
LSTM-A [20]	35.2	26.9	55.8	108.8	20.0	35.5	27.3	56.8	118.3	20.8
StackCap [6]	35.2	26.5	-	109.1	-	36.1	27.4	56.9	120.4	20.9
FC [15]	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 [2]	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

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 size 5.

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 [10] or CIDEr [14], are minimized using $\mathcal{L} = -\mathbb{E}_{\hat{c} \sim p_{\theta_g}}[\tilde{r}(\hat{c})]$ where $\tilde{r}(\hat{c}) = r(\hat{c}) - r(c^*)$ denotes the variance-reduced rewards [12], \hat{c} denotes the sampled captions using the probabilities over the vocabulary, and c^* 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 $\nabla_{\theta_g} \mathcal{L} = -\mathbb{E}_{\hat{c} \sim p}[\tilde{r}(\hat{c}) \nabla_{\theta_g} \log p(\hat{c})]$.

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 $\tilde{\mathcal{R}} = -\sum_{t=0}^T \mathbb{E}_{\hat{c}_{\leq t} \sim p}[\mathcal{L}_{s,t}]$ that is the accumulated expectation of the negative hidden state losses over time (t). Therefore, the new policy gradients are: $\nabla_{\theta_g} \tilde{\mathcal{L}} = \mathbb{E}_{\hat{c} \sim p}[\sum_{\tau=0}^T (\lambda \sum_{t=\tau}^T \mathcal{L}_{s,t} - \tilde{r}(\hat{c})) \nabla_{\theta_g} \log p(\hat{c}_\tau | \hat{c}_{<\tau})] + \lambda \mathbb{E}_{\hat{c} \sim p}[\sum_{t=0}^T \nabla_{\theta_g} \mathcal{L}_{s,t}]$.

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, the new policy gradients can be understood as rewarding the student caption decoder when it produces hidden states that match the teacher’s hidden states, and punishing it when the hidden states don’t match.

3 Experimental Evaluation

Dataset. We use the MSCOCO 2015 dataset [4] 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. We convert all sentences to lower case, tokenized on white spaces, and filter words that occur less than 5 times.

Comparison with the Base Decoders. In Table 1, we present the standard automatic evaluation for FC, Up-Down decoders trained using either Maximum Likelihood alone or using REINFORCE with CIDEr rewards. Metrics included are BLEU-4 [10], METEOR [3], ROUGE-L [9], CIDEr [14] and SPICE [1]. 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). We attribute the improvements to both HSG and the word-level intermediate rewards.

4 Conclusion

We have presented a novel image captioning framework that uses an image-conditioned caption auto-encoder. 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.

References

- [1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic Propositional image caption evaluation. In *ECCV*, pages 382–398, 2016.
- [2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-Up and Top-Down Attention for Image Captioning and VQA. In *CVPR*, volume 3, page 6, 2018.
- [3] S. Banerjee and A. Lavie. 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*, pages 65–72, 2005.
- [4] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*, 2015.
- [5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *CVPR*, pages 2625–2634, 2015.
- [6] J. Gu, J. Cai, G. Wang, and T. Chen. Stack-captioning: Coarse-to-fine learning for image captioning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [7] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016.
- [8] A. Karpathy and L. Fei-Fei. Deep Visual-semantic Alignments for Generating Image Descriptions. In *CVPR*, pages 3128–3137, 2015.
- [9] C.-Y. Lin. Rouge: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 2004.
- [10] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A Method for Automatic Evaluation of Machine Translation. In *ACL, ACL '02*, 2002.
- [11] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *NIPS*, pages 91–99, 2015.
- [12] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical Sequence Training for Image Captioning. In *CVPR*, volume 1, page 3, 2017.
- [13] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.
- [14] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-Based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.
- [15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and Tell: A Neural Image Caption Generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015.
- [16] J. Wu and R. J. Mooney. Faithful Multimodal Explanation for Visual Question Answering. *arXiv preprint arXiv:1809.02805*, 2018.
- [17] J. Wu, Z. Hu, and R. J. Mooney. Generating Question Relevant Captions to Aid Visual Question Answering. *arXiv preprint arXiv:1906.00513*, 2019.
- [18] 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 *ICML*, pages 2048–2057, 2015.
- [19] X. Yang, K. Tang, H. Zhang, and J. Cai. Auto-encoding graphical inductive bias for descriptive image captioning. *arXiv preprint arXiv:1812.02378*, 2018.

- [20] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4894–4902, 2017.
- [21] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In *ECCV*, pages 684–699, 2018.