Bitdefender

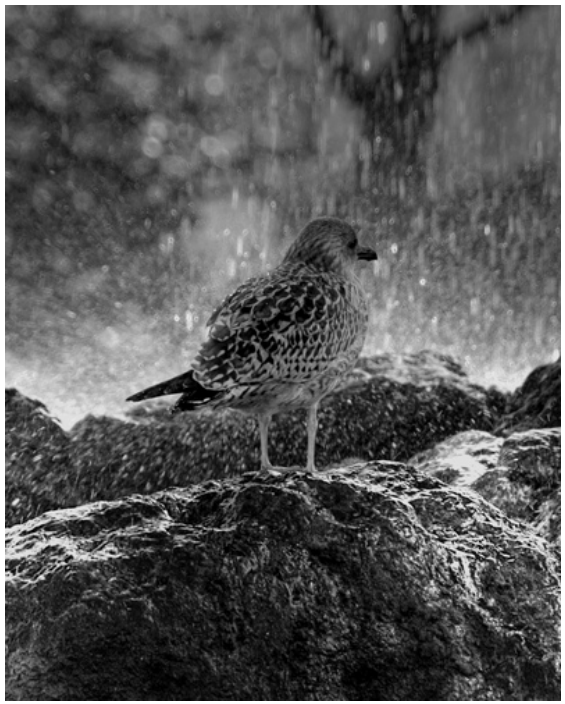
# Describing images and videos in natural language

Andrei Nicolicioiu
anicolicioiu@bitdefender.com

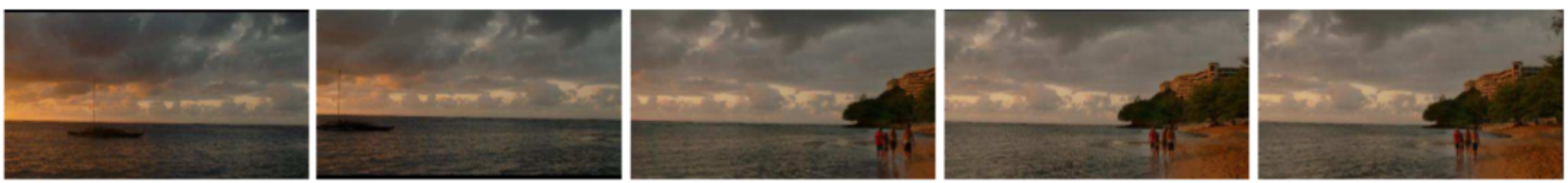
September 2, 2019

# Intro - Captioning

Bitdefender



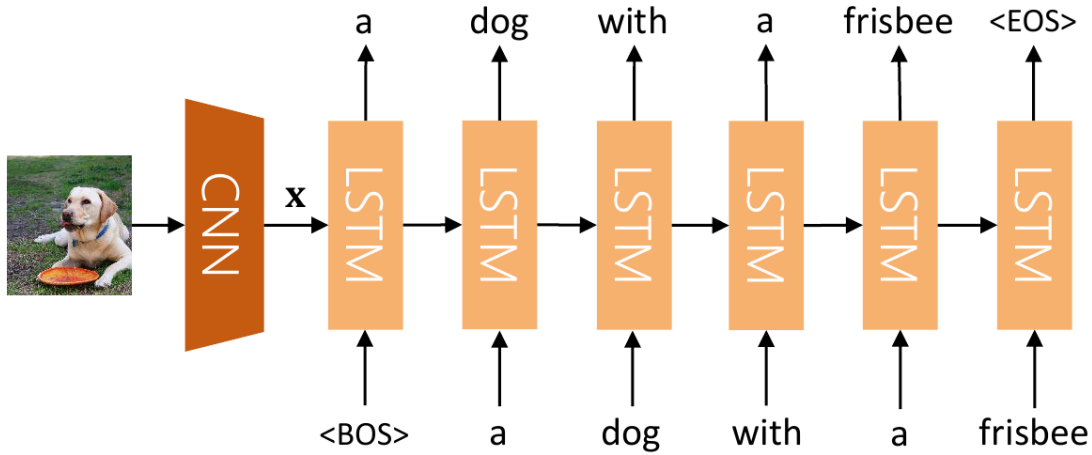
**Ours :** A bird is standing on a rock in the water.

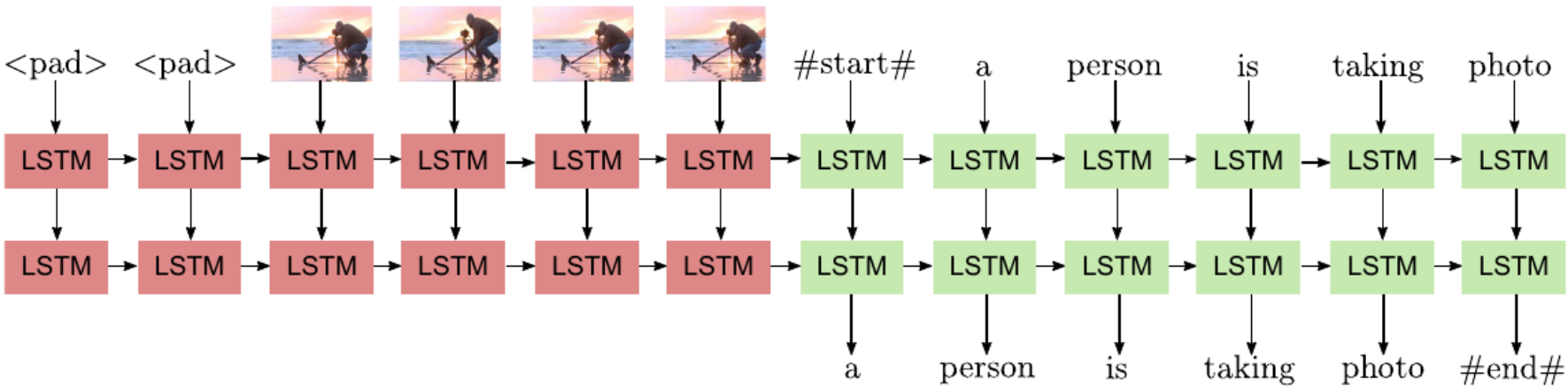


**GT** : The sun sets behind the watery horizon as the foursome continues along the shore toward a distant resort.

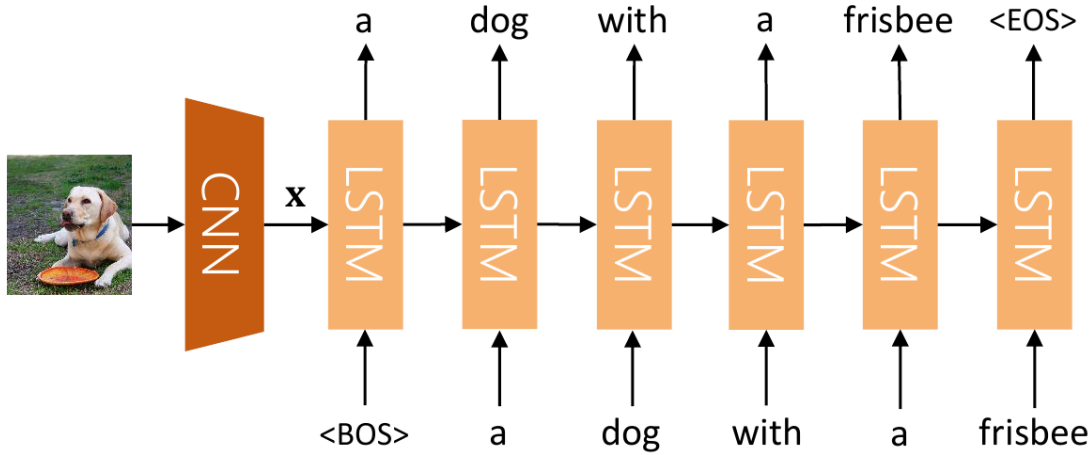
**Ours** : The sun shines as the sun sets to the horizon.

# Baseline Architectures



# Baseline Architectures



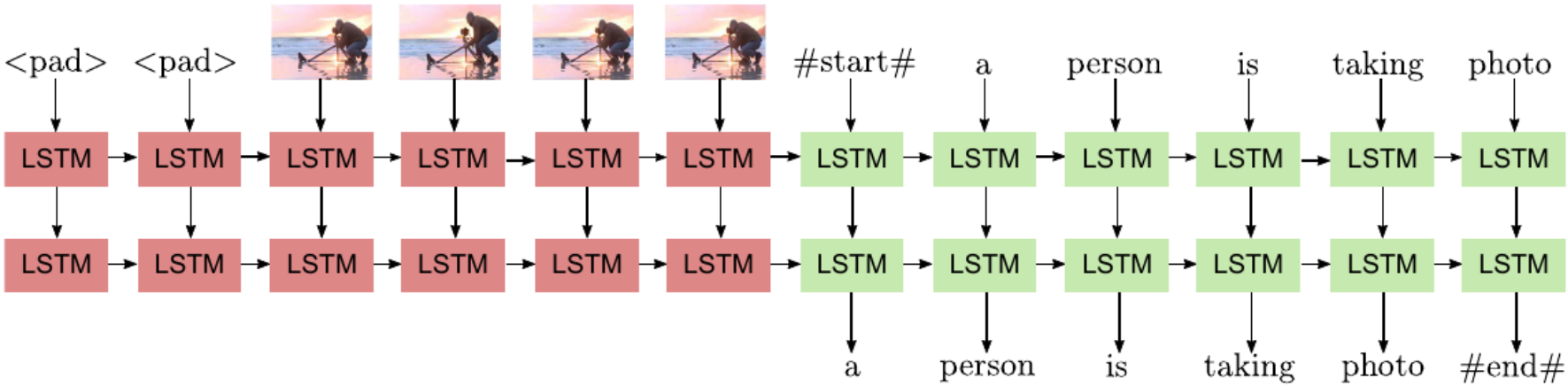

$$h_0 = CNN(Img)$$

$$w_0 =< BeginSentenceToken >$$

$$h_t = RNN(w_{t-1}, h_{t-1})$$

$$p(w_t) = f(h_t)$$

# Baseline Architectures




$$h_t^{enc} = RNN(CNN(Img_{t-1}), h_{t-1}^{enc})$$

$$h_0 = h_{last}^{enc}$$

$$w_0 = < BeginSentenceToken >$$

$$h_t = RNN(w_{t-1}, h_{t-1})$$

$$p(w_t) = f(h_t)$$

# Baseline Architectures - Problems

Bitdefender

$$h_t = RNN(w_{t-1}, h_{t-1})$$

$$p(w_t) = f(h_t)$$

# Baseline Architectures - Problems

Bitdefender

$$h_t = RNN(w_{t-1}, h_{t-1})$$

$$p(w_t) = f(h_t)$$

$$p(w_1, w_2, ..., w_m|visual) = \prod_{t=1}^{m} p(w_t|h_t)$$

# Baseline Architectures - Problems

Bitdefender

$$h_t = RNN(w_{t-1}, h_{t-1})$$

$$p(w_t) = f(h_t)$$

$$p(w_1, w_2, ..., w_m | visual) = \prod_{t=1}^{m} p(w_t | h_t)$$

- Big assumption: $p(w_t)$ are independent given $h_t$

# Baseline Architectures - Problems

Bitdefender

$$h_t = RNN(w_{t-1}, h_{t-1})$$

$$p(w_t) = f(h_t)$$

$$p(w_1, w_2, ..., w_m | visual) = \prod_{t=1}^{m} p(w_t | h_t)$$

- Big assumption: $p(w_t)$ are independent given $h_t$
  - **Q:** does $h_t$ contain enough information?

# Baseline Architectures - Problems

Bitdefender

$$h_t = RNN(w_{t-1}, h_{t-1})$$

$$p(w_t) = f(h_t)$$

$$p(w_1, w_2, ..., w_m | visual) = \prod_{t=1}^{m} p(w_t | h_t)$$

- Big assumption: $p(w_t)$ are independent given $h_t$
  - **Q:** does $h_t$ contain enough information?
- Learning by maximization at every step of cross-entropy between
  - distribution of words in the predicted sentence, $p(w_t)$
  - distribution of words in the ground truth sentence, $p(\hat{w}_t)$

# Baseline Architectures - Problems

Bitdefender

$$h_t = RNN(w_{t-1}, h_{t-1})$$

$$p(w_t) = f(h_t)$$

$$p(w_1, w_2, ..., w_m | visual) = \prod_{t=1}^{m} p(w_t | h_t)$$

- Big assumption: $p(w_t)$ are independent given $h_t$
  - **Q:** does $h_t$ contain enough information?
- Learning by maximization at every step of cross-entropy between
  - distribution of words in the predicted sentence, $p(w_t)$
  - distribution of words in the ground truth sentence, $p(\hat{w}_t)$
  - **Q:** is this the best way of computing a loss?

# Multi-Task Video Captioning with Video and Entailn Generation

Bitdefender

Pasunuru, Ramakanth, and Mohit Bansal. "Multi-Task Video Captioning with Video and Entailment Generation." In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, July 2017

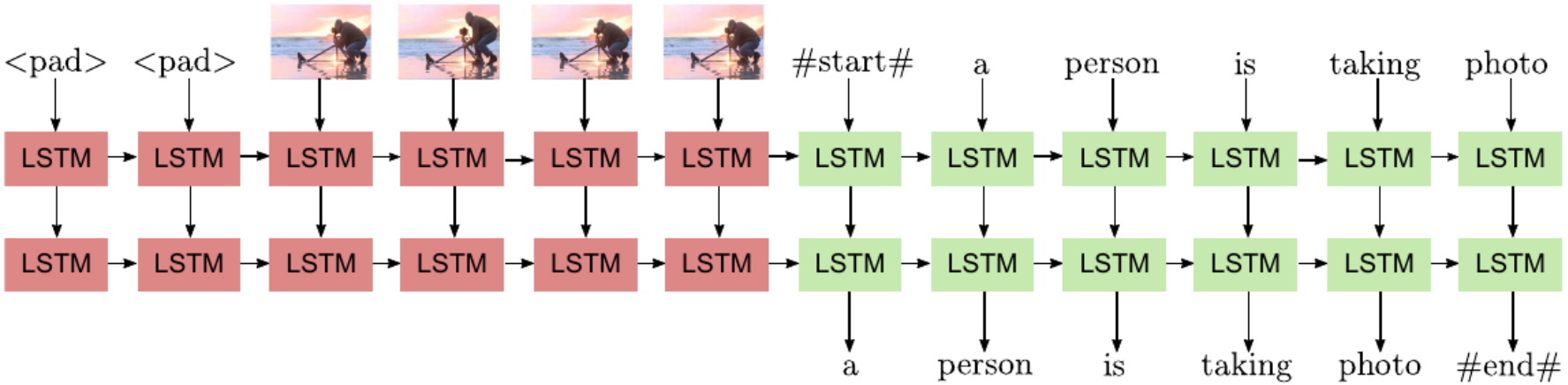
# Multi-Task Video Captioning with Video and Entailn Generation

Bitdefender

Pasunuru and Bansal [2017]

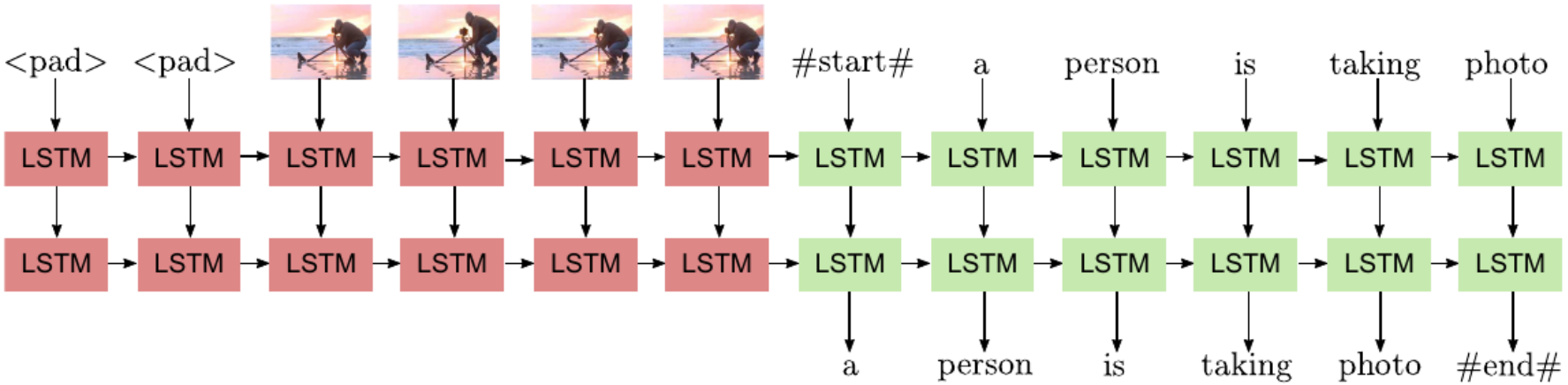
- improve captioning by sharing knowledge with other tasks
- use 2 other generative tasks
- a temporally-directed unsupervised video prediction
  - predict the next frames of a video
- logically-directed language entailment generation task
  - given a sentence, generate a sentence that can be seen as a consequence of the first

# Baseline: Seq2seq




$$p(w_1, w_2, ..., w_m | f_1, f_2, ..., f_n) = \prod_{t=1}^{m} p(w_t | h_t) \quad (1)$$

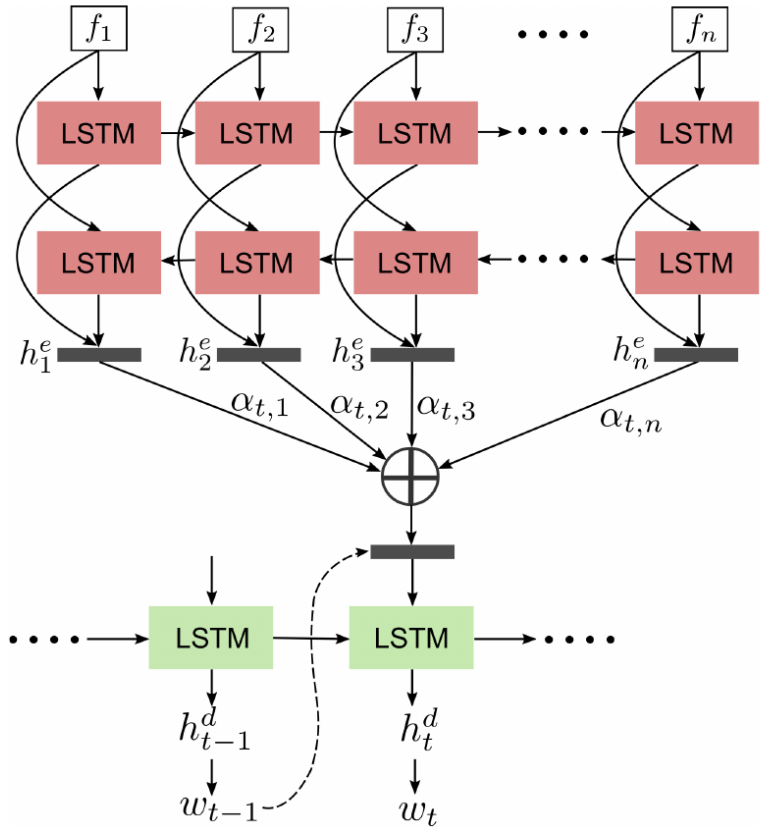
# Baseline: Seq2seq




$$p(w_1, w_2, ..., w_m | f_1, f_2, ..., f_n) = \prod_{t=1}^{m} p(w_t | h_t) \quad (1)$$

- **Q:** Is $h_t$ enough to represent all frames?

# Baseline: Seq2seq + Attention




$$h_t = f(h_{t-1}, w_{t-1}, c_t)$$

$$c_t = \sum_{i=i}^{N} att_{t,i} h_i^{enc}$$

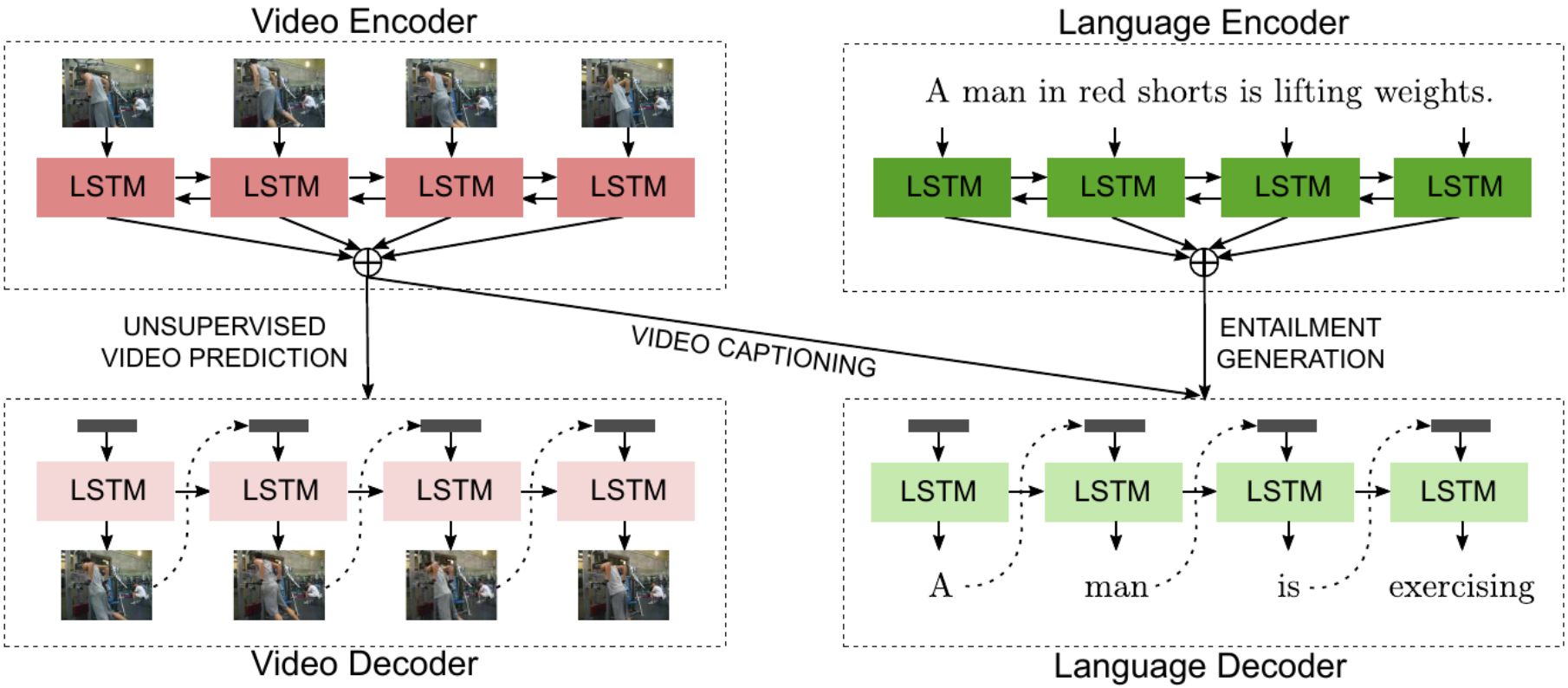
$$att_{t,i} = f(h_{t-1}, h_i^{enc})$$

# Use external language knowledge

Bitdefender

- Venugopalan et al. [2016] uses external language model **W** to improve video captioning model **C**
- learns language model with a recurrent model trained on Wikipedia
- improve captioning network
  - finetuning **C** from **W**
  - multiply the predictions of both models
  - use **W** hidden representation in the final prediction

# Overall architecture



# Entailment Generation

Bitdefender

Premise:

- A soccer game with multiple males playing.
- A dog jumping for a Frisbee in the snow

Entailment:

- Some men are playing a sport.
- An animal is outside in the cold weather, playing with a plastic toy

- given a sentence $\{w_1, w_2, .., w_m\}$ predict an entailment [1] $\{w_1^p, .., w_n^p\}$
- generate a sentence with a similar encoding-decoding architecture
  - encode the premise
  - decode the entailment

[1] `https://nlp.stanford.edu/projects/snli/`

# Unsupervised Video Prediction

Bitdefender

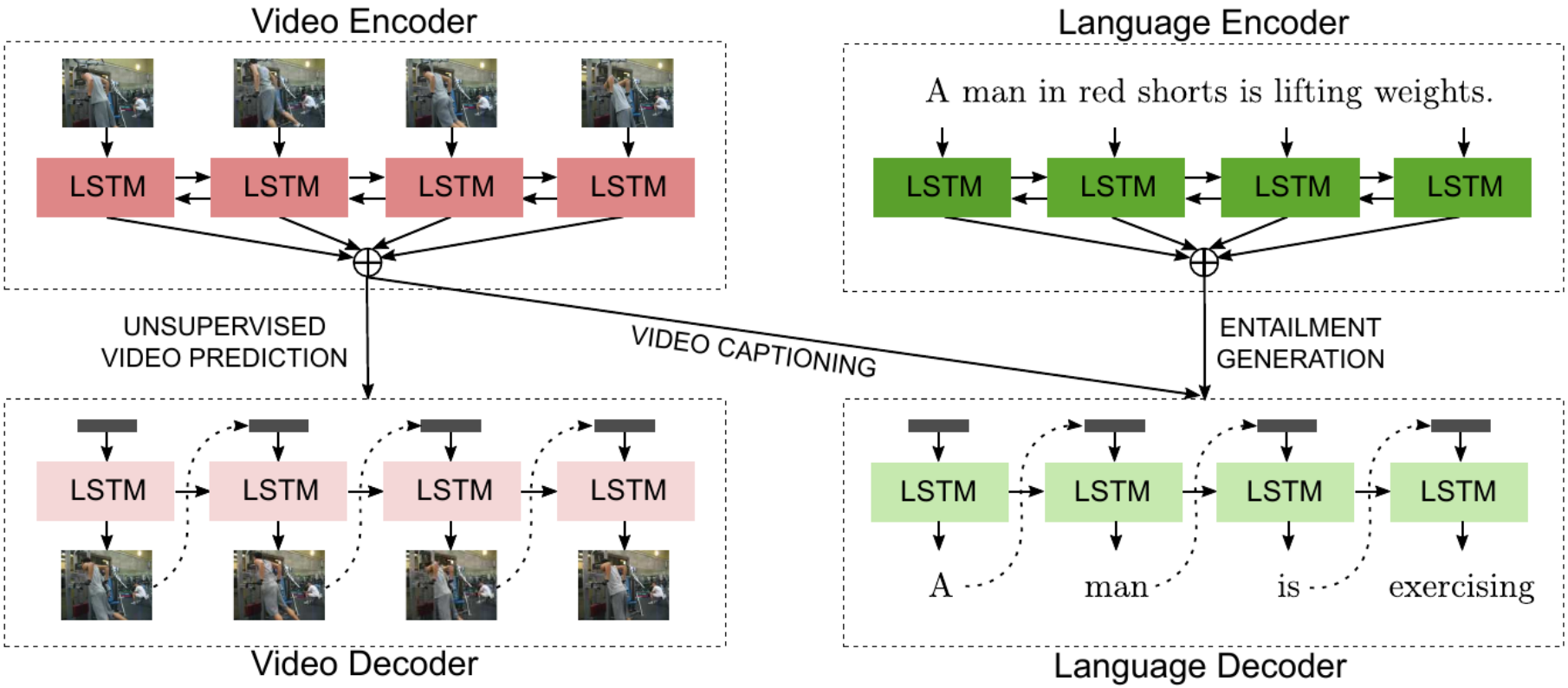
- given frame-level features $\{f_1, f_2, .., f_k\}$ predict the rest $\{f_{k+1}, .., f_n\}$
- predict future video frames with a similar encoding-decoding architecture
  - encode the features of the first k frames
  - decode the features of the remaining frames
- minimize the L2 norm between the predicted features and the ground truth

# Multi-Task Learning

Bitdefender

- improve captioning by sharing information with other tasks
  - share network parameters between tasks
- captioning shares video **encoder** video prediction task
- captioning shares video **decoder** entailment generation task
- attention parameters keept separated for the best results
- learn by alternating optimizing each task

# Overall architecture



# Results - Quantitative

Bitdefender

| Models | METEOR | CIDEr-D | ROUGE-L | BLEU-4 |
|---|---|---|---|---|
| PREVIOUS WORK | | | | |
| LSTM-YT (V) (Venugopalan et al., 2015b) | 26.9 | - | - | 31.2 |
| S2VT (V + A) (Venugopalan et al., 2015a) | 29.8 | - | - | - |
| Temporal Attention (G + C) (Yao et al., 2015) | 29.6 | 51.7 | - | 41.9 |
| LSTM-E (V + C) (Pan et al., 2016b) | 31.0 | - | - | 45.3 |
| Glove + DeepFusion (V) (E) (Venugopalan et al., 2016) | 31.4 | - | - | 42.1 |
| p-RNN (V + C) (Yu et al., 2016) | 32.6 | 65.8 | - | 49.9 |
| HNRE + Attention (G + C) (Pan et al., 2016a) | 33.9 | - | - | 46.7 |
| OUR BASELINES | | | | |
| Baseline (V) | 31.4 | 63.9 | 68.0 | 43.6 |
| Baseline (G) | 31.7 | 64.8 | 68.6 | 44.1 |
| Baseline (I) | 33.3 | 75.6 | 69.7 | 46.3 |
| Baseline + Attention (V) | 32.6 | 72.2 | 69.0 | 47.5 |
| Baseline + Attention (G) | 33.0 | 69.4 | 68.3 | 44.9 |
| Baseline + Attention (I) | 33.8 | 77.2 | 70.3 | 49.9 |
| Baseline + Attention (I) (E) $\otimes$ | 35.0 | 84.4 | 71.5 | 52.6 |
| OUR MULTI-TASK LEARNING MODELS | | | | |
| $\otimes$ + Video Prediction (1-to-M) | 35.6 | 88.1 | 72.9 | 54.1 |
| $\otimes$ + Entailment Generation (M-to-1) | 35.9 | 88.0 | 72.7 | 54.4 |
| $\otimes$ + Video Prediction + Entailment Generation (M-to-M) | **36.0** | **92.4** | **72.8** | **54.5** |

Results on YouTube2Text dataset

# Results - Qualitative



**Ground truth:** Two women are shopping in a store.
Two girls are shopping.

**Baseline model:** A man is doing a monkey in a store.

**Multi-task model:** A woman is shopping in a store.



**Ground truth:** Two men are fighting.
A group of boys are fighting.

**Baseline model:** A group of men are dancing.

**Multi-task model:** Two men are fighting.

(a)

# Results - Qualitative

**Ground truth:** A woman slices a shrimp tail.
A girl is cutting a fish tale.
**Baseline model:** A person is cutting the something.
**Multi-task model:** A woman is cutting a piece of meat.

**Ground truth:** Two men are talking aggressively.
The boy is talking.
**Baseline model:** A man is crying.
**Multi-task model:** A man is talking.

(b)

# Improved Image Captioning via Policy Gradient optimization of SPIDEr

Bitdefender

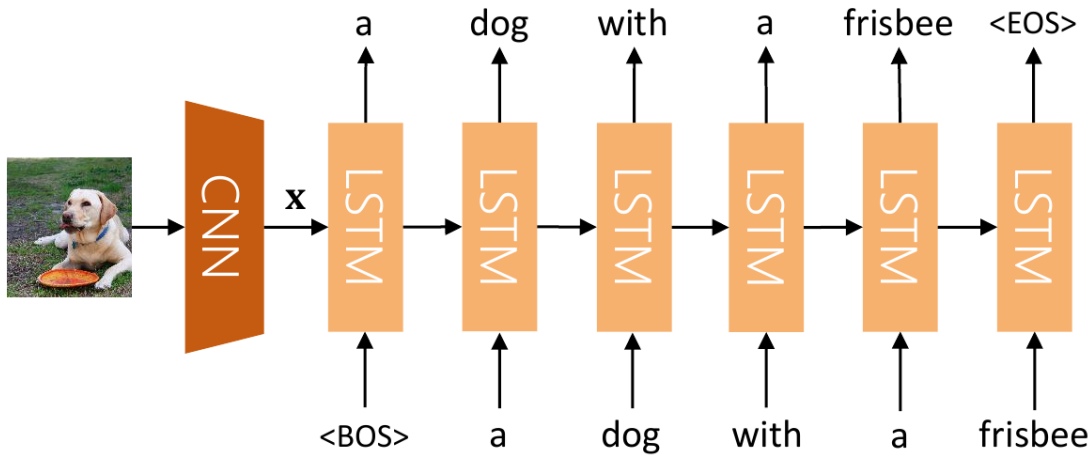
S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Improvedimage captioning via policy gradient optimization of spider. InThe IEEE International Conference on Computer Vision (ICCV),Oct 2017. Liu et al. [2017]

# Captioning

- at every step maximize cross-entropy between distribution of words in the predicted and in the ground truth sentences

$$J(\theta) = \sum_{n=1}^{N} \sum_{t=1}^{T_n} \text{Loss}(\pi_\theta(\hat{y}_t^n | \mathbf{x}^n, \hat{\mathbf{y}}_{t-1}^n)) \,,$$

$$\text{Loss}(\pi_\theta(\hat{y}_t^n | \mathbf{x}^n, \hat{\mathbf{y}}_{t-1}^n)) = -\log \pi_\theta(\hat{y}_t^n | \mathbf{x}^n, \hat{\mathbf{y}}_{t-1}^n)$$



# Problem of Maximum Likelihood

Bitdefender

- **exposure bias** Ranzato et al. [2015]:
  - at training time the previous ground truth word is used
  - at test time the previous generated word is used
  - **problem**: training/ testing use different distributions
  - **problem**: errors are accumulated with time
- word level supervision:
  - the loss is computed just based the current word

# Solutions?

Bitdefender

- generate the whole sentence just using the model own predictions
- compute a score for the sentence, **not** just for every word
  - optimize for sentence level metrics like: BLEU / METEOR/ CIDEr
  - problem: these scores are **not differentiable**
- use reinforcement learning method with **policy gradient**

# Policy gradient

Bitdefender

- cast the captioning task as a reinforcement learning task
- the words are predicted according to a policy: $\pi_\theta(y_t|\mathbf{x}, \mathbf{y}_{t-1})$
- actions at time t $<->$ word prediction at time t
- state at time t $<->$ image features and first t-1 words
- estimate $Q(s_t, a_t)$
  - expected future reward starting from state $s_t$ and choosing action $a_t$
  - expectation is over the future words

# Policy gradient - Training

Bitdefender

- we want to change the policy in order to predict better actions(words)
- better actions - better rewards
- maximize expected future reward

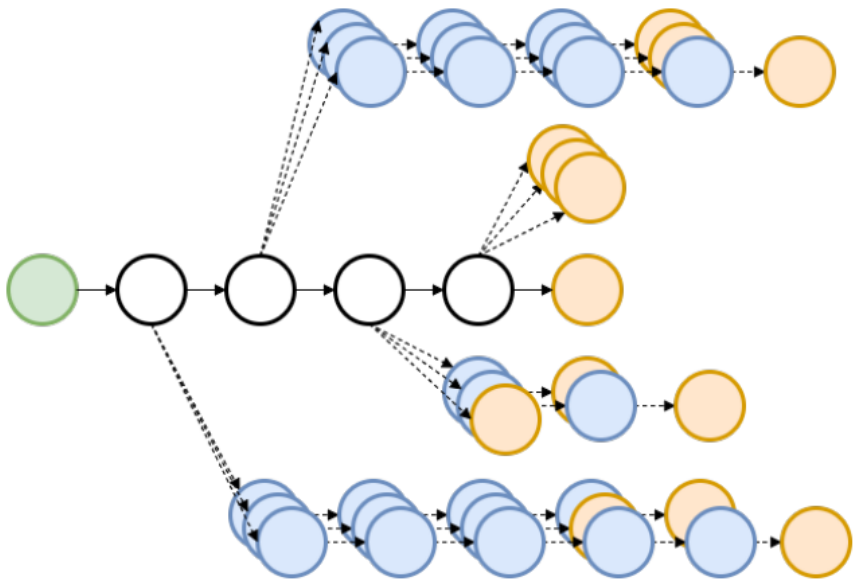
$$J(\theta) = \sum_t \mathbb{E}_{\mathbf{y_t}} \pi_\theta(y_t^m | \mathbf{x}, \mathbf{y}_{t-1}^m) Q(s_t^m, y_{t-1}^m) \tag{2}$$

- using policy gradient is computed by Sutton et al. [2000]:

$$\nabla_\theta J(\theta) = \frac{1}{M} \sum_m \sum_t \nabla_\theta \pi_\theta(y_t^m | \mathbf{x}, \mathbf{y}_{t-1}^m) Q(s_t^m, y_{t-1}^m) \tag{3}$$

# Estimating Q

- at every step we must estimate the expected reward $Q(s_t^m, y_t^m)$
- use Monte Carlo rollout to generate full sentences
- compute any kind of score:
  - BLEU, METEOR, CIDEr, SPICE



# Training Algorithm

Bitdefender

1. pretrain $\pi_\theta$ using Maxim Likelihood Estimation
2. repeat until $\pi_\theta$ converge
3. for each (image-sentence) pair
4. generate sentence according to $\pi$
5. for each word in sentence
6. estimate $Q$ using Monte Carlo rollouts
7. compute the gradient of the loss w.r.t $\theta$
8. update the policy $\pi_\theta$ using SGD

# Choosing Language Scores

Bitdefender

| Metric | Proposed to evaluate | Underlying idea |
|---|---|---|
| BLEU (Papineni et al., 2002) | Machine translation | $n$-gram precision |
| ROUGE (Lin, 2004) | Document summarization | $n$-gram recall |
| METEOR (Banerjee and Lavie, 2005) | Machine translation | $n$-gram with synonym matching |
| CIDEr (Vedantam et al., 2015) | Image description generation | *tf-idf* weighted $n$-gram similarity |
| SPICE (Anderson et al., 2016) | Image description generation | Scene-graph synonym matching |

2

- BLEU, METEOR, ROUGE, CIDEr (BMRC)
  - don't correlate with human evaluation of quality
  - human captions actually score lower on these metrics

[2] From Kilickaya et al. [2016]

# Choosing Language Scores

Bitdefender

- SPICE Anderson et al. [2016] strong correlate with human ratings
  - parses each of the reference sentences
  - derive an abstract scene graph representation
  - generated sentence is parsed, and compared to the graph
- SPICE ignores syntactic quality
  - generate ungrammatical sentences, with repeated phrases
- Solution: SPICE + CIDEr = SPIDEr


  - SPICE - captions are semantically faithful to the image
  - CIDEr - captions are syntactically fluent

# Results

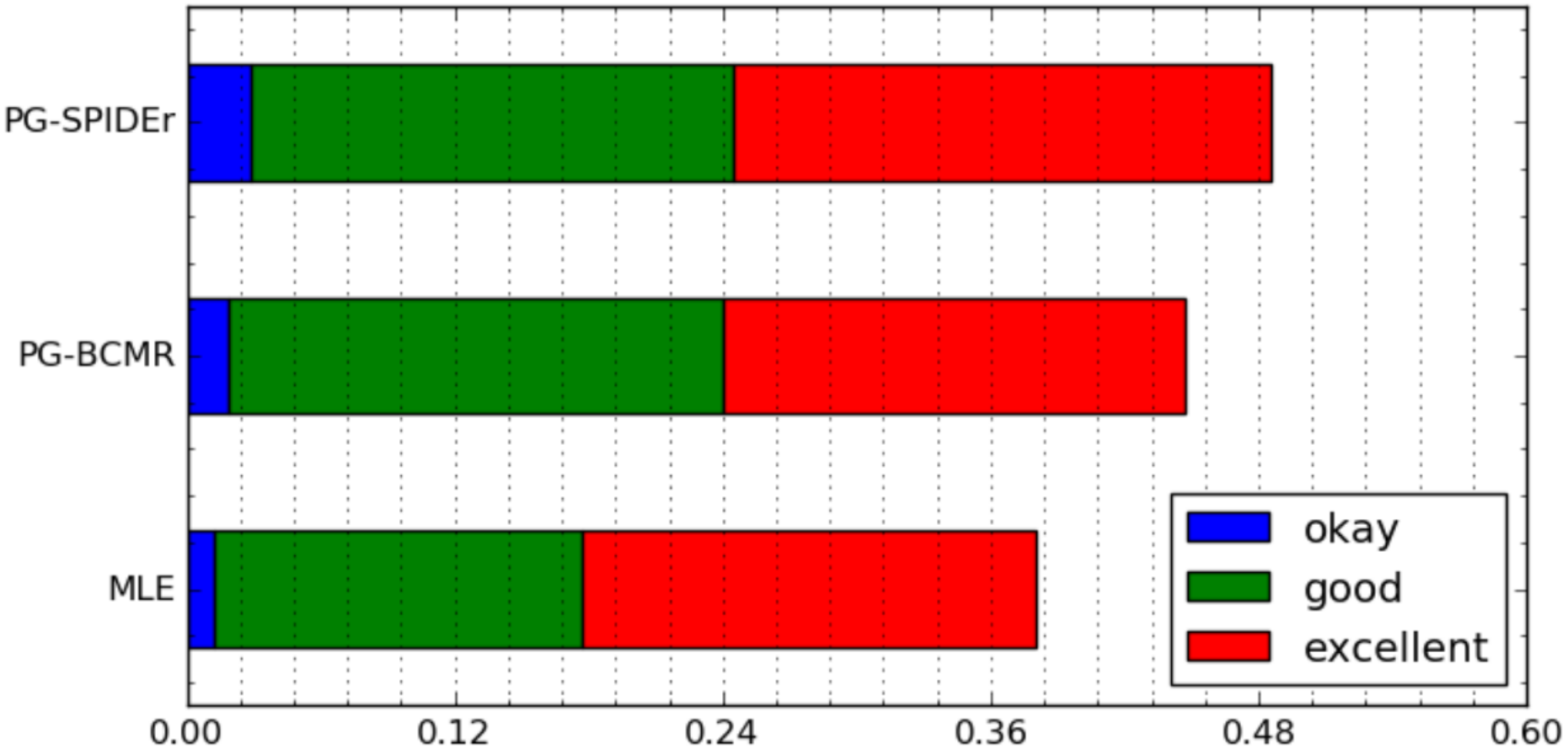
Bitdefender

| Submissions | CIDEr-D | Meteor | ROUGE-L | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
|---|---|---|---|---|---|---|---|
| MSM@MSRA [28] | 0.984 | 0.256 | 0.542 | 0.739 | 0.575 | 0.436 | 0.330 |
| Review Net [27] | 0.965 | 0.256 | 0.533 | 0.720 | 0.550 | 0.414 | 0.313 |
| ATT [29] | 0.943 | 0.250 | 0.535 | 0.731 | 0.565 | 0.424 | 0.316 |
| Google [22] | 0.943 | 0.254 | 0.530 | 0.713 | 0.542 | 0.407 | 0.309 |
| Berkeley LRCN [7] | 0.921 | 0.247 | 0.528 | 0.718 | 0.548 | 0.409 | 0.306 |
| MLE | 0.947 | 0.251 | 0.531 | 0.724 | 0.552 | 0.405 | 0.294 |
| PG-BLEU-4 | 0.966 | 0.249 | 0.550 | 0.737 | 0.587 | **0.455** | **0.346** |
| PG-CIDEr | 0.995 | 0.249 | 0.548 | 0.737 | 0.581 | 0.442 | 0.333 |
| MIXER-BCMR | 0.924 | 0.245 | 0.532 | 0.729 | 0.559 | 0.415 | 0.306 |
| MIXER-BCMR-A | 0.991 | **0.258** | 0.545 | 0.747 | 0.579 | 0.431 | 0.317 |
| PG-BCMR | **1.013** | **0.257** | **0.55** | **0.754** | **0.591** | 0.445 | 0.332 |
| PG-SPIDEr | 1.000 | 0.251 | 0.544 | 0.743 | 0.578 | 0.433 | 0.322 |

# Results - Human perception

Bitdefender

Human annotated captions with labels: "bad", "okay", "good" or "excellent"



# Results

Captions:

1. MLE: a woman walking down a street while holding an umbrella .
2. PG-SPICE: group of people walking down a street with a man on a street holding a traffic light and a traffic light on a city street with a city street
3. PG-BCMR: a group of people walking down a city street.
4. PPG-SPIDEr: a group of people walking down a street with a traffic light .

# Results

Captions:

1. MLE: a man sitting in front of a laptop computer .
2. PG-SPICE: a man sitting in front of a book and a laptop on a table with a laptop computer on top of a table with a laptop computer on top of
3. PG-BCMR: a man sitting in front of a book .
4. PG-SPIDEr: a man sitting at a table with a book .

# Show, Adapt and Tell: Adversarial Training of Cross-domain Image Captioner

Bitdefender

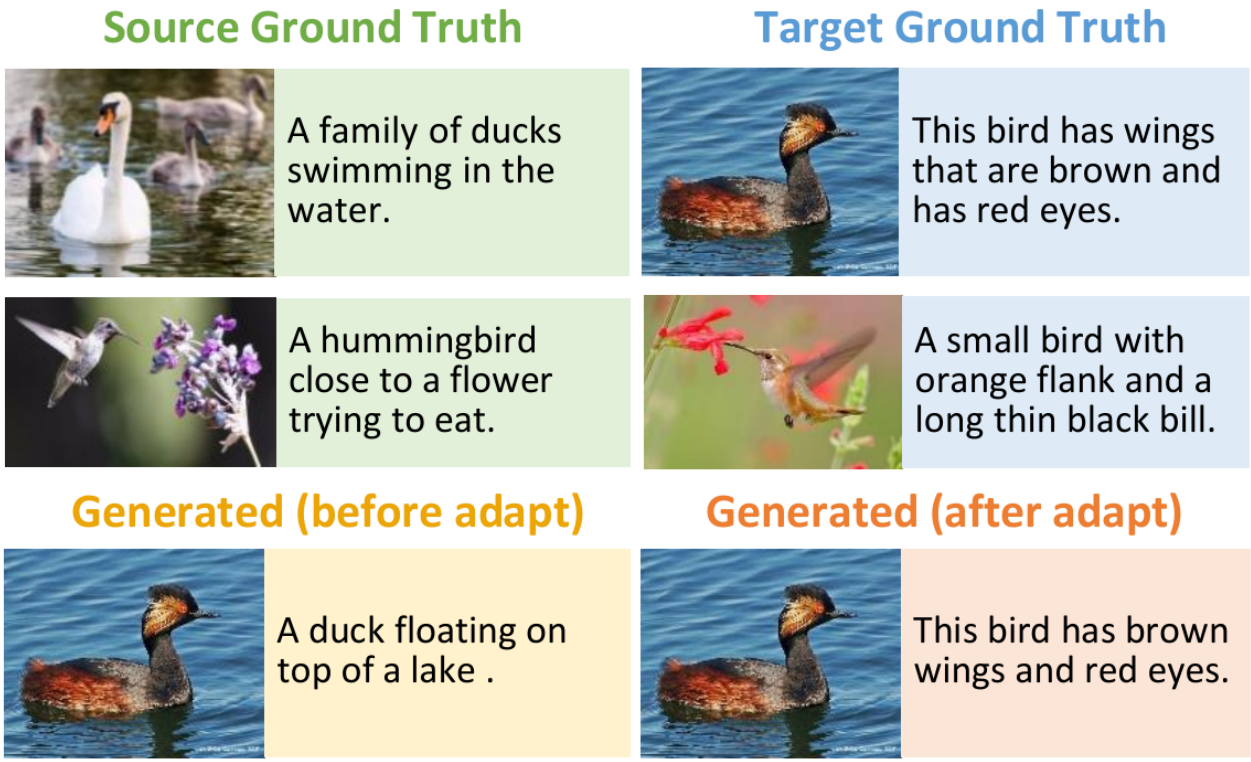
MLA Chen, Tseng-Hung, et al. "Show, adapt and tell": Adversarial training of cross-domain image captioner. InThe IEEE International Conference on Computer Vision (ICCV), Oct 2017

Chen et al. [2017]

# Show, Adapt and Tell: Adversarial Training of Cross-domain Image Captioner

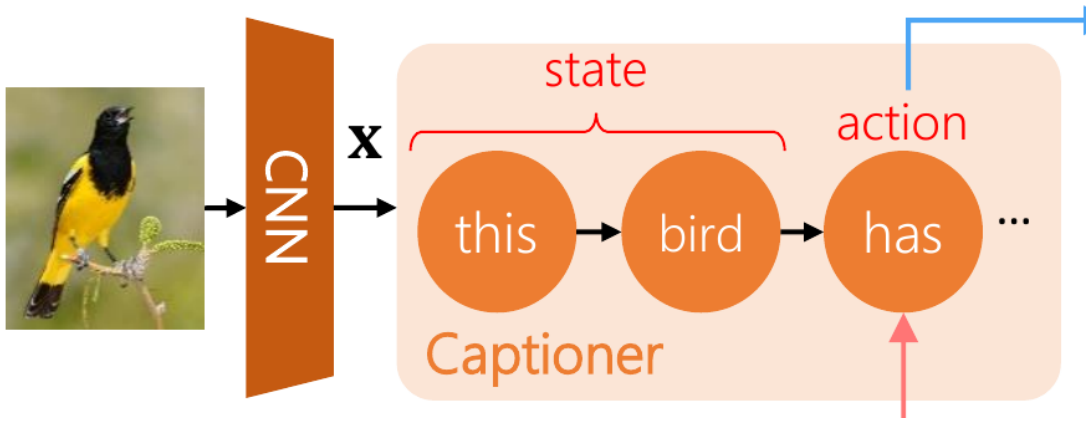
Bitdefender

- method for image captioning
- adapting to a new dataset that does **not** have image-descriptions pairs



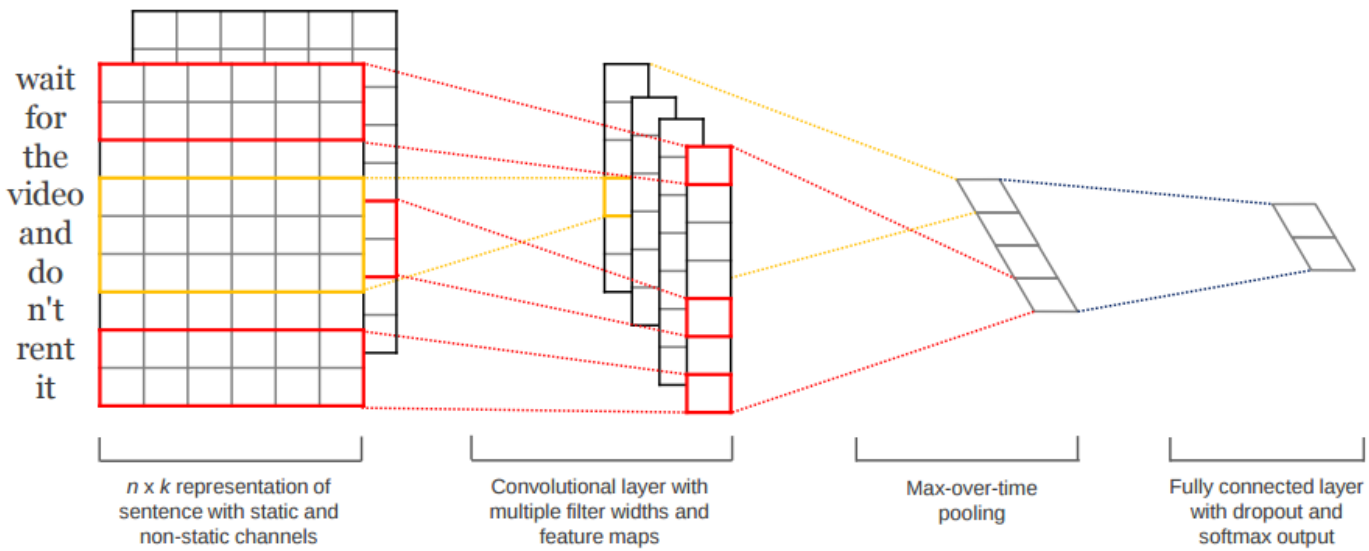
# Cross Domain Learning

- problem lack of ground truth (image, sentence) pairs
- cannot use maximum likelihood $->$ use **policy gradient**
  - generate complete sentences then evaluate them
- cannot calculate a final loss based on ground truth
  - Solution: use **critics**
  - Critic 1: similarity between 2 sentences
  - Critic 2: relevancy between an image and a sentence



# Domain critic

- reward sentence that resembles the sentence drawn from the target domain
- classify sentences as "source" domain, "target" domain, or "generated"
  - use convolutional network for classification Kim [2014]

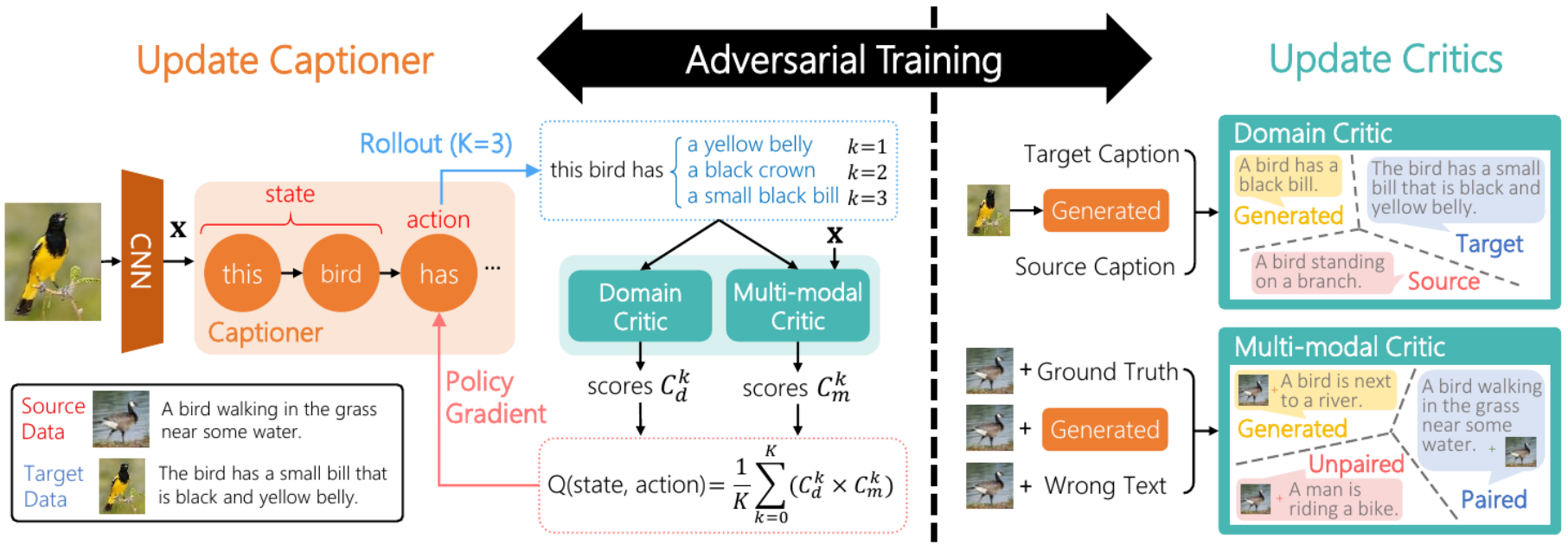


# Multi-modal critic

Bitdefender

- reward sentence relevant to the input image
- classify image-sentence pair as "paired:, "unpaired", or "generated" data
- use LSTM to encode the sentence
- final class assigned based on image and text encoding

# Overall Architecture



# Adversarial Training

Bitdefender

- given image x and generated caption y
- the policy $\pi_\theta$ is optimized to have high reward
  - critics should predict high values for CriticDomain($target|y$) and CriticModal($paired|x, y$)
- critics are optimized for
  - high values for CriticDomain($generated|y$) and CriticModal($generated|x, y$)
  - low values for CriticDomain($target|y$) and CriticModal($paired|X, Y$)

# Training

Bitdefender

- Pretrain $\pi_\theta$ on the source domain
- repeat until $\pi_\theta$ converge
  - repeat for $N_c$ steps
    - generate sentence from **target** domain image according to $\pi$
    - compute the gradients and update the domain critic
    - generate sentence from **source** domain image according to $\pi$
    - compute the gradients and update the multi-model critic
  - repeat for $N_g$ steps
    - generate sentence from **target** domain image according to $\pi$
    - for each step compute the reward according to critics with monte carlo rollouts
    - compute the gradient and update the policy $\pi_\theta$

# Results

Bitdefender

| Method | Target (test) | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | Meteor | ROUGE | CIDEr | SPICE |
|---|---|---|---|---|---|---|---|---|---|
| Source Pre-trained | CUB-200 | 50.8 | 28.3 | 13.9 | 6.1 | 12.9 | 33 | 3 | 4.6 |
| DCC | CUB-200 | 68.6 | 47.3 | 31.4 | 21.4 | 23.8 | 46.4 | 11.9 | 11.1 |
| Ours | CUB-200 | **91.4** | **73.1** | **51.9** | **32.8** | **27.6** | **58.6** | **24.8** | **13.2** |
| Fine-tuning | CUB-200 | 91.3 | 80.2 | 69.2 | 59 | 36.1 | 69.7 | 61.1 | 17.9 |
| Source Pre-trained | Oxford-102 | 48.3 | 21.6 | 6.2 | 1.3 | 10.5 | 25.8 | 3.1 | 4.4 |
| DCC | Oxford-102 | 51 | 33.8 | 24.1 | 16.7 | 21.5 | 38.3 | 6 | 9.8 |
| Ours | Oxford-102 | **85.6** | **76.9** | **67.4** | **60.5** | **36.4** | **72.1** | **29.3** | **17.9** |
| Fine-tuning | Oxford-102 | 87.5 | 80.1 | 72.8 | 66.3 | 40 | 75.6 | 36.3 | 18.5 |
| Source Pre-trained | TGIF | 41.6 | 23.3 | 12.6 | 7 | 12.7 | 32.7 | 14.7 | 8.5 |
| DCC | TGIF | 34.6 | 17.5 | 9.3 | 4.1 | 11.8 | 29.5 | 7.1 | 7.3 |
| Ours | TGIF | **47.5** | **29.2** | **17.9** | **10.3** | **14.5** | **37** | **22.2** | **10.6** |
| Fine-tuning | TGIF | 51.1 | 32.2 | 20.2 | 11.8 | 16.2 | 39.2 | 29.8 | 12.1 |
| Source Pre-trained | Flickr30k | 57.3 | 36.2 | 21.9 | 13.3 | 15.1 | 38.8 | 25.3 | 8.6 |
| DCC | Flickr30k | 54.3 | 34.6 | 21.8 | 13.8 | 16.1 | 38.8 | 27.7 | 9.7 |
| Ours | Flickr30k | **62.1** | **41.7** | **27.6** | **17.9** | **16.7** | **42.1** | **32.6** | **9.9** |
| Fine-tuning | Flickr30k | 59.8 | 41 | 27.5 | 18.3 | 18 | 42.9 | 35.9 | 11.5 |

# Results

Bitdefender

CUB-200

Before: A bird is standing on a table with flowers.

After: A small bird with a white belly and a black head.

Before: A red bird sitting on a tree branch.

After: This is a red bird with a black wing and a small beak.

Oxford-102

Before: A white flower in a vase on a table.

After: This flower has petals that are pink and has a yellow center.

Before: A yellow flower is in a clear vase.

After: This flower has petals that are yellow and has red lines.

# Results

Bitdefender

TGIF

Before: A cat is standing in a room with a cat.

After: A cat is playing with a toy in a room.


Before: A baseball player is a ball on a field.

After: A group of men are playing soccer on a field.

Flickr30k

Before: A young baseball player is a ball in the field.

After: A young baseball player is sliding into a base.

Before: A boy in a field playing with a frisbee.

After: A young boy playing with a soccer ball in a field.

# Questions?

Bitdefender

Thank you!

# References I


P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, and M. Sun. Show, adapt and tell: Adversarial training of cross-domain image captioner. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

M. Kilickaya, A. Erdem, N. Ikizler-Cinbis, and E. Erdem. Re-evaluating automatic metrics for image captioning. *arXiv preprint arXiv:1612.07600*, 2016.

Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

# References II


S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Improved image captioning via policy gradient optimization of spider. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

R. Pasunuru and M. Bansal. Multi-task video captioning with video and entailment generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, July 2017.

M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

# References III

S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko. Improving lstm-based video description with linguistic knowledge mined from text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1961–1966. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1204. URL http://www.aclweb.org/anthology/D16-1204.