HAAV: Hierarchical Aggregation of Augmented Views for Image Captioning

Chia-Wen Kuo
Georgia Tech
albert.cwkuo@gatech.edu

Zsolt Kira
Georgia Tech
zkira@gatech.edu

Appendix

A. COCO test results

We also evaluate our HAAV using an ensemble of four models same as other methods on the online MS-COCO test server, and report the results in Table 1. We can see that our HAAV outperforms previous methods by a large margin across all metrics.

B. Overfitting

Training a data-hungry transformer model on a medium-scale dataset of MS-COCO (around 0.6M training samples) is prone to overfitting. In HAAV, we propose to regard heterogeneous views as augmentations of the input image and encode the views independently with a shared encoder. We claim that this formulation increases data diversity and is more parameter and label-efficient. Furthermore, we add a contrastive loss to improve representation quality of encoded views, which is also beneficial for label efficiency. In Figure 1, we show the validation curve for concatenated views and HAAV. Compared to concatenated views, our HAAV indeed suffers less from overfitting. Due to overfitting, the CIDEr score of concatenated views drops by 3.6 from the highest to the end of training.

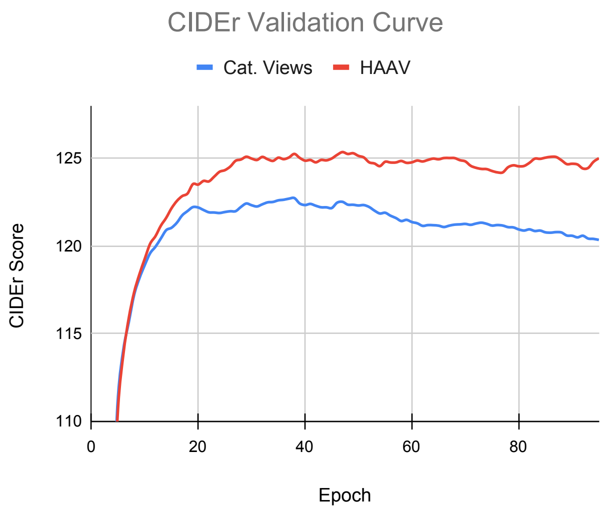


Figure 1. CIDEr validation curve of HAAV *v.s.* concatenated views.

C. Implementation Details

We provide a detailed list of hyperparameters including their values and whether they are tuned in Table 2 (cross-entropy training) and Table 3 (SCST training). For cross-entropy training, the model can be trained with a single Nvidia 2080 Ti GPU in 2 days. For SCST training, the model can be trained with a single Nvidia A40 GPUs in 4 days.

Table 1. MS-COCO test server results

	B-1		**B-2**		**B-3**		**B-4**		**M**		**R**		**C**	
Method	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [8]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7
Up-Down [1]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
AoANet [3]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
$\mathcal{M}^2$ [2]	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
X-LAN [7]	81.9	95.7	66.9	90.5	52.4	82.5	40.3	72.4	29.6	39.2	59.5	75.0	131.1	133.5
DLCT [6]	82.4	96.6	67.4	91.7	52.8	83.8	40.6	74.0	29.8	39.6	59.8	75.3	133.3	135.4
HAAV (ours)	**84.0**	**97.6**	69.1	93.3	54.3	85.8	**41.7**	**76.1**	**30.2**	**39.9**	**60.4**	**75.8**	**139.1**	**142.3**

Table 2. Hyperparameters for cross-entropy training. The values for untuned parameters are inherited from the base image captioning model, Xmodal-Ctx [4].

Hyperparameter	Value	Tuned	Note
N	3		Number of encoder layers
M	3		Number of decoder layers
lr	2e-5	✓	learning rate
bs	50		batch size
wd	0.05		weight decay
λ	0.05	✓	loss weight for $\mathcal{L}_{con}$
p_c	0.1		drop rate for channel-wise dropout
p_s	0.1		drop rate for sequence-wise dropout
p_v	0.1		drop rate for view-wise dropout
optimizer	AdamW		Adam with decoupled weight decay [5]
lr scheduler	constant with linear warmup		linearly warm up lr from 0.0, and then stay constant
warmup steps	10k		
K	8k	✓	size of memory buffer for MoCo contrastive learning
τ	0.06	✓	temperature scaling for MoCo contrastive learning
ema	0.999		exponential moving avergae for MoCo contrastive learning

Table 3. Hyperparameters for SCST [8] training. The values for untuned parameters are inherited from the base image captioning model, Xmodal-Ctx [4], or from the tuned value in cross-entropy training.

Hyperparameter	Value	Tuned	Note
N	3		Number of encoder layers
M	3		Number of decoder layers
lr	5e-6		learning rate
bs	40		batch size
wd	0.0		weight decay
λ	0.2	✓	loss weight for $\mathcal{L}_{con}$
p_c	0.1		drop rate for channel-wise dropout
p_s	0.1		drop rate for sequence-wise dropout
p_v	0.1		drop rate for view-wise dropout
optimizer	AdamW		Adam with decoupled weight decay [5]
lr scheduler	None		do not use any lr scheduler
K	8k	✓	size of memory buffer for MoCo contrastive learning
τ	0.06	✓	temperature scaling for MoCo contrastive learning
ema	0.999		exponential moving avergae for MoCo contrastive learning

D. Generated Captions

In Figure 2, we show some random examples of different captions generated by our HAAV and another trained-from-scratch SoTA method Xmodal-Ctx [4]. Qualitatively, HAAV is capable of generating captions in more details and more closely related to the input image rather than generating a generic sentence. For example, in Figure 2a, HAAV generates "*a man standing in a living room holding a nintendo wii game controller*", while Xmodal-Ctx generates a more generic description of "*a group of people sitting on a couch playing a video game*". Another example in Figure 2e shows that HAAV describes the train in more details as "*a yellow and purple train*" rather than just "*a train*" by Xmodal-Ctx.

E. Adaptive View Aggregation Weights

We show more examples of how the hierarchical decoder adaptively weighs the encoded views according to their effectiveness for caption generation at the view level and at the word level in Figure 3-8.

At the view level (figures on the left), we add noise to a view by randomly zeroing out tokens in a view to make a view *less effective*, and expect a drop of weights toward that noised view. To measure the weights, we take the multi-head attention weights of $\text{CrossAttn}_{\text{Lv2}}$ at the last decoder layer and average the attention weights across heads. Overall, the weights for the noised view drop consistently at each word prediction step compared to the same view *without* added noise. This means that our hierarchical decoder indeed learns to adaptively weigh the views according to their effectiveness at the view level.

At the word level (figures on the right), we randomly mask out a prominent region of the input image for a view, and expect a drop of the weights toward the masked view at the step of generating the word of that masked region. To measure the weights, we take the multi-head attention weights of $\text{CrossAttn}_{\text{Lv2}}$ at the last decoder layer and measure the attention weights of each head at the step of generating the word of that masked region. Overall, the weights for the masked view drops consistently across all attention heads compared to the same view without masking. This means that our hierarchical decoder indeed learns to adaptively weigh the input views according to their usefulness at the word level.

(a)
HAAV: a man standing in a living room holding a nintendo wii game controller
Xmodal-Ctx: a group of people sitting on a couch playing a video game

(b)
HAAV: a batter catcher and umpire during a baseball game
Xmodal-Ctx: a baseball player holding a bat on a field

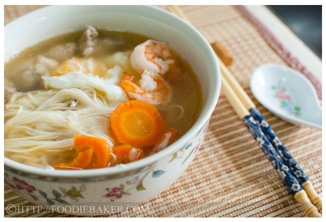
(c)
HAAV: a bunch of umbrellas hanging from a ceiling
Xmodal-Ctx: a bunch of flowers hanging from a ceiling

(d)
HAAV: two people playing a video game in a room
Xmodal-Ctx: a person standing in front of a tv

(e)
HAAV: a yellow and purple train parked at a train station
Xmodal-Ctx: a train that is sitting on the tracks

(f)
HAAV: a piece of cake on a plate with a flower
Xmodal-Ctx: a slice of cake on a plate with a fork

(g)
HAAV: a building with a clock tower in the middle of it
Xmodal-Ctx: a large building with a clock on the side of it



(h)
HAAV: a bowl of soup with vegetables and noodles
Xmodal-Ctx: a bowl of soup sitting on top of a table

Figure 2. Captions generated by HAAV and another trained-from-scratch SoTA method Xmodal-Ctx [4]

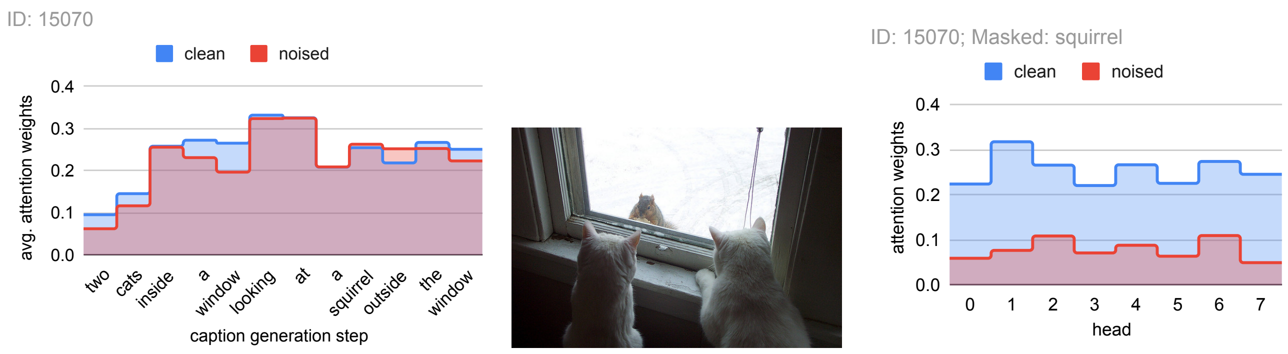


Figure 3. ***(left)*** The attention weights averaged across different attention heads for a noised view drop consistently at each caption generation step. ***(center)*** Input image with caption: "*two cats inside a window looking at a squirrel outside the window*". ***(right)*** The attention weights of different attention heads drop consistently at the step of generating the word "*squirrel*", which is masked out in the input image.

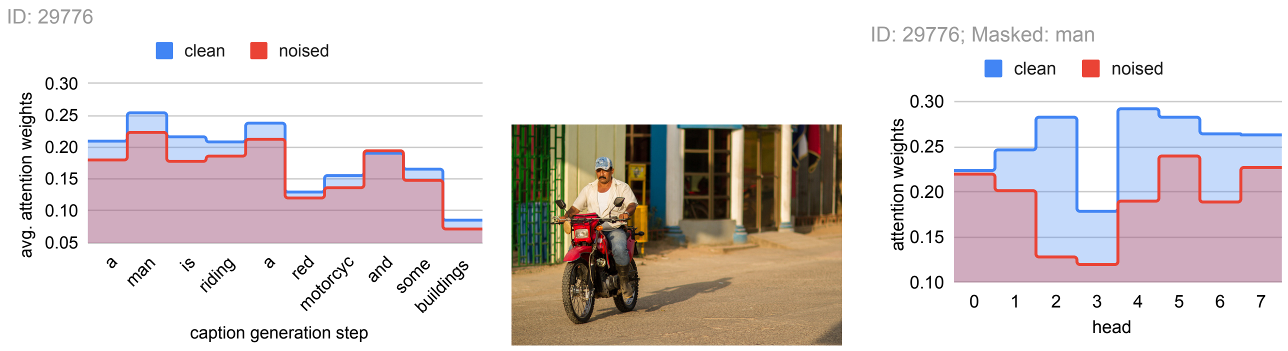


Figure 4. ***(left)*** The attention weights averaged across different attention heads for a noised view drop consistently at each caption generation step. ***(center)*** Input image with caption: "*a man is riding a red motorcycle and some buildings*". ***(right)*** The attention weights of different attention heads drop consistently at the step of generating the word "*man*", which is masked out in the input image.

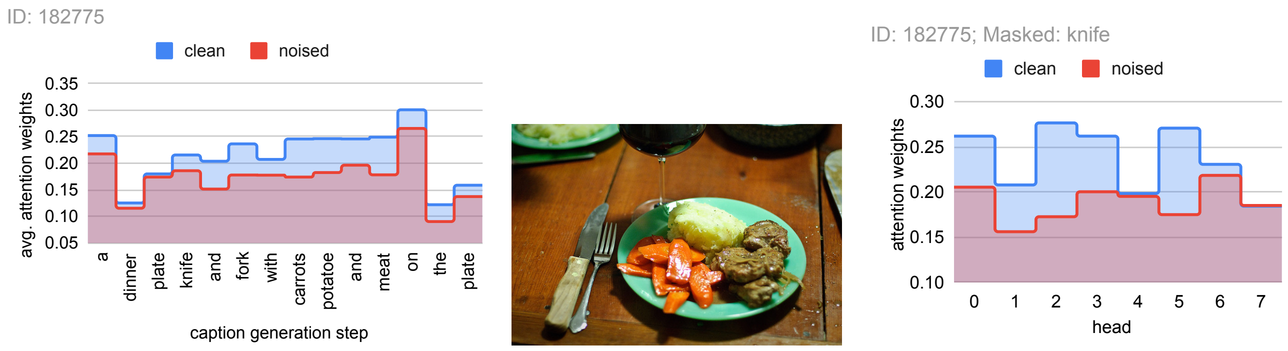


Figure 5. ***(left)*** The attention weights averaged across different attention heads for a noised view drop consistently at each caption generation step. ***(center)*** Input image with caption: "*a dinner plate knife and fork with carrots potatoes and meat on the plate*". ***(right)*** The attention weights of different attention heads drop consistently at the step of generating the word "*knife*", which is masked out in the input image.

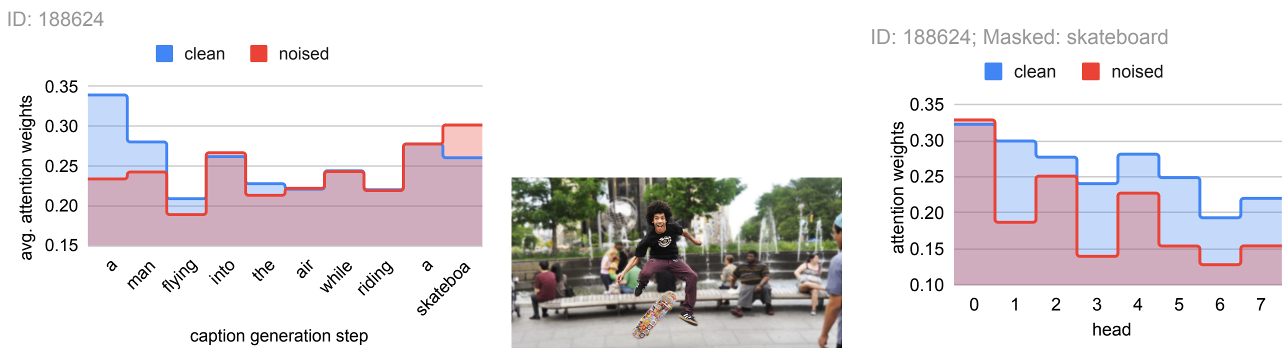


Figure 6. ***(left)*** The attention weights averaged across different attention heads for a noised view drop consistently at each caption generation step. ***(center)*** Input image with caption: "*a man flying into the air while riding a skateboard*". ***(right)*** The attention weights of different attention heads drop consistently at the step of generating the word "*skateboard*", which is masked out in the input image.

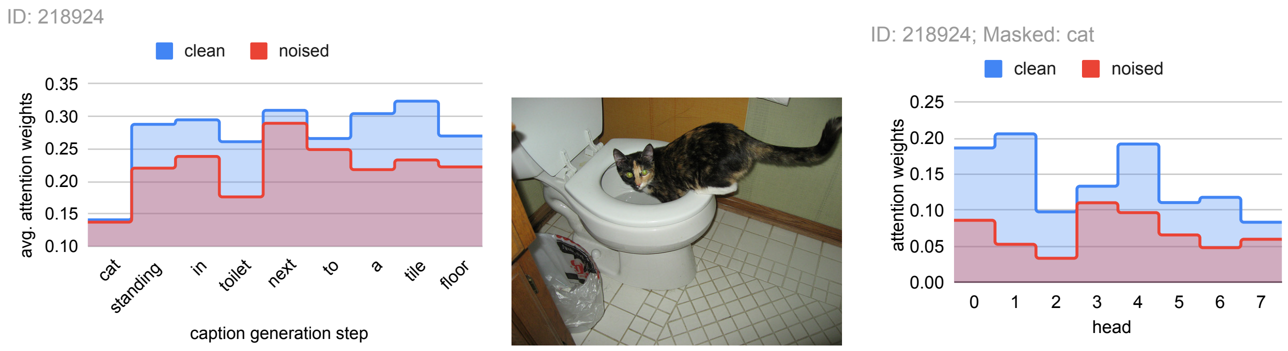


Figure 7. ***(left)*** The attention weights averaged across different attention heads for a noised view drop consistently at each caption generation step. ***(center)*** Input image with caption: "*cat standing in toilet next to a tile floor*". ***(right)*** The attention weights of different attention heads drop consistently at the step of generating the word "*cat*", which is masked out in the input image.

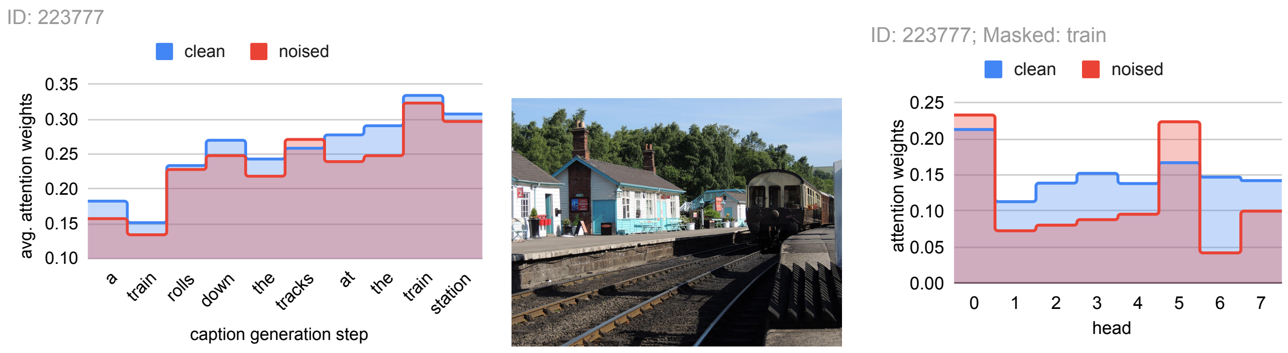


Figure 8. ***(left)*** The attention weights averaged across different attention heads for a noised view drop consistently at each caption generation step. ***(center)*** Input image with caption: "*a train rolls down the tracks at the train station*". ***(right)*** The attention weights of different attention heads drop consistently at the step of generating the word "*train*", which is masked out in the input image.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018. 1

[2] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020. 1

[3] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4634–4643, 2019. 1

[4] Chia-Wen Kuo and Zsolt Kira. Beyond a pre-trained object detector: Cross-modal textual and visual context for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[6] Yunpeng Luo, Jiayi Ji, Xiaoshuai Sun, Liujuan Cao, Yongjian Wu, Feiyue Huang, Chia-Wen Lin, and Rongrong Ji. Dual-level collaborative transformer for image captioning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 2286–2293, 2021. 1

[7] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10971–10980, 2020. 1

[8] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024, 2017. 1, 2