RIPL

Robotics Perception and Learning

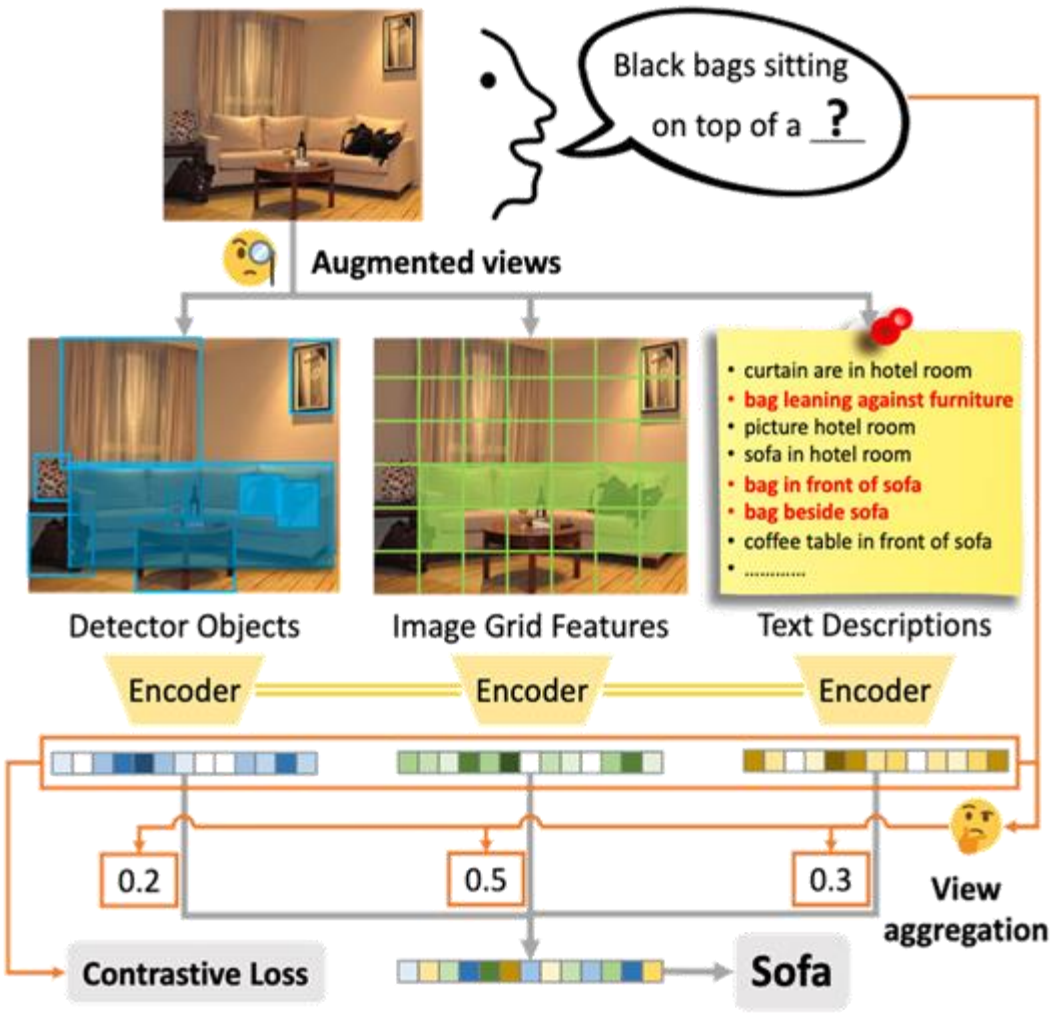
Georgia Tech

HAAV: Hierarchical Aggregation of Augmented Views for Image Captioning

Chia-Wen Kuo and Zsolt Kira

- Paper tag: WED-AM-267
  - Wed (Jun 21) morning session
  - West Building Exhibit Halls ABC
  - Poster# 267

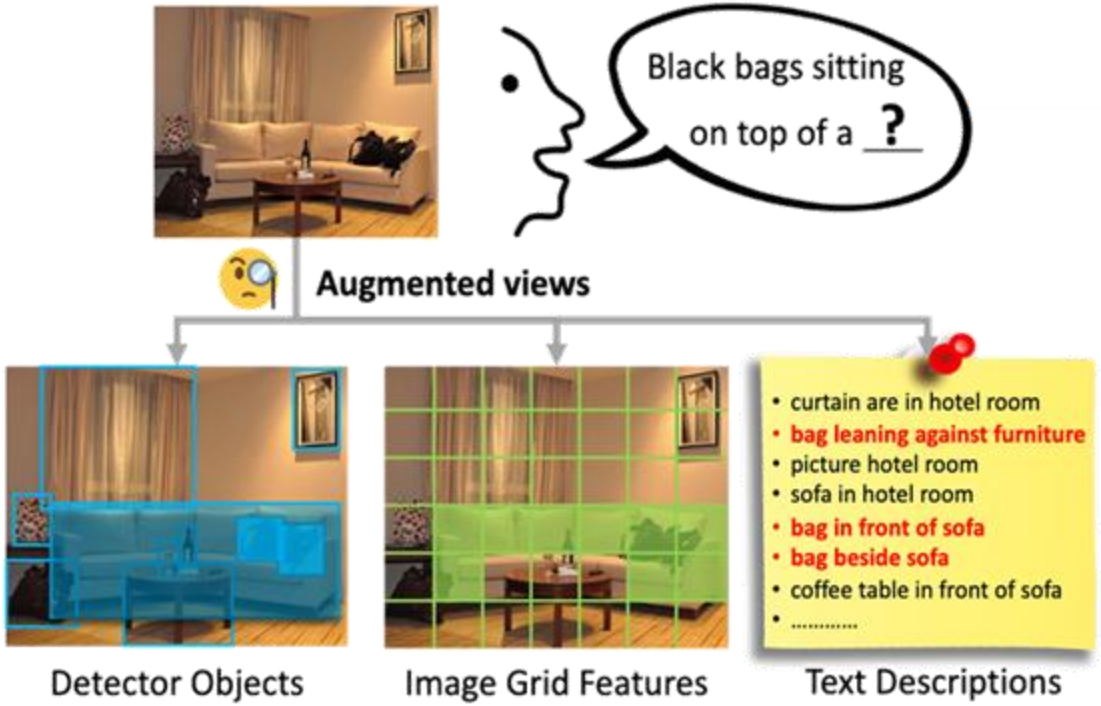
Summary



- Image captioning with multiple sources of input image encodings (or ***views***)
  - Detected objects, image grid features, text descriptions,……
- How to ***efficiently*** leverage these views?
  - Regard each view as an augmentation of the input image
  - Encode each view with a shared encoder independently
  - Add a contrastive loss across views
  - Improve computation, parameter, and label efficiency
- How to ***effectively*** leverage these views?
  - Our proposed hierarchical decoder layer
  - CrossAttn_{LV1} **models the effectiveness** of each view
  - CrossAttn_{LV2} **adaptively aggregates each view** according to their effectiveness

Problem Statement

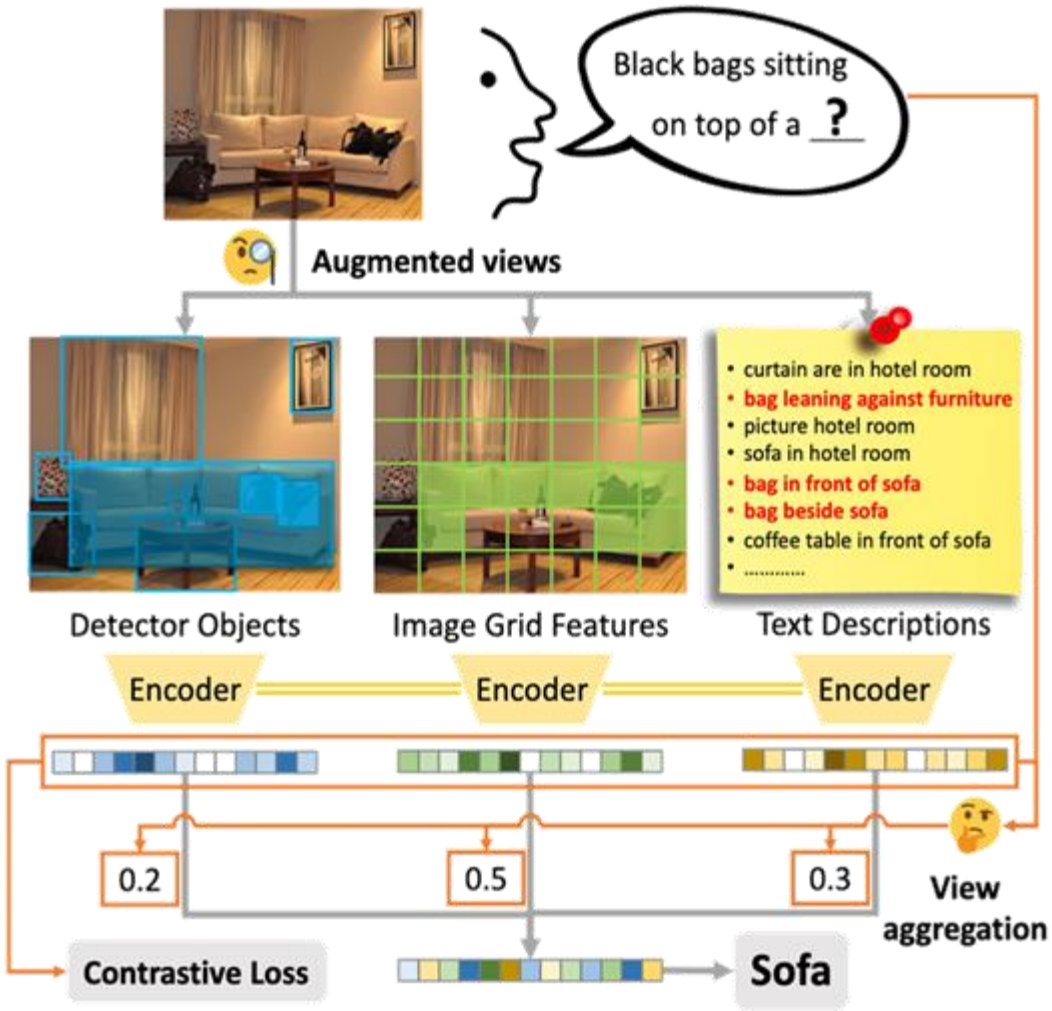
- Vision and language tasks (e.g. image captioning)
  - Multiple sources of input image encodings (or ***views***) from different pre-trained models
  - E.g. detected objects, image grid features, retrieved texts



- Encode-decoder transformer model
- How to efficiency encode these views?
  - Concatenation of views: computationally inefficient $O(V^2)$
  - Train a separate encoder for each view: Parameter inefficient $O(V)$
  - Transformer models are intrinsically data-hungry: Label inefficient
- How to effectively decode these views?
  - Different views have different noise levels
    - When predicting the next word of sofa
    - Object detector may fail to detect sofa
    - Down weigh the view of detected objects
    - Leverage other views with sofa info
  - Adaptively weigh each view according to their effectiveness

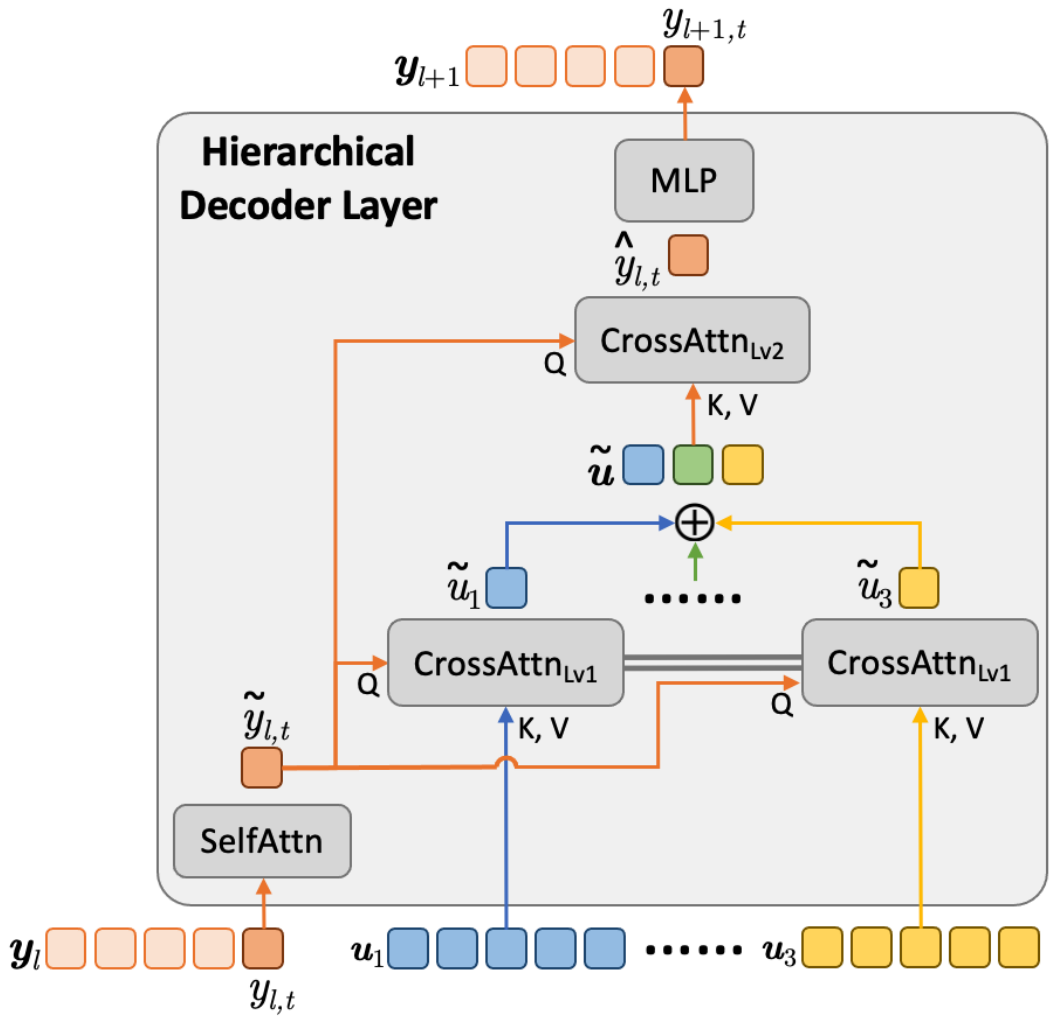
How to encode views efficiently

- Regard each view as an augmentation of the input image
- Just like what we do with data augmentations
  - Encode each view independently
    - Computational efficiency
    - $O(V^2) \Rightarrow O(V)$
  - Encode each view with a shared encoder
    - Parameter efficiency
    - $O(V) \Rightarrow O(1)$
  - Contrastive loss across views
    - Help representation learning
    - Label efficiency
    - Novel semi-supervised training



How to decode views effectively

- Our proposed hierarchical decoder layer adaptively aggregates the views according to their effectiveness
- $\text{CrossAttn}_{\text{LV1}}$
  - Aggregate within each view at the token level individually
  - Model the effectiveness of each view
- $\text{CrossAttn}_{\text{LV2}}$
  - Aggregate across views at the view level jointly
  - Adaptively aggregate each view according to their effectiveness



Main results

- Setting
  - MS-COCO dataset
  - HAAV is a trained-from-scratch small model
- Results
  - Compared with other large-scaled pre-trained larger models
  - HAAV achieves comparable and usually better performance.

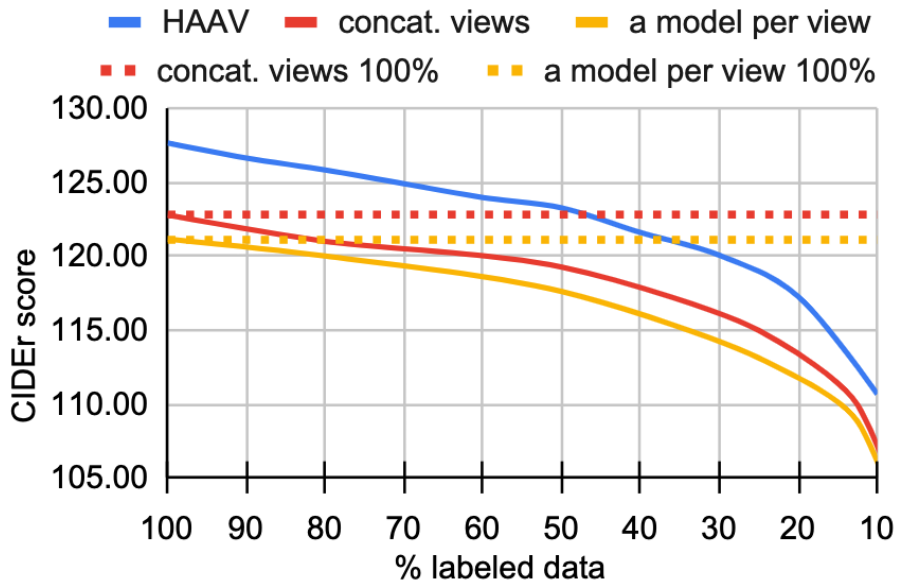
Method	Pretrain Data	B-4	M	C	S
VLP [62]	3M	39.5	29.3	129.3	23.2
X-VLM [59]	16M	40.4	-	139.3	-
Oscar [36]	6.5M	40.5	29.7	137.6	22.8
VinVL [60]	8.8M	40.9	30.9	140.4	**25.1**
GIT [52]	4M	**41.3**	30.4	139.1	24.3
ViTCap [13]	4M	41.2	30.1	138.1	24.1
$\text{SimVLM}_{\text{base}}$ [54]	1.8B	39.0	**32.9**	134.8	24.0
HAAV (ours)	None	41.0	30.2	**141.5**	23.9

- Setting
  - Supervised learning: Flickr30k dataset
  - Semi-supervised learning (SSL)
    - Labeled : Flickr30k dataset
    - Unlabeled: MS-COCO images
- Results
  - Compared with other trained-from-scratch small models
  - HAAV achieves substantial improvement
  - HAAV + SSL achieves additional improvement

Method	B-4	M	C	S
Show & Tell [53]	21.5	18.3	41.7	12.2
Show, Attend & Tell [57]	23.6	19.2	49.1	13.3
Up-Down [4]	28.3	21.6	63.3	15.9
$\mathcal{M}^2$ [10]	29.8	22.4	68.4	16.2
ORT [19]	30.1	22.8	68.8	16.9
HAAV (ours)	**34.3**	24.6	81.7	18.0
HAAV + SSL (ours)	**34.3**	**25.1**	**85.6**	**19.0**

Analyses for efficiency

- HAAV has better computation, parameter, and label efficiency
- Despite being more efficient, HAAV achieves the best performance
- We do not sacrifice performance in pursuit of efficiency

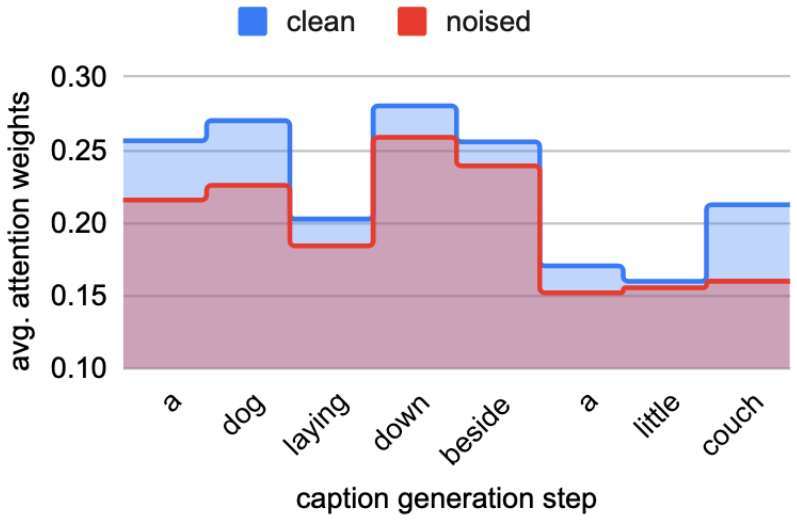


Conditions	$\mathcal{L}_{\text{con}}$	Computation		Parameter		Performance			
		complexity	iter/sec ↑	complexity	#params ↓	B-4	M	C	S
Model per View		$\mathcal{O}(\|V\|)$	2.23	$\mathcal{O}(\|V\|)$	52.5M	38.5	28.6	121.1	21.3
Concatenated views		$\mathcal{O}(\|V\|^2)$	4.20	$\mathcal{O}(1)$	13.1M	38.5	28.7	122.8	21.7
Unshared encoders		$\mathcal{O}(\|V\|)$	5.33	$\mathcal{O}(\|V\|)$	20.8M	39.7	29.1	125.4	22.1
Unshared encoders	✓	$\mathcal{O}(\|V\|)$	3.96	$\mathcal{O}(\|V\|)$	22.7M	39.7	29.3	125.8	22.2
Shared encoder		$\mathcal{O}(\|V\|)$	5.97	$\mathcal{O}(1)$	13.5M	39.7	29.1	125.6	22.1
Ours (Shared encoder)	✓	$\mathcal{O}(\|V\|)$	4.83	$\mathcal{O}(1)$	15.4M	**40.5**	**29.4**	**127.6**	**22.3**

Analyses for effectiveness

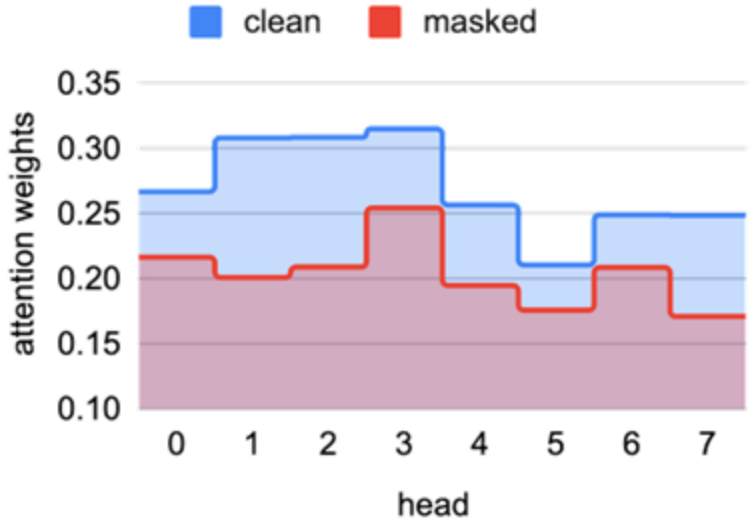
Adaptively weigh each view according to their effectiveness

- Add random noise to a view
  - Measure the attention weight at each caption generation step
  - The attention weight toward the noised view drops consistently across generation steps
- Mask out dog in one of the input views
  - Measure the attention weight at the step of generating "**dog**"
  - The attention weight toward the masked view drops consistently across heads



(a) Attention weights averaged across heads for a noised view at each caption generation step.

(b) Input image with caption: *a dog laying down beside a little couch*



(c) Attention weights of different attention heads at the step of generating "*dog*", which is masked out in the input image.

Thank You

- More details of HAAV can be found in the paper
- We release our code, and the augmented views

Paper tag: WED-AM-267

RIPL
Robotics Perception and Learning

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Georgia Tech