THE UNIVERSITY OF BRITISH COLUMBIA

Topics in AI (CPSC 532S):
Multimodal Learning with Vision, Language and Sound

Lecture 13: RNN Applications (Part 3)

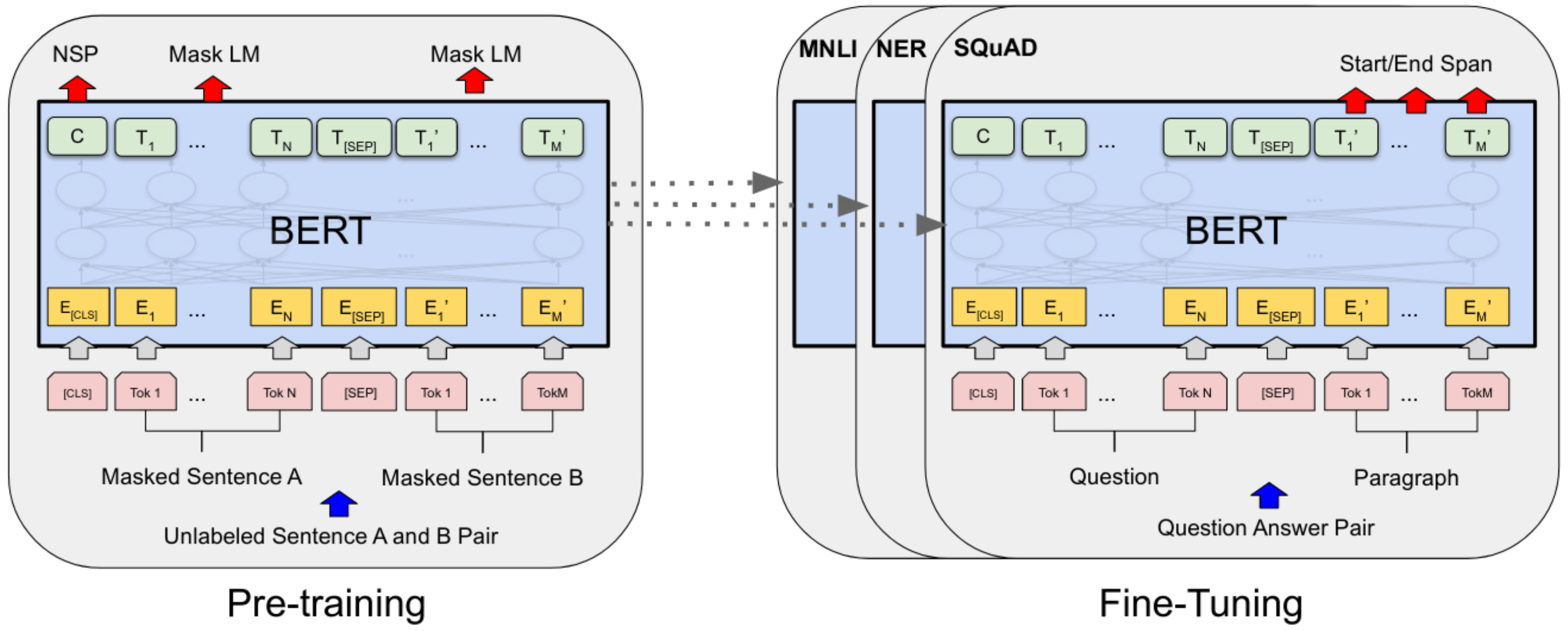
Logistics

Assignment 1 & 2 will be posted by Monday

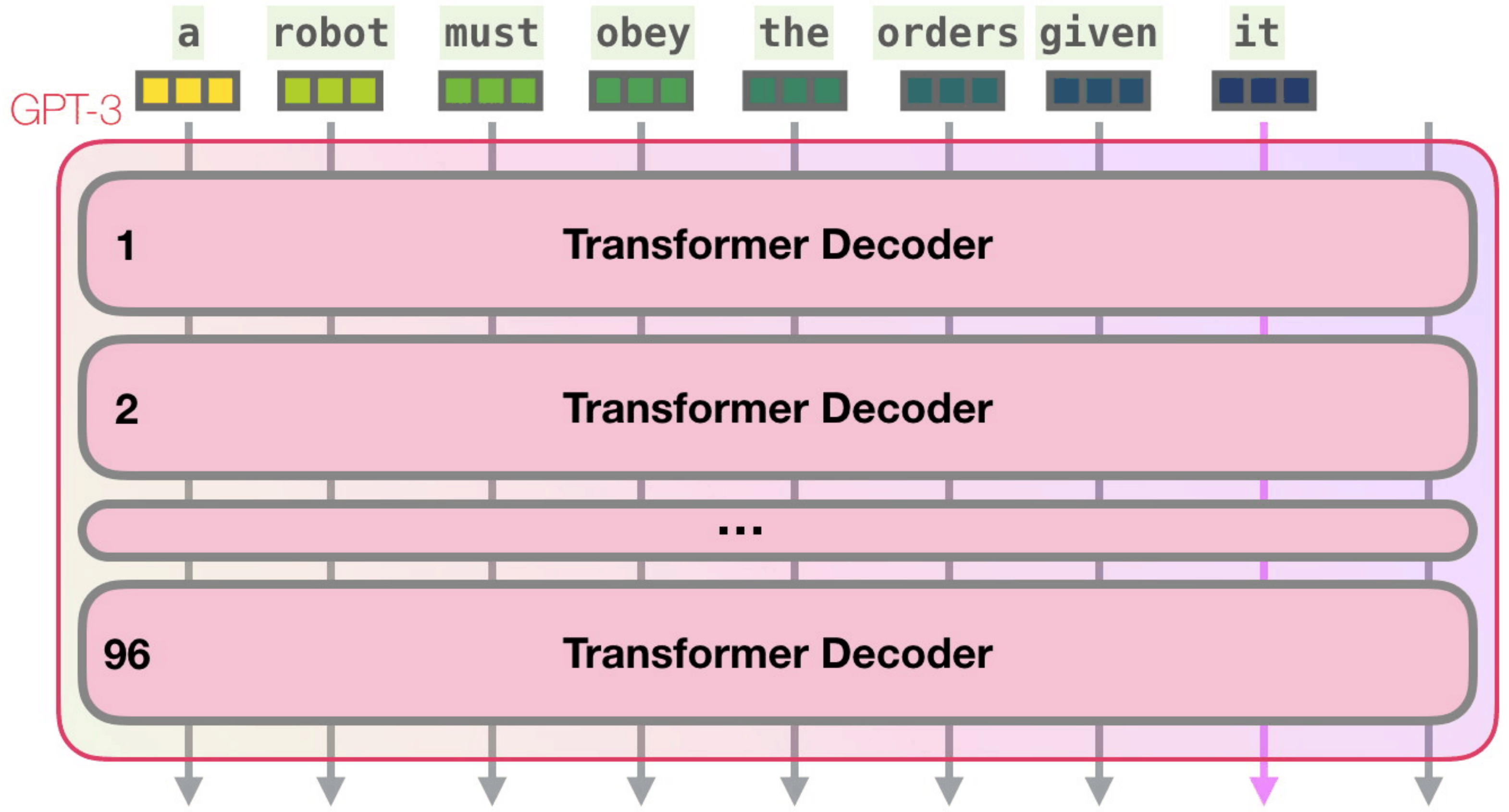
Group formation — **due today**

Brief **Review** + Lessons

BERT

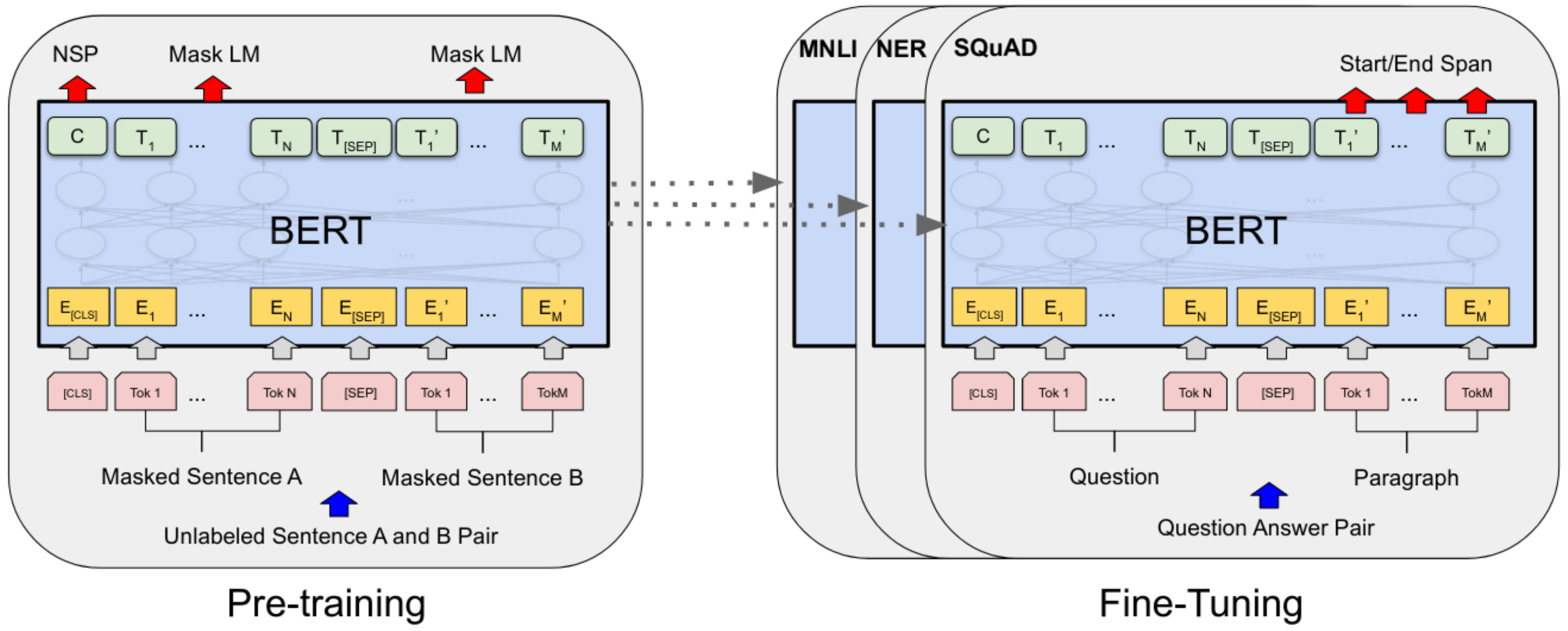


GPT3

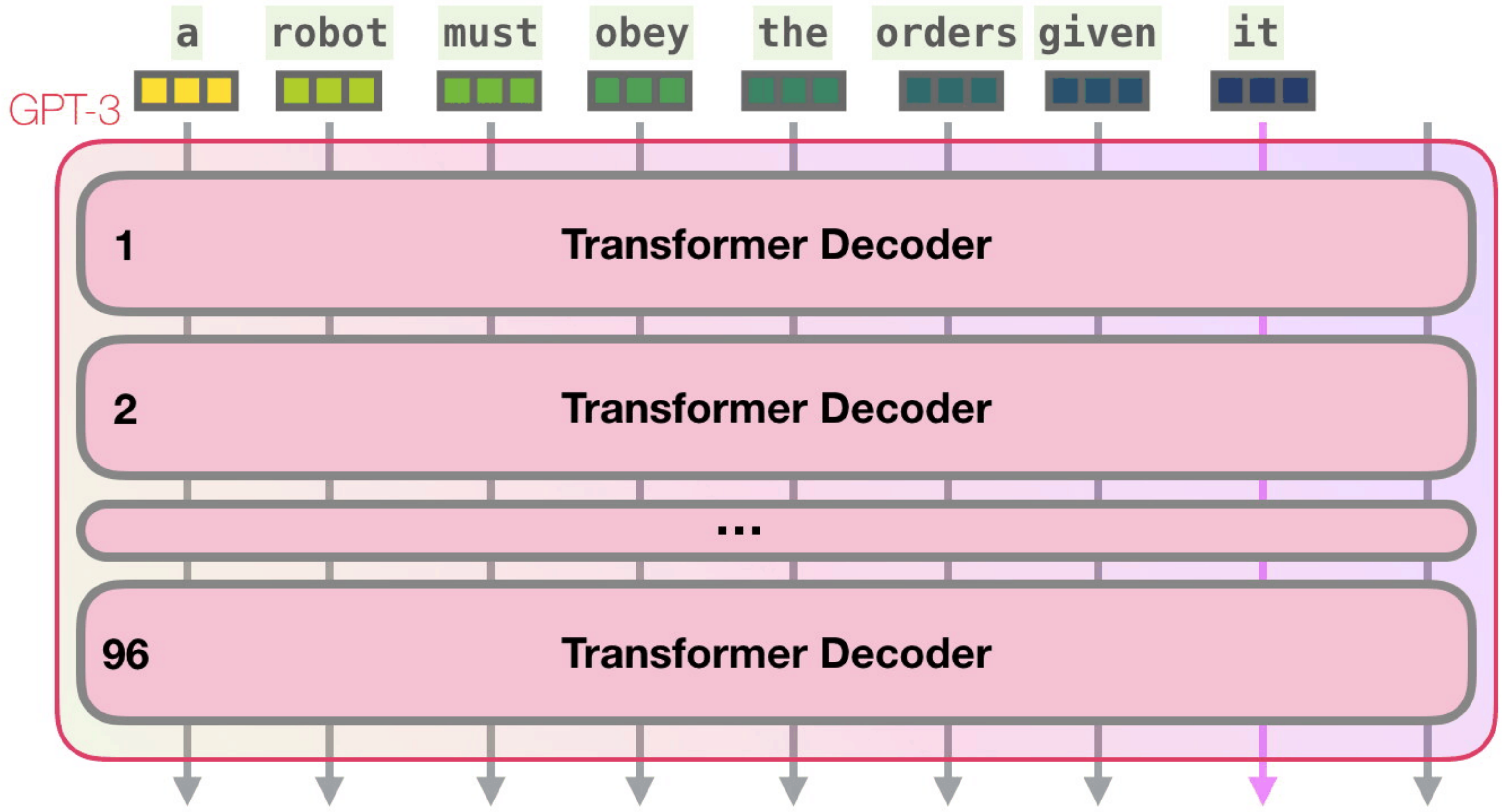


Brief **Review** + Lessons

BERT

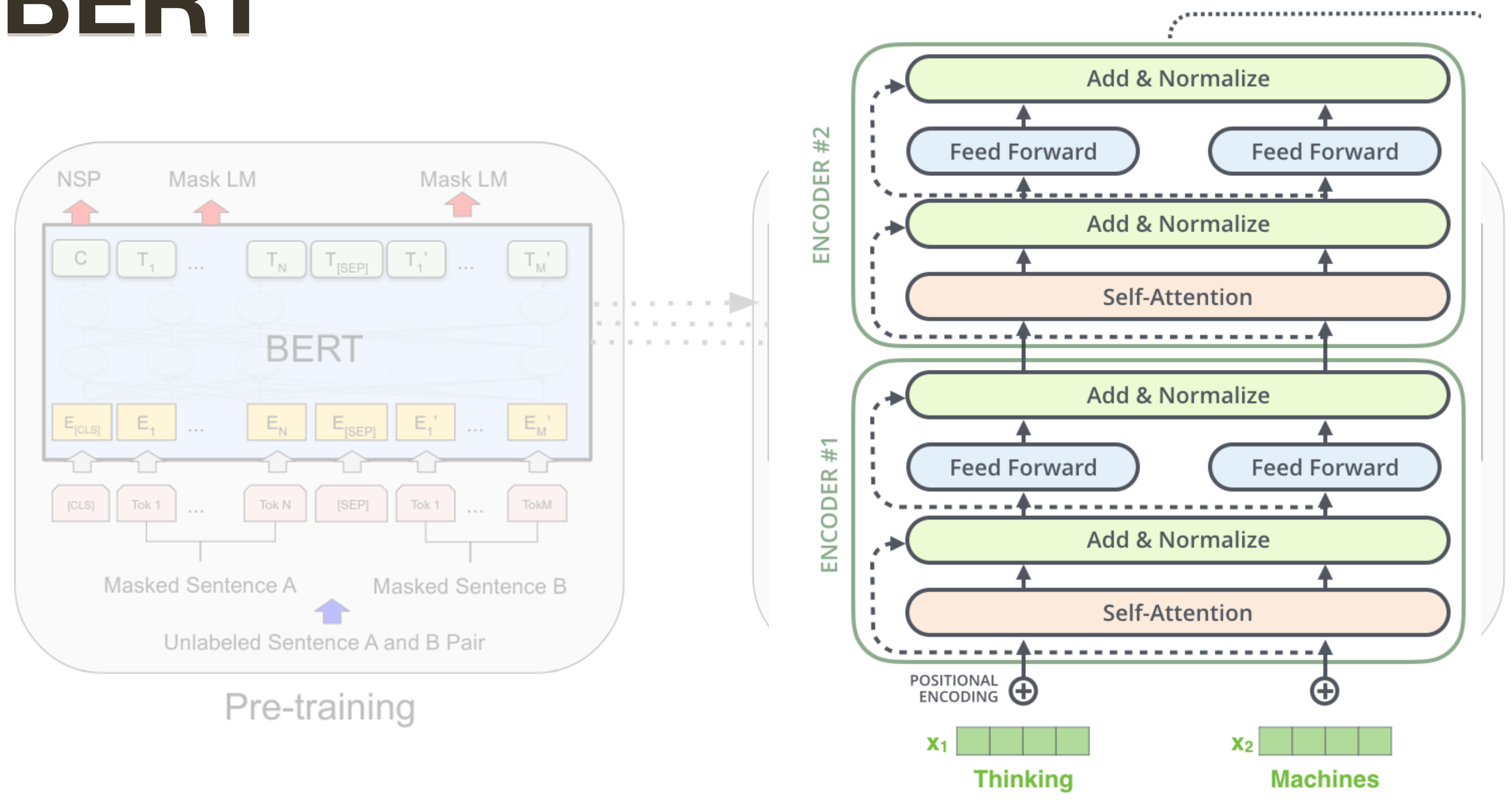


GPT3



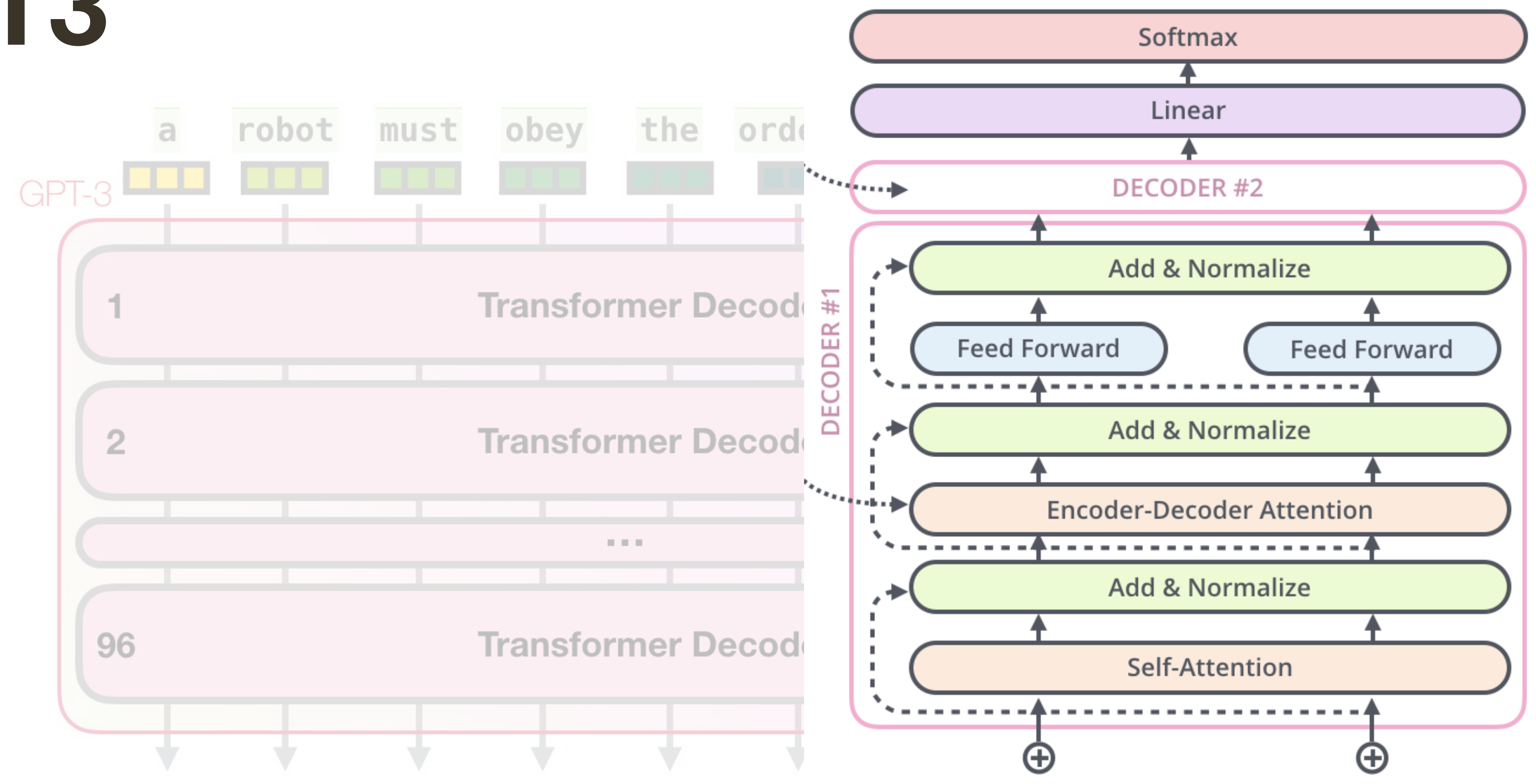
Brief **Review** + Lessons

BERT



— Encoder part of the Transformer

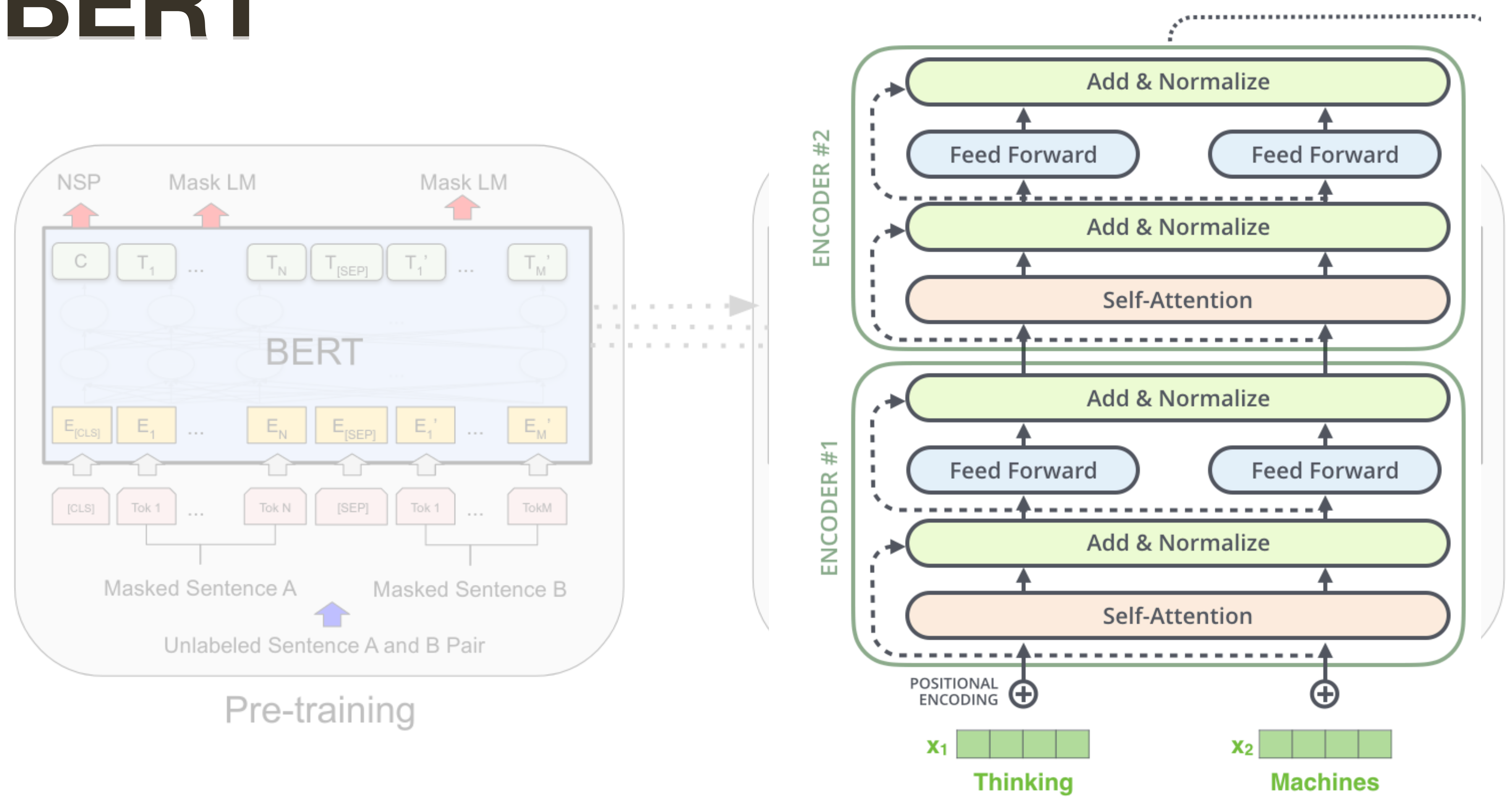
GPT3



— Decoder part of the Transformer

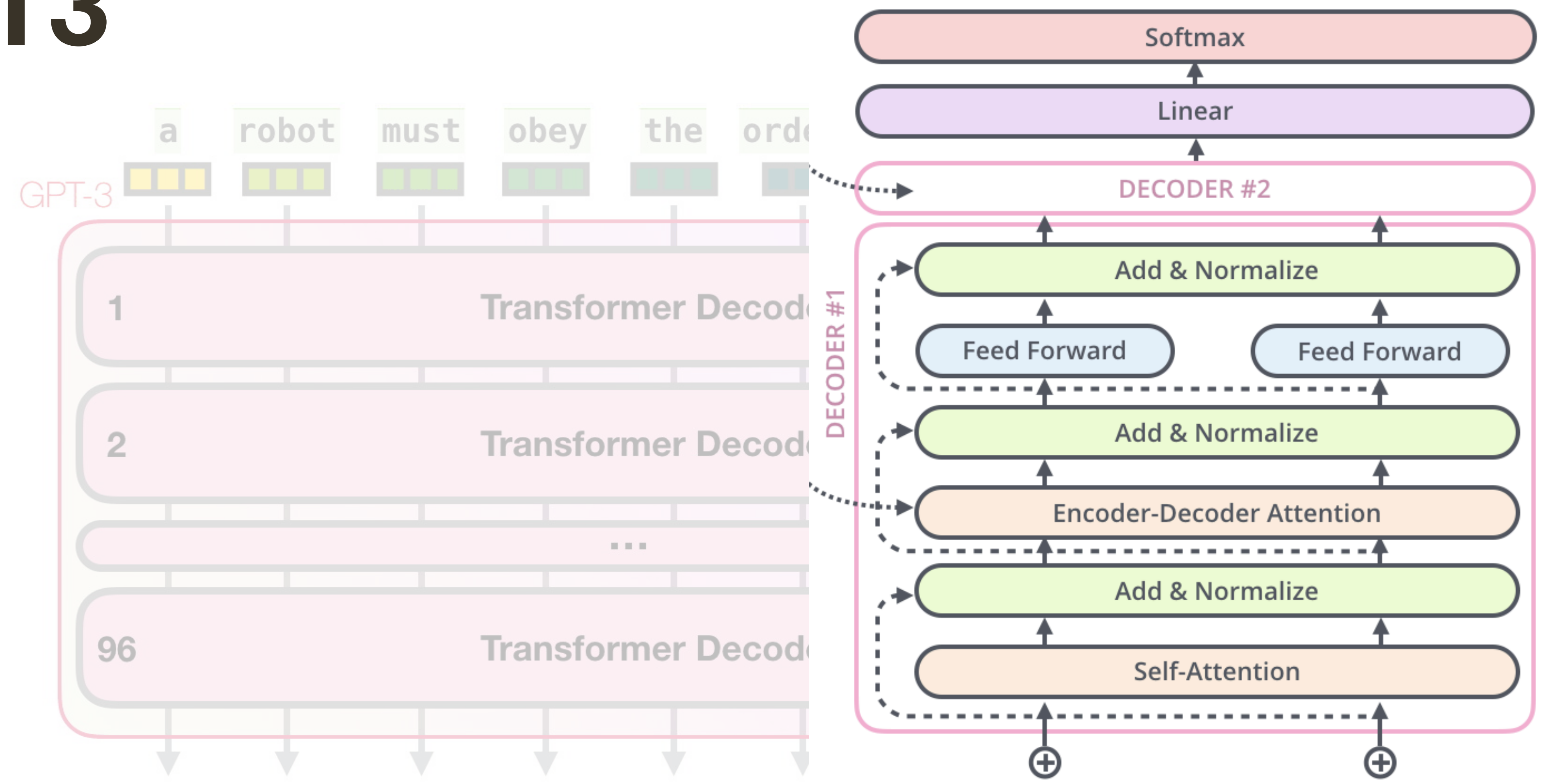
Brief **Review** + Lessons

BERT



— Encoder part of the Transformer

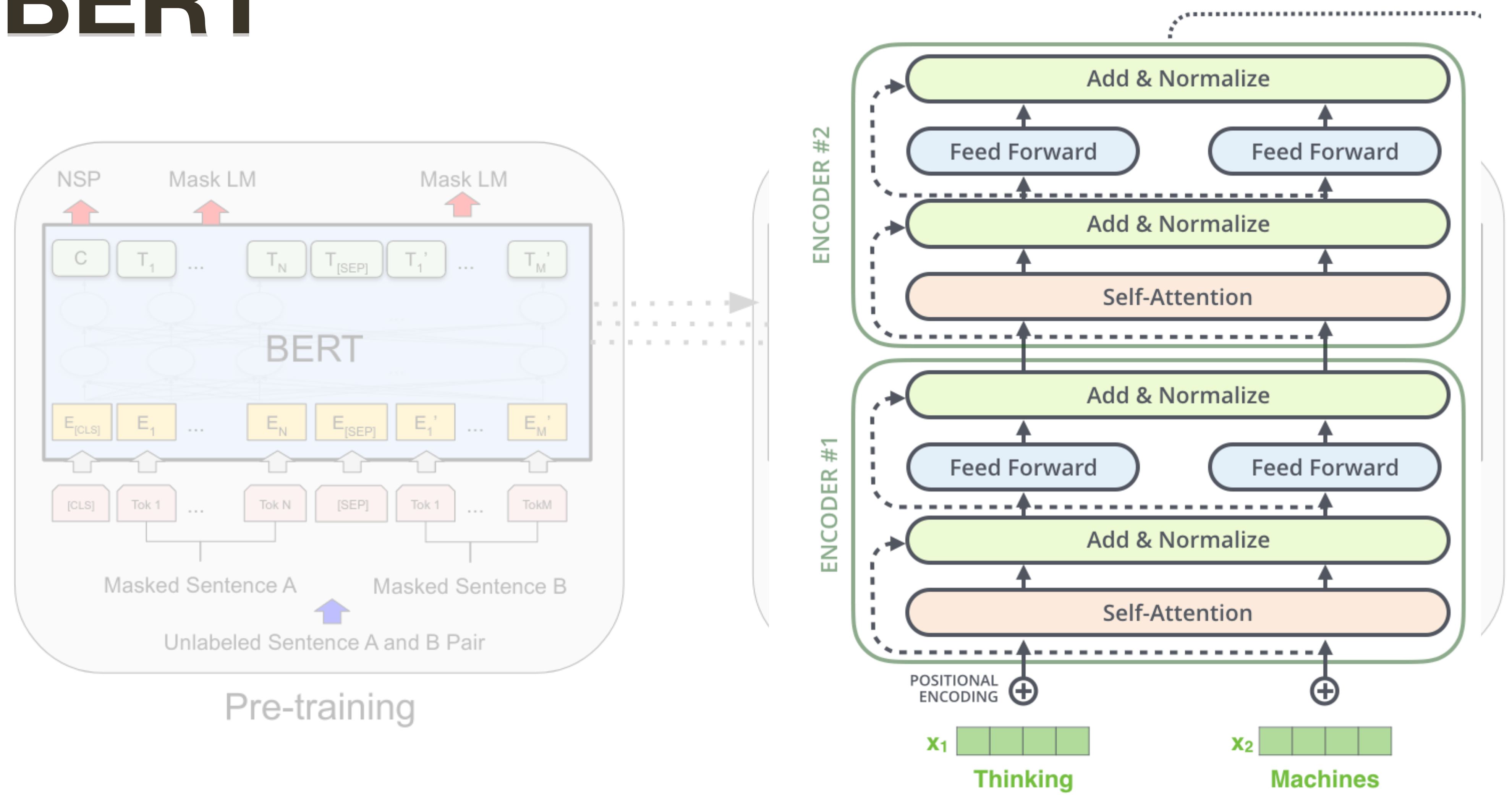
GPT3



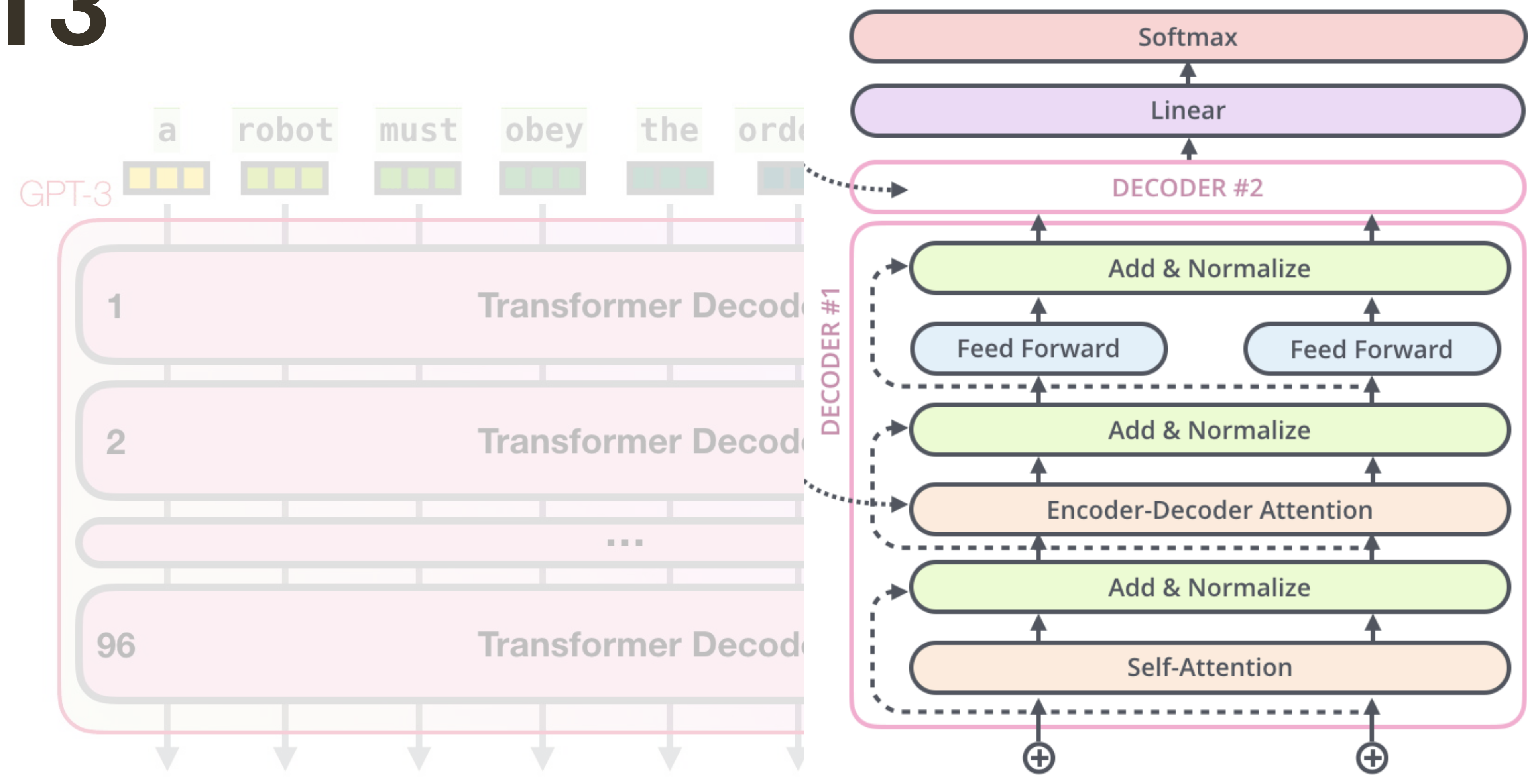
— Decoder part of the Transformer

Brief **Review** + Lessons

BERT



GPT3

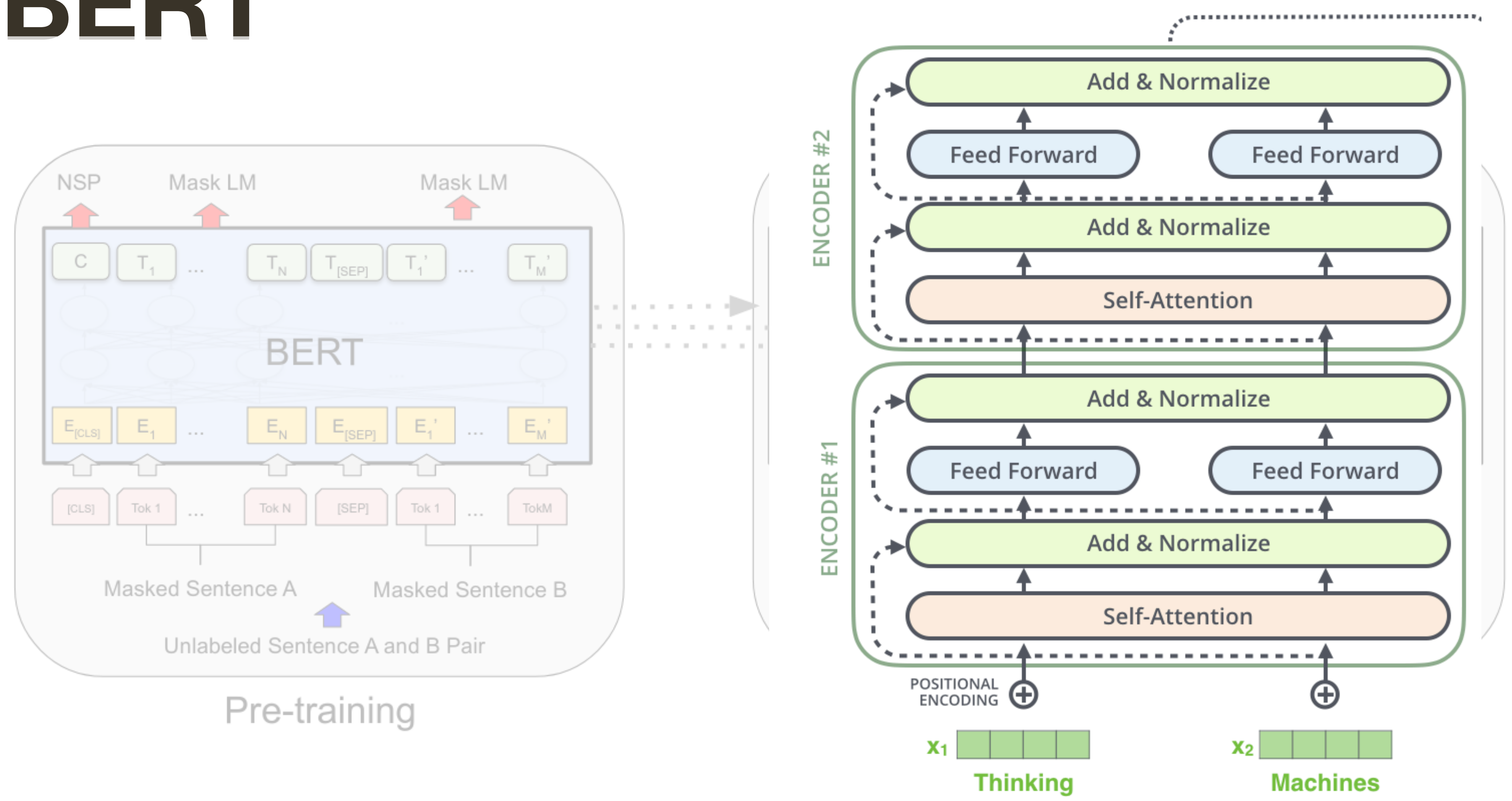


— Neither BERT nor GPT3 is really a “model” on its own, more like a **training strategy**

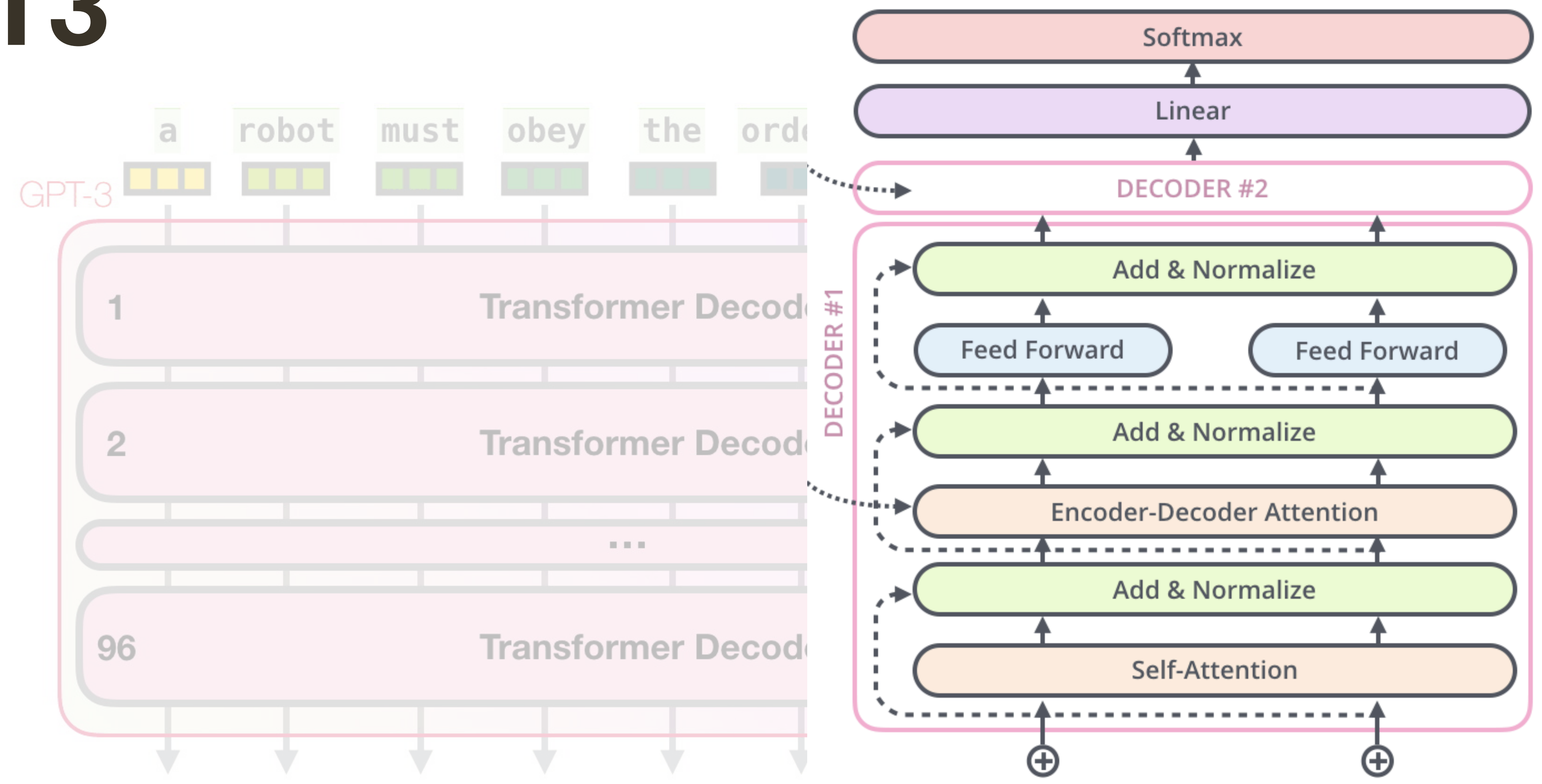
— Success of both stems from **large capacity** of this models and **extensive amounts of training data** (+ compute needed to train them)

Brief **Review** + Lessons

BERT



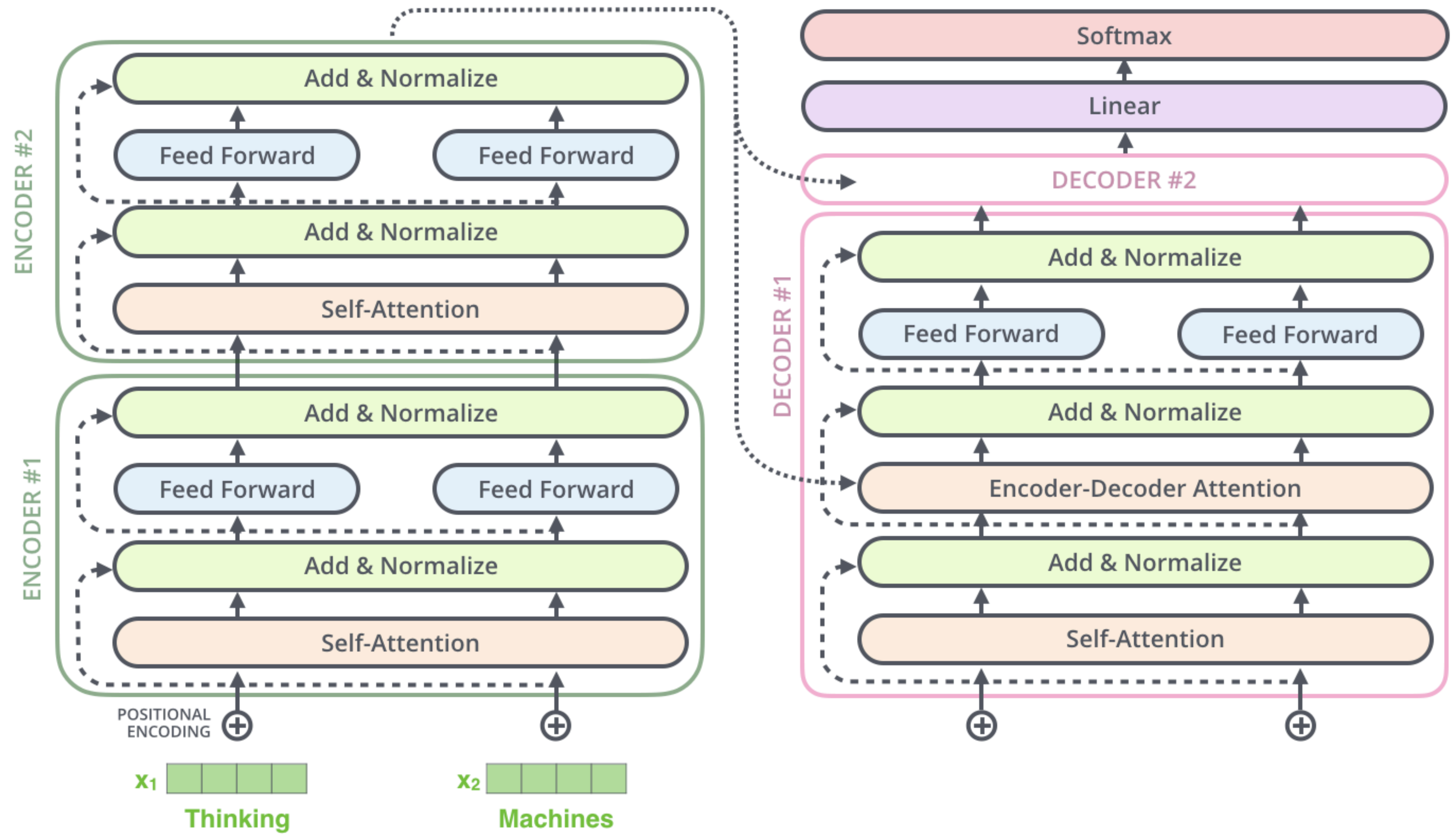
GPT3



— Neither BERT nor GPT3 is really a “model” on its own, more like a **training strategy**

— Success of both stems from **large capacity** of this models and **extensive amounts of training data** (+ compute needed to train them)

Why **Transformers** are so Effective?



— (Globally) **contextualized** representations -> better capable of capture meaning

— Allow **parallelized** training -> enables training with large amounts of data

— **Residual** layer structure -> good gradient flow for optimization

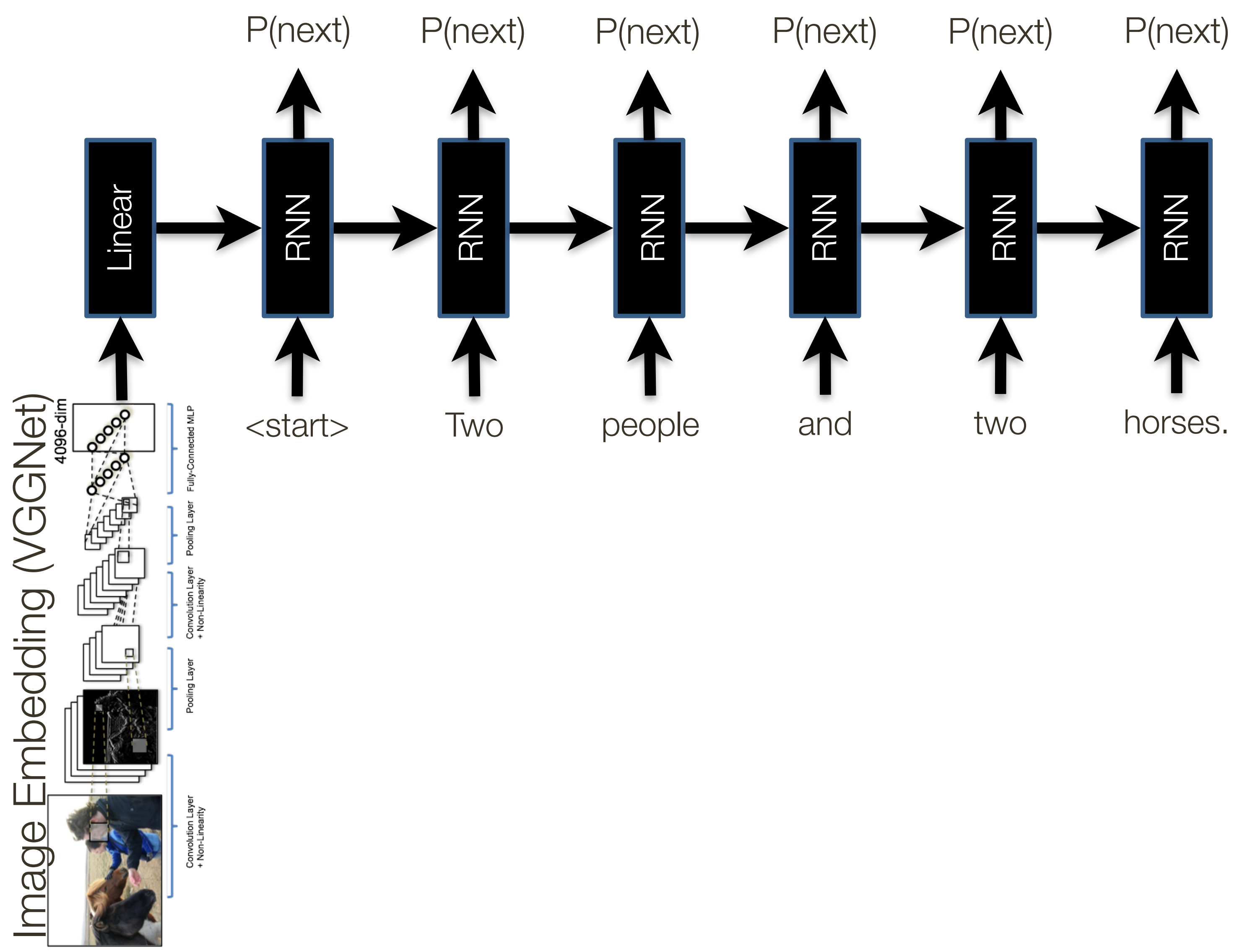
Brief **Review** + Lessons

Captioning, Visual Question Answering **(VQA)** :

— Encoders for images (e.g., CNNs) produce a vector-based representations

— Encoder for language (e.g., RNNs) also produces vector-based representation

This makes it very easy to combine encoders/decoders cross-modally to solve variety of visio-lingual tasks



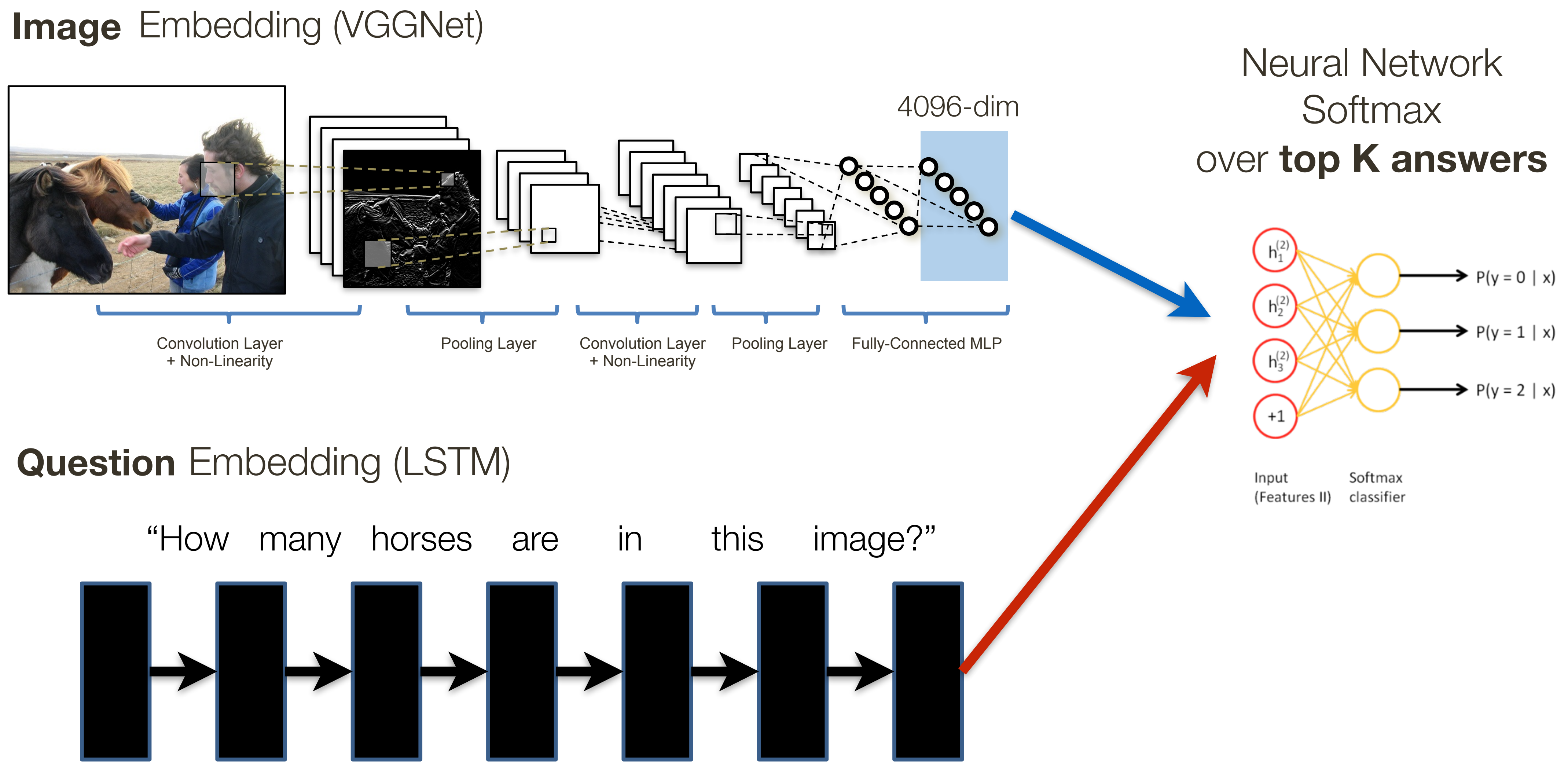


Image Embedding (VGGNet)
4096-dim
Convolution Layer + Non-Linearity
Pooling Layer
Convolution Layer + Non-Linearity
Pooling Layer
Fully-Connected MLP
Neural Network Softmax over top K answers
P(y = 0 | x)
P(y = 1 | x)
P(y = 2 | x)
+1
Input (Features II)
Softmax classifier
Question Embedding (LSTM)
“How many horses are in this image?”

Brief **Review** + Lessons

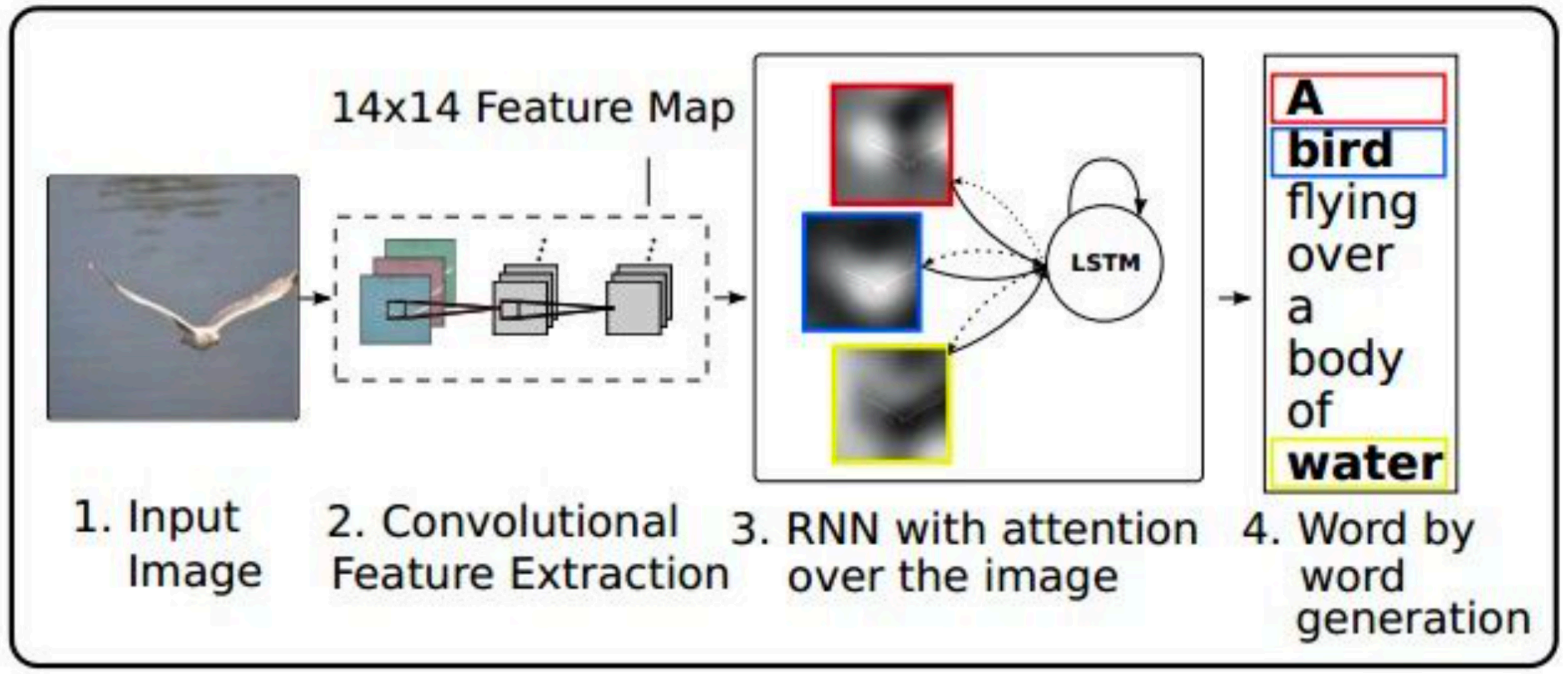
Captioning, Visual Question Answering **(VQA)** :

— Encoders for images (e.g., CNNs) produce a vector-based representations

— Encoder for language (e.g., RNNs) also produces vector-based representation

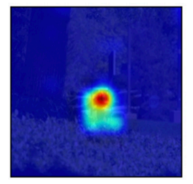
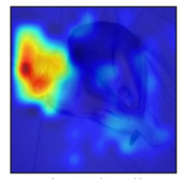
This makes it very easy to combine encoders/decoders cross-modally to solve variety of visio-lingual tasks

Note: Attention can be applied to images by treating each (x,y) feature column as effectively an encoder "token"


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Brief **Review** + Lessons — Visual **Dialogs**

You can use soft attention mechanisms as “**memory**” modules by simply modulating what is used for Keys and Values.

Question Turn	Key (hash)	Memory
1	f (**H:** Empty; **Q:** What color is a hydrant? **A:** It is red)	
2	f (**H:** …; **Q:** Is there a tree? **A:** Yes)	

You can (easily) **modify attention mechanisms** to encode priors that the problem may have, such as recency.

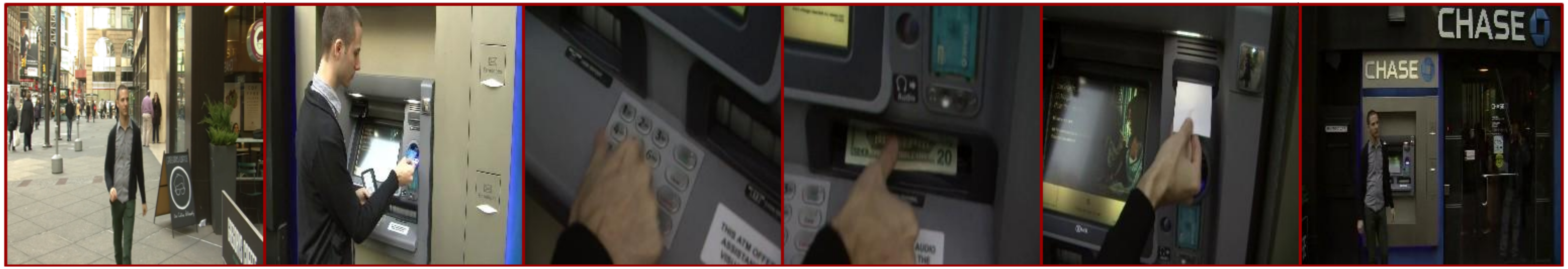
* learnable parameter

$$m_{t,\tau} = \left(\boldsymbol{W}^{\text{mem}}\boldsymbol{c}_t\right)^\top \boldsymbol{k}_\tau \boxed{+\,\theta\left(t-\tau\right)}$$

$$\boldsymbol{\beta}_t = \text{softmax}\left(\{m_{t,\tau}, 0 < \tau < t-1\}\right)$$

… and treating images as sequences

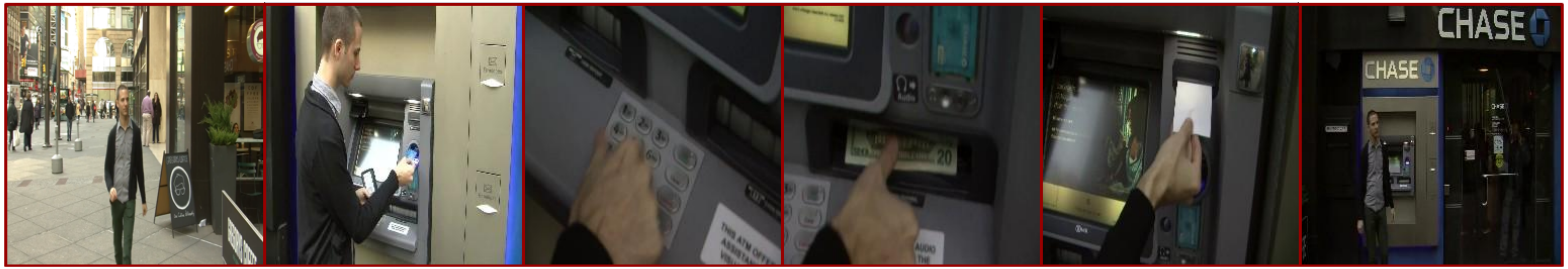
Applications: Activity Detection



[Ma et al., 2014]

Applications: Activity Detection

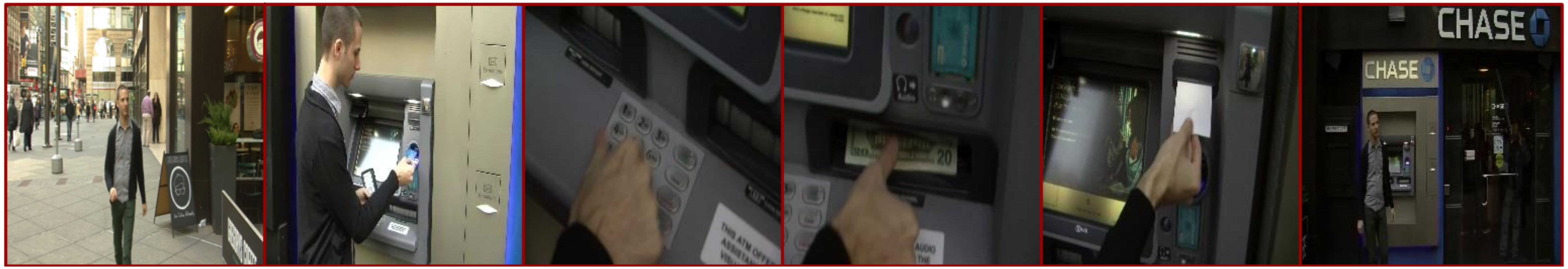
Activity: A collection of human/object movements with a particular semantic meaning



[Ma et al., 2014]

Applications: Activity Detection

Activity: A collection of human/object movements with a particular semantic meaning

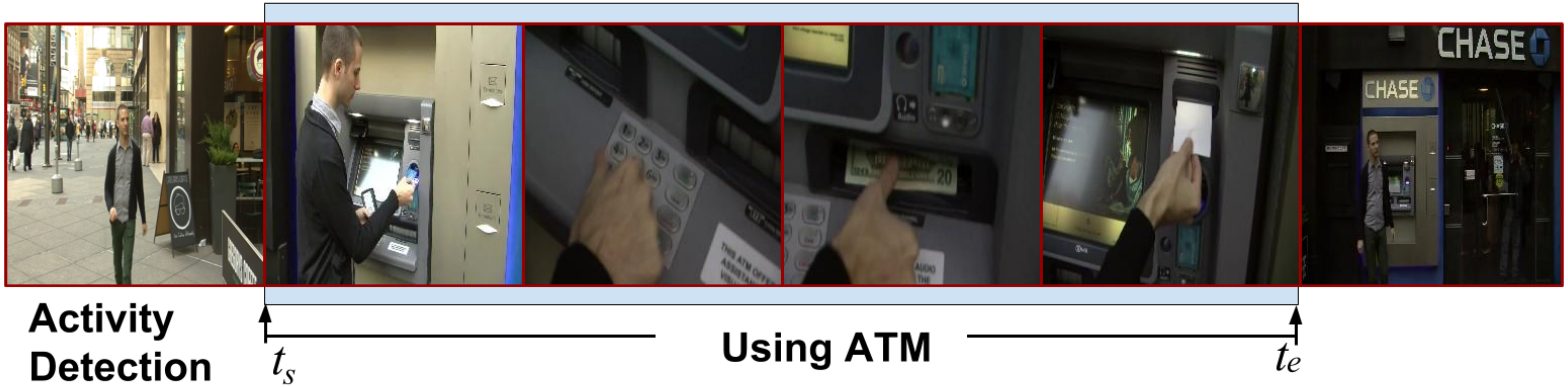


Action Recognition: Finding if a video segment contains such a movement

[Ma et al., 2014]

Applications: Activity Detection

Activity: A collection of human/object movements with a particular semantic meaning

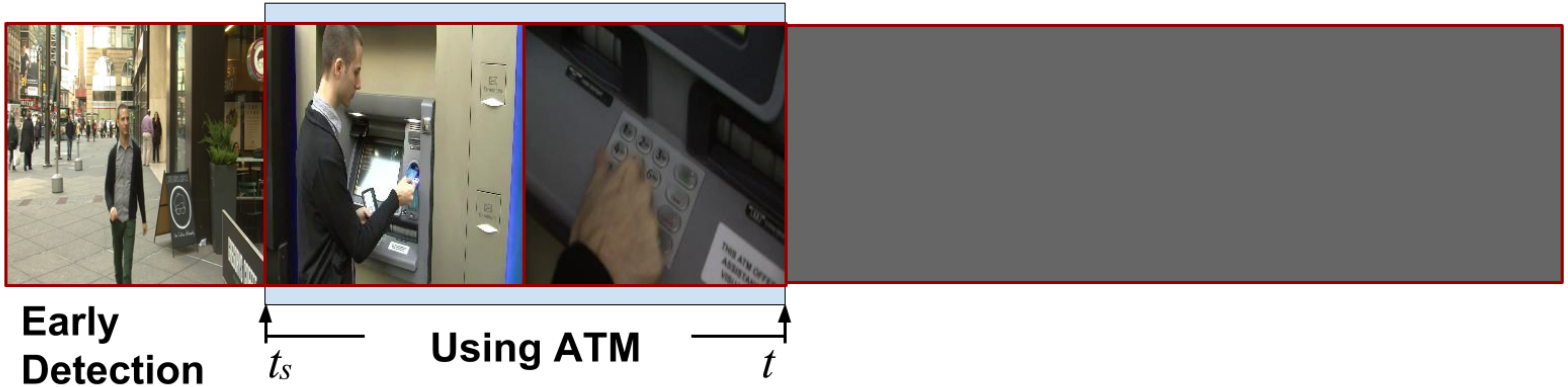


Action Recognition: Finding if a video segment contains such a movement

Action Detection: Finding a segment (beginning and start) and recognize the action in it

[Ma et al., 2014]

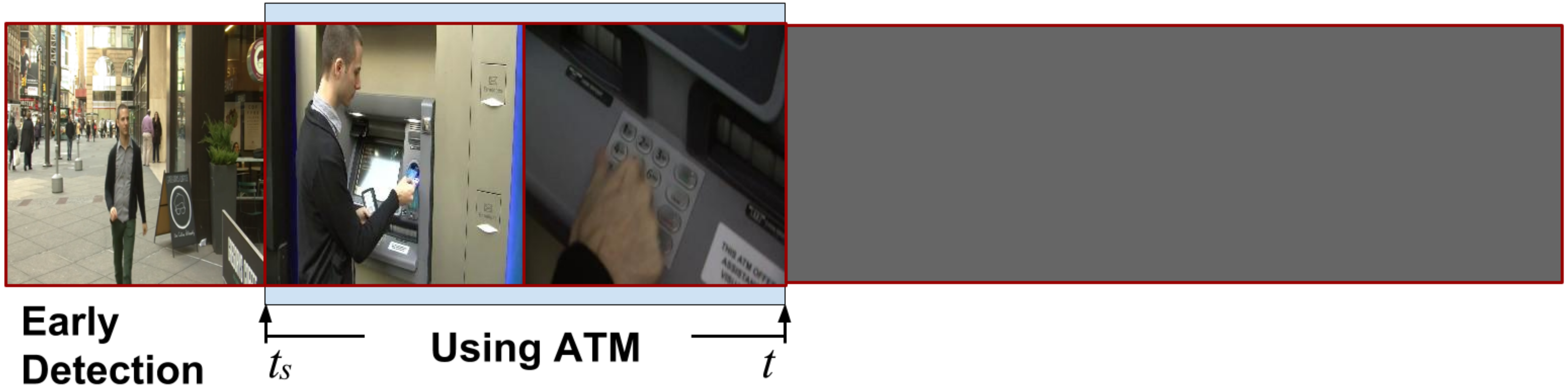
Applications: Activity Detection



[Ma et al., 2014]

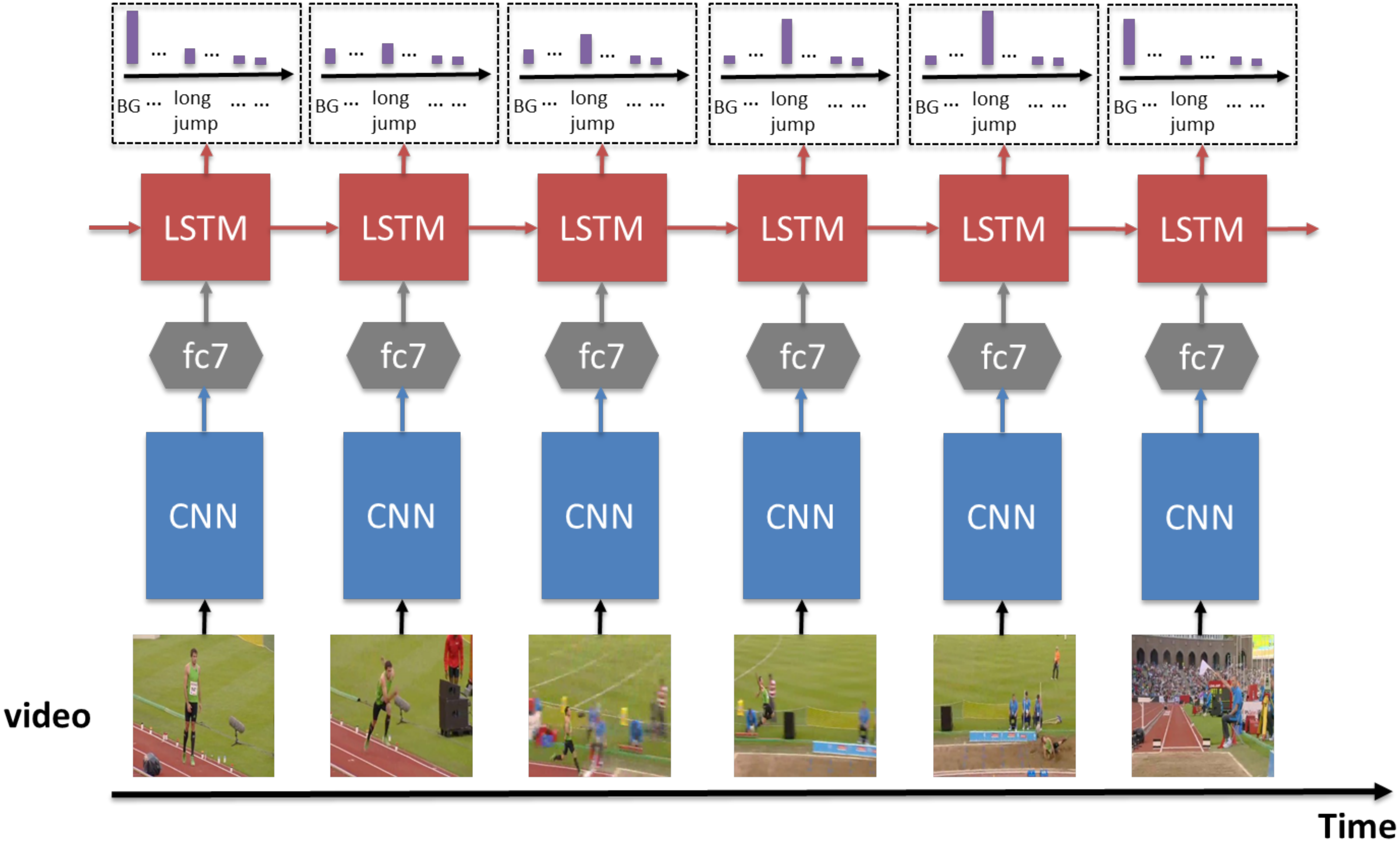
Applications: Activity Detection

Early Detection: Recognize when an action starts and try to predict which action is performed as quickly as possible.



[Ma et al., 2014]

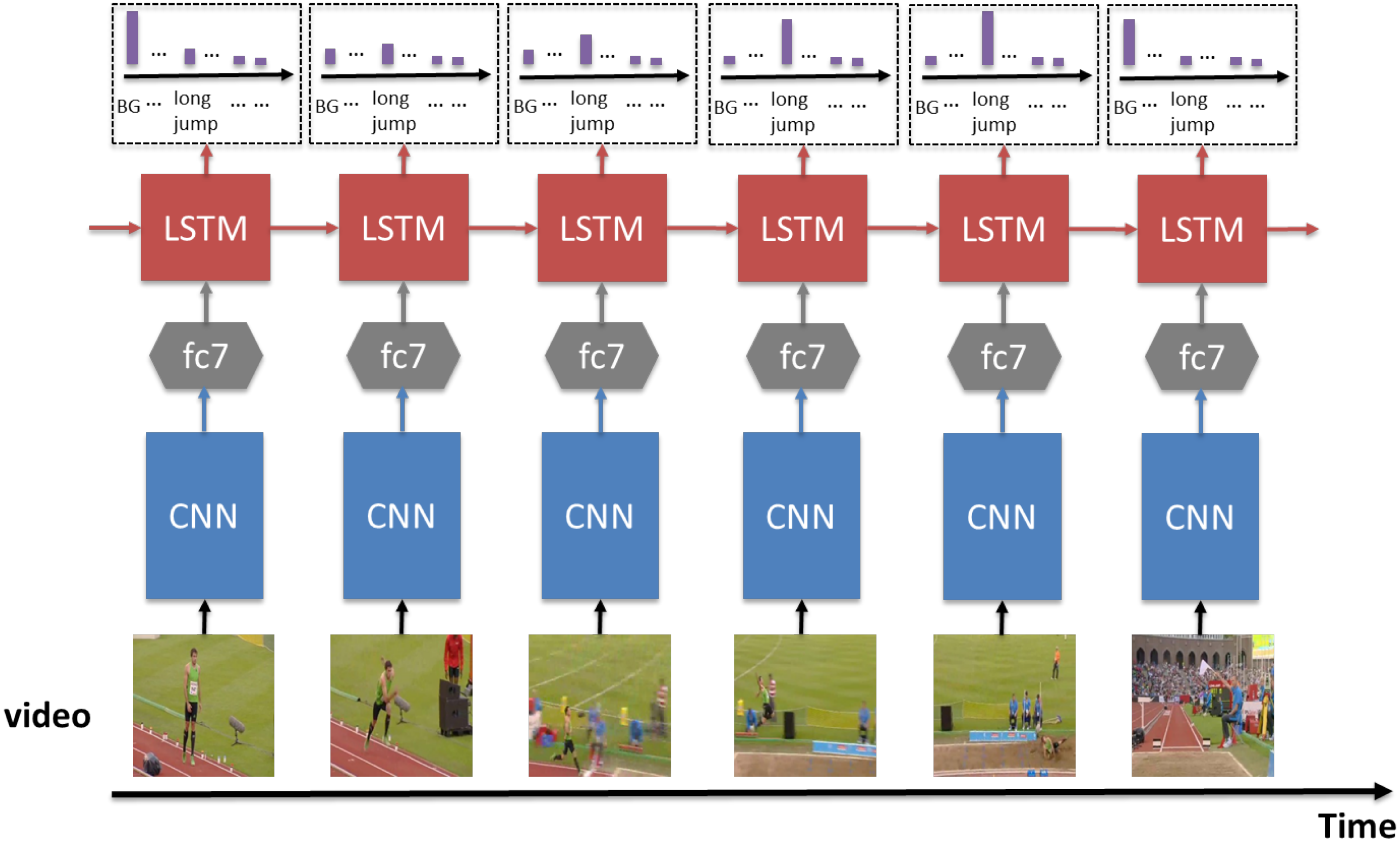
Applications: Activity Detection



[Ma et al., 2014]

Applications: Activity Detection

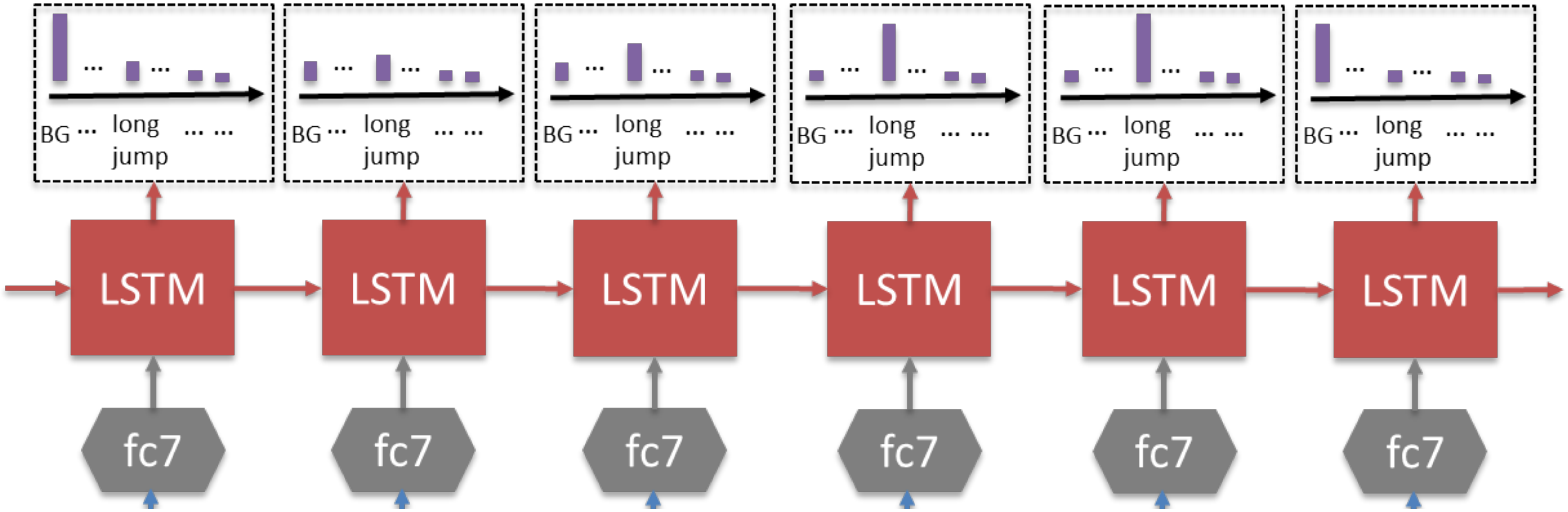
Penalty at every time step is the same

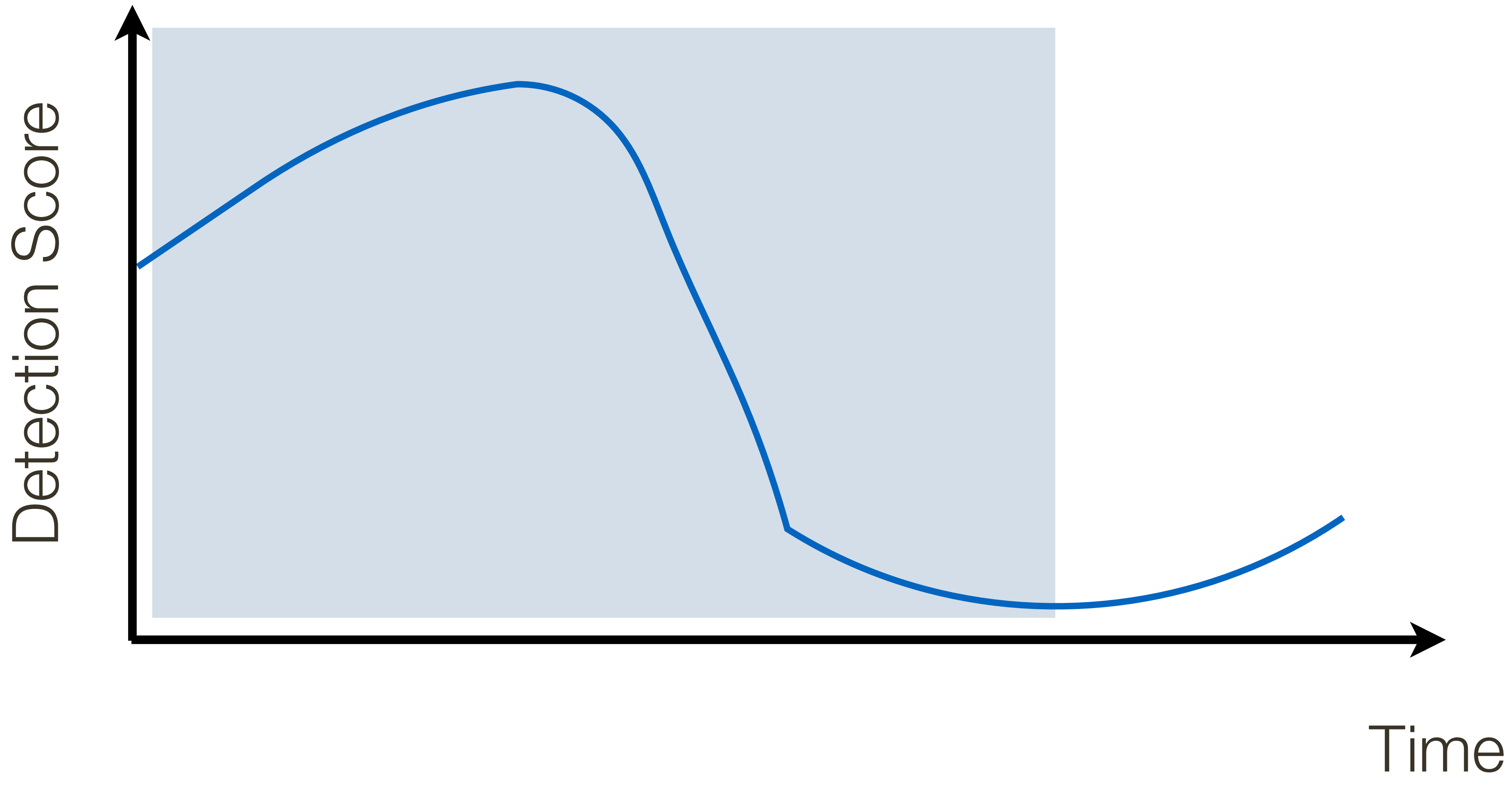


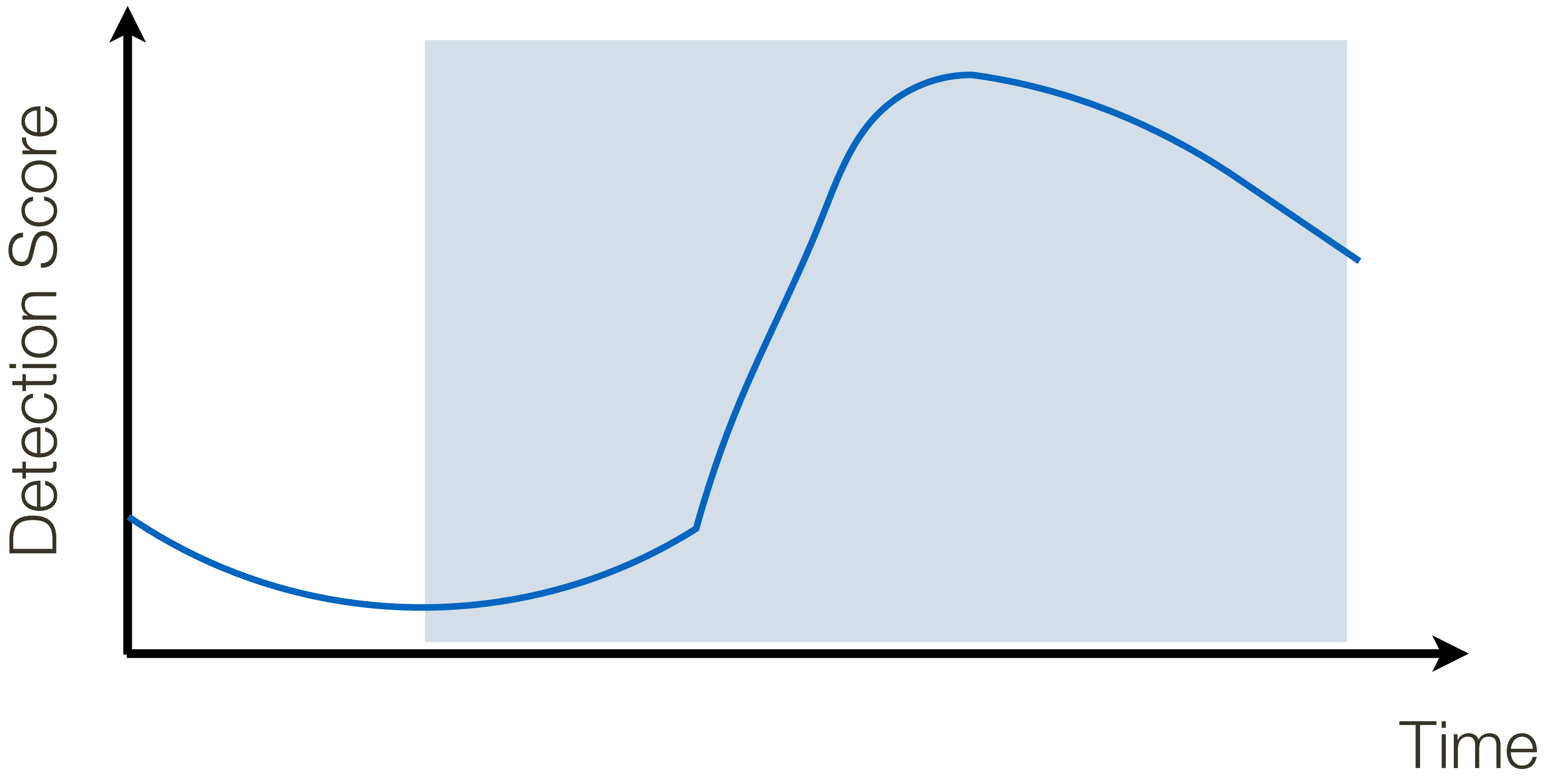
[Ma et al., 2014]

Applications: Activity Detection

Penalty at every time step is the same







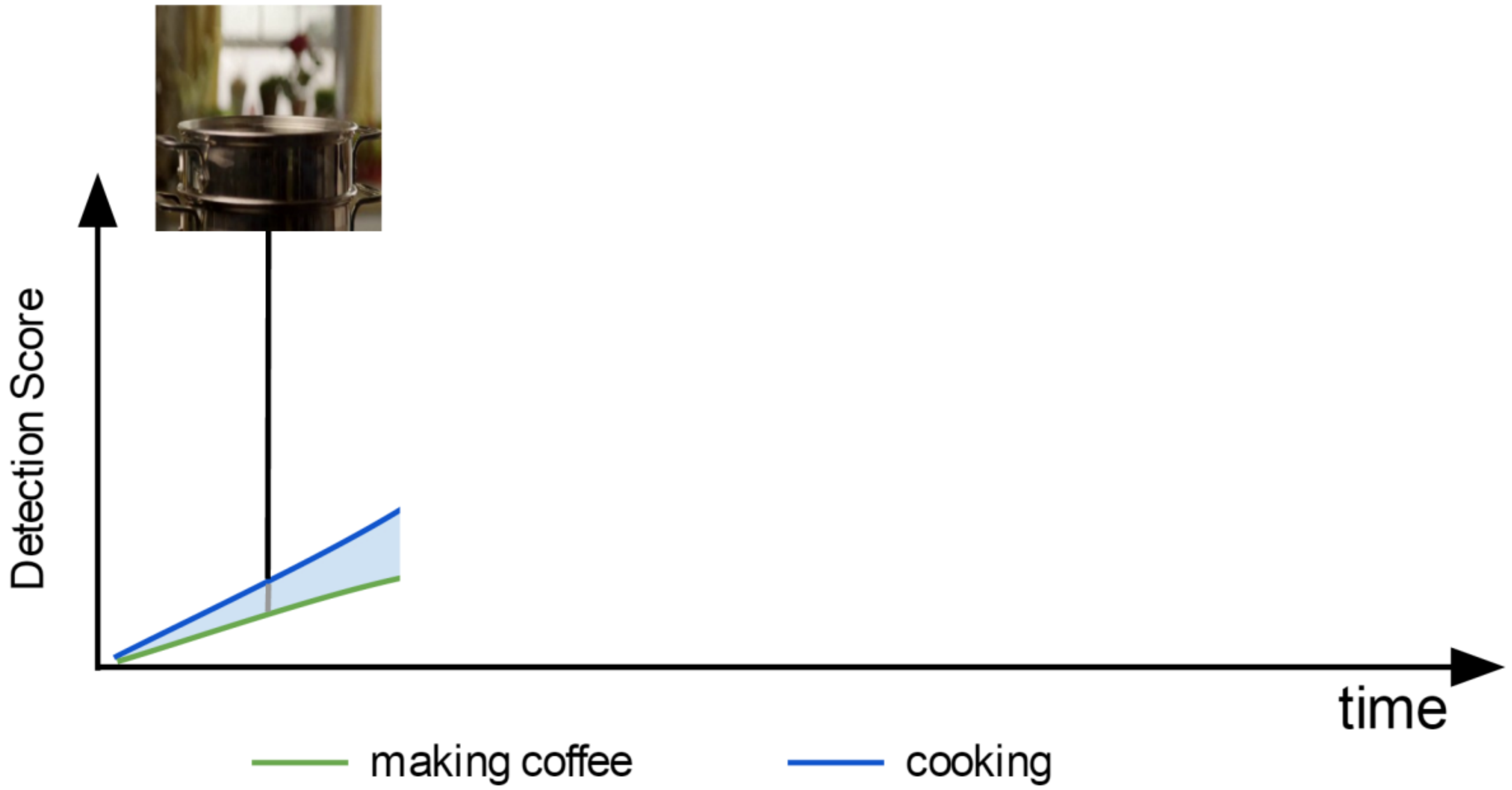
[Ma et al., 2014]

Applications: Activity Detection

As the detector sees more of an action, it should become more confident of

— Detecting the correct action class

— More confident that it is not the incorrect action class



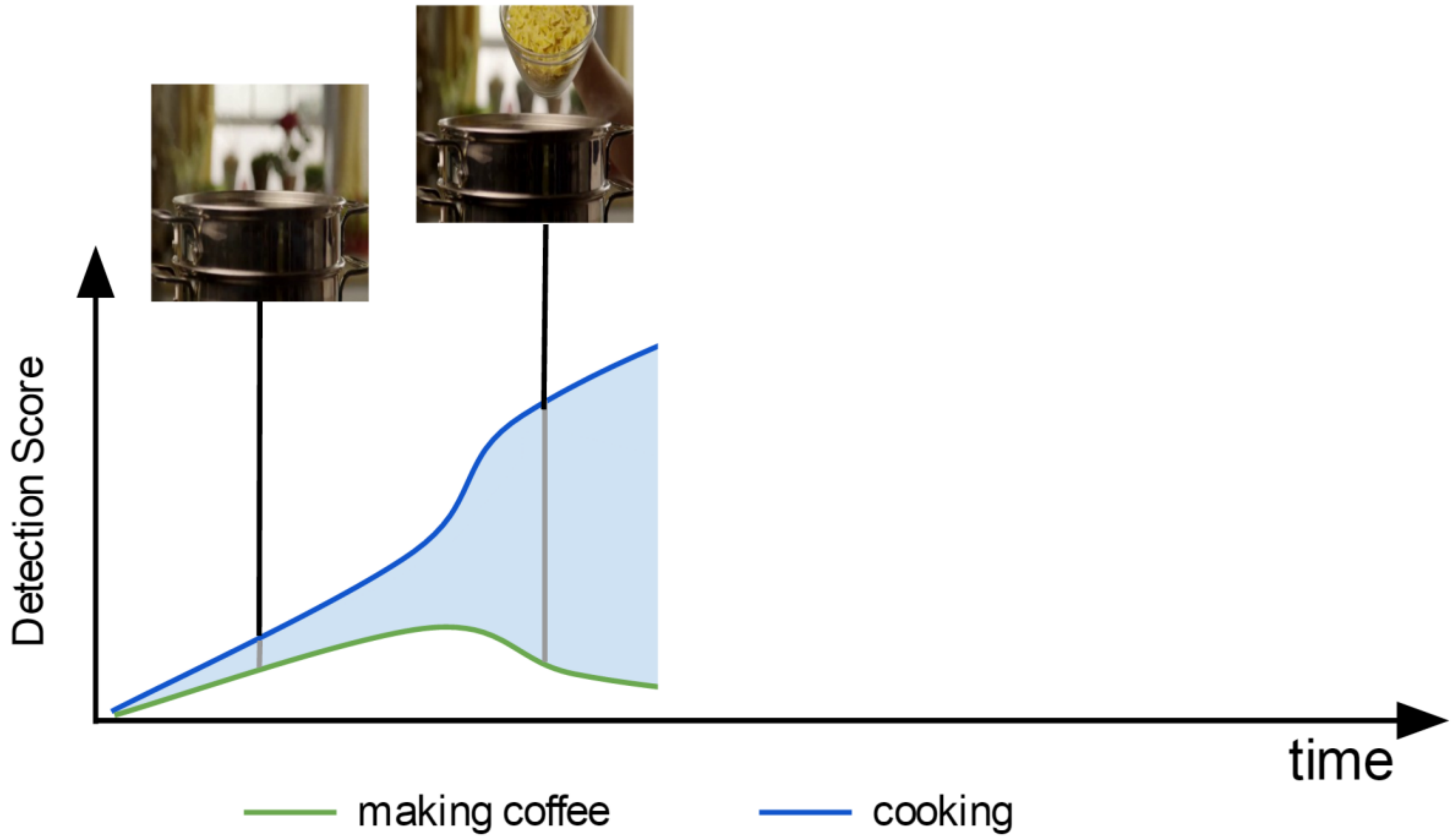
[Ma et al., 2014]

Applications: Activity Detection

As the detector sees more of an action, it should become more confident of

— Detecting the correct action class

— More confident that it is not the incorrect action class



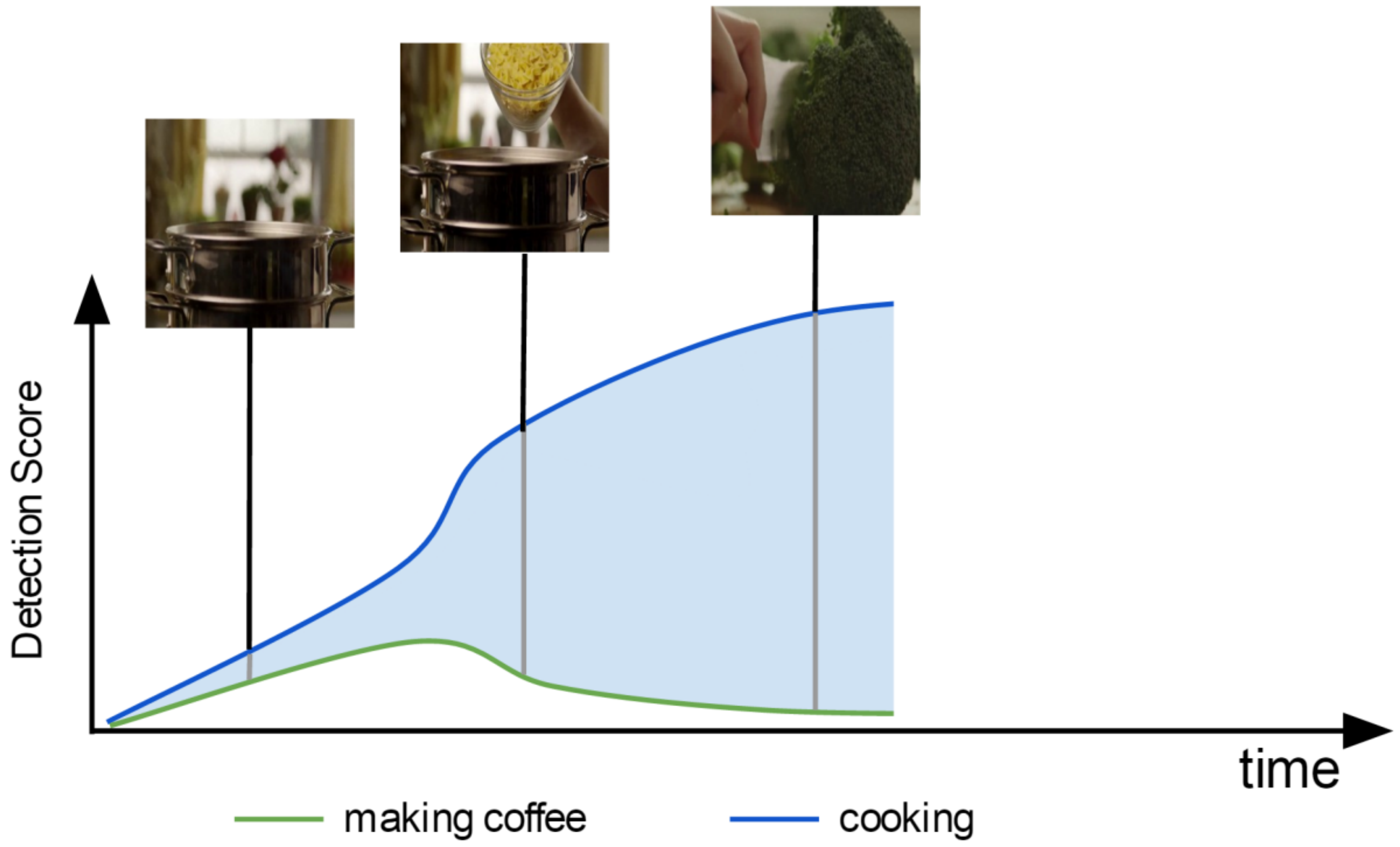
[Ma et al., 2014]

Applications: Activity Detection

As the detector sees more of an action, it should become more confident of

— Detecting the correct action class

— More confident that it is not the incorrect action class



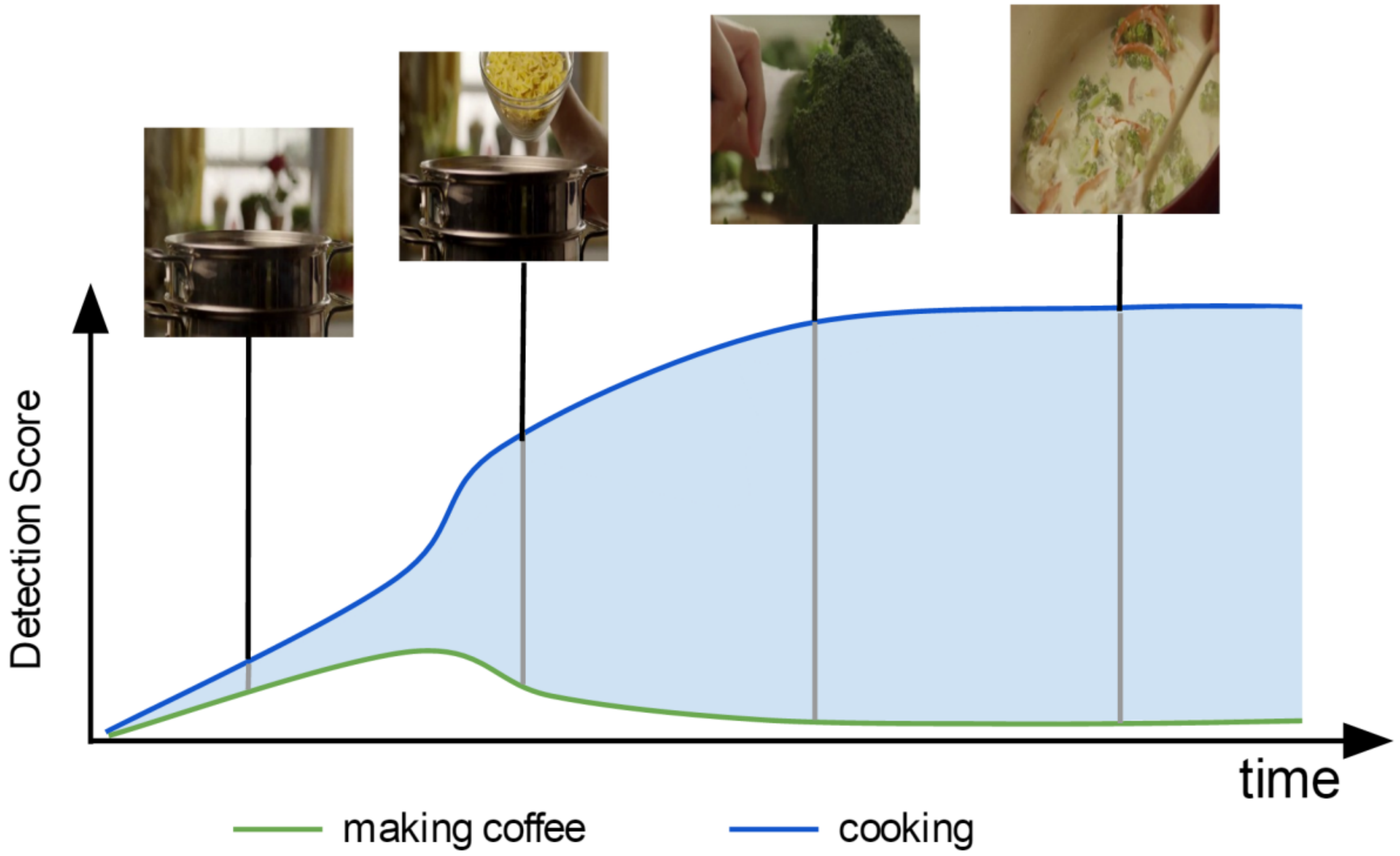
[Ma et al., 2014]

Applications: Activity Detection

As the detector sees more of an action, it should become more confident of

— Detecting the correct action class

— More confident that it is not the incorrect action class



[Ma et al., 2014]

New Class of Loss Functions

Classification loss at time t

Training loss at time t: $\mathcal{L}^t = \mathcal{L}_c^t + \lambda_r \mathcal{L}_r^t$

Ranking loss at time t

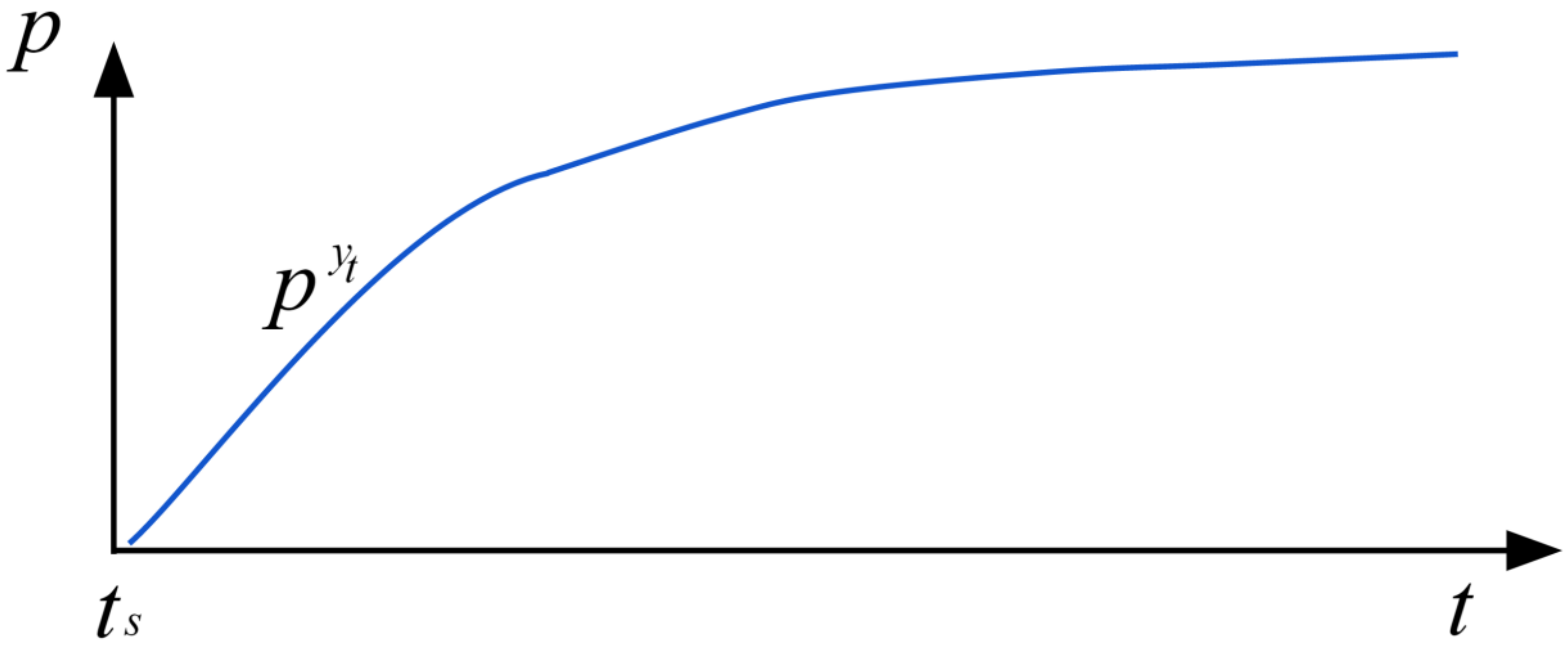
$\mathcal{L}_r^t$ is one of the following:

- $\mathcal{L}_s^t$ ranking loss on detection score
- $\mathcal{L}_m^t$ ranking loss on discriminative margin

[Ma et al., 2014]

Ranking Loss on Detection Score $\mathcal{L}_s^t$

Ideally what we want:

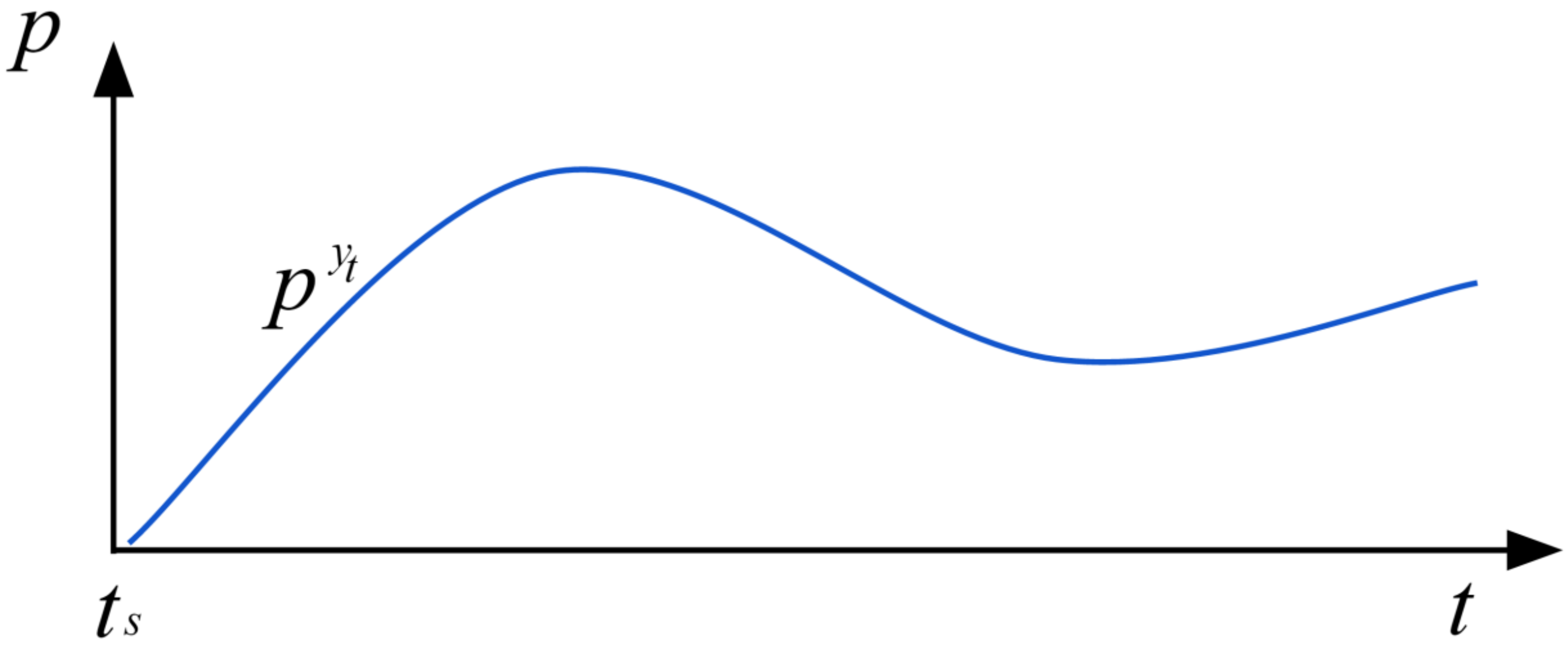


Prediction score of the ground truth action label

[Ma et al., 2014]

Ranking Loss on Detection Score $\mathcal{L}_s^t$

In **Practice:**



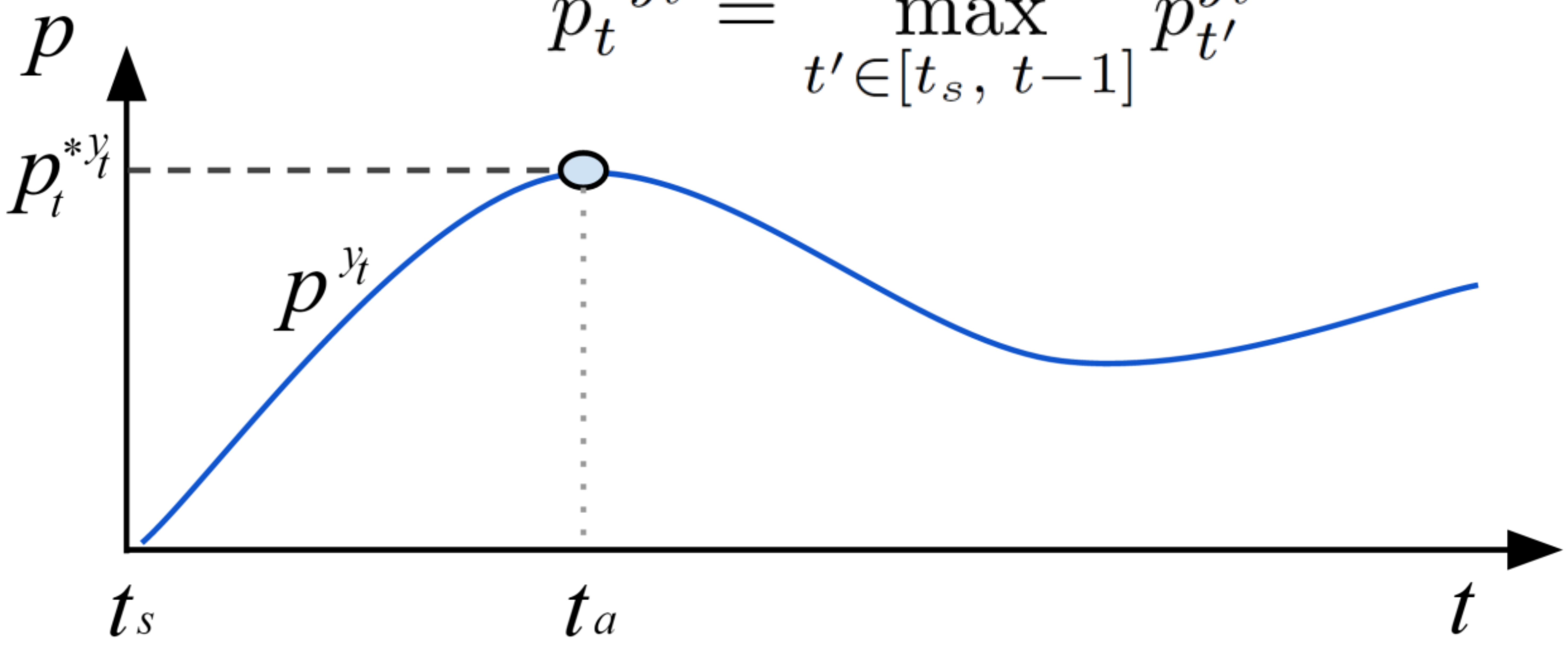
Prediction score of the ground truth action label

[Ma et al., 2014]

Ranking Loss on Detection Score $\mathcal{L}_s^t$

In **Practice:**

$$p_t^{*y_t} = \max_{t' \in [t_s,\, t-1]} p_{t'}^{y_t}$$

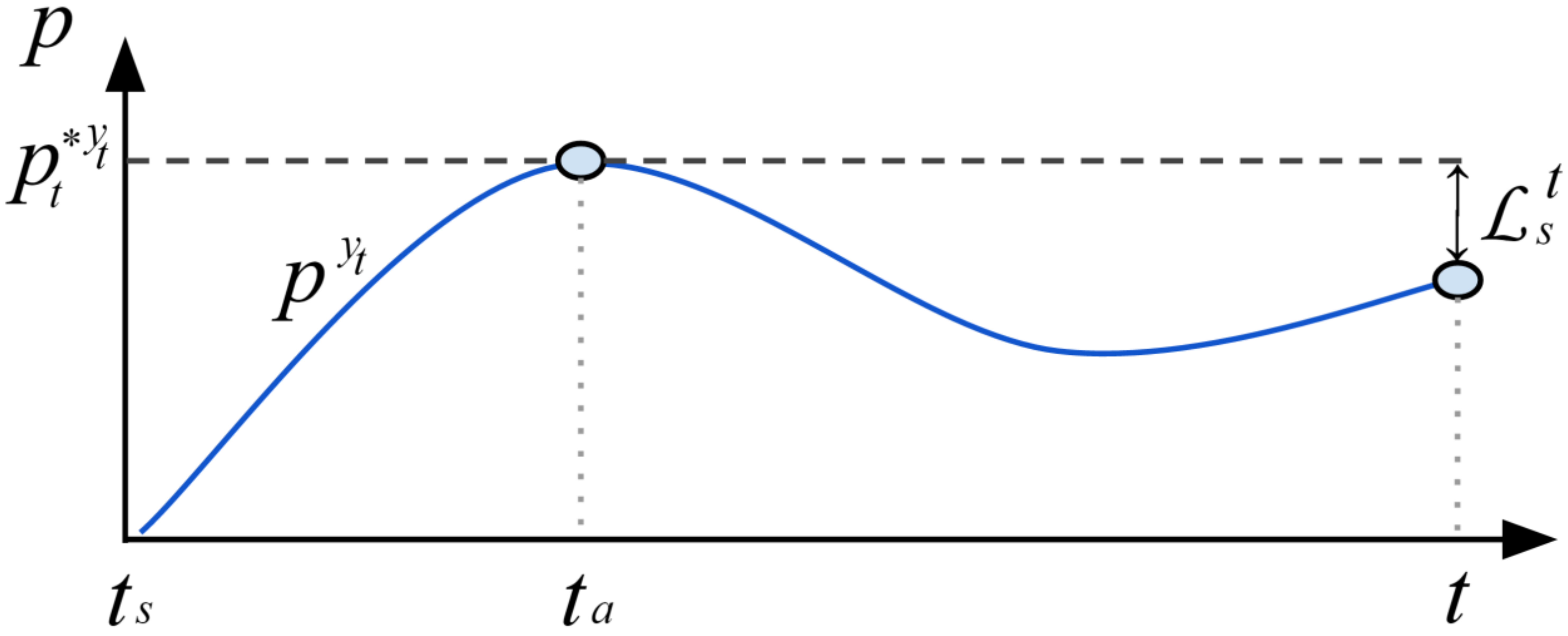


Prediction score of the ground truth action label

[Ma et al., 2014]

Ranking Loss on Detection Score $\mathcal{L}_s^t$

In **Practice:**

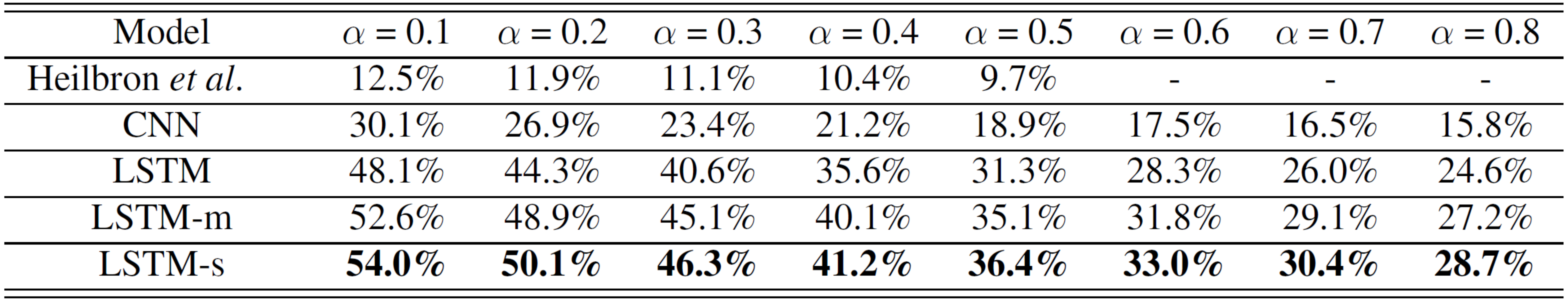


Prediction score of the ground truth action label

[Ma et al., 2014]

Applications: Activity Detection

Activity detection performance measured in mAP at different IOU thresholds



Model	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$
Heilbron *et al.*	12.5%	11.9%	11.1%	10.4%	9.7%	-	-	-
CNN	30.1%	26.9%	23.4%	21.2%	18.9%	17.5%	16.5%	15.8%
LSTM	48.1%	44.3%	40.6%	35.6%	31.3%	28.3%	26.0%	24.6%
LSTM-m	52.6%	48.9%	45.1%	40.1%	35.1%	31.8%	29.1%	27.2%
LSTM-s	**54.0%**	**50.1%**	**46.3%**	**41.2%**	**36.4%**	**33.0%**	**30.4%**	**28.7%**

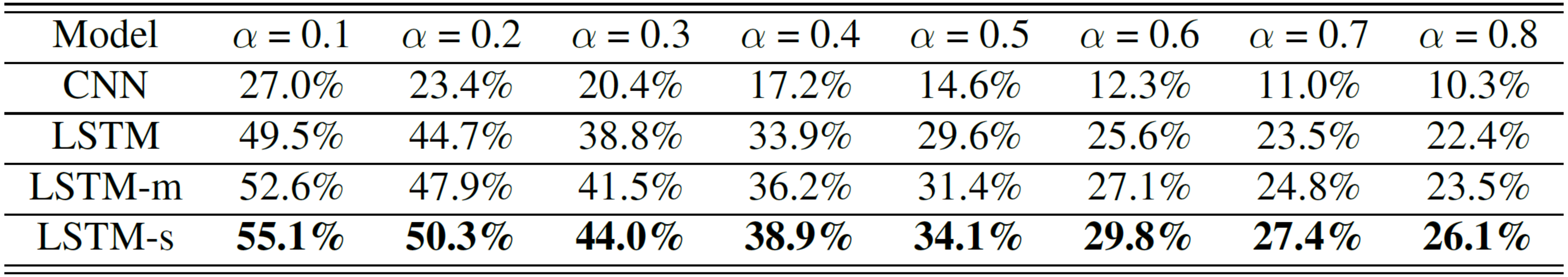
LSTM-m LSTM trained using both classification loss and rank loss on *discriminative margin*.

LSTM-s LSTM trained using both classification loss and rank loss on *detection score*.

[Ma et al., 2014]

Applications: Early Activity Detection

Activity early detection performance measured in mAP at different IOU thresholds



Model	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$
CNN	27.0%	23.4%	20.4%	17.2%	14.6%	12.3%	11.0%	10.3%
LSTM	49.5%	44.7%	38.8%	33.9%	29.6%	25.6%	23.5%	22.4%
LSTM-m	52.6%	47.9%	41.5%	36.2%	31.4%	27.1%	24.8%	23.5%
LSTM-s	**55.1%**	**50.3%**	**44.0%**	**38.9%**	**34.1%**	**29.8%**	**27.4%**	**26.1%**

Note: first 3/10 of activity is seen by a detector

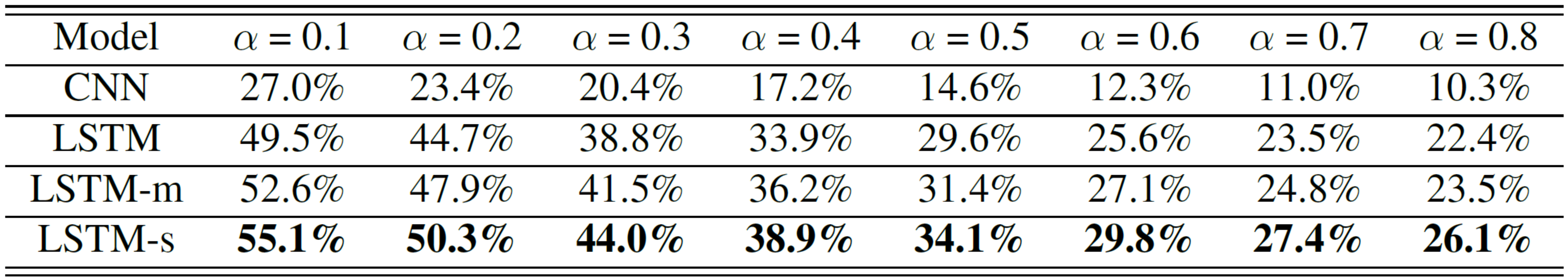
LSTM-m LSTM trained using both classification loss and rank loss on *discriminative margin*.

LSTM-s LSTM trained using both classification loss and rank loss on *detection score*.

[Ma et al., 2014]

Applications: Early Activity Detection

Activity early detection performance measured in mAP at different IOU thresholds



Model	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$
CNN	27.0%	23.4%	20.4%	17.2%	14.6%	12.3%	11.0%	10.3%
LSTM	49.5%	44.7%	38.8%	33.9%	29.6%	25.6%	23.5%	22.4%
LSTM-m	52.6%	47.9%	41.5%	36.2%	31.4%	27.1%	24.8%	23.5%
LSTM-s	**55.1%**	**50.3%**	**44.0%**	**38.9%**	**34.1%**	**29.8%**	**27.4%**	**26.1%**

Note: first 3/10 of activity is seen by a detector

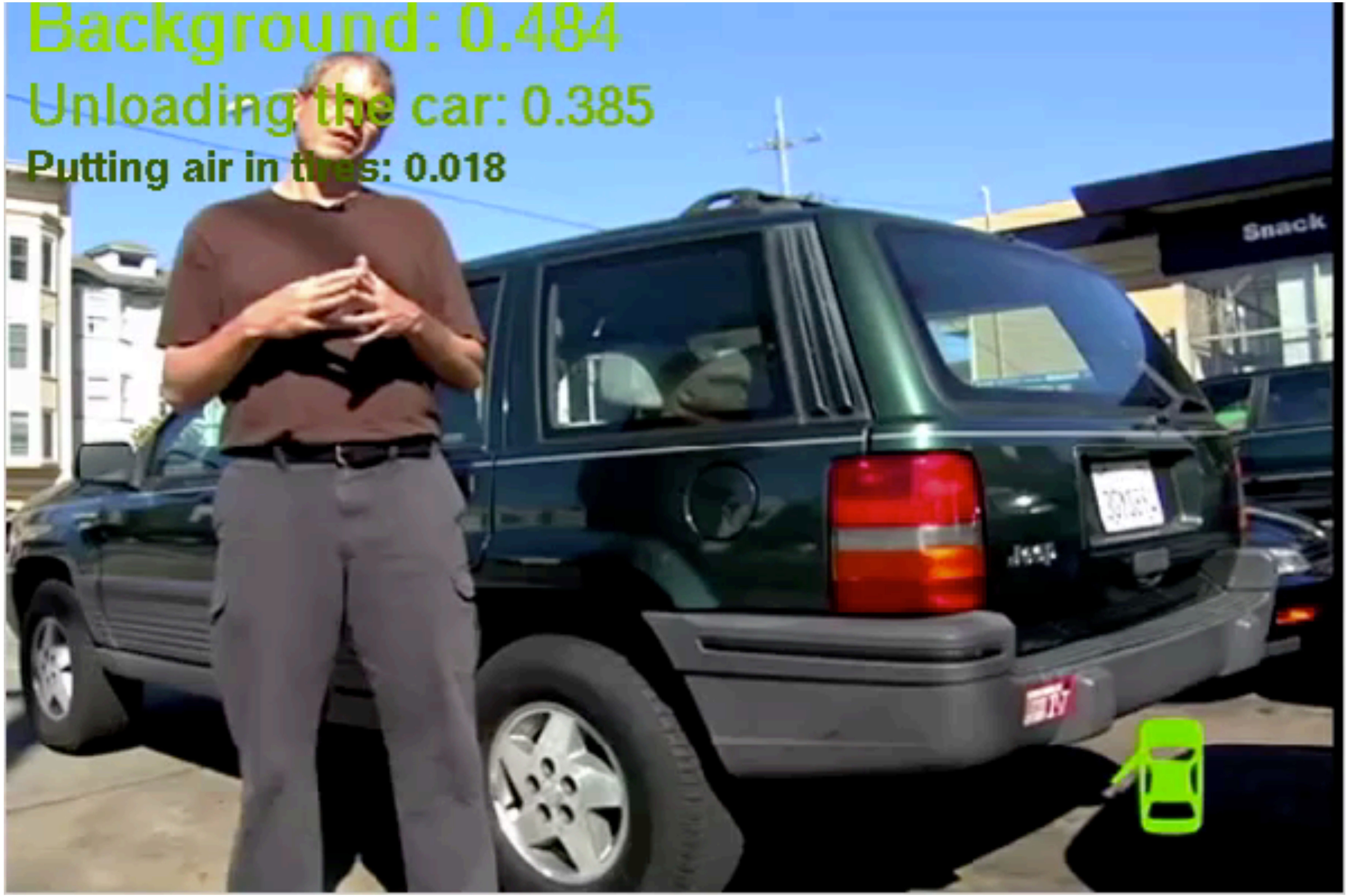
LSTM-m LSTM trained using both classification loss and rank loss on *discriminative margin*.

LSTM-s LSTM trained using both classification loss and rank loss on *detection score*.

Take home: Early detection is only 1-3% worse than sewing the whole sequence

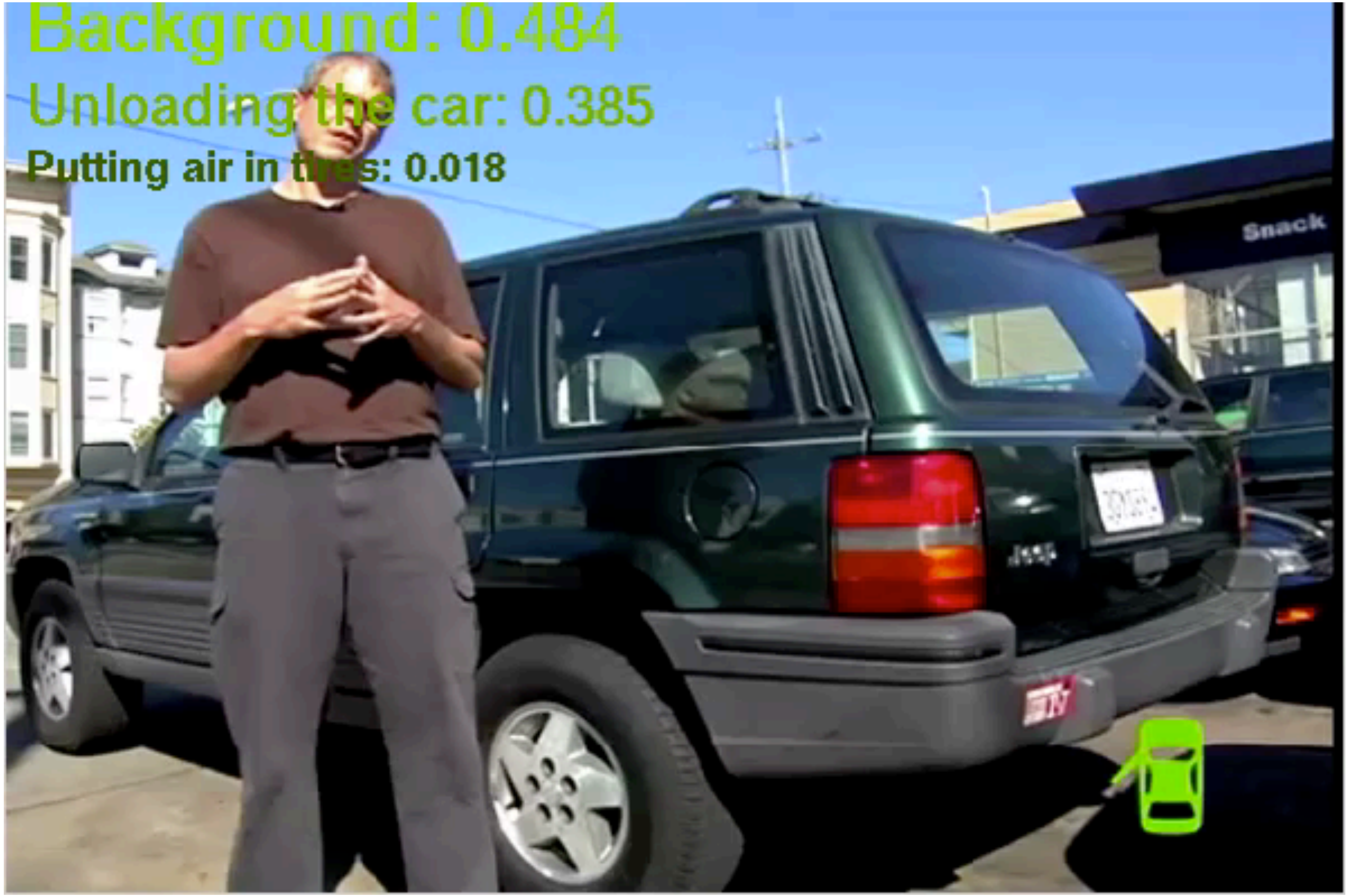
[Ma et al., 2014]

Applications: Activity Detection



[Ma et al., 2014]

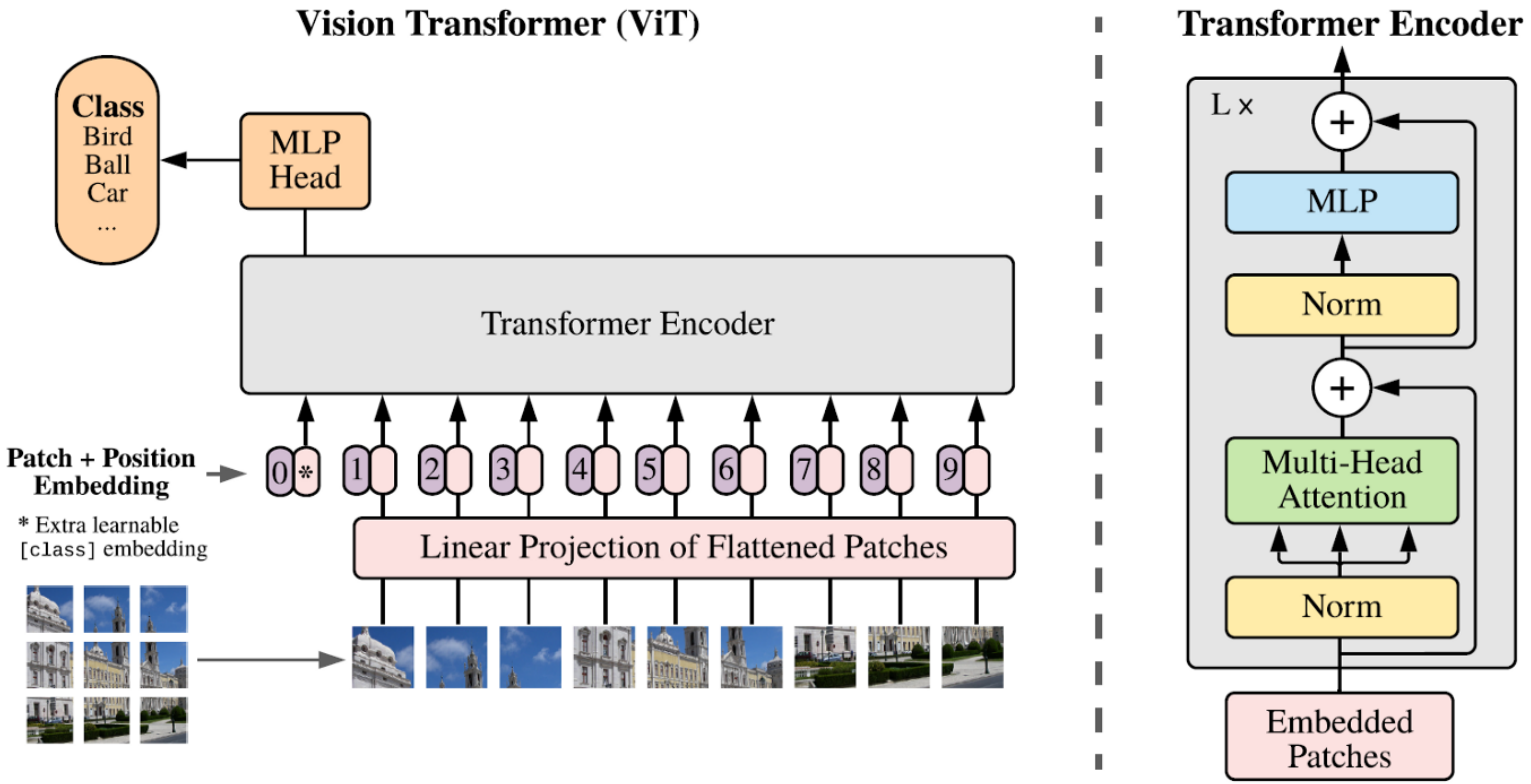
Applications: Activity Detection



[Ma et al., 2014]

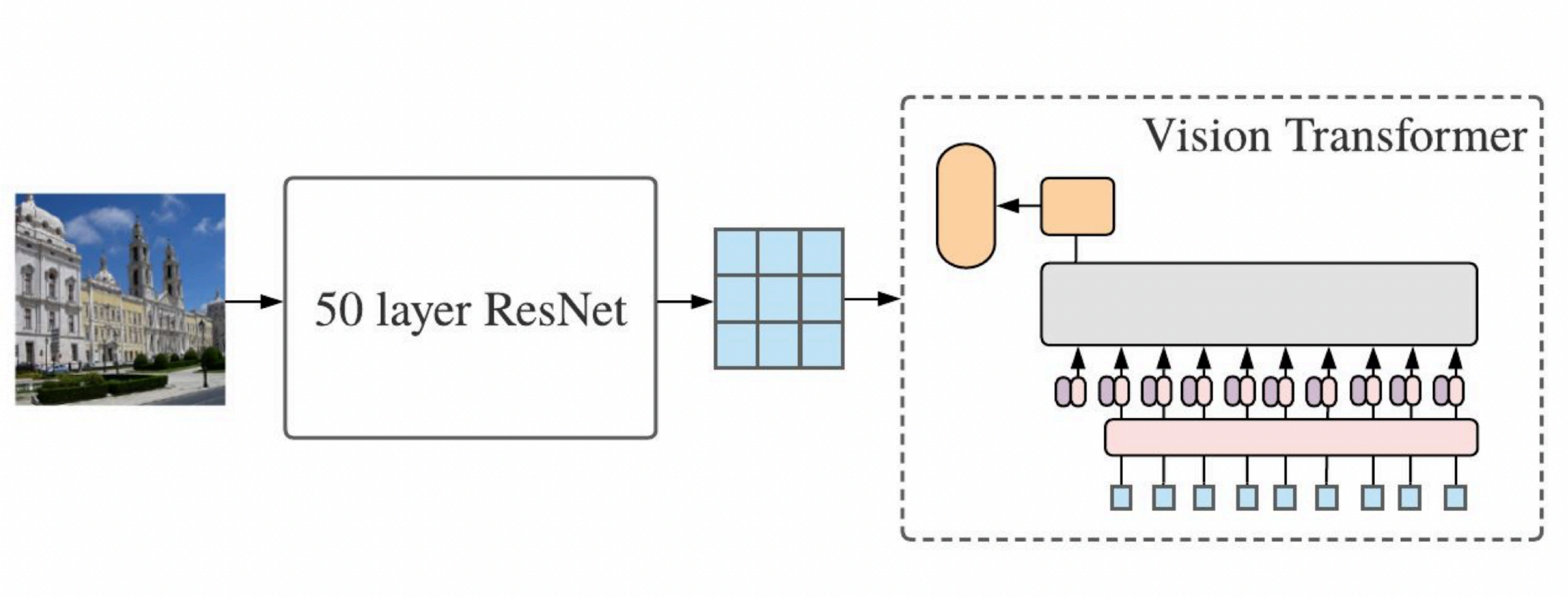
Vision Transformer

[Dosovitskiy et al., 2020]



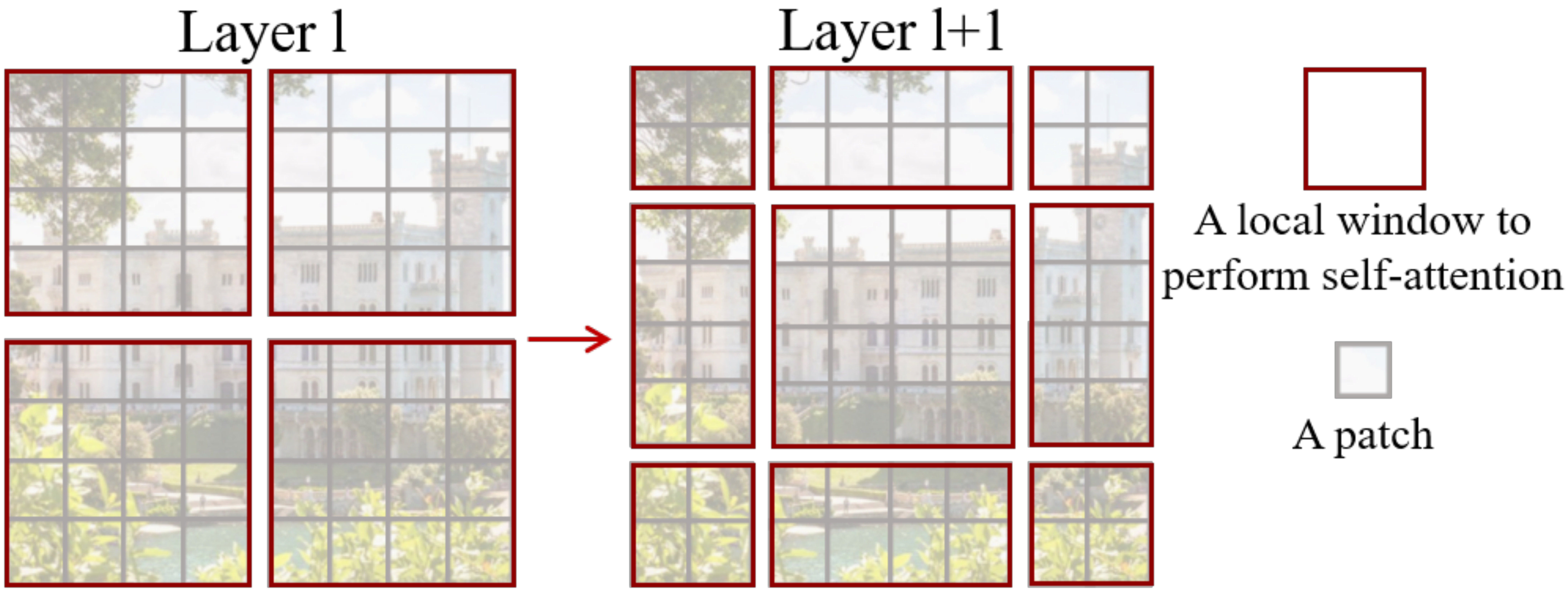
Vision Transformer

[Dosovitskiy et al., 2020]



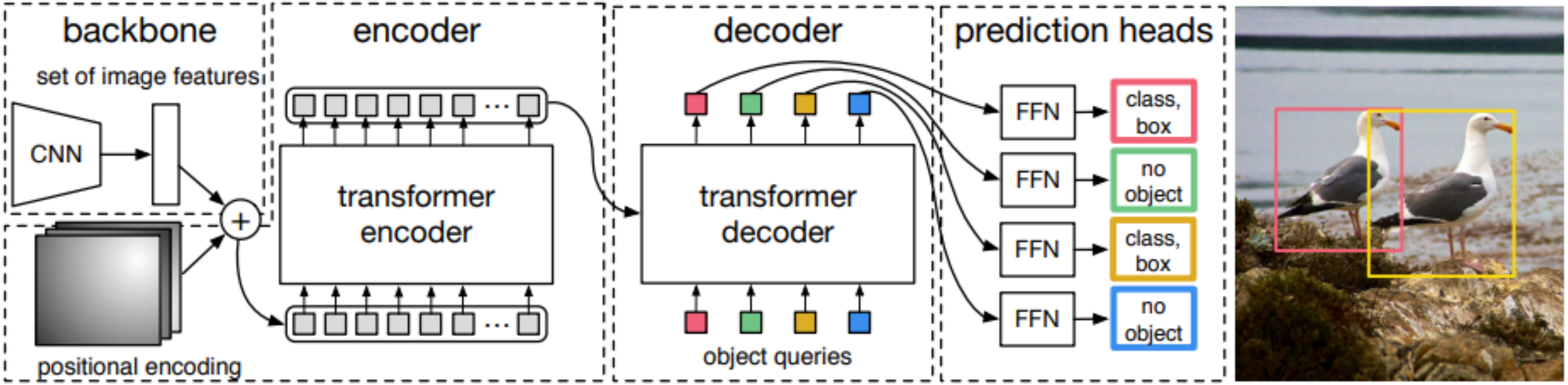
Swin Transformers

[Liu et al., 2021]



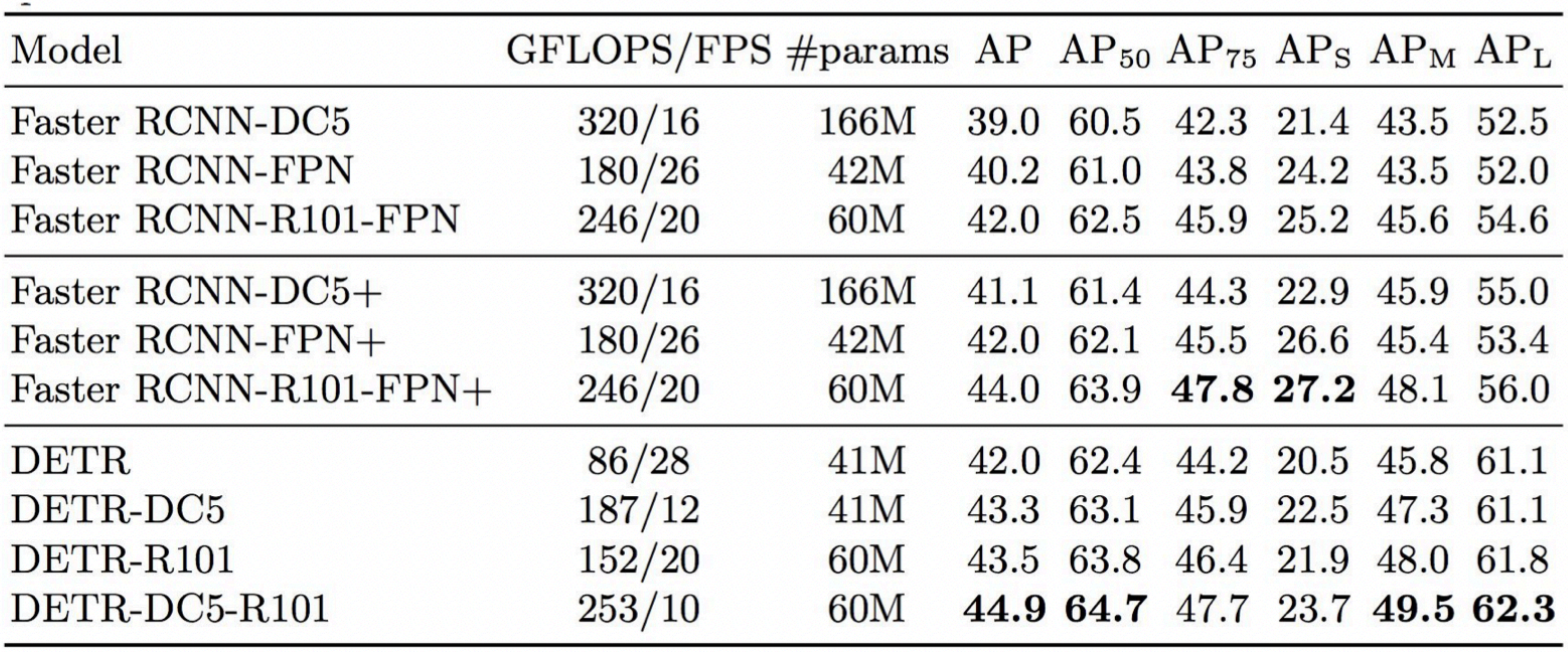
DEtection TRansformer (DETR)

[Carion et al., 2020]



DEtection **TR**ansformer (DETR)

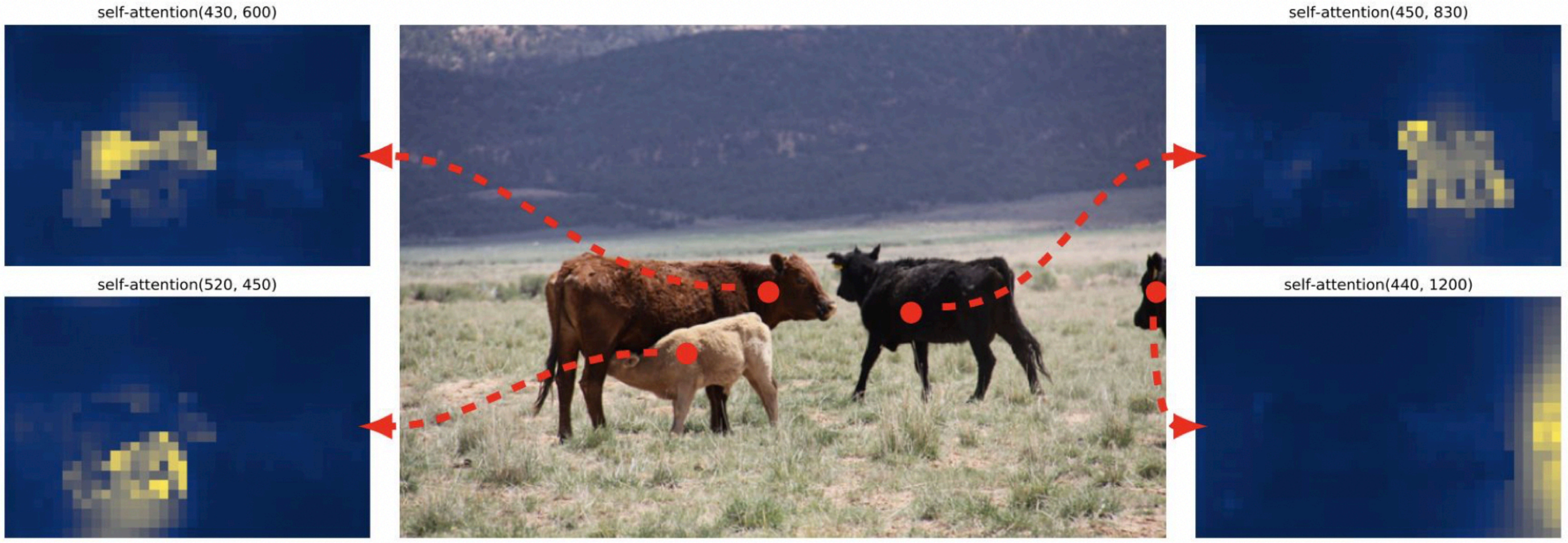
[Carion et al., 2020]



Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_{S}	AP_{M}	AP_{L}
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

DEtection TRansformer (DETR)

[Carion et al., 2020]



DEtection **TR**ansformer (DETR)

[Carion et al., 2020]

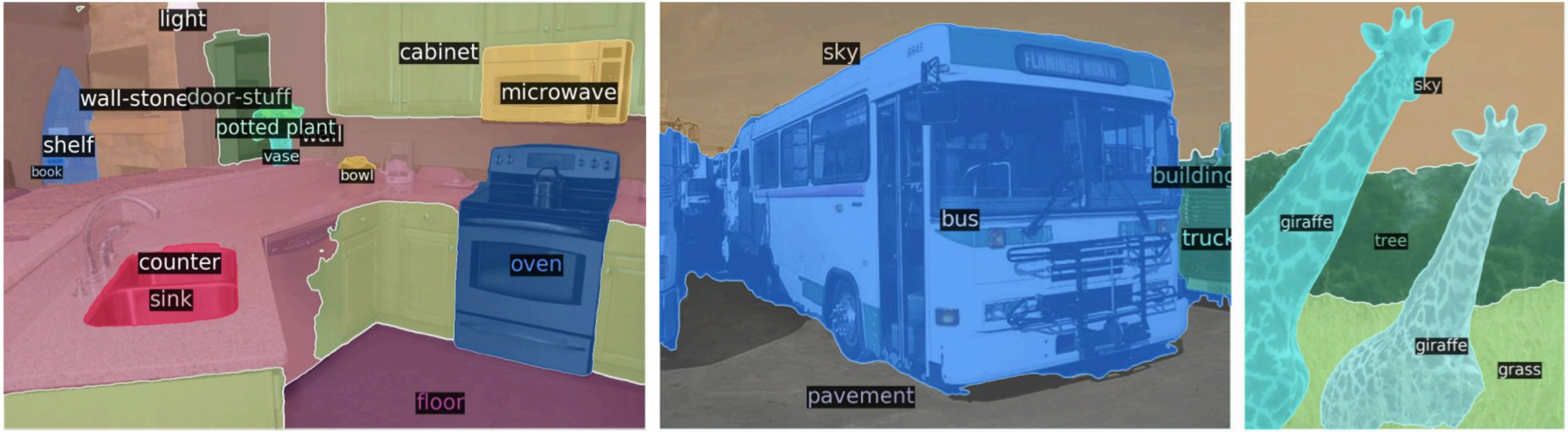
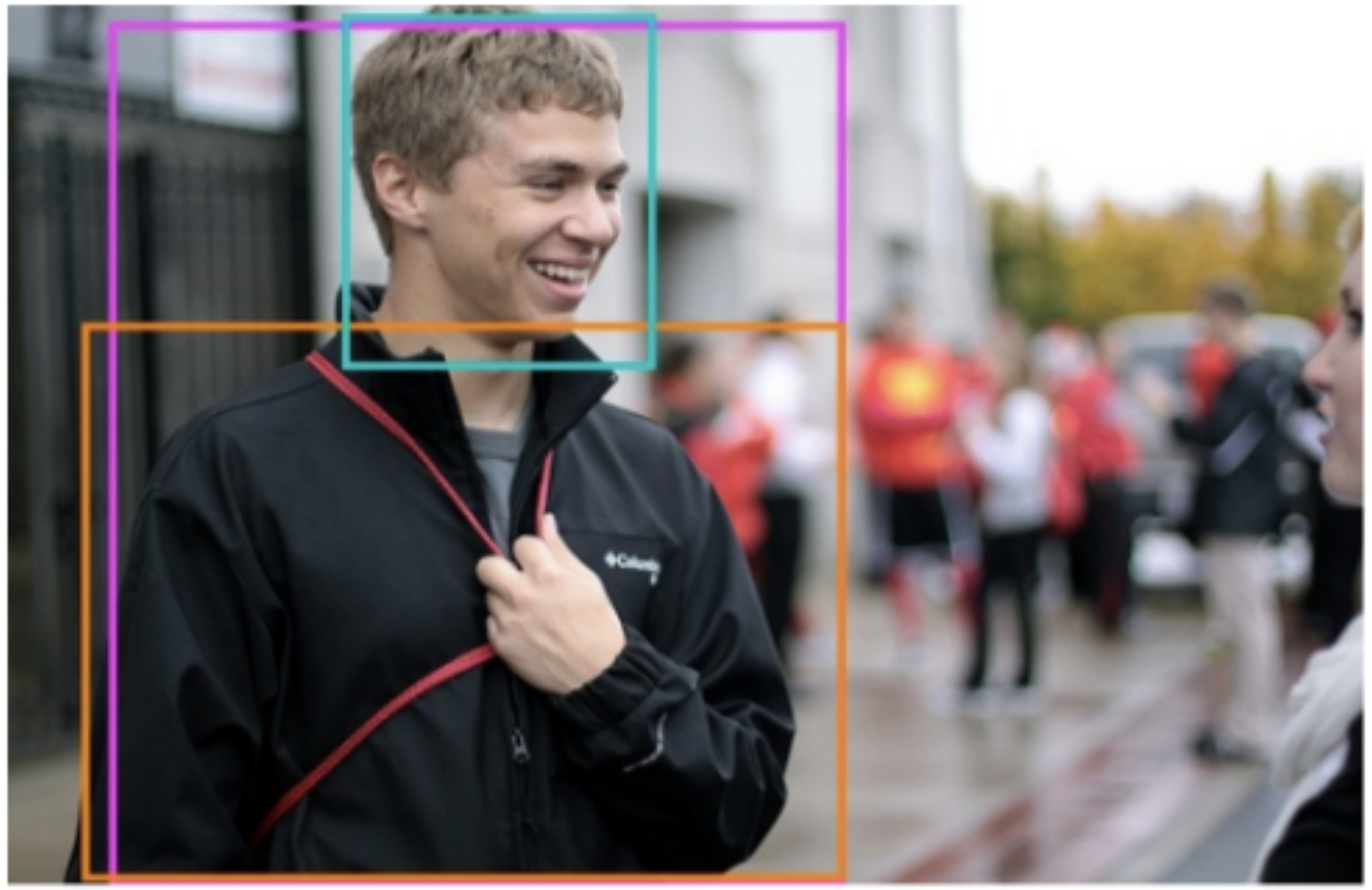


Image Grounding: Beyond Object Detection

[Li et al., 2021]

Given the **image** and one or more **natural language phrases**, locate regions that correspond to those phrases.

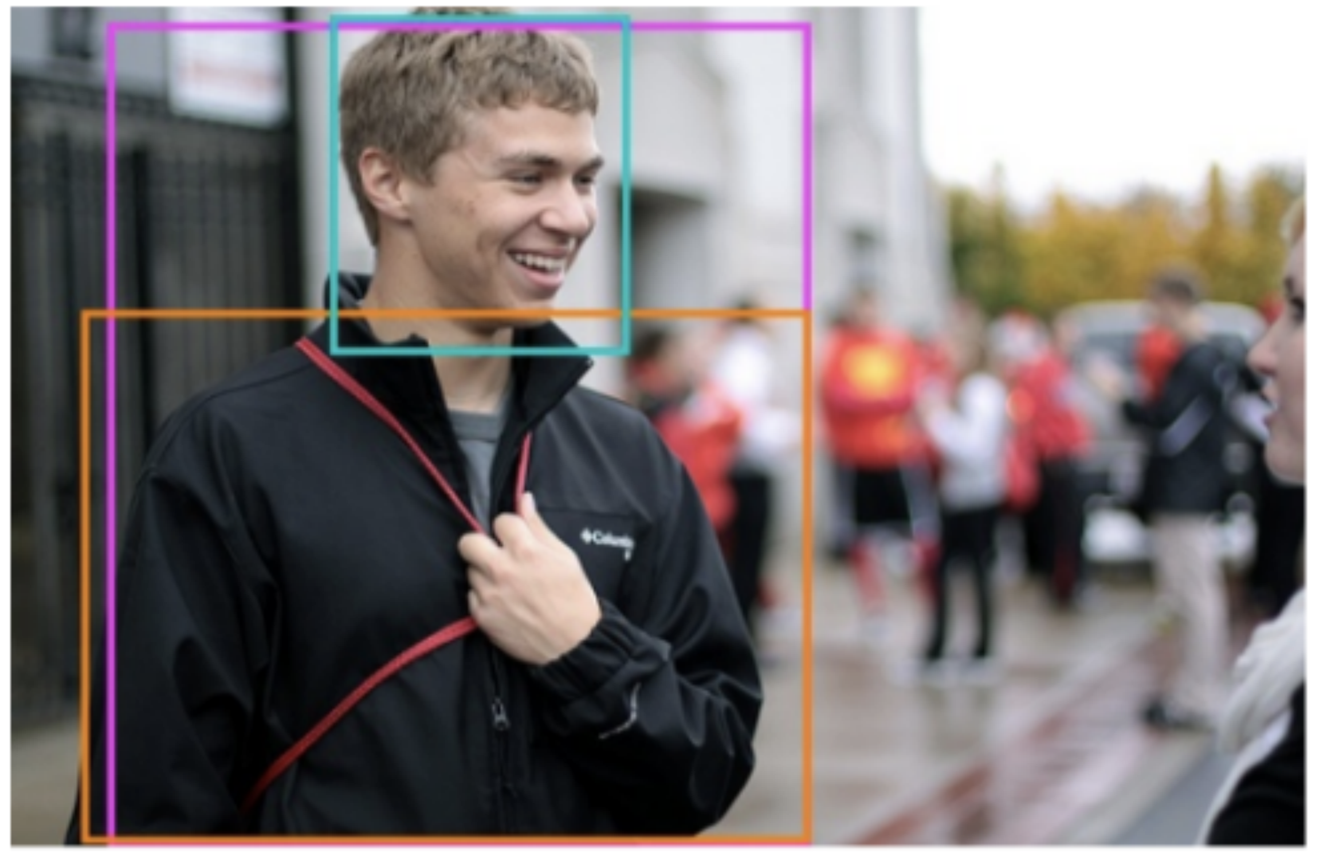


A man wearing a black-jacket has
a smile on his face.

Image Grounding: Beyond Object Detection

[Li et al., 2021]

Given the **image** and one or more **natural language phrases**, locate regions that correspond to those phrases.



A man wearing a black-jacket has a smile on his face.

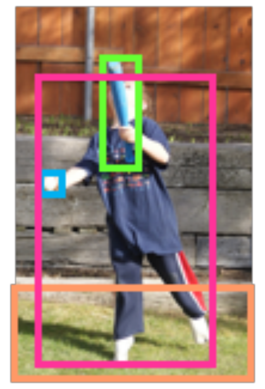
Fundamental task for **image / video understanding**

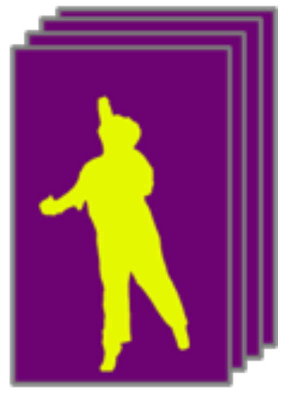
— Helps improve performance on other tasks (e.g., image captioning, VQA)

Approach

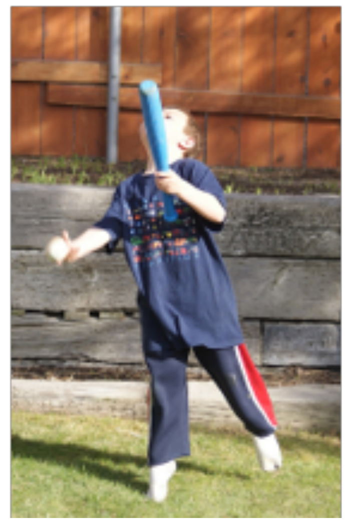
[Li et al., 2021]

Output:





Input:



A small boy playing in the grass with a blue bat and a ball

a small boy

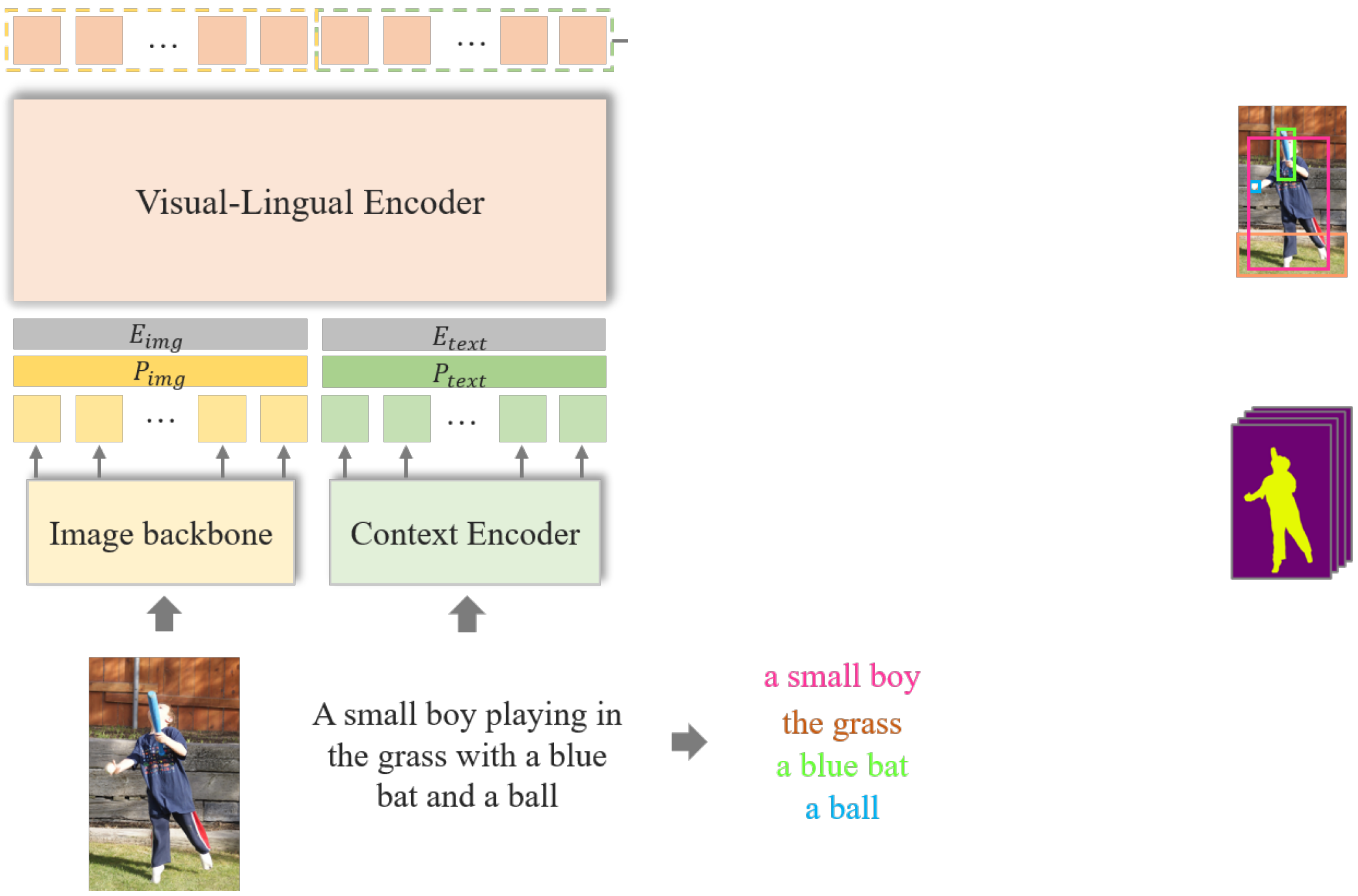
the grass

a blue bat

a ball

Approach

[Li et al., 2021]



A small boy playing in the grass with a blue bat and a ball

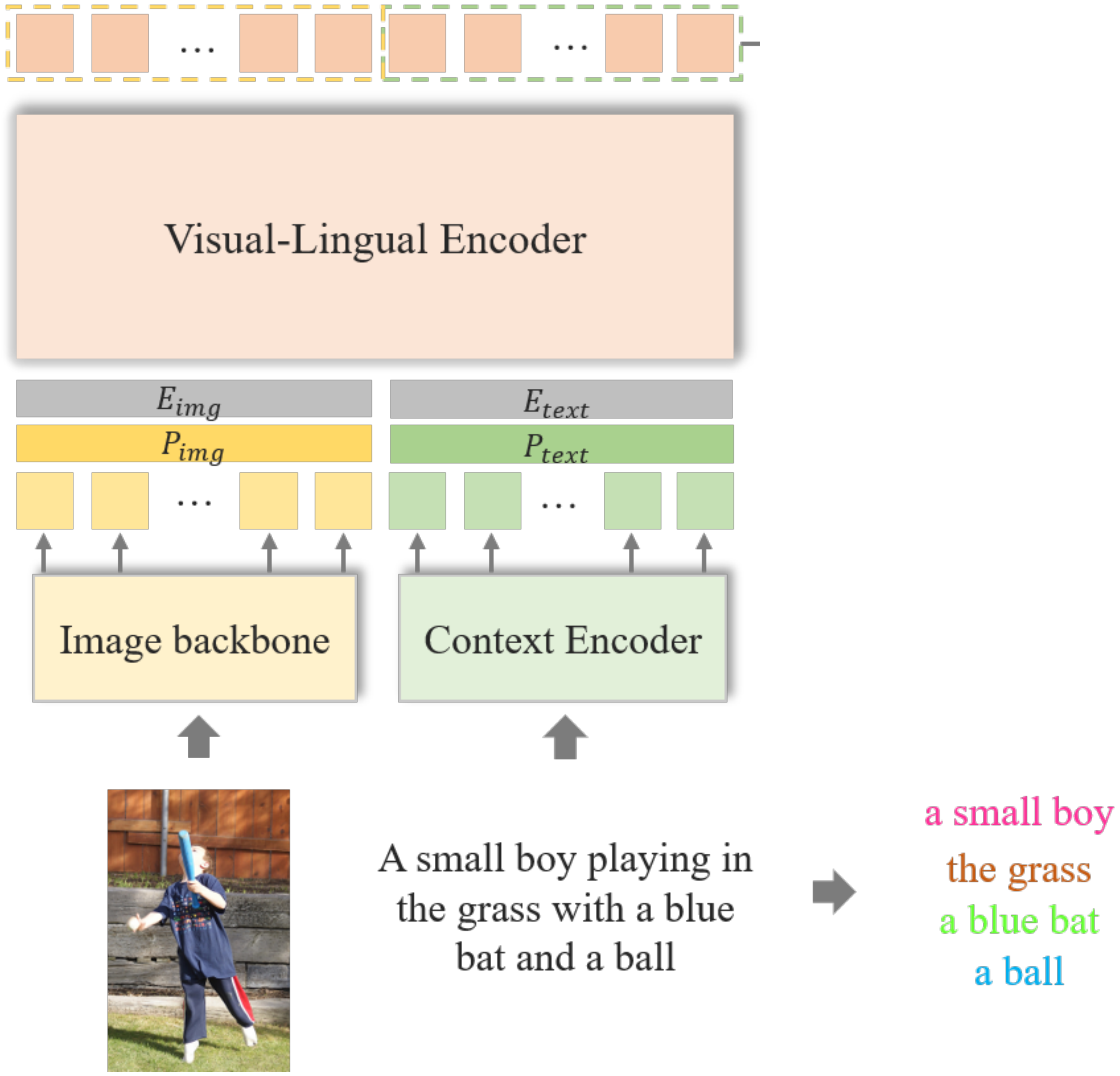
Features from different modalities are first extracted by corresponding backbone and then fused in the Visual-Lingual Encoder

Approach

[Li et al., 2021]

Modality Label

Positional Embedding



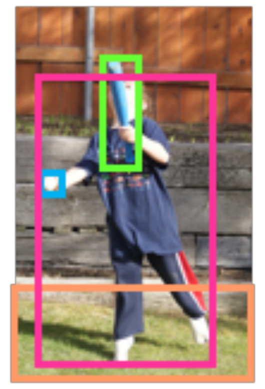
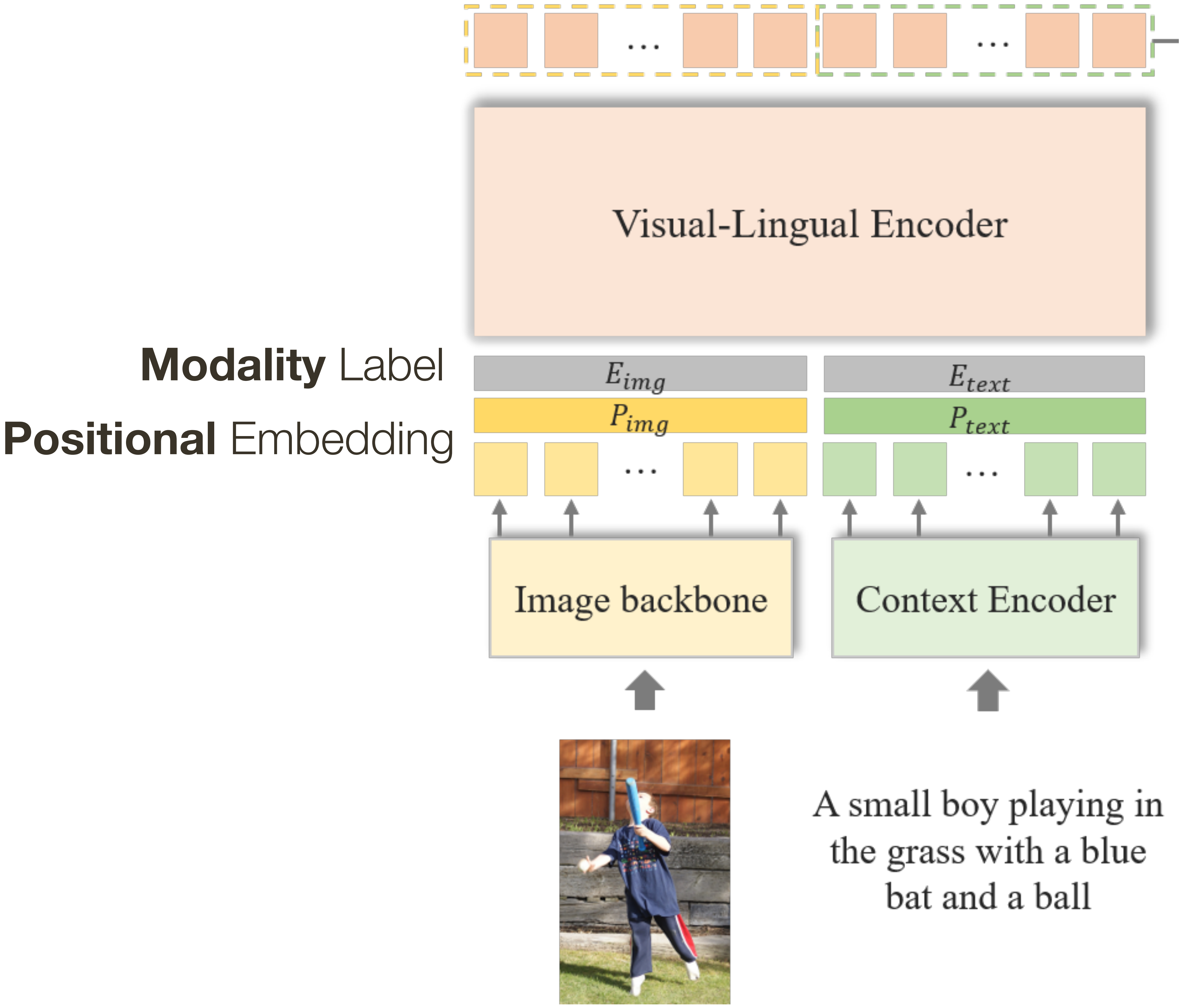


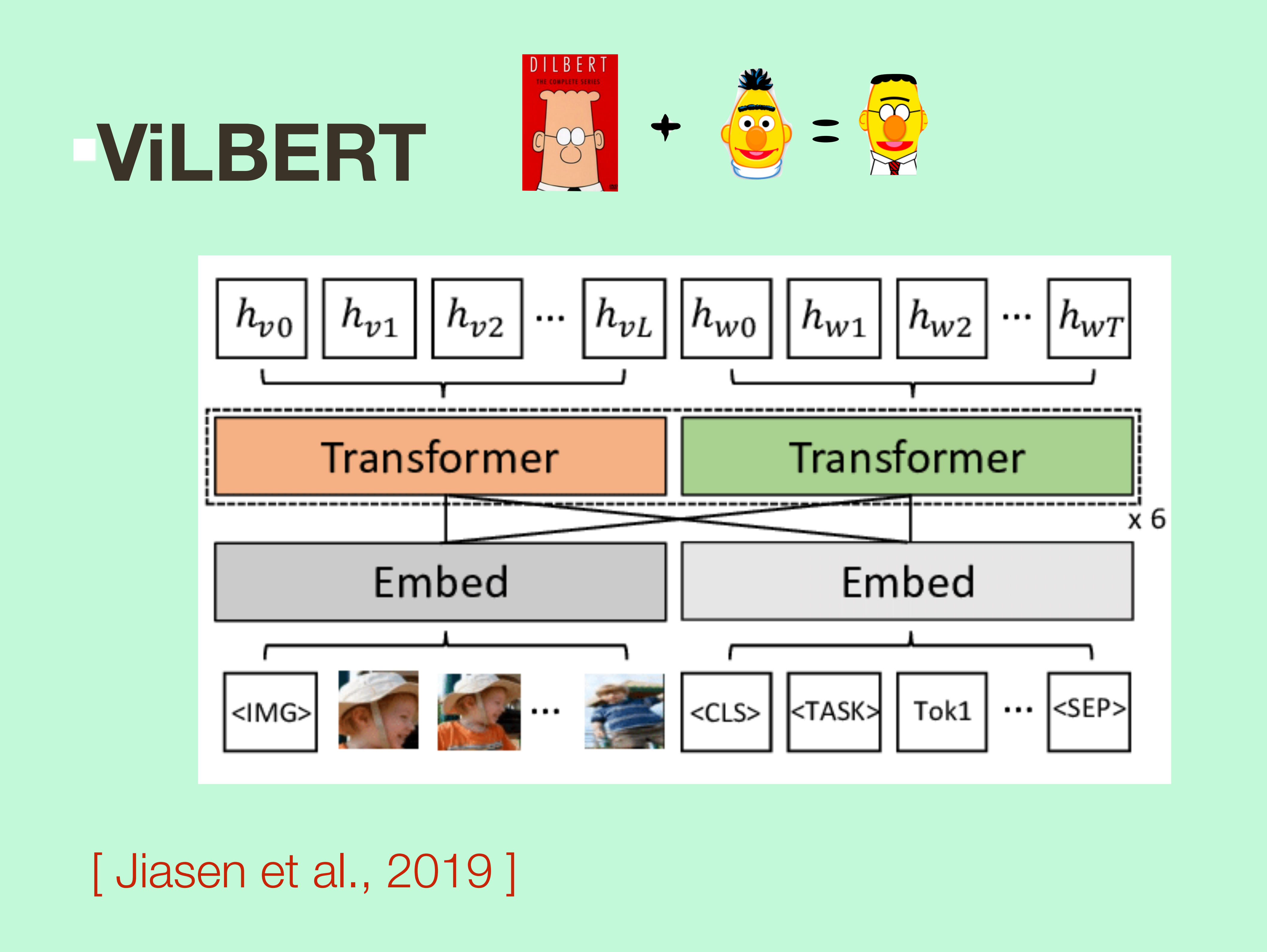
Image Backbone: ResNet (e.g., 16x16 -> 256 visual tokens)

Context Encoder: Pretrained Bert

Approach

[Li et al., 2021]




ViLBERT
DILBERT
h_{v0} h_{v1} h_{v2} ⋯ h_{vL} h_{w0} h_{w1} h_{w2} ⋯ h_{wT}
Transformer
Transformer
x 6
Embed
Embed
<IMG>
<CLS> <TASK> Tok1 ⋯ <SEP>

[Jiasen et al., 2019]

Image Backbone: ResNet (e.g., 16x16 -> 256 visual tokens)

Context Encoder: Pretrained Bert

Approach

[Li et al., 2021]

Modality Label

Positional Embedding

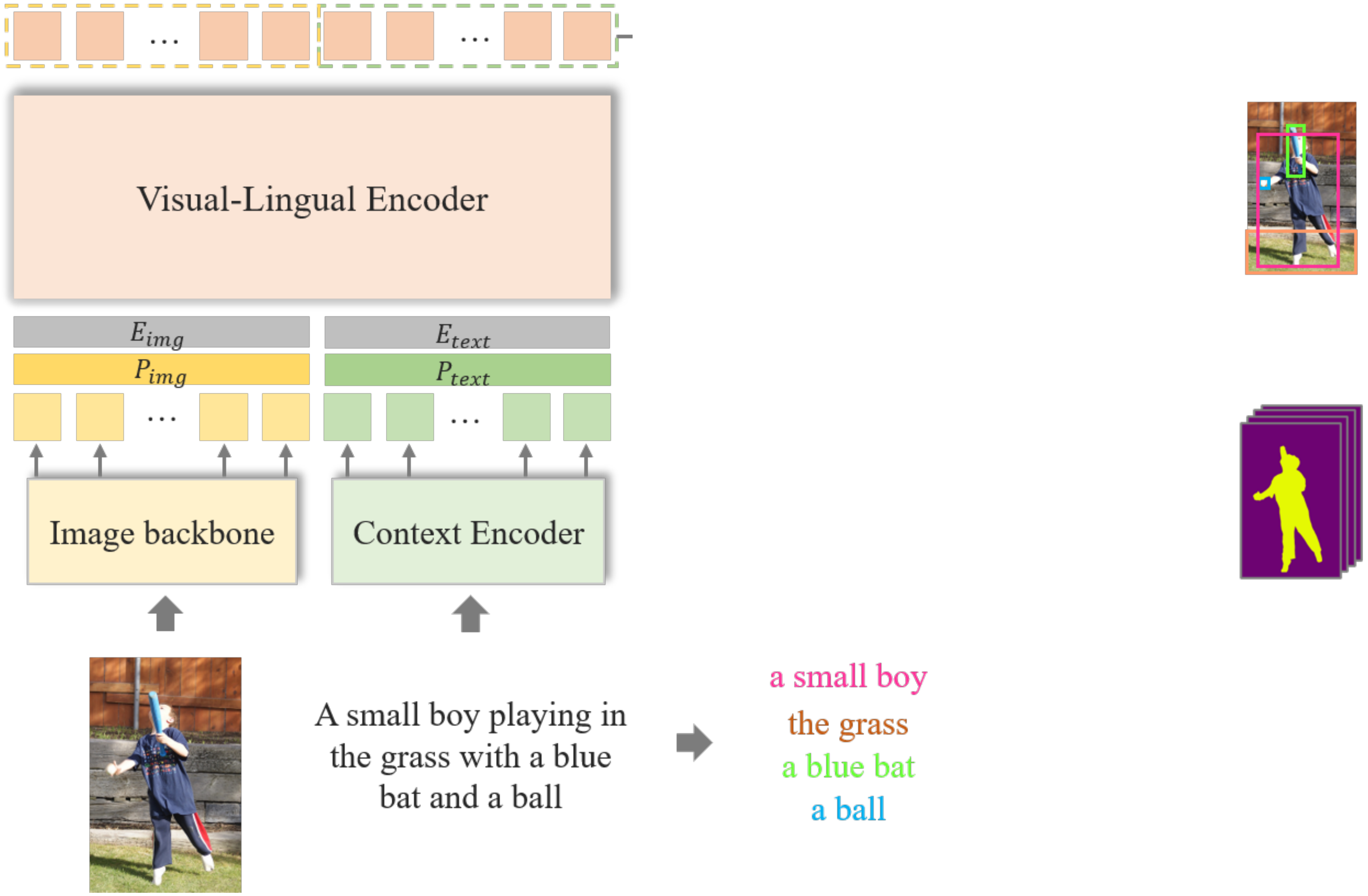
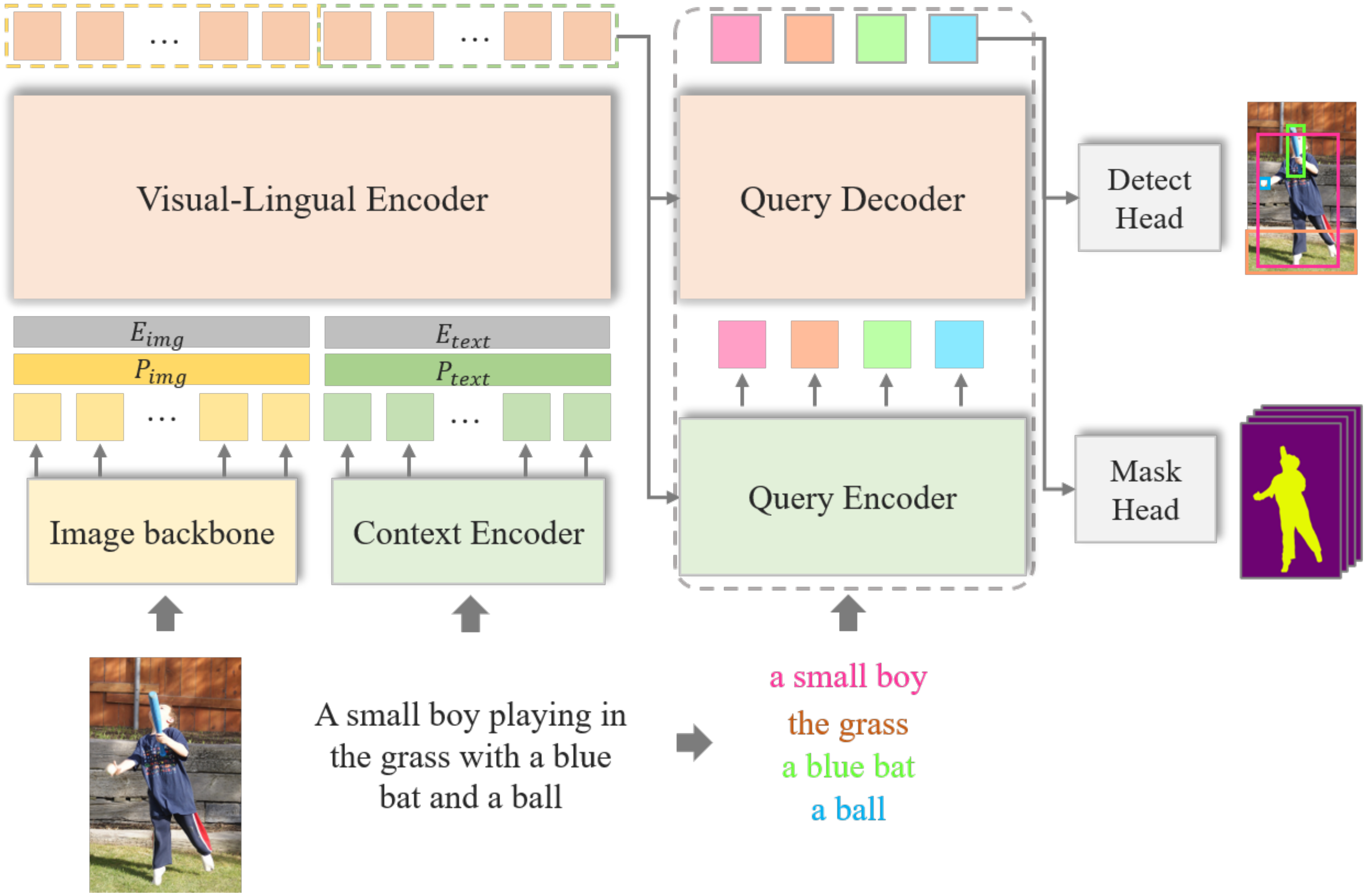


Image Backbone: ResNet (e.g., 16x16 -> 256 visual tokens)

Context Encoder: Pretrained Bert

Approach

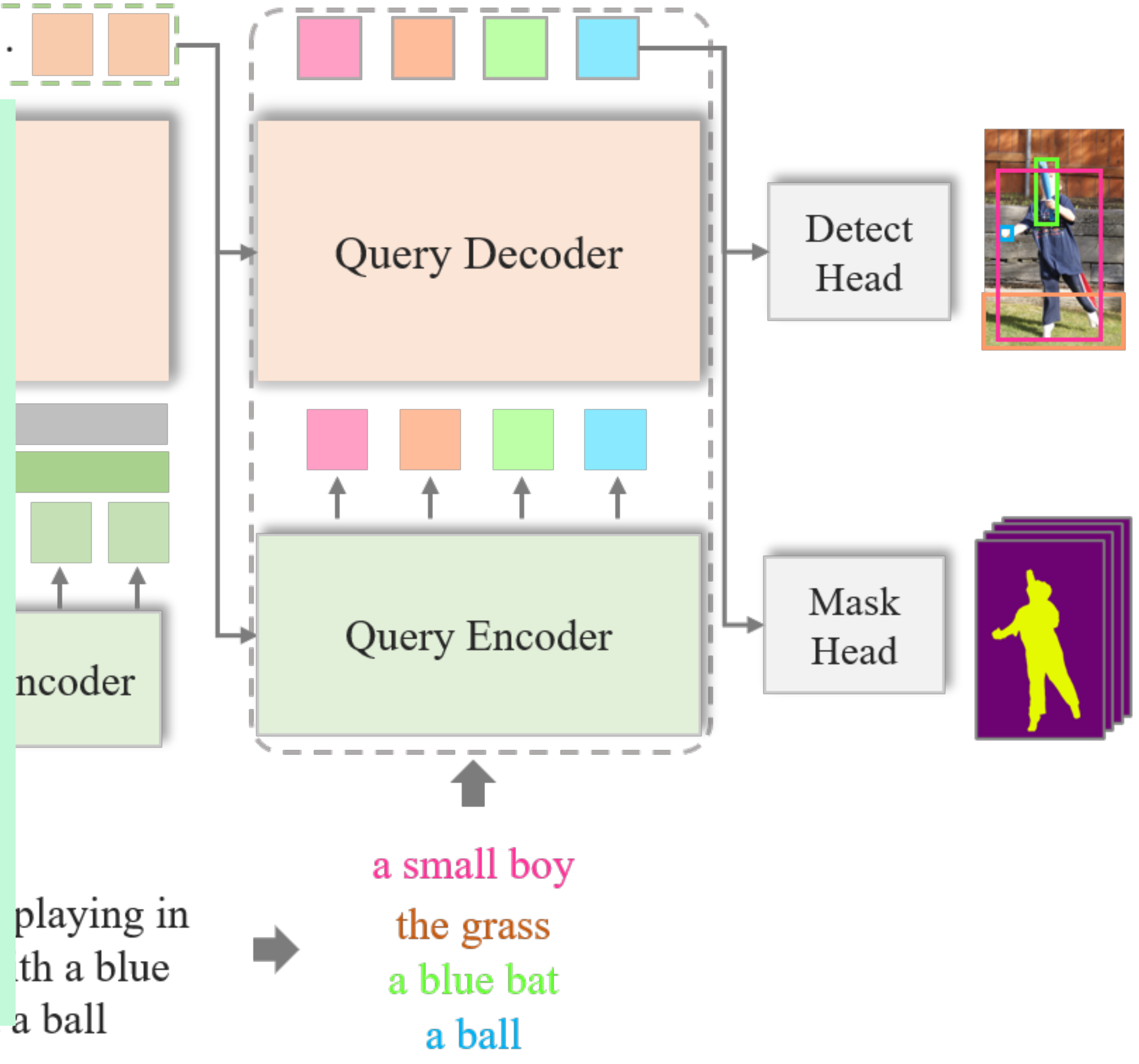
[Li et al., 2021]

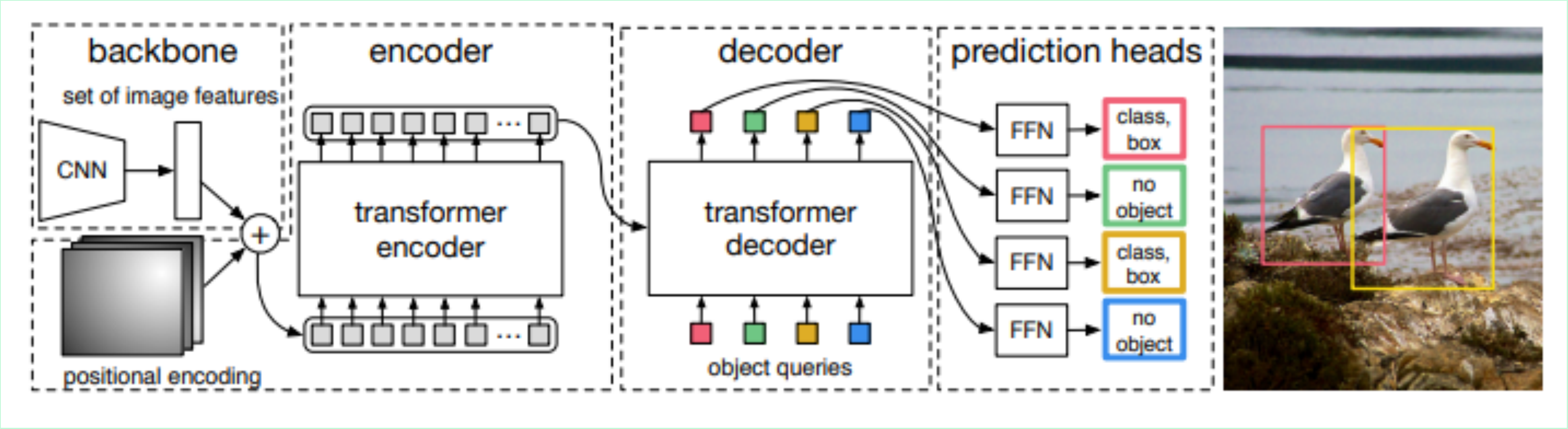


Query Encoder & **Decoder** are designed to encode phrase expression queries and decode them given corresponding multi-modal feature

Approach

[Li et al., 2021]





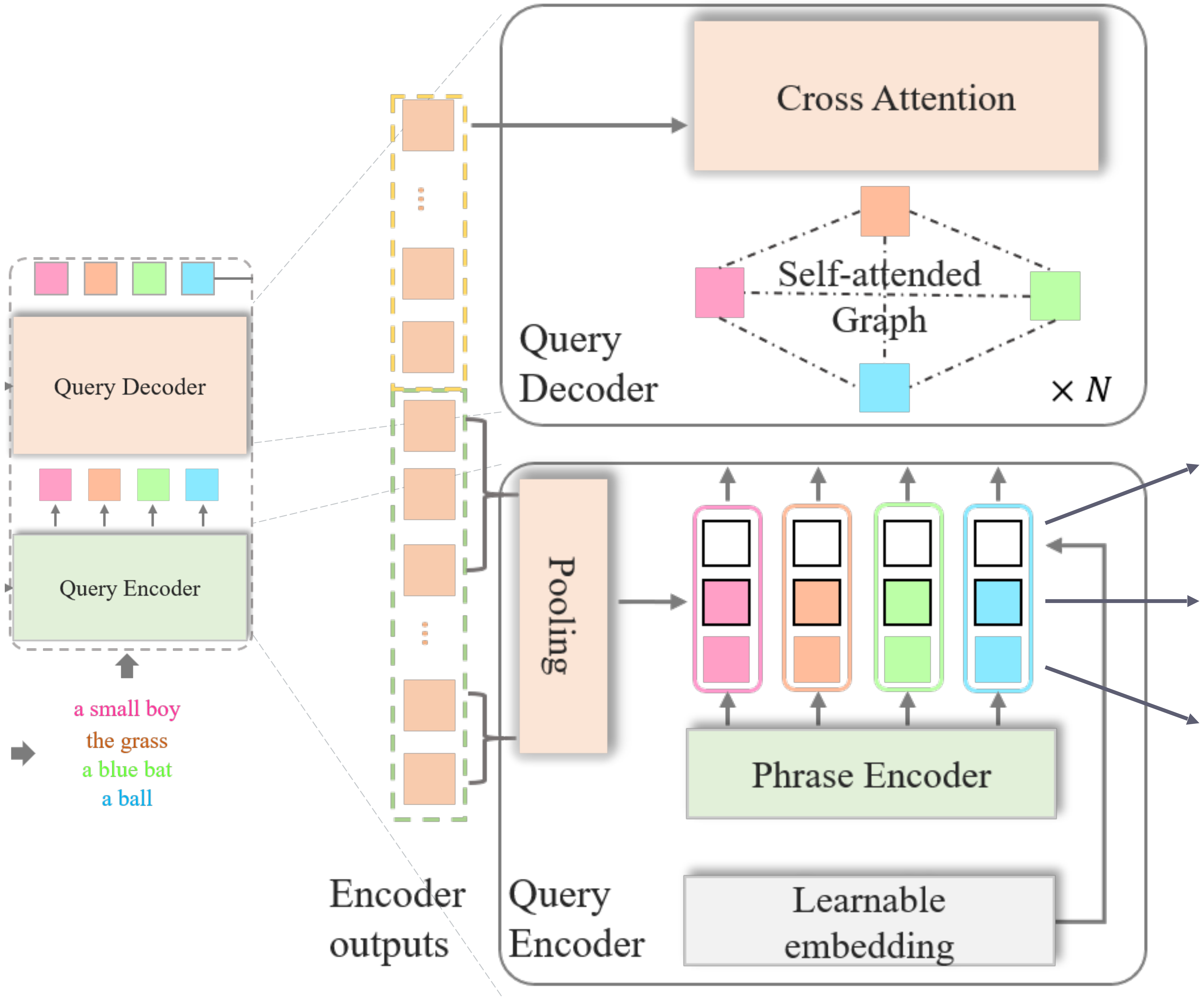
End-to-End Object Detection with Transformers **(DETR)**

[Carion et al., ECCV 2020]

Query Encoder & **Decoder** are designed to encode phrase expression queries and decode them given corresponding multi-modal feature

Query Encoder & Decoder

[Li et al., 2021]



$$\widehat{Q_{\mathbf{p}_i}} = \texttt{MLP}\left([\mathbf{f}_c(\mathbf{p}_i); \mathbf{f}_{\mathbf{p}_i}]\right) + E_p$$

$$\mathbf{f}_c(\mathbf{p}_i) = \frac{\sum \mathbf{f}_{vl}[l_{\mathbf{p}_i} : r_{\mathbf{p}_i}]}{r_{\mathbf{p}_i} - l_{\mathbf{p}_i}}$$

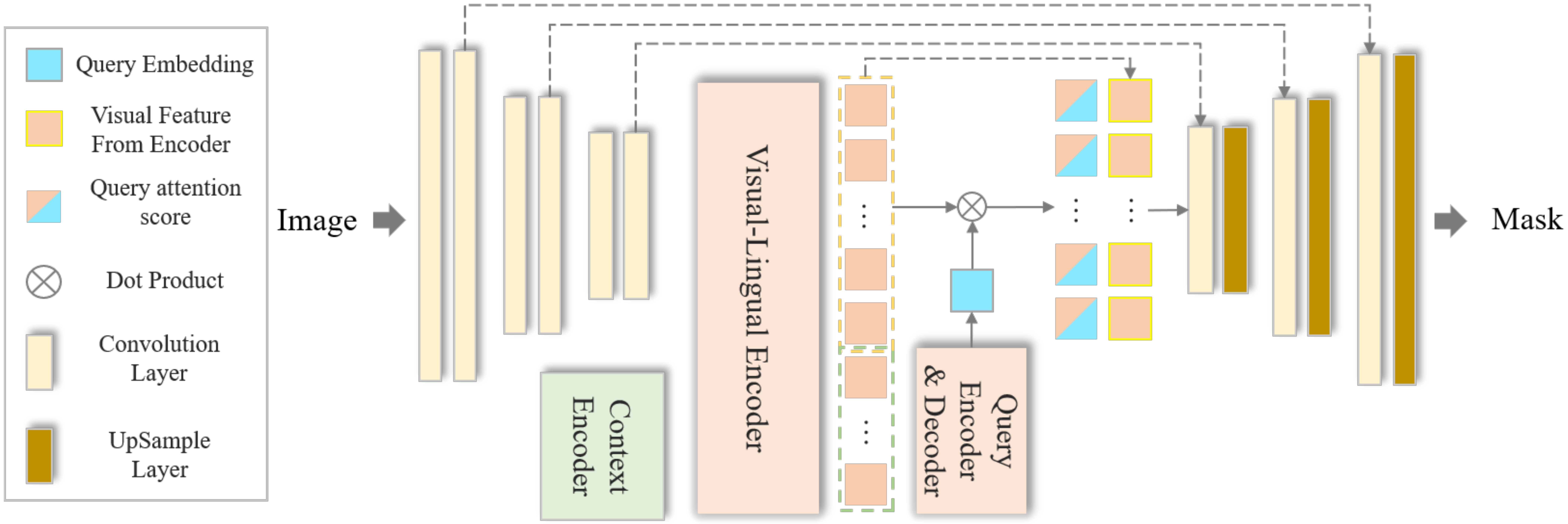
: Learnable bias

: Multi-modal context information

: Encoding of referred phrase

Task Heads

[Li et al., 2021]



REC Head: A linear layer that predicts a bounding box

RES Head: A FPN (U-Net type) structure with residual connections

Multi-task Supervision

[Li et al., 2021]

REC: Given predicted bounding box and ground truth bounding box

$$\mathcal{L}_{det} = \lambda_{iou}\mathcal{L}_{iou}(\mathbf{b}, \tilde{\mathbf{b}}) + \lambda_{L1}||\mathbf{b} - \tilde{\mathbf{b}}||_1$$

Generalized IOU loss

Standard L1 loss

Multi-task Supervision

[Li et al., 2021]

REC: Given predicted bounding box and ground truth bounding box

$$\mathcal{L}_{det} = \lambda_{iou}\mathcal{L}_{iou}(\mathbf{b}, \tilde{\mathbf{b}}) + \lambda_{L1}||\mathbf{b} - \tilde{\mathbf{b}}||_1$$

Generalized IOU loss

Standard L1 loss

RES: Given predicted segmentation and ground truth segmentation mask

$$\mathcal{L}_{seg} = \lambda_{focal}\mathcal{L}_{focal}(\mathbf{s}, \tilde{\mathbf{s}}) + \lambda_{dice}\mathcal{L}_{dice}(\mathbf{s}, \tilde{\mathbf{s}})$$

Focal loss

Dice loss: Generalized IOU loss for segmentation

Multi-task Supervision

[Li et al., 2021]

REC: Given predicted bounding box and ground truth bounding box

$$\mathcal{L}_{det} = \lambda_{iou}\mathcal{L}_{iou}(\mathbf{b}, \tilde{\mathbf{b}}) + \lambda_{L1}||\mathbf{b} - \tilde{\mathbf{b}}||_1$$

Generalized IOU loss

Standard L1 loss

RES: Given predicted segmentation and ground truth segmentation mask

$$\mathcal{L}_{seg} = \lambda_{focal}\mathcal{L}_{focal}(\mathbf{s}, \tilde{\mathbf{s}}) + \lambda_{dice}\mathcal{L}_{dice}(\mathbf{s}, \tilde{\mathbf{s}})$$

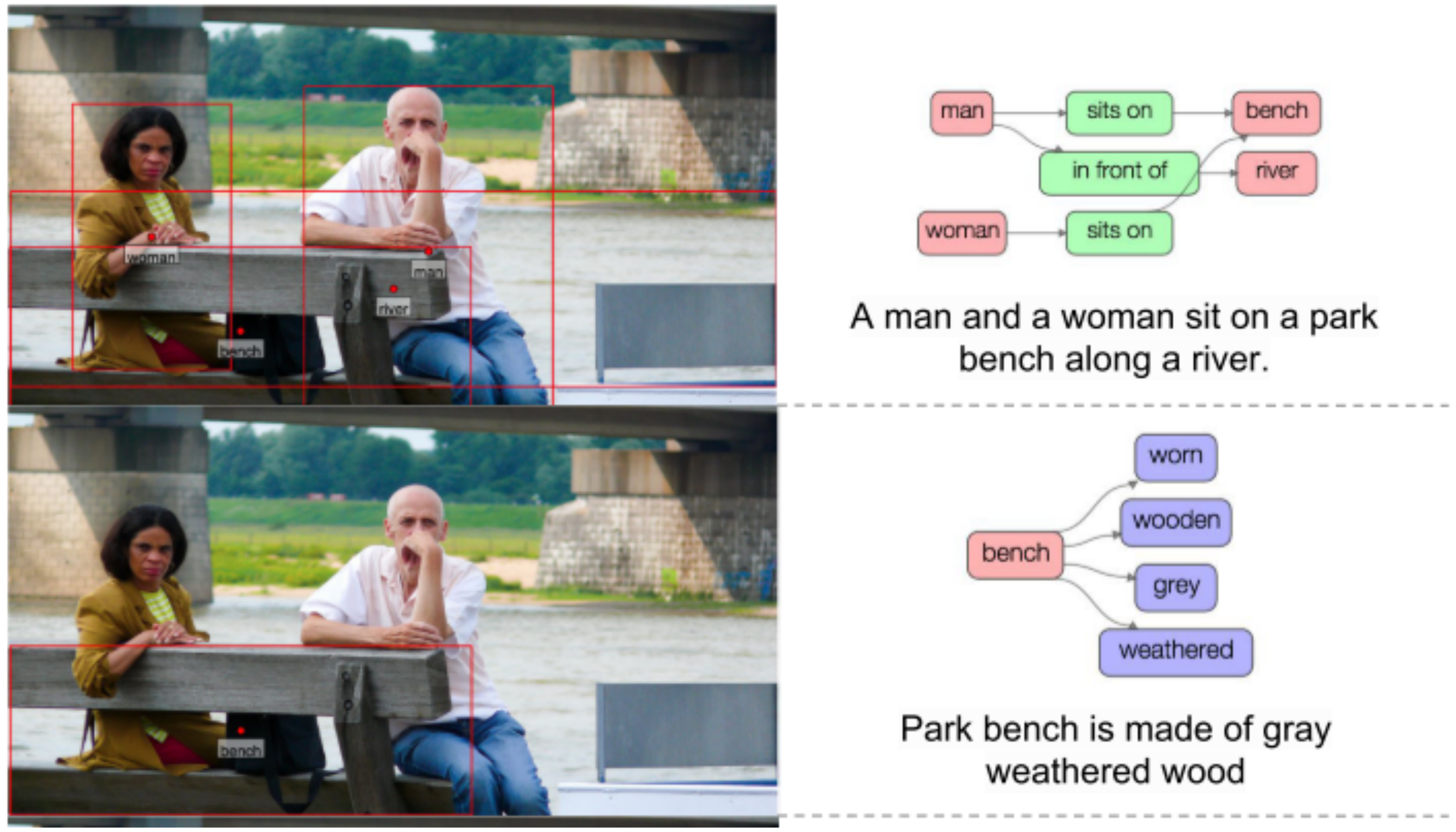
Focal loss

Dice loss: Generalized IOU loss for segmentation

$$\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{det}$$

Pre-training

[Li et al., 2021]



A man and a woman sit on a park bench along a river.

Park bench is made of gray weathered wood

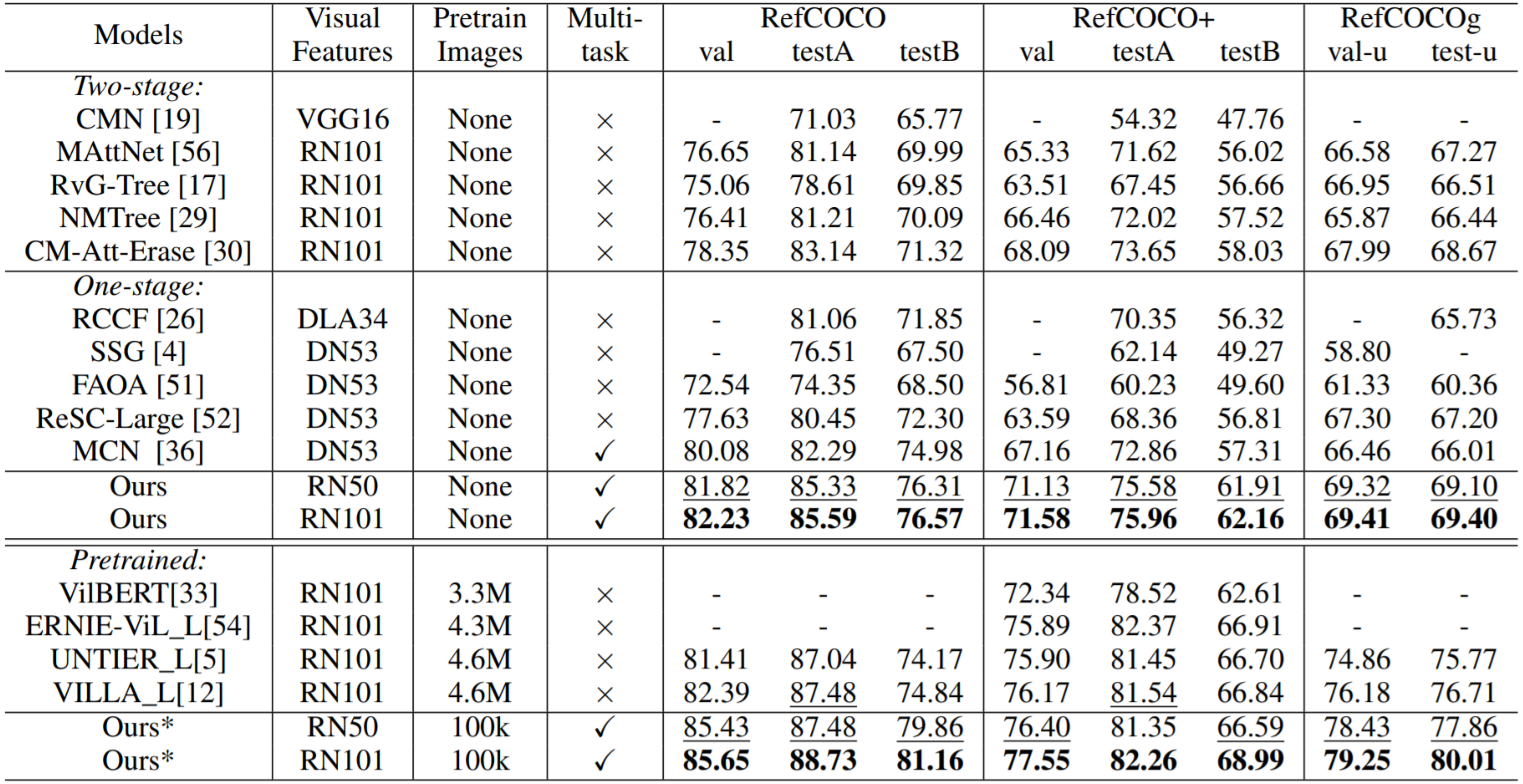
— Transformers can easily overfit

— Visual Genome (VG) contains description for each region

— We use the annotation from VG to pretrain our transformer by letting the network predict region bounding boxes given region description

Results on REC task (Multi-task Model)

[Li et al., 2021]

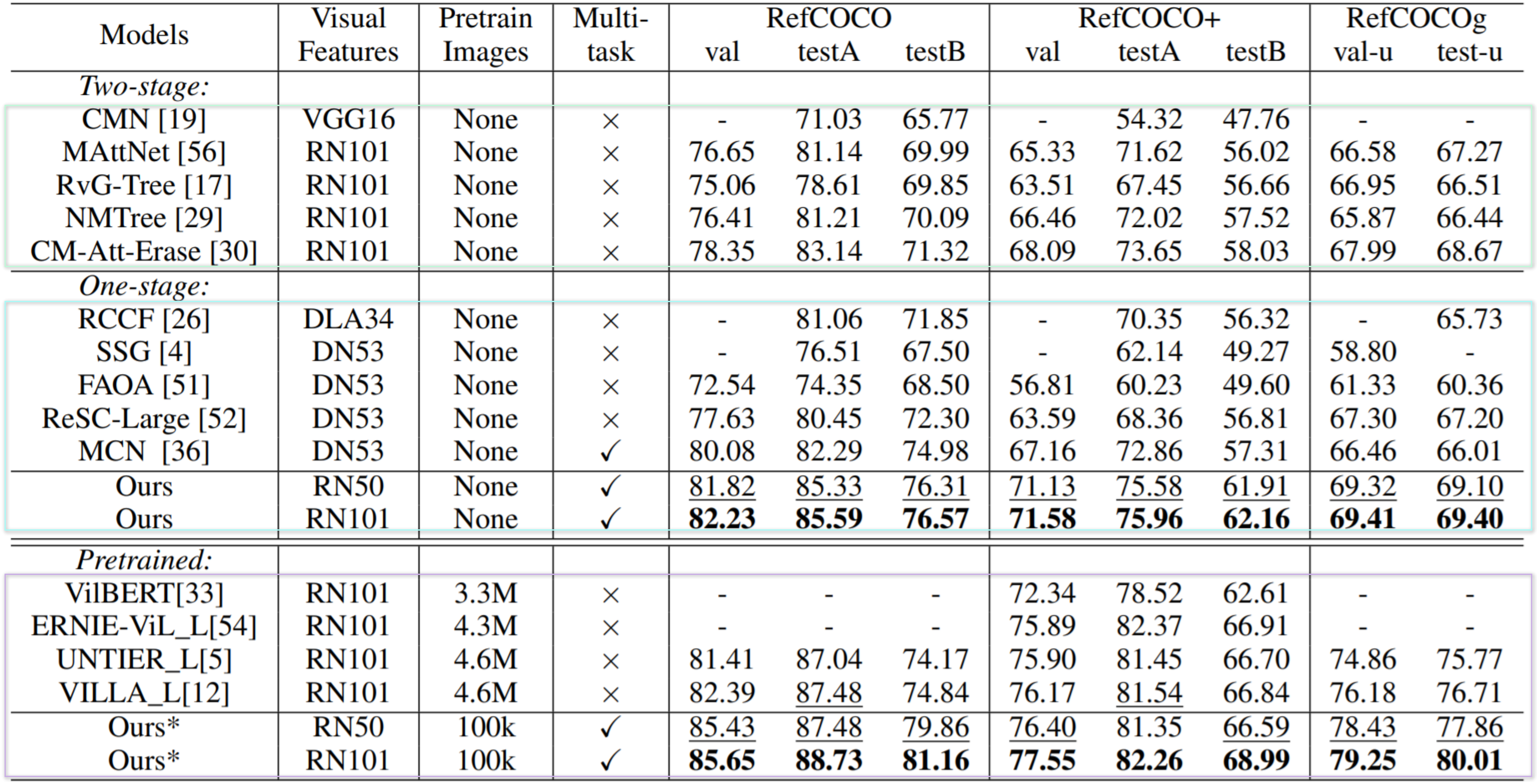


Models	Visual Features	Pretrain Images	Multi-task	RefCOCO			RefCOCO+			RefCOCOg	
				val	testA	testB	val	testA	testB	val-u	test-u
Two-stage:											
CMN [19]	VGG16	None	×	-	71.03	65.77	-	54.32	47.76	-	-
MAttNet [56]	RN101	None	×	76.65	81.14	69.99	65.33	71.62	56.02	66.58	67.27
RvG-Tree [17]	RN101	None	×	75.06	78.61	69.85	63.51	67.45	56.66	66.95	66.51
NMTree [29]	RN101	None	×	76.41	81.21	70.09	66.46	72.02	57.52	65.87	66.44
CM-Att-Erase [30]	RN101	None	×	78.35	83.14	71.32	68.09	73.65	58.03	67.99	68.67
One-stage:											
RCCF [26]	DLA34	None	×	-	81.06	71.85	-	70.35	56.32	-	65.73
SSG [4]	DN53	None	×	-	76.51	67.50	-	62.14	49.27	58.80	-
FAOA [51]	DN53	None	×	72.54	74.35	68.50	56.81	60.23	49.60	61.33	60.36
ReSC-Large [52]	DN53	None	×	77.63	80.45	72.30	63.59	68.36	56.81	67.30	67.20
MCN [36]	DN53	None	✓	80.08	82.29	74.98	67.16	72.86	57.31	66.46	66.01
Ours	RN50	None	✓	81.82	85.33	76.31	71.13	75.58	61.91	69.32	69.10
Ours	RN101	None	✓	**82.23**	**85.59**	**76.57**	**71.58**	**75.96**	**62.16**	**69.41**	**69.40**
Pretrained:											
VilBERT[33]	RN101	3.3M	×	-	-	-	72.34	78.52	62.61	-	-
ERNIE-ViL_L[54]	RN101	4.3M	×	-	-	-	75.89	82.37	66.91	-	-
UNTIER_L[5]	RN101	4.6M	×	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77
VILLA_L[12]	RN101	4.6M	×	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71
Ours*	RN50	100k	✓	85.43	87.48	79.86	76.40	81.35	66.59	78.43	77.86
Ours*	RN101	100k	✓	**85.65**	**88.73**	**81.16**	**77.55**	**82.26**	**68.99**	**79.25**	**80.01**

Evaluation Metric: Prec@0.5 (mark a detection as correct if its bounding box has a IOU>0.5 with the ground truth)

Results on REC task (Multi-task Model)

[Li et al., 2021]



Two-staged

One-staged

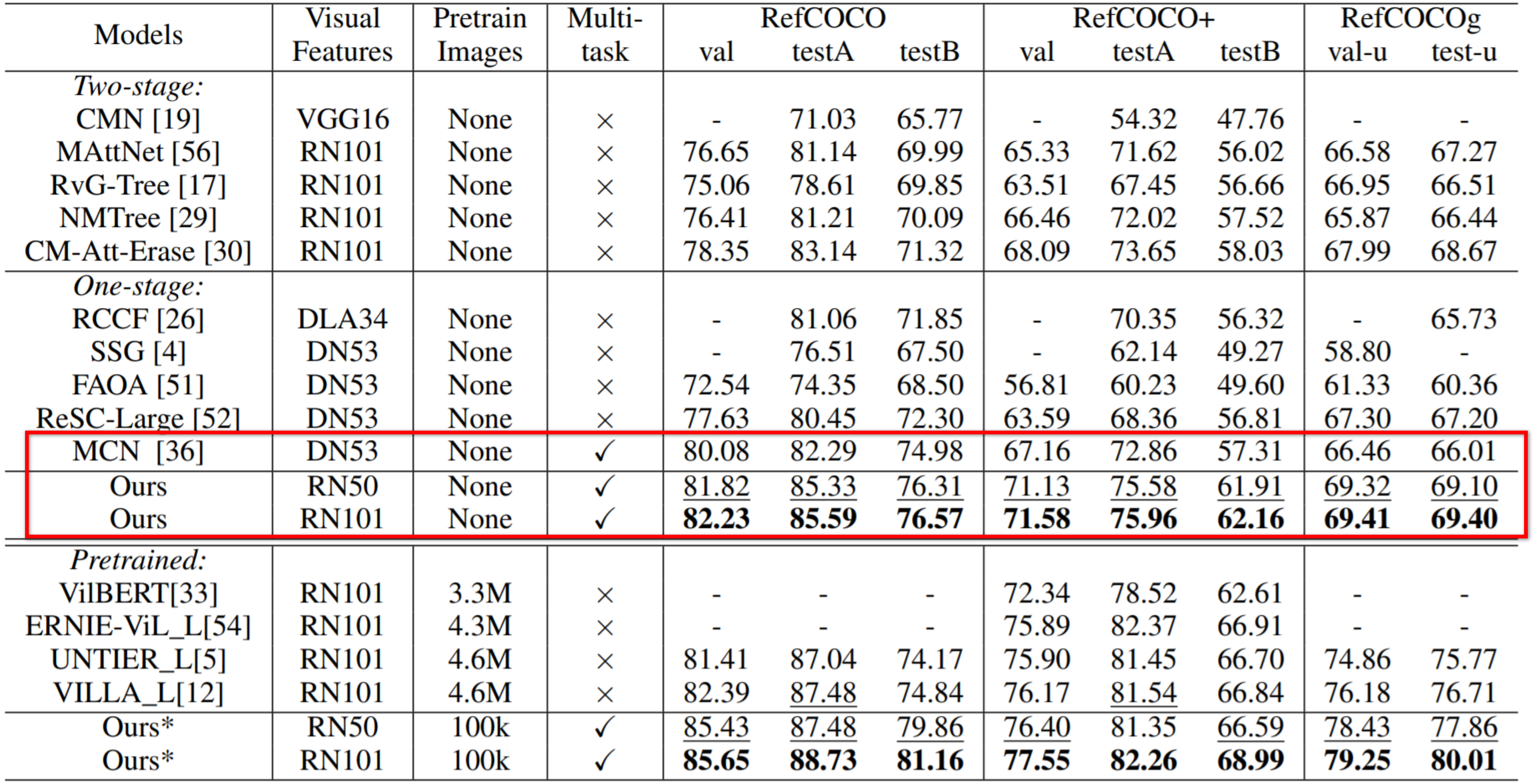
With Pretrain

Models	Visual Features	Pretrain Images	Multi-task	RefCOCO			RefCOCO+			RefCOCOg	
				val	testA	testB	val	testA	testB	val-u	test-u
Two-stage:											
CMN [19]	VGG16	None	×	-	71.03	65.77	-	54.32	47.76	-	-
MAttNet [56]	RN101	None	×	76.65	81.14	69.99	65.33	71.62	56.02	66.58	67.27
RvG-Tree [17]	RN101	None	×	75.06	78.61	69.85	63.51	67.45	56.66	66.95	66.51
NMTree [29]	RN101	None	×	76.41	81.21	70.09	66.46	72.02	57.52	65.87	66.44
CM-Att-Erase [30]	RN101	None	×	78.35	83.14	71.32	68.09	73.65	58.03	67.99	68.67
One-stage:											
RCCF [26]	DLA34	None	×	-	81.06	71.85	-	70.35	56.32	-	65.73
SSG [4]	DN53	None	×	-	76.51	67.50	-	62.14	49.27	58.80	-
FAOA [51]	DN53	None	×	72.54	74.35	68.50	56.81	60.23	49.60	61.33	60.36
ReSC-Large [52]	DN53	None	×	77.63	80.45	72.30	63.59	68.36	56.81	67.30	67.20
MCN [36]	DN53	None	✓	80.08	82.29	74.98	67.16	72.86	57.31	66.46	66.01
Ours	RN50	None	✓	81.82	85.33	76.31	71.13	75.58	61.91	69.32	69.10
Ours	RN101	None	✓	**82.23**	**85.59**	**76.57**	**71.58**	**75.96**	**62.16**	**69.41**	**69.40**
Pretrained:											
VilBERT[33]	RN101	3.3M	×	-	-	-	72.34	78.52	62.61	-	-
ERNIE-ViL_L[54]	RN101	4.3M	×	-	-	-	75.89	82.37	66.91	-	-
UNTIER_L[5]	RN101	4.6M	×	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77
VILLA_L[12]	RN101	4.6M	×	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71
Ours*	RN50	100k	✓	85.43	87.48	79.86	76.40	81.35	66.59	78.43	77.86
Ours*	RN101	100k	✓	**85.65**	**88.73**	**81.16**	**77.55**	**82.26**	**68.99**	**79.25**	**80.01**

Evaluation Metric: Prec@0.5 (mark a detection as correct if its bounding box has a IOU>0.5 with the ground truth)

Results on REC task (Multi-task Model)

[Li et al., 2021]

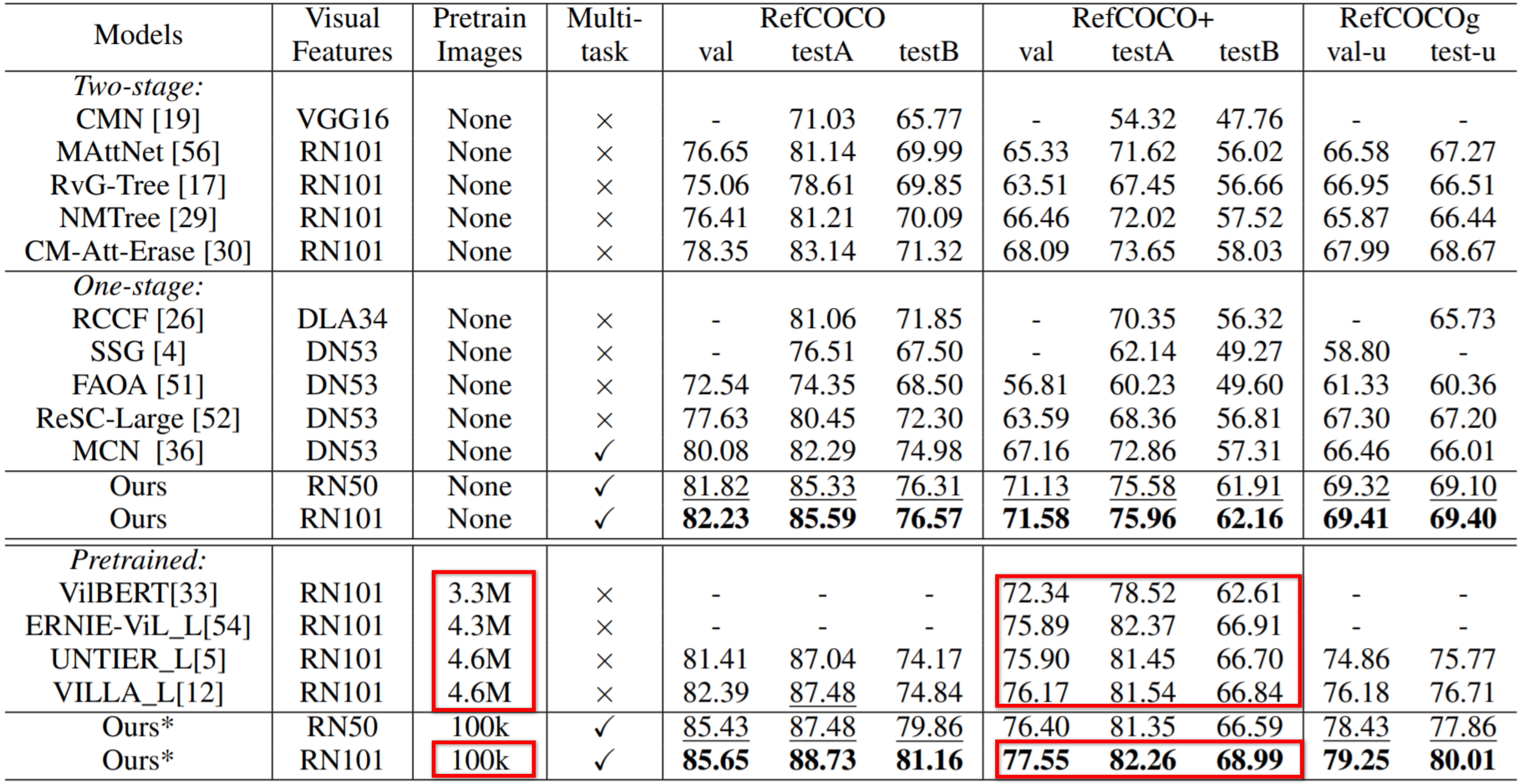


Models	Visual Features	Pretrain Images	Multi-task	RefCOCO			RefCOCO+			RefCOCOg	
				val	testA	testB	val	testA	testB	val-u	test-u
Two-stage:											
CMN [19]	VGG16	None	×	-	71.03	65.77	-	54.32	47.76	-	-
MAttNet [56]	RN101	None	×	76.65	81.14	69.99	65.33	71.62	56.02	66.58	67.27
RvG-Tree [17]	RN101	None	×	75.06	78.61	69.85	63.51	67.45	56.66	66.95	66.51
NMTree [29]	RN101	None	×	76.41	81.21	70.09	66.46	72.02	57.52	65.87	66.44
CM-Att-Erase [30]	RN101	None	×	78.35	83.14	71.32	68.09	73.65	58.03	67.99	68.67
One-stage:											
RCCF [26]	DLA34	None	×	-	81.06	71.85	-	70.35	56.32	-	65.73
SSG [4]	DN53	None	×	-	76.51	67.50	-	62.14	49.27	58.80	-
FAOA [51]	DN53	None	×	72.54	74.35	68.50	56.81	60.23	49.60	61.33	60.36
ReSC-Large [52]	DN53	None	×	77.63	80.45	72.30	63.59	68.36	56.81	67.30	67.20
MCN [36]	DN53	None	✓	80.08	82.29	74.98	67.16	72.86	57.31	66.46	66.01
Ours	RN50	None	✓	81.82	85.33	76.31	71.13	75.58	61.91	69.32	69.10
Ours	RN101	None	✓	**82.23**	**85.59**	**76.57**	**71.58**	**75.96**	**62.16**	**69.41**	**69.40**
Pretrained:											
VilBERT[33]	RN101	3.3M	×	-	-	-	72.34	78.52	62.61	-	-
ERNIE-ViL_L[54]	RN101	4.3M	×	-	-	-	75.89	82.37	66.91	-	-
UNTIER_L[5]	RN101	4.6M	×	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77
VILLA_L[12]	RN101	4.6M	×	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71
Ours*	RN50	100k	✓	85.43	87.48	79.86	76.40	81.35	66.59	78.43	77.86
Ours*	RN101	100k	✓	**85.65**	**88.73**	**81.16**	**77.55**	**82.26**	**68.99**	**79.25**	**80.01**

Our model and MCN are the only multi-task setting models

Results on REC task (Multi-task Model)

[Li et al., 2021]

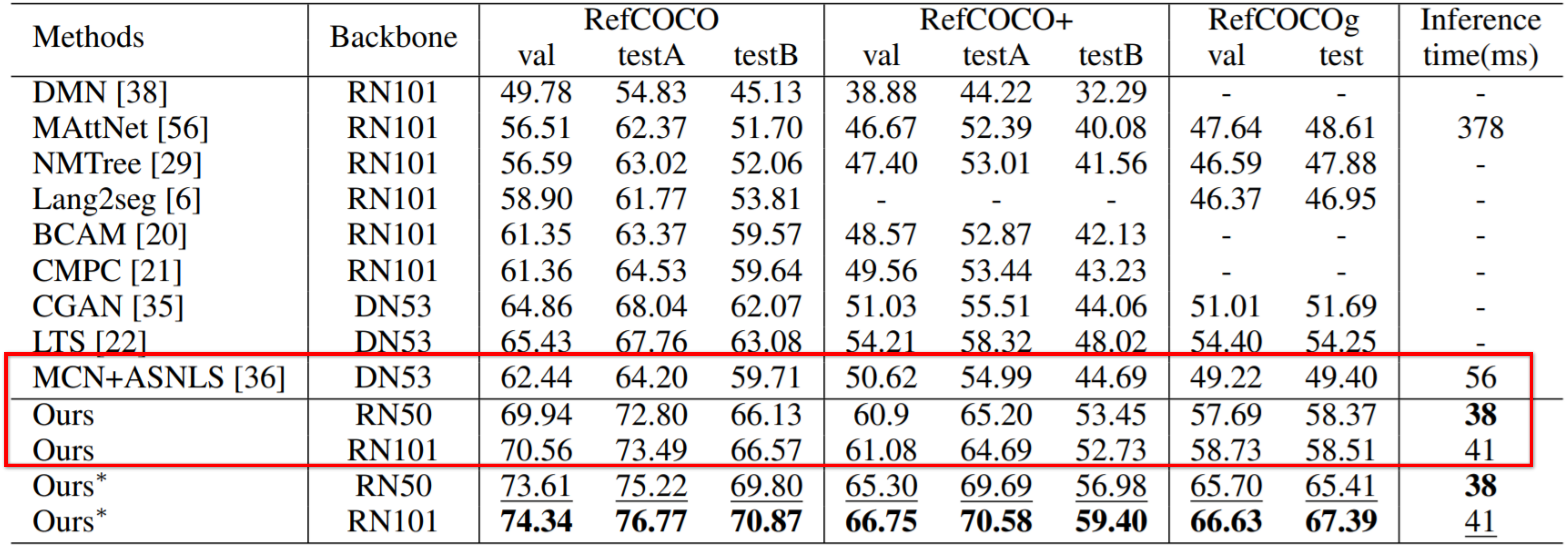


Models	Visual Features	Pretrain Images	Multi-task	RefCOCO			RefCOCO+			RefCOCOg	
				val	testA	testB	val	testA	testB	val-u	test-u
Two-stage:											
CMN [19]	VGG16	None	×	-	71.03	65.77	-	54.32	47.76	-	-
MAttNet [56]	RN101	None	×	76.65	81.14	69.99	65.33	71.62	56.02	66.58	67.27
RvG-Tree [17]	RN101	None	×	75.06	78.61	69.85	63.51	67.45	56.66	66.95	66.51
NMTree [29]	RN101	None	×	76.41	81.21	70.09	66.46	72.02	57.52	65.87	66.44
CM-Att-Erase [30]	RN101	None	×	78.35	83.14	71.32	68.09	73.65	58.03	67.99	68.67
One-stage:											
RCCF [26]	DLA34	None	×	-	81.06	71.85	-	70.35	56.32	-	65.73
SSG [4]	DN53	None	×	-	76.51	67.50	-	62.14	49.27	58.80	-
FAOA [51]	DN53	None	×	72.54	74.35	68.50	56.81	60.23	49.60	61.33	60.36
ReSC-Large [52]	DN53	None	×	77.63	80.45	72.30	63.59	68.36	56.81	67.30	67.20
MCN [36]	DN53	None	✓	80.08	82.29	74.98	67.16	72.86	57.31	66.46	66.01
Ours	RN50	None	✓	81.82	85.33	76.31	71.13	75.58	61.91	69.32	69.10
Ours	RN101	None	✓	**82.23**	**85.59**	**76.57**	**71.58**	**75.96**	**62.16**	**69.41**	**69.40**
Pretrained:											
ViLBERT[33]	RN101	3.3M	×	-	-	-	72.34	78.52	62.61	-	-
ERNIE-ViL_L[54]	RN101	4.3M	×	-	-	-	75.89	82.37	66.91	-	-
UNTIER_L[5]	RN101	4.6M	×	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77
VILLA_L[12]	RN101	4.6M	×	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71
Ours*	RN50	100k	✓	85.43	87.48	79.86	76.40	81.35	66.59	78.43	77.86
Ours*	RN101	100k	✓	**85.65**	**88.73**	**81.16**	**77.55**	**82.26**	**68.99**	**79.25**	**80.01**

Our model is state-of-the-art despite pre-training on less data

Results on RES tasks (Multi-task Model)

[Li et al., 2021]



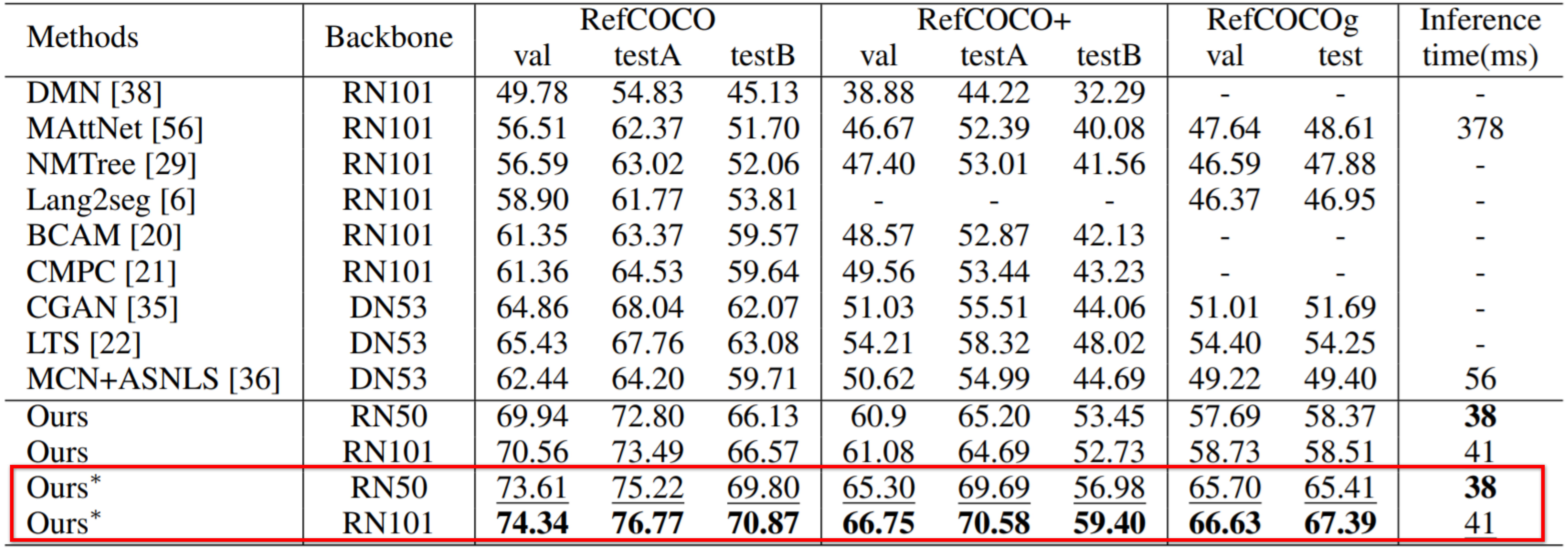
Methods	Backbone	RefCOCO			RefCOCO+			RefCOCOg		Inference time(ms)
		val	testA	testB	val	testA	testB	val	test	
DMN [38]	RN101	49.78	54.83	45.13	38.88	44.22	32.29	-	-	-
MAttNet [56]	RN101	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	378
NMTree [29]	RN101	56.59	63.02	52.06	47.40	53.01	41.56	46.59	47.88	-
Lang2seg [6]	RN101	58.90	61.77	53.81	-	-	-	46.37	46.95	-
BCAM [20]	RN101	61.35	63.37	59.57	48.57	52.87	42.13	-	-	-
CMPC [21]	RN101	61.36	64.53	59.64	49.56	53.44	43.23	-	-	-
CGAN [35]	DN53	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	-
LTS [22]	DN53	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
MCN+ASNLS [36]	DN53	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	56
Ours	RN50	69.94	72.80	66.13	60.9	65.20	53.45	57.69	58.37	**38**
Ours	RN101	70.56	73.49	66.57	61.08	64.69	52.73	58.73	58.51	41
Ours*	RN50	73.61	75.22	69.80	65.30	69.69	56.98	65.70	65.41	**38**
Ours*	RN101	**74.34**	**76.77**	**70.87**	**66.75**	**70.58**	**59.40**	**66.63**	**67.39**	41

Ours* denote the model is first pre-trained on Visual Genome.

Evaluation Metric: Mean IOU

Results on RES tasks (Multi-task Model)

[Li et al., 2021]

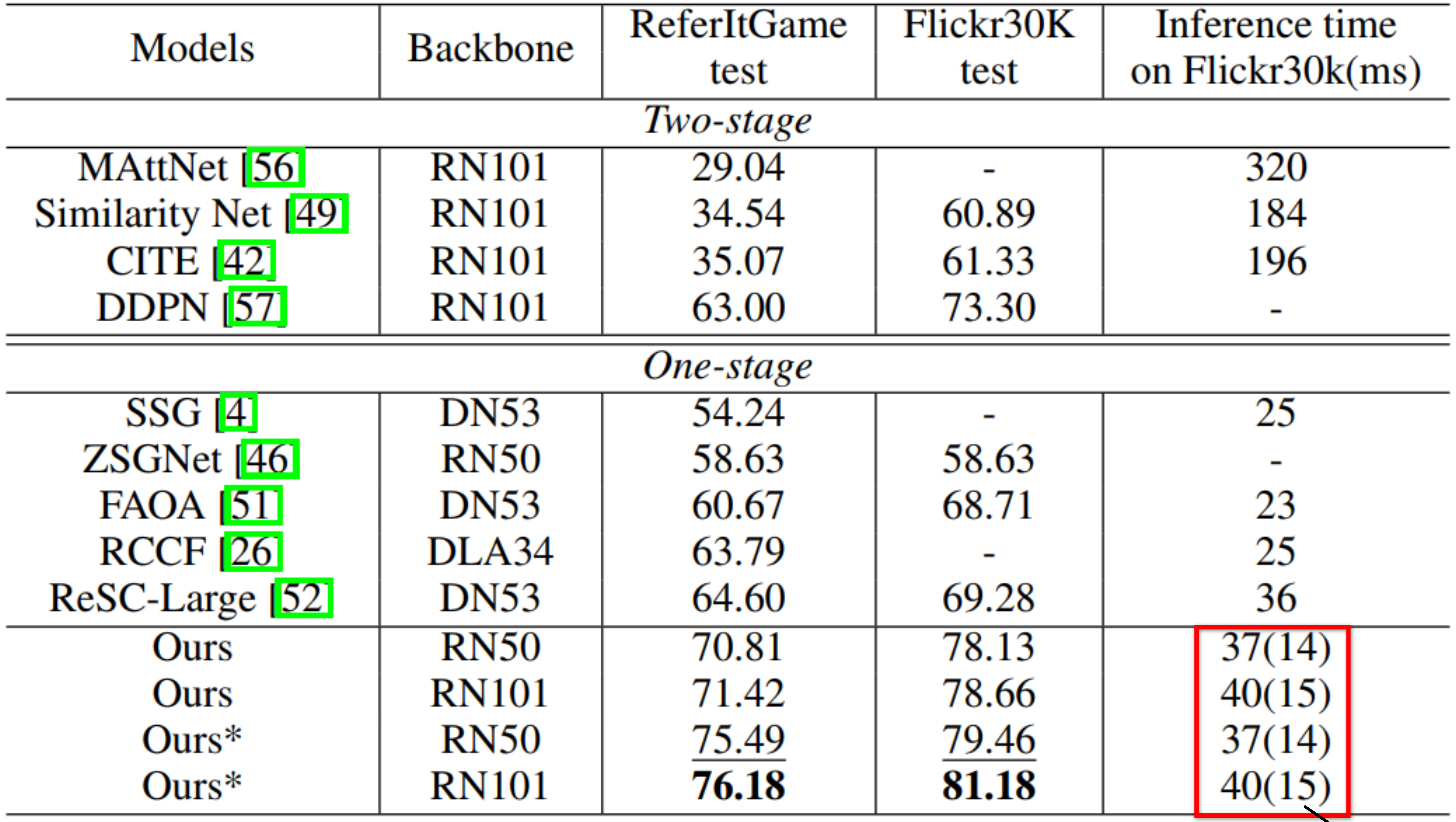


Methods	Backbone	RefCOCO			RefCOCO+			RefCOCOg		Inference
		val	testA	testB	val	testA	testB	val	test	time(ms)
DMN [38]	RN101	49.78	54.83	45.13	38.88	44.22	32.29	-	-	-
MAttNet [56]	RN101	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	378
NMTree [29]	RN101	56.59	63.02	52.06	47.40	53.01	41.56	46.59	47.88	-
Lang2seg [6]	RN101	58.90	61.77	53.81	-	-	-	46.37	46.95	-
BCAM [20]	RN101	61.35	63.37	59.57	48.57	52.87	42.13	-	-	-
CMPC [21]	RN101	61.36	64.53	59.64	49.56	53.44	43.23	-	-	-
CGAN [35]	DN53	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	-
LTS [22]	DN53	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
MCN+ASNLS [36]	DN53	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	56
Ours	RN50	69.94	72.80	66.13	60.9	65.20	53.45	57.69	58.37	**38**
Ours	RN101	70.56	73.49	66.57	61.08	64.69	52.73	58.73	58.51	41
Ours*	RN50	73.61	75.22	69.80	65.30	69.69	56.98	65.70	65.41	**38**
Ours*	RN101	**74.34**	**76.77**	**70.87**	**66.75**	**70.58**	**59.40**	**66.63**	**67.39**	41

Note that there is no segmentation annotation in the pre-training stage

More Result on REC tasks

[Li et al., 2021]



Models	Backbone	ReferItGame test	Flickr30K test	Inference time on Flickr30k(ms)
Two-stage				
MAttNet [56]	RN101	29.04	-	320
Similarity Net [49]	RN101	34.54	60.89	184
CITE [42]	RN101	35.07	61.33	196
DDPN [57]	RN101	63.00	73.30	-
One-stage				
SSG [4]	DN53	54.24	-	25
ZSGNet [46]	RN50	58.63	58.63	-
FAOA [51]	DN53	60.67	68.71	23
RCCF [26]	DLA34	63.79	-	25
ReSC-Large [52]	DN53	64.60	69.28	36
Ours	RN50	70.81	78.13	37(14)
Ours	RN101	71.42	78.66	40(15)
Ours*	RN50	75.49	79.46	37(14)
Ours*	RN101	**76.18**	**81.18**	40(15)

One Expression Phrase
per Inference

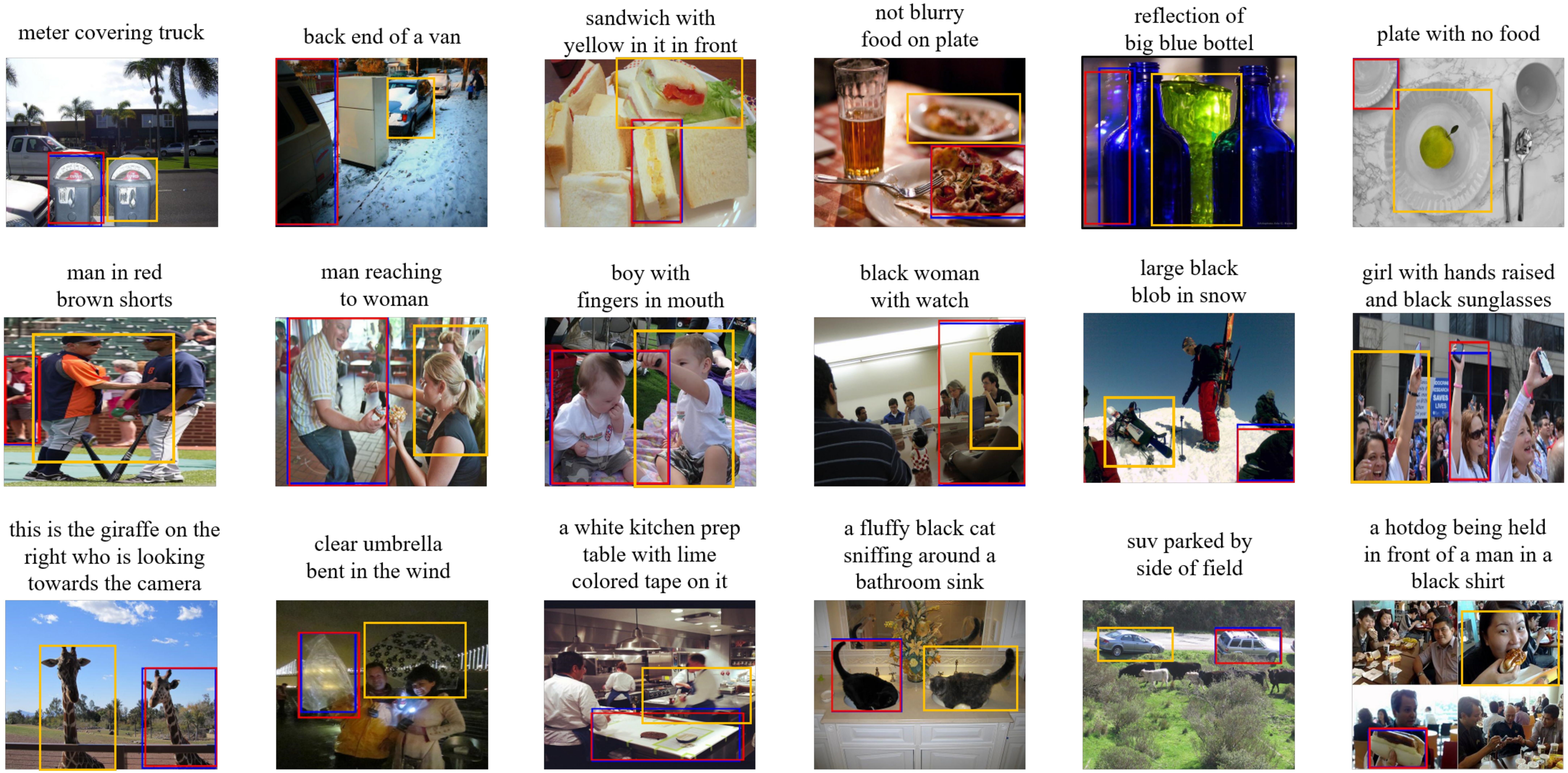
Multiple Expression Phrase
per Inference

Inference Time/ per Expression

In Flickr30k, context sentence is provided.

Qualitative result on REC tasks

Ground Truth **Our Model** **MCN** (baseline)



meter covering truck

back end of a van

sandwich with yellow in it in front

not blurry food on plate

reflection of big blue bottel

plate with no food

man in red brown shorts

man reaching to woman

boy with fingers in mouth

black woman with watch

large black blob in snow

girl with hands raised and black sunglasses

this is the giraffe on the right who is looking towards the camera

clear umbrella bent in the wind

a white kitchen prep table with lime colored tape on it

a fluffy black cat sniffing around a bathroom sink

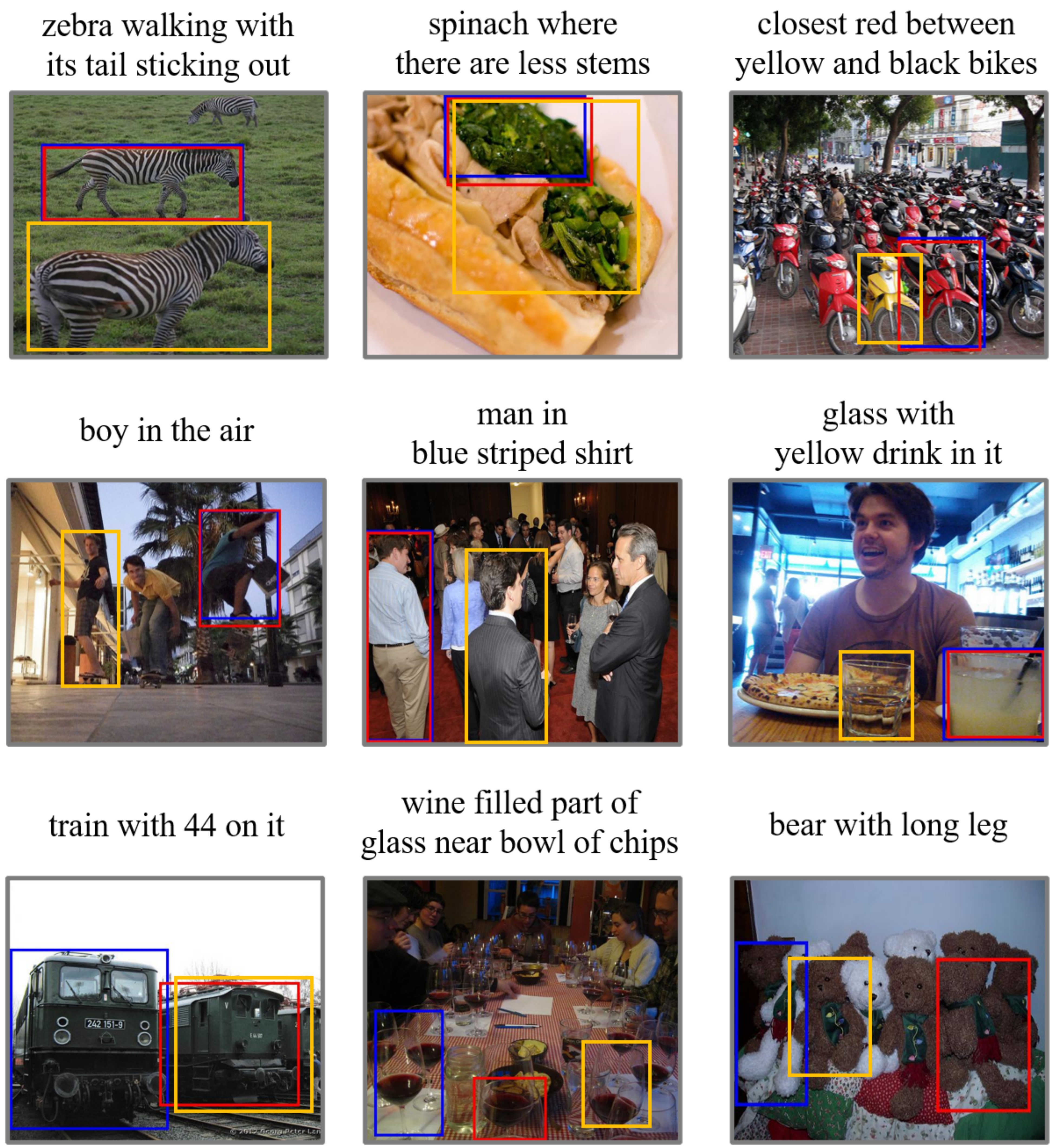
suv parked by side of field

a hotdog being held in front of a man in a black shirt

[Li et al., 2021]

Qualitative result on REC tasks

Ground Truth **Our Model** **MCN** (baseline)



zebra walking with its tail sticking out

spinach where there are less stems

closest red between yellow and black bikes

boy in the air

man in blue striped shirt

glass with yellow drink in it

train with 44 on it

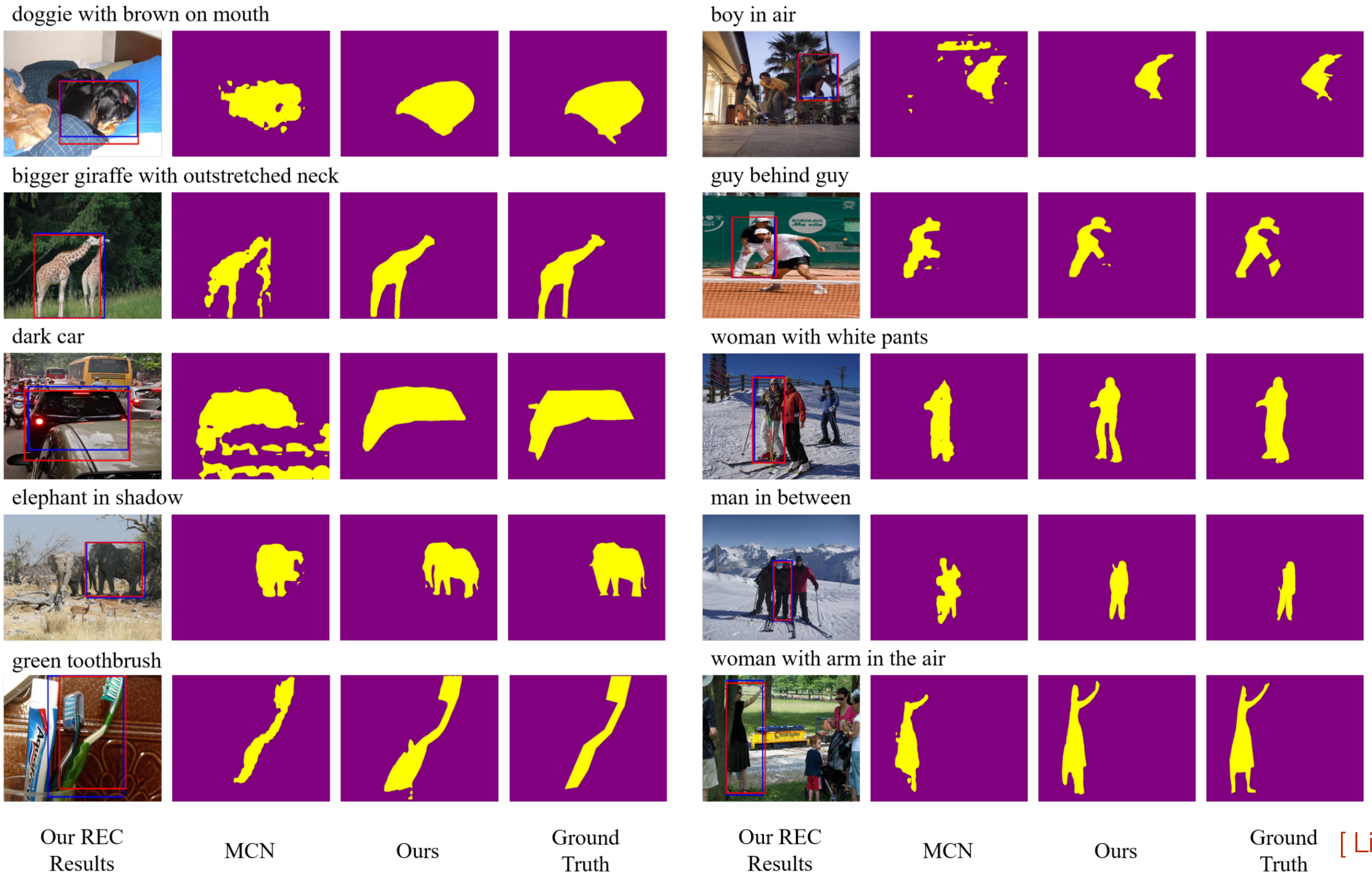
wine filled part of glass near bowl of chips

bear with long leg

Failure cases:

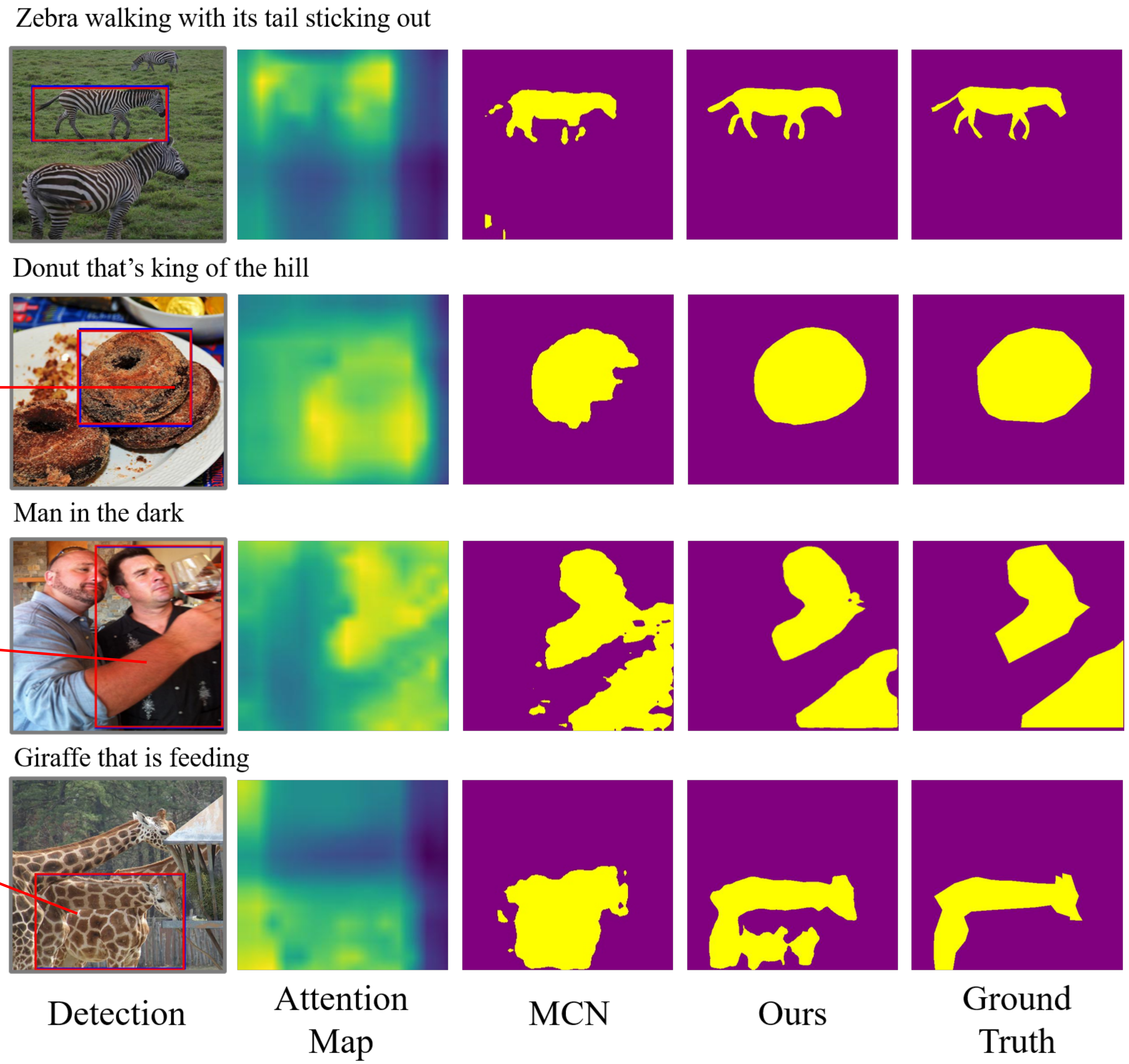
[Li et al., 2021]

Referring Expression **Segmentation** (RES)



[Li et al., 2021]

Referring Expression **Segmentation** (RES)



shadow / distortion

occlusion

challenging foreground / background

[Li et al., 2021]