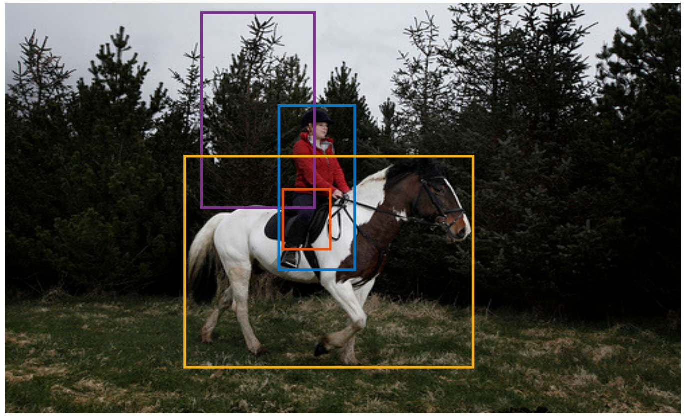
EGTR: Extracting Graph from Transformer for Scene Graph Generation

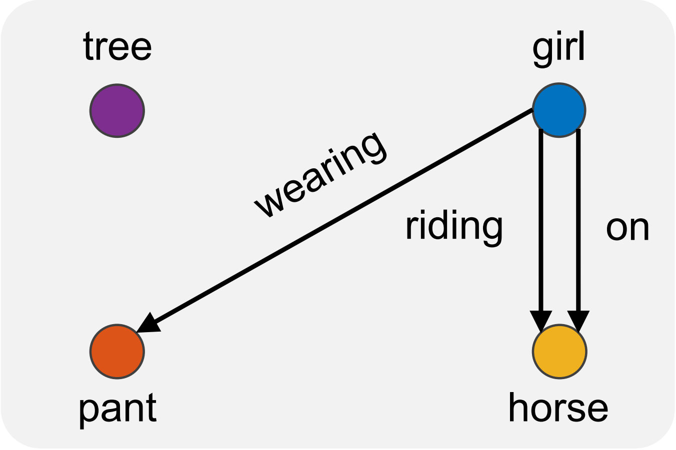
Jinbae Im[1], JeongYeon Nam[1], Nokyung Park[1, 2, 3], Hyungmin Lee[2], Seunghyun Park[1]

[1]NAVER Cloud AI, [2]NAVER, [3]Korea University

Preliminary: Scene Graph Generation (SGG)

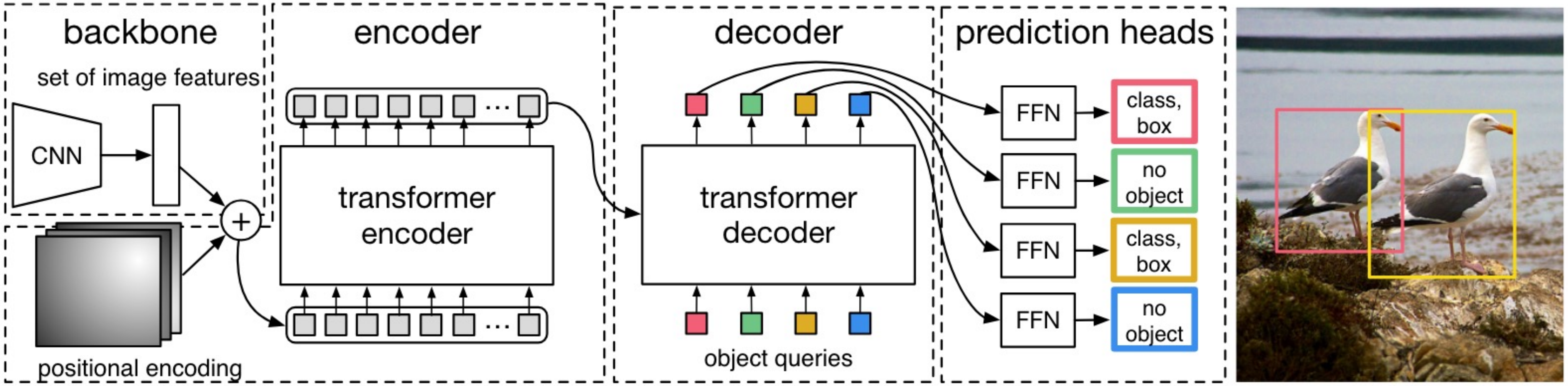
- Scene graph
  - $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
- Nodes: objects ($v_i \in \mathcal{V}$)
  - $v_i^c \in C_v$: object category label
  - $v_i^b \in R^4$: box coordinates
- Edges: relations ($e_j \in \mathcal{E}$)
  - e_j represents the j-th triplet (s_j, p_j, o_j)
  - $s_j \in \mathcal{V}$ & $o_j \in \mathcal{V}$: related objects
  - $p_j^c \in C_p$: relation category label





Scene Graph

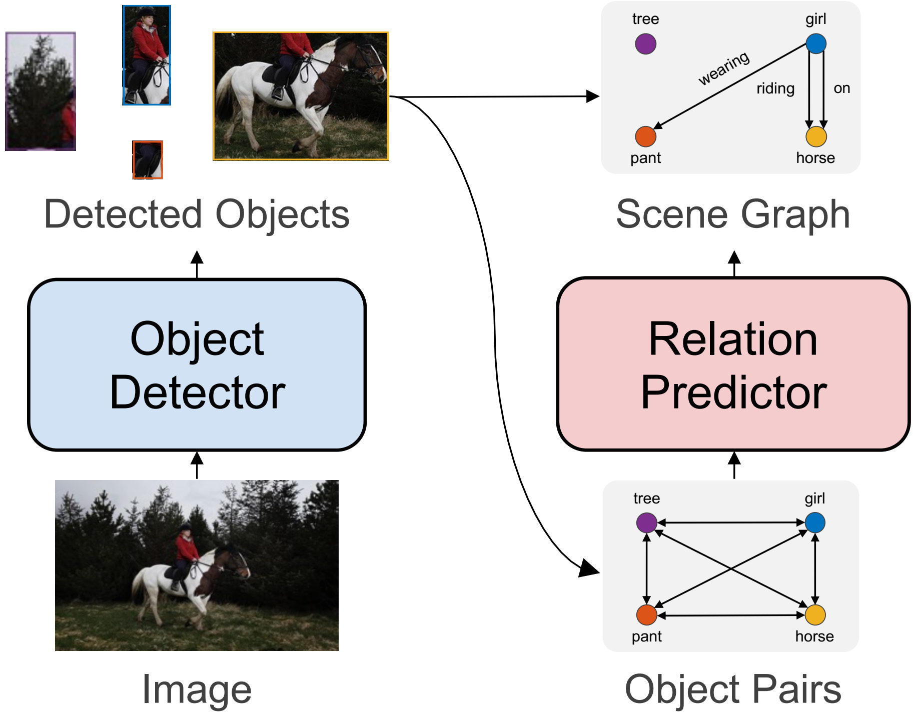
Preliminary: DETR



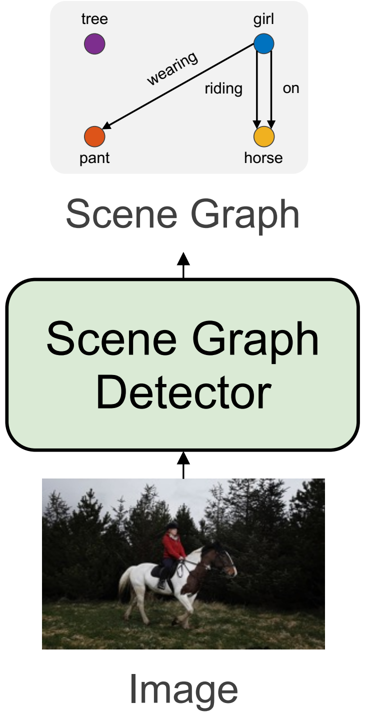
- One-stage (end-to-end) object detection model
- Each object query is used to detect each object
  - The number of object query (N) is set large enough to cover all objects
  - Bipartite matching between object queries and ground-truth objects is used

[ECCV 2020] End-to-End Object Detection with Transformers

SGG approaches

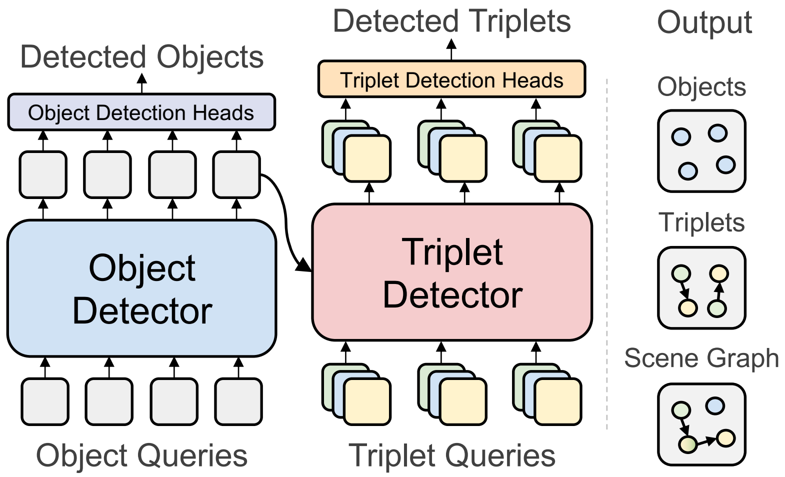


(a) Two-stage approaches



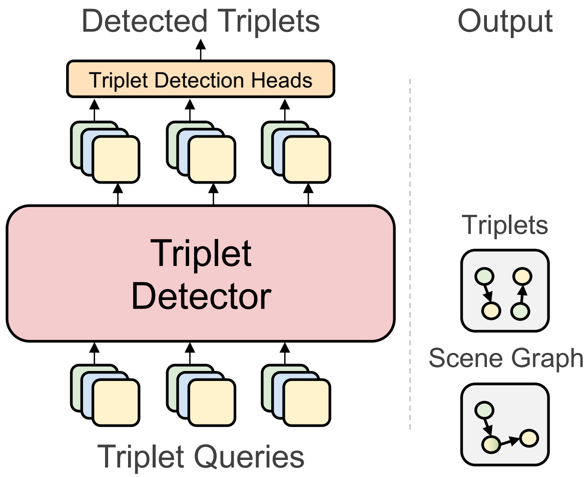
(b) One-stage approaches

Existing One-stage SGG Models



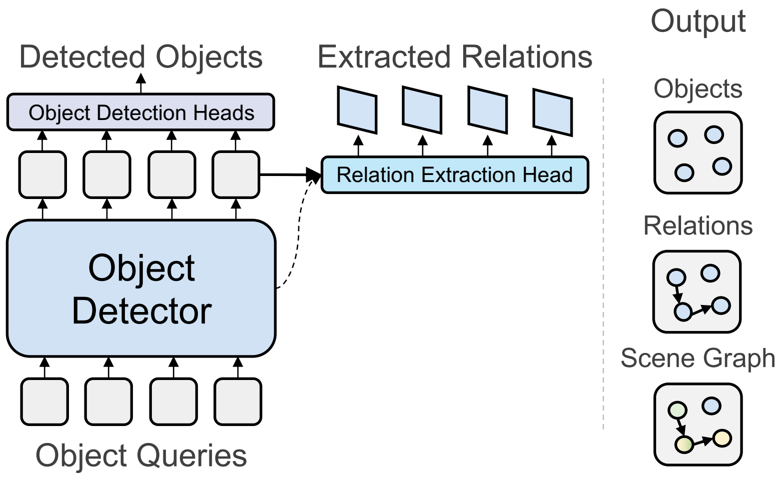
(a) Object-Triplet Detection Models

(e.g., RelTR, SGTR)



(b) Triplet Detection Models

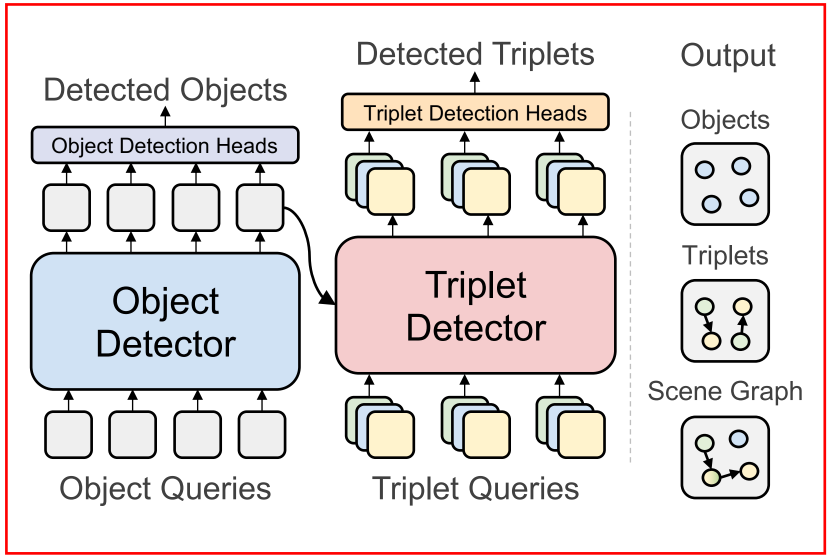
(e.g., Iterative SGG, Structured Sparse R-CNN)



(c) Relation Extraction Models

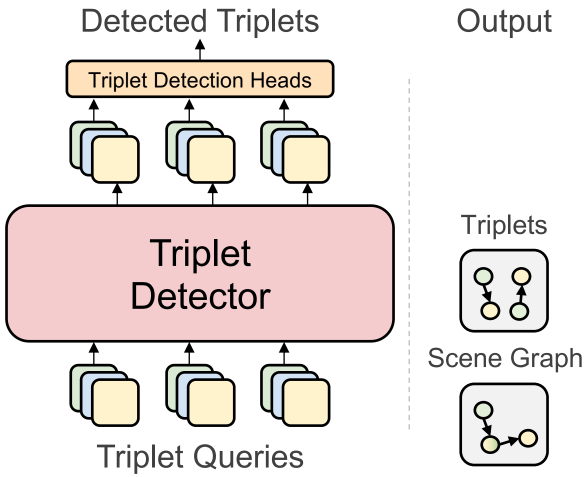
(e.g., Relationformer)

Existing One-stage SGG Models



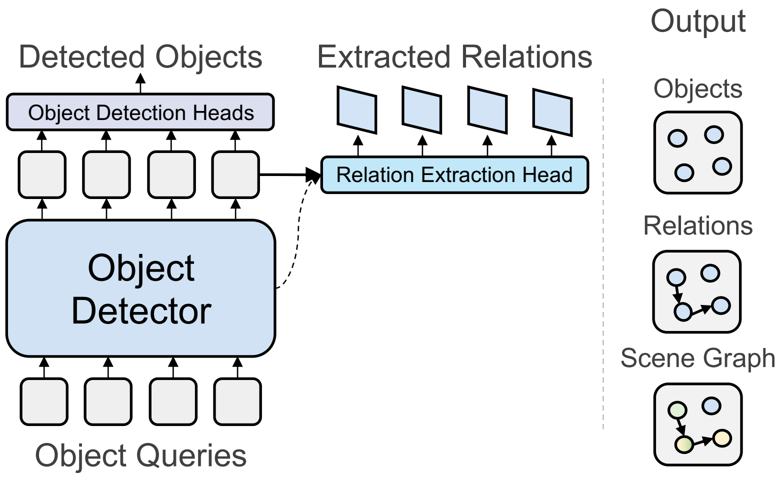
(a) Object-Triplet Detection Models

(e.g., RelTR, SGTR)



(b) Triplet Detection Models

(e.g., Iterative SGG, Structured Sparse R-CNN)

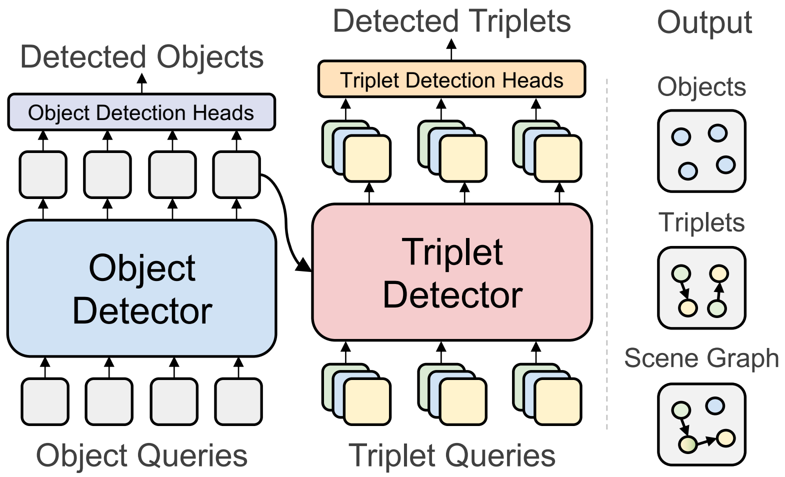


(c) Relation Extraction Models

(e.g., Relationformer)

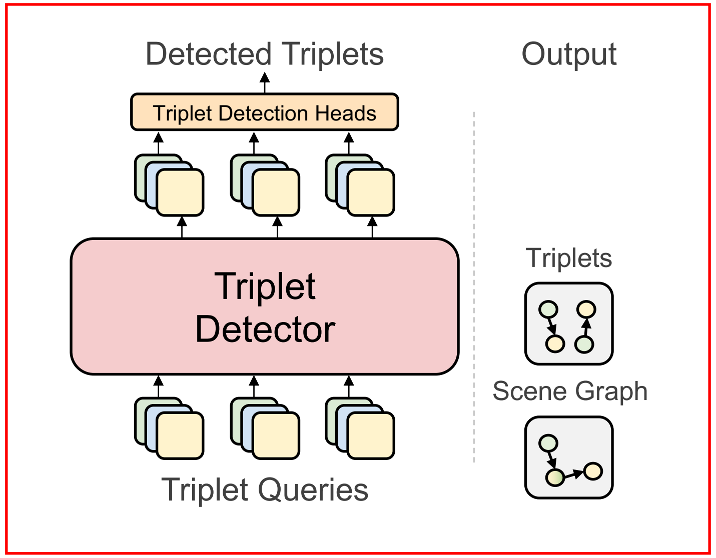
[IEEE TPAMI 2023] RelTR: Relation Transformer for Scene Graph Generation
[CVPR 2022] SGTR: End-to-end Scene Graph Generation with Transformer

Existing One-stage SGG Models



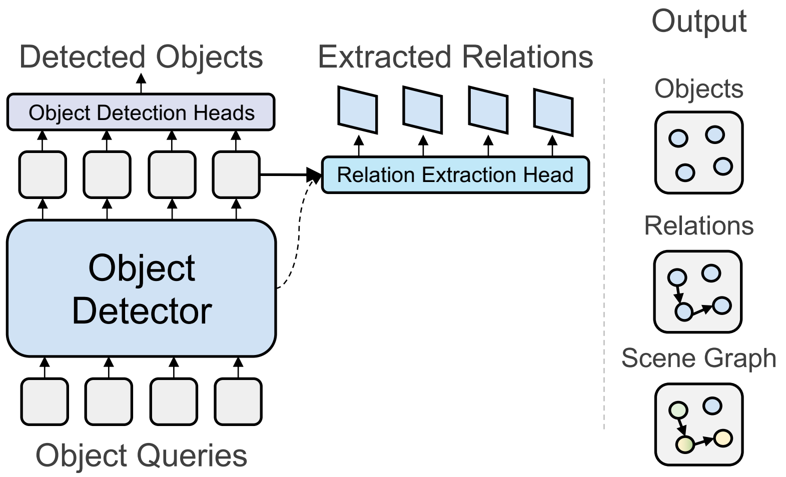
(a) Object-Triplet Detection Models

(e.g., RelTR, SGTR)



(b) Triplet Detection Models

(e.g., Iterative SGG, Structured Sparse R-CNN)

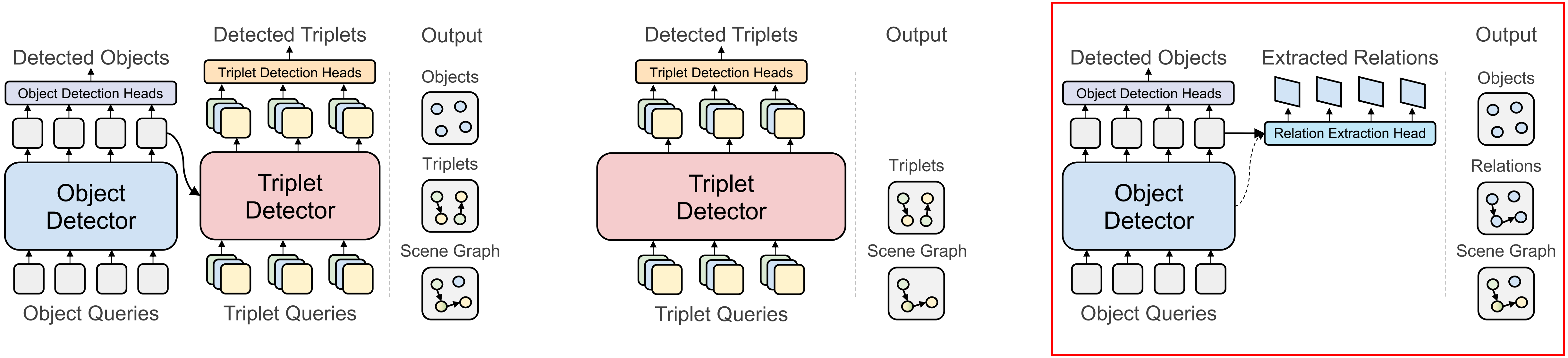


(c) Relation Extraction Models

(e.g., Relationformer)

[NeurIPS 2022] Iterative Scene Graph Generation
[CVPR 2022] Structured Sparse R-CNN for Direct Scene Graph Generation

Existing One-stage SGG Models



(a) Object-Triplet Detection Models

(e.g., RelTR, SGTR)

(b) Triplet Detection Models

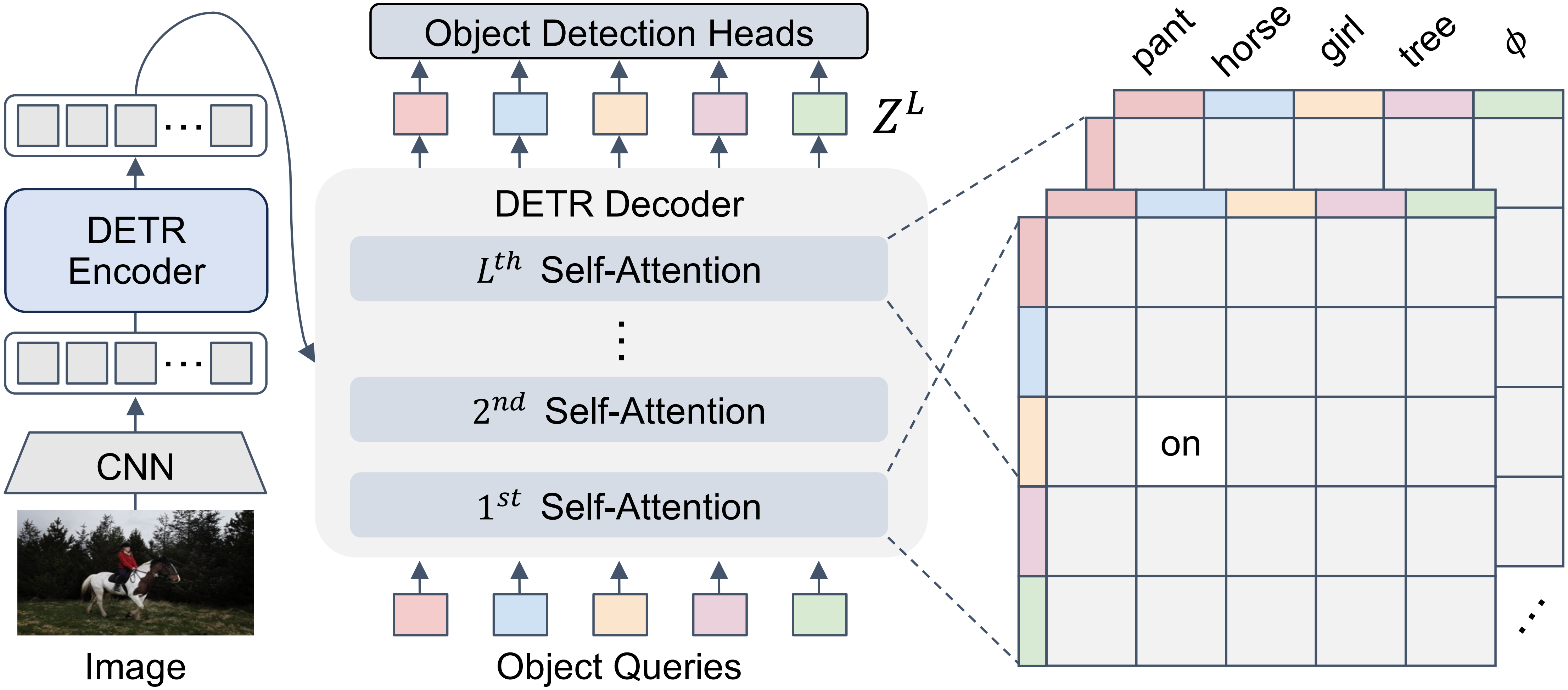
(e.g., Iterative SGG, Structured Sparse R-CNN)

(c) Relation Extraction Models

(e.g., Relationformer)

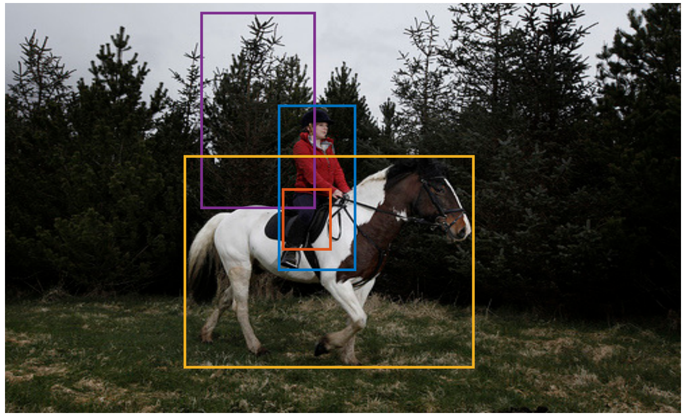
[ECCV 2022] Relationformer: A Unified Framework for Image-to-Graph Generation

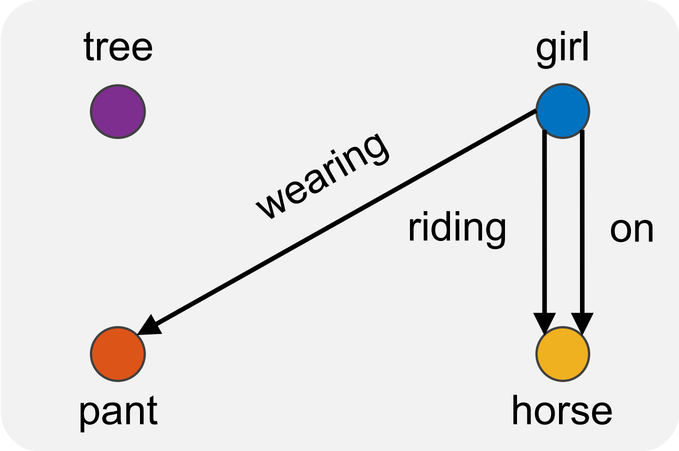
Motivation



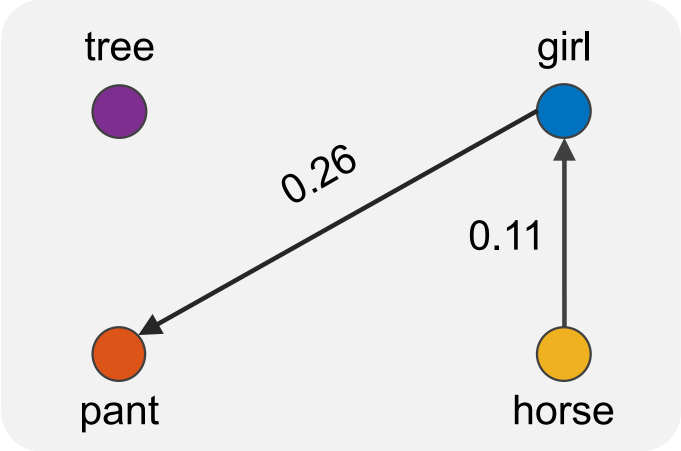
Object Detection Heads
Z^L
DETR Decoder
L^{th} Self-Attention
2^{nd} Self-Attention
1^{st} Self-Attention
Object Queries
DETR
Encoder
CNN
Image
pant
horse
girl
tree
ϕ
on

Motivation



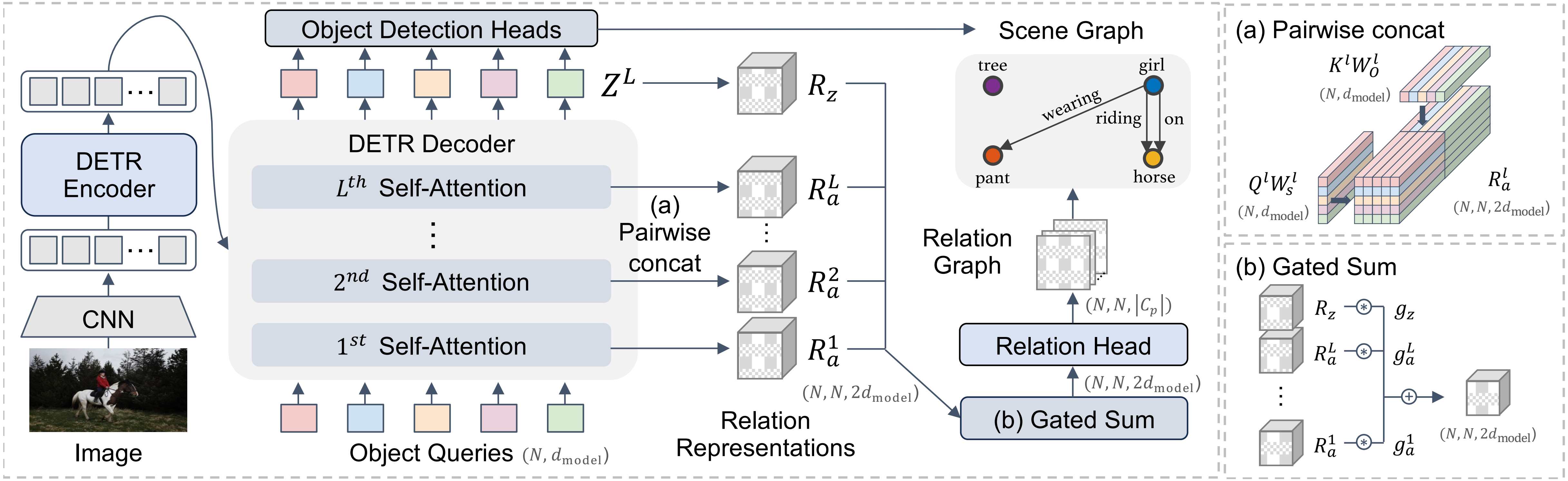


Scene Graph



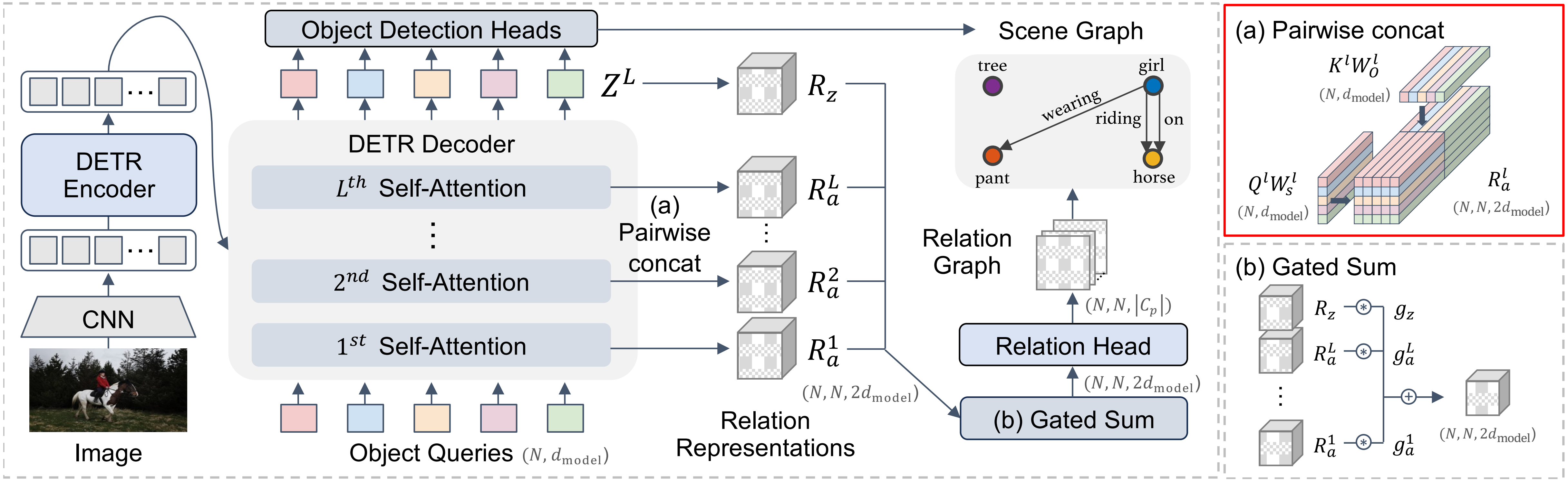
Attention Graph

Proposed Architecture



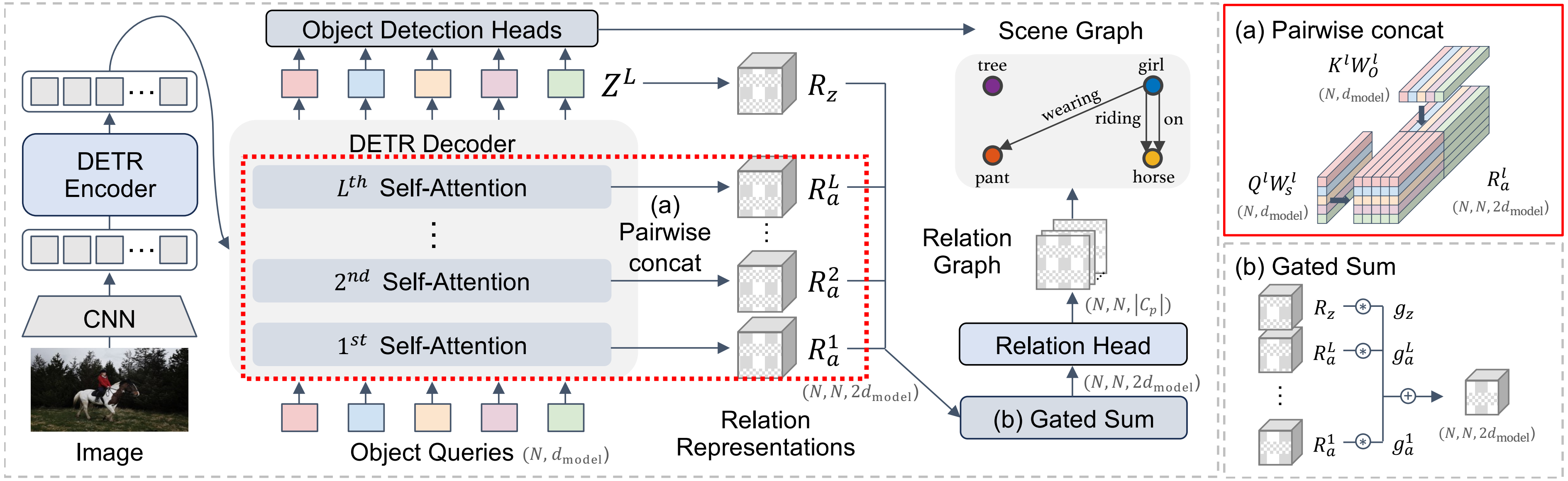
The overall architecture of EGTR

Proposed Architecture



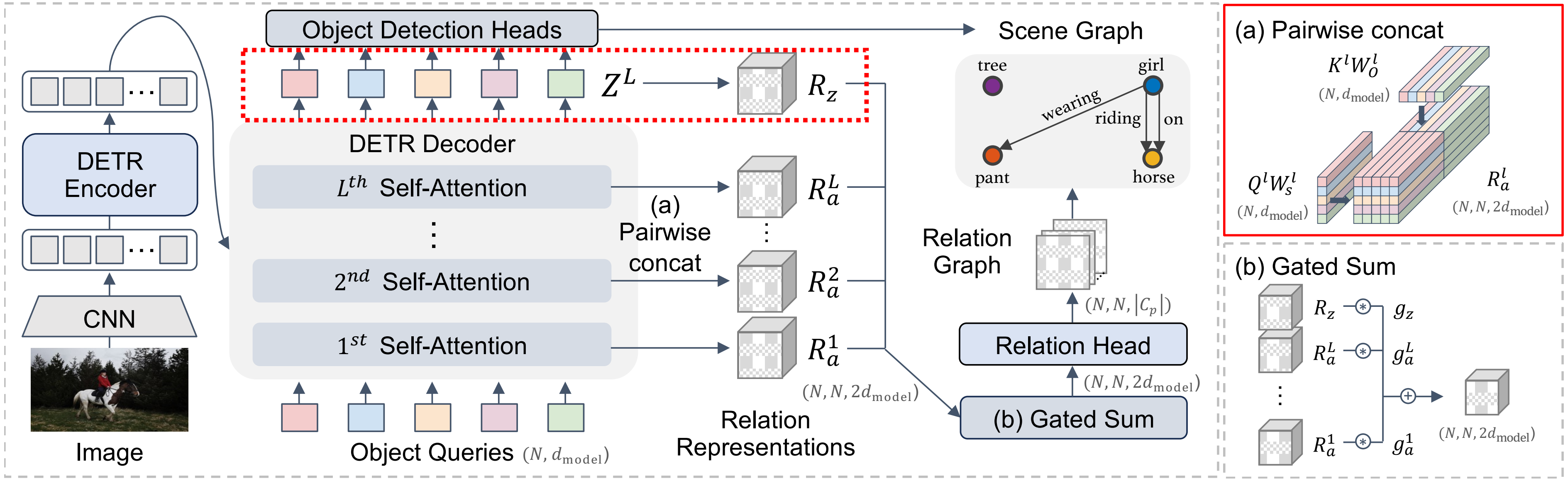
The overall architecture of EGTR

Proposed Architecture



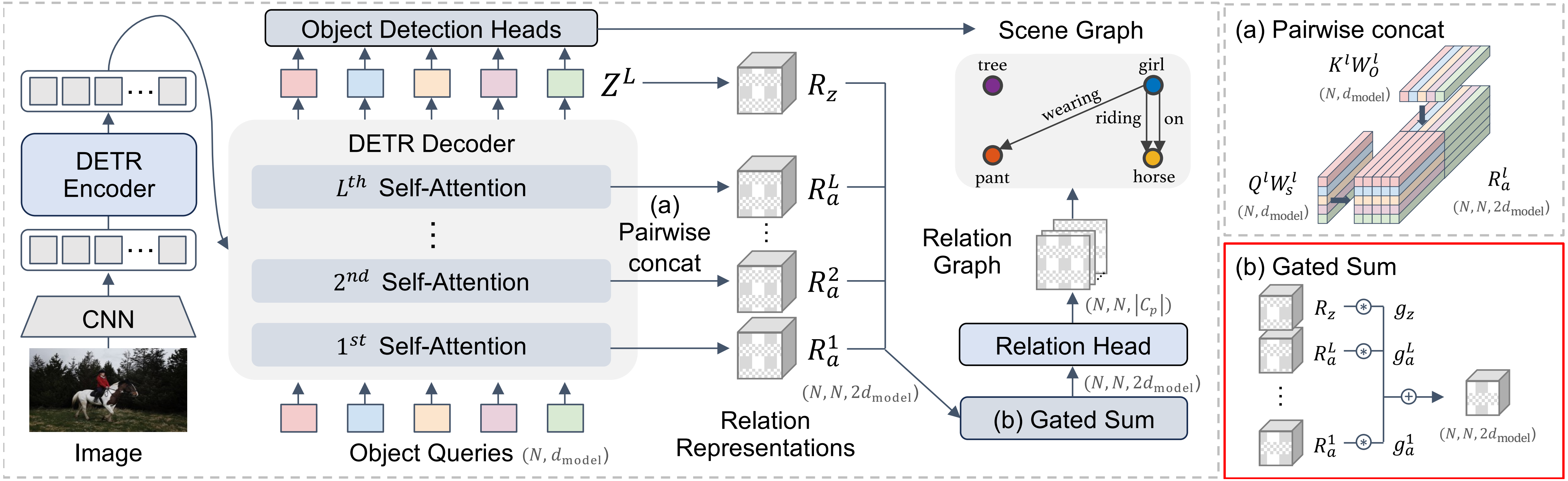
- $R_a^l \in R^{N \times N \times 2d_{\text{model}}} = [Q^l W_s^l; K^l W_o^l]$
  - $Q^l \in R^{N \times d_{\text{model}}}$: attention queries of the l-th layer
  - $K^l \in R^{N \times d_{\text{model}}}$: attention keys of the l-th layer

Proposed Architecture



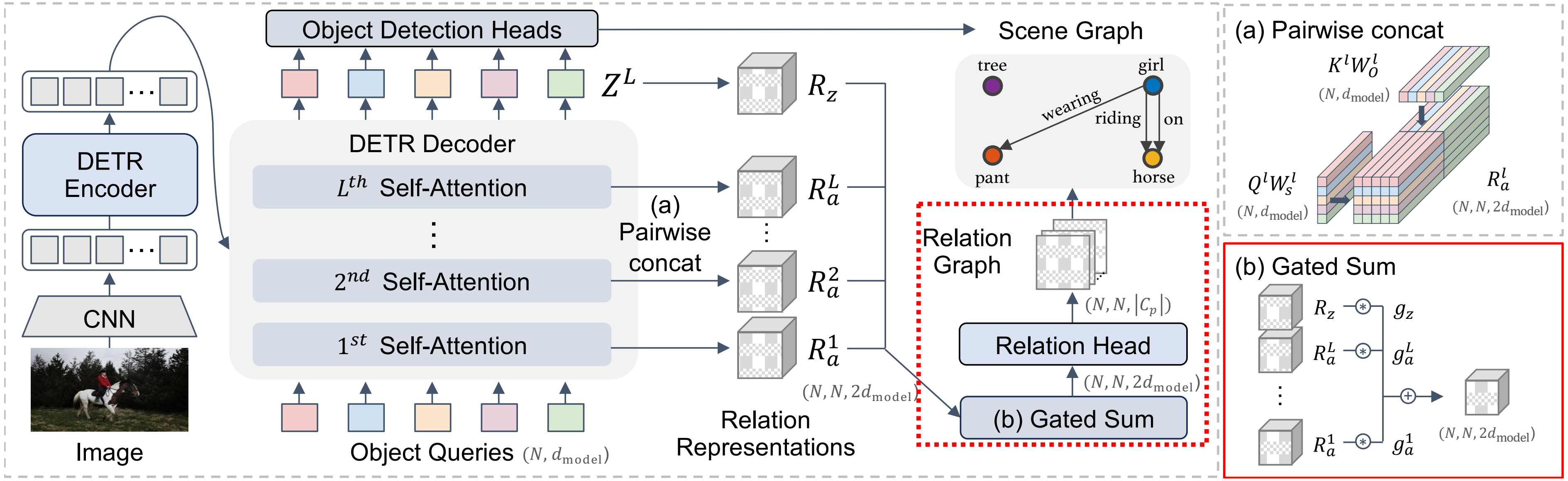
- $R_z \in R^{N \times N \times 2d_{\text{model}}} = [Z^l W_s; Z^l W_o]$
    - $Z^l \in R^{N \times d_{\text{model}}}$: the last layer representations of the object queries

Proposed Architecture



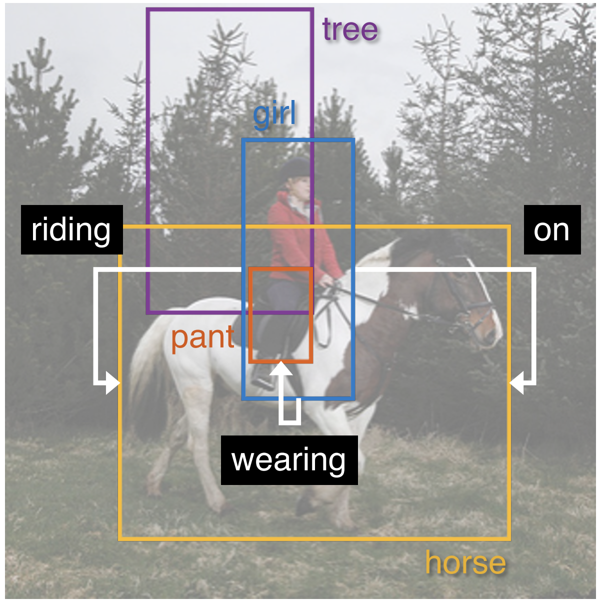
The overall architecture of EGTR

Proposed Architecture

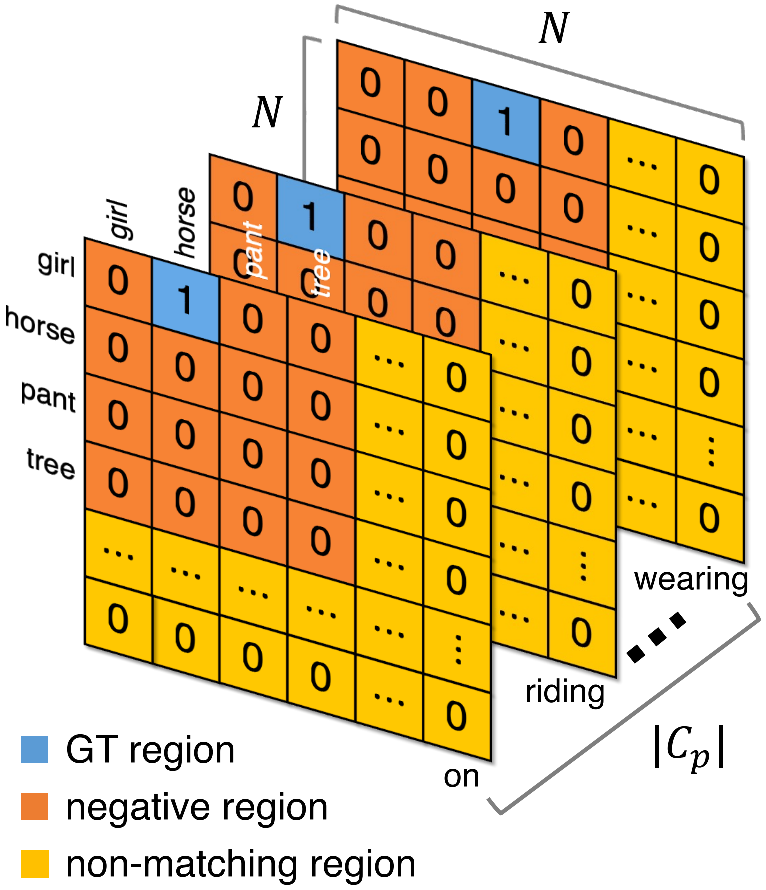


- $\hat{G} \in R^{N \times N \times |C_p|} = \sigma\left(\text{MLP}_{\text{rel}}\left(\sum_{l=1}^{L} g_a^l * R_a^l + g_z * R_z\right)\right)$
  - $g_a^l \in R^{N \times N \times 1} = \sigma(R_a^l W_G)$, $g_z \in R^{N \times N \times 1} = \sigma(R_z W_G)$
  - MLP_{rel}: a three-layer perceptron with ReLU activation

Proposed Techniques



(subject - predicate - object)
girl - on - horse
girl - riding - horse
girl - wearing - pant



Example of relation graph G

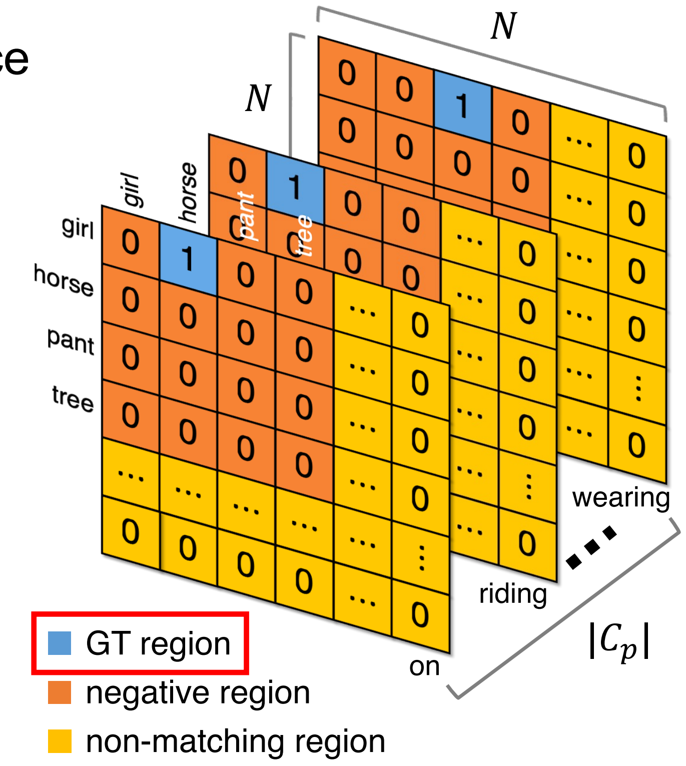
Proposed Techniques

(1) Adaptive smoothing

- Smooth the relation labels based on the object detection performance

$$u_i = \sigma\left(\mathrm{cost}_i - \mathrm{cost}_{\min} + \sigma^{-1}(\alpha)\right)$$

$$G_{ijk} = (1 - u_i)(1 - u_j)G_{ijk}$$



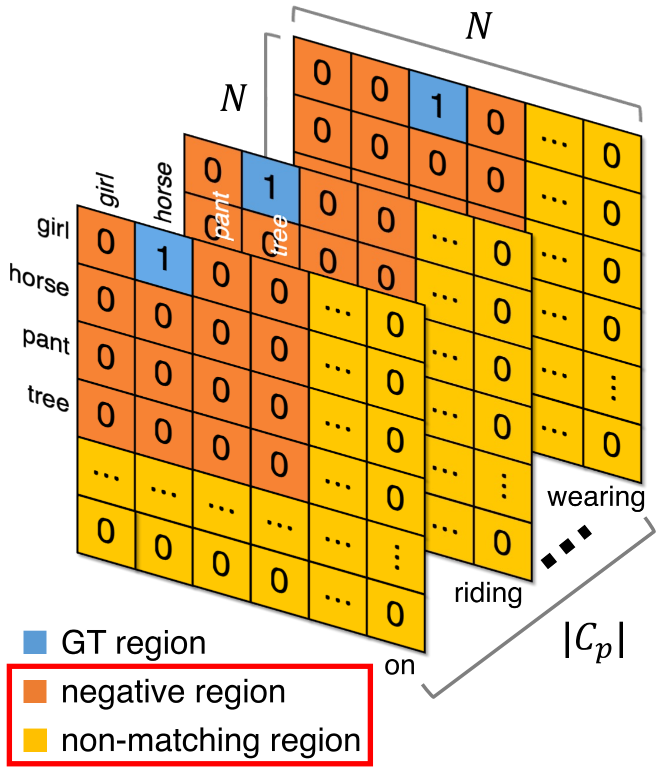
- u: uncertainty of each object query ($[\alpha, 1)$)
- cost: bipartite matching cost of each object query
- α: minimum uncertainty (hyper-parameter)
- G_{ijk}: k-th predicate category between subject entity v_i and object entity v_j

Proposed Techniques

(2) Sampling methodology

- Density is only 10^{-14} when N is set to 200 for Visual Genome
- Sample hard negatives & non-matchings
    - based on the predicted relation score
    - Choose the top $k_{\text{neg}} \times |\varepsilon|$ most challenging negatives
    - Choose the top $k_{\text{non}} \times |\varepsilon|$ most challenging non-matchings

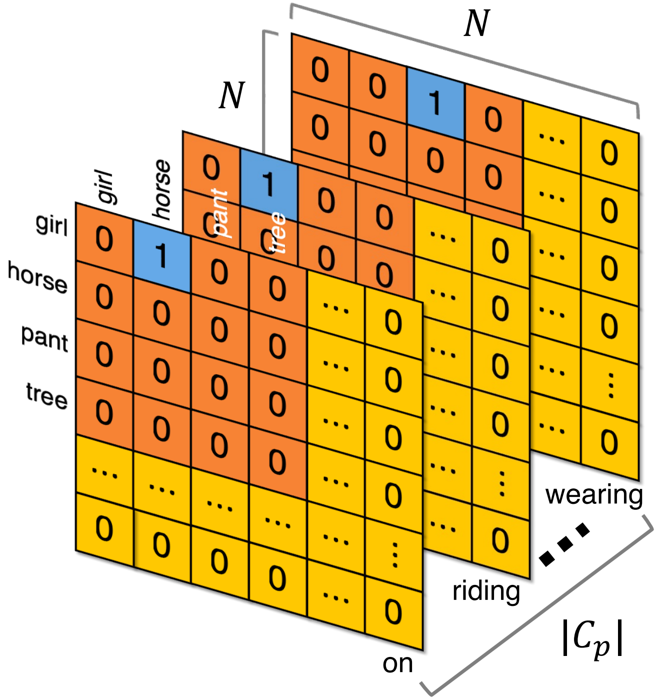
    ($|\varepsilon|$ denotes the number of the ground-truth edges)



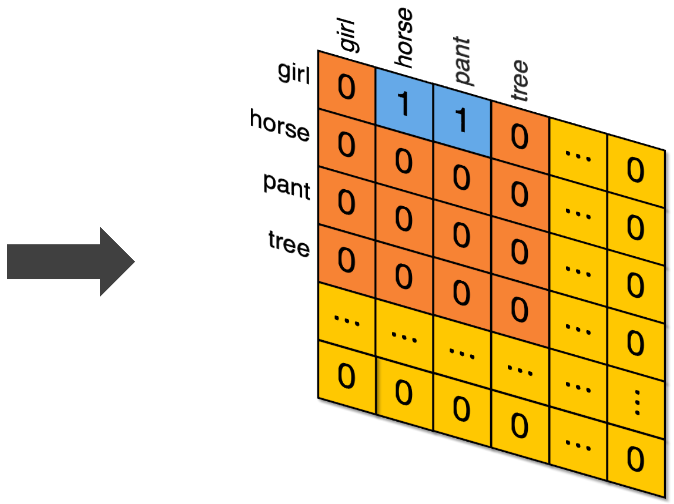
Proposed Techniques

(3) Connectivity prediction

- Auxiliary task for relation extraction

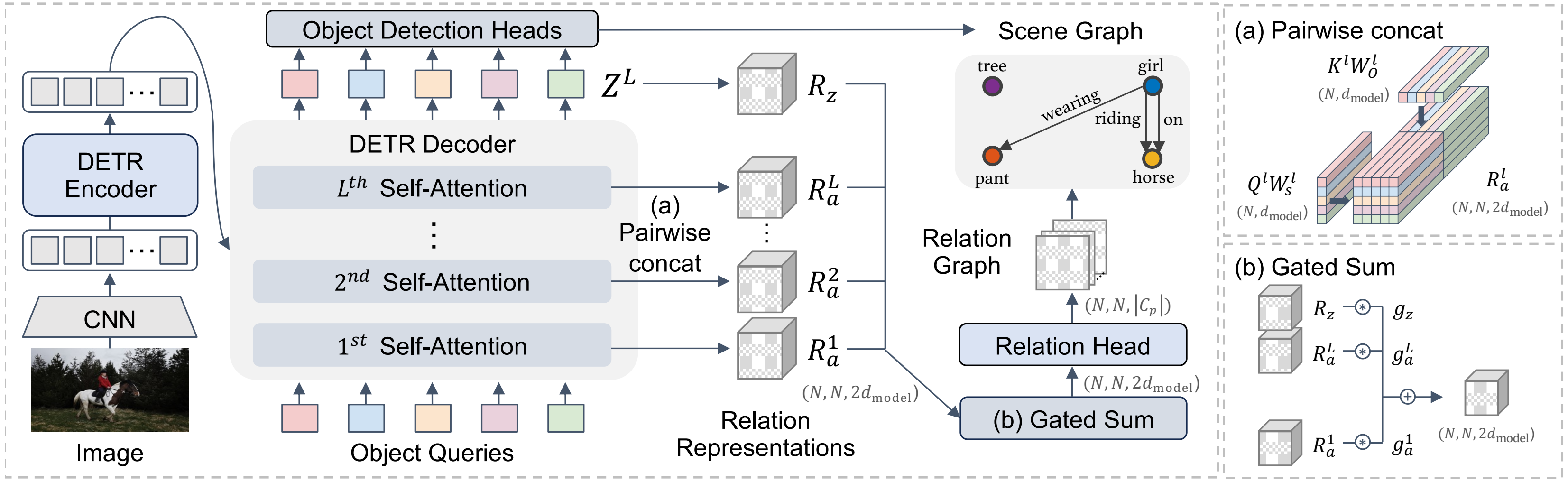


Relation graph G



Connectivity graph E

Multi-task Learning



$$\boxed{\mathcal{L} = \mathcal{L}_{\text{od}} + \lambda_{\text{rel}}\mathcal{L}_{\text{rel}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}}}$$

- $\mathcal{L}_{\text{od}}$: object detection loss (proposed in DETR)
- $\mathcal{L}_{\text{rel}}$: relation extraction loss (binary cross-entropy)
- $\mathcal{L}_{\text{con}}$: connectivity prediction loss (binary cross-entropy)

Datasets and Evaluation Settings

(1) Visual Genome: 150 object categories & 50 relation categories

- efficiency: # parameters & FPS
- object detection: AP50
- triplet detection
  - Recall@k (R@k): class agnostic measure
  - mean Recall@k (mR@k): aggregates the recalls for each predicate category

(2) Open Image V6: 601 object categories & 30 relation categories

- score: $0.2 \times$ micro-R@$50 + 0.4 \times \mathrm{wmAP_{rel}} + 0.4 \times \mathrm{wmAP_{phr}}$
  - micro-R@50
  - $\mathrm{wmAP_{rel}}$: predicting boxes of subject entity and object entity separately
  - $\mathrm{wmAP_{phr}}$: predicting a union box of subject entity and object entity

Implementation Details

- We employ Deformable DETR with ResNet-50 as a backbone
  - Deformable DETR improves the convergence speed of the DETR
  - Our approach can be extended to any object detector
    that incorporates self-attention mechanisms between object queries
- The number of object queries (N): 200
- Loss coefficients
  - λ_{rel}: 15
  - λ_{con}: 30 (Visual Genome) / 90 (Open Image V6)
- Smoothing minimum uncertainty (α): 10^{-14} / Sampling ratio ($k_{\mathrm{neg}} = k_{\mathrm{non}}$): 80

Quantitative Results

(1) Visual Genome

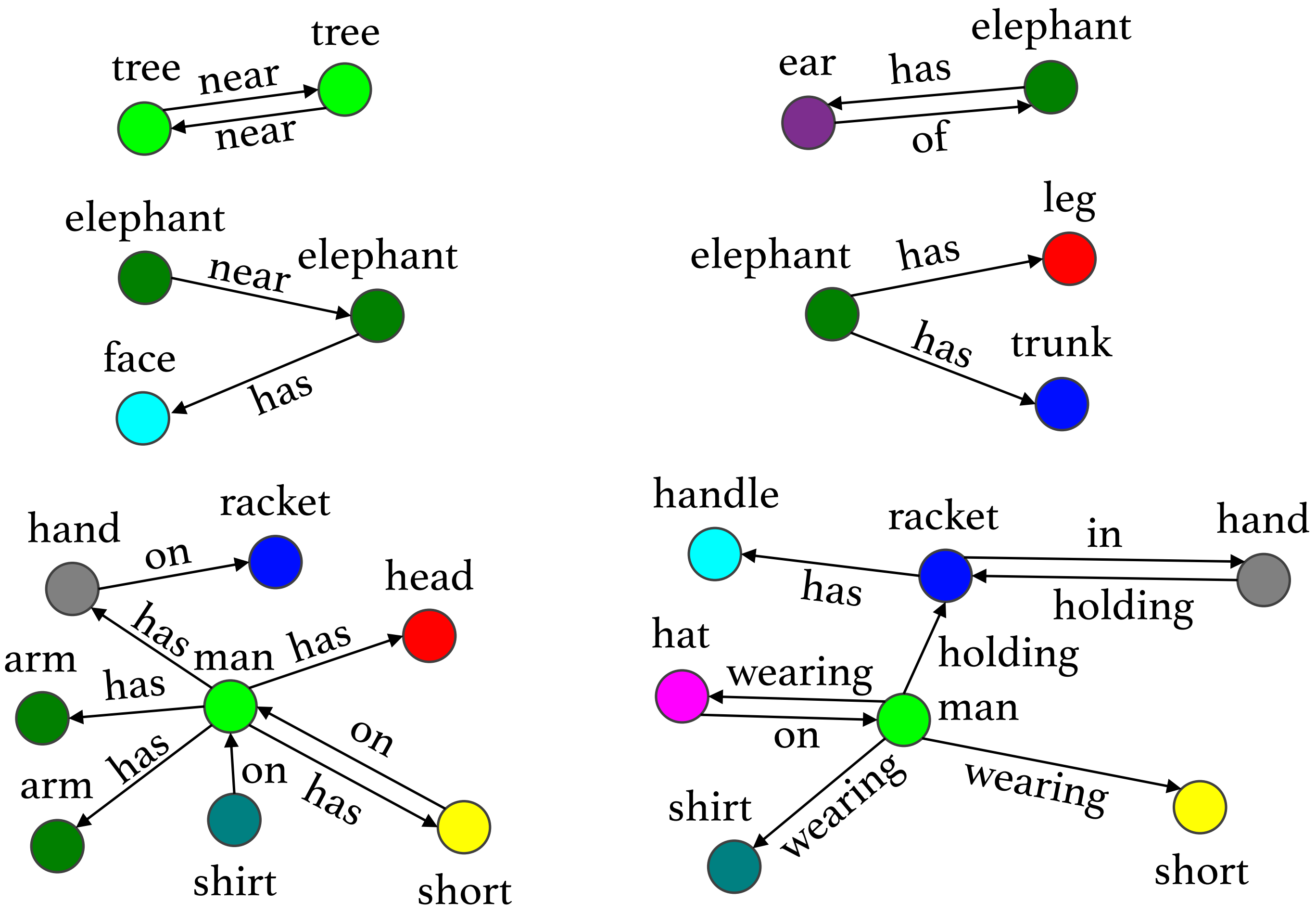
	Model	# params (M)	FPS	AP50	R@20	R@50	R@100	mR@20	mR@50	mR@100
two-stage	IMP (EBM) [34, 42]	*322.2*	*2.0*	28.1	18.1	25.9	31.2	2.8	4.2	5.4
	VTransE [47]	*312.3*	*3.5*	-	24.5	31.3	35.5	5.1	6.8	8.0
	Motifs [45]	*369.9*	*1.9*	28.1	25.1	32.1	36.9	4.1	5.5	6.8
	VCTree [36]	*361.5*	*0.8*	28.1	24.8	31.8	36.1	4.9	6.6	7.7
	VCTree (TDE) [36, 37]	*361.3*	*0.8*	28.1	14.0	19.4	23.2	6.9	9.3	11.1
	VCTree (EBM) [34, 36]	*372.5*	-	28.1	24.2	31.4	35.9	5.7	7.7	9.1
	GPS-Net [20]	-	-	-	-	31.1	35.9	-	6.7	8.6
	BGNN [16]	341.9	*1.7*	29.0	23.3	31.0	35.8	7.5	10.7	12.6
one-stage	FCSGG [21]	87.1	*6.0*	28.5	16.1	21.3	25.1	2.7	3.6	4.2
	RelTR [7]	63.7	*13.4*	26.4	21.2	27.5	-	6.8	10.8	-
	SGTR [17]	*117.1*	*6.2*	25.4	-	24.6	28.4	-	12.0	15.2
	Relationformer [32]	92.9	*8.5*	26.3	22.2	28.4	31.3	4.6	9.3	10.7
	Iterative SGG [9]	*93.5*	*6.0*	*27.7*†	-	29.7	32.1	-	8.0	8.8
	SSR-CNN [38]	*274.3*	*4.0*	*23.8*†	**25.8**	**32.7**	**36.9**	6.1	8.4	10.0
	SSR-CNN [38] $_{\mathrm{LA},\tau=0.3}$	*274.3*	*4.0*	*23.8*†	18.4	23.3	26.5	**13.5**	**17.9**	21.4
	EGTR (Ours)	**42.5**	**14.7**	**30.8**	23.5	30.2	34.3	5.5	7.9	10.1
	EGTR (Ours) $_{\mathrm{LA},\tau=0.7}$	**42.5**	**14.7**	**30.8**	15.7	18.7	20.5	12.1	17.8	**21.7**
	EGTR (Ours) $_{\mathrm{LA},\tau=0.5}$	**42.5**	**14.7**	**30.8**	19.7	24.2	26.7	11.0	17.1	21.4
	EGTR (Ours) $_{\mathrm{LA},\tau=0.3}$	**42.5**	**14.7**	**30.8**	22.4	28.2	31.7	8.8	14.0	18.3

Quantitative Results

(2) Open Image V6

Model	score	micro-R@50	$wmAP_{rel}$	$wmAP_{phr}$
Motifs [45]	38.9	71.6	29.9	31.6
VCTree [36]	40.2	74.1	34.2	33.1
GPS-Net [20]	41.7	74.8	32.9	34.0
BGNN [16]	42.1	75.0	33.5	34.2
RelTR [7]	43.0	71.7	34.2	37.5
SGTR [17]	42.3	59.9	37.0	38.7
SSR-CNN [38]	**49.4**	**76.7**	<u>41.5</u>	**43.6**
EGTR (Ours)	<u>48.6</u>	<u>75.0</u>	**42.0**	<u>41.9</u>

Qualitative Results



Ground Truth

Model Prediction

Analyses

(1) Ablation study – relation source

	R_a^l source	R_a^l	R_z	R@50	mR@50
①	Q^l & K^l	✓	✓	**30.2**	**7.9**
②	Z^l	✓	✓	29.6	7.4
③	-		✓	29.9	7.6
④	Q^l & K^l	✓		29.8	7.7

- ①: all attention layers & final hidden layer
- ②: all hidden layers
- ③: final hidden layer
- ④: all attention layers

Analyses

(1) Ablation study – pairwise function ($R_a^l = f(Q^l, K^l)$)

Pairwise function	# Params(M)	R@50	mR@50
dot product attention	**41.3**	25.9	6.2
dot product	**41.3**	27.4	6.8
Hadamard product	41.5	29.1	7.2
sum	41.5	29.5	7.3
concat	41.6	**29.9**	**7.9**

- dot product attention & dot product: $R^{N \times N \times (h \times 1)}$ (h denotes the number of self-attention heads)
- Hadamard product & sum: $R^{N \times N \times (h \times d_{\text{head}} = d_{\text{model}})}$
- concat: $R^{N \times N \times (h \times 2d_{\text{head}} = 2d_{\text{model}})}$

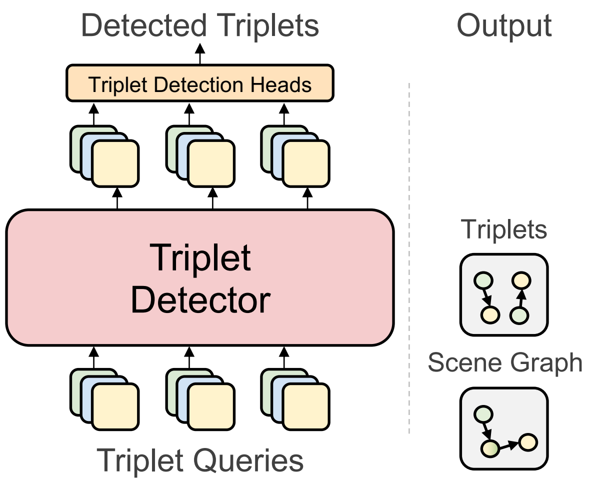
Analyses

(1) Ablation study – proposed techniques

adaptive smoothing	$\mathcal{L}_{\text{con}}$	sampling	R@50	mR@50
			26.6	5.3
✓			28.3	6.5
	✓		29.6	7.0
		✓	28.9	7.1
✓	✓	✓	**30.2**	**7.9**

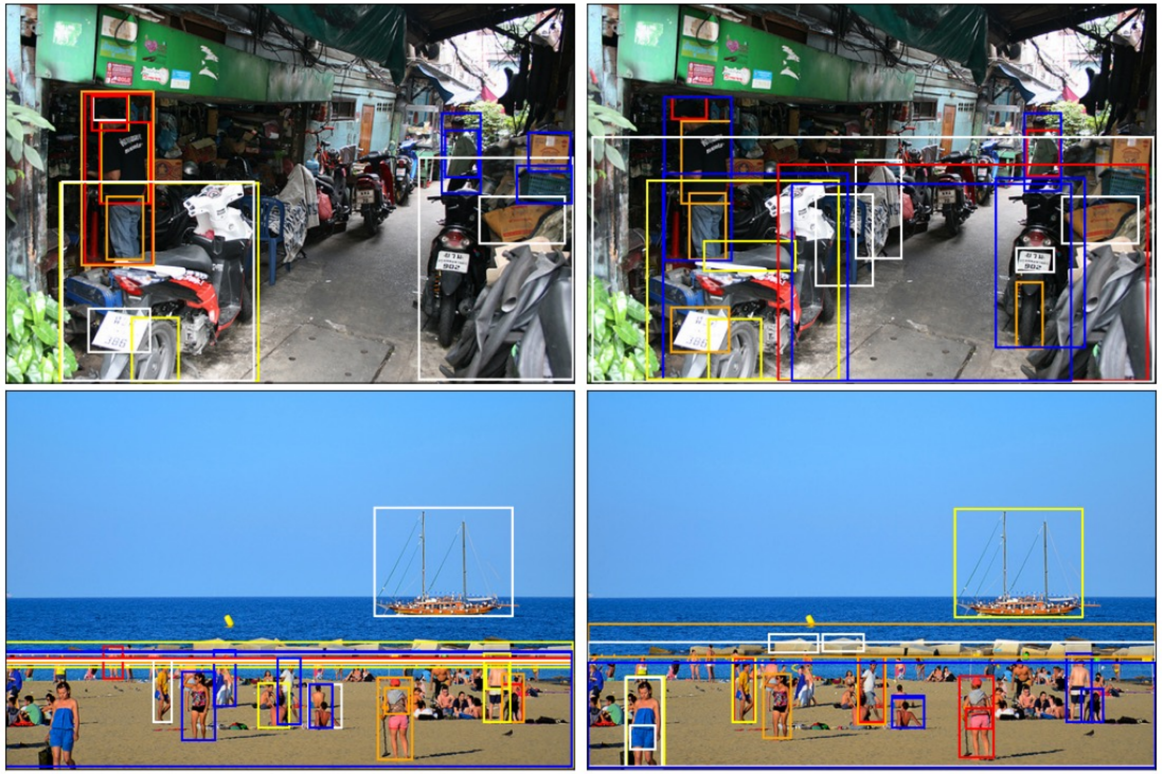
Analyses

(2) Object detection



Model	AP50	$AP50_{rel}$	$AP50_{no\text{-}rel}$
Iterative SGG [9]†	27.7	**24.3**	7.8
SSR-CNN [38]†	23.8	20.2	7.4
EGTR (Ours)	**30.8**	**24.3**	**10.7**

AP50 for two subsets of objects

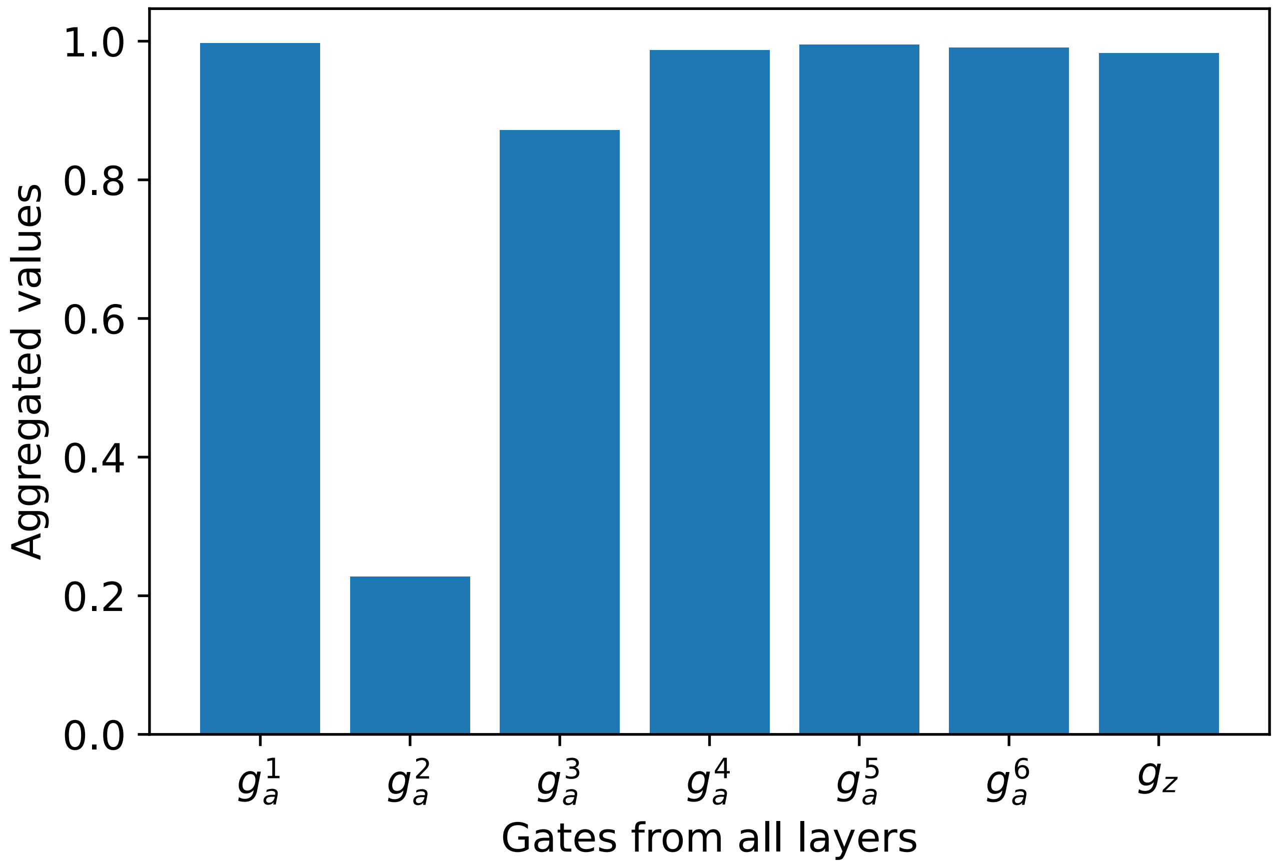


(a) SSR-CNN [38] (b) EGTR (Ours)

Detected subjects and objects

Analyses

(3) Gated sum



Conclusion

- **EGTR** that generates scene graphs efficiently and effectively by utilizing the *multi-head self-attention by-products* from the object detector
- **Adaptive smoothing** that helps *multi-task learning* of object detection and relation extraction
- **Connectivity prediction** as an *auxiliary* task of relation extraction
- The highest object detection performance and competitive triplet detection capabilities with **the highest efficiency**

Thank you

Poster Session

17:15 ~ 18:45

Arch 4A-E Poster #408

Email: jinbae.im@navercorp.com

Github: https://github.com/naver-ai/egtr