

# Performers [Choromanski+ (Google), ICLR21]

✖ Attention行列を線形のメモリ複雑度で格納

✖ Fast Attention Via positive Orthogonal Random features (FAVOR+)

✖ Q, Kを低ランクのQ', K'に近似

✖ Q(KV)の順で計算

$$\mathbf{A}(i, j) = \mathbf{K}(\mathbf{q}_i^\top, \mathbf{k}_j^\top) \quad \mathbf{K}(\mathbf{x}, \mathbf{y}) = \mathbb{E}[\phi(\mathbf{x})^\top \phi(\mathbf{y})]$$

$$\phi(\mathbf{x}) = \frac{h(\mathbf{x})}{\sqrt{m}} (f_1(\omega_1^\top \mathbf{x}), \dots, f_1(\omega_m^\top \mathbf{x}), \dots, f_l(\omega_1^\top \mathbf{x}), \dots, f_l(\omega_m^\top \mathbf{x}))$$

deterministic vectors  $\omega_i$  or  $\omega_1, \dots, \omega_m \stackrel{\text{iid}}{\sim} \mathcal{D}$  for some distribution  $\mathcal{D} \in \mathcal{P}(\mathbb{R}^d)$

$$\widehat{\text{Att}}_{\leftrightarrow}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \widehat{\mathbf{D}}^{-1}(\mathbf{Q}'((\mathbf{K}')^\top \mathbf{V})),$$

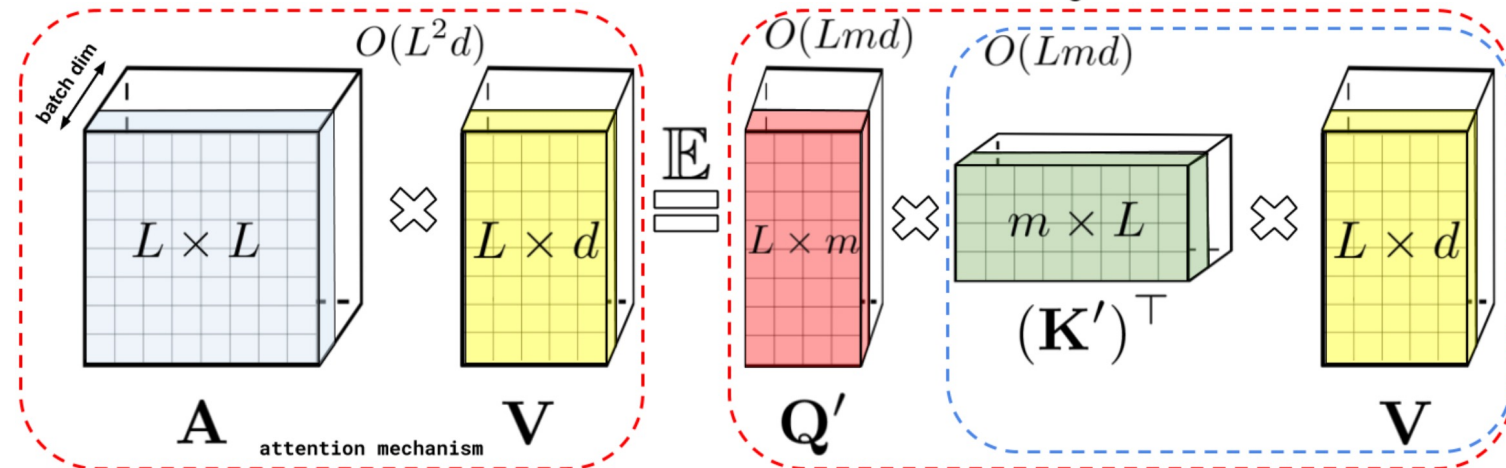
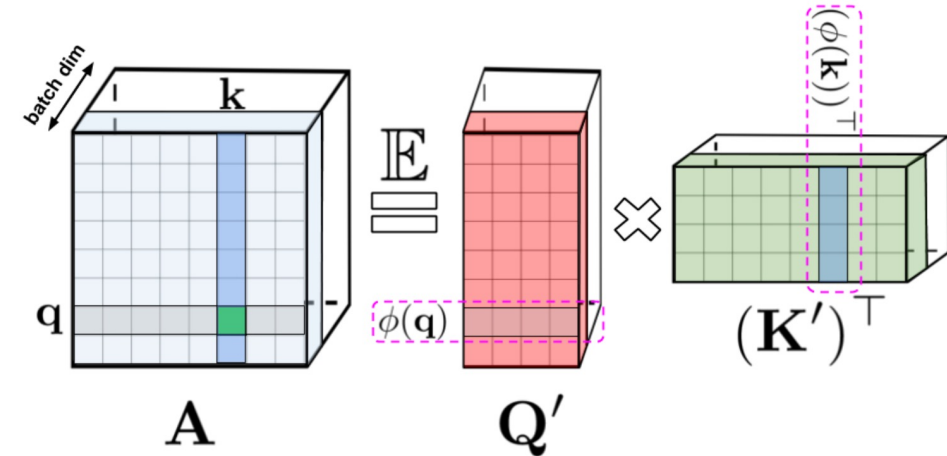
$$\widehat{\mathbf{D}} = \text{diag}(\mathbf{Q}'((\mathbf{K}')^\top \mathbf{1}_L))$$

✖ メモリ空間

✖  $O(L^2 + Ld) \rightarrow O(Lm + Ld + md)$

✖ 計算時間

✖  $O(L^2d) \rightarrow O(Lmd)$



# TAP [Yang+ (Microsoft), CVPR21]

✖ Text-VQAやText-Captionのため、シーンテキストを事前学習

✖ テキスト $w$ 、オブジェクト画像 $v^{obj}$ 、シーンテキスト画像 $v^{ocr}$   
開始トークン $P_0$

✖  $w = [w^q(\text{質問文}), w^{obj}(\text{オブジェクトのラベル}), w^{ocr}(\text{シーンテキスト})]$

✖ 物体検出はFaster R-CNN、シーンテキスト検出はOCRを使用

✖ Fusion Module

✖ 各埋め込みを連結してTransformerへ入力

✖ Scene-text language pre-training tasks

✖ MLM: maskされた $w$ をその位置の $f^w$ から復元

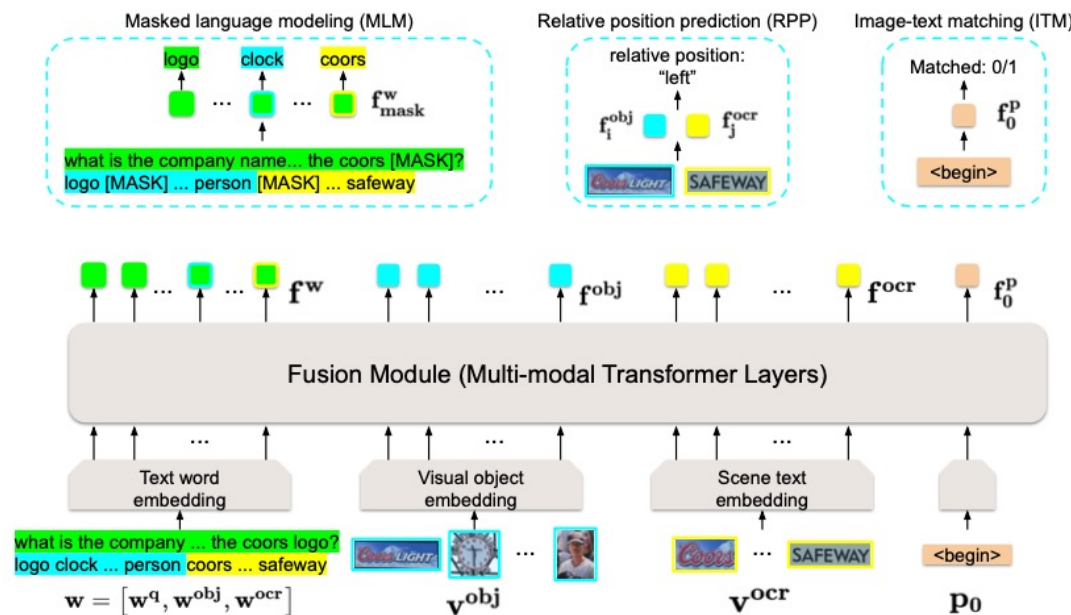
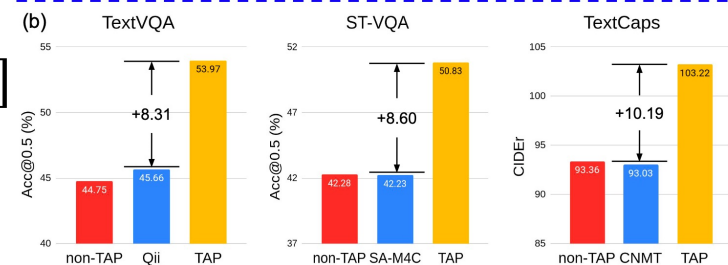
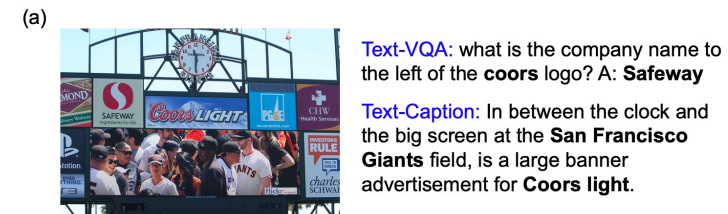
✖ ITM:  $f_0^P$ を入力として $w$ が他の画像のものか判断

✖  $w^q$ のみに適応したものは殆ど効果なし

✖ Scene-text visual pre-training tasks

✖ RPP:  $f^{obj}$ と $f^{ocr}$ から相対的な空間的位置を予測

✖ on、cover、overlap、unrelatedなど



# NS-VQA [Tenenbaum+ (MIT CSAIL), NeurIPS18]

## ✖ 画像認識や言語理解と、推論のためのsymbolic program executionを融合

- ✖ データセット CLEVR で99.8%の精度
- ✖ 推論時のメモリコストをSoTAから99%削減

## ✖ Scene Parser

- ✖ Mask R-CNNで全オブジェクトのSegmentを生成
- ✖ 色、素材、サイズ、形状などのラベルも予測
- ✖ 元画像と対になりResNet-34に送られ、3D座標などを抽出

## ✖ Question parser

- ✖ 質問をプログラムに変換
- ✖ bi-LSTMのEncoder-Decoder
- ✖  $\Phi_E, \Phi_D$ : Word Embedding

## ✖ Program executor

- ✖ プログラムを実行し答えを取得

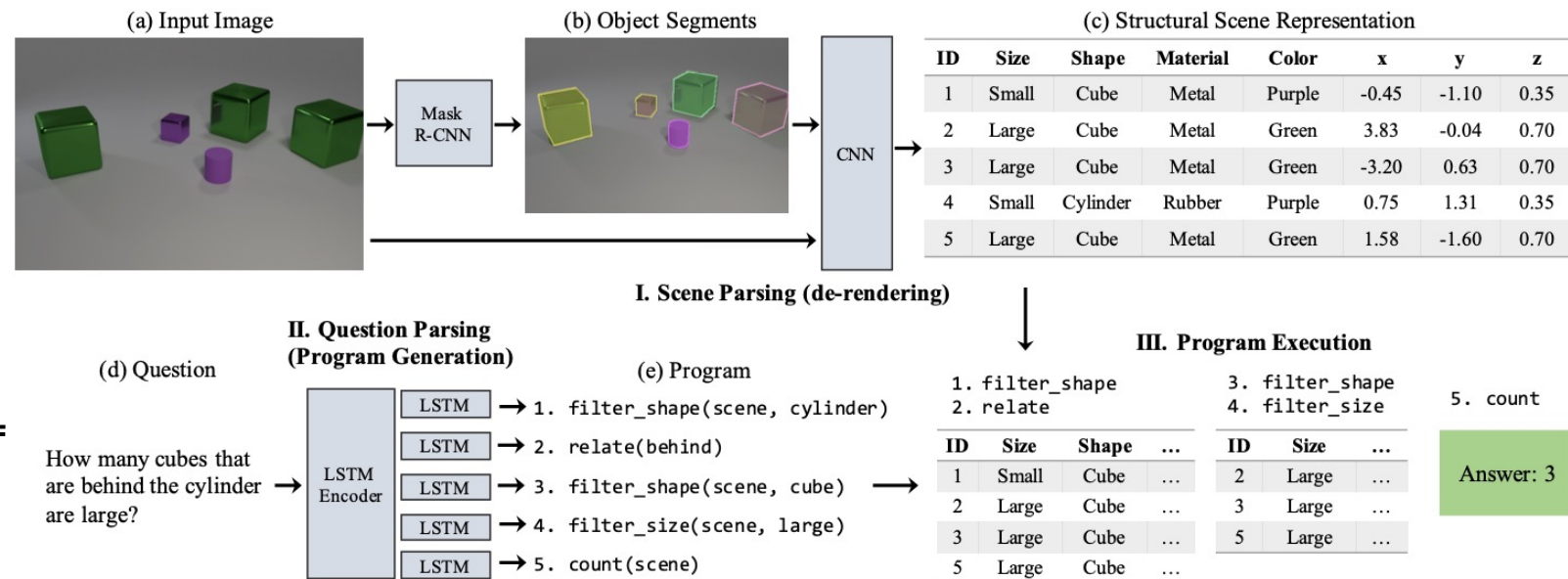
Question parser

$$e_i = [e_i^F, e_i^B], \text{ where } e_i^F, h_i^F = \text{LSTM}(\Phi_E(x_i), h_{i-1}^F)$$

$$e_i^B, h_i^B = \text{LSTM}(\Phi_E(x_i), h_{i+1}^B)$$

$$q_t = \text{LSTM}(\Phi_D(y_{t-1})), \alpha_{ti} \propto \exp(q_t^\top W_A e_i), c_t = \sum_i \alpha_{ti} e_i$$

$$y_t \sim \text{softmax}(W_O[q_t, c_t])$$





# HULANet [Wu+ (University of Massachusetts Amherst), CVPR20]

## PhraseCut Task

- ✗ 自然言語を用いて画像領域を分離するタスク
- ✗ データセット: VGPHRASECUTを使用
  - ✗ フレーズが **Category**、**Attribute**、**Relation**を示す単語で構成

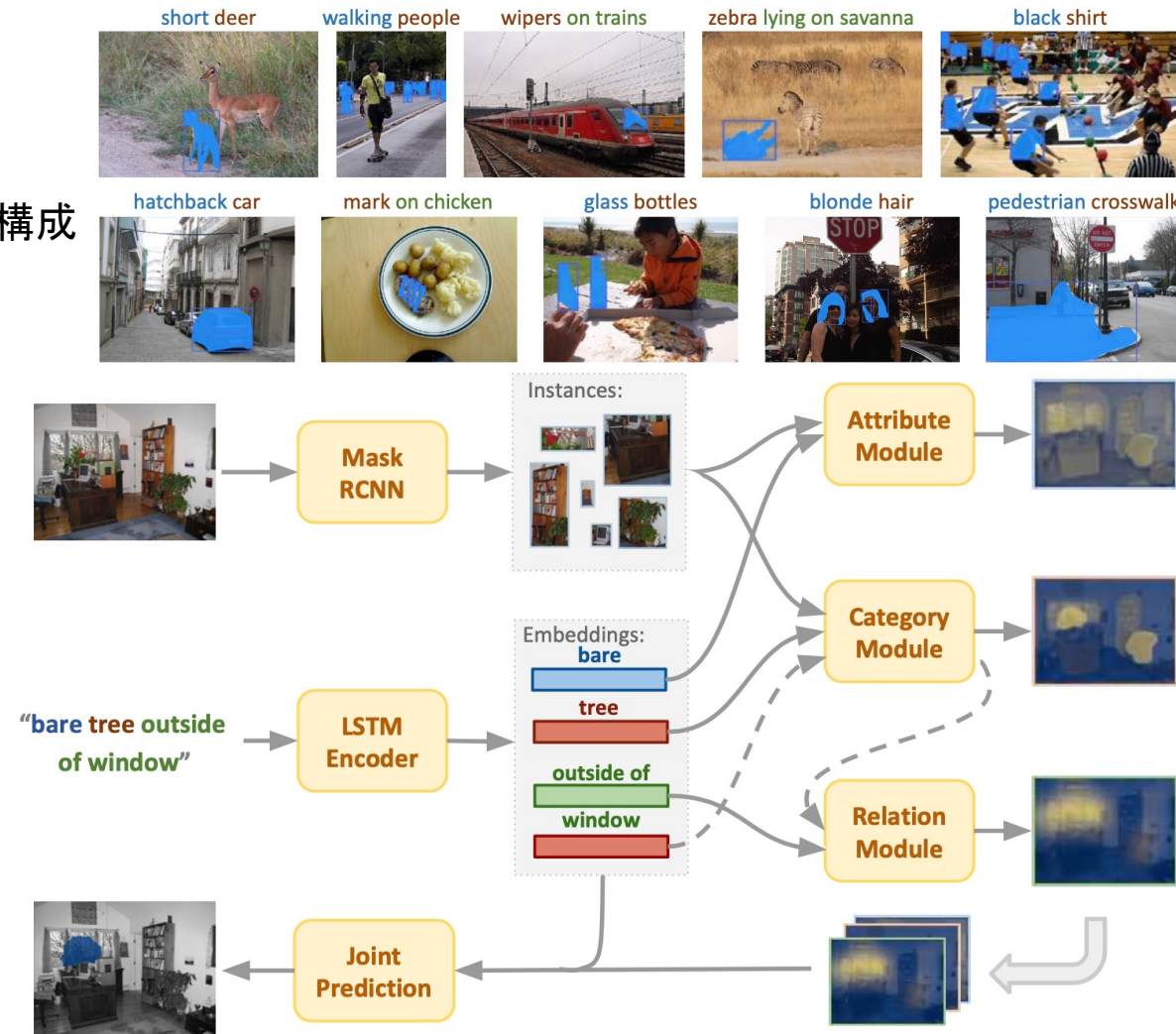
## HULANet: PhraseCutでSoTA

- ✗ Category module: ヒートマップ  $P$  を出力

$$P_{H \times W} = \sigma(a \cdot S_{H \times W} + b)$$

$$A = \sigma(f(e_{cat})) \quad S_{H \times W} = \sum_c A_c \cdot C_c$$

- ✗  $e_{cat}$  はフレーズの埋め込み
- ✗  $C$  はMask-RCNNで各セグメントに対するスコア
- ✗ Attribute module
  - ✗ Category moduleに分類器を追加
- ✗ Relation Module
  - ✗ カーネルサイズ7のCNN\*2を用いてscoresを取得
  - ✗ 各空間の関係に対応するフィルタを学習
- ✗ 最終的な出力  $O = \sum_t F_t w_t$ 
  - ✗ 各スコア  $P_c, P_a, P_r$ 、要素ごとの積  $P_i \circ P_j$ 、バイアスによる10チャンネルのF
  - ✗ 各フレーズの埋め込みを連結し、線形層と正規化を経た10次元の重み  $w$



# JRNet [Jain+ (IIT Hyderabad), 21]

✖ RESの3つのベンチマークでSoTA

✖ Joint Reasoning Module (JRM)

✖  $V^p, L^p$ : 各特徴量にPositional Encodingを加算

$$M = V^p \odot L^p$$

$$Z = \text{MultiHead}(M)$$

$$F = \text{MLP}(\text{LayerNorm}(Z))$$

✖ Cross Modal Multi-Level Fusion (CMMLF)

✖ 要素ごとの積 : チャネル方向に結合

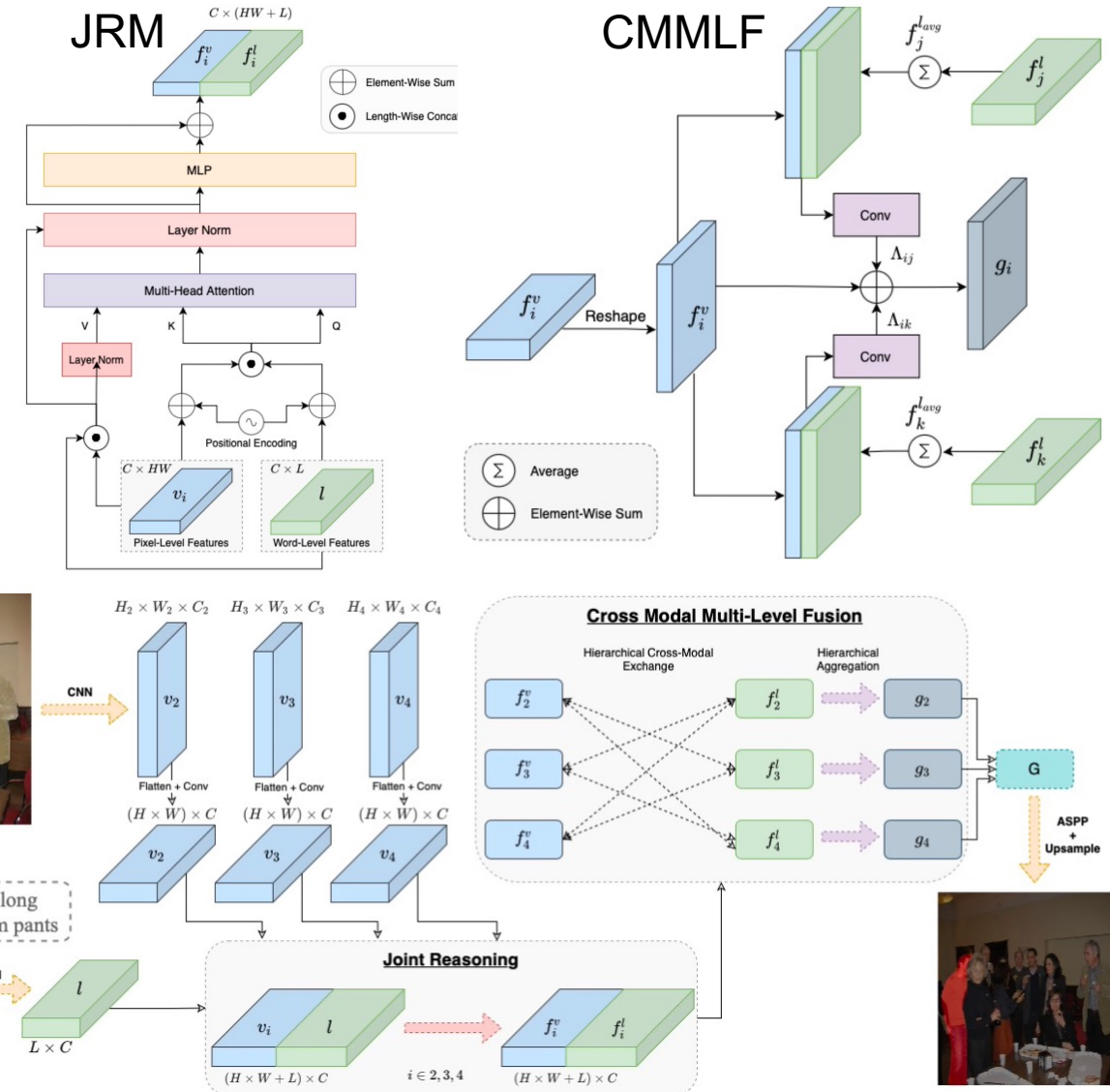
$$\Lambda_{ij} = \sigma(\text{Conv}([f_i^v; f_j^{l_{avg}}]))$$

$$g_i = f_i^v + \sum_{j \neq i} \Lambda_{ij} \circ f_i^v$$

$$G = \text{Conv3D}([g_2; g_3; g_4])$$

✖  $G$ はAtrous Spatial Pyramid Pooling (ASPP)デコーダとUp-samplingを経てmask  $S$ を予測

✖ Lossはbinary cross-entropy





# One-Stage Approach to Visual Grounding [Yang+ (Univ. of Rochester), ICCV21]

- ✖ テキストの埋め込みをYOLOv3に融合させた1ステージモデル
  - ✖ Phrase Localization、RECにおいて精度と推論速度を両立
    - ✖ 2ステージ(bboxの候補を予測した後、テキストとの類似性により選択)のモデルの10倍の推論速度

- ✖ 画像特徴抽出: Darknet-53、Feature Pyramid Network

- ✖  $8 \times 8 \times D_1$ 、 $16 \times 16 \times D_2$ 、 $32 \times 32 \times D_3$ の解像度
- ✖  $1 \times 1 \text{conv}$ で $D=512$ に統一

- ✖ 言語特徴抽出: BERT+全結合層 ( $D=512$ )

- ✖ Spatial feature encoding

- ✖  $\left( \frac{i}{W'}, \frac{j}{H'}, \frac{i+0.5}{W'}, \frac{j+0.5}{H'}, \frac{i+1}{W'}, \frac{j+1}{H'}, \frac{1}{W'}, \frac{1}{H'} \right)$
- ✖ 各 $i, j$ に対する左上、中央、右下の座標
- ✖  $W', H'$ は8/16/32

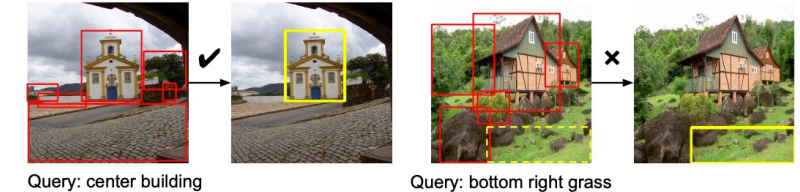
- ✖ 各 $i, j$ に言語特徴量をブロードキャストし画像特徴量と結合

- ✖  $1 \times 1 \text{conv}$ で $512+512+8 \rightarrow 512$ 次元

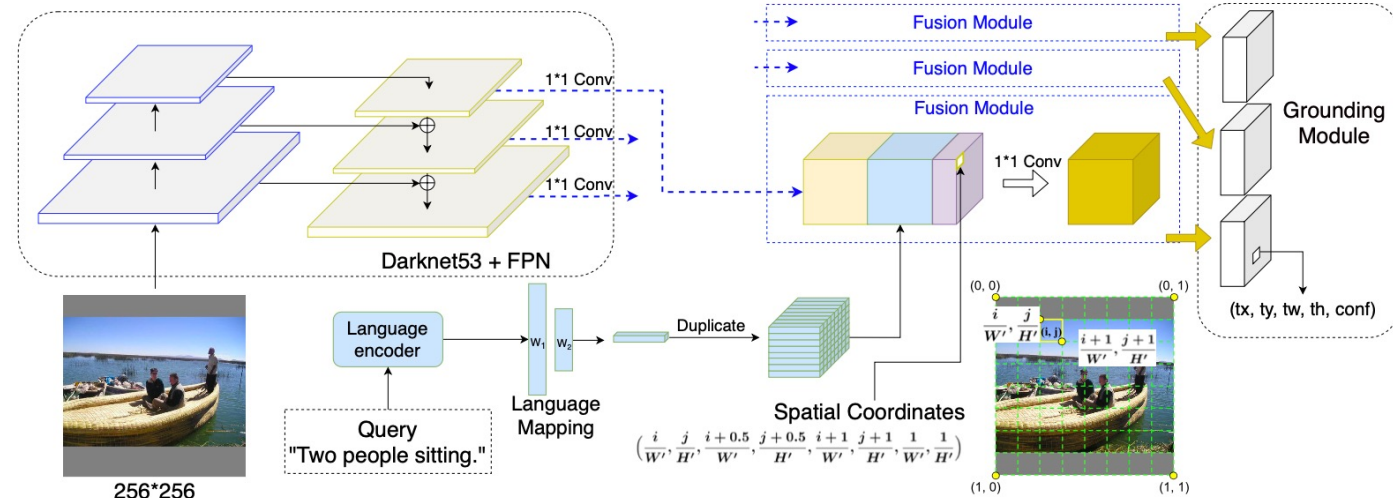
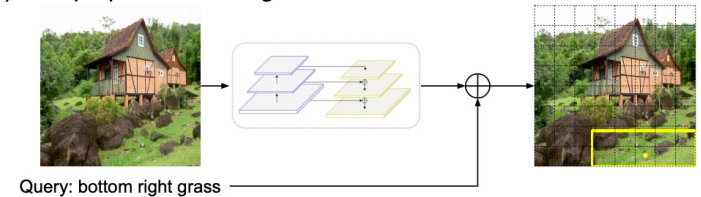
- ✖ 各 $i, j$ に対し3つのアンカーボックスを用意

- ✖  $(8 \times 8 + 16 \times 16 + 32 \times 32) \times 3 = 4032$ 個のボックスに対しSoftmaxを使用してConfidenceを予測
  - ✖ 損失関数は、このSoftmaxとone-hotベクトルとの間のcross entropyを使用

(a). Two-stage visual grounding



(b). The proposed one-stage method



# SSTVOS [Duke+ (Univ. of Toronto), CVPR21]

✖ TransformerをベースとしたVideo Object Segmentationの手法

✖ 入力: 長さ $T$ 、サイズ $H \times W$ のRGBフレーム  $\mathbf{S} \in \mathbb{R}^{3 \times T \times H \times W}$

✖  $f$ : ResNetで特徴抽出  $\mathbf{T} = f(\mathbf{S}) \quad \mathbf{T} \in \mathbb{R}^{C \times T \times H' \times W'}$

✖  $\tau$ 個のフレームの埋め込み $\mathbf{T}_\tau$ とPositional Encoding:  $\mathbf{P}$ を加算  $\tilde{\mathbf{T}} = \mathbf{T}_\tau + \mathbf{P}$   
 $\mathbf{P} \in \mathbb{R}^{C \times T \times H' \times W'}$

✖  $L$ 層のTransformerに入力  $\tilde{\mathbf{T}}_L = g_L \circ g_{L-1} \circ \dots \circ g_1(\tilde{\mathbf{T}})$

✖  $g$ はmulti-head attention

✖ Sparse Attentionを使用  $\text{SparseAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\mathbf{p}} = \text{softmax}(\mathbf{Q}_{\mathbf{p}} \mathbf{K}_{I_{\mathbf{p}}}^T) \mathbf{V}_{I_{\mathbf{p}}}$

✖  $I_{\mathbf{p}}$ は座標 $(i, j, k)$ のセット

✖ どのピクセル同士がAttentionするか決定

✖ Attention Mapをmax pooling

$$\mathbf{A}^l \in \mathbb{R}^{|I_{\mathbf{p}}| \times \tau \times H' \times W'} \quad \mathbf{A}_v^l(\mathbf{p}) = \max_{I_{\mathbf{p}}^o \cup \{0\}} \mathbf{A}^l$$

✖ Decoder

✖ Segmentation NetworkであるCFBIを使用

