

**個別ゼミ**

**2025年10月29日**

# 議題

今のモデル

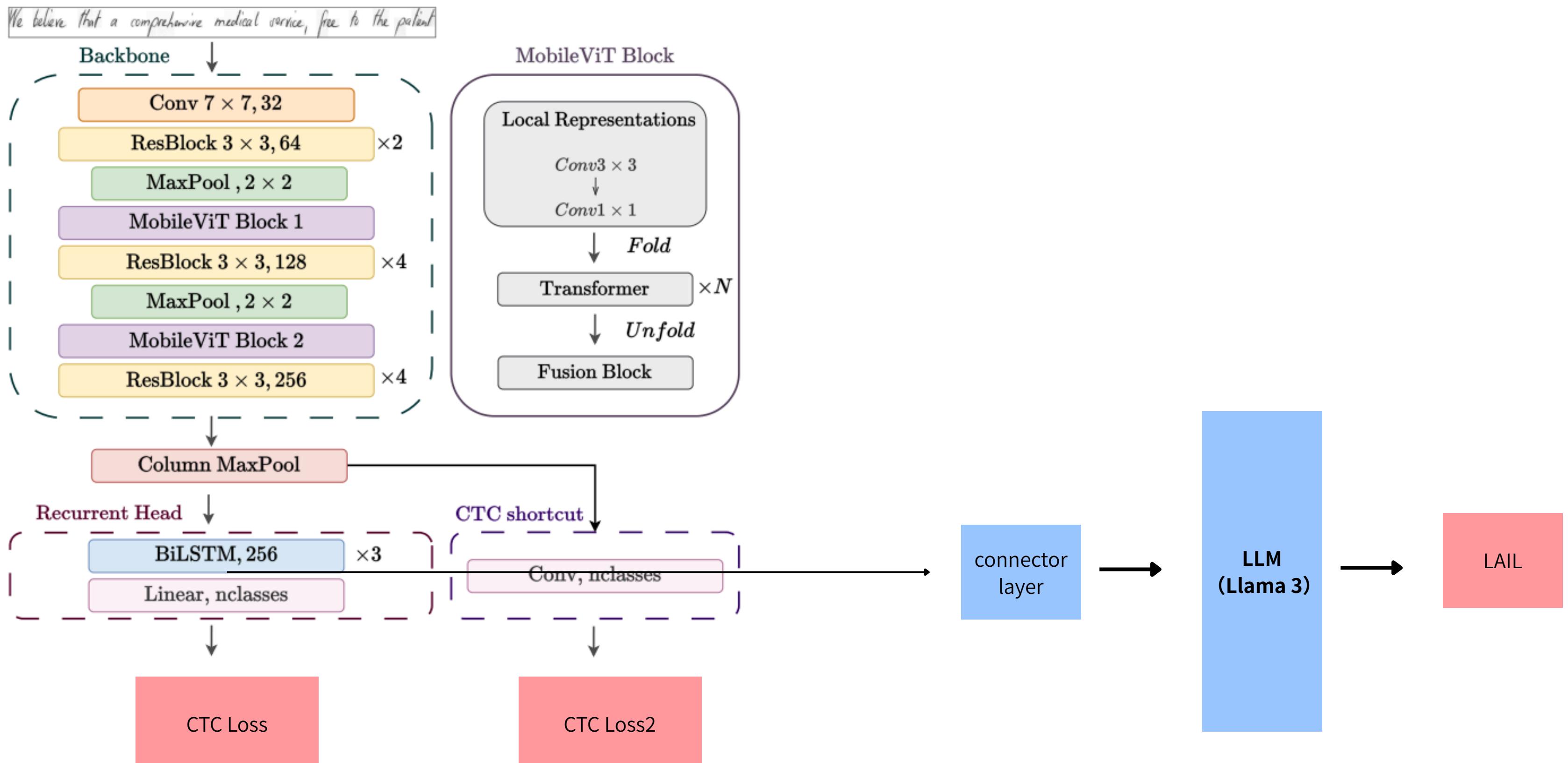
先週の内容

実装で追加内容

予稿モデル図

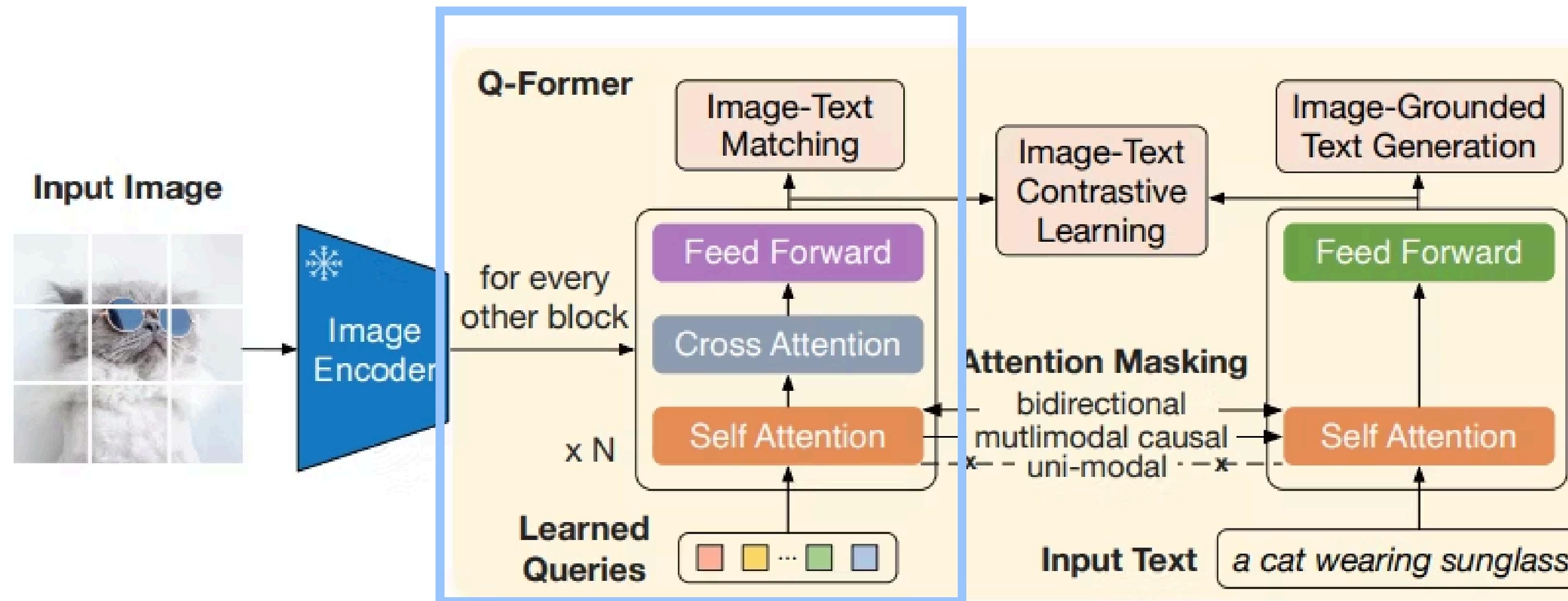
現状の結果

# 今のモデル



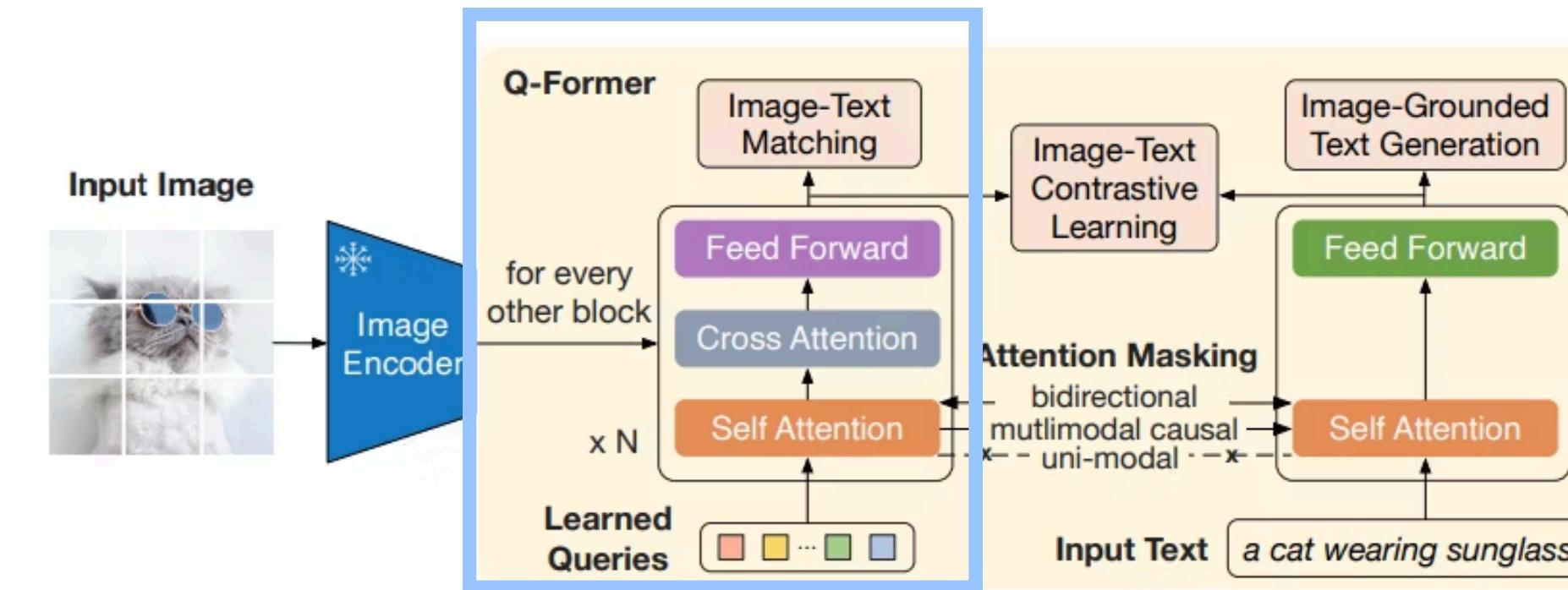
# 実装で追加内容

BLIP2



# 実装で追加内容

以下の層の追加  
青四角を追加

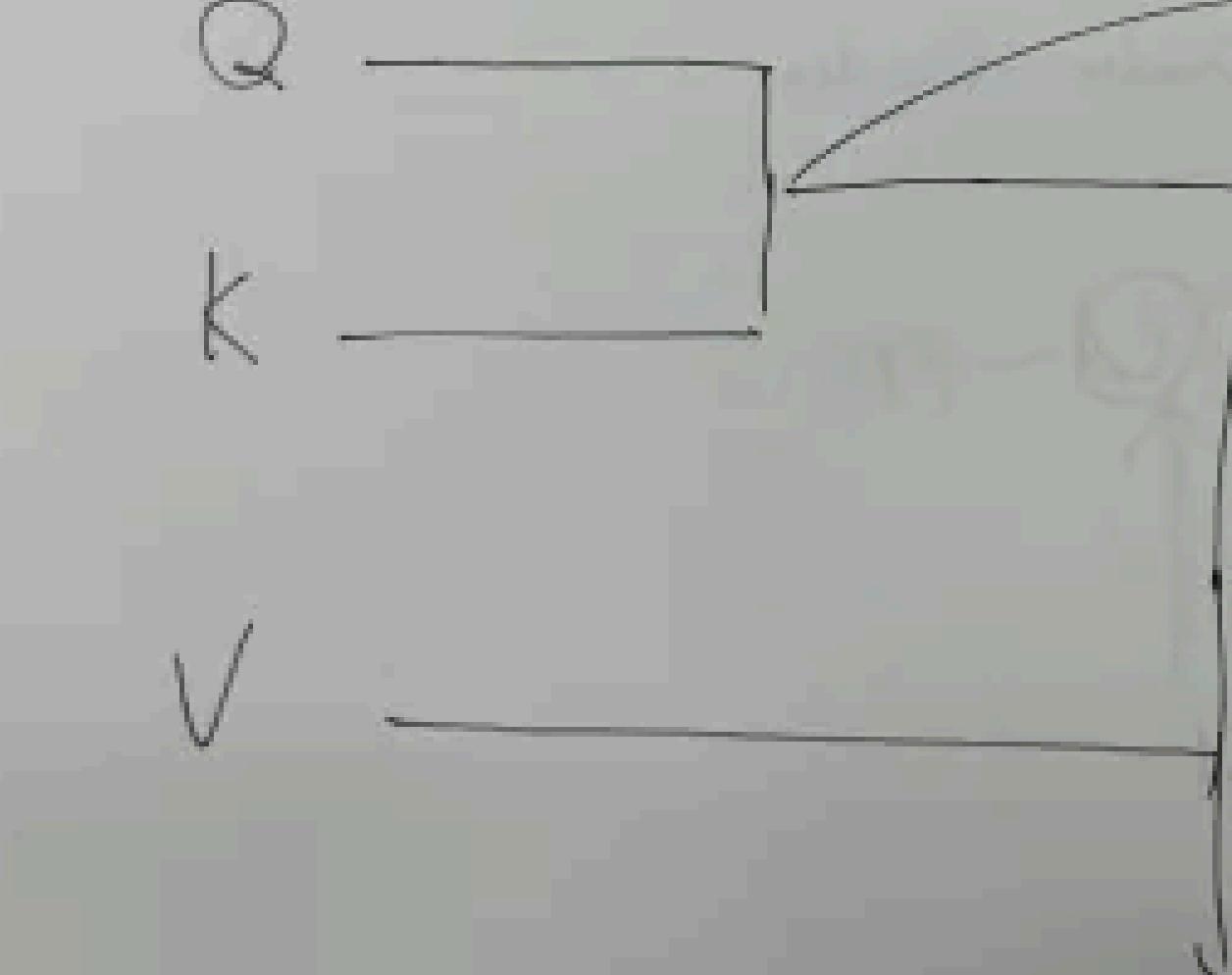


## Q-Formar層

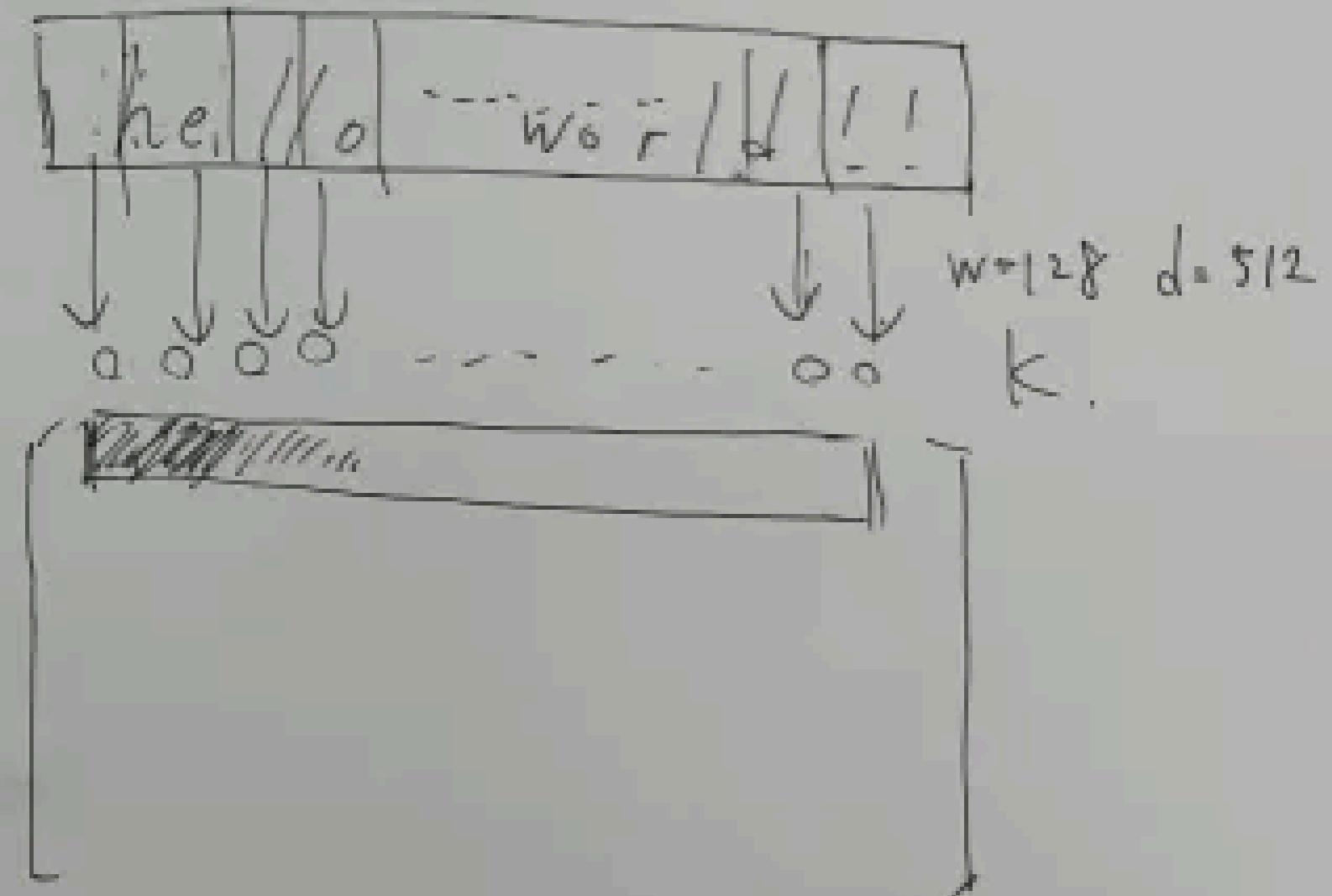
- 画像の特徴とそこでLLMタスクで使用する特徴ベクトル（言語意味空間）に変換させるため
- image encoderの中間表現と学習可能（中間表現でどこ参照すべきか学習）queryとでクロスアテンション言語タスクとの関連速度の高いものを抽出

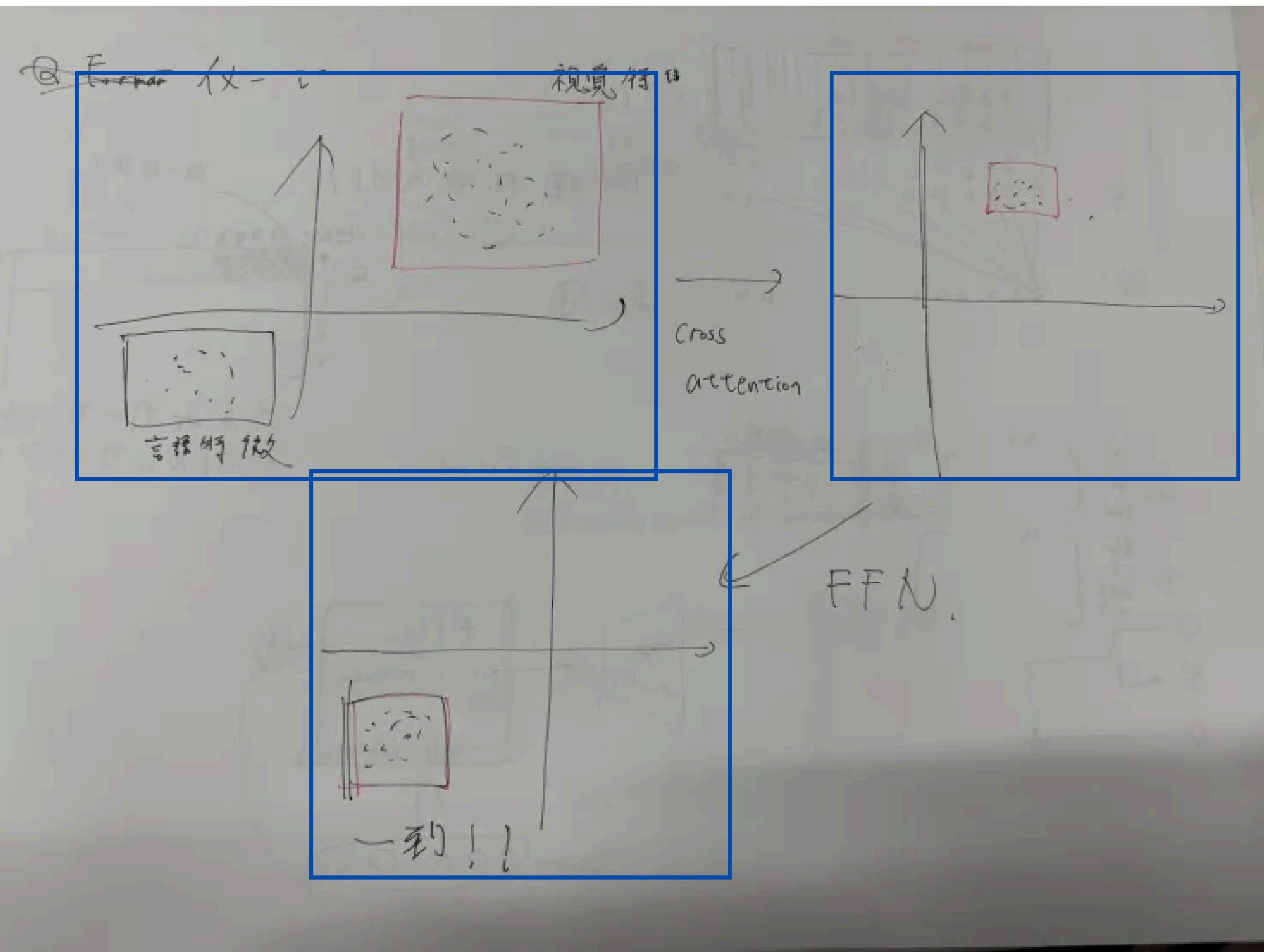
学習可能  $d = 512$

$2^0 + 2^1 \dots (16)$



「文字目」の特徴は?

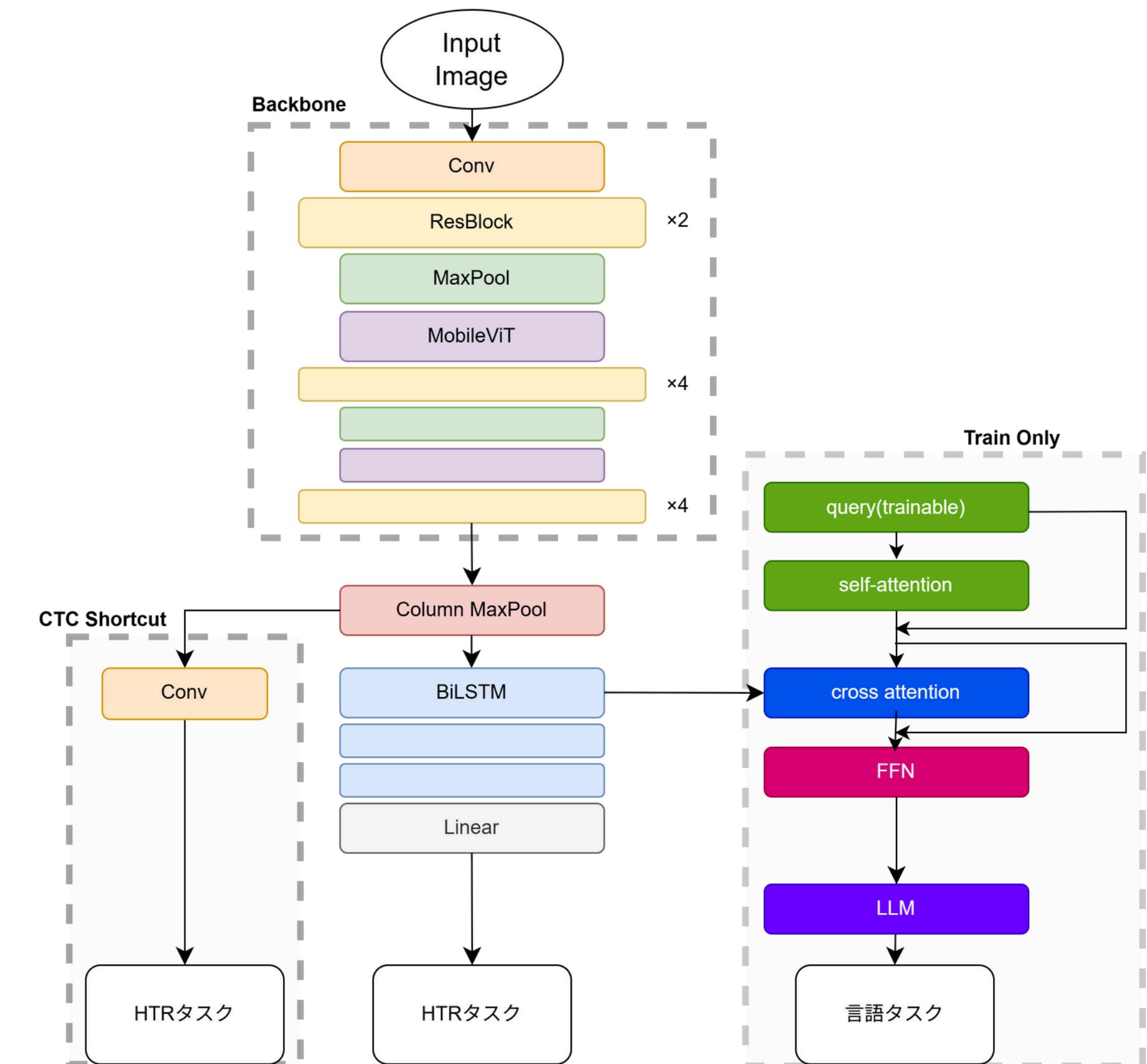




# 予稿モデル図

モデル図書き方

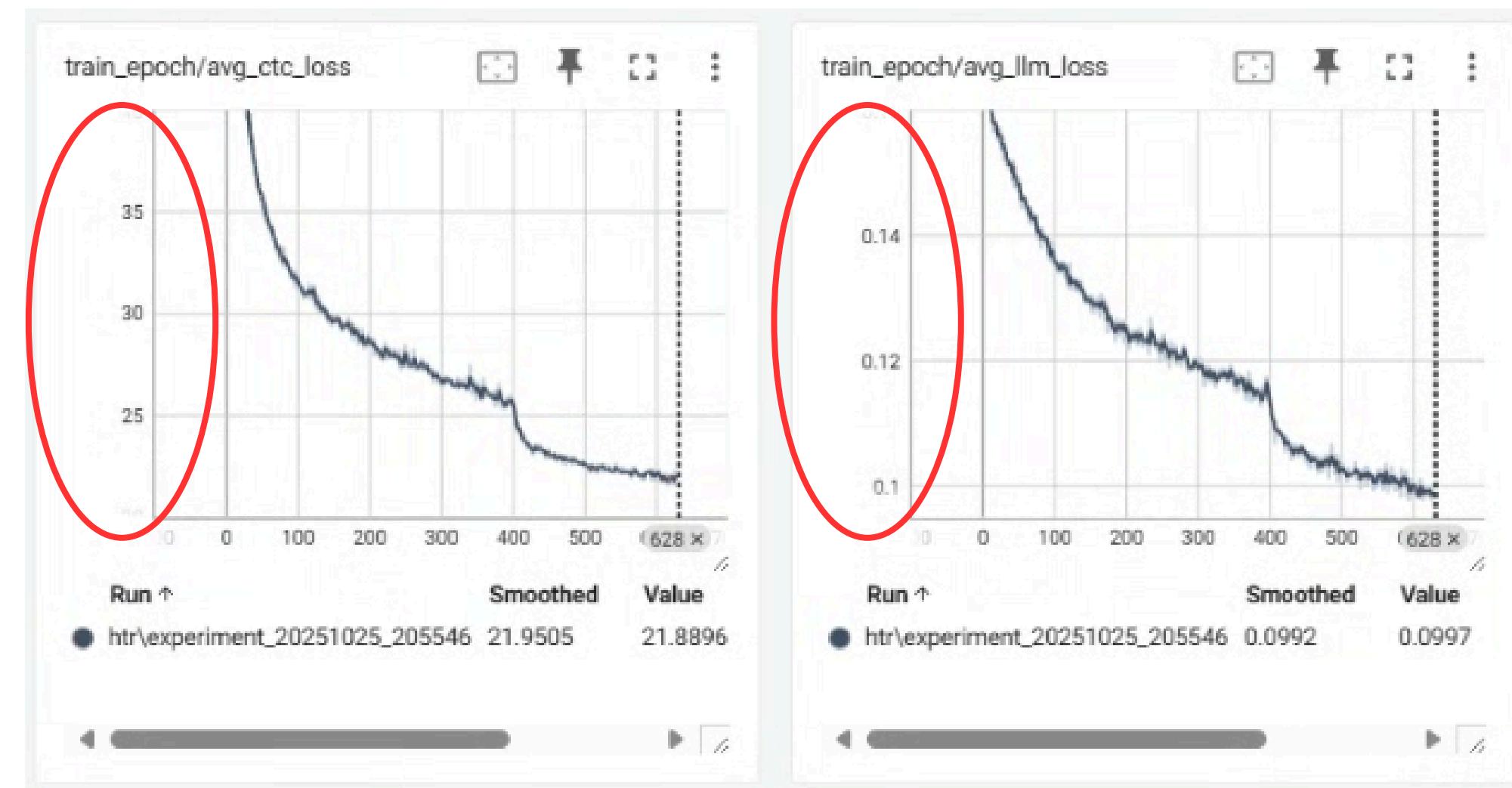
アドバイス一覧



# 現状の結果

現状

- LLMロスの有効性が消えるのでは？？
- 精度そんなに変わらない



# 現状の結果

## 対策

- CTCロスの値下げる
- LLMロスの値を大きくする
- 途中エポックでllmlossを途中epochで大きくする

# 現状の結果



# 研究テーマについて

高速化について

既存手法よりちょっとはやいで意味あるのか

リアルタイム性に必要な速度と比較してその基準にたいして既存手法の二倍

ましになったでは意味あるのか、また、基準を超えるか

本気で高速化ならCのほうがはやいのでCで書くか定量的にpython等で書いたコードがCの何倍になるのかわかってるならCで書かなくてもいいのかも

**高速化はむずかしいイメージ**

nips iclr,とかはアルゴリズムのイメージ