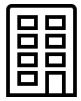
2022 하계 세미나

# Cross Attention Module

***Sogang University***
*Vision & Display Systems Lab, Dept. of Electronic Engineering*

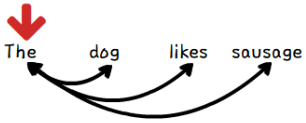
***Presented by***
**유현우**

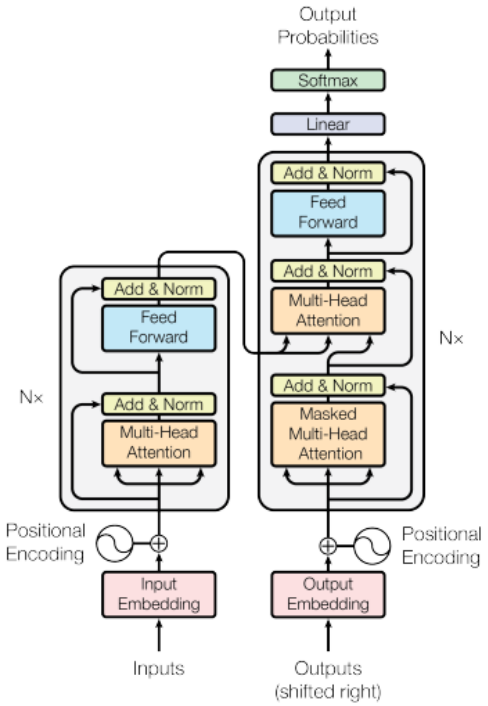
# Outline

- Background
  - Attention is all you need
- Cross attention module
  - Transformer backbone
    - CrossViT : Cross-Attention Multi-Scale Vision Transformer for Image Classification (ICCV 2021)
  - Multimodal
    - LAVT : Language-Aware Vision Transformer for Referring Image Segmentation (CVPR 2022)
- Conclusion

# Background

- Attention is all you need[1]
  - 자연어 처리 분야에서 transformer 구조를 제안
  - Encoder 단에서 self attention을 이용해 global feature representation을 수행
  - Decoder 단에서 encoder-decoder attention(cross attention)을 사용
    - Encoder 최종 출력 feature를 decoder에서 key, value로 사용
    - Decoder time step에 따라 iteration

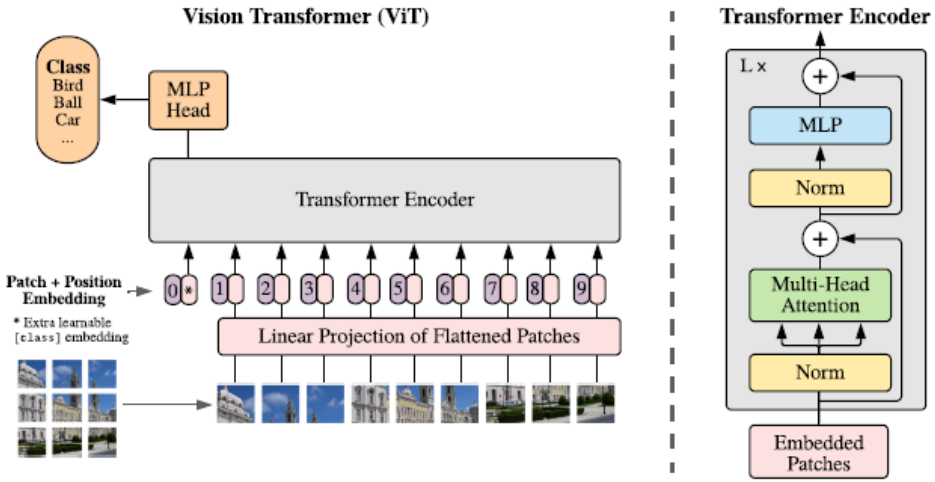


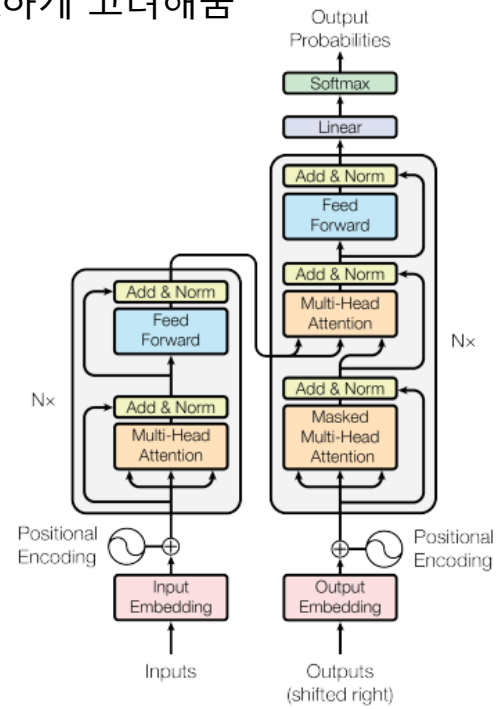

Transformer

# Background

- Vision transformer[1]
  - 자연어 처리에서 사용되는 transformer를 vision task에 적용한 연구
  - Image를 patch 단위로 나누어 embedding한 후 self-attention을 이용해 feature representation을 수행
    - Embedding된 patch간의 관계를 self-attention layer에서 global하게 고려해줌



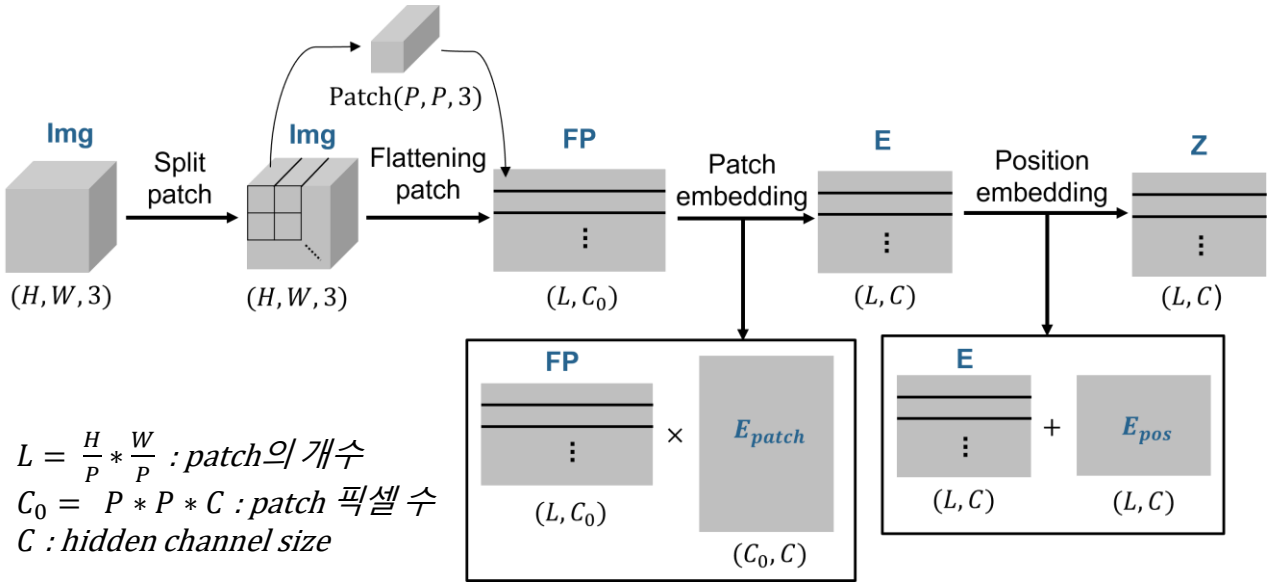

Vision Transformer




Transformer

# Background

- Self attention
  - Input patch를 flatten한 후 linear projection을 이용해 patch embedding 수행
    - Patch를 대표하는 feature를 1d token의 형태로 표현
  - Linear projection 연산은 MLP layer 또는 1x1 convolution으로 수행 가능




Embedding overview

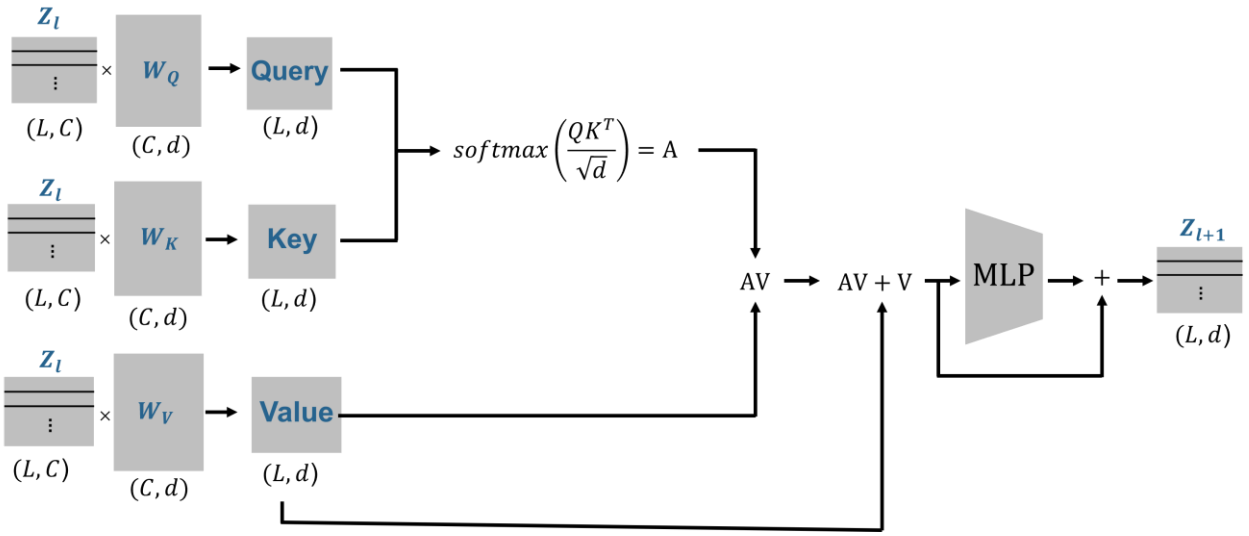
# Background

- Self attention
  - Learnable parameter로 linear projection을 수행하여 query, key, value 생성
  - Query, key간의 연산(Key · Query.transpose)을 통해 얻은 가중치를 value에 적용

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}_{qkv} \qquad \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h},$$

$$A = \text{softmax}\left(\mathbf{q}\mathbf{k}^{\top}/\sqrt{D_h}\right) \qquad A \in \mathbb{R}^{N \times N},$$

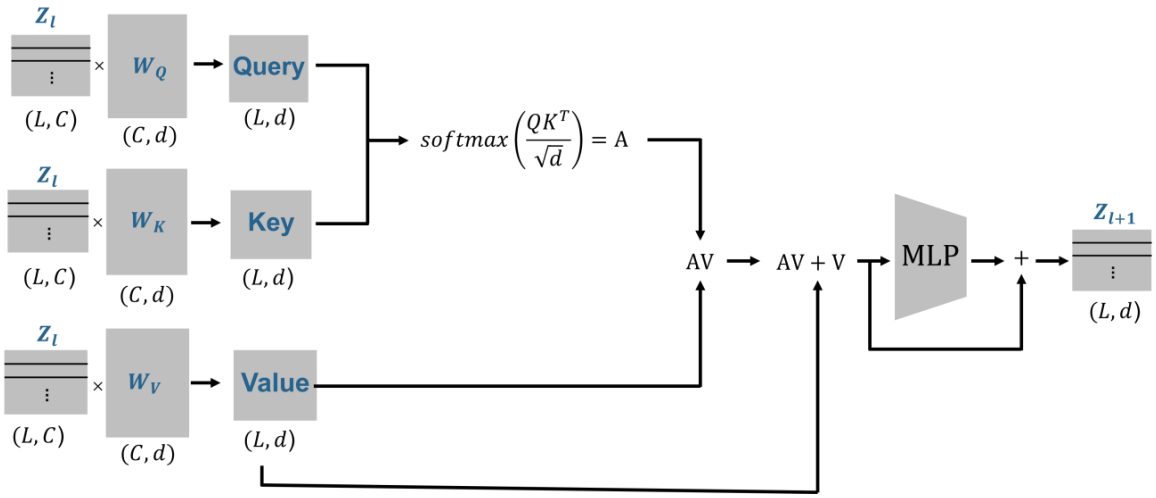
$$\text{SA}(\mathbf{z}) = A\mathbf{v}\,.$$




Self-attention overview

# Background

- Self attention
  - Global feature representation
    - 특정 patch가 모든 영역의 patch간의 관계를 참조
  - Query, key, value 기반의 attention 구조
    - Query와 key간의 관계를 참조해 얻은 정보를 value에 적용해서 출력하여
    - Query와 key간의 similarity를 기반으로 attention을 가해주는 방식
    - 다른 attention 기법 중 channel pooling해서 spatial relation을 고려해주거나 spatial pooling해서 channel relation을 고려해서 attention을 해주는 방법이 있음



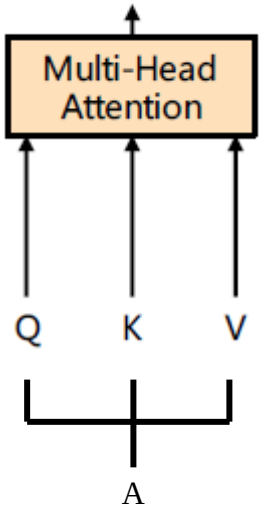

Self-attention overview

# Background

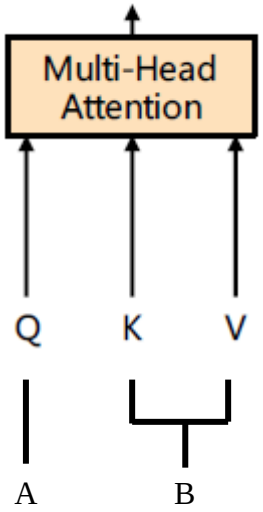
- Vision transformer
  - Transformer 기반 백본들은 전부 attention is all you need에 제안된 self attention 기반 encoder 만 사용하는 것
    - 그니까 self attention이 global feature representation이 가능하다는 장점에 주목
  - Global을 고려해주는 것을 대체할 수 있으면 self-attention은 필요 없다는 연구가 다수 등장
  - 하지만 Query key value 기반의 attention 구조를 활용하여 다양한 task에 적용할 수 있음

# Background

- Cross attention
  - Cross attention은 self attention과 mechanism은 동일하나 input의 출처가 다름
  - query, key, value 기반의 attention 구조의 장점을 활용하는 cross attention이 있음
  - key value값을 통해 어떻게 attention 해줄지를 설정
    - Key, value를 어떻게 설정하느냐에 따라서 특정 inductive bias를 가해줄 수 있음
  - Multi scale 또는 multi modal 등을 fusion 하는 연구에 사용



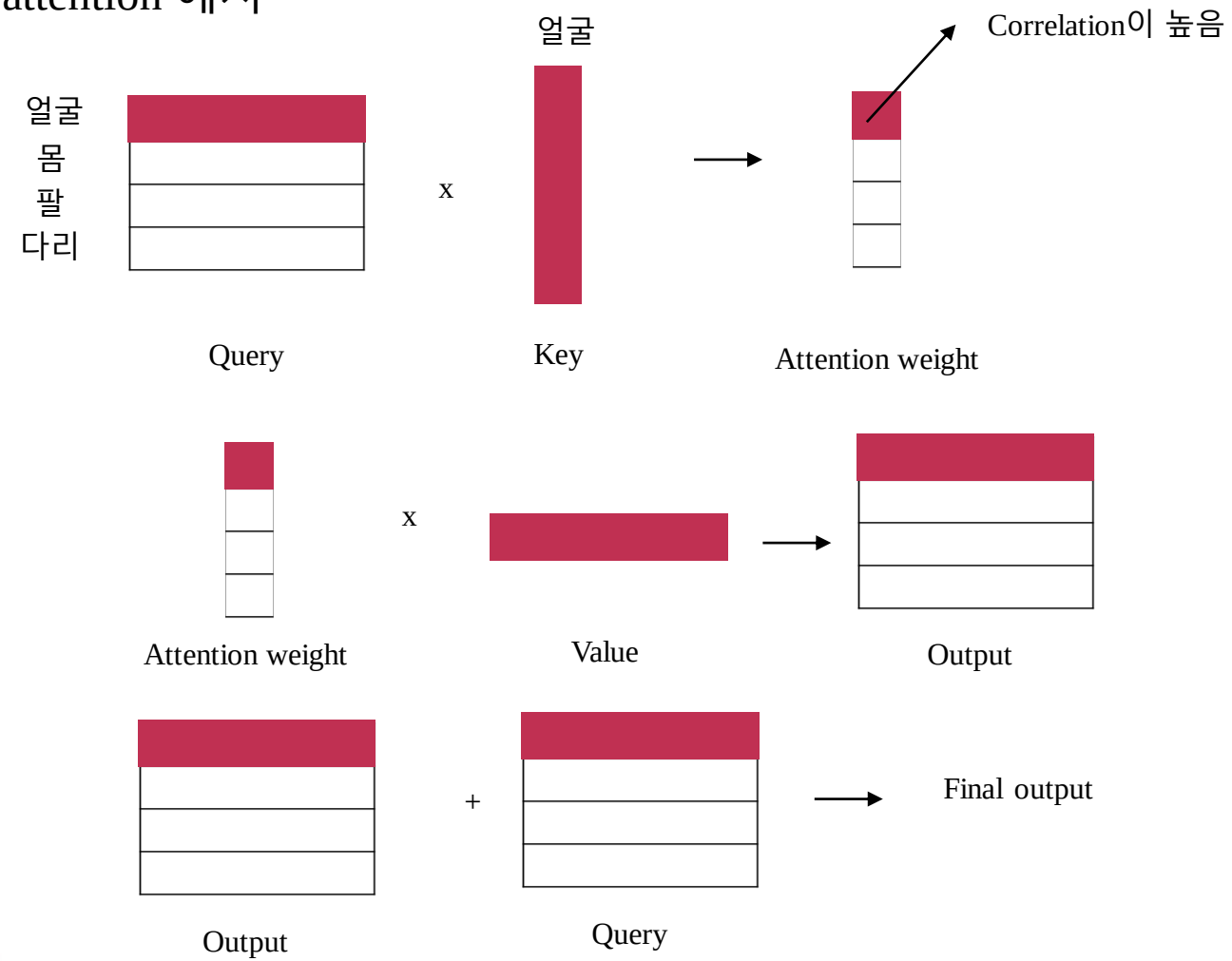

Self attention




Cross attention

# Background

- Cross attention 예시



# CrossViT[1]

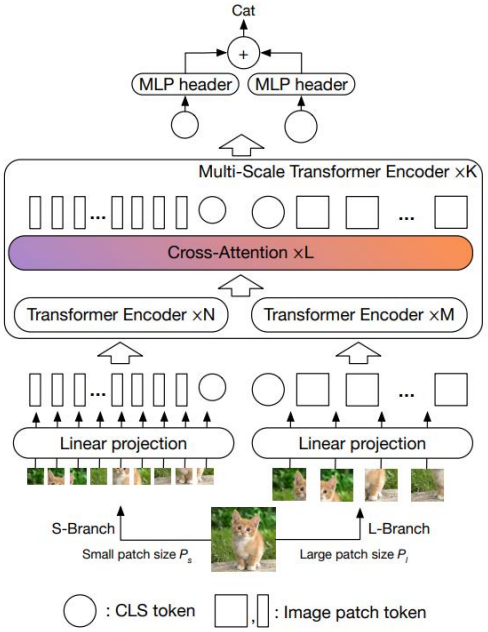
- Vision transformer 에서 patch size는 성능에 영향을 미침
  - Patch size 16x16과 32x32를 비교해 보았을 때 16x16에서 성능이 더 좋지만 self-attention에서 computational cost가 quadratic 하게 증가함
    - Self attention computational cost : $2cp^2$ (c: token dimension, p : patch 개수)
- 또한 multi scale feature representation은 CNN기반 vision task에서 좋은 성능을 보여줌
  - Multi scale feature가 CNN에 도움이 되는 것으로 나타났지만 vision transformer에서는 이러한 잠재적 이점이 검증되어야 함

| name | Epochs | ImageNet | ImageNet ReaL | CIFAR-10 | CIFAR-100 | Pets | Flowers | exaFLOPs |
|---|---|---|---|---|---|---|---|---|
| ViT-B/32 | 7 | 80.73 | 86.27 | 98.61 | 90.49 | 93.40 | 99.27 | 55 |
| ViT-B/16 | 7 | 84.15 | 88.85 | 99.00 | 91.87 | 95.80 | 99.56 | 224 |
| ViT-L/32 | 7 | 84.37 | 88.28 | 99.19 | 92.52 | 95.83 | 99.45 | 196 |
| ViT-L/16 | 7 | 86.30 | 89.43 | 99.38 | 93.46 | 96.81 | 99.66 | 783 |

Patch size에 따른 ViT 성능 비교

# CrossViT

- CrossViT architecture
  - CrossViT에서는 ViT 구조에서 multi scale feature representation을 하기 위해 patch embedding 단계에서 patch size를 다르게 설정해줌
  - 그리고 각각을 transformer layer를 이용해 embedding한 후 Cross-attention을 이용해 feature fusion



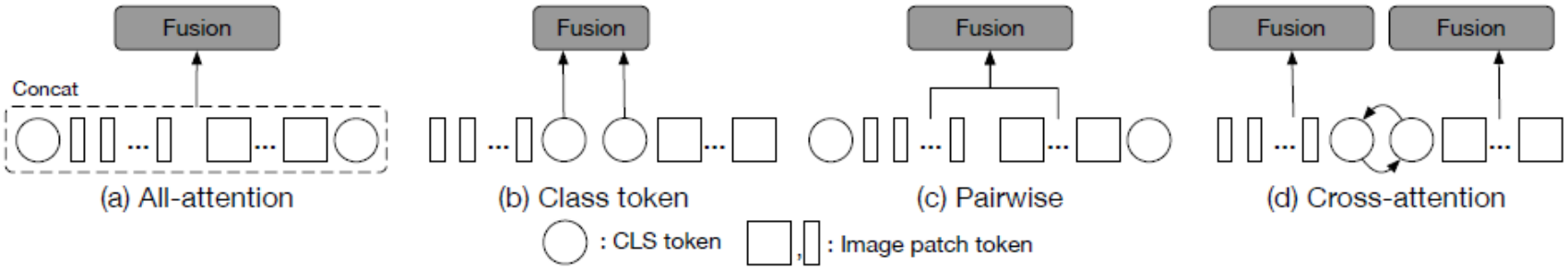

CrossViT architecture

# CrossViT

- Multi-scale fusion
  - Exchange information
    - (A) all attention fusion : simply concatenate 후 self-attention
    - (B) class token fusion : 두 branch의 class token을 sum
    - (C) pairwise fusion : 두 branch의 class token 끼리 sum하고 patch token끼리 sum
    - (D) cross-attention : 한 branch의 class token과 다른 branch의 patch token간의 cross attention

| Fusion | Top-1 Acc. (%) | FLOPs (G) | Params (M) | Single Branch Acc. (%) L-Branch | S-Branch |
|---|---|---|---|---|---|
| None | 80.2 | 5.3 | 23.7 | 80.2 | 0.1 |
| All-Attention | 80.0 | 7.6 | 27.7 | 79.9 | 0.5 |
| Class Token | 80.3 | 5.4 | 24.2 | 80.6 | 7.6 |
| Pairwise | 80.3 | 5.5 | 24.2 | 80.3 | 7.3 |
| Cross-Attention | 81.0 | 5.6 | 26.7 | 68.1 | 47.2 |

Table 6: **Ablation study with different fusions on ImageNet1K.** All models are based on CrossViT-S. Single branch Acc. is computed using `CLS` from one branch only.

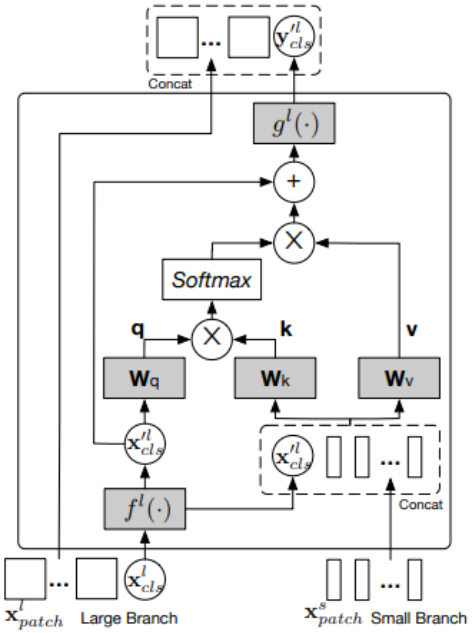



Multi-scale fusion method

# CrossViT

- Multi-scale fusion (large branch 기준)
  - Large branch의 class token을 query로 하고 small branch의 patch token을 key로 하여 cross attention 수행
    - Query를 class token만 사용하므로 Efficient함
  - Small branch가 query일 때는 위의 과정을 반대로 수행
  - 특정 branch의 class token과 다른 branch의 patch token이 Exchange information

$$\mathbf{q} = \mathbf{x}_{cls}'^{l}\mathbf{W}_q, \quad \mathbf{k} = \mathbf{x}'^{l}\mathbf{W}_k, \quad \mathbf{v} = \mathbf{x}'^{l}\mathbf{W}_v,$$
$$\mathbf{A} = \mathtt{softmax}(\mathbf{q}\mathbf{k}^T/\sqrt{C/h}), \quad \mathtt{CA}(\mathbf{x}'^{l}) = \mathbf{A}\mathbf{v},$$




Cross-attention for large branch

# CrossViT

- Results
  - 다른 fusion 방법론을 사용했을 때 결과 비교에서 sota가 아님

| Fusion | Top-1 Acc. (%) | FLOPs (G) | Params (M) | Single Branch Acc. (%) L-Branch | S-Branch |
|---|---|---|---|---|---|
| None | 80.2 | 5.3 | 23.7 | 80.2 | 0.1 |
| All-Attention | 80.0 | 7.6 | 27.7 | 79.9 | 0.5 |
| Class Token | 80.3 | 5.4 | 24.2 | 80.6 | 7.6 |
| Pairwise | 80.3 | 5.5 | 24.2 | 80.3 | 7.3 |
| Cross-Attention | 81.0 | 5.6 | 26.7 | 68.1 | 47.2 |

Table 6: **Ablation study with different fusions on ImageNet1K.** All models are based on CrossViT-S. Single branch Acc. is computed using `CLS` from one branch only.

| Model | Top-1 Acc. (%) | FLOPs (G) | Throughput (images/s) | Params (M) |
|---|---|---|---|---|
| DeiT-Ti | 72.2 | 1.3 | 2557 | 5.7 |
| CrossViT-Ti | 73.4 (+1.2) | 1.6 | 1668 | 6.9 |
| CrossViT-9 | 73.9 (+0.5) | 1.8 | 1530 | 8.6 |
| CrossViT-9† | **77.1** (+3.2) | 2.0 | 1463 | 8.8 |
| DeiT-S | 79.8 | 4.6 | 966 | 22.1 |
| CrossViT-S | 81.0 (+1.2) | 5.6 | 690 | 26.7 |
| CrossViT-15 | 81.5 (+0.5) | 5.8 | 640 | 27.4 |
| CrossViT-15† | **82.3** (+0.8) | 6.1 | 626 | 28.2 |
| DeiT-B | 81.8 | 17.6 | 314 | 86.6 |
| CrossViT-B | 82.2 (+0.4) | 21.2 | 239 | 104.7 |
| CrossViT-18 | 82.5 (+0.3) | 9.0 | 430 | 43.3 |
| CrossViT-18† | **82.8** (+0.3) | **9.5** | 418 | 44.3 |

Table 2: **Comparisons with DeiT baseline on ImageNet1K.** The numbers in the bracket show the improvement from each change. See Table 1 for model details.

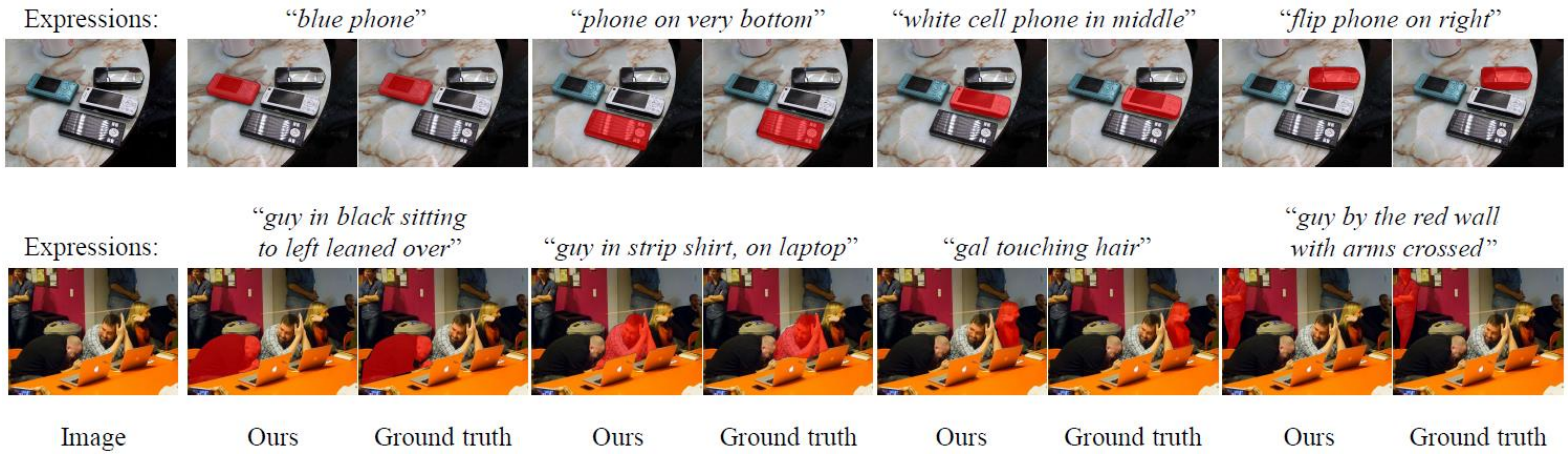
| Model | Top-1 Acc. (%) | FLOPs (G) | Params (M) |
|---|---|---|---|
| Peceiver [19] (arXiv, 2021-03) | 76.4 | – | 43.9 |
| DeiT-S [35] (arXiv, 2020-12) | 79.8 | 4.6 | 22.1 |
| CentroidViT-S [42] (arXiv, 2021-02) | 80.9 | 4.7 | 22.3 |
| PVT-S [38] (arXiv, 2021-02) | 79.8 | 3.8 | 24.5 |
| PVT-M [38] (arXiv, 2021-02) | 81.2 | 6.7 | 44.2 |
| T2T-ViT-14 [45] (arXiv, 2021-01) | 80.7 | 6.1* | 21.5 |
| TNT-S [14] (arXiv, 2021-02) | 81.3 | 5.2 | 23.8 |
| CrossViT-15 (Ours) | 81.5 | 5.8 | 27.4 |
| CrossViT-15† (Ours) | **82.3** | 6.1 | 28.2 |
| ViT-B@384 [11] (ICLR, 2021) | 77.9 | 17.6 | 86.6 |
| DeiT-B [35] (arXiv, 2020-12) | 81.8 | 17.6 | 86.6 |
| PVT-L [38] (arXiv, 2021-02) | 81.7 | 9.8 | 61.4 |
| T2T-ViT-19 [45] (arXiv, 2021-01) | 81.4 | 9.8* | 39.0 |
| T2T-ViT-24 [45] (arXiv, 2021-01) | 82.2 | 15.0* | 64.1 |
| TNT-B [14] (arXiv, 2021-02) | **82.8** | 14.1 | 65.6 |
| CrossViT-18 (Ours) | 82.5 | 9.0 | 43.3 |
| CrossViT-18† (Ours) | **82.8** | 9.5 | 44.3 |

*: We recompute the flops by using our tools.

Table 3: **Comparisons with recent transformer-based models on ImageNet1K.** All models are trained using only ImageNet1K dataset. Numbers are referenced from their recent version as of the submission date.

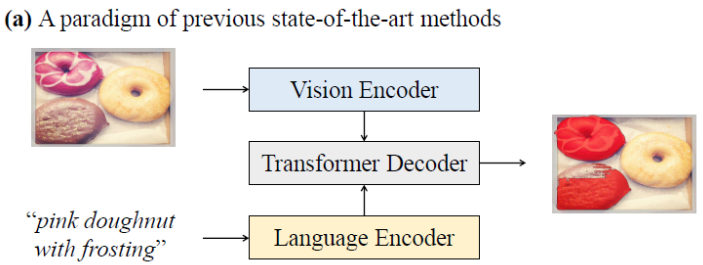
# LAVT[1]

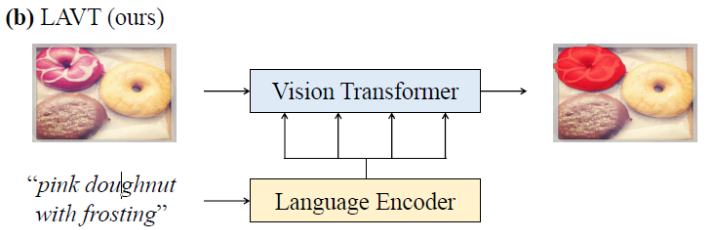
- Referring image segmentation
  - Multi modal : vision 뿐만 아니라 text, audio와 같은 다양한 modal의 data를 사용하여 학습하는 분야
  - Visual ground : text가 input 되었을 때 이미지 내에서 해당되는 영역을 detecting하는 task
  - Referring segmentation : visual grounding의 한 분야로 text가 input 되었을 때 해당 영역을 pixel단위로 segmentation 하는 task



# LAVT

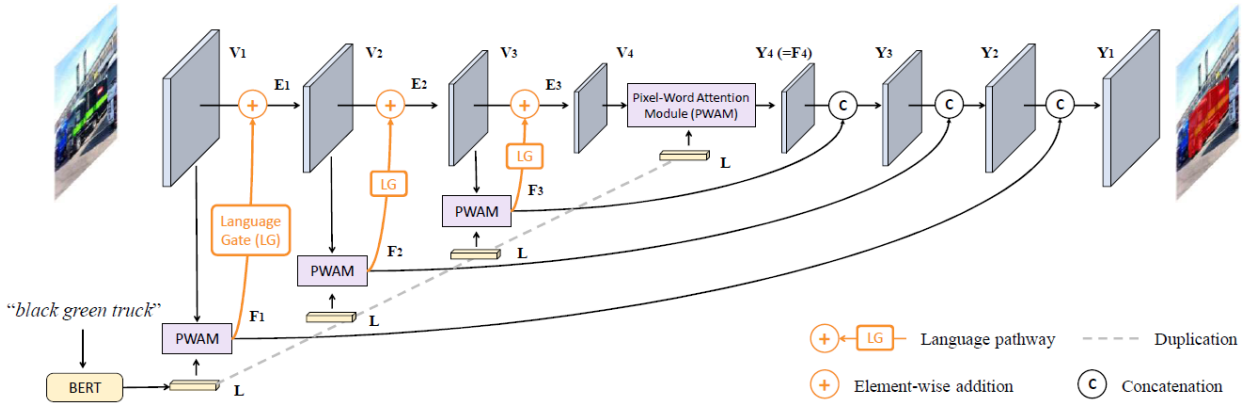
- Referring image segmentation
  - Referring segmentation을 하기 위해서는 vision encoder와 language encoder를 함께 학습시켜야 함
    - Vision network가 language를 aware해서 feature representation 하는 것이 목표
  - Vision encoder와 language encoder에서 독립적으로 추출된 feature를 alignment하기 위해 fusion
    - 여기서 alignment는 다른 modal의 feature를 동일 space로 mapping해서 다루기 위해 feature를 유사한 형태로 만들어주는 과정
  - 기존에는 transformer decoder를 이용해서 fusion
    - 즉, aligning the vision and language feature를 하는데 decoder단만 활용
    - Transformer decoder는 cross attention을 기반으로 이루어짐



# LAVT

- LAVT architecture
  - 두 modal간의 alignment를 vision encoder인 transformer layer에서 진행
    - Vision encoder는 swin transformer backbone을 사용하고 language encoder는 BERT를 사용
    - BERT를 이용해 input expression 을 high dimensional word vector로 embedding
    - Embedding된 language feature를 vision encoder를 학습시킬 때 사용
  - Vision encoder에서 stage별로 Pixel-Word Attention Module(PWAM)를 활용
    - Vision encoder에서 vision and language feature를 jointly embedding
      - Linguistic feature를 visual feature에 densely integrate




LAVT architecture

# LAVT

- Cross modal alignment
  - Input visual feature($V_i$)를 query, input linguistic feature ($L$) 를 key, value로 사용하여 cross attention 수행
    - Visual feature가 pixel 단위로 linguistic feature를 참조하여 linguistic information을 수집

$$V_{iq} = flatten(w_{iq}(V_i))$$

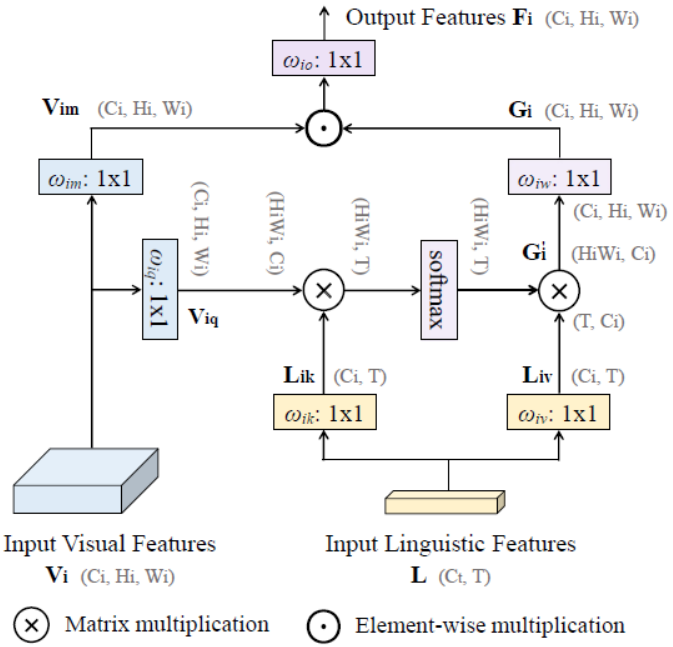
$$L_{ik} = w_{ik}(L), L_{iv} = w_{iv}(L)$$

$$G'_i = softmax(\frac{V_{iq}^T L_{ik}}{\sqrt{C_i}})L_{iv}^T$$

$$G_i = w_{iw}(unflatten(G'^T_i))$$

$$V_i \in \mathbb{R}^{C_i \times H_i \times W_i}$$

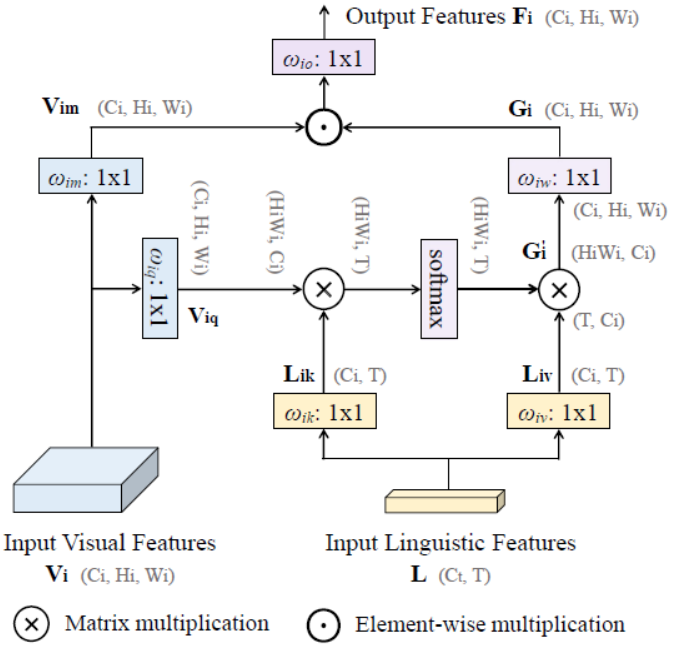
$$L \in \mathbb{R}^{C_t \times T}$$




PWAM

# LAVT

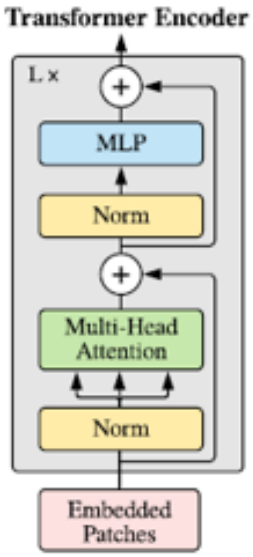
- Cross modal alignment
  - $G_i$는 $V_i$와 spatial size가 같은 set of linguistic feature maps
  - 이를 $V_{im}$과 element wise multiplication을 통해 visual feature map에 projection
    - $V_{im} = w_{im}(V_i)$
    - $F_i = w_{io}(V_{im} \odot G_i)$

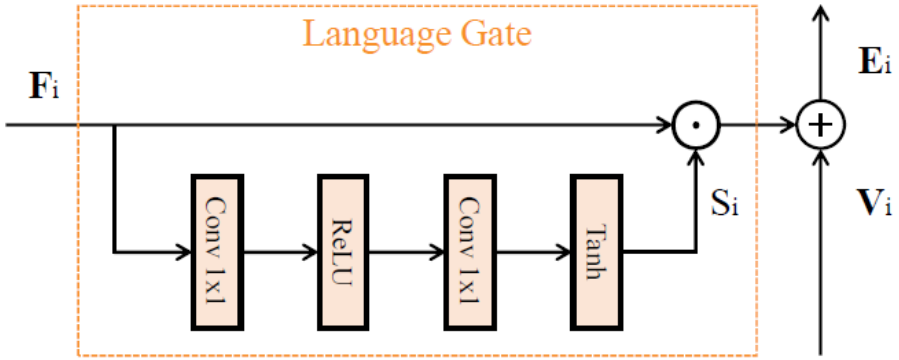



PWAM

# LAVT

- Cross modal alignment
  - PWAM을 수행한 후 language pathway를 통과하는데 이는 기존 transformer encoder에서 사용되는 feed-forward layer와 동일




Language pathway

# LAVT

- Results

| Method | Language Model | RefCOCO | | | RefCOCO+ | | | G-Ref | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | val | test A | test B | val | test A | test B | val (U) | test (U) | val (G) |
| DMN [43] | SRU | 49.78 | 54.83 | 45.13 | 38.88 | 44.22 | 32.29 | - | - | 36.76 |
| RRN [30] | LSTM | 55.33 | 57.26 | 53.93 | 39.75 | 42.15 | 36.11 | - | - | 36.45 |
| MAttNet [63] | Bi-LSTM | 56.51 | 62.37 | 51.70 | 46.67 | 52.39 | 40.08 | 47.64 | 48.61 | - |
| CMSA [62] | None | 58.32 | 60.61 | 55.09 | 43.76 | 47.60 | 37.89 | - | - | 39.98 |
| CAC [8] | Bi-LSTM | 58.90 | 61.77 | 53.81 | - | - | - | 46.37 | 46.95 | 44.32 |
| STEP [5] | Bi-LSTM | 60.04 | 63.46 | 57.97 | 48.19 | 52.33 | 40.41 | - | - | 46.40 |
| BRINet [23] | LSTM | 60.98 | 62.99 | 59.21 | 48.17 | 52.32 | 42.11 | - | - | 48.04 |
| CMPC [24] | LSTM | 61.36 | 64.53 | 59.64 | 49.56 | 53.44 | 43.23 | - | - | 49.05 |
| LSCM [25] | LSTM | 61.47 | 64.99 | 59.55 | 49.34 | 53.12 | 43.50 | - | - | 48.05 |
| CMPC+ [34] | LSTM | 62.47 | 65.08 | 60.82 | 50.25 | 54.04 | 43.47 | - | - | 49.89 |
| MCN [41] | Bi-GRU | 62.44 | 64.20 | 59.71 | 50.62 | 54.99 | 44.69 | 49.22 | 49.40 | - |
| EFN [15] | Bi-GRU | 62.76 | 65.69 | 59.67 | 51.50 | 55.24 | 43.01 | - | - | 51.93 |
| BUSNet [58] | Self-Att | 63.27 | 66.41 | 61.39 | 51.76 | 56.87 | 44.13 | - | - | 50.56 |
| CGAN [40] | Bi-GRU | 64.86 | 68.04 | 62.07 | 51.03 | 55.51 | 44.06 | 51.01 | 51.69 | 46.54 |
| LTS [27] | Bi-GRU | 65.43 | 67.76 | 63.08 | 54.21 | 58.32 | 48.02 | 54.40 | 54.25 | - |
| VLT [13] | Bi-GRU | 65.65 | 68.29 | 62.73 | 55.50 | 59.20 | 49.36 | 52.99 | 56.65 | 49.76 |
| LAVT (Ours) | BERT | **72.73** | **75.82** | **68.79** | **62.14** | **68.38** | **55.10** | **61.24** | **62.09** | **60.50** |

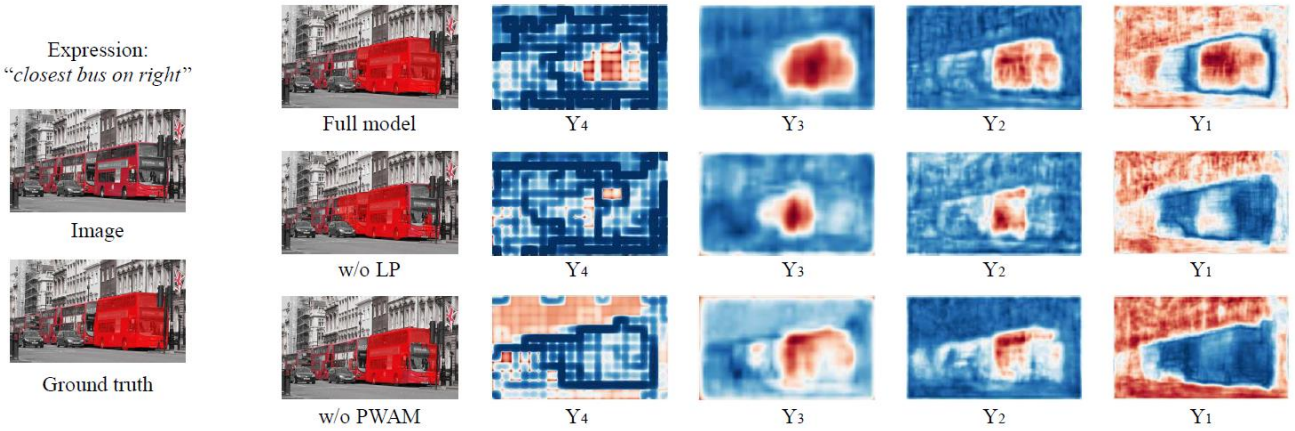
Table 1. Comparison with state-of-the-art methods in terms of overall IoU on three benchmark datasets. U: The UMD partition. G: The Google partition. We refer to the language model of each reference method as the main learnable function that transforms word embeddings before multi-modal feature fusion. Interested readers can refer to the respective papers for embedding initialization and other details.

# LAVT

- Results

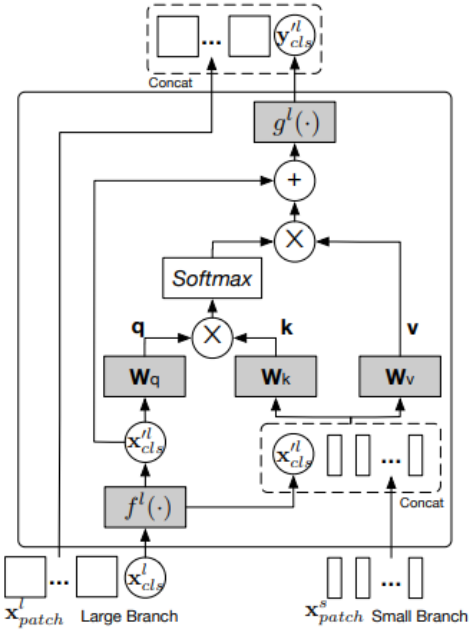
| LP | PWAM | P@0.5 | P@0.7 | P@0.9 | oIoU | mIoU |
|---|---|---|---|---|---|---|
| ✓ | ✓ | **84.46** | **75.28** | **34.30** | **72.73** | **74.46** |
| | ✓ | 81.46 | 70.80 | 30.95 | 70.78 | 71.96 |
| ✓ | | 81.76 | 72.76 | 32.46 | 71.03 | 72.31 |
| | | 77.87 | 66.93 | 27.95 | 68.82 | 68.87 |

Table 2. Main ablation results on the RefCOCO validation set.

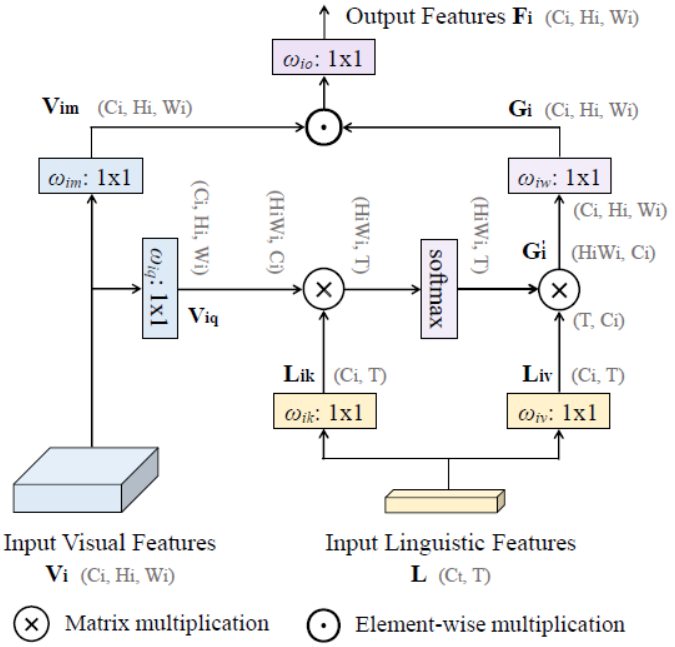


# Conclusion

- Multi scale에서 다양한 scale을 서로 참조할 때 사용
- Multimodal은 vision feature가 language feature를 참조하도록 할 때 사용




Cross-attention for large branch




PWAM