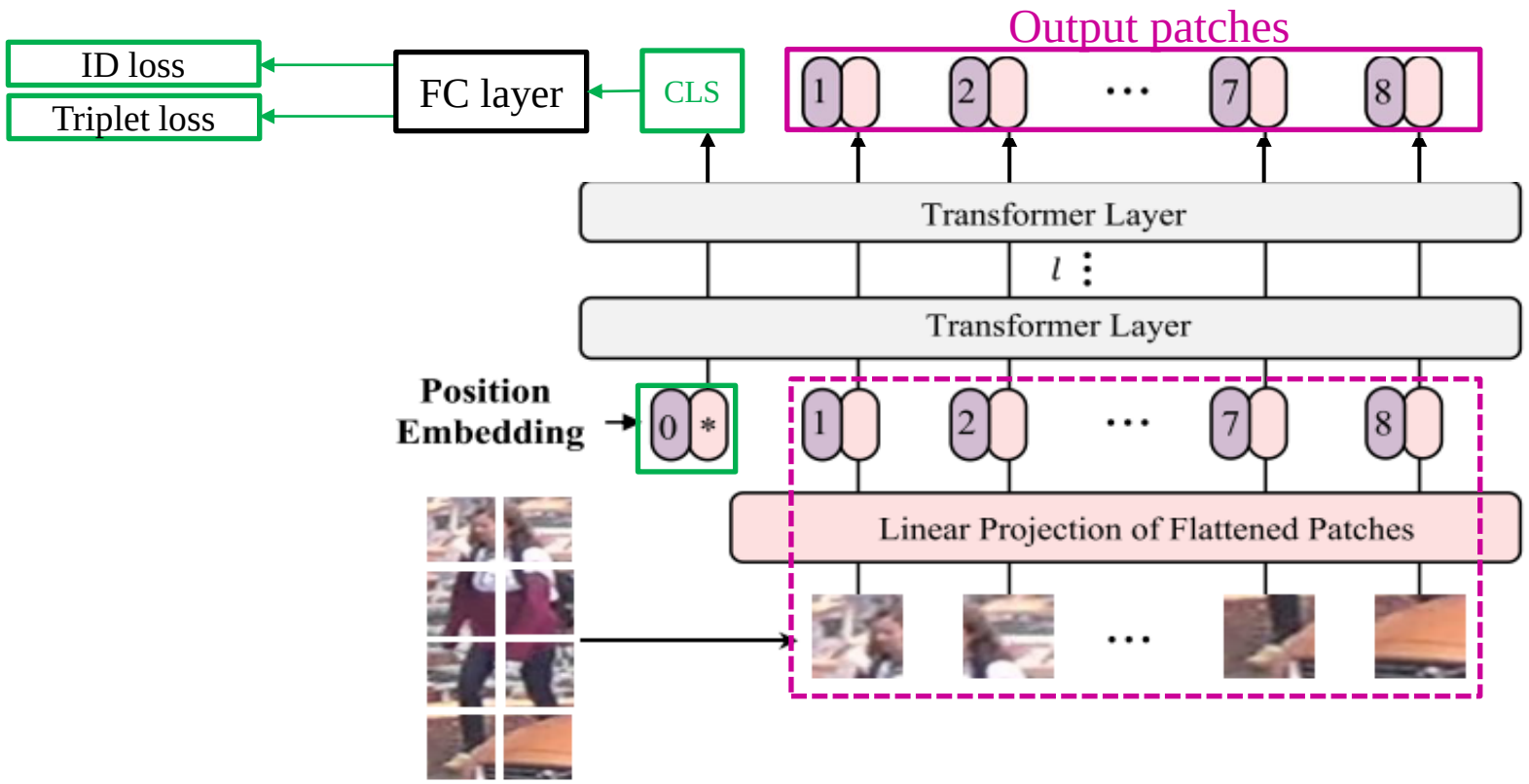
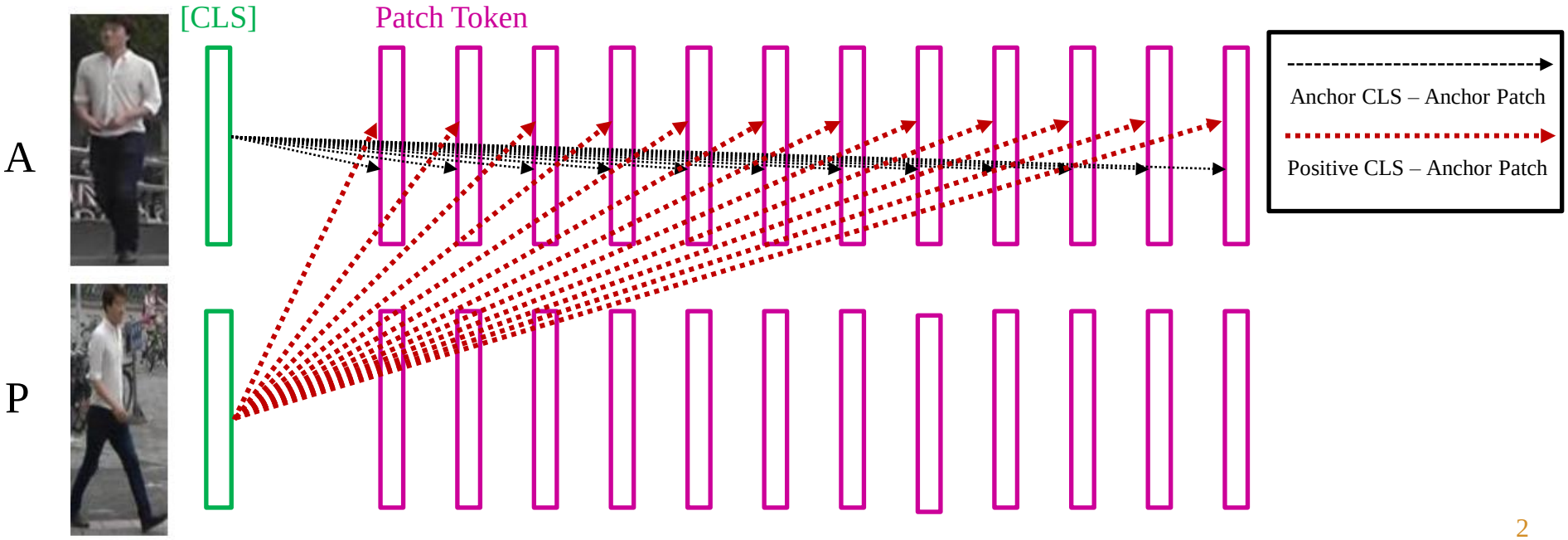
ViT[1]

- ViT[1]의 구조
  - Image가 여러 개의 patch들로 나뉘어 model의 input으로 입력
  - Pooling 된 feature를 사용하는 CNN과 다르게 CLS token 이라는 별도의 learnable parameter를 통해 모델을 학습
  - CLS token은 여러 transformer layer를 거치며 self-attention mechanism을 통해 sample을 구성하는 patch feature들의 weighted summation의 모습을 띔



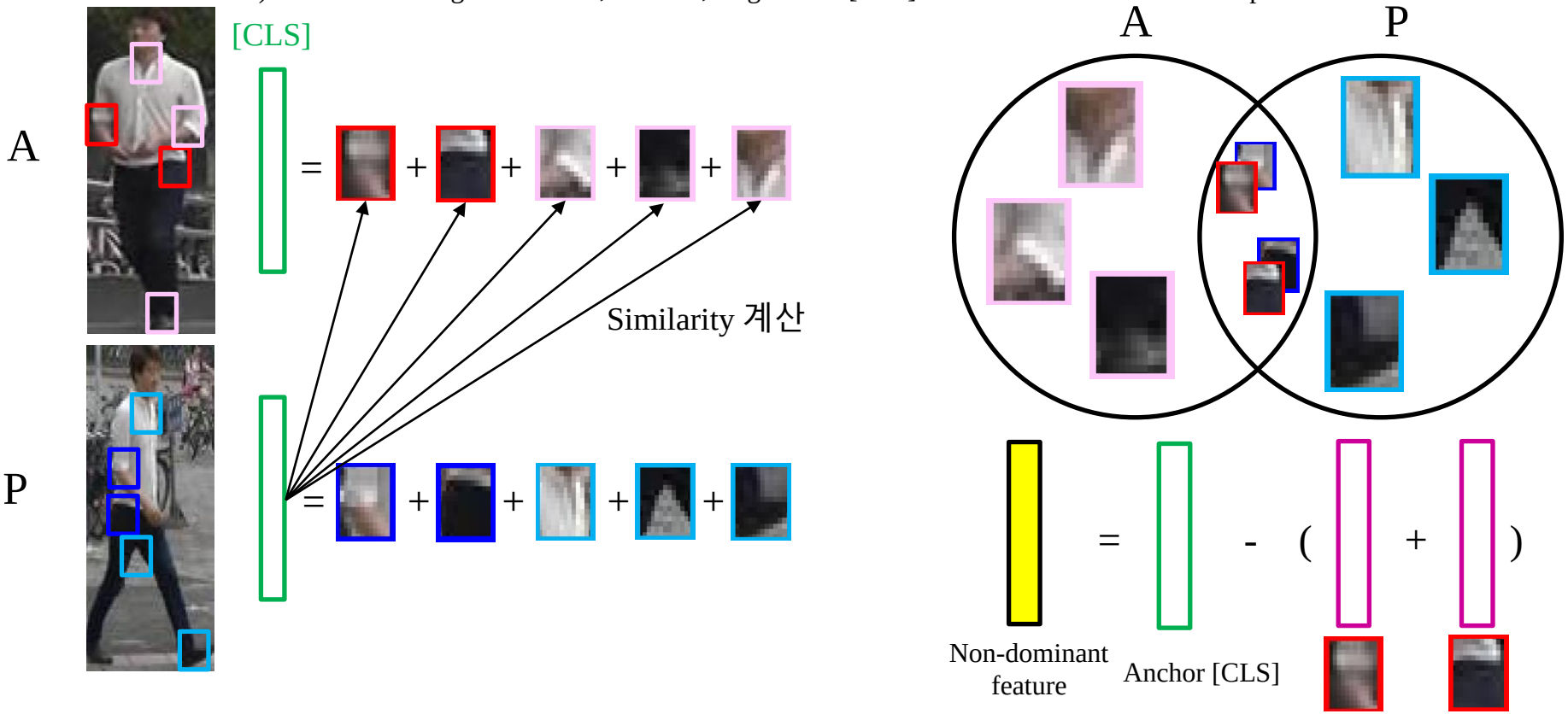
Proposed method 1

- Patch similarity based weighted Triplet loss
  - Triplet loss는 anchor sample과 positive sample의 feature distance를 줄이도록 설계되어 있으나, 두 sample 간 공통된 feature가 dominant하므로 두 sample간 미세한 차이점을 학습하도록 보완이 필요
  - Idea : Anchor sample과 positive sample간 공통된 feature 를 제외한 나머지 feature를 가깝게 하는데 집중할수 있도록 triplet loss를 강화
  - CLS token이 patch token들의 weighted summation이라는 점을 이용
    - 1) Anchor CLS token – Anchor Patch tokens 간의 similarity를 계산, similarity가 높은 상위 50%의 anchor patch 선별
    - 2) Positive CLS token – Anchor Patch tokens 간의 similarity를 계산, similarity가 높은 상위 50%의 anchor patch 선별
    - 3) 1과 2에서 선별된 두 patch 집합의 교집합에 해당하는 patch 선별
    - 4) Anchor 와 Positive의 CLS token에서 3에서 구한 patch들의 weighted summation을 뺀 값을 사용 (뒷 장에 계속)



Proposed method 1

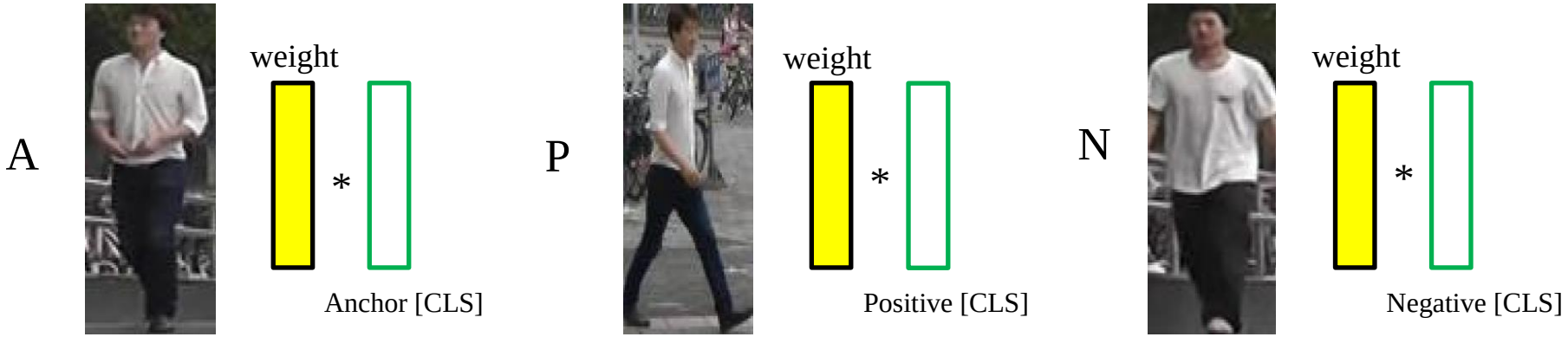
- Patch similarity based weighted Triplet loss
  - CLS token이 patch token들의 weighted summation이라는 점을 이용
    - 5) 4에서 얻은 feature vector를 통해, anchor의 [CLS]의 feature중 non-dominant feature를 구하고 각 feature의 magnitude를 weight로 사용
    - 6) 5에서 구한 weight를 Anchor, Positive, Negative의 [CLS]에 element wise로 곱한 후 triplet loss 계산



*Note
실제 code 상에서 anchor의 CLS token과 patch들의 weighted summation의 차를 구할땐 두 feature의 scale을 고려함

Proposed method 1

- Patch similarity based weighted Triplet loss
  - Weight vector : Non dominant feature를 normalize한 후, anchor [CLS]의 각 feature에 element wise로 곱할 weight vector를 생성



$$L(w * r_a, w * r_p, w * r_n) = \max(0, m + d(w * r_a, w * r_p) - d(w * r_a, w * r_n))$$

r_A, r_P, r_N: sample representations

w: $weight$

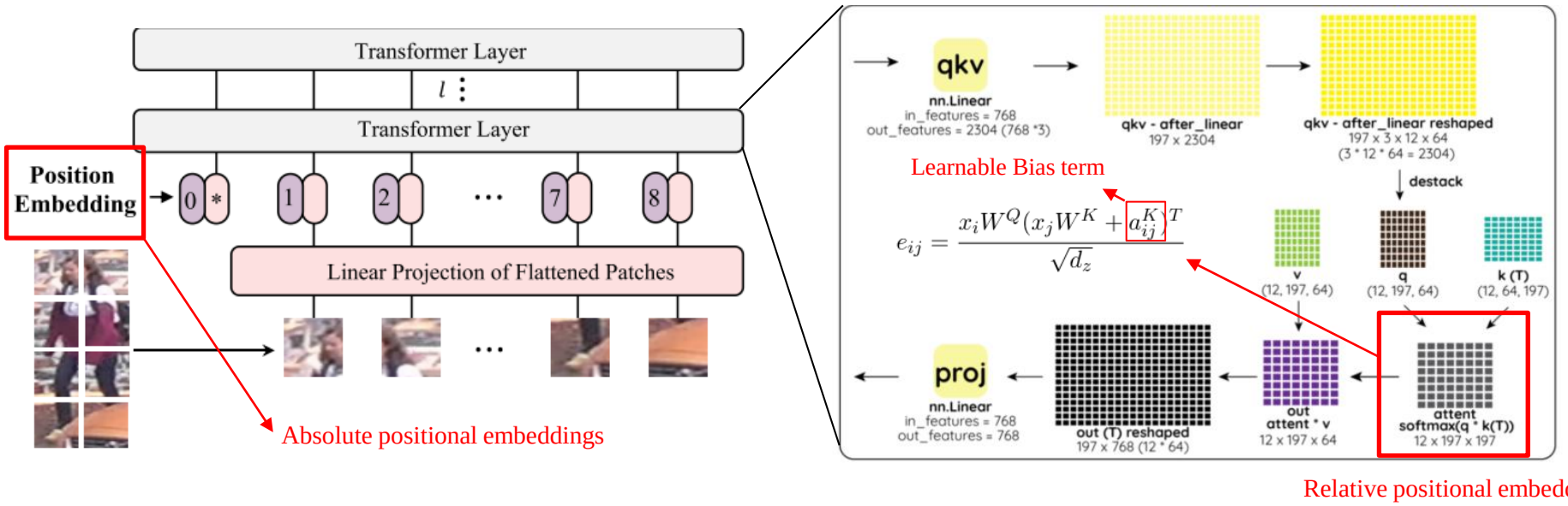
d : distance function

m : margin

  - Weight를 element-wise로 곱하는 것의 의미
    - Weight vector는 [CLS]와 patch token간 similarity를 기반으로 구함
    - Model이 triplet loss의 학습과정에서 Anchor sample과 Positive sample이 공통적으로 집중하는 patch들의 dominant feature를 제외한 나머지 부분을 효과적으로 배울수 있도록 지도

Proposed method 2

- *Relative position Jensen-Shannon Divergence loss*



- *Attention score에서 두 patch 간의 position 관계를 담고 있는 term만을 계산하여, 각 sample에서 모델이 집중하는 정보를 modeling*

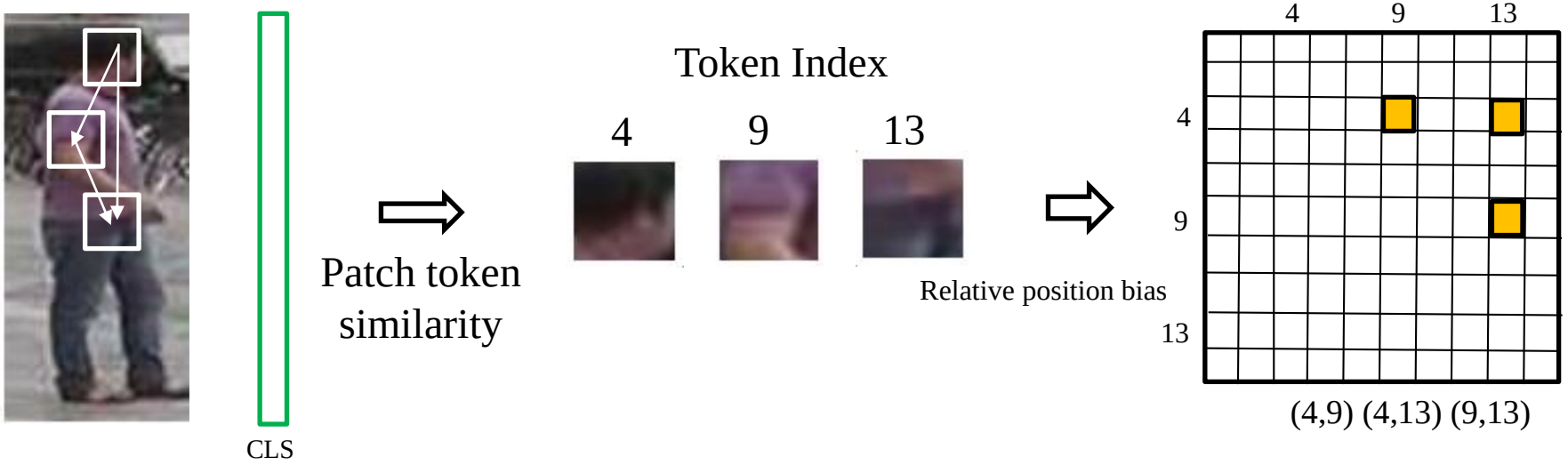
$$\text{Attention score} = (Q_1 + P_{abs1})(K_2 + P_{abs2})^T + P_{rel12}$$

Positional relationship between patch 1 and 2

$$\rightarrow P_{abs1} \cdot {P_{abs2}}^T + P_{rel12}$$

Proposed method 2

- *Relative position Jensen-Shannon Divergence loss*



- *Algorithm*

1. *Anchor, Positive 각각에 대해 CLS token과 similarity가 높은 N개의 patch 선별 (sorted)*
2. *N개의 patch에 대해 가능한 combination* ${}_N C_2$ *개의 쌍 (a,b)에 대해 positional relation:* $P_{abs\_a} \cdot P_{abs\_b}{}^T + P_{rel\_ab}$*를 계산하여 원소의 개수가* ${}_N C_2$ *개인 patch distance vector (ex: [relation(1,2), relation(1,3), relation(2,3)]) 생성*
3. *Anchor, Positive의 positional relation vector에 대해 softmax를 취해 probablity distribution으로 만든 후, Jensen-Shannon divergence loss를 통해 두 분포가 유사하도록 학습*

Proposed method 2

- *Relative position Jensen-Shannon Divergence loss*

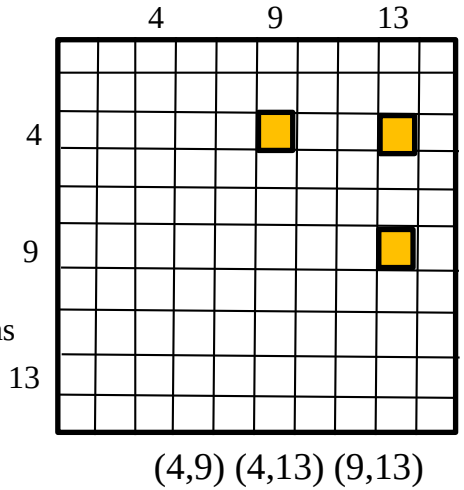
CLS

Patch token similarity

Token Index

4 9 13

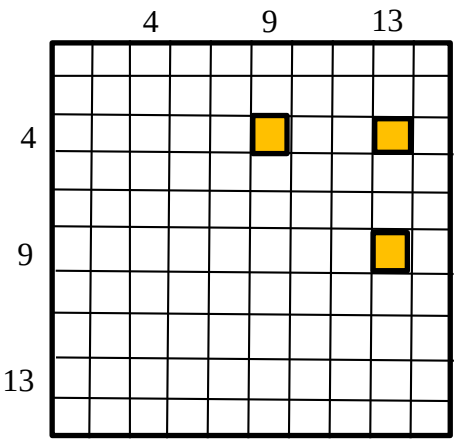
Relative position bias



(4,9) (4,13) (9,13)

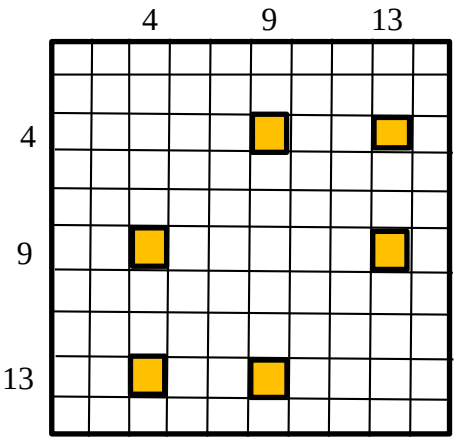
- *Details*
  - *Combinations VS Perturbation*

$_3C_2$



(4,9) (4,13) (9,13)

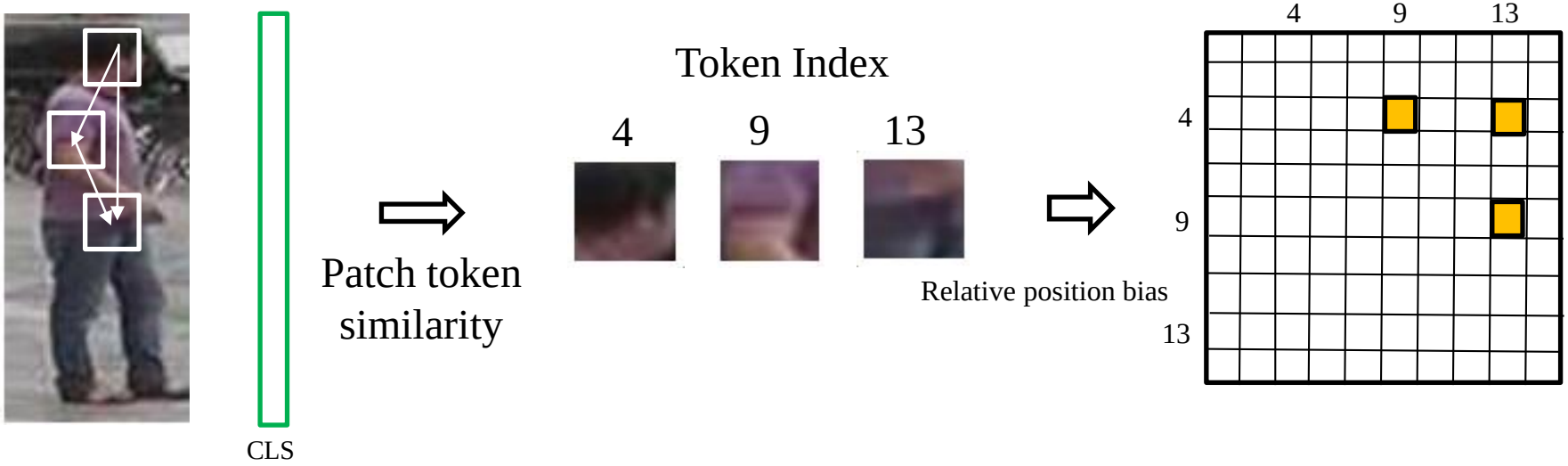
$_3P_2$



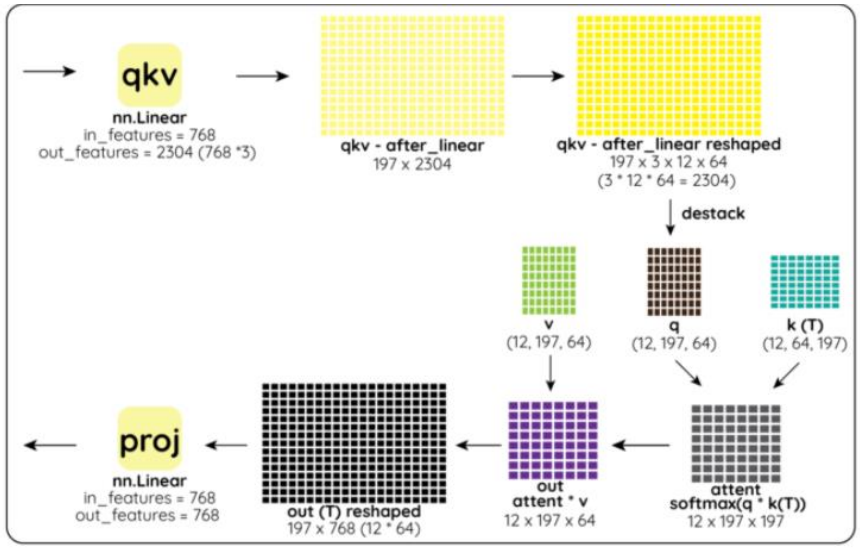
(4,9) (4,13) (9,4)
(9,13) (13,4) (13,9)

Proposed method 2

- *Relative position Jensen-Shannon Divergence loss*



- *Details*
  - *Number of relative position embedding bias*
    - *Num_heads * Num_layers*
    - *Head wise*
  - *Number of Patches*
    - *N개를 뽑을때 조합의 개수는 ${}_NC_2$*
  - *Size of distribution vector*
    - Head wise : ${}_NC_2$,*Num_heads * Num_layers)*
    - *Layer wise : ${}_NC_2$ * Num_heads , Num_layers)*



Proposed method 2

- *Relative position Jensen-Shannon Divergence loss*

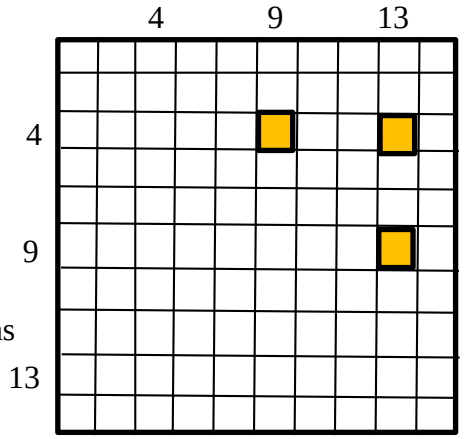
CLS

Patch token similarity

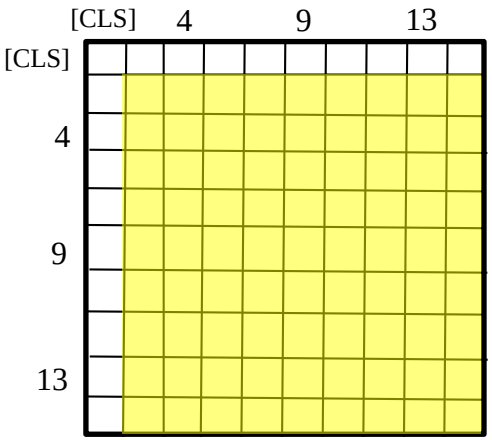
Token Index

4 9 13

Relative position bias



- *Details*
  - *Relative Position of CLS token*
    - *CLS token의 Absolute, Relative position 은 어떤 의미를 갖는가?*



Experimental results

- *Market1501*

Method	mAP	R1
TransReID-SSL(naïve triplet)	93.2	96.7
Proposed 1	93.59	96.7
Proposed 1 + RPE	93.67	96.85
Proposed 1 + Proposed 2	93.73	96.82

- *Occluded – Duke*

Method	mAP	R1
TransReID-SSL(naïve triplet)	63.77	73.08
Proposed 1 + Proposed 2	64.62	73.85