김성수

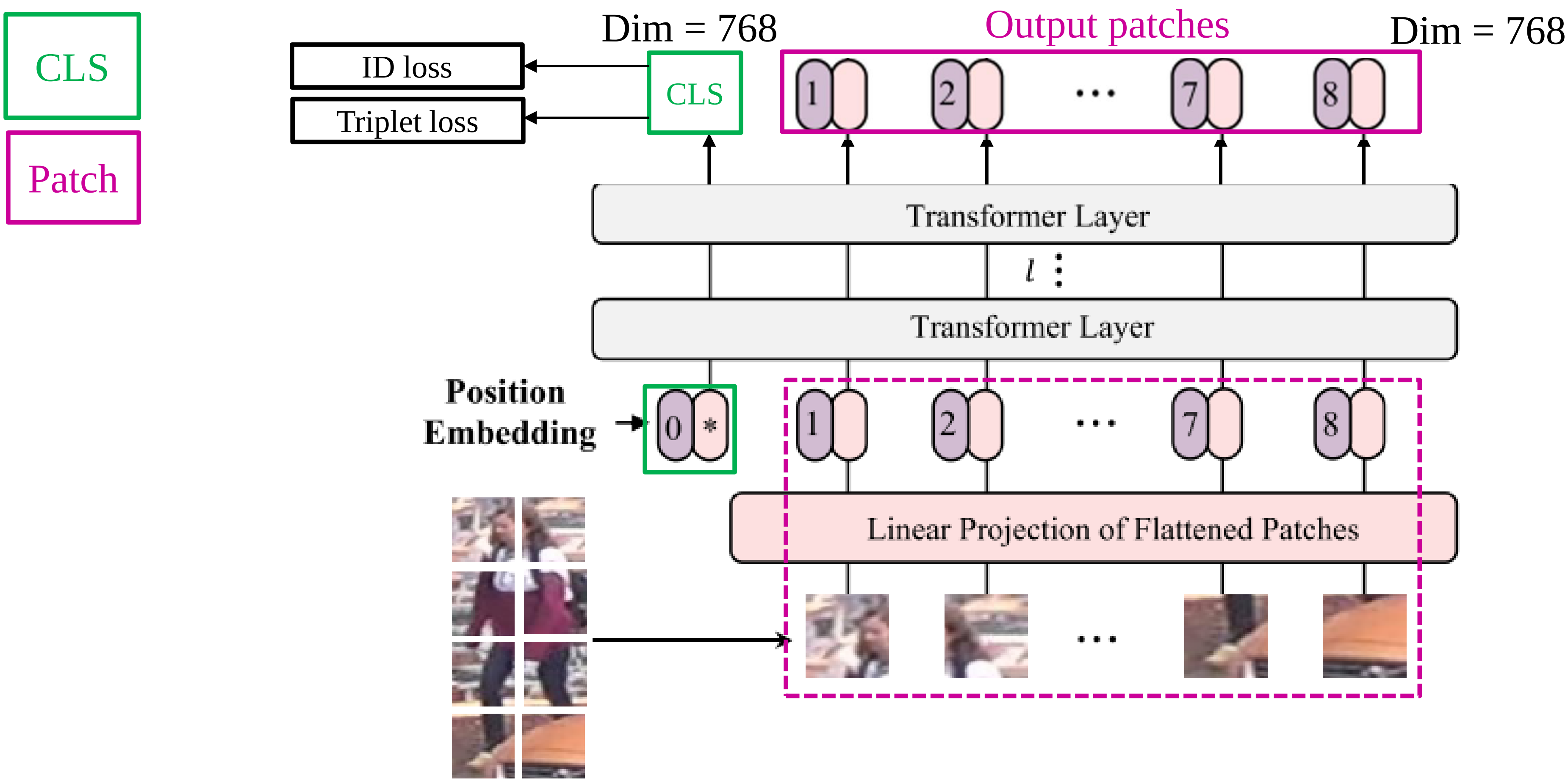
This week

- *Relation between CLS token & Patch tokens*

김성수

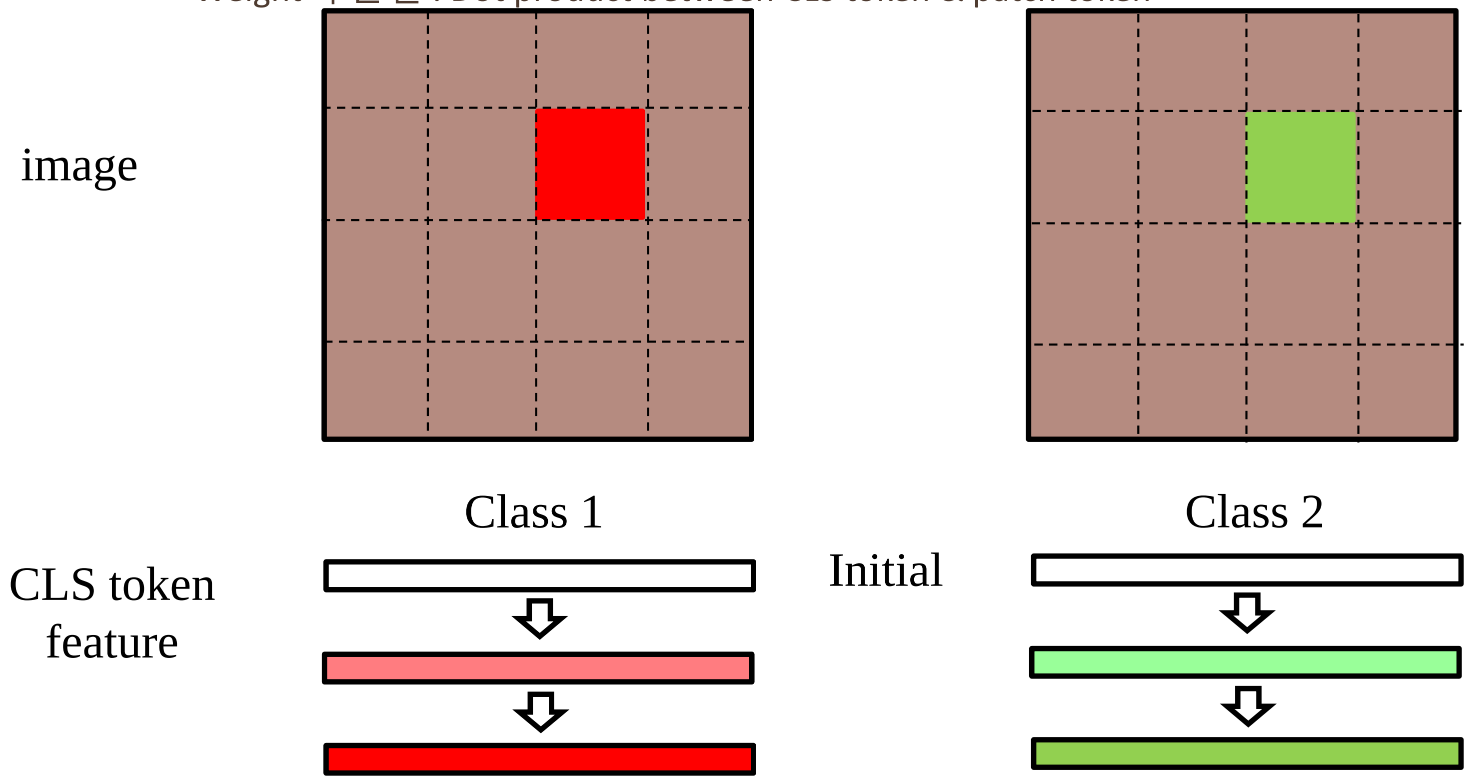
Transformer based Re-ID

- *CLS token 을 이용한 Triplet loss와 ID loss(Cross-entropy loss) 계산*



Transformer based Re-ID

- CLS token
  - Patch token의 weighted summation
    - Weight의 본질 : Dot product between CLS token & patch token

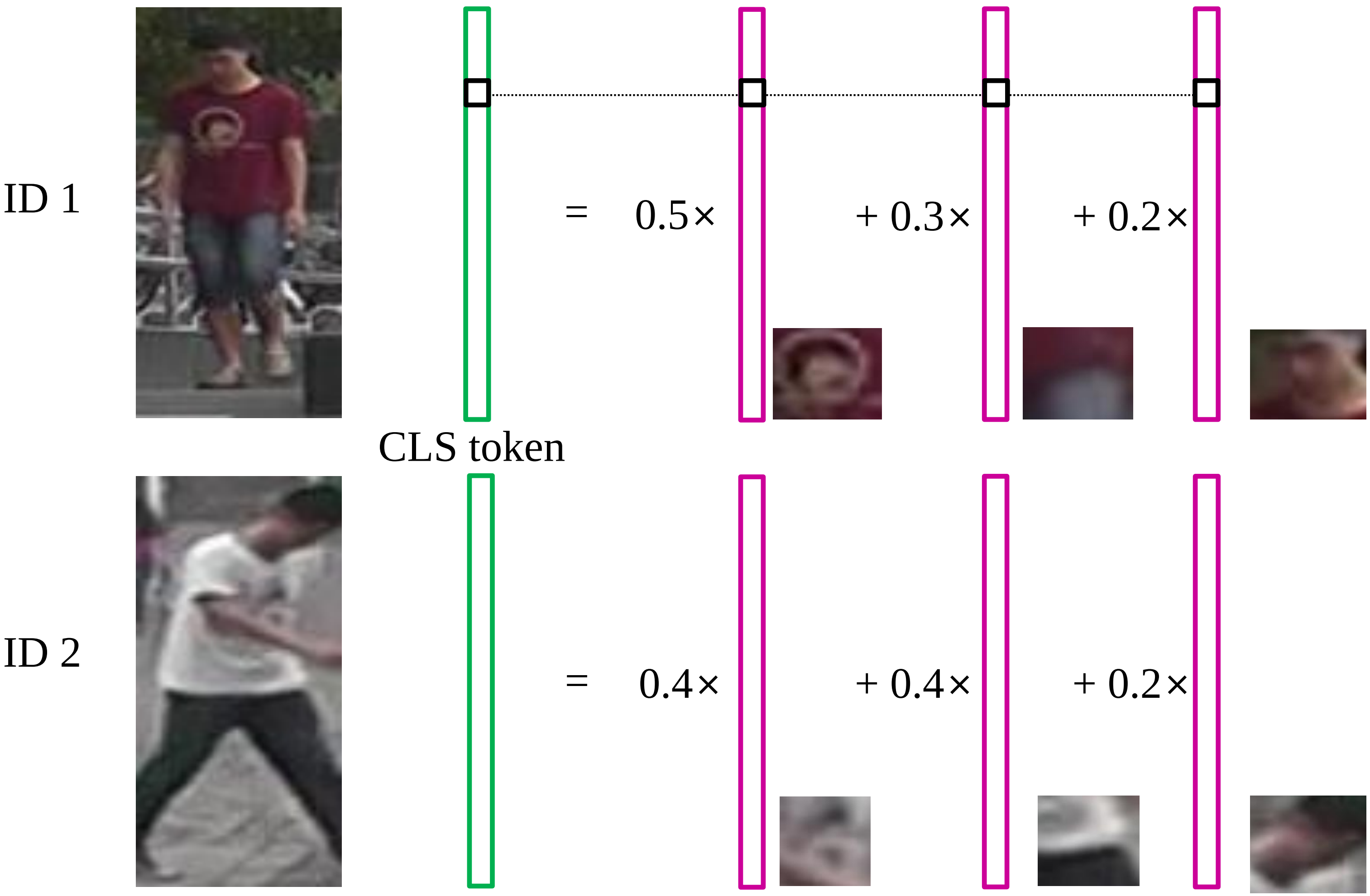


김성수

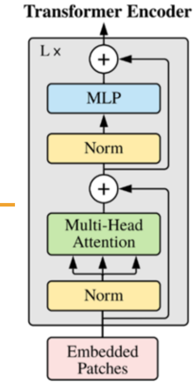
Transformer based Re-ID

CLS Patch

- CLS token
  - Patch token의 weighted summation



Vision Transformer[1]

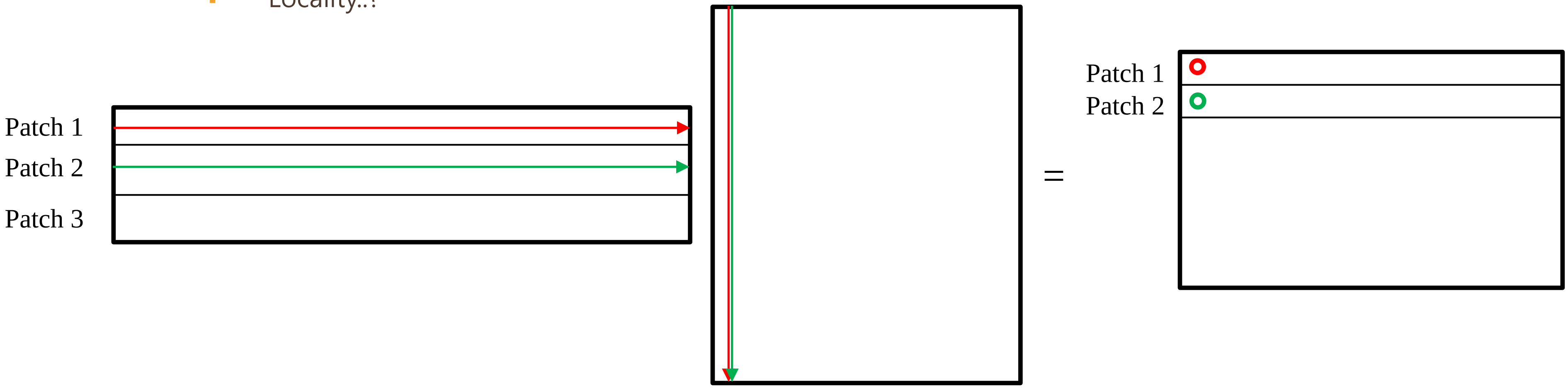


- Details

  3. Linear layer in MHSA & MLP layer

Inductive bias. We note that Vision Transformer has much less image-specific inductive bias than CNNs. In CNNs, locality, two-dimensional neighborhood structure, and translation equivariance are baked into each layer throughout the whole model. In ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global. The two-dimensional neighborhood

  - Translationally equivariant = weight sharing
  - Locality..?



  - Patch가 만들어진 원래 image 입장에서
    - 모든 patch들이 같은 weight를 share하며(patch의 순서에 상관없이)feature를 refine하기 때문에 translationally equivariant
    - MLP에서의 feature refining 과정은 patch 각각에 대해 수행되기 때문에 local

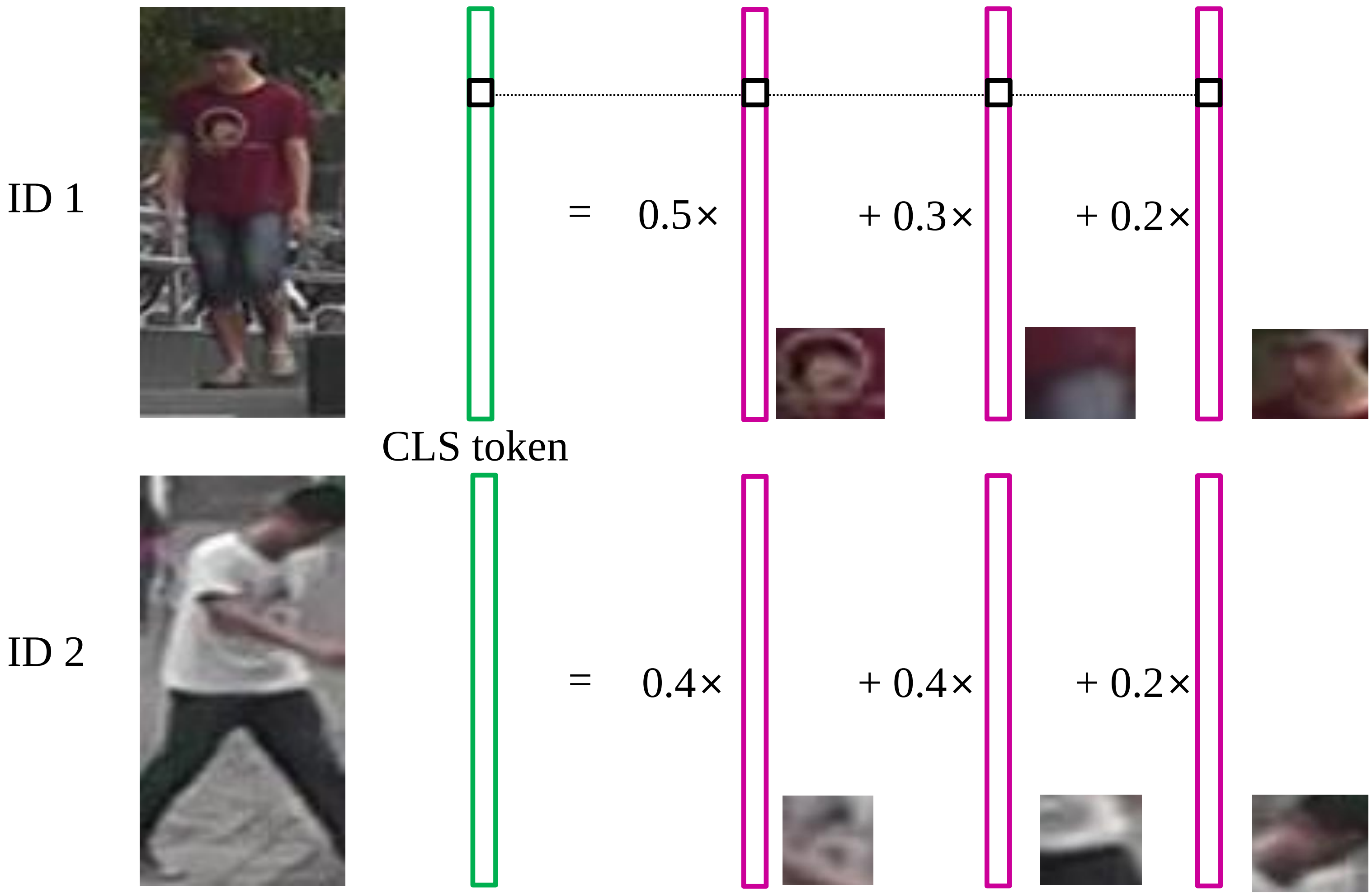
김성수

Transformer based Re-ID

CLS Patch

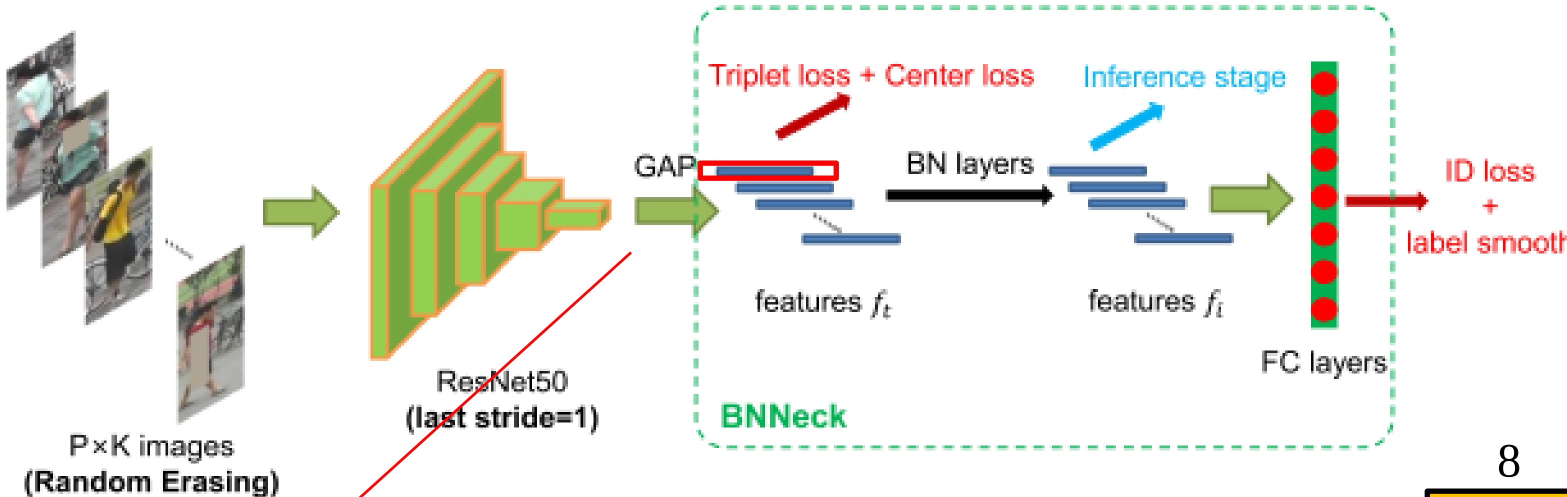
- Triplet loss
  - Euclidean distance

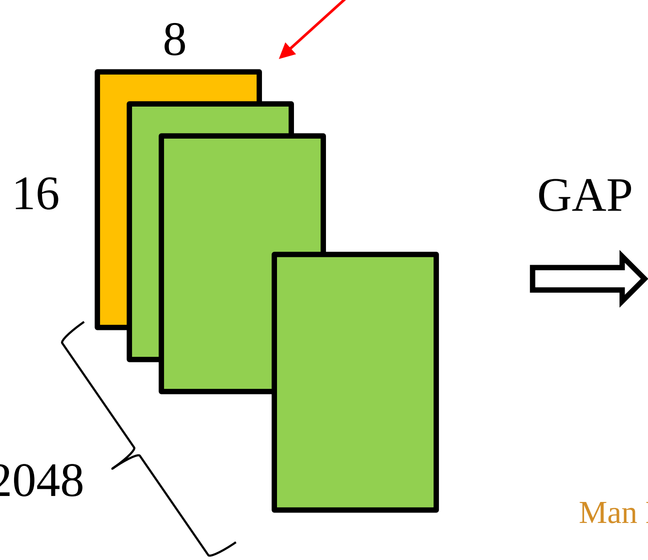
$$\mathcal{L}(A, P, N) = \max\left(\| \mathrm{f}(A) - \mathrm{f}(P) \|^2 - \| \mathrm{f}(A) - \mathrm{f}(N) \|^2 + \alpha, 0 \right)$$

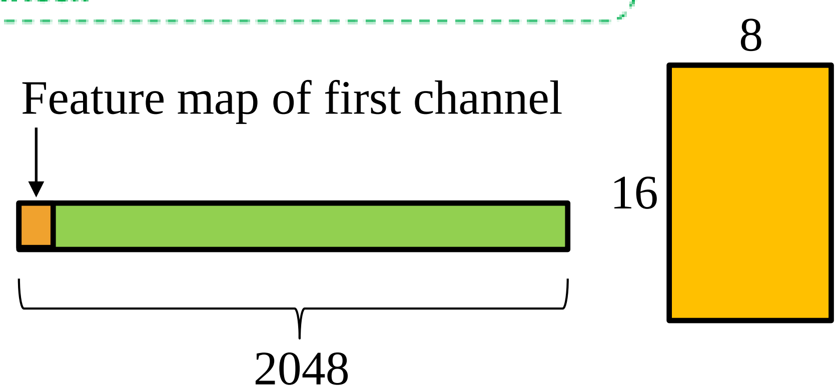


CNN features

- CNN based Re-ID model







Note:

각 dimension의 값은 Image 전체에 대한 특정 feature을 대표함

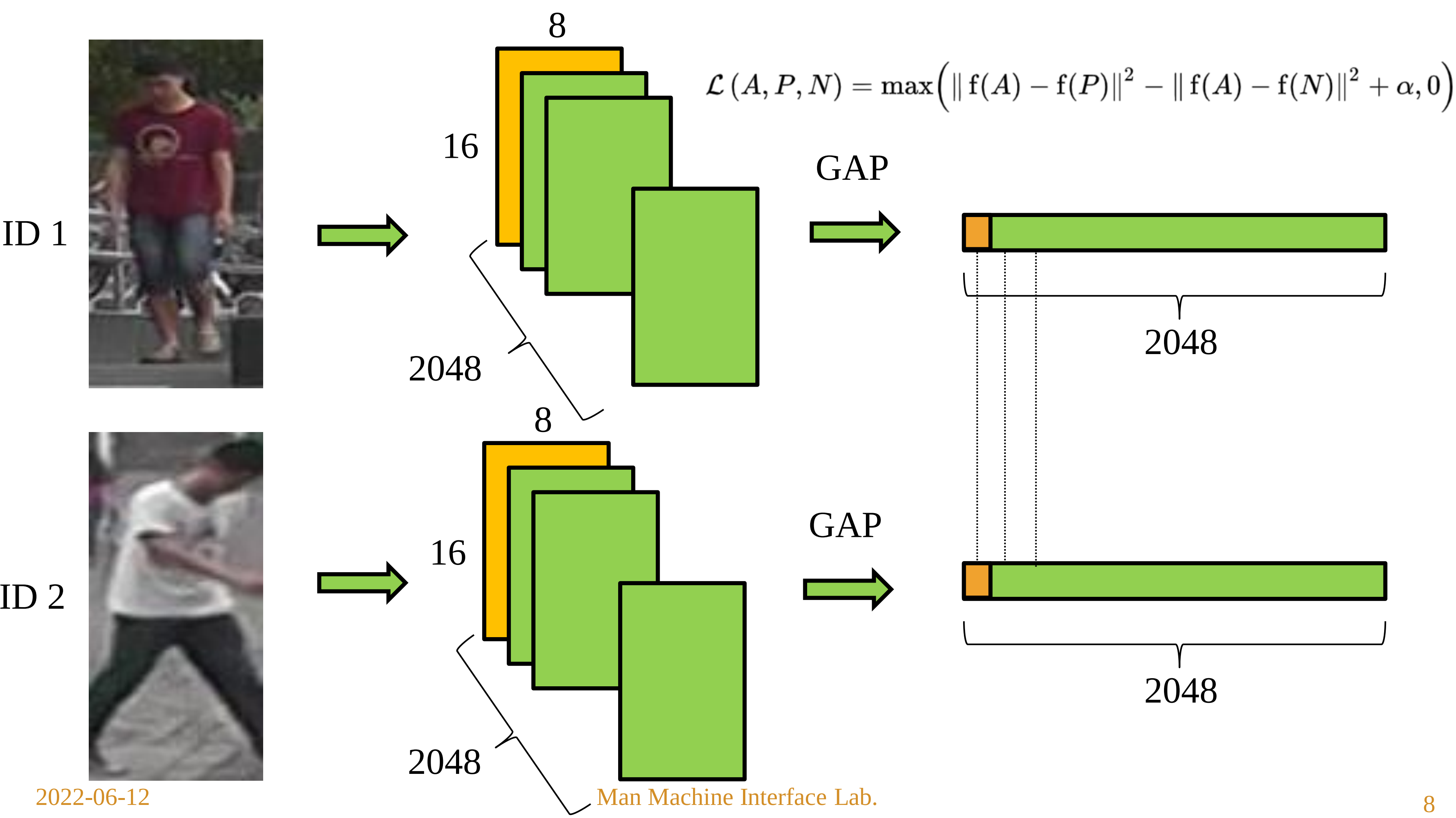
김성수

CNN features

Note:

각 dimension의 값은 Image 전체에 대한 특정 feature을 대표함

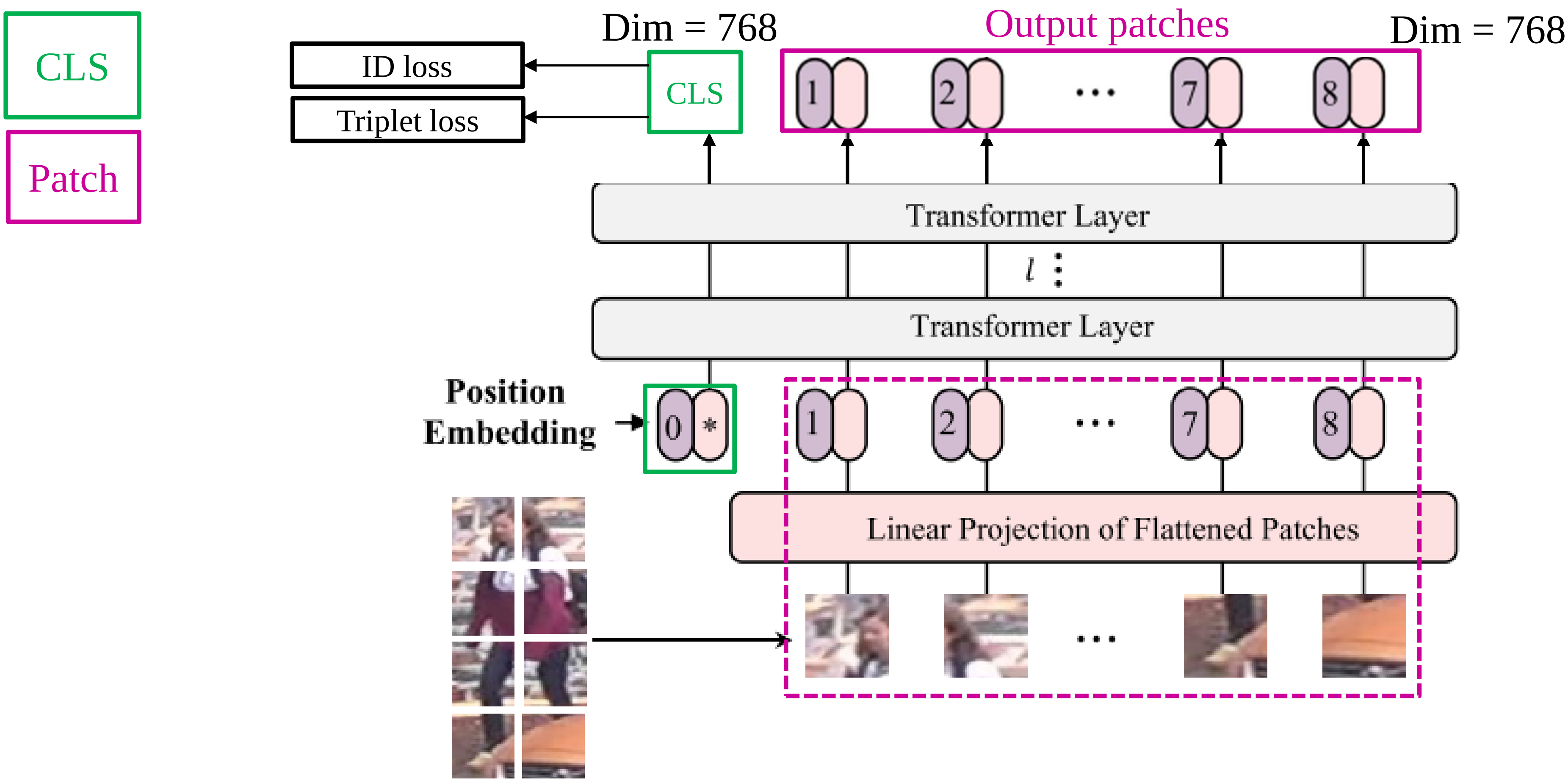
- CNN based Re-ID model



Transformer based Re-ID

- 문제의 point

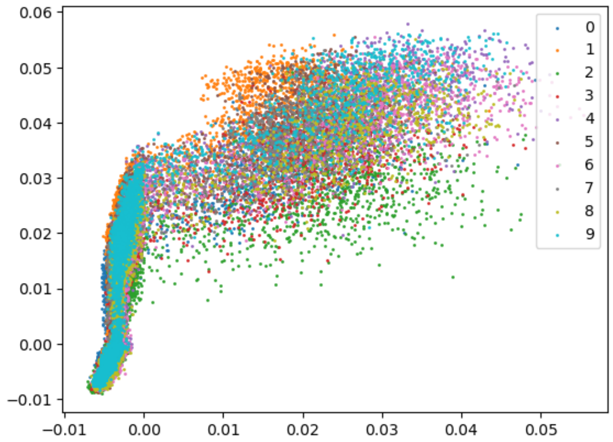
CNN의 output feature에 사용되던 Triplet loss 를 , ViT의 CLS token에 그대로 적용하는 것이 ViT를 이용한 metric learning에서 최선인가?



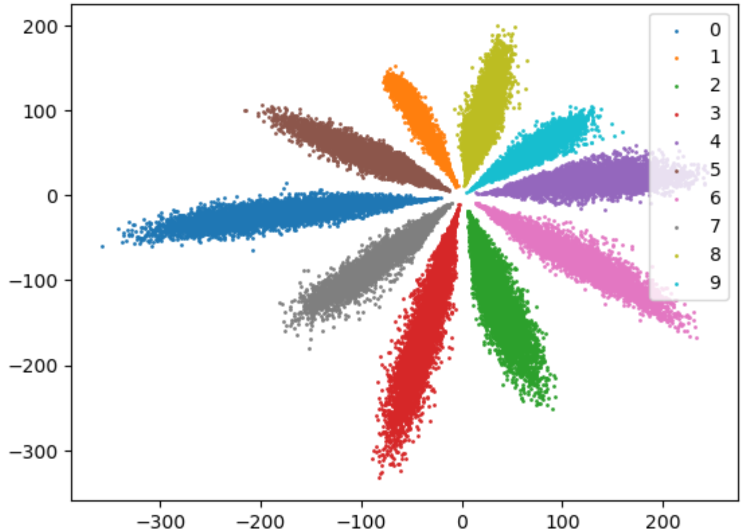
김성수

Back to the basic

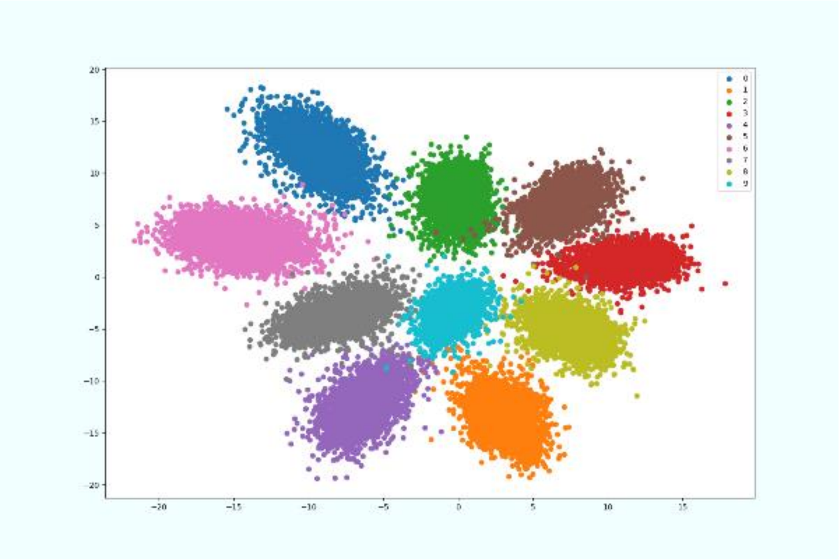
- Embedding space



Initial feature distribution



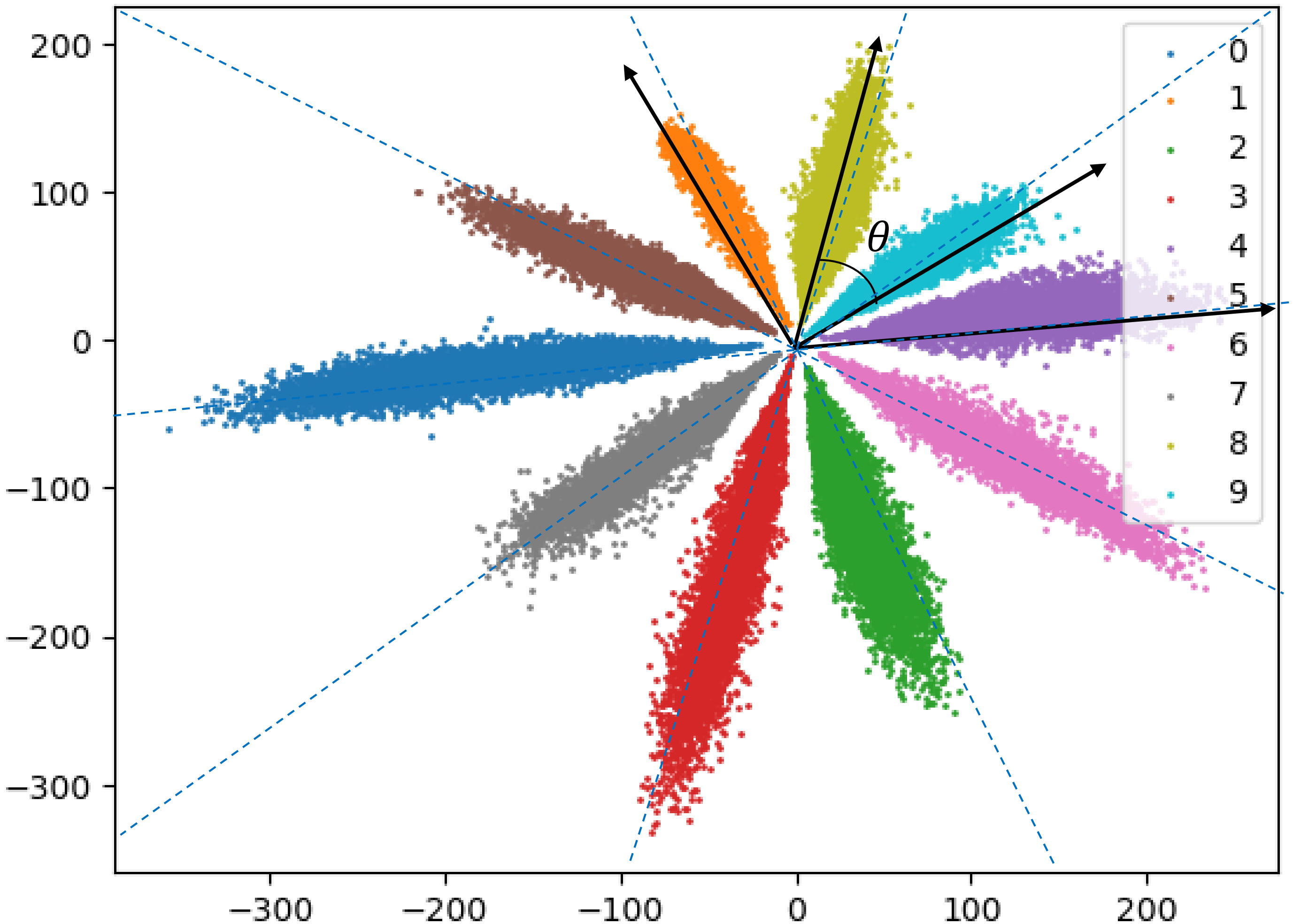
After 50 epoch (ID loss)



After 50 epoch (Triplet loss)

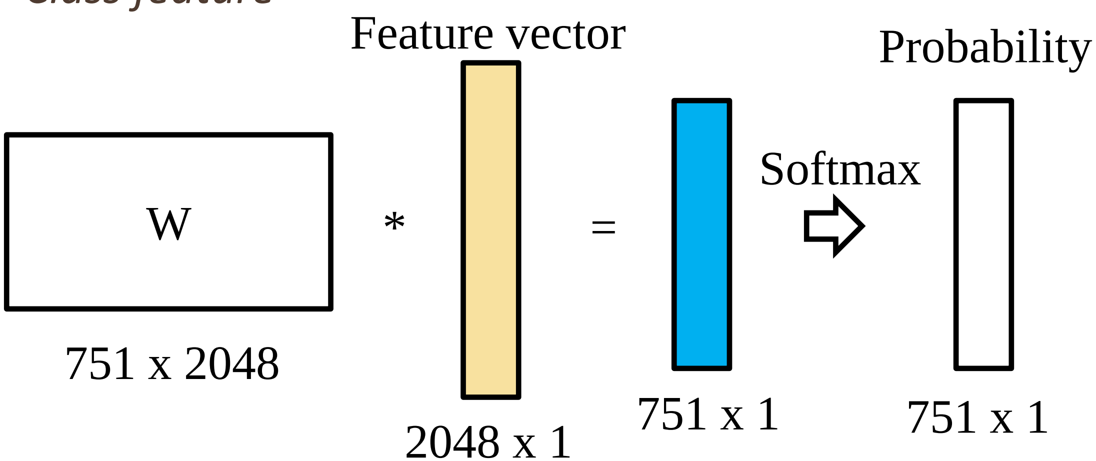
Back to the basic

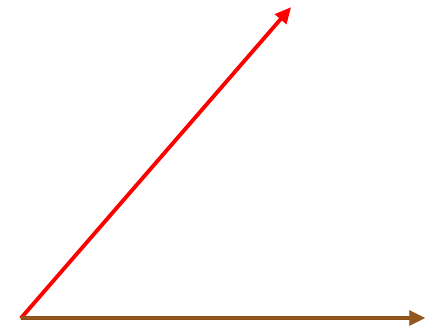
- Embedding space(ID Loss)



Back to the basic

- *Class feature*

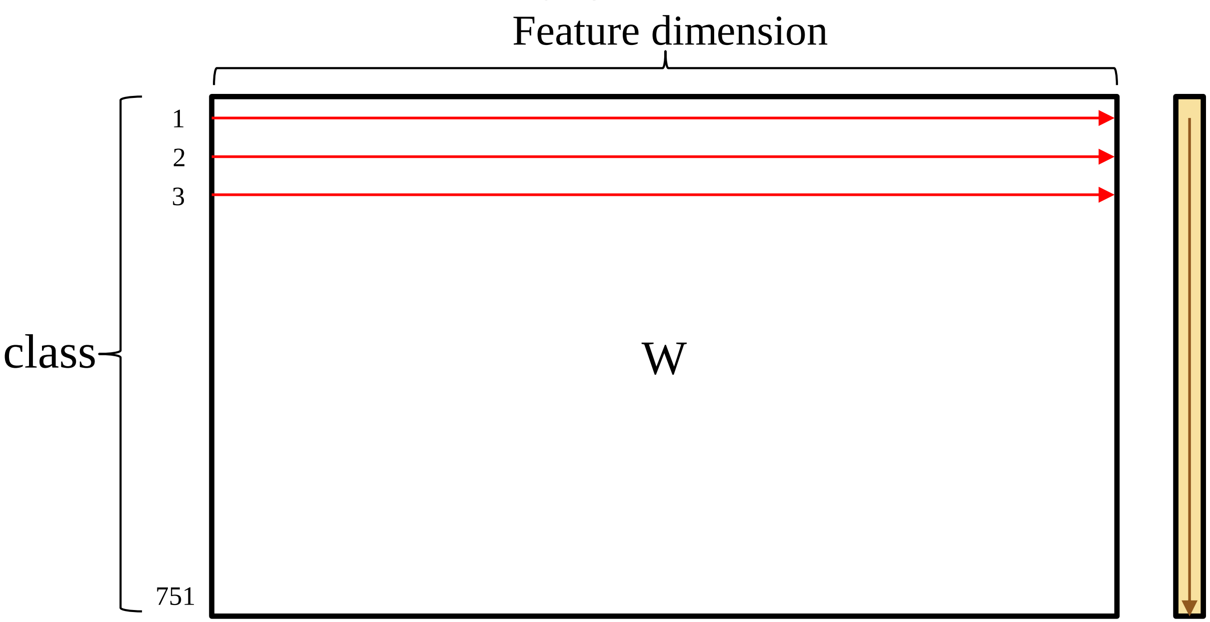




$$W_j^T x_i = \|W_j\| \|x_i\| cos\theta_j$$

j : class index

i : sample index

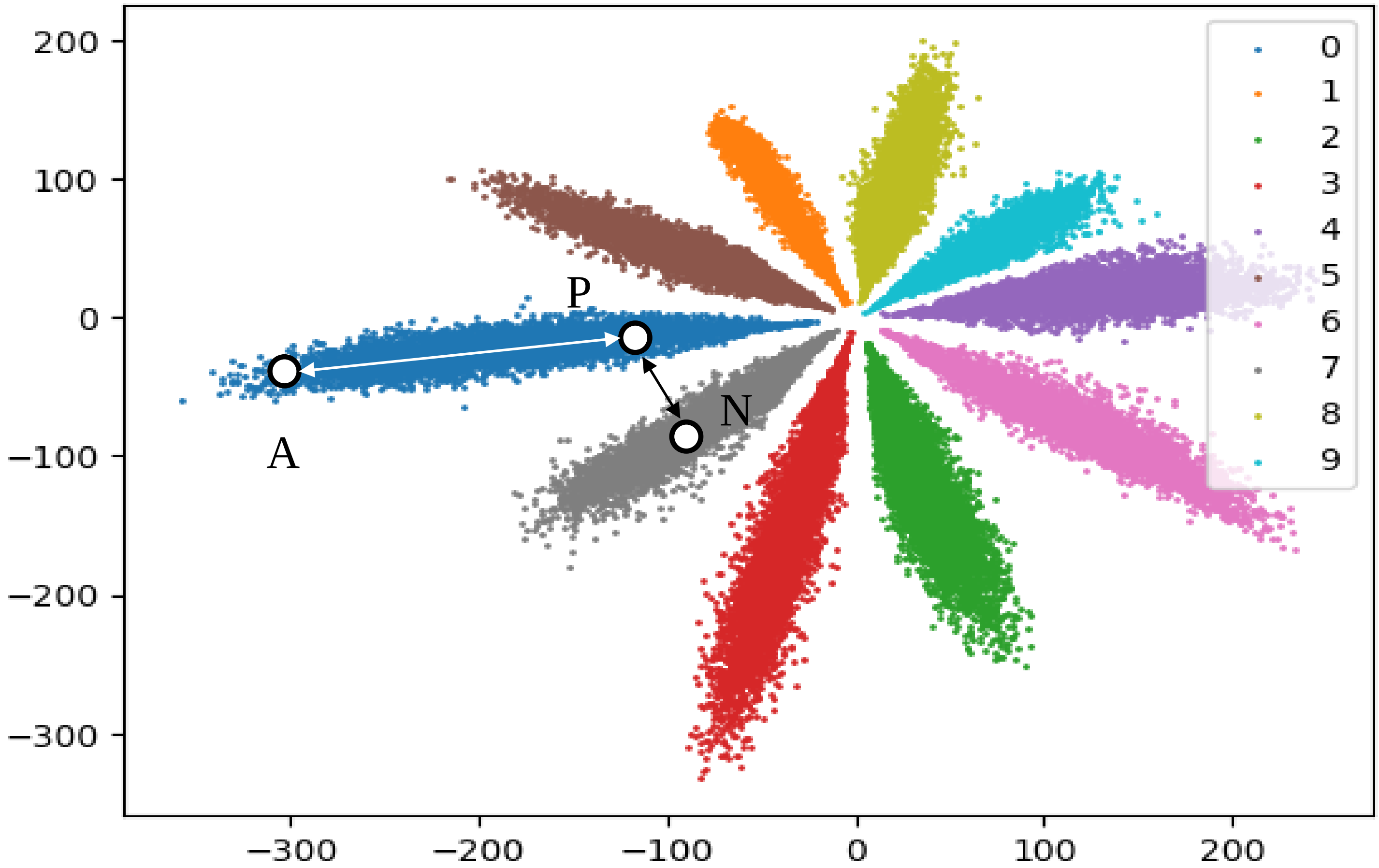


*Note

Training이 거듭될수록, Weight vector의 각 row는 해당 class를 대표하는 feature vector의 값에 가깝게 update됨

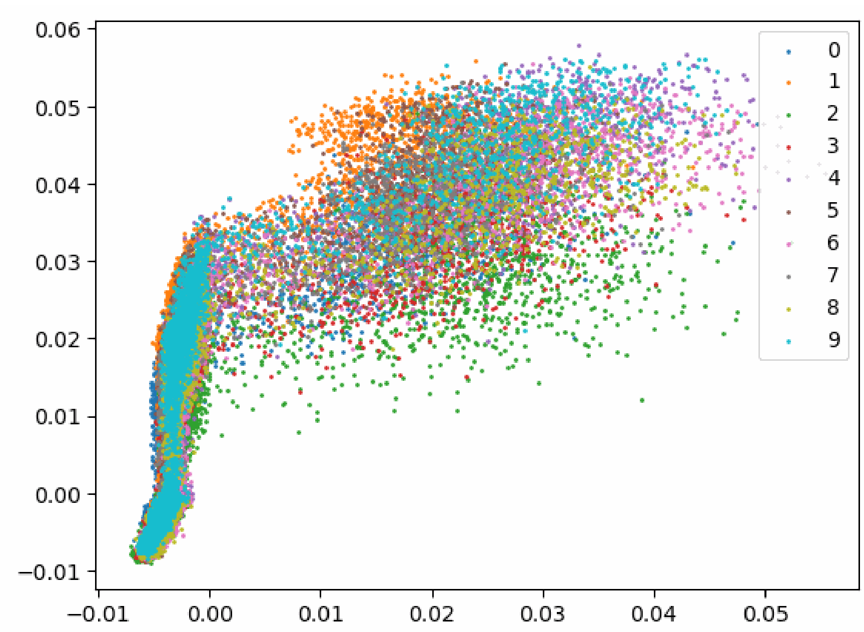
Back to the basic

- Embedding space(ID loss+Triplet Loss) $\mathcal{L}(A,P,N) = \max\left(\|\mathrm{f}(A)-\mathrm{f}(P)\|^2 - \|\mathrm{f}(A)-\mathrm{f}(N)\|^2 + \alpha, 0\right)$

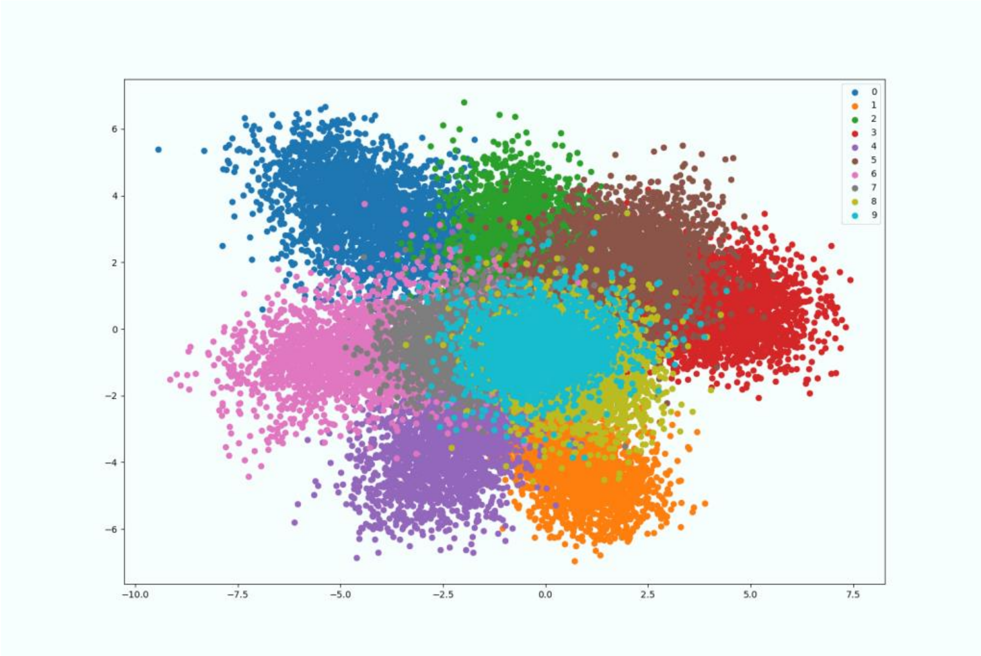


Back to the basic

- Embedding space(ID loss+Triplet Loss)



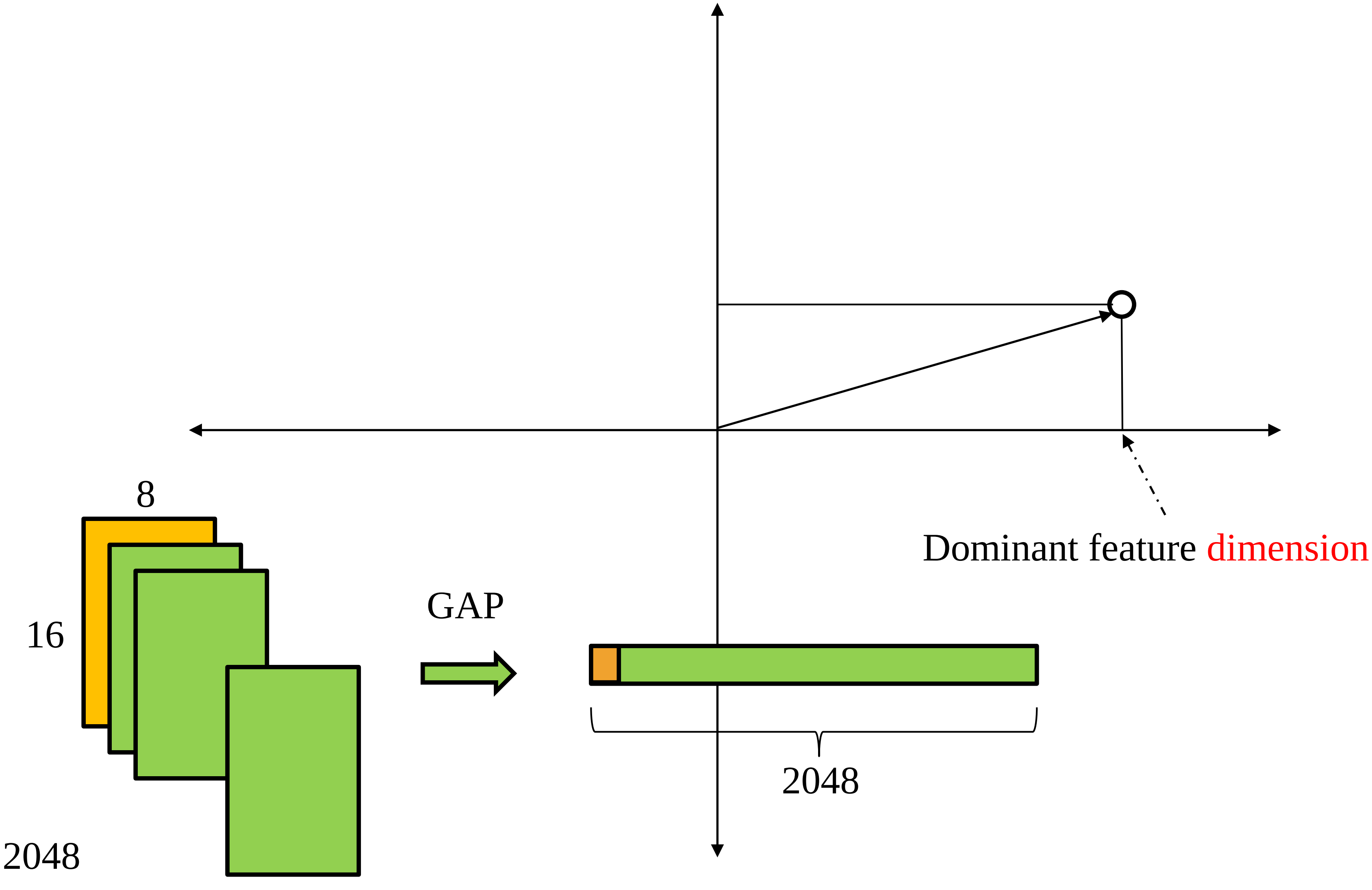
ID Loss



ID + Triplet Loss

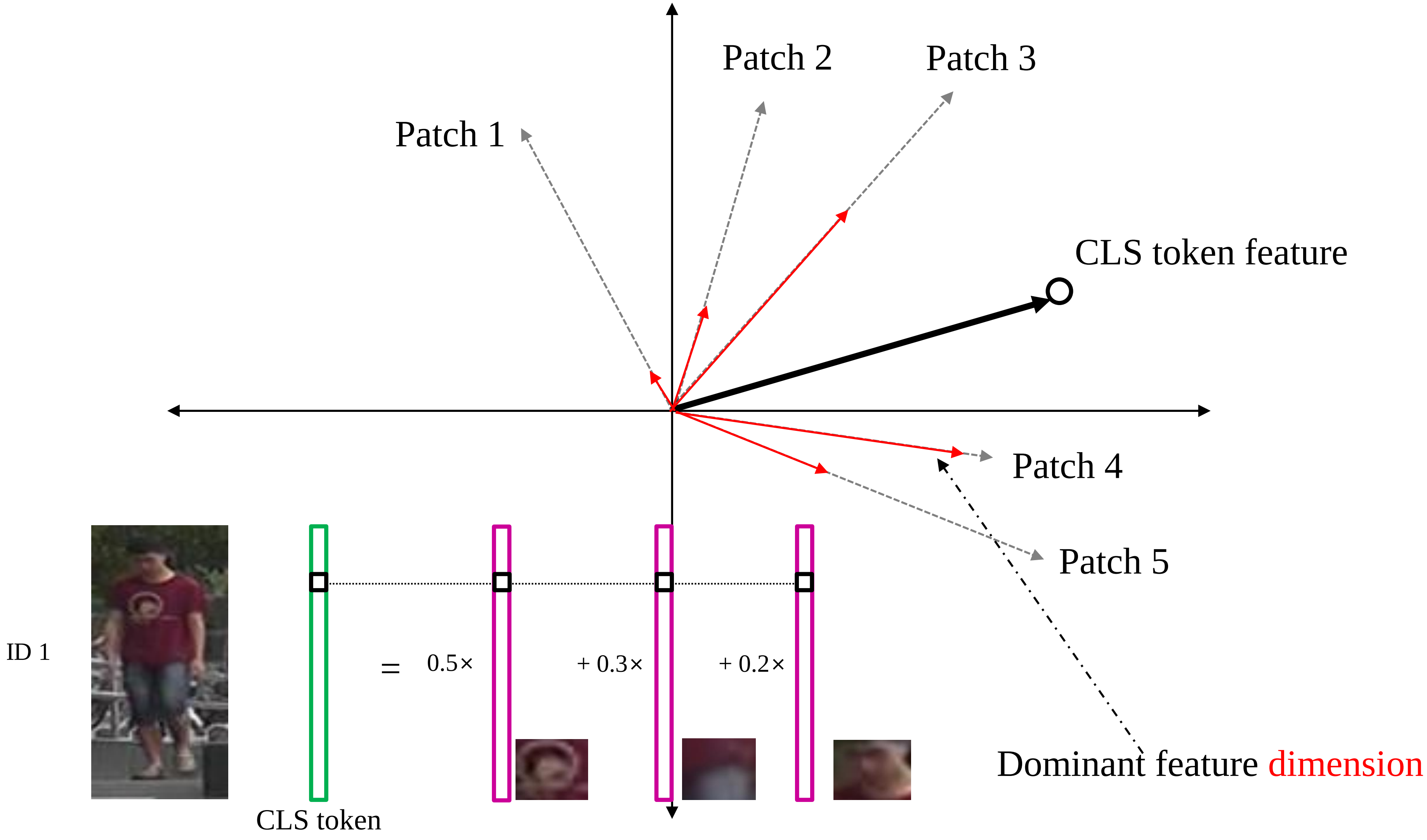
CNN vs ViT

- Embedding space(CNN)



CNN vs ViT

- Embedding space(ViT)

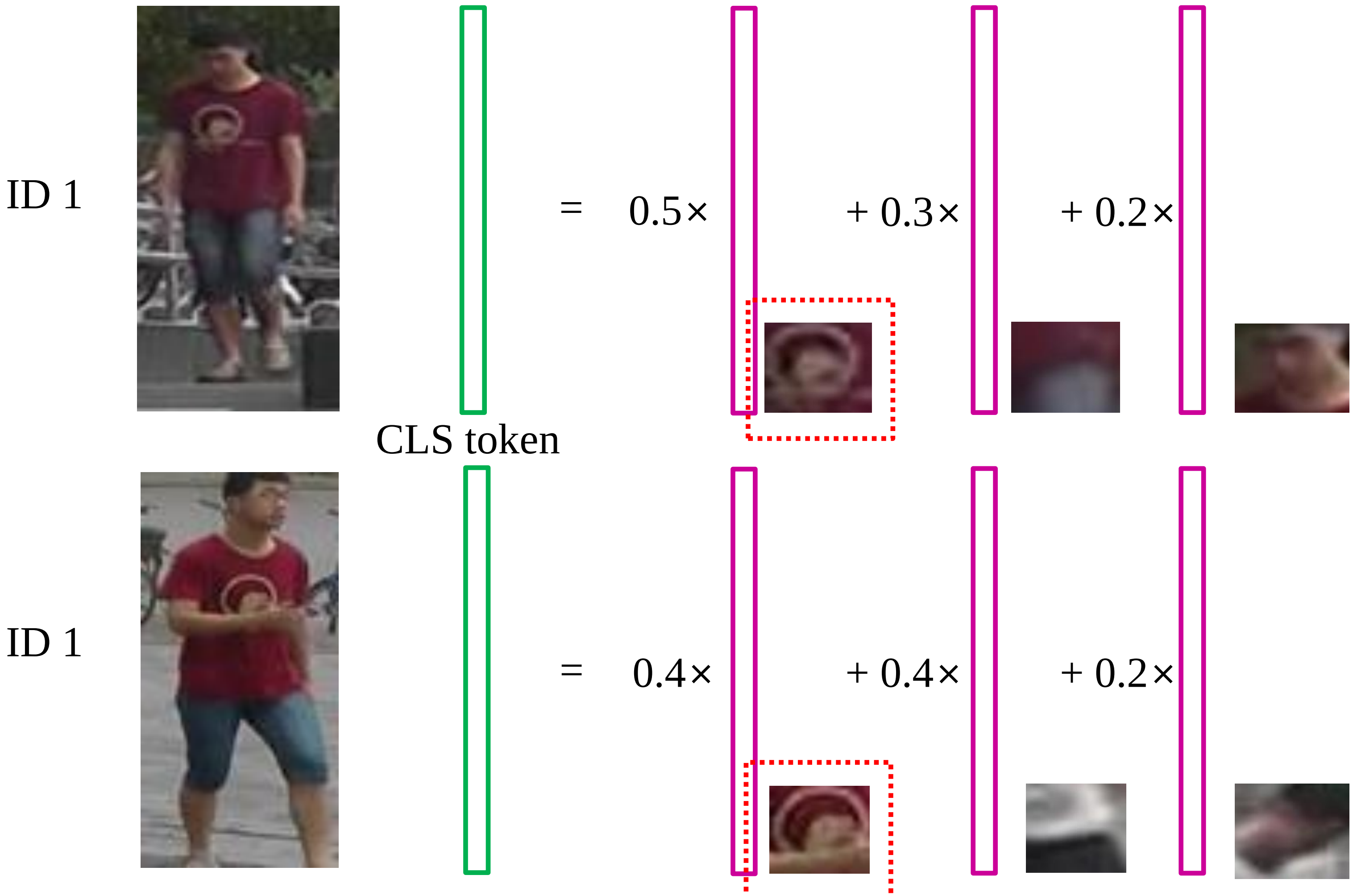


김성수

Transformer based Re-ID

CLS Patch

- 같은 ID의 sample들에 대해 각 CLS token이 공통적으로 집중하는 patch token을 찾을 수 있는가?



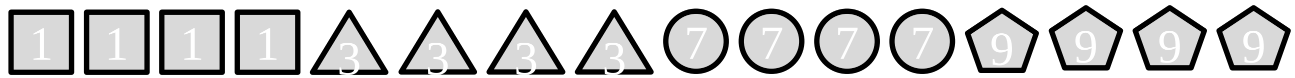
Transformer based Re-ID

- 같은 ID의 sample들에 대해 각 CLS token이 공통적으로 집중하는 patch token을 찾을 수 있는가?

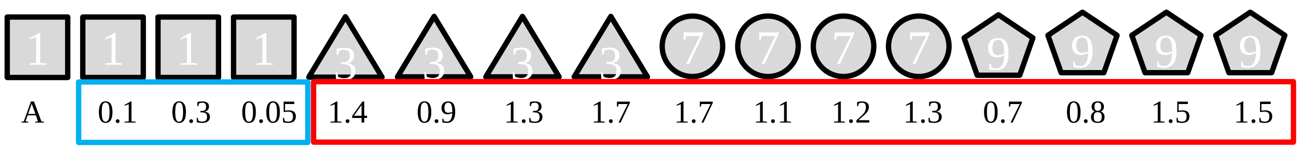



Transformer based Re ID

- Triplet hard mining
  - Ex) Batch size 16



  - Compute distance matrix for each pair → 16 x 16
  - Anchor : Current sample
  - Positive : The most disimilar sample among which have same ID with anchor
  - Negative : The most similar sample among which have different ID with anchor



Triplet

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

r_a, r_p, r_n : sample representations

d : distance function

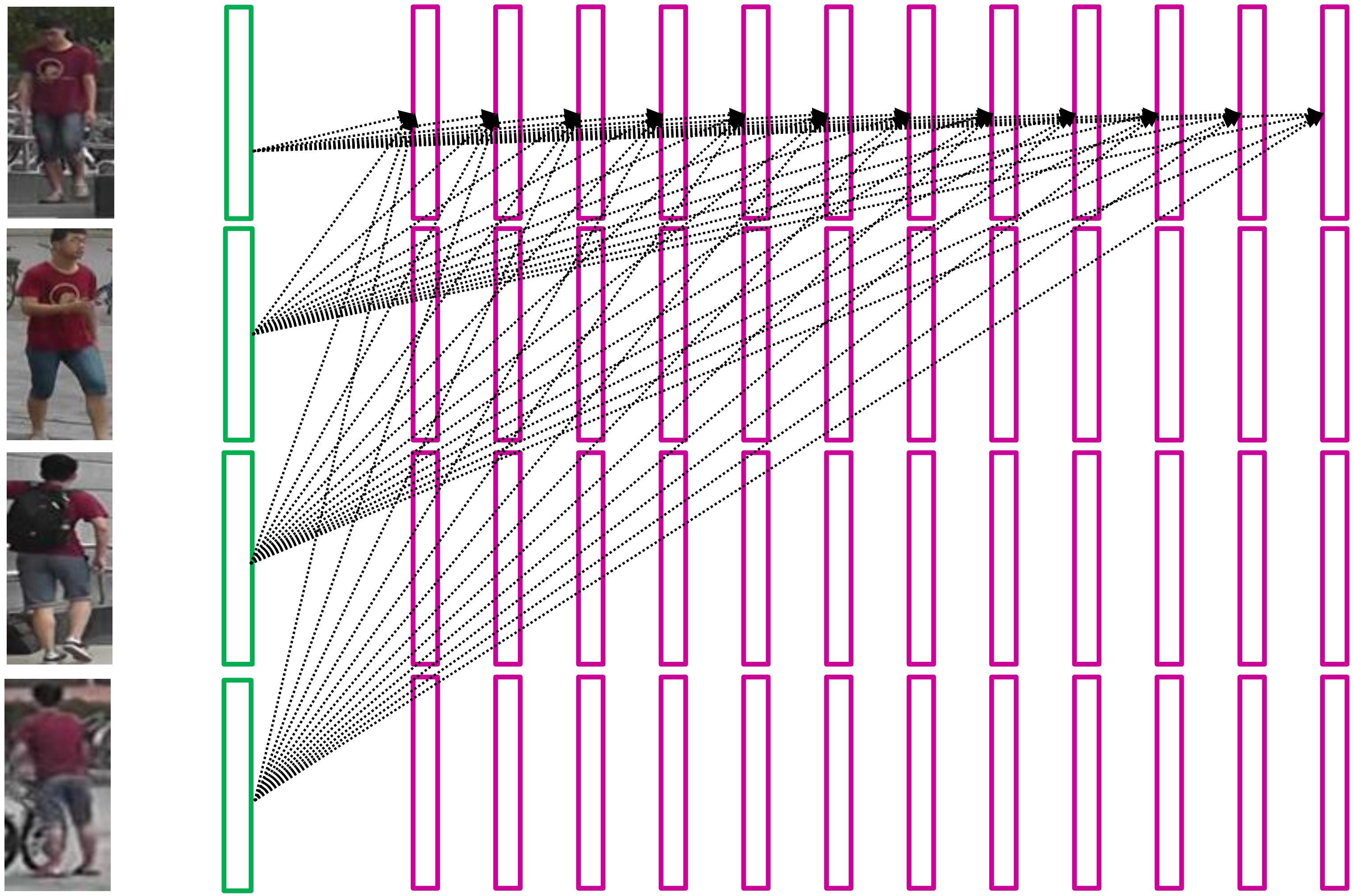
김성수

Transformer based Re-ID

CLS

Patch

- 같은 ID의 sample들에 대해 각 CLS token이 공통적으로 집중하는 patch token을 찾을 수 있는가?



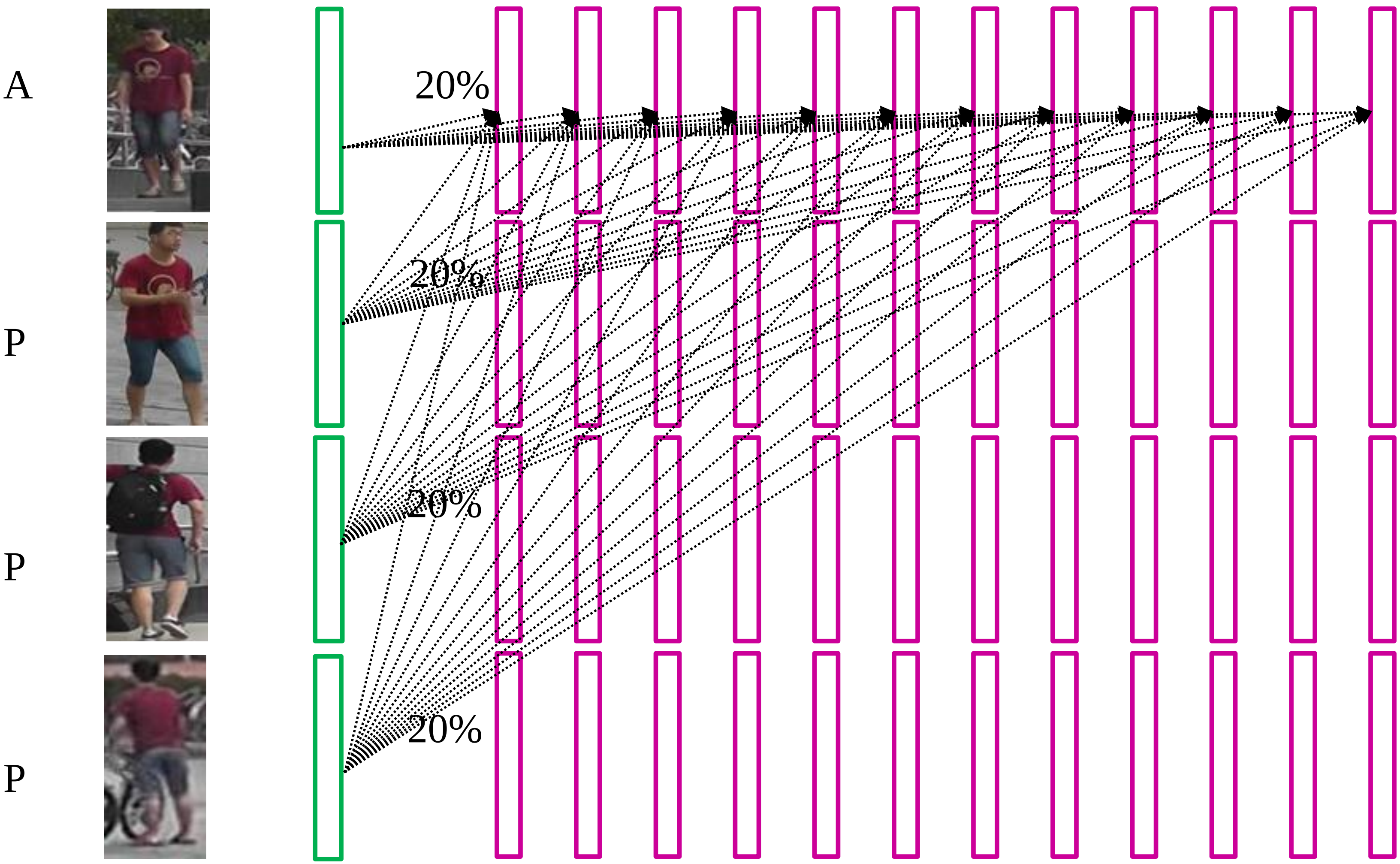
김성수

Transformer based Re-ID

CLS

Patch

- 같은 ID의 sample들에 대해 각 CLS token이 공통적으로 집중하는 patch token을 찾을 수 있는가?



Weighted Triplet loss for CLS token

- 정리
  - ViT를 이용해서 metric learning을 하고자 할때, CLS token만을 이용한 triplet learning이 과연 최선인가?
    - 1. Positive sample의 CLS token과 Anchor의 patch token들간의 similarity를 이용하여, Class를 대표하는 dominant patch token을 찾기
    - 2. 많은 positive sample과 similarity가 높은 patch token에 더 큰 weight를 부여하여 global average pool