UniCLIP: Unified Framework for Contrastive Language-Image Pretraining

[Abstract]

UniCLIP integrates the contrastive loss of both inter-domain pairs and intra-domain pairs into a single universal space. The discrepancies that occur when integrating contrastive loss between different domains are resolved by the three key components of UniCLIP:

(1) augmentation-aware feature embedding

(2) MP-NCE loss

(3) domain dependent similarity measure

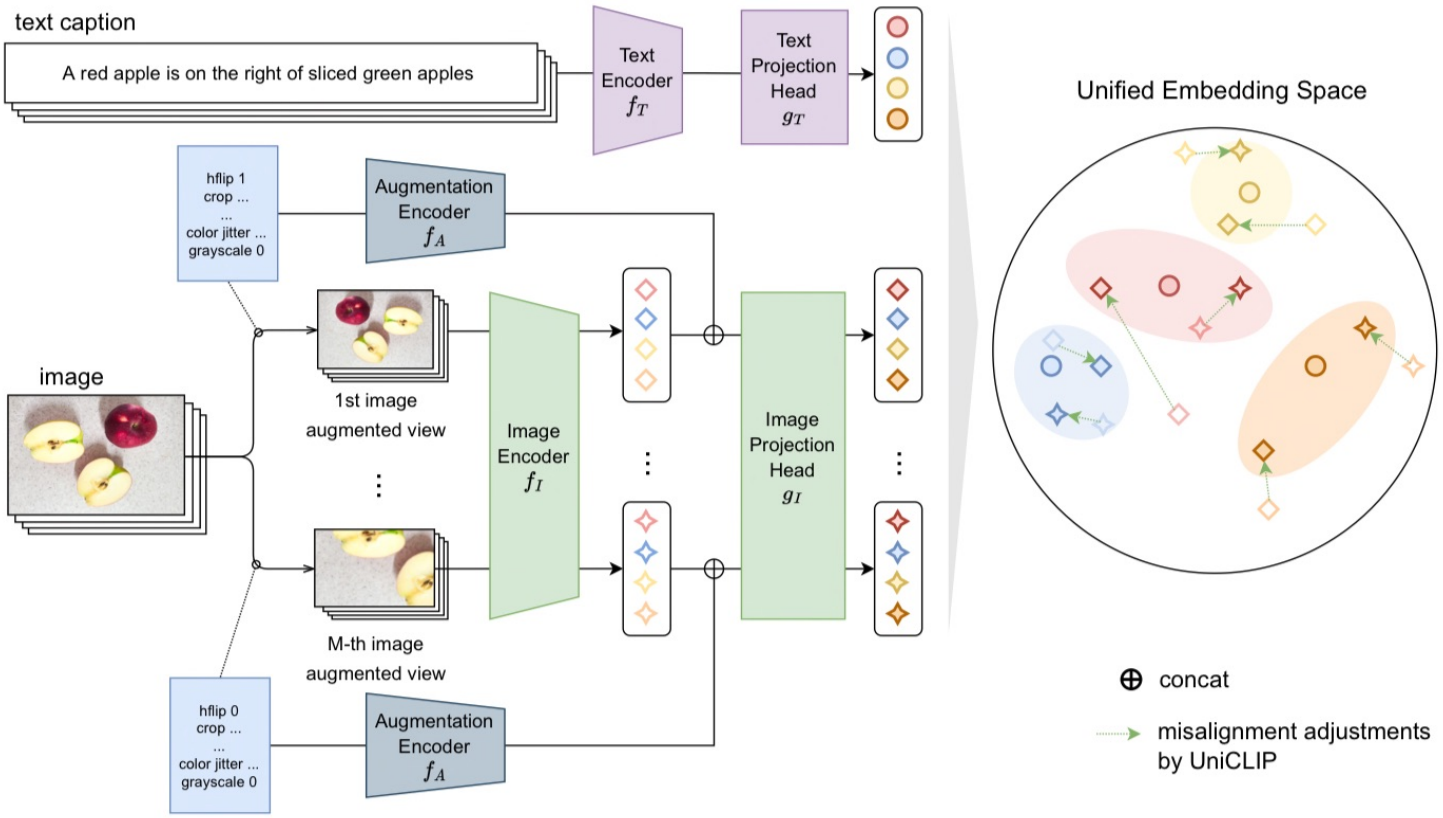
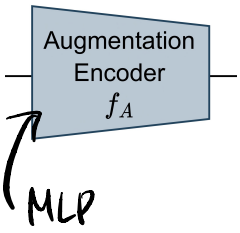


Figure 2: Overview of the UniCLIP framework.

[Methods]

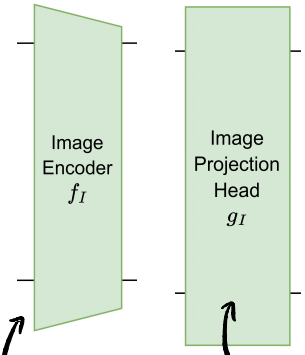
Augmentation Encoder



MLP

To use as an input to the network, augmentation instruction A is described as a real vector containing information about how much each basic transformation is applied to data

Image Encoder & Image Projection Head



augmentation_agnostic

augmentation-aware

The image encoder must be augmentation-agnostic and image projection head must be augmentation-aware.

→ The encoder can fully enjoy the benefits of data augmentation and generalize better, while the projection heads is still able to correct inter-domain misalignments caused by the augmentations.

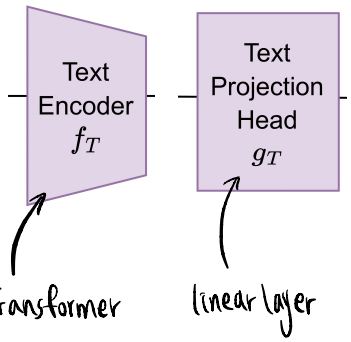
ViT

$h = f_I(\mathcal{A}(x))$

3 residual blocks

$z = g_I(f_I(\mathcal{A}(x)), f_A(\mathcal{A}))$

Text Encoder & Text Projection Head



Transformer

linear layer

tokenized text x: tokenize raw text by byte pair encoding

→ wrap with a start token and an end token

$h = f_T(x)$

$z = g_T(f_T(x))$

Contrastive loss functions for multiple positive pairs

$P_i = \{j | (z_i, z_j)$ is a positive pair and $j \neq i\}$
$N_i = \{j | (z_i, z_j)$ is a negative pair$\}$

MIL-NCE Loss MIL-NCE loss [23] for the i-th embedding is defined by

$$\mathcal{L}_i^{\text{MIL-NCE}} = -\log \frac{\sum_{p \in P_i} s_{i,p}}{\sum_{p \in P_i} s_{i,p} + \sum_{n \in N_i} s_{i,n}}.$$

maximize the sum of all positive pair similarity scores
and minimize the sum of all negative pair similarity scores

For some $q \in P_i$,

$$\frac{\partial \mathcal{L}_i^{\text{MIL-NCE}}}{\partial s_{i,q}} = -\frac{\sum_{n \in N_i} s_{i,n}}{\left(\sum_{p \in P_i} s_{i,p}\right)\left(\sum_{p \in P_i} s_{i,p} + \sum_{n \in N_i} s_{i,n}\right)}$$

the gradient will vanish to zero when $\sum_{p \in P_i} s_{i,p}$ is already large
even if the positive score $s_{i,q}$ is small (= hard positive pair)
→ hard positive pair cannot receive enough gradient
→ easy positive pairs hinder the training of hard positive pairs

SupCon Loss SupCon loss [17] for the i-th embedding is described by

$$\mathcal{L}_i^{\text{SupCon}} = \mathbb{E}_{p \in P_i}\left[-\log \frac{s_{i,p}}{\sum_{p' \in P_i} s_{i,p'} + \sum_{n \in N_i} s_{i,n}}\right]$$

each positive pair $s_{i,p}$ is compared with the negative pairs

$$\frac{\partial \mathcal{L}_i^{\text{SupCon}}}{\partial s_{i,q}} = \frac{s_{i,q} - \frac{1}{|P_i|}\left(\sum_{p \in P_i} s_{i,p} + \sum_{n \in N_i} s_{i,n}\right)}{s_{i,q}\left(\sum_{p \in P_i} s_{i,p} + \sum_{n \in N_i} s_{i,n}\right)}$$

hard positive pairs hinder the convergence of easy positive scores.

[Multi-positive NCE Loss]

$$\mathcal{L}_i = \mathbb{E}_{p \in P_i}\left[-\log \frac{s_{i,p}}{s_{i,p} + \sum_{n \in N_i} s_{i,n}}\right]$$

} Multi-positive version of InfoNCE loss

$$\frac{\partial \mathcal{L}_i}{\partial s_{i,q}} = -\frac{\sum_{n \in N_i} s_{i,n}}{|P_i| s_{i,q}\left(s_{i,q} + \sum_{n \in N_i} s_{i,n}\right)}$$

to make each positive pair independently contribute to the loss

⊕

the gradient is always negative!

$$\mathcal{L}_i = \mathbb{E}_{p \in P_i \cup \{i\}}\left[-\log \frac{s_{i,p}}{s_{i,p} + \sum_{n \in N_i} s_{i,n}}\right]$$

↓ weighted version

$$\mathcal{L}_i^{\text{MP-NCE}} = \mathbb{E}_{p \in P_i \cup \{i\}}\left[-w_{\mathcal{D}(i,p)} \log \frac{s_{i,p}}{s_{i,p} + \sum_{n \in N_i} s_{i,n}}\right]$$

$\mathcal{D}(i,p)$: the domain combination from which the i-th and p-th data were sampled

$w_{\mathcal{D}(i,p)}$: a domain-specific balancing hyperparameter which makes each inter-domain and intra-domain supervision equally contribute to the loss.

[Domain-Dependent Similarity Score]

$$s_{i,j} = \exp\left(\frac{1}{\tau}\left(\frac{z_i^\top z_j}{\|z_i\|\|z_j\|} - b\right)\right)$$

τ: trainable parameter, temperature to extend the range of cosine similarity

b: learnable threshold, amplifies the score if cosine similarity is greater than b, otherwise reduce it

s_{ij} { positive; if $sim > b$
negative; otherwise

$$s_{i,j} = \exp\left(\frac{1}{\tau_{\mathcal{D}(i,j)}}\left(\frac{z_i^\top z_j}{\|z_i\|\|z_j\|} - b_{\mathcal{D}(i,j)}\right)\right)$$ } domain-dependent similarity score

→ the offsets b are no longer cancelled out as negative pairs are sampled from multiple domains