It Takes Two to Tango: Mixup for Deep Metric Learning


Shashanka Venkataramanan


Bill Psomas


Ewa Kijak

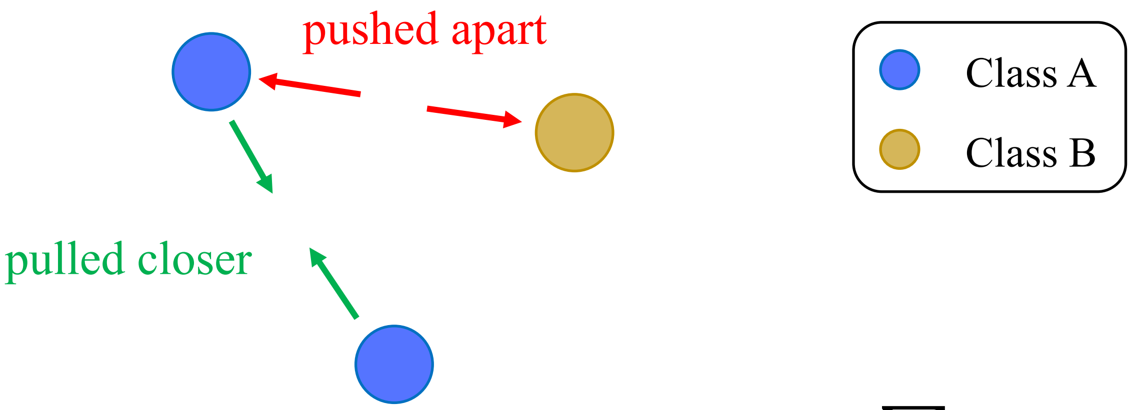

Laurent Amsaleg


Konstantinos Karantzalos


Yannis Avrithis

Deep Metric Learning

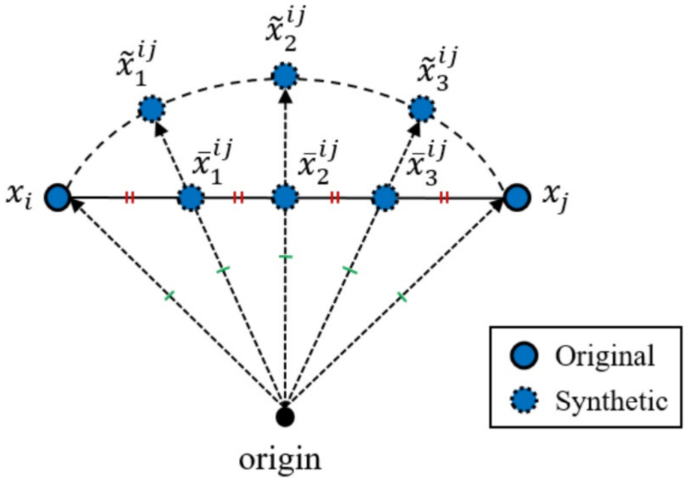
- GOAL – Learning a discriminative representation that generalizes to unseen classes.
- HOW? – Intra-class embeddings are pulled closer and inter-class embeddings are pushed apart.
- MOTIVATION – Classes during training and inference are different, interpolation-based data augmentation e.g. mixup plays significant role.



$$L_{cont} = \sum_{p \in P(a)} -s(a,p) + \sum_{n \in N(a)} [s(a,n) - m]_+$$

Interpolation for pairwise loss functions

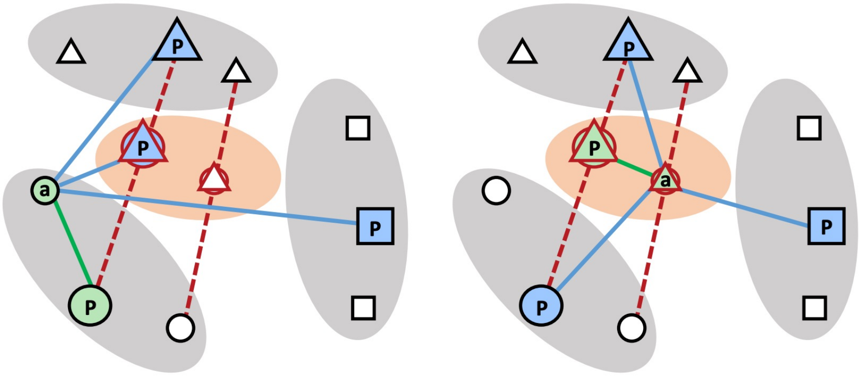
Embedding Expansion



Interpolate pairs of embeddings in a deterministic way within the same class.

Do not perform label interpolation.

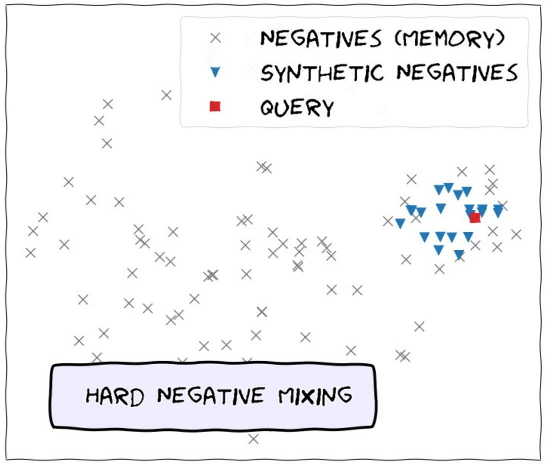
Proxy Synthesis



Interpolates between classes, applying to proxy-based losses only.

risks synthesizing false negatives when the interpolation factor λ is close to 0 or 1.

MoCHi



Interpolates anchor with negative embeddings.

do not interpolate labels, chooses $\lambda \in [0, 0.5]$ to avoid false negatives.

[Ko & Gu, CVPR'20, Gu *et al.,* AAAI'21, Kalantidis *et al.,* NeurIPS'20]

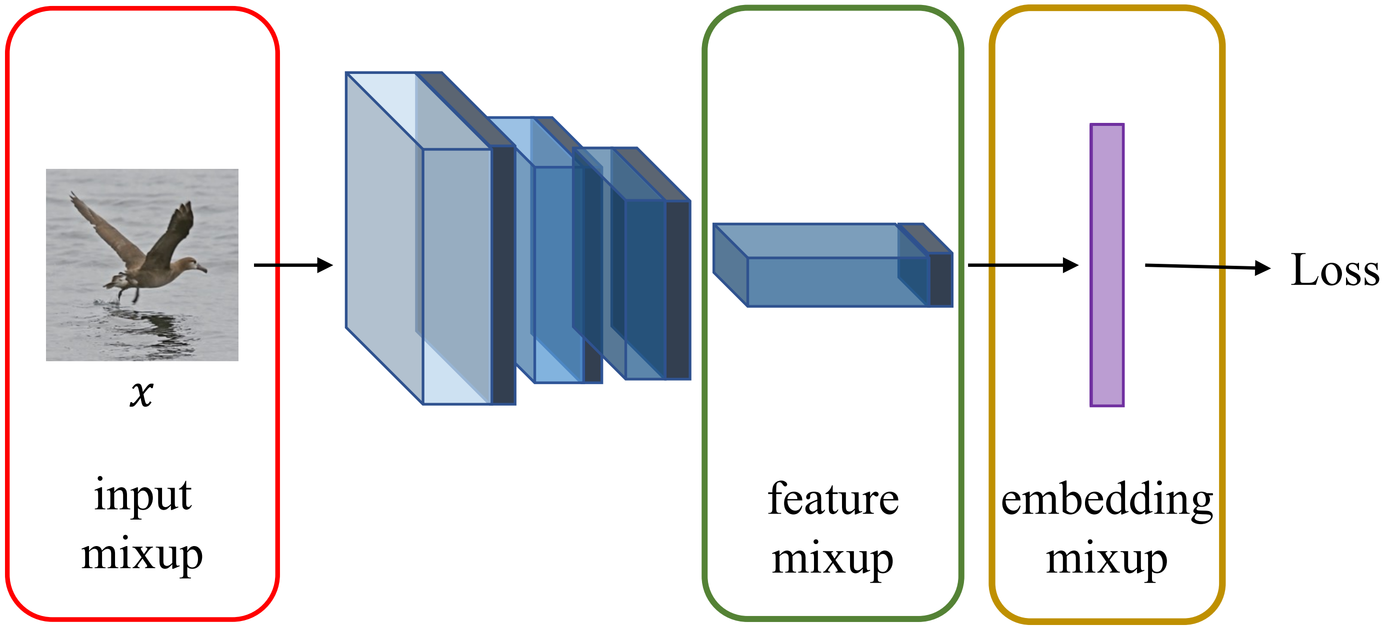
what is a proper way to define and interpolate labels
for deep metric learning ?

what is a proper way to define and interpolate labels
for deep metric learning ?

Metrix

Improving Representations Using Mixup

- Mixup: a data augmentation technique that interpolates between two examples (input or feature) and its corresponding labels.



$$\lambda \sim \mathrm{Beta}(\alpha, \alpha)$$
$$\mathrm{mix}_\lambda(a, b) = \lambda a + (1-\lambda) b'$$

[Zhang *et al.*, ICLR 2018, Verma *et al.*, ICML 2019]

Generic Loss Formulation

- Additive losses e.g., Contrastive and non-additive e.g., Multi-similarity involve a sum over positives $P(a)$ and a sum over negatives $N(a)$.

$$\ell(a;\theta) := \tau\left(\sigma^{+}\left(\underbrace{\boxed{\sum_{p\in P(a)} \rho^{+}(s(a,p))}}_{\text{sum over positives}}\right) + \sigma^{-}\left(\underbrace{\boxed{\sum_{n\in N(a)} \rho^{-}(s(a,n))}}_{\text{sum over negatives}}\right)\right)$$

Table 1 of our paper, shows the values of each of these terms for different loss functions.

Generic Loss Formulation

- Additive losses e.g., Contrastive and non-additive e.g., Multi-similarity involve a sum over positives $P(a)$ and a sum over negatives $N(a)$.

- They also involve a decreasing function of similarity $s(a,p)\ \forall\ p \in P(a)$ and an increasing function of similarity $s(a,n)\ \forall\ n \in N(a)$.

$$\ell(a;\theta) := \tau\left(\sigma^{+}\left(\sum_{p\in P(a)} \boxed{\rho^{+}}(s(a,p))\right) + \sigma^{-}\left(\sum_{n\in N(a)} \boxed{\rho^{-}}(s(a,n))\right)\right)$$

decreasing function of similarity

increasing function of similarity

Table 1 of our paper, shows the values of each of these terms for different loss functions.

Generic Loss Formulation

- Additive losses e.g., Contrastive and non-additive e.g., Multi-similarity involve a sum over positives $P(a)$ and a sum over negatives $N(a)$.

- They also involve a decreasing function of similarity $s(a,p)\ \forall\, p \in P(a)$ and an increasing function of similarity $s(a,n)\ \forall\, n \in N(a)$.

$$\ell(a;\theta) := \boxed{\tau}\left(\boxed{\sigma^+}\left(\sum_{p\in P(a)} \rho^+(s(a,p))\right) + \boxed{\sigma^-}\left(\sum_{n\in N(a)} \rho^-(s(a,n))\right)\right)$$

non-linear functions
(for non-additive losses)

Table 1 of our paper, shows the values of each of these terms for different loss functions.

Generic Loss Formulation

- In metric learning, positives $P(a)$ and negatives $N(a)$ of anchor a have the same or different class label as the anchor.

- We assign binary class label $y \in \{0,1\} \; \forall \; P(a) \cup N(a)$ s.t. $y = 1$ for positives and $y = 0$ for negatives.

$$\ell(a;\theta) := \tau\left(\sigma^+\left(\sum_{(x,y)\in U(a)} \boxed{y}\,\rho^+(s(a,x))\right) + \sigma^-\left(\sum_{(x,y)\in U(a)} \boxed{(1-y)}\,\rho^-(s(a,x))\right)\right)$$

y is binary, only one of the two contributions is nonzero.

Interpolating Labels Using Generic Formulation

- Given $M(a)$ which is the possible choices of mixing pairs (positive-positive or positive-negative or negative-negative), the labeled mixed embeddings is

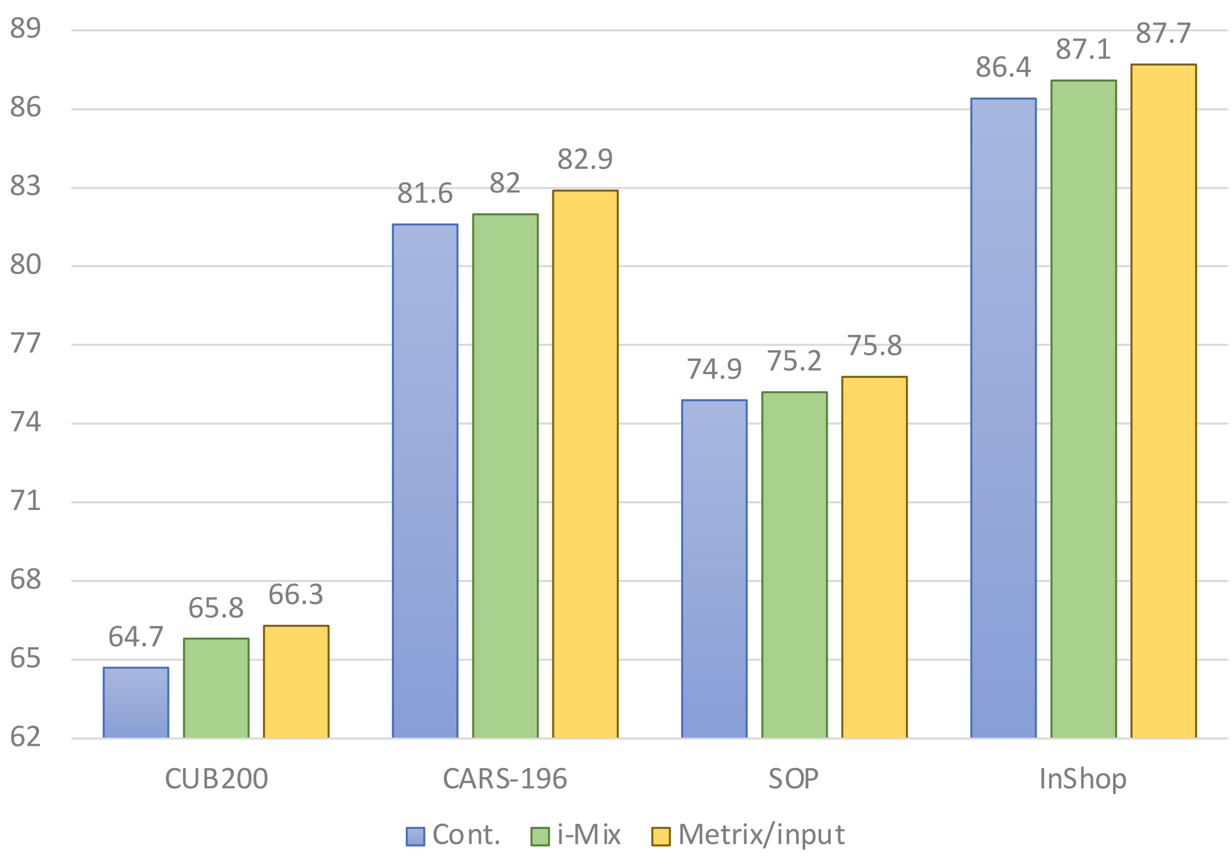
$$V(a) = \{f_\lambda(x, x'), \mathrm{mix}_\lambda(y, y') : ((x, y), (x', y') \in M(a)\}$$

$$\widetilde{\ell}(a;\theta) := \tau\left(\sigma^+\left(\sum_{(v,y)\in V(a)} \boxed{y}\,\rho^+(s(a,v))\right) + \sigma^-\left(\sum_{(v,y)\in V(a)} \boxed{(1-y)}\,\rho^-(s(a,v))\right)\right)$$

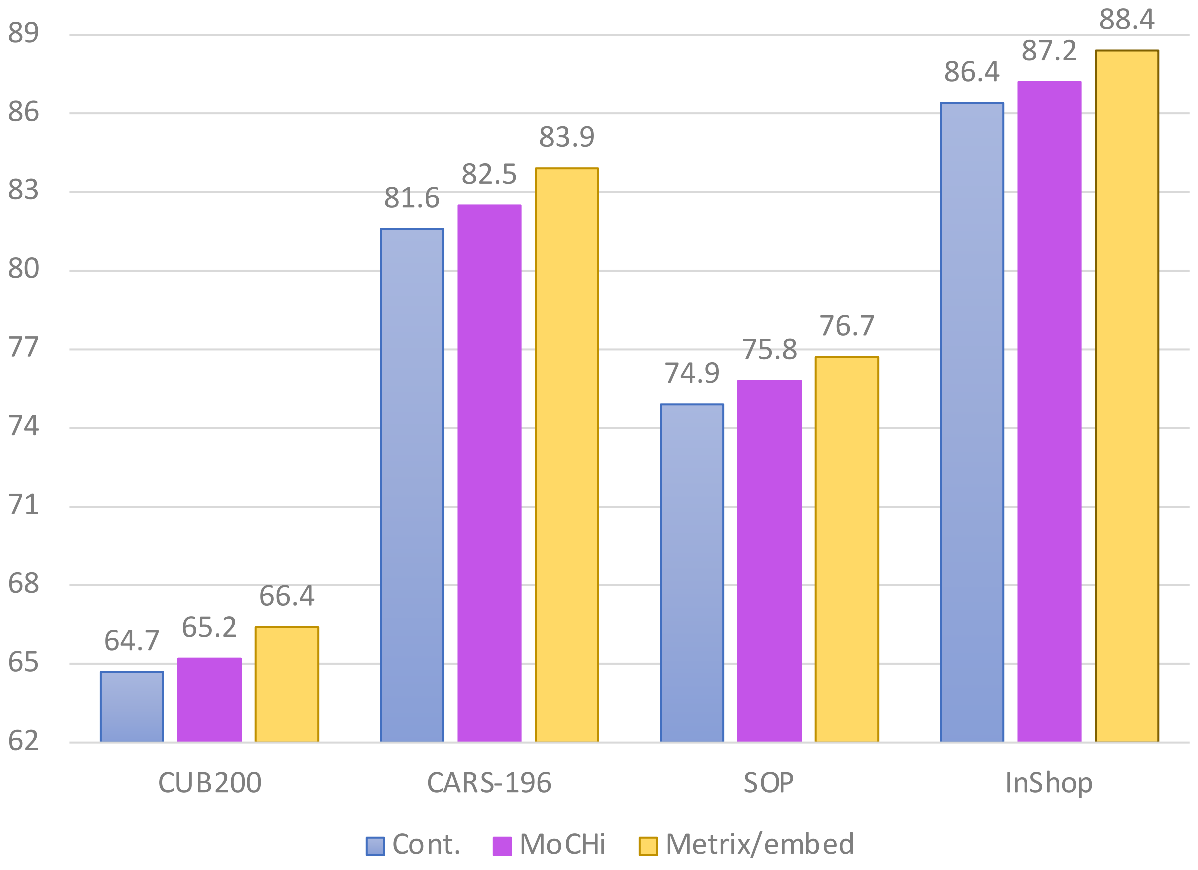
$y \in [0,1]$, both contributions are nonzero.

Comparison With Other Mixing Methods

Mixup in input space



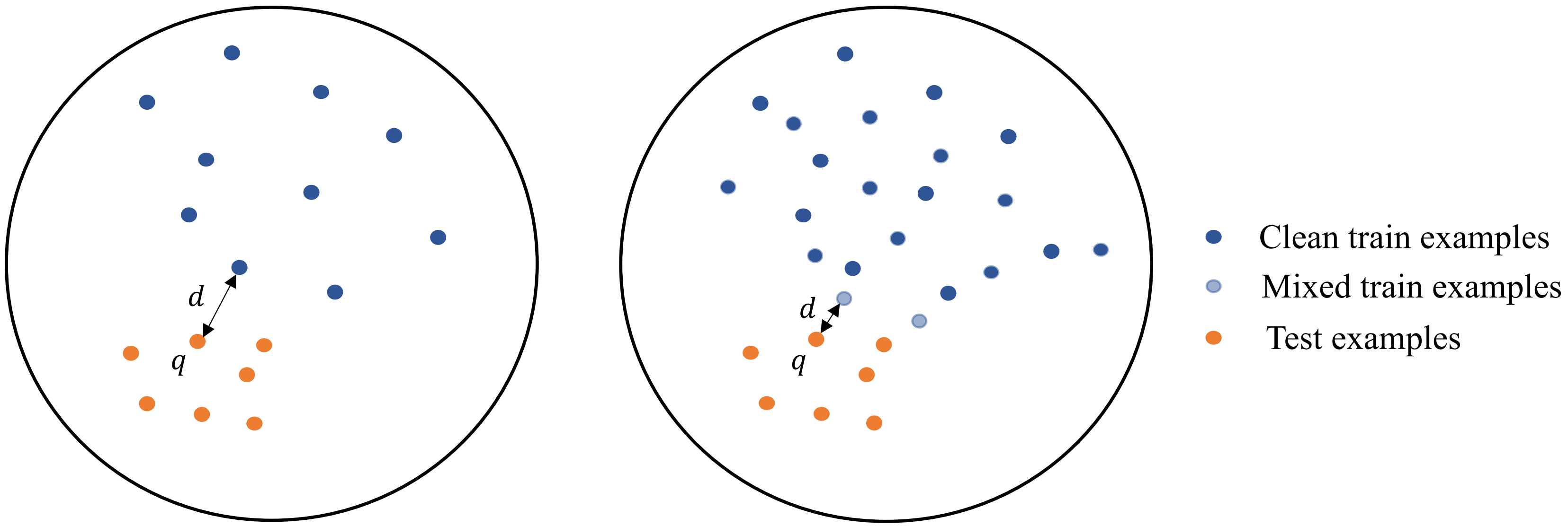
Mixup in embedding space



[Hadsell *et al.,* CVPR' 06; Wah *et al.,*2011; Krause *et al.,* ICCVW'13; Oh Song *et al.,* CVPR'16; Liu *et al.,* CVPR'16; Kalantidis *et al.*, NeurIPS'20; Lee *et al.,* ICLR'21]

How Does Mixup Improve Representations?

- Introduce a new evaluation metric - utilization and show that a representation more appropriate for test classes is implicitly learned during exploration of the embedding space in the presence of mixup.



without mixup

with mixup

Paper

Code