Analysis and applications of i-mix on domain agnostic environment

-Summary-

Introduction

- From DABS 2.0 [A Tamkin, 2022], it turned out that mix-up is an effective data augmentation technique regardless of data domain.

Pretrain Dataset	Transfer Dataset	Encoder	Baseline Performance	*e*-mix Performance
ImageNet	CIFAR10	Transformer	24.20%	39.43%
ImageNet	CU Birds	Transformer	1.62%	**3.86%**
ImageNet	VGG Flowers	Transformer	9.03%	**25.96%**
ImageNet	DTD	Transformer	7.39%	8.83%
ImageNet	Traffic Sign	Transformer	14.33%	**65.07%**
ImageNet	Aircraft	Transformer	2.70%	**10.15%**
PAMAP2	PAMAP2	Transformer	69.81%	79.48%
MSCOCO	VQA	Transformer	53.38%	**58.77%**
MSCOCO	Mismatched Caption	Transformer	49.41%	49.86%
CheXpert	CheXpert	Transformer	68.14%	**72.40%**

Introduction

- The suggested technique is called ‘e-mix’, which is variant of i-mix [K Lee, 2020], and have the following loss:

Definition: i-mix InfoNCE loss

Assume f is the encoder for normalized feature embedding with ReLU activation (so that $\nabla_x^2 f(x) = 0$) and $x^{(1)}$, $x^{(2)}, x^{(-)}$are given as two positive samples, and one negative sample. Then, the InfoNCE loss and i-mix InfoNCE loss are given as follows:

$$L_{un} = \mathbb{E}_{\substack{x^{(1)},x^{(2)}\sim D_{sim} \\ x^{(-)}\sim D_{neg} \\ \lambda\sim Beta(\alpha,\alpha)}}\left[-\log\left(\frac{e^{f\left(x^{(1)}\right)^T f\left(x^{(2)}\right)}}{e^{f\left(x^{(1)}\right)^T f\left(x^{(2)}\right)} + e^{f\left(x^{(1)}\right)^T f\left(x^{(-)}\right)}}\right)\right]$$

$$L_{un}^{i-mix} = \mathbb{E}_{\substack{x^{(1)},x^{(2)}\sim D_{sim} \\ x^{(-)}\sim D_{neg} \\ \lambda\sim Beta(\alpha,\alpha)}}\left[-\lambda\log\left(\frac{e^{f\left(x^{mix}\right)^T f\left(x^{(1)}\right)}}{e^{f\left(x^{mix}\right)^T f\left(x^{(1)}\right)} + e^{f\left(x^{mix}\right)^T f\left(x^{(-)}\right)}}\right) - (1-\lambda)\log\left(\frac{e^{f\left(x^{mix}\right)^T f\left(x^{(2)}\right)}}{e^{f\left(x^{mix}\right)^T f\left(x^{(2)}\right)} + e^{f\left(x^{mix}\right)^T f\left(x^{(-)}\right)}}\right)\right]$$

where $x^{(mix)} = \lambda x^{(1)} + (1-\lambda)x^{(2)}$, and D_{sim} : distribution for positive pairs, D_{neg} : distribution for negative sample.

Introduction

Domain	Dataset	N-pair	+ i-Mix	MoCo v2	+ i-Mix	BYOL	+ i-Mix
Image	CIFAR-10	93.3 ± 0.1	**95.6** ± 0.2	93.5 ± 0.2	**96.1** ± 0.1	94.2 ± 0.2	**96.3** ± 0.2
	CIFAR-100	70.8 ± 0.4	**75.8** ± 0.3	71.6 ± 0.1	**78.1** ± 0.3	72.7 ± 0.4	**78.6** ± 0.2
Speech	Commands	94.9 ± 0.1	**98.3** ± 0.1	96.3 ± 0.1	**98.4** ± 0.0	94.8 ± 0.2	**98.3** ± 0.0
Tabular	CovType	68.5 ± 0.3	**72.1** ± 0.2	70.5 ± 0.2	**73.1** ± 0.1	72.1 ± 0.2	**74.1** ± 0.2

Table 1: Comparison of contrastive representation learning methods and i-Mix in different domains.

- e-mix perform i-mix without any data-augmentation on the input data
  - Q: Why does it improves downstream performance significantly?
  - Q: Why can it be effective at even domain-circumstance?
    (Note: e-mix does not use positive pairs at all (no data augmentation at all).

- Hypothesis: There must be a strong regularization effect which acts as ‘indirect’ contrastive learning (which does not require positive samples)

Taylor expansion of i-mix loss

- Conventional analysis of mix-up loss is to apply 2nd order Taylor expansion [Zhang, 2021], [L Carratino, 2022], [C Park, 2022] (under the supervised mix-up setting)

Theorem : Approximated i-mix InfoNCE loss ($\lambda \cong 0$) [Informal]

The i-mix InfoNCE L_{un} can be approximated by 2nd order Taylor expansion near $\lambda = 0$ as follows:

$$\widetilde{L_{un}^{i-mix}} = \frac{1}{2} L_{un} + R^{(1)} + R^{(2)}$$

where

$$R^{(1)} = -\frac{1}{2} \mathbb{E}_{\substack{x^{(1)}, x^{(2)} \sim D_{sim} \\ x^{(-)} \sim D_{neg} \\ \lambda \sim Beta(\alpha, \alpha)}} \left[\log \frac{e}{e + e^{f(x^{mix})^T f(x^{(-)})}} \right]$$

Repel the negative sample $x^{(-)}$ from the mixup pair $x^{(mix)}$

And, $R^{(2)}$ is given as follows (next page) :

Taylor expansion of i-mix loss

Theorem : Approximated i-mix InfoNCE loss ($\lambda \cong 0$) [Informal] (Cont.)

The i-mix InfoNCE L_{un} can be approximated by 2nd order Taylor expansion near $\lambda = 0$ as follows:

$R^{(2)}$

Indirect Contrastive learning via tangential approximation (unclear)

Note: $\boldsymbol{R^{(2)}}$ terms effects gets significant as $\boldsymbol{\lambda \to \frac{1}{2}}$

$$= \mathbb{E}_{\substack{x^{(1)},x^{(2)} \sim D_{sim} \\ x^{(-)} \sim D_{neg} \\ \lambda \sim Beta(\alpha,\alpha)}} \Big[c_1 \cdot \frac{e^{f(x^{(2)})^T f(x^{(-)})}}{e^{f(x^{(2)})^T f(x^{(1)})} + e^{f(x^{(2)})^T f(x^{(-)})}} \cdot \left(\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} \nabla \boldsymbol{f}\left(\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} \left[\boldsymbol{f}\left(\boldsymbol{x}^{(1)}\right) - \boldsymbol{f}\left(\boldsymbol{x}^{(-)}\right)\right] + c_2 \cdot \frac{e^{f(x^{(2)})^T f(x^{(-)})}}{e + e^{f(x^{(2)})^T f(x^{(-)})}}$$

$$\cdot \left(\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} \nabla \boldsymbol{f}\left(\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} \left[\boldsymbol{f}\left(\boldsymbol{x}^{(2)}\right) - \boldsymbol{f}\left(\boldsymbol{x}^{(-)}\right)\right] + c_3 \cdot \frac{e^{f(x^{(2)})^T f(x^{(-)})}}{\left(e + e^{f(x^{(2)})^T f(x^{(-)})}\right)^2} \cdot \left[\left(\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} \nabla \boldsymbol{f}\left(\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} \left[\boldsymbol{f}\left(\boldsymbol{x}^{(2)}\right) - \boldsymbol{f}\left(\boldsymbol{x}^{(-)}\right)\right]\right]^2$$

Taylor expansion of i-mix loss

- Inside $R^{(2)}$, there are terms related with indirect contrastive loss
  - Attraction between 'tangential approximation of $x^{(1)}$ w.r.t $x^{(2)}$' and '$f\left(x^{(1)}\right)$', and repulsion between 'tangential approximation of $x^{(1)}$ w.r.t $x^{(2)}$' and '$f\left(x^{(-)}\right)$':

$$\left(\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} \nabla \boldsymbol{f}\left(\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}}\left[\boldsymbol{f}\left(\boldsymbol{x}^{(1)}\right)-\boldsymbol{f}\left(\boldsymbol{x}^{(-)}\right)\right]$$

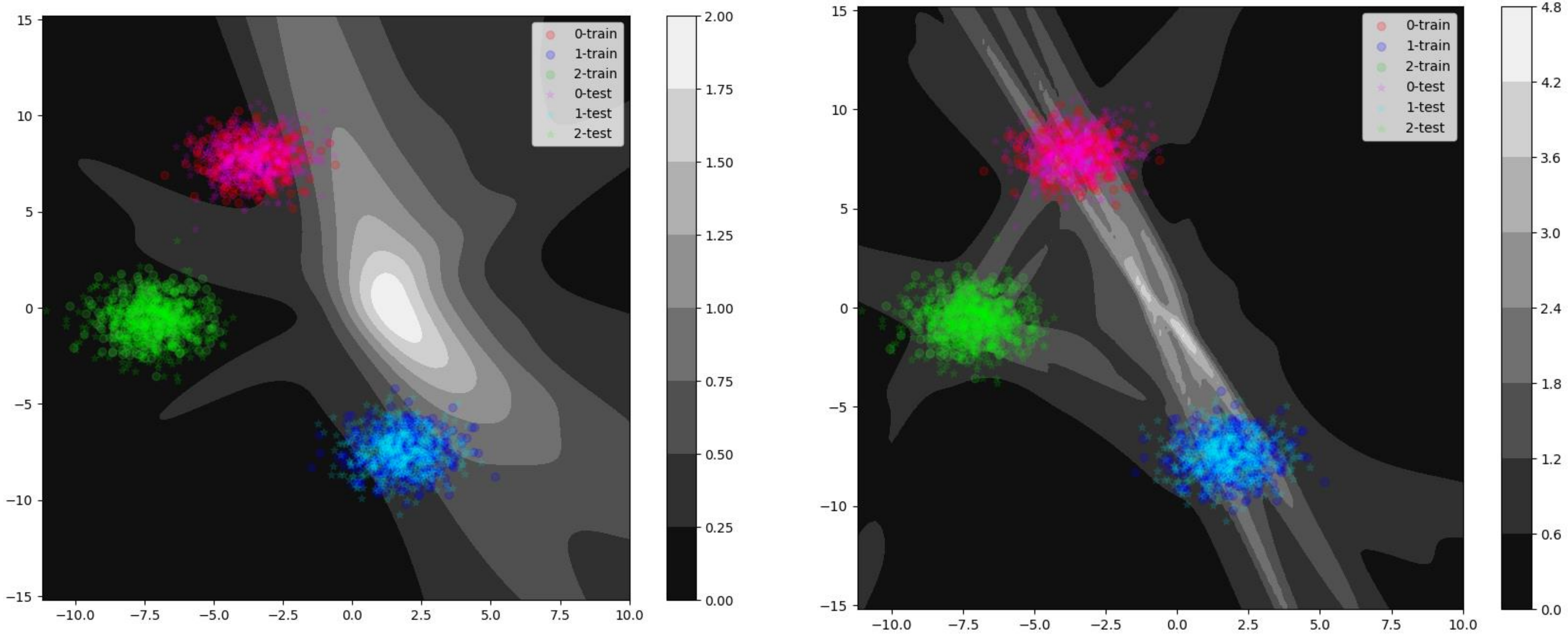
  - Attraction between 'tangential approximation of $x^{(1)}$ w.r.t $x^{(2)}$' and '$f\left(x^{(2)}\right)$', and repulsion between 'tangential approximation of $x^{(1)}$ w.r.t $x^{(2)}$' and '$f\left(x^{(-)}\right)$':

$$\left(\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} \nabla \boldsymbol{f}\left(\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}}\left[\boldsymbol{f}\left(\boldsymbol{x}^{(2)}\right)-\boldsymbol{f}\left(\boldsymbol{x}^{(-)}\right)\right]$$

- Unsolved Q: Why does it helpful? (why the input gradient control can be effective?)
  - Unclear, but it is speculated that they control the input gradients to make a regularized contrasted feature space. (which might be helpful for downstream performance)

Taylor expansion of i-mix loss

- Empirically, it turned out that the $\|\nabla f(x)\|_2$ is get more stronger at the convex hull of data set:

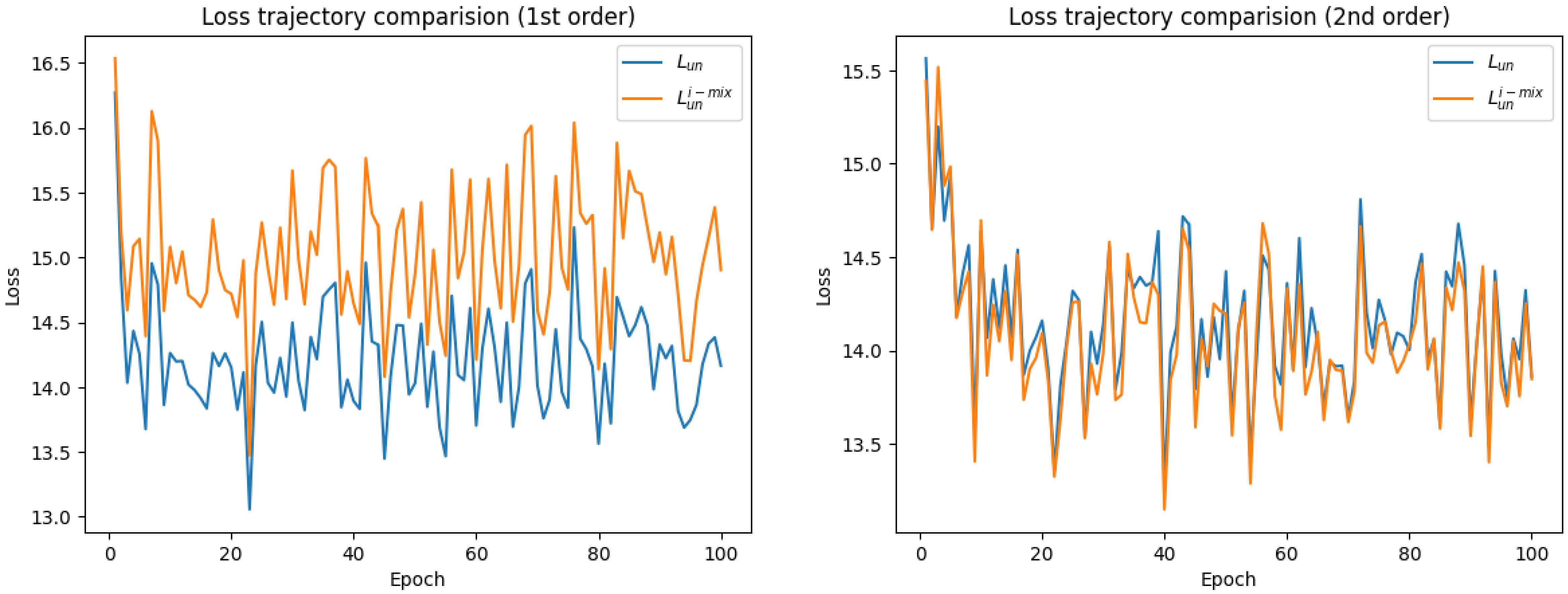


Input gradient norm trained under L_{un}

Input gradient norm trained under L_{un}^{i-mix}

Taylor expansion of i-mix loss

- Then, does the approximated mix-up loss is accurate compared to true mix-up loss?
  - Under the 3-layer ReLU Network + 2-dimensional blob dataset, the loss trajectory is given as follows (as did in [C. Park, 2022], [Zhang, 2021]): $(\lambda \sim Beta(1,1))$



Downstream performance by i-mix

- How about the downstream performance guarantee under i-mix loss?
- Using the framework from [Arora, 2019], [Verma, 2021], the following claim holds:

Lemma 1: Minimizing the empirical loss → minimizing the upper bound of true loss (similarly as in [Arora, 2019])

With probability at least $1-\delta$ over the training set $\mathcal{S}$, for all $f \in \mathcal{F} \coloneqq \left\{f: \mathcal{X} \to \mathbb{R}^d \;\middle|\; \|f\|_B < R \text{ for some } R > 0\right\}$

$$L_{un}^{i-mix}(\hat{f}) \leq \widehat{L_{un}}^{i-mix}(f) + O(R \cdot \frac{\mathcal{R}_S(\mathcal{F})}{M} + R^2\sqrt{\frac{\log\frac{1}{\delta}}{M}})$$

where $\mathcal{R}_S(\mathcal{F}) \coloneqq \mathbb{E}_{\sigma\sim\{\pm1\}^{3dM}}\left[\sup_{f\in\mathcal{F}} < \sigma, f_{|S} >\right]$ (= Rademacher complexity], and $M = |\mathcal{S}|$ (= sample size) , $\hat{f} \in argmin_{f\in\mathcal{F}}\, \widehat{L_{un}}(f)$

Downstream performance by i-mix

Theorem: Weak guarantee of downstream performance under i-mix training [similarly as in [Verma, 2021])

Let $\bar{\rho}(y) = P(y' \neq y \mid y) > 0$, $\rho(y) = P(y' = y|y)$, then, under the binary class contrastive learning and linear evaluation setting, the following holds:

$$l_{un}^{i-mix} \cong \frac{1}{2}\mathbb{E}_{\substack{(x^{(1)},y)\sim D \\ x^{(2)}\sim D_{1-y}}}\left[\rho(y) l_{sup}\left(f\left(x^{(1)},y\right)^T \widetilde{w}\right)\right] + \frac{1}{2}\mathbb{E}_y\left[\left(1-\bar{\rho}(y)\right)\mathcal{E}_y\right] + R^{(1)} + R^{(2)}$$

where

$$\mathcal{E}_y = \mathbb{E}_{\substack{x^{(1)},x^{(2)}\sim D_y \\ \lambda\sim Beta(\alpha,\alpha)}}\left[log\left(1 + exp\left(-\frac{f\left(x^{(1)}\right)^T}{\|f(x^{(1)}\|}\left(\frac{f\left(x^{(2)}\right)^T}{\|f(x^{(2)}\|} - \frac{f\left(x^{(-)}\right)^T}{\|f(x^{(-)}\|}\right)\right)\right)\right]$$

$$\widetilde{w} = \frac{1}{\|h(x^{(1)}\|}\left(\frac{1}{\left\|h\left(\pi_{y,1}\left(x^{(2)},x^{(-)}\right)\right)\right\|}h\left(\pi_{y,1}\left(x^{(2)},x^{(-)}\right)\right) - \frac{1}{\left\|h\left(\pi_{y,0}\left(x^{(2)},x^{(-)}\right)\right)\right\|}h\left(\pi_{y,0}\left(x^{(2)},x^{(-)}\right)\right)\right)$$

$$\pi_{y,y'}\left(x^{(2)},x^{(-)}\right) = P(y = y')\cdot x^{(2)} + P(y \neq y')\cdot x^{(-)}$$

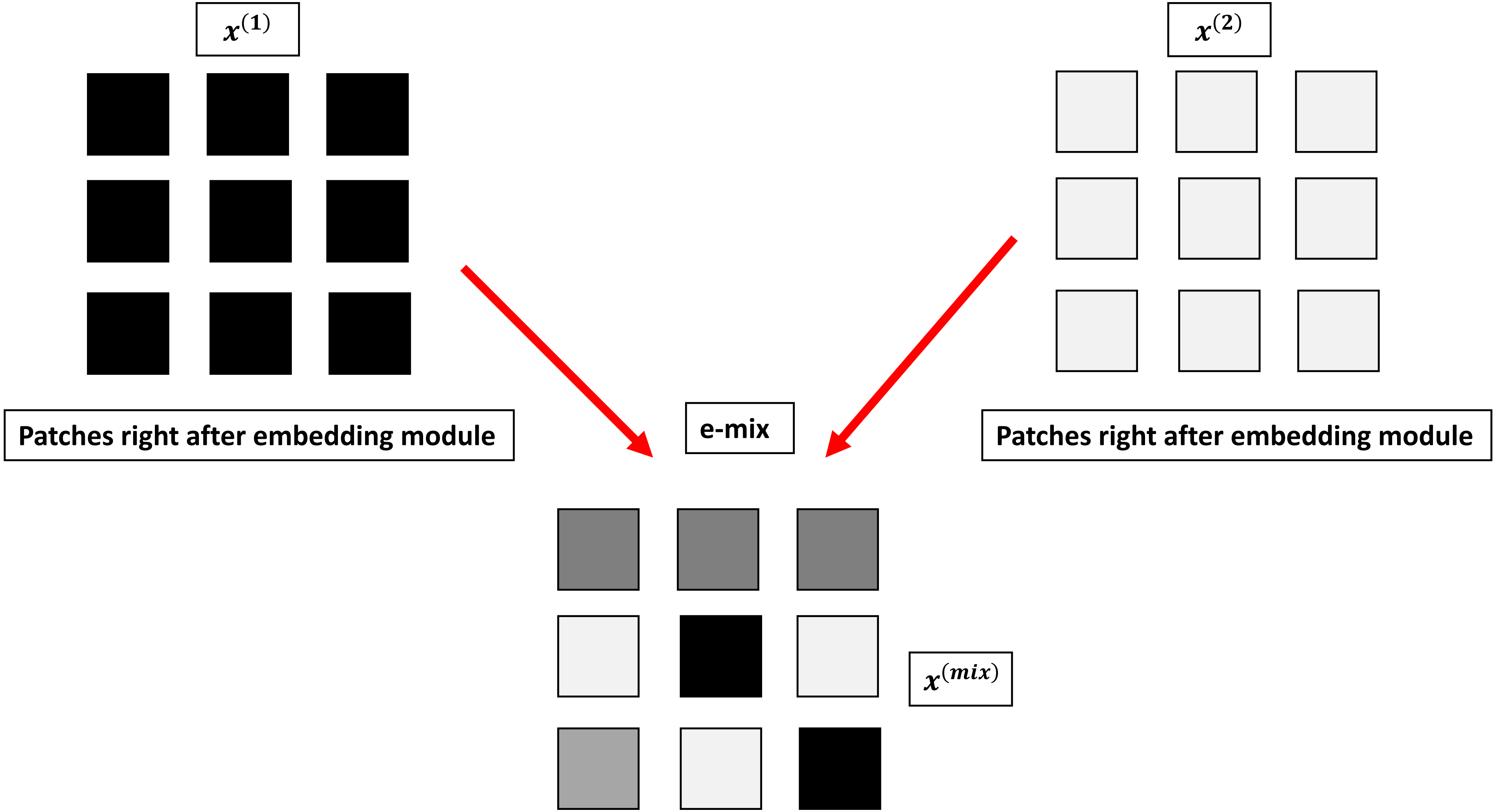
Application of i-mix on domain agnostic environment

- In DABS 2.0, they used ViT structure to cover domain agnostic data.
  - But, it turned out that the suggested domain-agnostic algorithms (ex: e-mix, ShED, ContPred) tend to show improvement in certain types of domain:
    - e-mix : strong at continuous data (spectrogram, vision, continuous tabular [sensor])
    - SheD (Shuffled embedding detection) / ContPred (mask certain portion of random patches) : strong at discontinuous data (text, vision, discontinuous tabular)

- But, How about exploiting the ViT structure for mix-up?
  - Use patch mix-up in discrete unit => may be helpful for both continuous / discontinuous data.

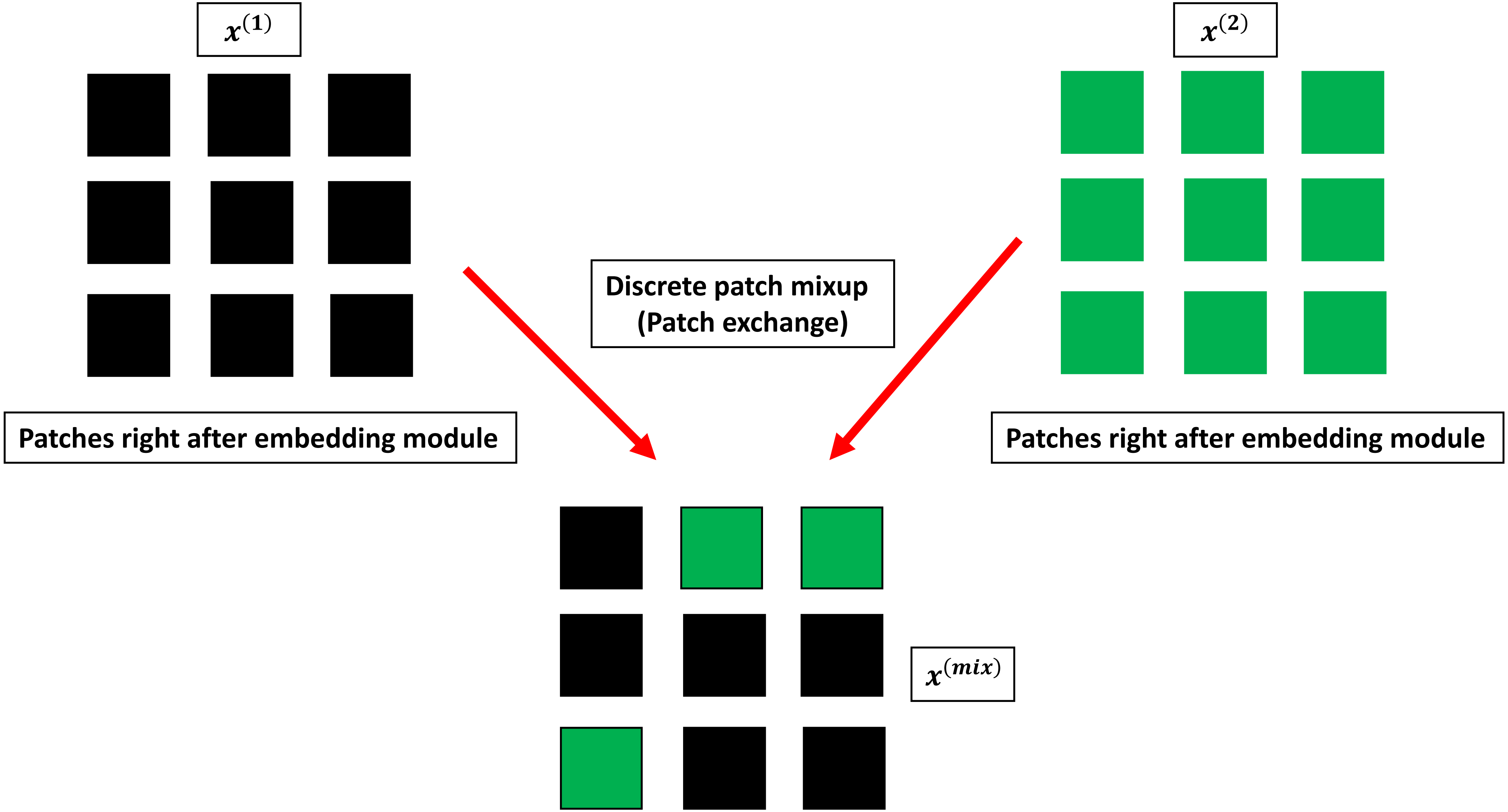
Application of i-mix on domain agnostic environment

- e-mix (from DABS 2.0)



Application of i-mix on domain agnostic environment

- Discrete patch mixup:



Application of i-mix on domain agnostic environment

- Performance comparison ('dmix' = discrete patch mixup)
  - Online evaluation test accuracy (not linear evaluation, but can be a substitute)

Dataset	CIFAR-10	CIFAR-100	PAMAP2	Euroset	CUBirds	DTD	Traffic Sign	MNLI (matched)
emix	38.4	10.1	81.8	**89.2**	**1.54**	7.71	55.4	32.6
ShED	36.4	No data	65.9	55.7	No data	6.12	28.4	**38.1**
dmix	**48.7**	**19.7**	**82.8**	87.4	1.45	**8.24**	**63.0**	32.6

- Clearly shows good performance compared to emix, but unclear on the NLP domain
  - We can further generalize dmix by combining ContPred algorithm (called 'Contdmix')
    - Then, on continuous data : shED < emix < contdmix < dmix
    - On discontinuous data : emix < dmix <= contdmix < shED