ANALYSIS OF I-MIX AND ITS APPLICATION ON DOMAIN-AGNOSTIC CONTRASTIVE LEARNING

Student: Kijung Jeon[1] **Teaching Assistant : Kyungmin Lee**[2] **Advisor: Jinwoo Shin**[1,2]
KAIST School of Electrical Engineering[1] / Graduate school of AI[2]

ABSTRACT

Recently, many studies have emerged in the field of contrastive learning that mixup regularization techniques can be used to significantly improve downstream performance. In addition, in the context of contrastive learning without using any data augmentation (domain-agnostic environment), several results have emerged that mixup has achieved better representation across various data domains. In this paper, we identify what regularization effect the mixup provides in the In- foNCE loss and present a simple yet effective variant of mixup techniques to im- prove downstream performance in a wide range of data domains by exploiting the ViT structure and Cutmix technique, which were previously proposed in a benchmark (DABS 2.0). We suggest that mixup on InfoNCE loss provides 'indirect input gradient contrastive learning' by using tangential approximation at each sample point and forming a controlled input gradient zone within the convex hull shape of data. Following from it, we provide a simple algorithm (Dmix) which discretely mixup the patches after embedding on ViT structure, and demonstrate Dmix outperforms on various domains of downstream tasks compared to the currently suggested domain agnostic algorithms in DABS 2.0.

1 INTRODUCTION

In supervised learning, the Mixup technique was introduced for input data regularization techniques to improve generalization performance and achieve more robustness (Zhang et al., 2018) while it was first suggested in contrastive learning by Lee et al. (2021), which is called i-mix. Recently, there are several works to demonstrate theoretically how does the mixup work in not only the supervised learning setting (Zhang et al., 2021), (Carratino et al., 2022), (Park et al., 2022), (Chidambaram et al., 2022) , but also in the unsupervised learning setting (Verma et al., 2021).
It is well known that a strategy to use semantically related data augmentation (such as Color jittering / Masking in the Image / Text domain) gives a better representation on each domain (Chen et al., 2020), furthermore, there is research work to learn how to perform data augmentation in a continuous domain using GAN structure (Tamkin et al., 2021). However, it is still unclear to incorporate various domain data in a single large model. To deal with this problem, a benchmark (DABS 2.0)(Tamkin et al., 2022) appeared to provide a standard measure for downstream performances on various domain data. In the work (DABS 2.0), ViT structure (Dosovitskiy et al., 2021) was used to incorporate domain agnostic environment, and provided several domain agnostic algorithms coupled with InfoNCE loss (van den Oord et al., 2019). Similarly, Verma et al. (2021) suggested a method to use directly contrasting two mixup samples, and showed it gives a better representation on various domains.
In this paper, we focus on a mixup regularization effect suggested on i-mix, and verify that mixup in InfoNCE can provide input gradient control to make a regularized learned feature space. Especially, this mixup technique not only contrasts positive and negative samples directly as in InfoNCE but also contrasts these samples via tangential approximation of them, which potentially helps to regularize the feature space embeddings of each sample. Also, we empirically verified that this method can impose a controlled input gradient zone within a convex hull of data. To link this finding to domain agnostic mixup technique, we devised a simple variant of mixup (Dmix) to cope well with domain agnostic environment, and demonstrated that they outperform the algorithms suggested on DABS 2.0 in several domains.

2 ANALYSIS OF REGULARIZATION EFFECT ON I-MIX

In this section, we provide how the i-mix provides a regularization effect during contrastive learning by suggesting 2nd order Taylor expansion of i-mix loss based on Zhang et al. (2021), Park et al. (2022) and link this loss to unsupervised loss following the scheme appeared on Verma et al. (2021). During the analysis, we follow the framework and most of the notations from Arora et al. (2019)

2.1 NOTATIONS AND PRELIMINARIES

Suppose we have an labeled data $\{x_i, y_i\}_{i=1}^N \sim \mathcal{D}^N$ where $x_i \in \mathcal{X} \subset \mathbb{R}^n$, $y_i \in \{0, 1\}$. The task is to perform contrastive learning using i-mix coupled with InfoNCE loss. We assume there are two positive samples as $\{x^{(1)}, x^{(2)}\} \sim \mathcal{D}_{sim}$ and corresponding k negative samples $\{x_1^-, x_2^-, \ldots, x_k^-\} \sim \mathcal{D}_{neg}^k$, and mixup sample for a positive pair $\{x^{(1)}, x^{(2)}\}$ is generated by $x^{mix} = \lambda \cdot x^{(1)} + (1 - \lambda) \cdot x^{(2)}$. Now, we define an encoder network $f \in \mathcal{F}$ where $\mathcal{F} := \{f : \mathcal{X} \to \mathbb{R}^d : \|f(x)\| \leq R \text{ for some } R > 0 \text{ and any } x \in \mathcal{X}\}$. Note that we normalize the embedding $f(x)$ in practice, which gives $R = 1$. After the contrastive learning, encoder network f is fixed, and the linear evaluator $W : \mathbb{R}^d \to \mathbb{R}$ is attached on top of the fixed encoder to perform supervised learning for achieving fined-tuned model. For the notation, we define $g(x) = \phi(W^T \cdot f(x))$ to denote the predicted probability value on the linear evaluation stage. where $\phi(\cdot)$ is a logistic function.

2.1.1 CONTRASTIVE LOSSES AND LINEAR EVALUATION

Here, we assume the number of negative samples $k = 1$ for simplicity. Then, the standard InfoNCE loss (or unsupervised loss, L^{un}) is given as follows:

$$L^{un}(f, x^{(1)}, x^{(2)}, x^{(-)}) = -\log \frac{e^{f(x^{(1)})^T f(x^{(2)})}}{e^{f(x^{(1)})^T f(x^{(2)})} + e^{f(x^{(1)})^T f(x^{(-)})}} \tag{1}$$

Next, we define i-mix loss coupled with InfoNCE loss as follows:

$$\begin{aligned} L^{mix}(f, x^{(1)}, x^{(2)}, x^{(-)}, \lambda) = &-\lambda \log \left(\frac{e^{f(x^{mix})^T f(x^{(1)})}}{e^{f(x^{mix})^T f(x^{(1)})} + e^{f(x^{mix})^T f(x^{(-)})}} \right) \\ &-(1 - \lambda) \log \left(\frac{e^{f(x^{mix})^T f(x^{(2)})}}{e^{f(x^{mix})^T f(x^{(2)})} + e^{f(x^{mix})^T f(x^{(-)})}} \right) \end{aligned} \tag{2}$$

where $\lambda \sim Beta(\alpha, \alpha)$, and $\alpha > 0$. The true losses $L_{true}^{un}(f), L_{true}^{mix}(f)$ can be obtained by taking expectation over $(x^{(1)}, x^{(2)}) \sim \mathcal{D}_{sim}$, $x^- \sim \mathcal{D}_{neg}$, $\lambda \sim Beta(\alpha, \alpha)$ (for the L^{mix} case) After the unsupervised training using above losses, the encoder network f is fixed and learned using supervised loss (L^{sup}) to measure downstream performance or learned representation quality. In this paper, we focus on binary classification with label $y_i \in \{0, 1\}$ in the downstream task, where L^{sup} is defined as $L^{sup}(f, W, x_i, y_i) = -y_i \cdot \log(g(x_i)) - (1 - y_i) \cdot \log(1 - g(x_i))$ (logistic loss)

2.2 INPUT GRADIENT CONTROLLING EFFECT OF I-MIX

In this subsection, we first show i-mix loss coupled with InfoNCE loss provides the input gradient control effect during training by approximating the L^{mix} using 2nd order taylor expansion. Next, we provide empirical verification of this approximation by comparing a true loss and approximated loss as similarly performed in Zhang et al. (2021), Park et al. (2022), Carratino et al. (2022). At the end, we verify that the input gradient controlling (which will be denoted as indirect gradient contrastive learning) happens in practice under the two dimensional blob data with a three layer neural network.

2.2.1 TAYLOR EXPANSION OF I-MIX LOSS COUPLED WITH INFONCE LOSS

By applying 2nd order taylor expansion on the equation (2) around $\lambda = 0$, we get the following result under the encoder network with ReLU activation functions:

Theorem 2.1. *Assuming a encoder network f with ReLU activation where the input gradient $\nabla_x f(x) = 0$, the 2nd order Taylor expansion of $L^{mix}(x^{(1)}, x^{(2)}, x^{(-)}, \lambda)$ is expressed as follows:*

$$L^{mix}_{approx}(f, x^{(1)}, x^{(2)}, x^{(-)}, \lambda) = -(1-\lambda) \cdot \log\left(\frac{d_2}{d_2 + d_-}\right) - \lambda \cdot \log\left(\frac{d_1}{d_1 + d_-}\right) - \lambda^2 \cdot \frac{d_-(l_1 - l_-)}{d_1 + d_-} + (\lambda^2 - \lambda) \cdot \frac{d_-(l_2 - l_-)}{d_2 + d_-} - \frac{\lambda^2}{2} \cdot \frac{d_2 d_-(l_2 - l_-)^2}{(d_2 + d_-)^2} \tag{3}$$

where d_k, l_k are given as follows for $k \in \{1, 2, -\}$:

$$d_k = e^{f(x^{(2)})^T f(x^{(k)})} \qquad l_k = \left(x^{(1)} - x^{(2)}\right)^T \nabla f(x^{(2)})^T f(x^{(k)})$$

Between first two terms, the former one is dominant around $\lambda = 0$, and the term gives a effect of repelling the distance between $x^{(2)}$ and $x^{(-)}$. After those terms, 2nd order approximated terms appear with $(l_1 - l_-)$, $(l_2 - l_-)$ terms on numerators which perform input gradient controlling, and they are represented as follows:

$$l_1 - l_- = \left(x^{(1)} - x^{(2)}\right)^T \nabla f(x^{(2)})^T (f(x^{(1)}) - f(x^{(-)}) \tag{4}$$

$$l_2 - l_- = \left(x^{(1)} - x^{(2)}\right)^T \nabla f(x^{(2)})^T (f(x^{(2)}) - f(x^{(-)}) \tag{5}$$

Our interpretation on these term is illustrated in figure 1. the $l_1 - l_-$ term makes the distance closer between the feature space location of $x^{(1)}$ and its tangential approximation of $x^{(1)}$ at the point $x^{(2)}$ (= $\nabla f(x^{(2)})(x^{(1)} - x^{(2)})$), and $l_2 - l_-$ term gives the same attraction effect between the feature space location of $x^{(2)}$ and and the tangential approximation of $x^{(1)}$ at the point $x^{(2)}$ while both terms repel the tangential approximation of $x^{(1)}$ from the feature space location of $x^{(-)}$. This effect implies that the achieved feature space space will be regularized after training with i-mix loss by controlling the input gradient of positive and negative samples, which eventually results in a better learned feature space as verified empirically in Lee et al. (2021).

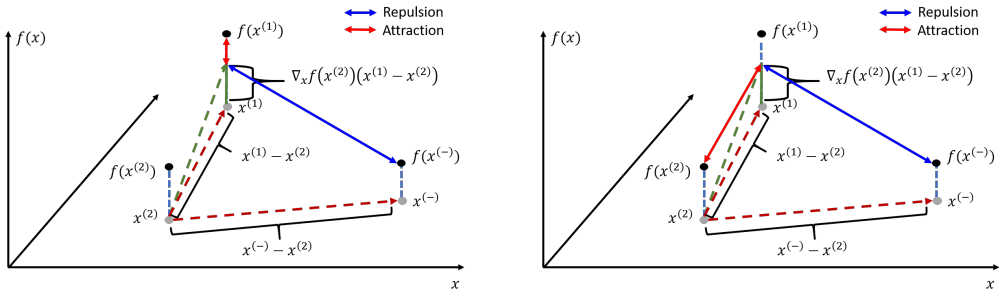
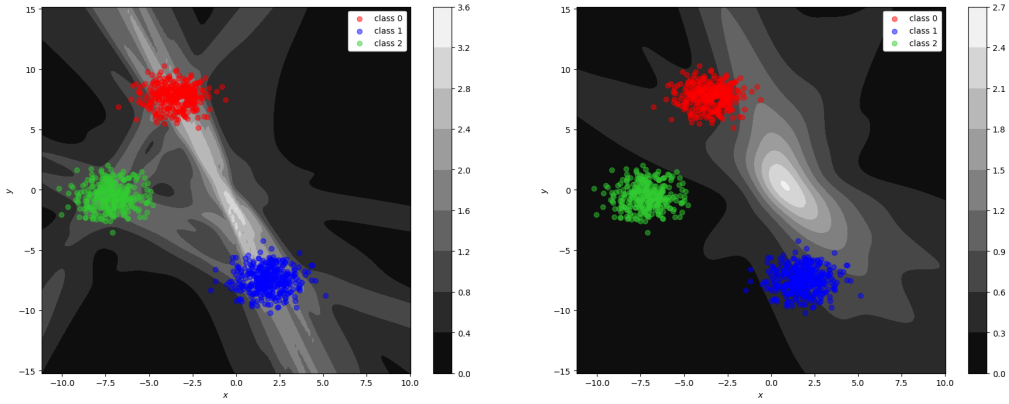


Figure 1: Illustration of $l_1 - l_-$ (left) and $l_2 - l_-$ (right) term effect

2.2.2 Empirical verification of input gradient controlling

Under the two dimensional blob data with a three layer encoder network, we verify empirically that the input gradient controlling effect activates in the convex hull of data samples (Figure 2). While the input gradients of samples are not affected when standard InfoNCE loss is used, i-mix loss effectively controls the input gradient within the convex hull shape of the data sample. To observe the three dimensional feature space shape after training, the whole input space was forwarded through trained encoder f and processed by one dimensional PCA for visualization of the trained feature space $f(x)$ (figure 3). The achieved feature space by i-mix is more tough compared to the one that achieved by InfoNCE. We presume that this mainly occurs due to the input gradient control illustrated on figure 1. However, it is still unclear how does the input gradient control and the resultant rough feature space gives better feature space at the end of the domain agnostic contrastive learning with i-mix loss, which we leave as one of the future studies.



(a) $\|\nabla_x f(x)\|_F$ of trained encoder (i-mix) (b) $\|\nabla_x f(x)\|_F$ of trained encoder (InfoNCE)

Figure 2: Illustration of input gradient distribution on the input space

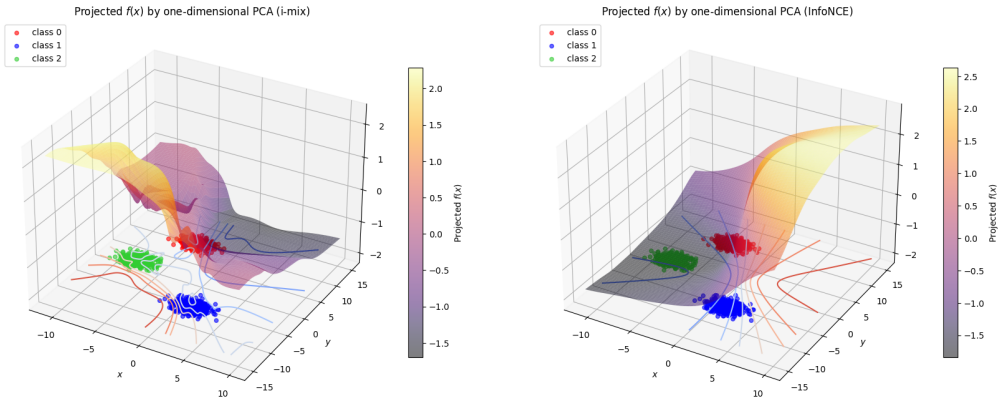


Figure 3: Approximation of feature space $f(x)$ by PCA (left: i-mix, right: InfoNCE)

2.2.3 Empirical verification of Taylor approximation for i-mix

As suggested on many researches (Zhang et al., 2021), (Park et al., 2022), (Carratino et al., 2022) trying to explain theoretically how does the mixup works, we claim our 2nd order taylor approximation of i-mix (Equation 3) is valid in practice. We perform a comparison experiment between true i-mix loss (Equation 2) and approximated i-mix loss (Equation 3) (Figure 4). As a result, it turns out that the 2nd order taylor approximation is sufficiently accurate to follow true loss while the 1st order taylor approximation is inaccurate. Indirectly, this phenomenon implies the input gradient controlling terms (Equation 4, 5) are not negligible terms in practice. For the experiment environment, we use the two dimensional blob data with three layer encoder network and sample $\lambda \sim Beta(1, 1)$ with temperature parameter (Wang & Liu, 2021) in loss function set to be 0.2.

2.2.4 Weak guarantee of downstream performance by i-mix loss

In this subsection, we show that minimizing the empirical i-mix loss $\hat{L}^{mix}(f)$ is approximately equivalent to minmizing the supervised loss $L^{sup}(f, W, x_i, y_i)$, where $\hat{L}^{mix}(f) = \mathbb{E}_{\lambda \sim Beta(\alpha,\alpha)}\left[\frac{1}{N}\sum_{i=1}^{N} L^{mix}(x_i^{(1)}, x_i^{(2)}, x_-^{(-)}, \lambda)\right]$. Here, we use the main framework from Arora et al. (2019) and Verma et al. (2021) to prove several facts. In the proof, we first show that minimizing the $\hat{L}^{mix}(f)$ is equivalent to minimize the $L^{mix}_{true}(\hat{f})$, and use one of the theorem in Verma et al. (2021) and Theorem 2.1 to directly link the $L^{mix}_{true}(\hat{f})$ and $L^{sup}_{true}(\hat{f}, W)$, where $\hat{f} = argmin_{f \in \mathcal{F}}(\hat{L}^{mix}(f))$, and W is a constant weight matrix.

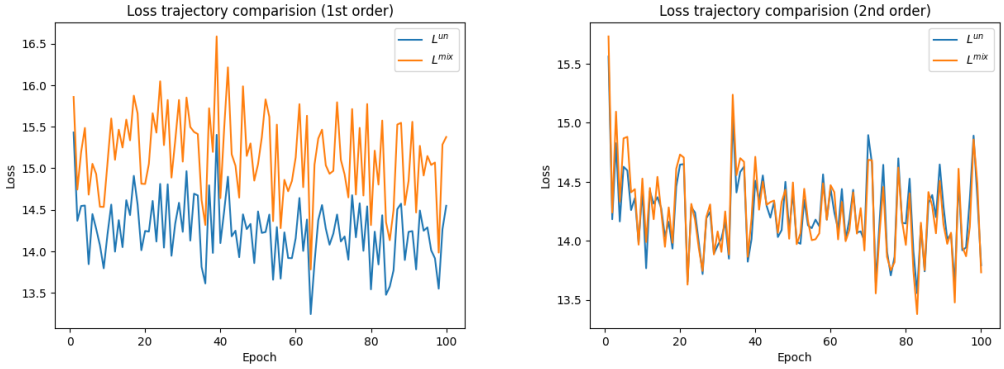


Figure 4: Loss trajectory comparison (left: 1st order, right: 2nd order)

Lemma 2.2. *Define* $\mathcal{S} := \{x_i^{(1)}, x_i^{(2)}, x_i^{(-)}\}_{i=1}^M$. *Then, with probability at least* $1-\delta$ *over the training set* $\mathcal{S}$, *for all* $f \in \mathcal{F} := \{f : \mathcal{X} \to \mathbb{R}^d : \|f(x)\| \leq R$ *for some* $R > 0$ *and any* $x \in \mathcal{X}\}$ *The following inequality holds:*

$$L_{true}^{mix}(\hat{f}) \leq \hat{L}^{mix}(f) + O(R \cdot \frac{\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + R^2\sqrt{\frac{log\frac{1}{\delta}}{M}}) \tag{6}$$

where $\mathcal{R}_{\mathcal{S}}(\mathcal{F}) := \mathcal{E}_{\sigma\sim\{\pm1\}^{3dM}}[\sup_{f\in\mathcal{F}} < \sigma, f_{|\mathcal{S}} >]$, $f_{|\mathcal{S}} := \{f(x_i^{(1)}), f(x_i^{(2)}), f(x_i^{(-)})\})_{i=1}^M$

Above lemma theoretically shows that minimizing the empirical loss $\hat{L}^{mix}(f)$ is equivalent to minimizing the upper bound of true loss with empirical minimizer $L_{true}^{mix}(\hat{f})$.

Corollary 2.3. *The true i-mix loss* L_{true}^{mix} *can be approximated using 2.1 as follows:*

$$L_{true}^{mix}(\hat{f}) \simeq \frac{1}{2}L_{true}^{un}(\hat{f}) + R^{(1)} + R^{(2)} \tag{7}$$

where

$$R^{(1)} = -\frac{1}{2}\mathbb{E}_{\substack{(x^{(1)},x^{(2)})\sim\mathcal{D}_{sim} \\ x^{(-)}\sim\mathcal{D}_{neg} \\ \lambda\sim Beta(\alpha,\alpha)}}\left[log\frac{d_2}{d_2+d_-}\right],$$

$$R^{(2)} = -\mathbb{E}_{\substack{(x^{(1)},x^{(2)})\sim\mathcal{D}_{sim} \\ x^{(-)}\sim\mathcal{D}_{neg} \\ \lambda\sim Beta(\alpha,\alpha)}}\left[\lambda^2 \cdot log\frac{d_2}{d_2+d_-} + (\lambda-\lambda^2)\cdot\frac{d_-(l_2-l_-)}{d_1+d_2} + \frac{\lambda^2}{2}\cdot\frac{d_2 d_-(l_2-l_-)^2}{(d_2+d_-)^2}\right]$$

And, the d_k, l_k *for* $\{k \in 1, 2, -\}$ *are the same as in Theorem 2.1 by changing* f *into* $\hat{f}$.

This corollary can be obtained by taking expectation on the Equation 3 and using the fact $\mathbb{E}_{\lambda\sim Beta(\alpha,\alpha)}[\lambda] = \frac{1}{2}$. For the side note, the $R^{(2)}$ term is related with the 'indirect input gradient contrastive learning' which we discussed above section and its effect gets enhanced as $\lambda \to \frac{1}{2}$.

Theorem 2.4 (i-mix variant of Theorem 1 in Verma et al. (2021))**.** *Let* $\bar{\rho}(y) = P(y' \neq y|y)$ *and* $\rho(y) = P(y' = y|y)$. *Also, define* $\mathcal{D}_x$ *be the marginal distribution of* x *and let* $\mathcal{D}_y$ *be the conditional distribution of* x *given* y. *Then, under the binary class linear evaluation following from i-mix contrastive learning with the data distribution* $\mathcal{D}$, *the following relation holds:*

$$L_{true}^{mix}(\hat{f}) \simeq \frac{1}{2}\mathbb{E}_{\substack{(x^{(1)},y)\sim\mathcal{D} \\ x^{(2)}\sim\mathcal{D}_{(1-y)}}}\left[\rho(y)L^{sup}(\hat{f}, \tilde{W}, x^{(1)}, y)\right] + \frac{1}{2}\mathcal{E}_y\left[(1-\bar{\rho}(y))\mathcal{E}_y\right] + R^{(1)} + R^{(2)} \tag{8}$$

where

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

$\tilde{W} = \frac{1}{\|\tilde{f}(x^{(1)})\|}\left(\frac{1}{\|\tilde{f}(\pi_{y,1}(x^{(2)},x^{(-)}))\|}\tilde{f}(\pi_{y,1}(x^{(2)},x^{(-)})) - \frac{1}{\|\tilde{f}(\pi_{y,0}(x^{(2)},x^{(-)}))\|}\tilde{f}(\pi_{y,0}(x^{(2)},x^{(-)}))\right)$,

$\pi_{y,y'}(x^{(2)},x^{(-)}) = \mathbb{1}_{\{y=y'\}}x^{(2)} + \mathbb{1}_{\{y\neq y'\}}x^{(-)}$*, and* $R^{(1)}, R^{(2)}$ *are the same as in Corollary 2.3.*

The above theorem can be obtained by combining theorem 2.1 and theorem 1 in Verma et al. (2021) under no perturbation circumstance.

By combining Lemma 2.2 and Corollary 2.3, Theorem 2.4 sequentially, we can claim that minimizing the empirical i-mix loss $\hat{L}^{mix}(f)$ approximately minimize the supervised loss $L^{sup}(\hat{f}, \tilde{W}, x^{(1)}, y)$, which provides the weak guarantee of downstream performance from training with i-mix loss.

3 Application of i-mix under domain-agnostic environment

One of the currently suggested domain-agnostic algorithms for contrastive learning is to use two mixup samples stem from one anchor sample (Verma et al., 2021), and use i-mix loss without data augmentation (Tamkin et al., 2022), which is called 'e-mix'. One of the problem of both algorithms is that these algorithms do not show good the downstream performance for discrete data (for example, text, or tabular data with categorical values, ...) while they are sufficiently strong for continuous data (for example, spectrogram, or image, graph, ...) In this paper, we focus on devising a variant of 'e-mix' which can show good downstream performance not only for continuous data but also for discrete data. The key idea for our algorithms is to combine masking (which is used on LLM model for text data (Devlin et al., 2019)) and Cutmix strategy (Yun et al., 2019) by exploiting the ViT model given in the DABS 2.0 benchmark ((Tamkin et al., 2022))
For notations, let us define $e(\cdot) : \mathbb{R}^n \to \mathbb{R}^{P\times m}$ to be embedding module of ViT, and the embedding of x_i is given by $e(x_i) = z_i = [z_{1,i}, ..., z_{P,i}]$, where P is the number of patches in embedding. Assume $g : \mathbb{R}^{P\times m} \to \mathbb{R}^d$ be the Transformer encoder (suggested on Dosovitskiy et al. (2021)), and, for convenience, define $\hat{L}^{mix}_{ViT}(z, z^{perm}, z^{mix}, \lambda) = \frac{1}{B}\sum_{i=1}^{B}\left(\lambda\cdot L^{un}(f, z_i, z_i^{mix}) + (1-\lambda)\cdot L^{un}(f, z_i^{perm}, z_i^{mix})\right)$ where B is a batch size and z, z^{perm}, z^{mix} are batch-wise embeddings, permuted embeddings, and mixup embedding, respectively. Now, we suggest 'Dmix' which binarily mix up the patches using the format of e-mix, and provide a brief illustration of e-mix (algorithm 1) and Dmix (algorithm 2) on figure 5.

Algorithm 1 e-mix (Tamkin et al., 2022)

Input: batch sample: $x = \{x_i\}_{i=1}^B$, encoder: g, embedding module: e
Output: contrastive loss : $L(e, g, x)$

$z \leftarrow e(x)$ ▷ Embed input batch x
Sample $\lambda \sim Beta(\alpha, \alpha)$
$[\sigma] \leftarrow$ permute ($[B]$) ▷ Permute 1st index
$z^{perm} \leftarrow z\,[\sigma]$ ▷ Batchwise permutaion
$z^{mix} \leftarrow \lambda\cdot z + (1-\lambda)\cdot z^{perm}$
$L \leftarrow \hat{L}^{mix}_{ViT}(z, z^{perm}, z^{mix}, \lambda)$
Return $L(e, g, x) = L$

Algorithm 2 Dmix (ours)

Input: batch sample: $x = \{x_i\}_{i=1}^B$, encoder: g, embedding module: e
Output: contrastive loss : $L(e, g, x)$

$z \in \mathbb{R}^{B\times P\times m} \leftarrow e(x)$
Sample $\lambda \sim Beta(\alpha, \alpha)$
$k \leftarrow \lceil \lambda\cdot B \rceil$
$M \sim \{0,1\}^P$ with sum$(M) = k$
$M \leftarrow M$.expand(B, P, m)
$[\sigma] \leftarrow$ permute ($[B]$) ▷ Permute 1st index
$z^{perm} \leftarrow z\,[\sigma]$
$z^{mix} \leftarrow M \odot z + (1-M)\odot z^{perm}$
$L \leftarrow \hat{L}^{mix}_{ViT}(z, z^{perm}, z^{mix}, \lambda)$
Return $L(e, g, x) = L$

3.1 Experiment setup

In this paper, we are exactly following the suggested environment in DABS 2.0 (Tamkin et al., 2022) for measuring the downstream performance of our algorithms with fairness.

3.1.1 BASELINE ALGORITHM

Among the algorithms suggested in DABS 2.0, we compare three baseline algorithms (e-mix, ShED) to our algorithms (Dmix).

- **e-mix** : a domain agnostic generalization of i-mix (Lee et al., 2021) without using data augmentation, and described in algorithm 1.
- **ShED** : shuffled embedding detection algorithm, which randomly permutes a subset of the input embeddings and trains the model to identify the permuted embeddings. (generalization of ELECTRA, (Clark et al., 2020))

3.1.2 DATASETS

For the comparison, we use the suggested dataset given in Tamkin et al. (2022). Among the various domain, we select RGB Image, Multi-spectral(MS) Image, Time-series, Token, Text data, which are briefly discussed below:

- **RGB Image** domain contains 3 channels (R, G, B) with two dimensional images. For comparison, CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009) and CUB (Wah et al., 2022), DTD (Cimpoi et al., 2013), Traffic Sign (Houben et al., 2013) are chosen. Also, we downscale all of the used dataset by 32×32 for a memory issue.
- **MS (Multi-spectral) Image** domain contains multi-channel two dimensional images. We adopts EuroSAT (Helber et al., 2019) dataset whose images contain 13 channels images captured by satelites.
- **Time-series** domain incorporates continuous data which change as time passes. In this experiment, we use PAMAP2 (Reiss & Stricker, 2012) dataset where the several activity data are recorded by sensors which are attached on subjects playing certain actions.
- **Token** domain consists of semantically discrete sequences such as nucleic sequence. For this domain, we choose to use Genomics (in-distribution) (Ren et al., 2019) and SCOP (Fox et al., 2013).
- **Text** domain indicates any text data written in human languages. In this experiment, we test an inference task (MNLI matched) from GLUE benchmark (Wang et al., 2019) and PAWS benchmark (Zhang et al., 2019), (Yang et al., 2019) for English and Korean.

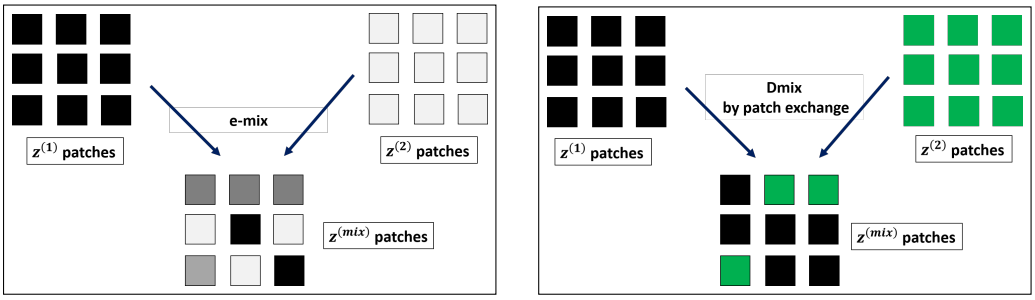


Figure 5: Illustration of mixup strategies (left: e-mix, right: Dmix)

3.2 EXPERIMENT RESULT

We provide the in-domain online evaluation results over various domains on table 1. One of the drawbacks of e-mix and ShEd is that they are complementary algorithm in terms of data type. For example, e-mix shows better results for continuous data (such as Image, Time-series), while it does not for discrete data (such as Text, Token), and this phenomenon appears in opposite for ShED. Our desired goal is to devise Dmix so that it unifies the advantages of both algorithms by adopting Cutmix algorithm (Yun et al., 2019) in discrete manners. We speculate that the performance drop of e-mix mainly happens due to its continuous mixing strategy, which does not give meaningful data augmentation for text data or discrete tabular data. To overcome this issue, we mix up patches after

Table 1: In-domain online evaluation over various domains. Values are reported in accuracy (%). Online evaluation (Tamkin et al., 2022) represents linear evaluation with frozen encoder for each epoch, which can be used as a heuristic for measuring transfer performance.

Domain	**RGB Image**					**MS Image**	**Time-series**
Dataset	CIFAR-10	CIFAR-100	CUB	DTD	Traffic Signs	EuroSAT	PAMAP2
e-mix	38.4	10.1	**1.54**	7.71	55.4	**89.2**	81.8
ShED	36.4	10.8	1.40	6.12	28.4	55.7	65.9
Dmix	**48.7**	**25.1**	1.45	**8.24**	**69.3**	87.4	**82.8**

Domain	**Token**		**Text**		
Dataset	Genomics (in-distribution)	SCOP	MNLI (matched)	PAWS (Ko)	PAWS (En)
e-mix	33.6	**9.10**	32.6	57.6	57.5
ShED	19.3	6.52	38.1	**59.0**	51.0
Dmix	**39.8**	8.42	**38.6**	58.6	**58.0**

embedding by discretely mixing up. By doing so, the model can perform both mix up and masking strategy simultaneously. For the RGB image domain, the Dmix shows significant downstream performance over 5 datasets. Especially, the in-domain online evaluation test accuracy improves by $38.4\% \rightarrow 48.7\%$ on CIFAR-10 and $10.1\% \rightarrow 25.1\%$ on CIFAR-100, $55.4\% \rightarrow 69.3\%$ on Traffic Sign dataset. While we target the improvement on discrete domain, significant performance gap is attained on RGB Image (continuous data). Conversely, there is no improvement on some dataset (EuroSAT, SOCP). This implies the Dmix is insufficient to give an appropriate context matching data augmentations on these domains.

4 Conclusion and Future work

4.1 Conclusion

In this paper, we analyzed how the i-mix affects contrastive learning and the difference between i-mix loss (equipped with InfoNCE) and InfoNCE loss in section 2. The fundamental difference between them is the input gradient controlling effect on the i-mix loss, and we theoretically show that it gives not only an effect of usual contrastive learning (as in InfoNCE) but also the indirect input gradient contrastive learning effect. This input gradient controlling usually appears at the boundary of the convex hull shape of data (Figure 2), and we empirically find that this effect induces rough trained feature space at the end of the contrastive learning with i-mix loss (Figure 3). From this point of view, we expect this input gradient controlling effect is the behind logic for the e-mix (Tamkin et al., 2022) algorithm to work as a contrastive learning even without data augmentation. In section 3, as an extension of our theory, we suggest a variant of e-mix (Dmix) to overcome the complementary behavior of e-mix and ShED. As a result, it shows significant improvement in the RGB image domain and maintains comparable performance as a level of ShED or e-mix on a discrete domain. However, considering the difficulty and task of the dataset, we consider it still fails to achieve reasonable performance on the Token and Text domain.

4.2 Future work

While we theoretically show how does the i-mix and InfoNCE loss work differently, it is still unclear why the input gradient controlling effect gives better downstream performance as suggested in Lee et al. (2021). To understand this, we plan to compare feature spaces of encoder trained by i-mix and InfoNCE with SupCon ((Khosla et al., 2021)) to exclude data augmentation scheme on unsupervised learning and focus on the effect of input gradient controlling on i-mix loss. For the Dmix algorithm, it does not show good performance for the discrete domain. One possible idea to elevate the downstream performance on the discrete domain is combining MAE (masked auto-encoder, (He et al., 2021)) to Dmix. In DABS 2.0 (Tamkin et al., 2022), the MAE shows better performance compared to e-mix or ShED for the overall domain. In this case, it seems that it is desirable to combine MAE as a main algorithm with Dmix as a regularizer.

REFERENCES

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning, 2019.

Luigi Carratino, Moustapha Cissé, Rodolphe Jenatton, and Jean-Philippe Vert. On mixup regularization, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

Muthu Chidambaram, Xiang Wang, Yuzheng Hu, Chenwei Wu, and Rong Ge. Towards understanding the data dependency of mixup-style training, 2022.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild, 2013.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Naomi K. Fox, Steven E. Brenner, and John-Marc Chandonia. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Research*, 42(D1):D304–D309, 12 2013.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, 2019.

Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning, 2021.

Chanwoo Park, Sangdoo Yun, and Sanghyuk Chun. A unified analysis of mixed sample data augmentation: A loss function perspective, 2022.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pp. 108–109, 2012. doi: 10.1109/ISWC.2012.13.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection, 2019.

Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning, 2021.

Alex Tamkin, Gaurab Banerjee, Mohamed Owda, Vincent Liu, Shashank Rammoorthy, and Noah Goodman. Dabs 2.0: Improved datasets and algorithms for universal self-supervision. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, pp. 38358–38372. Curran Associates, Inc., 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

Vikas Verma, Minh-Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc V. Le. Towards domain-agnostic contrastive learning, 2021.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. Cub-200-2011, 4 2022.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss, 2021.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proc. of EMNLP*, 2019.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization?, 2021.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*, 2019.