Mixing augmentation methods

3.1.1 RECAP OF MIX-UP

Mix-up (Zhang et al., 2018) is one of the most popular data augmentation strategies to improve the generalization of the learned model by enriching the diversity of the original domain. The core idea of mix-up is to create virtual samples by randomly interpolating two samples in a convex fashion. Specifically, given two samples (x_i, y_i) and (x_j, y_j), the virtual sample $(\tilde{x}, \tilde{y})$ is defined as:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j, \tag{1}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j, \tag{2}$$

1. x_i and x_j : Image(pixel)-level / Feature(representation)-level
2. x_i and x_j : Source Domain / Target Domain(distribution)
3. x_i and x_j : Similarity (OTmix)
4. x_i and x_j : Other than 2 images

1. λ : Progressive (historical)
2. λ : Classifier / Label / Loss Function
3. λ : Sample (λ_1) and Label (λ_2)

RecursiveMix: Mixed Learning with History (NeurIPS 2022)

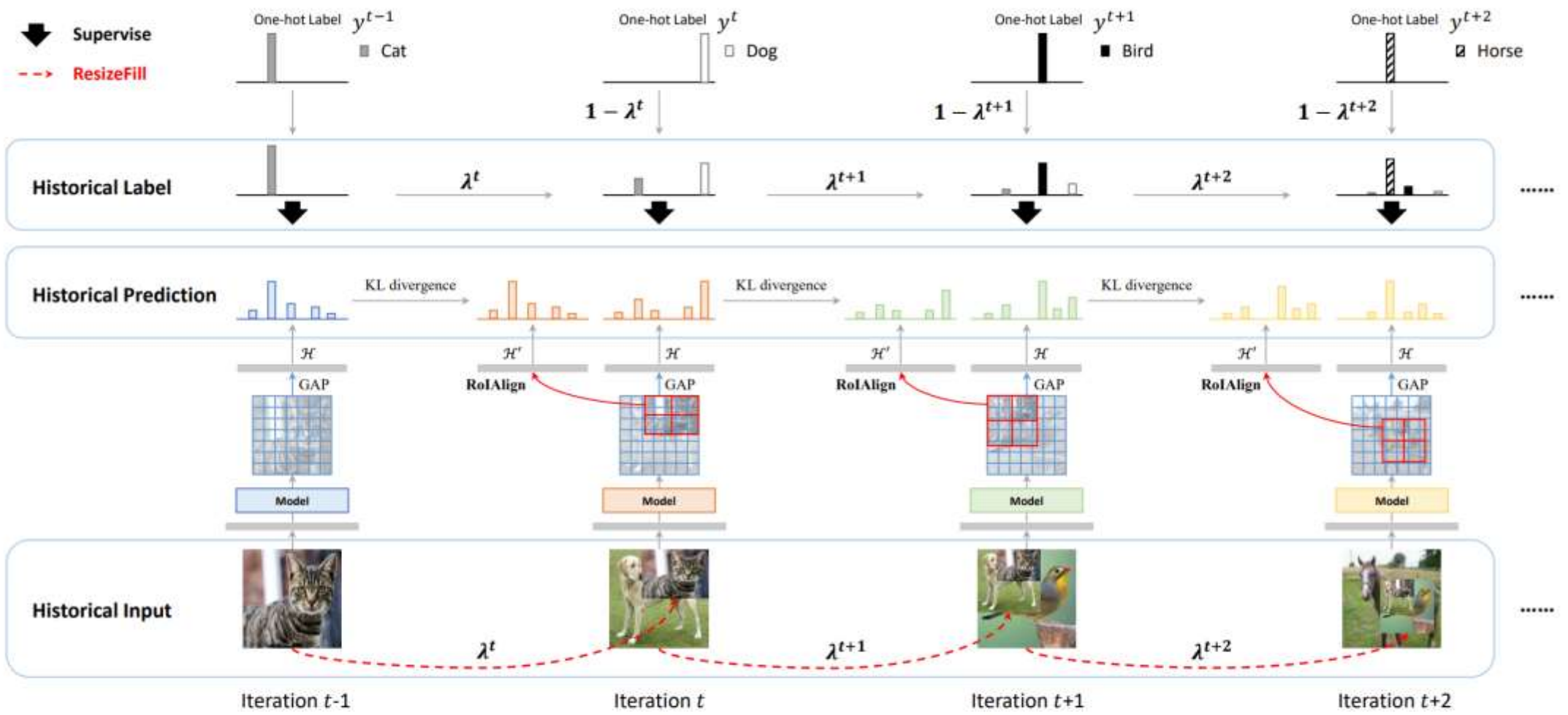


Figure 1: The illustration of the proposed RecursiveMix which leverages the historical input-prediction-label triplets. The historical input images are resized and then mixed into the current ones where the labels are fused proportionally (i.e., λ and $1-\lambda$) to the area of the operated patches, formulating a recursive paradigm.

PROGRESSIVE MIX-UP FOR FEW-SHOT SUPERVISED MULTI-SOURCE DOMAIN TRANSFER

Ronghang Zhu[1], **Xiang Yu**[2], **Sheng Li**[3,1]
[1]University of Georgia, [2]Amazon, [3]University of Virginia
ronghangzhu@uga.edu, yuxiang03@gmail.com, shengli@virginia.edu

ICLR 2023

task, model(method), and data

Such domain shift (Torralba & Efros, 2011) further results in performance degradation as models are overfitting to the training distributions. Domain adaptation (DA) (Xu et al., 2021; Zhu et al., 2021; Zhu & Li, 2021; Liu et al., 2023) has been extensively studied to address this challenge. Due to different settings regarding the source and target domains, DA problems vary into different categories such: unsupervised domain adaptation (UDA) (Zhu & Li, 2022a), supervised domain adaptation (SDA) (Motiian et al., 2017), and multi-source domain adaptation (MSDA) (Zhao et al., 2018). UDA aims to adopt knowledge from a fully labeled source domain to an unlabeled target domain. SDA intends to transfer knowledge from a fully labeled source domain to a partially labeled target domain. MSDA generalizes the UDA by adopting the knowledge from multiple fully labeled source domains to an unlabeled target domain. The main difficulty in the MSDA problem is how to achieve a meaningful alignment between the labeled source domains and the target domain that is unlabeled. Although DA has obtained some good achievements, assuming the availability of plenty of unlabeled/labeled target samples in real-world scenarios cannot be always guaranteed.

PROGRESSIVE MIX-UP

Domain adaptation (DA)	Source Domain	Target Domain
Unsupervised domain adaptation (UDA)	A fully labeled	An unlabeled
Supervised domain adaptation (SDA)	A fully labeled	A partially labeled
Multi-source domain adaptation (MSDA)	Multiple fully labeled	An unlabeled
Multi-source few-shot domain adaptation (MFDA)	A partially labeled In each source domain	An unlabeled
Few-shot Supervised Multi-source Domain Transfer (FSMDT)	Multiple fully labeled In each source domain	Few-labeled/One-labeled (no unlabeled samples)

3.1 PRELIMINARIES

In Few-shot Supervised multi-source domain Transfer (FSMDT) problem, we have M full labeled source domains and a target domain with few-shot labeled data. The i-th source domain $\mathcal{D}_{s,i} = \{(x_{s,i}^j, y_{s,i}^j)\}_{j=1}^{N_{s,i}}$ contains $N_{s,i}$ labeled samples drawn from the source distribution $P_{s,i}(x, y)$, and the target domain $D_t = \{(x_t^j, y_t^j)\}_{j=1}^{N_t}$ includes N_t labeled samples selected from the target distribution $P_t(x, y)$. Here, $N_t \ll N_{s,i}$, i.e., N_t can be as few as 1-shot per class. $P_t(x, y) \neq P_{s,i}(x, y)$, and $P_{s,i}(x, y) \neq P_{s,j}(x, y)$ where $i \neq j$. The multiple source domains and target domain have the same label space $Y = \{1, 2, \ldots, K\}$ with K categories. We aim to learn an adaptive model $\mathbf{H}$ on $\{\mathcal{D}_{s,i}\}_{i=1}^{M}$ and $\mathcal{D}_t$, that can generalize well on unseen samples from target domain. In general, $\mathbf{H}$ consists of two functions, i.e., $\mathbf{H} = \mathbf{F} \circ \mathbf{G}$. Here $\mathbf{G} : x \rightarrow g$ represents the feature extractor that maps the input sample x into an embedding space, and $\mathbf{F} : g \rightarrow f$ is the classifier with input the embedding to predict the category.

PROGRESSIVE MIX-UP

task, **model(method),** and data

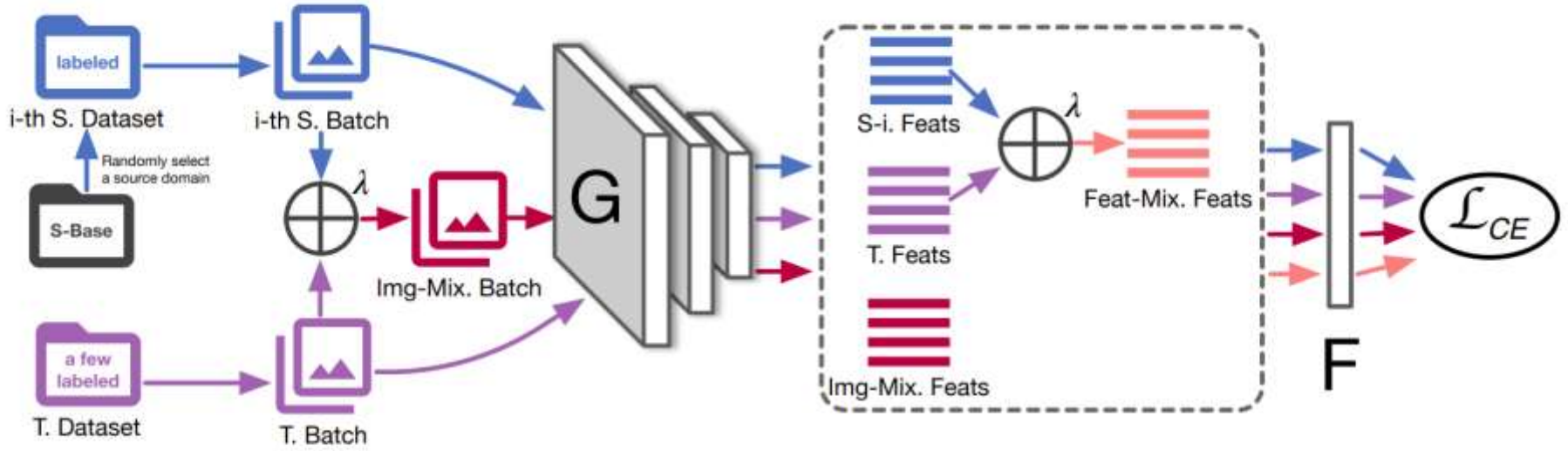


Figure 3: The training architecture. Both image and feature level P-Mixup are applied for the cross-entropy loss. **G** is the feature extractor and **F** is the classifier.

Cross-Domain Image-Level Mix-up. Motivated by the success of mix-up in self-supervised learning, we apply it to our domain transfer task, which can create new samples with new labels. We utilize it to largely enrich the target domain distribution as there are overly limited target samples. The source and target samples are linearly interpolated as:

$$\tilde{x}_{img} = \lambda x_{s,i} + (1-\lambda)x_t, \tag{3}$$

$$\tilde{y}_{img} = \lambda y_{s,i} + (1-\lambda)y_t, \tag{4}$$

where λ is the mix-up ratio. Notice that during training, such mix-up ratio can be adjusted, e.g., a larger λ generates closer-to-source samples and a smaller λ generates closer-to-target samples.

Cross-Domain Feature-Level Mix-up. On the learned feature representation manifold, mix-up at the feature level enables more intermediate virtual features to increase the feature diversity and can directly interact with the classifier $\mathbf{F}$ learning. Here, given a pair of source and target features and their corresponding labels: $(g_{s,i}, y_{s_i})$ and (g_t, y_t), we have

$$\tilde{g}_{feat} = \lambda g_{s,i} + (1-\lambda)g_t, \tag{5}$$

$$\tilde{y}_{feat} = \lambda y_{s,i} + (1-\lambda)y_t, \tag{6}$$

where λ is the mix-up ratio same as the one used in image-level mix-up. With exactly the same λ, we argue that the image-level mix-up samples lie in the same feature space as the feature-level mix-up samples. Thus, we can jointly utilize the two for penalty, i.e., the same class image-level mix-up and feature-level mix-up should go for the same classification result.

PROGRESSIVE MIX-UP

To alleviate it, we dig into the Wasserstein distance of "source-to-mixup" $d_w(\mathcal{G}_s, \mathcal{G}_{mix})$ and "target-to-mixup" $d_w(\mathcal{G}_t, \mathcal{G}_{mix})$, where $\mathcal{G}_s, \mathcal{G}_t, \mathcal{G}_{mix}$ stand for the embeddings of source, target and mix-up domains. We observe that during the training, if the mix-up domain initially is closer to the few-shot target domain, the alignment is relatively simple as $d_w(\mathcal{G}_t, \mathcal{G}_{mix})$ is already small while $d_w(\mathcal{G}_s, \mathcal{G}_{mix})$ can be effectively minimized as there are sufficient source domain data. When gradually increasing the mix-up ratio towards closer to source domains, since we already harness the "target-to-mixup" distance to be small, we are pushing the entire mix-up domain and few-shot target domain towards the source domains, as illustrated in Figure 2. Such progressively adjusted mix-up ratio, following the spirit of curriculum learning (Bengio et al., 2009), eases the initial large domain gap by mildly starting close to the target, and secures the entire transfer process smoothly.

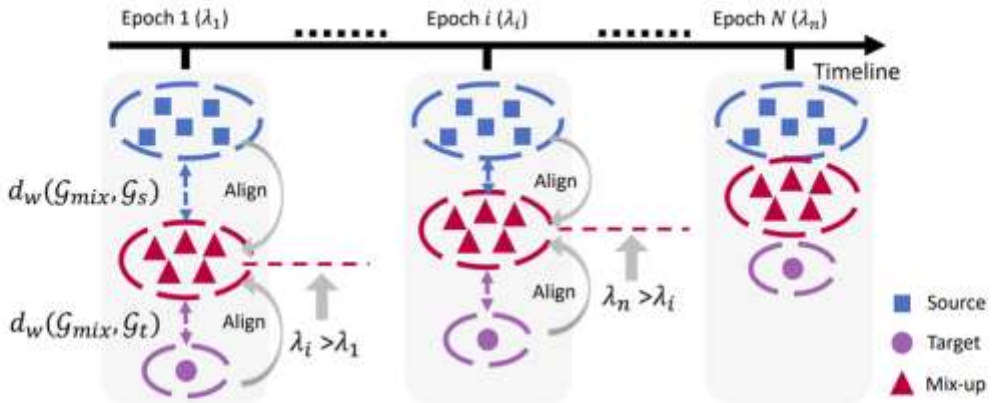


Figure 2: The flowchart of the proposed progressive mix-up. A mix-up domain (red) is introduced as initially closer to the target domain. By enforcing the mix-up ratio λ to be progressively increasing based on the wasserstein distance of source-to-mixup and target-to-mixup, we push the mix-up domain gradually to be closer to source domains, and thus achieving the alignment of multi-source domain to the few-shot target domain.

Specifically, we introduce a weighting factor q to depict the closeness to source as:

$$q = \exp(-\frac{d_w(\mathcal{G}_s, \mathcal{G}_{mix})}{(d_w(\mathcal{G}_s, \mathcal{G}_{mix}) + d_w(\mathcal{G}_t, \mathcal{G}_{mix}))T}), \tag{7}$$

where T is a temperature factor defined as 0.05. During training, by initializing $\mathcal{G}_{mix}$ closer to target domain, such q is small. To progressively adjust it, we consider to apply this closeness on top of the previous stage λ in a moving average manner. Further, a linearly incremental component is introduced to enforce the gradual closeness to the source domains. The progressive mix-up is formulated as:

$$\lambda_n = \frac{n(1-q)}{N} + q\lambda_{n-1}, \tag{8}$$

where N is the total number of iterations and n is the current iteration index. Initial weighting λ_0 towards source is 0. To numerically stabilize the training procedure, we introduce a uniform distribution U, a random perturbation on top of the current λ_n:

$$\tilde{\lambda}_n = \text{Clamp}(U(\lambda_n - \sigma, \lambda_n + \sigma), \text{min} = 0.0, \text{max} = 1.0), \tag{9}$$

where σ is a local perturbation range, i.e., we empirically set it as 0.2. $\tilde{\lambda}_n$ is then stochastically sampled and clamped into range $[0.0, 1.0]$ for each iteration n's mix-up ratio.

PROGRESSIVE MIX-UP

task, model(method), and **data**

Table 1: mAP(%) on Office-Home. Named in row is the target domain which contains 10 classes randomly selected from label space. (A: Art, C: Clipart, P: Product, R: Real_world)

$\mathcal{D}_t^{10}$	ERM-w/o	ERM-w	CCSA	MDAN	Mix-up	DAML	URT	Ours
A	60.48	60.83	64.52	58.49	59.50	60.30	59.48	**72.23**
C	44.06	46.35	56.32	44.17	51.03	49.88	52.41	**59.97**
P	72.55	72.75	75.89	70.43	73.03	72.96	82.11	**82.69**
R	79.32	76.40	79.17	74.28	76.91	76.17	81.52	**85.05**
Ave.	63.83	64.08	68.97	61.84	65.12	64.83	68.88	**74.99**

Table 2: mAP(%) on DomainNet. Named in column is the target domain contains 10/15 classes randomly selected from label space. (C: Clipart, P: Painting, R: Real, S: Sketch)

Method	$\mathcal{D}_t^{10}$					$\mathcal{D}_t^{15}$				
	C	P	R	S	Avg.	C	P	R	S	Avg.
ERM-w/o	53.58	48.03	57.70	45.57	51.22	51.90	45.68	57.11	43.51	49.55
ERM-w	54.91	48.39	58.27	48.27	52.46	54.50	47.51	58.23	45.17	51.35
CCSA	58.78	54.40	61.32	56.74	57.81	53.51	50.90	58.63	52.11	53.79
MDAN	56.07	48.50	59.32	47.40	52.82	54.98	46.95	59.24	45.36	51.63
Mix-up	61.81	59.49	66.41	56.17	60.97	56.85	53.98	64.51	50.25	56.40
DAML	58.26	49.33	55.53	47.15	52.57	56.56	47.63	57.39	46.45	52.01
URT	67.18	56.87	**84.20**	55.08	65.83	53.07	47.99	72.53	41.00	53.65
Ours	**79.32**	**70.18**	82.63	**69.88**	**75.50**	**72.46**	**64.37**	**76.88**	**60.41**	**68.53**

Table 3: Ablation study on Office-Home. Named in column is the target domain which contains 10 classes randomly selected from the label space.

Mix-up Ratio λ	Method	Art	Clipart	Product	Real_world	Ave.
N/A	ERM-w (no mix-up)	60.48	44.06	72.55	79.32	63.83
Random Sampling	Feat-Mix	60.06	48.21	69.06	73.56	62.72
	Img-Mix	59.50	51.03	73.03	76.91	65.12
	Feat-Mix + Img-Mix	66.40	56.18	76.51	78.98	69.52
Progressive Update	Feat-Mix	64.89	55.08	75.41	77.52	68.22
	Img-Mix	68.45	57.55	81.26	83.89	72.79
	Feat-Mix + Img-Mix	72.23	59.97	82.69	85.05	74.99

Geodesic Multi-Modal Mixup for Robust Fine-Tuning

Junhyuk So*[2], Changdae Oh[1], Yongtaek Lim[1], Hoyoon Byun[1], Minchul Shin[3], Kyungwoo Song†[1]

[1]Department of Artificial Intelligence, University of Seoul
[2]Department of Computer Science, POSTECH
[3]Reality Labs, Meta

Arxiv 2022

task, model(method), and data

2.3 Multi-modal Learning

With the rapid development of technology, from various sources, multi-modal data such as images, texts, and videos are flooding the Internet. Thereafter, the universal intelligence system that can process the multi-modal data separately or simultaneously has attracted huge attention from both industry and academy. There are two branches of such works: (1) universal architecture for multi-modal data flow [36, 37, 38], (2) representation learning under multi-modal interaction [6, 39, 40]. In this work, we focus on CLIP [6], one of the most popular multi-modal representation learning model.

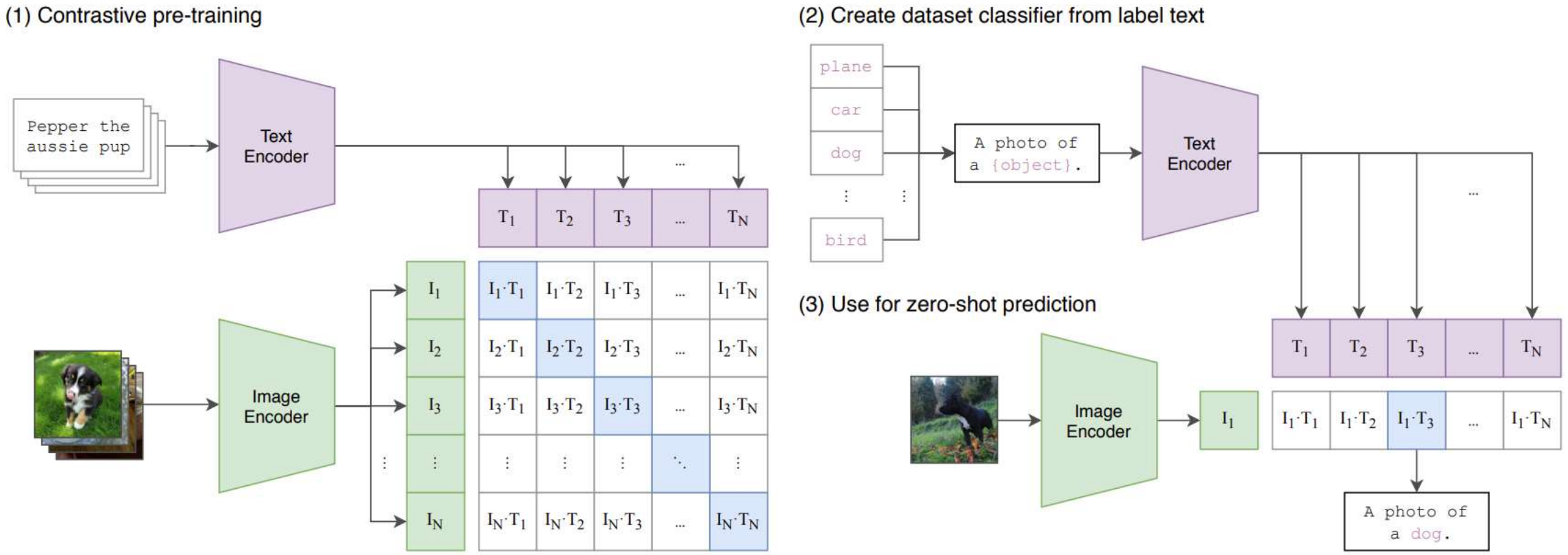


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Multi-modal learning handles the multi-modal dataset stemming from a heterogeneous domain such as image and text. For given batch $D = \{x_i, y_i\}_{i=1}^n$ where (x_i, y_i) denotes the i-th image-text positive pairs, and n is the number of batch size, the goal of multi-modal learning is to learn the relationship between the pairs of image and text. In other words, CLIP trains the image encoder $f(\cdot|\theta_1)$ and text encoder $g(\cdot|\theta_2)$ that $I_i = f(x_i)$ and $T_i = g(y_i)$ have similar embeddings as shown in Eq. 1, 2, where θ_1 and θ_2 are learnable parameters. It is noted that I_i and T_i are L^2 normalized unit vectors, and they lie on the hyperspherical embedding space. CLIP adopts the InfoNCE [5] loss $C(\cdot, \cdot)$ to encourage the similarity between positive pairs (x_i, y_i) and the dissimilarity between all remain negative pairs (x_i, y_j).

$$C(I,T) = \frac{1}{n}\sum_{i=1}^{n} -\log \frac{\exp(\text{sim}(I_i, T_i)/\tau)}{\sum_{j=1}^{n}\exp\left(\text{sim}(I_i, T_j)/\tau\right)} \tag{1}$$

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{2}(C(I,T) + C(T,I)) \tag{2}$$

$\text{sim}(\cdot)$ calculate the similarity between two given vectors, and CLIP uses simple dot product as a $\text{sim}(\cdot)$ operation. It is noted that, unlike many previous studies, τ is a *learnable* temperature that scales the magnitude of the calculated similarity.

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

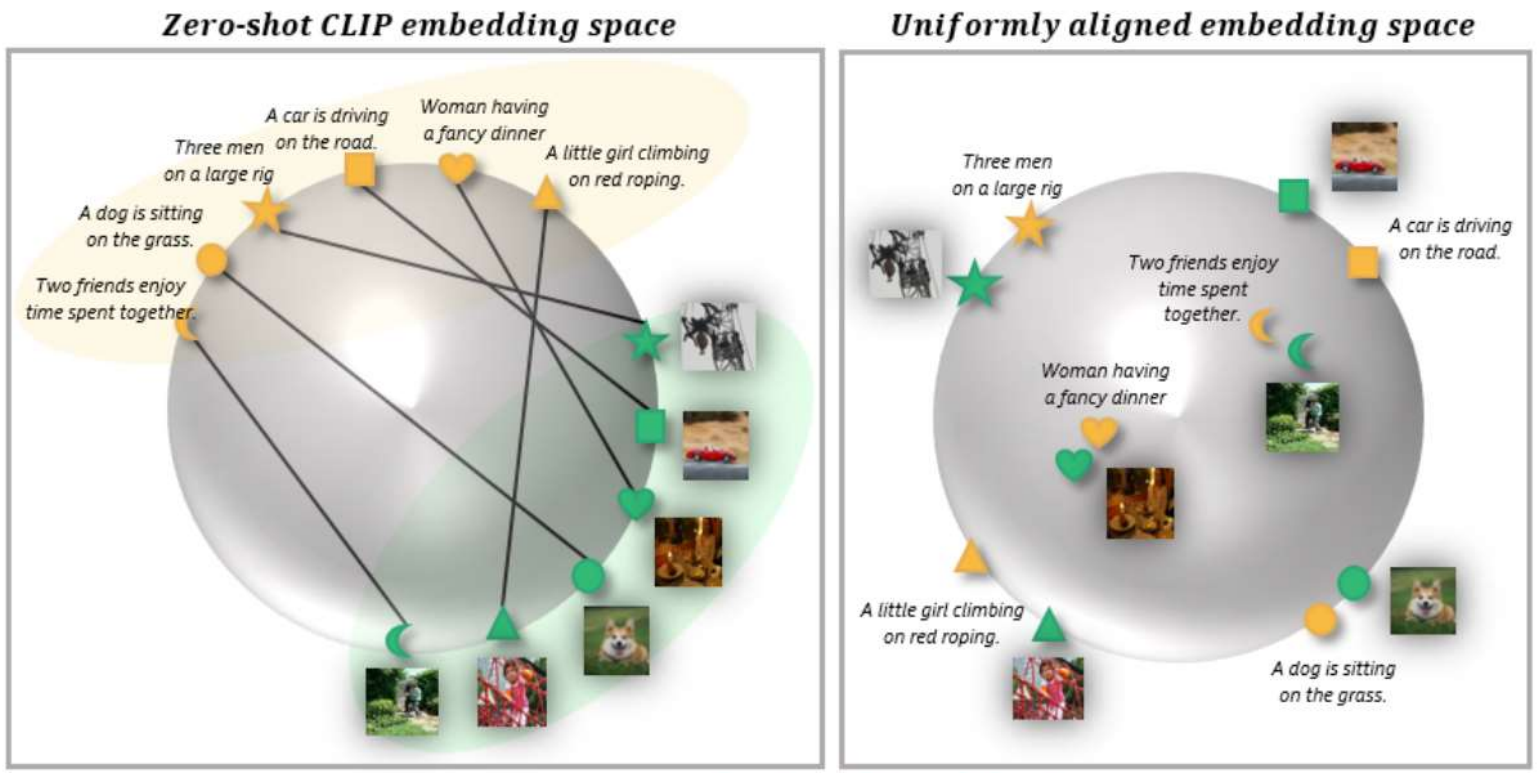
Geodesic Multi-Modal Mixup

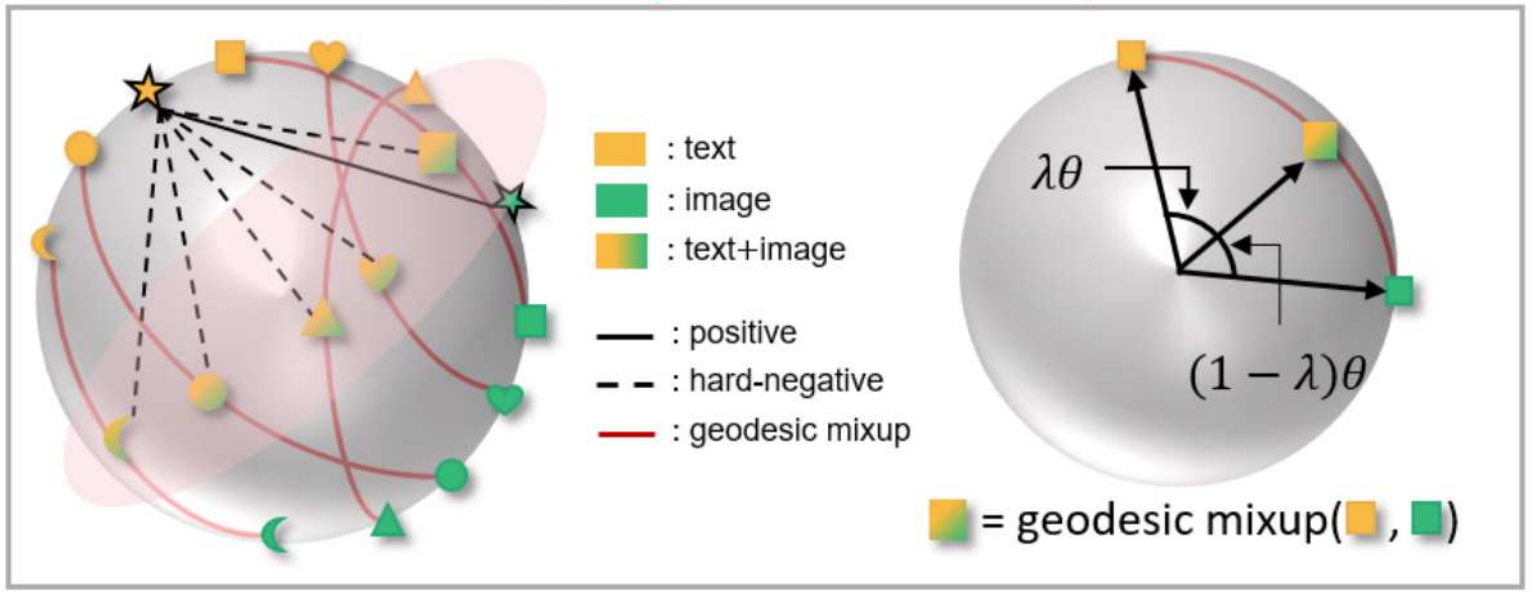
Recently, multi-modal learning such as CLIP also incorporates contrastive learning and the L^2 regularization. Different from uni-modal learning, multi-modal learning handles multiple heterogeneous data such as images and text. However, the analysis for multi-modal learning has not been well studied yet. In this paper, we find that the CLIP embedding has limited uniformity and alignments, and it may harm the downstream tasks.

persphere, the uniformity measures how the features are uniformly distributed, and the alignment measures how the features among the positive samples locate in close distances in the latent space

embedding, so-called *Uniformity and Alignment*. Uniformity indicates the degree of preserving maximal information throughout the whole embedding space, while alignment denotes whether similar inputs have similar features. Better uniformity and alignment are interpreted as representation ro-

Geodesic


Zero-shot CLIP embedding space
Uniformly aligned embedding space
A car is driving on the road.
Woman having a fancy dinner
Three men on a large rig
A little girl climbing on red roping.
A dog is sitting on the grass.
Two friends enjoy time spent together.
Three men on a large rig
A car is driving on the road.
Two friends enjoy time spent together.
Woman having a fancy dinner
A little girl climbing on red roping.
A dog is sitting on the grass.

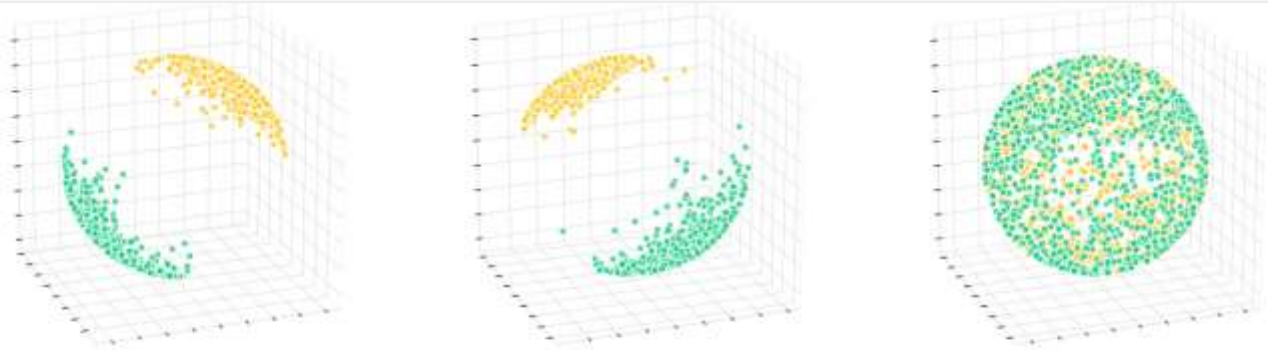

Fine-tuning with geodesic multi-modal mixup
: text
: image
: text+image
: positive
: hard-negative
: geodesic mixup
$\lambda\theta$
$(1-\lambda)\theta$
= geodesic mixup(,)

Geodesic Multi-Modal Mixup

Alignment (align.) evaluates the difference between the similarity of positive pairs compared with the hardest negative pairs, while uniformity (unif.) denotes the degree of maximal information. It is

$$\text{align.} = -\mathbb{E}_{(x_i,y_i)}\left[\|f(x_i) - g(y_i)\|_2^2 - \min_{k\neq i}\|f(x_i) - g(y_k)\|_2^2\right]$$

$$\text{unif.} = -\log \mathbb{E}_{(x_i,y_j)}\left[\exp\left(-2\|f(x_i) - g(y_j)\|_2^2\right)\right]$$



(a) ZS (0.04, 1.98) (b) FT (0.07, 1.97) (c) m^2-Mix (0.17, 3.71)

Figure 2: DOSNES [49] embedding visualization of (a) pre-trained CLIP (ZS), (b) fine-tuned CLIP (FT) with learnable τ, respectively. And (c) fine-tuned CLIP with our methods on Flickr 30k [50] test set. Green denotes the image embedding, and orange denotes the text embedding. The values in parentheses denotes the alignment and uniformity, respectively. We find that the embedding space of the pre-train CLIP (a) and its naively fine-tuned counterpart (b) have two separate regions, and it may harm the robustness. our methods (c) induce a more improved uniformed and well aligned representation. Improved learned representation contributes to improve the diverse downstream task performance.

Geodesic Multi-Modal Mixup

task, **model(method),** and data

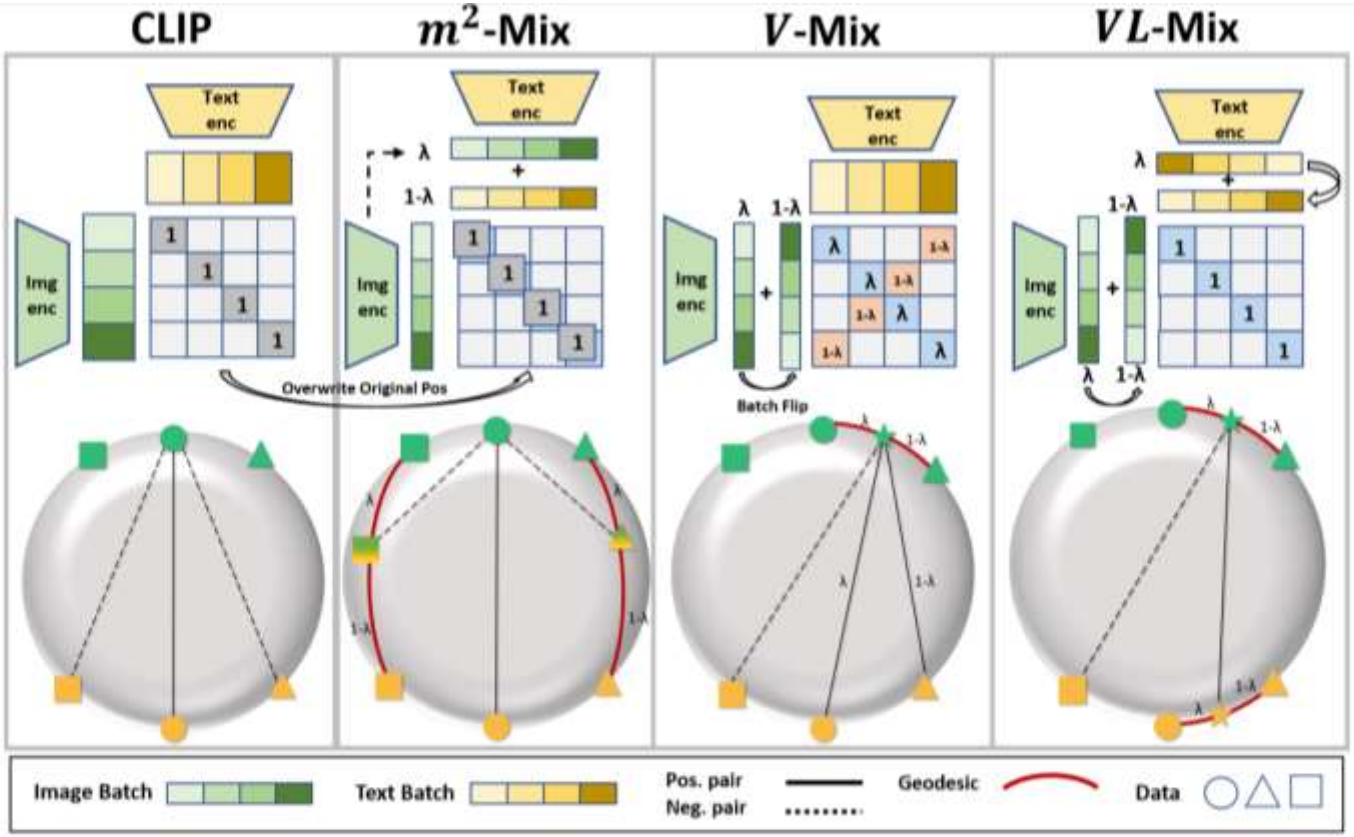
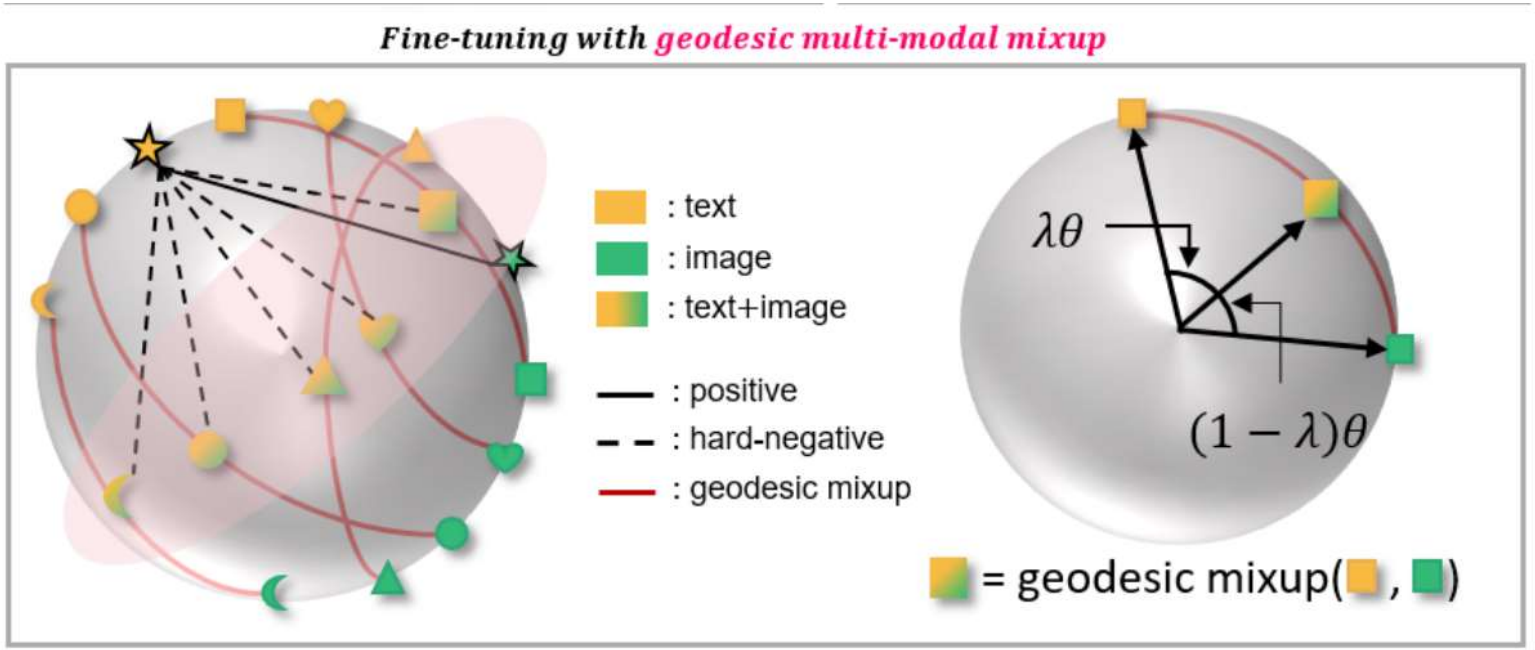


Figure 3: Overall model structure of CLIP and our proposed methods. CLIP compares the similarity between the original positive pairs and negative pairs. However, m^2-Mix generates the hard negative samples by mixing the heterogeneous data, image, and text. Additionally, we propose the V-Mix and VL-Mix, a uni-modal Mixup for CLIP. Multi-modal Mixup, m^2-Mix, and uni-modal Mixup, V-Mix, VL-Mix, contributes to the robust representation learning.

Geodesic Multi-Modal Mixup

$$mix_\lambda(a,b) = a\frac{\sin(\lambda\theta)}{\sin(\theta)} + b\frac{\sin((1-\lambda)\theta)}{\sin(\theta)} \tag{3}$$

$$\text{where } \theta = \cos^{-1}(a \cdot b) \text{ and } \lambda \sim \text{Beta}(\alpha, \beta)$$



Geodesic Multi-Modal Mixup

$$C(I,T) = \frac{1}{n}\sum_{i=1}^{n} -\log \frac{\exp(\text{sim}(I_i, T_i)/\tau)}{\sum_{j=1}^{n} \exp\left(\text{sim}(I_i, T_j)/\tau\right)} \tag{1}$$

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{2}(C(I,T) + C(T,I)) \tag{2}$$

$$C_{m^2}(I,T) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\exp(I_i \cdot T_i)}{\sum_{j=1}^{n} \exp(I_i \cdot mix_\lambda(I_j, T_j))} \tag{4}$$

$$\mathcal{L}_{m^2-\text{Mix}} = \frac{1}{2}(C_{m^2}(I,T) + C_{m^2}(T,I)) \tag{5}$$

Theorem 1. *Let's assume that two random variables x_1 and x_2 follow the $M_d(\mu_1, \kappa)$ and $M_d(\mu_2, \kappa)$, von Mises–Fisher distribution with mean direction μ_1, μ_2 and concentration parameter κ in $\mathbb{R}^d$, respectively. Let $\widetilde{x} = x_1 + x_2$ and $d = 2$. Then, $\mathrm{KL}(p(x_1)||p(\widetilde{x})) \leq \mathrm{KL}(p(x_1)||p(x_2))$ for sufficiently large κ.*

Proof. Let $M_d(\mu, \kappa) = C_d(\kappa)\exp(\kappa\mu^T x)$, where I denotes the modified Bessel function, and $C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{(d/2)} I_{d/2-1}(\kappa)}$. Let $x_1 \sim M_2(\mu_1, \kappa)$ and $x_2 \sim M_2(\mu_1, \kappa)$. From [51], $\widetilde{x} \sim M_2(\widetilde{\mu}, \widetilde{\kappa})$ where $\widetilde{\mu} = \mu_1 + \mu_2$ and $\widetilde{\kappa} = A^{-1}(A(\kappa)A(\kappa))$, approximately, where $A(\kappa) = \frac{I_1(\kappa)}{I_0(\kappa)}$ and A^{-1} is its inverse. From [52], $KL(p(x_1)||p(x_2)) = \kappa A(\kappa)(1-\cos(\mu_1 - \mu_2))$. Similarly, $KL(p(x_1)||p(\widetilde{x})) = \log I_0(\widetilde{\kappa}) - \log I_0(\kappa) + \widetilde{\kappa}A(\widetilde{\kappa})(1 - \cos(\mu_1 - \mu_2))$. From [53] and [54], $I_0(\kappa) \approx \frac{\exp(\kappa)}{\sqrt{2\pi\kappa}}(1 + \frac{1}{8\kappa})$ and $I_1(\kappa) \approx \frac{\exp(\kappa)}{\sqrt{2\pi\kappa}}(1 - \frac{3}{8\kappa})$ for sufficiently large κ.
Therefore, $\mathrm{KL}(p(x_1)||p(\widetilde{x})) \leq \mathrm{KL}(p(x_1)||p(x_2))$ for sufficiently large κ. □

4.2.1 V-Mix, L-Mix

$$C_V(I,T) = \frac{1}{n}\sum_{i=1}^{n} -\lambda \log \frac{\exp(mix_\lambda(I_i, I_{i'}) \cdot T_i)}{\sum_{j=1}^{n} \exp(I_i \cdot T_j))} - (1-\lambda) \log \frac{\exp(mix_\lambda(I_i, I_{i'}) \cdot T_{i'})}{\sum_{j=1}^{n} \exp(I_i \cdot T_j))}$$

$$C_L(I,T) = \frac{1}{n}\sum_{i=1}^{n} -\lambda \log \frac{\exp(I_i \cdot mix_\lambda(T_i, T_{i'}))}{\sum_{j=1}^{n} \exp(I_i \cdot T_j))} - (1-\lambda) \log \frac{\exp(I_{i'} \cdot mix_\lambda(T_i, T_{i'}))}{\sum_{j=1}^{n} \exp(I_i \cdot T_j))}$$

$$\mathcal{L}_{uni-\text{Mix}} = \frac{1}{2}(C_V(I,T) + C_L(I,T))$$

4.2.2 VL-Mix

$$C_{VL}(I,T) = \frac{1}{n}\sum_{i=1}^{n} -\log \frac{\exp(mix_\lambda(I_i, I_{i'}) \cdot mix_\lambda(T_i, T_{i'}))}{\sum_{j=1}^{n}\exp(I_i \cdot T_j))}$$

$$\mathcal{L}_{VL-\text{Mix}} = \frac{1}{2}(C_{VL}(I,T) + C_{VL}(T,I))$$

4.2.3 m^3-Mix

We find that our uni-modal Mixup and VL-Mixup on CLIP alleviate the over-confident problems as well as the m^2-Mix. We name the combination of m^2-Mix with uni-Mix and VL-mix as m^3-Mix, multiple multi-modal Mixup. The overall objective function of m^3-Mix is as follows.

$$\mathcal{L}_{m^3-\text{Mix}} = \mathcal{L}_{CLIP} + \mathcal{L}_{m^2-\text{Mix}} + \mathcal{L}_{uni-Mix} + \mathcal{L}_{VL-Mix}$$

Geodesic Multi-Modal Mixup

task, model(method), and **data**

In this study, we find that CLIP embedding has a large gap between image and text embedding. From our findings, we provide a m^2-Mix that generates the hard negative samples for robust representation learning. Section 5.2. provides the properties of generated samples, and Section 5.3-5.5 provide the extensive experiments including retrieval, calibration, structure-awareness [55], and robustness on modality missing in sentiment analysis (emotion classification) task [56]. We provide the implementation code and more detailed experimental results on Appendix A and Appendix B, respectively[2].

Table 1: Performance of image to text (i→t) retrieval and text to image retrieval (t→i). ZS and FT denote the pre-trained CLIP and fine-tuned CLIP, respectively, and R1 and R5 denote the recall at top-1 and top-5.

	Flickr30k				MS COCO			
	i→t		t→i		i→t		t→i	
	R1(↑)	R5(↑)	R1(↑)	R5(↑)	R1(↑)	R5(↑)	R1(↑)	R5(↑)
ZS	71.1	90.4	68.5	88.9	31.93	56.87	28.54	53.12
FT	81.2	95.4	80.70	95.8	36.65	63.60	36.85	63.88
FT($\tau = 0.05$)	81.2	95.3	80.62	95.5	32.04	60.18	33.4	61.3
FT($\tau = 0.10$)	71.2	93.5	75.5	92.2	25.23	53.51	27.52	53.9
i-Mix	78.2	95.0	77.4	94.4	19.84	43.86	19.76	44.32
Un-Mix	77.6	95.2	76.6	94.2	36.69	64.07	37.0	64.36
m^3-Mix	**82.1**	**96.8**	**81.9**	**95.9**	**37.99**	**66.47**	**39.94**	**67.80**

Table 3: The retrieval performance on Flickr30k. ZS denotes the combination of BERT and ResNet-50 that were pre-trained on each uni-modal dataset only, which has no capability for retrieval. Our m^3-Mix achieves best performance.

	i → t			t → i		
	R1(↑)	R5(↑)	R10(↑)	R1(↑)	R5(↑)	R10(↑)
ZS	0.1	0.4	0.9	0.1	0.2	0.9
FT	17.8	46.0	59.1	17.3	46.0	59.2
FT($\tau = 0.05$)	17.8	43.2	59.8	18.8	45.7	58.8
FT($\tau = 0.10$)	14.1	36.2	51.6	13.8	38.3	50.9
i-Mix [32]	14.2	40.4	56.4	17.9	42.9	56.6
Un-Mix [33]	18.7	46.6	60.2	19.3	47.8	62.2
m^3-Mix	**19.5**	**49.0**	**63.8**	**21.6**	**49.7**	**63.6**

Geodesic Multi-Modal Mixup

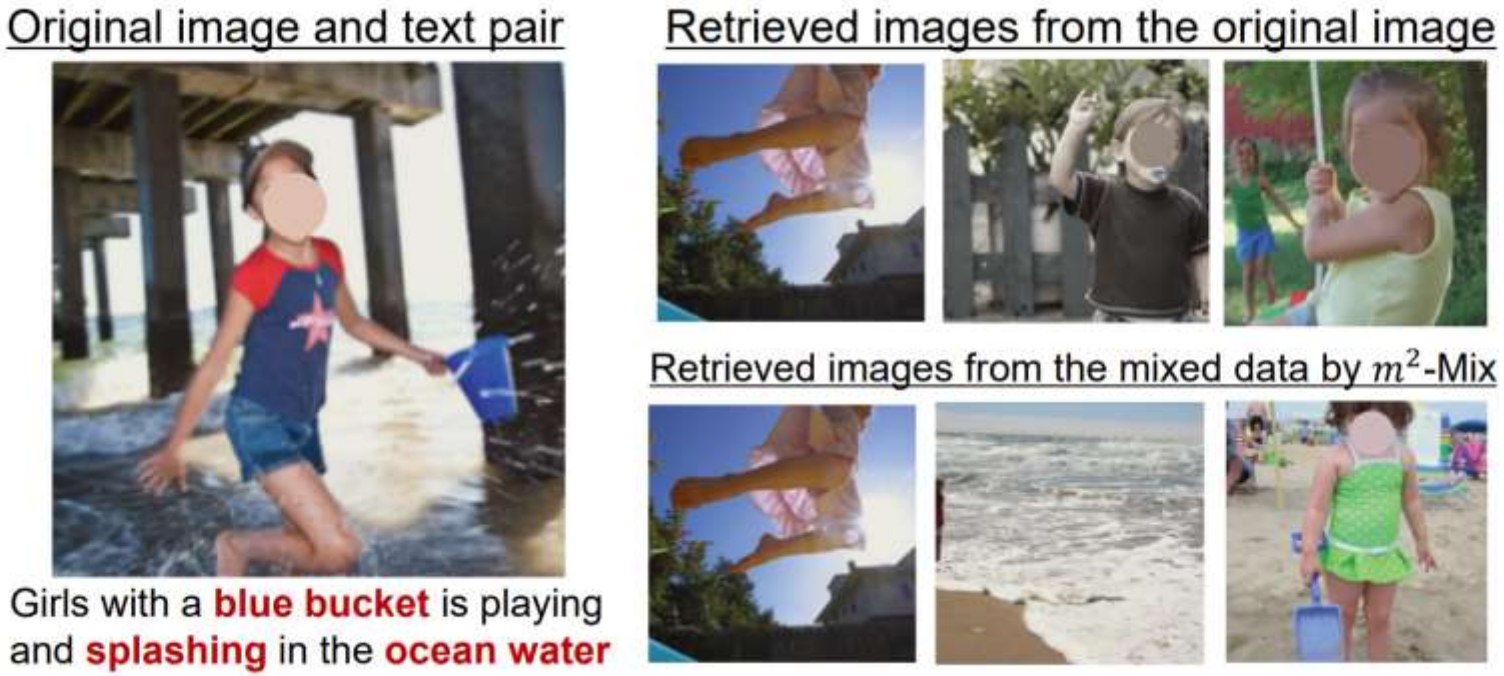


Figure 4: Left denotes the original image and text pair. The right top and bottom denote the retrieved top-3 images from the original image and the mixed data by m^2-Mix, respectively. The retrieved images from the m^2-Mix has both characteristics of the image and text, including the splashing, ocean, and blue bucket, as shown in the text and image. However, the retrieved images from the original images are not related to the blue bucket or ocean water. Appendix B provides the additional case studies.

Geodesic Multi-Modal Mixup

Two small children , one wearing a red winter jacket and the other a black one , are slipping and sliding down an icy slope .

Several men and women stand in front of a yellow table that is covered in piles of potatoes .

two girls are sitting together while playing their instruments .

Three men wearing white polo shirts and green baseball caps stand behind a banner between two buildings .

Geodesic Multi-Modal Mixup

Figure 8: SIMAT Transformation example from CLIP fine-tuned from our model. m^3-Mix shows the many correct cases compared to the zero-shot CLIP (ZS). The last column shows the failure case of ZS and m^3-Mix, which represents the "bear laying in bed", the relatively irregular example.

1. Long-tailed + CLIP (over-sampling)
2. λ (historical*)
3. Prompt (select)
4. Prompt + Image ->negative pairs (adversarial)

吉林大学

THANKS