



重庆工商大學
Chongqing Technology and Business University

Sample and Label Policies for Mixup in Supervised Learning

金 鑑

May, 2024

Contents

1. Sample Policies

2. Label Policies

1.1 Static Linear

1.2 Feature-based

1.3 Cutting-based

1.4 k Sample Mixup

1.5 Random Policies

1.6 Style-based

1.7 Saliency-based

1.8 Attention-based

1.9 Generating Samples

1.1 Static Linear

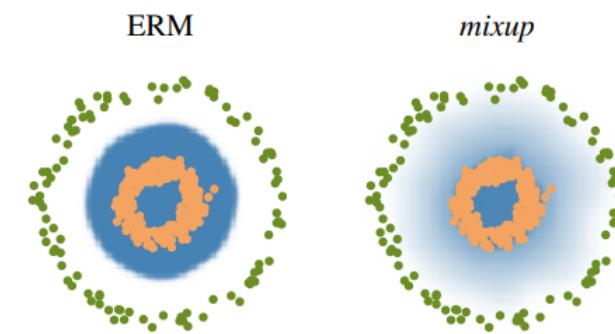
mixup: Beyond Empirical Risk Minimization (ICLR`2018)

*mixup*是混合增强手段的“始祖”，它通过将mini-batch中两对随机样本经过简单的全局线性插值得到增强混合样本。同样的，对于样本对应的标签，也是简单的按照比例 λ 进行差值得到增强标签。

其中， λ 被称为混合比例(mixing ratio)，是通过一个 $Beta(a, a)$ 分布采样得到。在得到混合样本以及标签后，输入到网络模型中再进行分类和损失计算，在*mixup*方法中，损失计算被称为*mixup cross entropy*(MCE)。

```
# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    optimizer.zero_grad()
    loss(net(x), y).backward()
    optimizer.step()
```

(a) One epoch of *mixup* training in PyTorch.



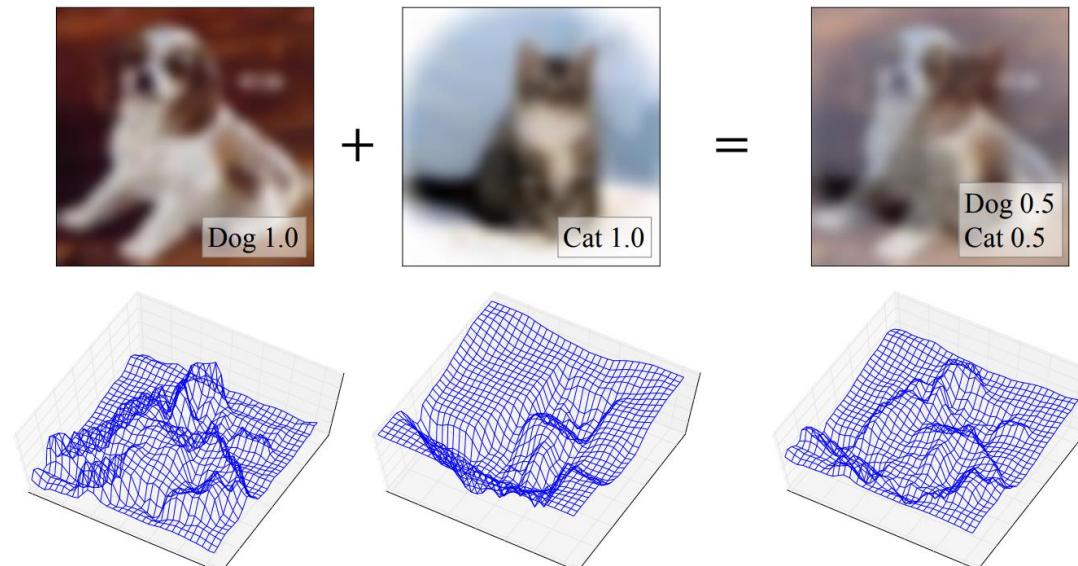
(b) Effect of *mixup* ($\alpha = 1$) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y = 1|x)$.

1.1 Static Linear

Between-class Learning for Image Classification (CVPR`2018)

BC是同年和mixup提出的方法，BC最开始是作者应用在语音识别领域，后应用在CV任务上。

它同样是按照全局线性插值得到增强样本，与mixup稍微不同的是，BC的混合比例 λ 是从一个均匀分布 $Uniform(0, 1)$ 得来。

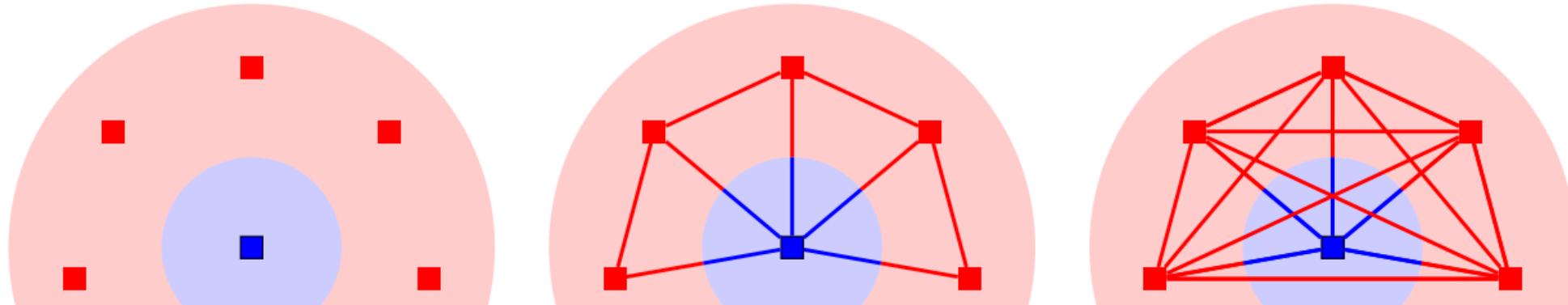


1.1 Static Linear

Preventing Manifold Intrusion with Locality: Local Mixup (EUSIPCO`2022)

Local Mixup目的是解决mixup在一些特殊的情况下产生“流形侵入”(Manifold Intrusion)问题。

通过对样本进行低纬空间上的度量，计算样本之间的距离并标记权重(越远权重越低)，在进行混合时，限制距离过远的样本进行混合，达到减少“流形侵入”的问题。

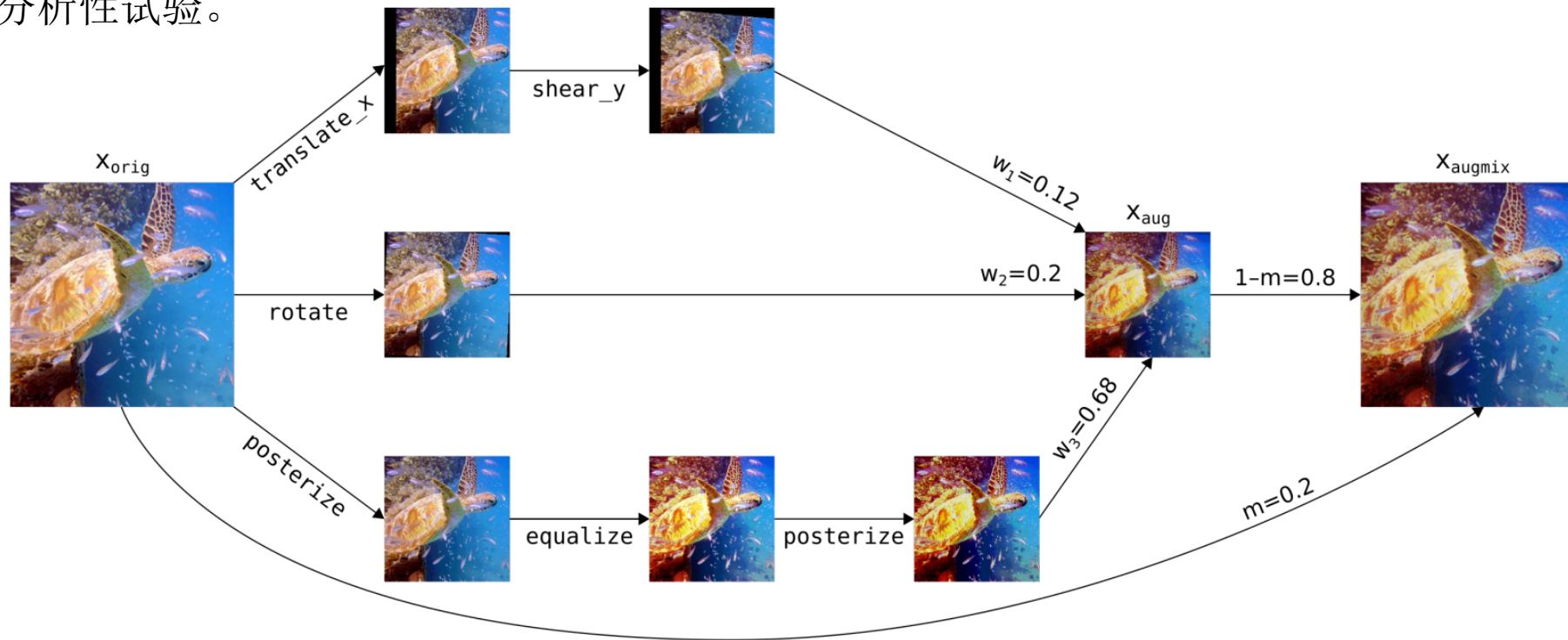


1.1 Static Linear

AugMix: A Simple data Processing Method to Improve Robustness and Uncertainty (ICLR`2020)

AugMix通过混合自身不同的增强样本得到混合增强样本。

AugMix的目的是提升神经网络的鲁棒性和不确定性能能力，所以在实验方面也是首次使用了CIFAR100-C和ImageNet-C等“腐蚀性数据集”来验证网络模型和mixup带来的增益。为后续方法提供了一个新的分析性试验。

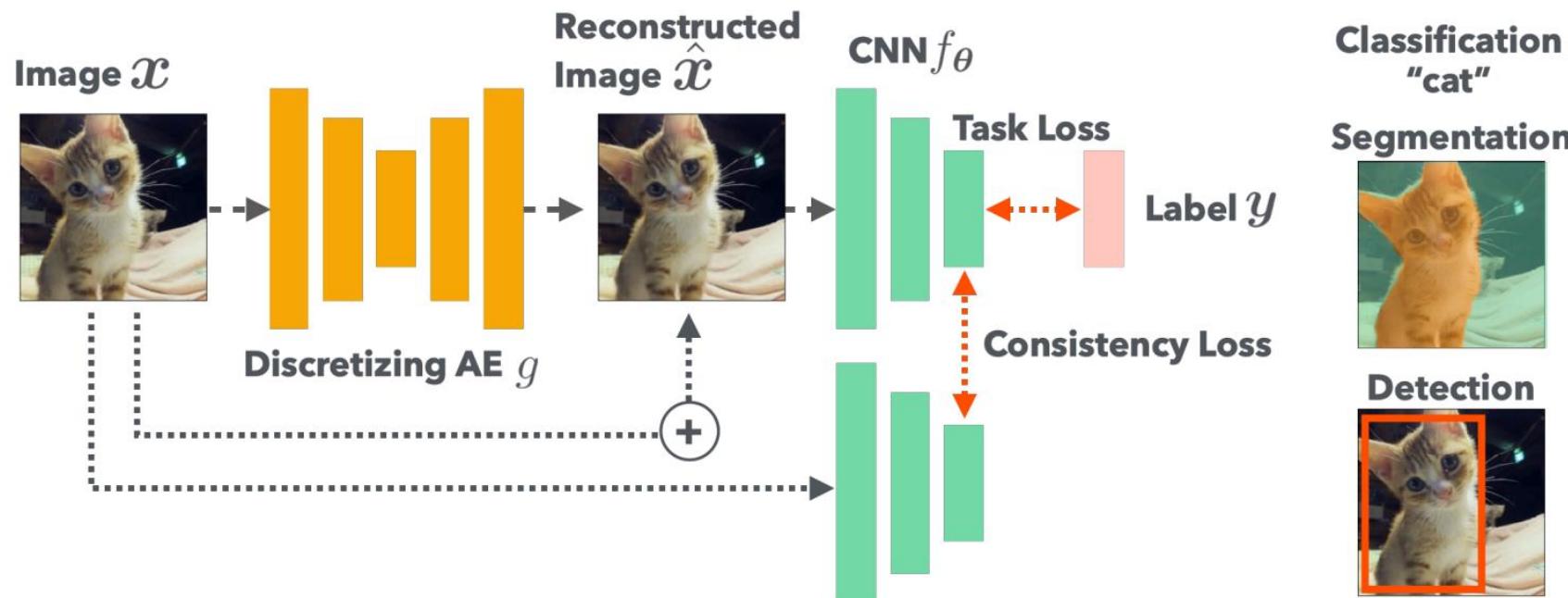


1.1 Static Linear

DJMIX: Unsupervised Task-agnostic Image Augmentation for Improving Robustness of Convolutional Neural Networks (IJCNN`2022)

CNNs容易受到输入图像上看不见的噪声影响，例如散焦模糊或JPEG压缩伪影。提高此类噪声的稳健性对于现实世界的应用非常重要。

DJMIX通过一个离散的AutoEncoder来重构样本作为源样本，在与Raw样本进行线性插值得到混合样本，通过分类/分割/目标检测等任务的损失和源样本&混合样本的一致性损失来进行训练。



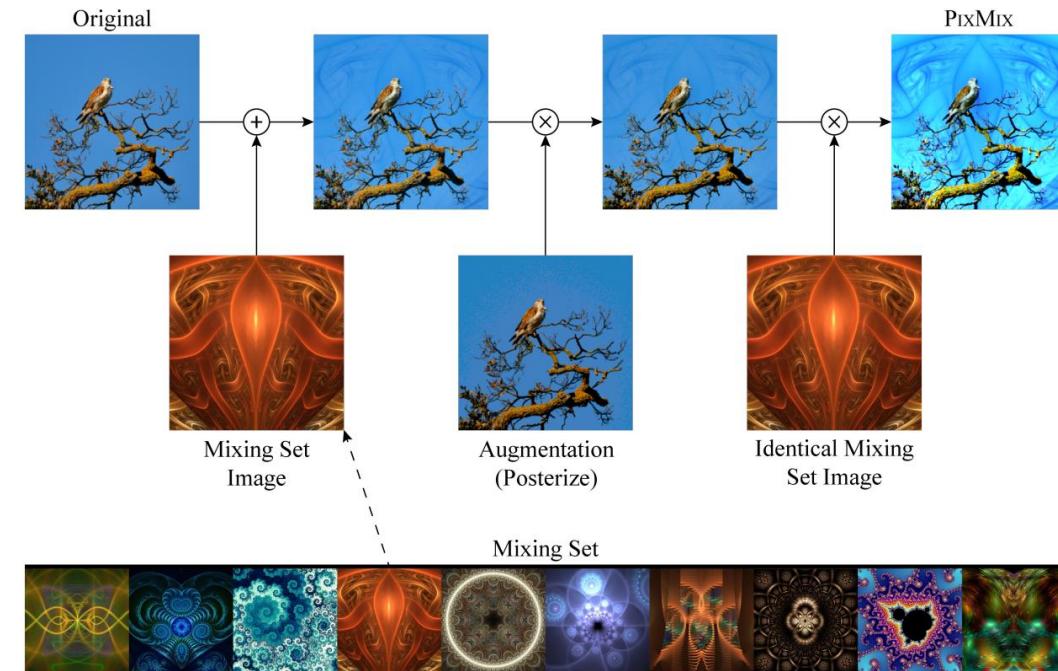
1.1 Static Linear

PixMix: Dreamlike Pictures Comprehensively Improve Safety Measures (CVPR`2022)

PixMix利用额外的纹理图像对样本进行混合。

不仅仅只是线性插值混合，也是用到了像素点积。目的也是改进网络模型的鲁棒性和calibration能力。

在Corruptions, Consistency, Adversaries, Calibration, Anomaly Detection指标下相比于其他方法都达到了最优，说明PixMix能够提升网络模型的OOD鲁棒性。



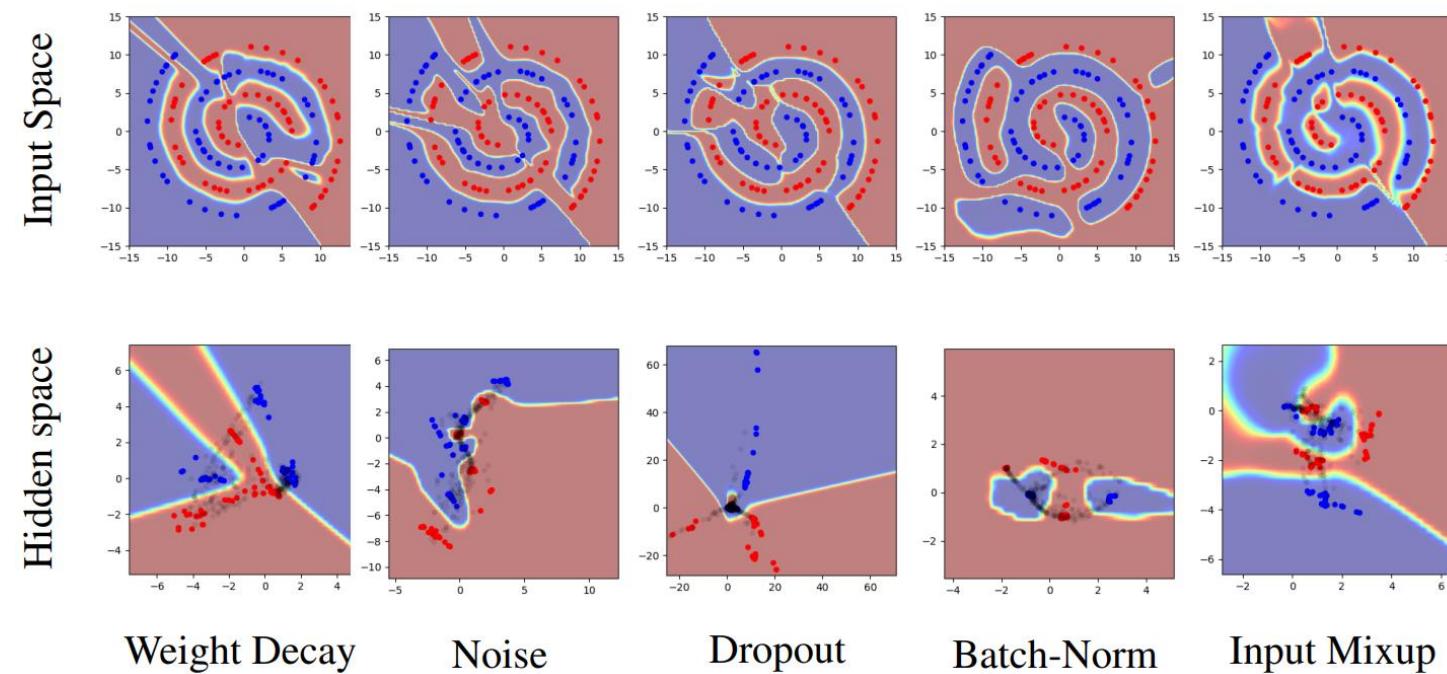
- 1.1 Static Linear
- 1.2 Feature-based
- 1.3 Cutting-based
- 1.4 k Sample Mixup
- 1.5 Random Policies
- 1.6 Style-based
- 1.7 Saliency-based
- 1.8 Attention-based
- 1.9 Generating Samples

1.2 Feature-based

Manifold Mixup: Better Representations by Interpolating Hidden States (ICML`2019)

Manifold Mixup将mixup从input layer转到了embedding layer，也就是所谓的隐空间中。同样的，在得到两个样本的特征表示后，按照比例 λ 进行简单的线性插值得到混合特征表示。

值得一提的是，Manifold Mixup中首次验证了mixup方法对神经网络calibration的影响，为后续工作提供了一个新的实验分析验证。

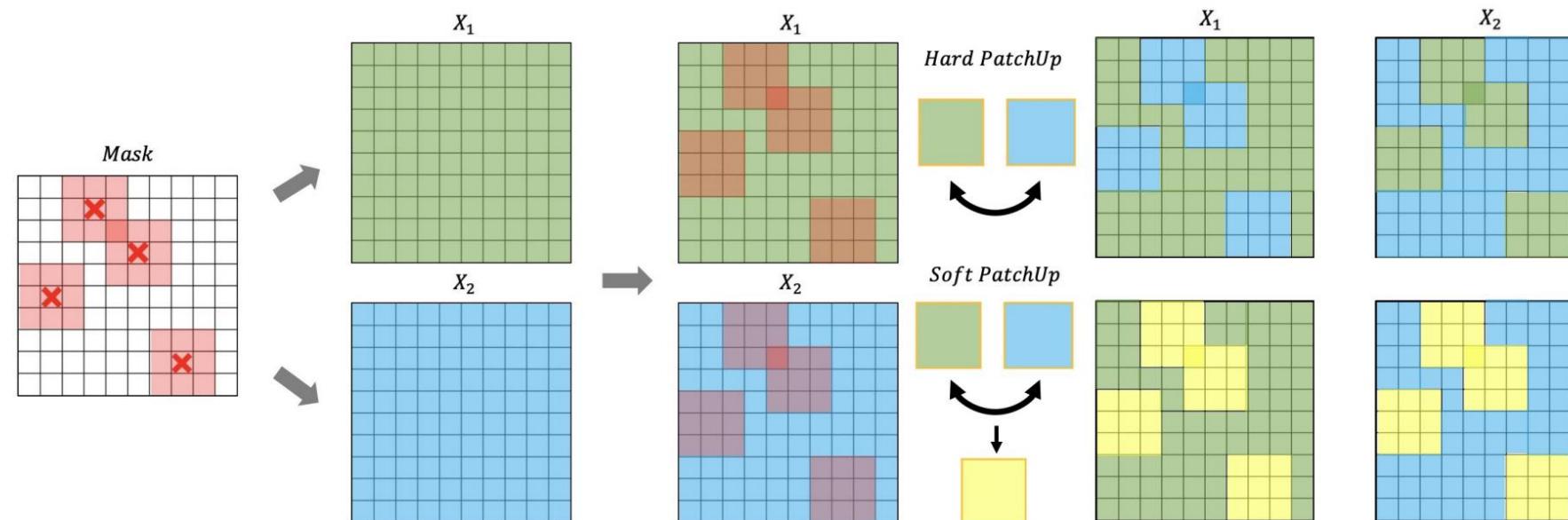


1.2 Feature-based

PatchUp: A Feature-Space Block-Level Regularization Technique for Convolutional Neural Networks
(AAAI'2022)

PatchUp采用隐空间的CutMix方法，按照比例和大小选择合适的patch个数，再按照CutMix的方法进行混合得到隐空间的混合表示。

右下表示Soft PatchUp，即在混合的patch按照线性插值得到soft patch representation。

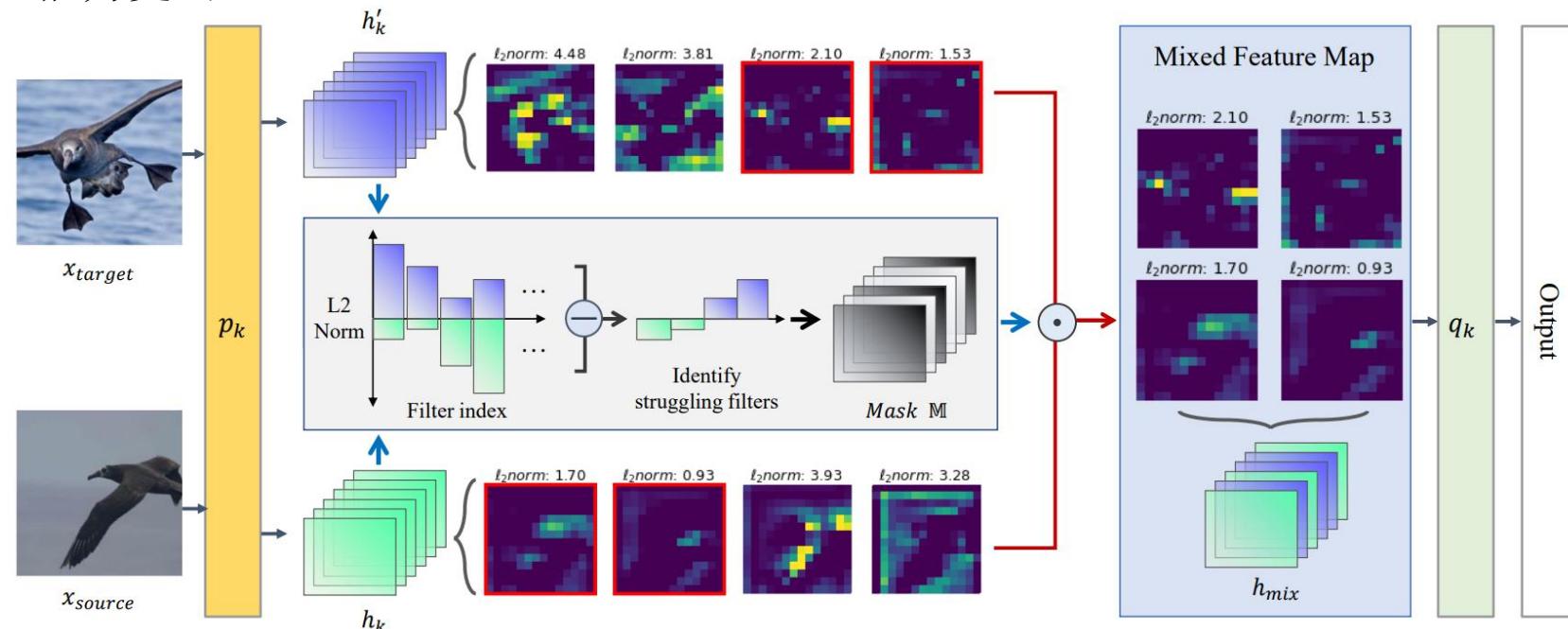


1.2 Feature-based

Catch-up Mix: Catch-up Class for Struggling Filters in CNN (AAAI'2024)

Catch-up Mix通过对样本feature map的 $l_2 norm$ 进行筛选，按照 λ 选择channel数量。通过排序后，选择low $l_2 norm$ 的feature map混合得到增强表示。

作者认为在CNNs中，网络中一些表达力强的卷积核会使网络倾向于它们，反而放弃了表达力弱的卷积核，弱卷积核能力被限制，这很容易产生过拟合，那么作者采用Catch-up Mix来进一步挖掘弱卷积核，使CNNs的表达能力更出色。



- 1.1 Static Linear
- 1.2 Feature-based
- 1.3 Cutting-based
- 1.4 k Sample Mixup
- 1.5 Random Policies
- 1.6 Style-based
- 1.7 Saliency-based
- 1.8 Attention-based
- 1.9 Generating Samples

1.3 Cutting-based

CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features (ICCV`2019)

CutMix将mixup从pixel level转到了region level。通过一个随机采样的 λ 转换成样本的面积，通过随机裁切源样本中 λ 大小的面积区域，再随机替换到目标样本中得到混合样本。

至此，CutMix开启了一系列通过patch来进行混合的方法，且通过patch占比的大小来决定标签混合时的比例 λ 。

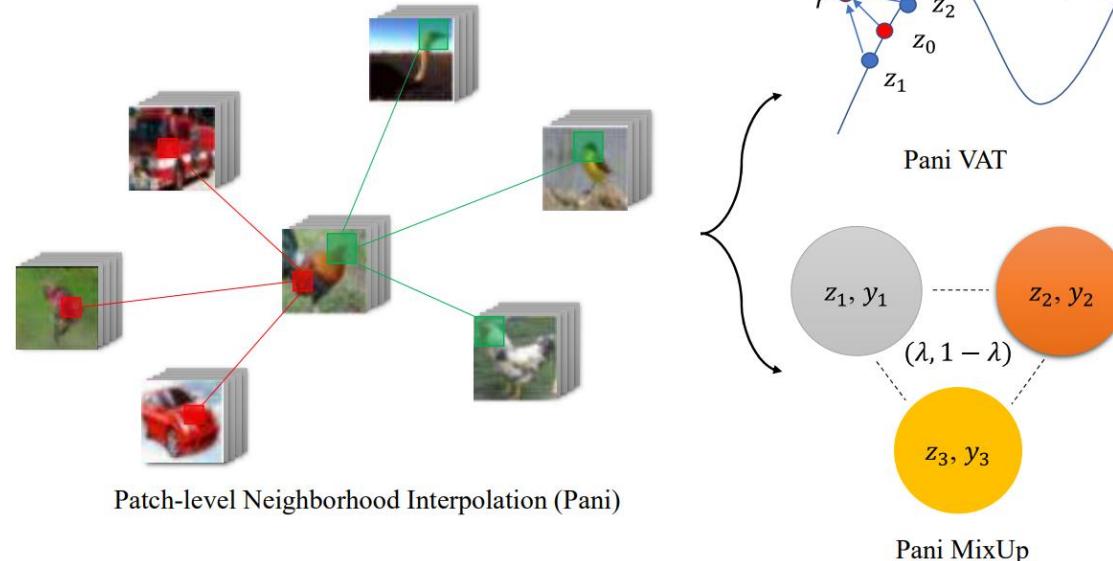
Image	ResNet-50	Mixup [48]	Cutout [3]	CutMix
				

1.3 Cutting-based

Patch-level Neighborhood Interpolation: A General and Effective Graph-based Regularization Strategy (ACML`2023)

Pani MixUp通过一个GNN来建立样本patch之间的关系进行插值混合增强。具体来说：

1. 通过随机匹配或者距离相似性计算得到一个集合S；
2. 经过GNN得到样本的representation后，构造一个patch集合P；
3. 利用patch的相似性度量，对P构造K近邻集合；
4. 对相似patch进行线性插值得到混合的patch；
5. 得到混合的patch后对网络进行训练。



1.3 Cutting-based

FMix: Enhancing Mixed Sample Data Augmentation (arXiv`2020)

FMix通过将样本映射到频域中进行混合操作。作者使用傅里叶变换将RGB信息转换成频域信息，通过傅里叶空间中采样低频gray mask，然后根据阈值将其转换为binary mask。相较于CutMix，作者认为FMix可以生成各种样式的mask，防止网络对正方形的mask具有过分的“记忆”。

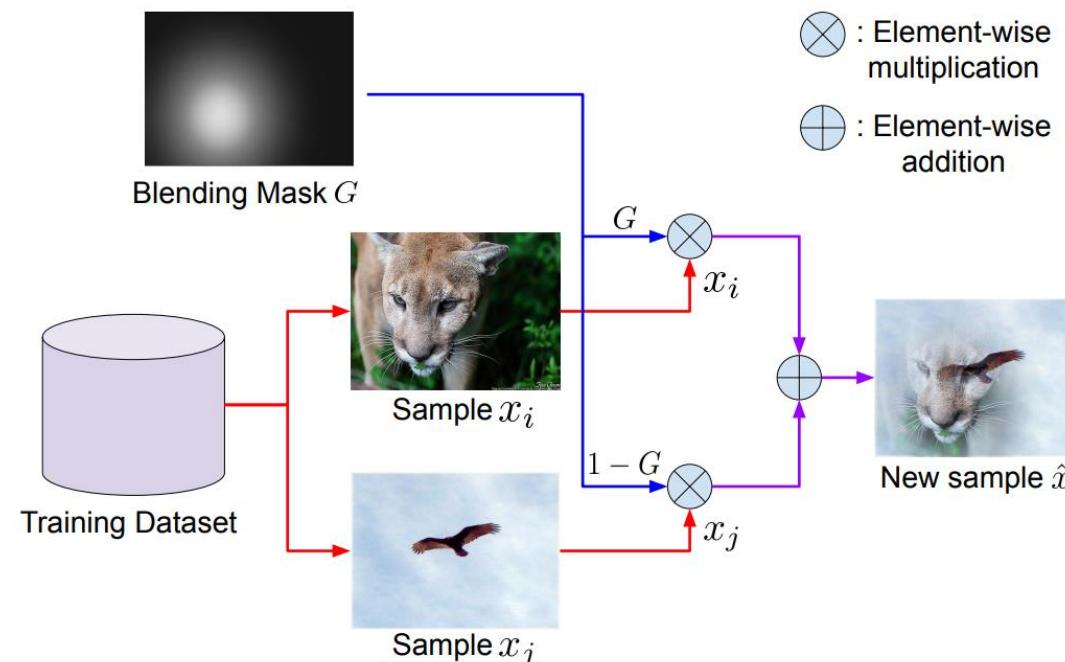


1.3 Cutting-based

SmoothMix: a Simple Yet Effective Data Augmentation to Train Robust Classifiers(CVPRW`2020)

CutMix的Smooth version。

通过一个设定的高斯核，将混合比例 λ ，kernel size, sample size输入得到一个平滑过渡的mask对样本进行混合得到增强样本。

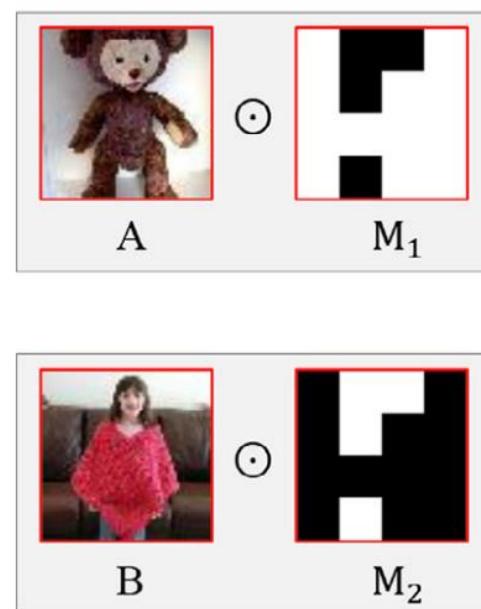


1.3 Cutting-based

GridMix: Strong regularization through local context mapping (PR`2021)

GridMask + CutMix的结合。

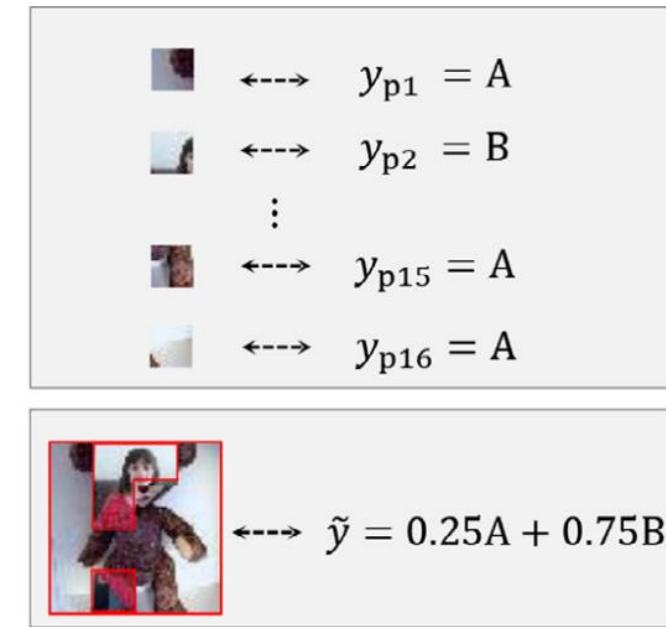
对于损失函数，作者使用local和global两种组成， local loss是将各个patch和其标签进行计算， global loss即MCE。



$$L_{local} = \sum_{i=1}^{16} \mathbb{E}[H(f_l(\tilde{x})_i, y_{pi})]$$



$$L_{global} = \mathbb{E}[H(f_g(\tilde{x}), \tilde{y})]$$

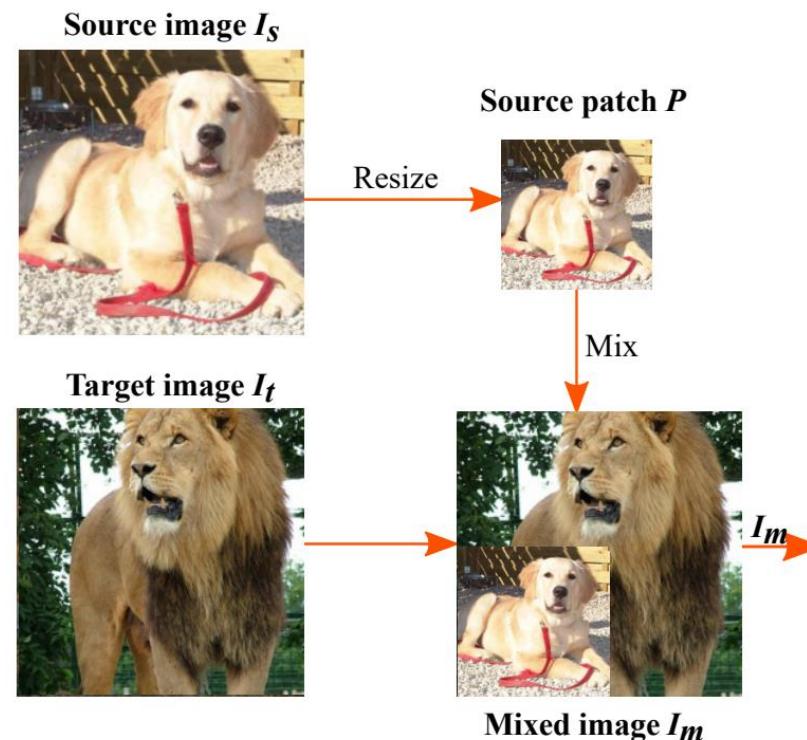


1.3 Cutting-based

ResizeMix: Mixing Data with Preserved Object Information and True Labels (CVMJ`2023)

Resize + CutMix。

通过将样本resize后进行CutMix得到混合样本，相较于以往的Cutting-based方法， ResizeMix可以将源样本所有的特征信息得到保留。对于混合样本而言，特征信息能够尽可能的变多。

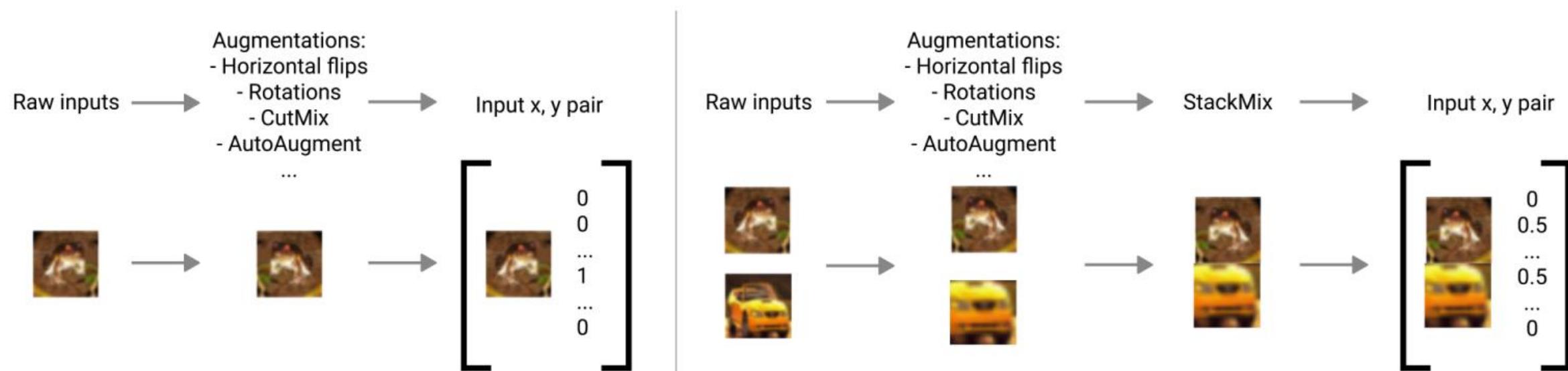


1.3 Cutting-based

StackMix: A complementary Mix algorithm (UAI'2022)

Resize + Stack Sample.

对Raw samples经过初步的增强后再直接上下/左右拼接得到混合样本，标签按照0.5比例。简单有效，在ImageNet-1K中涨点蛮多。



1.3 Cutting-based

SuperpixelGridCut, SuperpixelGridMean and SuperpixelGridMix Data Augmentation (arXiv`2022)

使用了经典的分割方法Superpixel提出了3中增强方法: SuperpixelGridCut, SuperpixelGridMean and SuperpixelGridMix。

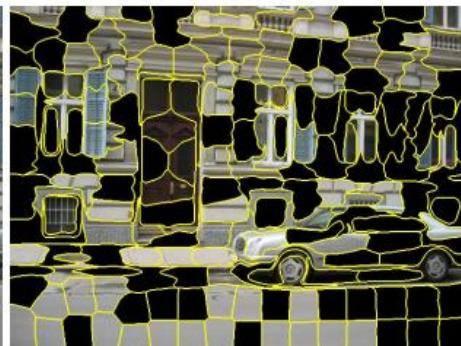
SuperpixelGridCut: 通过随机cutout多个patch得到增强样本;

SuperpixelGridMean: 通过将丢弃的patch进行均值填充信息得到增强样本;

SuperpixelGridMix: 通过对两个样本进行superpixel后，随机选取patch进行替换混合得到增强样本。



(a) Image 1



(b) SuperpixelGridCut



(c) SuperpixelGridMean



(d) Image 2



(e) SuperpixelGridMix

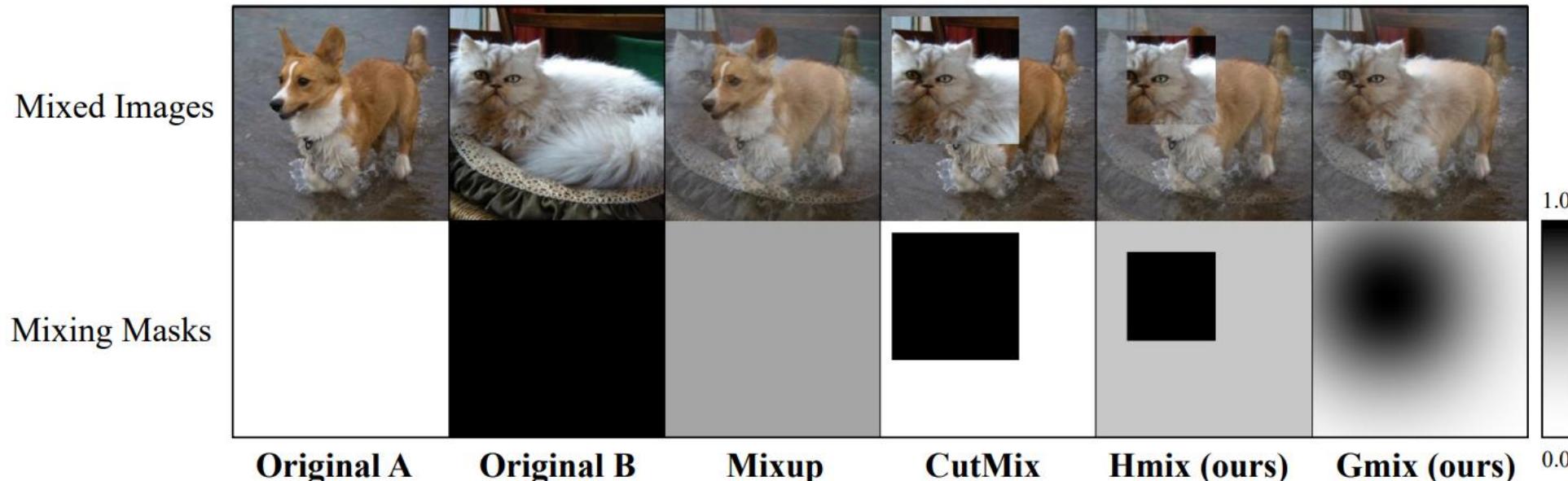
1.3 Cutting-based

A Unified Analysis of Mixed Sample Data Augmentation: A Loss Function Perspective (NIPS`2022)

对MixUp和CutMix进行组合两种新的混合方法——Hmix, Gmix。

$$\text{Hmix} = \text{CutMix} + \text{MixUp}$$

$$\text{Gmix} = \text{SmoothMix}$$

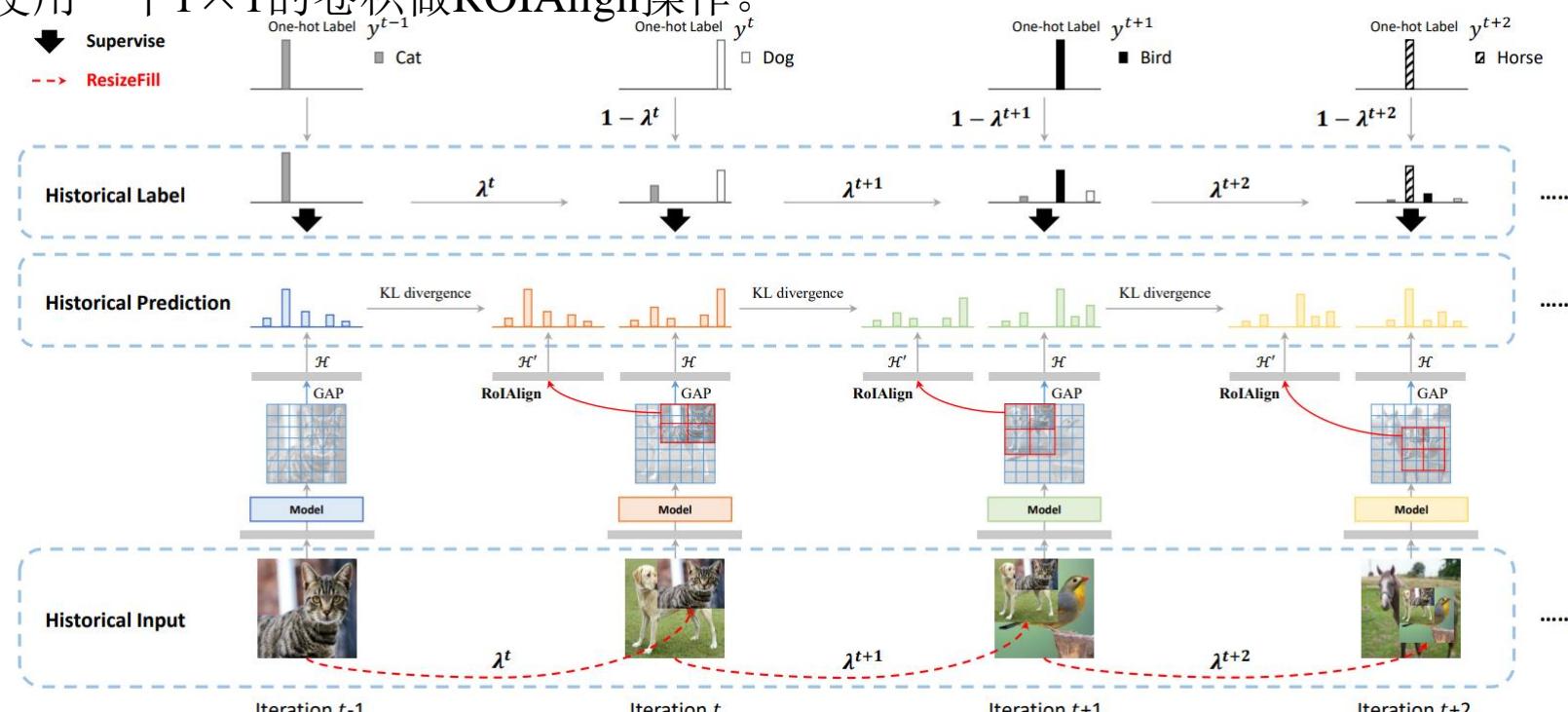


1.3 Cutting-based

RecursiveMix: Mixed Learning with History (NIPS`2022)

通过多次ResizeMix样本获得混合样本。注意的是，在前一次混合的样本不丢弃，反而作为新的源样本与目标样本进行混合。标签层面也是多次进行记录-迭代得到混合标签。

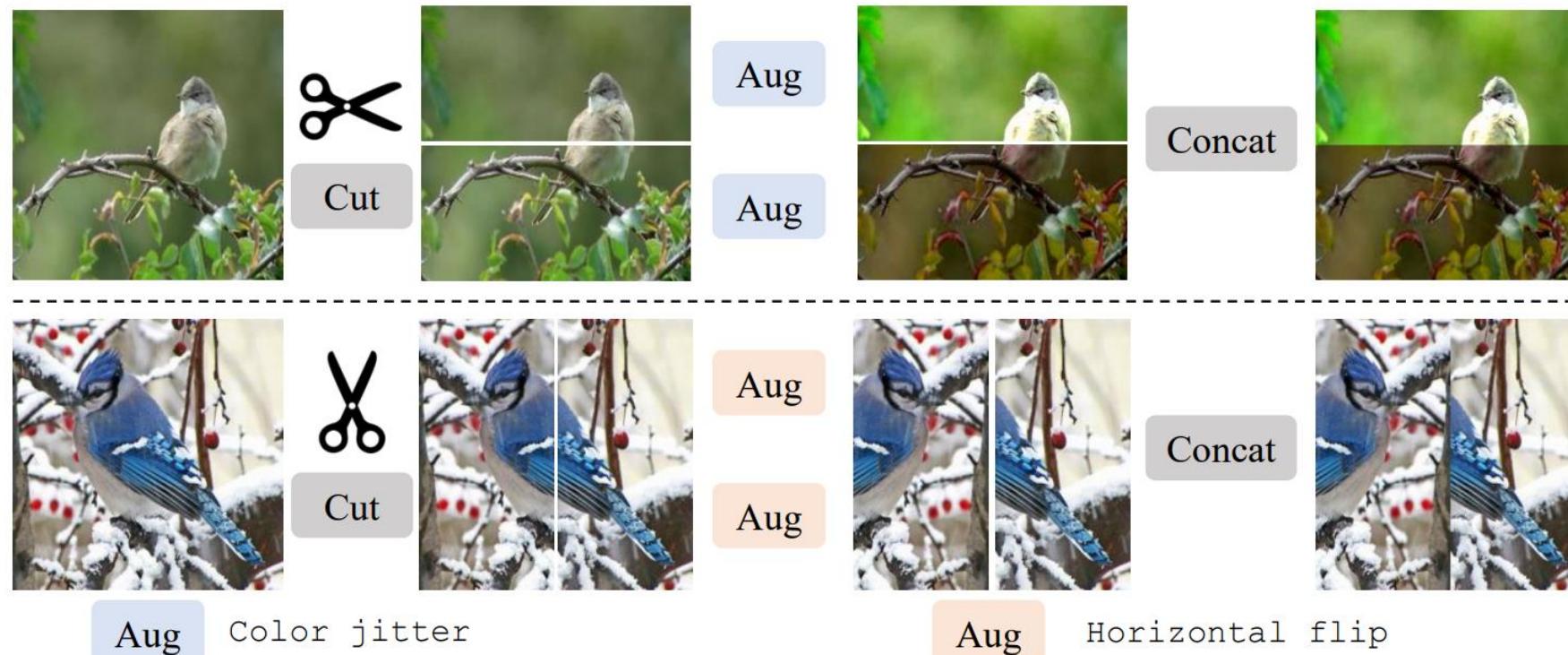
在损失函数方面，受对比学习的启发，作者使用上一次迭代得到的预测与本次的预测进行KL散度计算，本次预测使用一个 1×1 的卷积做ROIAlign操作。



1.3 Cutting-based

You Only Cut Once: Boosting Data Augmentation with a Single Cut (ICML`2022)

通过对样本进行随机选择横/竖裁切，再将得到的patch经过不同的随机增强手段后，进行拼接得到最终的混合增强样本。



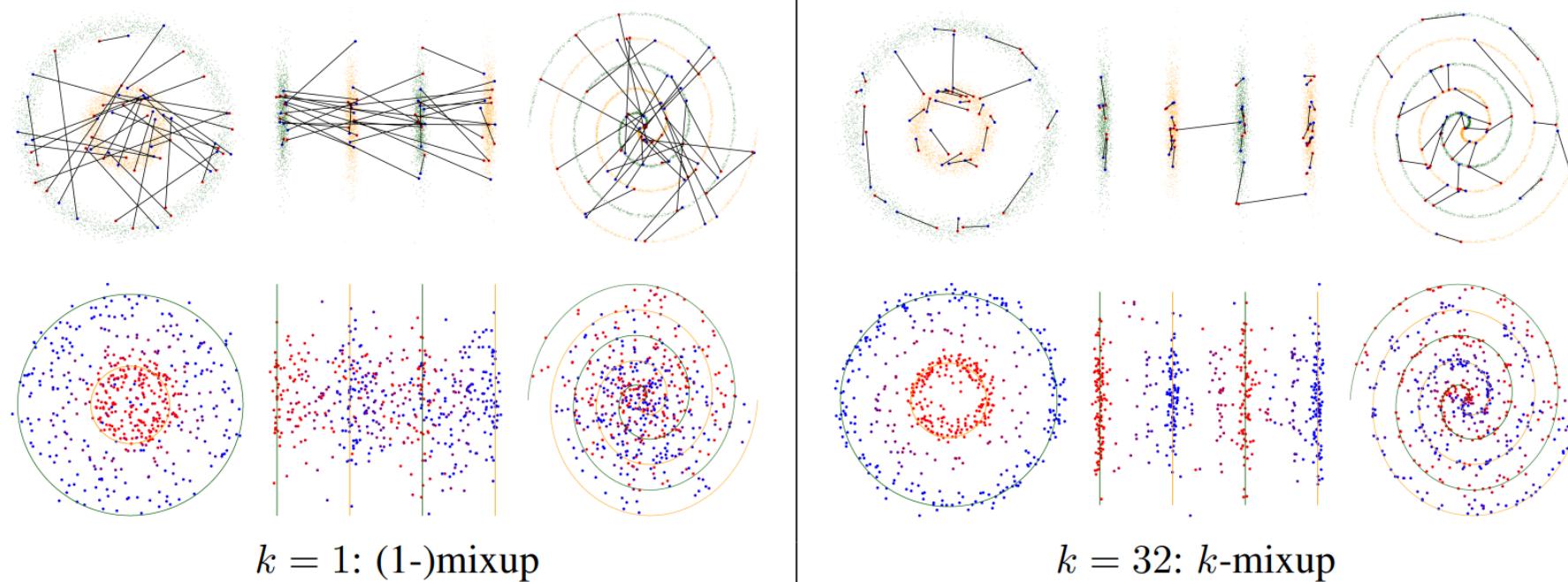
- 1.1 Static Linear
- 1.2 Feature-based
- 1.3 Cutting-based
- 1.4 k Sample Mixup
- 1.5 Random Policies
- 1.6 Style-based
- 1.7 Saliency-based
- 1.8 Attention-based
- 1.9 Generating Samples

1.4 k Sample Mixup

* k -Mixup Regularization for Deep Learning via Optimal Transport (ICML`2019)

作者通过在 Wasserstein 度量进行样本下插值，在其它 k 个样本方向上扰动 k 个实例，从而产生更多的增强数据。

(最优传输：一个分布→另一个分布的代价)

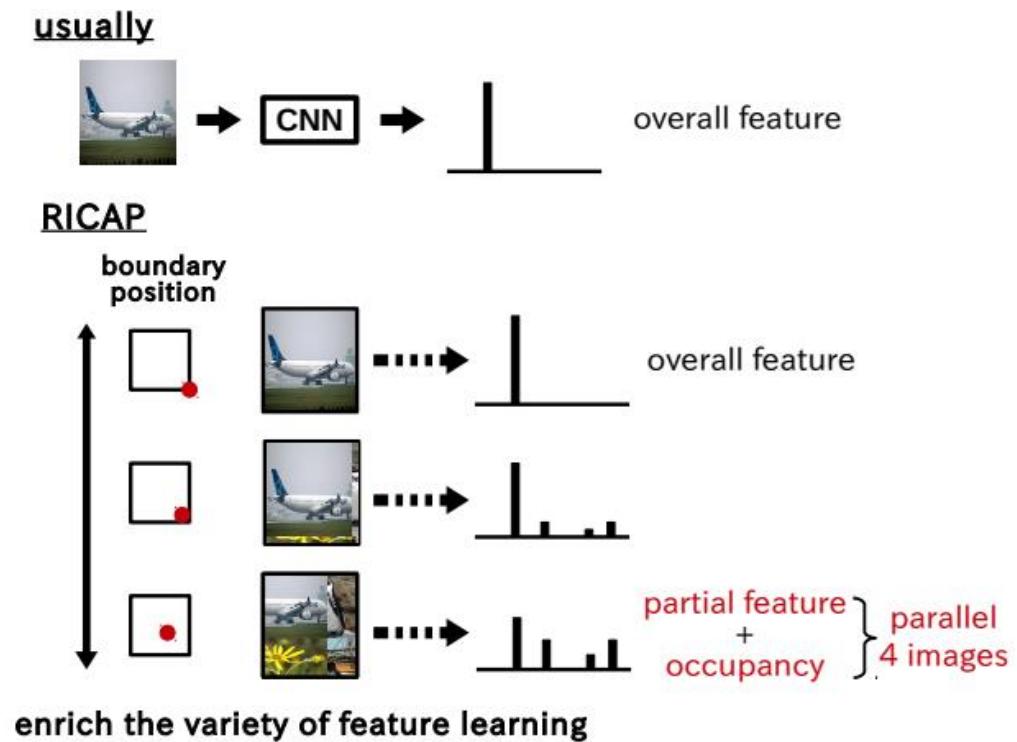
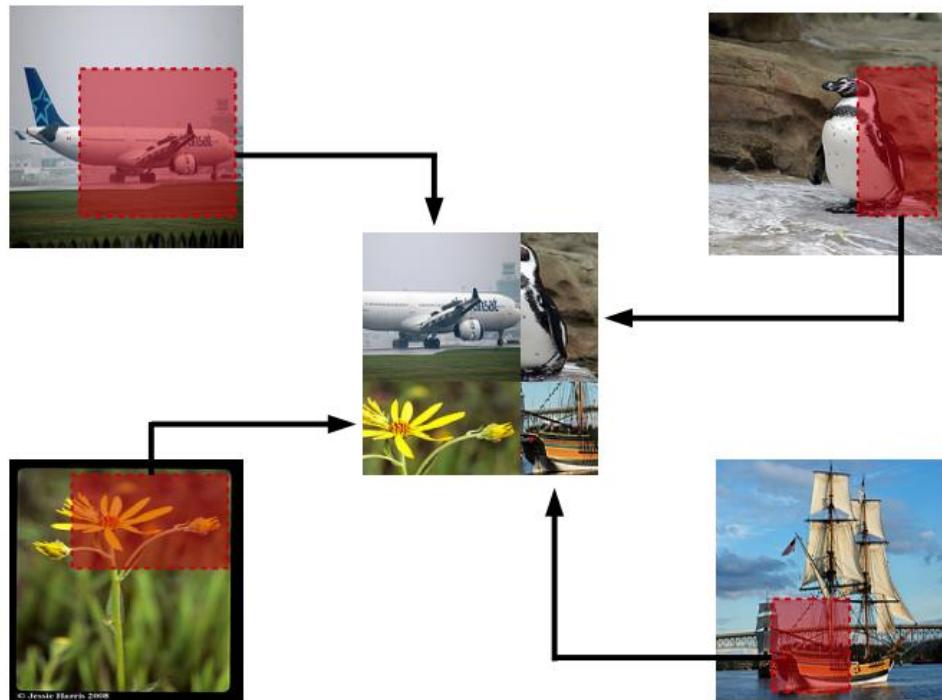


1.4 k Sample Mixup

Data Augmentation using Random Image Cropping and Patching for Deep CNNs (TCSVT`2020)

RICAP通过随机选择4个样本中的patch，进行拼接得到混合增强样本。

其中，再YOLO系列中一个很常见的增强手段Mosaic和RICAP类似，不过Mosaic是将4个样本直接拼接得到。



1.4 k Sample Mixup

DCutMix: Observations on K-image Expansion of Image-Mixing Augmentation for Classification (IEEE Access`2021)

DCutMix通过对多个样本进行CutMix得到增强样本。

作者通过实验分析了k mixup更好泛化背后的机制，分析发现多个样本混合的Landscape，可以驱动模型进入最宽(最平坦)和最深的局部最小值。

同时，作者将DCutMix应用在了NAS中，也得到了增益。

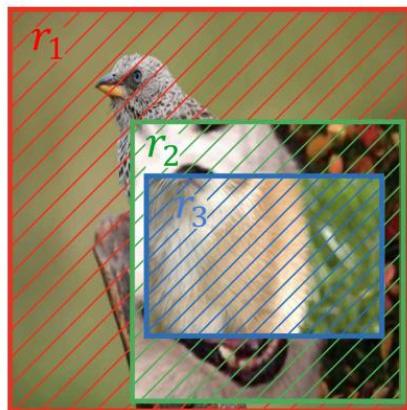


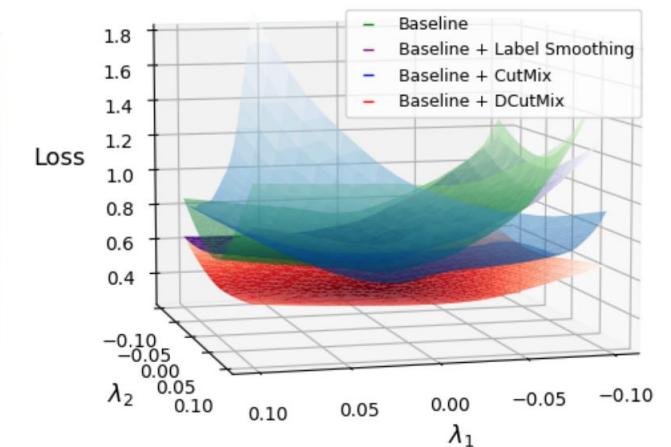
Image 1



Image 2



Image 3



1.4 k Sample Mixup

Cut-Thumbnail: A Novel Data Augmentation for Convolutional Neural Network (ACM MM`2022)

Cut-Thumbnail通过随机多次ResizeMix得到增强样本。



(a) Original Sample



(b) Self Thumbnail



(c) Mixed Single Thumbnail



(d) Mixed Multiple Thumbnails



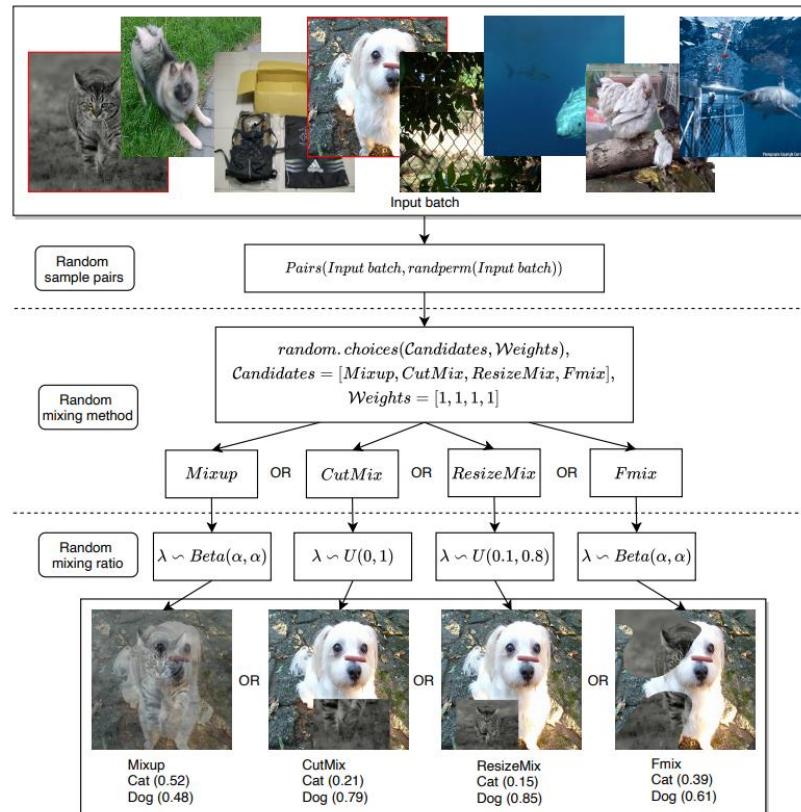
(e) Other Strategy

- 1.1 Static Linear
- 1.2 Feature-based
- 1.3 Cutting-based
- 1.4 k Sample Mixup
- 1.5 Random Policies
- 1.6 Style-based
- 1.7 Saliency-based
- 1.8 Attention-based
- 1.9 Generating Samples

1.5 Random Policies

RandomMix: A mixed sample data augmentation method with multiple mixed modes
(Multimedia Tools and Applications`2024)

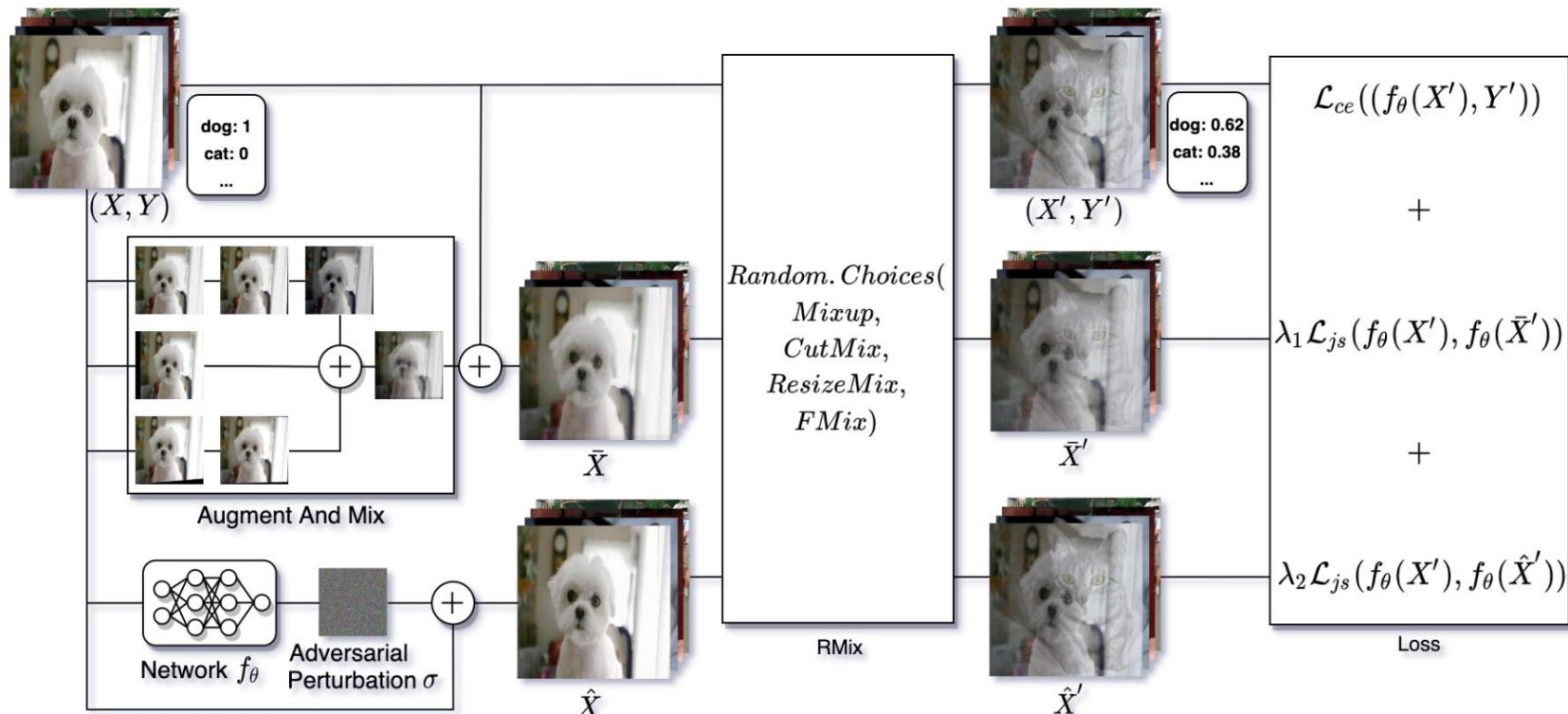
通过简单的对多种mixup方法进行组合，按照设定的权重进行随机选择。



1.5 Random Policies

AugRmixAT: A Data Processing and Training Method for Improving Multiple Robustness and Generalization Performance (ICME`2022)

通过多种增强手段(*e.g.* Augment, AugMix, Adv. per)得到多种增强样本后，再随机选择mixup方法得到最终增强样本进行训练。



- 1.1 Static Linear
- 1.2 Feature-based
- 1.3 Cutting-based
- 1.4 k Sample Mixup
- 1.5 Random Policies
- 1.6 Style-based
- 1.7 Saliency-based
- 1.8 Attention-based
- 1.9 Generating Samples

1.6 Style-based

StyleMix: Separating Content and Style for Enhanced Data Augmentation (CVPR`2021)

作者认为现有的mixup方法无法区分图像的Content和Style特征，故而提出StyleMix、StyleCutMix分别处理输入图像对的Content和Style information。

通过线性插值(StyleMix)或者裁切patch(StyleCutMix)的方法使用AdaIN(\cdot)分别提取图像内容和风格创建混合样本，增强神经网络的泛化能力。

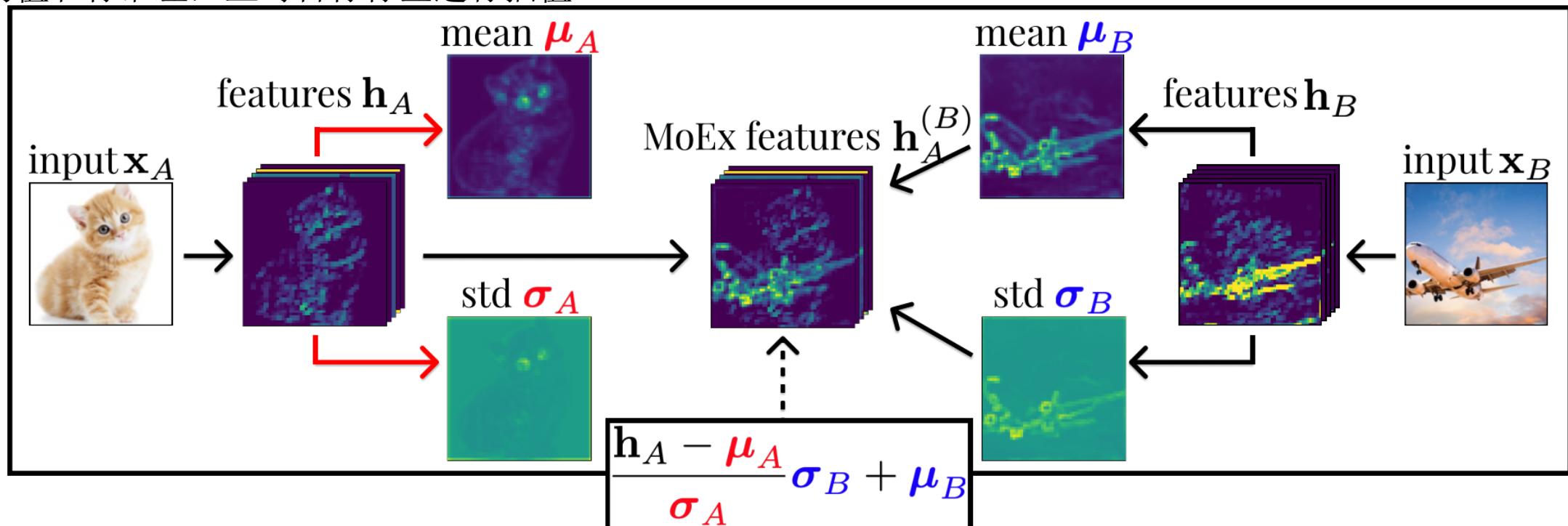
	Input 1		Input 2	
Method	Mixup	StyleMix	CutMix	StyleCutMix
Content label	Parrot 0.5 Panda 0.5	Parrot 0.4 Panda 0.6	Parrot 0.2 Panda 0.8	Parrot 0.2 Panda 0.8
Style label	X	Parrot 0.8 Panda 0.2	X	Parrot 0.6 Panda 0.4

1.6 Style-based

On Feature Normalization and Data Augmentation (CVPR`2021)

作者提出了一种隐式数据增强方法Moment Exchange (MoE)，将样本feature maps的平均值和标准差作为样本风格的特征而不是作为噪声。

方法是将一幅样本经过编码器后得到的feature maps的均值和标准差替换为另一幅样本feature maps的均值和标准差，且对目标标签进行插值。



1.6 Style-based

Domain Generalization with MixStyle (ICLR`2021)

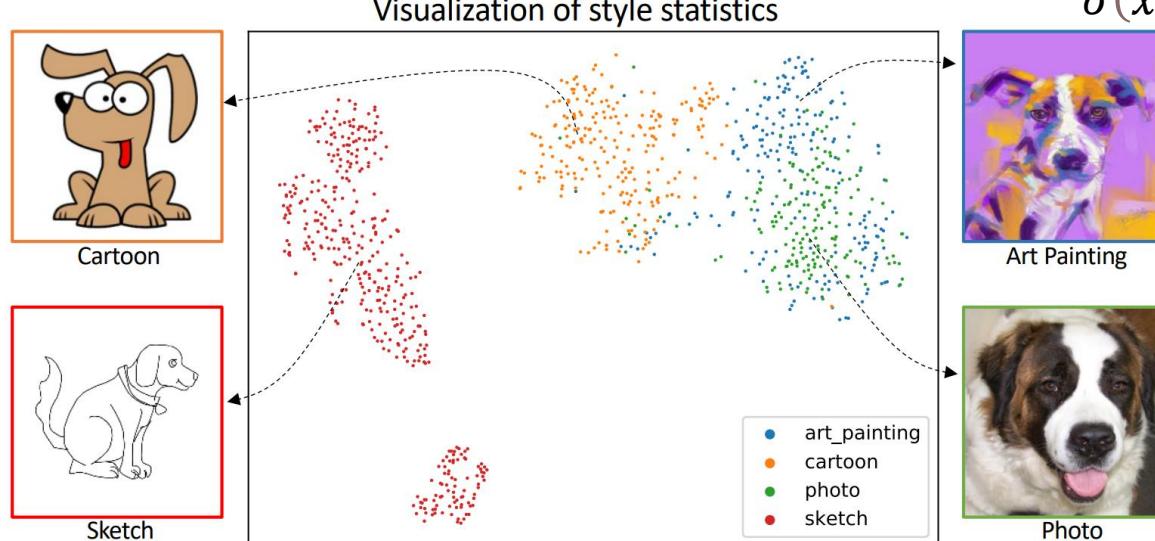
作者提出了一种简单而有效的领域泛化方法MixStyle。

MixStyle混合了两个实例的特征统计来合成新的领域，其灵感来自于在风格迁移工作中观察到的特征统计编码样式和领域相关的信息，是Style Transfer和mixup的巧妙结合。

$$\gamma_{mix} = \lambda\sigma(x) + (1 - \lambda)\sigma(\tilde{x})$$

$$\beta_{mix} = \lambda\mu(x) + (1 - \lambda)\mu(\tilde{x})$$

$$MixStyle(x) = \gamma_{mix} \frac{x - \mu(x)}{\sigma(x)} + \beta_{mix}$$



$$x = [(x_1 \text{ } x_2 \text{ } x_3 \text{ } x_4 \text{ } x_5 \text{ } x_6)]$$

$$\tilde{x} = [(x_5 \text{ } x_6 \text{ } x_4 \text{ } x_3 \text{ } x_1 \text{ } x_2)]$$

(a) Shuffling batch w/ domain label

$$x = [(x_1 \text{ } x_2 \text{ } x_3 \text{ } x_4 \text{ } x_5 \text{ } x_6)]$$

$$\tilde{x} = [(x_6 \text{ } x_1 \text{ } x_5 \text{ } x_3 \text{ } x_2 \text{ } x_4)]$$

(b) Shuffling batch w/ random shuffle

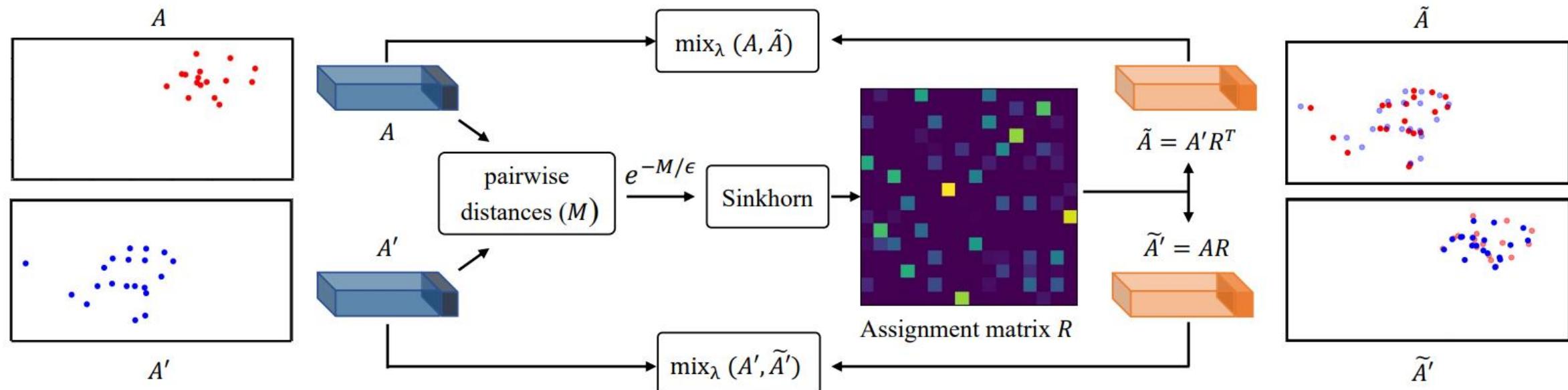
1.6 Style-based

AlignMixup: Improving Representations By Interpolating Aligned Features (CVPR`2022)

作者提出该方法的动机是mixup的想法作为一种自然的方式插值图像，其中一个图像可以在一个连续的方式下变形到另一个。

主要利用最优传输理论(文中使用Sinkhorn方法)来进行图像/特征对齐，然后将对齐后的图像/特征进行混合得到增强样本进行训练。

文中使用了一个Decoder来可视化混合样本，发现AlignMixup可以很好的保留一个样本的姿态和另一个样本的内容。



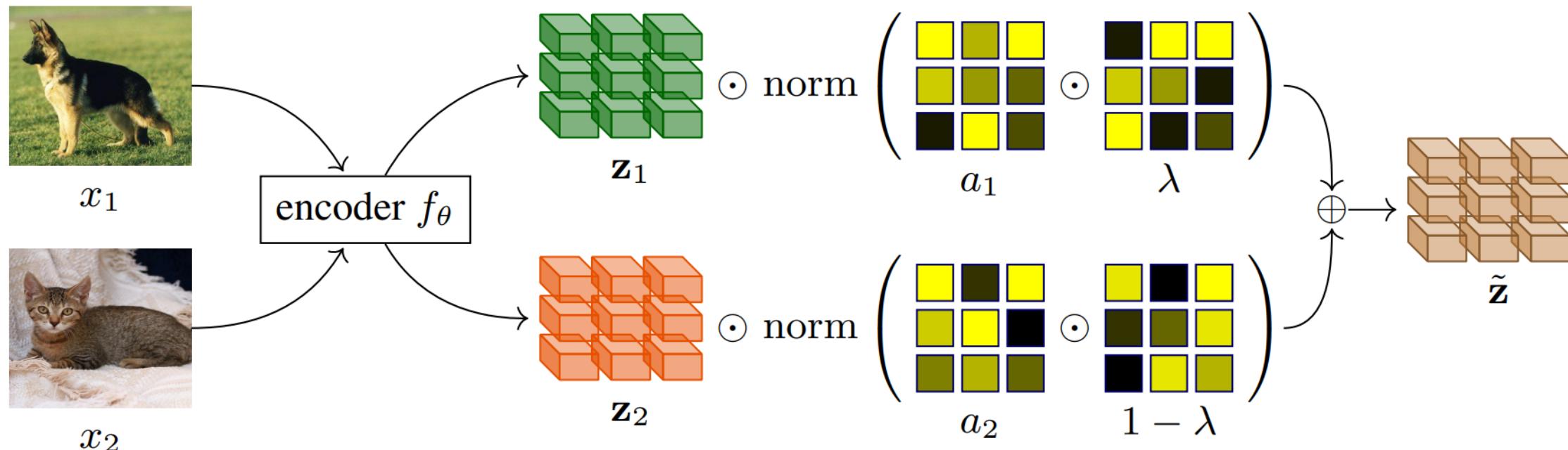
1.6 Style-based

Embedding Space Interpolation Beyond Mini-Batch, Beyond Pairs and Beyond Examples (NIPS`2023)

MultiMix也是通过在embedding layer进行混合得到增强的feature maps。

具体来说，将Raw samples经过Encoder后得到feature maps，再与normalization的attention* λ 的feature maps相乘得到样本最终的feature maps，该feature maps能够最大化的保留样本在embedding layer的特性信息。

Attention有俩种方法得到：1. z经过GAP(global average pooling)计算；2. z经过CAM后得到。

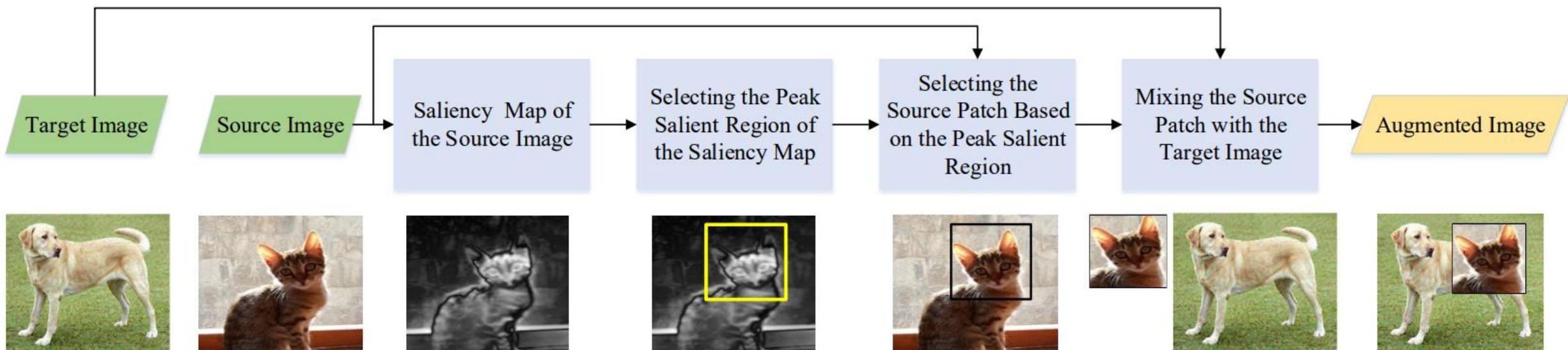


- 1.1 Static Linear
- 1.2 Feature-based
- 1.3 Cutting-based
- 1.4 k Sample Mixup
- 1.5 Random Policies
- 1.6 Style-based
- 1.7 Saliency-based
- 1.8 Attention-based
- 1.9 Generating Samples

1.7 Saliency-based

SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization (ICLR`2021)

SaliencyMix通过一个额外的显著性提取器来对Raw samples进行特征提取，并选择显著性信息区域最大的patch，随机粘贴到目标样本中得到混合样本。

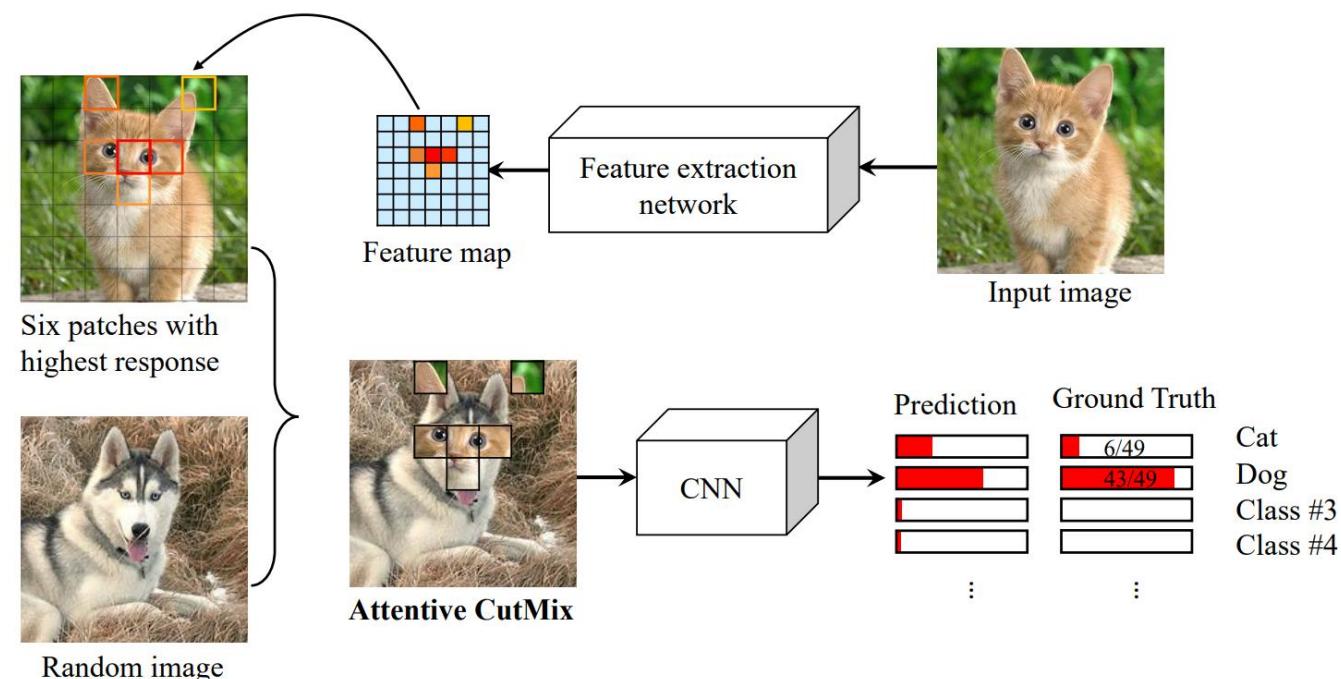


1.7 Saliency-based

Attentive CutMix: An Enhanced Data Augmentation Approach for Deep Learning Based Image Classification (ICASSP`2022)

Attentive-CutMix使用一个Pre-train Model来提取样本中的heat-map (*e.g.* 7×7), 从 7×7 网格中选择Top-N个Patch作为Attention Tokens, 从源样本中剪切出来, 粘贴到目标样本上的相应位置。

对于标签混合, 计算为选择样本的patch个数/总patch个数。

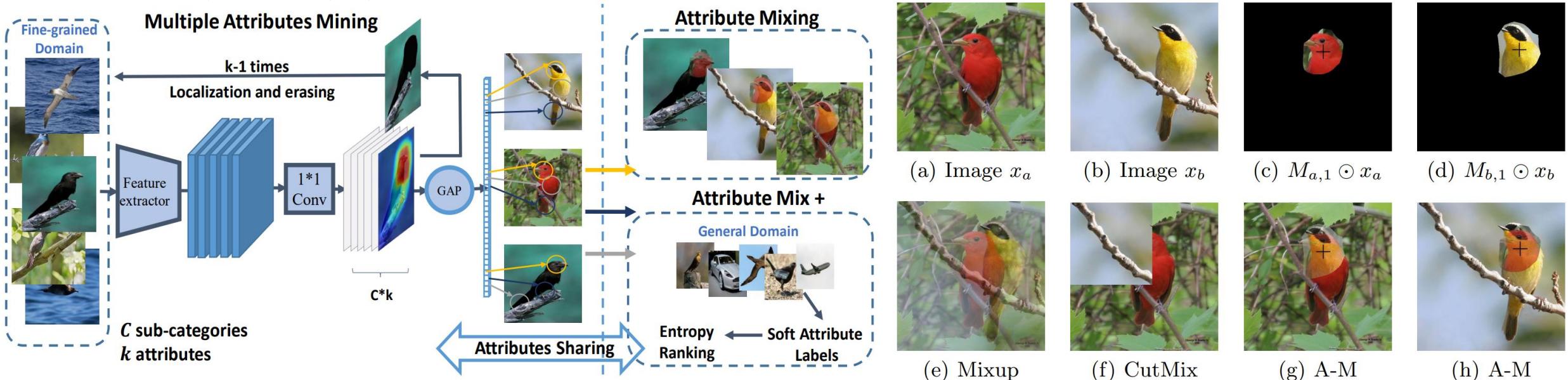


1.7 Saliency-based

Attribute Mix: Semantic Data Augmentation for Fine-grained Recognition (VCIP`2020)

Attribute Mix通过对细粒度数据集进行一个字典K的构造(作者可能认为对于细粒度这类数据集而言, Encoder对于样本所关注的重点是一致的)。样本经过Encoder后再经过一个 1×1 的卷积得到saliency maps, 通过定位生成mask后擦除该saliency info, 经过K-1次循环后, 得到 $C \times K$ 个通道, 每个样本都包含了K个属性的saliency map。

对于普通分类任务而言, 做Multi-hot标签分类; 对于mixup方法, 对K个字典的通道进行选取生成mask混合得到混合样本。

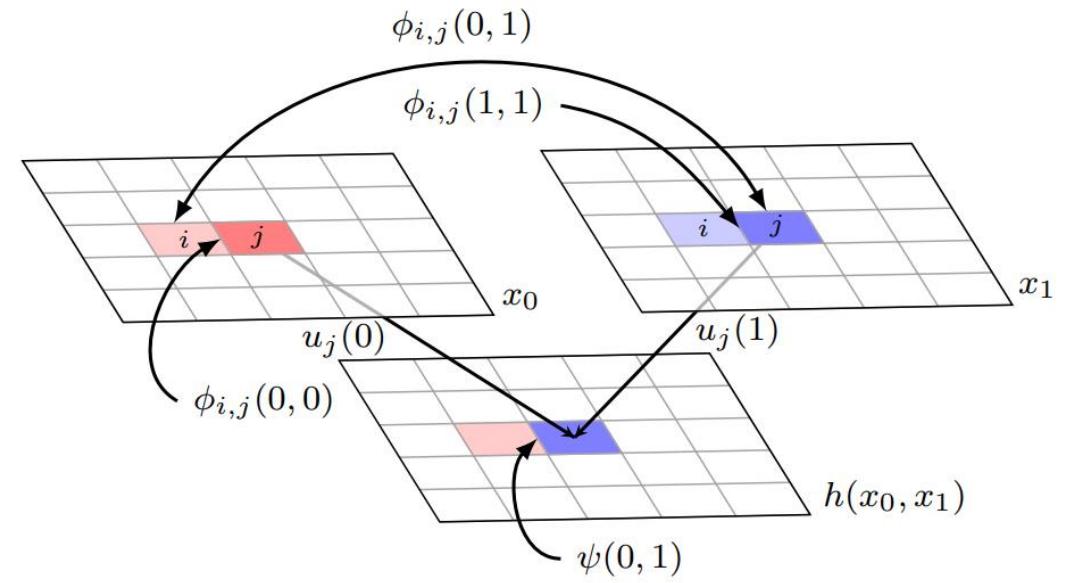
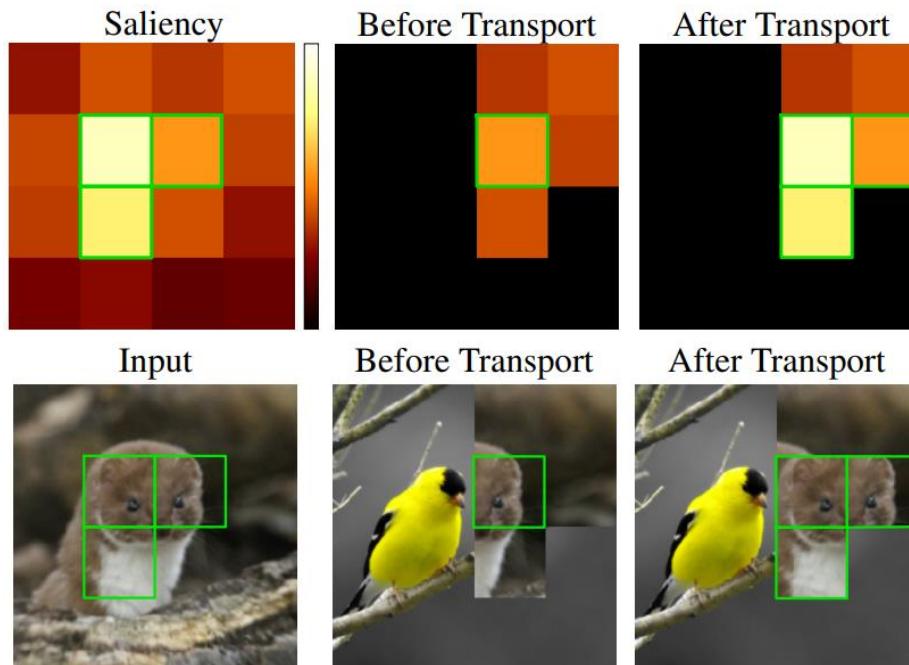


1.7 Saliency-based

Puzzle Mix: Exploiting Saliency and Local Statistics for Optimal Mixup (ICML`2020)

PuzzleMix通过对Raw样本编码分类得到梯度后进行反传得到样本的saliency map，将样本划分成tokens后生成mask进行混合。

值得注意的是，PuzzleMix考虑到了两幅样本的显著性信息在相同区域时如何最大化问题。作者采用了一个最优传输的方法，通过设定一个“Optimizing Transport”来进行特征区域的移动，以达到混合样本的显著性信息最大化。

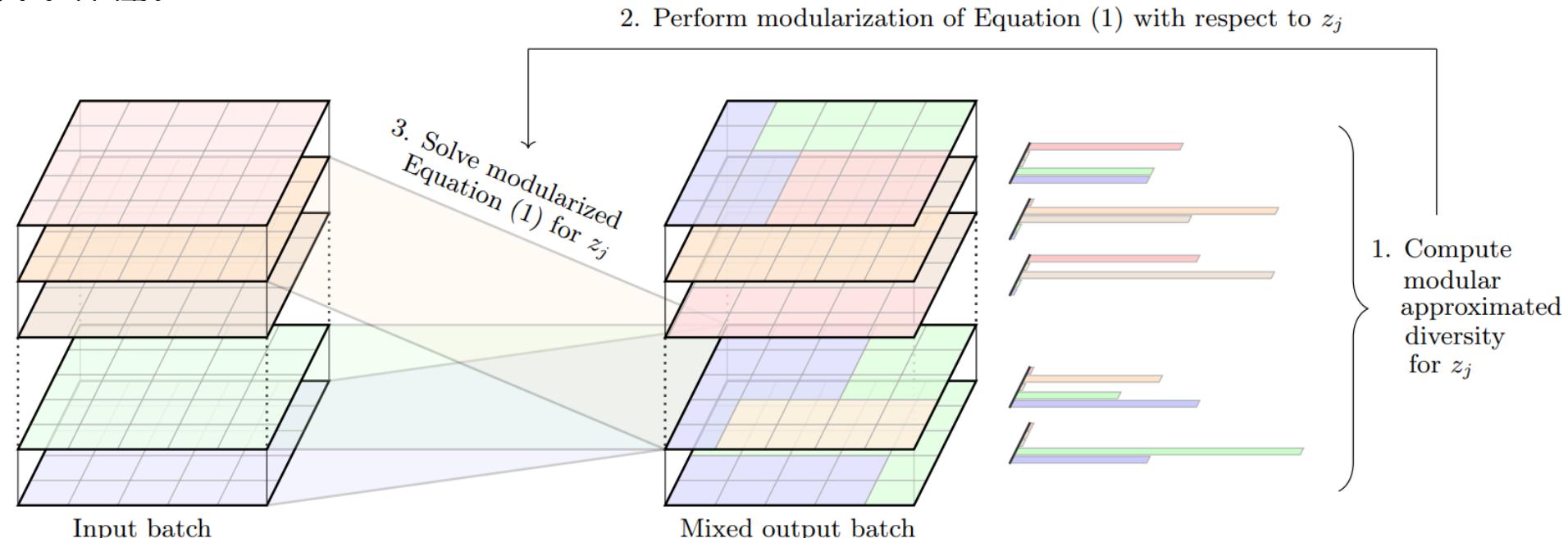


1.7 Saliency-based

Co-Mixup: Saliency Guided Joint Mixup with Supermodular Diversity (ICLR`2021)

Co-Mixup将随机选择进行混合的样本从Pair上升到Mini-Batch，并且通过多个mix目标将寻找最优的mixup方法转换成一个优化问题，以此来指导mixup过程。

优化目标包括：最大化mixup后的图像显著性(PuzzleMix)、mixup后图像尽可能满足局部平滑、为了提升产生图像的多样性，引入了一个正则化损失对生成过于相似的图像进行惩罚，使得mixup得到的增强图像更具多样性。

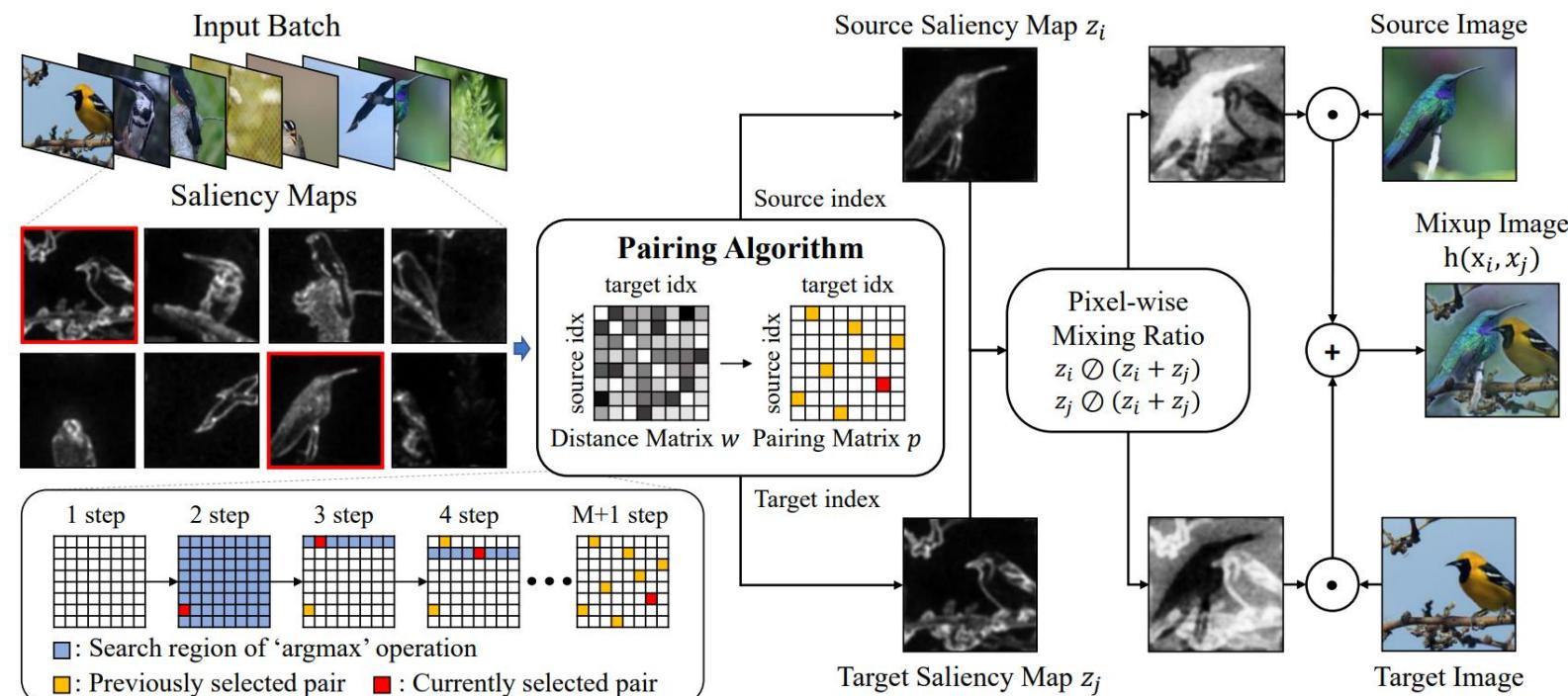


1.7 Saliency-based

GuidedMixup: An Efficient Mixup Strategy Guided by Saliency Maps (AAAI'2023)

通过神经网络来提取样本的saliency maps，然后使用“贪婪算法”选择Mini-batch中saliency map距离很远的样本作为集合；

此外，将token level转到了pixel level，最大限度地利用每个像素的显著性信息，并逐像素调整混合比例，更精细化，混合更加平滑。



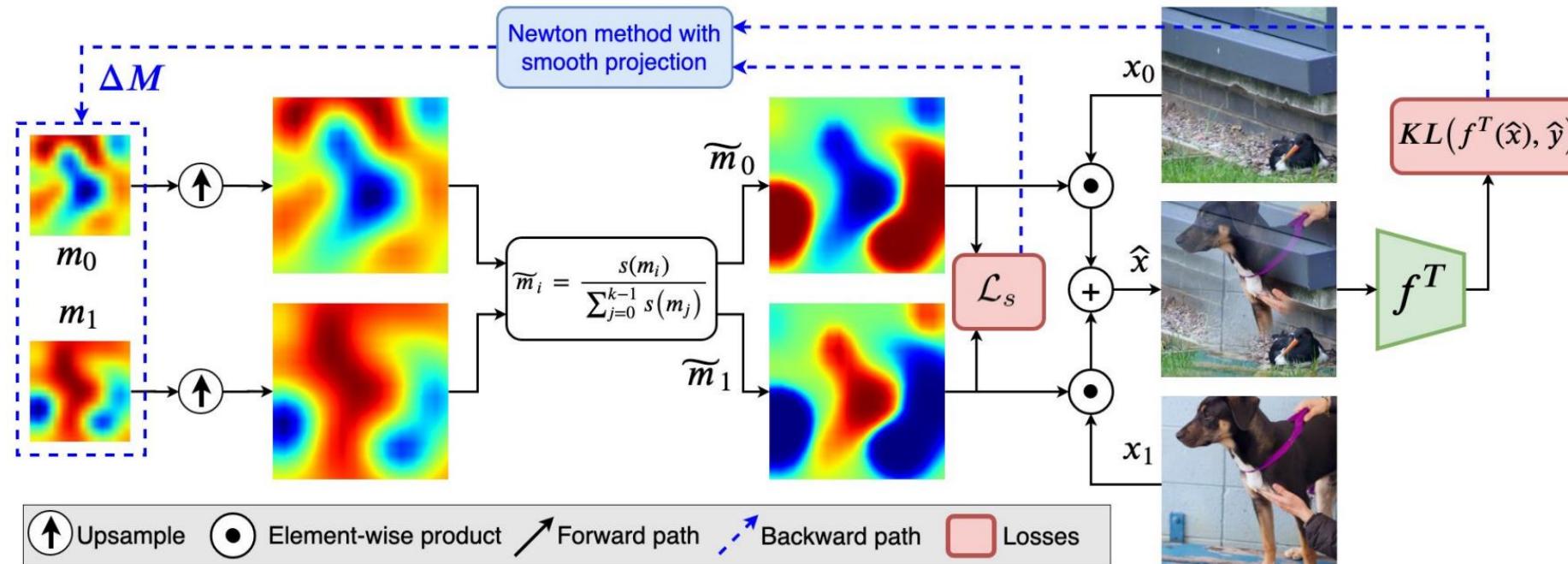
1.7 Saliency-based

SuperMix: Supervising the Mixing Data Augmentation (CVPR`2021)

SuperMix利用一个Pre-trained Model作为教师模型，利用教师模型的知识进行指导并优化。

首先，从Dirichlet distribution中采样生成一系列mask，得到混合样本。根据混合样本标签生成一个期望的标签分布，让教师模型在混合样本上的分布和目标分布更接近，以此来优化mask。

生成的图像必须靠近训练数据的流形，所以作者使用预测时最大概率和标签的关系来进行约束。其次，要保证生成的mask必须是稀疏的，防止在每个空间位置对多个图像进行平均并抑制重要特征。



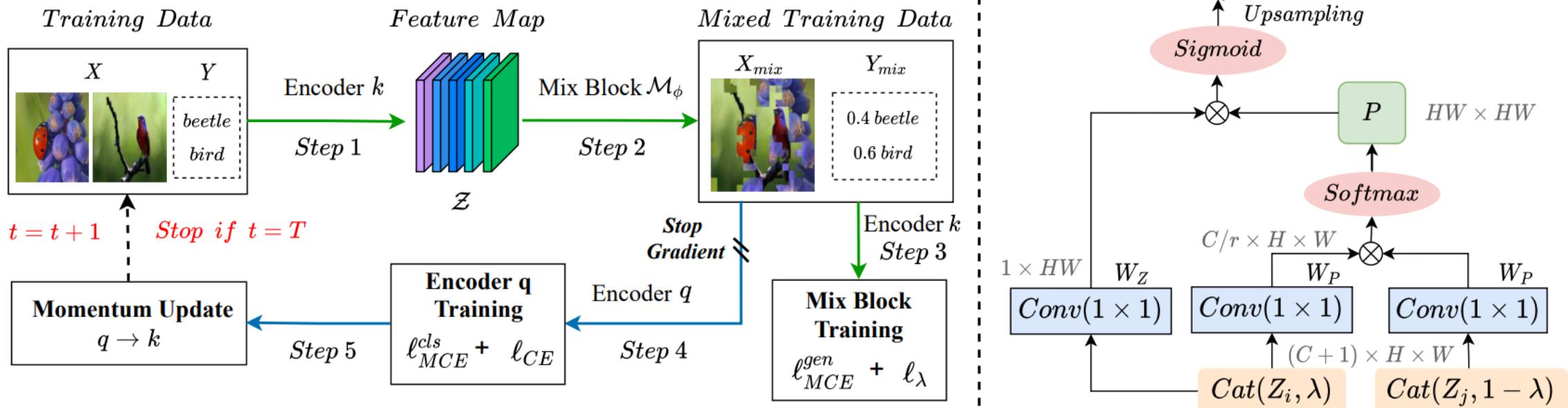
1.7 Saliency-based

AutoMix: Unveiling the Power of Mixup for Stronger Classifiers (ECCV`2022)

AutoMix采用一个双重优化的方法来平衡生成器和分类器。

使用一个具有相同架构但是没有梯度的Encoder k来解耦生成器和分类器。具体来说，样本经过k得到feature map，然后嵌入 λ map。送入到设计的MixBlock后经过cross-attention计算得到样本对之间的特征关系来生成mask得到混合样本。得到的混合样本一方面保持MixBlock的梯度送入k得到损失进行训练，另一方面截断梯度送入有梯度的q中训练分类器。k的能力由q通过EMA来传入。

双重优化的方法使得AutoMix在时间开销和分类性能上都达到了最优。



1.7 Saliency-based

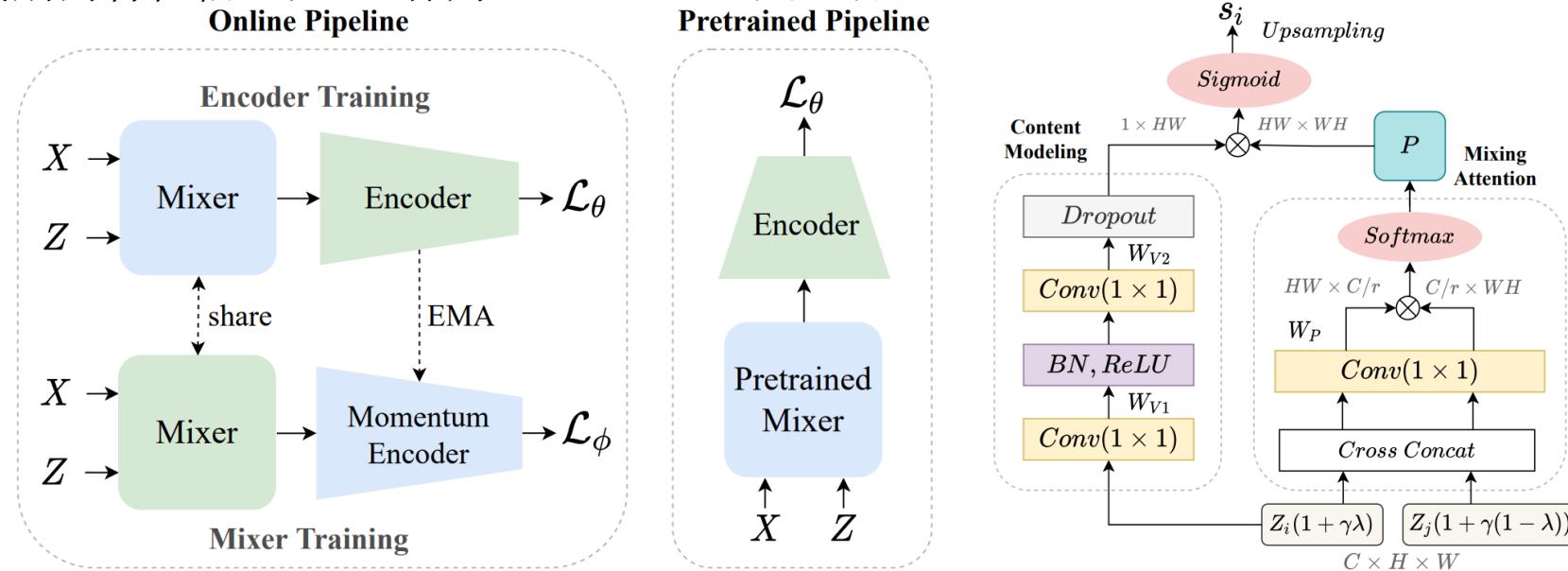
Boosting Discriminative Visual Representation Learning with Scenario-Agnostic Mixup (arXiv`2021)

SAMix主要改进两点：

在AutoMix的MixBlock中， v 的计算可以看成一个Linear，可能限制了性能，SAMix将其改进为MLP，性能也得到了提升； SAMix不仅仅针对SL任务，还可以应用于SSL任务中。

针对使用mixup方法的SSL设计了一个infoNCE， $\mathcal{L}^{cls} = l_+^{CE} + \eta l_-^{CE}$ 。

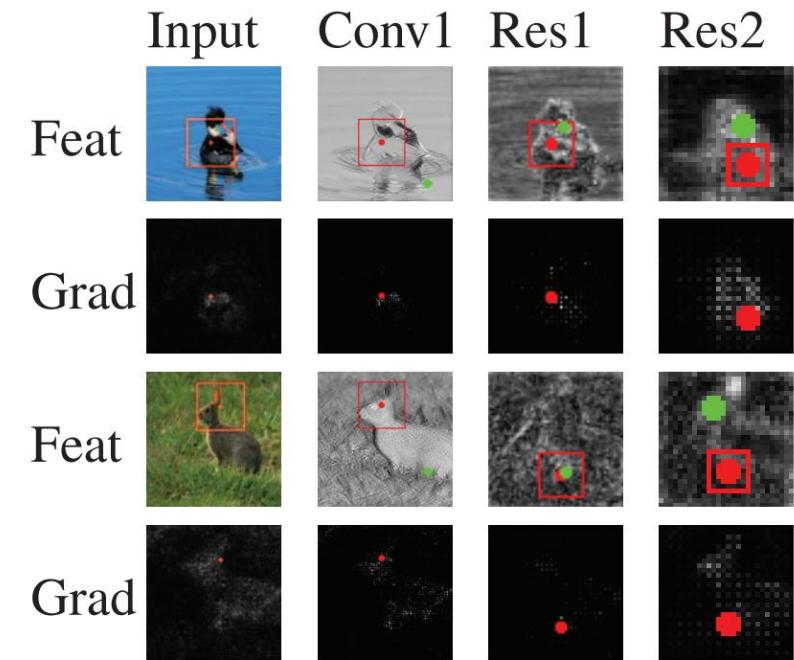
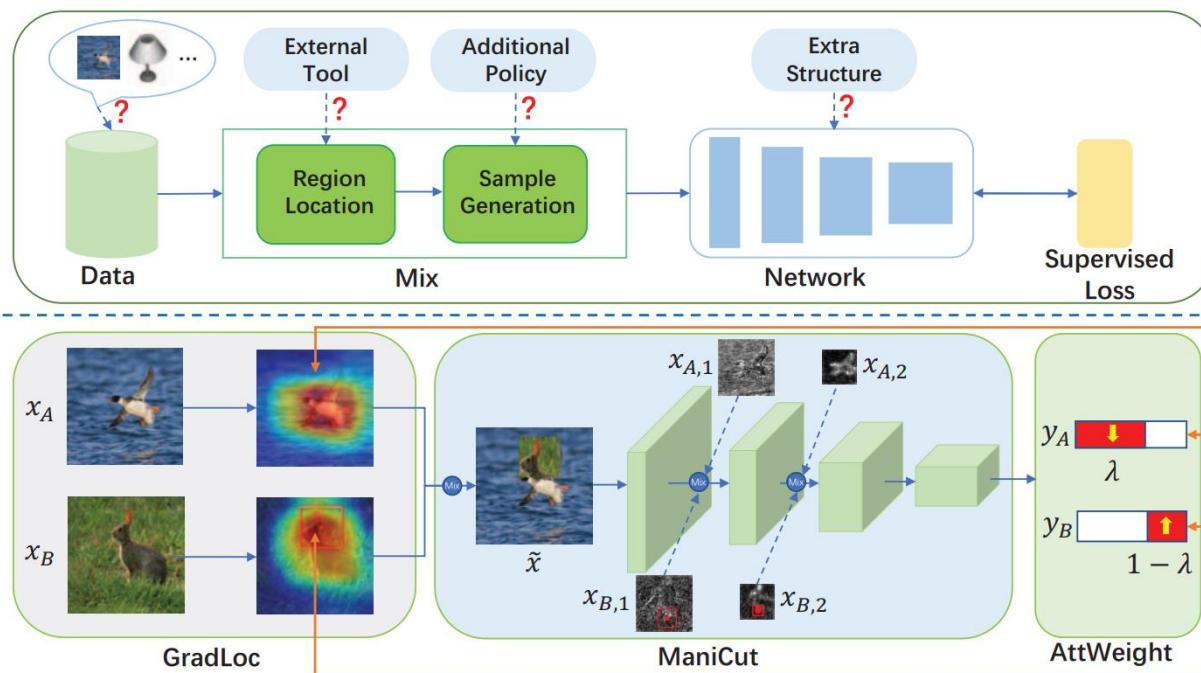
l_+^{CE} 称为Local Term，因为考虑到混合样本存在两个类别的信息，所以在原本的infoNCE损失分母中加入了另一个类别的特征信息； l_-^{CE} 称为Gobal Term，是混合infoNCE。



1.7 Saliency-based

GradSalMix: Gradient Saliency-Based Mix for Image Data Augmentation (ICME`2023)

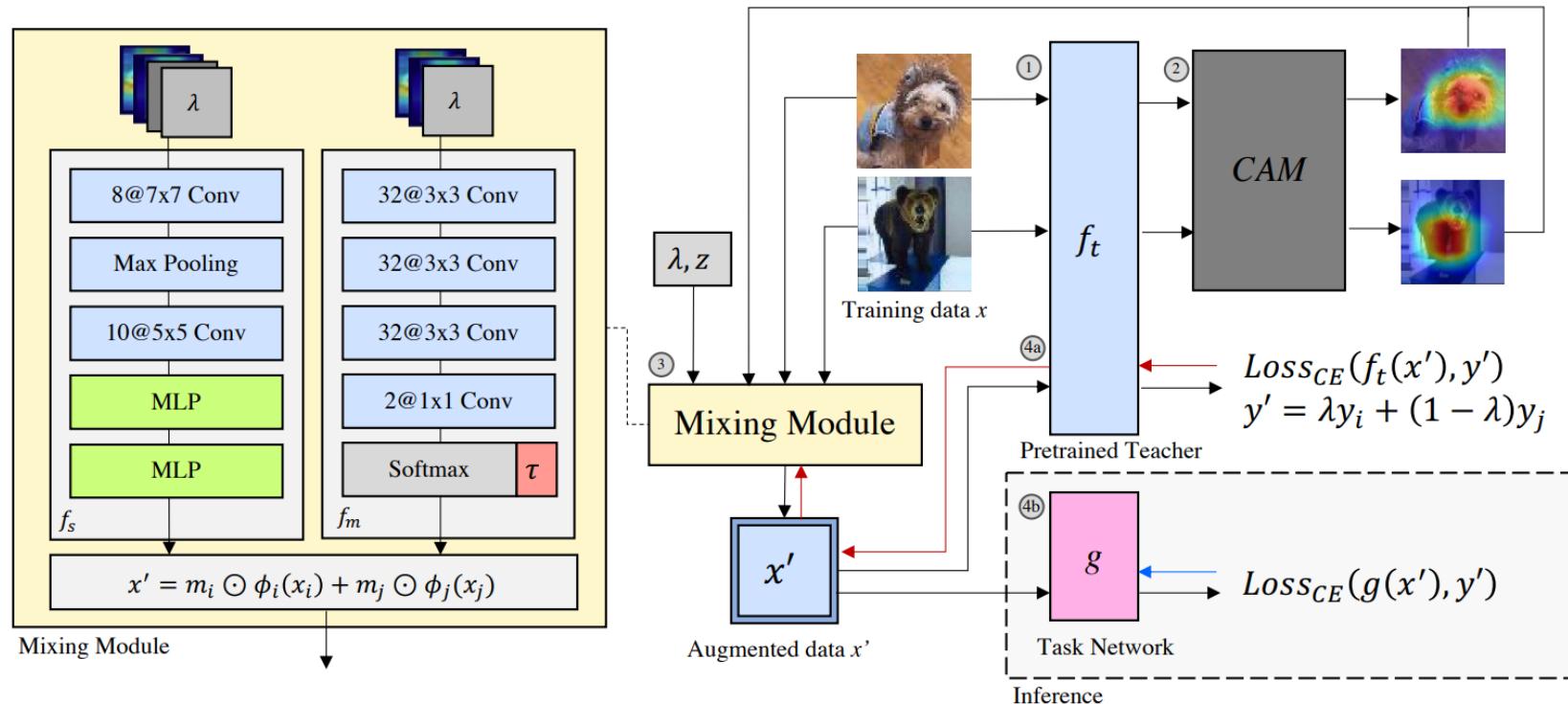
作者认为，以往的方法通过显著性信息 (CAM)并不可靠，作者采用了梯度的方式来定位选择最重要的patch。并且不仅仅在input level，在每层隐空间中也进行了混合。



1.7 Saliency-based

TransformMix: Learning Transformation and Mixing Strategies from Data (arXiv`2024)

TransformMix首先利用一个Pre-trained Model作为教师模型来获得Raw样本的CAM，将得到的CAM+Raw样本的feature map+ λ map送入到提出的Mixing Module来生成mask进行混合得到混合样本。Mixing Module由一个空间网络和mask预测网络组成。

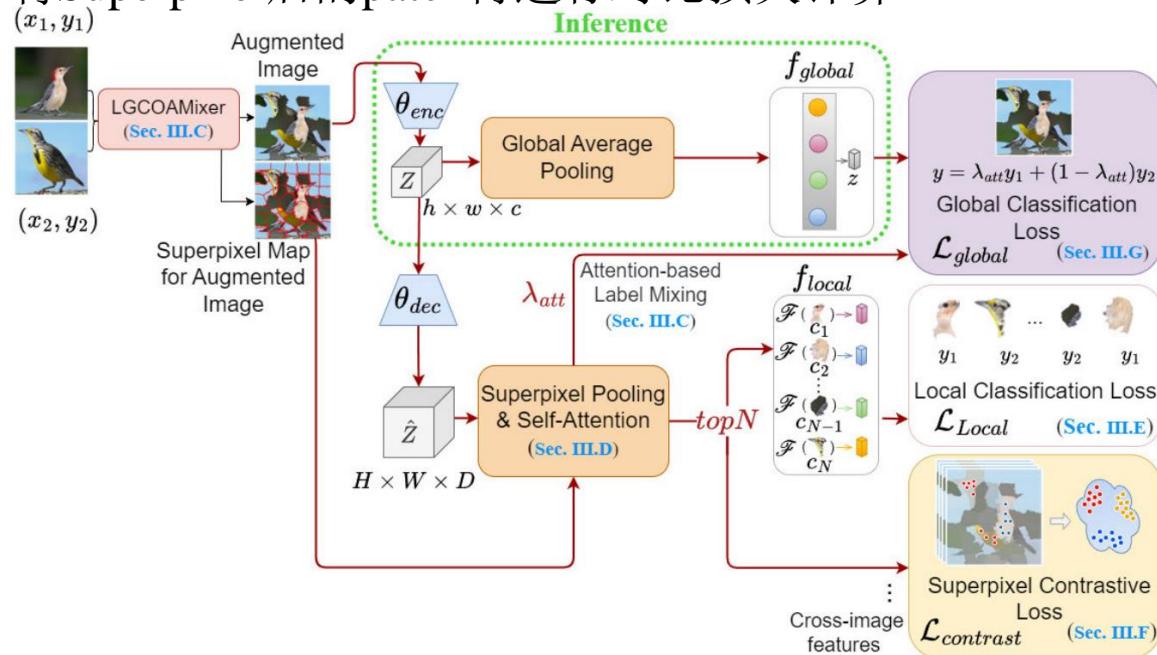


1.7 Saliency-based

LGCOAMix: Local and Global Context-and-Object-Part-Aware Superpixel-Based Data Augmentation for Deep Visual Recognition (TIP`2024)

作者将Raw samples经过Superpixel后进行混合得到增强样本，然后计算MCE损失。此外，还添加了多种不同的损失函数用于辅助。

将feature maps进行解码后进行注意力计算并获得新的混合比例；将解码后的样本分割多个patch，按照TopN来计算局部损失；将Superpixel后的patch再进行对比损失计算。



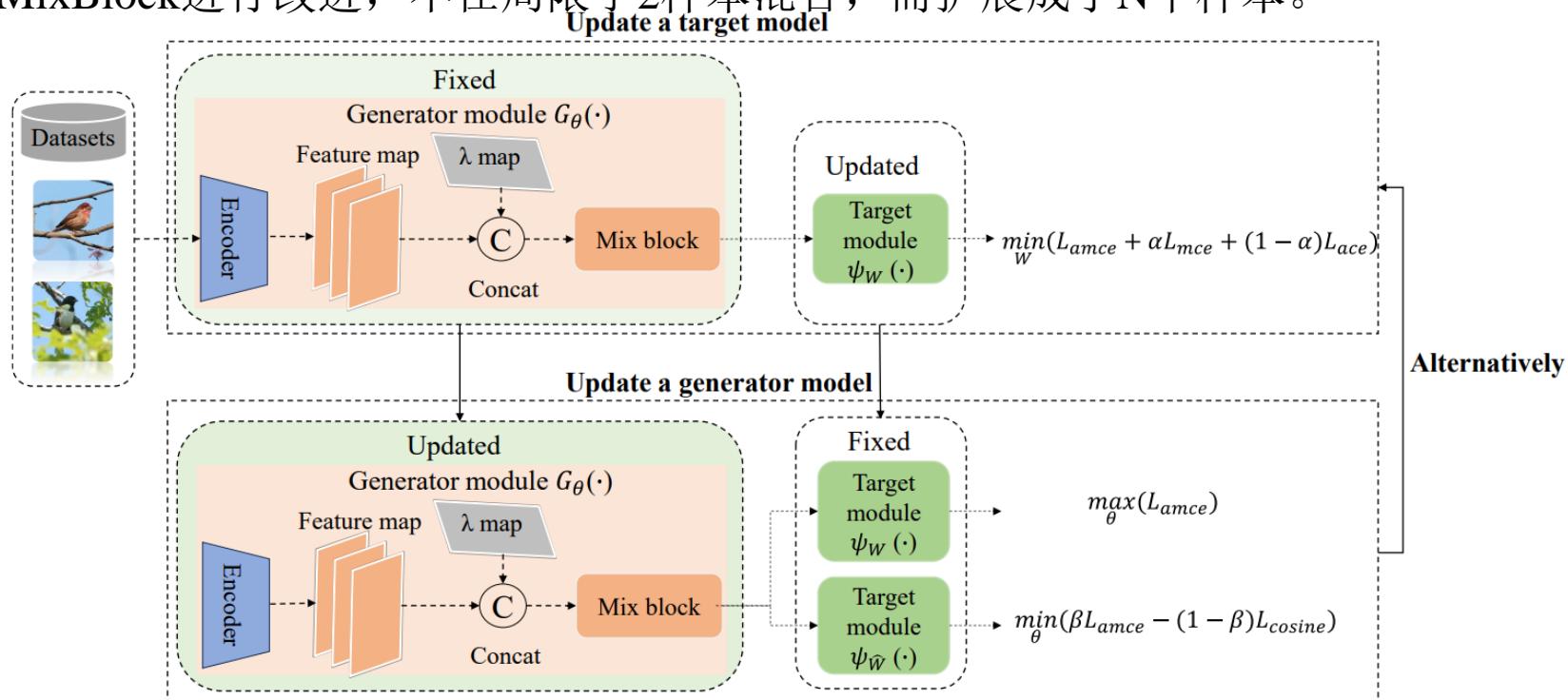
1.7 Saliency-based

Adversarial AutoMixup (ICLR`2024)

AdAutoMix对AutoMix进一步改进。

第一，AutoMix的生成器在经过大量epochs训练后，会产生过拟合的现象，限制了分类器的能力。AdAutoMix使用对抗训练的方式目的使MixBlock生成具有“挑战性”的样本，进一步挖掘分类器性能。

第二，将MixBlock进行改进，不在局限于2样本混合，而扩展成了N个样本。



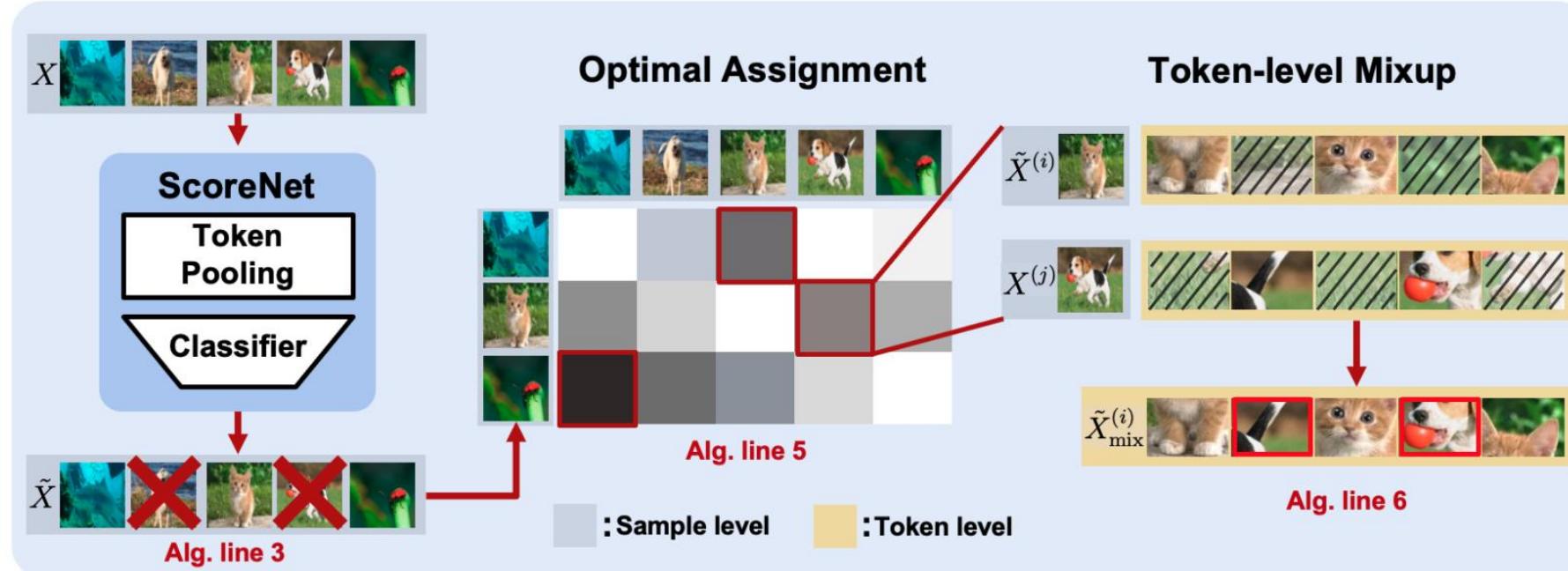
- 1.1 Static Linear
- 1.2 Feature-based
- 1.3 Cutting-based
- 1.4 k Sample Mixup
- 1.5 Random Policies
- 1.6 Style-based
- 1.7 Saliency-based
- 1.8 Attention-based
- 1.9 Generating Samples

1.8 Attention-based

TokenMixup: Efficient Attention-guided Token-level Data Augmentation for Transformers (NIPS`2022)

TokenMixup首先对于整个样本集合先使用ScoreNet的参数化模块(简单的MLP)评估输入样本的难度，该模块根据中间标记预测目标值。如果难度分数低于一定阈值，则将样本标记为易于增强。

为了最大的提升混合后的显著性，使用了Hungarian Matching的最优对齐方法将易样本在Mini-batch内进行最佳匹配。然后对于选择混合的样本对，使用注意力得分而不是基于梯度进行显著性检测选择混合。

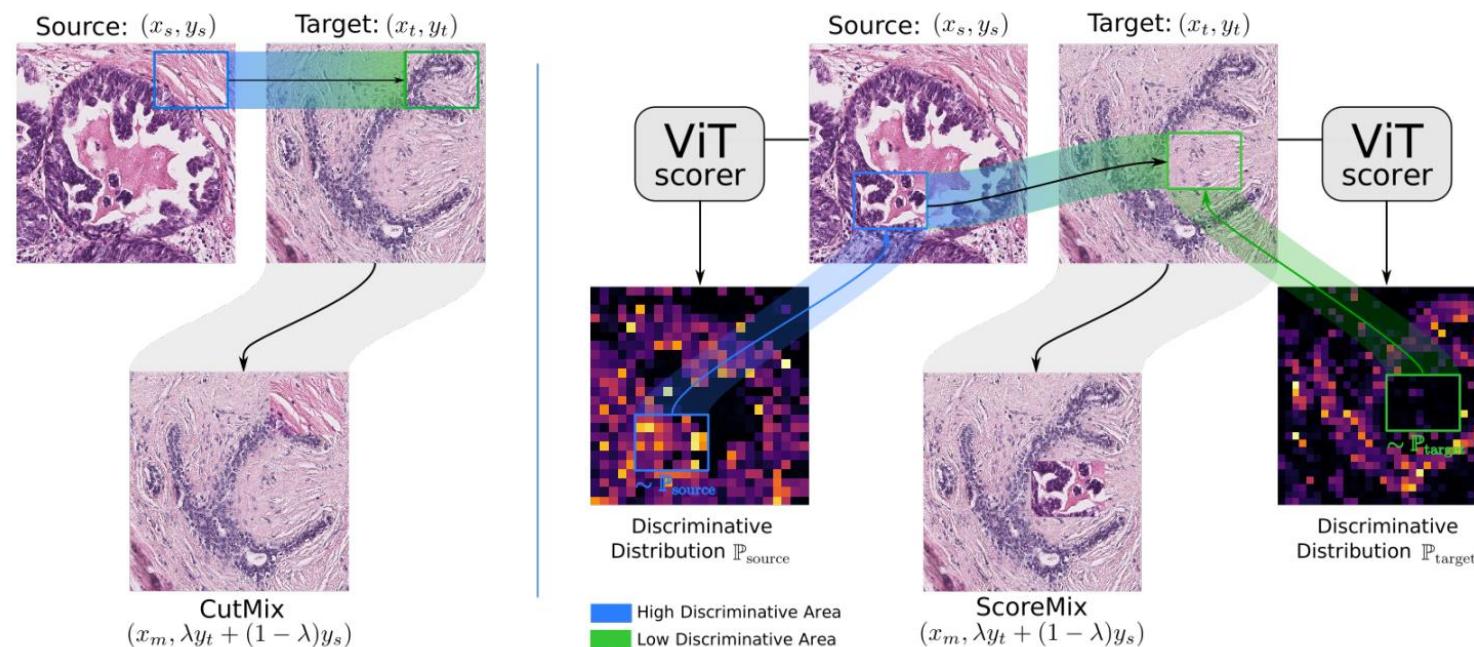


1.8 Attention-based

ScoreNet: Learning Non-Uniform Attention and Augmentation for Transformer-Based Histopathological Image Classification (WACV'2023)

根据样本在ViT中的注意力得分来进行选择性的CutMix。

将样本输入到ViT score后得到样本的显著性信息，然后通过挑选Max和Min区域进行混合得到混合样本。

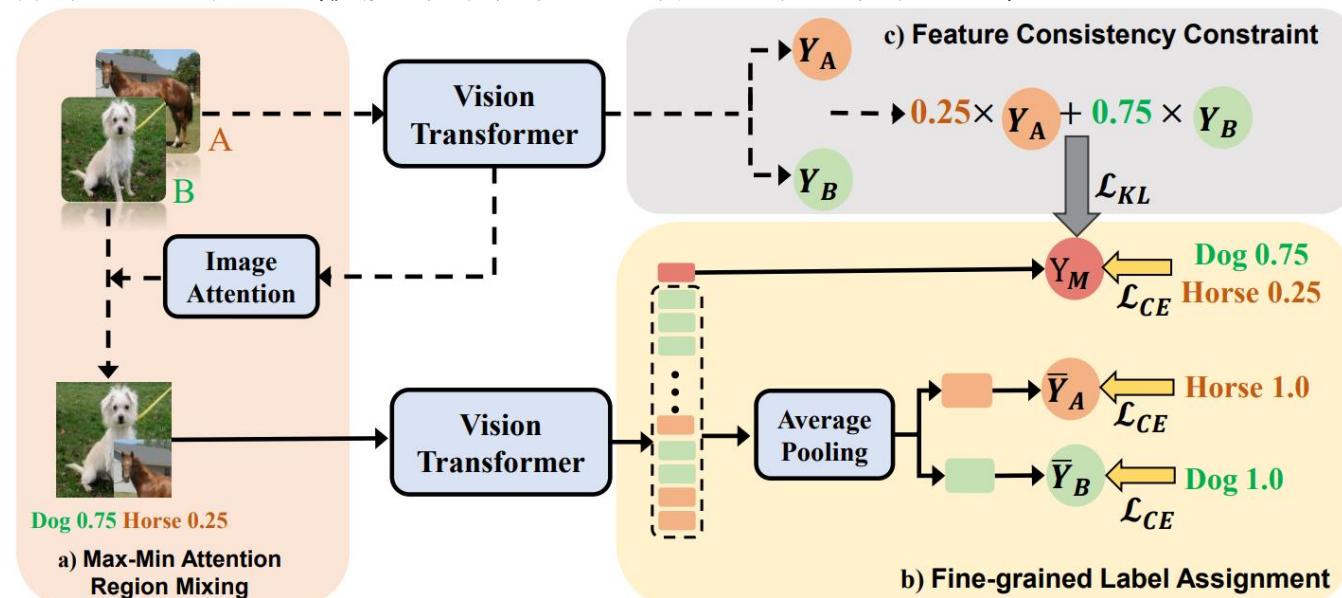


1.8 Attention-based

SMMix: Self-Motivated Image Mixing for Vision Transformers (ICCV`2023)

SMMix与之前提到的ScoreMix类似的混合方法，即利用ViTs的注意力得分来分别选择Max和Mini区域进行指导性的混合。

不过SMMix对标签层面进行了进一步约束。作者认为MCE损失只考虑混合样本的整体信息。然而，SMMix赋予混合样本内的对象 y_A 和 y_B 。因此，对不同标签的混合图像进行不同区域的监督是可行的。为了达到这个目的，作者将最终输出的token序列重塑为Raw样本进行不同类别的CE损失计算。此外，对于ViT得到的cls token，作者还进行KL散度预测与混合标签的一致性约束。



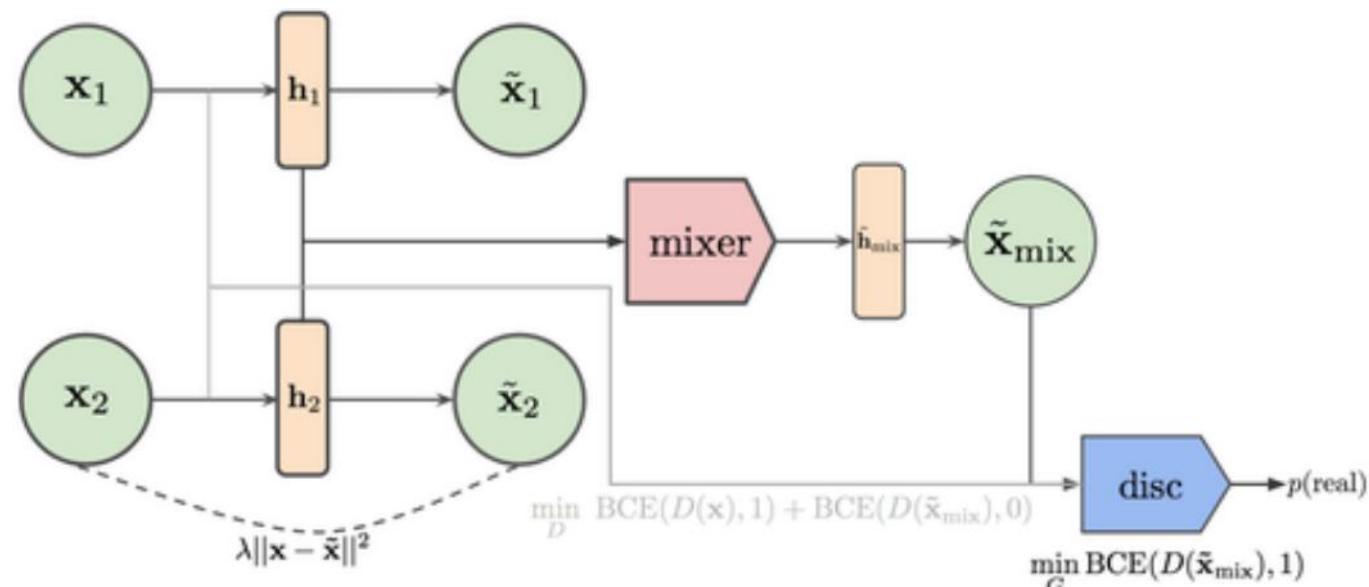
- 1.1 Static Linear**
- 1.2 Feature-based**
- 1.3 Cutting-based**
- 1.4 k Sample Mixup**
- 1.5 Random Policies**
- 1.6 Style-based**
- 1.7 Saliency-based**
- 1.8 Attention-based**
- 1.9 Generating Samples**

1.9 Generating Samples

On Adversarial Mixup Resynthesis (NIPS`2019)

AMR使用两个样本的feature map进行manifold mixup后，使用生成器生成增强样本。通过同时训练判别器(D)和欺骗器(F)来进行对抗训练。其中， l_{GAN} 是一个BCE Loss。

$$\begin{aligned} \min_F \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim p(\mathbf{x})} & \underbrace{\lambda ||\mathbf{x} - g(f(\mathbf{x}))||_2}_\text{reconstruction} + \underbrace{l_{GAN}(D(g(f(\mathbf{x}))), 1)}_\text{fool D with reconstruction} + \underbrace{l_{GAN}(D(g(\text{Mix}(f(\mathbf{x}), f(\mathbf{x}'))))), 1)}_\text{fool D with mixes} \\ \min_D \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim p(\mathbf{x})} & \underbrace{l_{GAN}(D(\mathbf{x}), 1)}_\text{label x as real} + \underbrace{l_{GAN}(D(g(f(\mathbf{x}))), 0)}_\text{label reconstruction as fake} + \underbrace{l_{GAN}(D(g(\text{Mix}(f(\mathbf{x}), f(\mathbf{x}'))))), 0)}_\text{label mixes as fake}. \end{aligned}$$

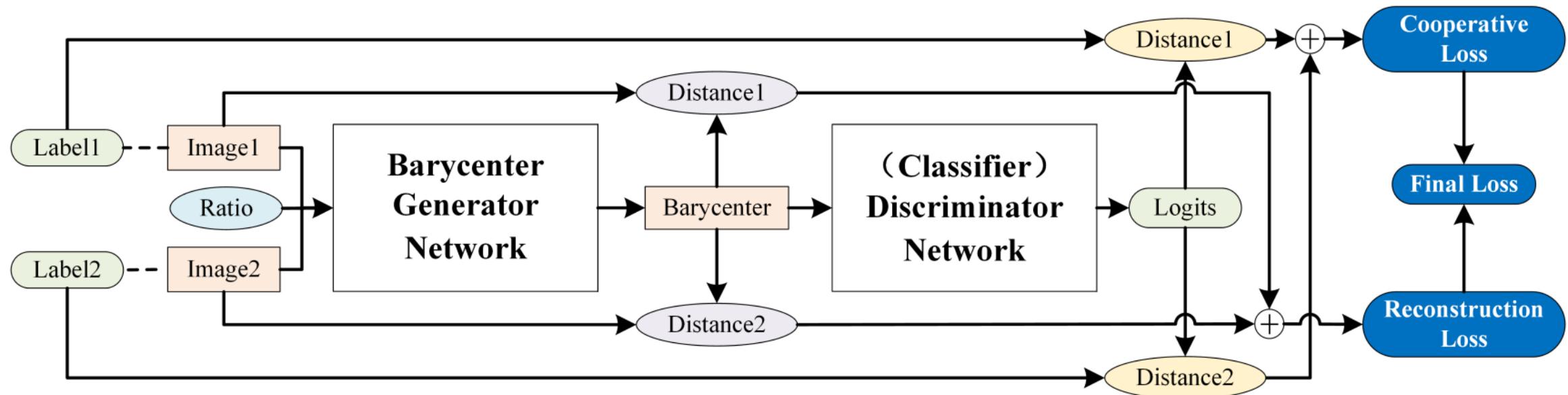


1.9 Generating Samples

AutoMix: Mixup Networks for Sample Interpolation via Cooperative Barycenter Learning (ECCV`2020)

AutoMix通过一个改进版U-Net来进行生成混合样本，然后根据KL散度来度量混合样本与2个原样本之间的距离得到重构损失；此外，再经过一个分类器(鉴别器)来和源样本的标签进行KL度量得到损失函数。

最终的损失函数则是：Reconstruction Loss + Cooperative Loss。

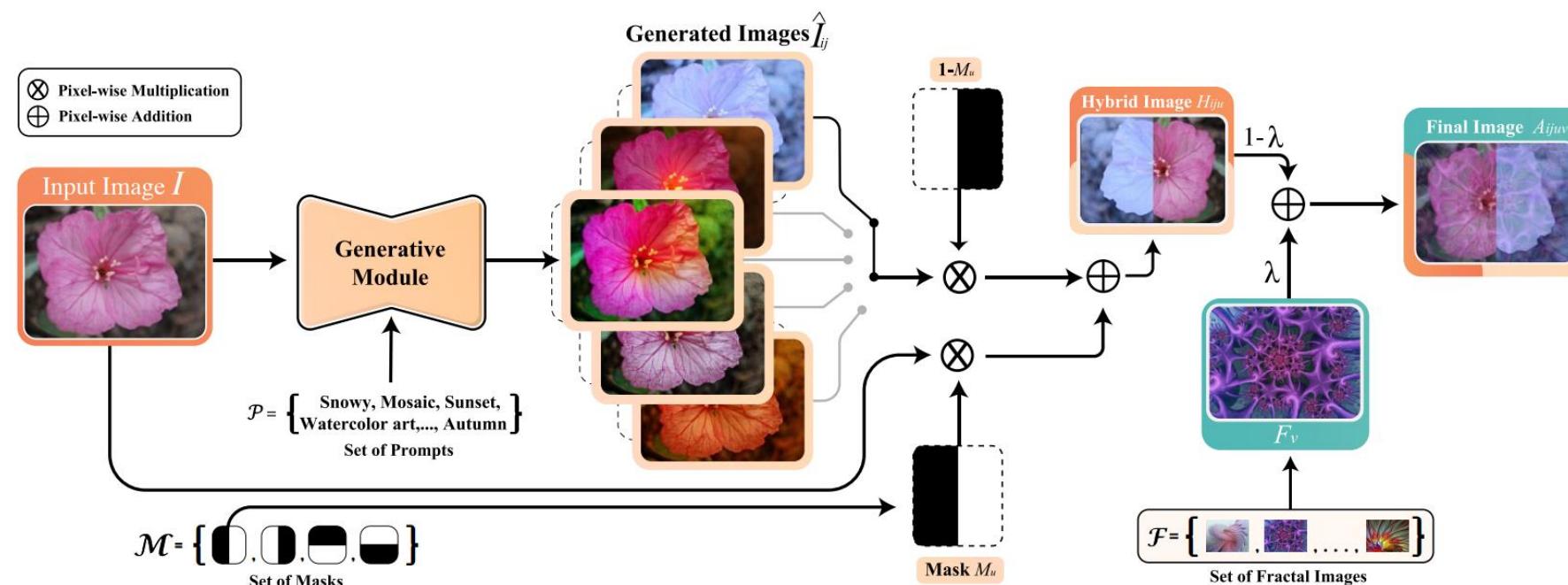


1.9 Generating Samples

DiffuseMix: Label-Preserving Data Augmentation with Diffusion Models (CVPR`2024)

DiffuseMix结合了提示词(prompt)和多模态生成模型来进行增强。

具体来说：1. 通过设定的增强信息的prompt和生成模型得到初步的多种增强样本；2. 按照设定的mask集合M来对增强样本和源样本进行混合增强；3. 再通过一个设定好的纹理集合F，随机挑选一个纹理样本后与混合增强样本进行线性插值得到最终的混合增强样本。



Contents

1. Sample Policies

2. Label Policies

2.1 Loss Object

2.2 Optimize Calibration

2.3 Optimize λ

2.4 Generate Label

2.5 Attention Score

2.6 Saliency Tokens

2.1 Loss Object

*MixupE: Understanding and Improving Mixup from Directional Derivative Perspective (UAI'2023)

文章分析了流行的mixup方法，表明它隐含地正则化了无限多个方向导数的所有阶数，与普遍认为可以用二阶正则化替代的观点相反。

至此，MixupE通过增强mixup对方向导数的隐含正则化效果，提出了一种改进的方法。作者在图像、表格数据、语音和图形等领域进行了实验，展示了所提出方法的有效性，一致显示出比mixup和ERM更好的泛化误差。

```
def beta_mean(alpha, beta):
    return alpha/(alpha+beta)

lam_mod_mean = beta_mean(alpha+1, alpha) # mean of beta distribution

# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    loss = loss_function(net(x), y) # mixup loss
    loss_scale = torch.abs(loss.detach().data.clone())
    f = net(x1)
    b = y1 - torch.softmax(f, dim=1)
    loss_new = torch.sum(f * b, dim=1)
    loss_new = (1.0 - lam_mod_mean) * torch.sum(torch.abs(loss_new)) / batch_size # additional loss term
    loss = loss + (mixup_eta * loss_new) # total loss
    loss_new_scale = torch.abs(loss.detach().data.clone())
    loss = (loss_scale / loss_new_scale) * loss # loss after scaling
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

2.1 Loss Object

Harnessing Hard Mixed Samples with Decoupled Regularizer (NIPS`2023)

DecoupleMix目的是解决mixup中softmax存在的“winner take all”现象。在mixup任务中，由于样本和标签是存在2个或者多个，在使用softmax后会大大削弱其中某一个类别信息，当其中一个 λ 过高时，会产生的高的熵，但置信度也许会很低，这样会严重影响单个类别的预测。因此，提出一个有效的适合mixup的目标函数，有一个解耦正则化(DM)，可以自适应地利用这个“硬混合”样本来挖掘特征。

DecoupleMix解耦mixup中两个类别，在最后预测进行softmax时在分母中去掉自身的概率，并使用two-hot做正则化。作者在监督和半监督任务上都做了大量的实验，发现解耦后的损失函数能带来显著的提升。

$$\phi(z_{(a,b)})^{i,j} = \frac{\exp(z_{(a,b)}^i)}{\exp(\cancel{z_{(a,b)}^j}) + \sum_{c \neq j} \exp(z_{(a,b)}^c)}.$$

$$\mathcal{L}_{DM(CE)} = - \left(\underbrace{y_{(a,b)}^T \log(\sigma(z_{(a,b)}))}_{\mathcal{L}_{MCE}} + \eta \underbrace{y_{[a,b]}^T \log(\phi(z_{(a,b)})) y_{[a,b]}}_{\mathcal{L}_{DM}} \right).$$

2.1 Loss Object

2.2 Optimize Calibration

2.3 Optimize λ

2.4 Generate Label

2.5 Attention Score

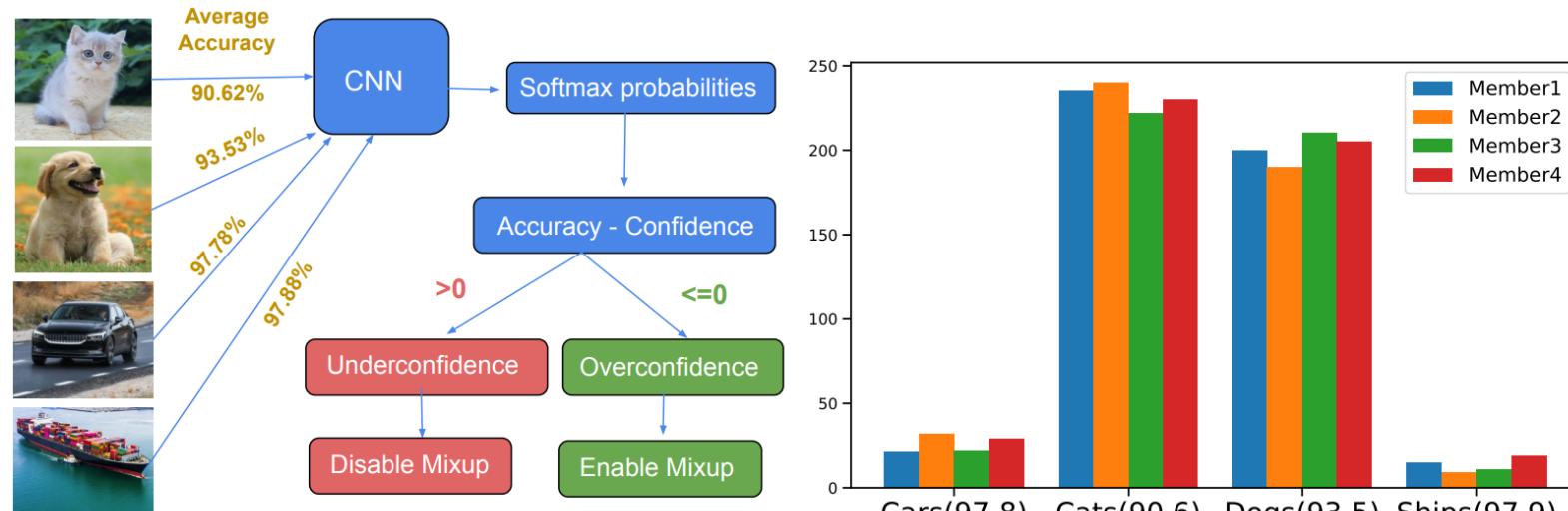
2.6 Saliency Tokens

2.2 Optimization Calibration

Combining Ensembles and Data Augmentation can Harm your Calibration (ICLR`2021)

作者发现MixUp与交叉熵(CE)简单应用会增加神经网络的期望校准误差(Expected Calibration Error, ECE)，这与目前大众的直觉相反。作者使用Label Smoothing做实验验证了是标签问题。

提出的CAMixup是根据类别分类的平均准确率与置信度之间的差异来调整混合比例 λ 。理想情况下，希望模型在预测简单类（如汽车和船舶）时更有信心。对于像猫和狗这样更难的类，模型被鼓励不那么自信地实现更好的校准。



(a) Proposed CAMixup method.

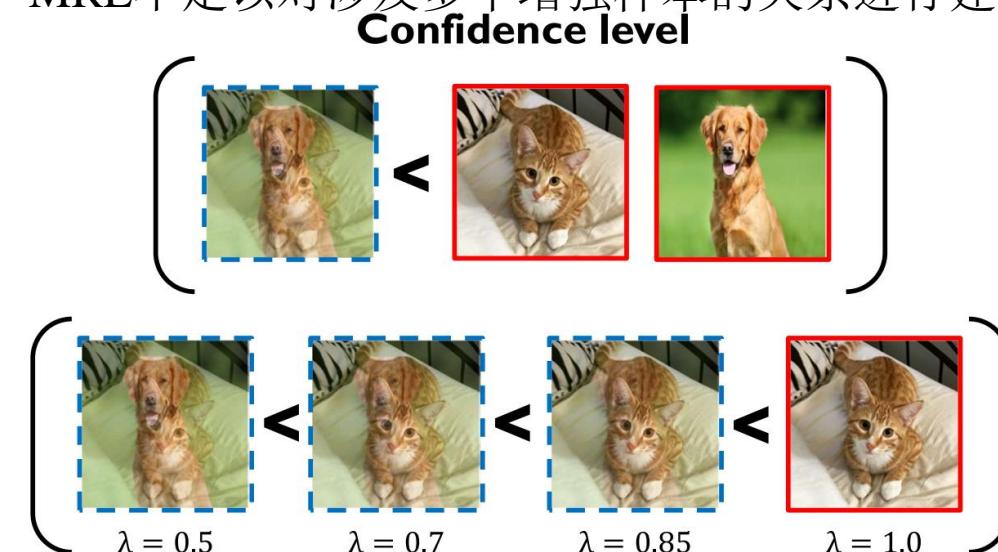
(b) # epochs in which Mixup was applied to each class.

2.2 Optimization Calibration

RankMixup: Ranking-Based Mixup Training for Network Calibration (ICCV`2023)

作者提出了一种新颖的基于mixup的网络校准框架 RankMixup，它缓解了利用标签混合带来的网络校准问题。

1. 作者认为易分类样本的置信水平应高于难分类样本的置信水平的关系。为此引入了Mixup-based Ranking Loss (MRL)。它鼓励混合样本的置信度比原始样本的置信度低一些，以保持排名关系。期望较高的置信度有利于 λ 较大的混合样本，反之亦然，这样置信度和混合系数的阶数彼此对齐。
2. 提出利用Raw样本和混合样本的置信度之间以及不同比例下的混合样本的置信度之间的顺序排序关系，将它们用作监督信号。MRL不足以对涉及多个增强样本的关系进行建模，所以提出M-NDCG解决多个样本的置信度排名问题。



2.1 Loss Object

2.2 Optimize Calibration

2.3 Optimize λ

2.4 Generate Label

2.5 Attention Score

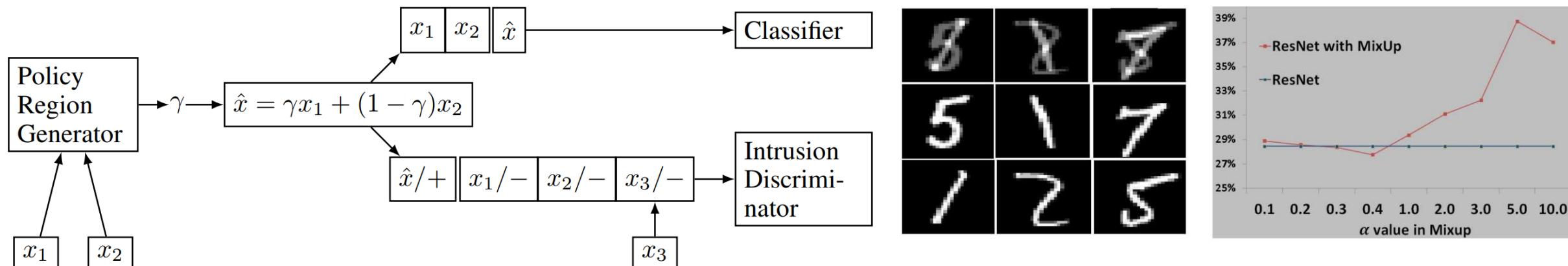
2.6 Saliency Tokens

2.3 Optimization λ

MixUp as Locally Linear Out-Of-Manifold Regularization (AAAI'2019)

AdaMixup指出了一个现象，即进行mixup的时候，可能混合样本与某些Raw样本会相同，就是“流形侵入”(manifold intrusion)。作者在实验中发现在 $\lambda=0.5$ 时影响最严重。故提出一个可学习 λ 的方法。引入两个神经网络 $\pi(\cdot)$ 和 $\varphi(\cdot)$ 分别用来生成混合所用的系数 λ 和判断新混合的数据是否造成流形入侵。

其中，作者利用重参数化的技巧来学习 λ ： $\gamma = \Delta\epsilon + \alpha$ ，这个 α 能够尽量的是混合样本避免“流形侵入”问题。



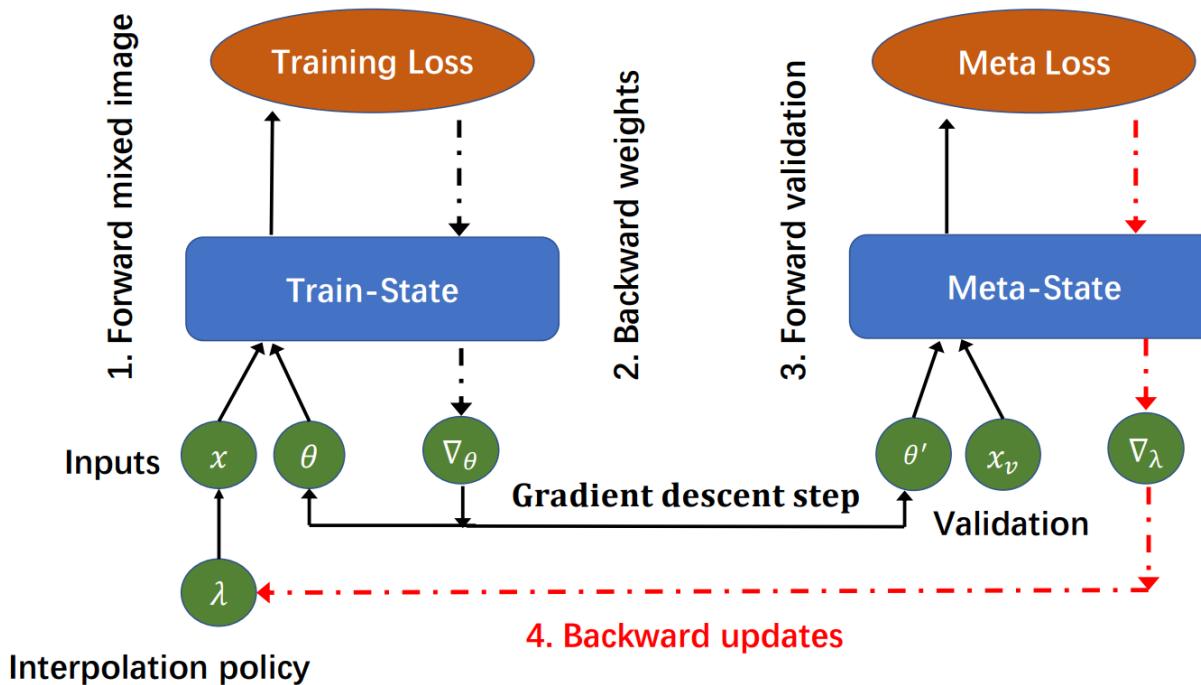
2.3 Optimization λ

MetaMixUp: Learning Adaptive Interpolation Policy of MixUp with Meta-Learning (TNNLS`2021)

MetaMixUp将mixup方法与Meta Learning相结合。目的是双重优化参数：神经网络和混合比例 λ 。

其中，Train-State利用学习的 λ 进行混合样本增强来训练网络，得到梯度训练网络参数，且将网络梯度传入Meta-State，Meta-State利用验证集得到梯度来修改混合比例 λ 。

循环以往，进行双重优化，让网络“学习如何学习”。



2.3 Optimization λ

LUMix: Improving Mixup by Better Modelling Label Uncertainty (ICASSP`2024)

LUMix不同于以往试图摆脱mixup过程带来的标签噪声，反而对标签噪声进行建模。为此，提出了两个因素，一是从神经网络预测中产生的，判断输入中是否存在显著目标。另一个从均匀分布中随机采样添加到标签分布中以模拟mixup过程中的标签不确定性，最终的比例为： $\lambda = \lambda_0(1 - r_s - r_r) + \lambda_s r_s + \lambda_r$
 λ_0 为人工设置， λ_s 是由预测分布计算而来， λ_r 则是从 $Beta(a, a)$ 采样得到。

$$\lambda_s = \frac{\hat{p}_A}{\hat{p}_A + \hat{p}_B}, \quad \hat{p}_i = \frac{e^{p_i}}{\sum_{j=1}^C e^{p_j}}$$

Models	#Params	FLOPs	Epochs	Top-1 (%)	+LUMix Top-1 (%)
ConvNeXt-T [23]	29M	4.5G	300	82.1	82.5
Swin-T [22]	28M	4.5G	300	81.3	81.7
DeiT-S [32]	22M	4.7G	300	79.8	80.6
XCiT-S [9]	26M	4.8G	400	82.0	82.3
ConvNeXt-S [23]	50M	8.7G	300	83.1	83.3
Swin-S [22]	50M	8.7G	300	83.0	83.1
ConvNeXt-B [23]	89M	15.4G	300	83.8	84.1
Swin-B [22]	88M	15.4G	300	83.5	83.6
XCiT-M [9]	84M	16.2G	400	82.7	83.2
DeiT-B [32]	87M	17.6G	300	81.8	82.2
XCiT-L [9]	189M	36.1G	400	82.9	83.6

Algorithm 1 Pseudo-code of RandMix in PyTorch style.

```

# B: number of images in a batch (batch size).
# C: number of classes
# logits: output of the network
# alpha: the hyper-parameter for Beta distribution
# y1, y2: labels for different samples
# r_s, r_r: ratios for lam_s and lam_r
# ce_loss: the default softmax cross entropy loss

def randmix(logits, y1, y2):
    # y1, y2: [B, C]
    scores = F.softmax(logits, dim=-1)
    prob1, prob2 = scores[y1 > y1.min()], scores[y2 > y2.min()]
    # lam_s, lam_r, lam: [B]
    lam_s = prob2 / (prob1 + prob2)
    # samples lam_r for B times from a beta
    # distribution.
    lam_r = beta(alpha, alpha, B)
    lam = lam0 * (1 - r_s - r_r) + lam_s * r_s + lam_r
    * r_r
    y = (1-lam) * y1 + lam * y2
    # generate the binarized label b
    b[y1 > y1.min()] = 1
    b[y2 > y2.min()] = 1
    loss = ce_loss(logits, y) + y * max(0, b - scores)

```

2.1 Loss Object

2.2 Optimize Calibration

2.3 Optimize λ

2.4 Generate Label

2.5 Attention Score

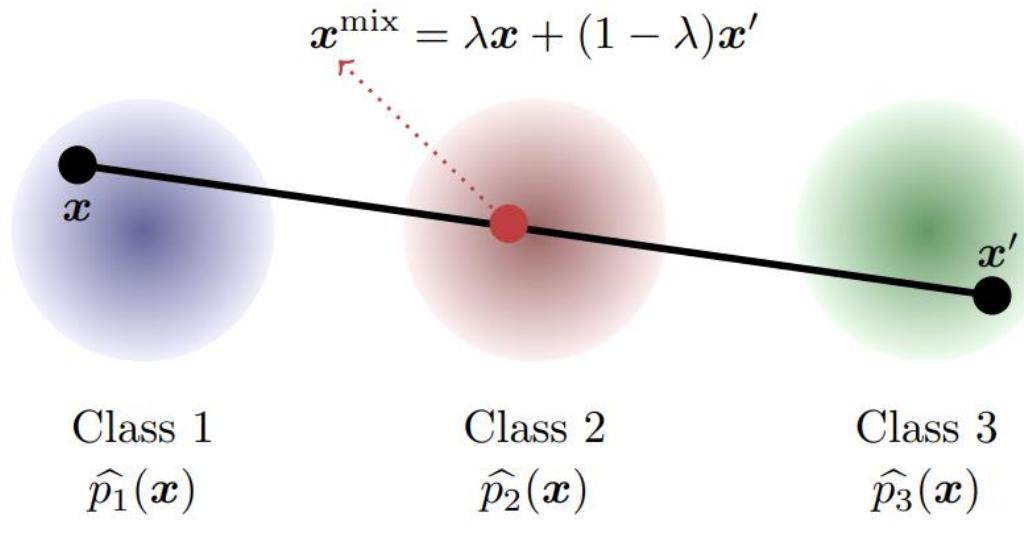
2.6 Saliency Tokens

2.4 Generate Label

GenLabel: Mixup Relabeling using Generative Models (ICML`2022)

*GenLabel*想要解决“流形侵入”(manifold intrusion)问题。再流形学习任务中，举个简单的例子MINST中特殊的1和2只是进行简单的线性插值话可能得到8这个样本特征，但是标签对应的还是1和2，那这是错误的。

*GenLabel*重新标记混合样本。首先学习每个类c的底层数据分布，表示为 $p_c(x)$ 。然后，基于从每个类别c中提取的混合样本的似然性 $p_c(x_{\text{mix}})$ ，得到相似性后将混合样本重新生成标签 y^{gen} 。



$$\mathbf{y}^{\text{mix}} = \lambda \mathbf{y} + (1 - \lambda) \mathbf{y}' = \lambda \mathbf{e}_1 + (1 - \lambda) \mathbf{e}_3$$

$$\mathbf{y}^{\text{gen}} = \sum_c \frac{\hat{p}_c(\mathbf{x}^{\text{mix}})}{\sum_{c'} \hat{p}_{c'}(\mathbf{x}^{\text{mix}})} \mathbf{e}_i \simeq \mathbf{e}_2$$

\uparrow

$\hat{p}_2(\mathbf{x}^{\text{mix}}) \gg \hat{p}_1(\mathbf{x}^{\text{mix}}),$
 $\hat{p}_2(\mathbf{x}^{\text{mix}}) \gg \hat{p}_3(\mathbf{x}^{\text{mix}})$

2.1 Loss Object

2.2 Optimize Calibration

2.3 Optimize λ

2.4 Generate Label

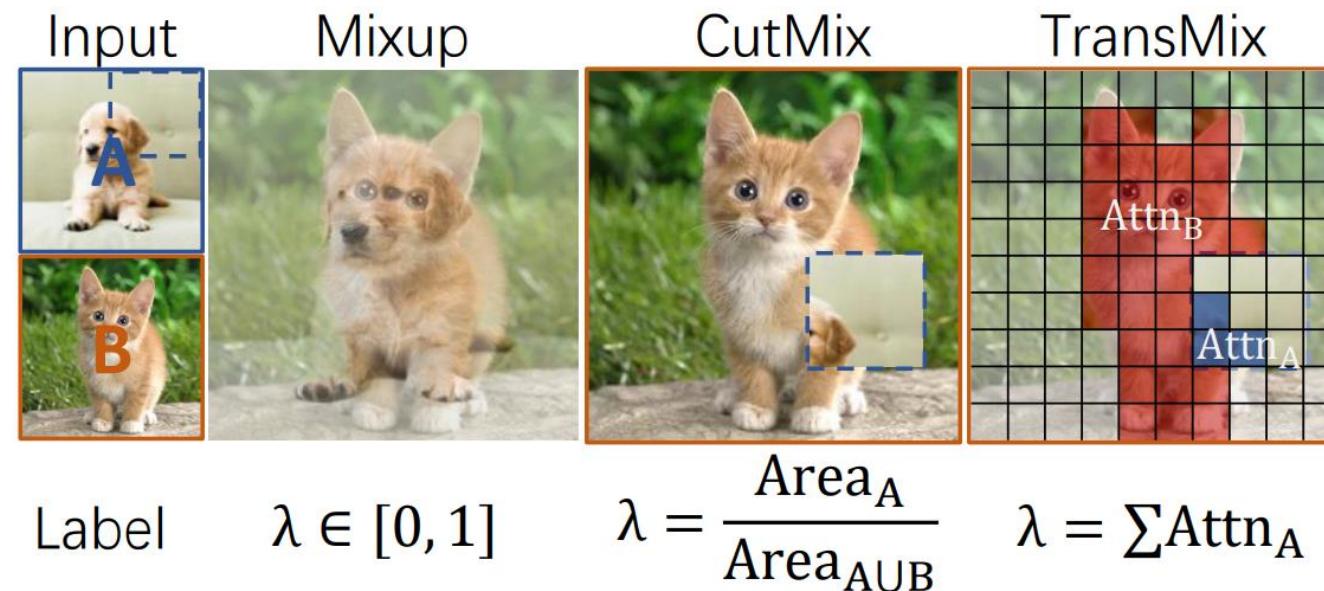
2.5 Attention Score

2.6 Saliency Tokens

2.5 Attention Score

TransMix: Attend to Mix for Vision Transformers (CVPR`2022)

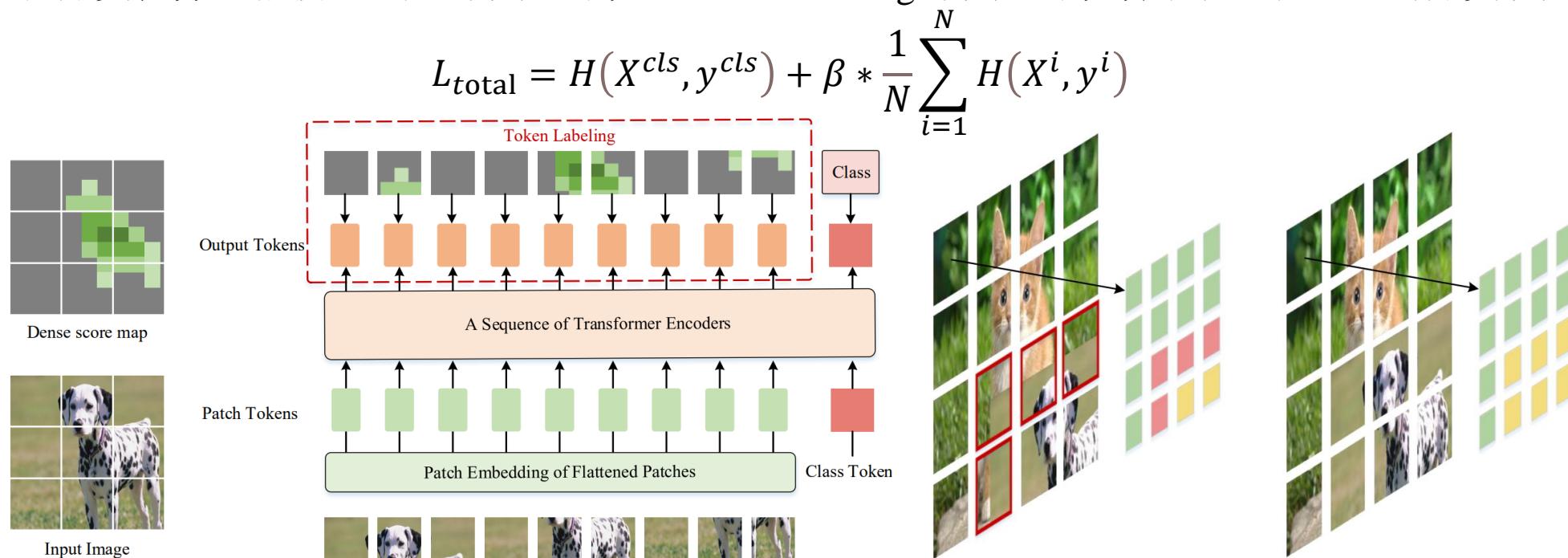
将ViTs框架应用在了mixup中，不同于CutMix使用面积来计算混合标签，TransMix使用ViTs中的attention作为各自的混合比例.



2.5 Attention Score

All Tokens Matter: Token Labeling for Training Better Vision Transformers (NIPS`2021)

之前使用ViTs做分类任务，往往会在开始加入一个类别 token，而最后经过模型相互关联，根据这个类别 token 为最后预测的结果。但是这样使用的信息非常少，因为在原图产生的很多特征 token 后仅使用最后一个类别 token，因此带来的监督效果也得到大幅下降。Token Labeling将ViTs中的tokens都打上标签用于进行分类任务。相较于原始的损失函数，Token Labeling的损失函数额外添加了token的分类损失。



2.5 Attention Score

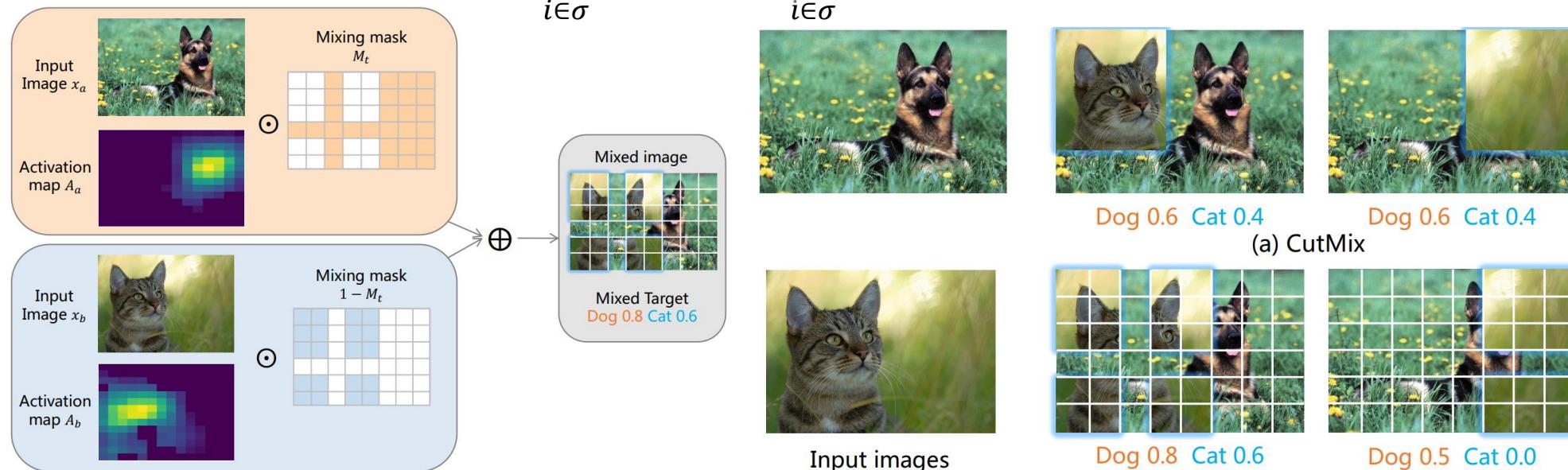
TokenMix: Rethinking Image Mixing for Data Augmentation in Vision Transformers (ECCV`2022)

利用Raw样本的attention score来进行标签比例的计算，再TokenMix中标签不一定总和为1。

具体来说：我们在得到Raw样本的attention score后进行normalization；创建一个随机的token mask M进行样本混合；通过M中每个token的attention score来计算标签的比例：

$$\tilde{x} = M_t * x_a + (1 - M_t) * x_b$$

$$\tilde{y} = \sum_{i \in \sigma} M_{ti} * A_{ai} + \sum_{i \in \sigma} (1 - M_{ti}) * A_{bi}$$

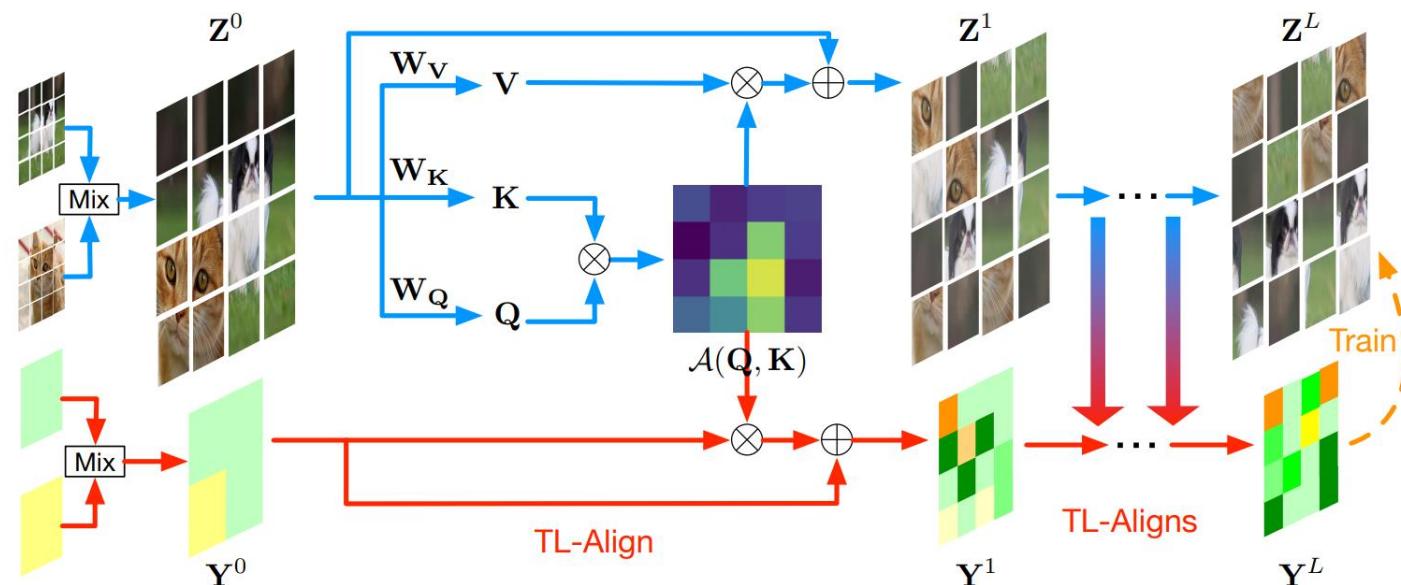


2.5 Attention Score

Token-Label Alignment for Vision Transformers (ICCV`2023)

利用混合样本进入ViTs时每一层的 Q^*K^T 的得分不仅与V算attention，也送入标签中。

标签先按tokens得到混合标签矩阵，再与A相乘和残差得到对齐标签，最后ViTs中的每一层都使标签进行对齐，得到最终的混合标签。

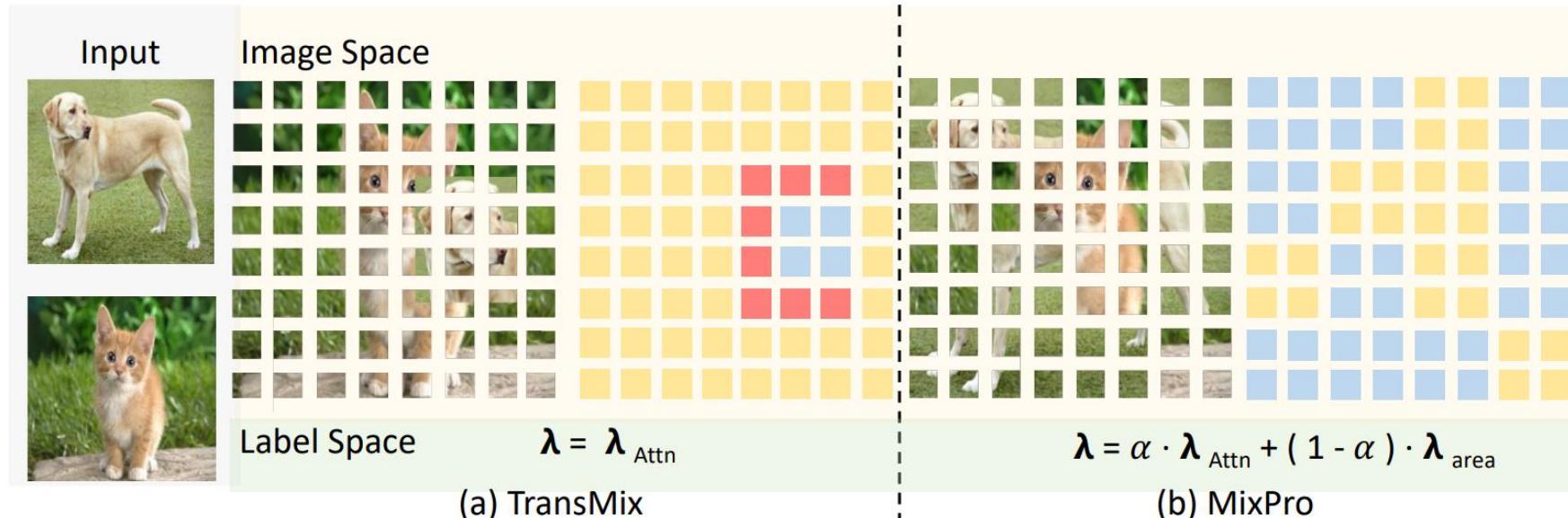


2.3 Optimization λ

MixPro: Data Augmentation with MaskMix and Progressive Attention Labeling for Vision Transformer (ICLR`2023)

MixPro改进TransMix中的俩个问题：1. 划分token是会产生一个token中“硬边界”问题；2. 前期ViTs中的注意力学的并不可靠。

对此，提出两种改进：重新划分token，提出MaskMix，现划分token，然后随机选择token变成mask；结合区域和注意力作为混合比例 λ ，使用一个渐进性因子 α 来平衡，前期更重视区域，随着ViT的能力增强，逐步重视注意力。



2.1 Loss Object

2.2 Optimize Calibration

2.3 Optimize λ

2.4 Generate Label

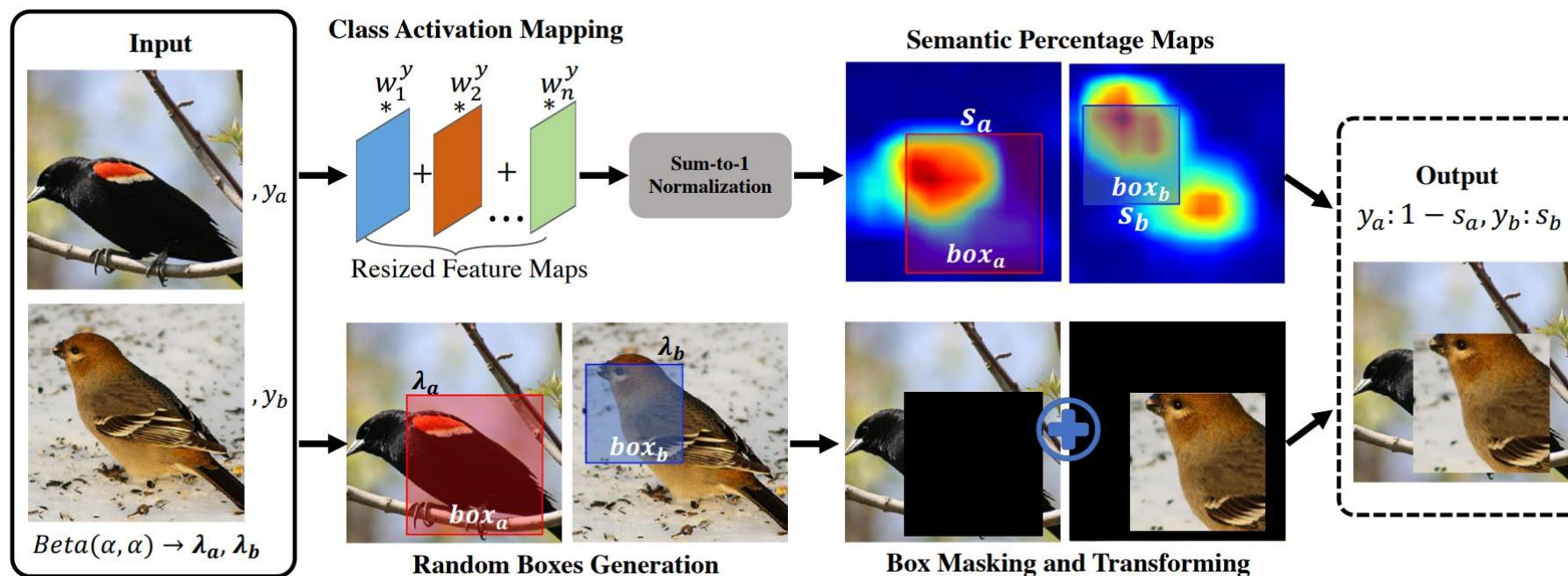
2.5 Attention Score

2.6 Saliency Tokens

2.6 Saliency Tokens

SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data (AAAI'2020)

SnapMix通过使用类激活映射(Class Activation Mapping, CAM)来得到Raw样本的显著性信息，再通过随机生成mask来进行混合得到混合样本。值得注意的是，标签是根据CMA得到的显著性信息来计算的，通过所得到mask所对应的样本位置的显著性信息来计算，这也就导致了SnapMix与之前提到的TokenMix一样，标签不在总和为1。

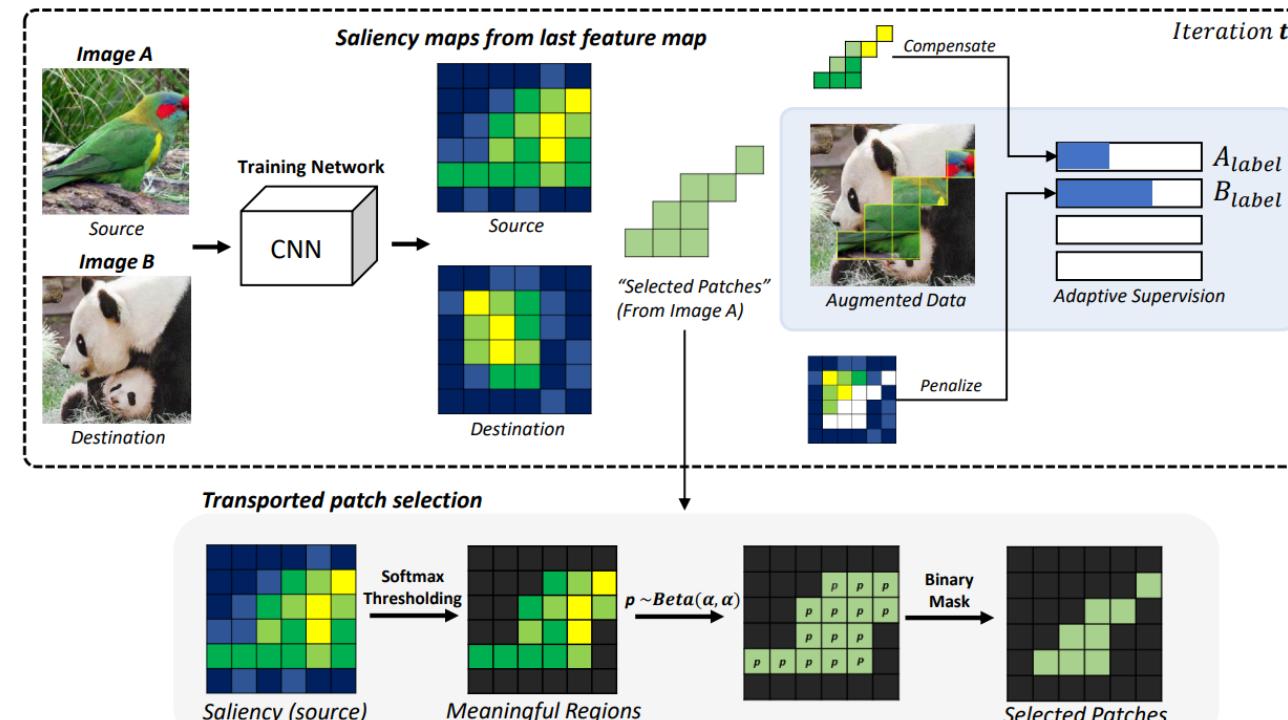


2.6 Saliency Tokens

Saliency Grafting: Innocuous Attribution-Guided Mixup with Calibrated Label Mixing (AAAI'2022)

Saliency Grafting通过CNN提取样本中的saliency map，将得到的saliency map经过一个softmax + 阈值判断后得到初步的二值化mask，阈值是由saliency map的均值得到。再通过一个相同维度的随机伯努利矩阵得到的P，得到最终的mask: $M = P*S$ 。

其中，标签的分配则是按照mask中的patch个数决定。



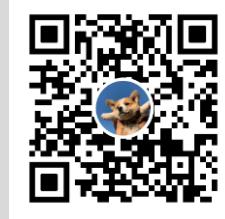


Thank you!

Google Scholar



GitHub



Homepage



Email: jinxin20001118@163.com, 158398730@qq.com

WeChat: xinxinxinxin_jin

Copyright by Xin Jin (金 鑫)

衷心感谢 西湖大学 李思远 的指导



Brief Description

- 这个PDF只是Mixup方法在监督领域中应用的简单描述，如果想要了解具体细节，还请查看原文；
- 一些方法其实做的很全面，不仅仅优化了样本混合的设计，也对标签、混合比例都进行改进，如 SMMix、SnapMix、MixPro等等。但在归类的时候为了减少重复的工作，我只归类到某一处；
- 每个方法的总结都是基于本人的理解，无法保证100%的正确。如有问题，欢迎联系我，我会加以改正；
- 后续会继续跟进Mixup方法在自监督，半监督以及其它下游任务的总结归纳；
- 感谢西湖大学李思远同学、刘梓丞同学、重庆工商大学朱泓宇同学的帮助；
- 如果想对Mixup方法了解、合作欢迎联系我！

I'm searching for a 2025
fall PhD position, if you are
interested, please feel free
to contact me !