SpliceMix: A Cross-scale and Semantic Blending Augmentation Strategy for Multi-label Image Classification

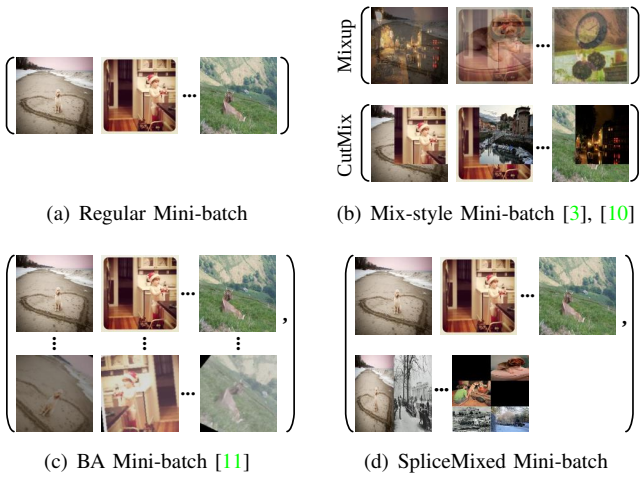
Lei Wang, Yibing Zhan, Leilei Ma, Dapeng Tao, Liang Ding, and Chen Gong, *Senior Member, IEEE*

***Abstract*—Recently, Mix-style data augmentation methods (*e.g.*, Mixup and CutMix) have shown promising performance in various visual tasks. However, these methods are primarily designed for single-label images, ignoring the considerable discrepancies between single- and multi-label images, *i.e.*, a multi-label image involves multiple co-occurred categories and fickle object scales. On the other hand, previous multi-label image classification (MLIC) methods tend to design elaborate models, bringing expensive computation. In this article, we introduce a simple but effective augmentation strategy for multi-label image classification, namely SpliceMix. The "splice" in our method is two-fold: *1)* Each mixed image is a splice of several downsampled images in the form of a grid, where the semantics of images attending to mixing are blended without object deficiencies for alleviating co-occurred bias; *2)* We splice mixed images and the original mini-batch to form a new SpliceMixed mini-batch, which allows an image with different scales to contribute to training together. Furthermore, such splice in our SpliceMixed mini-batch enables interactions between mixed images and original regular images. We also provide a simple and non-parametric extension based on consistency learning (SpliceMix-CL) to show the potential of extending our SpliceMix. Extensive experiments on various tasks demonstrate that only using SpliceMix with a baseline model (*e.g.*, ResNet) achieves better performance than state-of-the-art methods. Moreover, the generalizability of our SpliceMix is further validated by the improvements in current MLIC methods when married with our SpliceMix. The code is available at *https://github.com/zuiran/SpliceMix*.**

***Index Terms*—Multi-label learning, data augmentation, contextual bias, multi-scale learning, image classification.**

I. Introduction

As a fundamental task of computer vision, image classification (especially with single-label) has been well-studied from various aspects, such as network architectures [1], [2], data augmentation strategies [3], [4], and pre-trained models [5], [6]. However, some methods designed for single-label images are not always adaptive to the multi-label image classification (MLIC) task since a multi-label image usually contains multiple categories, which means more complicated scene and brings new challenges [7]–[9], *i.e.*, label dependency learning and small object recognition. In view of the fact that almost no data augmentation methods are especially designed for multi-label images according to their characters, it drives us to seek a simple but effective MLIC-adaptive augmentation strategy.



(a) Regular Mini-batch (b) Mix-style Mini-batch [3], [10]

(c) BA Mini-batch [11] (d) SpliceMixed Mini-batch

Fig. 1. Mini-batch comparisons between our SpliceMix and other methods.

In MLIC, label dependency has been widely studied in [12]–[16], which builds the co-occurred relationship of different categories for boosting recognition performance. Nonetheless, as claimed in [8], [17], the label co-occurrence pattern learned from the training set could be contextually biased to identify a category when its typical context is absent. On the one hand, the semantic context (*i.e.*, co-occurred relationship of categories) is crucial for improving MLIC. On the other hand, it may mislead the model in the scene of a category occurring in its unseen context. It is tricky to make a trade-off between learning label dependency and reducing co-occurred bias.

To some extent, Mix-style data augmentation methods naturally reduce the co-occurred bias. They usually generate a new image via mixing two images and their labels, which randomly produces a context-agnostic scene while losing the co-occurred relationship of categories. Two representative Mix-style methods, Mixup [3] and CutMix [10], have drawn lots of attention and their variants [18]–[20] have shown huge potential for many visual tasks recently. However, these methods are mainly

L. Wang and C. Gong are with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Jiangsu Key Laboratory of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China (e-mail: {lei_wang, chen.gong}@njust.edu.cn).

Y. Zhan and L. Ding are with the JD Explore Academy, Beijing 100000, China (e-mail: zhanyibing@jd.com; liangding.liam@gmail.com).

L. Ma is with the School of Computer Science and Technology, Anhui University, Heifei 230601, Anhui, China (e-mail: xiaoleilei1990@gmail.com).

D. Tao is with the FIST LAB, School of Information Science and Engineering, Yunnan University, Kunming 650091, Yunnan, China (e-mail: dapeng.tao@gmail.com).

Corresponding author: Chen Gong.

designed for single-label images, neglecting the considerable gap between single- and multi-label images. As aforementioned, they fail to capture label dependency. Moreover, the small objects are easy to be messed in Mixup or lost in CutMix. Although some variants extract salient object regions of an image to paste to another image that may not suffer from the above pains, we want to claim that these extracted regions could be unreliable and they require additional knowledge [21] or complicated design [18], [22].

In this article, we introduce a simple but effective, MLIC-adaptive Mix-style augmentation strategy, namely SpliceMix. As shown in Fig. 1, beyond previous Mix-style methods, the proposed SpliceMix constructs a new mini-batch with the original mini-batch and a few mixed samples, which is similar to batch augmentation (BA) [11] while requiring a smaller batch size. It is a specialized design for the MLIC task. Keeping the original mini-batch takes charge of leaning label dependency and generating mixed samples is conducive to reducing co-occurred bias. Both the two are indispensable in SpliceMix for MLIC, which is far more than a linear combination of image data [3]. An analysis of label dependency and co-occurred bias in a baseline using previous Mix-style methods and our SpliceMix is given in Sec. IV-G1. The mixed sample consists of several downsampled images in the form of a grid without the risk of missing small objects. It blends image semantics and produces an unusual scene for potential co-occurrences of both different objects and backgrounds-objects. If only mixed samples are utilized for training, there will exist two issues: *1)* The large resolution discrepancy between downsampled training images and testing images leads to poor inference performance; *2)* Chaotic category co-occurrences in mixed samples cut off the learning of label dependency. Therefore, we keep the original mini-batch and splice it with mixed samples. Besides learning label dependency, such splice also brings two advantages: *1)* At least two object scales between regular images in the original mini-batch and downsampled images in mixed samples are built, contributing to cross-scale learning for recognizing small objects; *2)* The regular images and their downsampled counterparts in the same SpliceMixed mini-batch enable interactions between the two, allowing us to design some SpliceMix variants flexibly. As a demonstration, we provide a non-parametric design based on consistency learning, i.e., SpliceMix-CL, to show the potential of extending SpliceMix.

Our contributions can be summarized as:

- To our best knowledge, the proposed SpliceMix is the first data augmentation strategy designed for MLIC. SpliceMix is not only simple yet effective, but also orthogonal to existing MLIC methods, which can boost these methods remarkably.
- A novel splice strategy is proposed that augments the mini-batch and samples simultaneously. Such splice enables cross-scale training and interactive learning between the original mini-batch and the mixed samples.
- We offer a non-parametric, consistency learning-based extension to show the potential of extending SpliceMix, where the fine knowledge of regular images is leveraged to learn better representation for mixed images.
- The two proposed methods achieve superior performance on several popular MLIC tasks and data sets. We also conduct comprehensive analysis experiments to clarify the effectiveness of the proposed SpliceMix and SpliceMix-CL.

II. Related Works

A. Multi-label Image Classification

Multi-label image classification (MLIC) is a challenging computer vision task and has attracted lots of attention. Existing MLIC methods prefer designing elaborate models to capture attention regions or label dependencies.

Discovering attention regions in a multi-label image is helpful to predict present categories or generate proposal candidates. Spatial Regularization Network (SRN) [23] learns weighted spatial attention maps with only image-level supervision and used multi-layer convolution to estimate class confidences from such attention maps. Different from SRN, [7] reuses classifier weights to obtain class-specific attention for discovering spatial discriminative regions. To generate informative proposal candidates, [24]–[26] utilize spatial attention to locate object regions. The difference among them is that [24], [26] extract local region features from feature maps directly instead of feeding local image regions into CNN again used in [25].

For label dependency-based methods, some techniques, *e.g.*, Recurrent Neural Network (RNN) [27], Graph Neural Network (GNN) [28] and Transformer [29], usually are utilized to model label co-occurred correlation. CNN-RNN [30] learns joint image-label embedding and builds high-order label dependency via long short term memory recurrent neurons [31]. Beneficial from the excellent ability of modeling correlation GNN possesses, a series of methods [13], [15], [16] are proposed. [13], [32] consider label co-occurrence statistics as the adjacency matrix and then exploit GNN to map the label graph to class-dependent classifier or label embeddings. In view of poor model generalizability of label statistics-based graph, [12], [33] introduce dynamic graph to eliminate co-occurrence bias from the training set. In [14], Transformer Decoder is utilized to learn complex dependencies between feature maps and label embeddings. [34], [35] leverages both Encoder and Decoder in Transformer to learn intra-feature and feature-label correlations.

Two recent works [8], [36] indicate that exploiting label dependency may cause contextually biased recognition. Coincidentally, some previous MLIC researches [7], [12] suggest that the label co-occurrence information from limited training data is not enough for MLIC, which could lead to over-fitting. Therefore, we propose the SpliceMix to mitigate co-occurred bias via generating unusual scenes where the semantics of several images are blended and the learned label co-occurrence pattern is ameliorated.

B. Mix-style Augmentation

Recently, Mix-style augmentation methods have shown promising performance in various single-label visual tasks, such as image classification [10], [19], super-resolution [37]

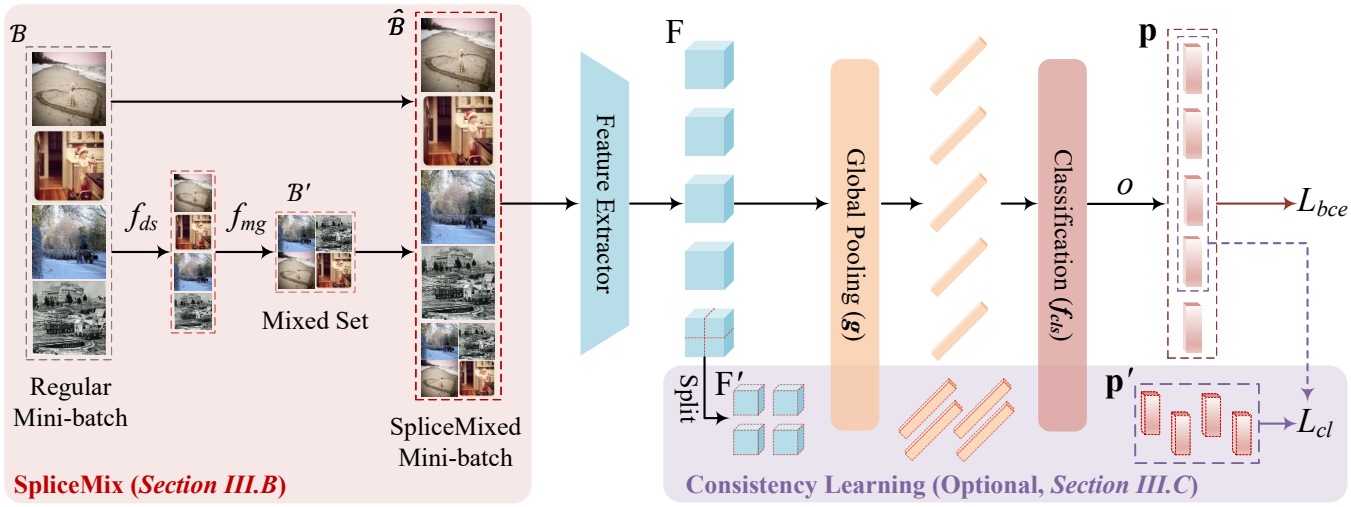


Fig. 2. Overview of SpliceMix and SpliceMix-CL with a 2×2 grid strategy. "$\xrightarrow{\bullet}$" denotes a function operation and "$\dashrightarrow$" denotes calculation without backpropagation. We simply introduce SpliceMix with the mini-batch samples whose size is 4. For generating unusual scene and preserving intact objects, SpliceMix makes a grid of downsampled images randomly sampled from the regular mini-batch to form the mixed set. The final SpliceMixed mini-batch is obtained by combining the regular mini-batch and mixed set. Furthermore, we offer a consistency learning-based design to show the flexible expansibility of our SpliceMix.

and semantic segmentation [20]. As a founding one, Mixup [3] generates a new image by linearly combining two images and their labels. Since then, a bunch of Mix-style methods have been proposed, where CutMix [10] is a representative work that joints Mixup and Cutout [38] via replacing local regions with a patch from another image. Similar to the way of cut-and-paste in CutMix, some methods insert new patches to target images via leveraging smooth transition [39], attention maps [40], saliency information [18], [22]. Considering limited benefits of CutMix for visual Transformers, [19] mixes two images at token-level and generates the mixed label from content-based activations of a teacher network. [37] extends CutMix to the image super-resolution task where images with high resolution and low resolution consist of sample-pairs. [20] exploits predicted object boundaries to mix two unlabeled images for semi-supervised semantic segmentation.

Although Mix-style methods behave well in diverse single-label visual tasks, they are not suitable to MLIC because of more complicated image semantic and more categories in an multi-label image. For instance, cut-and-paste methods fail to assign the weight of labels from two images since a local patch cannot represent a multi-label images. In view of this, we aim to design a MLIC-adaptive augmentation strategy and exploit entire images to attend to mixing, which preserves all objects and results in reliable mixed labels.

C. Batch Augmentation

Different from previous augmentation methods, Batch Augmentation (BA) [11] increases the mini-batch size by replicating samples within the same mini-batch with different data augmentations, which accelerates model convergence and shows good generalization. The original BA is proposed to improve performance of training with large mini-batch size. [41] empirically shows that BA also works well for small mini-batch size. Whereas, BA still requires augmenting each image repeatedly for building a mini-batch, which is demanding. In our SpliceMix, we slightly increase the mini-batch size with mixed samples, each of which consists of multiple downsampled regular images. Even with only a few mixed samples, SpliceMix can significantly boost performance. Same to [41], we also allow fixed mini-batch size, *i.e.*, keeping the same size to regular mini-batch via reducing regular samples to save space for mixed images.

III. Methodology

This section will detail the proposed SpliceMix. Benefiting from our generated SpliceMixed mini-batch that enables interactions between original samples and their mixed samples, we also give a non-parametric extension based on consistency learning, namely SpliceMix-CL, for further improvement.

A. Preliminary

The objective of a MLIC baseline (*e.g.*, ResNet [1]) is to minimize binary cross entropy loss $\mathcal{L}_{bce}$ between the true label distribution and the predicted distribution. In training time, the label dependency can be learned implicitly, which is reflected in the classifier weights. Next, we will present how a baseline can learn the implicit label dependency.

For a training sample $(\mathbf{x}, \mathbf{y})$, its prediction is $\mathbf{p} = o(\mathbf{z}) = o(\mathbf{w}^T f_\theta(\mathbf{x}))$ where f_θ is the feature extractor with global pooling, $\mathbf{w} = [\mathbf{w}_1; \mathbf{w}_2; \ldots; \mathbf{w}_L]^T$ denotes the linear classifier for L classes, and o denotes the sigmoid activation. The gradients of the loss for the classifier related to the i-th class is

$$\begin{aligned} \frac{d\mathcal{L}_{bce}}{d\mathbf{w}_i} &= \frac{\partial \mathcal{L}_{bce}}{\partial \mathbf{p}_i} \frac{\partial \mathbf{p}_i}{\partial \mathbf{z}_i} \frac{\partial \mathbf{z}_i}{\partial \mathbf{w}_i} \\ &= (\mathbf{p}_i - \mathbf{y}_i) f_\theta(\mathbf{x}) \ . \end{aligned} \tag{1}$$

As we can see that the gradient orientation of $\mathbf{w}_i$ depends on the difference between the i-th class prediction $\mathbf{p}_i$ and label $\mathbf{y}_i$,

since all classifiers share the same global feature $f_\theta(\mathbf{x})$. The value of $(\mathbf{p}_i - \mathbf{y}_i)$ will be less than 0 if $\mathbf{y}_i = 1$, vice versa. Obviously, for each sample, the classifier of positive classes will be updated with more similar orientation than negative classes by gradient descent. Hence, the learned classifiers of relevant classes are similar to each other and are different from their distant classes, which indicates that the implicit label dependency is learned.

Compared to existing Mix-style methods that produce unreliable label correlations for each sample, the proposed SpliceMix keeps the original mini-batch whose label correlations are accurate for mining label dependency. A visual comparison on learned label dependency of the baseline (*i.e.*, ResNet-101 [1]), Mixup, CutMix, and SpliceMix is given in Fig. 7.

B. SpliceMix

Given a mini-batch $\mathcal{B} = \{(\mathbf{x}_i, \mathbf{y}_i)|i \in [B]\}$, randomly sampling B points from the training set, where $\mathbf{x}$ and $\mathbf{y}$ are an image and its label, respectively. As illustrated in Fig. 2, SpliceMix will generate a new mixed sample set $\mathcal{B}'$ from $\mathcal{B}$, and then concatenate the original mini-batch and generated set to form the final mini-batch $\hat{\mathcal{B}}$ for training, *i.e.*, $\hat{\mathcal{B}} = \mathcal{B} \cup \mathcal{B}'$. Before turning sights to $\mathcal{B}'$, we would like to clarify some terms used in the rest to avoid confusion. We call the original mini-batch $\mathcal{B}$ as the regular batch whose samples $(\mathbf{x}, \mathbf{y})$s are regular ones, the new mixed sample set $\mathcal{B}'$ as the mixed set whose samples $(\mathbf{x}', \mathbf{y}')$s are mixed ones and the generated final mini-batch $\hat{\mathcal{B}}$ as the SpliceMixed batch.

Mixture of images: The proposed SpliceMix aims to generate new images, preserving complete objects and producing an unusual scene for present categories. To achieve this, we make a grid of chosen regular images from $\mathcal{B}$ to form the mixed image. Keeping the resolution consistent with regular images, the chosen images are downsampled before griding, which brings a benefit of enabling the model to learn multi-scale information from regular images and their mixed ones. Let f_{mg} denote the operation of making a grid and f_{ds} denote the operation of downsampling. We can express the mixed image as

$$\mathbf{x}' = f_{mg}(\{f_{ds}(\mathbf{x}_i)|i \in \Omega\}) \ , \tag{2}$$

where Ω is an index set of the regular batch that accounts for randomly selecting regular images to form the mixed image, *e.g.*, the cardinality of Ω will be 4 if we want to obtain a mixed image with a 2×2 grid of regular images.

Mixture of labels: Different from existing Mix-style methods that mix labels of two images via linear combination, the proposed SpliceMix sets the class label to be true if this class is present in the mixed image, formulated by

$$\mathbf{y}' = \cup_{i \in \Omega} \mathbf{y}_i \ . \tag{3}$$

The mixed label keeps multi-hot same to regular samples while containing more categories, which enriches the label diversity and is more suitable for MLIC training than the soft label used in existing Mix-style methods. Our SpliceMix method can be implemented readily in several lines of code. A PyTorch [42] implementation of 2×2 mixed strategy is presented in Fig. 3.

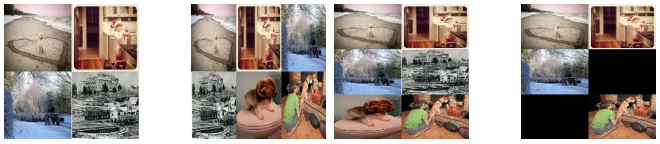
```
import random, torch
from torch.nn.functional import interpolate as f_ds
from torchvision.utils import make_grid as f_mg

def SpliceMix(X, Y):
# X: (Batch size, Channels, Height, Width)
# Y: (Batch size, classes)
    B, C, H, W = X.shape
    Omega = random.sample(range(B), B//4*4)
    X_ds = f_ds(X[Omega], size=(H//2, W//2),
        mode='bilinear', align_corners=True)
    X_ = f_mg(X_ds, nrow=2, padding=0)
    X_ = X_.split(H, dim=1)
    X_ = torch.stack(X_, dim=0)
    Y_ = Y[Omega].view(B//4, 4, -1).sum(1)
    Y_[Y_>0] = 1

    X_hat = torch.cat((X, X_), dim=0)
    Y_hat = torch.cat((Y, Y_), dim=0)
    return X_hat, Y_hat
```

Fig. 3. PyTorch implementation of SpliceMix with a 2×2 grid strategy.

Since we use the global images rather than local regions for mixing, there is almost no information depletion of present objects (except their resolutions are reduced) that means the label of each mixed image is reliable. More importantly, the proposed SpliceMix puts objects into a rare or complex scene, *i.e.*, mixed scenes in our generated image. The model will build correlations among objects who may be co-occurred unusually and between objects and backgrounds where objects may be seldomly present. The mixed samples blend the image semantics and give the model more chances to see these potential scenes, alleviating the semantic and contextual bias and enhancing the model's generalizability. On the other hand, we agree with the importance of inherent correlations from the training set and preserve the regular batch in our SpliceMixed batch in a BA-like way. The correlations on relevant objects still can be learned. In brief, our SpliceMixed batch consists of the regular batch and mixed set for the training time, from the former of which common label correlations are built and from the latter uncommon label correlations are reinforced. Both of them boost the final recognition and generalization performance.

Setting of grids: As stated in Eq. (2), a mixed image is made from chosen regular images via two operations, *i.e.*, f_{mg} and f_{ds}, where the downsampling operation f_{ds} is to resize the chosen regular images that lets the resolution of a mixed image match with that of regular images. For instance, given 4 regulars image whose resolutions are 448×448 for generating a mixed image with a 2×2 grid, we firstly resize them to 224×224 and then make a 2×2 grid of them to form the mixed image. The reduced scale f_{ds} depends on which kind of the grid used for generating a mixed image. The grid number of different mixed images can be various. In other words, our SpliceMix allows to mix images with different combinations of grids to make up a mixed set, such as mixed images with grids of 1×2, 2×2, 2×3, *etc*.

Here, we focus on the grid strategy introduced for SpliceMix. Several feasible grids are illustrated in Fig. 4. Accordingly, we can define a function family $\mathcal{F}_{mg}$ for sampling



2×2 2×3 3×2 $3 \times 2 - 2$

(a) Sym. grids (b) Asym. grids (c) Grids w/ dp

Fig. 4. Illustration of several feasible grid strategies for SpliceMix. Firstly, we assume that the regular images are square. Then, we can summary these grids as three types: (a) Symmetric grids with the same rows and columns; (b) Asymmetric grids with distinct rows and columns; (c) Grids with dropout where some sub-images are discarded randomly.

f_{mg}, that is

$$\mathcal{F}_{mg} = \{f^{r\times c} | r \in [G_r], c \in [G_c]\} \ , \tag{4}$$

where we rewrite f_{mg} to $f^{r\times c}$, an operation of making a $r \times c$ grid, and G_r, G_c denote the maximal rows and columns, respectively. Next, three types of the grid shown in Fig. 4 are discussed:

1) **Symmetric grids:** Sub-images in a symmetric grid are proportionally scaled from their regular images. Due to the same columns and rows in a symmetric grid, $f^{r\times c}$ is equal to $f^{c\times r}$. The number of grid dominates the complexity of the generated image. A more complicated mixed image needs more regular images whose resolution will be downsampled much more.
2) **Asymmetric grids:** The shape of sub-images in an asymmetric grid is squashed compared with their regular ones, which can be viewed as an additional transform besides rescaling. For a $f^{r\times c}$, if we keep the columns c larger than the rows r (*e.g.*, the left figure in Fig. 4 (b)), the model may learn to recognize stretched objects well while losing its generalizability to flatten objects. To avoid this, we throw a coin with a flipped probability of 0.5 to decide which of $\{f^{r\times c},\ f^{c\times r}\}$ will be adopted under this setting.
3) **Grids with dropout:** The goal of mixed images in the form of grids is to blend the image semantics and generate a new, unusual scene. Whereas, the generated scene may be over-complicated due to a large grid number, *i.e.* too many regular images attend to a mixed image. Therefore, we exploit a simple dropout mechanism to randomly mask a part of sub-images and corresponding labels in a mixed sample. After this, the complexity of a mixed image can be reduced, meanwhile the diversity of training samples is further enhanced.

C. SpliceMix-CL

The proposed SpliceMix allows interactions between sub-images from a mixed image and their regular versions, which means that the fine knowledge learned from regular images can be utilized to guide the model to learn better representation about coarse sub-images. In fact, there exist many techniques to help us learn consistent knowledge between regular images and sub-images, such as logits-based [43], [44] or feature-based [45], [46] knowledge distillation methods and class decoupling-based [12], [32] or attention-based [47] MLIC methods, where the knowledge distillation-aware methods are used for consistency learning and the MLIC-aware methods are used for task-specific knowledge extraction. Although these techniques may show promising performance with our SpliceMix, an elaborate model design is beyond our current study and we leave it to future work. Here, we offer a simple, non-parametric idea based on consistency learning [48], namely SpliceMix-CL, to show the potential of extending SpliceMix.

Assume that F is the last feature map extracted from a feature extractor, *e.g.*, CNNs [1], [49], g is a global pooling operation, f_{cls} is a linear classifier, o is sigmoid activation. Then, we can obtain the label prediction $\mathbf{p}$ of an input image, that is $\mathbf{p} = o(f_{cls}(g(\mathrm{F})))$. In a SpliceMixed batch, the classification loss, *i.e.*, binary cross entropy (BCE) loss, is

$$\mathcal{L}_{bce} = \sum_{i}^{|\hat{\mathcal{B}}|} -\mathbf{y}_i log(\mathbf{p}_i) - (1-\mathbf{y}_i) log((1-\mathbf{p}_i)) \ , \tag{5}$$

where log denotes the element-wise logarithmic function. For convenience, we slightly abuse $(\mathbf{x}, \mathbf{y})$ to denote any sample from $\hat{\mathcal{B}}$ and keep vector form of loss, each element of which is a class-specific loss. A scalar form of loss can be obtained by summing all class-specific loss values.

For each sub-image in a mixed image, we obtain its feature map F' via splitting F according to the grid form of the mixed image (see Fig. 2). By the way, we can simply exploit f_{mg} to recover F from F's: $\mathrm{F} = f_{mg}(\{\mathrm{F}'_i | i \in \Omega\})$, where the subscript i indicates the i-th image in $\mathcal{B}$ attending to the mixed image and we reuse such subscript to indicate the feature map of a sub-image since a regular image attends to mixed images at most once. Consequently, the prediction $\mathbf{p}'$ of a sub-image can be calculated by $o(f_{cls}(g(\mathrm{F}')))$. For consistency learning between sub-images and their regular images, we utilize the prediction distribution $\mathbf{p}$ from a regular image to supervise the output of its corresponding sub-image, whose objective can be formulated by

$$\mathcal{L}_{cl} = \sum_{i}^{|\mathcal{B}'|} \sum_{j\in\Omega_i} -\bar{\mathbf{p}}_j log(\mathbf{p}'_j) - (1-\bar{\mathbf{p}}_j) log(1-\mathbf{p}'_j) \ , \tag{6}$$

where $\bar{\mathbf{p}}$ is a copy of $\mathbf{p}$ without backpropagation and we reuse the subscript j to $\mathbf{p}'$ that is obtained from the sub-image of the i-th mixed image corresponding to j-th regular image (the subscript i is omitted for convenience).

Finally, the total training objective of SpliceMix-CL is

$$\mathcal{L} = \mathcal{L}_{bce} + \mathcal{L}_{cl} \ . \tag{7}$$

Although it seems that the consistency learning loss is only imposed on the pair of regular images and their sub-images, we claim that it is not simply equal to learning cross-scale knowledge. The better representation of sub-images helps the model discover possibly overlooked objects present in a mixed image, which is beneficial for blending image semantics and improving our SpliceMix.

TABLE I
COMPARISONS OF SPLICEMIX AND PREVIOUS MIX-STYLE METHODS. DENOTATIONS HIGHLIGHTED IN PURPLE AND TEAL ARE THE ADVANTAGES AND DRAWBACKS FOR MLIC, RESPECTIVELY.

Method	Mosaic	Mixup	CutMix	SpliceMix
Semantic Blending	✓	✓	✓	✓
Cross-scale Training	✗	✗	✗	✓
Label Correlation	✗	✗	✗	✓
In-batch Interaction	✗	✗	✗	✓
Bounding Box	✓	✗	✗	✗

D. Connection to Previous Methods

Here, we will discuss the relation and difference between SpliceMix and several previous Mix-style methods including Mosaic [50], Mixup [3], and CutMix [10]. Among them, Mosaic is a popular data augmentation method for object detection. The comparison results are given in Table I where all compared items, except for "In-batch Interaction" and "Bounding Box", are important to MLIC. The "In-batch Interaction" allows us to extend SpliceMix flexibly. The "Bounding Box" is essential to decide the annotation of a mixed image in Mosaic. However, it is not available in MLIC. In terms of the mixed images, Mosaic can be viewed as a special case of SpliceMix using bounding boxes for object detection. As we can see from Table I, the only similar point between Mosaic and our SpliceMix is that they both blend multiple regular images, which is also similar to Mixup and CutMix. Beyond this, the proposed mixing strategy is designed to cover the shortages of existing methods when tackling the MLIC task.

We want to claim the key role of the splice between the original batch and the mixed set for MLIC that is one of the main contributions and has not been considered in previous methods. The proposed splice strategy kills three birds with one stone. Firstly, it is a practical solution for steady training since just using mixed samples generated by SpliceMix results in worse performance than the baseline (see Table XI). Secondly, it enables the cross-scale training, which is favorable for recognizing small objects. Thirdly, it allows us to build consistency learning between regular images and their downsampled counterparts for performance improvement. With the splice strategy of an original batch plus a few mixed samples, SpliceMix can improve the baseline by 1.4% and 1.3% in mAP on MS-COCO [51] and Pascal VOC 2007 [52], respectively. Note that the splice of an original batch plus other Mix-style methods does not always achieve satisfactory performance, which will be discussed in Sec. IV-G5.

IV. EXPERIMENTS

In this section, we first introduce experimental settings including evaluation metrics and our training details. Next, computational time are analyzed and the proposed methods are verified via plentiful experiments. Finally, we conduct performance analyses to discuss the proposed methods detailedly.

A. Experimental Settings

Evaluation metrics: We follow previous works [12], [13], adopting mean of Average Precision (mAP) as the primary metric. Overall precision (OP), recall (OR), F1-score (OF1) and per-class precision (CP), recall (CR), F1-score (CF1) and their top-3 versions are also utilized to evaluate our results. All metric are in %. For all metrics except mAP, we set a category to be positive if its prediction is larger than 0.5, vice versa.

Training details: For the proposed methods and compared methods, we use ResNet-101 [1] as the base model where the last global average pooling is replaced with global max pooling that is a common setting [12], [13] to achieve good results for MLIC. We choose SGD as our optimizer with momentum of 0.9 and weight decay of 10^{-4}. The output dimension of the last fully connected (fc) layer in ResNet-101 is changed according to the class number of various data sets. We set the learning rate of all layers except the last fc layer to be one-tenth of the given learning rate. During training, we adopt the data augmentation suggested in [13], [33], *i.e.*, the input image is randomly cropped and resized to 448 × 448 with horizontal flips. We train our model for 80 epochs and decay the learning rate by a factor of 0.1 at 40-th and 60-th epoch. For our SpliceMix, we do not seek the optimal grid strategy deliberately and randomly sample f_{mg} from $\mathcal{F}_{mg} = \{f^{1\times2}, f^{2\times2}, f^{2\times3}\}$ with a probability of 0.3 to use a random dropout from 1 to $r \times c - 1$ per batch. The cardinality of the mixed set is set to a quarter of the regular batch size, *i.e.*, $|\mathcal{B}'| = \frac{1}{4}|\mathcal{B}|$. All experiments are implemented by PyTorch [42]. The hardware environment is based on Ubuntu 16.0 with a single NVIDIA GeForce RTX 3090 GPU.

B. Comparisons on Computational Time

We compare the average computational time per iteration of the baseline, previous Mix-style methods, and the proposed SpliceMix. For fairness, all methods keep the same batch size, *i.e.*, 32 on MS-COCO. The result is reported in Table II. The proposed SpliceMix takes the least time for a training iteration.. Taking batch size 32 as an example, SpliceMix reads 28 images and processes them to generate 4 mixed samples, so lower I/O throughput in a batch is required. In contrast, Mosaic [50] has to access 4 times more images than the batch size to generate new samples, which is inordinately time-consuming. Note that in our observation, SpliceMix can obtain higher mAP when the same number of training iterations is reached, compared to other methods. Hence, it is unprejudiced to keep the same final batch size to others for comparisons. The proposed SpliceMix-CL needs feature-level operations with an additional consistency loss. These do not increase excessive computational time and SpliceMix-CL still surpasses other Mix-style methods.

C. Multi-label Image Classification

We evaluate the proposed SpliceMix and SpliceMix-CL on two MLIC tasks, *i.e.*, regular MLIC and MLIC with missing labels, where common data sets MS-COCO 2014 [51] and Pascal VOC 2007 [52] are adopted.

TABLE II
COMPARISONS ON COMPUTATIONAL TIME PER ITERATION USING VARIOUS METHODS.

Method	Baseline [1]	Mixup [3]	CutMix [10]	Mosaic [50]	SpliceMix	SpliceMix-CL
Elapsed Time (ms)	257.6	259.6	264.7	273.7	253.8	258.2

TABLE III
COMPARISONS WITH STATE-OF-THE-ART METHODS ON MS-COCO. THE BEST AND SECOND BEST RESULTS (EXCEPT THE METHODS MARKED BY "*") ARE HIGHLIGHTED IN PURPLE AND CYAN, RESPECTIVELY. THE SAME HIGHLIGHTED WAY IS USED FOR THE REMAINING TABLES. "W/ BBX" IN MOSAIC DENOTES ADDITIONAL BOUNDING BOX ANNOTATIONS, SAME TO IV.

Method	mAP	All						Top 3					
		CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
DSDL [53]	81.7	84.1	70.4	76.7	85.1	73.9	79.1	88.1	62.9	73.4	89.6	65.3	75.6
CSRA [7]	83.5	84.1	72.5	77.9	85.6	75.7	80.3	88.5	64.2	74.4	90.4	66.4	76.5
SST [54]	84.2	86.1	72.1	78.5	87.2	75.4	80.8	89.8	64.1	74.8	91.5	66.4	76.9
CCD [8]	84.0	87.2	70.9	77.3	88.8	74.6	81.1	89.7	63.9	72.9	92.0	66.5	77.2
IDA-R101 [36]	84.3	-	-	78.5	-	-	81.1	-	-	73.6	-	-	77.3
Baseline [1]	83.0	85.0	72.0	77.9	86.9	75.2	80.3	89.0	64.1	74.5	90.7	66.3	76.6
Mixup [3]	83.7	84.9	72.8	78.4	86.5	75.7	80.7	89.3	64.8	75.1	91.1	66.5	76.9
CutMix [10]	83.5	84.6	72.9	78.3	86.2	75.5	80.5	89.0	64.8	75.0	90.9	66.3	76.7
ResizeMix [55]	83.1	86.1	71.0	77.8	88.0	73.4	80.1	90.1	64.0	74.8	91.7	65.3	76.3
RecursiveMix [56]	83.4	86.9	70.4	77.8	88.8	72.6	79.9	90.3	63.8	74.7	92.3	64.8	76.1
BA [11]	83.3	84.8	72.5	78.2	85.8	75.6	80.4	89.0	64.4	74.7	90.4	66.5	76.7
Mosaic [50] w/ bbx	82.7	76.3	76.8	76.5	79.3	80.1	79.7	82.6	66.1	73.5	86.3	68.2	76.2
SpliceMix	84.4	84.8	74.2	79.1	86.2	77.0	81.3	89.2	65.6	75.6	91.0	67.2	77.3
SpliceMix-CL	84.9	87.4	73.2	79.7	88.2	76.3	81.8	90.7	65.1	75.8	92.1	67.1	77.6
SpliceMix*	85.8	85.1	76.6	80.6	86.4	78.7	82.3	89.6	67.3	76.9	91.4	68.2	78.1
SpliceMix-CL*	86.4	85.5	76.9	81.0	87.0	79.0	82.8	89.8	67.6	77.1	91.7	68.6	78.5

1) Regular Multi-label Image Classification

MS-COCO 2014: Microsoft COCO 2014, short for MS-COCO, is a widely used benchmark for MLIC. There are 82,081 images in the training set and 40,137 images in the validation set with total 80 common objects categories. Since labels of the test set are unavailable, we compare our methods to other methods on the validation set. The learning rate for MS-COCO is set to 0.05 and the batch size is 32. The comparison results are presented in Table III. It is worth noting that SpliceMix can achieve comparable performance in ImageNet [57] classification. Hence, we also report the results of the two proposed methods adopting ImageNet pre-training based on SpliceMix and mark them by "*", same for Tables IV and V.

To ensure the fairness for compared methods, we make a grid search of the α in Mixup [3] and CutMix [10] from 0.1 to 1 with a step of 0.2. We find that $\alpha = 0.5$ is optimal for the two methods on both MS-COCO and Pascal VOC 2007. For BA, we choose the times of replicating samples (*i.e.*, M in [11]) from $\{2, 4, 6\}$ and find that replicating samples twice achieves the highest performance. For Mosaic [50], the bounding box annotations are used additionally. In its generated images, we drop object category whose areas of bounding boxes are less than half of those in the original images. Note that bounding boxes are not available in the MLIC task. As shown in Table III, our two methods have the best performance in most metrics. In compared MLIC methods, CCD [8] and IDA-R101 [36] are designed for removing co-occurred bias which obtain high mAP. With the similar objective, we blend multiple images to generate an unusual scene for existing categories. Although our SpliceMix is just a simple augmentation method, it achieves superior performance to two contextually debiased models. Moreover, our SpliceMix-CL also shows an excellent improvement, compared to SpliceMix and current advance methods, which suggests the large potential of extending our SpliceMix. From the last two rows of Table III, we can see that SpliceMix-based ImageNet pre-training performs well in the MLIC task.

Although Mosaic has achieved huge success in object detection, it performs poorly on MS-COCO. In the object detection task, Mosaic puts lots of objects into a mixed image for training an object detector that is encouraged to predict more positive samples. As a result, the positive-negative imbalance issue in object detection can be mitigated. However, it is not the primal issue in MLIC. From Table III, we can see that Mosaic obtains the highest CR and OR while the lowest CP and OP. The proposed SpliceMix alleviates this via splicing the regular batch and mixed images, leading to good performance for MLIC.

Pascal VOC 2007: Pascal Visual Object Classes Challenge (VOC 2007) is the most used data set in MLIC. It consists of 20 object categories from 9,963 images. We use the *trainval* set (5,011 images) for training and the *test* set (4,952 images) for evaluation. The learning rate for Pascal VOC 2007 is set to 0.01 and the batch size is 16. Considering that some current MLIC methods tend to train Pascal VOC 2007 not only with ImageNet pre-training but also with MS-COCO pre-training, where the latter can achieve better performance empirically, we give results on pre-training from ImageNet and MS-COCO, respectively.

The comparison of our methods and other methods is listed in Table IV. We can see that our SpliceMix and SpliceMix-CL

TABLE IV
COMPARISONS WITH STATE-OF-THE-ART METHODS ON PASCAL VOC 2007.

Method	mAP	
	ImageNet pre.	MS-COCO pre.
SSGRL [32]	93.4	95.0
ML-GCN [13]	94.0	-
ADDGCN [12]	-	96.0
CCD [8]	-	95.8
ASL [47]	94.6	95.8
Baseline [1]	93.3	95.8
Mixup [3]	93.8	95.6
CutMix [10]	93.8	95.5
ResizeMix [55]	93.3	95.5
RecursiveMix [56]	93.3	95.7
BA [11]	93.9	95.6
Mosaic [50] w/ bbx	93.9	95.4
SpliceMix	94.6	96.1
SpliceMix-CL	94.8	96.3
SpliceMix*	95.3	96.5
SpliceMix-CL*	95.5	96.7

TABLE V
COMPARISONS ON MS-COCO WITH DIFFERENT MISSING RATIO.

Method	40% label left			single label		
	mAP	CF1	OF1	mAP	CF1	OF1
BCE-LS [60]	73.1	44.9	41.2	70.5	40.9	37.3
WAN [60]	72.1	62.8	64.9	70.2	58.0	58.6
Focal [61]	71.7	48.7	44.0	70.2	47.0	41.4
ASL [47]	72.7	67.7	71.7	71.8	44.8	37.9
SPLC [59]	75.7	67.9	73.3	73.2	61.6	67.4
Baseline [1]	74.4	69.3	73.5	72.2	61.7	67.9
Mixup [3]	75.6	70.3	74.4	73.6	66.9	71.6
CutMix [10]	75.5	70.3	74.5	73.6	67.3	71.6
ResizeMix [55]	75.3	69.5	73.8	73.4	67.0	71.7
RecursiveMix [56]	74.9	69.3	73.8	73.0	66.4	70.9
BA [11]	74.6	69.7	73.8	72.5	65.7	70.0
SpliceMix	76.0	71.0	75.1	74.4	67.5	72.0
SpliceMix-CL	76.9	71.4	75.5	74.7	67.5	72.2
SpliceMix*	78.6	73.5	76.5	77.0	67.5	70.8
SpliceMix-CL*	79.6	69.7	73.8	77.6	70.7	74.2

outperform all other methods. In compared Mix-style methods, ResizeMix [55] and RecursiveMix [56] are highly customized for single-label images that show poor performance (also see Table III), that is unsuitable for MLIC. Notably, comparing the mAP of four previous Mix-style methods on MS-COCO pre-training that is inferior to the baseline, our SpliceMix enjoys the gain from MS-COCO hugely.

Mosaic shows comparable performance to other Mix-style methods on Pascal VOC 2007. Many images in this data set only contain one category, which means a few objects are blended in a mixed image. The issue of Mosaic existing on MS-COCO could not be serious on Pascal VOC 2007. The advantage of semantic blending in Mosaic gains the upper hand. However, it still lags far behind SpliceMix due to the absence of label dependency learning.

2) Multi-label Image Classification with Missing Labels

In real scenarios, it is usually difficult to collect all true labels of a multi-label image. An image may be annotated with only partial positive labels, which hinders the performance of MLIC. Multi-label learning with missing/partial-annotated labels has been a hot spot in recent researches [58], [59]. In a popular setting [58], partial positive and negative labels are accurately known and for unknown labels, they can be either positive or negative. In this article, we follow the previous work [59] and further relax such setting, where no accurate negative labels are required. In other words, only limited positive labels are known in our missing label setting which is more flexible in practice.

We conduct experiments on MS-COCO with missing label setting. Following the setting of [59], we choose two settings of missing ratio for experiments, that are 40% remaining positive labels and single positive label per images. Keeping the same architecture to previous work [59], we adopt ResNet-50 with global max pooling as our baseline. Table V lists the comparison results, where methods in the first block is originally reported in [59], and we implement the remaining methods for comparisons. It is easy to discover that our methods consistently outperform all other methods. Compared to the baseline, the proposed SpliceMix shows more robust results as the decrease of left label ratio, which indicates good capacities of our SpliceMix to handle the problem of missing labels. In addition, the ImageNet pre-training via SpliceMix helps our two methods boost a lot, which exhibits its advantage in model pre-training.

D. Improvement to State-of-the-Art Methods

The proposed SpliceMix is a simple and effective augmentation strategy, which is orthogonal to existing MLIC methods and various backbone models. We evaluate our SpliceMix on the performance improvement to four state-of-the-art MLIC methods and four advanced backbones. The result is listed in Table VI. In compared methods, Visual Transformer (ViT) [63] is fine-tuned from SWAG [65] weight and Swin Transformer (short for Swin) [64] is pre-trained by ImageNet-22k [66]. All other methods use the ImageNet-1k pre-training.

It can be seen from Table VI, SpliceMix consistently improves all methods in primary metrics that are mAP, CF1 and OF1. For state-of-the-art MLIC methods, SpliceMix boosts them at least 0.5% and at most 1.2% mAP. For advanced backbones including a lightweight network (ShuffleNetV2 [62]), a classical CNN (ResNet-101 [1]) and two Transformer-based models (ViT-B and Swin-B), SpliceMix boosts them at least 0.9% and at most 1.4% mAP.

Multi-label image data sets usually have the issue of positive-negative imbalance [47]. A model could predict positive classes with low confidences, leading to high precision and low recall. Our SpliceMix constructs the label of a mixed image by a union of regular labels, which improves the ratio of positive-negative classes in some degree. Comparing the CP and CR, or OP and OR of four backbones in Table VI, we can discover that CP and OP have larger values than CR and OR due to too low confidences of positive classes. In our SpliceMix, such circumstance is mitigated. SpliceMix helps these backbones increase the CR and OR with an acceptable

TABLE VI
THE PERFORMANCE IMPROVEMENT TO STATE-OF-THE-ART MLIC METHODS USING SPLICEMIX. ↑ DENOTES THE PERFORMANCE GAIN AND ↓ DENOTES THE PERFORMANCE DROP.

Method	mAP	All						Top 3					
		CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
MLGCN [13]	83.0	85.1	72.0	78.0	85.8	75.4	80.3	89.2	64.1	74.6	90.5	66.5	76.7
+SpliceMix	84.2 (1.2↑)	87.1 (2.0↑)	72.3 (0.3↑)	79.1 (1.1↑)	87.8 (2.0↑)	75.3 (0.1↓)	81.1 (0.8↑)	90.4 (1.2↑)	64.8 (0.7↑)	75.5 (0.9↑)	91.7 (1.2↑)	66.6 (0.1↑)	77.1 (0.4↑)
SSGRL [32]	83.8	89.9	68.5	76.8	91.3	70.8	79.7	91.9	62.5	72.7	93.8	64.1	76.2
+SpliceMix	84.7 (0.9↑)	87.3 (2.6↓)	72.7 (4.2↑)	79.3 (2.5↑)	87.9 (3.4↓)	75.8 (5.0↑)	81.4 (1.7↑)	90.5 (1.4↓)	64.6 (2.1↑)	75.4 (2.7↑)	91.9 (1.9↓)	66.8 (2.7↑)	77.4 (1.2↑)
ADDGCN (576) [12]	85.2	84.7	75.9	80.1	84.9	79.4	82.0	88.8	66.2	75.8	90.3	68.5	77.9
+SpliceMix	85.8 (0.6↑)	87.6 (2.9↑)	74.2 (1.7↓)	80.3 (0.2↑)	88.6 (3.7↑)	77.1 (2.3↓)	82.5 (0.5↑)	91.0 (2.2↑)	65.5 (0.7↓)	76.2 (0.4↑)	92.5 (2.2↑)	67.5 (1.0↓)	78.1 (0.2↑)
TDRG [33]	84.6	86.0	73.1	79.0	86.6	76.4	81.2	89.9	64.4	75.0	91.2	67.0	77.2
+SpliceMix	85.1 (0.5↑)	86.7 (0.7↑)	74.0 (0.9↑)	79.8 (0.8↑)	87.4 (0.8↑)	77.0 (0.6↑)	81.9 (0.7↑)	90.3 (0.4↑)	65.4 (1.0↑)	75.8 (0.8↑)	91.7 (0.5↑)	67.5 (0.5↑)	77.7 (0.5↑)
ShuffleNetV2 [62]	69.7	80.2	51.3	62.5	85.5	58.9	69.7	82.8	47.0	60.0	88.6	54.3	67.3
+SpliceMix	70.7 (1.0↑)	77.9 (2.3↓)	56.2 (4.9↑)	65.5 (3.0↑)	83.6 (1.9↓)	62.6 (3.7↑)	71.6 (1.9↑)	81.1 (1.7↓)	50.7 (3.7↑)	62.4 (2.4↑)	88.1 (0.5↓)	56.6 (2.3↑)	68.9 (1.6↑)
ResNet-101[a] [1]	80.5	83.4	68.0	74.9	86.3	72.5	78.9	87.0	60.5	71.4	90.6	64.3	75.2
+SpliceMix	81.9 (1.4↑)	79.2 (4.2↓)	74.1 (6.1↑)	76.6 (1.7↑)	83.2 (3.1↓)	77.1 (4.6↑)	80.0 (1.1↑)	84.5 (2.5↓)	65.1 (4.6↑)	73.5 (2.1↑)	89.2 (1.4↓)	66.8 (2.5↑)	76.4 (1.2↑)
ViT-B (384) [63]	86.6	87.5	74.9	80.7	88.7	77.6	82.8	90.9	66.6	76.9	92.7	68.4	78.7
+SpliceMix	87.5 (0.9↑)	86.6 (0.9↓)	78.0 (3.1↑)	82.1 (1.4↑)	87.7 (1.0↓)	80.2 (2.6↑)	83.8 (1.0↑)	91.4 (0.5↑)	68.4 (1.8↑)	78.3 (1.4↑)	92.4 (0.3↓)	69.5 (1.1↑)	79.4 (0.7↑)
Swin-B (224) [64]	84.9	86.4	73.7	79.6	87.5	76.3	81.5	89.9	65.8	76.0	91.6	67.7	77.8
+SpliceMix	85.8 (0.9↑)	85.4 (1.0↓)	76.2 (2.5↑)	80.5 (0.9↑)	86.6 (0.9↓)	78.6 (2.3↑)	82.4 (0.9↑)	90.2 (0.3↑)	67.5 (1.7↑)	77.2 (1.2↑)	91.4 (0.2↓)	68.9 (1.2↑)	78.6 (0.8↑)

[a]The vanilla ResNet is utilized here. In our main experiments, we replace the last global average pooling of ResNet-101 with global max pooling as our base model for fast convergence and good performance [13]. From this table, we can see our SpliceMix consistently improve ResNet no matter what the global pooling layer is.

(a) SpliceMix

(b) SpliceMix-CL

Fig. 5. AP improvement of our methods boosting the baseline. For clear presentation, the top-20 smallest and largest categories are selected from left to right.

drop of CP and OP. As a result, SpliceMix improves the CF1 and OF1 significantly.

E. Small Object Recognition

A multi-label image usually contains multiple objects whose scales are fickle. Small objects could be ignored due to low resolution [33] or global pooling operation [34], resulting in poor recognition performance. In SpliceMix, we deal with small object recognition problem via exploiting information learned from images with different scales in the same batch efficiently. Besides, we also introduce a consistency learning-based design to bridge the gap of objects between regular images and mixed images to further improve SpliceMix.

We conduct experiments on MS-COCO to evaluate our methods in boosting small object recognition performance. In fact, different from the object detection task, it is impossible to recognize all objects in a multi-label image due to the lack of bounding boxes, although small objects always impair the performance of MLIC. We calculate the average object size of each categories by the bounding box offered in the MS-COCO data set. And then, we sort all categories from small to large according to the average object size. The AP improvement of our methods compared to the baseline (ResNet-101) is illustrated in Fig. 5. The proposed SpliceMix and SpliceMix-CL boost the small object recognition performance remarkably. The mean AP improvement of the top-20 smallest and largest categories in SpliceMix is 1.8% and 0.9%, respectively. SpliceMix boosts the AP of small objects more than twice that of large objects. Comparing Figs. 5 (a) and (b), SpliceMix-CL further improves SpliceMix in almost all selected categories,

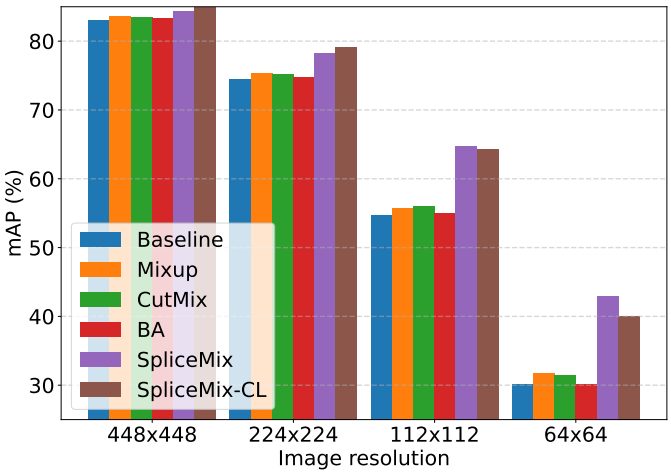


Fig. 6. Comparisons on various image resolutions in the inference phase. All methods are trained with 448×448 image resolution and inferred with the resolutions of 448×448, 224×224, 112×112, and 64×64, respectively.

which indicates the effectiveness of our consistency learning-based design.

Note that the improvement of small object recognition is not only due to cross-scale training in our SpliceMix, but also owing to semantic blending learning. Small objects tend to have poor correlation to other objects because they could be neglected easily. Our SpliceMix enhances the label dependency and reduces the co-occurred bias that can contribute to identify small objects both in their usual and unusual scenes.

F. Inference with Low Resolution Images

In practical applications, the quality of query images may be not guaranteed. A model could be trained with high resolution images, while inferring images under a low resolution circumstance. The model performance will be hindered severely, due to the large resolution discrepancy between training and inferring [67]. Our SpliceMix trains regular images and their downsampled versions simultaneously, which is beneficial to mitigate the above issue.

Fig. 6 compares the inference performance with various image resolutions on MS-COCO. We can see that low resolutions degrade the performance of all methods. The mAP of four compared methods drops rapidly as the inferring resolution declines. However, our SpliceMix and SpliceMix-CL present good robustness to low resolution images. For instance, the mAP of SpliceMix surpasses compared methods over at least 10% in the ultra-low 64×64 resolution. Besides, we may discover that SpliceMix-CL outperforms SpliceMix in the resolutions of 448×448 and 224×224, while losing its superiority in 112×112 and 64×64. In the default setting of SpliceMix (see Sec. IV-A), at least one of the height and width of a downsampled image is 224. The consistency learning between regular images and their downsampled images may focus on bridging the gap between 448×448 and 224×224, resulting in sub-optimal performance to lower resolutions.

G. Performance Analyses

1) Label Dependency and Co-occurred Bias

As stated in Sec. III-A, a baseline can learn implicit label dependency. The labels generated by Mix-style methods will contain inaccurate label correlations inevitably that could hinder the process of learning label dependency. Nevertheless, it is an either-or situation that is to enhance the label dependency or reduce the co-occurred bias [8], [17], [36]. On one hand, capturing label dependency is beneficial for a model to recognize co-occurred categories in their usual context, but on the other, it may hamper the model generalizability of recognizing categories occurring in their unusual context. It is challenging to seek a trade-off between the two. In view of this, the proposed SpliceMix gives a practical solution.

SpliceMix learns label dependency from the regular batch and reduces co-occurred bias via semantic blending learning in the mixed set. Compared to existing Mix-style methods, less noise of label correlations are introduced. Table VII gives the ratio of noisy label correlations when Mixup [3], CutMix [10], and SpliceMix are adopted on VOC 2007 and MS-COCO, respectively. The ratio is calculated by

$$r = \frac{1}{|\mathbb{B}| \times |\mathcal{B}|} \sum_{\mathcal{B} \in \mathbb{B}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{B}} \mathbb{I}(\mathbf{y}, \mathrm{A}) \ , \tag{8}$$

where $\mathbb{B}$ denotes all mini-batches in a training epoch, $\mathcal{B}$ denotes a mini-batch, $(\mathbf{x}, \mathbf{y})$ is a sample-label pair, and $\mathbb{I}(\mathbf{y}, \mathrm{A})$ counts the inaccurate label correlations in each sample. Here A is a matrix with the values in $\{0, 1\}$, indicating the co-occurrence of two categories in the training set. If classes i and j co-occur at least once, $\mathrm{A}_{i,j}$ is 1, vice versa. Further, we consider two ways of $\mathbb{I}(\mathbf{y}, \mathrm{A})$ that are in-sample ($\mathbb{I}_{in}$) and out-sample ($\mathbb{I}_{out}$), corresponding to r_{in} and r_{out} in Table VII. For $\mathbb{I}_{in}$, second-order correlations are calculated in positive labels $\mathbf{y}_P$ of each sample. For $\mathbb{I}_{out}$, 1 will be returned if existing two categories which are never co-occurred. Hence, they are formulated by

$$\mathbb{I}_{in}(\mathbf{y}, \mathrm{A}) = \frac{1}{|\mathbf{y}_P|^2} \sum_{i \in \mathbf{y}_P} \sum_{j \in \mathbf{y}_P} (1 - \mathrm{A}_{i,j}) \ , \tag{9}$$

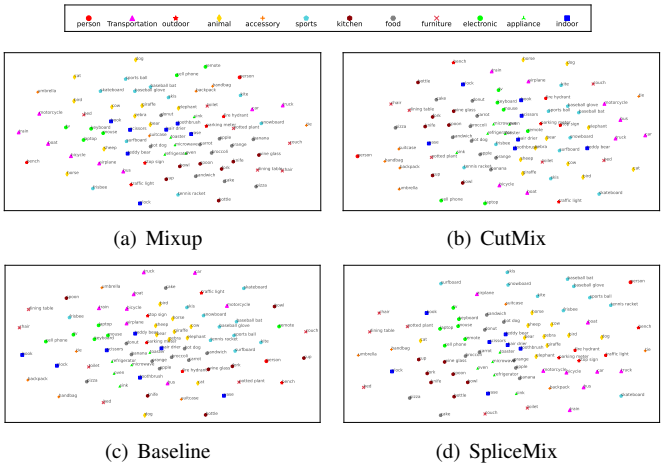
$$\mathbb{I}_{out}(\mathbf{y}, \mathrm{A}) = \begin{cases} 1, & \exists \mathrm{A}_{i,j} = 0, \ i, j \in \mathbf{y}_P \\ 0, & \text{otherwise} \end{cases} \ , \tag{10}$$

where $\mathbf{y}_P$ denotes the index set of positive labels in a sample. From Table VII, we can see that our SpliceMix has a fairly low ratio of noisy label correlations in the metric of r_{in}. Actually, since the label dependency is learned by updating gradient of classifier weights in the class-wise way, r_{in} is a more important evaluation metric. A slight noise of label correlations is beneficial for learning ameliorated label dependency, which we will present next.

To evaluate the ability of learning label dependency using different methods, we also apply T-SNE visualization to the classifier weight pre-trained on MS-COCO [51] in Fig. 7. As we can see, classifiers of relevant classes learned by Mixup and CutMix are both dispersive. A high ratio of noisy label correlations in the two methods results in weaker label dependency learning than the baseline. In contrast, SpliceMix can

TABLE VII
THE RATIO (IN %) OF NOISY LABEL CORRELATIONS USING MIXUP, CUTMIX, AND SPLICEMIX.

Method	VOC 2007		MS-COCO	
	r_{in}	r_{out}	r_{in}	r_{out}
Mixup [3]	12.8	31.0	2.5	17.8
CutMix [10]	12.2	30.7	2.5	17.9
SpliceMix	2.1	15.6	0.3	9.1



(a) Mixup (b) CutMix

(c) Baseline (d) SpliceMix

Fig. 7. T-SNE Visualization on the classifier weight of Mixup, CutMix, Baseline, and SpliceMix. Categories in the same shape and color belong to the same super set offered by MS-COCO.

TABLE VIII
PERFORMANCE COMPARISON ON EXCLUSIVE DISTRIBUTION AND CO-OCCURRED DISTRIBUTION. THE MAP METRIC IS REPORTED.

Method	Exclusive	Co-occur	Exclusive+Co-occur
Baseline [1]	21.6	65.5	87.1
SpliceMix	23.7	66.4	90.1

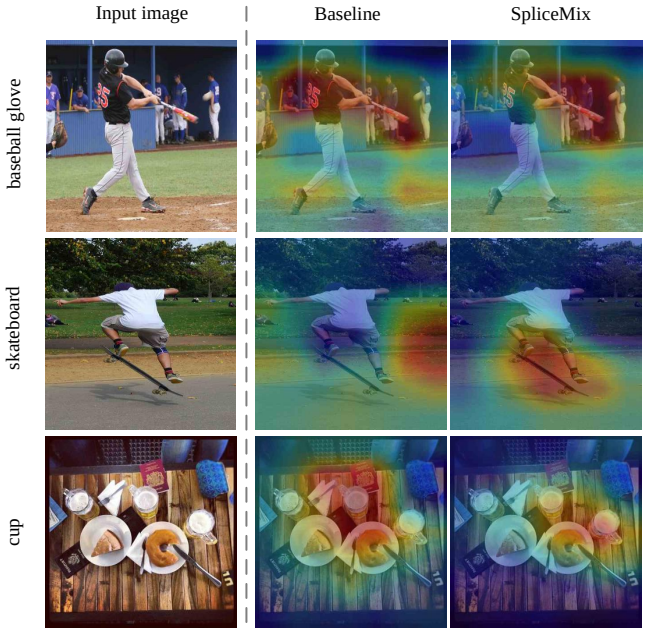


Fig. 8. CAM visualization of highly biased categories. The biased category pairs (baseball glove, person), (skateboard, person) and (cup, dining table) are in the 20 most biased category pairs.

capture better label correlations. For example, the categories "cup" and "bowl" lose their co-occurrence relationships in the baseline. However, in our SpliceMix, the two categories are near their correlated categories, which accurately reflects category relationships in their super set "kitchen". Moreover, SpliceMix also builds improved label dependencies for other super sets, such as "electronic", "animal" and "outdoor".

For studying co-occurred bias, we follow the previous work [17] and conduct experiments on COCO-Stuff [68] that is an augmented version of MS-COCO 2017 with pixel-wise annotations for 91 stuff classes. We run our SpliceMix based on a re-implementation of [17]. Before evaluation, the 20 most biased category pairs are determined. For each biased category pair (b, c), the test set can be divided into three sets: co-occurred images that contain both b and c, exclusive images that contain b but not c, and other images that do not contain b. Then, two test distributions can be constructed: 1) the "exclusive" distribution containing exclusive and other images and 2) the "co-occur" distribution containing co-occurred and other images. More details can be found in [17], [69]. The classification result is reported in Table VIII.

As we can see, SpliceMix improves the baseline (ResNet-50) by 2.1% mAP in exclusive distribution, which means our SpliceMix has better performance of recognizing categories when these categories occur in their unusual context. The co-occurred bias is mitigated by SpliceMix. In addition, for co-occurred categories, SpliceMix also outperforms the baseline by 0.9% mAP, which indicates ameliorated label dependency is learned in SpliceMix. A good trade-off between learning label dependency and reducing co-occurred bias can be sought. Furthermore, we compare the class activation map (CAM) [70] of the baseline and SpliceMix to present the issue existing in highly biased category pairs. As shown in Fig. 8, for the categories "baseball glove" and "cup", the baseline locates more areas on the objective co-occurred categories, *i.e.*, "person" and "dining table", respectively. Only a little information of the query category can be learned which may be less discriminative for recognizing this category in its rare context. Oppositely, our SpliceMix locates "baseball glove" and "cup" intensively, which means that less co-occurred bias is induced. Let us see the second row of Fig. 8, the baseline fails to discover the object, which could be because the baseline can not identify the easy category "person", resulting in that its correlated object (*i.e.*, "skateboard") is ignored. In our SpliceMix, such strong co-occurred relationship is not required. Hence, SpliceMix can recognize and locate "skateboard" effectively.

2) Ablation Studies

We study the effectiveness of the proposed methods from four aspects, *i.e.*, cross-scale learning, semantic blending learning, batch splice and consistency learning, where the batch splice denotes the splice of the regular batch and mixed set (see Fig. 2). Considering that the batch splice and consistency learning is not orthogonal to cross-scale learning and semantic

TABLE IX
ABLATION STUDY FOR DIFFERENT COMPONENTS OF SPLICEMIX AND SPLICEMIX-CL. THE FIRST ROW IS THE RESULTS OF THE BASELINE AND THE LAST TWO ROWS ARE THE RESULTS OF SPLICEMIX AND SPLICEMIX-CL, RESPECTIVELY.

f_{ds}	f_{mg}	$\cup$	L_{cl}	mAP	CF1	OF1
				83.0	77.9	80.3
✓				83.2	77.3	80.0
✓		✓		83.5	78.4	80.6
✓		✓	✓	84.1	78.5	80.7
✓	✓			82.9	77.8	80.2
✓	✓	✓		84.2	78.7	80.9
✓	✓	✓	✓	84.7	79.7	81.7

blending learning, we will discuss them among the latter. The experiments are conducted on MS-COCO with a 2×2 grid strategy of SpliceMix and the results are listed in Table IX where the denotations f_{ds} and f_{mg} account for cross-scale learning and semantic blending learning, respectively (see Eq. (2)). $\cup$ denotes batch splice and L_{cl} denotes consistency learning (see Eq. (6))..

Cross-scale learning: For removing the component of semantic blending, we adopt the setting of grids with dropout, *i.e.*, $2 \times 2 - 3$. Then, the mixed image is degraded to a downsampled image merely, which is corresponding to $(f_{ds} + \cup)$ in Table IX. The consistency learning can be built from regular images and their downsampled images, corresponding to $(f_{ds} + \cup + L_{cl})$. To reach only f_{ds}, we downsample regular images to half of their height and width, and then train downsampled images after training regular batch. From Table IX, we can see that the batch splice improves the performance of cross-scale learning, which indicates the cross-scale information from the same batch is better than that from different batch. Further, with the consistency learning, the performance of cross-scale learning is boosted a lot. In our SpliceMix-CL, the knowledge of fine regular images is leveraged to guide the model to learn better representation of coarse mixed images. In this part of only cross-scale learning, the gain from consistency learning is obvious, and it will be more with the joint of semantic blending learning. Besides, owing to cross-scale learning, SpliceMix has an advantage of recognizing small objects, even when the query image is low-resolution (see Sec. IV-F).

Semantic blending learning: Semantic blending in our SpliceMix is implemented by making a grid of several downsampled regular images. The object information is lossless except the resolution, which better contributes to producing an unusual scene for present objects than current Mix-style methods [3], [10]. Evaluating the semantic blending learning separately is spiny. Therefore, we evaluate its performance by incremental comparisons. Comparing $(f_{ds}+\cup)$ to $(f_{ds}+f_{mg}+\cup)$ in Table IX, we can find that with semantic blending, the mAP is improved by 0.7% that suggests the effectiveness of semantic blending learning to mitigate the co-occurred bias. We may also notice that $(f_{ds}+f_{mg})$ without batch splice harms the performance a little compared to the baseline, from which we can realize the importance of batch splice.

TABLE X
COMPARISONS OF TRAINING LOSS VALUES IN REGULAR BATCH AND MIXED SET.

Sample Set	Baseline [1]	SpliceMix	SpliceMix-CL
Regular Batch	0.0345	0.0330	0.0332
Mixed Set	0.2390	0.1052	0.1016

3) Analysis of SpliceMix-CL

Due to the implicit label dependency learned by a baseline, the prediction of a category is related to its co-occurred categories. When a category occurs solely or in an unusual scene, the prediction confidence could be low. A direct consequence is low recall and high precision in the inference phase, which can be observed in Tables III and VI. In SpliceMixed images, a category is put into a new scene, reducing the contextual bias from learning label dependency. Higher prediction confidence can be achieved using our SpliceMix. However, higher confidence could result in more false positive predictions. The SpliceMix-CL extends SpliceMix via building the consistency learning between the regular images and their downsampled counterparts. The model is encouraged to learn fine knowledge of mixed images from regular images. The prediction of mixed images will be close to their regular versions, mitigating the issue of overconfident prediction in SpliceMix. In addition, the cross-scale training is enhanced. As illustrated in Fig. 5, compared to the baseline, SpliceMix-CL improves the mean of AP in small categories by 2.3% that is 0.5% higher than SpliceMix.

Table X compares the training loss values in the regular batch and mixed set using the baseline, SpliceMix and SpliceMix-CL. Owing to consistency learning, SpliceMix-CL achieves lower training loss in the mixed set than SpliceMix, although the latter has reduced the loss value significantly compared to the baseline. For clarifying the role of SpliceMix-CL, we illustrate some examples in Fig. 9 where false positive categories are predicted in SpliceMix. We can see that SpliceMix tends to make overconfident predictions even though some categories are not present in the image. SpliceMix-CL mitigates such a situation with the aid of transferring fine knowledge from regular images to mixed images.

4) Ways of Batch Splice

The batch splice in our SpliceMix is a combination of the mixed set and regular batch, where the final SpliceMixed batch size is increased a little due to the introduction of the mixed set. Another way to batch splice is combining the mixed set and regular images that do not attend to mixing. In other words, there is no repeated image in such final batch whose size is less than the regular batch. We conduct experiments on MS-COCO to compare the performance of the batch splice used in our SpliceMix and the batch splice without repeated images. In our SpliceMix, the regular batch size is 32 and the cardinality of the mixed set is 8. To reach the same final batch size to SpliceMix, for the compared method, we choose the number of regular images and mixed images from the pairs of $\{(32, 8), (20, 20), (0, 40)\}$, where the pair of $(0, 40)$ only

Fig. 9. Illustration of predictions using SpliceMix and SpliceMix-CL. The false positive predictions are highlighted in teal.

TABLE XI
COMPARISONS OF BATCH SPLICE USED IN SPLICEMIX (OURS) AND WITHOUT REPEATED IMAGES. $(\cdot,\cdot)$ DENOTES THE NUMBER OF REGULAR IMAGES (THE FORMER) AND MIXED IMAGES (THE LATTER), RESPECTIVELY, SAME FOR TABLE XII.

Method	Ours	$(32, 8)$	$(20, 20)$	$(0, 40)$
mAP	84.2	83.7	82.6	81.0

uses the mixed images for training. A 2×2 grid strategy is adopted for all methods.

Table XI reports the results of the two ways of batch splice. When only mixed images are used (*i.e.*, $(0, 40)$ in Table XI) for training, the model performance degrades a lot. In the compared method, the mAP increases gradually with the decreased number of mixed images, which indicates the splice with regular images contributes to boosting performance. However, the performance boost of the compared methods is limited. With the same number of regular images and mixed images to our SpliceMix, the compared method achieves 83.7% mAP that is less than SpliceMix. As shown in Table XI, the batch splice used in our SpliceMix is superior to the compared method.

5) Batch Splice for Previous Mix-style Methods

As we claimed in this article, our SpliceMix is designed for MLIC according to their characters. The mixed images in SpliceMix preserve unbroken objects and blend the image semantics for alleviating co-occurred bias, which is more suitable for MLIC than previous Mix-style methods. To demonstrate this, we compare our SpliceMix with the splice of regular batch and mixed images from Mixup [3], CutMix [10], and Mosaic [50], respectively. In this experiment, the regular batch size is 32 for MS-COCO [51]. We consider two settings for compared methods, which are the cardinality of the mixed set same to regular batch size and same to that used in SpliceMix. For the former setting, we set the regular batch size to 20 and the final batch size for training will be 40 to reach the same final batch size to other methods. In our SpliceMix, the cardinality of the mixed set is 8.

The results of splice with Mixup, CutMix, and Mosaic are listed in Table XII. As we can see, mixed images generated by Mixup cannot work well with the joint of regular images. Mixup linearly combines two images, which may be too complex for MLIC. The large difference between regular images and mixed images from Mixup leads to poor learning ability in a spliced batch. On the other hand, the splice with CutMix achieves similar mAP to vanilla CutMix, no matter how many mixed images are used. Mosaic enjoys the benefit of the proposed splice strategy and obtains a high mAP. Both SpliceMix and Mosaic mix images without the deficiency of objects that are compatible with the proposed splice strategy. Superior to Mosaic, SpliceMix introduces cross-scale training in the same batch for small object recognition. Compared to the splice with previous Mix-style methods, our SpliceMix is an ideal way for batch splice, since mixed images from SpliceMix are suitable for MLIC training and perform well with the joint of regular images.

TABLE XII
COMPARISONS OF BATCH SPLICE WITH DIFFERENT MIX-STYLE METHODS.

Method	mAP	
	$(32, 8)$	$(20, 20)$
Mixup [3]	83.0	82.2
CutMix [10]	83.6	83.6
Mosaic [50]	84.0	83.3
SpliceMix	84.4	-

6) Performance on Various Grids

Here, we discuss the influence on performance with different grid settings and number of mixed images. The results on Pascal VOC 2007 trained with the regular batch size of 16 are illustrated in Fig. 10. To obtain steady results, we run experiments three times and calculate the mean and standard deviation for each grid setting.

Taking a look at the overview of Fig. 10, we can find that the great majority of grids outperform the baseline. Our SpliceMix is insensitivity to its parameters and only needs a few mixed samples to achieve remarkable performance. Secondly, we turn slights to the grids without dropout. As the number of mixed images increases, mAP of most grids descends. Too many mixed images, especially with large grids (*i.e.*, 2×3 and 3×3), harm the learning ability of the model, due to the large scale discrepancy [67] between regular images and their downsampled versions in mixed images, and the over-complicated semantic. For the latter, as stated in Sec. III-B, we introduce dropout to reduce the semantic complexity. Comparing the pair of each grid setting without and with dropout in Fig. 10, we can see that dropout consistently improves its plain setting, except the grid of 1×2. Actually, the grid of $1 \times 2 - 1$ is degraded to involve only cross-scale learning. This indicates that semantic blending plays a key role in our SpliceMix. Finally, we can summarize a referenced cardinality $|\mathcal{B}'|_{ref}$ for the mixed set, that is $|\mathcal{B}'|_{ref} = \lfloor |\mathcal{B}|/(r \times c - d) \rfloor$, where $|\mathcal{B}|$ denotes the regular batch size and r, c, d denote the rows, columns, the number of dropped sub-images in a grid setting of $r \times c - d$, respectively.

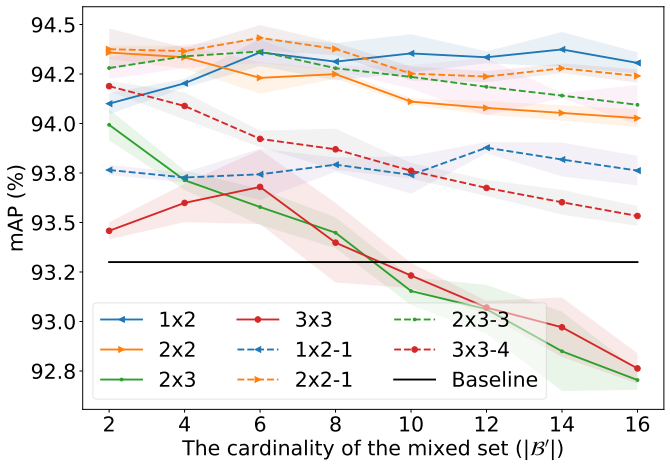


Fig. 10. Comparisons of various grid settings on mAP metric. The mean in a solid or dash line and standard deviation in shaded region are given. Each solid-dash pair in the same color is a grid setting without and with dropout.

V. Conclusion

In this article, we introduce a simple but effective augmentation strategy, namely SpliceMix, for multi-label image classification. SpliceMix augments the sample space and batch scale simultaneously. In augmented sample space, mixed images with semantic blending contribute to alleviating co-occurred bias. In augmented batch, the splice of regular images and mixed images enables cross-scale training and consistency learning. Hence, we also offer a non-parametric, consistency learning-based extension (SpliceMix-CL) to present the potential of extending our SpliceMix for further boosting MLIC. Extensive experiments on various tasks demonstrate the effectiveness of the proposed methods.

VI. Acknowledgement

This work was supported in part by the Major Science and Technology Innovation 2030 "New Generation Artificial Intelligence" key project (No. 2021ZD0111700), NSF of China (No. 62336003, No. 12371510, No. 62172354, No. 62076005), NSF of Jiangsu Province (No. BZ2021013), NSF for Distinguished Young Scholar of Jiangsu Province (No. BK20220080), Dreams Foundation of Jianghuai Advance Technology Center (No. 2023-ZM01Z015), Yunling Scholar Talent Program of Yunnan Province (No. K264202230207), and Yunnan Dengcheng Expert Workstation (No. 202305AF150202).

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778. 1, 3, 4, 5, 6, 7, 8, 9, 11, 12

[2] J. Mellor, J. Turner, A. Storkey, and E. J. Crowley, "Neural architecture search without training," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2021, pp. 7588–7598. 1

[3] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Representations*, 2018. 1, 2, 3, 6, 7, 8, 10, 11, 12, 13

[4] B. Li, F. Zhang, L. Wang, Y. Wang, T. Liu, Z. Lin, W. An, and Y. Guo, "Ddaug: Differentiable data augmentation for weakly supervised semantic segmentation," *IEEE Trans. Multimedia*, pp. 1–12, 2023. 1

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16 000–16 009. 1

[6] X. Yang, F. Liu, and G. Lin, "Effective end-to-end vision language pretraining with semantic visual loss," *IEEE Trans. Multimedia*, no. 99, pp. 1–10, 2023. 1

[7] K. Zhu and J. Wu, "Residual attention: A simple but effective method for multi-label recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 184–193. 1, 2, 7

[8] R. Liu, H. Liu, G. Li, H. Hou, T. Yu, and T. Yang, "Contextual debiasing for visual recognition with causal mechanisms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12 755–12 765. 1, 2, 7, 8, 10

[9] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, and S. Wen, "Cross-modality attention with semantic graph embedding for multi-label classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 07, 2020, pp. 12 709–12 716. 1

[10] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6023–6032. 1, 2, 3, 6, 7, 8, 10, 11, 12, 13

[11] E. Hoffer, T. Ben-Nun, I. Hubara, N. Giladi, T. Hoefler, and D. Soudry, "Augment your batch: Improving generalization through instance repetition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8129–8138. 1, 2, 3, 7, 8

[12] J. Ye, J. He, X. Peng, W. Wu, and Y. Qiao, "Attention-driven dynamic graph convolutional network for multi-label image recognition," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 649–665. 1, 2, 5, 6, 8, 9

[13] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5177–5186. 1, 2, 6, 8, 9

[14] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General multi-label image classification with transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16 478–16 488. 1, 2

[15] X. Deng, S. Feng, G. Lyu, T. Wang, and C. Lang, "Beyond word embeddings: Heterogeneous prior knowledge driven multi-label image classification," *IEEE Trans. Multimedia*, 2022. 1, 2

[16] W. Zhou, W. Jiang, D. Chen, H. Hu, and T. Su, "Mining semantic information with dual relation graph network for multi-label image classification," *IEEE Trans. Multimedia*, 2023. 1, 2

[17] K. K. Singh, D. Mahajan, K. Grauman, Y. J. Lee, M. Feiszli, and D. Ghadiyaram, "Don't judge an object by its context: learning to overcome contextual bias," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 070–11 078. 1, 10, 11

[18] J.-H. Kim, W. Choo, and H. O. Song, "Puzzle mix: Exploiting saliency and local statistics for optimal mixup," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2020, pp. 5275–5285. 1, 2, 3

[19] J. Liu, B. Liu, H. Zhou, H. Li, and Y. Liu, "Tokenmix: Rethinking image mixing for data augmentation in vision transformers," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 455–471. 1, 2, 3

[20] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson, "Classmix: Segmentation-based data augmentation for semi-supervised learning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1369–1378. 1, 3

[21] Y. Su, R. Sun, G. Lin, and Q. Wu, "Context decoupling augmentation for weakly supervised semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 7004–7014. 2

[22] A. S. Uddin, M. S. Monira, W. Shin, T. Chung, and S.-H. Bae, "Saliencymix: A saliency guided data augmentation strategy for better regularization," in *Proc. Int. Conf. Learn. Representations*, 2021. 2, 3

[23] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, "Learning spatial regularization with image-level supervisions for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5513–5522. 2

[24] T. Chen, Z. Wang, G. Li, and L. Lin, "Recurrent attentional reinforcement learning for multi-label image recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018. 2

[25] B.-B. Gao and H.-Y. Zhou, "Learning to discover multi-class attentional regions for multi-label image recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 5920–5932, 2021. 2

[26] J. Zhan, J. Liu, W. Tang, G. Jiang, X. Wang, B.-B. Gao, T. Zhang, W. Wu, W. Zhang, C. Wang *et al.*, "Global meets local: Effective multi-label image classification via category-aware weak supervision," in *Proc. ACM Int. Conf. Multimedia*, 2022, pp. 6318–6326. 2

[27] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014. 2

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016. 2

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017. 2

[30] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2285–2294. 2

[31] S. Narayan, A. Gupta, S. Khan, F. S. Khan, L. Shao, and M. Shah, "Discriminative region-based multi-label zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 8731–8740. 2

[32] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning semantic-specific graph representation for multi-label image recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 522–531. 2, 5, 8, 9

[33] J. Zhao, K. Yan, Y. Zhao, X. Guo, F. Huang, and J. Li, "Transformer-based dual relation graph for multi-label image recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 163–172. 2, 6, 9

[34] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Query2label: A simple transformer way to multi-label classification," *arXiv preprint arXiv:2107.10834*, 2021. 2, 9

[35] X. Zhu, J. Cao, J. Ge, W. Liu, and B. Liu, "Two-stream transformer for multi-label image classification," in *Proc. ACM Int. Conf. Multimedia*, 2022, pp. 3598–3607. 2

[36] R. Liu, J. Huang, G. Li, and T. H. Li, "Causality compensated attention for contextual biased visual recognition," in *Proc. Int. Conf. Learn. Representations*, 2023. 2, 7, 10

[37] J. Yoo, N. Ahn, and K.-A. Sohn, "Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8375–8384. 2, 3

[38] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017. 3

[39] J.-H. Lee, M. Z. Zaheer, M. Astrid, and S.-I. Lee, "Smoothmix: A simple yet effective data augmentation to train robust classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 756–757. 3

[40] D. Walawalkar, Z. Shen, Z. Liu, and M. Savvides, "Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification," *arXiv preprint arXiv:2003.13048*, 2020. 3

[41] S. Fort, A. Brock, R. Pascanu, S. De, and S. L. Smith, "Drawing multiple augmentation samples per image during training efficiently decreases test error," *arXiv preprint arXiv:2105.13343*, 2021. 3

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019. 4, 6

[43] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4320–4328. 5

[44] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 953–11 962. 5

[45] D. Chen, J.-P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen, "Cross-layer distillation with semantic calibration," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 8, 2021, pp. 7028–7036. 5

[46] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789–1819, 2021. 5

[47] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, "Asymmetric loss for multi-label classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 82–91. 5, 8

[48] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015. 5

[49] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 976–11 986. 5

[50] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. 6, 7, 8, 13

[51] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 740–755. 6, 10, 13

[52] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010. 6

[53] F. Zhou, S. Huang, and Y. Xing, "Deep semantic dictionary learning for multi-label image classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 4, 2021, pp. 3572–3580. 7

[54] Z.-M. Chen, Q. Cui, B. Zhao, R. Song, X. Zhang, and O. Yoshie, "Sst: Spatial and semantic transformers for multi-label image recognition," *IEEE Trans. Image Process.*, vol. 31, pp. 2570–2583, 2022. 7

[55] J. Qin, J. Fang, Q. Zhang, W. Liu, X. Wang, and X. Wang, "Resizemix: Mixing data with preserved object information and true labels," *arXiv preprint arXiv:2012.11101*, 2020. 7, 8

[56] L. Yang, X. Li, B. Zhao, R. Song, and J. Yang, "Recursivemix: Mixed learning with history," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 8427–8440. 7, 8

[57] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015. 7

[58] T. Chen, T. Pu, H. Wu, Y. Xie, and L. Lin, "Structured semantic transfer for multi-label recognition with partial labels," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 1, 2022, pp. 339–346. 8

[59] Y. Zhang, Y. Cheng, X. Huang, F. Wen, R. Feng, Y. Li, and Y. Guo, "Simple and robust loss design for multi-label learning with missing labels," *arXiv preprint arXiv:2112.07368*, 2021. 8

[60] E. Cole, O. Mac Aodha, T. Lorieul, P. Perona, D. Morris, and N. Jojic, "Multi-label learning from single positive labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 933–942. 8

[61] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988. 8

[62] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131. 8, 9

[63] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 8, 9

[64] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 10 012–10 022. 8, 9

[65] M. Singh, L. Gustafson, A. Adcock, V. de Freitas Reis, B. Gedik, R. P. Kosaraju, D. Mahajan, R. Girshick, P. Dollár, and L. Van Der Maaten, "Revisiting weakly supervised pre-training of visual perception models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 804–814. 8

[66] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," *arXiv preprint arXiv:2104.10972*, 2021. 8

[67] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, "Fixing the train-test resolution discrepancy," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019. 10, 13

[68] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1209–1218. 11

[69] S. S. Kim, S. Zhang, N. Meister, and O. Russakovsky, "[re] don't judge an object by its context: learning to overcome contextual bias," *arXiv preprint arXiv:2104.13582*, 2021. 11

[70] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929. 11

Lei Wang is currently pursuing the Ph.D. degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include multi-label learning and sparse representation.

Yibing Zhan received the bachelor's and doctor's degrees from the School of Information Science and Technology, University of Science and Technology of China, in 2012 and 2018, respectively. After graduating with a doctor's degree, from 2018 to 2020, he served as an associate researcher with the School of Computer Science, Hangzhou Dianzi University. Now, he works with the JD Explore Academy as an algorithm scientist. He mainly explores scene graph generation, foundation model, and graph neural networks. He has published many scientific papers in top conferences and journals such as NeurIPS, CVPR, ACM MM, ICCV, and IEEE Transactions on Multimedia.

Leilei Ma is currently pursuing the Ph.D. degree in the School of Computer Science and Technology, Anhui University, Hefei, China. His research interests focus on multi-label learning and weakly supervised learning.

Dapeng Tao (Member, IEEE) is currently a Professor with the School of Information Science and Engineering, Yunnan University, Yunnan, China. He has served as a Doctoral Advisor of Computer Science and Technology and a Doctoral Advisor of Control Science and Engineering with the University of the Chinese Academy of Sciences, Beijing, China. He is mainly engaged in research in the field of artificial intelligence. Prof. Tao has served as a Special Reviewer and a Guest Editor for more than ten international academic journals, including IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Multimedia, and IEEE Transactions on Biomedical Engineering.

Liang Ding (Member, IEEE) received Ph.D. from the University of Sydney, Australia. He was a Research Scientist and led the NLP Research Group at JD Explore Academy, JD.com, China, where he led the large-scale language model pretraining project and won the Superior AI Leader (SAIL) Award at the World Artificial Intelligence Conference 2022, and the highest technical award at JD.com - Technology Golden Award. He has authored more than 70 papers in NLP/ML venues, including ACL, EMNLP, ICLR, ICML, AAAI, IJCAI, CVPR, IEEE T-PAMI, IEEE T-KDE, IEEE T-NNLS, IEEE T-ASLP, and IEEE T-MM. He was the Area Chair (or Session Chair) for ACL, EMNLP, NAACL, AAAI, and SDM. He won numerous AI challenges, including SuperGLUE, GLUE, WMT 2022, IWSLT 2021, WMT 2020, and WMT 2019.

Chen Gong (Senior Member, IEEE) received his dual doctoral degree from Shanghai Jiao Tong University (SJTU) and University of Technology Sydney (UTS) in 2016 and 2017, respectively. Currently, he is a full professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests mainly include machine learning, data mining, and learning-based vision problems. He has published more than 130 technical papers at prominent journals and conferences such as JMLR, IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, IEEE T-CYB, IEEE T-CSVT, IEEE T-MM, IEEE T-ITS, ACM T-IST, ICML, NeurIPS, ICLR, CVPR, AAAI, IJCAI, ICDM, etc. He serves as the associate editor for IEEE T-CSVT and NePL, and also the Area Chair or Senior PC member of several top-tier conferences such as AAAI, IJCAI, ICLR, ICDM, ECML, AISTATS, ACM MM, etc. He won the "Excellent Doctorial Dissertation Award" of Chinese Association for Artificial Intelligence, "Young Elite Scientists Sponsorship Program" of China Association for Science and Technology, "Wu Wen-Jun AI Excellent Youth Scholar Award", and the Scientific Fund for Distinguished Young Scholars of Jiangsu Province. He was also selected as the "Global Top Chinese Young Scholars in AI" released by Baidu.