REMiXER: OBJECT-WARE MIXING LAYER FOR VISION TRANSFORMERS AND MIXERS

Hyunwoo Kang*, Sangwoo Mo*, Jinwoo Shin
Korea Advanced Institute of Science and Technology (KAIST)
{hyunwookang,swmo,jinwoos}@kaist.ac.kr

ABSTRACT

Patch-based models, e.g., Vision Transformers (ViTs) and Mixers, have shown impressive results on various visual recognition tasks, exceeding classic convolutional networks. While the initial patch-based models treated all patches equally, recent studies reveal that incorporating inductive biases like spatiality benefits the learned representations. However, most prior works solely focused on the position of patches, overlooking the scene structure of images. This paper aims to further guide the interaction of patches using the object information. Specifically, we propose ReMiXER, which reweights the patch mixing layers based on the patch-wise object labels extracted from pretrained saliency or classification models. We apply ReMiXER on various patch-based models using different patch mixing layers: ViT, MLP-Mixer, and ConvMixer, where our method consistently improves the classification accuracy and background robustness of baseline models.

1 INTRODUCTION

Patch-based models, i.e., models that process an input image as a sequence of visual patches, have arisen as a new paradigm of neural networks for visual data, alternating the prior standard convolutional neural networks (CNNs; LeCun et al. (1998)). Remarkably, patch-based models have achieved state-of-the-art results on various computer vision tasks by favoring the scaling properties (Dosovitskiy et al., 2021; Zhai et al., 2021). They also merit various advantages, including out-of-distribution generalization (Naseer et al., 2021), natural extension to video domains (Bertasius et al., 2021), integration with other domains like language or speech (Radford et al., 2021), and easily combined with state-of-the-art visual self-supervised learning (He et al., 2021).

The core concept of patch-based models is to update patch-wise representations by alternating the computation within patches and among patches, called channel mixing and patch (or token) mixing, respectively. The design of patch mixing layers is widely investigated: the pioneering Vision Transformer (ViT; Dosovitskiy et al. (2021)) and its descendants (Touvron et al., 2021a;b) considered self-attention (Vaswani et al., 2017), while other works considered feed-forward (Tolstikhin et al., 2021), convolution (Trockman & Kolter, 2022), or pooling operation (Yu et al., 2021). However, most patch-based models use self-attention or feed-forward mixing layers, which minimizes the inductive biases from the model by employing all patches equally (Khan et al., 2021).

While this data-centric approach is effective on large-scale scenarios, recent works claim that incorporating inductive biases is still essential for patch-based models, especially when learned from limited data (Steiner et al., 2021). To this end, many approaches incorporated spatial inductive bias for patch-based models following the analogy of convolution: designing a patch mixing layer utilizing the location of patches (d’Ascoli et al., 2021; Dai et al., 2021; Wu et al., 2021a; Trockman & Kolter, 2022) or building an architecture that combines convolutional or pooling layers with patch mixing layers (Liu et al., 2021; Yuan et al., 2021b; Wang et al., 2021; Fan et al., 2021). However, both approaches focus on spatial inductive bias, which overlooks the sample-specific object structures. We provide additional discussion on related work in Appendix A.

*Equal contribution
Contribution. We propose ReMixer, a novel reweighting scheme for patch mixing layers leveraging the object structure of images. We demonstrate that ReMixer improves classification accuracy and background robustness of patch-based models, outperforming the models considering spatiality.

2 ReMixer: Object-Aware Mixing Layer

The main idea of ReMixer is to strengthen the interaction of patches containing similar objects while regularizing the connection of different objects (and background). Intuitively, ReMixer improves the discriminability of objects (i.e., better classification) and reduces the spurious correlations between objects and backgrounds (i.e., robust to background and distribution shifts) by learning disentangled representations of objects. We first introduce a general framework of object-aware mixing layer in Section 2.1, then illustrate the specific instantiations for various architectures in Section 2.2.

2.1 Computing Reweighting Mask for ReMixer

The idea of patch-based models is to reshape a 2D image \( x^0 \in \mathbb{R}^{H \times W \times C} \) into a sequence of flattened 2D patches \( x^l \in \mathbb{R}^{N \times (P^2 \times C)} \), where \((H, W)\) is the resolution of original image, \(C\) is the number of color channels, \((P, P)\) is the resolution of each image patch, and \(N = HW/P^2\) is the number of patches. Patch-based models first convert the 2D patches into patch (or token) features \( x^1 := f_{\text{embed}}(x^0) \in \mathbb{R}^{N \times D} \) with latent dimension \(D\) using an embedding function \( f_{\text{embed}} \), then update the patch features by alternating two operations: (a) patch mixing layers \( f_{\text{mix}} : \mathbb{R}^N \rightarrow \mathbb{R}^N \) which mix the features among patches, and (b) channel mixing layers \( g_{\text{mix}} : \mathbb{R}^D \rightarrow \mathbb{R}^D \) which mix the features among channels, where \( l \) implies the operation of layer \( l \). Formally, the \( l \)-th layer of patch-based model updates an input vector \( x^l \in \mathbb{R}^{N \times D} \) to an output vector \( x^{l+1} \in \mathbb{R}^{N \times D} \) following:

\[
\begin{align*}
    z^{l+1} &= [z^{l+1}_{1, d}] = [f_{\text{mix}}(x^{l}_{1, 1, d}; \ldots; x^{l}_{N, d})] \\
    x^{l+1} &= [x^{l+1}_{n, 1, d}] = [g_{\text{mix}}(z^{l+1}_{n, 1}; \ldots; z^{l+1}_{N, d})]
\end{align*}
\]

where \( z_{n, d} \) and \( x_{n, d} \) denotes \( n \)-th patch, \( d \)-th channel value, and \( z^{l+1}_{n, 1} \in \mathbb{R}^N \) and \( x^{l+1}_{n, 1, d} \in \mathbb{R}^D \) denotes row-wise and column-wise subvector of \( z \) and \( x \), respectively.

We introduce ReMixer, a universal framework for improving patch mixing layers \( f_{\text{mix}} \) by incorporating the object structure of images. To this end, we utilize the patch-wise labels \( y \in \mathbb{R}^{N \times K} \) where \( K \) is the number of object classes. Using them, we compute the reweighting mask \( M^l \in \mathbb{R}^{N \times N} \) which strengthens the interaction of patches of similar objects while regularizing the connection of different objects and backgrounds. Formally, we set the \((i, j)\)-th value of the reweighting mask \( M^l_{ij} \) as a reverse distance between the object labels of two patches \( y_i \) and \( y_j \):

\[
M^l_{ij} := \exp(-\kappa^{l(i)} \cdot d(y_i, y_j)) \in (0, 1]
\]

where \( \kappa^{l(i)} \geq 0 \) is a learnable mask scale (scalar) parameter for each layer and \( d(\cdot, \cdot) : \mathbb{R}^K \times \mathbb{R}^K \rightarrow \mathbb{R} \) is a distance function for object labels. We initialize \( \kappa^{l(i)} \) by zero for training, i.e., consider the full interaction initially then focus on the objects as \( \kappa^{l(i)} \) increases. We observe that the model sets higher \( \kappa^{l(i)} \) for lower layers and lower \( \kappa^{l(i)} \) for higher layers (especially, \( \kappa^{l(i)} = 0 \) for the final layer) after training, i.e., ReMixer automatically attends the intra-object relations first then expand to the inter-object relations, which resembles the local-to-global structure of CNNs (see Table 3).

We aim to calibrate the \( N \times N \) interaction of patch mixing layers using the reweighting mask \( M \). If the patch mixing layer \( f_{\text{mix}} := L_{\text{mix}} \) is linear, one can simply (element-wise) multiply the mask to get the masked linear mixer \( M \odot L_{\text{mix}} \). While ReMixer can be applied to any patch layers in principle, one needs careful design to consider the nonlinear dynamics of each layer. We provide specific implementations of ReMixer on various representative models in the next section.

Obtaining object labels. We extract object labels from saliency (Voynov et al., 2021) or classification (Yun et al., 2021) models trained from other source datasets. See Appendix B for details.

2.2 ReMixer for ViT and Mixers

We briefly review three representative patch mixing layers: self-attention, feed-forward, and convolution, and describe the implementations of ReMixer for each layer.
ReMixer for self-attention. Self-attention (Vaswani et al., 2017) mixing layers update patch features by aggregating values with normalized importances (or attentions):

$$f_{\text{mix}}(x) := \text{Softmax}\left(\frac{QK^T}{\sqrt{D_N}}\right)V$$  \hspace{1cm}\text{(4)}$$

where $Q$, $K$, $V$ are query, key, and value, respectively, which are linear projections of input $x \in \mathbb{R}^{N \times D}$, given by $Q := xW_Q \in \mathbb{R}^{N \times D_K}$, $K := xW_K \in \mathbb{R}^{N \times D_K}$, and $V := xW_V \in \mathbb{R}^{N \times D_V}$. Here, we compute $H$ independent attention heads and aggregate outputs for the final output of size $N \times D$ for $D = H \cdot D_V$. Recall that self-attention is basically a matrix multiplication of attention $A$ and value $V$ matrices, and one can (element-wise) multiply the reweighting mask $M$ to the attention matrix to calibrate interaction. Then, we renormalize the masked attention $M \odot A$ to make the row-wise sum be 1 as the original self-attention. To sum up, ReMixer for self-attention is:

$$f_{\text{remix}}(x) := [\hat{A}_{ij}] \cdot V = \left[\frac{M_{ij} \cdot A_{ij}}{\sum_j M_{ij} \cdot A_{ij}}\right] \cdot V$$  \hspace{1cm}\text{(5)}$$

where $\hat{A}$ is the renormalized masked attention. We finally remark that patch-based models using self-attention often use the additional [CLS] token to aggregate the global feature. Here, we define the mask value between the [CLS] token and every other patch to be one and apply Eq. (5).

ReMixer for feed-forward. Feed-forward (or multi-layer perceptron; MLP) mixing layers update patch features with a channel-wise MLP. Since each channel is computed independently, we only consider a $N \times 1$ vector of a single channel. Then, the mixer layer is:

$$f_{\text{mix}}(x) := W_m \cdot \sigma(\underbrace{W_{m-1} \cdot \sigma(\cdots \sigma(W_1 \cdot x))}_{\text{channel-wise MLP}})$$  \hspace{1cm}\text{(6)}$$

where $W_1, \ldots, W_m$ are weight matrices and $\sigma$ is a nonlinear activation. However, it is nontrivial to apply the reweighting mask $M$ since $f_{\text{mix}}$ is nonlinear. To tackle this issue, we decompose the mixing layer $f_{\text{mix}}$ into a linear approximation $L^x_{\text{mix}} \cdot x \approx f_{\text{mix}}(x)$ for a (possibly data-dependent) matrix $L^x_{\text{mix}} \in \mathbb{R}^{N \times N}$ and a residual term $f_{\text{mix}}(x) - L^x_{\text{mix}} \cdot x$. Here, we only calibrate the linear term but omit the residual term. Then, ReMixer for feed-forward is given by:

$$f_{\text{remix}}(x) := \underbrace{L^x_{\text{mix}} \cdot x}_{\text{masked linear}} + \underbrace{(f_{\text{mix}}(x) - L^x_{\text{mix}} \cdot x)}_{\text{residual}}$$  \hspace{1cm}\text{(7)}$$

where $\odot$ is an element-wise product. While finding a good $L^x_{\text{mix}}$ is nontrivial in general, we found that a simple trick of dropping nonlinear activations gives an efficient yet effective solution:

$$L^x_{\text{mix}} := W_m W_{m-1} \cdots W_1 \in \mathbb{R}^{N \times N}.$$  \hspace{1cm}\text{(8)}$$

We observe that this (somewhat crude) approximation performs well in practice. We also tried some data-dependent variants but did not see gain despite their computational burdens.

ReMixer for convolution. Convolutional mixing layers update patch features with a channel-wise 2D convolution. Similar to the feed-forward case, we only consider a single channel input $x$ (which is $x_{1:N,d}$ formally), reshaped as a $1 \times h \times w$ tensor where $(H, W) = (H/P, W/P)$ is the resolution of patch features. Then, the mixer layer is:

$$f_{\text{mix}}(x) := W_{\text{conv}} * x$$  \hspace{1cm}\text{(9)}$$

where $W_{\text{conv}} \in \mathbb{R}^{1 \times 1 \times S \times S}$ is a kernel matrix with size $S$ and $*$ denotes convolution operation. Here, we consider the linearized version of kernel matrix (i.e., Toeplitz matrix) that substitutes the convolution to the matrix multiplication. Then, one can interpret the mixer layer as:

$$f_{\text{mix}}(x) = W_{\text{linear}} \cdot \hat{x}$$  \hspace{1cm}\text{(10)}$$

where $W_{\text{linear}} \in \mathbb{R}^{N \times N}$ is the corresponding matrix of $W_{\text{conv}}$ and $\hat{x}$ is a reshaped tensor of $x$ of size $N \times 1$, where $N = H \cdot W$. Here, one can directly multiply the reweighting mask $M$ to define the ReMixer for convolution:

$$f_{\text{remix}}(x) := (M \odot W_{\text{linear}}) \cdot \hat{x}$$  \hspace{1cm}\text{(11)}$$

where $\odot$ is an element-wise product. We also compare ReMixer with the models using different kernel matrix for each channel, i.e., $W_{\text{conv}} \in \mathbb{R}^{D \times 1 \times S \times S}$ (see Appendix D).
Table 1: ReMixer using various object labelers evaluated on the Background Challenge benchmark. ‘+’ denotes the modules added to the baseline (not accumulated), and parenthesis denotes the gain of each module. ReMixer consistently improves the classification accuracy (↑) and background robustness (↓) of various patch-based models: DeiT, MLP-Mixer, and ConvMixer.

<table>
<thead>
<tr>
<th>Patch labeler</th>
<th>Original (↑)</th>
<th>Only-BG-B (↓)</th>
<th>Only-FG (↑)</th>
<th>Mixed-Same (↑)</th>
<th>Mixed-Rand (↑)</th>
<th>BG-Gap (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-S</td>
<td>82.69</td>
<td>39.46</td>
<td>57.63</td>
<td>73.88</td>
<td>50.49</td>
<td>23.39</td>
</tr>
<tr>
<td>+ ReMixer (ours) BigBiGAN</td>
<td>83.88 (+1.29)</td>
<td>36.59 (+2.87)</td>
<td>60.10 (+2.47)</td>
<td>75.95 (+2.07)</td>
<td>54.07 (+3.58)</td>
<td>21.88 (+5.51)</td>
</tr>
<tr>
<td>+ ReMixer (ours) ReLabel</td>
<td>86.32 (+3.63)</td>
<td>31.78 (+5.28)</td>
<td>61.93 (+4.80)</td>
<td>78.44 (+5.56)</td>
<td>56.74 (+6.25)</td>
<td>21.70 (+6.89)</td>
</tr>
<tr>
<td>MLP-Mixer-S/16</td>
<td>-</td>
<td>84.99</td>
<td>40.96</td>
<td>63.63</td>
<td>76.52</td>
<td>54.72</td>
</tr>
<tr>
<td>+ ReMixer (ours) BigBiGAN</td>
<td>86.30 (+1.31)</td>
<td>36.64 (+2.52)</td>
<td>66.15 (+2.32)</td>
<td>78.47 (+1.95)</td>
<td>58.54 (+2.82)</td>
<td>19.93 (+3.87)</td>
</tr>
<tr>
<td>+ ReMixer (ours) ReLabel</td>
<td>87.68 (+2.82)</td>
<td>27.28 (+1.64)</td>
<td>67.88 (+2.45)</td>
<td>79.43 (+2.51)</td>
<td>60.44 (+5.72)</td>
<td>18.99 (+2.84)</td>
</tr>
<tr>
<td>ConvMixer-512/8</td>
<td>-</td>
<td>86.35 (+0.36)</td>
<td>37.09 (+4.29)</td>
<td>70.03 (+3.19)</td>
<td>80.27 (+4.68)</td>
<td>59.19 (+2.20)</td>
</tr>
<tr>
<td>+ ReMixer (ours) BigBiGAN</td>
<td>86.35 (+0.36)</td>
<td>37.09 (+4.29)</td>
<td>70.03 (+3.19)</td>
<td>80.27 (+4.68)</td>
<td>59.19 (+2.20)</td>
<td>21.09 (+6.51)</td>
</tr>
<tr>
<td>+ ReMixer (ours) ReLabel</td>
<td>88.49 (+2.17)</td>
<td>35.60 (+1.78)</td>
<td>71.60 (+4.86)</td>
<td>81.70 (+3.11)</td>
<td>63.93 (+9.94)</td>
<td>17.77 (+3.83)</td>
</tr>
</tbody>
</table>

We train the models on the ImageNet-9 (Xiao et al., 2021a), which is a 9 superclass subset of ImageNet. We report the results on the Background Challenge (Xiao et al., 2021a) benchmark to evaluate the background robustness of models. Background Challenge contains 8 datasets: ORIGINAL (↑), ONLY-BG-B (↓), ONLY-BG-T (↓), NO-FG (↓), ONLY-FG (↑), MIXED-SAME (↑), MIXED-RAND (↑), and MIXED-NEXT (↑), where the upper or lower arrows indicate the model

Table 2: Comparison of ReMixer (with ReLabel) vs. ConViT (spatial inductive bias). ‘+’ denotes the modules added to the baseline (not accumulated), and parenthesis denotes the gain of each module.

<table>
<thead>
<tr>
<th>Patch labeler</th>
<th>Original (↑)</th>
<th>Only-BG-B (↓)</th>
<th>Only-FG (↑)</th>
<th>Mixed-Same (↑)</th>
<th>Mixed-Rand (↑)</th>
<th>BG-Gap (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-S</td>
<td>82.69</td>
<td>39.46</td>
<td>57.63</td>
<td>73.88</td>
<td>50.49</td>
<td>23.39</td>
</tr>
<tr>
<td>+ ConViT</td>
<td>85.51 (+2.82)</td>
<td>38.94 (+0.52)</td>
<td>62.15 (+4.52)</td>
<td>76.40 (+2.52)</td>
<td>54.37 (+3.88)</td>
<td>22.03 (-1.36)</td>
</tr>
<tr>
<td>+ ReMixer (ours) BigBiGAN</td>
<td>86.32 (+3.63)</td>
<td>31.78 (+7.68)</td>
<td>61.93 (+4.38)</td>
<td>78.44 (+4.56)</td>
<td>56.74 (+6.25)</td>
<td>21.70 (+4.69)</td>
</tr>
<tr>
<td>+ ReMixer (ours) ReLabel</td>
<td>88.20 (+5.51)</td>
<td>30.79 (+4.67)</td>
<td>66.82 (+9.19)</td>
<td>79.16 (+5.26)</td>
<td>60.52 (+10.03)</td>
<td>18.64 (-4.75)</td>
</tr>
</tbody>
</table>

Table 3: Learned mask scales κ(l) over layers.

<table>
<thead>
<tr>
<th>Layer 1/4</th>
<th>Layer 2/4</th>
<th>Layer 3/4</th>
<th>Layer 4/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-S</td>
<td>1.571</td>
<td>0.783</td>
<td>0.945</td>
</tr>
<tr>
<td>MLP-Mixer-S/16</td>
<td>0.744</td>
<td>0.000</td>
<td>0.001</td>
</tr>
<tr>
<td>ConvMixer-512/8</td>
<td>0.871</td>
<td>2.347</td>
<td>1.498</td>
</tr>
</tbody>
</table>

(a) Original  (b) DeiT-S  (c) ReMixer

Table 4: Saliency map visualization.

3 EXPERIMENTS

We first verify the efficacy of ReMixer in Section 3.2, outperforming both vanilla patch-based models and models considering spatial inductive bias. Then, we demonstrate the robustness of ReMixer on out-of-distribution datasets in Section 3.3. We also show that ReMixer outperforms the method that requires the same-level of supervision in Section 3.4, confirming that the remixing of patch mixing layers gives improvements. The experimental settings are in Section 3.1.

3.1 EXPERIMENTAL SETTINGS

Models and training. We apply ReMixer on three representative patch-based models: DeiT (Touvron et al., 2021a), MLP-Mixer (Tolstikhin et al., 2021), and ConvMixer (Trockman & Kolter, 2022), using self-attention, feed-forward, and convolutional patch mixing layers, respectively. Specifically, we use DeiT-S, MLP-Mixer-S/16, and ConvMixer-512/8. ConvMixer-512/8 has kernel size 9 and patch size 16 following MLP-Mixer-S/16 configurations, and we share the convolution kernel over channels by default, while we provide the unshared results in Appendix D. We follow the default training setup of (Touvron et al., 2021a), but use 256 batch sizes due to memory issues in our GPUs.

Object labels. We consider three object labels: first two are BigBiGAN (Voynov et al., 2021) and ReLabel (Yun et al., 2021), representative methods for binary saliency and multi-class prediction maps, respectively. We also test Bbox+GrabCut, which extracts binary masks from the ground-truth bounding boxes using the GrabCut (Rother et al., 2004) algorithm.

3.2 MAIN RESULTS

Setup. We train the models on the ImageNet-9 (Xiao et al., 2021a), which is a 9 superclass subset of ImageNet. We report the results on the Background Challenge (Xiao et al., 2021a) benchmark to evaluate the background robustness of models. Background Challenge contains 8 datasets: ORIGINAL (↑), ONLY-BG-B (↓), ONLY-BG-T (↓), NO-FG (↓), ONLY-FG (↑), MIXED-SAME (↑), MIXED-RAND (↑), and MIXED-NEXT (↑), where the upper or lower arrows indicate the model
Table 5: Test accuracy of ReMixer (with BigBiGAN) trained on ImageNet-9 and tested on out-of-distribution. ‘+’ denotes the modules added to the baseline, and bold denotes the best results.

<table>
<thead>
<tr>
<th>ImageNet-9</th>
<th>ImageNetV2-9</th>
<th>Real-L-9</th>
<th>Rendition-9</th>
<th>Stylized-9</th>
<th>Sketch-9</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-S</td>
<td>82.69</td>
<td>73.22</td>
<td>80.21</td>
<td>28.60</td>
<td>21.90</td>
</tr>
<tr>
<td>+ ReMixer (ours)</td>
<td><strong>83.88 (+1.19)</strong></td>
<td><strong>75.68 (+2.46)</strong></td>
<td><strong>82.37 (+2.16)</strong></td>
<td><strong>29.43 (+0.83)</strong></td>
<td><strong>24.64 (+2.74)</strong></td>
</tr>
<tr>
<td>MLP-Mixer-S/16</td>
<td>84.99</td>
<td>76.57</td>
<td>82.70</td>
<td>34.25</td>
<td>27.08</td>
</tr>
<tr>
<td>+ ReMixer (ours)</td>
<td><strong>86.30 (+1.31)</strong></td>
<td><strong>76.51 (+0.06)</strong></td>
<td><strong>83.37 (+0.67)</strong></td>
<td><strong>34.29 (+0.04)</strong></td>
<td><strong>27.08 (+1.54)</strong></td>
</tr>
<tr>
<td>ConvMixer-512/8</td>
<td>86.32</td>
<td>76.38</td>
<td>83.16</td>
<td>33.33</td>
<td>24.21</td>
</tr>
<tr>
<td>+ ReMixer (ours)</td>
<td><strong>86.35 (+0.03)</strong></td>
<td><strong>76.97 (+0.59)</strong></td>
<td><strong>83.56 (+0.40)</strong></td>
<td><strong>33.81 (+0.48)</strong></td>
<td><strong>24.94 (+0.73)</strong></td>
</tr>
</tbody>
</table>

Table 6: Test accuracy of ReMixer (with ReLabel) trained on ImageNet. We compare TokenLabeling (TL) and TL+ReMixer (Ours), verifying our remixing scheme on the fair comparison setting.

<table>
<thead>
<tr>
<th></th>
<th>TL</th>
<th>TL + ReMixer</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-S</td>
<td>80.20</td>
<td>81.26 (+1.06)</td>
</tr>
<tr>
<td>DeiT-B</td>
<td>81.17</td>
<td>82.18 (+1.01)</td>
</tr>
</tbody>
</table>

should predict the class well or not, respectively. We also report BG-GAP (↓) which measures the accuracy gap between MIXED-SAME and MIXED-RAND. We omit ONLY-BG-T, NO-FG, and MIXED-NEXT results for the brevity of presentation (see Appendix C for discussion).

Results. Table 1 shows that ReMixer consistently improves classification accuracy and background robustness over various patch-based models and object labels. Table 2 compares ReMixer with ConViT (d’Ascoli et al., 2021), a patch mixing layer with spatial prior. It verifies that the object-centric structure of ReMixer is more effective than spatiality, yet gives orthogonal benefits.

Analysis. Table 3 reports the mask scales $\kappa(I)$ of trained models, averaged by each quarter of layers. The models set higher/lower $\kappa(I)$ for lower/higher layers, i.e., see the objects first then expand its view, like the local-to-global structure of CNNs. ConvMixer sets low $\kappa(I)$ for the early layers since it is hard to understand the objects due to the restricted view of convolution. Figure 4 visualizes the saliency maps (Chefer et al., 2021), verifying that ReMixer gives more object-centric view.

3.3 Robustness of ReMixer

Setup. We evaluate the robustness of ReMixer inferred on unseen out-of-distribution (OOD) data. To this end, we test the ReMixer trained on ImageNet-9 on various OOD datasets: 9 superclass (370 class) subset of ImageNetV2 (Recht et al., 2019), ImageNet-RealL (Beyer et al., 2020), ImageNet-R (Hendrycks et al., 2021), ImageNet-Stylized (Geirhos et al., 2019), and ImageNet-Sketch (Wang et al., 2019), denoted by adding ‘-9’ at the end.

Results. Table 5 shows the OOD generalization results. ReMixer performs well on OOD samples, confirming that both object annotators and learned masks are transferable. We use the BigBiGAN annotator since it gives more robust prediction results than ReLabel.

3.4 Fair Comparisons on Large-Scale Dataset

Setup. We compare ReMixer with TokenLabeling (TL; Jiang et al. (2021)) on ImageNet. TL uses the same extra patch-level label with ReMixer. The experiment settings are same as Section 3.1.

Results. Table 6 presents the comparison of TL and TL+ReMixer (Ours). TL+ReMixer outperforms TL, demonstrating that the guided patch interaction of ReMixer drives the performance gain further.

4 Conclusion

We propose ReMixer, a novel object-centric framework to refine any existing patch-based models. We demonstrate the efficacy of ReMixer on ViT, MLP-Mixer, and ConvMixer, while showing superior (yet compatible) performance over prior works using spatial inductive bias. We hope ReMixer could inspire new research directions for patch-based models and object-centric learning.
ACKNOWLEDGMENTS

This research was supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921). We thank Sihyun Yu, Jihoon Tack, Jaeho Lee, and Jongjin Park for constructive feedback.

REFERENCES


Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE Conference on Computer Vision and Pattern Recognition, 2009.


Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In IEEE International Conference on Computer Vision, 2021.


Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.


Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junskuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *IEEE International Conference on Computer Vision*, 2021.


A RELATED WORK

Patch-based models. Inspired by the success of Transformers (or self-attention; Vaswani et al. (2017)) in natural language processing (Devlin et al., 2019; Brown et al., 2020), numerous works have attempted to extend Transformers for computer vision (Khan et al., 2021). In particular, the seminal work named Vision Transformer (ViT; Dosovitskiy et al. (2021)) discovered that Transformer could achieve state-of-the-art performance, exceeding prior popular convolutional neural networks (CNNs; LeCun et al. (1998)). Thereafter, other studies revealed that different patch mixing layers such as feed-forward (Tolstikhin et al., 2021), convolution (Trockman & Kolter, 2022), or pooling (Yu et al., 2021) show comparable performance to self-attention, hypothesizing that the success of ViT comes from the patch-based architectures. Our work proposes an architecture-agnostic framework to improve the patch-based models by reweighting their patch mixing layers.

Inductive bias for patch-based models. Many patch-based models aim to remove inductive biases by using patch mixing layers without additional structures, e.g., self-attention. While they perform well on large data regimes, recent works reveal that inductive biases are still crucial for patch-based models, especially under limited data (Steiner et al., 2021). Consequently, extensive literature proposed approaches to incorporate additional structures for patch-based models, e.g., spatial structures of CNNs. One line of work aims to design patch mixing layers reflecting inductive biases. For example, ConViT (d’Ascoli et al., 2021) and CoAtNet (Dai et al., 2021) calibrate self-attention with spatial distance between patches, CvT (Wu et al., 2021a) and ConvMixer (Trockman & Kolter, 2022) utilize convolution operation for patch mixing, and AS-MLP (Lian et al., 2021) design a structured operation aggregating the values from different axises. Another line of work build an architecture that combines convolutional or pooling layers with patch mixing layers Liu et al. (2021); Yuan et al. (2021b); Wang et al. (2021); Fan et al. (2021); Heo et al. (2021); Yuan et al. (2021a); Xiao et al. (2021b). Our work falls into the first category; however, we leverage the object structure of images, unlike prior works focused on the spatial inductive bias. Using rich information, our proposed ReMixer outperforms ConViT and CoAtNet, where using both ConViT and ReMixer gives further improvements, implying that two methods contribute to the model differently (see Table 2). We also emphasize that ReMixer can be applied on any patch mixing layers under a common principle, unlike prior works designed for specific layers such as self-attention or feed-forward.

Incorporating object structures. Although objects are the atom of visual scenes, only a limited number of research has leveraged the object structure of images for visual recognition (e.g., classification). This is mainly due to two reasons: (a) the cost of collecting object labels and (b) non-triviality of reflecting object information to the black-box deep learning models. However, both challenges have been relaxed by the rapid advance of deep learning. First, the progress of supervised (He et al., 2017; Carion et al., 2020; Fang et al., 2021), weakly-supervised (Selvaraju et al., 2017; Chefer et al., 2021; Yun et al., 2021), and self-supervised (Voynov et al., 2021; Caron et al., 2021; Mo et al., 2021) detection significantly reduced the cost of object labels. We utilize the pretrained BigBiGAN (Voynov et al., 2021) and ReLabel (Yun et al., 2021) models for our experiments; one could also train weakly- or self-supervised object annotators on their datasets. Second, the patch-based models are well-suited with object information (unlike CNNs) as the patch embeddings preserve their spatial information (Raghu et al., 2021). Using this property, ReMixer adjusts the interaction of patch embeddings using object labels. ORViT (Herzig et al., 2021) also direct ViT to focus on the object regions by creating extra object tokens. However, their goal is to guide video Transformers to track the trajectory of objects and are less suited for image classification. On the other hand, TokenLabeling (Jiang et al., 2021) implicitly utilizes the object information by using them as additional supervision for patch embeddings; it provides an orthogonal gain from ReMixer (see Table 6). We finally note that several works (Locatello et al., 2020; Wu et al., 2021b; Kipf et al., 2021) aim to disentangle the object features explicitly. However, they do not scale yet to the complex real-world images due to the strong constraints in the model. In contrast, ReMixer can be applied to any existing patch-based models with minimal modification.
B  Obtaining object labels

One possible concern for ReMixer is the labeling cost of patch-wise object labels $y \in \mathbb{R}^{N \times K}$. However, we claim that this cost is not a critical issue since one can utilize the pretrained machine annotators. Notably, the object labels extracted from the models trained on some (source) datasets are still helpful for different (target) downstream datasets (see Section 3.3). In the remaining section, we describe two types of machine annotators: binary saliency and multi-class prediction with discussion on their pros and cons.

**Binary saliency map.** We first consider binary saliency maps, i.e., indicating whether the given pixel is object or background. There is a tremendous amount of work on extracting saliency maps in a self-supervised (Voynov et al., 2021; Caron et al., 2021; Mo et al., 2021) or weakly-supervised (i.e., using class labels; Selvaraju et al. (2017); Chefer et al. (2021)) manner. We use the saliency model called BigBiGAN (Voynov et al., 2021), which finds the salient region using BigGAN (Brock et al., 2019) trained on the ImageNet (Deng et al., 2009) dataset. We average the pixel-wise saliency values in the patch to get a soft label $y_n \in [0, 1]$, and use the $l_1$-distance (between the object labels of two patches) in Eq. (3).

**Multi-class prediction map.** We also consider multi-class prediction maps, i.e., pixel- or patch-level semantic segmentation. However, since segmentation labels are expensive, we utilize ReLabel (Yun et al., 2021), which predicts the dense label maps from an image classifier by applying the classifier on penultimate spatial features (i.e., before global average pooling). We use the NFNet-F6 (Brock et al., 2021) model trained on the ImageNet dataset to extract ReLabel map and apply region-of-interest (RoI) pooling (Girshick, 2015) to extract the patch label $y_n \in \mathbb{R}^K$. Here, computing $l_p$-distance between the object labels of patches is expensive since it handles a $N^2 K$ tensor. Instead, we use the cosine distance, efficiently computed by matrix multiplication.

**Comparison of two approaches.** Multi-class prediction map contains richer semantics and thus provide more informative reweighing masks. As a result, ReLabel (used as the object labels for ReMixer) has shown better results than BigBiGAN in our experiments, especially when the downstream tasks are close to the source dataset, ImageNet. However, ReLabel is often prone to distribution shifts, while BigBiGAN provides consistent gain under the same setup. Intuitively, predicting the salient objects is easier to generalize than predicting classes. Thus, we suggest the users choose binary vs. multi-class labels following their robustness vs. in-distribution accuracy trade-off.

We finally remark that while we utilize the pretrained annotators (from different source datasets) for simplicity, one can also apply our method without external datasets. Recall that the annotators we consider are trained in an unsupervised or weakly-supervised manner; it does not require ground-truth dense supervision and can be solely extracted from the downstream dataset.
C DISCUSSION ON BACKGROUND CHALLENGE

Figure 1 visualizes the datasets in the Background Challenge benchmark, which provides various combinations of foregrounds and backgrounds: ORIGINAL (↑), ONLY-BG-B (↓), ONLY-BG-T (↓), NO-FG (↓), ONLY-FG (↑), MIXED-SAME (↑), MIXED-RAND (↑), MIXED-NEXT (↑). The upper or lower arrows indicate the model should predict the class well or not, respectively. We omit ONLY-BG-T, NO-FG, and MIXED-NEXT results for brevity of presentation and due to the following reasons:

- MIXED-NEXT shows almost the same trend with MIXED-RAND.
- NO-FG is controversial to be predicted or not since the image contains the black shape of the object.
- ONLY-BG-T accuracy is proportional to the ORIGINAL accuracy. However, we remark that ReMixer only increases the ONLY-BG-T accuracy a little while ORIGINAL a lot, compared to the baseline models.

D SHARING WEIGHTS FOR CONVMIXER

Table 7: Comparison of ConvMixer sharing (white line) and not sharing (gray line) kernels over channels. We add `-Full' to denote the latter model, which uses $D$ times for parameters for patch mixing layers where $D$ is the number of channels.

<table>
<thead>
<tr>
<th></th>
<th>Original (↑)</th>
<th>Only-BG-B (↓)</th>
<th>Only-FG (↑)</th>
<th>Mixed-Same (↑)</th>
<th>Mixed-Rand (↑)</th>
<th>BG-Gap (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvMixer-512/8</td>
<td>86.32</td>
<td>41.38</td>
<td>66.84</td>
<td>78.59</td>
<td>56.99</td>
<td>21.60</td>
</tr>
<tr>
<td>+ ReMixer (ours)</td>
<td>88.49 (+2.17)</td>
<td>35.60 (-5.78)</td>
<td>71.60 (+4.76)</td>
<td>81.70 (+4.11)</td>
<td>63.93 (+6.94)</td>
<td>17.77 (-3.83)</td>
</tr>
<tr>
<td>ConvMixer-512/8-Full</td>
<td>88.67 (+2.35)</td>
<td>40.20 (-1.18)</td>
<td>70.72 (+3.88)</td>
<td>81.31 (+2.72)</td>
<td>62.84 (+5.85)</td>
<td>18.47 (-3.13)</td>
</tr>
</tbody>
</table>

We adopt ConvMixer (Trockman & Kolter, 2022) to share the convolution kernels over channels to reduce memory and computation. Remarkably, ReMixer applied on ConvMixer is comparable or even outperforms ConvMixer-Full, which uses different kernels for each channel, i.e., uses $D = 512$ times parameters for patch mixing layers.