Appendix

A1. The selection of the backbone

The proposed R-Block adopts a residual-like structure. This makes it most suitable for residual-style CNN backbones, of which the most commonly used are ResNet50 [S1] and ResNeXt50 [41]. The ResNeXt50 combines the architecture of ResNet and Inception nets, which is usually adopted as a higher baseline / backbone in comparison to its ResNet50 counterparts [41]. In our preliminary experiments as shown in Table S1, we find that the ResNeXt50 performs better than ResNet50 on both ESC-50 and DCASE2018 datasets, thus the ResNeXt50 is chosen as the backbone. Another thing worth noting in Table S1 is that, the R-Block is shown effective for both ResNet and ResNeXt architectures, demonstrating its robustness to the backbones.

Table S1 Performance of the different backbones with or without the R-Block

Dataset	Model	ACC(%)
ESC-50	ResNet50	90.0
	ResNet50 + RBlock	**91.0**
	ResNeXt50	90.3
	ResNeXt50 + RBlock	**91.9**
DCASE2018	ResNet50	76.65
	ResNet50 + RBlock	**78.12**
	ResNeXt50	77.24
	ResNeXt50 + RBlock	**79.15**

[S1] He, Kaiming, et al. "Identity mappings in deep residual networks." European conference on computer vision. Springer, Cham, 2016.

[41] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He, “Aggregated residual transformations for deep neural networks,” in Proc. CVPR, 2017, pp. 1492–1500.

A2. The selection of global pooling layers

The global max pooling layer (GMP) and the global average pooling layer (GAP) are the two most commonly used global pooling layers to obtain a vectorized representation in CNNs before applying the fully-connected layers. To choose the suitable global pooling layers, we constructed two backbone models ResNeXt50_GMP and ResNeXt50_GAP and evaluated their performance on each dataset. The results are shown in Table S2. As can be seen, the GMP performs better on ESC-50 while GAP performs better on DCASE2018.

Table S2 Effects of the global pooling layers

Dataset	Model	ACC(%)
ESC-50	ResNeXt50_GMP	**90.3**
	ResNeXt50_GAP	89.3
DCASE2018	ResNeXt50_GMP	75.97
	ResNeXt50_GAP	**77.24**

A3. The selection of the insert point

Since the increase in the number of layers in the CNN backbone can usually be seen as a process of semantic abstraction from low to high [S2], the position of the insertion point determines the semantic level at which the R-Block conducts the relation modeling. To determine the optimal level for relation modeling, we performed preliminary experiments on each dataset. More specifically, for the ResNeXt backbone as shown in Table S3, an R-Block is inserted after a certain *residual block* (highlighted on gray backgrounds in Table S3), then the relation augmented model is evaluated on each dataset.

Table S3 Configurations of the backbone model[1].

Stage	ResNeXt Backbone

[1] For a detailed description of the structure of the ResNeXt50, readers may refer to [41]. As explained in the

conv1	7×7, 64, stride 2
conv2	3×3 max pool, stride 2 $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C=32 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C=32 \\ 1\times1, 512 \end{bmatrix} \times 4$
conv4	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C=32 \\ 1\times1, 1024 \end{bmatrix} \times 6$
fc	global pooling layer (GAP/GMP) fc layer, softmax

The effects of the insert point on ESC-50 and DCASE2018 datasets are shown in Figure S1 and Figure S2, respectively. In the figures, the insert point is specified by a tuple of *two* elements, which indicate the index of the convolutional stage and the residual block. From top to bottom in the figures, the insert points are sorted from low to high accuracy. And the red dashed line indicates the accuracy of the baseline (i.e., without the R-Block). From the figures, we can see that the optimal insert points for ESC-50 and DCASE2018 are (4,5) and (2,3) respectively.

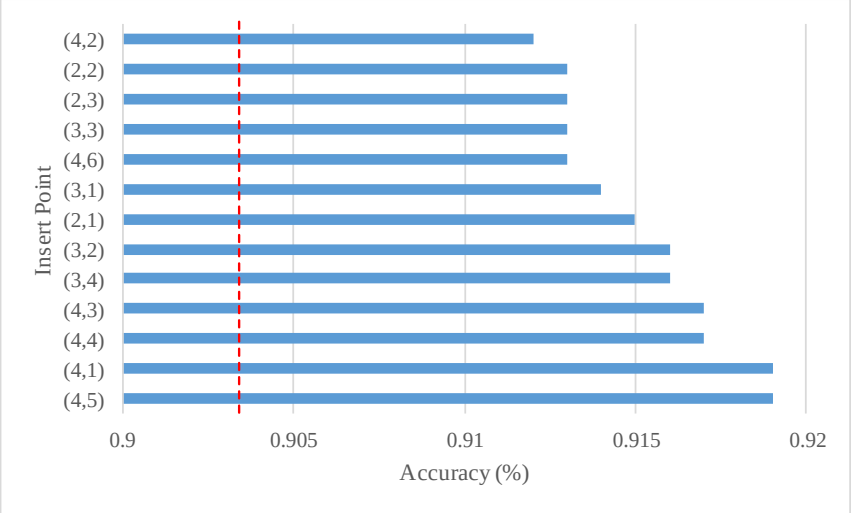

Figure S1 The effects of the insert point of the R-Block on ESC-50 dataset.

manuscript, the conv5 stage was removed from the backbone, which was found to boost generalization.

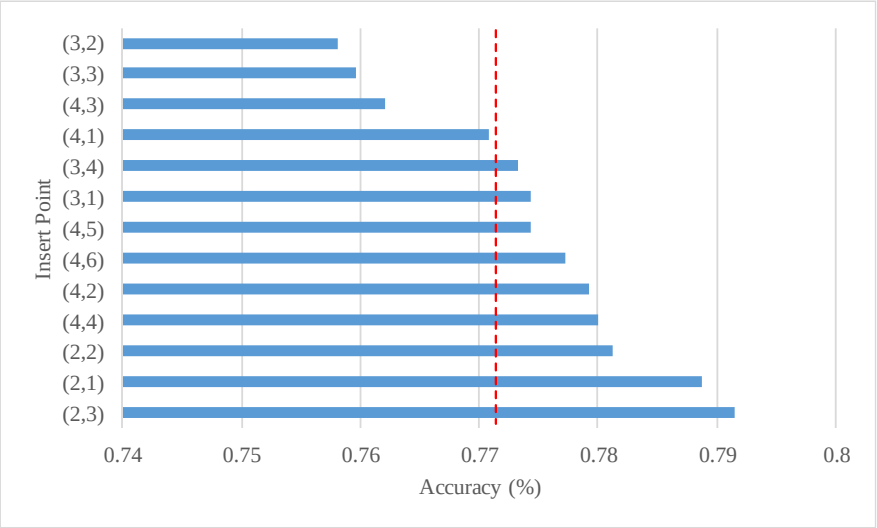

Figure S2 The effects of the insert point of the R-Block on DCASE2018 dataset.

[S2] Ma, Chao, et al. "Hierarchical convolutional features for visual tracking." Proceedings of the IEEE international conference on computer vision. 2015.

[41] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in Proc. CVPR, 2017, pp. 1492–1500.

A4. How to interpret the relation heatmaps?

i. How were the relation heatmaps generated?

This involves three steps, indicated by the arrows with circled numbers (①②③) in Figure S3,

①. The spectrogram is firstly processed by the CNN into a collection of nodes, which are inherently arranged on a time-frequency 2D grid. Each node (e.g., $\mathbf{x}_i$ in Figure S3(b)) represents the CNN feature of certain local patch (the boxed patch in Figure S3(a)) on the spectrogram.

②. Then, the inter-node relations are obtained through end-to-end learning. In Figure S3(c), for a particular selected node, its relations $\mathbf{E}_i$ (after training) towards all the other nodes are shown as pink arrows. The strength of each relation (connection) is reflected by the thickness of the pink arrows. Notably, for different selected nodes, the learned relation $\mathbf{E}_i$ is usually different. In addition, with different relation functions (either in the form of $\mathbf{E}_i$ in Eq. (1) or $\mathbf{E}_i^{\dagger}$ in Eq. (4)), the learned relations will also be different.

③. Since showing relations $\mathbf{E}_i$ in the node space is not as intuitive as in the spectrogram space. Therefore, we show the relation $\mathbf{E}_i$ in the spectrogram space by reshaping and interpolating $\mathbf{E}_i$ to the size of the spectrogram, which results in the heatmap as shown in Figure S3 (d).

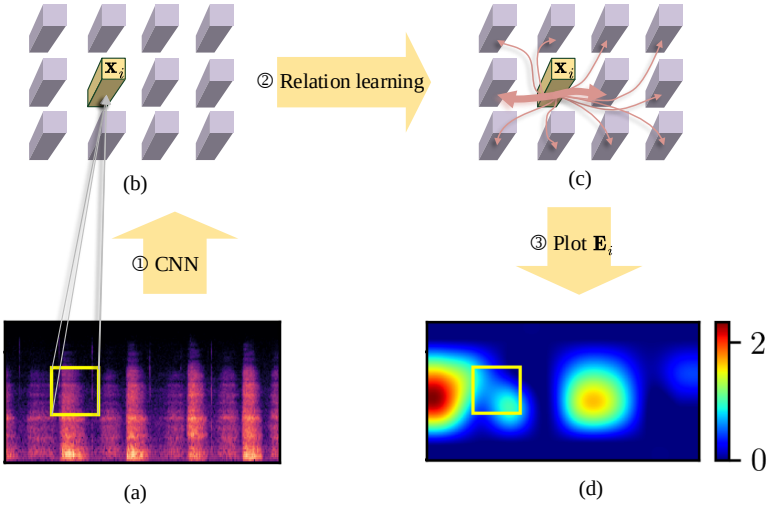

Figure S3 How is the relation heatmap generated?
(a) spectrogram (b) CNN nodes arranged on a 2D time-frequency grid (c) learned inter-node relation of a selected node $\mathbf{x}_i$ (d) relation heatmap of a selected node $\mathbf{x}_i$

To summarize, the relation heatmap shows the learned relation of a selected node to all the other nodes, where each node is the CNN feature of certain local patch on the spectrogram. The inter-node relations at the CNN feature level (shown in Figure S3(c)) reflect the inter-patch relations at the spectrogram level (shown in Figure S3(d)).

ii. The general way to interpret the relation heatmap.

In general, we can interpret any relation heatmap (e.g., Figure S3(d)) as follows: *the node centered at the boxed patch is strongly related to the nodes centered at the hot (red) spots.* More implications can be obtained if we further consider how the relation is utilized in the R-Block. As described in Section II.C, the proposed information aggregation operator generates the relation augmented representation of a node by aggregating node features from its related nodes. This leads to a deeper interpretation: *When representing the node centered at the boxed patch, the R-Block draws information from the highly relevant (red) area.*

iii. What to look at in a relation heatmap?

To understand the implications of the heatmaps, it helps to look at the heatmaps with a few questions in mind. For example: Which spots does the selected node (centered at the boxed patch) draws connection to? How the heatmap varies when the different nodes are selected? How the heatmap varies when different forms ($\mathbf{E}_i$ or $\mathbf{E}_i^{\dagger}$) of the relation function are used? What does the relation heatmap look like for different classes of sound events?

A5. What does heatmap activation mean in silence part?

In fact, this question implies two sub-questions that need to be answered.

i. Is silence useful for recognizing sound events?

Let us consider two types of silence. The first one is the natural pause / interval of a sound event. For example, the silence between the clock ticks, frog calling, water drops, etc. This type of interval silence reflects the natural rhythm of the sound sources, thus it is an important part of sound event semantics.

The second one is the preceding or trailing silence of a recording in the datasets. This type of silence is usually human-introduced when creating the datasets, which may be arbitrary in length. At first glance, this type of silence seems non-informative and useless. However, this is not the case. The preceding or trailing silence parts can provide a common 'ground' (or background) for learning models to make sense of non-silence parts in the recording, such as the salience or loudness of a sound event. Furthermore, when the silence length makes distributional difference for various classes in the datasets, the learning model may take advantage of this difference to discriminate sound classes. This is a typical example of dataset bias [S3] (undesired of course). This silence related bias in the dataset (if it exists) can create 'shortcuts' [S4] that data-driven models may take advantage of.

ii. How to interpret the heatmap activation in the silence part?

As explained above, silence could be useful for learning models to recognize sound events. For the interval silences, the proposed R-Block may use them to make sense of rhythms. For the preceding and trailing silence, the R-Block may use relation between sound event patterns to make sense of loudness / salience. For either cases, the R-Block gives opportunity to exploit the relation between the silence and the non-silence parts. In these situations, the relation heatmap may activate in the silence part, which means to draw information from it. This is the expected behavior of the RBlock.

[S3] Torralba, Antonio, and Alexei A. Efros. "Unbiased look at dataset bias." CVPR 2011. IEEE, 2011.

[S4] Bahng, Hyojin, et al. "Learning de-biased representations with biased representations." International Conference on Machine Learning. PMLR, 2020.