Table 8. **Comparison with hand-crafted spatial alignment methods**. The experiments on MS-COCO is based on Mask R-CNN architecture with the ResNet-50 FPN backbone and the 1×/2× schedule [41], following [40, 43, 44].

Method	Epoch	IN-1K		MS-COCO (1× Schedule)						MS-COCO (2× Schedule)					
		Acc@1	Acc@5	AP^{b}	$\text{AP}^{\text{b}}_{50}$	$\text{AP}^{\text{b}}_{75}$	AP^{m}	$\text{AP}^{\text{m}}_{50}$	$\text{AP}^{\text{m}}_{75}$	AP^{b}	$\text{AP}^{\text{b}}_{50}$	$\text{AP}^{\text{b}}_{75}$	AP^{m}	$\text{AP}^{\text{m}}_{50}$	$\text{AP}^{\text{m}}_{75}$
DenseCL [40]	200	63.6	85.8	40.3	59.9	44.3	36.4	57.0	39.2	41.2	61.9	45.1	37.3	58.9	40.1
ReSim [43]	200	66.1	-	39.8	60.2	43.5	36.0	57.1	38.6	41.4	61.9	45.4	37.5	59.1	40.3
LEWEL$_M$	200	68.1	88.6	40.0	59.8	43.7	36.1	57.0	38.7	-	-	-	-	-	-
LEWEL$_B$	200	**72.8**	**91.0**	**41.3**	**61.2**	**45.4**	**37.4**	**58.3**	**40.3**	**42.2**	**62.3**	**46.1**	**38.2**	**59.6**	**41.1**
PixelPro [44]	400	60.2	83.0	41.4	61.6	45.4	37.4	-	-	-	-	-	-	-	-
LEWEL$_B$	400	**73.8**	**91.7**	**41.9**	**62.4**	**46.0**	**37.9**	**59.3**	**40.7**	**43.4**	**63.5**	**47.7**	**39.1**	**60.7**	**42.4**

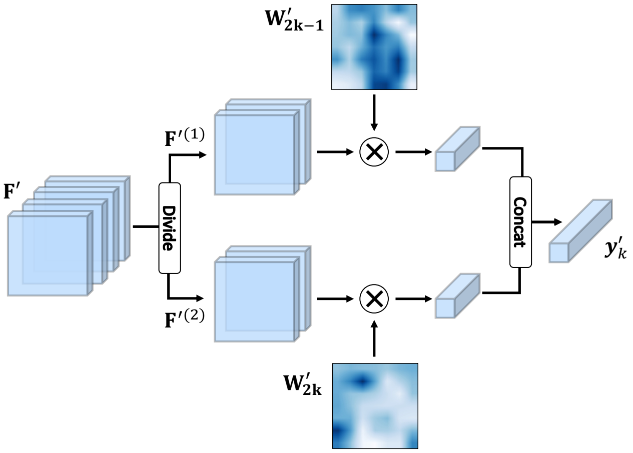

Figure 4. **An illustration of the channel grouping scheme**. Here the number of groups h is set to 2 for simplicity.

A. Additional Illustration of LEWEL

A.1. Illustration of the channel grouping operation.

Instead of using one alignment map to aggregate one aligned representation, we introduce a grouping scheme that divides the channels of $\mathbf{F}'$ uniformly into h equal-size groups, that is $\mathbf{F}' = [\mathbf{F}'^{(1)}, \cdots, \mathbf{F}'^{(h)}]$ where $[\cdot]$ denotes the concatenation operation. Given the alignment maps $\{\mathbf{W}'_k\}_{k=1}^{d}$, we can accordingly aggregate a set of aligned representations $\{\boldsymbol{y}'_k : \boldsymbol{y}'_k \in \mathbb{R}^D\}_{k=1}^{d/h}$, and

$$\boldsymbol{y}'_k = [\mathbf{W}'_{(k-1)\times h+1} \otimes \mathbf{F}'^{(1)}, \cdots, \mathbf{W}'_{k\times h} \otimes \mathbf{F}'^{(h)}], \forall k, \quad (6)$$

where $\otimes$ is the spatial aggregation operation defined in Eq. (1). For a more intuitive illustration, we summarize the overview of this channel grouping scheme in Fig. 4, where we set the number of groups $h = 2$ for simplicity.

B. Additional Experiment Results

B.1. Comparison with hand-crafted spatial alignment methods

Prior methods [40, 43, 44] are tailored for dense prediction and use pre-defined manual rules to match corresponding pixels. They emphasized on local feature learning and largely ignored the learning of global features that is also important in transferring to both classification and detection tasks (see Sec. 3.4 in [44]). In contrast, LEWEL is a generic method and benefits both image-level and dense predictions. LEWEL leverages the global projection head to predict the spatial alignment maps such that couples the learning of global features and aligned features. Our experimental results in Tab. 8 show that LEWEL significantly outperforms [40, 43, 44] in terms of classification by up to 13% while performing on par with or even better than [40, 43, 44] on detection/segmentation under 1×/2× training schedule, highlighting the generalization ability of LEWEL.

C. Additional Analyses on LEWEL

C.1. Visualization of the alignment maps.

In Fig. 5, we visualize the alignment maps predicted by LEWEL$_M$ on the ImageNet-1K validation set, which suggest that LEWEL can automatically find semantically consistent alignments for self-supervised learning. Specifically, we observe that the alignment maps may activate on the region of an object (*e.g.*, the visualization in the 3rd column, 1st-2nd rows), the region of multiple objects (*e.g.*, the visualization in the 3rd column, 3rd-4th rows), and on the global region (*e.g.*, the visualization in the 3rd column, 5th-6th rows). The visualization demonstrates that LEWEL is able to learn on both local and global representations simultaneously by manipulating the alignment maps.

C.2. Computational cost.

In Tab. 9, we compare the training time of LEWEL with that of the baselines. The comparison is performed on a single machine with eight V100 GPUs, CUDA 10.1, Py-

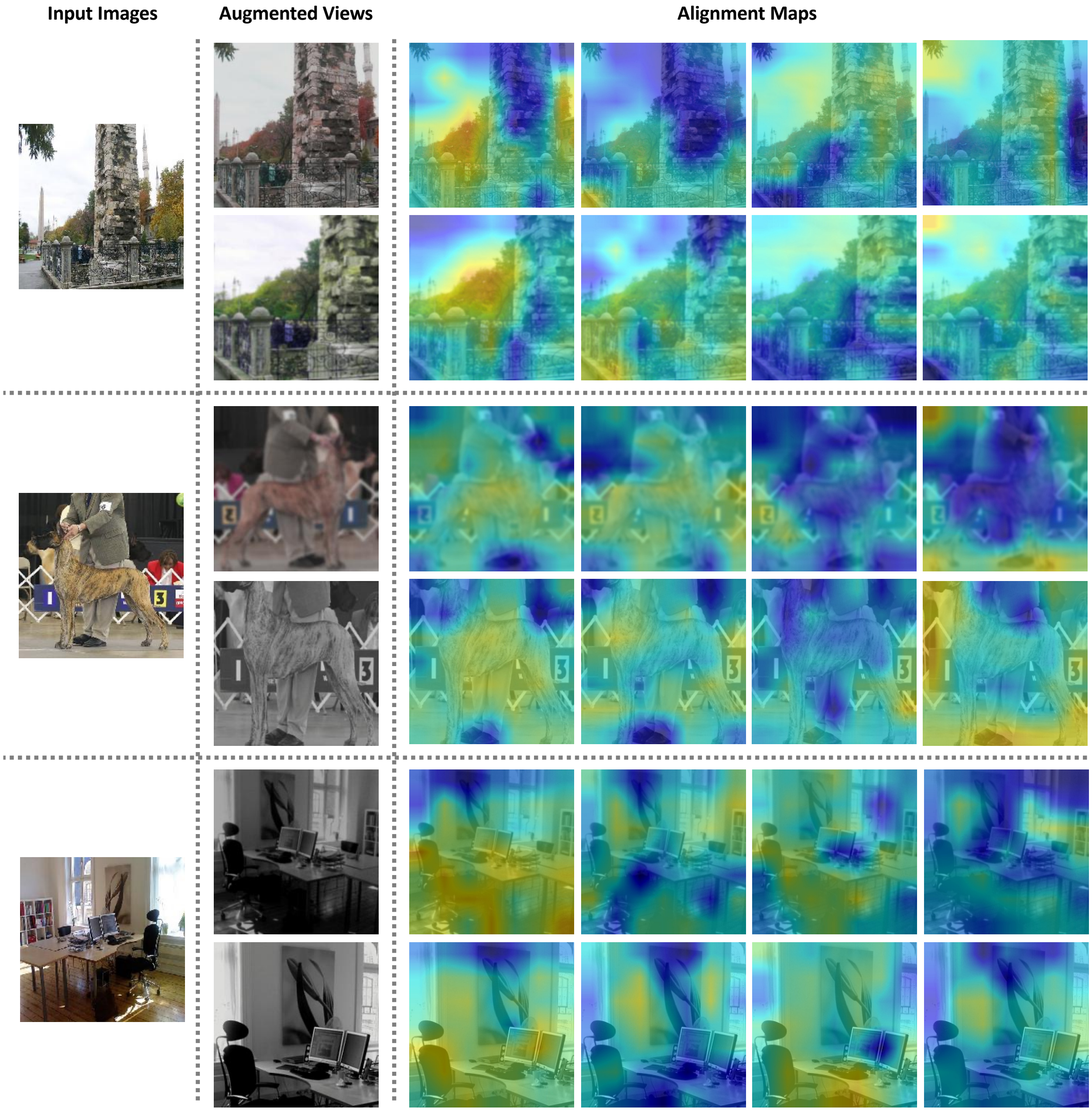

Figure 5. **Visualization of the alignment maps** predicted by LEWEL$_M$ on the ImageNet-1K validation set. First column: the input images from the ImageNet validation set. Second column: augmented views generated by random data augmentations. The rest columns: alignment maps predicted by LEWEL$_M$ based on augmented views. The visualization shows that LEWEL automatically finds semantically consistent alignments for self-supervised learning.

Torch 1.8, and the automatic mixed precision (AMP) training[1]. For all methods, we report their training time of one epoch. According to the results in Tab. 9, we can observe that the training time of our method is only marginally increased compared to the baselines. To be more concrete, LEWEL$_B$ requires only $\sim$4% additional overhead compared with BYOL while significantly outperforming BYOL, which demonstrates the efficiency of LEWEL.

C.3. Limitations.

One disadvantage of LEWEL is that the number of aligned representations largely depends on the output di-

[1] We find that AMP has little impact on the training time of MoCov2/LEWEL$_M$ but reduces that of BYOL/LEWEL$_B$ by $\sim$40%.

Table 9. **Training time comparison** on with MoCov2 and BYOL on the ImageNet-1K dataset. The comparison is performed on a single machine with eight V100 GPUs using the automatic mixed precision (AMP) training in PyTorch 1.8.

Method	Training Time/Epoch	Top1 Acc.@200 Epochs
MoCov2	1213s	64.5
LEWEL$_M$	1222s	66.1
BYOL	1141s	70.6
LEWEL$_B$	1191s	71.5

mensionality d of the coupled projection head g. In the cases where d is very large, LEWEL might incur additional training overheads. Nevertheless, we note that our channel grouping scheme can mitigate this drawback by using a larger number of groups to reduce the number of aligned representations.