A. Other Attempts to Avoid Collapsing

The standard consistency-based method (described in Section 3.1 and 3.2) suffers from collapsing problem in 2D pose estimation. In this section, we also investigate the effects of other factors like the number of labeled examples, and unsupervised loss weight on the collapsing problem.

The following experiments show that adjusting these factors can not fully solve the collapse problem. In all experiments, the response to the unlabeled sample always has a downward trend, and the final model accuracy is lower than the initial supervised model. It is worth noting that the augmentation parameters $\eta$ and $\eta'$ are sampled in the same distribution (See Section 3.1) in these experiments, and our easy-hard augmentation strategy is not involved.

### A.1. Decrease Unsupervised Loss Weight $\lambda$

![Figure 9. Effect of unsupervised loss weight $\lambda$. With small consistency coefficient, the response for unlabeled examples still decreases but in a relatively slow speed.](image)

The coefficient $\lambda$ controls the importance of consistency loss $L_u$. One possible opinion is that too large weight $\lambda$ leads to the instability in training process. The effect of decreasing $\lambda$ is shown in Figure 9. $\lambda = 1$ is the default setting and $\lambda = 0$ means that only supervised loss $L_s$ is used. All of these are trained with the same number of epochs. The result indicates that lower coefficient slightly reduce the degree of degradation but the degradation still happens. With different coefficient values, the response and AP are always worse than the supervised model.

### A.2. Increase Labeled Examples

<table>
<thead>
<tr>
<th>Labels</th>
<th>Response</th>
<th>AP</th>
<th>Supervised AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCO 1K</td>
<td>0.0002</td>
<td>0.8</td>
<td>31.5</td>
</tr>
<tr>
<td>COCO 5K</td>
<td>0.0023</td>
<td>13.0</td>
<td>46.4</td>
</tr>
<tr>
<td>COCO 10K</td>
<td>0.0105</td>
<td>26.2</td>
<td>51.1</td>
</tr>
<tr>
<td>COCO 150K (Full)</td>
<td>0.0122</td>
<td>46.8</td>
<td>67.1</td>
</tr>
</tbody>
</table>

Table 9. Increasing the labeled examples do not address the collapsing problem. “Supervised AP” represents the supervised model using only labeled examples. Even with sufficient labels, the SSL training still degrades the performance compared to supervised model.

Another question is whether too few labels causes the collapsing. As is shown in Table 9, with the increase of labels, the validation performance has improved (From 0.8% to 46.8%). However, it did not fully solve the collapsing problem, as the response level is still low and performance is degraded. The results indicate that the even under a large number of labeled samples, consistency loss still drives the network to generate low-response heatmap for unlabeled examples, and finally degrade the generalization performance of model. In contrast, our method can significantly improve the performance regardless of the number of labeled examples.

B. Dual Networks

More details about dual networks, including the algorithm flow, the motivation and advantages, are provided in this part. In dual networks learning, we jointly learn two networks $f_\theta$ and $f_\xi$. We first train them separately on labeled images from different initialization. Then we jointly train them on both labeled and unlabeled images. The Figure 10 demonstrates the framework of our method (dual networks) and Algorithm 1 describes the algorithm process.

In summary, for an unlabeled image. (1) First generate easy augmentation $I_e$ and hard augmentation $I_h$. (2) Then the easy augmentation is fed into two networks to produce...
targets \( H_{\theta,e} \), \( H_{\xi,e} \). (3) And the hard augmentation is used to produce student predictions \( H_{\theta,h} \), \( H_{\xi,h} \). (4) The target \( H_{\theta,e} \) guides the prediction \( H_{\xi,h} \) to optimize \( f_\xi \). And the target \( H_{\xi,e} \) guides the prediction \( H_{\theta,h} \) to update \( f_\theta \).

Such a symmetrical structure is very simple, but has several advantages. (1) It ensures that the teacher prediction is statistically more accurate than the student prediction, which avoid the collapsing problem. (2) The teacher and student come from two different and independent networks, which avoid the collapsing problem. (2) The teacher and student come from two different and independent networks, which avoid the collapsing problem. (3) The two networks can be updated and improved at the same time, and no network will become the bottleneck. In fact, the accuracy of two networks are very similar after training if they use the same network structure.

### C. Additional Experimental Results

#### C.1. Augmentation Hyper-Parameters

We study the effect of augmentation hyper-parameters in this part. The dual networks (with "easy-hard" augmentation method) is used as training method. The COCO 1K subset is used as labeled set and COCO TRAIN is used as unlabeled set. Joint Cutout and RandAugment are only used in hard augmentation. The best value of hyper-parameters is selected and used as the default setting.

**Joint Cutout** In Joint Cutout, \( m \) detected joints are randomly selected to be masked. More masked regions will increase the difficulty of prediction. We conduct experiments to analyze the effect of hyper-parameter \( m \). The experiment (Figure 11) shows that \( m = 5 \) achieves the best performance. Further increasing \( m \) can not boost accuracy, which is maybe because the the remaining image region is too small to have effective information for localization.

**Rand Augment** We also study the effect of the distortion magnitude [1] in RandAugment. Two augmentation transformations are randomly sampled and applied sequentially. The Figure 12 shows that the optimal magnitudes is 20. For simplicity, the geometric transformations in RandAugment like "Rotate", "TranslateX", etc. are excluded.

---

**Algorithm 1 Dual Networks Learning**

**Input:** \( L = \{ \{ I_s, H_s \} \}_{s=1}^k \): Batch of labeled data.

**Input:** \( U = \{ I_u \}_{u=1}^m \): Batch of unlabeled data.

**Input:** \( \theta, \xi \): Pre-trained model parameters

**Output:** \( \theta, \xi \): Updated model parameters

1: \( L_s = 0, L_\xi = 0, L_\eta = 0 \)

2: for each \( \{ I_s, H_s \} \in L \) do

3: Calculate supervised loss

4: \( L_s = L_s + ||f(I_s, \theta) - H_s||^2 + ||f(I_s, \xi) - H_\xi||^2 \),

5: end for

6: for each \( I_u \in U \) do

7: Randomly sample augmentations \( \eta_e \) and \( \eta_h \)

8: Compute teacher predictions \( H_{\theta,e}, H_{\xi,e} \) by

9: Compute student predictions \( H_{\theta,h}, H_{\xi,h} \) by

10: Calculate unsupervised loss \( L_\theta \) and \( L_\xi \) by

11: end for

12: \( L_{total} = L_s + L_\theta + L_\xi \)

13: update \( \theta, \xi \) by minimizing \( L_{total} \)
Table 10. The results on the VAL set of the MPII dataset. Our model is trained on the MPII TRAIN with labels and the AIC without labels. We compare to a supervised baseline that is trained only on the MPII TRAIN. “Extra Labels” means we use the labels from the AIC dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Extra Labels</th>
<th>Head</th>
<th>Shoulder</th>
<th>Elbow</th>
<th>Wrist</th>
<th>Hip</th>
<th>Knee</th>
<th>Ankle</th>
<th>Total</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleBaseline</td>
<td>ResNet-101</td>
<td>X</td>
<td>97.1</td>
<td>94.9</td>
<td>88.1</td>
<td>82.5</td>
<td>87.7</td>
<td>83.2</td>
<td>79.2</td>
<td>88.1</td>
<td></td>
</tr>
<tr>
<td>SimpleBaseline</td>
<td>ResNet-101</td>
<td>✓</td>
<td>97.3</td>
<td>96.2</td>
<td>91.1</td>
<td>86.9</td>
<td>89.3</td>
<td>87.9</td>
<td>83.8</td>
<td>90.8</td>
<td>2.7</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-101</td>
<td>X</td>
<td>97.1</td>
<td>96.1</td>
<td>90.4</td>
<td>85.0</td>
<td>89.2</td>
<td>86.1</td>
<td>81.7</td>
<td>89.8</td>
<td>↑1.7</td>
</tr>
<tr>
<td>SimpleBaseline</td>
<td>ResNet-152</td>
<td>X</td>
<td>96.6</td>
<td>95.0</td>
<td>89.0</td>
<td>83.4</td>
<td>84.4</td>
<td>80.3</td>
<td>88.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimpleBaseline</td>
<td>ResNet-152</td>
<td>✓</td>
<td>97.0</td>
<td>96.3</td>
<td>91.3</td>
<td>87.3</td>
<td>89.6</td>
<td>84.3</td>
<td>91.0</td>
<td>2.3</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-152</td>
<td>X</td>
<td>97.2</td>
<td>96.3</td>
<td>91.1</td>
<td>85.9</td>
<td>89.7</td>
<td>87.1</td>
<td>82.9</td>
<td>90.5</td>
<td>↑1.8</td>
</tr>
<tr>
<td>HRNet</td>
<td>HRNet-W32</td>
<td>X</td>
<td>97.0</td>
<td>95.7</td>
<td>89.4</td>
<td>85.6</td>
<td>87.7</td>
<td>85.8</td>
<td>82.0</td>
<td>89.5</td>
<td></td>
</tr>
<tr>
<td>HRNet</td>
<td>HRNet-W32</td>
<td>✓</td>
<td>97.4</td>
<td>96.7</td>
<td>92.1</td>
<td>88.4</td>
<td>90.8</td>
<td>88.6</td>
<td>85.0</td>
<td>91.7</td>
<td>2.2</td>
</tr>
<tr>
<td>Ours</td>
<td>HRNet-W32</td>
<td>X</td>
<td>97.4</td>
<td>96.6</td>
<td>91.8</td>
<td>87.5</td>
<td>89.6</td>
<td>87.6</td>
<td>83.8</td>
<td>91.1</td>
<td>↑1.6</td>
</tr>
</tbody>
</table>

Figure 11. Effect of the number of masked regions in Joint Cutout. The results show the randomly masking some joints in hard augmentation help to improve the performance and $m = 5$ achieves the best accuracy in this setting.

Figure 12. Effect of the distortion magnitudes in RandAugment.

C.2. MPII Validation Dataset

We test our method (Dual) in a more realistic setting where labeled and unlabeled images are from different datasets of MPII and AIC, respectively. Table 10 shows the results on the validation set of MPII. Our approach outperforms the supervised baseline [5] by a large margin on three different backbones. After applying our method to utilize unlabeled data, the error rate reduces by around 15%. It is worth noting that performing supervised learning on the combined MPII and AIC dataset with extra labels only gets slightly better accuracy than our semi-supervised approach which validates the effectiveness of our approach.

References


