# Joint COCO and LVIS workshop at ECCV 2020: COCO Keypoint Challenge Track Technical Report: UDP++

Junjie Huang[1*], Zengguang Shan[1*], Yuanhao Cai[1,3*], Feng Guo[1], Yun Ye[1], Xinze Chen[1], Zheng Zhu[2], Guan Huang[1], Jiwen Lu[2], and Dalong Du[1]

[1] XForwardAI Technology Co.,Ltd, Beijing, China
{junjie.huang, zengguang.shan, yuanhao.cai, feng.guo, yun.ye, xinze.chen, guan.huang, dalong.du}@xforwardai.com
[2] Tsinghua University, Beijing, China
zhengzhu@ieee.org, lujiwen@tsinghua.edu.cn
[3] Tsinghua Shenzhen International Graduate School, Shenzhen, China

**Abstract.** In this report, we present our human pose estimation system, UDP++. UDP++ takes deep HRNet as network backbone and is equipped with Unbiased Data Processing and Occlusion Augmentation supervision. On the COCO-2020 keypoint test-dev dataset, the UDP++ system achieves 80.8 AP, which surpasses the 2019 winning result by 1.6 AP. The source code will be publicly available for further research in https://github.com/HuangJunJie2017/UDP-Pose.

## 1 Introduction

Recent years have witnessed a rapid advance in human pose estimation especially in network structure. However, the overlook on other aspects like data processing and supervision makes the exiting work suffer from accuracy degradation. We firstly propose Unbiased Data Processing (UDP) [5] to wipe out the systemic error hidden in the data processing pipeline. And then Occlusion Augmentation (OA) is designed to force the agents to pay more attention on the constraint cue instead of overfitting the appearance cue. By combining UDP and OA, UDP++ boosts the performance of the baseline configuration HRNetW32-256x192 by 2.7 AP to 76.2 AP on COCO `test-dev` set. With some other methods, UDP++ finally scores 80.8 AP on COCO `test-dev` set and 77.4 AP on COCO `test-challenge` set.

## 2 Methodology

In this section, we introduce two main breakthroughs in UDP++. They solve the common problems which are widely presented in most human pose estimation systems.

*The first three authors contribute equally to this work.

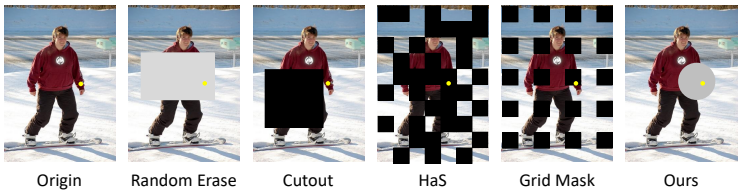


**Fig. 1.** The illustration of different information dropping methods.

### 2.1 Unbiased Data Processing

The data processing pipeline can be simply formulated as:

$$\hat{\mathbf{k}} = T'(D(E(T(\mathbf{k})))) \tag{1}$$

where $\mathbf{k}$ is the ground truth position of the keypoints. $T$ is the transformation between different coordinate systems. $T'$ is the inverse transformation of $T$. $E$ and $D$ denotes the common Encoding and Decoding operation. With an unbiased data processing pipeline, we should have:

$$\hat{\mathbf{k}} = \mathbf{k} \tag{2}$$

However, this equation does not hold in the most state-of-the-arts. We diagnose the systemic bias in the existing data processing and propose Unbiased Data Processing (UDP) to thoroughly solve this problem. UDP is effective, universal and efficient. More details can be found in [5].

### 2.2 Occlusion Augmentation

**Motivation** We rethink the relationship between the way that human locate the pose in images and the supervision of existing pose estimators. The base cue human used is the appearance of the keypoints. This inspires pioneers to use response exactly located at the keypoints as the supervision in the training process. This is intuitive and has been proved effective in most existing works. Besides, another cue is the constraints like the relationship in human pose or the interaction between human and environment. This cue helps human locate the keypoints under some challenging situations such as occlusion, ambiguity between left and right knees.

Although the powerful neural networks have potential to learn from data, the constraint cue is still too hard for neural network to completely grasp. By contrast, the appearance cue is intuitively easier for learning. When the appearance cue is always present and there's no penalty on the neglecting of constraint cue, the algorithms only with response map supervision tend to overfit the appearance cue. So customized supervision is needed for neural network to mine the important constraint cue from the training data.

To this end, we introduce a customized information dropping methods to explicitly force the neural network to focus on the constraint cue. Information dropping is a well known method for regularization and has been widely used in many other problems. By dropping information in images, the neural networks can learn discriminative features, resulting in a notable increase of model robustness. Inspired by this, we randomly drop the appearance information of a keypoint and maintain the response map supervision, which makes neural network pay more attention to the constraint cue. We call this `occlusion augmentation`.


**Fig. 2.** The illustration of the proposed keypoint-aware occlusion augmentation.

**Keypoint-aware Occlusion Augmentation** We firstly revising the existing information dropping methods. As illustrated in Figure 1, random erase [10] and cutout [4] drop a single continuous area centered at random position. By contrast, hide-and-seek [7] and GridMask [2] perform multi-area information dropping. Though all methods mentioned above have a certain probability to dropping appearance information of keypoints, the probability is too low to produce the best results. Thus keypoint-aware occlusion augmentation is proposed in this paper for dropping keypoints' appearance information.

As illustrated in Figure 2, given a set of $N$ annotated keypoints $\{\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, ...\mathbf{k}_N\}$, a specific keypoint $\mathbf{k}$ is randomly selected. Then the intensity in the neighbourhood area around $\mathbf{k}$ is set as zero. We simply set the shape of the neighbourhood area as cycle. The radius is randomly chosen within the range of $r \in [0.1 * w, 0.2 * w]$, where $w$ denotes the width of the network input. As the occlusion shape is simple, we need to prevent the neural network to overfit this cue, i.e. there is at least one keypoint located at the center of the occlusion area. To this end, we firstly shift the center of the occlusion area by a random vector $\delta$. Then we also dropping appearance information with random center in the image

plane. During the training processing, whether a occlusion is performed around a specific keypoint or a random center, the probabilities of them are both 50%.

**Training schedule** Optimisation schedule is of significance for training robust pose estimation networks with occlusion augmentation. Empirically, we found that directly applying occlusion augmentation in training process even degrades the performance of pose estimator. Thus, customized training schedule is required for eliminating the information loss caused by information dropping.

We can make it in two different ways. We can just simply double the training schedule, leaving enough time for the network to conquer the difficulty. And the other way is to split the training process into two schedules. In this way, the training process starts with a common schedule without occlusion augmentation as the previous works, followed by an extra refinement schedule as long as the first one with occlusion augmentation. The main advantage of second approach is that we can reuse the existing well training models to save computational resource. Empirically, the two training schedules mentioned above have the same effect and an algorithm can gain similar promotion from either of them.

## 3 Experiments

### 3.1 Implementation Details

We use the COCO [6] train17 dataset including 150K person instances and AIChallenge [9] including 380K person instances for training. Networks are implemented base on Mxnet[3] and trained on 2080Ti cluster servers. We evaluate our approach on the COCO `test-dev` set (20K images) and `test-challenge` set (20K images).

### 3.2 Ablation Study on Occlusion Augmentation

Table 1 reports the performance of the proposed occlusion augmentation on COCO `test-dev` set. The proposed occlusion augmentation boosts the performance of different configurations consistently by around 0.6 AP.

**Ablation study on Training Schedule** To explore the effect of training schedule, we firstly design three different training schedules:

S1. Normal training schedule from HRNet [8] with a base learning rate of 1e-3 and is dropped to 1e-4 and 1e-5 at the 170th and 200th epochs, repectively. The training process is terminated within 210 epochs.
S2. Double the length of the schedule S1. The learning rate is dropped at 380th and 410th epochs. The training process is terminated within 420 epochs.
S3. Repeat the Schedule S1 twice with different configuration.

| Method | Backbone | Input size | AP | $AP^{50}$ | $AP^{75}$ | $AP^{M}$ | $AP^{L}$ | AR |
|---|---|---|---|---|---|---|---|---|
| Simple+UDPv1[5] | ResNet-50 | 256 × 192 | 73.1 | 91.9 | 80.9 | 69.6 | 78.9 | 79.1 |
| **+OA** | ResNet-50 | 256 × 192 | **73.7(+0.6)** | **92.2** | **81.6** | **70.4** | **79.4** | **79.7** |
| RSN+UDPv1[5] | 2xRSN-50 | 256 × 192 | 76.0 | 92.5 | 83.6 | 73.2 | 81.4 | 82.3 |
| **+OA** | 2xRSN-50 | 256 × 192 | **76.6(+0.6)** | **92.8** | **84.3** | **73.6** | **82.1** | **82.5** |
| UDPv1[5] | HRNet-W32 | 256 × 192 | 75.6 | 92.3 | 83.2 | 72.3 | 81.4 | 81.4 |
| **+OA** | HRNet-W32 | 256 × 192 | **76.2(+0.6)** | **92.8** | **83.8** | **72.9** | **81.8** | **81.8** |
| UDPv1[5] | HRNet-W32 | 384 × 288 | 76.5 | 92.6 | 83.8 | 72.9 | 82.5 | 82.1 |
| **+OA** | HRNet-W32 | 384 × 288 | **77.0(+0.5)** | **92.9** | **84.5** | **73.6** | **82.8** | **82.5** |
| UDPv1[5] | HRNet-W48 | 256 × 192 | 76.1 | 92.4 | 83.7 | 72.7 | 81.9 | 81.8 |
| **+OA** | HRNet-W48 | 256 × 192 | **76.7(+0.6)** | **92.9** | **84.2** | **73.4** | **82.3** | **82.3** |
| UDPv1[5] | HRNet-W48plus | 384 × 288 | 76.8 | 92.8 | 84.1 | 73.4 | 82.6 | 82.4 |
| **+OA** | HRNet-W48plus | 384 × 288 | **77.5(+0.7)** | **92.9** | **84.9** | **74.2** | **83.4** | **83.0** |
| UDPv1*[5] | HRNet-W48plus | 384 × 288 | 78.2 | 92.8 | 85.4 | 74.8 | 84.2 | 83.6 |
| **+OA** | HRNet-W48plus | 384 × 288 | **78.7(+0.5)** | **93.1** | **85.9** | **75.2** | **84.5** | **84.0** |

**Table 1.** The improvement of AP on COCO `test-dev` set when the proposed occlusion augmentation is applied to the state-of-the-art methods. $*$ use extra data from AIchallenge[9].

| ID | schedule | occlusion augmentation |
|---|---|---|
| E1 | S1 | N |
| E2 | S1 | Y |
| E3 | S2 | N |
| E4 | S2 | Y |
| E5 | S3 | N/Y |

**Table 2.** Configurations of the contrast experiment for training schedule ablation study.

And experiments is constructed as list in Table 2. The training loss and the corresponding performance on COCO `val` AP metric are illustrated in Figure 3 and Figure 4, respectively. E1 and E2 use the same normal training schedule and the only variable is whether to use occlusion augmentation or not. With occlusion augmentation, the training loss is higher than that without occlusion augmentation. The performance of E2 is poor than E1 in the early training process, as the appearance information is the base of human pose estimation and information dropping disturbs the study of appearance feature. However, E1 and E2 have similar performance at the end of this training schedule. E3 and E4 adopt a longer schedule S2. The performance of E4 with occlusion augmentation starts surpassing E3 around 250 epochs. And this superiority gradually growth in the subsequent training process. Comparing E3 with E1 and E4 with E2, a longer schedule enables the algorithms learning more useful information when occlusion augmentation is used, but makes the algorithms overfit the training data when occlusion augmentation is absent. Comparing E4 with E5, Schedule2 and Schedule3 offer similar improvements.

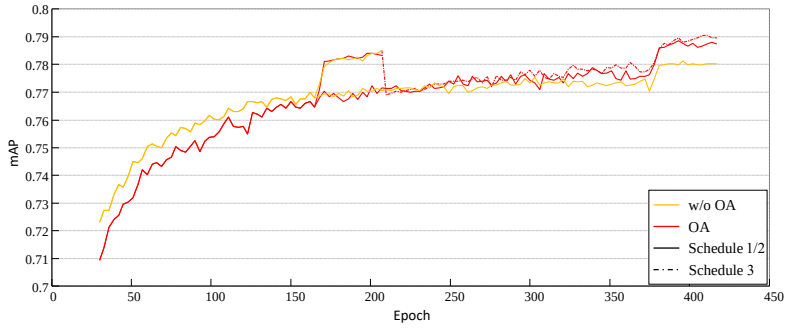


**Fig. 3.** The performance of different configurations.

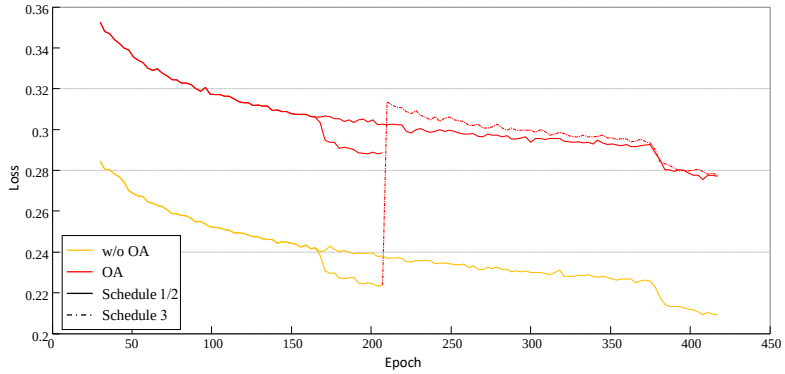


**Fig. 4.** The training loss of different configurations.

### 3.3 COCO 2020 Keypoint Road Map

The detailed road map for keypoint is listed in Table 3, where AP is reported on COCO `test-dev` dataset:

0. We take the original HRNet [8] as baseline with configuration of HRNetW32-256x192. The initial score is 73.5 AP.
1. UDP boosts the performance of the baseline by 2.0 AP to 75.5 AP.
2. Occlusion augmentation offers an extra improvement of 0.7 AP to 76.2 AP.
3. We further apply some other tricks (including other augmentations and hyper-parameters searching) to improve the baseline by 0.6 AP to 76.8 AP.
4. With larger input size 384x288, the small backbone scores 77.6 AP.
5. When replacing the small backbone with larger backbone HRNetW48plus, the result is boosted by 0.5 AP to 78.1 AP.
6. Extra data from AIChallenge [9] is used for training alone with COCO `train` set. This gives 1.3 AP improvement to 79.4 AP.
7. Refining the trained models with SyncBN offers an improvement of 0.2 AP to 79.6 AP.

| Methods | AP | Improvement |
|---|---|---|
| Baseline | 73.5 | - |
| +1.Unbiased Data Processing | 75.5 | +2.0 |
| +2.Occlusion Augmentation | 76.2 | +0.7 |
| +3.other tricks | 76.8 | +0.6 |
| +4.larger input size | 77.6 | +0.8 |
| +5.larger backbone | 78.1 | +0.5 |
| +6.extra data | 79.4 | +1.3 |
| +7.refine with SyncBN | 79.6 | +0.2 |
| +8.stronger human detection | 80.2 | +0.6 |
| +9.model ensemble | 80.8 | +0.6 |

**Table 3.** Keypoint road maps of COCO 2020 challenge.

8. The human detection bounding box result is of significance. We replace the open source HTC [1] result with ours which scores 59.8 AP on COCO Detection Challenge (Bounding Box) `test-dev` dataset. The score of `Person` class is 68.6 AP and this offers 0.6 improvement to 80.2 AP.
9. We ensemble 17 different models, and UDP++ finally scores 80.8 AP on COCO `test-dev` set.

## References


1. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., et al.: Hybrid task cascade for instance segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4974–4983 (2019)
2. Chen, P.: Gridmask data augmentation. arXiv preprint arXiv:2001.04086 (2020)
3. Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., Zhang, Z.: Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274 (2015)
4. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
5. Huang, J., Zhu, Z., Guo, F., Huang, G.: The devil is in the details: Delving into unbiased data processing for human pose estimation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
6. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)
7. Singh, K.K., Yu, H., Sarmasi, A., Pradeep, G., Lee, Y.J.: Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond. arXiv preprint arXiv:1811.02545 (2018)
8. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
9. Wu, J., Zheng, H., Zhao, B., Li, Y., Yan, B., Liang, R., Wang, W., Zhou, S., Lin, G., Fu, Y., et al.: Ai challenger: A large-scale dataset for going deeper in image understanding. arXiv preprint arXiv:1711.06475 (2017)
10. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI. pp. 13001–13008 (2020)