**Noname manuscript No.**
(will be inserted by the editor)

# Pixel-in-Pixel Net: Towards Efficient Facial Landmark Detection in the Wild

**Haibo Jin**[1] · **Shengcai Liao**[1] · **Ling Shao**[1,2]



Received: date / Accepted: date

**Abstract** Recently, heatmap regression models have become popular due to their superior performance in locating facial landmarks. However, three major problems still exist among these models: (1) they are computationally expensive; (2) they usually lack explicit constraints on global shapes; (3) domain gaps are commonly present. To address these problems, we propose Pixel-in-Pixel Net (PIPNet) for facial landmark detection. The proposed model is equipped with a novel detection head based on heatmap regression, which conducts score and offset predictions simultaneously on low-resolution feature maps. By doing so, repeated upsampling layers are no longer necessary, enabling the inference time to be largely reduced without sacrificing model accuracy. Besides, a simple but effective neighbor regression module is proposed to enforce local constraints by fusing predictions from neighboring landmarks, which enhances the robustness of the new detection head. To further improve the cross-domain generalization capability of PIPNet, we propose self-training with curriculum. This training strategy is able to mine more reliable pseudo-labels from unlabeled data across domains by starting with an easier task, then gradually increasing the difficulty to provide more precise labels. Extensive experiments demonstrate the superiority of PIPNet, which obtains state-of-the-art results on three out of six popular benchmarks under the supervised setting. The results on two cross-domain test sets are also consistently improved compared to the baselines. Notably, our lightweight version of PIPNet runs at 35.7 FPS and 200 FPS on CPU and GPU, respectively, while still maintaining a competitive accuracy to state-of-the-art methods. The code of PIPNet is available at https://github.com/jhb86253817/PIPNet.

**Keywords** Facial landmark detection · Pixel-in-pixel regression · Self-training with curriculum · Unsupervised domain adaptation

Shengcai Liao is the corresponding author

Haibo Jin
E-mail: haibo.nick.jin@gmail.com

Shengcai Liao
E-mail: scliao@ieee.org

Ling Shao
E-mail: ling.shao@ieee.org

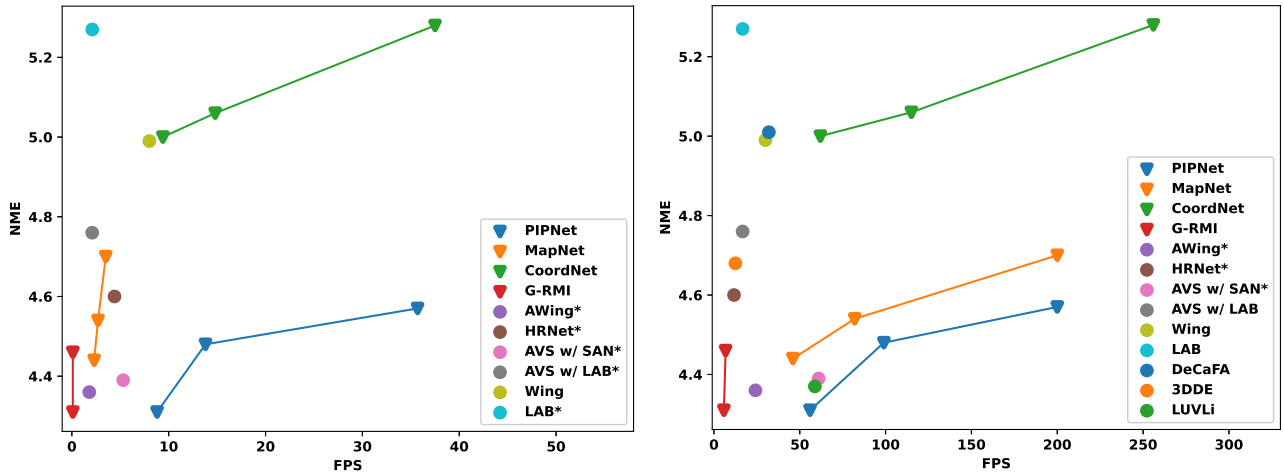
[1] Inception Institute of Artificial Intelligence (IIAI),
Abu Dhabi, UAE

[2] Mohamed bin Zayed University
of Artificial Intelligence (MBZUAI),
Abu Dhabi, UAE

## 1 Introduction

Facial landmark detection aims to locate predefined landmarks on a human face, the results of which are useful for several face analysis tasks, such as face recognition (Taigman et al. 2014; Liu et al. 2017b; Liao et al. 2013), face tracking (Khan et al. 2017), face editing (Thies et al. 2016), etc. These applications usually run on online systems in uncontrolled environments, requiring facial landmark detectors to be accurate, robust, and computationally efficient, all at the same time.

Over the last few years, significant progress has been made in this area, especially by deep convolutional neural networks (CNNs) that can be trained end-to-end. Among recent works, some (Feng et al. 2018; Wang et al. 2019b) aim at improving loss functions, some (Dong et al. 2018; Qian et al. 2019) focus on data augmentation for better generalization, and others (Wu et al. 2018; Liu et al. 2019) address the semantic ambiguity issue. However, few studies focusing on detection heads have been conducted, despite their es-




(a) CPU (b) GPU

**Fig. 1** Comparison with the existing methods in terms of speed-accuracy trade-off. The NMEs (%) are tested on the WFLW test set. The closer a model is to the bottom-right corner, the better its speed-accuracy trade-off. The existing methods with * were tested by us under the same environment as our methods. (a) Tested on CPU. (b) Tested on GPU.

sentialness to landmark detectors. Specifically, the detection head can affect the accuracy, robustness, and efficiency of a model. For deep learning based facial landmark detection, there are two widely used detection heads, namely heatmap regression and coordinate regression. Heatmap regression can achieve good results, but it has two drawbacks: (1) it is computationally expensive; (2) it is sensitive to outliers (see Figure 5(b)). In contrast, coordinate regression is fast and robust, but not accurate enough (see Figure 5(a)). Although coordinate regression can be used in a multi-stage manner to yield better performance, its inference speed becomes slow as a result. Accordingly, in this work, we aim to answer the following question: Is there a detection head that possesses the advantages of both heatmap regression and coordinate regression?

Generalization capability across domains is another challenge of facial landmark detection. As shown in (Valle et al. 2019), there are great performance gaps between intra-domain and cross-domain test sets. For a model to perform robustly under unconstrained environments, the domain gaps should be made as small as possible. Existing works (Wu et al. 2018; Zhu et al. 2019a; Qian et al. 2019) all address this problem by training a model with supervised learning, and then directly evaluating it on cross-domain datasets. We call this paradigm generalizable supervised learning (GSL). A drawback of GSL is that it relies on human-designed modules for cross-domain generalization, which are not scalable. One may suggest training the models on various datasets with supervised learning, but this is impractical due to the high labor costs of annotation. Therefore, we have decided to explore generalizable semi-supervised learning (GSSL) for facial landmark detection, which utilizes both labeled and unlabeled data across domains to obtain better generalization capability. Compared to GSL, GSSL is more scalable because it is data-driven, and unlabeled images are relatively easy to collect. Unsupervised domain adaptation (UDA), a special case of GSSL, has been successfully adopted in several vision tasks, including image classification (Long et al. 2015; Ganin and Lempitsky 2015; Kang et al. 2019), object detection (Chen et al. 2018; Zhu et al. 2019b; Saito et al. 2019), person re-identification (Peng et al. 2016; Yu et al. 2017; Zhong et al. 2018; Deng et al. 2018; Yu et al. 2019; Zhao et al. 2020), and so on. However, the effectiveness of UDA for facial landmark detection remains unknown. Figure 2 shows the difference between various training and testing paradigms. As shown in the figure, the main difference between GSSL and UDA is that GSSL is not very strict about the domain of the unlabeled data, while UDA usually requires the unlabeled and test data to be from the same domain. In this work, we investigate the feasibility of GSSL (including UDA) for better cross-domain generalization on facial landmark detection.

In order to obtain an efficient facial landmark detector that can run in the wild, we propose a new model named Pixel-In-Pixel Net (PIPNet). PIPNet consists of three essential parts: (1) Pixel-In-Pixel (PIP) regression; (2) a neighbor regression module; and (3) self-training with curriculum. PIP regression, the detection head of PIPNet, is based on heatmap regression, but further predicts offsets within each feature map pixel (grid) in addition to predicting scores. By doing so, the model can still achieve good results even when the stride of the network is large (i.e., the last feature map is of low resolution). Consequently, the upsampling layers for heatmap regression can be eliminated to save considerable computational cost, without sacrificing accuracy. The

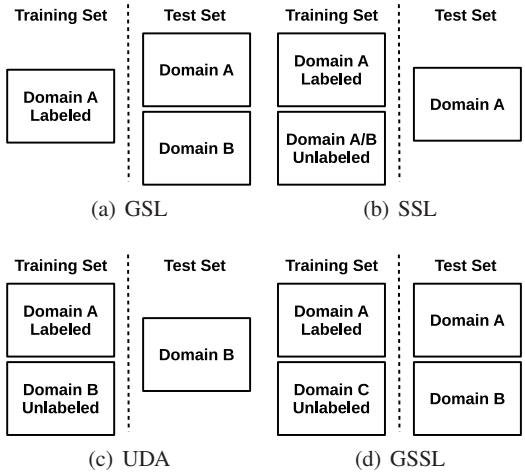



**Fig. 2** Different training and testing paradigms. (a) Generalizable supervised learning. (b) Semi-supervised learning. (c) Unsupervised domain adaptation. (d) Generalizable semi-supervised learning.

neighbor regression module is designed to enhance the robustness of the PIP regression, inspired by coordinate regression (see Section 3.2). For each landmark, the neighbor regression module predicts the locations of the neighboring landmarks within each feature map pixel. The predicted neighbors are then merged with the results of PIP regression during inference. With marginal extra cost, the neighbor regression module is able to improve the robustness of the model by introducing local constraints on the shapes of the predicted landmarks. With the help of PIP regression and the neighbor regression module, the proposed model inherits the advantages of both heatmap and coordinate regression. In fact, in Section 3.1, we show that heatmap and coordinate regression can be seen as two special cases of PIP regression with different strides. We also demonstrate the superiority of PIP regression (with neighbor regression) over the two alternatives in terms of bias-variance trade-off in Section 4.3.2. In order to better utilize unlabeled data across domains, we propose self-training with curriculum for generalizable semi-supervised learning. Different from standard self-training, self-training with curriculum starts with an easier task for the unlabeled data, and then gradually increases the difficulty to obtain more refined pseudo-labels. In this way, less errors are introduced from the estimated pseudo-labels, easing the mistake reinforcement problem of self-training.

Our contributions in this work are summarized as follows.

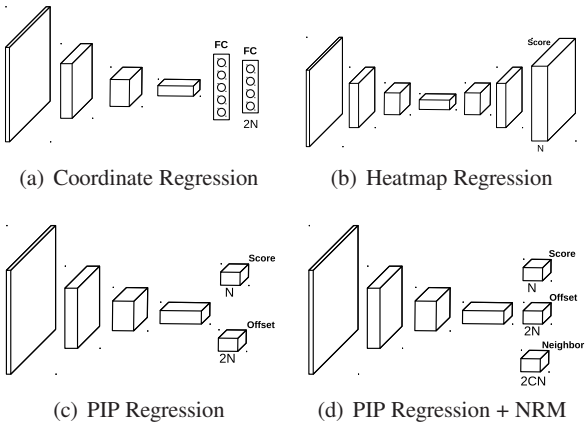
1. We propose PIP regression as a novel detection head for facial landmark detection, which achieves comparable accuracy to heatmap regression, but runs significantly faster on CPU (see Figure 1(a)). We also show that PIP regression is a generalization of the two popular detection heads. To the best of our knowledge, this is the first study in this area that discusses the connection between heatmap and coordinate regression.
2. A neighbor regression module is proposed to enhance the robustness of the PIP regression, especially on cross-domain datasets. Additionally, we show that PIP regression with the neighbor regression module yields better performance than the two alternatives from the perspective of bias-variance trade-off.
3. Aiming to further improve the generalization capability of PIPNet on unseen domains, a new method is designed under the GSSL paradigm, termed self-training with curriculum. Experiments show that self-training with curriculum achieves consistently better results than its baselines on two cross-domain datasets. As far as we know, this is the first study to utilize unlabeled data to improve generalization capability on facial landmark detection.
4. We observe that CNN-based landmark detectors make predictions using not only semantic features, but also positional features. Specifically, even if the input image does not contain a human face, the landmark detectors still give face-like predictions (see Figure 9(b)-9(d)), which can be seen as an implicit prior learned from data. Coordinate regression has a stronger implicit prior than heatmap regression, which also explains the different characteristics of the two detection heads.
5. The proposed PIPNet obtains state-of-the-art results on COFW, WFLW, and 300VW. Notably, PIPNet with ResNet-18 is able to run at 35.7 FPS and 200 FPS on CPU and GPU, respectively (see Figure1), while its accuracy is still competitive with state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 briefly reviews the related works. Section 3 introduces the proposed methods. The experimental results are presented in Section 4. Finally, we draw conclusions in Section 5.

## 2 Related Work

In this section, we review relevant works on deeply supervised facial landmark detection (coordinate regression models and heatmap regression models), semi-supervised facial landmark detection, and the generalization capability across domains in this area.

**Coordinate Regression Models.** Coordinate regression can be used to directly map an input image to landmark coordinates. In the context of deep learning, the features of the input image are usually extracted using a CNN, then mapped to coordinates through fully connected layers. Due to its fixed connections to specific locations of feature maps, the end-to-end prediction of coordinate regression is inaccurate and biased. Therefore, coordinate regression is usually cascaded (Sun et al. 2013; Zhu et al. 2015; Trigeorgis et al. 2016; Lv et al. 2017; Feng et al. 2018), integrated with extra




(a) Coordinate Regression (b) Heatmap Regression

(c) PIP Regression (d) PIP Regression + NRM

**Fig. 3** Architectures of various detection heads. (a) Coordinate regression. (b) Heatmap regression. (c) PIP regression. (d) PIP regression + NRM.

modules (Wu et al. 2018; Zhu et al. 2019a), or built upon heatmap regression (Valle et al. 2018; Liu et al. 2019).

**Heatmap Regression Models.** Heatmap regression maps an image to high-resolution heatmaps, each of which represents the probability of a landmark location. During inference, the location with the highest response on each heatmap is used. There are several ways to obtain high-resolution heatmaps. Stacked hourglass networks (Newell et al. 2016; Yang et al. 2017; Liu et al. 2019; Chen et al. 2019; Dong and Yang 2019; Zou et al. 2019; Wang et al. 2019b; Chandran et al. 2020) have been shown to perform well on landmark prediction through repeated downsampling and upsampling modules. U-Net (Ronneberger et al. 2015), originally developed for biomedical image segmentation, has also been successfully applied to facial landmark detection (Tang et al. 2018; Zou et al. 2019; Dapogny et al. 2019; Kumar et al. 2020). The convolutional pose machine (CPM) (Wei et al. 2016; Dong et al. 2018; Dong and Yang 2019) is a sequential architecture composed of CNNs, where the predictions are increasingly refined at each stage. Robinson et al. (2019) use consecutive bilinear upsampling layers to recover high-resolution heatmaps. Merget et al. (2018) maintain the input resolution through the whole network by not using any downsampling operations. Xiao et al. (2018) proposed a simple but effective architecture to obtain high-resolution heatmaps through several deconvolutional layers. High-Resolution Net (HRNet) (Wang et al. 2019a) maintains multi-resolution representations in parallel and exchanges information between these streams to obtain a final representation with great semantics and precise locations. However, existing works all require high-resolution heatmaps for heatmap regression, while PIP regression uses low-resolution heatmaps for reduced computational cost.

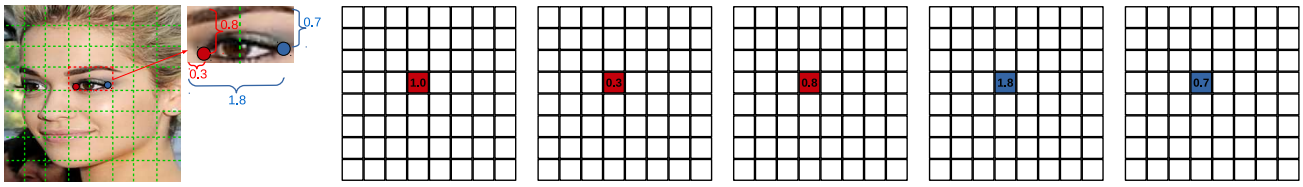
The model most related to ours is from (Papandreou et al. 2017, 2018). This model predicts disk-shaped heatmaps as well as 2D offsets within the disk area, and the two predictions are then aggregated through Hough voting for person keypoint detection. Although this model and our proposed method can both be seen as hybrids of classifition and regression, there are considerable differences between the two. Firstly, our model is based on low-resolution feature maps, while the prior model uses high-resolution maps. Secondly, PIP regression (without NRM) itself is a hybrid of heatmap and coordinate regression, and also a general case of the two, while the prior model is essentially a heatmap regression model, whose offset prediction is an extra module for better accuracy. Finally, the neighbor regression module in this work aims to improve the consistency of predicted landmarks, while the short-range and mid-range offsets in (Papandreou et al. 2018) are mainly for accuracy improvement and keypoints grouping, respectively.

**Cross-Domain Generalization.** Wu et al. (2018) introduced additional boundary information to help improve the robustness on unseen faces. Zhu et al. (2019a) designed a geometry-aware module to address the occlusion problem. Qian et al. (2019) proposed to augment the training data style with a conditional variational auto-encoder to enable models to generalize better on unseen domains. These methods all focus on improving cross-domain generalization through supervised learning, a paradigm which we call generalizable supervised learning. In contrast, we propose to address the cross-domain generalization issue for landmark detection through generalizable semi-supervised learning, enabling us to utilize massive amounts of unlabeled data.

**Semi-Supervised Facial Landmark Detection.** Honari et al. (2018) proposed a module that can leverage unlabeled images by maintaining the consistency of predictions with respect to different image transformations. Robinson et al. (2019) designed an adversarial training framework to leverage unlabeled data. Dong and Yang (2019) applied an interaction mechanism between a teacher and students in a self-training framework, where the teacher learns to estimate the quality of the pseudo-labels generated by the students. The key difference between the above methods and ours is that their labeled and test data are from the same domain, while we test on domains that do not contain any labeled (i.e., UDA) or even unlabeled data (i.e., GSSL). In other words, the focus of prior works is to improve the performance on the source domain with less labeled data, while we aim to obtain better generalization capability across domains, which is a more challenging task.

## 3 Our Method

In this section, we first introduce PIP regression (Section 3.1), and then present the proposed neighbor regression module (Section 3.2). We describe the self-training with curriculum framework in Section 3.3. Finally, we present the




(a) Input Image (b) Score Map (c) $x$-Offset Map (d) $y$-Offset Map (e) $x$-Neighbor Map (f) $y$-Neighbor Map

**Fig. 4** Mapping from a ground-truth landmark to heatmap labels for PIPNet. (a) A sample image as input. The red dot denotes the target ground-truth landmark, and the blue one is a neighboring landmark. (b) Label assignment for the score map. (c)-(d) Label assignment for the offset maps on $x$ and $y$ axes, respectively. (e)-(f) Label assignment for the neighbor maps on $x$ and $y$ axes, respectively.

implicit prior we observe from CNN-based facial landmark detectors in Section 3.4.

### 3.1 PIP Regression

Existing facial landmark detectors can be categorized into two classes, defined according to the type of detection head: coordinate regression and heatmap regression. As can be seen from Figure 3(a), coordinate regression outputs a vector with length $2N$ from fully connected layers, where $N$ represents the number of landmarks. Heatmap regression (see Figure 3(b)), on the other hand, first gradually upsamples the extracted feature maps to the same (or similar) resolution as the input, and then outputs a heatmap with $N$ channels, each of which reflects the likelihood of the corresponding landmark location. When comparing the two detection heads, it is easy to see that coordinate regression is more computationally efficient at locating a point because heatmap regression needs to either upsample the feature maps repeatedly or maintain high-resolution feature maps throughout the network. However, heatmap regression has been shown to consistently outperform coordinate regression in terms of detection accuracy. Despite its inefficiency, heatmap regression is able to achieve state-of-the-art accuracy with a single-stage architecture, while coordinate regression usually needs two or more stages. As such, we pose the following question: Is it possible to obtain a detection head that is both efficient and accurate at the same time?

We propose a novel detection head, termed PIP regression, which is built upon heatmap regression. We argue that upsampling layers are not necessary for locating points on feature maps. That is to say, low-resolution feature maps are sufficient for localization. By applying heatmap regression to low-resolution feature maps, we obtain the most likely grid on the heatmap for each landmark. To get more precise predictions, we also apply offset prediction within each heatmap grid on the $x$-axis and $y$-axis, relative to the top-left corner of the grid. It is worth noting that PIP regression is a single-stage method because the score and offset predictions are independent to each other, and can thus be computed in parallel. Figure 3(c) gives the architecture of PIP regression, where the outputs are a score map ($N \times H_M \times W_M$) and an offset map ($2N \times H_M \times W_M$). The proposed detection head can be simply implemented by a $1 \times 1$ convolutional layer.

Figure 4 demonstrates how to convert a ground-truth landmark to heatmap labels for PIPNet. Suppose Figure 4(a) is an input image of size $256 \times 256$, the red dot on the right inner-eye-corner is the ground-truth landmark, and the network stride is 32. Then, the last feature map is of size $8 \times 8$. As can be seen from the figure, there are 64 grids on the last feature map for each channel, and we denote the grid that the ground-truth falls into as the positive grid. For the score map (see Figure 4(b)), the positive grid is assigned 1, and the rest are 0. Because the ground-truth landmark has a 30% offset on the $x$-axis relative to the top-left corner of the positive grid, the positive grid on the $x$-offset map is assigned 0.3 (see Figure 4(c)). Similarly, the same grid on the $y$-offset map is assigned 0.8 (see Figure 4(d)), and the rest are 0. The training loss for PIP regression can be formulated as follows:

$$L = L_S + \alpha L_O, \tag{1}$$

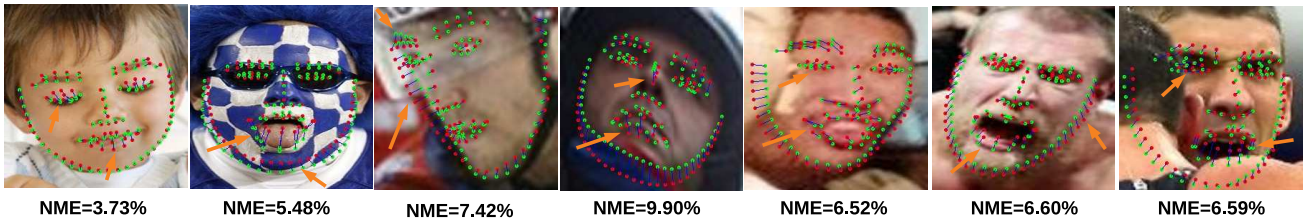
where $L_S$ is the loss for score prediction, $L_O$ is for offset prediction, and $\alpha$ is a balancing coefficient. Concretely, $L_S$ for a score map ($N \times H_M \times W_M$) is formulated as

$$L_S = \frac{1}{NH_MW_M} \sum_{i=1}^{N} \sum_{j=1}^{H_M} \sum_{k=1}^{W_M} (s^*_{ijk} - s'_{ijk})^2, \quad s^*_{ijk} \in \{0,1\}, \tag{2}$$

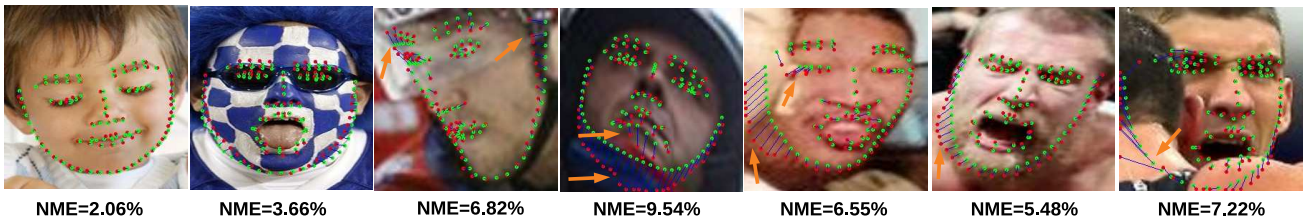
where $s^*_{ijk}$ and $s'_{ijk}$ denote the ground-truth and predicted score values, respectively, and $NH_MW_M$ is the normalization term. $L_O$ for an offset map ($2N \times H_M \times W_M$) is formulated as

$$L_O = \frac{1}{2N} \sum_{s^*_{ijk}=1} \sum_{l=1}^{2} |o^*_{ijkl} - o'_{ijkl}|, \quad o^*_{ijkl} \in [0,1], \tag{3}$$

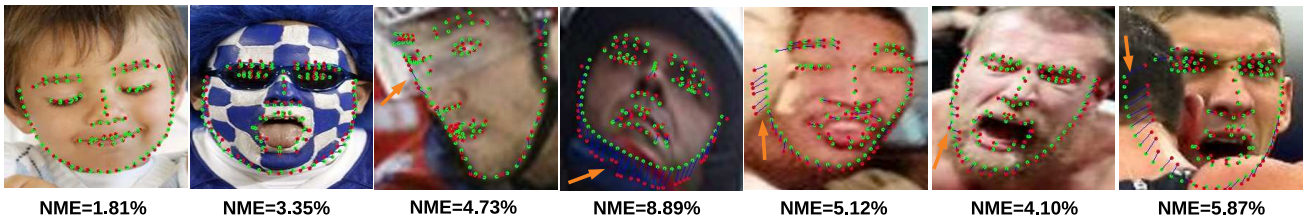
where $o^*_{ijkl}$ and $o'_{ijkl}$ denote the ground-truth and predicted offset values, respectively, and $2N$ is the normalization term. As can be seen from Equation 2 and 3, $L_S$ is applied to all the samples of a score map, while $L_O$ is applied to only positive samples of an offset map. Moreover, we use different loss fuctions for $L_S$ and $L_O$ because the former is actually a classification problem, while the latter is a regression problem. According to (Feng et al. 2018), the L1 loss yields better



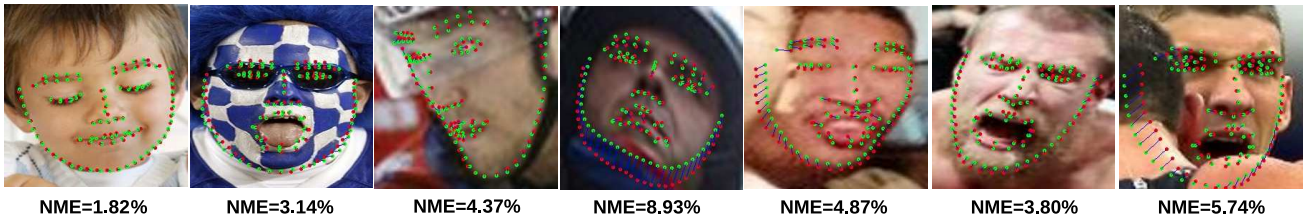

(a) Coordinate Regression (Avg. NME=**6.61**)




(b) Heatmap Regression, Stride=1 (Avg. NME=**5.90**)




(c) PIP Regression, Stride=32 (Avg. NME=**4.84**)




(d) PIP Regression + NRM, Stride=32 (Avg. NME=**4.67**)

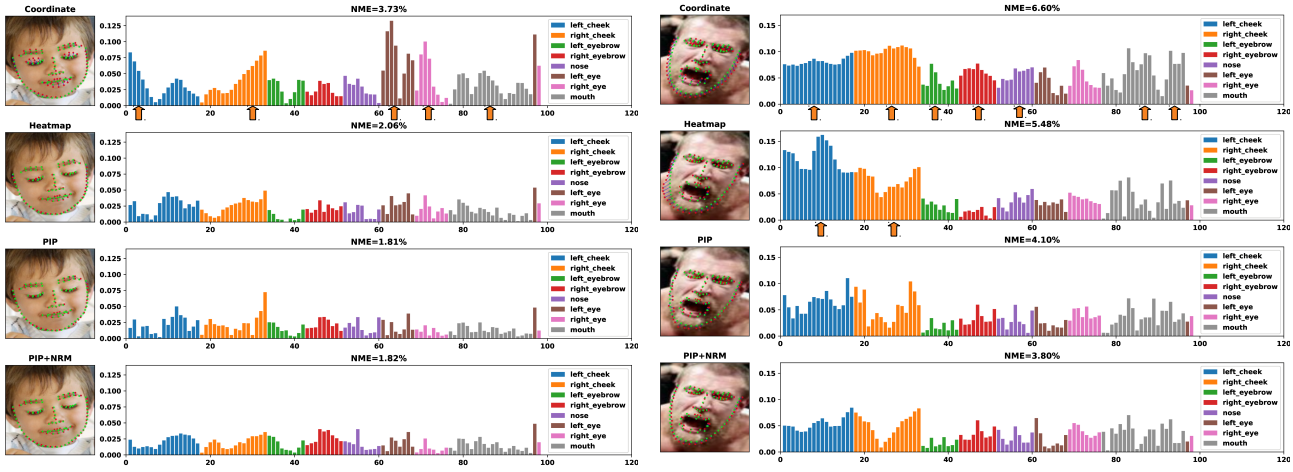
**Fig. 5** Visualization of predicted results on sample images from WFLW test set, with different detection heads. Green dots are ground-truths, red ones are predictions, and orange arrows show the areas with bad predictions. (a) Predictions of coordinate regression. (b) Predictions of heatmap regression, with stride 1. (c) Predictions of PIP regression, with stride 32. (d) Predictions of PIP regression + NRM, with stride 32.

results for regression, which is consistent with our experimental results. On the other hand, our experiments indicate that the L2 loss is better for classification. Therefore, we use the L2 loss for $L_S$ and the L1 loss for $L_O$. During inference, the final prediction of a landmark is computed as the grid location with the highest response refined by its corresponding offsets.

One hyperparameter of PIP regression is the stride of the network. Given the image size and network stride, the size of the heatmap can be determined as follows.

$$H_M = \frac{H_I}{S}, \quad W_M = \frac{W_I}{S}, \tag{4}$$

where $H_I$ and $W_I$ are the height and width of the input image, and $S$ denotes the network stride. Intuitively, PIP regression can be seen as a generalization of the two existing detection heads. When the network stride is equal to the image size (i.e., $H_M = W_M = 1$), and the score prediction module is removed, PIP regression can be seen as coordinate regression, where the conventional fully connected layers are replaced by convolutional layers. When the network stride is equal or close to 1, and the offset prediction is removed, then PIP regression is equivalent to heatmap regression (though there are still differences in implementation details, such as label smoothing and landmark inference). Consequently, PIP regression can be seen as conducting heatmap regression globally and coordinate regression locally at the same time, which is the reason it is called 'pixel-in-pixel'. Such a prop-




(a) (b)

**Fig. 6** Bar charts of normalized error on each landmark. The landmark IDs follow the original annotation, and are grouped into eight categories according to their regions for easy observation. Orange arrows show the areas with relatively large errors. (a) Images from 1st column of Figure 5. (b) Images from 6th column of Figure 5.

erty endows PIP regression with better flexibility and greater potential than the two alternatives.

### 3.2 Neighbor Regression Module

Although the proposed PIP regression addresses the computational efficiency issue of heatmap regression, it still suffers from poor robustness. Figure 5(a)-5(c) show some sample images with predictions from coordinate regression, heatmap regression, and PIP regression, respectively. As can be seen from Figure 5(a), coordinate regression outputs predictions with reasonable global shapes, even on large poses (e.g., 4th, 5th, and 7th images). However, it is not accurate in details, as we can observe obvious shifts between predictions and ground-truths in some areas (e.g., *eye* and *mouth* areas of the 1st image, *mouth* of the 2nd image, *left cheek* of the 3rd image, etc.). Consequently, coordinate regression may not be able to detect subtle changes such as eye blinking and mouth opening, which are essential functions of anti-spoofing. In contrast, as shown in Figure 5(b), heatmap regression is precise in details in general, but can give inconsistent shapes for images with extreme poses (e.g., 4th to 7th images). Similarly, PIP regression can also lack robustness under extreme poses (see 4th to 7th images of Figure 5(c)), despite obtaining better normalized mean error (NME) than heatmap regression. It is not difficult to understand the lack of robustness in heatmap-based models, because their predictions are based on different features (i.e., different locations on the feature map) and are thus independent to each other. In contrast, all the landmarks predicted by coordinate regression share the same feature, which we believe is the key to robustness.

Inspired by the above findings, we further propose a neighbor regression module (NRM) to help PIP regression predict more consistent landmarks. Specifically, in addition to the offsets of the landmark itself, each landmark also predicts the offsets of its $C$ neighbors. As shown in Figure 3(d), NRM further outputs a neighbor map of size $2CN \times H_M \times W_M$, where $C$ is the number of neighbors to predict. Concretely, mean shapes of the face landmarks are computed using the ground-truths in the training data, and the $C$ closest landmarks of the target landmark (excluding itself) are defined as its neighbors. In this work, we simply use Euclidean distance as the distance metric. We also explored correlation for defining neighbors, which gives a similar performance to Euclidean distance. Figure 4(e)-4(f) describe the label assignment for the neighbor maps of the ground-truth landmark (i.e., the red dot). Here, we use only one neighbor for illustration, but there can be more than one in practice. Assume that the blue dot on the right outer-eye-corner is a neighbor of the red dot in Figure 4(a). As can be seen from the figure, the blue dot has 180% and 70% offset on $x$ and $y$ axes respectively, relative to the top-left corner of the positive grid of the red dot. Thus, we assign 1.8 and 0.7 to the positive grids on $x$- and $y$-neighbor maps, respectively, and the remaining grids are 0. Note that the score, offset, and neighbor maps all belong to the red dot, while the blue dot has its own maps, which are not shown here.

After adding NRM, the training loss of PIPNet becomes

$$L = L_S + \alpha L_O + \beta L_N, \tag{5}$$

where $L_N$ is the loss for NRM, and $\beta$ is another balancing coefficient. We define $L_N$ as follows.

$$L_N = \frac{1}{2CN} \sum_{s^*_{ijk}=1} \sum_{l=1}^{2} \sum_{m=1}^{C} |n^*_{ijklm} - n'_{ijklm}|, \quad n^*_{ijklm} \in [0,1], \tag{6}$$

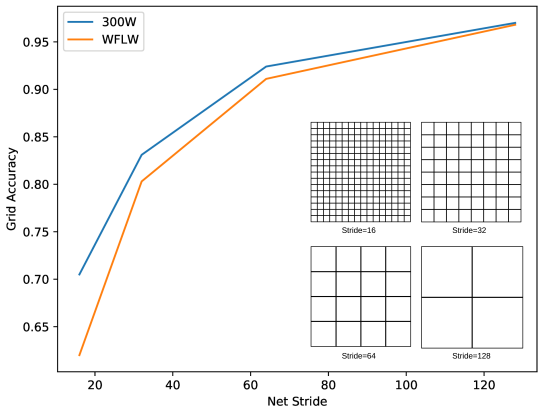



**Fig. 7** Classification accuracy on grids vs. network stride, tested on 300W and WFLW. The bottom-right of the figure gives the visualized heatmaps with different strides to better understand the difference in classification difficulty.

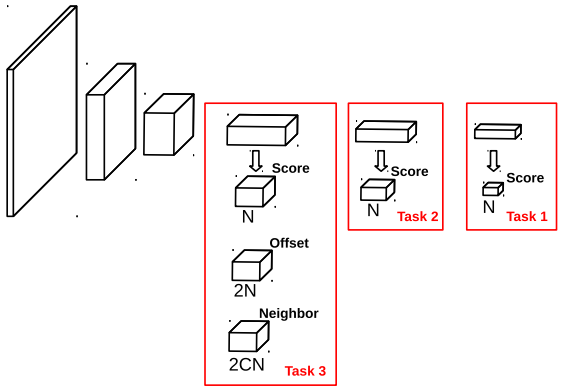



**Fig. 8** Architecture of PIPNet with the STC strategy. Two more higher-stride heatmap regression layers are added after PIP regression layer.

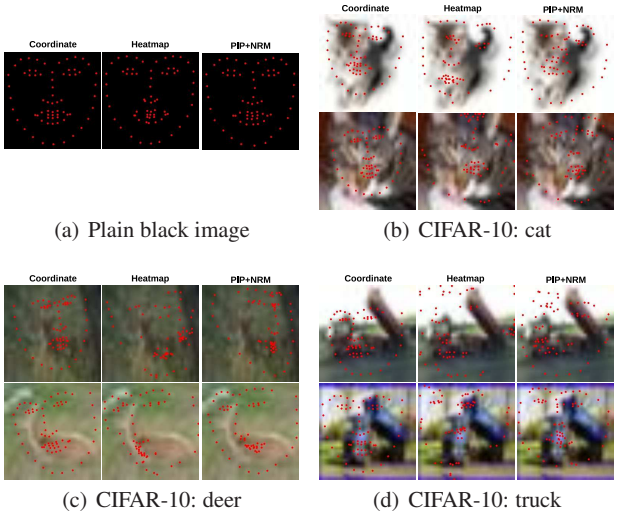
where $n^*_{ijklm}$ and $n'_{ijklm}$ denote the ground-truth and predicted neighbor offset values, respectively, and $2CN$ is the normalization term. Like $L_O$, $L_N$ also uses the L1 loss because it is a regression problem. During inference, each landmark collects its locations predicted by other landmarks as well as its own prediction, and then calculates the average of these as its final prediction.

With the help of NRM, PIP regression becomes more robust in addition to being accurate, as evidenced by Figure 5(d). Since the small errors are not easy to perceive through images, we further transform the errors to bar charts for a clearer illustration. The bar charts of two clolumns (i.e., 1st and 6th) from Figure 5 are presented in Figure 6(a) and 6(b) respectively, where each bar represents the normalized error of a landmark and the landmark IDs follow the original annotation (see supplementary material for landmark IDs and the other columns). To make it easy for observation, the bars are grouped into eight categories according to their regions. Although the image in Figure 6(a) is relatively simple, the predictions of coordinate regression have obvious shifts at *left and right cheek*, *left and right eye*, and *mouth* areas, which is the reason it obtains the worst NME=3.73%. In contrast, heatmap-based methods all achieve small NMEs thanks to their advantage on local accuracy. For the image in Figure 6(b), it is more difficult due to the expression and blurring issue. From the bar charts in Figure 6(b), we see that coordinate regression has large NME due to its accumulated local errors at multiple regions such as *left and right eyebrow*, *nose*, and *mouth*, despite its satisfactory global shape. Heatmap regression, on the contrary, obtains a large NME because of the inconsistent predictions at *left cheek* area. Being a heatmap-based method, PIP regression also predicts inconsistent landmarks on *left cheek* due to blurring and unclear boundaries. By adding NRM, PIP regression improves on both qualitative (consistency of predictions) and quantitative (NME value) results. Please refer to Section 4.2.2 and 4.3.2 for more quantitative results on NRM.

### 3.3 Self-Training with Curriculum

Although the neighbor regression module alleviates the unstableness of PIP regression, it is still not adequate for cross-domain datasets. Valle et al. (2019) pointed out the existence of domain gaps in facial landmark detection, and our experiments in Section 4.3.2 also confirm this problem. Therefore, we would like to further utilize unlabeled data across domains to improve the cross-domain generalization capability of our model. To this end, we propose self-training with curriculum (STC), which is built upon self-training. The key difference between traditional self-training and our method is that in the former the task is fixed, while in our method the difficulty of the task gradually increases, mimicking how humans learn. Different from original curriculum learning (Bengio et al. 2009), which presents training examples progressively for a specific task, our strategy applies curriculum learning at a task level. Such a design is based on the observation that grid classification on heatmaps becomes easier when the stride of the network becomes larger. This is easy to understand because a lower-resolution heatmap has less negative grids. Figure 7 shows the change in classification accuracy on grids when the network stride varies, tested on two datasets. It is clear that the classification accuracy increases consistently as the network stride becomes larger. We also observe that the advantage is more obvious for large network strides on harder datasets (WFLW contains more in-the-wild images than 300W), which indicates that the strategy will be more effective on harder unlabeled datasets. In the bottom-right corner of the figure, we give the heatmaps under different strides to better understand the differences in difficulty of classification.

Thanks to the flexibility of PIP regression, a PIPNet for supervised learning can easily be converted to a model




(a) Plain black image (b) CIFAR-10: cat

(c) CIFAR-10: deer (d) CIFAR-10: truck

**Fig. 9** Predictions from CNN-based landmark detectors, tested with three detection heads: coordinate regression, heatmap regression, and PIP regression + NRM. (a) Trained on plain black images but normal ground-truths from 300W, tested on plain black images. (b)-(d) Trained on 300W normally, tested on images from CIFAR-10.

with STC by simply adding higher-stride heatmap regression layers on top of the PIP regression. Figure 8 gives the architecture of PIPNet with the STC strategy. As can be seen, two more heatmap regression layers are added to a standard PIPNet. Assume that the input image is of size $256 \times 256$, and the standard PIP regression is of stride 32. Then, the heatmap size of the PIP regression layer is $8 \times 8$, and the sizes of the added ones are $4 \times 4$ and $2 \times 2$. In conventional self-training, the model iteratively learns from pseudo-labeled images on a fixed task (in our case, *Task 3* in the figure) until it converges. In contrast, in the proposed STC, the sequence of the tasks is arranged as *Task 1* $\rightarrow$ *Task 2* $\rightarrow$ *Task 3* $\circlearrowleft$, where the difficulty gradually increases until *Task 3*. By doing so, less errors from pseudo-labels are introduced and learned by the model so that the mistake reinforcement problem of self-training can be eased.

The pipeline of self-training with curriculum can be simply described as follows: (1) The modified PIPNet is trained with manually labeled data in a standard way (i.e., *Task 3*); (2) Pseudo-labels of the unlabeled data are estimated using the trained detector; (3) A new training set is formed with the manually labeled and pseudo-labeled data. (4) The modified PIPNet is trained on the new training set, using the manually labeled data to train the model through *Task 3*, and the pseudo-labeled data through *Task X*. Steps (2) to (4) are repeated until the model converges, and *Task X* is selected sequentially from the sequence *Task 1* $\rightarrow$ *Task 2* $\rightarrow$ *Task 3* $\circlearrowleft$. Empirically, we find that the model converges after three iterations of *Task 3*, which is used in all the relevant experiments. During inference, the model is used in the same way as the standard PIPNet, and the added heatmap regression layers are simply ignored or discarded.

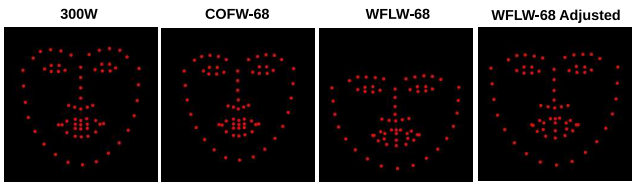



**Fig. 10** Mean faces of 300W, COFW-68, WFLW-68, and WFLW-68 after bounding box adjustment.

### 3.4 Implicit Prior

As pointed out by Islam et al. (2020), a CNN is able to encode position information through zero paddings. In other words, the neurons of a CNN know which part of an image they are looking at. To verify this, we train facial landmark detectors using all plain black images, but the ground-truth landmarks remain unchanged. Then, we input a plain black image for testing. The predictions of three different detection heads are shown in Figure 9(a). As can be seen from the figure, the models memorize the most likely positions of the landmarks regradless of which detection head they use, which proves their ability to perceive absolute positions. Therefore, CNNs do learn what (semantic features) and where (absolute positions) jointly (Islam et al. 2020). Different from multi-person keypoint detection and general object detection, facial landmark detectors locate landmarks through a cropped face image, where the facial features are correlated to certain positions (despite the use of augmentation techniques, such as translation and rotation, during training). To validate this, we train models with normal face images but test on images without human faces. As shown in Figure 9(b)- 9(d), when the input images come from CIFAR-10, the models still give landmark predictions close to a human face even if there is no facial feature information. That is to say, position information also contributes to the response of heatmaps. Intuitively, this can be seen as a prior implicitly learned by CNNs from training data. We also observe that coordinate regression gives more consistent predictions than the other two detection heads when no face is shown, which indicates that coordinate regression has a stronger prior of landmark positions. This may explain why coordinate regression is more robust but biased.

Another thing the implicit prior tells us is that it is important to have consistent cropped face images. While this may not be a problem in practice because the faces are usually detected by the same face detector, the benchmark datasets for facial landmark detection provide bounding boxes in different styles (see Figure 10). When conducting cross-domain evaluation or domain adaptation, bound-

ing box styles should be consistent to avoid causing performance degradation. In our case, we reduce the top area of the bounding boxes provided in WFLW-68 by 20% for the cross-domain setting. Figure 10 presents the mean faces of 300W, COFW-68, and WFLW-68 (before and after adjustment). The 300W and COFW-68 datasets have similar cropping styles, but WFLW-68 is shifted significantly downward. After adjustment, the mean face of WFLW-68 is roughly aligned with those of 300W and COFW-68.

## 4 Experiments

We first introduce the experimental settings in Section 4.1, and then discuss the hyperparameters of PIPNet in Section 4.2. In Section 4.3, we analyze the characteristics of PIPNet by comparing it to the baselines. We present the performance of PIPNet under the supervised and cross-domain setting in Sections 4.4 and 4.5, respectively. Finally, we demonstrate the advantage of PIPNet in terms of inference speed (Section 4.6).

### 4.1 Experimental Settings

#### *4.1.1 Datasets*

**300W** (Sagonas et al. 2013) provides 68 landmarks for each face in images collected from LFPW, AFW, HELEN, XM2VTS, and IBUG. Following (Ren et al. 2016), the 3,148 training images come from the training sets of LFPW and HELEN, and the full set of AFW. The 689 test images are from the test set of LFPW and HELEN, and the full set of IBUG. The test images are further divided into two sets: the common set (554 images) and the challenging set (135 images). Note that the common set is from LFPW and HELEN, and the challenging set is from IBUG.

**COFW** (Burgos-Artizzu et al. 2013) contains 1,345 training images and 507 test images, with the face images having large variations and occlusions. Originally, 29 landmarks were provided for each face. Ghiasi and Fowlkes (2014) reannotated the test set with 68 landmarks, which we denote as COFW-68. We use the original annotations for the supervised setting and the 68-landmark version for the cross-domain setting.

**WFLW** (Wu et al. 2018) consists of 7,500 training images and 2,500 test images from WIDER Face (Yang et al. 2016), with each face having 98 annotated landmarks. The faces in WFLW introduce large variations in pose, expression, and occlusion. The test set is further divided into six subsets for a detailed evaluation. These include pose (326 images), expression (314 images), illumination (698 images), make-up (206 images), occlusion (736 images), and blur (773 images). The original annotations are used under the supervised setting. To make WFLW applicable for the cross-domain setting, we generate 68-landmark annotations for the test set by converting the original 98 landmarks, and we name the new annotated dataset WFLW-68. Please refer to the supplementary materials for more details on the conversion.

**AFLW** (Koestinger et al. 2011) contains 24,386 face images in total, 20,000 of which are training images, with the remaining 4,386 used for testing. Following (Zhu et al. 2016), we use 19 landmarks of AFLW for training and testing.

**Menpo 2D** (Deng et al. 2019) contains more extreme poses, and it consists of two landmark configurations: semi-frontal (68 landmarks) and profile (39 landmarks). The semi-frontal track contains 5,658 training images and 5,335 test images. For the profile track, there are 1,906 and 1,946 images for training and testing, respectively. In this work, we train and test the proposed model on the two tracks separately.

**300VW** (Shen et al. 2015) is a popular benchmark for video-based facial landmark detection, with the same landmark configuration as 300W (Sagonas et al. 2013). It contains 114 videos, among which 50 are for training and 64 are for testing. The test videos are further divided into three categories based on their level of difficulty: (1) well-lit conditions (31 videos); (2) unconstrained conditions (19 videos); (3) completely unconstrained conditions (14 videos). To validate the robustness of our model, we train and test on the training and test images, respectively, without using any temporal information. To avoid overfitting, we sample every 5th frame from each training video. Since no face bounding boxes are provided, we use RetinaFace (Deng et al. 2020) to detect bounding boxes.

**CelebA** (Liu et al. 2015) is a large-scale attributes dataset with 202,599 face images. In this work, the images are only used as the unlabeled data in Section 4.5.

#### *4.1.2 Implementation Details*

**Supervised Setting**. The face images are cropped according to the provided bounding boxes, then resized to $256 \times 256$. To preserve more context, the bounding boxes of the datasets with 68 landmarks (i.e., 300W, Menpo 2D, and 300VW) are enlarged by 10%, and the ones with 98 landmarks (i.e., WFLW) are enlarged by 20%. We use ResNet-18 pretrained on ImageNet as the backbone by default. We also use ResNet-50 and ResNet-101 in some experiments to obtain better results. MobileNets (Sandler et al. 2018; Howard et al. 2019) are also adopted as backbones since they were designed for better efficiency. Adam (Kingma and Ba 2015) is used as the optimizer. The total number of training epochs is 60. The initial learning rate is 0.0001, decayed by 10 at epoch 30 and 50. The batch size is 16. When the network

**Table 1** NME (%) results of PIPNets (w/o NRM) with different network strides on WFLW validation set.

| Method | Net Stride | Heatmap Size | NME (%) |
|---|---|---|---|
| PIPNet | 16 | $16 \times 16$ | 4.03 |
| PIPNet | 32 | $8 \times 8$ | **3.94** |
| PIPNet | 64 | $4 \times 4$ | 3.96 |
| PIPNet | 128 | $2 \times 2$ | 4.10 |

stride varies, the balancing coefficients $\alpha$ and $\beta$ also need to be adjusted accordingly so that the loss values of classification (i.e., $L_S$) and regression (i.e., $L_O$ and $L_N$) are comparable. Concretely, $\alpha$ and $\beta$ are both set to 0.02, 0.1, 0.125, and 0.25 for stride 16, 32, 64, and 128, respectively. The data augmentation includes translation ($\pm$30 pixels on the $x$-axis and $y$-axis, $p = 0.5$), occlusion (rectangle with maximum 100 pixels as length, $p = 0.5$), horizontal flipping ($p = 0.5$), rotation ($\pm$30 degrees, $p = 0.5$), and blurring (Gaussian blur with maximum 5 radius, $p = 0.3$), where $p$ is the probability of execution.

**Cross-Domain Setting**. This setting consists of three paradigms: generalizable supervised learning (GSL), unsupervised domain adaptation (UDA), and generalizable semi-supervised learning (GSSL). For all paradigms, the 300W training set is used as labeled data and the test sets of 300W, COFW-68, and WFLW-68 are used for evaluation. As can be seen in Figure 2, the differences between the three paradigms are mainly in the unlabeled data: (1) GSL conducts evaluation directly after supervised learning, without using any unlabeled data; (2) UDA utilizes the unlabeled data from target domains (in our case, the training sets of COFW-68 and WFLW-68 without labels); and (3) GSSL utilizes unlabeled data that appears in neither the source domain nor the target domain (in our case, it comes from CelebA). As discussed in Section 3.4, it is essential to have consistent cropping styles for the cross-domain setting. Specifically, we enlarge the bounding boxes of 300W, COFW-68, and WFLW-68 by 30%, 30%, and 20%, respectively, for consistency. The boxes of WFLW-68 are also adjusted, as stated in Section 3.4. The bounding boxes of CelebA are detected by RetinaFace (Deng et al. 2020), then enlarged by 20%. The other implementation details are the same as in the supervised setting.

#### *4.1.3 Evaluation Metrics*

To compare with previous works, we use normalized mean error (NME) to evaluate our models, where the normalization distance is inter-ocular for 300W, 300VW, COFW, COFW-68, WFLW, and WFLW-68. For AFLW, we use image size as the normalization distance, following (Wang et al. 2019a). To deal with the large poses and profile faces in Menpo 2D, (Deng et al. 2019) proposed to use the face diagonal as the normalization distance, which is also adopted in this work for Menpo 2D.

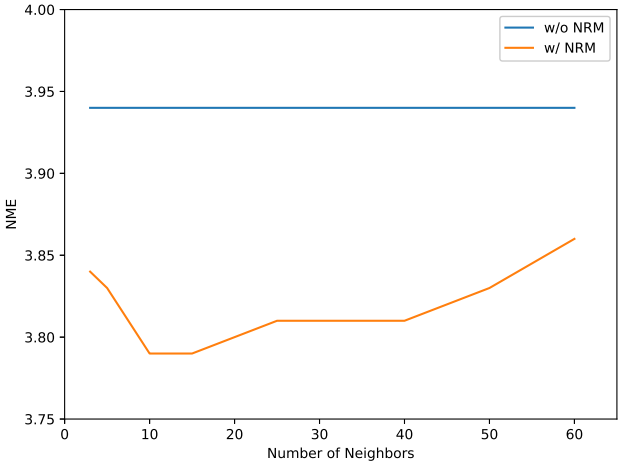


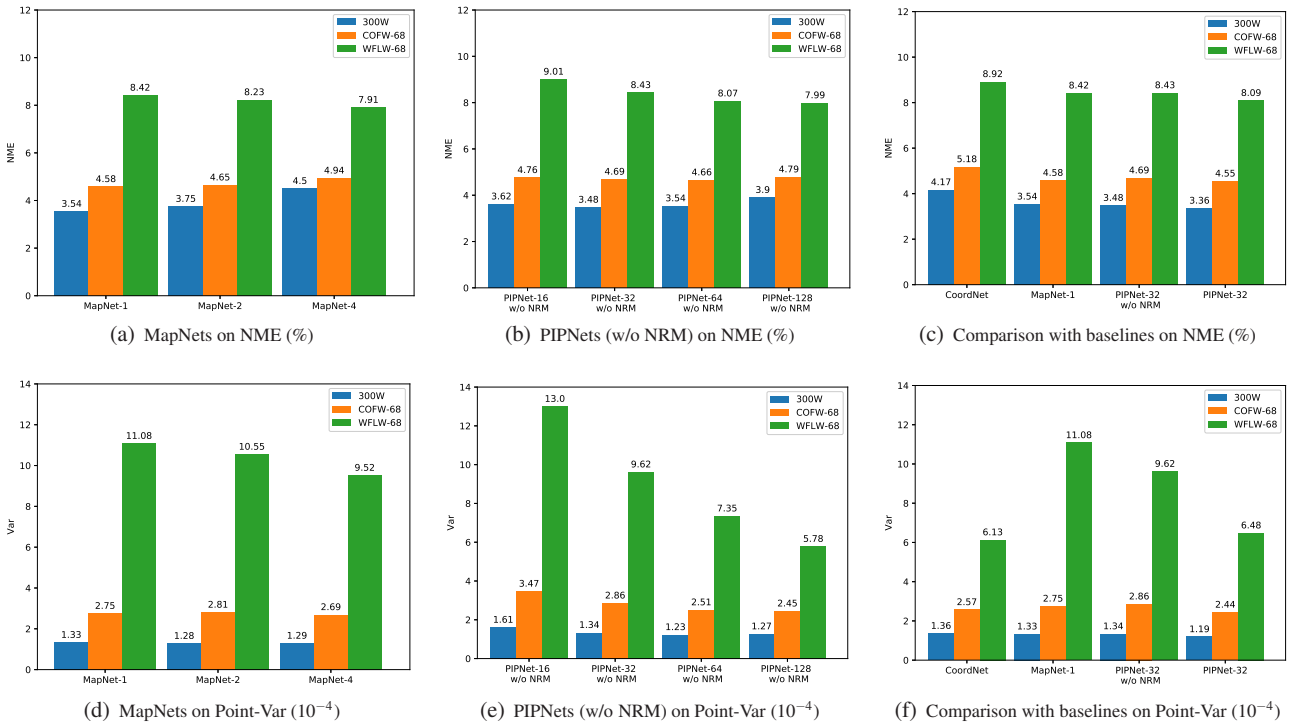

**Fig. 11** NME (%) results of PIPNets with different number of neighbors in NRM, tested on WFLW validation set. The result of PIPNet w/o NRM is also presented for comparison.

### 4.2 Hyperparameters

Two important hyperparameters of PIPNet are the network stride $S$ and the number of neighbors $C$ in the neighbor regression module. In this section, we conduct experiments on WFLW to select appropriate hyperparameters, where 1,500 images from the WFLW training set are randomly selected as our validation set and the rest are used for training (denoted as the sub-training set).

#### *4.2.1 Network Stride*

The default stride of ResNet is 32. To get PIPNets with larger strides, we simply add more Conv-BN-ReLU layers, where the convolutional layer is of 512 channels, kernel size $3 \times 3$, and stride 2. To get smaller strides, the convolutional layer is replaced by a deconvolutional layer with 512 channels, kernel size $4 \times 4$, and stride 2. We train PIPNet without the neighbor regression module on the WFLW sub-training set with different network strides. Table 1 shows the results on the WFLW validation set. As can be seen, $S = 32$ gives the best result, which will be used for the remaining experiments by default. Intuitively, it provides a trade-off between grid classification and offset regression. As discussed in Section 3.3, a larger network stride yields higher accuracy for grid classification, but also results in more errors for offset regression. Therefore, a moderate network stride gives the best performance overall.




(a) MapNets on NME (%) (b) PIPNets (w/o NRM) on NME (%) (c) Comparison with baselines on NME (%)

(d) MapNets on Point-Var ($10^{-4}$) (e) PIPNets (w/o NRM) on Point-Var ($10^{-4}$) (f) Comparison with baselines on Point-Var ($10^{-4}$)

**Fig. 12** NME (%) and Point-Var results of models with various detection heads on 300W, COFW-68, and WFLW-68. (a) NME (%) of MapNets with different strides. (b) NME (%) of PIPNets (w/o NRM) with different strides. (c) NME (%) of three baselines and the proposed model. (d) Point-Var of MapNets with different strides. (e) Point-Var of PIPNets (w/o NRM) with different strides. (f) Point-Var of three baselines and the proposed model.

#### *4.2.2 Number of Neighbors*

To determine an appropriate value of $C$ (number of neighbors in the neighbor regression module), we run PIPNet on the WFLW sub-training set, varying $C$. Figure 11 shows the NME results on the WFLW validation set. Firstly, we notice that neighbor regression module boosts the performance consistently, which confirms its effectiveness. Because the model achieves the best NME around $C = 10$, we use this number for the remaining experiments by default.

### 4.3 Model Analysis

In this section, we conduct experiments to analyze the characteristics of the existing detection heads and the proposed one under the generalizable supervised learning paradigm.

#### *4.3.1 Baselines*

To verify the effectiveness of the proposed detection head, we compare it with the existing ones, namely coordinate regression and heatmap regression. We implement CoordNet, which uses coordinate regression as its detection head and ResNet-18 as its backbone. CoordNet consists of three fully connected layers, each of which has 512, 512 and $2N$ channels respectively, where $N$ is the number of landmarks. Following (Feng et al. 2018), we use the L1 loss for CoordNet to get better results. For heatmap regression, we choose the model from (Xiao et al. 2018) because it is both effective and lightweight. We implement it as MapNet with ResNet-18 and different network strides. The original model uses a stride of 4 for higher speed. Since we observe an improvement in performance with smaller strides, we also implement MapNet with stride 2 and 1 for a more comprehensive comparison. It is worth noting that MapNet requires Gaussian smoothing on training labels, and the Gaussian radius need to be changed adaptively when the network stride varies so that MapNet can achieve optimal performance. In this work, we use 1, 2, and 4 as the radii for MapNet, with a network stride of 4, 2, and 1, respectively. In contrast, PIPNet does not use Gaussian smoothing, which indicates that PIP regression is easier to train than heatmap regression. The loss function of MapNet is the L2 loss. During the inference stage of MapNet, in addition to the location of the highest response, there is also a quarter offset in the direction from the highest response to the second highest response to make up for the loss of accuracy when the stride is larger than 1 (Xiao et al. 2018). The rest of the settings for MapNet are the same as PIPNet. In addition to CoordNet and MapNet,

**Table 2** A summary of the characteristics of various detection heads.

| Detection Head | Efficient | Accurate (Low Bias) | Robust (Low Variance) |
|---|---|---|---|
| Coordinate | ✔ | ✘ | ✔ |
| Heatmap | ✘ | ✔ | ✘ |
| PIP | ✔ | ✔ | ✘ |
| PIP + NRM | ✔ | ✔ | ✔ |

we also use PIPNet without the neighbor regression module as a baseline model.

#### *4.3.2 Bias-Variance Trade-Off*

Bias and variance are two main sources of prediction error. Due to the trade-off between them, a good model usually minimizes both jointly. According to our observations, coordinate regression gives robust but inaccurate landmark predictions, while heatmap regression is accurate on most samples but sensitive to unusual samples. Therefore, we believe neither is optimally minimized in terms of bias and variance jointly. In order to gain a deeper understanding of this situation, we run experiments on three datasets: 300W, COFW-68, and WFLW-68. All the models are trained on the 300W training set, then directly tested on the test sets of 300W, COFW-68, and WFLW-68.

Figure 12(a) gives the NMEs of MapNet with different strides. From the figure, MapNet with stride 1 achieves the lowest NME on 300W and COFW-68, while MapNet with stride 4 performs the best on WFLW-68. A better performance on 300W represents a better capability in fitting data (low bias) because the test data is more similar to the training data. On the other hand, performing better on WFLW-68 means a model has better generalization capability (low variance) because its test images are quite different from the training data. NME contains both bias and variance error, which is not convenient for analyzing the trade-off when comparing different models. Thus, we further compute the variance of the difference between the ground-truth and the predictions for each test image, then average them over a test set to get the Point-Var. Although Point-Var is not exactly the same thing as variance error, it reflects the consistency of predictions, so it can represent variance error to some extent. From the Point-Var results in Figure 12(d), we see that the variance on WFLW-68 decreases as the stride increases, which is consistent with Figure 12(a). Therefore, we observe that the variance of MapNet decreases as its network stride increases, but the bias also increases significantly.

Figure 12(b) and 12(e) give the NME and Point-Var results of PIPNets (without NRM) with different strides. Similar to MapNets, the variance decreases as the stride increases. In general, PIPNets (without NRM) have lower bias than MapNets because the bias of PIPNets does not increase significantly as the stride becomes larger. That is to say, PIP regression is a more general framework than heatmap regression when the network stride varies.

To compare between baselines, we choose a representative model from the MapNets and PIPNets (without NRM), respectively, namely MapNet-1 and PIPNet-32 (without NRM). Figures 12(c) and 12(f) show the NME and Point-Var results of the three baselines and the proposed model on the three test sets. From Figure 12(c), we first see that CoordNet performs poorly on all the datasets, which indicates that coordinate regression tends to have high bias. Compared to CoordNet, MapNet and PIPNet (without NRM) have lower bias but higher variance (see Figure 12(f)). Notably, with the help of the neighbor regression module, PIPNet achieves the lowest NME on all the datasets, which indicates its superiority over the three baselines. From Figure 12(f), we see that the variance of PIPNet-32 is comparable to that of CoordNet, which proves the effectiveness of the neighbor regression module on improving model robustness. Table 2 summarizes the characteristics of these models. As can be seen, PIP regression + NRM is efficient, accurate, and robust at the same time. Thus, we claim that it possesses the advantages of both coordinate and heatmap regression.

### 4.4 Comparison with State of the Arts

We compare PIPNet with several state-of-the-art methods, including RCN (Honari et al. 2016), DAC-CSR (Feng et al. 2017), TSR (Lv et al. 2017), LAB (Wu et al. 2018), Wing (Feng et al. 2018), SAN (Dong et al. 2018), RCN+ (Honari et al. 2018), DCFE (Valle et al. 2018), HG+SA+GHCU (Liu et al. 2019), $TS^3$ (Dong and Yang 2019), LaplaceKL (Robinson et al. 2019), HG-HSLE (Zou et al. 2019), ODN (Zhu et al. 2019a), AVS (Qian et al. 2019), HRNet (Wang et al. 2019a), 3DDE (Valle et al. 2019), AWing (Wang et al. 2019b), DeCaFA (Dapogny et al. 2019), ADA (Chandran et al. 2020), LUVLi (Kumar et al. 2020), Yang et al. (2017), He et al. (2017), Wu and Yang (2017), TCDCN (Zhang et al. 2016), CFSS (Zhu et al. 2015), FHR+STA (Tai et al. 2019), TSTN (Liu et al. 2017a), and Chandran et al. (2020), on six benchmarks, i.e., 300W, COFW, AFLW, WFLW, Menpo 2D, and 300VW.

Table 3 shows the NME results on 300W, COFW, and AFLW. From the table, we first observe that PIPNet with ResNet-18 gives slightly better results than the ones with MobileNets. Moreover, PIPNet with ResNet-18 achieves similar or even better results when compared to the best existing methods. Specifically, PIPNet with ResNet-18 achieves 3.31 NME on the full COFW test set, outperforming all the existing methods. On the full AFLW test set, PIPNet with ResNet-18 gets 1.48 NME, beaten only by Wing (1.47 NME) and LAB (1.25 NME). On the full 300W test set, our lightweight model also obtains very competitive

**Table 3** Comparison with state-of-the-art methods on 300W, COFW, and AFLW. The results are in NME (%), using inter-ocular distance for normalization. * denotes that the method uses external area data for training. Red indicates best, and blue is for second best.

| Method | Year | Backbone | 300W | | | COFW | AFLW |
|---|---|---|---|---|---|---|---|
| | | | Full | Com. | Cha. | Full | Full |
| RCN | 2016 | - | 5.41 | 4.67 | 8.44 | - | 5.6 |
| DAC-CSR | 2017 | - | - | - | - | 6.03 | 2.27 |
| TSR | 2017 | - | 4.99 | 4.36 | 7.56 | - | 2.17 |
| LAB | 2018 | ResNet-18 | 3.49 | 2.98 | 5.19 | 5.58 | 1.85 |
| Wing | 2018 | ResNet-50 | - | - | - | 5.07 | 1.47 |
| SAN | 2018 | ITN-CPM | 3.98 | 3.34 | 6.60 | - | 1.91 |
| RCN+ | 2018 | - | 4.90 | 4.20 | 7.78 | - | - |
| HG+SA+GHCU | 2019 | Hourglass | - | - | - | - | 1.60 |
| TS$^3$ | 2019 | Hourglass+CPM | 3.78 | - | - | - | - |
| LaplaceKL | 2019 | - | 4.01 | 3.28 | 7.01 | - | 1.97 |
| HG-HSLE | 2019 | Hourglass | 3.28 | 2.85 | 5.03 | - | - |
| ODN | 2019 | ResNet-18 | 4.17 | 3.56 | 6.67 | - | 1.63 |
| AVS w/ SAN | 2019 | ITN-CPM | 3.86 | 3.21 | 6.49 | - | - |
| HRNet | 2019 | HRNetV2-W18 | 3.32 | 2.87 | 5.15 | 3.45 | 1.57 |
| AWing | 2019 | Hourglass | **3.07** | 2.72 | **4.52** | - | - |
| DeCaFA | 2019 | Cascaded U-net | 3.69 | - | - | - | - |
| ADA | 2020 | Hourglass | 3.50 | **2.41** | 5.68 | - | - |
| LUVLi | 2020 | DU-Net | 3.23 | 2.76 | 5.16 | - | - |
| LAB* | 2018 | ResNet-18 | - | - | - | 3.92* | **1.25*** |
| RCN+* | 2018 | - | - | - | - | - | 1.59* |
| DCFE* | 2018 | - | 3.24* | 2.76* | 5.22* | - | 2.17* |
| TS$^3$* | 2019 | Hourglass+CPM | 3.49* | - | - | - | - |
| LaplaceKL* | 2019 | - | 3.91* | 3.19* | 6.87* | - | - |
| 3DDE* | 2019 | - | **3.13*** | **2.69*** | 4.92* | - | 2.01* |
| DeCaFA* | 2019 | Cascaded U-net | 3.39* | 2.93* | 5.26* | - | - |
| PIPNet (ours) | - | MobileNetV2 | 3.40 | 2.94 | 5.30 | 3.43 | 1.52 |
| PIPNet (ours) | - | MobileNetV3 | 3.36 | 2.94 | 5.07 | 3.40 | 1.52 |
| PIPNet (ours) | - | ResNet-18 | 3.36 | 2.91 | 5.18 | 3.31 | 1.48 |
| PIPNet (ours) | - | ResNet-50 | 3.24 | 2.80 | 5.03 | **3.18** | 1.44 |
| PIPNet (ours) | - | ResNet-101 | 3.19 | 2.78 | **4.89** | **3.08** | **1.42** |

**Table 4** Comparison with state-of-the-art methods on WFLW. The NME (%) results are evaluated on the full set and six subsets: pose set, expression set, illumination set, make-up set, occlusion set, and blur set, using inter-ocular distance for normalization. * denotes that the method uses external area data for training. Red indicates best, and blue is for second best.

| Method | Year | Backbone | Pose | Expr. | Illu. | M.u. | Occ. | Blur | Full |
|---|---|---|---|---|---|---|---|---|---|
| LAB | 2018 | ResNet-18 | 10.24 | 5.51 | 5.23 | 5.15 | 6.79 | 6.32 | 5.27 |
| Wing | 2018 | ResNet-50 | 8.43 | 5.21 | 4.88 | 5.26 | 6.21 | 5.81 | 4.99 |
| AVS w/ Res-18 | 2019 | ResNet-18 | 9.10 | 5.83 | 4.93 | 5.47 | 6.26 | 5.86 | 5.25 |
| AVS w/ LAB | 2019 | ResNet-18 | 8.21 | 5.14 | 4.51 | 5.00 | 5.76 | 5.43 | 4.76 |
| AVS w/ SAN | 2019 | ITN-CPM | 8.42 | 4.68 | **4.24** | 4.37 | 5.60 | **4.86** | 4.39 |
| HRNet | 2019 | HRNetV2-W18 | 7.94 | 4.85 | 4.55 | 4.29 | 5.44 | 5.42 | 4.60 |
| AWing | 2019 | Hourglass | **7.38** | 4.58 | 4.32 | **4.27** | **5.19** | **4.96** | **4.36** |
| DeCaFA | 2019 | Cascaded U-net | - | - | - | - | - | - | 5.01 |
| LUVLi | 2020 | DU-Net | - | - | - | - | - | - | 4.37 |
| 3DDE* | 2019 | - | 8.62* | 5.21* | 4.65* | 4.60* | 5.77* | 5.41* | 4.68* |
| DeCaFA* | 2019 | Cascaded U-net | 8.11* | 4.65* | 4.41* | 4.63* | 5.74* | 5.38* | 4.62* |
| PIPNet (ours) | - | MobileNetV2 | 8.76 | 4.86 | 4.56 | 4.60 | 6.04 | 5.53 | 4.79 |
| PIPNet (ours) | - | MobileNetV3 | 8.22 | 4.75 | 4.49 | 4.46 | 5.72 | 5.31 | 4.65 |
| PIPNet (ours) | - | ResNet-18 | 8.02 | 4.73 | 4.39 | 4.38 | 5.66 | 5.25 | 4.57 |
| PIPNet (ours) | - | ResNet-50 | 7.98 | **4.54** | 4.35 | **4.27** | 5.65 | 5.19 | 4.48 |
| PIPNet (ours) | - | ResNet-101 | **7.51** | **4.44** | **4.19** | **4.02** | **5.36** | 5.02 | **4.31** |

**Table 5** Comparison with the **three best** teams from the Menpo 2D Challenge on the Menpo 2D benchmark. The NME (%) results are evaluated using the face diagonal as the normalization distance. Red indicates best, and blue is for second best.

| Method | Year | Backbone | Semi-frontal | Profile |
|---|---|---|---|---|
| J. Yang et al. | 2017 | Hourglass | **1.20** | **1.72** |
| Z. He et al. | 2017 | - | 1.39 | 2.47 |
| W. Wu and S. Yang | 2017 | VGG-16 | 1.35 | 2.17 |
| PIPNet (ours) | - | Mob. NetV2 | 1.37 | 2.04 |
| PIPNet (ours) | - | Mob. NetV3 | 1.35 | 2.04 |
| PIPNet (ours) | - | ResNet-18 | 1.34 | 2.01 |
| PIPNet (ours) | - | ResNet-50 | 1.30 | 1.95 |
| PIPNet (ours) | - | ResNet-101 | **1.27** | **1.89** |

**Table 6** Comparison with state-of-the-art methods on 300VW. The NME (%) results are evaluated with inter-ocular being the normalization distance. Red indicates best, and blue is for second best.

| Method | Year | Backbone | Cat.1 | Cat.2 | Cat.3 |
|---|---|---|---|---|---|
| CFSS | 2015 | - | 7.68 | 6.42 | 13.7 |
| TCDCN | 2016 | - | 7.66 | 6.77 | 15.0 |
| TSTN | 2017 | - | 5.36 | 4.51 | 12.8 |
| FHR+STA | 2019 | Hourglass | 4.42 | 4.18 | 5.98 |
| DeCaFA | 2019 | Cas. U-net | 3.82 | 3.63 | 6.67 |
| HG+SA+GHCU | 2019 | Hourglass | 3.85 | **3.46** | 7.51 |
| P. Chandran et al. | 2020 | Hourglass | 4.17 | 3.89 | 7.28 |
| PIPNet (ours) | - | Mob. NetV2 | 3.10 | 3.60 | 5.60 |
| PIPNet (ours) | - | Mob. NetV3 | 3.07 | 3.55 | 5.57 |
| PIPNet (ours) | - | ResNet-18 | **3.04** | 3.51 | **5.32** |
| PIPNet (ours) | - | ResNet-50 | 3.05 | 3.49 | 5.35 |
| PIPNet (ours) | - | ResNet-101 | **3.03** | **3.42** | **5.28** |

result (3.36 NME), and is even better than some methods that use external area data, such as $TS^3$ (3.49 NME, with unlabeled AFLW training data), LaplaceKL (3.91 NME, with 70K unlabeled MegaFace images), and DeCaFA (3.39 NME, with WFLW and CelebA training sets). We notice that most of the state-of-the-art methods use heavyweight backbones like Hourglass, ResNet-50, and HRNetV2-W18. Therefore, we also equip PIPNet with heavier backbones to explore better results. Notably, PIPNet with ResNet-101 achieves state of the art on COFW (3.08 NME), and is significantly better than the best existing model (HRNet, 3.45 NME). On the full AFLW test set, our PIPNet with ResNet-101 is only outperformed by LAB (1.42 vs. 1.25), which uses external boundary data for training. As for 300W, PIPNet with ResNet-101 obtains the second best result on the full test set and challenging set, among the methods that do not use external area data.

Table 4 shows the NME results of the best existing methods and the proposed PIPNets with different backbones on the full WFLW test set and six subsets. As can be seen, our lightweight models (with MobileNetV3 and ResNet-18) already achieve comparable performance to the state of the arts. Again, PIPNet with ResNet-101 obtains the state of the art on the full set (4.31 NME) as well as three subsets. Among the six benchmarks, WFLW is the closest to an uncontrolled environment because it contains more diverse scenes and more in-the-wild images. Consequently, the superior performance on WFLW demonstrates the effectiveness of PIPNet on images in the wild. In Section 4.6, we also compare PIPNets with state-of-the-art methods in terms of speed-accuracy trade-off through WFLW results.

To evaluate the compatibility of our model on large poses, we compare PIPNet with the three best teams from the Menpo 2D Challenge (Zafeiriou et al. 2017). Table 5 shows the results. Notably, our lightweight models (with MobileNetV3 and ResNet-18) achieve better performance than the second and third best teams (He et al. 2017; Wu and Yang 2017) on both tracks. Our best model, PIPNet with ResNet-101, is slightly worse than the winner (Yang et al. 2017) of the challenge (1.27 vs. 1.20 on semi-frontal; 1.89 vs. 1.72 on profile). It is worth noting that our models are trained without any specific adaptation to Menpo 2D, while the winner utilized an extra face detector (Chen et al. 2016) and facial landmark detector (Bansal et al. 2016) for first-step transformation since the dataset contains faces with large view angles (Yang et al. 2017).

To further validate the robustness of the proposed method, we conduct an evaluation on 300VW. Table 6 shows the results of PIPNets and prior works. First of all, we notice that the performance gaps between PIPNets with different backbones are not significant. This is due to the fact that there is a limited number of identities in the training set, which may lead to overfitting for larger backbones. Despite not using temporal information, PIPNet with ResNet-101 obtains better results than all the existing methods on all three categories. In particular, our best model significantly outperforms prior works on category 1 and 3, which indicates the superiority of our model in accuracy and robustness, respectively. Three sample videos are presented in the supplementary materials to demonstrate the robustness of our models, including the ones trained on 300VW and WFLW with supervised learning and the one trained on CelebA with semi-supervised learning (see Section 4.5).

### 4.5 Self-Training with Curriculum

We first verify the effectiveness of the proposed self-training with curriculum (STC) strategy by running experiments under the UDA paradigm. Specifically, we use the 300W training set as the only labeled data, and our test data includes the test sets of 300W, COFW-68, and WFLW-68. The training sets of COFW-68 and WFLW-68 are used as unlabeled data. The self-training method without curriculum is used as a baseline for comparison. We also implemented the domain adversarial neural networks (DANN) (Ganin and Lempitsky

**Table 7** Comparison of PIPNets under different training paradigms and strategies on the 300W, COFW-68, and WFLW-68 test sets. The results are in NME (%), normalized by inter-ocular distance.

| Paradigm | Method | Unlabeled Training Data | Test Data | | |
|---|---|---|---|---|---|
| | | | 300W | COFW-68 | WFLW-68 |
| GSL | - | - | 3.36 | 4.55 | 8.09 |
| UDA | DANN | COFW-68+WFLW-68 | 3.42 (+1.8%) | 4.55 (-0.0%) | 8.01 (-1.0%) |
| | Self-training | COFW-68+WFLW-68 | 3.35 (-0.3%) | 4.34 (-4.6%) | 7.45 (-7.9%) |
| | STC | COFW-68+WFLW-68 | 3.34 (-0.6%) | 4.28 (-5.9%) | 7.28 (-**10.0%**) |
| GSSL | DANN | CelebA | 3.43 (+0.3%) | 4.56 (+0.2%) | 8.13 (+1.5%) |
| | Self-training | CelebA | 3.27 (-2.7%) | 4.32 (-5.1%) | 7.77 (-4.0%) |
| | STC | CelebA | 3.23 (-**3.9%**) | 4.23 (-**7.0%**) | 7.53 (-6.8%) |

**Table 8** Comparison with prior works on 300W and COFW-68 test sets. The results are in NME (%), normalized by inter-ocular distance.

| Method | Paradigm | Backbone | Test Data | |
|---|---|---|---|---|
| | | | 300W | COFW-68 |
| LAB | GSL | ResNet-18 | 3.49 | 4.62 |
| ODN | GSL | ResNet-18 | 4.17 | 5.30 |
| AVS w/ SAN | GSL | ITN-CPM | 3.86 | 4.43 |
| PIPNet (ours) | GSL | ResNet-18 | 3.36 | 4.55 |
| PIPNet (ours) | GSSL | ResNet-18 | **3.23** | **4.23** |

2015; Ganin et al. 2016), a classic UDA method for classification, as another baseline. Furthermore, the results under the GSL paradigm are also presented, where the model is trained on 300W with supervised learning, then evaluated on the test sets without adaptation. From Table 7, we see that PIPNet with ResNet-18 achieves 3.36, 4.55, and 8.09 NME on 300W, COFW-68, and WFLW-68, respectively, under the GSL paradigm. The results indicate large domain gaps between the three datasets, especially WFLW-68. The improvements of DANN against GSL paradigm on COFW-68 (-0.0%) and WFLW-68 (-1.0%) are limited, and its performance even degrades on 300W (+1.8%). This implies that it is difficult to directly apply UDA methods from classification task to facial landmark detection due to the intrinsic discrepancy between the two tasks. When applying the standard self-training method, the domain gaps are considerably reduced, with the NME on COFW-68 and WFLW-68 reduced by 4.6% and 7.9%, respectively. With the help of the proposed STC strategy, the NME reduction on COFW-68 and WFLW-68 further becomes 5.9% and 10.0%, respectively. As a by-product, the NME of 300W is also slightly improved by 0.6% due to the increased training data. Thus, the proposed STC is a simple yet effective method that is able to consistently boost the cross-domain performance. STC could also be applied to other vision tasks such as detection.

Despite the considerable improvement under UDA, it still may not be ideal for real applications. For example, unlabeled data is not always available for the target domains, or the target domains may even be unknown. Such situations are not uncommon for models running in the wild. Therefore, we aim to go beyond the UDA paradigm and further explore GSSL. To be more specific, the labeled data and test data remain the same as in UDA, but the unlabeled data is changed to CelebA. In this way, the model never sees an image (whether labeled or unlabeled) from the target domain during training. From Table 7, we observe that the results of standard self-training under GSSL are consistently better than those under the GSL paradigm, which indicates the feasibility of GSSL for real applications (i.e., unlabeled data does not necessarily need to be from target domains). In other words, it is feasible to improve the generalization capability of a model using massive amounts of unlabeled data, even if the target domain is unknown. As for DANN, its performance is even worse than the GSL baseline. Again, the STC strategy outperforms standard self-training on all the test sets under GSSL, which confirms its effectiveness. One interesting finding is that the GSSL paradigm obtains better results than UDA on 300W and COFW-68, but worse results on WFLW-68. This is because CelebA is a much larger dataset than COFW-68 and WFLW-68, and its domain is closer to that of 300W and COFW-68. These findings tell us that unlabeled data should not only be collected in large amounts, but also needs to be as diverse as possible under the GSSL paradigm in order to enable models to generalize better on more cross-domain datasets. To demonstrate the superiority of GSSL over GSL, we also compare our model with prior works that conduct direct cross-domain evaluation on COFW-68, including LAB (Wu et al. 2018), ODN (Zhu et al. 2019a), and AVS with SAN (Qian et al. 2019). Table 8 gives the NME results on the test sets of 300W and COFW-68, where the listed methods are all trained on the labeled 300W training set, with the images from COFW being unavailable during training. As shown in the table, under the same GSL paradigm, PIPNet already obtains the best result on 300W, and is quite competitive on COFW-68 (only inferior to AVS with SAN). Under the GSSL paradigm, PIPNet obtains even better results on 300W (3.23 NME), and outperforms AVS with SAN by 4.5% (4.23 NME vs. 4.43 NME) on COFW-68, yielding the new best result on the COFW-68 test set. Therefore, GSSL is a promising and scalable paradigm for improving cross-domain generalization ability.

**Table 9** Parameter size, GFLOPs and FPS of existing methods, baselines, and our model.

| Method | Year | Backbone | #Param. | GFLOPs | FPS (CPU) | FPS (GPU) |
|---|---|---|---|---|---|---|
| LAB | 2018 | ResNet-18 | 24.1M+28.3M | 26.7+2.4 | 2.1 | 16.7 |
| Wing | 2018 | ResNet-50 | 91.0M | 5.5 | 8 | 30 |
| AVS w/ LAB | 2019 | ResNet-18 | 28.3M | 2.4 | 2.1 | 16.7 |
| AVS w/ SAN | 2019 | ITN-CPM | 7.8M+19.4M | 32.7+30.1 | 5.3 | 61 |
| HRNet | 2019 | HRNetV2-W18 | 9.7M | 4.8 | 4.4 | 11.7 |
| AWing | 2019 | Hourglass | 24.1M | 26.7 | 1.8 | 24.2 |
| DeCaFA | 2019 | Cascaded U-net | 10M | - | - | 32 |
| 3DDE | 2019 | - | - | - | - | 12.5 |
| LUVLi | 2020 | DU-Net | - | - | - | 58.8 |
| G-RMI | 2017 | ResNet-50 | 23.9M | 25.5 | 0.1 | 6.8 |
| G-RMI | 2017 | ResNet-101 | 42.9M | 45.1 | 0.1 | 5.8 |
| CoordNet | - | ResNet-18 | 28.3M | 2.4 | **37.5** | **256** |
| CoordNet | - | ResNet-50 | 91.0M | 5.5 | 14.8 | 115 |
| CoordNet | - | ResNet-101 | 110.0M | 10.4 | 9.4 | 62 |
| MapNet (S=2) | - | ResNet-18 | 28M | 3.0 | 3.5 | 200 |
| MapNet (S=2) | - | ResNet-50 | 52.9M | 6.0 | 2.7 | 82 |
| MapNet (S=2) | - | ResNet-101 | 71.9M | 10.9 | 2.3 | 46 |
| PIPNet (ours) | - | MobileNetV2 | **4.2M** | 0.5 | 29.5 | 121 |
| PIPNet (ours) | - | MobileNetV3 | 4.5M | **0.4** | 28.4 | 80 |
| PIPNet (ours) | - | ResNet-18 | 12.0M | 2.4 | 35.7 | 200 |
| PIPNet (ours) | - | ResNet-50 | 26.7M | 5.6 | 13.8 | 99 |
| PIPNet (ours) | - | ResNet-101 | 45.7M | 10.5 | 8.8 | 56 |

### 4.6 Speed

Table 9 lists the parameter size, GFLOPs, and speed of the methods in Table 4. Additionally, we add CoordNets and MapNets equipped with different backbones. The MapNets with stride 2 are used here because they have better speed-accuracy trade-off than the ones with stride 1 and 4. G-RMI (Papandreou et al. 2017) is also implemented as a baseline, where binary cross-entropy and smooth L1 loss are used for classification and regression, and their loss scalars are set to 4 and 1 respectively. For G-RMI, the radius of the disk is set to 15, which gives the best performance. The speeds are averaged over WFLW test set with a batch size of 1, and given in frames per second (FPS). Our code was implemented in PyTorch. The CPU is Intel Xeon E5-2698 v4 @2.20GHz and the GPU is an NVIDIA Tesla V100. From the table, we find that MobileNets are not as efficient as ResNet-18, especially on GPU, although they have smaller GFLOPs. We believe this is related to their implementation in PyTorch, which is not fully optimized. Therefore, we use ResNets as the backbones of PIPNet for the comparison of speed-accuracy trade-off. Figures 1(a) and 1(b) show the speed-accuracy trade-off comparison of the existing methods, baselines, and PIPNet on CPU and GPU, respectively. The NME results are obtained on the WFLW test set with an image size of 256×256. The existing methods with * were tested by us under the same environment as our models. As we can see, PIPNet achieves the best speed-accuracy trade-off on both CPU and GPU, thanks to the lightweight detection head. G-RMI obtains comparable NMEs to PIPNet, but its speed is about 100× and 10× slower on CPU and GPU respectively, due to the heavy computations on high-resolution feature maps. Similarly, MapNet is much slower than PIPNet on CPU due to its heavy detection head, although the NMEs are satisfactory. Interestingly, MapNet becomes much faster on GPU, and we believe this is because the deconvolutional layers are highly optimized in PyTorch with GPU. In contrast, CoordNet is faster than PIPNet, but its accuracy is significantly worse due to the biased predictions. It is worth noting that PIPNet with ResNet-18 is the only model that runs in real-time (35.7 FPS) on CPU while still achieving competitive result to state-of-the-art methods.

## 5 Conclusion

In this work, we propose a novel facial landmark detection framework named PIPNet. PIPNet consists of three new modules, namely PIP regression, neighbor regression, and self-training with curriculum. PIP regression is a lightweight detection head based on heatmap regression. To be more specific, it only predicts low-resolution score heatmaps, where each heatmap pixel further predicts offsets within itself to yield accurate predictions. By eliminating the repeated upsampling layers, the proposed detection head saves considerable computational cost, especially on lightweight computing devices. The neighbor regression module is another lightweight module that aims to improve model robustness by fusing the predictions from neighboring landmarks. Self-training with curriculum is a new strategy that can utilize unlabeled data across domains. By gradually increasing the difficulty of the tasks for pseudo-labeled data,

self-training with curriculum introduces less errors from the estimated pseudo-labels, enabling the model to generalize better on cross-domain datasets. In summary, extensive experiments show that PIPNet is an efficient, accurate, and robust facial landmark detector that can run in the wild on lightweight devices.

## References


Bansal A, Nanduri A, Castillo CD, Ranjan R, Chellappa R (2016) Umdfaces: An annotated face dataset for training deep networks. arXiv: 161101484

Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. In: ICML

Burgos-Artizzu XP, Perona P, Dollár P (2013) Robust face landmark estimation under occlusion. In: ICCV

Chandran P, Bradley D, Gross M, Beeler T (2020) Attention-driven cropping for very high resolution facial landmark detection. In: CVPR

Chen D, Hua G, Wen F, Sun J (2016) Supervised transformer network for efficientface detection. In: ECCV

Chen L, Su H, Ji Q (2019) Face alignment with kernel density deep neural network. In: ICCV

Chen Y, Li W, Sakaridis C, Dai D, Gool LV (2018) Domain adaptive faster r-cnn for object detection in the wild. In: CVPR

Dapogny A, Bailly K, Cord M (2019) Decafa: Deep convolutional cascade for face alignment in the wild. In: ICCV

Deng J, Roussos A, Chrysos G, Ververas E, Kotsia I, Shen J, Zafeiriou S (2019) The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking. IJCV

Deng J, Guo J, Zhou Y, Yu J, Kotsia I, Zafeiriou S (2020) Retinaface: Single-stage dense face localisation in the wild. In: CVPR

Deng W, Zheng L, Ye Q, Kang G, Yang Y, Jiao J (2018) Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In: CVPR

Dong X, Yang Y (2019) Teacher supervises students how to learn from partially labeled images for facial landmark detection. In: ICCV

Dong X, Yan Y, Ouyang W, Yang Y (2018) Style aggregated network for facial landmark detection. In: CVPR

Feng ZH, Kittler J, Christmas W, Huber P, Wu XJ (2017) Dynamic attention-controlled cascaded shape regression exploiting training data augmentation and fuzzy-set sample weighting. In: CVPR

Feng ZH, Kittler J, Awais M, Huber P, Wu XJ (2018) Wing loss for robust facial landmark localisation with convolutional neural networks. In: CVPR

Ganin Y, Lempitsky V (2015) Unsupervised domain adaptation by backpropagation. In: ICML

Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. JMLR

Ghiasi G, Fowlkes CC (2014) Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In: CVPR

He Z, Zhang J, Kan M, Shan S, Chen X (2017) Robust fec-cnn: A high accuracy facial landmark detection system. In: CVPRW

Honari S, Yosinski J, Vincent P, Pal C (2016) Recombinator networks: Learning coarse-to-fine feature aggregation. In: CVPR

Honari S, Molchanov P, Tyree S, Vincent P, Pal C, Kautz J (2018) Improving landmark localization with semi-supervised learning. In: CVPR

Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V, Le QV, Adam H (2019) Searching for mobilenetv3. In: ICCV

Islam MA, Jia S, Bruce ND (2020) How much position information do convolutional neural networks encode? In: ICLR

Kang G, Jiang L, Yang Y, Hauptmann AG (2019) Contrastive adaptation network for unsupervised domain adaptation. In: CVPR

Khan MH, McDonagh J, Tzimiropoulos G (2017) Synergy between face alignment and tracking via discriminative globalconsensus optimization. In: ICCV

Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: ICLR

Koestinger M, Wohlhart P, Roth PM, Bischof H (2011) Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies

Kumar A, Marks TK, Mou W, Wang Y, Jones M, Cherian A, Koike-Akino T, Liu X, Feng C (2020) Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In: CVPR

Liao S, Jain AK, Li SZ (2013) Partial face recognition: Alignment-free approach. TPAMI

Liu H, Lu J, Feng J, Zhou J (2017a) Two-stream transformer networks for video-based face alignment. TPAMI

Liu W, Wen Y, Yu Z, Li M, Raj B, Song L (2017b) Sphereface: Deep hypersphere embedding for face recognition. In: CVPR

Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. In: ICCV

Liu Z, Zhu X, Hu G, Guo H, Tang M, Lei Z, Robertson NM, Wang J (2019) Semantic alignment: Finding semantically consistent ground-truth for facial landmark detection. In: CVPR

Long M, Cao Y, Wang J, Jordan MI (2015) Learning transferable features with deep adaptation networks. In: ICML

Lv J, Shao X, Xing J, Cheng C, Zhou X (2017) A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In: CVPR

Merget D, Rock M, Rigoll G (2018) Robust facial landmark detection via a fully-convolutional local-global context network. In: CVPR

Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In: ECCV

Papandreou G, Zhu T, Kanazawa N, Toshev A, Tompson J, Bregler C, Murphy K (2017) Towards accurate multi-person pose estimation in the wild. In: CVPR

Papandreou G, Zhu T, Chen LC, Gidaris S, Tompson J, Murphy K (2018) Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: ECCV

Peng P, Xiang T, Wang Y, Pontil M, Gong S, Huang T, Tian Y (2016) Unsupervised cross-dataset transfer learning for person re-identification. In: CVPR

Qian S, Sun K, Wu W, Qian C, Jia J (2019) Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation. In: ICCV

Ren S, Cao X, Wei Y, Sun J (2016) Face alignment via regressing local binary features. TIP

Robinson JP, Li Y, Zhang N, Fu Y, Tulyakov S (2019) Laplace landmark localization. In: ICCV

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), LNCS, vol 9351, pp 234–241

Sagonas C, Tzimiropoulos G, Zafeiriou S, Pantic M (2013) 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: ICCV Workshops

Saito K, Ushiku Y, Harada T, Saenko K (2019) Strong-weak distribution alignment for adaptive object detection. In: CVPR

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. arXiv: 180104381

Shen J, Zafeiriou S, Chrysos GG, Kossaifi J, Tzimiropoulos G, Pantic M (2015) The first facial landmark tracking in-the-wild challenge:

Benchmark and results. In: ICCVW
Sun Y, Wang X, Tang X (2013) Deep convolutional network cascade for facial point detection. In: CVPR
Tai Y, Liang Y, Liu X, Duan L, Li J, Wang C, Huang F, Chen Y (2019) Towards highly accurate and stable face alignment for high-resolution videos. In: AAAI
Taigman Y, Yang M, Ranzato M, Wolf L (2014) Deepface: Closing the gap to human-level performance in face verification. In: CVPR
Tang Z, Peng X, Geng S, Wu L, Zhang S, Metaxas D (2018) Quantized densely connected u-nets for efficient landmark localization. In: ECCV
Thies J, Zollhofer M, Stamminger M, Theobalt C, Niebner M (2016) Face2face: Real-time face capture and reenactment of rgb videos. In: CVPR
Trigeorgis G, Snape P, Nicolaou MA, Antonakos E, Zafeiriou S (2016) Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In: CVPR
Valle R, Buenaposada JM, Valdés A, Baumela L (2018) A deeply-initialized coarse-to-fine ensemble of regression trees for face alignment. In: ECCV
Valle R, Buenaposada JM, Valdés A, Baumela L (2019) Face alignment using a 3d deeply-initialized ensemble of regression trees. Computer Vision and Image Understanding 189
Wang J, Sun K, Cheng T, Jiang B, Deng C, Zhao Y, Liu D, Mu Y, Tan M, Wang X, Liu W, Xiao B (2019a) Deep high-resolution representation learning for visual recognition. TPAMI
Wang X, Bo L, Fuxin L (2019b) Adaptive wing loss for robust face alignment via heatmap regression. In: ICCV
Wei SE, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. In: CVPR
Wu W, Yang S (2017) Leveraging intra and inter-dataset variations for-robust face alignment. In: CVPRW
Wu WW, Qian C, Yang S, Wang Q, Cai Y, Zhou Q (2018) Look at boundary: A boundary-aware face alignment algorithm. In: CVPR
Xiao B, Wu H, Wei Y (2018) Simple baselines for human pose estimation and tracking. In: ECCV
Yang J, Liu Q, Zhang K (2017) Stacked hourglass network for robust facial landmark localisation. In: CVPRW
Yang S, Luo P, Loy CC, Tang X (2016) Wider face: A face detection benchmark. In: CVPR
Yu HX, Wu A, Zheng WS (2017) Cross-view asymmetric metric learning for unsupervised person re-identification. In: ICCV
Yu HX, Zheng WS, Wu A, Guo X, Gong S, Lai JH (2019) Unsupervised person re-identification by soft multilabel learning. In: CVPR
Zafeiriou S, Trigeorgis G, Chrysos G, Deng J, Shen J (2017) The menpo facial landmark localisation challenge: A step towards the solution. In: CVPRW
Zhang Z, Luo P, Loy CC, Tang X (2016) Learning deep representation for face alignment with auxiliary attributes. TPAMI
Zhao F, Liao S, Xie GS, Zhao J, Zhang K, Shao L (2020) Unsupervised domain adaptation with noiseresistible mutual-training for personre-identification. In: ECCV
Zhong Z, Zheng L, Li S, Yang Y (2018) Generalizing a person retrieval model hetero- and homogeneously. In: ECCV
Zhu M, Shi D, Zheng M, Sadiq M (2019a) Robust facial landmark detection via occlusion-adaptive deep networks. In: CVPR
Zhu S, Li C, Loy CC, Tang X (2015) Face alignment by coarse-to-fine shape searching. In: CVPR
Zhu S, Li C, Loy CC, Tang X (2016) Unconstrained face alignment via cascaded compositional learning. In: CVPR
Zhu X, Pang J, Yang C, Shi J, Lin D (2019b) Adapting object detectors via selective cross-domain alignment. In: CVPR
Zou X, Zhong S, Yan L, Zhao X, Zhou J, Wu Y (2019) Learning robust facial landmark detection via hierarchical structured ensemble. In: ICCV