Learning to Refine Human Pose Estimation

Mihai Fieraru Anna Khoreva Leonid Pishchulin Bernt Schiele
{mfieraru, khoreva, leonid, schiele}@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarland Informatics Campus
Saarbrücken, Germany

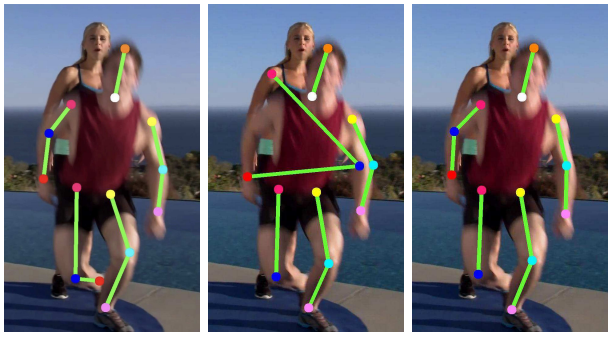
Abstract

Multi-person pose estimation in images and videos is an important yet challenging task with many applications. Despite the large improvements in human pose estimation enabled by the development of convolutional neural networks, there still exist a lot of difficult cases where even the state-of-the-art models fail to correctly localize all body joints. This motivates the need for an additional refinement step that addresses these challenging cases and can be easily applied on top of any existing method. In this work, we introduce a pose refinement network (PoseRefiner) which takes as input both the image and a given pose estimate and learns to directly predict a refined pose by jointly reasoning about the input-output space. In order for the network to learn to refine incorrect body joint predictions, we employ a novel data augmentation scheme for training, where we model "hard" human pose cases. We evaluate our approach on four popular large-scale pose estimation benchmarks such as MPII Single- and Multi-Person Pose Estimation, PoseTrack Pose Estimation, and PoseTrack Pose Tracking, and report systematic improvement over the state of the art.

1. Introduction

The task of human pose estimation is to correctly localize and estimate body poses of all people in the scene. Human poses provide strong cues and have shown to be an effective representation for a variety of tasks such as activity recognition, motion capture, content retrieval and social signal processing. Recently, human pose estimation performance has improved significantly due to the use of deep convolutional neural networks [36, 24, 46, 33, 18] and availability of large-scale datasets [31, 2, 3, 29].

Although great progress has been made, the problem remains far from being solved. There still exist a lot of challenging cases, such as person-person occlusions, close proximity of similar looking people, rare body configurations, partial visibility of people and cluttered backgrounds. Despite the great representational power, current deep learning-based approaches are not explicitly trained to address such hard cases and often output incorrect body pose predictions such as spurious body configurations, merges of body joints of different people, confusion between similarly looking left and right limbs, or missing joints.



Ground Truth Pose | Initial Pose Estimate | Refined Output Pose

Figure 1: Example of the proposed pose refinement. Starting from an image and an estimated body pose **(central)**, our refinement method `PoseRefiner` outputs a denoised body pose **(right)**. The system learns to fuse the appearance of the person and an estimation of its pose structure in order to better localize each body joint. It is trained to specifically target common errors of human pose estimation methods, e.g. merges of body joints of different people in close proximity and confusion between right/left joints.

In this work, we propose a novel human pose refinement approach that is explicitly trained to address such hard cases. Our simple yet effective pose refinement method can be applied on top of any body pose prediction computed by an arbitrary human pose estimation approach, and thus is complementary to current approaches [46, 33, 6, 23]. As we demonstrate empirically, the proposed pose refinement allows to push the state of the art on several standard benchmarks of single- and multi-person pose estimation [3, 2], as well as articulated pose tracking [2]. In more detail,

given an RGB image and a body pose estimate, we aim to output a refined human pose by exploiting the dependencies between the image and the inherent structure of the provided body pose (see Figure 1). This makes it easier for the network to identify what is wrong with input prediction and find a way to refine it. We employ a fully convolutional ResNet-based architecture and propose an elaborate data augmentation scheme for training. To model challenging cases, we propose a novel data augmentation procedure that allows to synthesize possible input poses and make the network learn to identify the erroneous body joint predictions and to refine them. We refer to the proposed approach as `PoseRefiner`.

We evaluate our approach on four human pose estimation benchmarks, namely MPII Single Person [3], MPII Multi-Person [3], PoseTrack Multi-Person Pose Estimation [2], and PoseTrack Multi-Person Pose Tracking [2]. We report consistent improvement after applying the proposed refinement network to pose predictions given by various state-of-the-art approaches [48, 14, 23, 26, 17, 46, 33, 44, 6, 7] across different datasets and tasks, showing the effectiveness and generality of the proposed framework. With our refinement network, we improve the best reported results for multi-person pose estimation and pose tracking on MPII Human Pose and PoseTrack datasets.

In summary, our contributions are as follows:

- We introduce an effective post-processing technique for body joint refinement in human pose estimation tasks, that works on top of any existing human body pose estimation approach. Our proposed pose refinement network is efficient due to its feed-forward architecture, simple and end-to-end trainable.
- We propose a training data augmentation scheme for error correction, which enables the network to identify the erroneous body joint predictions and to learn a way to refine them.
- We show that our refinement model allows to systematically improve over various state-of-the-art methods and achieve top performing results on four different benchmarks.

The rest of the paper is organized as follows. Section 2 provides an overview of the related work and positions the proposed approach with respect to earlier work. Section 3 describes the proposed pose refinement network and data augmentation for error correction of human body pose estimation. Experimental results are presented in Section 4. Section 5 concludes the paper.

2. Related Work

Our proposed approach is related to previous work on single- and multi-person pose estimation, articulated tracking as well as refinement/error correction methods, as described next.

Single-person pose estimation. Classical methods [15, 47, 4, 40, 10, 38] formulate single person pose estimation as a pictorial structure or graphical model problem and predict body joint locations using only hand-designed features. More recent methods [43, 42, 45, 34, 30, 46] rely on localizing body joints by employing convolutional neural networks (CNNs), which contributed to large improvement in human pose estimation. [43] directly predicts joint coordinates via a cascade of CNN pose regressors, while further improvement in the performance is achieved by predicting heatmaps of each body joint [42, 30] and using very deep CNNs with multi-stage architectures [45]. Our method is complementary to current approaches, as it is able to use their predictions as input and further improve their results, see Section 4.2 for details.

Multi-person pose estimation. Compared to single person pose estimation, multi-person pose estimation requires parsing of the full body poses of all people in the scene and is a much more challenging task due to occlusions, various articulations and interactions between people. Multi-person pose estimation methods can be grouped into two types: top-down and bottom-up approaches.

Top-down approaches [36, 21, 19, 12, 25, 16] employ a person detector and then perform single-person pose estimation for each detected person. These methods highly depend on the reliability of the person detector and are known to have trouble recovering poses of people in close proximity to each other and/or with overlapping body parts. Thus, the output predictions of top-down methods might benefit from an additional refinement step proposed in this work.

Bottom-up methods [6, 39, 24, 33, 27] first predict all body joints and then group them into full poses of different people. Instead of applying person detection, these methods rely on context information and inter body joint relationships. However, modeling the joint relationships might not be as reliable, causing mistakes like failure to disambiguate poses of different people or grouping body parts of the same person into different clusters. Our refinement approach can particularly help in this scenario, as it re-estimates joint locations by taking the structure of the body pose into account.

Articulated pose tracking. Most articulated pose tracking methods rely on a two-stage framework, which first employs a per-frame pose estimator and then smooths the predictions over time. [41] proposes a model combining a CNN and a CRF to jointly optimize per-frame predictions with the CRF, smoothing the predictions over space and time. [25, 23, 26] employ a bottom-up strategy, they first detect body joints of all people in all frames of the video and then an integer program optimization is solved grouping joints into people over time. [14] applies 3D Mask R-CNN [19] over short video clips, producing a tubelet with

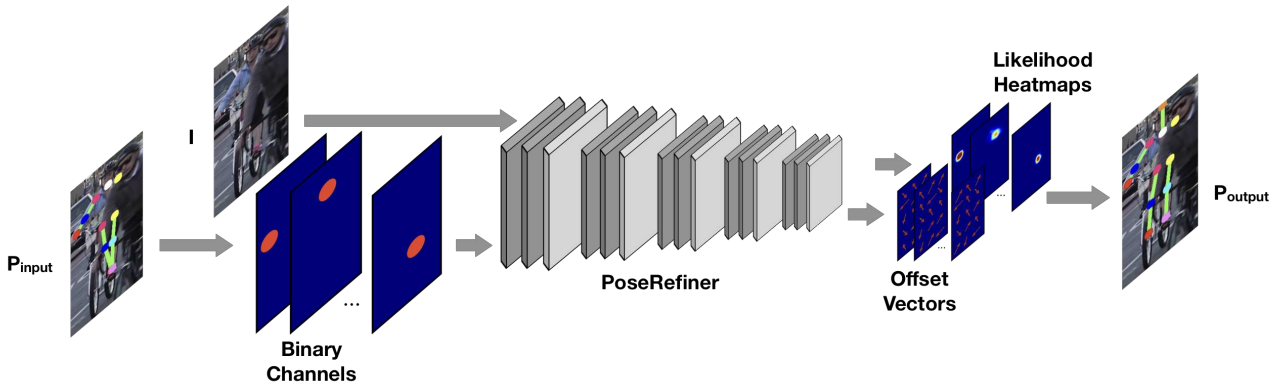


Figure 2: The overview of our `PoseRefiner` system. We take as input an image I and an initial estimate of a person body pose P_{input}. The input pose is encoded as n binary channels, where n is the number of joints, which are stacked together with the image RGB channels and used as an input to a fully convolutional network. The network learns to predict likelihood heatmaps for each joint type, as well as offset vectors to recover from the downscaled spatial resolution. The output pose P_{output} is a refined estimate of the initial input.

body joints per person, and then performs a lightweight optimization to link the predictions over time. [26] extends the work of [6] by rethinking the network architecture and developing a redundant part affinity fields (PAFs) mechanism, while [37] employs a geometric tracker to match the predicted poses frame-by-frame. All of these approaches heavily rely on accurate pose estimation in a single frame. We show in Section 4.4 that by refining the initial pose hypothesis in individual frames we are able to significantly improve pose tracking over time.

Refinement/error correction. Another group of work [7, 17, 5, 32, 28] aims to refine labels from the initial estimate by jointly reasoning about input-output space. [7] proposes to iteratively estimate residual corrections which are added to the initial prediction. In a similar spirit [17, 5] use RNN-like architectures to sequentially refine the results and [35, 9, 28] employ cascade CNNs with refinement stages. [13] decomposes the label improvement into three stages: first detecting the errors in the initial labels, replacing the incorrect labels with new ones and refining the labels by predicting residual corrections. Likewise [22] employs a parallel architecture that propagates correct labels to nearby pixels and replaces the erroneous predictions with refined ones, then fuses the intermediate results to obtain a final prediction. In contrast to these methods, our proposed refinement network is much simpler as it learns to directly predict the refined labels from the initial estimate using a simple feed-forward fully convolutional network.

3. Method

In this section, we describe our approach - `PoseRefiner` - in detail. We propose a pose refinement network which takes as input both an RGB image and a body pose estimate and aims to refine the initial prediction by jointly reasoning about the input and output space (see Figure 2). Exploiting the dependencies between the image and the predicted body pose makes it easier for the model to identify the errors in the initial estimate and how to refine them. For the network to be able to learn to correct the erroneous body joint predictions we employ a training data augmentation scheme, modeled to generate the most common failure cases of human pose estimators. This yields a model that is able to refine a human pose estimate derived from different pose estimation approaches and allows to achieve state-of-the-art results on the challenging MPII Human Pose and PoseTrack benchmarks.

Approach. We approach the refinement of pose estimation as a system on its own, which can be easily used as a post-processing step following any keypoint prediction task. Although there can be multiple estimated people poses in an image, we apply the refinement process on a per-person level. Given an estimated person pose, we initially rescale and crop around it to obtain a reference input. We then forward this obtained image I and pose estimate P_{input} as an input to a fully convolutional neural network f, modeled to compute a refined pose prediction P_{output}.

Formally, we refine an initial pose estimate P_{input} as: $P_{output} = f(I, P_{input})$, where:

- f is the **`PoseRefiner`**, the function to be learned. Since the output of this function is a single person pose, we model it using a fully convolutional network designed for single-person pose estimation.
- I is the original image in RGB format.
- P_{input} is the initial body pose, which needs to be refined. It is concatenated with the RGB image as n additional channels, where n is the number of body joints.

Figure 3: Examples of the proposed data synthesis for training. Starting from the ground truth P_{output}, we synthesize the initial pose estimate P_{input} to mimic the most common errors of pose estimators. For visualization purposes, we only illustrate one transformation at a time: **T1** shifts the left shoulder (yellow), **T2** switches the left ankle (pink) with the right ankle (red), **T3** replaces the right shoulder (fuchsia) by the left shoulder of the neighboring person and **T4** removes the right knee.

- P_{output} is the refined pose, in the form of n channels.

Both P_{input} and P_{output} are encoded using one binary channel for each body joint.

Architecture. We adopt the design choices that were shown successful in architectures with strong body joint detectors. As network architecture, we employ the ResNet-101 [20] backbone converted to a fully convolutional mode with stride of 8 px. Although the ResNet-101 network is designed to accept as input only 3 (RGB) channels, it can be easily extended to accept additional body joint channels by increasing the depth of the filters of the first convolutional layer (from 3 to $3 + n$), where n is the number of body joints.

Following [39, 24], we train the network to predict two types of output: likelihood heatmaps of each body joint and offsets from the locations on the heatmap grid to the ground truth joint locations. Likelihood heatmaps for each joint type are trained using sigmoid activations and cross entropy loss function. The shape of the output heatmaps is 8 times smaller in each spatial dimension than the shape of the input, due to the 8 pixel stride of the network. Hence to recover from the lost resolution, we learn to predict offset vectors from every heatmap location to the ground truth joint coordinate by regressing displacements $(\Delta x, \Delta y)$ using mean squared error.

At test time, every pixel location in each likelihood heatmap indicates the probability of presence of the particular joint at that coordinate. The pixel with the highest confidence in each likelihood heatmap is selected as the rough downscaled joint coordinate. The final coordinate is obtained by adding the offset vector $(\Delta x, \Delta y)$ to the upscaled joint location predicted at the lower resolution.

Training Data Synthesis. To train the network f, we need to have access to ground truth triplets $(I, P_{input}, P_{output})$. While (I, P_{output}) pairs are already available in large scale pose estimation datasets, we propose to synthesize P_{input} to mimic the most common failure cases of human pose estimators. The goal is for the model to be able to refine initial estimates and become robust to the "hard" body pose cases. In essence, P_{input} is a noisy version of P_{output}, which we synthesize from the ground truth by applying the following transformations (visualized in Figure 3):

(T1) Shift the coordinates of each joint by a displacement vector. The angle of the displacement vector is sampled uniformly from $[0, 2\pi]$. The length of the displacement vector is sampled with 90% chance from $[0px, 25px]$ and 10% chance from $[25px, 125px]$ to ensure both small and large displacements. In this way, the model is able to learn to do local refinements as well as to handle larger offsets of joints in spurious body configurations.

(T2) Switch symmetric joints of the same person (e.g. replace left shoulder by right shoulder) - with probability 10% per pair of joints. Such type of noise is a usual failure case of pose estimators, which occasionally confuse similarly looking left and right limbs or whether the joints are faced from the front or from the back.

(T3) Replace joints of a person by joints of the same/symmetric type of neighboring persons (e.g left hip of person A is replaced by the neighboring left/right hip of person B). Such synthesis is possible only when the pose estimation dataset contains multiple annotated people in the same

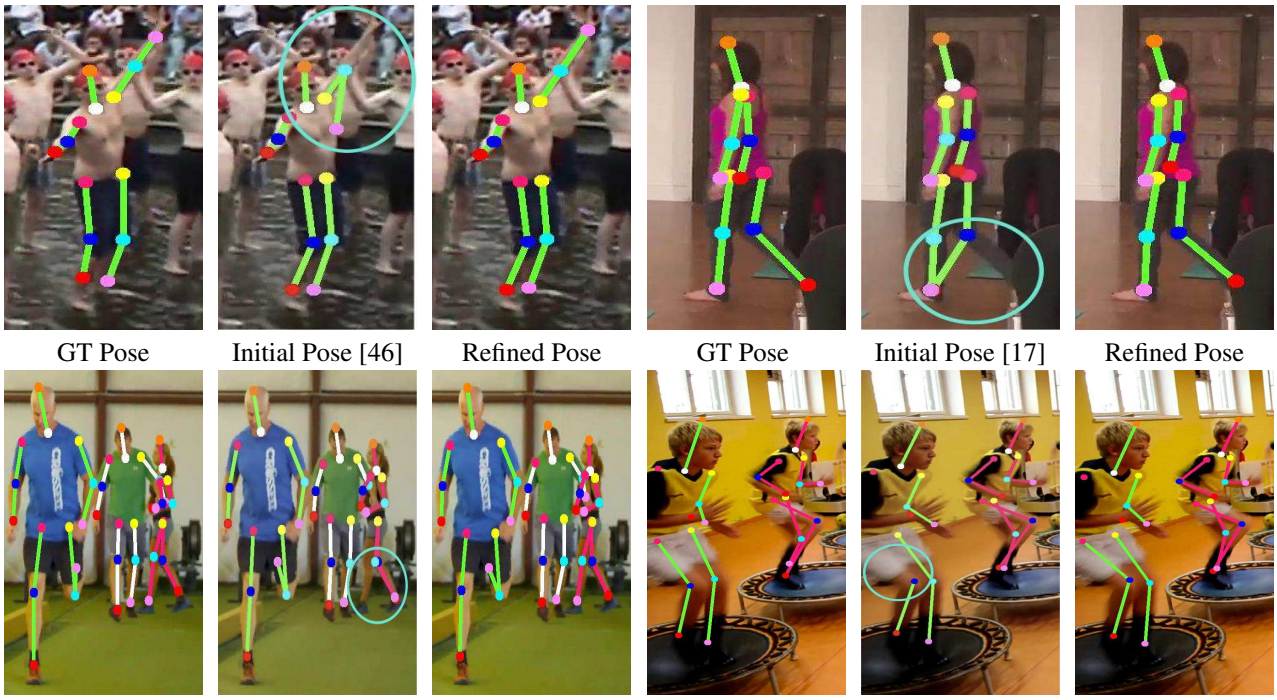


Figure 4: Qualitative results on the MPII Single-Pose dataset (**top**) and MPII Multi-Pose dataset (**bottom**). The blue circles denote the areas where the `PoseRefiner` brings significant improvement. Our refinement method provides better localization for the challenging keypoint extremities (ankles and wrists), can remove confusions between symmetrical joint types (right ankle in **top right** and left ankle in **bottom left** figures) and can recover spurious joints (left wrist in **top left** figures) or missing joints (right hip in **bottom right** figures) by reasoning about the pose structure of the target person.

image. If such a neighbor joint exists in a $75px$ vicinity, replacement is done with 30% probability. This transformation models the pose estimation errors arising in crowded scenes, when limbs of different people are merged together.

(T4) Remove body joint with 30% chance. This transformation helps to simulate the common missing joint error of body pose estimators, which is usually introduced by thresholding of low-confident keypoint detections.

Implementation Details. We implement our system using the publicly available TensorFlow [1] framework.

Following [24], we rescale the input pose and image such that the reference height of a person is 340 px. The height of a person is estimated either from the scale of the ground truth head bounding box (if available, as in the MPII Single Person Dataset), or directly from the estimated input pose. We also crop 250 px in each direction around the bounding box of the input pose. This should standardize the input and minimize the searching space of the joints, while providing enough context to the `PoseRefiner`.

The input body pose estimate P_{input} is encoded using one binary channel for each body joint. Each channel contains a circular blob of radius 15 px around the joint coordinate. If there is no coordinate for the particular joint, the respective channel will be the null matrix. This encoding follows the encoding of the P_{output} channels during training, which has been shown to work well for training strong body part detectors [39, 24].

Our training procedure contains a data augmentation step, which we employ for generating more training data. We apply random rescaling $\pm 30\%$ and random flipping around the vertical axis.

When no pre-training is applied, we initialize the network with the weights of models trained on ImageNet [11]. For initialization of the extra convolutional filters corresponding to additional channels of P_{input}, we reuse the weights corresponding to RGB channels of I.

Optimization is done using stochastic gradient descent with 1 image per batch, starting with learning rate $lr = 0.005$ for one third of an epoch and continuing with $lr = 0.02$ for 15 epochs, $lr = 0.002$ for 10 epochs and $lr = 0.001$ for 10 other epochs. Training on the MPII dataset ($\approx 29k$ people) runs for 40 hours on one GPU [1].

4. Results

We now evaluate the proposed approach on the tasks of articulated single- and multi-person pose estimation, and articulated pose tracking.

[1] We use NVIDIA Tesla V100 GPU with 16 GB RAM

4.1. Experimental Setup

We test our refinement network on three tasks involving human body pose estimation: single-person pose estimation, multi-person pose estimation and multi-person articulated tracking. In each of these tasks, we refine the predictions of several state-of-the-art methods by post-processing each initially estimated body pose using the `PoseRefiner`.

We experiment on four public challenges: MPII Human Pose [3] ("Single-Person" and "Multi-Person") and PoseTrack [2] ("Single-Frame Multi-Person Pose Estimation" and "Multi-Person Articulated Tracking").

Datasets. For fair comparison with the methods whose prediction we refine, we follow the most common practices in choosing the datasets for training the `PoseRefiner`.

For the MPII Human Pose challenges, we train on MPII Human Pose only, whose training set contains $\approx 29k$ poses. For evaluation, we report results on the test set of MPII Human Pose, which includes $7,247$ sufficiently separated poses used in the "Single-Person" challenge, as well as $4,485$ poses organized in groups of interacting people, used in the evaluation of the "Multi-Person" challenge. Although the protocol of the MPII Multi-Person task assumes that the location and the rough scale of each group of people is provided during test time, the `PoseRefiner` does not require any of this information.

In the case of the PoseTrack challenges, we pretrain on the COCO [31] train2017 set ($\approx 150k$ poses), then fine-tune on the MPII training set and afterwards on the PoseTrack training set. Pretraining is needed as the PoseTrack training set contains $\approx 61k$ poses, but only $2,437$ different identities, which do not cover a very high appearance variability. Since the set of joint types differs across datasets (MPII annotates 16 keypoints, PoseTrack annotates 15 and COCO annotates 17), we use the PoseTrack set of body joints as reference and map all the other types to their closest PoseTrack joint type. The COCO dataset does not provide annotations for *top-head* and *bottom-head*, so we heuristically use the top most semantic segmentation vertex as the *top-head* keypoint, and the midpoint between the *nose* and the midpoint of the *shoulders* as the *bottom-head* keypoint. Similarly, MPII does not provide annotations for the *nose* joint, so we use the midpoint between the *bottom-head* and *top-head* as a replacement. For evaluation, we report results on the PoseTrack validation set (50 videos, containing $18,996$ poses), which is publicly available.

Evaluation metrics. For each task, we adopt the evaluation protocol proposed by the respective challenges.

On the MPII Human Pose (Single-Person) dataset, we report the Percentage of Correct Keypoints metric calculated with the matching threshold of half the length of the head segment (PCK_h@0.5), averaged across all joint types. We also report the Area Under the Curve measure (AUC), corresponding to the curve generated by PCK_h measured over a range of percentages of the length of the head segment. Since none of the metrics is sensitive to false-positive joint detections, we do not remove non-confident predicted keypoints by thresholding them.

On the MPII Human Pose (Multi-Person) dataset, we report the mean Average Precision (mAP) based on the matching of body poses using PCK_h@0.5, following the evaluation kit of [39]. This metric requires providing a confidence score for each detected body joint in addition to its location. Since the confidence scores provided by the `PoseRefiner` are in fact conditional probabilities of detection, we use the initial confidence scores (before refinement) that the input pose predictions come with.

On the Single-Frame Multi-Person Pose Estimation task of PoseTrack, we report the same mAP metric as in MPII Human Pose, with the slight difference that the rough scale and location of people are not provided during test time, so the mAP evaluation in PoseTrack does not require it.

On the Multi-Person Articulated Tracking task of PoseTrack, we calculate the Multiple Object Tracking Accuracy (MOTA) for each joint, and report mMOTA, averaged across all joints. This metric requires providing a track ID for each detected body pose, but no confidence score for joint detections. Since the metric is sensitive to false positive keypoints, we threshold the low confidence joints with the aim of removing incorrect predictions. We experimentally find that removing all joint detections with confidence scores less than $\tau = 0.7$ provides the best trade-off between the number of missed joints and the number of false positive joints, both penalized in the calculation of MOTA.

4.2. Single-Person Pose Estimation

The effect that the `PoseRefiner` has on the test set of MPII Single-Person is shown in Table 1. We refine various single person pose estimates given by different methods [24, 17, 7, 8], including the state of the art [46].

One can notice that the performance of [46, 8] is already quite high, which motivates using less noise in the synthesis of the input pose for training the refinement model for these methods. Hence, we decrease the level of noise used in the generation of input poses during training by switching off all noise transformations, with the exception of (T1). We do not find it necessary to change the original set of noise transformations in any other experiment.

Using `PoseRefiner` as a post-processing step consistently increases the performance of each method, with the average improvement of $mPCK_h$@0.5 and AUC ranging from 0.3 to 6.8, while hurting neither metric. This shows the generality and effectiveness of our refinement method on single-person pose estimation, which already hints at its use in other more complex tasks involving keypoint detec-

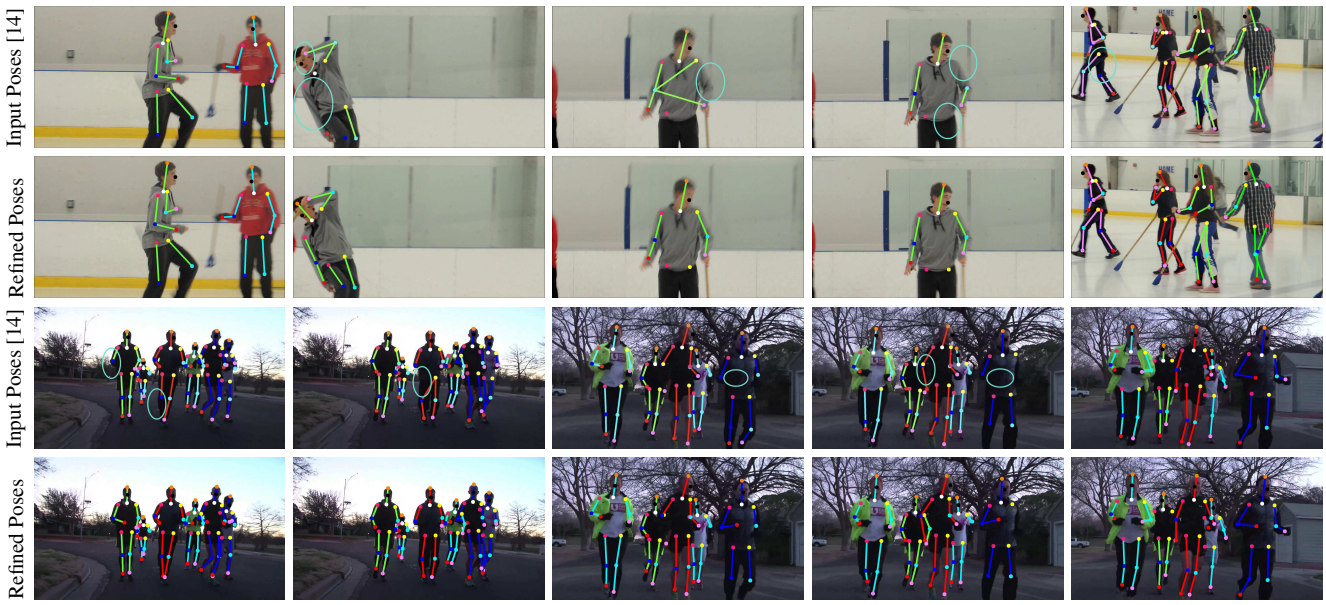


Figure 5: Qualitative results on the PoseTrack validation set, before and after applying the proposed refinement. The blue circles denote the areas where the proposed post-processing step brings significant improvement. The `PoseRefiner` recovers missing joints (e.g. right elbow and right hip in **top seq. - fr.2**, right wrist in **bottom seq. - fr.3**) and helps with confusions of symmetrical joints (left elbow in **top seq. - fr.3**, right hip in **bottom seq. - fr.2**).

Method	$mPCK_h$@0.5	AUC	Δ
Pyramid Residual Module [46]	**92.0**	64.2	-
+ Refinement [2]	**92.0**	64.7	+0.3
Adversarial PoseNet [8]	91.9	61.6	-
+ Refinement [2]	**92.1**	63.6	+1.1
DeeperCut [24]	88.5	60.8	-
+ Refinement	**89.1**	62.3	+1.0
Chained Predictions [17]	86.1	57.3	-
+ Refinement	**88.0**	61.2	+2.9
Iterative Error Feedback [7]	81.3	49.1	-
+ Refinement	**85.6**	58.4	+6.8

Table 1: Effect of the proposed refinement over different pose estimation methods on the MPII Single-Person [3] test set. Δ indicates the average improvement of mPCKh@0.5 and AUC after applying the pose refinement model.

tion, such as multi-person pose estimation in images and multi-person articulated tracking.

We present the qualitative results on MPII Single Pose in Figure 4. Our refinement network is able to correct confusion between different joint types, recover from spurious or missing keypoints and provide better overall localization of joints.

[2]Using a refinement model trained with only (T1) transformations.

Method	mAP	ΔmAP
Associative Embedding [33]	77.5	-
+ Refinement	**78.0**	+0.5
Part Affinity Fields [6]	75.6	-
+ Refinement	**76.9**	+1.3
ArtTrack [23]	74.2	-
+ Refinement	**75.1**	+0.9
Varadarajan et al., arXiv'17 [44]	72.2	-
+ Refinement	**75.1**	+2.9

Table 2: Effect of the proposed refinement over different methods on the MPII Multi-Person [3] test set. ΔmAP indicates the improvement of mAP after applying the proposed pose refinement network.

4.3. Multi-Person Pose Estimation

Since the output of a multi-person pose estimator is a set of body poses in an image, we can use the `PoseRefiner` to perform error correction on each estimated pose, independently of the others.

Table 2 shows the quantitative effect that the refinement post processing step has on several methods [33, 6, 23, 44] applied on the MPII Multi-Person test set. It proves to help the overall performance of each system, including the best performing method [33] on this dataset, setting a new state of the art of 78.0 mAP. Given that our system does not have any influence over non detected people/human body poses, the overall improvement (ranging from 0.5 mAP to

Method	mAP	ΔmAP
ML_Lab [48]	71.9	-
+ Refinement	**73.8**	+1.9
ArtTrack [23] (best mAP) [3]	68.6	-
+ Refinement (w/o nose)	**70.0**	+1.4
+ Refinement (with nose)	69.7	+1.1
BUTD [26] (best mAP)	67.8	-
+ Refinement	**70.9**	+3.1
Detect-and-Track [14]	60.4	-
+ Refinement	**65.7**	+5.3

Table 3: Effect of the proposed refinement on the PoseTrack [2] validation set, the Single-Frame Multi-Person Pose Estimation challenge. ΔmAP indicates the improvement of mAP after applying the pose refinement model.

Method	mAP	mMOTA	ΔmMOTA
BUTD [26] (best mMOTA)	62.5	56.0	-
+ Refinement	64.3	**58.4**	+2.4
Detect-and-Track [14]	60.4	55.1	-
+ Refinement	64.1	**57.3**	+2.2
ArtTrack [23] (best mMOTA) [4]	66.7	50.2	-
+ Refinement (w/o nose)	66.5	53.3	+3.1
+ Refinement (with nose)	67.0	**54.1**	+3.9
ML_Lab [48]	71.9	48.6	-
+ Refinement	70.1	**53.5**	+4.9

Table 4: Effect of the proposed refinement on the PoseTrack [2] validation set, the Multi-Person Articulated Tracking challenge. ΔmMOTA indicates the improvement of mMOTA after applying the pose refinement model.

2.9 mAP) can be considered significant for the localization of joints.

Table 3 shows the results on the PoseTrack validation set. We refine the pose predictions of methods proposed for the Single-Frame Multi-Person Pose Estimation case [48, 23, 26, 14]. They process images independently of each other and optimize the mAP metric. We again observe consistent improvements when employing the `PoseRefiner`, managing to increase the best reported performance on the dataset from 71.9 mAP [48] to 73.8 mAP.

In the case of ArtTrack [23], which does not output a *nose* joint, we remove the missing keypoint from the ground truth and from the evaluation procedure and report the obtained result (68.6 mAP) on the remaining subset of joints (64.0 mAP evaluated on all joints). After post processing with the `PoseRefiner`, the *nose* joint is recovered, and we report results for both evaluations: when removing the nose joint from the evaluation procedure (70.0 mAP) and when counting it into the evaluation (69.7 mAP). The fact that the difference between the two is small shows that the new *nose* joint is recovered and nearly as well localized as the other joints. In addition, the overall performance after the refinement step is increasing (68.6 → 69.7 mAP).

4.4. Multi-Person Articulated Tracking

Multi-Person Articulated Tracking involves detecting all people in each frame of a video, estimating their pose and linking their identities over time. We can therefore apply the `PoseRefiner` on each estimated pose independently of the others, while keeping the original identities of the detected people. Table 4 shows the quantitative effect of the proposed refinement step on the PoseTrack validation set. Note that there are cases in which the results of the same method differ in Table 3 from Table 4, depending on which metric the method optimizes. Although the refinement only updates the coordinates of already detected body poses and no tracklet IDs are changed, the overall mMOTA improvement obtained by our system is significant (from 2.2 to 4.9 mMOTA). We show systematic improvement on every tracking result we refine, including the predictions of the method with the highest performance [26]. The state of the art is hence extended, reaching 58.4 mMOTA on this benchmark. Similar to the Multi-Person Pose Estimation case, we recover the missing *nose* joint on ArtTrack and manage to refine its overall tracking results by 3.9 mMOTA. Qualitative results of multi-person articulated tracking are presented in Figure 5.

5. Conclusion

In this work we proposed a human pose refinement network which can be applied over a body pose estimate derived from any human pose estimation approach. In comparison to other refinement techniques, our approach provides a simpler solution by directly generating the refined body pose from the initial pose prediction in one forward pass, exploiting the dependencies between the input and output spaces. We report consistent improvement of our model applied over state-of-the-art methods across different datasets and tasks, highlighting its effectiveness and generality. Our experiments show that even top performing methods can benefit from the proposed refinement step. With our refinement network we improve the best reported results on MPII Human Pose and PoseTrack datasets for multi-person pose estimation and pose tracking tasks.

[3] ArtTrack does not output a *nose* joint, so the evaluation before refinement is performed without considering this joint. Our refinement network can recover the missing nose joint, leading to better performance (68.6 → 69.7 mAP).

[4] Although ArtTrack does not output a *nose* joint, our refinement network can recover the missing nose joint, while improving overall performance (50.2 → 54.1 mMOTA).

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] M. Andriluka, U. Iqbal, A. Milan, E. Insafutdinov, L. Pishchulin, J. Gall, and B. Schiele. Posetrack: A benchmark for human pose estimation and tracking. *CVPR*, 2018.

[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[4] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009.

[5] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In *Automatic Face and Gesture Recognition*, 2017.

[6] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[7] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *CVPR*, 2016.

[8] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. *CoRR, abs/1705.00389*, 2, 2017.

[9] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. *arXiv preprint arXiv:1711.07319*, 2017.

[10] M. Dantone, J. Gall, C. Leistner, and L. V. Gool. Human pose estimation using body parts dependent joint regressors. In *CVPR*, 2013.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[12] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.

[13] S. Gidaris and N. Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. 2017.

[14] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran. Detect-and-track: Efficient pose estimation in videos. *arXiv preprint arXiv:1712.09184*, 2017.

[15] G. Gkioxari, P. A. Arbeláez, L. D. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *CVPR*, 2013.

[16] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. Using k-poselets for detecting people and localizing their keypoints. In *CVPR*, 2014.

[17] G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, 2016.

[18] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] S. Huang, M. Gong, and D. Tao. A coarse-fine network for keypoint localization. In *ICCV*, 2017.

[22] Y.-H. Huang, X. Jia, S. Georgoulis, T. Tuytelaars, and L. V. Gool. Error correction for dense semantic image labeling. *arXiv preprint arXiv:1712.03812*, 2017.

[23] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. Arttrack: Articulated multi-person tracking in the wild. In *CVPR*, 2017.

[24] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[25] U. Iqbal, A. Milan, and J. Gall. Posetrack: Joint multi-person pose estimation and tracking. In *CVPR*, 2017.

[26] S. Jin, X. Ma, Z. Han, Y. Wu, W. Yang, W. Liu, C. Qian, and W. Ouyang. Towards multi-person pose tracking: Bottom-up and top-down methods. In *ICCV PoseTrack Workshop*, 2017.

[27] L. Ladicky, P. H. S. Torr, and A. Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. 2013.

[28] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. 2016.

[29] X. Liang, K. Gong, X. Shen, and L. Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *arXiv preprint arXiv:1804.01984*, 2018.

[30] I. Lifshitz, E. Fetaya, and S. Ullman. Human pose estimation using deep consensus voting. In *ECCV*, 2016.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[32] N. Neverova and I. Kokkinos. Mass displacement networks. *arXiv preprint arXiv:1708.03816*, 2017.

[33] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, 2017.

[34] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[35] J. Pang, W. Sun, J. S. J. Ren, C. Yang, and Q. Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. 2017.

[36] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, 2017.

[37] C. Payer, T. Neff, H. Bischof, M. Urschler, and D. Stern. Simultaneous multi-person detection and single-person pose estimation with a single heatmap regression network. In *ICCV PoseTrack Workshop*, 2017.

[38] L. Pishchulin, M. Andriluka, P. V. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, 2013.

[39] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016.

[40] B. Sapp, C. T. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *CVPR*, 2010.

[41] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *CVPR*, 2017.

[42] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.

[43] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[44] S. Varadarajan, P. Datta, and O. Tickoo. A greedy part assignment algorithm for real-time multi-person 2d pose estimation. *arXiv preprint arXiv:1708.09182*, 2017.

[45] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[46] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang. Learning feature pyramids for human pose estimation. In *ICCV*, 2017.

[47] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[48] X. Zhu, Y. Jiang, and Z. Luo. Multi-person pose estimation for posetrack with enhanced part affinity fields. In *ICCV PoseTrack Workshop*, 2017.