Test-Time Personalization with a Transformer for Human Pose Estimation

Miao Hao*
UC San Diego
mhao@ucsd.edu

Yizhuo Li*
Shanghai Jiaotong University
liyizhuo@sjtu.edu.cn

Zonglin Di*
UC San Diego
zodi@ucsd.edu

Nitesh B. Gundavarapu
UC San Diego
nbgundav@ucsd.edu

Xiaolong Wang
UC San Diego
xiw012@ucsd.edu

Abstract

We propose to personalize a 2D human pose estimator given a set of test images of a person without using any manual annotations. While there is a significant advancement in human pose estimation, it is still very challenging for a model to generalize to different unknown environments and unseen persons. Instead of using a fixed model for every test case, we adapt our pose estimator during test time to exploit person-specific information. We first train our model on diverse data with both a supervised and a self-supervised pose estimation objectives jointly. We use a Transformer model to build a transformation between the self-supervised keypoints and the supervised keypoints. During test time, we personalize and adapt our model by fine-tuning with the self-supervised objective. The pose is then improved by transforming the updated self-supervised keypoints. We experiment with multiple datasets and show significant improvements on pose estimations with our self-supervised personalization. Project page with code is available at https://liyz15.github.io/TTP/.

1 Introduction

Recent years have witnessed a large advancement in human pose estimation. A lot of efforts have been spent on learning a generic deep network on large-scale human pose datasets to handle diverse appearance changes [59, 64, 7, 15, 43]. Instead of learning a generic model, another line of research is to personalize and customize human pose estimation for a single subject [10]. For a specific person, we can usually have a long video (e.g., instructional videos, news videos) or multiple photos from personal devices. With these data, we can adapt the model to capture the person-specific features for improving pose estimation and handling occlusion and unusual poses. However, the cost of labeling large-scale data for just one person is high and unrealistic.

In this paper, we propose to personalize human pose estimation with unlabeled video data during test time, namely, *Test-Time Personalization*. Our setting falls in the general paradigm of Test-Time Adaptation [58, 35, 61, 69], where a generic model is first trained with diverse data, and then it is fine-tuned to adapt to a specific instance during test time without using human supervision. This allows the model to generalize to out-of-distribution data and preserves privacy when training is distributed. Specifically, Sun et al. [58] propose to generalize image classification by performing joint training with a semantic classification task and a self-supervised image rotation prediction task [18]. During inference, the shared network representation is fine-tuned on the test instance with the self-supervisory signal for adaptation. While the empirical result is encouraging, it is unclear how

*Equal contribution

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

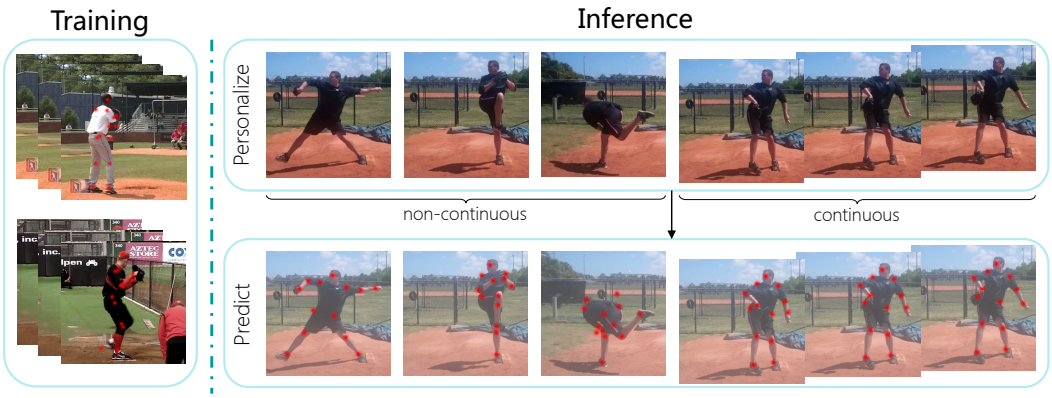

Figure 1: **Test-Time Personalization**. Our model is firstly trained on diverse data with both supervised and self-supervised keypoint estimation tasks. During test time, we personalize the model using only the self-supervised target in single person domain and then predict with the updated model. During Test-Time Personalization, no continuous data is required but only unlabeled samples belonging to the same person are needed. Our method boosts performance at test time without costly labeling or sacrificing privacy.

the rotation prediction task can help image classification, and what is the relation between two tasks besides sharing the same feature backbone.

Going beyond feature sharing with two distinct tasks, we introduce to perform joint supervised and self-supervised human keypoint estimation [26] tasks where the supervised keypoint outputs are directly transformed from the self-supervised keypoints using a Transformer [60]. In this way, when fine-tuning with the self-supervised task in test time, the supervised pose estimation can be improved by transforming from the improved self-supervised keypoints.

We adapt the self-supervised keypoint estimation task proposed by Jakab et al. [26]. The task is built on the assumption that the human usually maintains the appearance but changes poses across time in a video. Given a video frame, it trains a network to extract a tight bottleneck in the form of sparse spatial heatmaps, which only contain pose information without appearance. The training objective is to reconstruct the same frame by combining the bottleneck heatmaps and the appearance feature extracted from another frame. Note while this framework can extract keypoints to represent the human structure, they are not aligned with the semantic keypoints defined in human pose estimation. Building on this model, we add an extra keypoint estimation objective which is trained with human supervision. Instead of simply sharing features between two objectives as [58], we train a Transformer model on top of the feature backbone to extract the relation and affinity matrix between the self-supervised keypoint heatmap and the supervised keypoint heatmap. We then use the affinity matrix to transform the self-supervised keypoints as the supervised keypoint outputs. With our Transformer design, it not only increases the correlation between two tasks when training but also improves Test-Time Personalization as changing one output will directly contribute to the the output of another task.

We perform our experiments with multiple human pose estimation datasets including Human 3.6M [24], Penn Action [71], and BBC Pose [8] datasets. As shown in Figure 1, our Test-Time Personalization can perform on frames that continuously exist in a video and also with frames that are non-continuous as long as they are for the same person. We show that by using our approach for personalizing human pose estimation in test time, we achieve significant improvements over baselines in all datasets. More interestingly, the performance of our method improves with more video frames appearing online for the same person during test time.

2 Related Work

Human Pose Estimation. Human pose estimation has been extensively studied and achieved great advancements in the past few years [59, 64, 7, 15, 43, 67, 45, 21, 65, 13, 57, 75, 44, 5, 14]. For example, Toshev et al. [59] propose to regress the keypoint locations from the input images. Instead of direct location regression, Wei et al. [64] propose to apply a cascade framework for coarse to fine heatmap prediction and achieve significant improvement. Building on this line of research, Xiao et al. [65] provides a simple and good practice on heatmap-based pose estimation, which is utilized as

our baseline model. While in our experiments we utilize video data for training, our model is a single-image pose estimator and it is fundamentally different from video pose estimation models [1, 19, 62] which take multiple continuous frames as inputs. This gives our model the flexibility to perform pose estimation on static images and thus it is not directly comparable to approaches with video inputs. Our work is also related to personalization on human pose estimation from Charles et al. [10], which uses multiple temporal and continuity constraints to propagate the keypoints to generate more training data. Instead of tracking keypoints, we use a self-supervised objective to perform personalization in test time. Our method is not restricted to the continuity between close frames, and the self-supervision can be applied on any two frames far away in a video as long as they belong to the same person.

Test-Time Adaptation. Our personalization setting falls into the paradigm of Test-Time Adaptation which is recently proposed in [51, 50, 3, 58, 35, 61, 69, 28, 42, 20] for generalization to out-of-distribution test data. For example, Shocher et al. [51] propose a super-resolution framework which is only trained during test time with a single image via down-scaling the image to create training pairs. Wang et al. [61] introduce to use entropy of the classification probability distribution to provide fine-tuning signals when given a test image. Instead of optimizing the main task itself during test time, Sun et al. [58] propose to utilize a self-supervised rotation prediction task to help improve the visual representation during inference, which indirectly improves semantic classification. Going beyond image classification, Joo et al. [30] propose a method that proposes test time optimization for 3D human body regression. In our work for pose personalization, we try to bridge the self-supervised and supervised objectives close. We leverage a self-supervised keypoint estimation task and transform the self-supervised keypoints to supervised keypoints via a Transformer model. In this way, training with self-supervision will directly improve the supervised keypoint outputs.

Self-supervised Keypoint Estimation. There are a lot of recent developments on learning keypoint representations with self-supervision [55, 72, 26, 38, 32, 27, 68, 36, 40]. For example, Jakab et al. [26] propose a video frame reconstruction task which disentangles the appearance feature and keypoint structure in the bottleneck. This work is then extended for control and Reinforcement Learning [32, 36, 40], and the keypoints can be mapped to manual defined human pose via adding adversarial learning loss [27]. While the results are encouraging, most of the results are reported in relatively simple scenes and environments. In our paper, by leveraging the self-supervised task together with the supervised task, we can perform human pose personalization on images in the wild.

Transformers. Transformer has been widely applied in both language processing [60, 16] and computer vision tasks [63, 46, 23, 49, 56, 17, 11, 4, 73, 6, 37], specifically for pose estimation recently [66, 54, 41, 33]. For example, Li et al. [33] propose to utilize the encoder-decoder model in Transformers to perform keypoint regression, which allows for more general-purpose applications and requires less priors in architecture design. Inspired by these works, we apply Transformer to reason the relation and mapping between the supervised and self-supervised keypoints.

3 Method

Our method aims at generalizing better for pose estimation on a single image by personalizing with unlabeled data. The model is firstly trained with diverse data on both a supervised pose estimation task and a self-supervised keypoint estimation task, using a proposed Transformer design to model the relation between two tasks. During inference, the model conducts Test-Time Personalization which only requires the self-supervised keypoint estimation task, boosting performance without costly labeling or sacrificing privacy. The whole pipeline is shown in Figure 2.

3.1 Joint Training for Pose Estimation with a Transformer

Given a set of N labeled images of a single person $\mathbf{I} = \{I_1, I_2 \ldots, I_N\}$, a shared encoder ϕ maps them into feature space $\mathbf{F} = \{F_1, F_2 \ldots, F_N\}$, which is shared by both a supervised and a self-supervised keypoint estimation tasks. We introduce both tasks and the joint framework as follows.

3.1.1 Self-supervised Keypoint Estimation

For the self-supervised task, we build upon the work from Jakab et al. [26] which uses an image reconstruction task to perform disentanglement of human structure and appearance, which leads to self-supervised keypoints as intermediate results. Given two images of a single person I_s and I_t, the task aims at reconstructing I_t using structural keypoint information from target I_t and appearance information from source I_s. The appearance information F_s^{app} of source image I_s is extracted with a

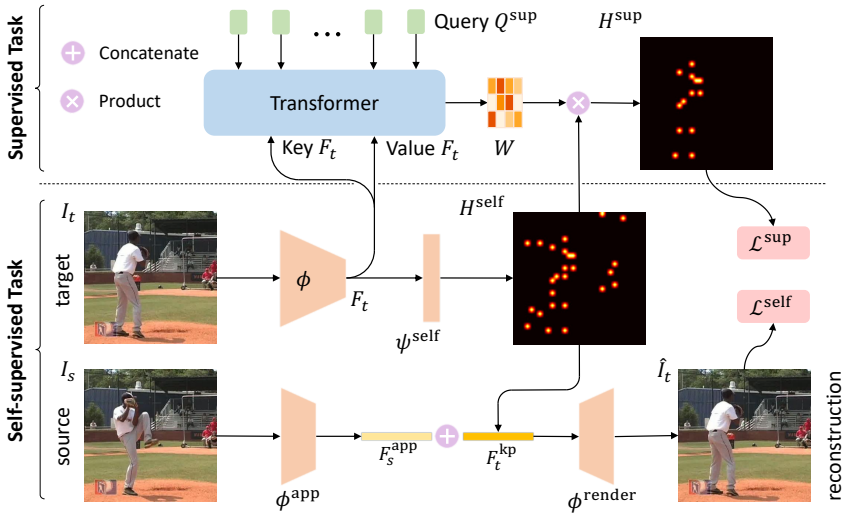

Figure 2: The proposed pipeline. 1) **Self-supervised task for personalization.** In the middle stream, the encoder ϕ encodes the target image into feature F_t. Then F_t is fed into the self-supervised head ψ^{self} obtaining self-supervised keypoint heatmaps H^{self}. Passing H^{self} into a keypoint encoder (skipped in the figure) leads to keypoint feature F_t^{kp}. In the bottom stream, a source image is forwarded to an appearance extractor ϕ^{app} which leads to appearance feature F_t^{app}. Together, a decoder reconstructs the target image using concatenated F_s^{app} and F_t^{kp}. 2) **Supervised task with Transformer.** On the top stream, a Transformer predicts an affinity matrix given learnable keypoint queries Q^{sup} and F_t. The final supervised heatmaps H^{sup} is given as weighted sum of H^{self} using W.

simple extractor ϕ^{app} (see the bottom stream in Figure 2). The extraction of keypoints information from the target image follows three steps as below (also the see the middle stream in Figure 2).

Firstly, the target image I_t is forwarded to the encoder ϕ to obtain shared feature F_t. The self-supervised head ψ^{self} further encodes the shared feature F_t into heatmaps H_t^{self}. Note the number of channels in the heatmap H_t^{self} is equal to the number of self-supervised keypoints. Secondly, H_t^{self} is normalized using a $\mathrm{Softmax}$ function and thus becomes condensed keypoints. In the third step, the heatmaps are replaced with fixed Gaussian distribution centered at condensed points, which serves as keypoint information F_t^{kp}. These three steps ensure a bottleneck of keypoint information, ensuring there is not enough capacity to encode appearance features to avoid trivial solutions.

The objective of the self-supervised task is to reconstruct the target image with a decoder using both appearance and keypoint features: $\hat{I}_t = \phi^{\text{render}}\left(F_s^{\text{app}}, F_t^{\text{kp}}\right)$. Since the bottleneck structure from the target stream limits the information to be passed in the form of keypoints, the image reconstruction enforces the disentanglement and the network has to borrow appearance information from source stream. The Perceptual loss [29] and L2 distance are utilized as the reconstruction objective,

$$\mathcal{L}^{\text{self}} = \text{PerceptualLoss}\left(I_t, \hat{I}_t\right) + \left\|I_t - \hat{I}_t\right\|^2 \tag{1}$$

Instead of self-supervised tasks like image rotation prediction [18] or colorization [70], choosing an explicitly related self-supervised key-point task in joint training naturally preserves or even improves performance, and it is more beneficial to test-time personalization. Attention should be paid that our method requires only label of one single image and unlabeled samples belonging to the same person. Compared to multiple labeled samples of the same person or even more costly consecutively labeled video, acquiring such data is much more easier and efficient.

3.1.2 Supervised Keypoint Estimation with a Transformer

A natural and basic choice for supervised keypoint estimation is to use an unshared supervised head ψ^{sup} to predict supervised keypoints based on F_t. However, despite the effectiveness of multi-task learning on two pose estimation tasks, their relation still stays plain on the surface. As similar tasks do not necessarily help each other even when sharing features, we propose to use a Transformer decoder to further strengthen their coupling. The Transformer decoder models the relation between

two tasks by learning an affinity matrix between the supervised and the self-supervised keypoint heatmaps.

Given the target image I_t, its feature F_t and self-supervised heatmap $H_t^{\text{self}} \in \mathbb{R}^{h \times w \times k^{\text{self}}}$ are extracted using encoder ϕ and self-supervised head ψ^{self} respectively, where h, w, k^{self} are the height, width and number of keypoints of the heatmap. The Transformer module learns the affinity matrix based on learnable supervised keypoint queries $Q^{\text{sup}} \in \mathbb{R}^{k^{\text{sup}} \times c}$ and context feature F_t.

A standard transformer decoder layer consists of a multi-head attention layer and a feed-forward network. The spatial feature F_t is flattened to n tokens such that $F_t \in \mathbb{R}^{n \times c}$. In a single-head attention layer,

$$Q = Q^{\text{sup}} T^Q,\ K = F_t T^K,\ V = F_t T^V \tag{2}$$

where $T^Q, T^K, T^V \in \mathbb{R}^{c \times c}$ are weight matrices. We use Q^{sup} as the query input and the network feature F_t as the key and value inputs. The attention weights A and attention results attn is given by,

$$A = \mathrm{Softmax}\left(QK^\top\right) \tag{3}$$

$$\mathrm{attn}\left(Q^{\text{sup}}, F_t, F_t\right) = AV \tag{4}$$

In multi-head attention $\mathrm{MHA}()$, Q^{sup} and F_t is split to $Q_1^{\text{sup}}, \ldots, Q_M^{\text{sup}}$ and $F_{(t,1)}, \ldots, F_{(t,M)}$, where M is the number of heads and every part is split to dimension $c' = c/M$,

$$\tilde{Q}^{\text{sup}} = \left[\mathrm{attn}_1(Q_1^{\text{sup}}, F_{(t,1)}, F_{(t,1)}); \ldots; \mathrm{attn}_M(Q_M^{\text{sup}}, F_{(t,M)}, F_{(t,M)})\right] \tag{5}$$

$$\mathrm{MHA}\left(Q^{\text{sup}}, F_t, F_t\right) = \mathrm{LayerNorm}\left(Q^{\text{sup}} + \mathrm{Dropout}\left(\tilde{Q}L\right)\right) \tag{6}$$

where $\mathrm{LayerNorm}$ is layer normalization [2], $\mathrm{Dropout}$ is dropout operation [53] and $L \in \mathbb{R}^{c \times c}$ is a projection. Passing the result to a feed-forward network which is effectively a two layer linear projection with ReLU activation followed also by residual connection, $\mathrm{Dropout}$ and $\mathrm{LayerNorm}$ completes the Transformer decoder layer. Stacking multiple layers gives us the affinity feature $F^{\text{aff}} \in \mathbb{R}^{k^{\text{sup}} \times c}$. Then F^{aff} is linearly projected to the space of supervised keypoints by weight matrix P and transformed using $\mathrm{Softmax}$ function among self-supervised keypoints into affinity matrix,

$$W = \mathrm{Softmax}\left(F^{\text{aff}} P\right) \tag{7}$$

Each row in $W \in \mathbb{R}^{k^{\text{sup}} \times k^{\text{self}}}$ represents the relation between self-supervised keypoints and corresponding supervised keypoint. Typically we have $k^{\text{sup}} \leq k^{\text{self}}$ for higher flexibility. The final supervised heatmaps is given by,

$$H_t^{\text{sup}} = H_t^{\text{self}} W^\top \tag{8}$$

That is, supervised heatmaps are a weighted sum or selection of the self-supervised heatmaps. This presents supervised loss as,

$$\mathcal{L}^{\text{sup}} = \left\| H_t^{\text{sup}} - H_t^{\text{gt}} \right\|^2 \tag{9}$$

where H_t^{gt} is the ground truth keypoint heatmap of target image built by placing a 2D gaussian at each joint's location [43, 65].

Our Transformer design explicitly models the relation between supervised and self-supervised tasks. Basic feature sharing model, even with the self-supervised task replaced by a similar pose estimation task, still fails to make sure that two tasks will cooperate instead of competing with each other. Learning an affinity matrix aligns self-supervised keypoints with supervised ones, avoiding the conflicts in multi-task training. During Test-Time Personalization, basic feature sharing model often lacks flexibility and is faced with the risk of overfitting to self-supervised task, due to the decoupling structure of two task heads. Our method, however, enforces the coupling between tasks using an affinity matrix and maintains flexibility as typically there are more self-supervised keypoints than supervised ones. Besides, compared to convolution model, Transformer shows superior ability to capture global context information, which is particularly needed when learning the relation between one supervised keypoint and all self-supervised ones.

Finally, we jointly optimize those two tasks during training. For a training sample, besides the supervised task, we randomly choose another sample belonging to the same person as the target to reconstruct. The final loss is given by

$$\mathcal{L} = \mathcal{L}^{\text{sup}} + \lambda \mathcal{L}^{\text{self}} \tag{10}$$

where λ is a weight coefficient for balancing two tasks.

3.2 Test-Time Personalization

During inference with a specific person domain, we apply Test-Time Personalization by fine-tuning the model relying solely on the self-supervised task. Given a set of N^{test} images of the same person $I_1^{\text{test}}, \ldots, I_{N^{\text{test}}}^{\text{test}}$, where $N^{\text{test}} > 1$, we first freeze the supervised Transformer part and update the shared encoder ϕ and the self-supervised head ψ^{self} with the reconstruction loss $\mathcal{L}^{\text{self}}$. Then the updated shared encoder ϕ^* and self-supervised head $\psi^{\text{self}*}$ are used along with the supervised head for final prediction. Specifically, during prediction, the updated features and self-supervised head will output improved keypoint heatmaps which leads to better reconstruction. These improved self-supervised heatmaps will go through the Transformer at the same time to generate improved supervised keypoints.

During the personalization process, we propose two settings including the *online* scenario which works in a stream of incoming data and the *offline* scenario which performs personalization on an unordered test image set. We illustrate the details below.

(i) The online scenario, which takes input as a sequence and requires real-time inference such as an online camera. In this setting, we can only choose both source I_s^{test} and target I_t^{test} with the constraint $s \leq T, t \leq T$ at time T for fine-tuning. Prediction is performed after each updating step.

(ii) The offline scenario, which has access to the whole person domain data and has no requirement of real-time inference, assuming we have access to an offline video or a set of unordered images for a person. In this setting, we shuffle the images in the dataset and perform offline fine-tuning, and then we perform prediction at once for all the images.

Compared to *online* scenario, *offline* scenario benefits from more diverse source and target sample pairs and avoids the variance drifts when updating the model. Since our method is designed to personalize pose estimation, the model is initialized with diversely trained weights when switching person identity. In each scenario, different re-initialization strategies can also be applied to avoid overfitting to a local minimum. The various combination of scenarios and reinitializing strategies engifts our method with great flexibility.

It should be noted that our method has *no requirement of consecutive or continuous* frames but only unlabeled images belonging to the same person, which is less costly and beyond the reach of temporal methods such as 3D convolution with multiple frames. Test-Time Personalization can be done at inference without annotations and thus is remarkably suitable for privacy protection: The process can be proceeded locally rather than uploading data of your own for annotating for specialization.

4 Experiment

4.1 Datasets

Our experiments are performed on three human pose datasets with large varieties to prove the generality and effectiveness of our methods. While the datasets are continuous videos, we emphasize that our approach can be generalized to discontinuous images. In fact, we take the datasets as unordered image collections when performing *offline* Test-Time Personalization. All input images are resized to 128×128 with the human located in the center.

Human 3.6M [24] contains 3.6 million images and provides both 2D and 3D pose annotations. We only use 2D labels. Following the standard protocol [74, 34], we used 5 subjects for training and 2 subjects for testing. We sample the training set every 5 frames and the testing set every 200 frames. We use the Percentage of Correct Key (PCK) as the metric and the threshold we use is the 20% distance of the torso length.

Penn Action [71] contains 2,236 video sequences of different people. 13 pose joints are given for each sample in the annotations. We use the standard training/testing split and also use PCK with threshold distance of half distance of torso as the evaluation metric.

BBC Pose [8] consists of 20 videos of different sign language interpreter. We use 610,115 labeled frames in the first ten videos for training, and we use 2,000 frames in the remaining ten videos (200 frames per video) with manual annotation for testing. The testing frames are not consecutive. The evaluation method of BBC Pose is the joint accuracy with d pixels of ground truth where d is 6 following [9, 12, 48, 26].

Table 1: Evaluation results on pose estimation. Our proposed method is denoted as *Transformer (keypoint)*. For Human 3.6M and Penn Action datasets, mPCK is employed as the metric while for BBC Pose we use mAcc. The proposed method not only performs better on the validation set but also enjoys more gain in Test-Time Personalization.

Method	TTP Scenario	Human 3.6M	Penn Action	BBC Pose
Baseline	w/o TTP	85.42	85.23	88.69
Feat. shared (*rotation*)	w/o TTP	87.37 (+1.95)	84.90 (−0.33)	89.07 (+0.38)
	Online	88.01 (+2.59)	85.86 (+0.63)	89.34 (+0.65)
	Offline	88.26 (+2.84)	85.93 (+0.70)	88.90 (+0.21)
Feat. shared (*keypoint*)	w/o TTP	87.41 (+1.99)	85.78 (+0.55)	89.65 (+0.96)
	Online	89.43 (+4.01)	87.27 (+2.04)	91.48 (+2.79)
	Offline	89.05 (+3.63)	88.12 (+2.89)	91.65 (+2.96)
Transformer (*keypoint*)	w/o TTP	87.90 (+2.48)	86.16 (+0.93)	90.19 (+1.50)
	Online	91.70 (+6.28)	87.75 (+2.52)	**92.51 (+3.82)**
	Offline	**92.05 (+6.63)**	**88.98 (+3.75)**	92.21 (+3.52)

4.2 Implementation Details

Network Architecture. We use ResNet [22] followed by three transposed convolution layers as encoder ϕ. Every convolution layer has 256 channels, consisting of BatchNorm and ReLU activation and upsampling 2 times to generate the final feature F of size $256 \times 32 \times 32$ and $c = 256$. Considering the diversity of datasets, we use ResNet50 for Penn Action and ResNet18 for both Human 3.6M and BBC Pose. We use one convolution layer as the supervised head ψ^{sup} and another convolution layer for self-supervised head ψ^{self}, where the supervised head ψ^{sup} denotes the standard structure used in Appendix A. For all three datasets, the output channel for self-supervised keypoints is $k^{\text{self}} = 30$. We adopt a 1-layer Transformer with 4 heads and the hidden layer in feed-forward has 1024 dimensions. The weight of self-supervised loss is set to $\lambda = 1 \times 10^{-3}$ for Penn Action and BBC Pose, $\lambda = 1 \times 10^{-5}$ for Human 3.6M.

Joint Training. We apply the same training schedule across methods. For all datasets, we use batch size of 32, Adam [31] optimizer with learning rate 0.001 and decay the learning rate twice during training. We use learning schedule [18k, 24k, 28k], [246k, 328k, 383k] and [90k, 120k, 140k] for BBC Pose, Penn Action, and Human 3.6M respectively. We divide the learning rate by 10 after each stage. The training schedule of BBC Pose is shortened since the data is less diverse.

Test-Time Personalization (TTP). During Test-Time Personalization, we use Adam optimizer with fixed learning rate 1×10^{-4}. The supervised head ψ^{sup} and Transformer are frozen at this stage. Test-Time Personalization is applied without weight reset unless specified. In offline scenario, even though the model can be updated for arbitrary steps, we adopt the same number of steps as the online scenario for a fair comparison. See Appendix C for more details.

Batching Strategy. In joint training and TTP offline scenario, both target and source images are randomly chosen and are different within a batch. In TTP online scenario, the target images are always the current frame, which are the same within a batch, whereas the source images are randomly chosen from the previous frames and are different in a batch. In all the scenarios, each target-source pair is performed data augmentation with same rotation angle and scale factor for the two images to make reconstruction easier.

4.3 Main Results

To better analyze the proposed method, in Table 1 we compare it with three baselines: (i) **Baseline.** The plain baseline trained with supervised loss only. (ii) **Feat. shared (rotation).** Instead of self-supervised keypoint estimation, we use rotation prediction to compute the self-supervised loss $\mathcal{L}^{\text{self}}$ in Eq. 10 following Sun et al. [58]. Rotation is predicted with a standalone supervised head ψ^{sup}. The two tasks have no direct relation except they share the same feature backbone. Weight coefficient λ is set to 1×10^{-4} for better performance. (iii) **Feat. shared (keypoint).** We use the self-supervised keypoint estimation task [26] as the self-supervised objective. However, supervised keypoints are still estimated with a standalone supervised head ψ^{sup} instead of our Transformer design. The two tasks are only connected by sharing the same feature backbone. See Appendix A for more details. Finally, our proposed method is denoted as **Transformer (keypoint)**.

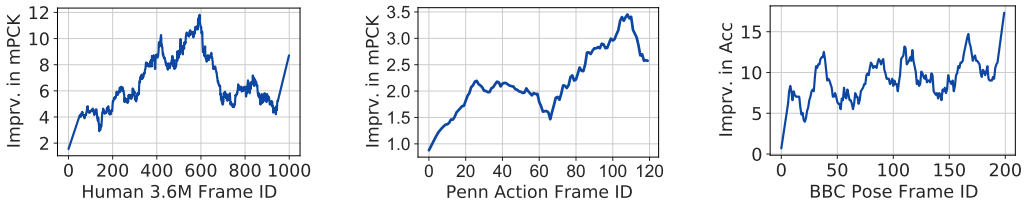

Figure 3: Improvement vs Frame ID in online scenario for 3 datasets. We plot the gap between the Test-Time Personalization and the baseline model for each frame step. We adopt the averaged metric across all test videos. In most cases, we observe TTP improves more over time.

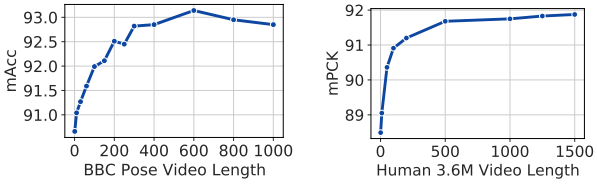

Figure 4: Test-Time Personalization with different numbers of unlabeled test samples. **Left:** mAcc for different video length on BBC Pose. **Right:** mPCK for different video length on Human 3.6M. Test-Time Personalization benefits from utilizing more unlabeled test samples.

Table 2: Test-Time Personalization in online scenario with different update iterations.

Iters	Penn Action	BBC Pose
1	87.75	92.51
2	**88.01**	**92.64**
3	76.27	92.59
4	76.13	92.53

Despite using calibrated self-supervised task weight, *Feat. shared (rotation)* still shows limited and even degraded performance on all three datasets, indicating that a general self-supervised task without a specific design is likely to hurt the performance of supervised one. On the other hand, *Feat. shared (keypoint)* presents superior performance over *Baseline*, even without Test-Time Personalization. This demonstrates the hypothesis that selecting a related or similar self-supervised task can facilitate the original supervised task and thus naturally leads to better performance in Test-Time Personalization. The results of Test-Time Personalization show the personalizing ability of our method. Personalizing for a single person results in significant improvement.

Transformer (keypoint) further boosts performance with Test-Time Personalization, with an improvement of 6.63 mPCK on Human 3.6M, 3.75 mPCK on Penn Action, and 3.82 mAcc on BBC Pose. More importantly, our design of learning an affinity matrix not only improves the performance of joint training but also achieves a higher improvement in Test-Time Personalization. For example, TTP in online scenario has an improvement of 2.32 mAcc with *Transformer (keypoint)* compared to an improvement of 1.83 mAcc with *Feat. shared (keypoint)* for BBC Pose. This demonstrates that using the proposed Transformer, two tasks cooperate better in joint training and have higher flexibility in Test-Time Personalization.

In terms of different scenarios for Test-Time Personalization, it is found that the offline scenario does not always surpass online scenario. For example in BBC Pose, both online scenario and offline scenario improve performance, yet in offline scenario, there is a small decrease in mAcc. This is expected for two reasons. Firstly the major advantage of offline scenario comes from the diversity of test samples while BBC Pose has a nonconsecutive validation set selected specifically for diversity. Secondly, we set the learning rate based on the performance of online scenario and follow it in all settings to demonstrates the generality of our method. Better results can be achieved if the learning rate is adjusted more carefully.

4.4 Analysis on Test-Time Personalization

Number of Unlabeled Test Samples. Our method exploits personal information using unlabeled samples in a single person domain. We observe that more unlabeled samples can further improve the performance in Test-Time Personalization. We study the number of unlabeled samples using extra validation samples of BBC Pose and Human 3.6M. We emphasize that although labels are also provided for extra validation samples, we only use images *without* labels. All experiments have the same setting as Transformer in online scenario and the prediction and evaluation are on the same fixed test set. In Figure 4, we report results of TTP by using different video lengths of samples in fine-tuning in an online manner. For video length smaller than the actual test sequences, we apply

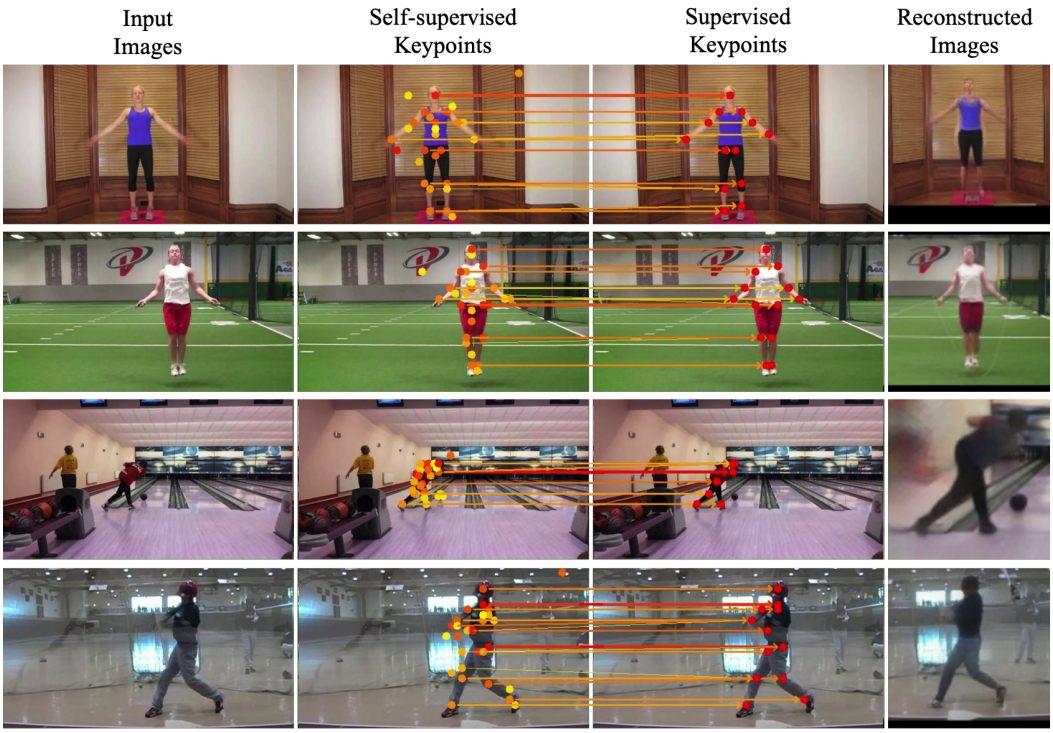

Figure 5: Visualization on Penn Action. The images from the left to the right are: the original image, the image with 30 self-supervised keypoints, the image with 13 supervised keypoints, and the reconstructed image from the self-supervised task. The arrows between keypoints indicate their correspondences obtained from the affinity matrix with the Transformer. Warmer color indicates higher confidence.

reinitializing strategy to simulate shorter videos. We observe that for Human 3.6M, the performance of our model increases as the number of unlabeled samples grows. Similar results appear in BBC Pose except that the performance reduces slightly after using more than 600 frames in fine-tuning. The reason is that the changes of the person images in BBC Pose are very small over time, which leads to overfitting.

Improvement in Online Scenario. Figure 3 shows the improvement curve within each test video in the online scenario with respect to the ID (n-th update) of frames in TTP. We compute the metric gap between our method using TTP and baseline without TTP for each ID. The results are averaged across all the test videos. In Human 3.6M, we report on a single subject S9. The curves are smoothed to reduce variance for better visualization. The result suggests the gap keeps increasing within a single test video, as the model updates at every frame. Moreover, in later frames, the model has seen more test samples, which helps enlarge the performance gap.

In Human 3.6M, which has much more samples in a single person domain, the performance improves at the beginning but the improvement starts to drop a bit at 600 time steps due to overfitting in later frames. This phenomenon is expected since the examples in Human 3.6M are also quite similar. Note that the gap still exists for later frames, it is only the improvement that becomes smaller.

Transfer Across Datasets. We test the generalization of the model by conducting the full training on one dataset and applying TTP on another dataset. We train our model on the Penn Action, and then perform TTP to transfer the model to Human 3.6M. As shown in Table 6, our method using TTP can improve over the baselines by a large margin. Our offline approach achieves around 8% improvement over the baseline without using Transformer and TTP. This shows TTP can significantly help generalize and adapt the pose estimator across data from different distribution.

Update Iterations. We show the ablation on update iterations in Table 2. Note that in our online scenario setting, we only update the model once for every incoming test image. We present results where we update more iterations in Table 2. It suggests that more update iterations do not help much. Specifically, for Penn Action the performance drops when we update 3 to 4 iterations. The reason

Table 5: Smoothing results for Penn Action. Our method is complementary to temporal methods.

Method	TTP Scenario	mPCK	w/ smoothing
Baseline	w/o TTP	85.23	85.68 (+0.45)
Transformer	w/o TTP	86.16	86.58 (+0.42)
Transformer	Online	87.75	88.31 (+0.56)
Transformer	Offline	**88.98**	**89.51 (+0.53)**

Table 6: The results of transferring the model trained on Penn to Human3.6M.

Method	TTP Scenario	mPCK
Baseline	w/o TTP	52.60
Transformer	w/o TTP	56.27
Transformer	Online	60.14
Transformer	Offline	61.04

is, in each step of online setting, we only perform training on one single frame, which can lead to overfitting to a particular image.

4.5 Comparisons with Video Models

Visualization. We provide visualization on Penn Action experiments in Figure 5. We visualize the self-supervised keypoints and the supervised keypoints (2nd and 3rd columns). The arrows from the self-supervised keypoints and supervised keypoints indicate the keypoint correspondence, according to the affinity matrix in the Transformer. We show arrows (correspondence) where the probability is larger than 0.1 in the affinity matrix. We use warmer color to indicate larger confidence for both keypoints and arrows. The last column visualizes the reconstructed target image in our self-supervised task, which has the size as the network inputs cropped from the original images. See Appendix B for more visualization results.

Complementary to Temporal Methods. Even though our method is designed for single image input and requires no consecutive frames like videos, it is complementary to temporal methods such as 3D convolution [47] or smoothing techniques [48]. We apply Savitzky–Golay filter for smoothing along with our methods for demonstration. In Table 5 we present the results on Penn Action, as Penn Action is the only dataset here with completely consecutive test samples. After smoothing, our method presents a similar improvement to baseline. The increase in accuracy when performing smoothing is too small so the performance gain of our method does not come from temporal information and can be further improved combined with temporal methods.

Table 3: Comparisons with state-of-the-art on Penn Action.

Method	Penn Action
Baseline	85.2
Ours	89.0
video-based methods	
Iqbal et al. [25]	81.1
Song et al. [52]	96.5
Luo et al. [39]	97.7

Table 4: Comparisons with state-of-the-art on BBC Pose. Result with (*) is reported in [26]. Ours *(best)* is with extra unlabeled samples.

Method	BBC Pose
*Charles et al. [9]	79.9
Baseline	88.7
Ours	92.5
Ours ***(best)***	93.1
video-based methods	
Pfister et al. [48]	88.0
Charles et al. [10]	95.6

Comparisons with State-of-the-Art. In Table 3 and Table 4 we compare our best results with state-of-the-art models on Penn Action and BBC Pose. Note that most of the methods on both datasets use multiple video frames as inputs and use larger input resolutions, which makes them not directly comparable with our method. We report the results for references. We argue that our approach with single image input has more flexibility and can be generalized beyond video data. Most works on Human 3.6M focus on 3D pose estimation thus are not reported.

5 Conclusion

In this paper, we propose to personalize human pose estimation with unlabeled test samples during test time. Our proposed Test-Time Personalization approach is firstly trained with diverse data, and then updated during test time using self-supervised keypoints to adapt to a specific subject. To enhance the relation between supervised and self-supervised tasks, we propose a Transformer design that allows supervised pose estimation to be directly improved from fine-tuning self-supervised keypoints. Our proposed method shows significant improvement over baseline on multiple datasets.

Societal Impact. The proposed method improves the performance on the human pose estimation task, which has a variety of applications such as falling detection, body language understanding, autonomously sports and dancing teaching. Furthermore, our method utilizes unlabeled personal data and thus can be deployed locally, reducing human effort and avoiding privacy violation. However, the proposed method can also be used for malicious purposes, such as surveillance. To avoid possible harmful applications, we are committed to limit our methods from any potential malicious usage.

Acknowledgments and Funding Transparency Statement. This work was supported, in part, by grants from DARPA LwLL, NSF National Artificial Intelligence (AI) Research Institute TILOS, NSF 1730158 CI-New: Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI), NSF ACI-1541349 CC*DNI Pacific Research Platform, and gifts from Qualcomm, TuSimple and Picsart.

References

[1] Bruno Artacho and Andreas Savakis. Unipose: Unified human pose estimation in single images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7035–7044, 2020. 3

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[3] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Trans. Graph.*, 38(4), 2019. 3

[4] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[5] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 2

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3

[7] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016. 1, 2

[8] James Charles, Tomas Pfister, Mark Everingham, and Andrew Zisserman. Automatic and efficient human pose estimation for sign language videos. *International Journal of Computer Vision*, 110(1):70–90, 2014. 2, 6

[9] James Charles, Tomas Pfister, D. Magee, David C. Hogg, and Andrew Zisserman. Domain adaptation for upper body pose tracking in signed tv broadcasts. In *BMVC*, 2013. 6, 10

[10] James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Personalizing human video pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3063–3072, 2016. 1, 3, 10

[11] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 3

[12] Xianjie Chen and Alan Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. *arXiv preprint arXiv:1407.3399*, 2014. 6

[13] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018. 2

[14] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5386–5395, 2020. 2

[15] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Structured feature learning for pose estimation. In *CVPR*, 2016. 1, 2

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 1, 4

[19] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 350–359, 2018. 3

[20] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A. Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *International Conference on Learning Representations*, 2021. 3

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[23] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 3

[24] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 2, 6

[25] Umar Iqbal, Martin Garbade, and Juergen Gall. Pose for action-action for pose. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 438–445. IEEE, 2017. 10

[26] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. *arXiv preprint arXiv:1806.07823*, 2018. 2, 3, 6, 7, 10

[27] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8787–8797, 2020. 3

[28] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. *arXiv preprint arXiv:2104.03304*, 2021. 3

[29] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 4

[30] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. *arXiv preprint arXiv:2004.03686*, 2020. 3

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[32] Tejas D Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *Advances in neural information processing systems*, 32:10724–10734, 2019. 3

[33] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. *arXiv preprint arXiv:2104.06976*, 2021. 3

[34] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014. 6

[35] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. *arXiv preprint arXiv:2012.03196*, 2020. 1, 3

[36] Yunzhu Li, Antonio Torralba, Animashree Anandkumar, Dieter Fox, and Animesh Garg. Causal discovery in physical systems from videos. *arXiv preprint arXiv:2007.00631*, 2020. 3

[37] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020. 3

[38] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2019. 3

[39] Yue Luo, Jimmy Ren, Zhouxia Wang, Wenxiu Sun, Jinshan Pan, Jianbo Liu, Jiahao Pang, and Liang Lin. Lstm pose machines. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5207–5215, 2018. 10

[40] Lucas Manuelli, Yunzhu Li, Pete Florence, and Russ Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. *arXiv preprint arXiv:2009.05085*, 2020. 3

[41] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. Tfpose: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*, 2021. 3

[42] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. *arXiv preprint arXiv:2104.07645*, 2021. 3

[43] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 1, 2, 5

[44] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6951–6960, 2019. 2

[45] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017. 2

[46] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 3

[47] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 10

[48] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015. 6, 10

[49] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 3

[50] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and remapping the "dna" of a natural image, 2018. 3

[51] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *CVPR*, pages 3118–3126. IEEE Computer Society, 2018. 3

[52] Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4220–4229, 2017. 10

[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 5

[54] Lucas Stoffl, Maxime Vidal, and Alexander Mathis. End-to-end trainable multi-instance pose estimation with transformers. *arXiv preprint arXiv:2103.12115*, 2021. 3

[55] Omer Sumer, Tobias Dencker, and Bjorn Ommer. Self-supervised learning of pose embeddings from spatiotemporal relations in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4298–4307, 2017. 3

[56] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 3

[57] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 2

[58] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020. 1, 2, 3, 7

[59] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 1, 2

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2, 3

[61] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 1, 3

[62] Manchen Wang, Joseph Tighe, and Davide Modolo. Combining detection and tracking for human pose estimation in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11088–11096, 2020. 3

[63] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

[64] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 1, 2

[65] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 2, 5

[66] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2020. 3

[67] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *proceedings of the IEEE international conference on computer vision*, pages 1281–1290, 2017. 2

[68] Xianfang Zeng, Yusu Pan, Mengmeng Wang, Jiangning Zhang, and Yong Liu. Realistic face reenactment via self-supervised disentangling of identity and pose. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12757–12764, 2020. 3

[69] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group distribution shift. *arXiv preprint arXiv:2007.02931*, 2020. 1, 3

[70] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 4

[71] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2248–2255, 2013. 2, 6

[72] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2694–2703, 2018. 3

[73] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085, 2020. 3

[74] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 6

[75] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

A Pipeline of the Alternative Method

For clarification, we show the alternative method we discussed and compared the proposed method with. It is denoted as *Feat. shared (keypoint)* in Section 4.3. Instead of using a Transformer to model the relation between two sets of keypoints, we simply use a supervised head ψ^{sup} to predict H^{sup}. Two tasks are only connected by sharing a feature backbone.

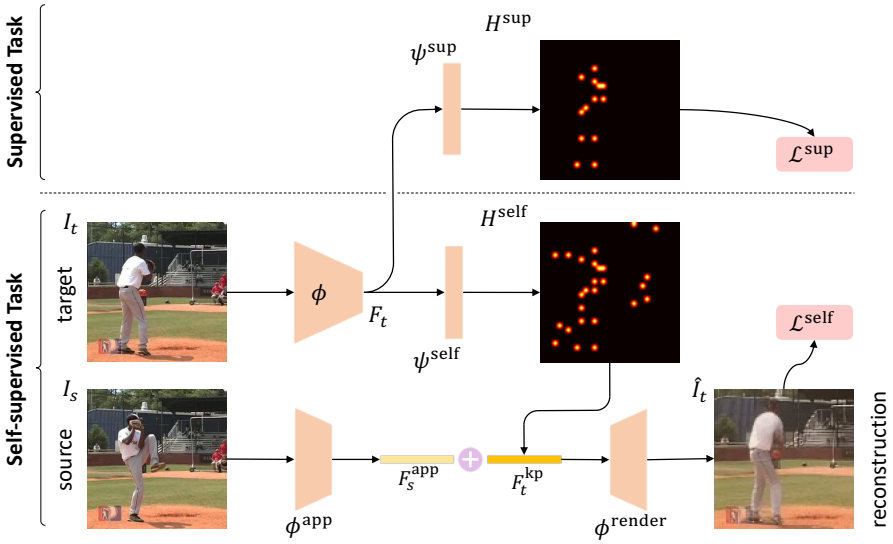

Figure 6: The alternative pipeline *Feat. shared (keypoint)* we discussed and compared the proposed method with.

B Visualization

In Figure 7 and Figure 8 we visualize our predictions on Penn Action validation set. From top to bottom, the images are: (i) **target image** I_t, i.e. the input image. (ii) **source image** I_s, which provides appearance. (iii) **reconstruction** $\hat{I}_t$. (iv) **self-supervised keypoints**. There are 30 self-supervised keypoints in our setting. (v) **supervised keypoints**. (vi) **ground-truth**.

For self-supervised keypoints, we show the contribution of each keypoint to the final pose estimation with color. This is computed as follows. Recall that the Transformer models the relation between two tasks as the affinity matrix

$$W \in \mathbb{R}^{k^{\text{sup}} \times k^{\text{self}}}, \tag{11}$$

where k^{sup} and k^{self} are the number of supervised and self-supervised keypoints. Also recall that

$$H_t^{\text{sup}} = H_t^{\text{self}} W^{\top}. \tag{12}$$

An entry $W_{i,j}$ actually represents the weight of j-th self-supervised keypoint in computing the i-th supervised keypoint. We then define the contribution of j-th self-supervised keypoint to the final pose prediction as follows

$$c_j = \sum_{i=1}^{k^{\text{sup}}} W_{i,j}. \tag{13}$$

The keypoints with larger c_j are more important to the final pose prediction. Whereas the keypoints with smaller c_j are less important to the final pose prediction and serve to facilitate the self-supervised task of reconstruction.

In Figure 7 and Figure 8 we show the self-supervised keypoints with their contribution to the final pose estimation in the fourth row. It is clear that the points that align with the position of supervised keypoints usually have higher contribution. Other points with deviated positions have

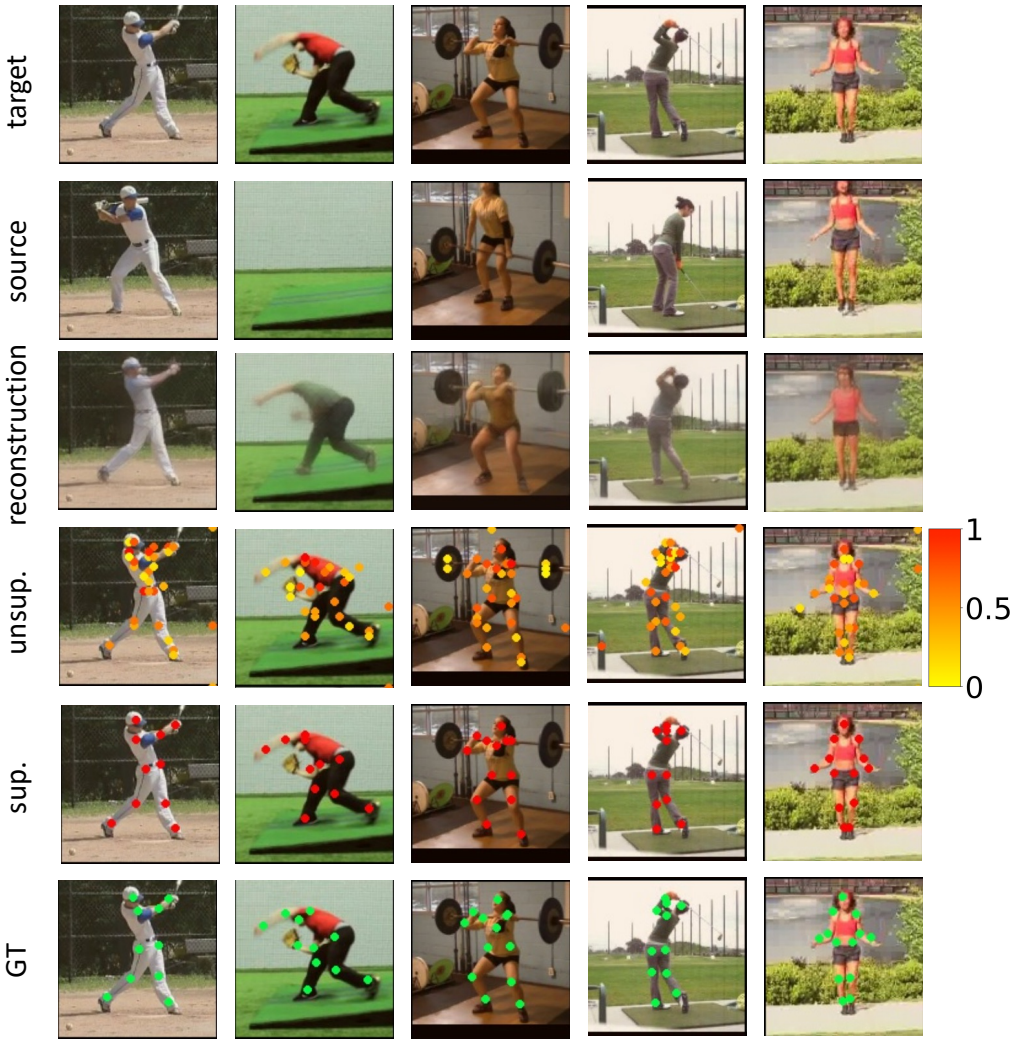

Figure 7: Visualization of our proposed method on Penn Action validation set.

lower contribution. They appear to serve more to facilitate the self-supervised task. Interestingly, some keypoints with minor contribution locate the non-human objects in Penn Action (barbell, bowling ball).

C More Implementation Details

More Model Details. Normally, the images I_s and I_t are 128×128. The heatmaps H_t^{sup} and H_t^{self} are 32×32. In self-supervised task, appearance information F_s^{app} and keypoint information F_t^{kp} has size 16×16 with 256 channels. For the perceptual loss, we use a VGG-16 network pretrained on ImageNet to extract semantic informations. We do not use flip test during inference.

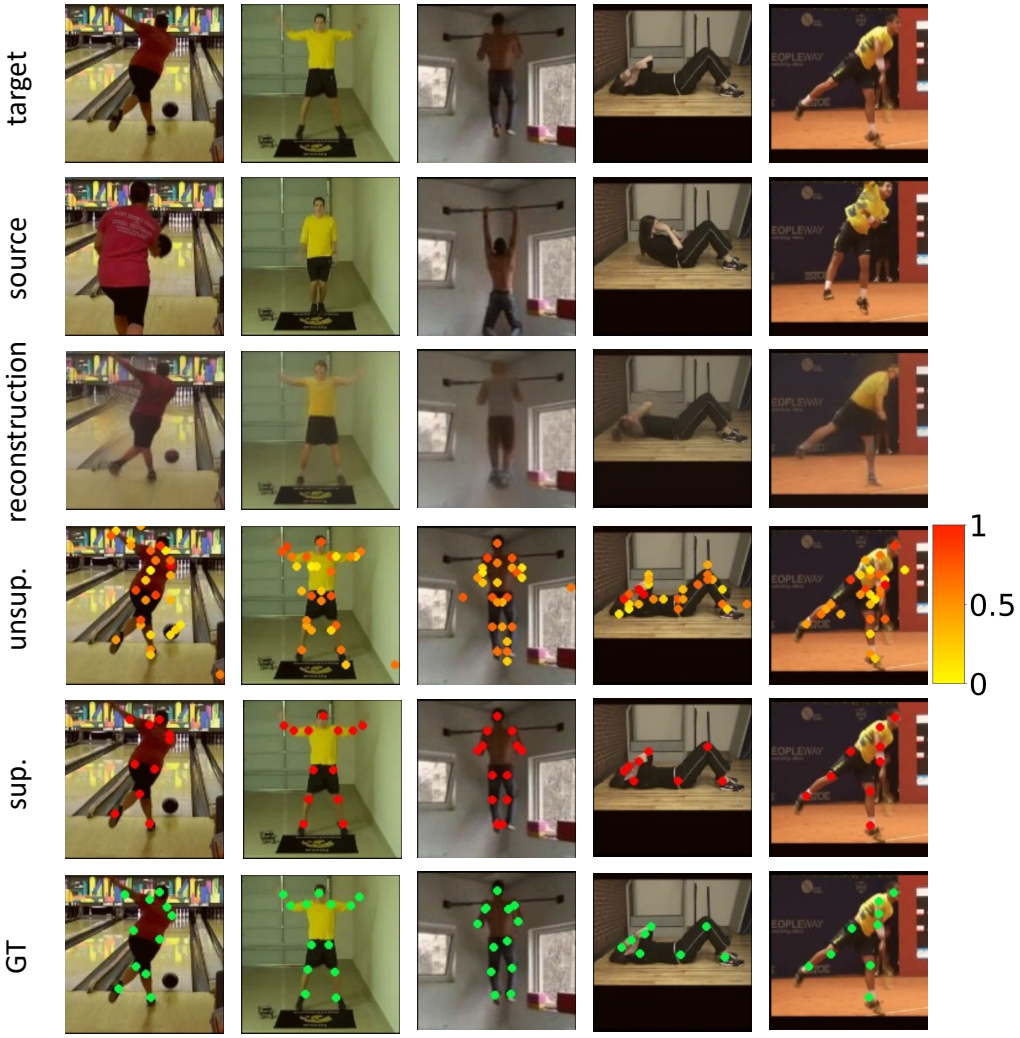

Figure 8: Visualization of our proposed method on Penn Action validation set.