Fast Online Video Pose Estimation by Dynamic Bayesian Modeling of Mode Transitions

Ming-Ching Chang, Lipeng Ke, Honggang Qi, Longyin Wen, Siwei Lyu

***Abstract*—We propose a fast *online* video pose estimation method to detect and track human upper-body poses based on a conditional dynamic Bayesian modeling of pose modes without referring to future frames. Estimation of human body poses from video is an important task with many applications. Our method extends fast image-based pose estimation to live video streams by leveraging the temporal correlation of articulated poses between frames. Video pose estimation are inferred over a time window using a conditional dynamic Bayesian network (CDBN), which we term T-CDBN. Specifically, latent pose modes and their transitions are modeled and co-determined from the combination of three modules: (1) inference based on current observations, (2) the modeling of mode-to-mode transitions as a probabilistic prior, and (3) the modeling of state-to-mode transitions using a multi-mode softmax regression. Given the predicted pose modes, the body poses in terms of arm joint locations can then be determined more accurately and robustly. Our method is suitable to investigate high frame rate (HFR) scenarios, where pose mode transitions can effectively capture action-related priors to boost performance. We evaluate our method on a newly collected HFR-Pose dataset and four major video pose datasets (VideoPose2, TUM Kitchen, FLIC and Penn_Action). Our method achieves improvements in both accuracy and efficiency over existing online video pose estimation methods.**

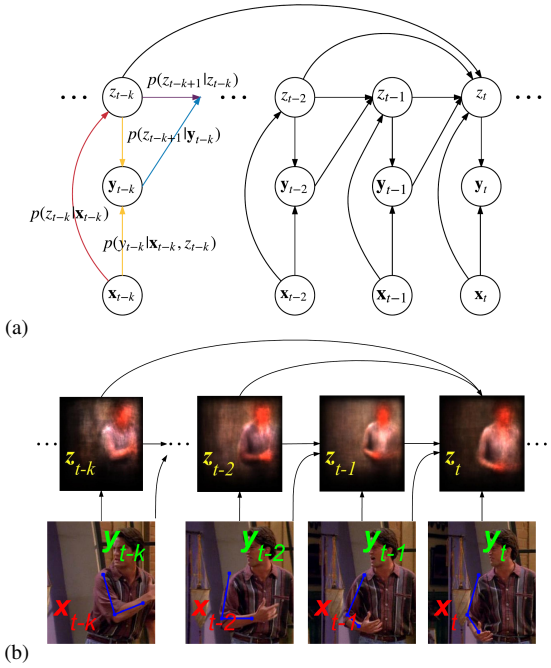

Fig. 1. **Method overview.** *(a) The proposed T-CDBN model structure for online video pose estimation and (b) an example of the T-CDBN-MODEC based on the MODEC single-image pose estimation [1] applied to online video pose estimation.* $\mathbf{x}_t$ *corresponds to observations* (i.e., *image features*) *at time* t, $\mathbf{y}_t$ *corresponds to body pose* (i.e., *joint locations*), *and* z_t *is the latent pose mode in individual frames. See Section III-A for explanations.*

I. Introduction

As the basis for understanding human actions and behaviors from visual imagery, upper-body pose estimation from videos has many applications, including gesture recognition, human computer interaction, gaming, sign language recognition, and the study of affective and social behaviors. With the ubiquity of inexpensive video cameras on mobile devices, it has become increasingly easy to capture live feed videos, from which upper-body poses or gestures can be estimated as a continuous time series for further processing. As such, we focus on electro-optical (RGB) videos rather than RGB+D videos (those obtained from *e.g.* Microsoft Kinect) that operate based on depth sensors, for the consideration of generality and applicability in real-world usages.

With the maturity of efficient image-based pose estimation methods [2], [3], [1], [4], [5], [6], [7], one naive solution to video pose estimation is to apply image-based methods to individual frames of a video as if they are independent images. However, this approach works only to a certain extent. The main drawback is that the strong temporal correlations of articulated poses between video frames are discarded. It is intuitive that human activities usually involve smooth and continuous hand movements. Thus the continuity of poses in consecutive video frames provides strong cues for robust pose estimation through tracking and prediction. Treating individual frames without considering temporal correlations leads to inefficient algorithm and inaccurate estimations, due to ambiguities and occlusions in a single frame. In contrast, estimating upper-body poses as a continuous temporal sequence leads to better handling of occlusions and robustness in estimation.

Methods for pose estimation from videos (and particularly the ones focusing on upper-body poses) have advanced significantly in recent years *e.g.* [8], [9], [10], [11], [12], [13], [14]. However, the majority of existing methods are *offline* in nature, *i.e.*, upper-body poses in the current frame are inferred using both the past and future frames. The performance of these methods on estimating poses usually comes at the price of complicated inference procedures, which significantly reduces the running efficiency. Thus these methods do not address the practical needs of fast video pose estimation.

Ming-Ching Chang is with the Computer Science Department, University at Albany, SUNY, NY, USA.

Lipeng Ke and Honggang Qi are with the School of Computer and Control Engineering, University of the Chinese Academy of Sciences, Beijing, China.

Longyin Wen is with the GE Global Research Center, NY, USA.

Siwei Lyu is with the Computer Science Department, University at Albany, SUNY, NY, USA.

Siwei Lyu is the corresponding author.

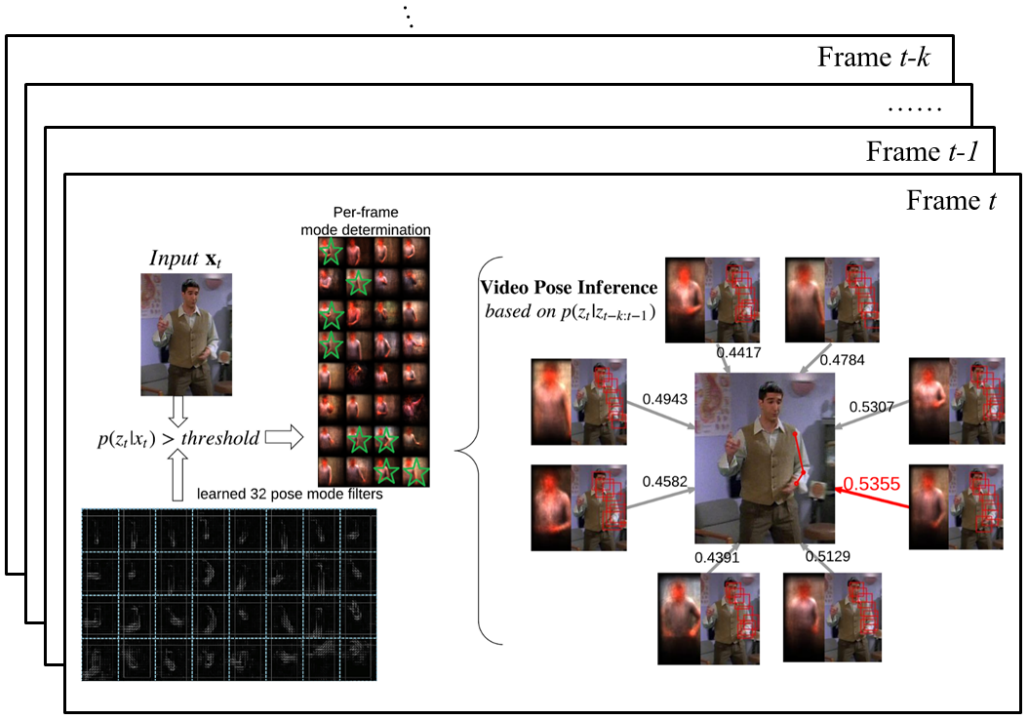

Fig. 2. **T-CDBN-MODEC video pose estimation by dynamic modeling of mode transitions.** *At each frame, candidate poses are estimated by a cascaded prediction of first the coarse modes (left) and then the fine parts (right). For each left or right half-body, we estimate the possible coarse pose modes out of $M = 32$ total modes, by convolving the image with learned mode filters. We construct a graph model to infer the most probable mode across several temporal frames using a multi-label selecting method. We then determine finer body joint locations using finer parts models, guided by the most probable mode. Input image is from the Friends TV show in the VideoPose2 dataset.*

In this work, we describe a fast *online* method for video pose estimation. We aim to extend existing image-based pose estimation methods to video streams, by leveraging the temporal continuity of articulated poses between frames over a time window. We directly model the latent *pose modes* (or "poselets" [15]) to improve motion consistency, in a paradigm similar to detect-and-track for object tracking. Specifically, we explicitly model the **transitions** *between a finite number of pose modes*. Thus, the problem is formulated as (1) a base module of pose detection in individual frames and (2) the inference of the underlying pose mode transitions between frames over a time window. Our method is formulated base on a general *time-windowed conditional dynamic Bayesian network* (T-CDBN) model, which is a combination of two widely used probabilistic graphical models, *i.e.* dynamic Bayesian network (DBN) [16] and conditional random field (CRF) [17]. Fig. 1 illustrates the structure of T-CDBN as a graphical model. The DBN aspect of our model captures the temporal correlations between variables in a sequence, and the CRF aspect incorporates the complex relations between the observations and latent variables. Fast *online* estimation is achieved with an efficient *particle filtering* implementation of the inference. [1]

A key characteristic of the T-CDBN model is that it is an "open architecture", as it can incorporate different underlying CRF models (including future ones) into the DBN structure, as long as the image-based CRF pose estimation can represent latent pose modes. When applying to *online* video pose estimation, our approach becomes advantageous, as it allows the resulting algorithm to incorporate effective single frame pose estimation methods into a dynamic framework that models intra-frame correlations. In this work, we adopt part of the efficient CRF pipeline from the MODEC method [1] as the CRF model in our T-CDBN framework, see Fig. 2. We term our method T-CDBN-MODEC for video pose estimation, as the pose CDBN inference is performed within a time window based on the MODEC single-image-based pose estimation.

Our method is suitable for online video pose estimation from high frame rate (HFR) videos. This is because in HFR videos, body motion or hand movements are likely to be captured by many frames. In this case effective modeling of pose mode transitions between frames can provide critical prior knowledge to predict and infer the unknown poses. The performance of pose estimation increases as the underlying input frame rate increases at run time, and the overall performance is bounded by the pose estimation processing speed. Our method thus aims to find a balance between performance and speed, while maximizing the utility of computational resource. To better evaluate the practical needs calling for online HFR video pose estimation, we create a new HFR labeled dataset (in Section V-A) to investigate the aspects which are less addressed in most existing works.

For single-image pose estimation, recent deep learning approaches have improved largely in accuracy by applying cascade body-joint regression [18] and body-joint heat-map regression [19], where popular networks including the Convo-

[1]Our approach is generic to handle full body poses, and we only focus on upper-body pose estimation from on-line videos in this work.

lutional Neural Networks (CNN) and AlexNet are trained on large datasets. For video pose estimation [20], [21], [22], [23], motion flow are directly leveraged in networks such as the VGG and GoogLeNet to ensure temporal consistency across frames. However, all existing deep network based methods operate as black box models, thus the explicit modeling of pose modes is not possible. We did not integrate deep network based pose estimation methods into our T-CDBN framework, as they are not suitable in this regard. In addition, most deep methods rely on GPU implementations, which limits their applicability on mobile or low-power platforms.

An early version of this work based on CDBN was published in [24]. This paper improves this previous work in the following aspects. First, we extend the estimation of latent pose mode transitions which was manually aggregated and tweaked from a small dataset in [24] to a completely automatic method in the new T-CDBN formulation with two improvements. (1) The mode-to-mode transitions are learned automatically from robust statistics of actions in a sufficiently large time window. (2) The pose-to-mode predictions are learned using a new multi-mode *softmax* regression, which further improves the robustness pose estimation across frames. Secondly, we comprehensively evaluate the proposed method on two significantly larger and challenging and datasets — TUM Kitchen [25] and FLIC [1]datasets, in addition to the VideoPose2 dataset and our own HFR-Pose dataset. T-CDBN outperforms the original CDBN [24] and other online video pose estimation methods on the benchmark datasets. T-CDBN is also the fastest among all available methods. It achieves 9.4 FPS processing speed on the TUM Kitchen dataset without relying on GPU. Details on performance and speed evaluation will be presented in Fig. 7 and Section V-E.

The rest of the paper is organized as follows. Section II discusses related literature in details. Section III describes the general T-CDBN formulation. Section IV applies the T-CDBN framework to online video pose estimation and describes the learning and inference of the framework. Section V evaluates our method against state-of-art methods on several benchmark datasets. Section VI concludes the paper with discussions and future works.

II. Related Works

The vast amount of literature in human pose estimation can be organized into three main categories based on the forms of inputs: (1) a single image, (2) a video sequence, or (3) a live video stream.

A. Image Based Pose Estimation

There exists extensive works on pose estimation from a single image *e.g.*, [4], [26], [1], [2], [3], [5], [6], [7]. Yang and Ramanan [3] introduced a Flexible Mixture-of-Parts model for human pose recognition. This method allows parts to be selected from several types, where a tree-based Deformable Parts Model (DPM) [27] is learned. Bourdev and Malik [15] introduced the concept of the "poselet" by robustly clustering the 3D part coordinates with respect to their 2D appearances, such that 2D part detectors can be effectively trained from 3D pose annotations. Sapp and Taskar's *multimodal decomposable models* (MODEC) [1] introduced multi-modality at the coarse-body and fine-part (shoulder, elbow and wrist) granularities. They divide the upper-body pose into two half-side bodies, and for each side estimate pose *modes*. The pose modes are essentially the poselet clusters for the half-side body, which can guide the tracking of arm and joint locations (similar to the DPM scheme) in a single-path tree structure. Given a torso bounding box as an input, shoulder positions are estimated more accurately than the elbows and wrists. Section IV-A will describe the MODEC method in further details.

B. Offline *Video Pose Estimation*

Significant efforts have been invested on the estimation of human pose from videos [8], [9], [10], [11], [12], [13], [14]. Motion flow cues and temporal features are the keys in this category of methods [11], [12], [13]. Many methods exploit optical flow for pose estimation. Ferrari *et al.* [12] first used segmentation to aid the detection of body pose in each frame, and then calculated the motion of lower limbs across frames. The "flow puppets" of Zuffi *et al.* [13] used a 2D upper-body shape model to track articulated motions, where motion cues were integrated jointly with the pose inference. The offline method of Cherian *et al.* [11] consists of two steps. The first step extends the work of [3] by adding motion flow links during the inference of poses between consecutive frames. Across-frame links between elbows and wrists introduce loops in the inference, thus loopy belief propagation (BP) is introduced as a solver. The second step decomposes the candidate poses into individual limbs, where the limbs are recomposed after temporal smoothing. This method achieves high accuracy among *offline* pose estimation methods as shown in Fig. 7 in our evaluation. However its computation remains burdensome due to the extensive use of dense optical flow and loopy belief propagation in the inference.

C. Online *Video Pose Estimation*

There are relatively fewer works addressing the problem of online pose estimation from *continuous* video streams. Lim *et al.* [28] proposed an *online* algorithm to jointly segment a person from the background and estimate the upper-body pose from a video. Weiss *et al.* [10] presented a Dynamic Structured Model Selection method (MODEC-S) based on their MODEC method [1] that uses meta features in structured learning to automatically determine models to choose for the inference. In general, existing methods in this category have difficulties processing real-time video streams due to the extensive use of features such as dense optical flow, or otherwise specific hardware such as the GPU are required. To our best knowledge, the development of real-time *online* video pose estimation from RGB videos is still an open problem, and the demand of such development continuous to grow.

D. Deep Learning Based Pose Estimation

Pose estimation using deep neural networks (DNN) has shown superior performance in recent years [4], [19], [20],

[21], [22], [23], [29], [30], [31], due to the availability of larger training datasets and powerful GPUs. The DeepPose by Toshev *et al.* [4] was an initial work, where a cascade of DNNs was applied to pose estimation. However this method does not allow useful structure information from the articulated poses to be taken into consideration. Tompson *et al.*'s unified learning framework [19] combined convolutional network with a part-based spatial-model, where structural constraints such as geometric relationships between body joint locations can be exploited. Following this trend, additional deep learning works [20], [21], [22], [23] focused on leveraging motion flow in the videos, where image-based matching were augmented with optical flow in the matching of temporally distant frames. Recent breakthrough was on the dedicate modeling of human body part structure. Chu *et al.* [29] proposed a bi-directional tree-structured model to optimize the inter-relationships between highly correlated joints. Wei *et al.* [30] learned image features and image-dependent spatial models in a pose machine framework. Carreira *et al.* [31] applied a rich structured representation by introducing a top-down feedback step to expand the expressive power of the hierarchical feature extractors. Jain *et al.* [32] used a convolutional neural network to incorporate both the color and motion features into video pose estimation. In summary, deep learning methods are generally superior in feature presentation and learning. However the searching of a good representation to effectively incorporate structural information into the inference process (which resembles the utility of the pose mode transitions in our work) is still a research question.

To sum up image based pose estimation methods do not efficiently use the temporal information, pose estimation result can not use the relations between the poses at different frames. On the other hand the offline video pose estimation methods are usually time consuming because the use of optical flow, and the offline setting makes it hard to be generalized to the realistic application. Further more, recently deep learning base methods require specific hardware like GPUs, which limits the deployment of these method in mobile platforms. To address these problem, we extend the image based pose estimation method by leveraging the temporal correlation of articulated pose between frames and optimize the code to speed up the inference. Specifically we construct a conditional dynamic Bayesian network to model the pose mode-mode transition over a time window, and use a simple softmax network to infer the state-mode transition.

III. Time-Windowed Conditional Dynamic Bayesian Network (T-CDBN)

We first formulate the time-windowed conditional dynamic Bayesian network (T-CDBN) model in a general context. We then derive a specific T-CDBN formulation for online video pose estimation, and then discuss the learning and inference of the model in Section IV. Specifically, video pose estimation can be regarded as the process of determining the unknown "states" (in our case the poses in terms of joint locations) by augmenting a dynamic process to determine the statistical patterns of mixed " modes" (of the states) in a time series. Once the latent modes are modeled statistically, the states will be determined based on the inferred modes. The T-CDBN explicitly models the dynamic evolution of the mode-to-mode transitions within a specific time window, as shown in Fig. 1.

We consider the following dynamic model that involves three time-varying variables: *(i)* an input sequence of observation variables $\mathbf{x}_{0:t}$, *(ii)* an output temporal sequence of latent state variables $\mathbf{y}_{0:t}$, and *(iii)* the sequence of latent mode estimation variables $z_{0:t}$. We use each boldface variable to represent a vector or matrix. We use subscript $0:t$ to represent a set of variables ranging from the index of 0 to t. Each $z_\delta \in \{1, \cdots, M\}$ is a scalar indicating one of the M modes that is active at time δ. We use a *time window* to simplify our model, *i.e.* we consider only up to k previous frames for their influence on the current mode estimation. The T-CDBN model represents the dependencies of these variables with a dynamic probabilistic model, which corresponds to a factorization of probability distribution $p(z_{0:t}, \mathbf{y}_{0:t}|\mathbf{x}_{0:t})$ according to the graphical structure in Fig. 1(a), as:

$$p(z_{0:t}, \mathbf{y}_{0:t}|\mathbf{x}_{0:t}) = p(z_0|\mathbf{x}_0) \prod_{\delta=0}^{t} p(\mathbf{y}_\delta|z_\delta, \mathbf{x}_\delta) \quad (1)$$
$$\cdot\ p(z_1|z_0, \mathbf{y}_0, \mathbf{x}_0) \cdot p(z_2|z_{0:1}, \mathbf{y}_1, \mathbf{x}_1) \cdot p(z_3|z_{0:2}, \mathbf{y}_2, \mathbf{x}_2)$$
$$\cdots \cdot p(z_t|z_{t-k:t-1}, \mathbf{y}_{t-1}, \mathbf{x}_{t-1}).$$

The joint model $p(z_{0:t}, \mathbf{y}_{0:t}|\mathbf{x}_{0:t})$ can be decomposed as shown in Eq.(1) to form the basis for dynamic Bayesian inference. As an example of probabilistic graphical model, T-CDBN can be regarded as a conditional random field (CRF), but the output and latent mode variables $\mathbf{y}_t$ and z_t are conditioned on the input observation $\mathbf{x}_t$ from a dynamic Bayesian network (DBN).

We will show in the next subsections that Eq.(1) can be decomposed into two categories of conditional distributions and be solved accordingly in the T-CDBN framework: *(i)* conditional distributions $p(\mathbf{y}_t|z_t, \mathbf{x}_t)$ and $p(z_t|\mathbf{x}_t)$ which concern the inference using variables of the same time index, as such they form the **inference modules** (Section III-A); *(ii)* conditional distributions $p(z_t|z_{t-i})$, $i = 1, ..., k$ and $p(z_t|\mathbf{y}_{t-1})$ which describe the correlations of variables across time steps, as such they form the **dynamic modules** in the T-CDBN model (Section III-B).

A. Online Inference of the T-CDBN Model

The online inference of the T-CDBN model corresponds to the computation of posterior distribution of the output given the input, *i.e.*, $p(\mathbf{y}_t|\mathbf{x}_{0:t})$, which is obtained by expanding all possible modes z_t as:

$$p(\mathbf{y}_t|\mathbf{x}_{0:t}) = \sum_{z_t=1}^{M} p(z_t, \mathbf{y}_t|\mathbf{x}_{0:t}). \quad (2)$$

We can further expand $p(z_t, \mathbf{y}_t|\mathbf{x}_{0:t})$ by applying the Markovian properties in the joint model as

$$p(z_t, \mathbf{y}_t|\mathbf{x}_{0:t}) = p(\mathbf{y}_t|z_t, \mathbf{x}_t) p(z_t|\mathbf{x}_{0:t}). \quad (3)$$

The first term in Eq.(3) corresponds to the estimation of the output variable $\mathbf{y}_t$ given the current input $\mathbf{x}_t$ and latent mode

z_t. The second term is the mode estimation from the input $\mathbf{x}_{0:t}$. Under the *time window* assumption, the second term in Eq.(3) $p(z_t|\mathbf{x}_{0:t}) = p(z_t|\mathbf{x}_{t-k:t})$ can be further expanded as:

$$\begin{aligned} p(z_t|\mathbf{x}_{t-k:t}) &= \sum_{z_{t-k}} \cdots \sum_{z_{t-1}} \int_{\mathbf{y}_{t-1}} p(z_t, z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t}) d\mathbf{y}_{t-1} \\ &= \sum_{z_{t-k}} \cdots \sum_{z_{t-1}} \int_{\mathbf{y}_{t-1}} p(z_t|z_{t-k:t-1}, \mathbf{y}_{t-1}, \mathbf{x}_{t-k:t}) \\ &\quad \cdot \; p(z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t-1}) d\mathbf{y}_{t-1}, \end{aligned} \tag{4}$$

where $p(z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t}) = p(z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t-1})$ by safely dropping the $\mathbf{x}_t$ term from the previous frames. Eqs.(3-4) provide the recursive Chapman-Kolmogorov update of the posterior distribution $p(z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t})$ over the modes, from which we can build the dynamic inference algorithm for T-CDBN.

We further introduce the concept of *weighting decay* for the impacts and relevancy of the previous modes $z_{t-k:t-1}$ in the time window, such that frames closer to the current frame t are assigned with larger weights. Specifically,

$$p(z_t|z_{t-i}) = \omega_i \cdot p(z_t|z_{t-1}), i = 1, ..., k \tag{5}$$

where ω_i is a *weighting decay* parameter for the previous i frames. We normalize all weighting factors such that $\sum_{i=1}^{k} \omega_i = 1$. [2] The first integral term in Eq.(4) is then:

$$\begin{aligned} & p(z_t|z_{t-k:t-1}, \mathbf{y}_{t-1}, \mathbf{x}_{t-k:t}) \\ & \propto p(z_t|\mathbf{x}_{t-k:t}) \cdot p(z_t|\mathbf{y}_{t-1}) \cdot p(z_t|z_{t-k:t-1}) \\ & = p(z_t|\mathbf{x}_t) \cdot p(z_t|\mathbf{y}_{t-1}) \cdot p(z_t|z_{t-k:t-1}) \\ & = p(z_t|\mathbf{x}_t) \cdot p(z_t|\mathbf{y}_{t-1}) \cdot \sum_{i=1}^{k} \omega_i \cdot p(z_t|z_{t-i}). \end{aligned} \tag{6}$$

The first line of Eq.(6) is based on an assumption that the multiple $z, \mathbf{y}, \mathbf{x}$ terms on the right side are conditionally independent. The $\propto$ is to indicate the skipping of a constant term. The second line of Eq.(6) is derived based on the Markovian assumption that the $\mathbf{x}$ in the history is not considered according to the graph in Fig. 1. The third line of Eq.(6) is derived based on our newly proposed time window formulation, that the $p(z|z_{t-i})$ terms are treated as independent such that they can be weighted summed by ω_i. k is the time window, $p(z_t|\mathbf{x}_{t-k:t}) = p(z_t|\mathbf{x}_t)$ according to the frame-based conditional dependency assumption.

Finally, according to the model graph in Fig. 1, the T-CDBN mode estimation, which involves a complete formulation of $p(z_t|x_{t-k:t})$ in Eq.(4) and Eq.(6), can be organized as:

$$\begin{aligned} p(z_t|\mathbf{x}_{t-k:t}) \propto \; & \underbrace{p(z_t|\mathbf{x}_t)}_{\text{observation-mode est.}} \cdot \sum_{z_{t-k}} \cdots \sum_{z_{t-1}} \underbrace{p(z_t|z_{t-k:t-1})}_{\text{mode-mode est.}} \\ & \cdot \int_{\mathbf{y}_{t-1}} \underbrace{p(z_t|\mathbf{y}_{t-1})}_{\text{state-mode est.}} \cdot \underbrace{p(z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t-1})}_{\text{previous posterior}} d\mathbf{y}_{t-1}. \end{aligned} \tag{7}$$

[2]The case of $i = 1$ in Eq.(5) indicates that $\omega_i = 1$ as the initial condition. However we enforce the normalization constrain $\sum_{i=1}^{k} \omega_i = 1$ in the remaining steps. Section V-C shows the empirical values we used to determine the initial values of ω_i (and not the values after normalization) to avoid excessive use of symbols.

In summary, Eqs.(2, 3, 7) show that the T-CDBN model can be completely specified with the following four conditional distributions given the observations, which corresponds to the four types of arcs illustrated in colors in Fig. 1(a):

- **state estimation** $p(\mathbf{y}_t|z_t, \mathbf{x}_t)$: conditional probability distribution of the current state given the current observation and mode;
- **observation-mode estimation** $p(z_t|\mathbf{x}_t)$: conditional probability distribution of the current mode given the current observation;
- **mode-mode transition estimation** $p(z_t|z_{t-i})$, $i = 1, ..., k$: conditional probability distributions of the current mode given previous modes in the time window;
- **state-mode estimation** $p(z_t|\mathbf{y}_{t-1})$: conditional probability distribution of the current mode given previous state.

A straightforward implementation of Eq.(7) is challenging due to the need to integrate over the space of state variable $\mathbf{y}_{t-1}$, which usually does not afford a closed form efficient numerical procedure. In the next section, we solve it by adopting a particle filter approach, where the posterior distribution of $p(z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t-1})$ is approximated with weighted samples of $\mathbf{y}_{t-1}$ as the particles.

B. Dynamic Update of T-CDBN Using Particle Filtering

We derive the dynamic update of the posterior term $p(z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t-1})$ in Eq.(7) using particle filtering. In principle, we should use multiple particles from $p(z_{t-k:t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t-1})$. But here we use a simpler particle generation scheme to achieve maximum running efficiency. Specifically, we follow the strategy of *particle filter with posterior mode tracking* (PT-MT) [33], and use only *one* particle $\phi(z_t)$ that corresponds to one local maximum of the posterior distribution to represent the continuous output variable $\mathbf{y}_t$ for each value of the latent mode z_t with unnormalized weight $\psi(z_t)$. [3] We denote $p(z_t)^{(1:M)}$ the vector of the probabilities of all M modes at time t, that:

$$p(z_t)^{(1:M)} = \begin{bmatrix} p(z_t = 1) \\ p(z_t = 2) \\ \cdots \\ p(z_t = M) \end{bmatrix}. \tag{8}$$

We denote $p(\mathbf{Z}_{t-k:t}) = p(z_t)_{t-k:t}^{(1:M)}$ a $M \times (k+1)$ matrix containing probabilities of all modes in the time window from $t-k$ to t, such that:

$$p(\mathbf{Z}_{t-k:t}) = \left[p(z_t)_{t-k}^{(1:M)}, p(z_t)_{t-k+1}^{(1:M)}, \; \cdots, p(z_t)_{t}^{(1:M)} \right]. \tag{9}$$

Particle $\phi(z_t)$ is obtained using:

$$\begin{aligned} \phi(z_t) &= \underset{\mathbf{y}_t}{\operatorname{argmax}} \, p(\mathbf{Z}_{t-k:t}, \mathbf{y}_t|\mathbf{x}_{t-k:t}) \\ &= \underset{\mathbf{y}_t}{\operatorname{argmax}} \, p(\mathbf{y}_t|\mathbf{Z}_{t-k:t}, \mathbf{x}_{t-k:t}) \cdot p(\mathbf{Z}_{t-k:t}|\mathbf{x}_{t-k:t}) \\ &= \underset{\mathbf{y}_t}{\operatorname{argmax}} \, p(\mathbf{y}_t|z_t, \mathbf{x}_t) \cdot p(\mathbf{Z}_{t-k:t}|\mathbf{x}_{t-k:t}) \\ &= \underset{\mathbf{y}_t}{\operatorname{argmax}} \, p(\mathbf{y}_t|z_t, \mathbf{x}_t). \end{aligned} \tag{10}$$

[3]We can certainly use several particles corresponding to different local maximums of the posterior distribution, but with one particle we can obtain a good balance between performance and running time requirement.

Algorithm 1 Dynamic inference for T-CDBN at time t

Input: Observation $\mathbf{x}_{t-k:t}$, mode transition probability matrix $p(\mathbf{Z}_{t-k:t})$

Output: State prediction $\mathbf{y}_t^*$

1: Compute mode probability $p(z_t|\mathbf{x}_{t-k:t})$ using Eq.(12) at time step t for all modes.
2: For each mode z_t, compute state $p(\mathbf{y}_t|z_t, \mathbf{x}_t)$ using Eqs.(10)-(11).
3: Output the most probable state $\mathbf{y}_t^* = \underset{\mathbf{y}_t}{\operatorname{argmax}}\, p(\mathbf{y}_t|z_t, \mathbf{x}_t)$.

Steps 2-3 in Eq.(10) are derived based on the condition that $\mathbf{y}_t$ only depends on $\mathbf{x}_t$ and z_t. Steps 3-4 are obtained based on the fact that the second term in step 3 is a constant. Eq.(10) aims to solve for $\phi(z_t)$ and the most probable unknown $\mathbf{y}_t$ from the argmax. The resulting probability $p(\mathbf{y}_t|z_t, \mathbf{x}_t)$ can be regarded as a confidence of the state estimation.

Particle weight $\psi(z_t)$ is calculated similar to Eq.(10) but using max instead of argmax, and replacing $\mathbf{y}_t$ with $\phi(z_t)$:

$$\begin{aligned}\psi(z_t) &= \max_{\mathbf{y}_t} p(\mathbf{Z}_{t-k:t}, \mathbf{y}_t|\mathbf{x}_{t-k:t}) \\ &= p(\mathbf{Z}_{t-k:t}, \phi(z_t)|\mathbf{x}_{t-k:t}) \\ &= p(\mathbf{Z}_{t-k:t}|\mathbf{x}_{t-k:t}) \cdot p(\phi(z_t)|z_t, \mathbf{x}_t).\end{aligned} \tag{11}$$

The first term of Eq.(11) can be expanded based on that $\phi(z_t)$ only depends on z_t and $\mathbf{x}_t$ and the conditional simplifications of $p(z_t|z_{t-1})$, $p(z_t|z_{t-2})$, ... , $p(z_t|z_{t-k})$ from the graph structure in Fig. 1(a), that:

$$p(\mathbf{Z}_{t-k:t}|\mathbf{x}_{t-k:t}) = p(z_t|\mathbf{x}_t) \cdot \prod_{i=1}^{k} p(z_{t-i}|\mathbf{Z}_{t-k:t-i}, \mathbf{x}_{t-k:t-i}).$$

In other words, for each possible value of the mode variable z_t, we represent the posterior distribution $p(\mathbf{y}_t|z_t, \mathbf{x}_{t-k:t})$ with a particle-weight pair $(\phi(z_t), \psi(z_t))$, which corresponds to one local maximum of $p(\mathbf{y}_t|z_t, \mathbf{x}_{t-k:t})$. Using the particle filter approach, Eq.(7) is approximately computed with

$$\begin{aligned}p(z_t|\mathbf{x}_{t-k:t}) \;\approx\; & p(z_t|\mathbf{x}_t) \sum_{i=1}^{k} \omega_i \cdot p(z_t|z_{t-i}) \\ & \cdot\; p(z_t|\phi(z_{t-1})) \frac{\psi(z_{t-1})}{\sum_{z'_{t-1}} \psi(z'_{t-1})},\end{aligned} \tag{12}$$

where the domain of z'_{t-1} is all the M possible modes at a single time. Eq.(12) is then combined with Eq.(3) to recursively find $\phi(z_t)$ and $\psi(z_t)$. We use $\operatorname{argmax}_{\mathbf{y}_t} p(\mathbf{y}_t|\mathbf{x}_{t-k:t})$ to obtain the optimal estimation of the output state $\mathbf{y}_t^*$, based on the calculation of the posterior distribution $p(z_{t-1}, \mathbf{y}_{t-1}|\mathbf{x}_{t-k:t-1})$. Algorithm 1 summarizes the online inference and dynamic update of T-CDBN.

IV. Online Video Upper-Body Pose Estimation Using T-CDBN-MODEC

We apply T-CDBN to *online* video pose estimation, where the observation $\mathbf{x}_t$ corresponds to image features, and the latent state variable $\mathbf{y}_t$ corresponds to body joint locations (shoulder, elbow and wrist) in each video frame t. Latent mode variable z_t corresponds to the *pose mode* that clusters similar poses into groups, as depicted in Fig. 2.

According to the T-CDBN framework described in Section III, we need to specify the **inference module** (*i.e.*, conditional distributions $p(\mathbf{y}_t|z_t, \mathbf{x}_t)$ and $p(z_t|\mathbf{x}_t)$) and the **dynamic module** (*i.e.*, conditional distributions $p(z_t|z_{t-i})$, $i = 1, ..., k$ and $p(z_t|\mathbf{y}_{t-1})$). Because the inference module only relies on variables from a single frame, in principle, we can use any single image pose estimation method that provides clustering of poses into explicit "modes".We choose the MODEC model [1] as the basis for the inference module, where pose mode $p(z_t|\mathbf{x}_t)$ is computed by matching HOG pyramid features. Section IV-A reviews the MODEC method and specifically focuses on the CRF pose estimation with known mode prior *i.e.* the estimation of $p(\mathbf{y}_t|z_t, \mathbf{x}_t)$. Section IV-B describes how the two dynamic modules $p(z_t|z_{t-i})$, $i = 1, ..., k$ and $p(z_t|\mathbf{y}_{t-1})$ are learned in a data-driven approach in our T-CDBN framework.

A. Review of MODEC Image-Based Pose Estimation

We briefly review Sapp and Taskar's MODEC single image-based upper-body pose estimation [1]. Given an input image, MODEC relies on a human torso detector [15] to initialize and localize any existing upper-body in the image. This initialization step can be replaced with a fast face detector with a conversion from the head bounding box to a torso box. With a known torso box, MODEC exploits the symmetry of the human body, and only perform inference on the left-half of the upper-body. Poses of the right-half of the body are recognized by mirroring and applying the models obtained from the left-half body. Two sets of HOG feature pyramids are computed in MODEC accordingly: *(i)* A coarse *side-model* HOG pyramid is constructed to determine the pose modes z_t (out of M possible modes). The training of the side-model is performed by a K-means clustering of left body poses into $M = 32$ pose modes, where the pose in the cluster center is used to calculate the HOG side-model features. *(ii)* Seven *part-models* (HOG pyramids) are constructed to further refine the joint locations (head, neck, shoulder, upper-arm, elbow, lower-arm, and wrist). Both the side-model and part-models are learned automatically from labeled data. In the MODEC cascaded inference, pose mode determined from the side-model is used as a prior to guide the inference of the part-models to determine refined joint locations. The inference of the part-models is performed using a single-path tree-based CRF, which allows for efficient parallel implementations. [4]

The incorporation of the single-image MODEC CRF pose estimation into our T-CDBN framework is a major advantage to model pose mode transitions for video pose estimation. Instead of determining the mode from a single image, our method retains a robust conditional dynamic modeling of the modes, which provides key knowledge to improve performance. Specifically, in our T-CDBN framework, the likelihood of all pose modes in the k-frame time window are jointly considered in the three types of arcs in the graphical model in

[4]Matlab implementation of MODEC is available at http://vision.grasp.upenn.edu/cgi-bin/index.php?n=VideoLearning.MODEC.

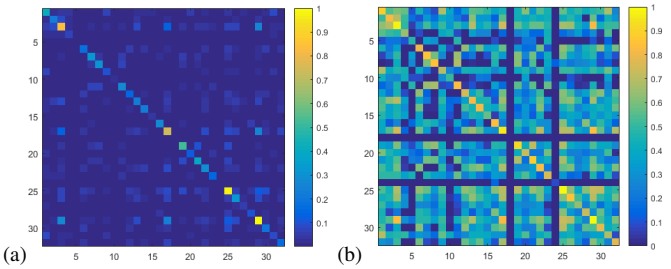

Fig. 3. *(a)* **Mode transition probability matrix** $\mathcal{T}$ *learned from the FLIC dataset, where the transition probability from the* n*-th to the* m*-th mode is shown at* $\mathcal{T}[n,m]$ *with color code. (b) The* log *matrix* $\mathcal{T}_l = \log(\mathcal{T})$ *is less sparse and thus more effective for the prediction of mode-mode transitions in T-CDBN.*

Fig. 1 to infer the conditional distribution of mode estimation $p(z_t|z_{t-k:t-1}, \mathbf{y}_{t-1}, \mathbf{x}_{t-k:t-1})$. (see Section III-B for more details). We will show in Section V that the T-CDBN-MODEC effectively captures motion consistency and performs well especially for high frame rate videos.

B. Learning of Dynamic Pose Mode Prediction

We describe the learning of the two dynamic modules, *i.e.*, the mode-mode transition estimation $p(z_t|z_{t-i})$, $i = 1, ..., k$ and the state-mode estimation $p(z_t|\mathbf{y}_{t-1})$ of T-CDBN from labeled data and their inference.

Determining groundtruth pose modes. Since most pose estimation datasets provide labels of poses in joint coordinates, we must convert the pose labels into mode labels. We obtain the groundtruth mode $\hat{z}$ by mapping the groundtruth pose $\hat{\mathbf{y}}$ into their closest mode cluster (out of the $M = 32$ possible modes as shown in Fig. 2). We denote $\tilde{\mathbf{y}}^{(z)}$ the representative pose at the cluster center of the z-th mode in the training set. Specifically,

$$\hat{z} = \underset{z=1,..,M}{\operatorname{argmin}} \|\hat{\mathbf{y}} - \tilde{\mathbf{y}}^{(z)}\|_2, \tag{13}$$

where $\|.\|_2$ denotes the ℓ_2 norm for the comparison of joint locations.

Learning the mode-mode transition prior $p(z_t|z_{t-i})$, $i = 1, ..., k$. The pose mode transition prior from a previous frame $t - i$ to the current frame t, *i.e.*, $p(z_t|z_{t-i})$ can be expressed in a matrix $\mathcal{T}_i$ of size $M \times M$ for each frame i in the time window under consideration, where $\mathcal{T}_i[r, c]$ stores $p(z_t = c|z_{t-i} = r)$ at the r-th row and c-th column of the matrix $\mathcal{T}_i$. The learning of $\mathcal{T}_i$ thus requires a large enough labeled dataset; otherwise the sparsity of $\mathcal{T}_i$ can significantly hinder its usability. To this end, we make an simple but effective assumption, that all of the inter-frame conditional probabilities $p(z_t|z_{t-i})$ for $i = 1, ..., k$ can be represented as a single matrix $\mathcal{T} = p(z_t|z_{t-1})$ from the immediate previous frame $i = 1$. For previous frames $i = 2, ..., k$, we apply gradually decaying weights that mimic the decaying weights $\{\omega_i\}$ defined in Eq.(6) in Section III-A. This allows us to learn a denser $\mathcal{T}$ from a small dataset using a simple voting scheme. Specifically,

$$p(z_t = c|z_{t-i} = r) = \frac{p(\hat{z}_t = c, \hat{z}_{t-i} = r)}{p(\hat{z}_{t-i} = r)}, \tag{14}$$

for $r, c \in \{1, ..., M\}$ and frames $i = 1, ..., k$. The weighting decay is calculated from the voting weight ω_i for all frames i in the window. Specifically, for each labeled mode-mode transition from $\hat{z}_{t-1} = r$ to $\hat{z}_t = c$, we increase the transition frequency count of the position $[r, c]$ of the voting matrix $\mathcal{T}_v$ by 1. For any larger frame differences (*i.e.*, $2 \leq i \leq k$, the frequency count is increased by a smaller number $\omega_i = 0.9^i$, which is identical to the definition of the weighting decay parameters in Sections III-A and V-C. We normalize each row of the voting matrix $\mathcal{T}_v$ to obtain the mode-mode conditional probability distribution $\mathcal{T}$. In practice, we further take the logarithm of $\mathcal{T}$, *i.e.*, $\mathcal{T}_l = \log(\mathcal{T})$ to further reduce the sparsity of the matrix. Fig. 3 compares $\mathcal{T}$ and $\mathcal{T}_l$ in terms of their sparsity to demonstrate the effectiveness of this step. At run time, we use $\mathcal{T}_l$ to estimate $p(z_t|z_{t-1})$. For larger frame differences $2 \leq i \leq k$, we estimate $p(z_t|z_{t-i})$ by multiplying $\mathcal{T}_l$ with the weighting decay again, *i.e.*, $p(z_t|z_{t-i}) = p(z_t|z_{t-1}) \cdot \omega_i$.

Learning state-mode estimation $p(z_t|\mathbf{y}_{t-1})$ **using multi-mode softmax.** Since pose modes are generated from unsupervised clustering, when used in pose estimation, more than one possible pose modes can fit well for a putative pose. To this end, we allow multiple pose modes during the learning and fitting of state-mode estimation $p(z_t|\mathbf{y}_{t-1})$ in T-CDBN. Specifically, we learn $p(z_t|\mathbf{y}_{t-1})$ using a multi-mode softmax classification.

In the training step, for each labeled pose $\hat{\mathbf{y}}$ from the groundtruth set, we calculate the distances between the central pose $\tilde{\mathbf{y}}^{(z)}$ of each of the $M = 32$ mode clusters and the groundtruth $\hat{\mathbf{y}}$, then select the top Q best modes as the groundtruth modes. Empirically we found $Q = 5$ yields plausible performance. We then train Q individual softmax models indexed by $q = 1, ..., 5$, and combine the Q softmax modes together to obtain a probability distribution for the state-mode estimation $p(z_t|\mathbf{y}_{t-1})$ in Eq.(7). Fig. 4 shows the effectiveness of such multi-mode modeling strategy, by comparing the performance of the single-mode ($Q = 1$) *vs.* multi-mode ($Q = 5$) learning on the **VideoPose2** dataset.

To robustly infer $p(z_t|\mathbf{y}_{t-1})$ at run time, we consider the putative poses from all possible M modes at the current time t in the softmax formulation. We denote $\mathbf{y}_t^{(1:M)}$ the concatenation of all possible M poses (obtained from the given mode $z \in \{1 : M\}$) at time t, that

$$\mathbf{y}_t^{(1:M)} = \left(\mathbf{y}_t^{(z=1)}, \mathbf{y}_t^{(z=2)}, ..., \mathbf{y}_t^{(z=M)}\right). \tag{15}$$

Specifically, we solve $p(z_t|\mathbf{y}_{t-1})$ by estimating $p(z_t|\mathbf{y}_t^{(1:M)}, \mathbf{y}_{t-1})$ using multiple softmax regressions to estimate a M-vector of mode probabilities $p(z_t)^{(1:M)} = \left(p(z_t)^{(1)}, p(z_t)^{(2)}, ..., p(z_t)^{(M)}\right)$. The following features are used in the softmax regression:

- *The* ℓ_2 *distance* $d^{(1:M)}$ between each of the M putative poses $\mathbf{y}_t^{(1:M)}$ and the previous pose $\mathbf{y}_{t-1}$ respectively. Specifically, $d^{(1:M)} = \left\|\mathbf{y}_t^{(1:M)} - \mathbf{y}_{t-1}\right\|_2$.
- *Scale difference* between each of the estimated scales $s_t^{(1:M)}$ and s_{t-1} corresponding to $\mathbf{y}_t^{(1:M)}$ and $\mathbf{y}_{t-1}$ respectively, which can be obtained from the resulting level of the MODEC fine-grained HOG pyramid. Specifically, $s^{(1:M)} = \left|s_t^{(1:M)} - s_{t-1}\right|$.

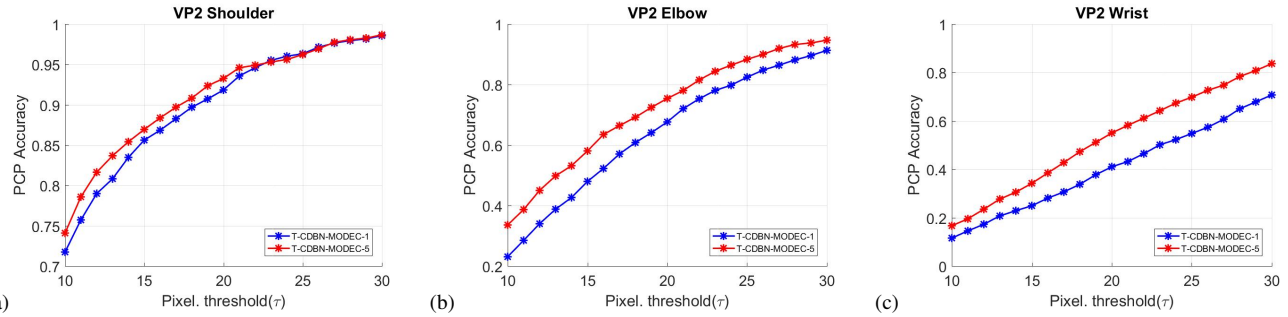

Fig. 4. *Comparison on the use of single-mode ($Q = 1$) vs. multi-mode ($Q = 5$) softmax regression in learning the state-mode estimation $p(z_t|\mathbf{y}_{t-1})$ on the* **VideoPose2** *dataset.*

Algorithm 2 Learning state-mode estimation $p(z_t|\mathbf{y}_{t-1})$ by multi-mode softmax solving for $p(z_t^{(1:M)}|\mathbf{y}_t^{(1:M)}, \mathbf{y}_{t-1})$

Input:

1: $\mathbf{y}_t^{(1:M)}$: M-vector of putative poses from all M modes at the current frame t

2: $\mathbf{y}_{t-1}$: pose estimation from previous frame $t-1$

Output: M-vector of current state-mode probabilities $p(z_t|\mathbf{y}_{t-1}) = p(z_t)^{(1:M)}$ at time t

3: **for** each of the top Q modes, *i.e.*, $q = 1, ..., Q$ **do**

4: $d_q^{(m)} \leftarrow \left\| \mathbf{y}_t^{(z=m)}, \mathbf{y}_{t-1} \right\|_2$, for all $m = 1, ..., M$

5: $s_q^{(m)} \leftarrow \left| s_t^{(z=m)}, s_{t-1} \right|$, for all $m = 1, ..., M$

6: $\alpha_q^{(m)} \leftarrow \left| \beta_t^{(z=m)}, \beta_{t-1} \right|$, for all $m = 1, ..., M$

7: $p(z_q)^{(1:M)} \leftarrow \mathbf{softmax}_q \left(d_q^{(1:M)}, s_q^{(1:M)}, \alpha_q^{(1:M)} \right)$

8: **end for**

9: $p(z_t)^{(1:M)} = \frac{p(z_1)^{(1:M)} + p(z_2)^{(1:M)} + ... + p(z_Q)^{(1:M)}}{Q}$

- *Elbow angle difference* $\alpha^{(1:M)}$ between each of the elbow joint angles $\beta_t^{(1:M)}$ and β_{t-1} corresponding to $\mathbf{y}_t^{(1:M)}$ and $\mathbf{y}_{t-1}$ respectively. Specifically, $\alpha^{(1:M)} = \left| \beta_t^{(1:M)} - \beta_{t-1} \right|$.

Algorithm 2 summarizes the softmax regression steps in learning the state-mode estimation probability. The inference of sofmax can be performed following similar steps. We take the M-vector output from the softmax as the final state-mode estimation $p(z_t|\mathbf{y}_{t-1})$.

Our multi-mode regression strategy shares a similar idea of heat-map regression [19] that are leveraged in recent deep learning pose estimation works, which yields smoother mode predicting and improves pose estimation accuracy in video pose estimation.

V. Experimental Results

We evaluate the performance of T-CDBN-MODEC on three major public video pose estimation datasets, and we further created a new high frame rate pose (HFR-Pose) dataset to evaluate the HFR scenario that are not particularly addressed in existing works. We compare the accuracy and efficiency with several recent video pose estimation methods, including both *online* and *offline* ones. In general, *offline* methods perform better than *online* ones, as information from the whole video sequence is leveraged in the pose inference process for individual frames. In comparison, *online* methods generally run faster, as only a few recent frames in a time window are considered at a time.

A. Datasets for Evaluation

We perform evaluation on the following existing datasets (VideoPose2, FLIC, TUM Kitchen) for three main purposes.*(i)* These datasets are large enough to provide sufficient training samples to learn a sufficiently dense pose mode transition matrix $\mathcal{T}$. *(ii)* Their frame rates are high enough that explicit modeling of pose mode transitions are effective. *(iii)* They contain large varieties of real-world human poses which are challenging to recognize. We summarize each dataset in the following.

- The **VideoPose2** dataset [2] consists of video clips from popular TV shows *Friends* and *Lost*. Every other frames of the original video sequences are selected, resulting in videos with an average frame rate of 10 FPS. There are 44 clips of 2-3 seconds in length, with a total of 1,286 frames. To compare with existing works as in [10], we use 26 clips to train the mode transition model, and report performance on the remaining 18 clips.
- The **High Frame Rate Pose (HFR-Pose)** dataset: We collect a new high frame rate upper-body pose dataset using Microsoft Kinect, where the depth information is available however not used in this work. We will make it public once this paper is published. This dataset consists of subjects performing simple actions (*e.g.*, hand-on-hip, touching face, arm crossing) that are suitable for behavior recognition. There are 18 clips of 30 FPS with per-frame poses manually labeled for the positions of head, shoulders, elbows, and wrists. We use 12 clips to train the mode transition model and report performance on the remaining 6 clips.
- The **TUM Kitchen** dataset [25] provides a comprehensive collection of activity sequences recorded using multiple complementary sensors set up in a kitchen environment. It consists of 21 clips taken from four fixed overhead cameras running at 25 Hz, where each camera captures around 36k frames. We use 60% of data for training and the rest for evaluation and testing.
- The **Frames Labeled In Cinema (FLIC)** dataset [1] consists of $5,003$ selected images obtained from Holly-

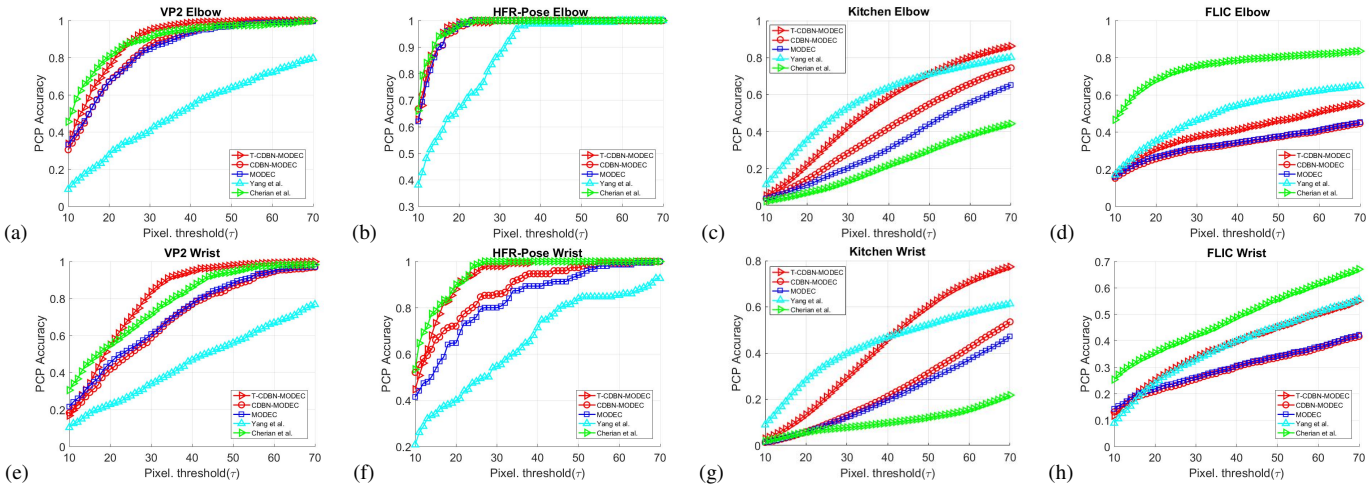

Fig. 5. **Evaluation of T-CDBN-MODEC**, *CDBN-MODEC [24], MODEC [1], Cherian* et al. *[11], and Yang & Ramanan [3] in the PCP metric on the VideoPose2, HFR-Pose, TUM Kitchen, and FLIC datasets. Note that image-based methods [1] and [3] are listed here to show the performance gain of video-based methods that leverage temporal information.*

wood movies. The FLIC dataset was originally prepared for image-based pose evaluation, due to that the selected images are discontinuous 'shots' from the original video. So when the selected images are treated as a video, the frame rate is extremely low. In addition, human body joints in each image are manually annotated for only one selected subjects in a frame (who typically appears more frontal and non-occluded), and the selected subject can switch across frames when there exists multiple subjects. Due to these reasons, the FLIC dataset is less suitable for the evaluation of video-based methods. We include the evaluation results from the FLIC dataset for completeness and to examine the limit of our method. 80% of FLIC data are used for training, and the remaining 20% (1,016 images) are used for evaluation and testing.

- **Penn Action Dataset** [34] (University of Pennsylvania) contains 2326 video sequences of 15 different actions and human joint annotations for each sequence. The annotations consist of action class labels, 2D keypoint positions (13 in all) in each video frame and their corresponding visibilities, and camera viewpoints.

We did not consider several other datasets as they are not suitable for video pose estimation. Specifically, we do not use the MPII Human Pose Dataset [35] as its main purpose is single-image pose estimation. Similarly, we did not use the BBC Pose dataset [36] for that relatively fewer manual groundtruth labels are provided and many labels are generated using a semi-automatic tracker. Last, we did not perform evaluation on the Pose-in-the-Wild dataset [11] due to its low frame rate and lacking of a training set.

B. Evaluation Metric

We use the *percentage of predicted parts* (PCP) [1] as the metric to evaluate the accuracy of pose estimation. For a video sequence of N frames, PCP measures the percentage of frames where the estimated pose $\mathbf{y}$ is close to the groundtruth poses $\hat{\mathbf{y}}$. We evaluate the detected joint locations including the elbows and wrists (which are the most representative and challenging), while skipping the shoulders, arms and head locations although these are available as well. We use j to index joints, and t to index frames. To even out the scaling effects (as the human body can appear to be closer or far away in the image), we normalize the pose estimation difference $\|\hat{\mathbf{y}}_t^j - \mathbf{y}_t^j\|_2$ by first dividing it with the torso box diagonal D_t, and then rescaling to 100 pixels. We keep track of the instances when the normalized detected pose joint error is less than an **accuracy threshold** τ in each frame using an indicator function $1(\mathcal{C})$, which is one if the condition $\mathcal{C}$ is satisfied, and zero otherwise. Finally we normalize the final score *w.r.t.* video sequence length by dividing the counts by the total frame number N and rescale to 100 to show the PCP in percentage:

$$\text{PCP}_j(\tau) = \frac{100}{N}\sum_{t=1}^{N}\mathbf{1}\left(\frac{\|\hat{\mathbf{y}}_t^j - \mathbf{y}_t^j\|_2}{D_t/100} \leq \tau\right). \qquad (16)$$

Note that the PCP reflects more of the 'precision' than the 'recall' part of the metric, as that it considers only the correct detections (within τ) and neglects the erroneous results no matter how they are closer or further from the threshold.

C. Meta-parameters for T-CDBN-MODEC

The T-CDBN-MODEC framework can learn most of the pose estimation parameters in the training phase, where the following meta-parameters can be set uniformly or tuned separately: (1) total number of pose modes M used in the training phase, (2) the time window size k (the corresponding decaying weights ω_i, $i = 1, ..., k$), and (3) the number of modes m considered in the softmax regression. (4) The decaying window type

We performed a series of meta-experiments to investigate the effect of different meta-parameter settings in our method. The ablation study is conducted on the VP2 dataset. And we use $M = 32$ and $m = 5$ throughout this work in not explained specially.

- **Number of pose modes** M: We conduct a series of experiment on a set of M. As it is showed in Fig.

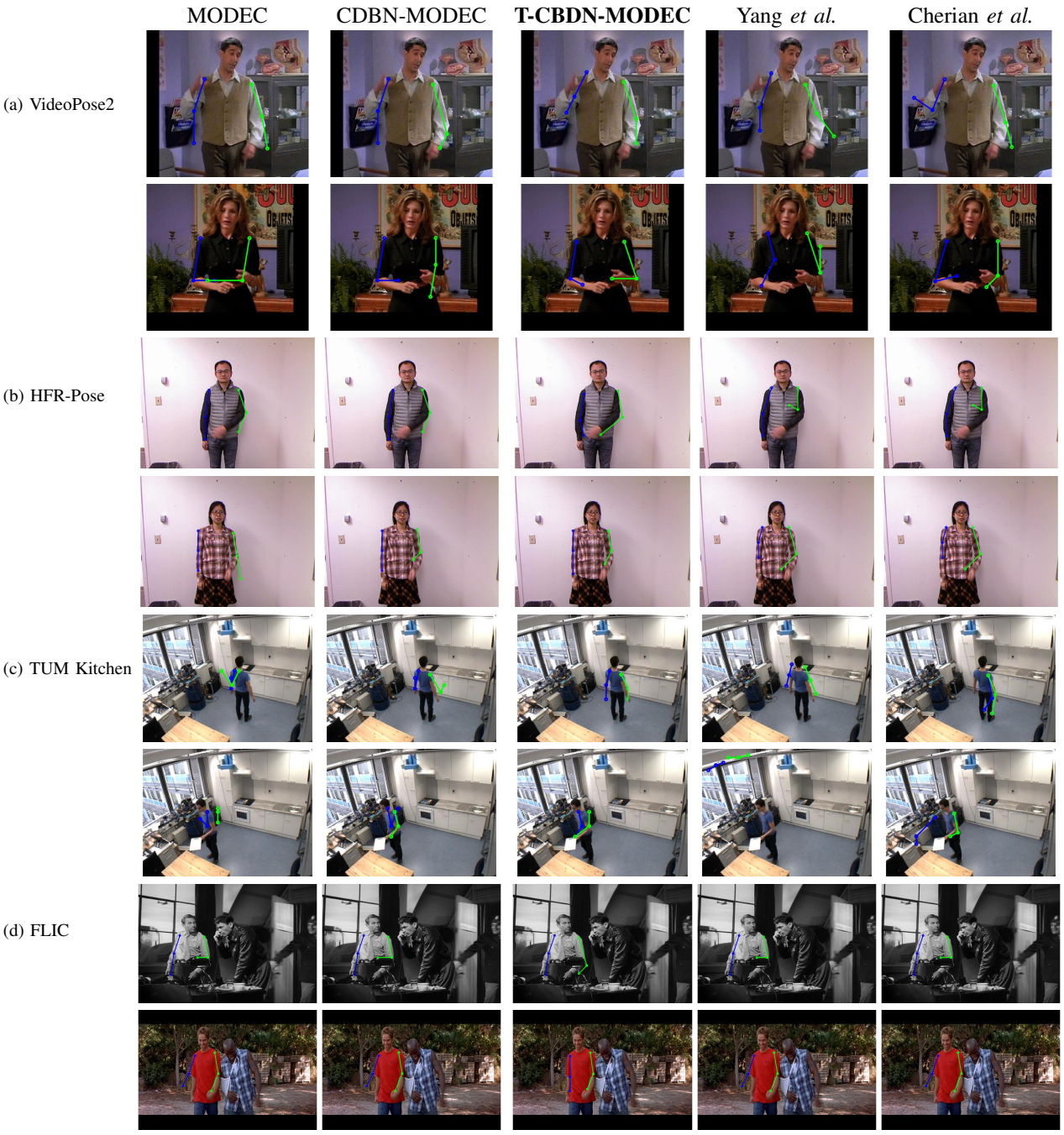

Fig. 6. **Visual result of T-CDBN-MODEC video pose estimation** on the VideoPose2, HFR-Pose, TUM Kitchen, and FLIC datasets.

8 (a), with the increase of M, the performance firstly increases slowly and drop quick very soon, because the mode number M reflect the Hypothesis Space of the estimation result, when M is small the expression ability of the MODEC is limited, however if M is too big the transition probability matrix will be excessive sparse thus introduce too much noise to the dynamic Bayesian graph model and do harm to the performance.

- **Time window size** k: As it is showed in the Fig. 8 (b), the window size of the has a small influence on the performance once the size $k > 4$, and to constrain the computation we tend to use a smaller window size. Also we note that the time window k in T-CDBN should reflect the video frame rate. In our experiments, we use $k = 4$ for all experiments, except that $k = 6$ is used in our HFR-Pose dataset.
- **Softmax Top-m Mode**: We also discuss the least number of modes to be considered in the softmax in Algorithm 2 to speed up the estimation. To determine the number of mode to be considered in softmax state-mode estimation, we conduct a meta-experiment on softmax mode number. As the Fig. 8 (c) shows, top-5 mode for softmax state-

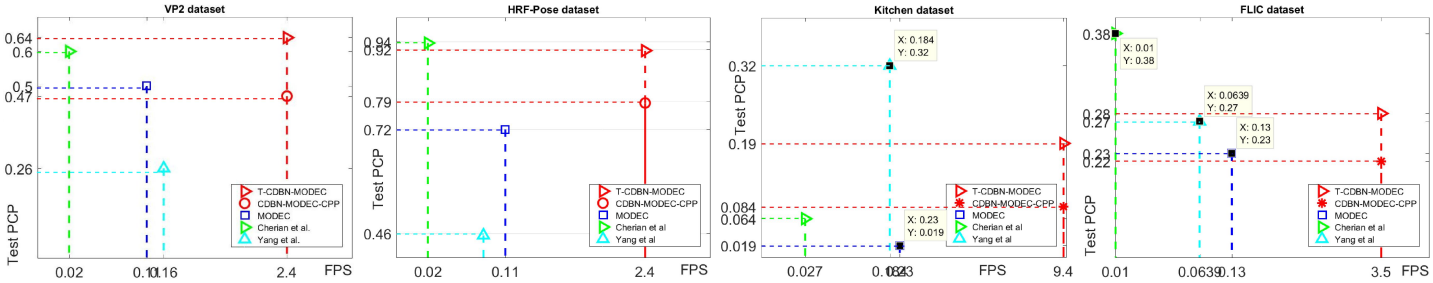

Fig. 7. **Comparison of running time versus accuracy** *for the T-CDBN-MODEC and other methods on the (a) VideoPose2 and (b) HFR-Pose datasets. Method appearing closer to the upper-right corner is considered both accurate and fast. Observe that T-CDBN-MODEC is significantly faster than other compared methods with little or no performance loss.*

mode estimation is sufficient.

- **Decay type**: We use a simple exponential decay for the weighting ω_i of $p(z_t|z_{t-i})$ in Eq.(5) and in Section IV-B, that $\bar{\omega}_i = 0.9^i$ for $i = 1, 2, ..., k$ before normalization. After the normalization of $\omega_i = \frac{\bar{\omega}_i}{\sum_{i=1}^{k} \bar{\omega}_i}$, we obtain $\sum_{i=1}^{k} \omega_i = 1$. We found empirically that the choice of the base number 0.9 has very little impact on pose estimation performance (less than 1.5% accuracy variations). And we also try different type of weight decay method, including exponential decay, none-decay, linear decay, results can be found in Fig. 8 (d), experiment shows that different type of decay have a tiny influence on the result.

D. Comparison with State-of-the-Art Methods

We compare T-CDBN-MODEC with other mainstream pose estimation methods on the four datasets: *(i)* single image pose estimation methods including the original MODEC [1] and Yang & Ramanan [3], *(ii)* the *offline* method of Cherian *et al.* [11], and *(iii)* our previous *online* method of CDBN-MODEC [24]. We note that the comparison of our on-line method to off-line methods is not completely fair, as the off-line method has access to the full video including the future frames, thus the performance is expected to be better.

Fig. 5 shows that T-CDBN-MODEC outperforms both the *online* and single image methods of MODEC and Yang & Ramanan, and is slightly inferior to the *offline* method of Cherian *et al.* Our performance gain is due to the effective modeling of both the between-frame and mode-to-mode correlations. We also compare the T-CDBN-MODEC against our previous work of CDBN-MODEC on the four datasets. We report the $\text{PCP}_j(\tau)$ for all experiments with the threshold τ varying from 10 to 70 pixels.

On our newly created HFR-Pose dataset, we compare T-CDBN-MODEC with MODEC [1] applied to individual frames, our previous work of CDBN-MODEC, and the *offline* method of Cherian *et al.*. In Fig. 5, our T-CDBN-MODEC as an *online* method performs as well as the *offline* state-of-the-art method of Cherian *et al.*, where our method gains considerable acceleration in running time in Fig 7. This is mainly because the HFR-Pose dataset include videos of high frame rates. In HFR-Pose, there are frames which are roughly identical, while in other datasets only keyframes with significant motions are retained. As the method of Cherian *et al.* relies on optical flow in inference, these keyframes can cause problems arisen from weak optical flows.

We emphasize the advantage of our method when applied to real-time on-line video pose estimation, where a sweet spot is achieved in balancing between the performance and the required computational resource. Fig. 7 plots the overall *performance* as the the area under curve (AUC) of the plots from Fig. 5 against the *speed* as the computational frame rate. Observe that T-CDBN-MODEC is more than 10 to 300 times faster (depending on the datasets) than the off-line method of Cherian *et al.* with merely little performance loss, without using specific acceleration hardware such as the GPU.

We note that in Fig. 5(c,g) for the evaluation on the TUM Kitchen dataset, the T-CDBN-MODEC outperforms all comparison methods except for Yang & Ramanan [3] when the threshold τ is relatively small. This is due to the aforementioned bias of the PCP metric in favoring the precision of results. In addition, observe in Fig. 6(c) that the person appears in the TUM Kitchen dataset is relatively small, thus the threshold τ is in fact much stricter, that a few pixels off represents large error in the metric. These explain why most of the methods exhibit low performance in this case.

In Fig. 5(d,h) for the evaluation on the FLIC dataset, T-CDBN-MODEC outperforms all comparison methods, except that it inferior to the offline method of Cherian *et al.*, and in some cases the Yang & Ramanan's method. We list two main reasons. First, the FLIC dataset consists of discontinuous frames, thus the frame rate is extremely low thus temporal continuity is lost. Secondly, there are frequent labeled person switches. Thus only limited extent of temporal consistency can be leveraged, which reduces the performance of our method. In Fig. 9 (left), we compare our method with the state-of-the-art works on Penn Action dataset. It can be observed that while the MODEC and CDBN-MODEC show inferior performance compared with the offline method of Cherian *et al.* and Yang & Ramanan, our T-CDBN-MODEC achieve compititive accuracy on elbow joint and outperform the offline ones on the more challenge wrist joint. Which provide an strong evidence that our dynamic Beyesian Graph model can benefit from the pose relations of the adjacent frames. Fig. 9 (right) shows a comparison of the CDBN method against a stripped down version of a linear dynamic system without hidden state z, which demonstrates the efficacy of the CDBN in handling the dynamic and non-linear nature of human poses and actions.

T-CDBN-MODEC outperforms all comparison methods,

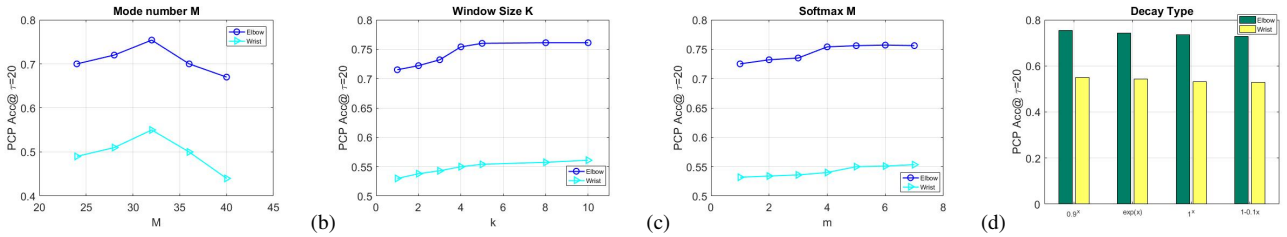

Fig. 8. *Meta-experiment on VP2 dataset. (a) number of Pose Modes M, (b) time window size k, (c) the number of modes M, (d) decaying window type. The experiment evaluate the PCP accuracy at nomalized pixel error 20.*

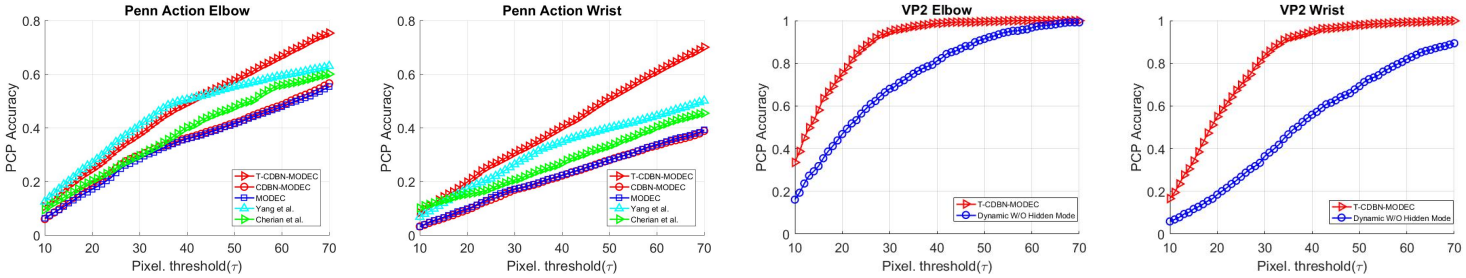

Fig. 9. **Evaluation of T-CDBN-MODEC**, *CDBN-MODEC [24], MODEC [1], Cherian* et al. *[11], and Yang & Ramanan [3] in the Penn_Action datasets. The two figures on the right show the linear dynamic system of our method without hidden state z for comparison*

except that it inferior to the offline method of Cherian *et al.*, and in some cases the Yang & Ramanan's method. We list two main reasons. First, the FLIC dataset consists of discontinuous frames, thus the frame rate is extremely low thus temporal continuity is lost. Secondly, there are frequent labeled person switches. Thus only limited extent of temporal consistency can be leveraged, which reduces the performance of our method.

Fig. 6 presents qualitative and visual results from our method and the comparison state-of-arts on selected consecutive frames from the four datasets. The T-CDBN-MODEC achieves reliable estimations due to the successful modeling of mode transitions. In comparison, single-image based methods cannot take such advantage, thus frequent gross errors are observed. The *offline* method of Cherian *et al.* is robust in detecting poses, however information from the whole video is taken into consideration and the speed is compromised.

E. Run-time Efficiency

We compare the run time efficiency versus accuracy for several methods, including MODEC [1], our previous work of CDBN-MODEC [24], Cherian *et al.* [11], Yang & Ramanan [3] and this work of T-CDBN-MODEC in Fig. 7. We use an C++ implementations of our T-CDBN-MODEC, which contains further optimization using (1) multi-thread programming and (2) adapting levels of HOG pyramids in the inference. Our C++ implementation generally takes 320ms for a pose estimation after parallelization on VP2 and HFR-Pose dataset, where the computation of HOG pyramids from [27] takes about 200ms, and the rest of the steps including feature convolution in the refined HOG layers, inference, back tracking, and the CDBN filtering take only about 120ms. All reported running time of the methods that we have source code (*i.e.*, T-CDBN-MODEC, MODEC, and Cherian *et al.*) are based on a machine with a 3.6GHz processor with 8 cores and 20GB memory. The accuracy is measured with the AUC values of wrist curve at the threshold range of 15 to 30 normalized pixel error. Note that the C++ implementation of T-CDBN-MODEC achieves a 15 to 40 times frame rate on the evaluation dataset when compared with the second fasted method. Notably, on the TUM Kitchen dataset, Yang & Ramanan's method outperforms all recent methods in archiving the highest PCP AUC, which is counter intuitive and not consistent with other experiments. Our interpretation is that the views in the TUM Kitchen dataset are mostly top-down and the backgrounds are relatively cleaner. In contrast, the views in other three datasets are horizontal and the backgrounds are complicated. Also the subjects appear to be smaller in the TUM Kitchen dataset, making their poses harder to determine. Noteably, T-CDBN-MODEC achieves 9.4 FPS in processing speed on the TUM Kitchen dataset, which is 40 times faster than Yang & Ramanan on the TUM Kitchen dataset. The speed up is mostly due to the smaller subject in appearance, which reduces the main cost of computing the HOG pyramid features and the convolution for calculating the body joint locations. [5] On the FLIC dataset, T-CDBN-MODEC is 350 times faster than Cherian *et al.*, while Cherian *et al.* takes more than 15GB peak memory to optimize the pose on the whole sequence.

VI. Conclusion

In this work, we describe a fast online pose estimation method based on the T-CDBN-MODEC model. The proposed algorithm presents competitive pose estimation performance on both accuracy and running speed by two complementary system of conditional dynamic Bayesian network and multi-modal decomposable model. We collect a new high frame rate upper-body pose dataset that better reflects practical scenarios calling for fast *online* video pose estimation. When evaluated on this dataset and the other benchmark datasets, our method

[5] Note that we can further speed up T-CDBN-MODEC on the VideoPose2, HFR-Pose, and FLIC datasets by applying an image pyramid or simply down-sampling the images, to match the performance of the TUM Kitchen experiments. However such engineering efforts are omitted.

outperforms other *online* video pose estimation methods. Our method shows comparable performance to the state-of-the-art *offline* methods, with our method provides significant running time acceleration.

There are a few directions we would like to further improve the current work. First, we can further take advantage of the flexibility of the T-CDBN model and combine it with newly effective single-image pose estimation methods, such as those based on deep neural networks. This will involve developing a mode estimation module that can be derived with the original methods. Second, in our current method, we separate the effect of pose $\mathbf{y}_t$ and mode z_t in the pose estimation inference. We believe more accurate prediction can be obtained by considering them jointly and use a learned predictor from labeled data.

Acknowledgments. This work is partially supported by the following project: US NSF CAREER Award IIS-0953373 (S. Lyu), US NSF Research Grant CCF-1319800, and NSF of China Research Grant No. 61472388 (H. Qi).

REFERENCES

[1] B. Sapp and B. Taskar, "Multimodal Decomposable Models for Human Pose Estimation," in *Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3674–3681. 1, 2, 3, 6, 8, 9, 11, 12

[2] B. Sapp, A. Toshev, and B. Taskar, "Cascaded Models for Articulated Pose Estimation," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 406–420. 1, 3, 8

[3] Y. Yang and D. Ramanan, "Articulated Pose Estimation with Flexible Mixtures-of-Parts," in *Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1385–1392. 1, 3, 9, 11, 12

[4] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," *Computer Vision and Pattern Recognition (CVPR)*, pp. 1653–1660, 2013. 1, 3, 4

[5] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial Structures for Object Recognition," in *International Journal of Computer Vision (IJCV)*, vol. 61, 2005, pp. 55–79. 1, 3

[6] M. Eichner and V. Ferrari, "Appearance Sharing for Collective Human Pose Estimation," in *Asian Conference on Computer Vision (ACCV)*, 2013, pp. 138–151. 1, 3

[7] S. Johnson and M. Everingham, "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation," in *British Machine Vision Conference (BMVC)*, 2010, pp. 12.1–12.11. 1, 3

[8] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Strike a Pose: Tracking People by Finding Stylized Poses," in *Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 271–278. 1, 3

[9] B. Sapp, D. Weiss, and B. Taskar, "Parsing Human Motion with Stretchable Models," in *Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1281–1288. 1, 3

[10] D. Weiss, B. Sapp, and B. Taskar, "Dynamic Structured Model Selection," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 2656–2663. 1, 3, 8

[11] A. Cherian, J. Mairal, K. Alahari, and C. Schmid, "Mixing Body-Part Sequences for Human Pose Estimation," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2361–2368. 1, 3, 9, 11, 12

[12] K. Fragkiadaki, H. Hu, and J. Shi, "Pose from Flow and Flow from Pose," in *Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2059–2066. 1, 3

[13] S. Zuffi, J. Romero, C. Schmid, and M. J. Black, "Estimating Human Pose with Flowing Puppets," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 3312–3319. 1, 3

[14] V. Ferrari, M. Marin-jimenez, and A. Zisserman, "2D Human Pose Estimation in TV Shows," in *Statistical and Geometrical Approaches to Visual Motion Analysis, Volume 5604*, 2009, pp. 128–147. 1, 3

[15] L. Bourdev and J. Malik, "Poselets: Body Part detectors trained using 3D human pose annotations," in *International Conference on Computer Vision (ICCV)*, 2009, pp. 1365–1372. 2, 3, 6

[16] K. P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," in *PhD Thesis*, 2002. 2

[17] J. Lafferty, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *International Conference on Machine Learning (ICML)*, 2001, pp. 282–289. 2

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1097–1105. 2

[19] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 1799–1807. 2, 4, 8

[20] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1913–1921. 3, 4

[21] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos," in *Asian Conference on Computer Vision*, 2014. 3, 4

[22] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *IEEE International Conference on Computer Vision*, 2015. 3, 4

[23] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman, "Personalizing human video pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4

[24] M. Chang, H.Qi, X. Wang, H. Cheng, and S. Lyu, "Fast online upper body pose estimation from video," in *British Machine Vision Conference (BMVC)*, Swansea, England, 2015. 3, 9, 11, 12

[25] M. Tenorth, J. Bandouch, and M. Beetz, "The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition," in *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009*, 2009. 3, 8

[26] X. Chen and A. L. Yuille, "Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 1736–1744. 3

[27] P. F. Felzenszwalb, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," in *Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8. 3, 12

[28] T. Lim, S. Hong, B. Han, and J. H. Han, "Joint Segmentation and Pose Tracking of Human in Natural Videos," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 833–840. 3

[29] X. Chu, W. Ouyang, H. Li, and X. Wang, "Structured feature learning for pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[30] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," *arXiv preprint arXiv:1602.00134*, 2016. 4

[31] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," *arXiv preprint arXiv:1507.06550*, 2015. 4

[32] A. Jain, J. Tompson, Y. Lecun, and C. Bregler, "MoDeep: A Deep Learning Framework Using Motion Features for Human Pose Estimation," *LNCS, Asian Conference on Computer Vision (ACCV)*, vol. 9004, pp. 302–315, 2014. 4

[33] S. Das, A. Kale, and N. Vaswani, "Particle filter with mode tracker (PF-MT) for visual tracking across illumination changes," *IEEE Trans. Image Processing*, 2012. 5

[34] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," pp. 2248–2255, 2013. 9

[35] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 9

[36] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman, "Domain adaptation for upper body pose tracking in signed TV broadcasts," in *British Machine Vision Conference*, 2013. 9