Motion Adaptive Pose Estimation from Compressed Videos

Zhipeng Fan[1] Jun Liu[2*] Yao Wang[1]
Tandon School of Engineering, New York University, Brooklyn NY, USA[1]
Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore[2]
zf606@nyu.edu jun_liu@sutd.edu.sg yw523@nyu.edu

Abstract

*Human pose estimation from videos has many real-world applications. Existing methods focus on applying models with a **uniform** computation profile on **fully decoded** frames, ignoring the **freely-available motion signals** and **motion-compensation residuals** from the compressed stream. A novel model, called Motion Adaptive Pose Net is proposed to exploit the compressed streams to efficiently decode pose sequences from videos. The model incorporates a Motion Compensated ConvLSTM to propagate the spatially aligned features, along with an adaptive gate to dynamically determine if the computationally expensive features should be extracted from fully decoded frames to compensate the motion-warped features, solely based on the residual errors. Leveraging the informative yet readily available signals from compressed streams, we propagate the latent features through our Motion Adaptive Pose Net efficiently Our model outperforms the state-of-the-art models in pose-estimation accuracy on two widely used datasets with only around half of the computation complexity.*

1. Introduction

Human pose estimation has drawn increasing amount of attentions over the years [1, 27, 38, 22, 45, 34, 41, 25]. It has a wild range of applications in action recognition, human computer interactions, AR/VR and robotics. Over the years, there have been growing interests in the pose estimation from videos, in which human dynamics has been faithfully captured compared to still images. In applications like intelligent surveillance camera analysis or imitation learning for robots, thousands of hours videos need to be analyzed by the deep models, which draws attention to more efficient approaches [23, 49, 7] to process the frames.

Directly adopting the state-of-the-art models to perform pose estimation on each frame is sub-optimal as it not only ignores the valuable temporal dynamics embedded across consecutive frames but also results in huge amount of redundant computations. Recent approaches [19, 20, 41, 25] adopt temporal modules to model the temporal dynamics frame by frame, *e.g.* Recurrent Neural Net [19], LSTM models [20, 41] and temporal convolution models [25], etc. However, features from each frame are still extracted independently without considering the natural coherence between neighboring frames.

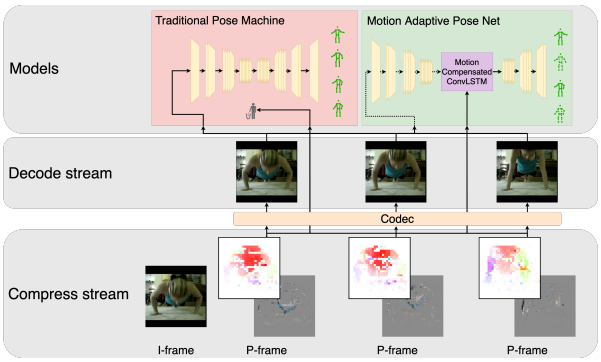


Figure 1. We introduce the Motion Adaptive Pose Net, which exploit the usage of the free of charge yet valuable motion vectors and motion-compensation residuals from compressed streams to dramatically boost the efficiency of the video based pose estimation models. Exploiting the readily available motion and residual information stored in the compressed streams, we obtain state of the art performance with about half of the computation.

On the other hand, video compression techniques relies heavily on the temporal coherence to drastically reduce the size of the videos. The modern standard [30, 29] split the entire videos into group of pictures (**GOP**) and only fraction of frames are encoded in its complete form. For the remaining frames, only sparse motion vectors (**MV**) and residual errors (**R**) are stored. The SOTA pose machines operate on fully decoded frames while constantly ignore the free of charge yet valuable motion field encoded in the compressed streams.

Motivated by the tremendous amount of space savings

*Corresponding author.

brought by the modern video codec, we explore the usage of the compressed streams for efficient pose estimation from videos. The encoded motion field between frames provide valuable insights for how the pose changes across frames while the residual errors provides a direct measurement for the quality of the motion vectors used for compression. Considering the unique properties of these two cost-free representations, we propose an adaptive pose machine that dynamically switch between the light MV-warped features from a ConvLSTM and the accurate features extracted from decoded frames, based on the residual errors. Whenever the model determines motion warped features are reliable, the computationally heavy feature extraction stage is skipped, which offers drastic computation savings.

We validate our models on two widely used datasets and showcase that with the valuable intermediate representations during decoding, we could develop 1) efficient operations that provide relatively reliable features for pose inference as well as 2) fast verification mechanism on the reliability of the morphed features. As the result, our Motion Adaptive Pose Net outperform the previous SOTA models in terms of both the efficiency and the accuracy. In summary, our contributions are as followed:

- We exploit the internal motion signals and residual errors in compressed videos for pose estimation, which are free of cost yet preserve valuable motion information.

- A dynamic model is developed to efficiently utilize the compressed signals, which drastically reduce the computation costs compared to SOTA models.

- We evaluate our proposed model on two widely used datasets: Penn Action [48] and Sub-JHMDB [15]. We outperforms existing methods in both accuracy and efficiency.

2. Related works

2.1. Pose estimation

Most of the existing works on human pose estimation focus on pose estimation accuracy. Recently, a few works study the efficiency of pose estimations in videos, either in terms of the sampling efficiency [3] during training or the inference efficiency [23, 49], which is consistent with our work. However, none of them utilize the freely-available and information-rich representations from compressed videos.

2.1.1 Pose estimation from still images

Traditional methods employ pictorial structures to model the human skeletons [1, 27, 28, 47, 37] like hierarchical tree [35, 37]. More recently, the surge of deep learning took the deep neural net to the center stage. DeepPose [38] introduced a multi-stage network to directly regress the coordinates of joints from the frame, while later works mainly adopt probabilistic heatmap representations to encode the joint positions [22, 43, 45, 5]. An encoder-decoder structure is often employed for deriving joint heatmaps, e.g. hourglass model [22] with a balance encoder and decoder, Simple Baseline [45] with more computations on the encoder side. The HRNet [34, 40] proposed to maintain high resolution feature maps in the model to benefit the pose estimation with higher precision.

Our framework is orthogonal to any single frame pose estimator. We adopted the Simple Baseline [45] as our basis model as it allocate most of the computations to the encoder side, which could be skipped in our framework for certain frames to reduce the computation.

2.1.2 Pose estimation from videos

Optical flows are often employed as the motion clues to derive pose sequences from the videos [33, 26, 4]. However, estimating optical flows is computationally heavy, which incurs additional computations for video based pose estimation. Recurrent Neural Nets (RNN) have also been integrated into pose machines to learn the pose dynamics from data. Those models share a general structure of using CNN to encode every frames sequentially and followed by a temporal model (e.g. RNN[19], LSTM[20], Seq2Seq[6]) to refine the estimations. On the efficiency side, DKD [23] replace the heavy flow estimation module or RNN with a light pose kernel, whereas [49] introduced a frame proposal module to determine a set of keyframes for pose estimation and then interpolate from the estimated key poses.

Our models also focus on the efficiency of pose estimation from video. We introduce the *cost-free* motion vectors to infuse motion signals. While being readily available, the motion vectors are often noisy. Therefore, we dynamically determine whether an accurate feature extraction is required based on the information-rich residual errors. The explicit and rich signal in residual errors allows us to use an extremely light ConvNet, comparing to the ResNet series used in [49], to determine keyframes. The introduced dynamic gate determines the computation profile adaptively, comparing to the interpolation mechanism used in [49].

2.2. Deep learning models on compressed videos

Compressed video formats have only been studied recently in the context of deep learning [44, 32, 2, 12, 11]. CoViAR [44] is one of the pioneers to exploit this modality for action recognition. Three set of models were independently developed on the complete frames, motion vectors and the residuals respectively to derive the actions. Even fewer works exploit the usage of compressed information in the task of object detection [18, 42, 21] and segmentation from videos [14, 36, 8]. To our best knowledge, we are the first work to introduce the compressed streams for efficient

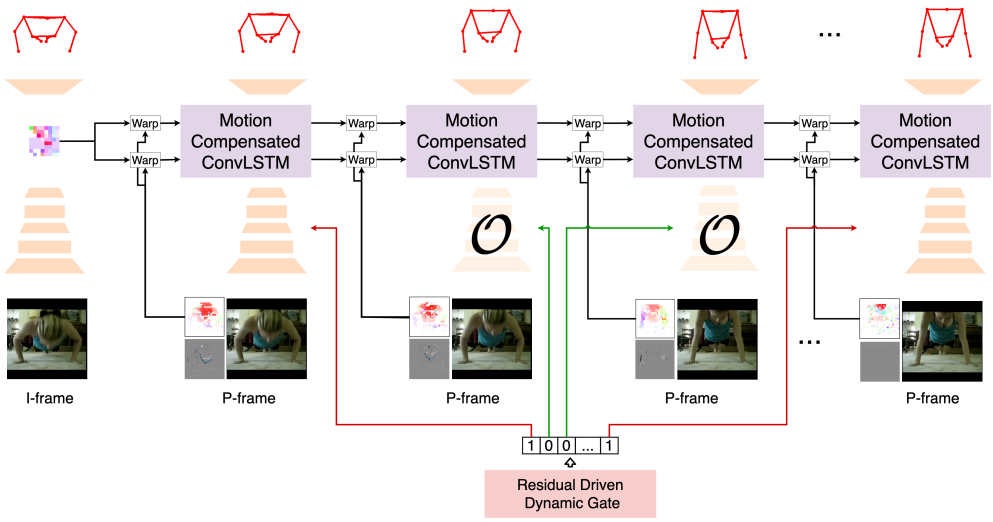


Figure 2. Our pose estimation pipeline consists of two key components: the Motion Compensated ConvLSTM and the Residual Driven Dynamic Gate. The Motion Compensated ConvLSTM warps the cell and hidden state of the ConvLSTM, facilitating better feature alignments w.r.t. the dynamic inputs. The gate adaptively determines if feature extraction could be skipped (marked with $\mathcal{O}$) based on the residual information. The input from the residual maps to the gate are skipped for readability. Both the warping and the gate are much more efficient than running the ResNet for feature extraction and key frame proposal as in [49], therefore, much computations are saved.

pose estimations from videos. Instead of directly contributing to the pose estimation accuracy, motion vectors and residual errors are introduced to construct efficient Motion Compensated features and dynamically determine the computation profiles respectively, which allows us to achieve SOTA accuracy while remain low computation complexity.

3. Methods

3.1. Overview

Our Motion Adaptive Pose Net consists of two key components: the Motion Compensated ConvLSTM and the Residual Driven Dynamic Gate. We will briefly cover the background of video compression in section 3.2 and followed by the Motion Compensated ConvLSTM in section 3.3. We will then introduce the Residual Driven Dynamic Gate in more details in section 3.4. Finally, we explain the loss function and the training strategy in section 3.5.

3.2. Video compression standards

Video compression aims at reducing the bits storing the redundant information that exists across the consecutive frames. The efficient storage and distribution of the videos relies on powerful video codec: H.264/MPEG-4 Part 10/AVC for Advanced Video Coding [30, 29] is one of the commonly used formats for video compression. The key components in H.264 standard are residual coding and block based motion compensation. Motion compensation refers to the technique to warp the previous frames based on the motion information, while residual coding refers to the step of only coding the difference between the warped frame and the actual frame. With a relatively accurate estimation of the motion field, the difference map is sparse and efficient for storage. For the efficiency of encoding, block based motion estimation is used, which assumes all pixels in a block follow the same motion vector with variable block sizes ranging from 4×4 to 16×16. Combining these two techniques, massive bits could be saved from the raw sequences. However, the block simplification results in noisier motion field compared to traditional flows fields, which inspires us to further introduce the dynamic gate mechanism to compensate the errors.

More specifically, the H.264 standard splits the videos into Group of Pictures (GOP), which are further split into intra frame (**I-frame**) and the predictive inter frame (**P-frame**). The I-frames is self-contained and requires more bits to encode. The P-frames, on the other hand, only stores the motion vectors and the residual errors with respect to a previous frame. To decode the P-frames, the codec warps the previous reference I-frame/P-frames F_{t-1} with the motion vector MV_t and then adds the residual errors R_t.

$$\mathrm{F}_t = \mathbf{Warp}(\mathrm{F}_{t-1}, \mathrm{MV}_t) + \mathrm{R}_t \tag{1}$$

3.3. Motion Compensated ConvLSTM

Inspired by ConvLSTM [31, 42], we design our Motion Compensated ConvLSTM *with adaptive inputs*. To assist the learning of temporal dynamics, the free of charge motion vectors are used explicitly to warp the cell and the hidden state of the ConvLSTM to: 1) align the feature maps; 2) decode pose heatmaps. As shown in Fig. 3, we replace the linear layers in the LSTM with convolution layers to accom-

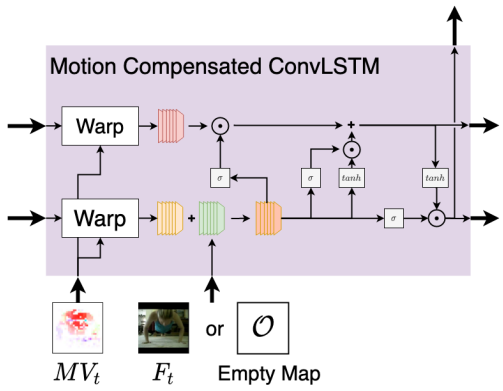


Figure 3. Our Motion Compensated ConvLSTM warps the hidden state and the cell state based on the cost-free Motion Vectors MV_t at each time step. It further takes dynamic inputs (2D features from decoded frames or empty map) to update the cell state following the general LSTM design.

modate the 2D feature maps for decoding the probabilistic pose heatmaps.

Theoretically, the initial state and the input of the Motion Compensated ConvLSTM could come from any single-frame pose estimator. Here we employ the Simple Baseline [45] as our base feature extractor since it retains the heavy computations to the encoder side using pre-trained ResNet models, which could be skipped based on decisions made by our Residual Driven Dynamic Gate. Assuming given a GOP of N frames: $\{\mathrm{F}_0, \mathrm{F}_1, \mathrm{MV}_1, \mathrm{R}_1, ..., \mathrm{F}_{N-1}, \mathrm{MV}_{N-1}, \mathrm{R}_{N-1}\}$, after estimating poses for the I-frame using the Simple Baseline, the ResNet features serve as the initialization for the cell state c and the hidden state h of the ConvLSTM:

$$x_0 = \mathbf{ResNet}(\mathrm{F}_0) \tag{2}$$
$$h_0 = x_0; c_0 = x_0 \tag{3}$$

Denoting the input to the ConvLSTM as x_t, $t \in 1, 2, .., N-1$ the Motion Compensated ConvLSTM executes the following dynamics:

$$h'_{t-1} = \mathbf{Warp}(h_{t-1}, \mathrm{MV}_t) \tag{4}$$
$$c'_{t-1} = \mathbf{Warp}(c_{t-1}, \mathrm{MV}_t) \tag{5}$$
$$i_t = \mathbf{Sigmoid}(\mathbf{Conv}(x_t + h'_{t-1}; w_i) + b_i)) \tag{6}$$
$$f_t = \mathbf{Sigmoid}(\mathbf{Conv}(x_t + h'_{t-1}; w_f) + b_f)) \tag{7}$$
$$o_t = \mathbf{Sigmoid}(\mathbf{Conv}(x_t + h'_{t-1}; w_o) + b_o)) \tag{8}$$
$$g_t = \mathbf{Tanh}(\mathbf{Conv}(x_t + h'_{t-1}; w_g) + b_g)) \tag{9}$$
$$c_t = f_t \odot c'_{t-1} + i_t \odot g_t \tag{10}$$
$$h_t = o_t \odot \mathbf{Tanh}(c_t) \tag{11}$$

We use i_t,f_t,o_t,g_t to denote the input gate, forget gate, output gate and the candidate state following the terminology of LSTM. The output of the ConvLSTM h_t are fed into 3 deconvolutional layers to decode the heatmaps H_t^j of joint j at time t following $H_t^j = \mathbf{M_{deconv}}(h_t)$.

Notice that the input x_t to the ConvLSTM is dynamically determined based on the decision of the Residual Driven Dynamic Gate. We only employ the ResNet to extract features from the current frame F_t following $x_t = \mathbf{ResNet}(\mathrm{F}_t)$ when accurate feature map is deemed necessary, otherwise we set $x_t = \mathcal{O}$.

3.4. Residual Driven Dynamic Gate

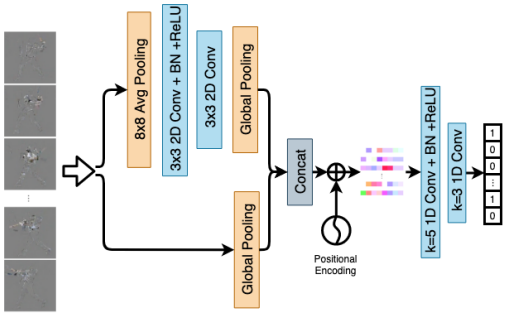


Figure 4. Our dynamic gate makes discrete decisions ($\{0,1\}$) based on all the residual frames within a GOP. The residual frame explicitly measures the difference between the current frame and the motion-compensated previous frame, allowing us to use lightweight gate model here to determine the skipping policy.

Inspired by the fact that the residual errors stores the difference between the warped frame $\mathbf{Warp}(\mathrm{F}_{t-1}, \mathrm{MV}_t)$ and the actual frame F_t, to minimize the computation complexity, we introduce the light gate model based solely on the residual frames to determine the input of the ConvLSTM.

As shown in Fig.4, we first downsample the residual frames to 1/8 of the original size to reduce the computation complexity since we care less about the detail of the difference map. Two consecutive layers of 2D convolutions are applied to the downsampled residual frames for feature extraction. The resulting feature maps are globally pooled to 1D vectors, which are then concatenated with globally pooled raw residual errors. We introduce the Sinusoidal Position Encoding (PE) following [39] to inject position information. The position injected 1D features are then fed into temporal convolution layers to enable the GOP level reasoning. Finally, we generate decisions from the final logits.

Improved Semantic Hashing: We expect "hard" decisions instead of the "soft" attention scores to be able to completely skip the computation on the selected P-frames. However, this kind of binary decisions naturally introduce discontinuity into the models, and therefore prohibit the gradients to be back-propagated to the earlier layers. To address this issue, we adopted the Improved Semantic Hashing techniques, which was first introduced in [16, 17].

During training, we add additional Gaussian noises ϵ with zero mean and standard deviation of 1 to the predicted

logits g from the gate output, which encourages the gate to explore more space randomly. Then two vectors are computed from the noise contaminated logits $g_\epsilon = g + \epsilon$:

$$g_c = \sigma'(g_\epsilon) \text{ and } g_d = \mathbf{1}(g_\epsilon > 0) \tag{12}$$

σ' corresponds to the saturating sigmoid function:

$$\sigma'(x) = max(0, min(1, 1.2\sigma(x) - 0.1)) \tag{13}$$

where σ is the original sigmoid function. Here, the g_c remains **c**ontinuous differentiable almost everywhere while g_d is the binarized **d**iscrete decision with value $\{0, 1\}$, which is non-differentiable.

Following [16, 17], we considers the gradient of g_c with respect to g_ϵ an approximation of the gradients for updating the parameters from the discrete gate g_d. This gradient replacement operation could be realized by $g_d = g_d + g_c - g_c.\text{detach}()$ in PyTorch notations. During training, we randomly blend the continuous gate output g_c and the discrete gate output g_d with equal probabilities. Denote the final output of the gate as g_{mix}, the input to the ConvLSTM becomes:

$$x_t = \text{ResNet}(F_t) \cdot g_{\text{mix}} \tag{14}$$

During inference, we skipped the Gaussian Noise sampling step and directly use the the discrete output, i.e. $g_{\text{mix}} = \mathbf{1}(g > 0)$.

3.5. Training Strategy and loss

Benefiting from the simpleness of the improved semantic hashing, the loss function takes the following form:

$$\mathcal{L} = \frac{1}{Nj} \sum_{t=1}^{N} \sum_{j=1}^{J} (\tilde{H}_t^j - H_t^j)^2 + \lambda \frac{||g_{\text{mix}}||_1}{c} \tag{15}$$

where λ controls the relative weights between the mean squared error on the heatmaps and the second $l1$ error on the activation of the gates. We use the $l1$ term to encourage the sparsity of the activation, leading to fewer frames needed for feature extraction. When different λ is used, the loss balance the trade off between the accuracy and efficiency during training, resulting in models with diverse computation complexities and overall accuracy.

For model training, we first train the pose encoder, Motion Compensated ConvLSTM and the Residual Driven Gate separately and then jointly finetune them. When training the gate independently, we freeze the rest of the models.

4. Experiments

4.1. Dataset and evaluation metrics

Penn Action [48] is a large scale unconstrained video based dataset covering the 15 daily activities of human beings. It contains 2326 video sequences in total, out of which 1258 videos are separated for training and the rest are reserved for testing. The resolutions of the videos are within 640×480 with an average duration of 70 frames. Rich annotations are provided in addition to the action labels, including the 2D poses, human keypoints visibility and bounding boxes.

Sub-JHMDB [15] contains 316 videos of $11,200$ frames in total and 12 different action categories. It provides annotations for 15 body joints along with the puppet flow and mask for each frame. 3 splits are provided for performance estimation. Following the protocol developed by [33, 20, 23, 49], we independently train our models on each split and reports the average performance over the 3 splits.

Following the previous work [33, 20, 23, 49], we adopted the Percentage of the Correct Keypoints (PCK) to evaluate our models. A body joint is considered to be correct if it falls into a range of βL pixels to the ground truth positions. L is defined as the maximum between the height and the width of the subject's bounding box while β controls the thresholds for different precision requirements. We set it to 0.2 following the previous works [33, 20, 23, 49].

4.2. Implementation details

We first encode the dataset to videos with FFmpeg [9] and then employ the FFmpeg again to retrieve the encoded motion vectors and residual errors associated with each P-frame. Following [44], we use the MPEG-4 encoded videos, in which each GOP starts with an I-frame and then followed by 11 P-frames in general. For the last GOP within each video, we pad it to 12 frames with dummy frames.

Following [20, 49, 23], We crop the I-frame, P-frame, motion vectors and the residual frame using the provided bounding box for Penn Action. For Sub-JHMDB, we generate the bounding boxes from the puppet mask following [20]. Each GOP shares the unique bounding box, which is the mean bounding box between the I-frame of the current GOP and the next GOP. The cropped frames are resized to 256 by 256. Please refer to the supp. for more details.

4.3. Ablation Studies

4.3.1 Usage of compression representation

We first conduct ablation studies on the Penn Action dataset to analyze the efficacy of few important design choices, including the Motion Compensated ConvLSTM as well as the Residual Driven Gate. To verify the intuition that motion vector and motion compensated residuals provide free of

Table 1. Ablation study on the usage of the compressed information. Introducing the motion compensation mechanism into the ConvLSTM provides us with 2% PCK improvements using almost no extra computations compared to the baseline **a**. The Residual Driven Dynamic Gate based on residuals (baseline **e**) raises 14% fewer frames for the computationally heavy feature extractions, compared to baselines **d** using P-frames for the input, which greatly reduces the computation. Please refer to the section 4.3.1 for the detailed experiments setup.

ID	F_{P}	Warp	Gate	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean	Keyframe	GFLOPs
a	×	×	×	95.8	95.6	91.1	87.5	96.8	96.0	94.8	93.8	8.3%	1.52
b	×	✓	×	98.3	98.0	93.8	90.3	97.8	97.2	95.8	95.8	8.3%	1.54
c	✓	✓	×	99.2	98.8	97.5	97.0	98.6	98.1	97.7	98.1	100%	10.32
d	✓	✓	F_{P}	98.8	98.6	96.7	95.9	98.2	97.8	97.3	97.5	43.2%	5.36
e	✓	✓	Res.	98.6	98.4	96.9	95.7	98.5	98.1	97.7	97.7	29.0%	4.10

Table 2. Ablation study on design of the gate on Sub-JHMDB. At the similar accuracy, with temporal models and Positional Encoding design, our models could outperform the baseline with 10% less frames for feature extraction.

Temporal	P.E.	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean	Keyframe	GFLOPs
×	×	98.2	97.3	91.7	85.7	99.2	96.6	92.1	94.8	45.3%	3.14
✓	×	98.2	97.3	91.6	84.9	99.2	96.6	92.2	94.6	38.7%	2.84
✓	✓	98.2	97.4	91.7	85.2	99.2	96.7	92.2	94.7	35.2%	2.70

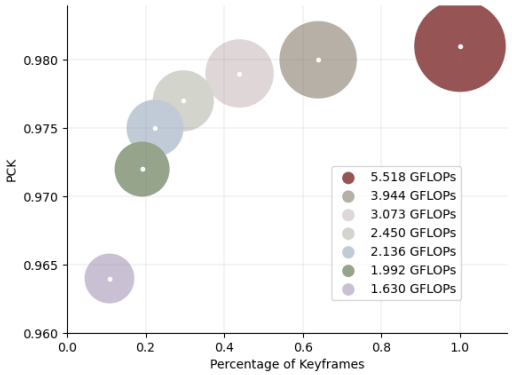


Figure 5. Relationships between PCK w.r.t. the percentage of frames selected as Keyframes for feature extraction on Penn Action. The radius corresponds to the computation complexities. Around 35% of frames could be dropped without affecting the PCK much. To maintain the SOTA 0.975 PCK, only 25% frames are necessary. This indicates the huge computation redundancies to perform pose estimation on each frame.

cost yet valuable representations for efficiently decoding the poses for each frame, we devise the following experiments as shown in Table 1 using the ResNet34 as the encoders:

a: In this experiment, only I-frame is employed to extract pose related features, which are then fed into the ConvLSTM to learn the temporal dynamics and decode poses for the following P-frames. Neither motion vectors nor features from P-frame were provided to the LSTM.

b: Motion information is infused into the temporal models through internal state warping. Features from the P-frames are not provided. Instead of extracting features from motion vectors, we warp the internal state within the ConvLSTM using the motion vectors directly, which incurs minimal amount of extra computation.

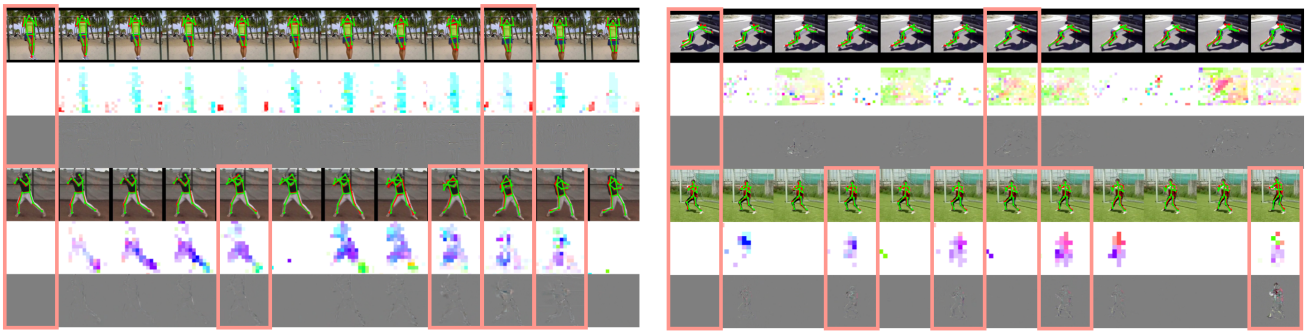
c: Motion vectors are explicitly used in the ConvLSTM to warp the ConvLSTM state as in **b**. Furthermore, features are extracted from each P-frame and always input to the ConvLSTM. The warped hidden and cell state assist the alignment between the hidden state and the input features.

d: Based on the models built in experiment **c**, we additionally introduce a dynamic gate to adaptively control if P-frame features are needed to fed into the LSTM. In this experiment, the gate takes the actual decoded P-frame as the input to derive skipping policies.

e: This is the proposed model, where we use the residual driven dynamic gate to determine whether feature extraction is necessary for each P-frame, based on the residual frames. Motion compensation is applied to the ConvLSTM to better align the features along the temporal dimension.

The efficacy of infusing motion to the ConvLSTM could be shown by comparing between experiment **a**, **b** and **c**. By simply warping the hidden state and cell state using the motion vectors, the mean PCK improved from 93.8 to 95.8. This performance gain verifies Motion Vector provides useful information when the features from P-frames are omitted. While for experiment **c**, the accuracy of the models could be further improved as accurate features from the corresponding P-frames are provided. The experiment **c** serves as the upper bound of the Motion Adaptive Pose Net.

By comparing experiment d, e, we investigate the effect of the Residual Driven Dynamic Gate. Compared to the gate that making decisions based on the fully decoded P-frames, our Residual Driven Dynamic Gate obtains 0.2% higher PCK with 14% more frames skipped for feature ex-



(a) Penn Action (b) Sub-JHMDB

Figure 6. Visualization of the decisions made by the Dynamic Gate and the estimated poses for (a) Penn Action and (b) Sub-JHMDB. Estimated poses are marked with green while red is for ground truth. Frames selected for feature extraction are marked with boxes in coral. Each example is organized as decoded frames, motion fields and residual frames from top to bottom. We plot sequences with slow motions and therefore less activations on top. While in bottom rows, our gate adapts to the more challenging motions and activates more often.

traction. The computation complexity difference will be magnified when more complex models are used. This is in line with our intuition that the motion compensation error (i.e. residual frame) provides a better measurement for the quality of the motion vectors and has higher information densities, allowing better decisions to be made by the gate.

4.3.2 Gate design

Our Residual Driven Dynamic Gate also involves a few important design choices including the temporal convolutions and positional encoding. We compare the performance of the gate with or without such designs in Table 2 on Sub-JHMDB. With about similar accuracy, introducing the temporal convolutions to the gate skips 7% more frames. Further adding the Positional Encoding could allow us to skip around 4% more frames. Note that without temporal convolution and position encoding, the gate essentially makes decision for each frame based on the residual error from this frame only. Compared to using all P-frames in a GOP, this option is appropriate for low-delay real-time applications with only a small increase in computational cost.

Introducing the temporal convolutions allows the gate to reason based on neighbouring motion compensated residuals. Intuitively, with limited chances to extract features from the actual frames, checking those frames with local maximum errors will bring max returns. Furthermore, adding the Positional Encoding allows the gate to have a sense on the distance to its neighbouring features and its absolute distance to the first I-frame. Higher probabilities therefore could be assigned to those frames that are relatively far away from the first I-frame.

4.3.3 Gate weight λ

By varying the weight λ, we could control the relative weight between the pose estimation accuracy and the number of activation in Eq. 15 during training. As a result, we could obtain models with diverse complexity profiles and performances. We plot the relationships between the computation complexities and the pose estimation accuracy in Fig. 5 for Penn Action dataset. As indicated in the plot, applying the uniform computation architectures on each individual frame would result in huge amount of wastes. Keeping only the selected 65% of frames leads to merely 0.001 drop for PCK. Around 77% of frames could be skipped if we want to maintain a SOTA PCK of 0.975. Also, as indicated in the plot, maintaining around 30% of frames serves like the sweet spot for balancing computation reduction and pose accuracy. The pose estimation accuracy drops more significantly when less than 30% of frames are kept.

4.4. Comparisons with state-of-the-arts

Finally, we compare both the accuracy and the efficiency of our models with the SOTA and report the results in Table 3 and Table 4. Our Motion Adaptive Pose Net achieves the highest PCK yet maintains the lowest computation profiles. Compared to the previous state-of-the-art, we obtain around 0.2% to 0.3% improvements in PCK with only 1/2 of the computation on both datasets using the ResNet18 as the backbones. We also include the result of our Motion Adaptive Pose Net using the same ResNet34 backbone as [49]. Compared to [49], we could further reduce the complexity, while improving the PCK by 0.3%. The savings in computation complexity is largely due to the explicit error map stored in the residuals, allowing us to use significantly lighter modules for our frame selection gate. While in KFP[49], ResNet based backbone has to be applied to each frame for feature extraction and followed by key frame proposal. Between our models using the ResNet34 vs. ResNet18 as the backbone, although the ResNet34 based model skipped more frames, the overall complexity is still 2x the Resnet18 model, while producing similar accuracy.

Table 3. Results on Penn Action dataset. Our Motion Adaptive Pose Net outperforms the SOTA models in both the efficiency and accuracy. Comparing to the KFP[49], we obtain 0.3% higher accuracy with 8% less keyframes. Exploiting the more efficient compressed signals, we outperform the SOTA models with around a half of computation.

Method	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean	Keyframe	GFLOPs
Nie et al.[46]	64.2	55.4	33.8	22.4	56.4	54.1	48.0	48.0	N/A	-
Iqbal et al.[13]	89.1	86.4	73.9	73.0	85.3	79.9	80.3	81.1	N/A	-
Gkioxari et al. [10]	95.6	93.8	90.4	90.7	91.8	90.8	91.5	91.9	N/A	-
Song et al. [33]	98.0	97.3	95.1	94.7	97.1	97.1	96.9	96.8	N/A	-
Luo et al. [20]	98.9	98.6	96.6	96.6	98.2	98.2	97.5	97.7	N/A	70.98
DKD(smallCPM) [23]	98.4	97.3	96.1	95.5	97.0	97.3	96.6	96.8	N/A	9.96
baseline [45]	98.1	98.2	96.3	96.4	98.4	97.5	97.1	97.4	N/A	11.96
DKD(ResNet50) [23]	98.8	98.7	96.8	97.0	98.2	98.1	97.2	97.8	N/A	8.65
KFP(ResNet34) [49]	98.2	98.2	96.0	93.6	98.7	98.6	98.4	97.4	38.0%	4.68
Ours(ResNet34)	98.6	98.4	96.9	95.7	98.5	98.1	97.7	97.7	29.0%	4.10
Ours(ResNet18)	98.9	98.7	96.9	96.3	98.4	98.0	97.4	97.7	29.7%	2.46

Table 4. Results on Sub-JHMDB dataset. The results are average from the 3 splits. Exploiting the more efficient compressed signals, we outperform the SOTA models with around half of computation. Furthermore, this experiment indicates that the Residual Driven Gate could develop an effective skipping policy with relatively small amount of data.

Method	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Mean	Keyframe	GFLOPs
Park et al.[24]	79.0	60.3	28.7	16.0	74.8	59.2	49.3	52.5	N/A	-
Nie et al. [46]	83.3	63.5	33.8	21.6	76.3	62.7	53.1	55.7	N/A	-
Iqbal et al. [13]	90.3	76.9	59.3	55.0	85.9	76.4	73.0	73.8	N/A	-
Song et al. [33]	97.1	95.7	87.5	81.6	98.0	92.7	89.8	92.1	N/A	-
Luo et al. [20]	98.2	96.5	89.6	86.0	98.7	95.6	90.0	93.6	N/A	70.98
DKD(ResNet50) [23]	98.3	96.6	90.4	87.1	99.1	96.0	92.9	94.0	N/A	8.65
baseline [45]	97.5	97.8	91.1	86.0	99.6	96.8	92.6	94.4	N/A	11.96
KFP(ResNet18) [49]	94.7	96.3	95.2	90.2	96.4	95.5	93.2	94.5	40.8%	4.68
Ours(ResNet18)	98.2	97.4	91.7	85.2	99.2	96.7	92.2	94.7	35.2%	2.70

Therefore the ResNet18 backbone is a better choice overall.

4.5. Visualization

We further visualize the decision made by the gate along with the predicted pose for selected GOPs in Fig. 6. Our models derive accurate poses from the frame sequences while the gate is only sparsely activated for frames with locally large residual errors, which coincides with our motivations to design the dynamic gate based on residual errors. As a result, only a fraction of frames are used to extract accurate features, while the remaining simply use the efficient motion compensated features, which leads to significant savings in computation. More results are in the supp.

5. Conclusion and future works

We develop the novel Motion Adaptive Pose Net to efficiently exploit the cost-free motion vectors and motion-compensation residuals from the compressed streams for pose estimation. A Motion Compensated ConvLSTM is proposed to spatially align the hidden and cell states over time and take dynamic inputs. Furthermore, an adaptive gate module is introduced to adaptively skip feature extractions for P-frames based on the residual information. Evaluating on the widely-used Penn Action and Sub-JHMDB datasets, the proposed Motion Adaptive Pose Net outperforms the SOTA models in PCK with significantly less computations. We hope this work could further inspire more studies on the usage of compressed signals for human pose estimation from videos.

In the future, we plan to explore more general frameworks for multi-person pose estimation from compressed videos. Theoretically, when treating each subject individually, our current framework could be applied to multi-person setups. However, this will incur different computation profiles for different subjects within the same frame instead of one global profile for the entire frame. We plan to develop an unified bottom up pose estimation model for multi-person scenarios as the future extension.

References

[1] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1014–1021. IEEE, 2009.

[2] Barak Battash, Haim Barad, Hanlin Tang, and Amit Bleiweiss. Mimic the raw domain: Accelerating action recognition in the compressed domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 684–685, 2020.

[3] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely-labeled videos. *arXiv preprint arXiv:1906.04016*, 2019.

[4] 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.

[5] 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.

[6] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Structured feature learning for pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4715–4723, 2016.

[7] Zhipeng Fan, Jun Liu, and Yao Wang. Adaptive computationally efficient network for monocular 3d hand pose estimation. In *European Conference on Computer Vision*, pages 127–144. Springer, 2020.

[8] Junyi Feng, Songyuan Li, Yifeng Chen, Fuxian Huang, Jiabao Cui, and Xi Li. How to train your dragon: Tamed warping network for semantic video segmentation. *arXiv preprint arXiv:2005.01344*, 2020.

[9] FFmpeg. Ffmpeg/ffmpeg.

[10] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *European Conference on Computer Vision*, pages 728–743. Springer, 2016.

[11] Hezhen Hu, Wengang Zhou, Xingze Li, Ning Yan, and Houqiang Li. Mv2flow: Learning motion representation for fast compressed video action recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(3s):1–19, 2020.

[12] Yuqi Huo, Xiaoli Xu, Yao Lu, Yulei Niu, Mingyu Ding, Zhiwu Lu, Tao Xiang, and Ji-rong Wen. Lightweight action recognition in compressed videos. In *European Conference on Computer Vision*, pages 337–352. Springer, 2020.

[13] 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.

[14] Samvit Jain and Joseph E Gonzalez. Fast semantic segmentation on video using block motion-based feature interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[15] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, Dec. 2013.

[16] Łukasz Kaiser and Samy Bengio. Discrete autoencoders for sequence models. *arXiv preprint arXiv:1801.09797*, 2018.

[17] Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In *International Conference on Machine Learning*, pages 2390–2399. PMLR, 2018.

[18] Lingchao Kong, Rui Dai, and Yuchi Zhang. A new quality model for object detection using compressed videos. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3797–3801. IEEE, 2016.

[19] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3d pose sequence machines. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 810–819, 2017.

[20] 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.

[21] Gedeon Muhawenayo and Georgia Gkioxari. Compressed object detection. *arXiv preprint arXiv:2102.02896*, 2021.

[22] 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.

[23] Xuecheng Nie, Yuncheng Li, Linjie Luo, Ning Zhang, and Jiashi Feng. Dynamic kernel distillation for efficient pose estimation in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6942–6950, 2019.

[24] Dennis Park and Deva Ramanan. N-best maximal decoders for part models. In *2011 International Conference on Computer Vision*, pages 2627–2634. IEEE, 2011.

[25] 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.

[26] 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.

[27] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2013.

[28] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Strong appearance and expressive spatial models for human pose estimation. In *Proceedings of the IEEE international conference on Computer Vision*, pages 3487–3494, 2013.

[29] Iain Richardson. White paper: an overview of h. 264 advanced video coding. *Vcodex/OneCodec*, 2011(7), 2007.

[30] Iain E Richardson. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[31] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*, 2015.

[32] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1268–1277, 2019.

[33] 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.

[34] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[35] Min Sun and Silvio Savarese. Articulated part-based model for joint object detection and pose estimation. In *2011 International Conference on Computer Vision*, pages 723–730. IEEE, 2011.

[36] Zhentao Tan, Bin Liu, Qi Chu, Hangshi Zhong, Yue Wu, Weihai Li, and Nenghai Yu. Real time video object segmentation in compressed domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(1):175–188, 2020.

[37] Yuandong Tian, C Lawrence Zitnick, and Srinivasa G Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *European Conference on Computer Vision*, pages 256–269. Springer, 2012.

[38] 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.

[39] 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.

[40] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[41] Keze Wang, Liang Lin, Chenhan Jiang, Chen Qian, and Pengxu Wei. 3d human pose machines with self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1069–1082, 2019.

[42] Shiyao Wang, Hongchao Lu, and Zhidong Deng. Fast object detection in compressed video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7104–7113, 2019.

[43] 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.

[44] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6026–6035, 2018.

[45] 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.

[46] Bruce Xiaohan Nie, Caiming Xiong, and Song-Chun Zhu. Joint action recognition and pose estimation from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1293–1301, 2015.

[47] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392. IEEE, 2011.

[48] 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.

[49] Yuexi Zhang, Yin Wang, Octavia Camps, and Mario Sznaier. Key frame proposal network for efficient pose estimation in videos. In *European Conference on Computer Vision*, pages 609–625. Springer, 2020.