Pose-Forecasting Aided Human Video Prediction With Graph Convolutional Networks

YANG ZHAO AND YONG DOU
National University of Defense Technology, Changsha 410073, China
Corresponding author: Yang Zhao (zhaoyang10@nudt.edu.cn)

ABSTRACT
Human video prediction is still a challenging problem due to the uncertainty of future actions and complexity of frame details. Recent methods tackle this problem in two steps: firstly to forecast future human poses from the initial ones, and then to generate realistic frames conditioned on predicted poses. Following this framework, we propose a novel Graph Convolutional Network (GCN) based pose predictor to comprehensively model human body joints and forecast their positions holistically, and also a stacked generative model with a temporal discriminator to iteratively refine the quality of the generated videos. The GCN based pose predictor fully considers the relationships among body joints and produces more plausible pose predictions. With the guidance of predicted poses, a temporal discriminator encodes temporal information into future frame generation to achieve high-quality results. Furthermore, stacked residual refinement generators make the results more realistic. Extensive experiments on benchmark datasets demonstrate that the proposed method produces better predictions than state-of-the-arts and achieves up to 15% improvement in PSNR.

INDEX TERMS
Human video prediction, GCN, GAN, temporal discriminator, stacked residual refinement.

I. INTRODUCTION
Video prediction is receiving increasing research attention, but it is still challenging due to complex video contexts consisting of both high-level semantics like walking and running and low-level pixel information [1]. For human action videos, human pose contains much semantic information, which intuitively can be leveraged to benefit accurate video prediction. Many methods use the already generated frames to make further predictions in a recursive paradigm [2]–[9], but the quality of generated videos would degrade over time due to accumulative pixel-level errors. Recent works [1], [10]–[13] adopt a hierarchical framework of disentangling the predictions of high-level semantics and low-level pixels, which avoids pixel-level error accumulation and effectively improves the quality of generated videos. However, most of them do not consider the structural relationships among semantic components in the videos. In addition, some other techniques are also used for improving the visual quality of generated results, like integrating temporal information to obtain higher-quality videos [14]–[18] and performing refinement for better results [19], [20]. Meanwhile, several works [21]–[24] aim to generate new realistic scenes, not to predict the real future.

In this work, we aim to forecast and generate high-quality future human videos from a single frame and a sequence of poses, which can be viewed as a combined problem of future pose forecasting and conditioned video generation. To solve this problem, we fully exploit the relations among human joints and leverage them for better human motion dynamics modeling and forecasting. We build a hierarchical framework containing two novel components for better pose prediction and video generation, as is shown in Figure 1.

The pose prediction module deploys a graph convolutional network (GCN) [25] which is more suitable and powerful for exploring relations among human body joints through intrinsic graph-based information propagation in representation learning. In particular, the graph takes body joints as nodes and models the relations of these nodes in a global view. For the video generation component, we introduce a stacked residual refinement module consisting of a sequence of generators that iteratively refine future frames over their residuals with the previous one to achieve better quality. The solution of the problem in residual domain focuses on...
FIGURE 1. Framework illustration of our proposed video prediction model. It consists of a pose prediction module and a stacked residual refinement module. The former deploys LSTM and GCN to predict future poses through exploring temporal context and body joints relations, and the output poses are fed into the latter for pose based video generation. The stacked residual module performs video refinement iteratively based on residual learning. Each stage generates a residual map to refine the final result conditioned on the predicted pose and video frames from the previous stage. The final stage integrates all the residual refinements and outputs the generated video.

motions and simplifies the difficulty. Moreover, we use a multi-stage refinement process which further improves the quality progressively. For each generator, we utilize Generative Adversarial Networks (GANs) and integrate a temporal discriminator which encodes temporal information. By distinguishing the generated video clip from the real one, the discriminator teaches the generator to produce more temporally fluent results.

Our contributions are summarized as follows:
1) We apply GCN to pose prediction. We find GCN achieves accurate prediction by effectively exploiting the relations of human body joints, in particular in presence of some distracting factors like self-occlusion.
2) We propose to encode the temporal information into the process of each single frame generation with a temporal discriminator to obtain more temporally fluent results.
3) We design a stacked residual refinement framework to iteratively improve the realism of the generated videos significantly.

II. RELATED WORK
Traditional video prediction approaches follow a recursive prediction process at pixel level [7], [9], and their performance often gets corrupted due to pixel-level error accumulation. Recent proposed hierarchical prediction methods can avoid pixel-level error accumulation by separating high-level structure (e.g. face landmarks, pose) prediction and structure-analogy video generation. Villegas et al. [1] first made high-level structure predictions, and then generated future videos by visual-structure analogy. Yang et al. [13] proposed a PSGAN for pose prediction and an SCGAN for video generation. Wang et al. [26] aimed at converting an input semantic video, such as videos of human poses or segmentation masks, to an output photorealistic video, and Wang et al. [24] extended it to synthesize videos of previously unseen subjects or scenes by leveraging a few example images of the target at test time. Existing hierarchical prediction works are rather limited and are still struggling for generating accurate pose predictions and realistic videos.

For pose prediction, recent methods [1], [13], [19] do not explore relations of joints w.r.t. the human pose. In Graph Convolutional Network (GCN) [25], relations among components of a structure are explored via the intrinsic graph-based information propagation, which achieves state-of-the-art performance in many domains such as text classification [27], [28], machine translation [29], [30] and image recognition [31]. Some works [32], [33] build visual classification models upon GCN to fuse semantic embeddings and categorical relationships. We apply GCN to encoding the internal association of the joints to make better predictions in videos.

Video generation is a challenging problem. GAN [34] based generative models are proposed benefiting from the development of computing power and achieve great success in image generation [35]–[41], but video generation is much harder for GANs due to more data contents and extra temporal information. Many works [14], [16], [18], [42] directly implement 3D convolutions to encode temporal information. Ji et al. [14] applied 3D convolutional networks to human action recognition. Vondrick et al. [18] used 3D convolutional networks for video generation. Besides 3D convolution, a coarse-to-fine strategy is also applied for more realistic results [19], [43]–[45]. Ma et al. [43] used L1 loss
to generate a coarse result, and then used GANs to refine it. Zhao et al. [19] utilized image GANs to generate a coarse result, followed by 3D convolutional GANs for refinement. However, 3D convolutions often lead to fixed-size outputs, which means low scalability, and their coarse-to-fine strategy is usually two-stage. We formulate the temporal information by temporal discriminator in a flexible way. Besides, a stacked residual refinement scheme is devised to recursively refine the target to gain more realistic results.

III. METHOD

A. PROBLEM DEFINITION

Let \( x_{1:m} = \{x_1, x_2, \ldots, x_m\} \) be a sequence from the first to the \( m \)-th frame in a human action video and \( p_{1:m} = \{p_1, p_2, \ldots, p_n\} \) be the sequence of the corresponding poses for these frames. Here each pose \( p_i \in \mathbb{R}^{2 \times L} \) is represented by the 2D coordinates of \( L \) human body joints. We aim to solve the problem of predicting \( n \) future video frames \( x_{m+1}, \ldots, x_{m+n} \) based on \( p_{1:m} \) and a random frame \( x_s \) from \( x_{1:m} \).

We solve this problem through two phases: we first predict future poses \( p_{m+1} \) to \( p_{m+n} \) based on \( p_{1:m} \) in x-y coordinates and then generate the video frames \( x_{m+1}, \ldots, x_{m+n} \) based on the forecasted poses which are converted into heatmaps. Through such a two-step solution strategy, we alleviate the difficulties by disentangling the pose and frame appearance modeling and leveraging the pose information as effective guidance for frame generation. Figure 1 provides an architecture overview for our proposed model. The model consists of a pose prediction module and a stacked residual refinement module. The pose prediction module deploys GCN to make better prediction, since GCN defines different relations between joints in a graph. For example, hands and shoulders contain more useful information regarding elbows than knees, which can guide more efficient information communication with GCN. The temporal discriminator takes a sequence of generated frames as an entirety to increase the temporal fluency. Stacked refinements further optimize the results in a residual domain by an iterative way.

B. GRAPH CONVOLUTIONAL NETWORK BASED POSE PREDICTION

Pose prediction is a sequence-to-sequence learning problem, which requires to predict the future poses \( p_{m+1:m+n} \) from the observed past ones \( p_{1:m} \). One natural choice of the model for solving this problem is the LSTM based RNN, which has proved to be very effective in other similar problems [1]. However, we observe LSTM-RNN performs not so well in presence of common distracting factors for pose prediction, e.g., self-occlusion and unusual viewpoint. This would introduce errors and degrade the quality of relatively far future poses when the errors accumulate. Considering human poses are well structured, we therefore deploy graph convolutional networks (GCNs) to fully exploit relations among body joints and address such issues.

The pose prediction module in our proposed model consists of encoder, LSTM and GCN. A linear encoder is applied to convert the the pose motion dynamics into hidden states. LSTM encodes the hidden states and predicts the future poses; GCN explores the relations among body joints to produce a better pose prediction.

The encoder and LSTM predictor encode \( p_{1:m} \) in a chronological order, which encodes \( p_t \) at time \( t, t \in [1, m] \), into hidden states:

\[
e_t = \sigma(W_e p_t),
\]

\[
[h^0_t, c_t] = \text{LSTM} \left( e_t, h^0_{t-1}, c_{t-1} \right)
\]

where \( \sigma(\cdot) \) denotes an activation function and ReLU(\()\) is adopted here. \( e_t \in \mathbb{R}^e \) represents the hidden state of \( p_t, h^0_t \in \mathbb{R}^{d_0} \) represents the observed dynamics up to time \( t, c_t \) is the memory cell that retains information from the history of pose inputs. The encoder encodes the poses \( p_t \) into hidden states \( e_t \). The LSTM receives the past \( m \) poses \( e_{1:m} \) to recurrently predict the future \( n \) hidden states \( h^0_{m+1}, \ldots, h^0_{m+n} \).

Based on the hidden states \( h^0_t, t \in [m+1, m+n] \), a GCN layer is applied to predict future poses \( p_{m+1:m+n} \). More specifically, after obtaining \( h^0_t \) from LSTM, a fully connected layer is at first used to convert the hidden state \( h^0_t \) containing holistic features into a vector \( h^1_t \in \mathbb{R}^{d_1} \). And a nonparametric reshape operation converts it into a matrix \( h^2_t \in \mathbb{R}^{L \times d_2} \), satisfying \( d_1 = L \times d_2 \), with \( h^2_t \) representing the \( d_2 \)-dimensional features for each of the \( L \) body joints. Then, a GCN layer is integrated as follows:

\[
h^3_t = \sigma \left( \bar{D}^{-1/2} \bar{A} \bar{D}^{-1/2} h^2_t W_2 \right),
\]

where \( \bar{A} = A + I_N \in \mathbb{R}^{L \times L} \) is the augmented adjacency matrix of the undirected graph \( G \) over the \( L \) body joints, with added self-connections. Here \( I_N \) is the identity matrix; \( D_{ii} = \sum_j \bar{A}_{ij} \), and \( W_s \) are trainable weight matrices.

We build the graph \( G \) for capturing body joints relations and the human body structure as follows. For each element \( a_{ij} \in A \), we set it as 1 if the corresponding joints \( i \) and \( j \) are adjacent or symmetrical to each other based on human kinematic priors [46], and 0 otherwise.

Finally, another GCN layer converts \( h^3_t \) to predicted poses \( \tilde{p}_t, t \in [m+1, m+n] \), where we apply tanh to the activation function, and other parameters are similar to Eq.2. The predictor is trained with the standard mean squared error (MSE) loss of the predicted \( n \) poses:

\[
L_p = \frac{1}{n} \sum_{i=1}^{n} \|p_{m+i} - \tilde{p}_{m+i}\|^2.
\]

C. POSE AIDED VIDEO GENERATION

With the above predicted poses, the video generation module predicts the video frames with realistic contents compatible with the predicted poses. This is achieved by a stacked refinement architecture that generates videos through multi-stage
residual learning. Within each generation stage, the generation component predicts a plausible video frame in a residual form.

Concretely, the generation module has an encoder-decoder architecture. In the first generation stage, a past frame with its pose and a future pose are taken as input, and the output is the corresponding future frame. To match each keypoint to a position on human body, we encode the poses as heatmaps following [43], each heatmap filled with 1 in a radius of 2 pixels around the corresponding keypoint and 0 elsewhere.

The generator learns the residual map to reduce learning redundancy:

$$\Delta (\tilde{x}_t, x_s) = \tilde{x}_t - x_s = G (x_s, g(p_s), g(p_t)),$$  

(4)

where $x_s$, $p_s$ are the past frame and its pose. $\tilde{x}_t$, $p_t$ are the future frame we generate and its pose. $G(\cdot)$ is the function that converts x-y coordinates to heatmaps. $G(\cdot)$ represents the generator we need to learn. It is a difficult task to synthesize the whole frame, and modeling the residual map can simplify this problem and enable the generator to focus on motions. Particularly, U-Net [47] is employed as the backbone of the generator, where skip-connections are conducted over features at both semantic and texture levels as is shown in Figure 1. With such a backbone, our model concatenates the frame and heatmaps as input to the same encoder to make feature fusion at the two levels. Semantic features give instructions to generate frames with rough poses, and texture features help improve the details of the generated frames.

To improve the quality and realism of generated video frames, we propose a temporal discriminator to supervise and train the generation model, which effectively encodes temporal information (e.g., temporal consistency and smoothness) into the video generation to make generated frames more realistic.

We develop and adopt the following four losses to stabilize the training process and gain realistic results. We use the sparse reconstruction loss to compare the generation $\tilde{x}_t$ with the target frame $x_s$, for which $\ell_1$ distance in the image space is adopted:

$$\mathcal{L}_{rec} (\tilde{x}_t, x_s) = \|x_s - \tilde{x}_t\|_1.$$  

(5)

Reconstruction loss guides the generation to make a rough prediction that reflects most details of the target and stabilizes the training process.

We find using only $\ell_1$ loss leads to reasonable but blurry results, which is also observed in other works [35]. We thus add a perception loss in the feature space to exploit the feature differences and obtain more reasonable results, where VGG19 network is chosen for feature extraction [48], [49]. The perception loss is defined as

$$\mathcal{L}_{per} = \| VGG(x_s) - VGG(\tilde{x}_t) \|_1.$$  

(6)

where VGG(\cdot) outputs the pretrained VGGNet [50] features at dent layers of the network.

Following common practice in image and video generation [35], [43], [49], we also adopt the following adversarial spatial loss of Conditional Image Discriminator. A conditional image discriminator $D_I$ ensures the output frame to resemble the target one w.r.t. the given pose, and $G$ is our generator introduced in Equation (4). The objective of a conditional adversarial loss [51] is expressed as

$$\min_{D_I} V(D_I) = \frac{1}{2} \mathbb{E} \left[ (D_I(x_s, g(p_s), g(p_t)) - b)^2 \right] + \frac{1}{2} \mathbb{E} \left[ (D_I(\tilde{x}_t, x_s, g(p_s), g(p_t)) - a)^2 \right],$$

$$\min_{G} V(G) = \frac{1}{2} \mathbb{E} \left[ (D_I(\tilde{x}_t, x_s, g(p_s), g(p_t)) - c)^2 \right].$$  

(7)

where we assign $b = 1, c = 1$ to denote real samples, $a = 0$ to denote fake samples. $x_s$ and $x_t$ denote the source frame and the ground truth target frame respectively. $p_s$ and $p_t$ are the corresponding poses. $\tilde{x}_t$ represents the generated target frame. Specifically, the loss for generator of $D_I$ is

$$\mathcal{L}_{D_I} = \frac{1}{2} \mathbb{E} \left[ (D_I(\tilde{x}_t, x_s, g(p_s), g(p_t)) - c)^2 \right].$$  

(8)

A real video not only looks real in every frame, but is continuous and fluent in contents. We argue that temporal information can make a video look fluent. It is not enough to only generate realistic images. We utilize a temporal discriminator to encode temporal information into frame generation, aiming to get temporally fluent videos. The temporal GAN is trained by

$$\min_{D_T} V(D_T) = \frac{1}{2} \mathbb{E} \left[ (D_T(x_t | x_{t-d+1}) - b)^2 \right] + \frac{1}{2} \mathbb{E} \left[ (D_T(\tilde{x}_t | x_{t-d+1}) - a)^2 \right],$$

$$\min_{G} V(G) = \frac{1}{2} \mathbb{E} \left[ (D_T(\tilde{x}_t | x_{t-d+1}) - c)^2 \right].$$  

(9)

with the same notations as Equation (7). The loss for generator of $D_T$ is

$$\mathcal{L}_{D_T} = \frac{1}{2} \mathbb{E} \left[ (D_T(\tilde{x}_t | x_{t-d+1}) - c)^2 \right].$$  

(10)

The purpose of this discriminator is to make new generated frames look real. By combining the previous frames and the present frame, the generator targets at getting a fluent result in contexts.

To sum up, our final loss is composing the aforementioned losses with linear combination:

$$\mathcal{L} = \lambda_{rec} \mathcal{L}_{rec} + \lambda_{per} \mathcal{L}_{per} + \lambda_{D_I} \mathcal{L}_{D_I} + \lambda_{D_T} \mathcal{L}_{D_T}.$$  

(11)

D. STACKED RESIDUAL REFINEMENT

The quality of frames generated with one stage as introduced above usually suffers slight artifacts and blurriness. We here introduce a stacked residual refinement module. Generators are integrated in a stacked paradigm to refine the target recursively.

For a $k$-stage network, there are $k$ repeated generators. The first generator is the same as what we described in Section III-C, which is denoted as $G_1$ and produces a residual
map $\Delta_1$. At the end of the first stage we get $\tilde{x}_i^{(1)} = x_i + \Delta_1$. The generator for stage 2, denoted as $G_2$, has the same structure as $G_1$, but with a different input. It takes not only all the inputs of $G_1$, including $x_i$, $p_i$, $p_j$, but also $\Delta_1$, which is the result of $G_1$. The result of $G_2$, denoted as $\Delta_2$, is a refinement of $\Delta_1$, so the results of both stages are summed to form a new residual map $\Delta_2 = \Delta_1 + \Delta_2$. The result of the second stage is $x_i^{(2)} = x_i + \Delta_2$. Similarly, for stage $i$, the inputs are $x_i$, $p_i$, and $\Delta_{i-1}$, where $\Delta_{i-1} = \sum_{j=1}^{i-1} \Delta_j$. The residual map calculated from this stage is $\Delta_i$ and $\Delta_i = \Delta_{i-1} + \Delta_i$. The corresponding result is $x_i^{(i)} = x_i + \Delta_i$.

**Algorithm 1** Stacked Residual Refinement Module

---

1: procedure Refine($x_i$, $p_i$, $p_j$, $k$) $\triangleright$ K-stage refinement  
2: Train $G_1$ $\triangleright$ Stage 1 is special  
3: $\Delta_{[1]} \leftarrow \Delta_1 \leftarrow G_1(x_i, g(p_i), g(p_j))$  
4: $\tilde{x}_i^{(1)} \leftarrow x_i + \Delta_1$  
5: $i \leftarrow 2$  
6: while $i \leq k$ do $\triangleright$ Every loop defines a refinement stage  
7: Train $G_i$  
8: $\Delta_i \leftarrow G_i(x_i, g(p_i), g(p_j), \tilde{\Delta}_{i-1})$  
9: $\tilde{x}_i^{(i)} \leftarrow \tilde{x}_i + \Delta_i$  
10: $i \leftarrow i + 1$  
11: $\tilde{x}_i \leftarrow \tilde{x}_i^{(k)}$  
12: return $\tilde{x}_i$  
---

**IV. EXPERIMENTS**

We evaluate the proposed algorithm by first comparing it with baselines [1], [3], [18], [19], and then performing component studies w.r.t. adversarial loss from temporal discriminator, stacked residual refinement, and GCN. We adopt Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity index (SSIM) and Percentage of Correct Keypoints (PCK) as quantitative metrics.

**A. DATASETS**

Penn Action [52] contains 2, 326 video sequences of 15 different actions. Humans in its video sequences are annotated by 13 joints. We follow the official train split and action selection in [1], [19], using actions of baseball pitch, baseball swing, clean and jerk, golf swing, jumping jacks, jump rope, tennis, tennis, and tennis serve.

Human3.6M [53] contains 3.6 million 3D human poses and corresponding images with 11 professional actors in 17 scenarios. Humans are annotated by 32 joints in both 2D and 3D coordinates. Since several joints are close, we select 17 joints as pose information, including pelvis, right hip, right knee, right ankle, left hip, left knee, left ankle, spine, neck, head, head top, left shoulder, left elbow, left wrist, right shoulder, right elbow and right wrist.

UCF101 [54] contains 101 action classes, over 13k clips and 27 hours of video data. In our experiments, we choose the data from two action classes, pull-ups and Taichi. We run Openpose [55] on the frames and take the results as pseudo-labels.

**B. IMPLEMENTATION**

Following problem setting in [19], the predictor sees 10 past frames and predicts 32 frames. For generators, the traditional U-Net backbone [47] consists of 4 downsampling layers and corresponding upsampling layers. Firstly, the input is encoded into a feature with the channel number of 128 by a convolutional layer. Then, the downsampling layers are implemented as convolutional layers, where the kernel size is set to 3 and the stride is set to 2. Comparing the upsampling layers are implemented as transposed convolutional layers, where the kernel size is set to 3 and the stride is set to 2. BatchNorm and ReLU are used after every convolutional and transposed convolutional layer. Adam optimizer [56] is chosen for back propagation. Hyperparameters are set as $\lambda_{rec} = 100$, $\lambda_{per} = 10$, $\lambda_{D_i} = 1$ and $\lambda_{D_P} = 1$ in all the experiments without further search. In temporal discriminator, $d$ is set to 1 to look back for one frame. For equality, we choose our stacked residual refinement number $k = 2$, since most algorithms define a coarse network and a refinement network. In training, the model is trained stage by stage, where parameters are fixed after the corresponding training stage. The number of training epoch is set to 10 in all experiments.

**C. COMPARISON WITH BASELINES**

The comparisons with state-of-the-arts over prediction of eight actions from Penn Action in terms of PSNR are shown in Figure 2. The legends in the first subfigure (Baseball Pitch) show the compared methods denoted in different colors, and in the bracket the models used to generate prior poses are given. As we can see from the curves, our method in ground truth pose setting has a higher PSNR than other methods for stable generation. In predicted pose setting, our method also achieves better performance than others. These results clearly validate the superiority of our method over the state-of-the-art baselines.

Table 1 shows the comparisons between the proposed method and Villegas et al. [1] using PSNR and SSIM metrics. We calculate the average results of all 32 predicted frames. In both ground truth setting and predicted pose setting, our method always achieves better quantitative results. Especially in the ground truth setting, our algorithm outperforms the baseline by 15% in PSNR.

**TABLE 1. Comparisons between our proposed model and state-of-the-arts for video generation in terms of PSNR and SSIM on Human3.6M.**

|                | PSNR  | SSIM  |
|----------------|-------|-------|
| Villegas et al. (LSTM) | 19.23 | 0.74  |
| Villegas et al. (GT)   | 21.82 | 0.79  |
| Ours (GCN)            | 21.22 | 0.83  |
| Ours (GT)             | 25.10 | 0.88  |
Some examples of generated frames are shown in Figure 3. With available ground truth poses, the baseline successfully reserves the red shirt in each frame, but the actor’s head in the fourth frame is posed inconsistently with adjacent frames. Comparatively, our algorithm gives a more temporally fluent result. In the setting of predicted poses, in the frame the actor’s hand and elbow look weird, while the arm is normal in our method. The artifact is possibly caused by the conflict of predicted elbow and wrist while GCN in our method can avoid it.

Experiments in pull-ups and Taichi from UCF101 are also implemented. Table 2 shows the results comparing vid2vid [26]. Since vid2vid does not include the prediction part, we just implement the ground truth pose setting for it. The quantitative results show that our method outperforms the baseline in both pose settings. Figure 4 shows the qualitative result of pull-ups. We can see the frames generated by vid2vid
have unstable scenes and the first frame collapses. Since our method learns the results in residual domain, it produces much more stable scenes in both pose settings.

## D. COMPONENT STUDY

### 1) POSE PREDICTION

The pose prediction results of LSTM and our LSTM-GCN in terms of average PCK@0.2 on both datasets are given in Table 3. It can be seen that GCN is helpful to encode the relations between joints. Better pose prediction will contribute to better video prediction.

|                  | LSTM      | LSTM-GCN (ours) |
|------------------|-----------|-----------------|
| Penn Action      | 0.6227    | 0.6331          |
| Human3.6M        | 0.7794    | 0.7868          |

### 2) VIDEO GENERATION

We conduct the following experiments to test the effects of the temporal discriminator and refinement components in our method: (1) Ours w/o Temp. Temporal discriminator is removed in this setting to verify the contribution of temporal information. (2) Ours w/o Refine. In this setting, results from only stage 1 are calculated to analyze the influence of stacked refinement.

|                  | Stage #1 | Stage #2 | Stage #3 |
|------------------|----------|----------|----------|
| Ours w/o Temp.   | 24.10    | 24.35    | 24.55    |
| Ours w/o Refine. | 25.10    | 25.12    |          |
| Ours             | 25.12    | 25.12    |          |

### 3) STACKED RESIDUAL REFINEMENT

Table 5 shows stacked residual refinement improves the final results in different settings. Higher video quality is obtained as the number of stages increases.

## V. CONCLUSION

This paper presented a novel method for human video prediction, which was separated into pose prediction and pose based video generation. GCN explored the relations among joints to make better pose predictions. For video generation, GANs paradigm was followed and temporal discriminator encoded temporal information to produce more realistic results. Meanwhile, stacked residual refinement of generators further improved the final performance in a recursive way. Our method achieved higher performance than state-of-the-arts on Human3.6M, Penn Action and UCF101 datasets. In the future, we aim to explore more relationships between the poses and generated frames to further promote the accuracy.

## REFERENCES

[1] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, “Learning to generate long-term future via hierarchical prediction,” in Proc. 34th Int. Conf. Mach. Learn., vol. 70, 2017, pp. 3560–3569.

[2] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, “Action-conditional video prediction using deep networks in Atari games,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 2863–2871.

[3] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” 2015, arXiv:1511.05440. [Online]. Available: http://arxiv.org/abs/1511.05440

[4] R. Goroshin, M. F. Mathieu, and Y. LeCun, “Learning to linearize under uncertainty,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 1234–1242.

[5] N. Srivastava, E. Mansimov, and R. Salakhutdinov, “Unsupervised learning of video representations using LSTMs,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 843–852.

[6] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, “Video (language) modeling: A baseline for generative models of natural videos,” 2014, arXiv:1412.6604. [Online]. Available: http://arxiv.org/abs/1412.6604

[7] C. Finn, I. Goodfellow, and S. Levine, “Unsupervised learning for physical interaction through video prediction,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 64–72.
[8] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, “Decomposition motion and content for natural video sequence prediction,” in Proc. ICLR, 2017, pp. 1–22.

[9] W. Lotter, G. Kreiman, and D. Cox, “Deep predictive coding networks for video prediction and unsupervised learning,” in Proc. ICLR, 2017, pp. 18.

[10] J. Walker, A. Gupta, and M. Hebert, “Patch to the future: Unsupervised visual prediction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 3302–3309.

[11] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng, “Forecasting human dynamics from static images,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3643–3651.

[12] J. Yuan and A. Torralba, “A data-driven approach for event prediction,” in Proc. 11th Eur. Conf. Comput. Vis. (ECCV), Berlin, Germany: Springer-Verlag, 2010, pp. 707–720. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888028.1888082

[13] C. Yang, Z. Wang, X. Zhu, C. Huang, J. Shi, and D. Lin, “Pose guided human video generation,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 204–219.

[14] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 221–231, Jan. 2013.

[15] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Proc. Neural Inf. Process. Sys., 2014, pp. 568–576.

[16] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (ICCV), Dec. 2015, pp. 4489–4497.

[17] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, “MoCoGAN: Decomposing motion and content for video generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1526–1535.

[18] M. Vondríček, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” in Proc. Neural Inf. Process. Syst., 2016, pp. 613–621.

[19] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. Metaxas, “Learning to forecast and refine residual motion for image-to-video generation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 387–403.

[20] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4724–4732.

[21] D. Wei, X. Xu, H. Shen, and K. Huang, “GAC-GAN: A general method for appearance-controllable human video motion transfer,” 2019, arXiv:1911.10672. [Online]. Available: http://arxiv.org/abs/1911.10672

[22] Y. Balaji, M. R. Min, B. Bai, R. Chellappa, and H. P. Graf, “Conditional GAN with discriminative filter generation for text-to-video synthesis,” in Proc. 28th Int. Joint Conf. Artif. Intell., Aug. 2019, pp. 1995–2001.

[23] W. Liu, Z. Piao, J. Min, W. Luo, L. Ma, and S. Gao, “Liquid warping-GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 5904–5913.

[24] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, “Few-shot video-to-video synthesis,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 5014–5025.

[25] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. Int. Conf. Learn. Represent., 2017, pp. 1–14.

[26] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, “Video-to-video synthesis,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 1144–1156.

[27] J. Atwood and D. F. Towsley, “Diffusion-convolutional neural networks,” in Proc. Neural Inf. Process. Syst., 2016, pp. 1993–2001.

[28] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in Proc. Neural Inf. Process. Syst., 2016, pp. 3844–3852.

[29] J. Bastsings, I. Titov, W. Aaziz, D. Marcheggiani, and K. Simaan, “Graph convolutional encoders for syntax-aware neural machine translation,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2017, pp. 1957–1967.

[30] D. Marcheggiani, J. Bastsings, and I. Titov, “Exploiting semantics in neural machine translation with graph convolutional networks,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., 2018, pp. 486–492.

[31] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, “Multi-label image recognition with graph convolutional networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 5177–5186.
[54] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” 2012, arXiv:1212.0402. [Online]. Available: http://arxiv.org/abs/1212.0402

[55] Z. Cao, G. H. Martinez, T. Simon, S.-E. Wei, and Y. A. Sheikh, “OpenPose: Realtime multi-person 2D pose estimation using part affinity fields,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Jul. 17, 2019, doi: 10.1109/TPAMI.2019.2929257.

[56] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent., 2015, pp. 1–15.

**YANG ZHAO** was born in Shandong, China, in 1992. He received the B.S. degree and the M.S. degree in computer science and technology from the National University of Defense Technology, Hunan, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree. His research interests include machine learning and computer vision.

**YONG DOU** was born in Jilin, China, in 1966. He is currently the Director of the Science and Technology on Parallel and Distributed Processing Laboratory. He is also the Project Leader building large-scale parallel computers for deep learning applications in the National University of Defense Technology.