Unsupervised Keypoint Learning for Guiding Class-Conditional Video Prediction

Author: Yunji Kim , Seonghyeon Nam , In Cho , and Seon Joo Kim

Published Venue: NeurIPS 2019

Presenter: Xianhang Li

Faculty: Dr. Yogesh S Rawat



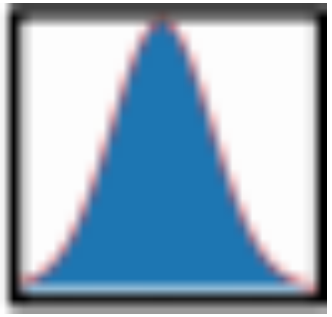
Contents

- Problem
- Motivation
- Overview
- Proposed Methods
- Dataset
- Results
- Conclusion



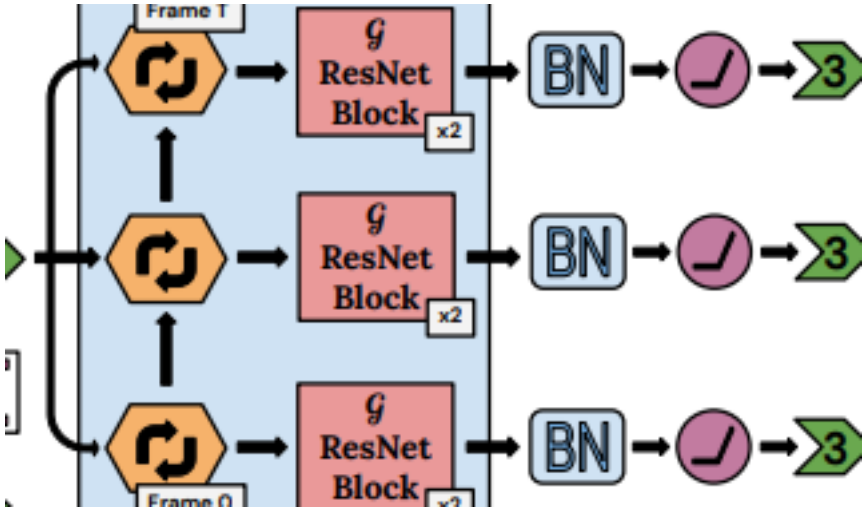
Problem

Video Generation



Standard Gaussian Noise

Unconditional generation



Generative model

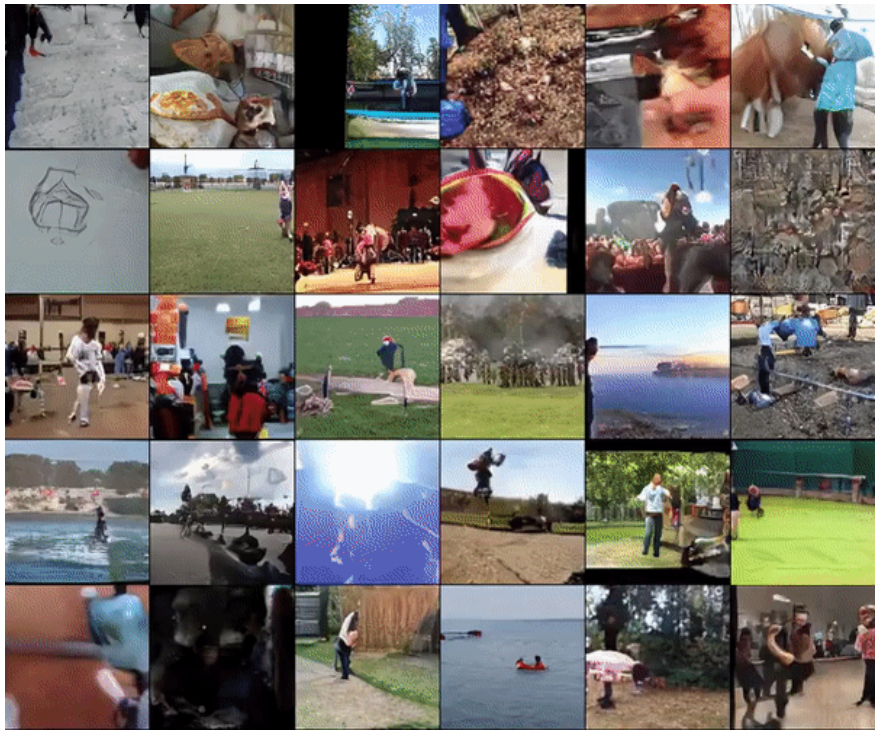
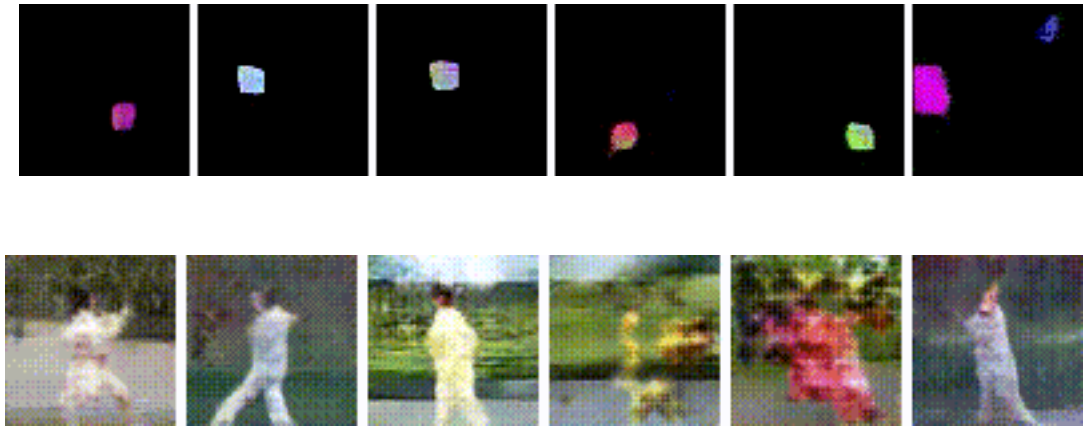


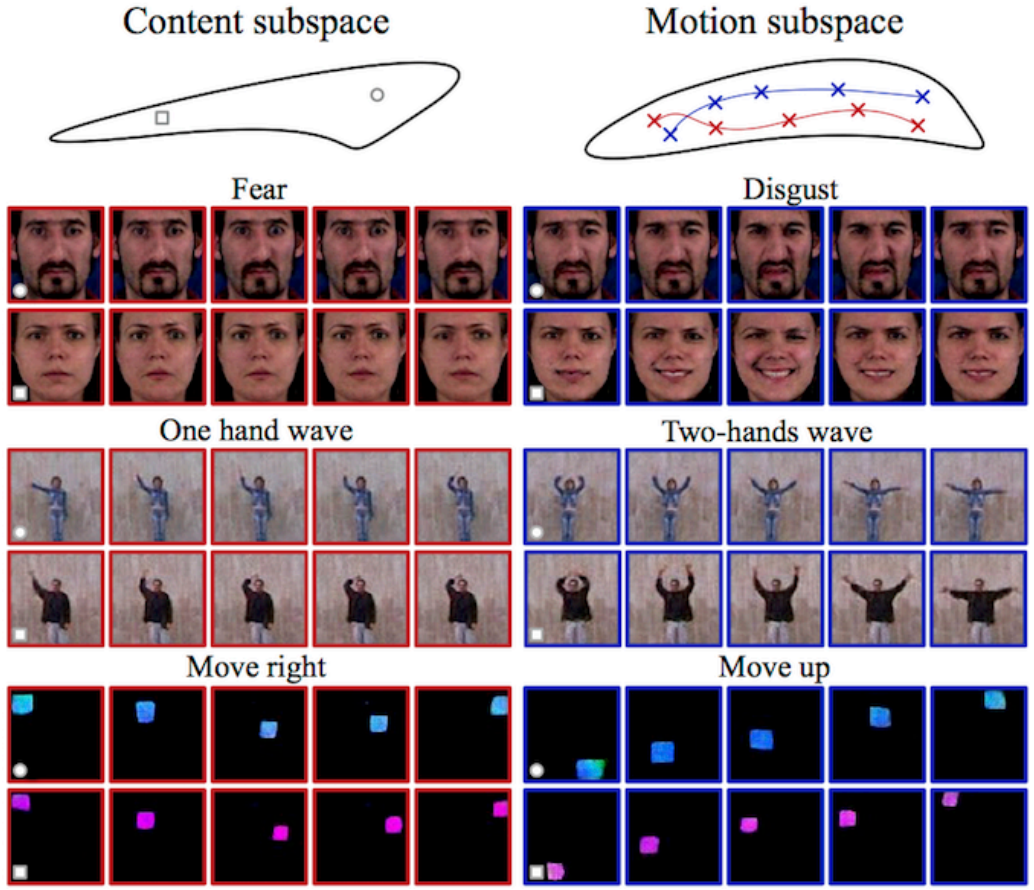
Image Source: https://medium.com/syncedreview/deepmind-dvd-gan-impressive-step-toward-realistic-video-synthesis-12027d942e53

Problem

Video Generation

Unconditional generation





Tulyakov, M. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in CVPR, 2018.

Problem

Video Generation Conditional generation

Video-to-Video Synthesis

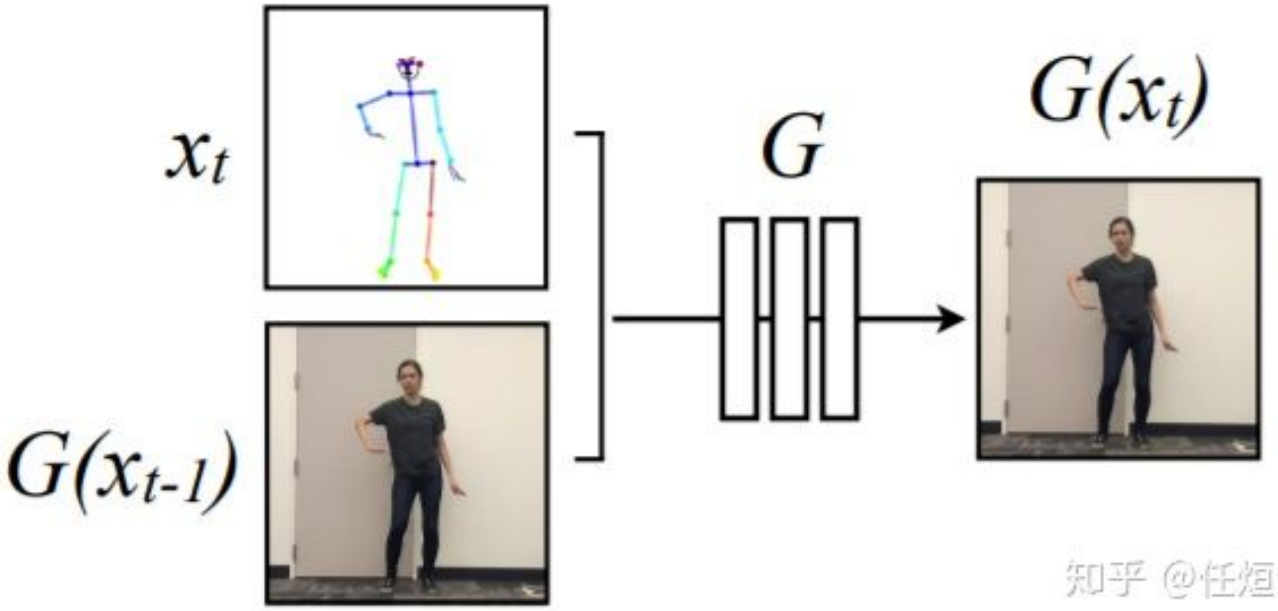


Image Source： https://tcwang0509.github.io/vid2vid/

Problem

Video Generation

Conditional generation



Frame and pose

Generative model

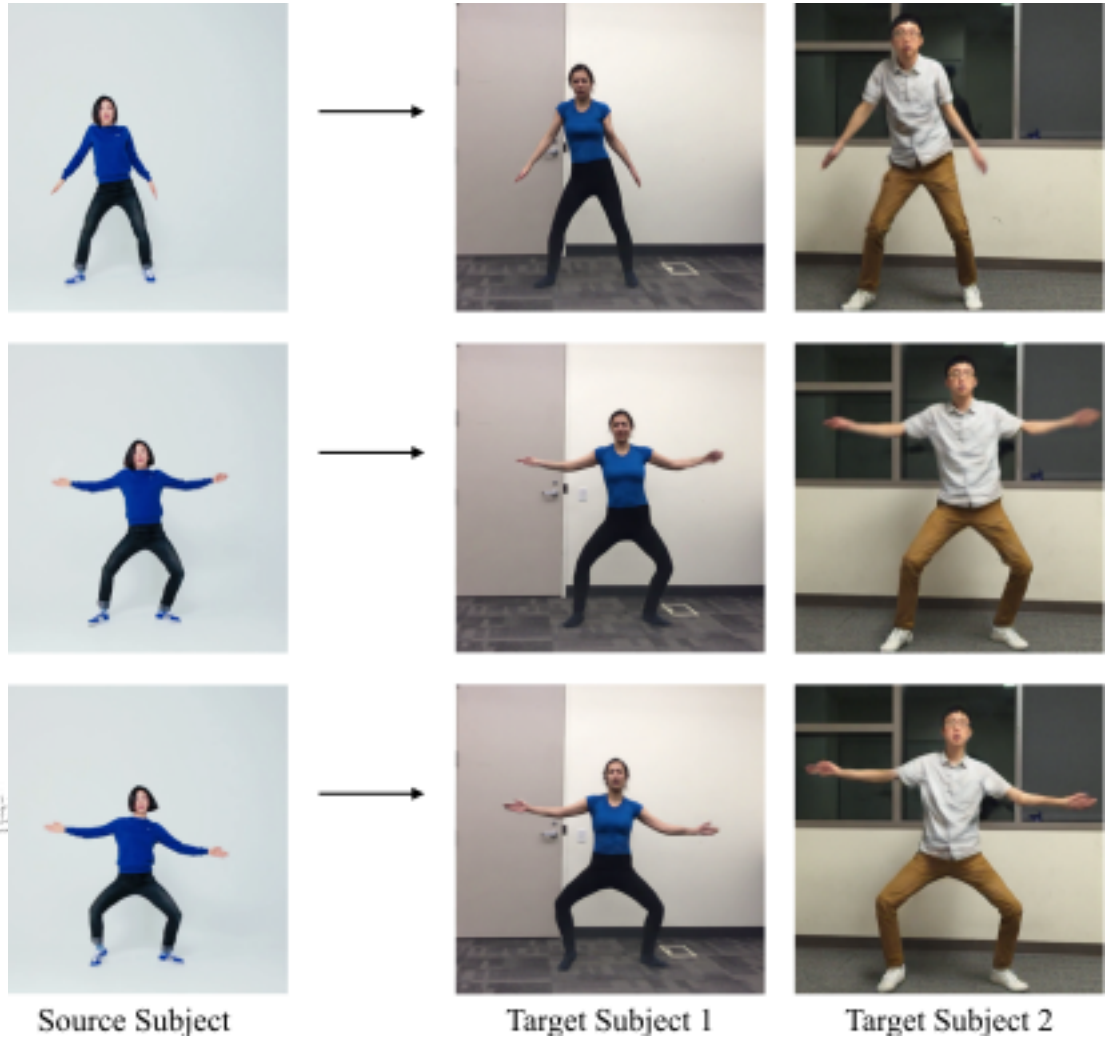
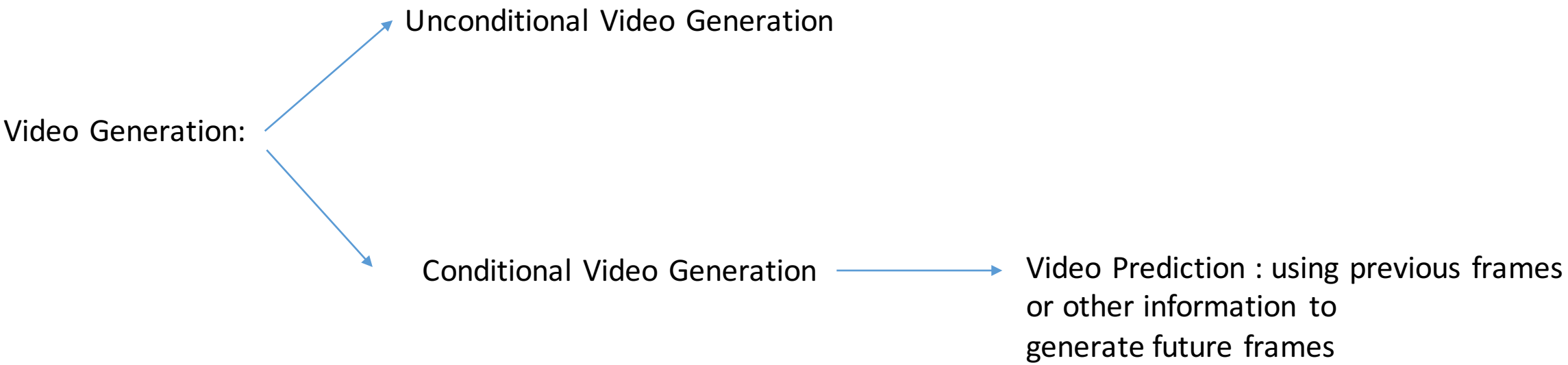


Image Source： https://zhuanlan.zhihu.com/p/67942949

Problem


Video Generation:
Unconditional Video Generation
Conditional Video Generation
Video Prediction : using previous frames or other information to generate future frames

Problem

Main Challenge **Temporal Coherence**

- Continuous frames should be associated with each other

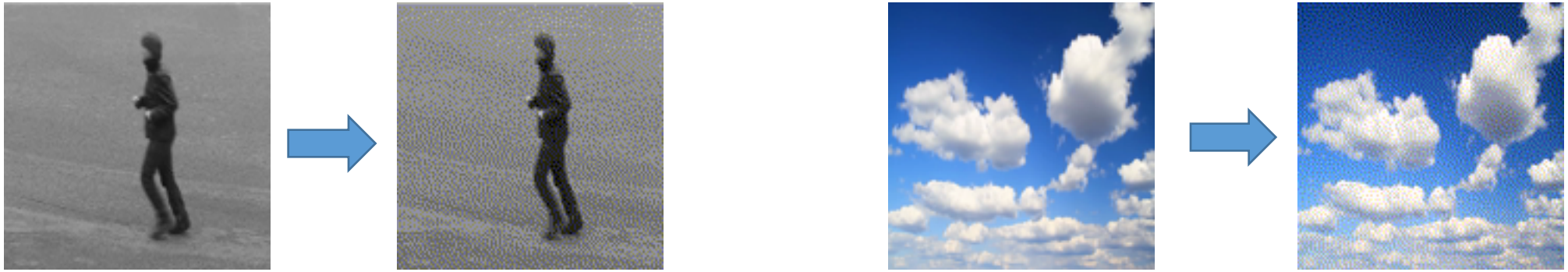
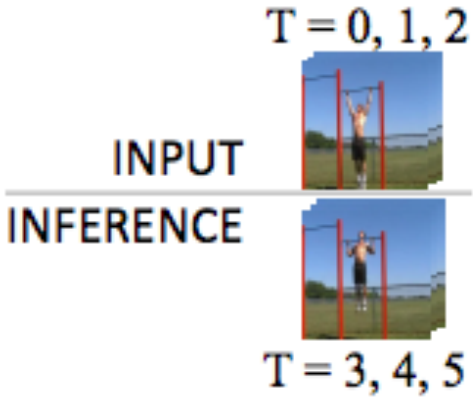


Image Source： https://github.com/Yijunmaverick/FlowGrounded-VideoPrediction

Motivation : Previous Solutions

Condition1 : Previous frames



- **predict Transformation instead of pixel**

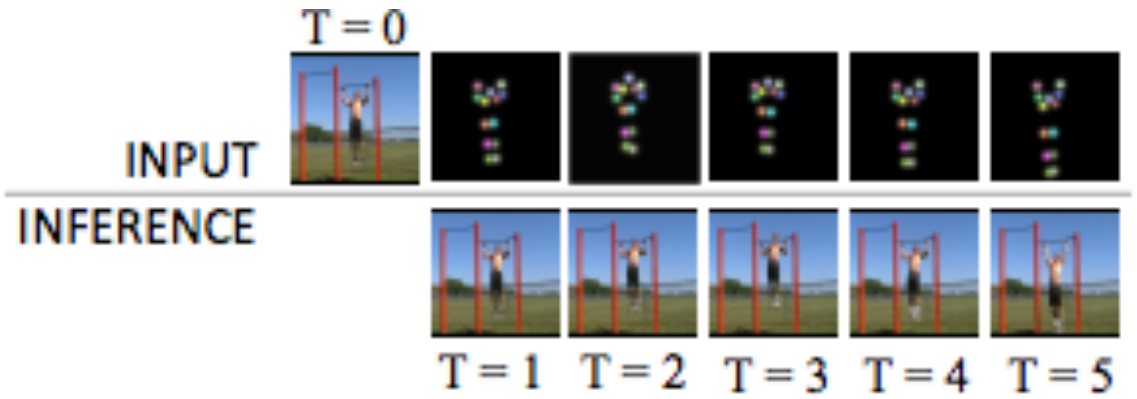
Eg. affine transformation, Convolution

- **Decompose the parts that are constant and change**

Eg. Background and foreground, content and motion

Motivation : Previous Solutions

Condition2: Pose information + previous frame



- **utilize labels of human joint positions**
- **use a key-points sequence**

Motivation : Previous Solutions

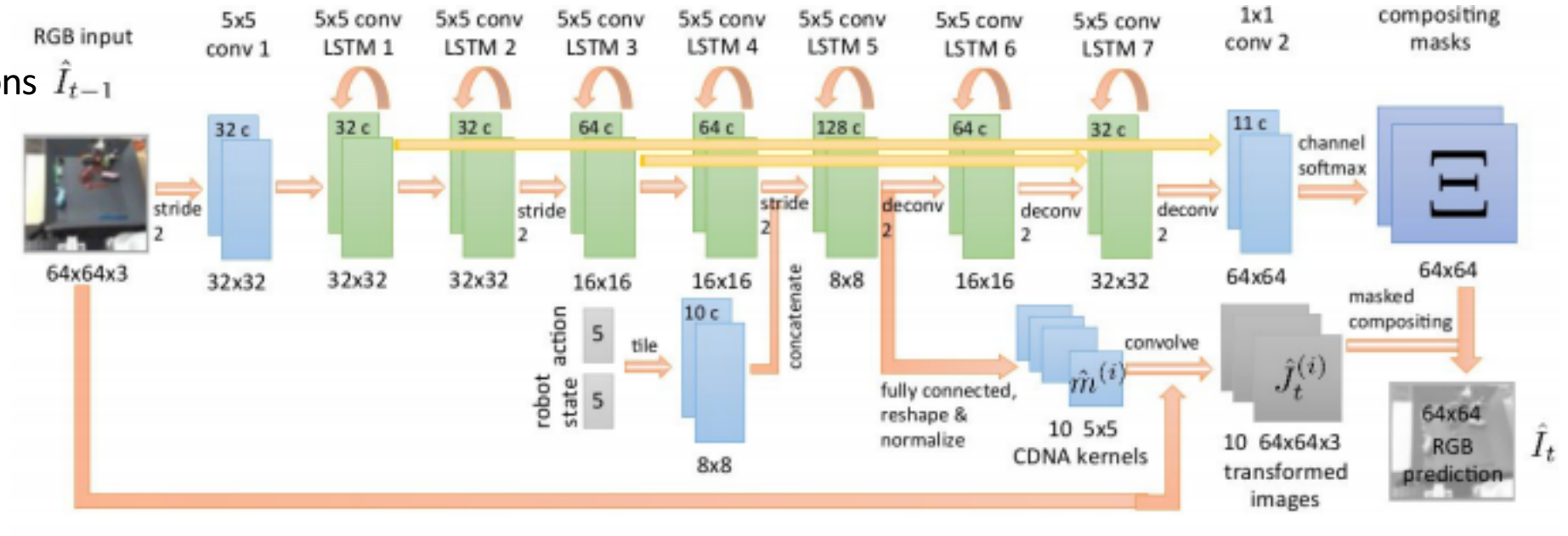
Condition 1 example: Unsupervised learning for physical interaction through video prediction

Three kinds of Transformations

Dynamic neural advection

Convolution

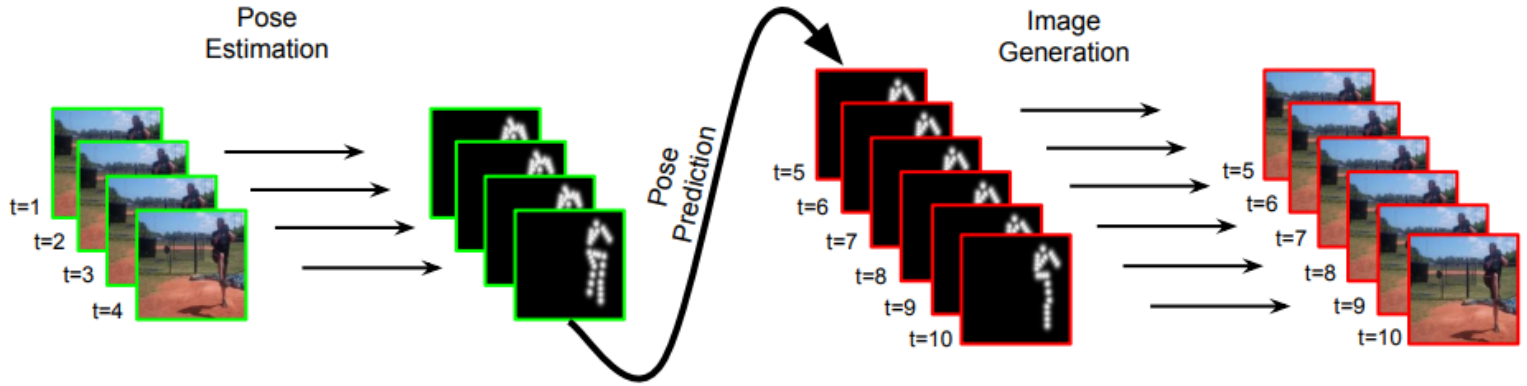
Affine transformation

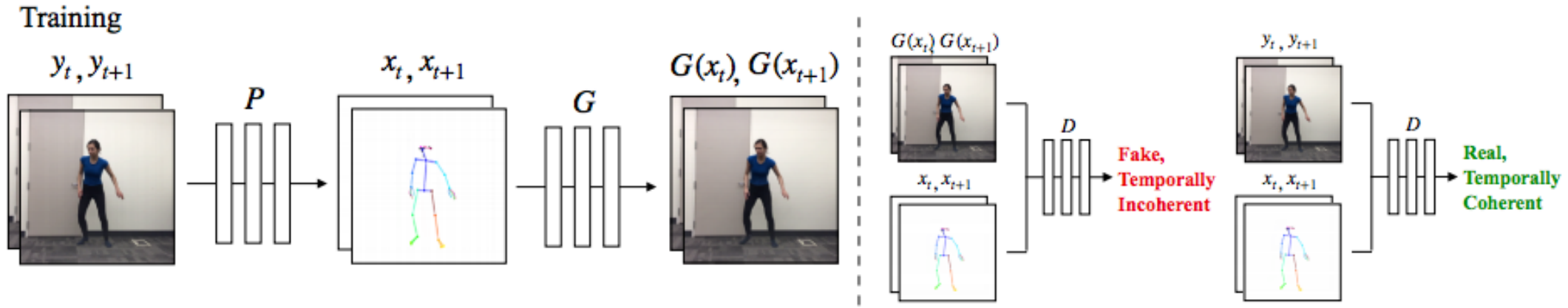


C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In NIPS, 2016.

Motivation : Previous Solutions

Condition 2 examples:





R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in ICML, 2017.
C. Chan, S. Ginosar, T. Zhou, and A. Efros, "Everybody dance now," in arXiv:1808.07371, 2018

Motivation

Previous Methods shows

- **Predict transformation will work** → **Use keypoints guidance to transform the original frame**
- **Decompose the components in a video will work** → **Separate Background and forground objects**
- **Using extra key-points label will work** → **Generate class conditional keypoints in unsupervised manner**

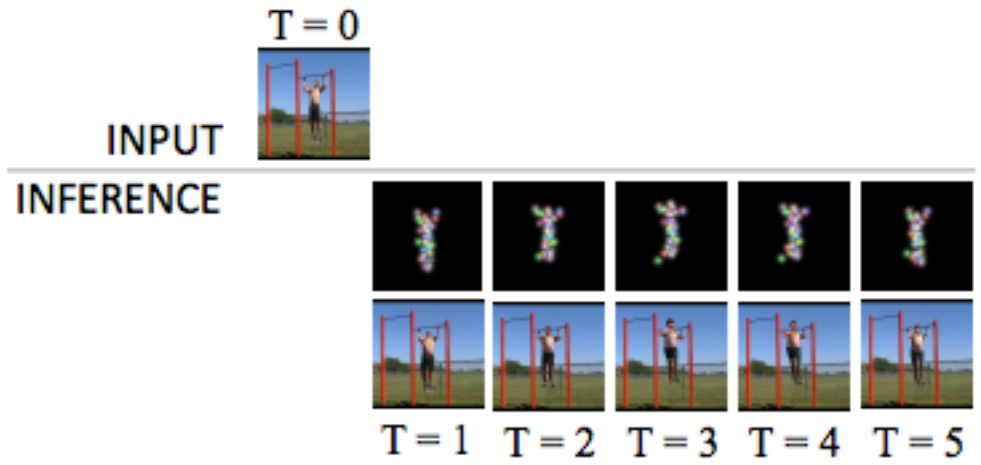
Motivation

Condition3: Pose information +Previous frame + Action guidance

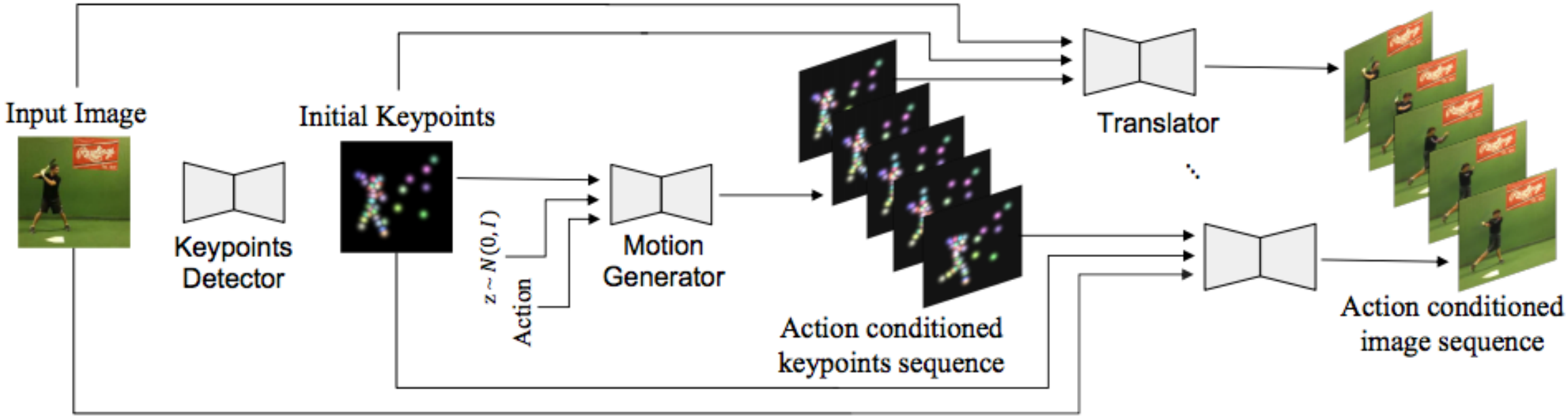
Use keypoints guidance to transform the original frame

Separate Background and forground objects

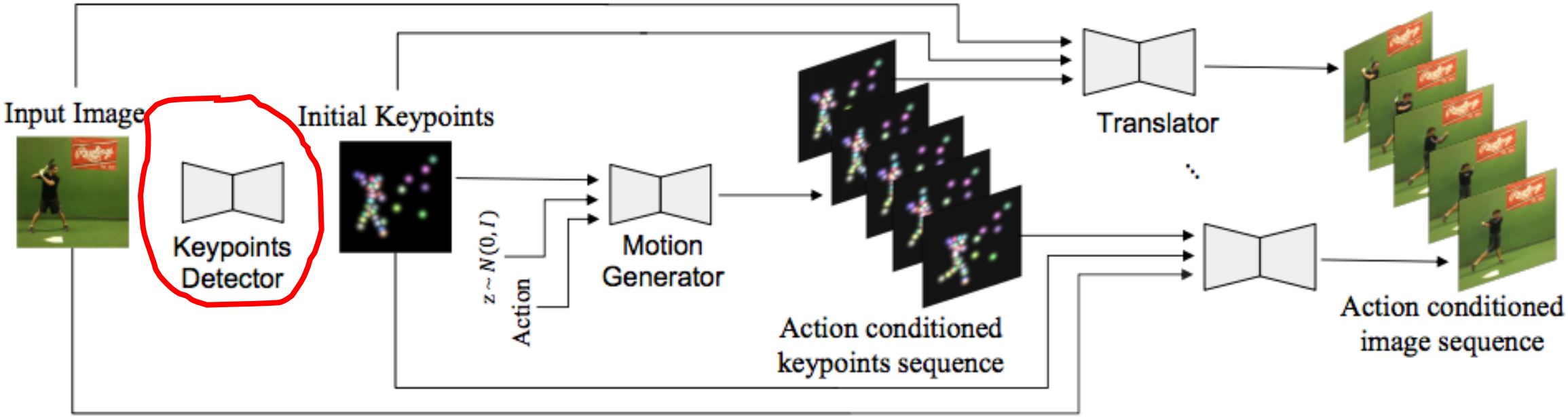
Generate class conditional keypoints in unsupervised manner



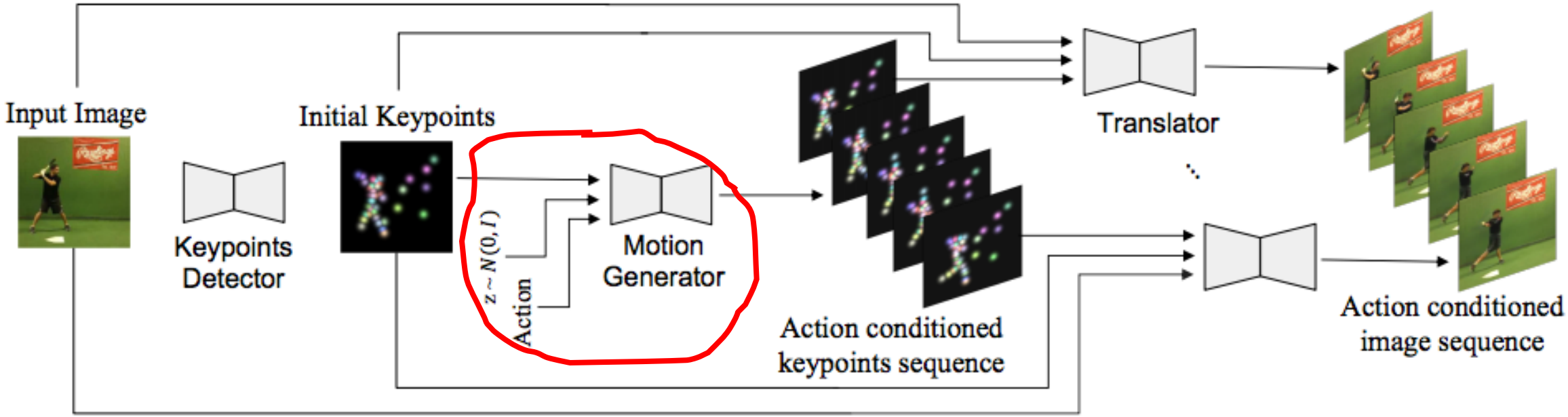
Overview


Input Image
Keypoints Detector
Initial Keypoints
$z \sim N(0, I)$
Action
Motion Generator
Action conditioned keypoints sequence
Translator
Action conditioned image sequence

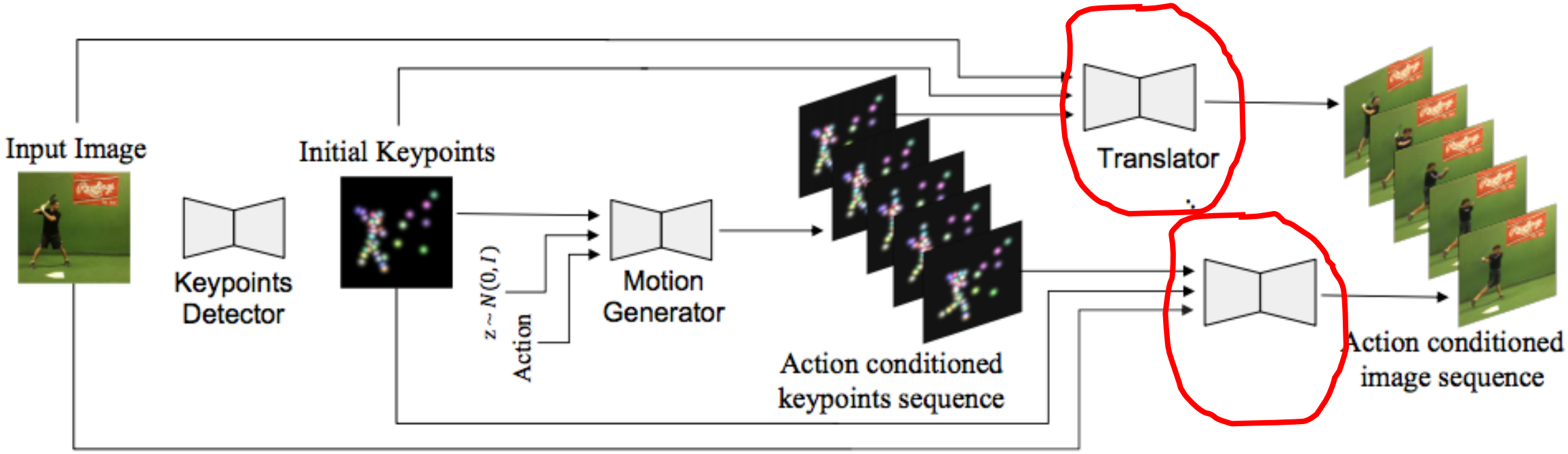
Overview


Input Image
Keypoints Detector
Initial Keypoints
$z \sim N(0, I)$
Action
Motion Generator
Action conditioned keypoints sequence
Translator
Action conditioned image sequence

Overview

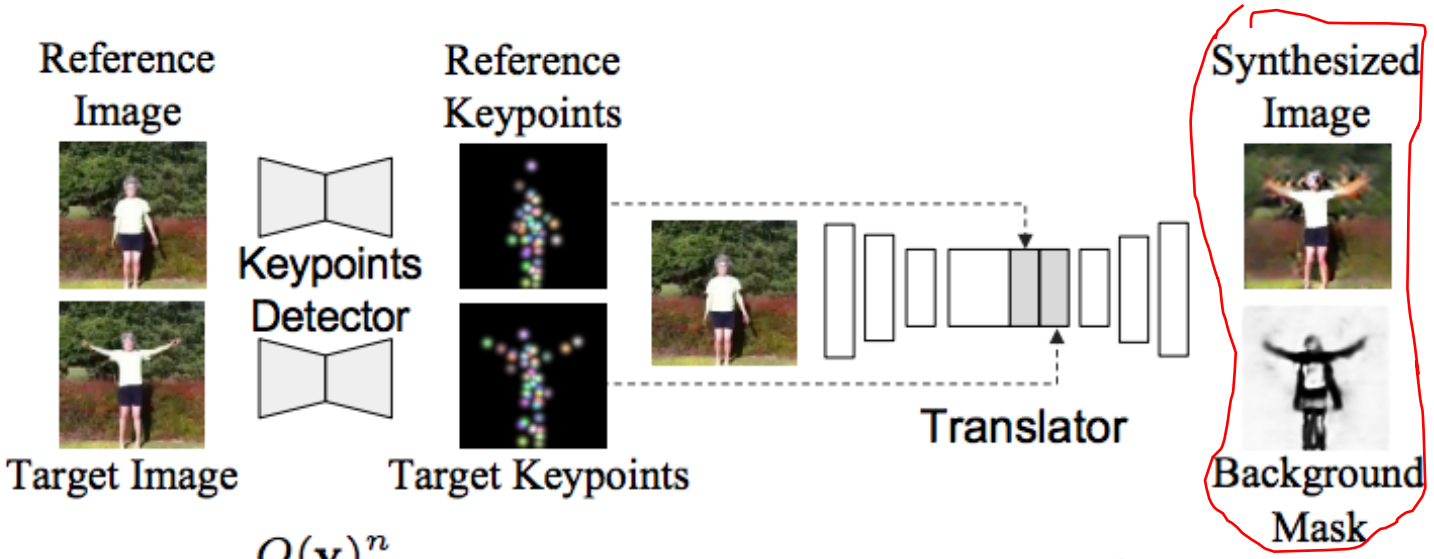

Input Image
Keypoints Detector
Initial Keypoints
$z \sim N(0, I)$
Action
Motion Generator
Action conditioned keypoints sequence
Translator
Action conditioned image sequence

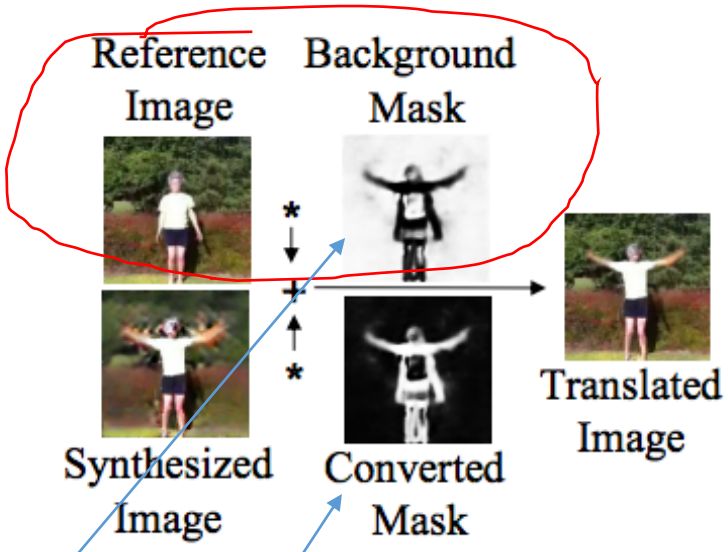
Overview


Input Image
Keypoints Detector
Initial Keypoints
$z \sim N(0, I)$
Action
Motion Generator
Action conditioned keypoints sequence
Translator
Action conditioned image sequence

Proposed Method

Learning the keypoints detector with the image translator





$$\mathbf{l}^n = \frac{e^{Q(\mathbf{v})^n}}{\sum_{\mathbf{u}} e^{Q(\mathbf{v})^n_{\mathbf{u}}}}, \qquad \hat{\mathbf{d}}^n_{\mathbf{u}'} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\mathbf{u}'-\hat{\mathbf{k}}^n\right)^2 / 2\sigma^2},$$

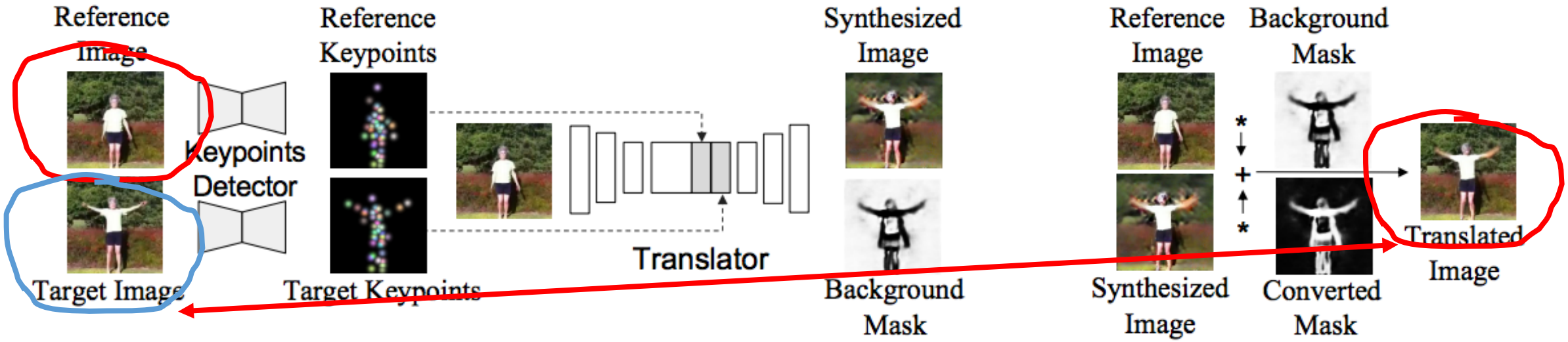
$$\hat{\mathbf{k}}^n = \sum_{\mathbf{u}} \mathbf{u} \cdot \mathbf{l}^n_{\mathbf{u}},$$

$$\hat{\mathbf{v}} = \mathbf{m} \odot \mathbf{v} + (1 - \mathbf{m}) \odot \mathbf{s},$$

Proposed Method

Question1: How to detect key-points in the unsupervised manner?

High Level Idea



detect the keypoints of a foreground object by learning the image translation between two frames (v, v') in the same video

translating v close to v' enforces the network to automatically find the most dynamic parts of the image

Proposed Method

Question1: How to detect key-points in the unsupervised manner?

$$L_{D_{im}} = -\log D_{im}(\mathbf{v}') - \log(1 - D_{im}(\hat{\mathbf{v}}))$$
$$L_{Q,T} = -\log D_{im}(\hat{\mathbf{v}}) + \lambda_1 \mathbb{E}_l \|\Phi_l(\hat{\mathbf{v}}) - \Phi_l(\mathbf{v}')\|,$$

Proposed Method

Question1: How to detect key-points in the unsupervised manner?

$$\mathbf{l}^n = \frac{e^{Q(\mathbf{v})^n}}{\sum_{\mathbf{u}} e^{Q(\mathbf{v})^n_{\mathbf{u}}}},$$

$$\hat{\mathbf{k}}^n = \sum_{\mathbf{u}} \mathbf{u} \cdot \mathbf{l}^n_{\mathbf{u}},$$

```
def get_coord(x, other_axis, axis_size):
    # get "x-y" coordinates:
    g_c_prob = tf.reduce_mean(x, axis=other_axis)  # B,W,NMAP
    g_c_prob = tf.nn.softmax(g_c_prob, axis=1)  # B,W,NMAP
    coord_pt = tf.to_float(tf.linspace(-1.0, 1.0, axis_size))  # W
    coord_pt = tf.reshape(coord_pt, [1, axis_size, 1])
    g_c = tf.reduce_sum(g_c_prob * coord_pt, axis=1)
    return g_c, g_c_prob
```

Input :CxHxW (h represents the y coordinate, w is x)
Step 1: CxW or CxH (when you want to calculate x or y coordinate)

Step2: softmax at the W or H axis to get a soft attention along this axis donates as P_w or P_h

Step3: Matrix multiplication of P_w or P_h with Coordinates (shape is W or H)
CxW x Wx1 or CxH x Hx1
Output is C keypoints' x axis or y axis coordinate, Then concat them we get Cx2 keypoints coordinates



Proposed Method

Question2: Why use Gaussian function to generate heatmap?

$$\hat{\mathbf{d}}_{\mathbf{u}'}^{n} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\mathbf{u}'-\hat{\mathbf{k}}^{n}\right)^{2}/2\sigma^{2}},$$

It is an extremely nonlinear process to make the network output two-dimensional coordinates to optimize the learning directly and the loss function has a weak constraint on the weight

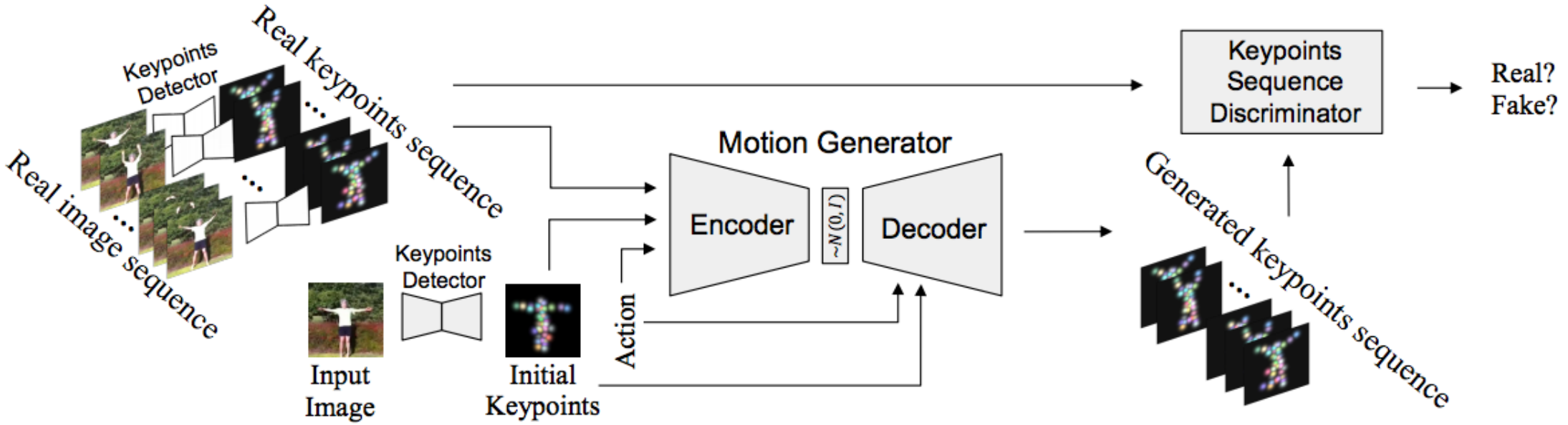
1.make the network full convolutional

2.the corresponding heatmap of an input image has correlation between keypoints
, which can be used to guide the network for learning

3 captures the contrast between the foreground and the background and can also be used to guide the network to learn

Proposed Method

Learning the motion generator with pseudo-labeled data



$$L_{D_{seq}} = -\log D_{seq}(\hat{\mathbf{k}}_{1:T}) - \log(1 - D_{seq}(\tilde{\mathbf{k}}_{1:T}))$$

$$L_M = D_{KL}(q_\phi(\mathbf{z}|\hat{\mathbf{k}}_{1:T}; \hat{\mathbf{k}}_0; \mathbf{a}) \| p_z(\mathbf{z})) + \lambda_2 \|\tilde{\mathbf{k}}_{1:T} - \hat{\mathbf{k}}_{1:T}\|_1 - \lambda_3 \log D_{seq}(\tilde{\mathbf{k}}_{1:T}),$$

Dataset

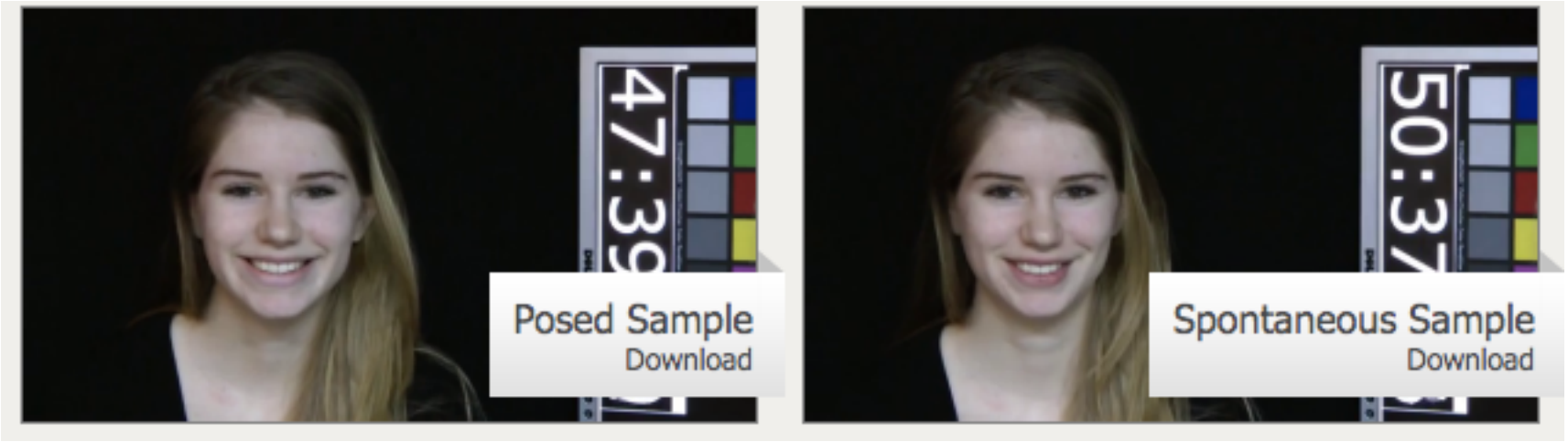
- Penn Action
- UvA-NEMO
- MGIF



Dataset

- Penn Action
- UvA-NEMO
- MGIF

1234 videos of smiling human faces,
which is split into 1110 videos for the training set 124 for the evaluation set



Dataset

- Penn Action
- UvA-NEMO
- MGIF

videos of cartoon animal characters simply walking or running on a white colored background.
For this dataset, 900 videos are used for the training and 100 videos are used for the evaluation



Results

Evaluation Measures	
PSNR	measures the peak signal-to-noise ratio between two monochrome images I and K to assess the quality of a generated image compared to its corresponding real image
SSIM	a well-characterized perceptual similarity measure that aims to discount aspects of an image that are not important for human perception
Inception Score	KLD between conditional and marginal label distributions over generated data
Fréchet Inception Distance	Wasserstein-2 distance between multi-variate Gaussians fitted to data embedded into a feature

https://arxiv.org/pdf/1802.03446.pdf

Results

Quantitative results

Dataset	[39]	[16]	[21]	Ours
Penn Action [25]	4083.3	3324.9	2187.5	**1509.0**
UvA-NEMO [38]	666.9	265.2	-	**162.4**
MGIF [31]	683.1	1079.6	-	**409.1**

Table 1: Fréchet Video Distance (FVD) [40] of generated videos. On every datasets, our method achieved the best score. (The lower is better.)

Results

Quantitative results

Method	Accuracy
Ours	**68.89**
Ours w/o a	63.33
[21]	47.14
[16]	40.00
[39]	15.55

Table 2: Action recognition accuracy.

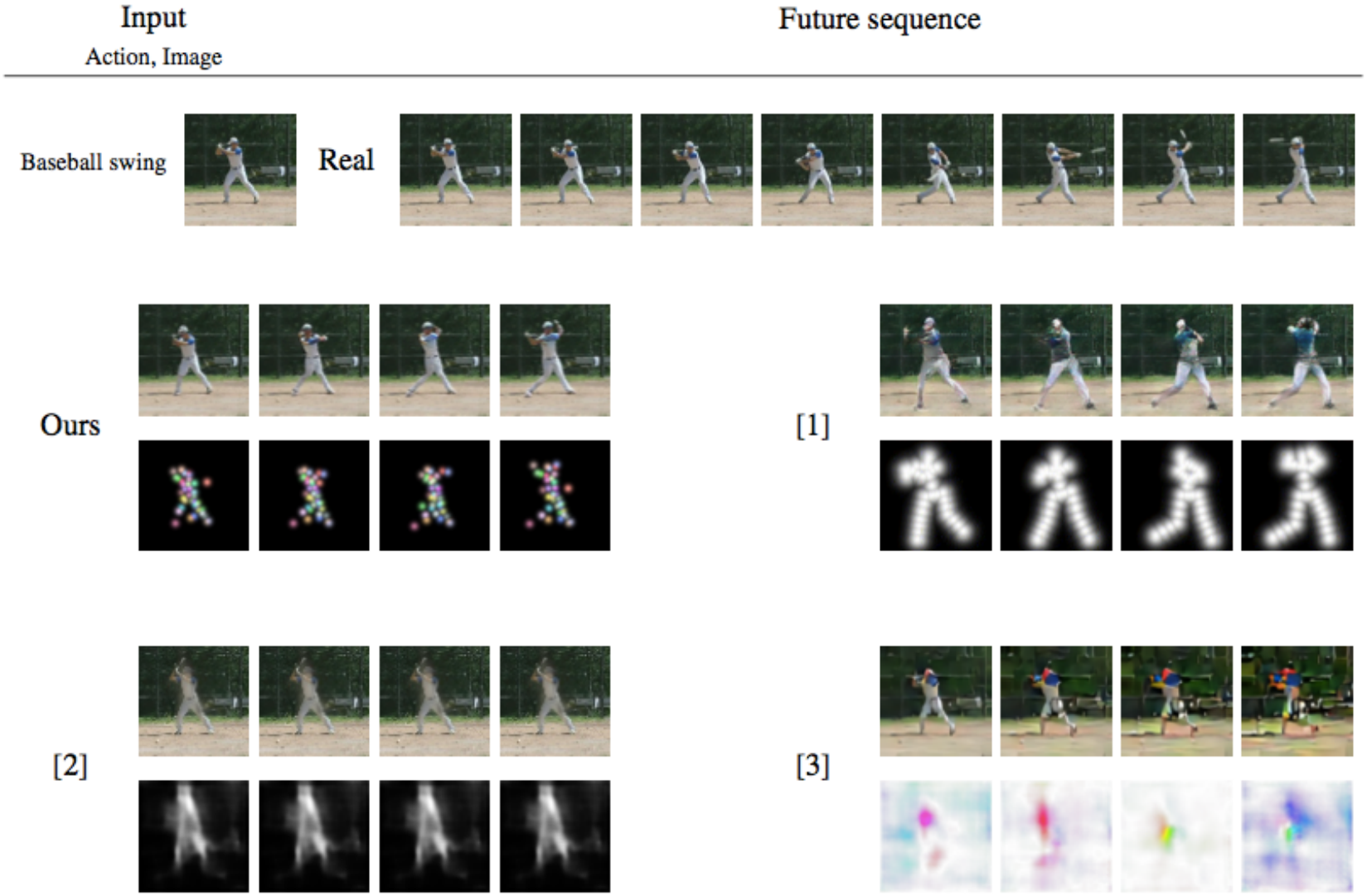
Method	Ranking
Ours	$\mathbf{1.81} \pm 1.02$
[21]	2.44 ± 0.98
[16]	3.14 ± 1.09
[39]	2.61 ± 0.96

Table 3: Quantitative result of the user study. The values refer to average rankings.



Results

Qualitative results



Results

Qualitative results

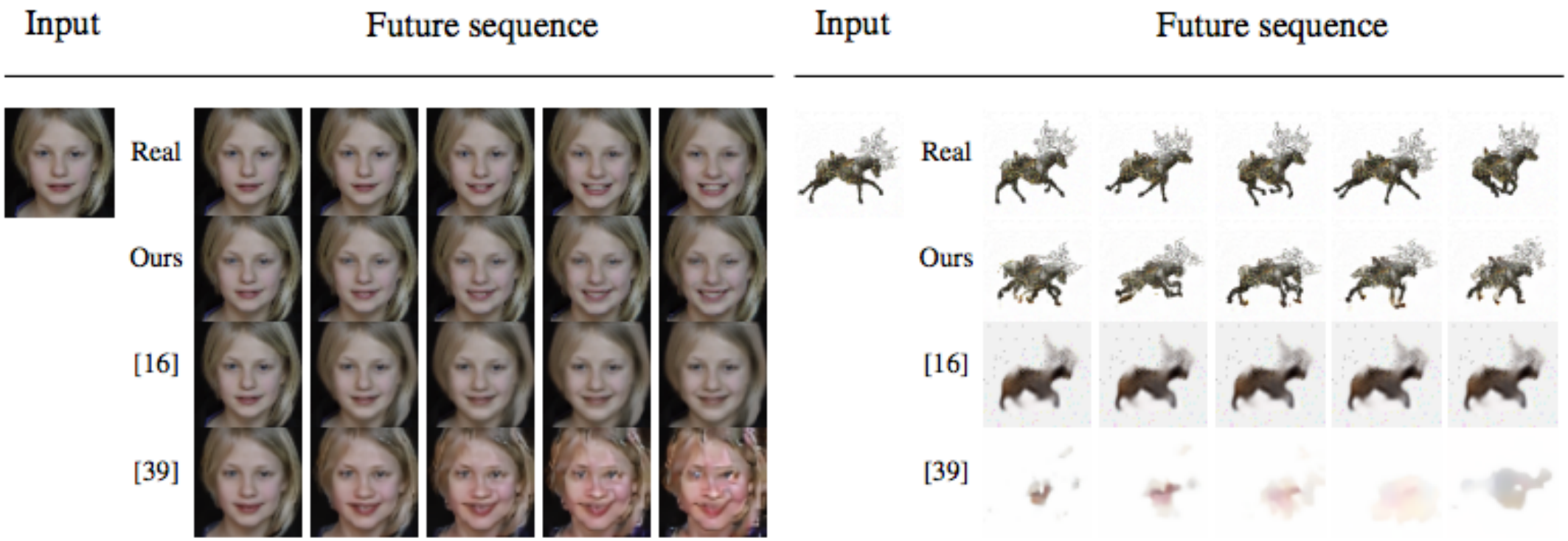


Figure 7: Video prediction results on UvA-NEMO and MGIF datasets.

Results

Compare with image synthesis

SinGan:



BigGan:



R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in ICML, 2017.
C. Chan, S. Ginosar, T. Zhou, and A. Efros, "Everybody dance now," in arXiv:1808.07371, 2018

Results

Compare with image synthesis









Conclusions

This paper proposes an action–conditional video prediction approach based on unsupervised keypoint detection

(1)detects keypoints from an input image

(2)generates a sequence of future keypoints of the given action category

(3) translates each keypoint frame into the target RGB image

Questions?



Thank You!