# Disentangling Content and Pose with an Adversarial Loss

## CVPR2018 GAN Tutorial

**Emily Denton**
**Department of Computer Science**
**New York University**

박사과정 김성빈 chengbinjin@inha.edu,
지도교수 김학일 교수 hikim@inha.ac.kr
인하대학교 컴퓨터비전 연구실

2018.07.31

# Content

- **Part I:** Disentangling content and pose with an adversarial loss
  - E. Denton, et al., Unsupervised Learning of Disentangled Representations from Video, NIPS2017
  - pp.1~49
  - 2 reference papers

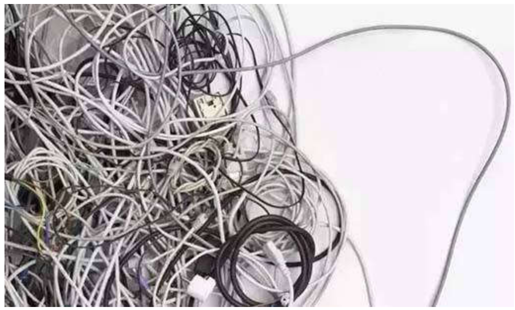
- **Part II:** Survey of adversarial losses in feature space
  - pp. 50~59
  - 13 reference papers

# Unsupervised Learning of Disentangled Representations from Video

## NIPS2017

# Unsupervised Learning of Disentangled Representations from Video

## NIPS2017



Video

Disentangle

Content

Pose

Disentangled representation

# Disentangled Representation Net (DrNet)

- Disentangling auto-encoder that factorizes image sequences into **temporally constant (content)** and **temporally varying (pose)** components

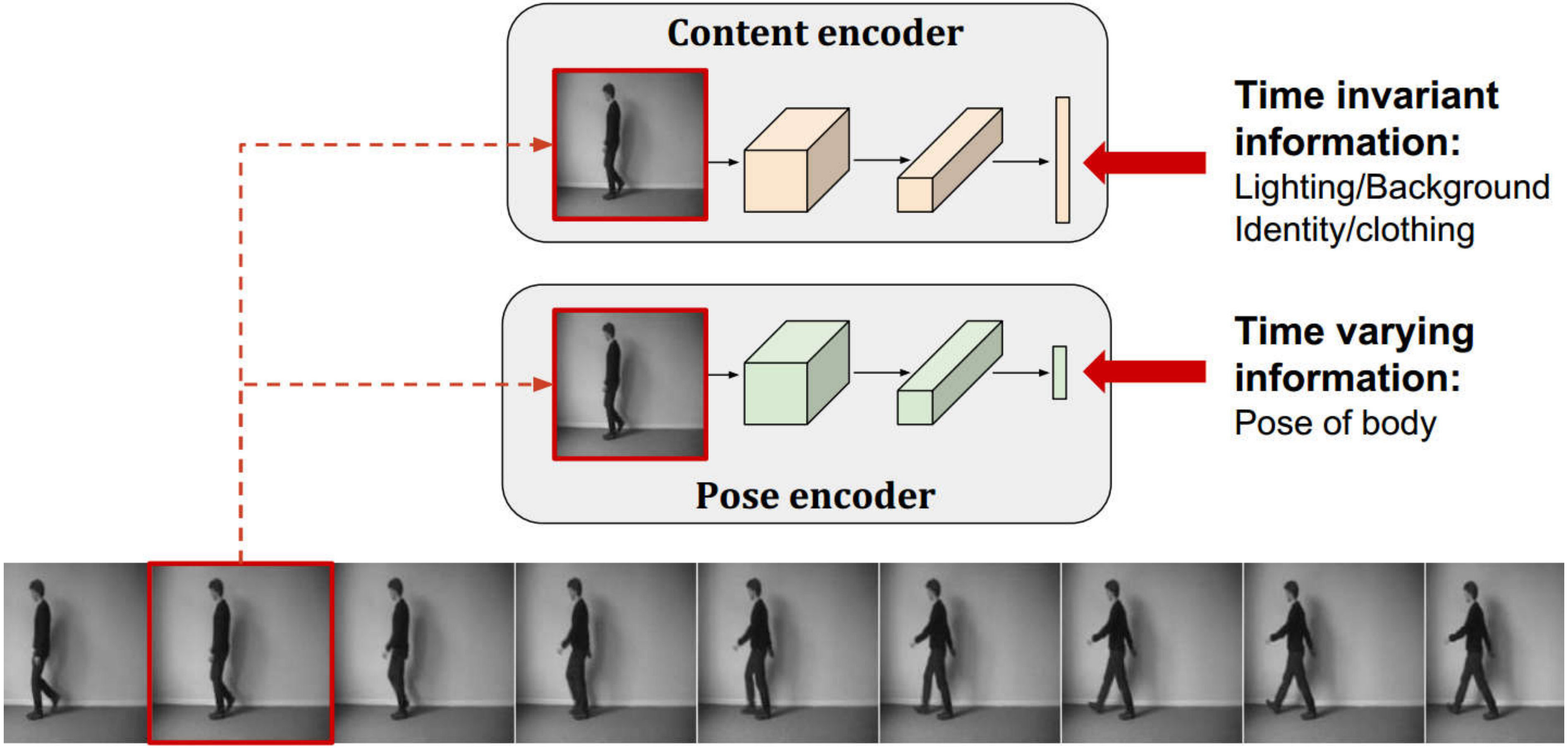
**Time varying information:** Pose of body

**Time invariant information:** Lighting, background, identity, clothing

**Assumption:** Simple background

# DrNet: Two Separate Encoders

- $|h_c|$ = 128 (MNIST, NORB, SUNCG, KTH)
- $|h_p|$ = 5 (MNIST, KTH), 10 (NORB, SUNCG)



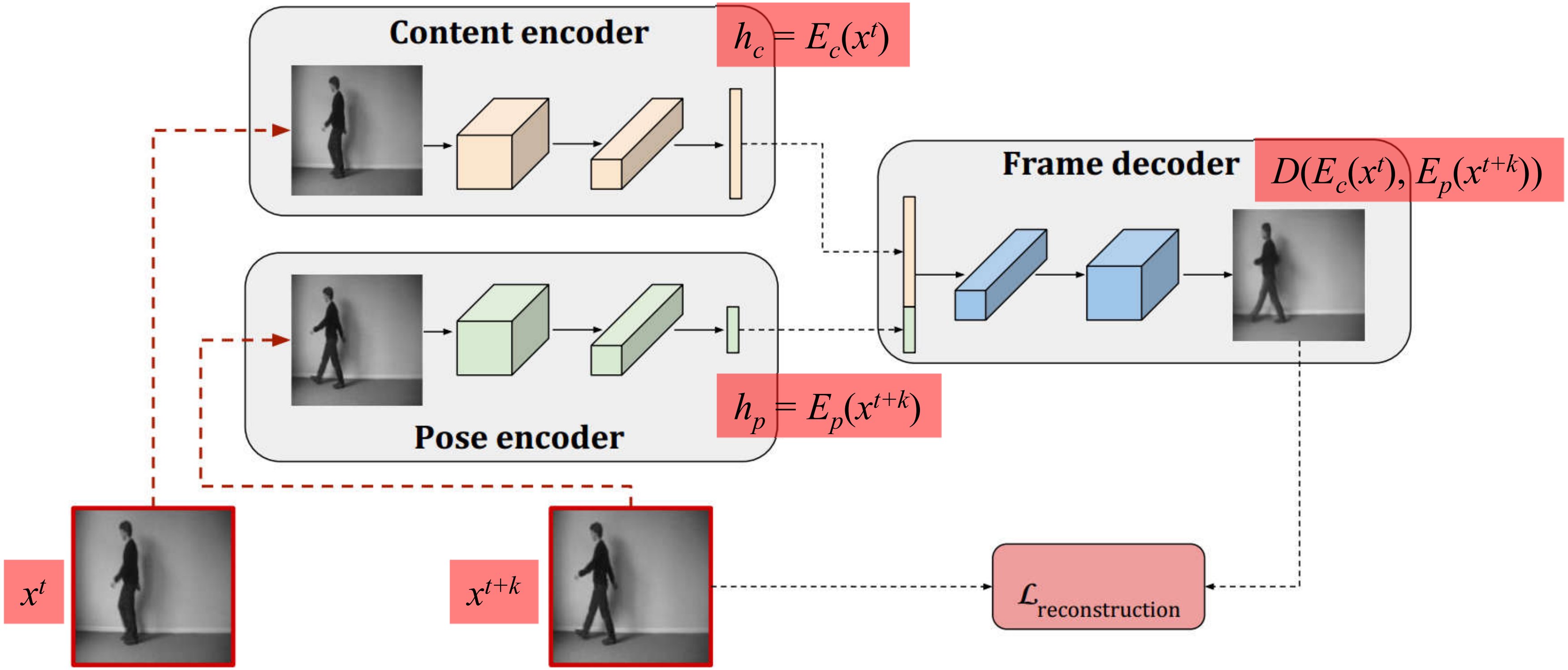

**Time invariant information:**
Lighting/Background
Identity/clothing

**Time varying information:**
Pose of body

# DrNet: Training

- **Reconstruction loss** drives training
- **Similarity loss** makes content vectors invariant across time
- **Adversarial loss** enforces pose vectors to only contain info that changes across time

# I. Reconstruction Loss




$$L_{reconstruction}\left(E_c, E_p, D\right) = \left\| D\left(E_c\left(x^t\right), E_p\left(x^{t+k}\right)\right) - x^{t+k} \right\|_2^2 \quad (1)$$

- $E_c$: content encoder
- $E_p$: pose encoder
- $D$: decoder

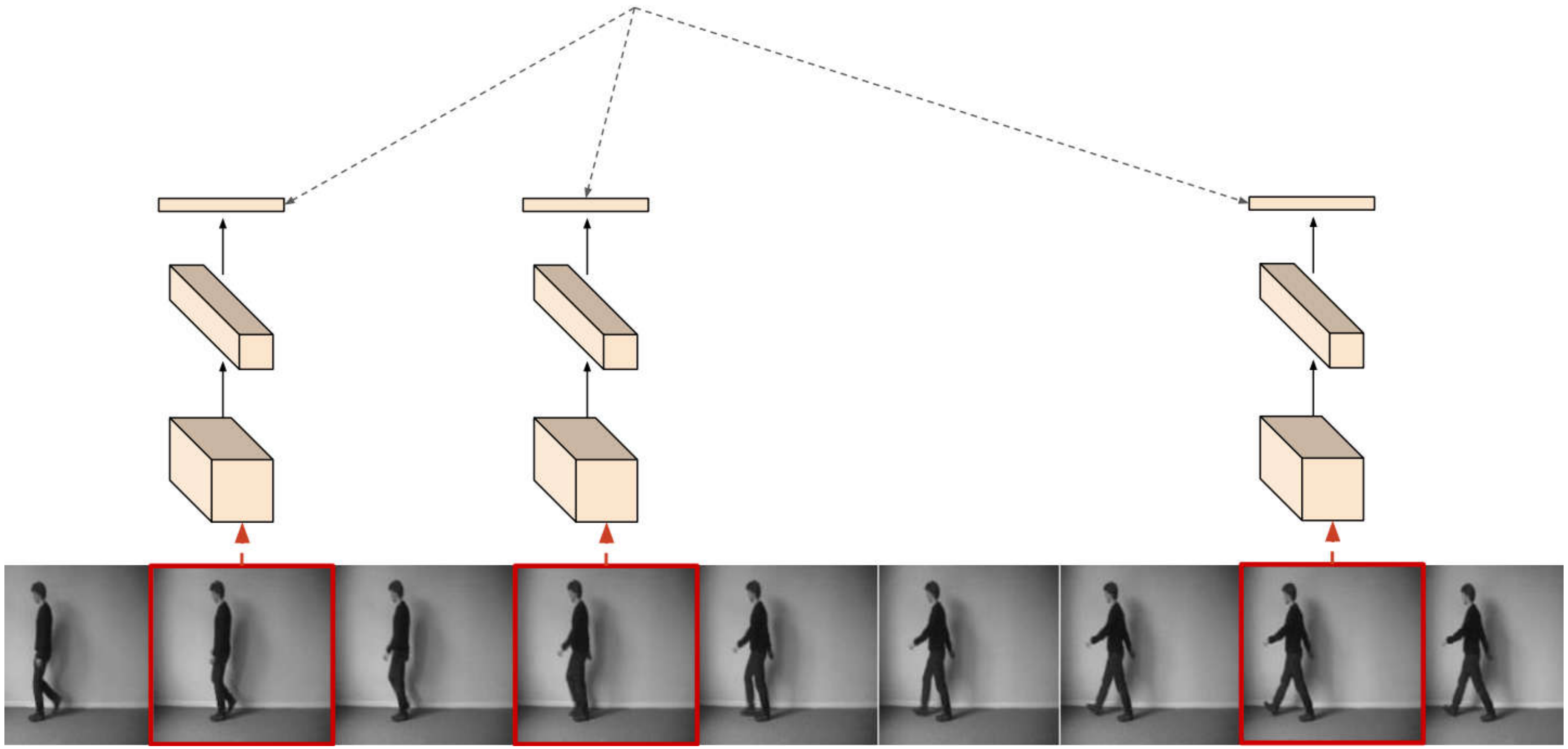
- $x^t$: input frame of index $t$
- $x^{t+k}$ : input frame of index $t+k$
- $k$: random frame offset $k \in [0, K]$

# II. Similarity Loss [1/2]

- **Similarity loss** makes content vectors invariant across time

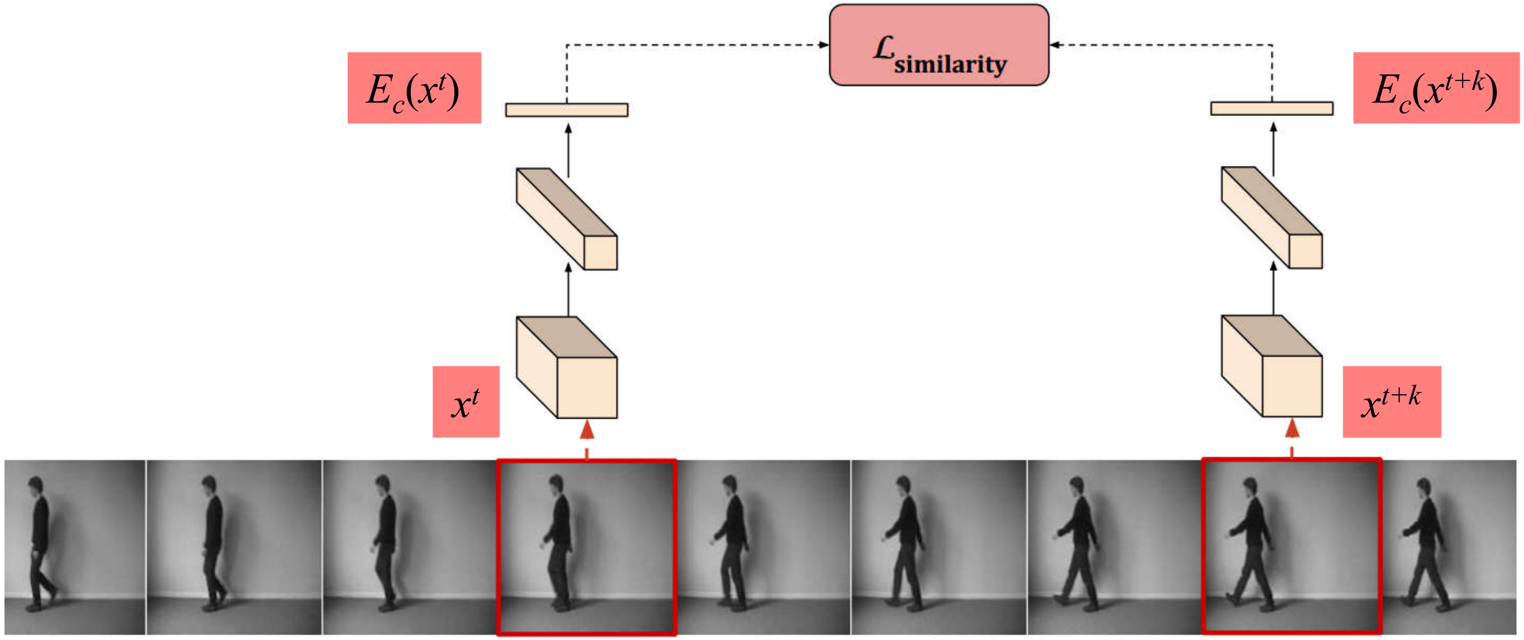
**Time invariant information:**
Lighting, background, identity, clothing

Content vectors should be invariant across time



# II. Similarity Loss [2/2]

- l2 similarity loss on temporally nearby content vectors

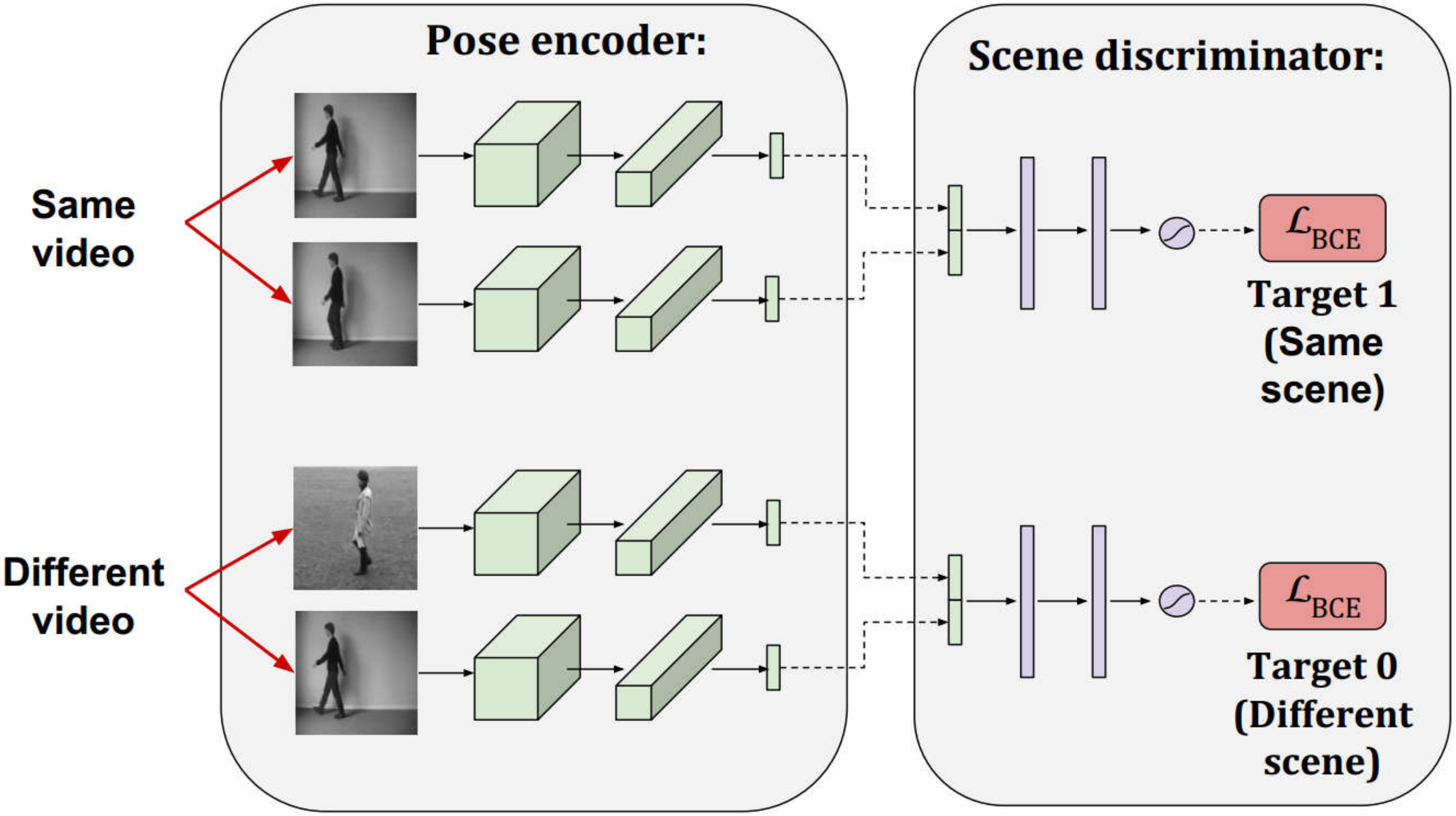



$$L_{similarity}(E_c) = \left\| E_c(x^t) - E_c(x^{t+k}) \right\|_2^2 \quad (2)$$

- $E_c$: content encoder
- $x^t$: input frame of index $t$
- $x^{t+k}$ : input frame of index $t+k$
- $k$: random frame offset $k \in [0, K]$

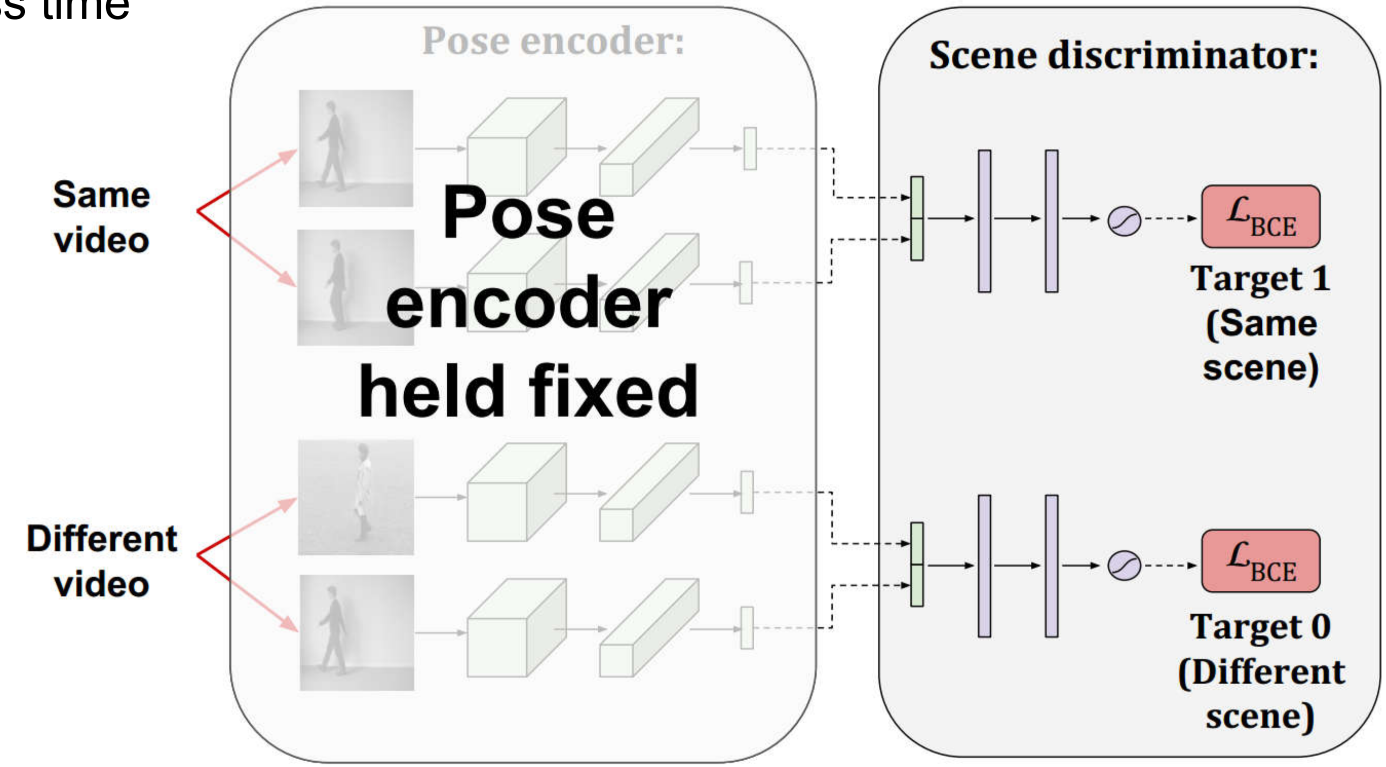
# III. Adversarial Loss [1/3]

- **Adversarial loss** enforces pose vectors to only contain info that changes across time



# III. Adversarial Loss [2/3]

- **Adversarial loss** enforces pose vectors to only contain info that changes across time

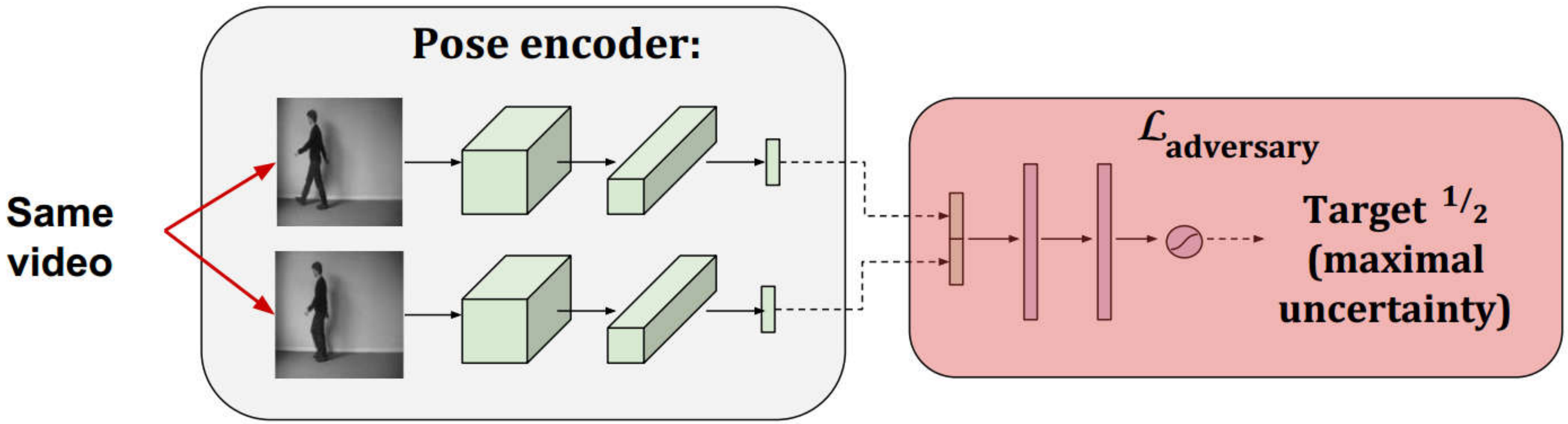



$$-L_{adversarial}(C) = \log\left(C\left(E_p\left(x_i^t\right), E_p\left(x_i^{t+k}\right)\right)\right) + \log\left(1 - C\left(E_p\left(x_i^t\right), E_p\left(x_j^{t+k}\right)\right)\right) \quad (3)$$

- *C*: scene discriminator
- $E_p$: pose encoder
- $x_i^t$: frame *t* of the input video clip *i*
- $x_i^{t+k}$: frame *t*+*k* of the input video clip *i*
- $x_j^{t+k}$: frame *t*+*k* of the input video clip *j*
- *k*: random frame offset $k \in [0, K]$

# III. Adversarial Loss [3/3]

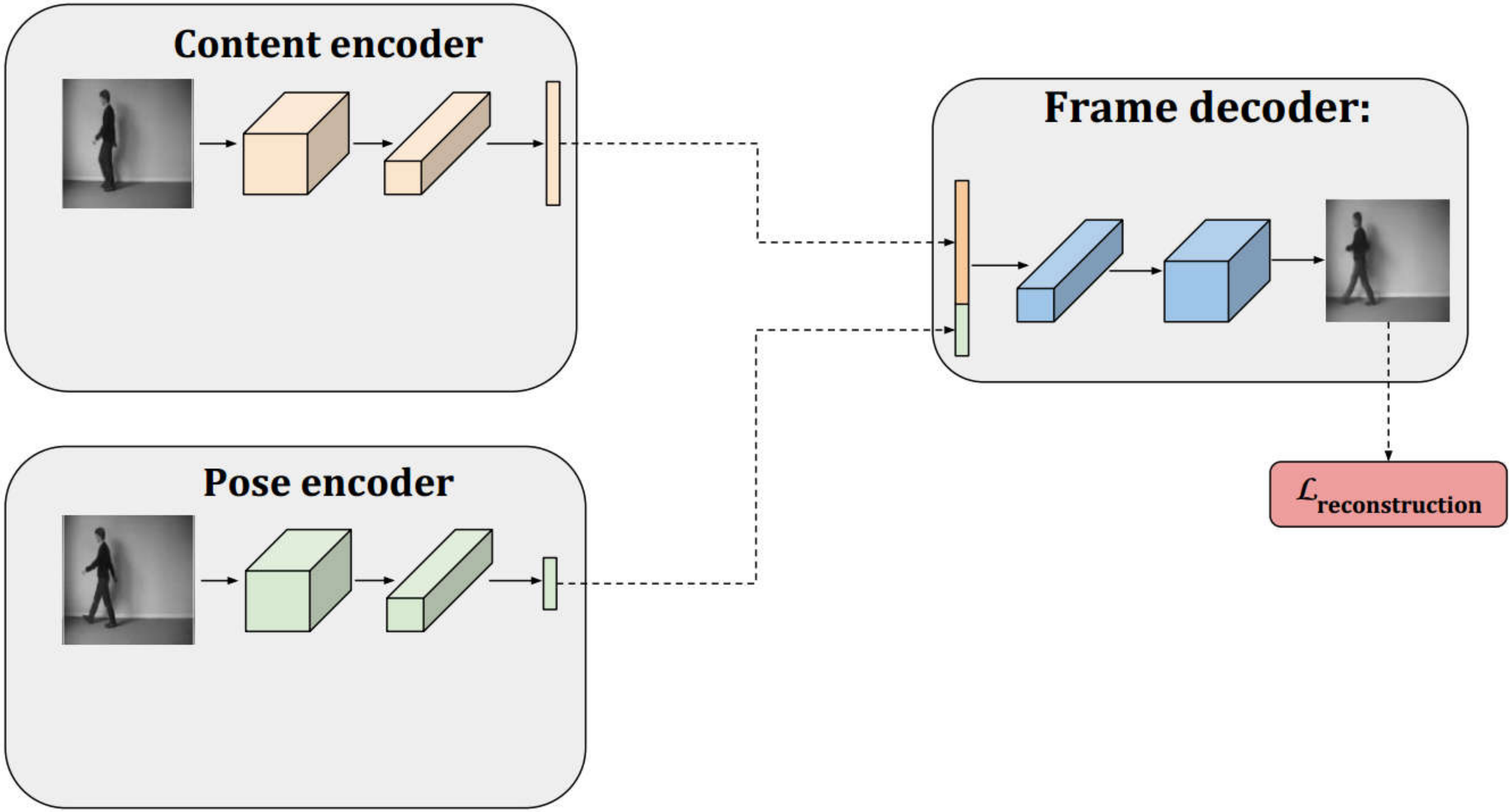
- Train pose encoder to produce pose vectors that make the discriminator **maximally uncertain** about the content of the video




$$-L_{adversarial}\left(E_p\right)=\frac{1}{2}\log\left(C\left(E_p\left(x_i^t\right),E_p\left(x_i^{t+k}\right)\right)\right)+\frac{1}{2}\log\left(1-C\left(E_p\left(x_i^t\right),E_p\left(x_i^{t+k}\right)\right)\right) \quad (4)$$

- *C*: scene discriminator
- $E_p$: pose encoder
- $x_i^t$: frame *t* of the input video clip *i*
- $x_i^{t+k}$: frame *t*+*k* of the input video clip *i*
- *k*: random frame offset $k \in [0, K]$

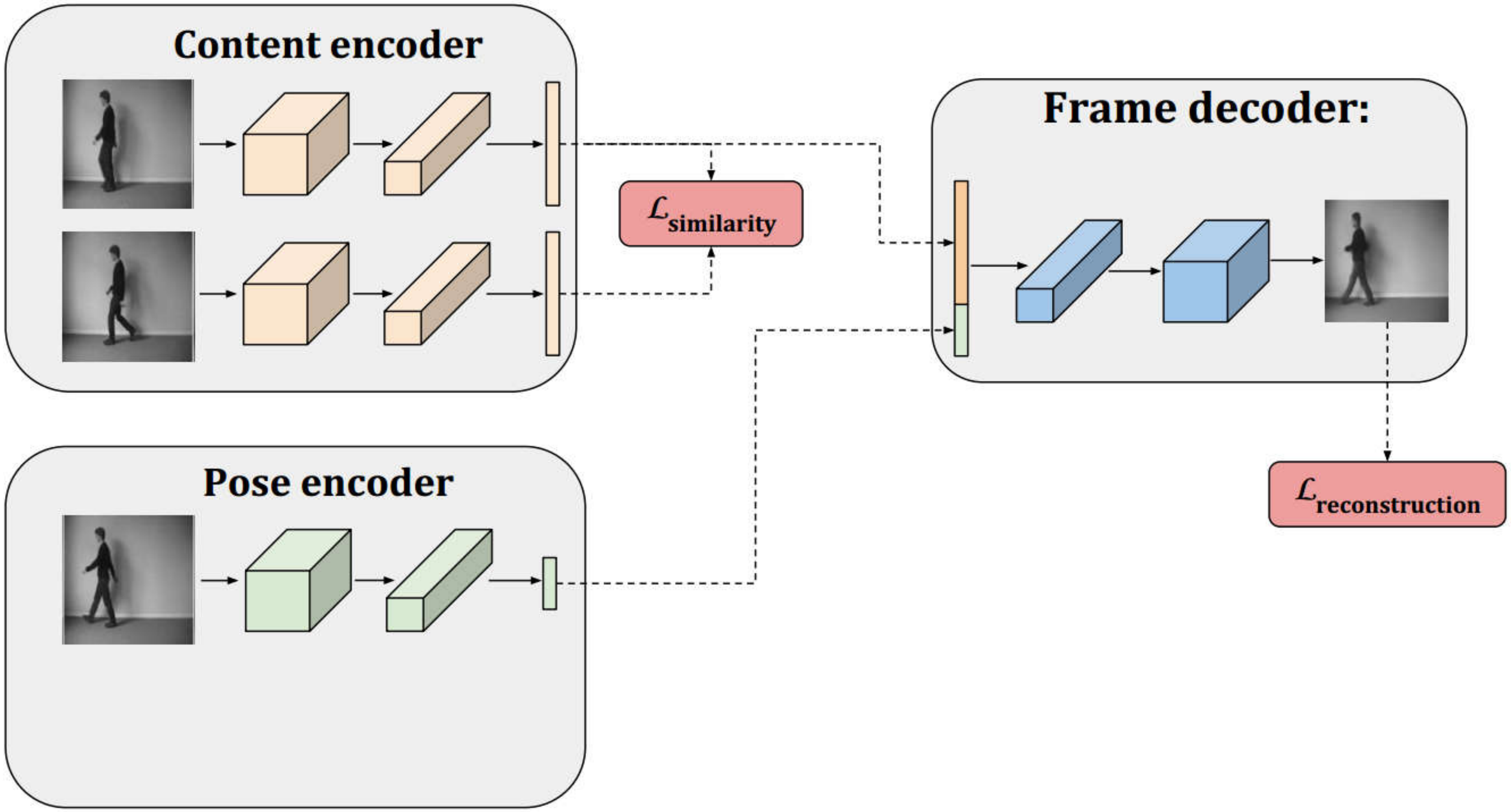
# Overall Training Objective [1/3]




$$L = L_{reconstruction}\left(E_c, E_p, D\right) + \alpha L_{similarity}\left(E_c\right) + \beta\left(L_{adversarial}\left(E_p\right) + L_{adversarial}\left(C\right)\right) \quad (5)$$

- $\alpha$=1 for all datasets
- $\beta$=0.1 for MNIST, NORB and SUNCG and $\beta$=0.0001 for KTH experiments

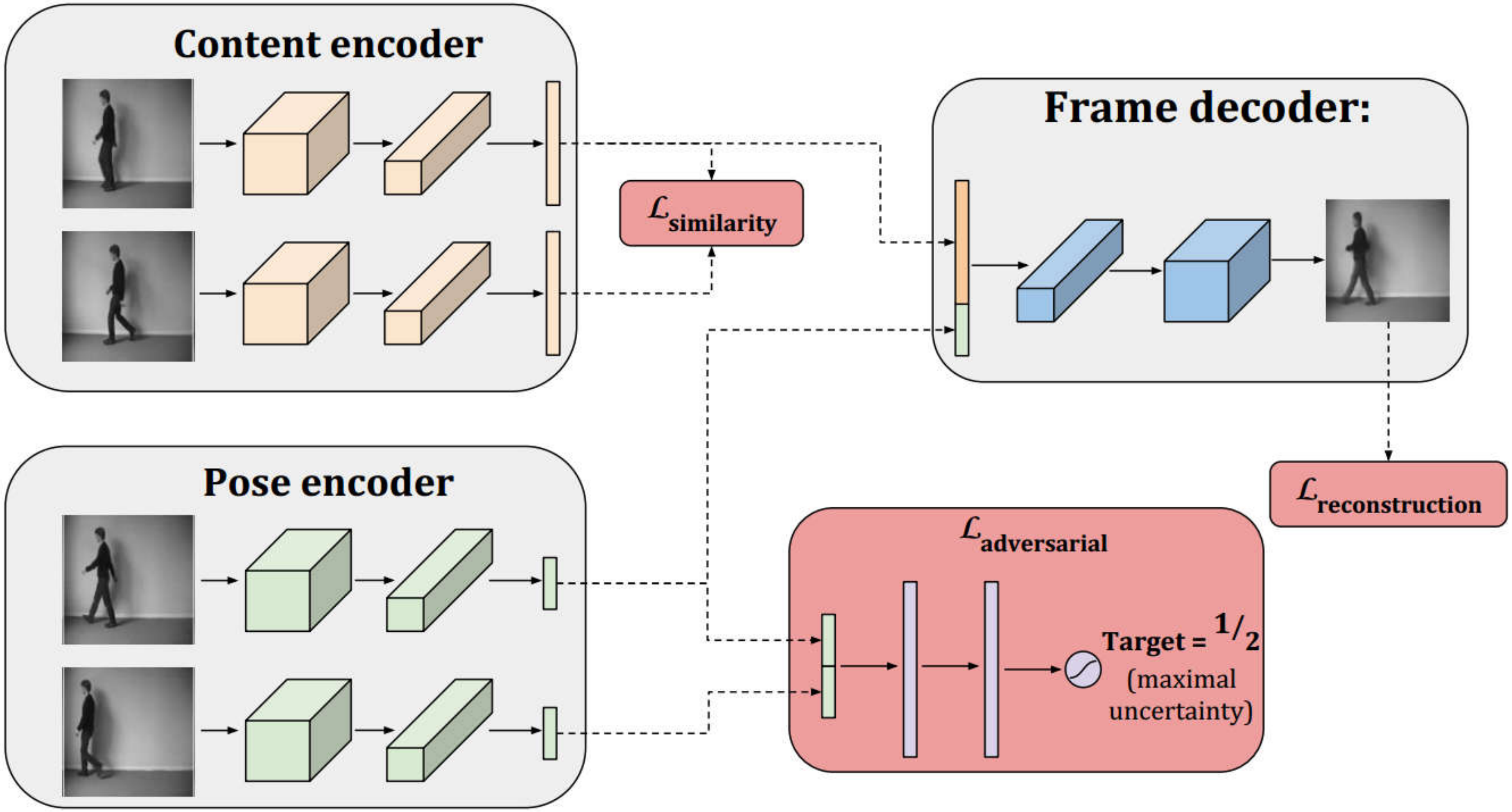
# Overall Training Objective [2/3]




$$L = L_{reconstruction}\left(E_c, E_p, D\right) + \alpha L_{similarity}\left(E_c\right) + \beta\left(L_{adversarial}\left(E_p\right) + L_{adversarial}\left(C\right)\right) \quad (5)$$

- $\alpha=1$ for all datasets
- $\beta=0.1$ for MNIST, NORB and SUNCG and $\beta=0.0001$ for KTH experiments

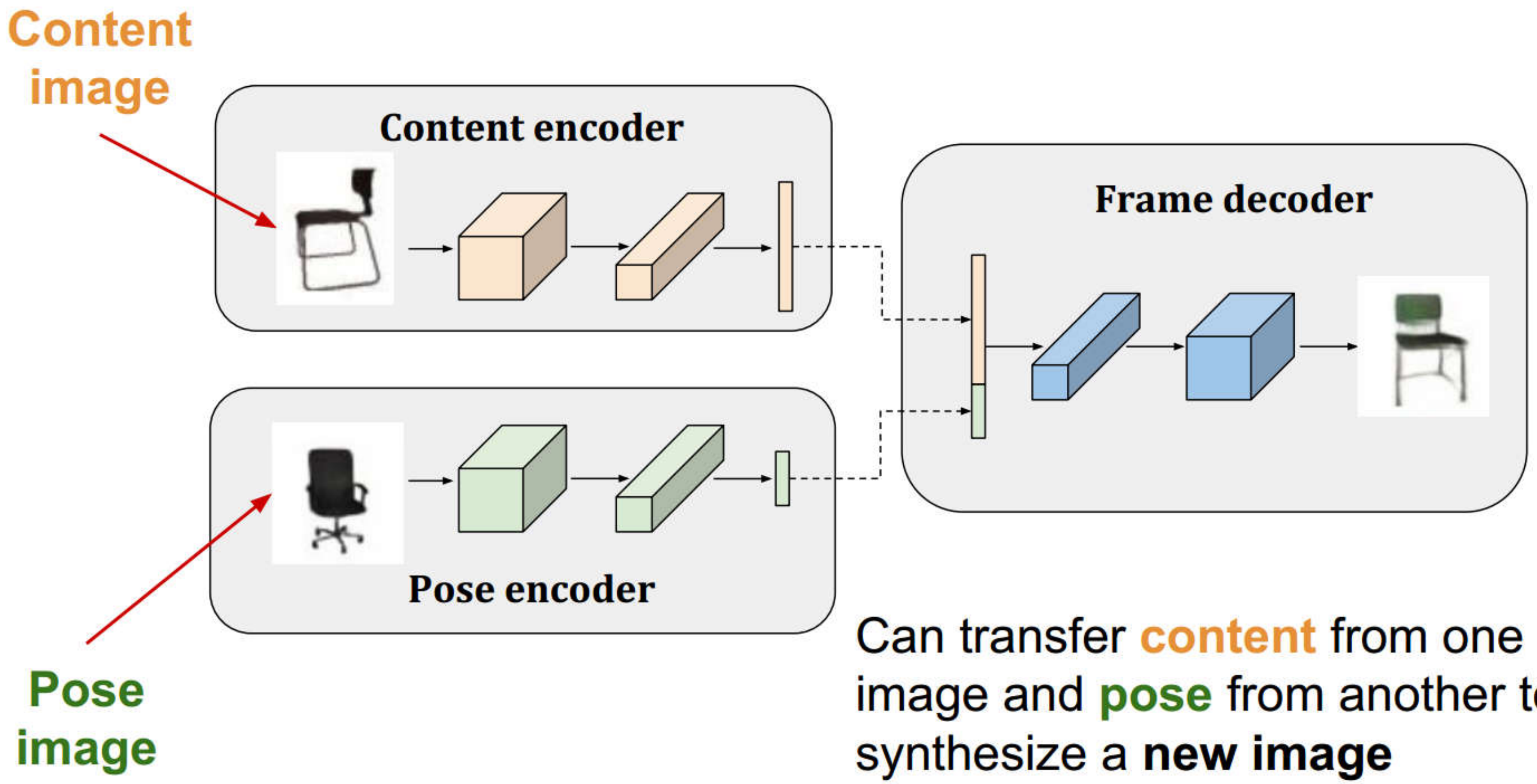
# Overall Training Objective [3/3]




$$L = L_{reconstruction}\left(E_c, E_p, D\right) + \alpha L_{similarity}\left(E_c\right) + \beta\left(L_{adversarial}\left(E_p\right) + L_{adversarial}\left(C\right)\right) \quad (5)$$

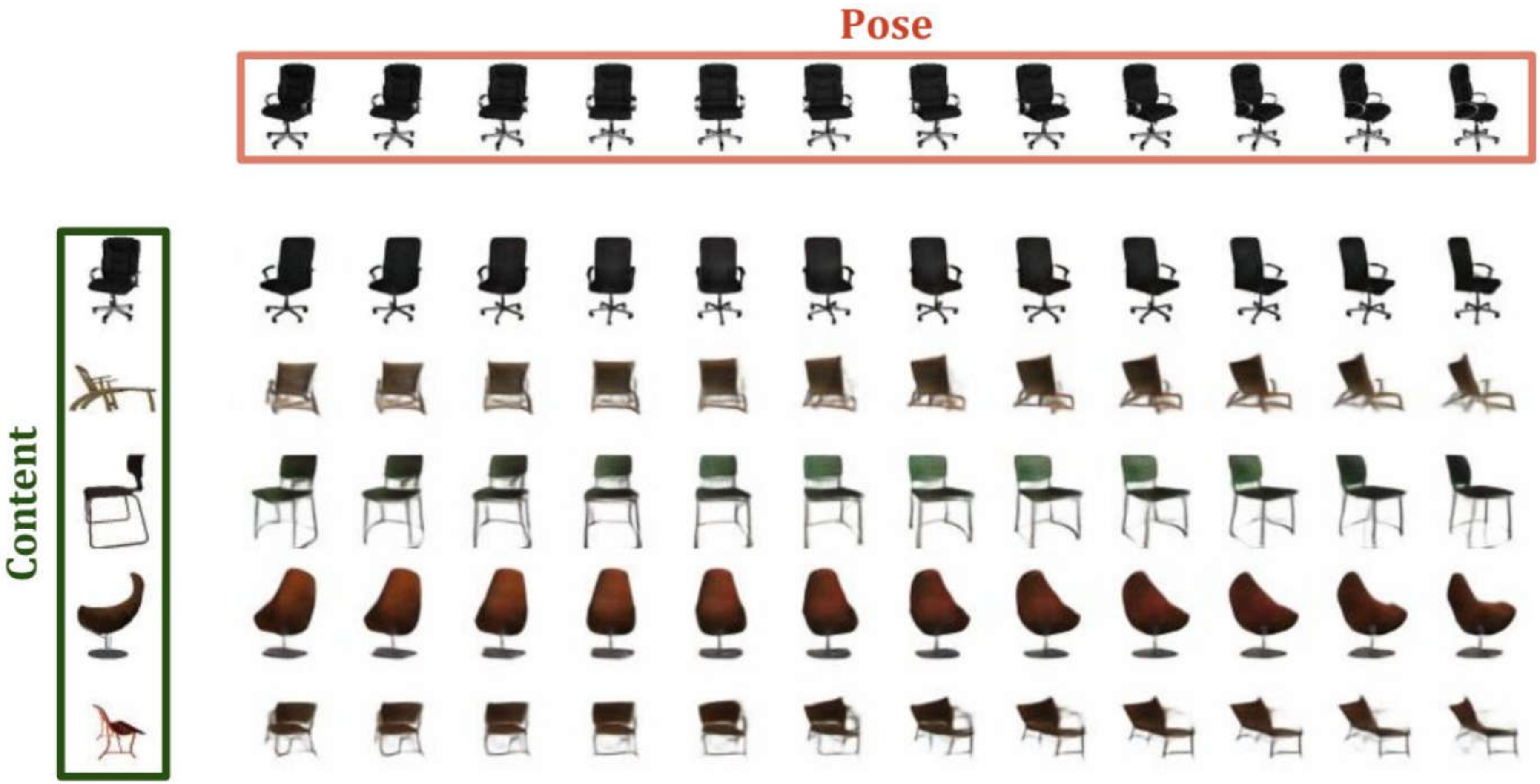
- $\alpha$=1 for all datasets
- $\beta$=0.1 for MNIST, NORB and SUNCG and $\beta$=0.0001 for KTH experiments

# Image Synthesis by Analogy [1/4]




Can transfer **content** from one image and **pose** from another to synthesize a **new image**

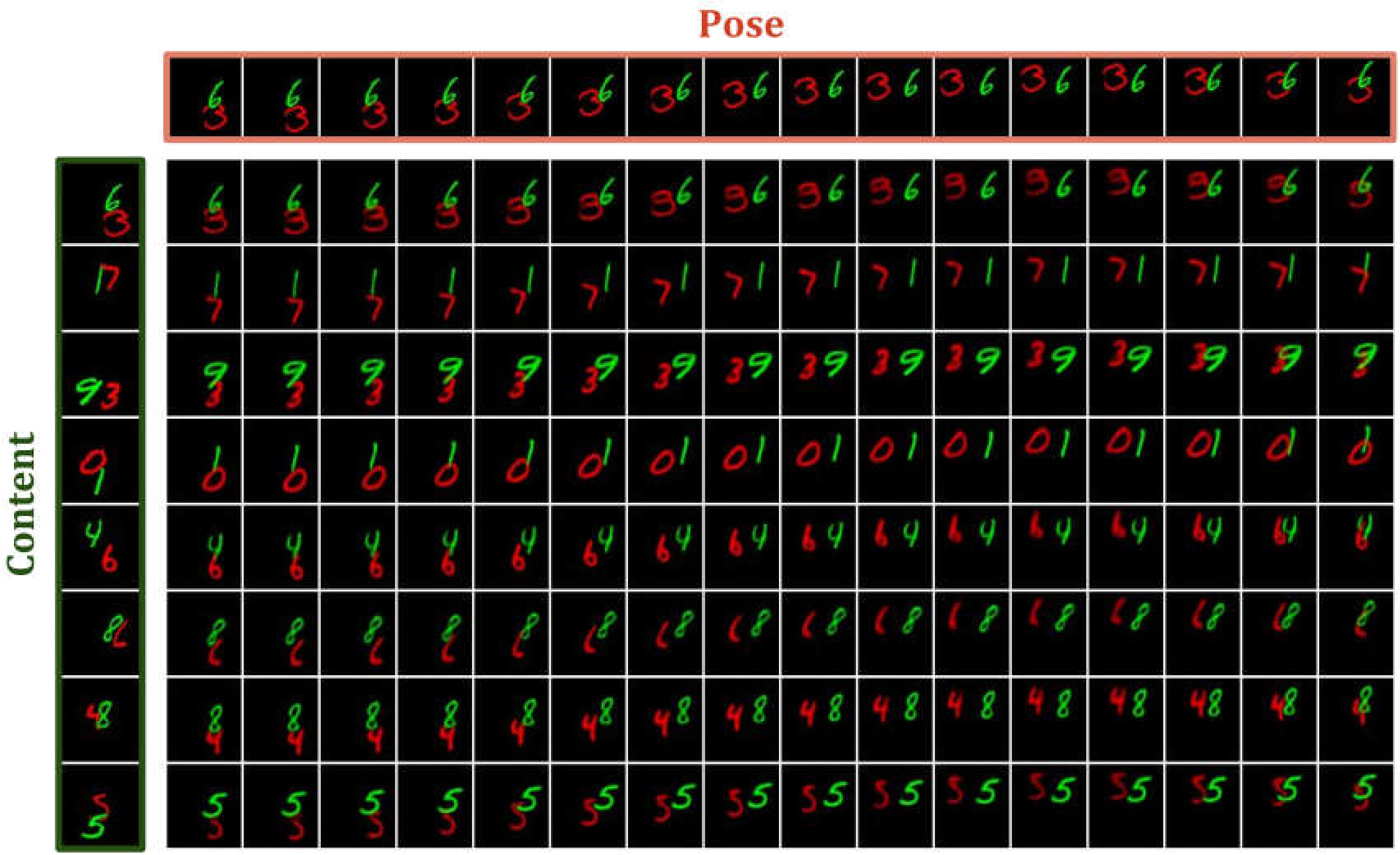
# Image Synthesis by Analogy [2/4]



# Image Synthesis by Analogy [3/4]

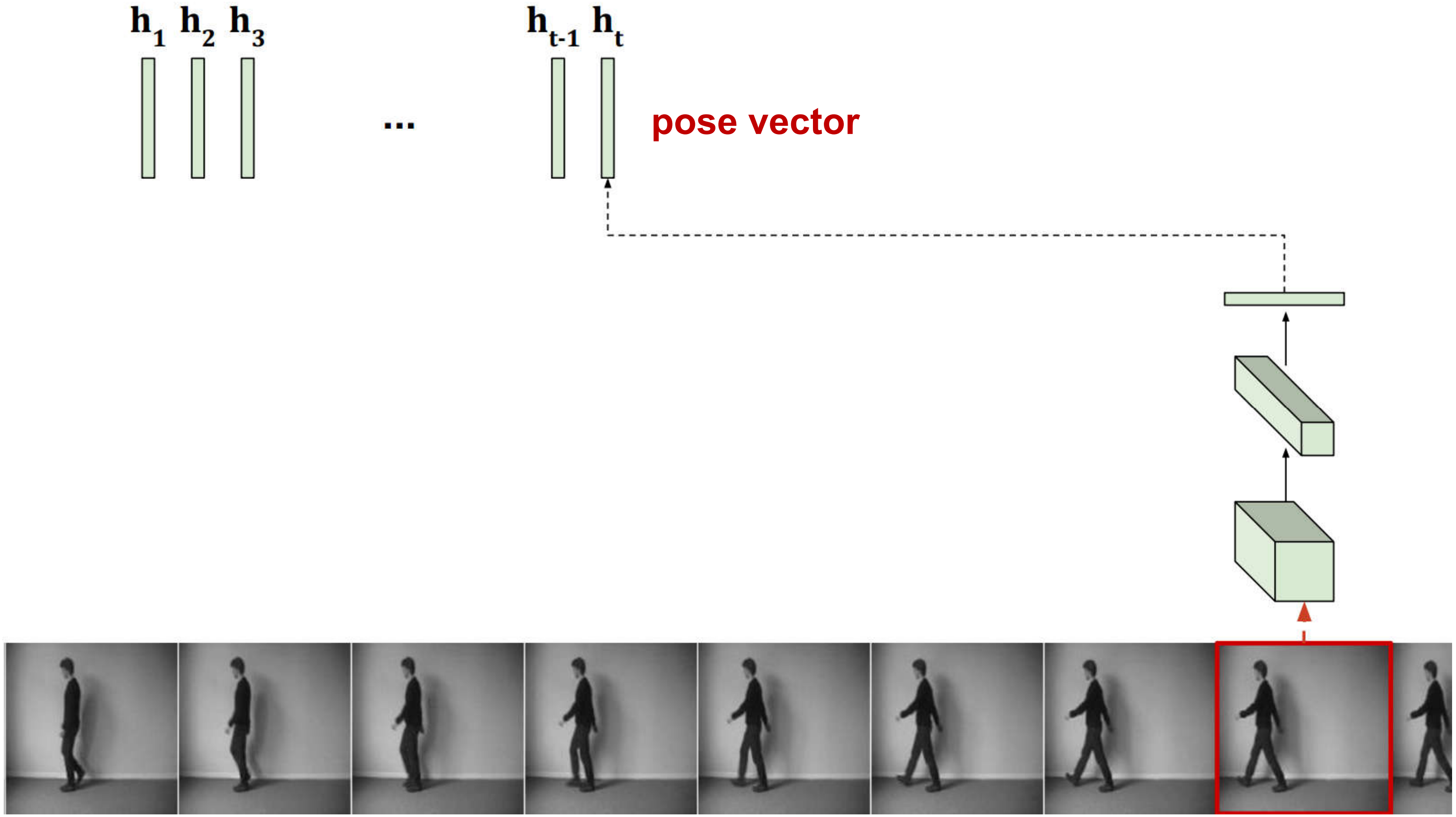
- Interpolation in pose space

# Image Synthesis by Analogy [4/4]



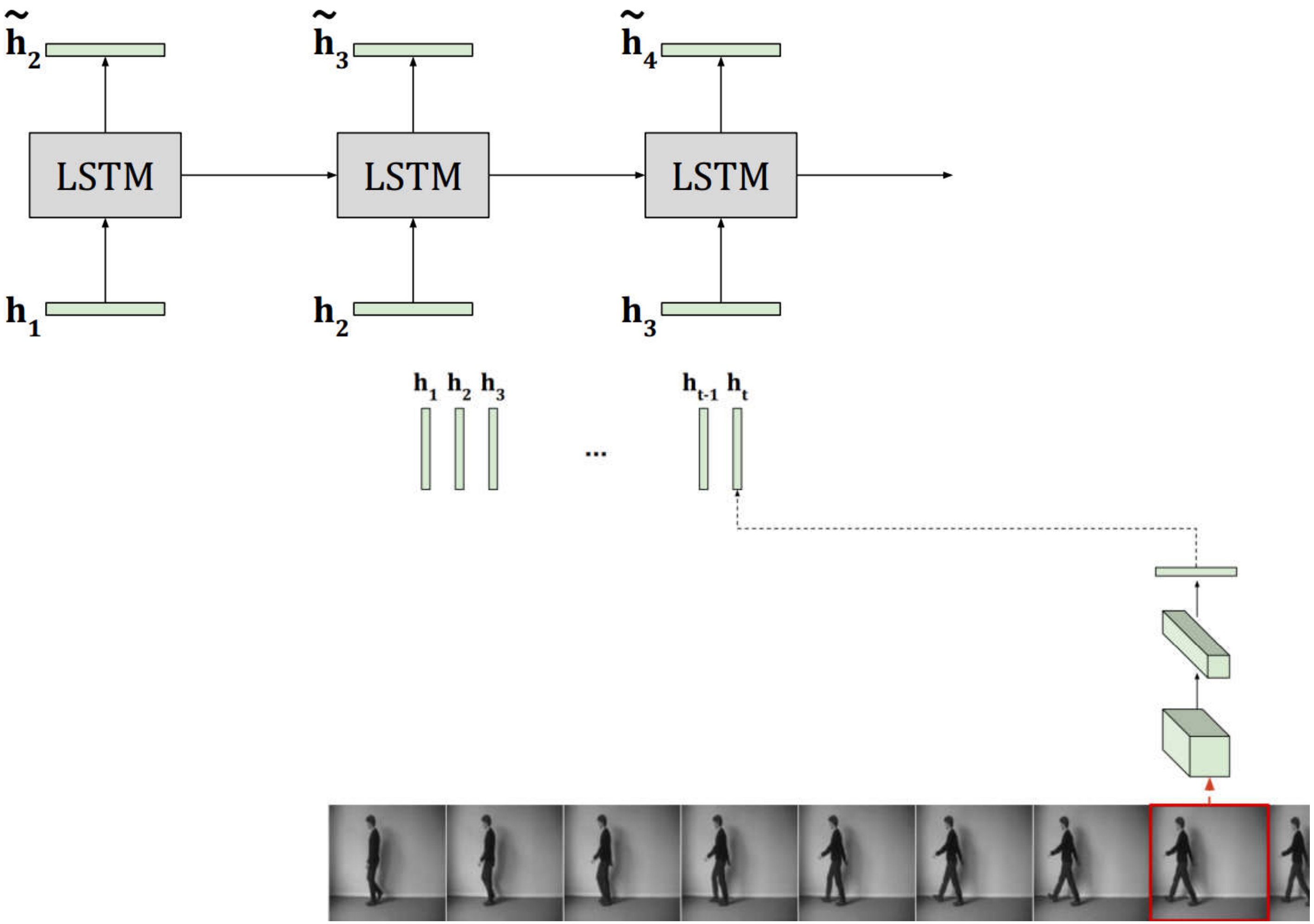
# Video Prediction [1/2]

- Instead modeling how the entire scene changes, **only need to predict the temporally varying component**
- **Prediction** done entirely in latent **pose space**



# Video Prediction [2/2]

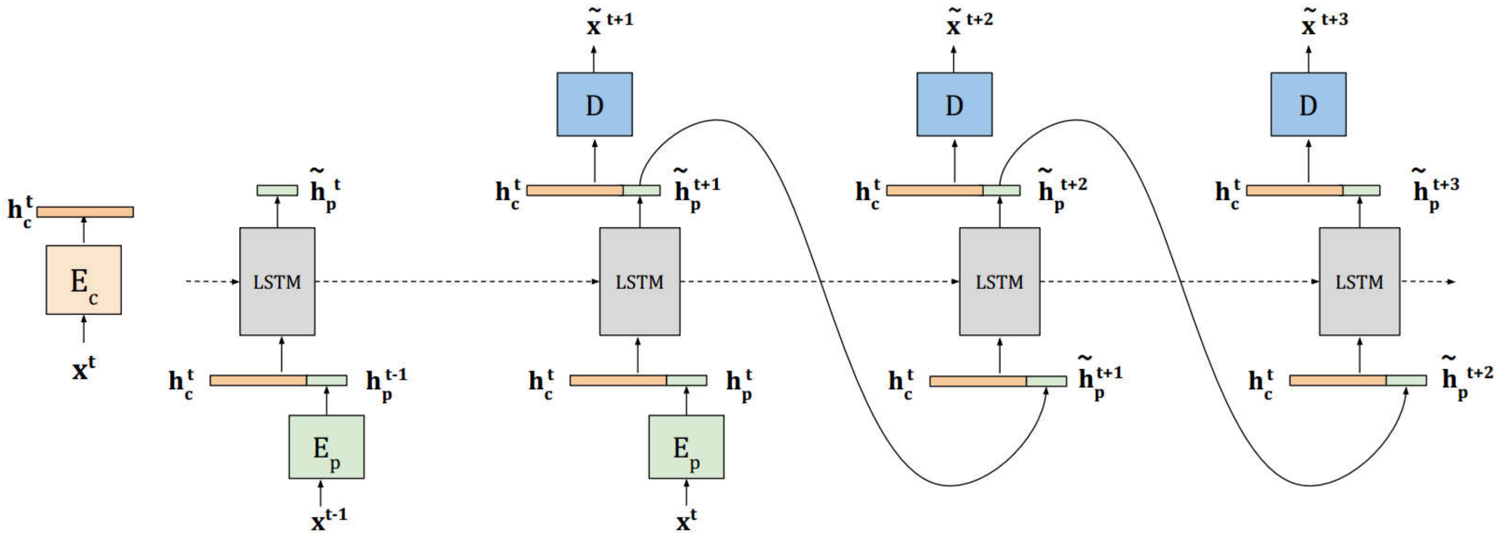
- Train LSTM to **predict future pose vectors**




- Don’t have to worry about content vectors they are fixed across time by design

# Test Time: Generating A Video Sequence

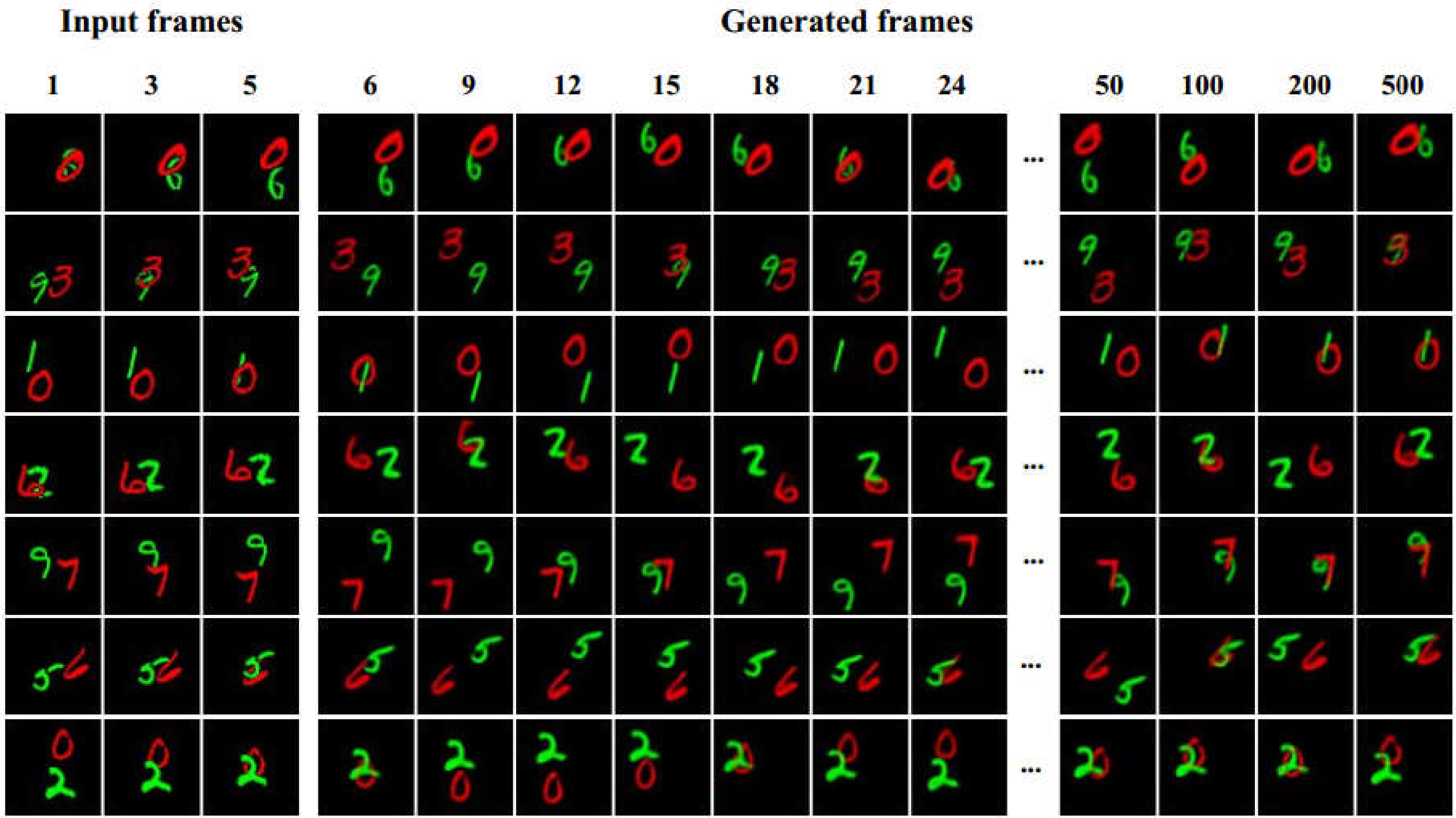
- Feed predicted pose vectors back into model



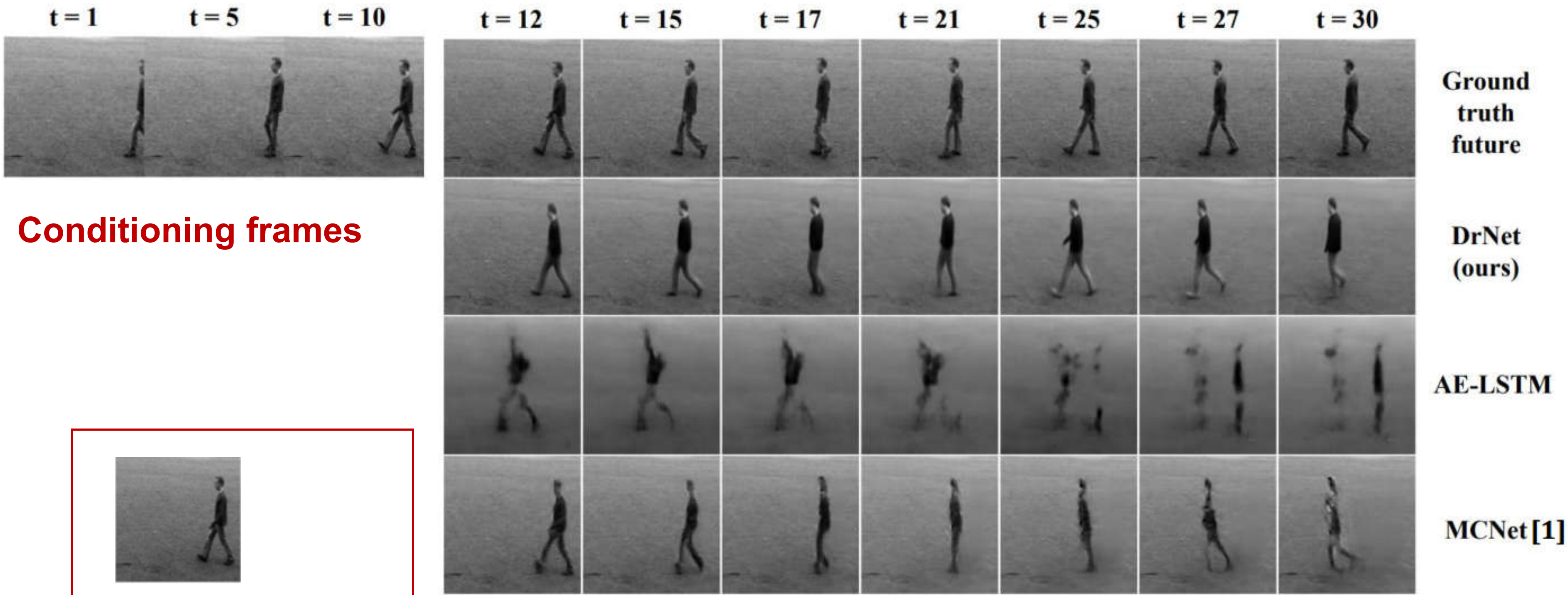

$$
\tilde{h}_p^{t+1} = LSTM\left(E_p\left(x^t\right), h_c^t\right) \qquad \tilde{x}^{t+1} = D\left(\tilde{h}_p^{t+1}, h_c^t\right)
$$

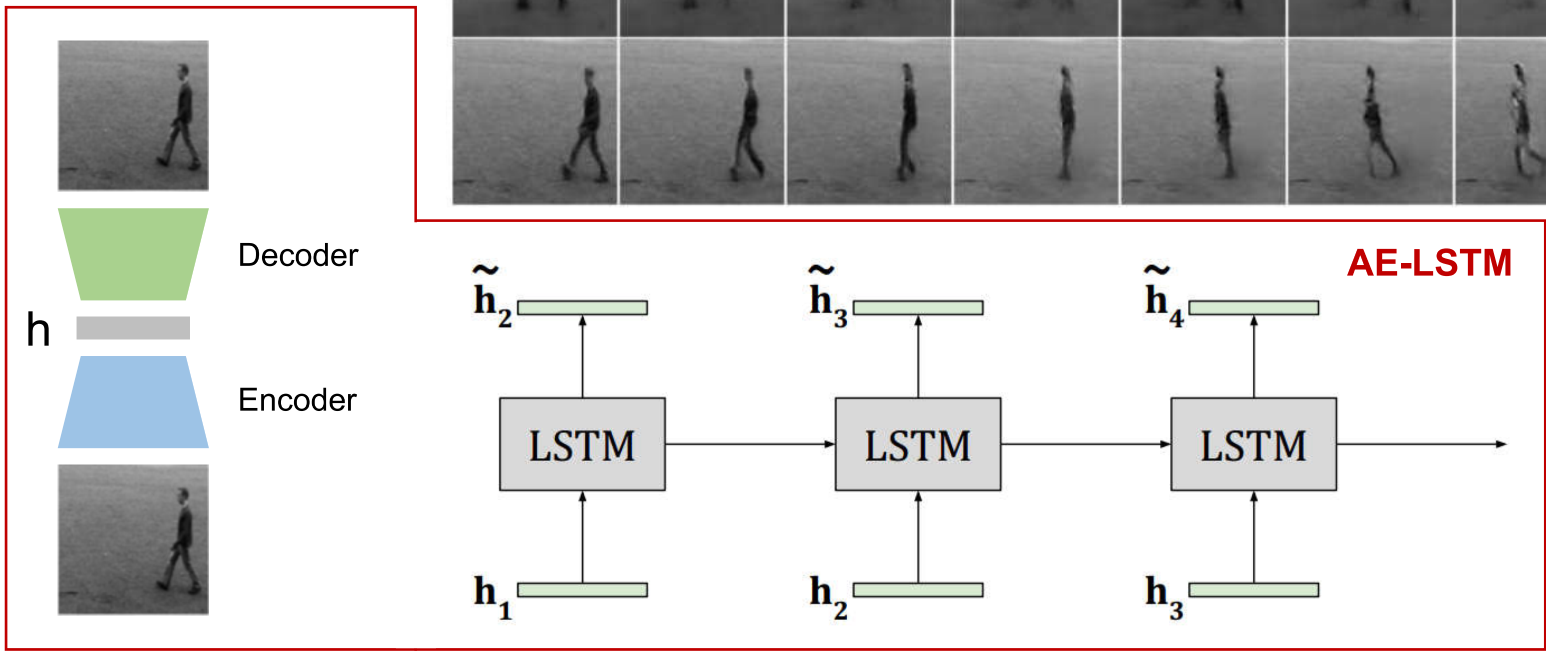
$$
\tilde{h}_p^{t+2} = LSTM\left(\tilde{h}_p^{t+1}, h_c^t\right) \qquad \tilde{x}^{t+2} = D\left(\tilde{h}_p^{t+2}, h_c^t\right) \tag{6}
$$

# Moving MNIST: Generating Forever…



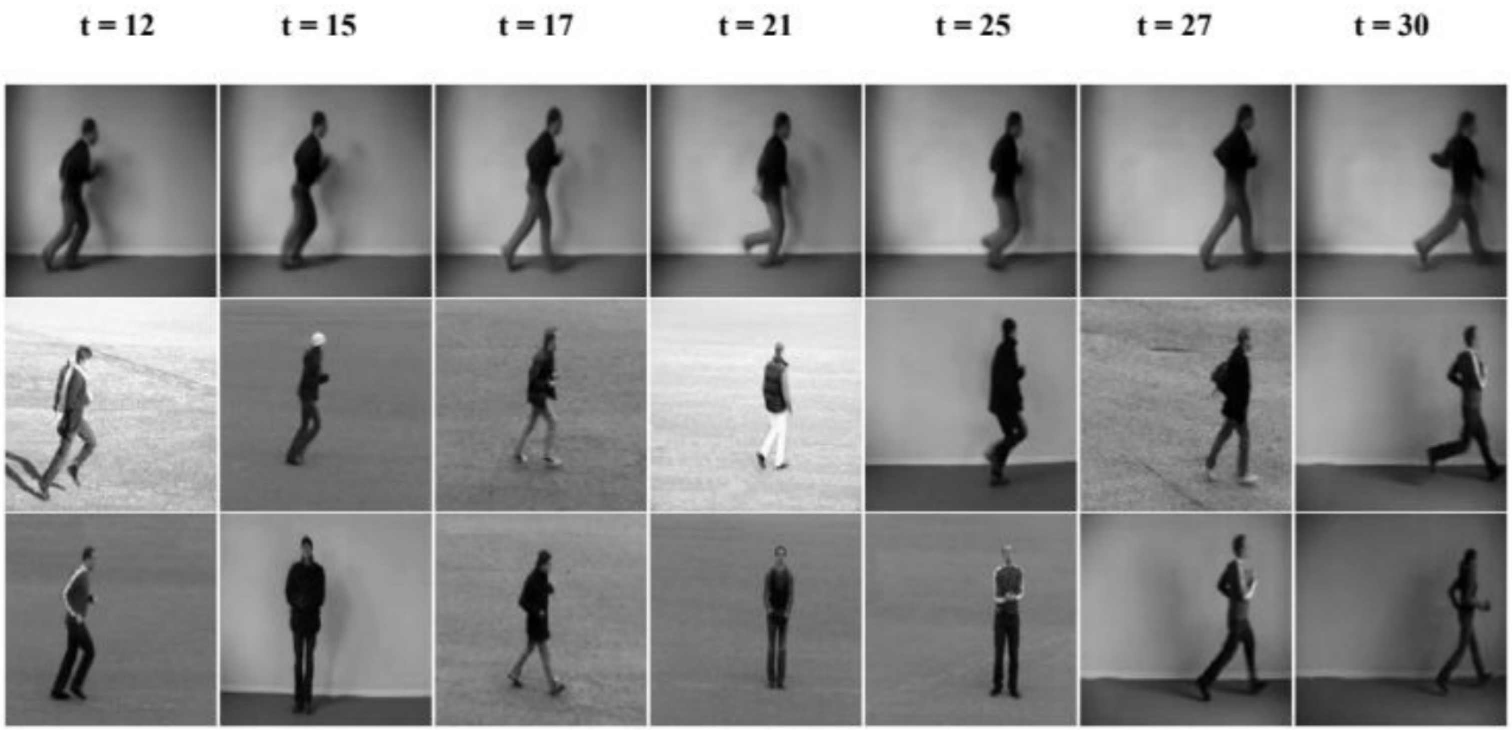
# KTH Video Generation




[1] Villegas et al., Decomposing motion and content for natural video sequence prediction, ICLR2017

# KTH Nearest Neighbors

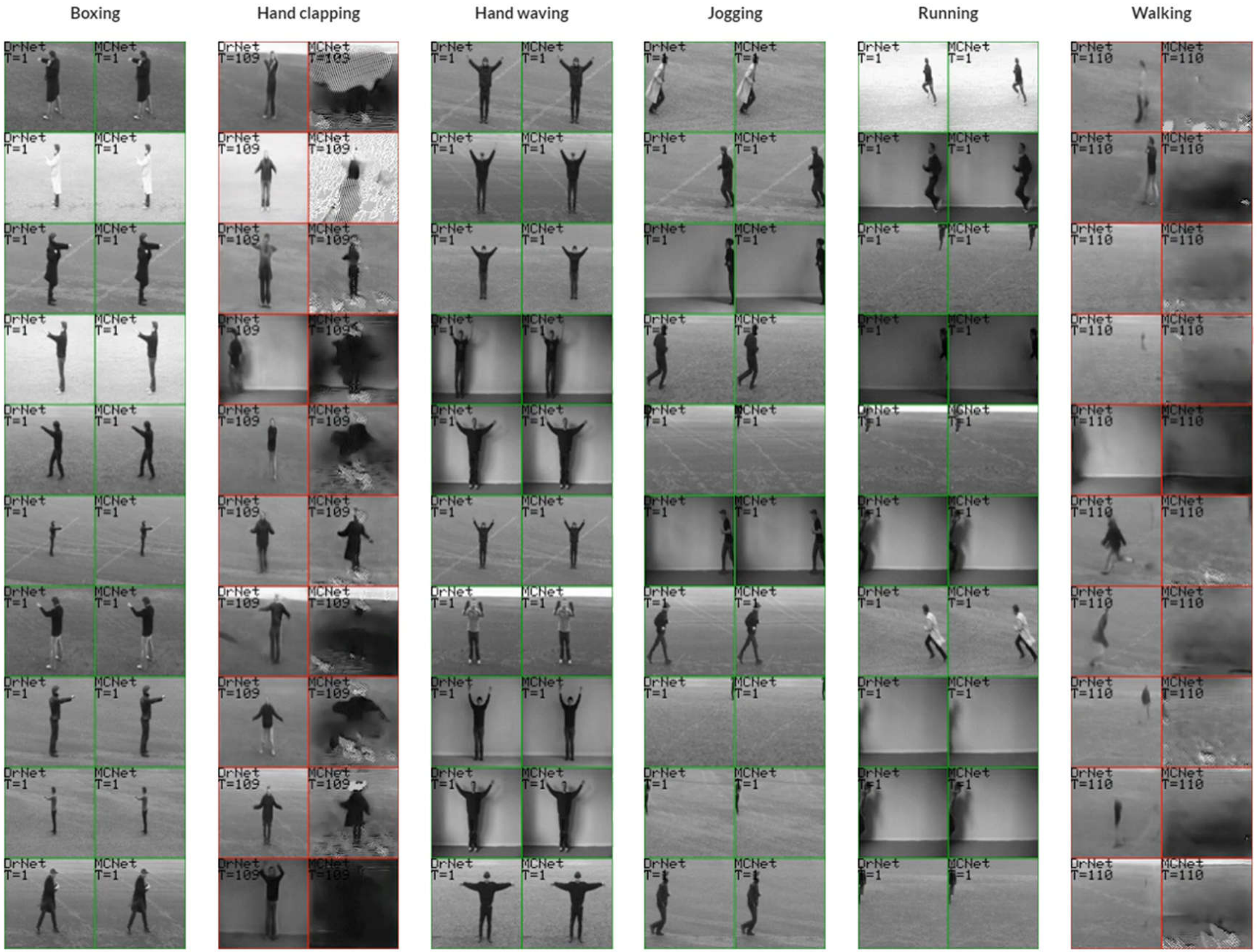



DrNet generations

Nearest neighbor in pose space

Nearest neighbor in pose + content space

# Further Examples




https://sites.google.com/view/drnet-paper//

# Thank you for your attention!