





Detection and Tracking of Humans and Faces using Machine Learning

Dimitris Metaxas

dnm@cs.rutgers.edu

Center for Computational Biomedicine, Imaging and Modeling

Rutgers University

July 20, 2020

Center for Computational Biomedicine, Imaging and Modeling (CBIM)

Established in 2001

Members

13 Faculty Members

65 PhD students

5 Postdocs

3 Visitors

Collaborations

UPENN, Boston Univ.
Columbia NYU Medical School
MIT Stanford
Brookhaven National Laboratory
Siemens Healthcare, GE
Adobe Systems
SenseTime

Research Areas

Computational Biomedicine, Computer Vision, Computer Graphics, Scientific Computations, Learning and Robotics, CS, EE, Biomedical Engineering, Cell Biology, Medical Schools, Neuroscience Linguistics

Interpretable Learning-Based Models for Visual Recognition Tasks


Understanding Facial Behavior

Thrust 2 Goal

Thrust 1: Cross-Cultural Attitudinal Observatory	• Design of a modified version of the MAFIA game so as to elicit dominance-deference, like-dislike, and deceptive behavior which will be video recorded
Thrust 2: Audio,Video,Verbal,Nonverbal Signal Extraction	• Design of methods to automatically extract *verbal* (pitch, amplitude, pauses, dis-fluencies)) signals from the videos, *non-verbal* signals (facial expressions, gesture, head/eye movement), *linguistic* (sentiment, emotion)
Thrust 3: Culture-Dependent SCAN Construction	• Design of methods to use the extracted signals in Thrust 2 to learn predictive models of who likes /dislikes who, and who dominates/defers to who
Thrust 4: Culture-Dependent SCAN Construction Deception Detection	• Design of methods that use audio-visual signals, deception centrality, and deception transition graphs showin g the dynamics of deceptive/honest actors over time to classify actors as deceptive or honest.

We were focusing on analysis of the visual cues from faces (non-verbal)

Activities (3 parts corresponding to the presentation)

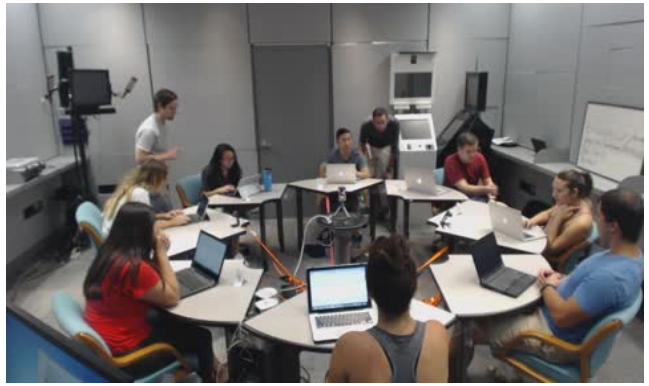
Part1: Combine our face tracking, head gesture detection and expression recognition modules, a fully automatic visual cue extraction system is introduced to process the data collected by UA and UCSB teams.

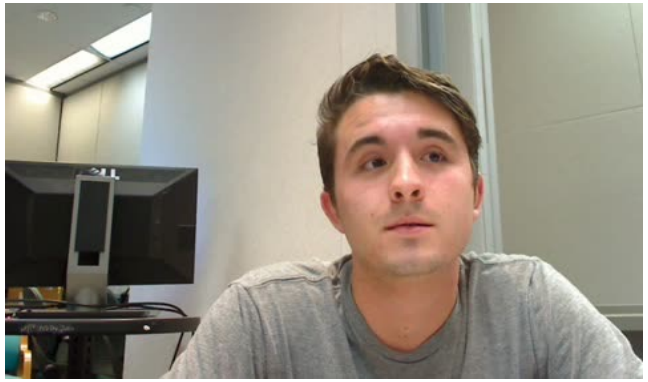
Part2: The spy detection is formulated as a classification problem and the 3D Convolutional Neural (C3D) Net is used to model Spies and Villagers' facial movement

Part3: Instead of using Neural Net as a black-box, propose the attention mechanism to discover the dynamic cues of facial movement that the model attends to, which is discriminative for spy detection

Part1: Analysis of players' faces and Geometric Setting

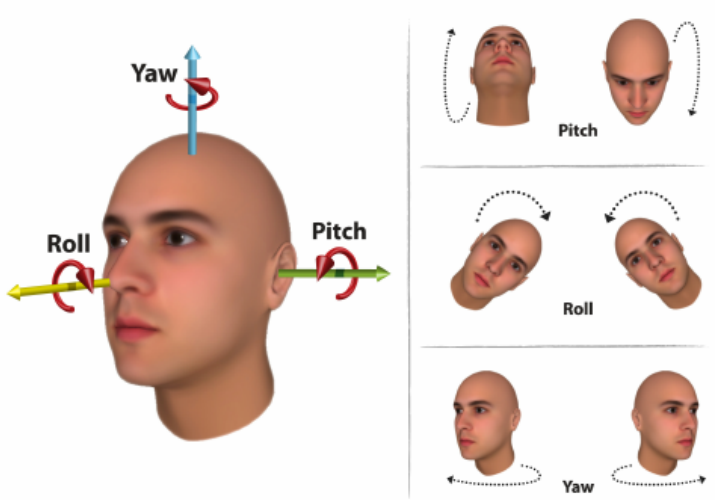
- Videos are captured in three views: Overhead, 360-view and standard-view
  - Overhead and 360-view depict the scene topology of all players
  - Standard-view shows the details of each player's face
  - Our current algorithm mainly deal with the videos in the standard view
  - Combination of visual cues from two views can predict 'who is talking to who?'





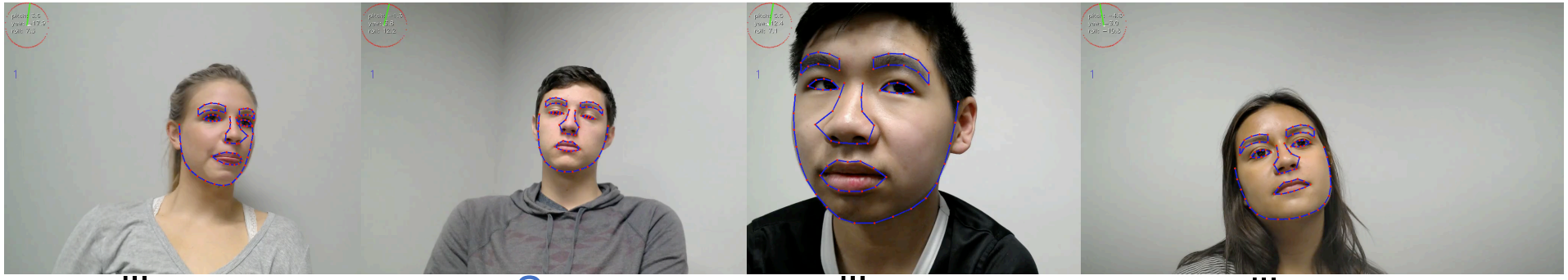
Part1: Analysis of players' faces and Geometric Setting

- The visual cues over time are extracted from videos in standard view
  - The head pose is measured in three angles: pitch, roll and yaw
  - 68 key-points are tracked to measure the players' facial movement
  - Player's expression in every frame are mapped into three categories: positive, neutral and negative



Part1: Analysis of players' faces and Geometric Setting

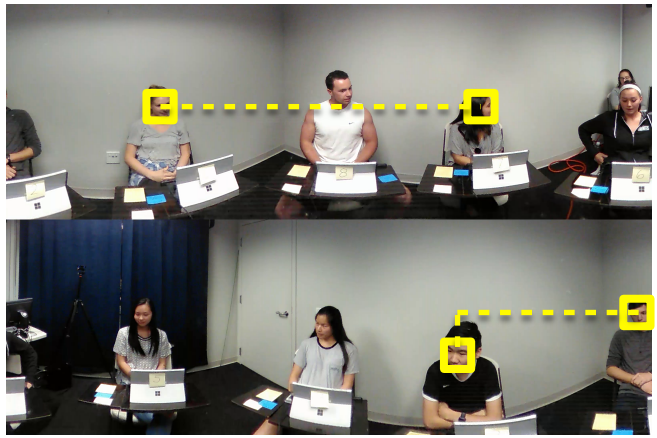
- Three angles for head pose are visualized on the left top corner; numbers (blue) indicates the frame ID, and 68 facial key points are plotted
- The player's faces and facial keypoints can be detected automatically at the same time
  - We proposed a coupled-encoder and decoder network to achieve the tow tasks jointly [*published in journal Image&Video Computing 2018*]

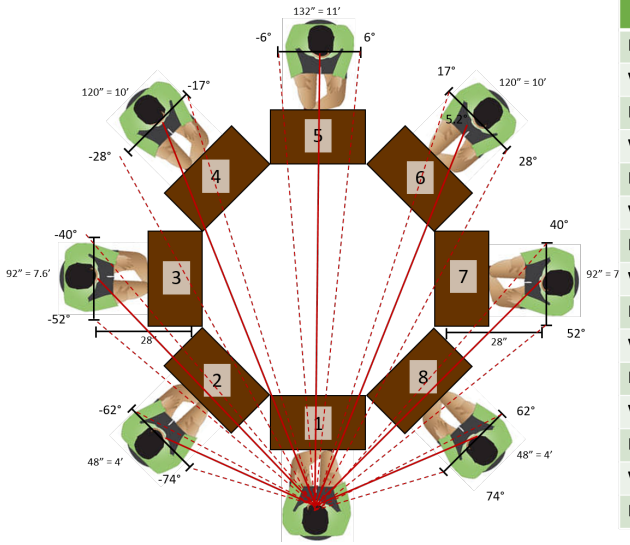


villager Spy villager villager

Part1: Analysis of players' faces and Geometric Setting

- The players are setting around a circle, so we analyze the geometry setting of the group via investigation of the distance between players in the video recorded by 360 camera



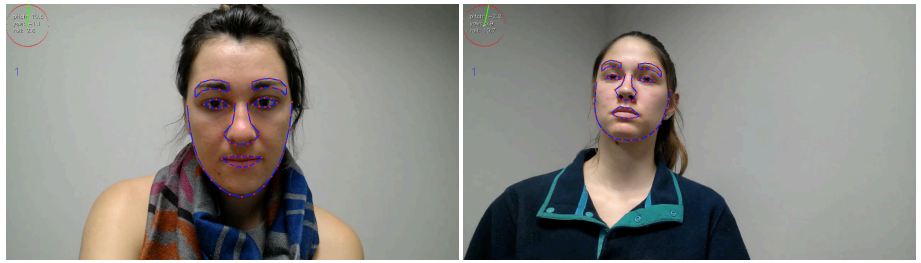


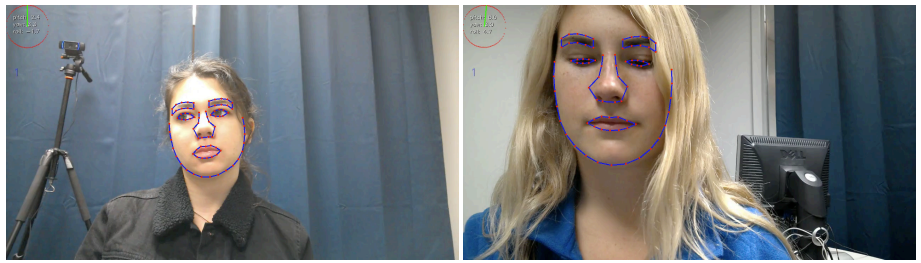
Extent	1st Angle	2nd Angle	Angle Width
Left Gap	-90	-73.57	16.42
View P2	**-73.57**	**-63.03**	**10.53**
P2-P3 Gap	-63.03	-51.26	11.76
View P3	**-51.26**	**-40.87**	**10.39**
P3-P4 Gap	-40.87	-28.00	12.86
View P4	**-28.00**	**-17.86**	**10.14**
P4-P5 Gap	-17.86	-5.71	12.15
View P5	**-5.71**	**5.71**	**11.42**
P5-P6 Gap	5.71	17.86	12.15
View P6	**17.86**	**28.00**	**10.14**
P6-P7 Gap	28.00	40.87	12.86
View P7	**40.87**	**51.26**	**10.39**
P7-P8 Gap	51.26	63.03	11.76
View P8	**63.03**	**73.57**	**10.53**
Right Gap	73.57	90	16.42

- Picture from Bradley Walls

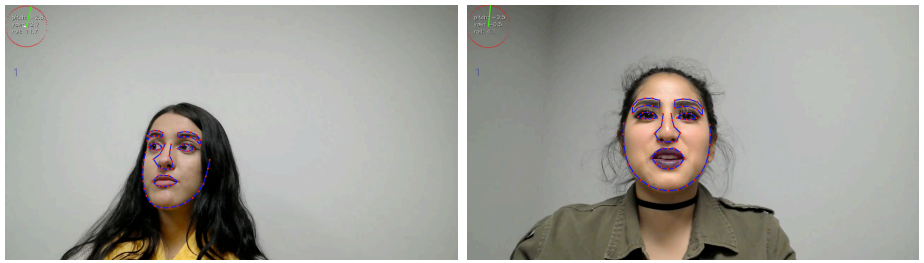
Part2: Understanding who is spy or villager

SPY





Villager

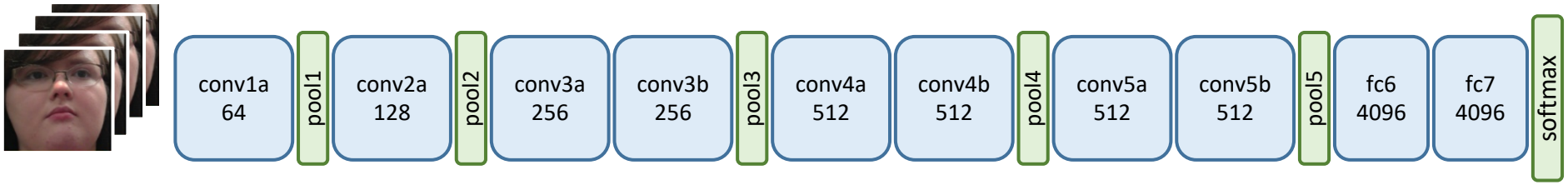


Part2 & Part3: Understanding who is spy or villager

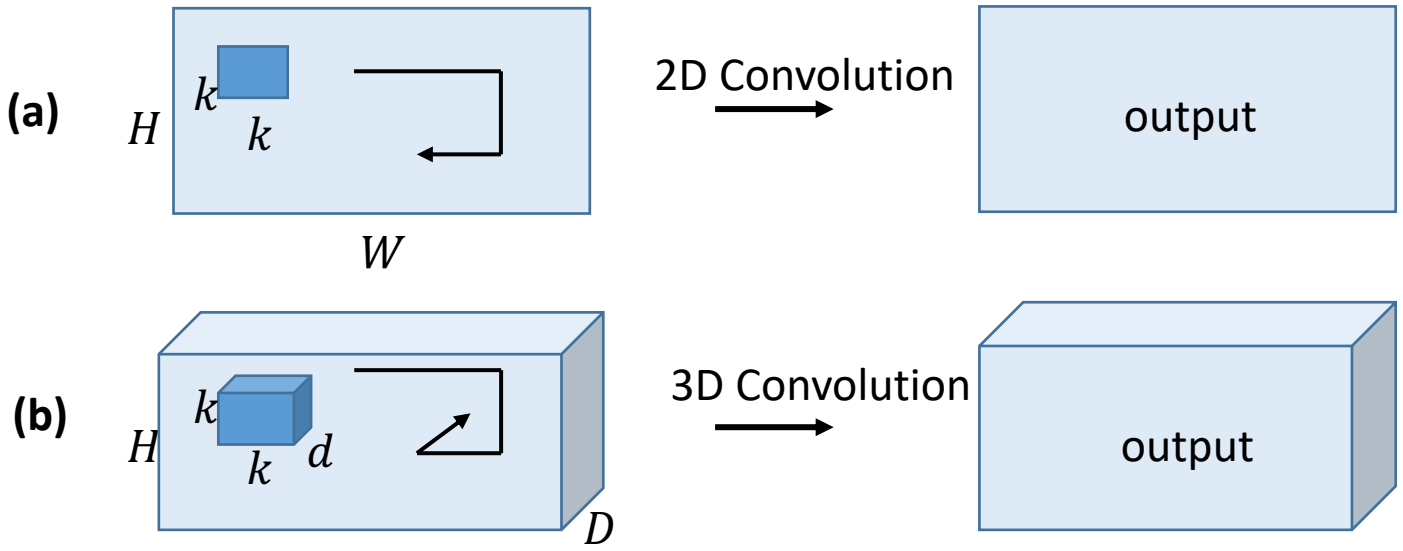
- We have the video-level label (0:villager, 1:spy) of the player's role over the whole video
- The valuable data hides in the larger amount of noise
  - Most of the time, spy and villager have the same behavior
  - Labelling the facial difference between spy and villager is difficult
  - We are trying to answer the following three questions
    - Q1: How to perform the video-level spy/villager classification
    - Q2: Where there is spying or villager behavior in the video (Understandability)
    - Q3: How do we characterize this behavior in terms of features

Question 1: Spy vs Villager Classification

- We apply a 3D convolutional neural (C3D) network to classify a player as the Spy or Villager
- The C3D takes a video clip of a player's facial behavior as input directly and output the spy probability
- Given a video file in training stage, we apply random sampling to get the video clips instead of applying the C3D on the whole video



Question 1: 3D Convolution for Video

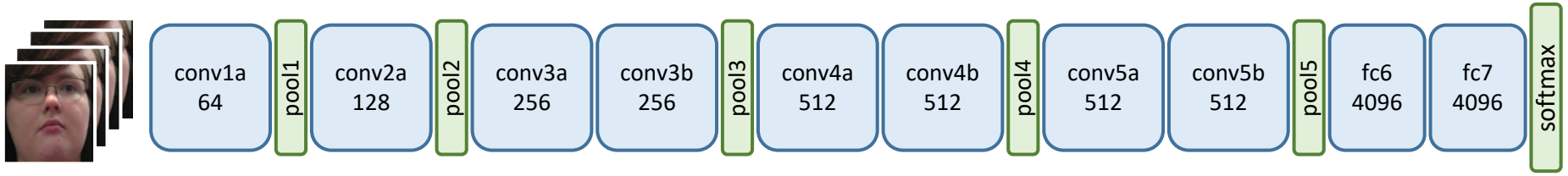


2D and 3D convolution operations. (a) Applying 2D convolution on an image results in an image. (b) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal.

Question 1: 3D Convolution (C3D) for Video

In our C3D architecture, there are:

- 8 convolution, 5 max-pooling, and 2 fully connected layers followed by a softmax layer.
- 3D convolution kernels are 3 × 3 × 3 with stride 1 in both spatial and temporal dimensions.
- Number of filters are denoted in each box.
- The 3D pooling layers are denoted from pool1 to pool5. All pooling kernels are 2 × 2 × 2, except for pool1 is 1 × 2 × 2.
- Each fully connected layer has 4096 output units.
- The output has 2 dimensions for binary classification



Train the C3D model

- Video data comes from the Round3 of homogeneous Games:
  - 006AZ/ 009SB/ 011SB/ 012AZ/009ISR/011NTU
  - There are 43 players' videos in total, including 18 spies and 25 villagers
  - Total length is ~20700 seconds (~345min)
- We apply our face tracker to each player's video, cropping the facial regions as the input of C3D net
  - The faces are cropped by taking the bounding boxes [xmin-10: xmax+10, ymin-30:ymin+20] where the xmin, xmax, ymin, ymaxs stands for the minimum and maximum values of landmarks in horizontal or vertical axis

Train the C3D model

- We random select videos from one/two game as validation data and the rest as the training
  - There is no duplicate players appearing in both training and validation set
- The round3 videos are segmented according to the fine-grained timestamps into five subsections of:
  - Start(ding)/ leader_discusstion/ leader _reveal/ team_discusstion/ team_reveal/ mission_reaction
  - There are 43 (no. of players) *5 = 215 video files used for train the C3D model
  - During training, given a video file, we random sample 16 frames as the input of C3D and each frame is re-sized to 112x112
  - The temporal order is kept among the selected frames

Train the C3D model

- The performance of the baseline C3D + randomly frame selections

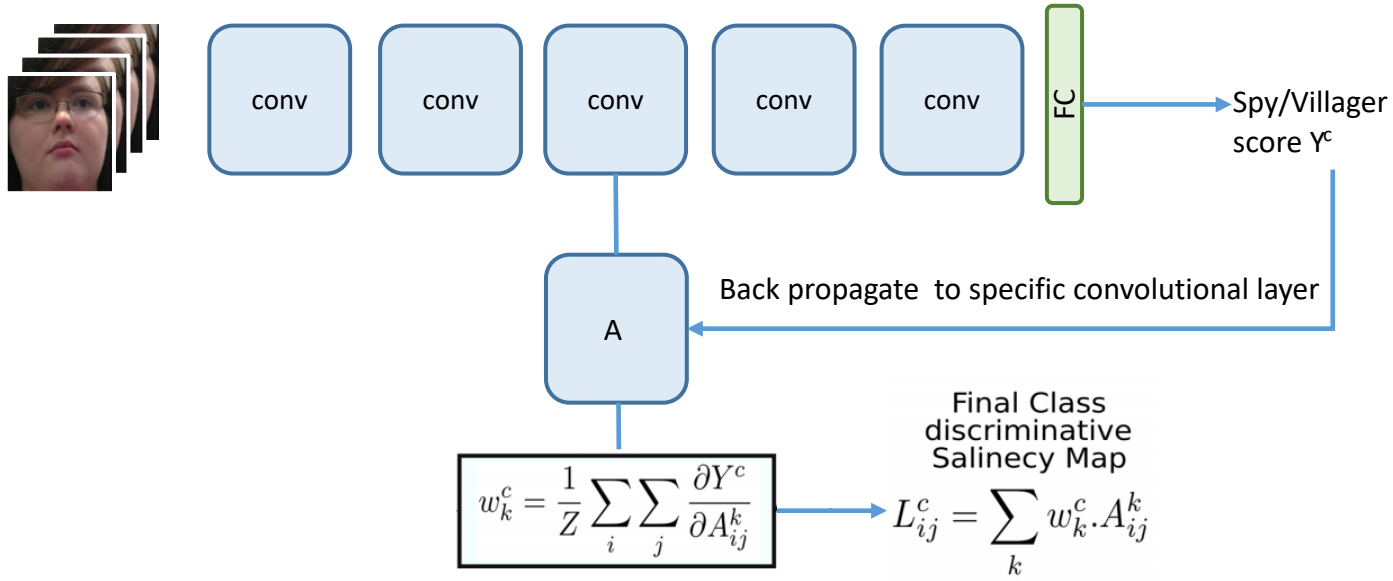
#Validation/Training Games	Classification Accuracy
1/5	65.71
2/4	62.37

- Ways to improve the performance:
  - **Data augmentation: generate faces in different views** [IJCAI18, ECCV18]
  - Apply **attention model** to understand the discriminative patterns between spy and villager
    - Training with the attention selected data vs. random selection
    - Humans are 70% and we have not trained yet on a lot of data
    - Different players for train and different for testing

Question 2: Model Attention discovers where the Spies are in the video?

- We do not want to apply the deep neural net as a black box
- We are trying to discover the spatial and temporal information which is essential for network to make the prediction
- We compute the class-oriented attention maps for the input image sequences, visualizing the important pixels and frames:
  - Given a video clip and C3D output, we compute the gradient wrt to the feature maps
  - The gradients indicates the pixel importance to the final prediction

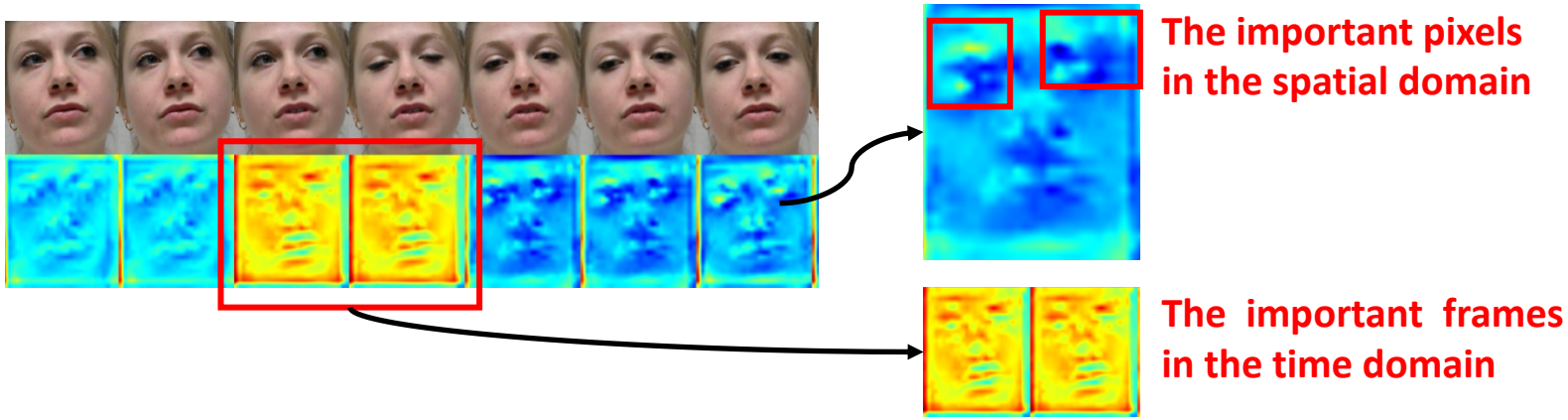
Question 2: Class-oriented Attention Map



- We use backpropagation to compute the gradient of score Y^c wrt the specific convolution features A
- The gradient $\frac{\partial Y^c}{\partial A_{ij}^k}$ indicates the spatial and temporary importance and is used to compute the weights w
- We use the w to combine the features, generating the attention map L

Question 2: What does our model learn?

- We observe that C3D starts by focusing on appearance in the first few frames and tracks the salient motion in the subsequent frames.
- In the following example, it first focuses on the eyes, mouth and then tracks the motion (variance) happening around them.
- Thus C3D differs from standard 2D ConvNets in that it selectively attends to both motion and appearance.
- Attention technique highlights the spatial and temporary information which has the positive contribution for the final prediction



Question3: Characterize model attending behaviors

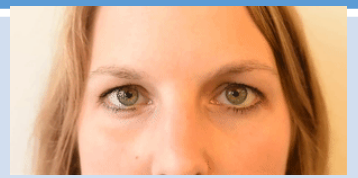
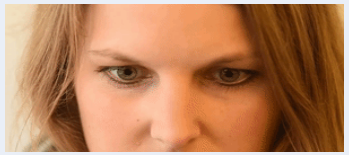
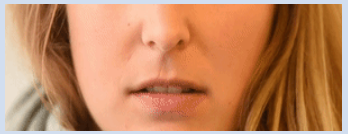
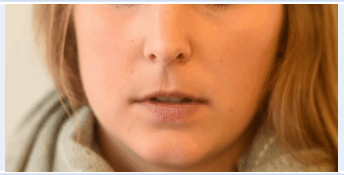
- In testing stage, we break the video into different clips
- The short clips are forwarded into the C3D model
- Clips with high spy probability can be reserved for deep investigation
- We compare what the model attends to with what the latest deception research would predict regarding face and head
  - The facial cues are coded as facial action units (AU)
- We assume that spies are more deceptive than villagers

Question3: Characterize model attending behaviors

- In the latest deception theory, deception is represented by the combination of facial Action Unit(AU), including:
  - More blinks (AU45) with emotional responding and masking, fewer blinks with cognitively loaded responses and efforts at neutralization
  - Sneer (AU9 + AU10) while feigning sadness
  - Lip adaptors (AU18, AU19, AU23, AU24)
  - etc.
- Sources for the above come from various articles and include:
  - DePaulo (2003) (but this is seriously outdated)
  - Cohn, Zlochower, Lien & Kanade (1999)
  - Porter & ten Brinke (2008)
  - Waller, Cray, & Burrows (2008)
  - Kessous Castellano & Caridakis (2009)
  - Matsumoto, Willingham & Olide (2009)
  - Hurley & Frrank (2011)
  - ten Brinke & Porter (2012)
  - ten Brinke, Porter & Baker (2012)
  - Matsumoto & Hwang (2017)

Question3: Deception Cues vs. Model attention

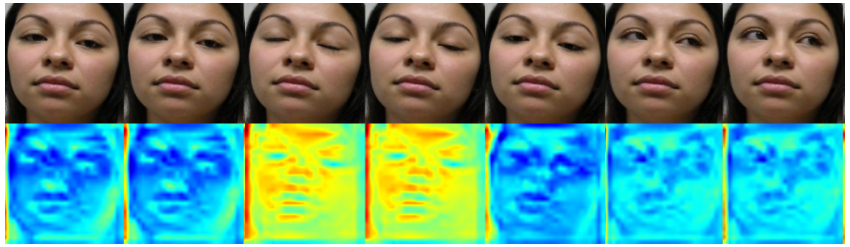
- Samples of Action Units are considered as deception:

Action Unit	Description	Facial Muscle	Example (Hover to Play)
AU45	Blink	Relaxation of *Levator Palpebrae and Contraction of Orbicularis Oculi, Pars Palpebralis.*	
Sneer **AU9** + AU10	Nose Wrinkler	*Levator labii superioris alaquae nasi*	
Sneer AU9 + **AU10**	Upper Lip Raiser	*Levator Labii Superioris, Caput infraorbitalis*	
Lip adaptors (**AU24**)	*Lip Pressor*	*Orbicularis oris*	
Faked happiness (**AU12**, but missing AU6)	Lip Corner Puller	*Zygomatic Major*	

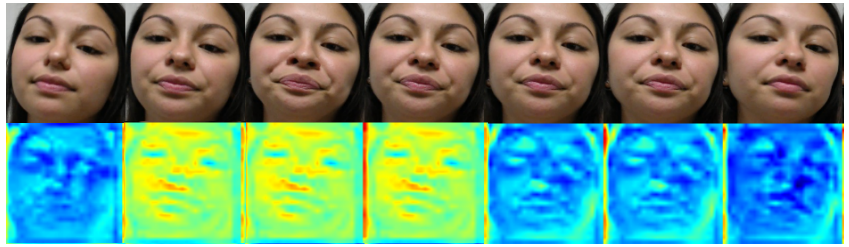
Question3: Deception Cues vs. Model attention

- Looks like the network is finding what seems to be known about deception: Here are some AUs addressed by the model attention
- But also their dynamics

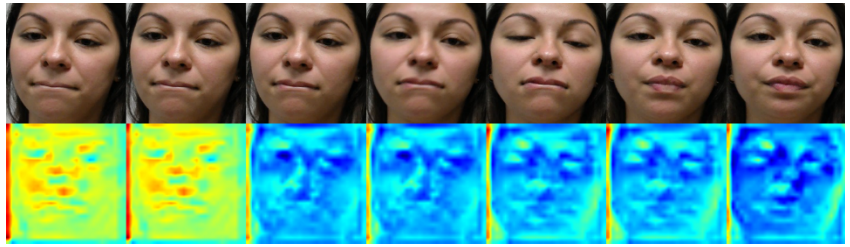
AU45: blinks



AU20: Lip stretcher
AU13: Cheek Puffer

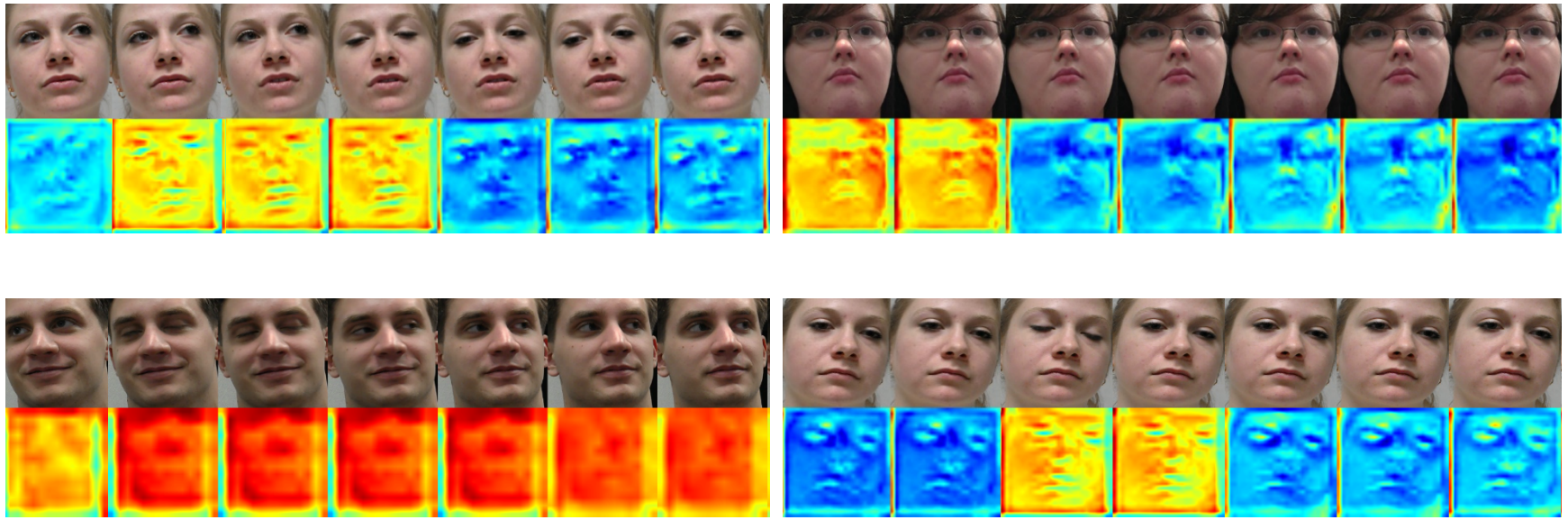


AU24: Lip Pressor



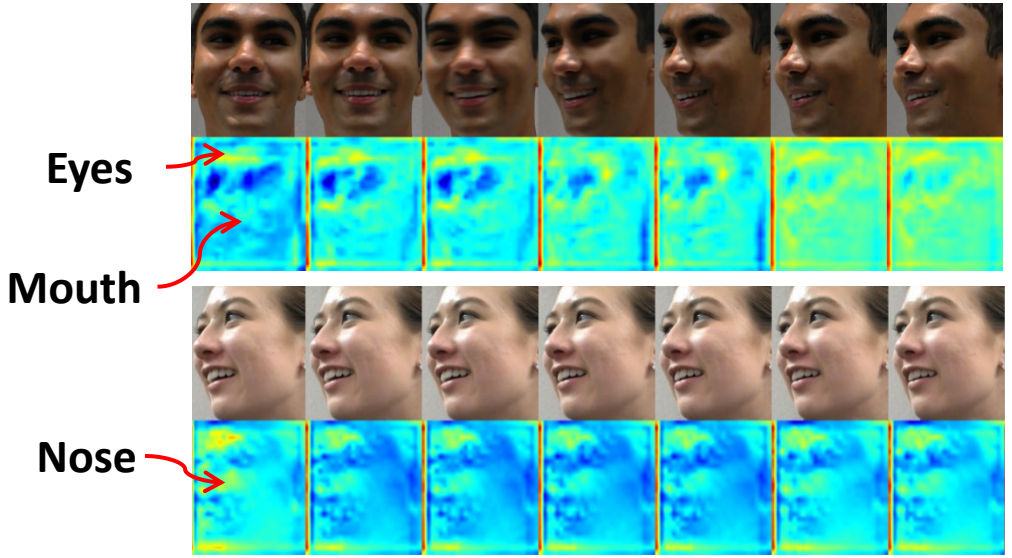
Question3: Deception Cues vs. Model attention

These are good results and we see: **eyes closed**, **fake smile**, **changes in lips**.
Here are some which fall into the Spy category but are more subtle.



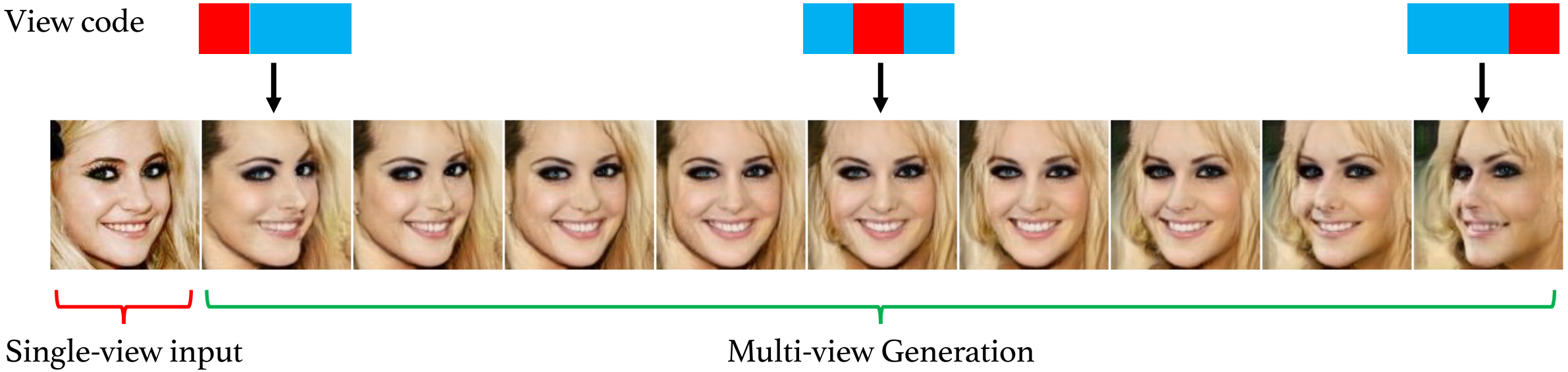
Question3: Deception Cues vs. Model attention

- Villager frames with the attention maps
- In the spatial domain, the C3D model attends to the facial parts such as **eyes, nose** and **mouth**
- In the temporal domain, there is no sharp intensity changing comparing to the attention maps of spies



"Creativity"
Learning How to Learn: Multi-view Generation

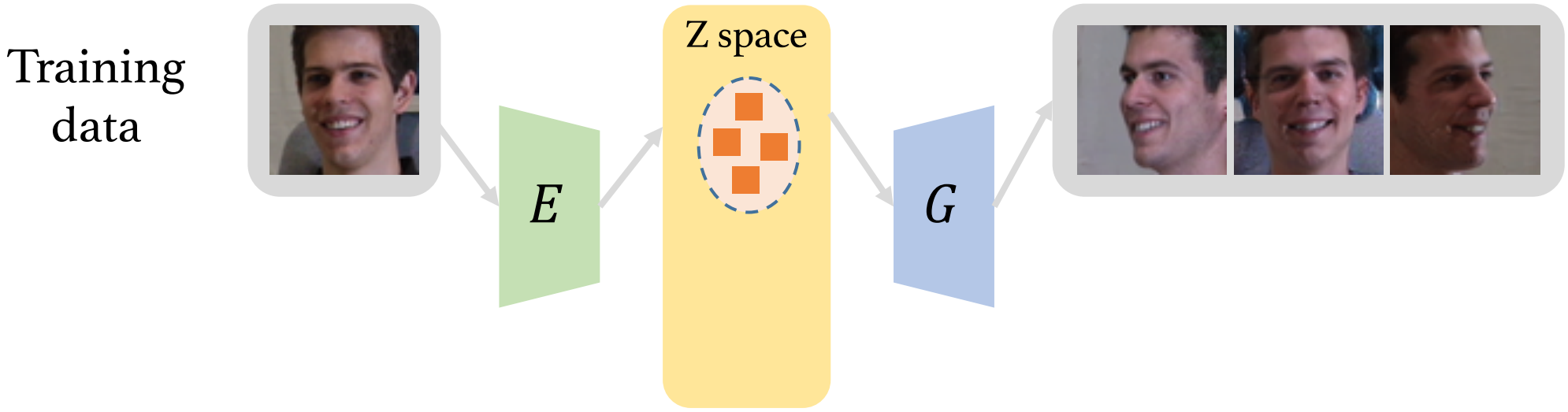
- Generating multi-view face images from a single-view input.
- Create images for downstream tasks (e.g. Face recognition)
- Disentangled representations.
- Extend to other applications.



Y. Tian et al. "CR-GAN: Learning Complete Representations for Multi-view Generation". IJCAI'18

Prior works: Approach

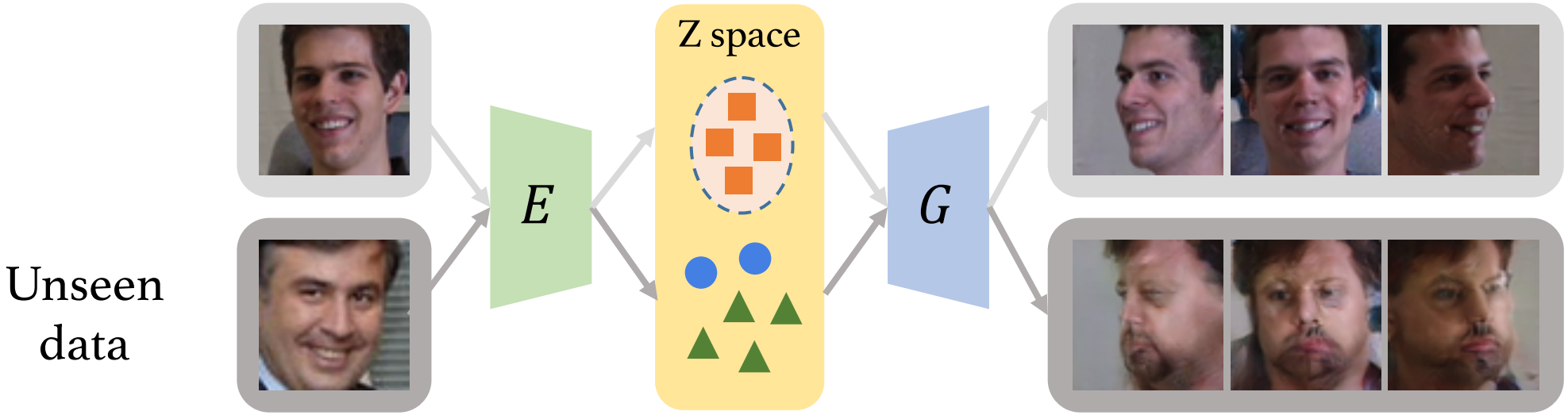
- Inspired by GAN: Encoder-Generator-Discriminator network.
- Encoder maps training data to representation space Z.
- Generator trained on mapped representations.



Y. Tian et al. “CR-GAN: Learning Complete Representations for Multi-view Generation”. IJCAI’18

Prior works: Limitations

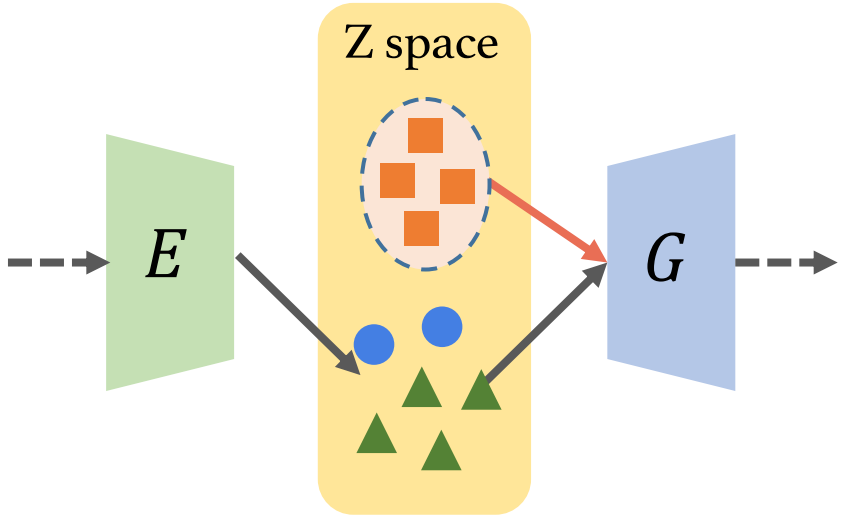
- Encoder maps finite training data to subspace.
- Generator only trained on this subspace.
- "Unseen" data may map out of the space → Undefined behaviors.



Y. Tian et al. "CR-GAN: Learning Complete Representations for Multi-view Generation". IJCAI'18

Proposed method: Complete Representation (CR)-GAN

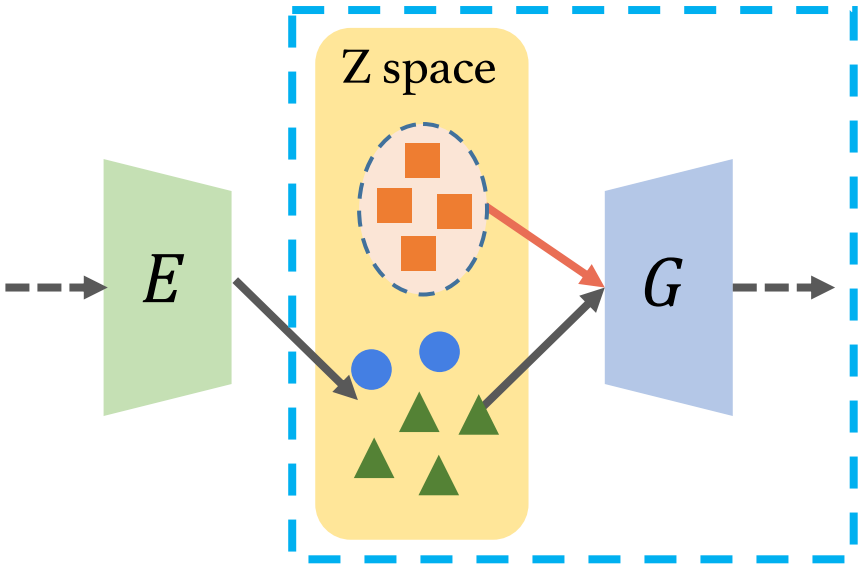
- Two pathway framework.
- Work with complete representations.



Y. Tian et al. “CR-GAN: Learning Complete Representations for Multi-view Generation”. IJCAI’18

CR-GAN: Generation Path

- Conditional generation: WGAN-gp + ACGAN.
- Generator is trained in complete space.



D maximizes:

$$\mathop{\mathbb{E}}_{\mathbf{z}\sim\mathbb{P}_\mathbf{z}}[D_s(G(v,\mathbf{z}))] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_\mathbf{x}}[D_s(\mathbf{x})] +$$

$$\lambda_1 \mathop{\mathbb{E}}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2] - \lambda_2 \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_\mathbf{x}}[P(D_v(\mathbf{x}) = v)]$$

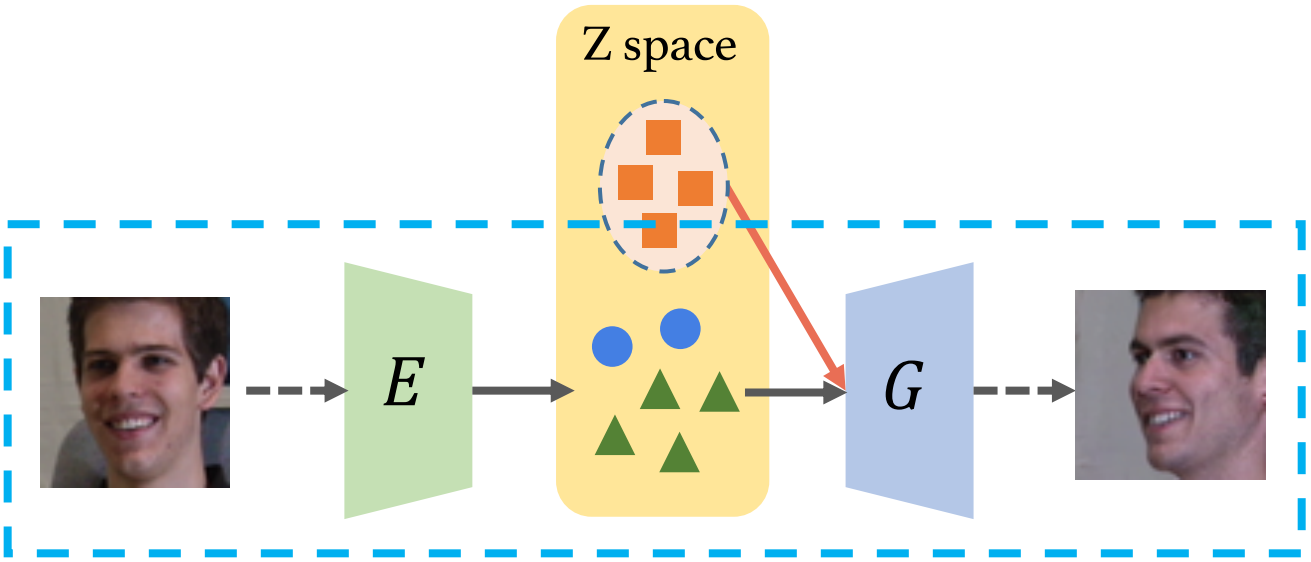
G minimizes:

$$\mathop{\mathbb{E}}_{\mathbf{z}\sim\mathbb{P}_\mathbf{z}}[D_s(G(v,\mathbf{z}))] + \lambda_3 \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathbb{P}_\mathbf{z}}[P(D_v(G(v,\mathbf{z})) = v)]$$

Y. Tian et al. “CR-GAN: Learning Complete Representations for Multi-view Generation”. IJCAI’18

CR-GAN: Reconstruction Path

- Encoder to reconstruct all training data in different view.
- Dataset (All views of same identity): Multi-PIE, 300WLP.
- Li-loss to enforce identity.



D maximizes:

$$\mathbb{E}_{\mathbf{x}_i,\mathbf{x}_j\sim\mathbb{P}_\mathbf{x}}[D_s(\tilde{\mathbf{x}}_j) - D_s(\mathbf{x}_i)] +$$
$$\lambda_1 \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}[(\|\bigtriangledown_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2] - \lambda_2 \mathbb{E}_{\mathbf{x}_i\sim\mathbb{P}_x}[P(D_v(\mathbf{x}_i) = v_i)],$$

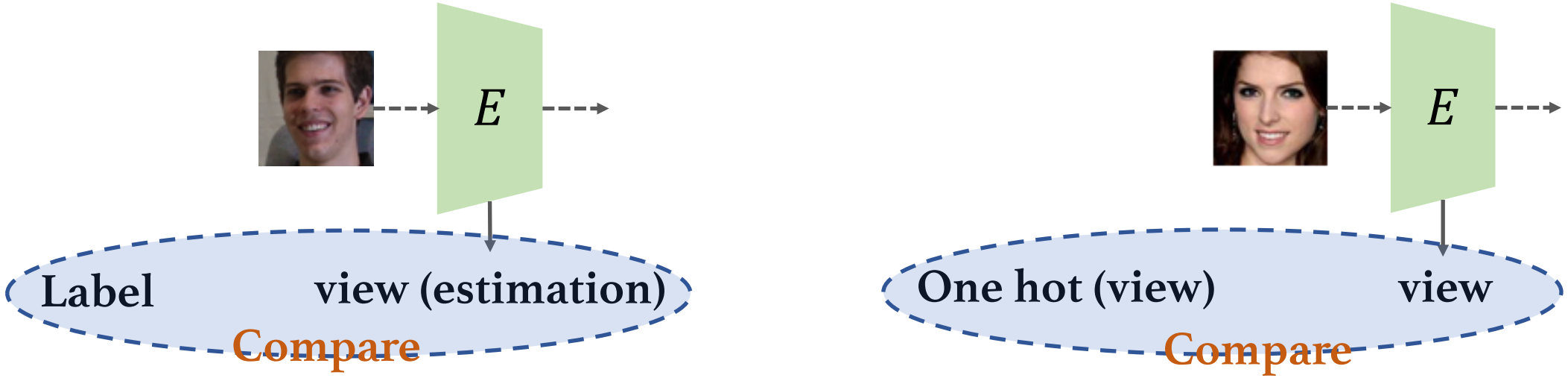
E minimizes:

$$\mathbb{E}_{\mathbf{x}_i,\mathbf{x}_j\sim\mathbb{P}_\mathbf{x}}[D_s(\tilde{\mathbf{x}}_j) + \lambda_3 P(D_v(\tilde{\mathbf{x}}_j) = v_j) -$$
$$\lambda_4 L_1(\tilde{\mathbf{x}}_j, \mathbf{x}_j) - \lambda_5 L_v(E_v(\mathbf{x}_i), v_i)],$$

Y. Tian et al. “CR-GAN: Learning Complete Representations for Multi-view Generation”. IJCAI’18

CR-GAN: Self-supervised learning

- Incorporate **unlabeled** data in training.
- Stage 1: training with labeled data, E be a good view estimator.
- Stage 2: let One hot (view) be the label of view.



Y. Tian et al. "CR-GAN: Learning Complete Representations for Multi-view Generation". IJCAI'18

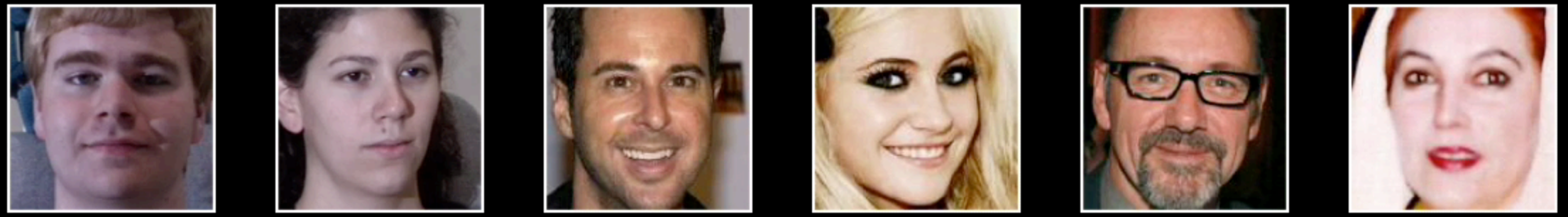
Experimental Results: Multi-view Generation

- Training data: Multi-PIE (labeled), 300wLP (labeled), CelebA (unlabeled).
- Test data: Multi-PIE, CelebA, IJB-A (unseen data).

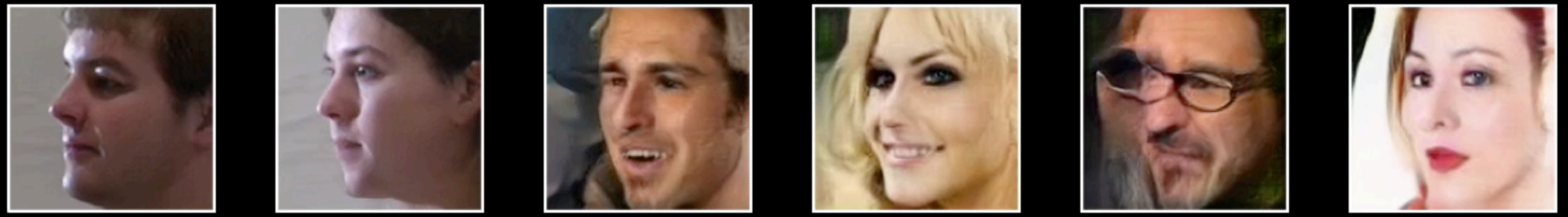
Y. Tian et al. “CR-GAN: Learning Complete Representations for Multi-view Generation”. IJCAI’18

Viewpoint: -60^0

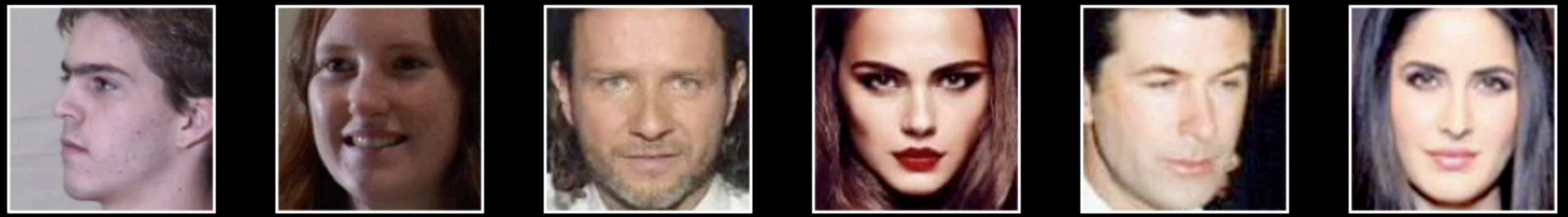
Inputs



CR-GAN



Inputs



CR-GAN

Experimental Results: single pathway vs. two pathway

- Training data: Multi-PIE (labeled).

Multi-PIE:

single:

two path:

(a)

IJB-A:

single:

two path:

(b)

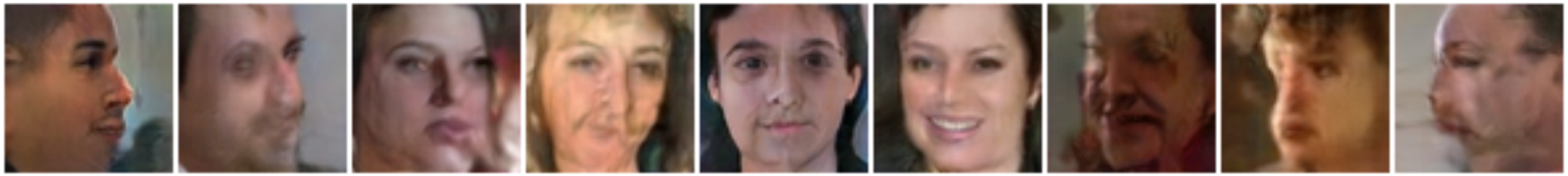
Experimental Results: supervised vs. self supervised

- Training data: supervised: Multi-PIE + 300wLP
- Training data: self-supervised: Multi-PIE + 300wLP + CelebA

Experimental Results: Compare with DR-GAN

(a) Generation from random noise:

DR-GAN



CR-GAN

CR-GAN: Y. Tian et al. “CR-GAN: Learning Complete Representations for Multi-view Generation”. IJCAI’18
DR-GAN: L. Tran et al. “Disentangled Representation Learning GAN for Pose-Invariant Face Recognition”. CVPR’17

Experimental Results: Compare with DR-GAN

(b) Multi-view generation on IJB-A:

DR-GAN

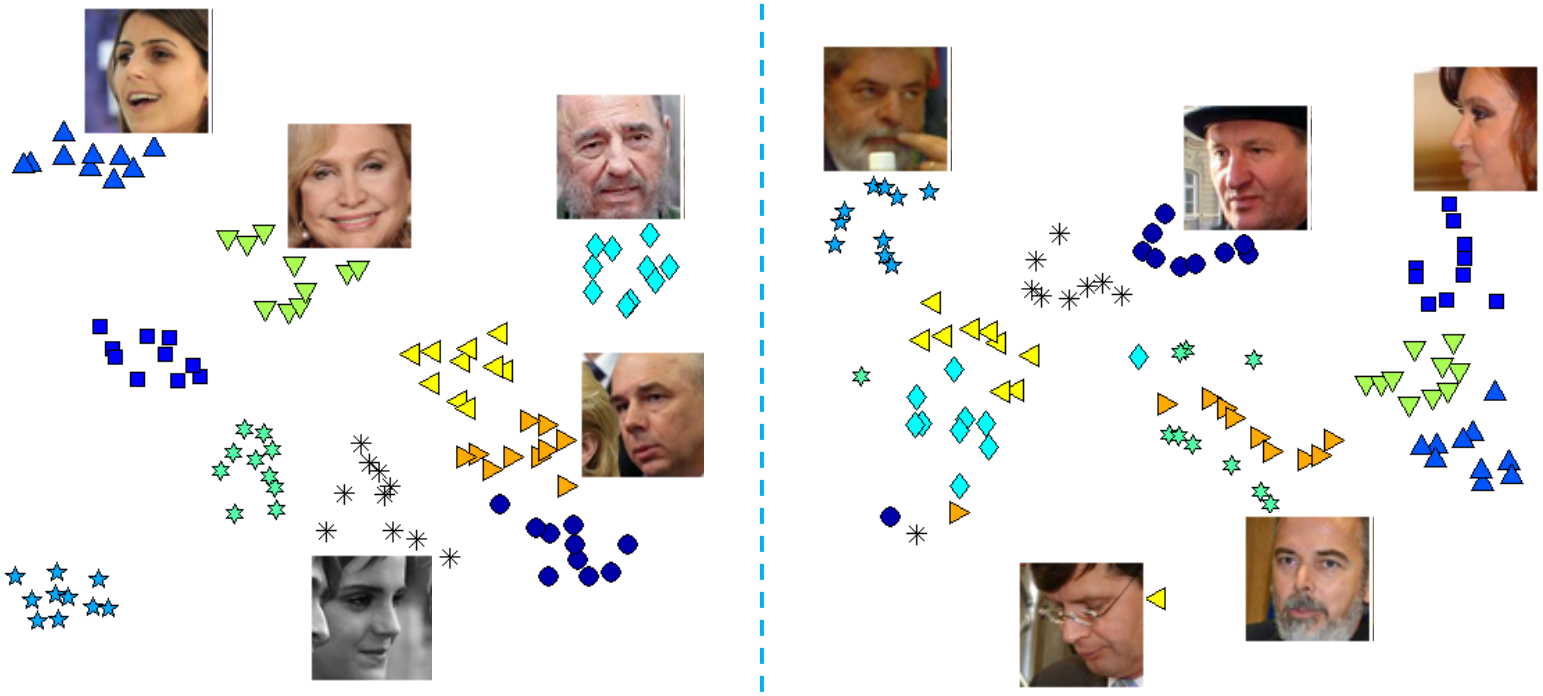
CR-GAN

DR-GAN

CR-GAN

Experimental Results: Compare with DR-GAN

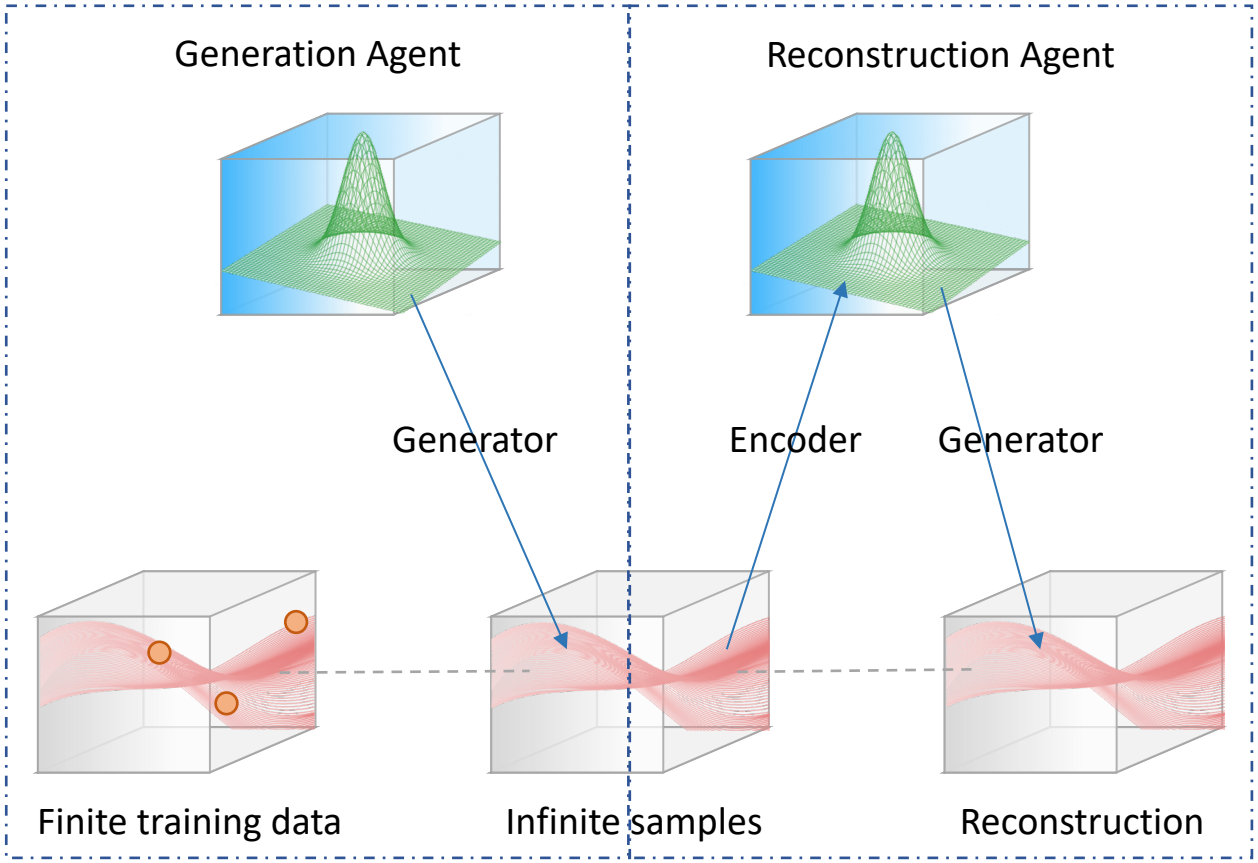
(c) Identity similarities:



	DR-GAN	CR-GAN
Multi-PIE	1.073±0.013	**1.018±0.019**
CelebA	1.281±0.007	**1.214±0.009**
IJB-A	1.295±0.008	**1.217±0.010**

Dual Agent learning: Method

- Dual-agent framework.
- Generation Agent: generate infinite samples similar to real data.
- Reconstruction Agent: reconstruct Generation Agent's output.
- Generation Agent is a "regularizer" of the single pathway approach.

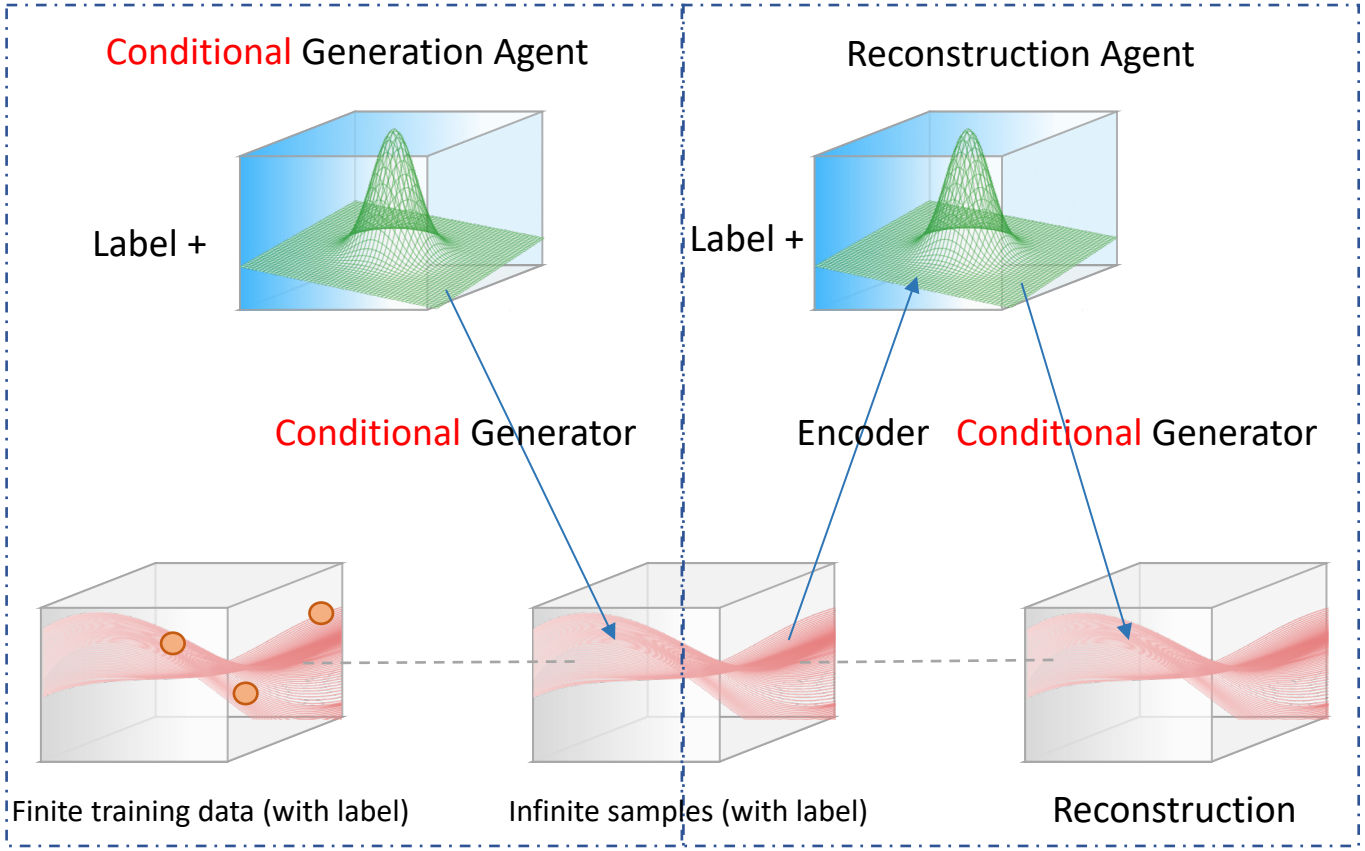


Y. Tian et al. "A dual-agent learning paradigm for improved bidirectional adversarial learning". NIPS'18 (under review)

Dual Agent learning: Method

Extension: Disentanglement learning.

- Labels as input of Generation Agent.
- Conditional Generation Agent: generates samples under the label. -> learns to disentangle label from other inputs.
- Reconstruction Agent: reconstruct Generation Agent's output.



Y. Tian et al. "A dual-agent learning paradigm for improved bidirectional adversarial learning". NIPS'18 (under review)

Dual Agent learning: theories

- By further reconstructing on training data, Dual Agent learning covers all modes. → better generation.
- More stable mapping in representation space.

Y. Tian et al. “A dual-agent learning paradigm for improved bidirectional adversarial learning”. NIPS’18 (under review)

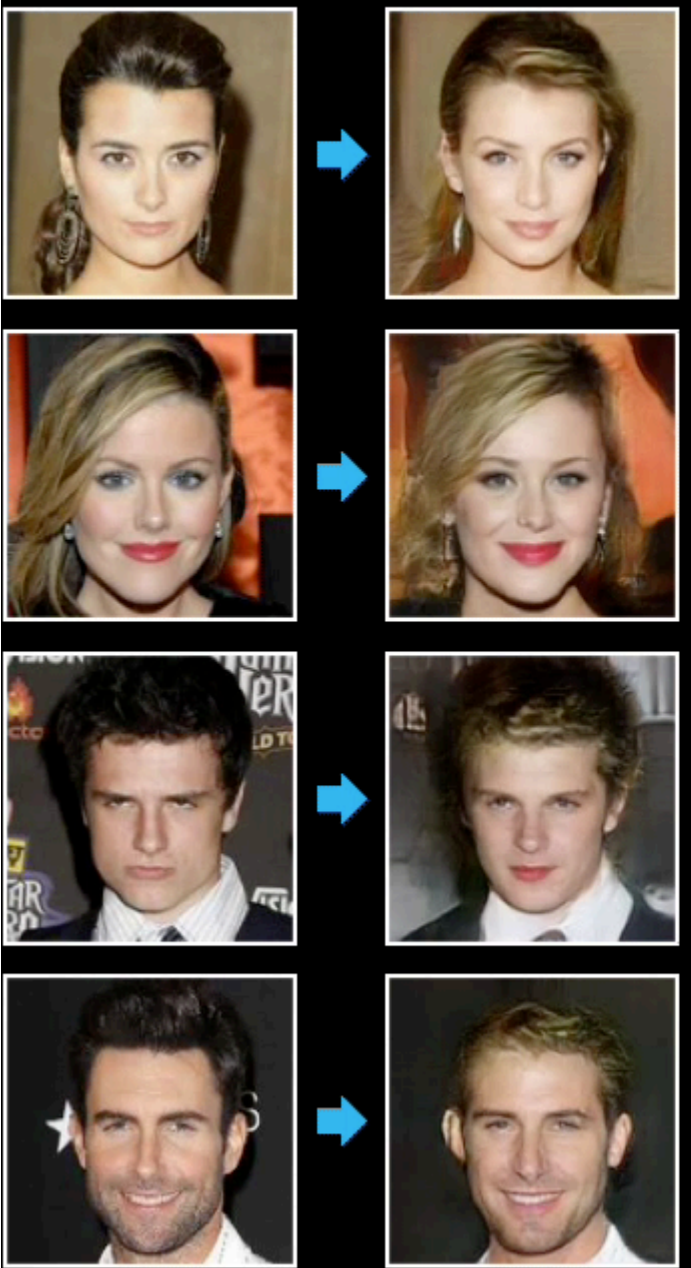
“Creativity”
Experimental results: Facial attributes manipulation

- Training data: CelebA (Attibutes labeled).
- Test data: CelebA.

Y. Tian et al. “A dual-agent learning paradigm for improved bidirectional adversarial learning”. NIPS’18 (under review)

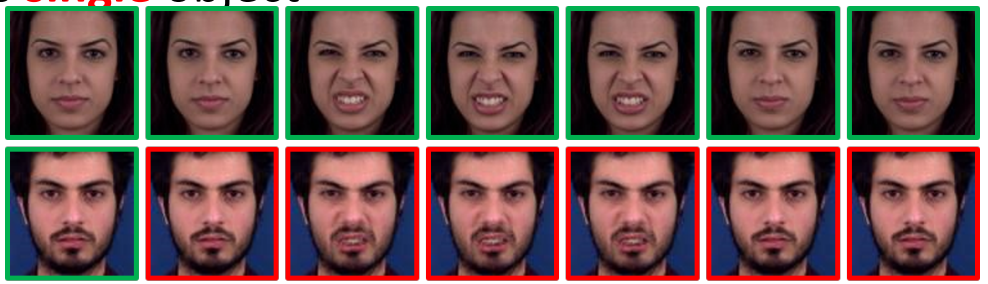
Input

Blond hair



Learning to learn for dynamic data generation: motion forecasting and video generation (ECCV'18 and ECCV'2020)

- A dynamic data driven problem:
  - Input: A **single** object in one or more images
  - Output: Generate a video containing motions of this **single** object
- Application:
  a) Facial Expression Retargeting
  b) Human Motion Forecasting
- Challenges:
  - Keep the object identity
  - Generate realistic-looking motions
  - Maintain video coherence



(a)

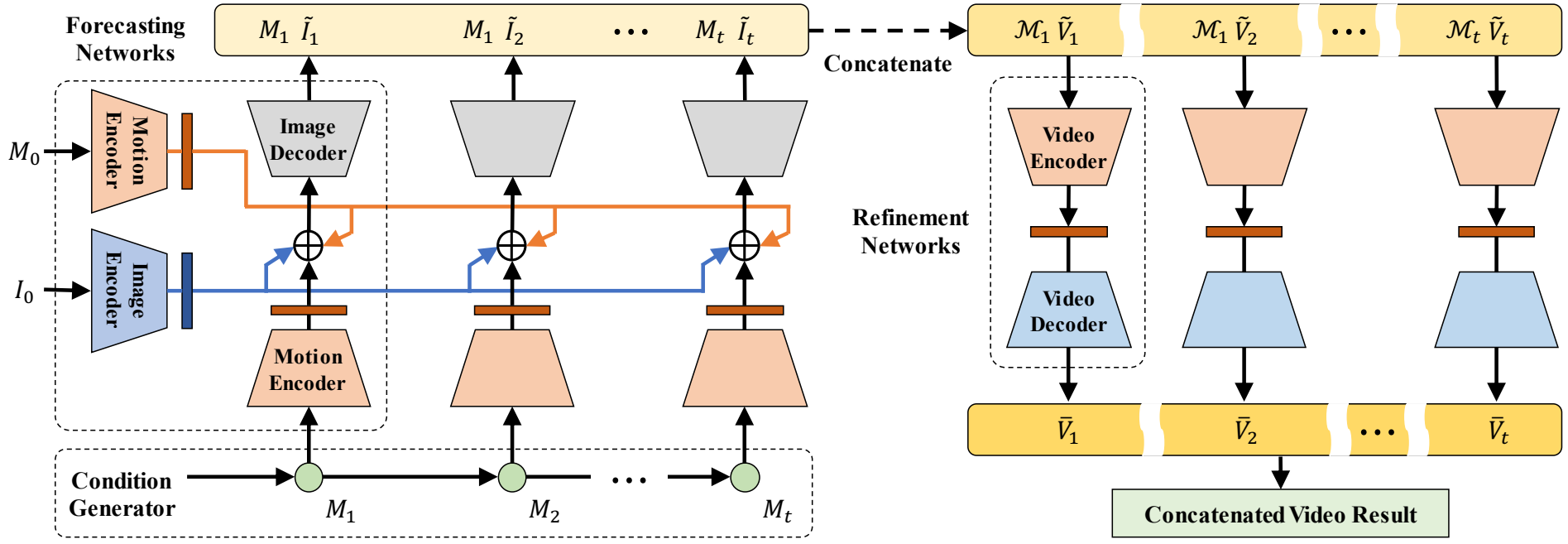
(b)

- Related Publications

[1] L. Zhao et al. "Learning to Forecast and Refine Residual Motion for Image-to-Video Generation". ECCV'18
[2] L. Zhao et al. "Sketch-Based Face Editing in Video Using Identity Deformation Transfer". TVCG, 2018 (under review)

Learning to learn for dynamic data generation: motion forecasting and video generation (ECCV'18)

- A **two-stage** generation framework: videos are (a) generated from conditions and then (b) refined. Our framework consists of three components: *a condition generator*, *motion forecasting networks* and *refinement networks*.

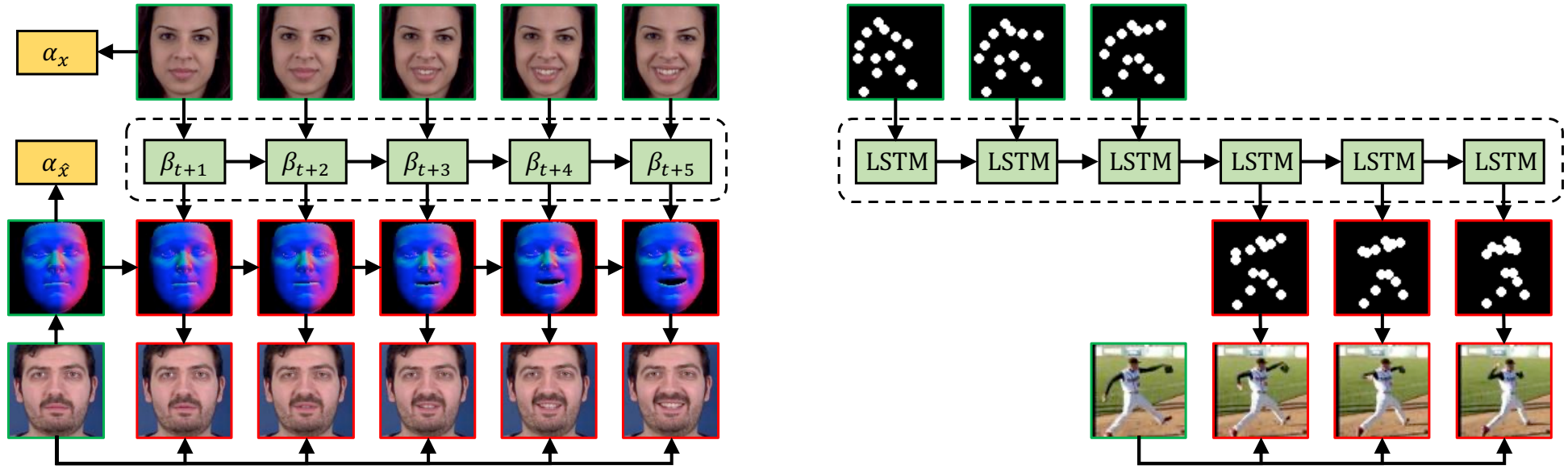


(a) Forecasting (b) Refinement

L. Zhao et al. "Learning to Forecast and Refine Residual Motion for Image-to-Video Generation". ECCV'18

Learning to learn for dynamic data generation: motion forecasting and video generation (ECCV'18)

- A **two-stage** generation framework based on GAN:
- Condition generator: Novelty – use domain knowledge to guide generation
  - (a) Facial expression retargeting: 3D Morphable Model to disentangle identity & expression
  - (b) Human motion forecasting: 2D positions of joints + LSTM

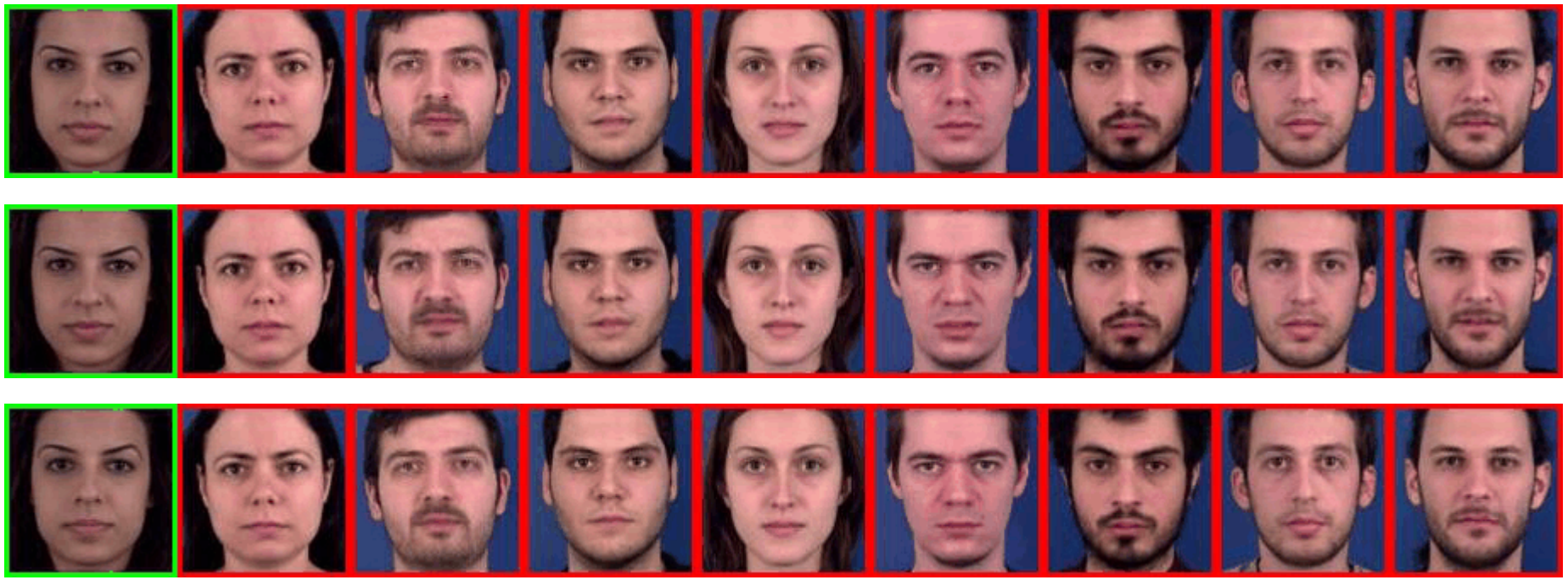


(a) (b)

L. Zhao et al. "Learning to Forecast and Refine Residual Motion for Image-to-Video Generation". ECCV'18

Learning to learn for dynamic data generation: motion forecasting and video generation (ECCV'18)

- Facial Expression Retargeting



[Ours] L. Zhao et al. "Learning to Forecast and Refine Residual Motion for Image-to-Video Generation". ECCV'18

Learning to learn for dynamic data generation: motion forecasting and video generation (ECCV'18)

- Human Motion Forecasting

[Ours] L. Zhao et al. "Learning to Forecast and Refine Residual Motion for Image-to-Video Generation". ECCV'18
[HP-GAN] Ruben Villegas et al. "Learning to Generate Long-term Future via Hierarchical Prediction". ICML'17

Learning to learn for dynamic data generation: motion forecasting and video generation

- Human Motion Forecasting (Baseball)

Ground Truth

Ours

HP-GAN

[Ours] L. Zhao et al. “Learning to Forecast and Refine Residual Motion for Image-to-Video Generation”. ECCV’18
[HP-GAN] Ruben Villegas et al. “Learning to Generate Long-term Future via Hierarchical Prediction”. ICML’17

Learning to learn for dynamic data generation: motion forecasting and video generation

- Human Motion Forecasting (Clean and Jerk)

[Ours] L. Zhao et al. “Learning to Forecast and Refine Residual Motion for Image-to-Video Generation”. ECCV’18
[HP-GAN] Ruben Villegas et al. “Learning to Generate Long-term Future via Hierarchical Prediction”. ICML’17

Learning to learn for dynamic data generation: motion forecasting and video generation

- Human Motion Forecasting (Golf Swing)

[Ours] L. Zhao et al. “Learning to Forecast and Refine Residual Motion for Image-to-Video Generation”. ECCV’18
[HP-GAN] Ruben Villegas et al. “Learning to Generate Long-term Future via Hierarchical Prediction”. ICML’17

Learning to learn for dynamic data generation: motion forecasting and video generation

- Human Motion Forecasting (Jump Rope)

[Ours] L. Zhao et al. “Learning to Forecast and Refine Residual Motion for Image-to-Video Generation”. ECCV’18
[HP-GAN] Ruben Villegas et al. “Learning to Generate Long-term Future via Hierarchical Prediction”. ICML’17

Learning to learn for dynamic data generation: motion forecasting and video generation

- Human Motion Forecasting (Jumping Jacks)

[Ours] L. Zhao et al. “Learning to Forecast and Refine Residual Motion for Image-to-Video Generation”. ECCV’18
[HP-GAN] Ruben Villegas et al. “Learning to Generate Long-term Future via Hierarchical Prediction”. ICML’17

Learning to learn for dynamic data generation: motion forecasting and video generation

- Human Motion Forecasting (Tennis)

[Ours] L. Zhao et al. “Learning to Forecast and Refine Residual Motion for Image-to-Video Generation”. ECCV’18
[HP-GAN] Ruben Villegas et al. “Learning to Generate Long-term Future via Hierarchical Prediction”. ICML’17

New ways to track accurately faces and bodies

Motivation

- Previous work computes eyebrow height from the landmarks extracted via face trackers, which is not accurate enough due to occlusions, large head pose changes and video focus issues.
- Our model is trained to predict eyebrow height generated from high-level annotations, without reliance on face trackers.

Our Approach

- Resnet is used to extract and embed the visual cues, such as wrinkles in the forehead from frames
- LSTM is used to extract complementary information from nearby frames to handle occlusions.

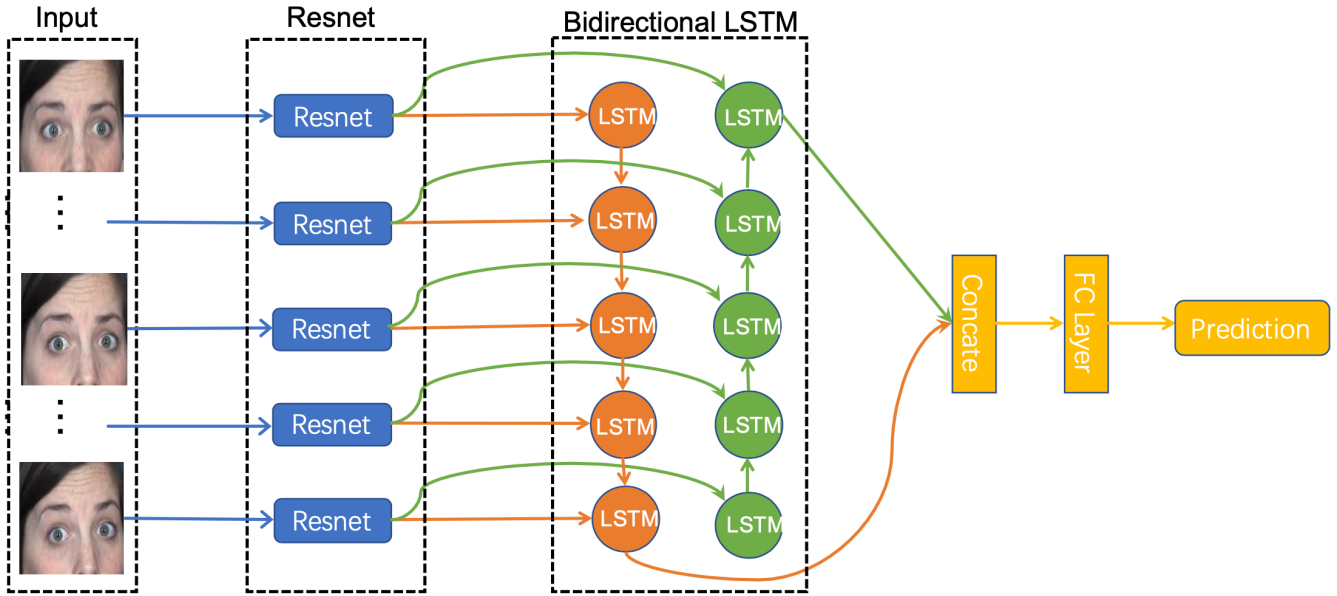
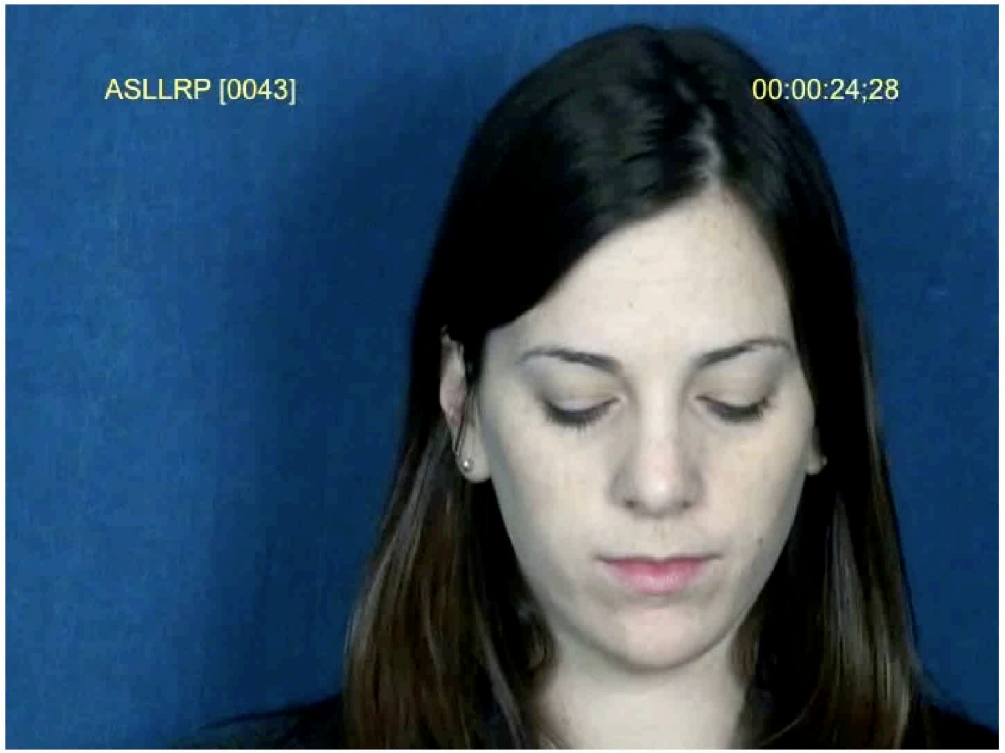
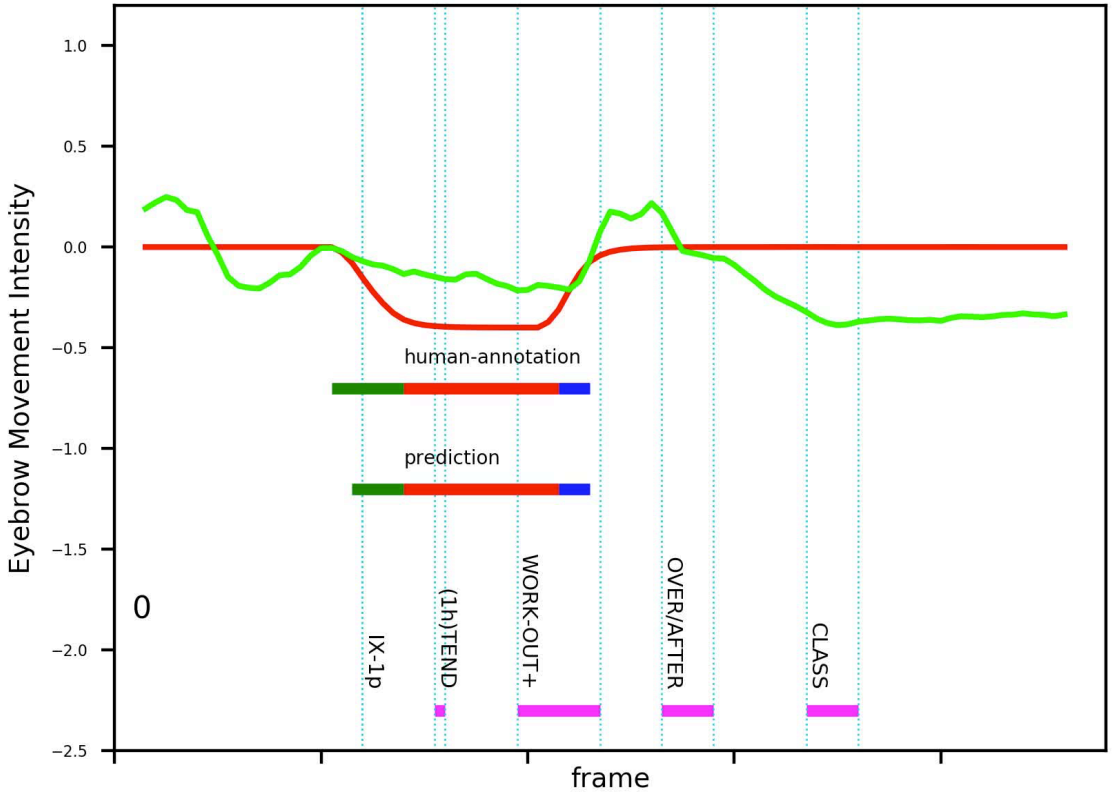


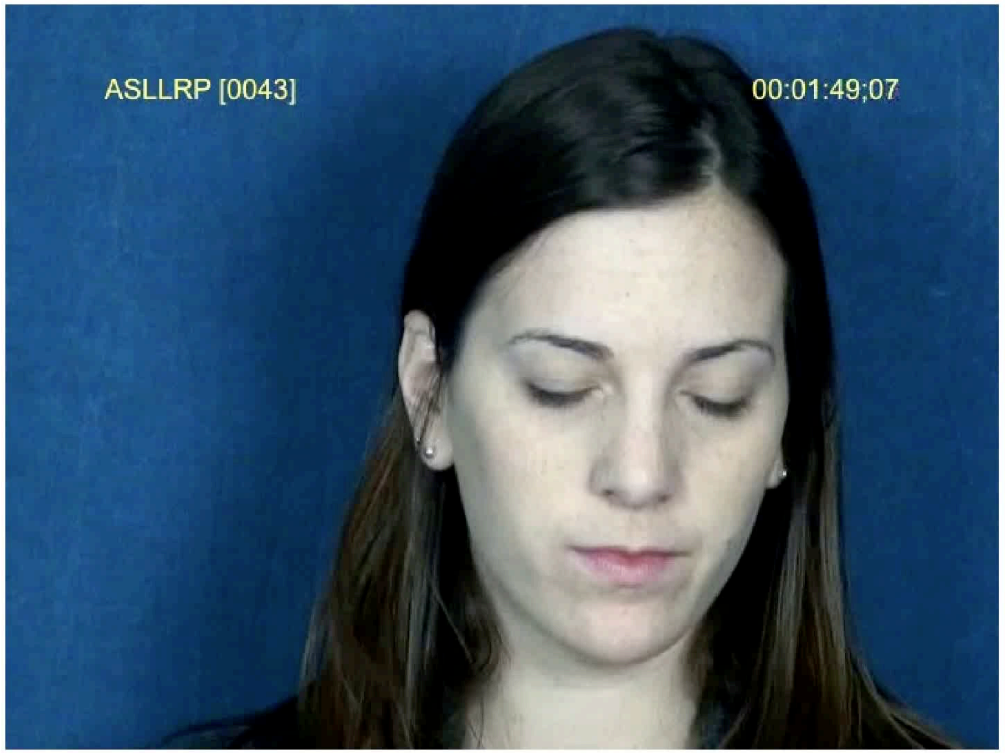
Fig. 2: Illustration of our network for eyebrow deformation intensity estimation.

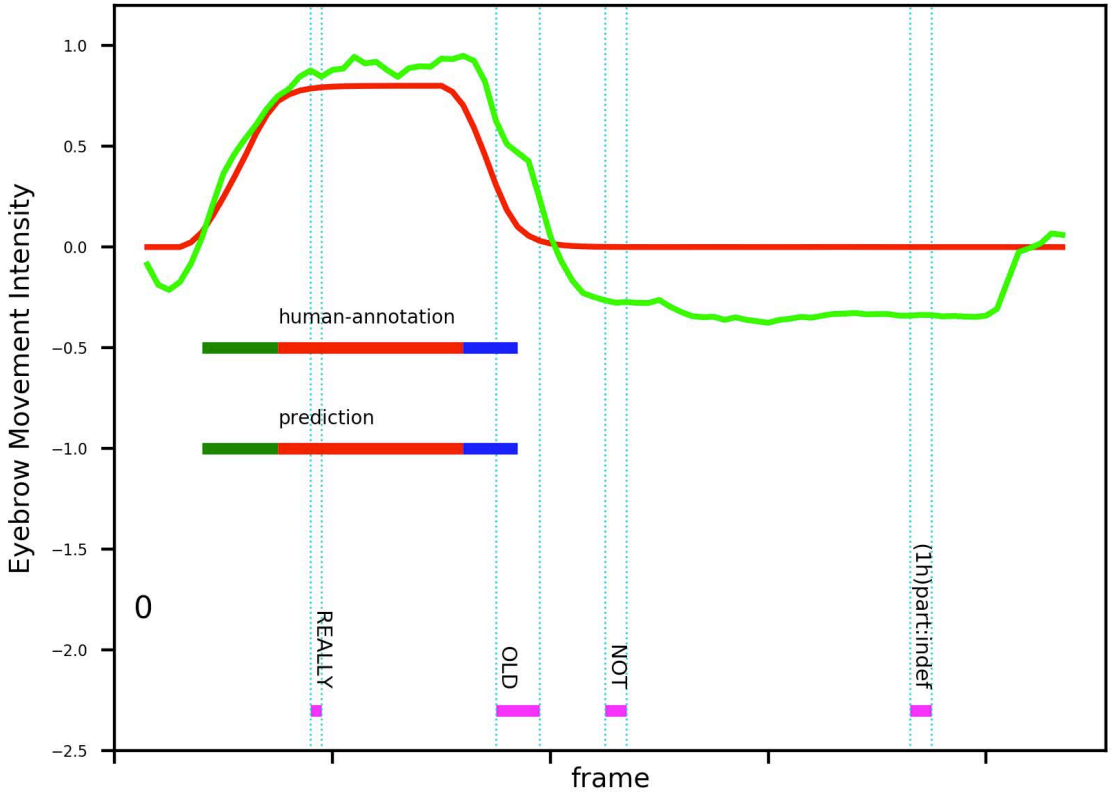
Result


ASLLRP [0043]
00:00:24;28

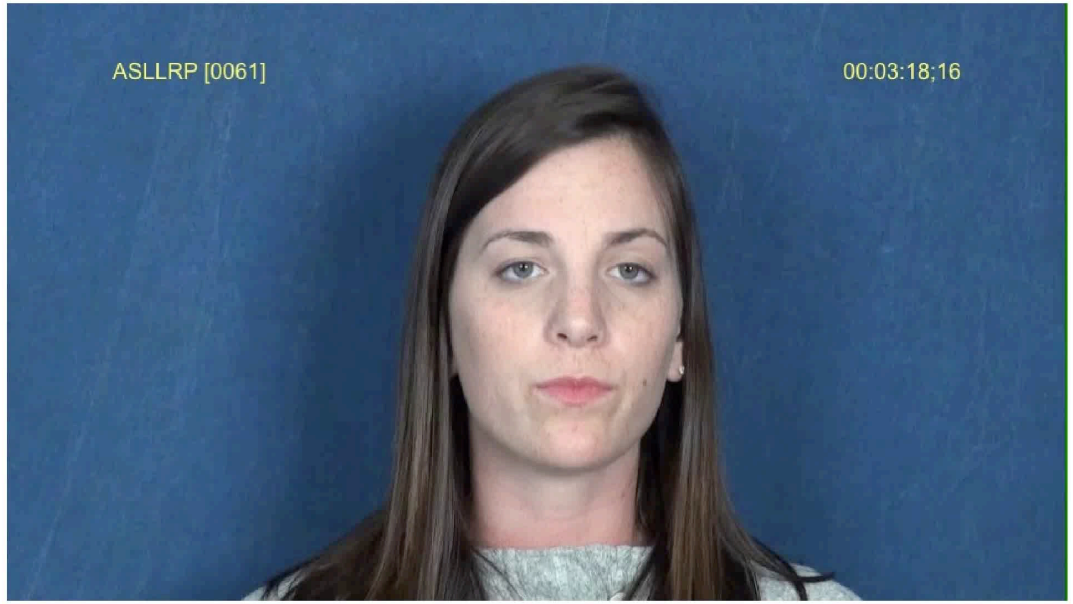

Eyebrow Movement Intensity
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
human-annotation
prediction
0
IX-1p
(1h)TEND
WORK-OUT+
OVER/AFTER
CLASS
frame

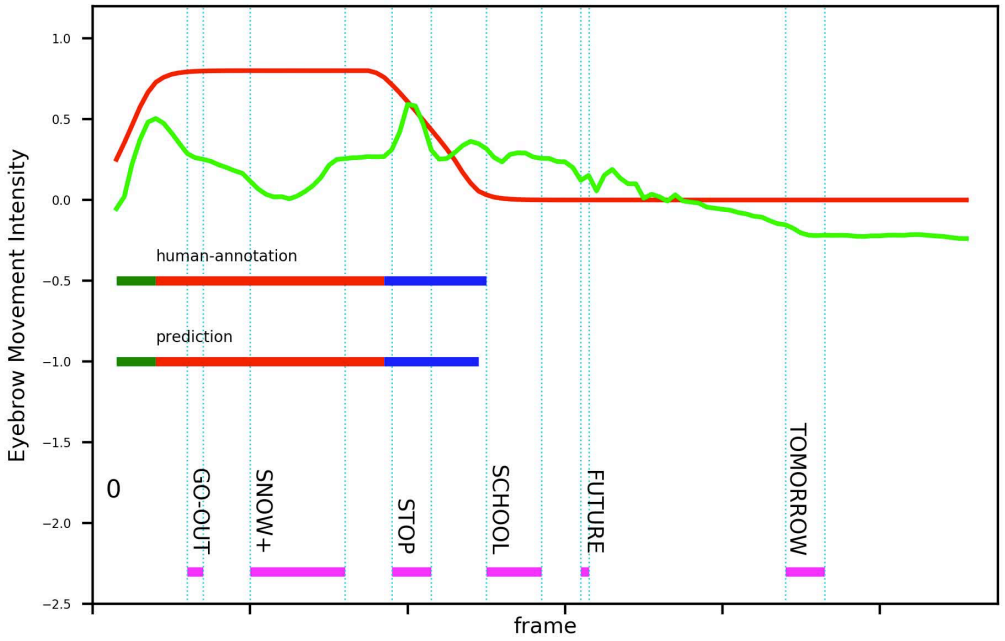
Result


ASLLRP [0043]
00:01:49;07


Eyebrow Movement Intensity
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
human-annotation
prediction
0
REALLY
OLD
NOT
(1h)part:indef
frame

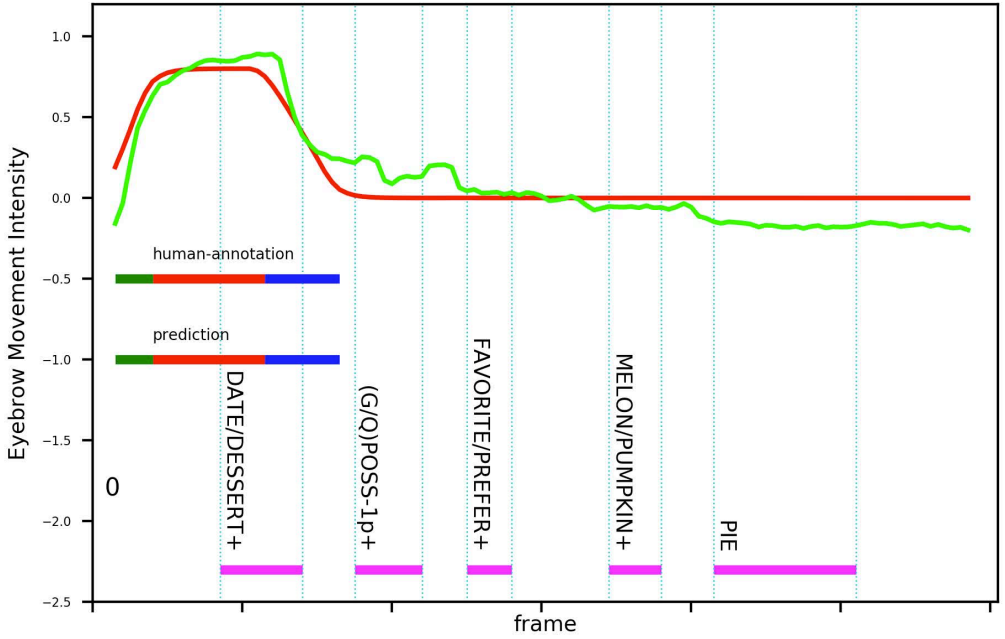
Result


ASLLRP [0061]
00:03:18;16


Eyebrow Movement Intensity
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
human-annotation
prediction
0
GO-OUT
SNOW+
STOP
SCHOOL
FUTURE
TOMORROW
frame

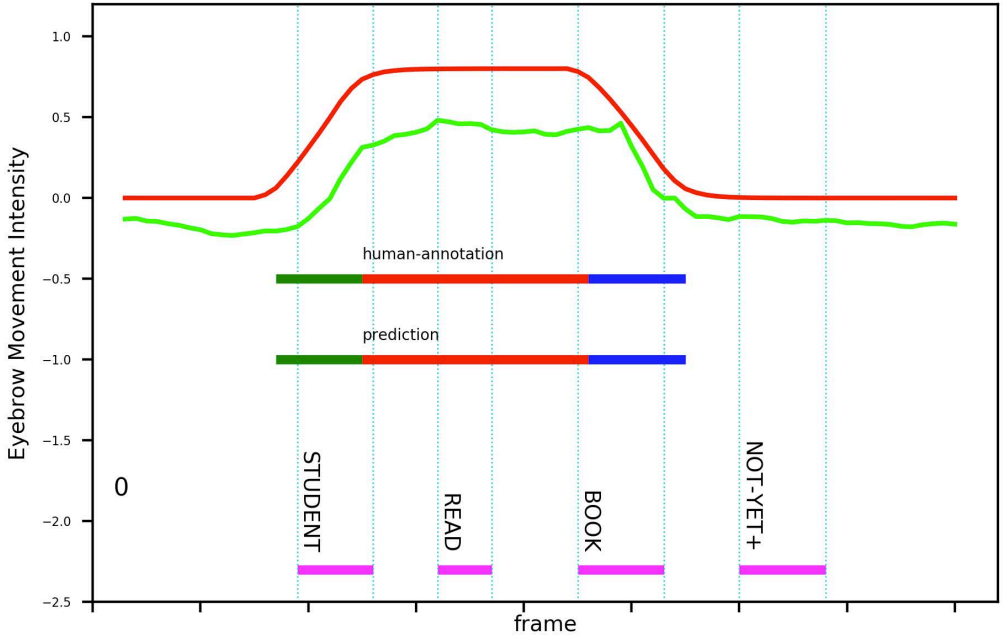
Result


ASLLRP [0107]
00:08:35;21


Eyebrow Movement Intensity
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
human-annotation
prediction
0
DATE/DESSERT+
(G/Q)POSS-1p+
FAVORITE/PREFER+
MELON/PUMPKIN+
PIE
frame

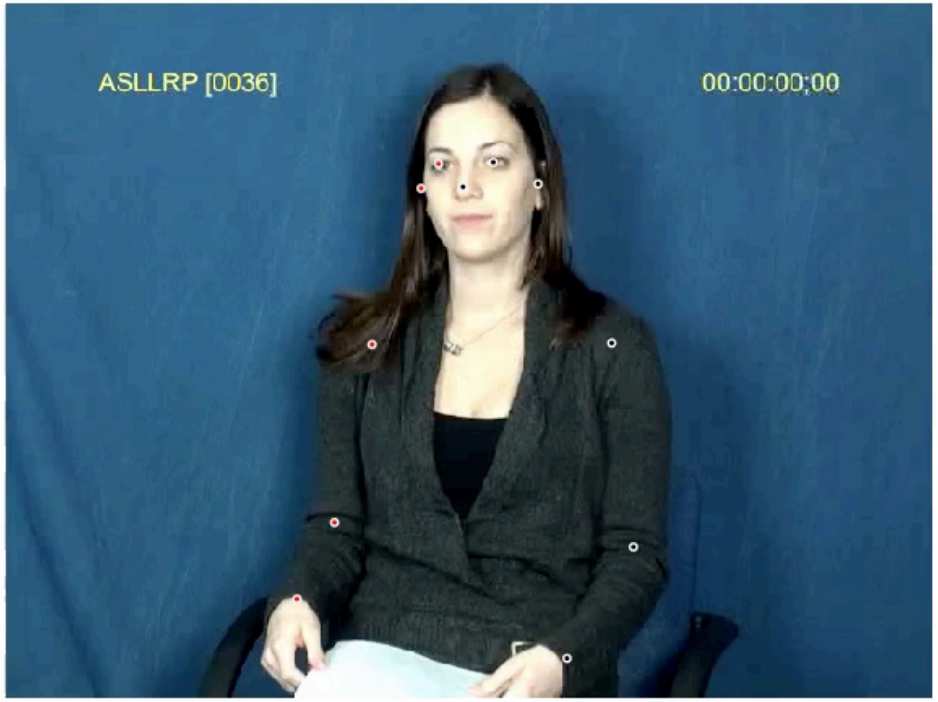
Result


ASLLRP [0107]
00:09:41;10

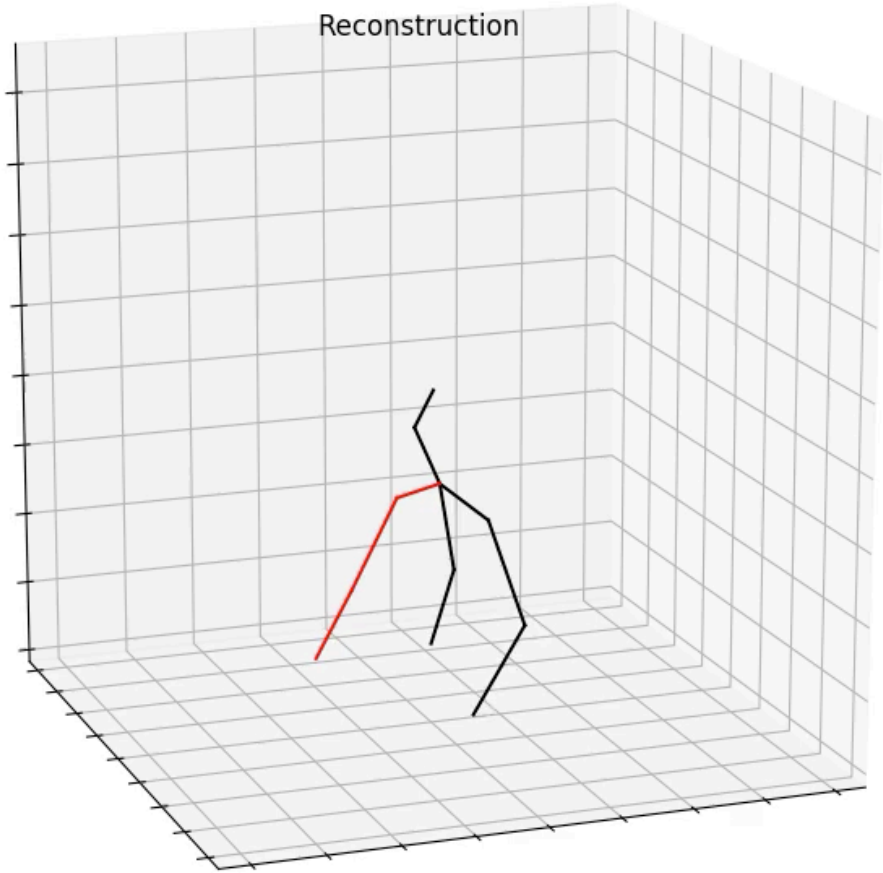

Eyebrow Movement Intensity
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
human-annotation
prediction
0
STUDENT
READ
BOOK
NOT-YET+
frame

Tracking Bodies and hands for Structured Environments ASL

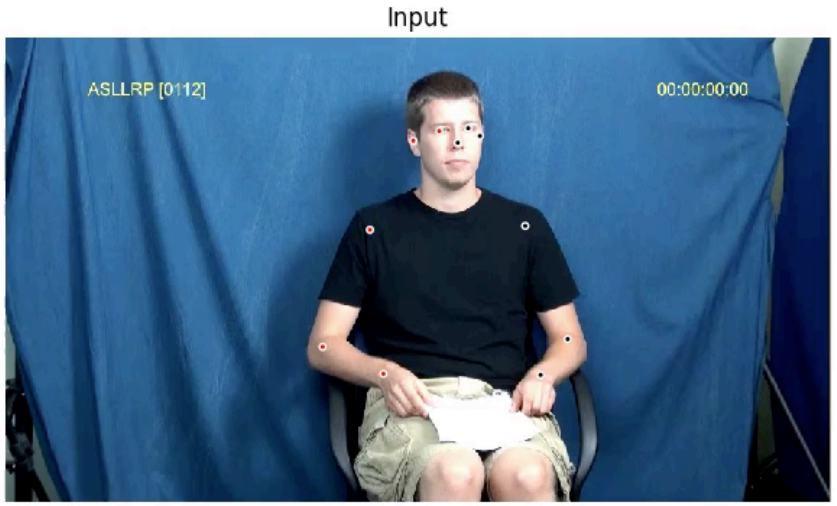
Input

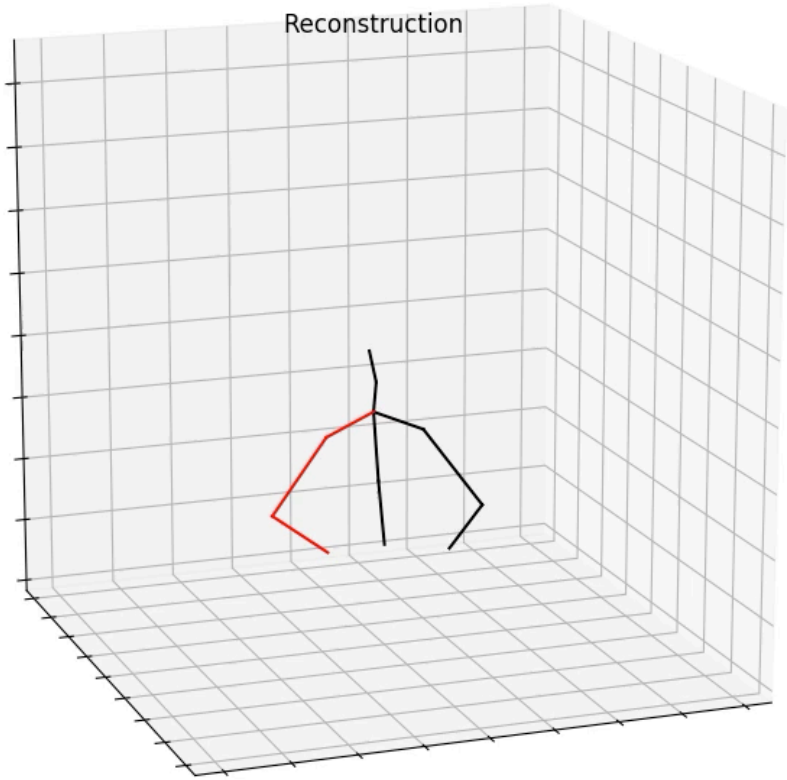


Reconstruction



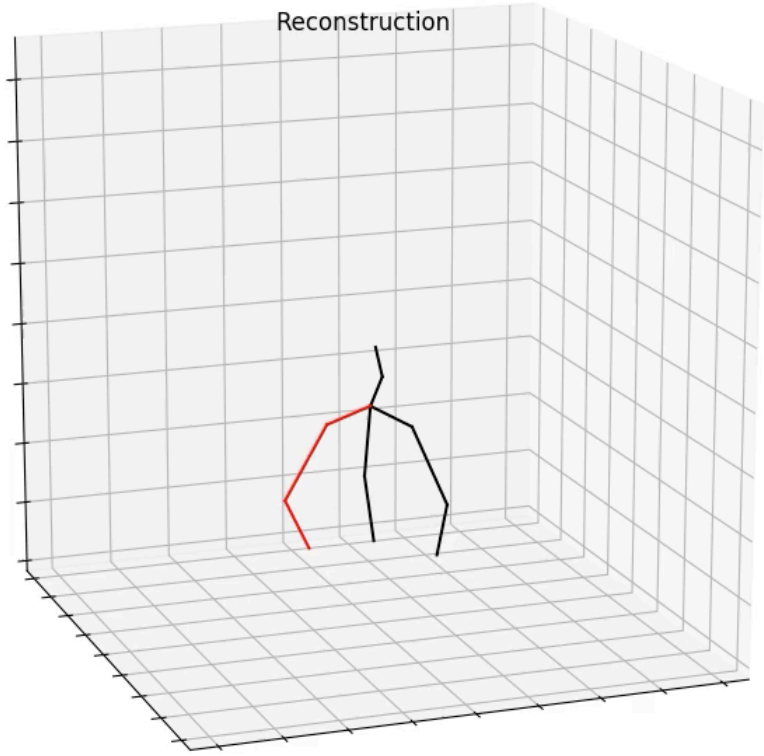
Tracking Bodies and hands for Structured Environments ASL





Tracking Bodies and hands for Structured Environments ASL

Input



Knowledge as Priors: Cross-Modal Knowledge Generalization for Datasets without Superior Knowledge

Long Zhao[1] Xi Peng[2] Yuxiao Chen[1] Mubbasir Kapadia[1] Dimitris Metaxas[1]

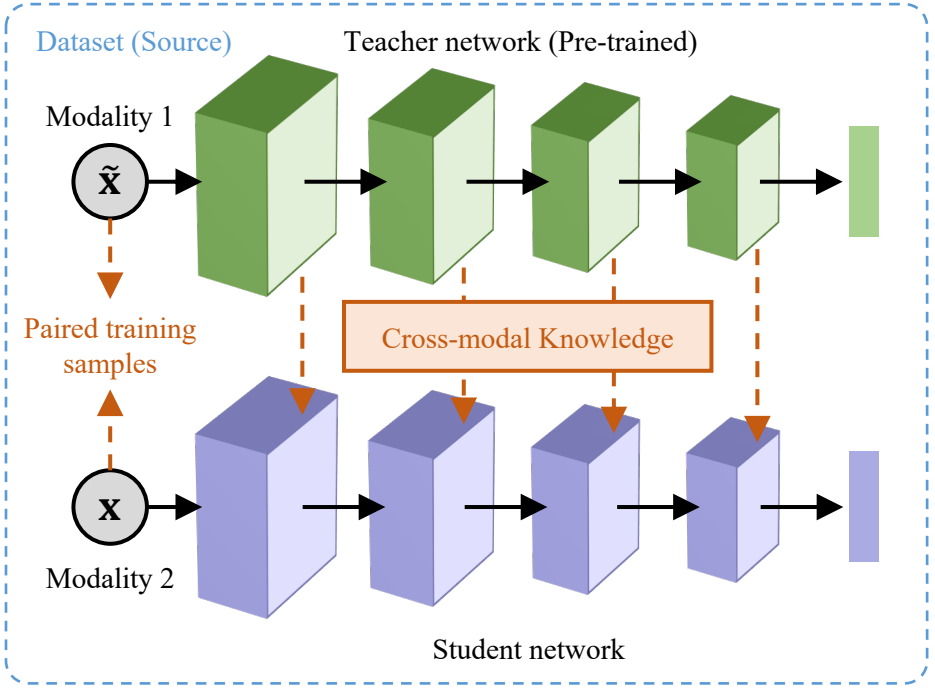
[1] Department of Computer Science, Rutgers University

[2] Department of Computer & Information Sciences, University of Delaware

CVPR SEATTLE WASHINGTON JUNE 16-18 2020

What is Cross-Modal Knowledge Generalization?

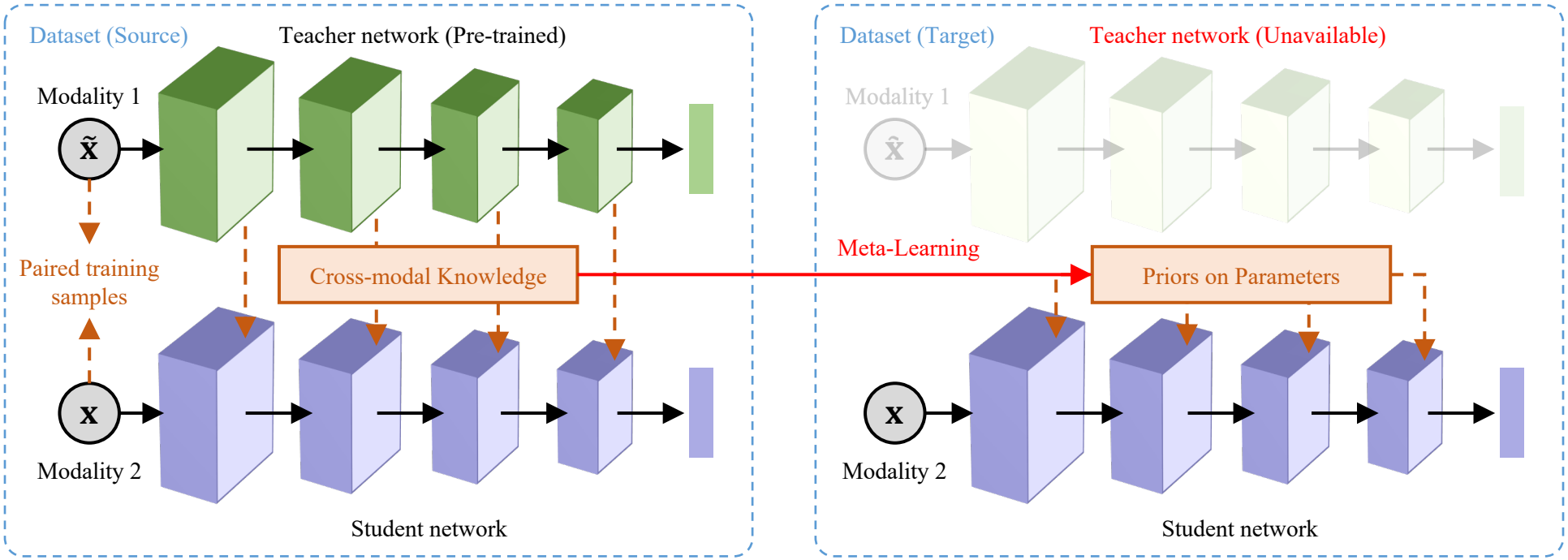
- Existing approaches distill cross-modal knowledge from the teacher to student in one dataset.



(a) Knowledge Distillation

What is Cross-Modal Knowledge Generalization?

- Existing approaches distill cross-modal knowledge from the teacher to student in one dataset.
- We propose cross-modal knowledge generalization which transfers learned knowledge in the source to a target dataset where the superior knowledge, i.e., the teacher, is unavailable.



(a) Knowledge Distillation

(b) Knowledge Generalization

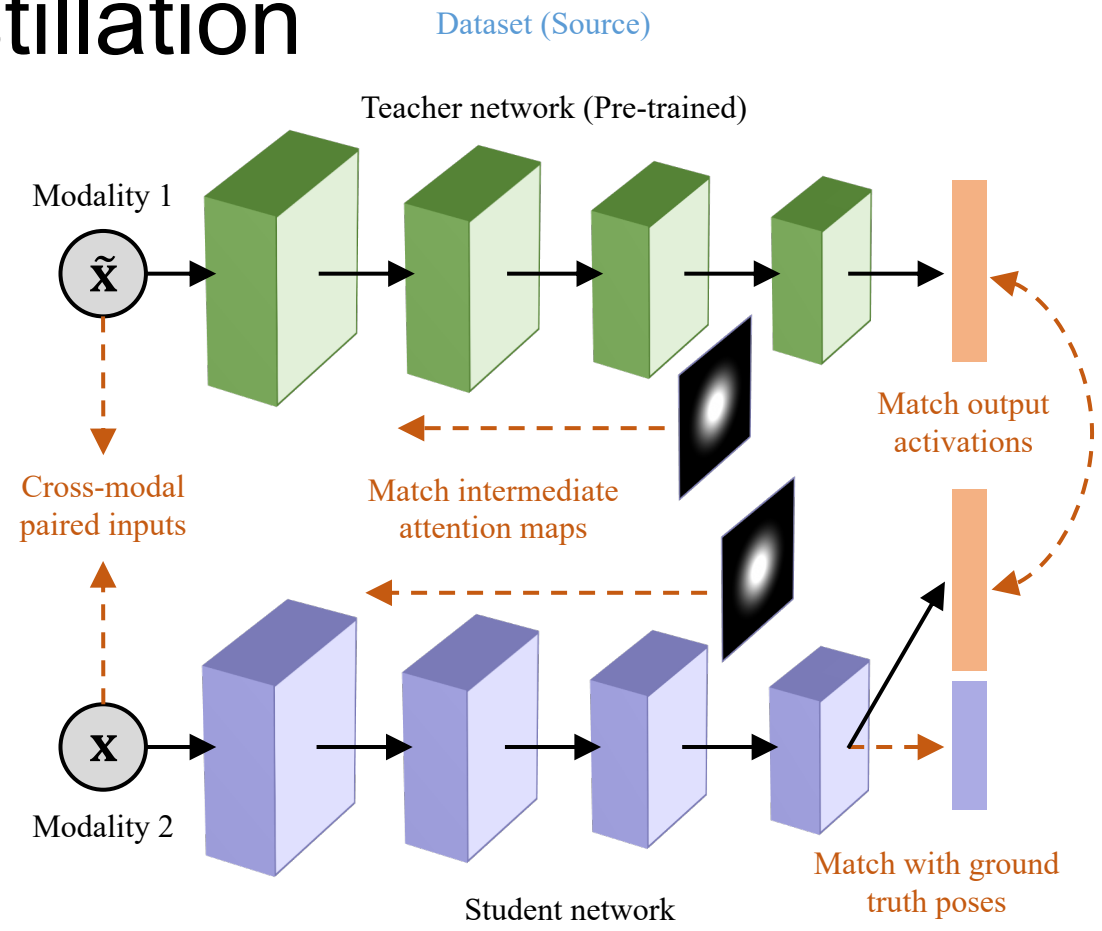
Cross-Modal Knowledge Distillation

- The goal of cross-modal knowledge distillation is to improve the learning process by transferring the knowledge from the teacher to student.

- Regression Loss $\mathcal{L}_{\text{REG}}$
- Activation Loss $\mathcal{L}_{\text{ACT}}$
- Attention loss $\mathcal{L}_{\text{ATT}}$

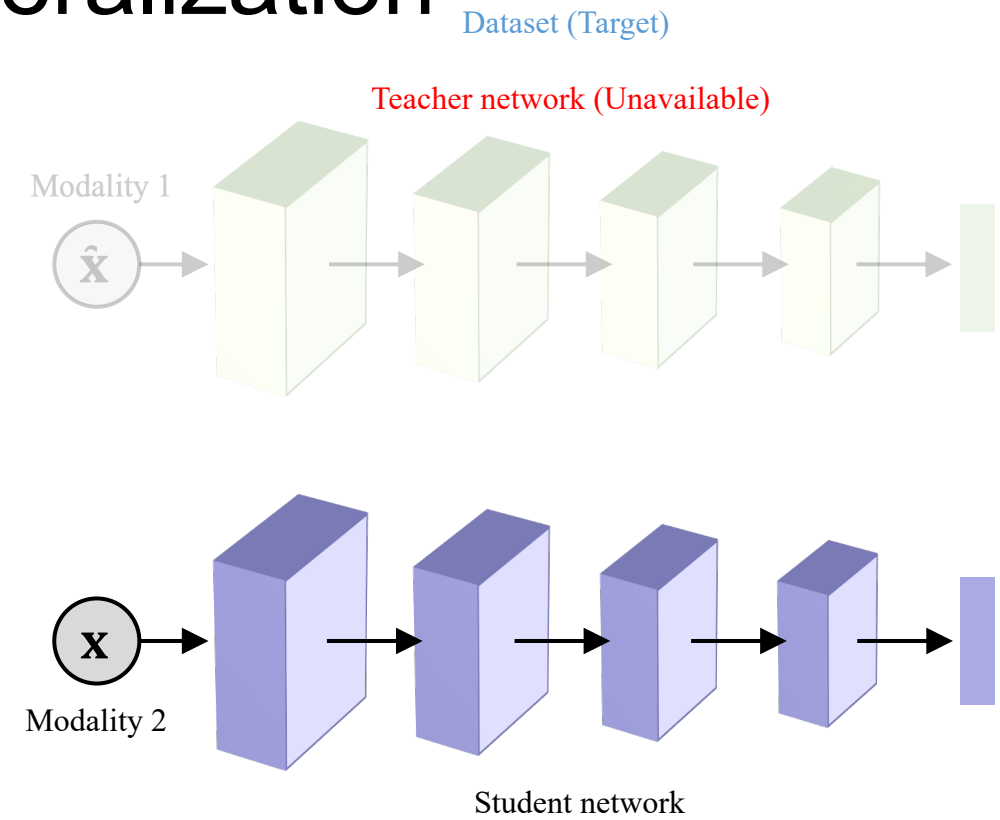
$$\mathcal{L}_{\text{DIST}} = \mathcal{L}_{\text{ACT}} + \lambda \cdot \mathcal{L}_{\text{ATT}}$$

$$\mathcal{G} = \mathcal{L}_{\text{REG}} + \mathcal{L}_{\text{DIST}}$$



Cross-Modal Knowledge Generalization

- The goal of cross-modal knowledge distillation leverages meta-learning to generalize the learned knowledge from the source dataset to the target dataset by treating it as priors on the parameters of the student network.



Source Dataset $\mathcal{G} = \mathcal{L}_{\text{REG}} + \mathcal{L}_{\text{DIST}}$

Target Dataset $\mathcal{F} = \mathcal{L}_{\text{REG}} + \cancel{\mathcal{L}_{\text{DIST}}}$

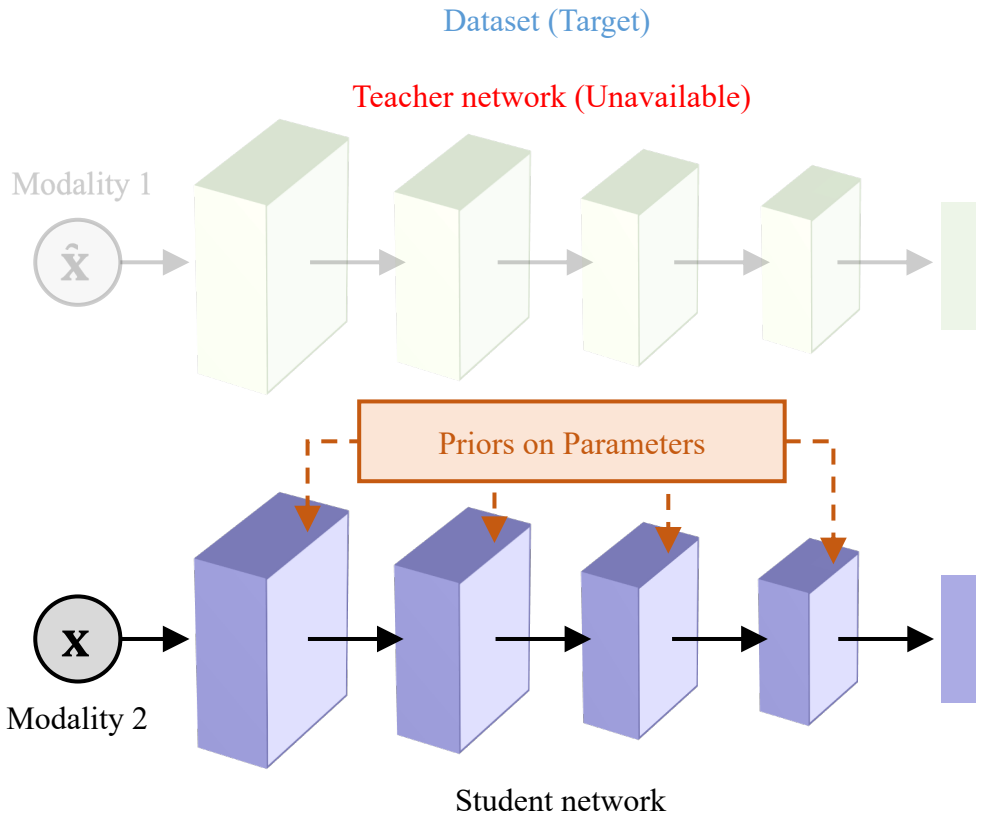
Cross-Modal Knowledge Generalization

- The goal of cross-modal knowledge distillation leverages meta-learning to generalize the learned knowledge from the source dataset to the target dataset by treating it as priors on the parameters of the student network.

Meta-Learning

Source Dataset $\mathcal{G} = \mathcal{L}_{\text{REG}} + \mathcal{L}_{\text{DIST}}$ $\longleftrightarrow$ $\mathcal{R}(\theta)$

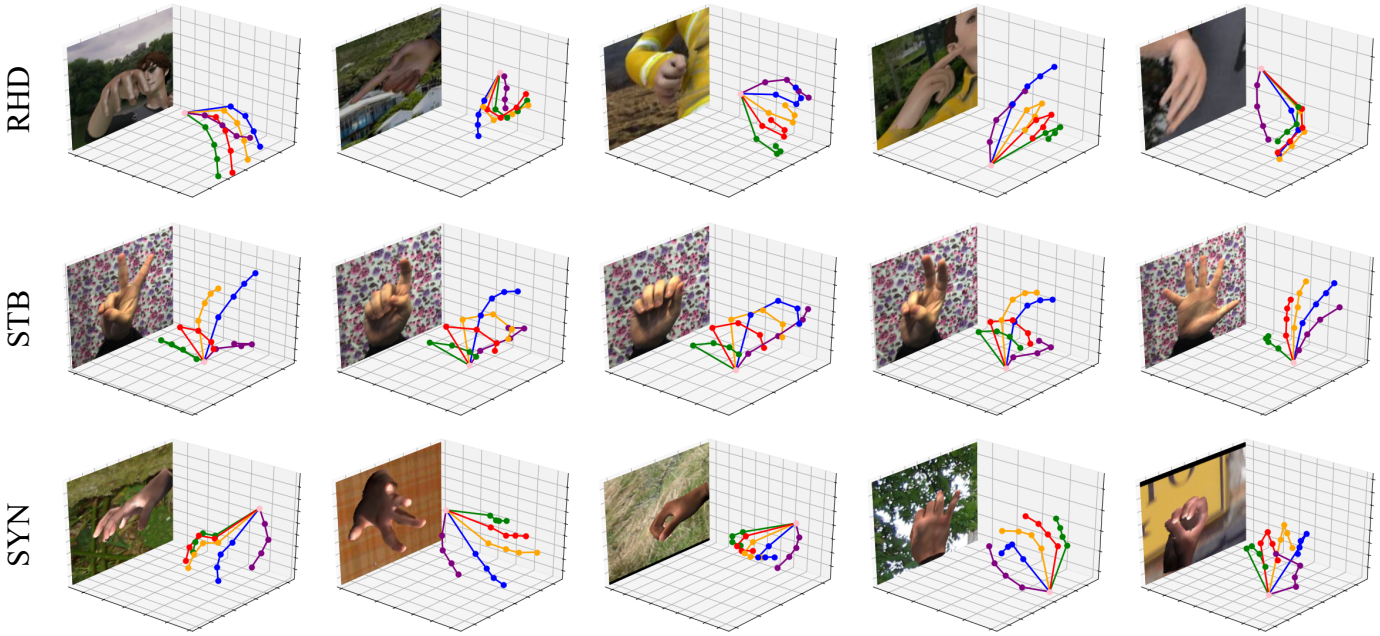
Target Dataset $\mathcal{F} = \mathcal{L}_{\text{REG}} + \mathcal{R}(\theta)$

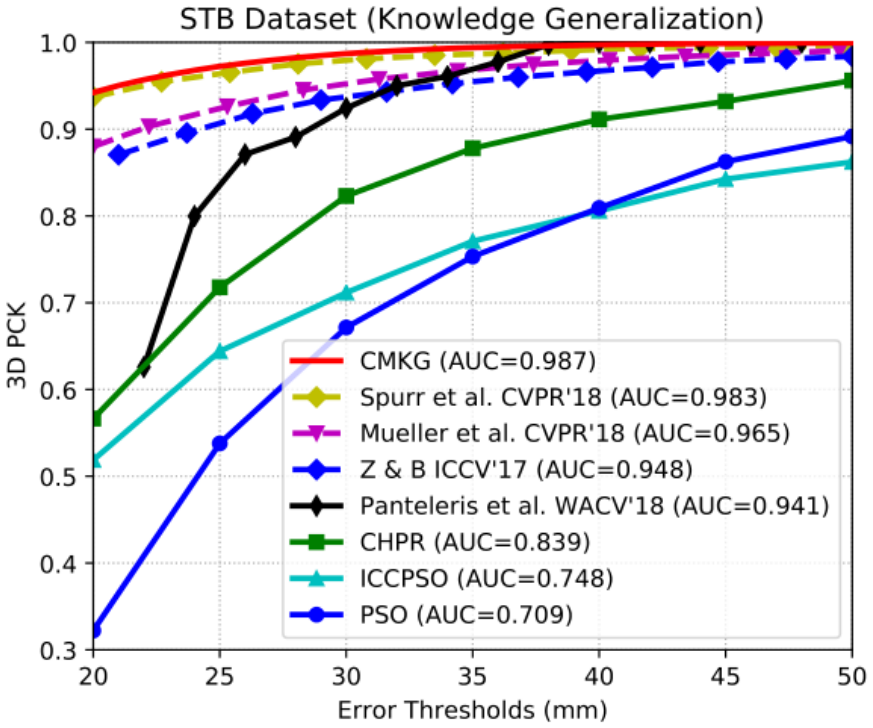


Results

- Quantitative Results (Source: RHD, Target: STB).

- We transfer knowledge from Depth maps to RBG images for 3D hand pose estimation.





Regularizer	EPE (mm)	AUC
None	15.67	0.915
$\ell^1, \sigma = 1.0 \times 10^{-4}$	$11.41_{\downarrow 4.26}$	$0.972_{\uparrow 0.057}$
$\ell^1, \sigma = 1.0 \times 10^{-6}$	$11.82_{\downarrow 3.85}$	$0.964_{\uparrow 0.049}$
$\ell^2, \sigma = 1.0 \times 10^{-3}$	$12.28_{\downarrow 3.39}$	$0.957_{\uparrow 0.042}$
$\ell^2, \sigma = 1.0 \times 10^{-5}$	$12.02_{\downarrow 3.65}$	$0.964_{\uparrow 0.049}$
$\mathcal{R}$, ℓ^1-regularized	$8.86_{\downarrow 6.81}$	$0.985_{\uparrow 0.070}$
$\mathcal{R}$, ℓ^2-regularized	$\underline{8.18}_{\downarrow 7.49}$	$\underline{0.987}_{\uparrow 0.072}$


Thank You!

Contact: Dimitris Metaxas, dnm@cs.rutgers.edu