Factorized Variational Autoencoders for Modeling Audience Reactions to Movies

WONG SHING MING

NATIONAL TSING HUA UNIV.

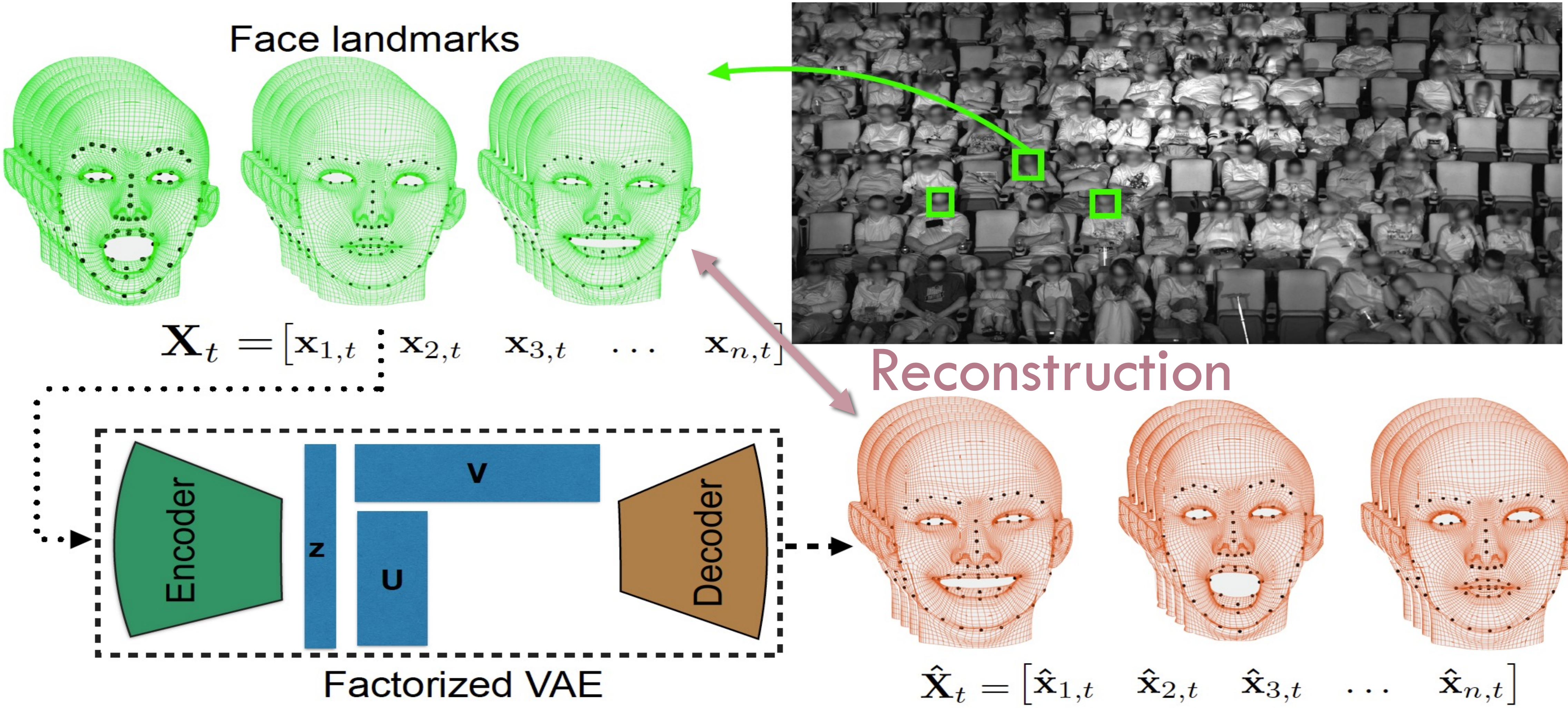
23 Mar. 2021

OUTLINE

Info. of paper

- CVPR 2017
- Times Cited: 54 (from Web of Science Core Collection at 23Mar2020)
- Disney Research
  - ➢Zhiwei Deng(Simon Fraser University), Rajitha Navarathna, Peter Carr, Stephan Mandt, Yisong Yue(Caltech), Iain Matthews, Greg Mori(Simon Fraser University)
- Acquire latent expression by matrix decomposition x VAE of the facial expressions of the audience watching the movie

Overview



Face landmarks
$\mathbf{X}_t = [\mathbf{x}_{1,t} \quad \mathbf{x}_{2,t} \quad \mathbf{x}_{3,t} \quad \dots \quad \mathbf{x}_{n,t}]$
Encoder
z
V
U
Decoder
Factorized VAE
Reconstruction
$\hat{\mathbf{X}}_t = [\hat{\mathbf{x}}_{1,t} \quad \hat{\mathbf{x}}_{2,t} \quad \hat{\mathbf{x}}_{3,t} \quad \dots \quad \hat{\mathbf{x}}_{n,t}]$

What field do they focus &
What difficulties do they face

Goal

- Learning a **representation** for a large dataset of facial expressions extracted from movie-watching audiences

Difficulties

- Matrix/Tensor data expression tends to be super high-dimensional
- Time varying facial expressions does not decompose linear

Hypothesis

- World data is made up of a small number of **patterns**
- there are **underlying example** facial expressions which form a **basis** to re-construct the observed reaction of each audience
- Learning a **representation** --> finding underlying low-dimension patterns

Related work – Audience Analysis

- Self-reports

➢Cons: **Subjective** and to **consciously** think about what they are watching

- Heart rate or galvanic skin response

➢Cons: **Obtrusive**, **Inhibited**

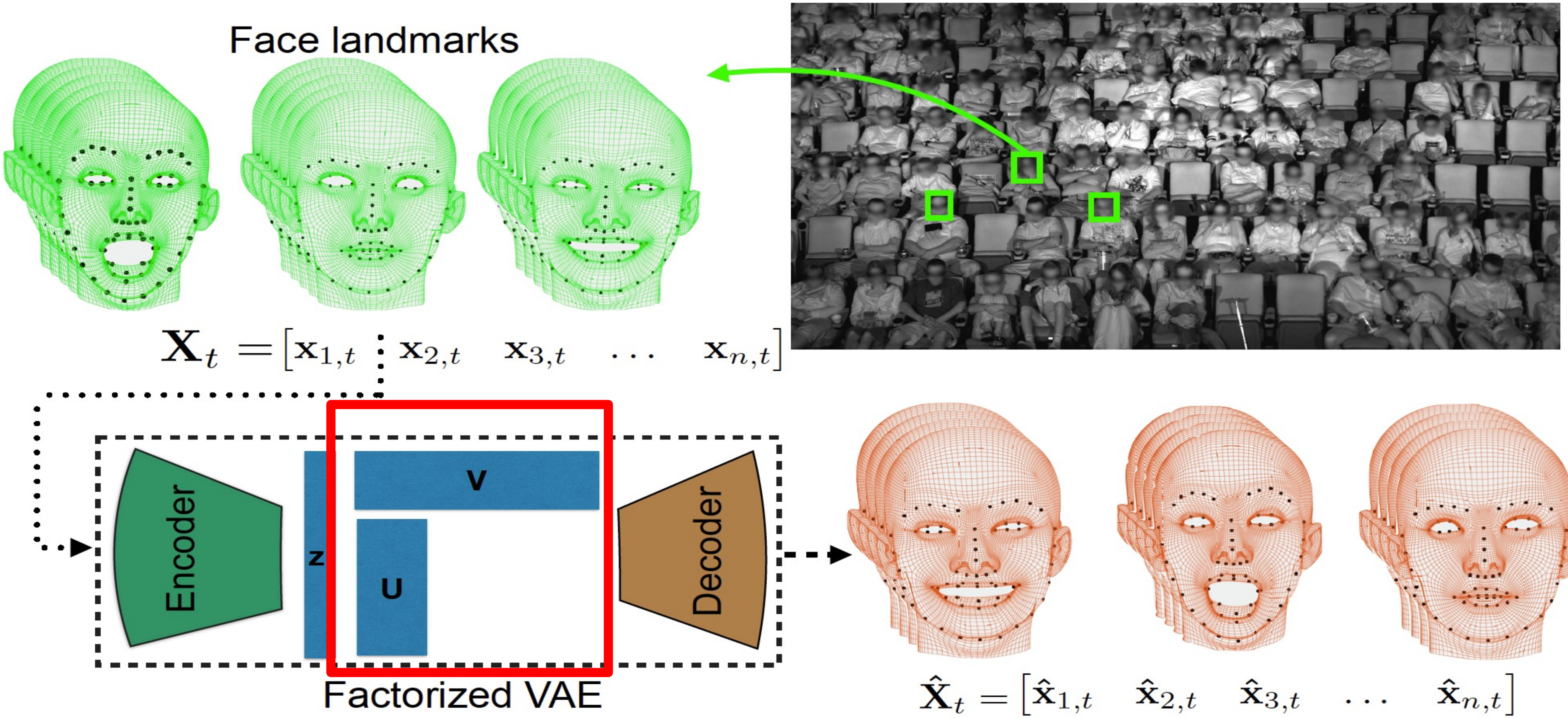
Related work – Recognizing the Facial Expression

1. McDuff et al. Using the smile that gauge a test audience's reaction to advertisements

2. Whitehill et al. Using the facial expressions to investigate student engagement in a classroom setting

3. Navarathna et al. Attempting to automate viewer sentiment analysis which measured the distribution of short-term correlations of audience motions to predict the overall rating of a movie

Background - Matrix Factorization



Background - Matrix Factorization

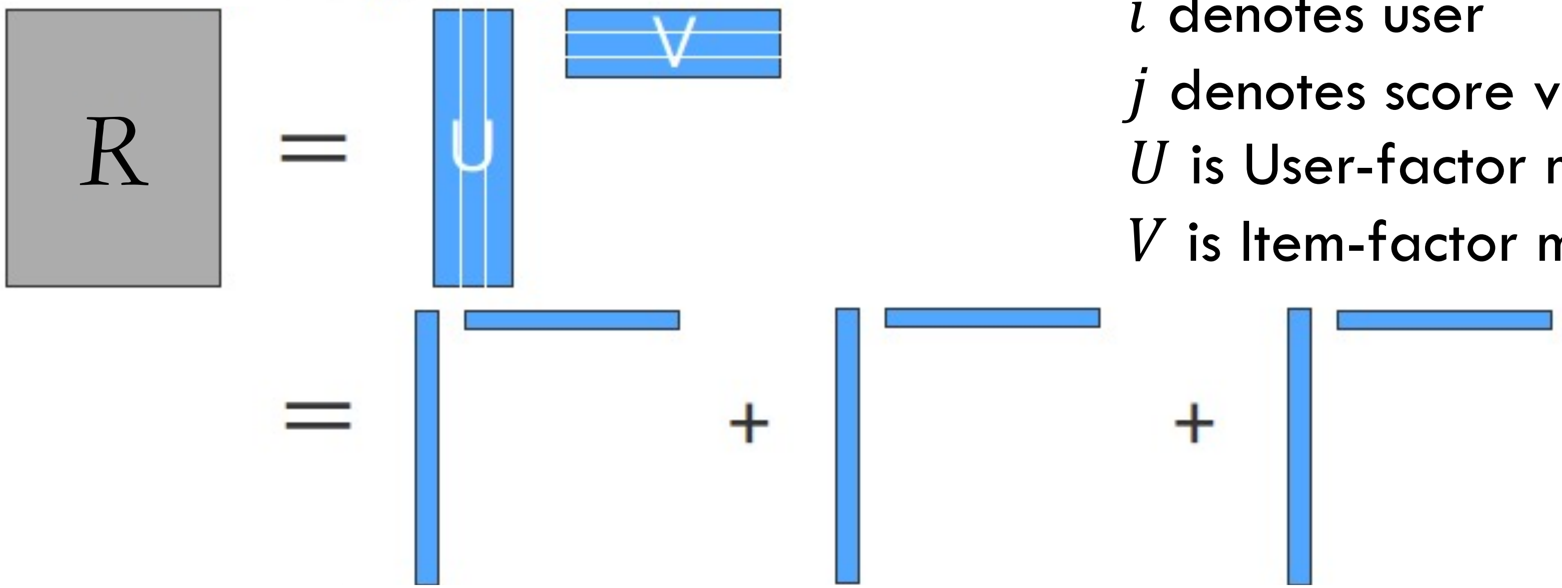
-Matrix factorization has three obvious uses

1. Processing the **dimensionality reduction**
2. Missing **data filling** (or "sparse data filling")
3. Mining the **implicit relationship**

Background - Matrix Factorization

$$R = UV^T$$

$R_{[m,n]}$
$U_{[m,k]}$
$V_{[n,k]}$



$r_{i,j}$ denotes user i rate the value for item j

i denotes user

j denotes score value of the goods

U is User-factor matrix

V is Item-factor matrix

When $k = rank(R) \ll \min(m,n)$, the above processing is Matrix Factorization

$k < rank(R)$, the process of matrix factorization as a low-rank approximation problem

Background - Matrix Factorization

least squares method

$R \approx UV^T$

$$\hat{r}_{ij} = \left(UV^T\right)_{ij} = \sum_{q=1}^{k} u_{iq} \cdot v_{jq}$$

$$\min J = \frac{1}{2} \sum_{(i,j) \in S} e_{ij}^2 = \frac{1}{2} \sum_{(i,j) \in S} \left(r_{ij} - \sum_{q=1}^{k} u_{iq} \cdot v_{jq} \right)^2$$

Background - CP decomposition

The CP decomposition factorizes a tensor into a sum of component rank-one tensors. For example, given a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, we wish to write it as

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \qquad (3.1)$$

where R is a positive integer and $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$ for $r = 1, \ldots, R$. Elementwise, (3.1) is written as

$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir}\, b_{jr}\, c_{kr} \text{ for } i = 1, \ldots, I,\ j = 1, \ldots, J,\ k = 1, \ldots, K.$$

P.S. the outer product

給定$m \times 1$ 列向量$\mathbf{u}$和$1 \times n$ 行向量$\mathbf{v}$，它們的外積$\mathbf{u} \otimes \mathbf{v}$被定義為$m \times n$矩陣$\mathbf{A}$

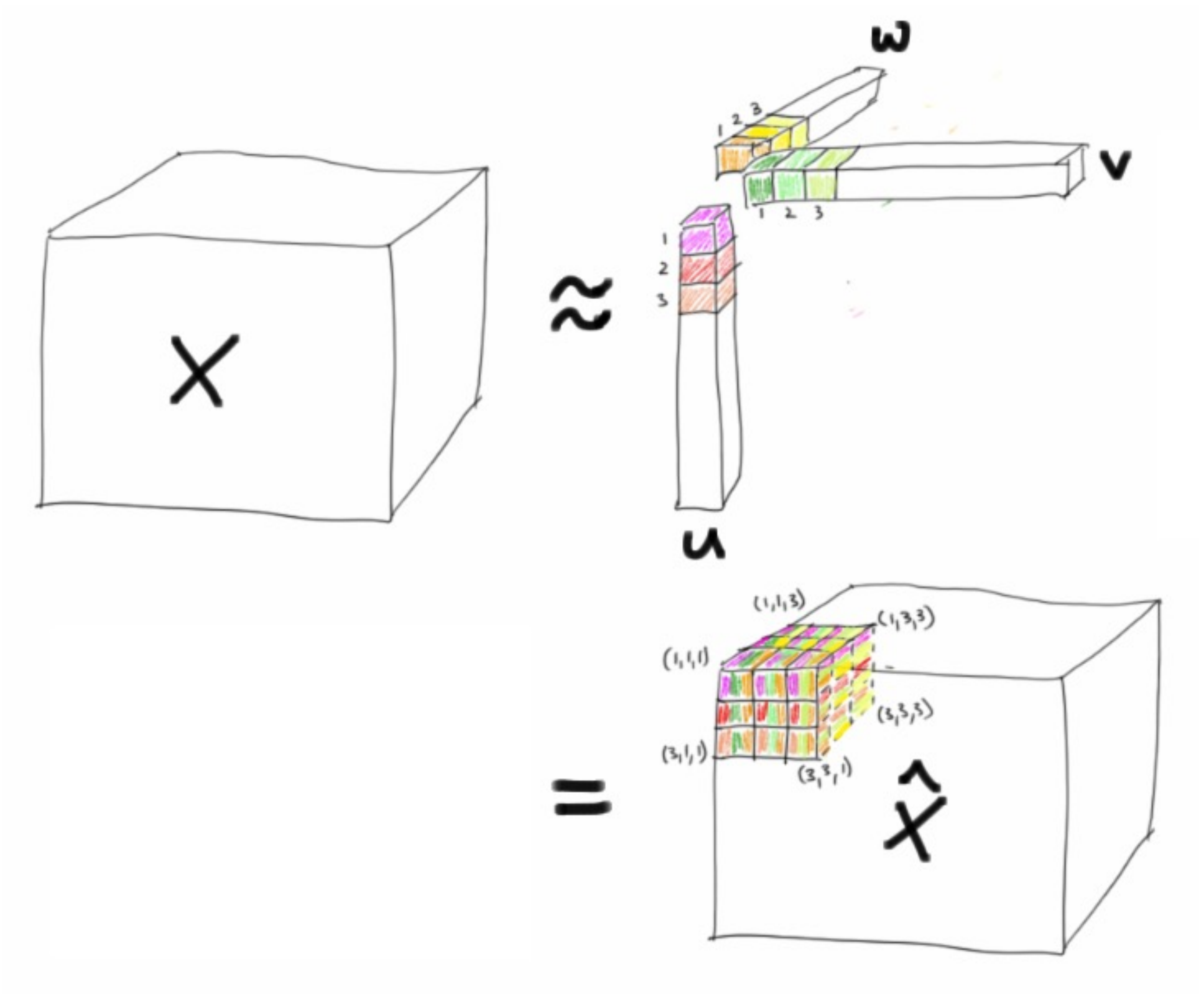
给定向量$\vec{a} = (1,2)^T$，向量$\vec{b} = (3,4)^T$，则$\vec{a} \circ \vec{b} = \vec{a}\vec{b}^T = \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$，运算符号“∘”表示外积。另给定向量$\vec{c} = (5,6,7)^T$，若$\mathcal{X} = \vec{a} \circ \vec{b} \circ \vec{c}$，则

$$\mathcal{X}(:,:,1) = \begin{bmatrix} 1\times3\times5 & 1\times4\times5 \\ 2\times3\times5 & 2\times4\times5 \end{bmatrix} = \begin{bmatrix} 15 & 20 \\ 30 & 40 \end{bmatrix},$$

$$\mathcal{X}(:,:,2) = \begin{bmatrix} 1\times3\times6 & 1\times4\times6 \\ 2\times3\times6 & 2\times4\times6 \end{bmatrix} = \begin{bmatrix} 18 & 24 \\ 36 & 48 \end{bmatrix},$$

$$\mathcal{X}(:,:,3) = \begin{bmatrix} 1\times3\times7 & 1\times4\times7 \\ 2\times3\times7 & 2\times4\times7 \end{bmatrix} = \begin{bmatrix} 21 & 28 \\ 42 & 56 \end{bmatrix},$$

其中，$\mathcal{X}$是一个三维数组（有三个索引），对于任意索引(i,j,k)上的值为$x_{ijk} = a_i \cdot b_j \cdot c_k, i=1,2, j=1,2, k=1,2,3$，在这里，向量$\vec{a}$, $\vec{b}$, $\vec{c}$的外积即可得到一个第三阶张量（third-order tensor），如图1所示。


X
≈
w
1 2 3
v
1 2 3
1
2
3
u
=
(1,1,3)
(1,3,3)
(1,1,1)
(3,3,3)
(3,1,1)
(3,3,1)
X̂

Problem

- To get potential expressions from the facial expressions of the audience watching the movie
  - ➢Observe the facial expressions of N audience in each movie (T frame)
  - ➢Each audience i at time t: 68 facial landmarks (x, y) => D=136

$$\textit{Facial expression of user i at time t} : x_{it} = \mathbb{R}^{136}$$

$$\text{Tensor } X \in \mathbb{R}^{N * T * D}$$

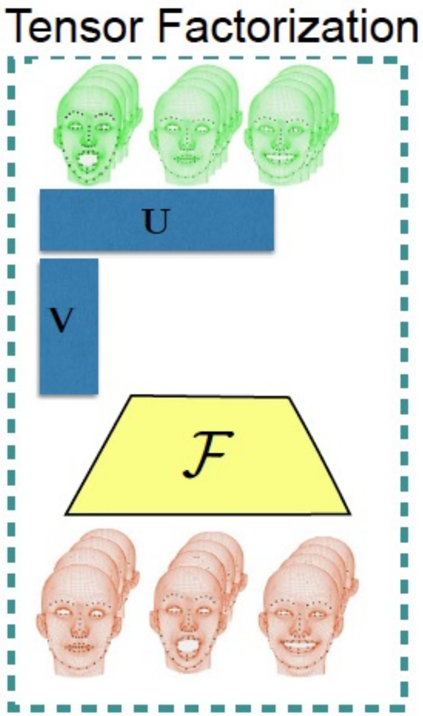
CP decomposition

To factorize X into matrices $U \in \mathbb{R}^{N*K}, V \in \mathbb{R}^{T*K}, F \in \mathbb{R}^{D*K}$

Each element of the original tensor is a linear combination of K latent factors from each matrix -> **Low expressiveness**

- U is User-factor matrix
- V is time series matrix
- F is latent space factor(facial expression) conversion matrix

$$x_{itd} = \sum_{k=1}^{K} U_{ik} V_{tk} F_{dk} \qquad \mathbf{x}_{it} = (\mathbf{U}_i \circ \mathbf{V}_t)\mathbf{F}^\mathsf{T}$$



Limitations of matrix factorization and tensor factorization

-Factorization approaches **rely** on the **data decomposing linearly**

∵ not suitable for **complex processes** (time-varying facial expressions)

Non-linear version of probabilistic tensor factorization

Assumption: Observation data X follows a Gaussian distribution

$$\log P(\mathbf{X}|\mathbf{U},\mathbf{V}) = \prod_{i=1}^{I}\prod_{j=1}^{J} N(x_{ij}|\mathbf{u}_i^T\mathbf{v}_j, 1)$$

Maximum likelihood estimation $\log P(\mathbf{X}|\mathbf{U},\mathbf{V}) = -\frac{IJ}{2}\log 2\pi - \sum_{i,j}\log\frac{1}{2}\|x_{i,j} - \mathbf{u}_i^T\mathbf{v}_j\|^2$

MAP estimation: The prior distribution of U and V is assumed to be Gaussian distribution $U, V \sim N(0, \lambda^{-1}I)$

$$\log P(\mathbf{X}|\mathbf{U},\mathbf{V})P(\mathbf{U})P(\mathbf{V}) = -\frac{IJ}{2}\log 2\pi - \sum_{i,j}\left\{\log\frac{1}{2}\|x_{i,j} - \mathbf{u}_i^T\mathbf{v}_j\|^2\right\} - \frac{\lambda}{2}(\|\mathbf{U}\|^2 + \|\mathbf{V}\|^2)$$

Non-linear version of probabilistic tensor factorization

$U_i \equiv$ (

$V_t \sim$ J

$x_{ij} \sim$

When the data set is large, the posterior is sharply peaked around its maximum mode. For inference, we can thus simply replace the latent variables by point estimates (MAP approximation). The Gaussian priors then simply become quadratic regularizers, and the objective to maximize is:

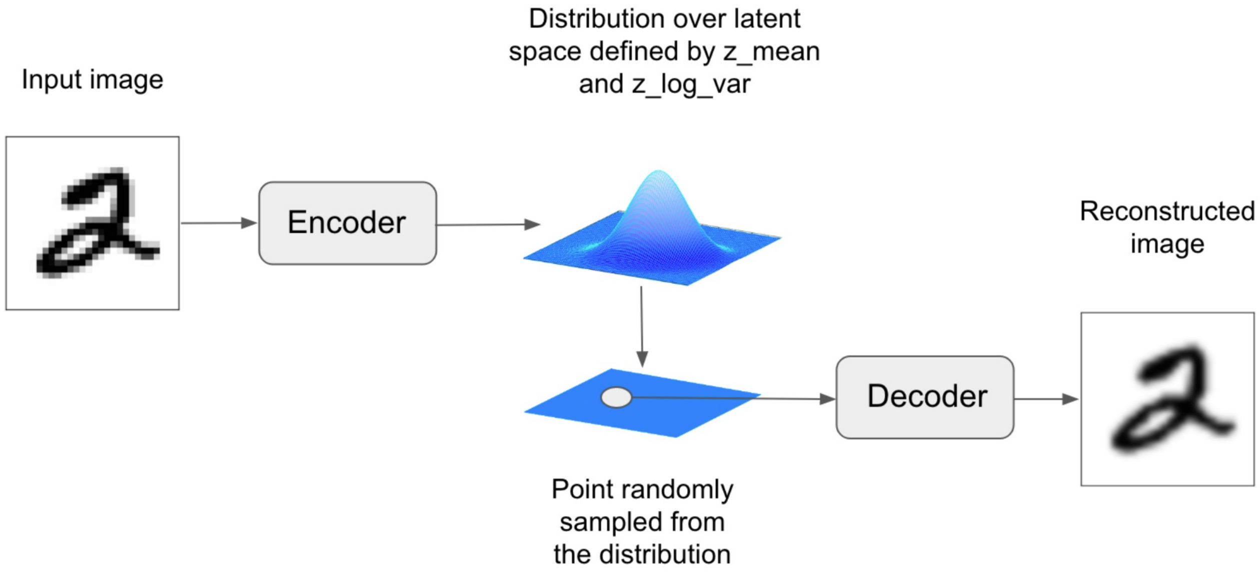
$$\mathcal{L}(\mathbf{U},\mathbf{V},\theta) = \log p_\theta(\mathbf{x}|\mathbf{U},\mathbf{V}) + \log p(\mathbf{U}) + \log p(\mathbf{V})$$
$$= \sum_{it} ||\mathbf{x}_{it} - \mathbf{f}_\theta(e^{\mathbf{u}_i} \circ \mathbf{V}_t)||_2^2 + \sum_i ||\mathbf{u}_i||_2^2 + \sum_t ||\mathbf{V}_t||_2^2. \quad (3)$$

1. doe ... xpression
2. evei

Proposed: Nonlinear tensor decomposition + VAE

Variational Autoencoders

- VAEs can learn a very compact encoding of the raw data
- statistical distribution: a mean and a variance



Variational Autoencoders

Assuming x is generated from the K-dimensional latent variable z

Interested posterior: $p_\theta(\mathbf{z}|\mathbf{x}) \;=\; p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})/p_\theta(\mathbf{x})$

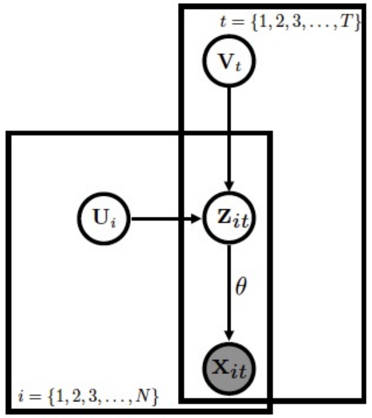
$$\forall_{i,t} : \mathbf{z}_{it} \sim \mathcal{N}(0, \mathbf{I}), \mathbf{x}_{it} \sim \mathcal{N}(\mathbf{f}_\theta(\mathbf{z}_{it}), \mathbf{I})$$

$$q_\lambda(\mathbf{z}|\mathbf{x}) \;=\; \prod_{i,t} \mathcal{N}(\mu_\lambda(\mathbf{x}_{it}), \mathbf{\Sigma}_\lambda(\mathbf{x}_{it}))$$

Maximum likelihood estimation: $\mathcal{L}(\theta, \lambda) = \mathbb{E}_q[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL(q_\lambda(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).$

Reconstruction works well with VAE

Factorized Variational Autoencoders (FVAE)



Graphical model of Factorized VAE

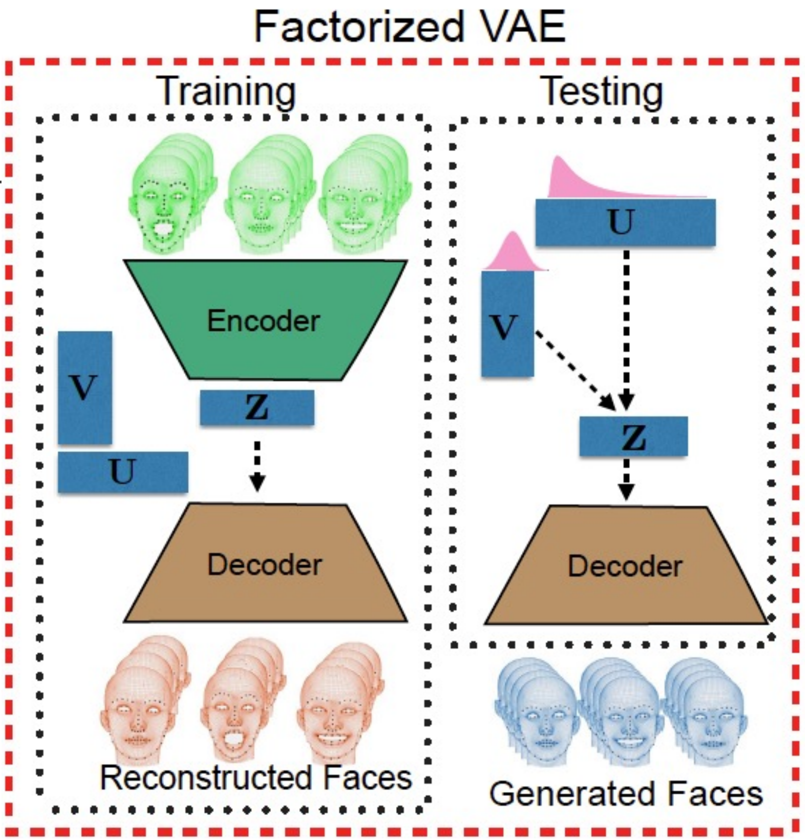
- VAE: local latent variables (User i info. at time t) ?
- FVAE: local + global latent variables (Other time and user information) ?
  1. Jointly learns a non-linear encoding z_{it} for each face reaction x_{it}
  2. Jointly carries out a factorization in z

$$\mathbf{z}_{it} \sim \mathcal{N}(\mathbf{U}_i \circ \mathbf{V}_t, \mathbf{I})$$

$$\mathbf{x}_{it} \sim \mathcal{N}(\mathbf{f}_\theta(\mathbf{z}_{it}), \mathbf{I})$$

$$\mathcal{L}(\mathbf{U}, \mathbf{V}, \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| \mathcal{N}(\mathbf{U} \circ \mathbf{V}, \mathbf{I})) + \log p(\mathbf{U}) + \log p(\mathbf{V})$$

Prediction(filling) and generation by FVAE



Matrix prediction - Facial expression of user i at time t

1. $U_i \circ V_t$ get the corresponding latent factor z_{it}

2. Predicts the output by pushing z_{it} into the decoder

If user i **not observed** at time t

1. Accomplished by drawing U_i and V_t from the **prior distributions**

2. Predicts the output by pushing the Hadamard product into the decoder

Movie audience dataset

- To capture the audience by an IR camera during the movie screening
  - 2750x2200 pixels, 12 FPS
    1. Rear seat face: 15x25 pixels (last three rows of the theatre)
    2. Front seat face: 40x55 pixels

- Screening time: 90-140 minutes
- Audience: 30-120

Movie	# Sessions	Time [min]	Genre
Ant Man	29	117	Action
Big Hero 6	11	102	Animation
Bridge of Spies	09	141	Drama
Inside Out	28	94	Animation
Star Wars: The Force Awakens	25	135	Action
The Finest Hours	06	115	Drama
The Good Dinosaur	13	93	Animation
The Jungle Book	17	105	Action
Zootopia	15	105	Animation

Table 1. **Movies.** The number of viewings of each film.

Pre-processing of movie audience datasets

Face detection

- Max-Margin Object Detection to learn Histgram of Gradients Apply (using DLib)
- 800 training images
- 10,000 frames for validation

	Precision	Recall
前方座席	99.5%	92.2%
後方座席	98.1%	71.1%

Landmark Detection

- Ensemble of regression trees(ERT)

Method	Dataset		
	LFPW	HELEN	IBUG
RCPR	0.035	0.065	–
SDM	0.035	0.059	0.075
ESR	**0.034**	0.059	0.075
ERT	0.038	**0.049**	**0.064**

Pre-processing of movie audience datasets

Front face

- 3D modeling based on 68 detected landmarks
- calculate the 3D rotation matrix R -> generate a frontalized view of the 68 landmarks
- 3D face mesh by Face Warehouse

$$\mathbf{x}_{it} = [x_{1,t}, y_{1,t}, \ldots, x_{68,t}, y_{68,t}]$$

Experiments

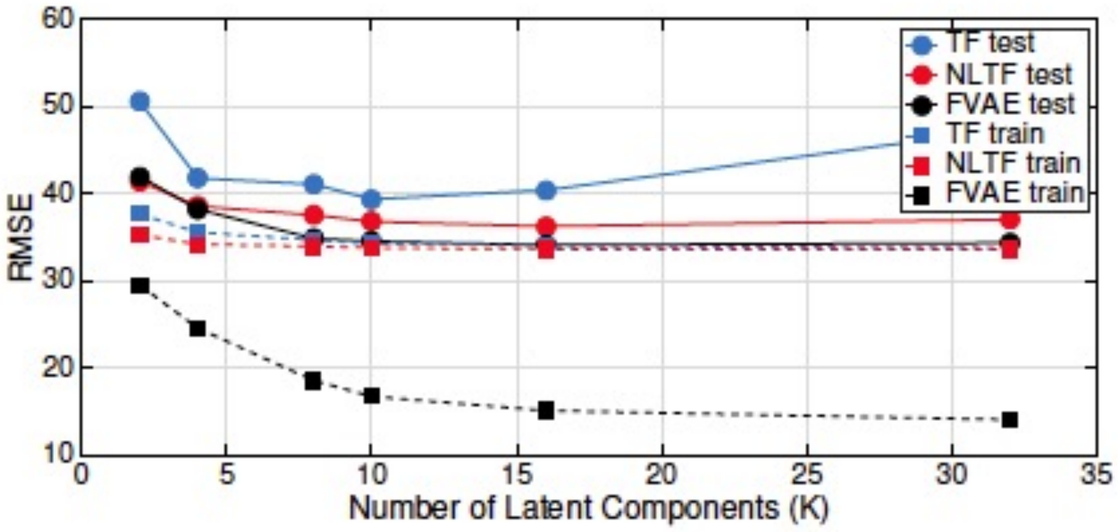
- For each movie : N x T x D tensors (audience x frame x landmark)
- Missing data rate ≈ 13% (overall)

Finally => 9 movies, 3179 audience members(16 million total face landmarks)

- NN:
  - ➢mini-batch size: 10
  - ➢Adam
  - ➢3 stacked fully connected layers with ReLU

Matrix Completion for Missing Data

- Divide each user's observations into 5 : 1(training set : test set)
- Use 4 movies to determine the value of K
- Inside Out, The Jungle Book, The Good Dinosaur, Zootopia
- K = {2; 4; 8; 10; 16; 32}
- K = 16 => FVAE is the best

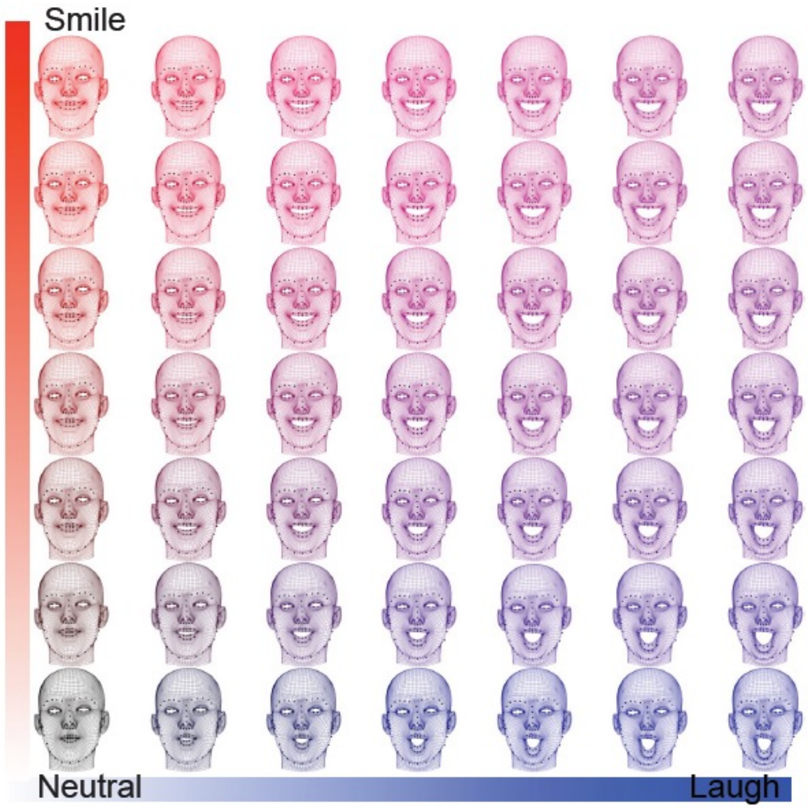


Performance

Movie	Reconstruction MSE			Prediction MSE		
	TF	NLTF	FVAE	TF	NLTF	FVAE
Ant Man	1287.7	1261.5	**292.7**	1897.7	1349.1	**1325.7**
Big Hero 6	1394.4	1371.7	**275.3**	1505.6	1557.3	**1424.9**
Bridge of Spies	942.8	910.9	**184.3**	1288.3	1062.9	**960.0**
The Good Dinosaur	1156.5	1132.1	**275.4**	1328.4	1416.1	**1244.7**
Inside Out	1214.7	1132.7	**262.1**	1977.9	1240.3	**1161.4**
The Jungle Book	1150.0	1115.3	**186.4**	1622.1	1200.2	**1099.8**
Star Wars: TFA	1080.7	1047.4	**201.4**	1519.0	1192.2	**1085.8**
The Finest Hours	1015.3	962.9	**223.5**	1101.4	1114.0	**1038.8**
Zootopia	1181.0	1153.8	**189.9**	1277.4	1407.5	**1153.7**
Average	1158.1	1120.9	**232.3**	1502.0	1282.2	**1166.1**

Table 3. **Performance.** The performance of all three models with their best K values. FVAEs acheive the lowest training and testing error for all movies.

Visualization of latent factors

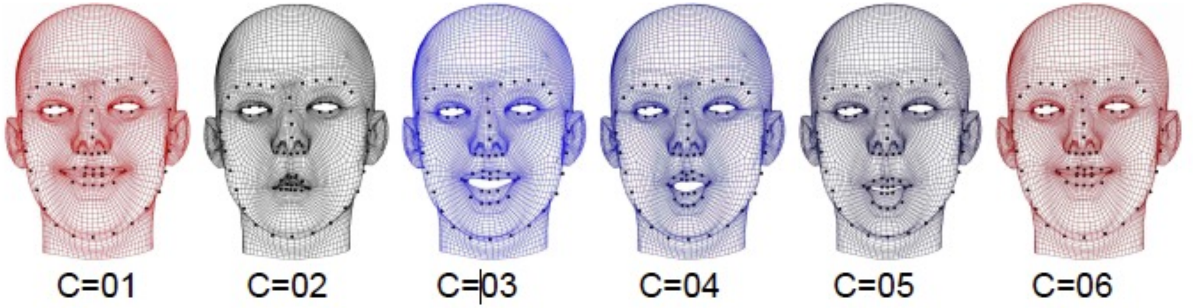


Concept Vector ->
learn concepts of smiling and laughing

well-structured latent spaces

Analyzing Group behaviour

- clustering the rows of U
- exemplar faces for a humorous moment in the movie



01 and 06 correspond to smiling

03, 04 and 05 correspond to laughing

02 has no laughter or smile