Electrical Engineering & Computer Science

COLLEGE OF ENGINEERING

Some Thoughts and New Designs of Recurrent and Convolutional Architectures

Fuxin Li

AUGUST 1ST, 2018

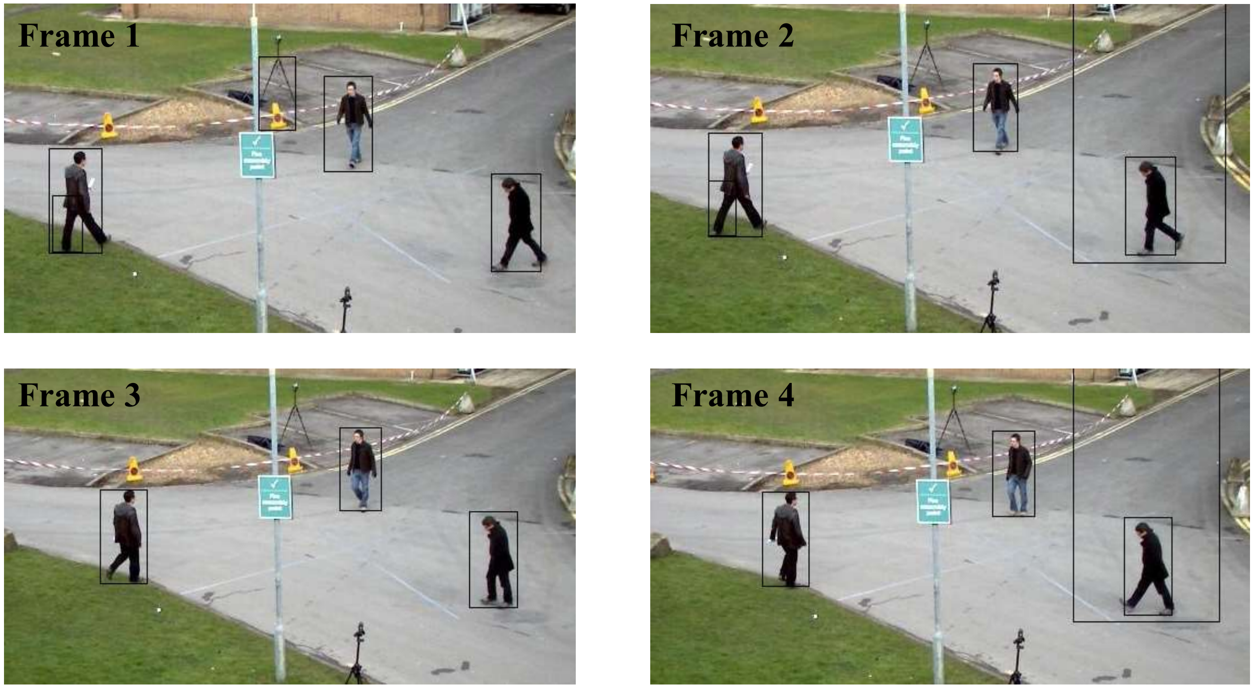
Today's Talk

- Multi-Target Tracking with bilinear LSTM
  - Novel LSTM model coming from studies on tracking
- Understanding more about CNNs
  - Generalization Theory based on Gaussian Complexity and Redesigns
  - XNN: Explaining CNN to human

Today's Talk

- **Multi-Target Tracking with bilinear LSTM**
  - Novel LSTM model coming from studies on tracking
- Understanding more about CNNs
  - Generalization Theory based on Gaussian Complexity and Redesigns
  - XNN: Explaining CNN to human

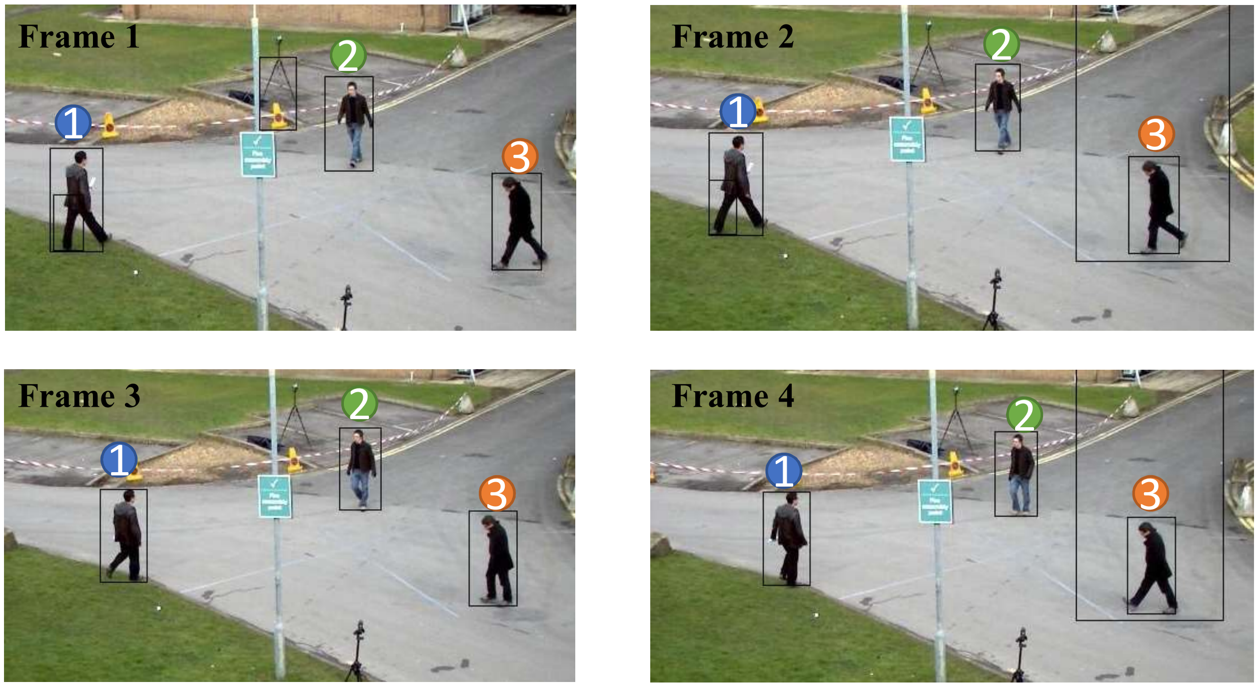
Multi-Target Tracking by Detection



Link person detections in each frame into tracks

Search space reduced by using a person detector

Multi-Target Tracking by Detection



Link person detections in each frame into tracks

Search space reduced by using a person detector

Multi-Target Tracking Illustration

The Essence of Tracking

Appearance Cues

- People (targets) look different, they wear different clothes

Motion Cues

- People (targets) move in a smooth/piecewise-smooth manner

Appearance Cues

Identity (ID) Switch!

Multiple Appearances + Motion

Successful tracking algorithms combine appearance and motion cues

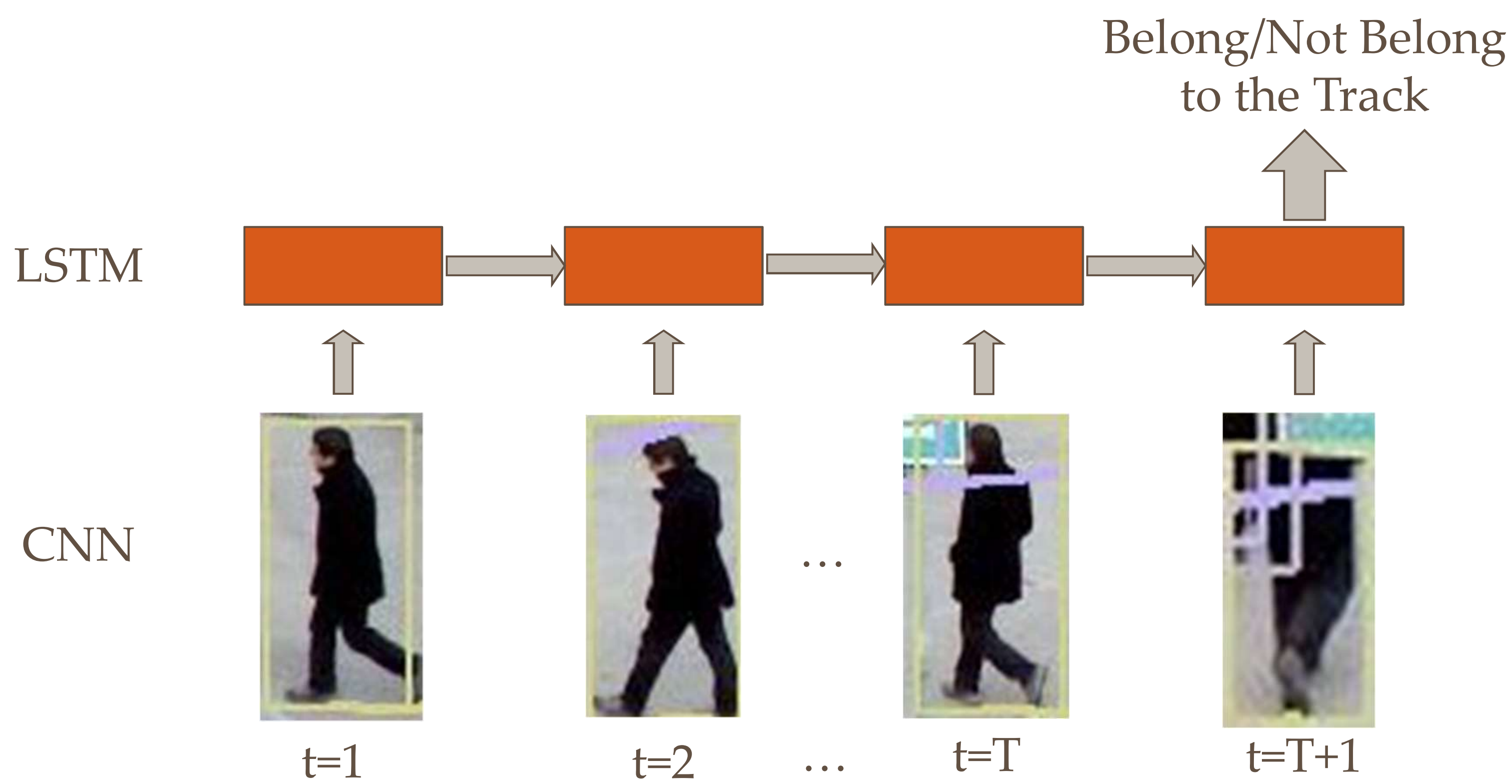
Each object can have many appearances, this need to be handled too

Goal: End-to-End Training

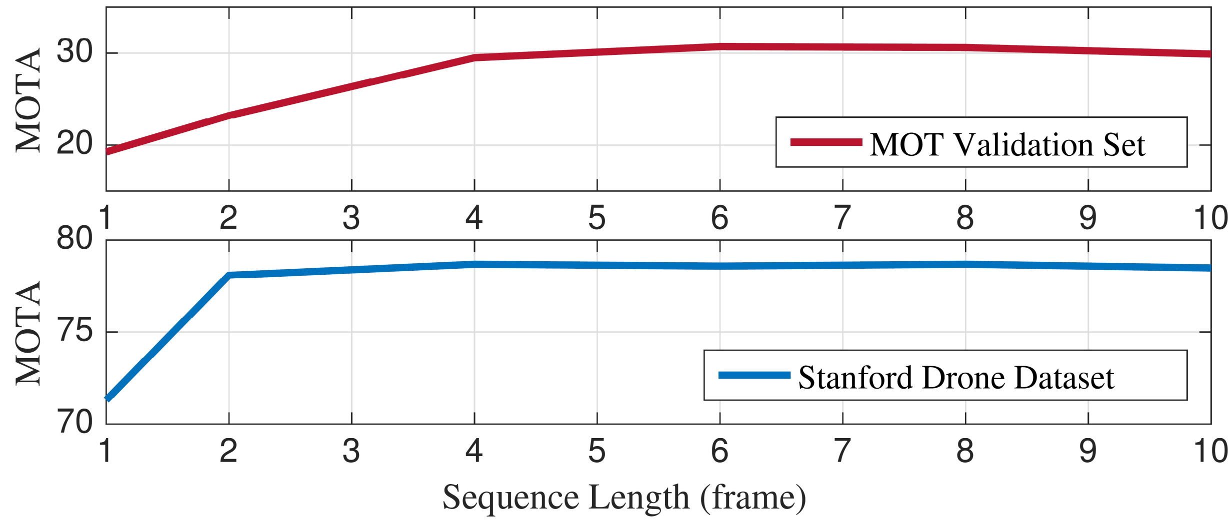
- Interestingly, tracking is rarely trained end-to-end
  - There is often an appearance model that is updated online
    - e.g. MHT-DAM [Kim et al. 2015], STAM [Chu et al. 2017]
  - And then a motion model that is separately updated
    - Most likely, a heuristic motion model (linear, constant velocity)
    - Or Kalman filter (e.g. [Kim et al. 2015])
  - And then post-processing
- There should be a few benefits for end-to-end training
  - Using more complex nonlinear motion models
  - Have the motion and appearance models better work together

Previous attempts on using a recurrent model

- A standard approach to train on a video sequence would be a convolution + recurrent model
  - Tried a couple of times (Milan et al. 2017, Sadeghian et al. 2017) with some success



Interesting Phenomenon on a Recurrent Model



(b)

Using longer sequences to train the
LSTM does not seem to bring any benefit!

(image cf. Sadeghian et al. 2017)

Reflect about this Longer Training Sequence issue:

Appearance Part

Multiple Appearances!

Longer sequence in training should be beneficial

Motion Part

Single Motion Trajectory!

Longer sequence may not be beneficial

Longer Training Sequence

Appearance Part

Multiple Appearances!

Longer sequence in training
should be beneficial

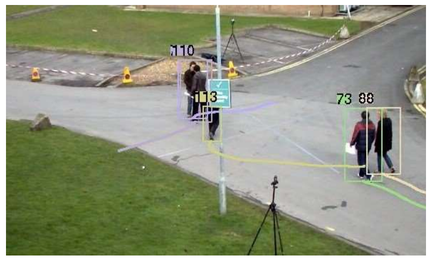
Hypothesis:

LSTM in multi-target tracking may **not** be modeling multiple appearances properly

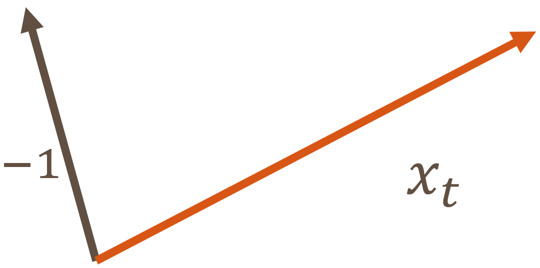
The Dilemma of the LSTM Memory

Memory

x_t

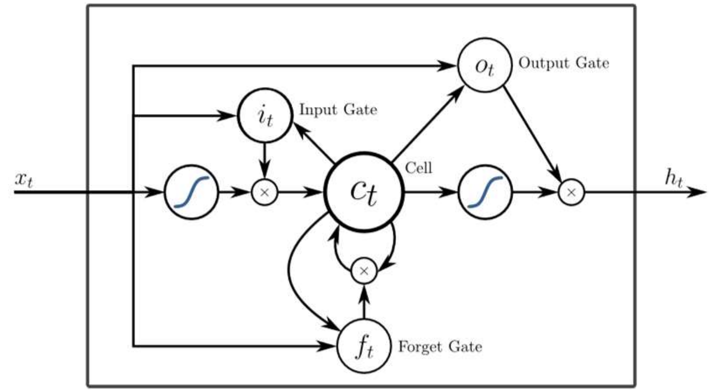


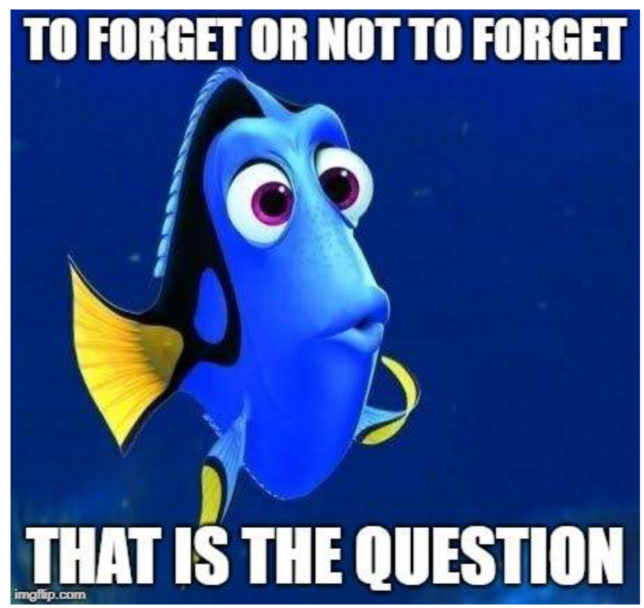
c_{t-1}



x_t

LSTM





Why is there not an option of:
put the memory aside?

In the Quest for a New LSTM

- We check a non-deep appearance modeling approach
- Recursive least squares
  - Used in several work, e.g. DCF/KCF (Henriques et al. 2012), SPT (Li et al. 2013), MHT-DAM (Kim et al. 2015)
  - As well as being a classic tracking approach in robotics
  - Global optimal online appearance modeling framework
  - Appearance model is a classifier/regressor
  - Capable of modeling multiple appearances

How does it work

- Tracker is a regressor
  - Appearance model: classifies any new appearance to object/not object

$$w_t = \arg\min_{w} ||w^\top x_{0:t} - y_{0:t}||^2 + \lambda||w||^2$$

Appearance Features (e.g. CNN) from Positive and Negative Examples

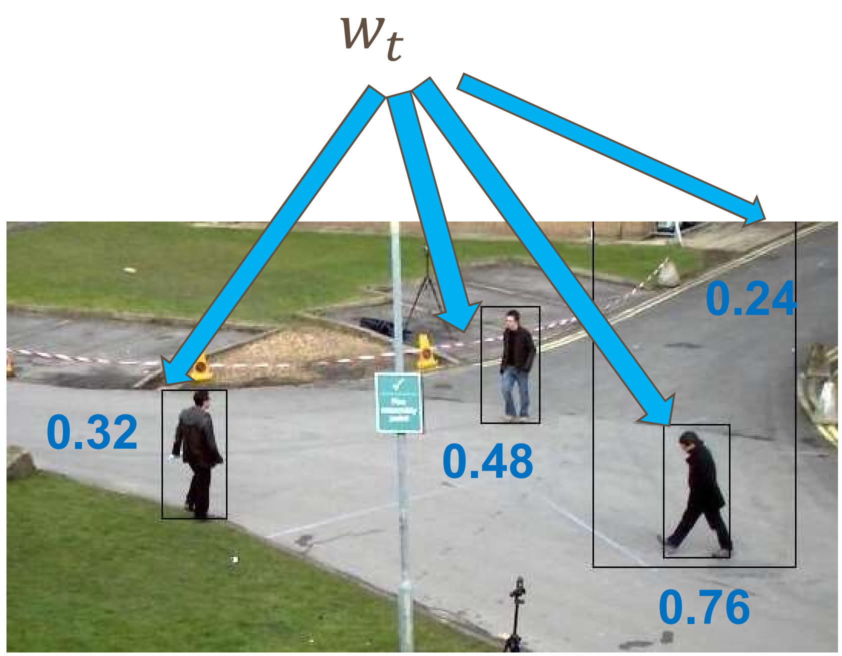
(Soft) Labels
e.g. Jaccard index

Negative (label = 0)

Positive (label = 1)

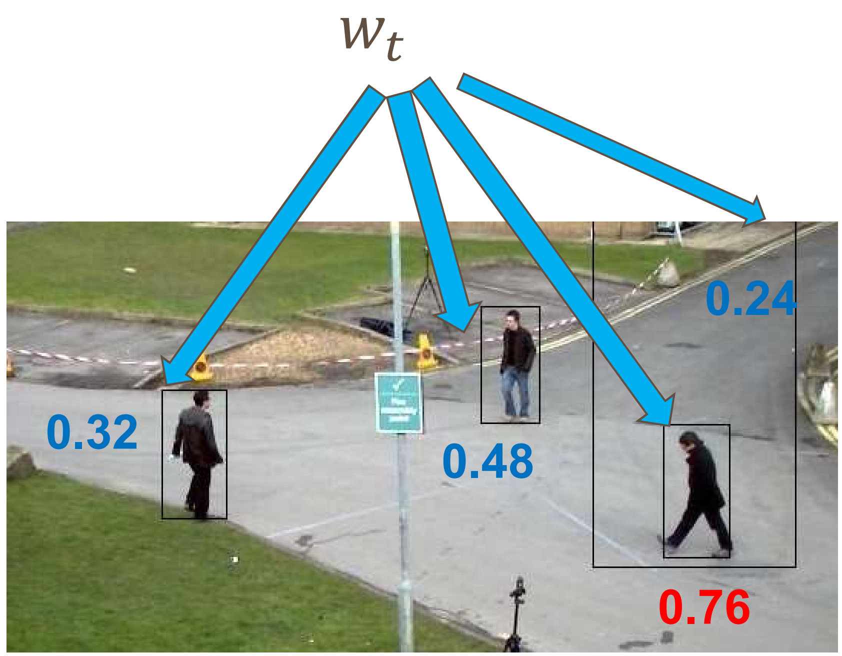
Testing and recursive training

- Test model on all detections:



Testing and recursive training

- Decide which one is matched to the track



Testing and recursive training

- Generate training examples for time t+1
- Solve for w_{t+1}

$$w_{t+1} = \arg\min_{w} ||w^{\top} x_{0:t+1} - y_{0:t+1}||^2 + \lambda ||w||^2$$

(Some of the) good stuff with least squares

Solution of w:

$$\boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} = (\boldsymbol{H} + \lambda \boldsymbol{I})^{-1} \boldsymbol{c}$$

$$\boldsymbol{H}_k = \boldsymbol{X}_{(1:k-1)}^\top \boldsymbol{X}_{(1:k-1)} + \mathbf{X}_{(k)}^\top \mathbf{X}_{(k)}$$

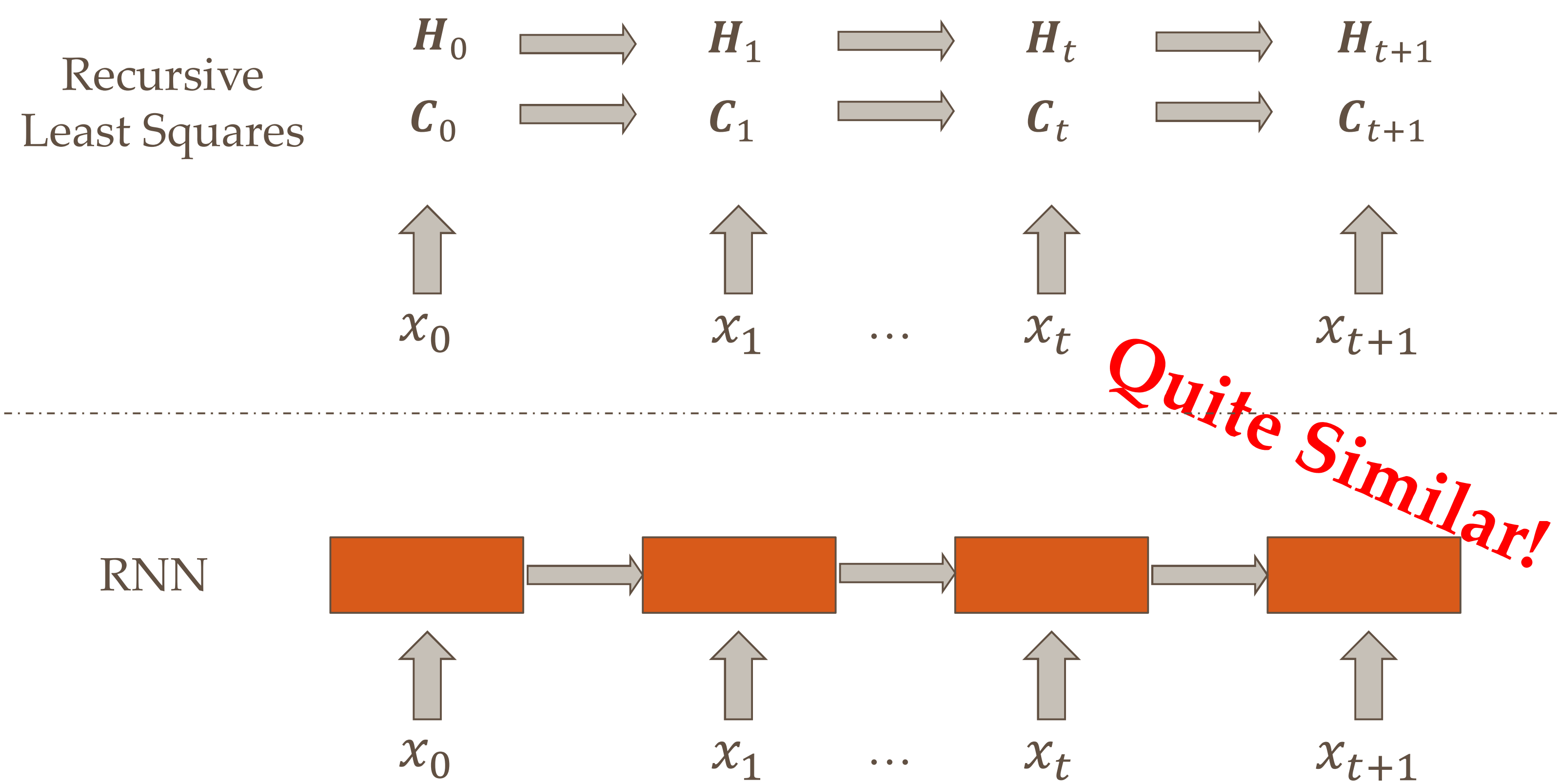
$$\boldsymbol{c}_k = \mathbf{X}_{(1:k-1)}^\top \mathbf{y}_{(1:k-1)} + \mathbf{X}_{(k)}^\top \mathbf{y}_{(k)}$$

1) Each frame is separable!
2) Inversion does not depend on number of targets (tracks)

- In DCF/KCF (Henriquez et al. 2012, 2014), more computational savings with Fourier domain transformations
- In MHT-DAM (Kim et al. 2015), this is used to learn a different appearance model for each branch in an MHT tree

The "Recurrent Model" Version of Least Squares

Problem: Storing $d \times d$ matrix H in RNN is too memory-consuming



Low-rank Approximation

- Go back to the solution formula

$$\boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} = (\boldsymbol{H} + \boldsymbol{\lambda} \boldsymbol{I})^{-1} \boldsymbol{c}$$

$$\boldsymbol{w}^\top \boldsymbol{x} \approx \sum_{i=1}^{r} \boldsymbol{c}^\top \boldsymbol{h}_i \boldsymbol{h}_i^\top \boldsymbol{x} = \sum_{i=1}^{r} \mu_i \boldsymbol{h}_i^\top \boldsymbol{x}$$

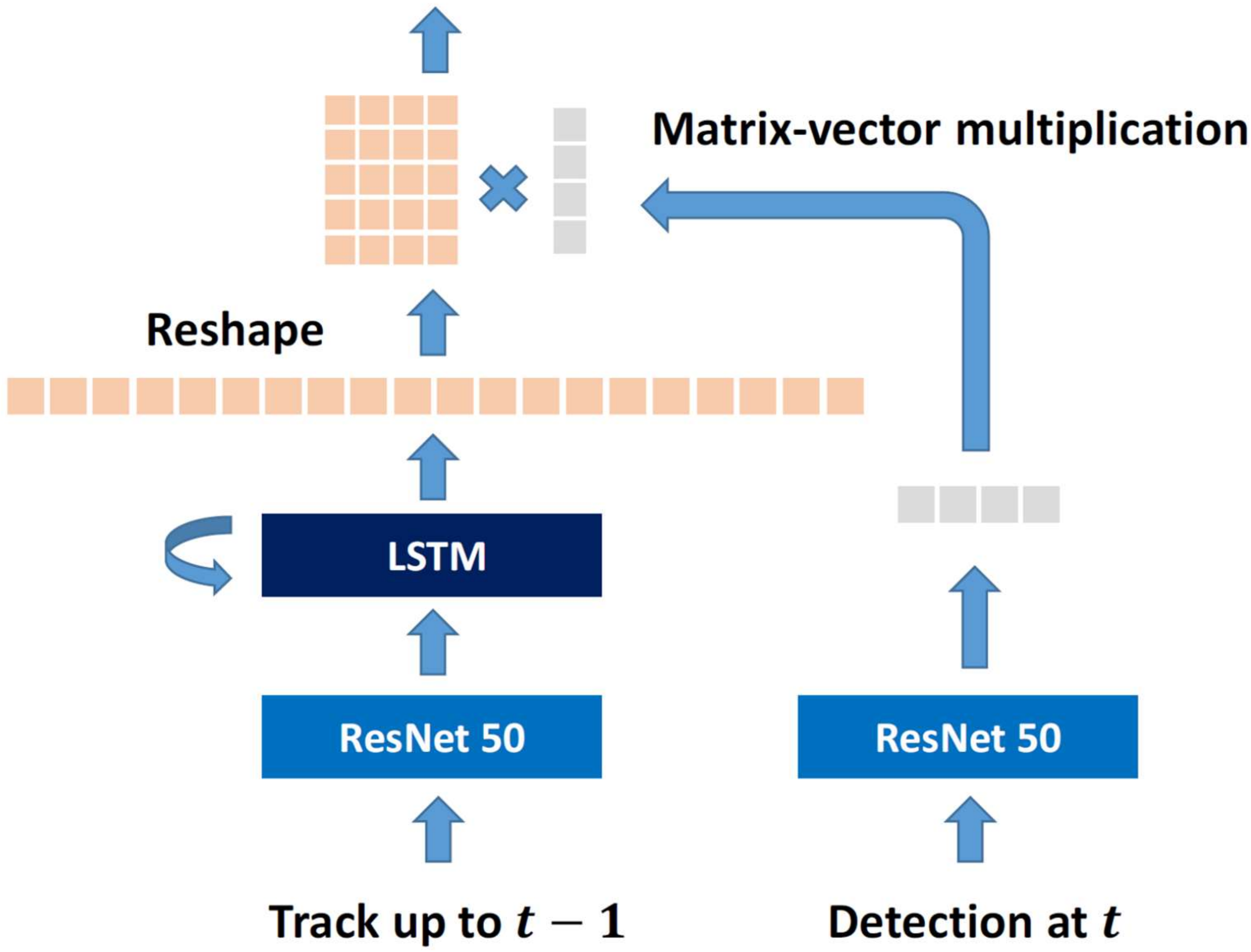
Track-specific layer

Memory

Feature input (e.g. CNN)

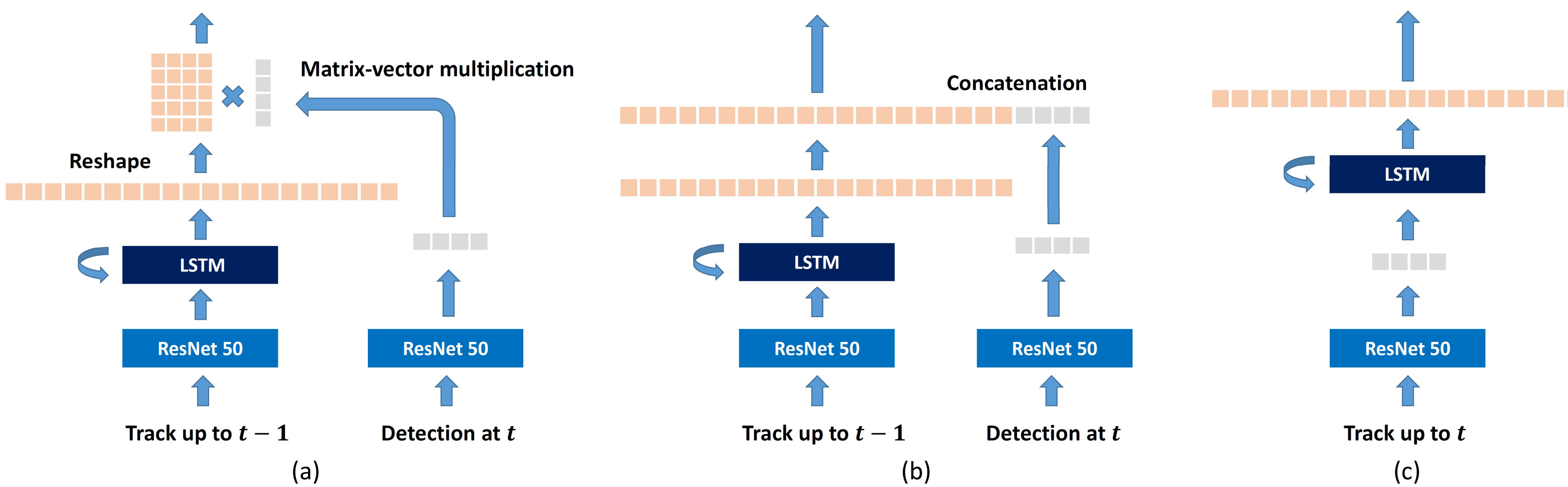
The difference between this and a normal RNN/LSTM update?

Bilinear LSTM



Bilinear LSTM Model Study

- Tried 3 models for
  - Appearance LSTM
  - Motion LSTM



Bilinear LSTM

Concatenate Memory and Input

Normal LSTM

Experiment Details

- MOT-17 dataset (without 17-09 and 17-10) + ETH + PETS + TUD + TownCentre + KITTI16 + KITTI19 as training
- MOT-17-09, MOT-17-10 as validation
- Faster R-CNN detector with ResNet 50 head
- Public Detections

- Detailed model architecture for appearance:

Soft-max			
Matrix-vector Multiplication-relu 8			
Reshape	8×256	Reshape	256×1
LSTM	2048		
FC-relu	256	FC-relu	256
ResNet-50	2048	ResNet50	2048
Input at $t-1$	$128 \times 64 \times 3$	Input at t	$128 \times 64 \times 3$

(a)

Soft-max			
FC-relu 512			
Concatenation $2048 + 256$			
LSTM	2048		
FC-relu	256	FC-relu	256
ResNet-50	2048	ResNet50	2048
Input at $t-1$	$128 \times 64 \times 3$	Input at t	$128 \times 64 \times 3$

(b)

Soft-max	
FC-relu	512
LSTM	2048
FC-relu	256
ResNet-50	2048
Input at t	$128 \times 64 \times 3$

(c)

Comparison between different appearance LSTMs

- Bilinear LSTM significantly better than other LSTM variants
  - ID switches almost halved
- Longer training sequence make a difference
  - The best sequence length is now between 20-40 frames

LSTM	MOTA	IDF1	IDS
Bilinear	**52.33**	**59.07**	**233**
Baseline1	50.43	51.28	412
Baseline2	50.97	51.49	462

State dim.	MOTA	IDF1	IDS
512	52.14	56.66	283
1024	52.36	55.85	**222**
2048	52.33	**59.07**	233

$N_{\max}$	MOTA	IDF1	IDS
10	51.96	54.36	271
20	52.27	58.38	228
40	52.33	**59.07**	233
80	52.32	57.21	239
160	52.41	55.19	**222**

Table 4: Ablation Study for Appearance Gating Networks. Baseline1 and Baseline2 are the networks shown in Table 2 (b) and (c) resepectively. **(Left)** State dim. = 2048, $N_{\max} = 40$ **(Middle)** LSTM: Bilinear, $N_{\max} = 40$, **(Right)** LSTM: Bilinear, State dim. = 2048

Comparison between different motion LSTMs

- Bilinear LSTM does not work as well as regular LSTM in motion LSTM
  - Maybe the single modality of motion LSTM makes regular LSTM more suitable

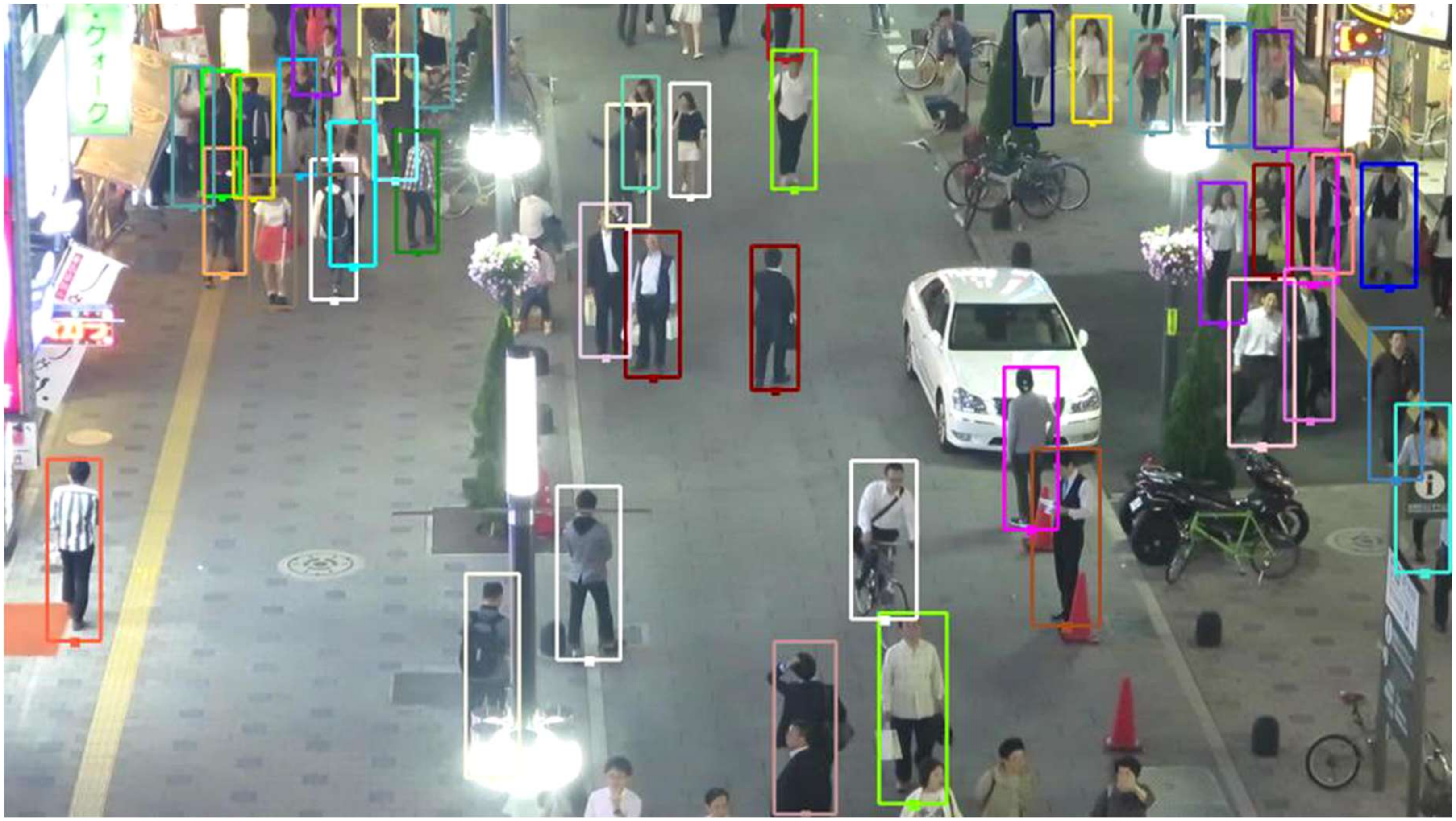
LSTM	MOTA	IDF1	IDS
Bilinear	39.68	41.22	226
Baseline1	38.90	19.38	449
Baseline2	40.14	**44.11**	**106**

State dim.	MOTA	IDF1	IDS
64	40.14	44.11	106
128	40.16	44.26	**97**
256	40.15	44.48	103

$N_{\max}$	MOTA	IDF1	IDS
20	39.76	28.50	206
40	40.14	44.11	106
80	40.15	**45.29**	104
160	40.20	**45.15**	**91**

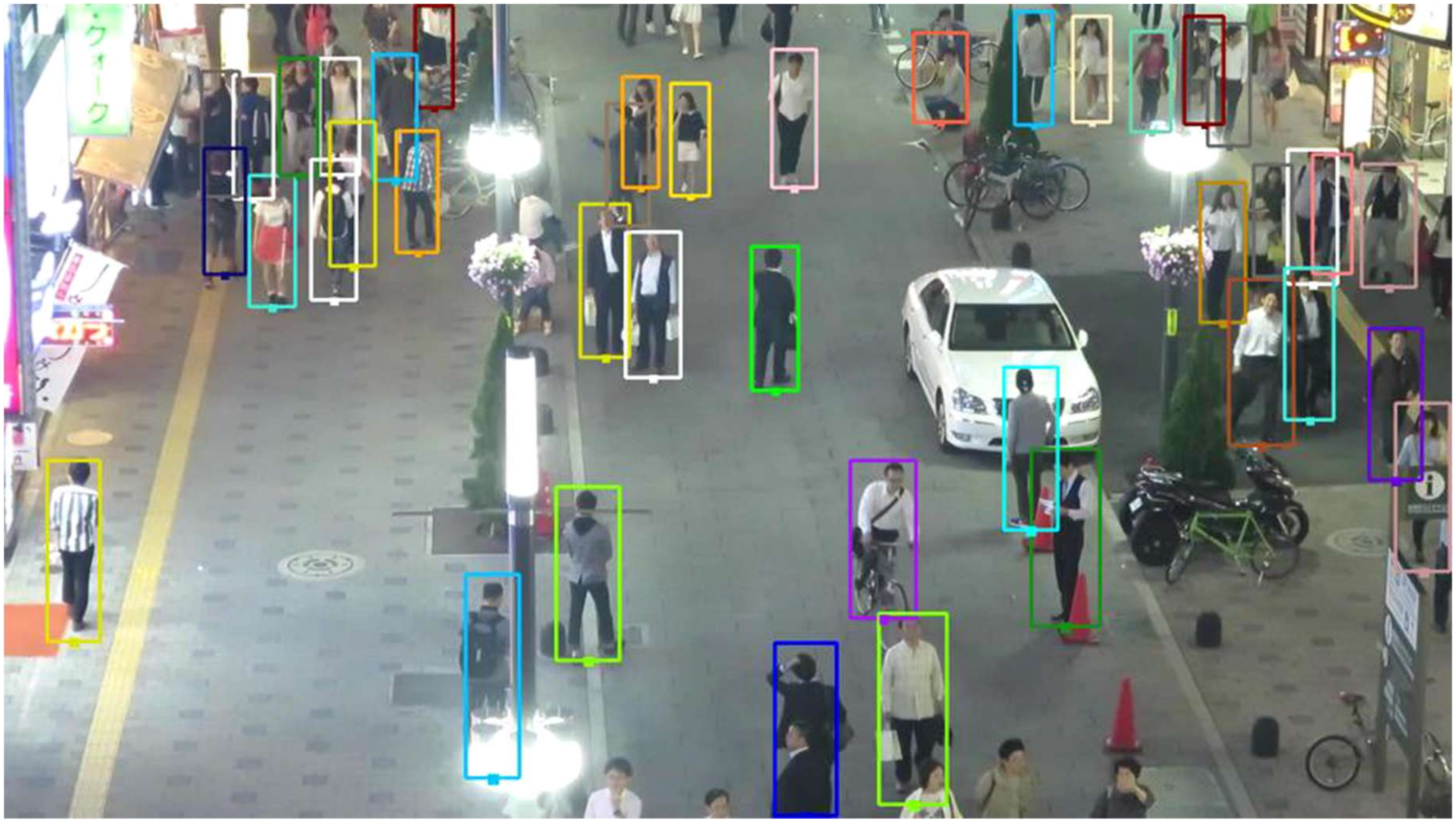
Table 2: Ablation Study for Motion Gating Networks **(Left)** State dim. = 64, $N_{\max}$ = 40 **(Middle)** LSTM:Baseline2, $N_{\max} = 40$, **(Right)** LSTM:Baseline2, State dim. = 64

Final MOT-17 Result Videos



MHT-DAM (Kim et al. 2015)

Final MOT-17 Result Videos



MHT-bLSTM

C. Kim, FL, J. Rehg. ECCV 2018

Final MOT Results

- Showing all the top non-anonymous results on MOT-17 (as of 7/31/18), sorted by IDF1:

Tracker	Avg Rank	MOTA	↑IDF1	MT	ML	FP	FN	ID Sw.	Frag	Hz	Detector
eHAF17 2.	13.5	51.8 ±13.2	54.7	23.4%	37.9%	33,212	236,772	1,834 (31.6)	2,739 (47.2)	0.7	Public
	TCSVT-02141-2018										
jCC 3.	14.6	51.2 ±14.5	54.5	20.9%	37.0%	25,937	247,822	1,802 (32.1)	2,984 (53.2)	1.8	Public
	M. Keuper, S. Tang, Y. Zhongjie, B. Andres, T. Brox, B. Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. In arXiv preprint arXiv:1607.06317, 2016.										
MOTDT17 7.	15.8	50.9 ±11.9	52.7	17.5%	35.7%	24,069	250,768	2,474 (44.5)	5,317 (95.7)	18.3	Public
	C. Long, A. Haizhou, Z. Zijie, S. Chong. Real-time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-identification. In ICME, 2018.										
MHT_bLSTM 9. NEW	20.5	47.5 ±12.6	51.9	18.2%	41.7%	25,981	268,042	2,069 (39.4)	3,124 (59.5)	1.9	Public
	C. Kim, F. Li, J. Rehg. Multi-object Tracking with Neural Gating Using Bilinear LSTM. In ECCV, 2018.										
EDMT17 12.	16.4	50.0 ±13.9	51.3	21.6%	36.3%	32,279	247,297	2,264 (40.3)	3,260 (58.0)	0.6	Public
	J. Chen, H. Sheng, Y. Zhang, Z. Xiong. Enhancing Detection Model for Multiple Hypothesis Tracking. In BMTT-PETS CVPRw, 2017.										
PHD_GSDL17 17.	22.8	48.0 ±13.6	49.6	17.1%	35.6%	23,199	265,954	3,998 (75.6)	8,886 (168.1)	6.7	Public
	Z. Fu, P. Feng, F. Angelini, J. Chambers, S. Naqvi. Particle PHD Filter based Multiple Human Tracking using Online Group-Structured Dictionary Learning. In IEEE Access, 2018.										
FWT 26.	16.4	51.3 ±13.1	47.6	21.4%	35.2%	24,101	247,921	2,648 (47.2)	4,279 (76.3)	0.2	Public
	R. Henschel, L. Leal-Taixé, D. Cremers, B. Rosenhahn. Fusion of Head and Full-Body Detectors for Multi-Object Tracking. In Trajnet CVPRW, 2018.										
MHT_DAM 28.	18.0	50.7 ±13.7	47.2	20.8%	36.9%	22,875	252,889	2,314 (41.9)	2,865 (51.9)	0.9	Public
	C. Kim, F. Li, A. Ciptadi, J. Rehg. Multiple Hypothesis Tracking Revisited. In ICCV, 2015.										

Ours

Best in MOT 2017

Conclusion: Bilinear LSTM

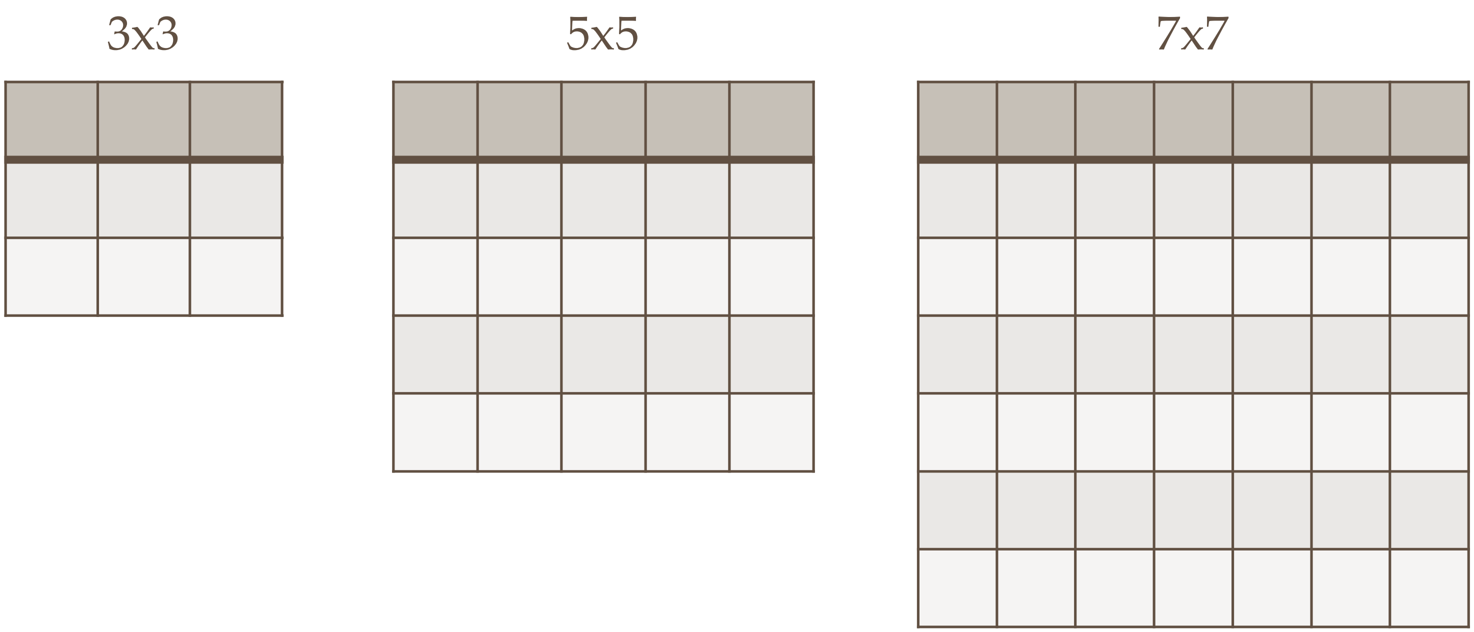
- We proposed Bilinear LSTM as an approach to learn long-term appearance model in tracking
- Experiments show that it significantly outperforms regular LSTM, especially in terms of identity switches
  - Bilinear LSTM seems capable of learning appearance model with multiple different appearances, where traditional LSTM struggles
- We hope that this methodology can be potentially useful in other scenarios beyond tracking

Today's Talk

- Multi-Target Tracking with bilinear LSTM
  - Novel LSTM model coming from studies on tracking
- **Understanding more about CNNs**
  - Generalization Theory based on Gaussian Complexity and Redesigns
  - XNN: Explaining CNN to human

Generalization Theory of CNN

- Have we ever questioned why are CNN filters always squares?



Why does a Sobel CNN filter generalize?

Sobel filter

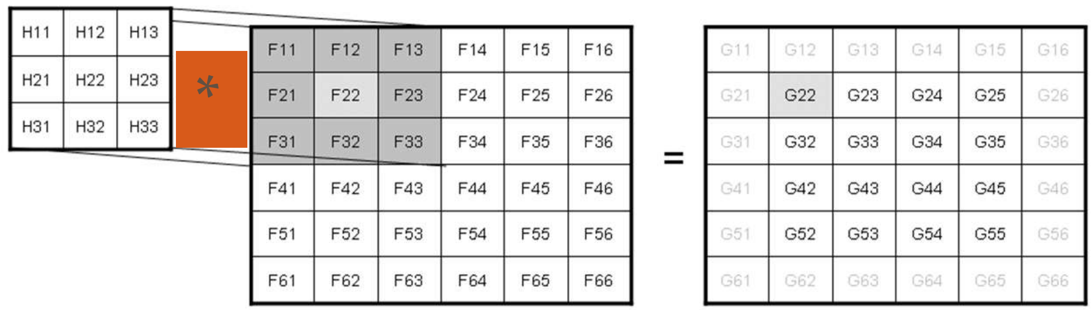
$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

Gx

$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Gy

Convolution



$$G_{ij} = \sum H_{kl} F_{i+k,j+l}$$

I

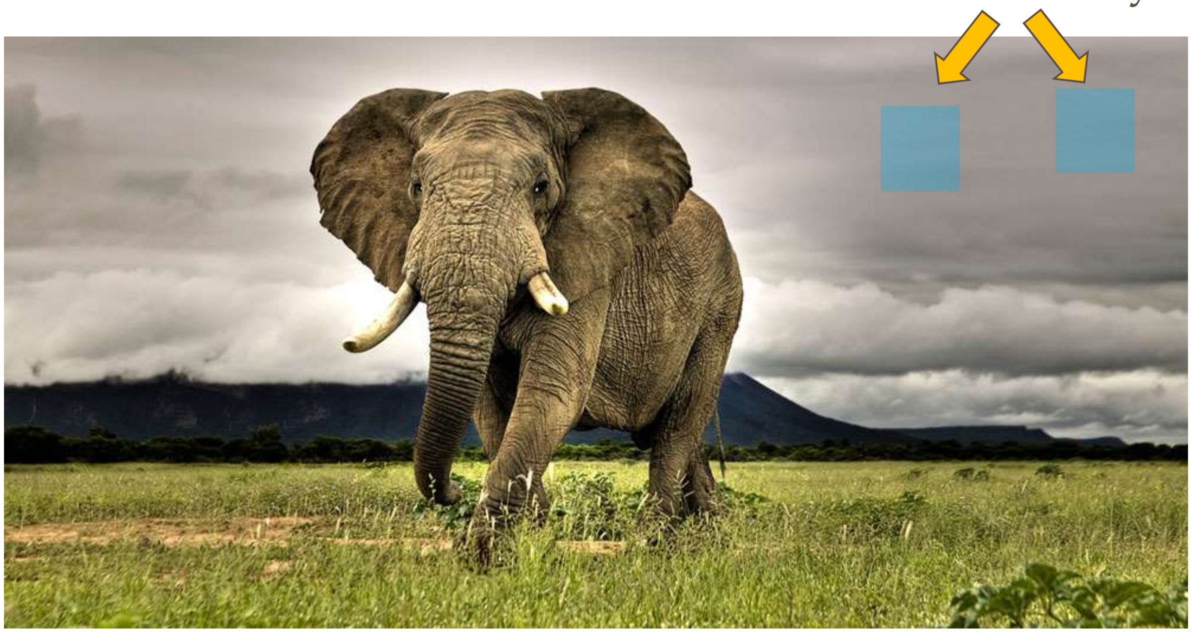
Convolution

$I * Gx$

Intuition of Generalization Capability

- In an image most of the time there is no boundary
  - A boundary is a pattern
  - A pattern is generalizable if it occurs rarely and most of the time there is no pattern

No boundary



Theory of Generalization Capability

Theorem: For a simple 2-layer Network:

$$F = \{\mathbf{x} \longrightarrow \sum_i v_i \sigma(\mathbf{w}_i * \mathbf{x}) : \| \mathbf{v} \| \leq 1 \| \mathbf{w} \| \leq 1\}$$

For any $x_1, \ldots, x_N \in \mathbb{R}^d$, the Gaussian complexity $(\widehat{\boldsymbol{G}}_N)$ of F satisfies

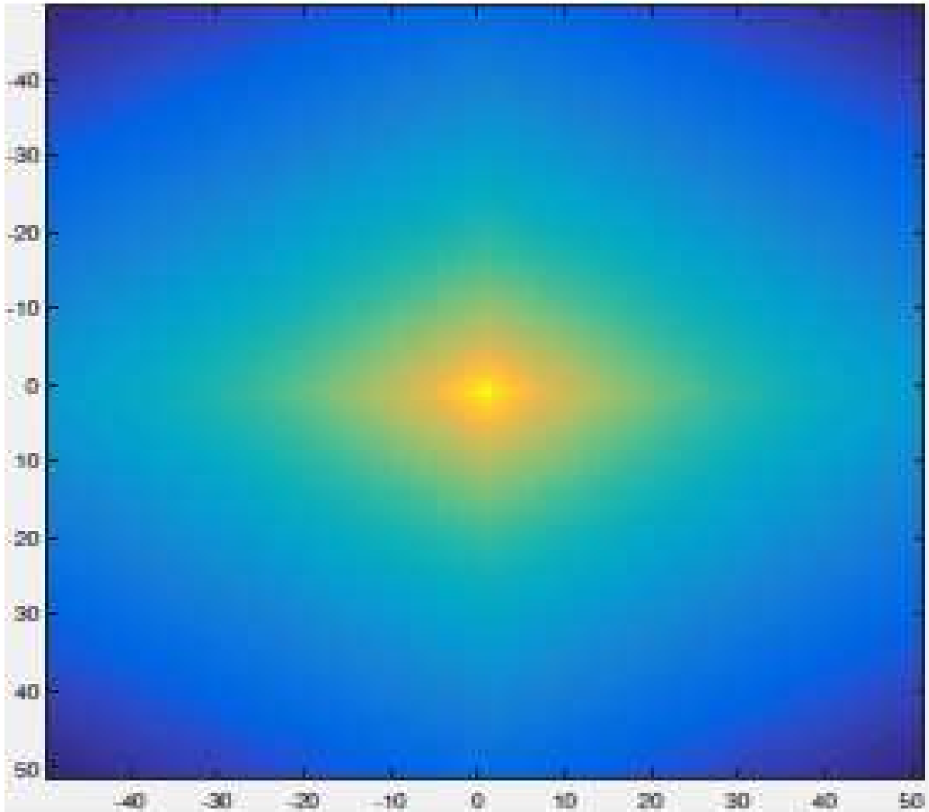
$$\widehat{\boldsymbol{G}}_N(F) \leq \frac{cB(\ln d)^{1/2}}{N} \max_{\mathbf{j}-\mathbf{j}' \in \mathcal{N}} \sqrt{\sum_1^N \| \mathrm{x}_i(\mathbf{j}) - \mathrm{x}_i(\mathbf{j}') \|^2}$$

where $\mathbf{j} - \mathbf{j}' \in \mathcal{N}$ means $\mathbf{j}$ and $\mathbf{j}'$ fall within the same filter

In simpler terms: in order to generalize, the CNN filter needs to choose a neighborhood in which the input are **highly correlated** with each other.

X. Li, FL, X. Fern, R. Raich. ICLR 2017

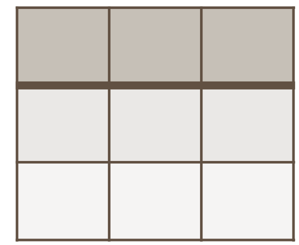
Cross-Correlation of Natural Images



Each pixel represents the cross-correlation between (x_0, y_0) and $(x_0 + \Delta x, y_0 + \Delta y)$

Averaged over all pixels on PASCAL VOC

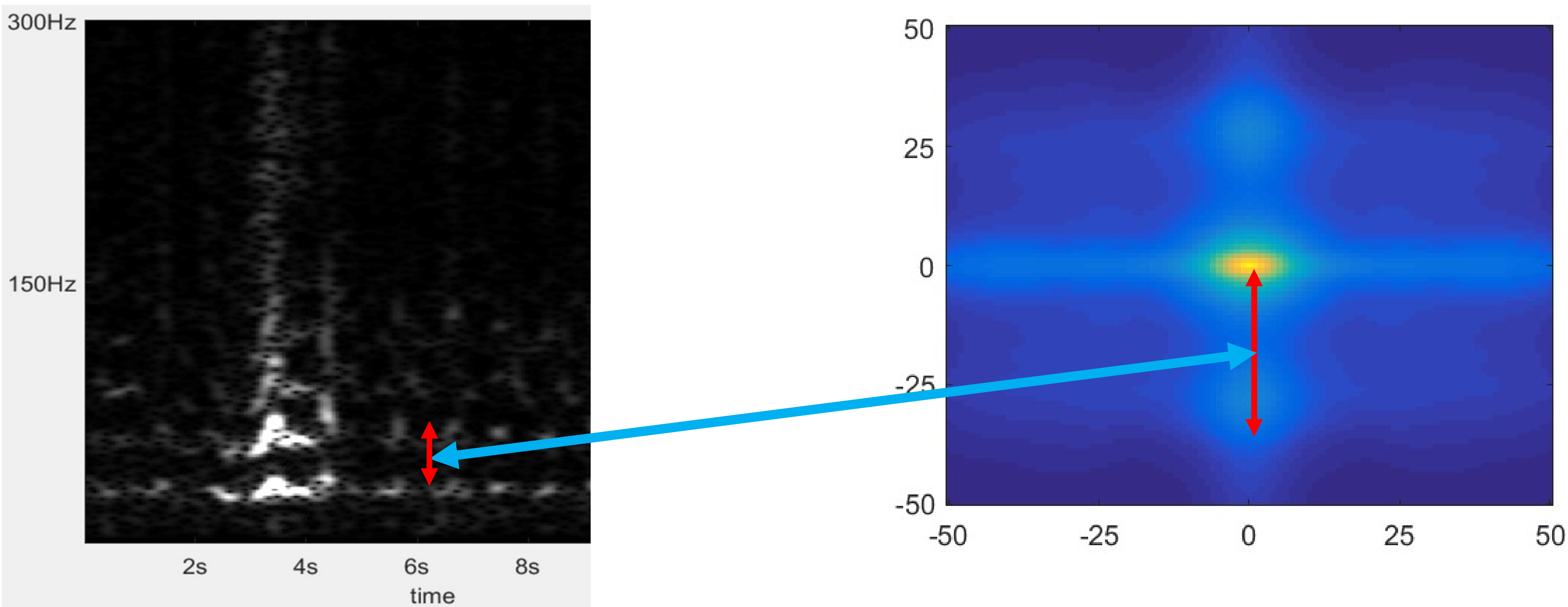
3x3



is the best!

What's the use of this?

- Consider a domain where the cross-correlation pattern is different:

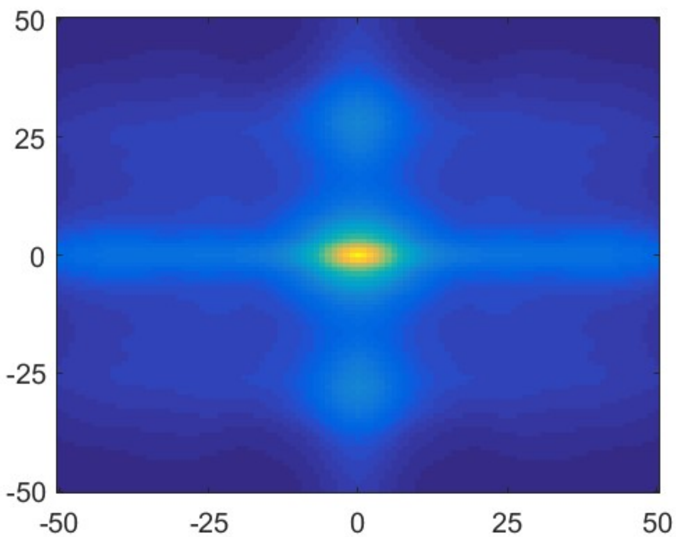


The CNN filter shape should be different too!

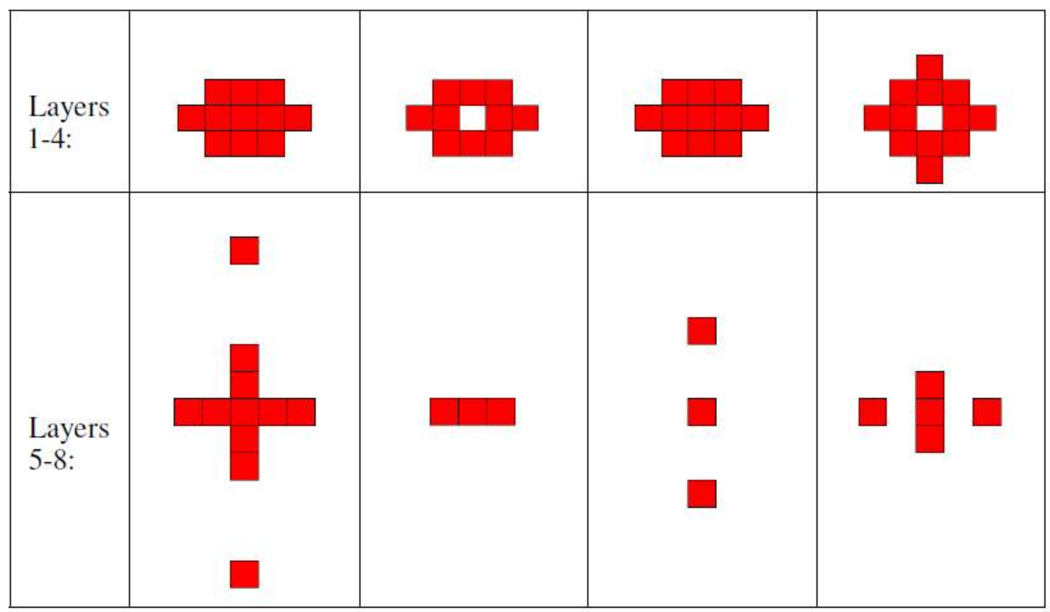
An Algorithm to Decide CNN Filter Shapes

- We proposed a LASSO algorithm that recursively selects the highest-correlated locations based on the correlation image
  - Which can learn filter shapes from unsupervised data

e.g. for this pattern



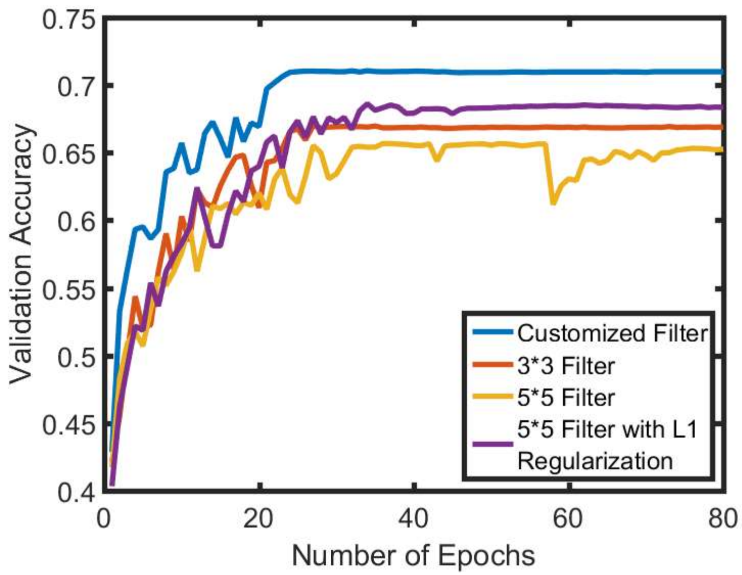
We learned CNN should have filters of these shapes



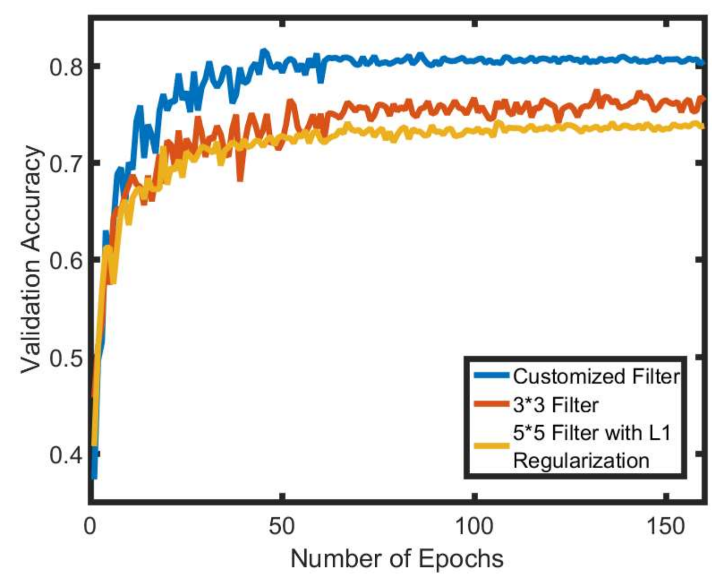
Experiments

- Recordings of hummingbird wingbeats and bird songs
  - Spectrogram data
  - 434 wingbeats recordings, 122 birdsong recordings
  - Cross-validation accuracy is reported

Bird Wingbeats Spectrogram



Birdsong Spectrogram

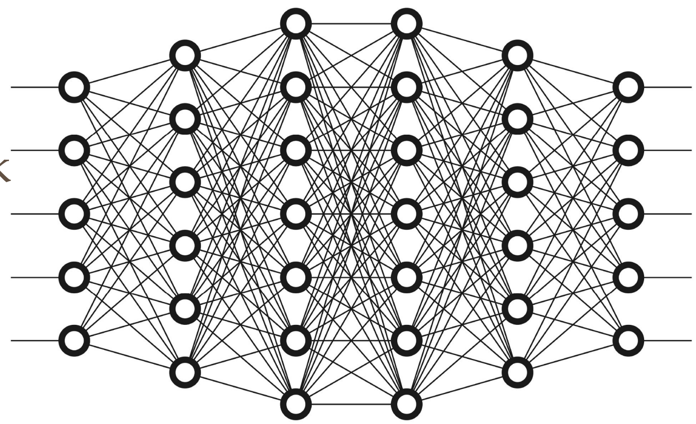


Explainable Deep Learning

- How can human understand a very deep network?
- How can human trust a deep network?

Very complex Deep Network

10-100M parameters

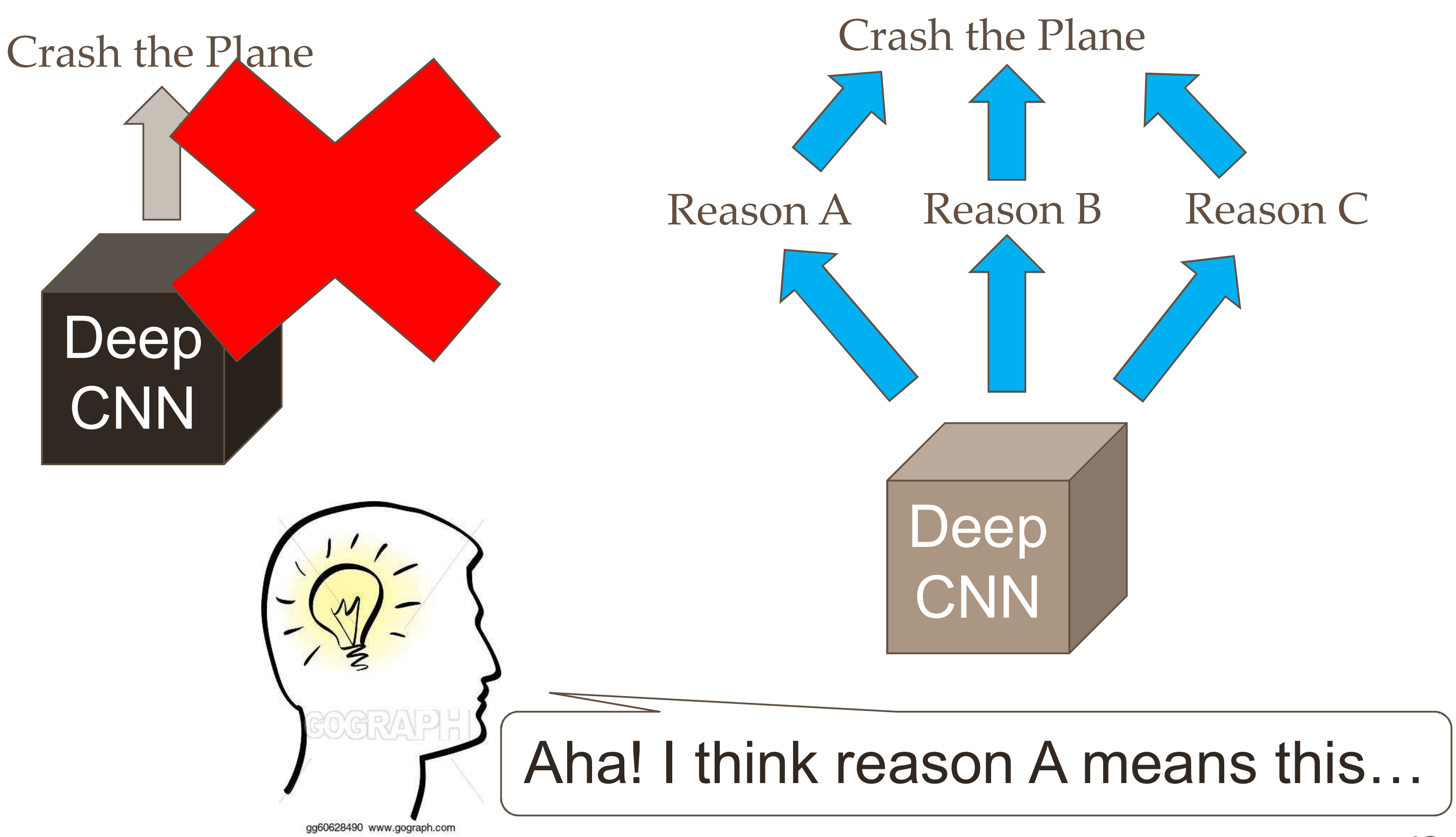


- Esp. in crucial decision making scenarios
- In an airplane, deep learning makes decision: Force land right now!
- In autonomous driving, deep learning makes decision: steer left to hit the highway separator!

- Need to generate *mental model* of deep learning that human can understand!

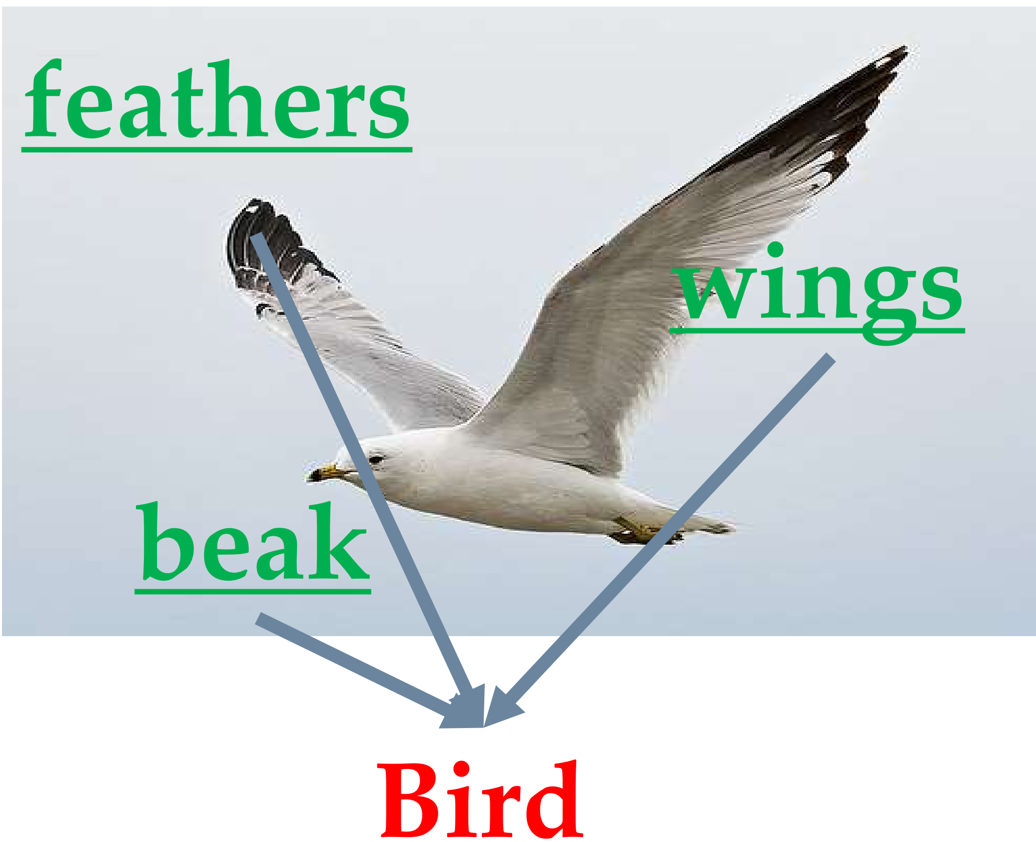
Explaining Deep Learning Predictions

Idea: Use the Deep Learning in Human Brain



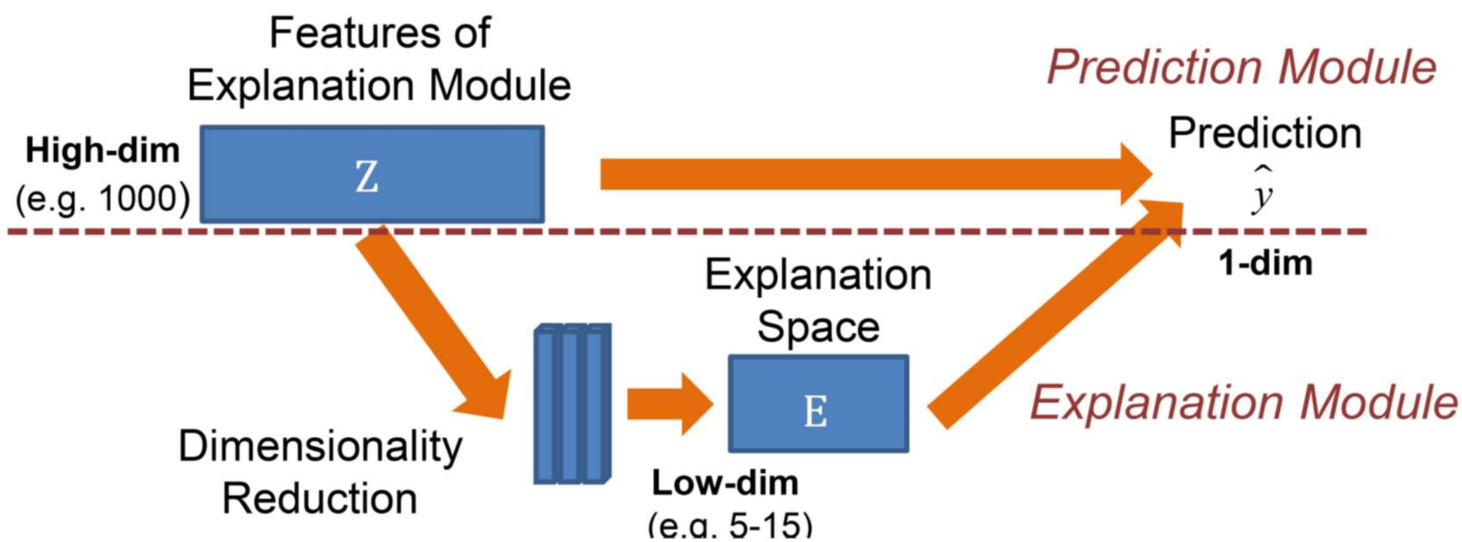
Explaining Deep Learning Predictions

"**A** is something because of **B**, **C**, and **D**".



B, C, and D need to be
*(1) **concise*** and
*(2) **high-level*** concepts.

XNN (Explanation Neural Network)



Explanation features need to be:

1) Faithful to the DNN it is explaining
2) Do not include irrelevant concepts
3) Each feature represents a different concept

XNN (Explanation Neural Network)

$$\min_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}},\mathbf{v}} \frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{v}^{\top}\mathbf{E}(\mathbf{Z}^{(i)};\boldsymbol{\theta}) - \hat{y}^{(i)}\right\|^2$$
$$+ \frac{\beta}{S_z}\sum_{k=1}^{S_z}\log(1 + q\cdot\frac{1}{M}\sum_{i=1}^{M}\left\|\phi^{-1}\left(\mathbf{E}(\mathbf{Z}^{(i)};\boldsymbol{\theta});\widetilde{\boldsymbol{\theta}}\right)_k - Z_k^{(i)}\right\|^2)$$
$$+ \eta\cdot\frac{1}{n(n-1)}\sum_{l=1}^{n}\sum_{l'\neq l}\left(\frac{\mathbf{E}_l^T\mathbf{E}_{l'}}{\|\mathbf{E}_l\|\,\|\mathbf{E}_{l'}\|}\right)^2$$

Faithfulness: attempts to be faithful to the original DNN

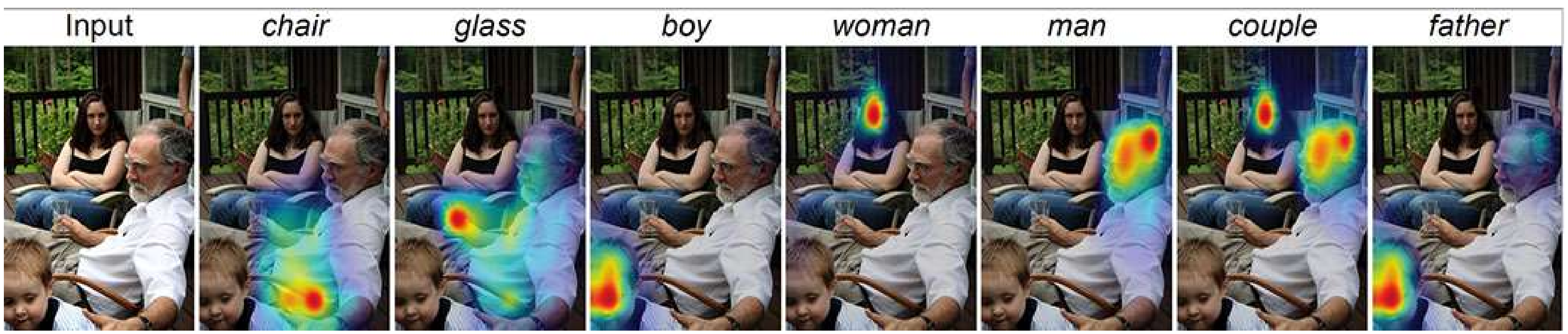
Sparse reconstruction: attempts to selectively reconstruct some dimensions of the features in a deep network

Orthogonality: attempts to make features orthogonal to each other

Visualization

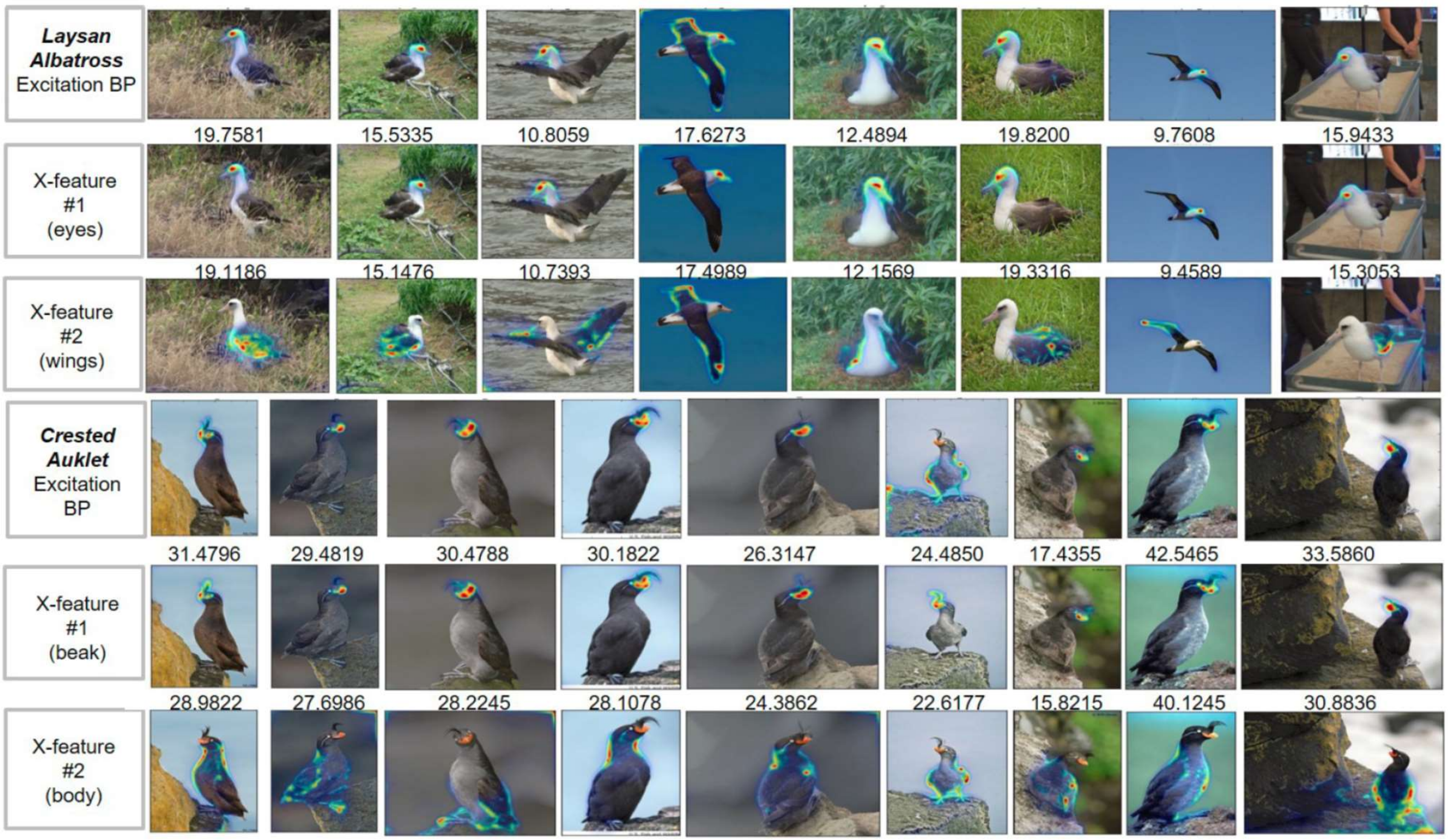
We can use heatmap tools to visualize the explanation features (x-features)

Heatmap tool:



They used to be used on classifications
Now used on explanation features

XNN Explaining Bird Classifications



Zhongang Qi, Saeed Khorram, FL.
Arxiv: 1709.05360

Quantatitive Evaluations

Important for explanation

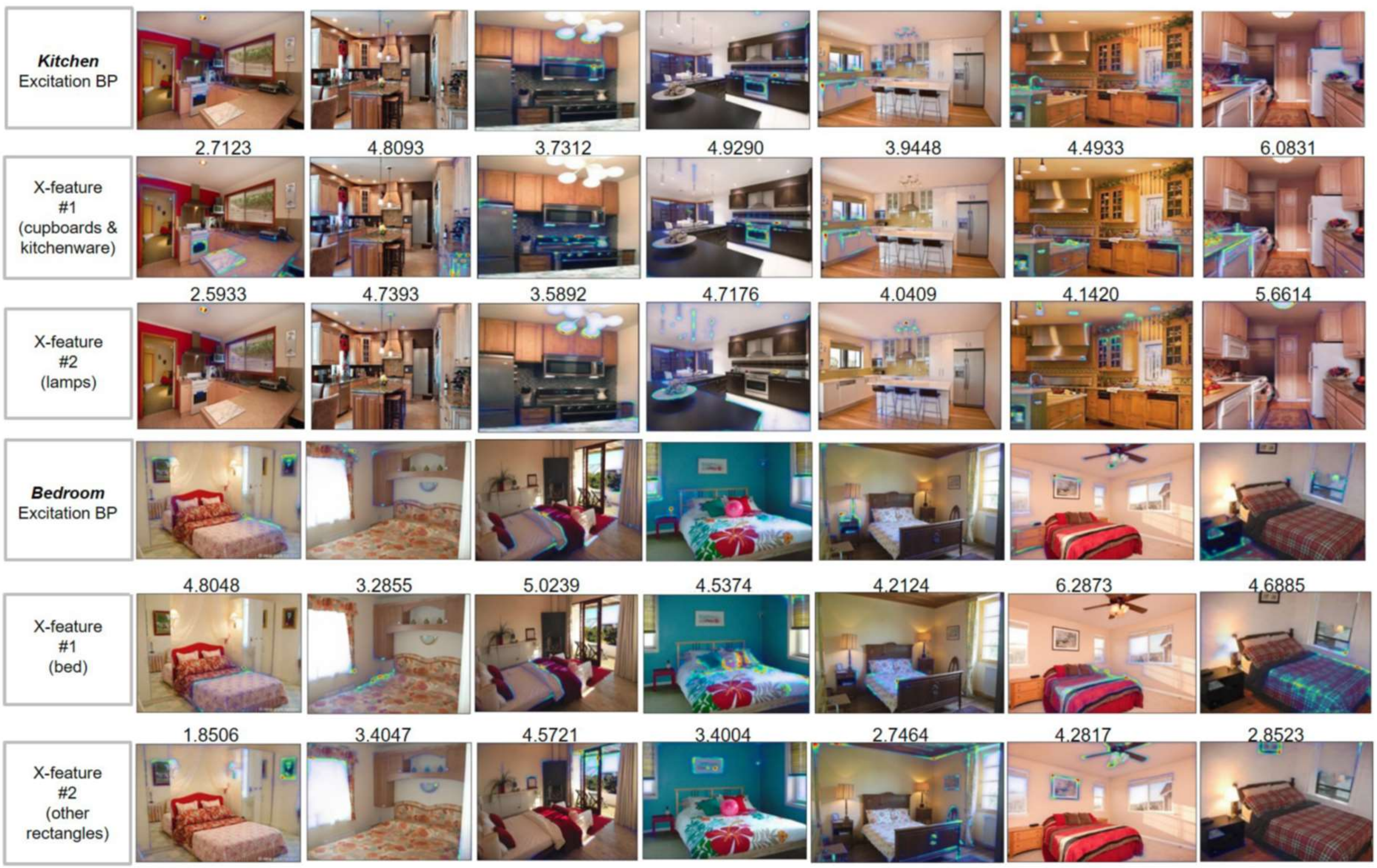
We evaluate 1) Faithfulness; 2) Orthogonality; 3) Locality (log of number of parts covered by each x-feature)

Locality evaluated because bird classification should be based on parts

Method		SRAE	NN	SAE	Lasso	CAE	Z	ExcitationBP
F_{reg}	Training	0.0812	0.0696	0.0972	3.5785	4.1513	—	—
	Testing	0.1659	0.1304	0.1981	3.7928	4.0021	—	—
F_{cls}	Training	99.99%	100.0%	99.99%	73.14%	65.34%	—	—
	Testing	99.99%	100.0%	99.98%	71.53%	69.28%	—	—
O1	Positive	**0.6554**	0.9765	0.8794	1.2052	**0.6301**	—	—
O2	Positive	**2.4312**	4.9112	3.5057	3.9851	**2.3884**	—	—
Locality	Positive	**1.9713**	2.4360	2.1997	2.1082	2.1227	**1.9685**	2.5659

Places-365 Dataset

Explain why CNN classify this room as a particular type



Places-365 Quantitative Evaluations

Method		SRAE	NN	SAE	Lasso	CAE	ExcitationBP
F_{reg}	Training	0.5527	0.3346	1.4768	4.0726	4.3579	—
	Testing	1.0260	0.8736	1.5505	4.3366	4.6553	—
F_{cls}	Training	97.22%	97.17%	94.59%	90.19%	90.11%	—
	Testing	94.79%	94.86%	93.29%	88.55%	88.42%	—
O1	Positive	**0.2252**	0.3472	0.4578	0.4729	0.2741	—
O2	Positive	**0.5617**	0.8852	1.0799	0.9194	0.5945	—
Locality	Positive	**2.7208**	2.7756	2.7819	2.7282	2.7627	2.7591

Conclusion about the second part

- We proposed 2 approaches that provided more understanding into CNN
- Gaussian complexity-based generalization theory explains why are CNN filters square-shaped
- Also provides an approach to learn filter shape if the data is not natural image
- XNN provides explanations of individual CNN predictions
- In the form of high-level heatmaps human can then read and reason about
- Many future work ahead

Thank You!

Fuxin Li: http://web.engr.oregonstate.edu/~lif
Email: lif@oregonstate.edu

2077 Kelley Engineering Center,
Oregon State University
Corvallis OR 97331

I would like to thank my collaborators who contributed to the work in these slides:

Georgia Tech:
Chanho Kim, James M. Rehg

Oregon State University:
Xingyi Li, Zhongang Qi, Saeed Khorram, Xiaoli Fern, Weng-Keen Wong