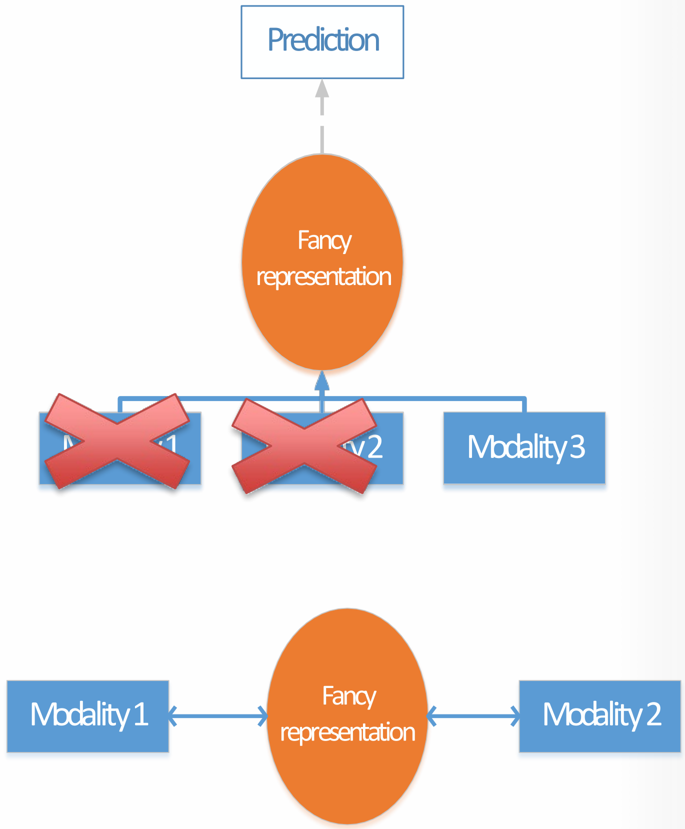
MULTIMODAL LEARNING

Chih-Chung Hsu (許志仲)
Assistant Professor
ACVLab, Institute of Data Science
National Cheng Kung University
https://cchsu.info

MULTIMODAL REPRESENTATIONS

Multimodal representations

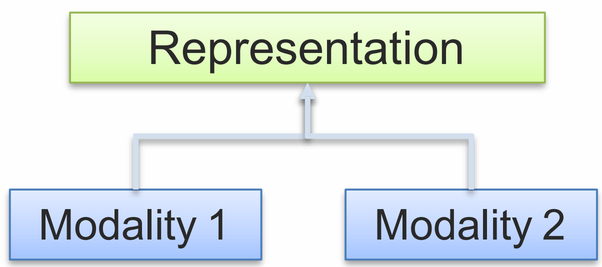
- What do we want from multi-modal representation
  - Similarity in that space implies similarity in corresponding *concepts*
  - Useful for various discriminative tasks – retrieval, mapping, fusion etc.
  - Possible to obtain in absence of one or more modalities
  - Fill in missing modalities given others (map between modalities)



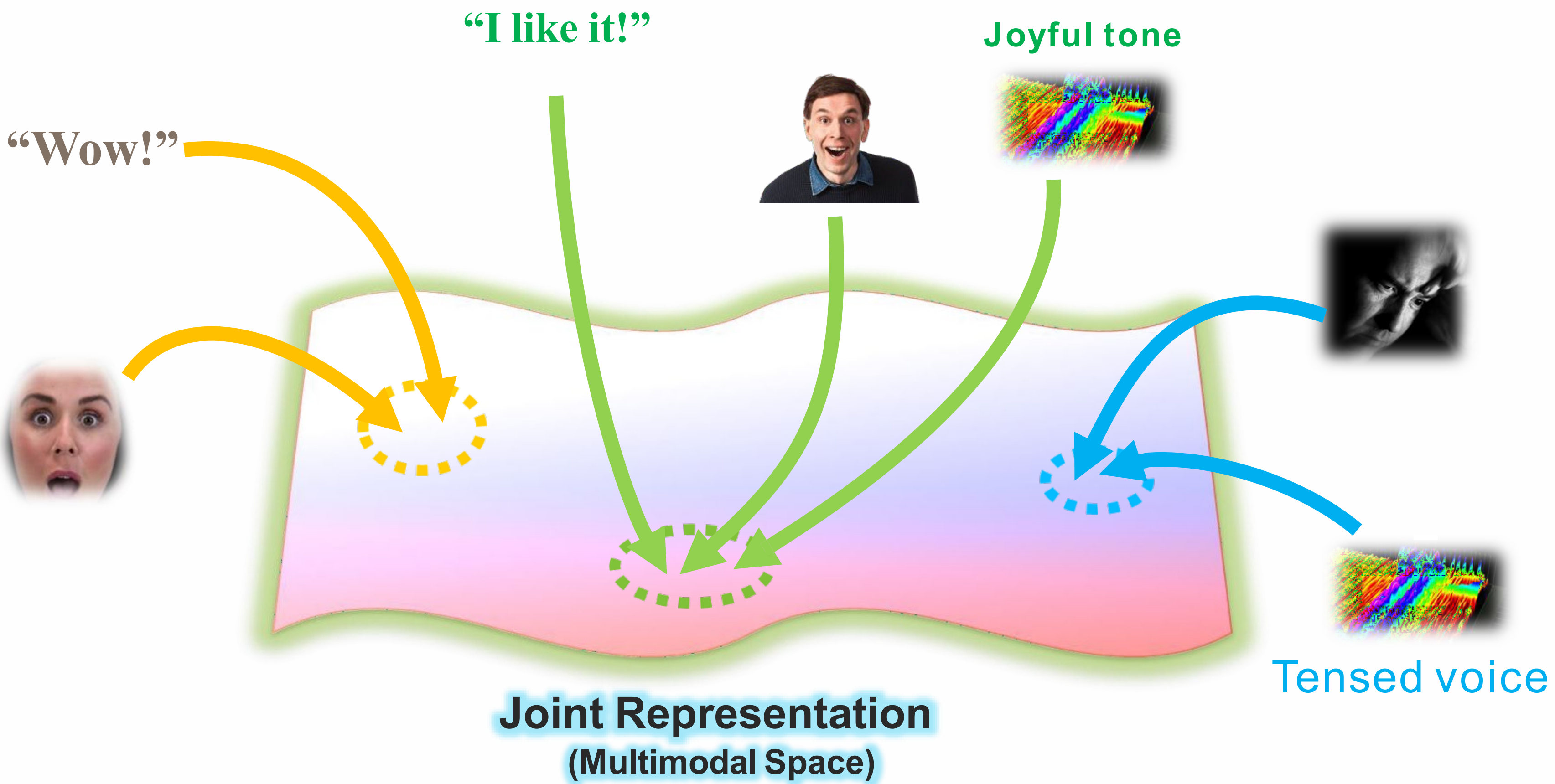
Core Challenge: Multimodal Representation

Definition: Learning how to represent and summarize multimodal data in away that exploits the complementarity and redundancy.

A Joint representations:



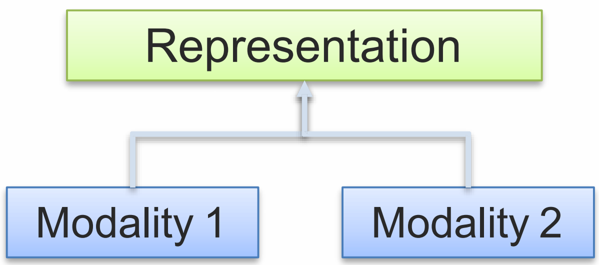
Joint Multimodal Representation



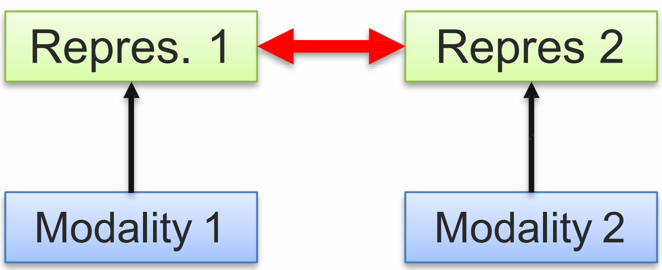
Core Challenge 1: Representation

Definition: Learning how to represent and summarize multimodal data in away that exploits the complementarity and redundancy.

A Joint representations:



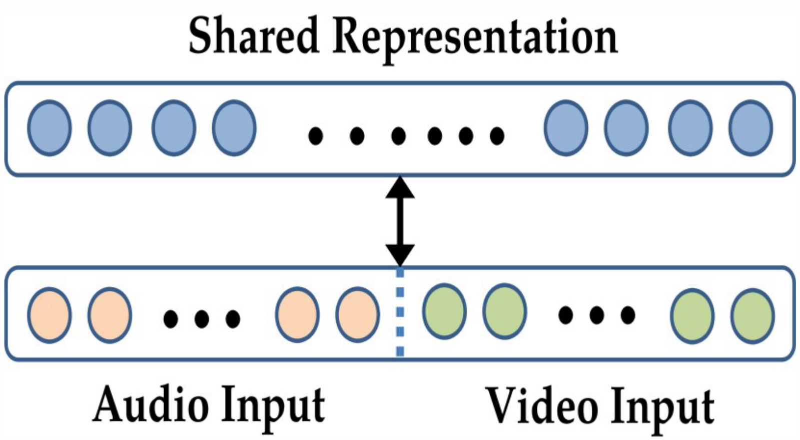
B Coordinated representations:



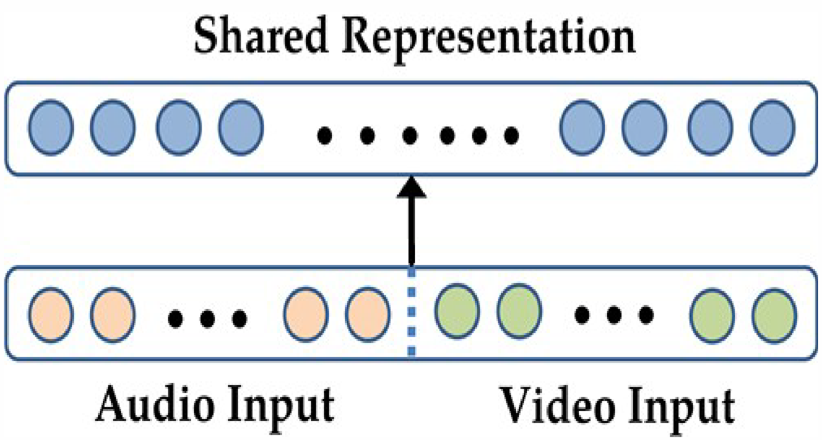
UNSUPERVISED
JOINT REPRESENTATIONS

Shallow multimodal representations

- Want deep multimodal representations
  - Shallow representations do not capture complex relationships
  - Often shared layer only maps to the shared section directly



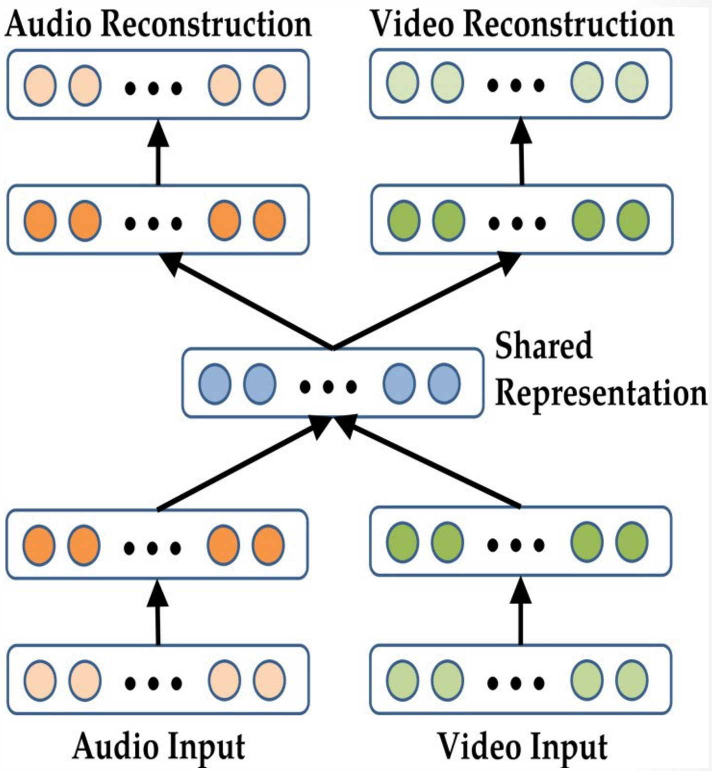
Shallow RBM



Shallow Autoencoder

Deep Multimodal autoencoders

- A deep representation learning approach
- A bimodal auto-encoder
  - Used for Audio-visual speech recognition

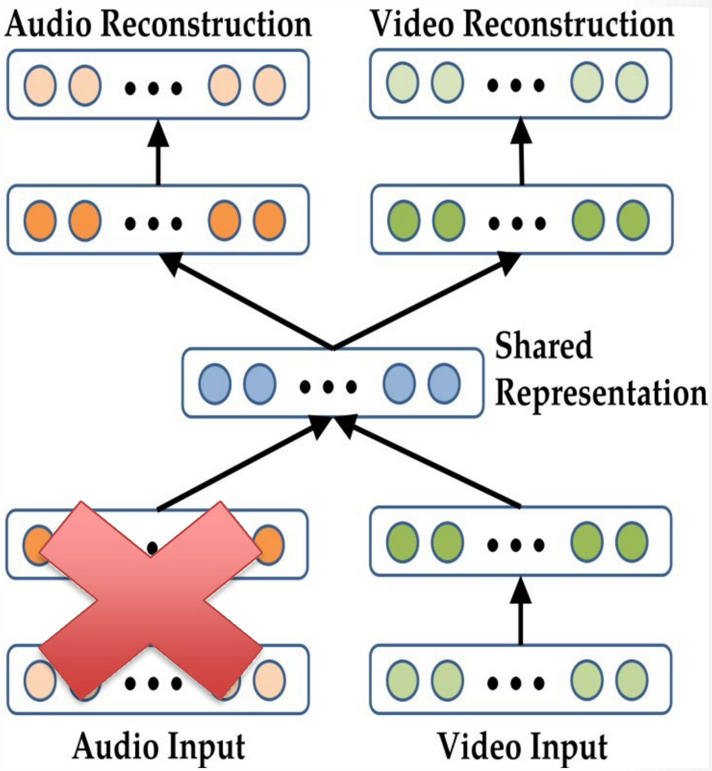


- [Ngiam et al., Multimodal Deep Learning, 2011]

Deep Multimodal autoencoders - training

- Individual modalities can be pre-trained
  - Denoising Autoencoders

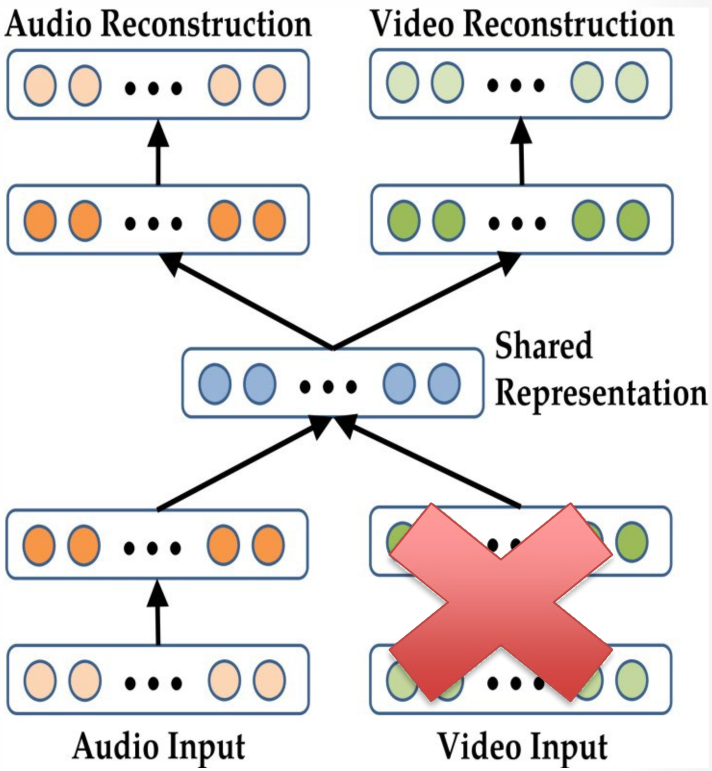
- To train the model to reconstruct the other modality
  - Use both
  - Remove audio



Deep Multimodal autoencoders - training

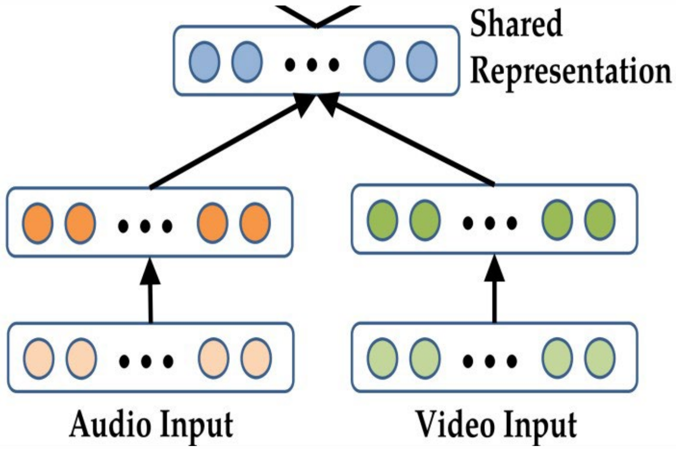
- Individual modalities can be pretrained
  - RBMs
  - Denoising Autoencoders

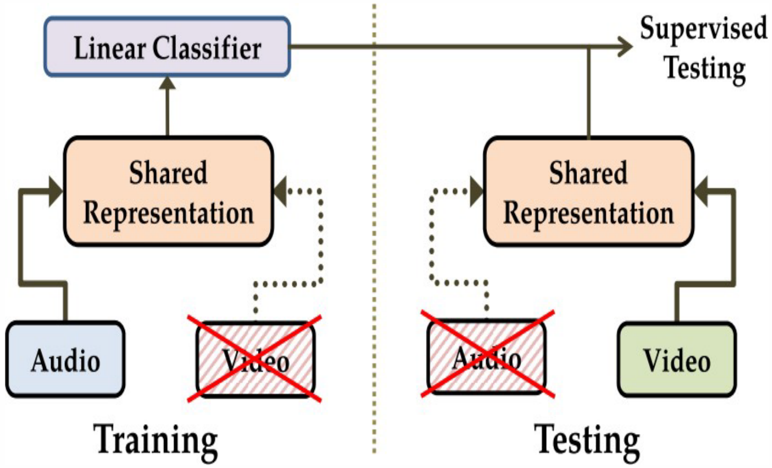
- To train the model to reconstruct the other modality
  - Use both
  - Remove audio
  - Remove video



Deep Multimodal autoencoders

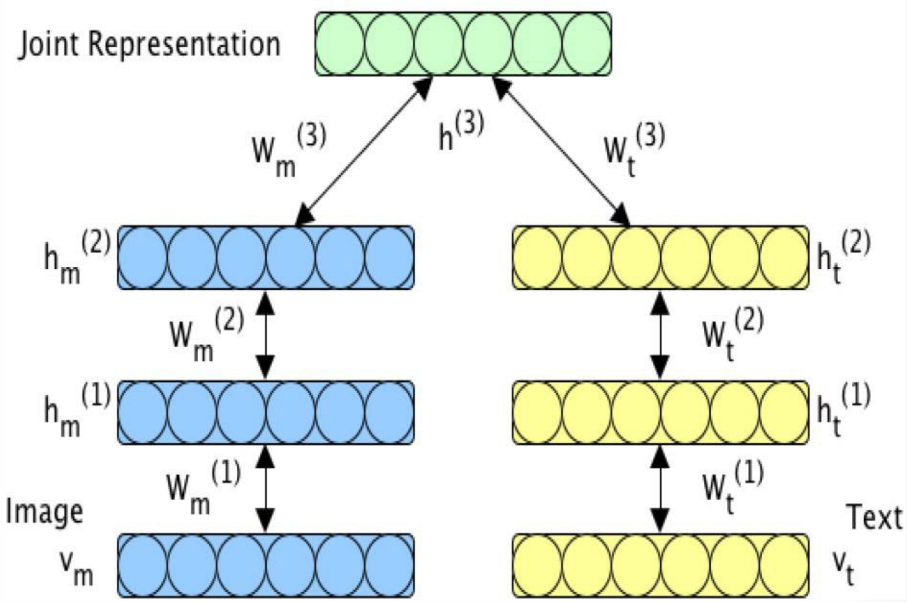
- Can now discard the decoder and use it for the AVSR task
- Interesting experiment
  - "Hearing to see"





Deep Multimodal Boltzmann machines

- Generative model
- Individual modalities trained like a DBN
- Multimodal representation trained using Variational approaches
- Used for image tagging and cross-media retrieval
- Reconstruction of one modality from another is a bit more "natural" than in autoencoder representation
- Can actually sample text and images

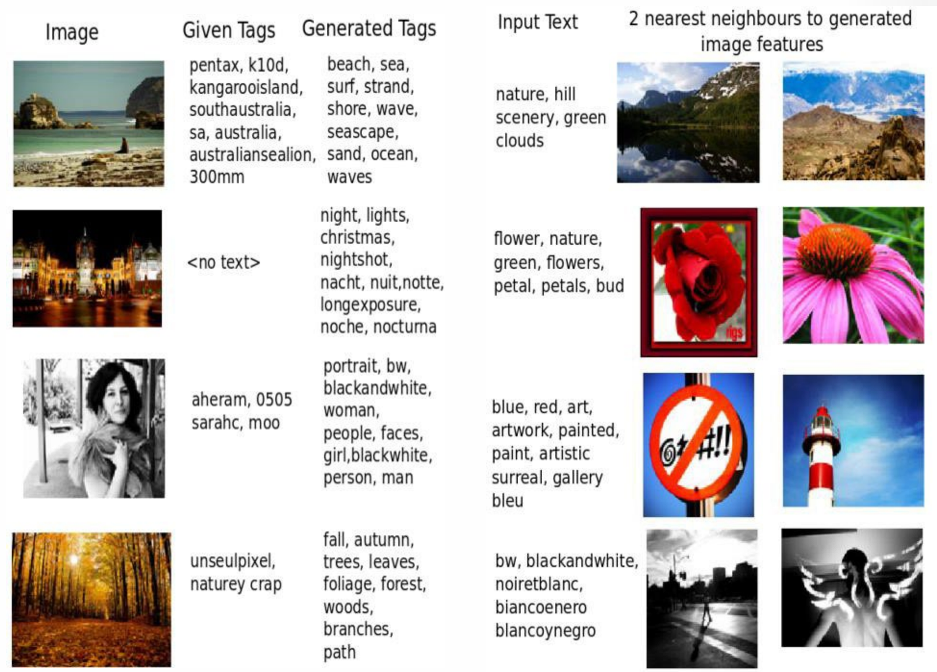


- [Srivastava and Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, 2012, 2014]

Deep Multimodal Boltzmann machines

- Pre-training on unlabeled data helps
- Can use generative models

Model	MAP	Prec@50
Random	0.124	0.124
SVM (Huiskes et al., 2010)	0.475	0.758
LDA (Huiskes et al., 2010)	0.492	0.754
DBM	0.526 ± 0.007	0.791 ± 0.008
DBM (using unlabelled data)	$\mathbf{0.585} \pm 0.004$	$\mathbf{0.836} \pm 0.004$



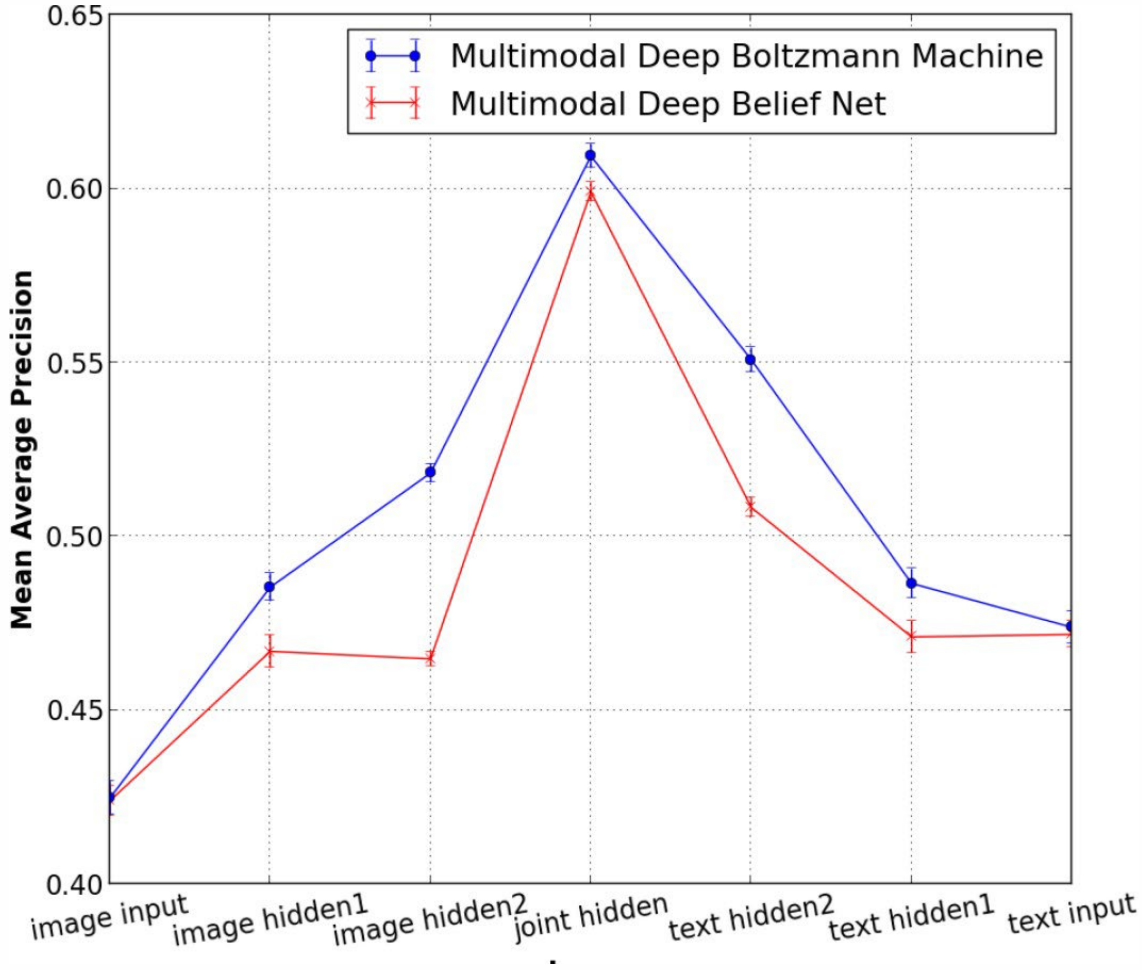
- Code is available
  - http://www.cs.toronto.edu/~nitish/multimodal/

Deep Multimodal Boltzmann Machines

- Text information can help visual predictions!
  - Image retrieval task on MIR Flickr dataset

Model	MAP	Prec@50
Image LDA (Huiskes et al., 2010)	0.315	-
Image SVM (Huiskes et al., 2010)	0.375	-
Image DBN	0.463 ± 0.004	0.801 ± 0.005
Image DBM	0.469 ± 0.005	0.803 ± 0.005
Multimodal DBM (generated text)	$\mathbf{0.531 \pm 0.005}$	$\mathbf{0.832 \pm 0.004}$

Analyzing Intermediate Representations



Comparing deep multimodal representations

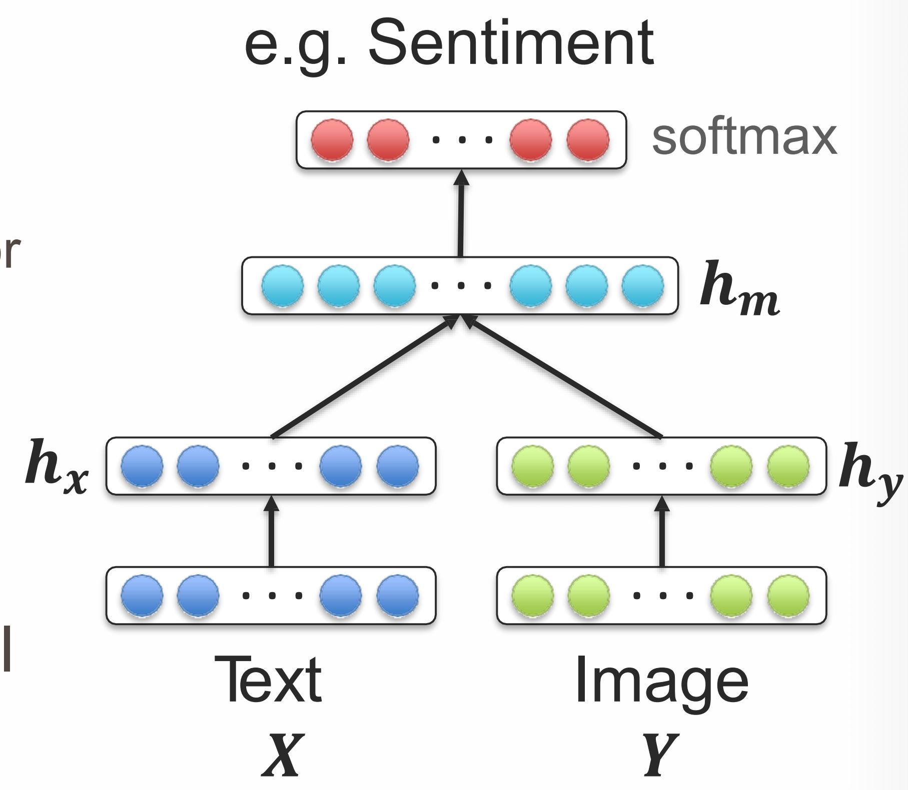
- Difference between them and the RBMs and the autoencoders
- Overall very similar behavior

Model	DBN	DAE	DBM
Logistic regression on joint layer features	0.599 ± 0.004	0.600 ± 0.004	0.609 ± 0.004
Sparsity + Logistic regression on joint layer features	0.626 ± 0.003	0.628 ± 0.004	0.631 ± 0.004
Sparsity + discriminative fine-tuning	0.630 ± 0.004	0.630 ± 0.003	0.634 ± 0.004
Sparsity + discriminative fine-tuning + dropout	0.638 ± 0.004	0.638 ± 0.004	$\mathbf{0.641} \pm \mathbf{0.004}$

SUPERVISED
JOINT REPRESENTATIONS

Multimodal Joint Representation

- For supervised learning tasks
- Joining the unimodal representations:
  - Simple concatenation
  - Element-wise multiplication or summation
  - Multilayer perceptron
- How to explicitly model both unimodal and bimodal interactions?



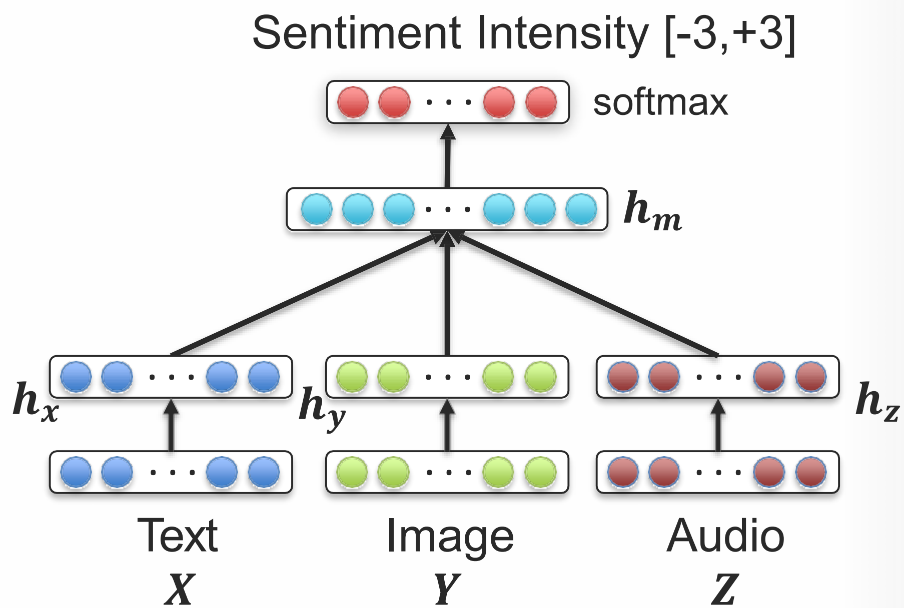
Multimodal Sentiment Analysis

MOSI dataset (Zadeh et al, 2016)

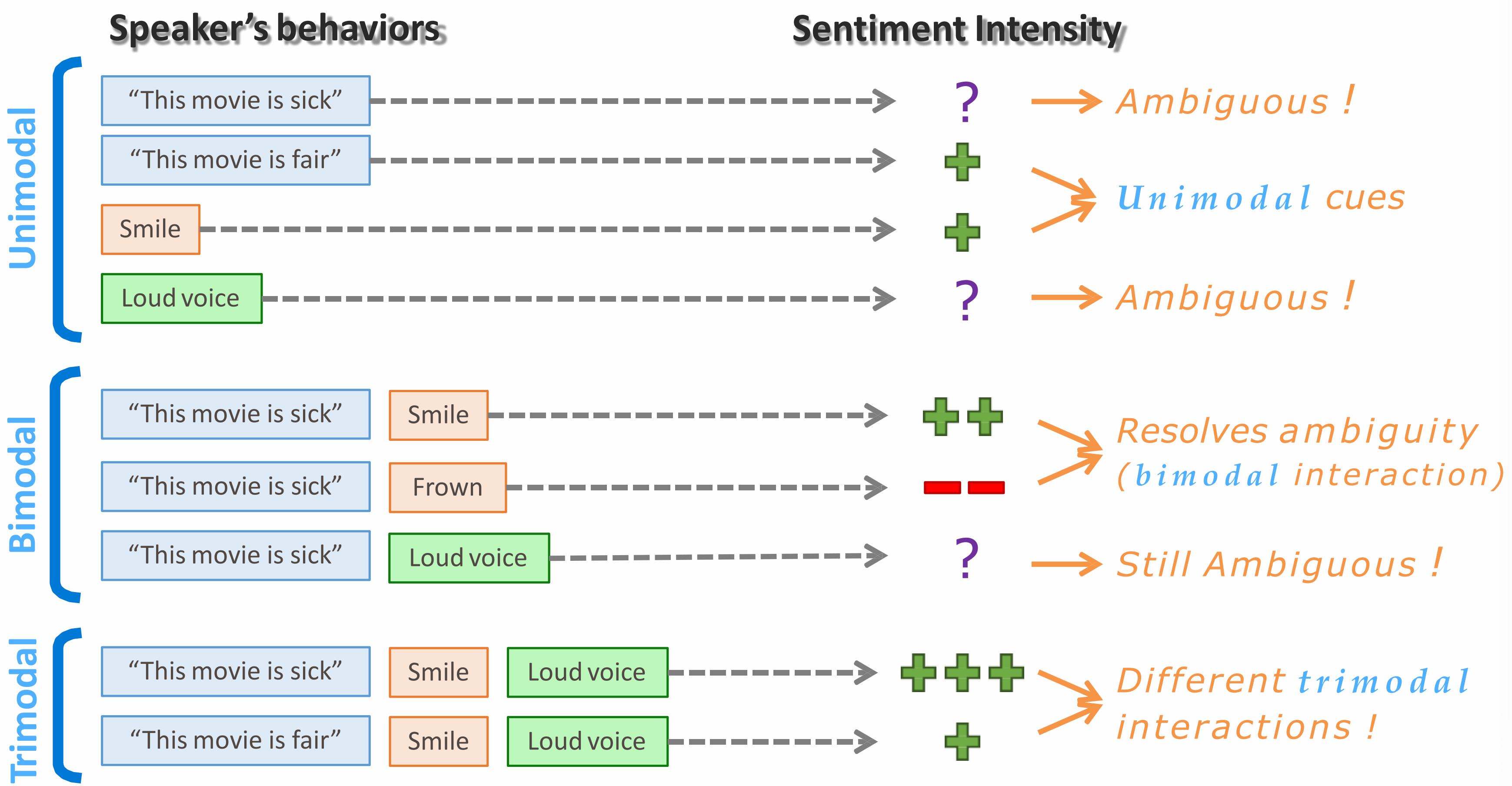
- 2199 subjective video segments
- Sentiment intensity annotations
- 3 modalities: text, video, audio

Multimodal joint representation:

$$h_m = f\left(W \cdot \left[h_x, h_y, h_z\right]\right)$$



Unimodal, Bimodal and Trimodal Interactions

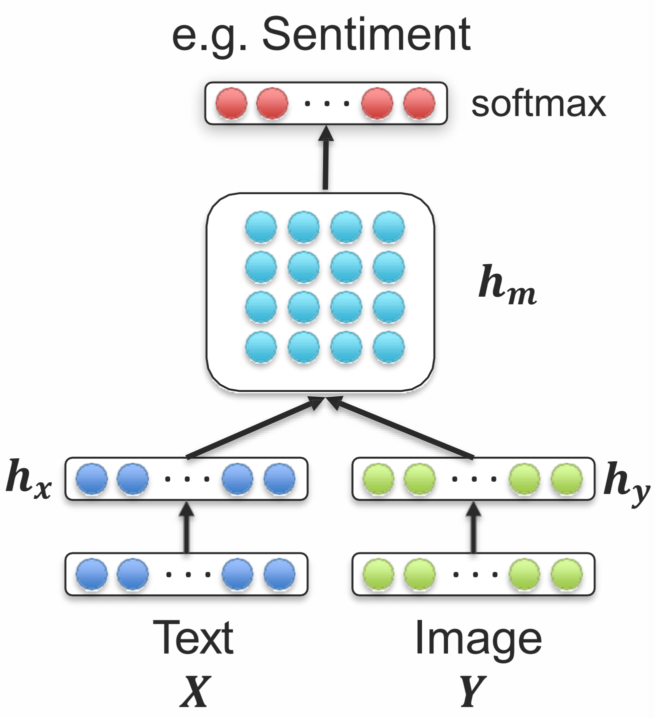


Bilinear Pooling

Models bimodal interactions:

$$\boldsymbol{h_m} = \boldsymbol{h_x} \otimes \boldsymbol{h_y} = \boldsymbol{h_y} \otimes \boldsymbol{h_x}$$

[Tenenbaum and Freeman, **2000**]

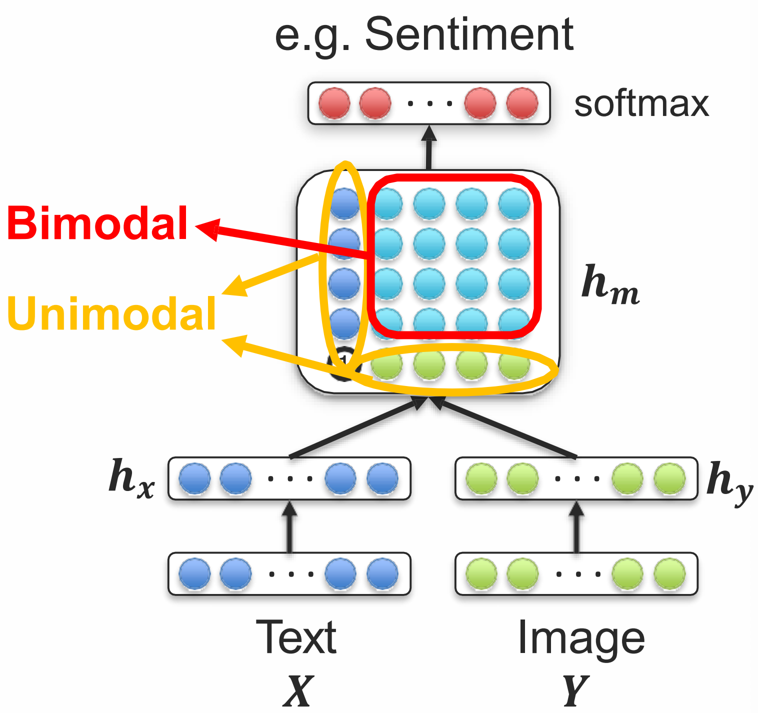


Multimodal Tensor Fusion Network (TFN)

Models both unimodal and bimodal interactions:

$$\boldsymbol{h}_m = \begin{bmatrix} \boldsymbol{h}_x \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{h}_y \\ 1 \end{bmatrix} = \begin{bmatrix} \boxed{\boldsymbol{h}_x} & \boxed{\boldsymbol{h}_x \otimes \boldsymbol{h}_y} \\ 1 & \boxed{\boldsymbol{h}_y} \end{bmatrix}$$

Important !

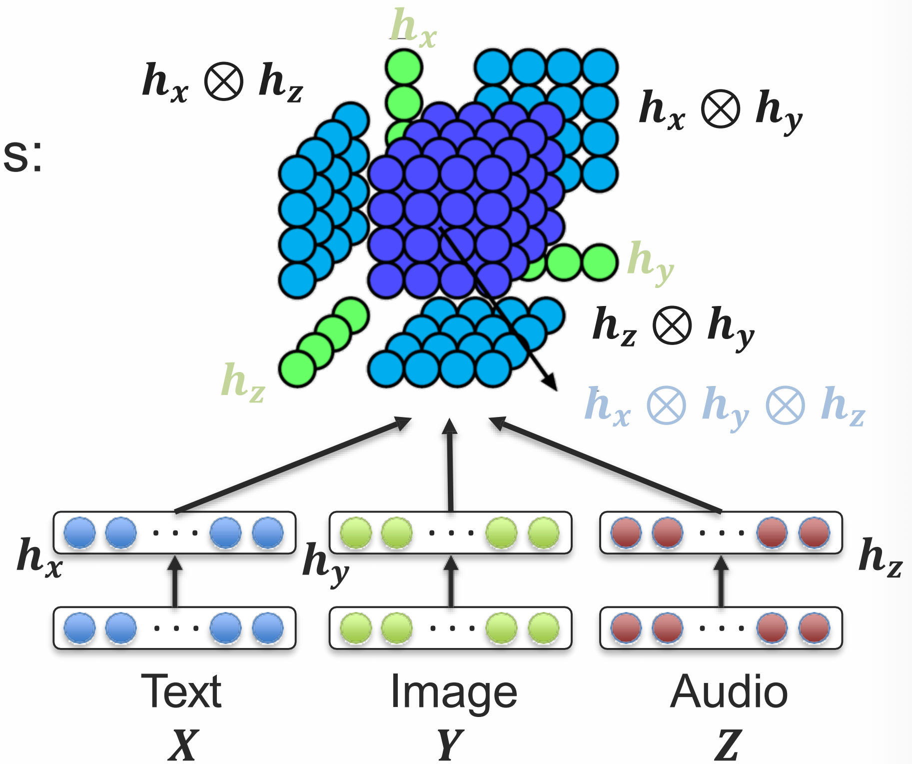


[Zadeh, Jones and Morency, **EMNLP 2017**]

Multimodal Tensor Fusion Network (TFN)

Can be extended to three modalities:

$$\boldsymbol{h}_m = \begin{bmatrix} \boldsymbol{h}_x \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{h}_y \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{h}_z \\ 1 \end{bmatrix}$$



Explicitly models unimodal, bimodal and trimodal interactions !

[Zadeh, Jones and Morency, **EMNLP 2017**]

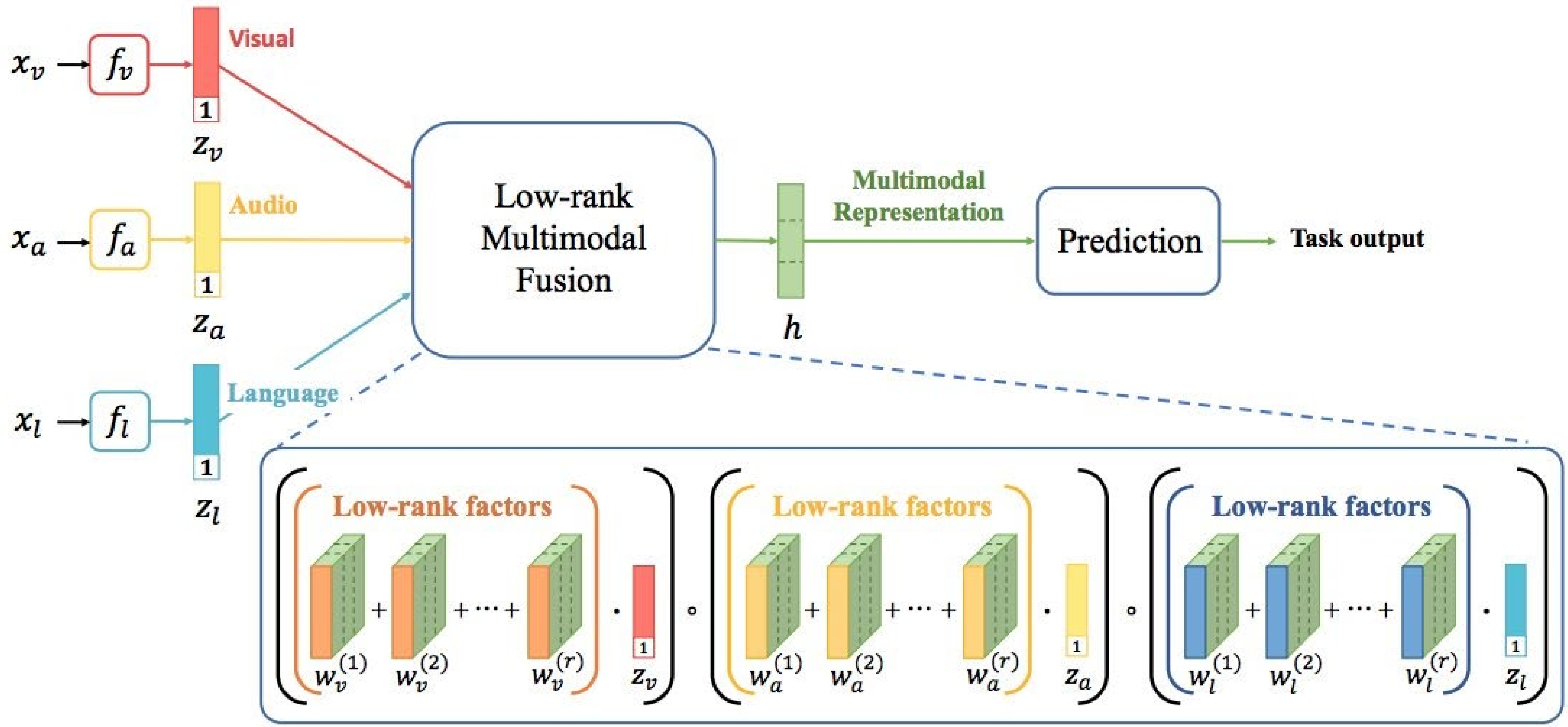
Experimental Results – MOSI Dataset

Multimodal Baseline	Binary		5-class	Regression	
	Acc(%)	F1	Acc(%)	MAE	r
Random	50.2	48.7	23.9	1.88	-
C-MKL	73.1	75.2	35.3	-	-
SAL-CNN	73.0	-	-	-	-
SVM-MD	71.6	72.3	32.0	1.10	0.53
RF	71.4	72.1	31.9	1.11	0.51
TFN	**77.1**	**77.9**	**42.0**	**0.87**	**0.70**
Human	85.7	87.5	53.9	0.71	0.82
Δ^{SOTA}	$\uparrow$ 4.0	$\uparrow$ 2.7	$\uparrow$ 6.7	$\downarrow$ 0.23	$\uparrow$ 0.17

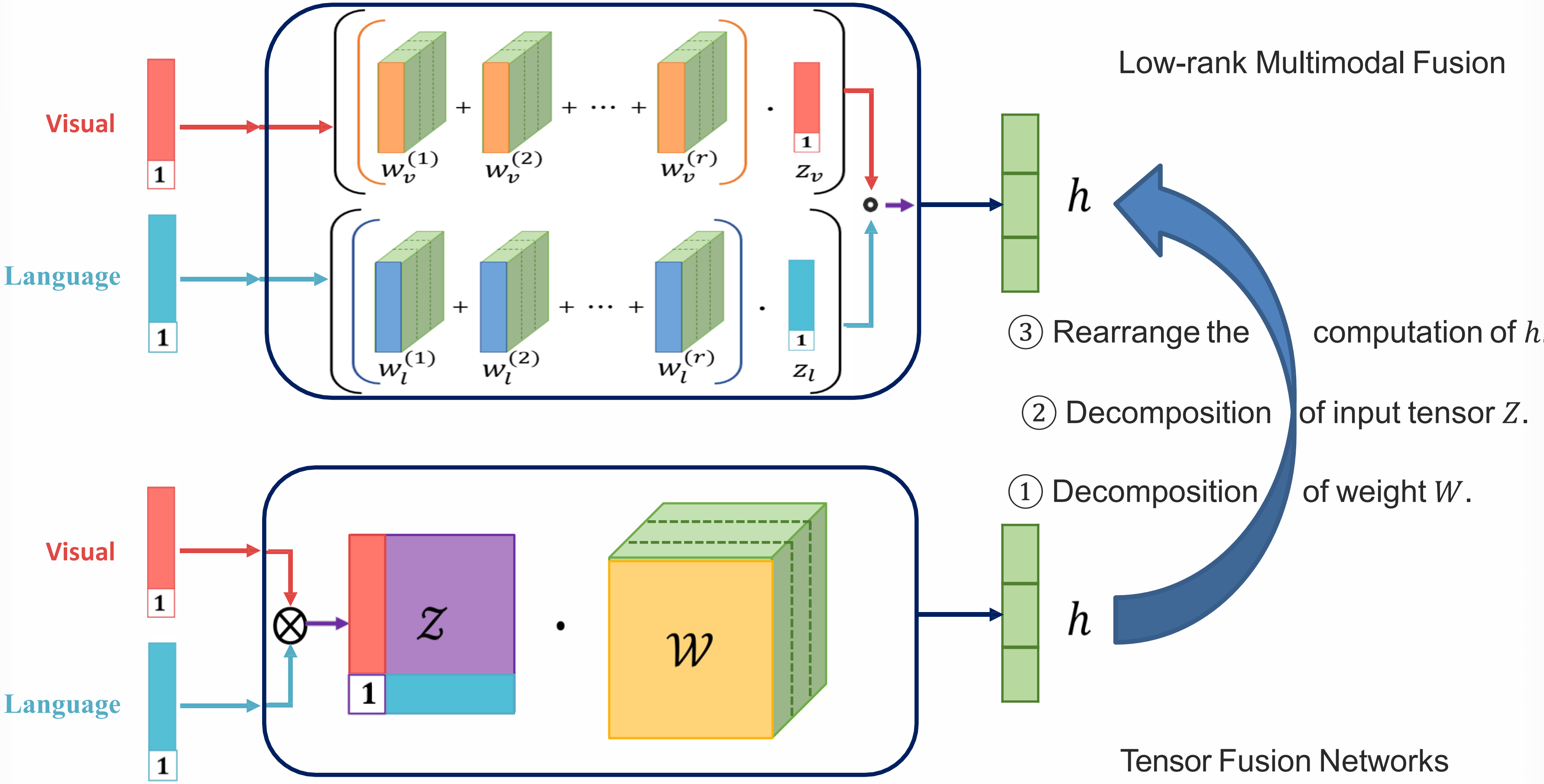
Improvement over State-Of-The-Art

Baseline	Binary		5-class	Regression	
	Acc(%)	F1	Acc(%)	MAE	r
$\text{TFN}_{language}$	74.8	75.6	38.5	0.99	0.61
TFN_{visual}	66.8	70.4	30.4	1.13	0.48
$\text{TFN}_{acoustic}$	65.1	67.3	27.5	1.23	0.36
$\text{TFN}_{bimodal}$	75.2	76.0	39.6	0.92	0.65
$\text{TFN}_{trimodal}$	74.5	75.0	38.9	0.93	0.65
$\text{TFN}_{notrimodal}$	75.3	76.2	39.7	0.919	0.66
TFN	**77.1**	**77.9**	**42.0**	**0.87**	**0.70**
TFN_{early}	75.2	76.2	39.0	0.96	0.63

From Tensor Representation to Low-rank Fusion

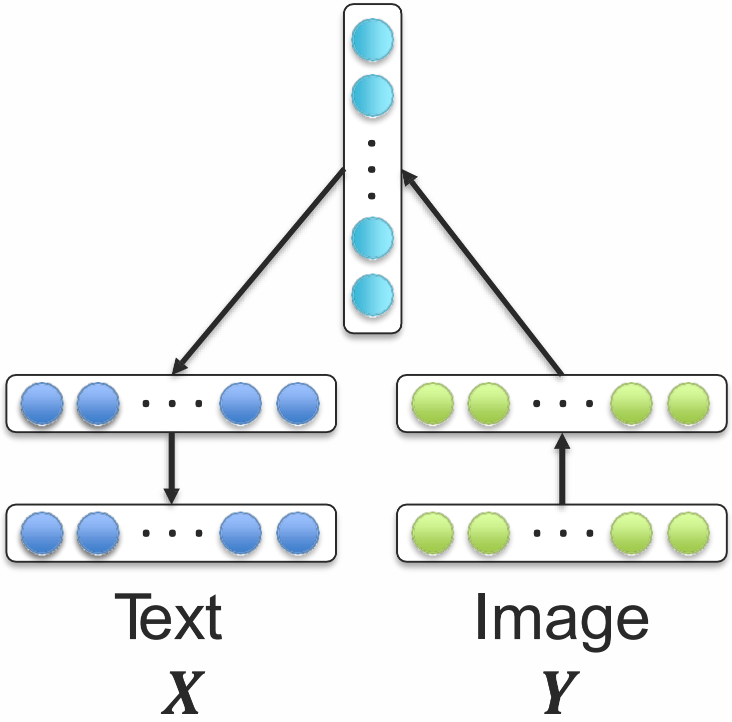


From Tensor Representation to Low-rank Fusion



Multimodal Encoder-Decoder

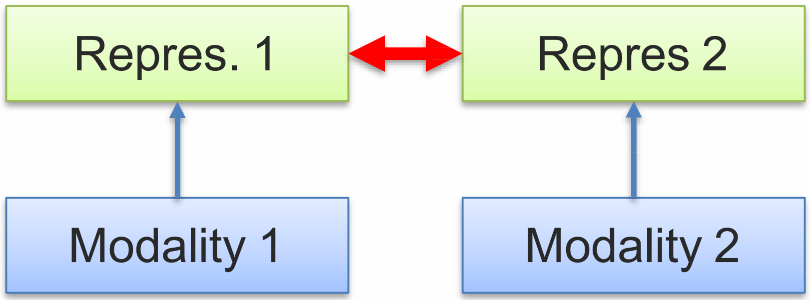
- Visual modality often encoded using CNN
- Language modality will be decoded using LSTM
  - A simple multilayer perceptron will be used to translate from visual (CNN) to language (LSTM)



COORDINATED MULTIMODAL REPRESENTATIONS

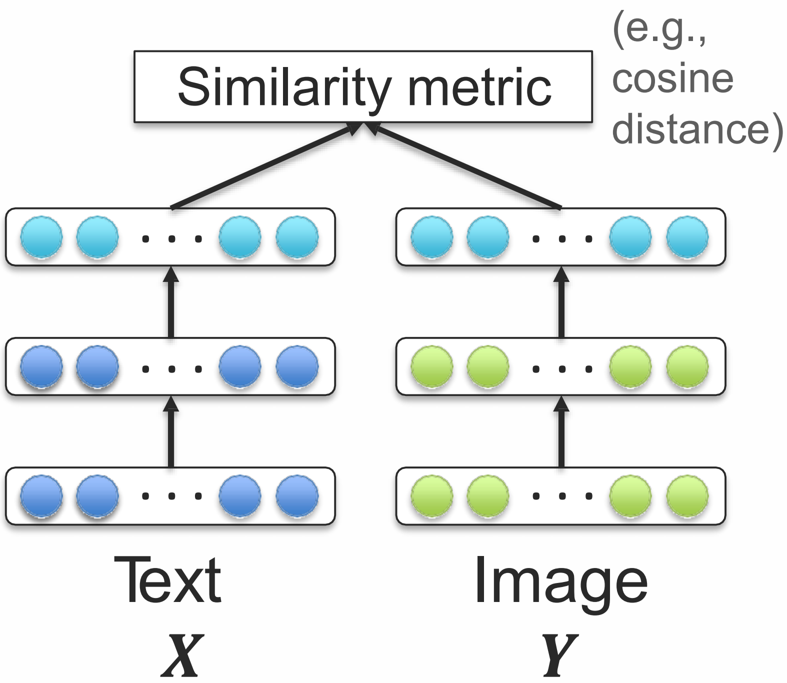
Coordinated multimodal embeddings

- Instead of projecting to a joint space enforce the similarity between unimodal embeddings



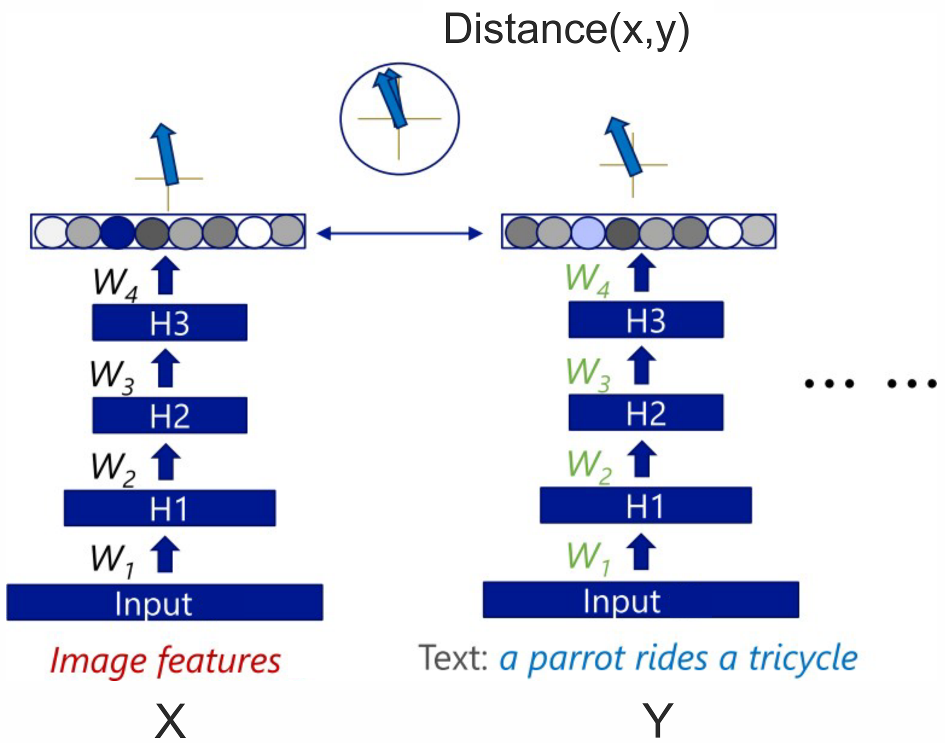
Coordinated Multimodal Representations

- Learn (unsupervised) two or more coordinated representations from multiple modalities.
- A loss function is defined to bring closer these multiple representations.



Coordinated Multimodal Embeddings

What should be the loss function?



[Frome et al., DeViSE: A Deep Visual-Semantic Embedding Model, NIPS 2013]

Max-Margin Loss – Multimodal Embeddings

What should be the loss function?

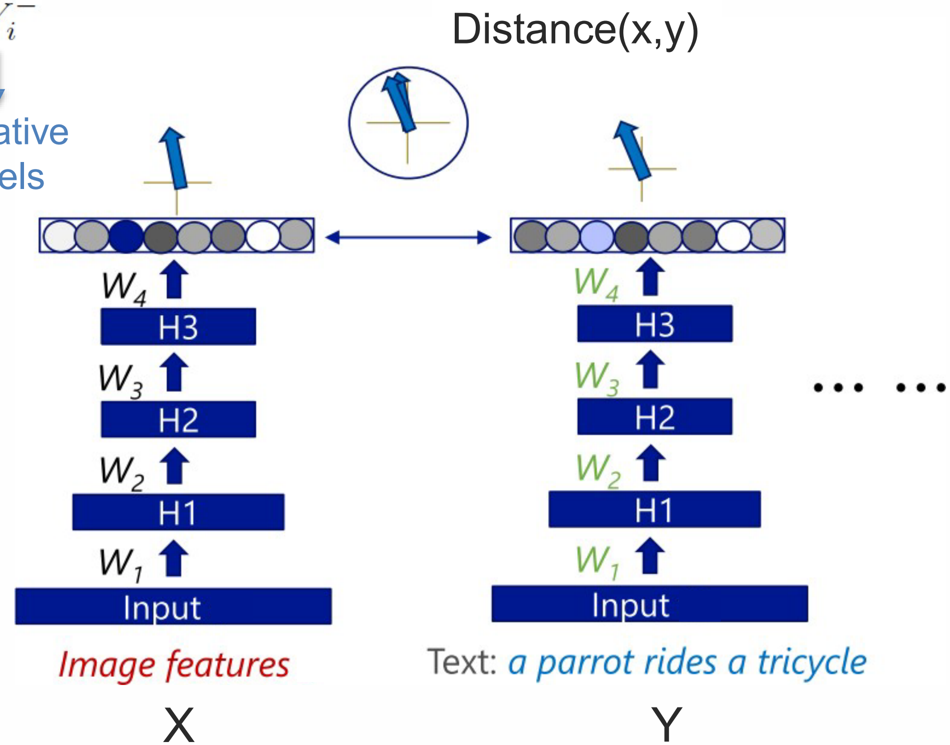
Max-margin:

$$d(x_i, y_j) + m < d(x_i, y_k) \quad \forall y_j \in Y_i^+, \forall y_k \in Y_i^-$$

Margin

Positive labels

Negative labels



[Frome et al., DeViSE: A Deep Visual-Semantic Embedding Model, NIPS 2013]

Structure-preserving Loss – Multimodal Embeddings

Symmetric max-margin:

$$d(x_i, y_j) + m < d(x_i, y_k) \quad \forall y_j \in Y_i^+, \forall y_k \in Y_i^-$$

$$d(x_{j'}, y_{i'}) + m < d(x_{k'}, y_{i'}) \quad \forall x_{j'} \in X_{i'}^+, \forall x_{k'} \in X_{i'}^-$$

+

Neighborhood of x_i:
images that share the same meaning (text)

Structure-preserving constraints

$$d(x_i, x_j) + m < d(x_i, x_k) \quad \forall x_j \in N(x_i), \forall x_k \notin N(x_i)$$

$$d(y_{i'}, y_{j'}) + m < d(y_{i'}, y_{k'}) \quad \forall y_{j'} \in N(y_{i'}), \forall y_{k'} \notin N(y_{i'})$$

[Wang et al., Learning Deep Structure-Preserving Image-Text Embeddings, CVPR 2016]

EXAMPLE:

AN ITERATIVE REFINEMENT APPROACH FOR SOCIAL MEDIA HEADLINE PREDICTION

ACM MULTIMEDIA 2019

Outline

- Introduction
- Proposed iterative refinement
  - Outlier detection
  - Refinement based on ensemble regressor
- Experimental results
- Conclusions

Outline

- Introduction
- Proposed iterative refinement
  - Outlier detection
  - Refinement based on ensemble regressor
- Experimental results
- Conclusions

Task

- View count prediction

Image from a post

Meta-data

Posted date
Comment count
Title Length
Description Length
Tags

User id
Follower
Group
Avg. view count
...

Evaluation metric
Mean Squared Error
Mean Absolute Error
Spearman Ranking Correlation

View count

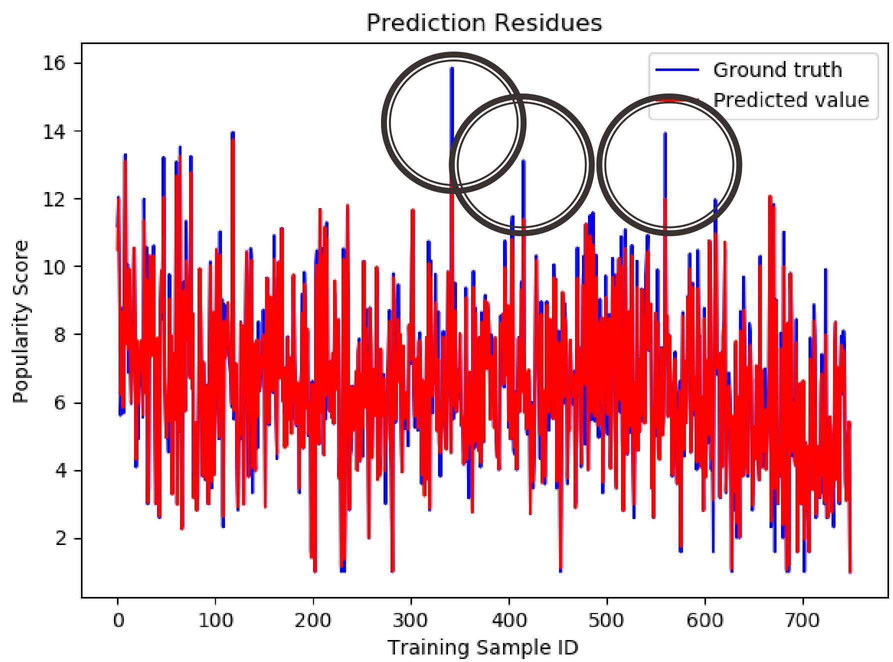
Overview

- Heterogeneous data
  - Image
  - Meta-data
    - Date, unique id, …etc
- We treat this task as regression problem
- Various regression models
  - Support vector regressor (SVR) [1]
  - Random forest regressor (RFR) [6]
  - Deep neural network regressor (DNNR) [5]

[1] Chih-Chung Chang and Chih-JenLin.2011. LIBSVM:alibraryforsupportvector machines. ACM transactions on intelligent systems and technology (TIST) 2, 3 (2011),27.
[5] YannLeCun,YoshuaBengio,andGeoffreyHinton.2015. Deep learning. Nature 521,7553(2015),436–444.
[6] AndyLiaw,MatthewWiener,etal.2002. Classification and regression by randomForest. R news 2,3(2002),18–22.

Overview

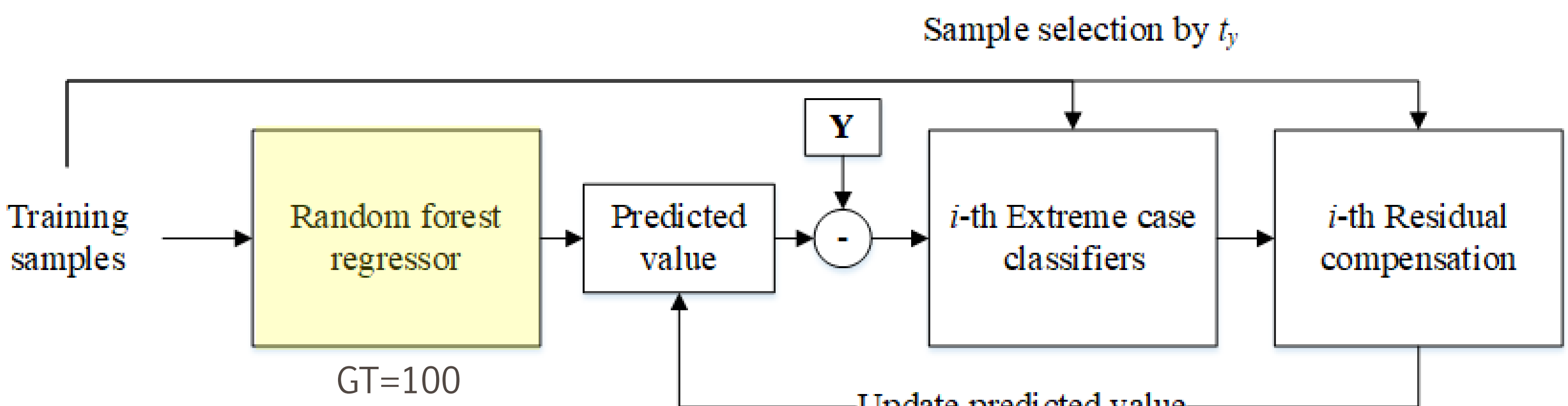
- It is well known that the most of regression methods fail to predict extreme values
  - The # pc
  - Leading
    - To ob   s tend to well
      predi   n the training set.
  - A single
- In this pap   regression
  model to o



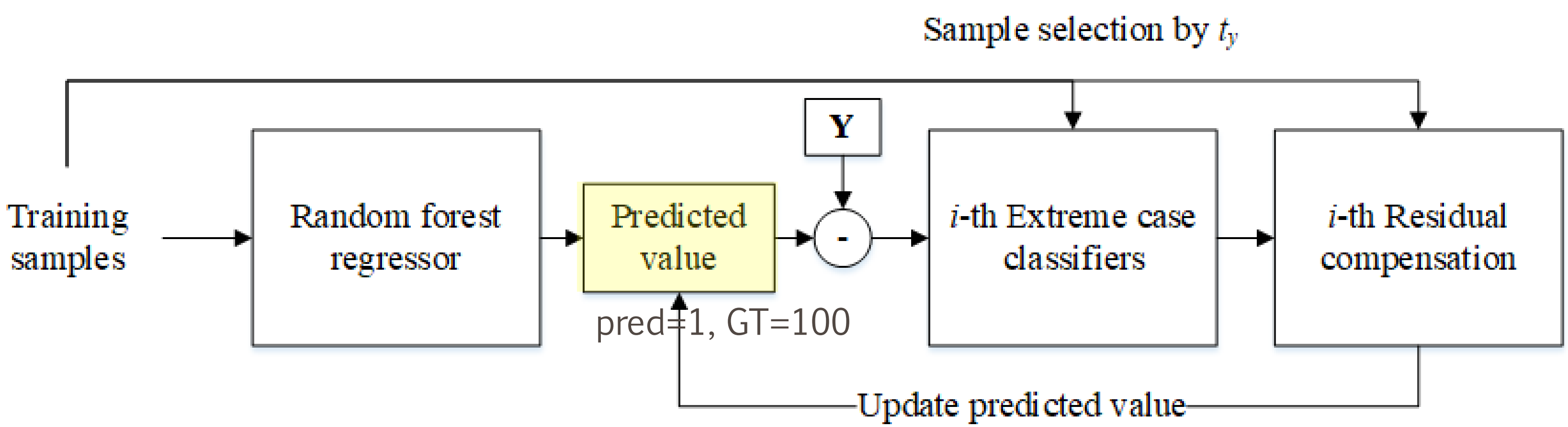
Outline

- Introduction
- Proposed iterative refinement
  - Outlier detection
  - Refinement based on ensemble regressor
- Experimental results
- Conclusions

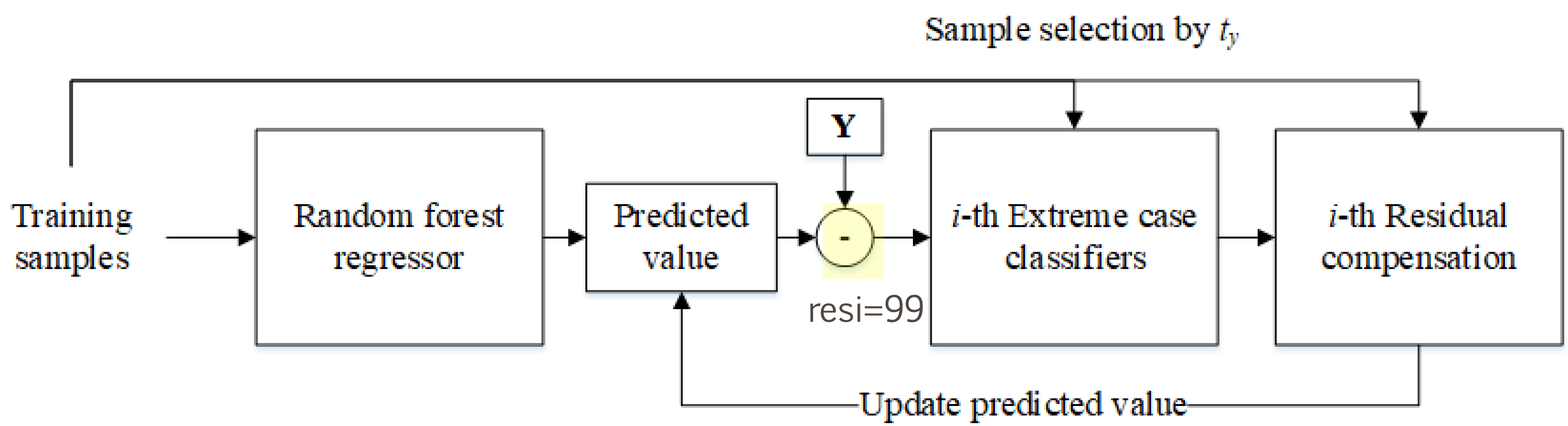
The Proposed Framework



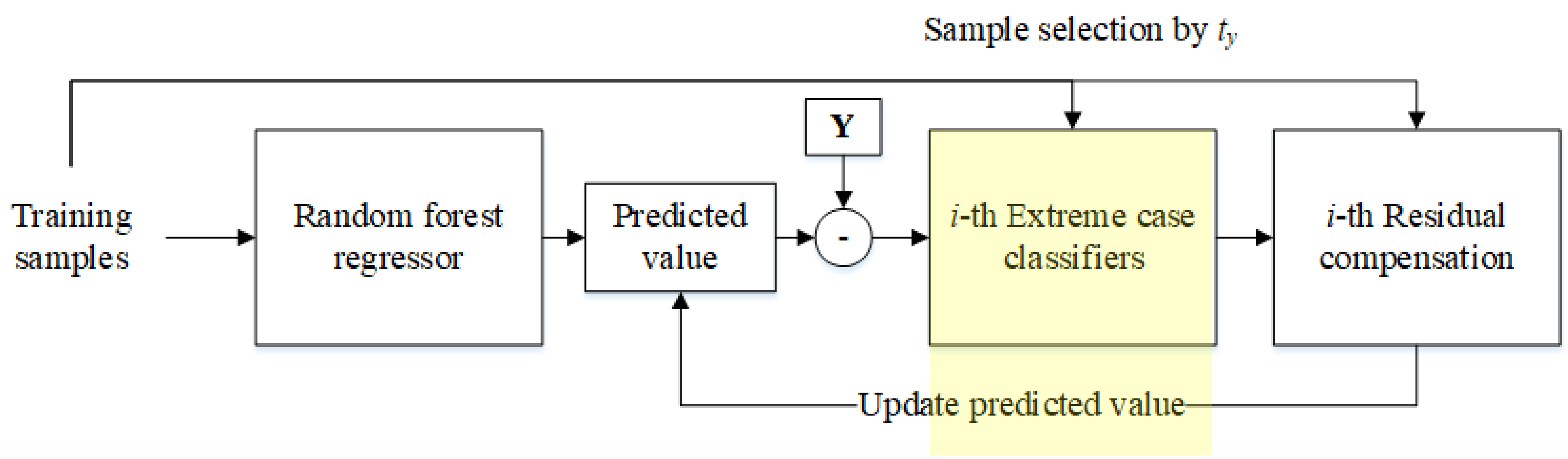
The Proposed Framework



The Proposed Framework

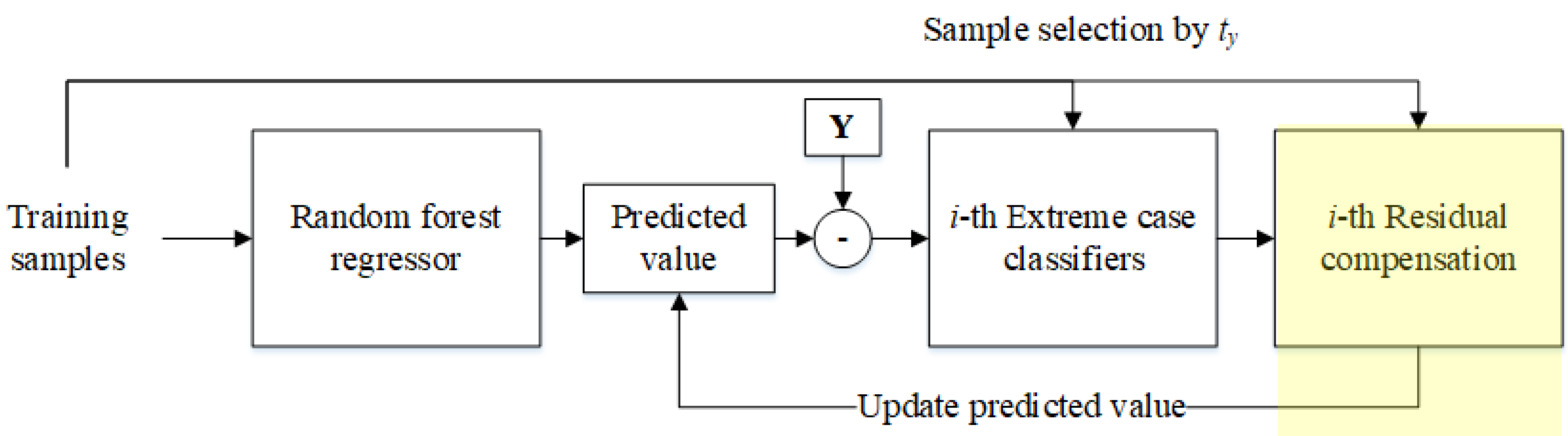


The Proposed Framework



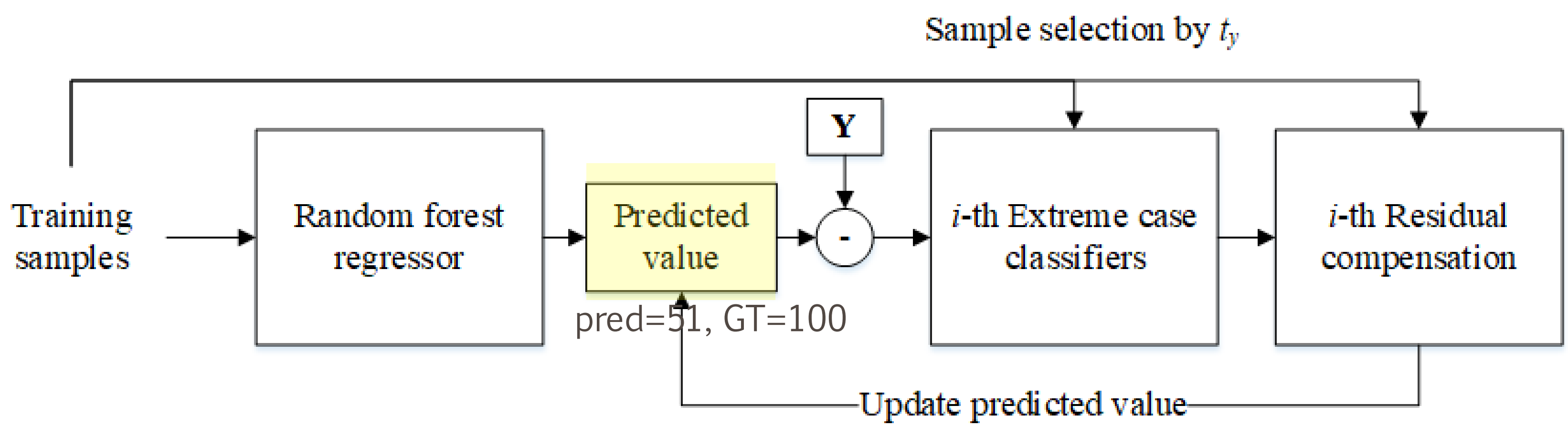
Is extreme case?
Yes!
Move forward

The Proposed Framework

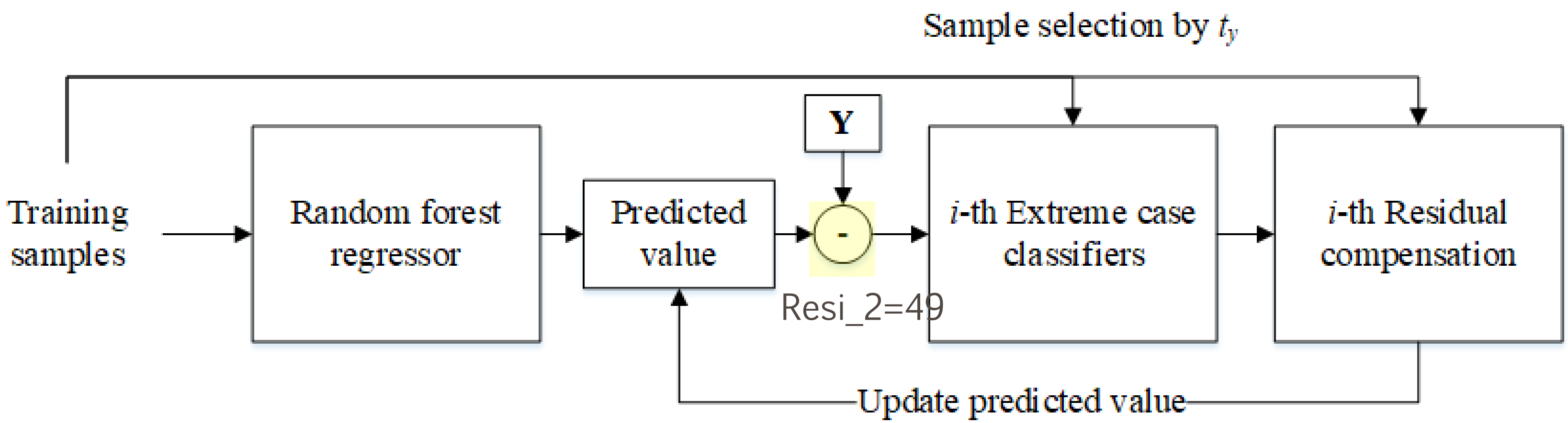


Resi_comp = 50
Pred_2 =resi+ Resi_cor
=51

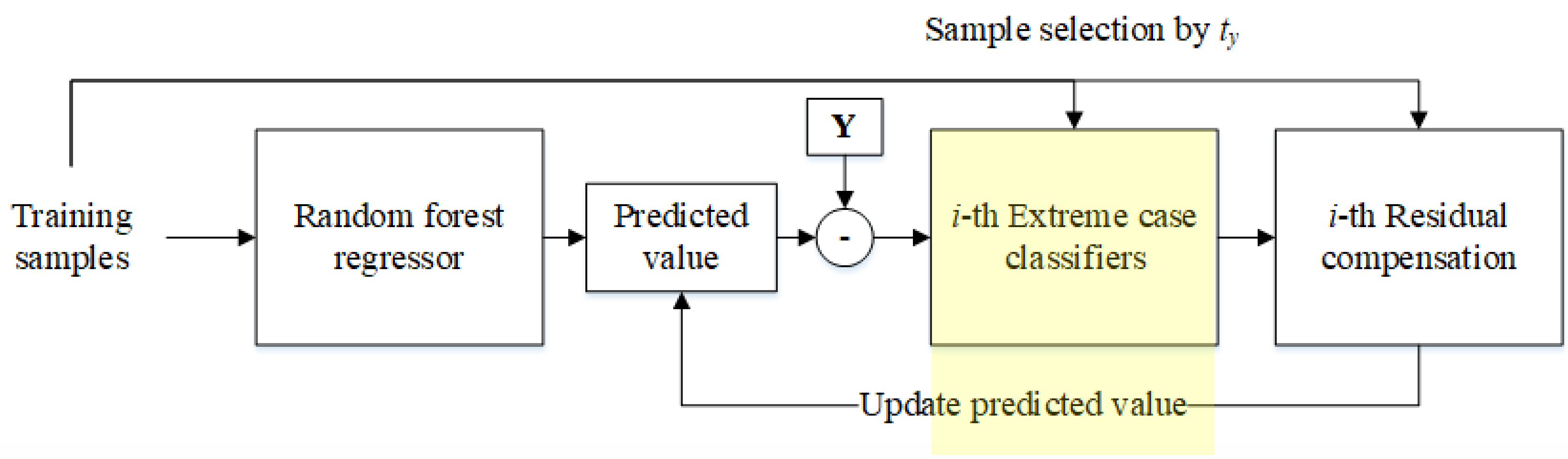
The Proposed Framework



The Proposed Framework

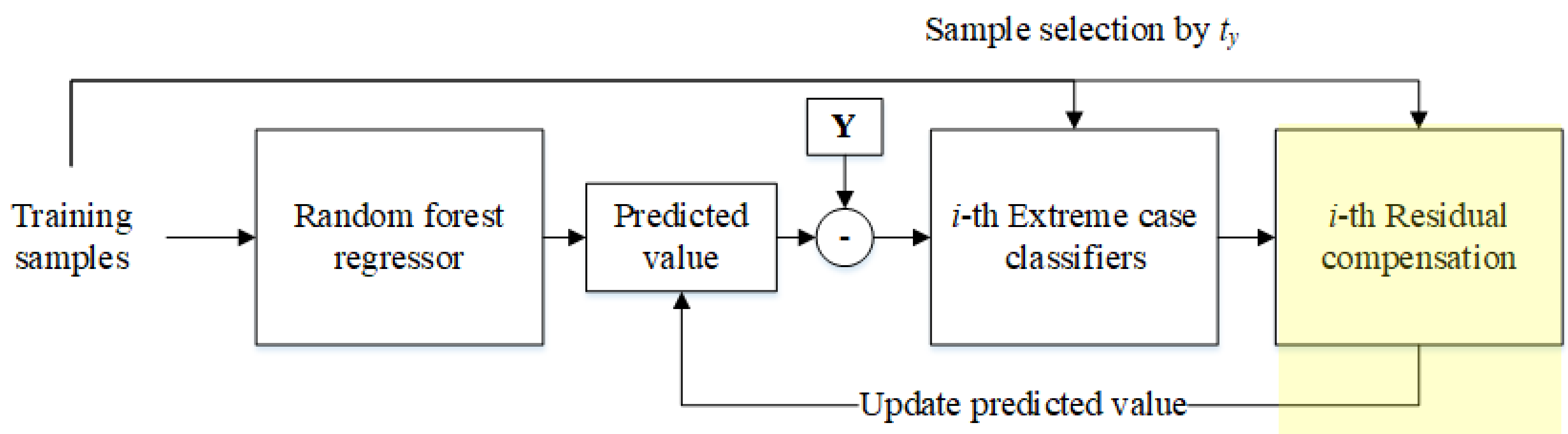


The Proposed Framework



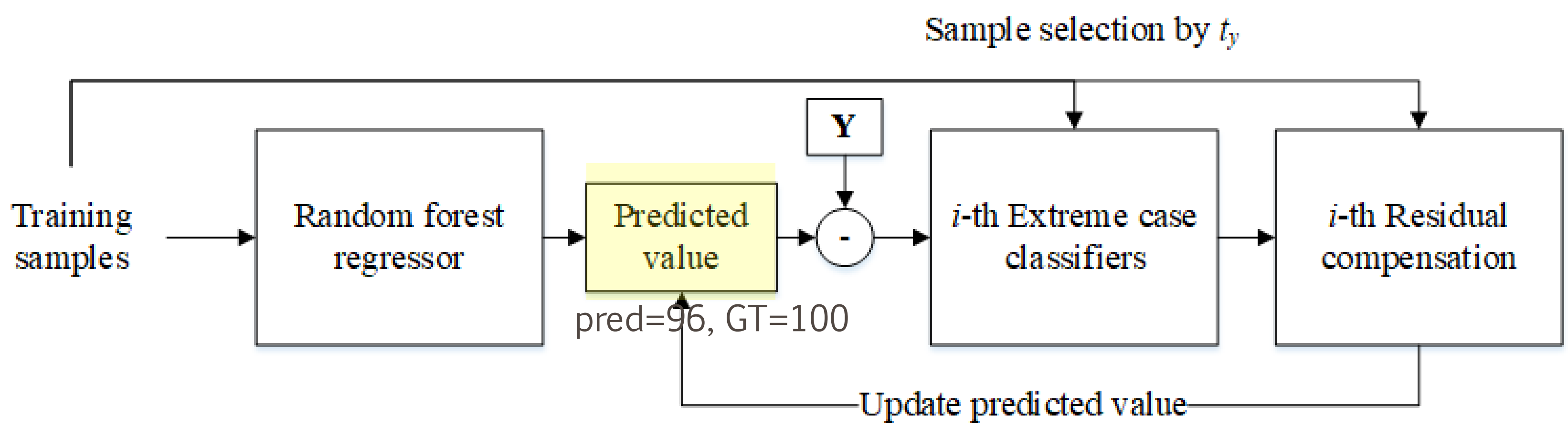
Is extreme case?
Yes!
Move forward

The Proposed Framework

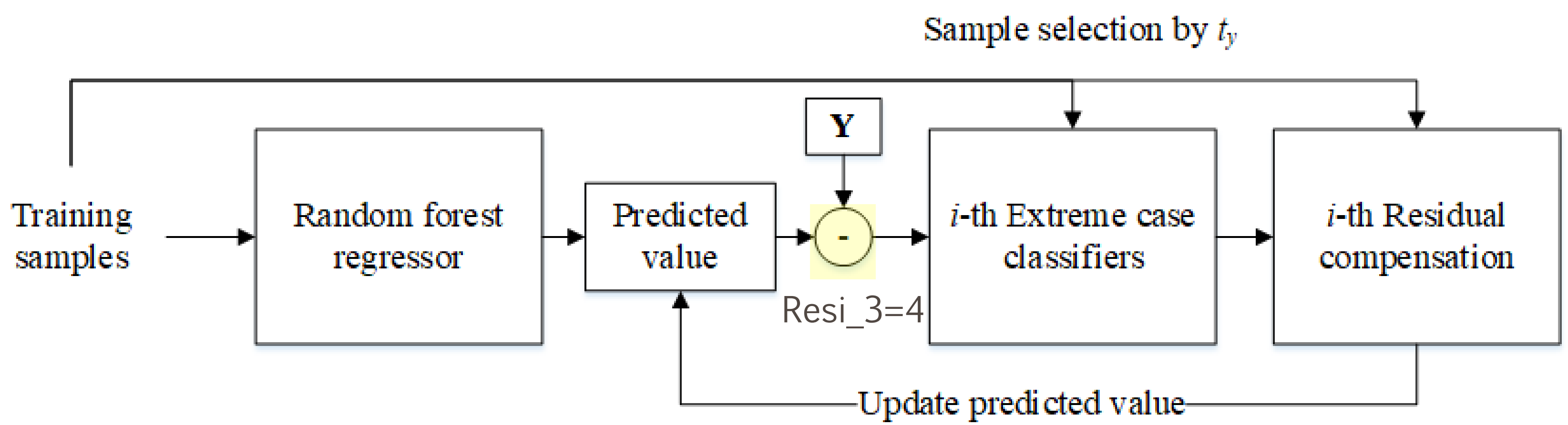


Resi_comp = 45
Pred_3 =resi_2+ Resi_c
=96

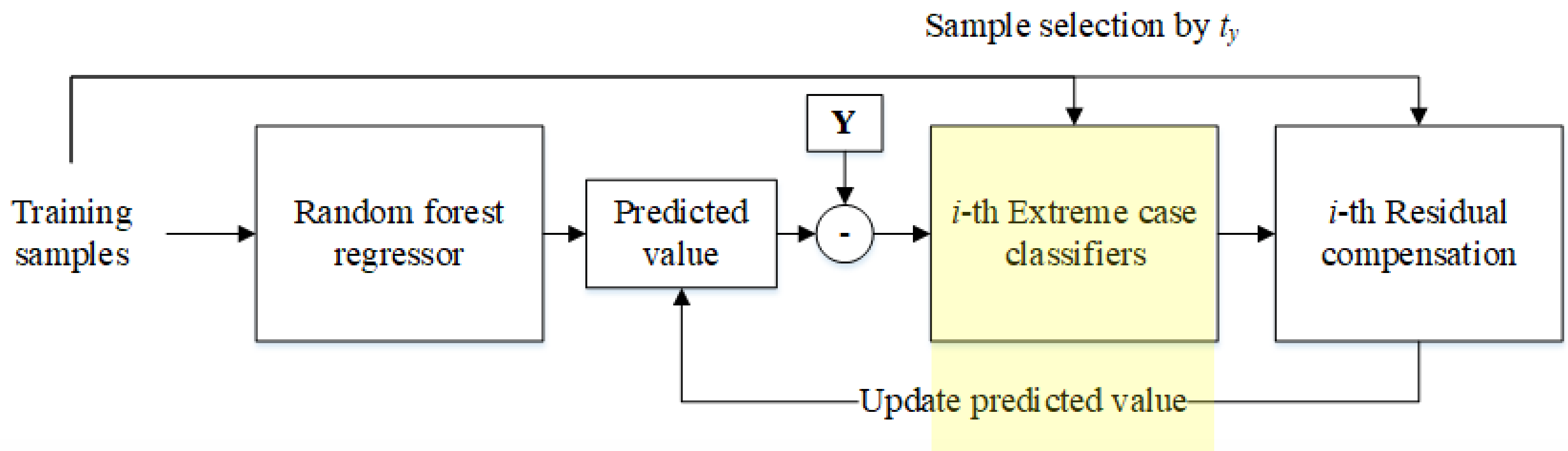
The Proposed Framework



The Proposed Framework



The Proposed Framework



Is extreme case?
No!
Stop here!

Outlier Detection

- Since some extreme values are hard to predict, we tend to detect those extreme values at first.
  - We first design a classifier

$$g(\mathrm{X}_s) = C(\mathrm{X}_s, |\theta_s),$$

    - g(Xs) indicates either -1 (non-extreme value) or 1 (extreme value).

$$L(\mathrm{X}_s) = \sum_{i=0}^{N} l(C(\mathrm{X}_s, \mathrm{R})),$$

  - The loss function can be defined as

$$L(\mathrm{X}_s) = \sum_{i=0}^{N} l(C(\mathrm{X}_s, \mathrm{R}_t)),$$

  - However, the residual R is not a binary class data, leading to learning difficulty
    - We predefine a threshold value t to partition R into two class data (extreme & non-extreme)

Iterative Refinement Approach

- For the predicted values along to extreme class
  - Refine them by another regressor

$$P_{s_i} = R_i + P_{s_{i-1}} = h_i(X_{R_i}, \theta_i) + h_{i-1}(X_{R_i}, \theta_{i-1}),$$

where X_{R_i} will be X_s at iteration 0 and $X_{R_i} = [X_s | g_i(X_s) = 1]$.

  - Given parameter k, the i_{th} regressor h_i can be used to compensate $(i\text{-}1)_{th}$ predicted value
  - The size of **R** will be reduced iteratively
  - Each regressor can have its own parameter setting
    - Called ensemble regressor
- In this paper, the classifier and regressor are adopt AdaBoosting and Random Forest respectively.

Outline

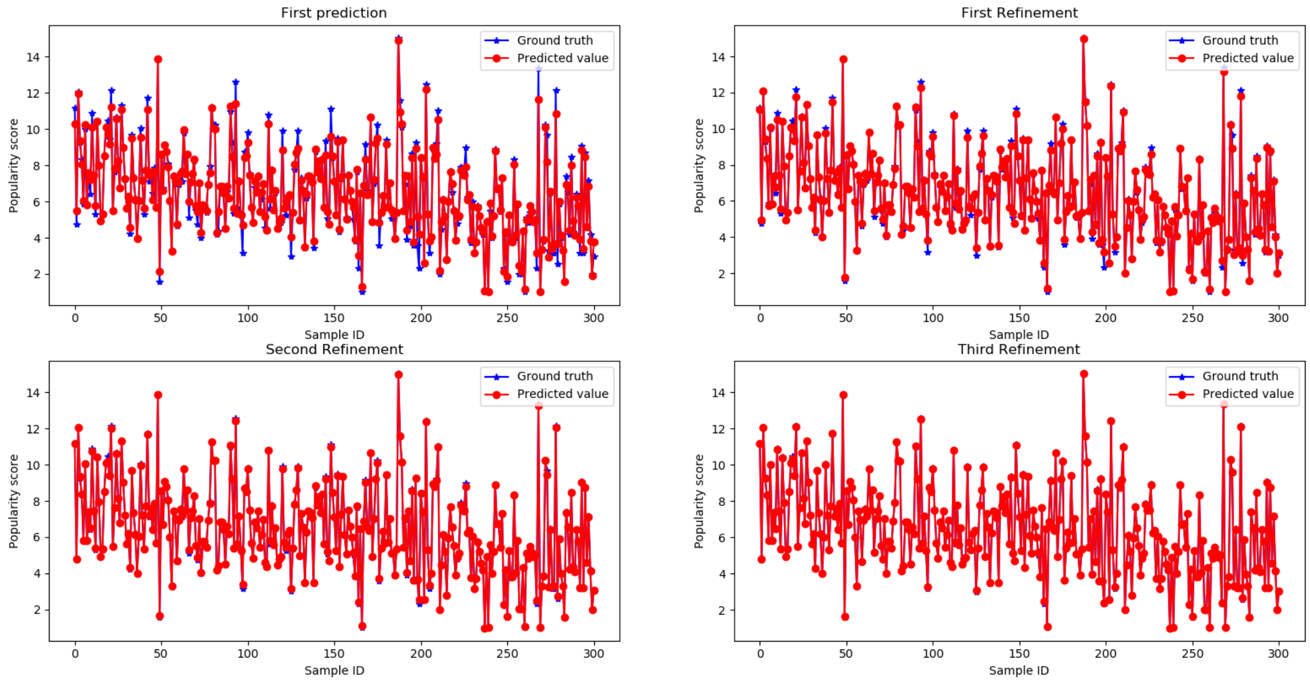
- Introduction
- Proposed iterative refinement
  - Outlier detection
  - Refinement based on ensemble regressor
- Experimental results
- Conclusions

Experimental results

- Social media headline prediction challenge dataset (SMHPD)
  - 305, 614 posts
  - 300, 000 training samples and 5, 614 test samples
- Two experimental settings
  - Training set and test set are partitioned based on time-order
  - Training set and test set are partitioned randomly
- Evaluation metric
  - Mean Squared Error
  - Mean Absolute Error
  - Spearman Ranking Correlation

[10] BoWu,Wen-HuangCheng,YongdongZhang,andTaoMei.2016. TimeMatters: Multi-scaleTemporalizationofSocialMediaPopularity.InProceedingsofthe2016 ACM on Multimedia Conference (ACMMM)
[11] BoWu,Wen-HuangCheng,YongdongZhang,HuangQiushi,LiJintao,andTao Mei.2017. SequentialPredictionofSocialMediaPopularitywithDeepTemporal Context Networks. In International Joint Conference on Artificial Intelligence (IJCAI).
[12] Bo Wu, Tao Mei, Wen-Huang Cheng, and Yongdong Zhang. 2016. Unfolding TemporalDynamics:PredictingSocialMediaPopularityUsingMulti-scaleTemporalDecomposition.InProceedingsoftheThirtiethAAAIConferenceonArtificial Intelligence (AAAI).

Experimental Results



Experimental results

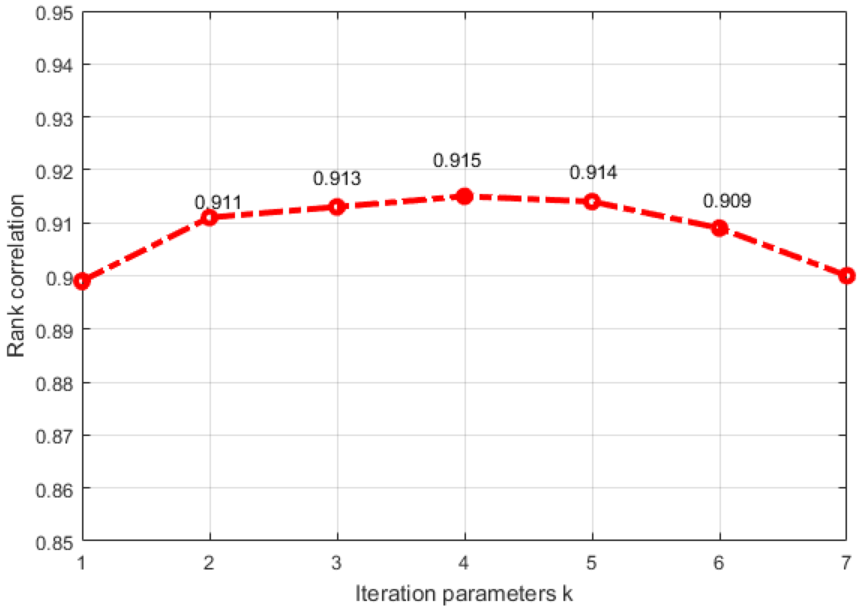
- Two different experimental settings
  - Right: Training data split by time order
  - Left: Training data split by randomly processing

Methods	Rank correlation	MSE	MAE
Naive Bayer Regressor	0.312	7.595	2.107
SVR	0.351	5.411	1.846
Linear Regression	0.423	5.068	1.785
AdaBoosting Regression	0.883	1.442	0.671
Random Forest	0.886	1.415	0.662
Multi-model Approach [?]	0.901	1.283	0.630
Proposed method	**0.919**	**1.185**	**0.593**

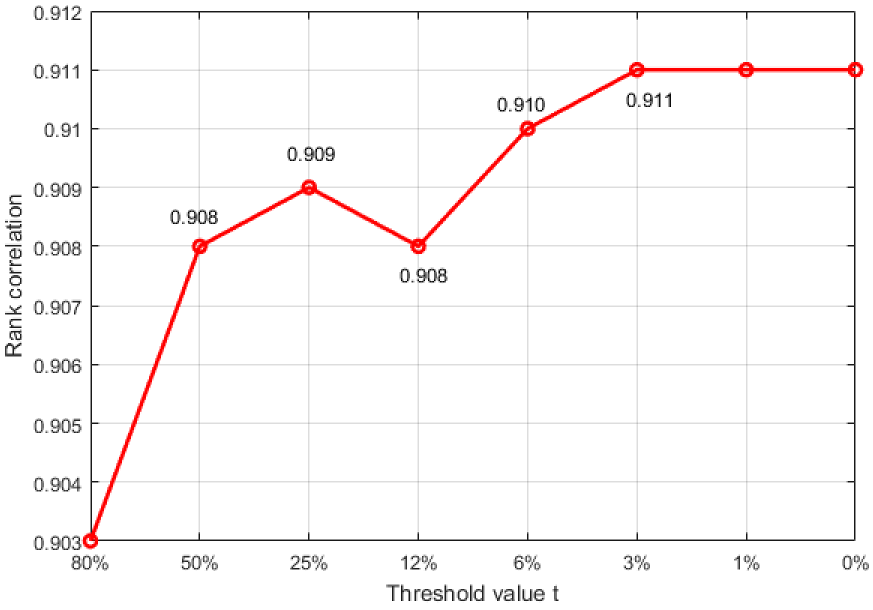
Methods	Rank correlation	MSE	MAE
Naive Bayer Regressor	0.417	5.196	1.814
SVR	0.441	4.999	1.769
Linear Regression	0.424	5.186	1.803
AdaBoosting Regression	0.594	3.967	1.541
Random Forest	0.886	1.418	0.663
Multi-model Approach [?]	0.846	1.838	0.748
Proposed method	**0.908**	**1.193**	**0.600**

[?] Our previous method for social media prediction last year.

Selection of Parameter k and t



Parameter k



Parameter t

INCORPORATING IMAGES

Popularity Prediction for a Post

- Given a post with **heterogeneous** data, predict the “view count”

A post in Flickr, Facebook, Pinterest, Instagram, Twitter...

Description (text)

Tags

Numerical data (EXIF)

Image

View count!!

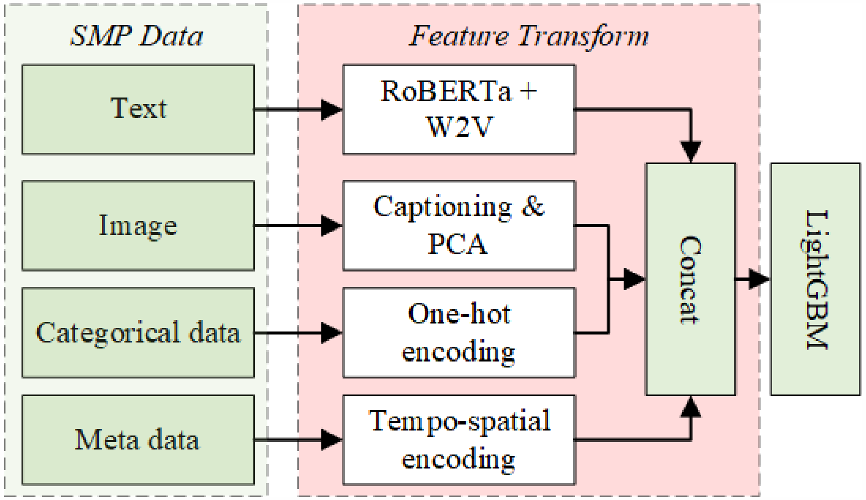
Training set we have

Baseline Model and Stacking

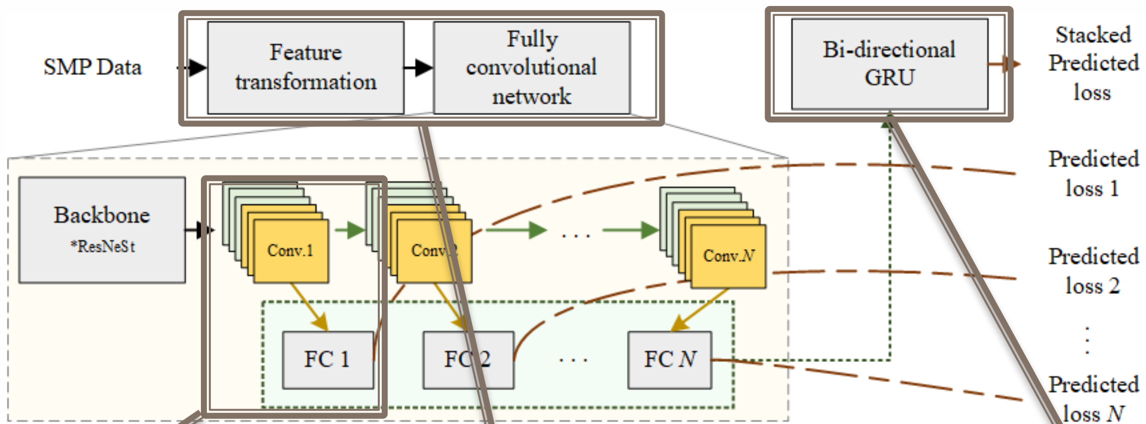
- Feature importance
  - 1st group: text-type such as description, tags
  - 2nd group: numerical data such as meta-data
  - 3rd group: image data (ResNet-50 feature) [Implying we did not work it well]
- Enhanced text-type feature representation
  - W2V model is insufficient for complicated context
  - We adopt RoBERTa to extract text-type feature instead
- Model stacking
  - Combining multiple complex models with different data partitions
  - Very time consuming!!
- We propose a novel Recurrent unit-based Stacking Model (RSM)
  - Only one model is all you need
  - Efficient and Effective

Baseline Model

- Based on our previous one, we have added ToBERTa to extract more meaningful information from text



Recurrent-based Stacking Model



Partial features only
[Feature partition]

Batch-based input
[Data partition]

Prediction stacking over different sub-model
[Model stacking]

Results and Conclusion

- Model stacking seems still to be powerful
- Our RSM shows good performance!
- Our RSM
  - Faster for training
    - Model stacking
      - 44 hours
    - RSM
      - 3.5 hours only

Table 1: Performance comparison among the different regression methods evaluated on the testing set.

Methods	SRC	MSE	MAE
Baseline-I	0.448	7.595	2.107
Baseline-II	0.450	5.411	1.846
Baseline-III	0.461	5.068	1.785
Baseline-IV	0.470	5.442	1.871
MM [5]	0.528	5.891	1.942
IR [6]	0.537	5.872	1.939
EW [21]	0.548	5.856	1.938
MMF [13]	0.656	3.561	1.497
Proposed baseline	0.704	3.216	1.417
Proposed + Model stacking	0.765	**2.916**	**1.345**
Proposed RSM	**0.774**	2.933	1.361