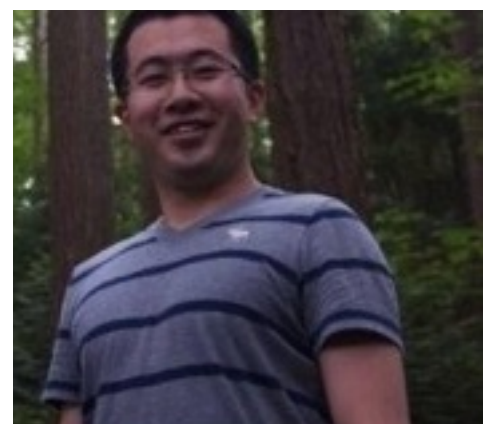
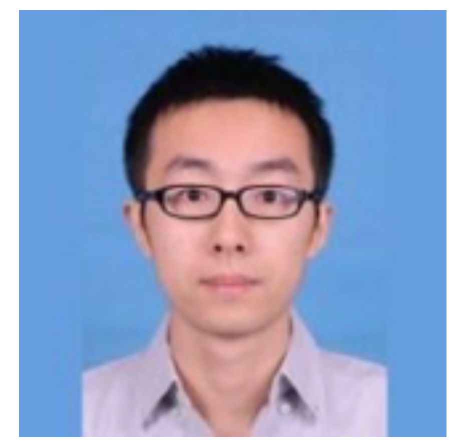
UIC Department of Computer Science
UNIVERSITY OF ILLINOIS AT CHICAGO
COLLEGE OF ENGINEERING

Multilinear Factorization Machines for Multi-Task Multi-View Learning

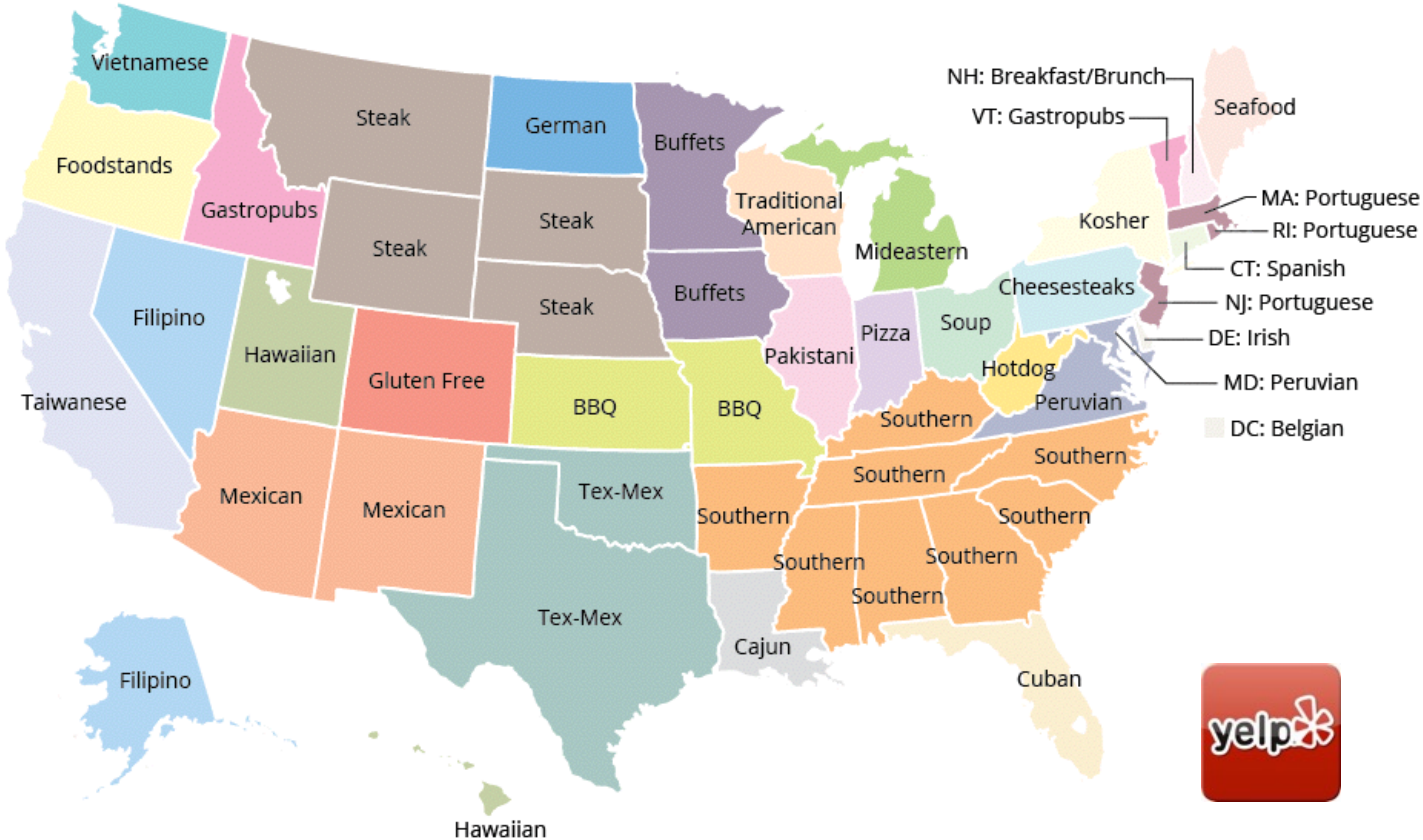



Chun-Ta Lu Lifang He Weixiang Shao Bokai Cao Philip S. Yu

University of Illinois at Chicago
Presenter: Chun-Ta Lu

Example of Multiple Tasks



Recommendation in different cities

Source: http://www.huffingtonpost.com/2015/01/14/most-popular-cuisine-state_n_6457252.html

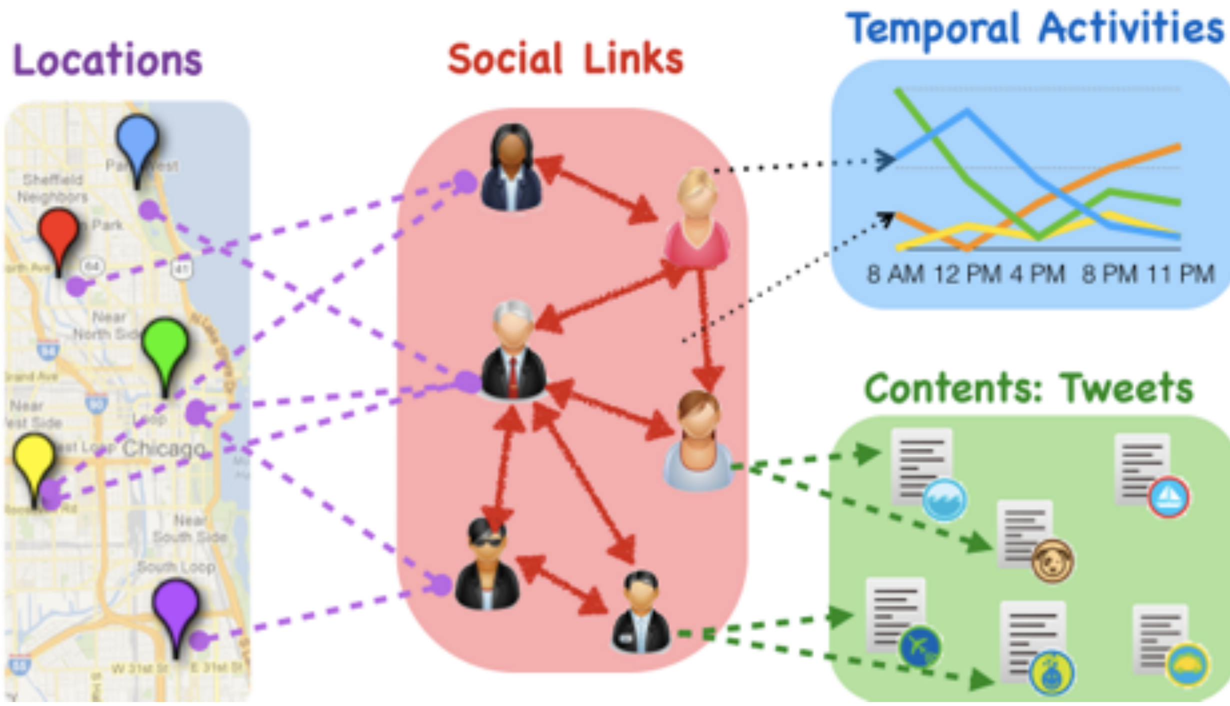
Example of Multiple Tasks

Human? **Dog?** **Male?** **Female?**

Each label as a task

http://www.cs.cornell.edu/~kilian/research/multitasklearning/multitasklearning.html

Example of multiple views



Online Social Network

Social Links

Textual contents

Checkin histories

Temporal activities

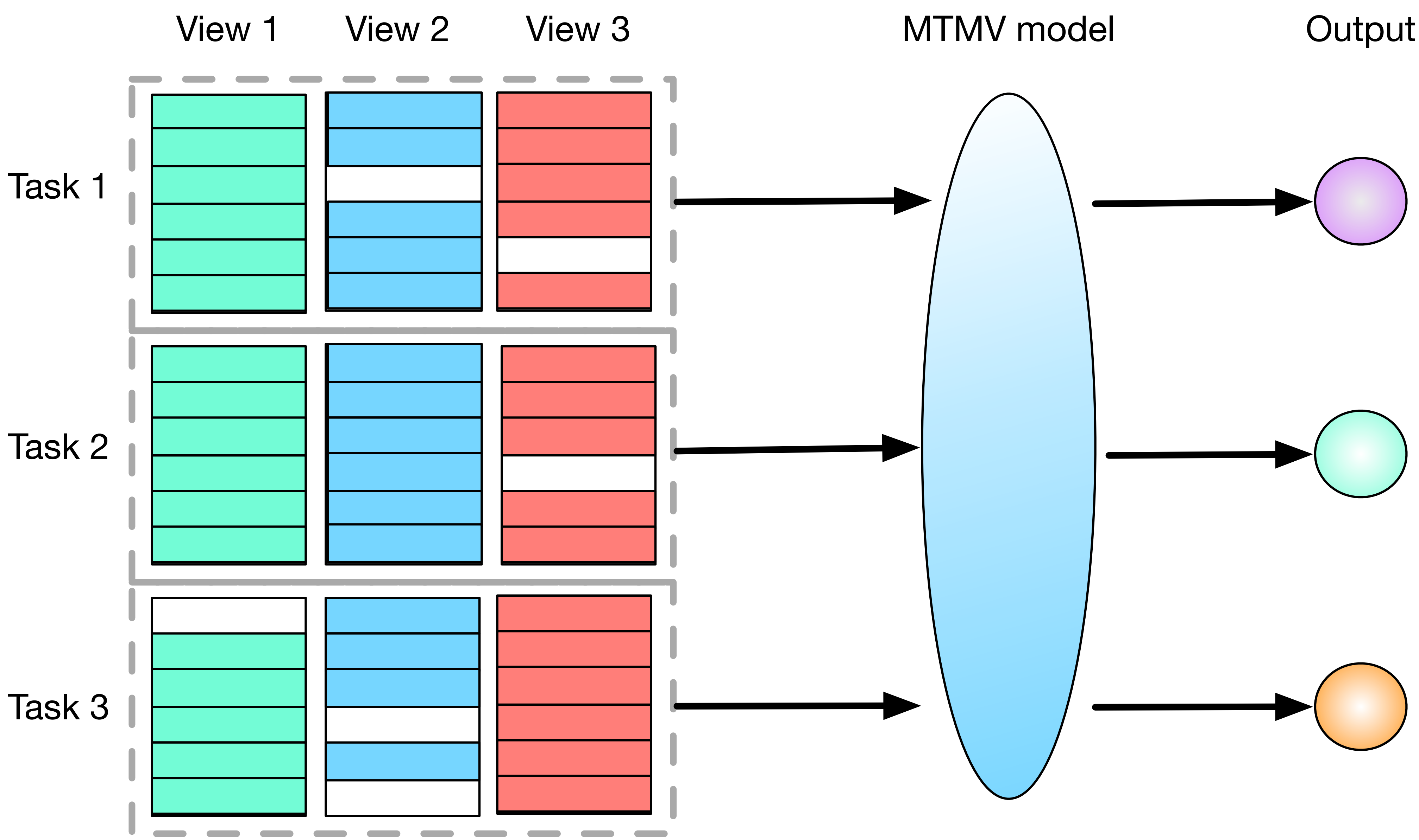
Web images on Instagram

Visual information

Textual tags

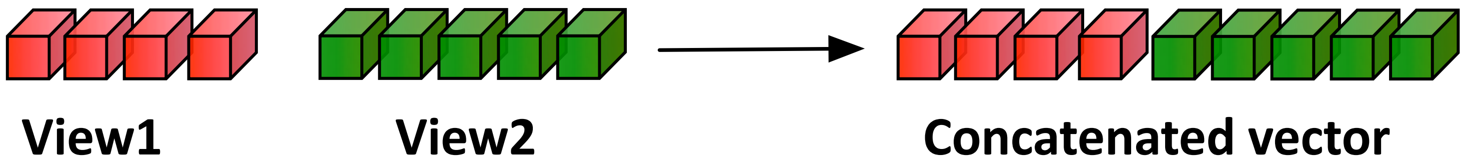
Multi-Task Multi-View Learning

Combine different views (e.g., images and texts) to learn multiple related tasks together.

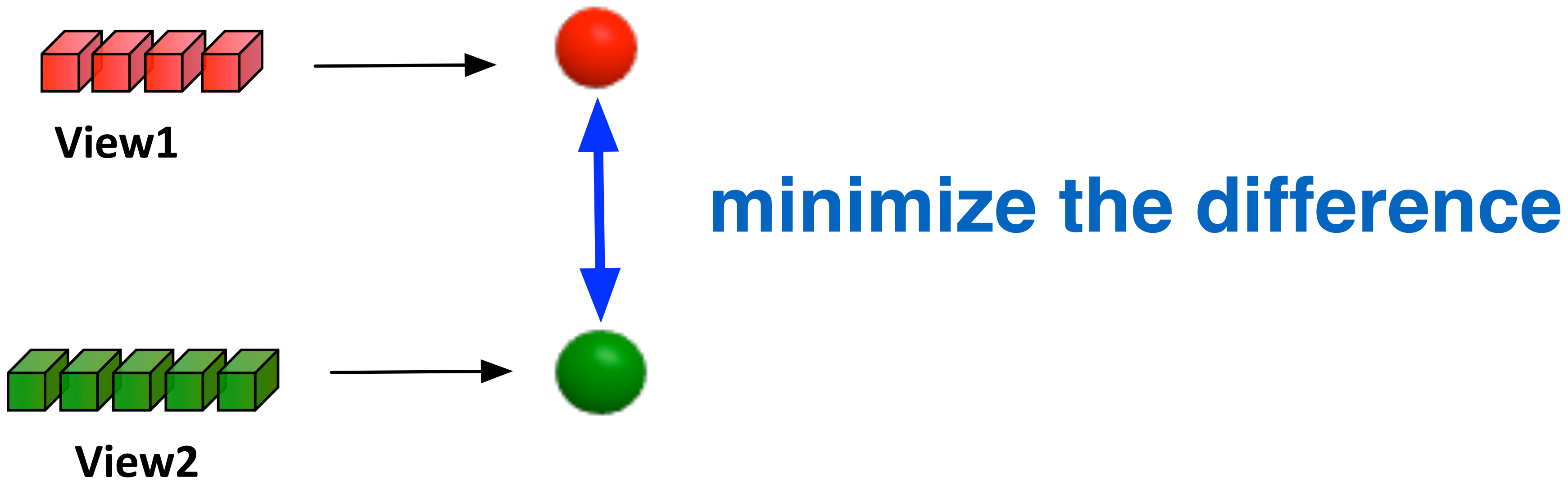


Traditional Approaches

- **Concatenate features from all the views as a single vector?**

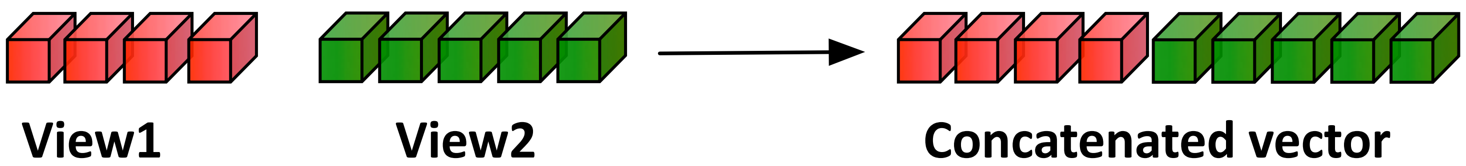


- **Each view provides a prediction, and minimize the difference?**

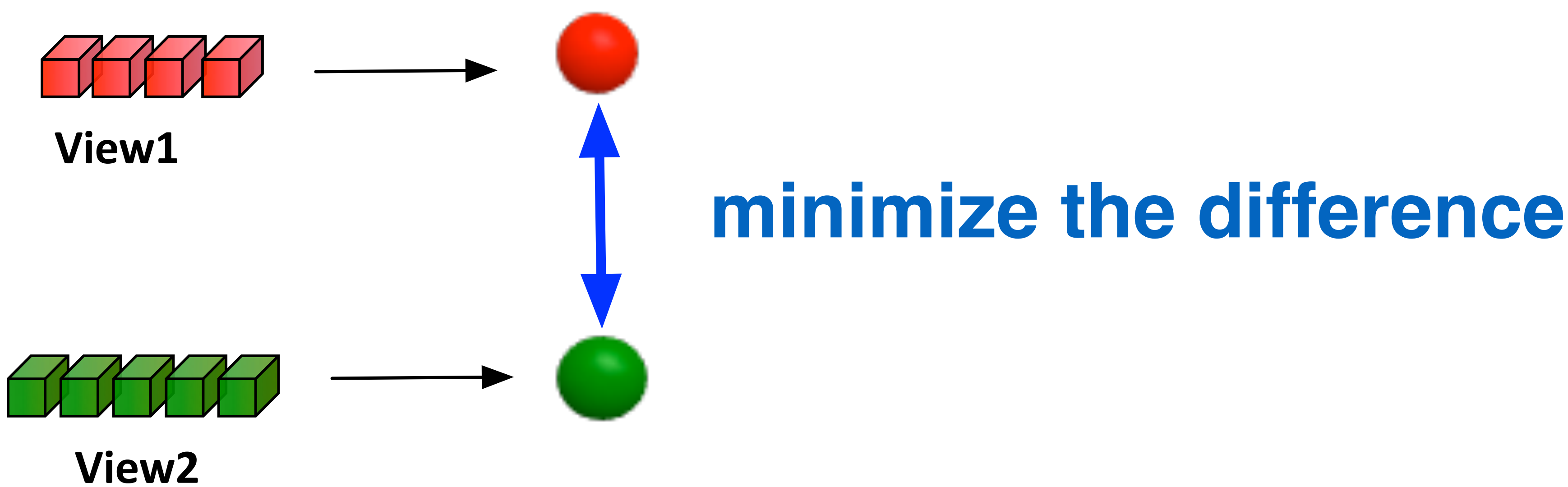


Traditional Approaches

- **Concatenate features from all the views as a single vector?**



- **Each view provides a prediction, and minimize the difference?**

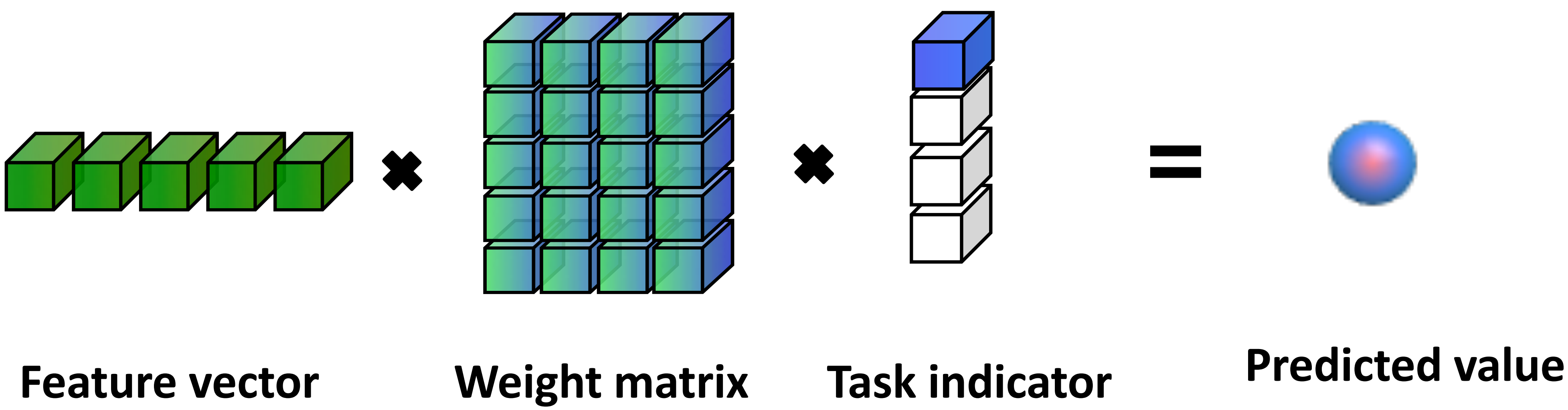


- **Different views provide complementary information.**
- **Loss important feature interactions**

Multilinear Predictive Models

Observation: MTL w/ single view is to learn a bilinear map

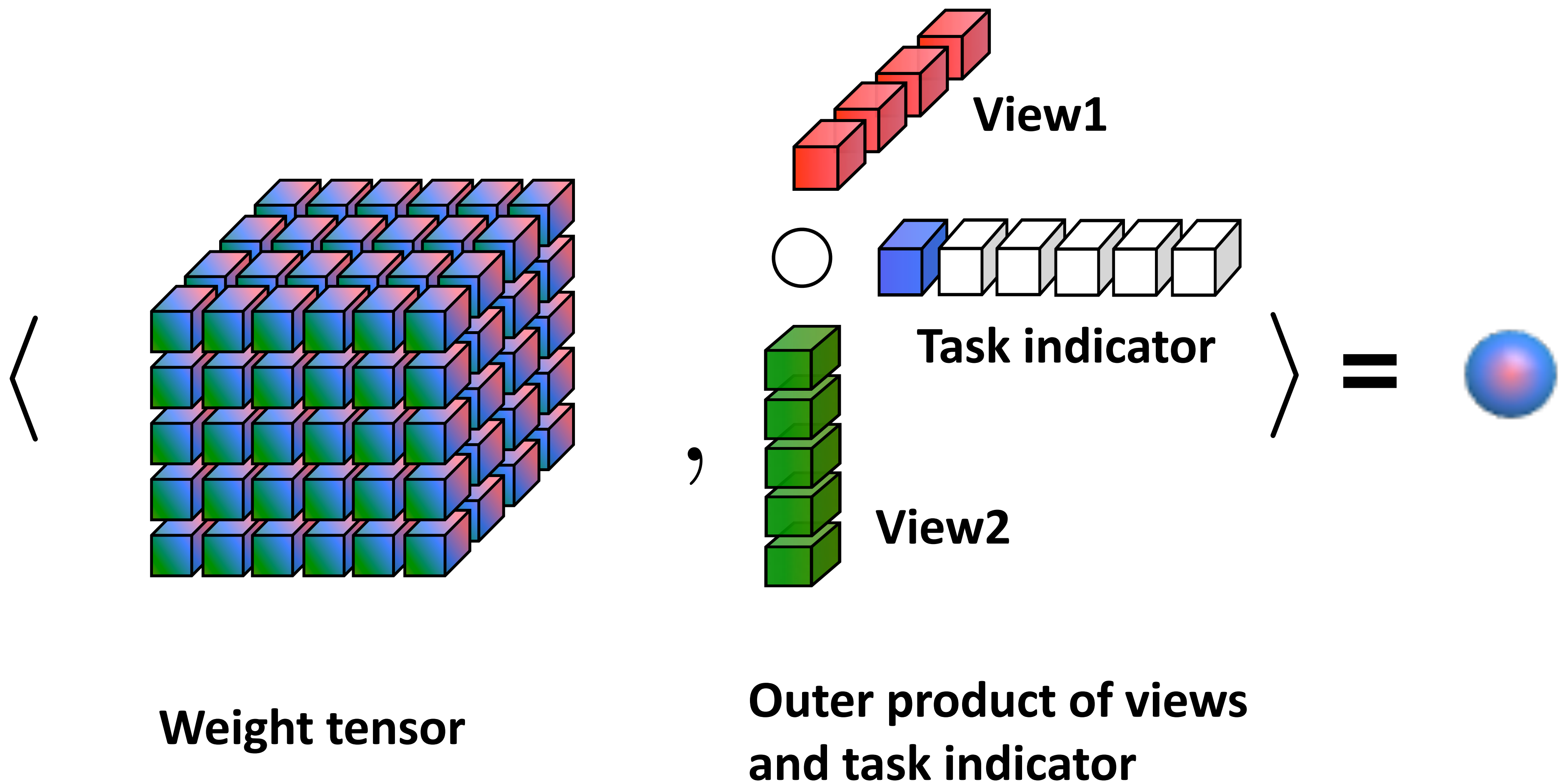
Define the task indicator $\mathbf{e}_t = [\underbrace{0, \cdots, 0}_{\text{t-1}}, 1, 0, \cdots, 0]^{\mathrm{T}}$



$$f_t(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{w}_t = \mathbf{x}^{\mathrm{T}}\mathbf{W}\mathbf{e}_t = \langle \mathbf{W}, \mathbf{x} \circ \mathbf{e}_t \rangle = f(\{\mathbf{x}, \mathbf{e}_t\})$$

Multilinear Predictive Models

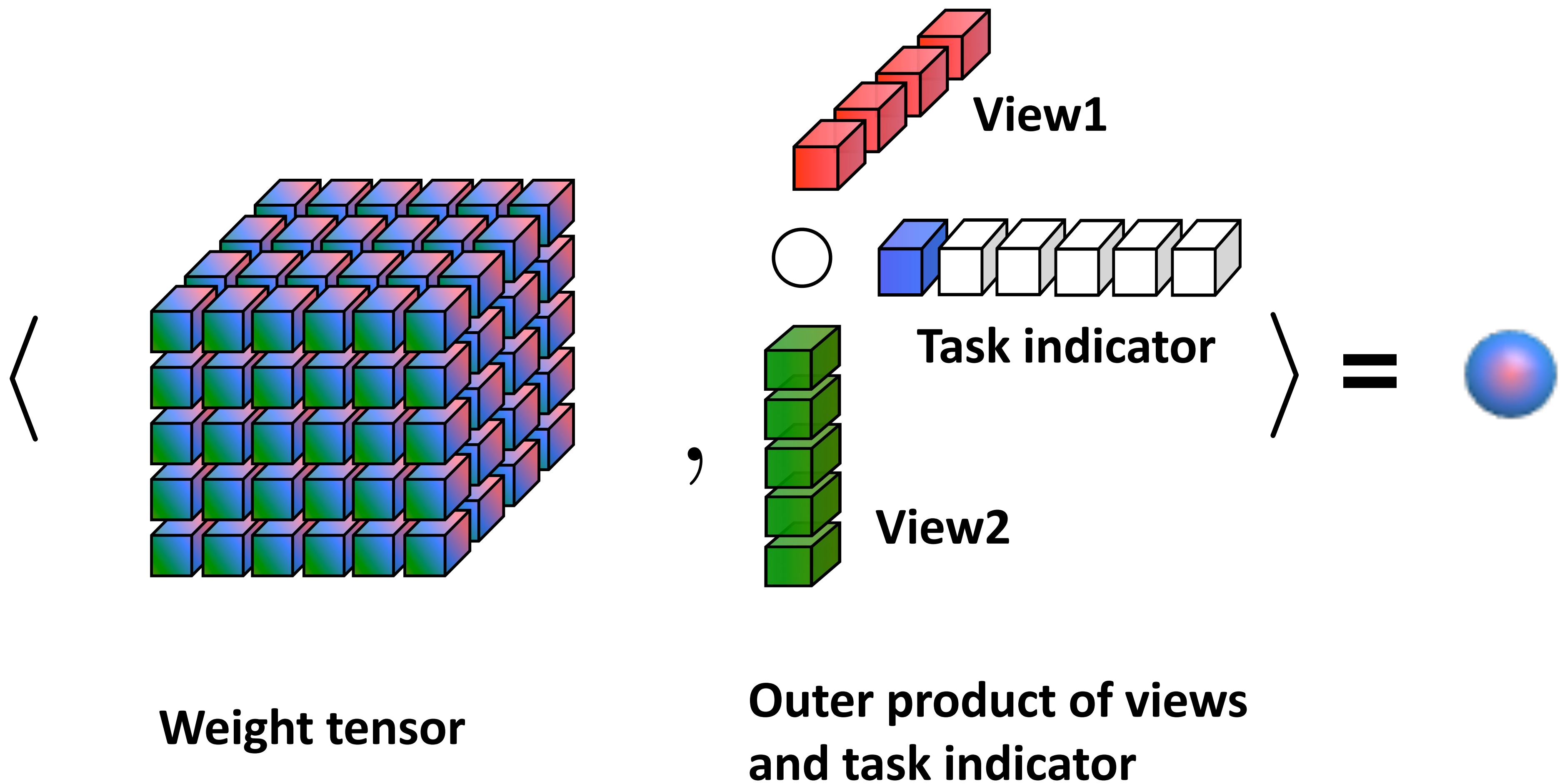
Extend to MTMV learning is to learn a multilinear map



$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = \mathbf{x}^{(1)^{\mathrm{T}}} \mathbf{W}_t \mathbf{x}^{(2)} = \left\langle \mathcal{W}, \mathbf{x}^{(1)} \circ \mathbf{x}^{(2)} \circ \mathbf{e}_t \right\rangle$$

Multilinear Predictive Models

Extend to MTMV learning is to learn a multilinear map

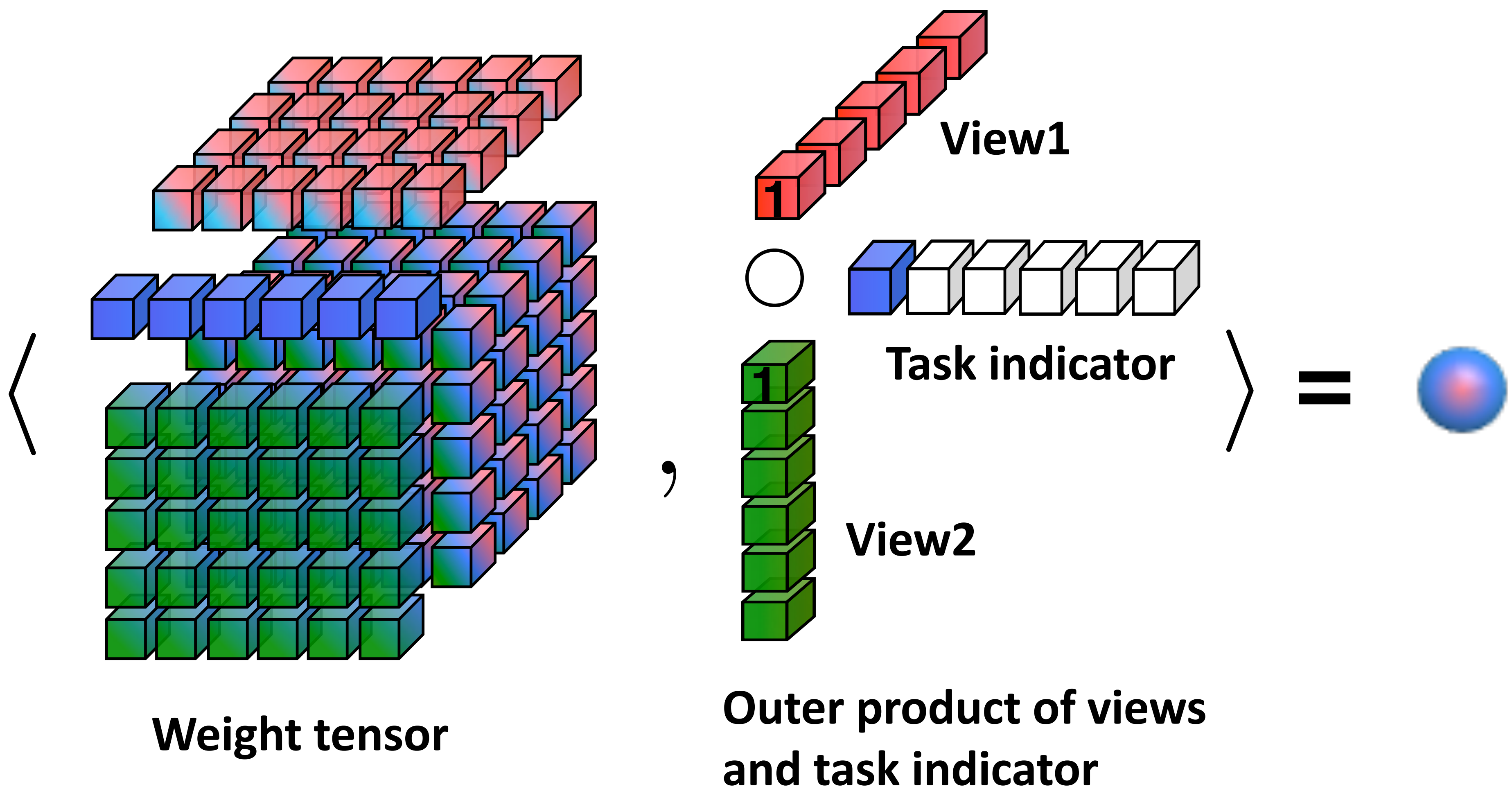


$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = \mathbf{x}^{(1)^{\mathrm{T}}} \mathbf{W}_t \mathbf{x}^{(2)} = \left\langle \mathcal{W}, \mathbf{x}^{(1)} \circ \mathbf{x}^{(2)} \circ \mathbf{e}_t \right\rangle$$

Cannot deal with incomplete view

Multilinear Predictive Models

Nesting all interactions up to full-order

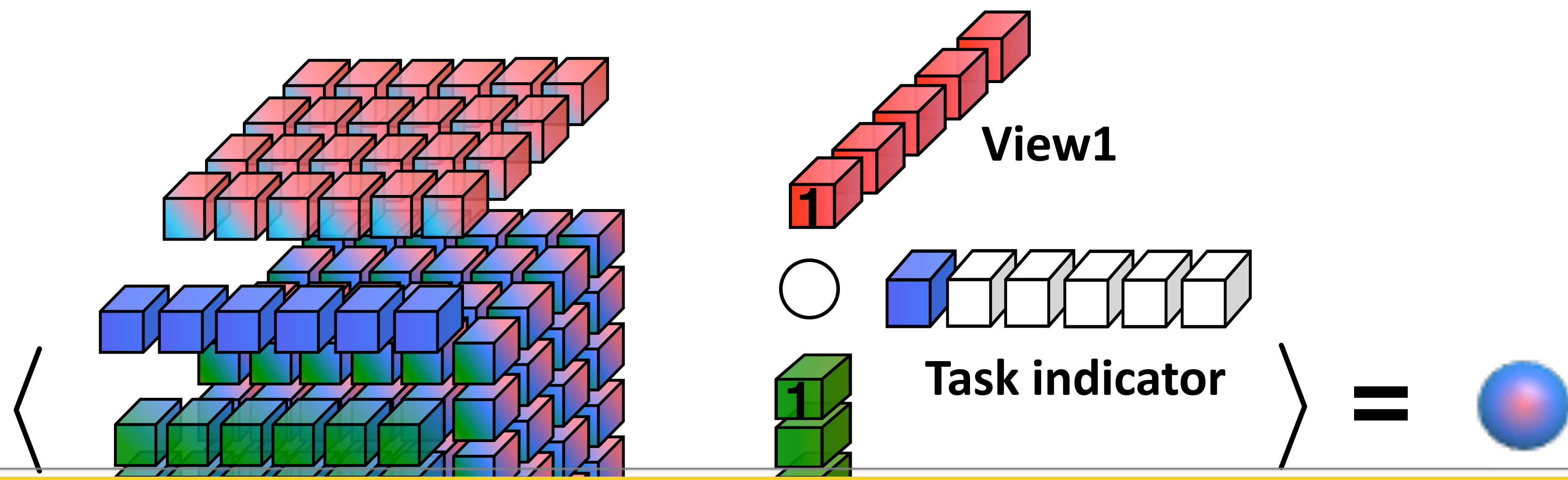


$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = w_t + \sum_{v=1}^{2} \mathbf{x}^{(v)^{\mathrm{T}}} \mathbf{w}_t^{(v)} + \mathbf{x}^{(1)^{\mathrm{T}}} \mathbf{W}_t \mathbf{x}^{(2)} = \left\langle \mathcal{W}, [1; \mathbf{x}^{(1)}] \circ [1; \mathbf{x}^{(2)}] \circ \mathbf{e}_t \right\rangle$$

$$= \left\langle \mathcal{W}, \mathbf{z}^{(1)} \circ \mathbf{z}^{(2)} \circ \mathbf{e}_t \right\rangle = \langle \mathcal{W}, \mathcal{Z}_t \rangle$$

Multilinear Predictive Models

Nesting all interactions up to full-order



Not realistic to learn the weight tensor directly, since #parameters grows exponential to #features.

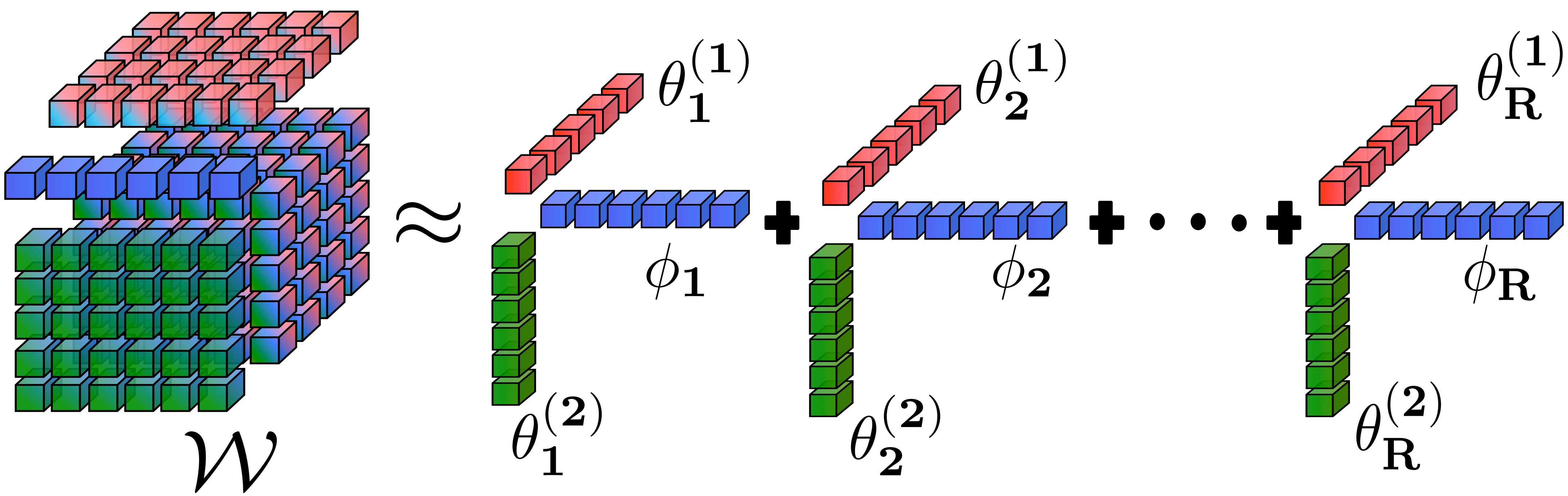
High-dimensional parameters are learned independently

$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = w_t + \sum_{v=1}^{2} \mathbf{x}^{(v)^{\mathrm{T}}} \mathbf{w}_t^{(v)} + \mathbf{x}^{(1)^{\mathrm{T}}} \mathbf{W}_t \mathbf{x}^{(2)} = \left\langle \mathcal{W}, [1; \mathbf{x}^{(1)}] \circ [1; \mathbf{x}^{(2)}] \circ \mathbf{e}_t \right\rangle$$

$$= \left\langle \mathcal{W}, \mathbf{z}^{(1)} \circ \mathbf{z}^{(2)} \circ \mathbf{e}_t \right\rangle = \langle \mathcal{W}, \mathcal{Z}_t \rangle$$

Multilinear Factorization Machines (MFMs)

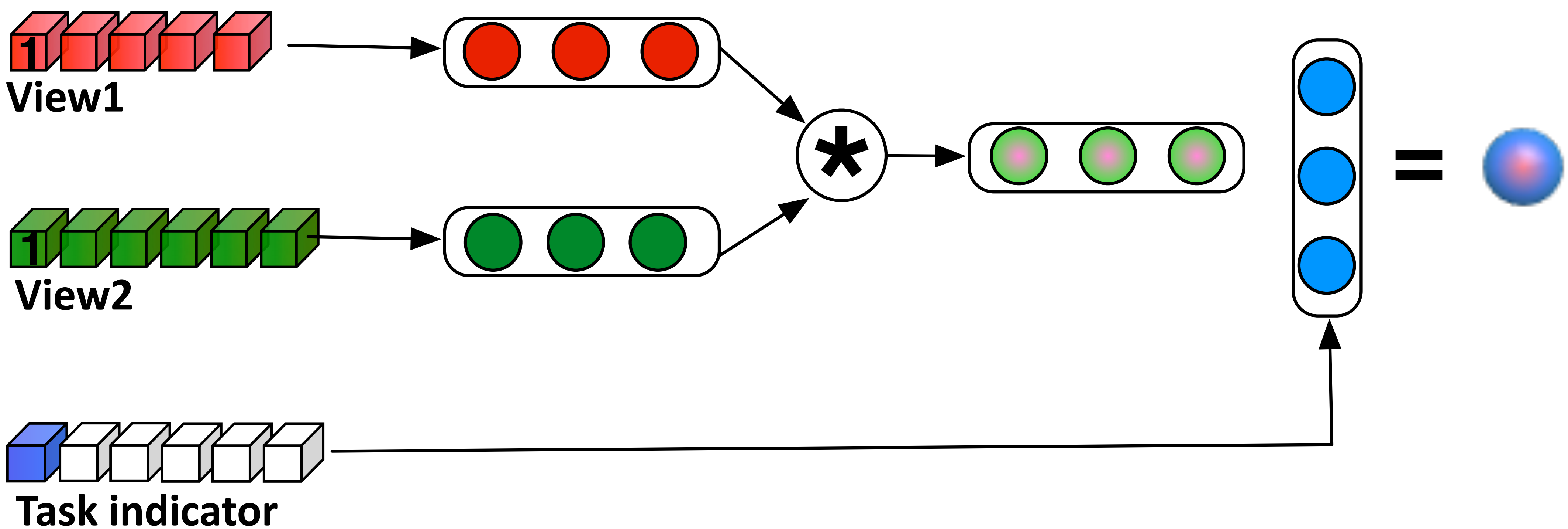
Apply CP tensor factorization on the weight tensor



$$\mathcal{W} = \sum_{r=1}^{R} \phi_{\mathbf{r}} \circ \theta_{\mathbf{r}}^{(\mathbf{1})} \circ \theta_{\mathbf{r}}^{(\mathbf{2})}$$

Multilinear Factorization Machines (MFMs)

After some calculation, $\langle \mathcal{W}, \mathcal{Z}_t \rangle = \phi^t \prod_{v=1}^{V} * \left(\mathbf{z}^{(v)^{\mathrm{T}}} \mathbf{\Theta}^{(v)} \right)^{\mathrm{T}}$

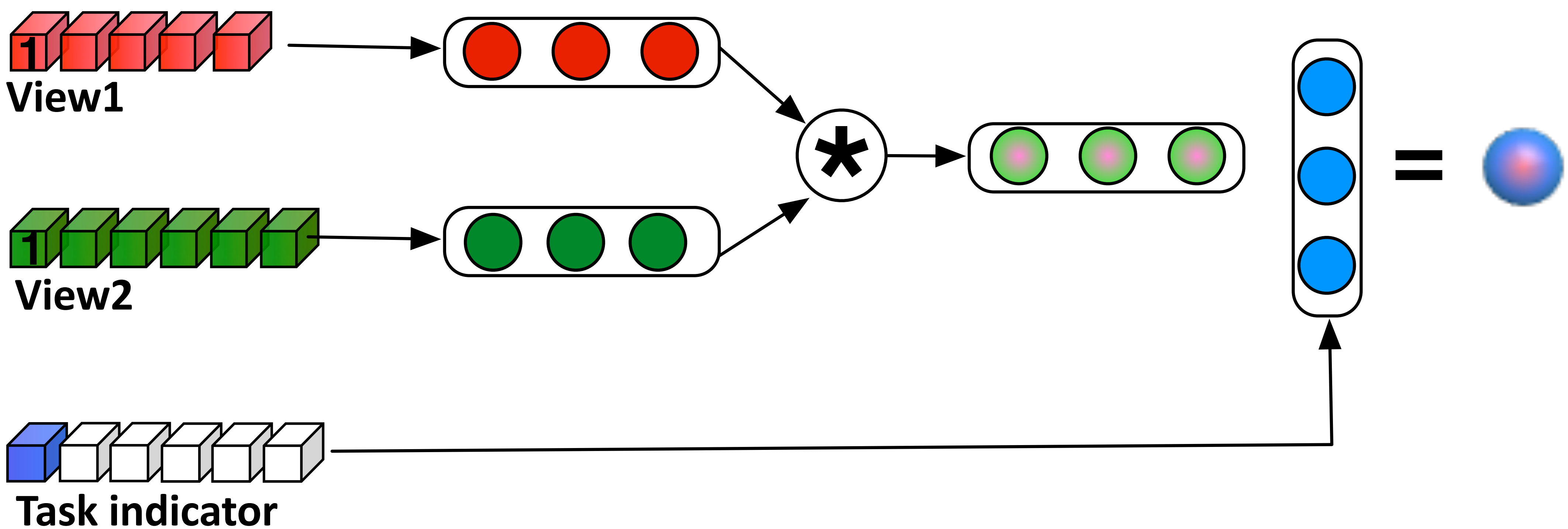


First project each view to a latent space

Then obtain joint representation of multi-view data by element-wise multiplication.

Multilinear Factorization Machines (MFMs)

After some calculation, $\langle \mathcal{W}, \mathcal{Z}_t \rangle = \phi^t \prod_{v=1}^{V} * \left(\mathbf{z}^{(v)^{\mathrm{T}}} \mathbf{\Theta}^{(v)} \right)^{\mathrm{T}}$



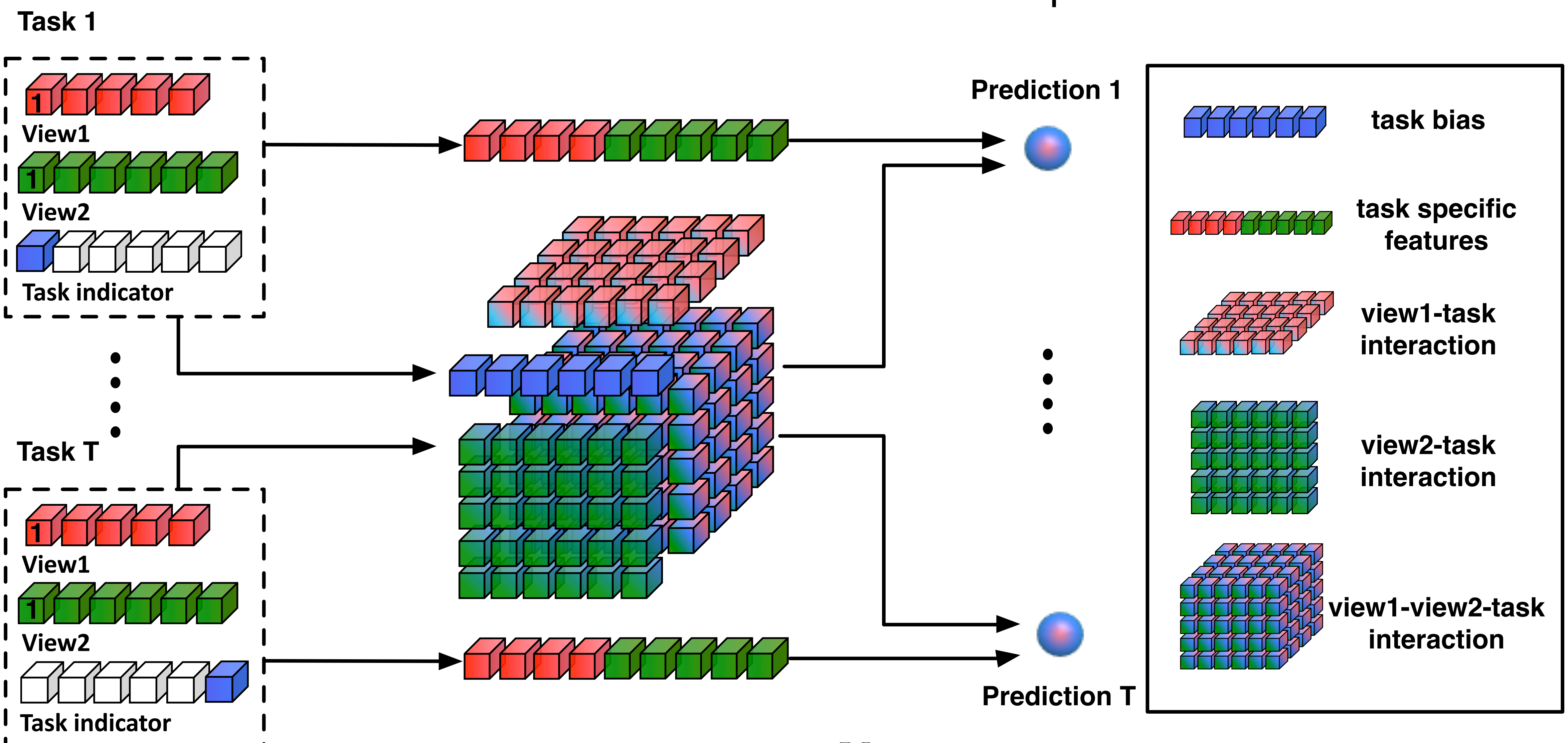
First project each view to a latent space

Then obtain joint representation of multi-view data by element-wise multiplication.

Too restrict to assume all tasks share the same subspace

Multilinear Factorization Machines (MFMs)

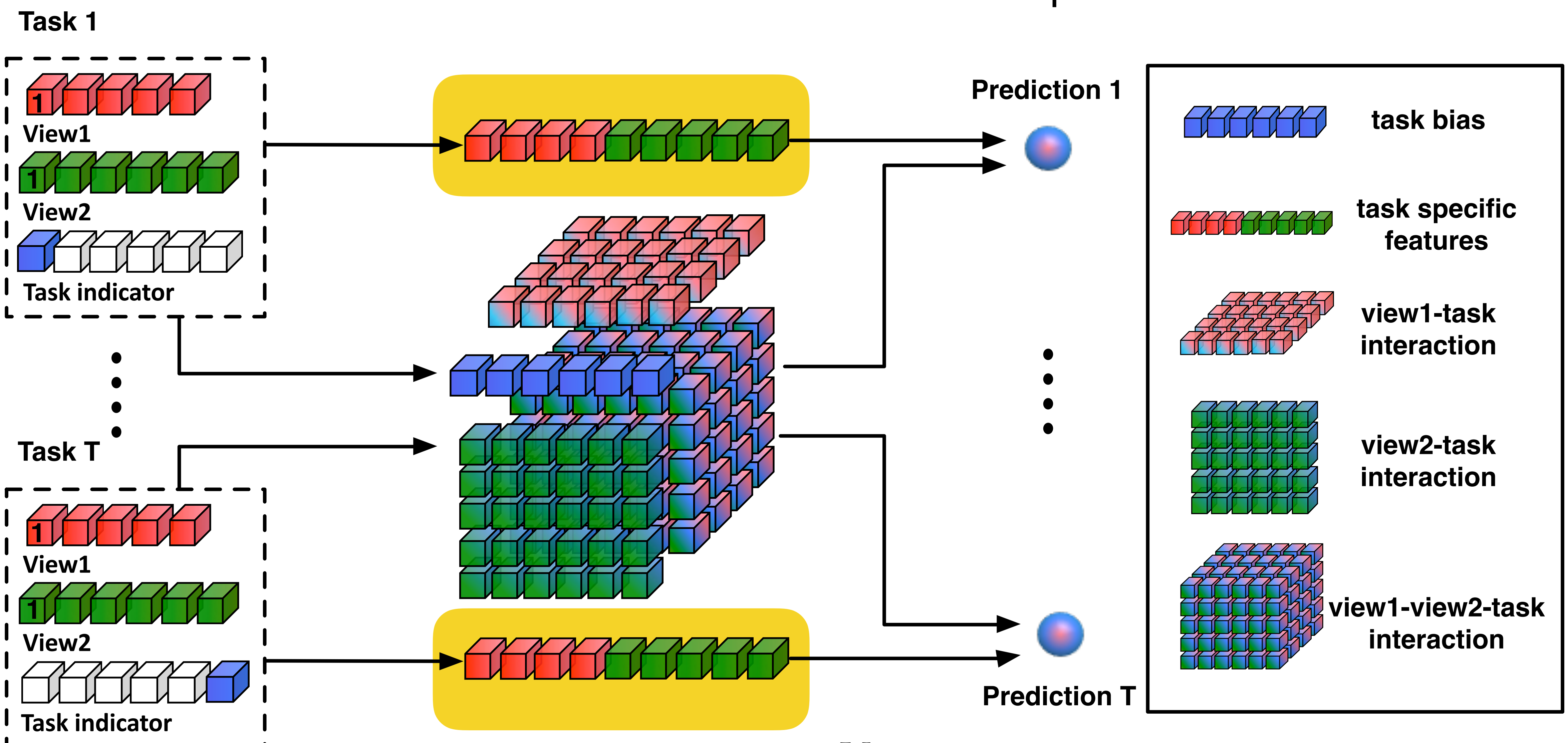
Learning from task-specific linear space and the task-view shared multilinear feature space



$$f_t(\{\mathbf{x}^{(v)}\}) = \mathbf{x}^{\mathrm{T}}\mathbf{u}_t + \boldsymbol{\phi}^t \prod_{v=1}^{V} * \left(\mathbf{z}^{(v)^{\mathrm{T}}} \boldsymbol{\Theta}^{(v)}\right)^{\mathrm{T}}$$

Multilinear Factorization Machines (MFMs)

Learning from task-specific linear space and the task-view shared multilinear feature space



$$f_t(\{\mathbf{x}^{(v)}\}) = \mathbf{x}^{\mathrm{T}}\mathbf{u}_t + \boldsymbol{\phi}^t \prod_{v=1}^{V} * \left(\mathbf{z}^{(v)^{\mathrm{T}}} \boldsymbol{\Theta}^{(v)}\right)^{\mathrm{T}}$$

Learning MFMs

$$\min \mathcal{R}(\mathbf{\Phi}, \{\mathbf{\Theta}^{(v)}\}, \mathbf{U}) = \sum_{t=1}^{T} \mathcal{L}_t(f_t(\{\mathbf{X}_t^{(v)}\}), \mathbf{y}_t)$$

$$+ \lambda \Omega_\lambda(\mathbf{\Phi}, \{\mathbf{\Theta}^{(v)}\}) + \gamma \Omega_\gamma(\mathbf{U})$$

$\mathbf{\Theta}^{(v)} \in \mathbb{R}^{(I_v+1)\times R}$ factor matrix for each view

$\mathbf{\Phi} \in \mathbb{R}^{T\times R}$ factor matrix for tasks

$\mathbf{U} \in \mathbb{R}^{I\times T}$ weight matrix for linear mapping

$\mathcal{L}_t(f_t(\{\mathbf{X}_t^{(v)}\}), \mathbf{y}) = \frac{1}{N_t} \sum_{n=1}^{N_t} \ell\left(f_t(\{\mathbf{x}_{t,n}^{(v)}\}), y_{t,n}\right)$ empirical loss

Solved by alternating block coordinate descent

Experiments - Dataset & Evaluation

- FOX: **multi-class** w/ image and text features
- DBLP: **multi-label** w/ textual and linkage features
- MovieLens: **regression** w/ users, items, tags
- Amazon: **large-scale regression** w/ users, items, text

Classification	#Feature	T	N_p	N_n
FOX	image(996), text(2,711)	4	178∼635	888∼1,345
DBLP	linkage(4,638), text(687)	6	635∼1,950	2,688∼3,985

Regression	#Feature	T	N	Density
MovieLens	users(943), movies(1,599), tags(1,065)	10	758∼39,895	6.3%
Amazon	users(1,805,364), items(192,978), text(83,143)	5	349,038∼1,015,189	0.001%

Average results of 10 times of random sampling:

n% labeled instances as training set (n=10,20, and 30)

10% as validation set, 40% as testing set

Experiments - Compared Methods

- **rMTFL:** robust multi-task feature learning algorithm
- **IteM²:** transductive MTMV classification algorithm
- **CSL-MTMV:** state-of-the-art inductive MTMV learning algorithm
- **Factorization Machine (FM)**: state-of-the-art factorization model
- **Tensor Factorization (TF):** factorize highest-order weight tensor
- **Multilinear Tensor Factorization (MFM):** proposed method
  - **MFM-T:** only using tensor part (**U** is fixed as a zero matrix)
  - **MFM-F:** using F-norm regularizers for all parameters
  - **MFM-F-S:** using $\ell_{2,1}\text{-norm}$ on **U** for joint feature selection
    F-norm on the rest parameters

Experiments - Classification on FOX dataset

Training Ratio	Measure	rMTFL	FM	TF	IteM^2
10%	ACC	0.8816±0.011	0.7883±0.011	0.8460±0.035	0.4052±0.076
	F1	0.6911±0.035	0.2930±0.046	0.6362±0.044	0.3598±0.030
	AUC	0.9109±0.013	0.7764±0.018	0.8681±0.038	0.5326±0.036
20%	ACC	0.9039±0.013	0.8087±0.011	0.8546±0.025	0.5091±0.078
	F1	0.7654±0.026	0.3764±0.050	0.6632±0.051	0.3306±0.068
	AUC	0.9353±0.016	0.8260±0.012	0.8751±0.029	0.4954±0.043
30%	ACC	0.9314±0.005	0.8255±0.007	0.8767±0.082	0.4289±0.134
	F1	0.8051±0.015	0.4448±0.026	0.7302±0.132	0.3314±0.056
	AUC	0.9709±0.005	0.8393±0.012	0.9010±0.091	0.5365±0.039

Training Ratio	Measure	CSL-MTMV	MFM-T	MFM-F	MFM-F-S
10%	ACC	0.8986±0.011	0.9259±0.019	**0.9343±0.012**	**0.9364±0.011**
	F1	0.7335±0.029	0.7799±0.053	**0.8076±0.038**	**0.8119±0.027**
	AUC	0.9342±0.011	0.9678±0.015	**0.9763±0.008**	**0.9777±0.009**
20%	ACC	0.9264±0.005	0.9551±0.005	**0.9569±0.010**	**0.9612±0.005**
	F1	0.8004±0.012	0.8721±0.012	**0.8769±0.027**	**0.8882±0.014**
	AUC	0.9705±0.003	0.9883±0.003	**0.9885±0.006**	**0.9922±0.002**
30%	ACC	0.9390±0.004	0.9641±0.007	**0.9709±0.003**	**0.9697±0.004**
	F1	0.8341±0.012	0.9000±0.018	**0.9185±0.010**	**0.9149±0.010**
	AUC	0.9812±0.003	0.9916±0.003	**0.9949±0.001**	**0.9949±0.001**

Experiments - Classification on FOX dataset

Training Ratio	Measure	rMTFL	FM	TF	IteM^2
10%	ACC	0.8816±0.011	0.7883±0.011	0.8460±0.035	0.4052±0.076
	F1	0.6911±0.035	0.2930±0.046	0.6362±0.044	0.3598±0.030
	AUC	0.9109±0.013	0.7764±0.018	0.8681±0.038	0.5326±0.036
20%	ACC	0.9039±0.013	0.8087±0.011	0.8546±0.025	0.5091±0.078
	F1	0.7654±0.026	0.3764±0.050	0.6632±0.051	0.3306±0.068

MFMs consistently outperform compared methods.

MFMs improve 6~10% over the best compared methods

Training Ratio	Measure	CSL-MTMV	MFM-T	MFM-F	MFM-F-S
10%	ACC	0.8986±0.011	0.9259±0.019	**0.9343±0.012**	**0.9364±0.011**
	F1	0.7335±0.029	0.7799±0.053	**0.8076±0.038**	**0.8119±0.027**
	AUC	0.9342±0.011	0.9678±0.015	**0.9763±0.008**	**0.9777±0.009**
20%	ACC	0.9264±0.005	0.9551±0.005	**0.9569±0.010**	**0.9612±0.005**
	F1	0.8004±0.012	0.8721±0.012	**0.8769±0.027**	**0.8882±0.014**
	AUC	0.9705±0.003	0.9883±0.003	**0.9885±0.006**	**0.9922±0.002**
30%	ACC	0.9390±0.004	0.9641±0.007	**0.9709±0.003**	**0.9697±0.004**
	F1	0.8341±0.012	0.9000±0.018	**0.9185±0.010**	**0.9149±0.010**
	AUC	0.9812±0.003	0.9916±0.003	**0.9949±0.001**	**0.9949±0.001**

Experiments - Classification on DBLP dataset

Training Ratio	Measure	rMTFL	FM	TF	IteM^2
10%	ACC	0.8057±0.004	0.7264±0.004	0.7471±0.011	0.6223±0.004
	F1	0.5395±0.015	0.0732±0.019	0.5606±0.011	0.3176±0.007
	AUC	0.7888±0.007	0.6264±0.023	0.7723±0.009	0.5310±0.007
20%	ACC	0.8319±0.004	0.7628±0.007	0.7878±0.007	0.6309±0.003
	F1	0.6447±0.008	0.2680±0.038	0.6247±0.014	0.3494±0.006

MFMs consistently outperform compared methods.

MFMs improve 6~10% over best compared methods

Training Ratio	Measure	CSL-MTMV	MFM-T	MFM-F	MFM-F-S
10%	ACC	0.7290±0.005	0.8008±0.004	**0.8058±0.004**	**0.8062±0.005**
	F1	0.4402±0.004	0.5278±0.018	**0.5469±0.014**	**0.5471±0.015**
	AUC	0.6890±0.006	0.8039±0.010	**0.8113±0.010**	**0.8120±0.009**
20%	ACC	0.7760±0.002	0.8346±0.004	**0.8374±0.004**	**0.8371±0.004**
	F1	0.5295±0.007	0.6274±0.013	**0.6499±0.012**	**0.6508±0.012**
	AUC	0.7655±0.005	0.8531±0.006	**0.8658±0.005**	**0.8632±0.005**
30%	ACC	0.8037±0.003	0.8501±0.004	**0.8527±0.004**	**0.8535±0.004**
	F1	0.5869±0.007	0.6800±0.013	**0.6891±0.012**	**0.6892±0.009**
	AUC	0.8083±0.006	0.8757±0.005	**0.8866±0.006**	**0.8866±0.006**

Experiments - Regression on MovieLens dataset

Training Ratio	Measure	rMTFL	FM	TF	CSL-MTMV
10%	RMSE	1.1861±0.008	1.0251±0.003	1.5679±0.099	1.05013±0.005
	MAE	0.8516±0.004	0.8422±0.004	1.2497±0.088	0.8516±0.004
20%	RMSE	1.0631±0.005	0.9898±0.003	1.2519±0.069	1.0214±0.004
	MAE	0.8539±0.005	0.7997±0.004	0.9801±0.053	0.8294±0.004
30%	RMSE	0.9917±0.003	**0.9765±0.003**	1.2066±0.061	1.0082±0.003
	MAE	0.8159±0.003	**0.7815±0.003**	0.9380±0.045	0.8189±0.003

Training Ratio	Measure	MFM-T	MFM-F	MFM-F-S
10%	RMSE	1.0078±0.005	**1.0069±0.005**	**0.9976±0.004**
	MAE	0.8142±0.005	**0.8082±0.005**	**0.8022±0.004**
20%	RMSE	**0.9877±0.003**	0.9977±0.003	**0.9857±0.003**
	MAE	**0.7987±0.003**	0.8023±0.003	**0.7927±0.004**
30%	RMSE	0.9795±0.003	0.9887±0.004	**0.9785±0.003**
	MAE	0.7885±0.002	0.7823±0.004	**0.7789±0.004**

MFM-T perform well in most cases, indicating that task-specific linear feature map is less important for regression

Experiments - Regression on Amazon dataset

Due to memory overhead, rMTFL and CSL-MTMV are not compared

Training Ratio	Measure	FM	TF	MFM-T	MFM-F	MFM-F-S
10%	RMSE	0.9834±0.001	3.6044±0.003	**0.9775±0.001**	0.9857±0.001	**0.9825±0.002**
	MAE	0.7420±0.001	3.4574±0.005	0.7249±0.001	**0.7158±0.002**	**0.7129±0.001**
20%	RMSE	0.9814±0.001	3.5611±0.018	**0.9764±0.001**	0.9845±0.001	**0.9775±0.001**
	MAE	0.7343±0.002	3.3965±0.030	0.7255±0.001	**0.7112±0.001**	**0.7086±0.001**
30%	RMSE	0.9782±0.002	3.4962±0.018	**0.9705±0.002**	0.9841±0.001	**0.9733±0.001**
	MAE	0.7257±0.002	3.2945±0.034	**0.7001±0.001**	0.7115±0.001	**0.7078±0.001**

MFM-T perform well in most cases, indicating that task-specific linear feature map is less important for regression

Conclusion

1. A simple way to learn **joint representation** of multi-view data, and demonstrate it effectiveness.

2. Consider both **linear feature map** and the **shared multilinear structure** can improve the performance

3. Time complexity and space complexity of MFMs are **linear** in the feature dimensionality.

Code available at GitHub.
Thanks for SIGIR/WSDM Travel Grant!