Learning and Selecting Features Jointly with Point-wise Gated Boltzmann Machines

Kihyuk Sohn, Guanyu Zhou, Chansoo Lee, Honglak Lee
Dept. of Electrical Engineering and Computer Science
University of Michigan

ICML | Atlanta
International Conference on Machine Learning

UNIVERSITY OF MICHIGAN

Outline

- Overview
- Preliminary
- Point-wise Gated Boltzmann Machines
- Experimental results
- Conclusion

Learning from *scratch*

- Unsupervised feature learning

  (Hinton et al., 2006, Bengio et al., 2007, Ranzato et al., 2007, Bengio, 2009)

  – Powerful in **discovering** representations from unlabeled data.

  – However, not all patterns (or data) are equally important.

    - When data contains lots of distracting factors, learning meaningful representations can be challenging.

- Feature selection

  (Jain & Zongker, 1997, Yang & Pedersen, 1997, Weston et al., 2001, Guyon & Elisseeff, 2003)

  – Powerful in **selecting** features from labeled data.

  – However, it assumes existence of discriminative features.

    - There may not be such features at hand.

Motivating Example

- Learning features from images for object recognition.

- Want to learn “person” specific high-level features for good recognition performance.

Motivating Example

- Learning features from images for object recognition.

- Want to learn “person” specific high-level features for good recognition performance.
- There are lots of irrelevant patterns in the background other than person.

Motivating Example

- Learning features from images for object recognition.

- Want to learn “person” specific high-level features for good recognition performance.
- There are lots of irrelevant patterns in the background other than person.
- Class labels may be helpful, though they don’t specify where to focus in the image.

Motivating Example

- Learning features from images for object recognition.

- Want to learn “person” specific high-level features for good recognition performance.
- There are lots of irrelevant patterns in the background other than person.
- Class labels may be helpful, though they don’t specify where to focus in the image.

Q. How can we **learn task-relevant high-level features** using (weak) supervision?

→ We develop a joint model for feature learning and feature selection.

Related Work

- Feature learning using class labels
  - Convolutional Deep Neural Networks (Lecun et al., Neural Computation 1989; Krizhevsky et al., NIPS 2012, Ciresan et al., Neural Computation 2011, etc.)
  - Deep (Belief) Networks (Hinton and Salakhutdinov, Science 2006, Bengio et al., NIPS 2006, Hinton et al., Neural Computation 2006, etc.)
  - Discriminative RBMs (Larochelle & Bengio, ICML 2008)

Related Work

- Foreground and background modeling with Boltzmann machines.
  - Robust Boltzmann machines (Tang et al., CVPR 2012)
  - Masked RBMs (Le Roux et al., Neural Computation 2011; Heess et al., ICANN 2011)
  - Our model makes use of class labels and perform generative feature selection while feature learning.

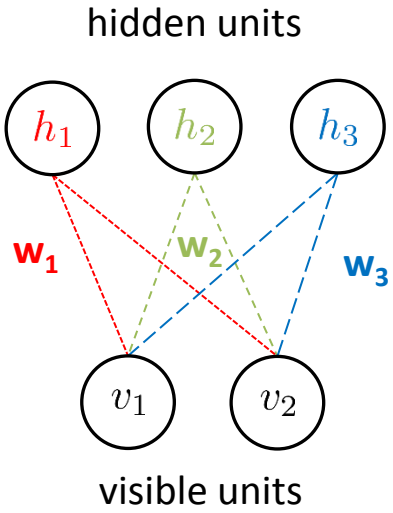
Restricted Boltzmann Machines

- Representation
  - Undirected bipartite graphical model.
  - $\mathbf{v} \in \{0,1\}^D$: binary visible (observed) units.
  - $\mathbf{h} \in \{0,1\}^K$: binary hidden units.

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp\left(-E(\mathbf{v},\mathbf{h})\right)$$

$$E(\mathbf{v},\mathbf{h}) = -\sum_{ij} v_i W_{ij} h_j - \sum_j b_j h_j - \sum_i c_i v_i$$

$$= -\mathbf{v}^T\mathbf{W}\mathbf{h} - \mathbf{b}^T\mathbf{h} - \mathbf{c}^T\mathbf{v}$$



Inference and Learning in RBM

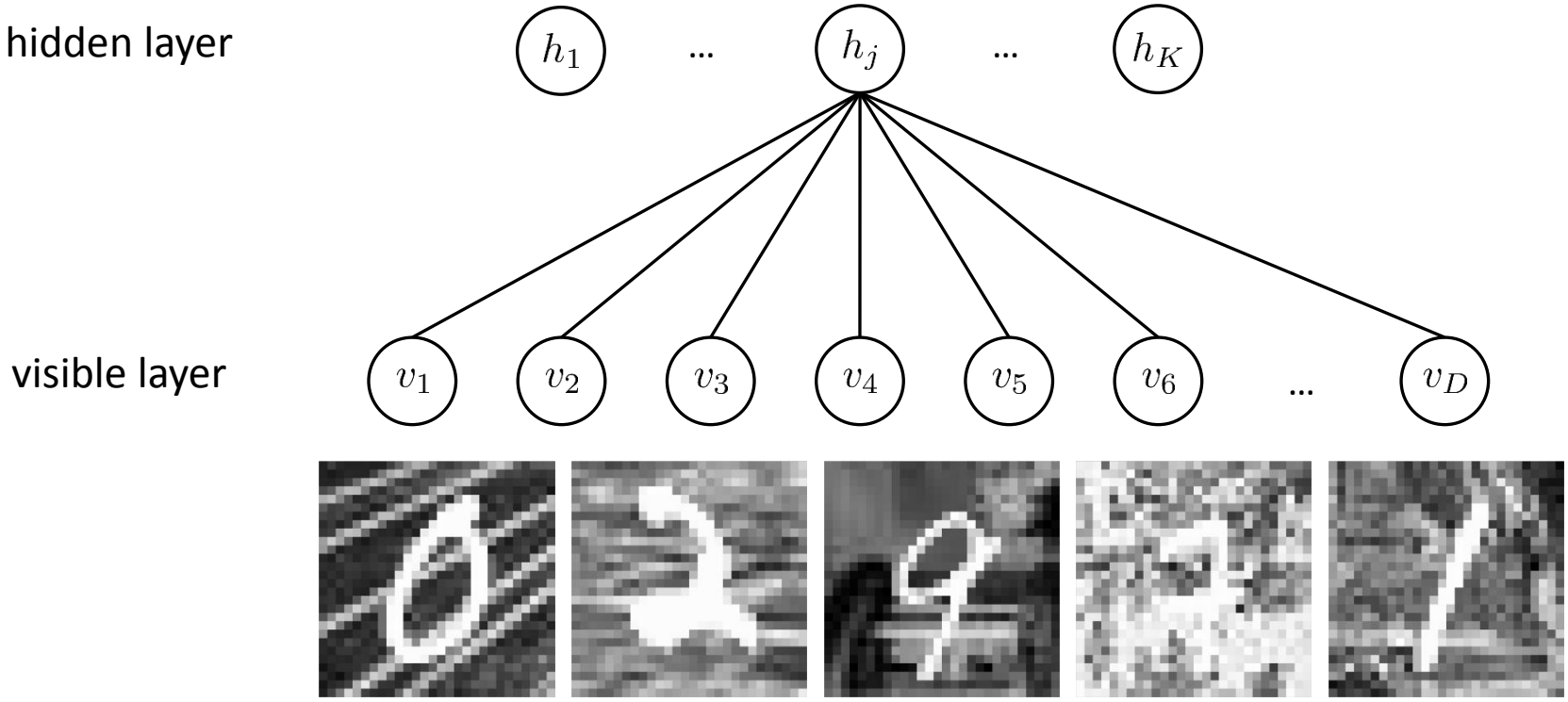
- Inference
  - Efficient and exact due to conditional independence.

$$P(h_j = 1|\mathbf{v}) = \text{sigmoid}(\sum_i v_i W_{ij} + b_j)$$

$$P(v_i = 1|\mathbf{h}) = \text{sigmoid}(\sum_j W_{ij} h_j + c_i)$$

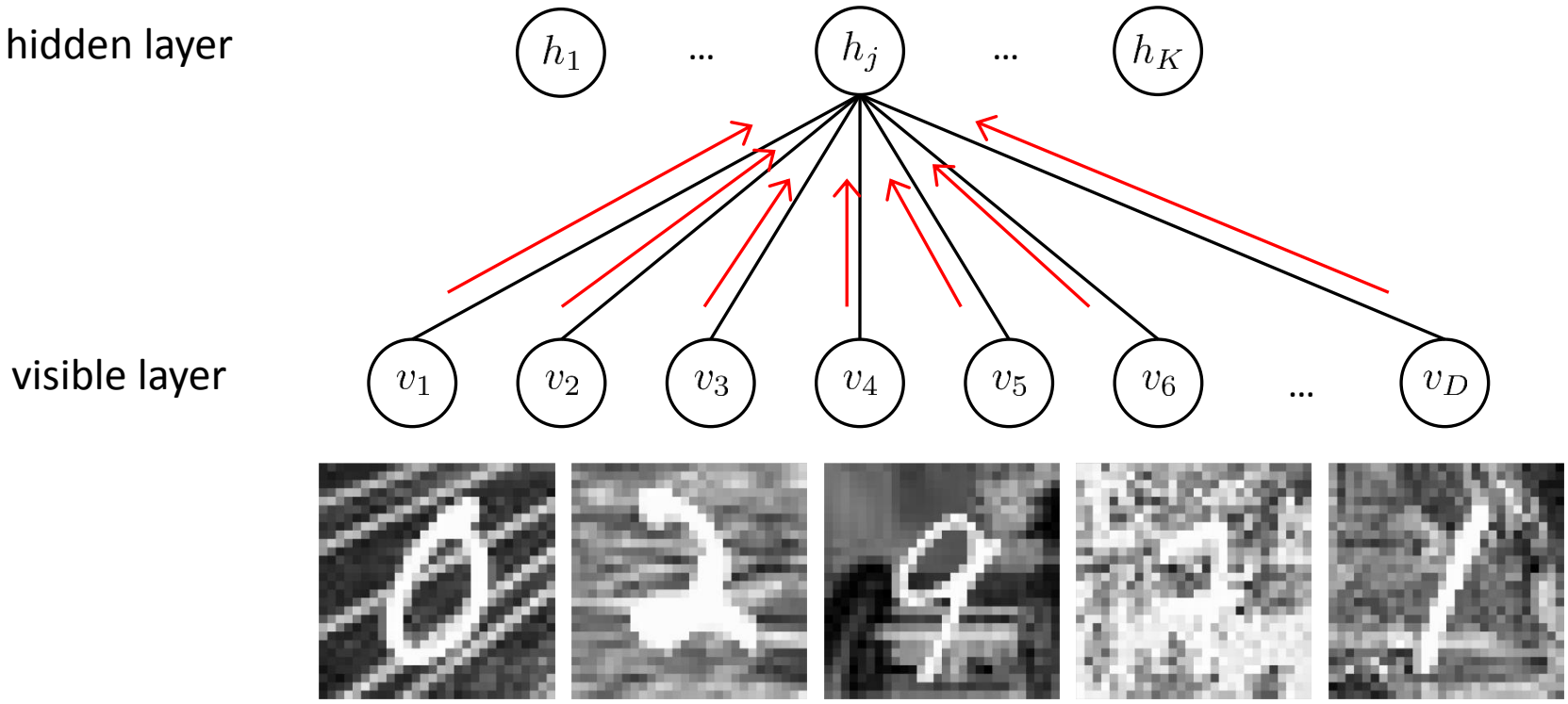
  - Joint probability can be estimated using Gibbs sampling.
  - Posterior $P(h_j = 1|\mathbf{v})$ can be used as a feature.
- Training: maximum-likelihood.
  - Stochastic gradient descent using sampling-based approximation (e.g., contrastive divergence).

Feature Encoding in RBM



Samples from variations of MNIST with natural images in the background.
(Larochelle et al., ICML 2007)

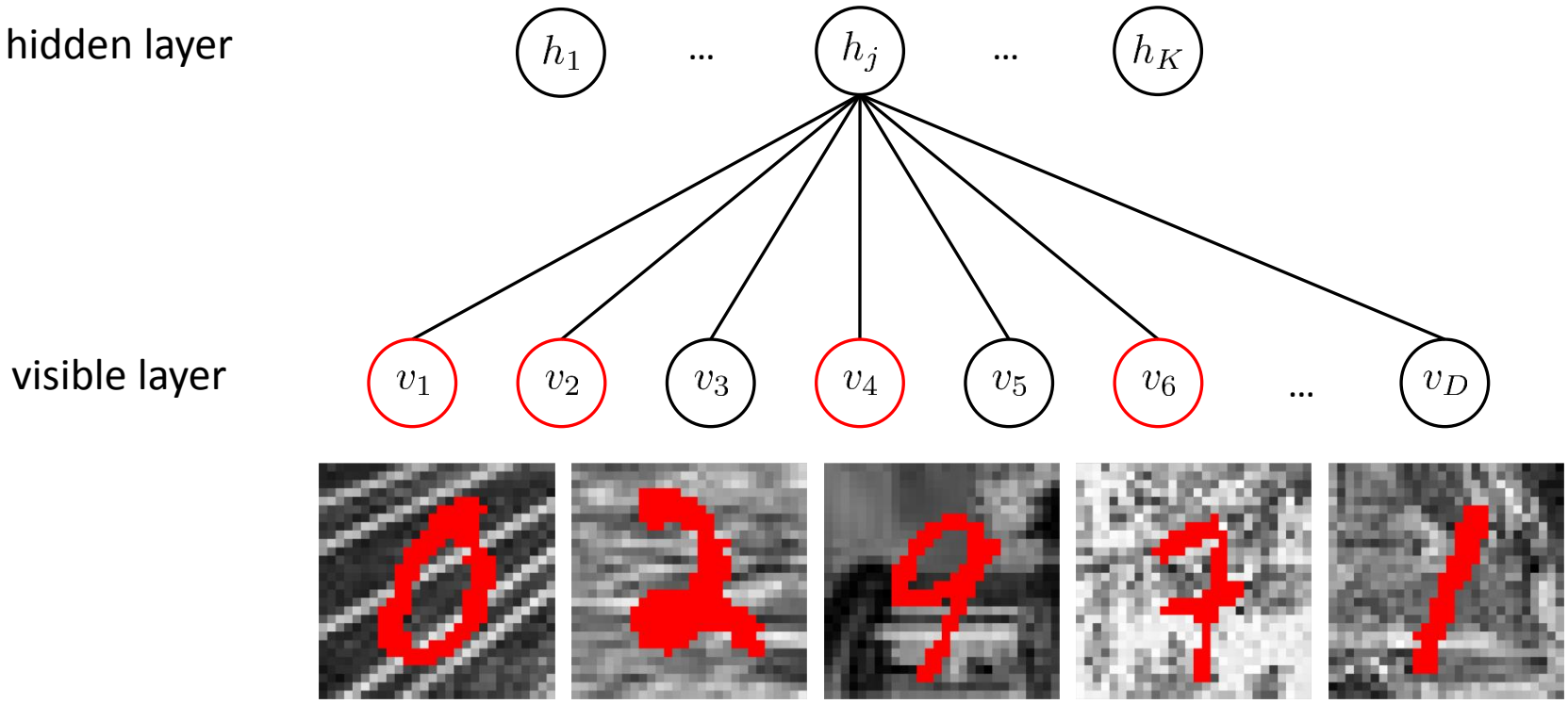
Feature Encoding in RBM



Issues with standard RBMs:

1. RBMs assume all input features are useful (e.g., task-relevant), but it may not be true in many scenarios.

Feature Encoding in RBM



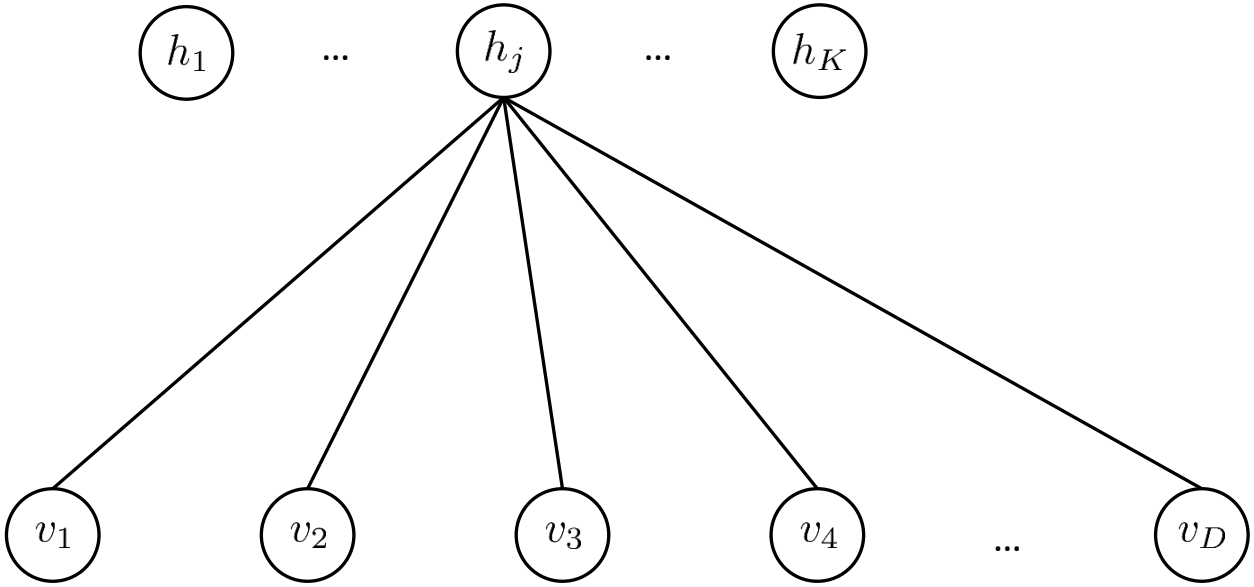
Issues with standard RBMs:

1. RBMs assume all input features are useful (e.g., task-relevant), but it may not be true in many scenario.
2. Set of useful input features may vary across examples.

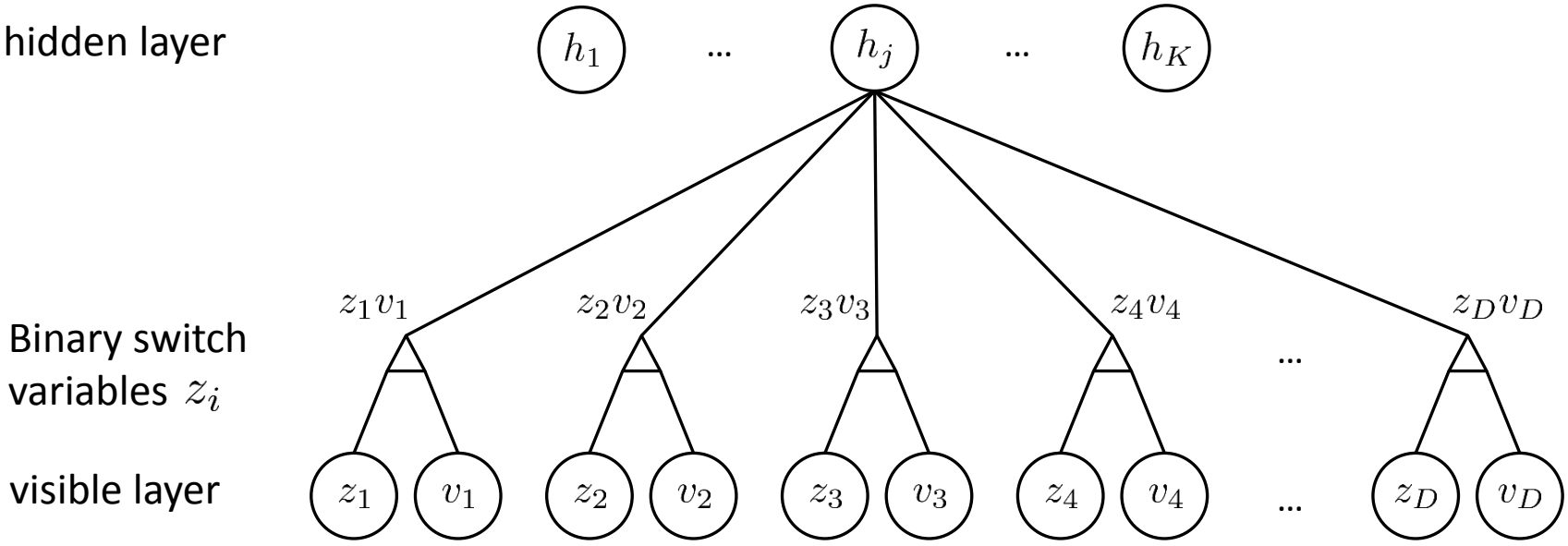
Point-wise Gated Boltzmann Machines

hidden layer

visible layer



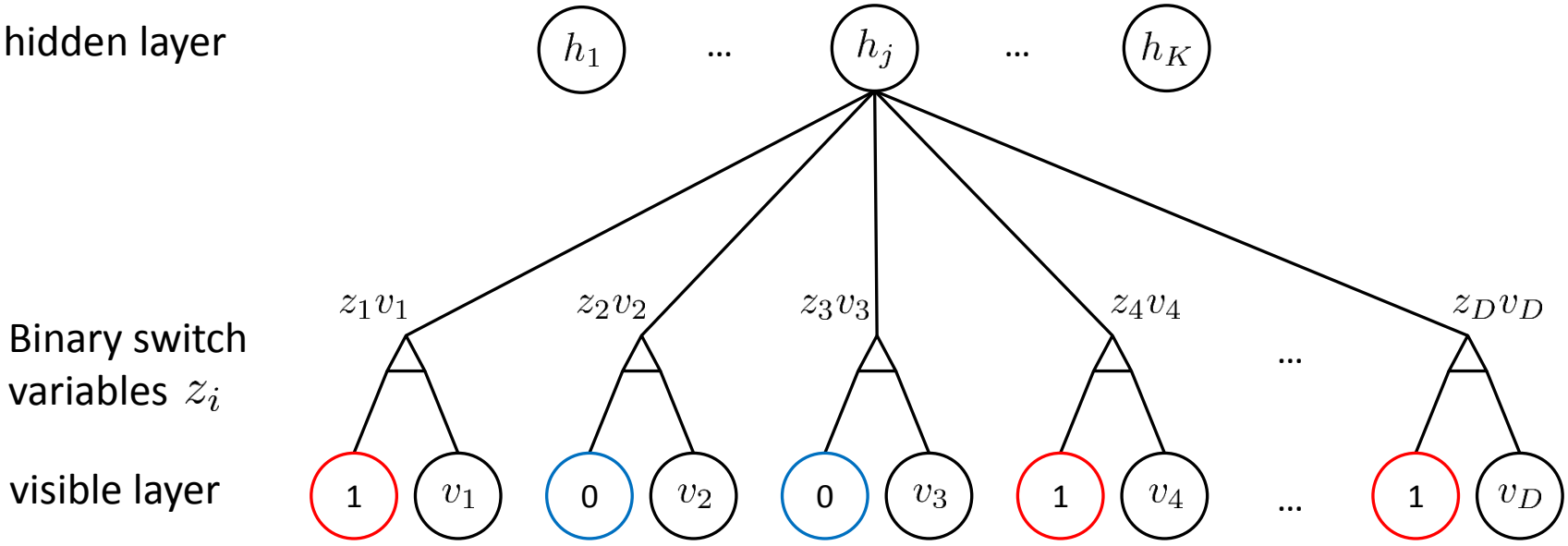
Point-wise Gated Boltzmann Machines



Point-wise Gated Boltzmann Machines (PGBM)

- Point-wise (or input coordinate-wise) multiplicative interaction between switch and visible variables.

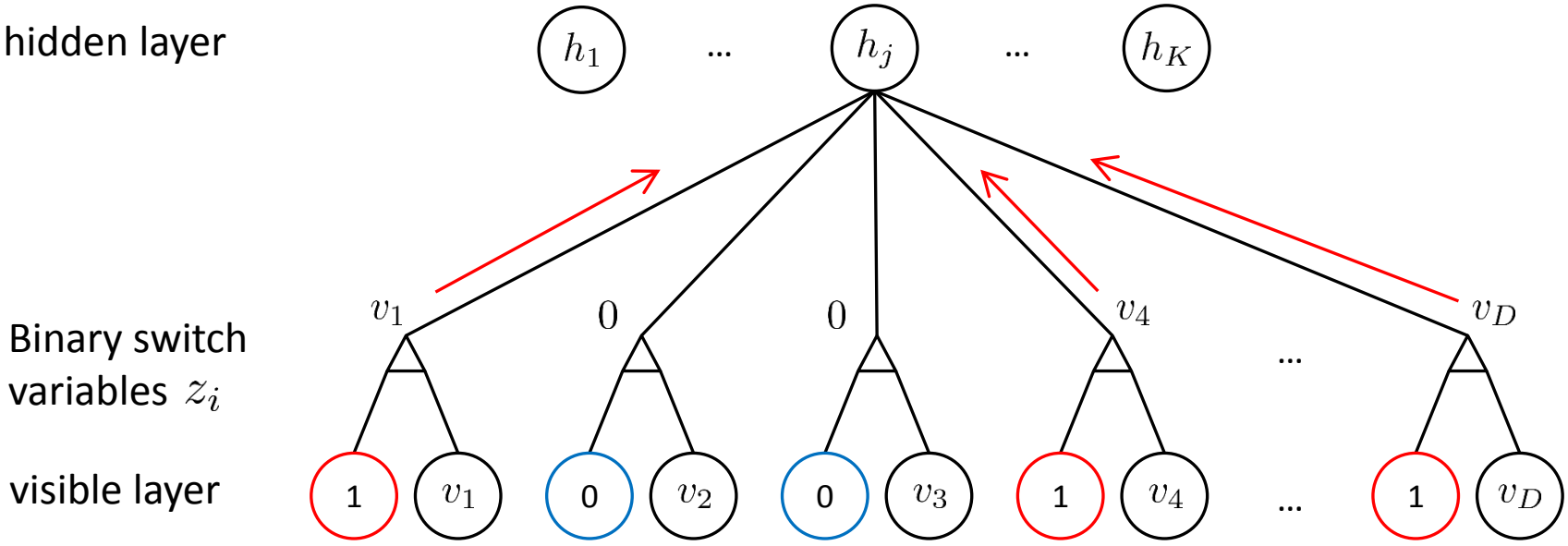
Point-wise Gated Boltzmann Machines



Point-wise Gated Boltzmann Machines (PGBM)

- Point-wise (or input coordinate-wise) multiplicative interaction between switch and visible variables.
- Per-visible-unit switch variable (z_1,…,z_D ; binary) gates the contribution of each visible variable only when it is useful.

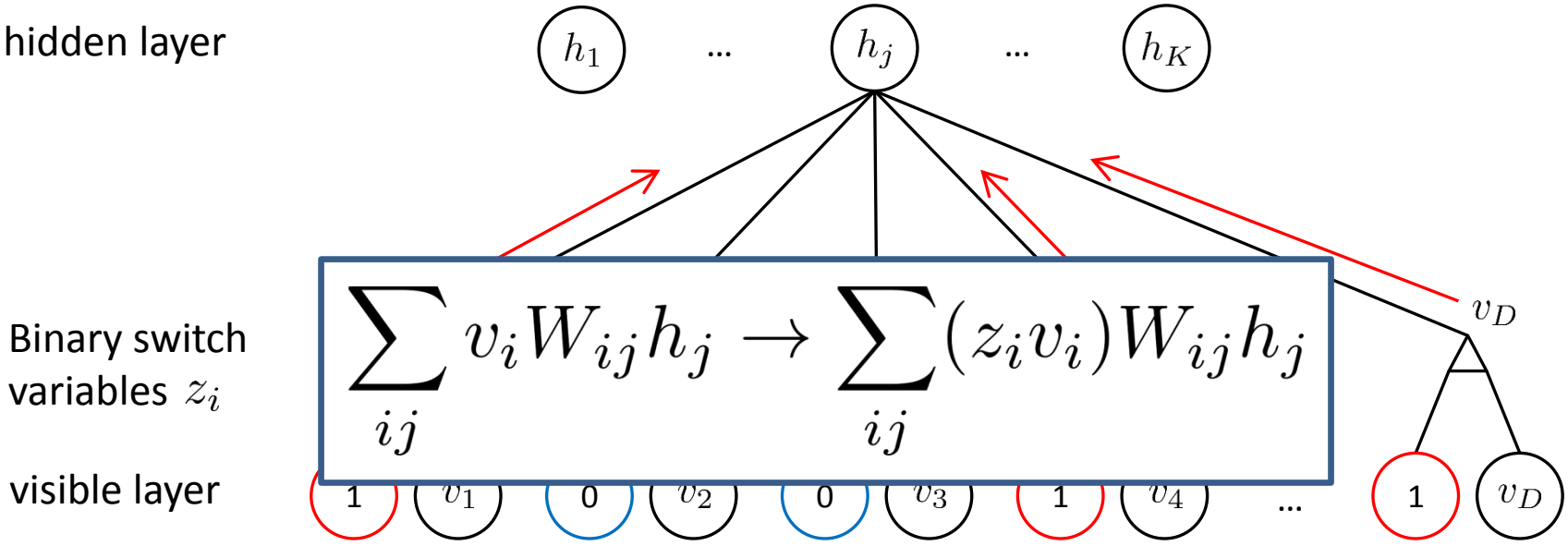
Point-wise Gated Boltzmann Machines



Point-wise Gated Boltzmann Machines (PGBM)

- Point-wise (or input coordinate-wise) multiplicative interaction between switch and visible variables.
- Per-visible-unit switch variable (z_1,…,z_D ; binary) gates the contribution of each visible variable only when it is useful.

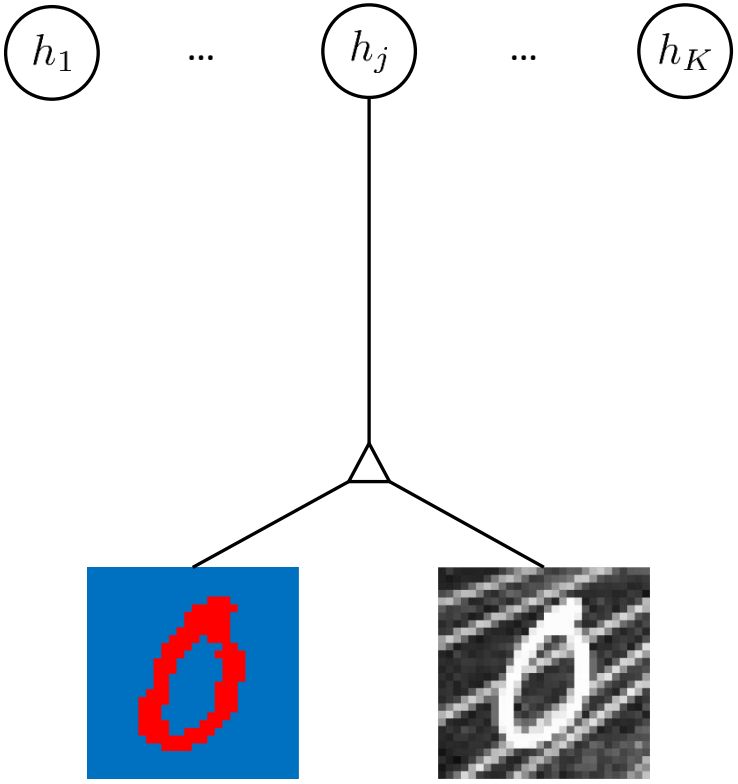
Point-wise Gated Boltzmann Machines



Point-wise Gated Boltzmann Machines (PGBM)

- Point-wise (or input coordinate-wise) multiplicative interaction between switch and visible variables.
- Per-visible-unit switch variable (z_1,…,z_D ; binary) gates the contribution of each visible variable only when it is useful.

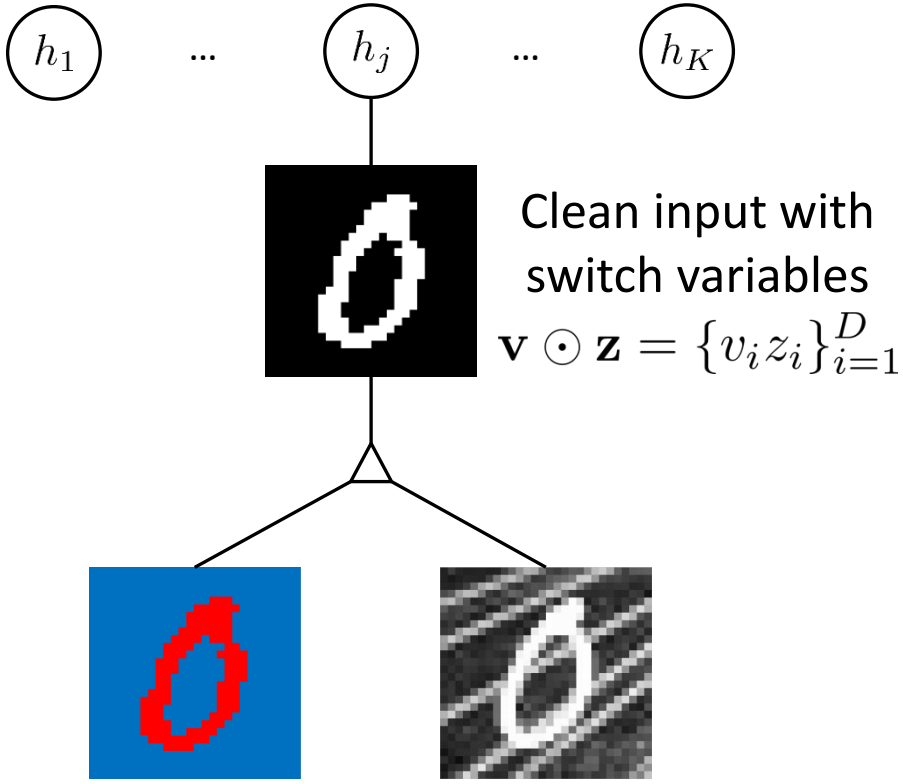
Point-wise Gated Boltzmann Machines



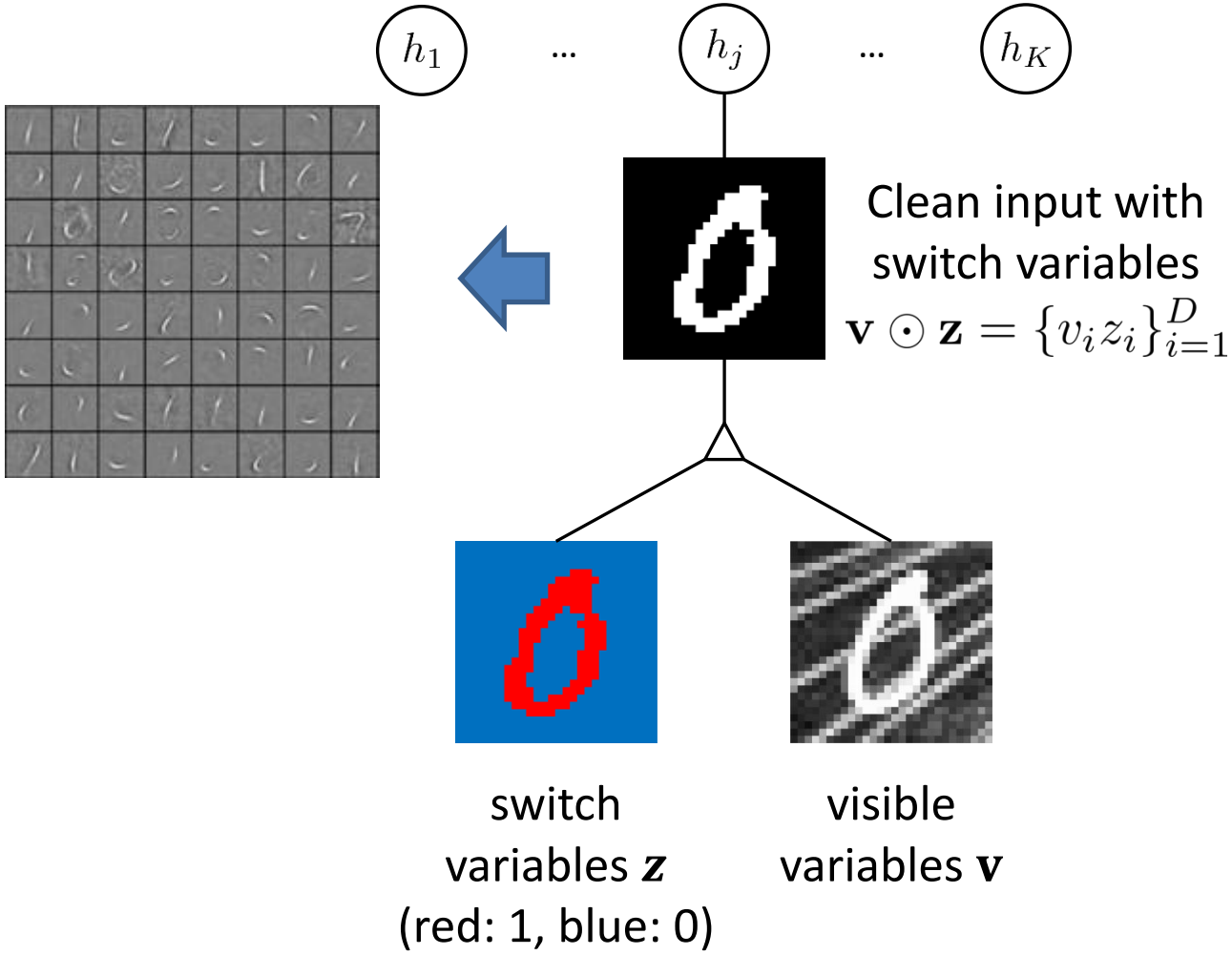
switch variables $\boldsymbol{z}$ (red: 1, blue: 0)

visible variables $\mathbf{v}$

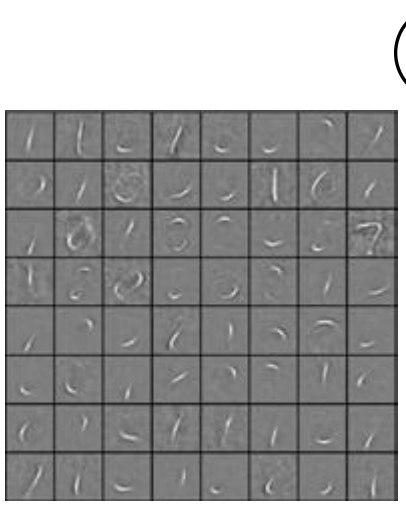
Point-wise Gated Boltzmann Machines

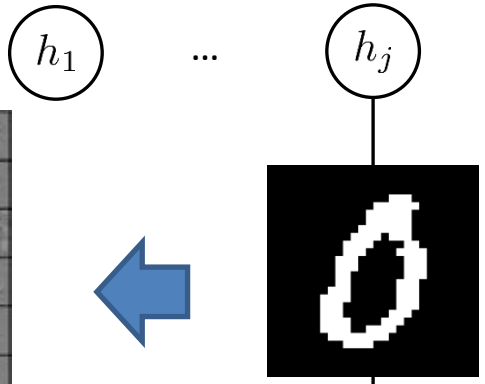


Point-wise Gated Boltzmann Machines



Point-wise Gated Boltzmann Machines

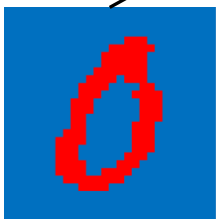




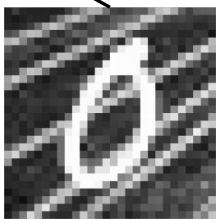
Clean input with switch variables

$$\mathbf{v} \odot \mathbf{z} = \{v_i z_i\}_{i=1}^{D}$$

Focus on different set of input features *dynamically*.



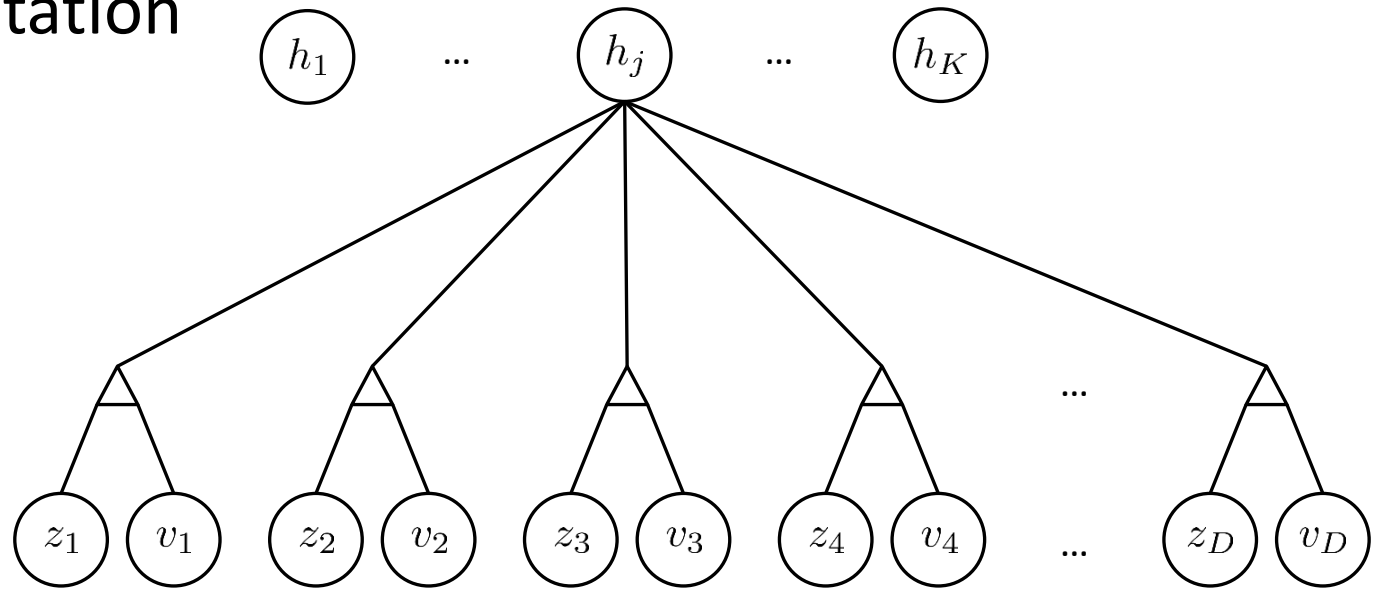
switch variables $\boldsymbol{z}$ (red: 1, blue: 0)



visible variables $\mathbf{v}$

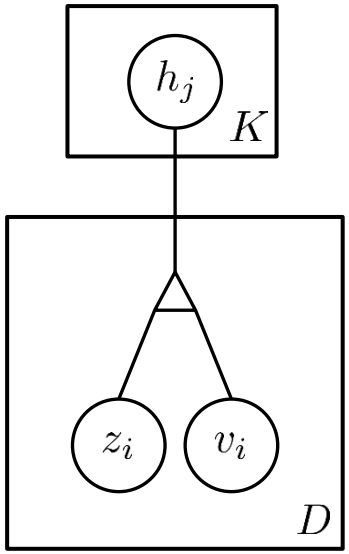
Point-wise Gated Boltzmann Machines

- Plate notation



Point-wise Gated Boltzmann Machines

- Plate notation

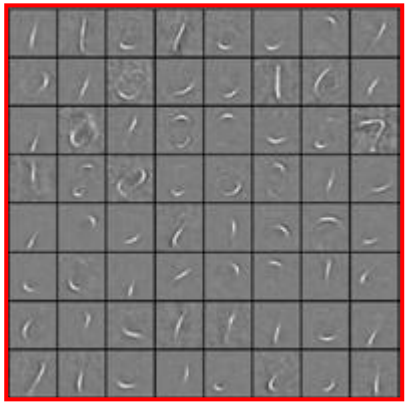


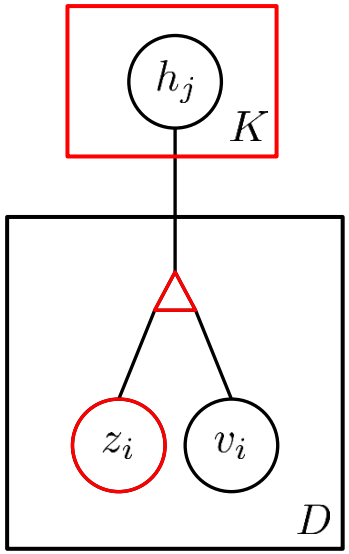
Point-wise Gated Boltzmann Machines (PGBM)

Point-wise Gated Boltzmann Machines

- Plate notation

Focus on modeling
useful input features



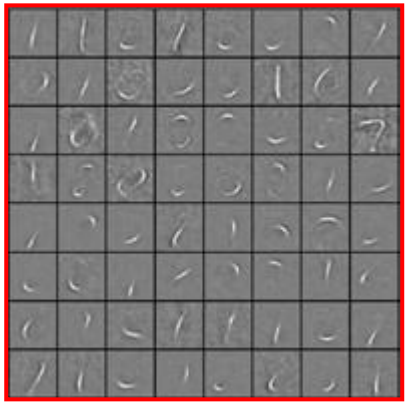


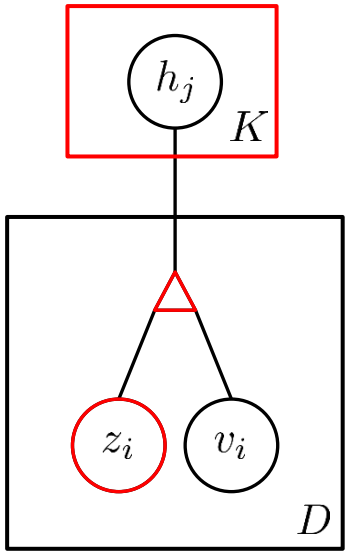
Point-wise Gated Boltzmann Machines (PGBM)

Point-wise Gated Boltzmann Machines

- Plate notation

Focus on modeling
useful input features





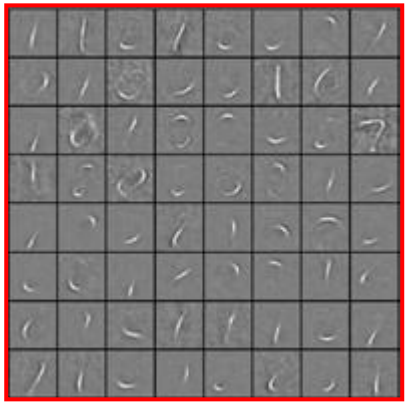
Point-wise Gated Boltzmann Machines (PGBM)

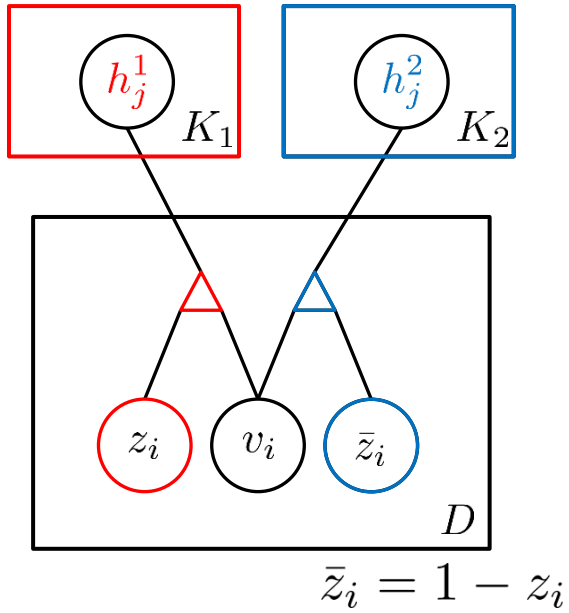
- How about the irrelevant patterns?

Point-wise Gated Boltzmann Machines

- Plate notation

Focus on modeling
useful input features





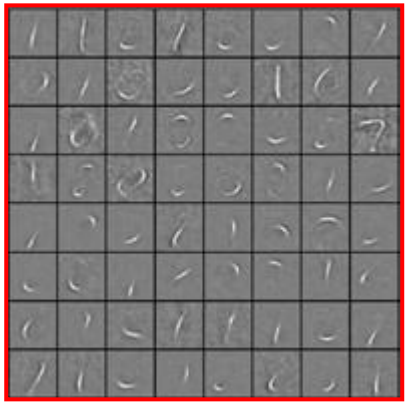
Point-wise Gated Boltzmann Machines (PGBM)

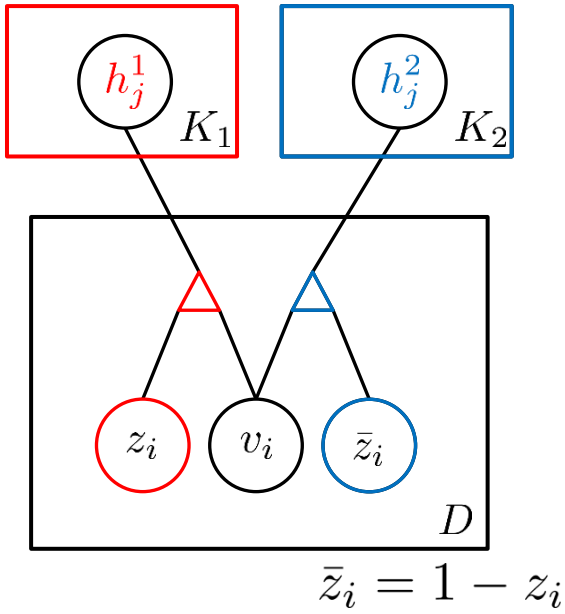
- PGBM models irrelevant patterns using another set of hidden variables.

Point-wise Gated Boltzmann Machines

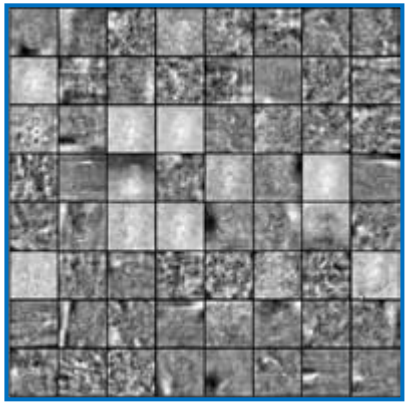
- Plate notation

Focus on modeling useful input features





Focus on modeling the rest of input features



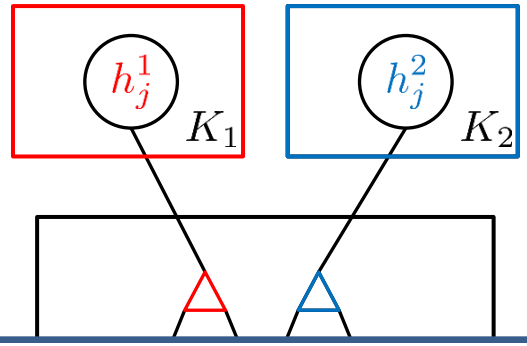
Point-wise Gated Boltzmann Machines (PGBM)

- PGBM models irrelevant patterns using another set of hidden variables.
- Modeling irrelevant patterns helps distinguishing relevant patterns from irrelevant patterns.

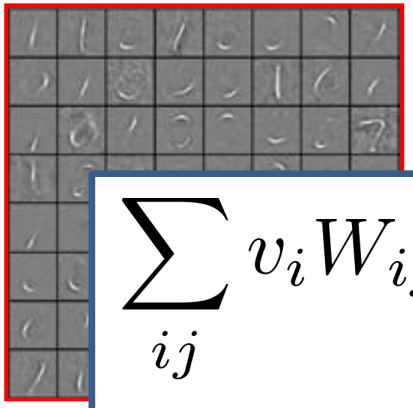
Point-wise Gated Boltzmann Machines

- Plate notation

Focus on modeling useful input features



Focus on modeling the rest of input features



$$\sum_{ij} v_i W_{ij} h_j \rightarrow \sum_{ij} (z_i v_i) W_{ij} h_j$$

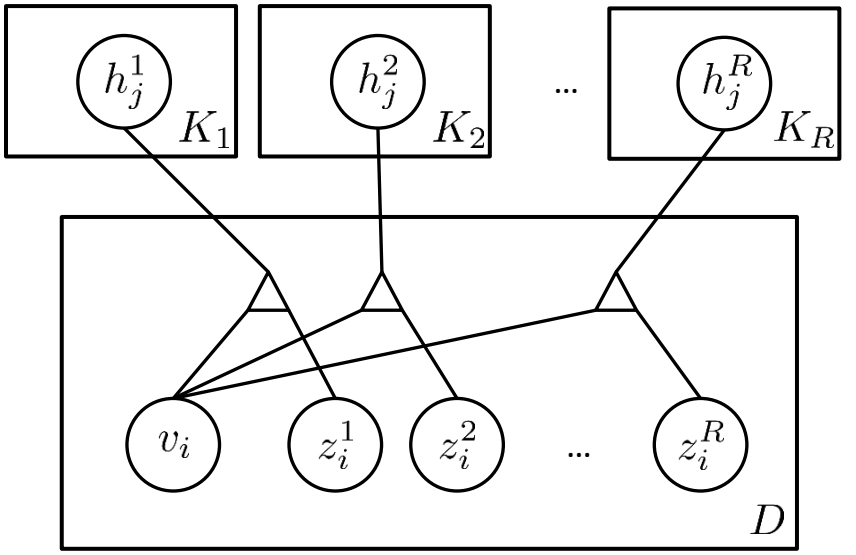
$$\rightarrow \sum_{ij} (z_i v_i) W_{ij}^1 h_j^1 + \sum_{ij} (\bar{z}_i v_i) W_{ij}^2 h_j^2$$

Point-wise

- PGBM n variables.
- Modeling irrelevant patterns helps distinguishing relevant patterns from irrelevant patterns.

Point-wise Gated Boltzmann Machines

- Plate notation

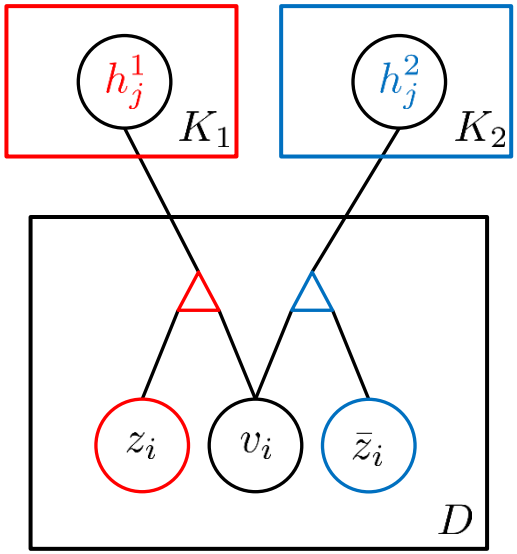


Point-wise Gated Boltzmann Machines (PGBM)

- Modeling with multiple components than two is also possible.

Point-wise Gated Boltzmann Machine

- Representation
  - $\mathbf{v} \in \{0,1\}^D$: binary visible units.
  - $\mathbf{h} \in \{0,1\}^K$: binary hidden units.
  - $\mathbf{z} \in \{0,1\}^D$: binary switch units.



$$P(\mathbf{v},\mathbf{z},\mathbf{h}) = \frac{1}{Z}\exp\left(-E(\mathbf{v},\mathbf{z},\mathbf{h})\right)$$

$$E(\mathbf{v},\mathbf{z},\mathbf{h}) = -\sum_{ij}(z_i v_i)W_{ij}^1 h_j^1 - \sum_{ij}(\bar{z}_i v_i)W_{ij}^2 h_j^2$$

$$- \sum_j b_j^1 h_j^1 - \sum_i (z_i v_i)c_i^1 - \sum_j b_j^2 h_j^2 - \sum_i (\bar{z}_i v_i)c_i^2$$

$$= E(\mathbf{z}\odot\mathbf{v},\mathbf{h}^1) + E(\bar{\mathbf{z}}\odot\mathbf{v},\mathbf{h}^2)$$

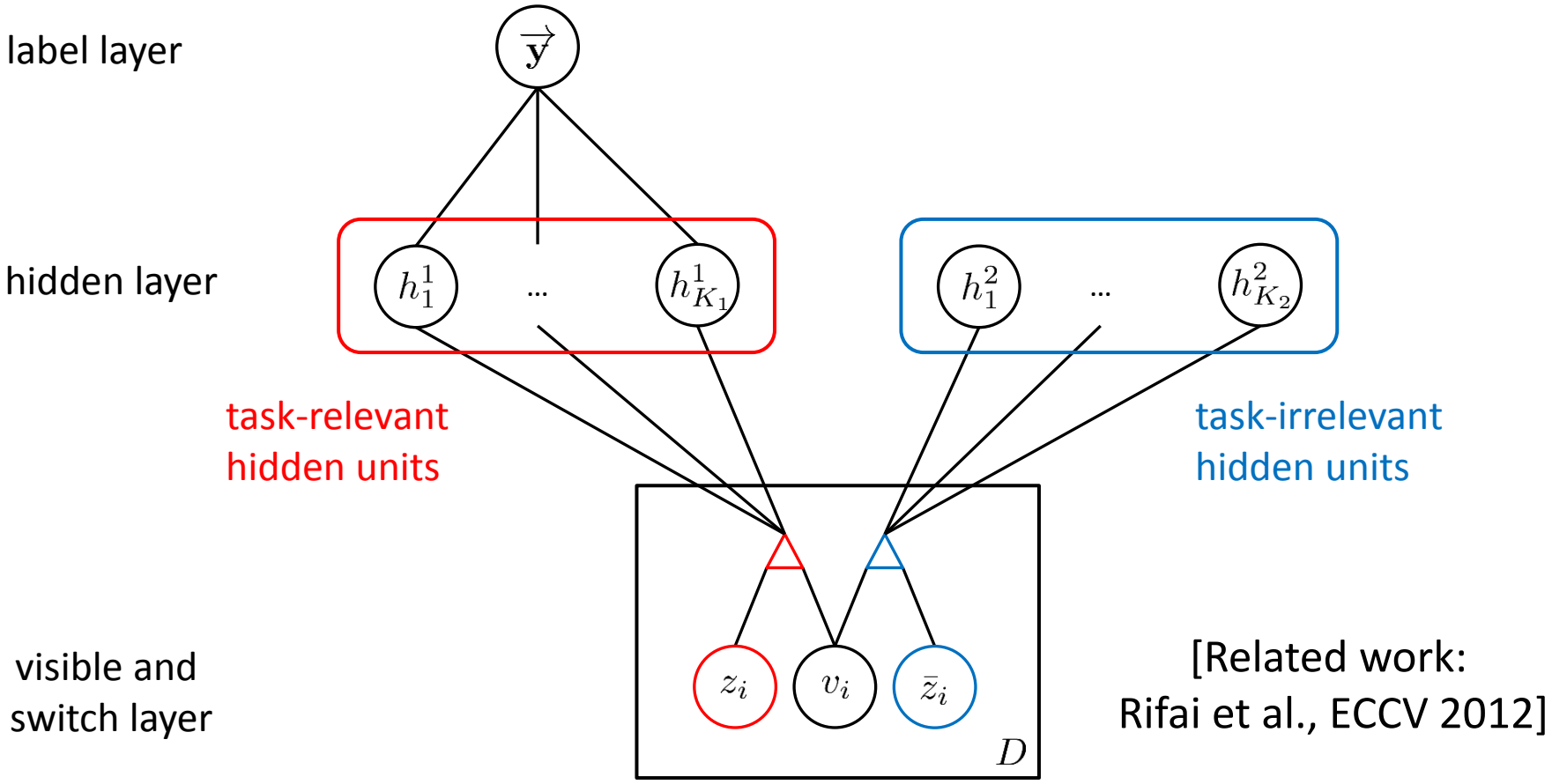
Learning Discriminative Features in PGBM

- PGBM is an unsupervised learning algorithm, and it can only group semantically distinct features with each group of hidden units.
- How to make PGBM to learn discriminative features using class labels?

→ Supervised PGBM

Supervised PGBM

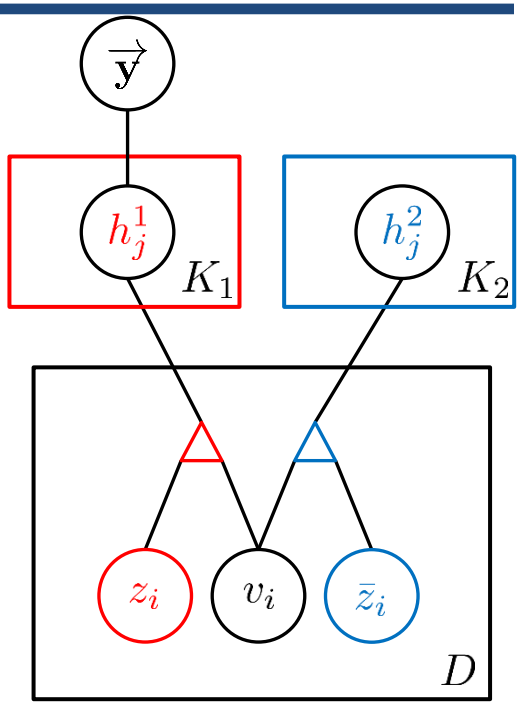
- Labels are connected to one group of hidden units.



[Related work:
Rifai et al., ECCV 2012]

Supervised PGBM

- Representation
  - $\mathbf{v} \in \{0,1\}^D$: binary visible units.
  - $\mathbf{h} \in \{0,1\}^K$: binary hidden units.
  - $\mathbf{z} \in \{0,1\}^D$: binary switch units.
  - $\overrightarrow{\mathbf{y}} \in \{0,1\}^L$: 1-of-*L* label units.



$$P(\mathbf{v}, \mathbf{z}, \mathbf{h}, \overrightarrow{\mathbf{y}}) = \frac{1}{Z} \exp\left(-E(\mathbf{v}, \mathbf{z}, \mathbf{h}, \overrightarrow{\mathbf{y}})\right)$$

$$E(\mathbf{v}, \mathbf{z}, \mathbf{h}, \overrightarrow{\mathbf{y}}) = E(\mathbf{z} \odot \mathbf{v}, \mathbf{h}^1) + E(\bar{\mathbf{z}} \odot \mathbf{v}, \mathbf{h}^2)$$

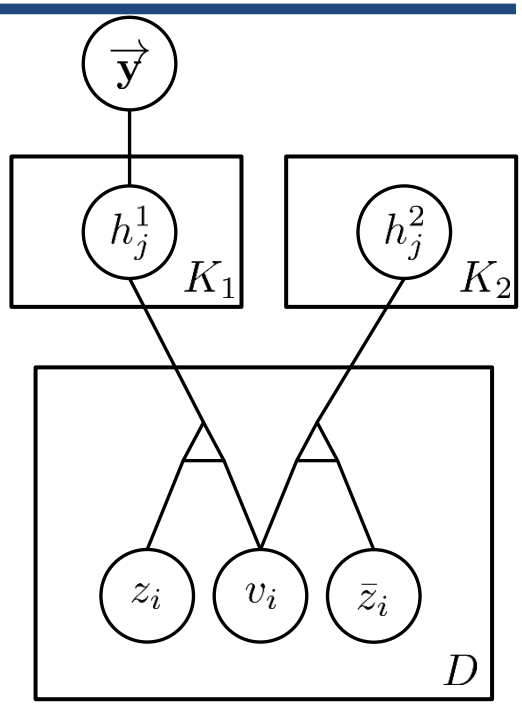
$$- \sum_{jl} h_j^1 U_{jl} y_l - \sum_l y_l d_l$$

Inference and Learning in PGBM

- Inference
  - Mean-field or alternate Gibbs sampling for approximate inference.
  - Conditional independence of single type of variables given other variables.

Inference and Learning in PGBM

- Conditional probabilities

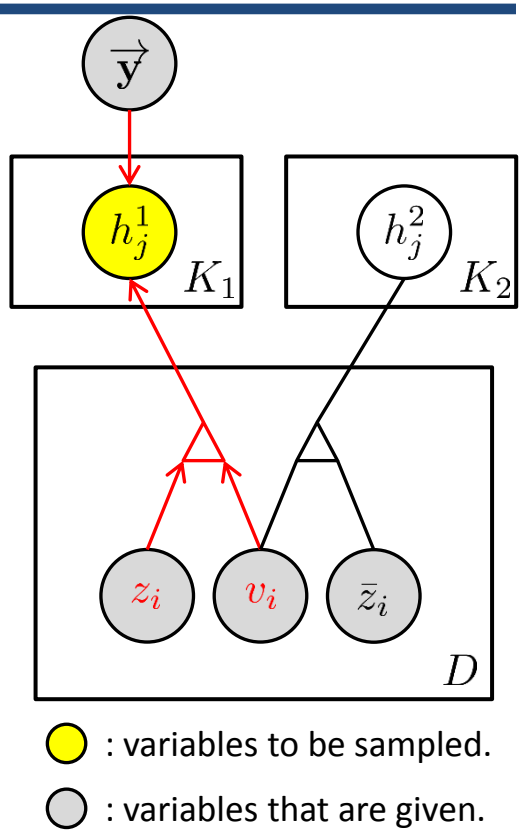


Inference and Learning in PGBM

- Conditional probabilities $P(\mathbf{h}^1|\text{-})$

$$P(h_j^1 = 1|-) \quad = \quad \text{sigmoid}\left(\sum_i z_i v_i W_{ij}^1 + b_j^1 + \sum_l U_{jl} y_l\right)$$

h^1 focus on the *task-relevant* part of the input features that are gated with switch variables.

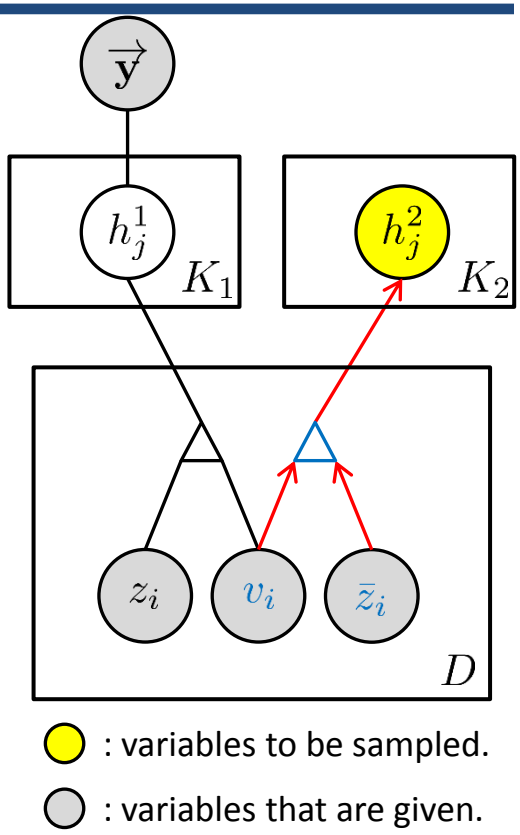


Inference and Learning in PGBM

- Conditional probabilities $P(\mathbf{h}^2|\text{-})$

$$P(h_j^2 = 1|-) \quad = \quad \text{sigmoid}\left(\sum_i \bar{z}_i v_i W_{ij}^2 + b_j^2\right)$$

$\mathbf{h}^2$ focus on the *task-irrelevant* part of the input features that are gated with (complement of) switch variables.



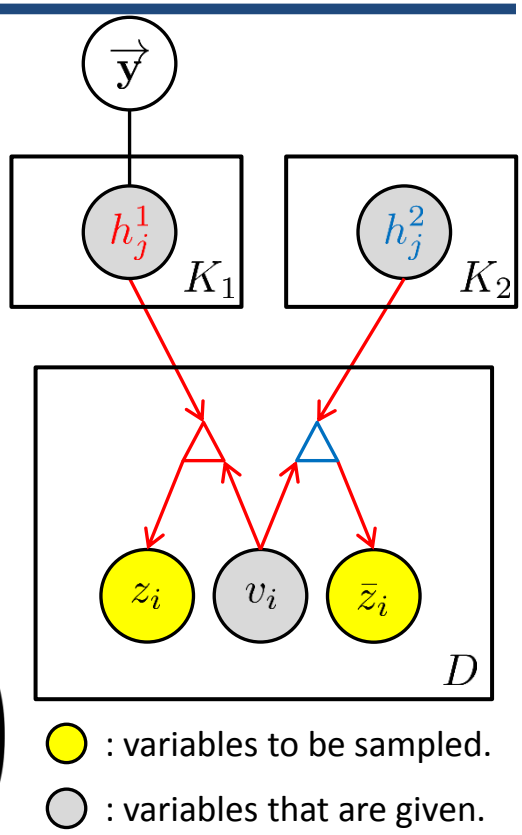
Inference and Learning in PGBM

- Conditional probabilities $P(\mathbf{z}|\text{-})$

$$P(z_i = 1|-)$$

$$= \frac{\exp\left(v_i\left(\sum_j W^1_{ij} h^1_j + c^1_i\right)\right)}{\exp\left(v_i\left(\sum_j W^1_{ij} h^1_j + c^1_i\right)\right) + \exp\left(v_i\left(\sum_j W^2_{ij} h^2_j + c^2_i\right)\right)}$$

$$= \text{sigmoid}\left(v_i\left(\sum_j W^1_{ij} h^1_j + c^1_i\right) - v_i\left(\sum_j W^2_{ij} h^2_j + c^2_i\right)\right)$$



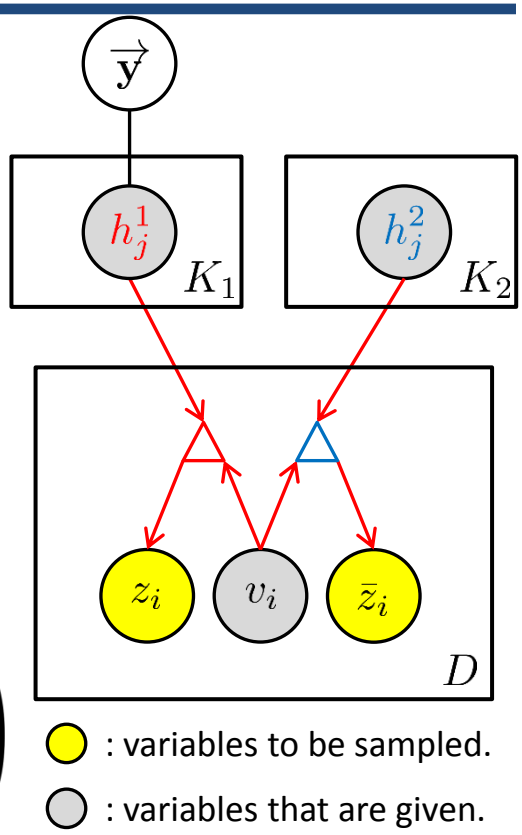
Inference and Learning in PGBM

- Conditional probabilities $P(\mathbf{z}|\text{-})$

$$P(z_i = 1|-)$$

$$= \frac{\exp\left(v_i\left(\sum_j W^1_{ij} h^1_j + c^1_i\right)\right)}{\exp\left(v_i\left(\sum_j W^1_{ij} h^1_j + c^1_i\right)\right) + \exp\left(v_i\left(\sum_j W^2_{ij} h^2_j + c^2_i\right)\right)}$$

$$= \text{sigmoid}\left(v_i\left(\sum_j W^1_{ij} h^1_j + c^1_i\right) - v_i\left(\sum_j W^2_{ij} h^2_j + c^2_i\right)\right)$$



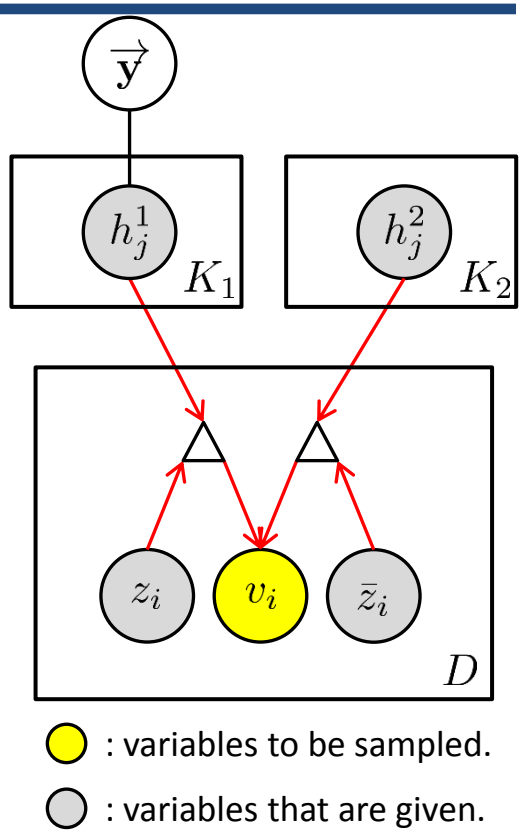
The switch variable is determined through the competition between $\mathbf{h}^1$ and $\mathbf{h}^2$ based on the matching between visible variable and the contribution (reconstruction) from each group of hidden units.

Inference and Learning in PGBM

- Conditional probabilities $P(\mathbf{v}|\text{-})$

$$P(v_i = 1|-)$$
$$= \text{sigmoid}\left(z_i \left(\sum_j W^1_{ij} h^1_j + c^1_i \right) + \bar{z}_i \left(\sum_j W^2_{ij} h^2_j + c^2_i \right) \right)$$

The visible variable is determined with both groups of hidden units.

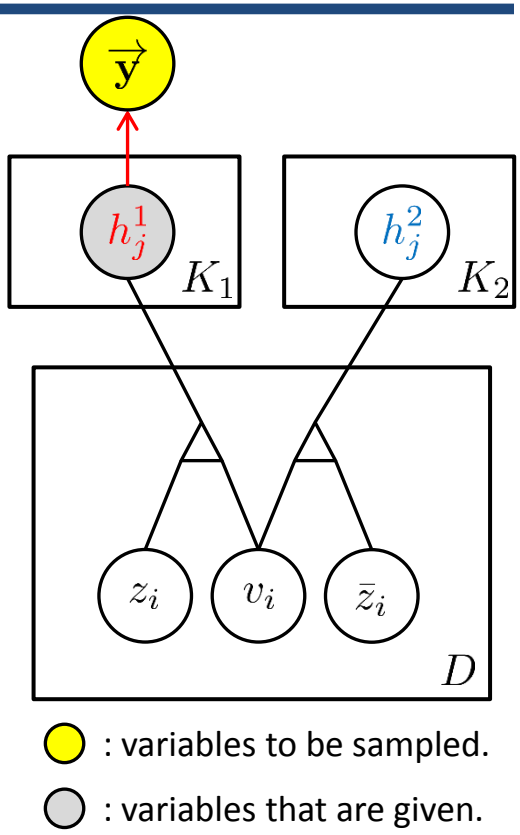


Inference and Learning in PGBM

- Conditional probabilities $P(\mathbf{y}|\text{-})$

$$P(y_l = 1|\mathbf{h}^1) = \text{softmax}\left(\sum_j h_j^1 U_{jl} + d_l\right)$$

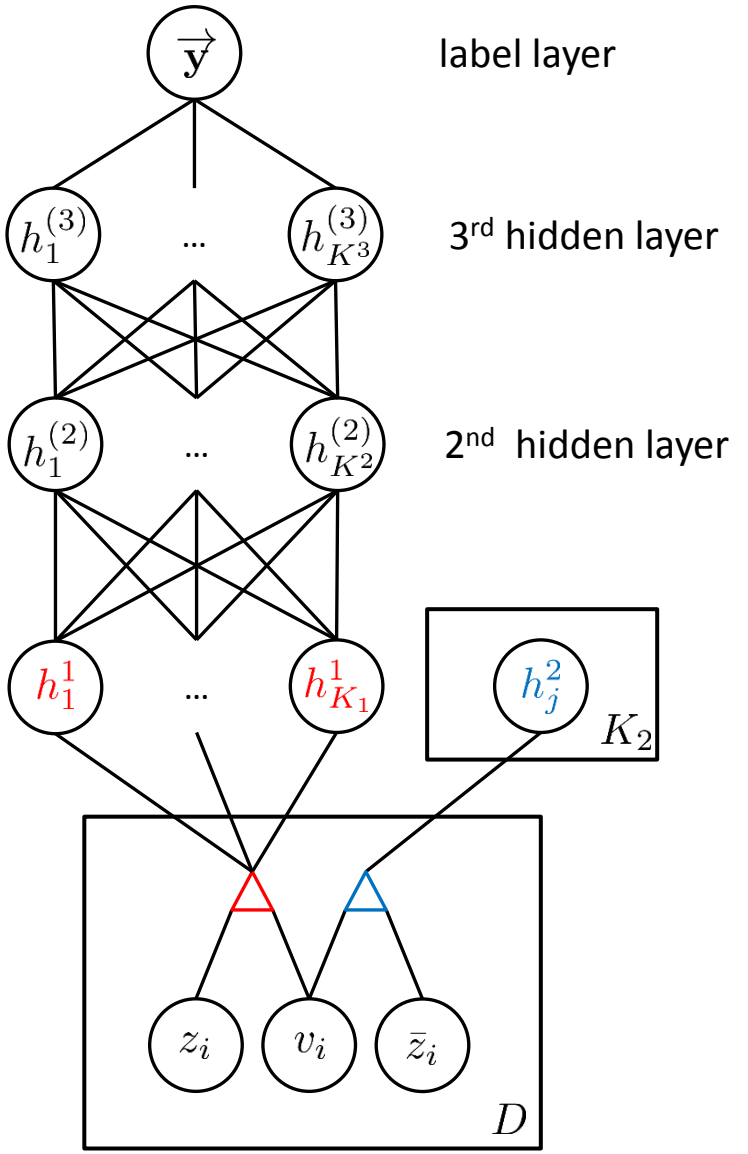
The label variable is inferred only with $\mathbf{h}^1$, not $\mathbf{h}^2$.



Inference and Learning in PGBM

- Inference
  - Mean-field or alternate Gibbs sampling for approximate inference.
  - Conditional independence of single type of variables given other variables.
- Training
  - Maximum-likelihood for joint distribution $P(\mathbf{v}, \overrightarrow{\mathbf{y}})$.
  - Stochastic gradient descent using contrastive divergence.

Extensions – deeper architecture



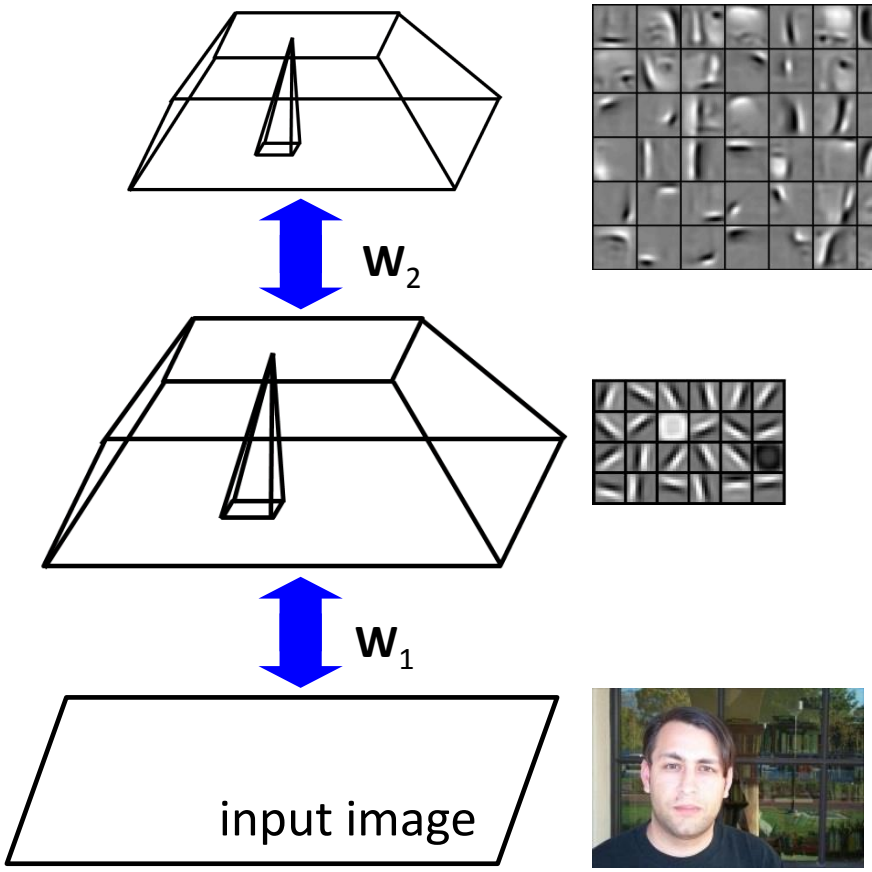
- Propagate only “task-relevant” information to higher layers.
- Stack multiple layers of neural networks on top of task-relevant group of hidden units.

Extensions – convolutional PGBM

- Convolutional Point-wise Gated Deep Network (CPGDN)
  - Convolutional architecture is good at dealing with spatially (or temporally) correlated data.
  - Convolutional deep belief network (CDBN; Lee et al., ICML 2009, Desjardins and Bengio, 2008)

Extensions – convolutional PGBM

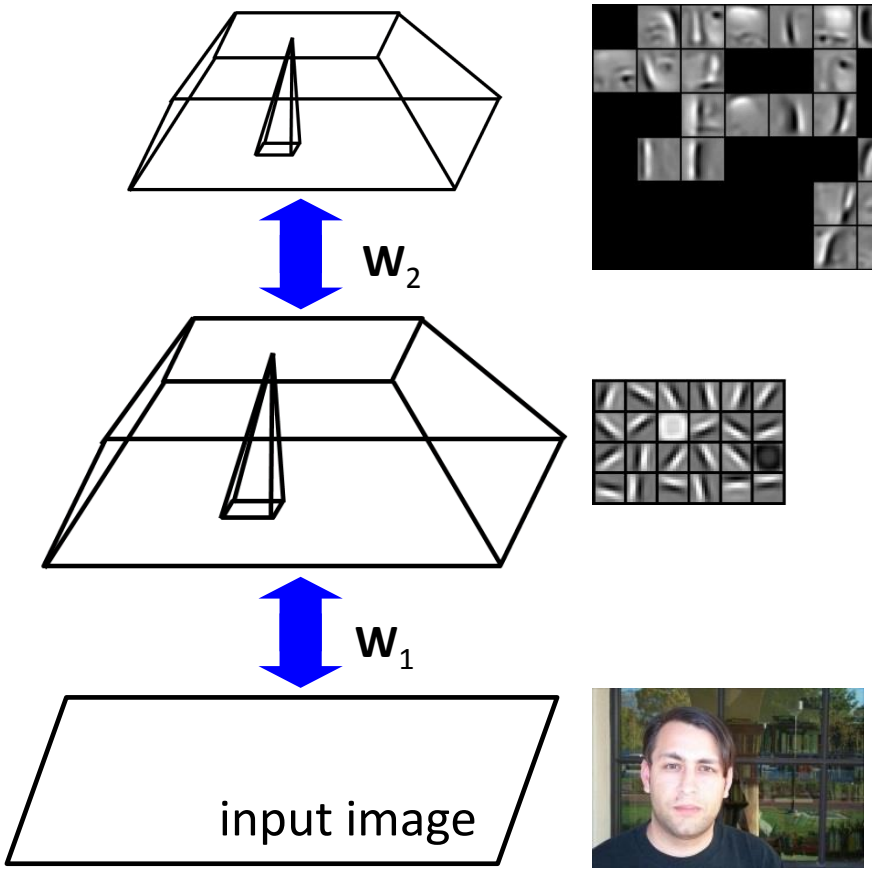
- Low-level features are generic patterns (e.g., edges)
- High-level features are semantically meaningful.



[CDBN, Lee et al., ICML 2009]

Extensions – convolutional PGBM

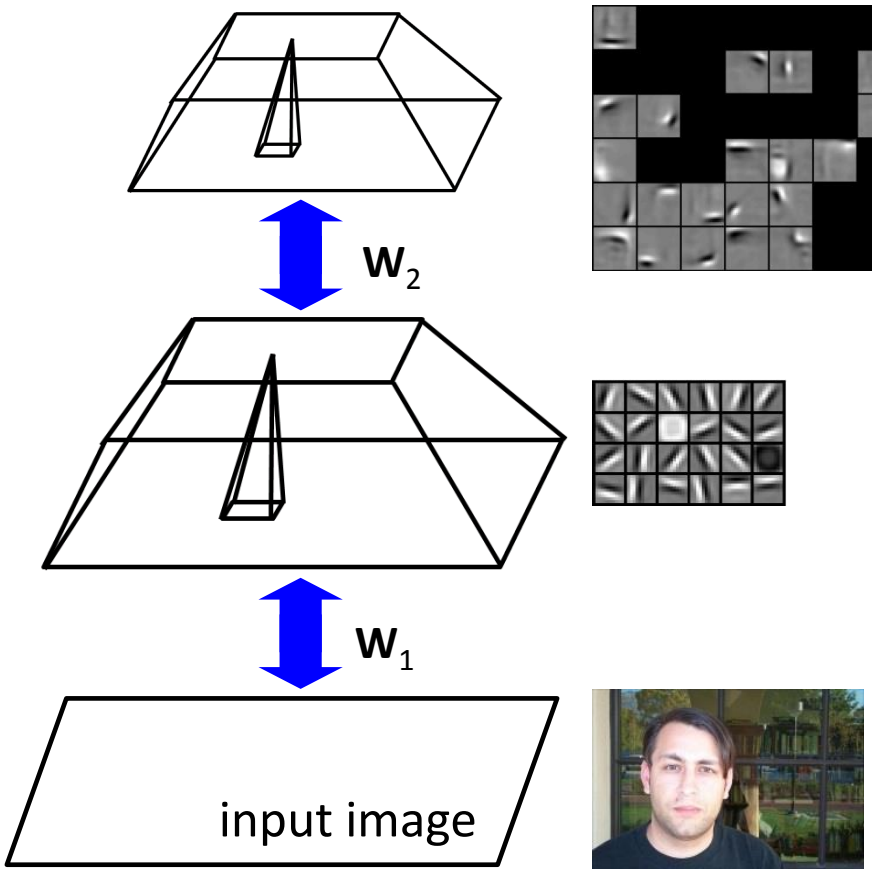
- Low-level features are generic patterns (e.g., edges)
- High-level features are semantically meaningful.



[CDBN, Lee et al., ICML 2009]

Extensions – convolutional PGBM

- Low-level features are generic patterns (e.g., edges)
- High-level features are semantically meaningful.

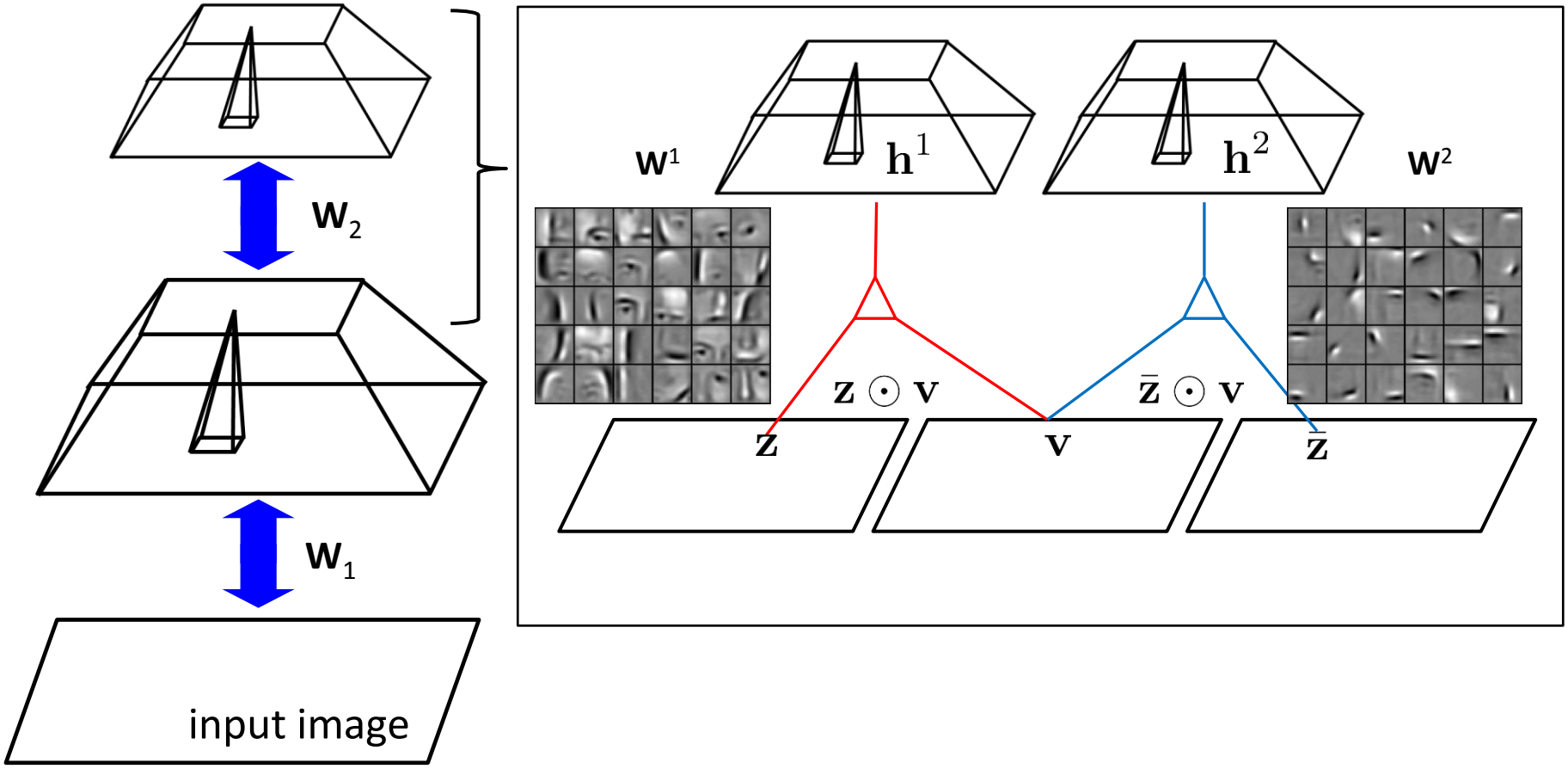


There are some irrelevant patterns as well.

[CDBN, Lee et al., ICML 2009]

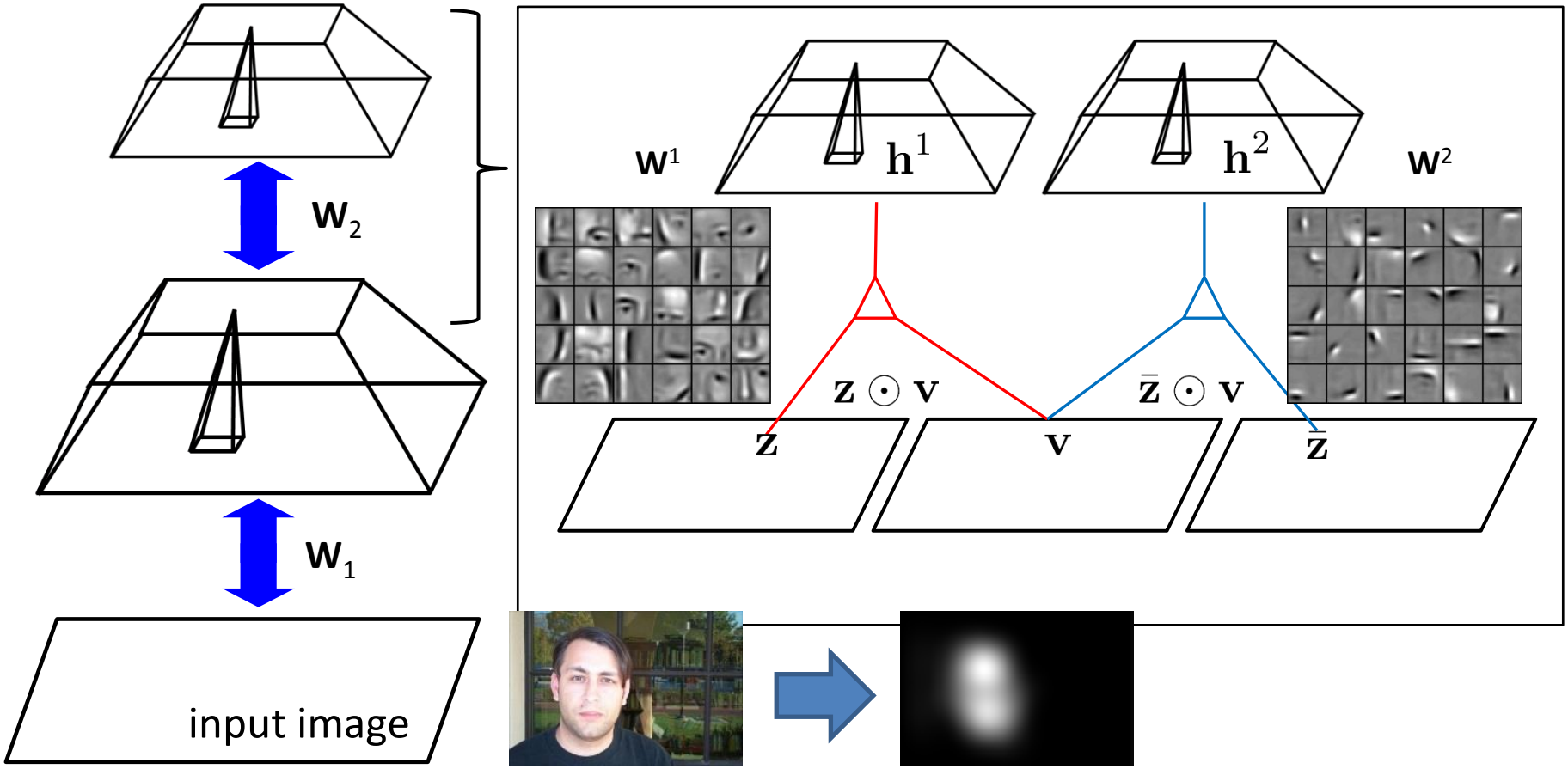
Extensions – convolutional PGBM

– We can distinguish between task-relevant and irrelevant features with point-wise gating idea while feature learning.



Extensions – convolutional PGBM

– We can distinguish between task-relevant and irrelevant features with point-wise gating idea while feature learning.



Experiments

- Task 1: handwritten digit recognition in the presence of background noise.

- Task 2: learning from large images with cluttered background in application to weakly supervised object localization and object recognition.

Experiments – variations of MNIST with background noise

- Recognizing handwritten digits in the presence of background noise.
  - uniform random noise or natural images in the background.
  - rotation transformations are applied.

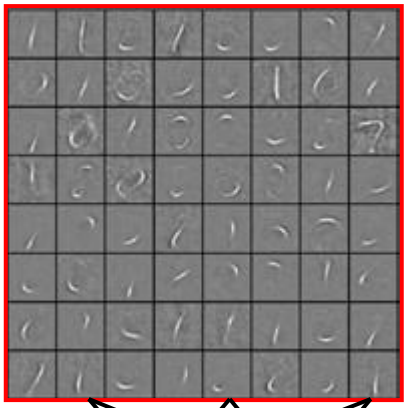
original back-rand back-image rot-back-rand rot-back-image

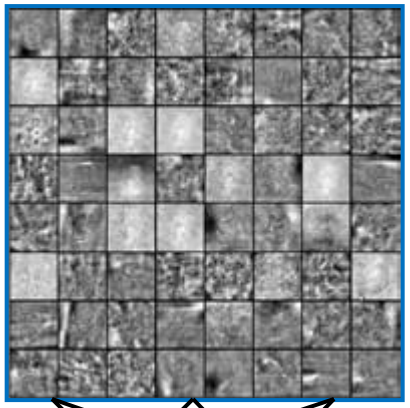
  - Due to significant amount of distracting factors, learning good features become much more challenging, and this results in poor recognition performance.

Experiments – visualizations

- Learning from noisy handwritten digits with PGBM

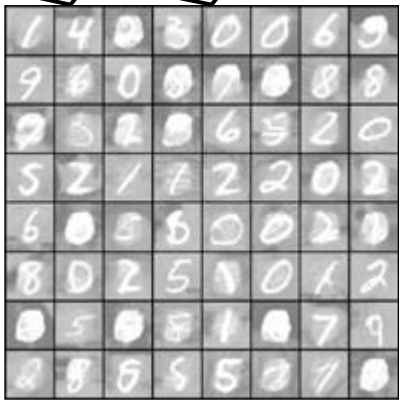
Learned task-relevant hidden unit weights: mostly ***pen-strokes***





Learned task-irrelevant hidden unit weights: noisy patterns

Noisy digit images (mnist-back-image)



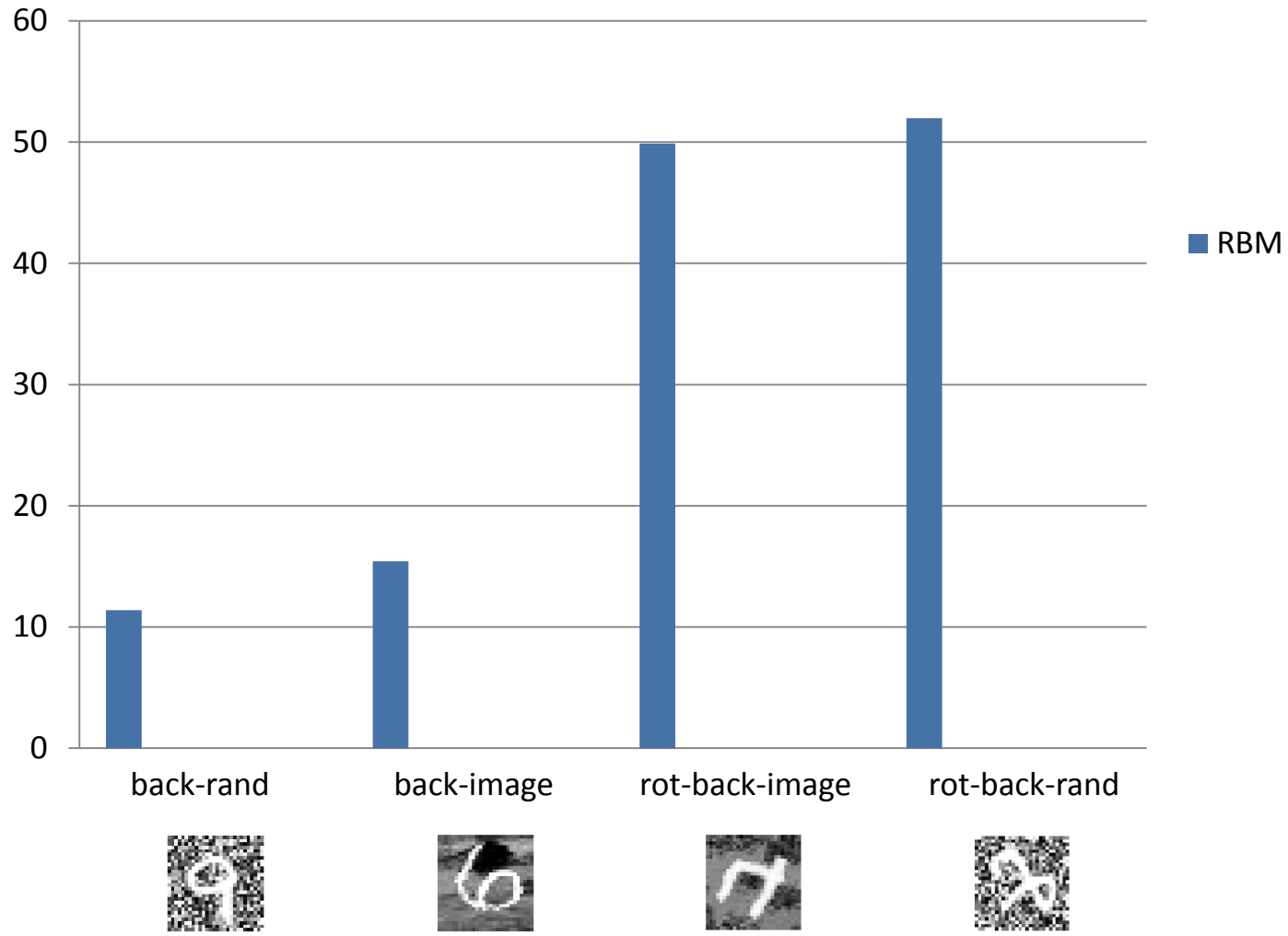
Inferred switch variables

Experiments – digit recognition

- Handwritten digit recognition
  - Evaluated on variations of MNIST.
  - Compared with several variations of RBMs:
    - Standard RBM.
    - Implicit mixture of RBM (imRBM; Nair and Hinton, NIPS 2008) – multiple groups of hidden units.
    - Discriminative RBM (discRBM; Larochelle and Bengio, ICML 2008) – supervised, semi-supervised training.
    - Standard RBM + feature selection.

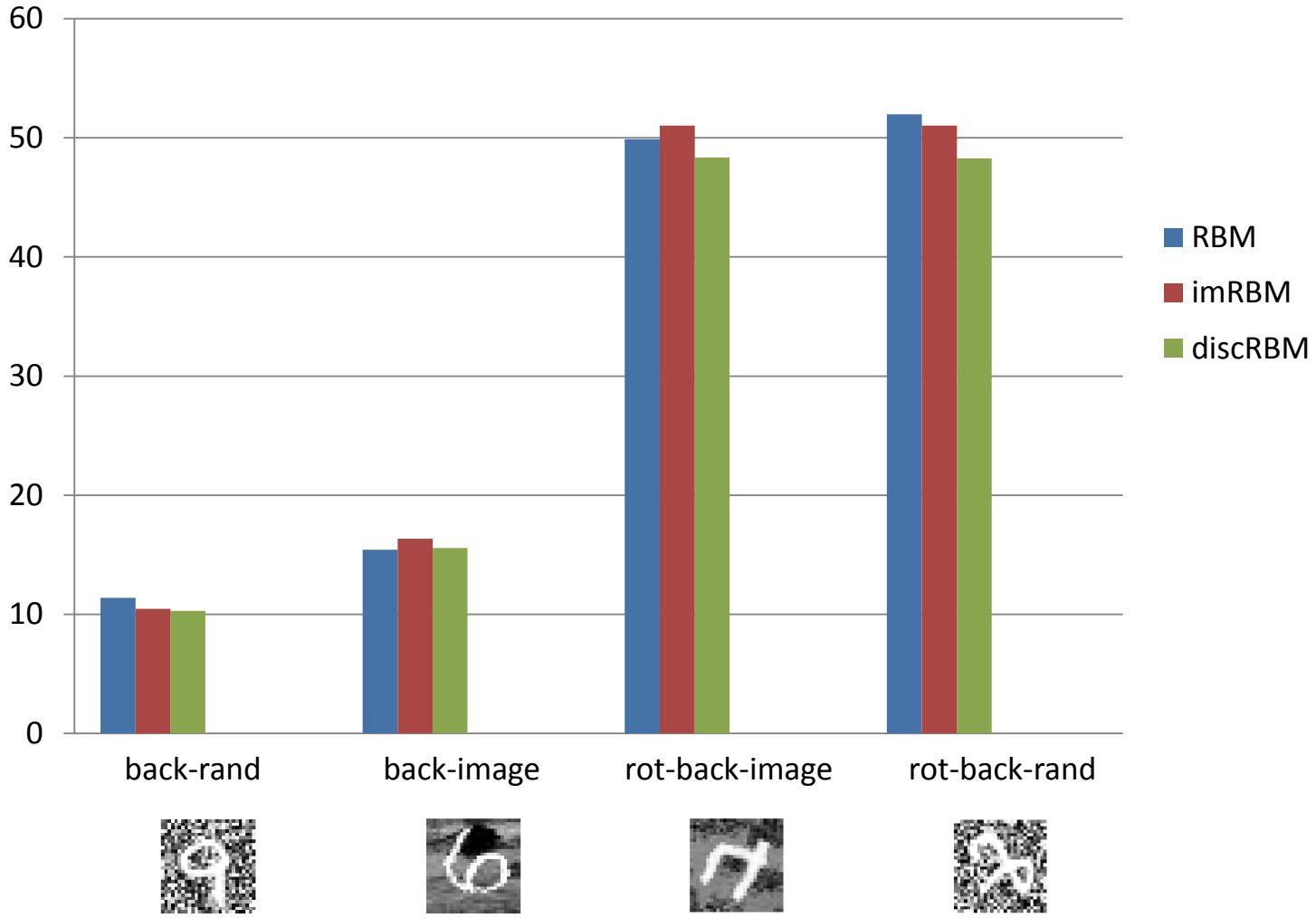
Experiments – digit recognition

- Handwritten digit recognition error rates



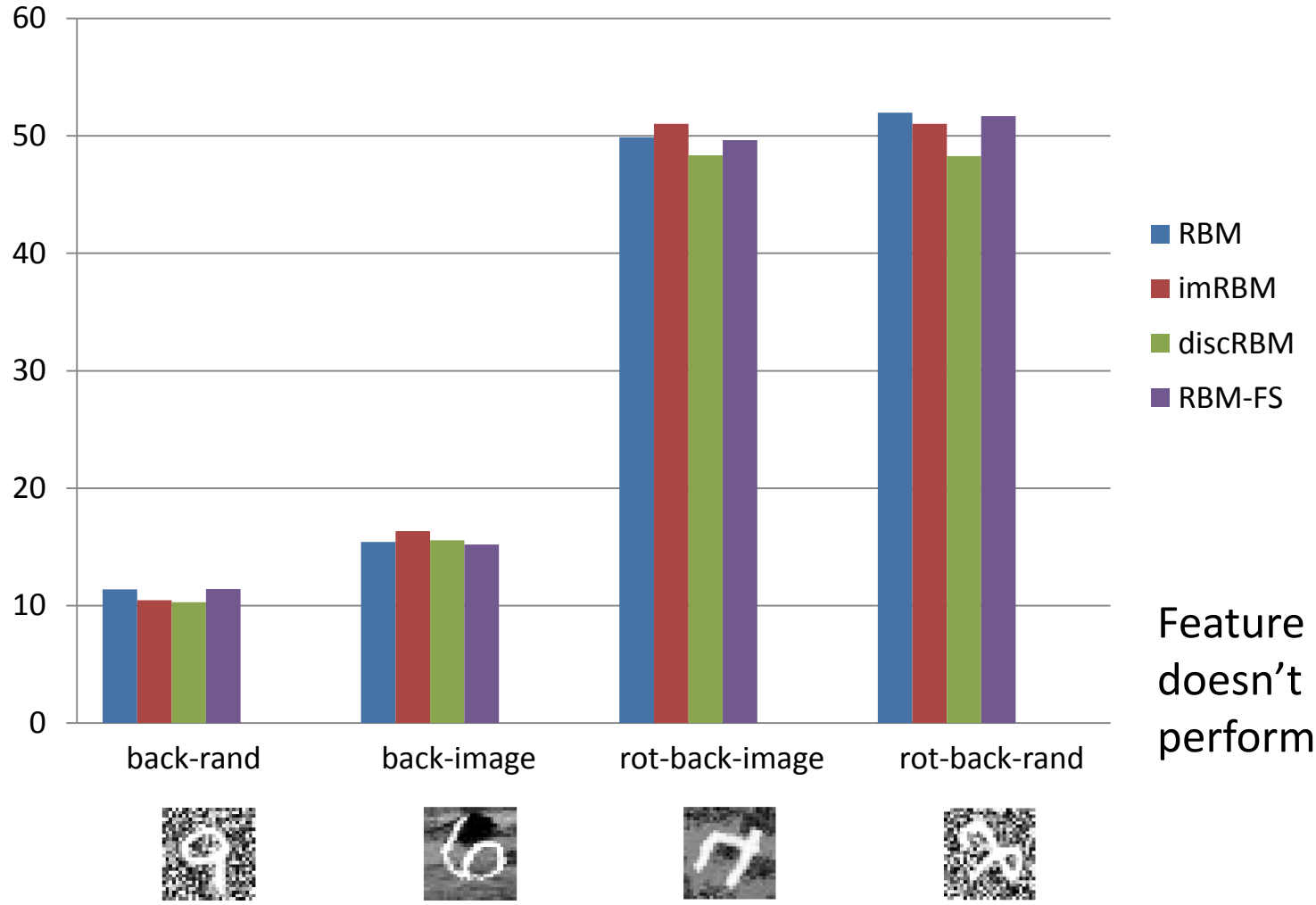
Experiments – digit recognition

- Handwritten digit recognition error rates



Experiments – digit recognition

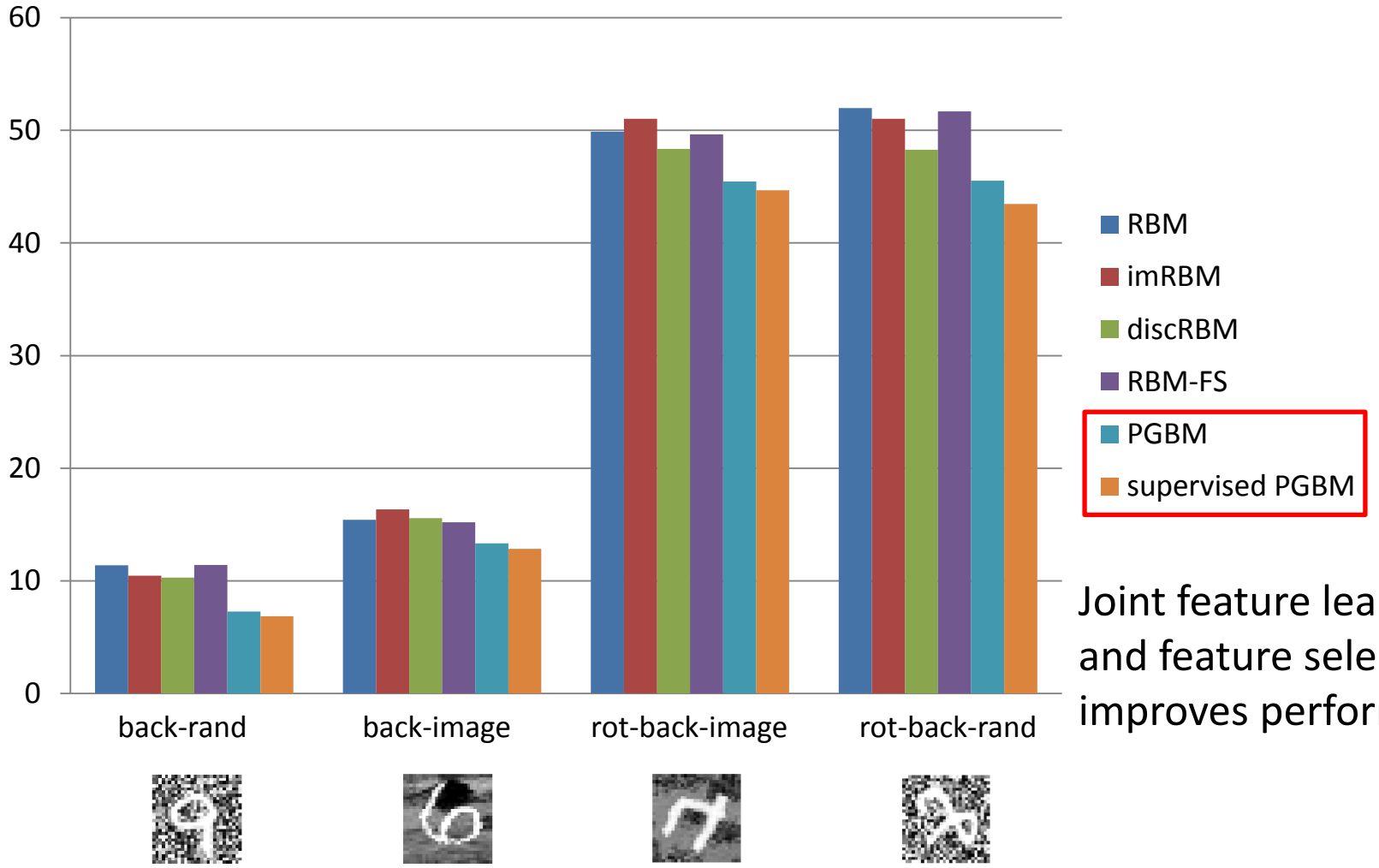
- Handwritten digit recognition error rates



Feature selection doesn't improve the performance.

Experiments – digit recognition

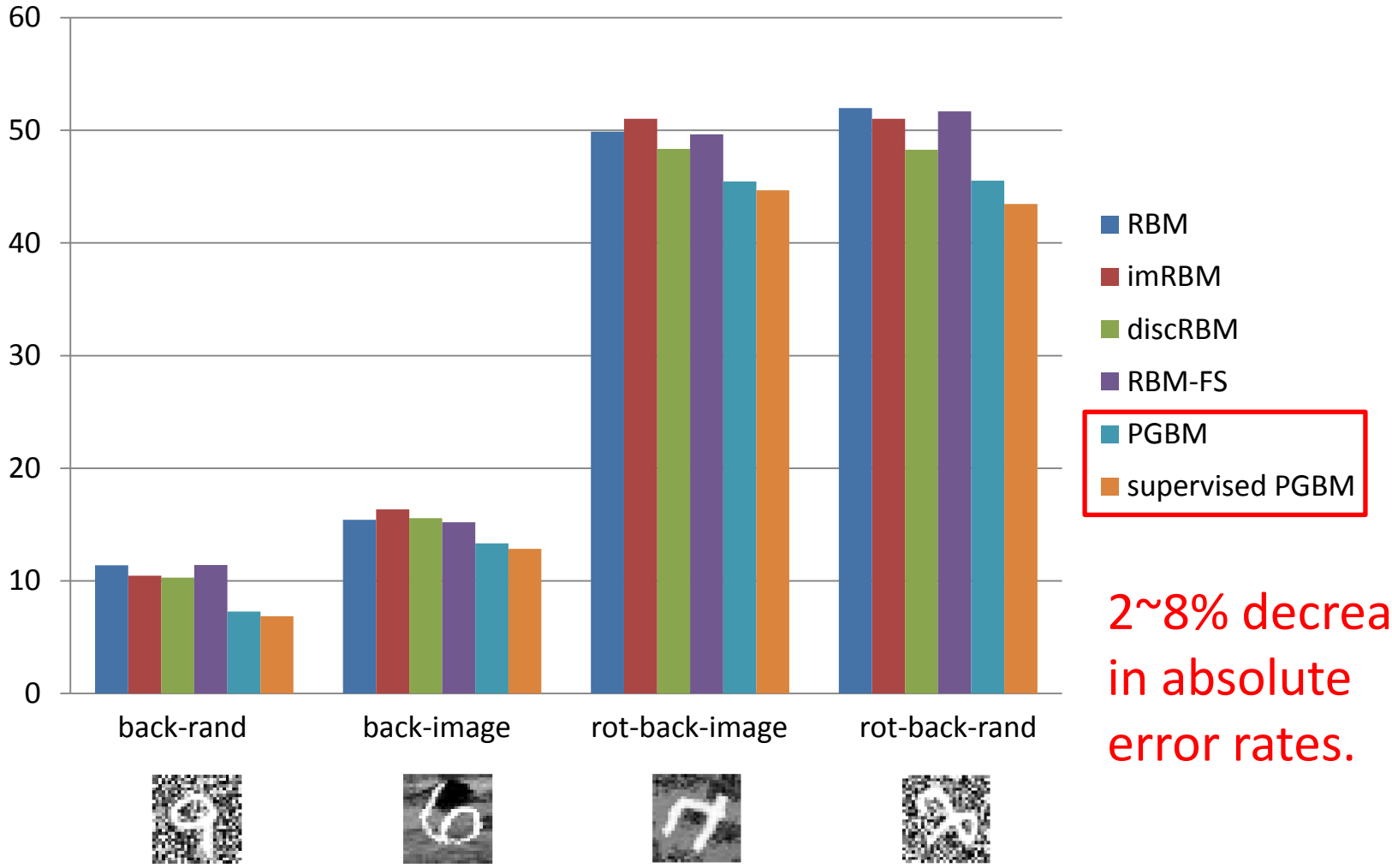
- Handwritten digit recognition error rates



Joint feature learning and feature selection improves performance.

Experiments – digit recognition

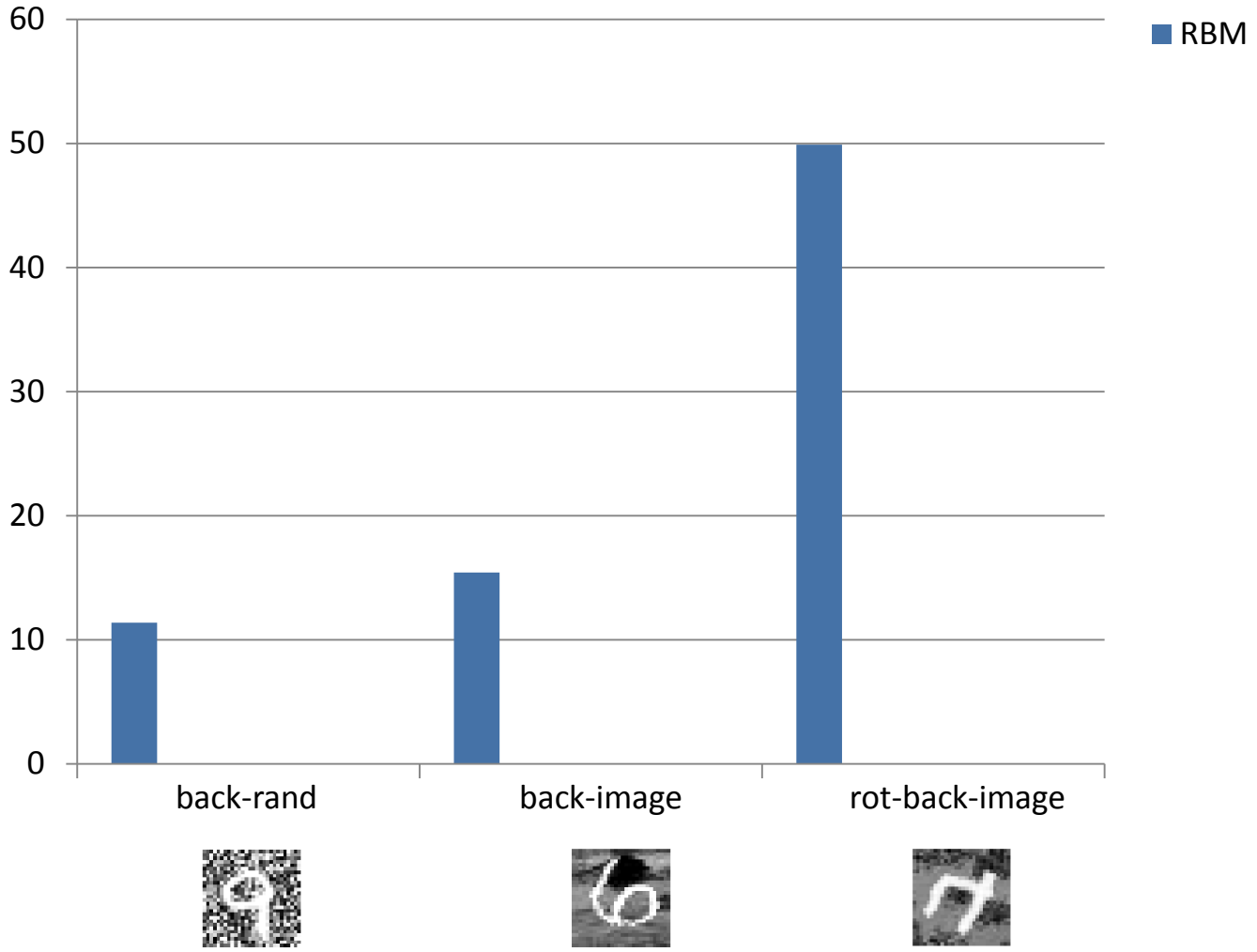
- Handwritten digit recognition error rates



2~8% decrease in absolute error rates.

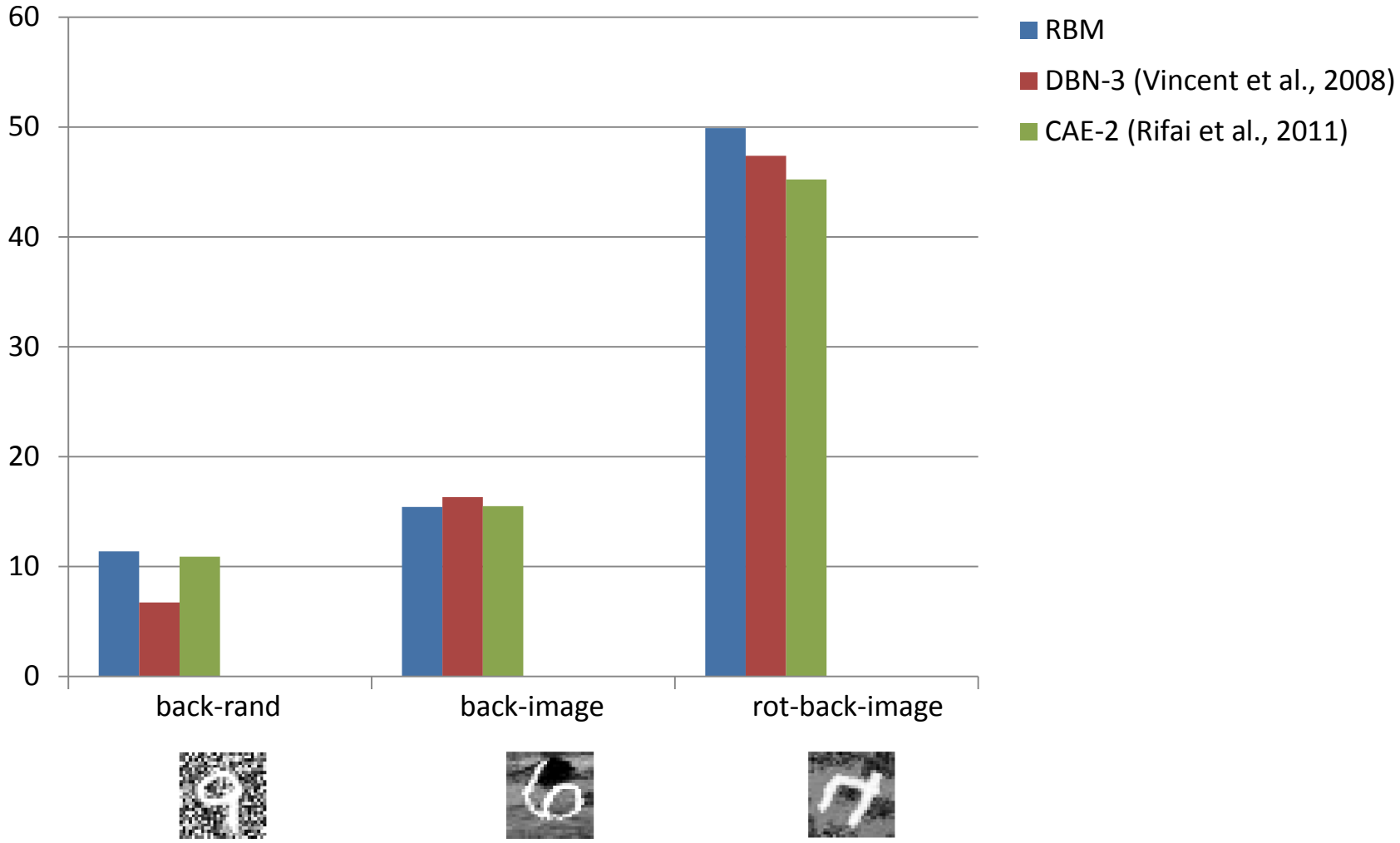
Experiments – digit recognition

- Comparison to other deep learning methods



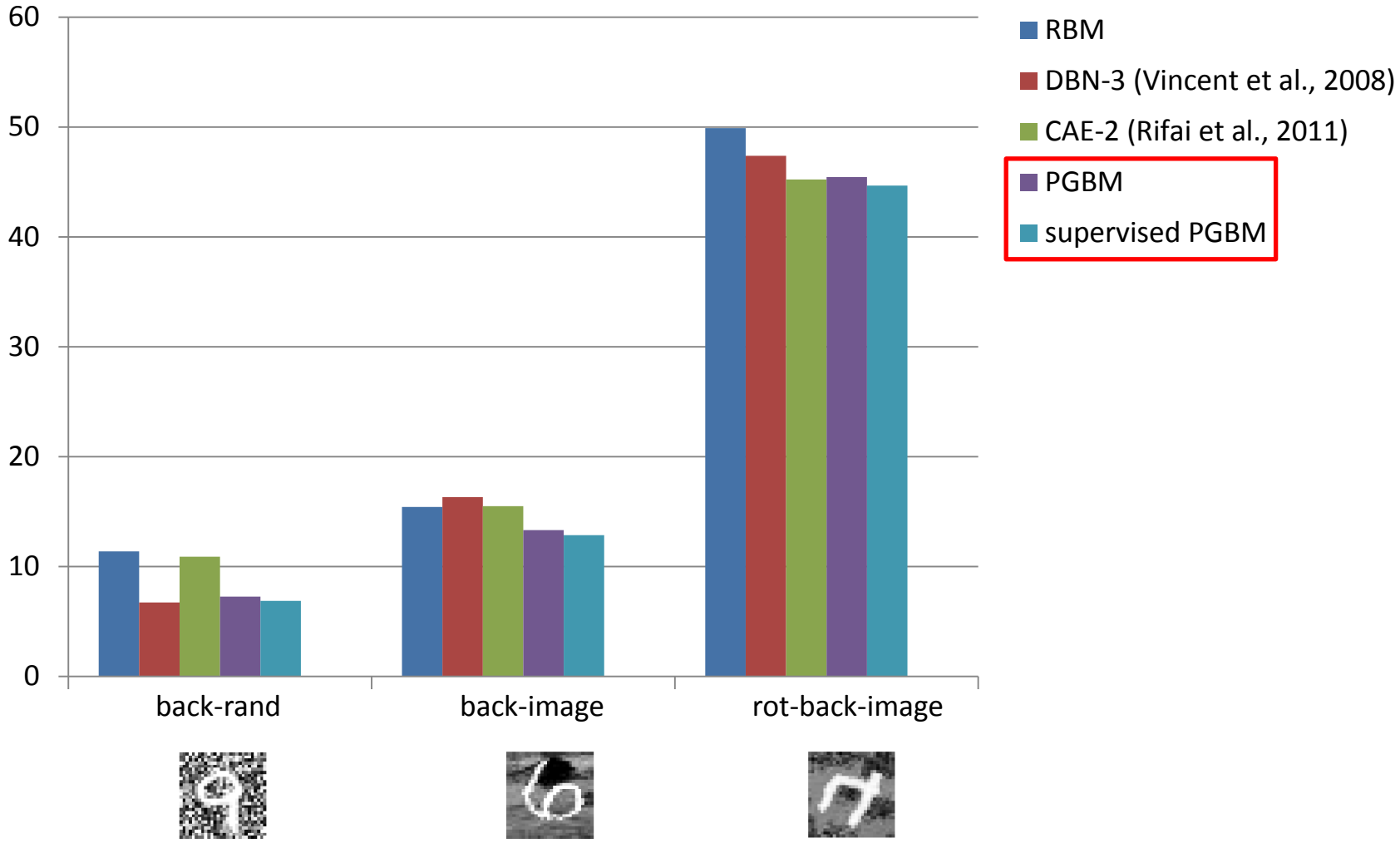
Experiments – digit recognition

- Comparison to other deep learning methods



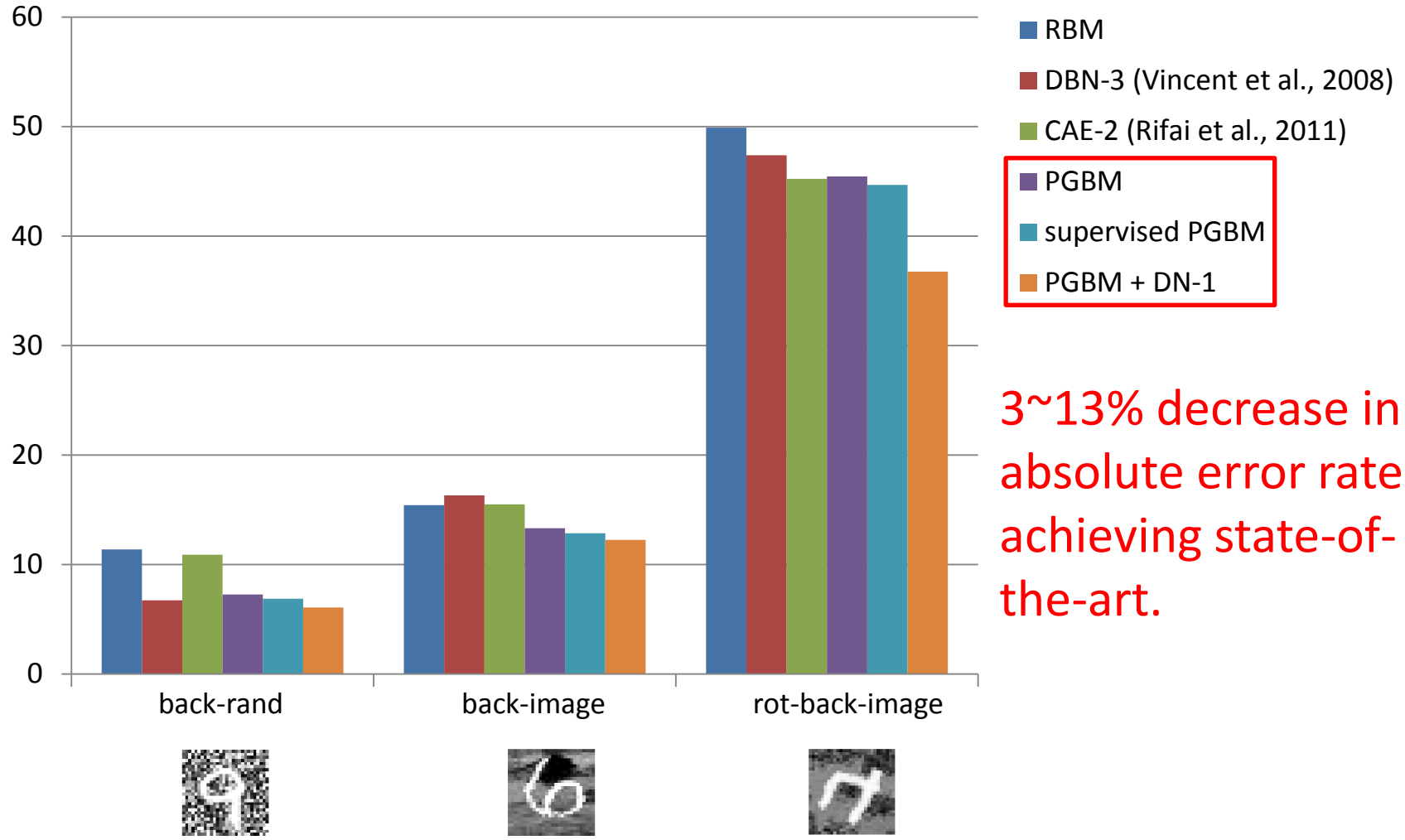
Experiments – digit recognition

- Comparison to other deep learning methods



Experiments – digit recognition

- Comparison to other deep learning methods



3~13% decrease in absolute error rates, achieving state-of-the-art.

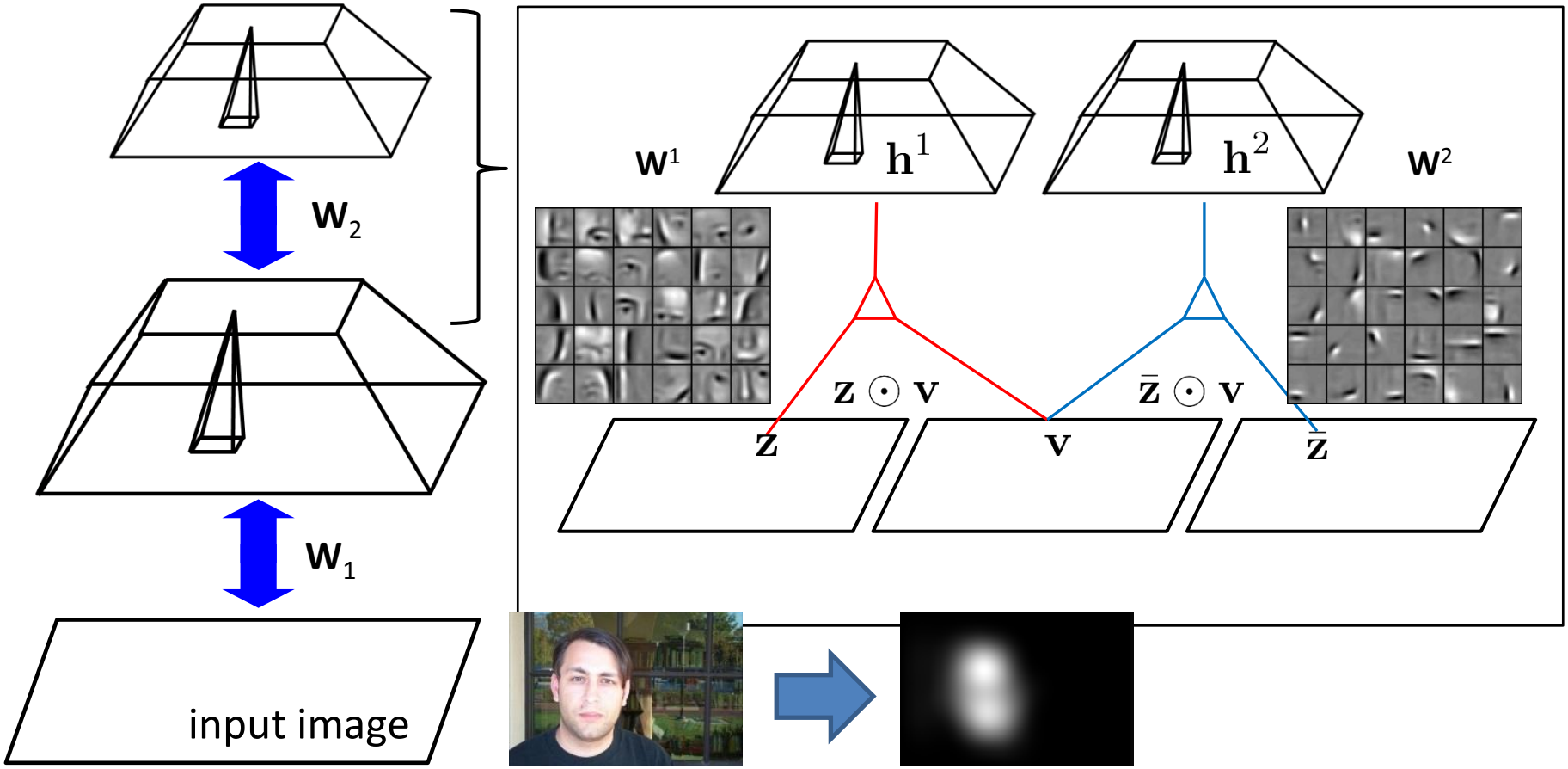
Experiments – learning from large images with cluttered background

- Given cluttered, high-resolution images, how can we find relevant foreground features?
  - Weakly supervised setting (no bounding box is given).

- Convolutional point-wise gating for generative feature selection while feature learning from large images.

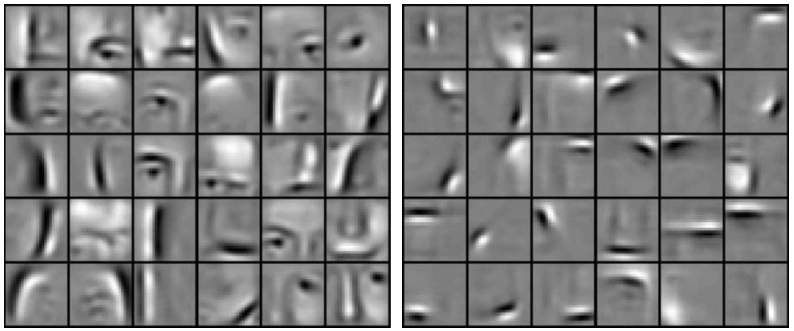
Experiments – CPGDN

- CPGDN on Caltech 101 dataset

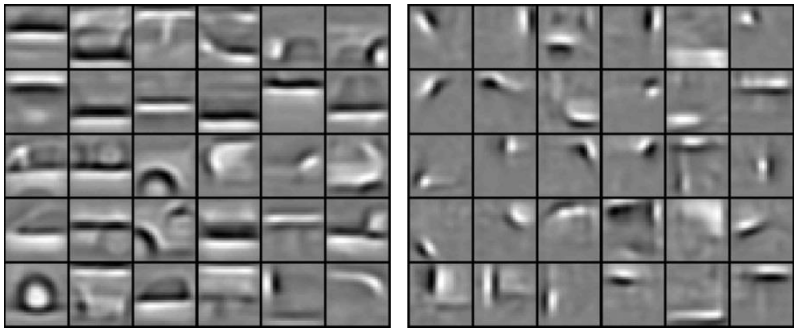


Experiments – weakly supervised object segmentation

- Learned set of filters (task-relevant/irrelevant)



Caltech101 - Faces



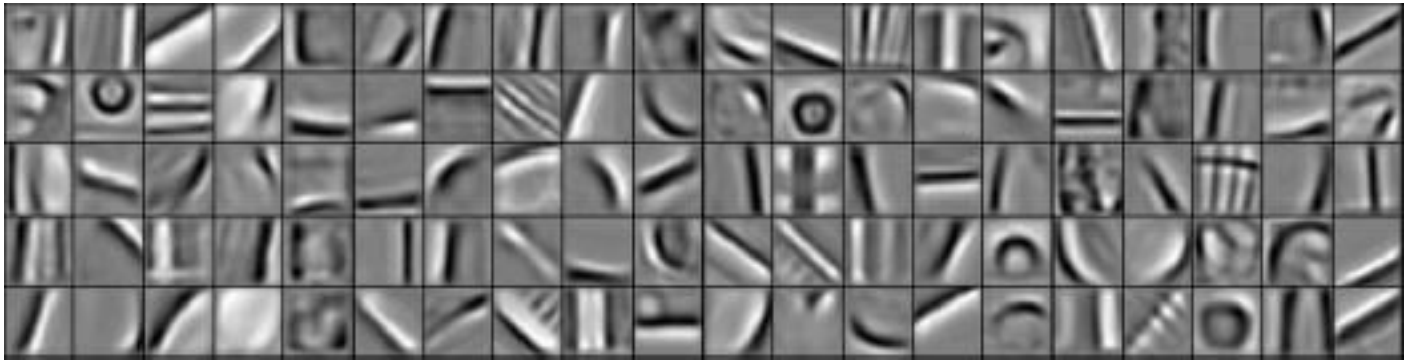
Caltech101 – car side

- (Weakly supervised) object localization

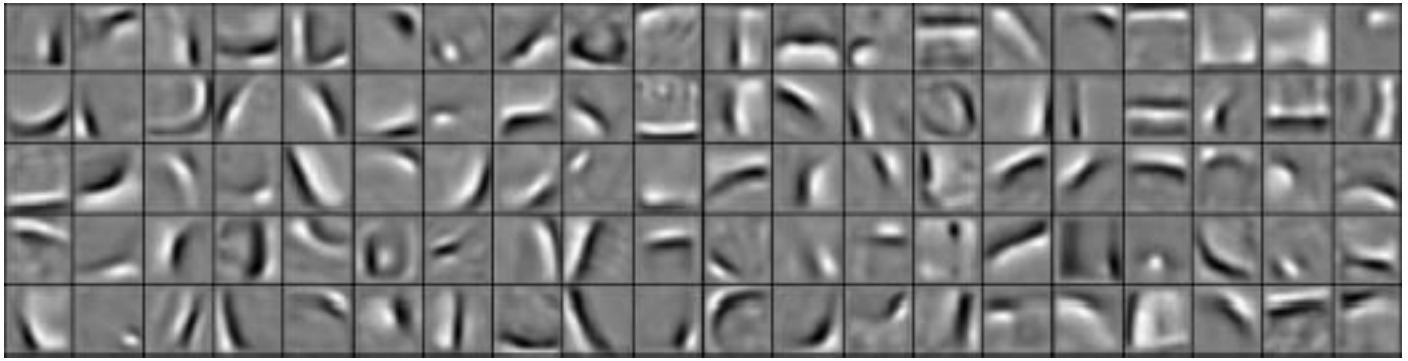
1st row: switch unit activation map,
2nd row: predicted and ground truth bounding box.

Experiments – weakly supervised object segmentation

- Learned set of bases from 101 classes

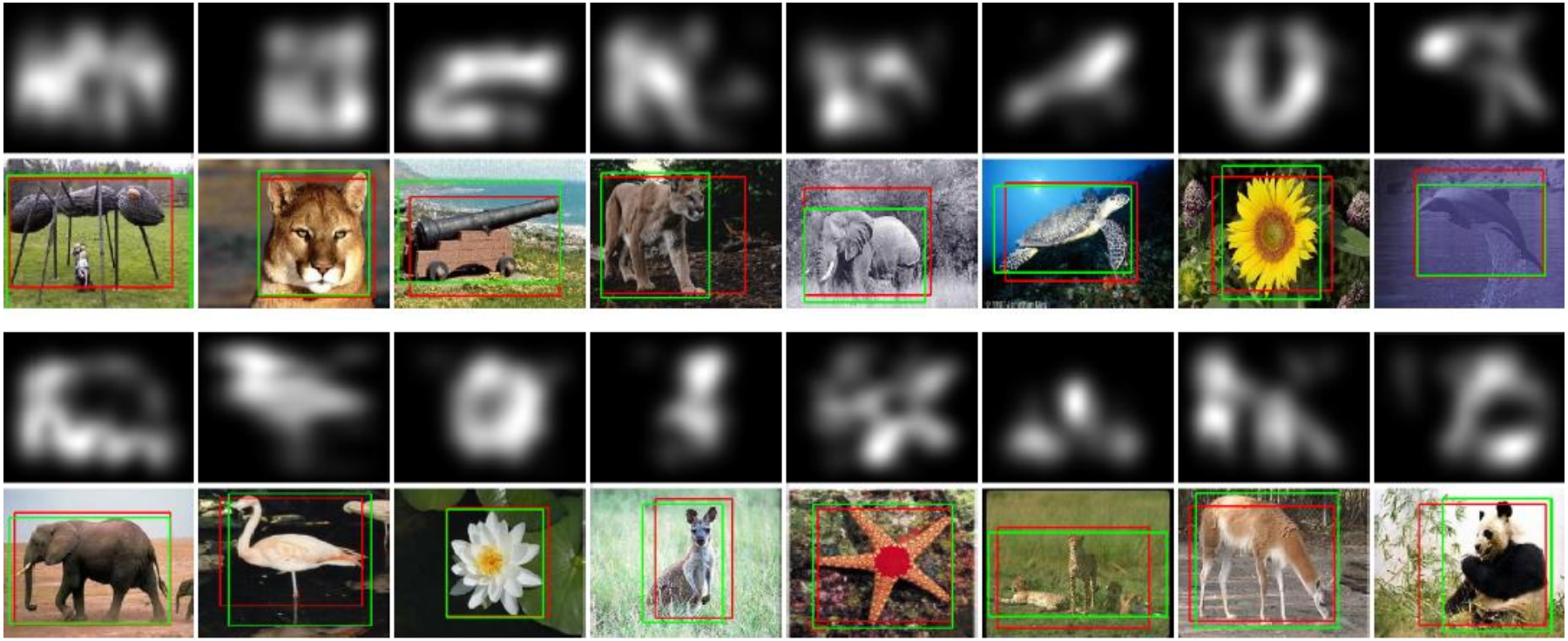


Caltech101 – task-relevant



Caltech101 – task-irrelevant

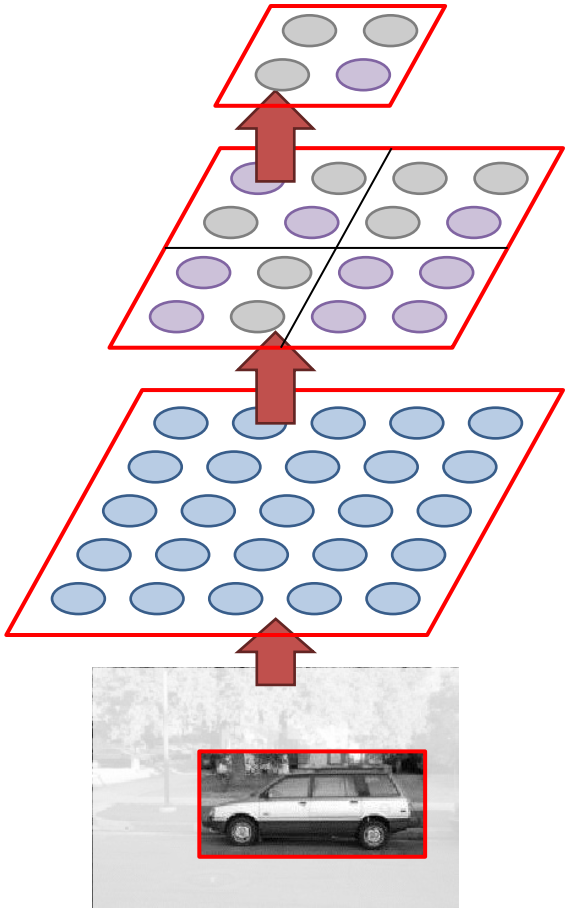
Experiments – weakly supervised object segmentation



1st row: switch unit activation map,
2nd row: predicted and ground truth bounding box.

Experiments – object recognition

- Object recognition with predicted bounding boxes.



4. Spatial pyramid pooling, followed by linear SVM.

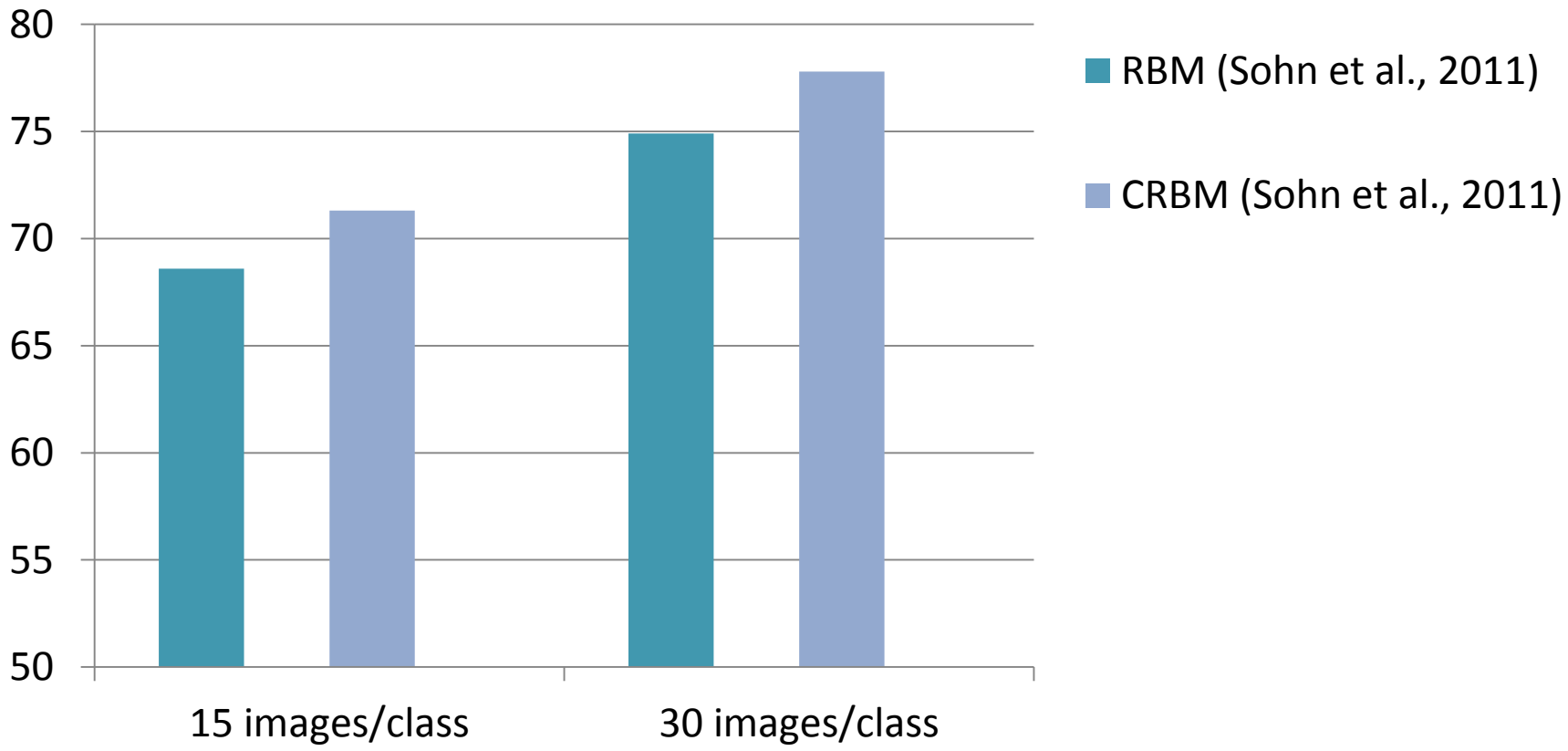
3. Feature encoding with Gaussian RBM or CRBM (Sohn et al., ICCV 2011).

2. Dense SIFT feature extraction from cropped image.

1. Bounding box prediction using CPGDN.

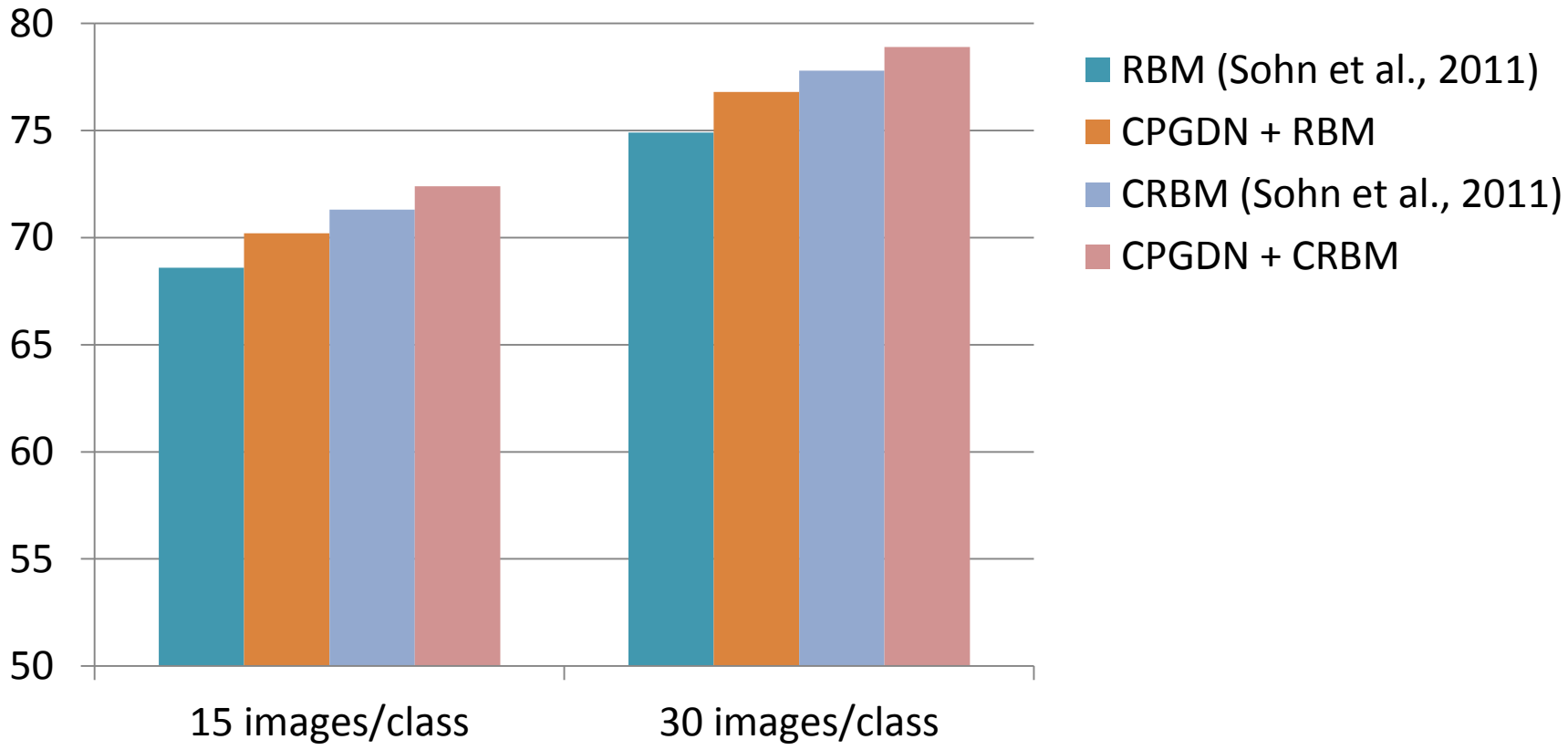
Experiments – object recognition

- Classification accuracy on Caltech 101 dataset



Experiments – object recognition

- Classification accuracy on Caltech 101 dataset



Experiments – object recognition

- Comparison to other results on Caltech 101 dataset
  - With single type of features.

Training images	15	30
VQ (Lazebnik et al., 2006)	56.4%	64.6%
ScSPM (Yang et al., 2009)	67.0%	73.2%
LLC (Wang et al., 2010)	65.4%	73.4%
Macrofeature (Boureau et al., 2010)	-	75.7%
CRBM (Sohn et al., 2011)	71.3%	77.8%
Sparse and Selective RBM (Goh et al., 2012)	71.1%	**78.9%**
CPGDN + CRBM (ours)	**72.4%**	**78.9%**

  - MKL (Yang et al., 2009): 84.3%
  - Multipath sparse coding (Bo et al., 2013): 82.5%
  - ….

Conclusion

- We propose the PGBMs that jointly perform the feature learning and feature selection in a unified framework.

- The PGBM effectively learn useful representations from the data containing significant irrelevant or distracting patterns.

Thank you

demo code is available:
http://umich.edu/~kihyuks/pubs/pgbm.zip