电子科技大学
University of Electronic Science and Technology of China

Net2Vec: towards Knowledge Embedding (Unsupervised via Sync)

Wei Han

Data Mining Lab,
Big Data Research Center, UESTC
Email：weihan@std.uestc.edu.cn

Outline

- Preliminary: Network Interpretability
- Net2Vec: Supervised Knowledge Embedding
- Our proposals: Unsupervised Knowledge Embedding
- Understanding Sync Mechanism

Preliminary:

Network Interpretability

Preliminary

- Why interpretability?
  - Incompleteness problems
  - High Reliability Requirement
  - Ethical and Legal Requirement
  - Interactions and Social acceptance
  - Debugged and Audited

Taxonomy of Interpretability Methods

Dimension 1 — Passive vs. Active Approaches	
Passive	Post-hoc explain trained neural networks
Active	Actively change the network architecture or training process for better interpretability
Dimension 2 — Type of Explanations (in the order of increasing explanatory power)	
To explain a prediction/class by	
Examples	Provide example(s) which may be considered similar or as prototype(s)
Attribution	Assign credit (or blame) to the input features (e.g. feature importance, saliency masks)
Hidden semantics	Make sense of certain hidden neurons/layers
Rules	Extract logic rules (e.g. decision trees, rule sets and other rule formats)
Dimension 3 — Local vs. Global Interpretability (in terms of the input space)	
Local	Explain network's *predictions on individual samples* (e.g. a saliency mask for a input image)
Semi-local	In between, for example, explain a group of similar inputs together
Global	Explain the network *as a whole* (e.g. a set of rules/a decision tree)

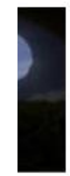
■ Research Hotspots

		Local	Semi-local	Global
Passive	**Rule**	CEM [38], CDRPs [47]	Anchors [34], Interpretable partial substitution [48]	KT [49], M of N [50], NeuralRule [51], NeuroLinear [52], GRG [53], Gyan^{FO} [54], •^{FZ} [55], [56], Trepan [57], • [58], DecText [59], Global model on CEM [60]
	Hidden semantics	(*No explicit methods but many in the below cell can be applied here.)	—	Visualization [40], [61]–[66], Network dissection [18], Net2Vec [67], Linguistic correlation analysis [68]
	Attribution[1]	LIME [17], MAPLE [69], Partial derivatives [40], DeconvNet [41], Guided backprop [70], Guided Grad-CAM [71], Shapley values [72]–[75], Sensitivity analysis [41], [76], [77], Feature selector [78], CVE[2] [79], Bias attribution [80]	DeepLIFT [81], LRP [82], Integrated gradients [83], Feature selector [78], MAME [37]	Feature selector [78], TCAV [42], ACE [84], SpRAy[3] [36], MAME [37]
	By example	Influence functions [43], Representer point selection [85]	—	—
Active	**Rule**	—	Regional tree regularization [86]	Tree regularization [87]
	Hidden semantics	—	—	"One filter, one concept" [39]
	Attribution	ExpO [88], DAPr [89]	—	Dual-net (feature importance) [90]
	By example	—	—	Network with a prototype layer [46], ProtoPNet [91]

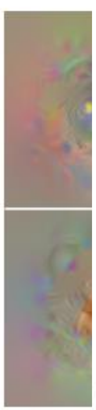
Background

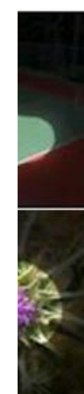
■ Why

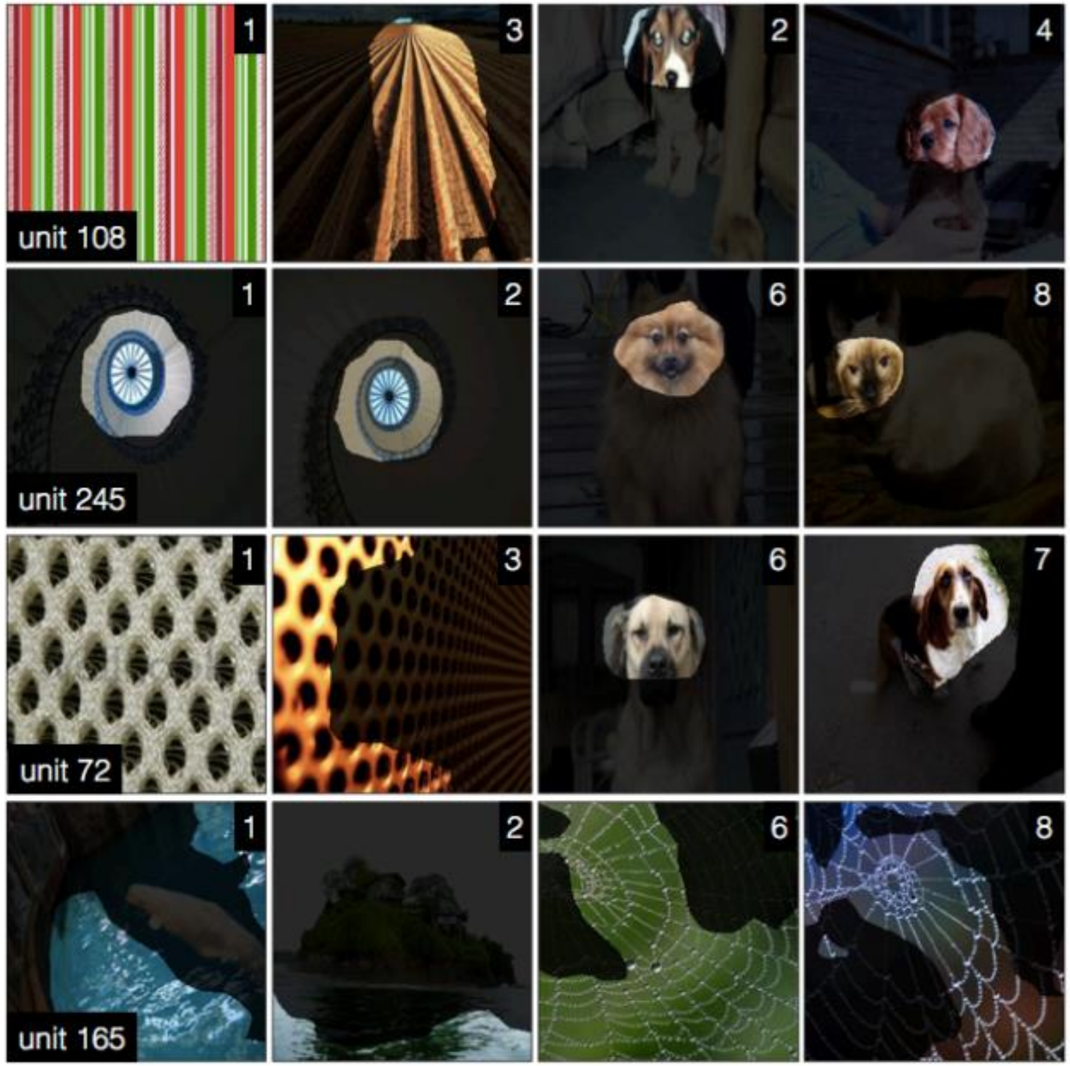
Zeiler & 2015



Mahe 2017

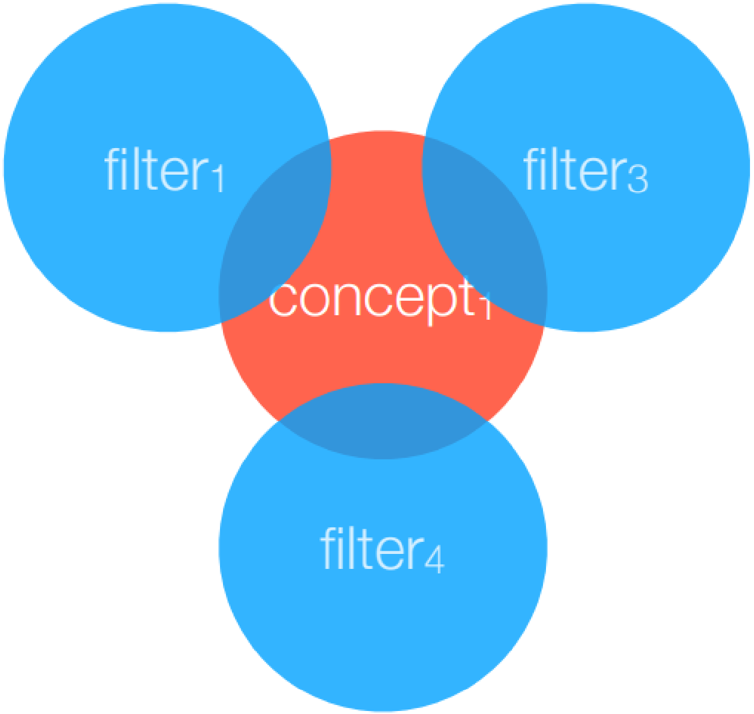


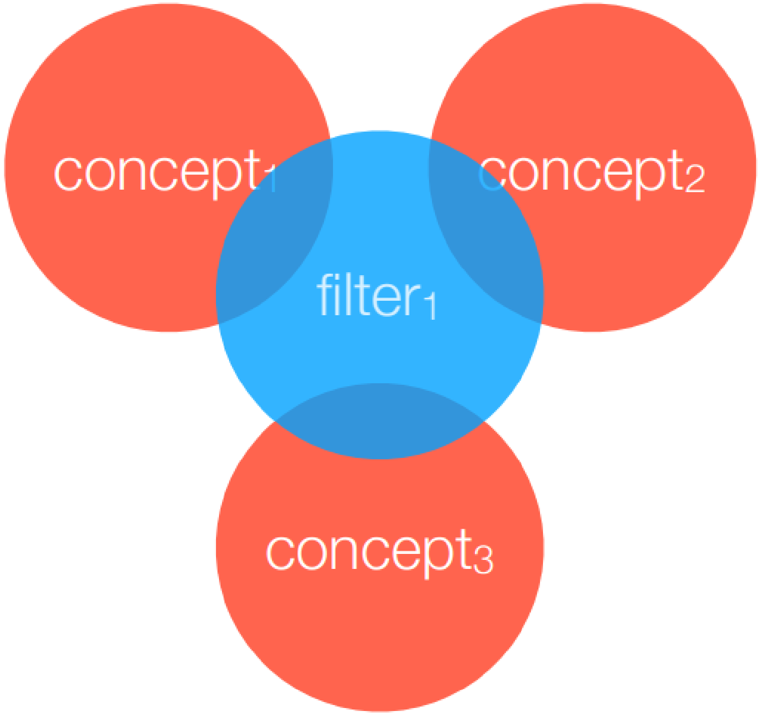




Modified Network Dissection visualization of AlexNet conv5 filters [Bau, et al., 2017]

- Why model-intrinsic?

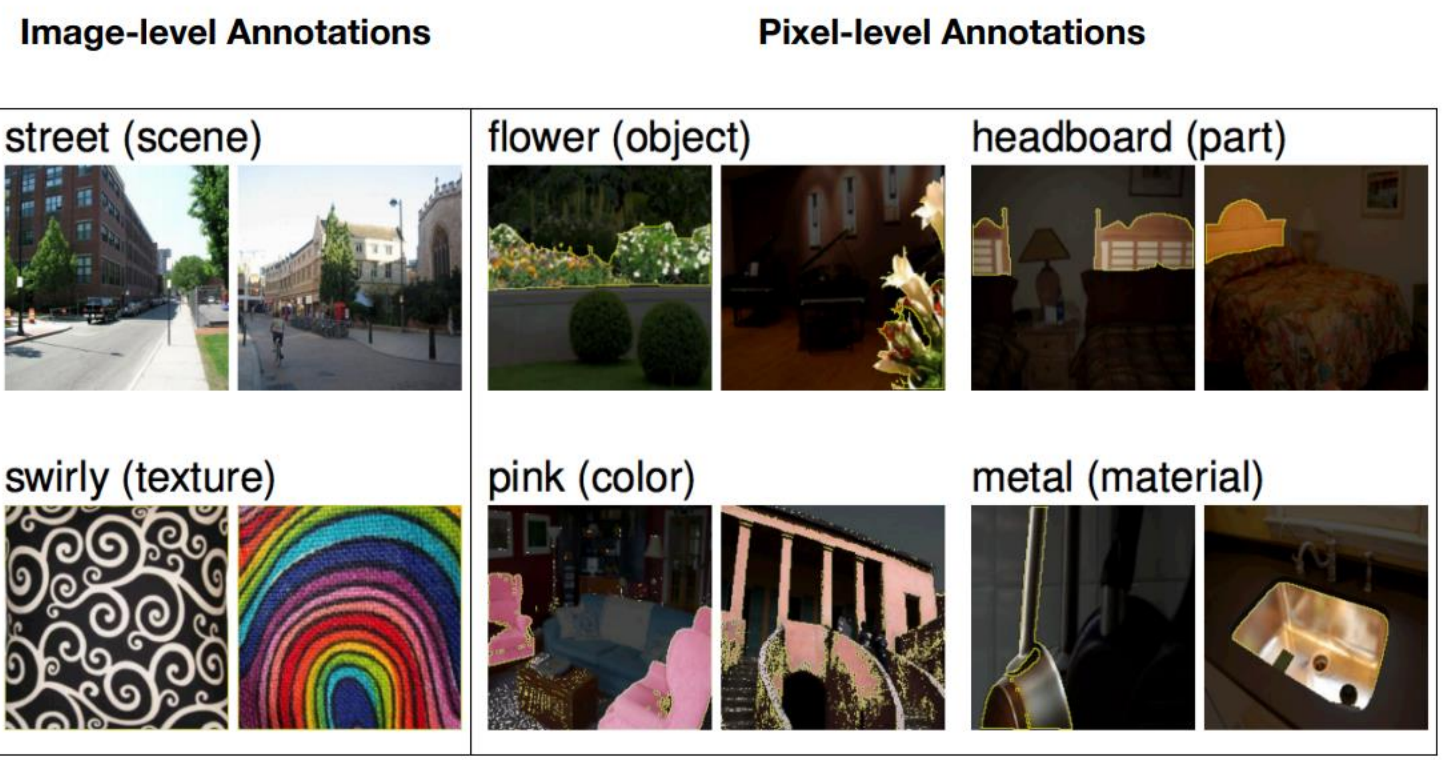




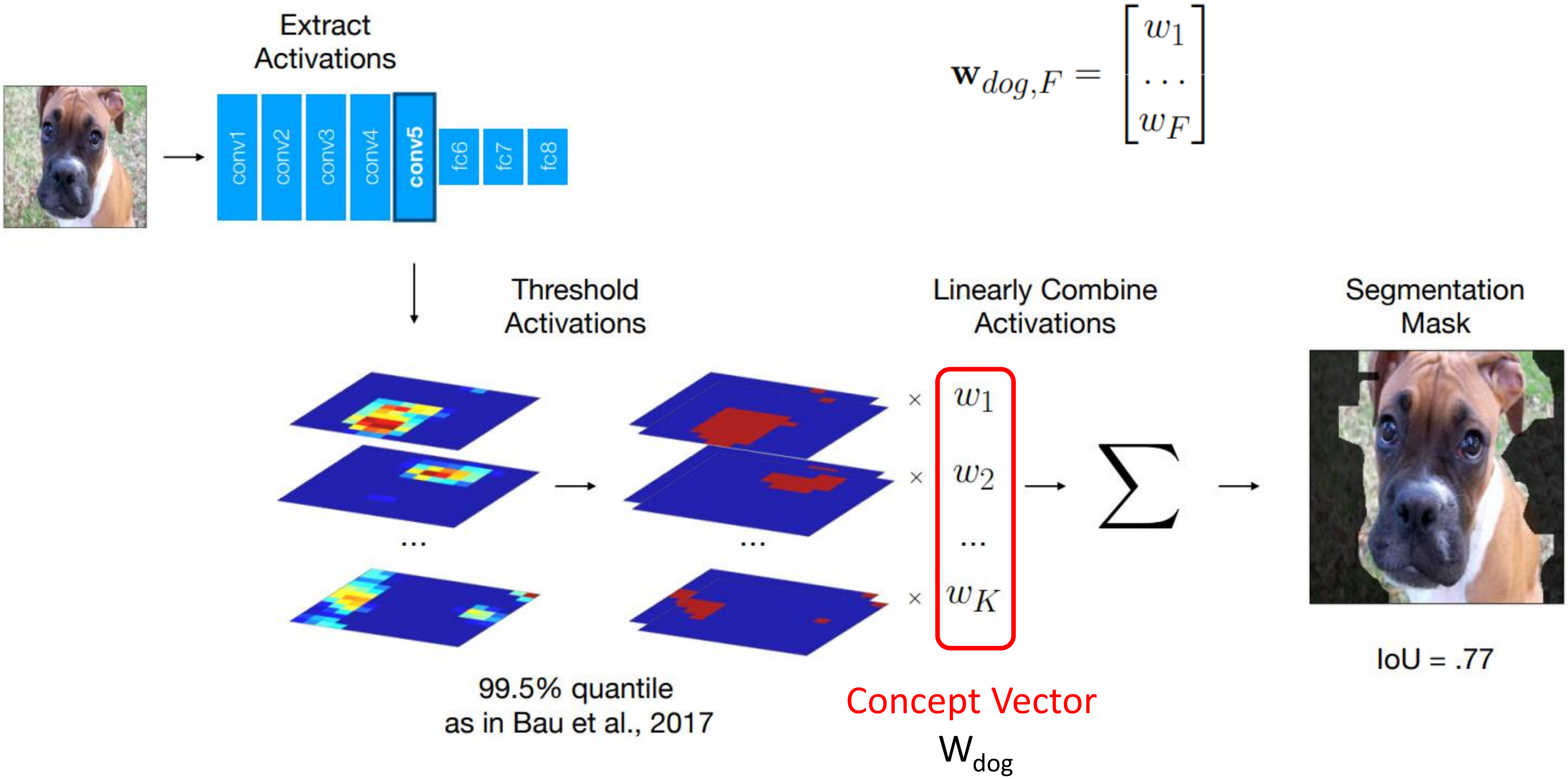
Net2Vec:

Supervised Knowledge Embedding

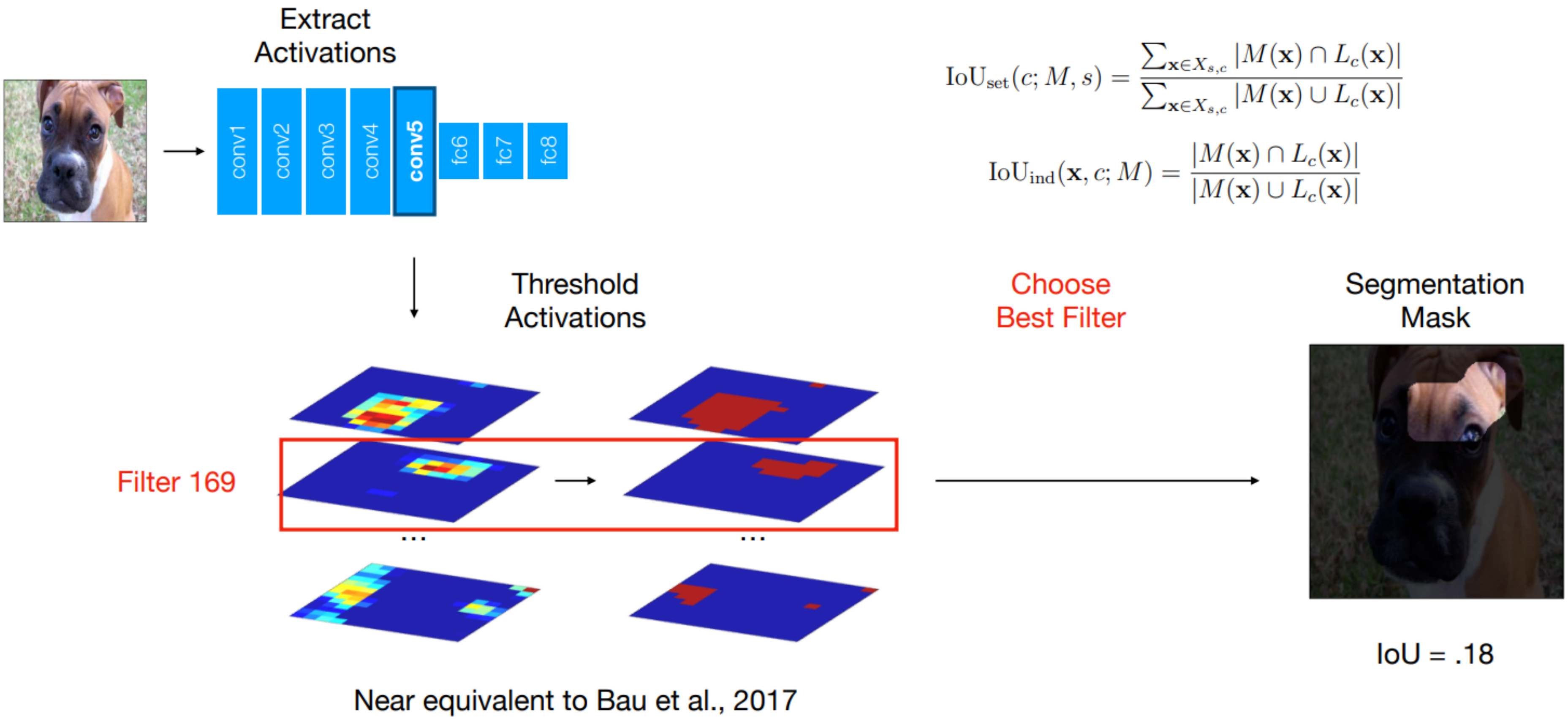
- Probe a network with a dataset
- Concept embedding via filter activation weights



■ Segmentation task



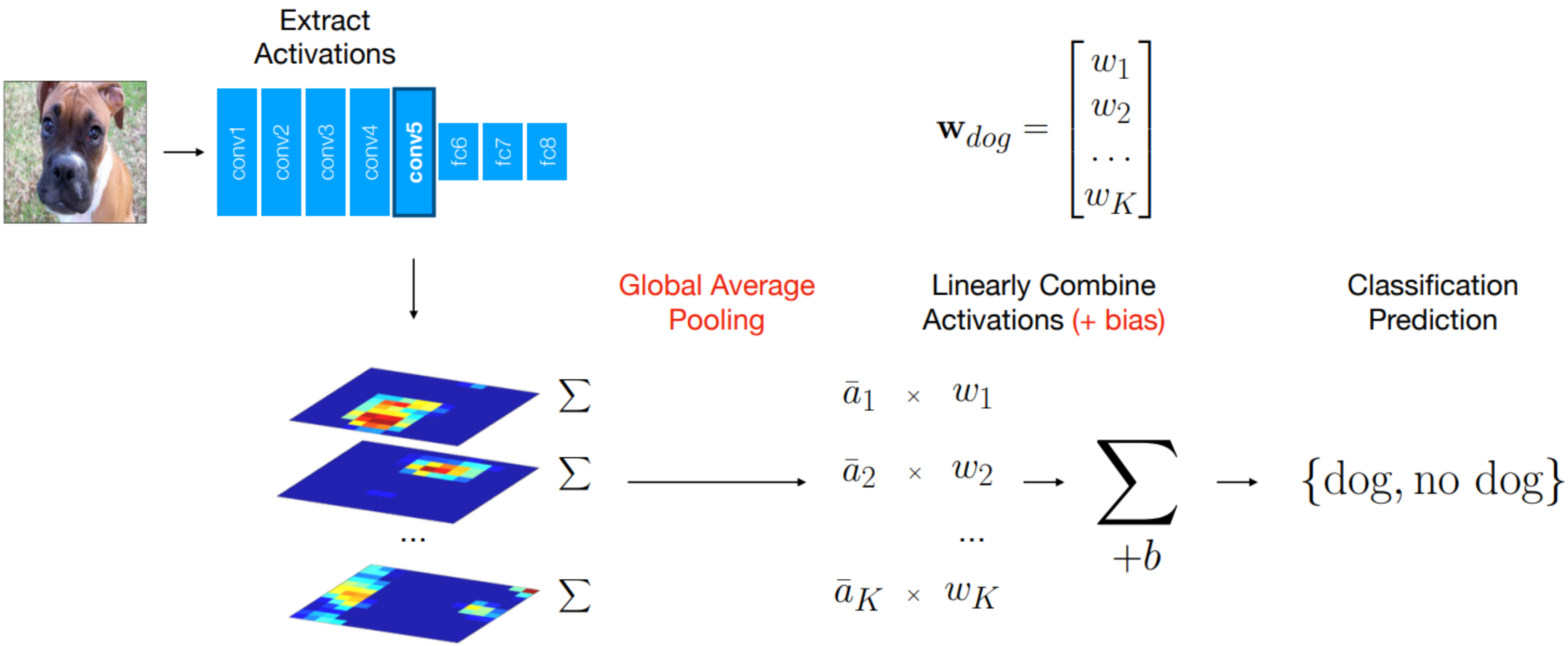
■ Segmentation task



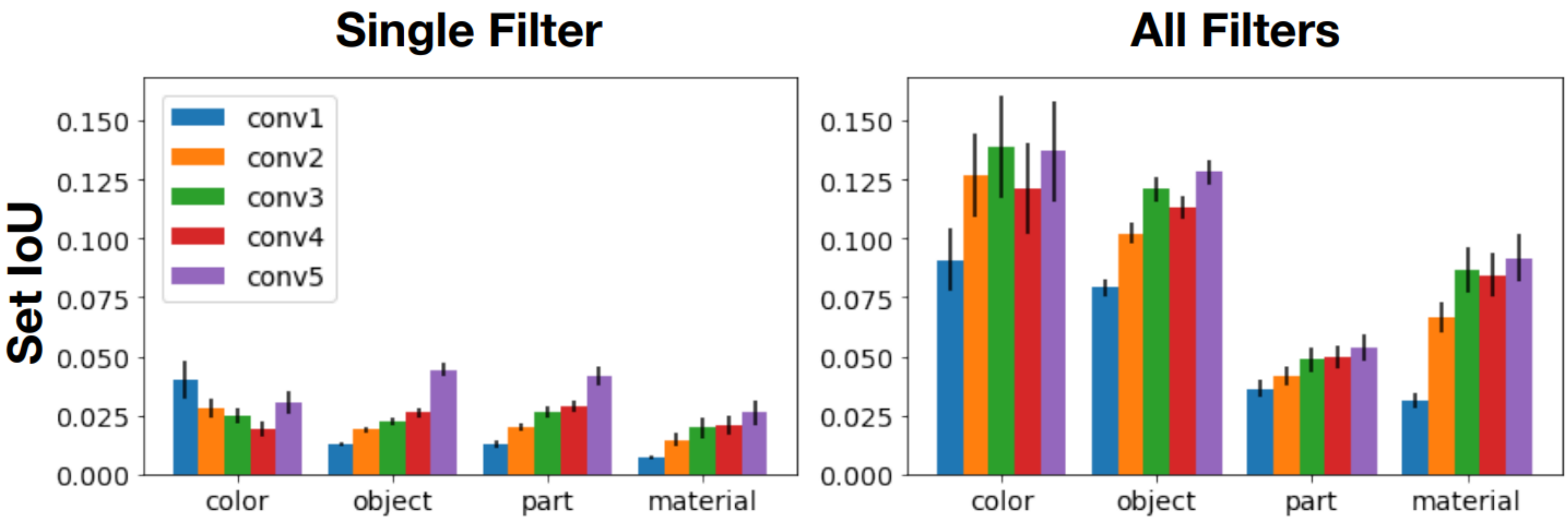
$$\mathrm{IoU}_{\mathrm{set}}(c; M, s) = \frac{\sum_{\mathbf{x} \in X_{s,c}} |M(\mathbf{x}) \cap L_c(\mathbf{x})|}{\sum_{\mathbf{x} \in X_{s,c}} |M(\mathbf{x}) \cup L_c(\mathbf{x})|}$$

$$\mathrm{IoU}_{\mathrm{ind}}(\mathbf{x}, c; M) = \frac{|M(\mathbf{x}) \cap L_c(\mathbf{x})|}{|M(\mathbf{x}) \cup L_c(\mathbf{x})|}$$

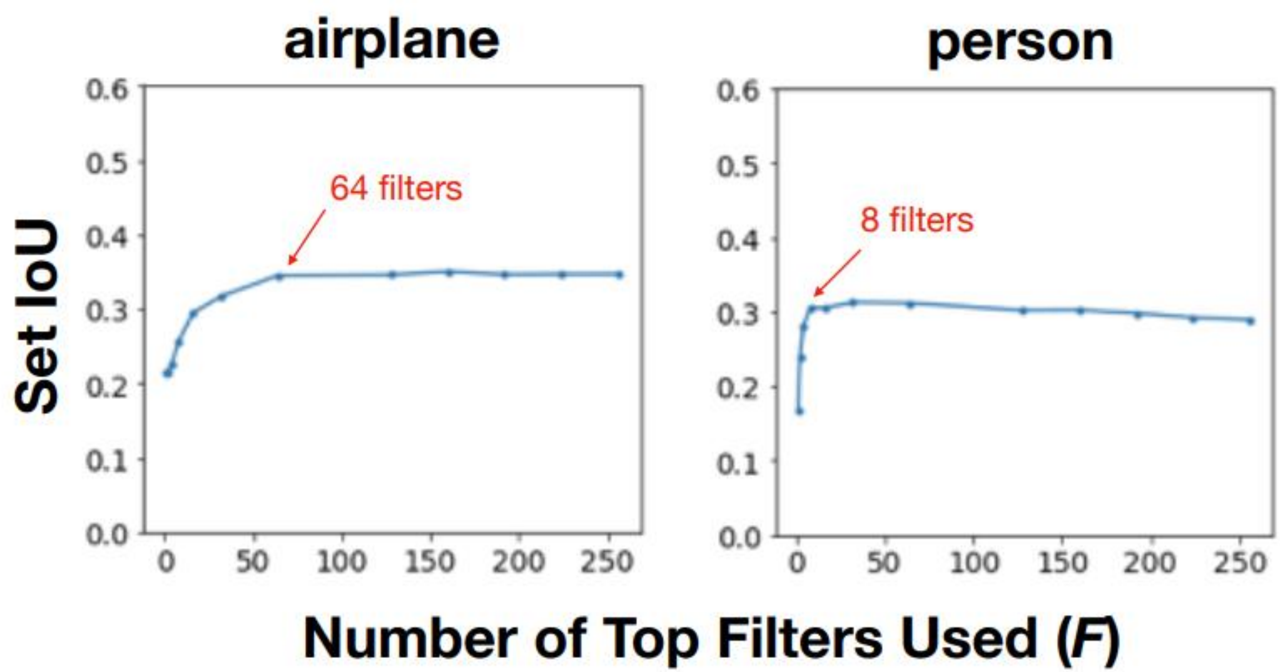
■ Classification task



■ Single vs. Combination



■ Concept Complexity



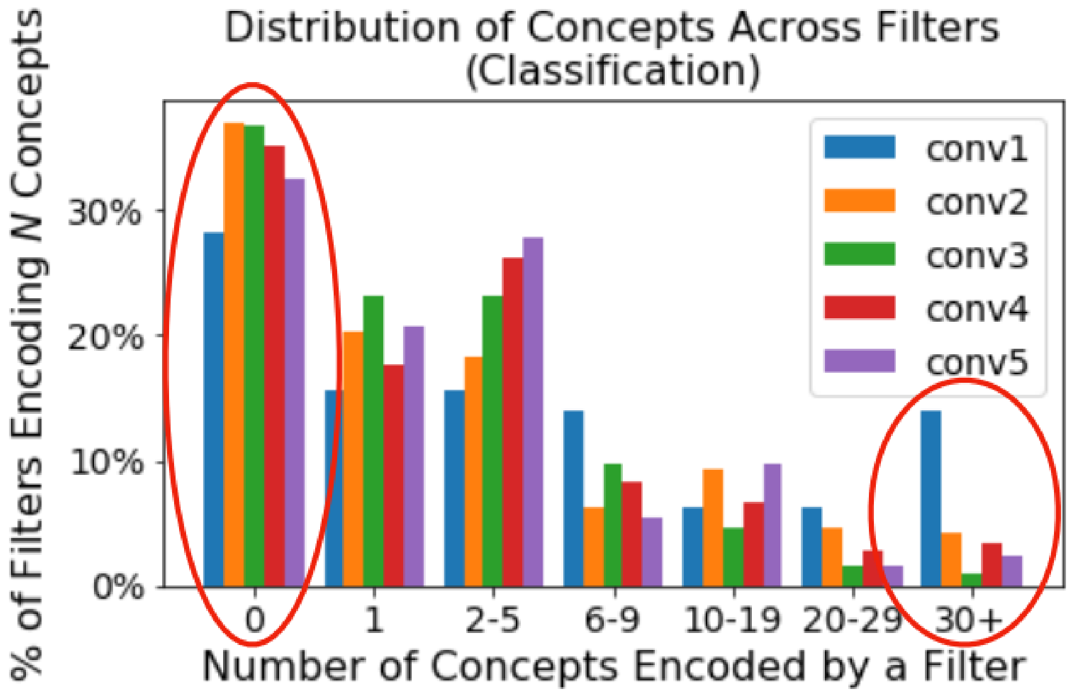
■ Training Setting

Performance Improvement (Single Filter → All Filters):

- Self-supervised networks: 5-6x
- Fully-supervised networks: 2-4x

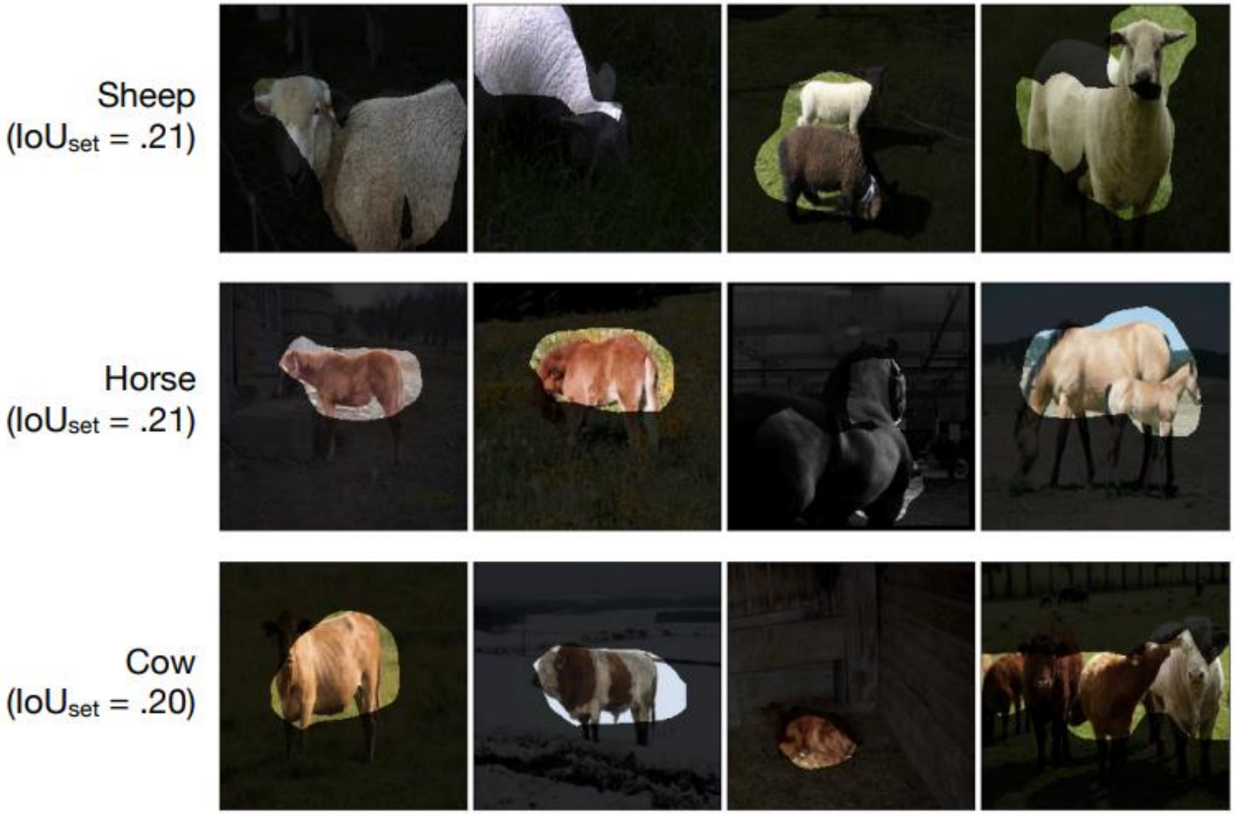
■ Concepts per Filters

- Many filters aren’t selective for any concepts
- A few filters are selective for many concepts



■ Concepts per Filters

- AlexNet conv5 unit 66 is highly selective for various farm animals



■ Concept Vector

Table 2. Nearest concepts (in cos distance) using segmentation (left sub-columns) and classification (right) conv5 embeddings.

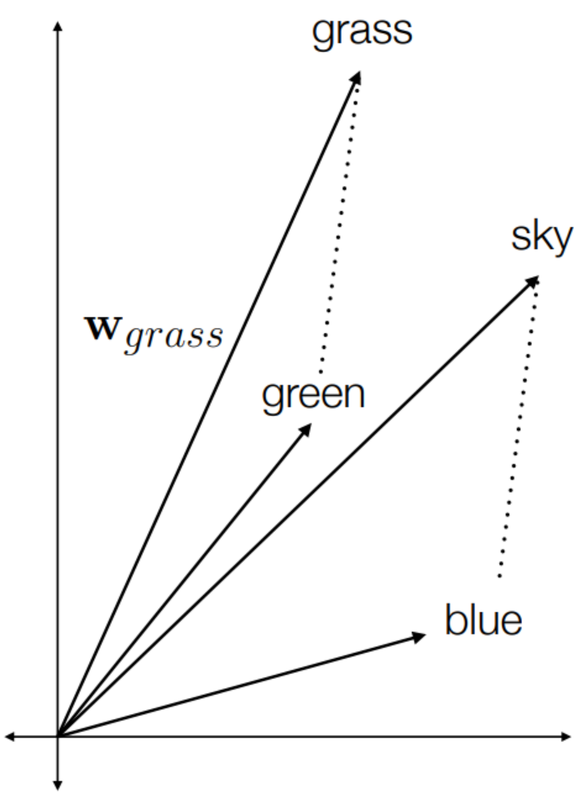
dog		**house**		**wheel**		**street**		**bedroom**	
cat (0.81)	muzzle (0.73)	building (0.77)	path (0.56)	bicycle (0.86)	headlight (0.66)	n/a	sidewalk (0.74)	n/a	headboard (0.90)
horse (0.73)	paw (0.65)	henhouse (0.62)	dacha (0.54)	motorbike (0.66)	car (0.53)	n/a	streetlight (0.73)	n/a	bed (0.85)
muzzle (0.73)	tail (0.52)	balcony (0.56)	hovel (0.54)	carriage (0.54)	bicycle (0.52)	n/a	license plate (0.73)	n/a	pillow (0.84)
ear (0.72)	nose (0.47)	bandstand (0.54)	chimney (0.53)	wheelchair (0.53)	road (0.51)	n/a	traffic light (0.73)	n/a	footboard (0.82)
tail (0.72)	torso (0.44)	watchtower (0.52)	earth (0.52)	water wheel (0.48)	license plate (0.49)	n/a	windshield (0.71)	n/a	shade (0.74)

数据挖掘实验室
Data Mining Lab

■ Concept Vector

Table 3. Vector arithmetic using segmentation, conv5 weights.

grass + blue − green	grass − green	tree − wood	person − torso
sky (0.17)	earth (0.22)	plant (0.36)	foot (0.12)
patio (0.10)	path (0.21)	flower (0.29)	hand (0.10)
greenhouse (0.10)	brown (0.18)	brush (0.29)	grass (0.09)
purple (0.09)	sand (0.16)	bush (0.28)	mountn. pass (0.09)
water (0.09)	patio (0.15)	green (0.25)	backpack (0.09)



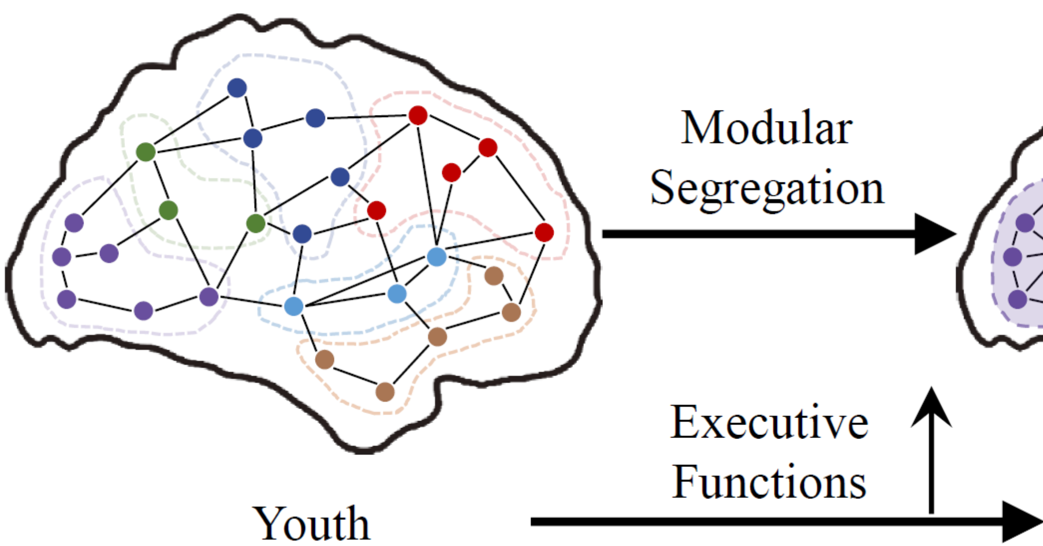
- What’s right?
  - Compared to other interpretable methods, knowledge embedding is a general format!

- What’s wrong?
  - Concept & filter overlap
  - Supervised -> not general

Our proposals

■ Motivation

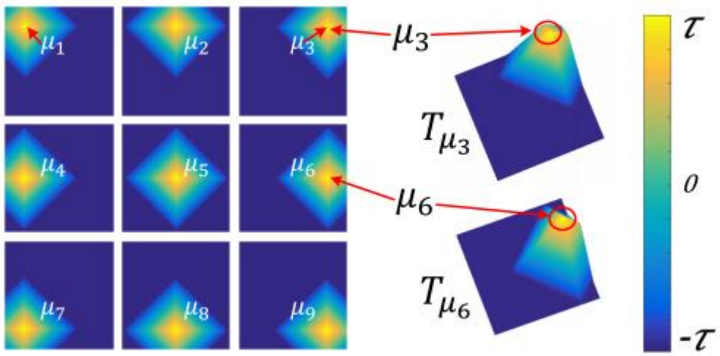
- Interpretation and Efficient via Localization
- A Feature Map as an Entity -> Sync as Modules



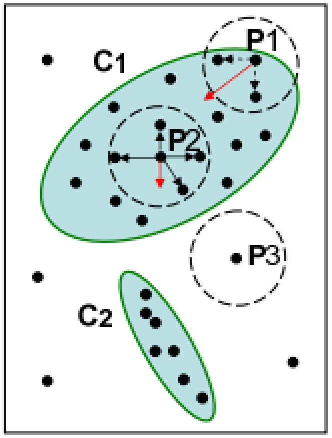
Proposal 1

■ Idea

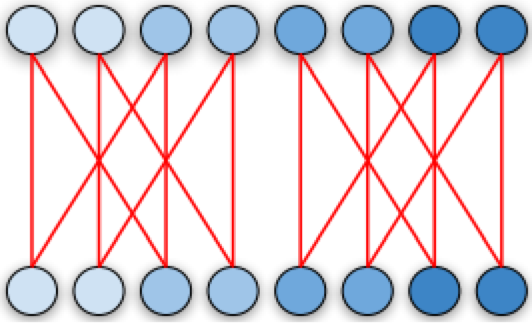
- Minimize Entropy from three levels
- Sync (unsupervised) for Network Segmentation



Feature Map



Functional Module



Network Structure

Feature Map Level

- A feature map -> A specific pattern
- Single-peak Gaussian Activation (only high-level?)
- Previous Work:

$$\begin{aligned}\mathbf{Loss}_f &= -MI(\mathbf{X};\mathbf{T}) \quad \text{for filter } f \\ &= -\sum_T p(T) \sum_x p(x|T) \log \frac{p(x|T)}{p(x)}\end{aligned}$$

- Our work (for computational complexity?)

$$H(u_x|u_\mu) = -\sum_\mu p(u_\mu) \sum_x p(u_x|u_\mu) \log p(u_x|u_\mu)$$

$$H(u_\mu) = -\sum_\mu p(u_\mu) \log p(u_\mu)$$

Functional Module Level

- SynC in clustering

$$\frac{d\theta_i}{dt} = \omega_i + \frac{S}{N}\sum_{j=1}^{N}\sin(\theta_j - \theta_i), (i = 1, \dots, N)$$

$$Nb_\epsilon(x) = \{y \in \mathcal{D} | dist(y, x) \leq \epsilon\}$$

$$x_i(t+1) = x_i(t) + \frac{1}{|Nb_\epsilon(x(t))|} \cdot \sum_{y \in Nb_\epsilon(x(t))} \sin(y_i(t) - x_i(t))$$

- SynC in mini-batch

$$L_{sync} = \frac{1}{N}\sum_{x_i}\frac{1}{|Nb_\epsilon(x_i)|}\sum_{x_j \in Nb_\epsilon(x_i)} dist(x_i, x_j)$$

Methodology

数据挖掘实验室
Data Mining Lab

Network Structure Level

- Pruning in the sync procedure
- Differentiable Continuous met

$$o_{ij} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} f_{(i+m)(j+n)} (U \bigodot \omega_{wn})$$

$$y^{(k)}(w, y^{(k-1)}; \alpha) = m \odot \text{Conv-BN-ReLU}(w, y^{(k-1)})$$

$$\text{s.t.} \quad m_i \sim \text{Bernoulli}(p_i), \quad \sum_{i=1}^{C} p_i = \sum_{i=1}^{C} f(\alpha, b_i) = \alpha C, \quad i = 1, \cdots, C$$

Similarity (Neighborhood)

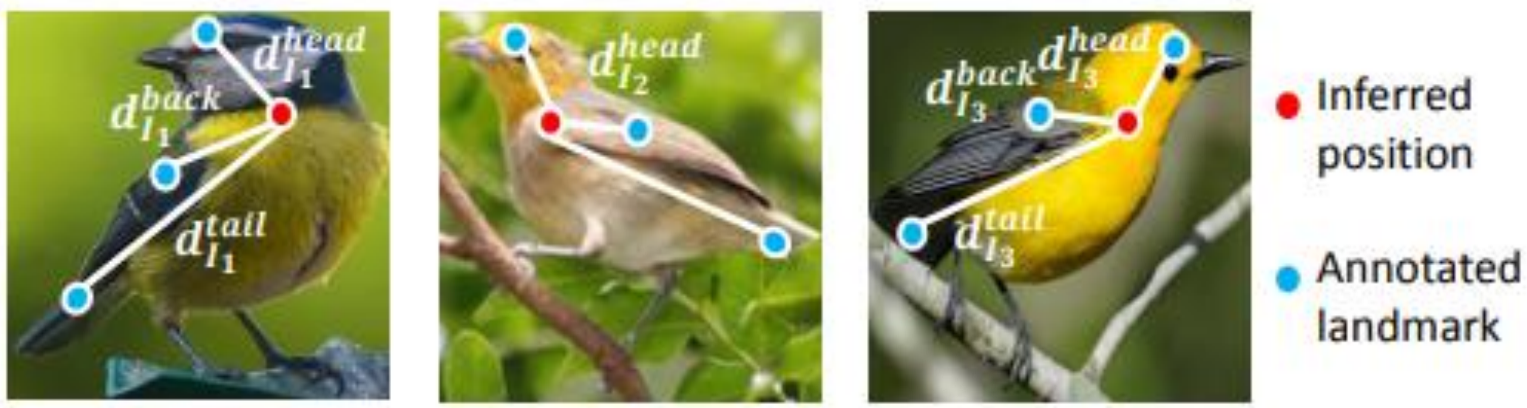
$$f(b_i, \beta_1, \beta_2) = \text{Sigmoid} \circ \text{Log}(b_i) = \frac{1}{1+(\frac{b_i}{\beta_1})^{-\beta_2}}, \quad \beta_1, \beta_2 > 0.$$

Epoch ↑ β_2 ↑ β_1 -> threshold

Challenge

■ Feature Map Level

- Previous Work:



No pytorch implementation can run properly!

hard to adapt on Matlab code

Challenge

数据挖掘实验室
Data Mining Lab

■ Feature Map Level

- Our approach:

$$Var(X) = (X-\mu)^2$$

L_2 regularization!

$$H(u_x|u_\mu) = -\sum_{\mu} p(u_\mu) \sum_{x} p(u_x|u_\mu) \log q(u_x|u_\mu)$$

Out of Memory!

Point-wise: C * hw * hw
Channel-wise: C * hw

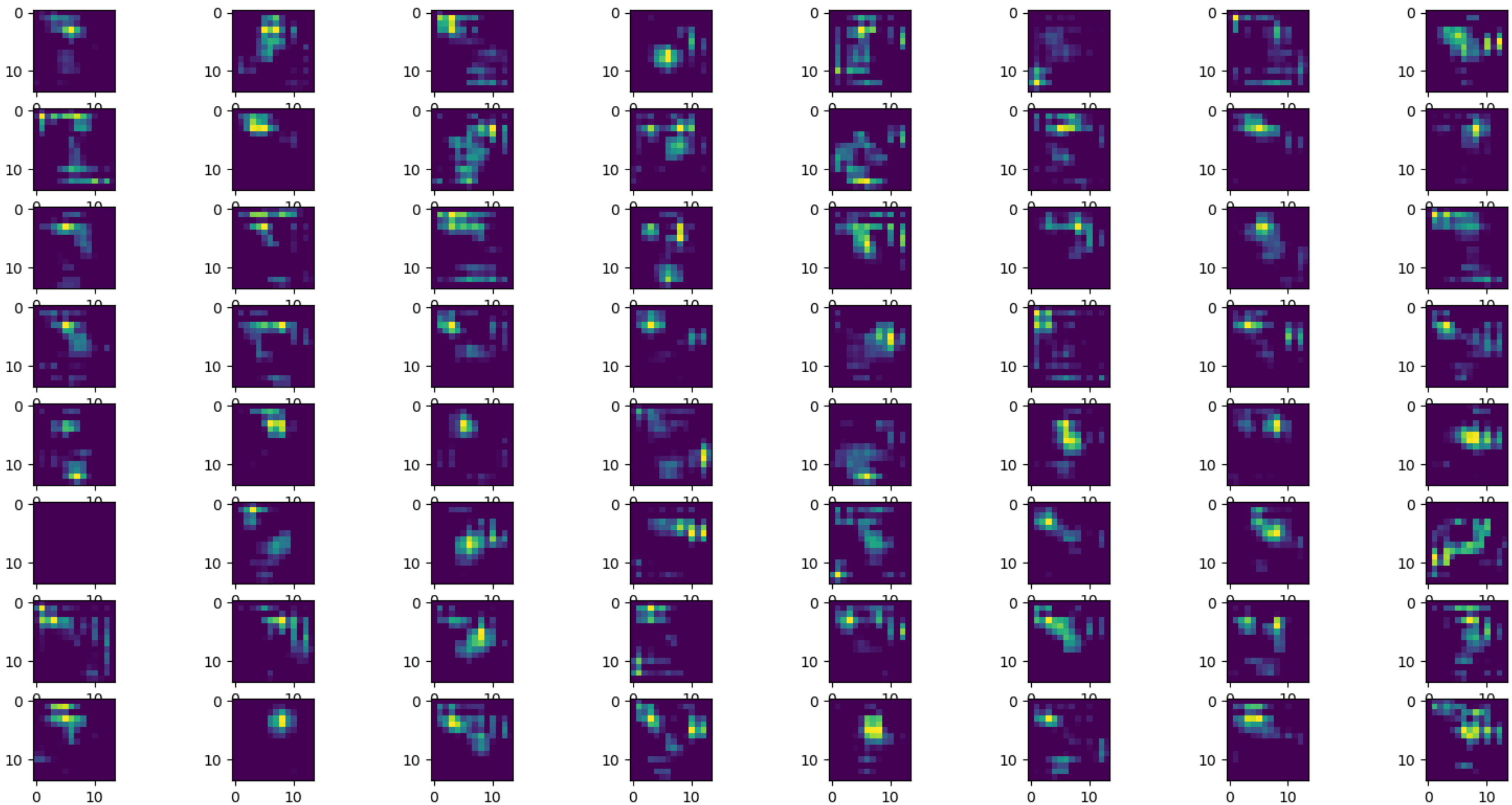
$$H(u_x|u_\mu = \mu) = -\sum_{x} p(u_x|\mu) \log q(u_x|\mu)$$

Poor diversity!

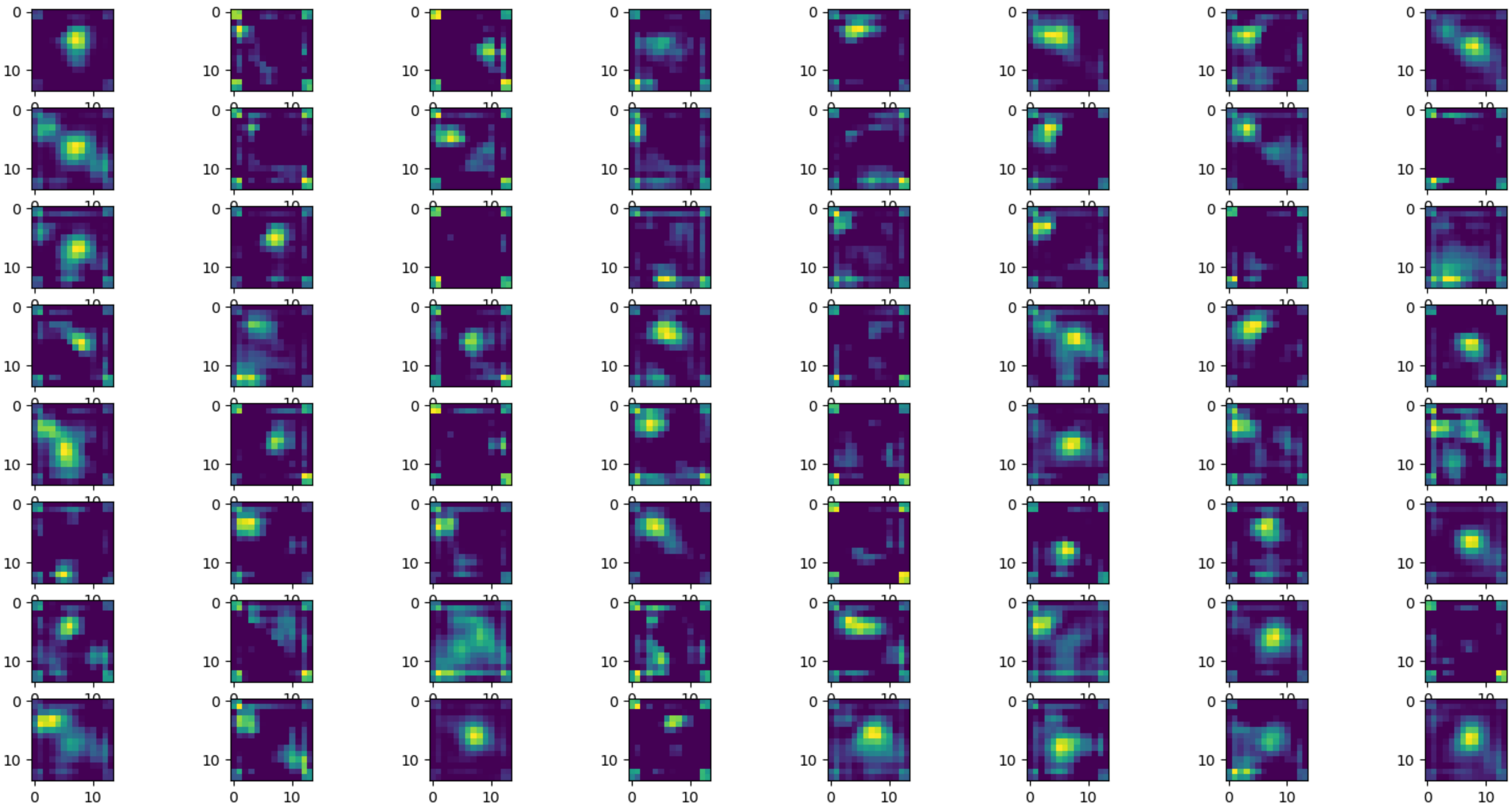
Challenge

■ Feature Map Level

- Raw CNN:



- Feature Map Level
  - Pytorch ICNN (not work well):

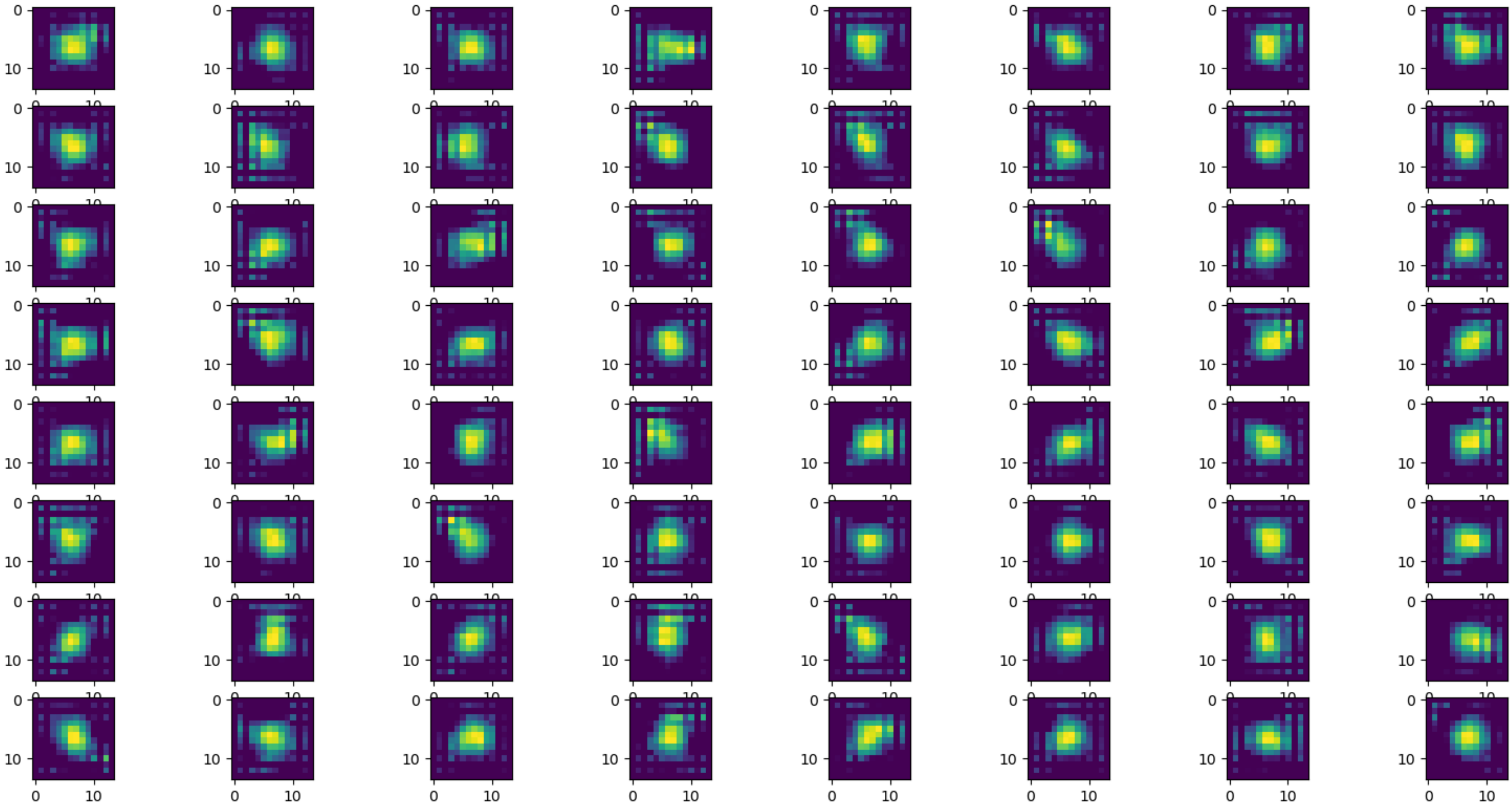


Challenge

数据挖掘实验室
Data Mining Lab

■ Feature Map Level

- Point-wise (Centralization):



Challenge

■ Feature Map Level

- L1 or softmax
- Log_softmax
- Train test

	acc	loss	L1
raw	81.76%	pw 5.0309	0.0946
pw 1e-1	81.79%	4.8441	0.0941
pw 1e0	81.88%	3.989	0.0785
~~pw 1e1~~	~~57.30%~~	~~3.7992~~	~~0.0526~~
w/o H(T) 1e-1	81.39%	0.0889	0.0928
w/o H(T) 1e0	80.43%	0.0408	0.0903
pw 1e-1 w/o p	56.35%	5.142	0.1025
~~pw 1e0 w/o p~~	~~53.64%~~	~~4.0978~~	~~0.062~~

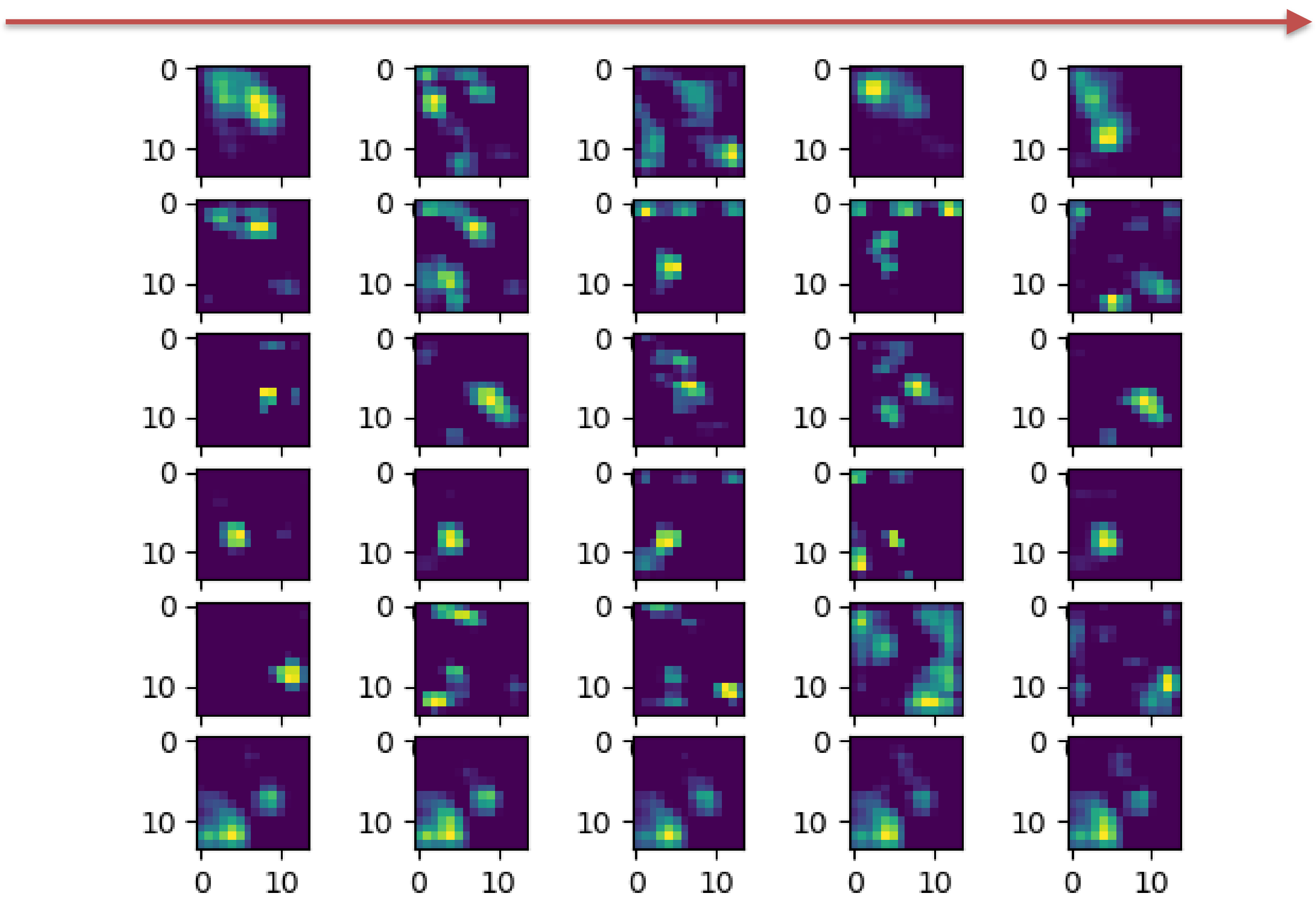
Functional Module Level

- w/o constraint on feature & w/ cos distance:

	acc	sloss	平均有邻率	平均邻居数量	平均最大簇规模	平均邻居距离	Stability (L1)
Raw 0.1	57.68%	3.7922	281.7	8.5	36	0.0635	0.0959
sync 0.1 1e-3	58.42%	3.6692	272.4	9.2	37.6	0.0629	0.1025
sync 0.1 1e-2	60.20%	3.3406	249	8.2	34.2	0.0665	0.096
th 0.1 1e-2	61.48%		296.6	11.7	44.8	0.055	0.099
sync 0.1 1e-1	58.94%	3.2877	251.8	15.2	57.2	0.0741	0.0929
th 0.1 1e-1	58.18%		360.8	64.2	140.2	0.0078	0.0893
Raw 0.2	57.68%	5.4638	409.2	20	74.4	0.1279	0.0959
sync 0.2 1e-3	59.15%	5.5122	413	20.7	76.6	0.1265	0.099
sync 0.2 1e-2	61.84%	5.3647	402.4	24.6	85.5	0.1262	0.1038
th 0.2 1e-2	59.32%		421.4	38.1	117.8	0.1058	0.0989
sync 0.2 1e-1	58.30%	4.9417	382.5	24.3	105.9	0.1575	0.0825
th 0.2 1e-1	59.67%		446.7	193.7	298.2	0.0277	0.0951
Raw 0.3	57.68%	6.1908	467.9	36.3	114.8	0.1932	0.0959
sync 0.3 1e-3	60.13%	6.1995	468.8	39.4	121.1	0.1905	0.1034
sync 0.3 1e-2	59.63%	6.1532	465.9	37.2	117.7	0.1947	0.0987
sync 0.3 1e-1	59.96%	5.8254	450.9	32.3	141.4	0.2239	0.0858
th 0.3 1e-1	60.60%		479.2	196.9	314	0.0537	0.1052

■ Functional Module Level

- Sync:

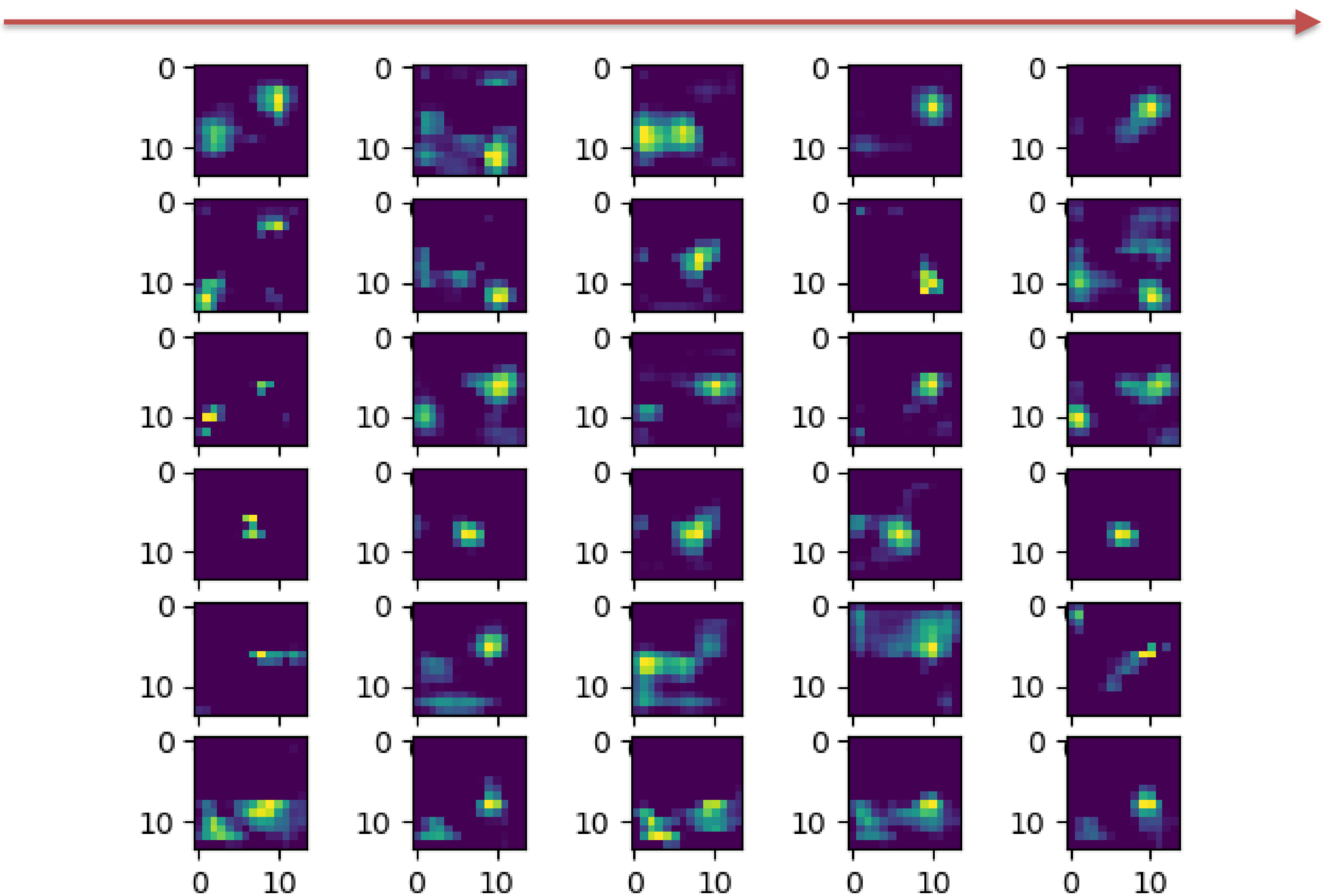


Challenge

数据挖掘实验室
Data Mining Lab

■ Functional Module Level

- Sync:



Plan

- Improve point-wise constraint
- Combine centraloss + syncloss
- Block-wise multi-layer minimum entropy
- Optional: Entropy on network structure

Understanding Sync Mechanism

- Sync in mini-batch

$$L_{sync} = \frac{1}{N}\sum_{x_i}\frac{1}{|Nb_\epsilon(x_i)|}\sum_{x_j \in Nb_\epsilon(x_i)} dist(x_i, x_j)$$

Proposal 2

■ Motivation

- Interpretable Modules with attention
- Unsupervised knowledge embedding

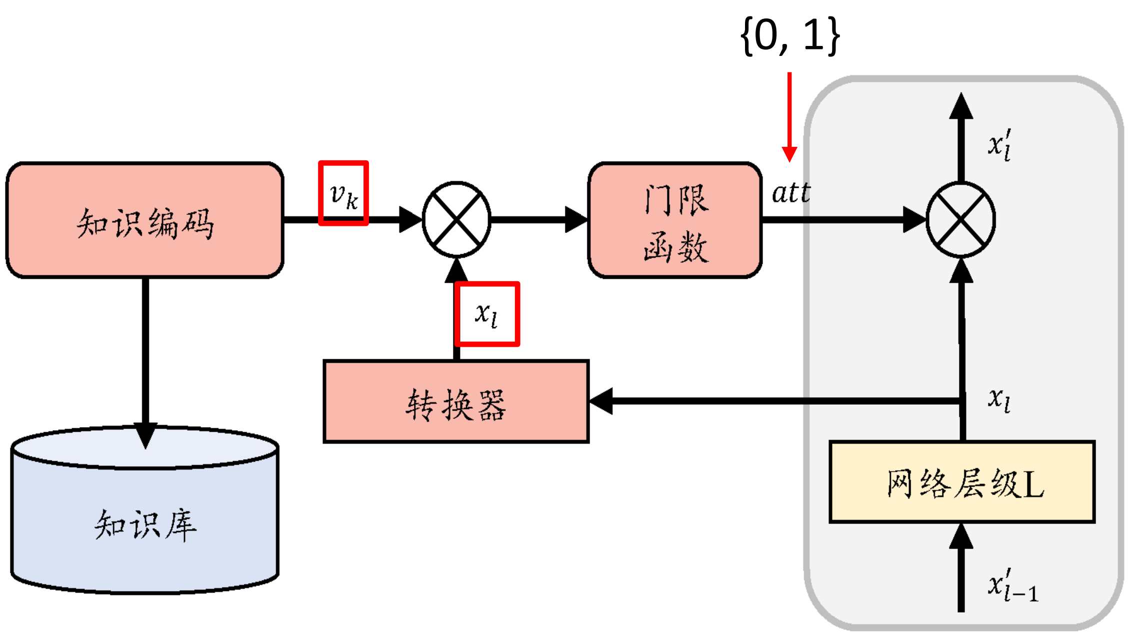
■ Idea

- Attention structure (Key vec as knowledge embedding)
- Sync to regular representation space
- For downstream tasks of GAI

Proposal 2

■ Attention Structure

- Knowledge embedding as Key vector
- Hard attention -> activate the knowledge or not



Proposal 2

- **Knowledge embedding for downstream tasks**
  - Local, Disentangled, Interpretable, Comparable
  - Knowledge management for Continual learning
  - Multi-source, partial, model-based Transfer learning
  - Zero-shot via compositional generalization
  - Match with existing knowledge systems
  - Interact with human beings
  - ……

Proposal 2

■ Knowledge embedding for downstream tasks

- 添加旁路（通过init实现）
- 旁路的激活（前传：基于index/基于映射）
- 旁路的来源（直接init/通过XX判定）
- 旁路的训练（涉不涉及额外的参数）
- 网络参数（包括旁路的）剪枝+扩张

Thank you