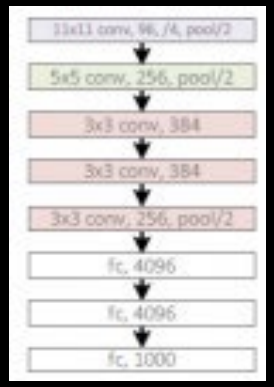
# Unified Understanding of Deep Neural Networks

Quanshi Zhang
Associate Professor
**John Hopcroft Center**
Shanghai Jiao Tong University

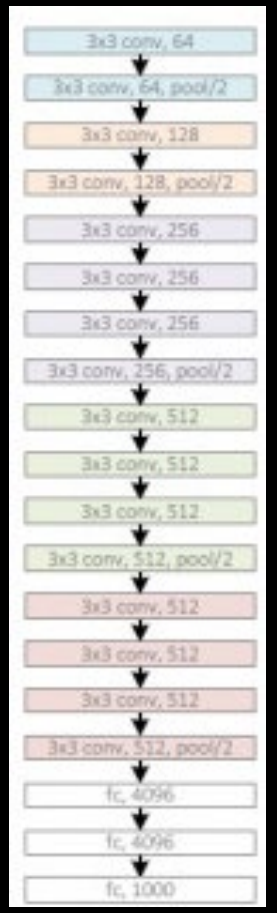
Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, Xing Xie, “Towards A Deep and Unified Understanding of Deep Neural Models in NLP” in ICML 2019
Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu, “Interpreting CNNs via Decision Trees” in CVPR, 2019
Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu, "Interpretable Convolutional Neural Networks" in CVPR (Spotlight) 2018
Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu, "Interpreting CNN Knowledge via an Explanatory Graph" in AAAI, 2018
Quanshi Zhang, Wenguan Wang, and Song-Chun Zhu, “Examining CNN Representations with respect to Dataset Bias” in AAAI, 2018
Quanshi Zhang, Yu Yang, Yuchen Liu, Ying Nian Wu, and Song-Chun Zhu, "Unsupervised Learning of Neural Networks to Explain Neural Networks" extended abstract in AAAI-19 Workshop on Network Interpretability for Deep Learning, 2019
Quanshi Zhang, Yu Yang, Qian Yu, Ying Nian Wu, and Song-Chun Zhu, “Network Transplanting” extended abstract in AAAI-19 Workshop on Network Interpretability for Deep Learning, 2019

# Neural networks

simple → complex → complexer




AlexNet



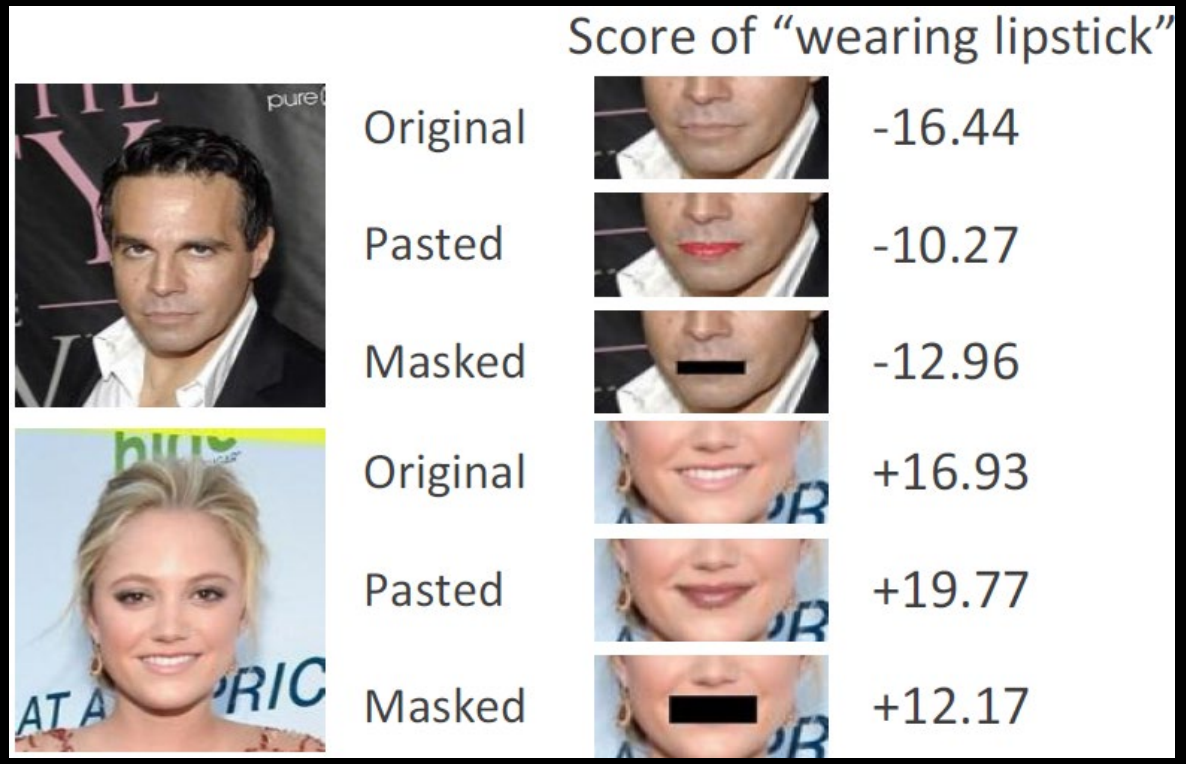

VGG-19

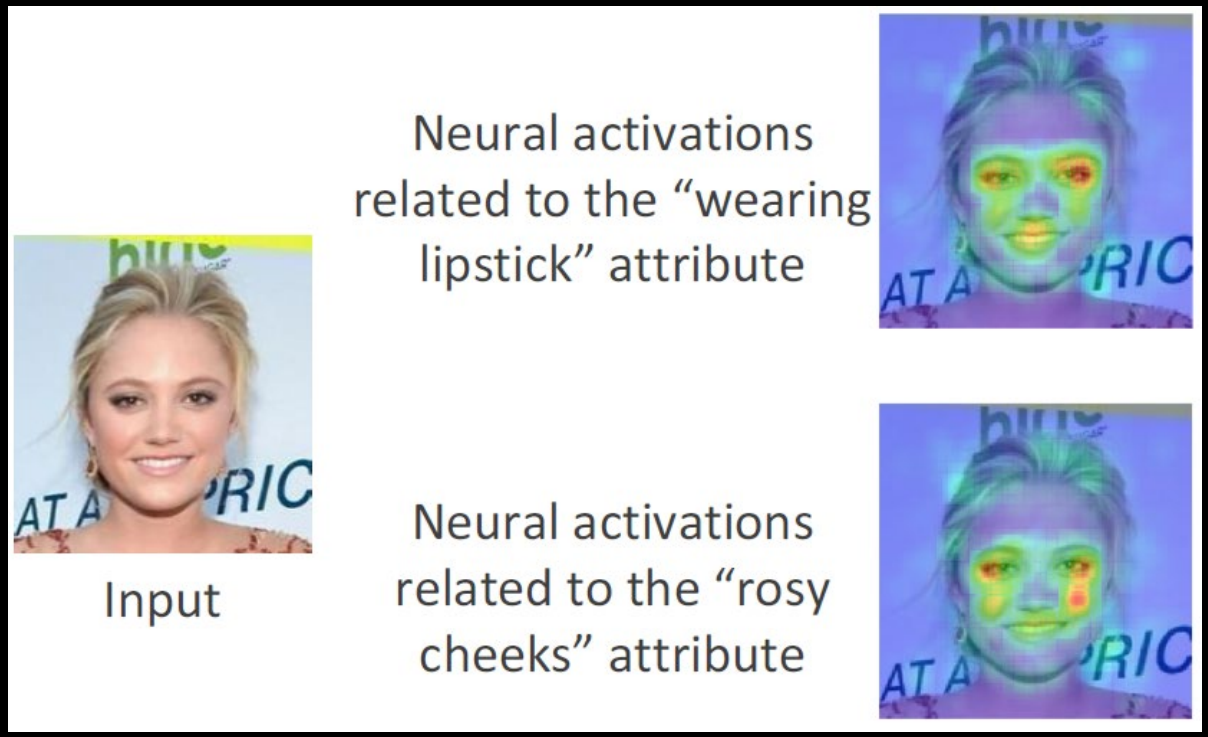
GoogleNet

ResNet

# Really know it? Using incorrect reasons to make correct predictions

Because the “wearing lipstick” attribute always co-appears with “rosy cheeks,” the CNN uses cheek features to represent the lipstick.




**Q. Zhang** and S.-C. Zhu, “Diagnosing CNN representations without Testing Samples” in AAAI 2018

# Explanations → Trustiness, diagnosis

- How to make human beings trust a computer?

**Computer:** We must make a surgery on your head?
**Human:** Why should I trust you and let you cut my head
**Computer:** It is because
1) Filter 1 detected a lesion in Organ A
2) Filter 2 detected a lesion in Organ B
…

# Quantitative explanation

An accident happed.

Human: tell me the reason for road planning before the traffic accident.

**Computer:** It is because
1) Filter 1 detected a tree
2) Filter 2 detected a person
3) Filter 3 detected the road
4) Filter 4 detected **another road**
…

Human: I find Filter 4 considers a river as a road.

Fix representation flaws in the CNN

# Deep learning, a science or a technology?

Deep neural network → a piecewise linear model → unexplainable
→ We will never get accurate explanation for 100% information of a DNN

Alchemy?

- Explain semantic knowledge hidden in intermediate layers
- Mathematically model and explain the representation capacity of DNNs

# Deep learning, a science or a technology?

1. Explain semantic knowledge hidden in intermediate layers

Alchemy?

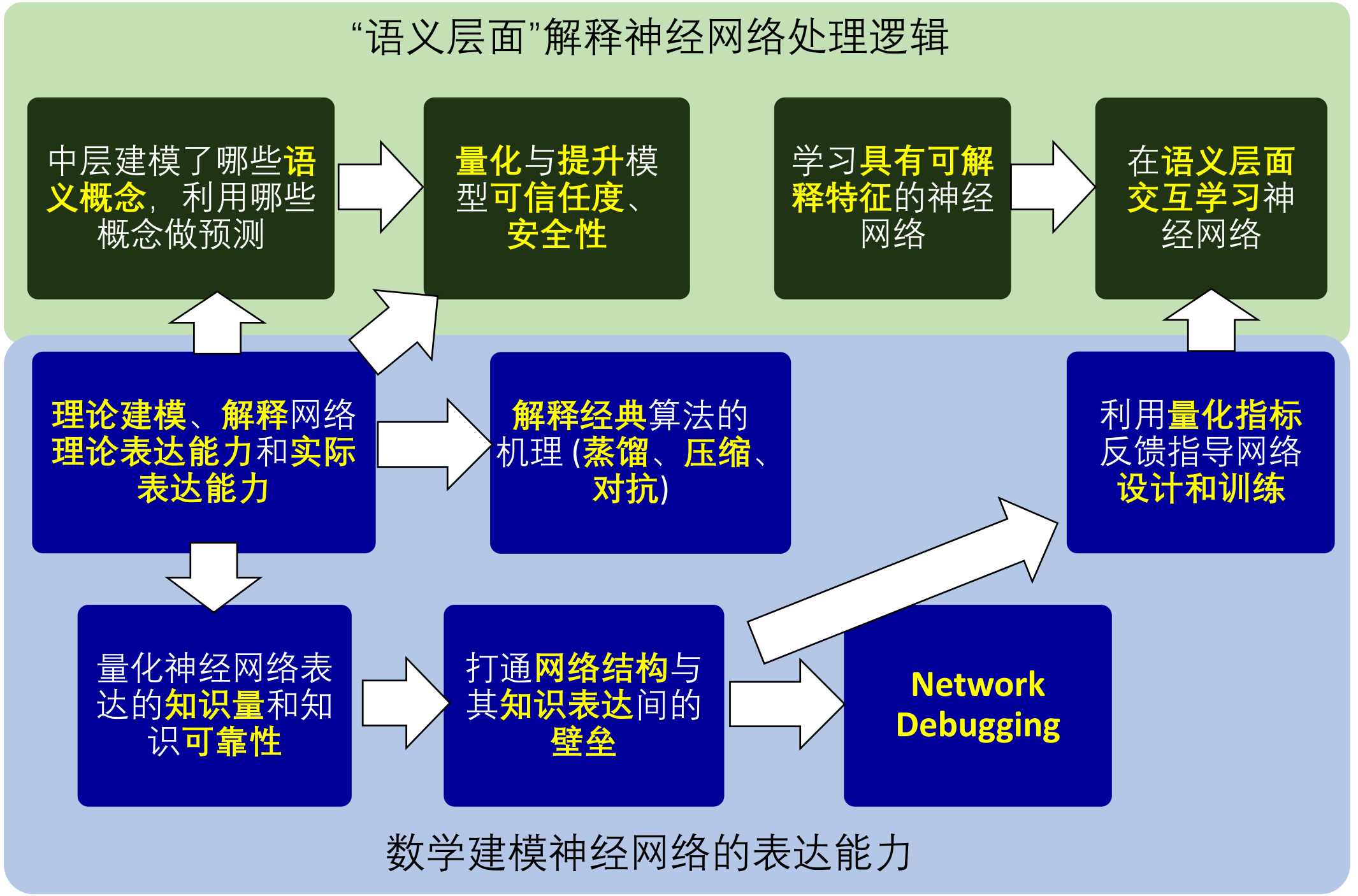
- Explain features in intermediate layers
- Semantically
- Quantitatively
  - **What visual concepts are learned**
  - **Given an image, which visual concepts are used for inference.**
  - **E.g. 90% information is interpretable**
    - **83% represents object parts**
    - **7% represents textures**
    - **10% cannot be interpreted**

# Deep learning, a science or a technology?

2. Explain the representation capacity of deep neural networks

Alchemy?

- Mathematically model and explain the representation capacity
- **Generically**
  - **Without tricky assumptions**
- **Coherently**
  - **Enable comparisons through different layers**
  - **Enable comparisons through different models**
  - **Explaining existing deep-learning methods**


“语义层面”解释神经网络处理逻辑
中层建模了哪些语义概念，利用哪些概念做预测
量化与提升模型可信任度、安全性
学习具有可解释特征的神经网络
在语义层面交互学习神经网络
理论建模、解释网络理论表达能力和实际表达能力
解释经典算法的机理 (蒸馏、压缩、对抗)
利用量化指标反馈指导网络设计和训练
量化神经网络表达的知识量和知识可靠性
打通网络结构与其知识表达间的壁垒
Network Debugging
数学建模神经网络的表达能力

# Outline

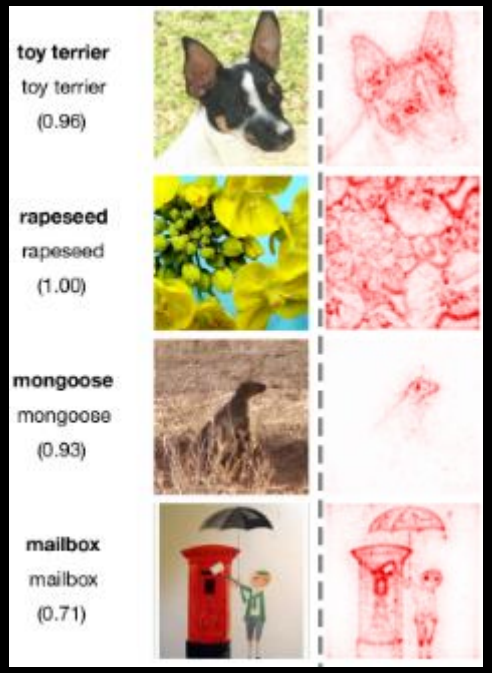
- Explain semantic knowledge hidden in intermediate layers
  - How to represent CNNs using semantic graphical models
  - How to learn disentangled, interpretable features in middle layers
  - 在语义层面定量解释神经网络预测结果
- Explain representation capacity of deep neural networks
  - 对神经网络中层信息处理的量化分析与评测
  - 对神经网络特征表达可靠性的评测

# Outline

- Explain semantic knowledge hidden in intermediate layers
  - **How to represent CNNs using semantic graphical models**
  - How to learn disentangled, interpretable features in middle layers
  - 在语义层面定量解释神经网络预测结果
- Explain representation capacity of deep neural networks
  - 对神经网络中层信息处理的量化分析与评测
  - 对神经网络特征表达可靠性的评测

# Network visualization & diagnosis

Visualization of appearance encoded by a filter




Pixels related to the final prediction output

Can only visualize salient information

The key problem is to explain most information (e.g. 70%--90%) in a network

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. Distill, 2017. https://distill.pub/2017/feature-visualization.

Pieter-Jan Kindermans, Kristof T. Sch¨utt, Maximilian Alber, Klaus-Robert M¨uller, Dumitru Erhan, Been Kim, and Sven D¨ahne. Learning how to explain neural networks: Patternnet and patternattribution. In arXiv: 1705.05598, 2017.

# Background: Learning explanatory graphs for CNNs

- Given a CNN that is pre-trained for object classification
  - **How many types of visual patterns are memorized by a convolutional filter of the CNN?**

Distribution of activations in a feature map

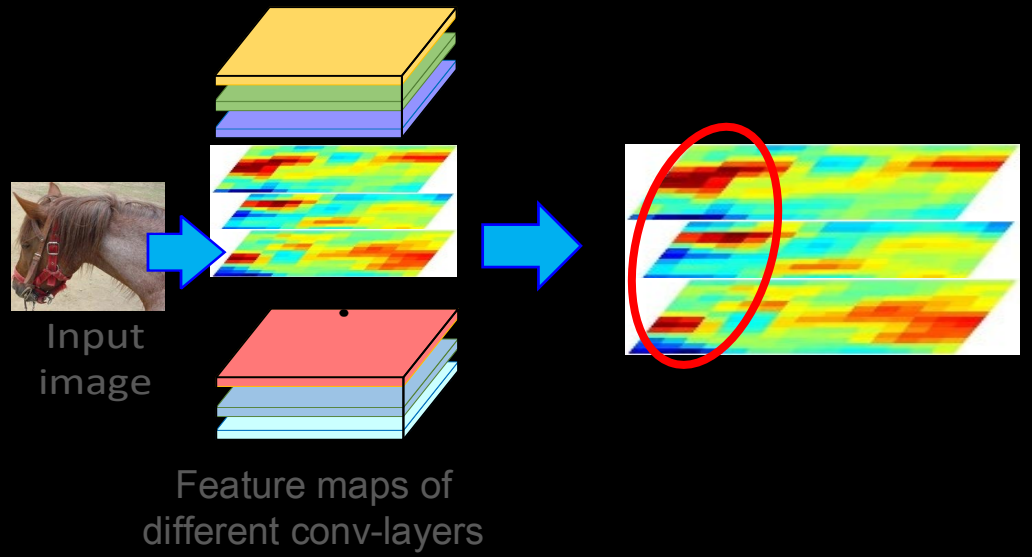
A head pattern

??? pattern


Quanshi Zhang et al. “Interpreting CNN Knowledge via an Explanatory Graph” in AAAI 2018

# Background: Learning explanatory graphs for CNNs

- Given a CNN that is pre-trained for object classification
  - How many types of visual patterns are memorized by a convolutional filter of the CNN?
  - **Which patterns are co-activated to describe a part?**




These filters are co-activated in certain area to represent the head of a horse.

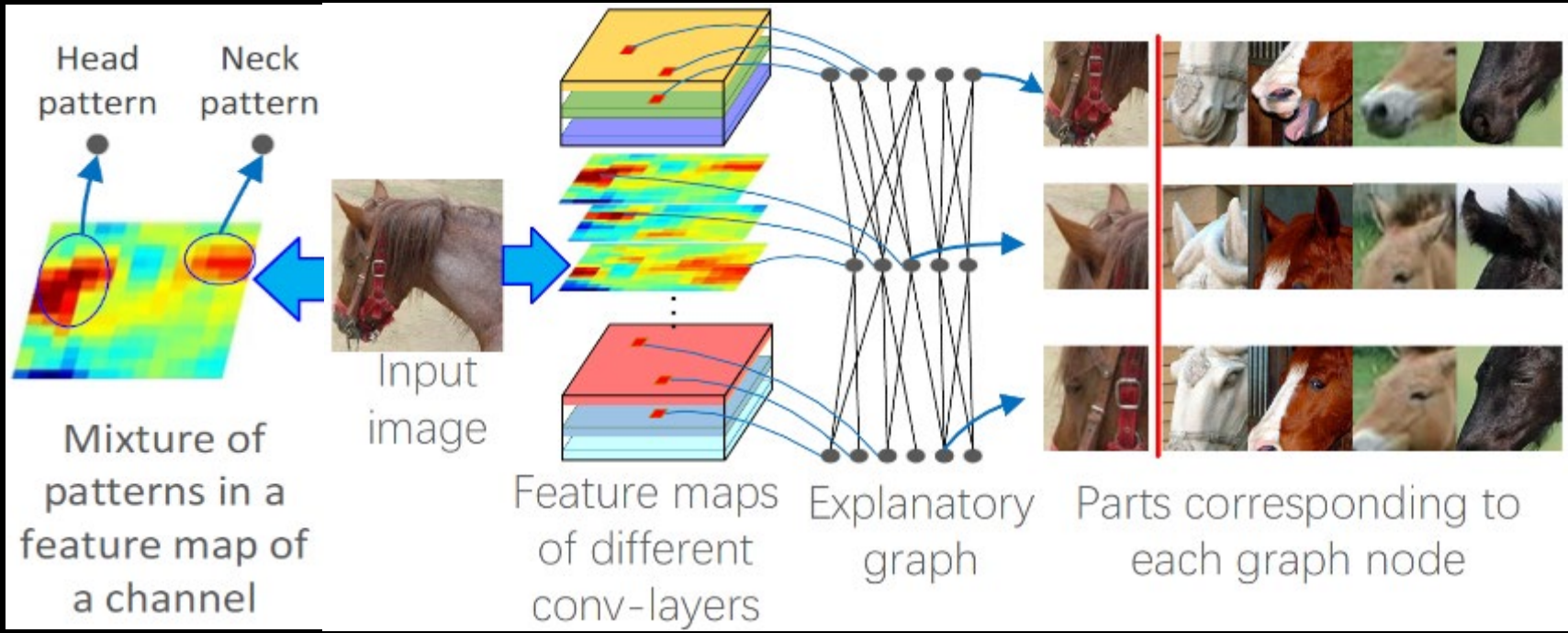
Quanshi Zhang et al. “Interpreting CNN Knowledge via an Explanatory Graph” in AAAI 2018

# Background: Learning explanatory graphs for CNNs

- Given a CNN that is pre-trained for object classification
  - How many types of visual patterns are memorized by a convolutional filter of the CNN?
  - Which patterns are co-activated to describe a part?
  - **What is the spatial relationship between two patterns?**

Quanshi Zhang et al. “Interpreting CNN Knowledge via an Explanatory Graph” in AAAI 2018

# Objective: Summarize knowledge in a CNN into a semantic graph



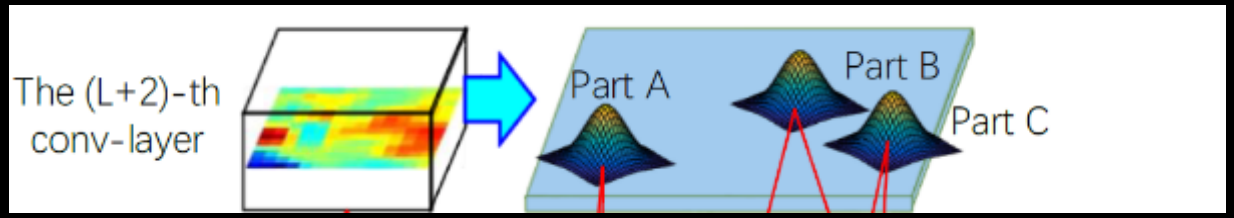

- The graph has multiple layers → multiple conv-layers of the CNN
- Each node → a pattern of an object part
- A filter may encode multiple patterns (nodes)→ disentangle a mixture of patterns from the feature map of a filter
- Each edge → co-activation relationships and spatial relationships between two patterns

# Input & Output

- Input:
  - A pre-trained CNN
    - trained for classification, segmentation, or ...
    - AlexNet, VGG-16, ResNet-50, ResNet-152, and etc.
  - **Without any annotations of parts or textures**
- Output: an explanatory graph

Quanshi Zhang et al. “Interpreting CNN Knowledge via an Explanatory Graph” in AAAI 2018

# Mining an explanatory graph




Just like GMM, we use a mixture of patterns to fit activation distributions of a feature map.

a feature map of a filter
→ a distribution of "activation entities"

$$\text{argmax}_{\boldsymbol{\theta}_L} \prod_{I \in \mathbf{I}} P(\mathbf{X}_L^I | \mathbf{R}_{L+1}^I, \boldsymbol{\theta}_L)$$

$X_L$: the feature map of the L-th conv-layer
$R_{L+1}$: the position inference results for patterns in the (L+1)-th conv-layer

$$P(\mathbf{X}_L|\mathbf{R}_{L+1}, \boldsymbol{\theta}_L) = \prod_{x \in \mathbf{x}_L} P(\mathbf{p}_x|\mathbf{R}_{L+1}, \boldsymbol{\theta}_L)^{F(x)}$$

$$= \prod_{x \in \mathbf{x}_L} \left\{ \sum_{V \in \Omega_{L,d} \cup \{V_{\text{none}}\}} P(V) P(\mathbf{p}_x|V, \mathbf{R}_{L+1}, \boldsymbol{\theta}_L) \right\}_{d=d_x}^{F(x)}$$

For each unit x, F(x) represents the strength of the activation (i.e. the number of activation entities)

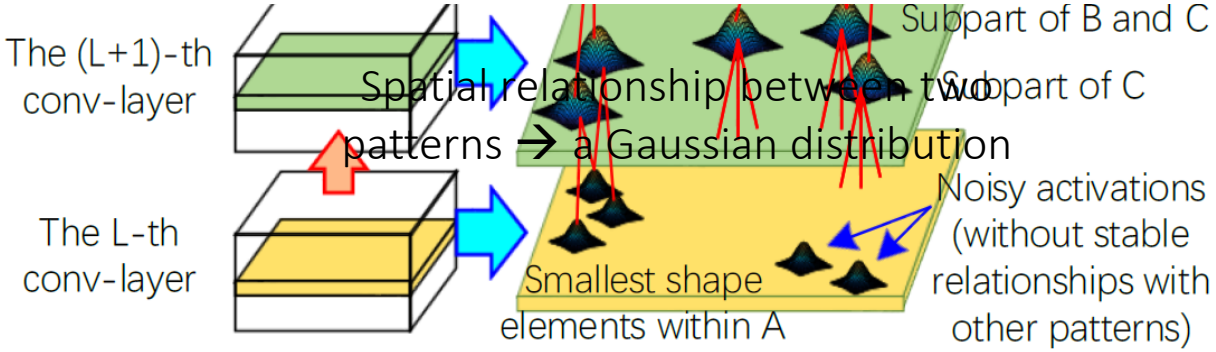
We use a set of pattern candidates to describe the "activation entities" in the position of x.

Quanshi Zhang et al. "Interpreting CNN Knowledge via an Explanatory Graph" in AAAI 2018

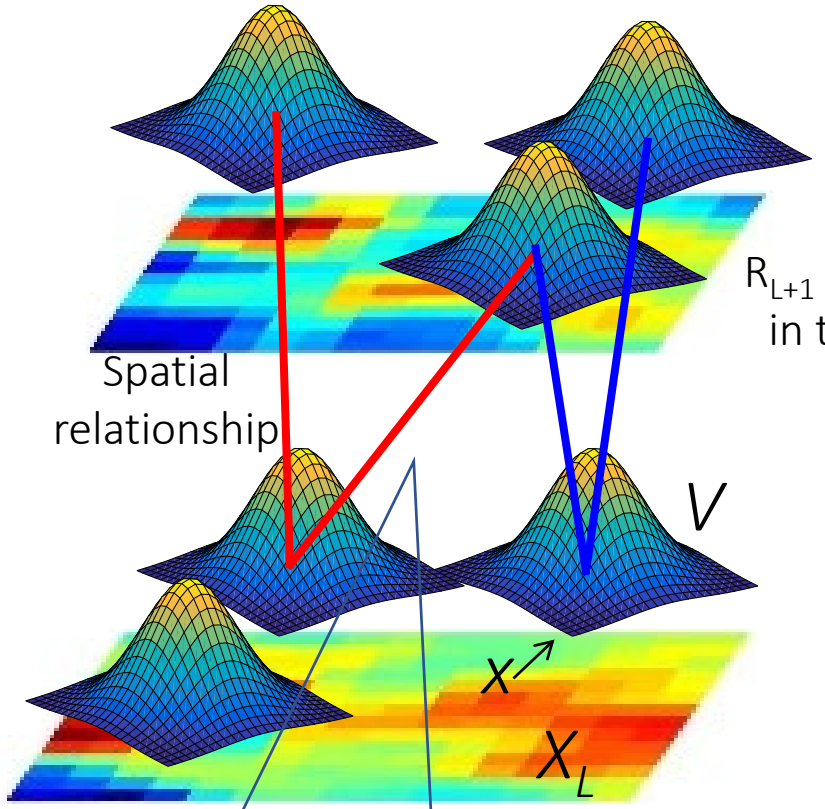
# Mining an explanatory graph

$$P(\mathbf{p}_x|V,\mathbf{R}_{L+1},\boldsymbol{\theta}_L)=\begin{cases}\gamma \prod_{V'\in E_V} P(\mathbf{p}_x|\mathbf{p}_{V'},\boldsymbol{\theta}_L)^{\lambda}, V\in\Omega_{L,d_x}\\ \gamma\tau, \qquad V=V_{\text{none}}\end{cases}$$

$$P(\mathbf{p}_x|\mathbf{p}_{V'},\boldsymbol{\theta}_L)=\mathcal{N}(\mathbf{p}_x|\mu_{V'\to V},\sigma^2_{V'})$$




Spatial relationship between two patterns → a Gaussian distribution

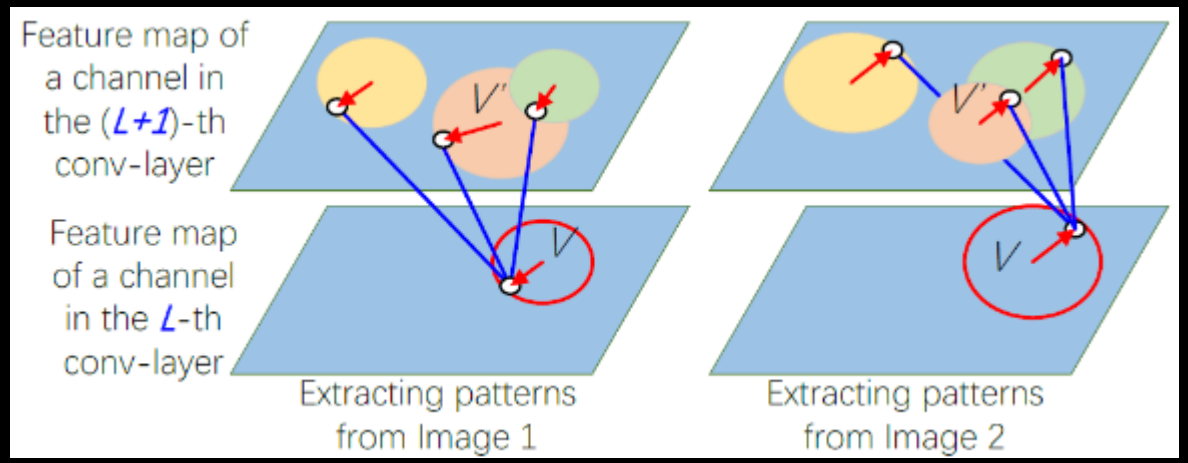



$R_{L+1}$ :Pattern inference in the (L+1)-th layer

Maximum likelihood estimation

Edges: spatial relationships between co-activated patterns in different conv-layers

# Mining an explanatory graph



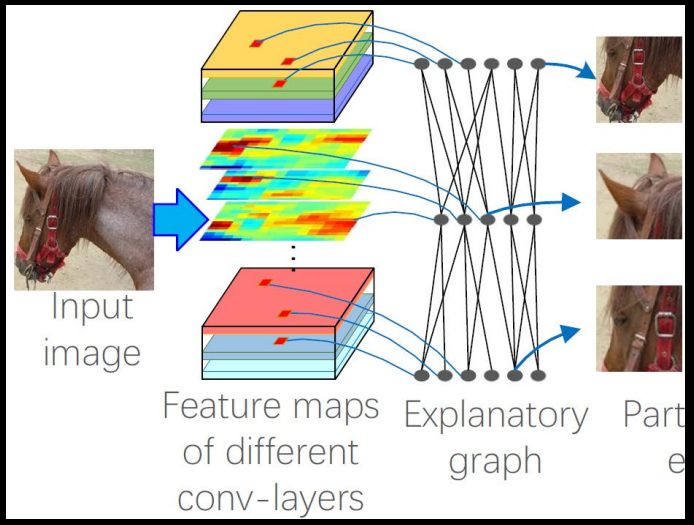

Learning node connections
Learning spatial relationship between nodes

Mining a number of cliques: a node V with multiple parents, which keep certain spatial relationships among different images.

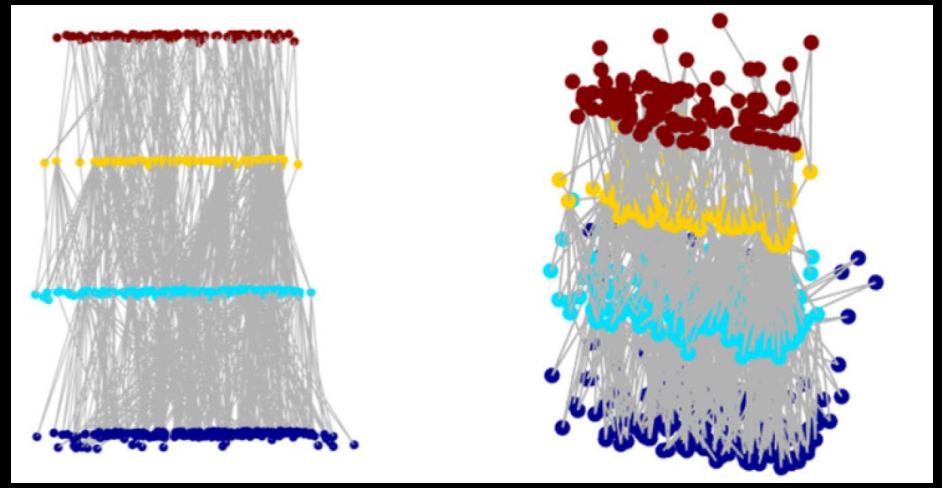
Quanshi Zhang et al. “Interpreting CNN Knowledge via an Explanatory Graph” in AAAI 2018

# Top-down mining process

- First mine patterns in the top conv-layer
- Given patterns in the top conv-layers, mine patterns in the second conv-layer
- …
- Given patterns in the (L+1)-th conv-layers, mine patterns in the L-th conv-layer




Explanatory graph for four conv-layers of a VGG-16 network



For clarity, we only show 10% of the patterns

# Using each node in the explanatory graph for part localization


Nodes in the explanatory graph


Raw filters in the CNN

We disentangle each pattern component
from each filter's feature map.

# Manually annotating the purity of semantic meaning

## Manually labeling outliers of each filter


| | VGG-16 | ResNet-50 | ResNet-152 |
|---|---|---|---|
| Raw filter map | 19.8 % | 22.1 % | 19.6 % |
| Raw filter peak | 43.8 % | 36.7 % | 39.2 % |
| Ours | **95.4 %** | **80.6 %** | **76.2 %** |

Purity of part semantics

Let people label images in each row that do not correspond to
1) the common part
OR
2) the common texture
of this filter/node

# Visualization of the mined patterns in the explanatory graph

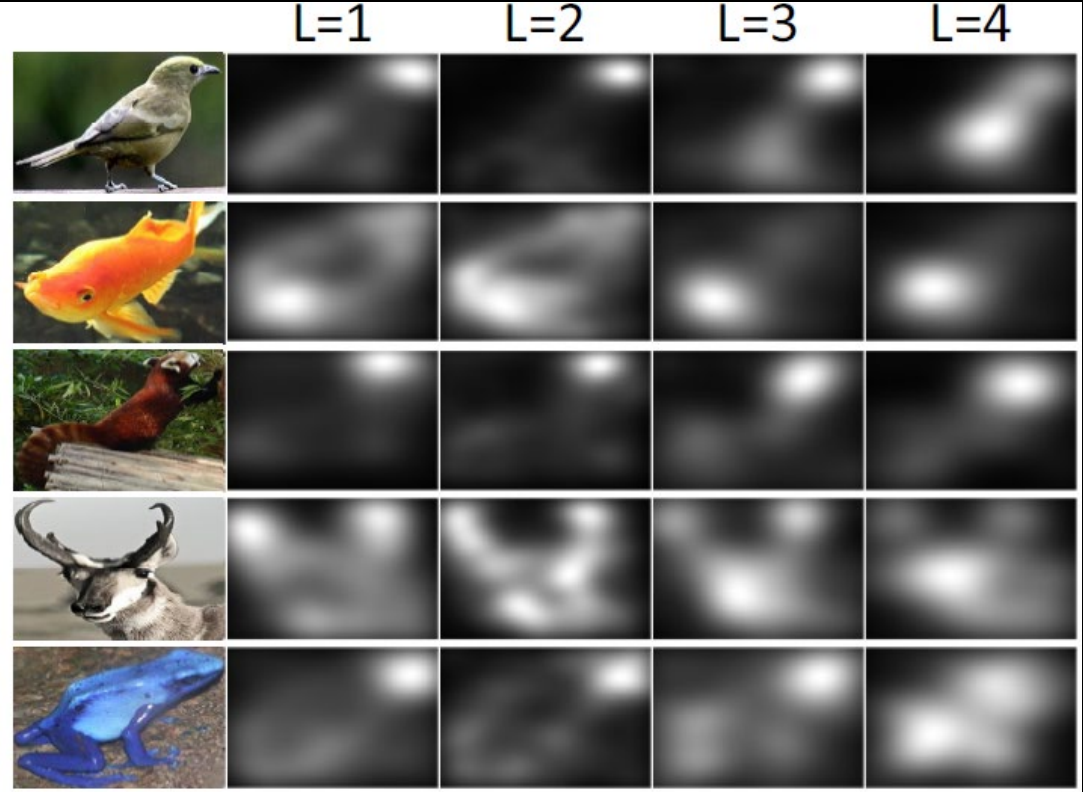



Figure 6. Heat maps of patterns.

Figure 8. Image patches corresponding to each pattern in the explanatory graph.

# Visualization of the mined patterns in the explanatory graph

Figure 7. Image synthesis result (right) based on patterns activated on an image (left). The explanatory graph only encodes major

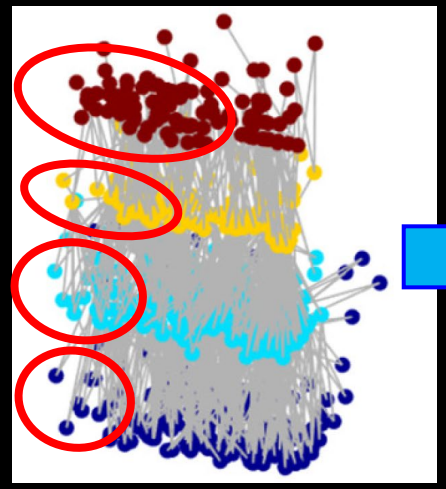
Pattern-based image synthesis has demonstrated that patterns are mainly extracted from the foreground of objects

# Knowledge transferring → One/multi-shot part localization

- The part pattern in each node is sophisticatedly learned using numerous images.
  - The retrieved nodes are not overfitted to the labeled part, but represent the common shape among all images

A single part annotation

Retrieve certain nodes for the part



Part localization

Quanshi Zhang et al. “Interpreting CNN Knowledge via an Explanatory Graph” in AAAI 2018

# Building And-Or graph for semantic hierarchy

Input:

1) An explanatory graph
2) Very few (1—3) annotations for each semantic part

Output:

An AOG as an interpretable model for semantic part localization

Associating the mined patterns with semantic parts of objects

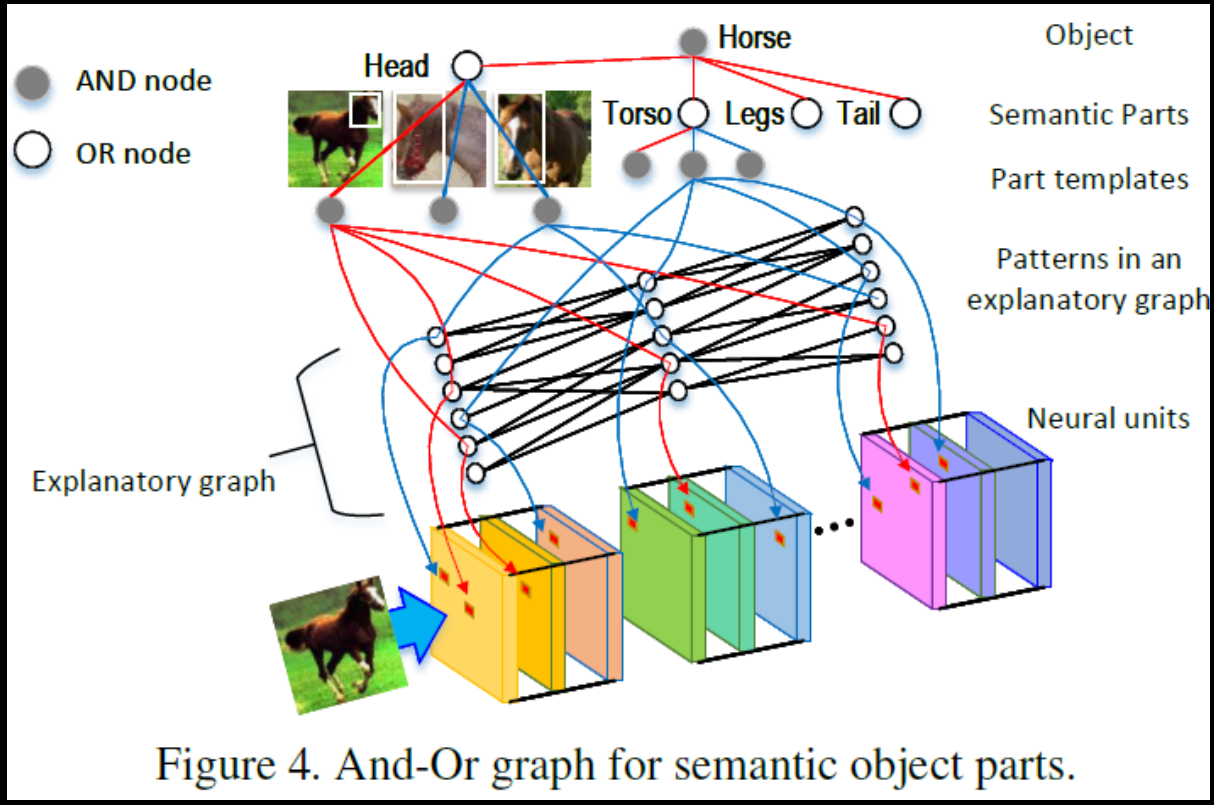
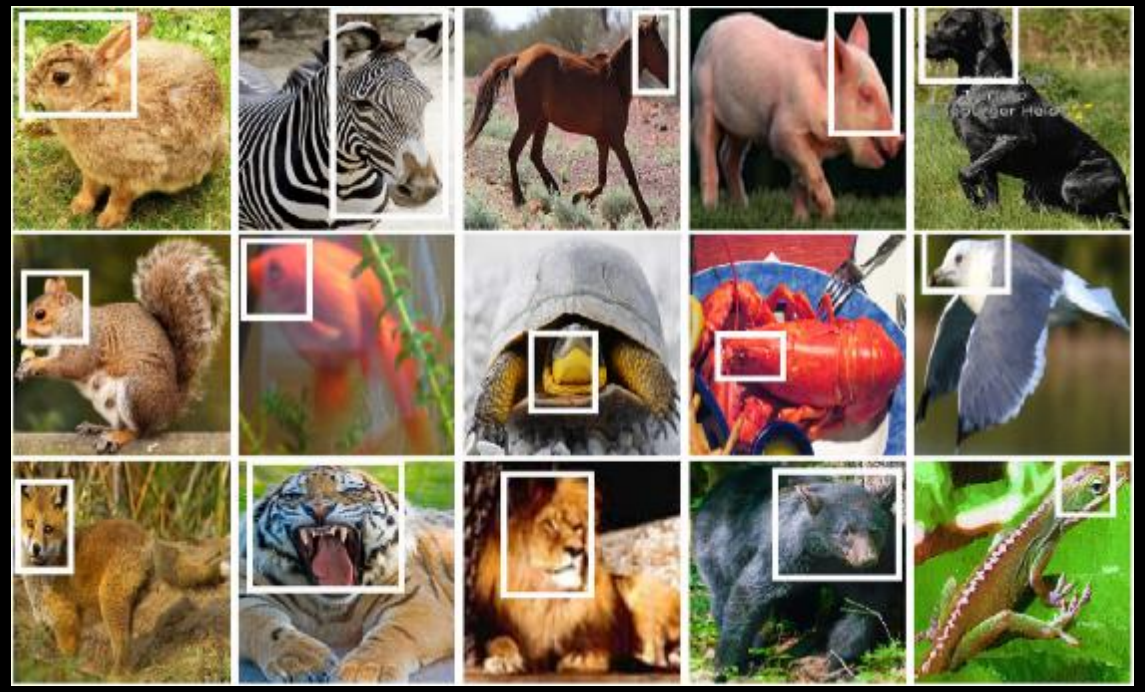



Figure 4. And-Or graph for semantic object parts.

# Explanatory graph: disentangling potential semantic components from chaotic feature maps

# Performance of few (3)-shot semantic part localization



Decrease 1/3—2/3 localization errors

| | Method | Finetune | normalized distance |
|---|---|---|---|
| no-RL | SS-DPM-Part [2] | N | 0.3469 |
| | PL-DPM-Part [18] | N | 0.3412 |
| | Part-Graph [3] | N | 0.4889 |
| unsup$^7$-RL | fc7+linearSVM | Y | 0.3120 |
| | fc7+sp+linearSVM | Y | 0.3120 |
| | Ours | Y | **0.0862** |
| sup-RL | CNN-PDD [26] | N | 0.2333 |
| | CNN-PDD-ft [26] | Y | 0.3269 |
| | Fast-RCNN (1 ft) [9] | N | 0.4517 |
| | Fast-RCNN (2 fts) [9] | Y | 0.4131 |

Table 2. Normalized distance of part localization on the CUB200-

| | | bird | cat | cow | dog | horse | sheep | **Avg** |
|---|---|---|---|---|---|---|---|---|
| no-RL | SS-DPM-Part [2] | 0.356 | 0.270 | 0.264 | 0.242 | 0.262 | 0.286 | 0.280 |
| | PL-DPM-Part [18] | 0.294 | 0.328 | 0.282 | 0.312 | 0.321 | 0.840 | 0.396 |
| | Part-Graph [3] | 0.360 | 0.208 | 0.263 | 0.205 | 0.386 | 0.500 | 0.320 |
| unsup$^7$-RL | fc7+linearSVM | 0.247 | 0.174 | 0.251 | 0.217 | 0.261 | 0.317 | 0.244 |
| | fc7+sp+linearSVM | 0.247 | 0.174 | **0.249** | 0.217 | 0.261 | 0.317 | 0.244 |
| | Ours | **0.162** | **0.130** | 0.258 | **0.137** | **0.181** | **0.192** | **0.177** |
| sup-RL | CNN-PDD [26] | 0.301 | 0.246 | 0.220 | 0.248 | 0.292 | 0.254 | 0.260 |
| | CNN-PDD-ft [26] | 0.358 | 0.268 | 0.220 | 0.200 | 0.302 | 0.269 | 0.269 |
| | Fast-RCNN (1 ft) [9] | 0.324 | 0.324 | 0.325 | 0.272 | 0.347 | 0.314 | 0.318 |
| | Fast-RCNN (2 fts) [9] | 0.350 | 0.295 | 0.255 | 0.293 | 0.367 | 0.260 | 0.303 |

Table 3. Normalized distance of part localization on the Pascal VOC Part dataset.

# Outline

- Explain semantic knowledge hidden in intermediate layers
  - How to represent CNNs using semantic graphical models
  - **How to learn disentangled, interpretable features in middle layers**
  - 在语义层面定量解释神经网络预测结果
- Explain representation capacity of deep neural networks
  - 对神经网络中层信息处理的量化分析与评测
  - 对神经网络特征表达可靠性的评测

# Background

In traditional CNNs, feature maps of a filter are usually chaotic.

Feature maps of Filter 1

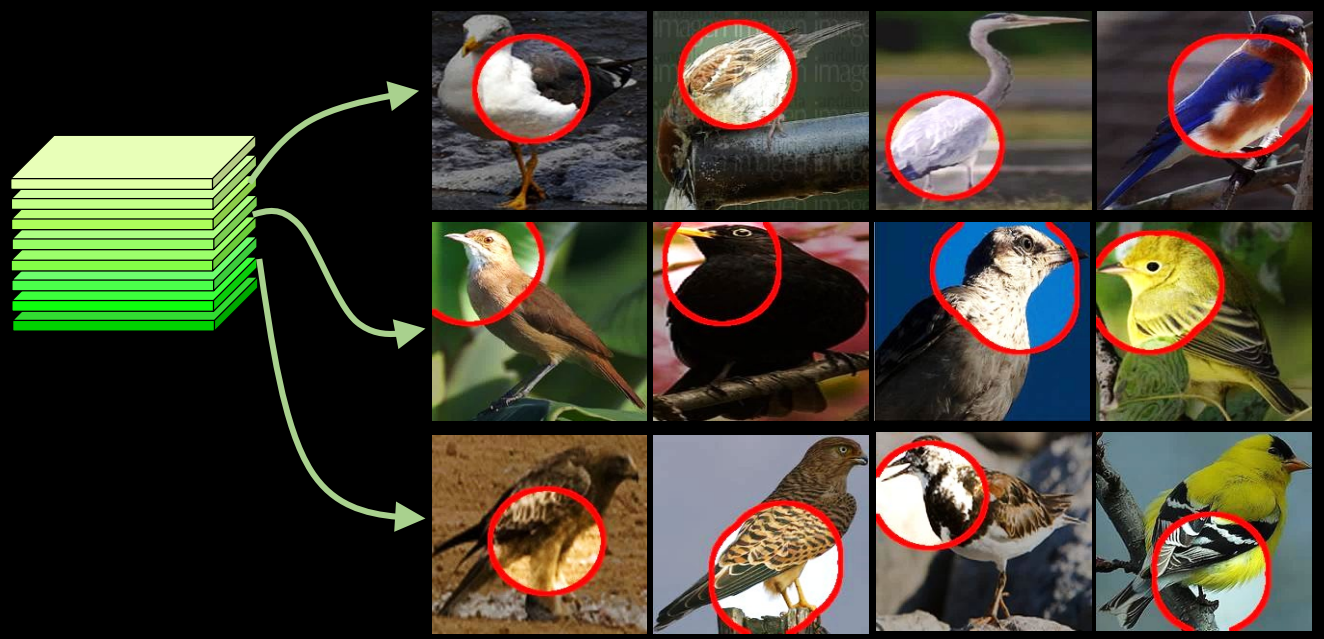
Feature maps of Filter 2

Feature maps of Filter 3

Quanshi Zhang et al. “Interpretable Convolutional Neural Networks” in CVPR 2018

# Objective

**Without** additional part annotations, learn a CNN, where each filter represents a specific part through different objects.



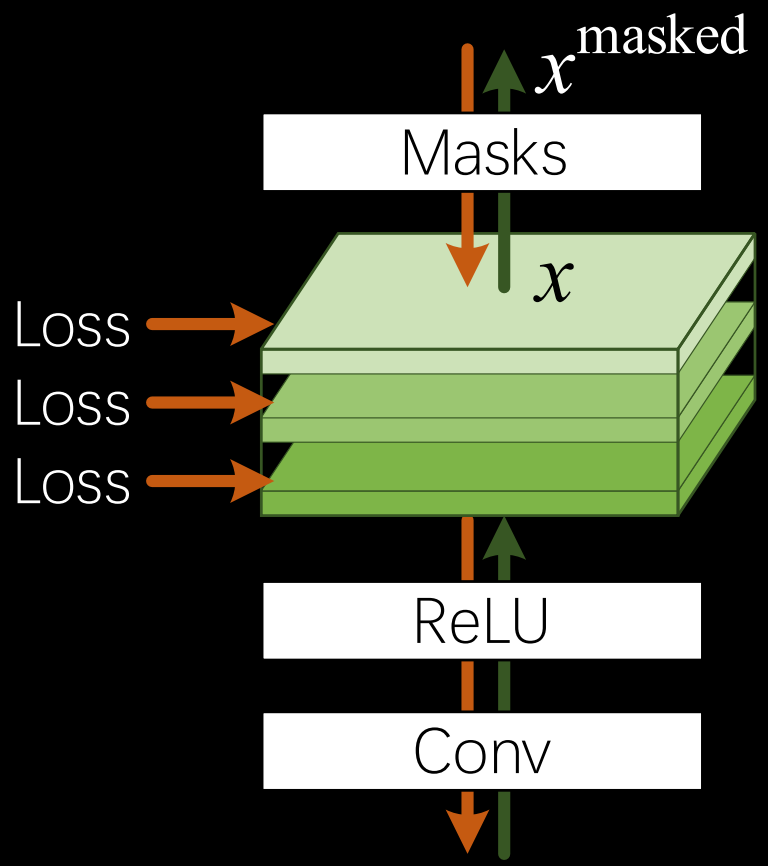
Neural activations of 3 interpretable filters


# Input & Output: Interpretable CNNs

- Input
  - Training samples $(X_i, Y_i)$ for a certain task
    - Applicable to different tasks, e.g., classification & segmentations
    - Applicable to different CNNs, e.g., AlexNet, VGG-16, VGG-M, VGG-S
  - **No** annotations of parts or textures are used.
- Output
  - An interpretable CNN with disentangled filters

# Network structure

We add a loss to each channel to construct an interpretable layer

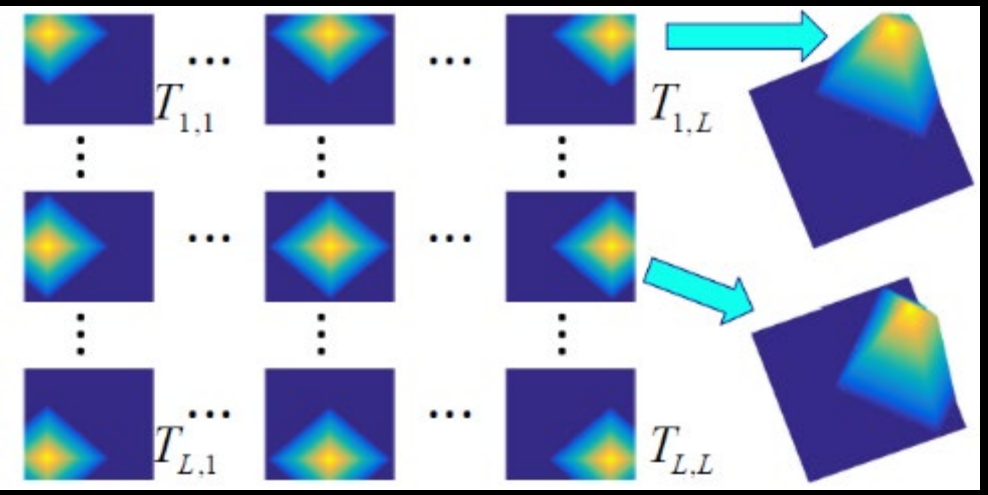



$$Loss = \underbrace{Loss(\hat{y}, y^*)}_{\text{task loss}} + \sum_f \underbrace{Loss_f(x)}_{\text{filter loss}}$$

The filter loss boosts the mutual information between feature maps $\boldsymbol{X}$ and a set of pre-defined part locations $\boldsymbol{T}$.

$$\mathbf{Loss}_f = -\,MI(\mathbf{X};\mathbf{T}) \quad \text{for filter } f$$

# Network structure




$$Loss = \underbrace{Loss(\hat{y}, y^*)}_{\text{task loss}} + \sum_f \underbrace{Loss_f(x)}_{\text{filter loss}}$$

$$\mathbf{Loss}_f = -\,MI(\mathbf{X};\mathbf{T}) \quad \text{for filter } f$$

$$Loss = -\,H(\mathbf{T}) + H(\mathbf{T}' = \{T^-, \mathbf{T}^+\}|\mathbf{X}) + \sum_x p(\mathbf{T}^+, x) H(\mathbf{T}^+ = \{T_\mu\}|X = x)$$

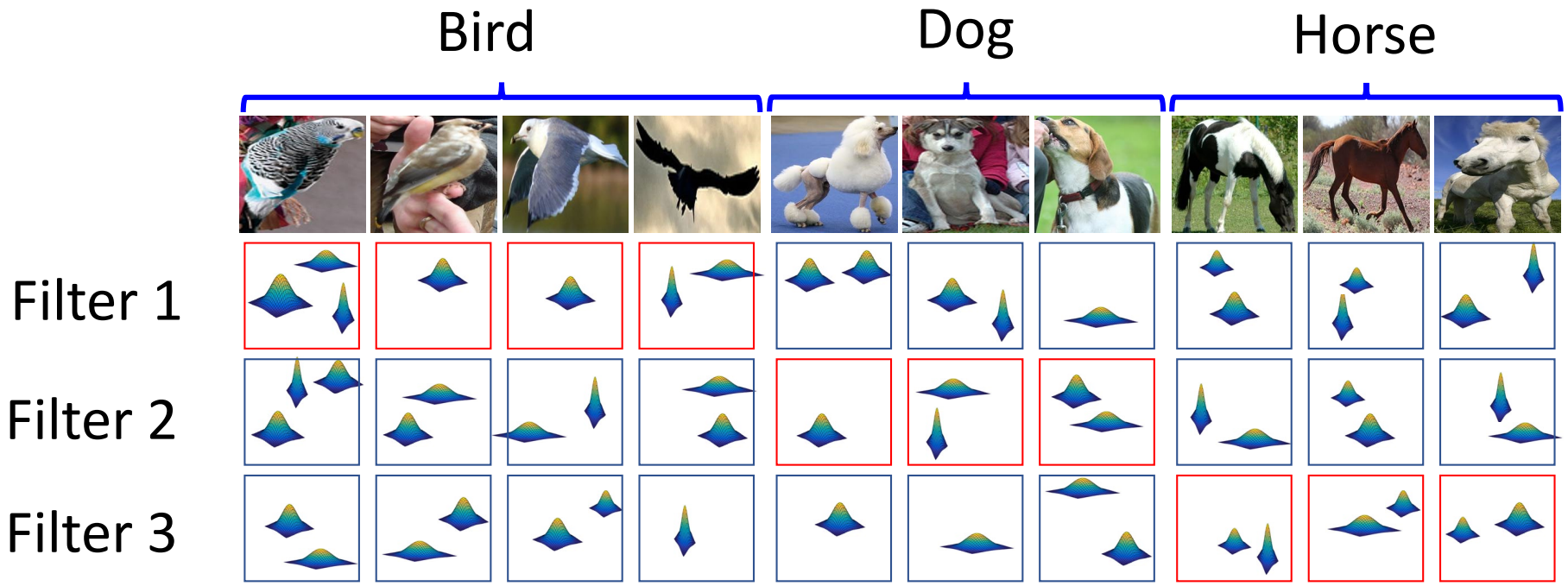
A constant

Entropy of Inter-category activations

Entropy of the spatial distribution of activations

# Learning

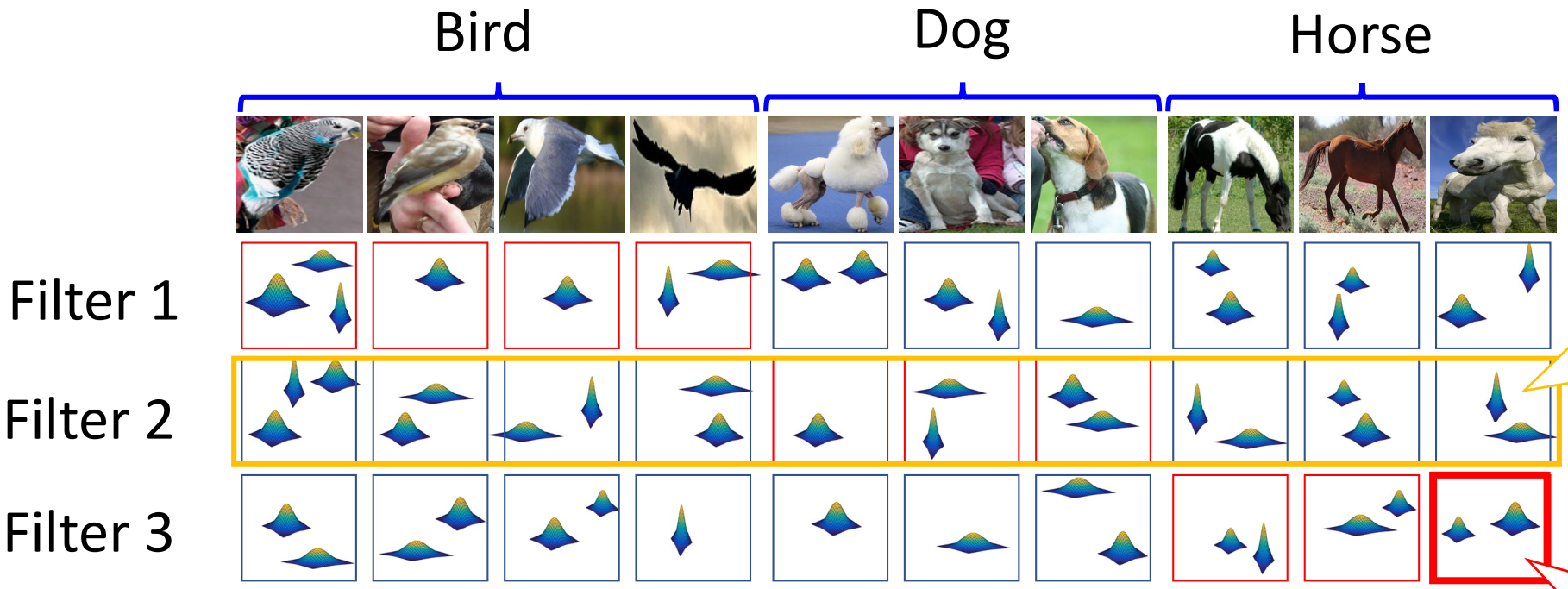





From chaotic feature maps to the disentangled maps of object parts

# Learning





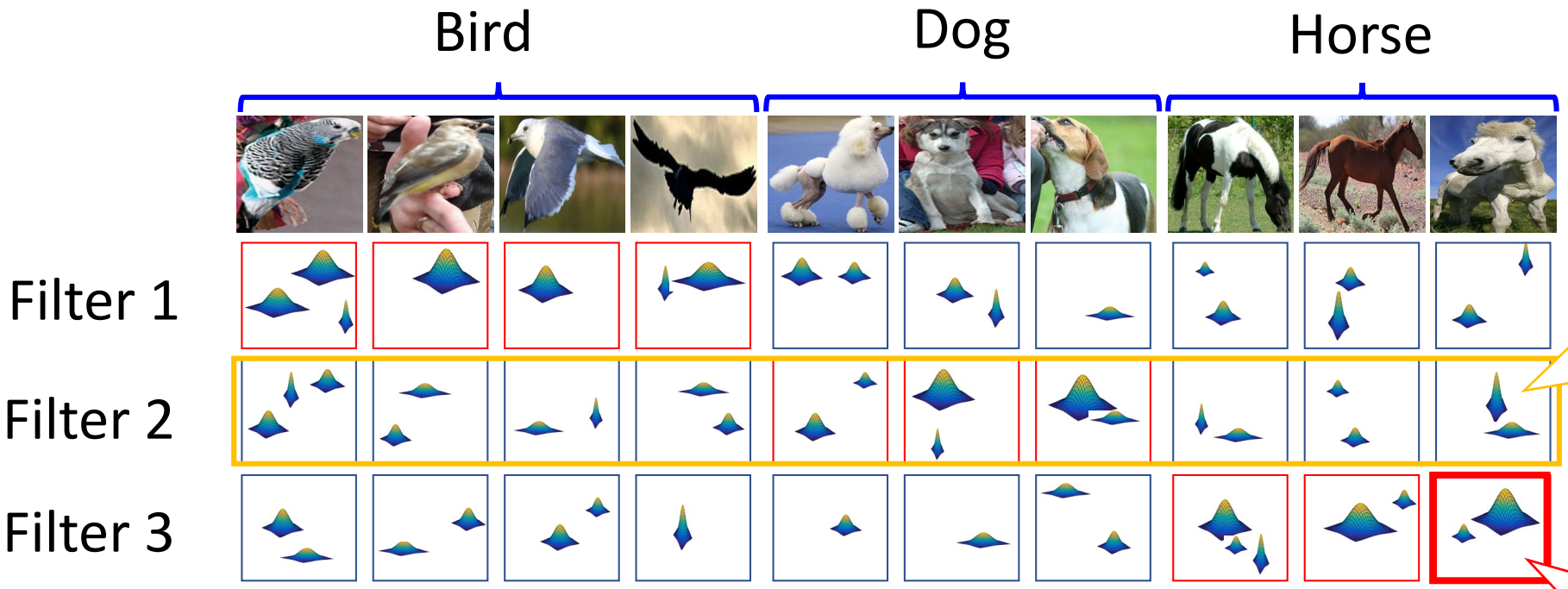

The loss encourages a low inter-class entropy

The loss encourages a low spatial entropy

From chaotic feature maps to the disentangled maps of object parts

# Learning





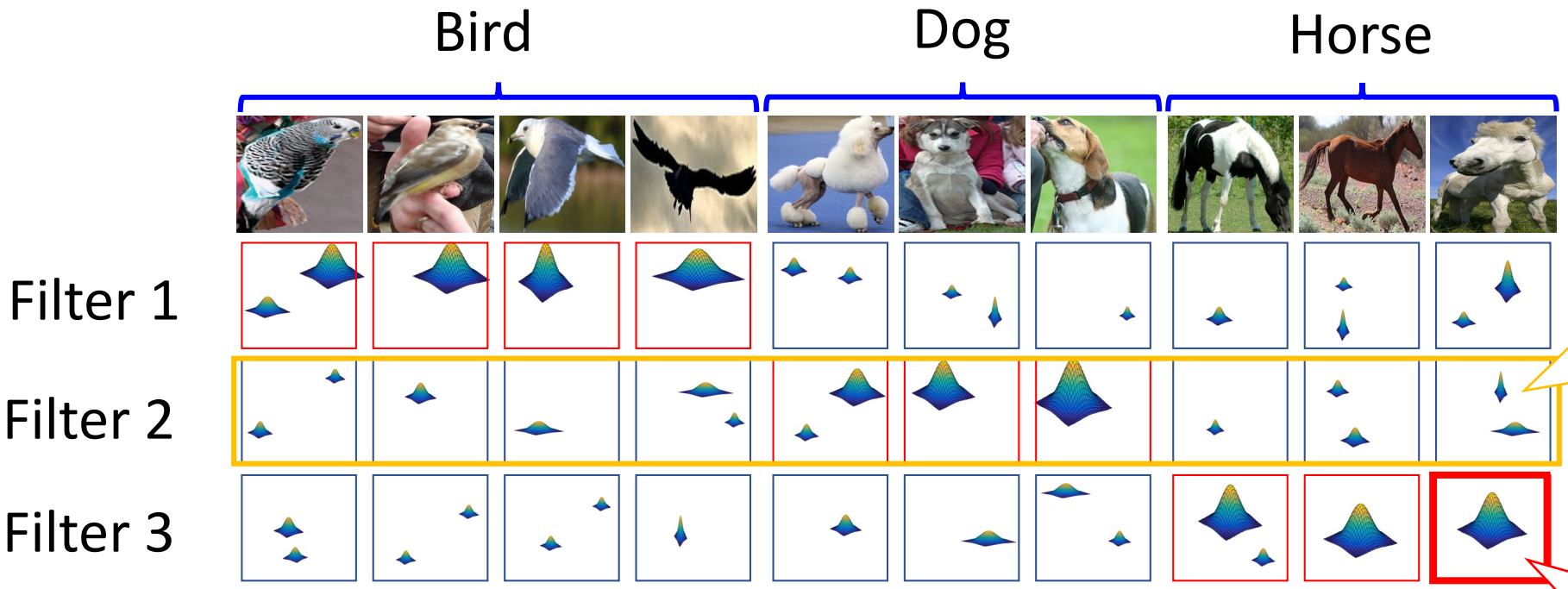

The loss encourages a low inter-class entropy

The loss encourages a low spatial entropy

From chaotic feature maps to the disentangled maps of object parts

# Learning





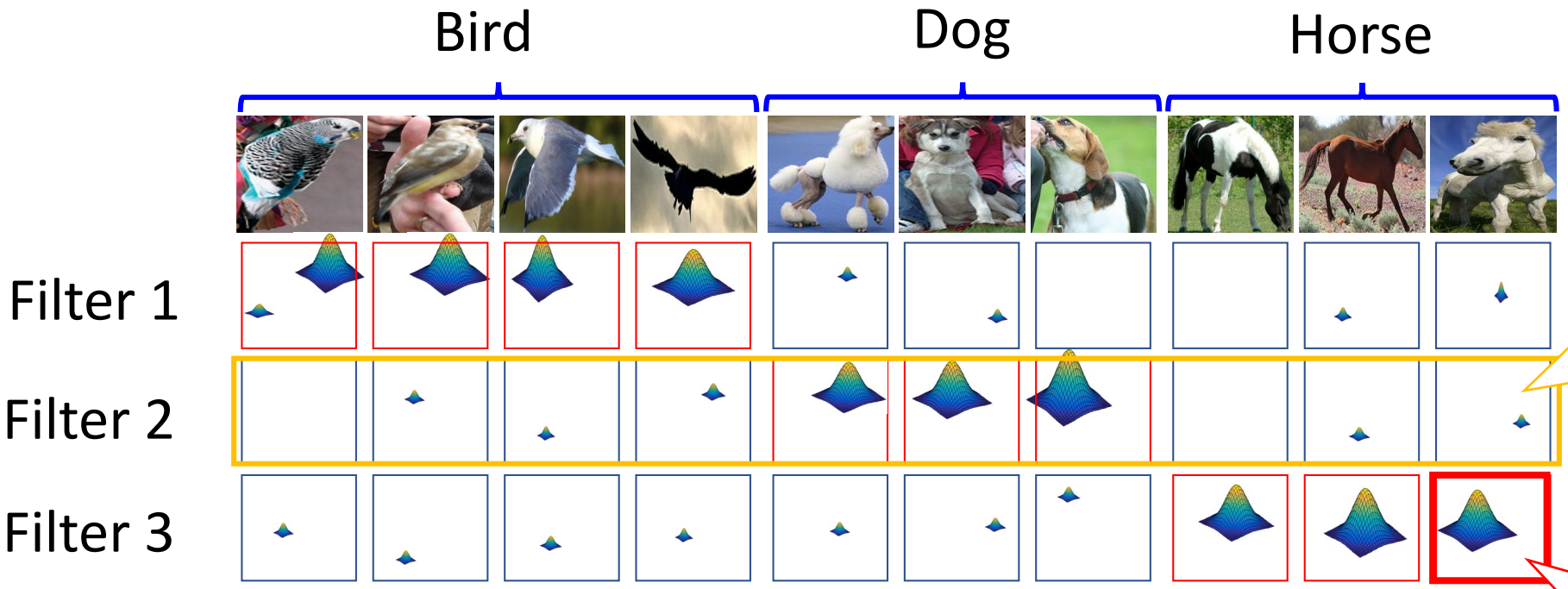

The loss encourages a low inter-class entropy

The loss encourages a low spatial entropy

From chaotic feature maps to the disentangled maps of object parts

# Learning





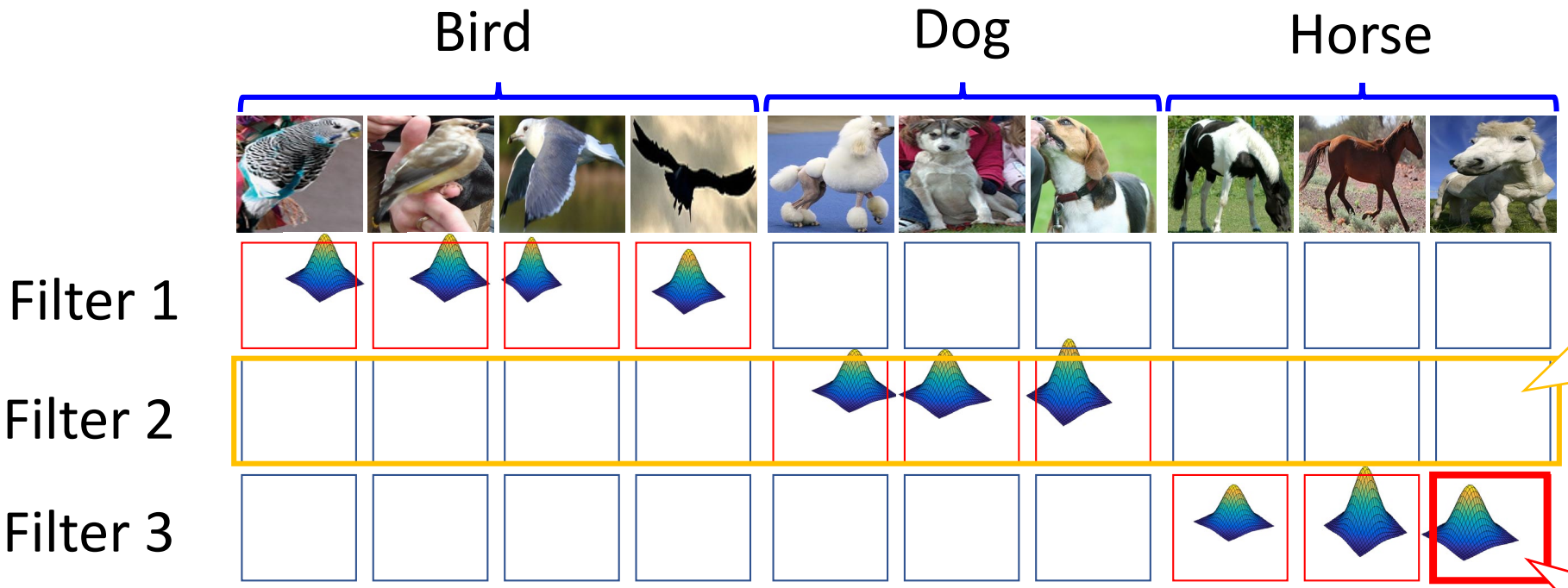

The loss encourages a low inter-class entropy

The loss encourages a low spatial entropy

From chaotic feature maps to the disentangled maps of object parts

# Learning






The loss encourages a low inter-class entropy

The loss encourages a low spatial entropy

From chaotic feature maps to the disentangled maps of object parts

# Activation regions of interpretable filters

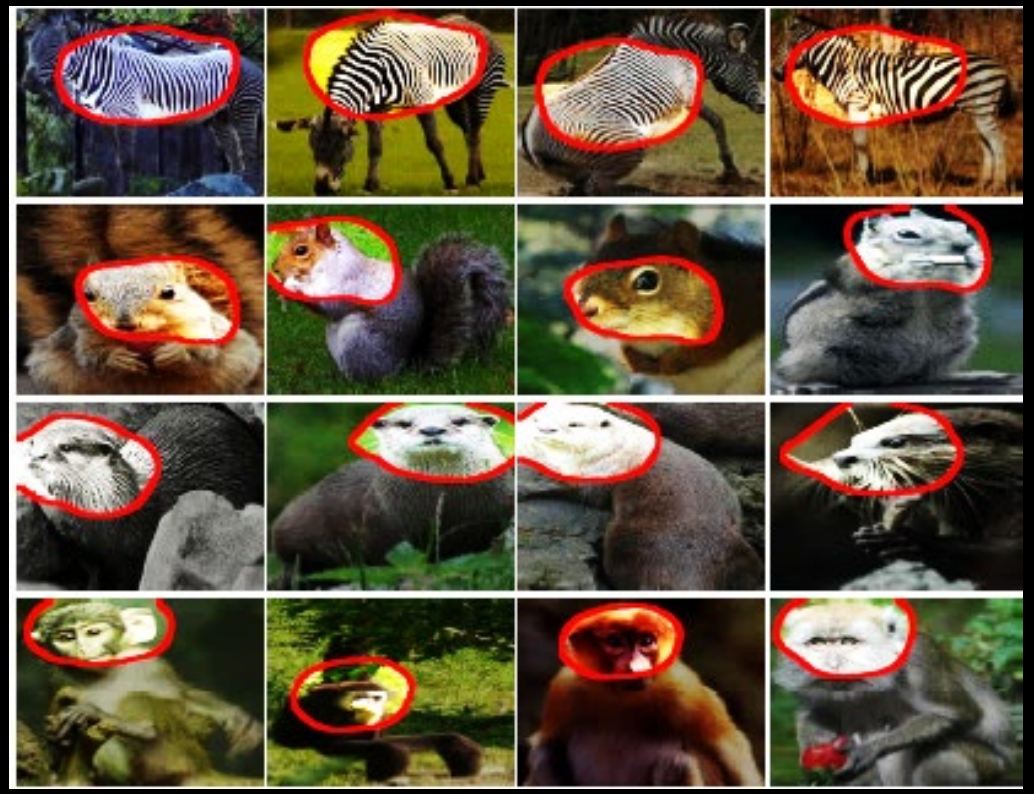
Filter 1

Filter 2

Filters 3 & 4

Filter

Filter



# Our method learns filters with much higher interpretability

| | gold. | bird | frog | turt. | liza. | koala | lobs. | dog | fox | cat | lion | tiger | bear | rabb. | hams. | squi. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AlexNet | 0.161 | 0.167 | 0.152 | 0.153 | 0.175 | 0.128 | 0.123 | 0.144 | 0.143 | 0.148 | 0.137 | 0.142 | 0.144 | 0.148 | 0.128 | 0.149 |
| AlexNet, interpretable | **0.084** | **0.095** | **0.090** | **0.107** | **0.097** | **0.079** | **0.077** | **0.093** | **0.087** | **0.095** | **0.084** | **0.090** | **0.095** | **0.095** | **0.077** | **0.095** |
| VGG-16 | 0.153 | 0.156 | 0.144 | 0.150 | 0.170 | 0.127 | 0.126 | 0.143 | 0.137 | 0.148 | 0.139 | 0.144 | 0.143 | 0.146 | 0.125 | 0.150 |
| VGG-16, interpretable | **0.076** | **0.099** | **0.086** | **0.115** | **0.113** | **0.070** | **0.084** | **0.077** | **0.069** | **0.086** | **0.067** | **0.097** | **0.081** | **0.079** | **0.066** | **0.065** |
| VGG-M | 0.161 | 0.166 | 0.151 | 0.153 | 0.176 | 0.128 | 0.125 | 0.145 | 0.145 | 0.150 | 0.140 | 0.145 | 0.144 | 0.150 | 0.128 | 0.150 |
| VGG-M, interpretable | **0.088** | **0.088** | **0.089** | **0.108** | **0.099** | **0.080** | **0.074** | **0.090** | **0.082** | **0.103** | **0.079** | **0.089** | **0.101** | **0.097** | **0.082** | **0.095** |
| VGG-S | 0.158 | 0.166 | 0.149 | 0.151 | 0.173 | 0.127 | 0.124 | 0.143 | 0.142 | 0.148 | 0.138 | 0.142 | 0.143 | 0.148 | 0.128 | 0.146 |
| VGG-S, interpretable | **0.087** | **0.101** | **0.093** | **0.107** | **0.096** | **0.084** | **0.078** | **0.091** | **0.082** | **0.101** | **0.082** | **0.089** | **0.097** | **0.091** | **0.076** | **0.098** |

| | horse | zebra | swine | hippo | catt. | sheep | ante. | camel | otter | arma. | monk. | elep. | red pa. | gia.pa. | **Avg.** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AlexNet | 0.152 | 0.154 | 0.141 | 0.141 | 0.144 | 0.155 | 0.147 | 0.153 | 0.159 | 0.160 | 0.139 | 0.125 | 0.140 | 0.125 | 0.146 |
| AlexNet, interpretable | **0.098** | **0.084** | **0.091** | **0.089** | **0.097** | **0.101** | **0.085** | **0.102** | **0.104** | **0.095** | **0.090** | **0.085** | **0.084** | **0.073** | **0.091** |
| VGG-16 | 0.150 | 0.153 | 0.141 | 0.140 | 0.140 | 0.150 | 0.144 | 0.149 | 0.154 | 0.163 | 0.136 | 0.129 | 0.143 | 0.125 | 0.144 |
| VGG-16, interpretable | **0.106** | **0.077** | **0.094** | **0.083** | **0.102** | **0.097** | **0.091** | **0.105** | **0.093** | **0.100** | **0.074** | **0.084** | **0.067** | **0.063** | **0.085** |
| VGG-M | 0.151 | 0.158 | 0.140 | 0.140 | 0.143 | 0.155 | 0.146 | 0.154 | 0.160 | 0.161 | 0.140 | 0.126 | 0.142 | 0.127 | 0.147 |
| VGG-M, interpretable | **0.095** | **0.080** | **0.095** | **0.084** | **0.092** | **0.094** | **0.077** | **0.104** | **0.102** | **0.093** | **0.086** | **0.087** | **0.089** | **0.068** | **0.090** |
| VGG-S | 0.149 | 0.155 | 0.139 | 0.140 | 0.141 | 0.155 | 0.143 | 0.154 | 0.158 | 0.157 | 0.140 | 0.125 | 0.139 | 0.125 | 0.145 |
| VGG-S, interpretable | **0.096** | **0.080** | **0.092** | **0.088** | **0.094** | **0.101** | **0.077** | **0.102** | **0.105** | **0.094** | **0.090** | **0.086** | **0.078** | **0.072** | **0.090** |

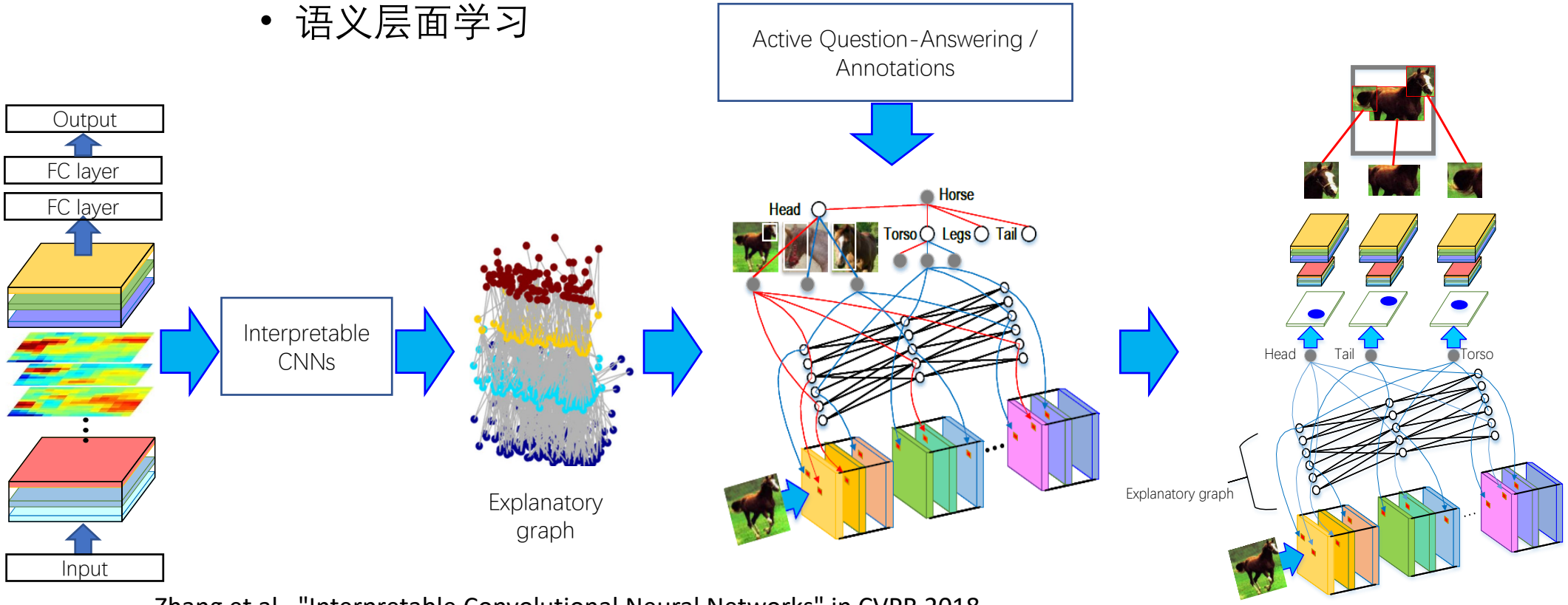
Table 3. Location instability of filters ($\mathbf{E}_{f,k}[D_{f,k}]$) in CNNs that are trained for single-category classification using the ILSVRC 2013 DET Animal-Part dataset [36]. Filters in our interpretable CNNs exhibited significantly lower localization instability than ordinary CNNs.

# Classification performance

| | multi-category | | | single-category | | |
|---|---|---|---|---|---|---|
| | ILSVRC Part | VOC Part | | ILSVRC Part | VOC Part | CUB200 |
| | logistic[4] | logistic[4] | softmax | | | |
| AlexNet | – | – | – | **96.28** | **95.40** | **95.59** |
| interpretable | – | – | – | 95.38 | 93.93 | 95.35 |
| VGG-M | 96.73 | 93.88 | 81.93 | **97.34** | **96.82** | **97.34** |
| interpretable | **97.99** | **96.19** | **88.03** | 95.77 | 94.17 | 96.03 |
| VGG-S | 96.98 | 94.05 | 78.15 | **97.62** | **97.74** | **97.24** |
| interpretable | **98.72** | **96.78** | **86.13** | 95.64 | 95.47 | 95.82 |
| VGG-16 | – | 97.97 | 89.71 | **98.58** | **98.66** | **98.91** |
| interpretable | – | **98.50** | **91.60** | 96.67 | 95.39 | 96.51 |

Our interpretable CNNs outperformed traditional CNNs in multi-category classification.

- 打通神经网络模型与语义图模型的壁垒
  - 如何将神经网络中层知识转化为语义图模型表达?
  - 人的大脑信号是混沌的，但是人的知识体系是清晰符号化的。人如何做到对混沌信息的提炼与总结?
  - 如何交互式学习?
    - 小样本学习
    - 语义层面学习




Zhang et al., "Interpretable Convolutional Neural Networks" in CVPR 2018
Zhang et al., "Interpreting CNN Knowledge via an Explanatory Graph" in AAAI 2018

# Outline

- Explain semantic knowledge hidden in intermediate layers
  - How to represent CNNs using semantic graphical models
  - How to learn disentangled, interpretable features in middle layers
  - **在语义层面定量解释神经网络预测结果**
- Explain representation capacity of deep neural networks
  - 对神经网络中层信息处理的量化分析与评测
  - 对神经网络特征表达可靠性的评测

# 在语义层面定量解释神经网络预测结果

## 研究成果

- Runjin Chen, Hao Chen, Jie Ren, Ge Huang, Quanshi Zhang, “Explaining Neural Networks Semantically and Quantitatively” in ICCV, 2019 (Oral)

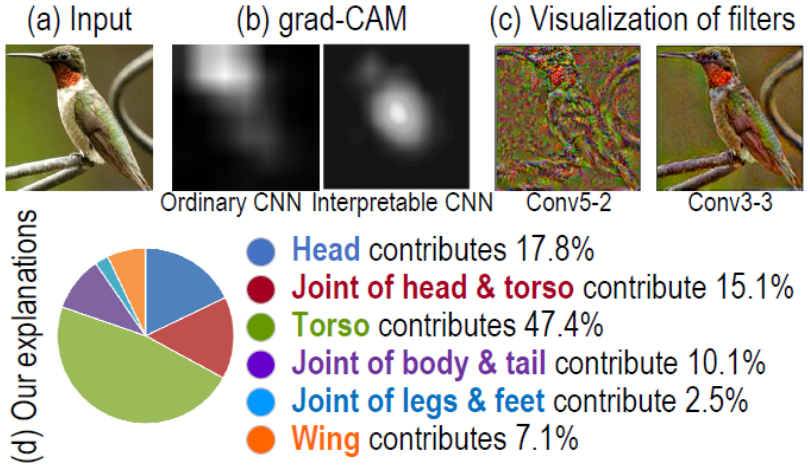
## 语义层面解释

## 定量解释

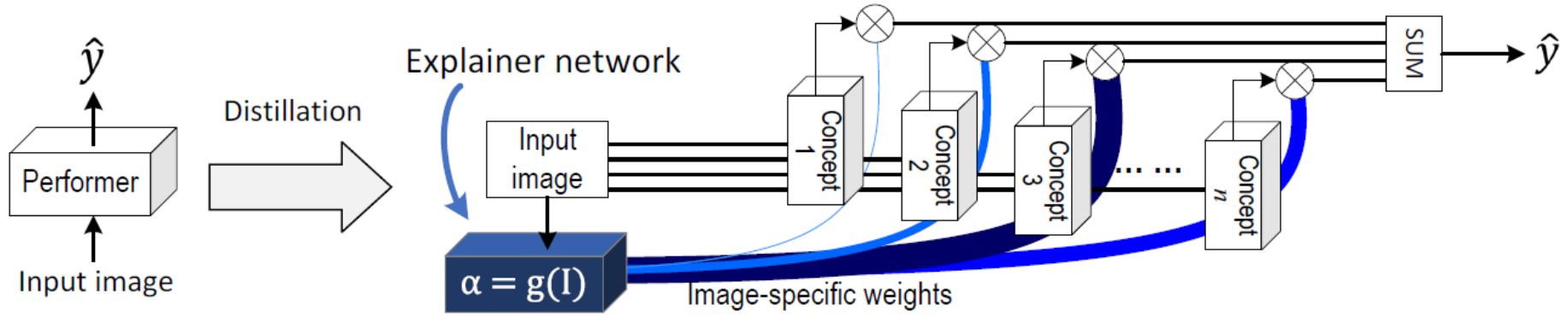
- 多少特征可以被解释
- 可解释的特征分别代表什么语义
- 每种语义分别贡献多少分数
- 多少特征尚不能被解释

# 问题描述

- 输入
  - 预训练神经网络
  - 若干预训练语义特征
- 输出
  - $Y=+y_1(l)w_1(l)+y_2(l)w_2(l)+y_3(l)w_3(l)+\cdots+y_n(l)w_n(l)+b$
  - 计算W(l)辅助解释性神经网络，将神经网络输出结果量化为各个语义分量的线性和。



# 解释建模




$$\hat{y} \approx g_{\boldsymbol{\theta}}(I)^{\top}\mathbf{y} + b = b + \sum\nolimits_i \alpha_i \cdot y_i$$

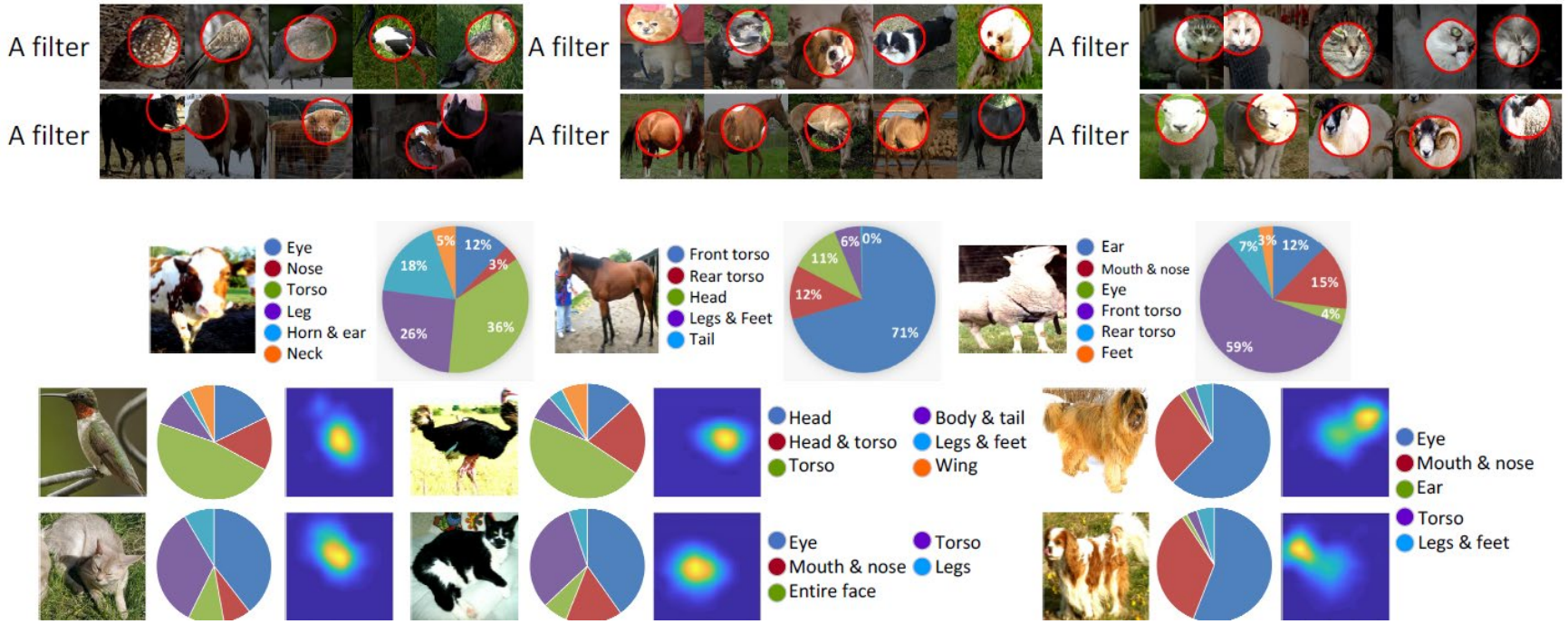
Quantitative contribution of the $i$-th visual concept

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^{\top} = g_{\boldsymbol{\theta}}(I)$$

$$\mathbf{y} = [y_1, y_2, \ldots, y_n]^{\top}, \qquad y_i = f_i(I), \quad i = 1, 2, \ldots, n$$

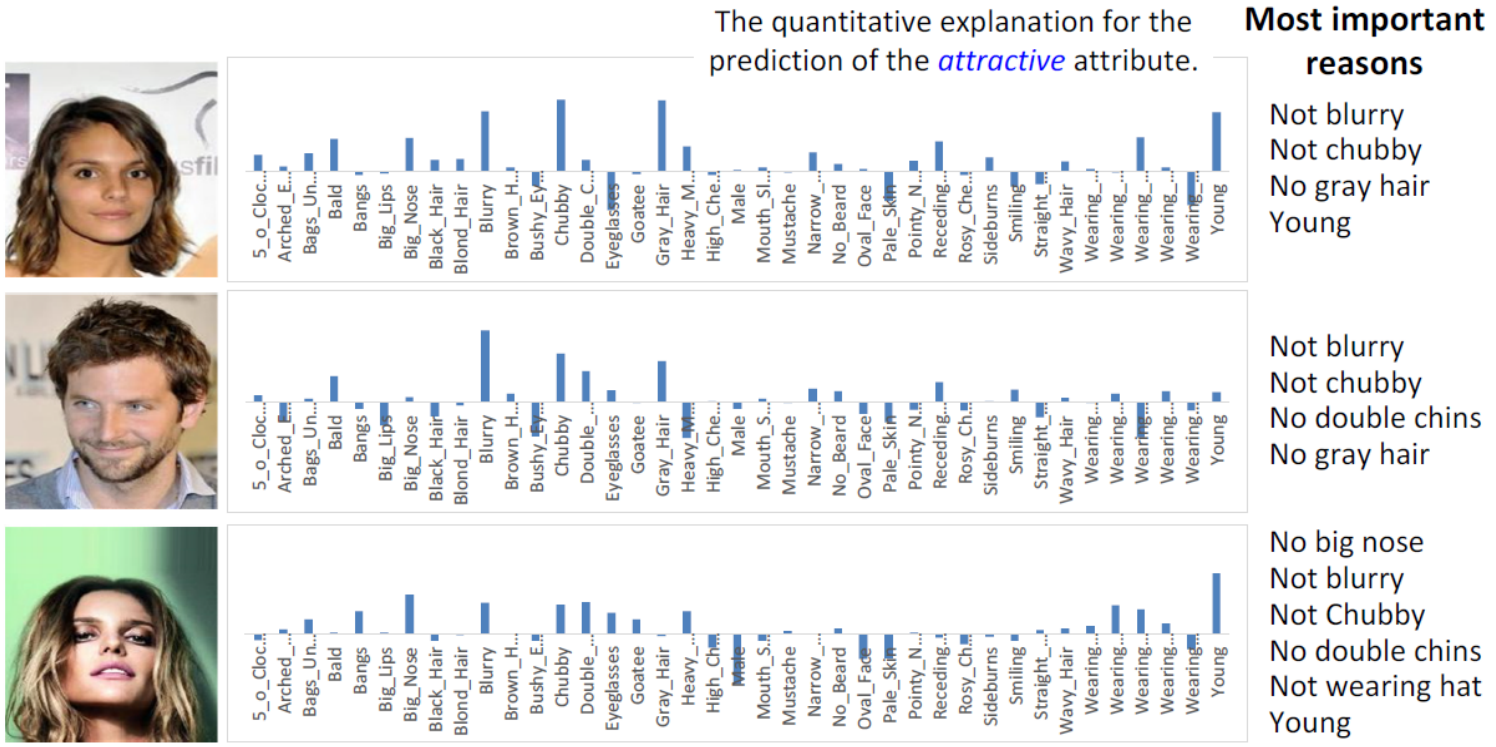
- 知识蒸馏： 目标网络→ 解释性网络
- 解释性神经网络alpha，建模了神经网络的各语义权重的非线性变换。

# 物体检测→解释为对part的检测




- 我们CVPR2018工作，保证每个中层卷积和建模某个特定的物体组成部分

# 解释何为漂亮




- 为不同的人脸，分析出不同的漂亮/帅气的原因。

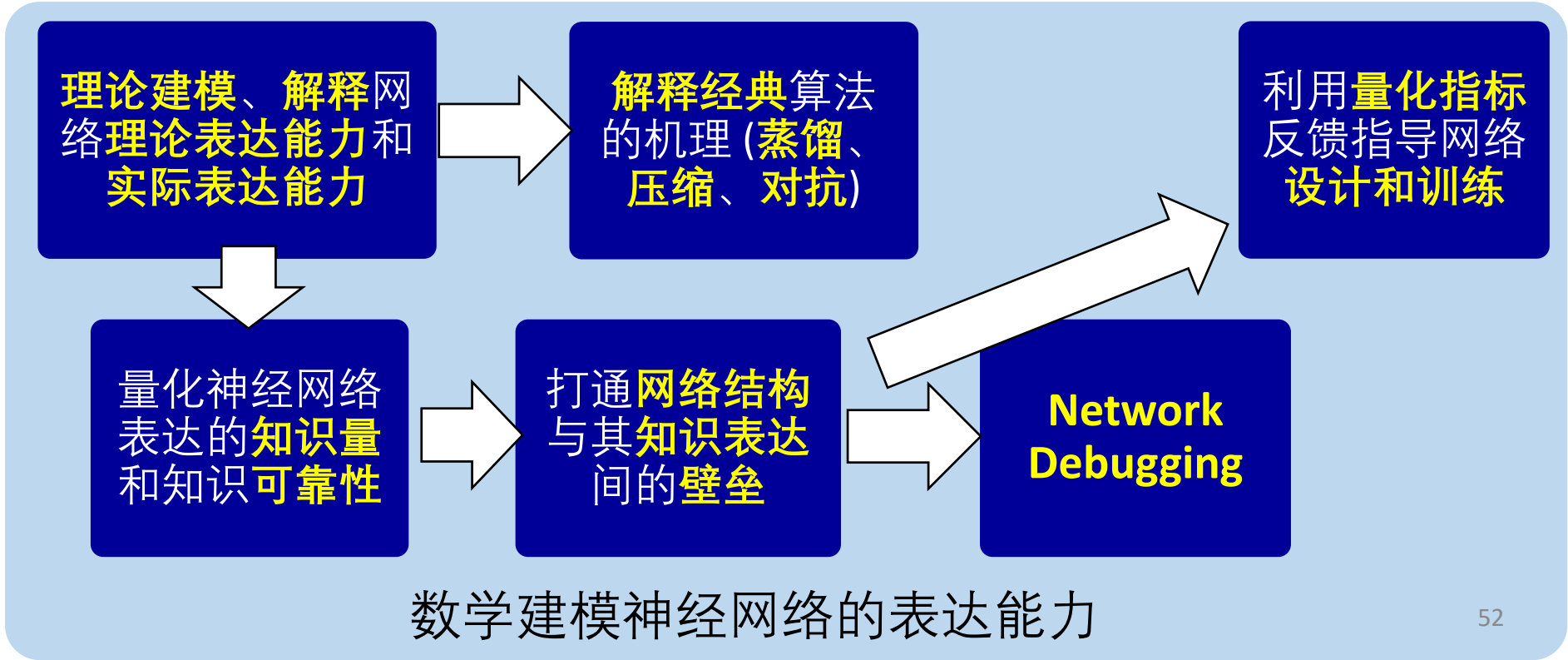
# Outline

- Explain semantic knowledge hidden in intermediate layers
  - How to represent CNNs using semantic graphical models
  - How to learn disentangled, interpretable features in middle layers
  - 在语义层面定量解释神经网络预测结果
- Explain representation capacity of deep neural networks
  - **对神经网络中层信息处理的量化分析与评测**
  - 对神经网络特征表达可靠性的评测

# 研究概述

- 对神经网络中层信息处理的量化分析与评测
- 对神经网络特征表达可靠性的评测

## 神经网络可解释性问题：炼丹 → 化学






52

# 对神经网络中层信息处理的量化分析与评测

## 论文成果

- Chaoyu Guan, Xiting Wang, **Quanshi Zhang (Corresponding author)**, Runjin Chen, Di He, Xing Xie, “Towards A Deep and Unified Understanding of Deep Neural Models in NLP” in ICML, 2019
- Haotian Ma, Yinqing Zhang, Fan Zhou, **Quanshi Zhang**, “Quantifying Layerwise Information Discarding of Neural Networks” in arXiv:1906.04109, 2019

# 对神经网络中层信息处理的量化分析与评测

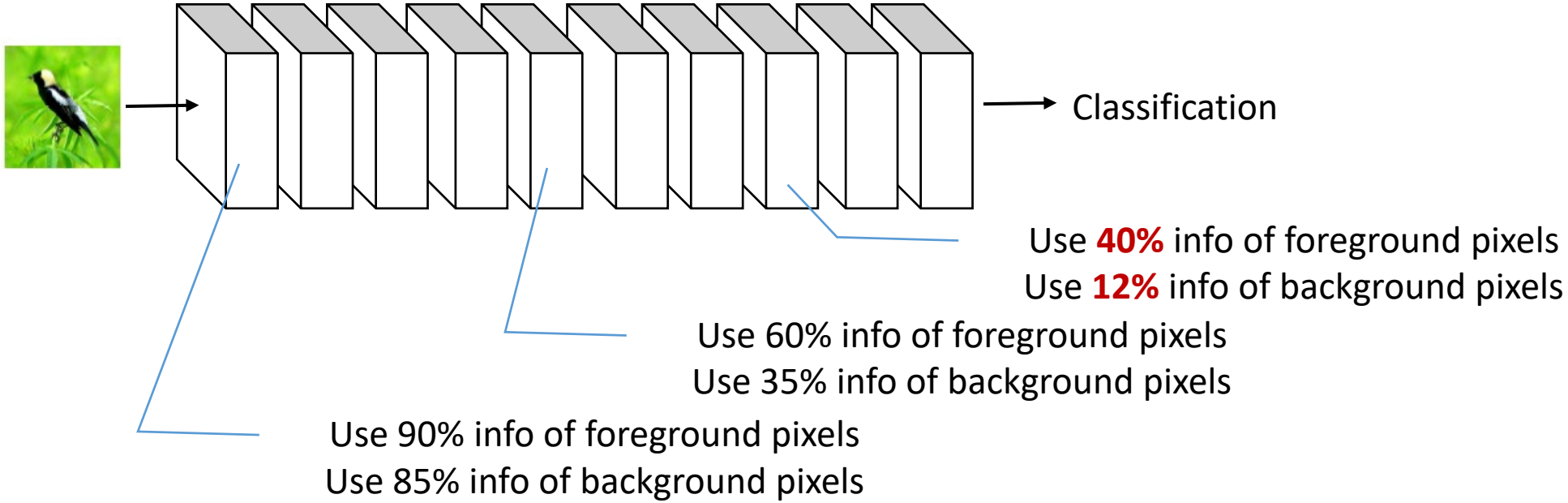
- 量化神经网络中层特征知识信息量，作为一般性理论工具
  - 诊断经典神经网络的特征表达特点
  - 解释现有深度学习技术的信息处理特点
    - 网络压缩
    - 知识蒸馏
    - 神经网络结构修改

# Understanding DNNs as layerwise discarding of input information

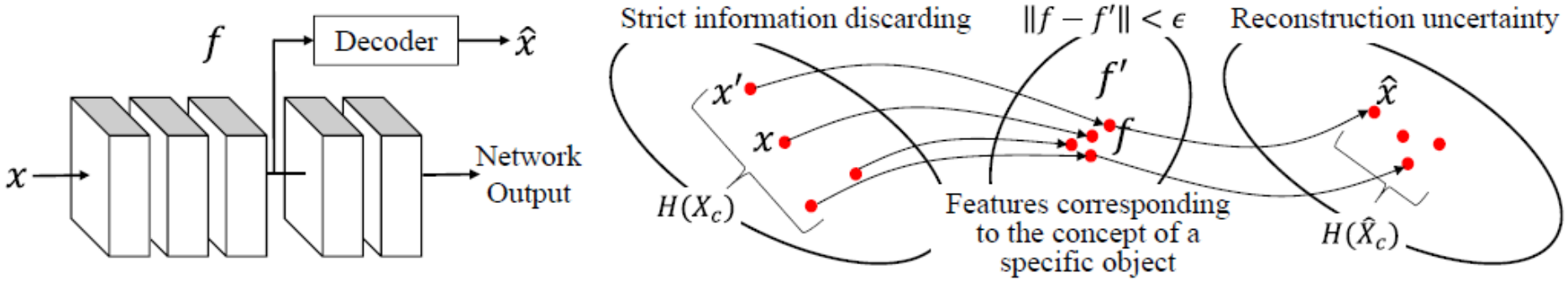
- A DNN → layerwise discarding of input information
  - Discard less foreground information
  - Discard more background information

  } Enable reliable predictions

- Measure two types of information discarding
  - How much information of the input **is used to** compute the feature
  - How much information of the input **can be recovered from** the feature



# Quantifying layerwise discarding of the input information



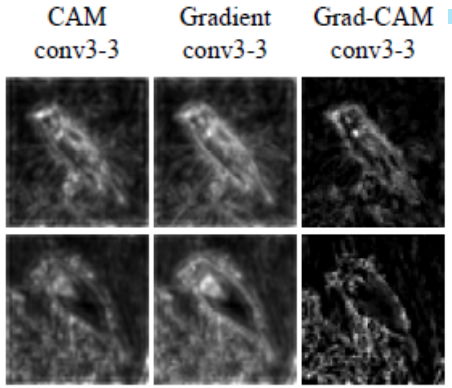

$$\max\ H(X_c) = -\sum_{x' \in X_c} p(x') \log p(x'), \qquad \text{s.t.}\quad \mathbb{E}_{f' \in F_c}\left[\|f' - f\|^2\right] = \epsilon$$

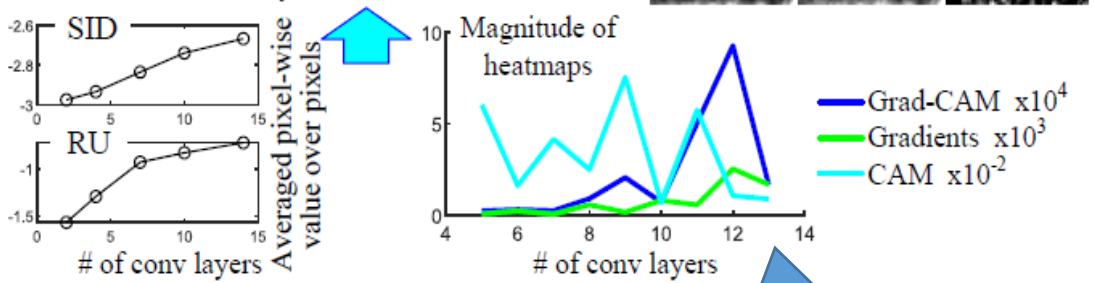
- **Assumption:** The concept of a specific object instance is assumed to be represented as a small range of intermediate-layer features $\mathbb{E}_{f' \in F_c}\left[\|f' - f\|^2\right] = \epsilon$
- Adding noise to input. Compute the entropy of the input, when features satisfies $\mathbb{E}_{f' \in F_c}\left[\|f' - f\|^2\right] = \epsilon$
  - Measure how much information in each input pixel **is discarded for** the computation
- When features satisfies $\mathbb{E}_{f' \in F_c}\left[\|f' - f\|^2\right] = \epsilon$, compute the entropy of the input reconstruction
  - Measure how much information in each input pixel **cannot be recovered from** the feature

# Generality & coherency → enable comprehensive comparisons

| | Coherency | | Generality |
|---|---|---|---|
| | Layers | Nets | |
| Gradient-based | No | No | No |
| Perturbation-based | No | No | No |
| CAM-based | No | No | No |
| ours | Yes | Yes | Yes |

Comparisons of different methods in terms of generality and coherency. Our method provides coherent results across layers and networks.




Signal magnitudes of layerwise heatmaps

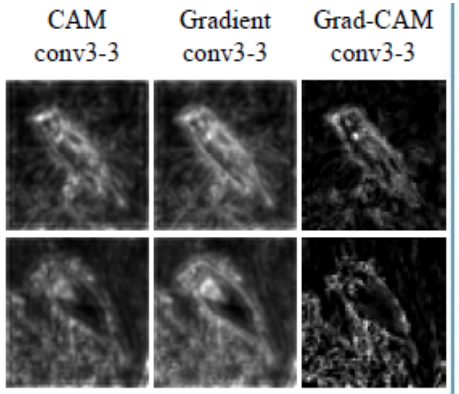
Not enable fair layerwise comparisons

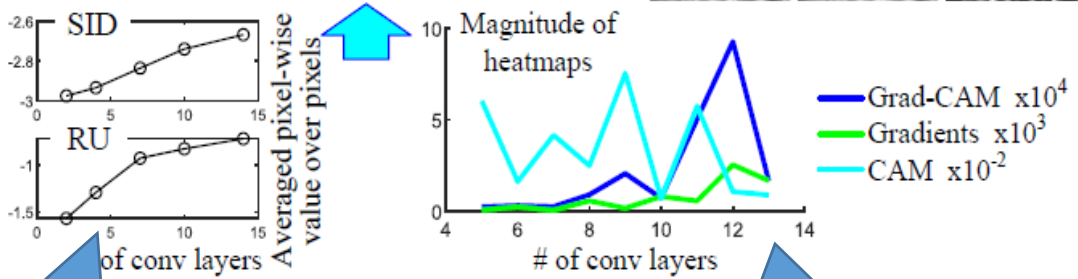
- Coherency: How to enable fair comparisons between layerwise attentions?
  - Previous methods of computing the pixel-wise attention / saliency / attribution / importance
    - Grad-CAM
    - Gradients-based
    - CAM
    - etc.

# Generality & coherency → enable comprehensive comparisons

| | Coherency | | Generality |
|---|---|---|---|
| | Layers | Nets | |
| Gradient-based | No | No | No |
| Perturbation-based | No | No | No |
| CAM-based | No | No | No |
| ours | Yes | Yes | Yes |

Comparisons of different methods in terms of generality and coherency. Our method provides coherent results across layers and networks.

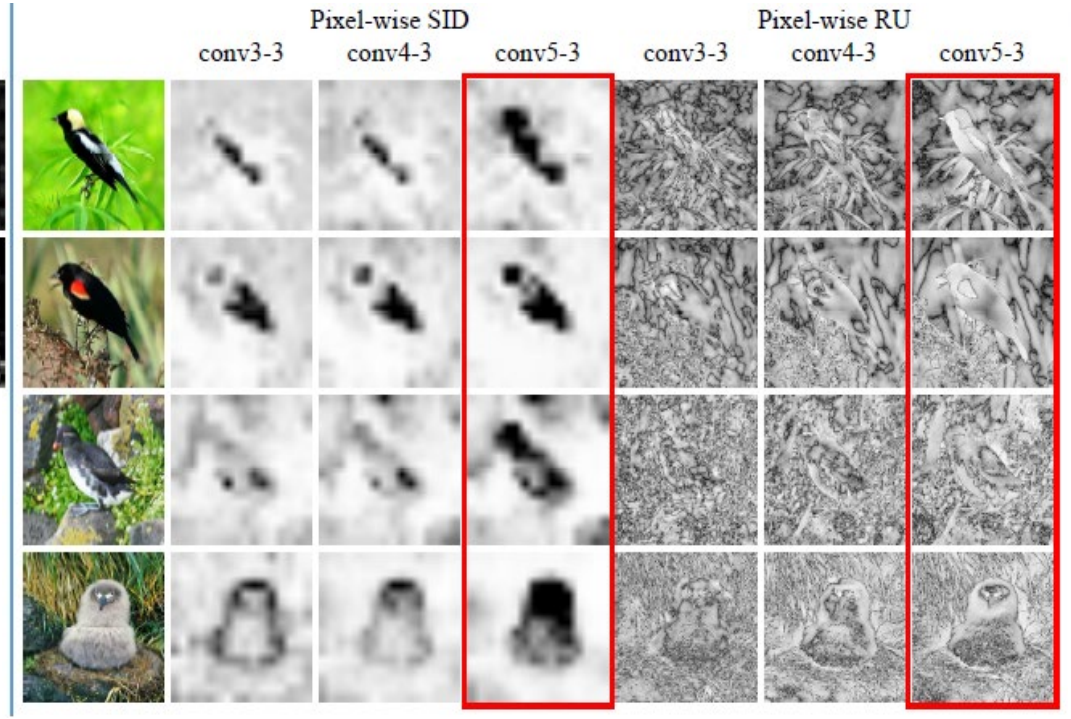



Signal magnitudes of layerwise heatmaps

Enable fair layerwise comparisons

Not enable fair layerwise comparisons

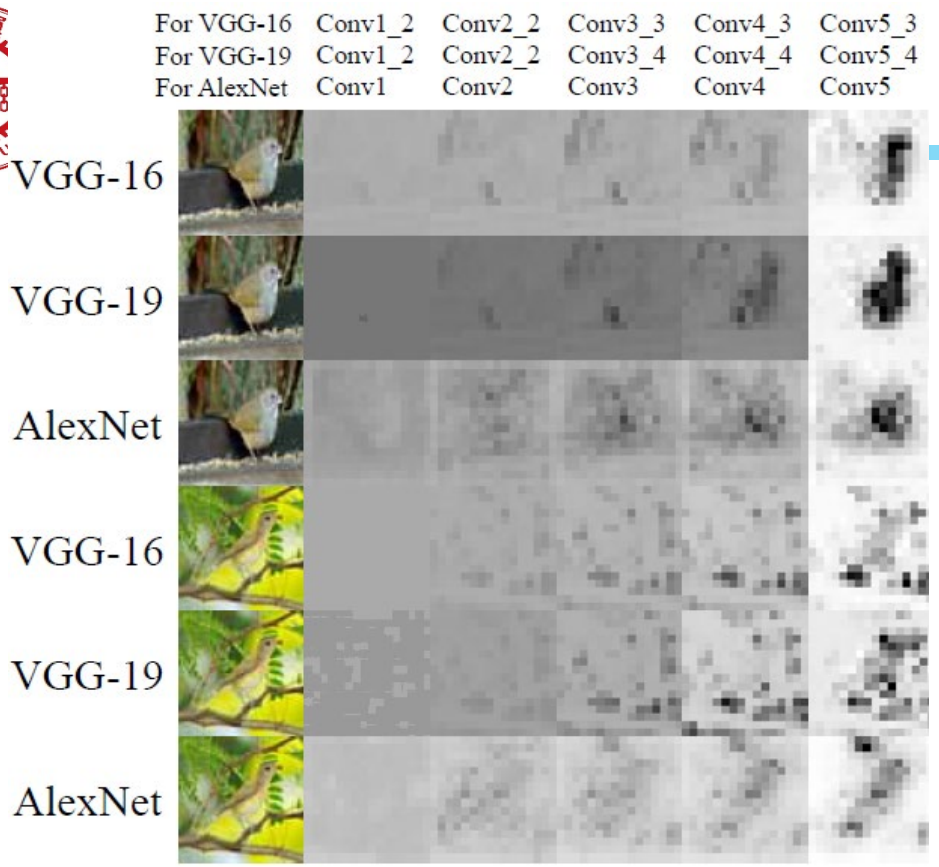



Pixel-wise entropies of input information

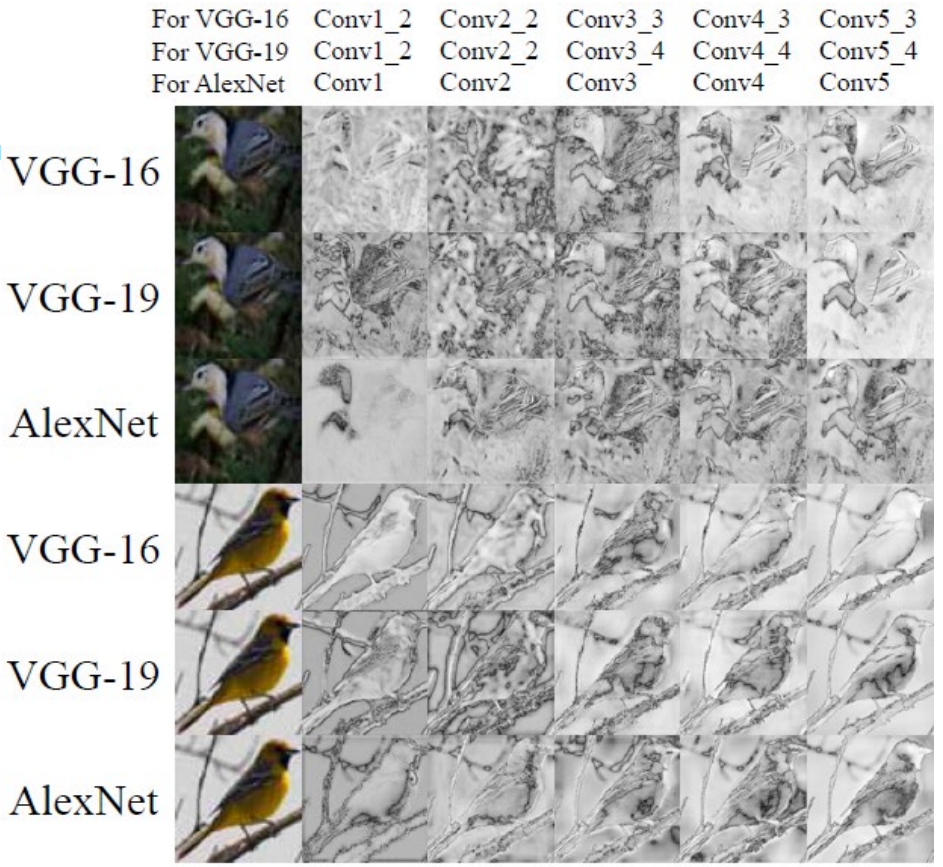
Pixel-wise entropies of input reconstruction

Comparing the discarding of the foreground information and the background information

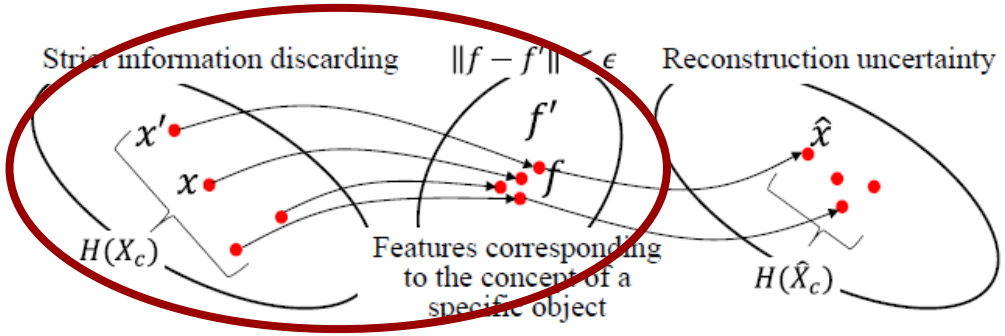
1. Enable fair layerwise comparisons within a specific DNN
2. Enable fair comparisons between specific layers of different DNNs
3. Enable fair comparisons between different DNNs learned using the same input but for different tasks

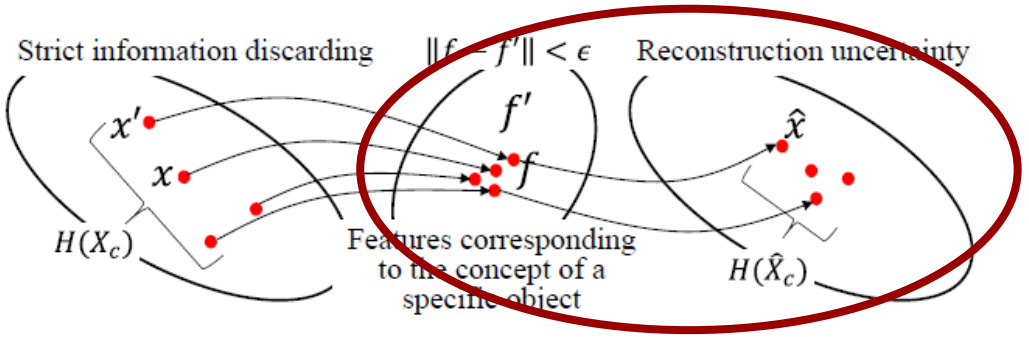



Discarding of input information:
gradually discard the background



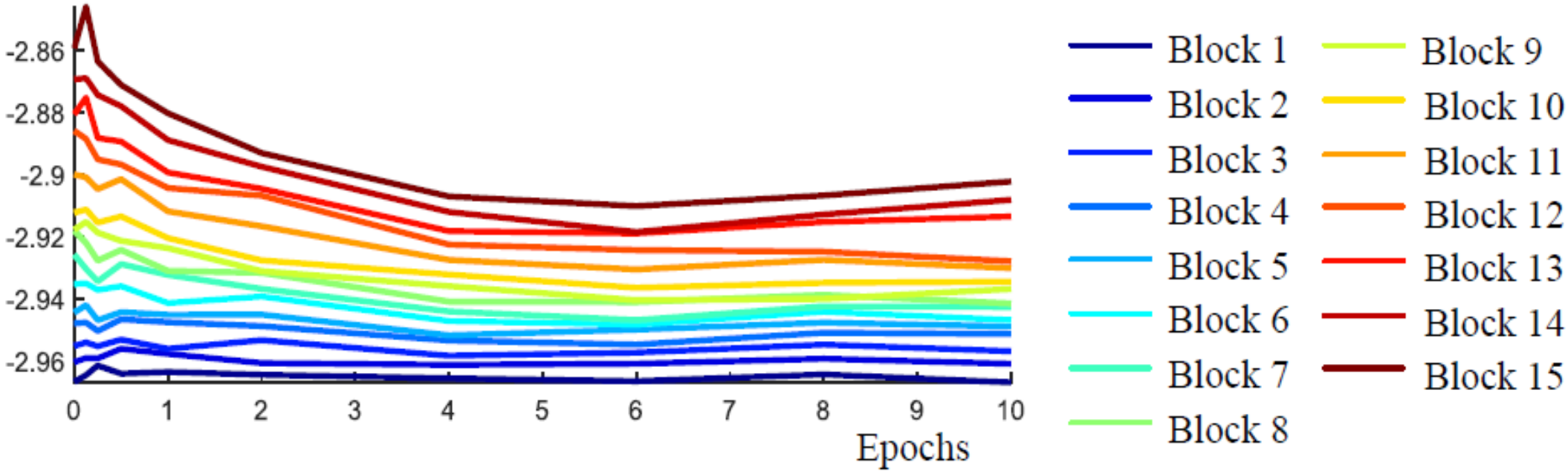

Discarding of input reconstruction:
edges are less discarded than colors



# Layerwise discarding of input information after different epochs

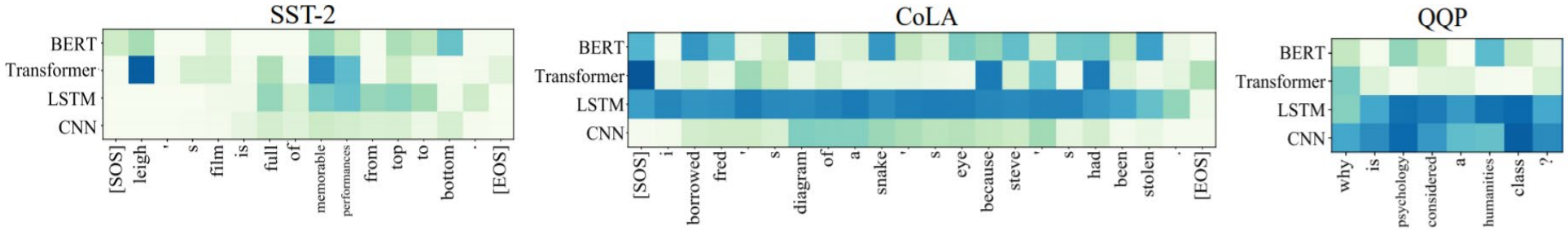
Entropy of the input information



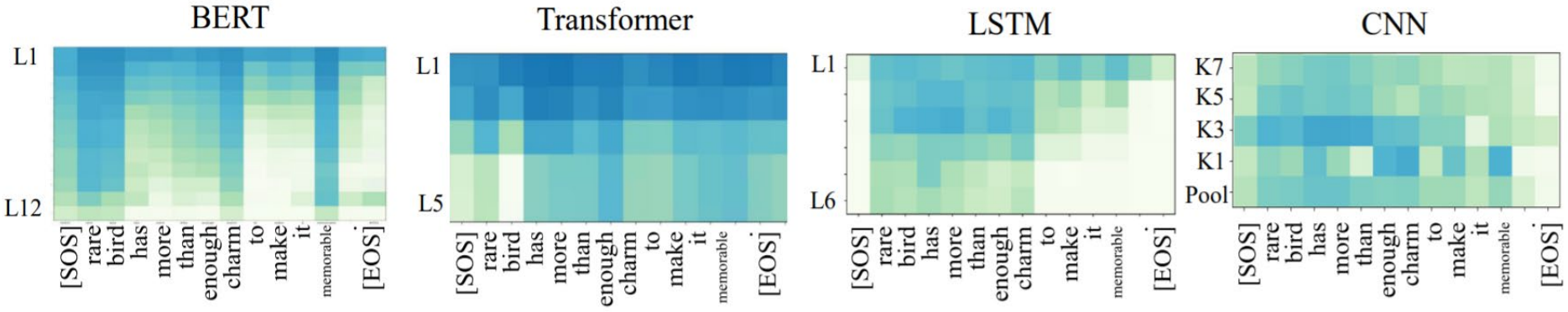

ResNet-32 learned on the CIFAR-10 dataset

# 分析比较自然语言处理中的神经网络

下图显示了不同神经网络的信息处理特点。CNN、LSTM往往使用连续的一段文字进行预测。而BERT、Transformer能够更精准地挑选出重要的单词。




下图显示了不同神经网络的信息遗忘的位置。CNN没有稳定的信息遗忘层，LSTM无法将不同单词的注意力区分开。而BERT，Transformer往往在神经网络前1/3的位置开始选择与目标任务相关的单词信息。



# Outline

- Explain semantic knowledge hidden in intermediate layers
  - How to represent CNNs using semantic graphical models
  - How to learn disentangled, interpretable features in middle layers
  - 在语义层面定量解释神经网络预测结果
- Explain representation capacity of deep neural networks
  - 对神经网络中层信息处理的量化分析与评测
  - **对神经网络特征表达可靠性的评测**

# 神经网络中层信息可靠性的量化分析与评测

## ◘研究成果

- Ruofan Liang, Tianlin Li, Longfei Li, **Quanshi Zhang**, “Knowledge Isomorphism between Neural Networks” in arXiv:1908.01581, 2019

## ◘量化神经网络中层特征可靠性，作为一般性理论工具

- 量化与评测神经网络中层特征表达的可靠性
  - 不需要额外的测试样本和任何额外标注
- 在没有额外监督信息的条件下，进一步提升神经网络性能
- 解释现有经典深度学习算法成功的原因
  - 知识蒸馏
  - 网络压缩
  - 神经网络对抗

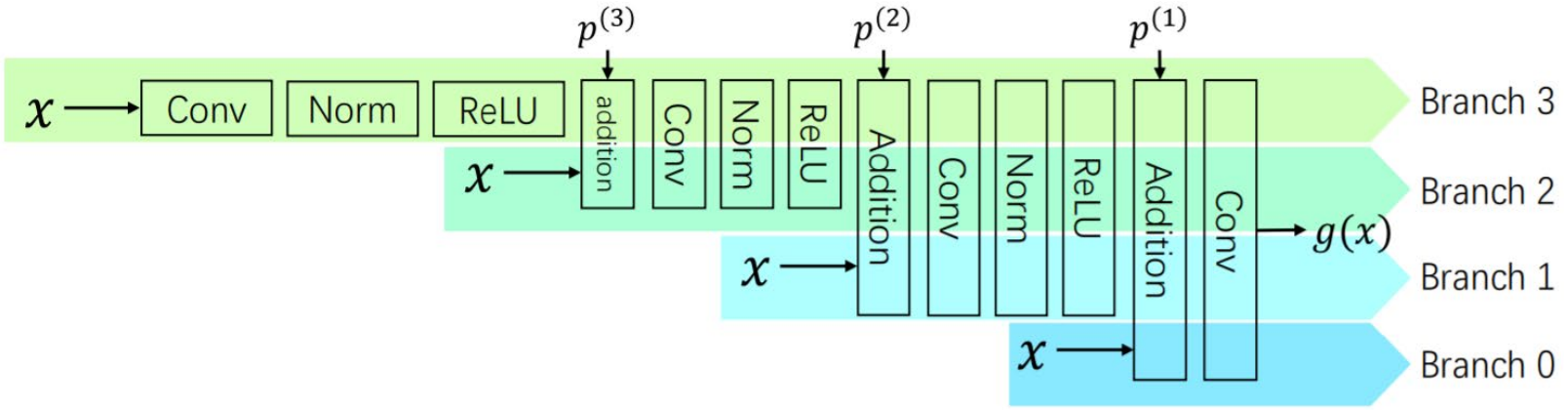
# 神经网络的知识同构

- 两个神经网络A和B是否建模了相同的知识
  - 特征相似：在某一特征空间上，特征距离比较近
  - 知识同构：可能具有完全不同的特征，但是表示相同的视觉概念。比如，利用相似的像素进行特征计算，并建模了相似的物体组成部分信息和纹理信息。
- 当利用不同神经网络去完成相同的任务
  - 可靠性：网络间同构的知识往往更为可靠
  - 分类性：可靠特征往往只建模前景信息，而不被背景出发（见中层信息量研究）
  - 全面性：神经网络是否全面的建模了全部特征，还是只用部分特征进行预测。
    - 不同构的特征分量——或某一神经网络所忽略的特征，或不可靠的特征。

# 知识同构的建模

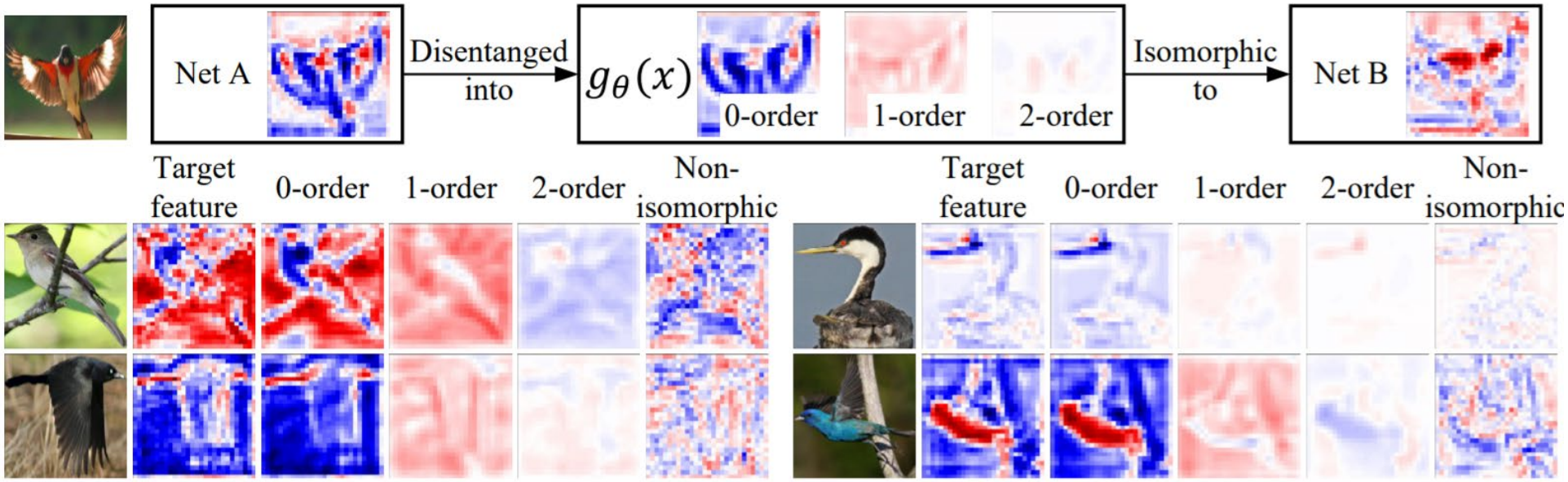
具体来说，$f_A$和$f_B$分别表示神经网络A与神经网络B的中层特征，当$f_A$可以通过线性变换得到$f_B$时，我们可认为$f_A$和$f_B$零阶同构；当$f_A$可以通过一次非线性变换得到$f_B$时，可认为$f_A$和$f_B$一阶同构；类似的，当$f_A$可以通过n次非线性变换得到$f_B$时，可认为$f_A$和$f_B$为n阶同构。

$$x^* = g_\theta(x) + x^\Delta, \qquad g_\theta(x) = x^{(0)} + x^{(1)} + \cdots + x^{(K)}$$



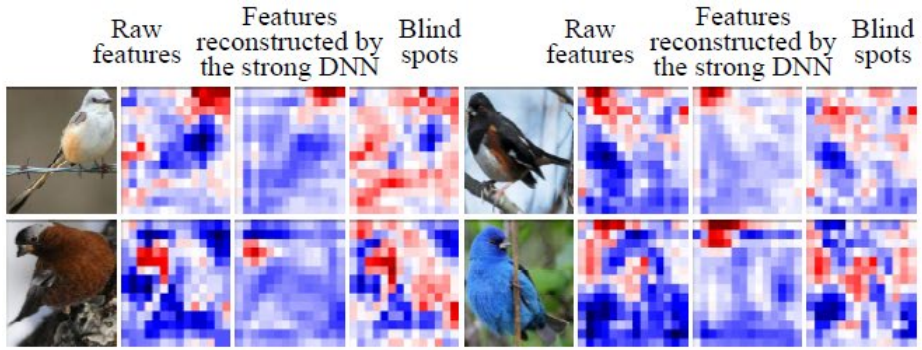
# 不同阶的知识同构

- 如下图所示，当对同一任务训练多个神经网络时，神经网络间彼此同构的特征分量往往代表更可靠的信息。从而，在不增加训练样本标注的前提下，我们可以利用神经网络间的同构信息进一步提升神经网络的分类精度。
- 下图显示了神经网络之间的0-2阶同构特征分量。低阶同构分量往往表示相对可靠的特征，而不同构分量则表示神经网络中的噪声信号。

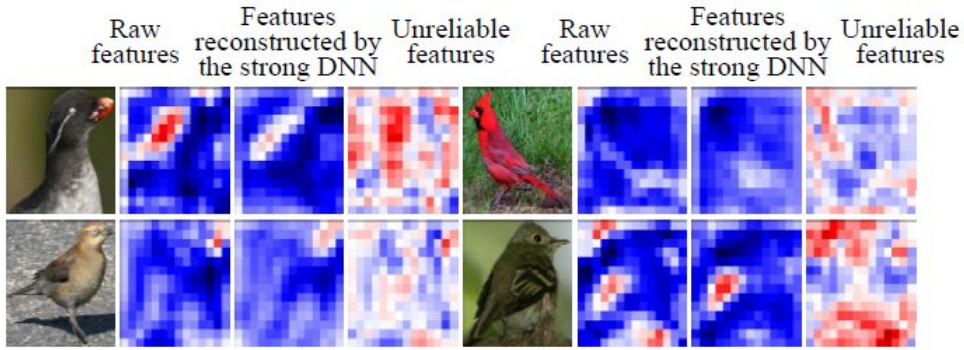


# 检测神经网络的不可靠特征和知识盲点

- 将一个深层高性能网络作为标准的知识表达，去分析诊断一个相对浅层的神经网络的不可靠特征和知识盲点（浅层神经网络有自己特定的应用价值，比如用在移动端）。
- 当利用浅层神经网络特征去重建深层神经网络特征时，深层神经网络中的不同构特征分量往往代表着浅层神经网络的知识盲点。
- 当利用深层神经网络特征去重建浅层神经网络特征时，浅层神经网络中的不同构特征分量往往代表着其中不可靠的特征分量。




(a) Blind spots of the weak DNN



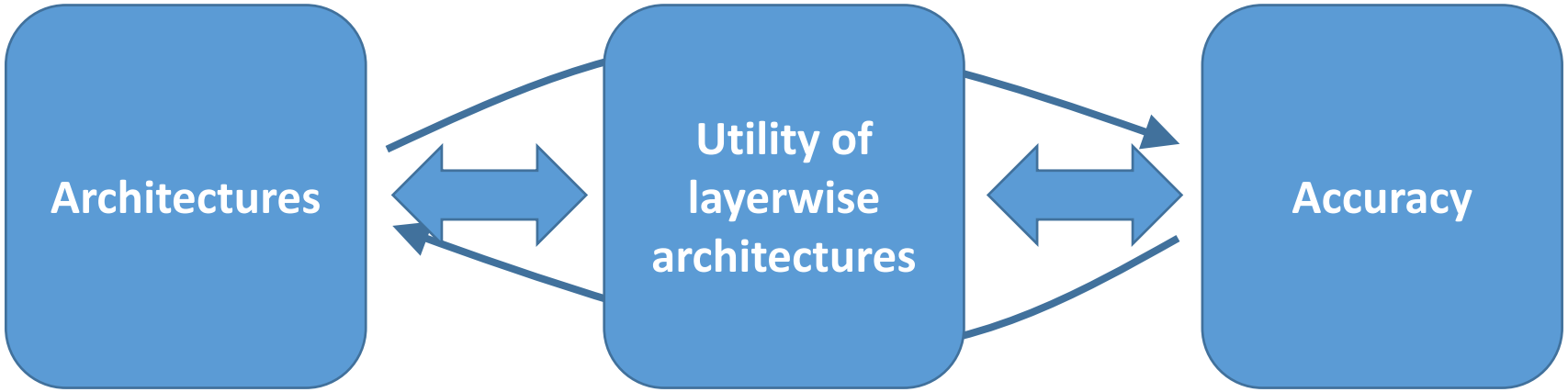

(a) Unreliable/noisy features of the weak DNN

# Outlines

- Understanding at the semantic level
  - Using a hierarchical explanatory graph to represent the semantic hierarchy in a CNN
  - End-to-end learning object-part features
  - Explaining network prediction semantically and quantitatively
- Functional utilities of layerwise architectures
  - In terms of discarding the foreground and background information
  - **In terms of the robustness of feature representations**
- Evaluation of attribution maps

# Utility of architectures

- People usually focus on the architecture and the performance

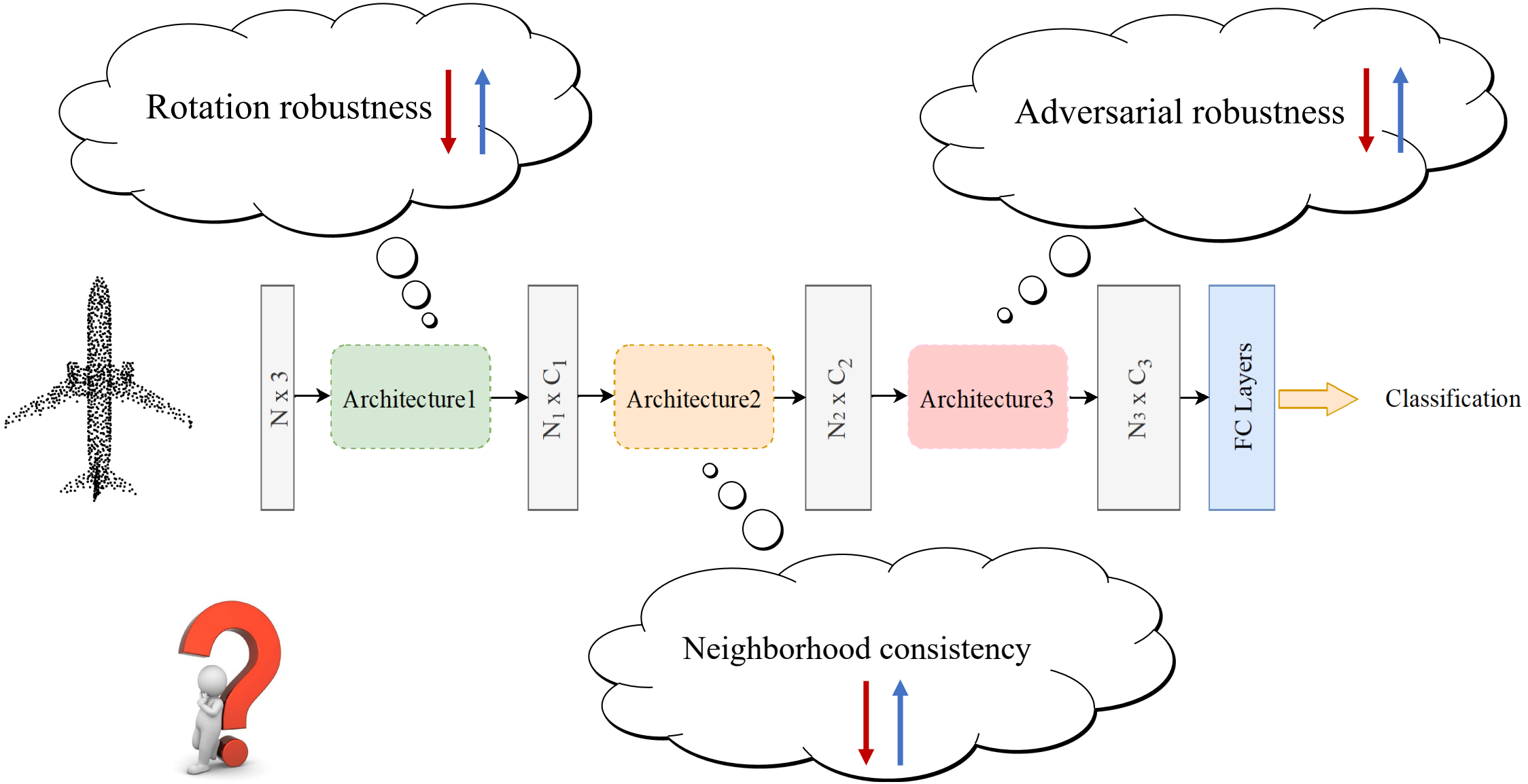


# Objective: analyzing the utility of layerwise architectures

- Different layerwise architectures → Different utilities ?

Utilities
- Rotation robustness
- Adversarial robustness
- Neighborhood consistency

# Objective: analyzing the utility of layerwise architectures



# Utility

## Rotation robustness

- Whether a DNN is supposed to use the same logic to recognize the same object when a point cloud has been rotated by a random angle

Essentially the same!

# Utility

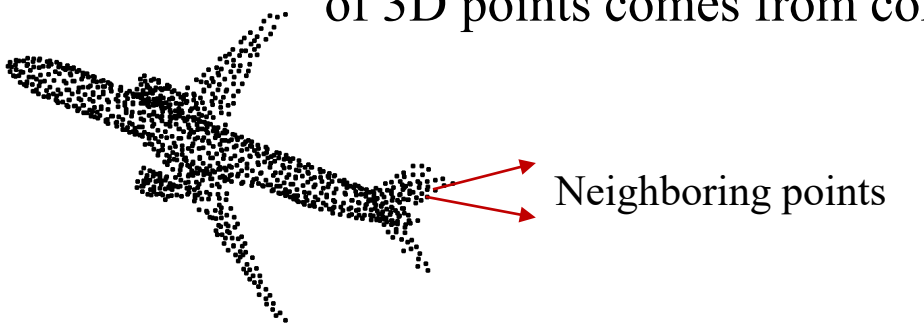
- Rotation robustness
- Adversarial robustness
  - A reliable DNN should be robust to adversarial attacks.

# Utility

- Rotation robustness
- Adversarial robustness
- Neighborhood consistency
  - Whether adjacent points in a point cloud have similar importance in the computation of an intermediate-layer feature?


Different contributions

Without informative colors, the information of 3D points comes from contextual shapes.




Same contributions???

# Hypotheses

Focus on four typical layerwise architectures in state-of-the-art DNNs.

## Hypothesis 1

- Architecture 1[1] uses the local density information to reweight features.
- Increase adversarial robustness

## Hypothesis 2

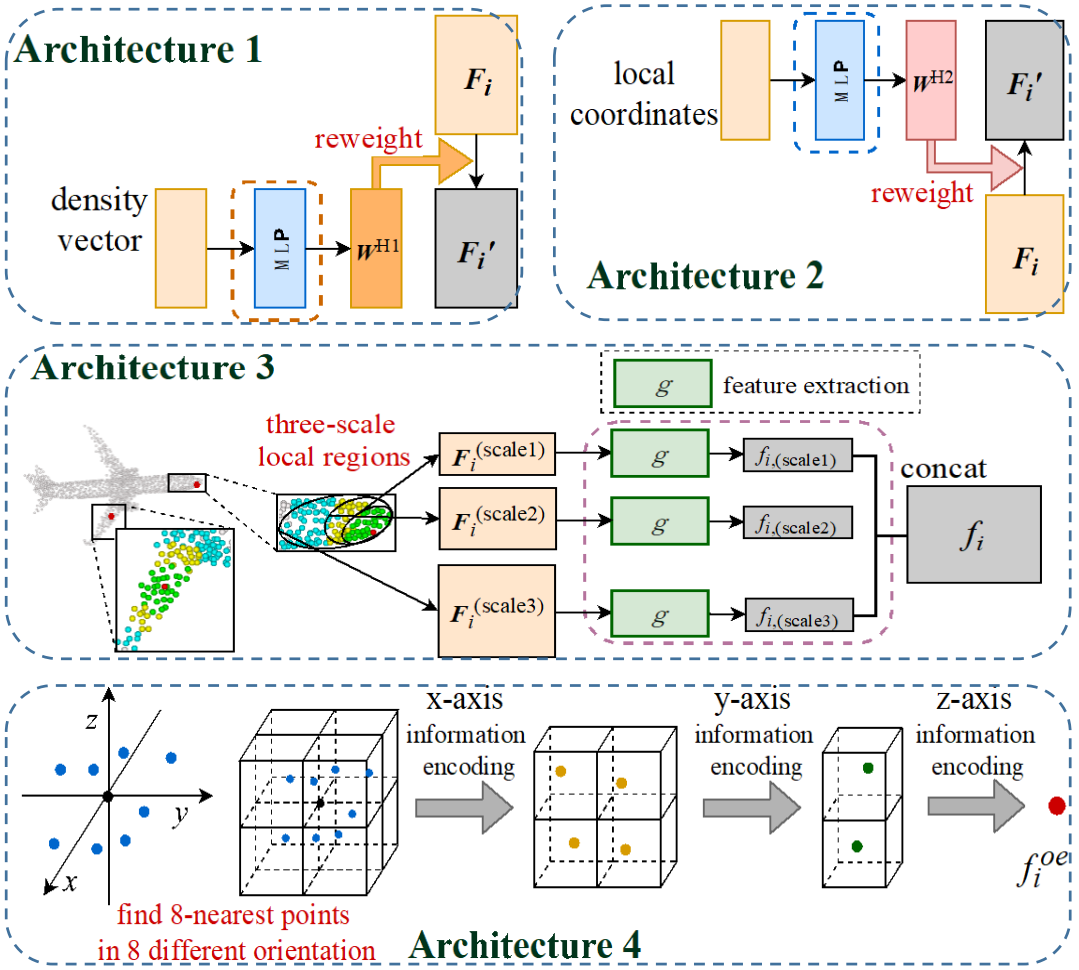
- Architecture 2[1] uses local 3D coordinates' information to reweight features.
- Increase rotation robustness

## Hypothesis 3

- Architecture 3[2,3] uses of multi-scale contextual information.
- Increase adversarial robustness and neighborhood consistency

## Hypothesis 4

- Architecture 4[4] encodes features using orientation information.
- Increase rotation robustness





[1] Wenxuan Wu. et al. "Pointconv: Deep convolutional networks on 3d point clouds." in CVPR 2019
[2] Charles R Qi et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." in NIPs 2017
[3] Xinhai Liu et al. "Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network."
[4] Mingyang Jiang et al. "Pointsift: A sift-like network module for 3d point cloud semantic segmentation."

# Comparative study

## ☐ Rotation robustness

Compare the rotation robustness between
1. DNNs with Architecture 2/4
2. DNNs replacing Architecture 2/4 with ordinary architectures

Architecture 2: using local point coordinates to reweight features

$$\mathbf{F}'_i = \mathbf{F}_i(\boldsymbol{W}^{\mathrm{H2}})^\top, \qquad \text{where} \quad \boldsymbol{W}^{\mathrm{H2}} = MLP(\{x_j | j \in \mathbf{N}(i)\})$$

Architecture 4: encoding features from the x, y, z axes, respectively

$$f_i^{\mathrm{oe}} = Conv^{\mathrm{oe}}(\mathbf{F}_i^{\mathrm{oe}})$$

| Architecture | Model | ModelNet40 | | | ShapeNet | | | 3D MNIST | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | w/ | w/o | Δ | w/ | w/o | Δ | w/ | w/o | Δ |
| Architecture 2 | PointConv | 3.918 | 3.954 | **0.036** | 4.250 | 2.703 | -1.547 | 5.140 | 6.221 | **1.081** |
| | PointNet++ | 2.658 | 5.000 | **2.342** | 4.186 | 6.709 | **2.523** | 5.256 | 6.754 | **1.498** |
| | Point2Sequence | 4.020 | 5.786 | **1.766** | 2.821 | 5.222 | **2.401** | 4.590 | 7.410 | **2.820** |
| Architecture 4 | PointSIFT | 4.747 | 7.090 | **2.343** | 4.598 | 5.118 | **0.520** | 7.851 | 6.154 | -1.697 |
| | PointNet++ | 5.505 | 5.000 | -0.505 | 6.152 | 6.709 | **0.557** | 5.298 | 6.754 | **1.456** |
| | Point2Sequence | 2.917 | 5.786 | **2.869** | 2.909 | 5.222 | **2.313** | 5.942 | 7.410 | **1.468** |

# Comparative study

## Adversarial robustness

Compare the adversarial robustness between
1. DNNs with Architecture 1/3
2. DNNs replacing Architecture 1/3 with ordinary architectures

Architecture 1: using local point density to reweight features

$$\mathbf{F}'_i = \mathbf{F}_i \; diag[\boldsymbol{W}^{\mathrm{H1}}], \qquad \text{where} \quad \boldsymbol{W}^{\mathrm{H1}} = MLP(density(\mathbf{N}(i)))$$

Architecture 3: using multi-scale features

$$f_i^{\mathrm{upper}} = concat \left\{ \begin{array}{c} f_{i,(\mathrm{scale}=K_1)}^{\mathrm{upper}} \\ f_{i,(\mathrm{scale}=K_2)}^{\mathrm{upper}} \\ \vdots \\ f_{i,(\mathrm{scale}=K_T)}^{\mathrm{upper}} \end{array} \right\}, \qquad \text{where } f_{i,(\mathrm{scale}=K_t)}^{\mathrm{upper}} = g(\mathbf{F}_i^{\mathrm{scale}=K_t})$$

| Architecture | Model | ModelNet40 | | | ShapeNet | | | 3D MNIST | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | w/ | w/o | Δ | w/ | w/o | Δ | w/ | w/o | Δ |
| Architecture 1 | PointConv | 2.878 | 2.629 | **0.249** | 2.407 | 2.271 | **0.136** | 2.737 | 2.530 | **0.207** |
| | PointNet++ | 2.519 | 2.504 | **0.015** | 2.496 | 2.437 | **0.059** | 2.427 | 2.352 | **0.075** |
| | Point2Sequence | 2.544 | 2.526 | **0.018** | 2.500 | 2.520 | -0.020 | 2.475 | 2.468 | **0.007** |
| Architecture 3 | PointNet++ | 3.010 | 2.504 | **0.506** | 2.987 | 2.437 | **0.550** | 2.604 | 2.352 | **0.252** |
| | Point2Sequence (4 scales *vs.* 3 scales) | 2.526 | 2.521 | **0.005** | 2.520 | 2.514 | **0.006** | 2.468 | 2.479 | -0.011 |
| | Point2Sequence (4 scales *vs.* 2 scales) | | 2.513 | **0.013** | | 2.488 | **0.032** | | 2.460 | **0.008** |

# Comparative study

## ☐ Neighborhood consistency

Compare the neighborhood consistency between
1. DNNs using multi-scale features
2. DNNs without using multi-scale features

Architecture 3: using multi-scale features

$$f_i^{\text{upper}} = concat\left\{\begin{array}{l} f_{i,(\text{scale}=K_1)}^{\text{upper}} \\ f_{i,(\text{scale}=K_2)}^{\text{upper}} \\ \vdots \\ f_{i,(\text{scale}=K_T)}^{\text{upper}} \end{array}\right\}, \quad \text{where } f_{i,(\text{scale}=K_t)}^{\text{upper}} = g(\mathbf{F}_i^{\text{scale}=K_t})$$

| Model | ModelNet40 | | | ShapeNet | | | 3D MNIST | | |
|---|---|---|---|---|---|---|---|---|---|
| | w/ | w/o | Δ | w/ | w/o | Δ | w/ | w/o | Δ |
| PointNet++ | 3.149 | 3.451 | **0.302** | 3.321 | 3.346 | **0.025** | 3.496 | 3.519 | **0.023** |
| Point2Sequence (4 scales *vs.* 3 scales) | 3.655 | 4.182 | **0.527** | 3.091 | 3.179 | **0.088** | 3.226 | 3.411 | **0.185** |
| Point2Sequence (4 scales *vs.* 2 scales) | | 4.253 | **0.598** | | 3.199 | **0.108** | | 3.537 | **0.311** |

# 谢谢