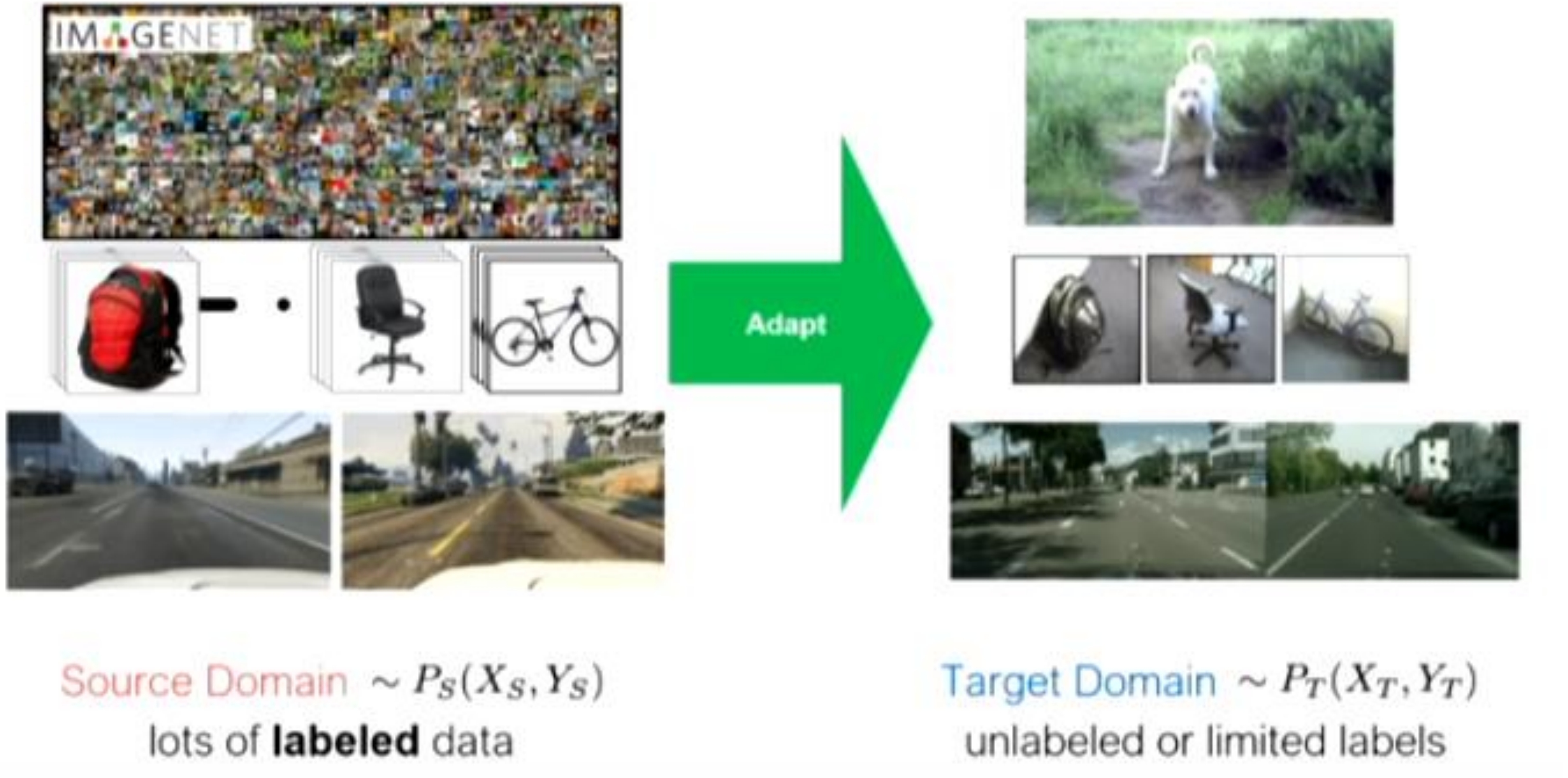
Towards Adaptive and Explainable Artificial Intelligence

Summary of CS294-131 Fa18 09/04/18 Talk

Trevor Darrell

Part I: Domain Adaptation

- Train on Source, Test on Target



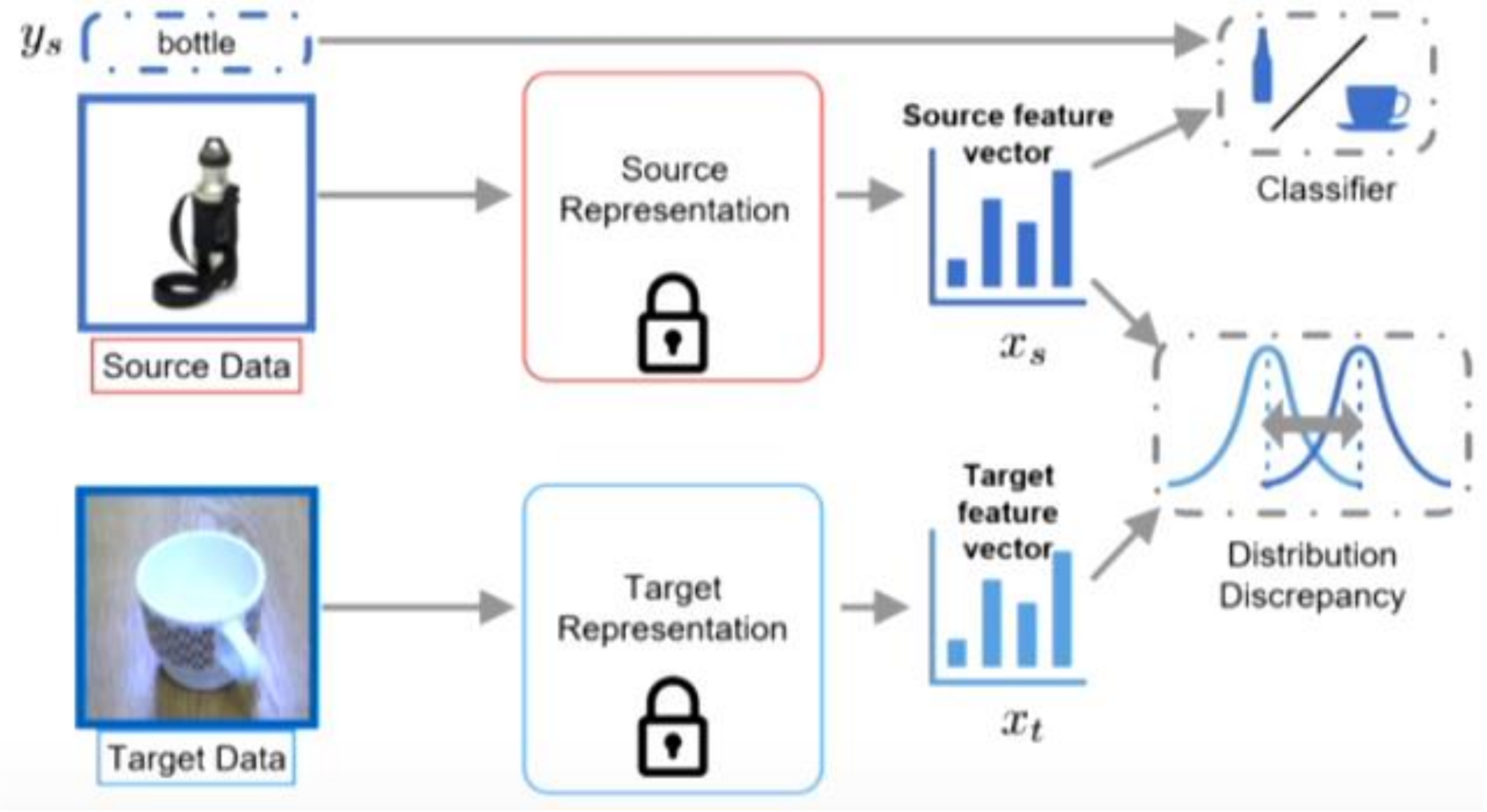
Domain Adaptation Paradigms

- **Feature Augmentation**-add training data transformed based on knowledge of domain.

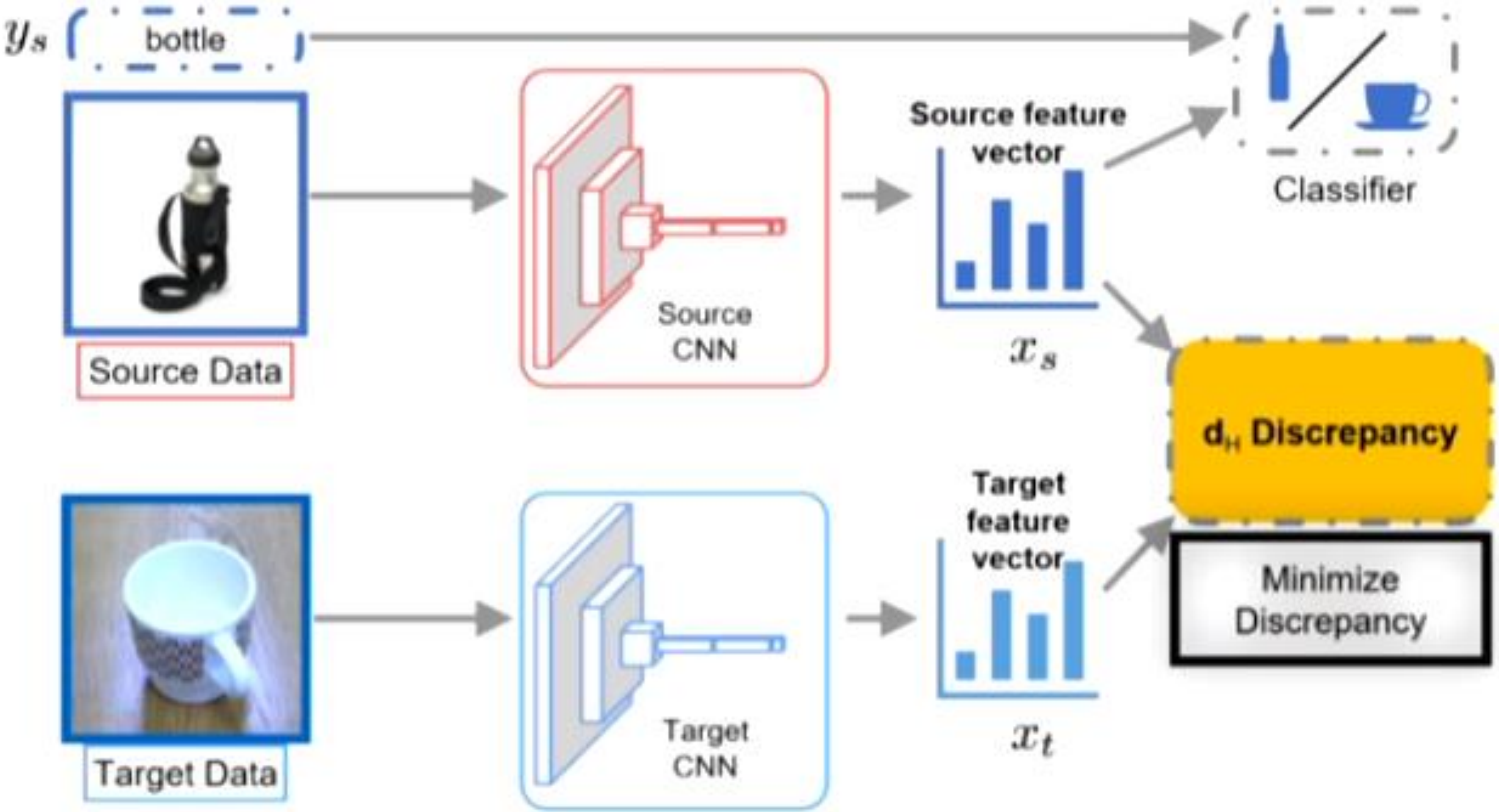
- **Bootstrapping**, a.k.a. “self-ensembling”, …- take high confidence predictions of source-only model and add them to training data, iterate…

- **Distribution alignment or transformation**: domain adversarial learning / domain confusion using GAN-like models…

Classic Domain Adaptation

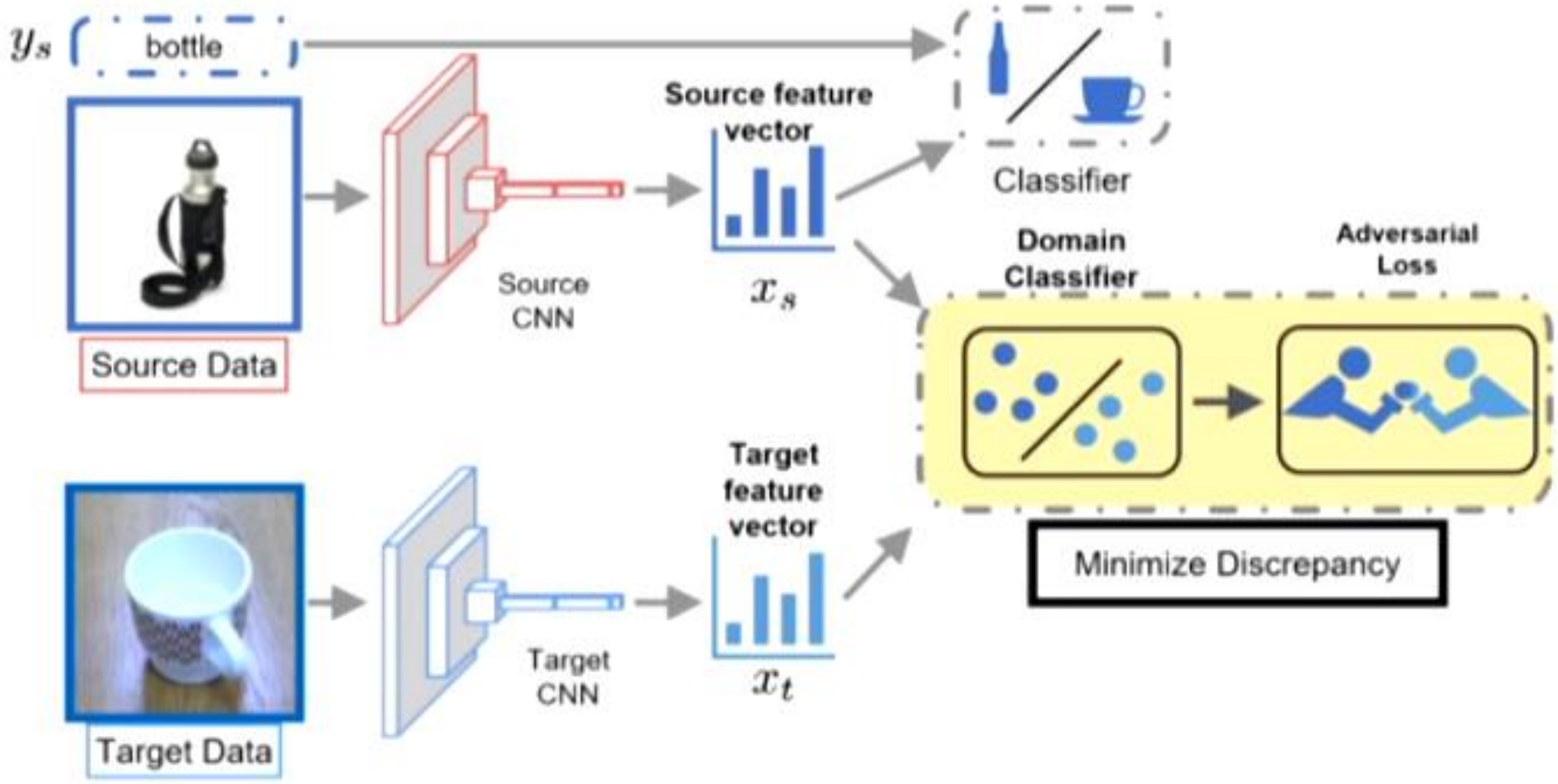


Deep Domain Adaptation



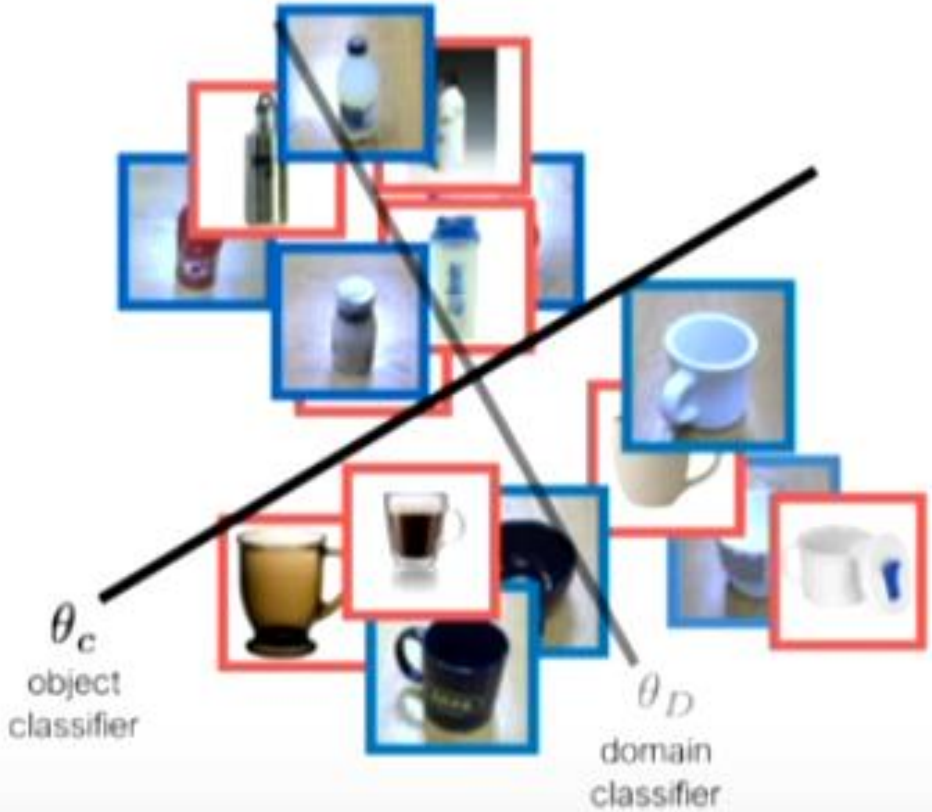
Tzeng, **Hoffman**, Saenko, Darrell. *CVPR* 2017.

Adversarial Domain Adaptation



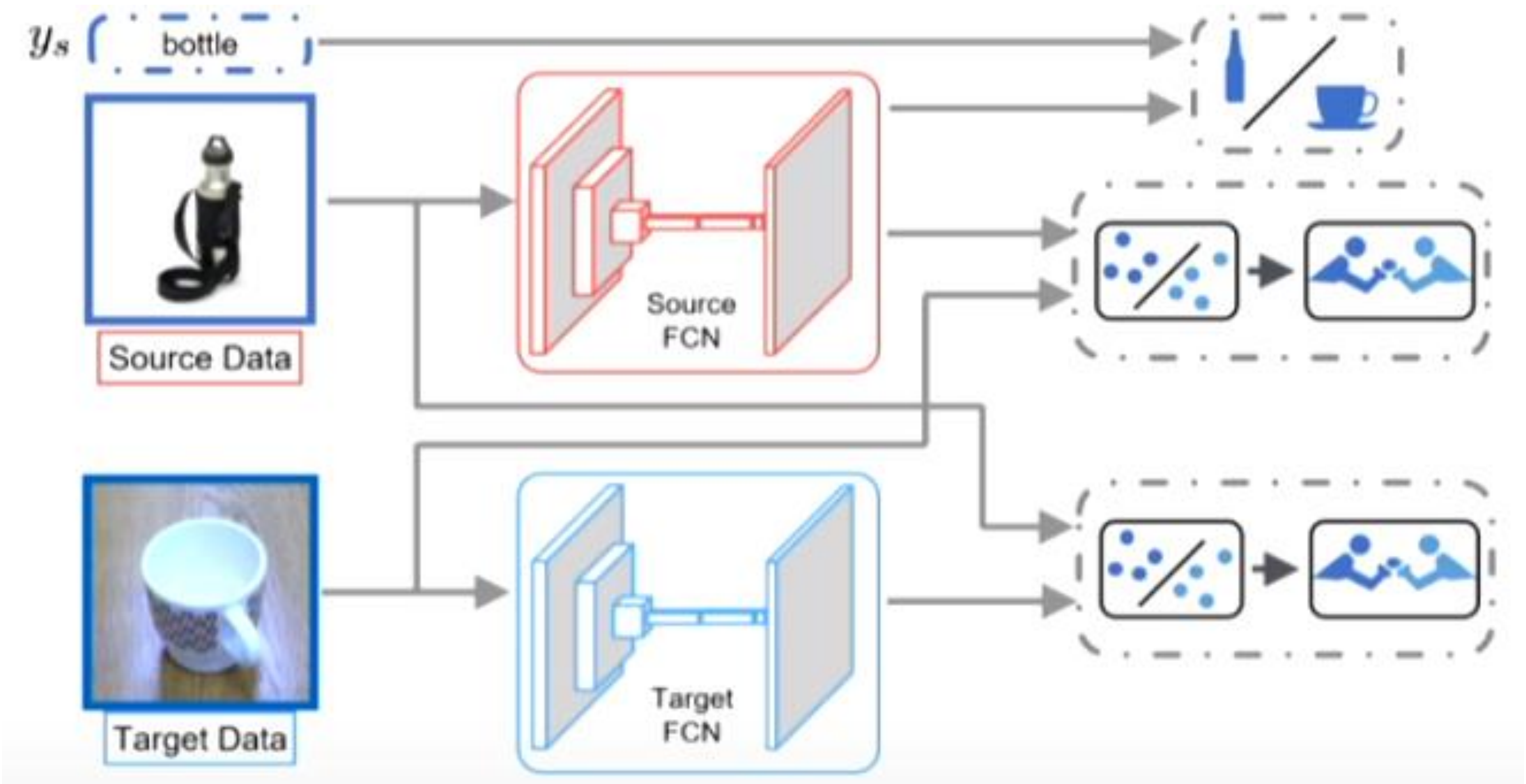
Tzeng, **Hoffman**, Saenko, Darrell. *CVPR* 2017.

$$\min_{\theta_D} \mathcal{L}_{\text{dom}}(\mathbf{X}_s, \mathbf{X}_t, \theta_R; \theta_D) \qquad \min_{\theta_R} \mathcal{L}_{\text{rep}}(\mathbf{X}_s, \mathbf{X}_t, \theta_D; \theta_R) \qquad \min_{\theta_C, \theta_R} \mathcal{L}_{\text{cls}}(\mathbf{X}_s, \mathbf{Y}_s; \theta_C, \theta_R)$$

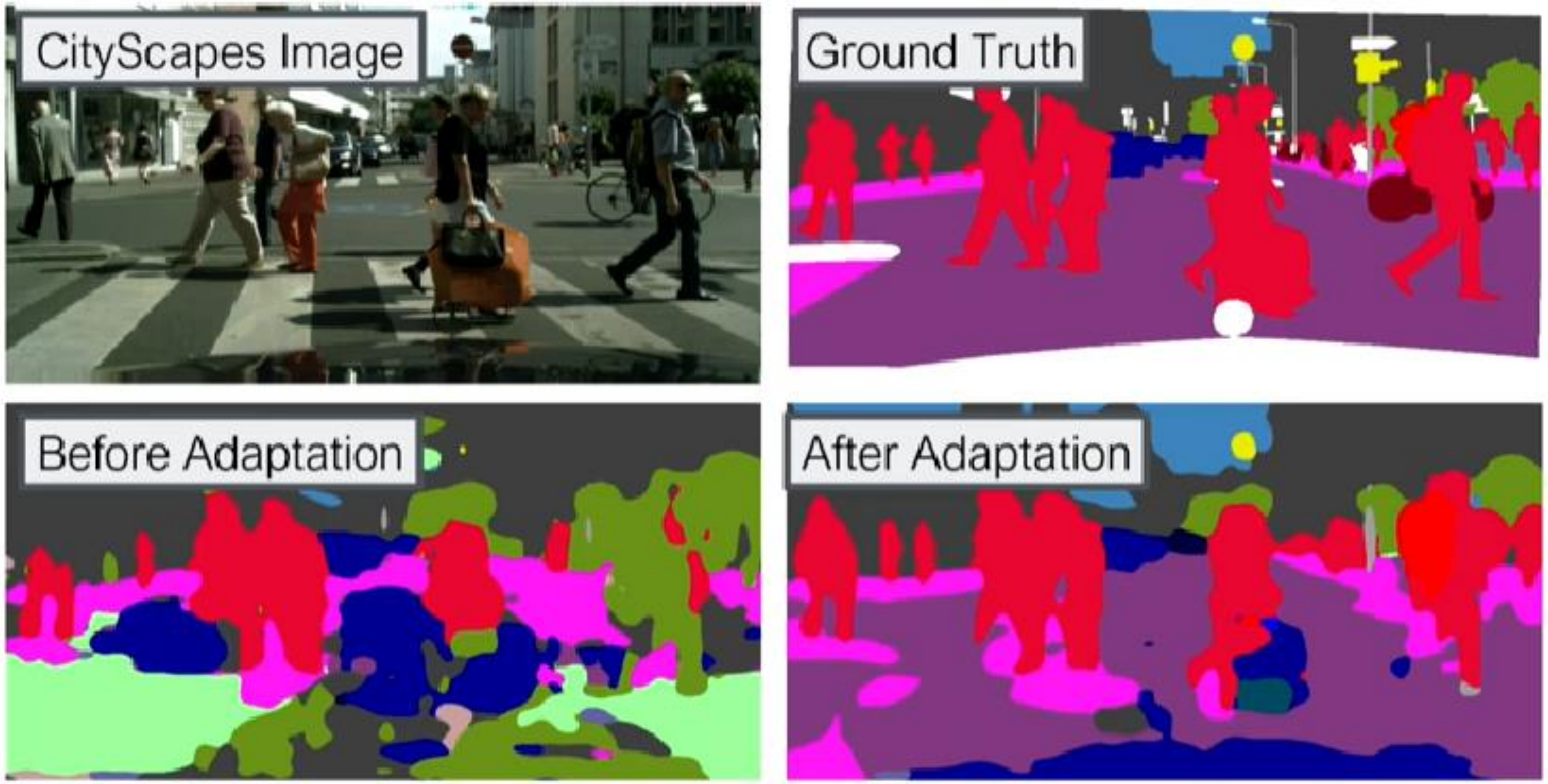


Tzeng, **Hoffman**, Saenko, Darrell. *CVPR* 2017.

Pixel level+feature level

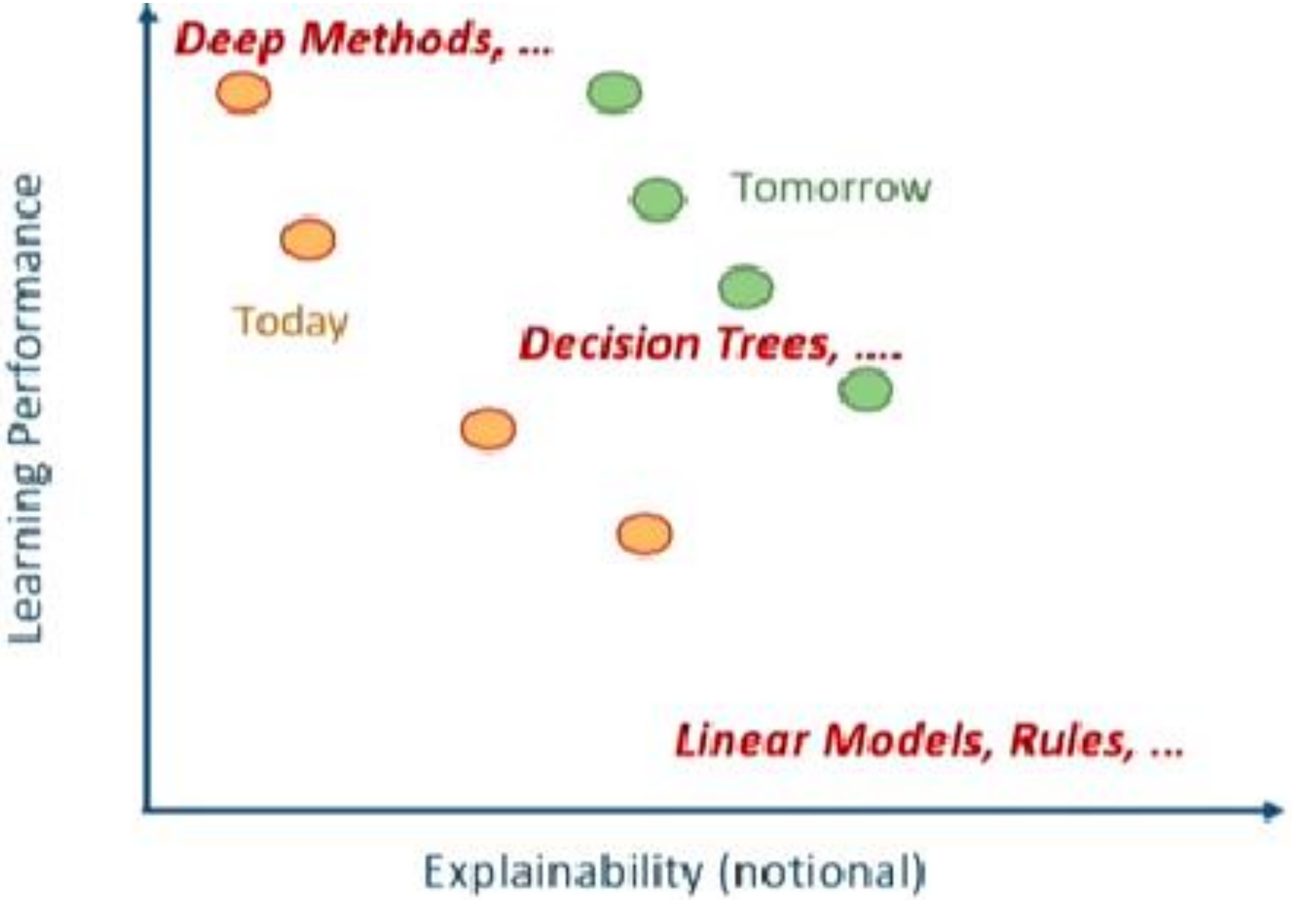


Results: Train on GTA, test on Cityscapes



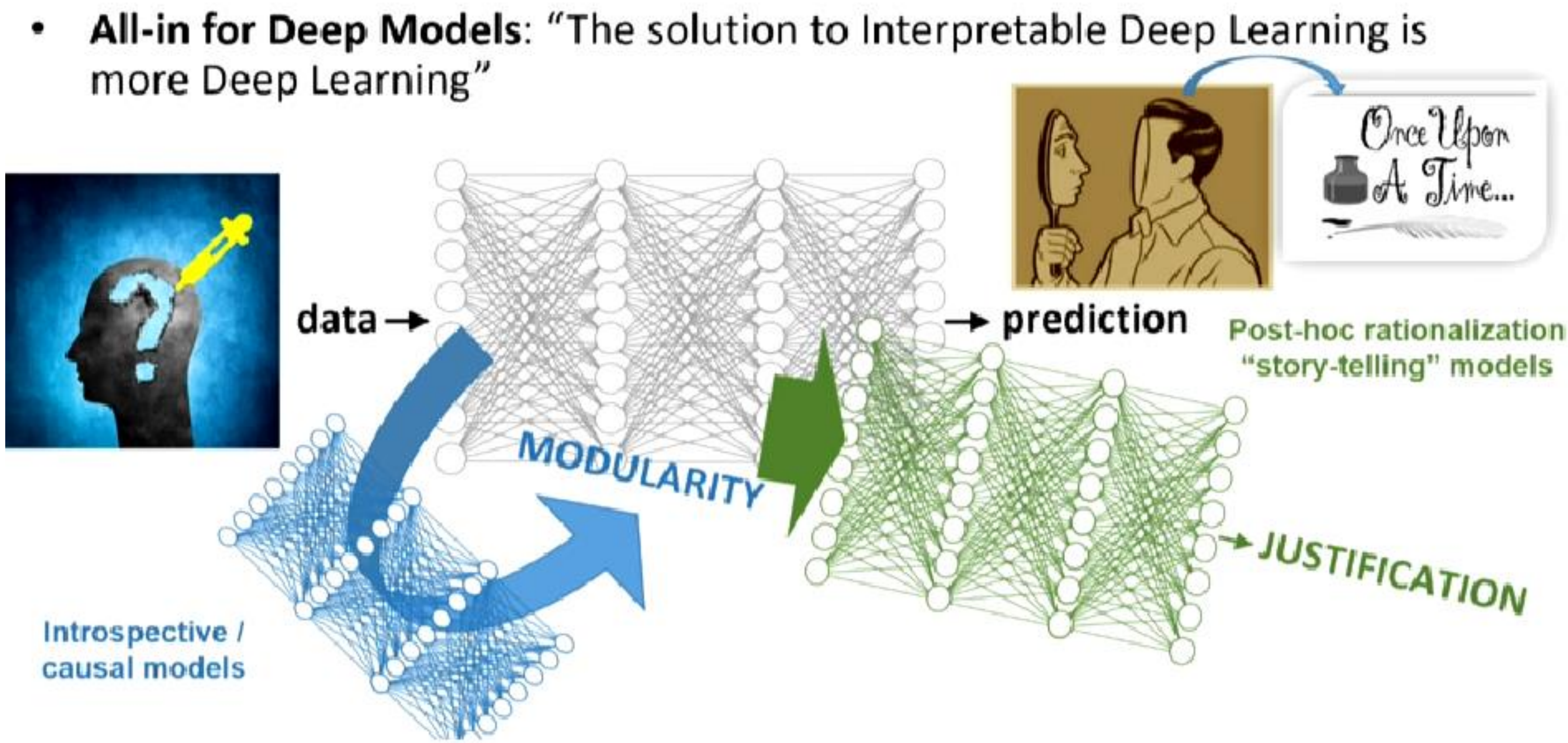
Hoffman, Tzeng, Park, Zhu, Isola, Saenko, Efros, Darrell, arXiv 2017.

Part II: Explainable AI



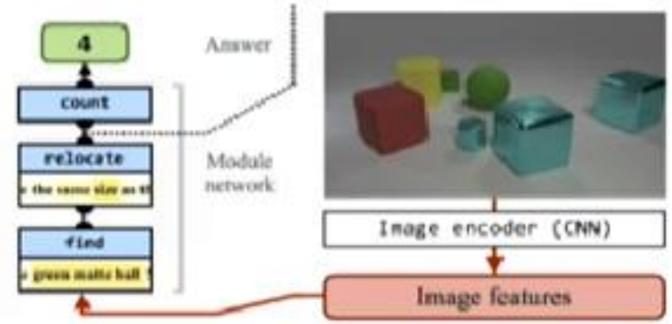
(source: DARPA XAI)

- Modularity
- Justification

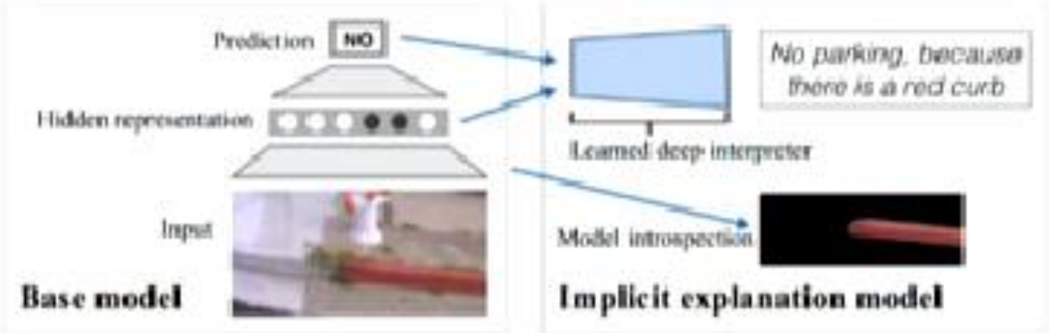


Deep Explanation Models

- Explicit / Introspective models: interpretable interval visualization

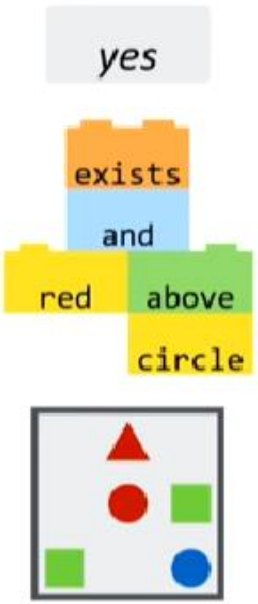


- Implicit /Justification models: post-hoc rationalization

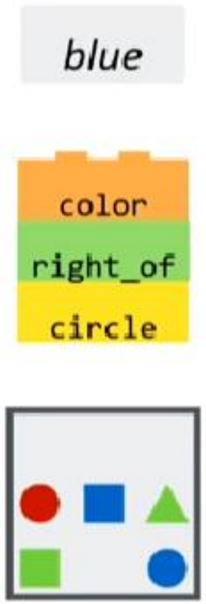


Modularity

- Main reading: Learning to Reason: End-to-End Module Networks for Visual Question Answering

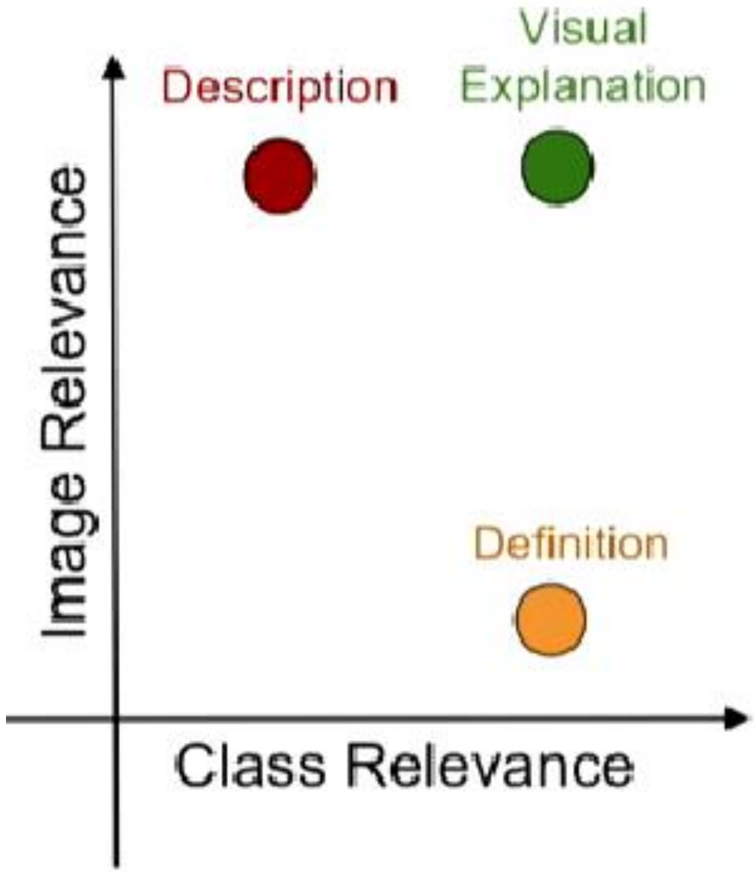


Is there a red shape above a circle?



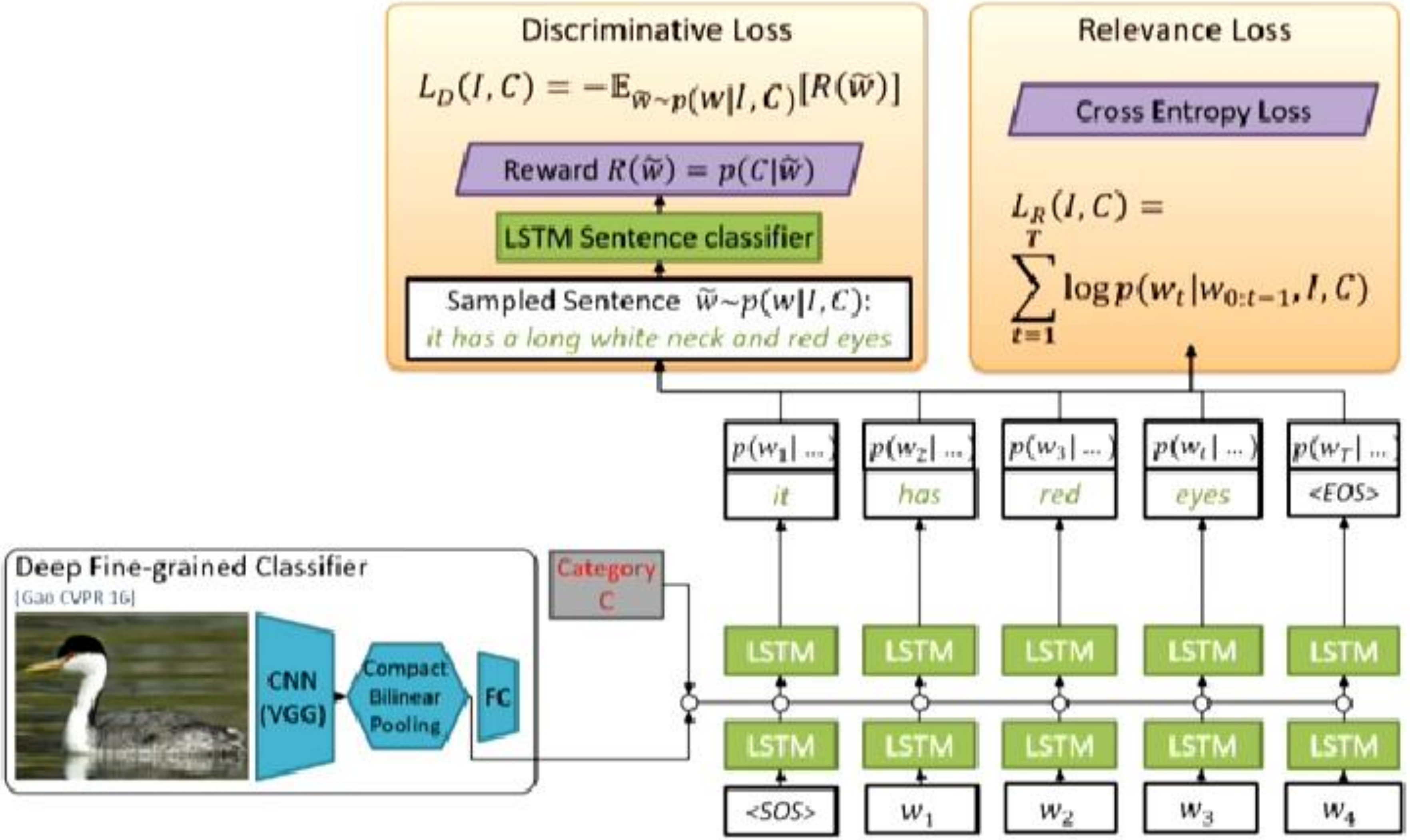
What color is the shape right of a circle?

Justification

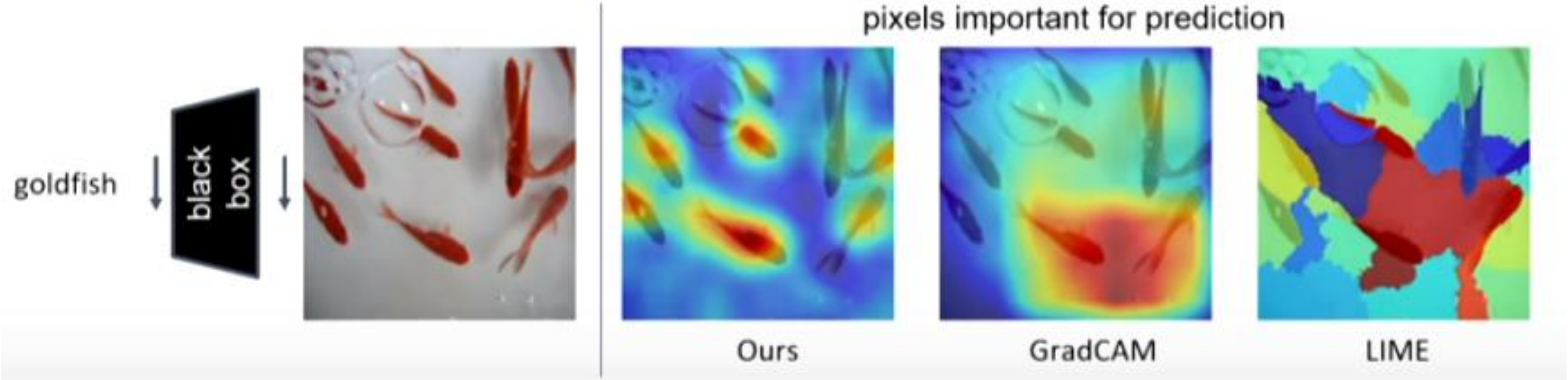


A large bird with a **white neck** in the **water**.
Western Grebe has **yellow pointy beak**.
This is a *Western Grebe* because it has a **long white neck**, **pointy yellow beak** and **red eye**.

Visual explanation-training time



‘High-fidelity’ explanation

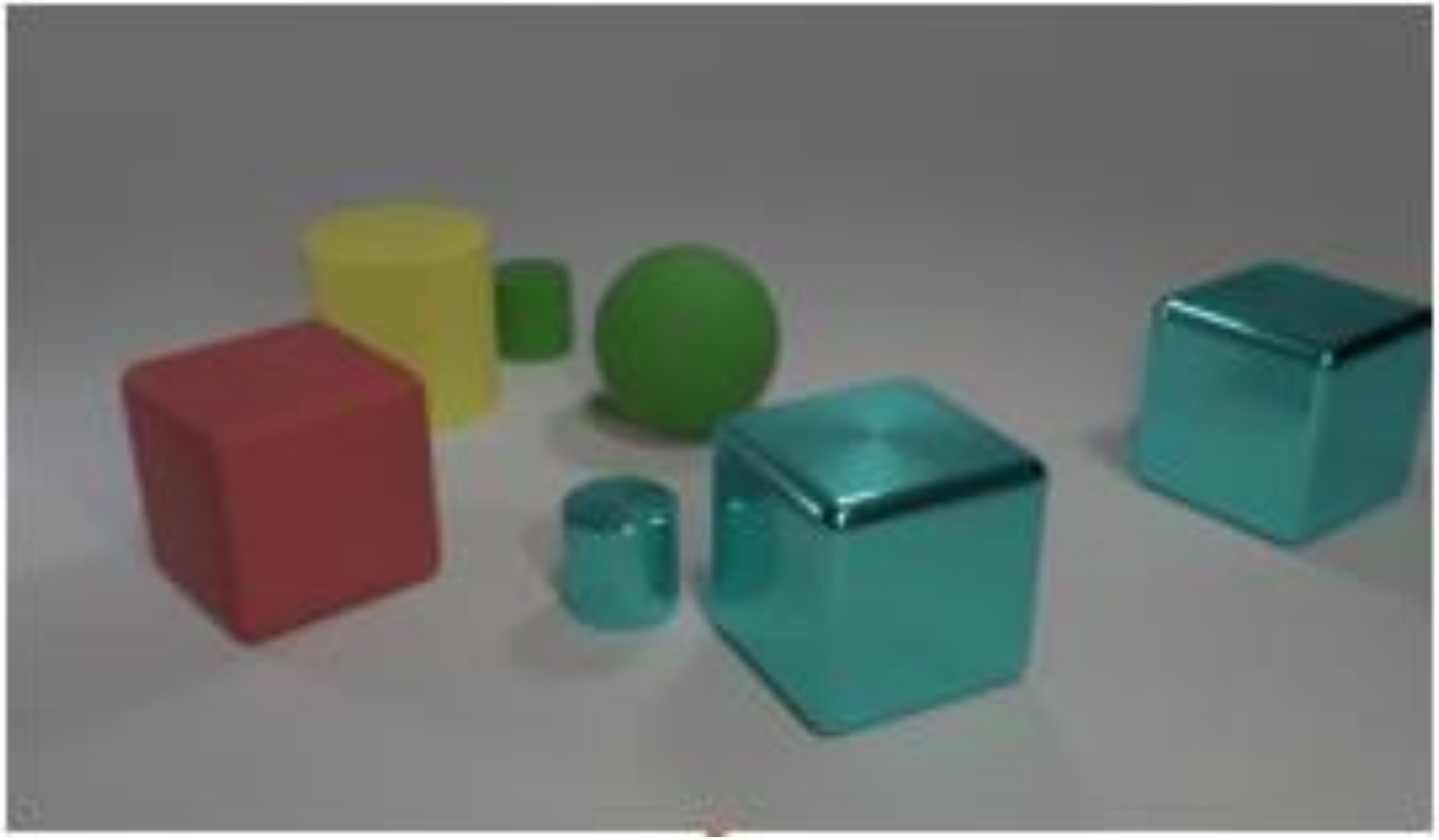


Sumary of Papers

Task Definition

- Visual Question Answer:
  - A VQA system takes an image as input and a free-form, open-ended, natural-language question about the image and produces a natural-language answer as the output.
- Involved problems:
  - Natural Language Process
  - Object recognition
  - Multi-step reasoning
  - Explain-ability

How many other things are of the same size as the green matte ball?



Learning to Compose Neural Networks for Question Answering

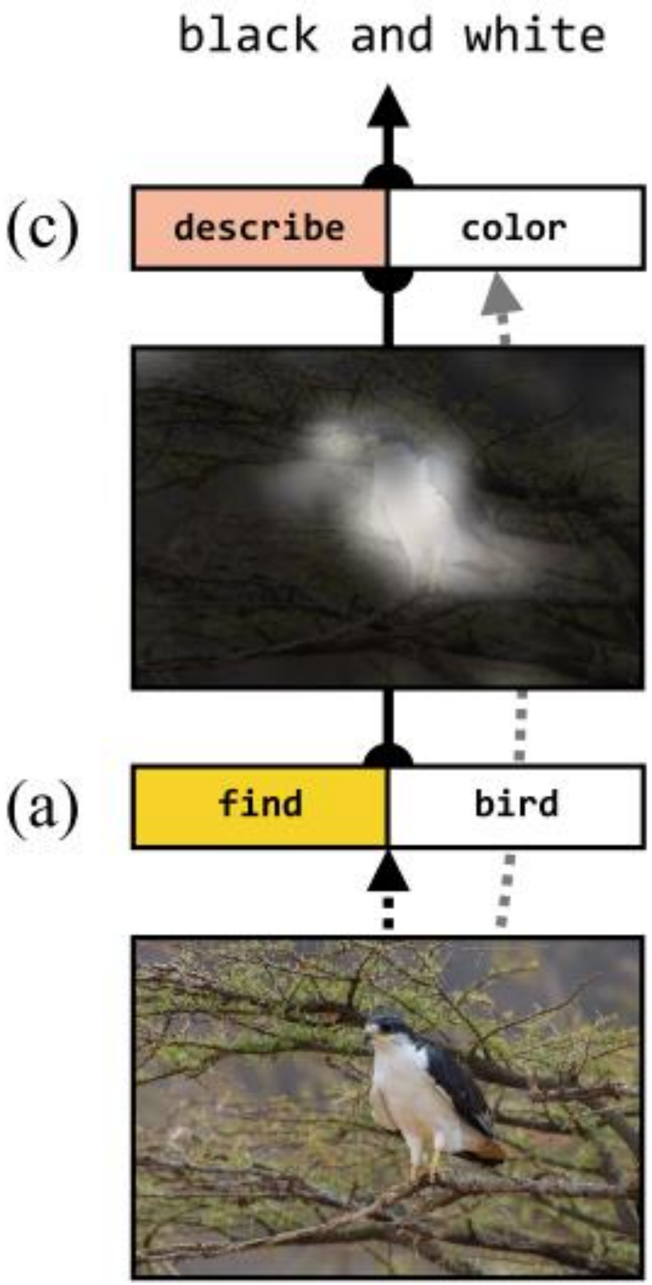
Jacob Andreas, Marcus Rohrbach, Trevor Darrell, Dan Klein

Motivation:

- All of previous models assume that a fixed computation can be performed on the image and question to compute the answer, rather than adapting the structure of the computation to the question.

- In this paper, we present a model for learning to select such structures from a set of automatically generated candidates. We call this model a dynamic neural module network.

What color is the bird?



Model

- The DNMN model is built around two distributions: a **layout model** $p(z|x;\theta_l)$which chooses a layout for a sentence, and a **execution model** $p_z\ (y|w;\theta_e)$ which applies the network specified by z to w.

- Execution model:
  - Given a layout **z**, optimize the neural modules in layout **z** by:

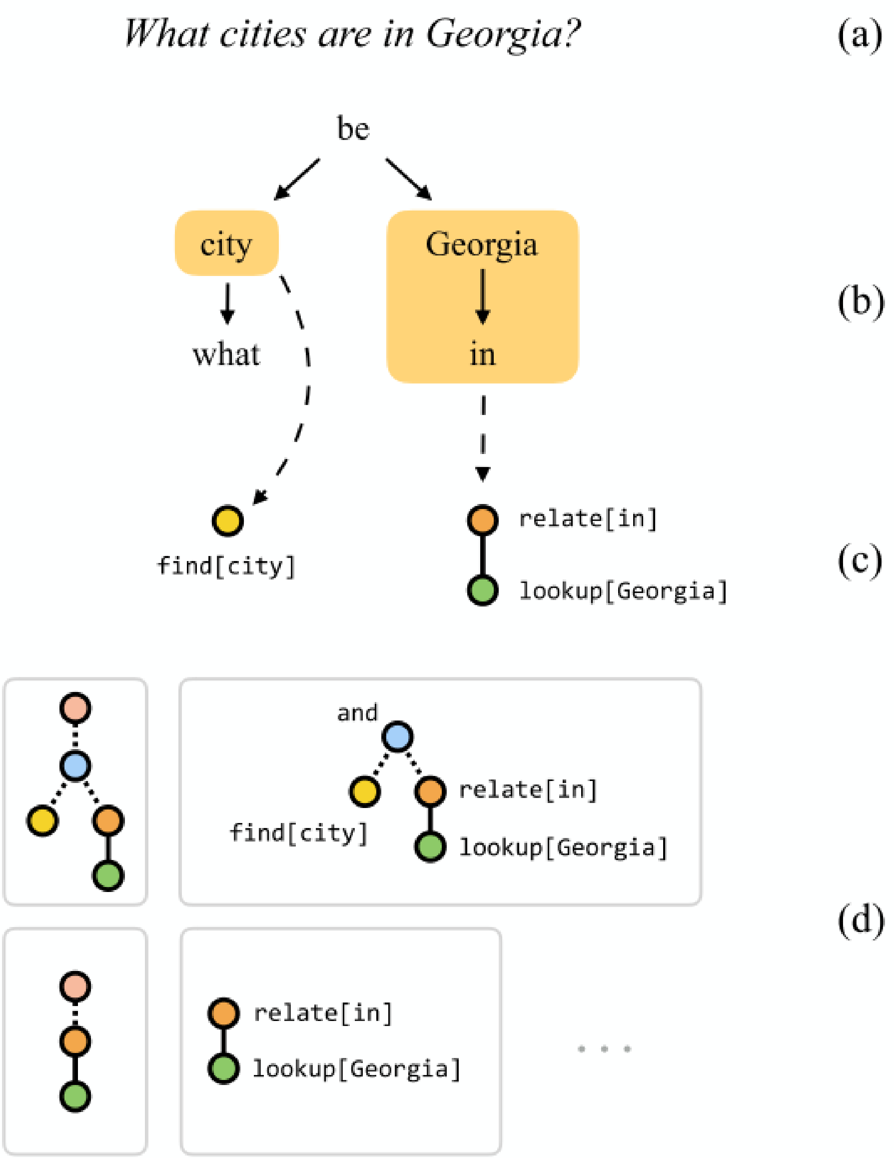
$$maximize \sum_{(w,y,z)} \log p_z(y|w,\theta_e)$$

- Layout model:
  - The modules used in this paper are shown below:

    Lookup, Find, Relate, And, Describe, Exists
- First use a fixed syntactic parse to generate a small set of candidate layouts in figure (d).
- Then we need to score them. This is a ranking problem.

$$s(z_i|x) = a^\top \sigma(Bh_q(x) + Cf(z_i) + d)$$

$$p(z_i|x;\theta_\ell) = e^{s(z_i|x)} \Big/ \sum_{j=1}^{n} e^{s(z_j|x)}$$



Jointly learning by RL

- Because the hard selection of **z** is non-differentiable, we optimize $p(z|x;\theta_l)$ using a policy gradient method. The gradient of the reward surface J with respect to the parameters of the policy is

$$\nabla J(\theta_\ell) = \mathbb{E}[\nabla \log p(z|x;\theta_\ell) \cdot r]$$

- We take the reward **r** to be identically the negative log-probability from the execution phase:

$$\mathbb{E}[(\nabla \log p(z|x;\theta_\ell)) \cdot \log p(y|z,w;\theta_e)]$$

Learning to Reason: End-to-End Module Networks for Visual Question Answering

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, Kate Saenko

Motivation

- Limitations of previous work:
  - Rely on an external parser.
  - None of the existing methods can learn to predict a suitable structure for every input in an end-to-end manner.
- Our approach learns to optimize over the full space of network layouts rather than acting as a reranker, and requires no parser at evaluation time.

Contributions

- 1) a method for learning a layout policy that dynamically predicts a network structure for each instance, without the aid of external linguistic resources at test time
- 2) a new module parameterization that uses a soft attention over question words rather than hard-coded word assignments

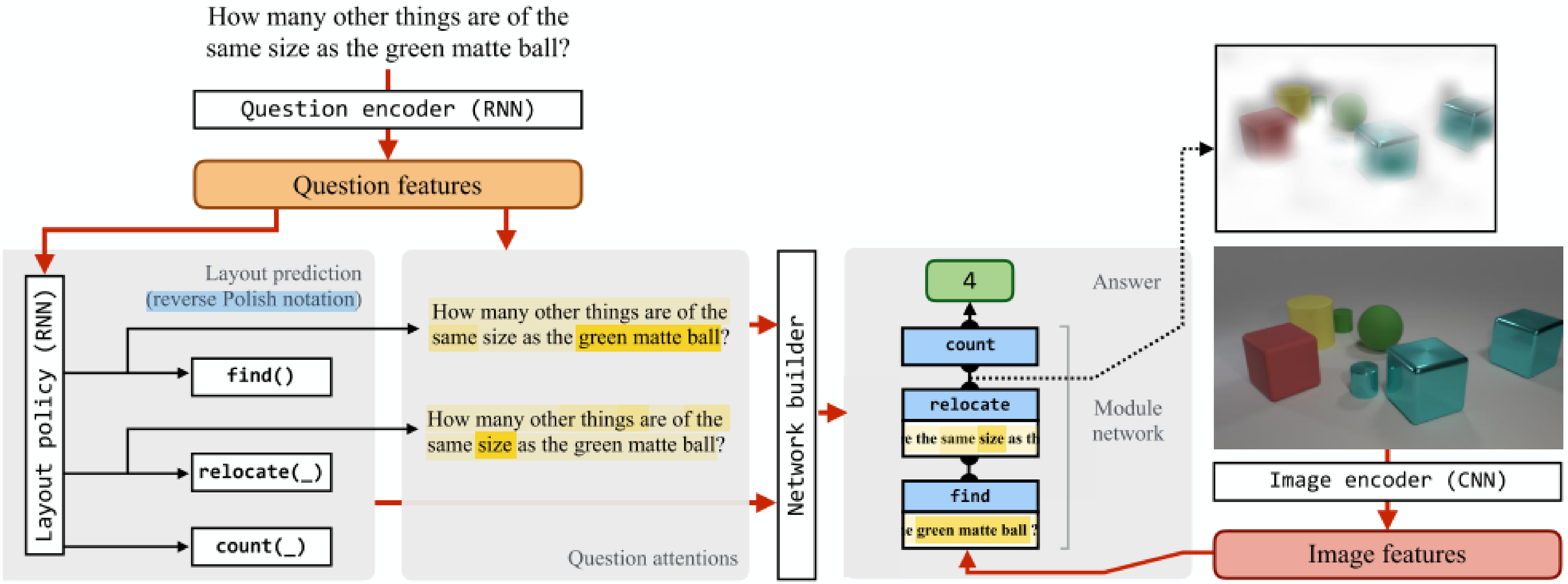
Learning to Reason: End-to-end Module Networks for Visual Question Answering

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko

UC Berkeley and Boston University

https://arxiv.org/abs/1704.05526

End-to-End Module Networks



Module name	Att-inputs	Features	Output	Implementation details
`find`	(none)	x_{vis}, x_{txt}	att	$a_{out} = \text{conv}_2\left(\text{conv}_1(x_{vis}) \odot W x_{txt}\right)$
`relocate`	a	x_{vis}, x_{txt}	att	$a_{out} = \text{conv}_2\left(\text{conv}_1(x_{vis}) \odot W_1 \text{sum}(a \odot x_{vis}) \odot W_2 x_{txt}\right)$
`and`	a_1, a_2	(none)	att	$a_{out} = \text{minimum}(a_1, a_2)$
`or`	a_1, a_2	(none)	att	$a_{out} = \text{maximum}(a_1, a_2)$
`filter`	a	x_{vis}, x_{txt}	att	$a_{out} = \texttt{and}(a, \texttt{find}[x_{vis}, x_{txt}]())$, *i.e.* reusing `find` and `and`
`[exist, count]`	a	(none)	ans	$y = W^T \text{vec}(a)$
`describe`	a	x_{vis}, x_{txt}	ans	$y = W_1^T \left(W_2 \text{sum}(a \odot x_{vis}) \odot W_3 x_{txt}\right)$
`[eq_count, more, less]`	a_1, a_2	(none)	ans	$y = W_1^T \text{vec}(a_1) + W_2^T \text{vec}(a_2)$
`compare`	a_1, a_2	x_{vis}, x_{txt}	ans	$y = W_1^T \left(W_2 \text{sum}(a_1 \odot x_{vis}) \odot W_3 \text{sum}(a_2 \odot x_{vis}) \odot W_4 x_{txt}\right)$

Table 1: The full list of neural modules in our model. Each module takes 0, 1 or 2 attention maps (and also visual and textual features) as input, and outputs either an attention map a_{out} or a score vector y for all possible answers. The operator $\odot$ is element-wise multiplication, and sum is summing the result over spatial dimensions. The vec operation is flattening an attention map into a vector, and adding two extra dimensions: the max and min over attention map.

End-to-end training

- During training, we jointly learn the layout policy $p(l|q)$ and the parameters in each neural module, and minimize the expected loss from the layout policy.

$$L(\theta) = E_{l \sim p(l|q;\theta)}[\tilde{L}(\theta, l; q, I)]$$

- estimated using Monte-Carlo sampling as

$$\nabla_\theta L \approx \frac{1}{M} \sum_{m=1}^{M} \left(\tilde{L}(\theta, l_m) \nabla_\theta \log p(l_m|q;\theta) + \nabla_\theta \tilde{L}(\theta, l_m) \right)$$

- Behavioral cloning from expert polices for pre-train.

Experiments

Method	Accuracy
NMN [3]	90.80%
ours - behavioral cloning from expert	100.00%
ours - policy search from scratch	96.19%

Table 2: Performance of our model on the SHAPES dataset. "ours - behavioral cloning from expert" corresponds to the supervised behavioral cloning from the expert policy p_e, and "ours - policy search from scratch" is directly optimizing the layout policy without utilizing any expert policy.

Method	Visual feature	Accuracy
NMN [3]	LRCN VGG-16	57.3
D-NMN [2]	LRCN VGG-16	57.9
MCB [9]	ResNet-152	64.7
ours - cloning expert	LRCN VGG-16	61.9
ours - cloning expert	ResNet-152	64.2
ours - policy search after cloning	ResNet-152	**64.9**

Table 4: Evaluation of our method on the VQA test-dev set. Our model outperforms previous work NMN and D-NMN and achieves comparable performance as MCB.

Conclusions

- Commons:
  - Dynamic structure.
  - Break down the main task.
- Advantages:
  - Smaller space in subtask.
  - Better explanation.
- Future work:
  - How to solve the more general question?

Discussion

- How about the similar methods on other task?
  - eg. Math Problem, Question Answer, Conversation …