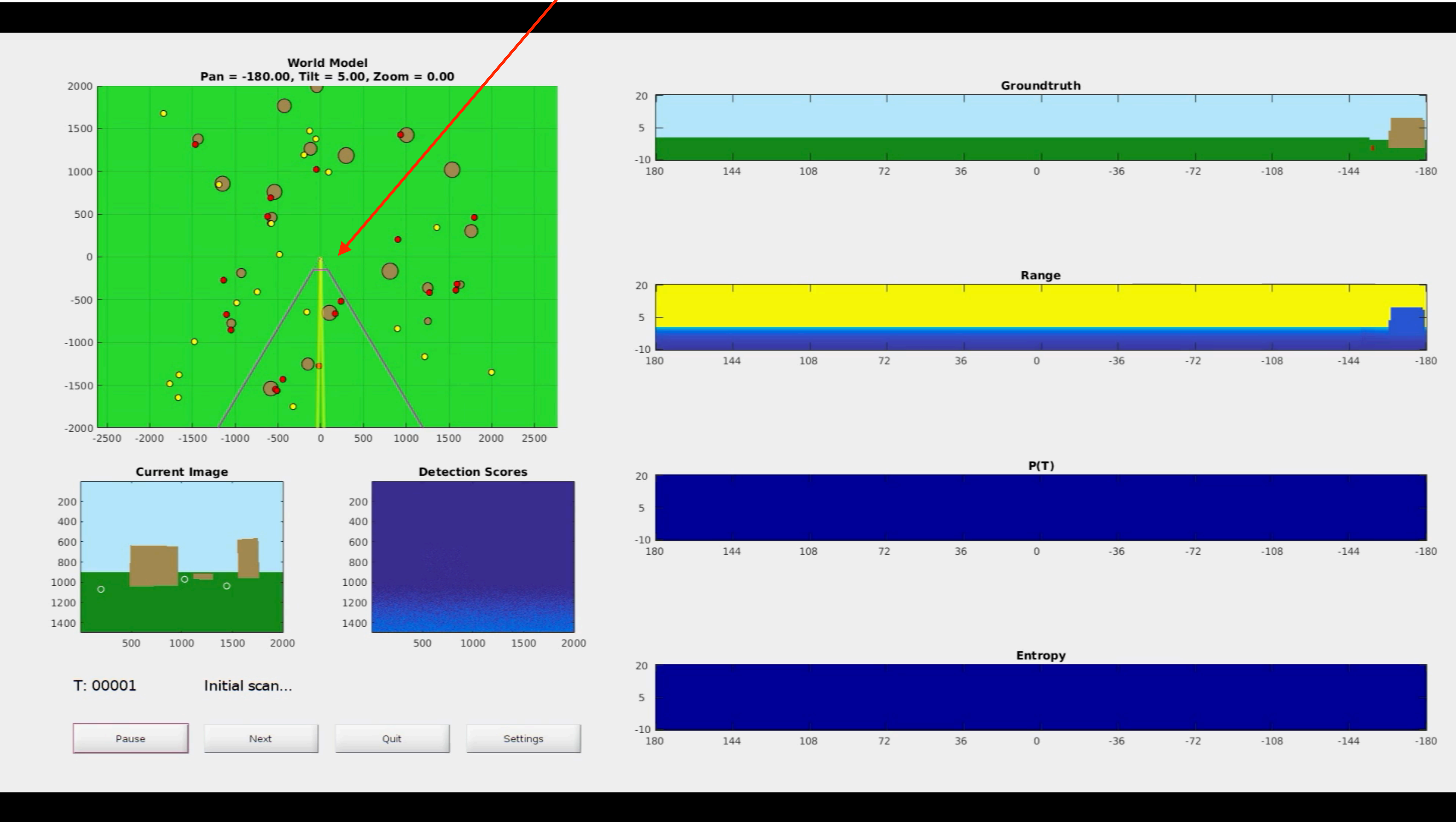
Curriculum Domain Adaptation:

Using Simulation for Real

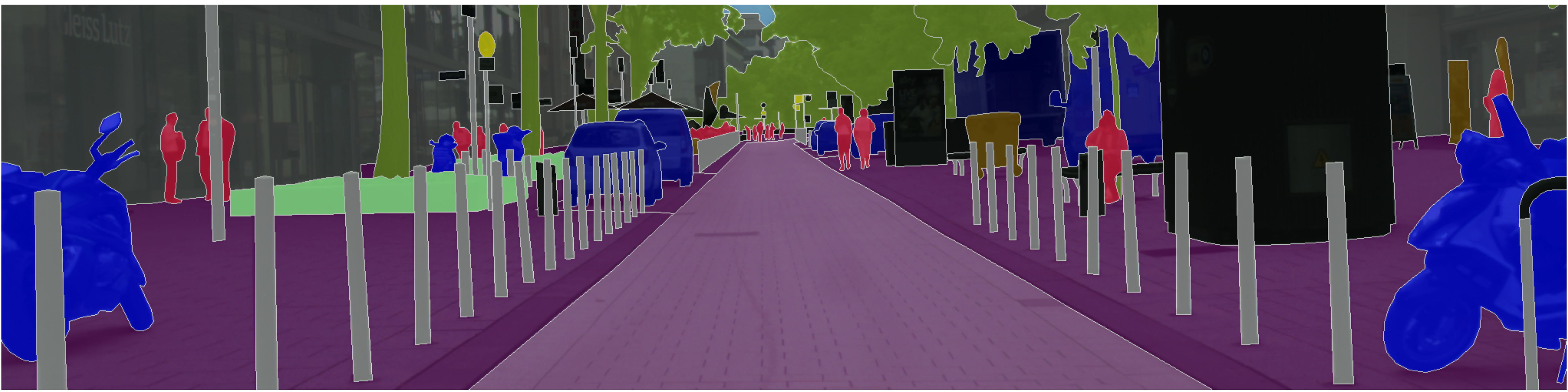
Boqing Gong

Tencent AI Lab

An intelligent robot



Semantic segmentation of urban scenes

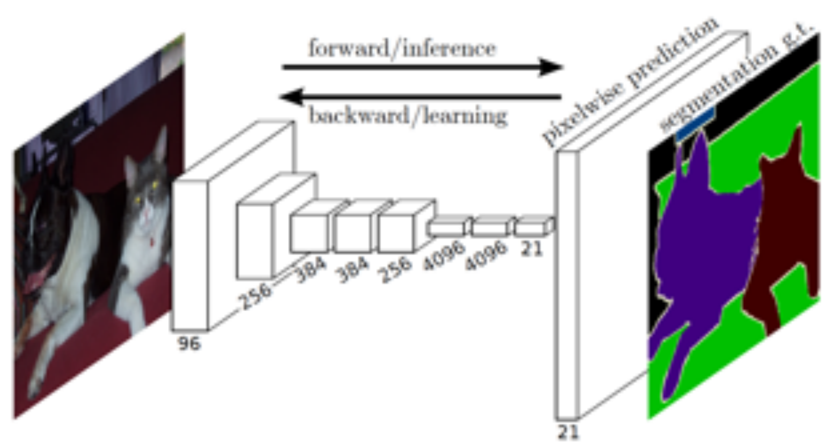


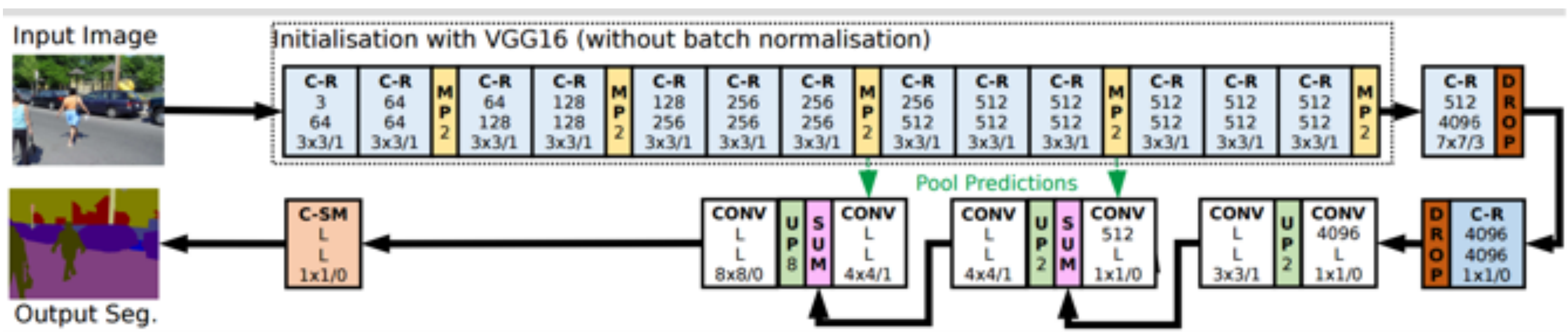
Assign each pixel a semantic label

An appealing application: **self-driving**

Image credit: https://www.cityscapes-dataset.com/

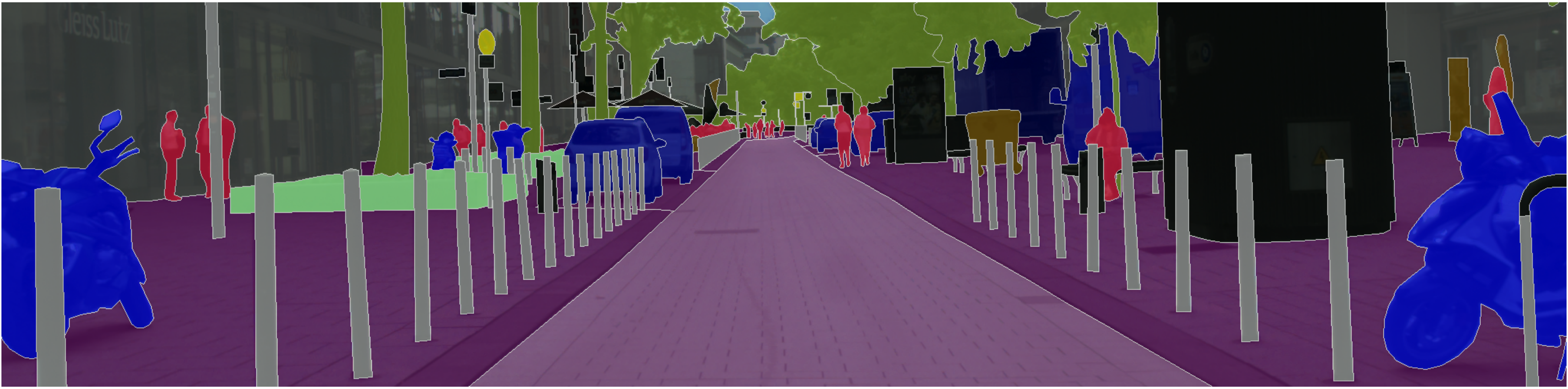
Triumphal approach: CNNs convolutional neural networks





Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

To teach/train CNNs to segment images and videos



About 1.5 hrs to label one such image!

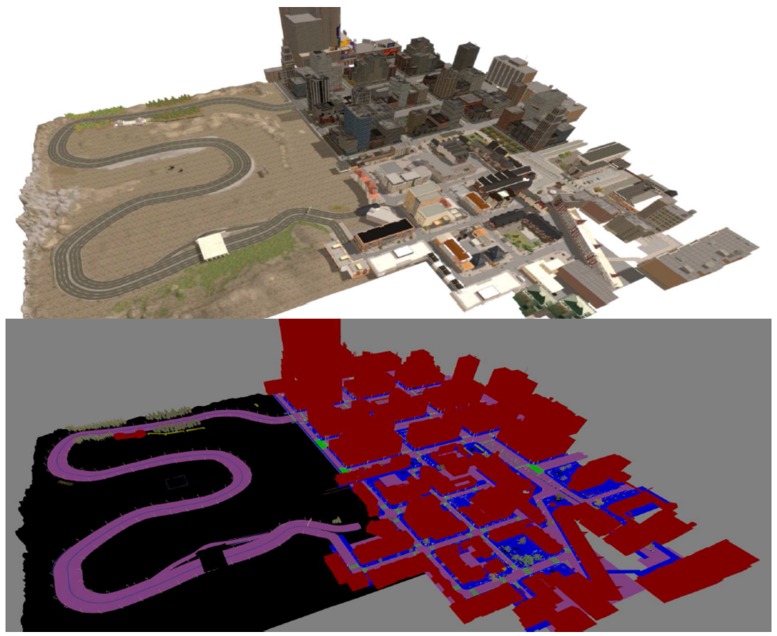
Cityscapes: largest publicly available dataset

30k images captured from 50 cities

Only 5k are well labeled thus far

Image credit: https://www.cityscapes-dataset.com/

Labeling-free training data by simulation



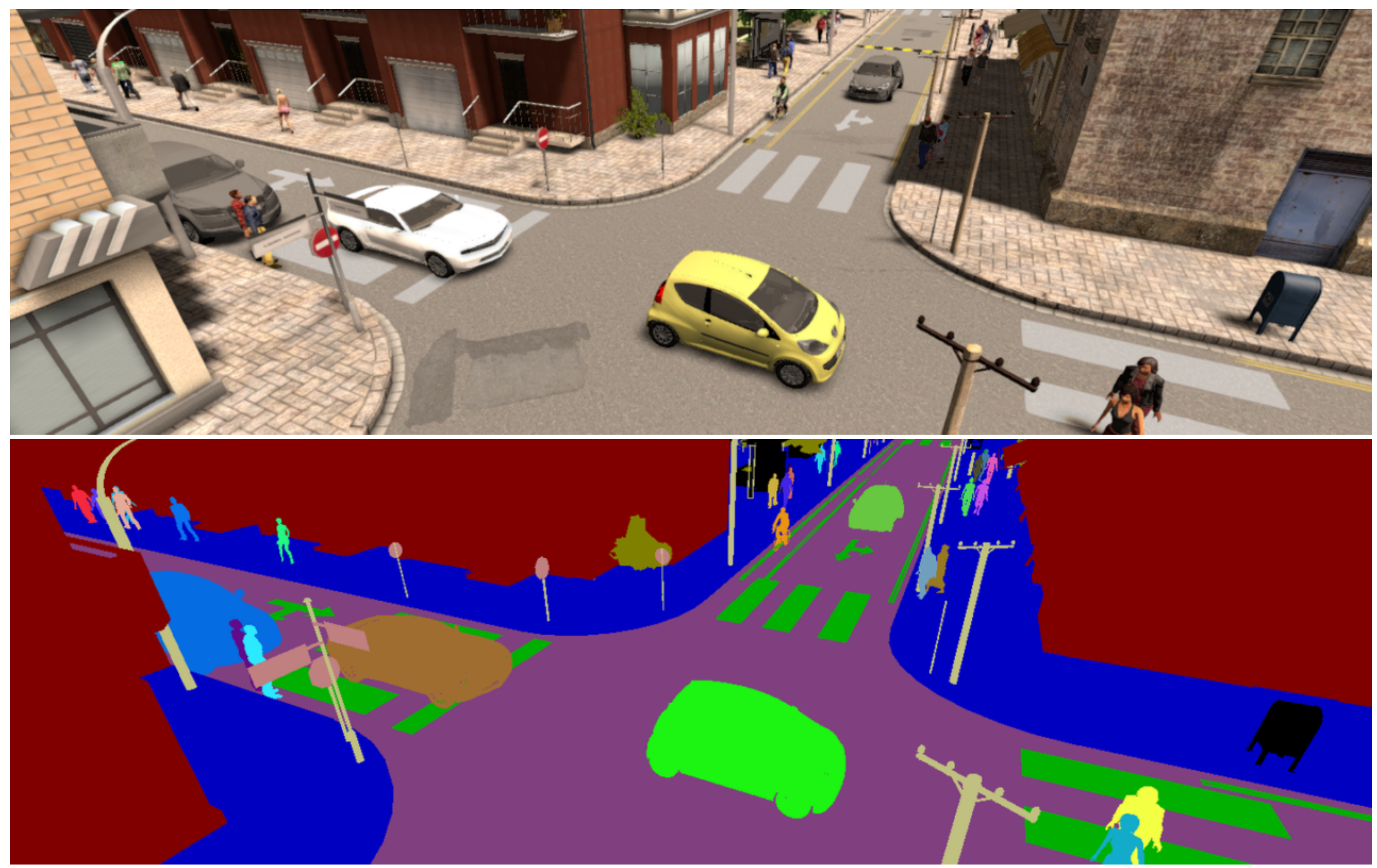
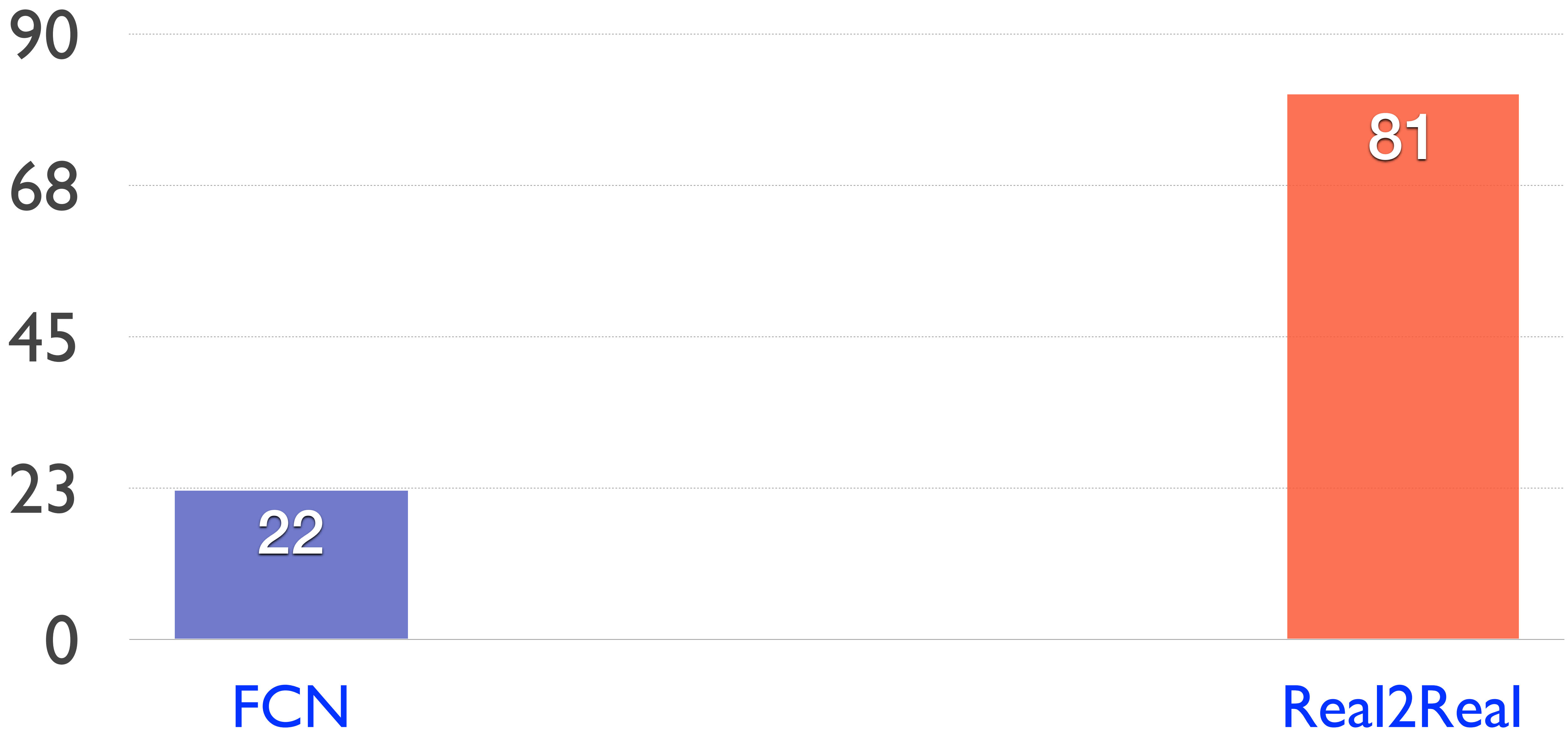


Image credit: http://synthia-dataset.net/

Simulation to real world: catastrophic performance drop



The perils of mismatched domains

Cause: standard assumption in machine learning

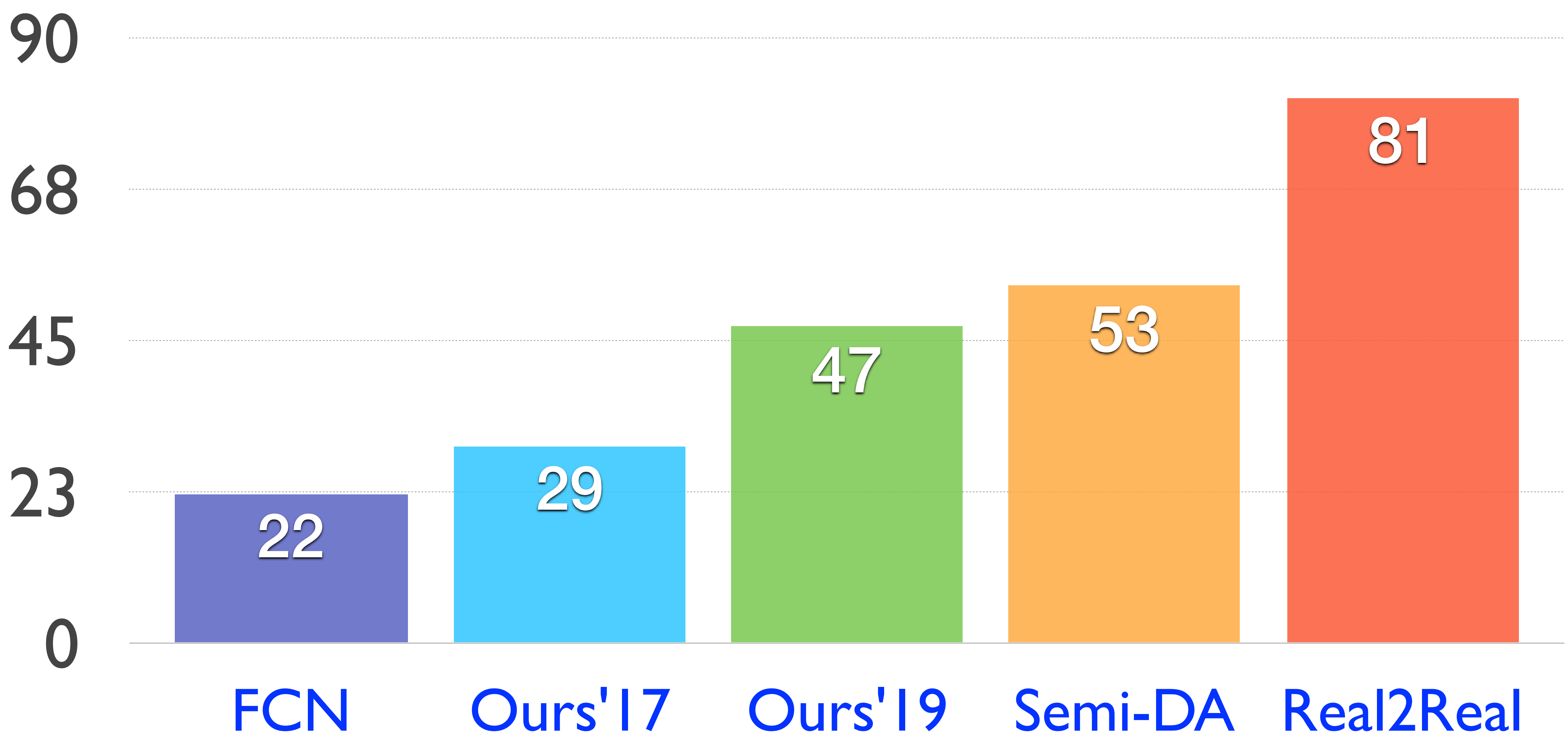
Same underlying distribution for training and testing

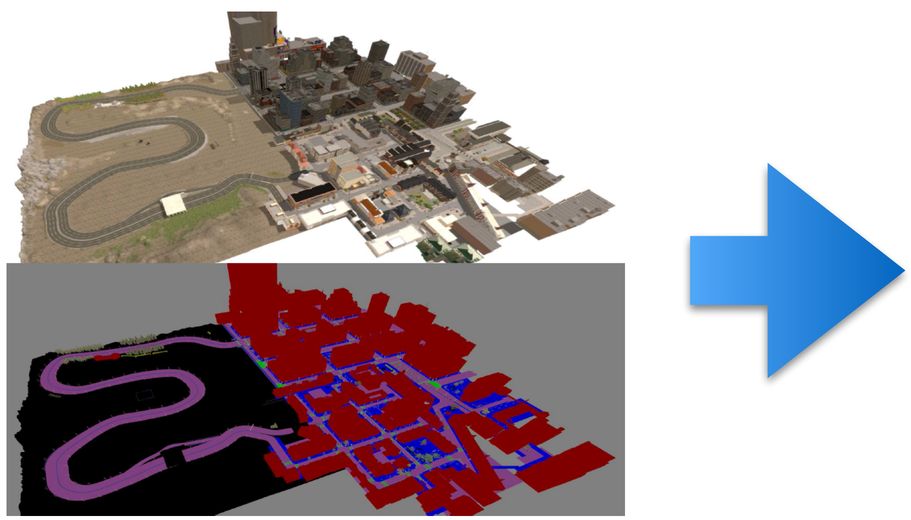
Consequence:

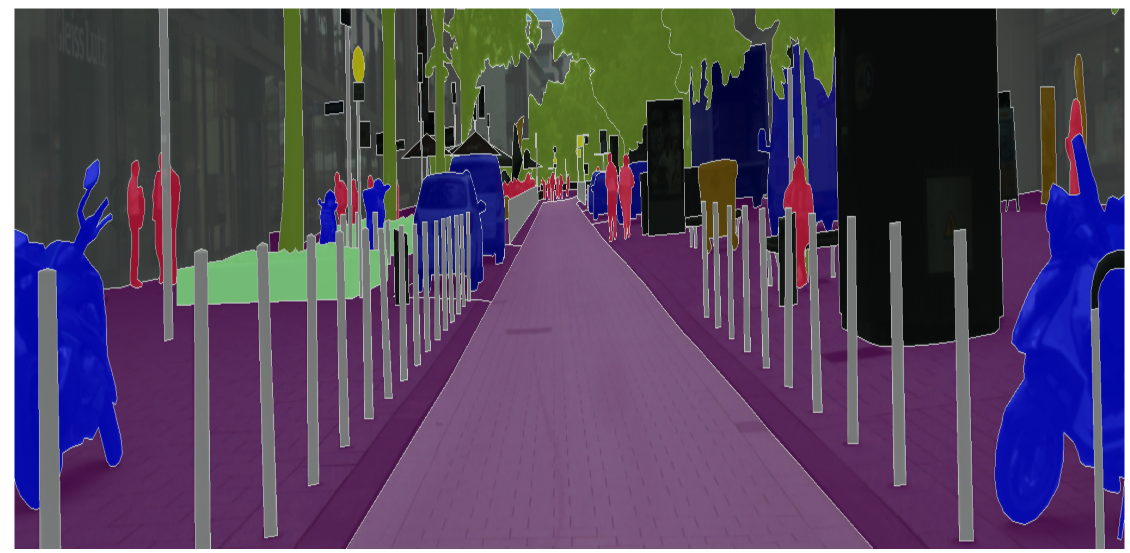
Poor cross-domain generalization

Brittle systems in dynamic and changing environment

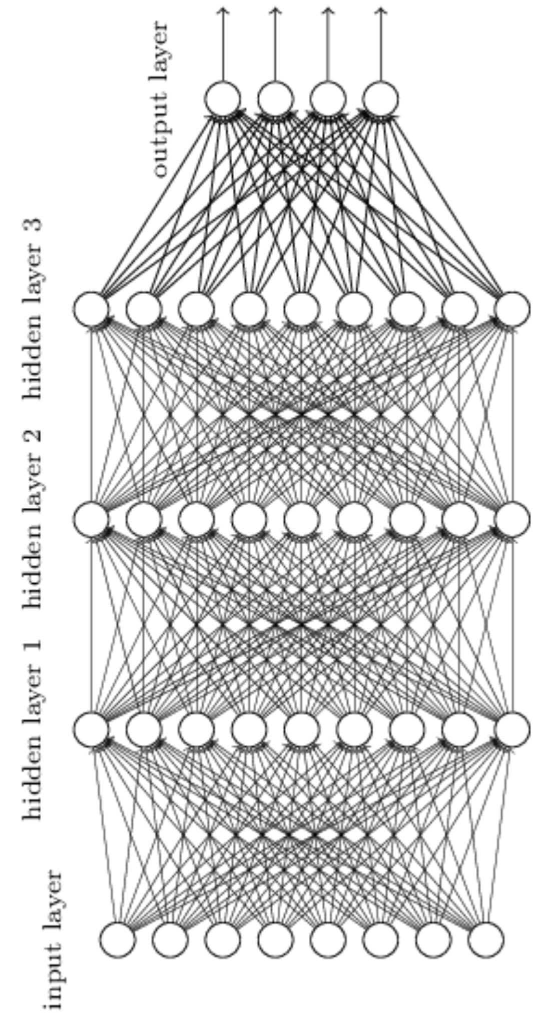
Simulation to real world: closing the performance gap?



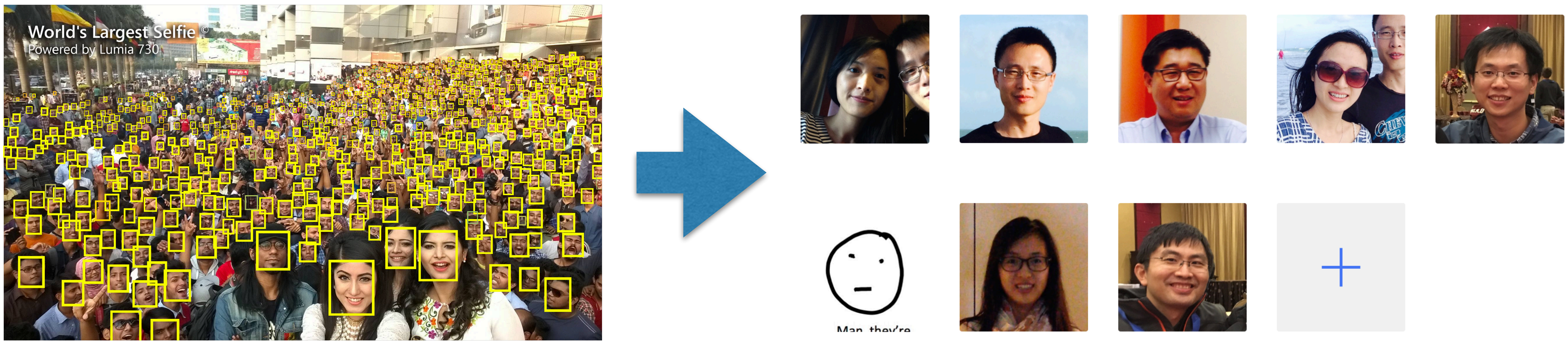




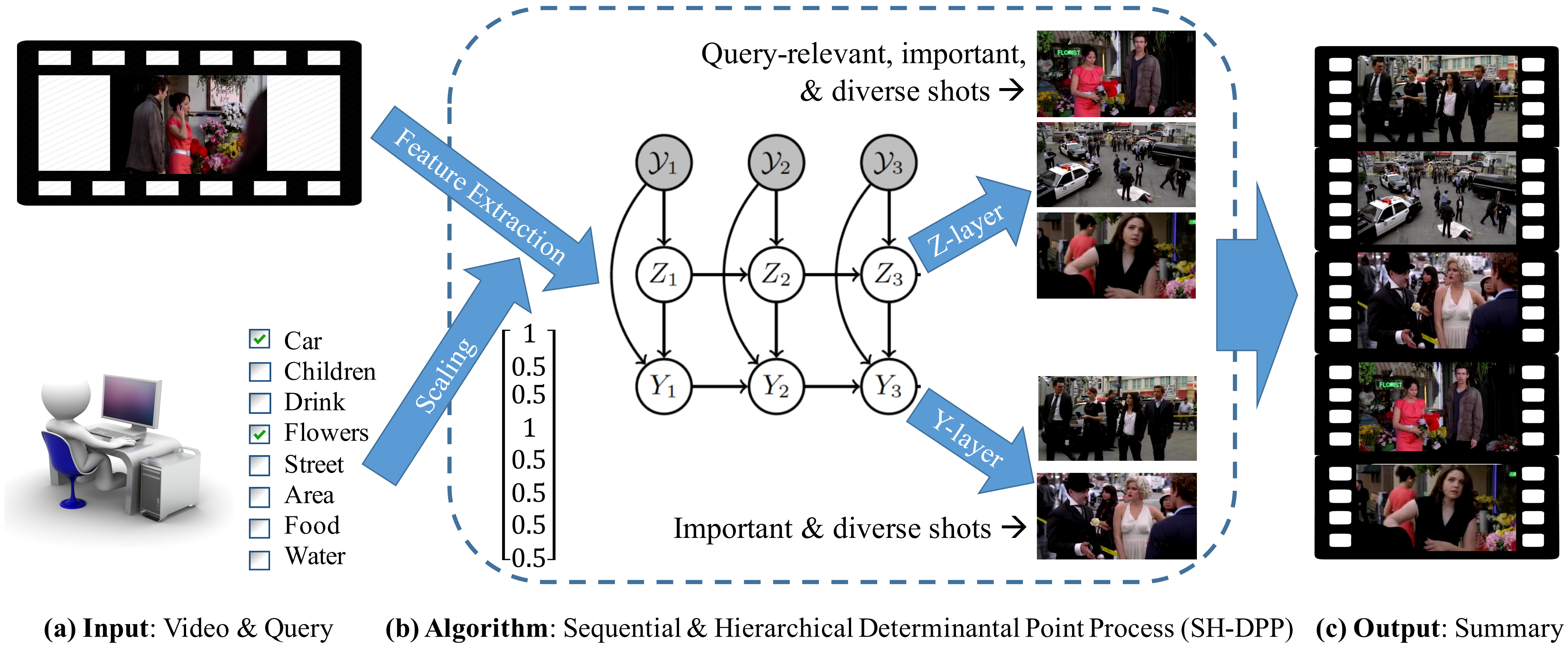
Synthetic imagery → Real photos



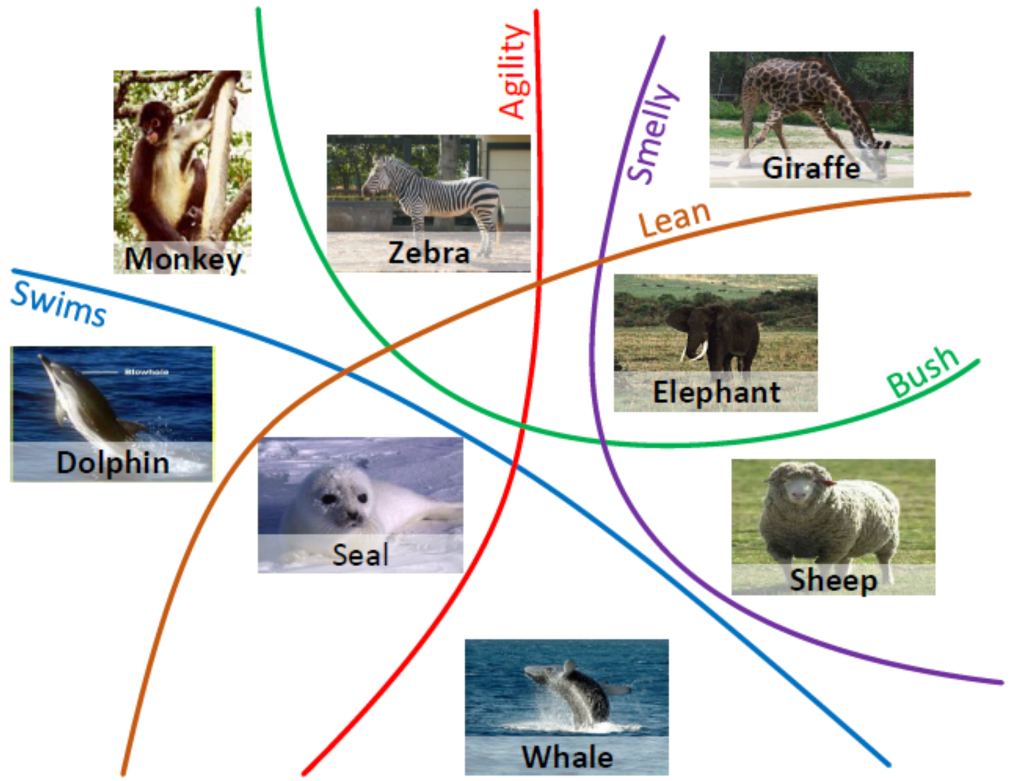
Webly supervised learning



Adapting face detector to a user's album



Personalization of video summarizers



Middle-level concepts to describe objects, faces, etc.

Shared by different categories

Attribute detection

Abstract form: *unsupervised* domain adaptation (DA)

Setup

Source domain (with labeled data)

$$D_{\mathcal{S}} = \{(x_m, y_m)\}_{m=1}^{\mathsf{M}} \sim \boxed{P_{\mathcal{S}}(X, Y)}$$

Target domain (no labels for training)

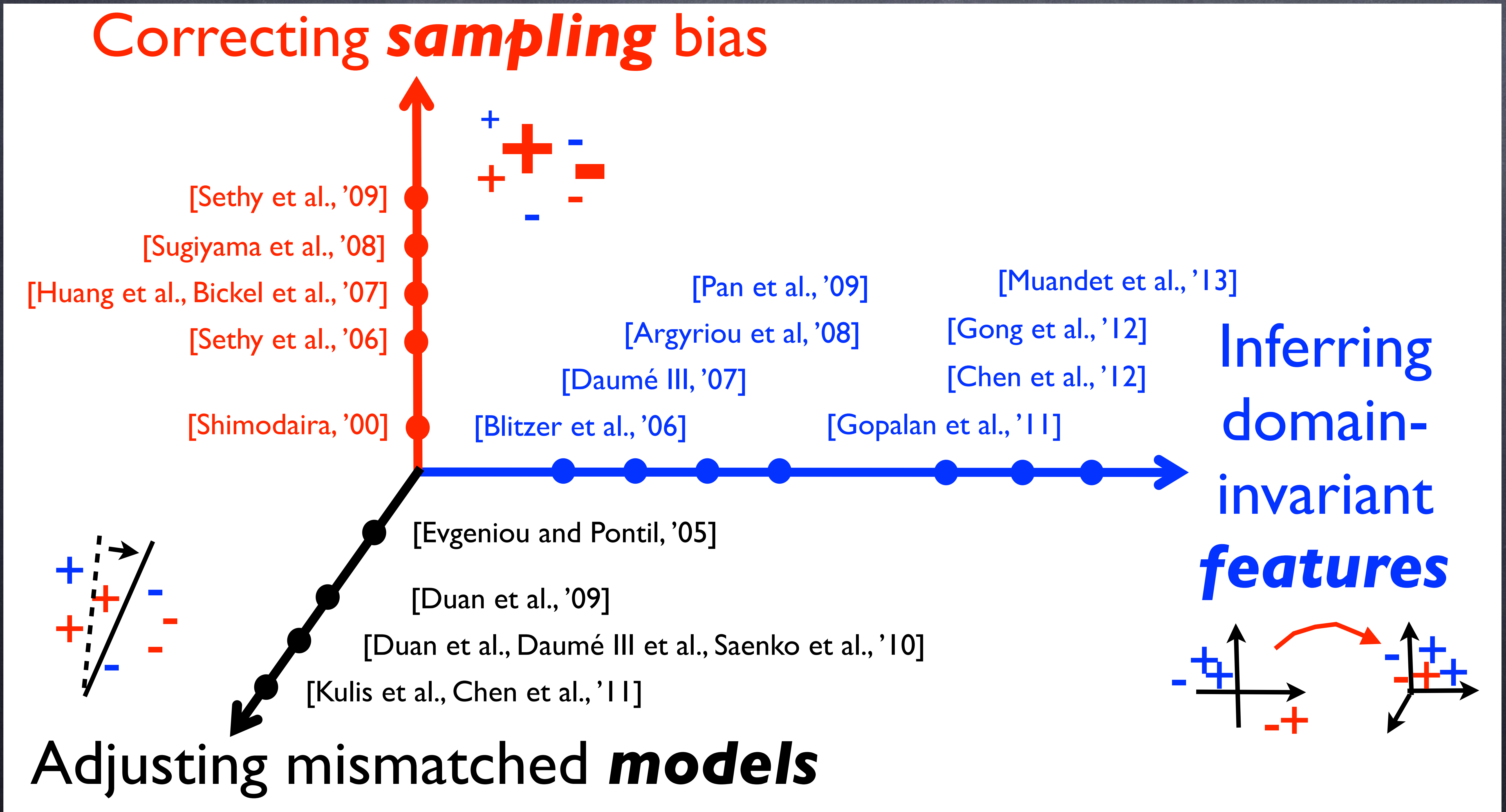
$$D_{\mathcal{T}} = \{(x_n, \textbf{?})\}_{n=1}^{\mathsf{N}} \sim \boxed{P_{\mathcal{T}}(X, Y)}$$

Different distributions

Objective

Learn models to work well on target

This talk


Correcting sampling bias
[Sethy et al., '09]
[Sugiyama et al., '08]
[Huang et al., Bickel et al., '07]
[Sethy et al., '06]
[Shimodaira, '00]
[Pan et al., '09]
[Muandet et al., '13]
[Argyriou et al, '08]
[Gong et al., '12]
[Daumé III, '07]
[Chen et al., '12]
[Blitzer et al., '06]
[Gopalan et al., '11]
Inferring domain-invariant features
[Evgeniou and Pontil, '05]
[Duan et al., '09]
[Duan et al., Daumé III et al., Saenko et al., '10]
[Kulis et al., Chen et al., '11]
Adjusting mismatched models

This talk

Correcting ***sampling*** bias

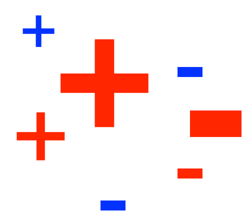
[Sethy et al., '09]

[Sugiyama et al., '08]

[Huang et al., Bickel et al., '07]

[Sethy et al., '06]

[Shimodaira, '00]



$$P_{\mathcal{L}}(\text{landmarks}) \approx P_{\mathcal{T}}(\text{target})$$

$$\min_{\text{landmarks}} \quad d(P_{\mathcal{L}}, P_{\mathcal{T}})$$

Selecting most adaptable source instances

Landmarks are labeled source instances distributed similarly to the target domain.

Source

Target

[ICML'13]

Selecting most adaptable source instances

Landmarks are labeled source instances distributed similarly to the target domain.

Source

Identifying landmarks:

$$P_{\mathcal{L}}(\text{landmarks}) \approx P_{\mathcal{T}}(\text{target})$$

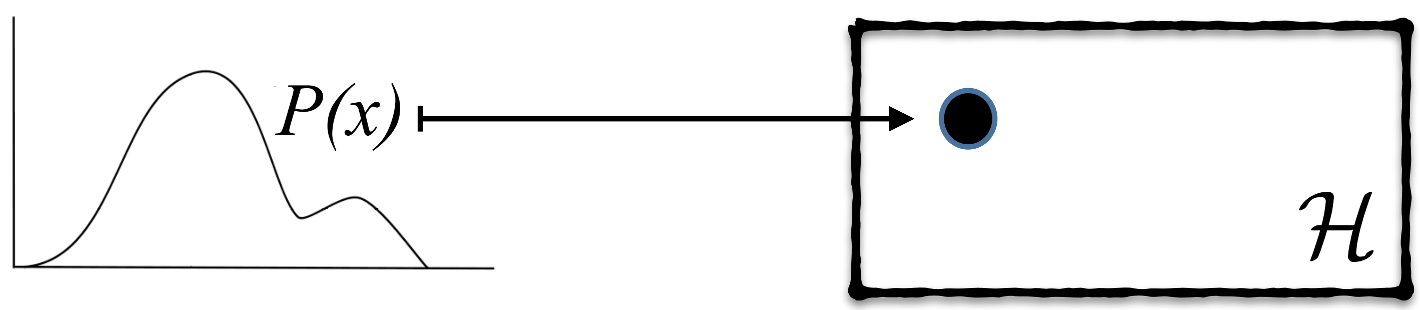
$$\min_{\text{landmarks}} \quad d(P_{\mathcal{L}}, P_{\mathcal{T}})$$

Target

[ICML'13]

Kernel embedding of distributions

$$\mu[P] \triangleq \mathbb{E}_x[\phi(x)]$$



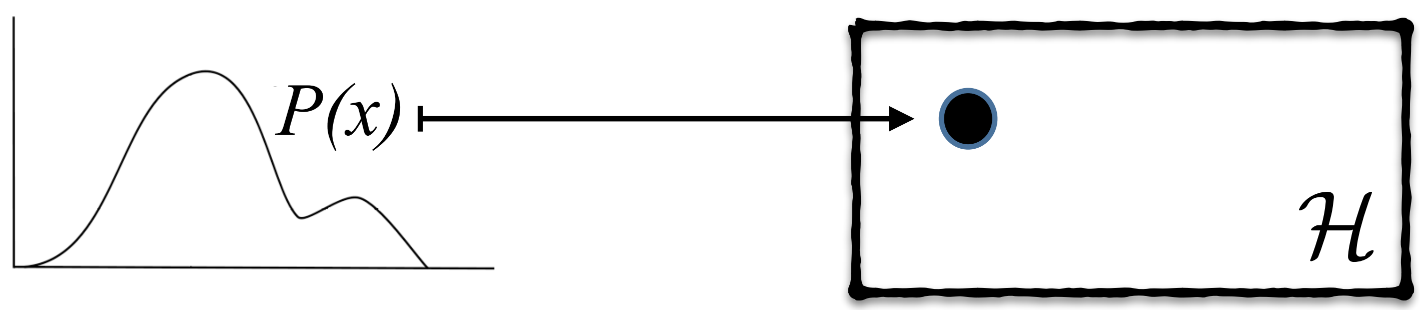
μ maps distribution P to Reproducing Kernel Hilbert Space

μ is injective if $\phi(\cdot)$ is characteristic

[Müller'97,Gretton et al.'07,Sriperumbudur et al.'10]

Kernel embedding of distributions

$$\mu[P] \triangleq \mathbb{E}_x[\phi(x)]$$



Empirical kernel embedding:

$$\hat{\mu}[P] = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i), \quad x_i \sim P$$

Identifying landmarks by matching kernel embeddings

Integer programming

$$\min_{\{\alpha_m\}} \left\| \frac{1}{\sum_i \alpha_i} \sum_{m=1}^{\mathsf{M}} \alpha_m \phi(x_m) - \frac{1}{\mathsf{N}} \sum_{n=1}^{\mathsf{N}} \phi(x_n) \right\|_{\mathcal{H}}^2$$

where

$$\alpha_m = \begin{cases} 1 & \text{if } x_m \text{ is a } \textbf{landmark} \text{ wrt target} \\ 0 & \text{else} \end{cases}$$

$$m = 1, 2, \cdots, \mathsf{M}$$

Solving by relaxation

Convex relaxation

$$\min_{\{\alpha_m\}} \left\| \boxed{\frac{1}{\sum_i \alpha_i}} \sum_{m=1}^{\mathsf{M}} \boxed{\alpha_m} \phi(x_m) - \frac{1}{\mathsf{N}} \sum_{n=1}^{\mathsf{N}} \phi(x_n) \right\|_{\mathcal{H}}^2$$

$$\boxed{\beta_m = \frac{\alpha_m}{\sum_i \alpha_i} \rightarrow \text{Quadratic programming}}$$

$$\min_{\beta} \quad \beta^T K^s \beta - \frac{2}{\mathsf{N}} \beta^T K^{st} \mathbf{1}$$

Other details

Class balance constraint

Recovering $\alpha_m^\star$ from $\beta_m^\star (= \frac{\alpha_m}{\sum_i \alpha_i})$

Multi-scale analysis

(See [Gong et al., ICML'13, IJCV'14] for details)

Experimental study

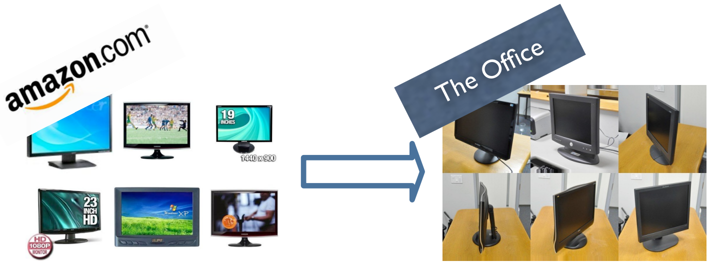
Four vision datasets/domains on visual object recognition

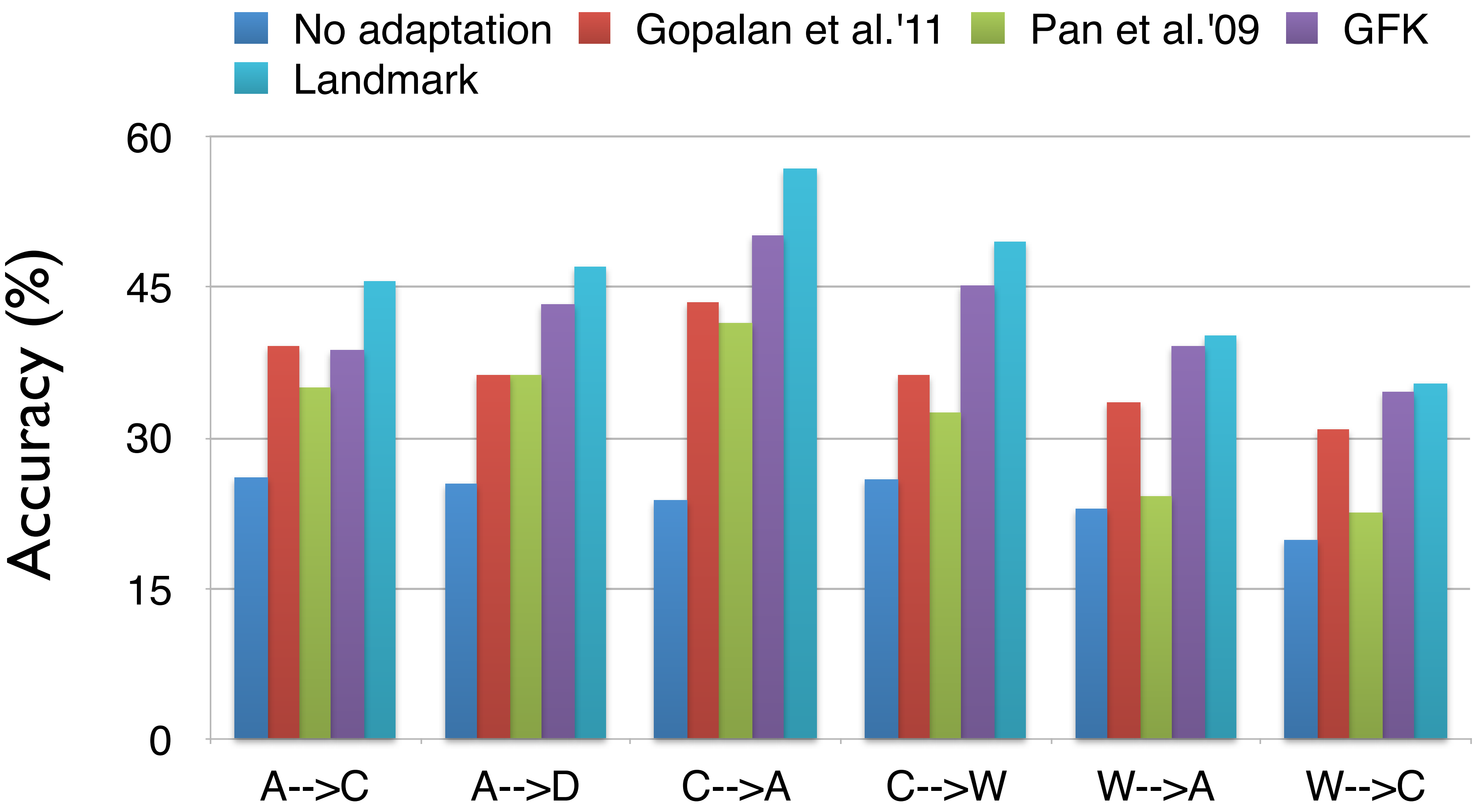
[Griffin et al. '07, Saenko et al. 10']

Four types of product reviews on sentiment analysis

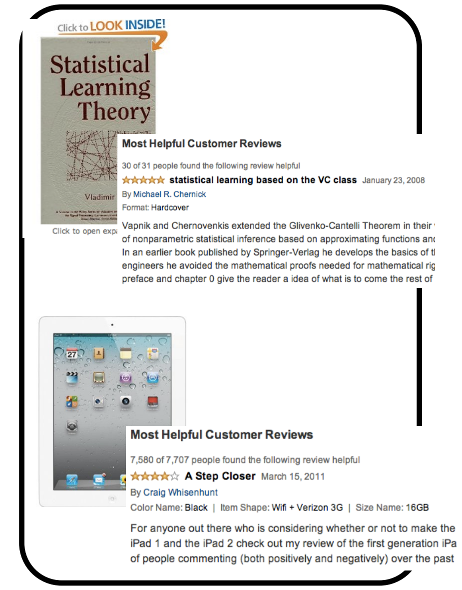
Books, DVD, electronics, kitchen appliances [Biltzer et al. '07]

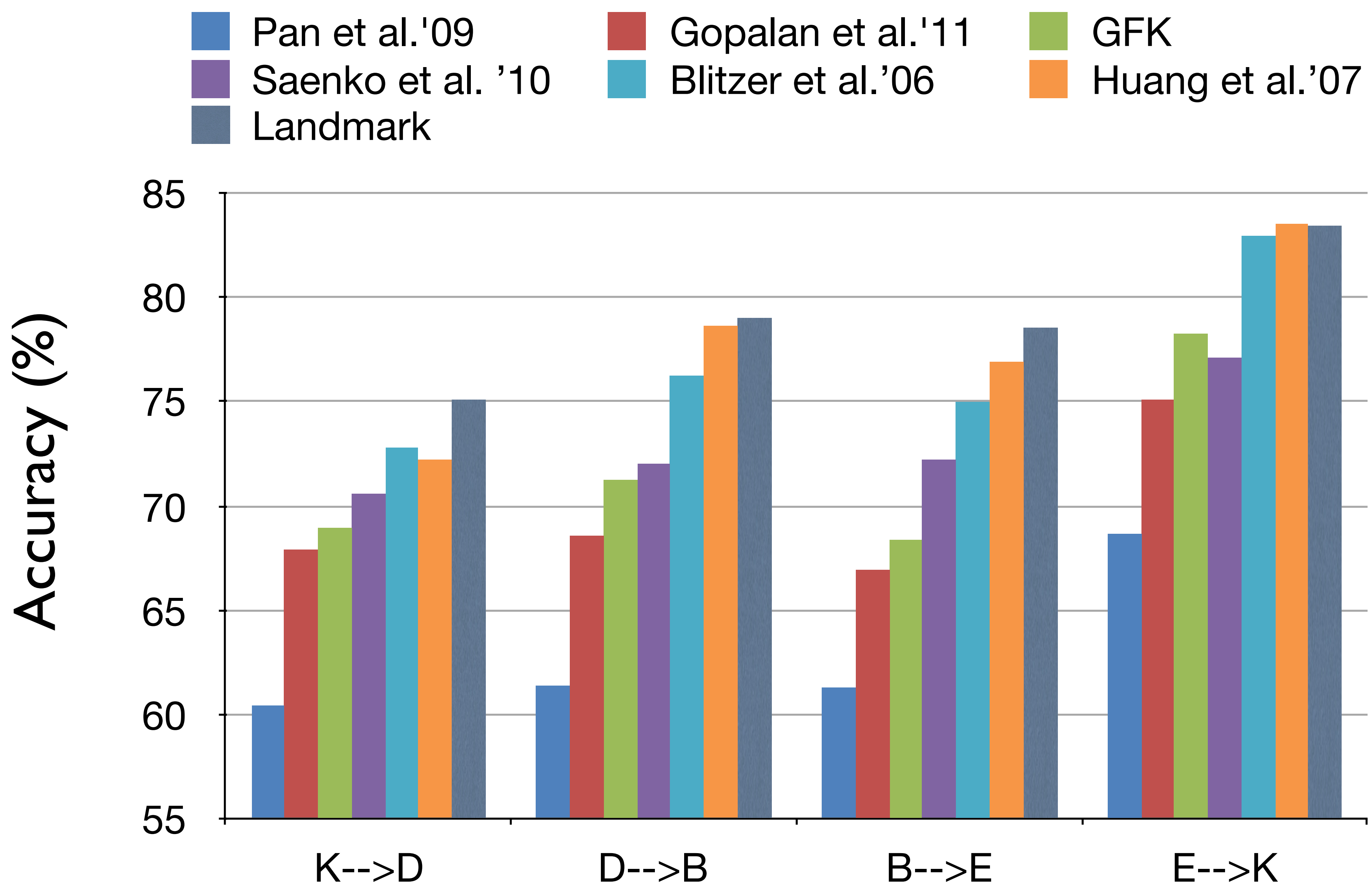
Comparison results: object recognition





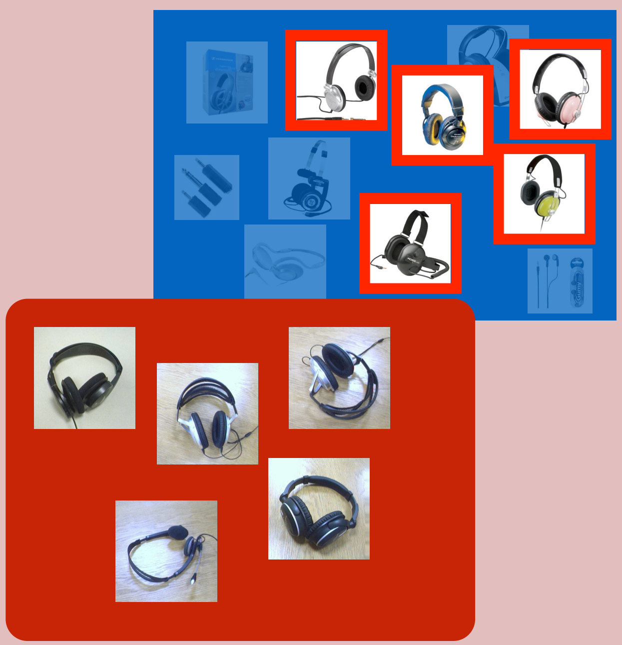
Comparison results: sentiment analysis





What do landmarks look like?

Summary - Landmarks



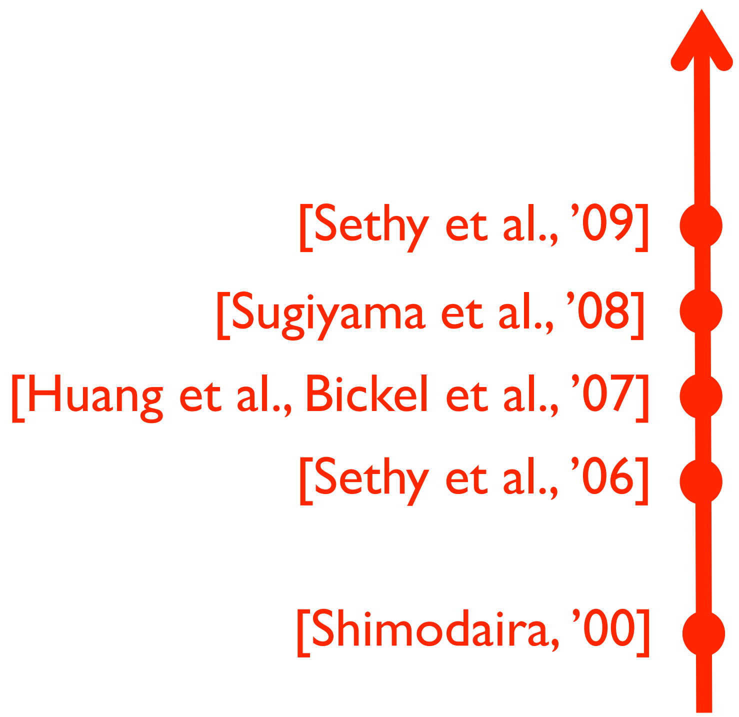
Landmarks

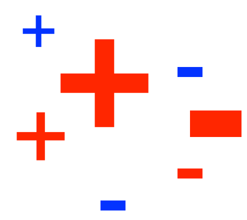
[Gong *et al.*, ICML'13]

- *Labeled source instances, distributed similarly to target*
- *Better approximation of discriminative loss of target*
- *Automatically identifying landmarks*
- *Benefiting other adaptation methods*

Snags in Landmarks

Correcting ***sampling*** bias



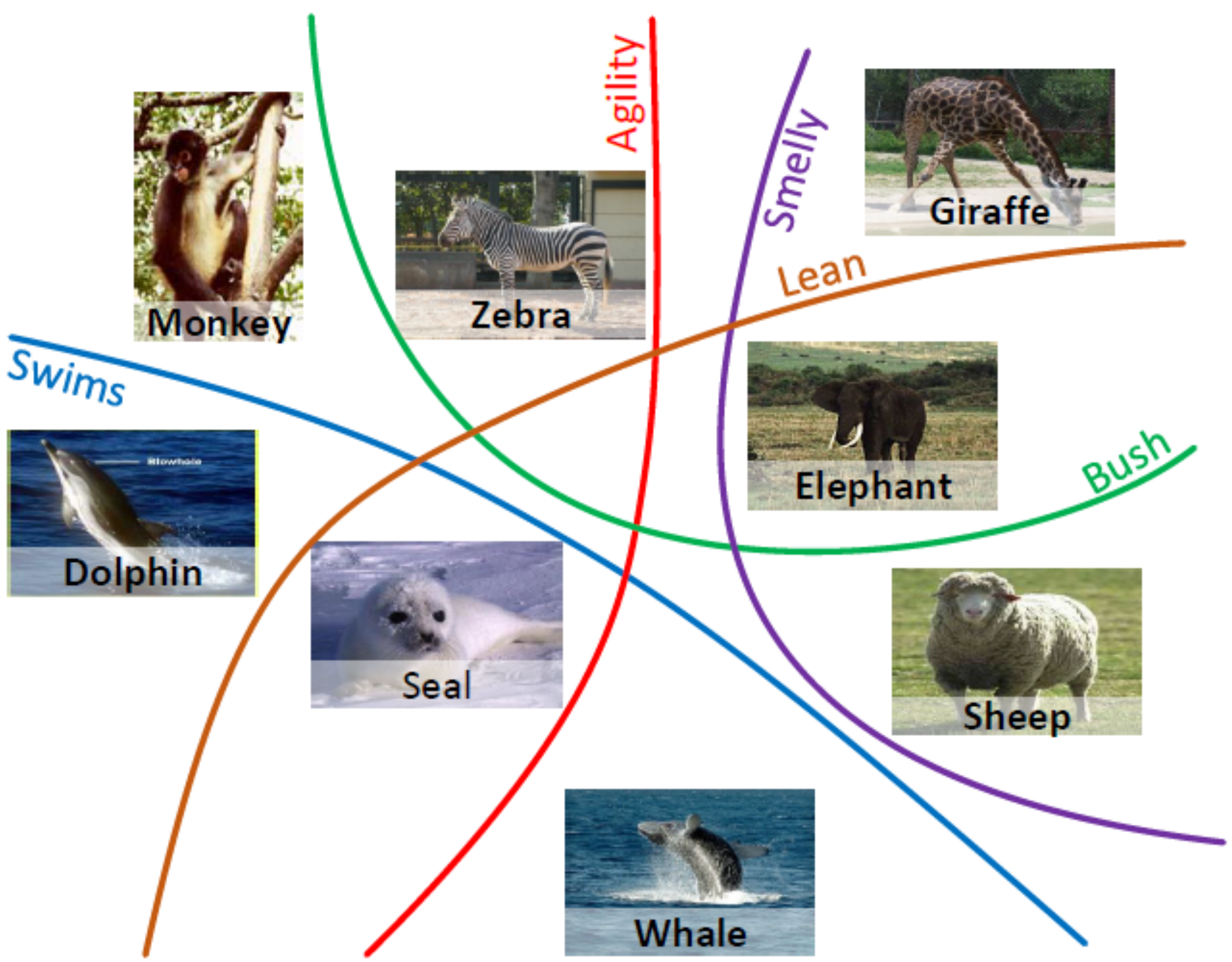


$$P_{\mathcal{L}}(\text{landmarks}) \approx P_{\mathcal{T}}(\text{target})$$

$$\min_{\text{landmarks}} \quad d(P_{\mathcal{L}}, P_{\mathcal{T}})$$

Large inter-domain discrepancy ***(seal vs whale)?***

What makes a good attribute detector?



Effective, efficient, … and *generalize well across different activity categories*, including previously unseen ones.

Boundaries between middle-level attributes and high-level object classes cross each other.

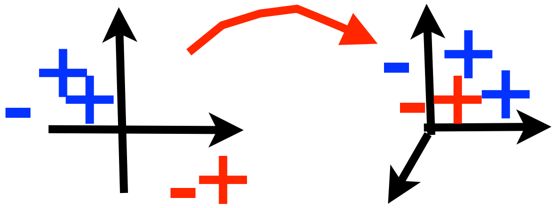
This talk

$$\mathbf{x} \mapsto \mathbf{z}, \quad \text{s.t.}$$

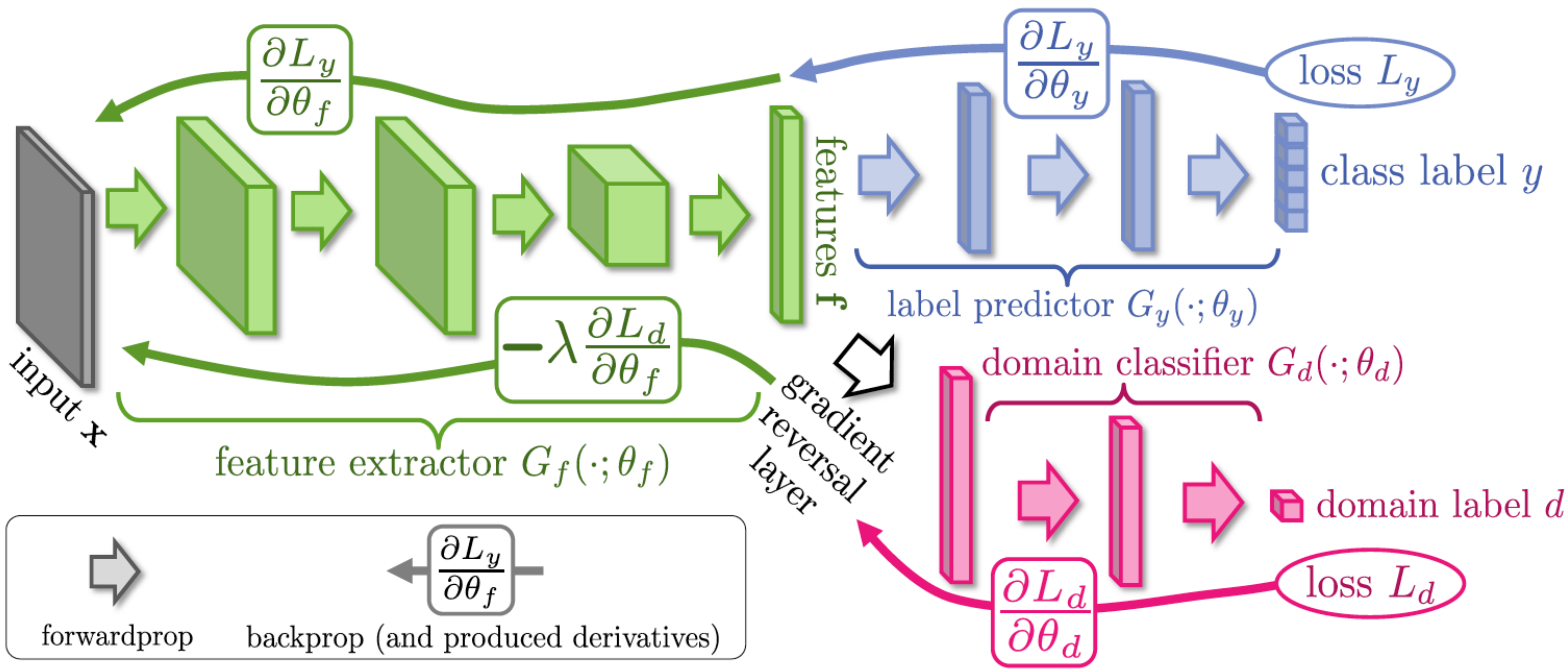
$$P_{\mathcal{S}}(z, y) \approx P_{\mathcal{T}}(z, y)$$

[Blitzer et al., '06]
[Daumé III, '07]
[Argyriou et al, '08]
[Pan et al., '09]
[Gopalan et al., '11]
[Chen et al., '12]
[Gong et al., '12]
[Muandet et al., '13]

Inferring domain-invariant ***features***



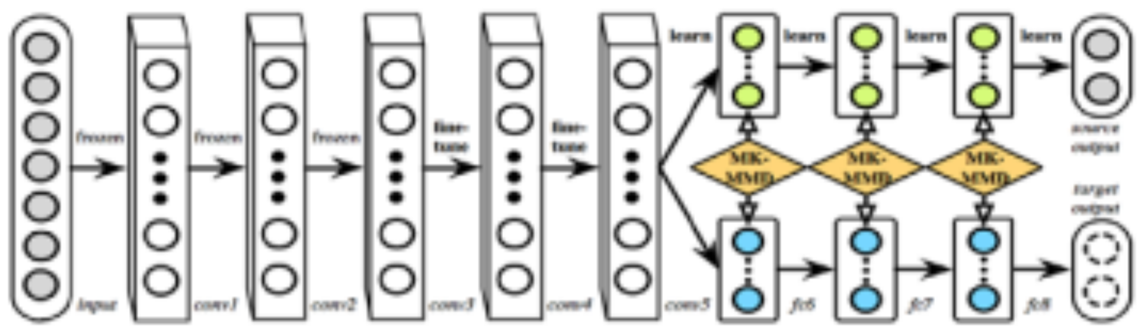
Review: maximizing the domain classification loss



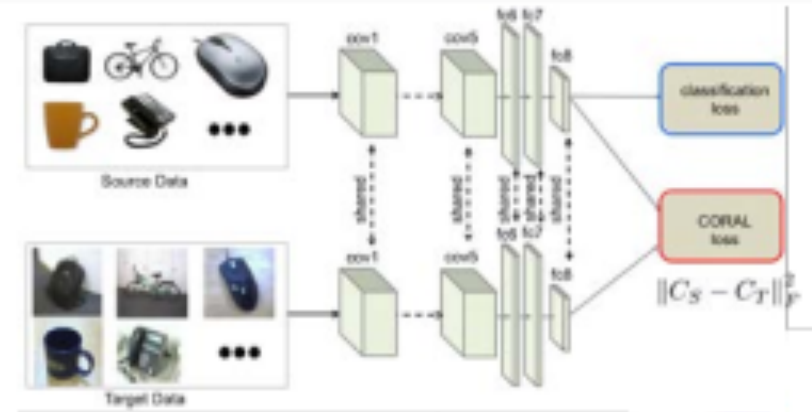
Ganin, Y., & Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. *International Conference on Machine Learning*.

Review

- by minimizing distance between distributions, e.g.

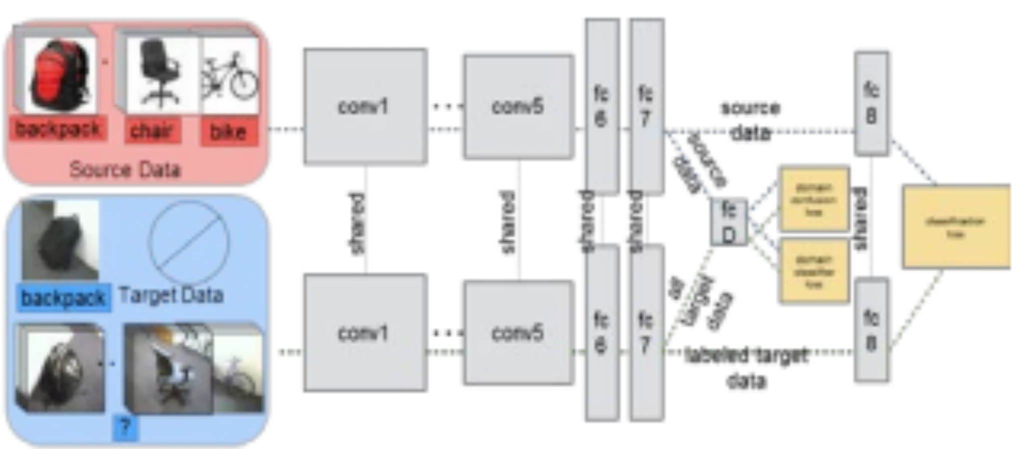


Maximum Mean Discrepancy M. Long, et al. ICML 2015

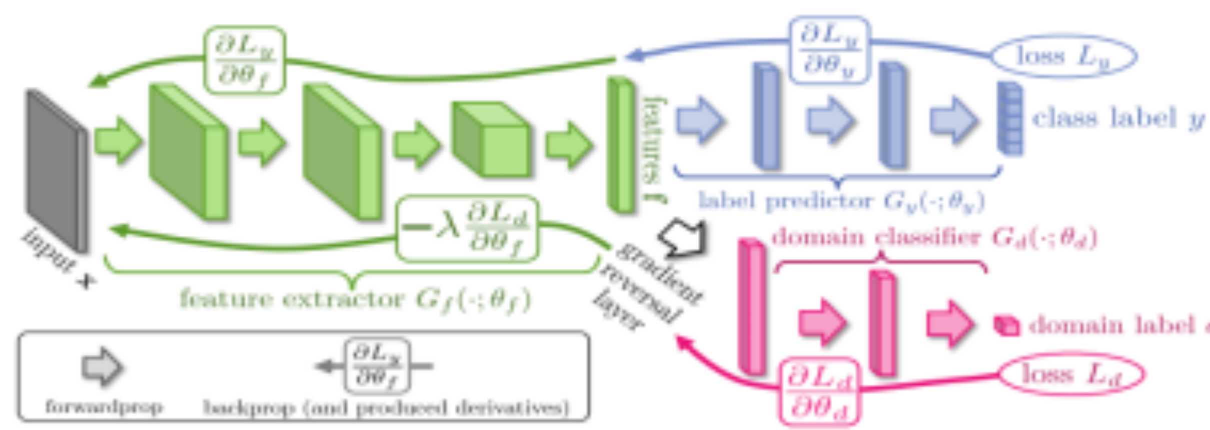


CORrelation ALignment Sun and Saenko, AAAI 2016

- …or by adversarial domain alignment, e.g.



Domain Confusion E. Tzeng et al. ICCV 2015

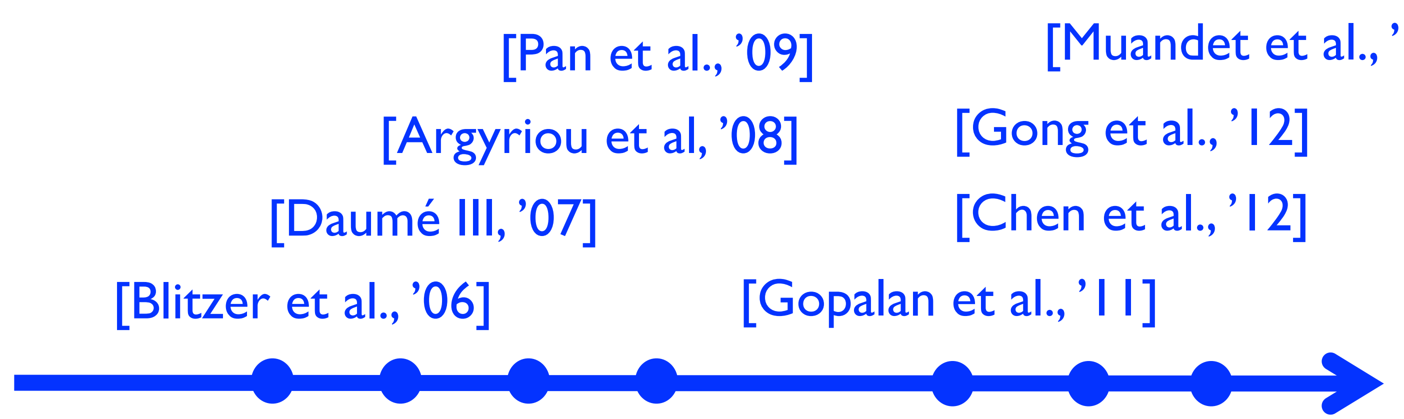


Reverse Gradient Y. Ganin and V. Lempitsky ICML 2015

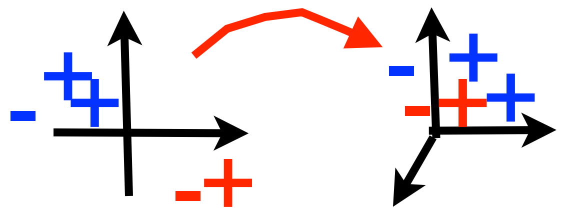
Pros: effective for large inter-domain discrepancy

$$\mathbf{x} \mapsto \mathbf{z}, \quad \text{s.t.}$$

$$P_{\mathcal{S}}(z, y) \approx P_{\mathcal{T}}(z, y)$$



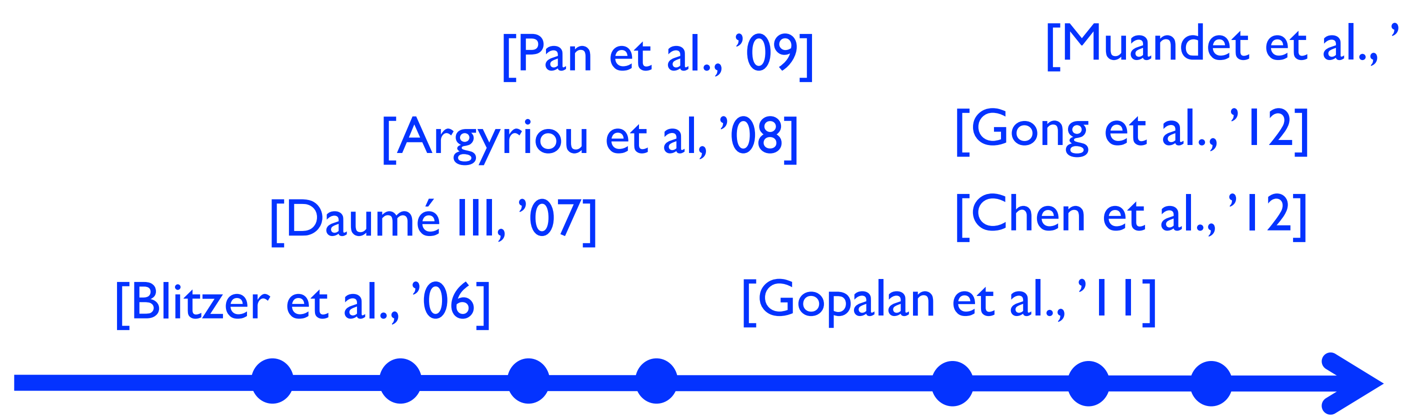
Inferring domain-invariant ***features***



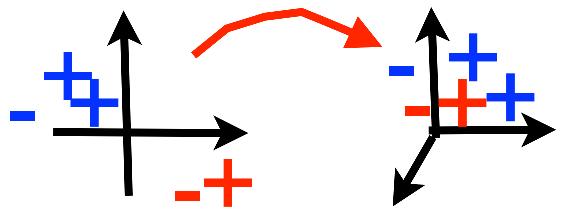
Cons: not discriminative enough for fine-grained tasks

$$\mathbf{x} \mapsto \mathbf{z}, \quad \text{s.t.}$$

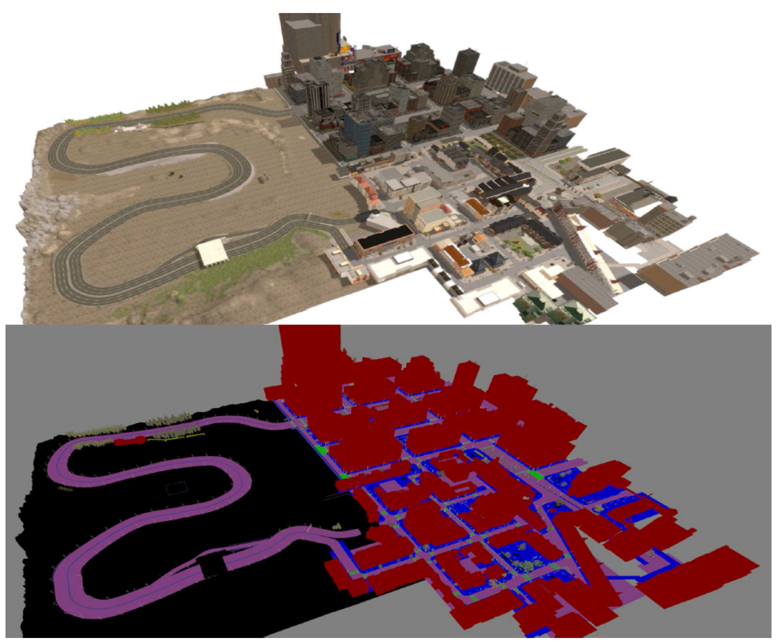
$$P_{\mathcal{S}}(z, y) \approx P_{\mathcal{T}}(z, y)$$

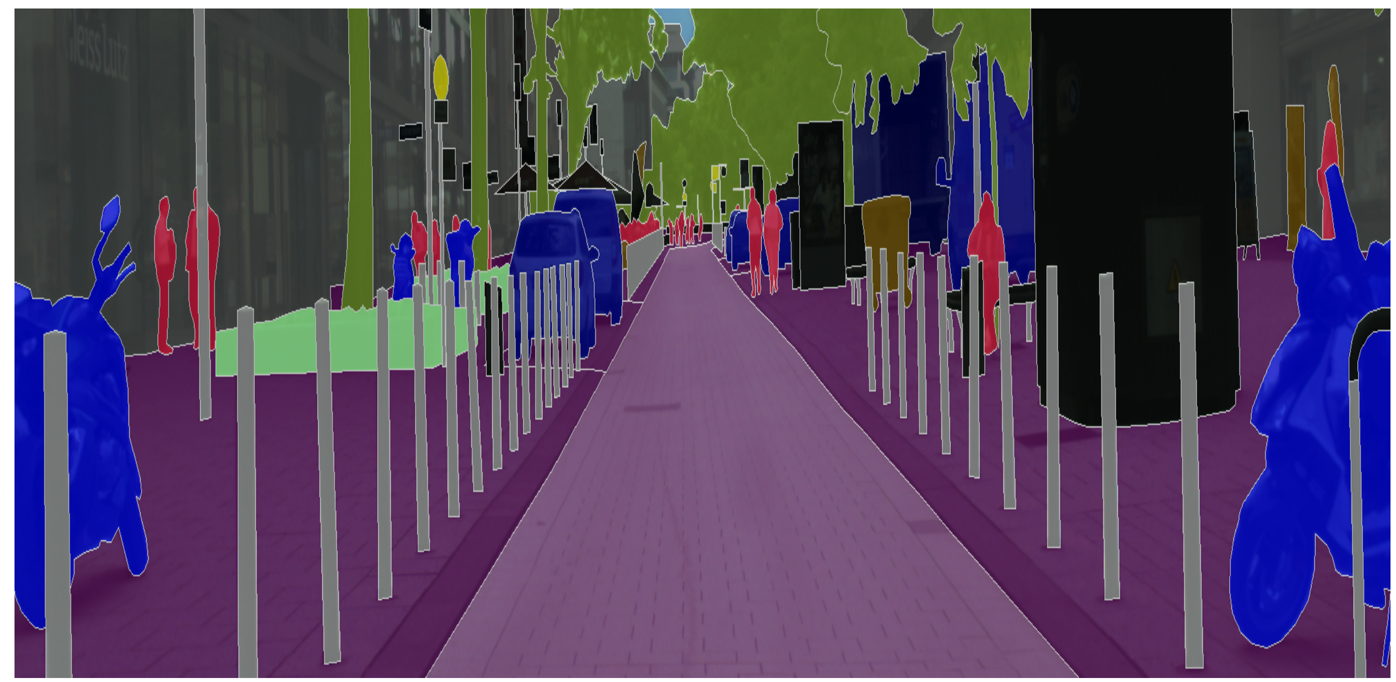


Inferring domain-invariant ***features***



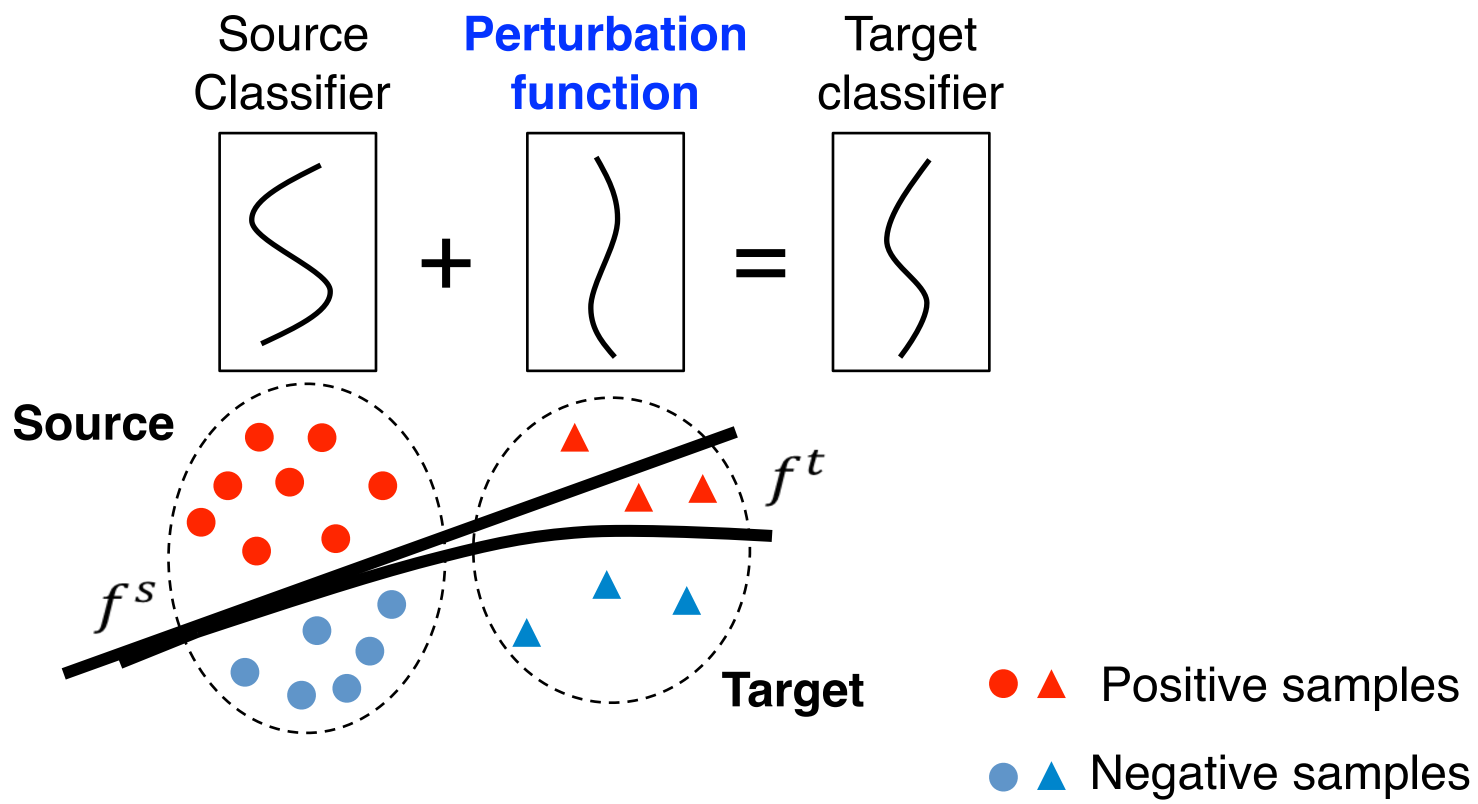
Cons: not discriminative enough for fine-grained tasks





E.g., semantic segmentation

Directly adapt classifiers/models



[Evgeniou and Pontil, ’05]

[Duan et al., ’09]

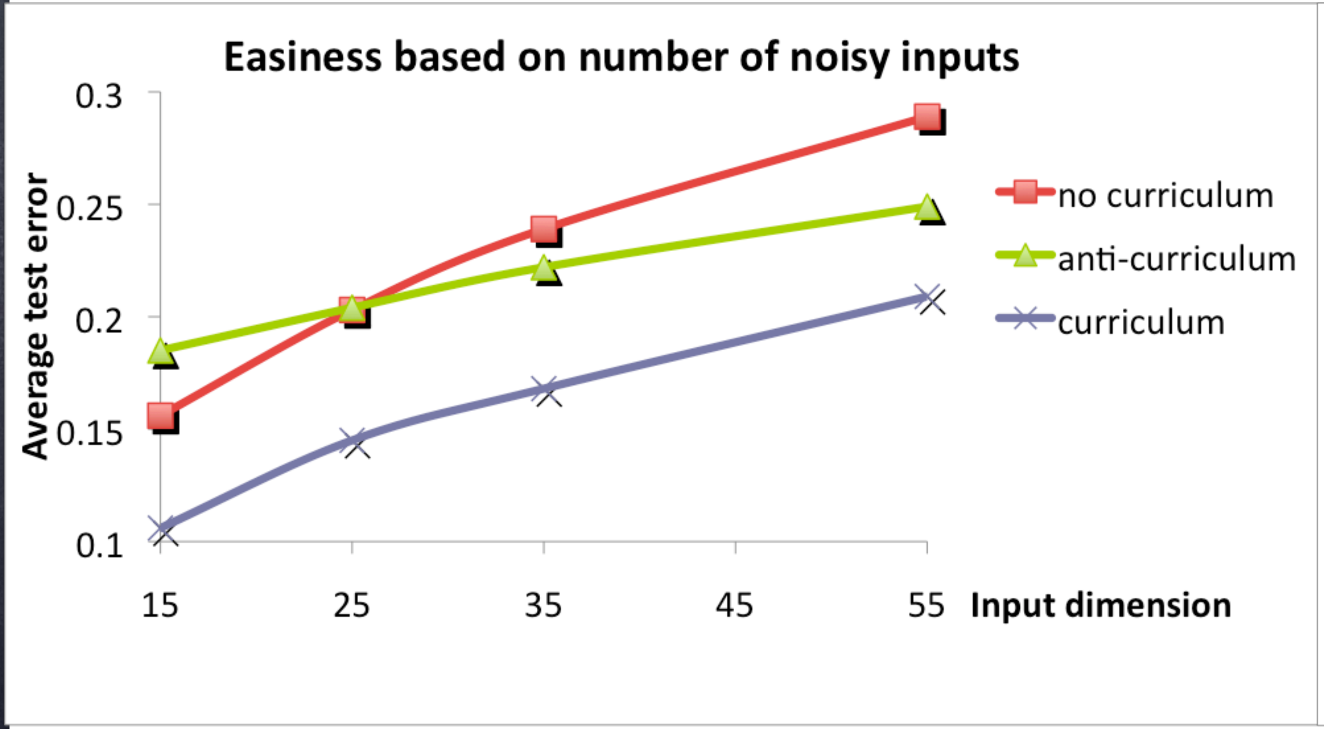
[Duan et al., Daumé III et al., Saenko et al., ’10]

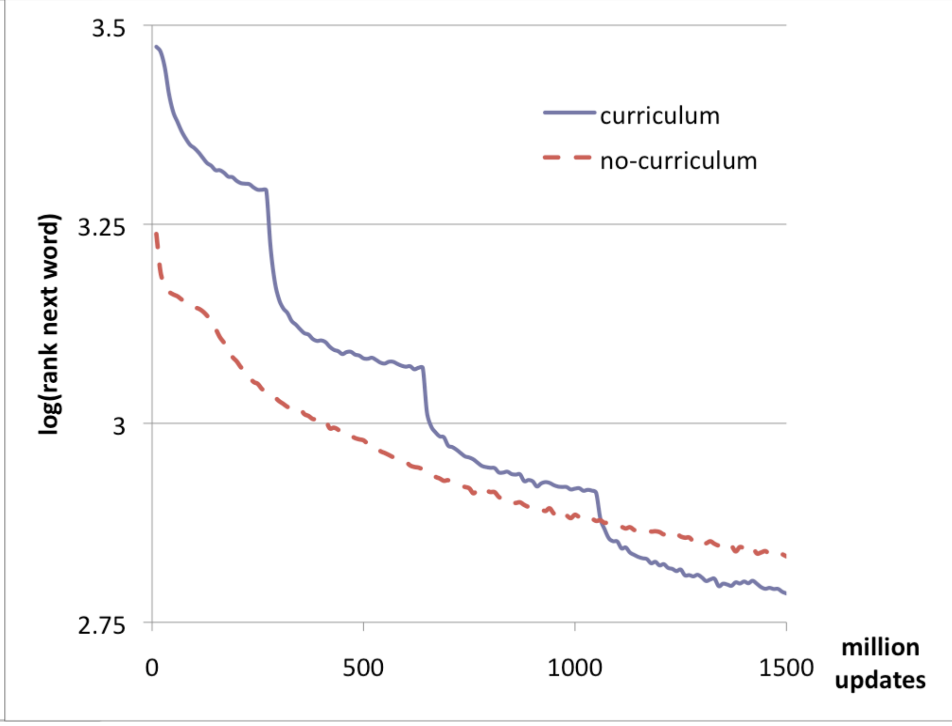
[Kulis et al., Chen et al., ’11]

Adjusting mismatched ***models***

Detour: Curriculum learning

Feed a learning system "easy" **examples** first
Gradually introduce more difficult ones



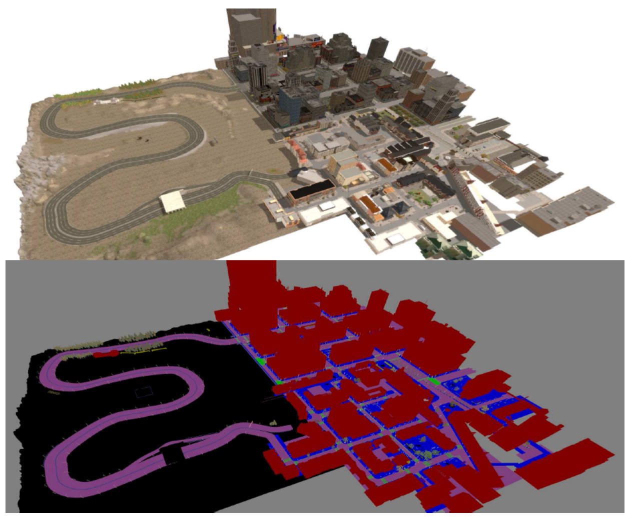


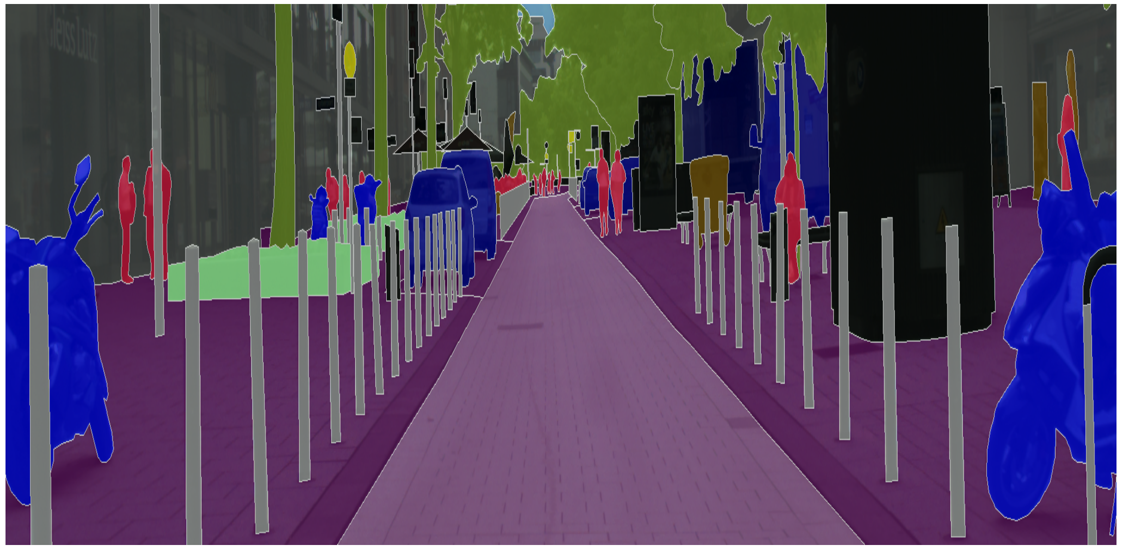
[Bengio et al., ICML'09]

Curriculum domain adaptation

Feed a learning system "easy" **tasks** first

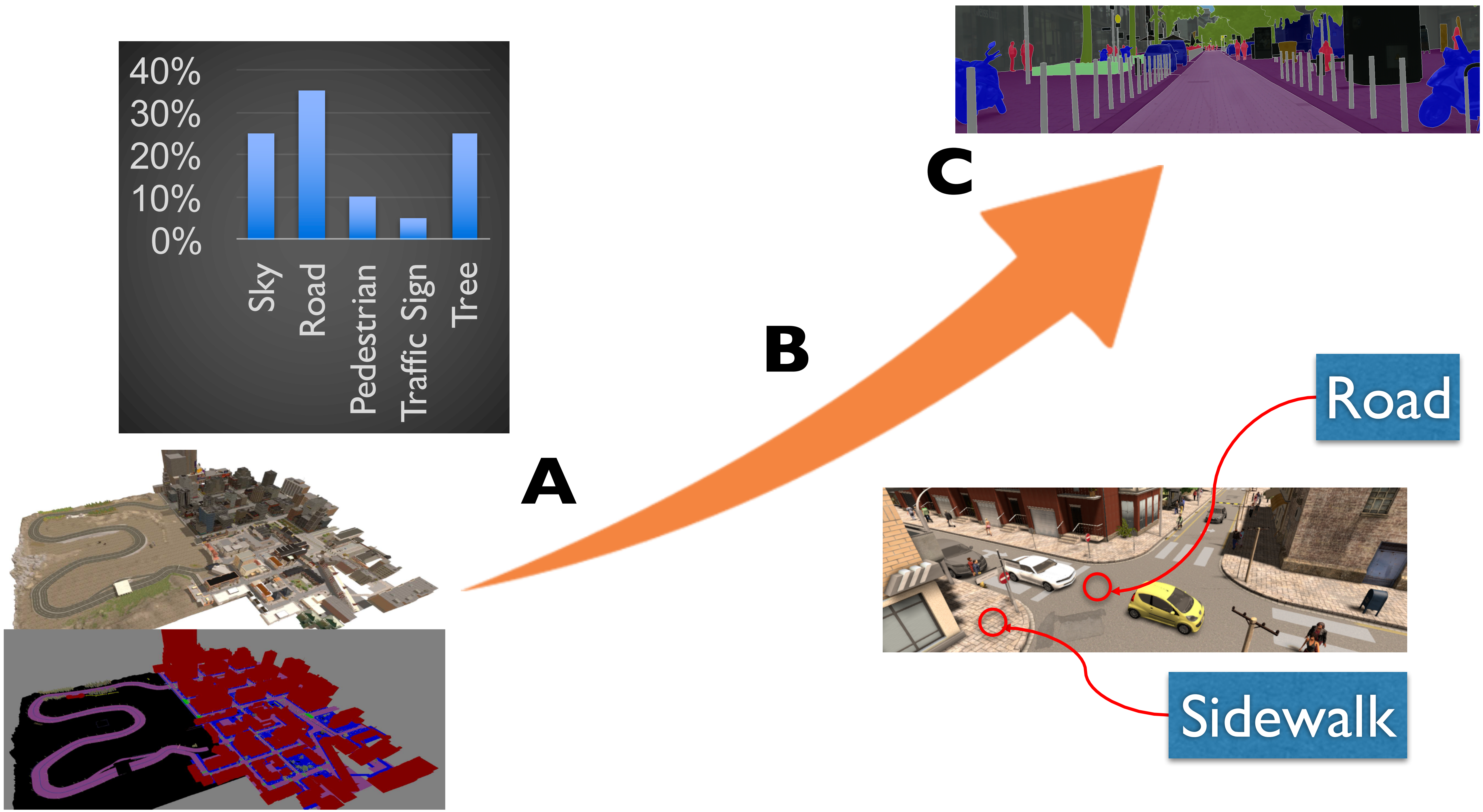
The solutions to them find good local optima, acting as an effective regularizer





Synthetic imagery → Real photos

Curriculum domain adaptation

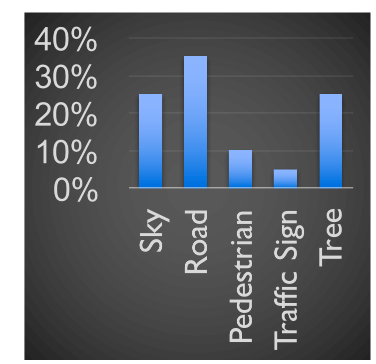


Curriculum domain adaptation *for training CNNs*

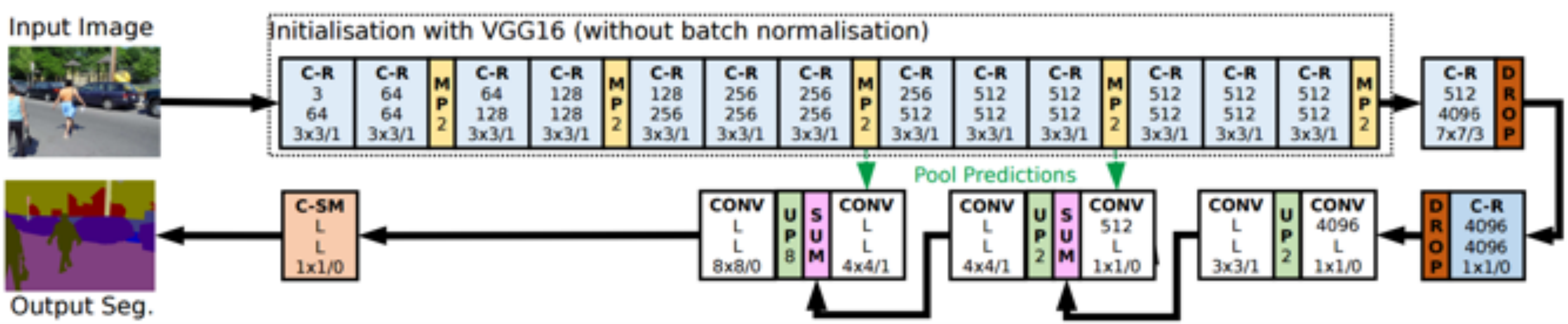
$$\min_{\Theta} \; \mathcal{L}(Y_s, \widehat{Y}_s) + d(p_t, \widehat{p}_t(\widehat{Y}_t))$$

s : Source, $\quad t$: Target

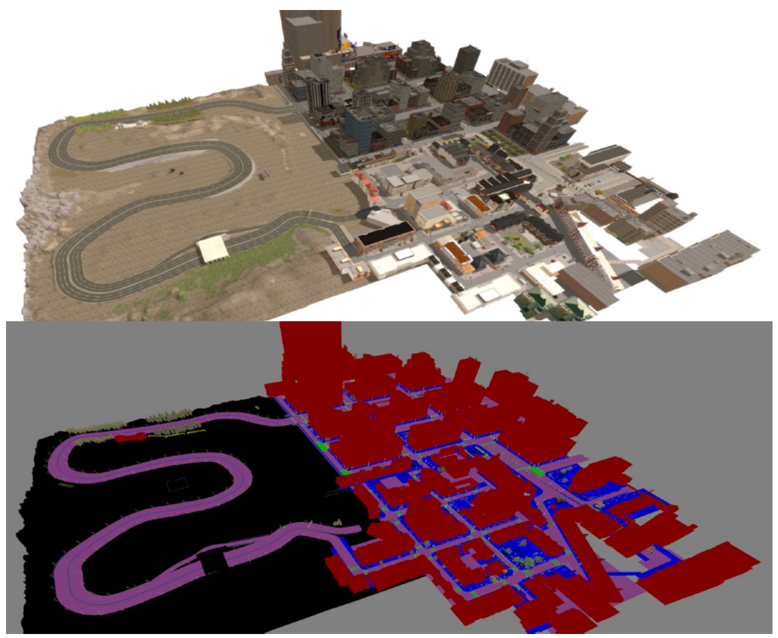
p_t : Perturbation function

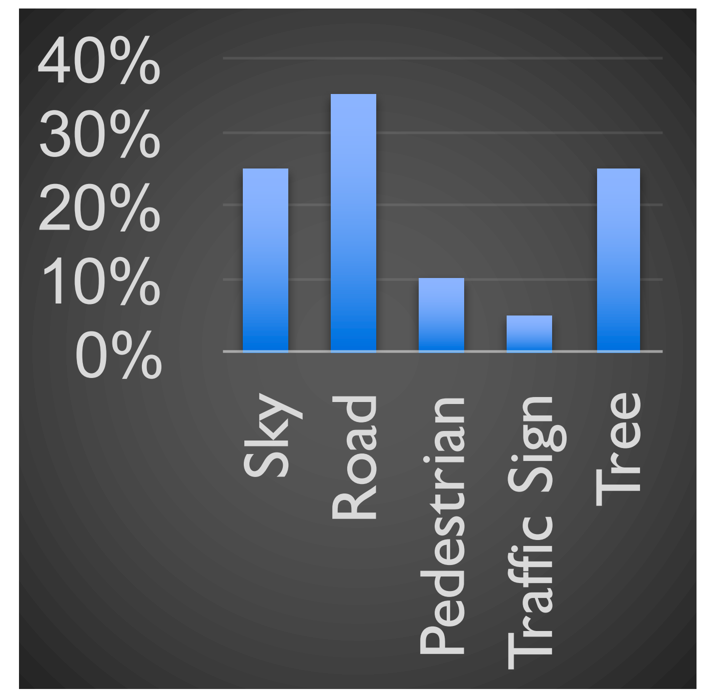


$\widehat{Y}$



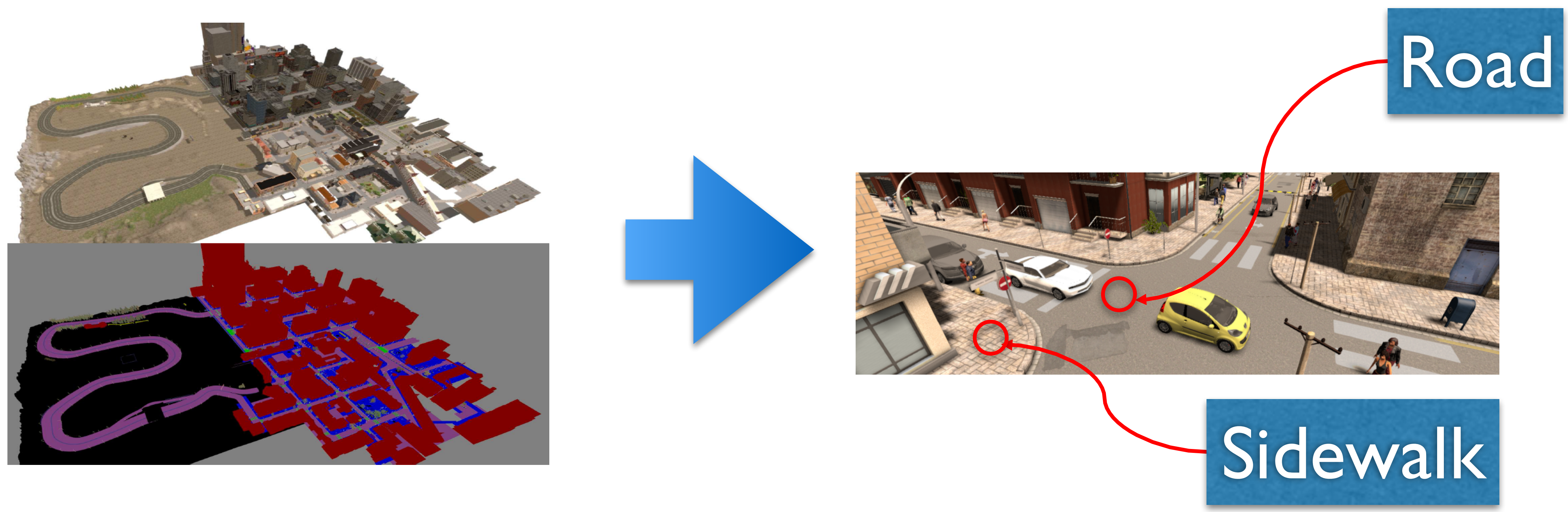
Perturbation functions for semantic segmentation (I)





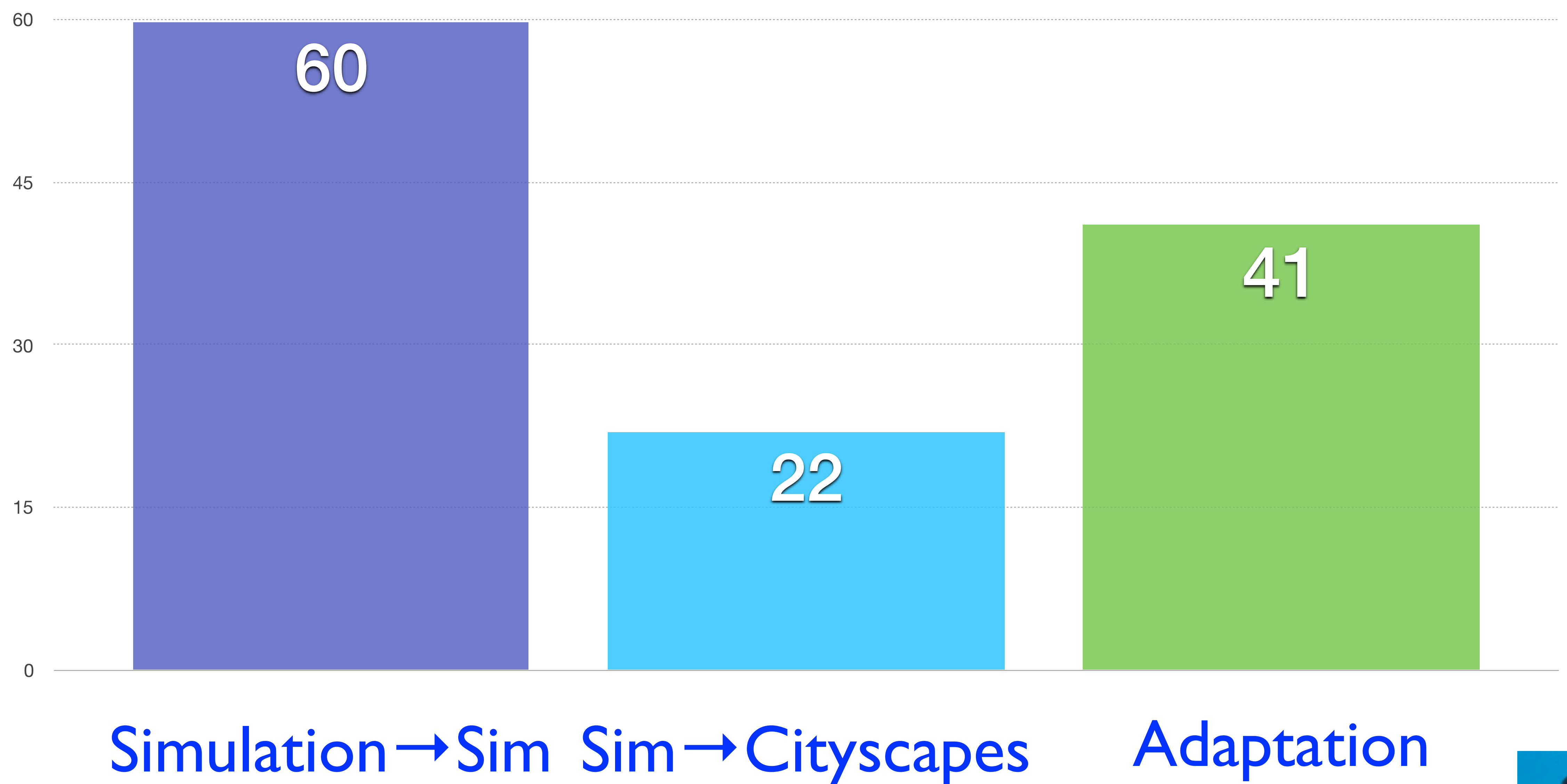
Input: An urban scene image
Algorithm: Logistic regression
Output: Label distributions

Perturbation functions for semantic segmentation (2)

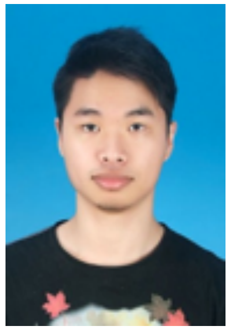


Input: An urban scene image
Algorithm: Super-pixel + Logistic regression
Output: Labels of some super-pixels

Simulation to real world: ~~catastrophic~~ performance drop



[Zhang et al., ICCV'17]



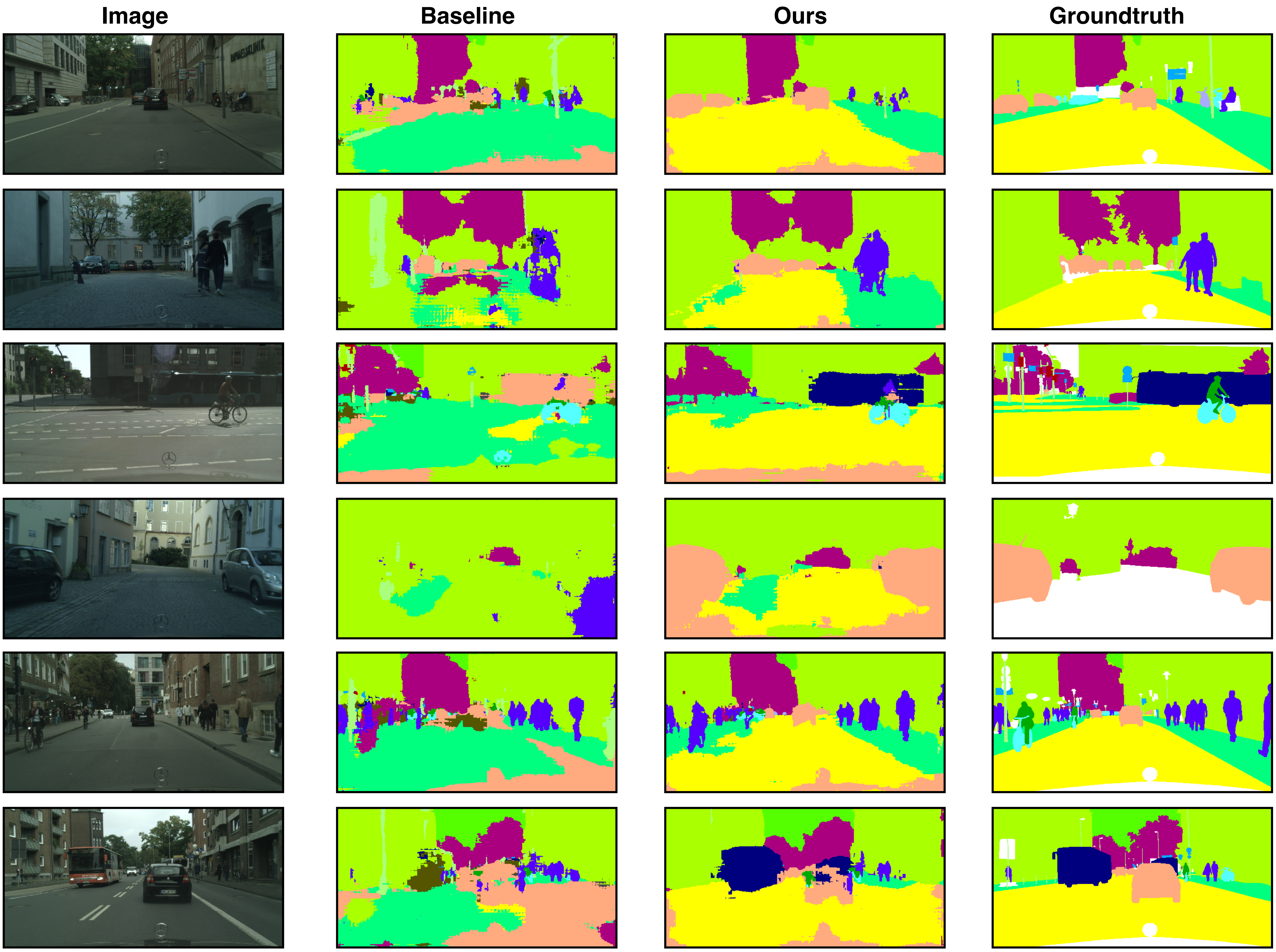
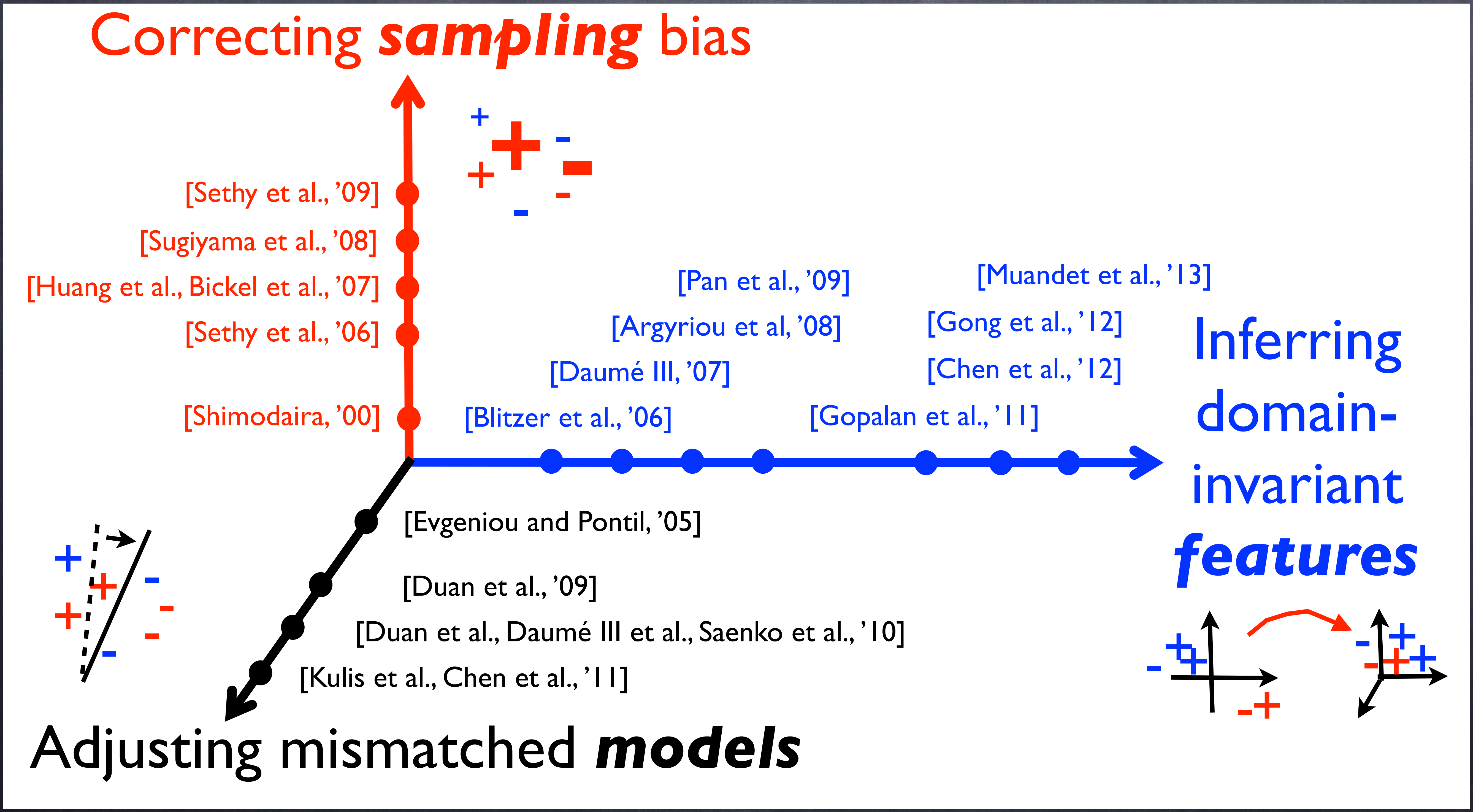
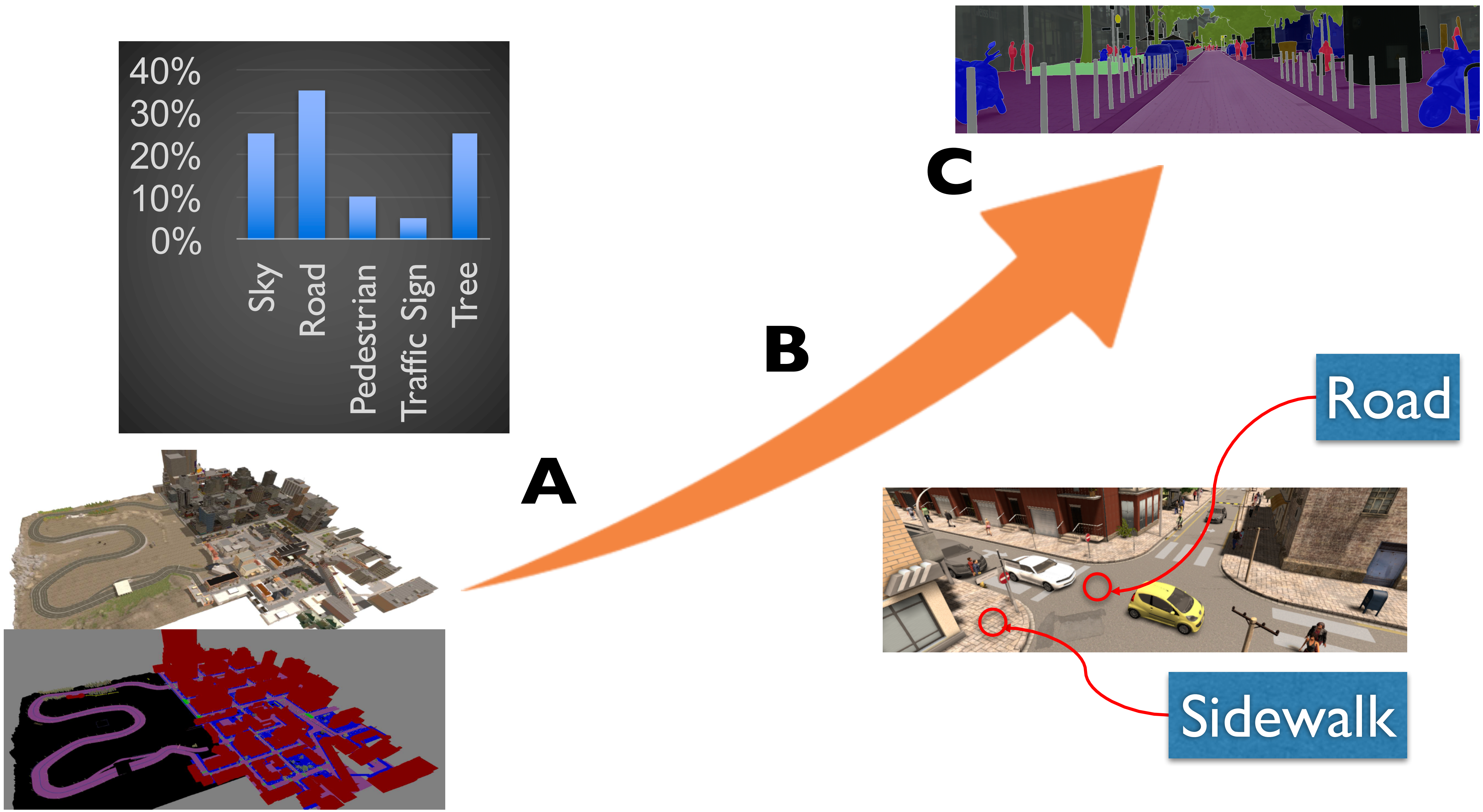


Image
Baseline
Ours
Groundtruth

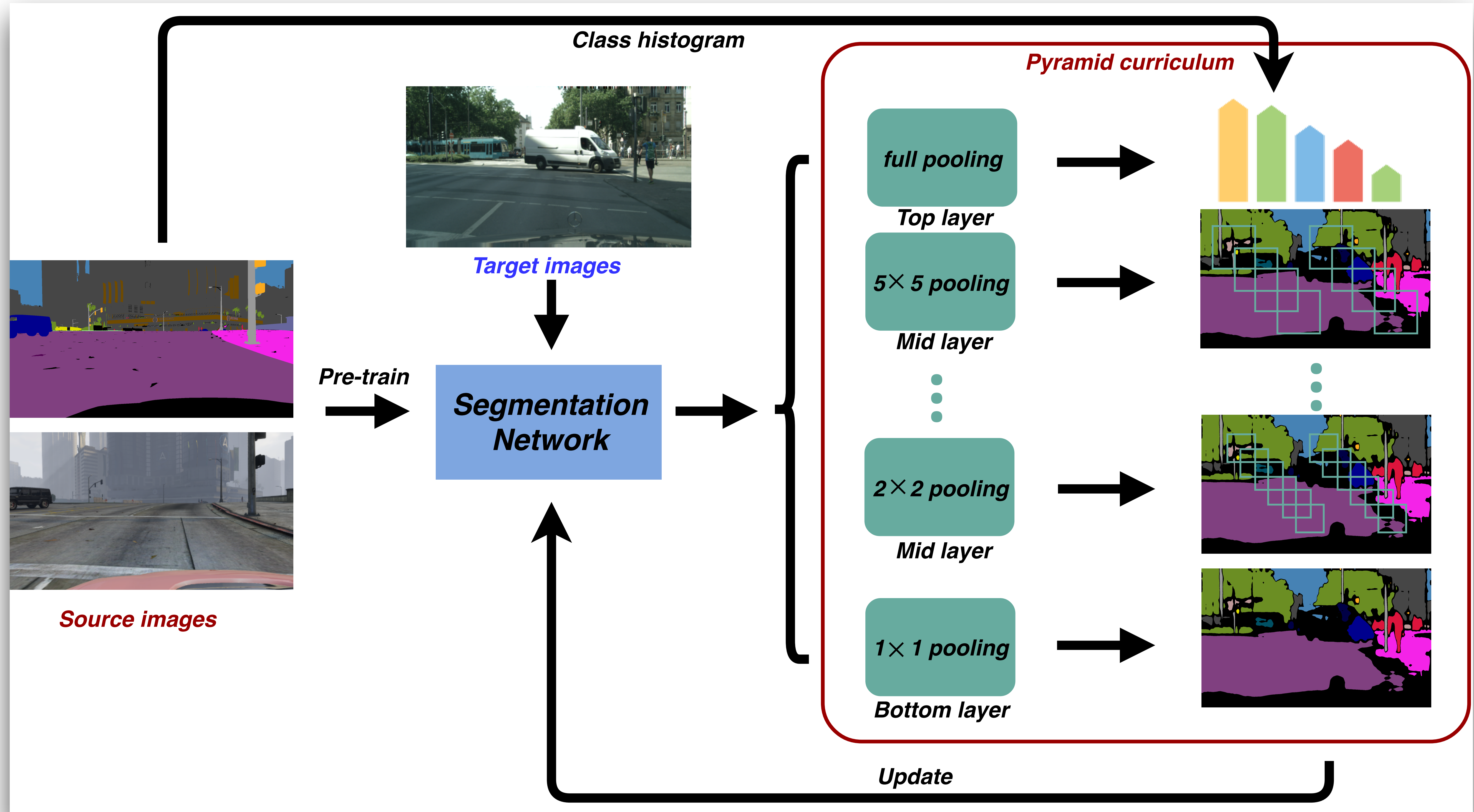
This talk


Correcting *sampling* bias
[Sethy et al., '09]
[Sugiyama et al., '08]
[Huang et al., Bickel et al., '07]
[Sethy et al., '06]
[Shimodaira, '00]
[Pan et al., '09]
[Argyriou et al, '08]
[Daumé III, '07]
[Blitzer et al., '06]
[Muandet et al., '13]
[Gong et al., '12]
[Chen et al., '12]
[Gopalan et al., '11]
Inferring domain-invariant *features*
[Evgeniou and Pontil, '05]
[Duan et al., '09]
[Duan et al., Daumé III et al., Saenko et al., '10]
[Kulis et al., Chen et al., '11]
Adjusting mismatched *models*

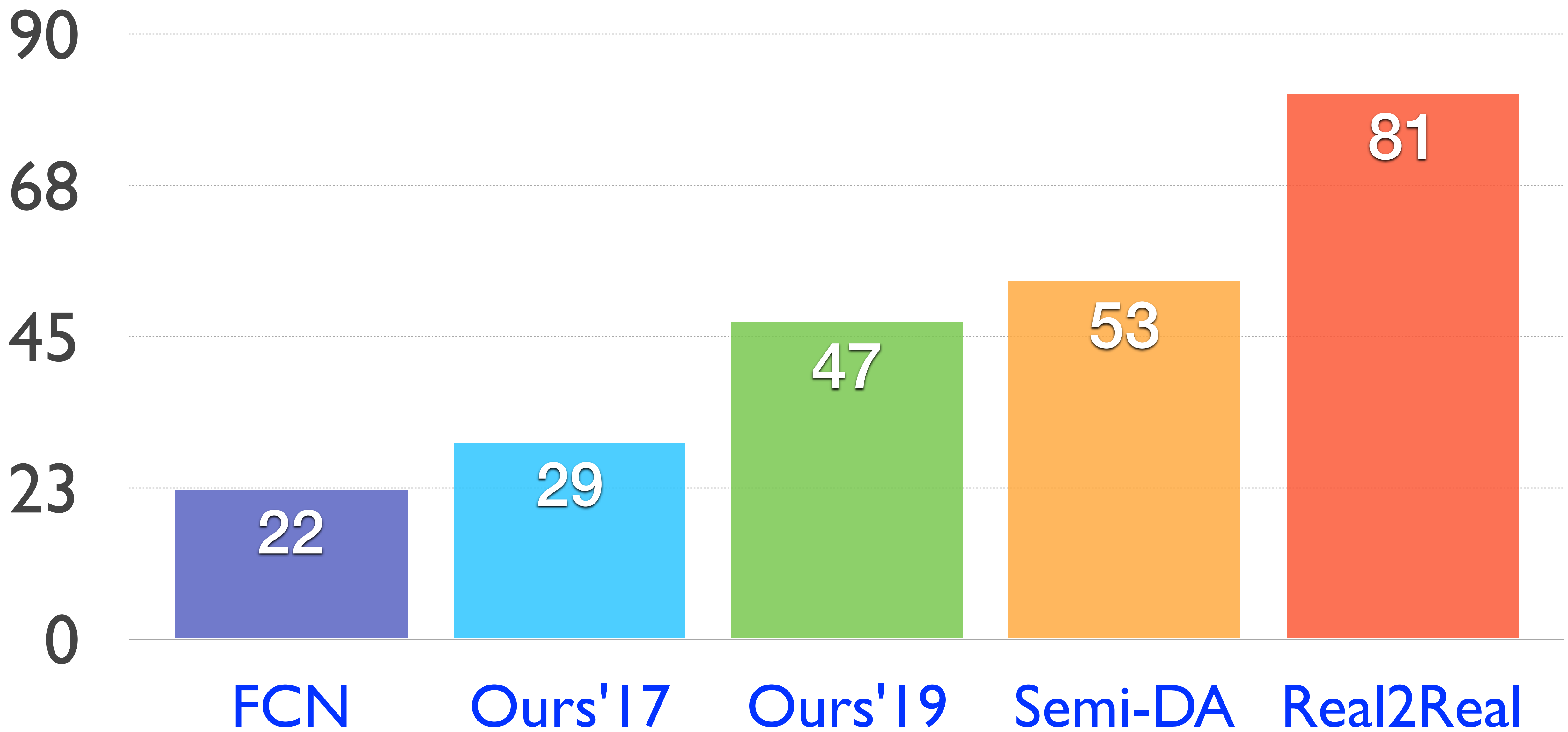
Curriculum domain adaptation



Pyramid Curriculum domain adaptation

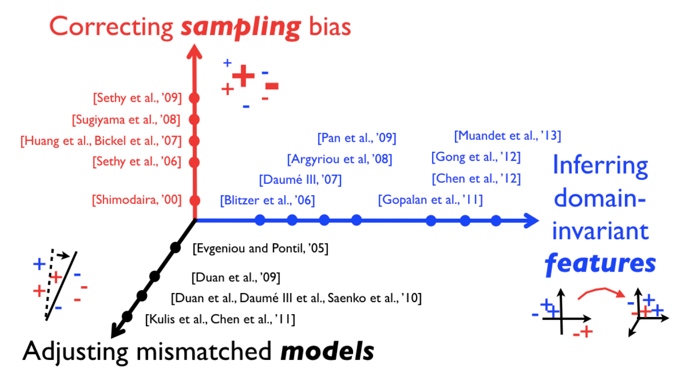


Simulation to real world:
closing the performance gap?

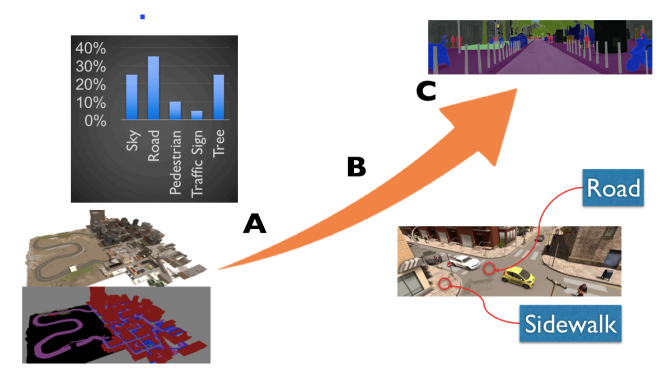


Domain adaptation: key to use simulation "for real"

Domain-invariant **features**
Importance sampling of **data**
Adapt background **models**
etc.

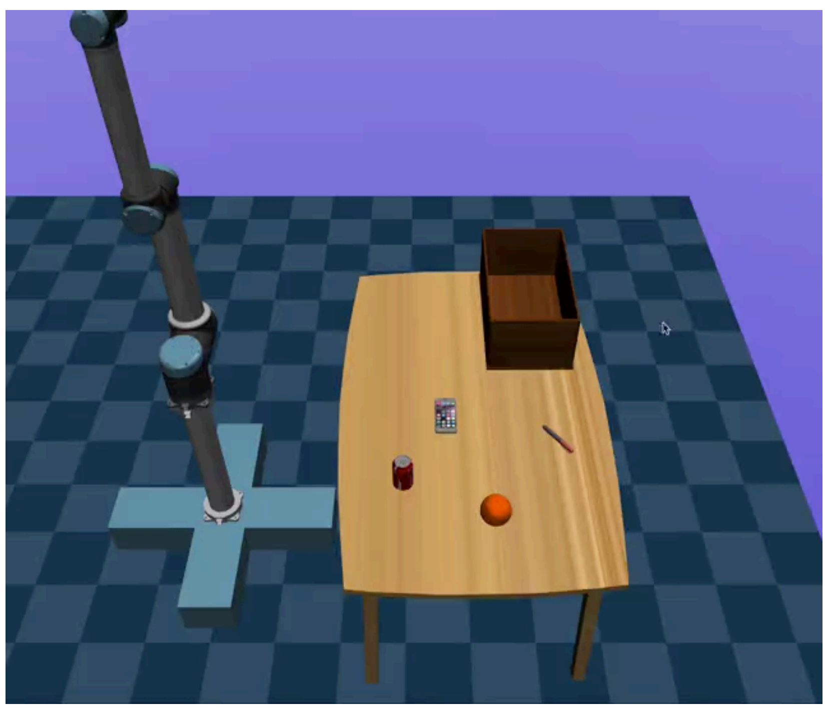


Curriculum domain adaptation
Style transfer, etc.



Simulation to reality for segmentation, detection, dynamics planning & control, etc.

Domain adaptation: key to use simulation “for real”



Simulation to reality for segmentation, detection, Dynamics planning & control, etc.

Acknowledgements

U. Southern California: Fei Sha

U. Texas at Austin: Kristen Grauman

Yang Zhang, Phil David, Qing Lian, and Lixin Duan