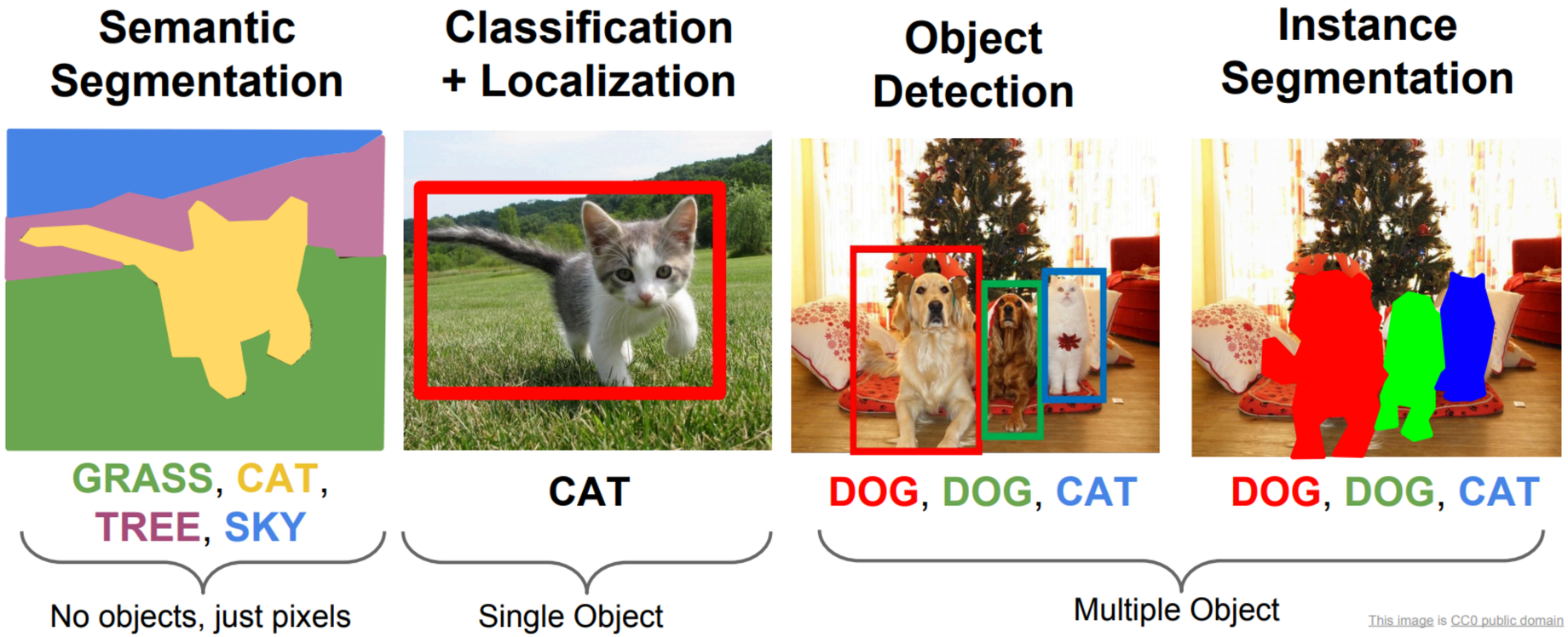
# Learning to Segment Every Thing

Ronghang HU, Piotr Dollar, Kaiming He, Trevor Darrell, Ross Girshick
BAIR, UC Berkeley and Fackbook AI Research (FAIR)

# Instance Segmentation



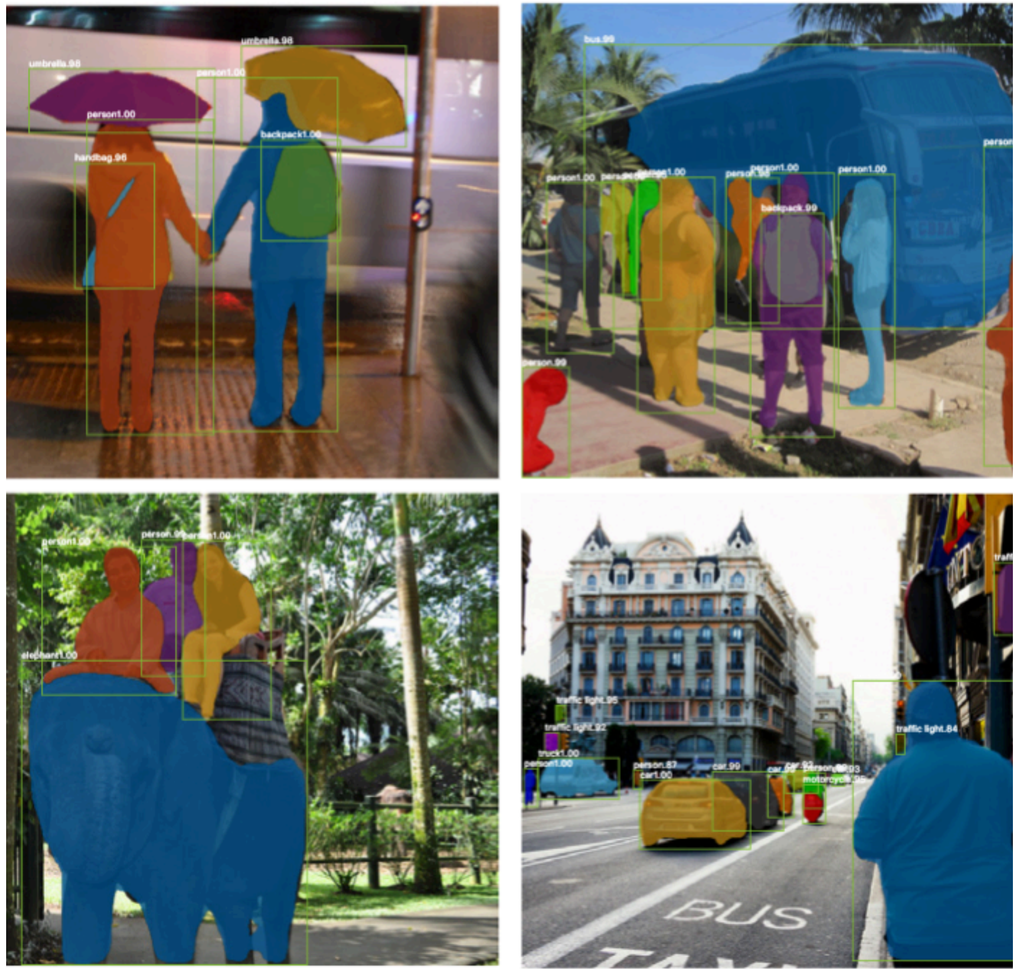

- Most methods for object instance segmentation require **all training examples** to be labeled with **segmentation masks**. (*Strong Supervision*)

- This requirement makes it **expensive** to annotate new categories and has restricted instance models to ~**100 well-annotated classes**.

Credit: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

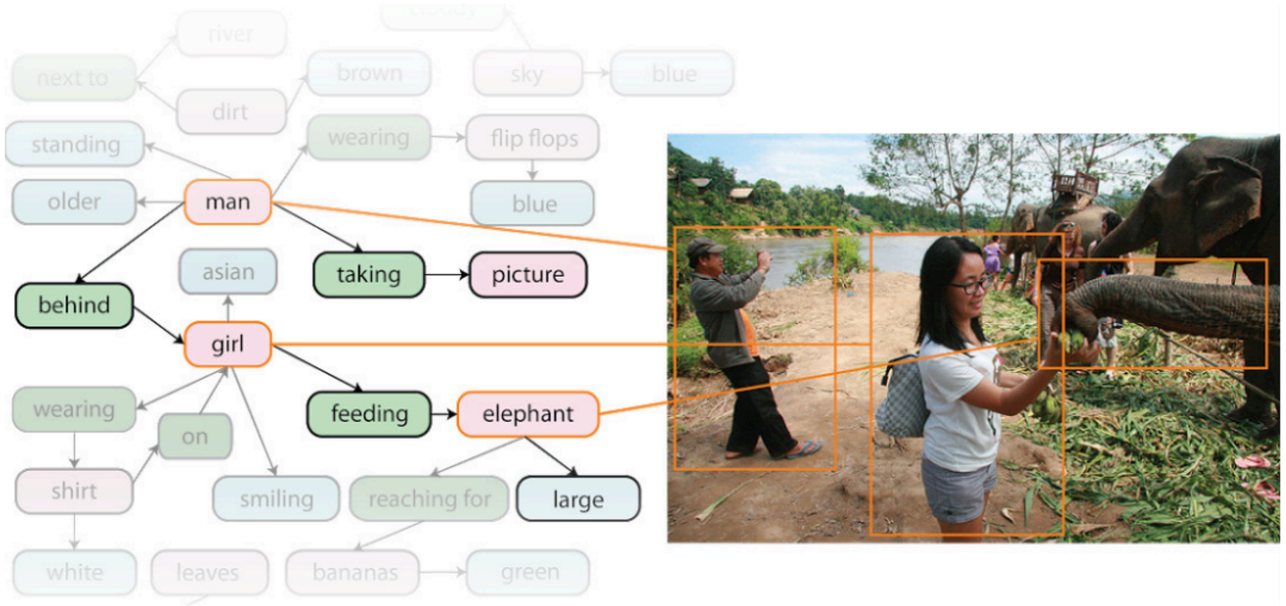
# Dataset: COCO

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints



# Dataset: Visual Genome

- ✓ 108,077 Images
- ✓ 5.4 Million Region Descriptions
- ✓ 1.7 Million Visual Question Answers
- ✓ 3.8 Million Object Instances
- ✓ 2.8 Million Attributes
- ✓ 2.3 Million Relationships
- ✓ Everything Mapped to Wordnet Synsets
- ✓ 3000 classes with bounding box



# Relax strong supervision

- Weakly-supervised training:

  **✓Image-level labels (bounding box)**

  **✓Noisy web labels**

  **✓Scribble-level labels**

  **✓Extreme points**

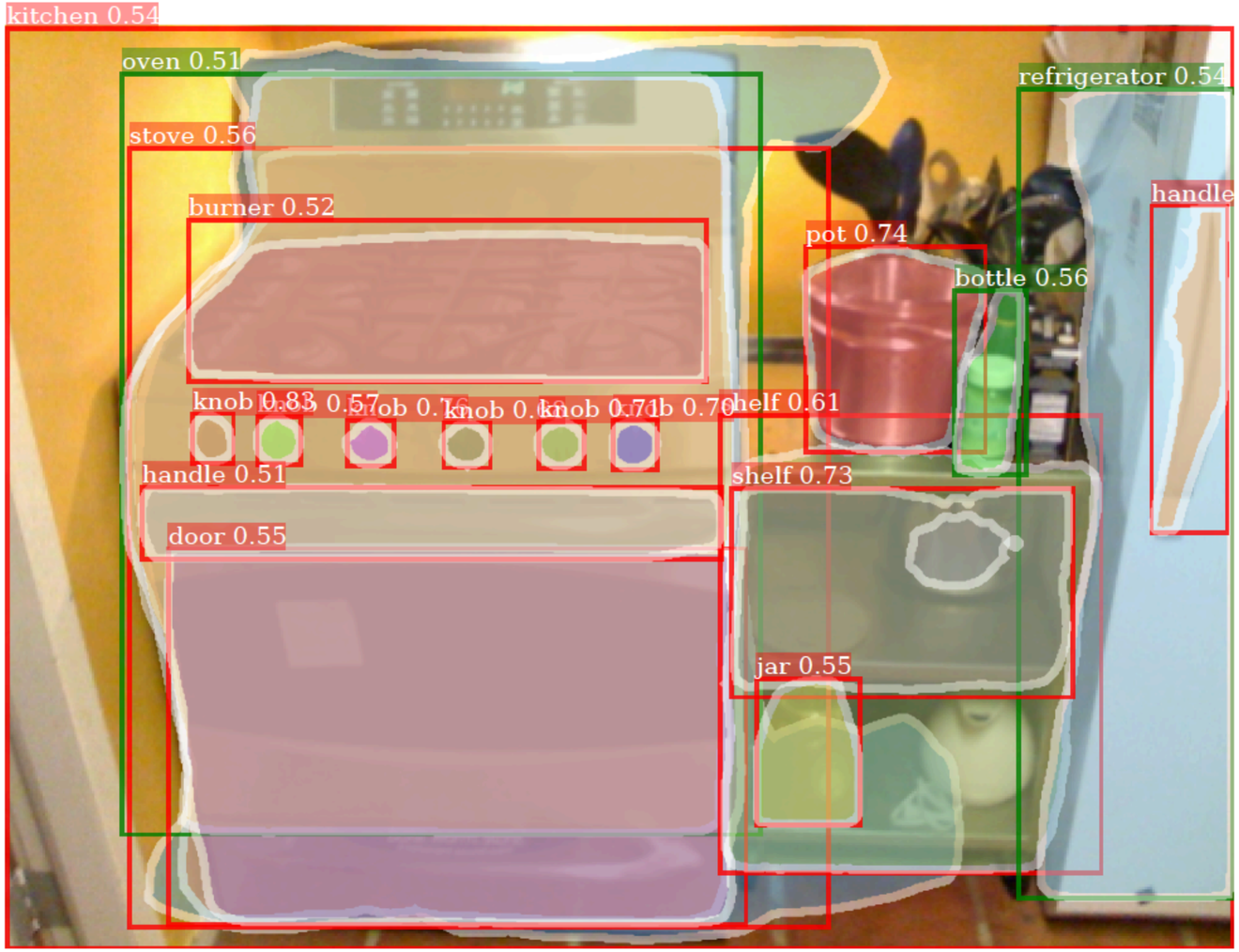
# Relax strong supervision

- **Partially supervised** training (This paper):

  ✓ Given a set of categories of interest, a **small** subset has instance mask annotations, while the other categories have **only** bounding box annotations;

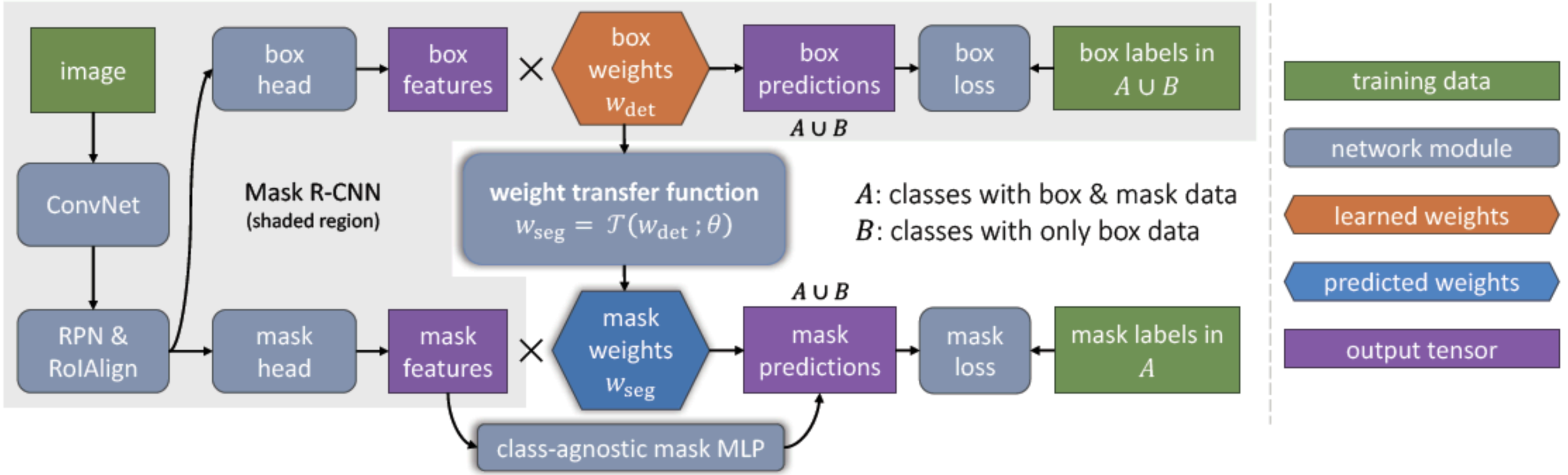
  ✓ the instance segmentation algorithm should utilize this data to fit a model that can segment instances of all object categories in the set of interest.

- The training data is a **mixture of strongly annotated examples (those with masks) and weakly annotated examples (those with only boxes)**, we refer to the task as **partially supervised**.




A subset of classes (green boxes) have instance mask annotations during training; the remaining classes (red boxes) have only bounding box annotations. This image shows output from our model trained for 3000 classes from Visual Genome, using mask annotations from only 80 classes in COCO.

# Mask$^X$ R-CNN




- Instead of directly learning the mask prediction parameters $w_{\text{seg}}$, Mask$^X$ R-CNN predicts a category's segmentation parameters $w_{\text{seg}}$ from its corresponding box detection parameters $w_{\text{det}}$, using a learned weight transfer function *T* . For training, *T* only needs mask data for the classes in set A, yet it can be applied to all classes in set A U B at test time.

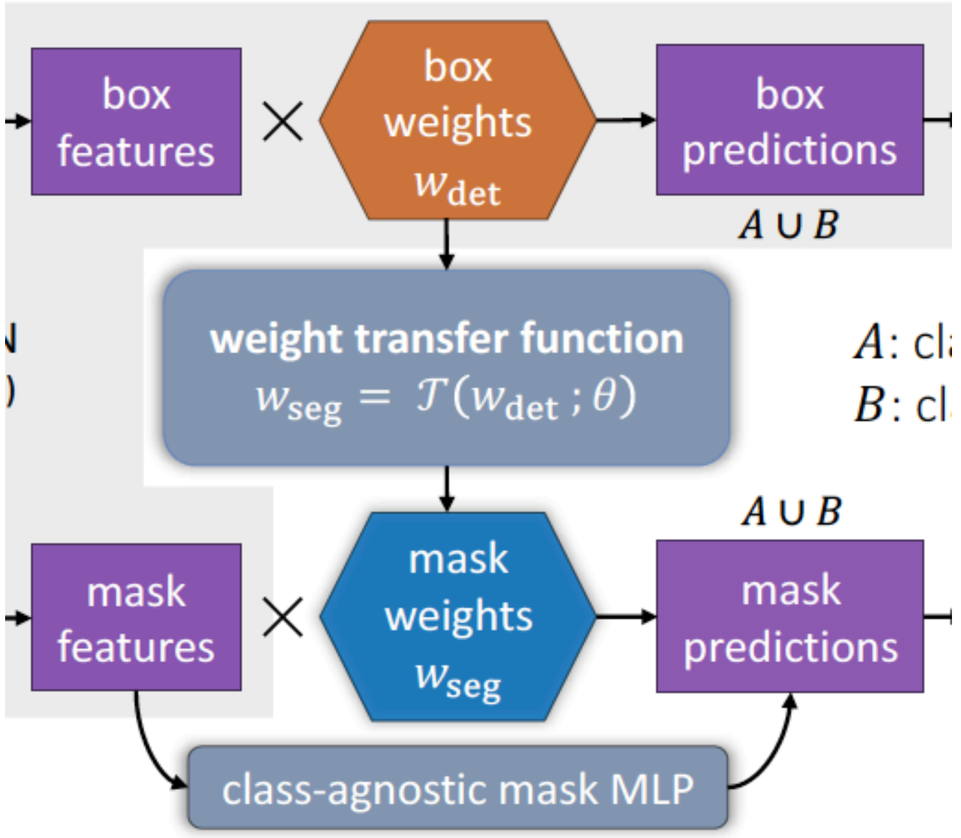
# Mask Prediction Using Weight Transfer

- In Mask R-CNN, **the last layer in the bounding box branch** and **the last layer in the mask branch** both contain **category-specific parameters** that are used to perform bounding box classification and instance mask prediction, respectively, for each category.

- Instead of learning the category-specific bounding box parameters and mask parameters **independently**, we propose to predict a category's mask parameters from its bounding box parameters using a generic, category-agnostic weight transfer function that can be **jointly** trained as part of the whole model.

# Weight Transfer

- For a given category c
- $w_{det}^{c}$:The class-specific object detection weights in the last layer of bounding box head
- $w_{seg}^{c}$ :The class-specific mask weights in the mask head
- Weight transfer: theta is class-agnostic, learned parameter
- $$w_{seg}^{c} = \mathcal{T}(w_{det}^{c}; \theta)$$



# Details of Transfer Function

- Transfer function can be implemented as a **small fully-connected** neural network.

- Two types of detection weights:

  ✓**The ROI classification weights**

  ✓**The bounding box regression weights**

  ✓**Or concatenation of these two type of weights**

# Training

- Stage-wise Training

  ✓ In the first stage, we train a Faster R-CNN using only the bounding box annotations of the classes in A U B

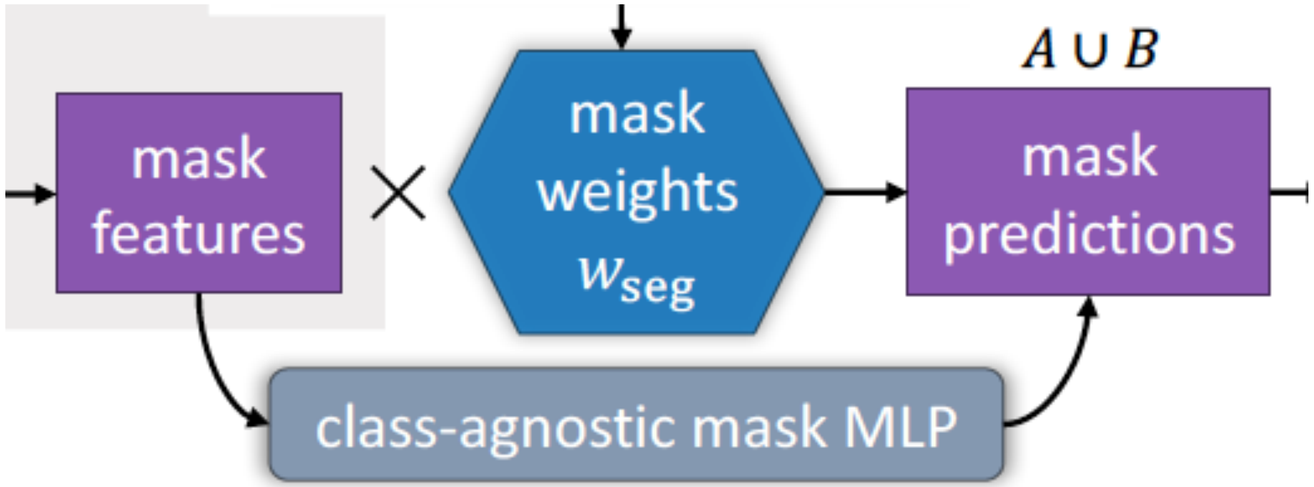
  ✓ and then in the second stage the additional mask head is trained while keeping the convolutional features and the bounding box head fixed

- End-to-end joint training

  ✓ In principle, one can directly train with backpropagation using the box losses on classes in A U B and the mask loss on classes in A.

  ✓ Address the discrepancy in class-specific, stop the gradient w.r.t. w_det.

# Fused FCN and MLP




- Intuitively, the MLP mask predictor may better capture the ‘gist’ of an object while the FCN mask predictor may better capture the details (such as the object boundary).

# Two experiments

- **Split the COCO dataset into sets A and B**
- **COCO dataset as A, and Visual Genome as B**

# Ablation Exp: Input to *T*

| method | voc → non-voc | | non-voc → voc | |
|---|---|---|---|---|
| | AP on $B$ | AP on $A$ | AP on $B$ | AP on $A$ |
| transfer w/ randn | 15.4 | 35.2 | 19.9 | 31.1 |
| transfer w/ GloVe [31] | 17.3 | 35.2 | 21.9 | 31.1 |
| transfer w/ cls | 18.1 | 35.1 | 25.2 | 31.1 |
| transfer w/ box | 19.8 | 35.2 | 25.7 | 31.1 |
| transfer w/ cls+box | **20.2** | 35.2 | **26.0** | 31.2 |
| class-agnostic (baseline) | 14.2 | 34.4 | 21.5 | 30.7 |
| fully supervised (oracle) | 30.7 | 35.0 | 35.0 | 30.7 |

- **randn**: a random Gaussian vector (‘randn’) assigned to each class
- **GloVe**: an NLP-based word embedding using pretrained GloVe vectors for each class
- **cls**: the weights from the Mask R-CNN box head classifier
- **box**: the weights from the box regression
- **cls+box**: the concatenation of both weights (‘cls+box’).
- **voc**: the same 20 categories as the PASCAL VOC.

# Ablation Exp: Structure of *T*

| method | voc → non-voc | | non-voc → voc | |
|---|---|---|---|---|
| | AP on *B* | AP on *A* | AP on *B* | AP on *A* |
| transfer w/ 1-layer, none | 19.2 | 35.2 | 25.3 | 31.2 |
| transfer w/ 2-layer, ReLU | 19.7 | 35.3 | 25.1 | 31.1 |
| transfer w/ 2-layer, LeakyReLU | **20.2** | 35.2 | **26.0** | 31.2 |
| transfer w/ 3-layer, ReLU | 19.3 | 35.2 | **26.0** | 31.1 |
| transfer w/ 3-layer, LeakyReLU | 18.9 | 35.2 | 25.5 | 31.1 |

- as a **simple affine transformation**, or as **a neural network with 2 or 3 layers** or **different activation functions**
- A **2-layer MLP with LeakyReLU** gives the best mask AP on set B
- We select the '**cls+box, 2-layer, LeakyReLU**' implementation of *T* for all subsequent experiments

# Ablation Exp: Impact of MLP mask

| method | voc → non-voc | | non-voc → voc | |
|---|---|---|---|---|
| | AP on $B$ | AP on $A$ | AP on $B$ | AP on $A$ |
| class-agnostic | 14.2 | 34.4 | 21.5 | 30.7 |
| class-agnostic+MLP | 17.1 | 35.1 | 22.8 | 31.3 |
| transfer | 20.2 | 35.2 | 26.0 | 31.2 |
| transfer+MLP | **21.3** | 35.4 | **26.6** | 31.4 |

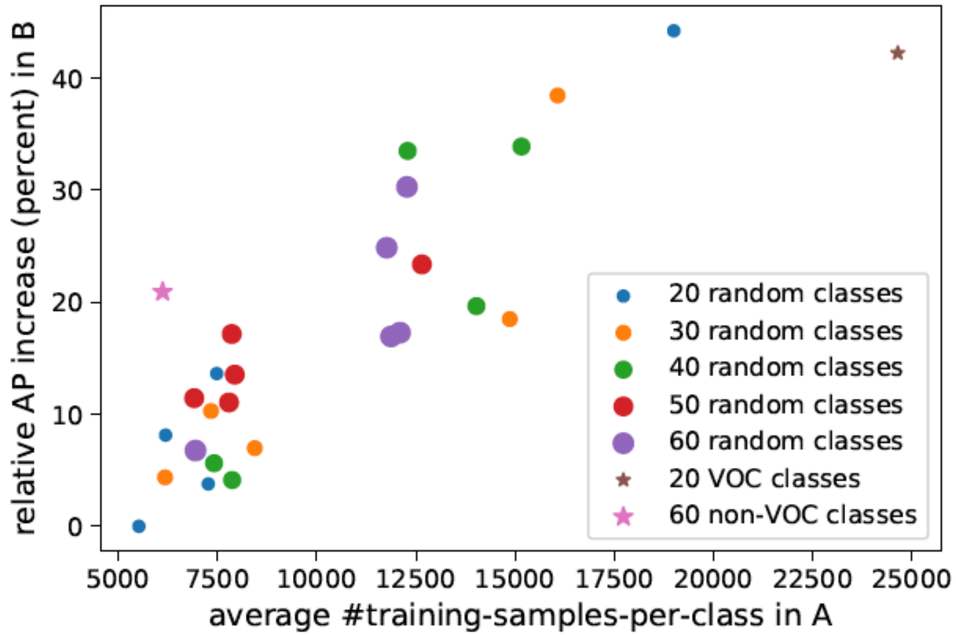
- Either mask head fused with the MLP mask branch **consistently** outperforms the corresponding unfused version. This confirms our intuition that FCN-based mask heads and MLP-based mask heads are complementary in nature.

# Ablation Exp: Effect of end-to-end

| method | training | stop grad on $w_{\text{det}}$ | voc → non-voc AP on $B$ | voc → non-voc AP on $A$ | non-voc → voc AP on $B$ | non-voc → voc AP on $A$ |
|---|---|---|---|---|---|---|
| class-agnostic | sw | n/a | 14.2 | 34.4 | 21.5 | 30.7 |
| transfer | sw | n/a | 20.2 | 35.2 | 26.0 | 31.2 |
| class-agnostic | e2e | n/a | 19.2 | 36.8 | 23.9 | 32.5 |
| transfer | e2e | | 20.2 | 37.7 | 24.8 | 33.2 |
| transfer | e2e | ✓ | **22.2** | 37.6 | **27.6** | 33.1 |

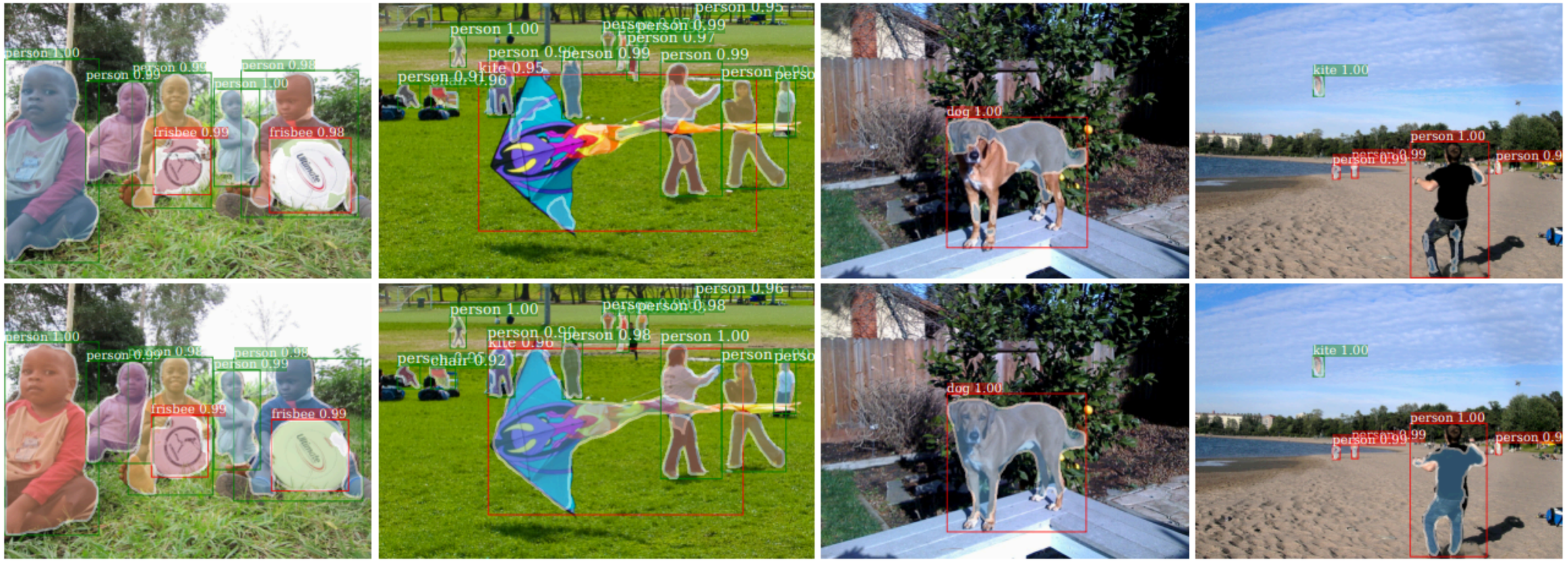
- End-to-end training can bring improved results, however only when backpropagation from *T* to $w_{\text{det}}$ is disabled .

# Ablation Exp:  Random A /B  splits

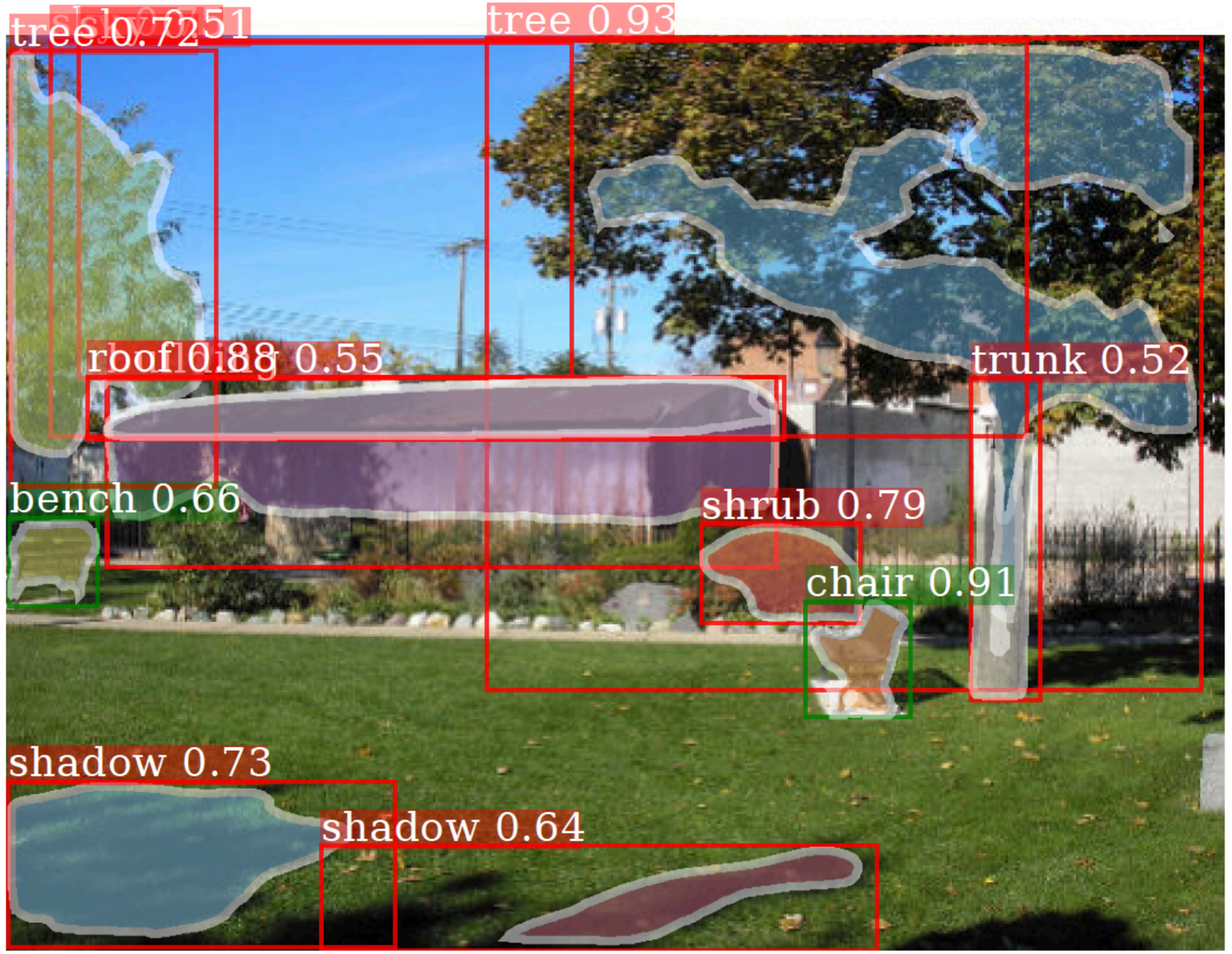



- This indicates that to maximize transfer performance to classes in set B it may be more effective to collect a larger number of instance mask samples for each object category in set A .

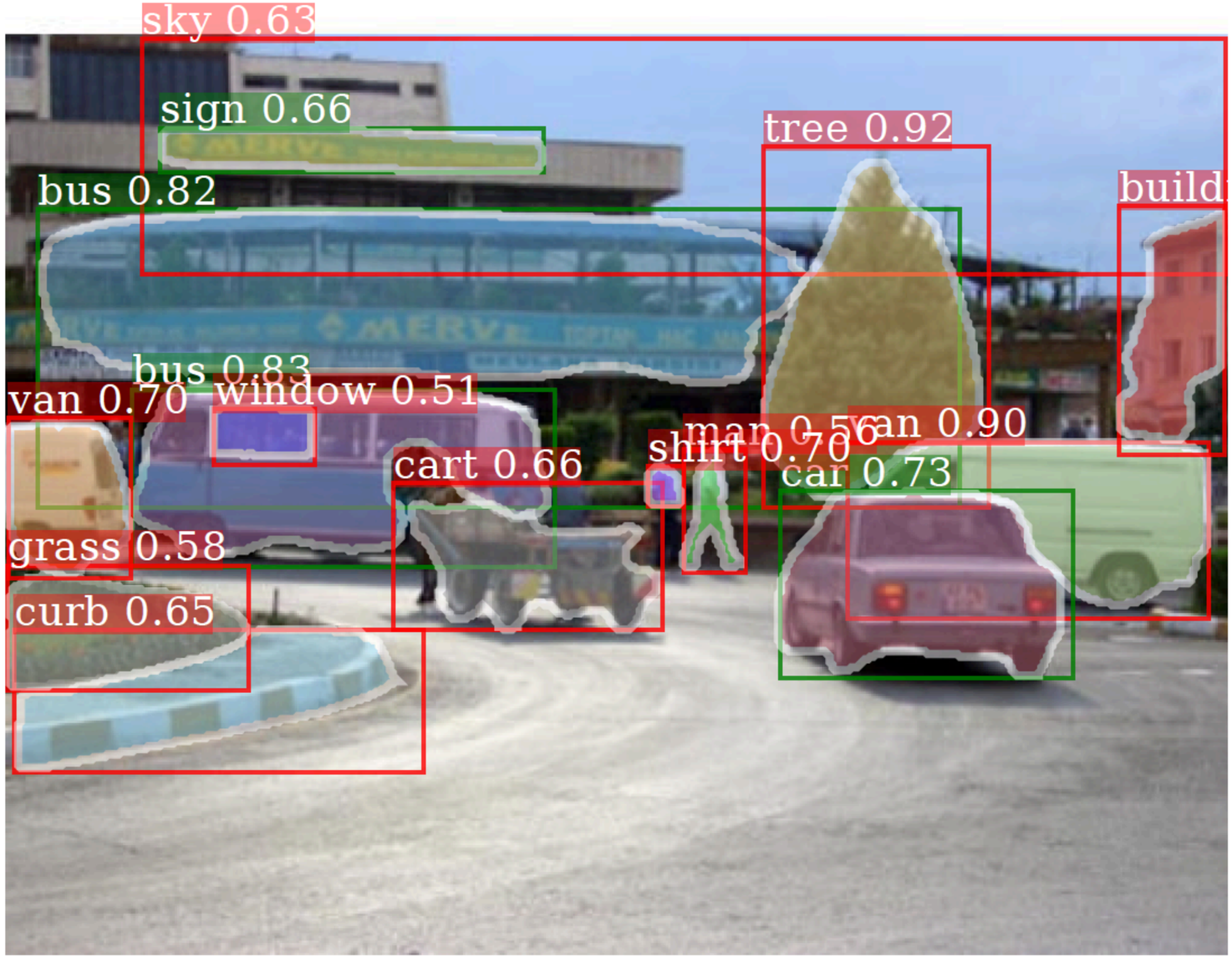
# Class-agnostic baseline v.s. Mask$^X$ R-CNN



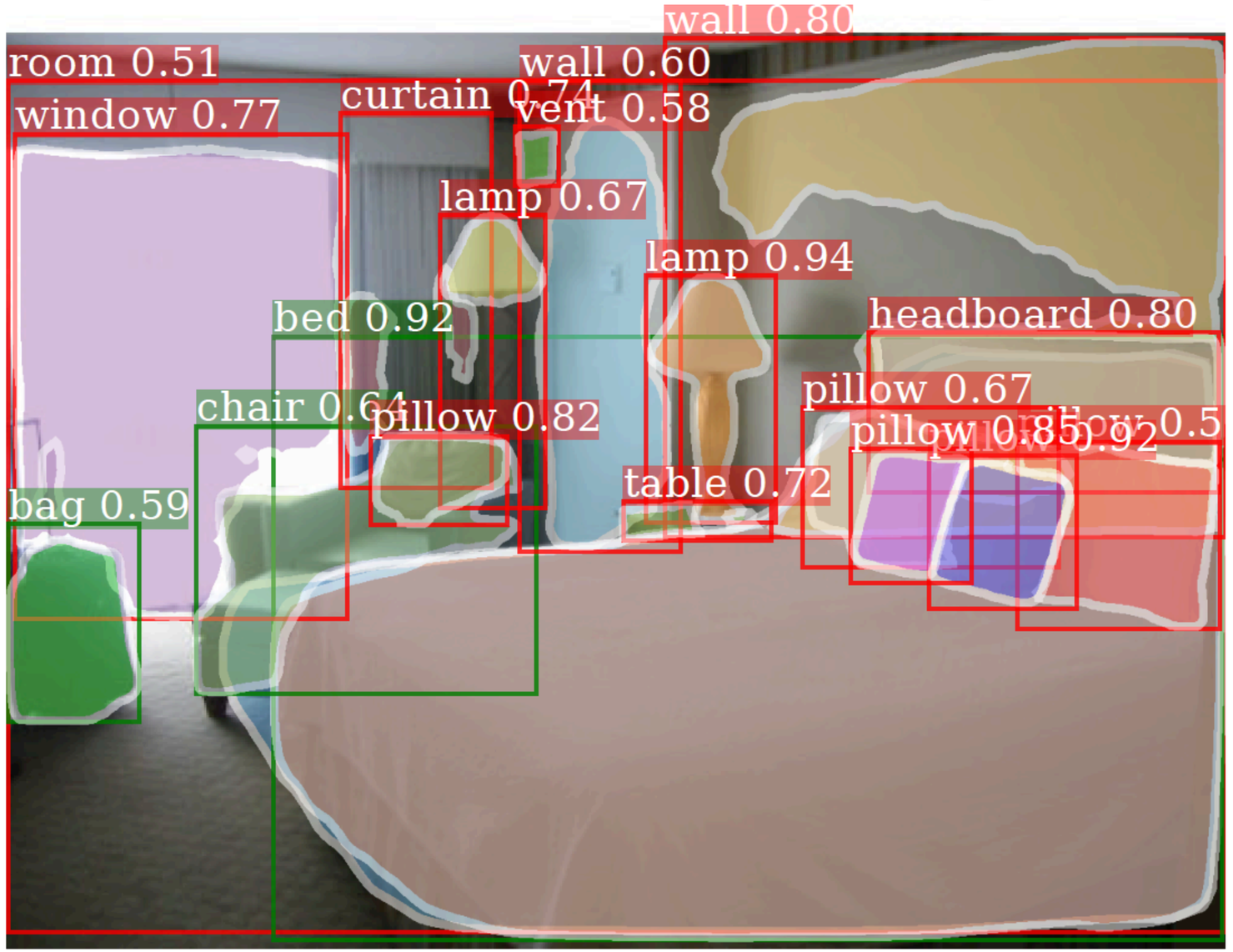
# Large-Scale Instance Segmentation



# Large-Scale Instance Segmentation



# Large-Scale Instance Segmentation



# Large-Scale Instance Segmentation