Multi-Task, Adaptation, Transfer Learning

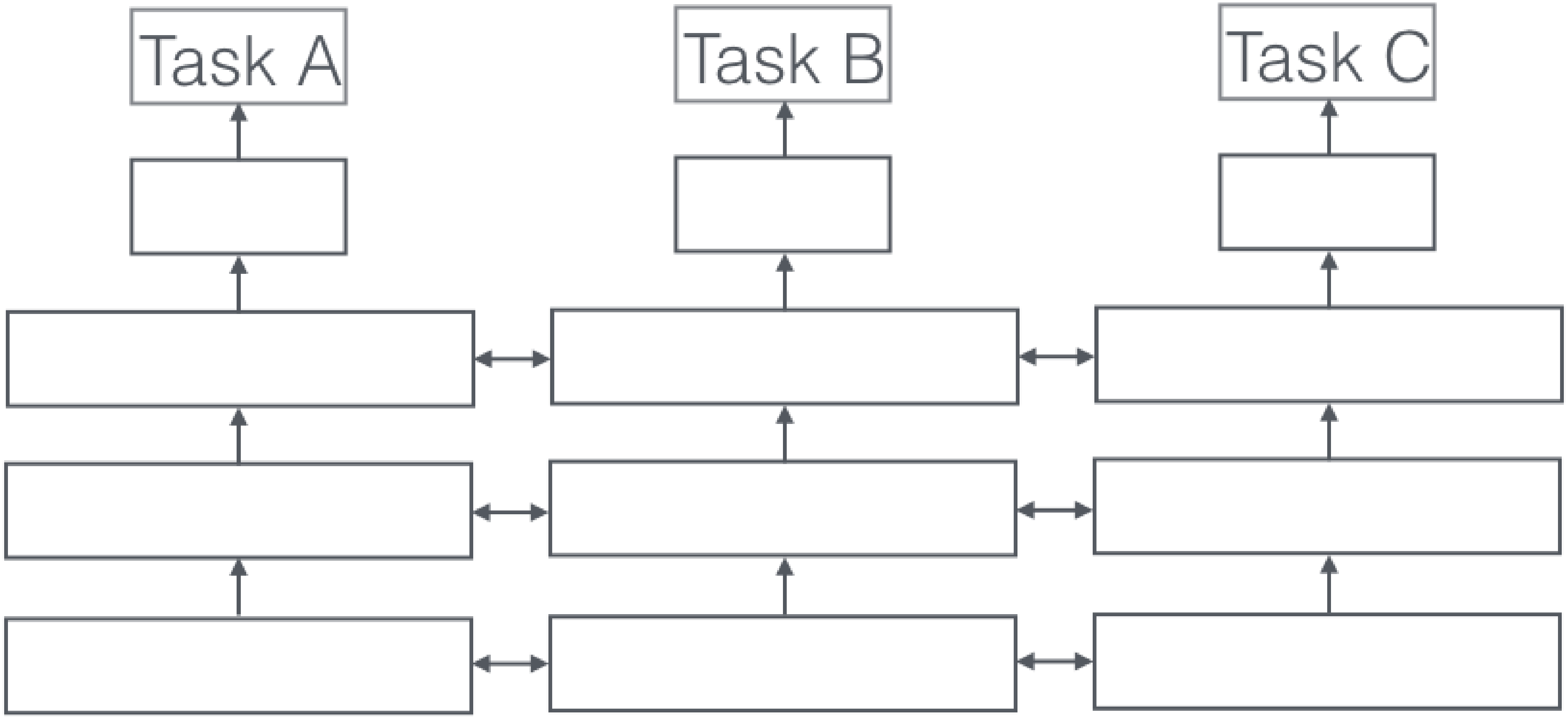
Xiaolong Wang

This Class

- Multi-task Learning
- Domain Adaptation
- Semi-supervised Learning

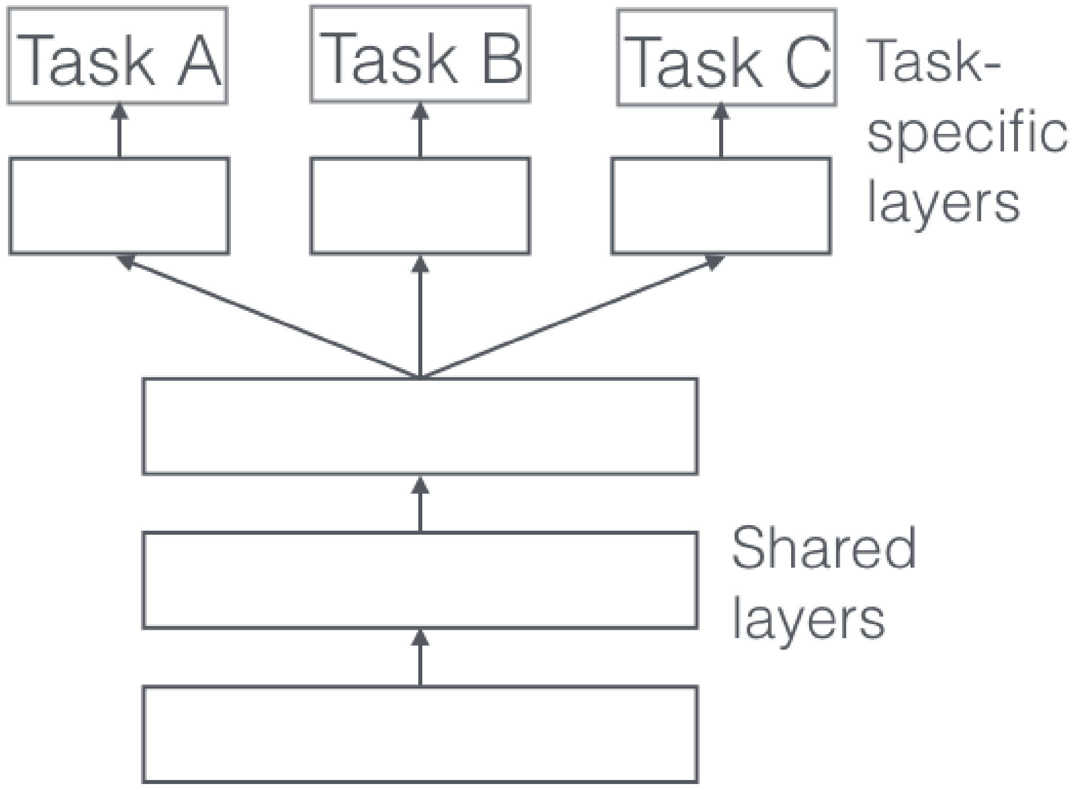
Multi-task Learning

Multi-task with Multiple Networks



https://ruder.io/multi-task/

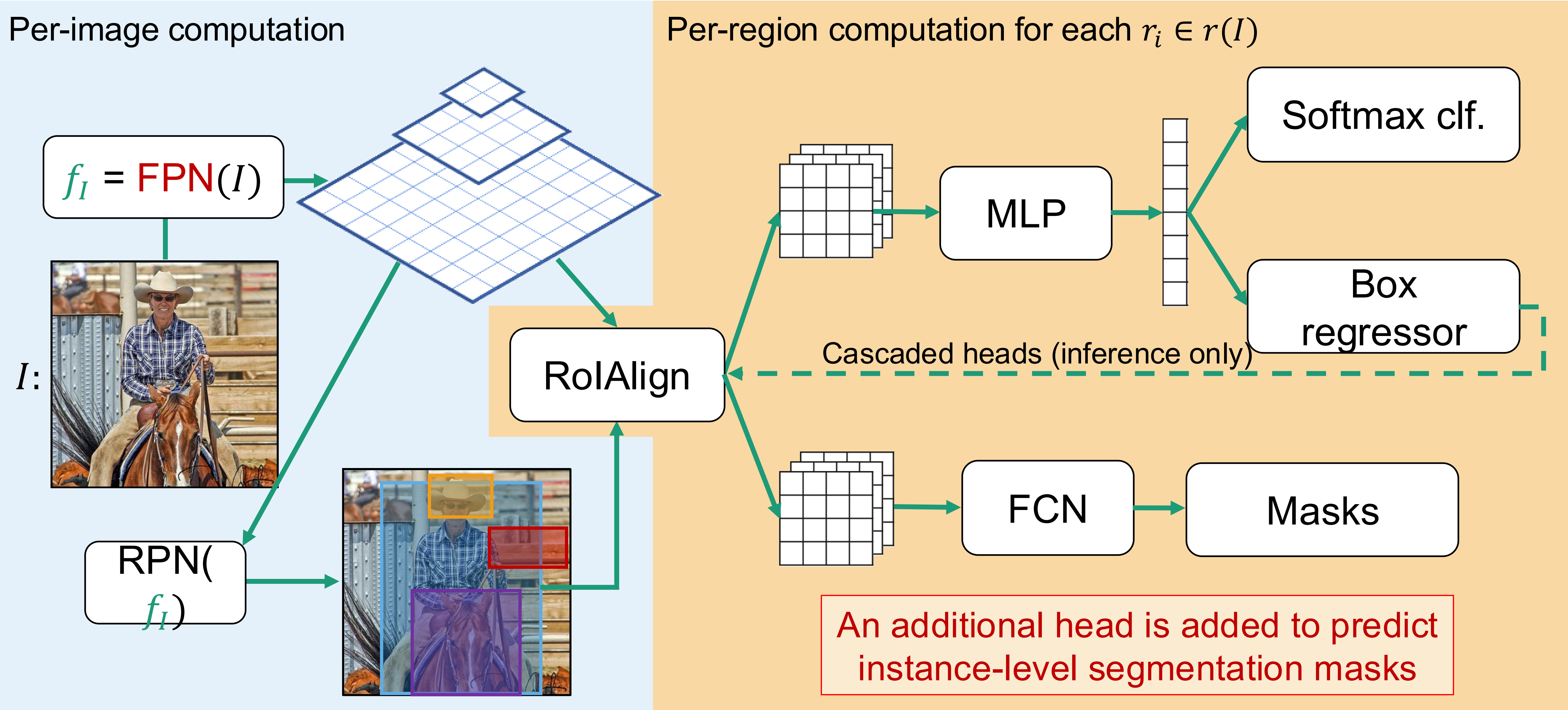
Multi-task with One Network



https://ruder.io/multi-task/

He et al. Mask R-CNN. ICCV 2017.

We have already seen in detection



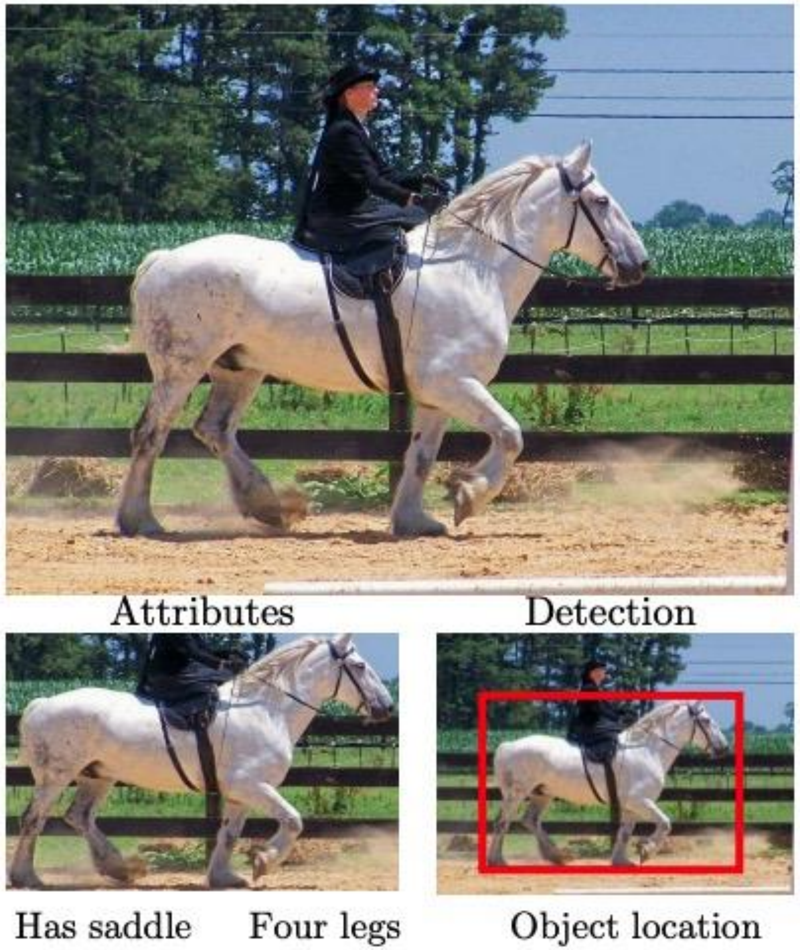
He et al. Mask R-CNN. ICCV 2017.

We have already seen in detection

	AP^{bb}_{person}	AP^{mask}_{person}	AP^{kp}
Faster R-CNN	52.5	-	-
Mask R-CNN, mask-only	**53.6**	**45.8**	-
Mask R-CNN, keypoint-only	50.7	-	64.2
Mask R-CNN, keypoint & mask	52.0	45.1	**64.7**

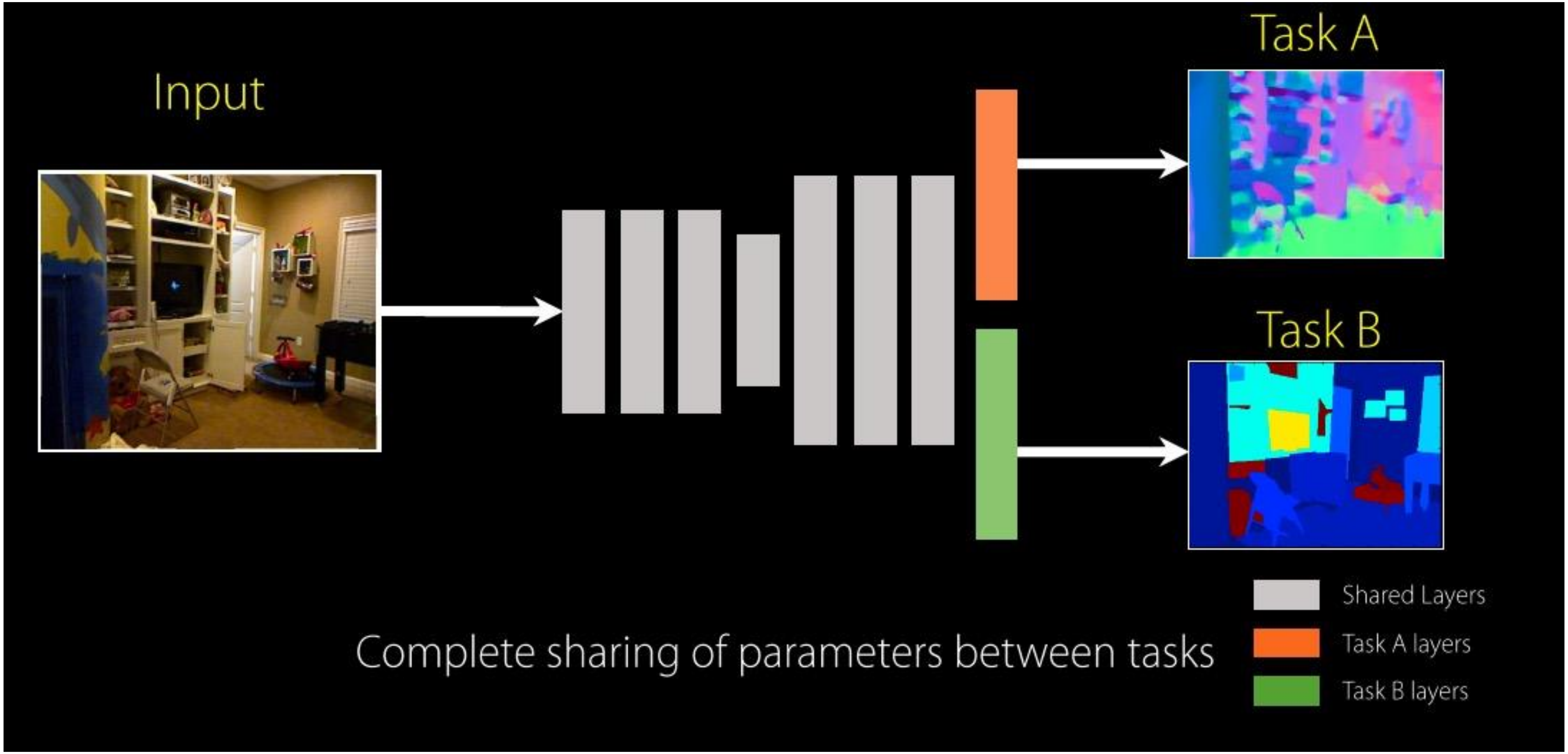
Table 5. **Multi-task learning** of box, mask, and keypoint about the *person* category, evaluated on `minival`. All entries are trained on the same data for fair comparisons. The backbone is ResNet-50-FPN. The entry with 64.2 AP on `minival` has 62.7 AP on `test-dev`. The entry with 64.7 AP on `minival` has 63.1 AP on `test-dev` (see Table 4).

Multi-task Learning

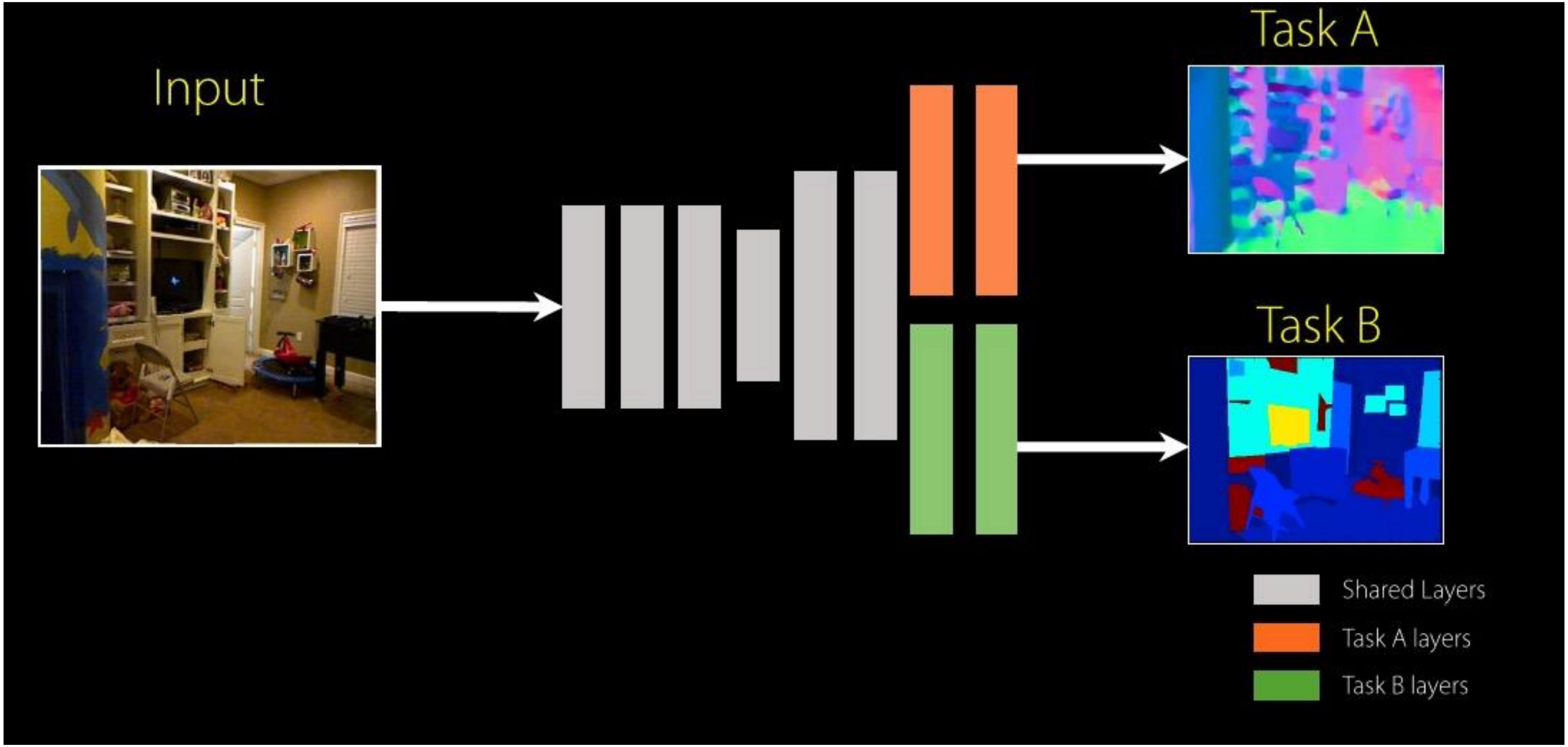


Misra et al. Cross-stitch Networks for Multi-task Learning. 2016.

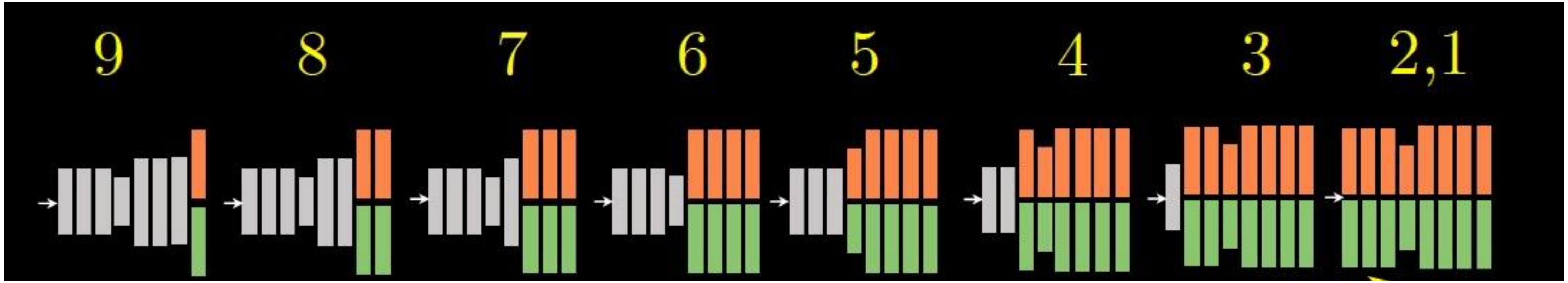
Multi-task Learning



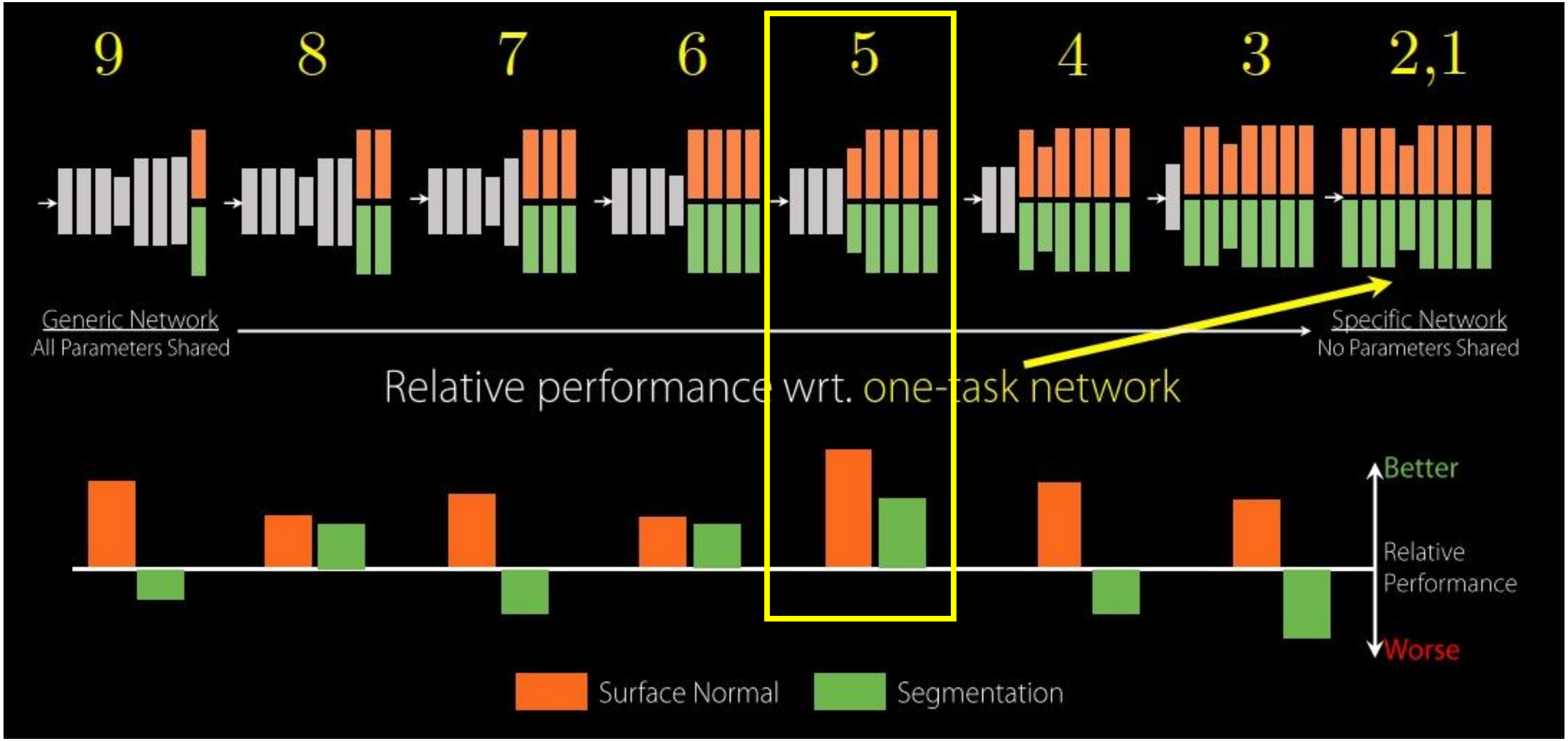
Multi-task Learning



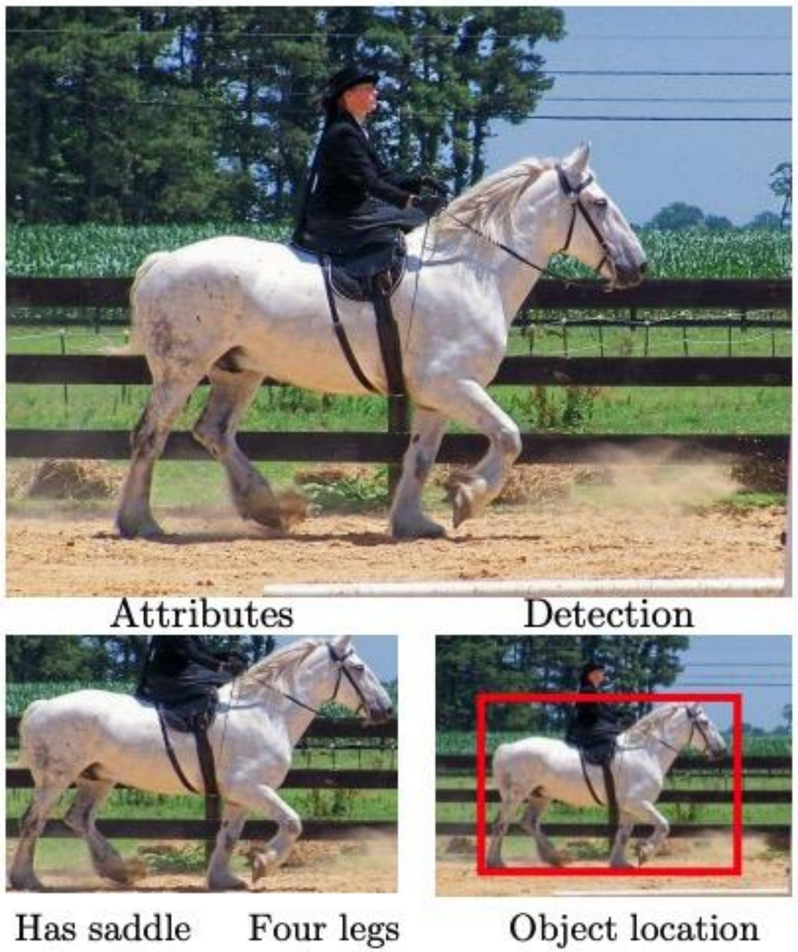
How to share?



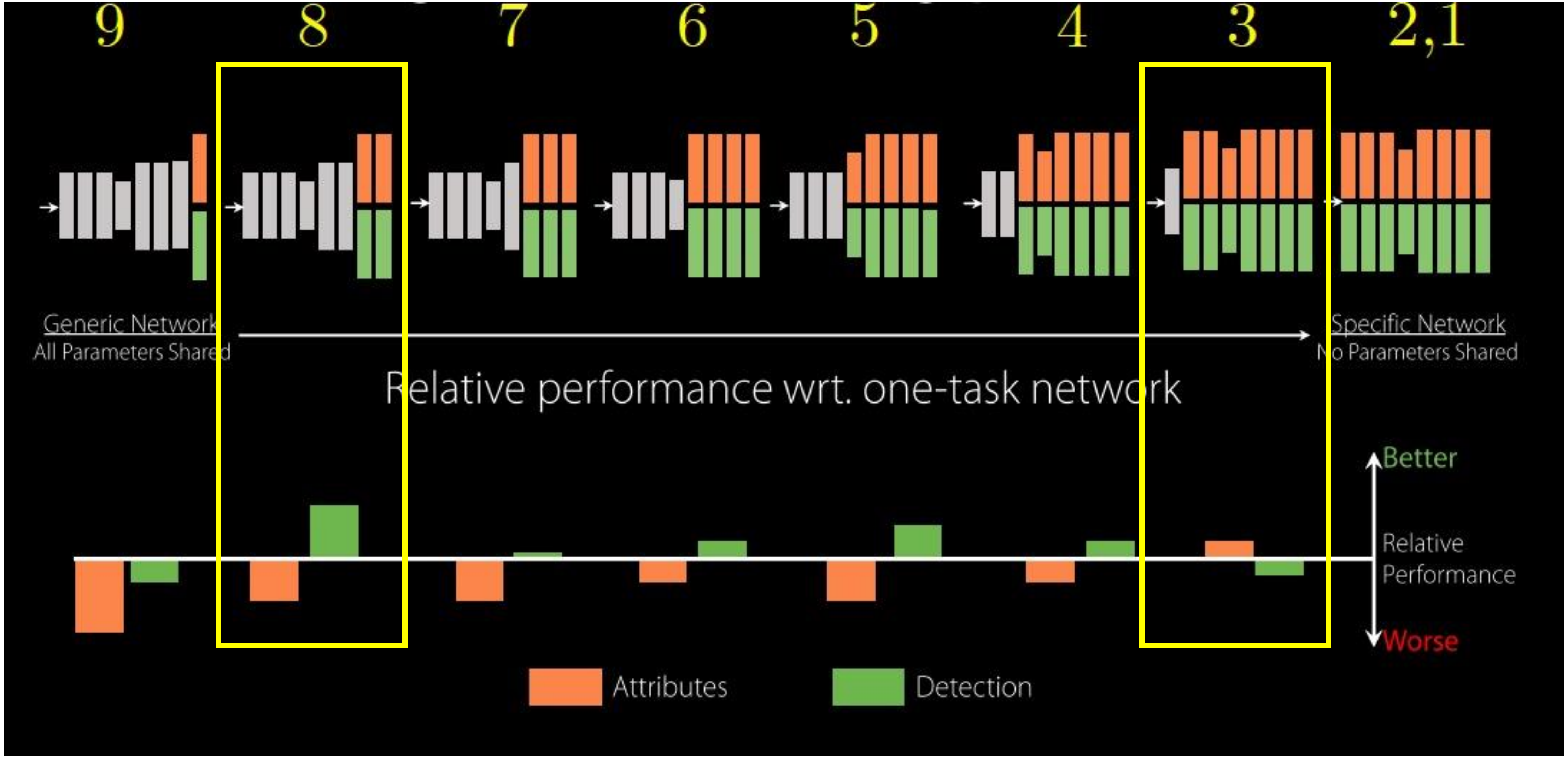
How to share?


9
8
7
6
5
4
3
2,1
Generic Network
All Parameters Shared
Specific Network
No Parameters Shared
Relative performance wrt. one-task network
Better
Relative Performance
Worse
Surface Normal
Segmentation

Another Task



How to share?

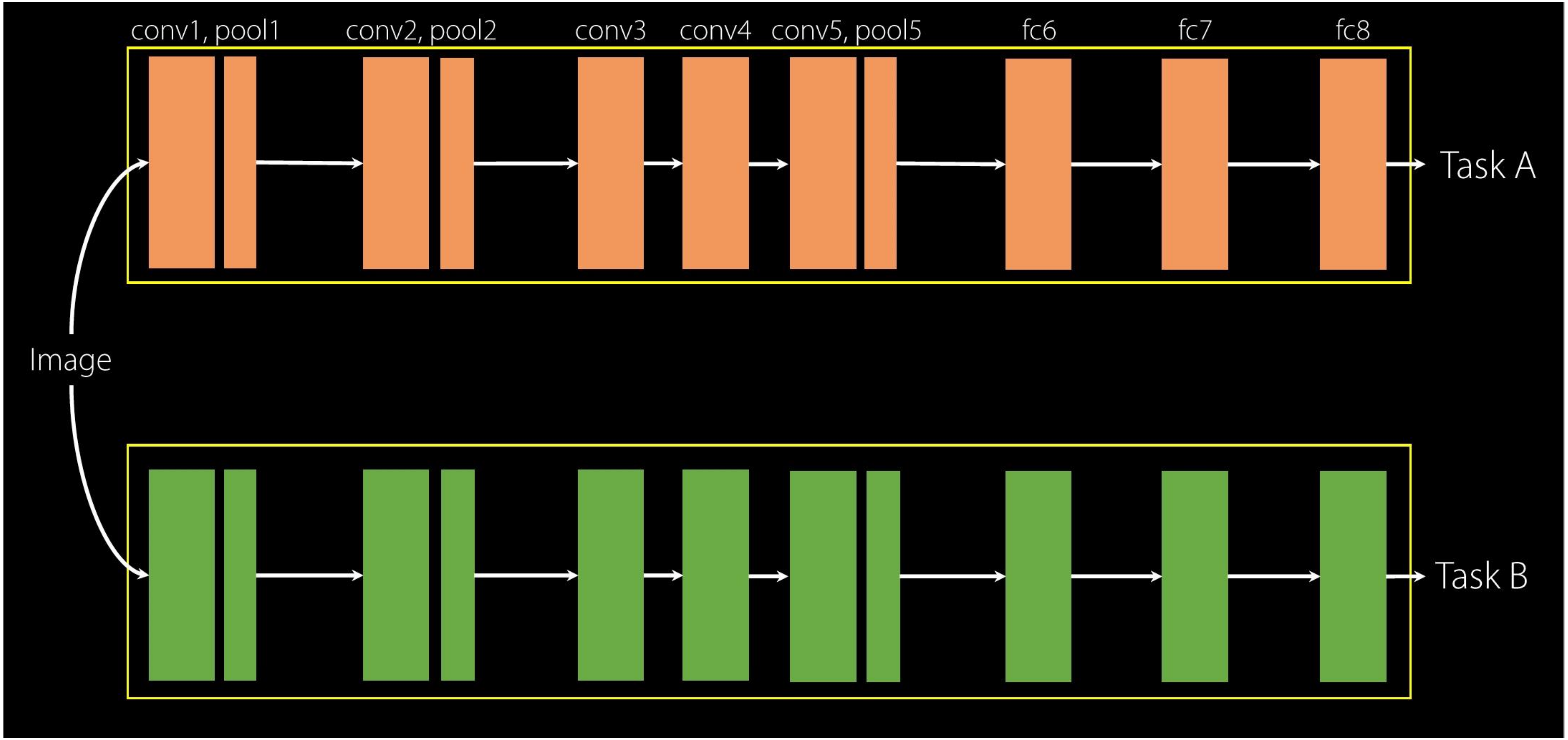


9
8
7
6
5
4
3
2,1
Generic Network
All Parameters Shared
Specific Network
No Parameters Shared
Relative performance wrt. one-task network
Better
Relative
Performance
Worse
Attributes
Detection

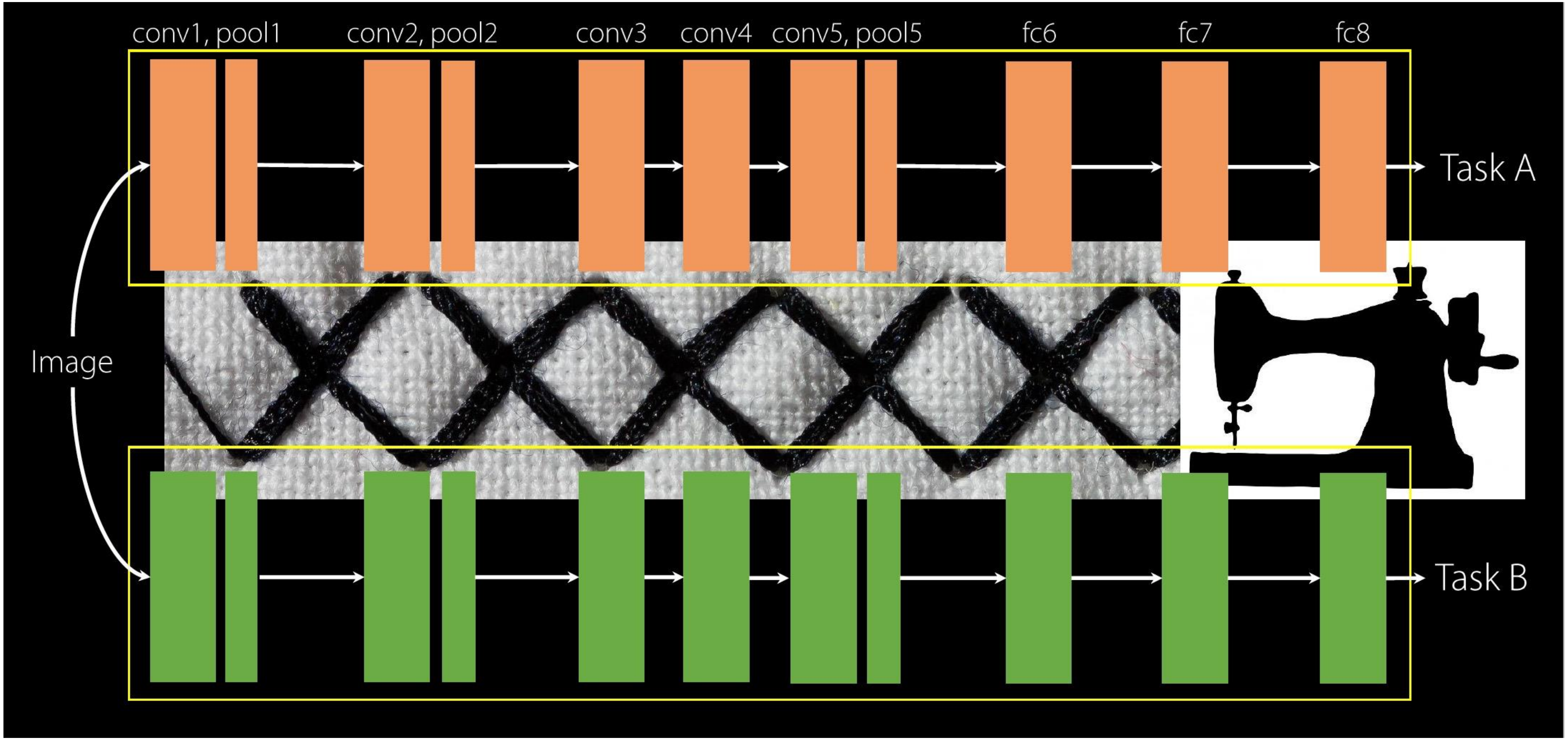
What do people usually do?

- Given a new set of tasks:
    - Enumerate many ConvNet architectures
    - Train all of them
    - Pick the best architecture

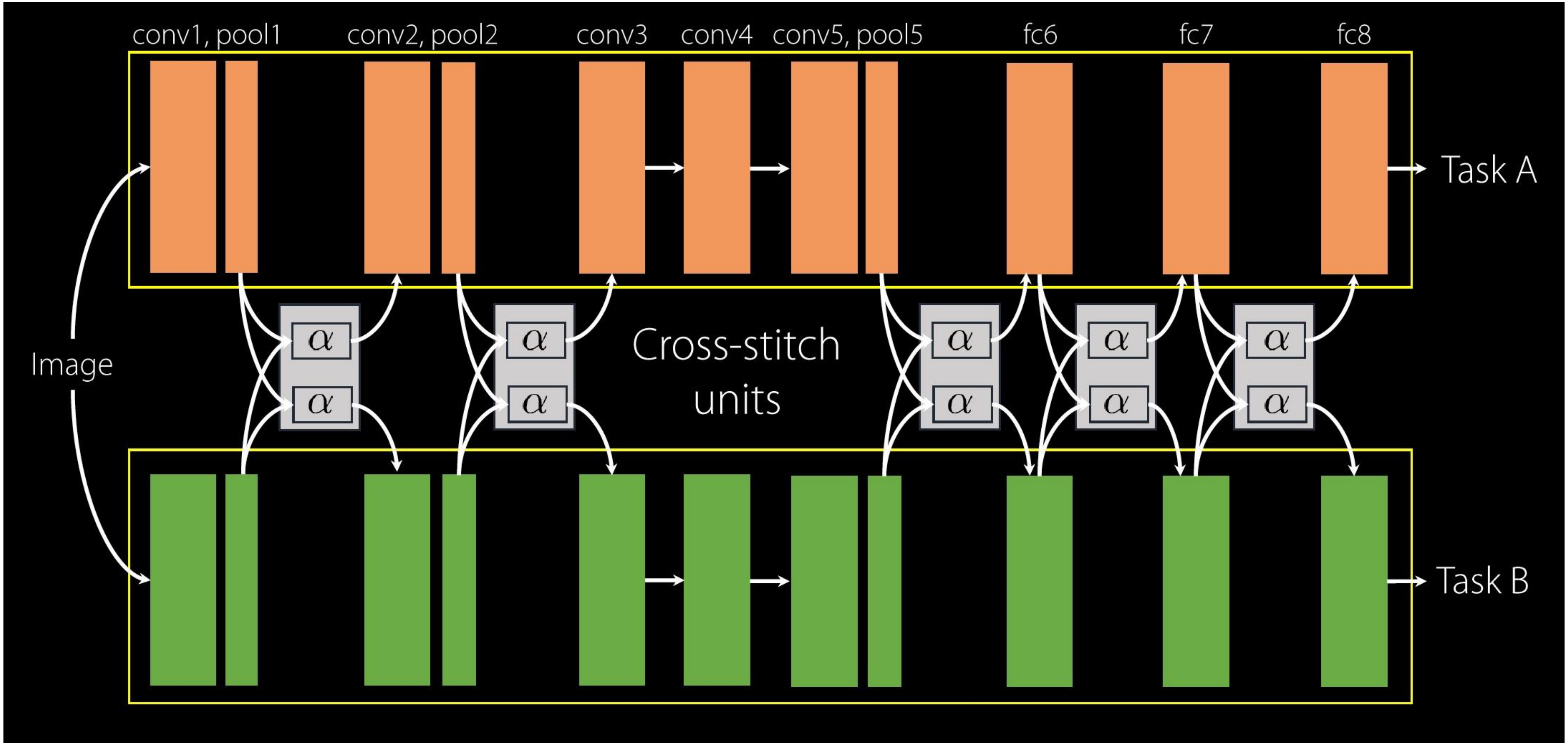
Cross-stitch Networks



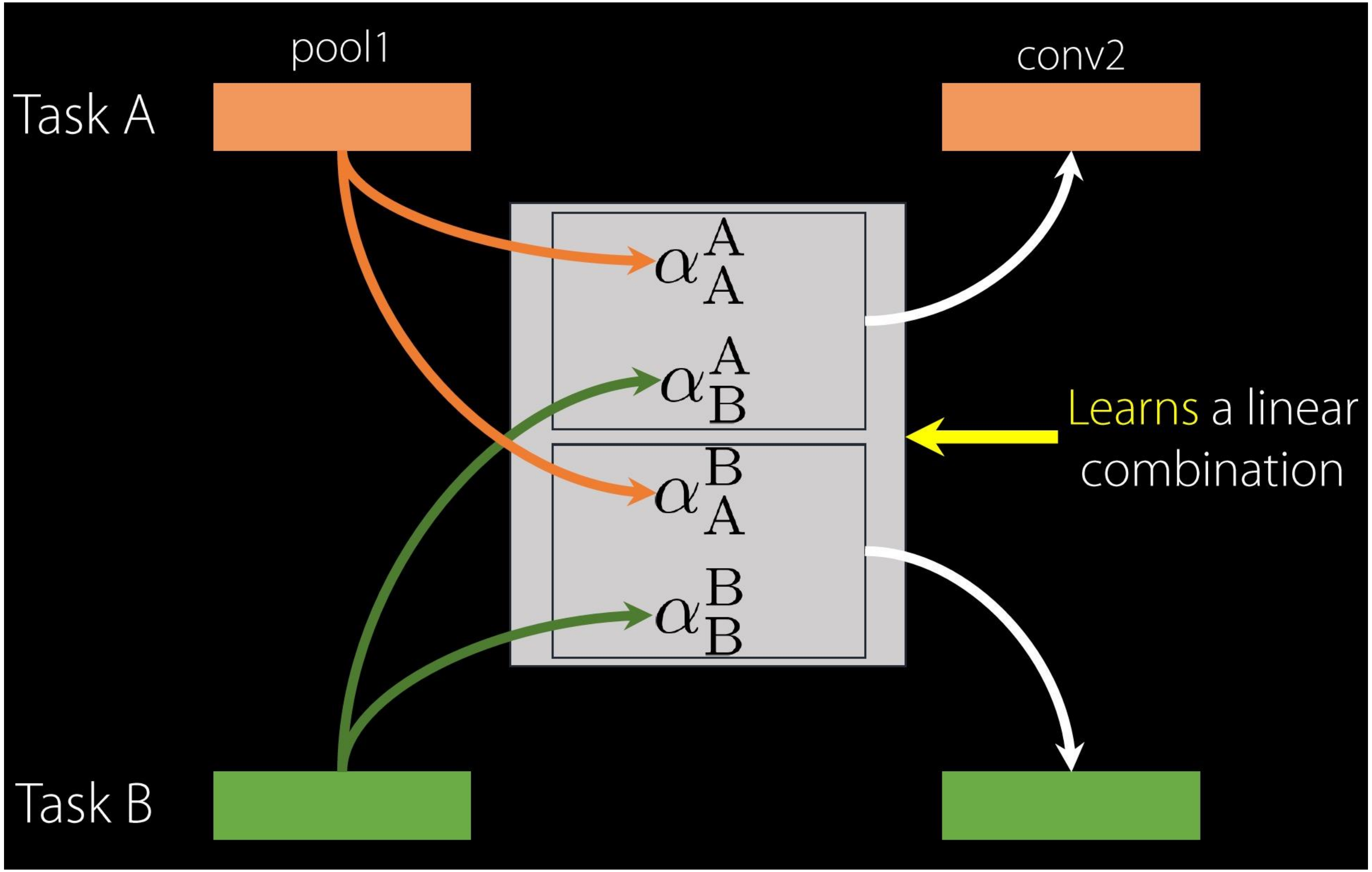
Cross-stitch Networks



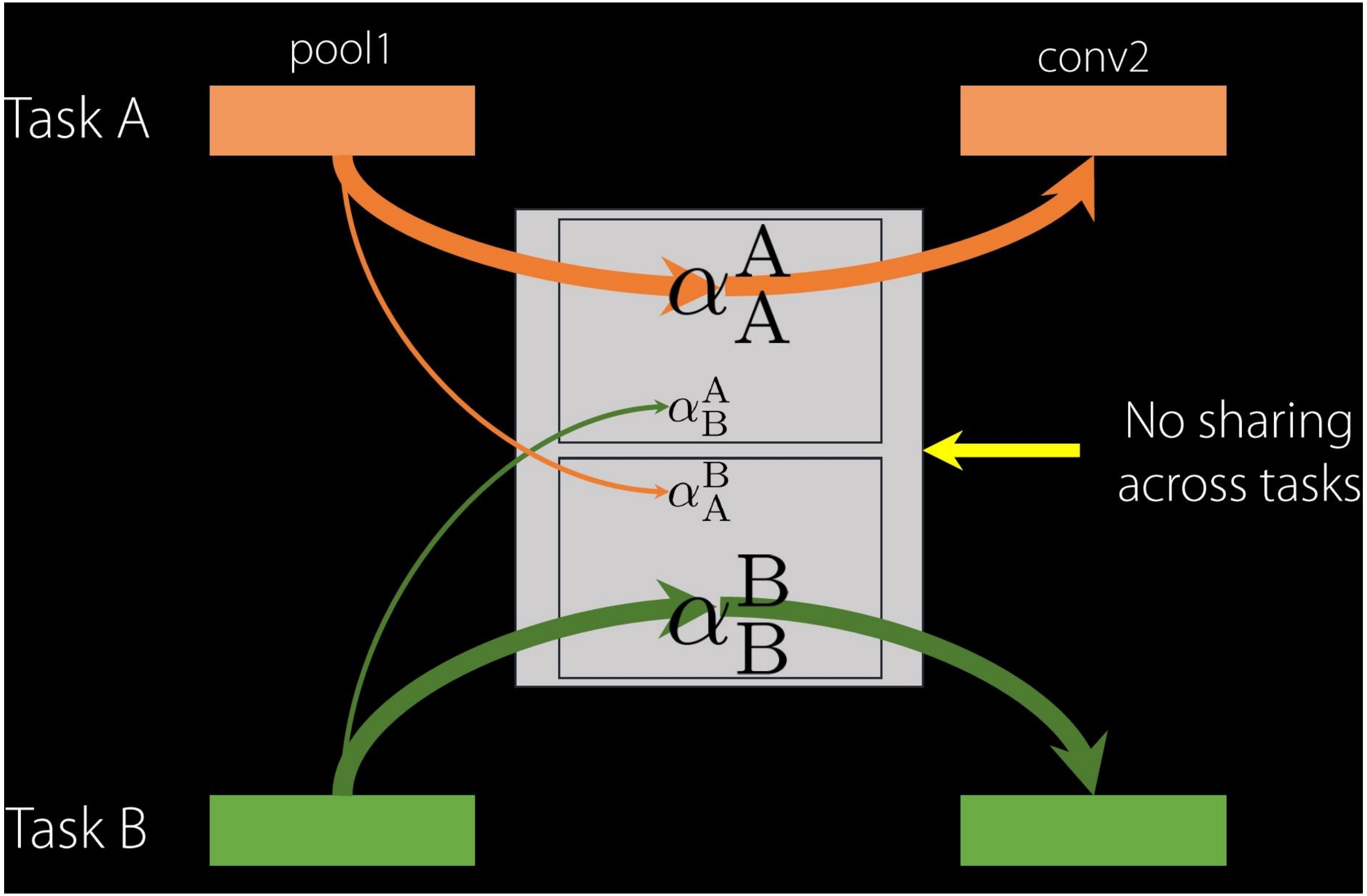
Cross-stitch Networks



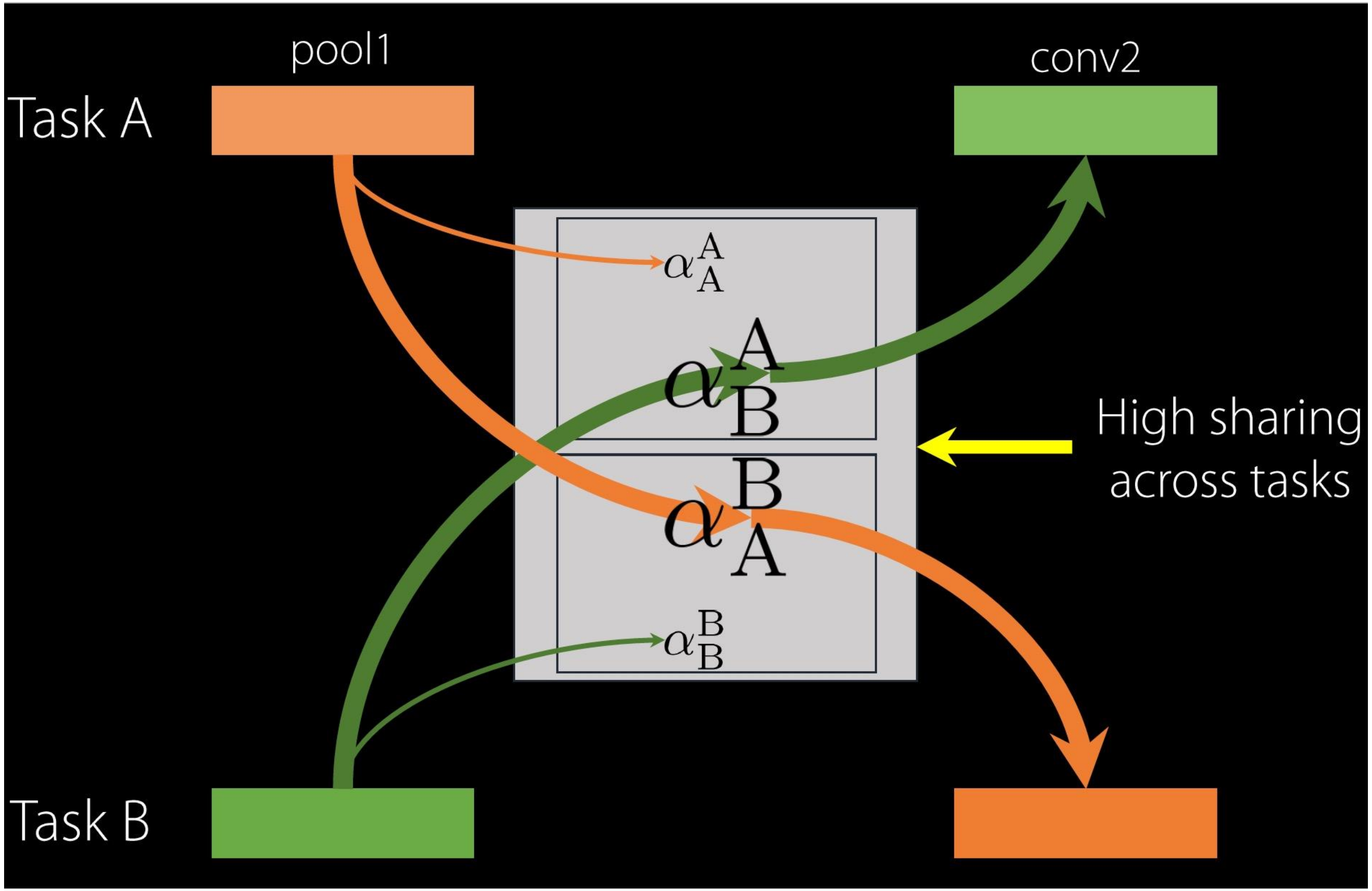
Cross-stitch Networks

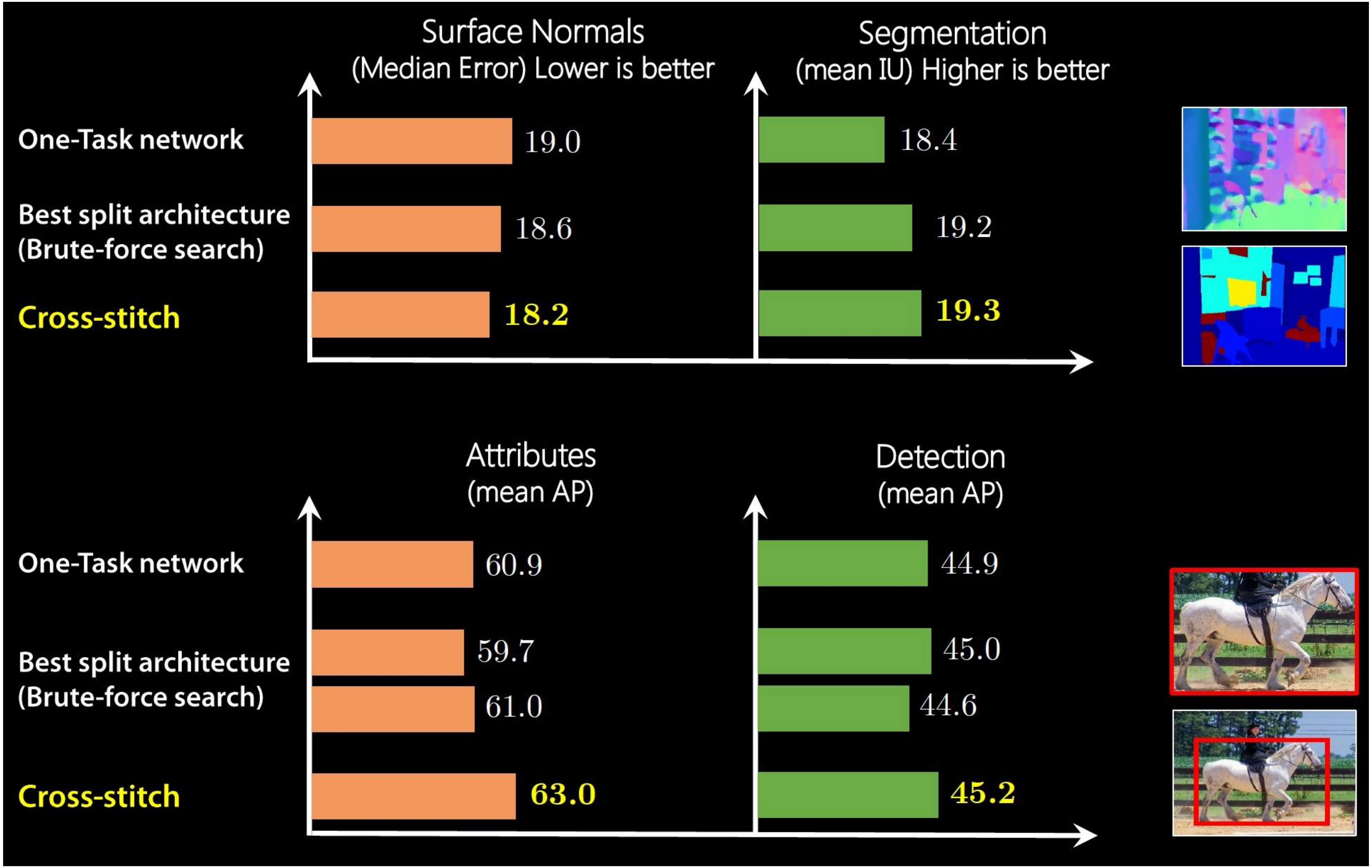


Cross-stitch Networks



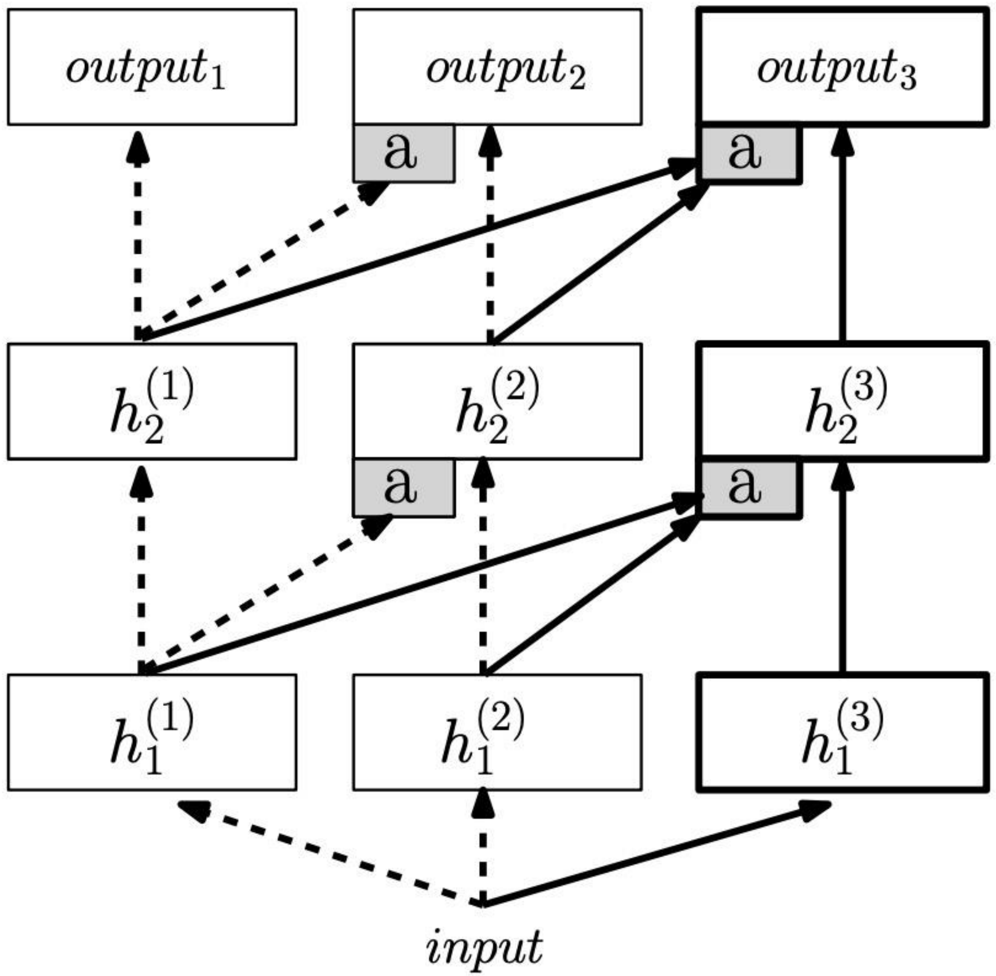
Cross-stitch Networks





Surface Normals
(Median Error) Lower is better
Segmentation
(mean IU) Higher is better
One-Task network
Best split architecture
(Brute-force search)
Cross-stitch
19.0
18.6
18.2
18.4
19.2
19.3
Attributes
(mean AP)
Detection
(mean AP)
One-Task network
Best split architecture
(Brute-force search)
Cross-stitch
60.9
59.7
61.0
63.0
44.9
45.0
44.6
45.2

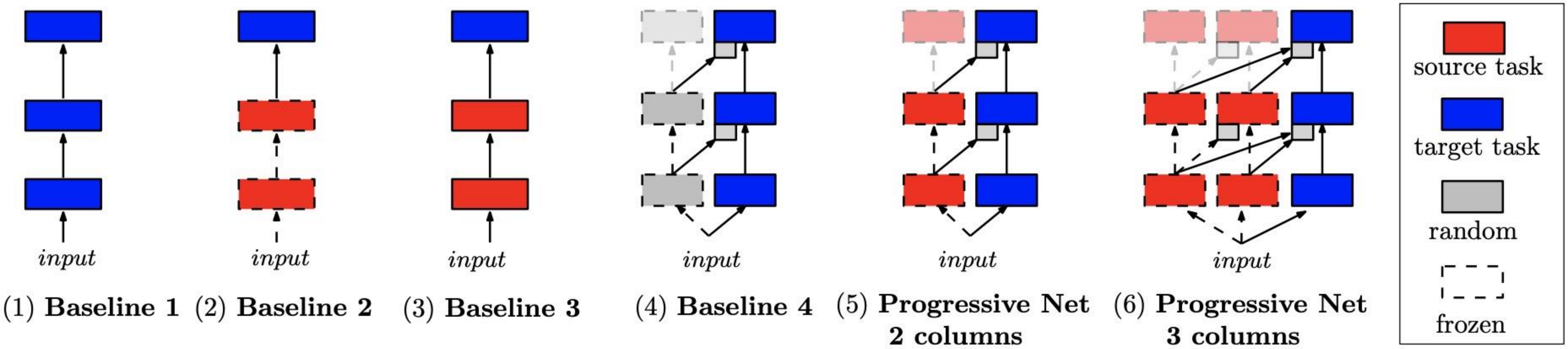
Progressive Networks



$$h_i^{(k)} = f\left(W_i^{(k)} h_{i-1}^{(k)} + \sum_{j<k} U_i^{(k:j)} h_{i-1}^{(j)}\right)$$

Rusu et al. Progressive Neural Networks. 2016.

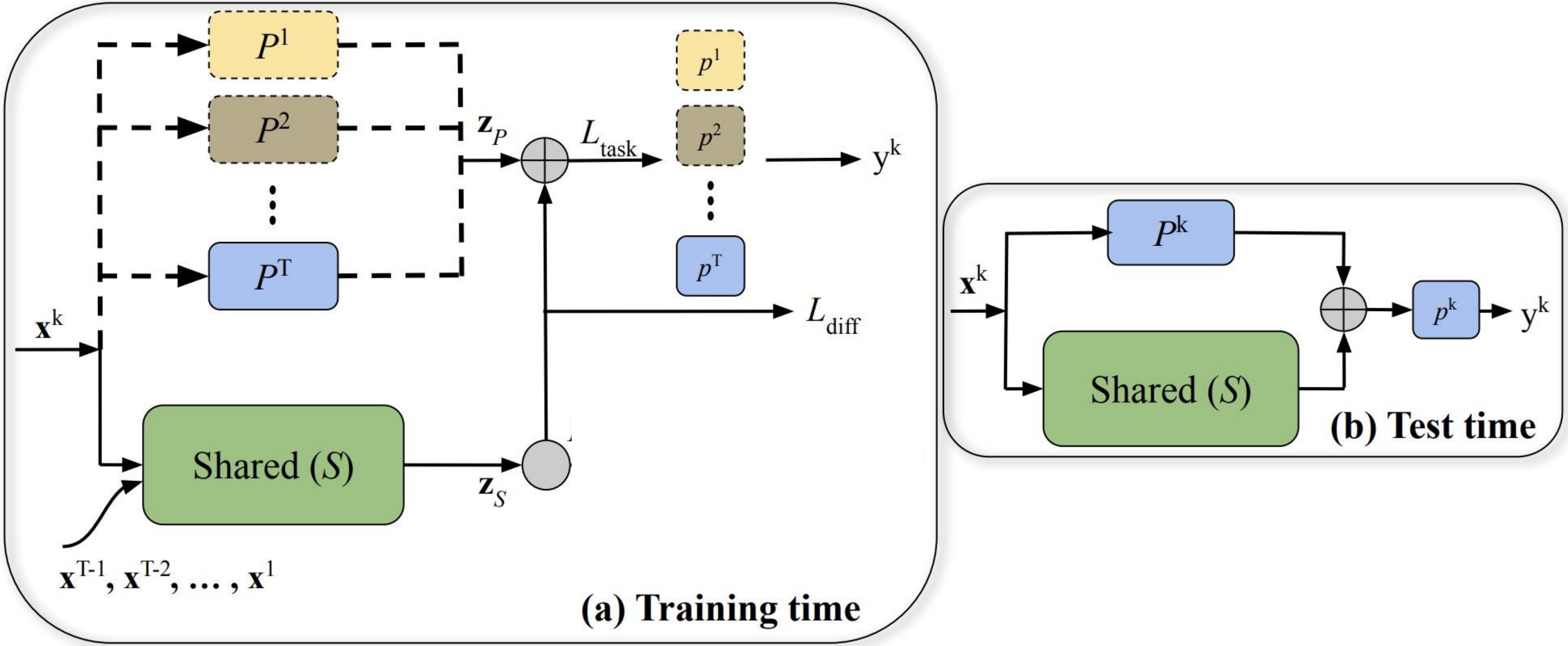
Progressive Networks



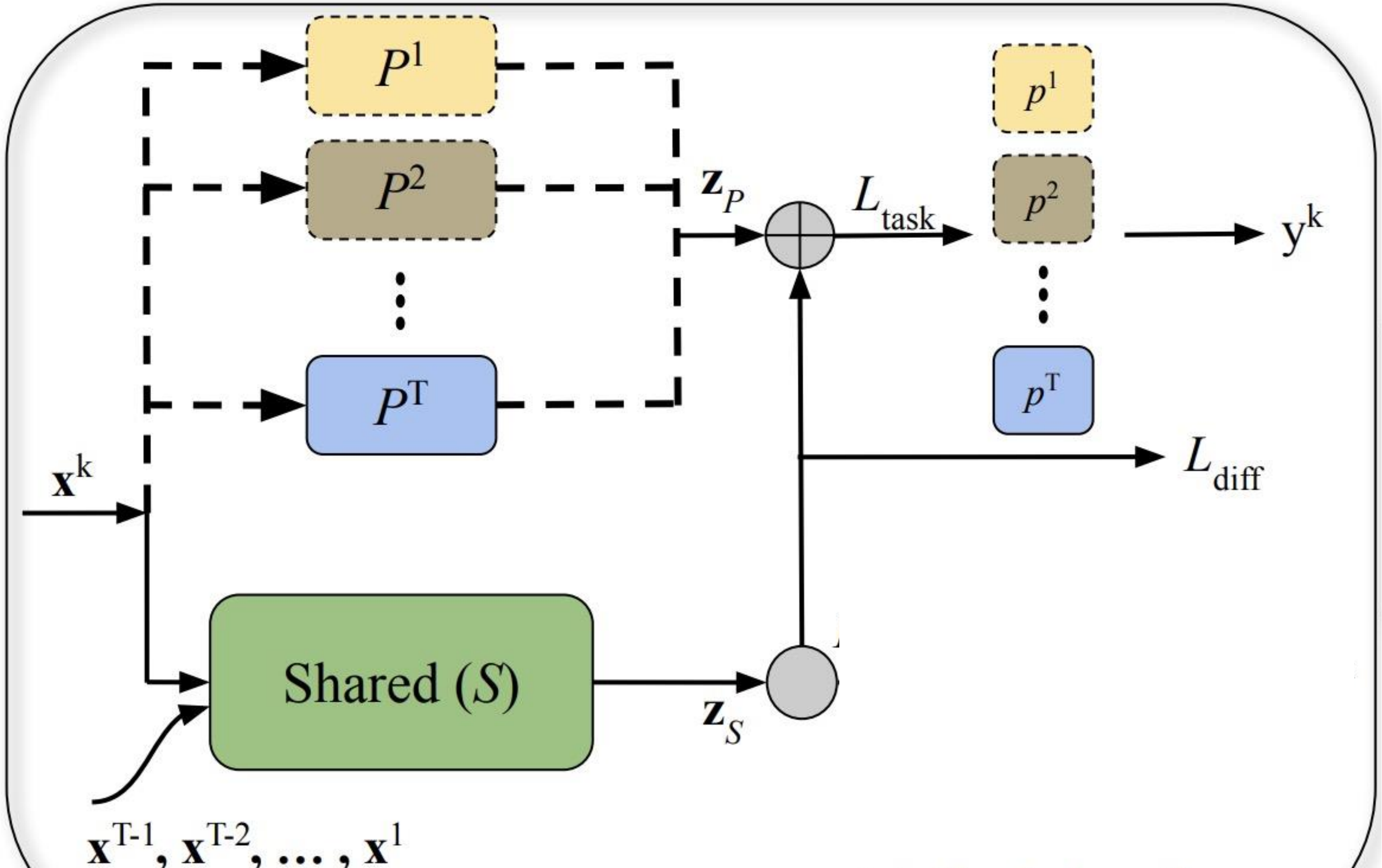
Continual Learning

- We want to re-use the past experience as much as possible
- But we do not want to forget about the old tasks, i.e., **catastrophic forgetting**

Continual Learning



Continual Learning

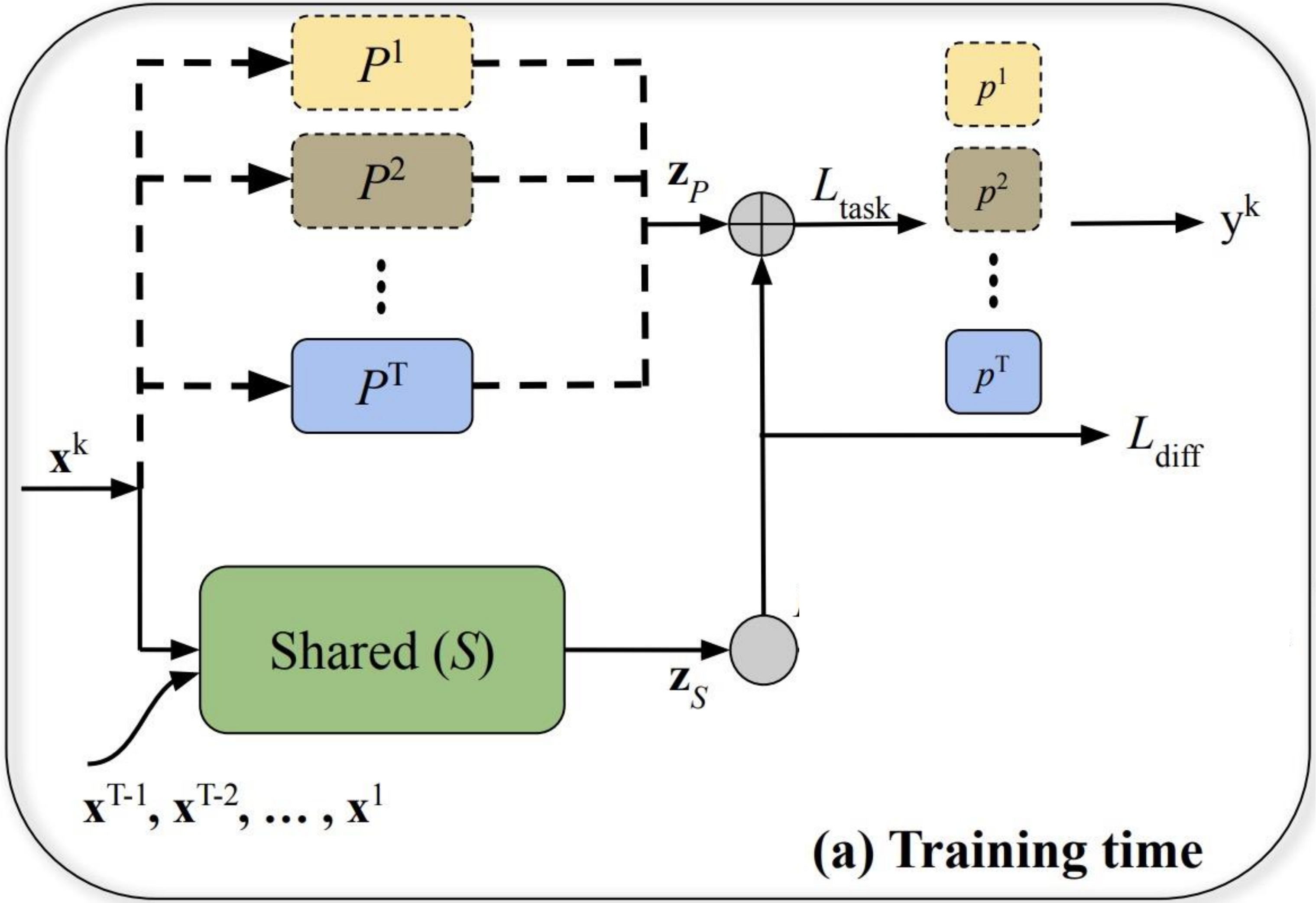


(a) Training time

When train a new task, we need to train the task head and shared network together.

Changing shared features might lead to **catastrophic forgetting**

Continual Learning

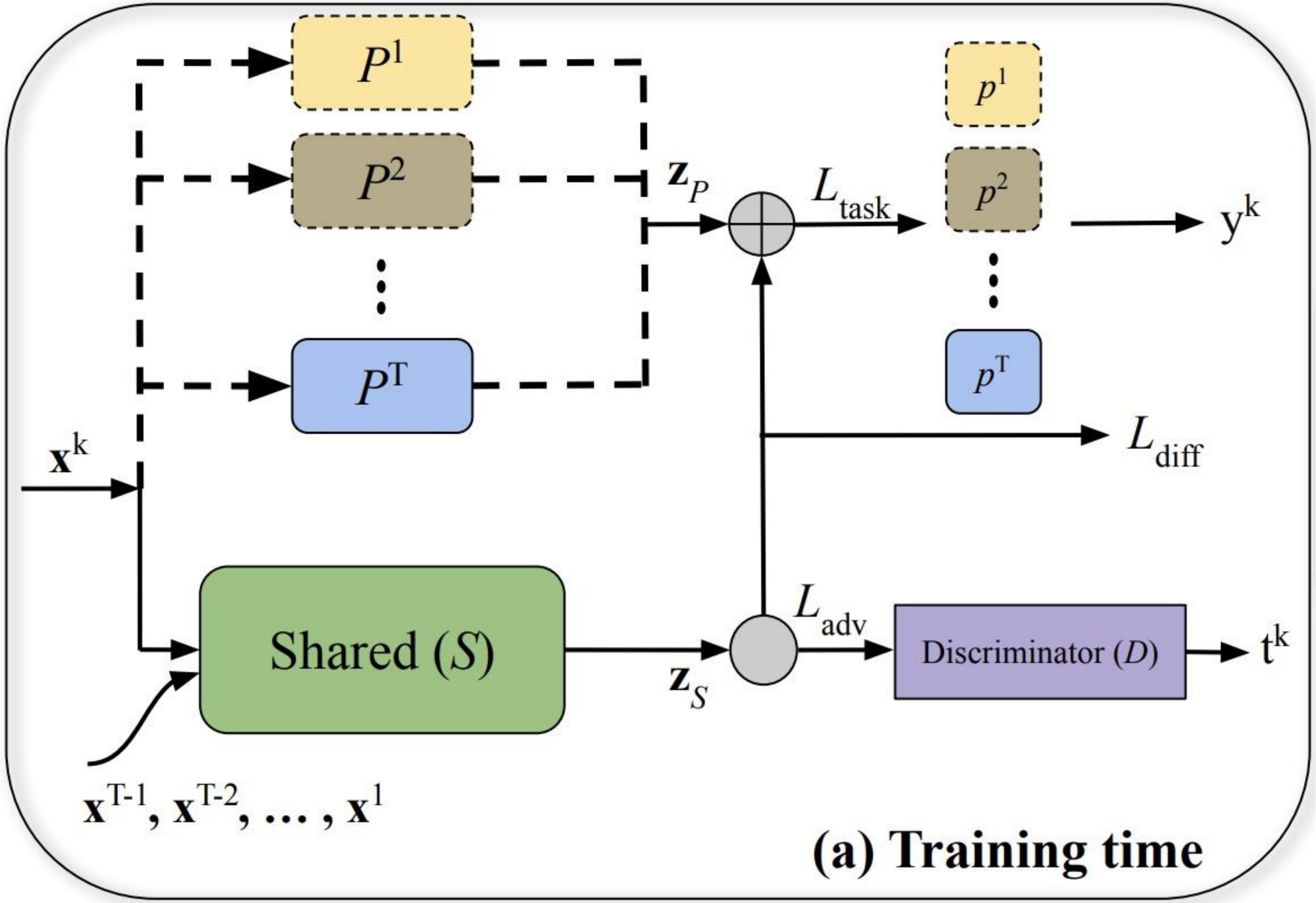


Disentanglement:

Shared network should capture the commons across tasks

Individual heads model task-specifics

Continual Learning



Adversarial Learning:

Input different task and images to shared backbone, the output features should be in the same distribution.

Use D to force it.

Domain Adaptation

Domain Adaptation: Train on Source adapt to Target

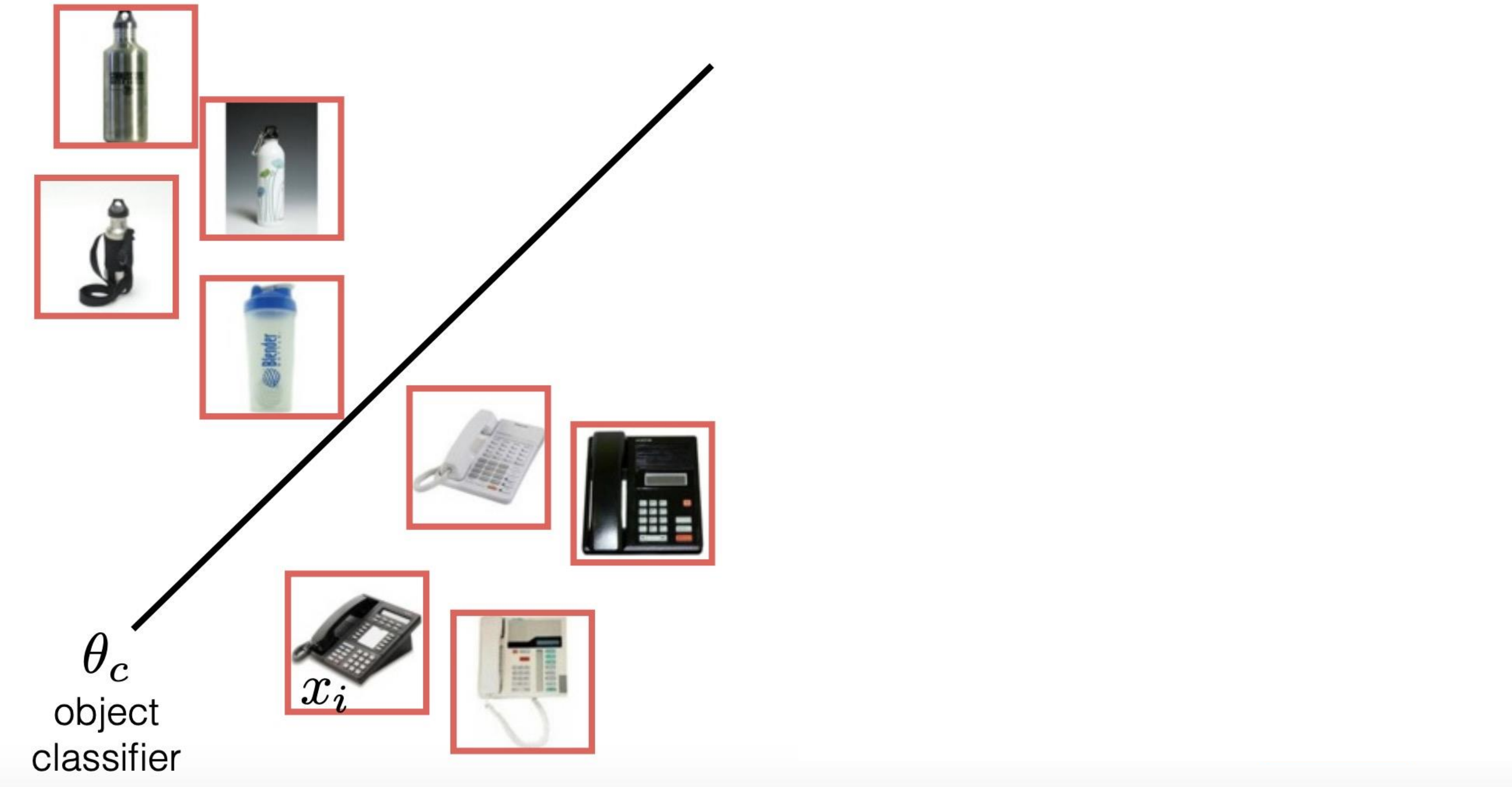
Source Domain $\sim P_S(X, Y)$

lots of **labeled** data

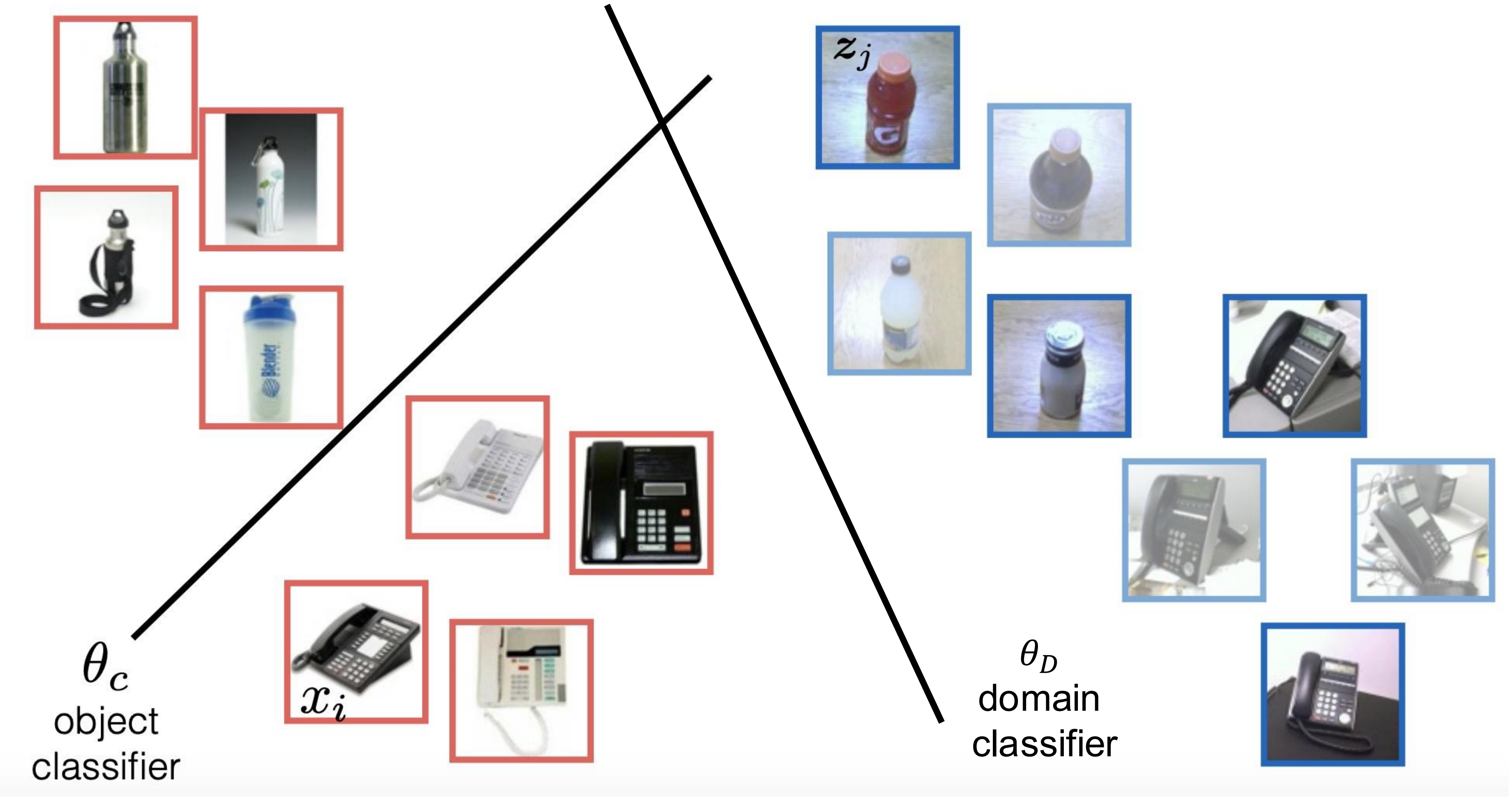
$$D_S = \{(\mathbf{x}_i, y_i), \forall i \in \{1, \dots, N\}\}$$

Credits: https://people.eecs.berkeley.edu/~jhoffman/talks/taskcv-iccv15.pdf

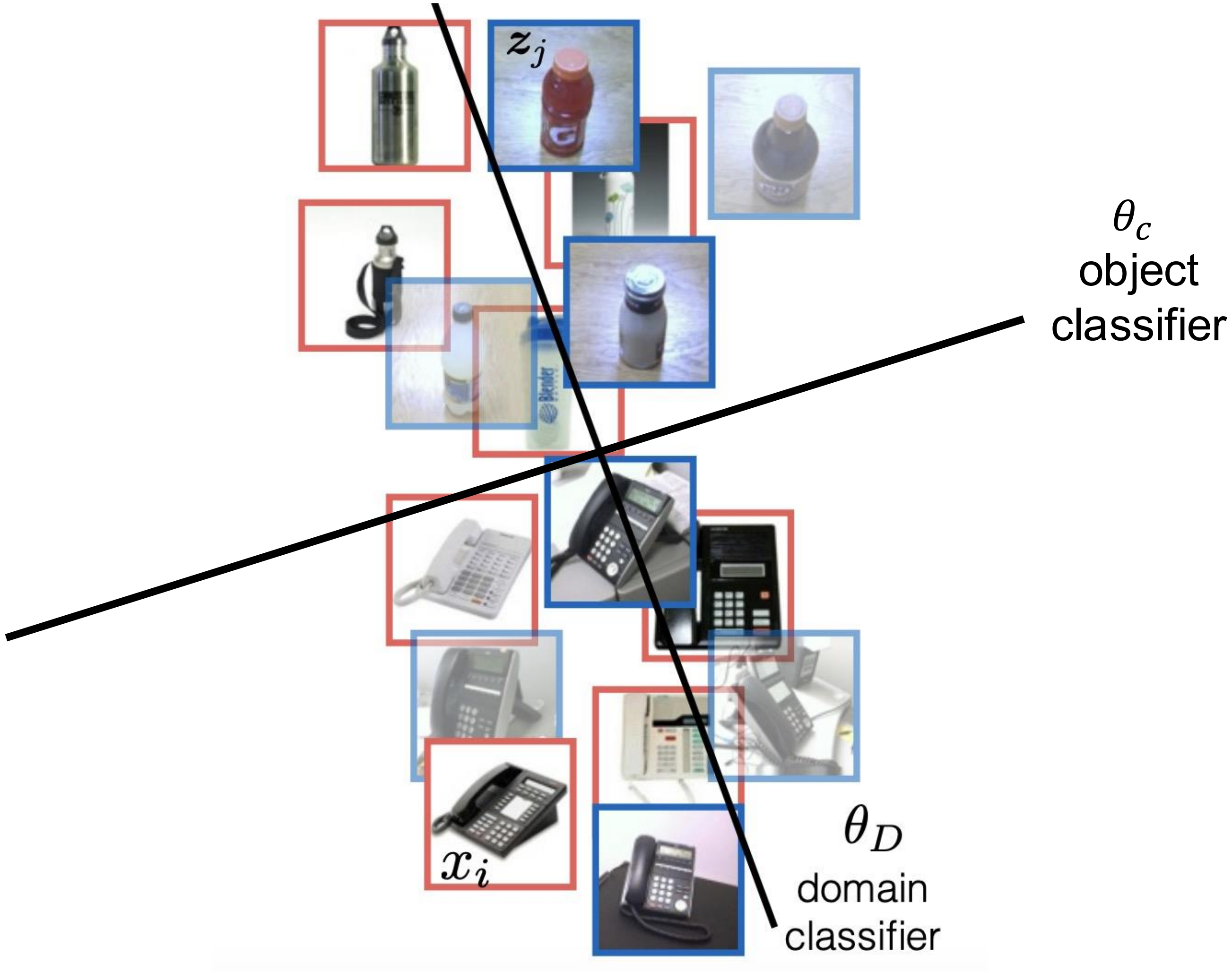
Train on Source adapt to Target



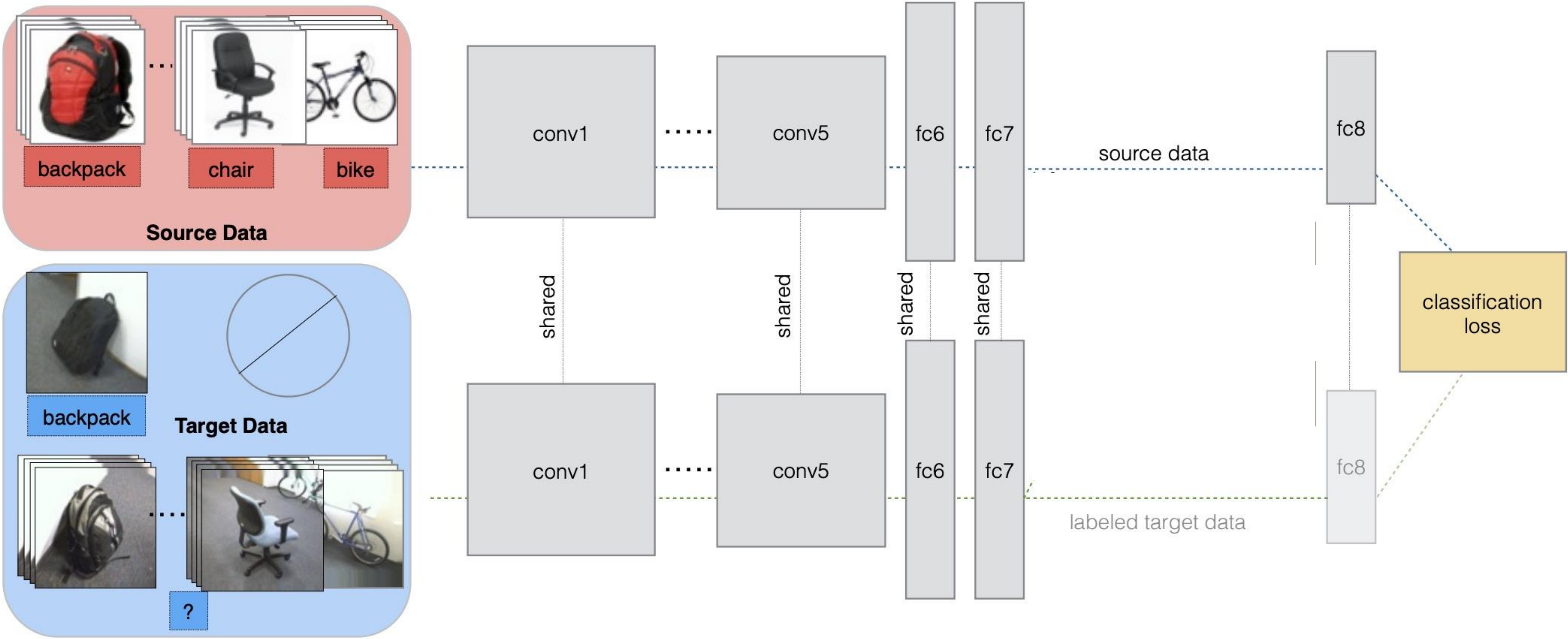
Train on Source adapt to Target



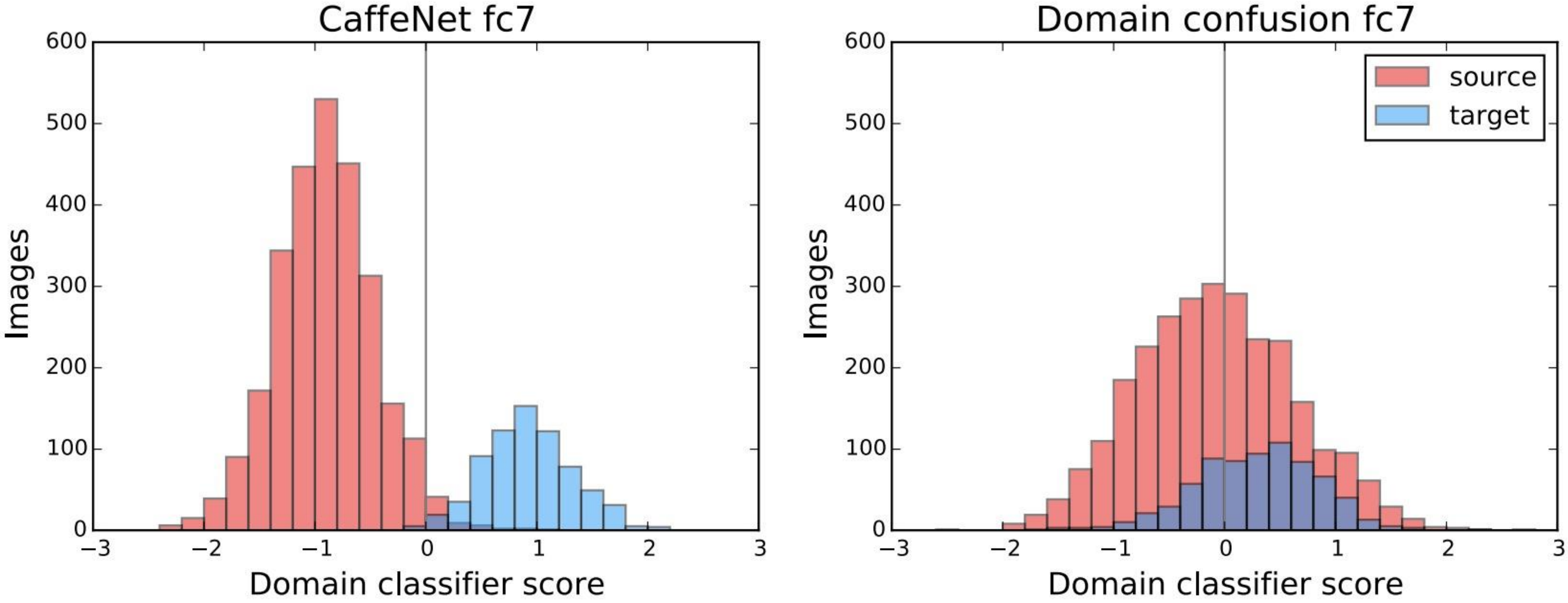
Adversarial Training: Learn Representations Confuse θ_D



Adversarial Training: Learn Representations Confuse θ_D



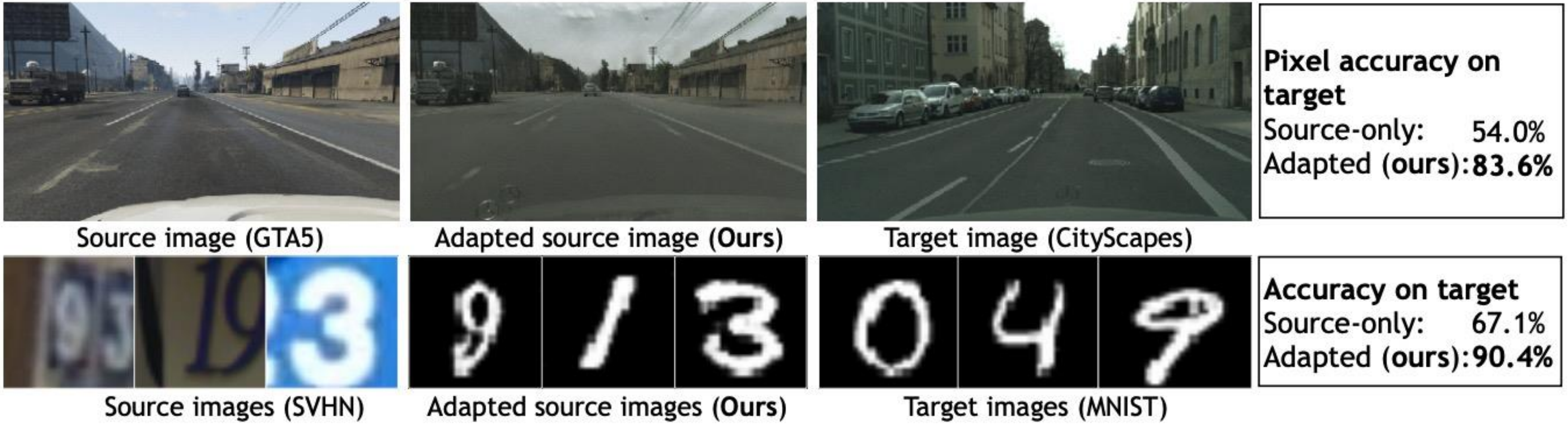
Adversarial Training:
Learn Representations Confuse θ_D



Adversarial Training:
Learn Representations Confuse θ_D

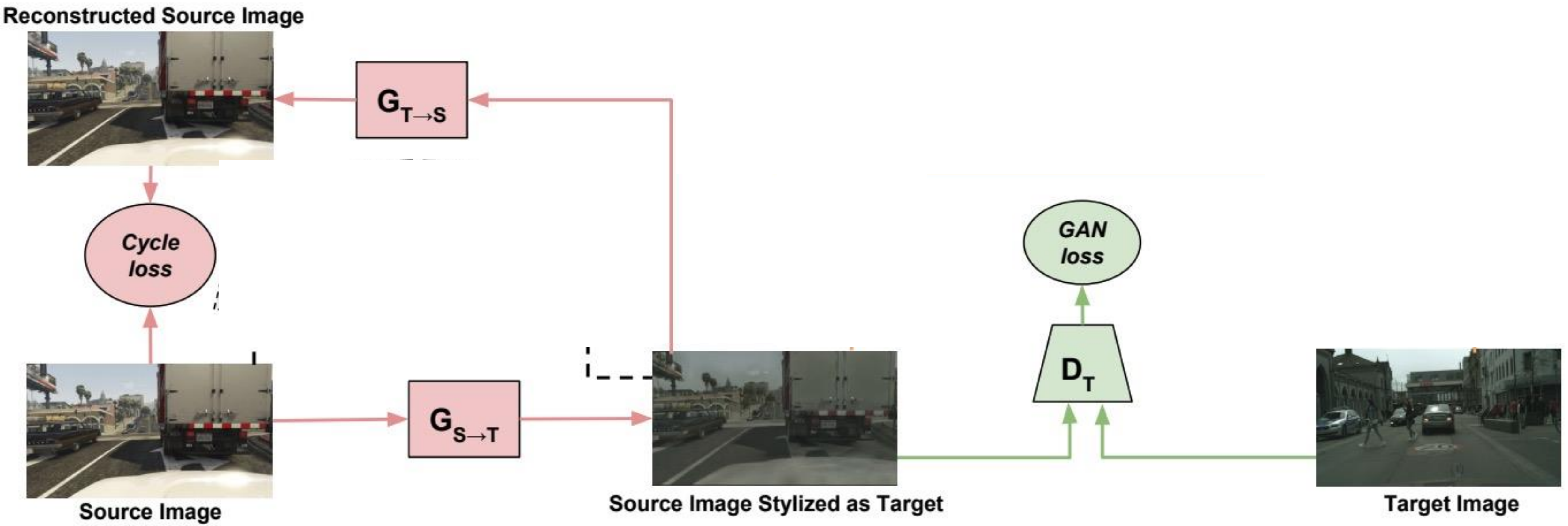
	$A \to W$	$A \to D$	$D \to A$	$D \to W$	$W \to A$	$W \to D$	Average
MMDT [18]	–	44.6 ± 0.3	–	–	–	58.3 ± 0.5	–
Source CNN	54.2 ± 0.6	63.2 ± 0.4	36.4 ± 0.1	89.3 ± 0.5	34.7 ± 0.1	94.5 ± 0.2	62.0
Ours: dom confusion only	55.2 ± 0.6	63.7 ± 0.9	41.2 ± 0.1	$\mathbf{91.3 \pm 0.4}$	$\mathbf{41.1 \pm 0.0}$	96.5 ± 0.1	64.8

CycleGAN for Domain Adaptation

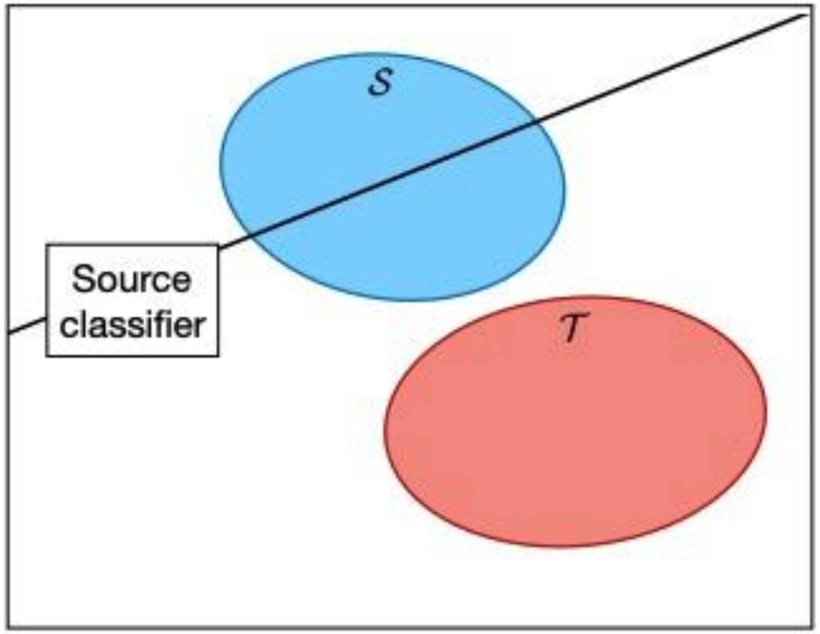


Hoffman et al. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. 2018

CycleGAN


Reconstructed Source Image
$G_{T \to S}$
Cycle loss
GAN loss
D_T
Source Image
$G_{S \to T}$
Source Image Stylized as Target
Target Image

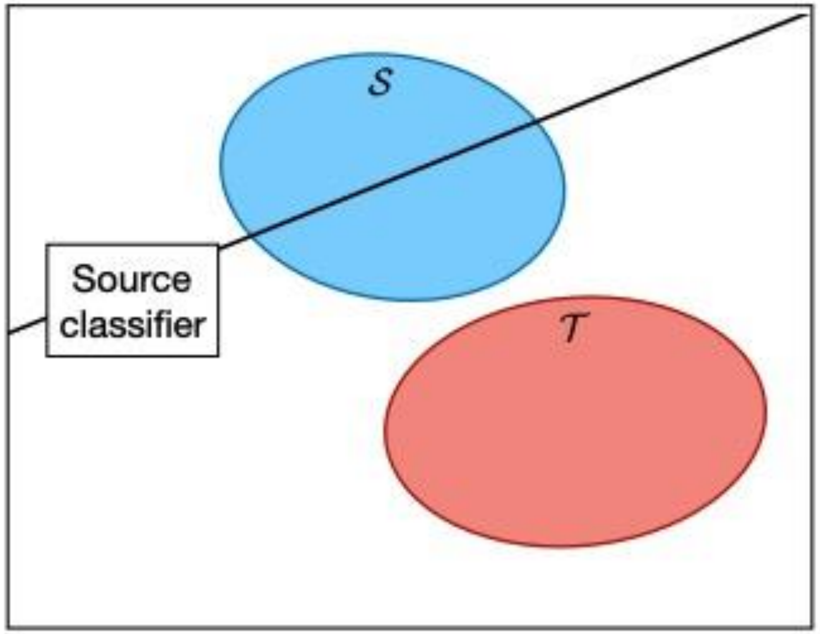
Self-Supervised Learning for UDA



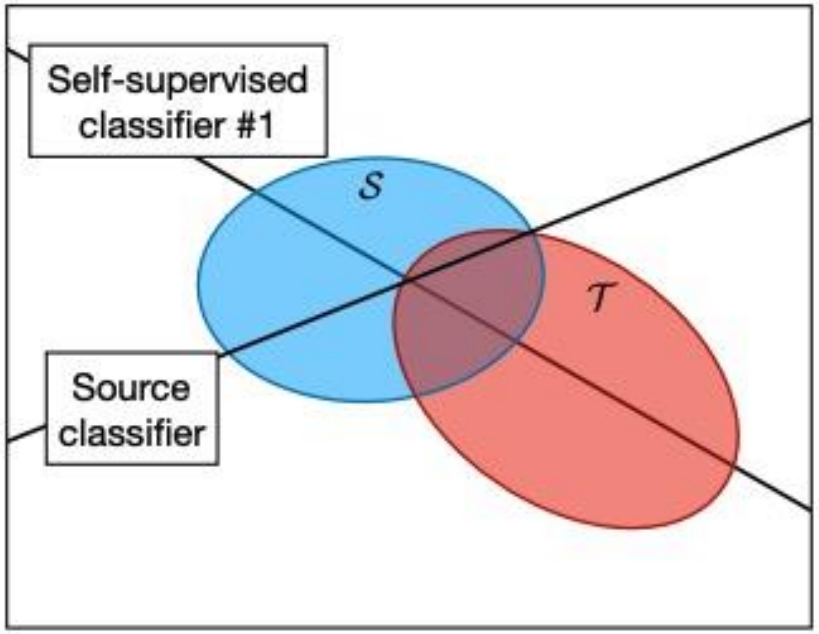
(a) Source classifier only

Sun et al. 2019

Self-Supervised Learning for UDA



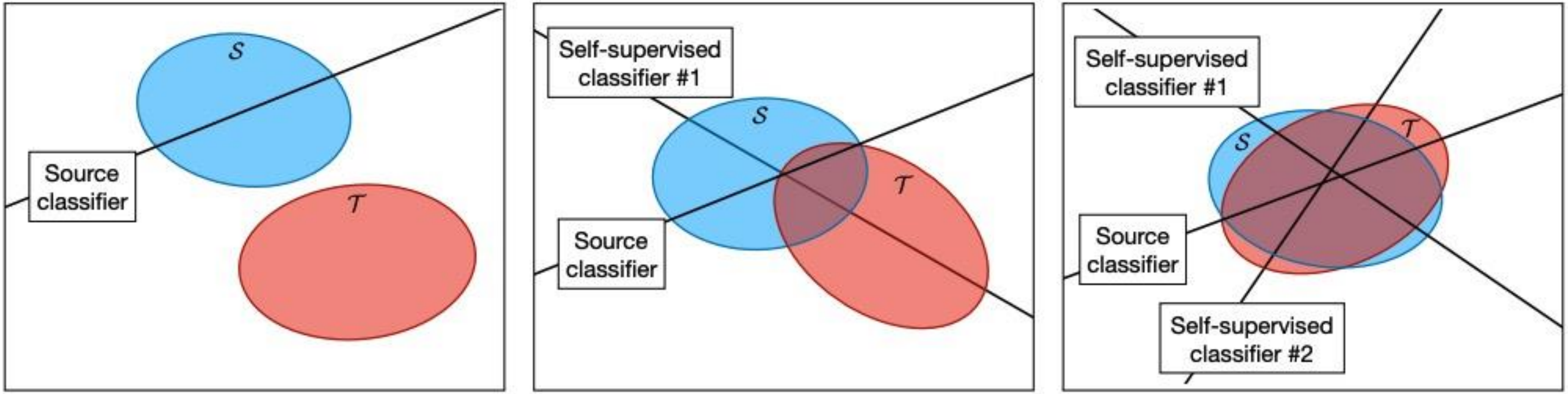
(a) Source classifier only



(b) adding a self-supervised task

Sun et al. 2019

Self-Supervised Learning for UDA



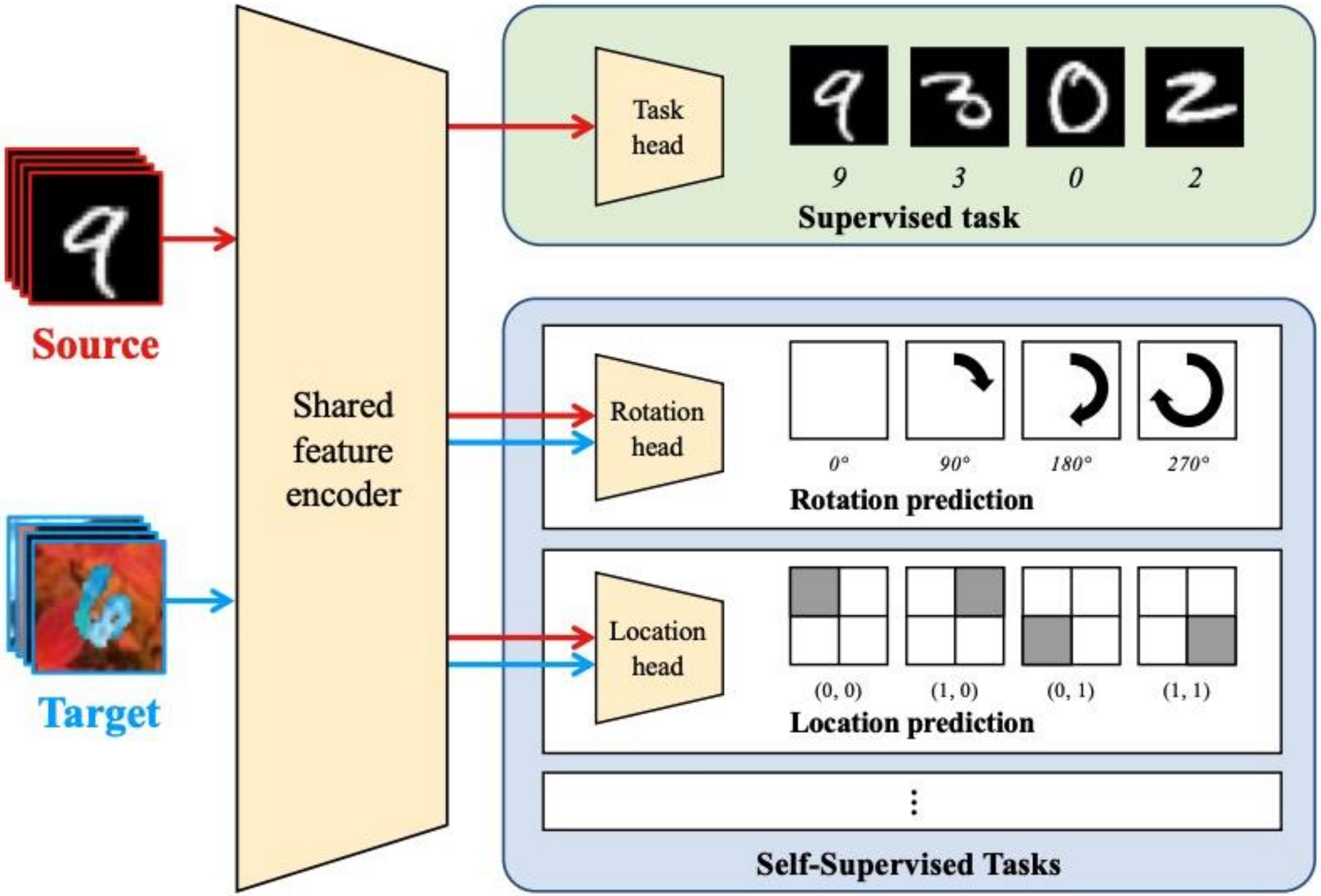
(a) Source classifier only (b) adding a self-supervised task (c) adding more tasks

Sun et al. 2019

Self-Supervised Learning for UDA

Task	Images and self-supervised labels			
Rotation				
	0°	90°	180°	270°

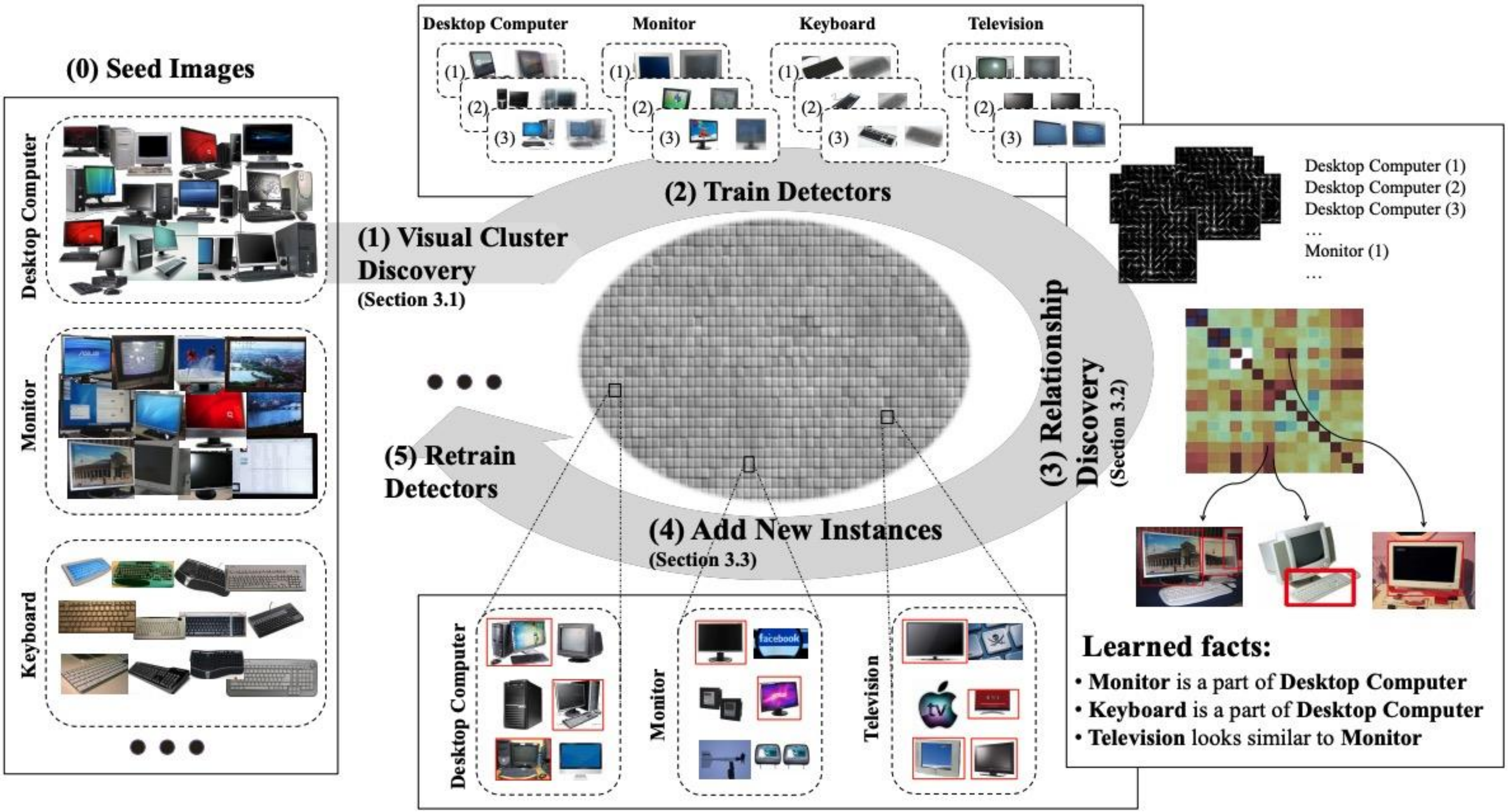
Self-Supervised Learning for UDA



Self-Supervised Learning for UDA

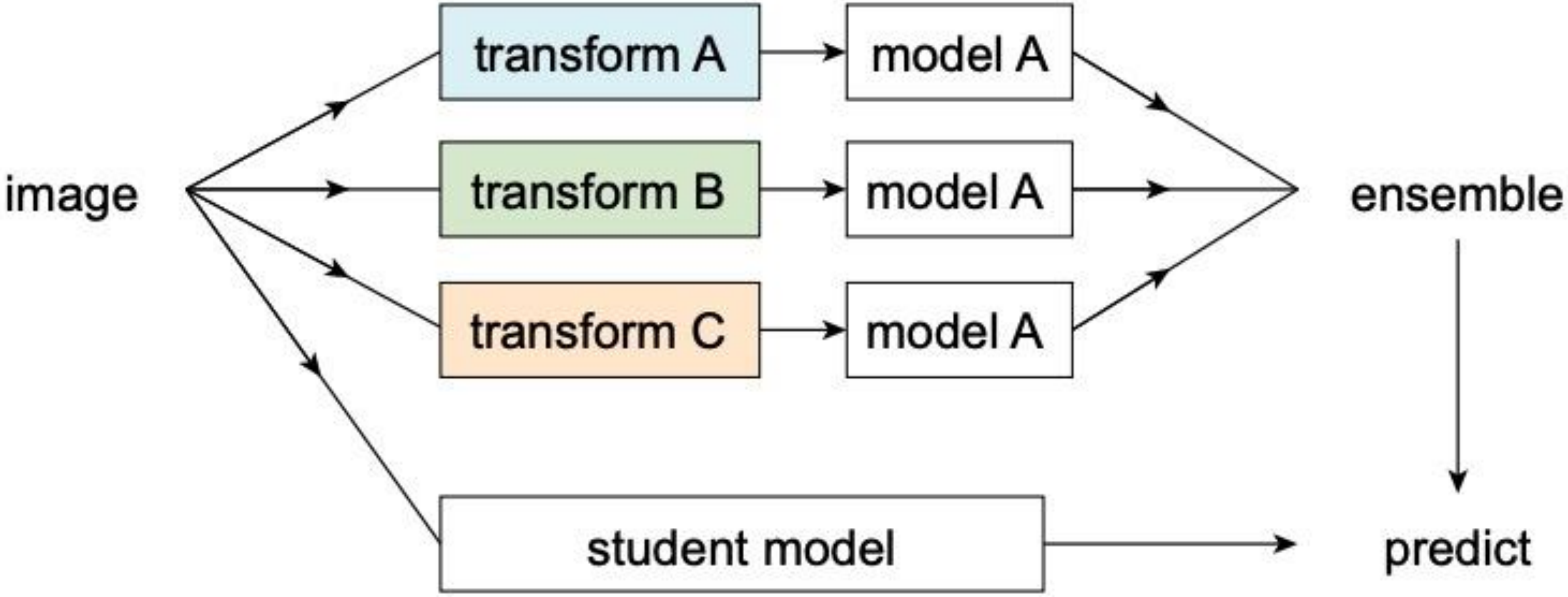
GTA5 → Cityscapes																				
	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bicycle	**mIoU**
Source only	28.8	12.7	39.6	9.4	3.5	18.1	22.7	9.4	80.9	12.4	45.8	53.9	9.6	74.7	20.9	15.0	0.0	19.4	3.9	25.3
Ours	69.9	22.7	69.7	18.1	9.9	13.5	18.7	8.9	80.3	19.4	58.4	53.8	2.6	75.1	13.6	5.2	0.3	8.1	1.2	28.9
CyCADA	79.1	33.1	77.9	23.4	17.3	32.1	33.3	31.8	81.5	26.7	69.0	62.8	14.7	74.5	20.9	25.6	6.9	18.8	20.4	39.5
Ours + CyCADA	86.6	37.8	80.8	29.7	16.4	28.9	30.9	22.2	83.8	37.1	76.9	60.1	7.8	84.1	30.8	32.1	1.2	23.2	13.3	41.2
Oracle	97.3	79.8	88.6	32.5	48.2	56.3	63.6	73.3	89.0	58.9	93.0	78.2	55.2	92.2	45.0	67.3	39.6	49.9	73.6	67.4

Semi-Supervised Learning



Chen et al. 2013

Data Distillation: Towards Omni-Supervised Learning



Data Distillation

Radosavovic et al. 2017

Data Distillation: Towards Omni-Supervised Learning

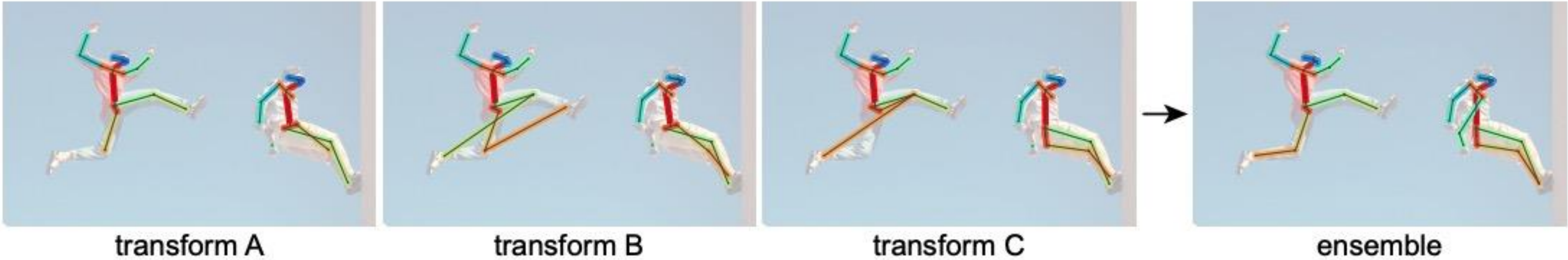


Figure 2. **Ensembling keypoint predictions from multiple data transformations can yield a single superior (automatic) annotation.** For visualization purposes all images and keypoint predictions are transformed back to their original coordinate frame.

This Class

- Multi-task Learning
- Domain Adaptation
- Semi-supervised Learning