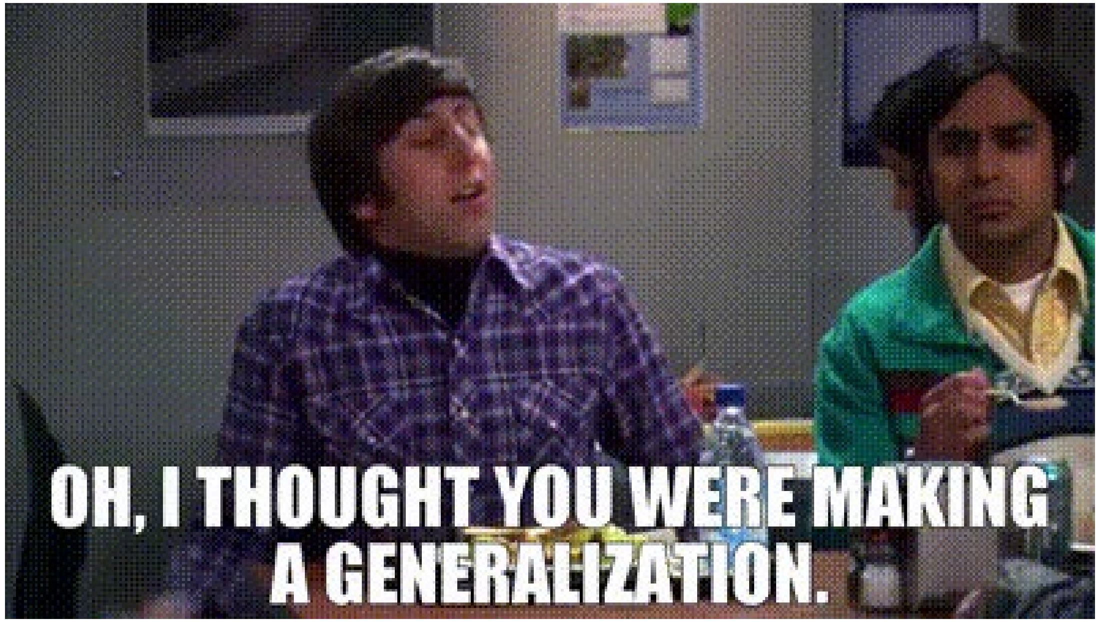
CMSC 491/691 Robust Machine Learning

Topic 2: Domain Generalization



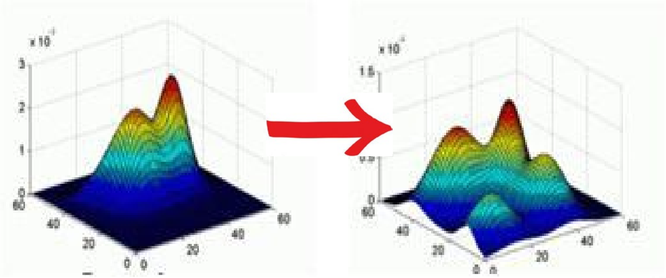
UMBC

Domain Adaptation and Generalization Visualization

(thanks to Tatiana Tomassi's ECCV 2020 Tutorial)

Classical Domain Adaptation

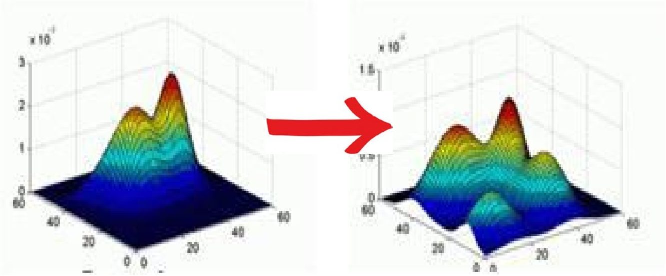
Source
(Train)



Target
(Test)

Classical Domain Adaptation

Source
(Train)

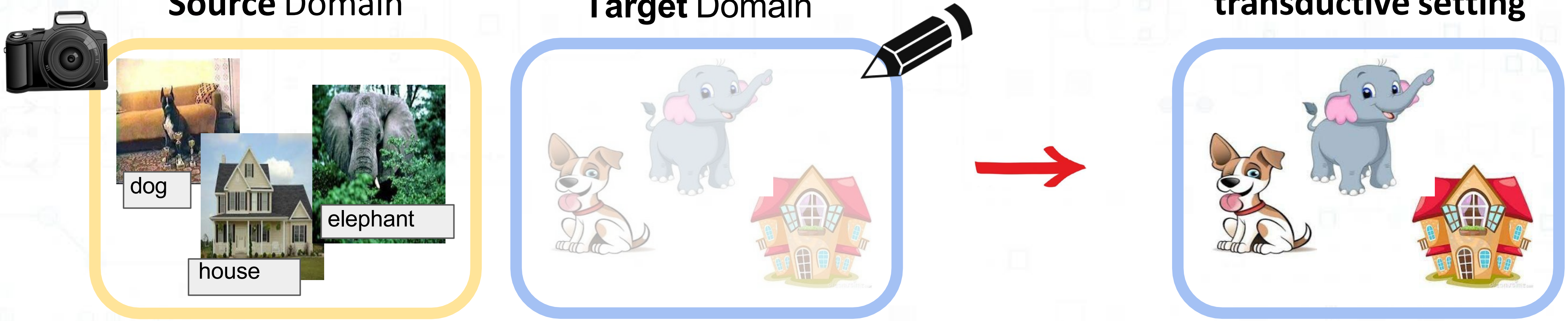


Target
(Test)

Labelled
Source Domain

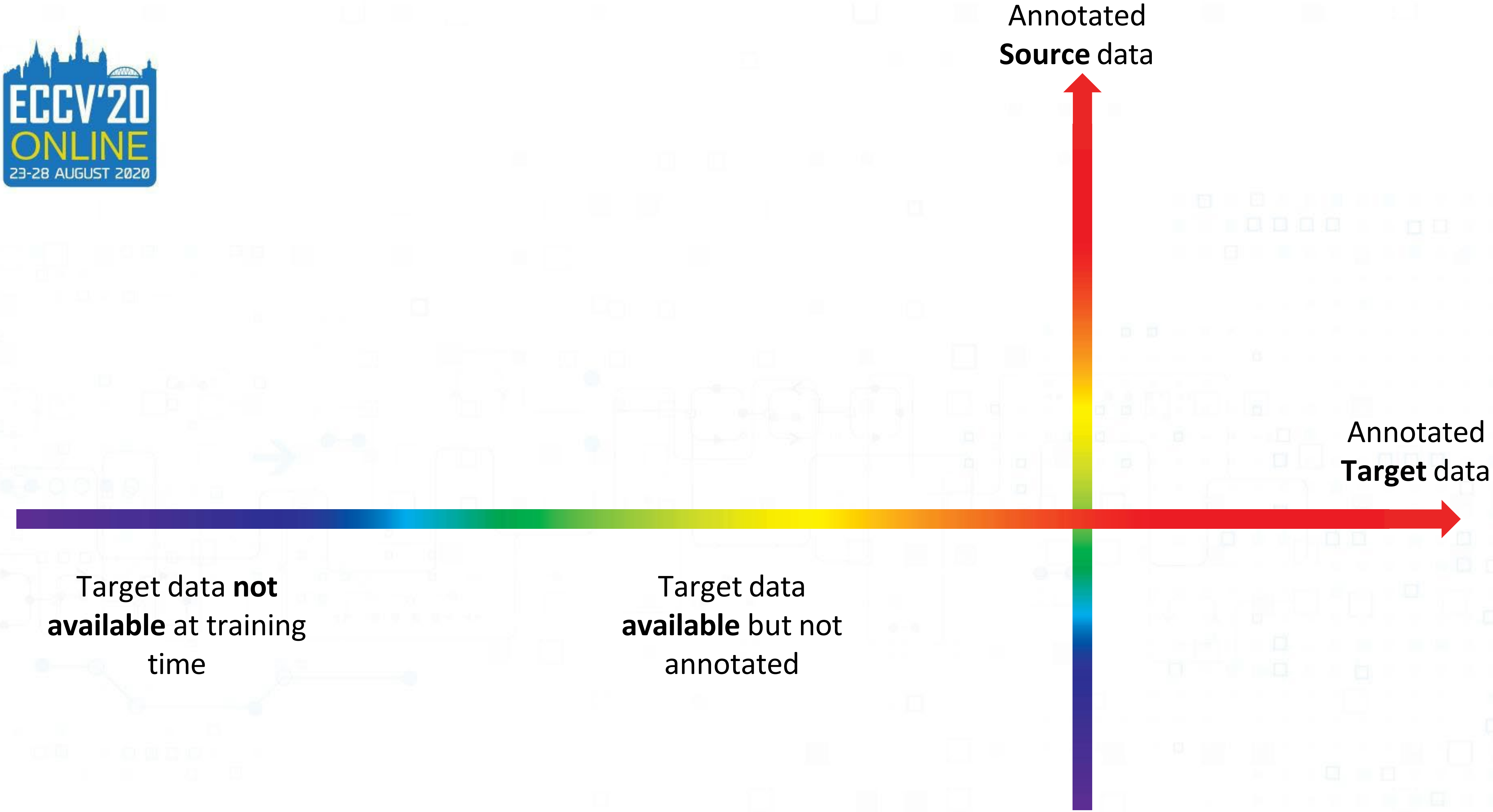
Unlabelled
Target Domain

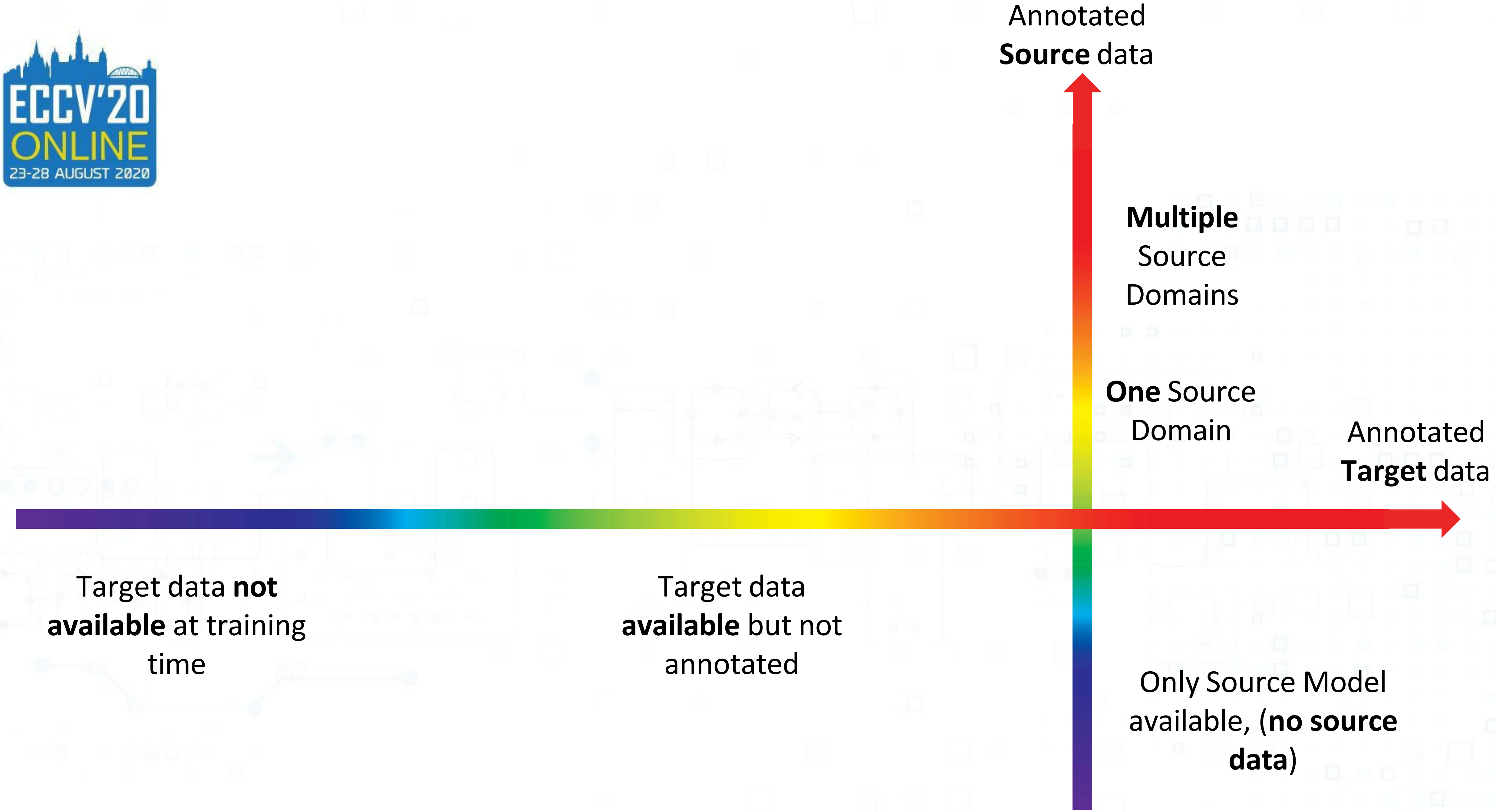
Unsupervised DA,
transductive setting



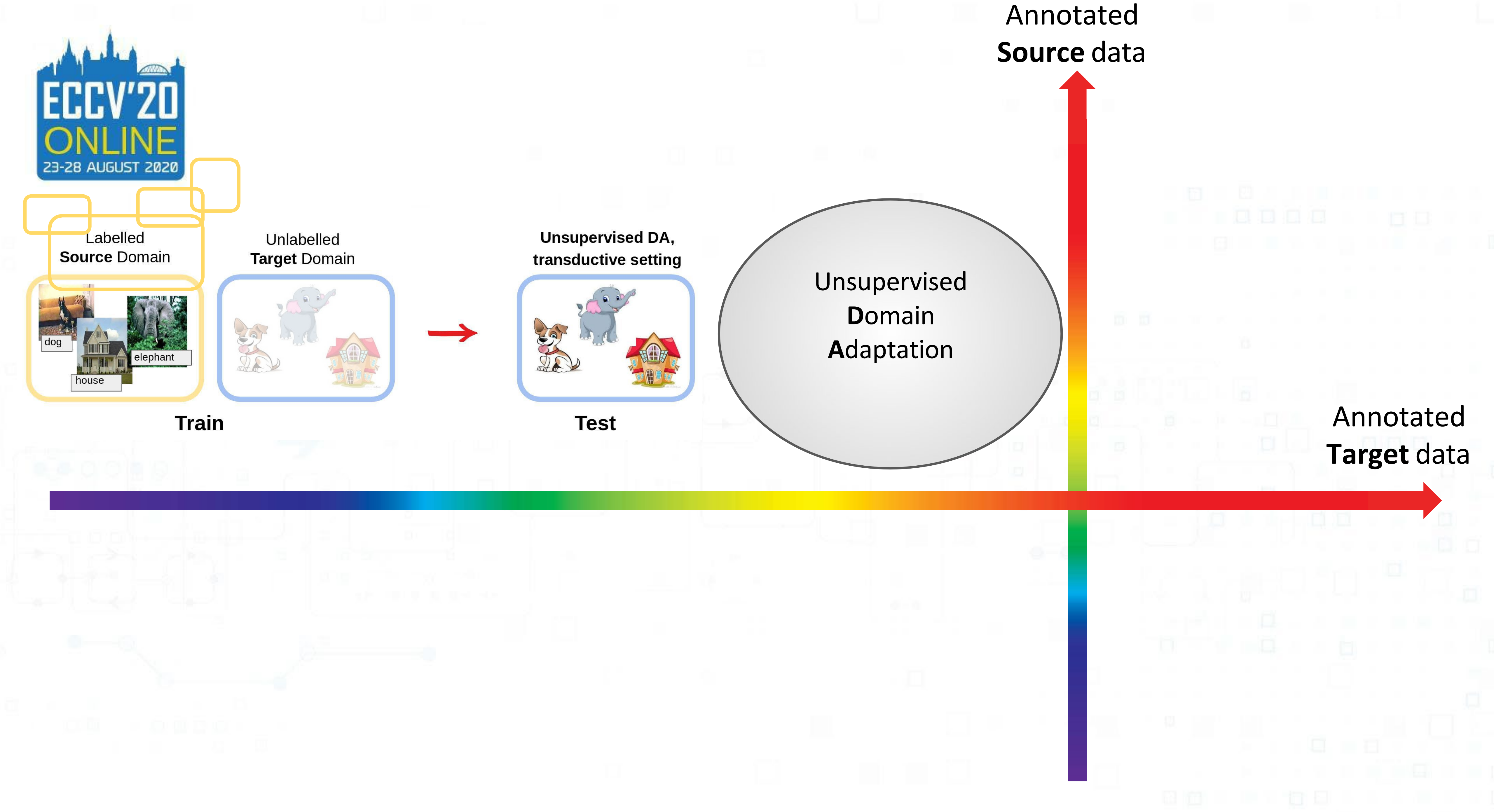
Train

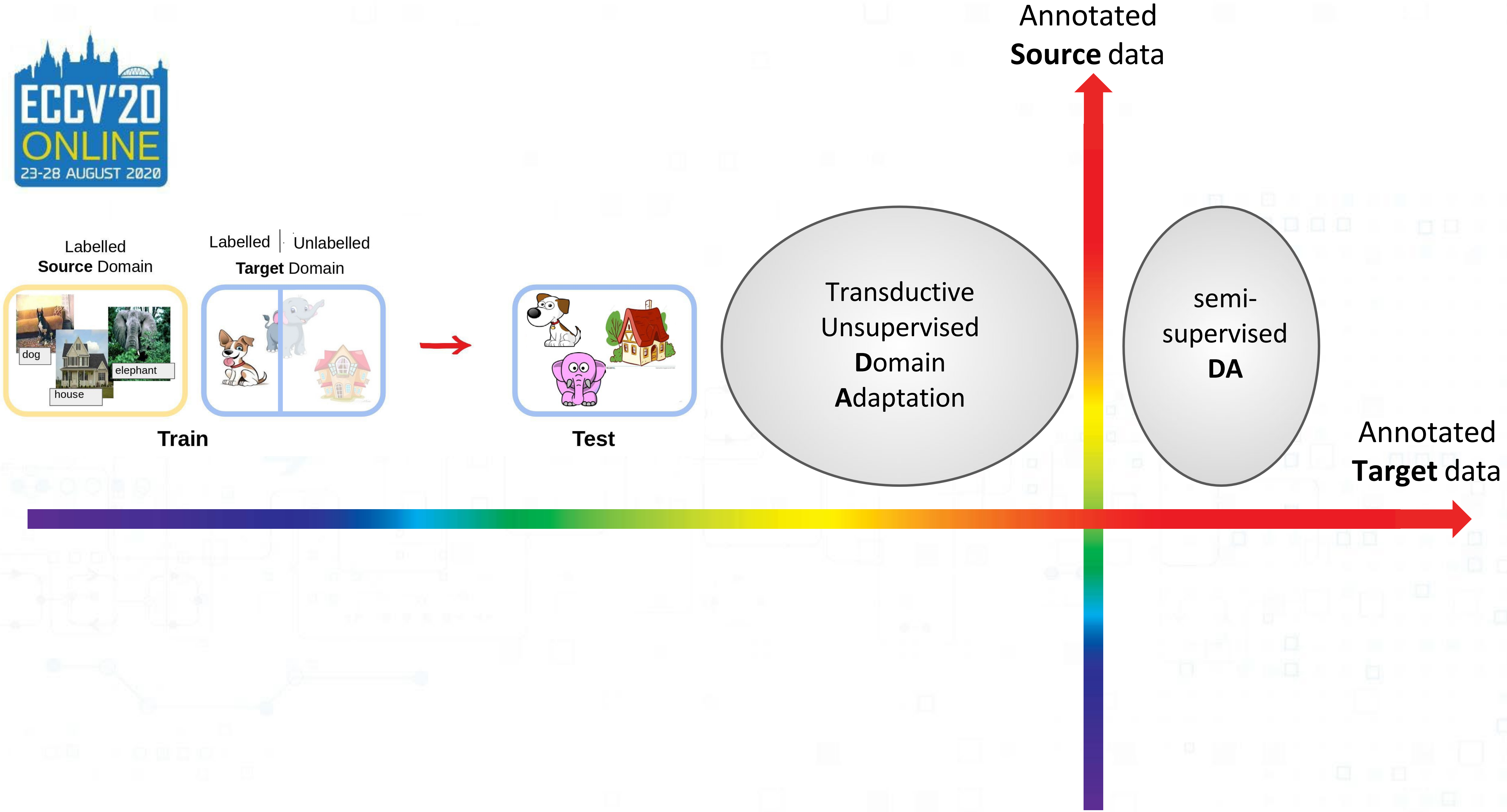
Test

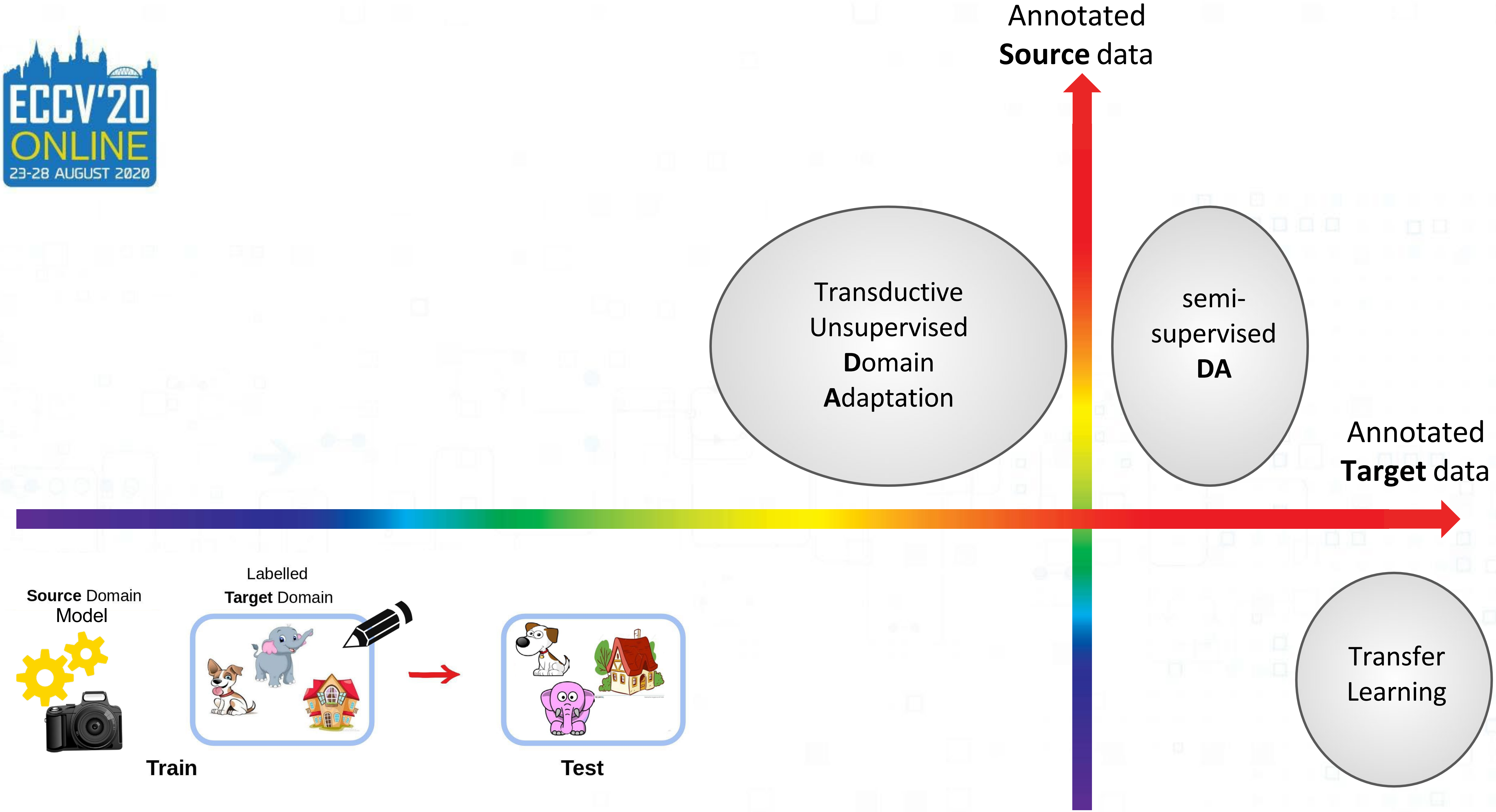

Annotated **Source** data
Annotated **Target** data
Target data **not available** at training time
Target data **available** but not annotated

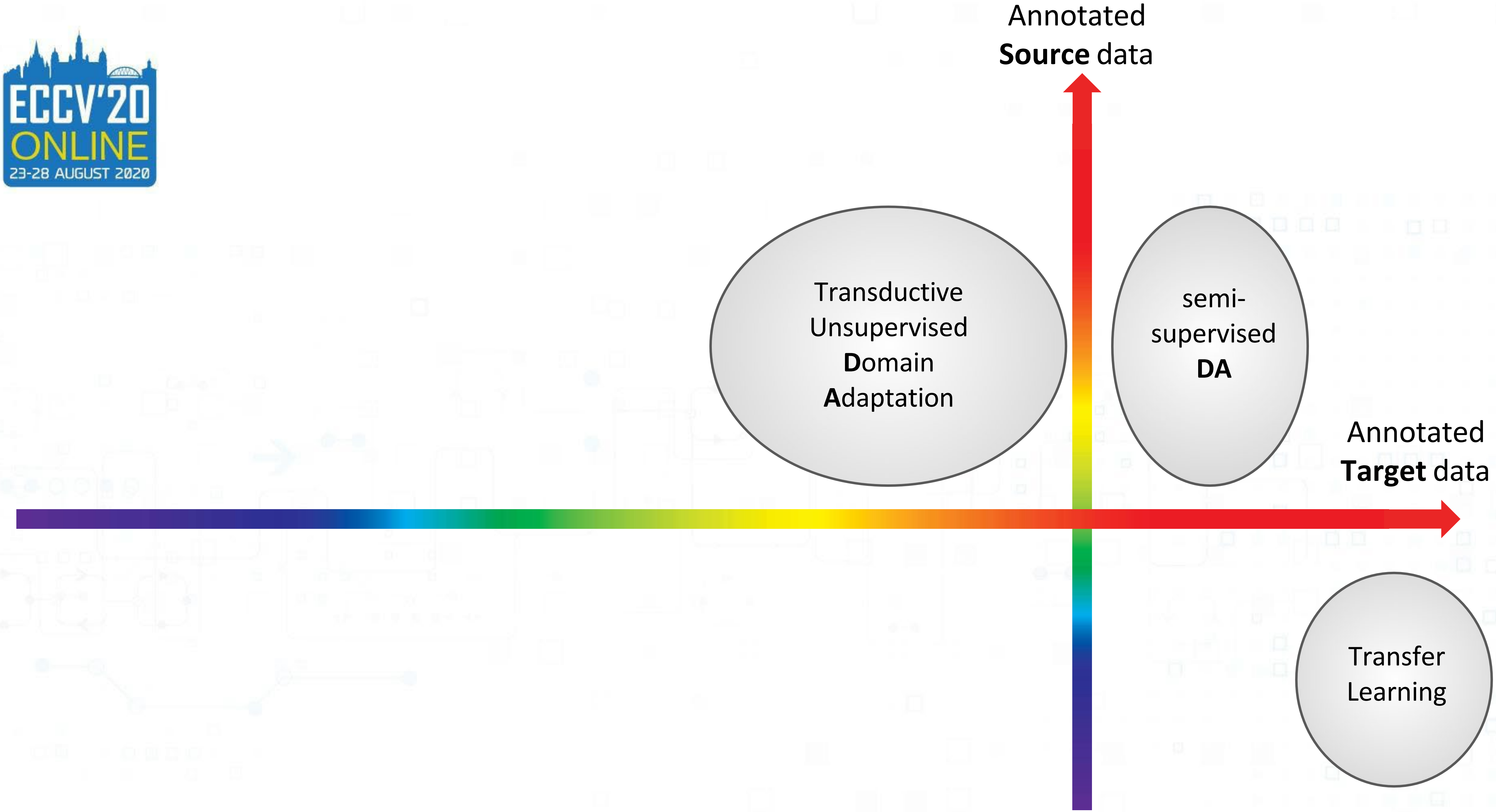


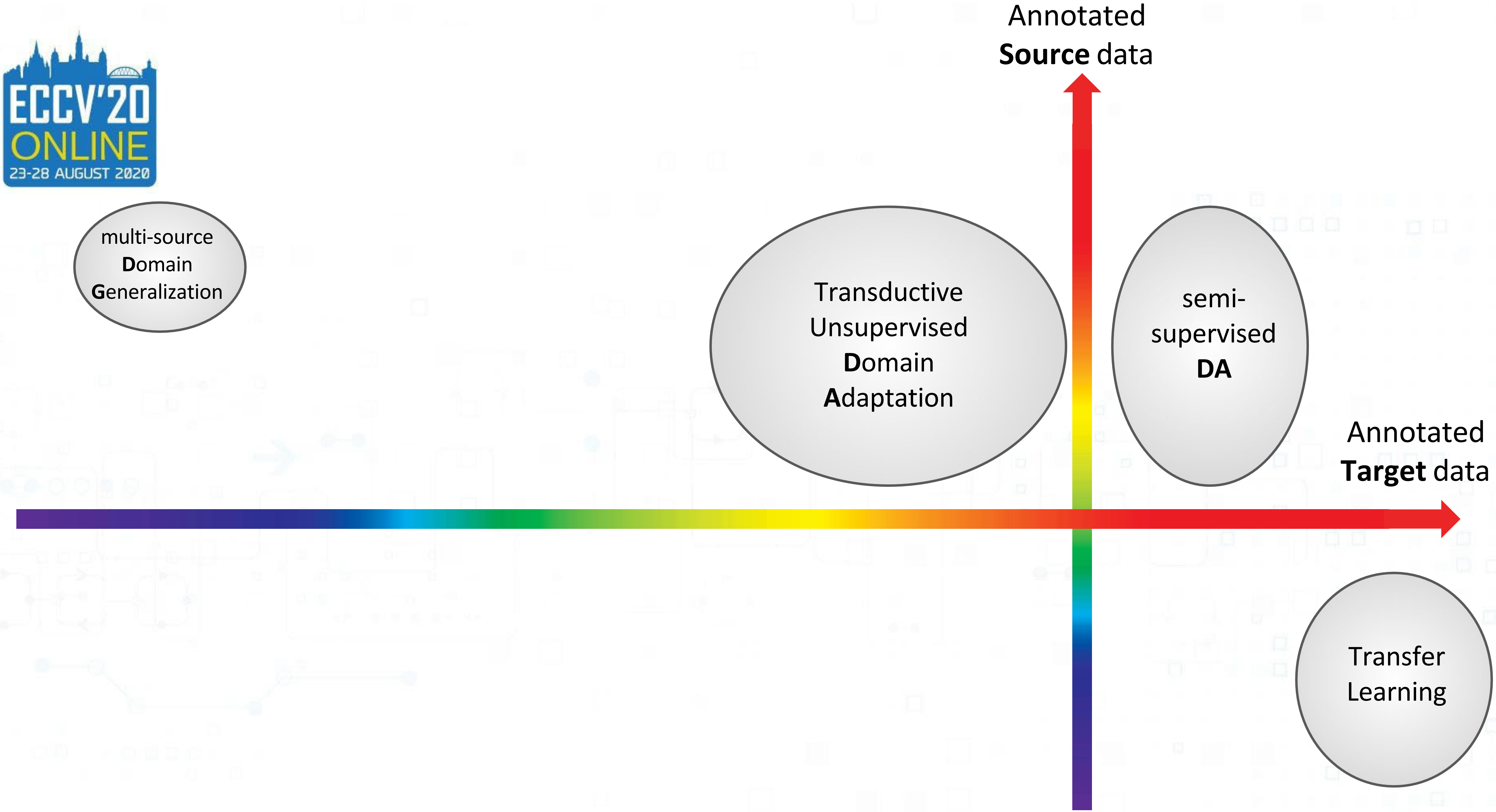
Annotated **Source** data
Multiple Source Domains
One Source Domain
Annotated **Target** data
Target data **not available** at training time
Target data **available** but not annotated
Only Source Model available, (**no source data**)


ECCV'20
ONLINE
23-28 AUGUST 2020
Labelled
Source Domain
dog
elephant
house
Unlabelled
Target Domain
Unsupervised DA,
transductive setting
Train
Test
Unsupervised
Domain
Adaptation
Annotated
Source data
Annotated
Target data

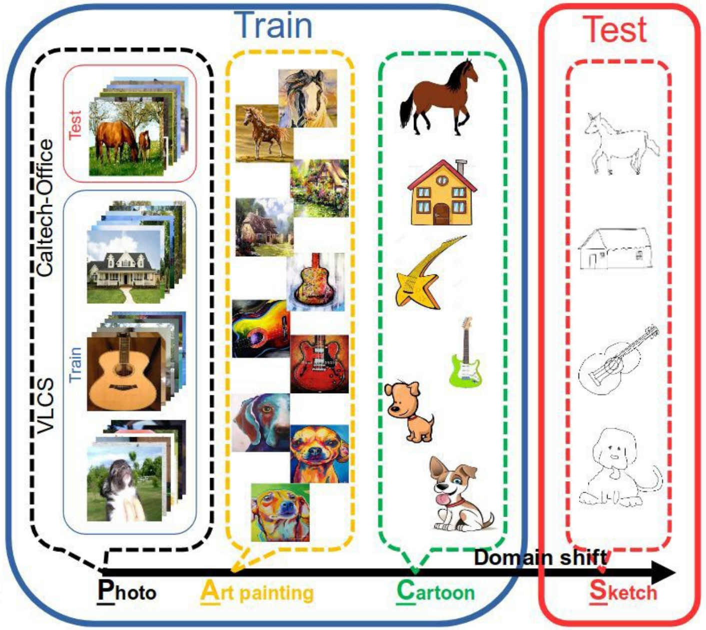

Labelled
Source Domain
Labelled
Unlabelled
Target Domain
dog
elephant
house
Train
Test
Transductive Unsupervised Domain Adaptation
semi-supervised DA
Annotated Source data
Annotated Target data


Annotated
Source data
Transductive
Unsupervised
Domain
Adaptation
semi-
supervised
DA
Annotated
Target data
Transfer
Learning
Source Domain
Model
Labelled
Target Domain
Train
Test


Annotated
Source data
Transductive
Unsupervised
Domain
Adaptation
semi-
supervised
DA
Annotated
Target data
Transfer
Learning


Annotated
Source data
multi-source
Domain
Generalization
Transductive
Unsupervised
Domain
Adaptation
semi-
supervised
DA
Annotated
Target data
Transfer
Learning

multi-source **D**omain **G**eneralization



[Deeper, Broader and Artier Domain Generalization, ICCV 2017]

Annotated **Source** data

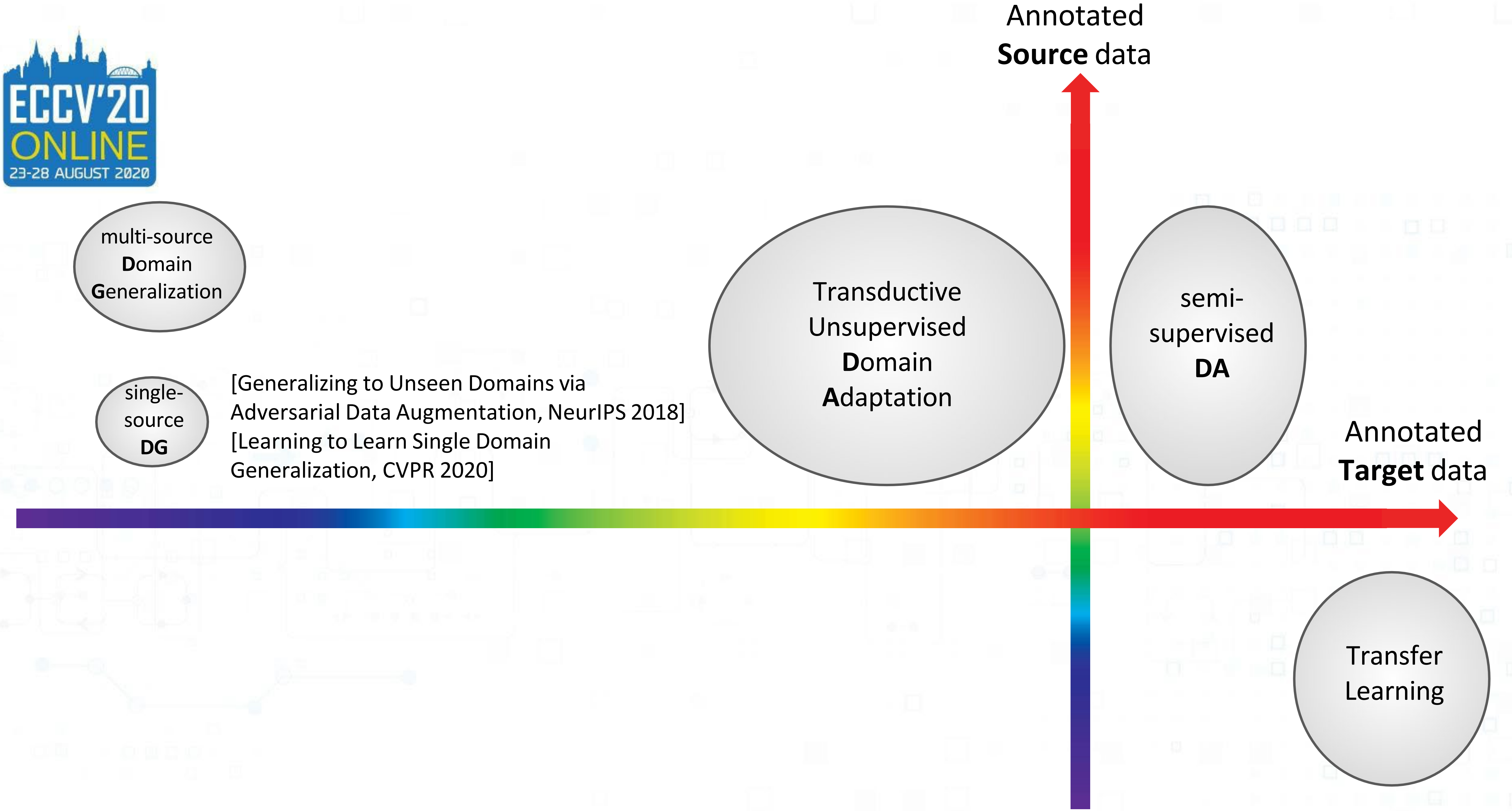
Transductive Unsupervised **D**omain **A**daptation

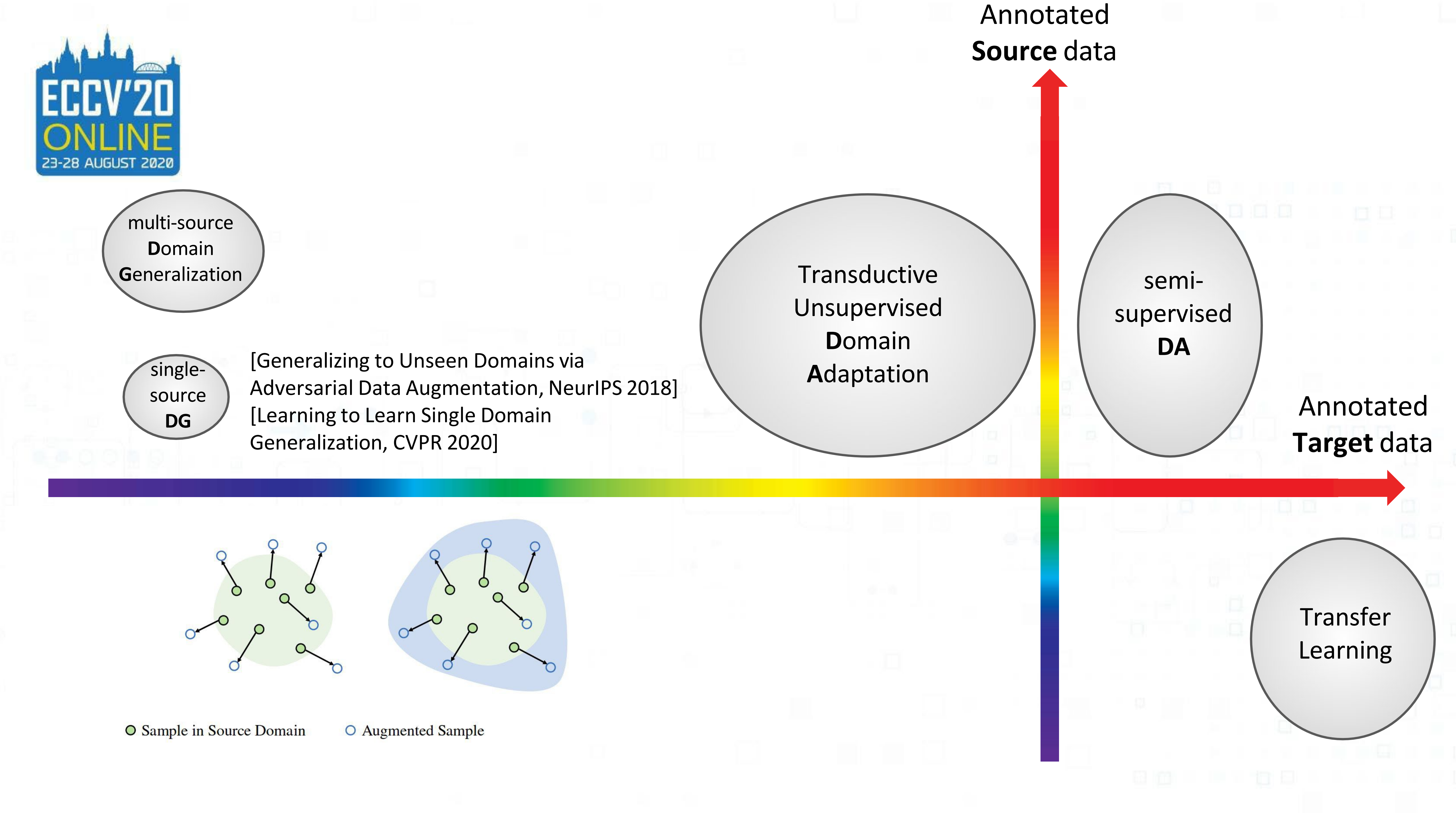
semi-supervised **DA**

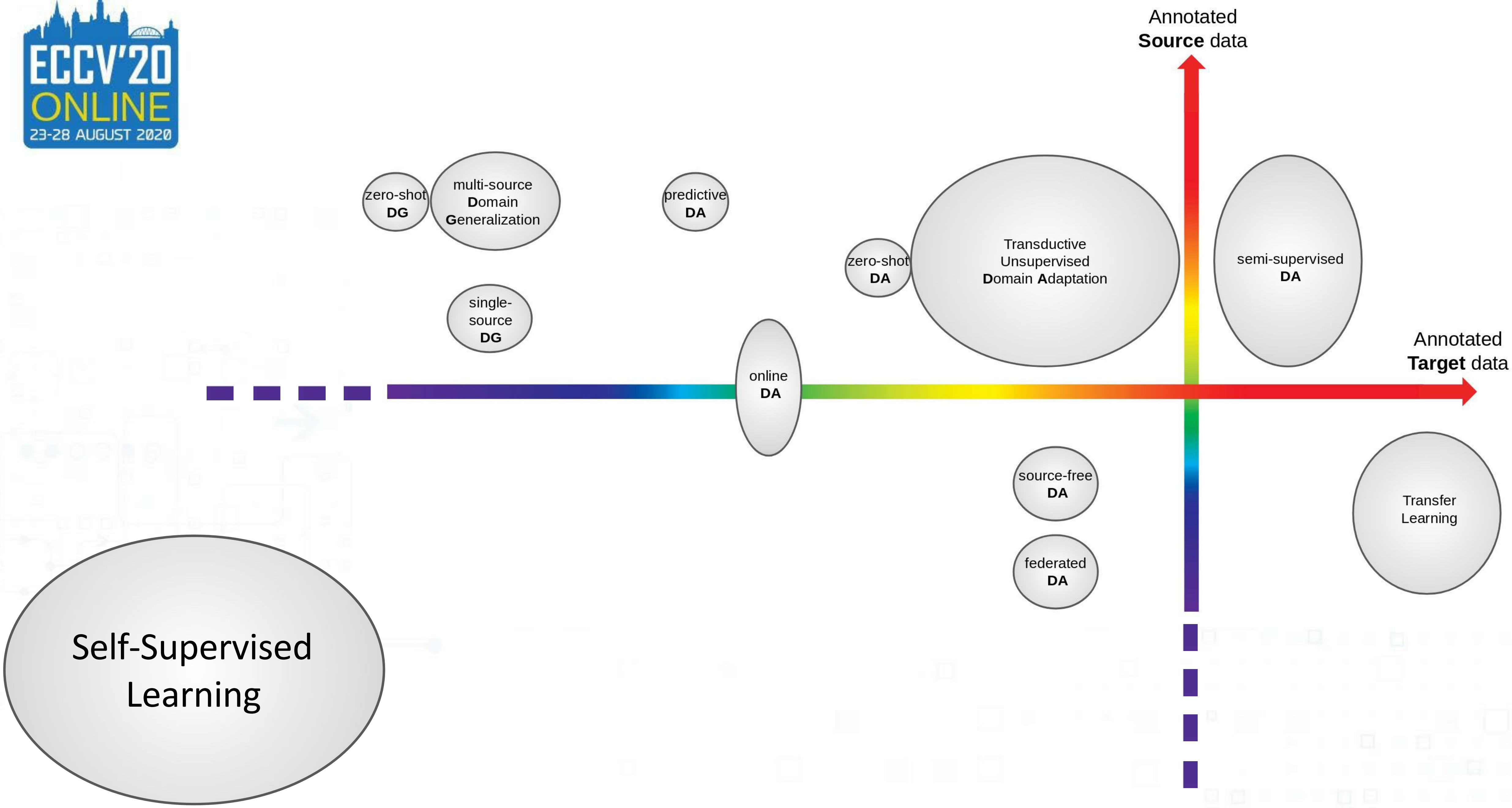
Annotated **Target** data

Transfer Learning

ECCV'20 ONLINE 23-28 AUGUST 2020



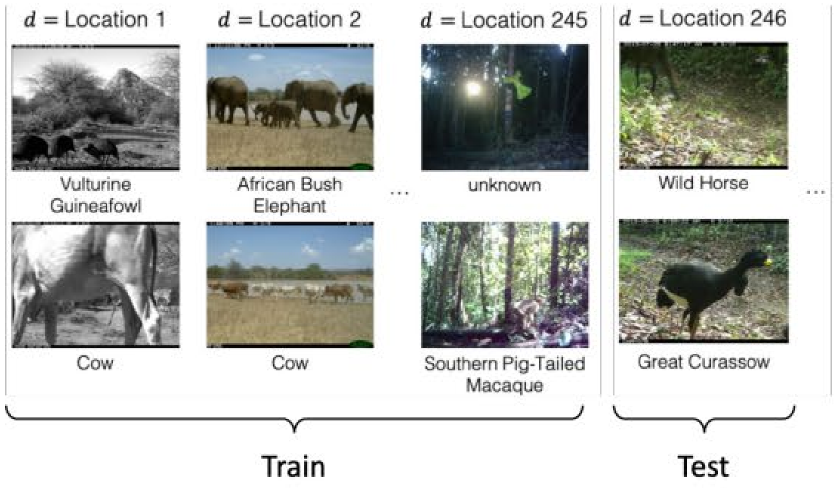

ECCV'20
ONLINE
23-28 AUGUST 2020
Annotated
Source data
multi-source
Domain
Generalization
Transductive
Unsupervised
Domain
Adaptation
semi-
supervised
DA
single-
source
DG
[Generalizing to Unseen Domains via
Adversarial Data Augmentation, NeurIPS 2018]
[Learning to Learn Single Domain
Generalization, CVPR 2020]
Annotated
Target data
Transfer
Learning
Sample in Source Domain
Augmented Sample



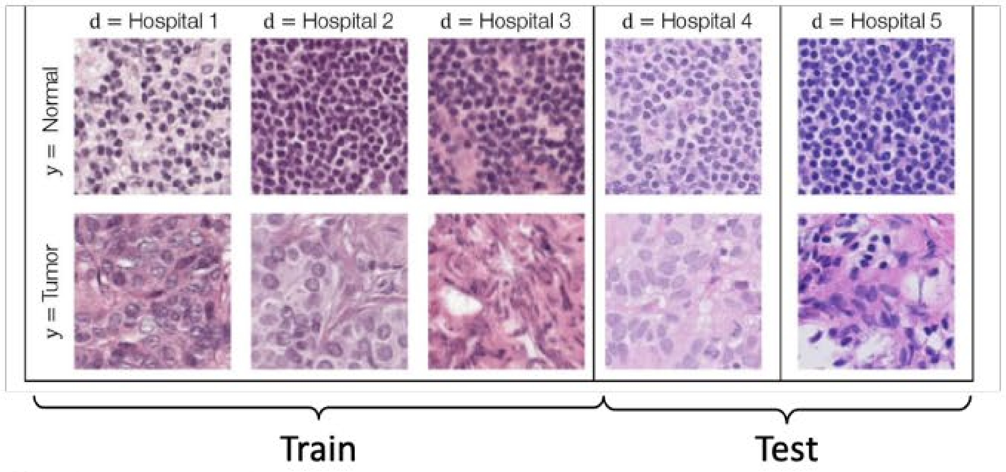
Annotated
Source data
Annotated
Target data
zero-shot
DG
multi-source
Domain
Generalization
predictive
DA
zero-shot
DA
Transductive
Unsupervised
Domain Adaptation
semi-supervised
DA
single-
source
DG
online
DA
source-free
DA
federated
DA
Transfer
Learning
Self-Supervised
Learning

Domain Generalization: Applications

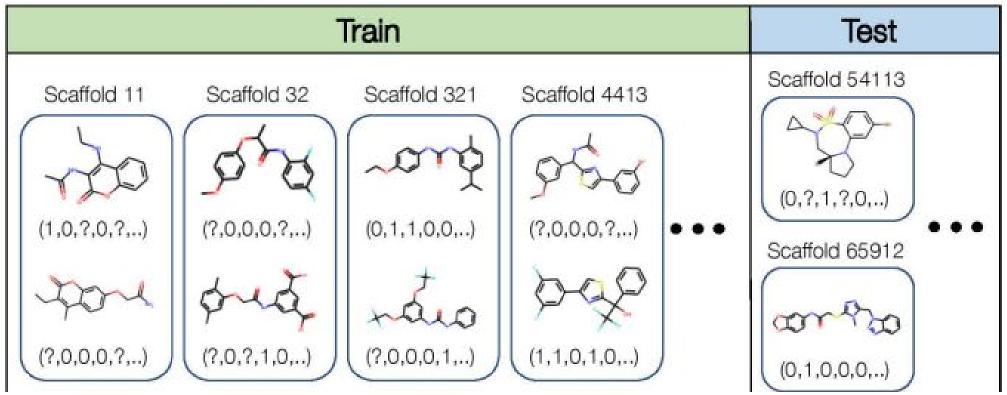
Wildlife recognition



Tissue classification



Molecule property prediction



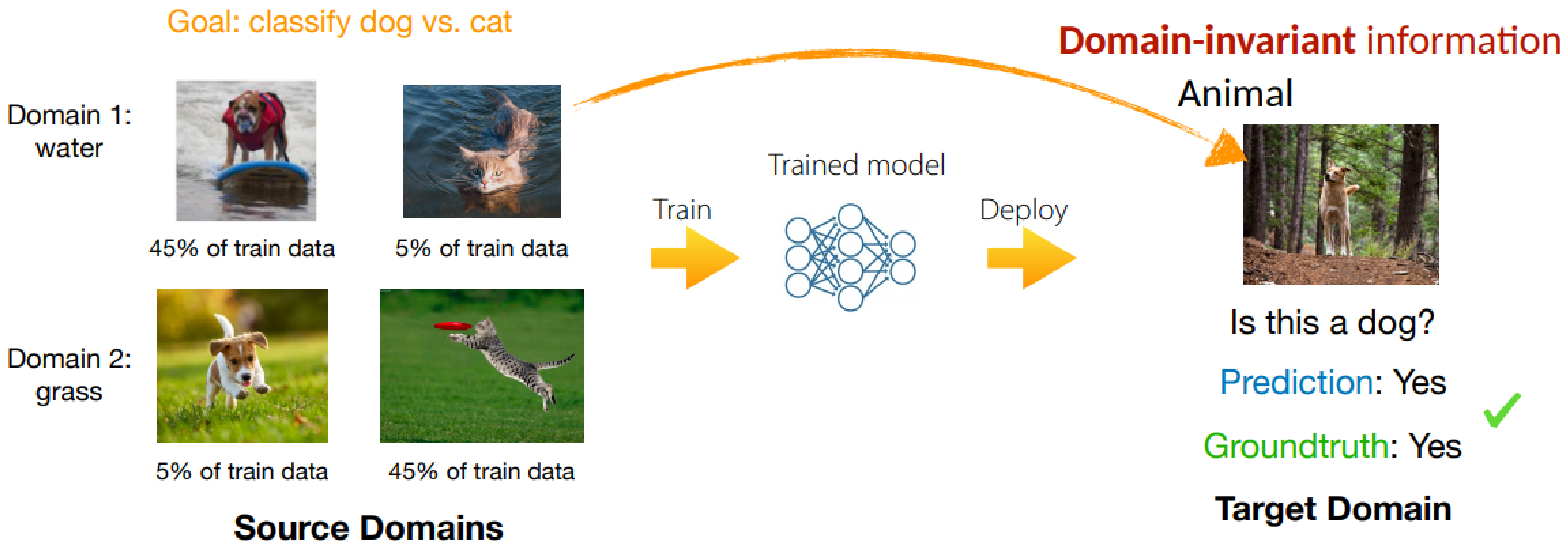
Code completion

	Repository ID (d)	Source code context (x)	Next tokens (y)
Train	Repository 1	... from easyrec.gateway import EasyRec <EOL> gateway = EasyRec('tenant','key') <EOL> item_type = gateway. ___	get_item_type
		... response = gateway.get_other_users() <EOL> get_params = HTTPretty. ___	last_request
	Repository 2	import numpy as np ... <EOL> if np.linalg.norm(target - prev_target) > far_threshold: <EOL> norm = np. ___	linalg
		... new_trans = np.zeros((n_beats + max_beats, n_beats) <EOL> new_trans[:n_beats,:n_beats] = np. ___	max
	⋮		
Test	Repository 6,001	... if e.errno == errno.ENOENT: <EOL> continue <EOL> p = subprocess.Popen () <EOL> stdout = p. ___	communicate
		... command = shlex.split(command) <EOL> command = map(str, command) <EOL> env = os. ___	environ
	⋮		

How to Learn Generalizable Representations?

To overcome spurious correlation —> train a neural network to learn **domain invariance**

Domain invariance: we want to learn features that don't change across domains

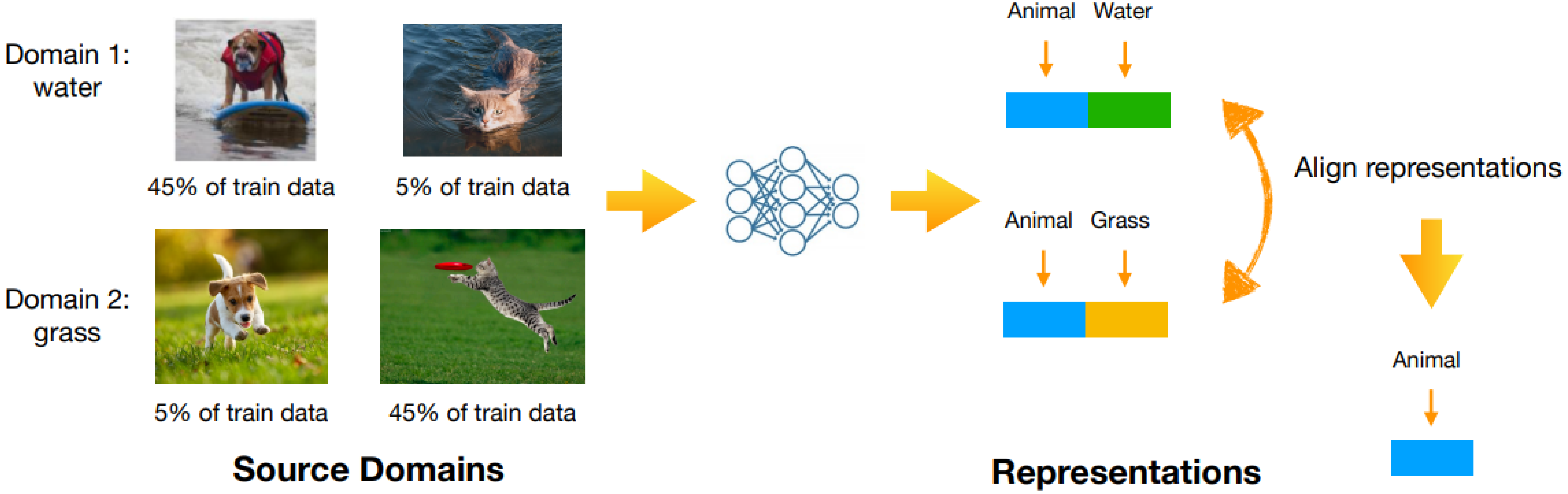


Idea #1 Regularization

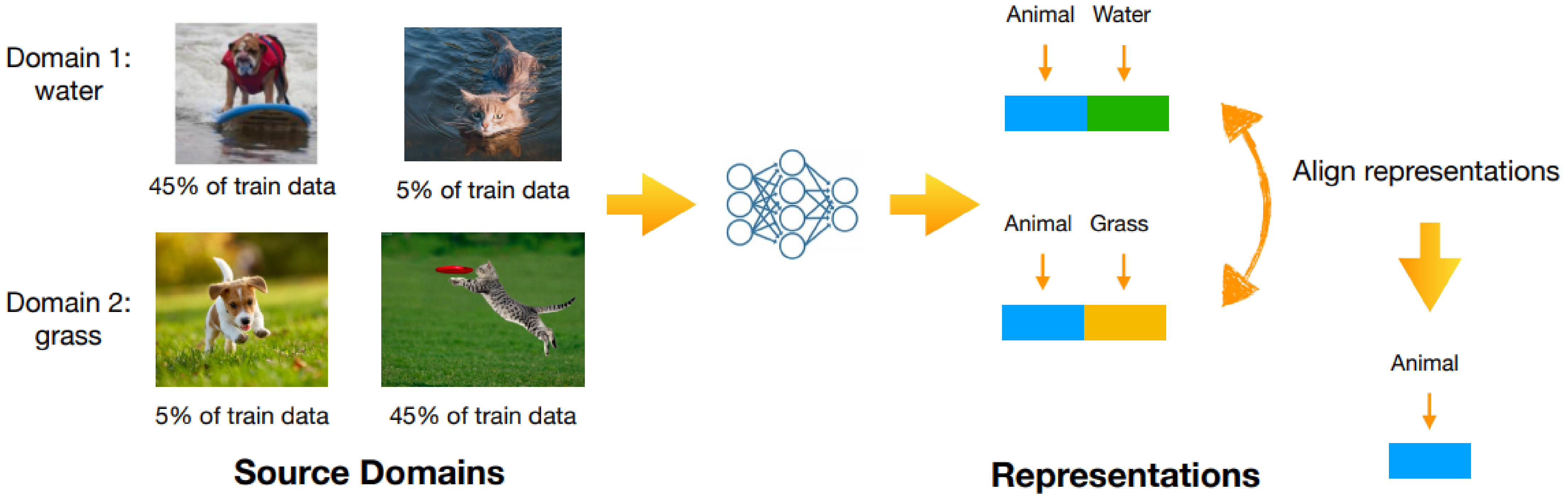
Regularization-based Method

Key idea: Use a regularizer to align representations across domains

—> get domain-invariant representation



Regularization-based Method



Label classification loss

$$\min_{\theta} \mathbb{E}_{(x,y)}[\ell(f_\theta(x), y)] + \lambda \mathscr{L}_{reg}$$

Explicit regularizer to learn domain-invariant representation

Average over training examples

Domain Adversarial Training
(one of the student presentations)

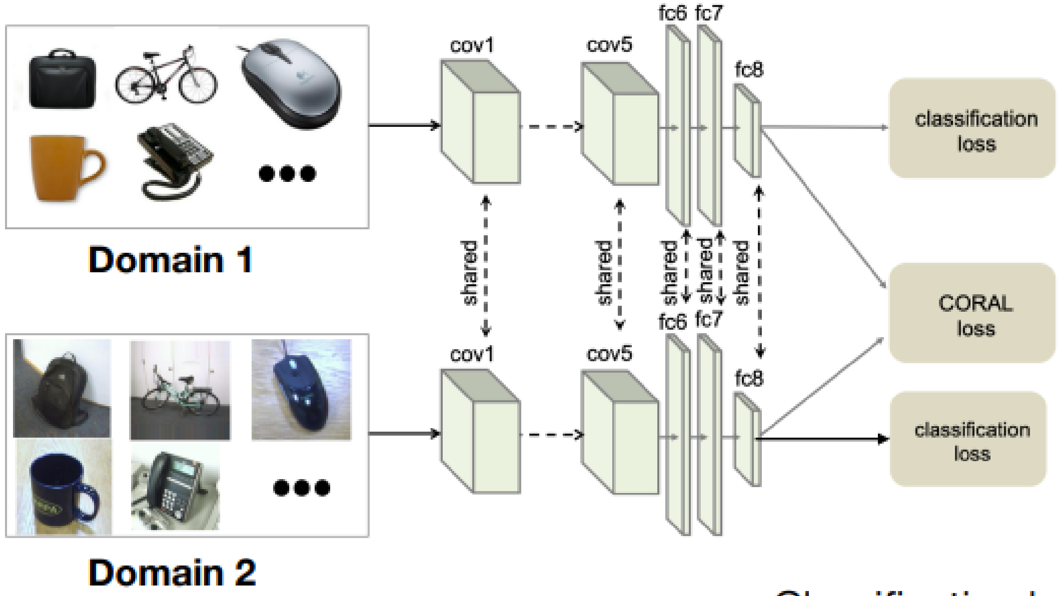
Tzeng et al. Deep Domain Confusion. arXiv '14

Ganin et al. Domain-Adversarial Training of Neural Networks. JMLR '16

Alternative Approach — CORAL

Key idea: directly aligning representations between different domains with some similarity metrics

CORAL: Correlation Alignment for Domain Adaptation (usually also used in DG)



Notations $\mathbf{X}_1 \in \mathbb{R}^{n_1 \times k}$ $\mathbf{X}_2 \in \mathbb{R}^{n_2 \times k}$

k: num of features $\mu_1 = \frac{1}{n_1}\mathbf{1}^T\mathbf{X}_1 \in \mathbb{R}^{1\times k}$ $\mu_2 = \frac{1}{n_2}\mathbf{1}^T\mathbf{X}_2 \in \mathbb{R}^{1\times k}$

Calculate covariance matrices

$$C_1 = \frac{1}{n_1 - 1}\sum_{i=1}^{n_1}(\mathbf{X}_1 - \mu_1)^T(\mathbf{X}_1 - \mu_1)$$

$$C_2 = \frac{1}{n_2 - 1}\sum_{i=1}^{n_2}(\mathbf{X}_2 - \mu_2)^T(\mathbf{X}_2 - \mu_2)$$

CORAL loss

$$\mathscr{L}_{coral} = \frac{1}{4k^2}\|C_1 - C_2\|_F^2$$

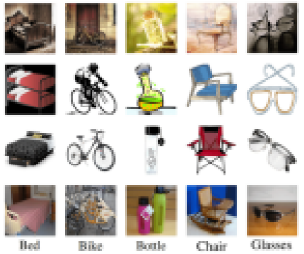
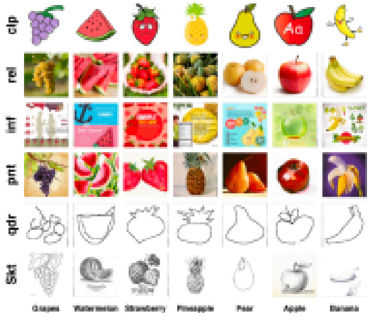
Classification loss

$$\mathscr{L} = \sum_{j=1}^{n_1+n_2}\mathscr{L}_c(f_\theta(x_i), y_i) + \lambda\mathscr{L}_{coral}$$

Explicit regularizer to learn domain-invariant representation

Sun et al. Correlation Alignment for Deep Domain Adaptation. arXiv '16

Results

		ERM	CORAL	DANN
OfficeHome		66.5%	**68.7%**	65.9%
DomainNet		40.9%	**41.5%**	38.3%
iWildCam		30.8%	**32.7%**	n/a

Idea #2 Data Augmentation

Recap: Spurious Correlation

Recap: spurious correla5on between domains and labels

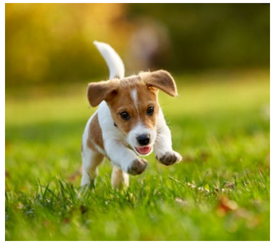
Goal: classify dog vs. cat

Domain 1: water

45% of train data

5% of train data

Domain 2: grass



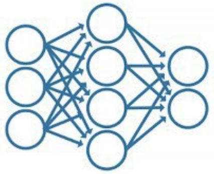
5% of train data

45% of train data

Source Domains

Train

Trained model



Deploy

Spurious informa5on

Grass

Is this a dog?

Prediction: No

Groundtruth: Yes ✗

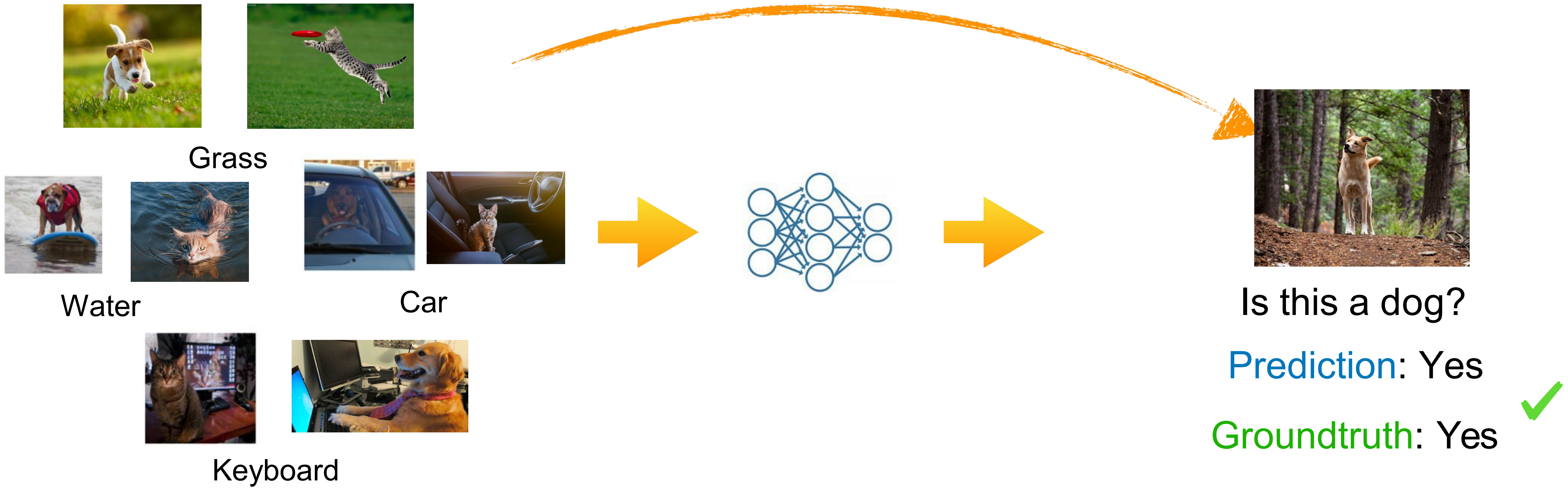
Target Domain

Data Augmentation

If we can collect more data

Question: Will the network still associate dogs with water background in source domains?

NO! There are many more backgrounds. We can't recognize dogs only with grass background.



Source Domains

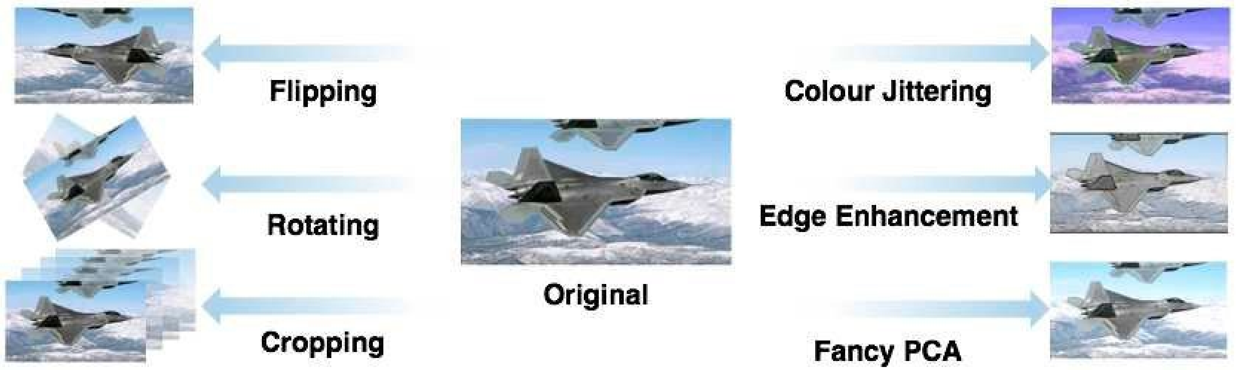
Target Domain

Challenge

We can not collect more data →→ Let's generate data!

Data Augmentation

Generating data with **simple operators**



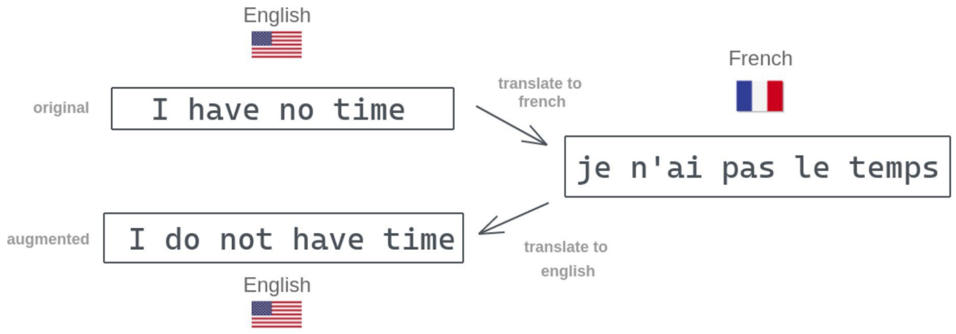


Figure: Back Translation

Requires knowledge of the problem domain

Any general approaches?

https://amitness.com/2020/02/back-translation-in-google-sheets/

Data Augmentation — Mixup

Interpolating training examples

A learning model $\mathcal{D}_{tr} = \{x_i, y_i\}_{i=1}^{N} \rightarrow$ Classifier,

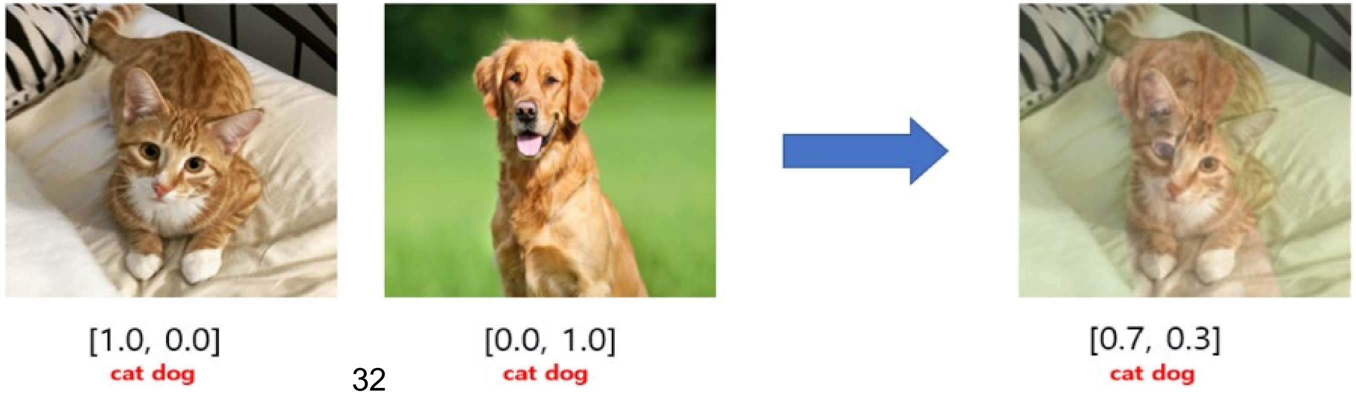
Mixup $\widetilde{\mathcal{D}}_{tr} = \{\tilde{x}_i, \tilde{y}_i\}_{i=1}^{N} \rightarrow$ Classifier,

where

$$\tilde{x}_i = \lambda x_i + (1-\lambda) x_j, \tilde{y}_i = \lambda y_i + (1-\lambda) y_j$$

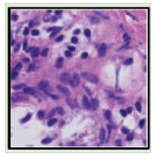
$$\lambda \sim \mathrm{Beta}(\alpha, \beta)$$

Generating some virtual examples between two classes

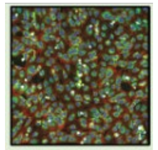


Data Augmentation — Mixup

Mixup can improve the performance on domain generalization

	Empirical Risk Minimiza1on	mixup
 Camelyon17	70.3%	71.2%
 FMoW	32.8%	34.2%

But it is not always good!

	Empirical Risk Minimiza1on	mixup
 RxRx1	29.9%	26.5%

Original mixup only focuses on data augmentation instead of learning domain invariance.

How to Improve it?

Regularization-based v.s. Augmentation-based Methods

Regularization-based Method

+ General to all kinds of data and networks

+ Some theoretical guarantee

- Rely on the design of regularizers

Augmentation-based Method

+ Easy to understand and simple to implement

+ No need to worry about how to design regularizers

- Largely limited to classification

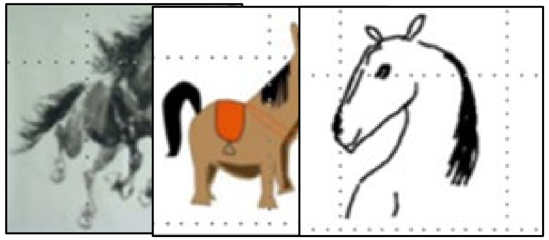
Discovering Adversarial Data Augmentation

(Series of Work by Tejas)

Problem Setting: Single Source Domain Generalization

How can classifiers trained on one domain generalize to other ***unseen domains?***

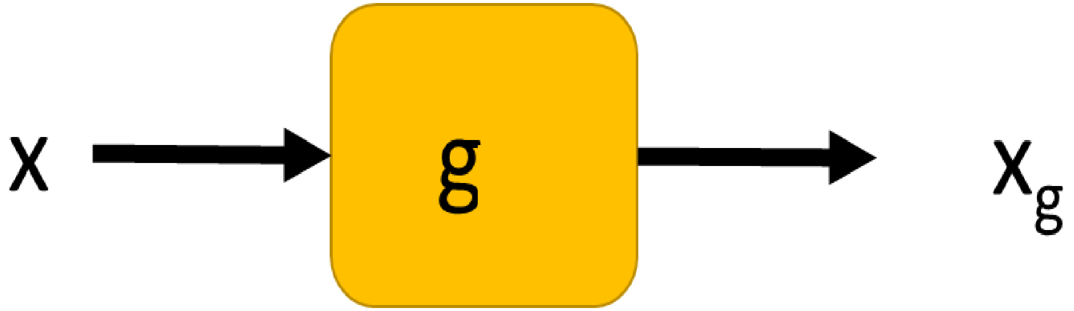
- Given:  Labeled training data from “Source” Domain

- ***Not Given:***  Target examplars; knowledge of unseen domains

SSDG: Single Source Domain Generalization

How can classifiers trained on one domain generalize to other ***unseen domains?***

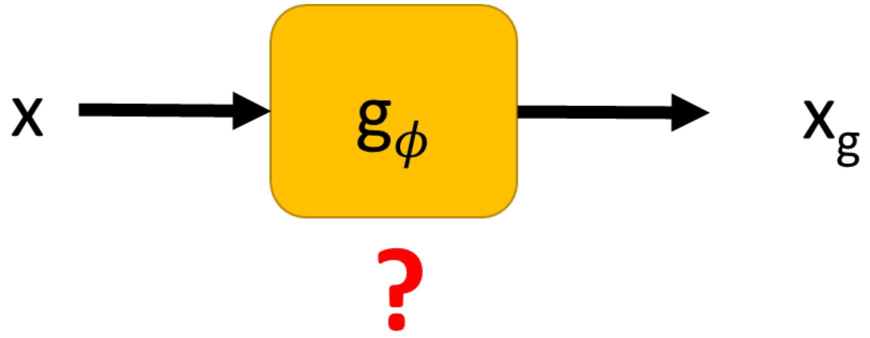
- For SSDG, Data Augmentation is crucial !!!
  - *To increase diversity of training data*
  - *To simulate new domains*
  - *To cover distributions that may be encountered at test-time*



Data Augmentation is Crucial … But which augmentation?

How can classifiers trained on one domain generalize to other ***unseen domains?***

- For SSDG, Data Augmentation is crucial !!!
  - *To increase diversity of training data*
  - *To simulate new domains*
  - *To cover distributions that may be encountered at test-time*

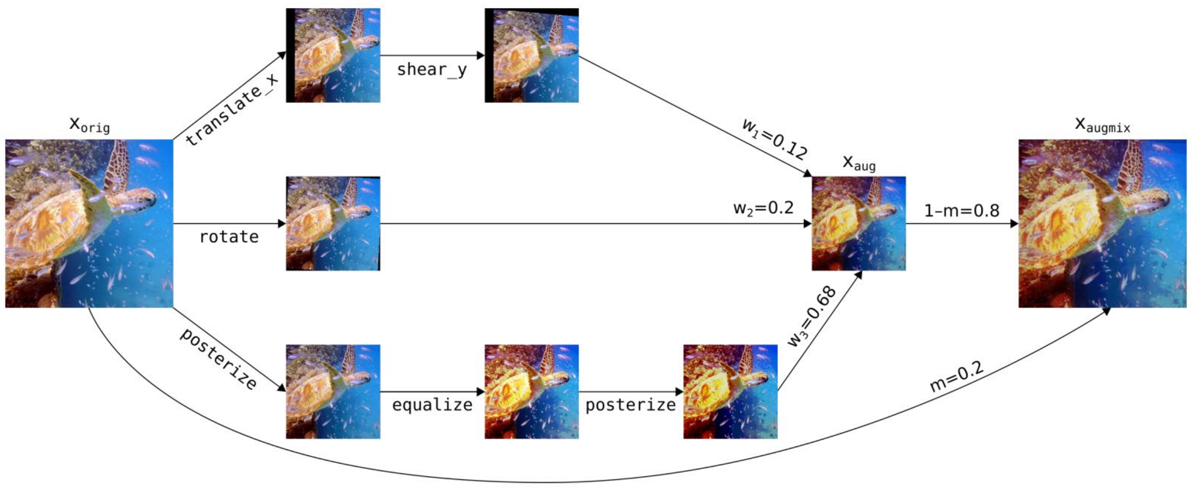


How do we know which data augmentations will be useful?
(We don’t have access to the test domains)

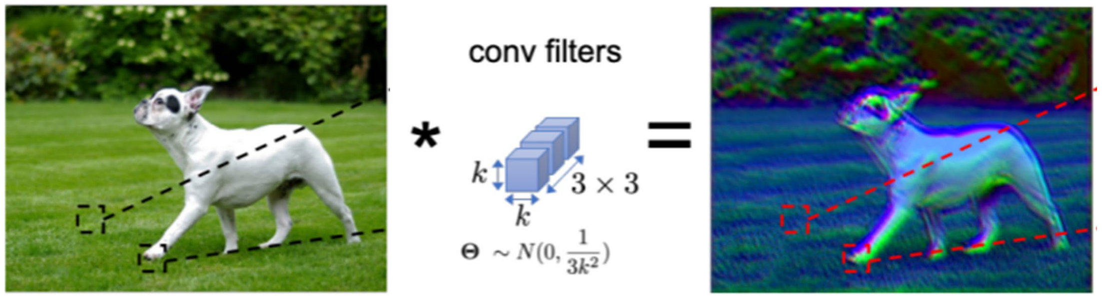
Data Augmentation is Crucial … But which augmentation?

- Existing Data Augmentation techniques
  - Introduce a strong preference towards certain types of diversity

AugMix *(Hendrycks et al. ICLR 2019)*
Combination of geometric transforms and image filters



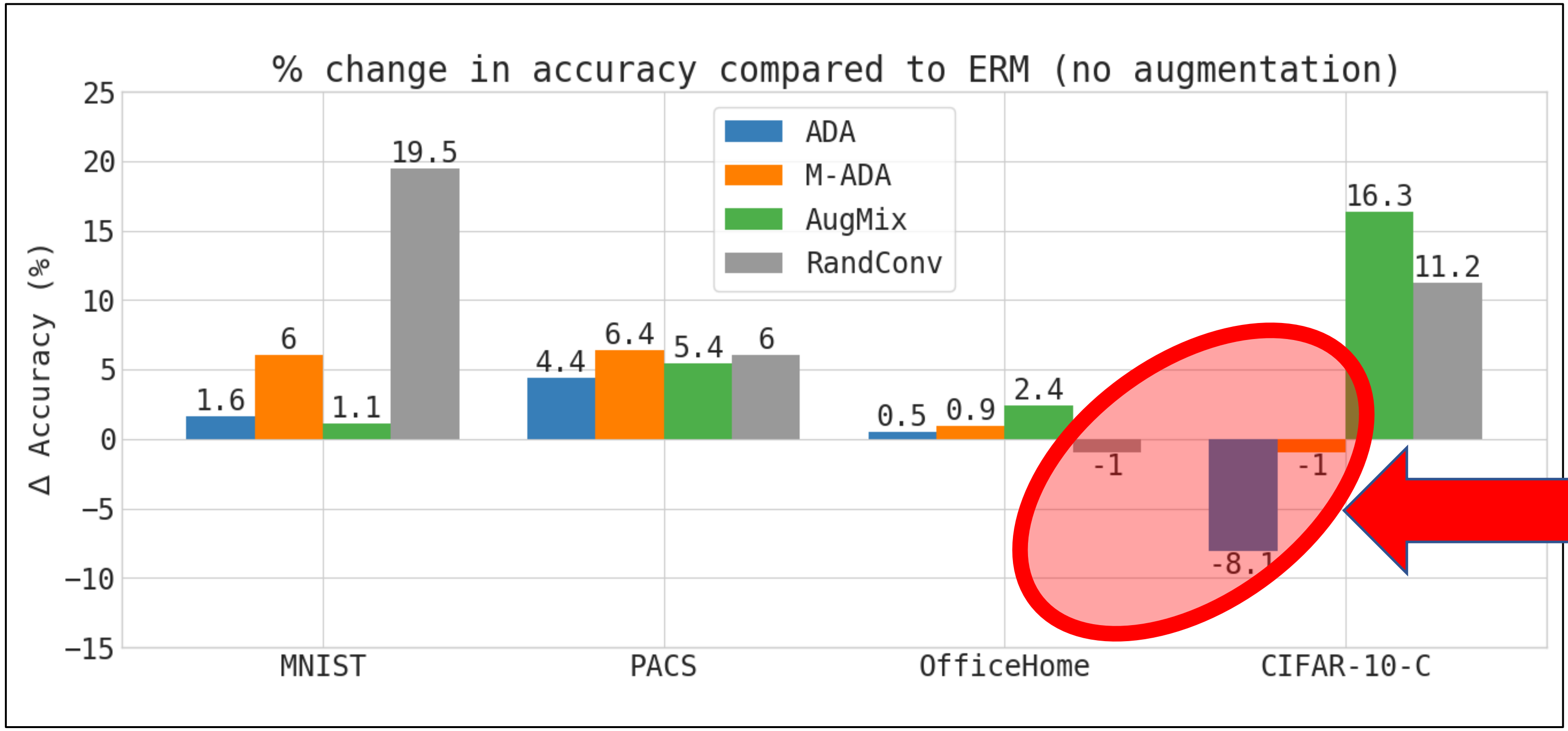
RandConv *(Xu et al. ICLR 2021)*
Convolve image with random filter



AugMix: Hendrycks et al. ICLR 2019; RandConv: Xu et al. ICLR 2020

Data Augmentation is Crucial … But which augmentation?

- Existing Data Augmentation techniques
  - Introduce a strong preference towards certain types of diversity
  - Mixed results on different domain shift datasets



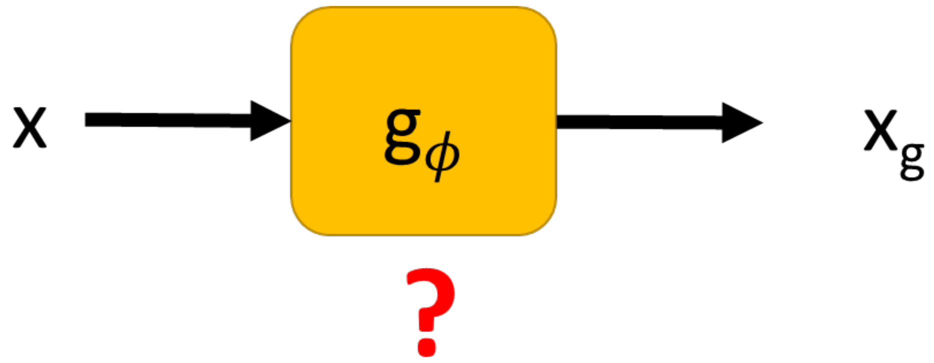
Worse than standard training

ADA: Volpi et al. Neurips 2018; M-ADA: Qiao et al. CVPR 2019; AugMix: Hendrycks et al. ICLR 2019; RandConv: Xu et al. ICLR 2020

Rethink Data-Augmentation

Go Beyond STATIC, PRE-DEFINED Augmentations

Our Solution: *Discover* Data Transformations *During Training*



How?

Classifier's failures are informative – leverage them for guiding data augmentation

Tune parameters of a image-to-image network g() to transform images

Key Finding

Data transformations discovered during training are more effective

Instead of using pre-defined static augmentations

"ALT"
Generalizing to Domain Shift *via* Adversarially Learned Transformations

Tejas Gokhale, Rushil Anirudh, Jay Thiagarajan, Bhavya Kailkhura, Chitta Baral, Yezhou Yang

Arizona State University

Lawrence Livermore National Laboratory

WACV 2023

ALT: Adversarially Learned Transformations

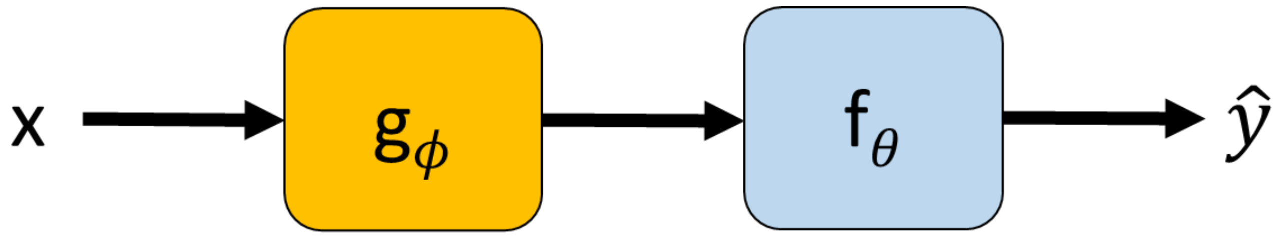
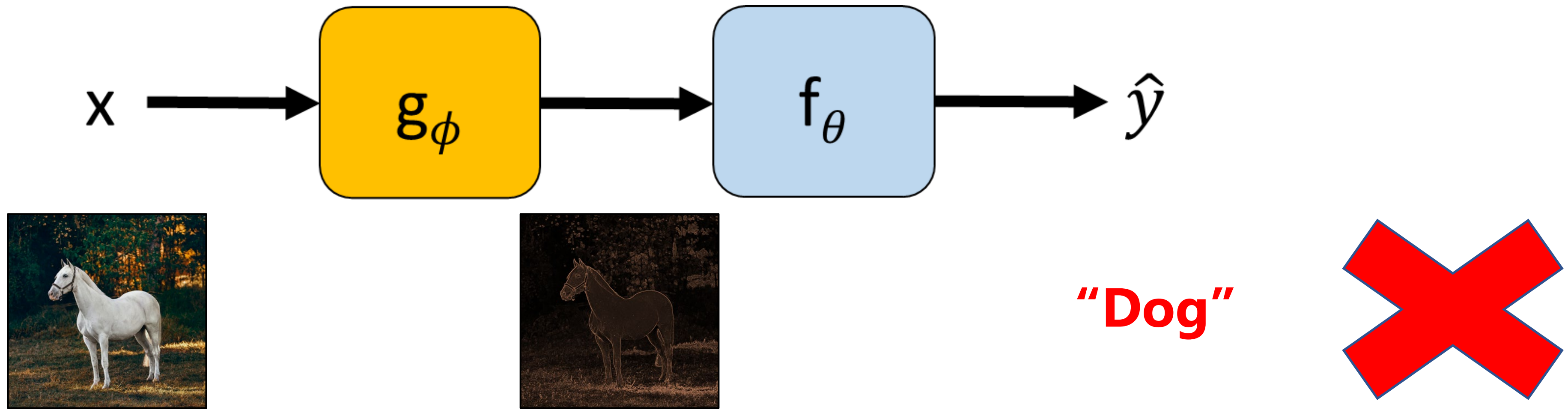


Image Transformation Network
$g()$
With Parameters ϕ

Classifier
$f()$
With Parameters θ

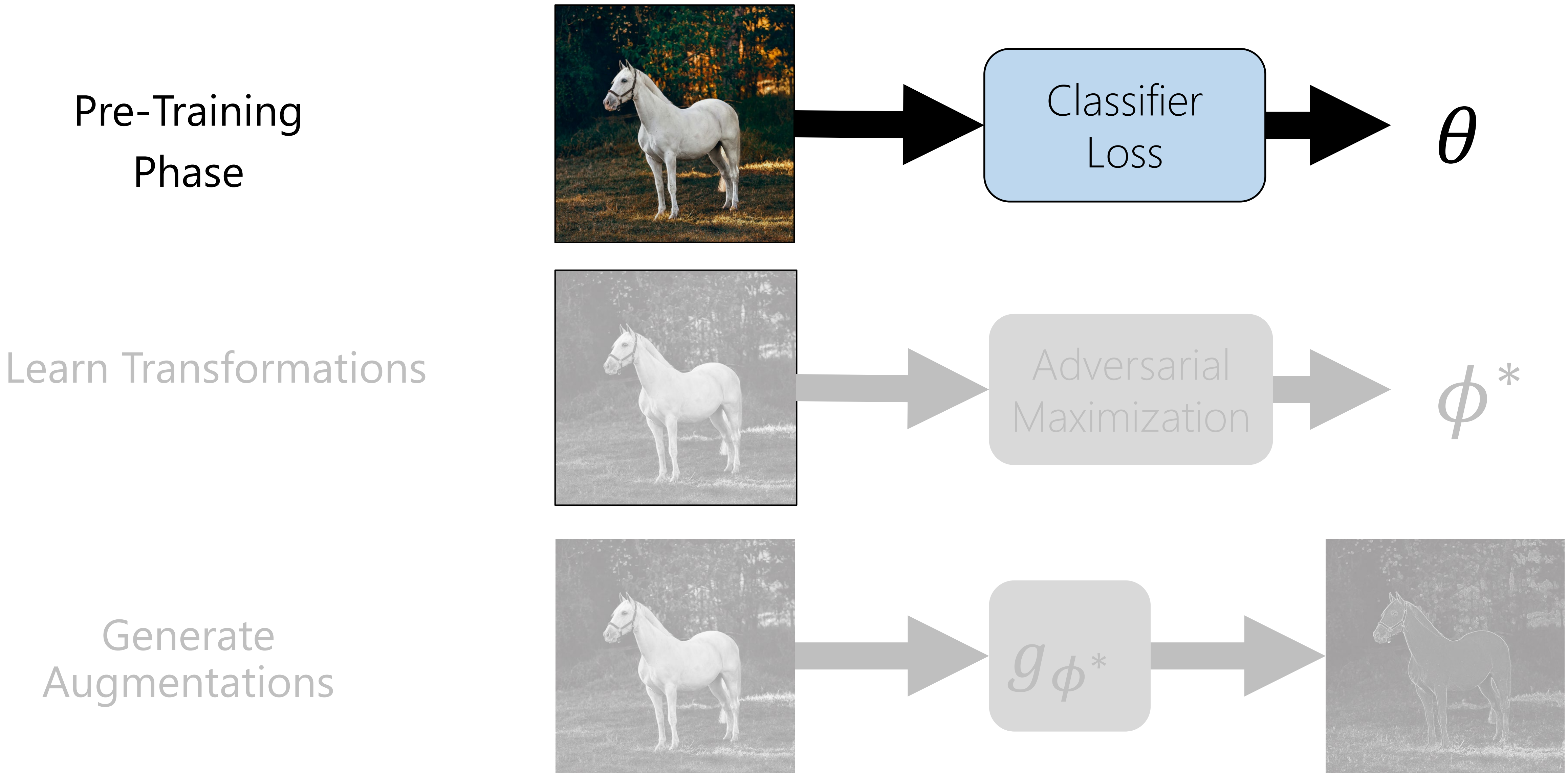
ALT: Adversarially Learned Transformations



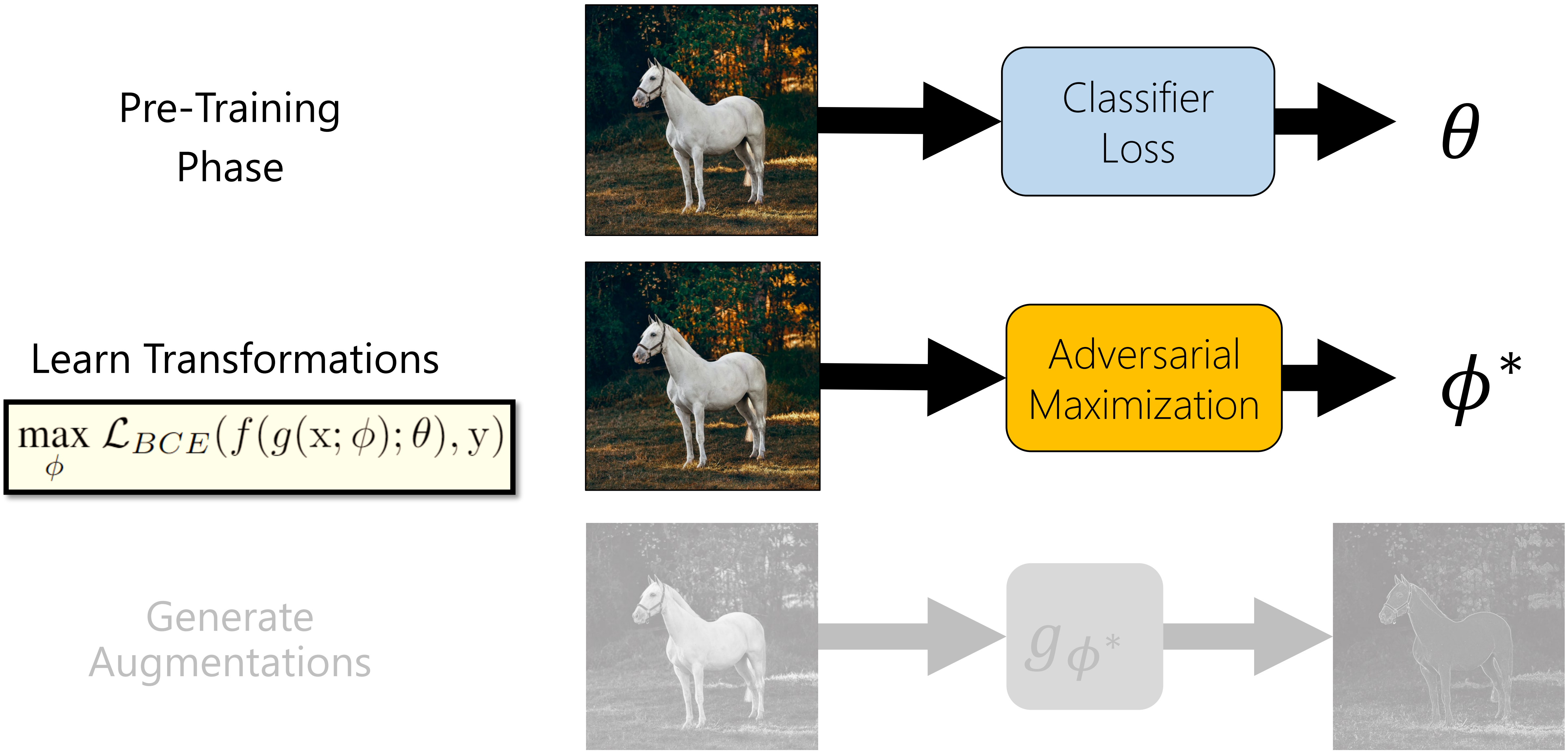
For each batch learn perturbations of ϕ to maximize classifier loss

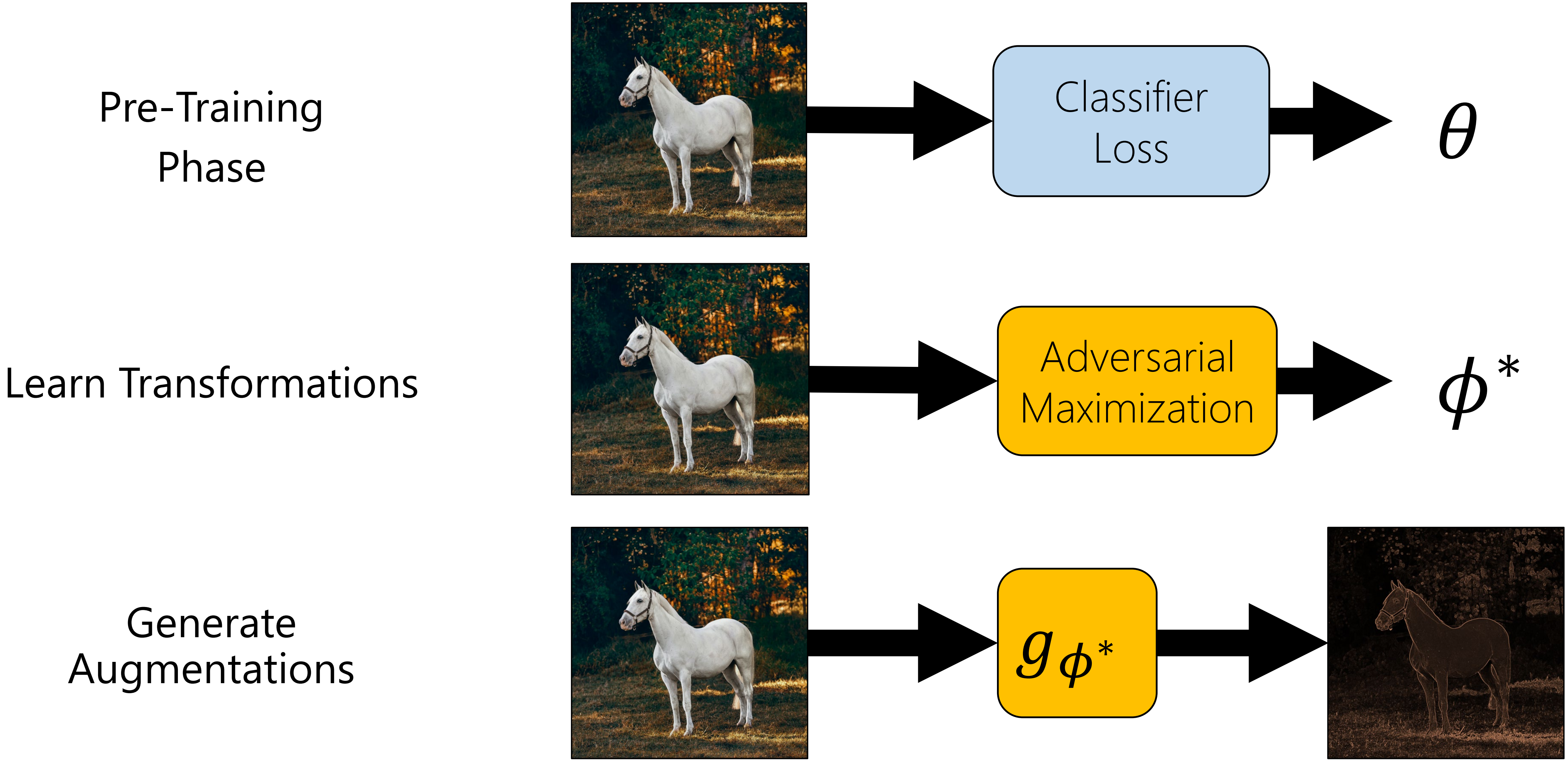
$$\boxed{\max_{\phi} \mathcal{L}_{BCE}(f(g(\mathrm{x};\phi);\theta),\mathrm{y})}$$

$$\phi \leftarrow \phi + \nabla(L_{cls}(f(g(x;\phi)),y) - L_{TV}(x_g)$$



Pre-Training Phase
Classifier Loss
θ
Learn Transformations
Adversarial Maximization
ϕ*
Generate Augmentations
g_ϕ*


Pre-Training
Phase
Classifier
Loss
θ
Learn Transformations
$\max_{\phi} \mathcal{L}_{BCE}(f(g(\mathrm{x};\phi);\theta),\mathrm{y})$
Adversarial
Maximization
ϕ^*
Generate
Augmentations
g_{ϕ^*}



These transformed images are used for training

Enforcing Consistency on Classifier's Predictions

$p \quad = f($  $)$

$p_g = f($  $)$

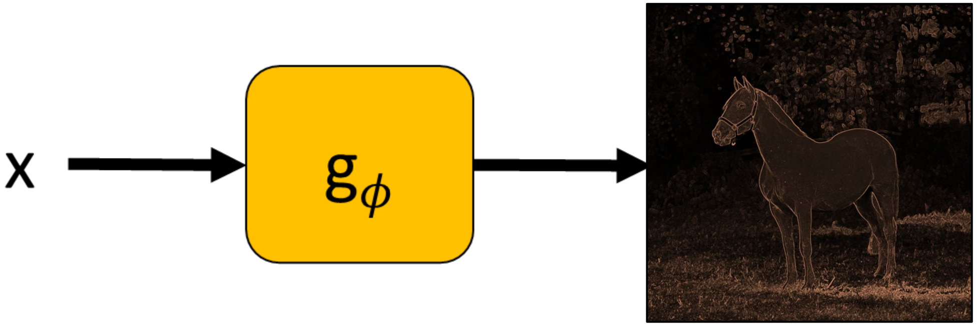
$$p_{mix} = \frac{p + p_g}{2}$$

$$L_{consistency} = D_{KL}(p_{mix} | p) + D_{KL}(p_{mix} | p_g)$$

Improving Diversity with ALT

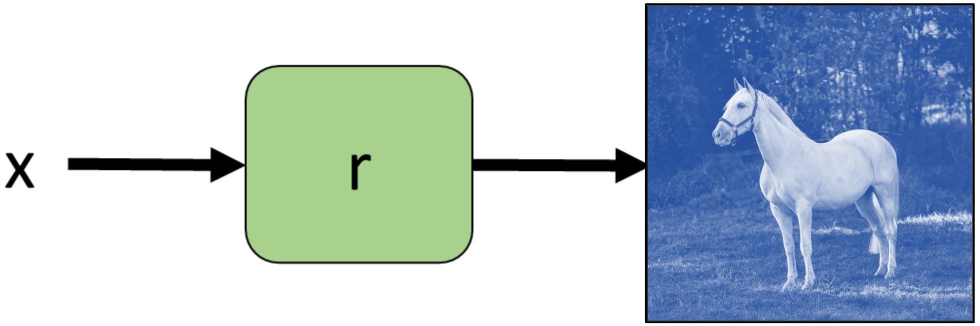
ALT

via adversarial perturbations of ϕ



STATIC AUGMENTATION

via static data augmentations
e.g. AugMix, RandConv



Use ALT in conjunction with static data augmentations from previous work

This further boosts performance compared to using $g()$ only

Enforcing Consistency on Classifier's Predictions

$p = f($  $)$

$p_g = f($  $)$

$p_r = f($  $)$

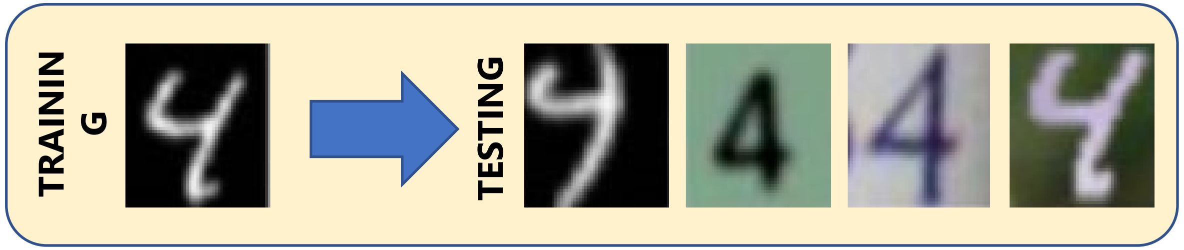
$$p_{mix} = \frac{p_c + p_g + p_r}{3}$$

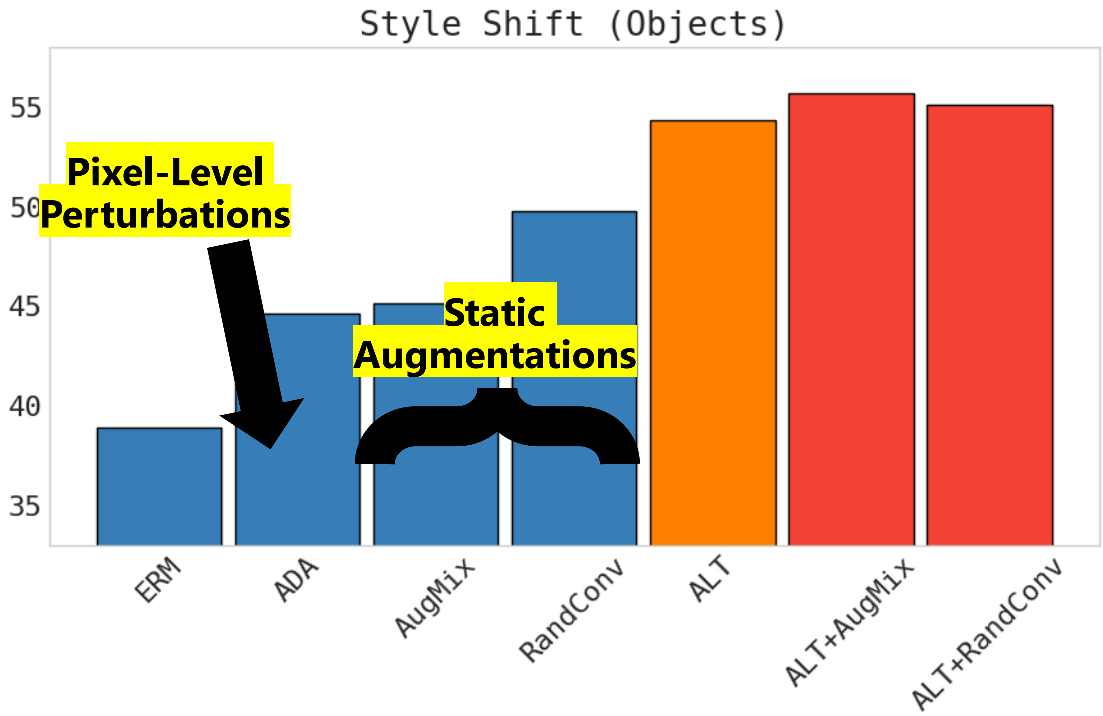
$$L_{consistency} = D_{KL}(p_{mix}|p) + D_{KL}(p_{mix}|p_g) + D_{KL}(p_{mix}|p_r)$$

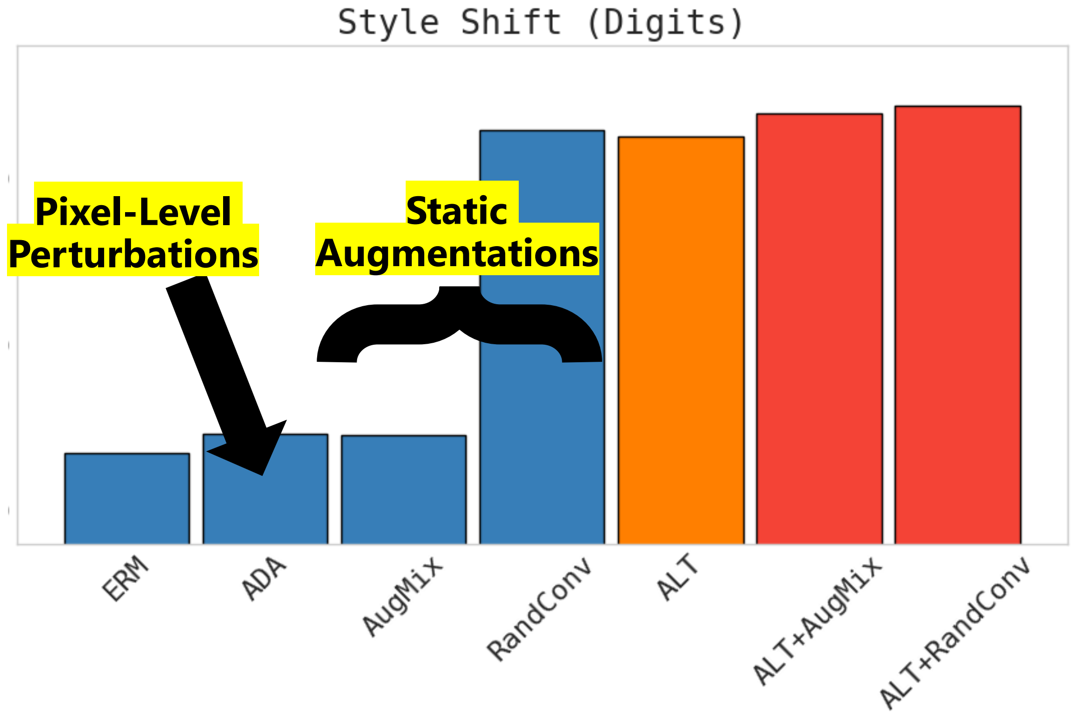
Results: (Style Shift)

Object Classification

Digit Classification





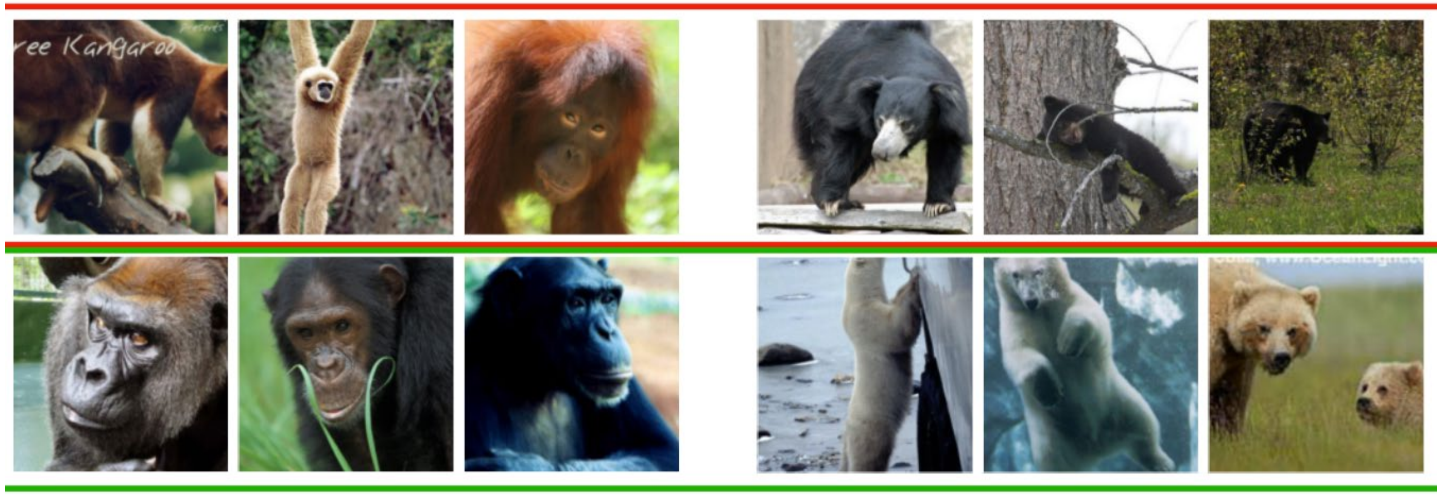


Li et al. "PACS Dataset" ICCV 2017

(Subpopulation Shift) Animal Classification

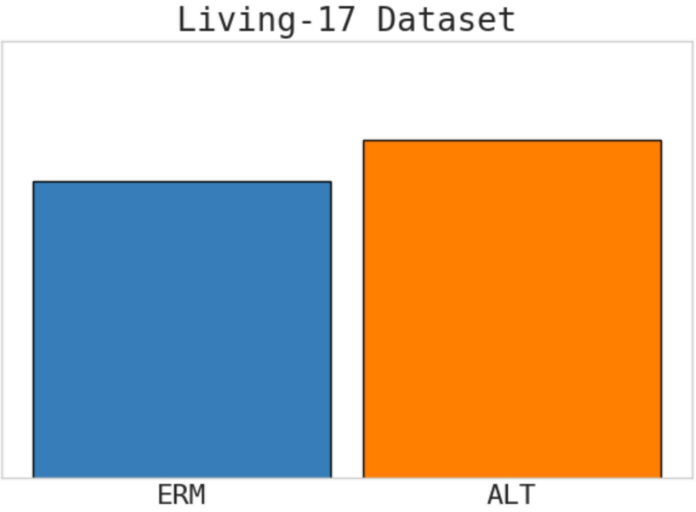
TRAININ
G

Test



Ape

Bear



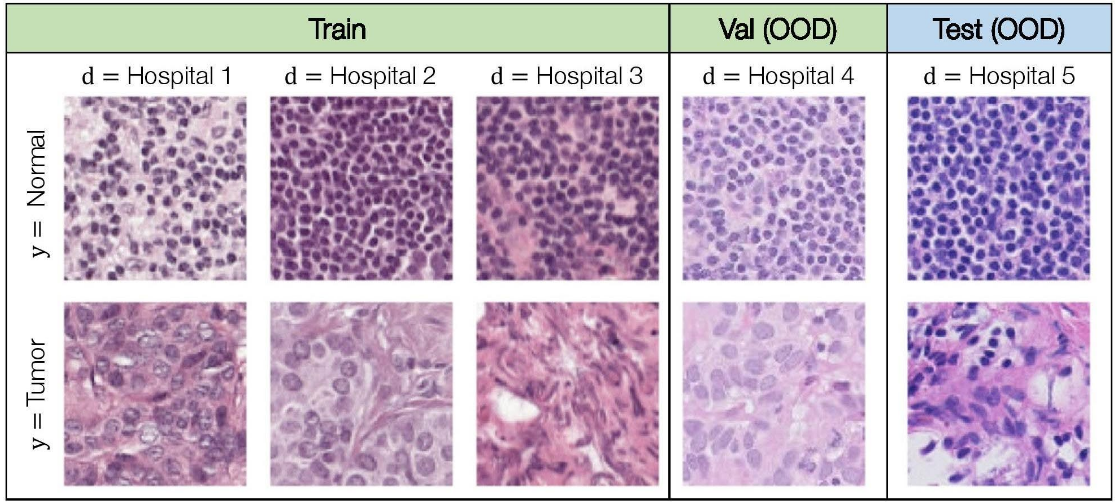
- Trained on one set of sub-species *(apes: gibbon/orangutan)*
- Tested on a different set of sub-species *(apes: gorilla/chimpanzee)*

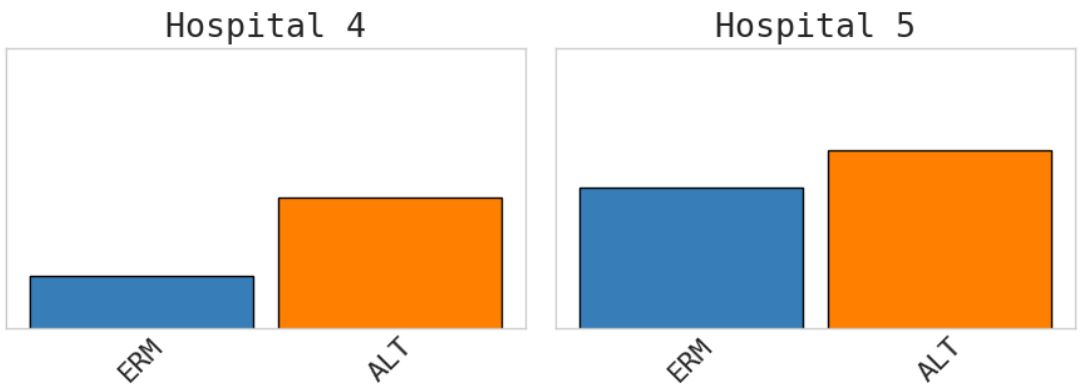
ALT improves robustness to Subpopulation Shift

Santurkar et al. "BREEDS Benchmark" ICLR 2021

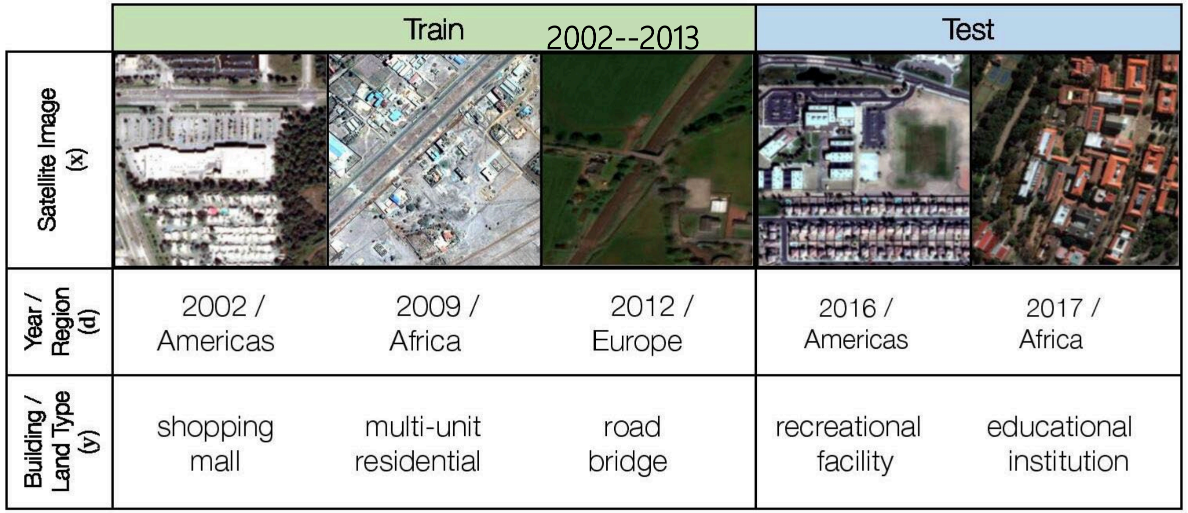
Results: Application to Societal Challenges

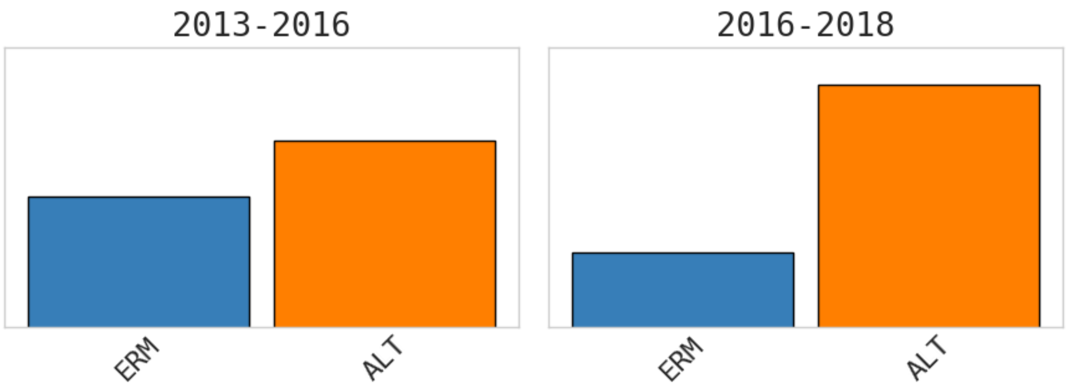
(Hospital Shift)
Tumor Classification





(Terrain Shift)
Land-Use Classification





Koh et al. "WILDS Benchmark", ICML 2021

What if knowledge about unseen domains is available?

How can we leverage that knowledge

to discover image transformations?

"AGAT"
Generalizing to Domain Shift *via* Attribute Guided Adversarial Training

Tejas Gokhale, Rushil Anirudh, Bhavya Kailkhura, Jay Thiagarajan, Chitta Baral, Yezhou Yang

Arizona State University

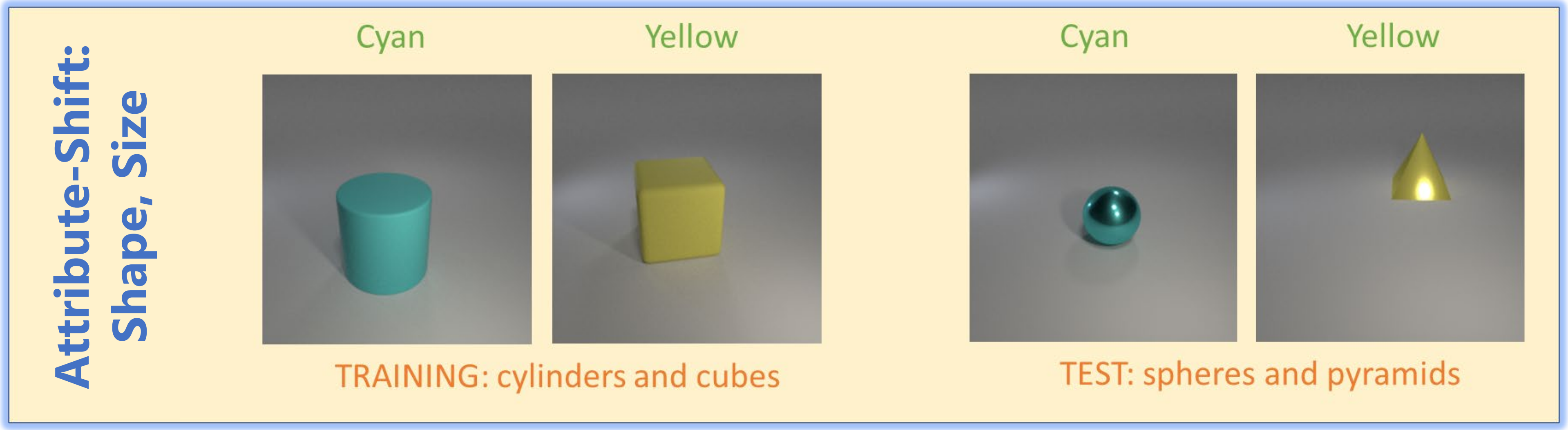
Lawrence Livermore National Laboratory

AAAI 2021

Using Attribute Knowledge for Domain Generalization

In real-world scenarios, test examples can vary along attributes

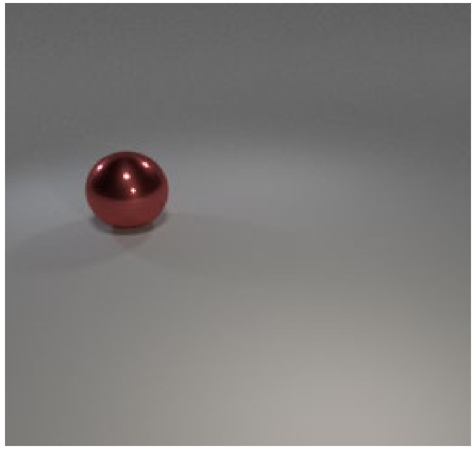
Size, Shape, Materials, Geometric Parameters, Lighting, …



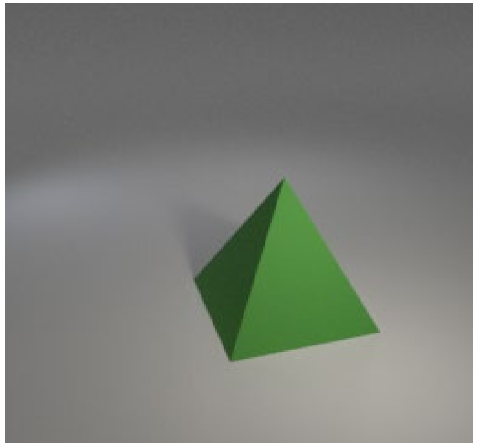
In ALT, we assumed no access to such attributes

How can we leverage attributes to learn useful image transformations?

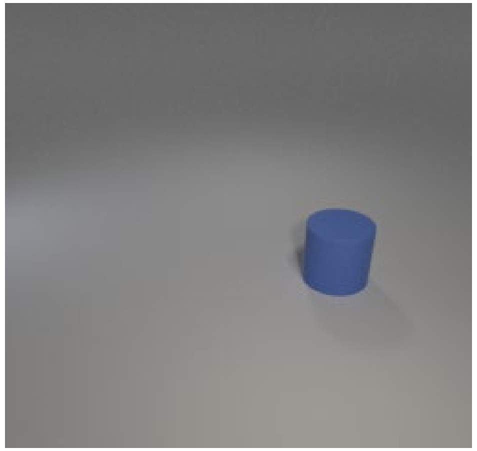
CLEVR-Singles: A Dataset for Studying Attribute-Level Domain Shift



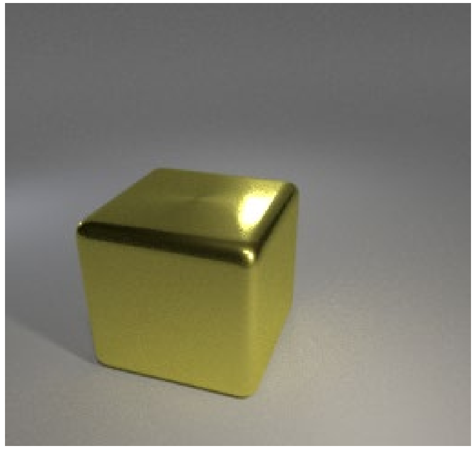
RED
{medium, sphere, metal, northwest}



GREEN
{large, pyramid, rubber, southeast}



BLUE
{small, cylinder, rubber, northeast}



YELLOW
{large, cube, metal, southwest}

- Photorealistic rendering of single objects. Controlled setting for studying attribute-level domain shift
- (Classification) task attribute: **Color**; Task-invariant Attributes: **Size, Shape, Material, Position**

Size	Small, Medium, Large	3
Shape	Sphere, Cylinder, Cube, Pyramid	4
Material	Rubber, Metal	2
Position	NW, SW, NE, SE	4
Color	Red, Blue, Green, Yellow, Cyan, Purple, Grey, Brown	8

CLEVR-Singles Open-Source Data & Code: https://github.com/tejas-gokhale/CLEVR-Singles

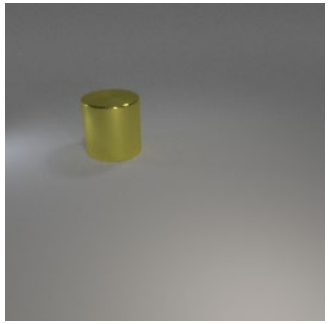
CLEVR-Singles: A Dataset for Studying Attribute-Level Domain Shift

Create train—test dataset splits s.t. attribute combinations at test time are not seen during training

e.g. unseen **Material + Position** combinations

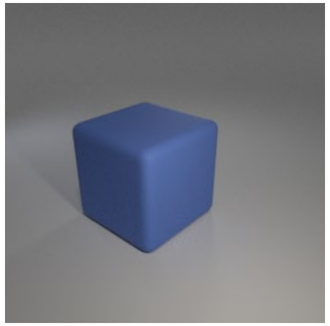
TRAINING

Metal objects far from the camera
Rubber objects close to the camera



Metal, far

Metal, far

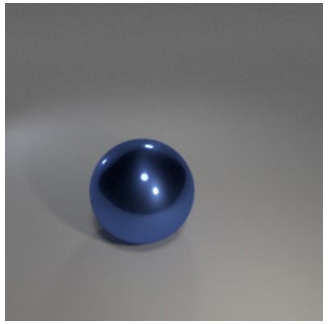


Rubber, close

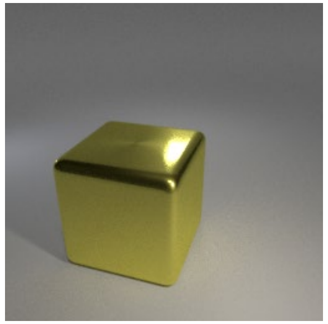
Rubber, close

TESTING

Metal objects close to the camera
Rubber objects far from the camera

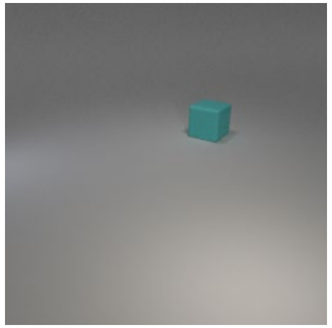


Metal, close



Metal, close

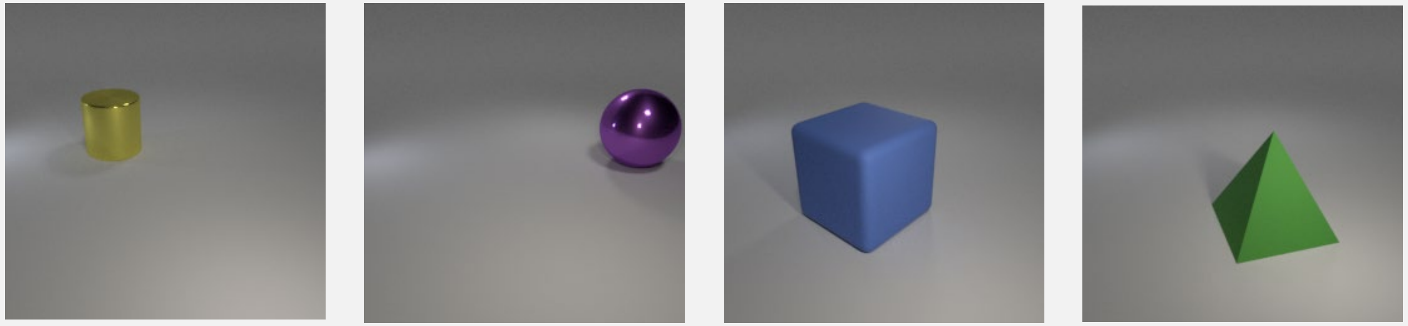
Rubber, far



Rubber, far

Problem Setting

TRAINING DATASET



Attribute Set e.g. ["Size", "Shape", "Material", "Position"]

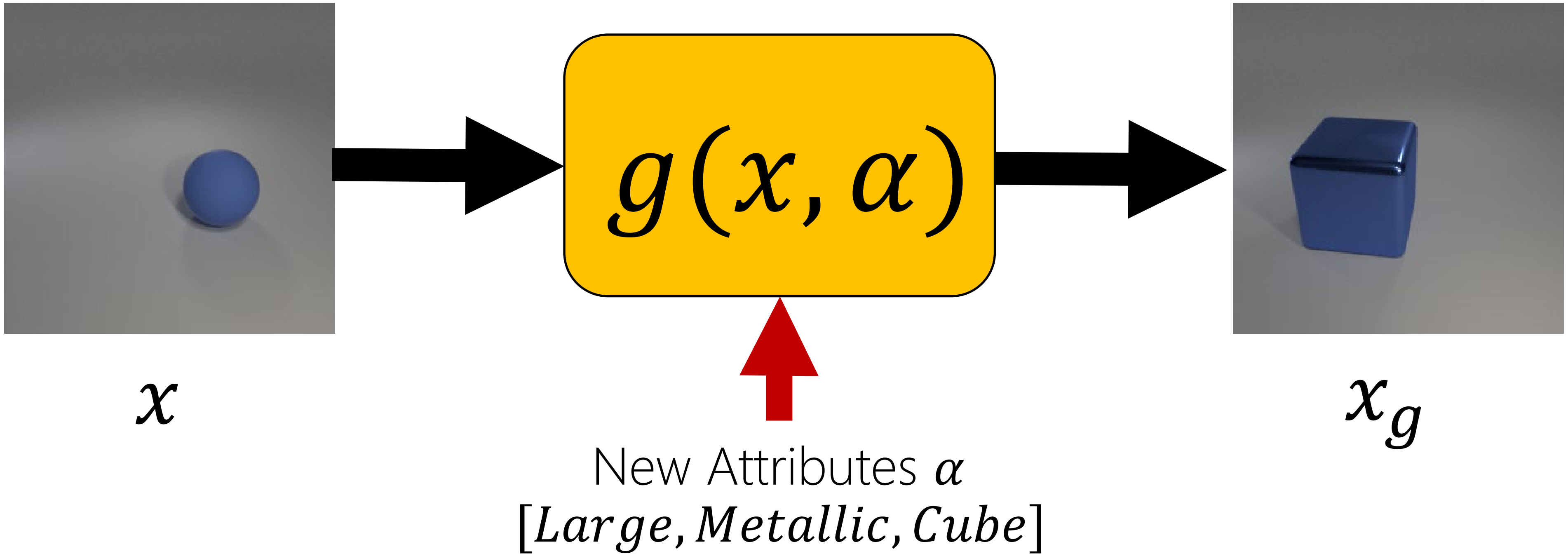
- **Unknown:**
  - which attributes will change at test time
  - by what magnitude
  - in what combination
- **No Access To:**
  - Validation set
  - exemplars representing attribute shift

Goal

Train a classifier that can generalize to attribute-level domain shift

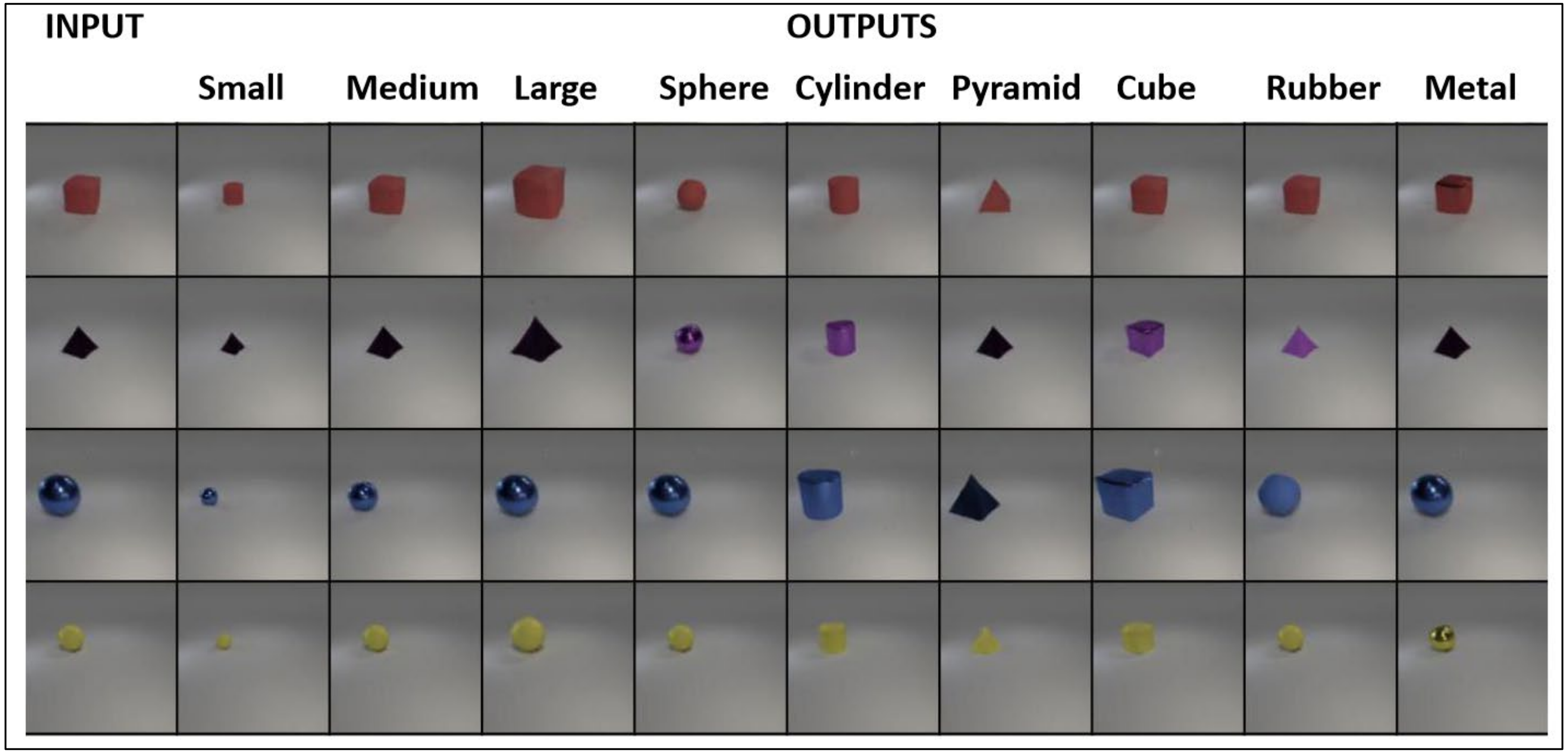
Attribute-Guided Adversarial Training

- Parameterize input space by attributes α
  - Train a Generative Model conditioned on the attributes
- Maximize exploration of input space by learning attribute-level transformations

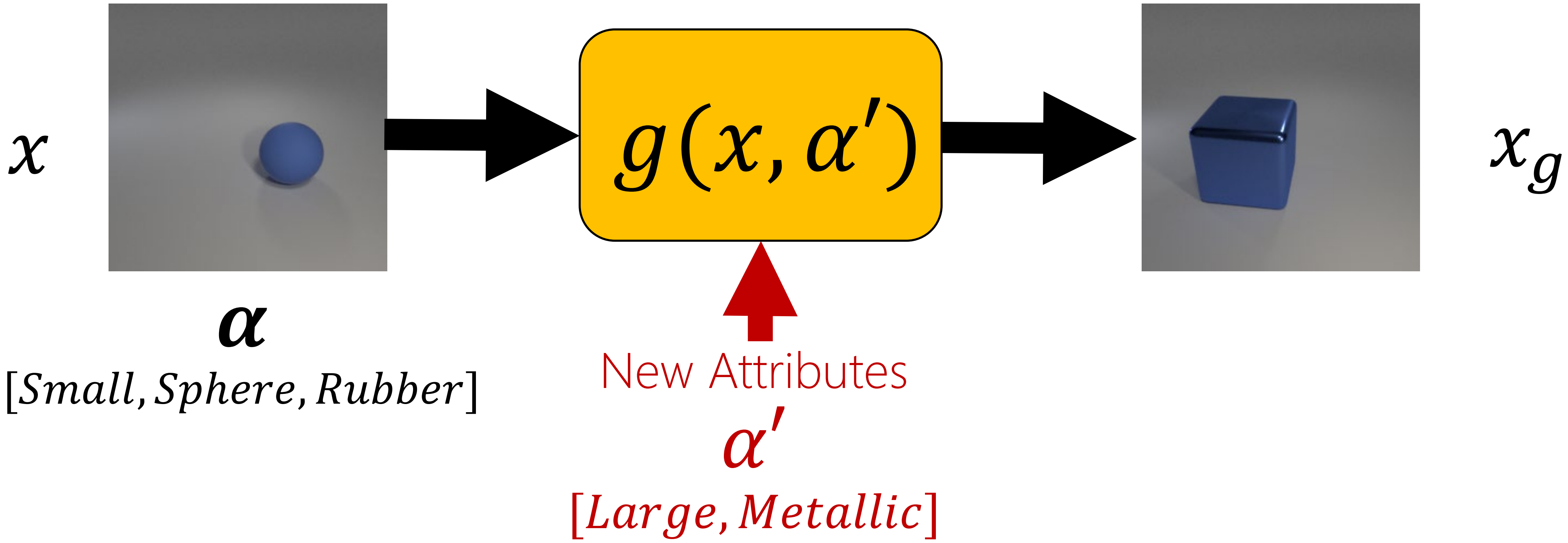


Attribute-Guided Adversarial Training

- Desirable Properties of Generative Function $g()$
  - Generate plausible and diverse perturbations of attributes
  - Reflect a larger coverage of attribute space than training data
  - Generate novel attribute combinations



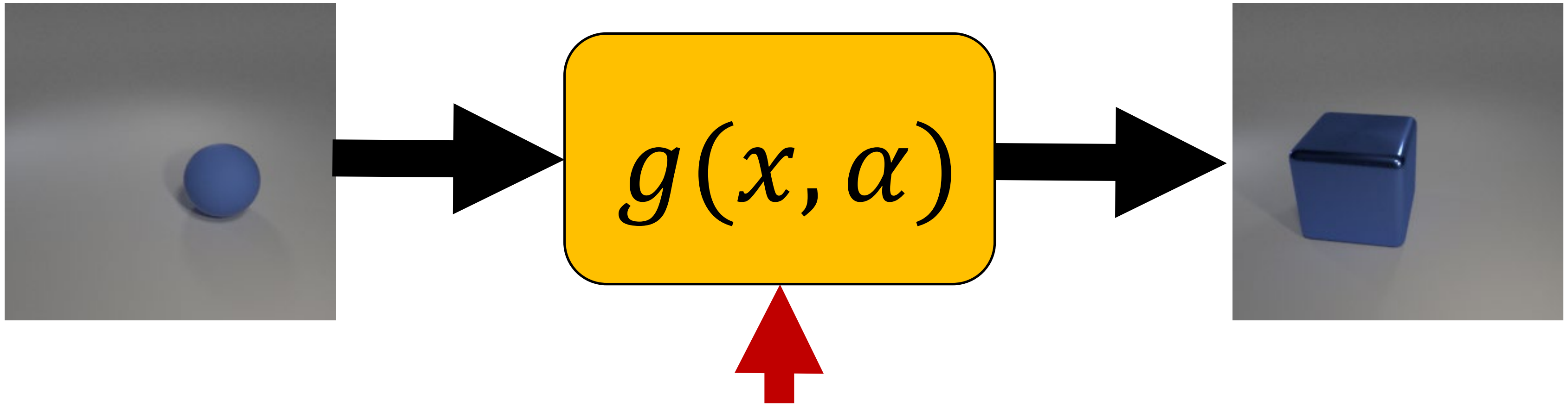
Attribute-Guided Adversarial Training



Discover attribute combinations that are adversarial to the classifier.

$$\max_{\alpha'} \quad \ell\left(f\left(g(x, \alpha')\right), y\right)$$

Attribute-Guided Adversarial Training



New Attributes

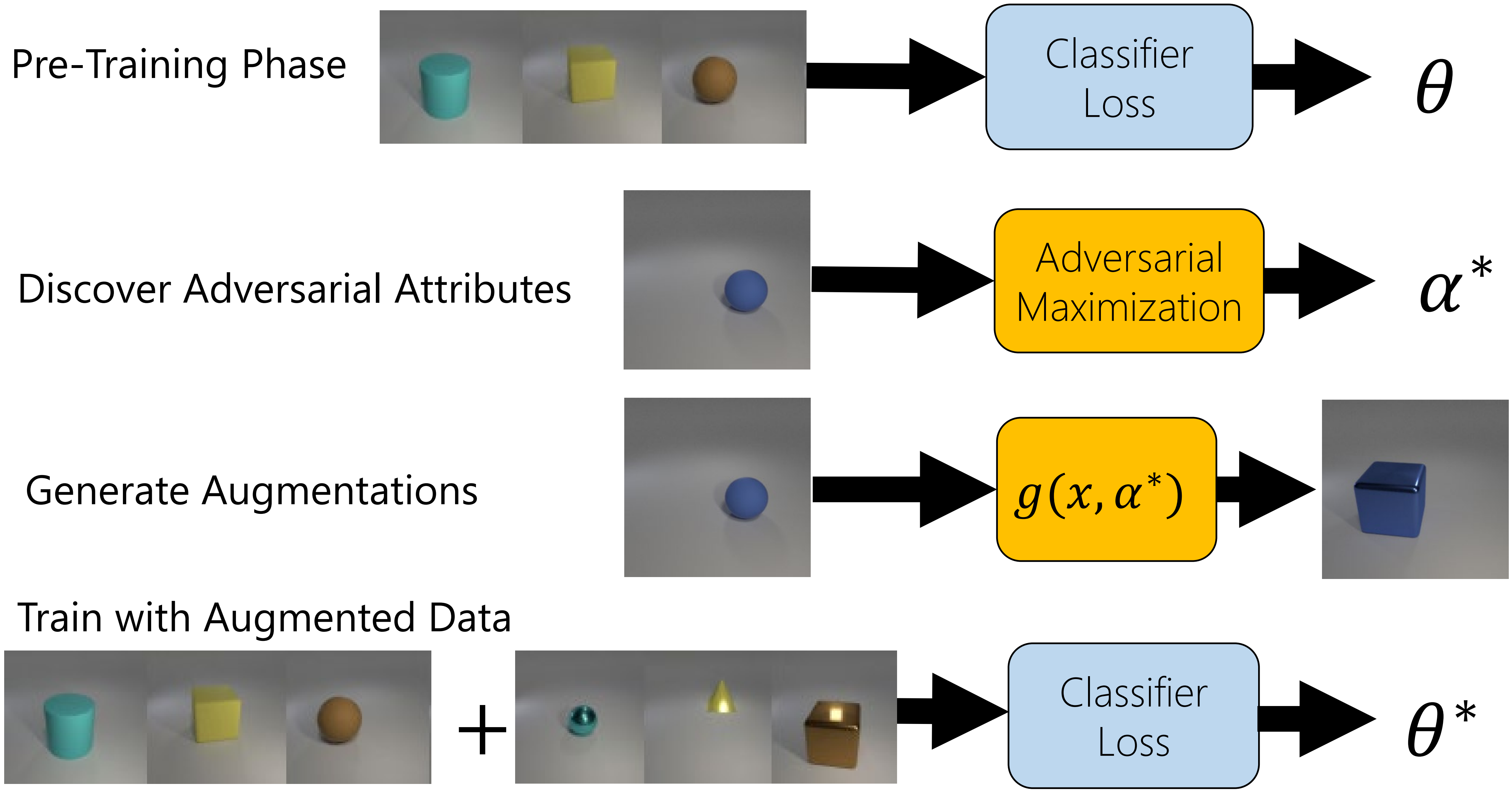
α'

$[Large, Metallic]$

Discover attribute combinations that are adversarial to the classifier.

AND explore new regions in the attribute space

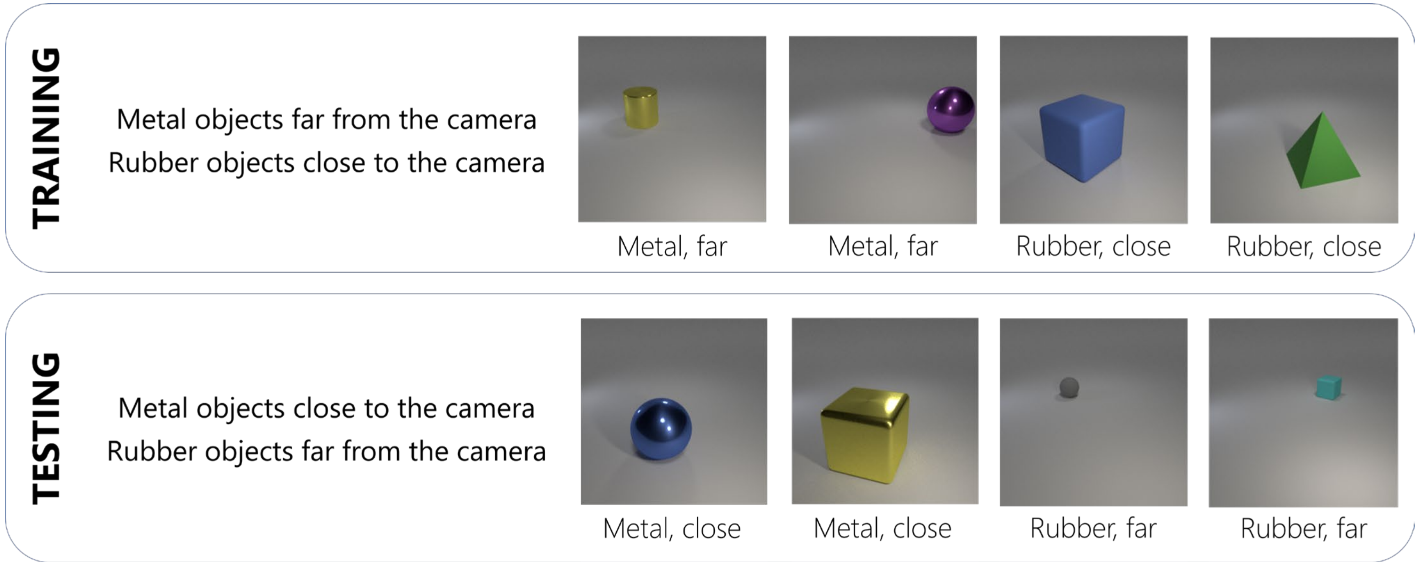
$$\max_{\alpha'} \quad \ell\big(f\big(g(x, \alpha')\big), y\big) + \gamma \big|\big|\alpha - \alpha'\big|\big|_2$$

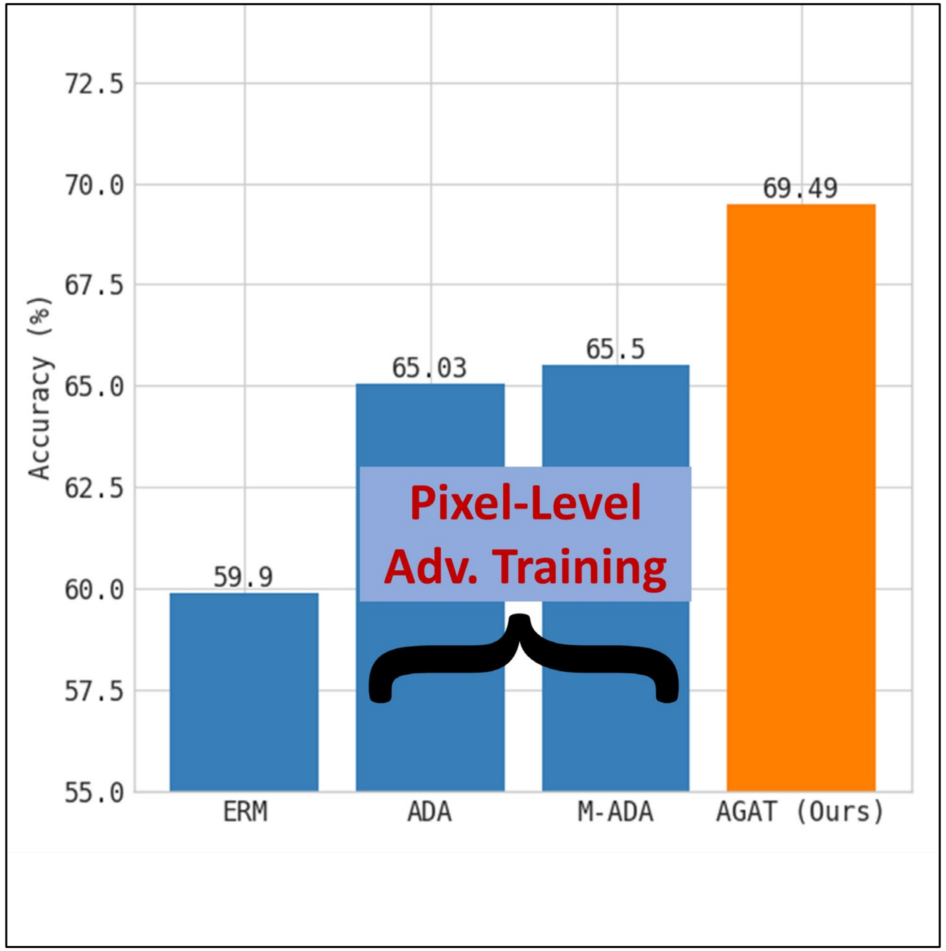


Pre-Training Phase
Classifier Loss
θ
Discover Adversarial Attributes
Adversarial Maximization
α^*
Generate Augmentations
$g(x, \alpha^*)$
Train with Augmented Data
+
Classifier Loss
θ^*

AGAT is effective for Discrete Attribute-Shift

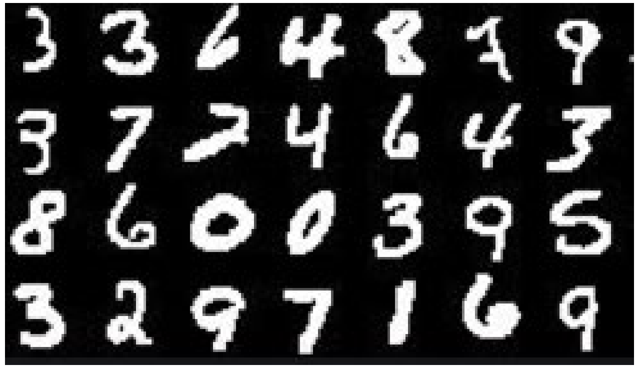
- CLEVR-Singles — ***TASK: color classification***
- Attributes: Size, Shape, Material, Position
- Train—Test split s.t.
  - Limited attribute combinations are observed in during training
  - Performance is evaluated on all combinations



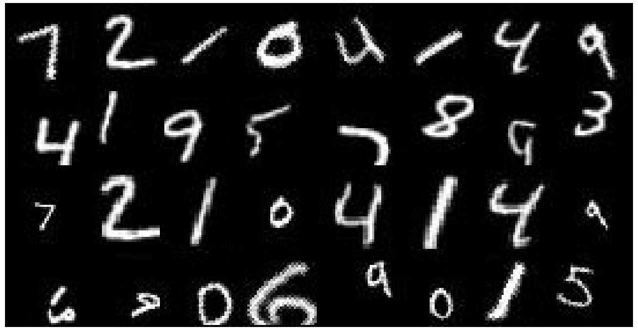


AGAT is effective for Geometric Shift

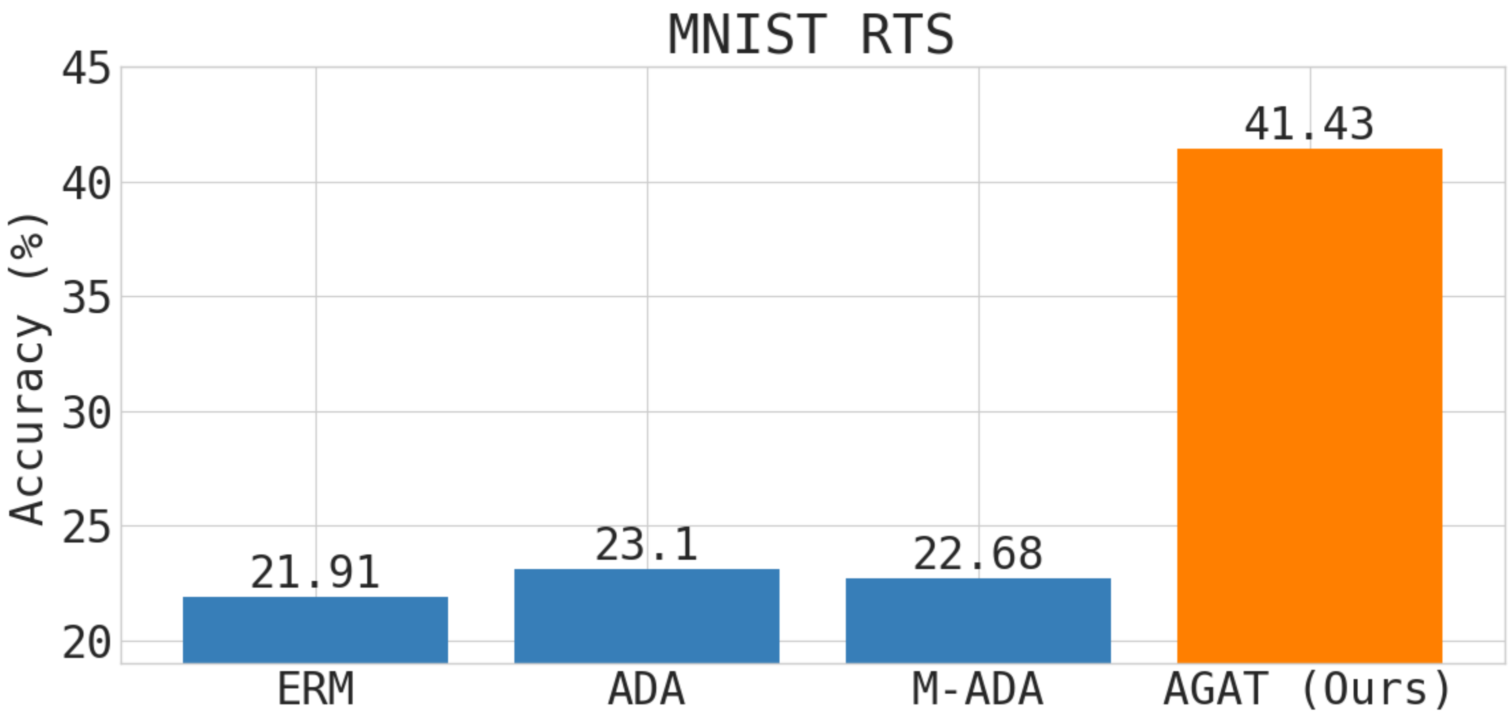
Training



Testing
(Rotation/Translation/Scaling)

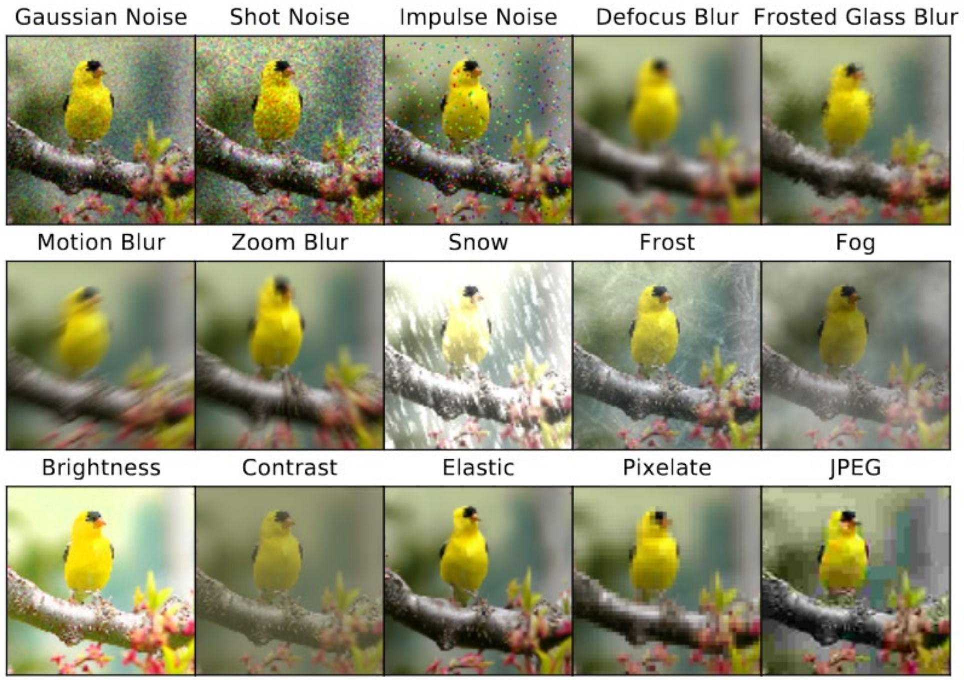


- Training data: MNIST digits
- Test data: ***unknown*** **rotation, translation, scaling**
- AGAT outperforms prior work by a large margin



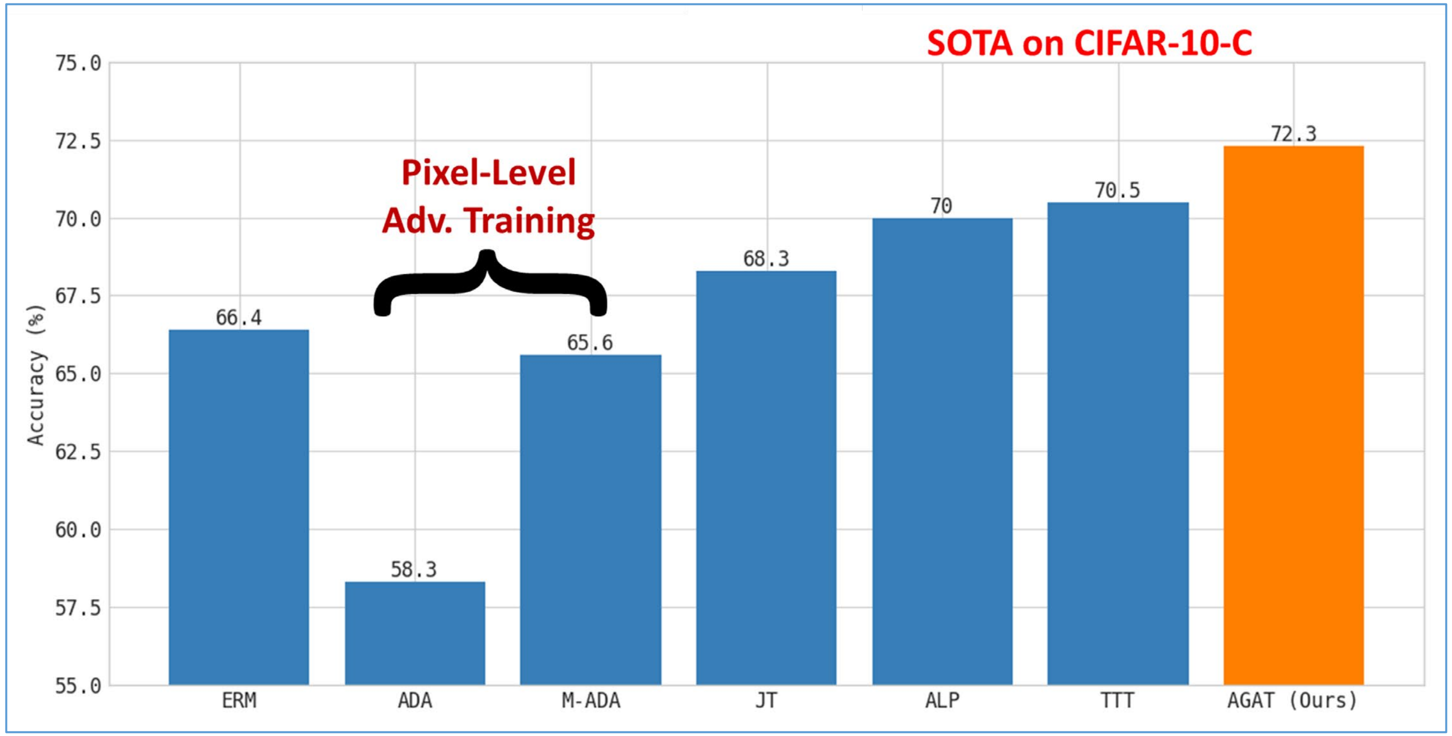
Results (3): AGAT is Effective for Natural Corruptions

Commonly occurring corruptions
(CIFAR-10-C, *Hendrycks et al. ICLR 2019*)



- No access to specific attributes
- We use pseudo-attributes (Gaussian coefficients α_1, α_2)

$$x_g = \frac{1}{\alpha_1\sqrt{2\pi}}\, e^{\frac{x^2}{2\alpha_1^2}} + n\,; \quad \text{where } n \sim \mathcal{N}(0, \alpha_2)$$



Adversarially Discovering Image Transformations

Without domain knowledge: **ALT**

With attribute-level knowledge: **AGAT**

Strong results for many types of distribution shift

Style (Objects)	Style (Digits)	Sub-population	Medical Images	Land-Use / Terrain	Discrete Attributes	Geometric Transform	Common Corruption
+15%	+19%	+2%	+3%	+9%	+10%	+19%	+6%