Domain Adaptation

CS 330

Course Reminders

Poster session next **Wednesday**.

Project report due the following **Monday**

Azure: Form on Ed for requesting more credits for project.

Plan for Today

Domain Adaptation
- Problem statements
- Algorithms
  - Data reweighting
  - Feature alignment

Domain Adaptation -> Domain Generalization

Goals for this lecture:

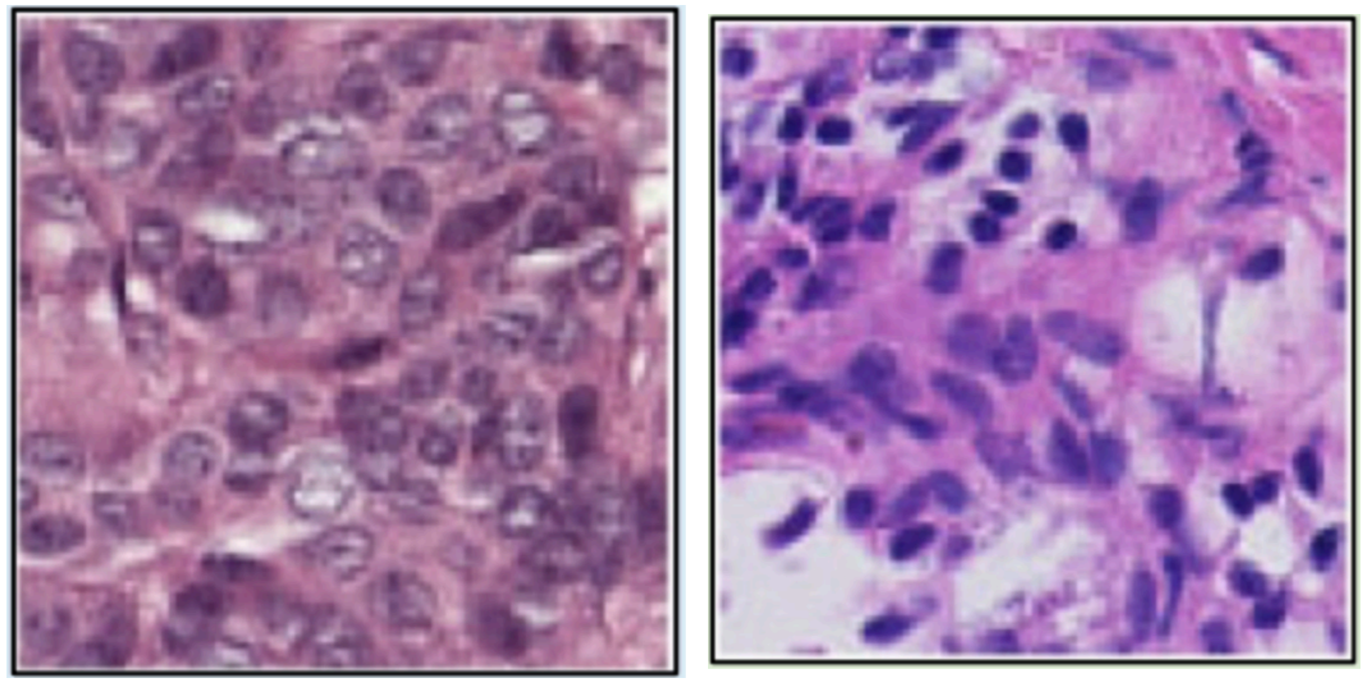
- Understand domain adaptation & generalization problems, how they relate to multi-task learning and transfer learning

- Understand two general approaches and when to use one vs. another

Example domain adaptation problems

Tumor detection & classification

Source hospital Target hospital



varying imaging techniques, different demographics

Land use classification

Source region Target region

appearance of buildings, plants; weather conditions, pollution

Text classification, generation

Source corpus Target corpus

Simple English Wikipedia

arXiv

PubMed

differing sentence structure, vocabulary, word use

Problem Settings Recap

Multi-Task Learning

Solve multiple tasks $\mathcal{T}_1, \cdots, \mathcal{T}_T$ at once.

$$\min_\theta \sum_{i=1}^{T} \mathcal{L}_i(\theta, \mathcal{D}_i)$$

Transfer Learning

Solve target task $\mathcal{T}_b$ after solving source task(s) $\mathcal{T}_a$

by *transferring* knowledge learned from $\mathcal{T}_a$

Meta-Learning Problem

Transfer Learning with Many Source Tasks

Given data from $\mathcal{T}_1, \ldots, \mathcal{T}_n$, solve new task $\mathcal{T}_{\text{test}}$ more quickly / proficiently / stably

What is domain adaptation?

Perform well on target domain $p_T(x, y)$,

using training data from source domain(s) $p_S(x, y)$

A form of transfer learning, with **access to target domain data during training** ("transductive" learning)

Unsupervised domain adaptation: access to unlabeled target domain data

Semi-supervised domain adaptation: access to unlabeled and labeled target domain data

Supervised domain adaptation: access to labeled target domain data.

We will focus on ***unsupervised domain adaptation***.

What is domain adaptation?

Perform well on target domain $p_T(x, y)$,

using training data from source domain(s) $p_S(x, y)$

A form of transfer learning, with **access to target domain data during training**

("transductive" learning)

Unsupervised domain adaptation: access to unlabeled target domain data

Common assumptions:

- Source and target domain only differ in domain of the function, i.e. $p_S(y \mid x) = p_T(y \mid x)$
- There exists a single hypothesis with low error.

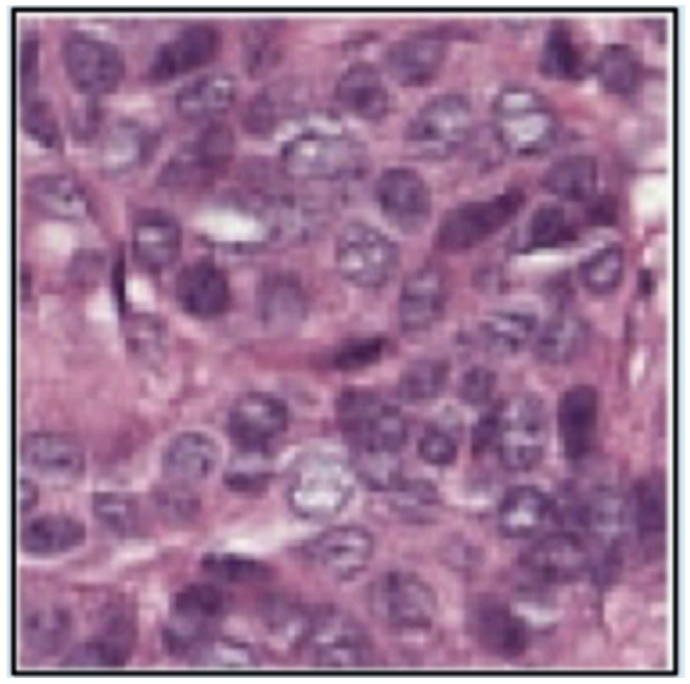
A "domain" is a special case of a "task"

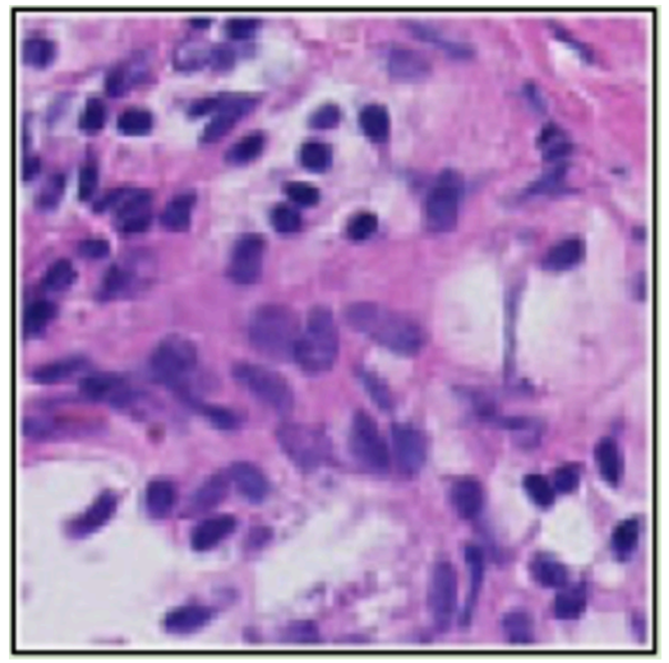
A task: $\mathscr{T}_i \triangleq \{p_i(\mathbf{x}), p_i(\mathbf{y} \mid \mathbf{x}), \mathscr{L}_i\}$ A domain: $d_i \triangleq \{p_i(\mathbf{x}), p(\mathbf{y} \mid \mathbf{x}), \mathscr{L}\}$

Example domain adaptation problems

Tumor detection & classification

Source hospital Target hospital





varying imaging techniques, different demographics

Land use classification

Source region Target region

appearance of buildings, plants; weather conditions, pollution

Text classification, generation

Source corpus Target corpus

Simple English Wikipedia

arXiv

PubMed

differing sentence structure, vocabulary, word use

Revisiting assumptions:

- Access to target domain data during training.
- There exists a single hypothesis $f(y|x)$ with low error.

Question: Should you condition on a task identifier in domain adaptation problems?

Plan for Today

Domain Adaptation
- Problem statements
- Algorithms
  - **Data reweighting**
  - Feature alignment

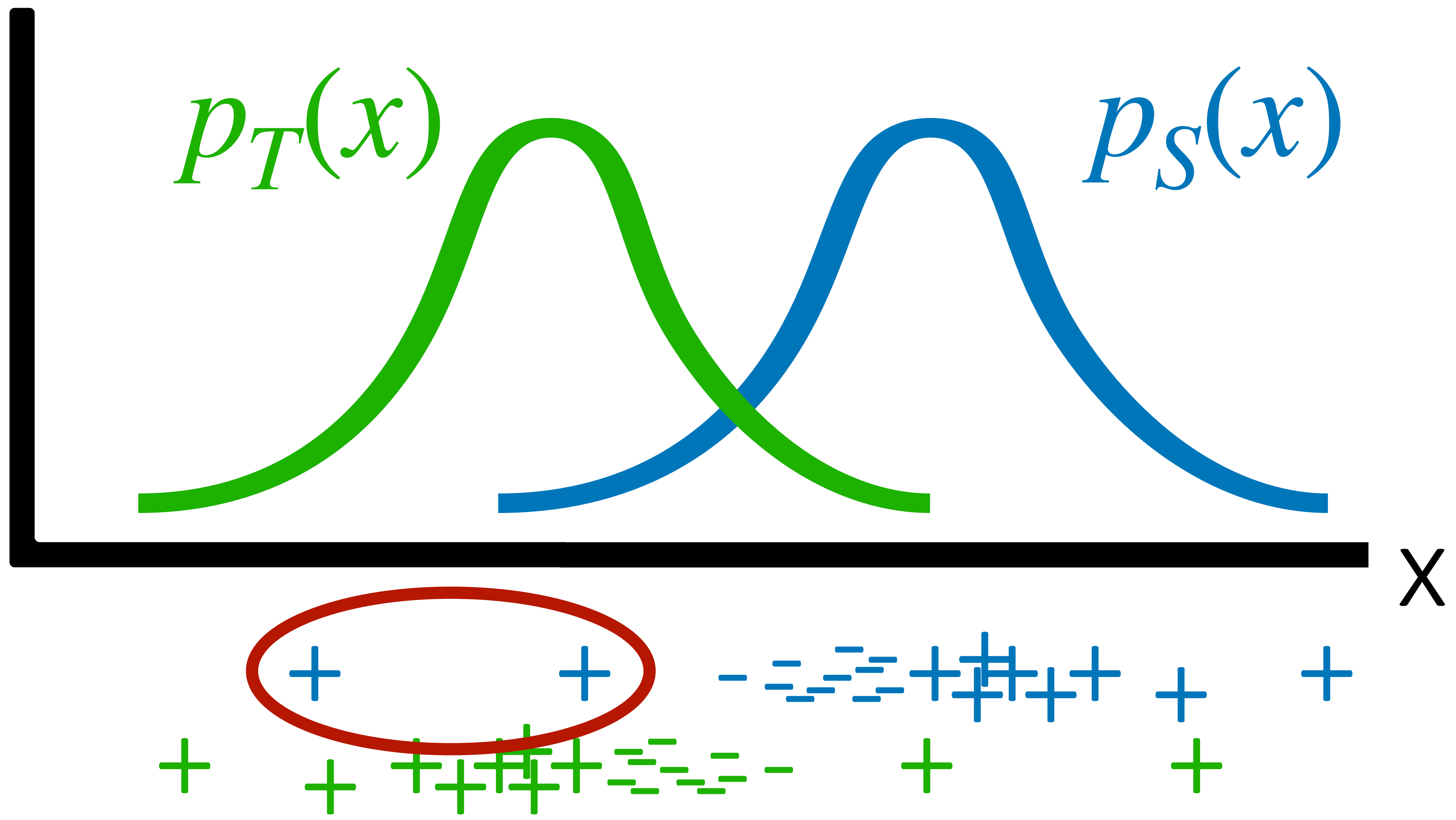
Domain Adaptation -> Domain Generalization

Goals for this lecture:

- Understand domain adaptation & generalization problems, how they relate to multi-task learning and transfer learning

- Understand two general approaches and when to use one vs. another

Toy domain adaptation problem



e.g. sample selection bias

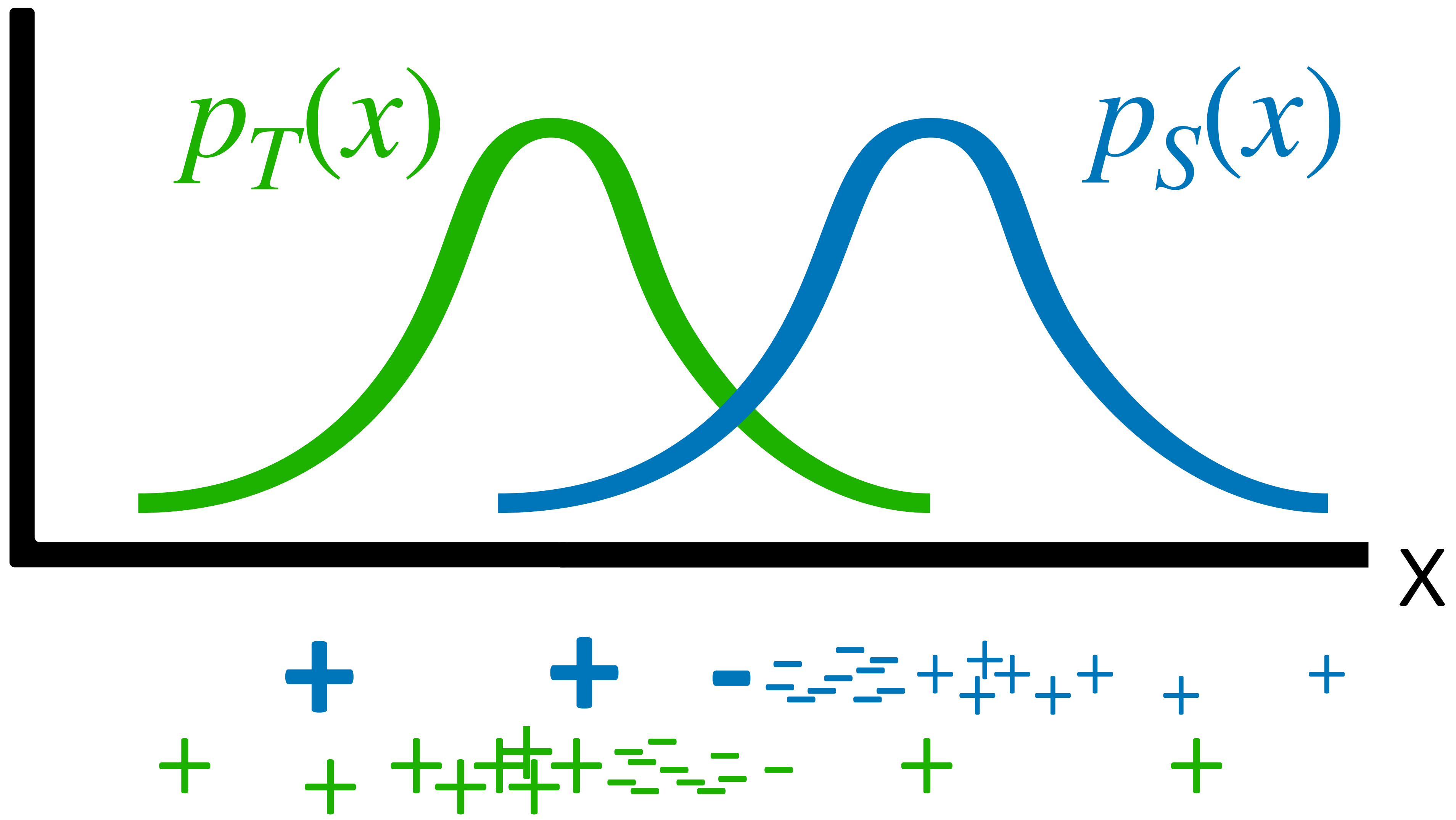
Problem: Classifier trained on $p_S(x)$ pays little attention to examples with high probability under $p_T(s)$

How can we learn a classifier that does well on $p_T(x)$?

(using labeled data from $p_S(x)$ & unlabeled data from $p_T(x)$)

Problem adapted from Blitzer & Daume ICML '10

Toy domain adaptation problem



e.g. sample selection bias

Problem: Classifier trained on $p_S(x)$ pays little attention to examples with high probability under $p_T(s)$

Solution: Upweight examples with high $p_T(x)$ but low $p_S(x)$

Why does this make sense mathematically?

Problem adapted from Blitzer & Daume ICML '10

Domain adaptation via importance sampling

Empirical risk minimization on source data: $\min_\theta \mathbb{E}_{p_S(x,y)}[L(f_\theta(x), y)]$

Goal: ERM on target distribution: $\min_\theta \mathbb{E}_{p_T(x,y)}[L(f_\theta(x), y)]$

$$\mathbb{E}_{p_T(x,y)}[L(f_\theta(x), y)] = \int p_T(x, y) L(f_\theta(x), y) dxdy$$

$$= \int p_T(x, y) \frac{p_S(x, y)}{p_S(x, y)} L(f_\theta(x), y) dxdy$$

$$= \mathbb{E}_{p_S(x,y)} \left[\frac{p_T(x, y)}{p_S(x, y)} L(f_\theta(x), y) \right]$$

Note: $p(y \mid x)$ cancels out if it is the same for source & target

Solution: Upweight examples with high $p_T(x)$ but low $p_S(x)$

Domain adaptation via importance sampling

$$\min_{\theta} \mathbb{E}_{p_S(x,y)} \left[\frac{p_T(x)}{p_S(x)} L(f_\theta(x), y) \right]$$

How to estimate the importance weights $\frac{p_T(x)}{p_S(x)}$?

Option 1: Estimate likelihoods $p_T(x)$ and $p_S(x)$, then divide. But, difficult to estimate accurately.

Can we estimate the ratio ***without*** training a generative model?

Bayes rule:

$$p(x \mid \text{target}) = \frac{p(\text{target} \mid x) p(x)}{p(\text{target})}$$

$$p(x \mid \text{source}) = \frac{p(\text{source} \mid x) p(x)}{p(\text{source})}$$

$$\frac{p_T(x)}{p_S(x)} = \frac{p(x \mid \text{target})}{p(x \mid \text{source})} = \frac{p(\text{target} \mid x) p(\text{source})}{p(\text{source} \mid x) p(\text{target})}$$

a constant

can estimate with binary classifier!

Bickel, Bruckner, Scheffer. Discriminative Learning Under Covariate Shift. JMLR '09

Domain adaptation via importance sampling

$$\min_{\theta} \mathbb{E}_{p_S(x,y)} \left[\frac{p_T(x)}{p_S(x)} L(f_\theta(x), y) \right]$$

$$\frac{p_T(x)}{p_S(x)} = \frac{p(x \mid \text{target})}{p(x \mid \text{source})} = \frac{p(\text{target} \mid x) p(\text{source})}{p(\text{source} \mid x) p(\text{target})}$$

a constant

can estimate with
binary classifier!

Full algorithm:

1. Train binary classifier $c(\text{source} \mid x)$ to discriminate between source and target data.
2. Reweight or resample data $\mathscr{D}_S$ according to $\dfrac{1 - c(\text{source} \mid x)}{c(\text{source} \mid x)}$.
3. Optimize loss $L(f_\theta(x), y)$ on reweighted or resampled data.

Bickel, Bruckner, Scheffer. Discriminative Learning Under Covariate Shift. JMLR '09

What assumption does this make?

$$\min_\theta \mathbb{E}_{p_S(x,y)}\left[\frac{p_T(x)}{p_S(x)} L(f_\theta(x), y)\right]$$

Source $p_S(x)$ needs to cover the target $p_T(x)$.

Formally: if $p_T(x) \neq 0$, then $p_S(x) \neq 0$.

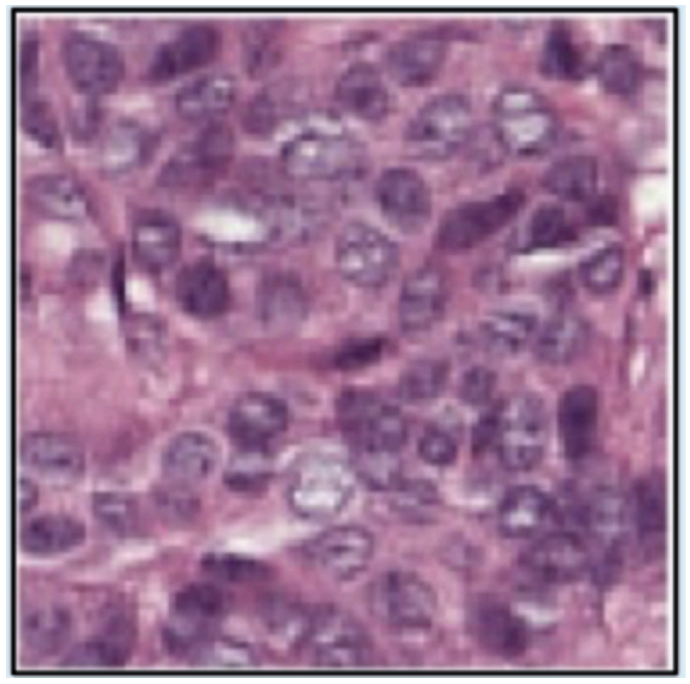
Text classification, generation

Source corpus Target corpus

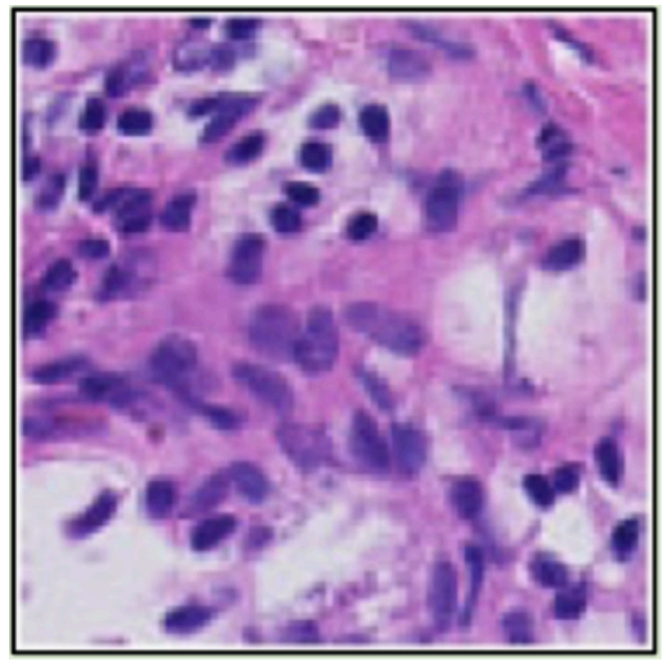
—> May have enough coverage of distr.

Tumor detection & classification

Source hospital



Target hospital



—> Source probably won't cover target distr!

Plan for Today

Domain Adaptation
- Problem statements
- Algorithms
  - Data reweighting
  - **Feature alignment**

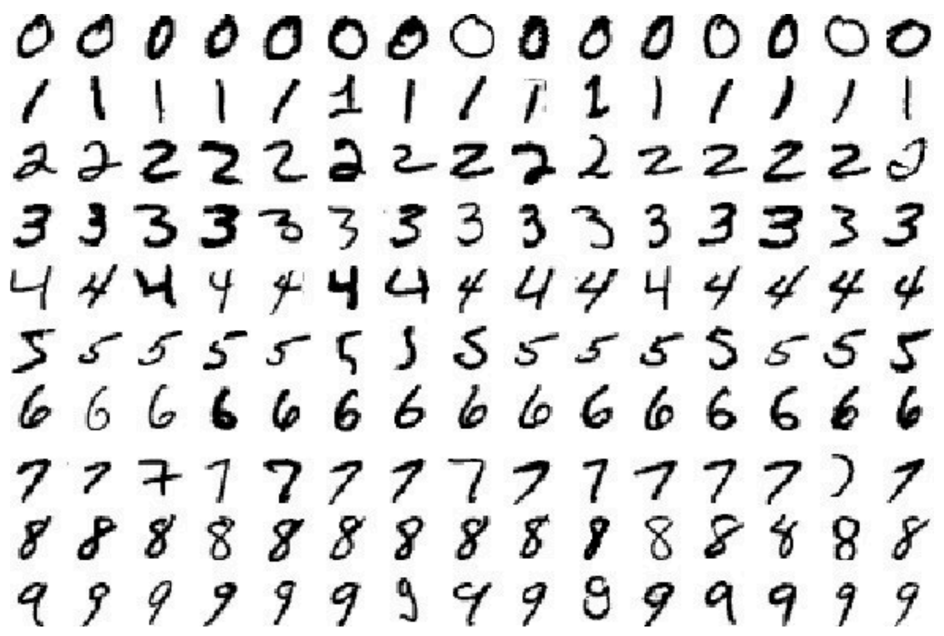
Domain Adaptation -> Domain Generalization

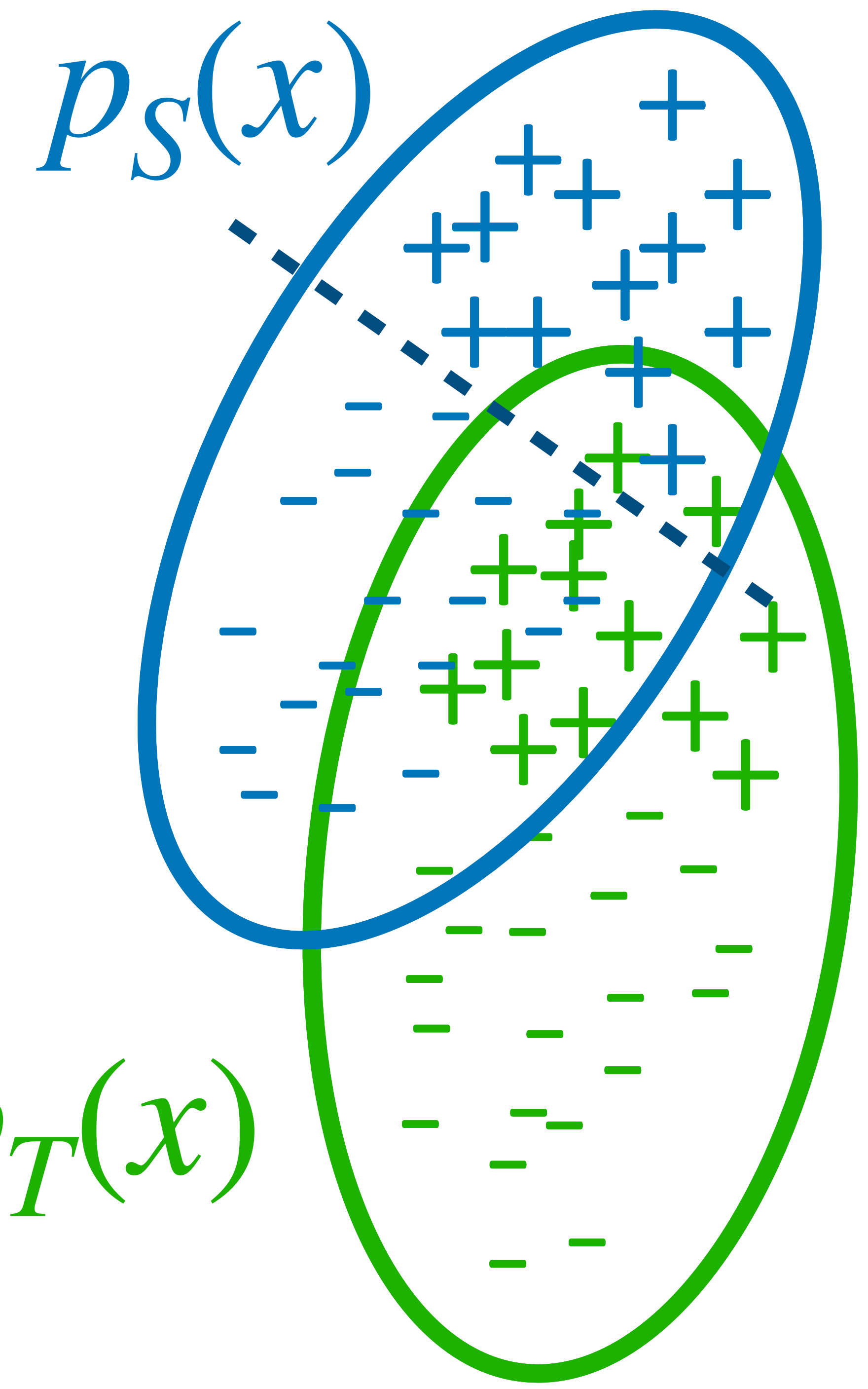
Goals for this lecture:

- Understand domain adaptation & generalization problems, how they relate to multi-task learning and transfer learning

- Understand two general approaches and when to use one vs. another

Domain adaptation if support is not shared?



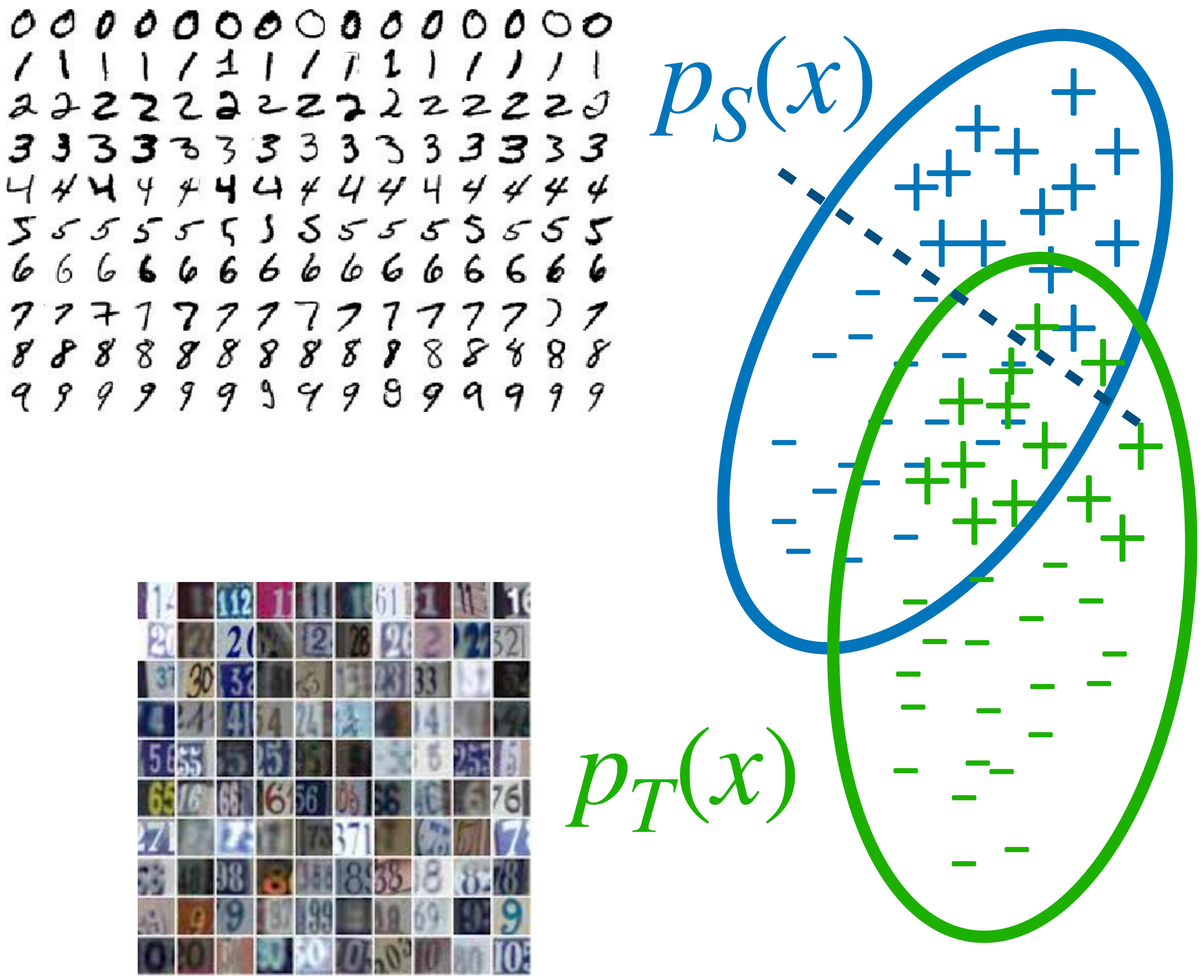


Can we align the features?

Source classifier in ***aligned feature space*** is more accurate in target domain.

How to align the features?

Domain adaptation if support is not shared?



How to align the features?

Need to match features at *population-level*.

i.e. make encoded samples $f(x), x \sim p_S(\cdot)$

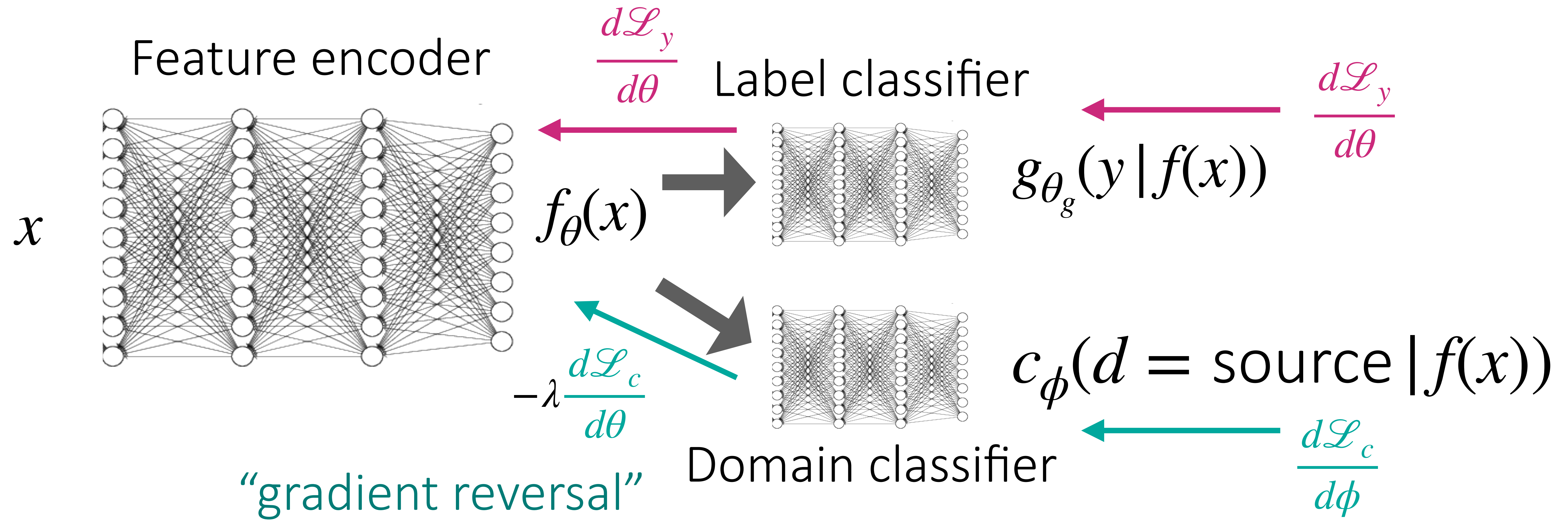
indistinguishable from $f(x), x \sim p_T(\cdot)$

Key idea: Try to fool a domain classifier $c(d = \text{source} \,|\, f(x))$.

If samples are indistinguishable to discriminator, then distributions are the same.

Domain adaptation via feature alignment

Key idea: Try to fool a domain classifier $c(d = \text{source} \mid f(x))$.

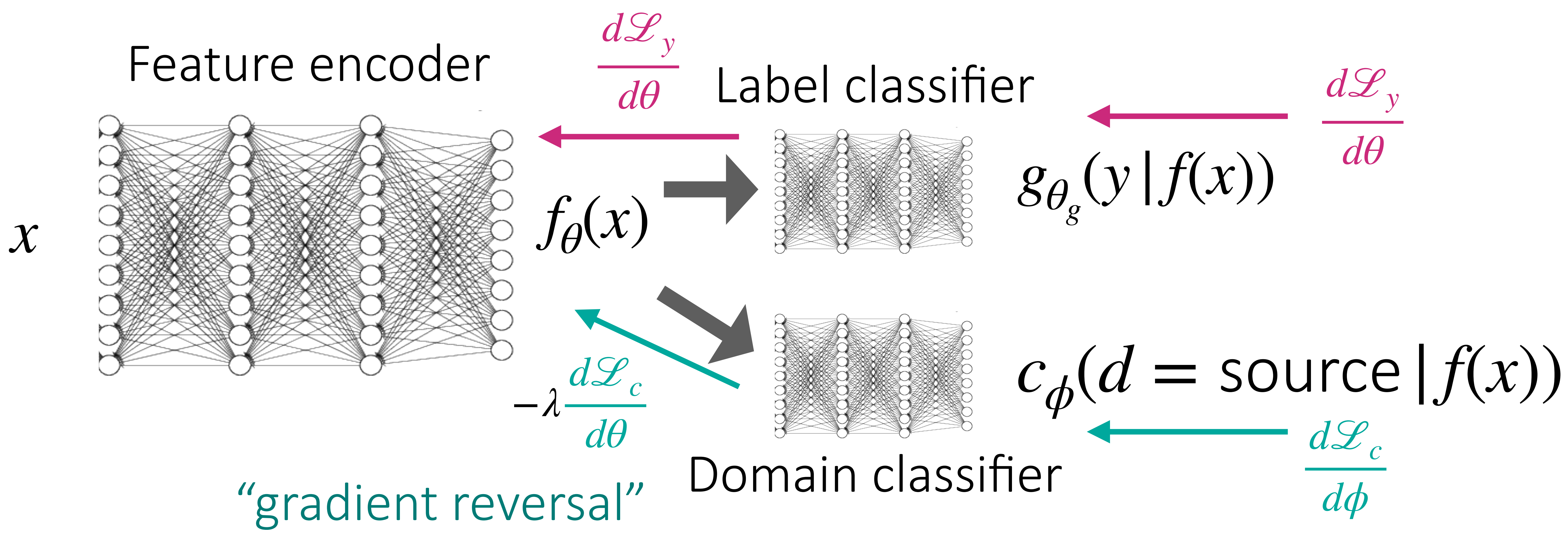


Minimize label prediction error & maximize "domain confusion"

Tzeng et al. Deep Domain Confusion. arXiv '14

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

Domain adaptation via feature alignment



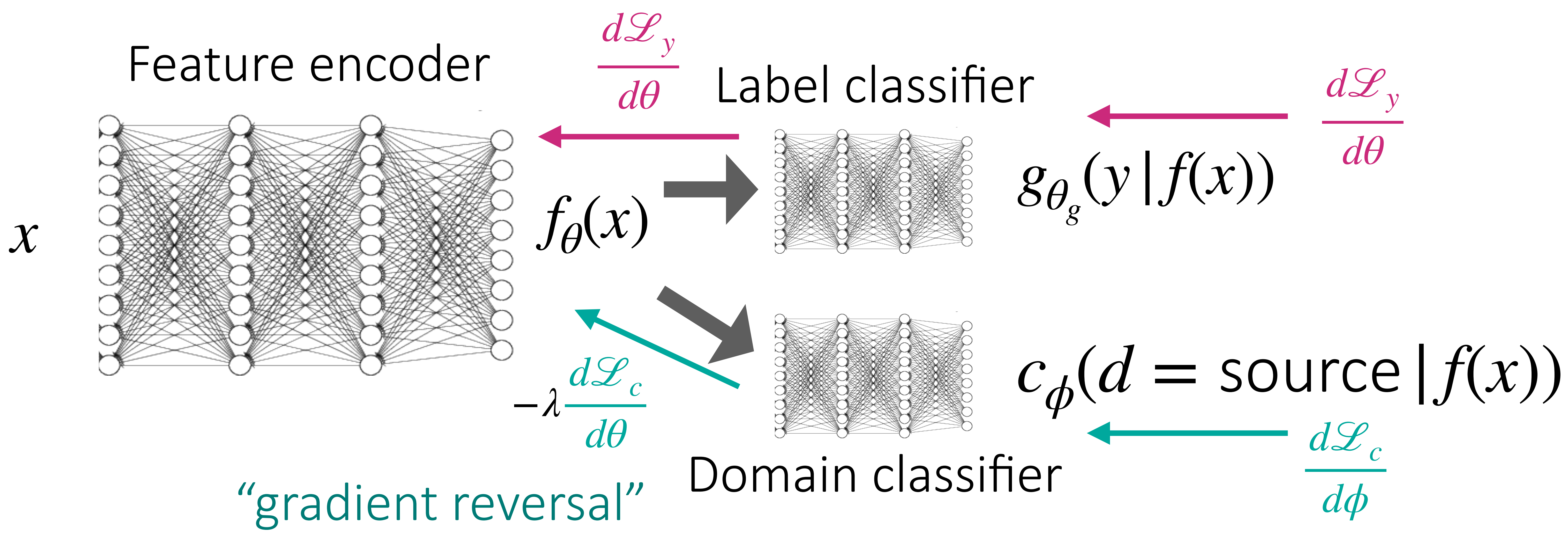
Full algorithm:

1. Randomly initialize encoder f_θ, label classifier g_{θ_g}, domain classifier c_ϕ
2. Update domain classifier: $\min_{\phi} \mathscr{L}_c = -\mathbb{E}_{x \sim D_S}[\log c_\phi(f(x))] - \mathbb{E}_{x \sim D_T}[1 - \log c_\phi(f(x))]$.
3. Update label classifier & encoder: $\min_{\theta, \theta_g} \mathbb{E}_{(x,y) \sim D_S}[L\left(g_{\theta_g}(f_\theta(x)), y\right)] - \lambda \mathscr{L}_c$
4. Repeat steps 2 & 3.

Tzeng et al. Deep Domain Confusion. arXiv ‘14

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

Domain adaptation via feature alignment



Slightly different forms of domain adversarial training.

Option 1: Maximize domain classifier loss (gradient reversal, same as GANs)

Option 2: Optimize for 50/50 guessing

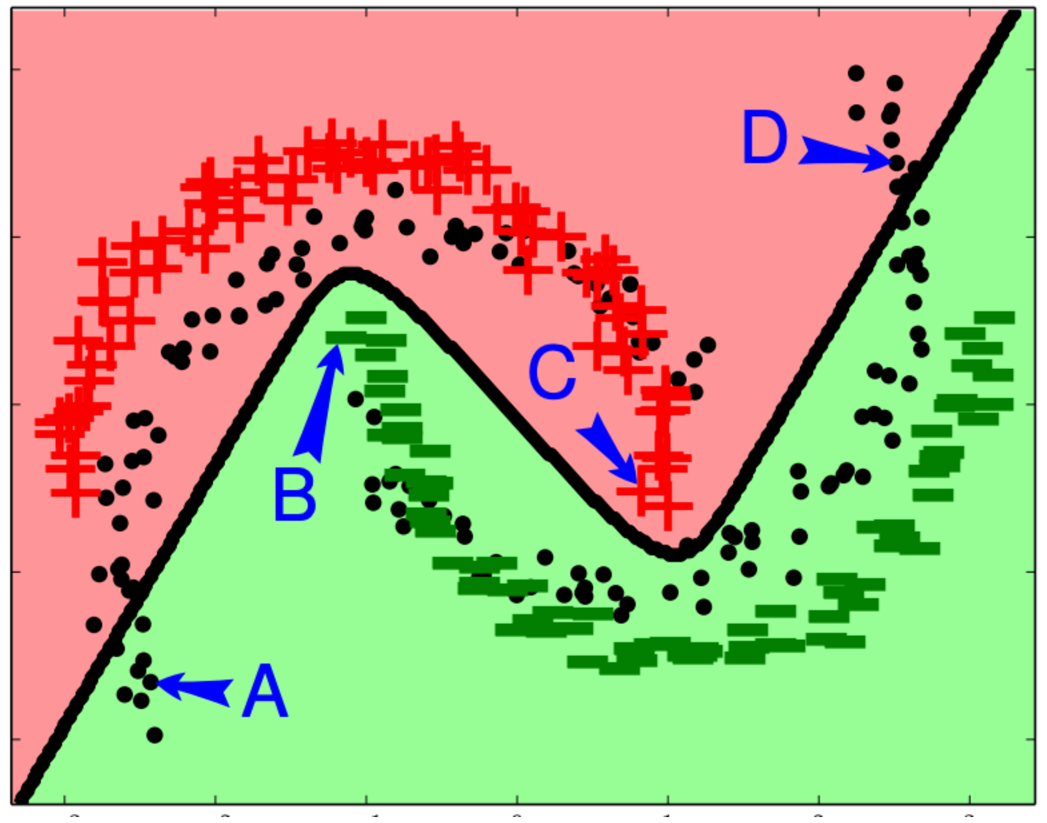
Tzeng et al. Deep Domain Confusion. arXiv '14
Ganin et al. Domain-Adversarial Training of Neural Networks. JMLR '16

Domain adaptation via feature alignment

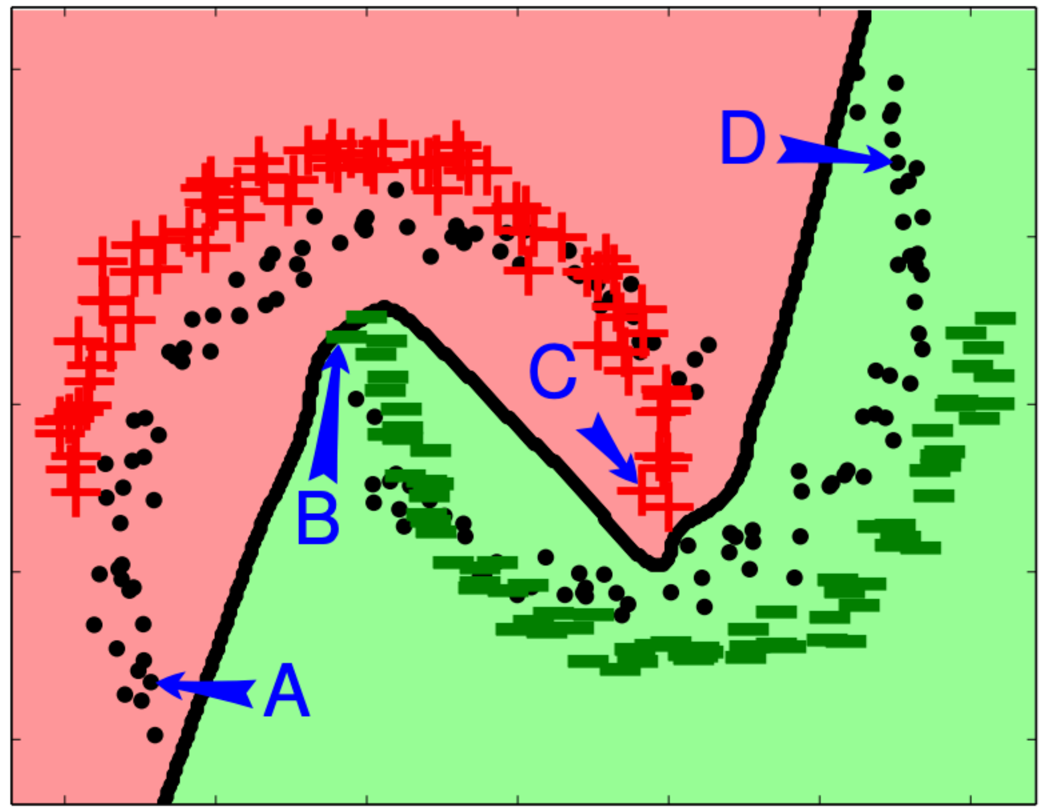
Toy example

source domain: +, —

target domain data: •



standard NN training



domain adversarial training

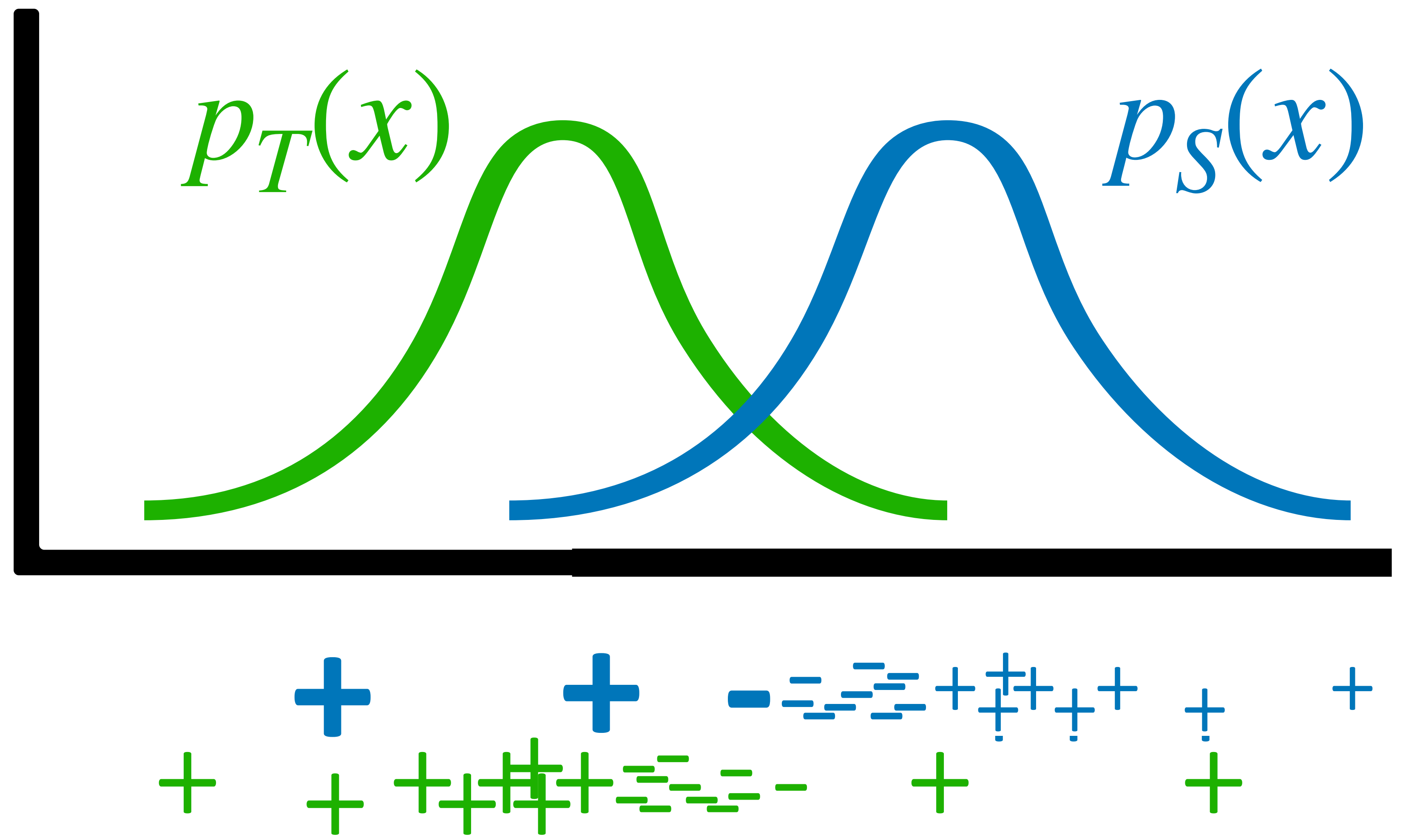
Source

Target

Method	Source / Target	MNIST / MNIST-M	Syn Numbers / SVHN	SVHN / MNIST	Syn Signs / GTSRB
Source only		.5225	.8674	.5490	.7900
DANN		**.7666** (52.9%)	**.9109** (79.7%)	**.7385** (42.6%)	**.8865** (46.4%)
Train on target		.9596	.9220	.9942	.9980

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

Importance weighting

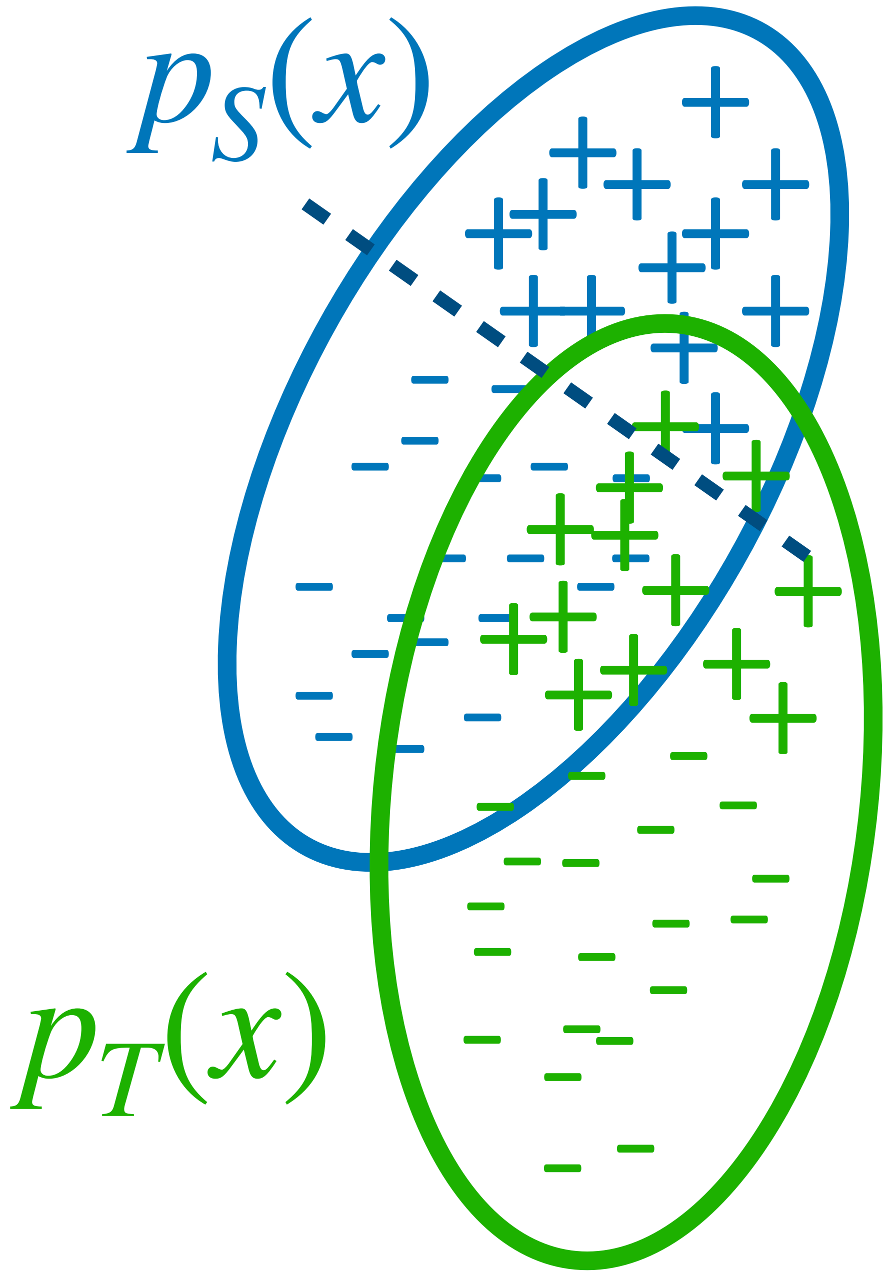


$$\min_{\theta} \mathbb{E}_{p_S(x,y)}\left[\frac{p_T(x)}{p_S(x)} L(f_\theta(x), y)\right]$$

\+ simple, can work well

— requires source distr. to cover target

Feature alignment



\+ fairly simple to implement, can work quite well

\+ doesn't require source data coverage

— involves adversarial optimization

— requires clear alignment in data

Plan for Today

Domain Adaptation
- Problem statements
- Algorithms
  - Data reweighting
  - Feature alignment

Domain Adaptation -> Domain Generalization

Goals for this lecture:

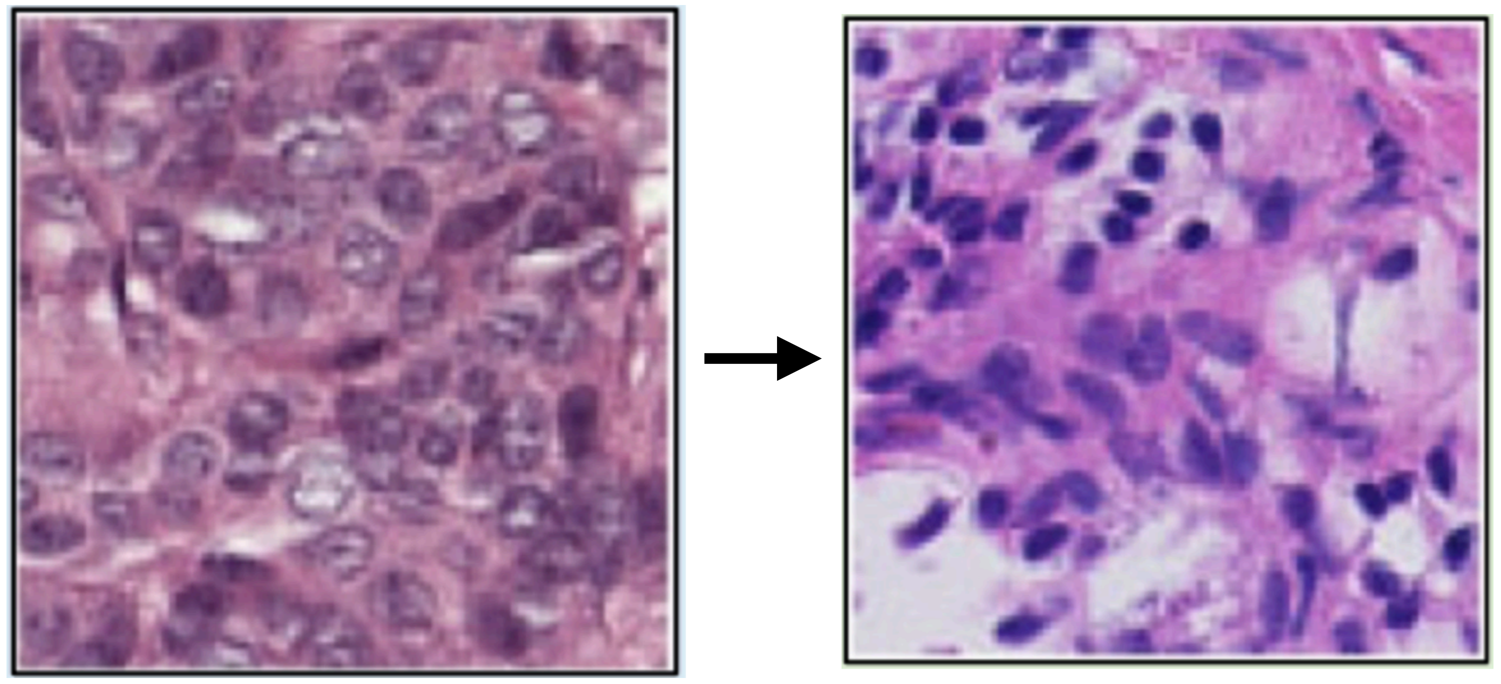
- Understand domain adaptation & generalization problems, how they relate to multi-task learning and transfer learning

- Understand two general approaches and when to use one vs. another

What if we don’t have unlabeled data from the target domain?

Domain adaptation

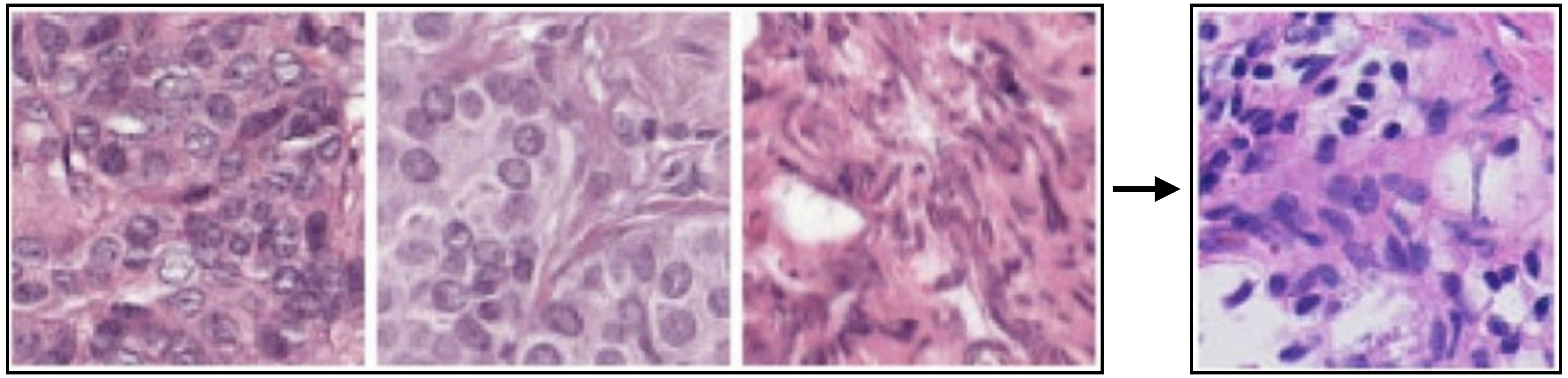
Source hospital Target hospital



- one source domain
- unlabeled data from target domain

Domain generalization

Source hospitals Target hospital



- multiple source domains
- no data from target domain
  (zero-shot generalize to new domain)

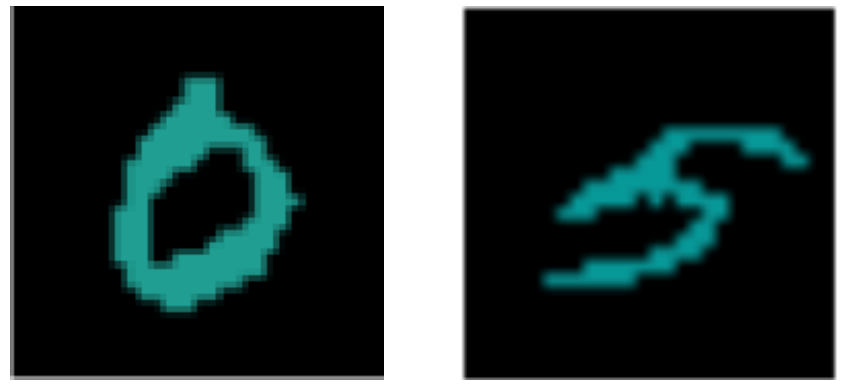
Toy example

Training data

Label: digit

Domain: color

Test data
(from new domain)



Features: (digit, color, style)

↓

prediction

Better Features: (digit, style)

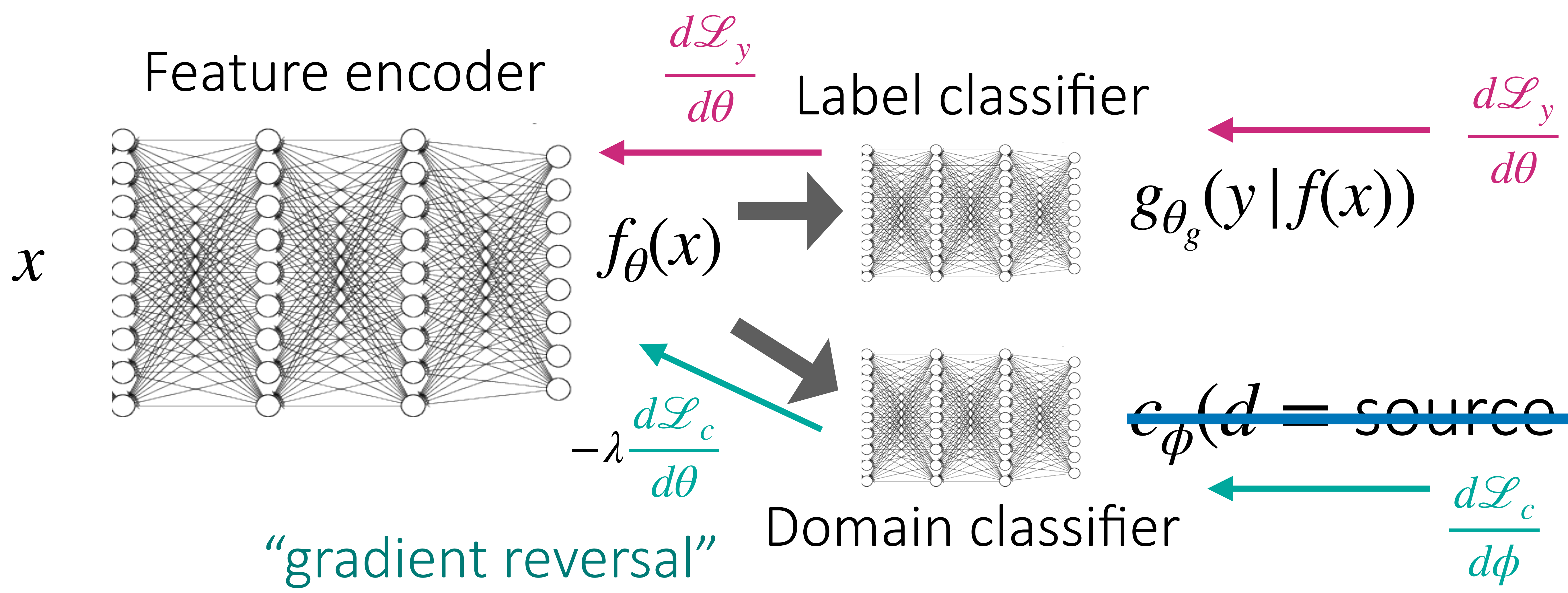
↓

prediction

These features are domain invariant!

A key concept in domain generalization: ***domain invariance***

How to learn domain invariant features?



Domain adversarial training!

Let d_x be domain label of example x.

Training

1. Randomly initialize encoder f_θ, label classifier g_{θ_g}, domain classifier c_ϕ
2. Update domain classifier: $\min_\phi \mathscr{L}_c = -\mathbb{E}_{x \sim D}[\log c_\phi(d = d_x \mid f(x))]$.
3. Update label classifier & encoder: $\min_{\theta, \theta_g} \mathbb{E}_{(x,y) \sim D_S}[L\left(g_{\theta_g}(f_\theta(x)), y\right)] - \lambda \mathscr{L}_c$
4. Repeat steps 2 & 3.

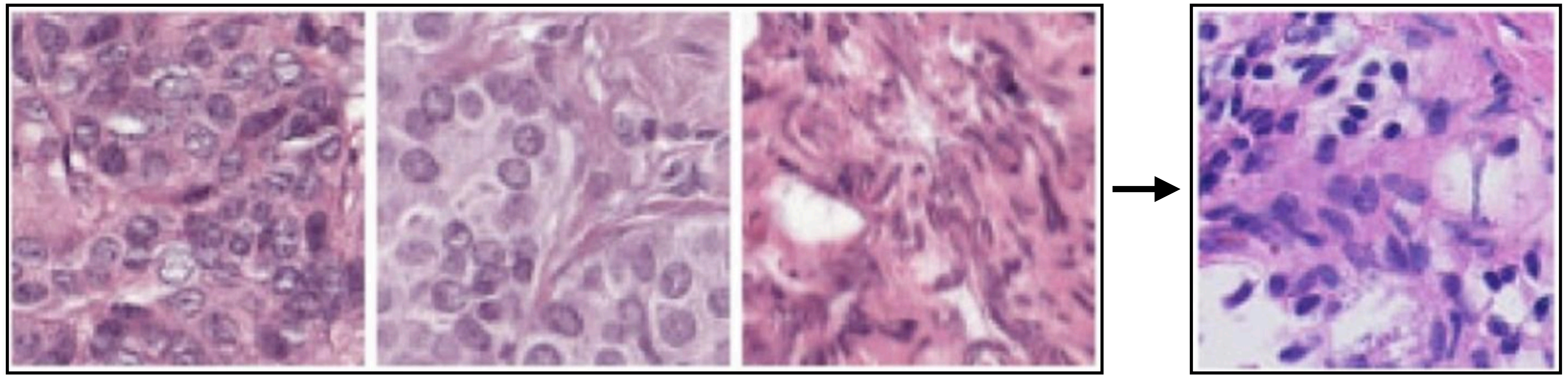
Testing

Apply model to examples from a new domain

Camelyon17 dataset

Source hospitals

Target hospital

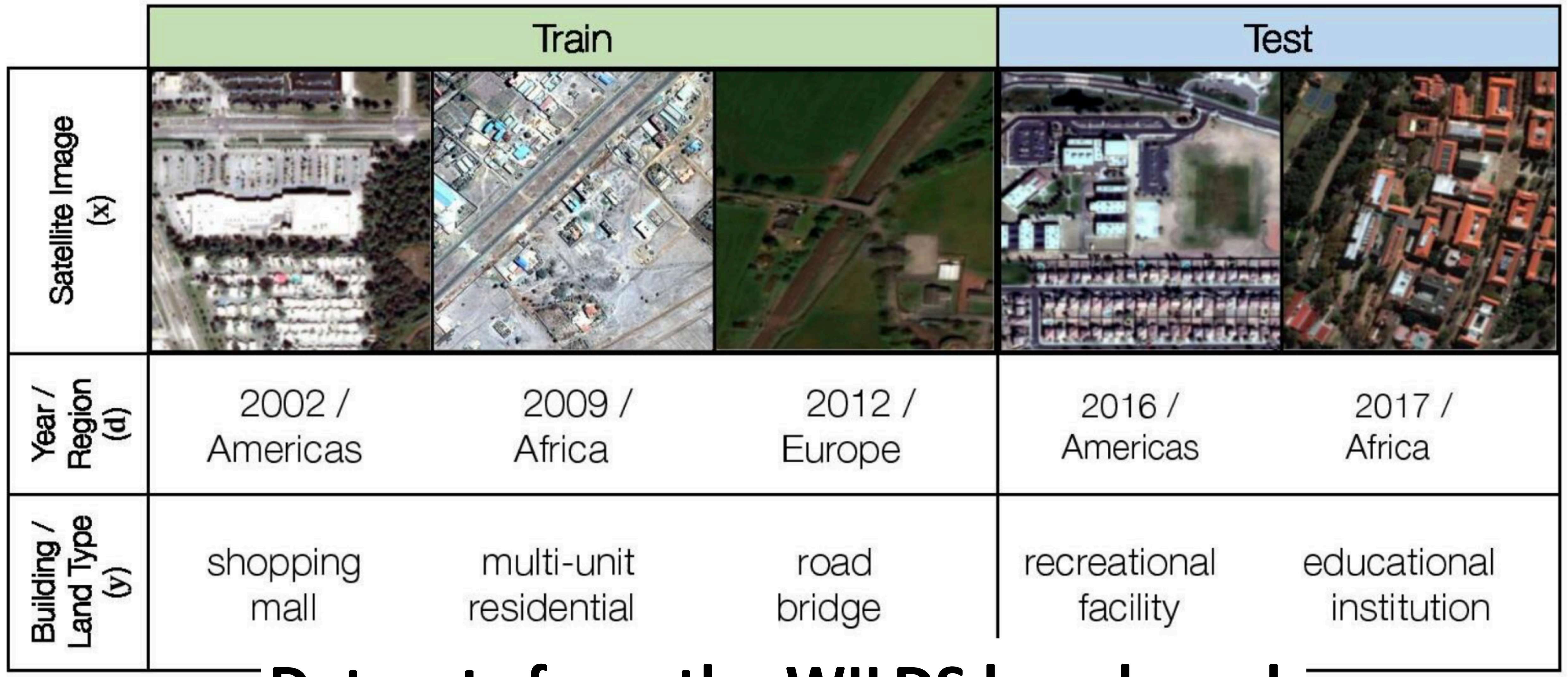


Accuracy on target hospital

ERM (standard training)	70.3%
Fish	74.7%
LISA	77.1%

(Methods that aim for domain invariance)

Functional Map of the World (FMoW) dataset



Accuracy on worst region

ERM (standard training)	32.3%
Fish	34.6%
LISA	35.5%

Datasets from the WILDS benchmark

When might this fail?

train

test

Perfect correlation between labels and domains

What will happen if you train for domain invariance in this case?

(pollev.com/330)

Another limitation: need to know the domain label for each example.

Summary

Domain: Tasks with $p(x)$ data distributions, same $p(y|x)$, $\mathscr{L}$

Domain adaptation	**Domain generalization**
Adapt w/ unlabeled target domain data	Zero-shot generalize to new domain
As few as ***one domain*** in training data	Need data from ***multiple*** training domains

Two general approaches:

Reweight the data
if you have good coverage

Encourage domain invariance
if there is a clear way to align features

Plan for Today

Domain Adaptation
- Problem statements
- Algorithms
  - Data reweighting
  - Feature alignment

Domain Adaptation -> Domain Generalization

Goals for this lecture:

- Understand domain adaptation & generalization problems, how they relate to multi-task learning and transfer learning

- Understand two general approaches and when to use one vs. another

Course Reminders

Poster session next **Wednesday**.

Project report due the following **Monday**

Azure: Form on Ed for requesting more credits for project.

Next time: Frontiers & Open Problems!

Time Permitting

Which frontiers would you rather see in the frontiers & open problems lecture?

1. meta reinforcement learning
2. meta-learning for adapting LLMs, VLMs