Project and Probe: Sample-Efficient Domain Adaptation by Interpolating Orthogonal Features

Annie S. Chen^{*1}, Yoonho Lee^{*1}, Amrith Setlur², Sergey Levine³, Chelsea Finn¹ Stanford University¹, Carnegie Mellon University², UC Berkeley³



Deep Learning: Classics and Trends, ML Collective March 17, 2023



Distribution Shifts



Figures from "Wilds: A benchmark of in-the-wild distribution shifts" (2021)

Transfer Learning for Adaptation

data from the new *target domain*.

Problem setting:

Backbone pre-trained on generic large dataset (optional)

Large dataset from relevant source domain

A reliable way of adapting to distribution shifts: leverage a small amount of labeled

Small dataset from target domain

Less inductive bias, more flexible

- + Reusable
- Inflexible
- Susceptible to shortcuts

- Fine-Tuning w/Target
- + Adaptive
- May overfit

- Susceptible to shortcuts

(a) **Project with Large Source Dataset**

Project and Probe

(b) **Probe with Small Target Dataset**

Step 1: **Project** with Source Data

(a) **Project with Large Source Dataset**

 $\Pi_i = \arg\min \mathbb{E}_{(x,y)\sim \mathcal{D}_S} \mathcal{L}(\Pi_i(f))$

For binary classification:

- Initialize d classifiers (D x d matrix)
- Train each classifier with:
 - Cross-entropy loss on source data
 - Orthogonality constraint w.r.t. all previous

$$f(x)$$
, y s.t. $\Pi_j \perp \Pi_i$ for all $j < i$.

Step 1: Project with Source Data


```
def learn_feature_space_basis(x, y, num_features):
    projection = torch.nn.Linear(x.shape[1], num_features)
    opt = torch.optim.AdamW(projection.parameters(), lr=0.01, weight_decay=0.01)
    max_steps = 100
    for i in range(max_steps):
        logits = projection(x)
        loss = F.binary_cross_entropy_with_logits(logits, y, reduction="none").mean()
        opt.zero_grad()
        loss.backward()
        opt.step()
        # Enforce orthogonality; we're performing projected gradient descent
        Q, R = torch.linalg.qr(linear_model.weight.detach().T)
        projection.weight.data = (Q * torch.diag(R)).T
   feature_space = projection.weight.detach().T
    return feature_space
```

Simple: 15 lines of PyTorch code!

(a) **Project with Large Source Dataset**

Step 2: Probe with Target Data

(b) **Probe with Small Target Dataset**

 $\arg\min \mathbb{E}_{(x,y)\sim \mathcal{D}_T} \mathcal{L}(g(\Pi(f(x))), y).$

Project and Probe (Pro²)

(a) **Project with Large Source Dataset**

Project: Learn linear projection of pre-trained embeddings onto orthogonal directions **Probe**: Interpolate between projected features w/ a small target dataset

+ Very lightweight: 30,000 experiments in <24 hrs, on CPUs only!

(b) **Probe with Small Target Dataset**

Pro² induces a favorable bias-variance tradeoff

Theorem 3 (bias-variance tradeoff). When the conditions in Lemma 2 hold and when $\|\mathbf{x}\|_{\infty} = \mathcal{O}(1)$, for B-bounded loss l, w.h.p. $1 - \delta$, the excess risk for the solution \hat{w}_d of PRO² that uses d features is $\mathcal{L}_T(\hat{w}_d) - \delta$ $\min_{\mathbf{w}\in\mathcal{W}}\mathcal{L}_T(\mathbf{w})$

$$\lesssim \|(\boldsymbol{I}_D - \boldsymbol{\Pi}_d) \mathbf{w}_T^*\|_2 + \left(\frac{\sqrt{d} + B\sqrt{\log(1/\delta)}}{\sqrt{M}}\right)$$

where the first term controls the bias and the second controls the variance.

A small dataset entails high variance.

We can reduce variance with a low-dimensional projection.

The projection introduces additional bias, which is low when the most important directions are covered (possible when the shift is not too severe).

Pro² recovers important & diverse features to reduce bias.

(2)

For a small dataset size, bias & variance can be balanced using a smaller projection dim if important directions are covered.

Project and Probe (Pro²)

(a) **Project with Large Source Dataset**

1) 2) Probe: Interpolate b/w projected features w/ a small amt of target data

(b) **Probe with Small Target Dataset**

- Project: Learn linear projection of pre-trained embeddings onto orthogonal directions
 - + Adaptive b/c projection learns diverse (orthogonal) features + Efficient b/c projection learns useful (predictive) features

Waterbirds

Comparisons

Random Projection: Project onto random orthogonal features DFR (Kirichenko et al. 2022): standard linear probing on target data

Experiments

Camelyon17

4-way Collages

- Teney et al. 2022: minimize alignment of input gradients over pairs of features

Waterbirds

CelebA

Test Accuracy

Target Train Data Size

Camelyon17

Collages

0.8

0.7

0.6 **t**

Test Accuracy

Camelyon17

Collages

32

8

Target Train Data Size

Random Projection
DFR (Kirichenko et al.)

Waterbirds

Test Accuracy

Camelyon17

Collages

Target Train Data Size

Waterbirds

CelebA

Test Accuracy

Camelyon17

Collages

Pro² bias-variance tradeoff

Waterbirds

Pro ² (ours) on Spurious											
1024-	89.3	90.0	96.4								
256-	89.0	90.8	92.2	97.3							
64-	90.0	91.2	95.9	97.4							
16-	90.1	94.3	96.6	97.9							
4 -	95.0	95.7	97.0	97.3							
1-	97.3	97.2	97.2	97.4							

Pro ² (ours) on Balanced										
1024-	81.6	86.4	93.6							
256-	82.9	85.5	90.2	93.7						
64-	81.9	87.6	91.4	94.3						
16-	84.2	89.6	92.4	92.9						
4 -	90.3	92.3	92.5	92.4						
1-	92.2	92.0	91.8	92.6						

Pro² (ours) on Minority

1024 -	83.3	91.6	95.1	97.0						
256-	82.5	91.1	94.3	95.3						
64 -	83.1	88.3	91.7	93.7						
16-	78.1	82.8	87.7	89.2						
4 -	84.3	86.9	88.3	89.8						
1	87.3	86.2	88.3	88.3						
	ż	8	32	128						

Number of Features

More severe shifts

Smaller

shifts

Target Train Data Size

Pro² bias-variance tradeoff

\sim

Waterbirds						Cele	ebA		Camelyon17					Collages					
Pro ² (ours) on Spurious				Pro ² (ours) on Spurious				Pro ² (ours) on ID Test			Pro ² (ours) on Collages-MNIST								
1024	89.3	90.0	91.9	96.4	1024 -	93.6	96.0	96.9	97.7	1024 -	80.3	83.3	88.9	91.4	1024	71.9	84.8	95.5	98.0
256-	89.0	90.8	92.2	97.3	256	94.2	95.8	96.9	97.6	256	79.0	87.0	88.8	91.7	256	73.4	87.4	95.9	97.7
64 -	90.0	91.2	95.9	97.4	64	94.6	95.1	96.2	97.0	64	81.7	87.8	90.7	93.1	64 ·	74.1	88.3	92.9	95.1
16	90.1	94.3	96.6	97.9	16	94.7	96.2	96.6	96.9	16-	85.3	87.7	92.1	93.8	16	65.1	87.7	90.6	93.3
4 -	95.0	95.7	97.0	97.3	4 ·	97.2	96.9	96.8	97.0	4.	81.1	90.6	93.0	94.5	4 ·	75.0	86.8	91.0	91.9
1	97.3	97.2	97.2	97.4	1	98.0	97.8	97.8	97.9	1.	80.8	93.3	93.4	93.6	1.	70.1	74.7	74.9	75.7
	Pro ²	(ours)	on Balan	iced		Pro ²	(ours) (on Balan	iced	. '	Pro ²	(ours) o	n OOD ⁻	Test	Pro ² (ours) on Collages-CIFAR				
1024-	81.6	86.4	90.2	93.6	1024 -	63.2	70.4	76.6	83.7	1024	68.4	86.2	90.8	93.5	1024	79.3	83.5	88.6	90.9
256-	82.9	85.5	90.2	93.7	256-	64.2	70.5	80.4	83.3	256-	79.8	87.4	90.8	93.4	256	78.9	84.6	88.0	90.3
64 -	81.9	87.6	91.4	94.3	64	66.7	73.3	80.7	83.1	64	83.0	88.0	90.8	92.1	64 ·	79.3	82.4	83.4	86.2
16-	84.2	89.6	92.4	92.9	16-	69.9	78.0	80.4	80.7	16	86.0	85.9	90.6	92.5	16	77.6	82.3	83.0	83.6
4 -	90.3	92.3	92.5	92.4	4-	70.5	76.3	76.9	77.5	4 -	86.3	91.4	92.3	92.9	4 -	76.8	78.5	80.8	81.8
1	92.2	92.0	91.8	92.6	1.	63.2	61.5	62.4	63.9	1.	90.9	91.9	93.2	93.2	1.	62.9	68.6	71.8	71.9
	Pro	² (ours)	on Mino	rity		Pro	² (ours)	on Mino	rity		ż	8	32	128	Pro	2 (ours)	on Colla	ges-Fas	hion MN
1024 -	83.3	91.6	95.1	97.0	1024 -	88.1	90.9	93.7	96.0						1024 ·	59.0	69.8	79.8	85.0
256-	82.5	91.1	94.3	95.3	256-	89.8	91.0	93.2	94.6						256	58.6	65.3	78.3	83.3
64 -	83.1	88.3	91.7	93.7	64	87.0	91.6	90.8	93.1						64 ·	58.0	63.0	70.3	76.8
16	78.1	82.8	87.7	89.2	16	90.2	90.8	90.3	89.2						16	55.6	58.0	61.8	66.6
4 -	84.3	86.9	88.3	89.8	4	85.6	86.0	86.3	84.2						4 -	57.3	60.4	62.3	66.6
1	87.3	86.2	88.3	88.3	1	73.8	75.1	76.8	76.0						1.	57.4	61.0	59.2	61.2
	ż	8	32	128		ż	8	32	128							2	8	32	128

Smaller shifts

Number of Features

More severe shifts

...

Target Train Data Size

Orthogonality is important for learning a diverse set of features

Pro² improves with better pretrained feature extractors

Takeaways

Pro2 is a lightweight, sample-efficient framework for adaptation. **Project:** extract a diverse + predictive feature-space basis **Probe:** interpolate to adapt to varying target distributions

low data settings.

- Standard linear probing may not be best for few-shot adaptation.
- Pro2 better balances this tradeoff by learning diverse predictive features.

Key Insight: The tradeoff between expressivity and inductive bias is critical in

Future Work

Interesting future directions, including:
(1) Extending to *other problem settings*, such as active learning
(2) Exploring other methods to determine a *good feature-basis for adaptation*(3) *Integrating with other fine-tuning methods* to further improve performance
(4) Select features to use in an *unsupervised* fashion

Paper: https://arxiv.org/pdf/2302.05441.pdf Emails: asc8@stanford.edu and yoonho@stanford.edu

Annie S. Chen*

Yoonho Lee*

Thank you!

Amrith Setlur

Sergey Levine

Chelsea Finn