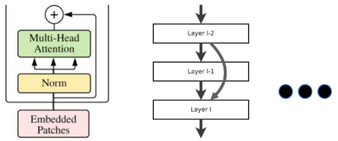
ModelDiff: A Framework for Comparing Learning Algorithms

Harshay Shah*, Sung Min Park*, Andrew Ilyas*, Aleksander Mądry

Comparing Learning Algorithms

ML pipelines entail many design choices

Model architecture

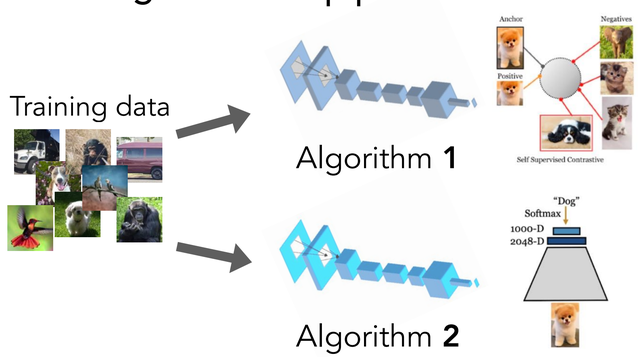


Transformers **or** ResNets?

Augmentation schemes

Random Crop **or** Flip **or** Median Blur?

Recurring Q: Which pipeline to choose?



Conventional approach: Performance comparisons

ModelDiff: Model-agnostic feature comparisons

Algorithm Comparisons with ModelDiff

Objective: *Find input-space feature transformations F that disparately impact models trained with two different algorithms:*

$$\overbrace{\mathbb{E}[L_1(F(x), y_c) - L_1(x, y_c)]}^{\text{Counterfactual effect of } F \text{ on } M_1} \geq \delta \qquad \overbrace{\mathbb{E}[L_2(F(x), y_c) - L_2(x, y_c)]}^{\text{Counterfactual effect of } F \text{ on } M_2} \leq \epsilon$$

Approach: *Compare <u>how training examples in influence models</u> trained with different algorithms*

Case study: Study effect of ImageNet pre-training with ModelDiff

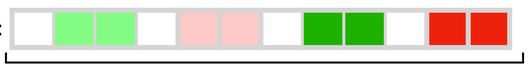
Setup: Compare models trained on Waterbirds data with and without ImageNet pre-training

Algorithm 1: Fine-tune ImageNet model **Algorithm 2:** Train from scratch

Step 1: Compute *datamodels* for each algorithm

Example x

$\rightarrow \theta_x^{(1)} =$ 

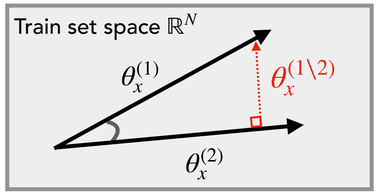
$\rightarrow \theta_x^{(2)} =$ 

j^{th} *coordinate = dependence on* j^{th} *training example*

All N training examples

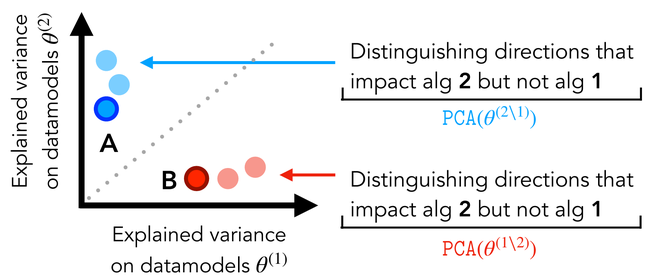
Datamodels of algorithms **1** and **2** share the same (training set) space!

Step 2: Analyze *residual datamodels*



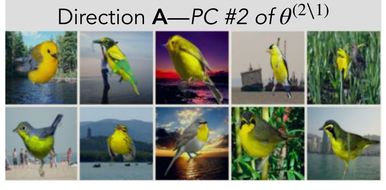
Residual datamodels capture training directions specific to algorithm **1** but not **2**, and vice-versa

PCA on residual datamodels extract *distinguishing training directions*



Each PC corresponds to a *distinguishing subpopulation* of test examples

Direction **A**—*PC #2 of* $\theta^{(2\setminus 1)}$



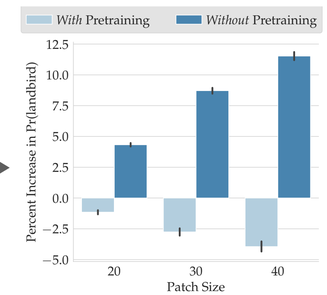
Direction **B**—*PC #2 of* $\theta^{(1\setminus 2)}$

Step 3: Infer + Test distinguishing transformations

Without ImageNet pre-training, models spuriously rely on "yellow color"

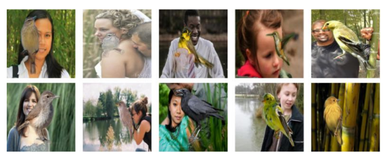
Subpopulation **A**: "Yellow color"

"Yellow color" feature transformation



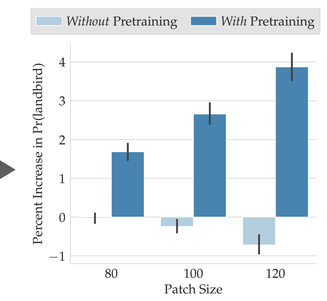
$\Pr(\hat{y} = \text{landbird} \mid \text{do(yellow color)})$

With ImageNet pre-training, models spuriously rely on "human face"



Subpopulation **B**: "Human face"

"Human face" feature transformation



$\Pr(\hat{y} = \text{landbird} \mid \text{do(human face)})$

Takeaways

→ ModelDiff = data-centric comparisons of learning algorithms

→ Datamodels = model-agnostic embeddings in train set space

→ Verify distinguishing transformations via counterfactuals

Paper Code Blog post