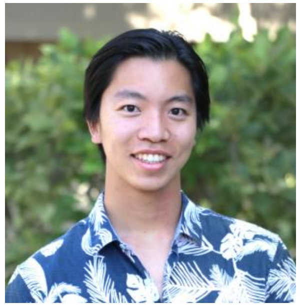
Connect, Not Collapse: Explaining Contrastive Learning for Unsupervised Domain Adaptation


Kendrick Shen*


Robbie Jones*


Ananya Kumar*


Sang Michael Xie*

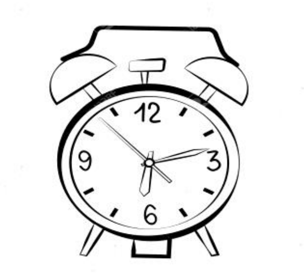
Jeff Z. HaoChen


Tengyu Ma


Percy Liang

Unsupervised domain adaptation (UDA)

Labeled source domain



Clock

Unsupervised domain adaptation (UDA)

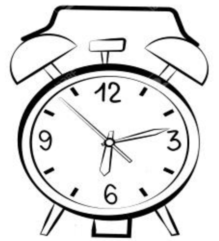
Labeled source domain

Clock

Unlabeled target domain

?

Unsupervised domain adaptation (UDA)

Labeled source domain	Unlabeled target domain
	
Clock	*?*

Goal: high accuracy on target domain (without labels)

Classical approach for UDA

Labeled source domain

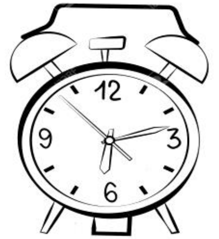
Source representations

Unlabeled target domain

Target representations

Classical approach for UDA

Labeled source domain



Source representations

High accuracy (given labels)

Unlabeled target domain

Target representations

Classical approach for UDA

Labeled source domain

Source representations

High accuracy (given labels)

Unlabeled target domain

Target representations

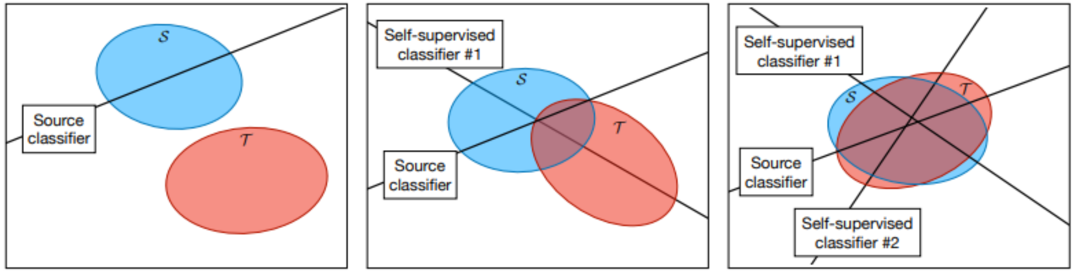
Match distributions

Classical approach for UDA

Motivated by theories such as $H\Delta H$ divergence (Ben-David et al 2010): want source and target reps to be “indistinguishable” to get good target accuracy

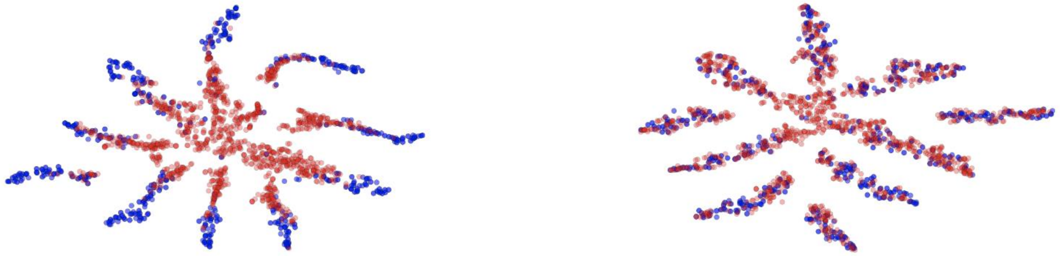
Classical approach for UDA

Motivated by theories such as $H\Delta H$ divergence (Ben-David et al 2010): want source and target reps to be “indistinguishable” to get good target accuracy



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

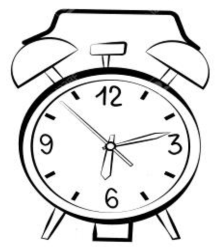
UDA-SS (Sun et al. 2019)


(a) Non-adapted (b) Adapted

DANN (Ganin et al. 2016)

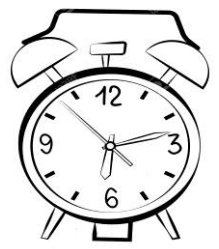
Pre-training for UDA

Step 1: pre-train on unlabeled data (combined source + target)



Pre-training for UDA

Step 1: pre-train on unlabeled data (combined source + target)



Step 2: fine-tune on labeled data (source)

Pre-training for UDA

Step 1: pre-train on unlabeled data (combined source + target)

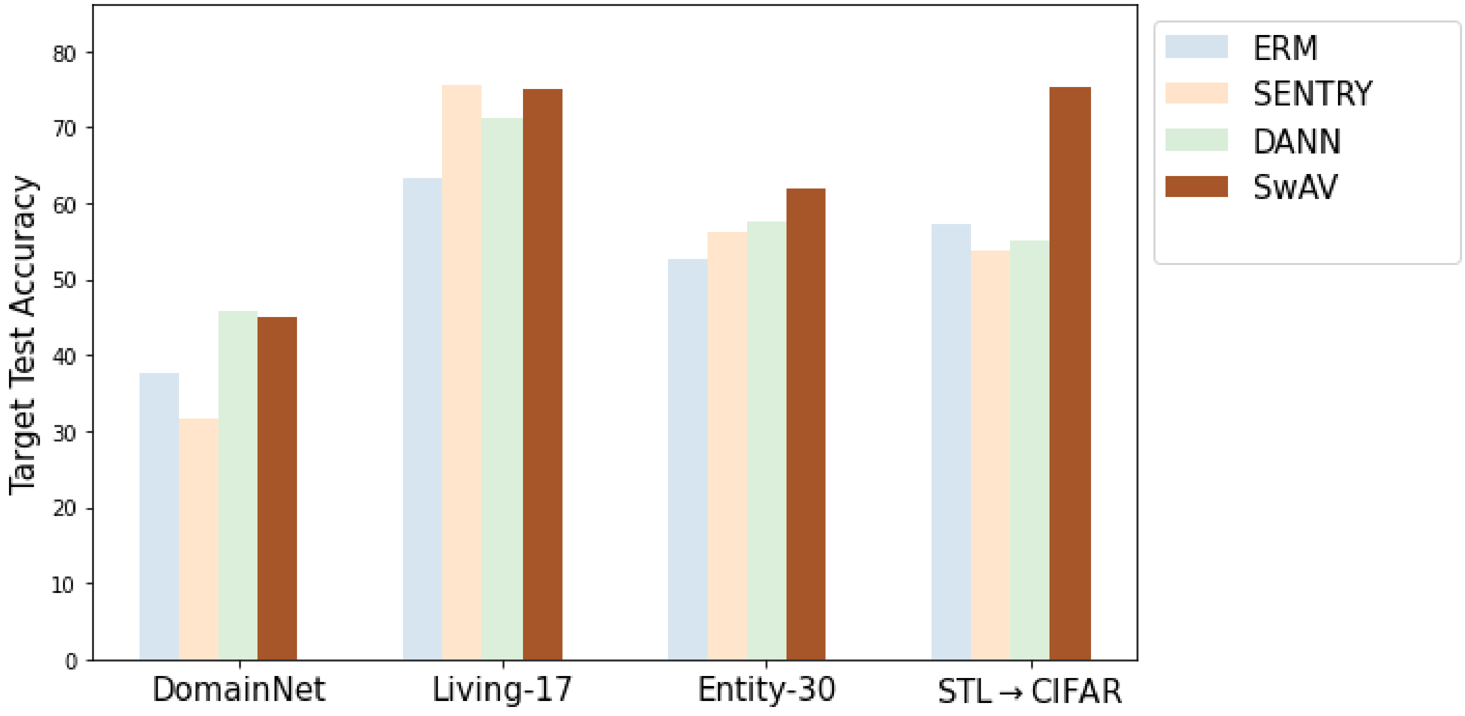
Step 2: fine-tune on labeled data (source)

Step 3: evaluate accuracy (target)

Inspired by e.g., Blitzer et al 2007

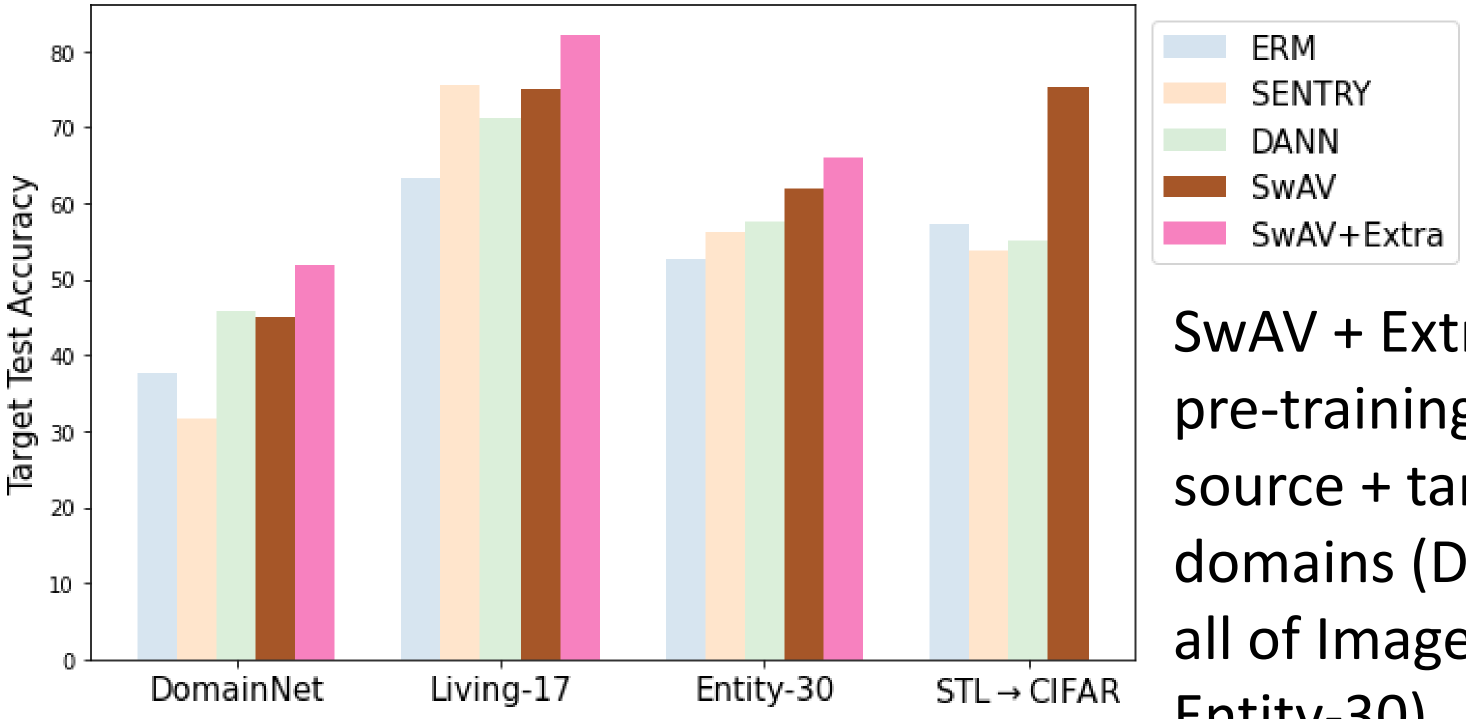
Contrastive pre-training for UDA

Contrastive pre-training (SwAV, Caron et al. 2020) is competitive with UDA methods (even when all methods use the same augmentations)



Contrastive pre-training for UDA

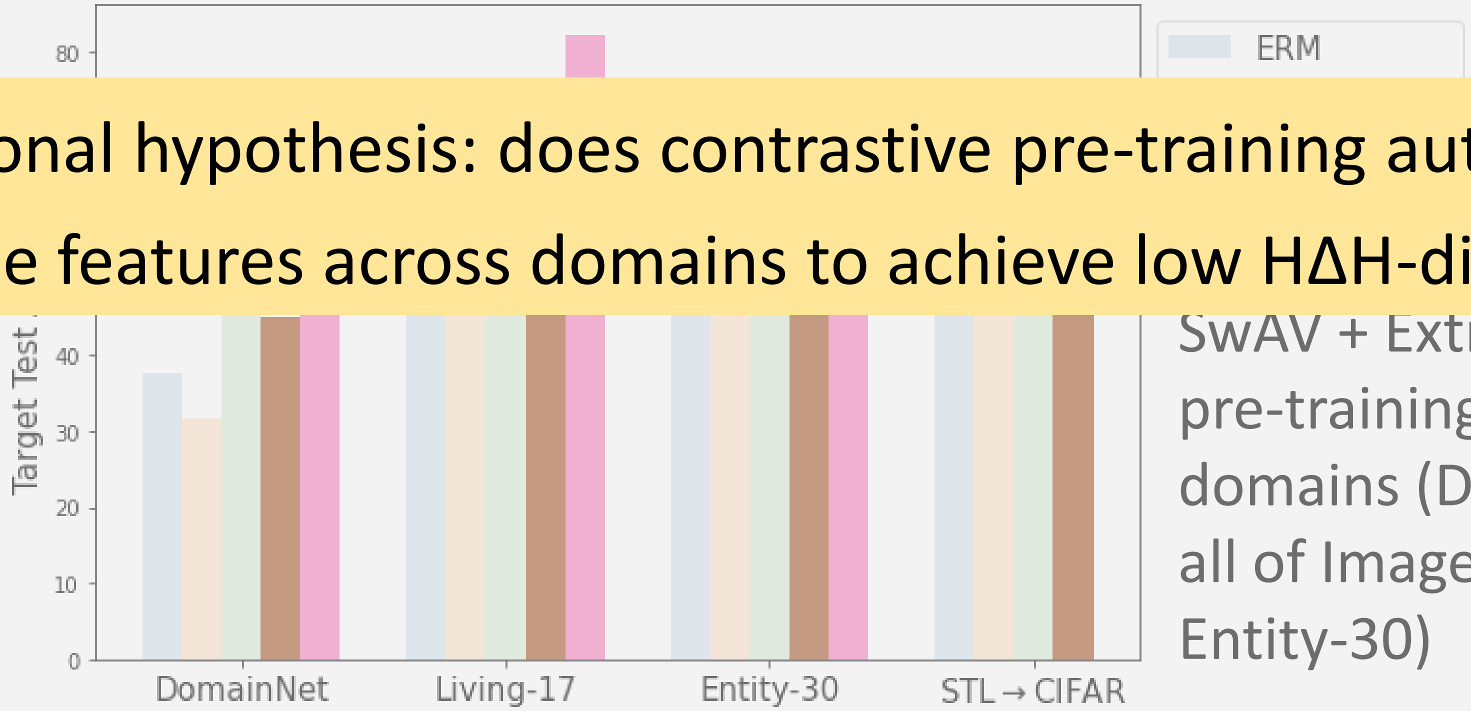
Contrastive pre-training (SwAV, Caron et al. 2020) is competitive with UDA methods (even when all methods use the same augmentations)



SwAV + Extra: unlabeled pre-training data beyond source + target = all 4 domains (DomainNet) or all of ImageNet (Living-17, Entity-30)

Contrastive pre-training for UDA

Contrastive pre-training (SwAV, Caron et al. 2020) is competitive with UDA methods (even when all methods use the same augmentations)

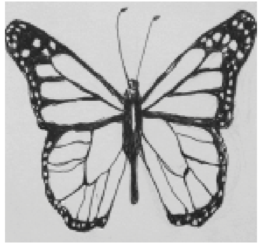
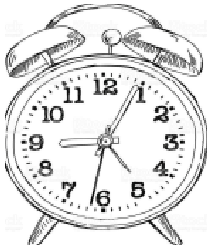


Conventional hypothesis: does contrastive pre-training automatically merge the features across domains to achieve low HΔH-divergence?

SwAV + Extra: unlabeled pre-training data = all 4 domains (DomainNet) or all of ImageNet (Living-17, Entity-30)

Contrastive pre-training doesn't bring domains together

Inspect DANN vs contrastive learning features: train discriminator between domains or between classes

	Domain 1 (Sketch)	Domain 2 (Real)
Class 1 (Butterfly)		
Class 2 (Clock)		

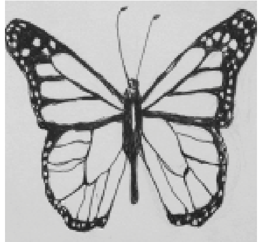
Contrastive pre-training doesn’t bring domains together

Inspect DANN vs contrastive learning features: train discriminator between domains or between classes

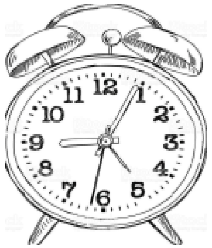
Domain 1 (Sketch)

Domain 2 (Real)

Class 1 (Butterfly)



Class 2 (Clock)

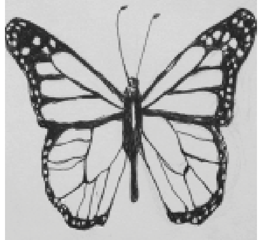
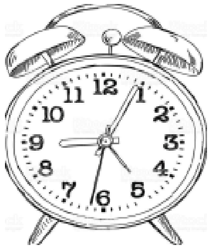


Between domains

Contrastive: 8% err

Contrastive pre-training doesn’t bring domains together

Inspect DANN vs contrastive learning features: train discriminator between domains or between classes

	Domain 1 (Sketch)		Domain 2 (Real)
Class 1 (Butterfly)			
Class 2 (Clock)		Between domains DANN: 14% err Contrastive: 8% err	

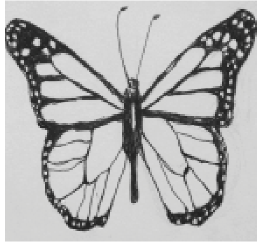
Contrastive pre-training doesn't bring domains together

Inspect DANN vs contrastive learning features: train discriminator between domains or between classes

Domain 1 (Sketch)

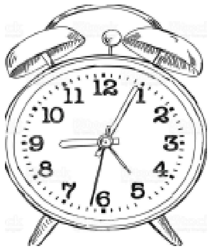
Domain 2 (Real)

Class 1
(Butterfly)



Between classes
DANN: 6% err
Contrastive: 7% err

Class 2
(Clock)



Between domains
DANN: 14% err
Contrastive: 8% err

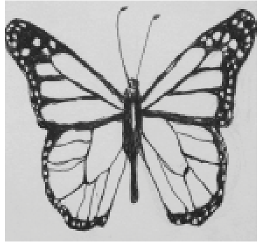
Contrastive pre-training doesn't bring domains together

Inspect DANN vs contrastive learning features: train discriminator between domains or between classes

Domain 1 (Sketch)

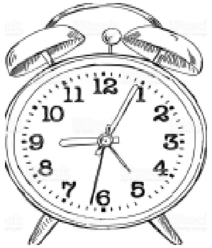
Domain 2 (Real)

Class 1
(Butterfly)



Between classes
DANN: 6% err
Contrastive: 7% err

Class 2
(Clock)



Between domains
DANN: 14% err
Contrastive: 8% err

Pre-training does not produce domain invariant features,
and domains are about as "far apart" as classes!

Contrastive pre-training for UDA

- Performs competitively with strong baselines: SENTRY (Prabhu et al. 2021), DIRT-T (Shu et al. 2018), and DANN (Ganin et al. 2016)

Contrastive pre-training for UDA

- Performs competitively with strong baselines: SENTRY (Prabhu et al. 2021), DIRT-T (Shu et al. 2018), and DANN (Ganin et al. 2016)
- Instead of collapsing domains together, learns features that vary substantially across domains

Contrastive pre-training for UDA

- Performs competitively with strong baselines: SENTRY (Prabhu et al. 2021), DIRT-T (Shu et al. 2018), and DANN (Ganin et al. 2016)
- Instead of collapsing domains together, learns features that vary substantially across domains

Why do these features still generalize to the target without domain invariance?

Outline

- Setup: augmentation graph
- Intuitions and theoretical results
  - Main intuitions (toy example)
  - Results for stochastic block model & beyond
  - Contrastive pre-training vs. ERM & DANN
- Test theoretical predictions on real data

Outline

- Setup: augmentation graph
- Intuitions and theoretical results
  - Main intuitions (toy example)
  - Results for stochastic block model & beyond
  - Contrastive pre-training vs. ERM & DANN
- Test theoretical predictions on real data

Setup: augmentation graph

- Contrastive learning hinges on *positive pairs* (augmentations of the same original input)

Setup: augmentation graph

- Contrastive learning hinges on *positive pairs* (augmentations of the same original input)
- Contrastive objective:
  - map positive pairs to similar features

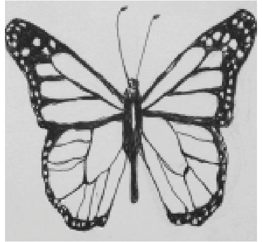
Setup: augmentation graph

- Contrastive learning hinges on *positive pairs* (augmentations of the same original input)
- Contrastive objective:
  - map positive pairs to similar features
  - map augmentations of different inputs to different features

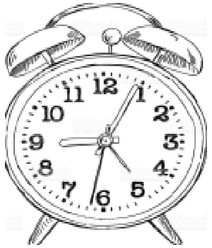
Setup: augmentation graph

Domain 1 (Sketch) | Domain 2 (Real)

Class 1 (Butterfly)



Class 2 (Clock)



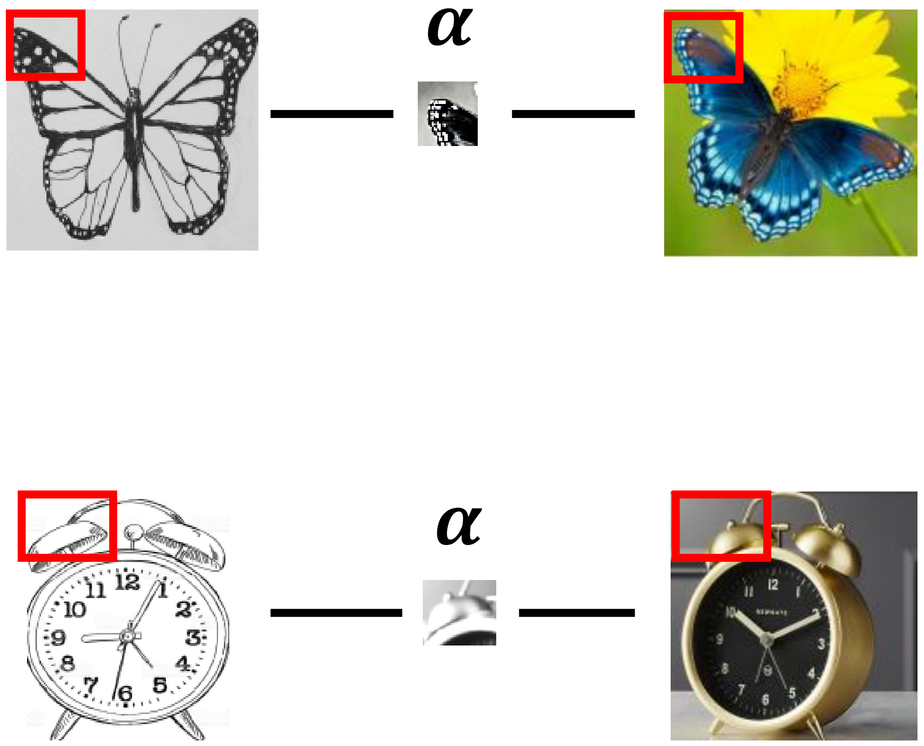
Setup: augmentation graph

Domain 1 (Sketch)

Domain 2 (Real)

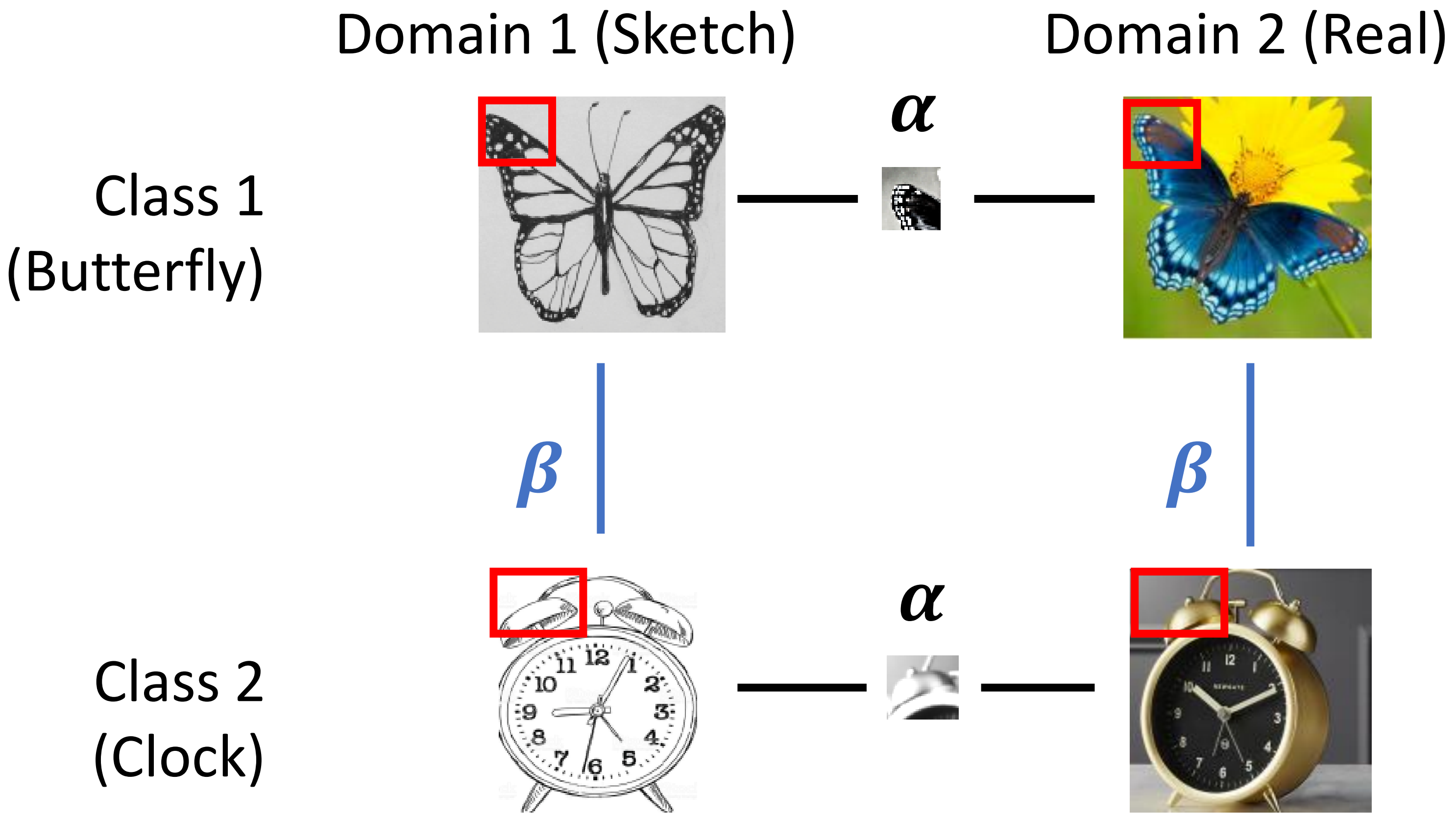
Class 1
(Butterfly)

Class 2
(Clock)

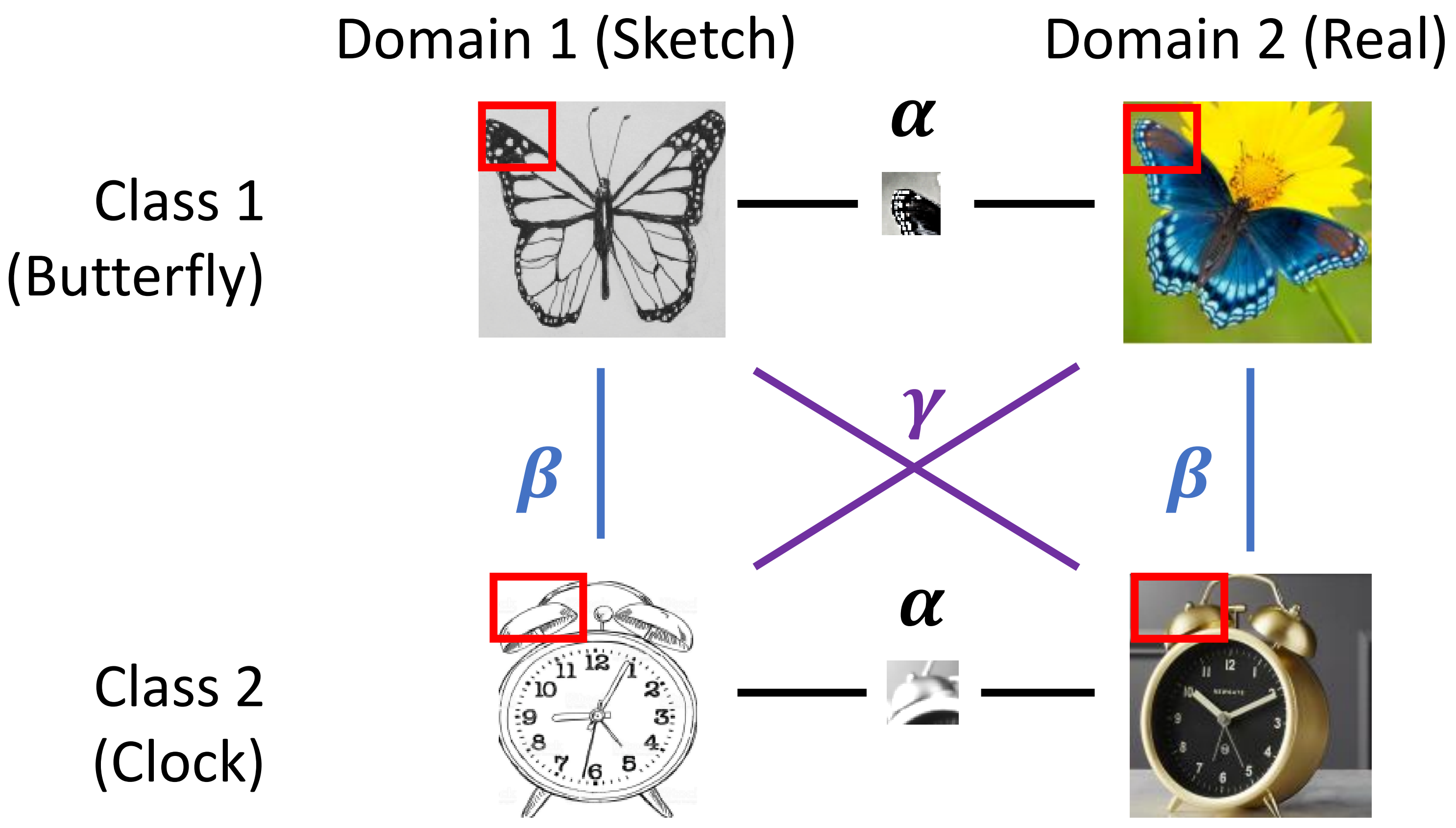


α: probability that augmentations of images coincide

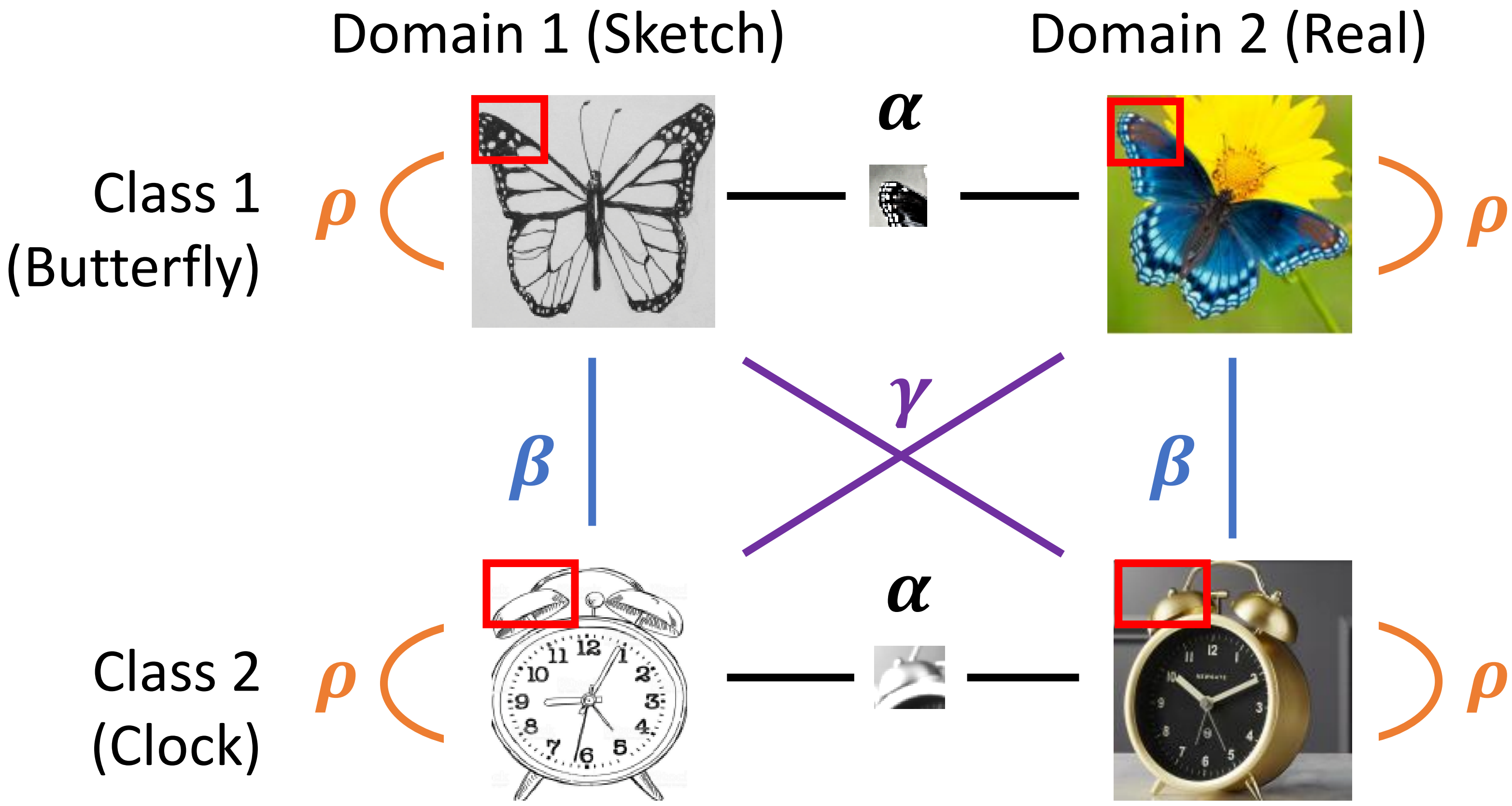
Setup: augmentation graph



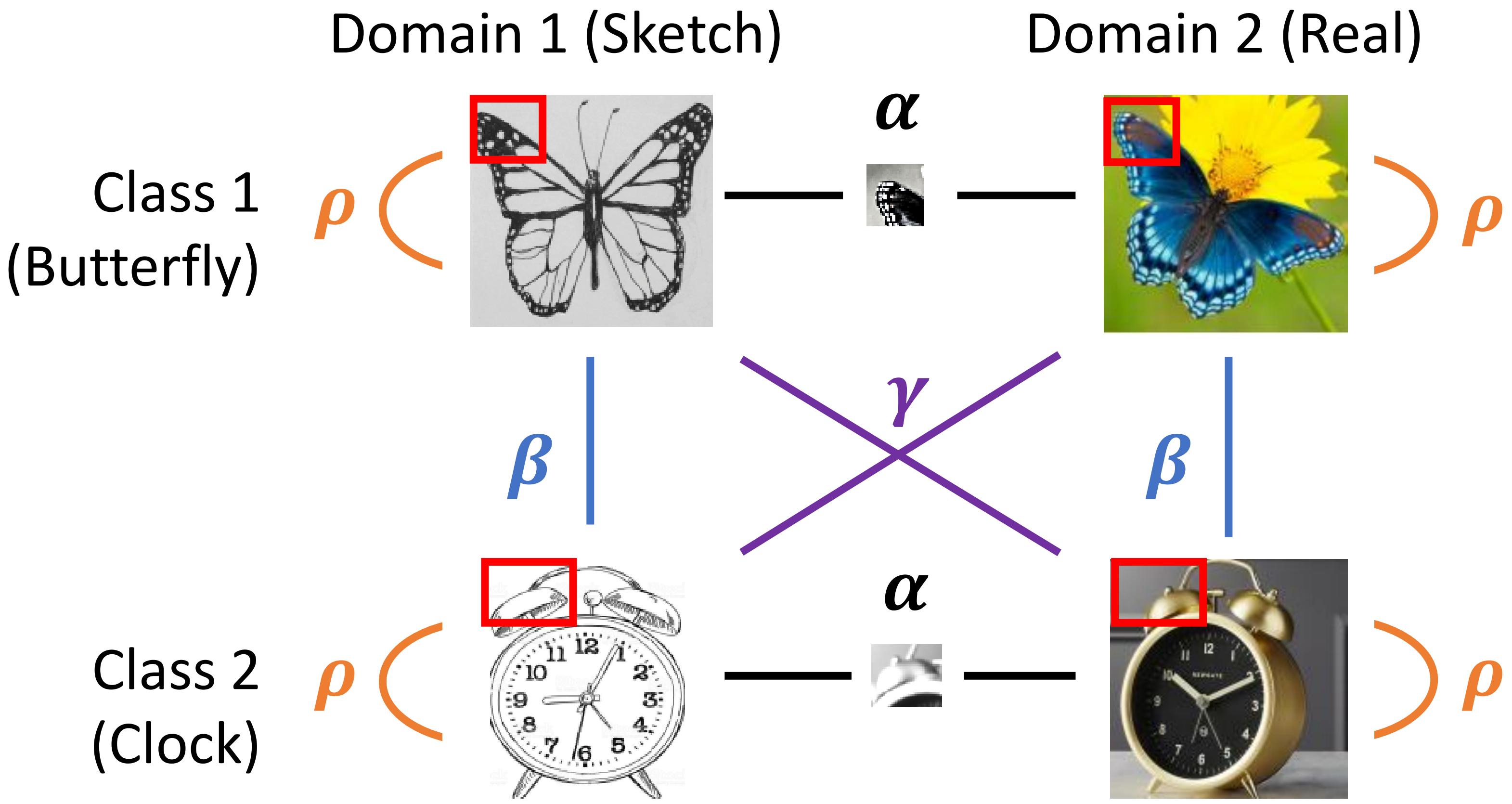
Setup: augmentation graph



Setup: augmentation graph

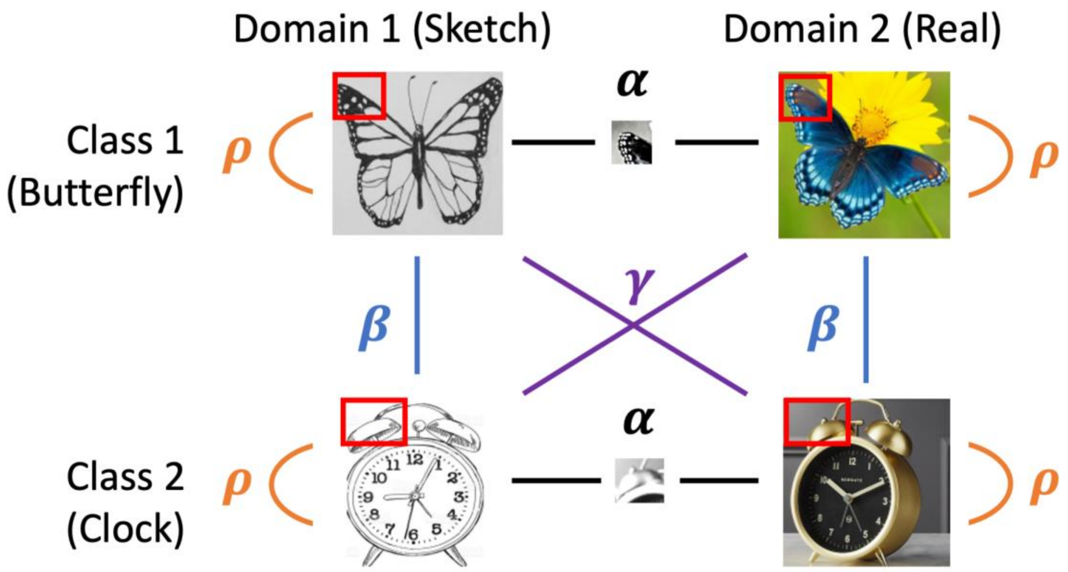


Setup: augmentation graph



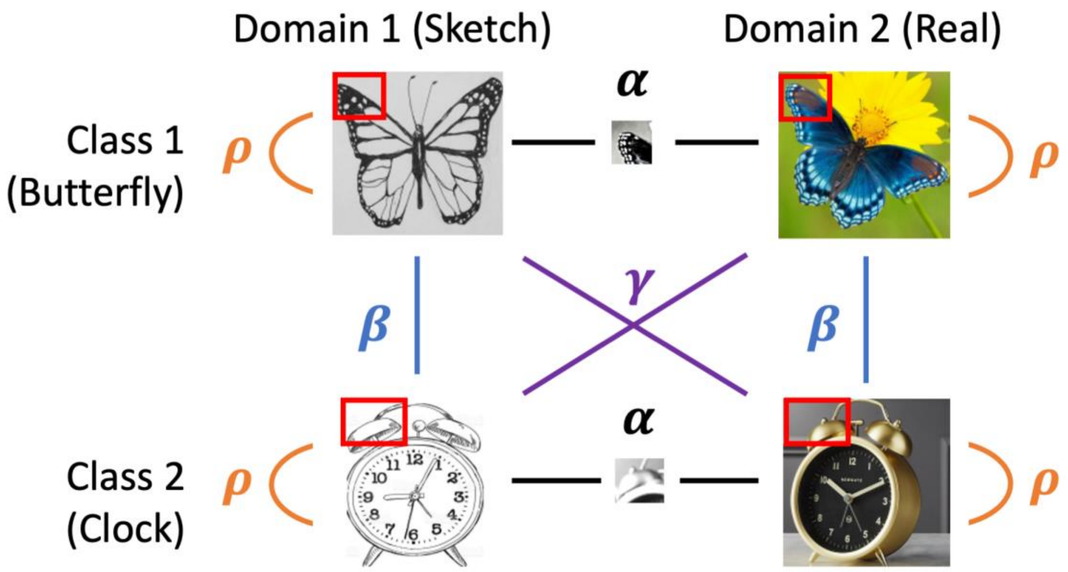
Magnitudes of connectivity parameters ρ, α, β, and $\gamma \approx$ similarity of augmentations

Setup: augmentation graph



Can express augmentation graph using adjacency matrix A

Setup: augmentation graph



	Sketch clock	Sketch butterfly	Real clock	Real butterfly
Sketch clock	ρ	β	α	γ
Sketch butterfly	β	ρ	γ	α
Real clock	α	γ	ρ	β
Real butterfly	γ	α	β	ρ

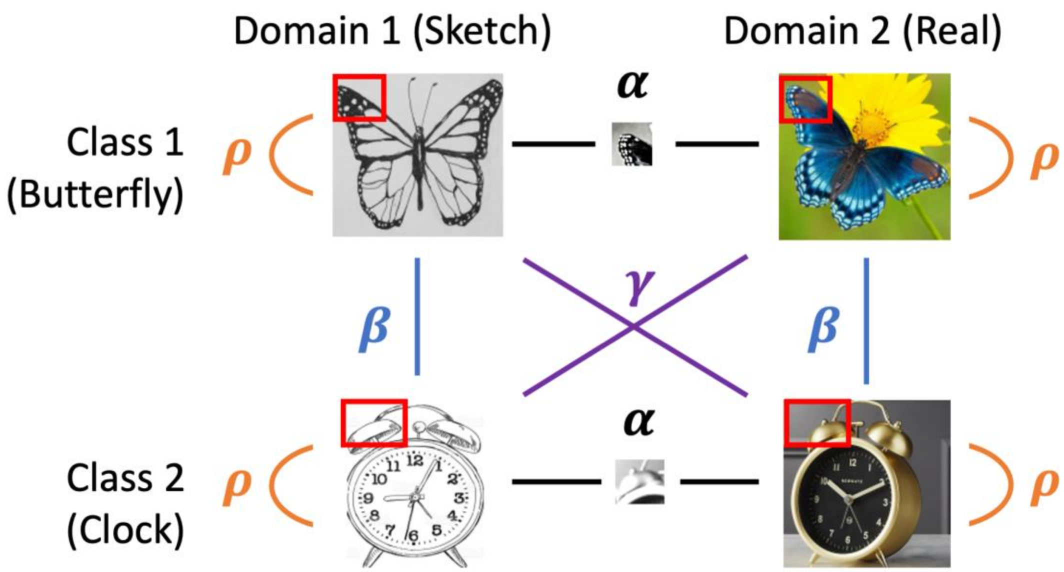
Can express augmentation graph using adjacency matrix A

Outline

- Setup: augmentation graph
- Intuitions and theoretical results
  - Main intuitions (toy example)
  - Results for stochastic block model & beyond
  - Contrastive pre-training vs. ERM & DANN
- Test theoretical predictions on real data

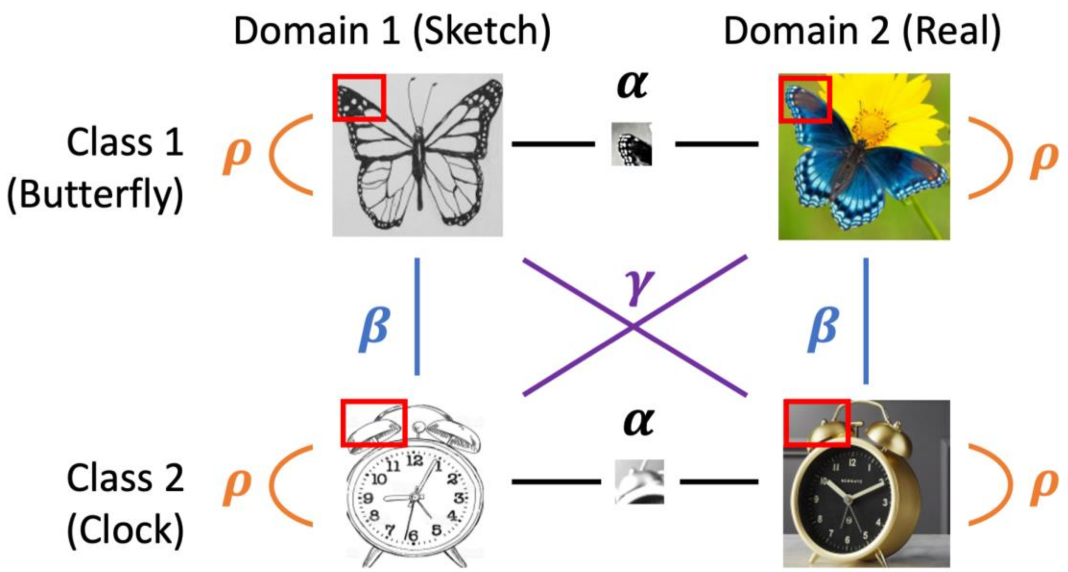
Intuitions & toy example

- Binary classification, 1 example per class and domain (4 examples total)
- Let $\hat{F}: R^{4\times3}$ be a matrix whose rows contain learned features



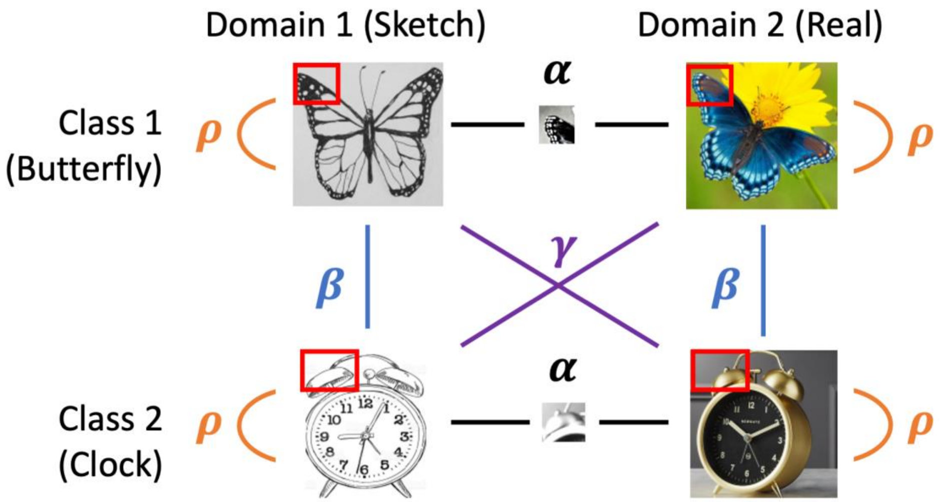
Intuitions & toy example

- Binary classification, 1 example per class and domain (4 examples total)
- Let $\hat{F}: R^{4\times 3}$ be a matrix whose rows contain learned features



$$\hat{F} = \begin{bmatrix} \longleftarrow & \hat{f}\,(\text{sketch clock}) & \longrightarrow \\ \longleftarrow & \hat{f}\,(\text{sketch butterfly}) & \longrightarrow \\ \longleftarrow & \hat{f}\,(\text{real clock}) & \longrightarrow \\ \longleftarrow & \hat{f}\,(\text{real butterfly}) & \longrightarrow \end{bmatrix}$$

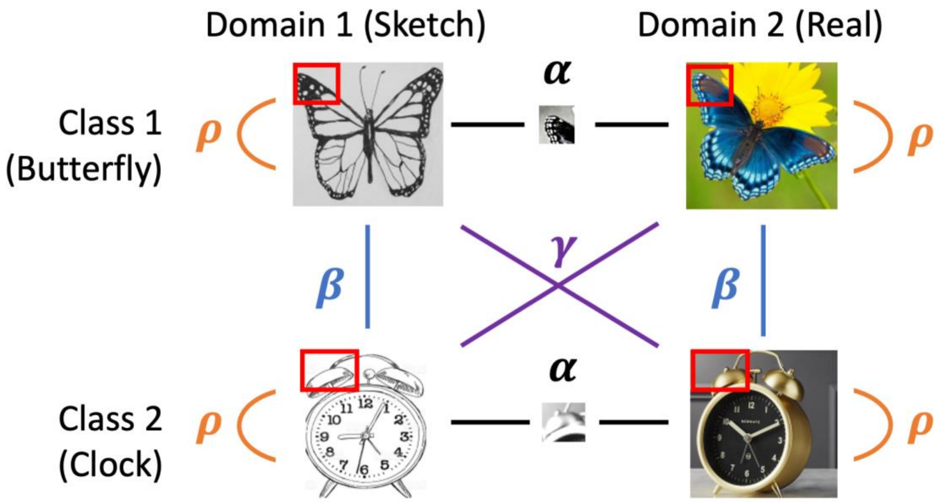
Intuitions & toy example



Augmentation graph

Intuitions & toy example

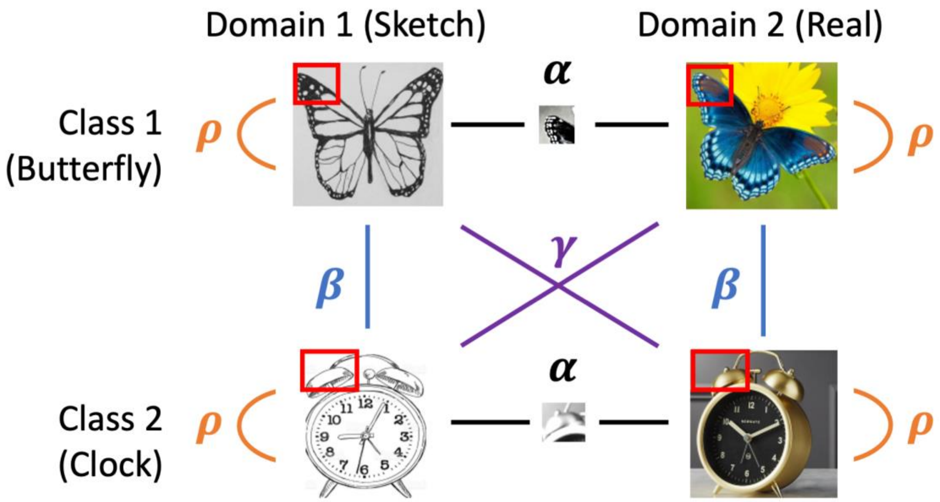
If $\mathbf{min}(\boldsymbol{\alpha}, \boldsymbol{\beta}) > \boldsymbol{\gamma}$ (and self-loop ρ is the largest):



Augmentation graph

Intuitions & toy example

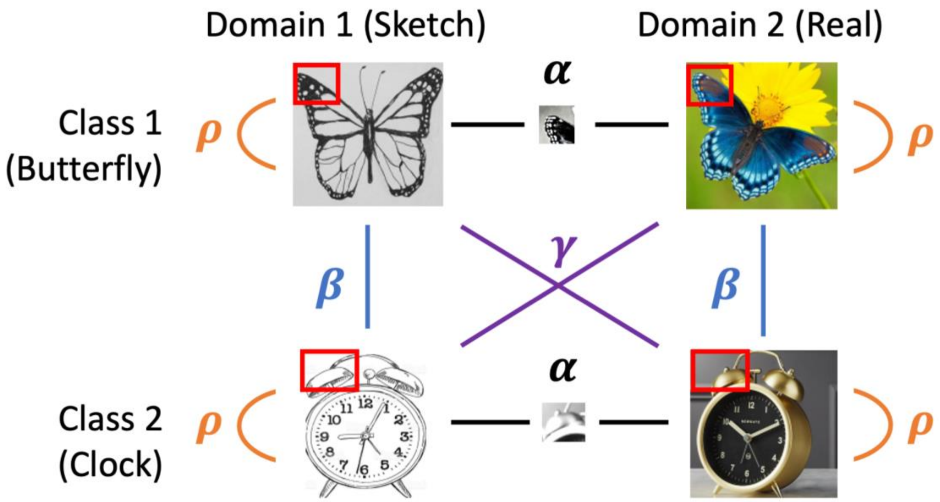
If $\mathbf{min}(\boldsymbol{\alpha}, \boldsymbol{\beta}) > \boldsymbol{\gamma}$ (and self-loop ρ is the largest):



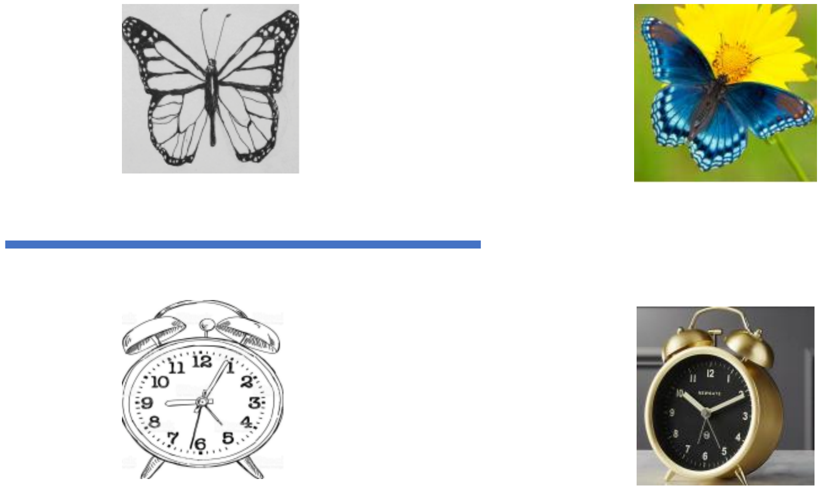
Augmentation graph

Learned representation space

Intuitions & toy example

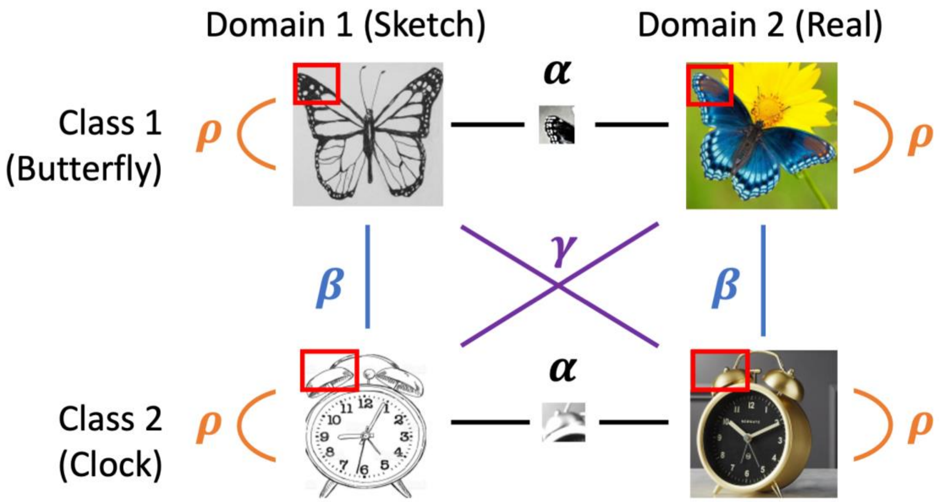


If $\mathbf{min}(\boldsymbol{\alpha}, \boldsymbol{\beta}) > \boldsymbol{\gamma}$ (and self-loop ρ is the largest):

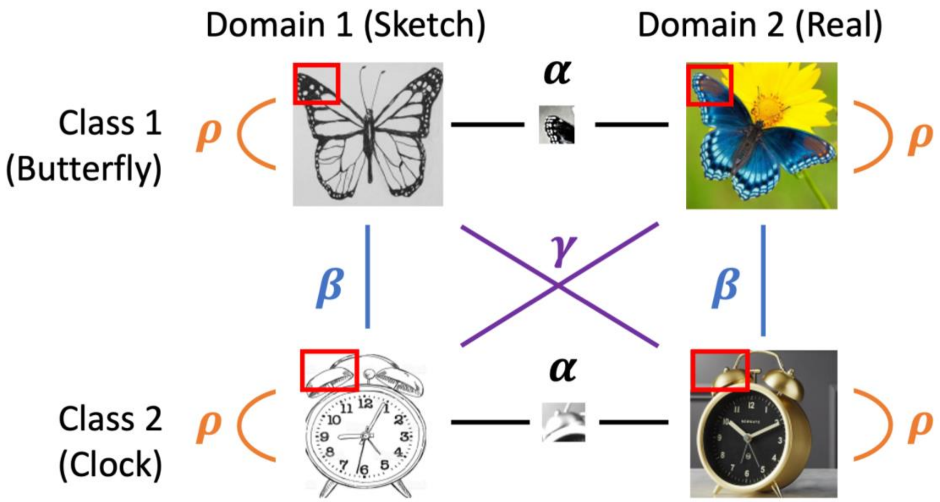


Intuitions & toy example

If $\mathbf{min}(\boldsymbol{\alpha}, \boldsymbol{\beta}) > \boldsymbol{\gamma}$ (and self-loop ρ is the largest):



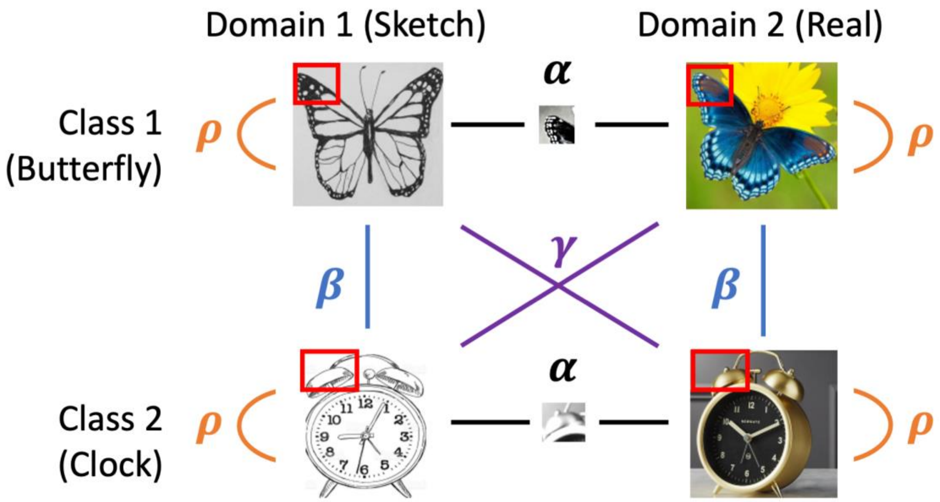
Intuitions & toy example



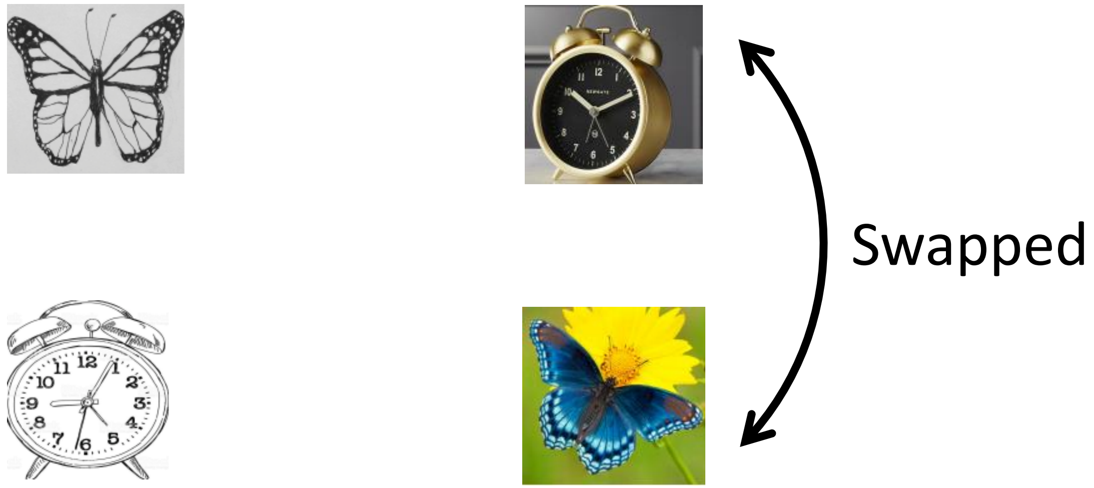
If $\mathbf{min}(\boldsymbol{\alpha}, \boldsymbol{\beta}) > \boldsymbol{\gamma}$ (and self-loop ρ is the largest):

Key condition for transfer: augmentations are more likely to change **only domain** (α) or **only class** (β) than **both domain and class** (γ)

Intuitions & toy example

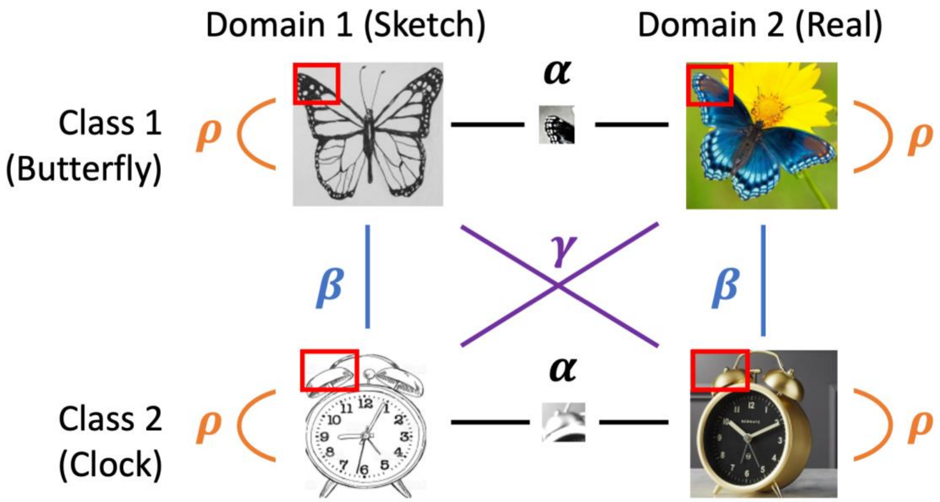


If instead $\alpha < \gamma$:



Intuitions & toy example

If instead $\alpha < \gamma$:



If the condition is violated, the target features can be "swapped" so that a source-trained linear classifier fails to generalize

Generalization beyond simple example

- Consider stochastic block model (SBM): extends to multiple domains, multiple classes, and multiple examples per class/domain

Generalization beyond simple example

- Consider stochastic block model (SBM): extends to multiple domains, multiple classes, and multiple examples per class/domain
- We prove: **same conditions** ($\min(\alpha, \beta) > \gamma$ and ρ is largest) allow contrastive pre-training to learn linearly transferable features (with easily separable source and target features)

Generalization beyond simple example

- Consider stochastic block model (SBM): extends to multiple domains, multiple classes, and multiple examples per class/domain
- We prove: **same conditions** ($\min(\alpha, \beta) > \gamma$ and ρ is largest) allow contrastive pre-training to learn linearly transferable features (with easily separable source and target features)
- Follow-up work generalizes beyond random graph models, with asymmetry: HaoChen et al. 2022

Outline

- Setup: augmentation graph
- Intuitions and theoretical results
  - Main intuitions (toy example)
  - Results for stochastic block model & beyond
  - Contrastive pre-training vs. ERM & DANN
- Test theoretical predictions on real data

Connectivity predicts target accuracy

- Our theory predicts that target accuracy depends on α, β, γ and requires that $\alpha > \gamma$ and $\beta > \gamma$

Connectivity predicts target accuracy

- Our theory predicts that target accuracy depends on α, β, γ and requires that $\alpha > \gamma$ and $\beta > \gamma$
- Estimate α, β, γ by training a classifier to predict between augmented images of different domains/classes, evaluate on held out examples

Connectivity predicts target accuracy

- Our theory predicts that target accuracy depends on α, β, γ and requires that $\alpha > \gamma$ and $\beta > \gamma$
- Estimate α, β, γ by training a classifier to predict between augmented images of different domains/classes, evaluate on held out examples

$$\text{target accuracy} \approx (\alpha/\gamma)^{w_1} \cdot (\beta/\gamma)^{w_2}$$

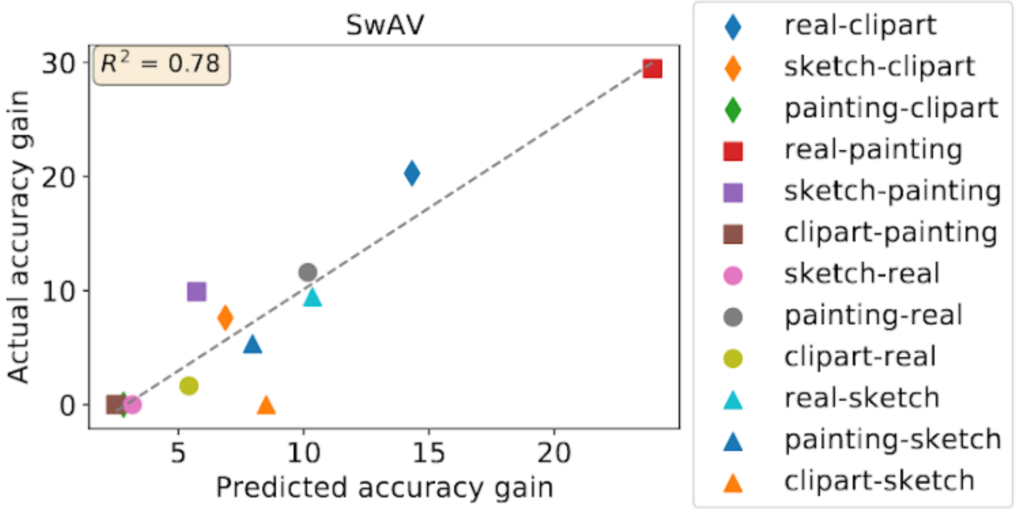
Connectivity predicts target accuracy

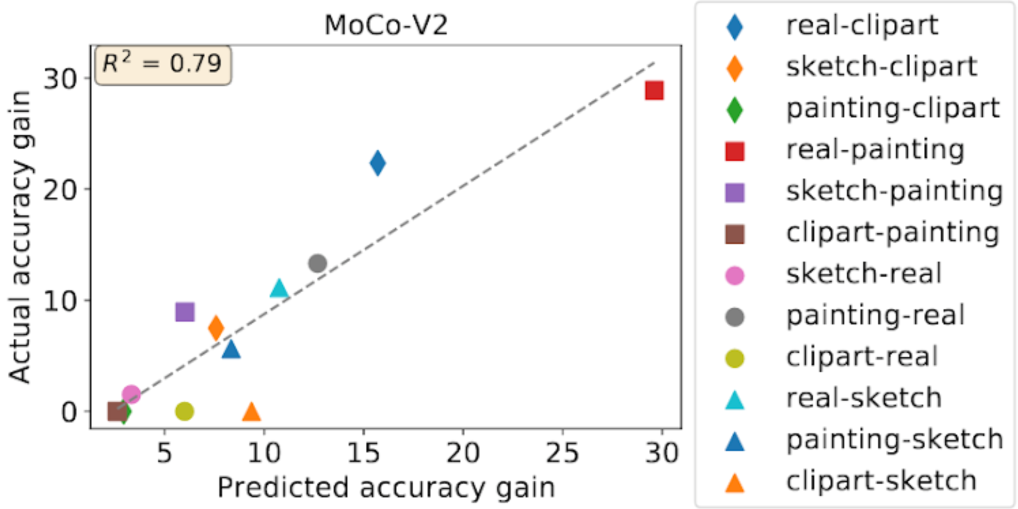
- Our theory predicts that target accuracy depends on α, β, γ and requires that $\alpha > \gamma$ and $\beta > \gamma$
- Estimate α, β, γ by training a classifier to predict between augmented images of different domains/classes, evaluate on held out examples

$$\text{target accuracy} \approx (\alpha/\gamma)^{w_1} \cdot (\beta/\gamma)^{w_2}$$

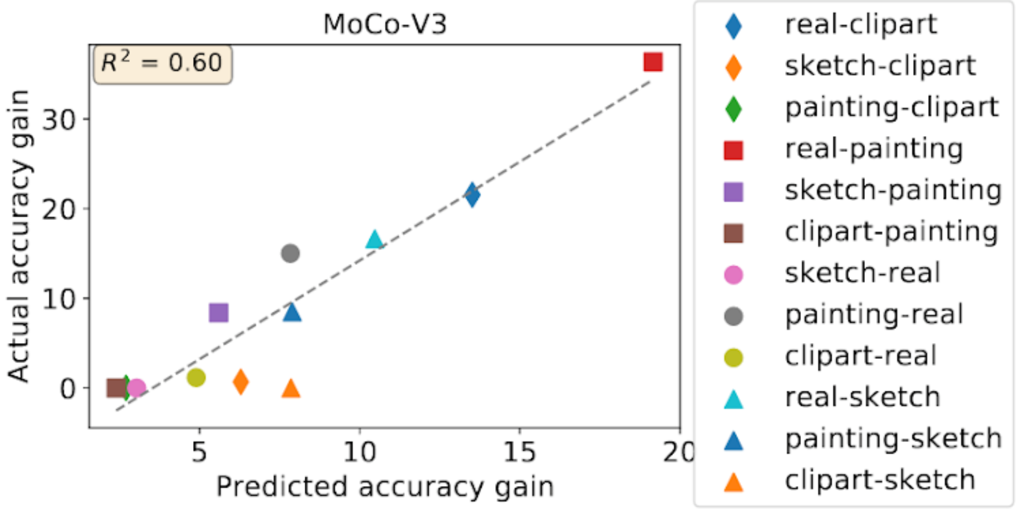
- Estimate w_1, w_2 by fitting a linear function in log space and determine quality of fit compared to a control

Predicting target accuracy (contrastive methods)

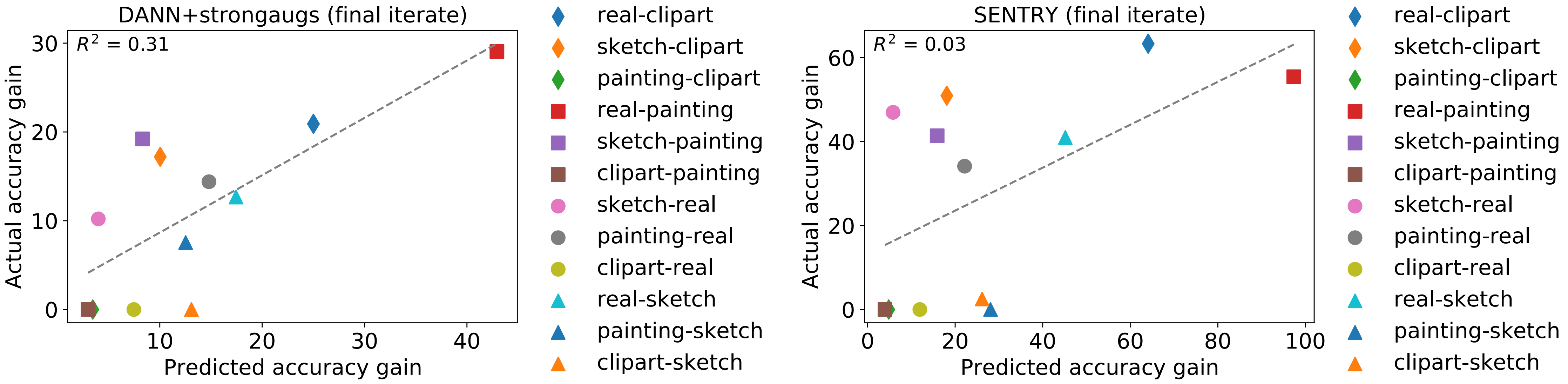




Method	R^2
SwAV	0.78
MoCo-V2	0.79
MoCo-V3	0.60

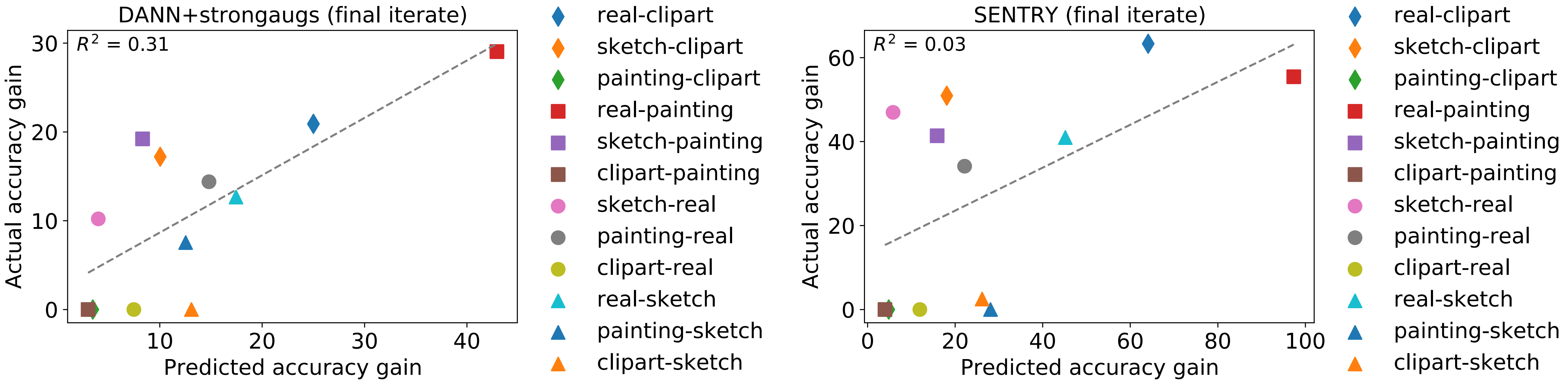


Predicting target accuracy (controls)



Method	R^2
SwAV	0.78
MoCo-V2	0.79
MoCo-V3	0.60

Predicting target accuracy (controls)



Method	R^2
SwAV	0.78
MoCo-V2	0.79
MoCo-V3	0.60
DANN	0.31
SENTRY	0.03

Lower quality of fit for non-contrastive methods: DANN and SENTRY

Class and domain are disentangled

- We train a linear probe for class and domain information in the contrastive features, finding that class and domain classifiers have low cosine similarity

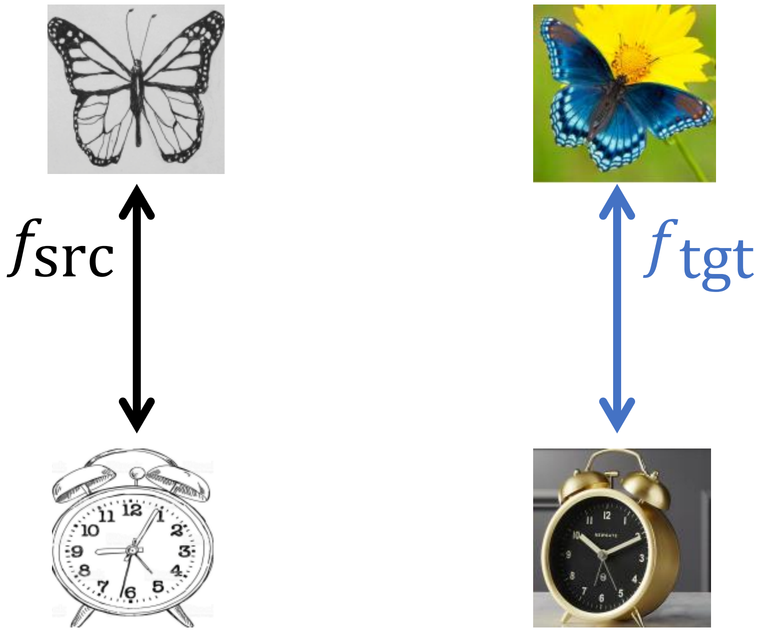
Class and domain are disentangled

- We train a linear probe for class and domain information in the contrastive features, finding that class and domain classifiers have low cosine similarity


	f_{src} vs. f_{tgt}	f_{src} vs. f_{dom}	f_{tgt} vs. f_{dom}
Living-17	0.397	0.013	0.016
DomainNet	0.187	0.018	0.018


Class and domain are disentangled

- We train a linear probe for class and domain information in the contrastive features, finding that class and domain classifiers have low cosine similarity

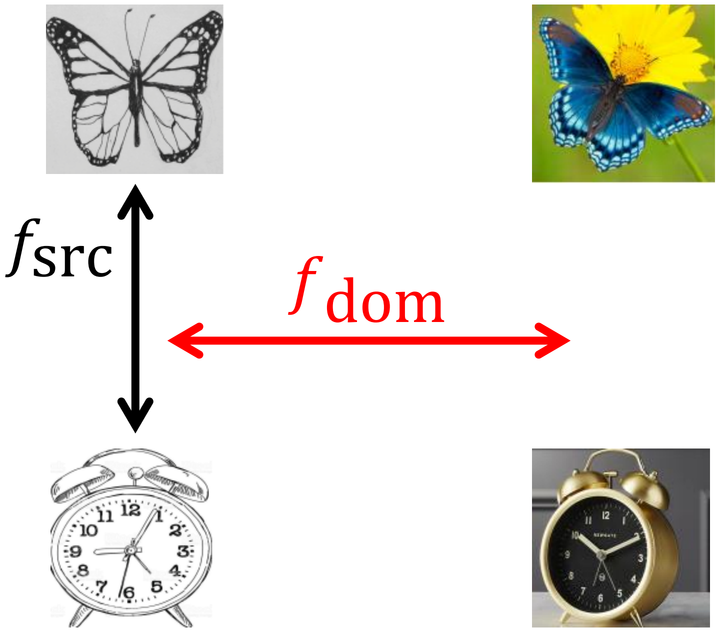


	f_{src} vs. f_{tgt}	f_{src} vs. f_{dom}	f_{tgt} vs. f_{dom}
Living-17	0.397	0.013	0.016
DomainNet	0.187	0.018	0.018

Aligned

Class and domain are disentangled

- We train a linear probe for class and domain information in the contrastive features, finding that class and domain classifiers have low cosine similarity

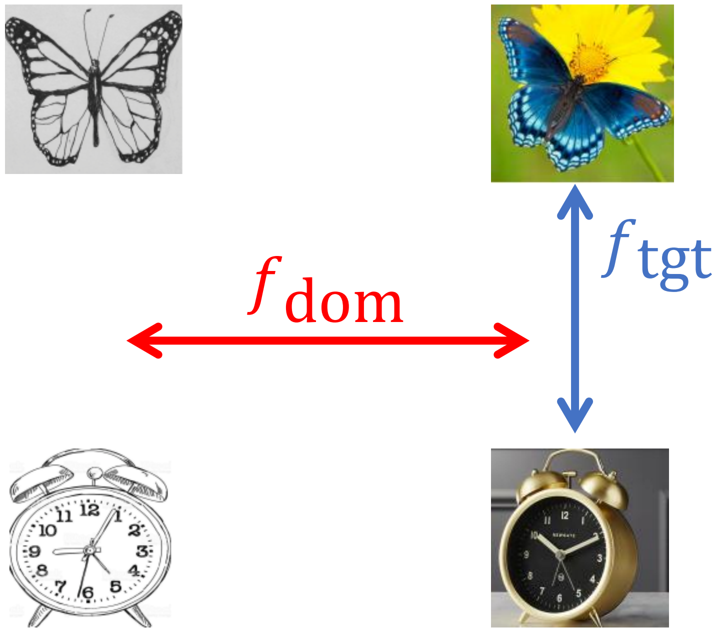


	f_{src} vs. f_{tgt}	f_{src} vs. f_{dom}	f_{tgt} vs. f_{dom}
Living-17	0.397	0.013	0.016
DomainNet	0.187	0.018	0.018

Orthogonal

Class and domain are disentangled

- We train a linear probe for class and domain information in the contrastive features, finding that class and domain classifiers have low cosine similarity



	f_{src} vs. f_{tgt}	f_{src} vs. f_{dom}	f_{tgt} vs. f_{dom}
Living-17	0.397	0.013	0.016
DomainNet	0.187	0.018	0.018

Orthogonal

Target Unlabeled Data is Important

- Access to target unlabeled examples is important for robustness (pretraining on source examples alone does not lead to robustness gains)

Target Unlabeled Data is Important

- Access to target unlabeled examples is important for robustness (pretraining on source examples alone does not lead to robustness gains)

	ERM	SwAV (S)	SwAV (T)	**SwAV (S+T)**
Living-17	63.29	62.71	70.41	**75.12**
Entity-30	52.52	52.33	60.33	**62.03**

Concluding Thoughts: Why Pretraining?

- Rich organization can pretrain once, everyone can fine-tune for many tasks cheaply

Concluding Thoughts: Why Pretraining?

- Rich organization can pretrain once, everyone can fine-tune for many tasks cheaply
- This approach gets SoTA on many robustness datasets: WILDS-FMoW, WILDS-iWildCam, ImageNet robustness, DomainNet

Concluding Thoughts: Why Pretraining?

- Rich organization can pretrain once, everyone can fine-tune for many tasks cheaply
- This approach gets SoTA on many robustness datasets: WILDS-FMoW, WILDS-iWildCam, ImageNet robustness, DomainNet
- Our paper: why does pretraining help? Is it just about having lots of data?

Conclusion

- Contrastive pre-training is a competitive method for UDA

Conclusion

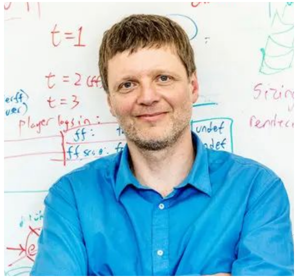
- Contrastive pre-training is a competitive method for UDA
- Works without collapsing source and target representations

Conclusion

- Contrastive pre-training is a competitive method for UDA
- Works without collapsing source and target representations
- Instead, disentangles class and domain information, enabling transfer
  - Consequence of the structure of connections between domains and classes via data augmentations

Subgroup Robustness Grows on Trees: An Empirical Baseline Investigation

IFDS Workshop on Distributional Robustness
Aug. 5, 2022



Josh Gardner
jpgard@cs.washington.edu

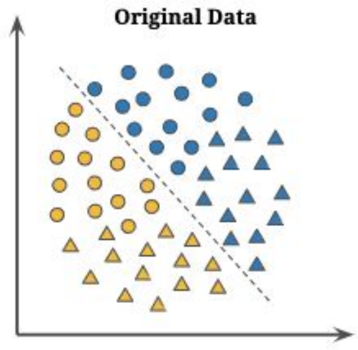
Zoran Popović
zoran@cs.washington.edu

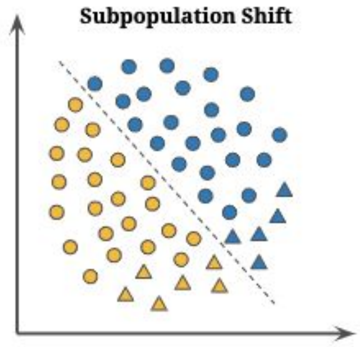
Ludwig Schmidt
schmidt@cs.washington.edu

Subgroup Robustness

Subgroup Robustness refers to the ability of a model to achieve good performance across discrete subgroups in a distribution.

This is an extreme version of **subpopulation shift** where we evaluate shift on target datasets entirely of a single (demographic) subpopulation.





With Great Progress Comes Great…Confusion?

Rapid Progress In Robust Learning

- Maximum Weighted Loss Discrepancy (Khani et al. 2019)
- DRO (e.g. Duchi and Namkoong 2018, Levy et al. 2020)
- Group DRO (Sagawa et al. 2020)
- DORO (Zhai et al. 2021)
- …many more!

With Great Progress Comes Great…Confusion?

Rapid Progress In Robust Learning

- Maximum Weighted Loss Discrepancy (Khani et al. 2019)
- DRO (e.g. Duchi and Namkoong 2018, Levy et al. 2020)
- Group DRO (Sagawa et al. 2020)
- DORO (Zhai et al. 2021)
- …many more!

Need for Reliable Evaluation

In other fields, large-scale empirical baseline evaluations have been critical to (re)assessing progress (Liao et al. 2021).

The use of unreliable statistical inference methods in particular has led to misleading signals of progress (Agarwal et al. 2021).

With Great Progress Comes Great…Confusion?

Rapid Progress In Robust Learning

- Maximum Weighted Loss Discrepancy (Khani et al. 2019)
- DRO (e.g. Duchi and Namkoong 2018, Levy et al. 2020)
- Group DRO (Sagawa et al. 2020)
- DORO (Zhai et al. 2021)
- …many more!

Need for Reliable Evaluation

In other fields, large-scale empirical baseline evaluations have been critical to (re)assessing progress (Liao et al. 2021).

The use of unreliable statistical inference methods in particular has led to misleading signals of progress (Agarwal et al. 2021).

What is the current SOTA for subgroup robustness in tabular data?

Outline

Introduction

Two Perspectives on Subgroup Robustness

Study Design + Datasets

Results

Accuracy-Robustness Frontiers

Evaluating Evaluation Metrics + Model Selection Effects

Hyperparameter Sensitivity

Conclusions

Implications for Practice + Future Work

Outline

Introduction

Two Perspectives on Subgroup Robustness

Study Design + Datasets

Results

Accuracy-Robustness Frontiers

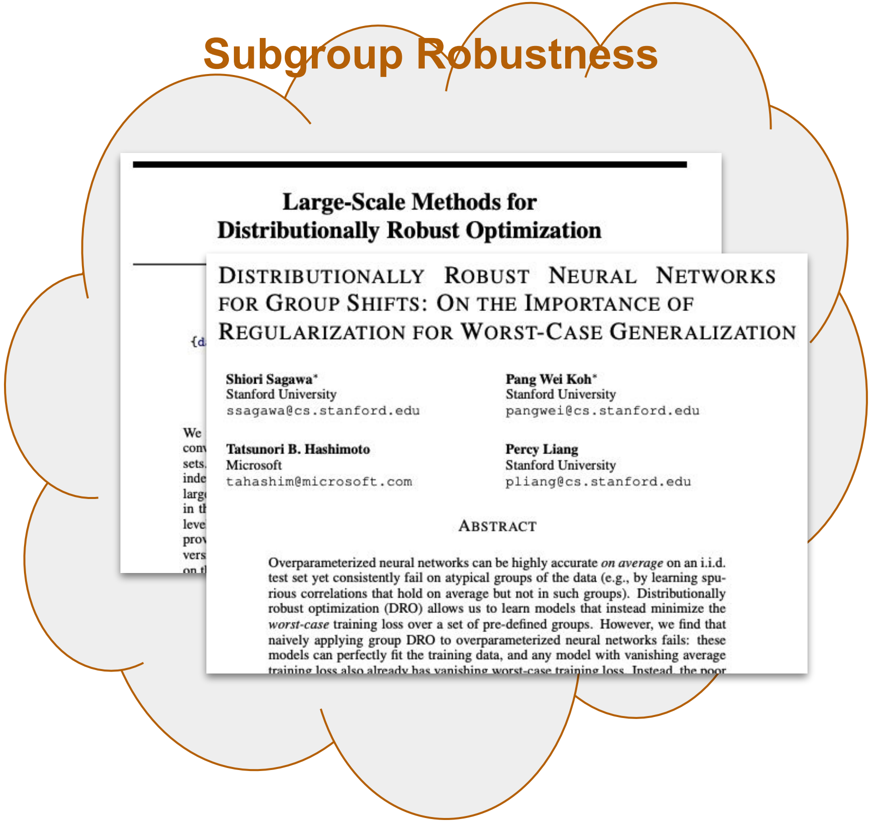
Evaluating Evaluation Metrics + Model Selection Effects

Hyperparameter Sensitivity

Conclusions

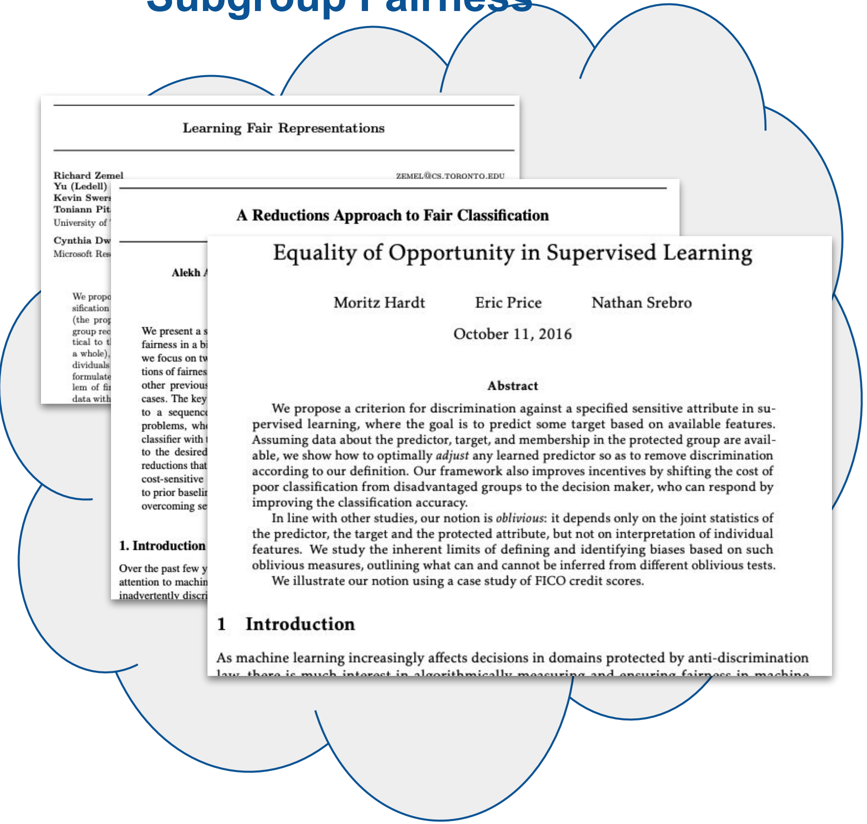
Implications for Practice + Future Work

Subgroup Robustness and Fairness

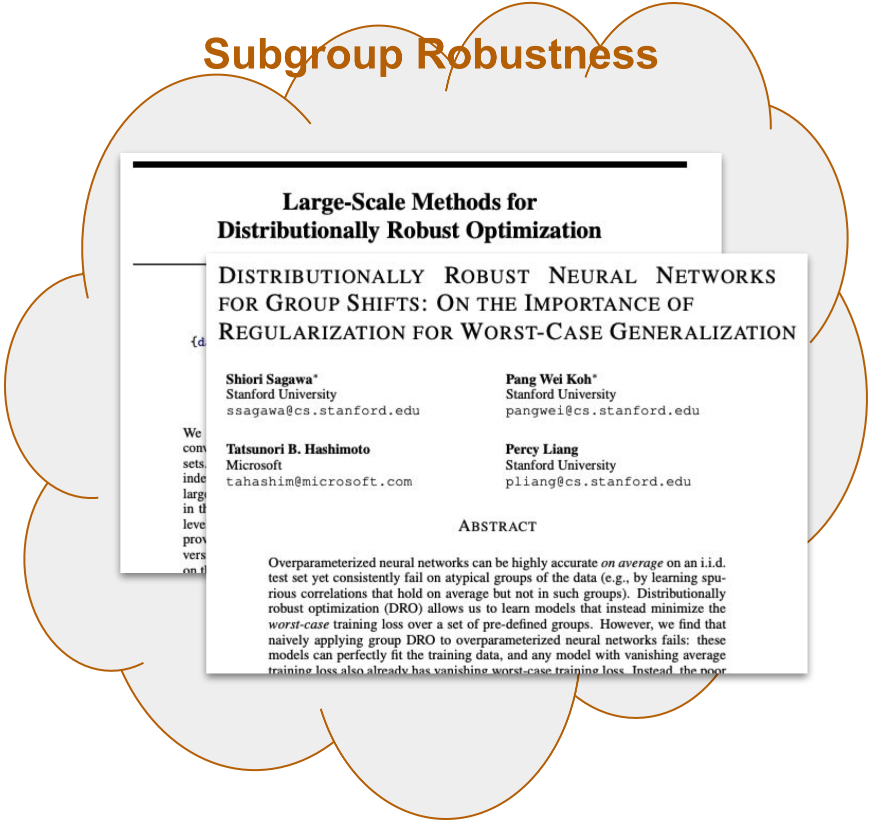


Subgroup Robustness and Fairness

Subgroup Fairness



Subgroup Robustness



Tabular Data: Deep Learning's "Unconquered Castle"

Widely used in practice but often a secondary focus in evaluating robust models

Often **directly encodes** sensitive subgroups of interest

Challenging to model and SOTA performance is achieved with **non-neural methods**

[Sagawa et al. 2019]

Toxic	Comment Text	Male	Female	LGBTQ	White	Black	...	Christian
0	I applaud your father. He was a good man! We need more like him.	1	0	0	0	0	...	0
0	As a Christian, I will not be patronizing any of those businesses.	0	0	0	0	0	...	1
0	What do Black and LGBT people have to do with bicycle licensing?	0	0	1	0	1	...	0
0	Government agencies track down foreign baddies and protect law-abiding white citizens. How many shows does that describe?	0	0	0	1	0	...	0
1	Maybe you should learn to write a coherent sentence so we can understand WTF your point is.	0	0	0	0	0	...	0

[Koh et al. 2020]

	HELOC		Adult		HIGGS		Covertype		Cal. Housing
	Acc ↑	AUC ↑	Acc ↑	AUC ↑	Acc ↑	AUC ↑	Acc ↑	AUC ↑	MSE ↓
Linear Model	73.0±0.0	80.1±0.1	82.5±0.2	85.4±0.2	64.1±0.0	68.4±0.0	72.4±0.0	92.8±0.0	0.528±0.008
KNN [65]	72.2±0.0	79.0±0.1	83.2±0.2	87.5±0.2	62.3±0.1	67.1±0.0	70.2±0.1	90.1±0.2	0.421±0.009
Decision Tree [197]	80.3±0.0	89.3±0.1	85.3±0.2	89.8±0.1	71.3±0.0	78.7±0.0	79.1±0.0	95.0±0.0	0.404±0.007
Random Forest [198]	82.1±0.2	90.0±0.2	86.1±0.2	91.7±0.2	71.9±0.0	79.7±0.0	78.1±0.1	96.1±0.0	0.272±0.006
XGBoost [53]	83.5±0.2	92.2±0.0	87.3±0.2	92.8±0.1	77.6±0.0	85.9±0.0	**97.3±0.0**	**99.9±0.0**	0.206±0.005
LightGBM [78]	83.5±0.1	92.3±0.0	**87.4±0.2**	**92.9±0.1**	77.1±0.0	85.5±0.0	93.5±0.0	99.7±0.0	**0.195±0.005**
CatBoost [79]	**83.6±0.3**	**92.4±0.1**	87.2±0.2	92.8±0.1	77.5±0.0	85.8±0.0	96.4±0.0	99.8±0.0	0.196±0.004
Model Trees [199]	82.6±0.2	91.5±0.0	85.0±0.2	90.4±0.1	69.8±0.0	76.7±0.0	-	-	0.385±0.019
MLP [200]	73.2±0.3	80.3±0.1	84.8±0.1	90.3±0.2	77.1±0.0	85.6±0.0	91.0±0.4	76.1±3.0	0.263±0.008
DeepFM [15]	73.6±0.2	80.4±0.1	86.1±0.2	91.7±0.1	76.9±0.0	83.4±0.0	-	-	0.260±0.006
DeepGBM [70]	78.0±0.4	84.1±0.1	84.6±0.3	90.8±0.1	74.5±0.0	83.0±0.0	-	-	0.856±0.065
RLN [72]	73.2±0.4	80.1±0.4	81.0±1.6	75.9±8.2	71.8±0.2	79.4±0.2	77.2±1.5	92.0±0.9	0.348±0.013
TabNet [5]	81.0±0.1	90.0±0.1	85.4±0.2	91.1±0.1	76.5±1.3	84.9±1.4	93.1±0.2	99.4±0.0	0.346±0.007
VIME [88]	72.7±0.0	79.2±0.0	84.8±0.2	90.5±0.2	76.9±0.2	85.5±0.1	90.9±0.1	82.9±0.7	0.275±0.007
TabTransformer [98]	73.3±0.1	80.1±0.2	85.2±0.2	90.6±0.2	73.8±0.0	81.9±0.0	76.5±0.3	72.9±2.3	0.451±0.014
NODE [6]	79.8±0.2	87.5±0.2	85.6±0.3	91.1±0.2	76.9±0.1	85.4±0.1	89.9±0.1	98.7±0.0	0.276±0.005
Net-DNF [57]	82.6±0.4	91.5±0.2	85.7±0.2	91.3±0.1	76.6±0.1	85.1±0.1	94.2±0.1	99.1±0.0	-
STG [201]	73.1±0.1	80.0±0.1	85.4±0.1	90.9±0.1	73.9±0.1	81.9±0.1	81.8±0.3	96.2±0.0	0.285±0.006
NAM [202]	73.3±0.1	80.7±0.3	83.4±0.1	86.6±0.1	53.9±0.6	55.0±1.2	-	-	0.725±0.022
SAINT [9]	82.1±0.3	90.7±0.2	86.1±0.3	91.6±0.2	**79.8±0.0**	**88.3±0.0**	96.3±0.1	99.8±0.0	0.226±0.004

Current tabular SOTA

Model/baseline for most robustness experiments

[Borisov et al. 2022]

Models

Robustness Methods	Fairness Methods	Tabular Tree-Based	Supervised Baselines
DORO (Chi^2, CVar) DRO (Chi^2, CVar) MWLD Group DRO	LFR Inprocessing (ExpGrad) Postprocessing	XGBoost LightGBM GBM Random Forest	L2 Log. Reg. SVM MLP

X

Hyperparameter/Architecture Grid Search

X

Datasets (incl. 2 sensitive attributes)

→ **317k total training iterations**

Datasets

Dataset	**Label**	**Sens.**	***n***	***d***	***Smallest Test Subgroup***
ACS Income*	High/Low Income	Race, Sex	499,350	20	18,134
ACS PubCov*	Public Ins.	Race, Sex	379,430	19	14,689
BRFSS*	Diabetes	Race, Sex	175,745	28	1,133
LARC	At-Risk (Grade)	URM Status, Sex	169,032	26	8,377
Adult	High/Low Income	Race, Sex	48,845	14	518
COMPAS	Recidivism	Race, Sex	7,215	10	57
Comm. & Crime	Elevated Crime	Income Lvl, Race	1,994	113	36
German Credit	Credit Risk	Age, Sex	1,000	22	11

Datasets

Dataset	**Label**	**Sens.**	***n***	***d***	***Smallest Test Subgroup***
ACS Income*	High/Low Income	Race, Sex	499,350	20	18,134
ACS PubCov*	Public Ins.	Race, Sex	379,430	19	14,689
BRFSS*	Diabetes	Race, Sex	175,745	28	1,133
LARC	At-Risk (Grade)	URM Status, Sex	169,032	26	8,377
Adult	High/Low Income	Race, Sex	48,845	14	518
COMPAS	Recidivism	Race, Sex	7,215	10	57
Comm. & Crime	Elevated Crime	Income Lvl, Race	1,994	113	36
German Credit	Credit Risk	Age, Sex	1,000	22	11

Outline

Introduction

- Two Perspectives on Subgroup Robustness
- Study Design + Datasets

Results

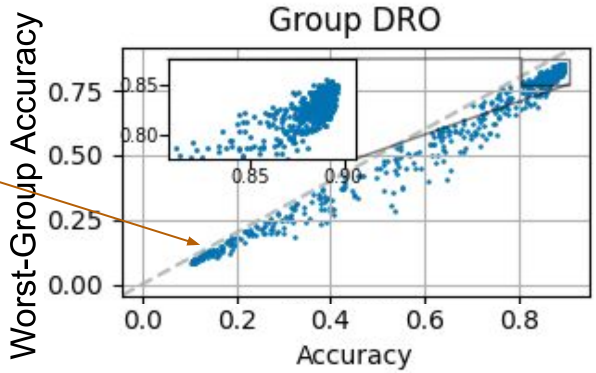
- Accuracy-Robustness Frontiers
- Evaluating Evaluation Metrics + Model Selection Effects
- Hyperparameter Sensitivity

Conclusions

- Implications for Practice + Future Work

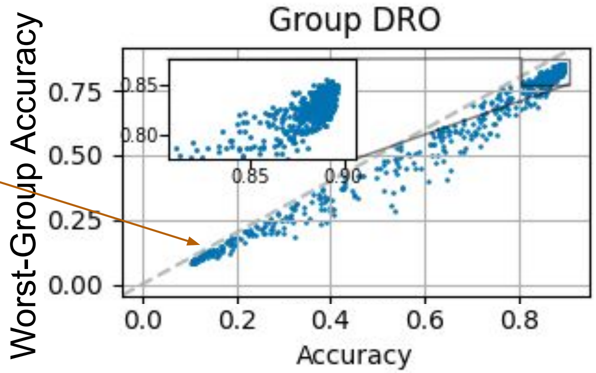
Example: Experiment Results (Group DRO, BRFSS)

each point = 1 hyperparameter configuration (1,620 total shown)

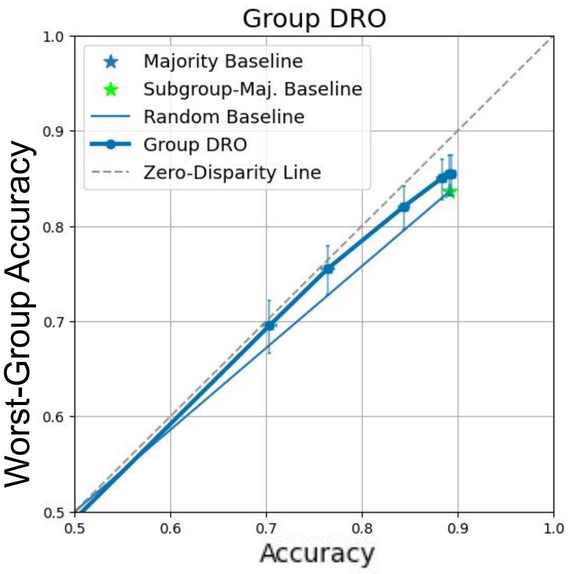


Example: Experiment Results (Group DRO, BRFSS)

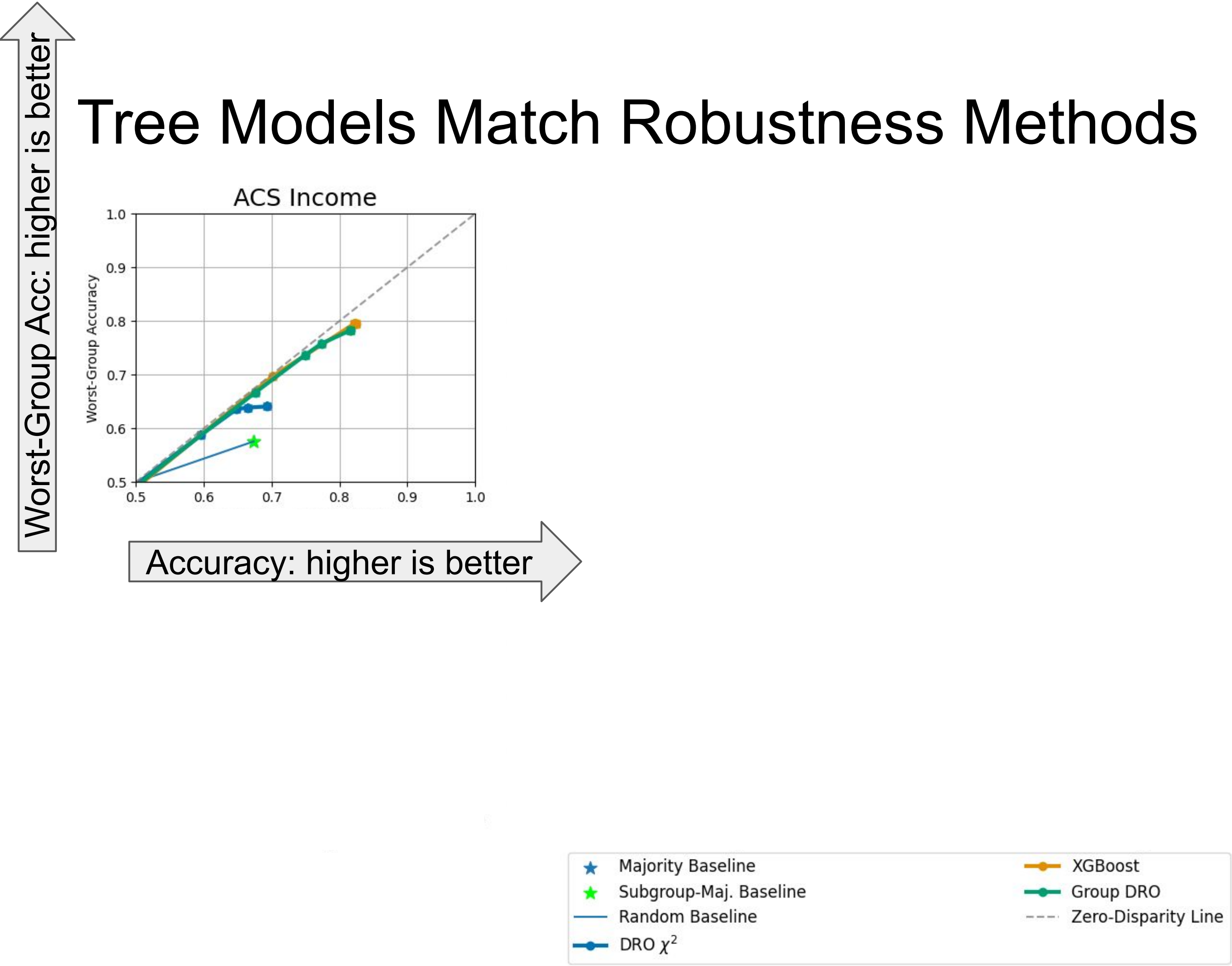
each point = 1 hyperparameter configuration (1,620 total shown)



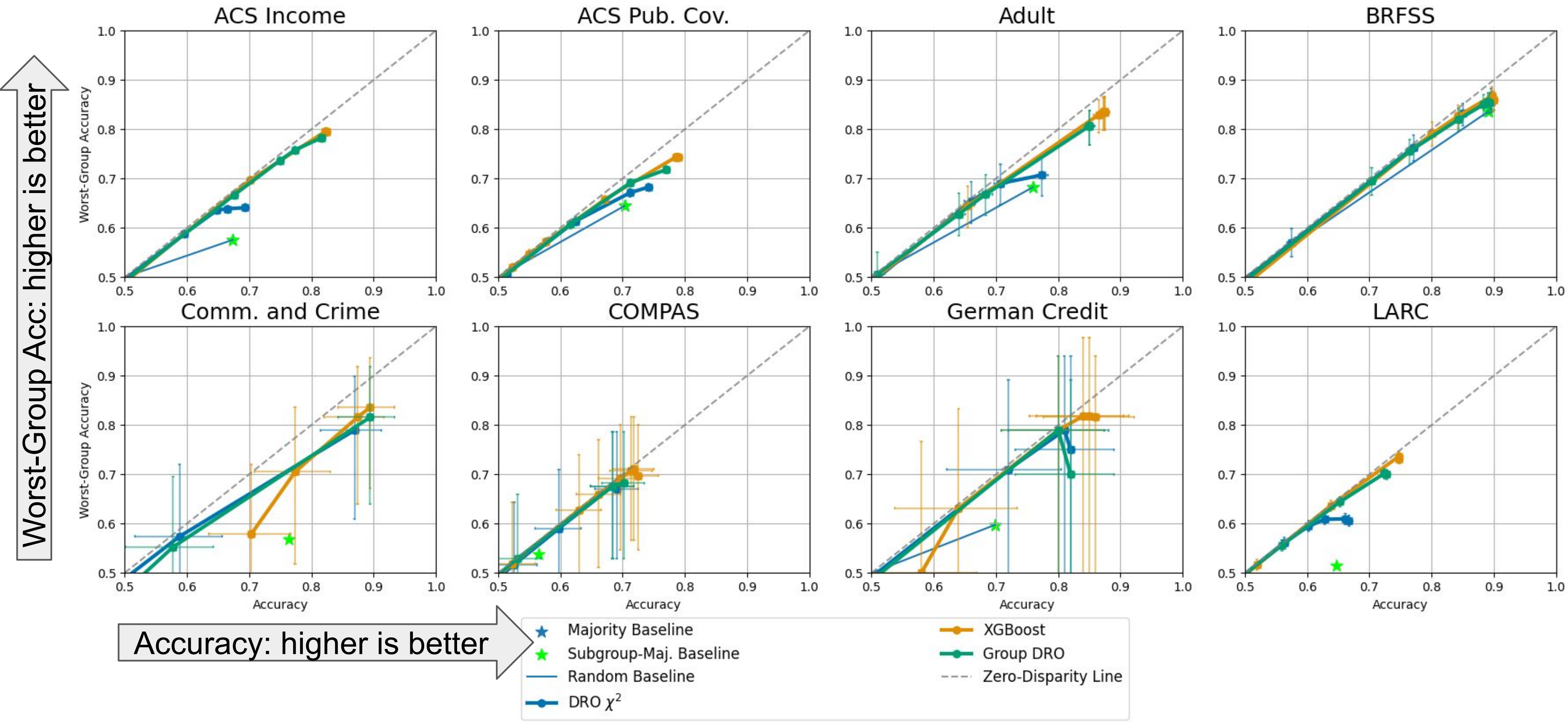
Trace Convex Hull



Tree Models Match Robustness Methods

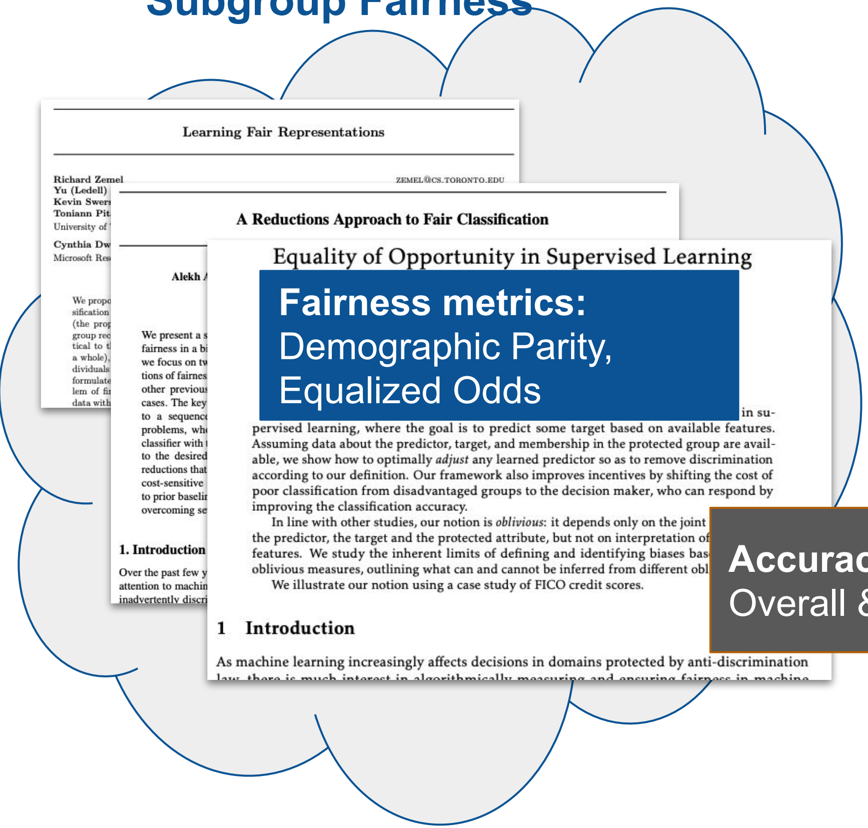


Tree Models Match Robustness Methods



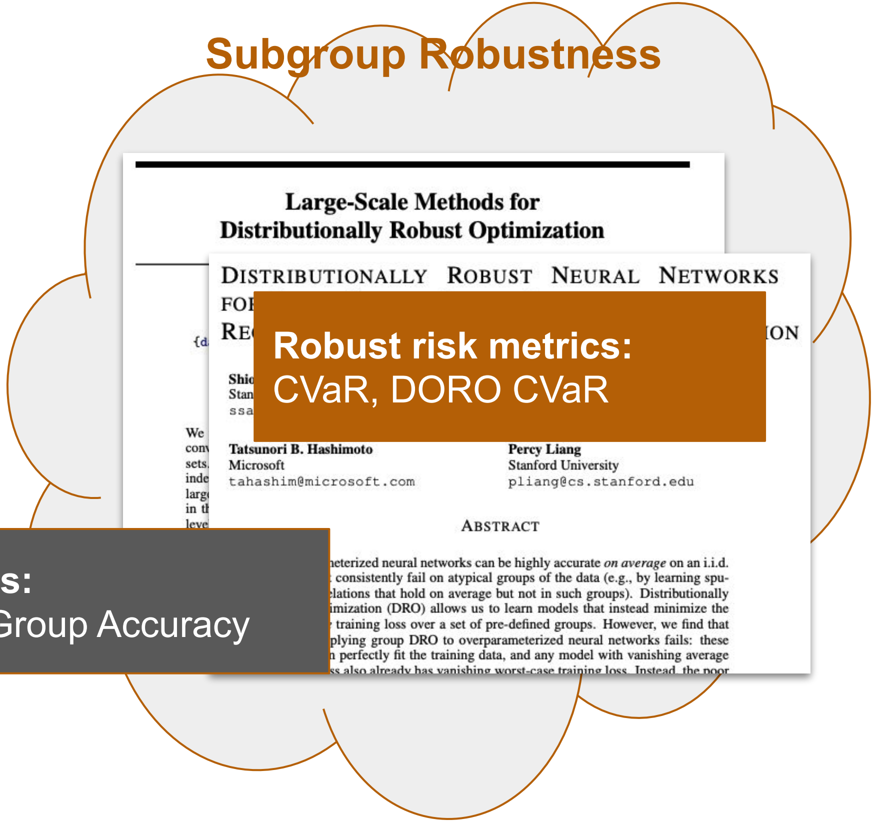
Metrics: Does what we measure matter?

Subgroup Fairness



Fairness metrics:
Demographic Parity,
Equalized Odds

Subgroup Robustness

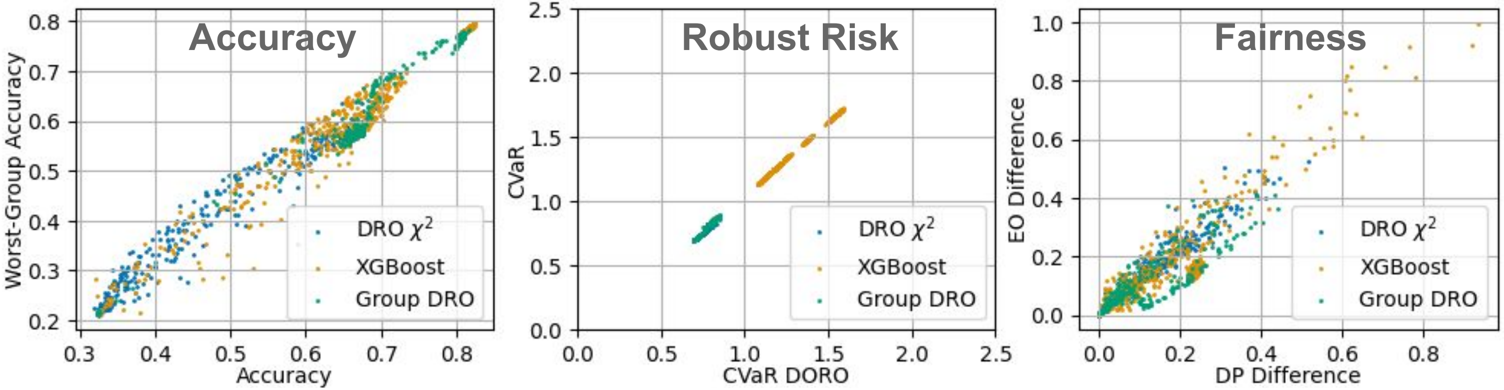


Robust risk metrics:
CVaR, DORO CVaR

Accuracy Metrics:
Overall & Worst-Group Accuracy

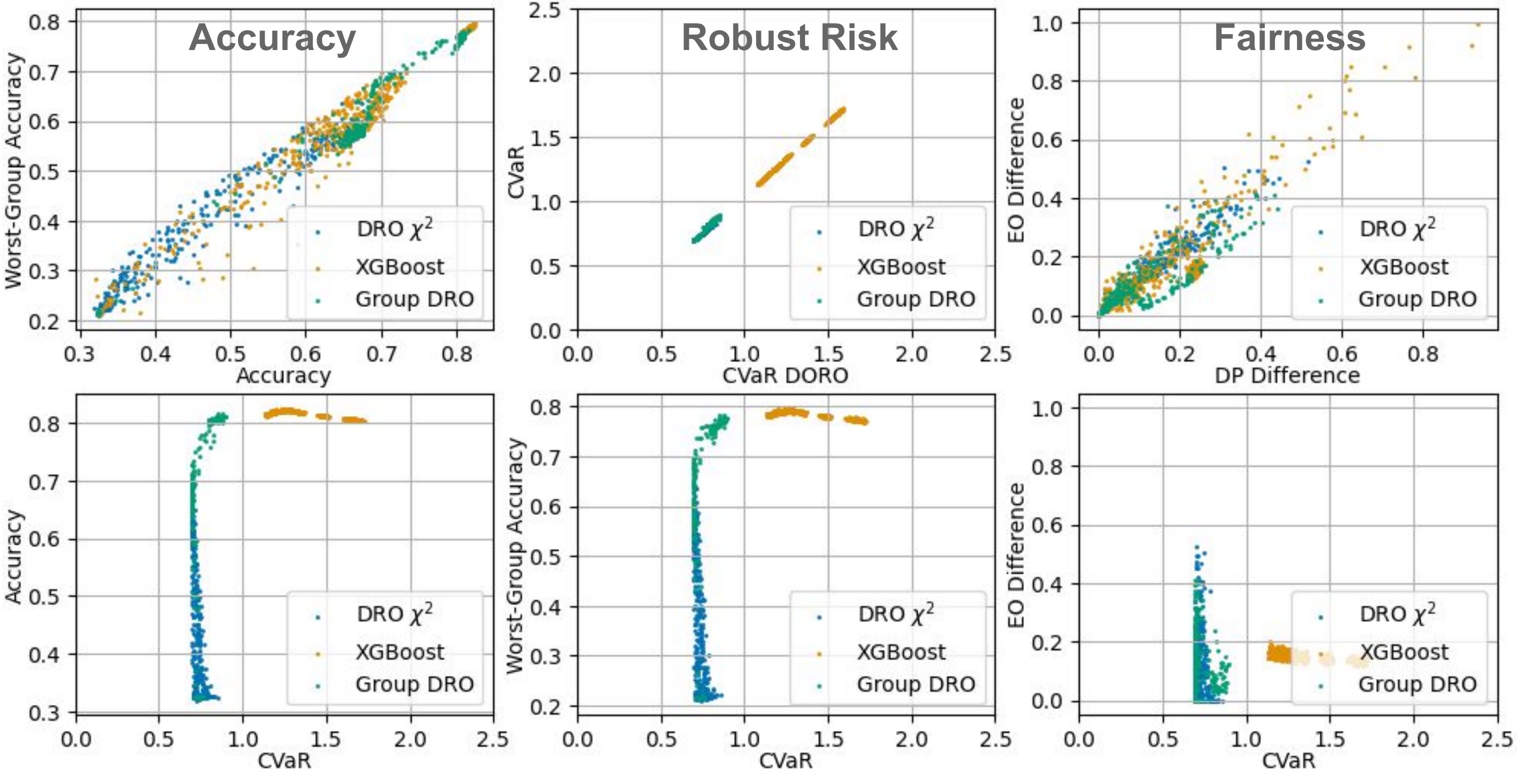
Model Performance Metrics: One Size Does Not Fit All

ACS Income Dataset

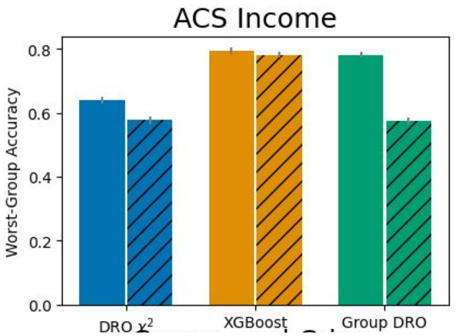


Model Performance Metrics: One Size Does Not Fit All

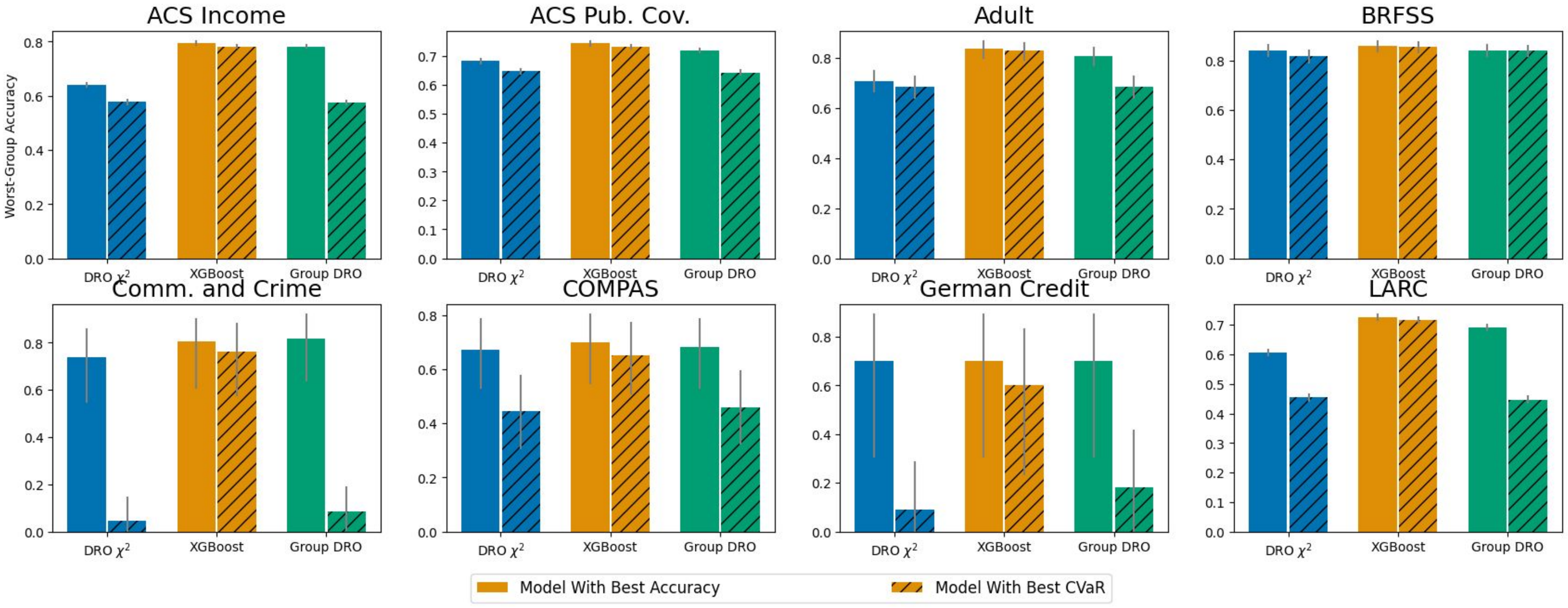
ACS Income Dataset



Trees are Robust to Model Selection Effects

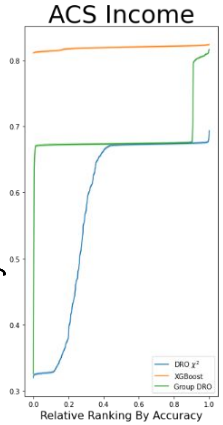


Trees are Robust to Model Selection Effects

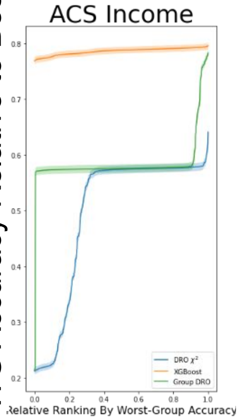


Trees are Less Sensitive to Choice of Hyperparameters

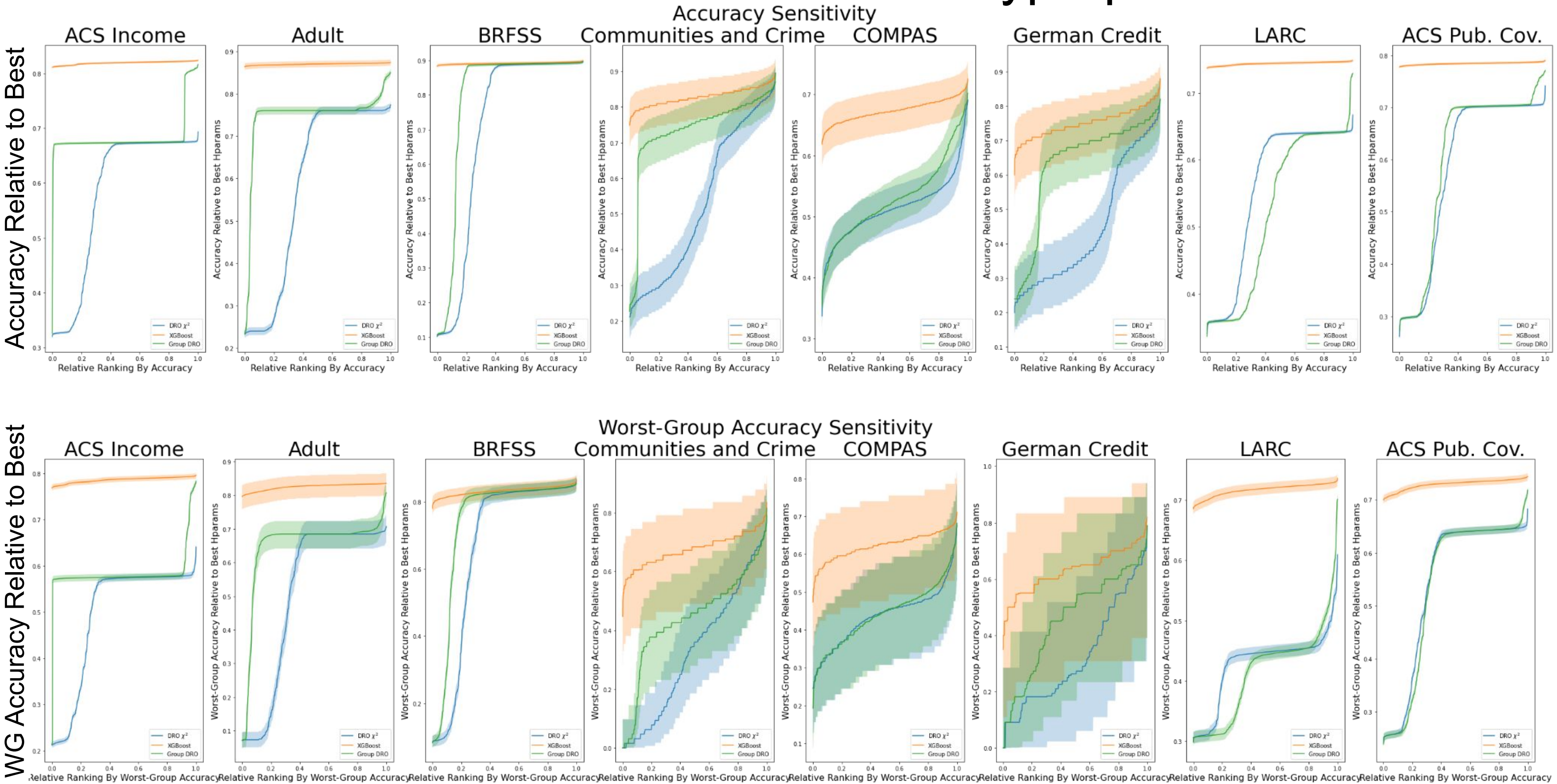
Accuracy Relative to Best



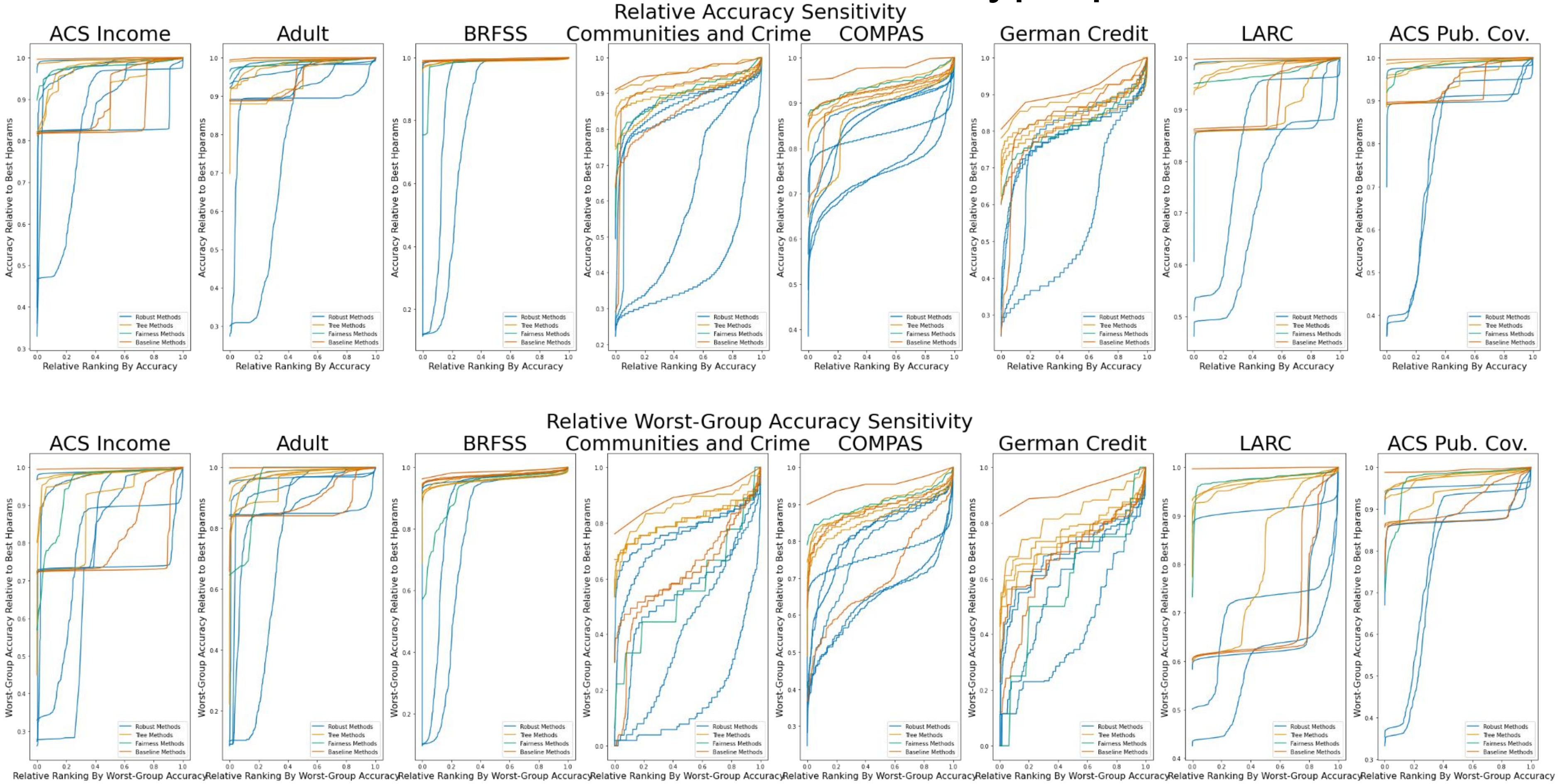
WG Accuracy Relative to Best



Trees are Less Sensitive to Choice of Hyperparameters



Trees are Less Sensitive to Choice of Hyperparameters



Outline

Introduction

Two Perspectives on Subgroup Robustness

Study Design + Datasets

Results

Accuracy-Robustness Frontiers

Evaluating Evaluation Metrics + Model Selection Effects

Hyperparameter Sensitivity

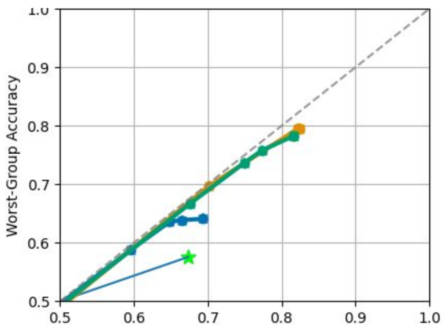
Conclusions

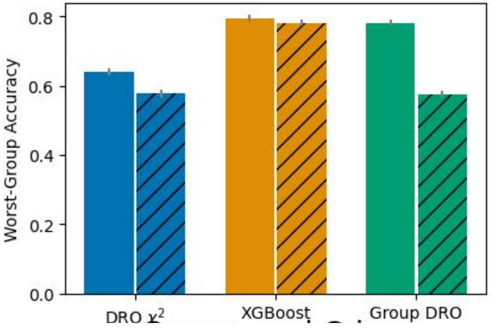
Implications for Practice + Future Work

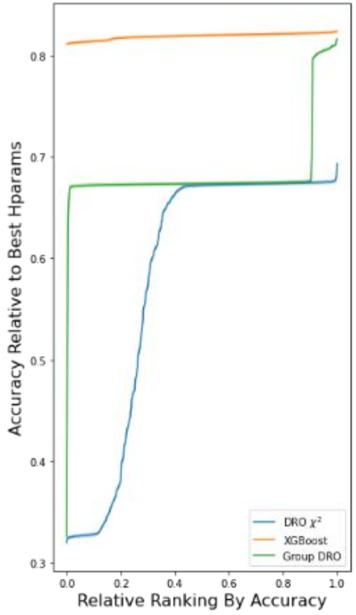
Takeaways

Tree-based models (XGBoost, LightGBM, etc.) are **surprisingly strong** subgroup robustness baselines.

These models are **cheaper to train**, **less sensitive to hyperparameters**, and **less sensitive to the model selection** metric.







Future Directions

This finding is specific to **MLP-based models**, which are the exclusive (tabular) model evaluated in the robustness works we sought to benchmark.

→ Does shifting away from MLPs close the gap with trees?

This may be an artifact of well-known relationship between in-distribution and out-of-distribution accuracy (Miller et al. 2021).

→ How can we make neural architectures more tree-like (or adopt differentiable techniques for tree training to use robust learning) to take advantage of this near-linear empirical relationship?

References

Agarwal, Rishabh, et al. "Deep reinforcement learning at the edge of the statistical precipice." Advances in neural information processing systems 34 (2021): 29304-29320.

Borisov, Vadim, et al. "Deep neural networks and tabular data: A survey." arXiv preprint arXiv:2110.01889 (2021).

J. Duchi and H. Namkoong. Learning models with uniform performance via distributionally robust optimization. arXiv preprint arXiv:1810.08750, 2018.

Kadra, Arlind, et al. "Well-tuned simple nets excel on tabular datasets." Advances in neural information processing systems 34 (2021): 23928-23941.

Khani, Fereshte, Aditi Raghunathan, and Percy Liang. "Maximum weighted loss discrepancy." arXiv preprint arXiv:1906.03518 (2019).

Levy, Daniel, et al. "Large-scale methods for distributionally robust optimization." Advances in Neural Information Processing Systems 33 (2020): 8847-8860.

Liao, Thomas, et al. "Are we learning yet? a meta review of evaluation failures across machine learning." Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). 2021.

Miller, John P., et al. "Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization." International Conference on Machine Learning. PMLR, 2021.

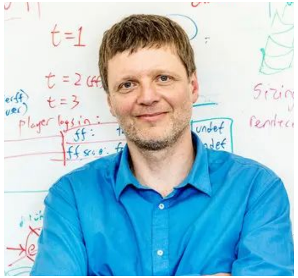
Sagawa, Shiori, et al. "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization." arXiv preprint arXiv:1911.08731 (2019).

Shwartz-Ziv, Ravid, and Amitai Armon. "Tabular data: Deep learning is not all you need." Information Fusion 81 (2022): 84-90.

Zhai, Runtian, et al. "Doro: Distributional and outlier robust optimization." International Conference on Machine Learning. PMLR, 2021.

Subgroup Robustness Grows on Trees: An Empirical Baseline Investigation

IFDS Workshop on Distributional Robustness
Aug. 5, 2022


Josh Gardner
jpgard@cs.washington.edu

Zoran Popović
zoran@cs.washington.edu

Ludwig Schmidt
schmidt@cs.washington.edu

Robust Sparse Mean Estimation via Sum-of-Squares

Sushrut Karmalkar
University of Wisconsin-Madison

Joint work with:
Ilias Diakonikolas Daniel Kane Ankit Pensia Thanasis Pittas

[COLT'2022]

Robust Statistics

Robust Statistics

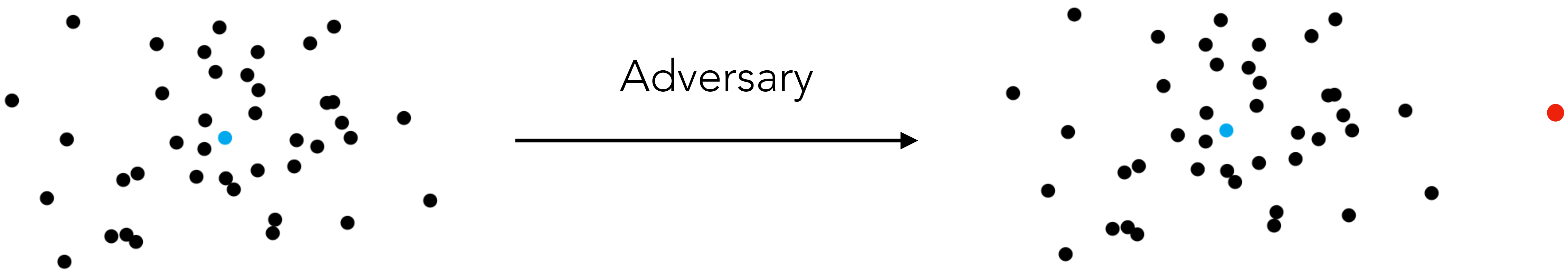
Goal: Signal recovery in the presence of **arbitrary, adversarial** corruptions.

Robust Statistics

Goal: Signal recovery in the presence of **arbitrary, adversarial** corruptions.

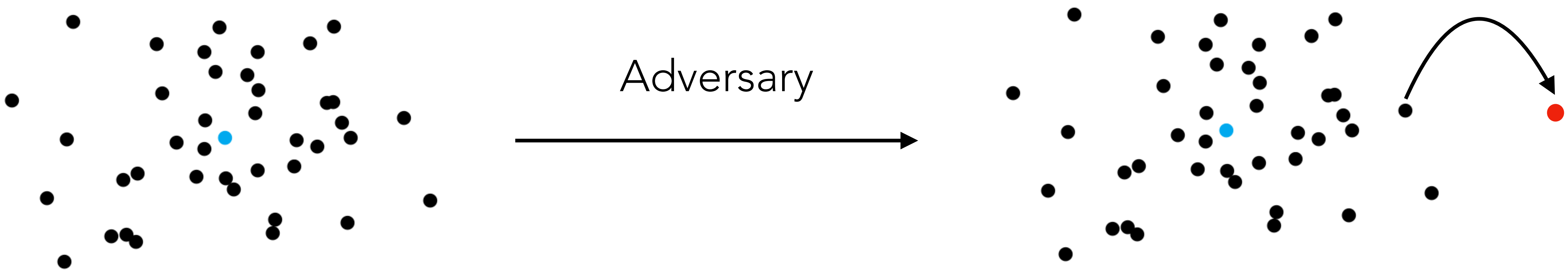
Robust Statistics

Goal: Signal recovery in the presence of **arbitrary, adversarial** corruptions.



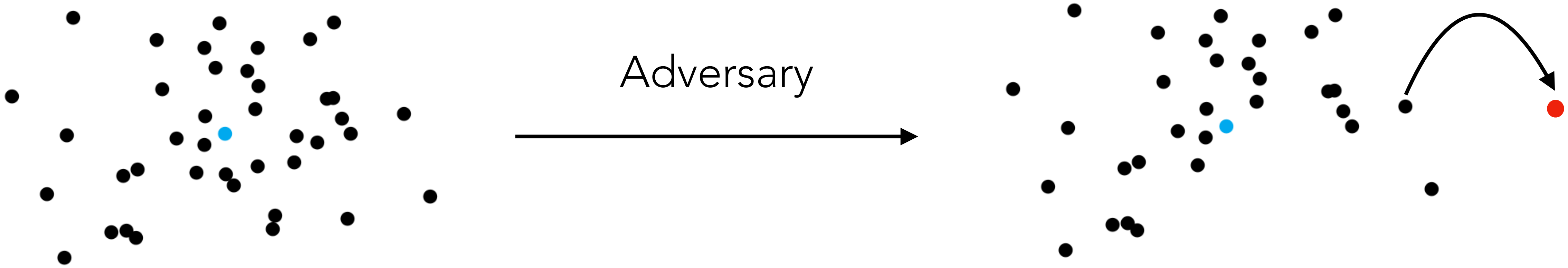
Robust Statistics

Goal: Signal recovery in the presence of **arbitrary, adversarial** corruptions.



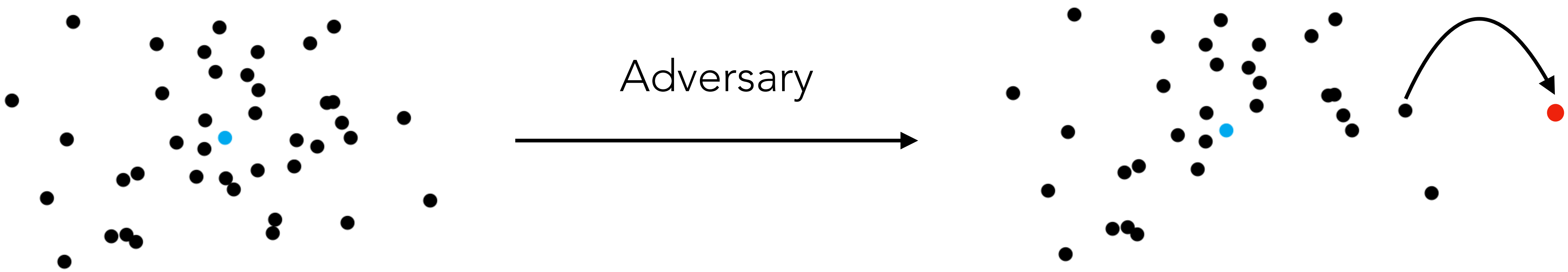
Robust Statistics

Goal: Signal recovery in the presence of **arbitrary, adversarial** corruptions.



Robust Statistics

Goal: Signal recovery in the presence of **arbitrary, adversarial** corruptions.



An estimator is **robust**, if it is able to estimate the signal, even in the presence of these corruptions.

Robust Statistics

Given: Samples from a distribution that is adversarially shifted in TV.

Recover: Signal when you know some properties of the inlier distribution

Parameters of Interest

Parameters of Interest

Fraction of Corruptions (ϵ): As **large** as possible.

Parameters of Interest

Fraction of Corruptions (ϵ): As **large** as possible.

Sample complexity: As **small** as possible for the given ϵ.

Parameters of Interest

Fraction of Corruptions (ϵ): As **large** as possible.

Sample complexity: As **small** as possible for the given ϵ.

Runtime: As **small** as possible, as a function of the input size.

Parameters of Interest

Fraction of Corruptions (ϵ)

Sample complexity

Runtime

Parameters of Interest

Fraction of Corruptions (ϵ)

Classical Robust Statistics

Sample complexity

[Tukey'60, Huber'64].

Runtime

Parameters of Interest

Fraction of Corruptions (ϵ)

Sample complexity

Algorithmic Robust Statistics

Runtime

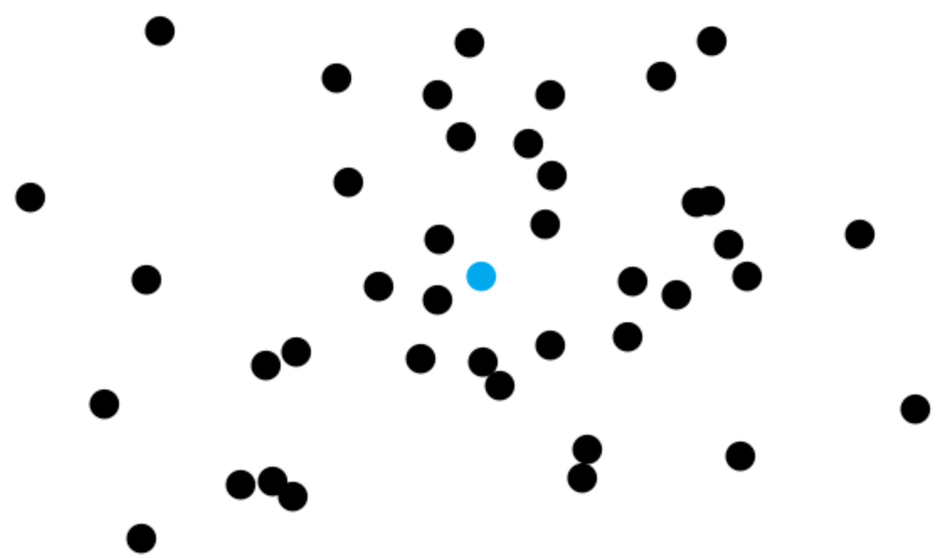
[Diakonikolas-Kane-Kamath-Li-Moitra-Stewart'16, Lai-Rao-Vempala'16]

Mean Estimation

Mean Estimation

Given: poly(d) samples drawn from $\mathscr{D}$ on $\mathbb{R}^d$ with mean μ.

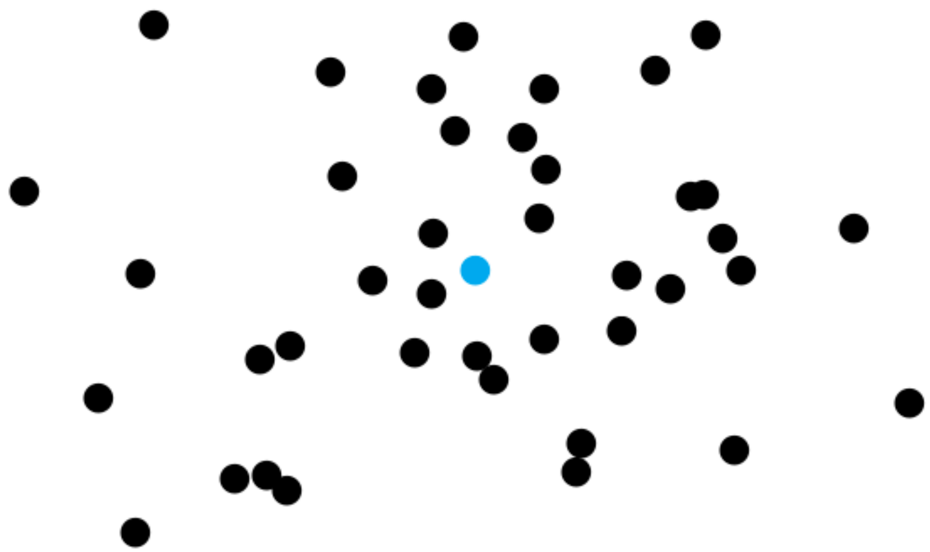
Recover: $\hat{\mu}$ such that $\|\hat{\mu} - \mu\|_2$ is small.



Mean Estimation

Given: poly(d) samples drawn from $\mathscr{D}$ on $\mathbb{R}^d$ with mean μ.

Recover: $\hat{\mu}$ such that $||\hat{\mu} - \mu||_2$ is small.

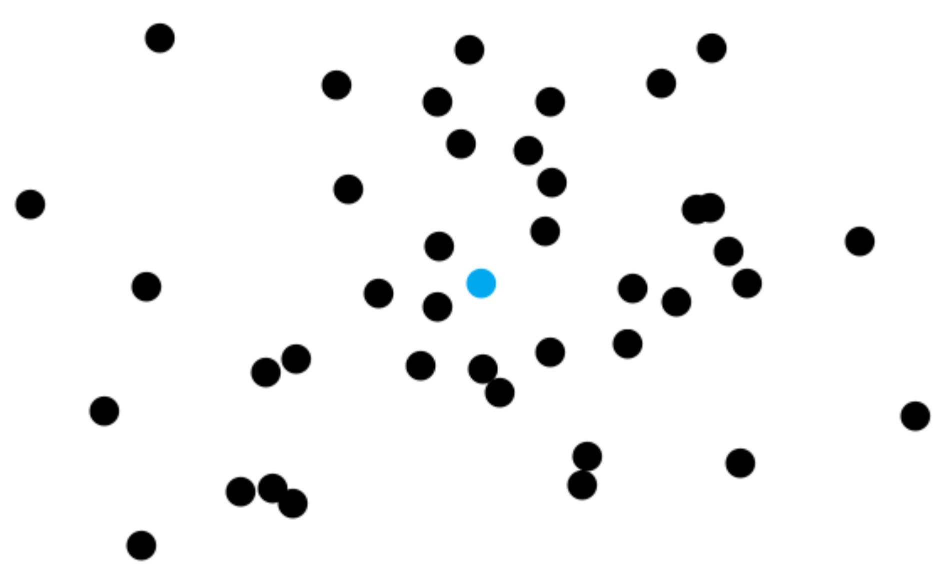


Need $\mathscr{D}$ to be structured for the **robust** setting - typically Gaussian, Log-concave etc.

Sparse Mean Estimation

Given: poly$(k, \log(d))$ samples, drawn from $\mathscr{D}$ with mean μ, where μ is k-sparse.

Recover: $\hat{\mu}$ such that $\|\hat{\mu} - \mu\|_2$ is small.



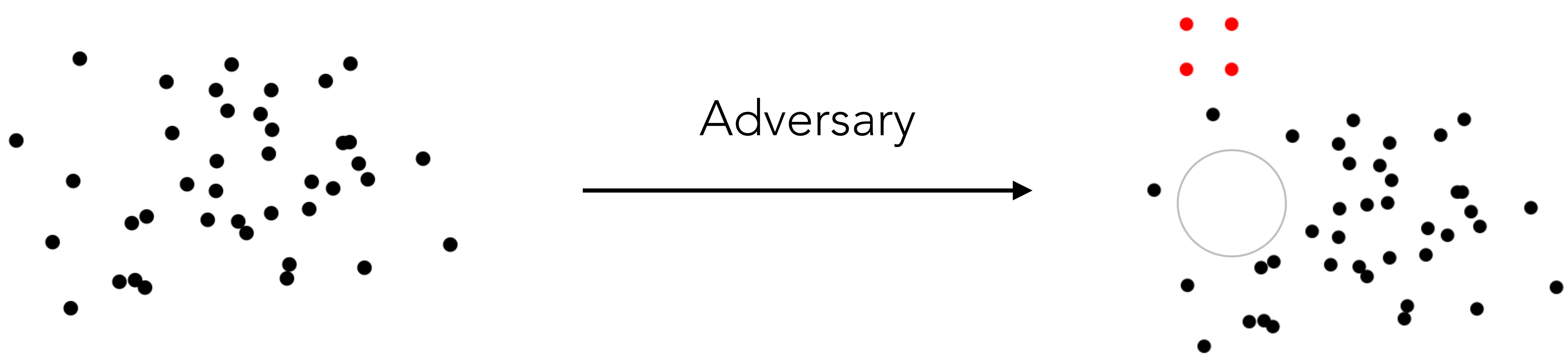
Outlier Model

Outlier Model

A sample set is ϵ-**corrupted**, if an adversary has been allowed to inspect and arbitrarily corrupt an ϵ fraction of the sample set.

Outlier Model

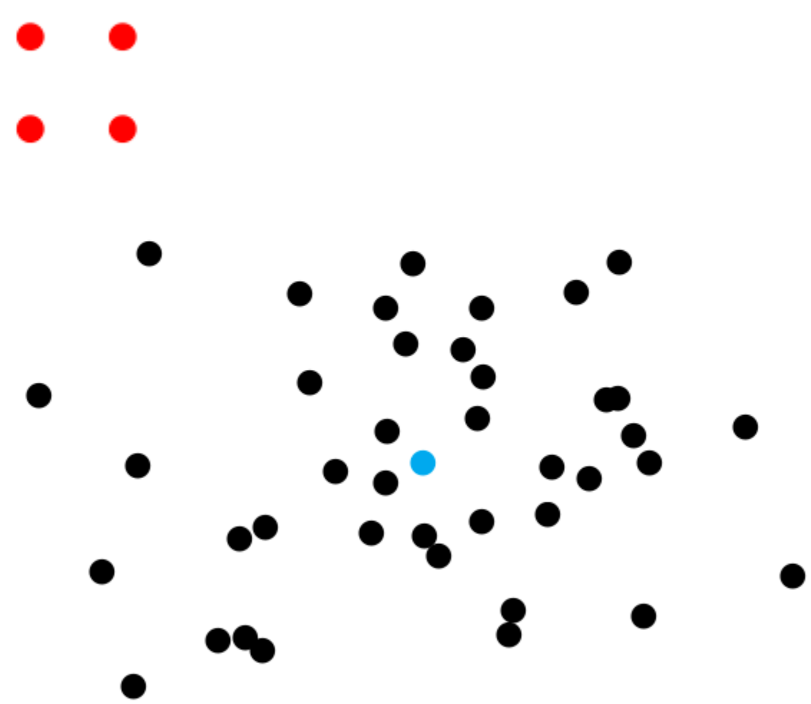
A sample set is ϵ**-corrupted**, if an adversary has been allowed to inspect and arbitrarily corrupt an ϵ fraction of the sample set.



Robust Sparse Mean Estimation

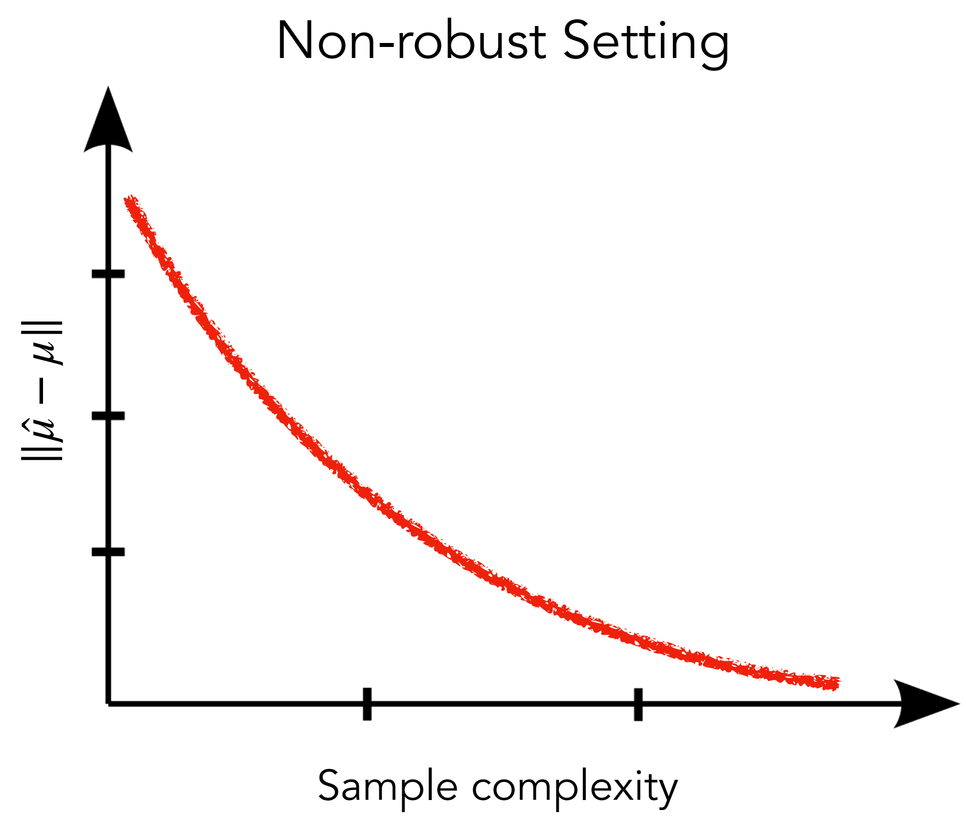
Given: ϵ-corrupted poly$(k, \log(d))$ size sample set, inliers drawn from $\mathscr{D}$ on $\mathbb{R}^d$ with a k-sparse mean μ.

Recover: $\hat{\mu}$ such that $\|\hat{\mu} - \mu\|_2$ is small.



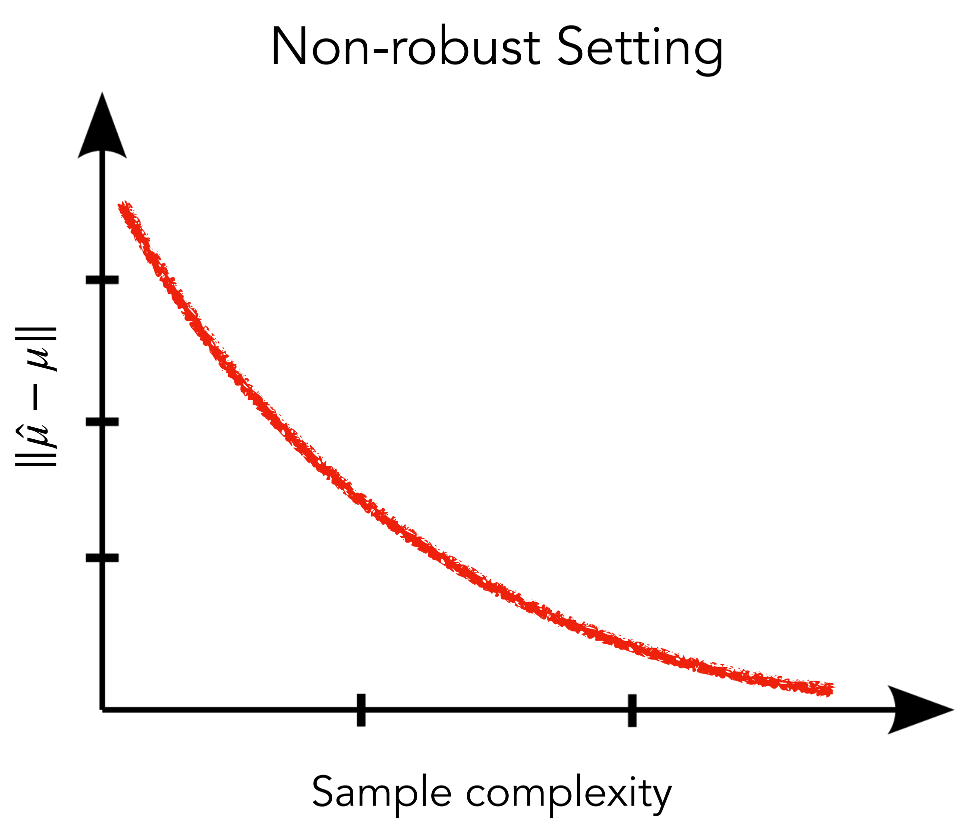
Goal: Non-robust vs robust

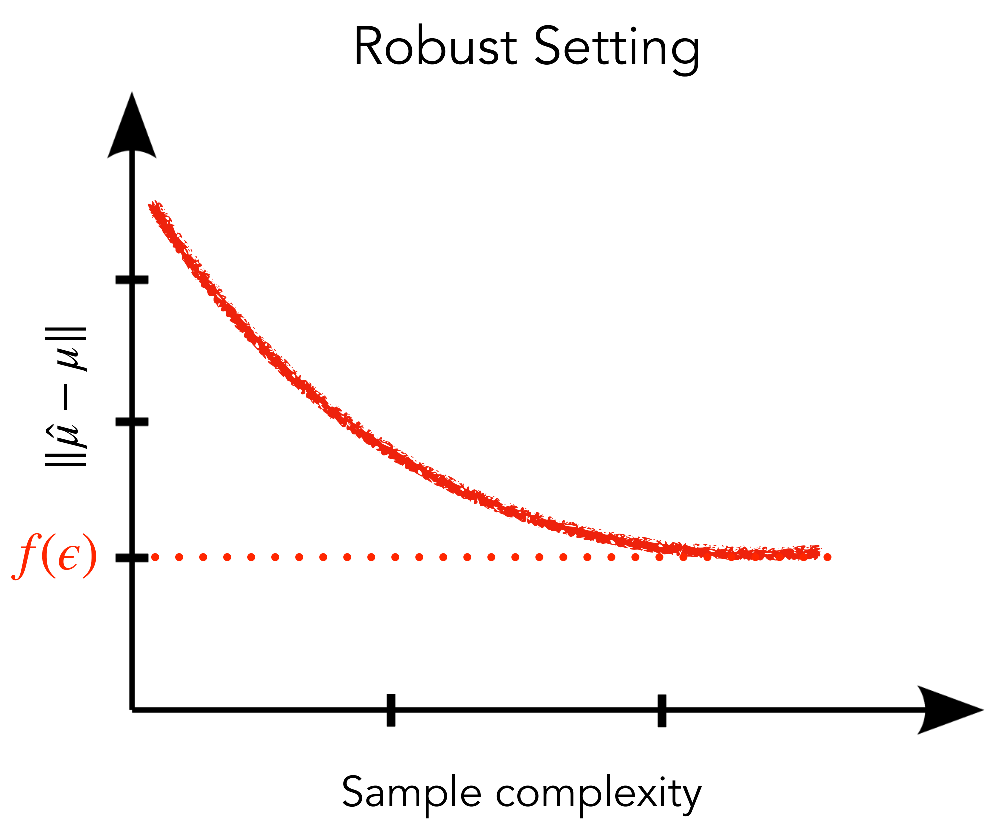
Goal: Non-robust vs robust



Find an algorithm achieving fastest rate of convergence.

Goal: Non-robust vs robust





Find an algorithm achieving fastest rate of convergence.

Find algorithm achieving slowest growing f.

Prior Work

Prior Work

High-dimensional Mean Estimation:

Prior Work

High-dimensional Mean Estimation:

- [Diakonikolas-Kane-Kamath-Li-Moitra-Stewart’16, Lai-Rao-Vempala’16]: Can tolerate a $O(1)$ / $O(1/\text{polylog}(d))$ fraction of Gaussian samples being corrupted respectively.

Prior Work

High-dimensional Mean Estimation:

- [Diakonikolas-Kane-Kamath-Li-Moitra-Stewart'16, Lai-Rao-Vempala'16]: Can tolerate a $O(1)$ / $O(1/\text{polylog}(d))$ fraction of Gaussian samples being corrupted respectively.
- [Kothari-Steurer'18, Hopkins-Li'18]: Recover results for more general class of distributions.

Prior Work

High-dimensional Mean Estimation:

- [Diakonikolas-Kane-Kamath-Li-Moitra-Stewart'16, Lai-Rao-Vempala'16]: Can tolerate a $O(1)$ / $O(1/\text{polylog}(d))$ fraction of Gaussian samples being corrupted respectively.
- [Kothari-Steurer'18, Hopkins-Li'18]: Recover results for more general class of distributions.

High-dimensional Sparse Mean Estimation:

Prior Work

High-dimensional Mean Estimation:

- [Diakonikolas-Kane-Kamath-Li-Moitra-Stewart'16, Lai-Rao-Vempala'16]: Can tolerate a $O(1)$ / $O(1/\text{polylog}(d))$ fraction of Gaussian samples being corrupted respectively.
- [Kothari-Steurer'18, Hopkins-Li'18]: Recover results for more general class of distributions.

High-dimensional Sparse Mean Estimation:

- [Balakrishnan-Du-Li-Singh'17]: Solves the problem for $\mathcal{N}(\mu, I_d)$. Requires ellipsoid method + SDP.

Prior Work

High-dimensional Mean Estimation:

- [Diakonikolas-Kane-Kamath-Li-Moitra-Stewart'16, Lai-Rao-Vempala'16]: Can tolerate a $O(1)$ / $O(1/\text{polylog}(d))$ fraction of Gaussian samples being corrupted respectively.
- [Kothari-Steurer'18, Hopkins-Li'18]: Recover results for more general class of distributions.

High-dimensional Sparse Mean Estimation:

- [Balakrishnan-Du-Li-Singh'17]: Solves the problem for $\mathcal{N}(\mu, I_d)$. Requires ellipsoid method + SDP.
- [Diakonikolas-**K**-Kane-Price-Stewart'19, Cheng-Diakonikolas-Kane-Ge-Gupta-Soltanokotabi'21]: More practical algorithms for $\mathcal{N}(\mu, I_d)$.

Prior Work

High-dimensional Mean Estimation:

- [Diakonikolas-Kane-Kamath-Li-Moitra-Stewart'16, Lai-Rao-Vempala'16]: Can tolerate a $O(1)$ / $O(1/\text{polylog}(d))$ fraction of Gaussian samples being corrupted respectively.
- [Kothari-Steurer'18, Hopkins-Li'18]: Recover results for more general class of distributions.

High-dimensional Sparse Mean Estimation:

- [Balakrishnan-Du-Li-Singh'17]: Solves the problem for $\mathcal{N}(\mu, I_d)$. Requires ellipsoid method + SDP.
- [Diakonikolas-**K**-Kane-Price-Stewart'19, Cheng-Diakonikolas-Kane-Ge-Gupta-Soltanokotabi'21]: More practical algorithms for $\mathcal{N}(\mu, I_d)$.

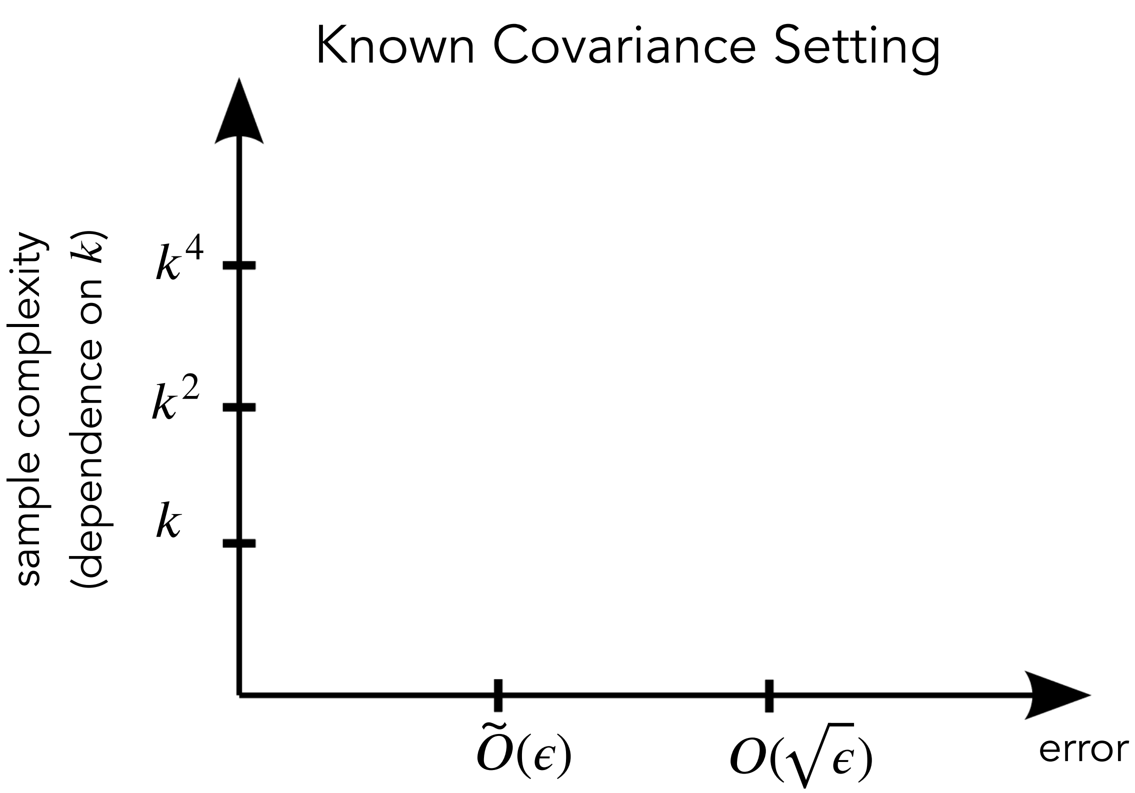
Question: Is there an algorithm in the **sparse** setting which can achieve near-optimal guarantees with **bounded, unknown** covariance?

Landscape: Gaussian Setting

Landscape: Gaussian Setting

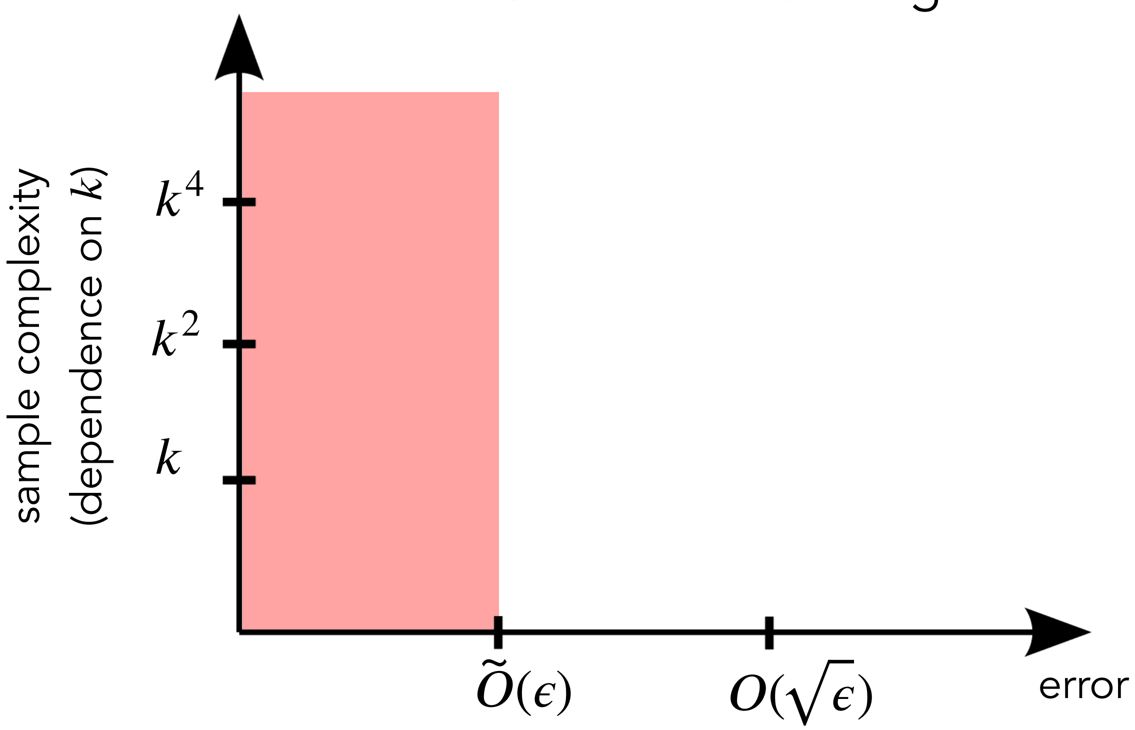
Known Covariance Setting

Landscape: Gaussian Setting

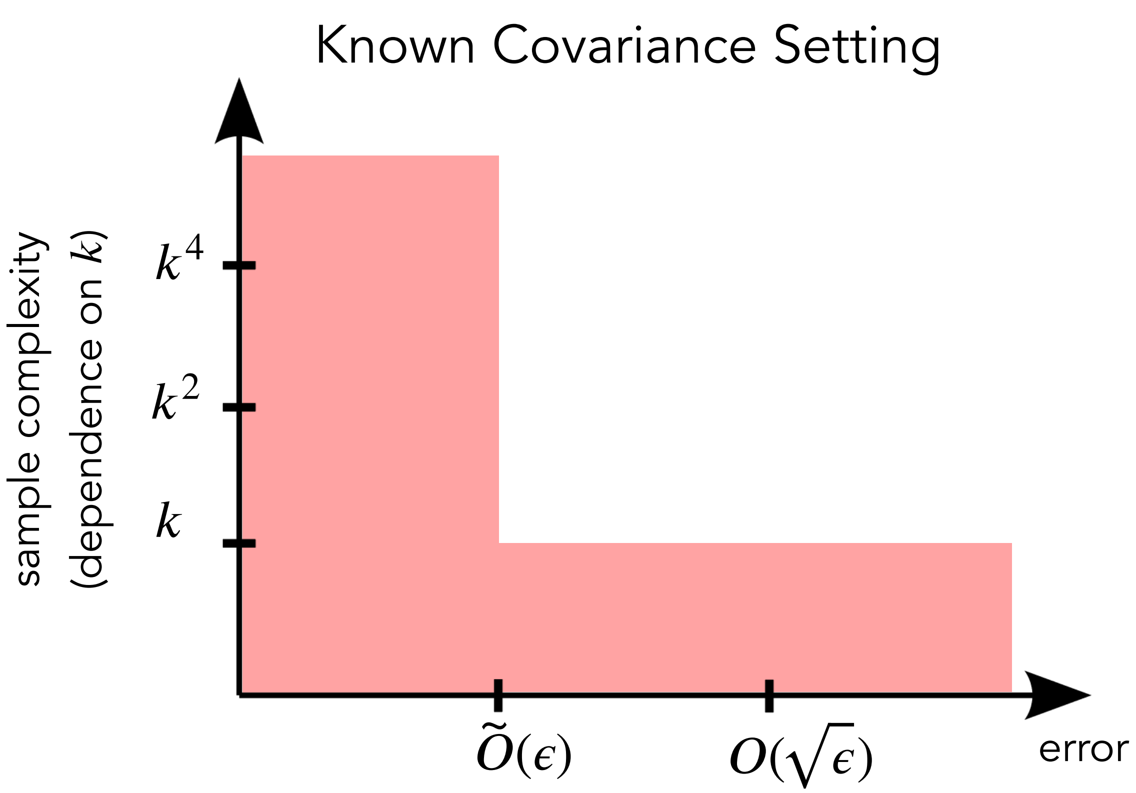


Landscape: Gaussian Setting

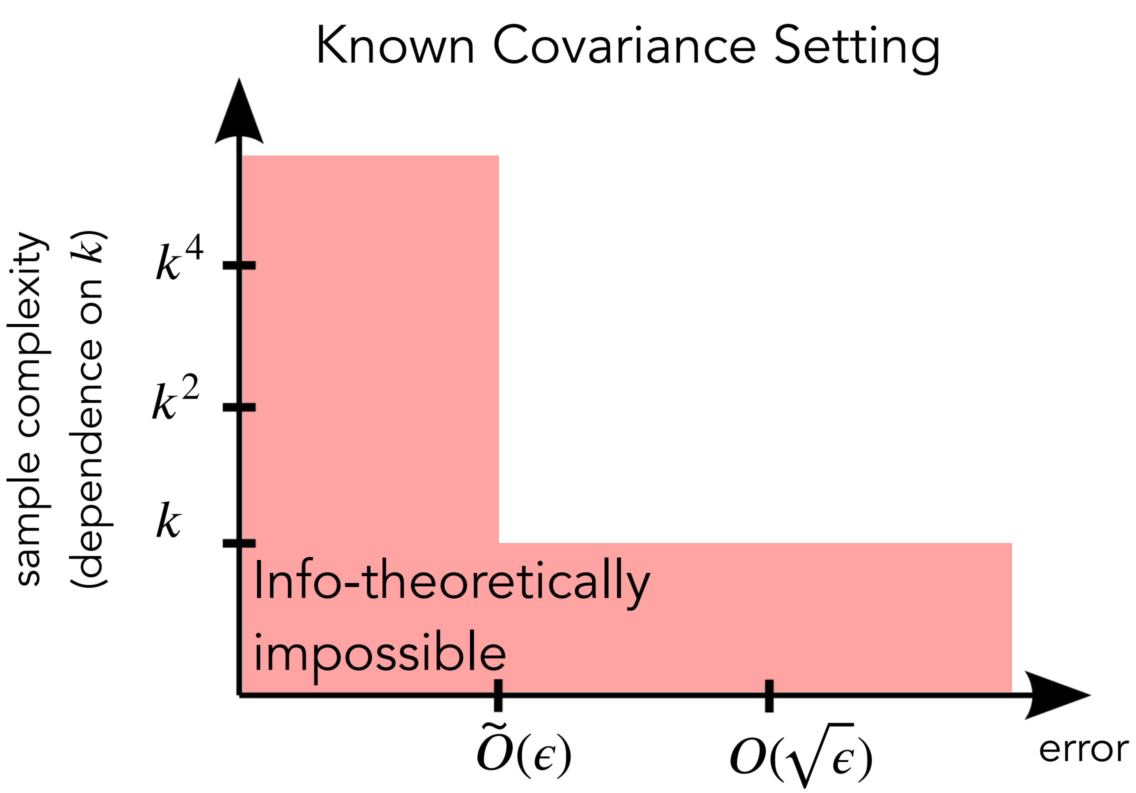
Known Covariance Setting



Landscape: Gaussian Setting

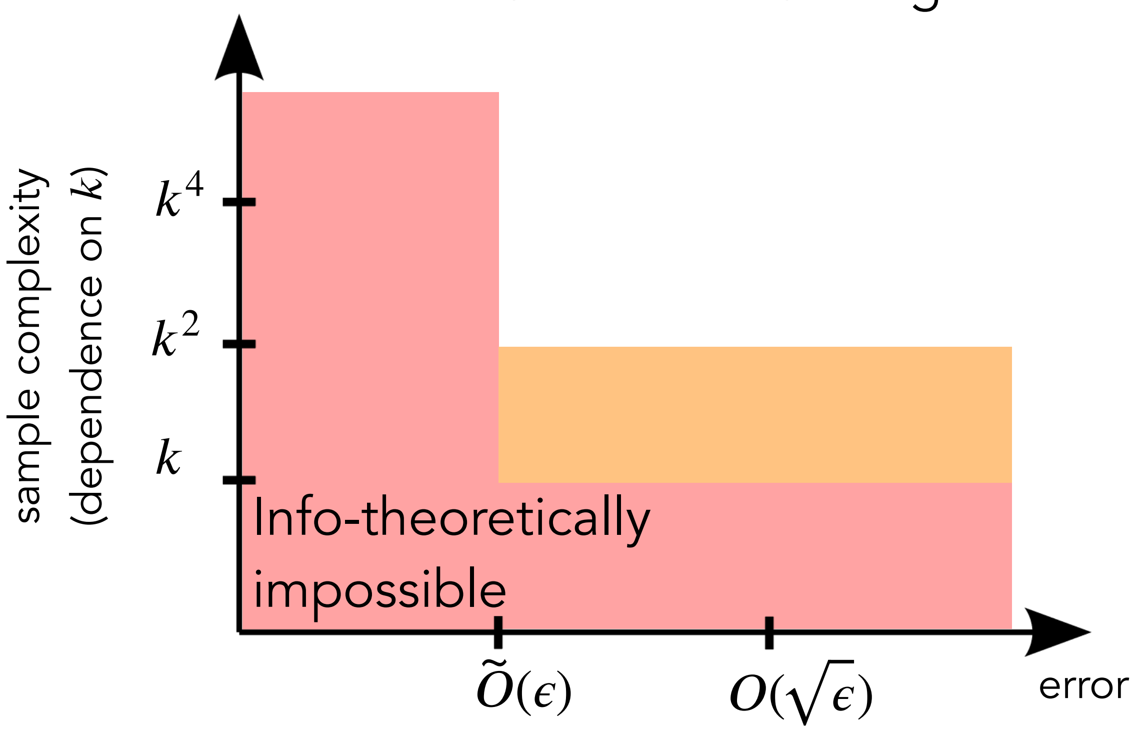


Landscape: Gaussian Setting

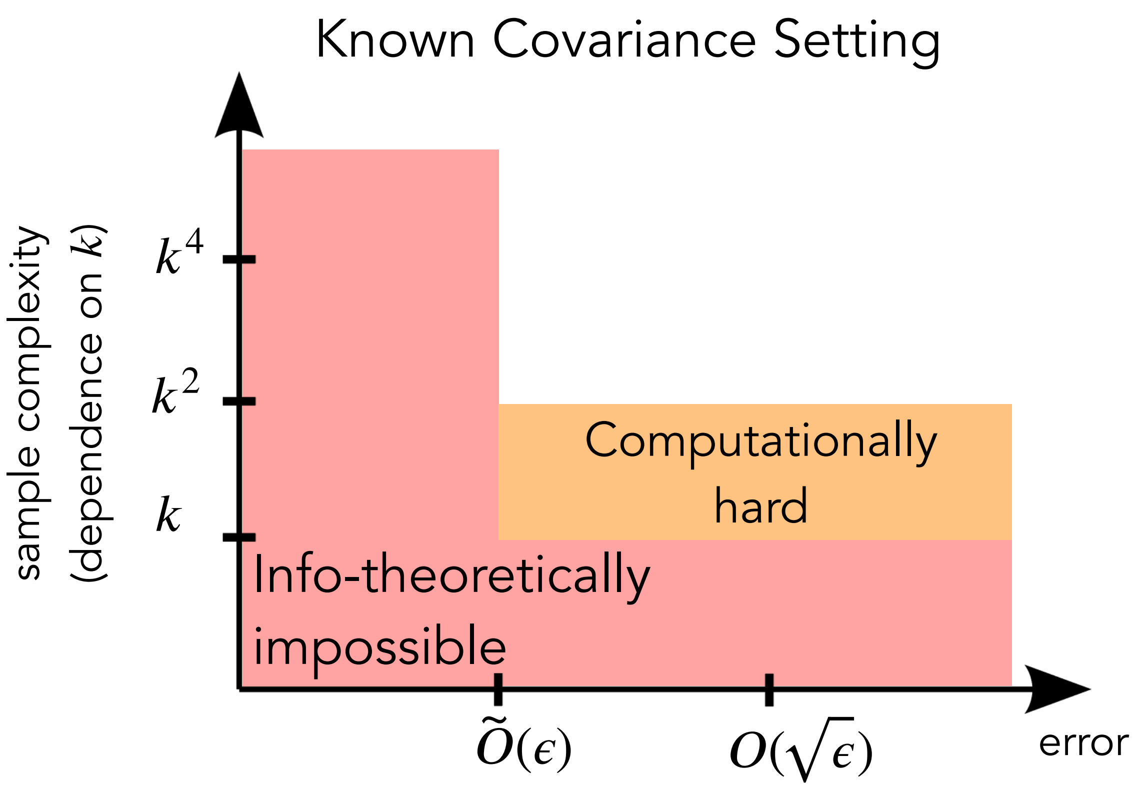


Landscape: Gaussian Setting

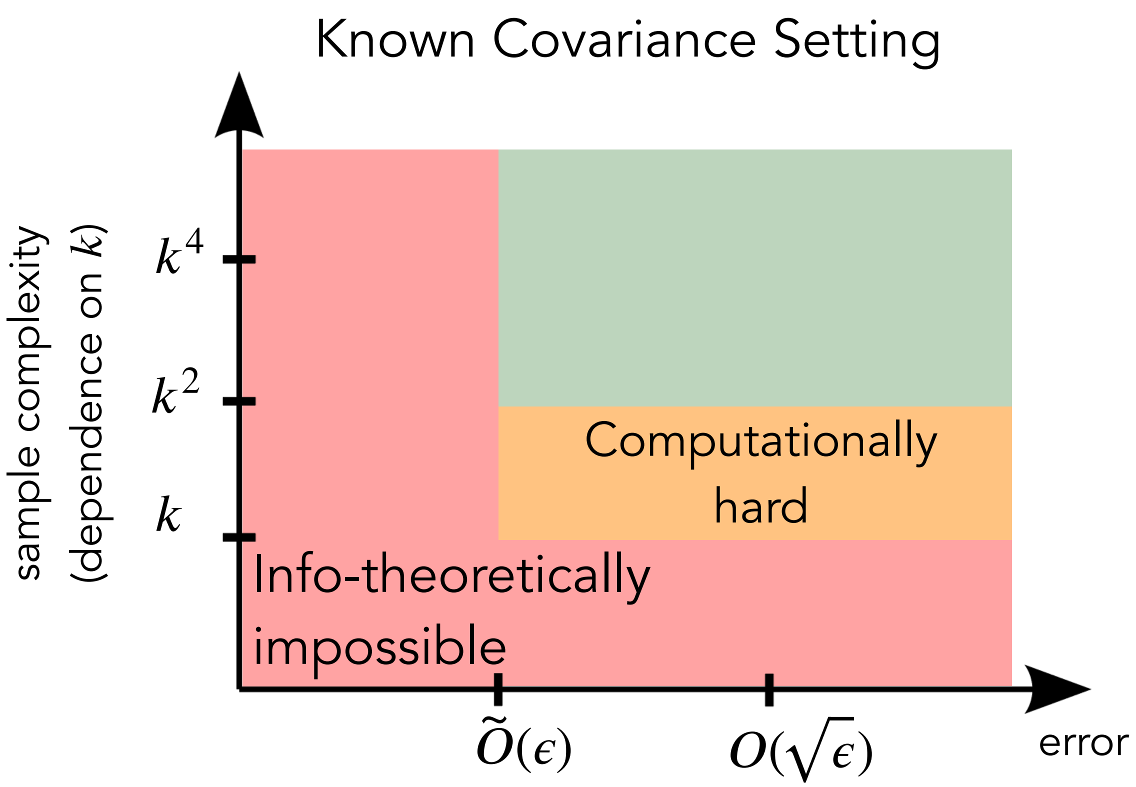
Known Covariance Setting



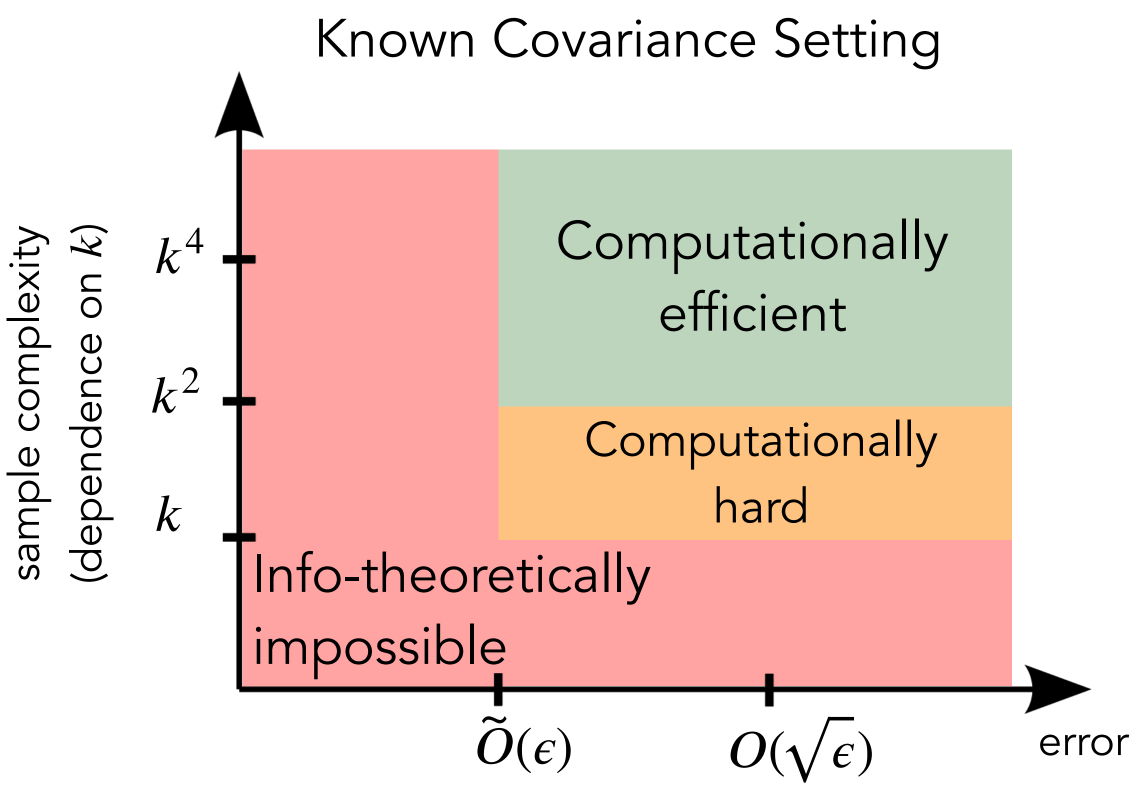
Landscape: Gaussian Setting



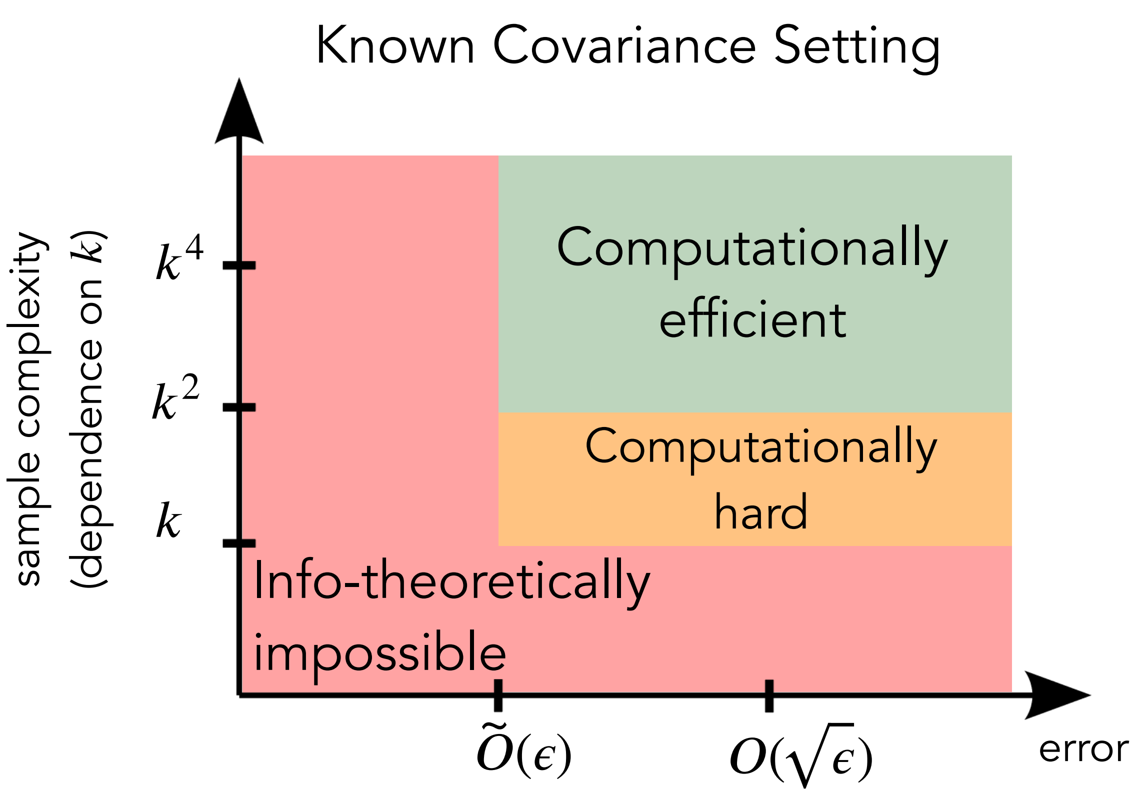
Landscape: Gaussian Setting

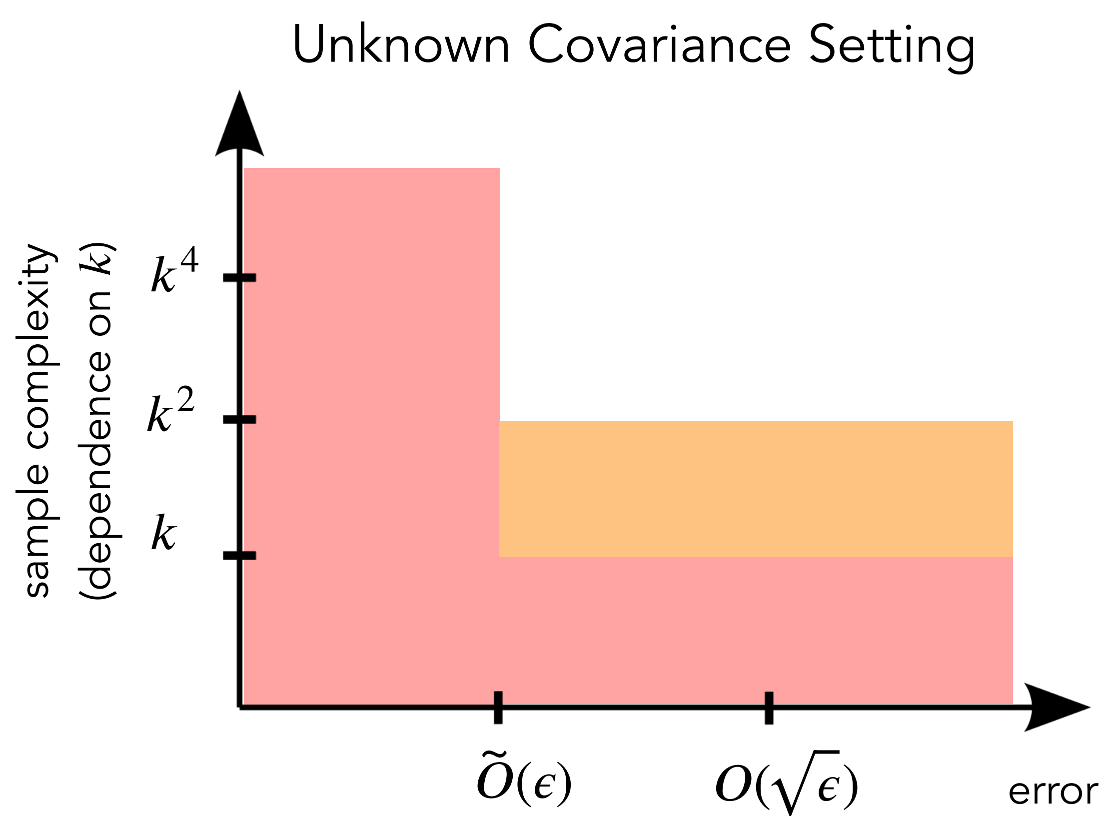


Landscape: Gaussian Setting

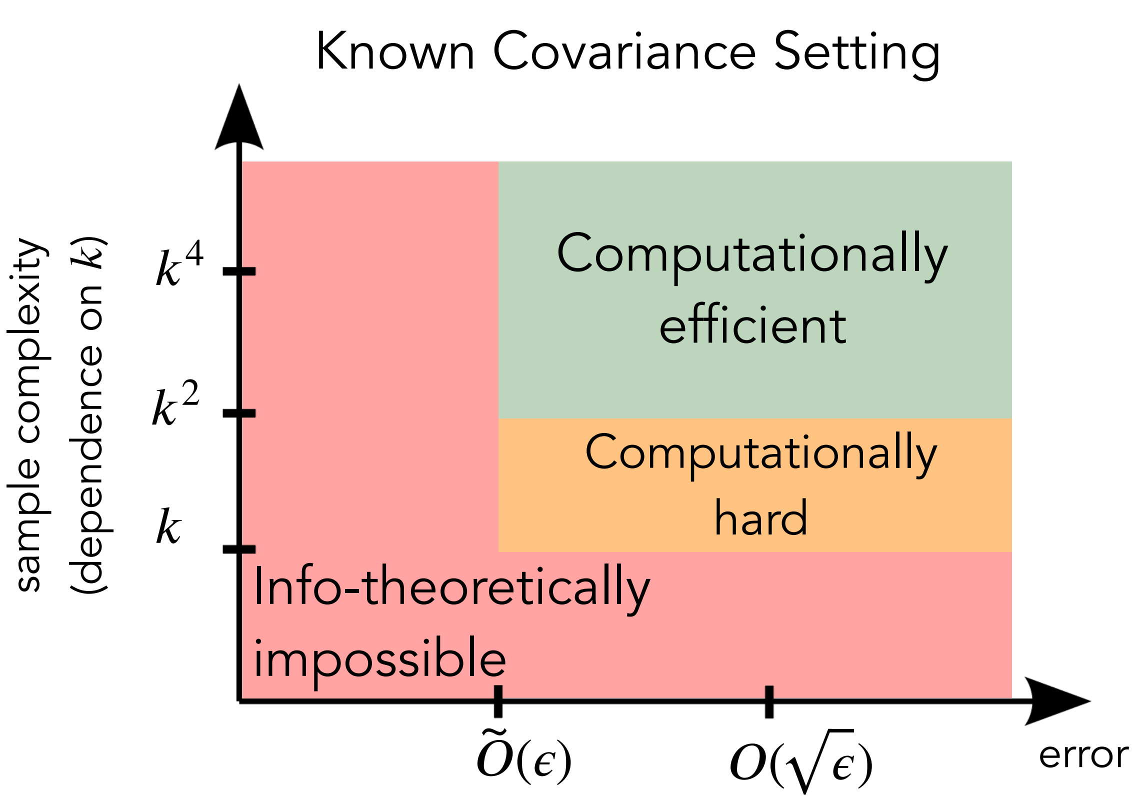


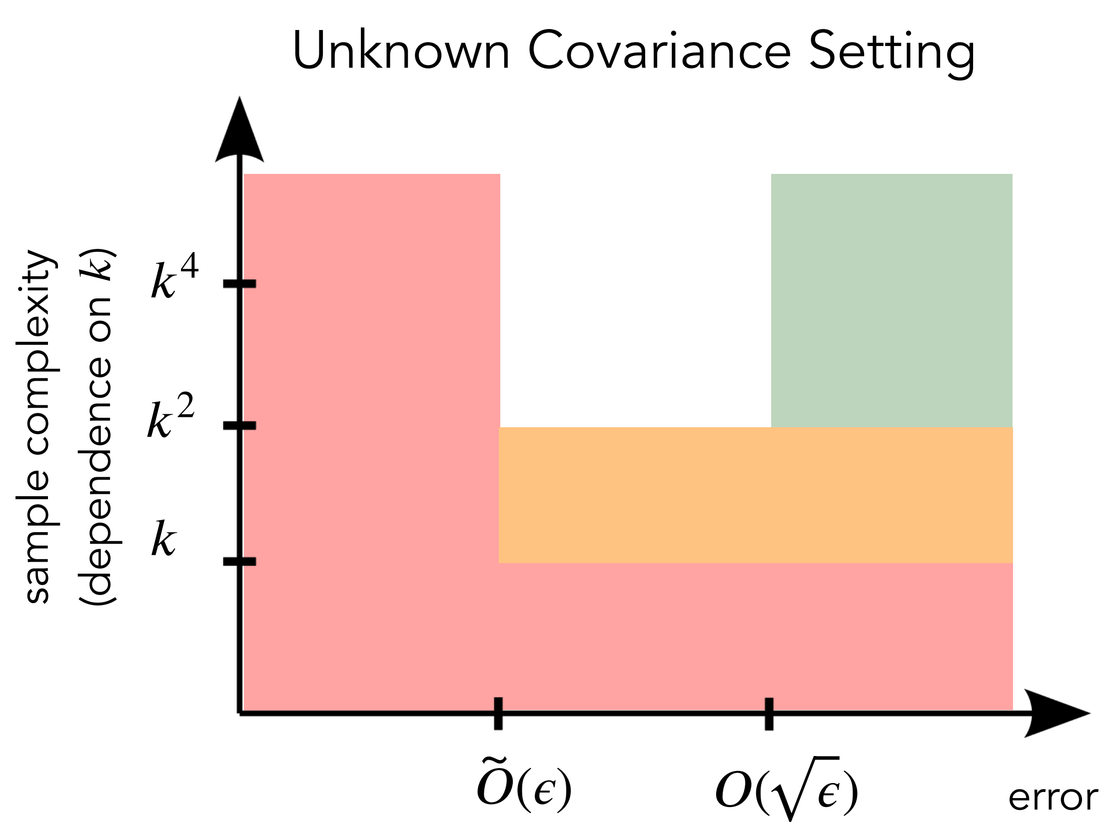
Landscape: Gaussian Setting



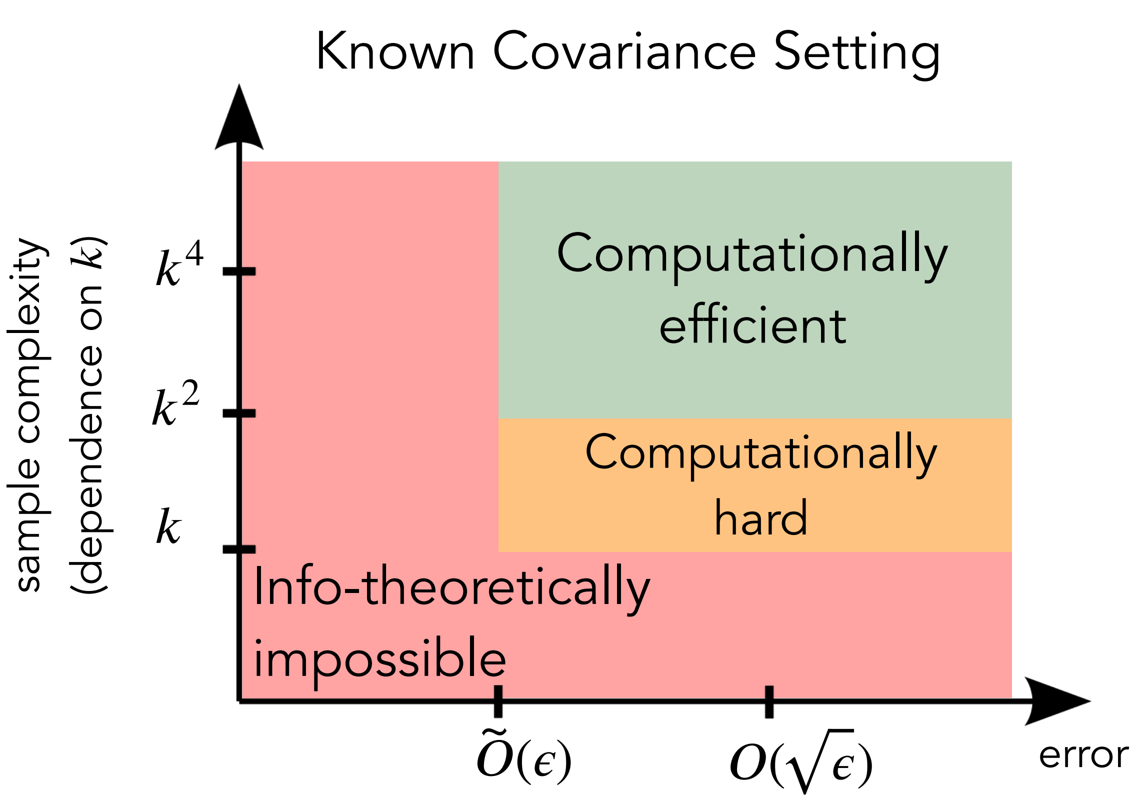


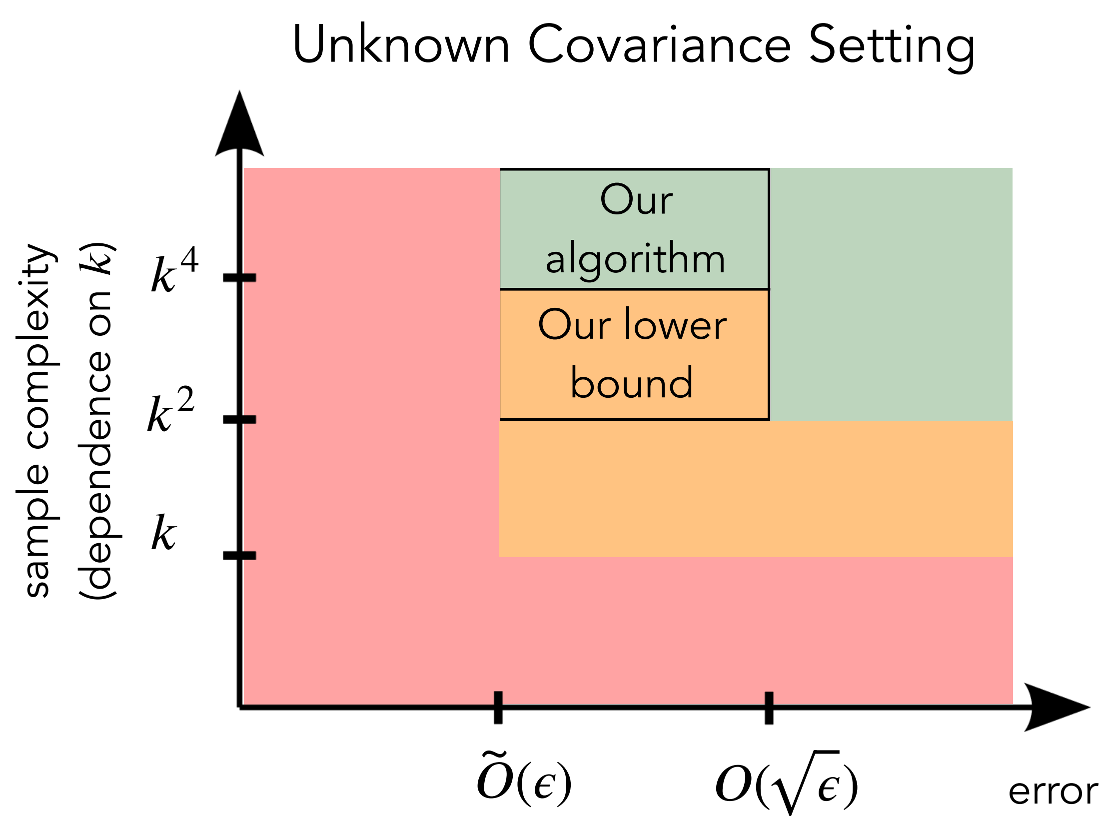
Landscape: Gaussian Setting





Landscape: Gaussian Setting





Our Results: Gaussian Setting

Our Results: Gaussian Setting

Theorem: There exists an algorithm which,

Our Results: Gaussian Setting

Theorem: There exists an algorithm which,

- Takes $n = O((k^4/\epsilon^2)$ polylog$(d/\epsilon))$ ϵ-corrupted samples from $\mathcal{N}(\mu, \Sigma)$ for unknown μ, Σ.

Our Results: Gaussian Setting

Theorem: There exists an algorithm which,

- Takes $n = O((k^4/\epsilon^2)\ \text{polylog}(d/\epsilon))$ ϵ-corrupted samples from $\mathcal{N}(\mu, \Sigma)$ for unknown μ, Σ.
- Runs in time $\text{poly}(nd)$.

Our Results: Gaussian Setting

Theorem: There exists an algorithm which,

- Takes $n = O((k^4/\epsilon^2)$ polylog$(d/\epsilon))$ ϵ-corrupted samples from $\mathcal{N}(\mu, \Sigma)$ for unknown μ, Σ.
- Runs in time poly(nd).
- Recovers $\hat{\mu}$ satisfying, $\|\hat{\mu} - \mu\|_2 \leq \tilde{O}(\epsilon)\sqrt{\|\Sigma\|_{\mathrm{op}}}$ w.h.p.

Our Results: Gaussian Setting

Theorem: There exists an algorithm which,

- Takes $n = O((k^4/\epsilon^2)$ polylog$(d/\epsilon))$ ϵ-corrupted samples from $\mathcal{N}(\mu, \Sigma)$ for unknown μ, Σ.
- Runs in time poly(nd).
- Recovers $\hat{\mu}$ satisfying, $\|\hat{\mu} - \mu\|_2 \leq \tilde{O}(\epsilon)\sqrt{\|\Sigma\|_{\text{op}}}$ w.h.p.

We give nearly matching Statistical Query lower bound suggesting that $\mathbf{\Omega(k^4)}$ samples are necessary.

Our Results: Bounded Moments

Our Results: Bounded Moments

Setting:

1. The inlier distribution has its first t moments *"certifiably"* bounded by $O(1)$.*
2. The **covariance is unknown** to the statistician.

*We also need the first $t \log(d)$ moments bounded by $O(1)$.

Our Results: Bounded Moments

Setting:

1. The inlier distribution has its first t moments "*certifiably*" bounded by $O(1)$.*
2. The **covariance is unknown** to the statistician.

Theorem: There exists an algorithm which,

*We also need the first $t\log(d)$ moments bounded by $O(1)$.

Our Results: Bounded Moments

Setting:

1. The inlier distribution has its first t moments "*certifiably*" bounded by $O(1)$.*
2. The **covariance is unknown** to the statistician.

Theorem: There exists an algorithm which,

- Takes $n = \dfrac{(k\log(d))^{O(t)}}{\epsilon^2}$ $\epsilon-$corrupted samples.

*We also need the first $t\log(d)$ moments bounded by $O(1)$.

Our Results: Bounded Moments

Setting:
1. The inlier distribution has its first t moments "*certifiably*" bounded by $O(1)$.*
2. The **covariance is unknown** to the statistician.

Theorem: There exists an algorithm which,

- Takes $n = \dfrac{(k\log(d))^{O(t)}}{\epsilon^2}$ $\epsilon-$corrupted samples.
- Runs in time poly$((nd)^t)$.

*We also need the first $t\log(d)$ moments bounded by $O(1)$.

Our Results: Bounded Moments

Setting:

1. The inlier distribution has its first t moments "*certifiably*" bounded by $O(1)$.*
2. The **covariance is unknown** to the statistician.

Theorem: There exists an algorithm which,

- Takes $n = \dfrac{(k\log(d))^{O(t)}}{\epsilon^2}$ $\epsilon-$corrupted samples.
- Runs in time poly$((nd)^t)$.
- Returns $\hat{\mu}$ satisfying $\|\hat{\mu} - \mu\|_2 \leq O(\epsilon^{1-1/t})$ w.h.p.

*We also need the first $t\log(d)$ moments bounded by $O(1)$.

Our Results: Bounded Moments

Setting:

1. The inlier distribution has its first t moments "*certifiably*" bounded by $O(1)$.*
2. The **covariance is unknown** to the statistician.

Theorem: There exists an algorithm which,

- Takes $n = \dfrac{(k\log(d))^{O(t)}}{\epsilon^2}$ $\epsilon-$corrupted samples.
- Runs in time poly$((nd)^t)$.
- Returns $\hat{\mu}$ satisfying $\|\hat{\mu} - \mu\|_2 \leq O(\epsilon^{1-1/t})$ w.h.p.

We give nearly matching Statistical Query lower bound suggesting that this is the optimal guarantee possible.

*We also need the first $t\log(d)$ moments bounded by $O(1)$.

Summary

Summary

- Algorithms for robust sparse mean estimation when the covariance is unknown.

Sushrut Karmalkar

email: skarmalkar@wisc.edu

Summary

- Algorithms for robust sparse mean estimation when the covariance is unknown.
    - Gaussian

Sushrut Karmalkar email: skarmalkar@wisc.edu

Summary

- Algorithms for robust sparse mean estimation when the covariance is unknown.
    - Gaussian
    - Bounded moments

Summary

- Algorithms for robust sparse mean estimation when the covariance is unknown.
  - Gaussian
  - Bounded moments
- Statistical query lower bounds suggesting optimality.

Sushrut Karmalkar email: skarmalkar@wisc.edu

Summary

- Algorithms for robust sparse mean estimation when the covariance is unknown.
  - Gaussian
  - Bounded moments
- Statistical query lower bounds suggesting optimality.

Questions?

Sushrut Karmalkar email: skarmalkar@wisc.edu

Thank You