Cross-validation: how to properly assess predictive performance?

Pradeep Reddy Raamana

crossinvalidation.com

Follow @raamana_

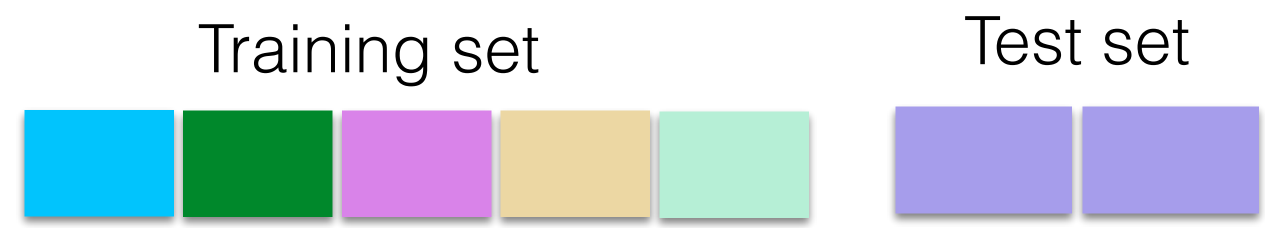
Baycrest

CONP PCNO

ONTARIO BRAIN INSTITUTE

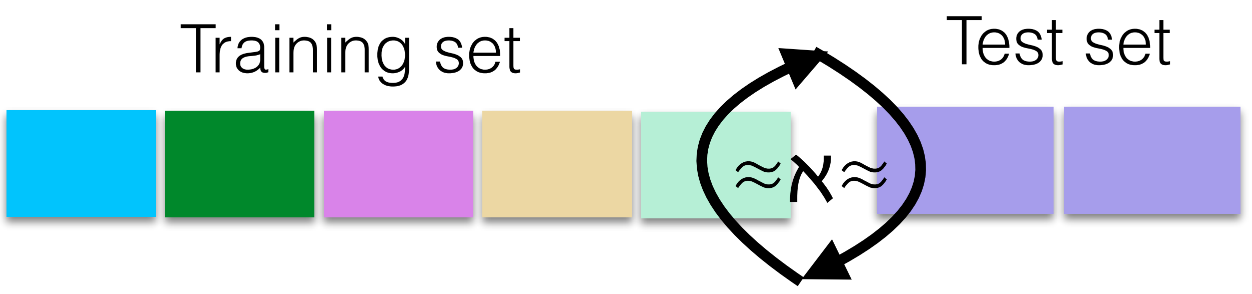
Goals for Today

- What is cross-validation?



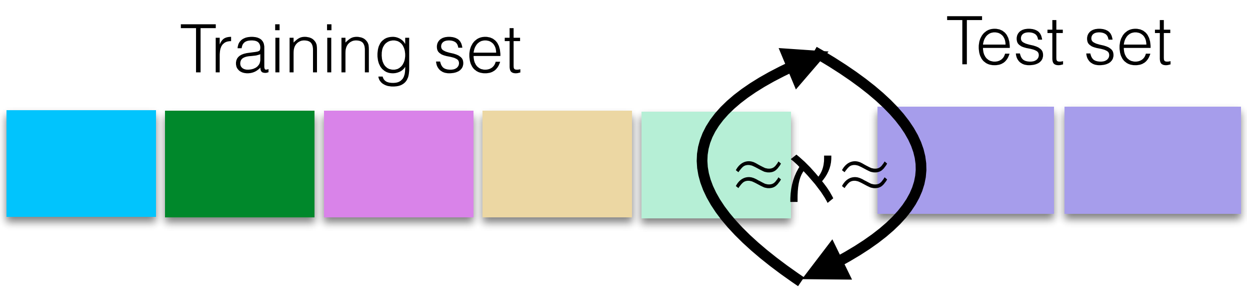
Goals for Today

- What is cross-validation?
- How to do it correctly?



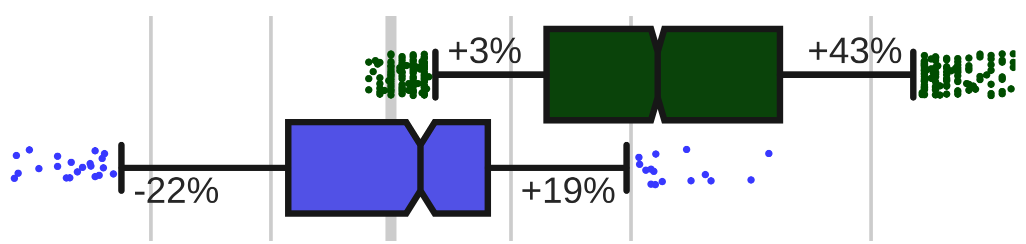
Goals for Today

- What is cross-validation?



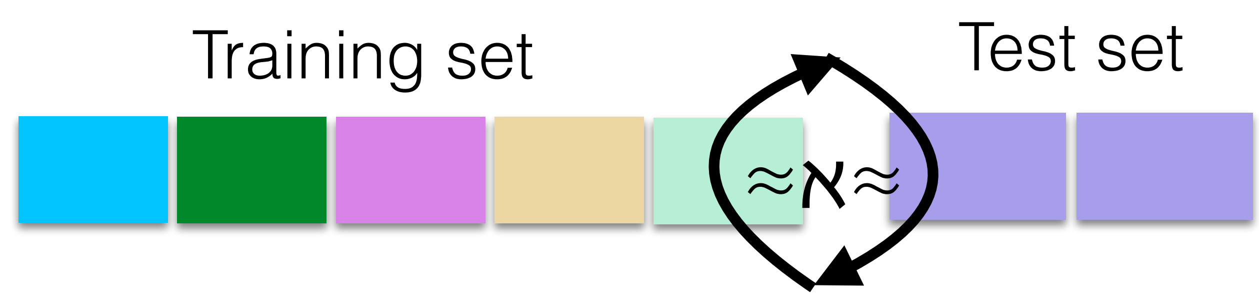
- How to do it correctly?

- What are the effects of different CV choices?

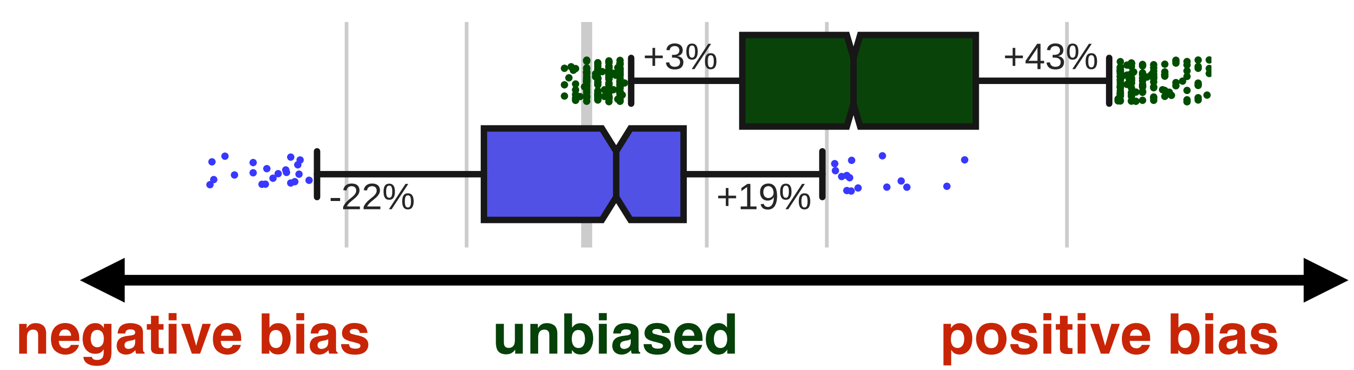


Goals for Today

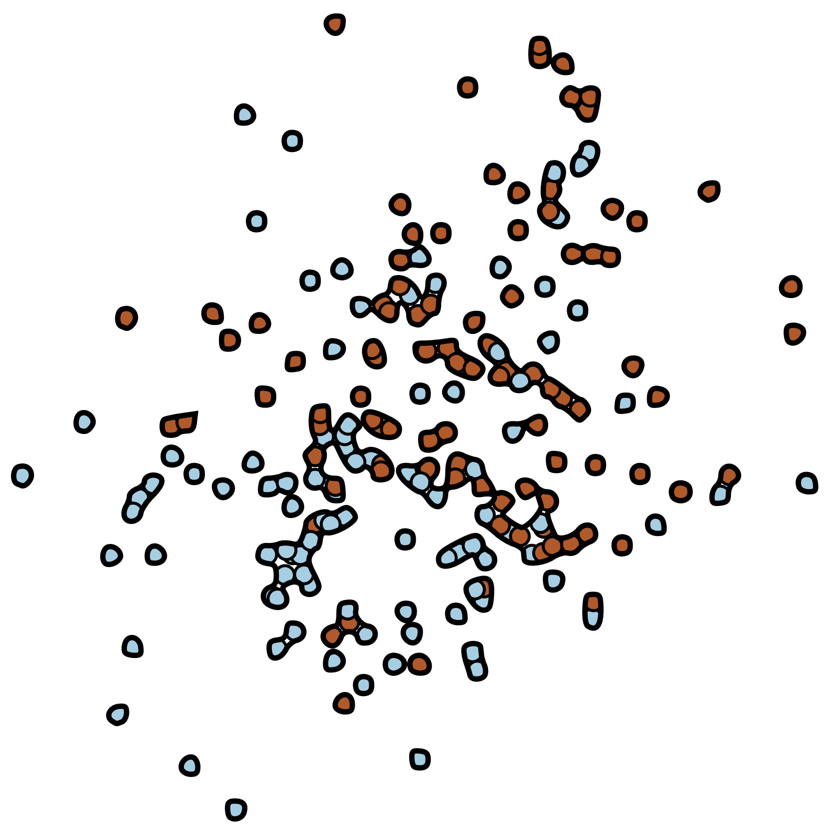
- What is cross-validation?



- How to do it correctly?
- What are the effects of different CV choices?



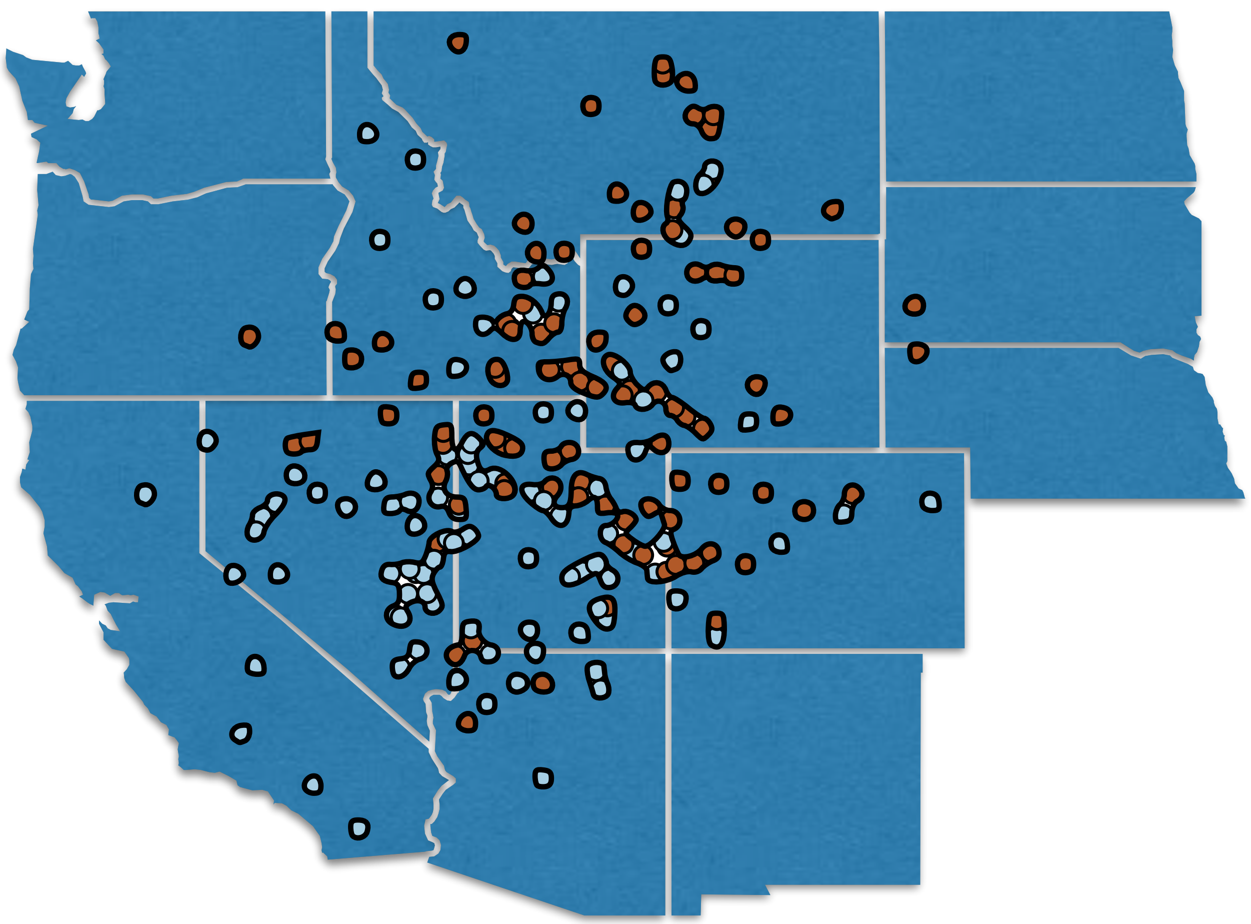
What is generalizability?



available
data (sample*)

*has a statistical definition

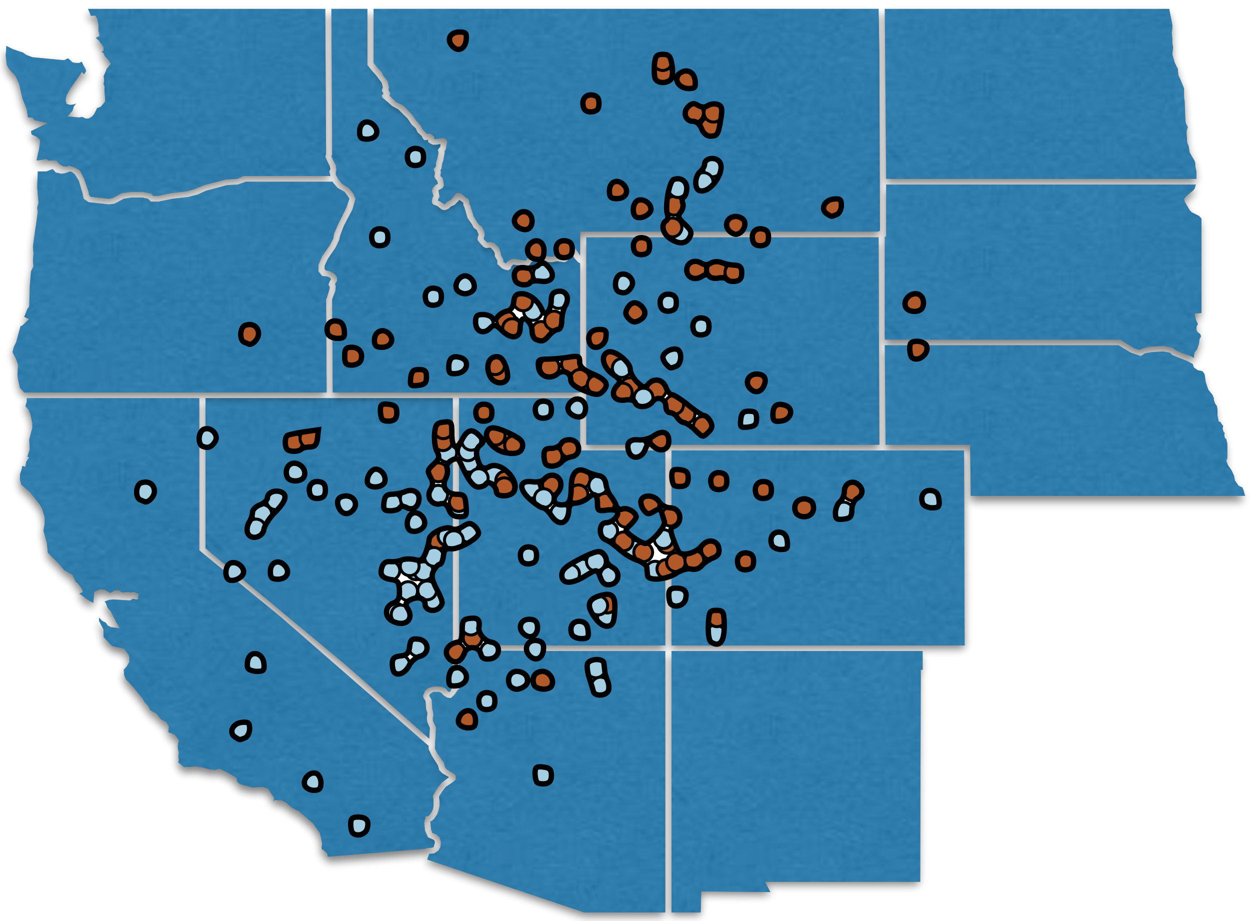
What is generalizability?



available
data (sample*)

*has a statistical definition

What is generalizability?

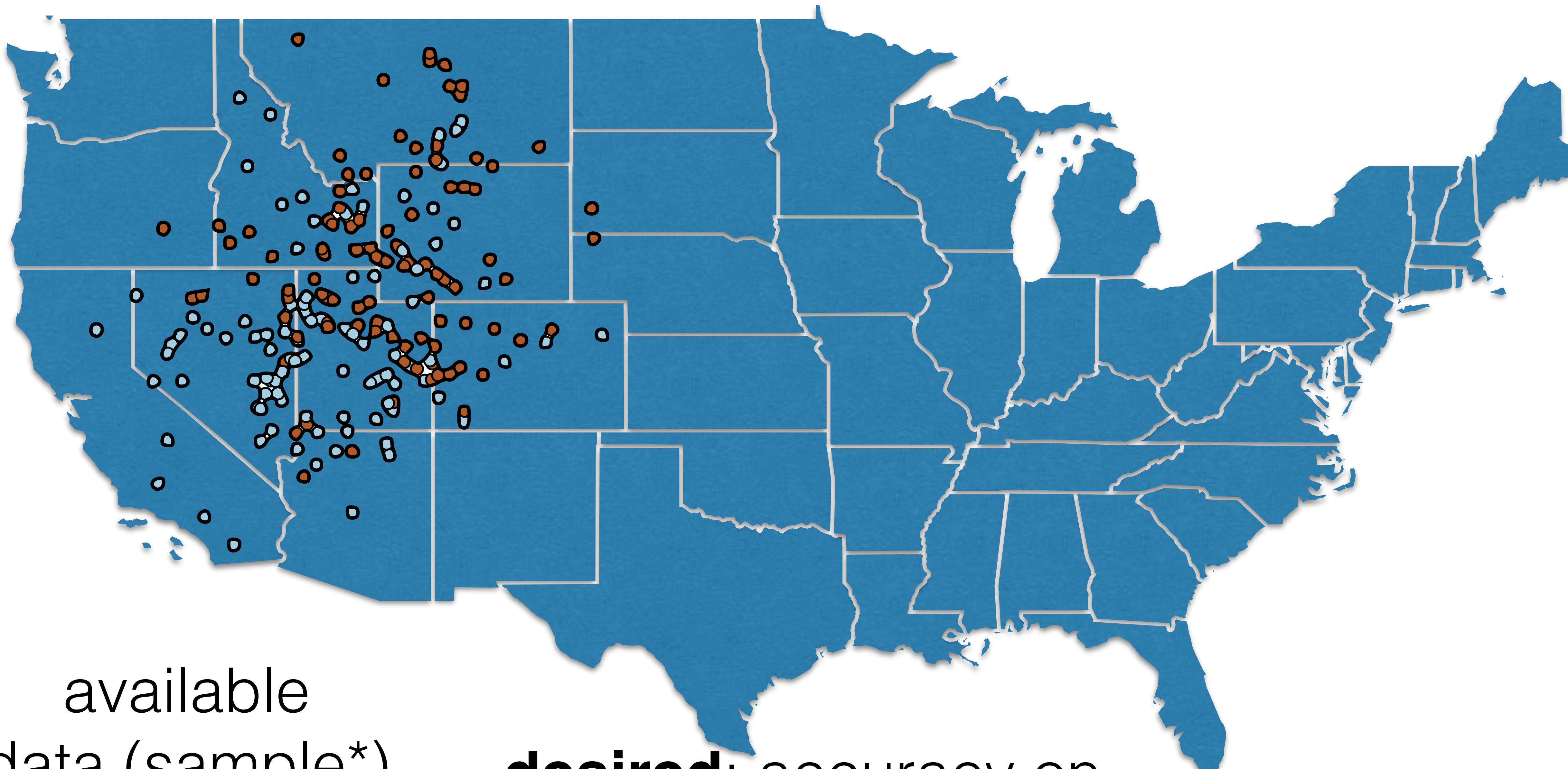


available
data (sample*)

desired: accuracy on
unseen data (population*)

*has a statistical definition

What is generalizability?

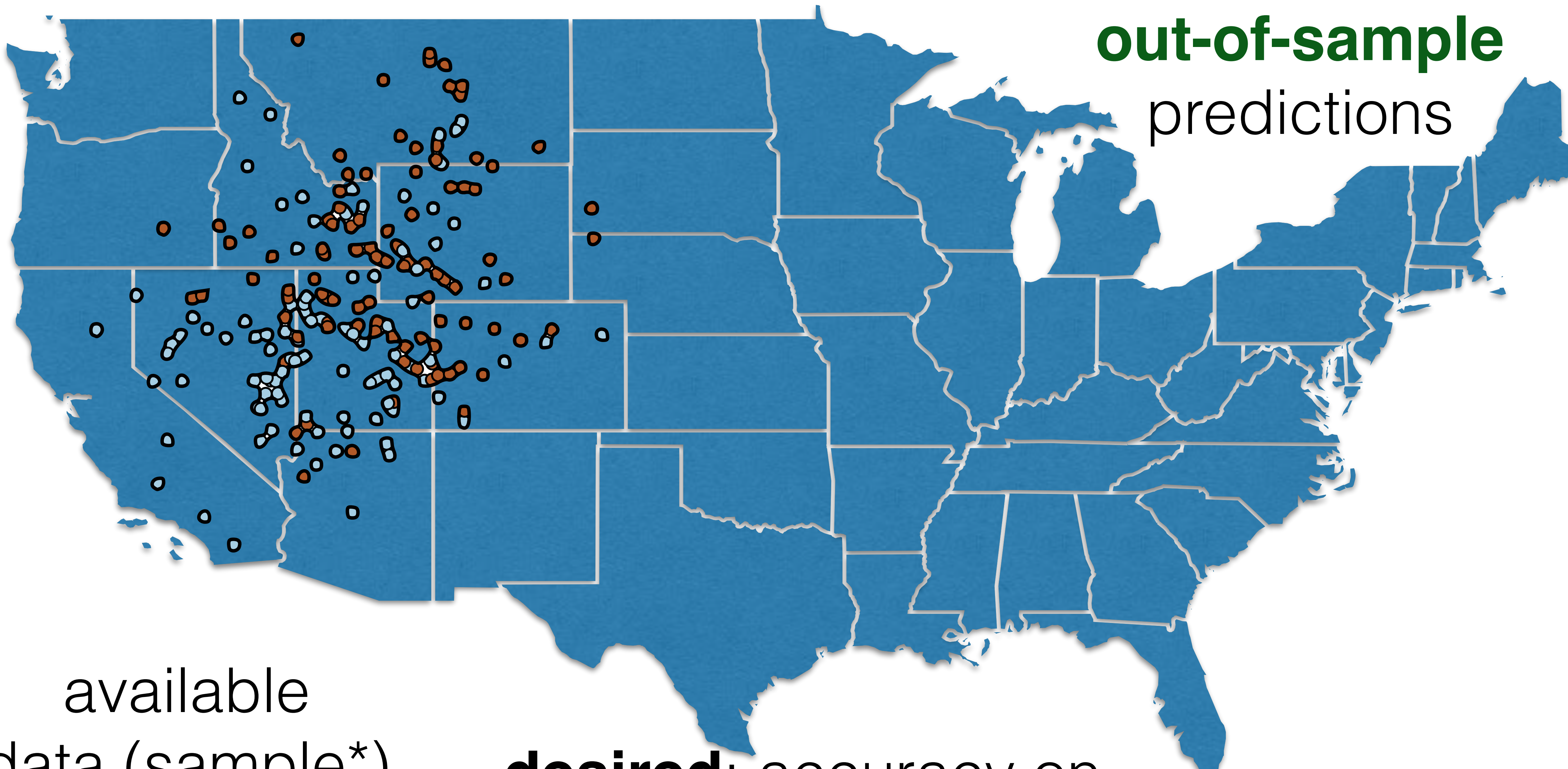


available
data (sample*)

desired: accuracy on
unseen data (population*)

*has a statistical definition

What is generalizability?



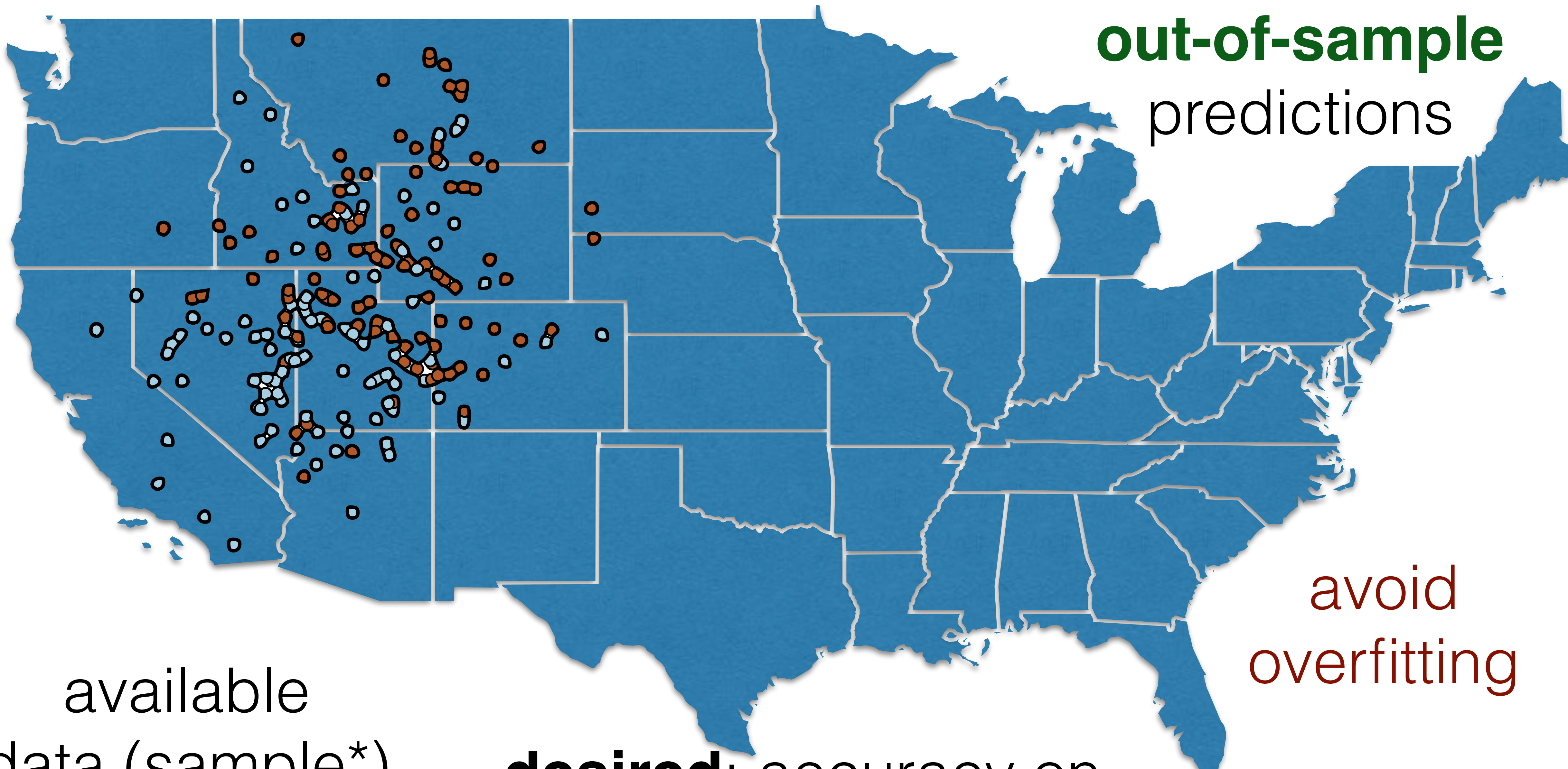
out-of-sample predictions

available data (sample*)

desired: accuracy on **unseen** data (population*)

*has a statistical definition

What is generalizability?



out-of-sample predictions

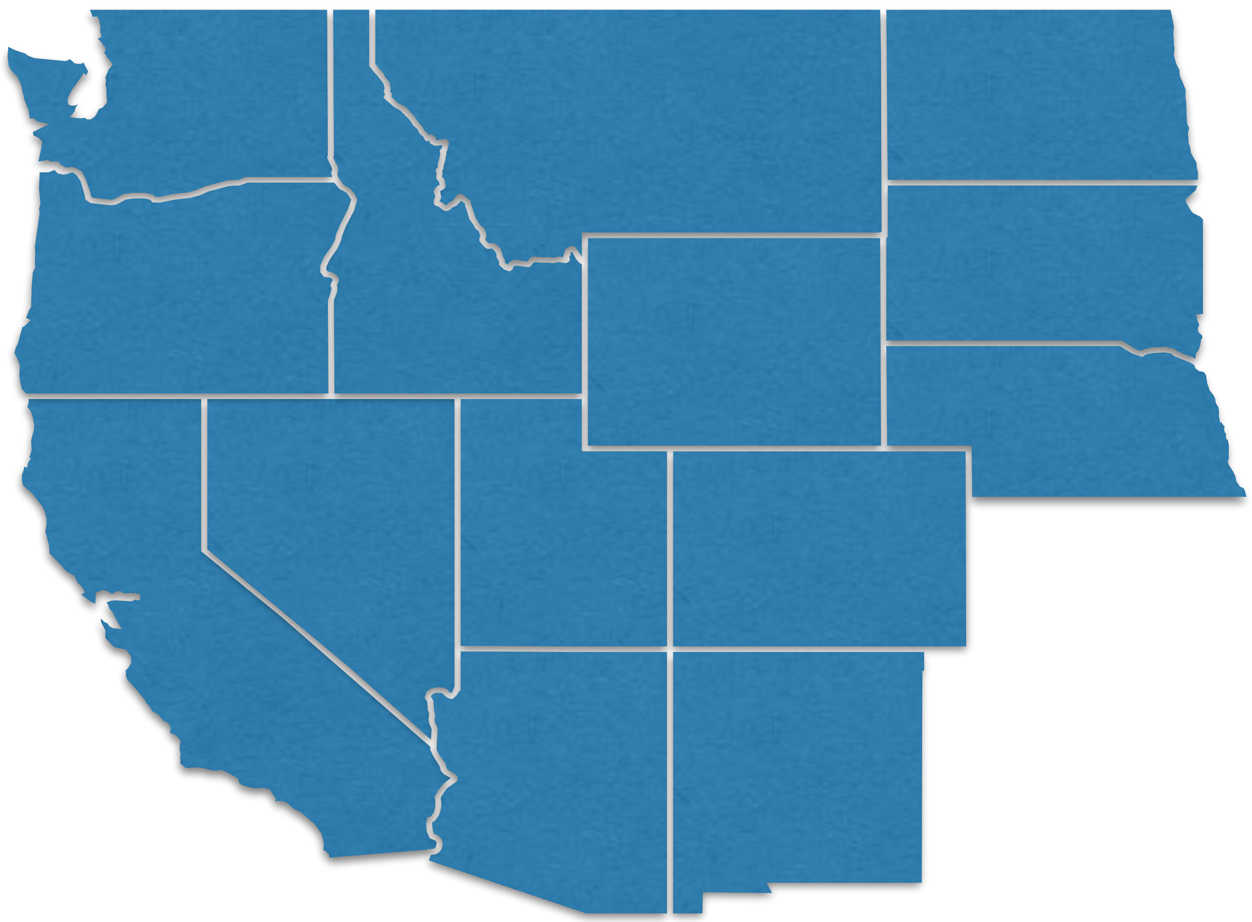
avoid overfitting

available data (sample*)

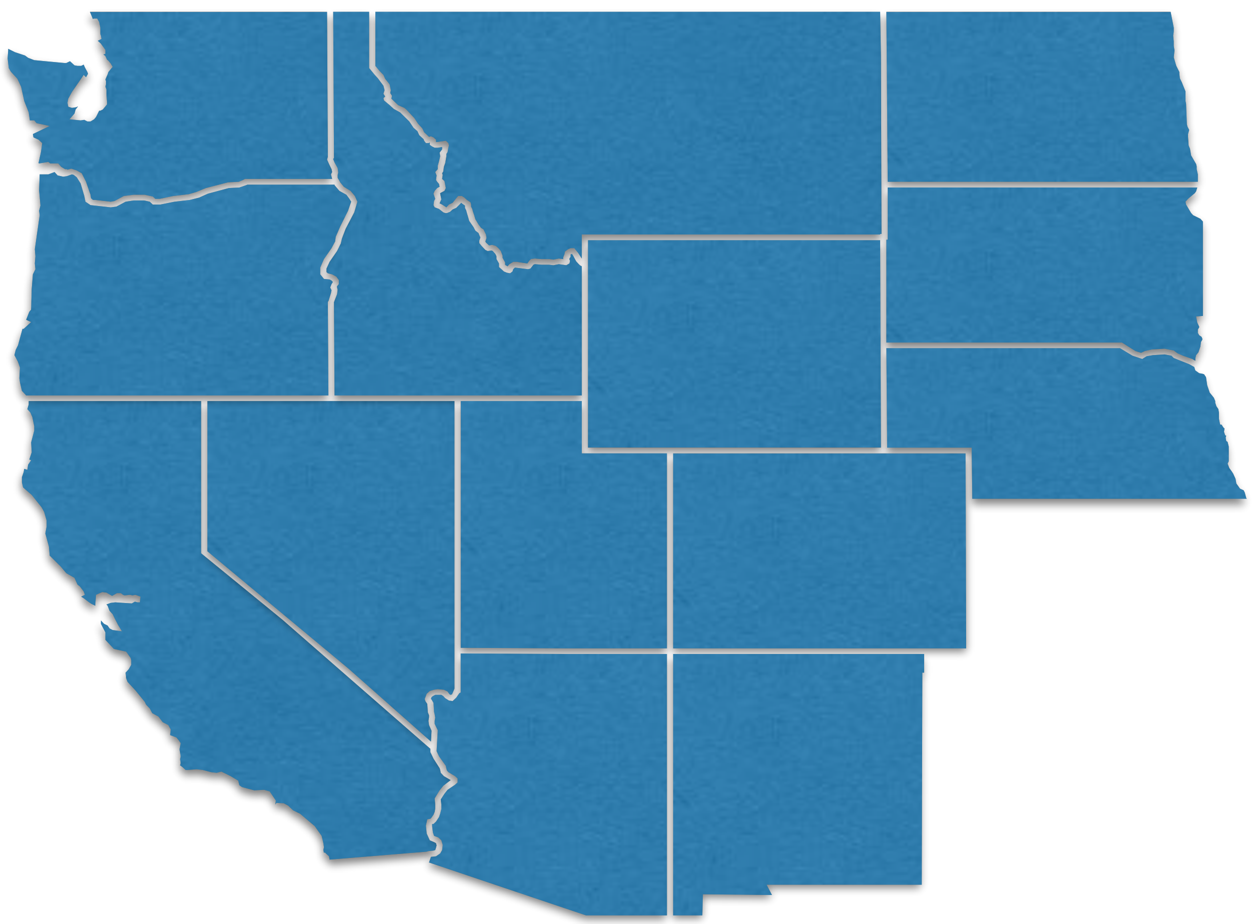
desired: accuracy on **unseen** data (population*)

*has a statistical definition

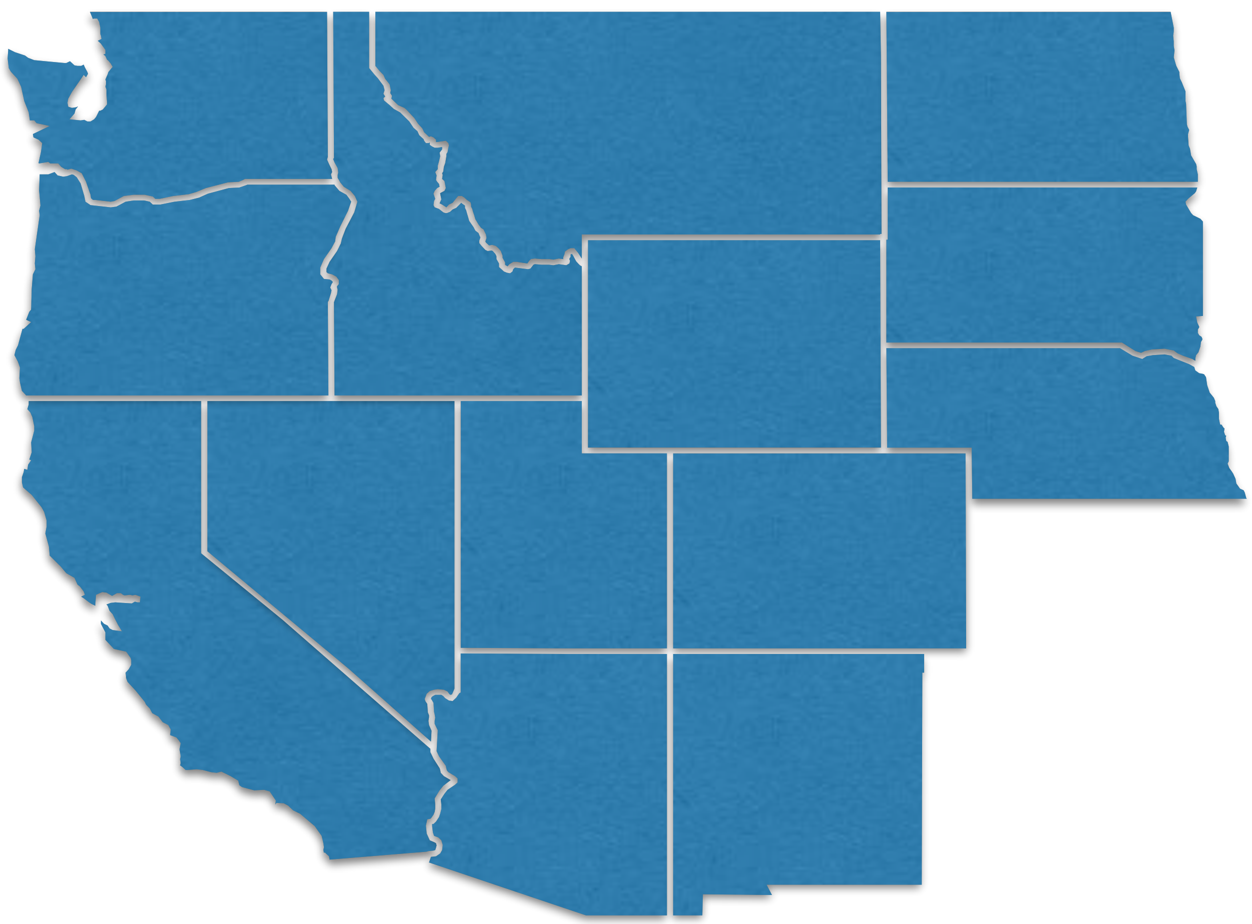
CV helps quantify generalizability



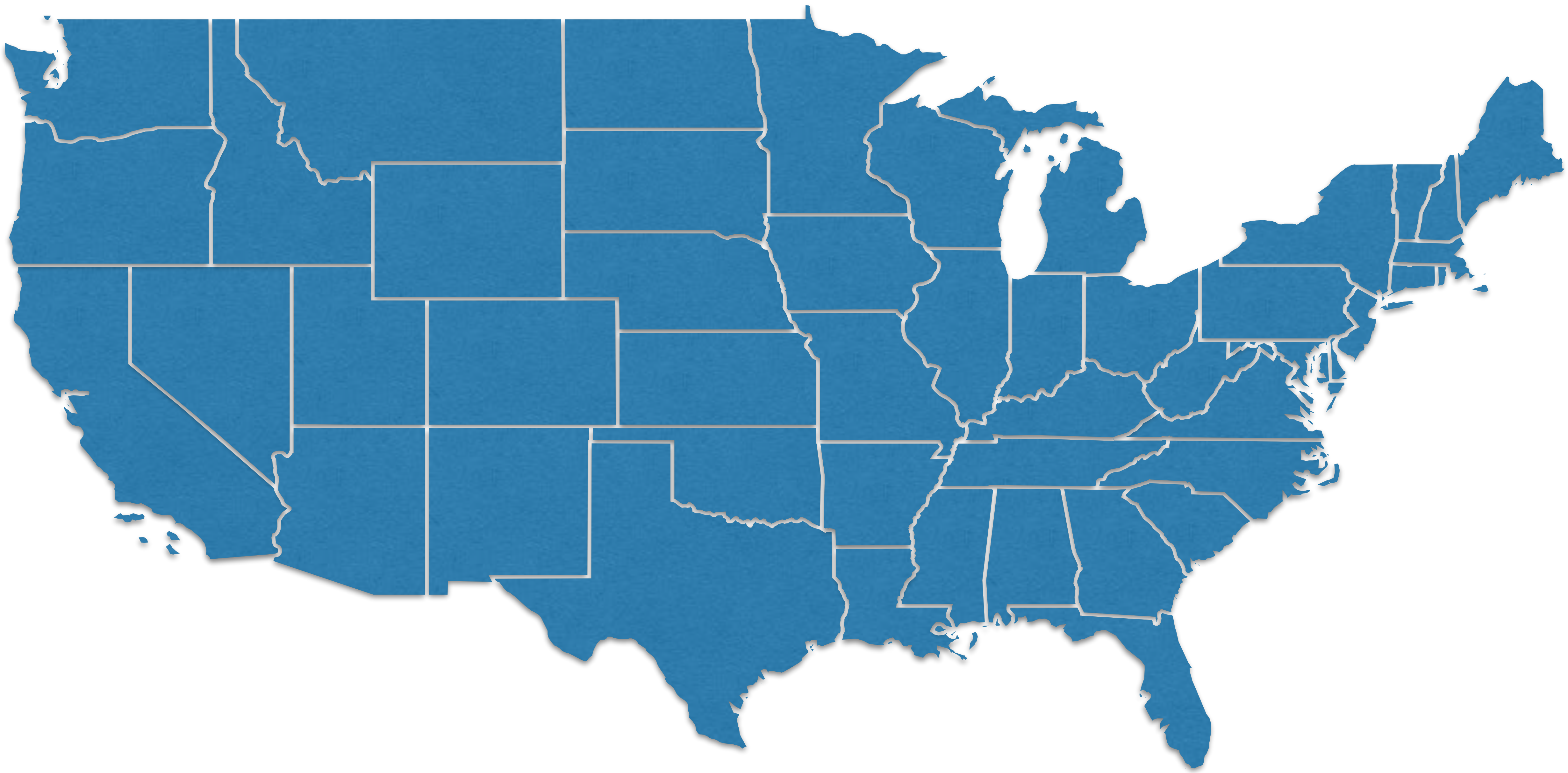
CV helps quantify generalizability



CV helps quantify generalizability



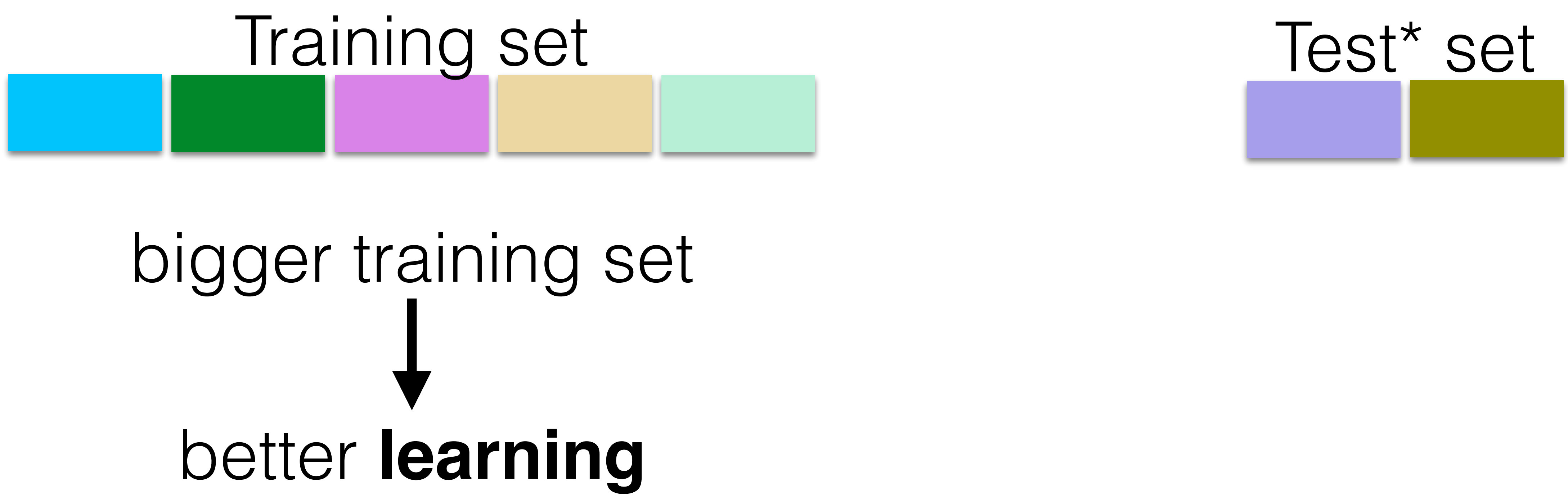
CV helps quantify generalizability



Why cross-validate?

test set will later be specified as tuning / reporting set

Why cross-validate?

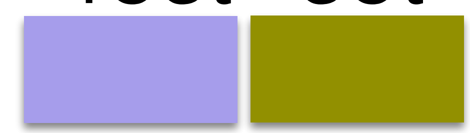


test set will later be specified as tuning / reporting set

Why cross-validate?

Training set

Test* set



bigger training set

↓

better **learning**

bigger test set

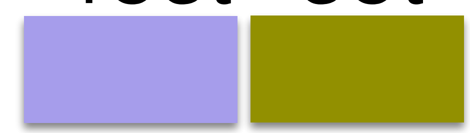
↓

better **evaluation**

Why cross-validate?

Training set

Test* set



bigger training set → better **learning**

bigger test set → better **evaluation**

Key: Train & test sets must be **disjoint.**

test set will later be specified as tuning / reporting set

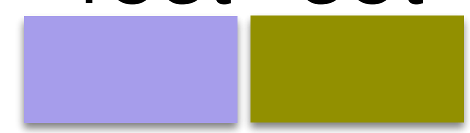
Why cross-validate?

Training set

bigger training set

better **learning**

Test* set



bigger test set

better **evaluation**

Key: Train & test sets must be **disjoint.**
And the dataset or sample size is fixed.

test set will later be specified as tuning / reporting set

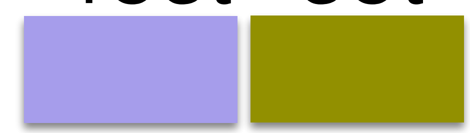
Why cross-validate?

Training set

bigger training set

better **learning**

Test* set



bigger test set

better **evaluation**

Key: Train & test sets must be **disjoint.**
And the dataset or sample size is fixed.
They grow at the expense of each other!

test set will later be specified as tuning / reporting set

Why cross-validate?

bigger training set → better **learning**

bigger test set → better **evaluation**

Key: Train & test sets must be **disjoint.**
And the dataset or sample size is fixed.
They grow at the expense of each other!

test set will later be specified as tuning / reporting set

Why cross-validate?

bigger training set → better **learning**

bigger test set → better **evaluation**

Key: Train & test sets must be **disjoint.**
And the dataset or sample size is fixed.
They grow at the expense of each other!

⇒ **cross**-validate to maximize both

test set will later be specified as tuning / reporting set

Use cases for CV

Use cases for CV

- "When setting aside data for parameter estimation and validation of results can not be afforded, cross-validation (CV) is typically used"

Use cases for CV

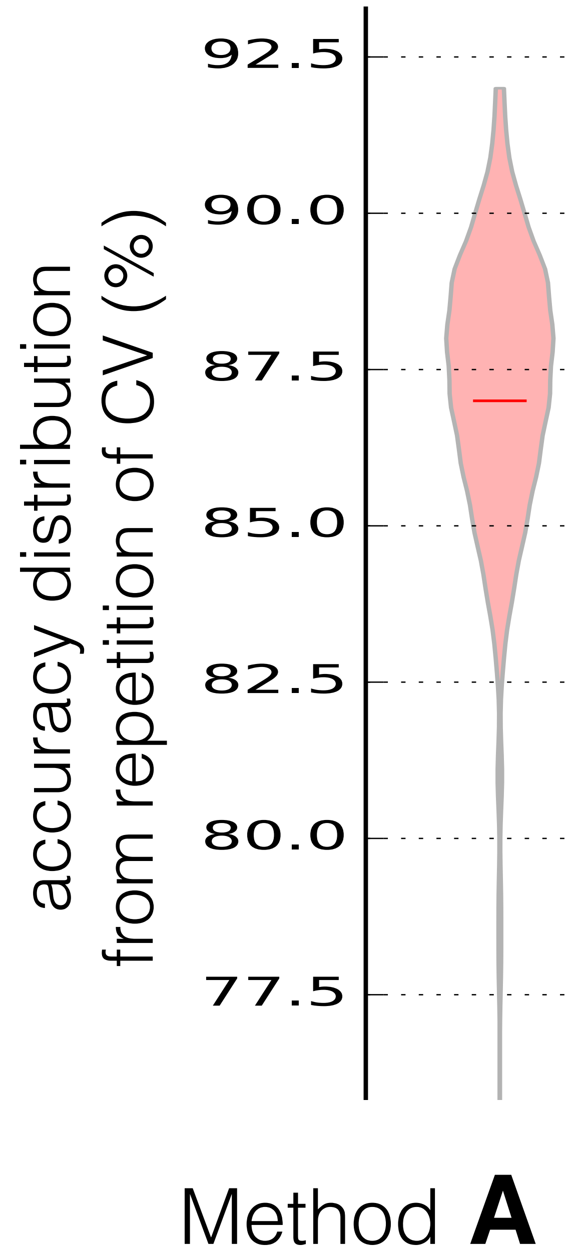
- "When setting aside data for parameter estimation and validation of results can not be afforded, cross-validation (CV) is typically used"
- Use cases:

Use cases for CV

- "When setting aside data for parameter estimation and validation of results can not be afforded, cross-validation (CV) is typically used"

- Use cases:

    - to estimate generalizability (reporting accuracy)



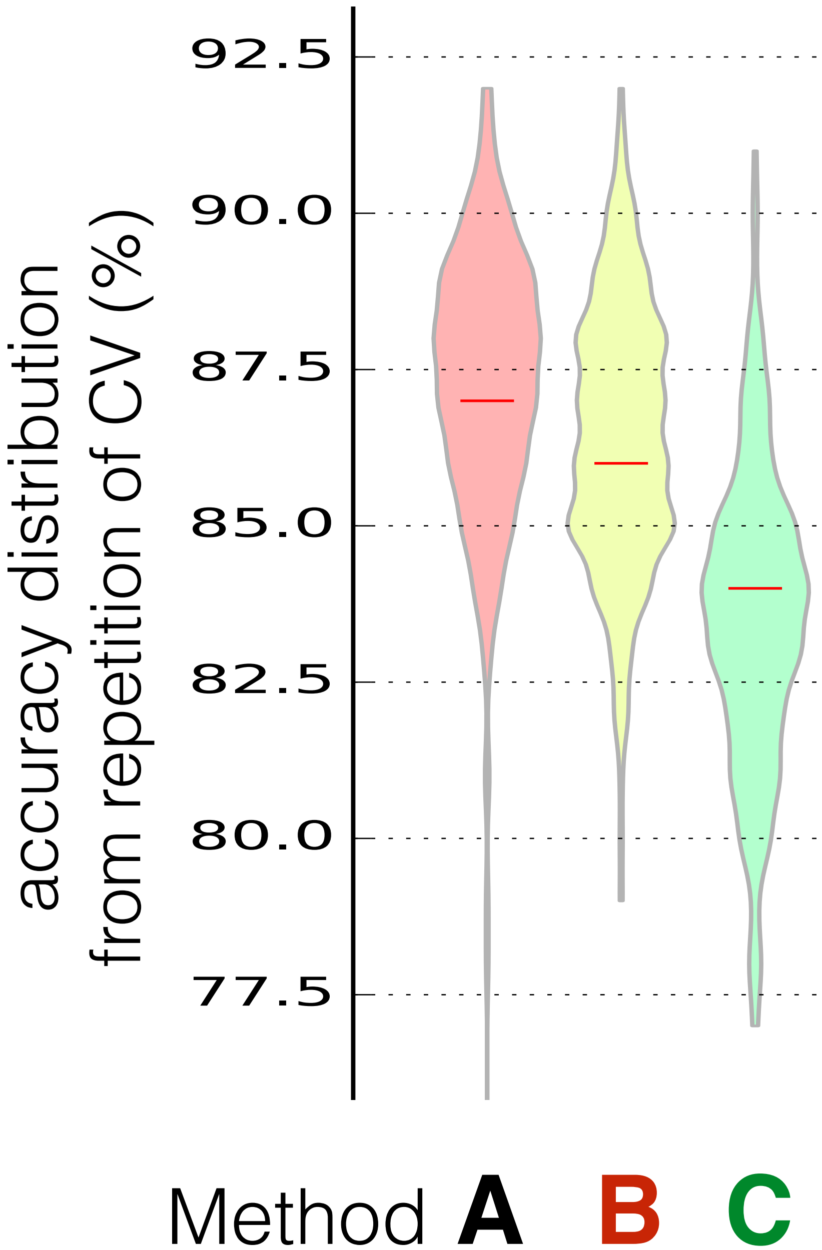
Use cases for CV

- "When setting aside data for parameter estimation and validation of results can not be afforded, cross-validation (CV) is typically used"

- Use cases:

    - to estimate generalizability (reporting accuracy)

    - to pick optimal parameters (model selection)



Use cases for CV

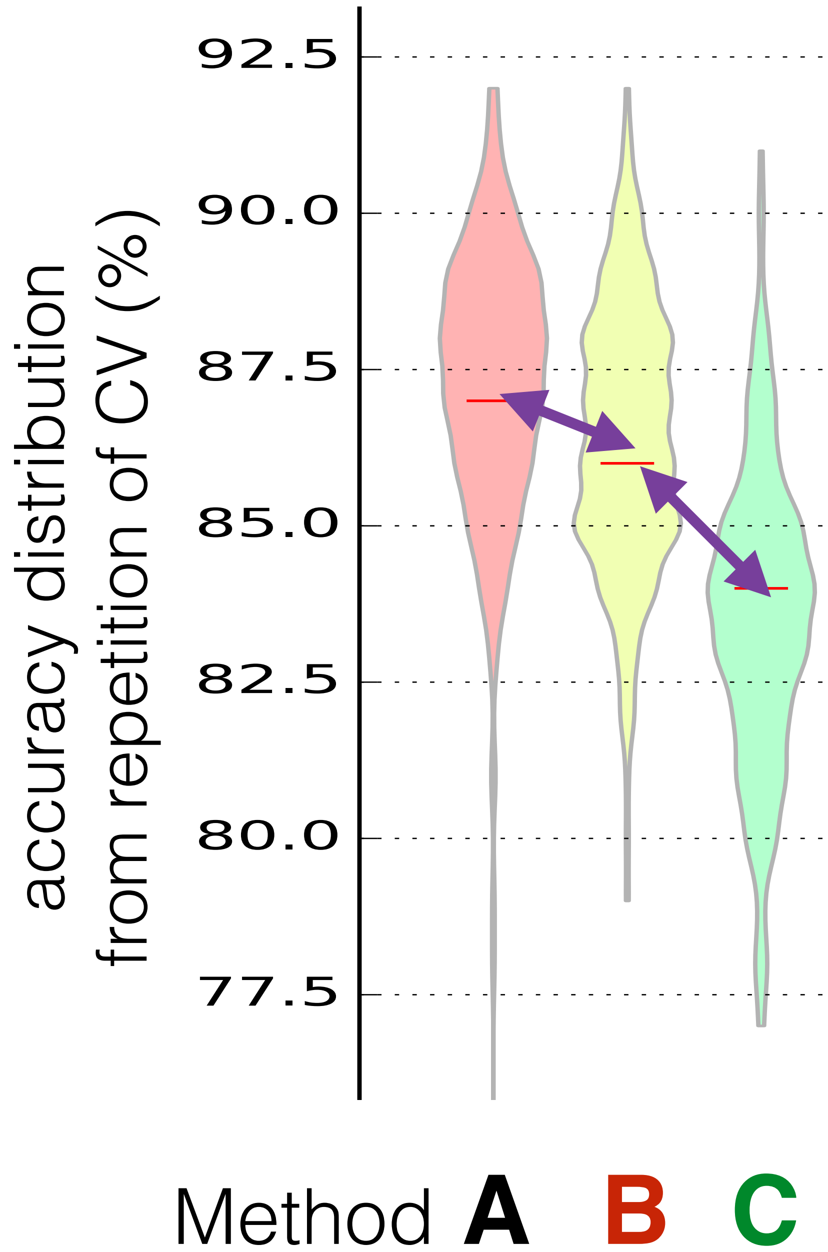
- "When setting aside data for parameter estimation and validation of results can not be afforded, cross-validation (CV) is typically used"

- Use cases:

    - to estimate generalizability (reporting accuracy)

    - to pick optimal parameters (model selection)

    - to compare performance (model comparison).



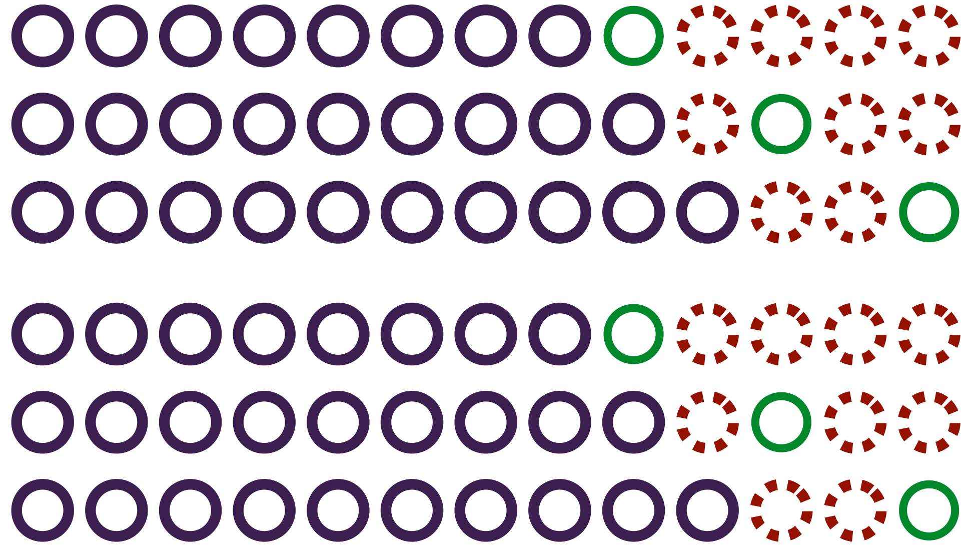
Key aspects of CV

Key aspects of CV

1. **How you split** the dataset into train/test

Key aspects of CV

1. **How you split** the dataset into train/test
    - maximizing independence between training and test sets



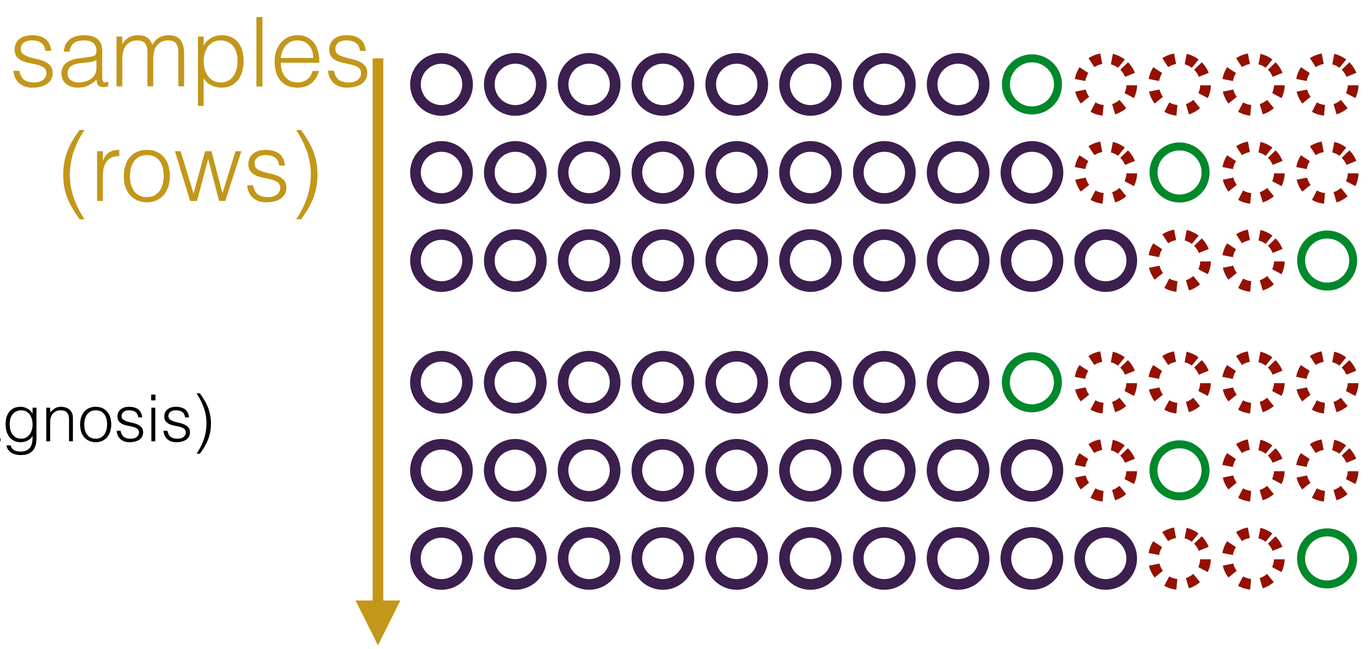
Key aspects of CV

1. **How you split** the dataset into train/test

    - maximizing independence between training and test sets

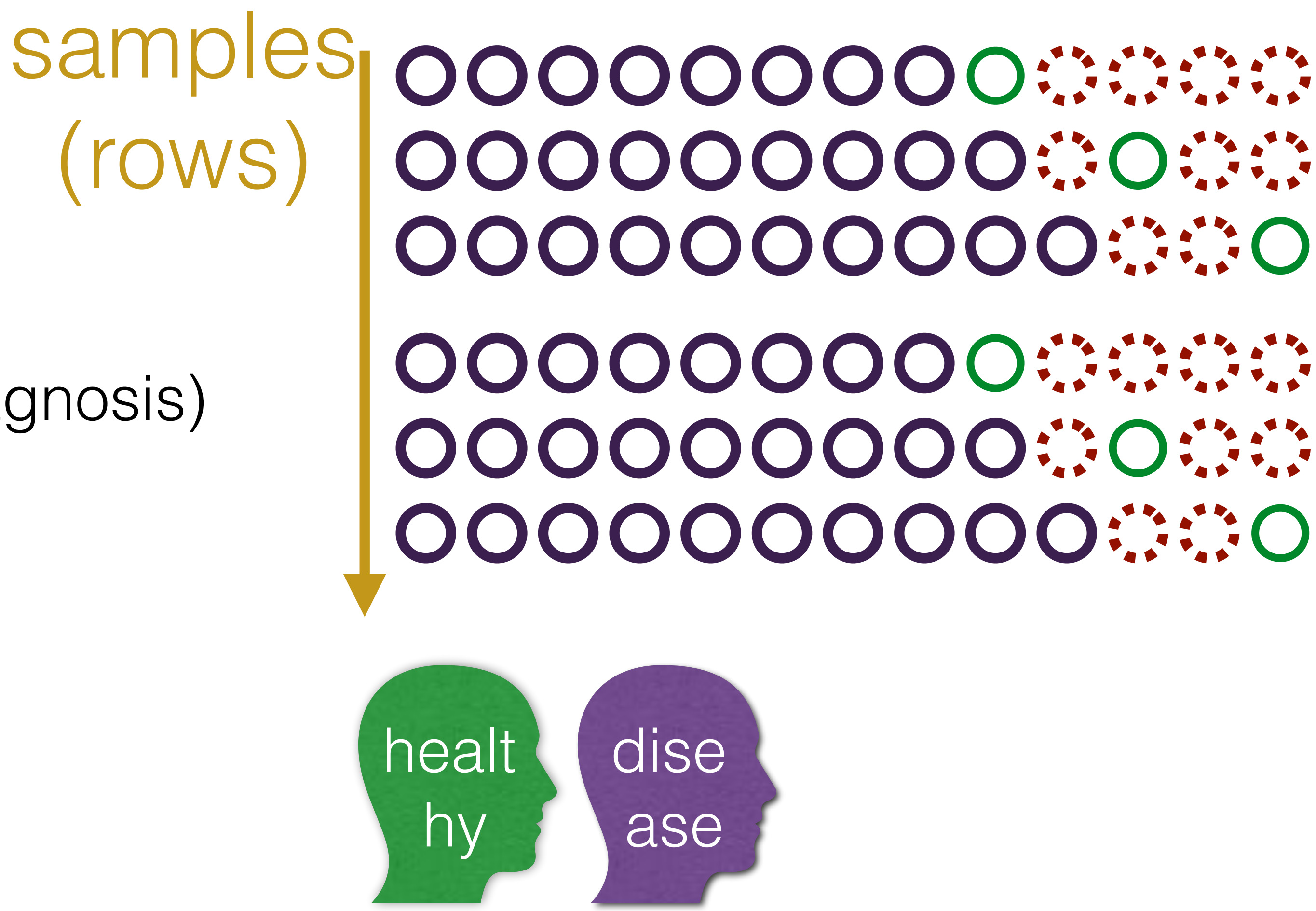
    - the split could be

        - over samples (e.g. indiv. diagnosis)



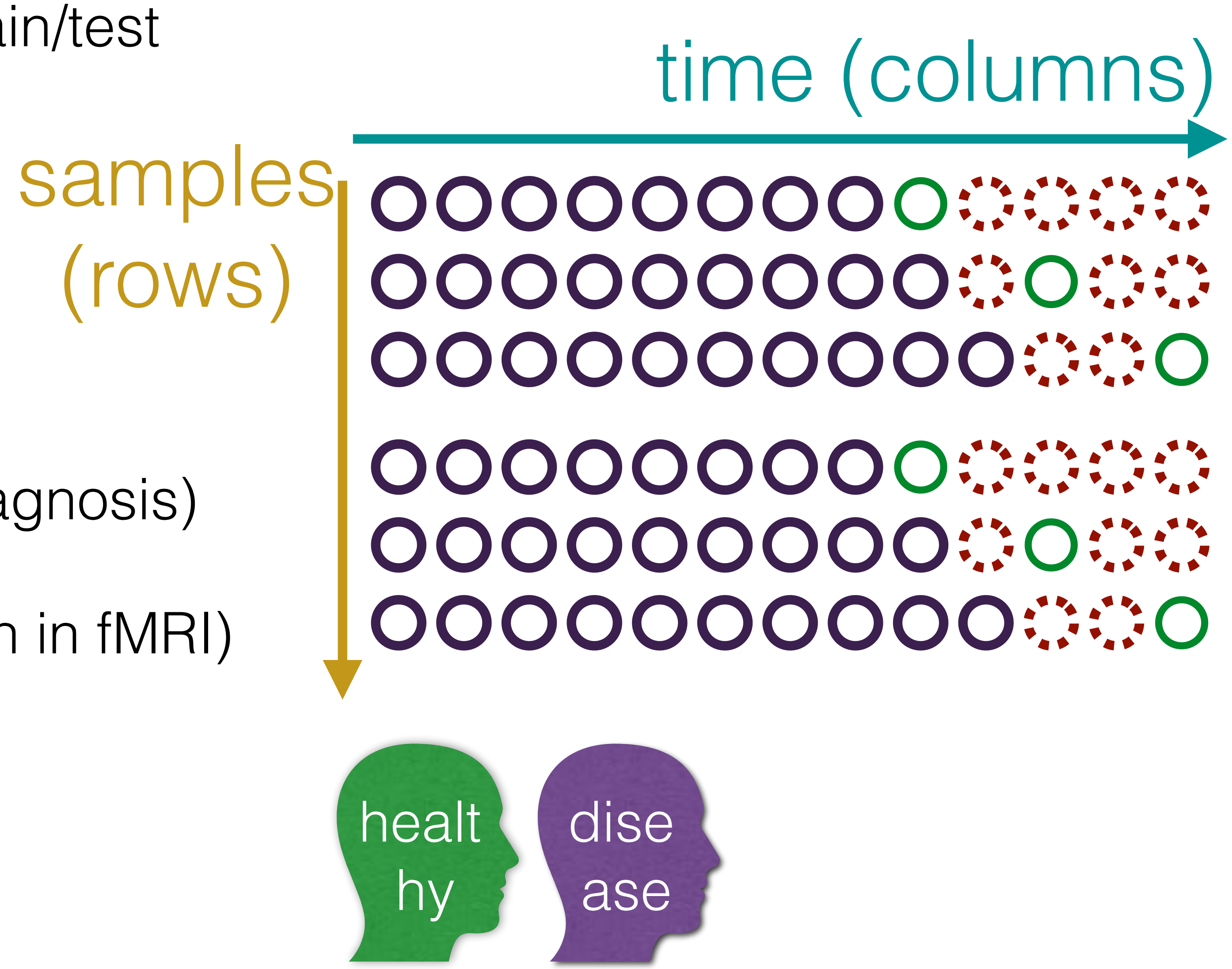
Key aspects of CV

1. **How you split** the dataset into train/test
   - maximizing independence between training and test sets
   - the split could be
     - over samples (e.g. indiv. diagnosis)



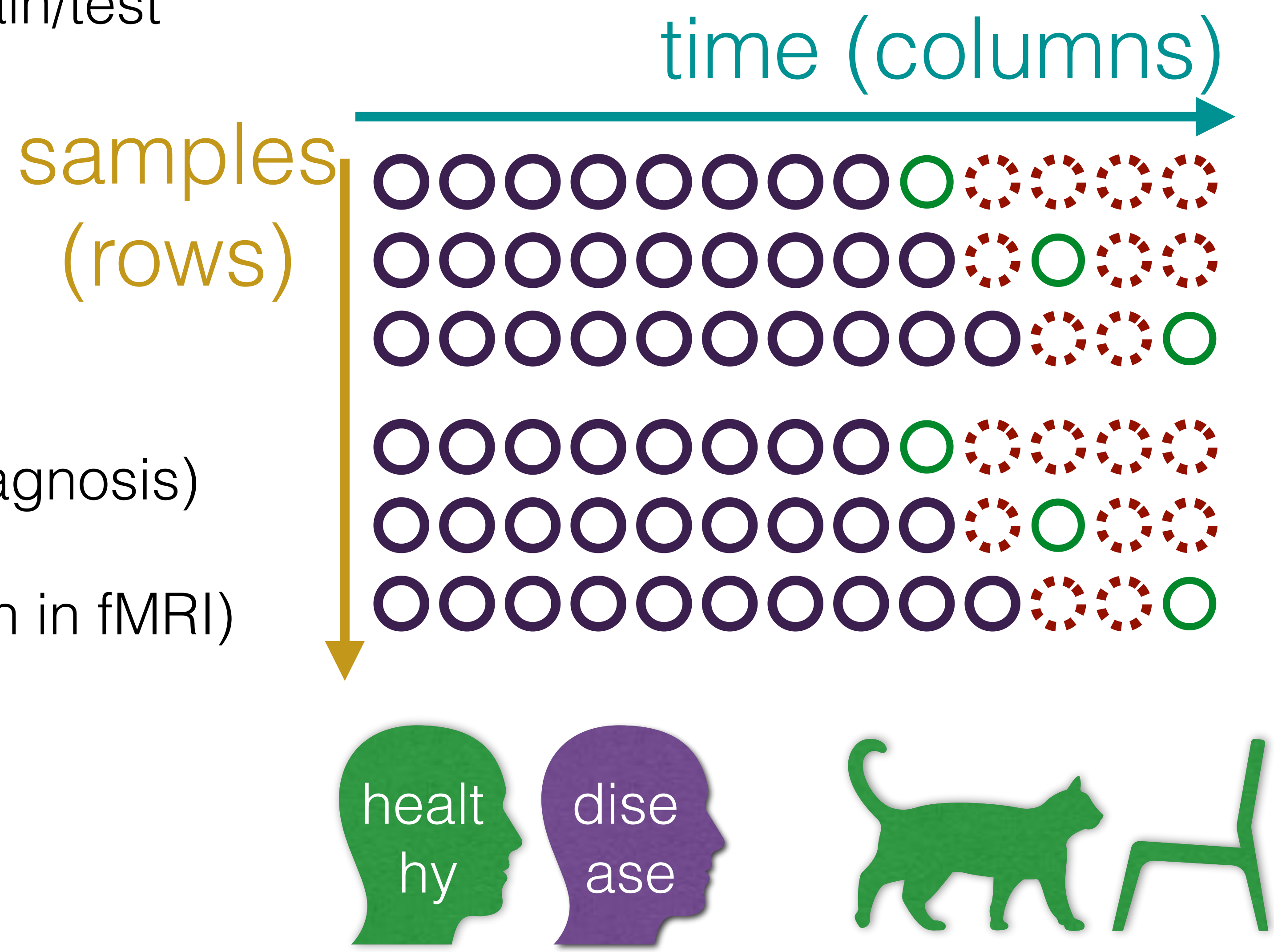
Key aspects of CV

1. **How you split** the dataset into train/test
   - maximizing independence between training and test sets
   - the split could be
     - over samples (e.g. indiv. diagnosis)
     - over time (for task prediction in fMRI)



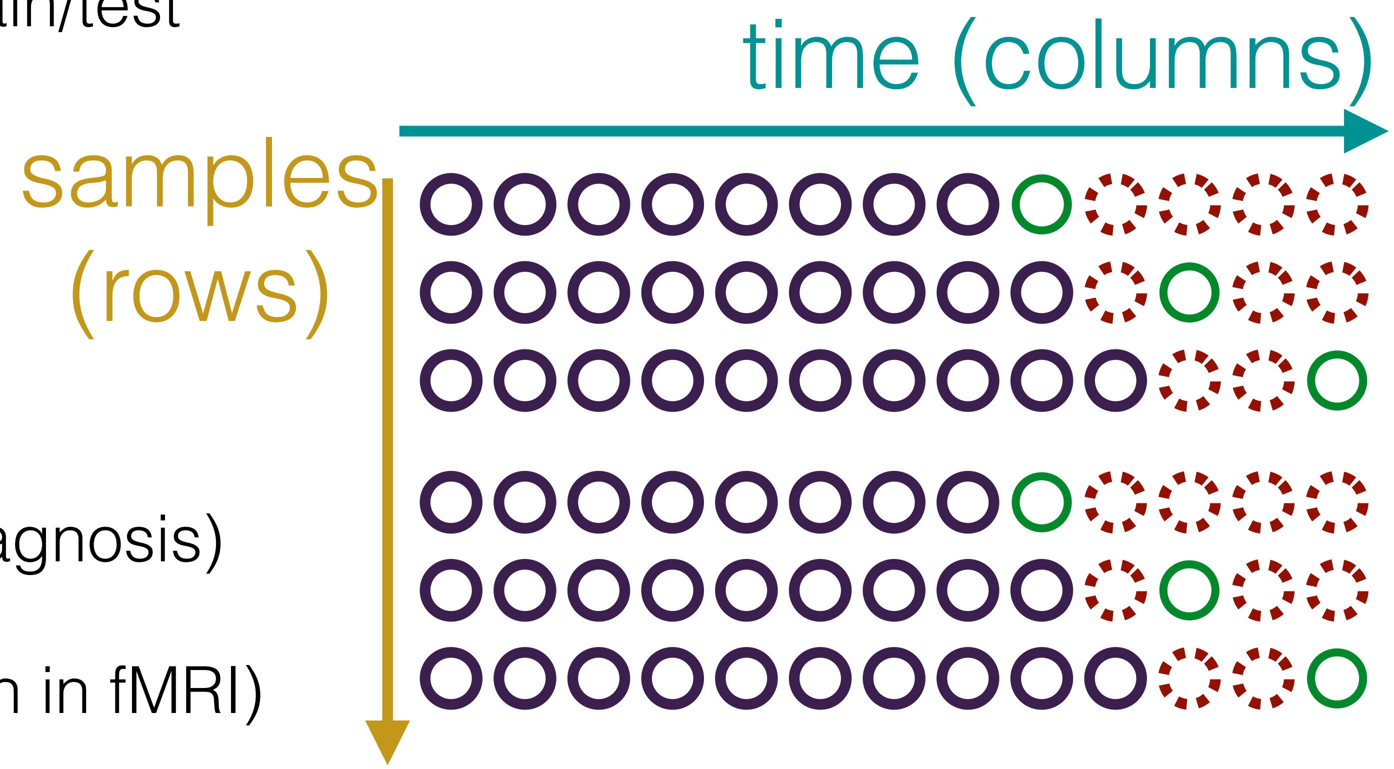
Key aspects of CV

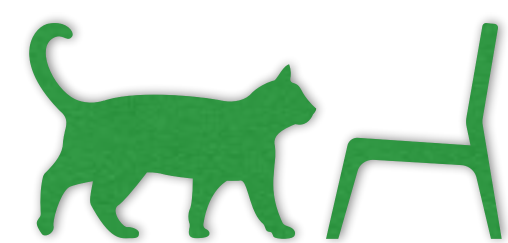
1. **How you split** the dataset into train/test
   - maximizing independence between training and test sets
   - the split could be
     - over samples (e.g. indiv. diagnosis)
     - over time (for task prediction in fMRI)



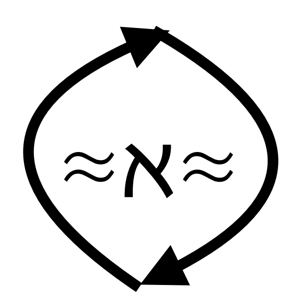
Key aspects of CV

1. **How you split** the dataset into train/test
   - maximizing independence between training and test sets
   - the split could be
     - over samples (e.g. indiv. diagnosis)
     - over time (for task prediction in fMRI)

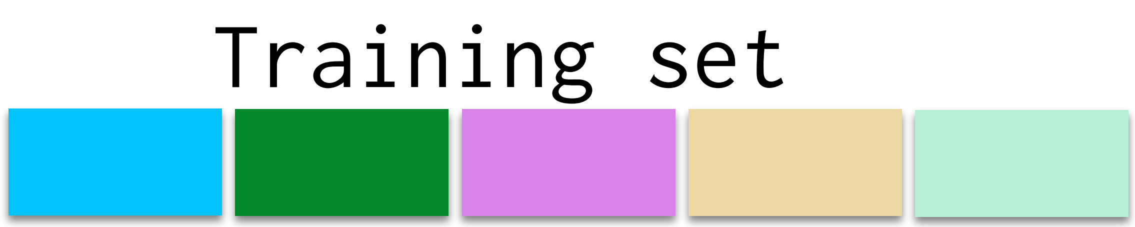




2. **How often you repeat randomized splits?**
   - to expose classifier to full variability
   - as many as times as you can e.g. 100

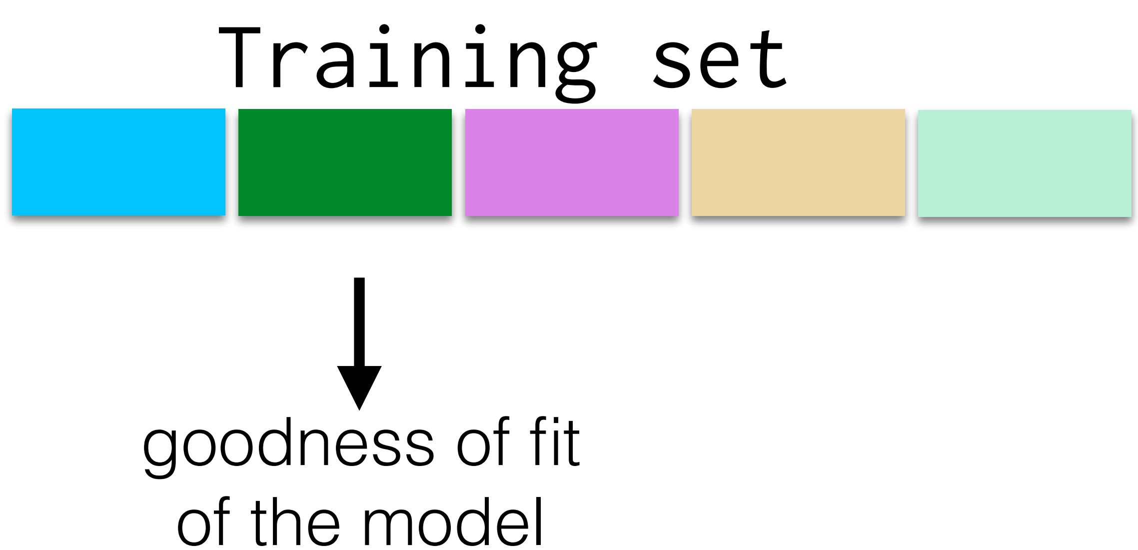


Full setup for CV



*biased towards X —> overfit to X

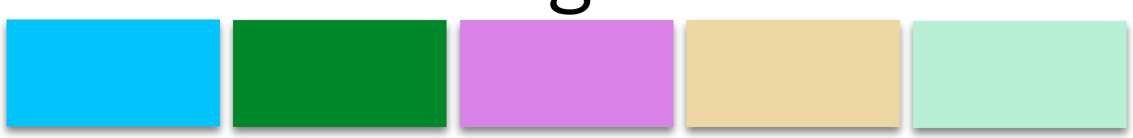
Full setup for CV



*biased towards X —> overfit to X

Full setup for CV

Training set

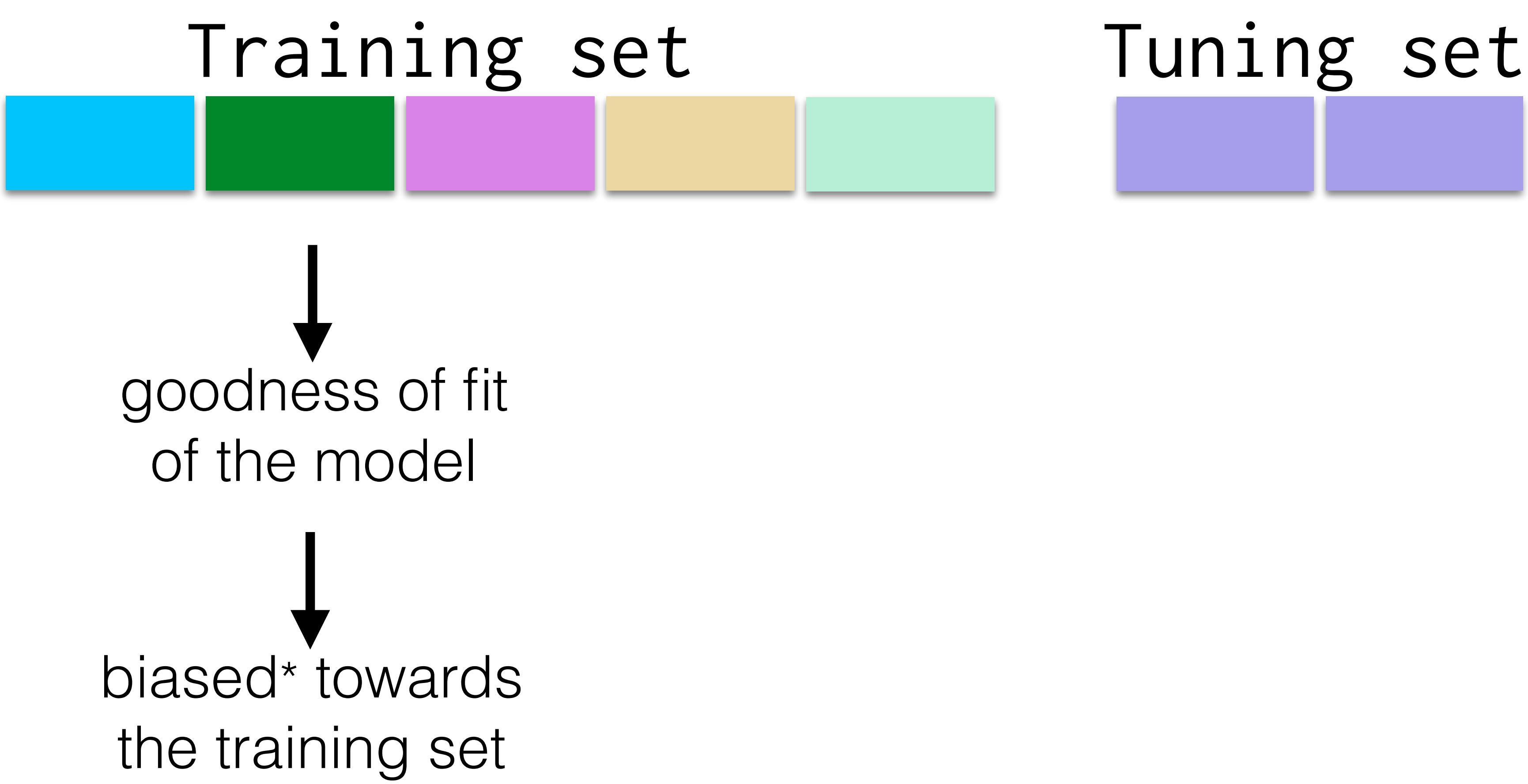


goodness of fit
of the model

biased* towards
the training set

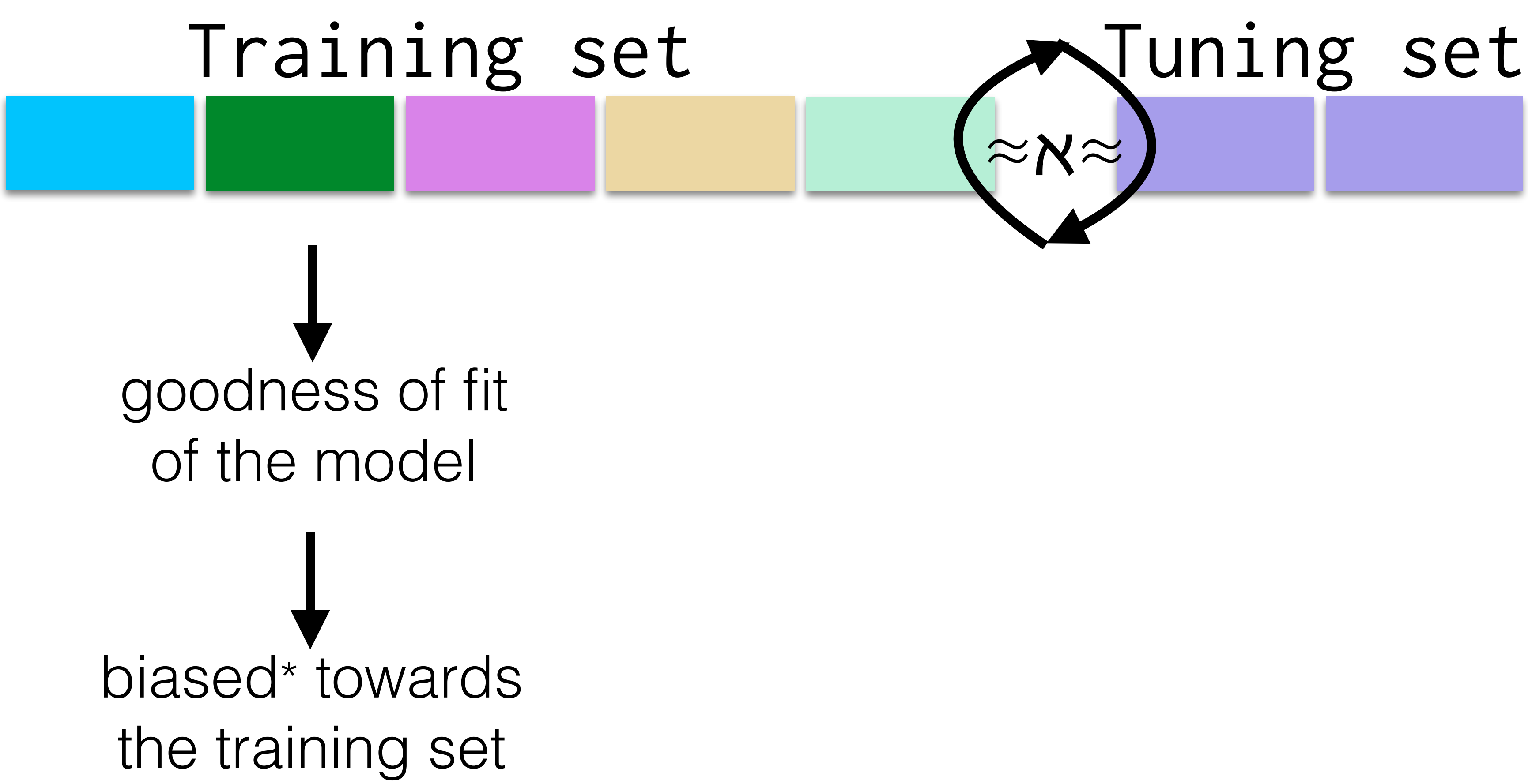
*biased towards X —> overfit to X

Full setup for CV



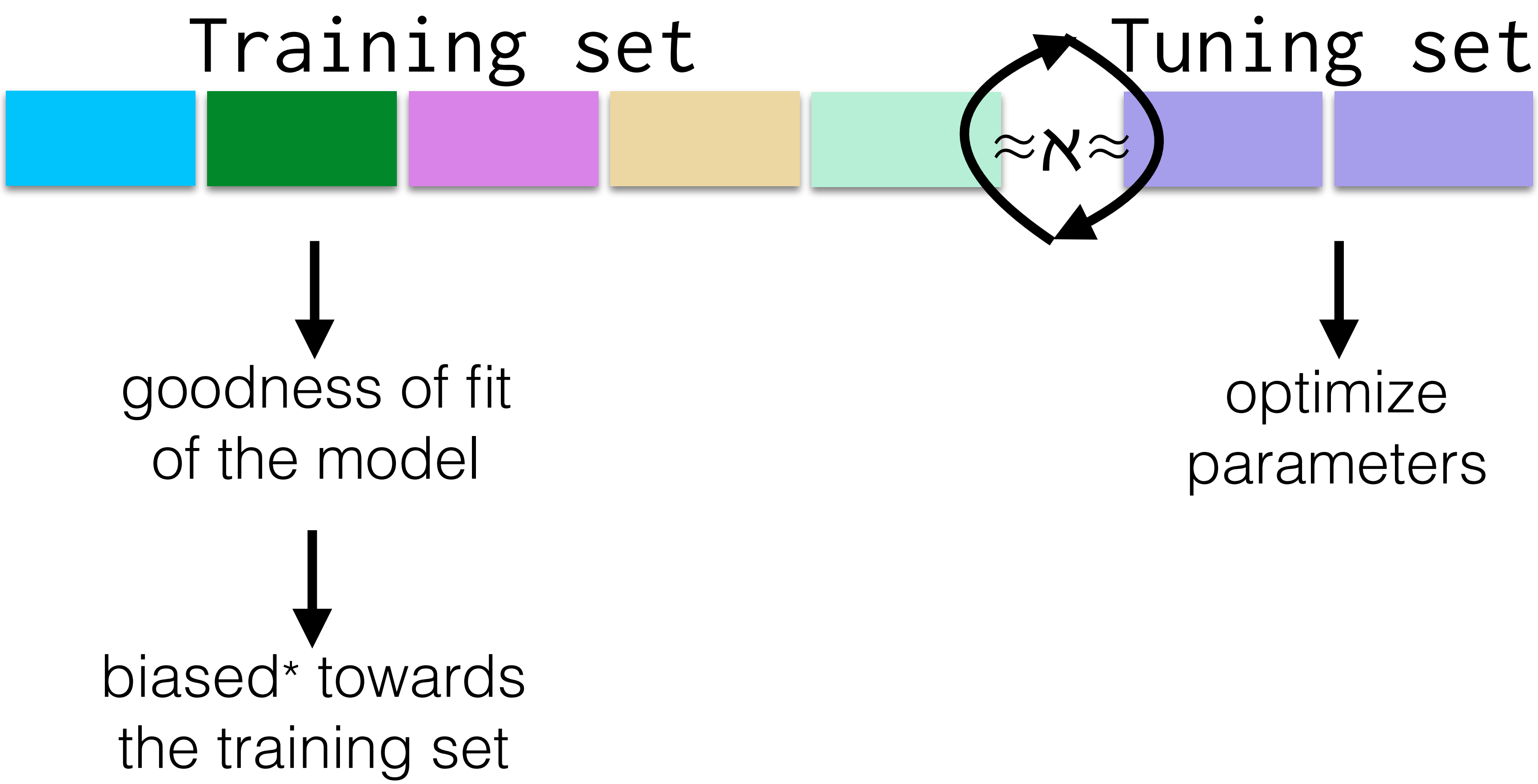
*biased towards X —> overfit to X

Full setup for CV



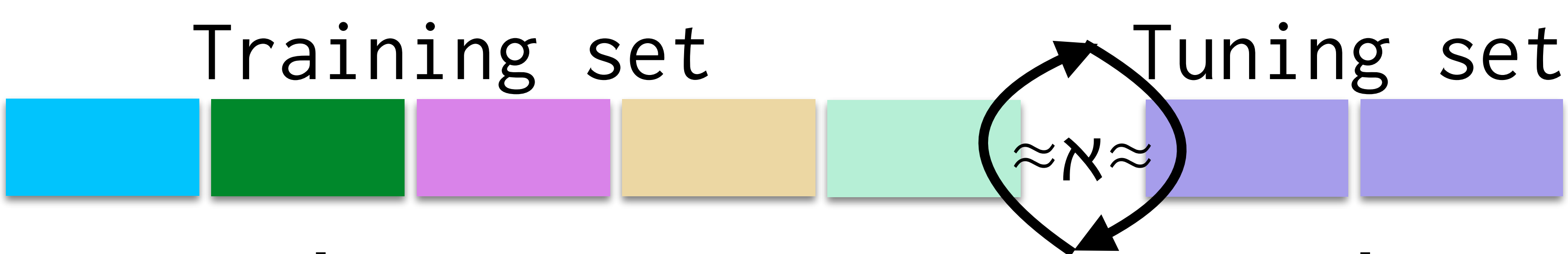
*biased towards X —> overfit to X

Full setup for CV



*biased towards X —> overfit to X

Full setup for CV



goodness of fit
of the model

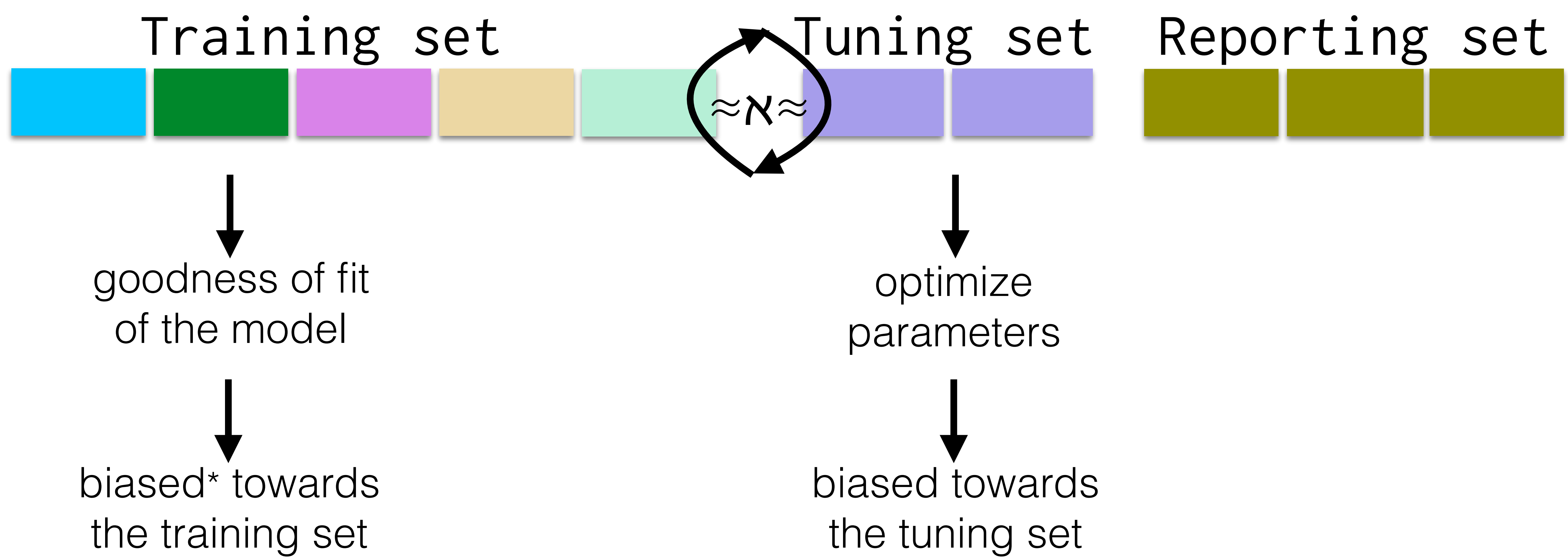
biased* towards
the training set

optimize
parameters

biased towards
the tuning set

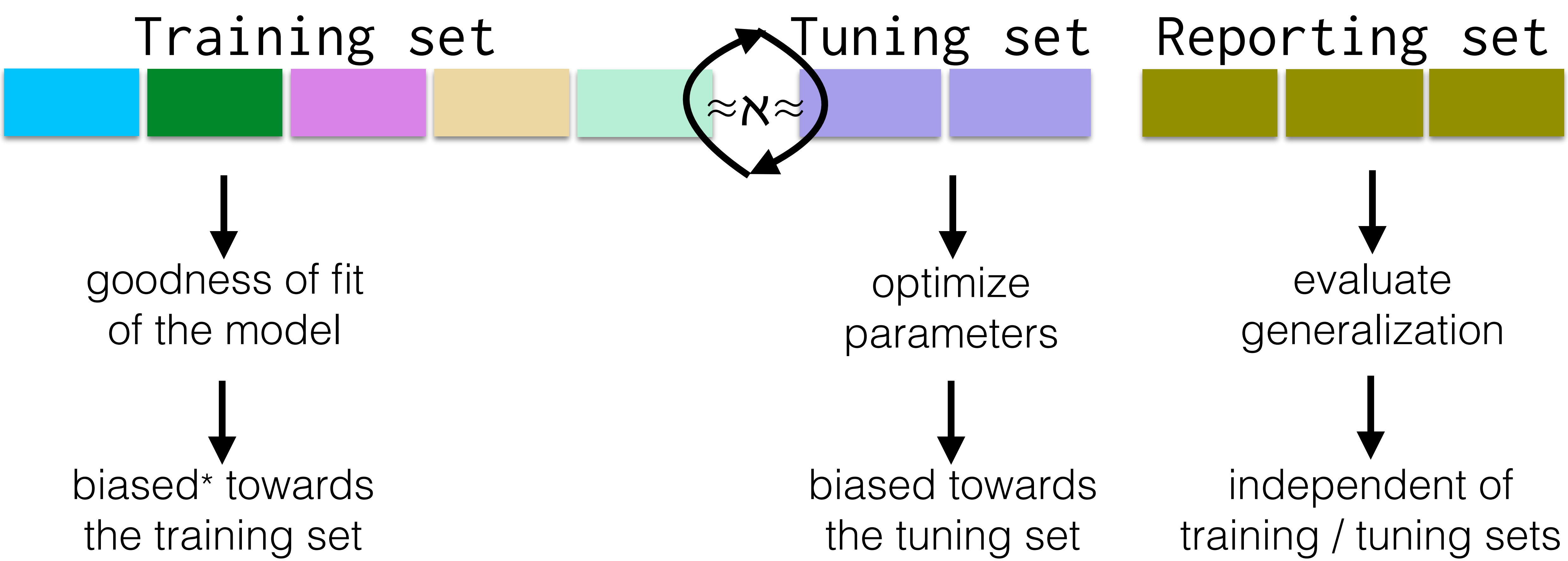
*biased towards X —> overfit to X

Full setup for CV



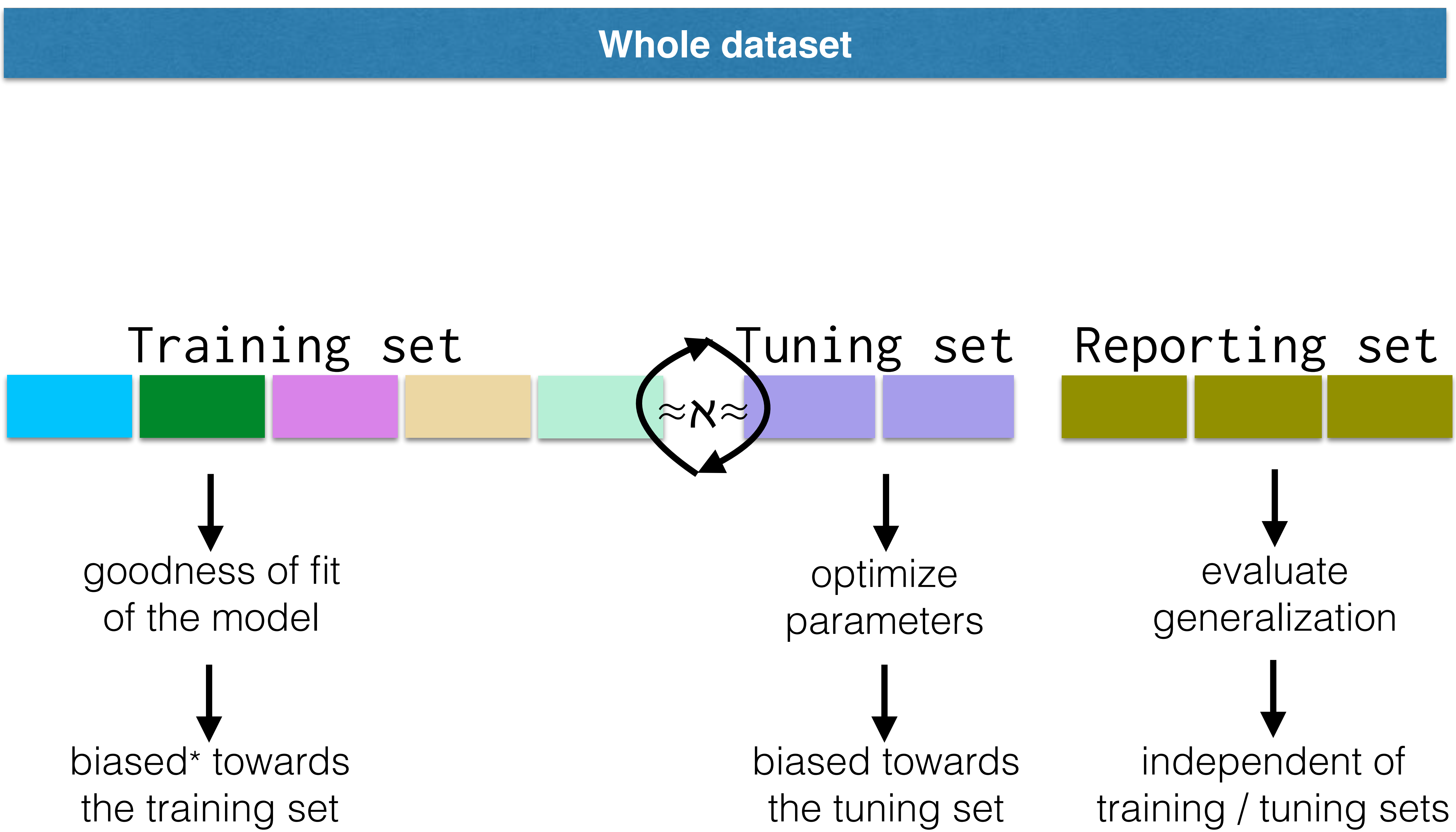
*biased towards X —> overfit to X

Full setup for CV



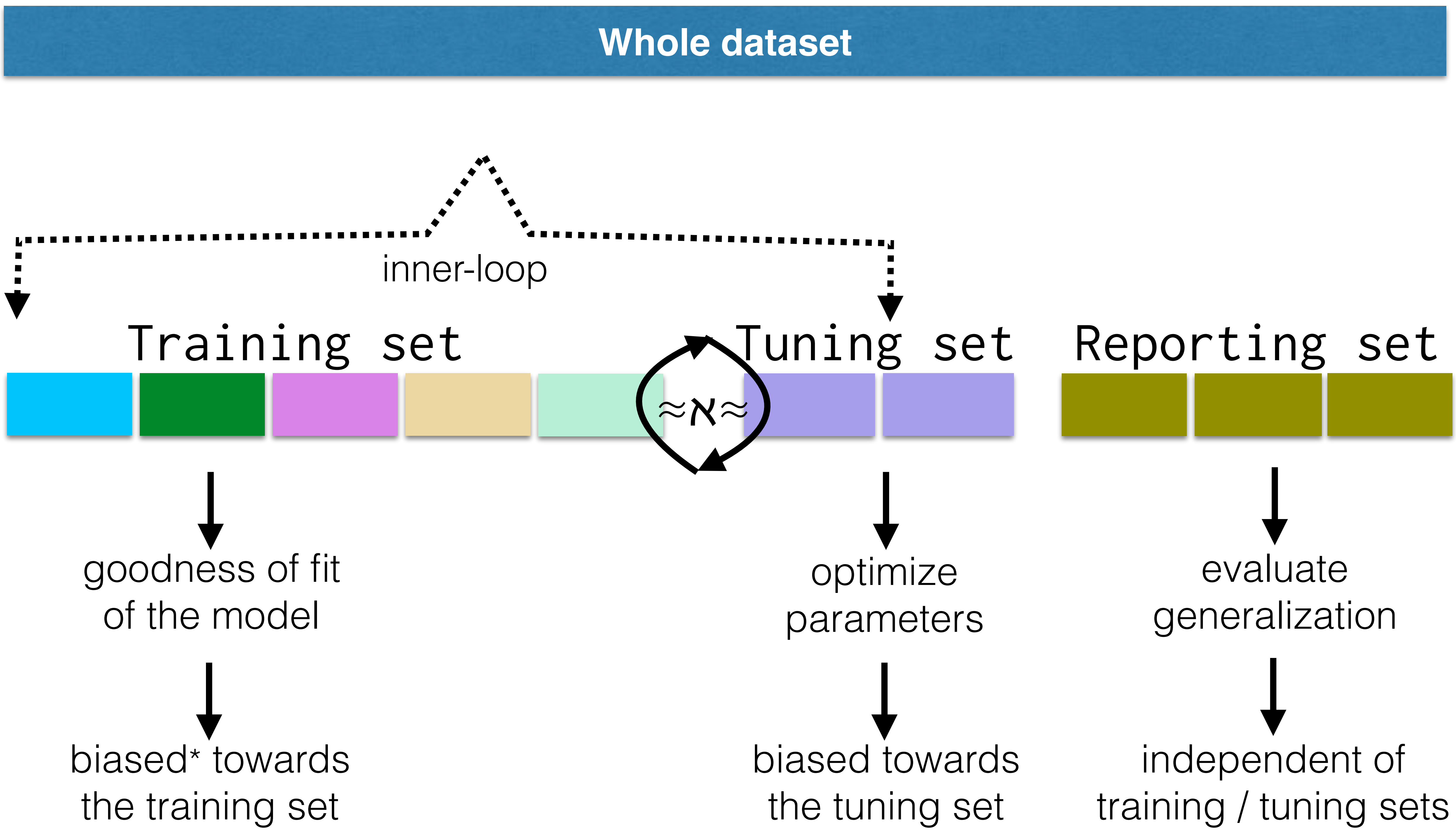
*biased towards X —> overfit to X

Full setup for CV



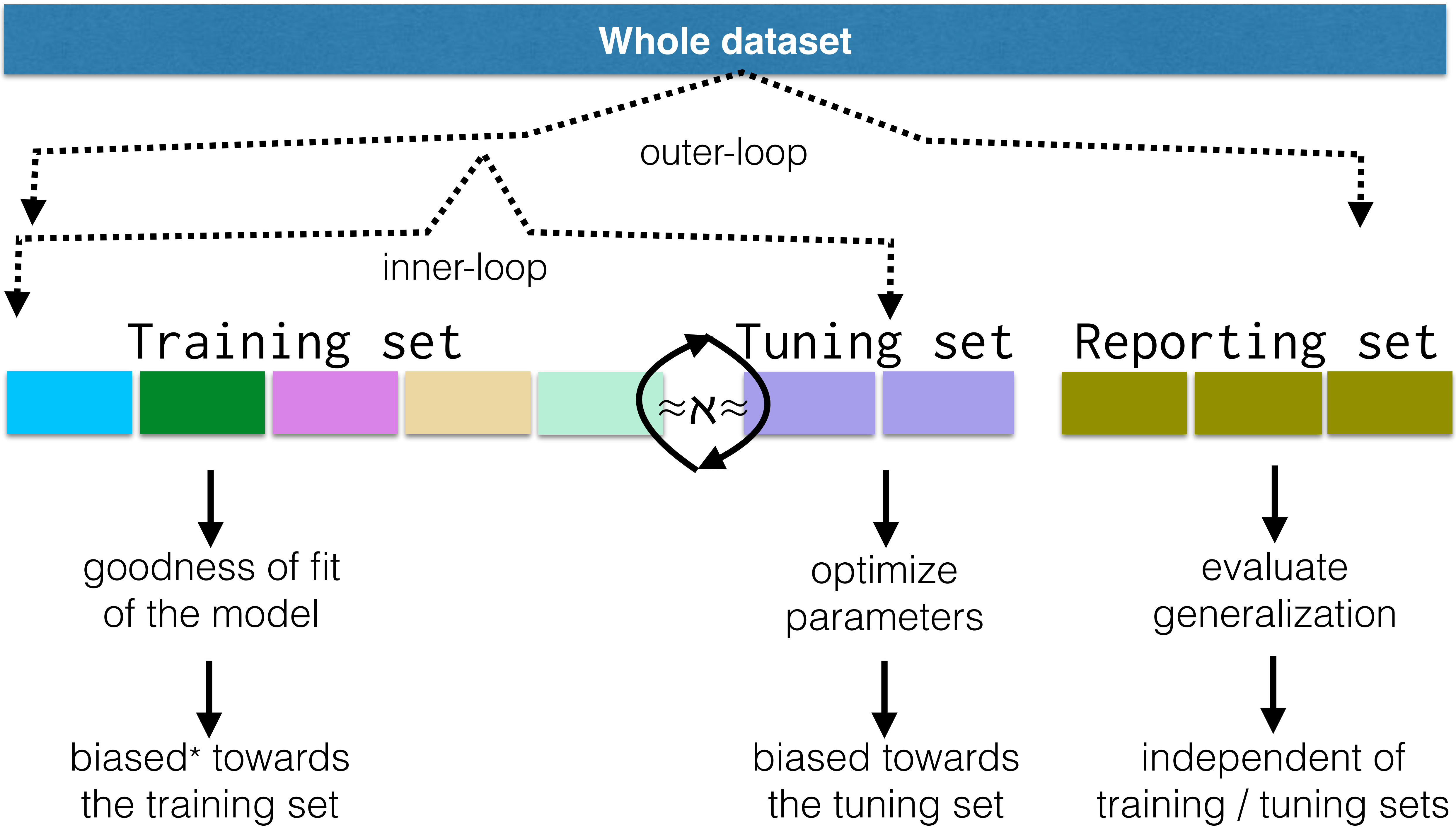
*biased towards X —> overfit to X

Full setup for CV



*biased towards X —> overfit to X

Full setup for CV



*biased towards X —> overfit to X

Do’s and Don’ts with data splits

Do's and Don'ts with data splits

Data split

Training

Tuning

Reporting

Do’s and Don’ts with data splits

Data split	Purpose (Do’s)
Training	Train model to learn its core parameters
Tuning	Optimize hyper-parameters
Reporting	Evaluate fully-optimized classifier to report performance

Do's and Don'ts with data splits

Data split	Purpose (Do's)	Don'ts (Invalid use)
Training	Train model to learn its core parameters	**Don't report training error as the reporting error!**
Tuning	Optimize hyper-parameters	**Don't do feature selection or anything supervised on tuning set to learn or optimize!**
Reporting	Evaluate fully-optimized classifier to report performance	**Don't use it in any way to train classifier or optimize parameters**

Do’s and Don’ts with data splits

Data split	Purpose (Do’s)	Don’ts (Invalid use)	Other names in different domains
Training	Train model to learn its core parameters	**Don’t report training error as the reporting error!**	training (no confusion)
Tuning	Optimize hyper-parameters	**Don’t do feature selection or anything supervised on tuning set to learn or optimize!**	validation / test set (more accurately tuning set)
Reporting	Evaluate fully-optimized classifier to report performance	**Don’t use it in any way to train classifier or optimize parameters**	test / validation set (more accurately reporting set)

Do’s and Don’ts with data splits

Data split	Purpose (Do’s)	Don’ts (Invalid use)	Other names in different domains
Training	Train model to learn its core parameters	**Don’t report training error as the reporting error!**	training (no confusion)
Tuning	Optimize hyper-parameters	**Don’t do feature selection or anything supervised on tuning set to learn or optimize!**	validation / test set (more accurately tuning set)
Reporting	Evaluate fully-optimized classifier to report performance	**Don’t use it in any way to train classifier or optimize parameters**	test / validation set (more accurately reporting set)

Note: the term “test set” is often used to loosely refer to a split different from training set!

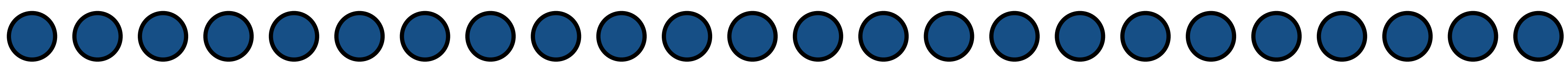
Do’s and Don’ts with data splits

Data split	Purpose (Do’s)	Don’ts (Invalid use)	Other names in different domains
Training	Train model to learn its core parameters	**Don’t report training error as the reporting error!**	training (no confusion)
Tuning	Optimize hyper-parameters	**Don’t do feature selection or anything supervised on tuning set to learn or optimize!**	validation / test set (more accurately tuning set)
Reporting	Evaluate fully-optimized classifier to report performance	**Don’t use it in any way to train classifier or optimize parameters**	test / validation set (more accurately reporting set)

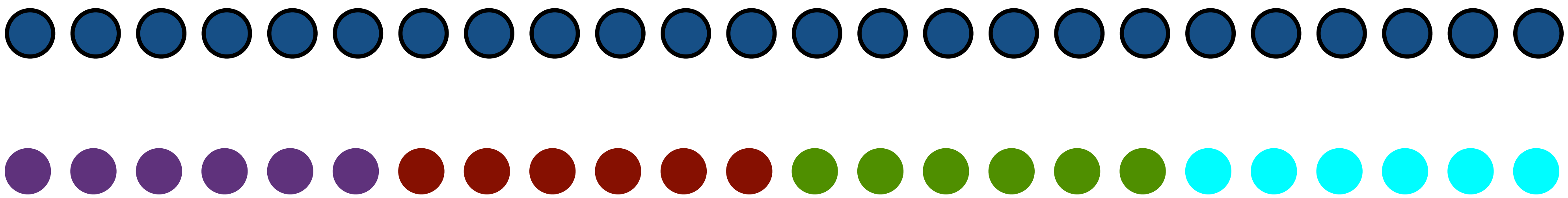
Note: the term “test set” is often used to loosely refer to a split different from training set!

And the term “training set” absorbs **both** training and tuning sets!

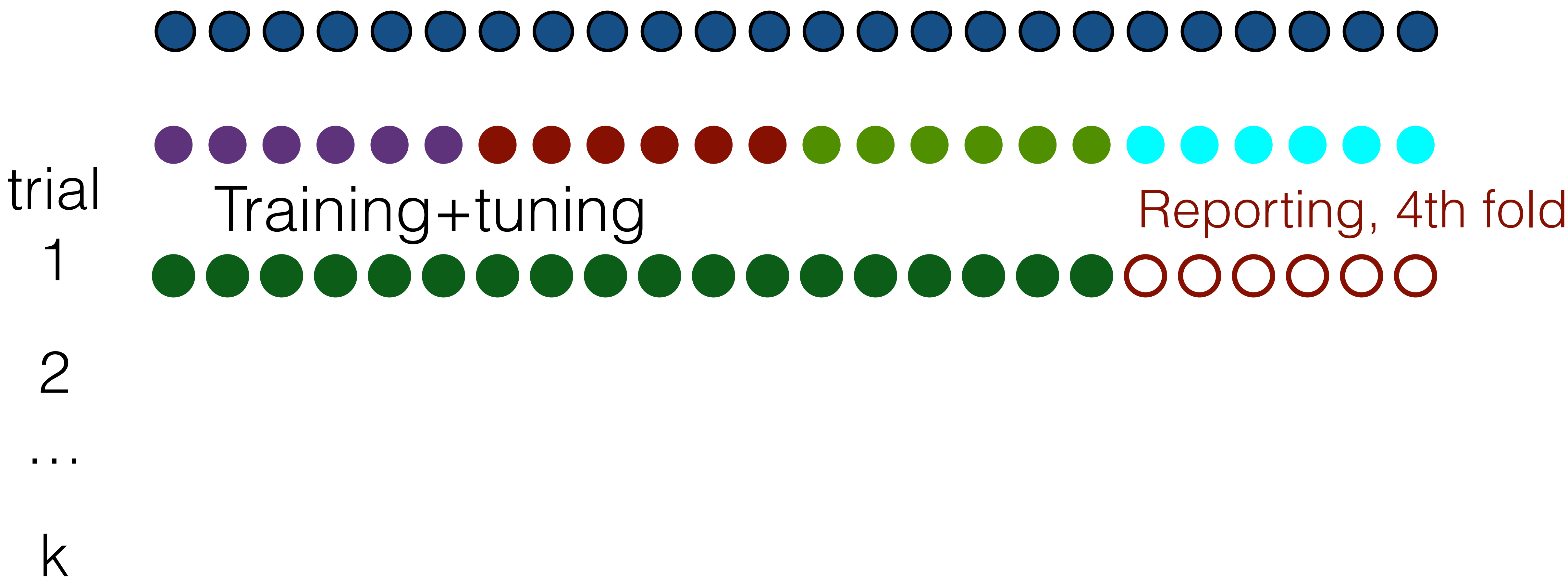
K-fold CV



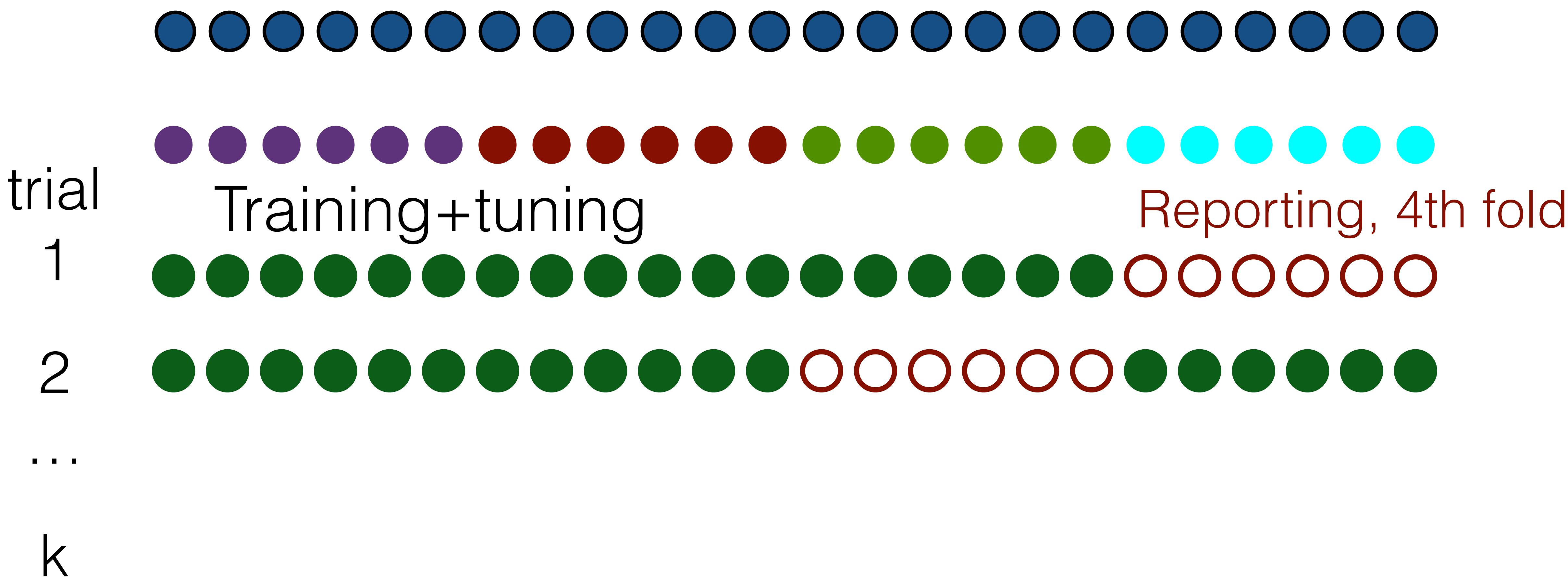
K-fold CV



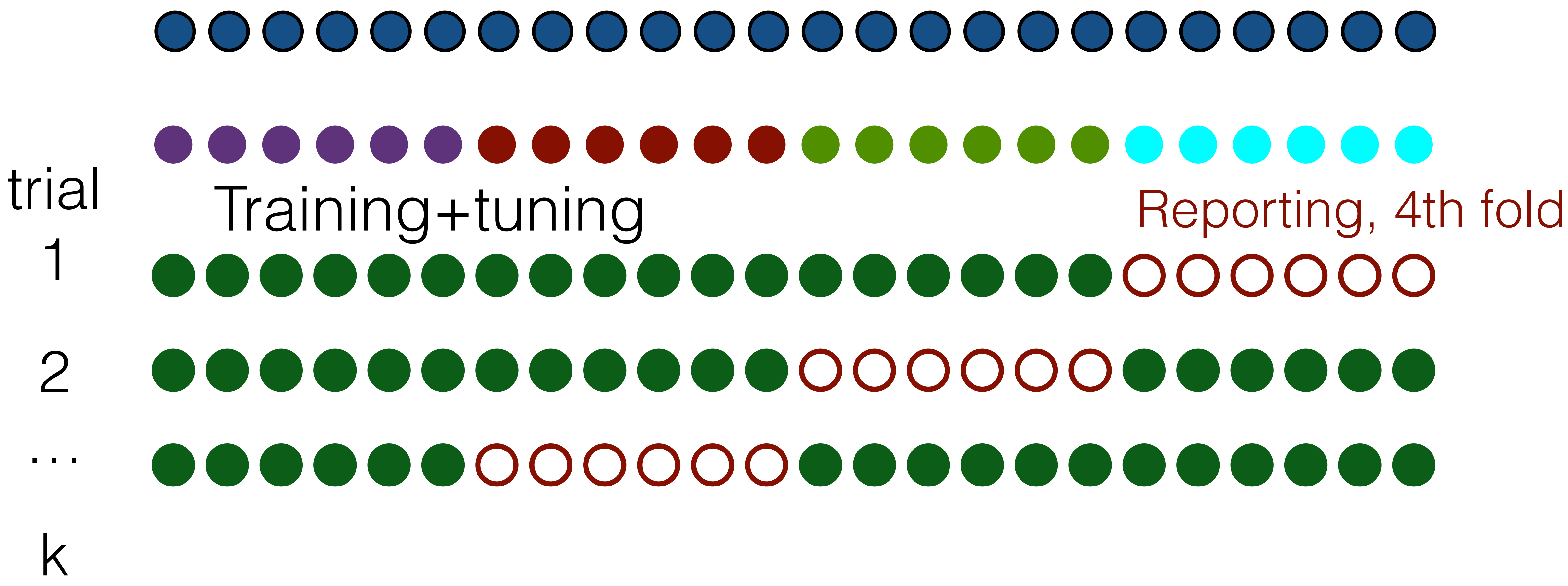
K-fold CV



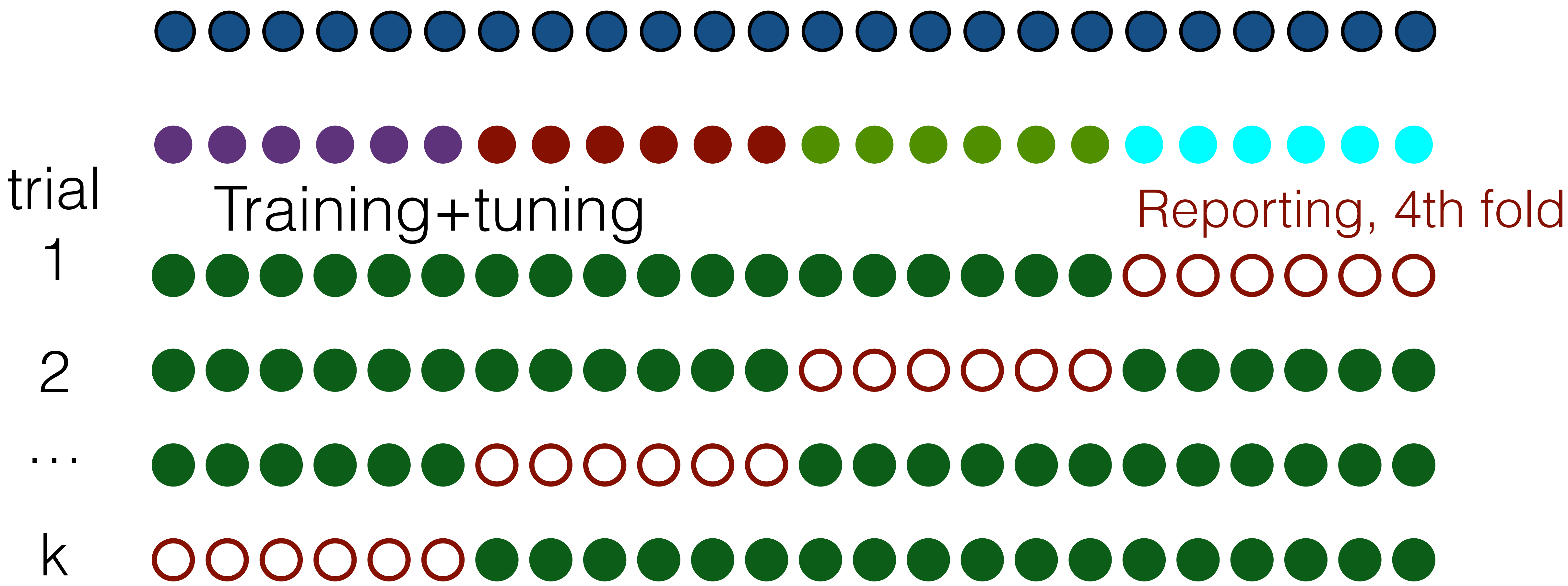
K-fold CV



K-fold CV

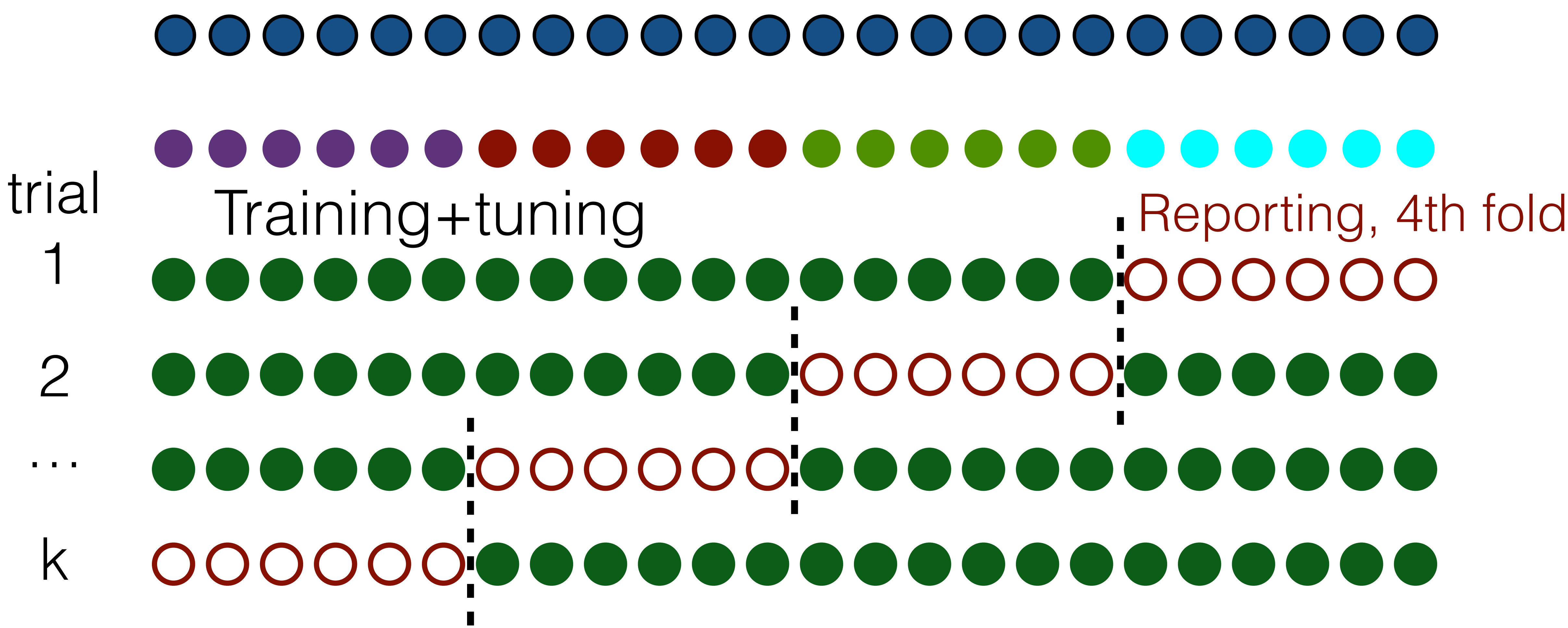


K-fold CV



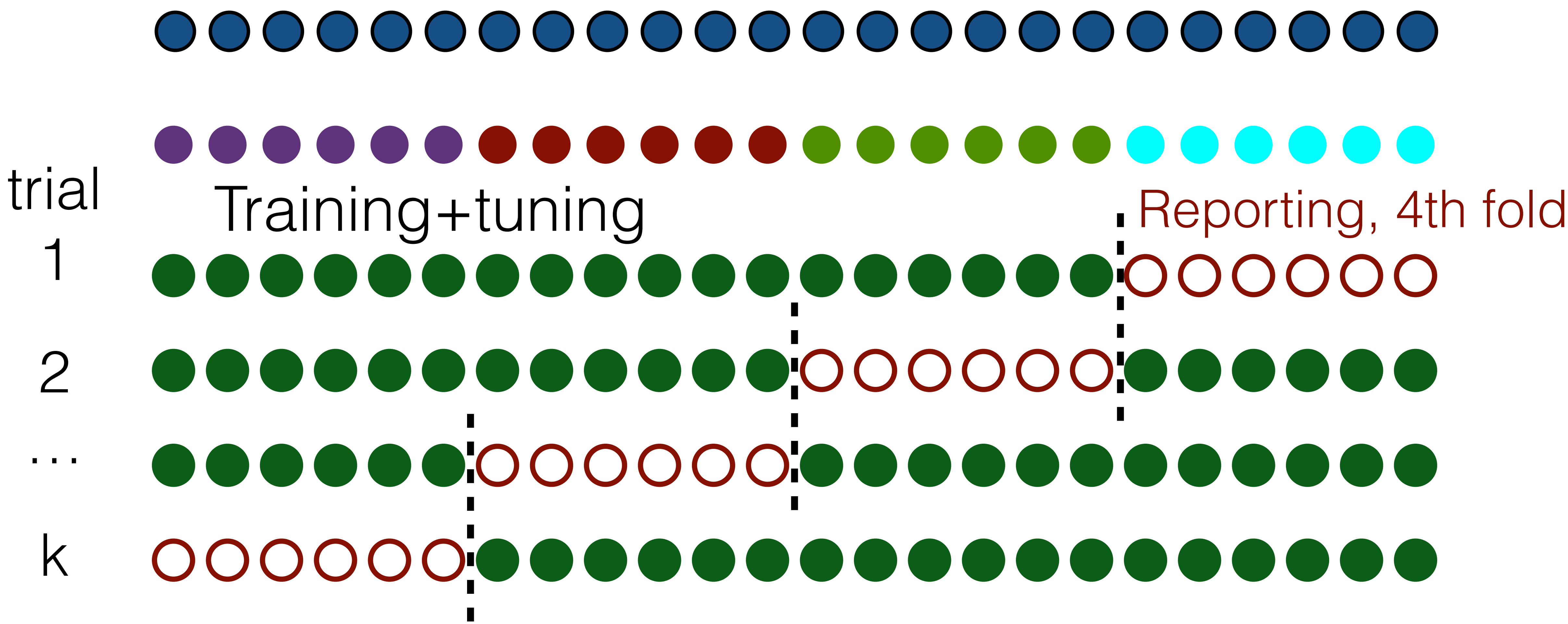
K-fold CV

Reporting sets in different trials are mutually disjoint!



K-fold CV

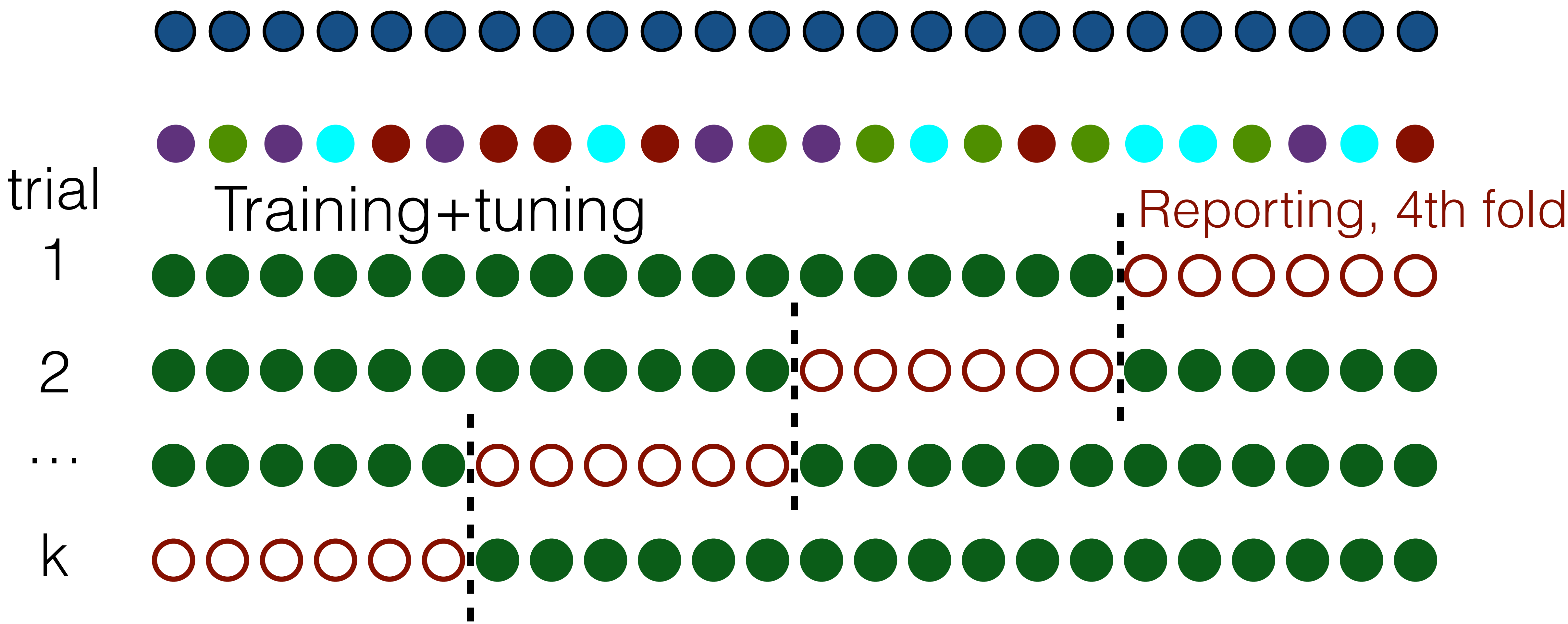
Reporting sets in different trials are mutually disjoint!



Note: different folds won't be contiguous.

K-fold CV

Reporting sets in different trials are mutually disjoint!

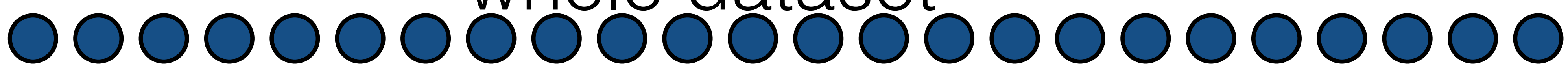


Note: different folds won't be contiguous.

Repeated Holdout CV

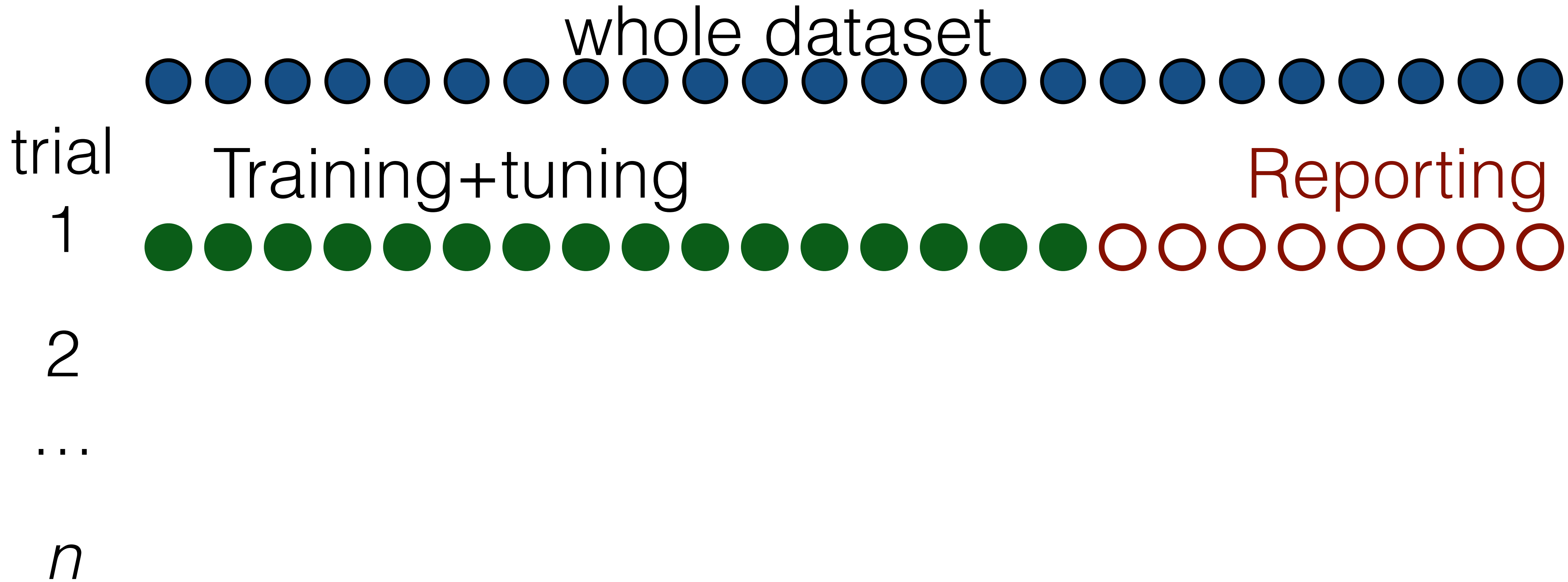
Set aside an independent subsample (e.g. 30%) for reporting

whole dataset



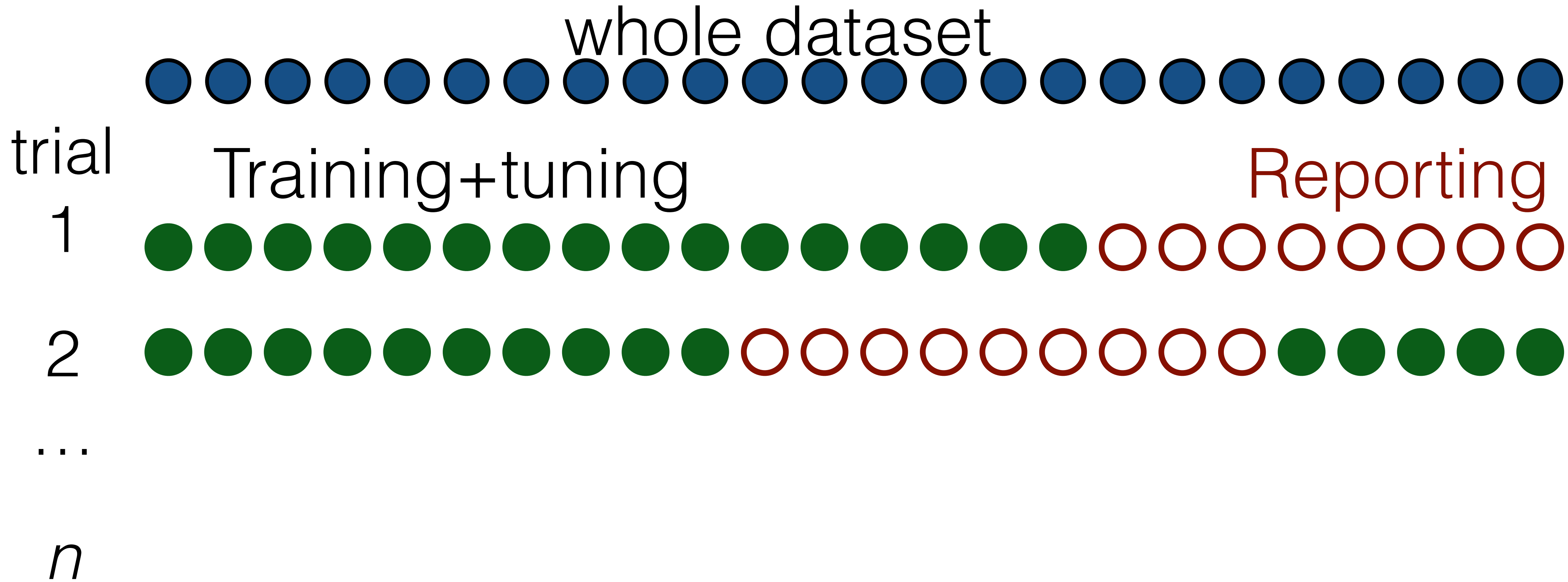
Repeated Holdout CV

Set aside an independent subsample (e.g. 30%) for reporting



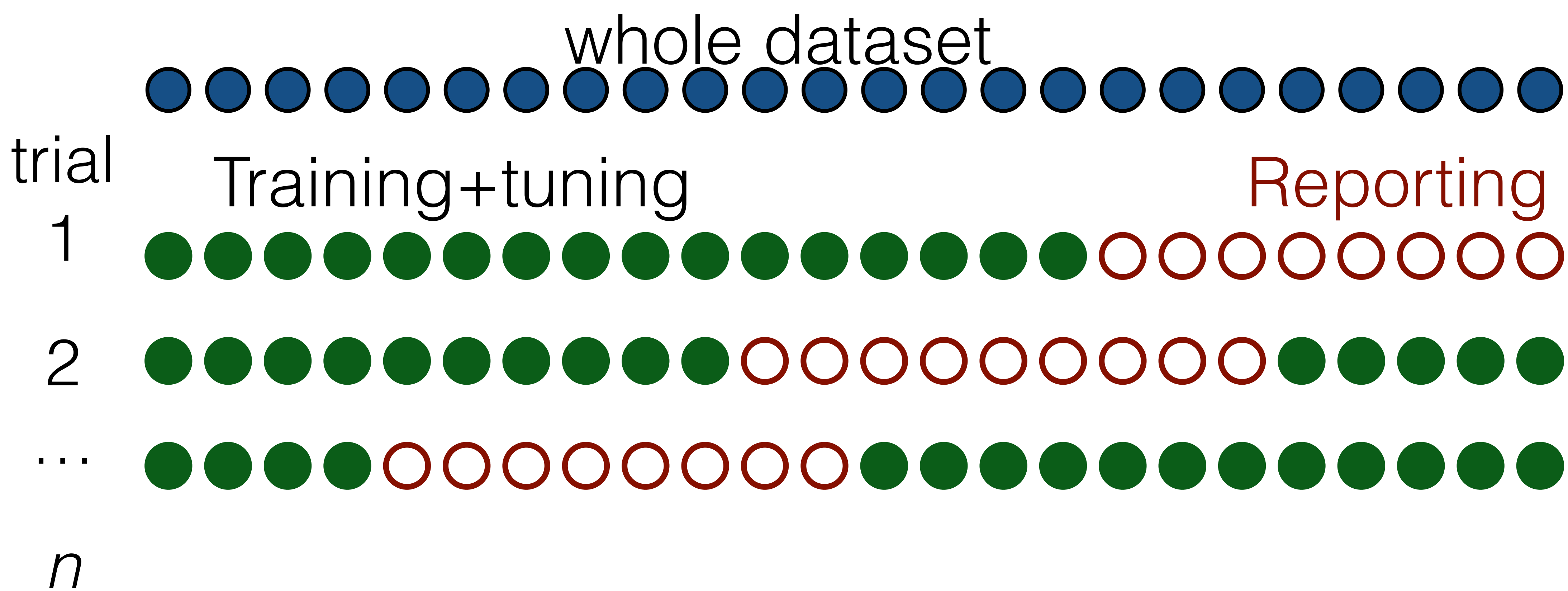
Repeated Holdout CV

Set aside an independent subsample (e.g. 30%) for reporting



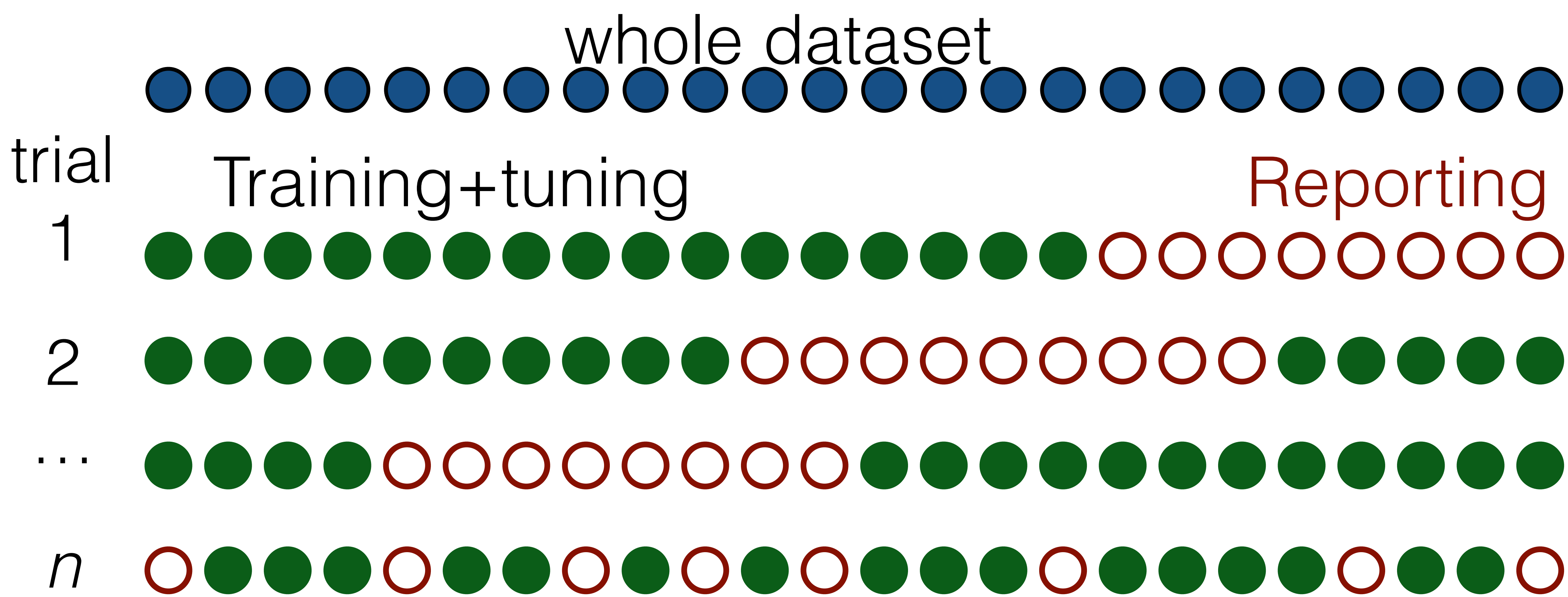
Repeated Holdout CV

Set aside an independent subsample (e.g. 30%) for reporting



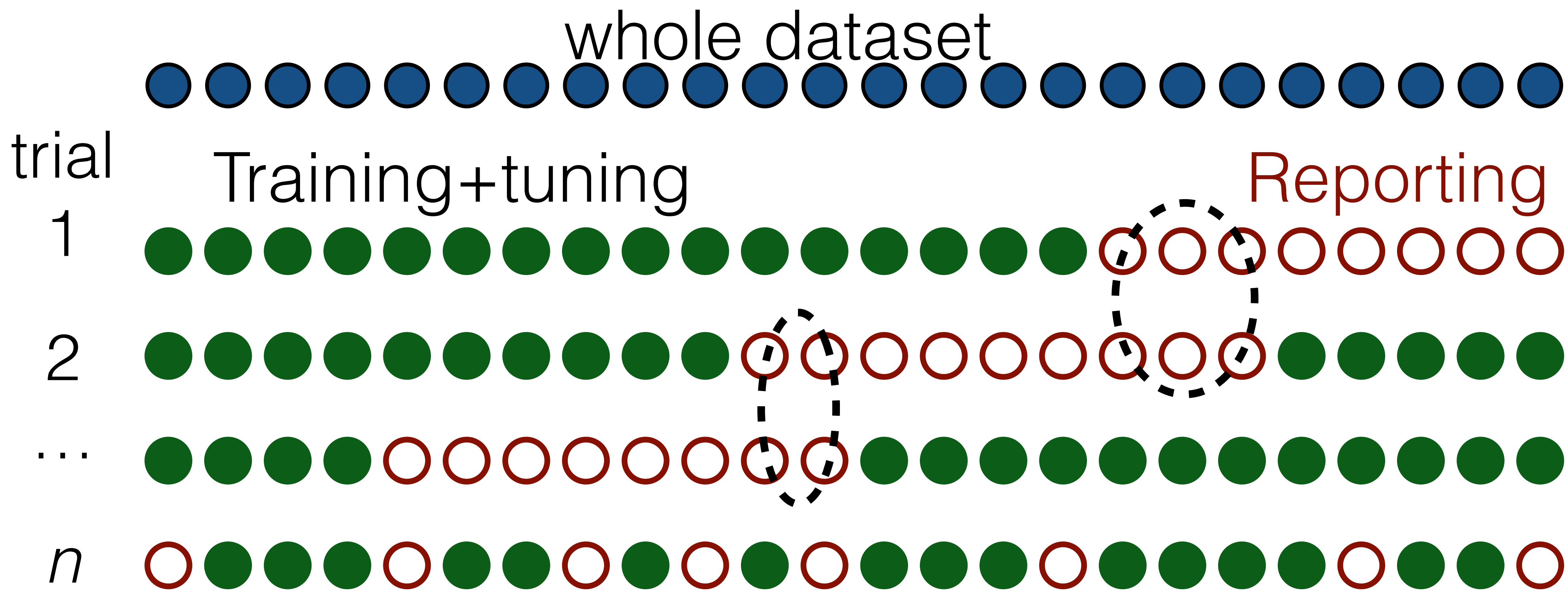
Repeated Holdout CV

Set aside an independent subsample (e.g. 30%) for reporting



Repeated Holdout CV

Set aside an independent subsample (e.g. 30%) for reporting



Note: there could be **overlap** among the reporting sets from different trials! Hence, a large *n* is recommended.

CV has many variations!

CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20

CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)

CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
    - train % = 50, 63.2, 75, 80, 90

CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
    - train % = 50, 63.2, 75, 80, 90
    - **stratified**

CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting

CV has many variations!

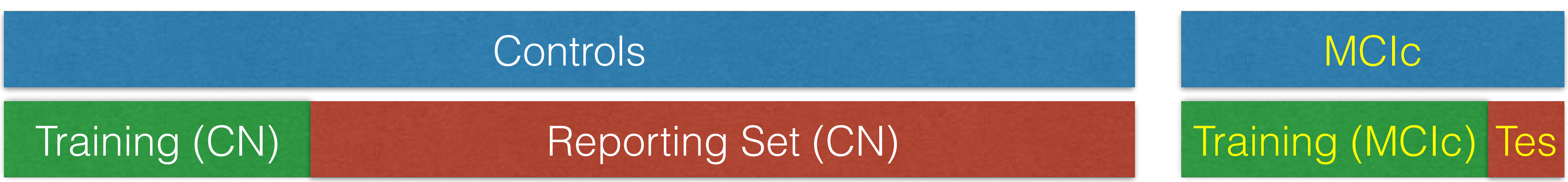
- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting

CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting

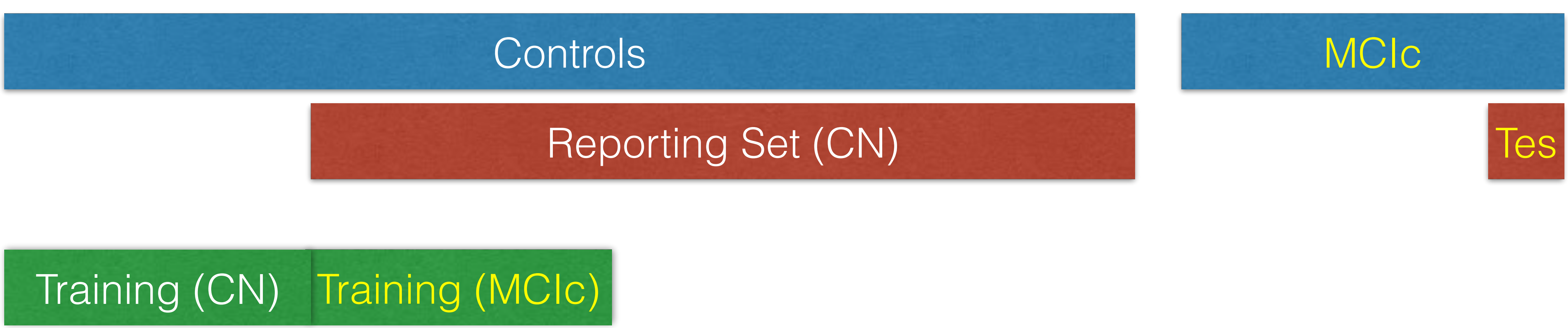
CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting



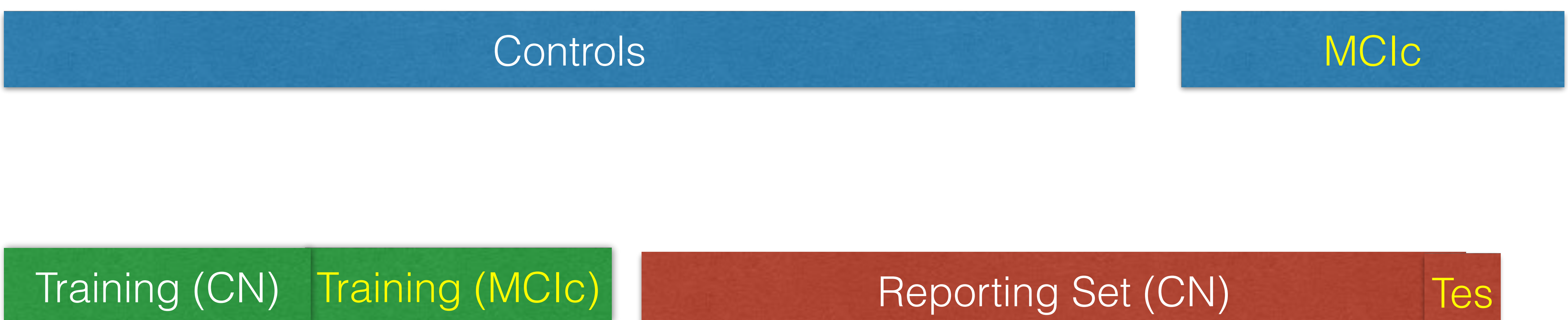
CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
    - train % = 50, 63.2, 75, 80, 90
    - **stratified**
        - across train/reporting



CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting

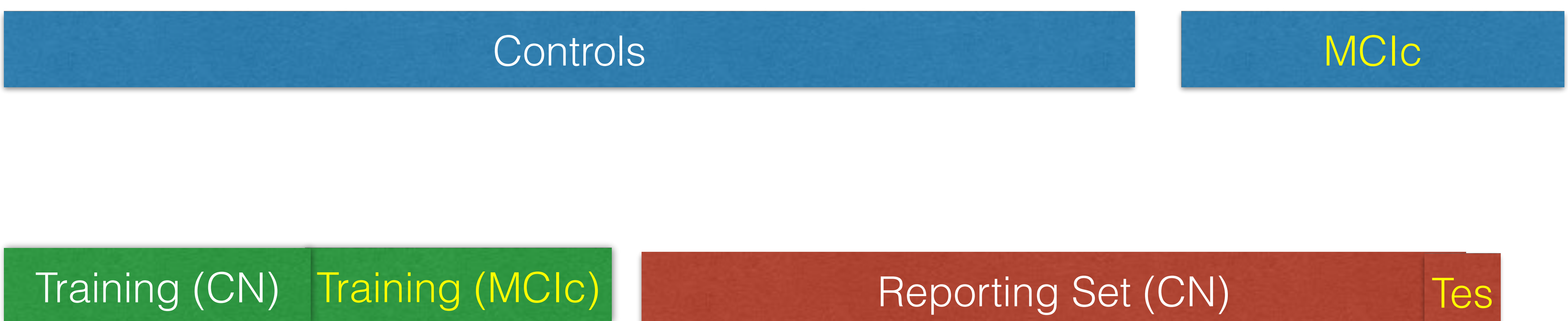


CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting
    - across classes

CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting
    - across classes
- **inverted**:
  very small training, large reporting

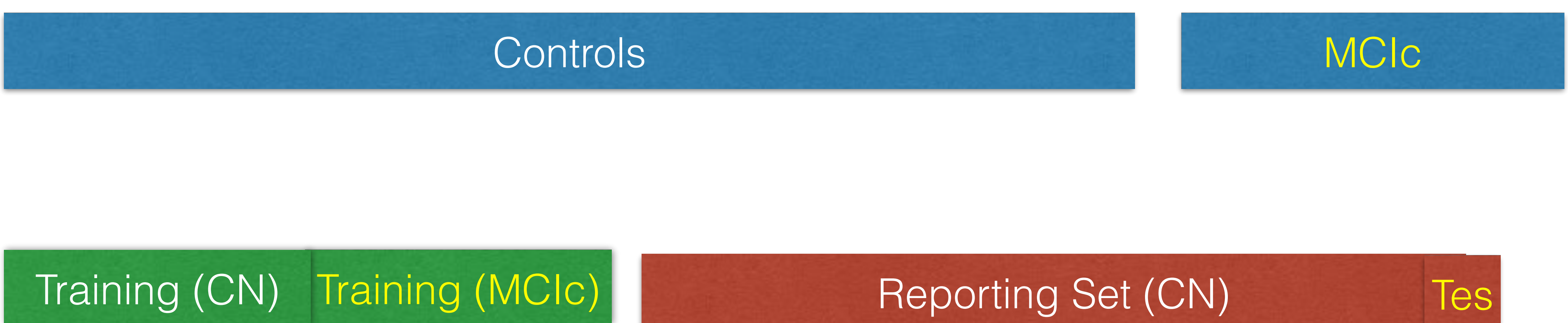


CV has many variations!

- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting
    - across classes
- **inverted**:
  very small training, large reporting
- leave one [unit] out:

CV has many variations!

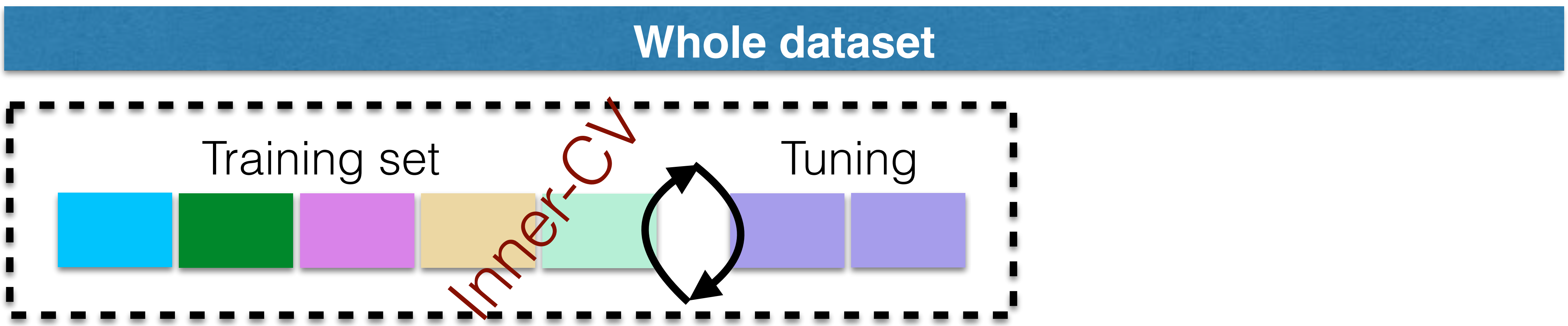
- k-fold, k = 2, 3, 5, 10, 20
- repeated hold-out (random subsampling)
  - train % = 50, 63.2, 75, 80, 90
  - **stratified**
    - across train/reporting
    - across classes
- **inverted**:
  very small training, large reporting
- leave one [unit] out:
  - unit —> sample / pair / tuple / condition / task / block out



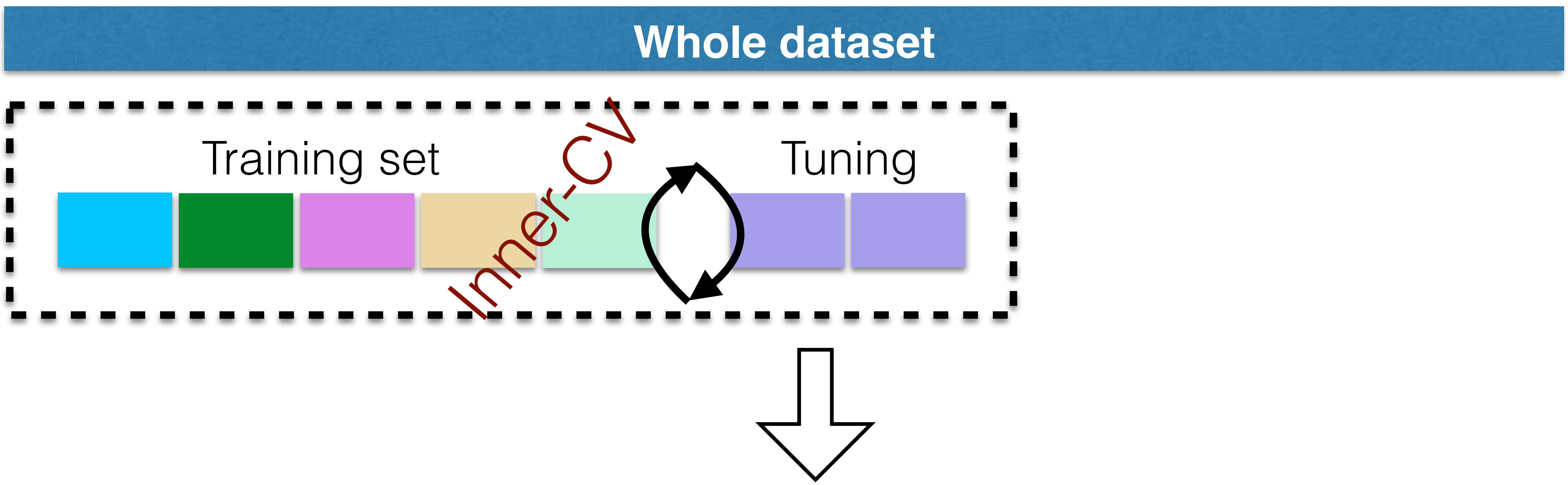
Measuring bias in CV measurements

Whole dataset

Measuring bias in CV measurements

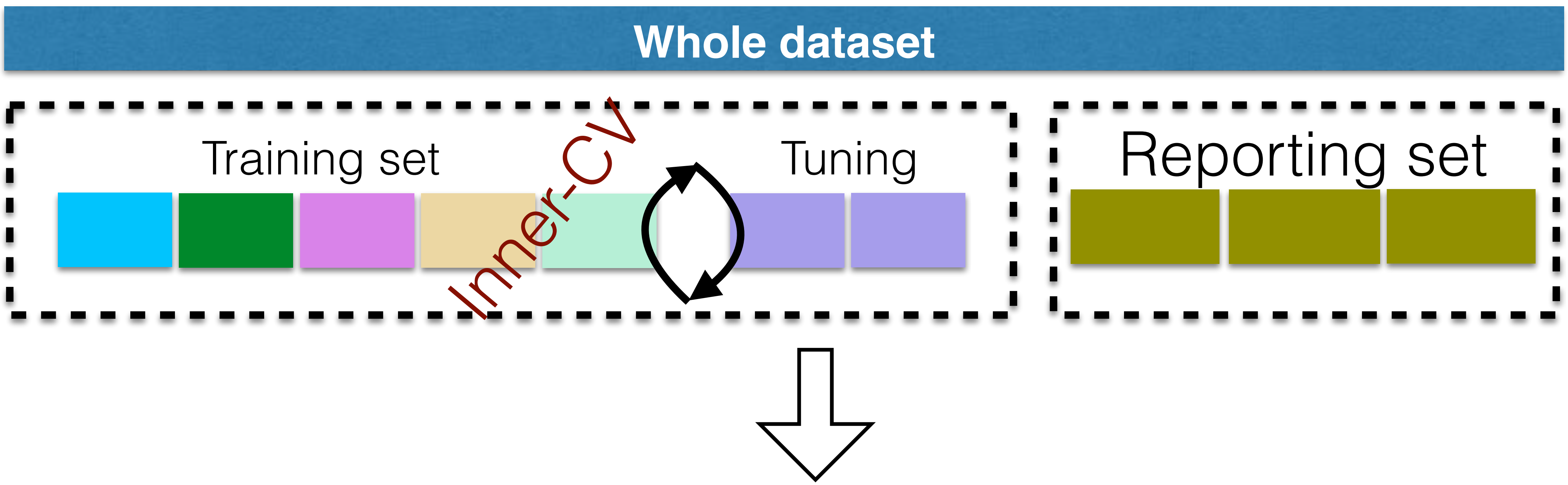


Measuring bias in CV measurements



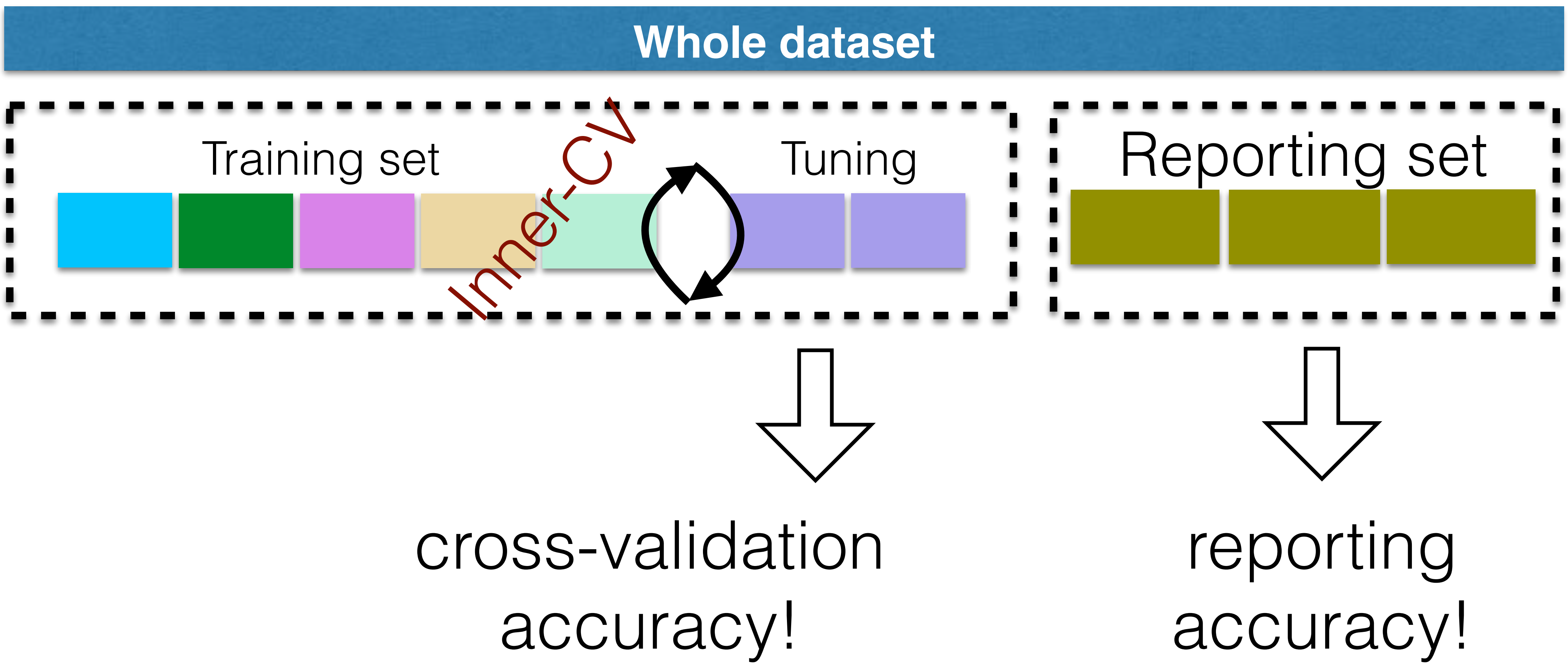
cross-validation accuracy!

Measuring bias in CV measurements

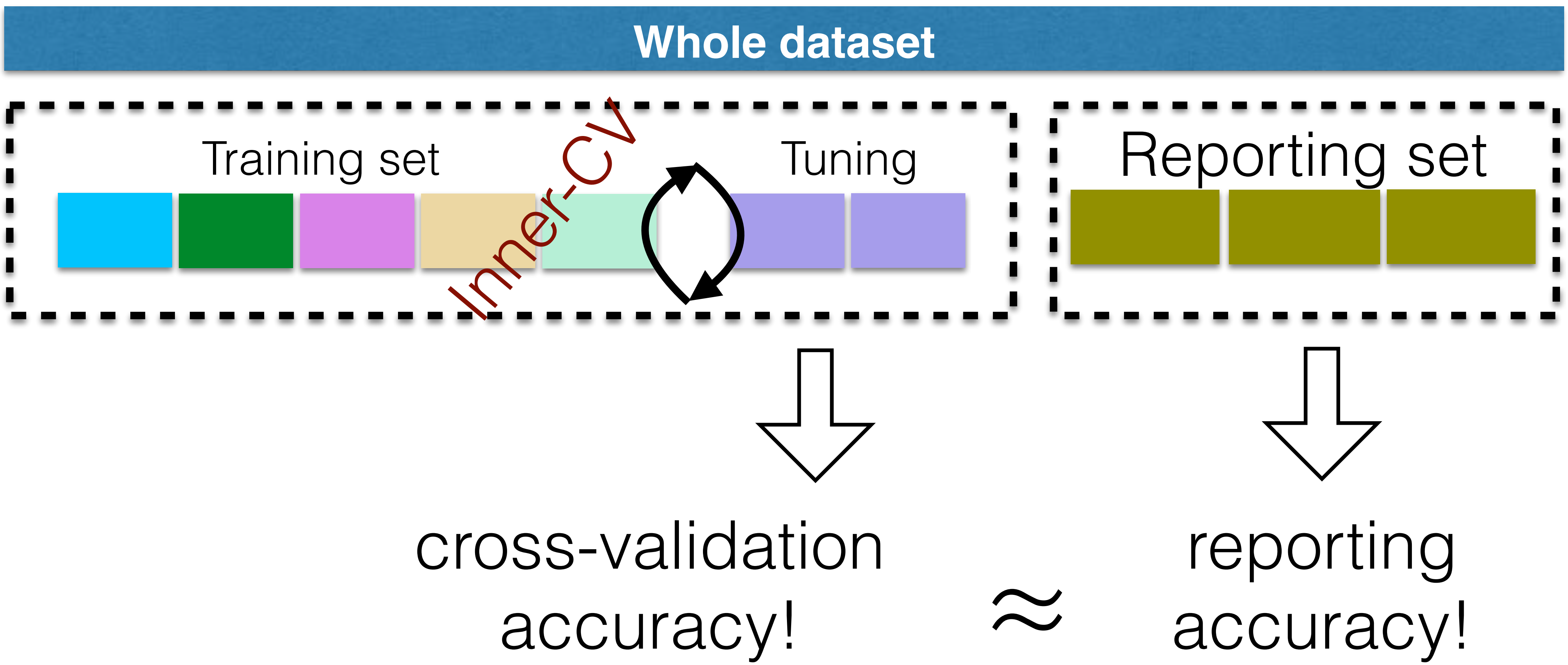


cross-validation accuracy!

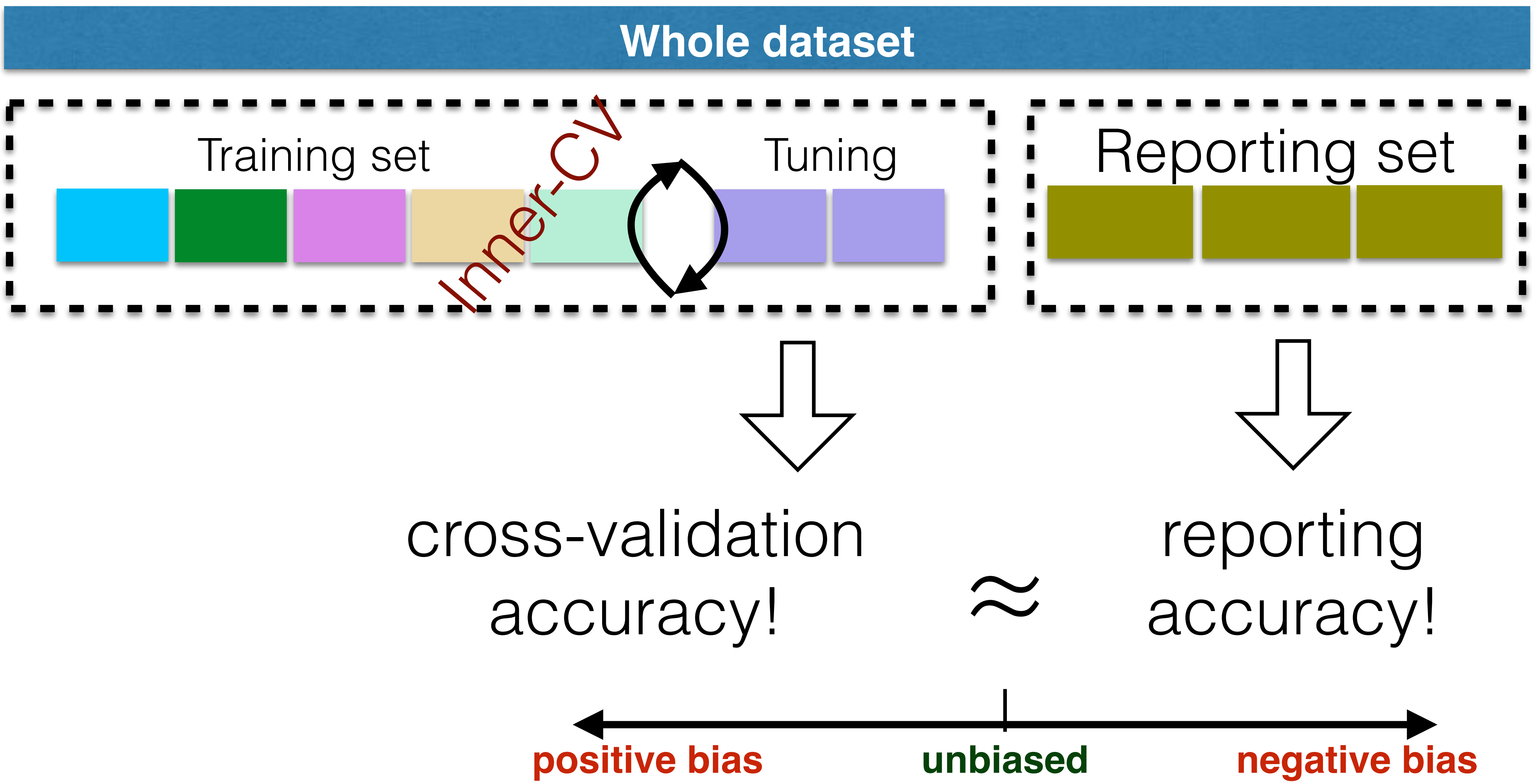
Measuring bias in CV measurements



Measuring bias in CV measurements



Measuring bias in CV measurements

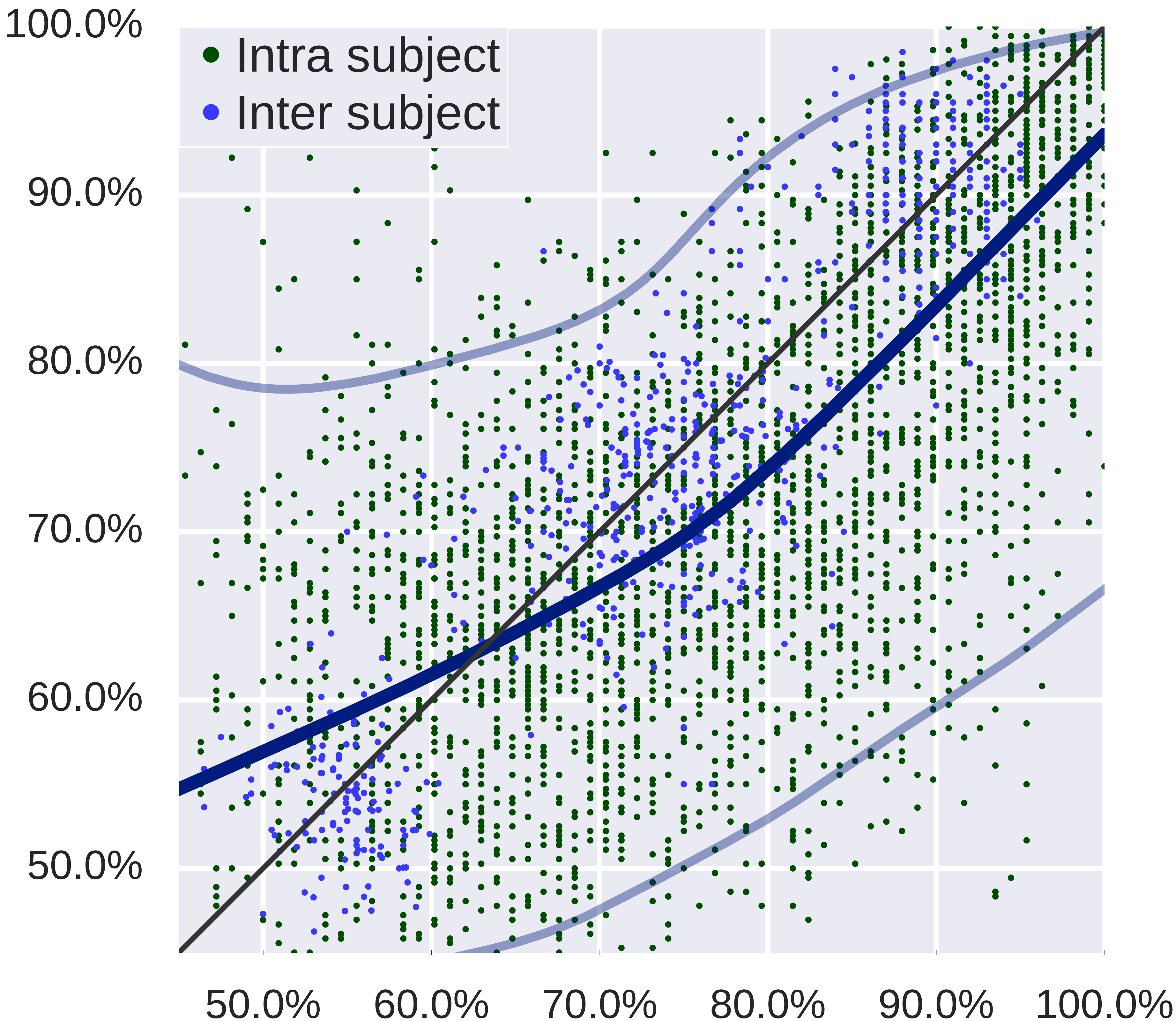


fMRI datasets

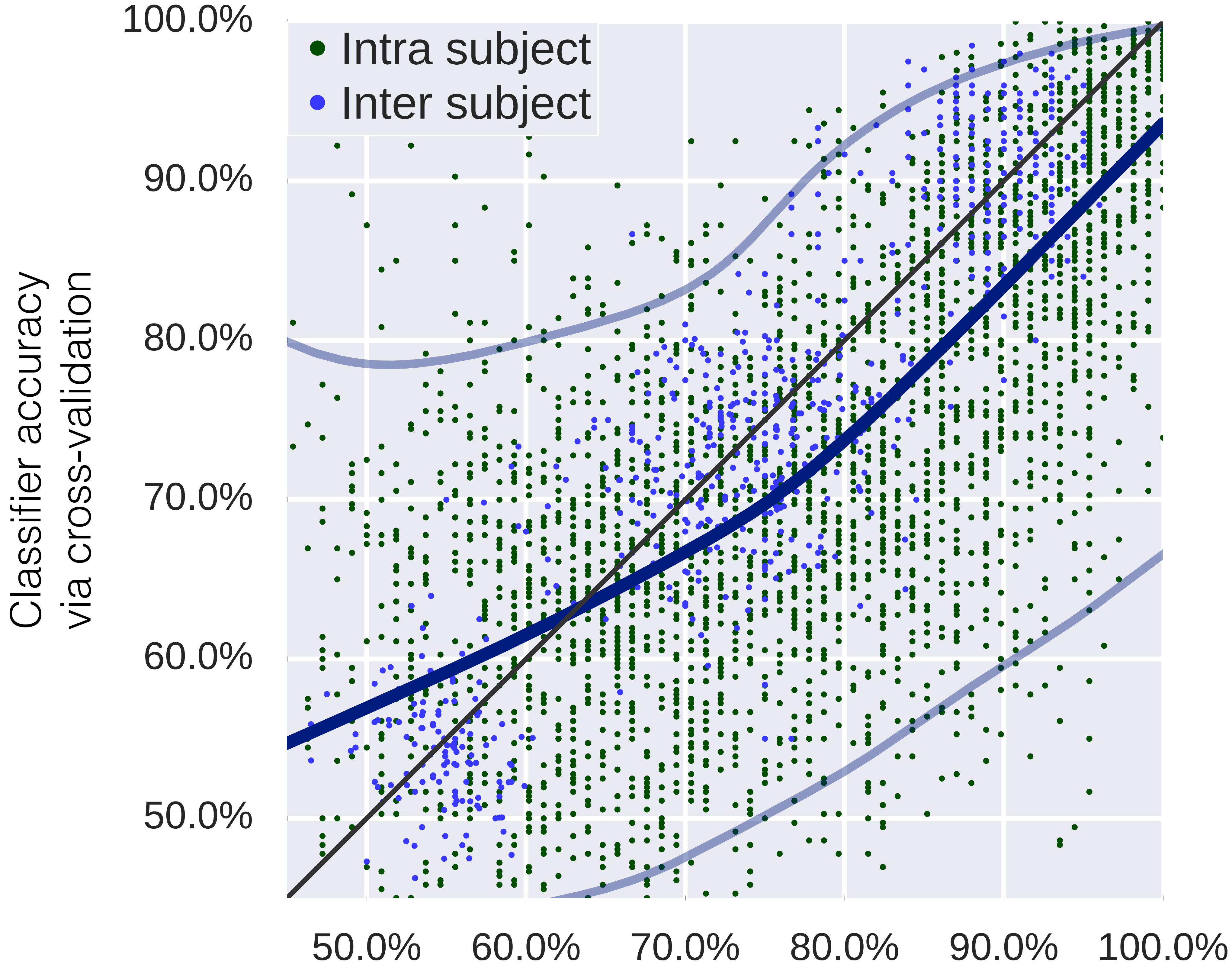
Dataset	Intra- or inter?	# samples	# blocks (sessions or subjects)	Tasks
Haxby	**Intra**	209	12 seconds	various
Duncan	Inter	196	49 subjects	various
Wager	Inter	390	34 subjects	various
Cohen	Inter	80	24 subjects	various
Moran	Inter	138	36 subjects	various
Henson	Inter	286	16 subjects	various
Knops	Inter	14	19 subjects	various

Reference: Varoquaux, G., Raamana, P. R., Engemann, D. A., Hoyos-Idrobo, A., Schwartz, Y., & Thirion, B. (2016). **Assessing and tuning brain decoders: cross-validation, caveats, and guidelines.** NeuroImage.

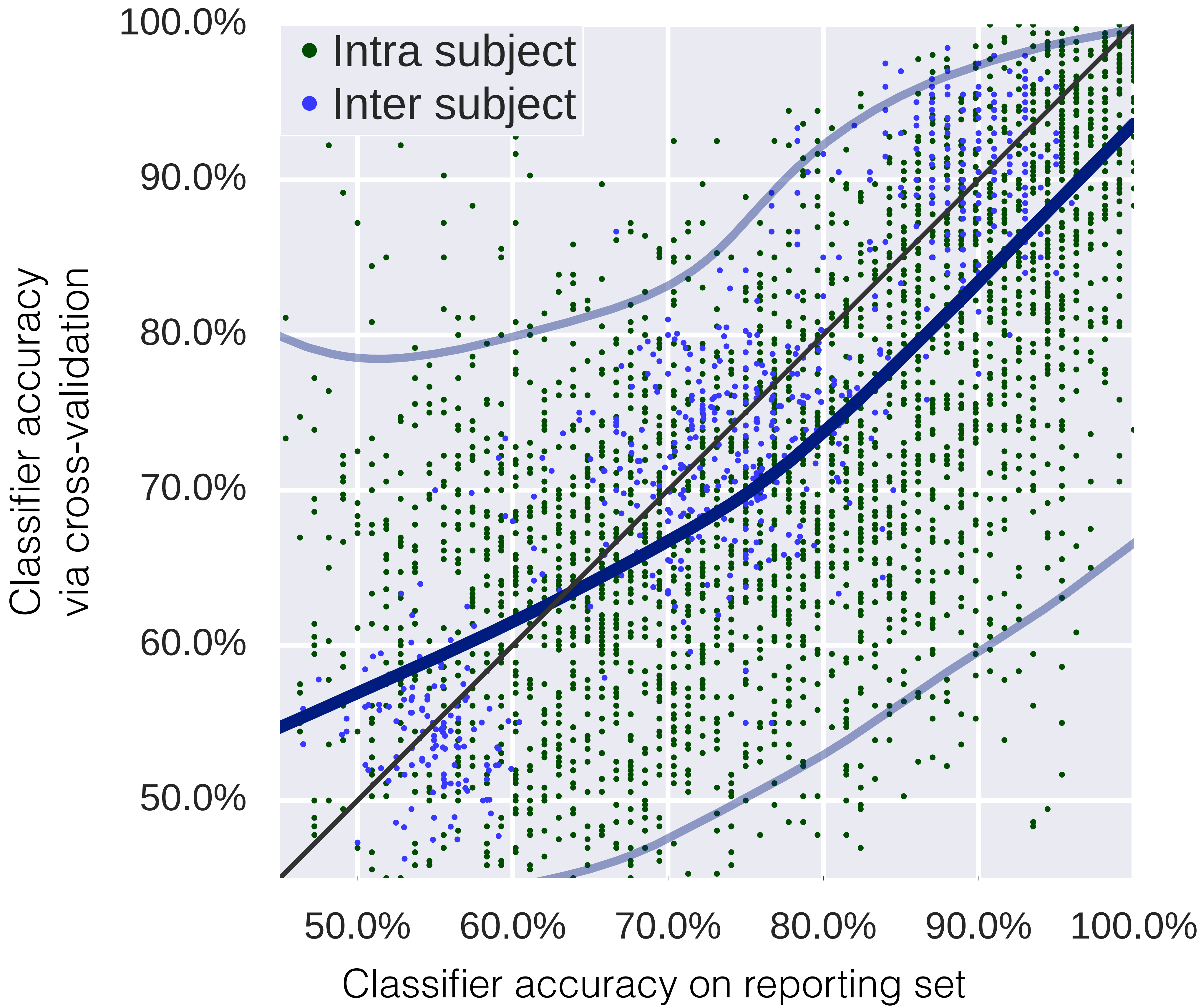
Repeated holdout (10 trials, 20% reporting)



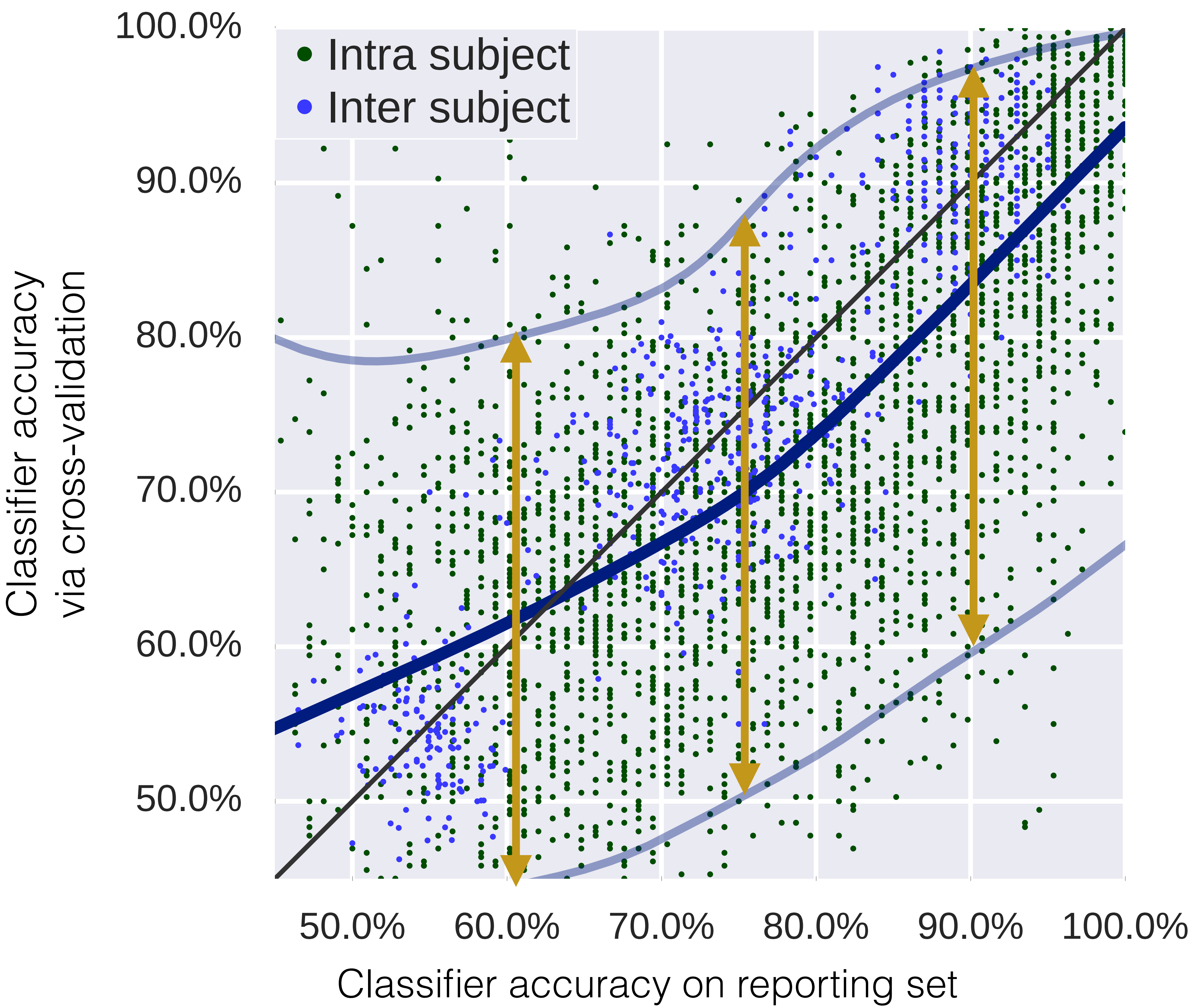
Repeated holdout (10 trials, 20% reporting)



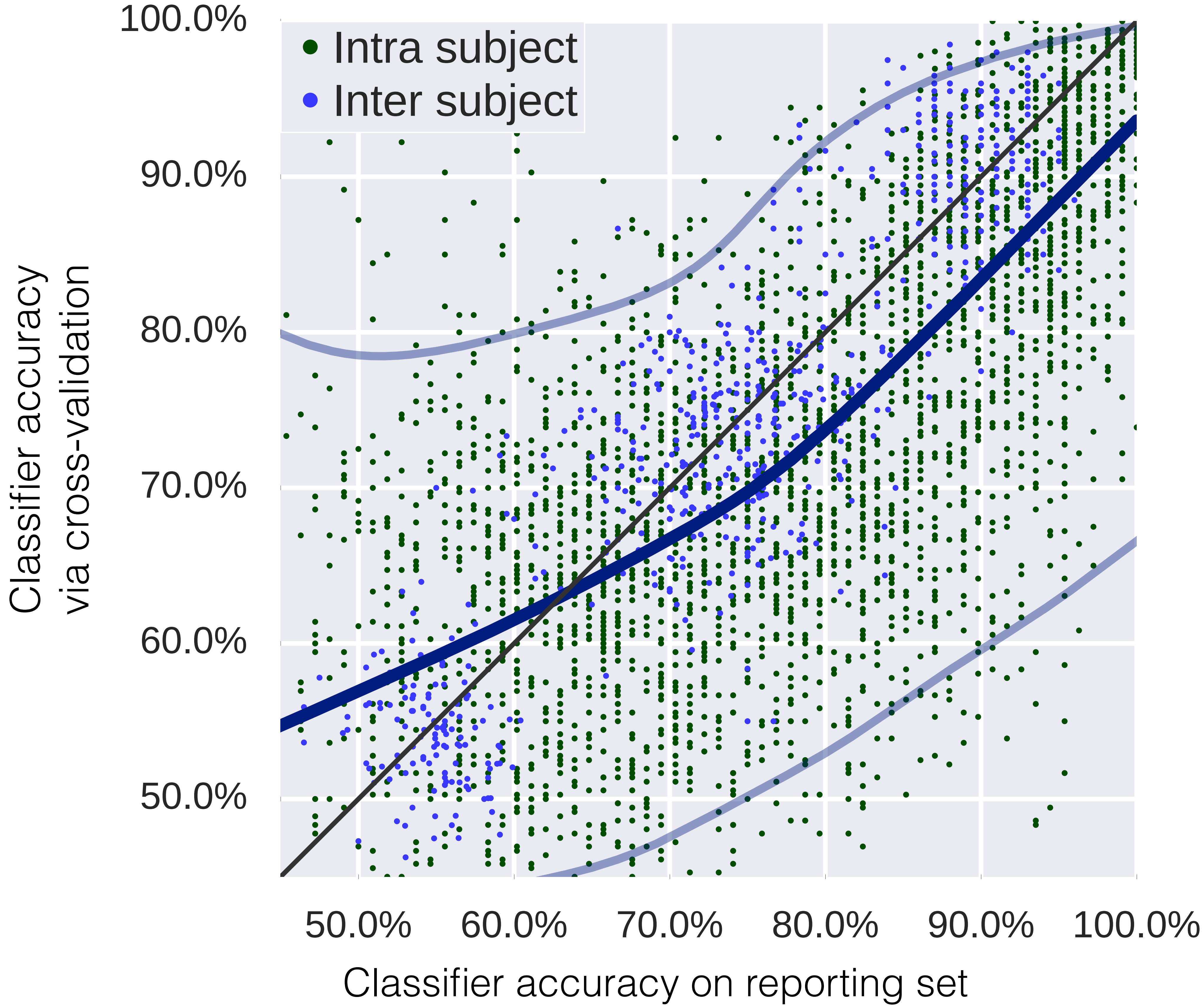
Repeated holdout (10 trials, 20% reporting)



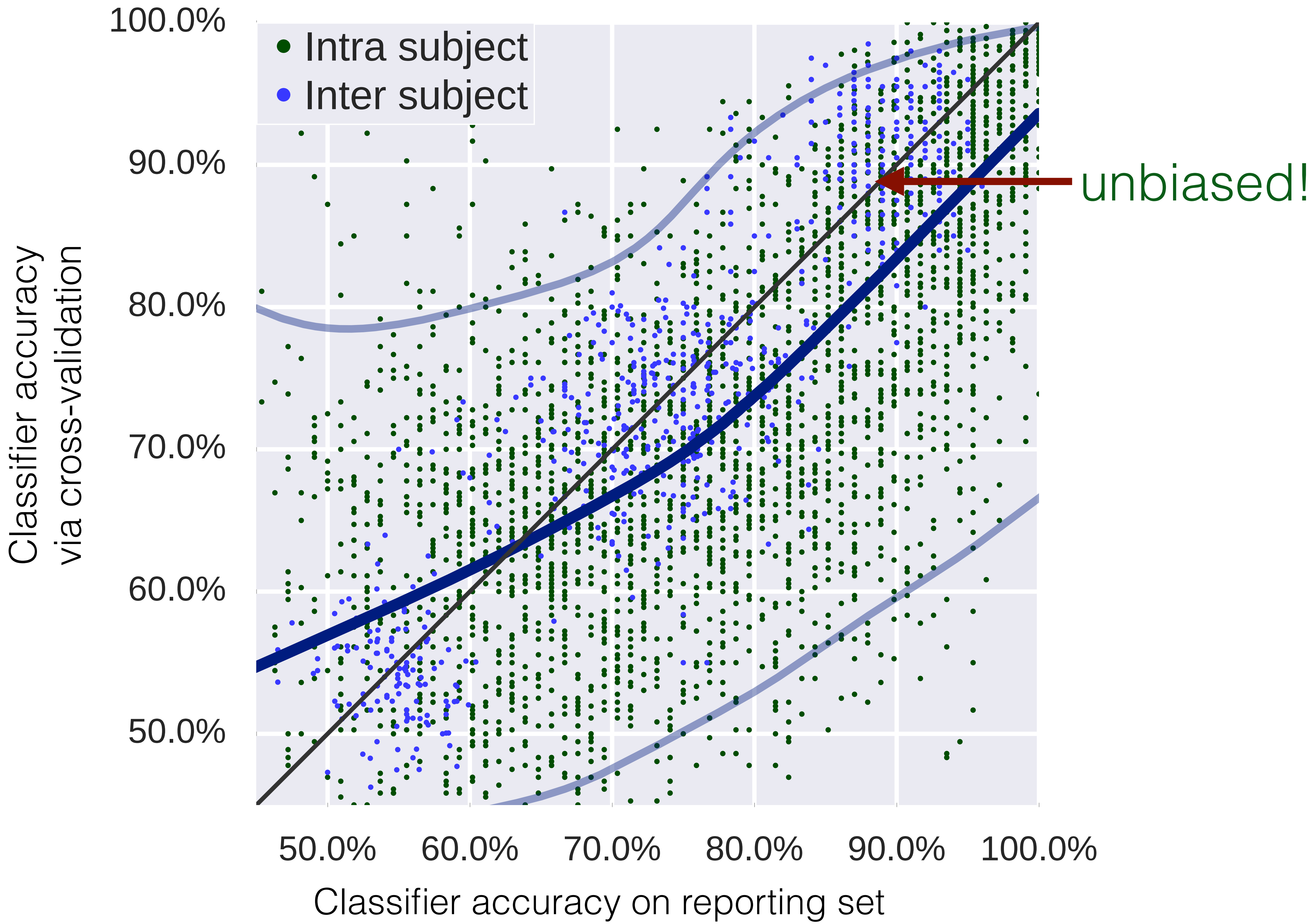
Repeated holdout (10 trials, 20% reporting)



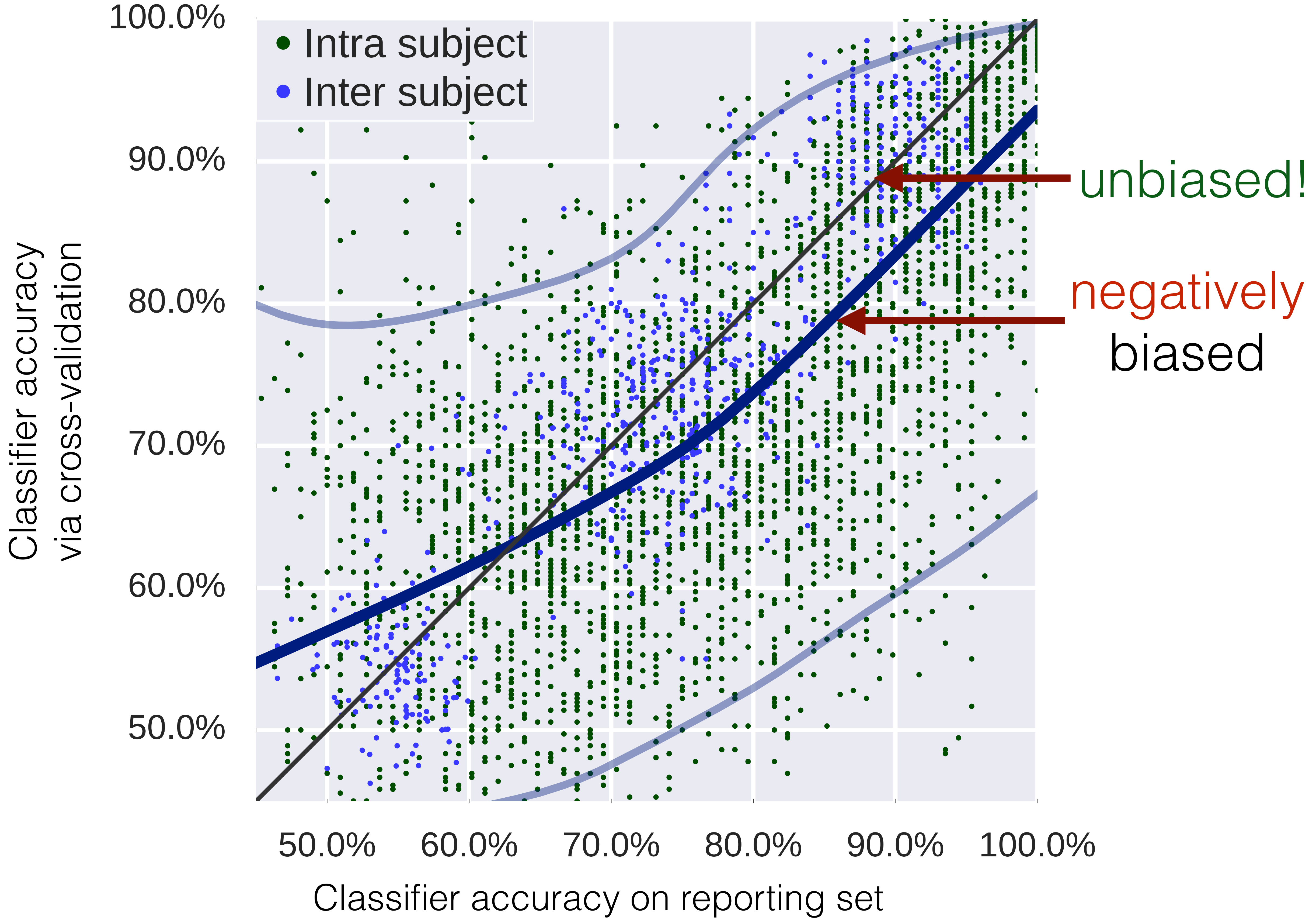
Repeated holdout (10 trials, 20% reporting)



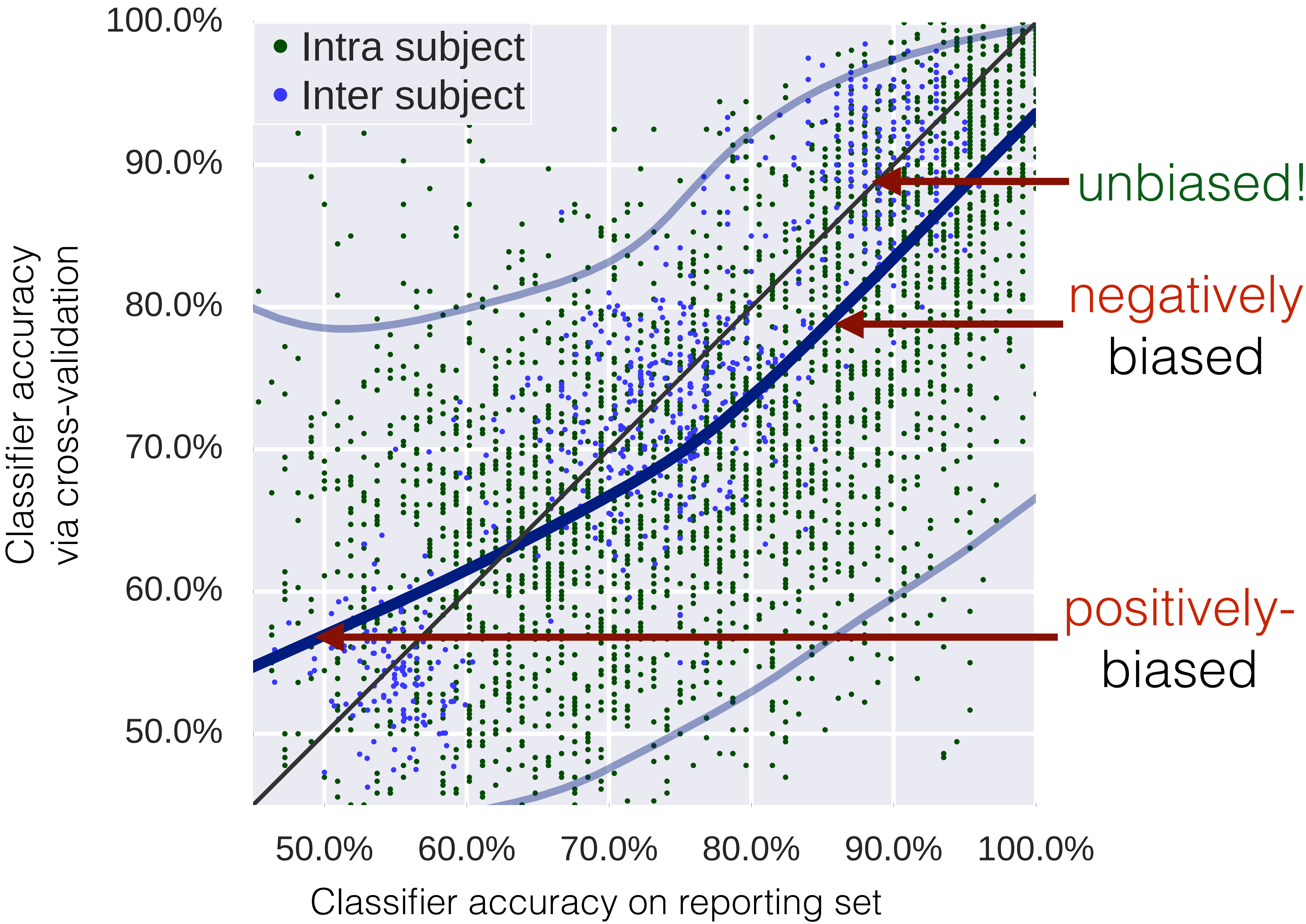
Repeated holdout (10 trials, 20% reporting)



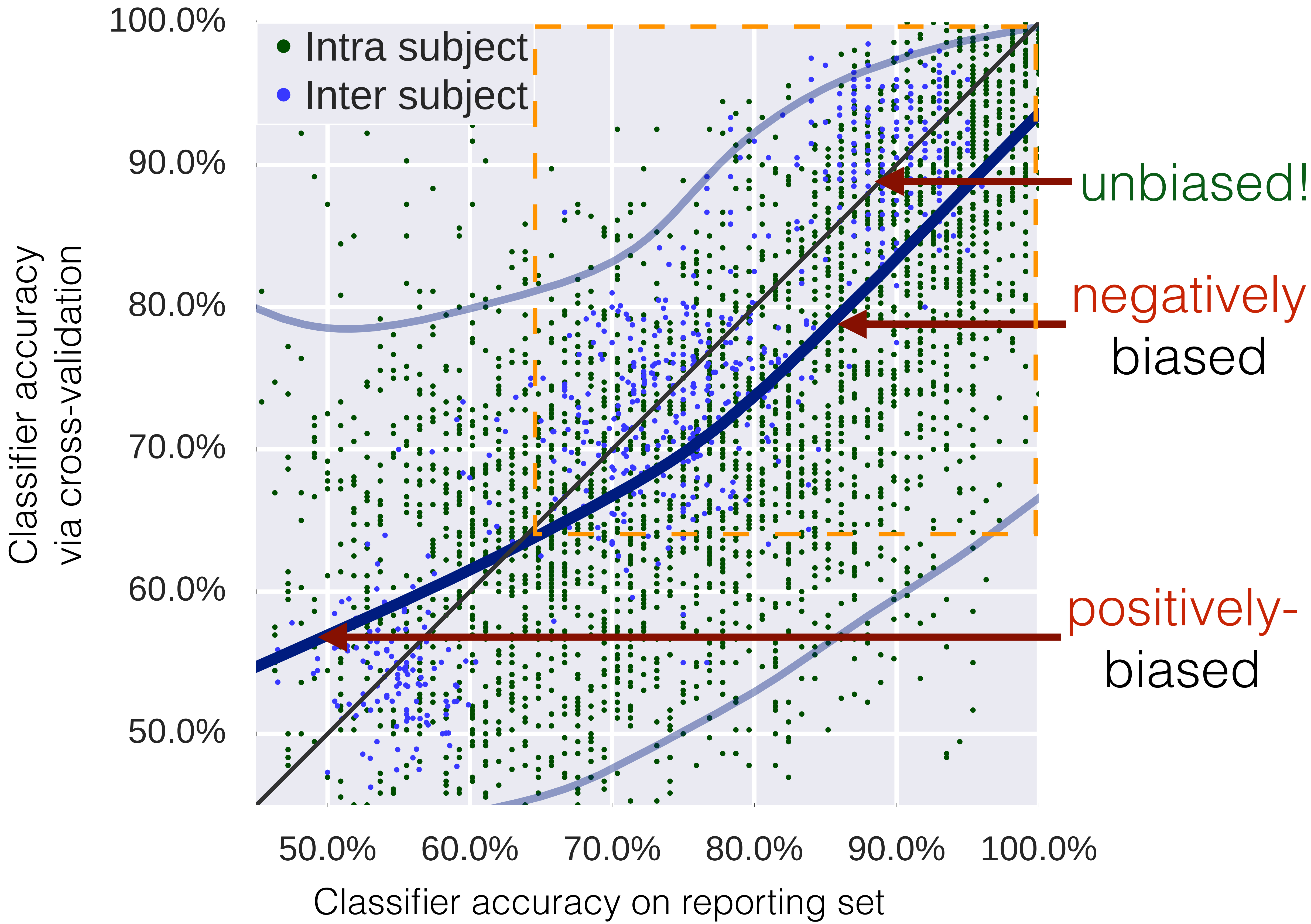
Repeated holdout (10 trials, 20% reporting)



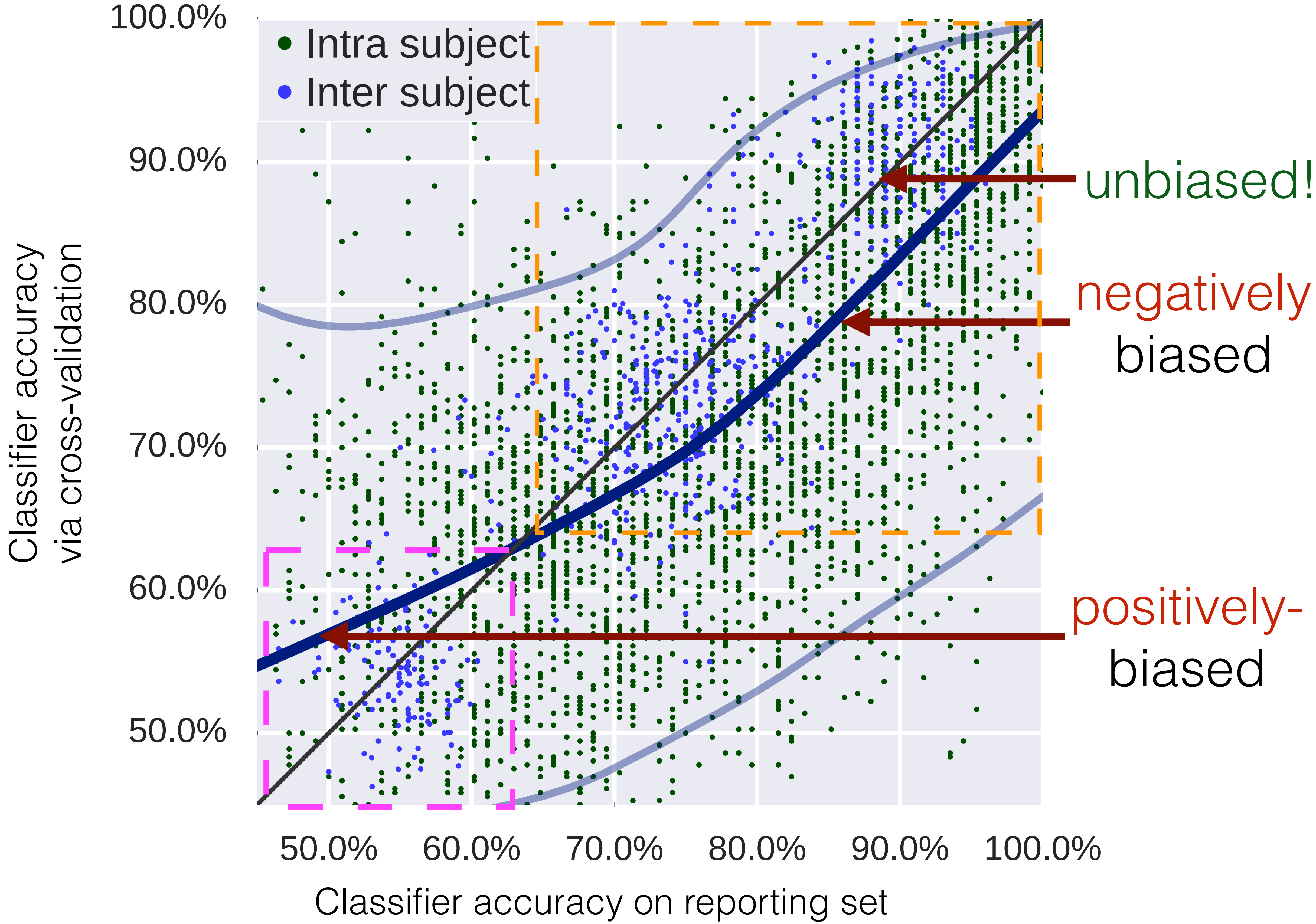
Repeated holdout (10 trials, 20% reporting)



Repeated holdout (10 trials, 20% reporting)

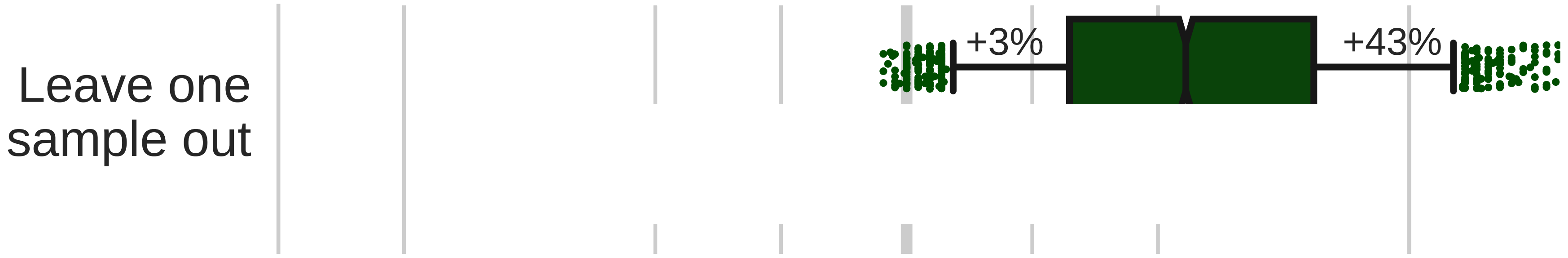


Repeated holdout (10 trials, 20% reporting)



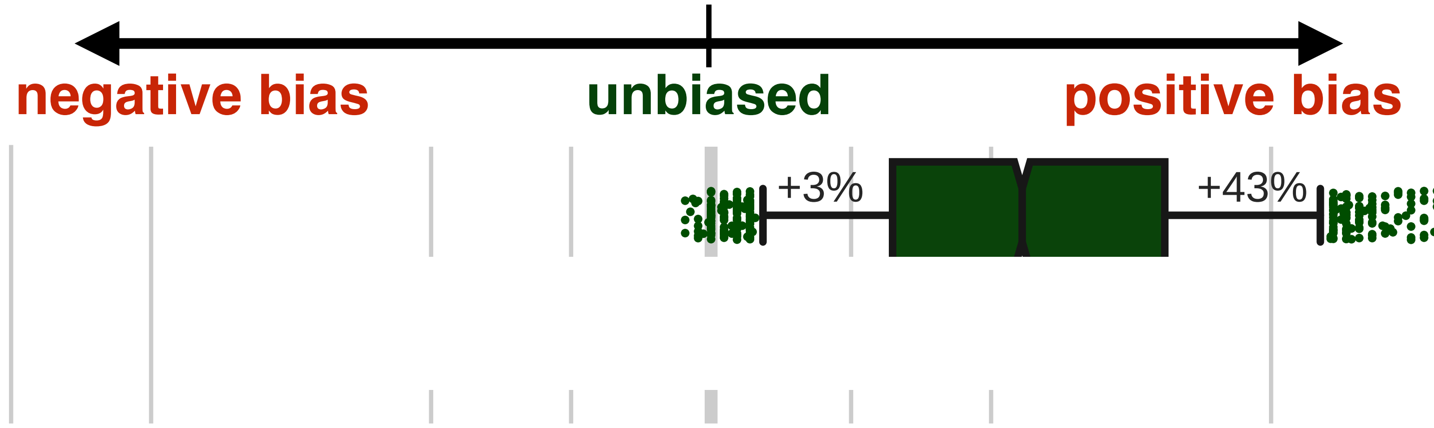
CV vs. Validation: real data

conservative



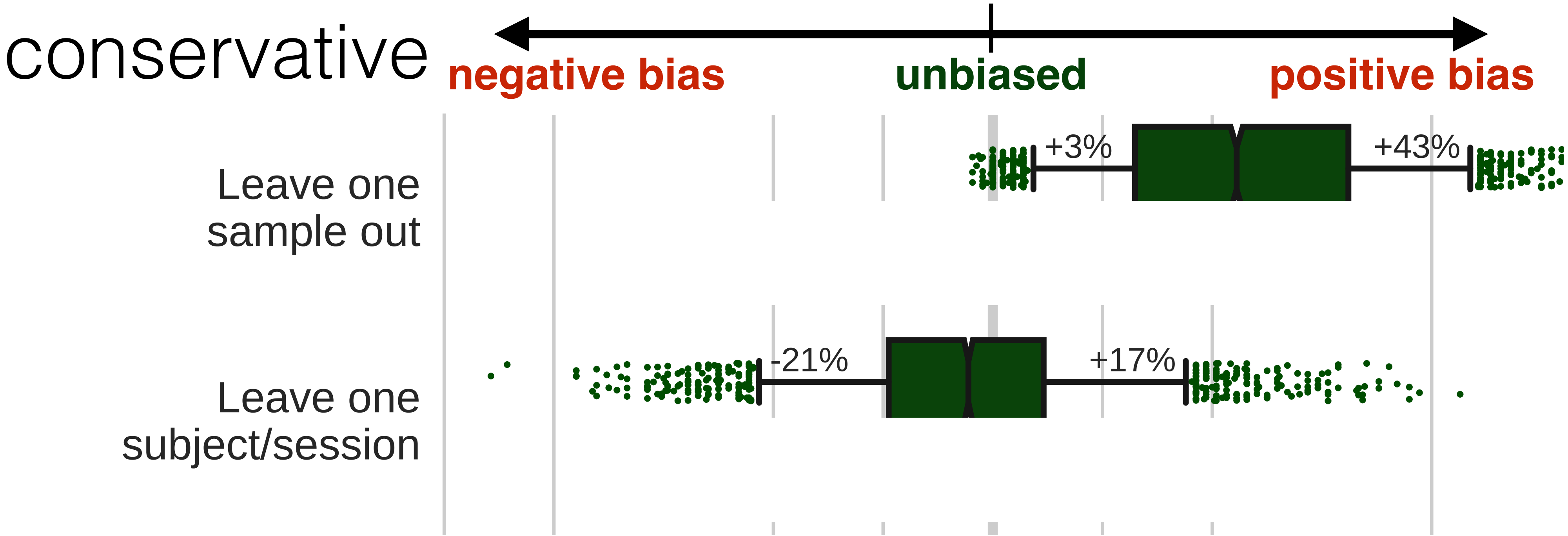
CV vs. Validation: real data

conservative

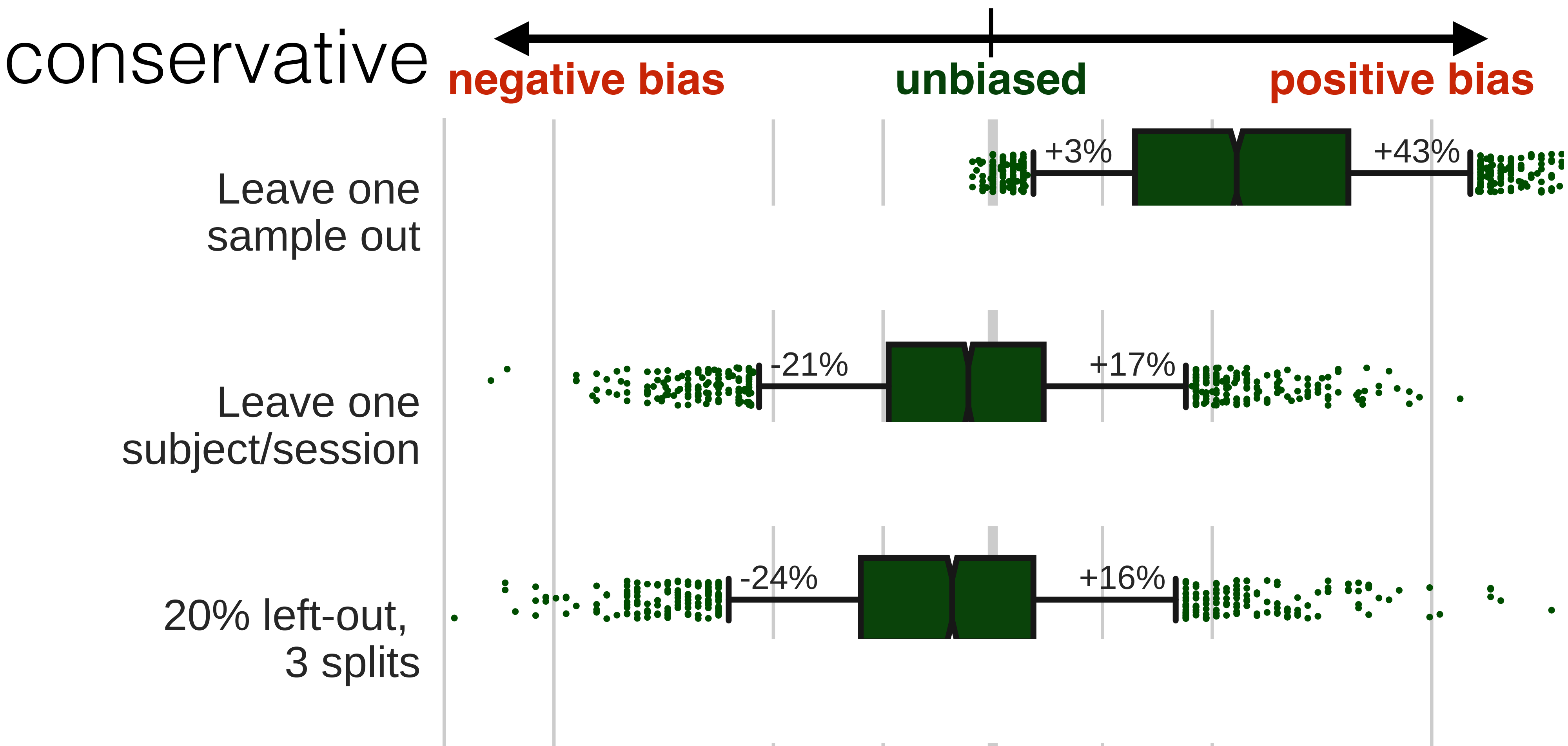


Leave one
sample out

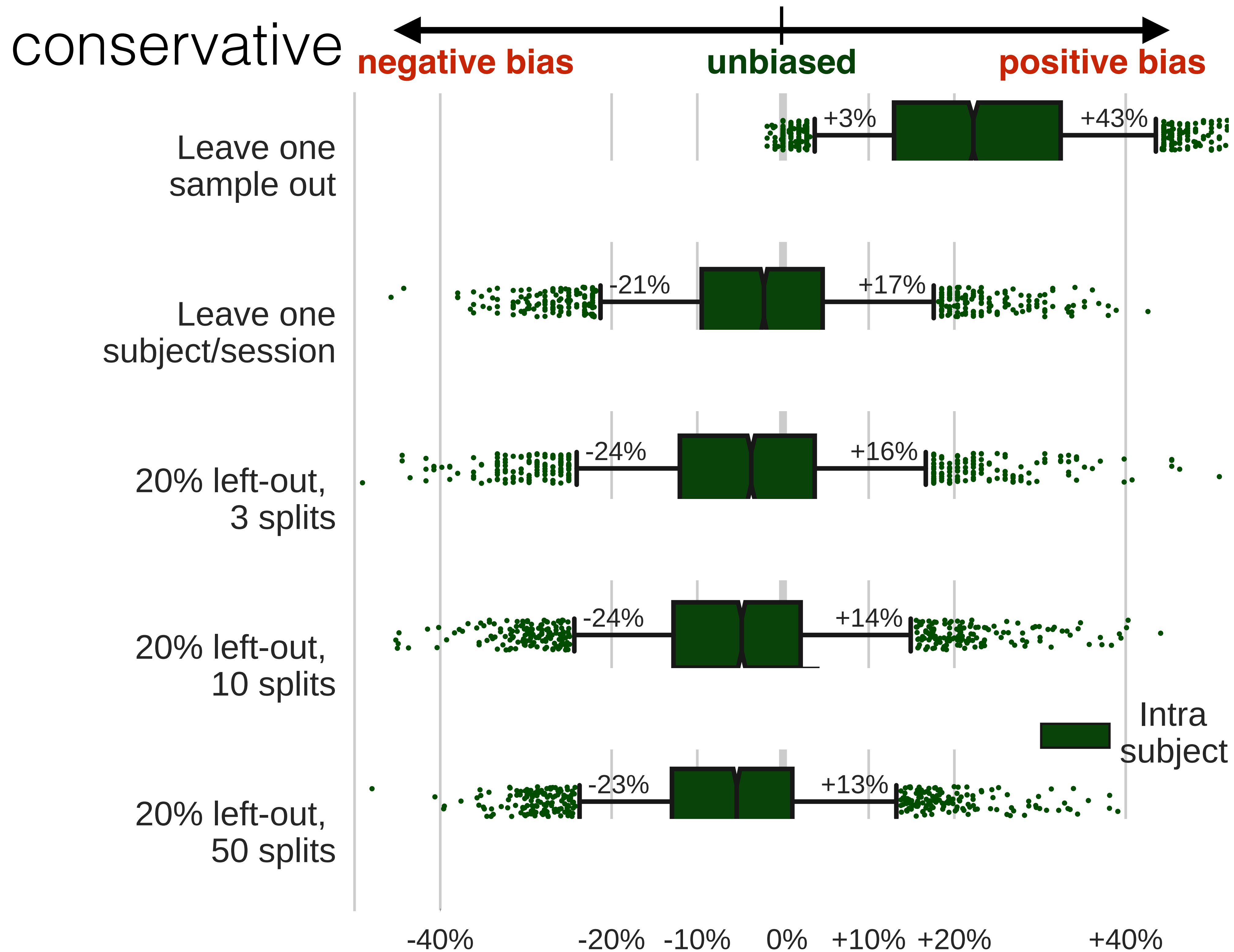
CV vs. Validation: real data



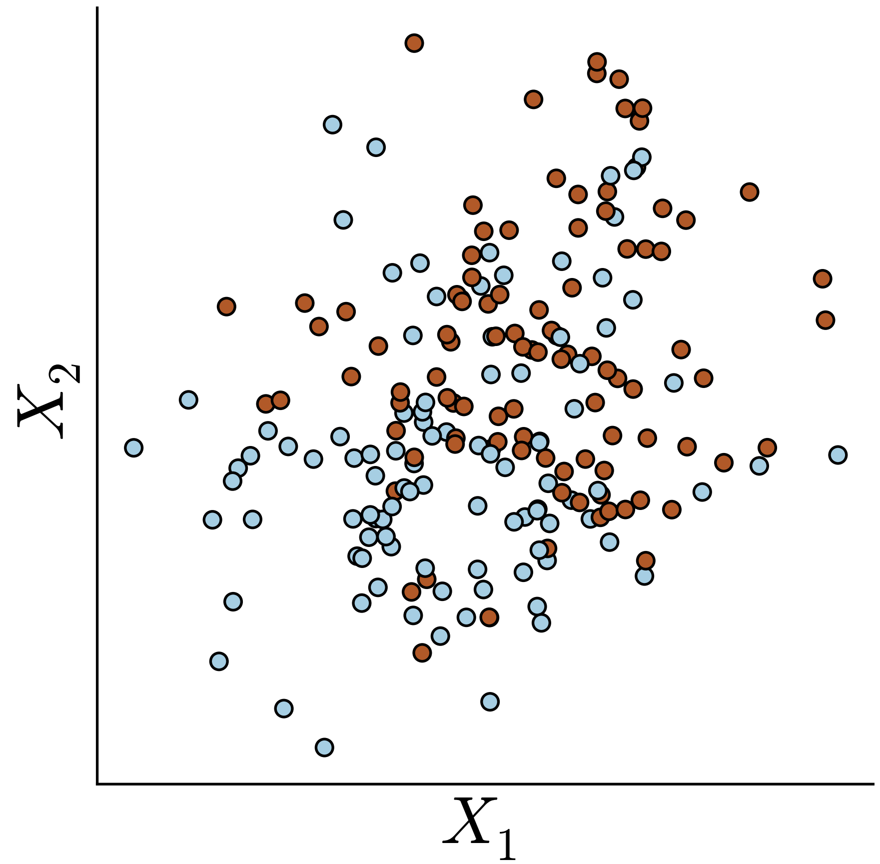
CV vs. Validation: real data



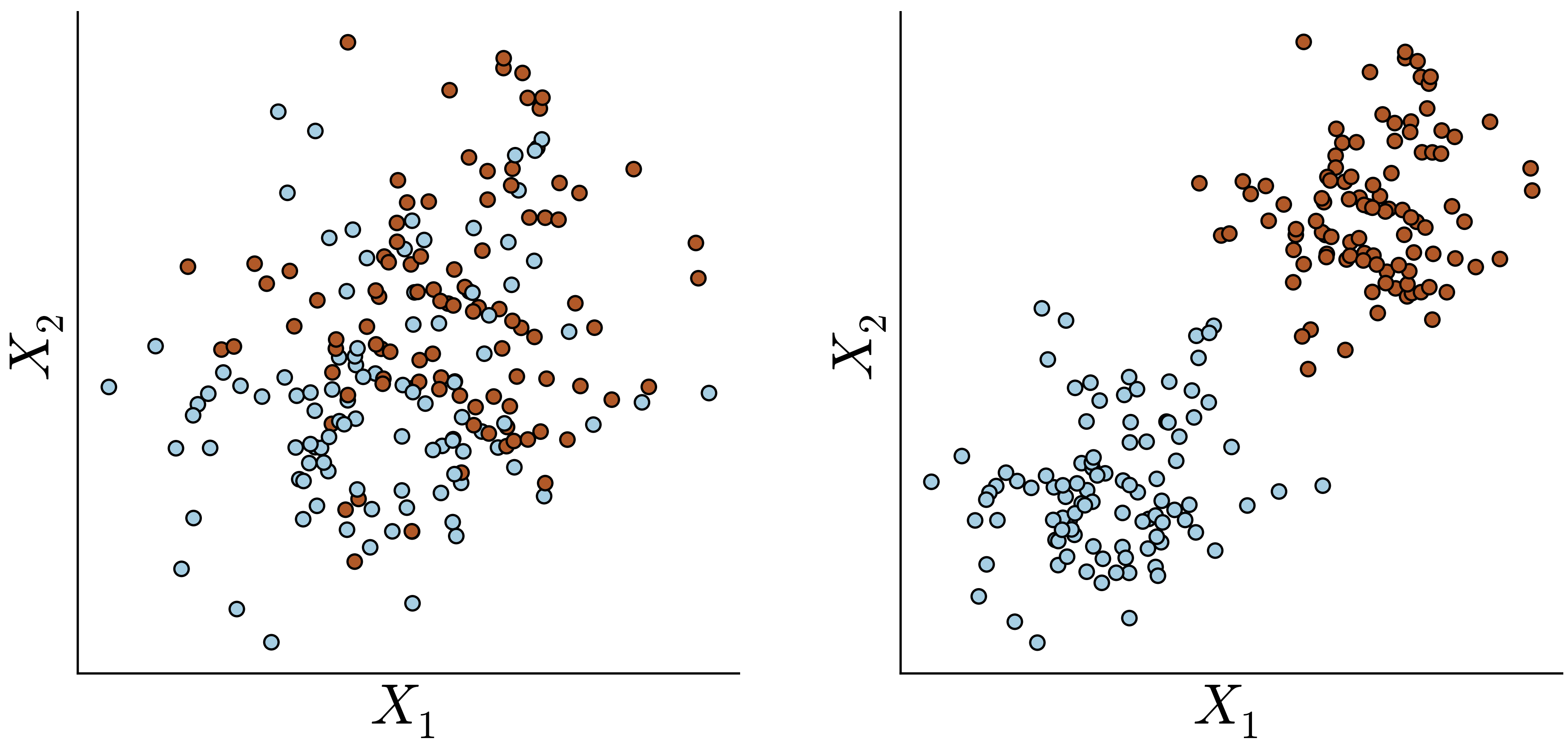
CV vs. Validation: real data



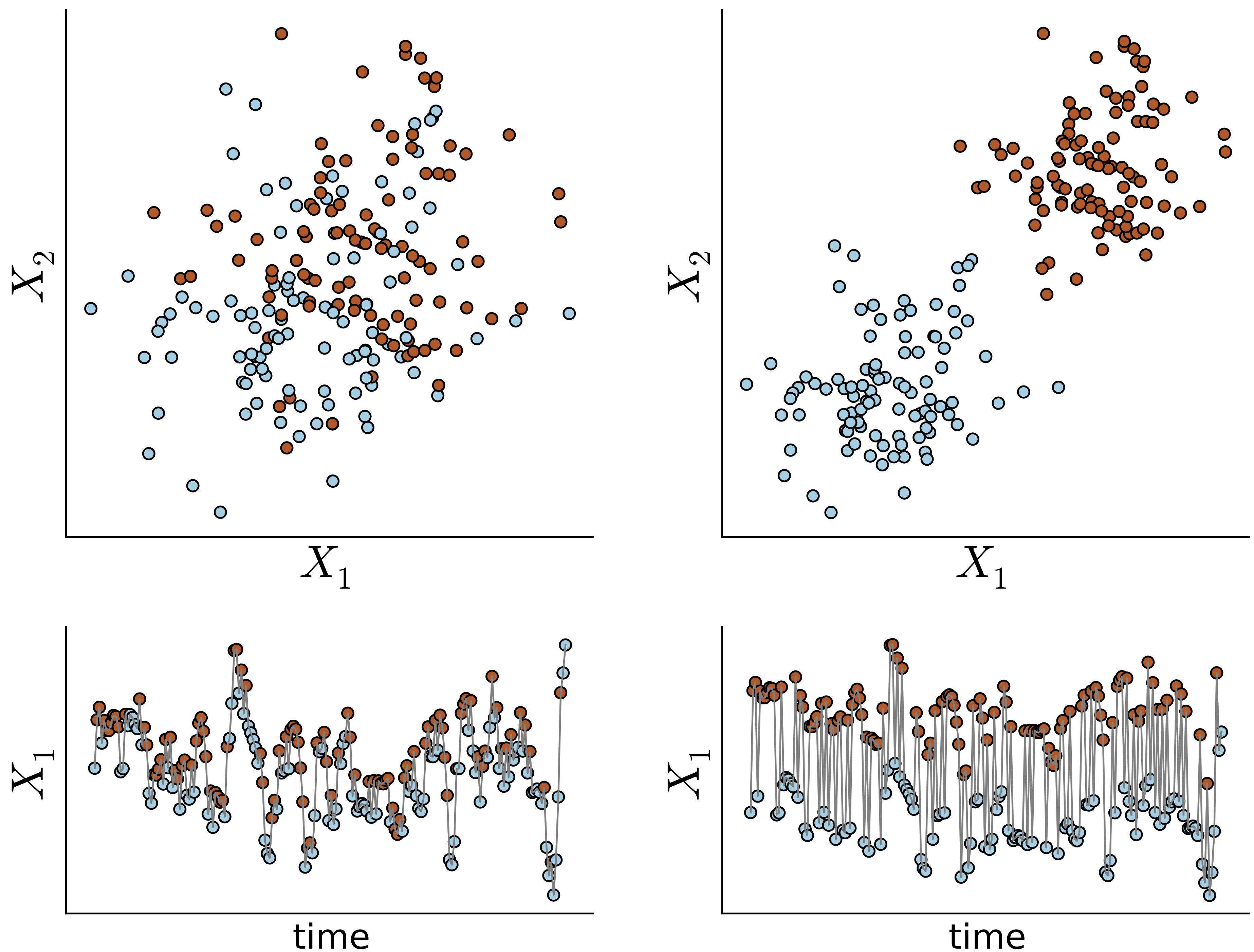
Simulations: known ground truth



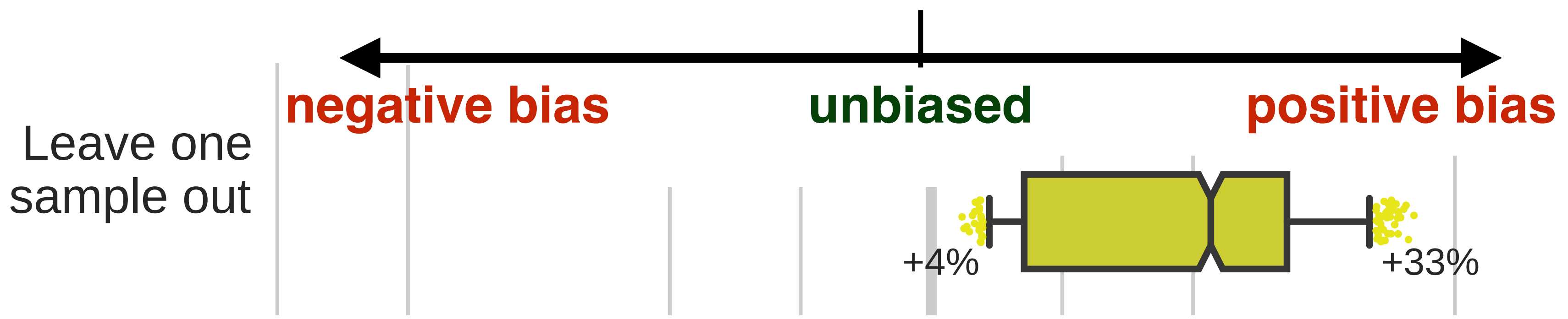
Simulations: known ground truth



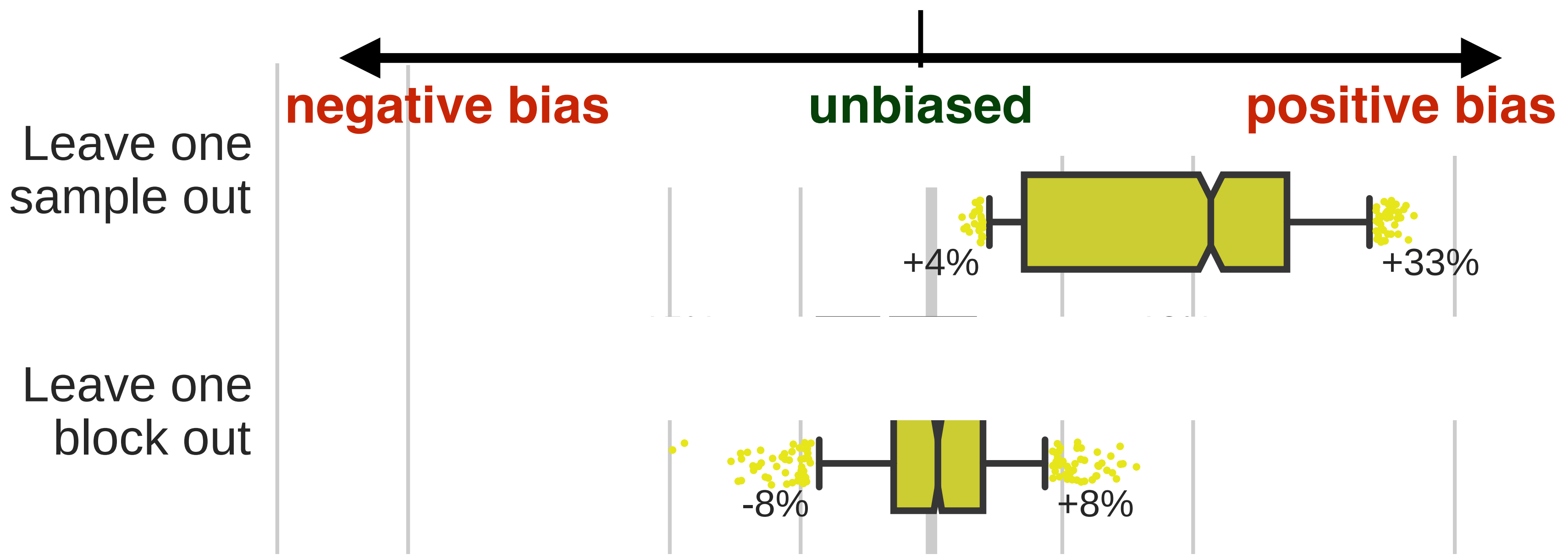
Simulations: known ground truth



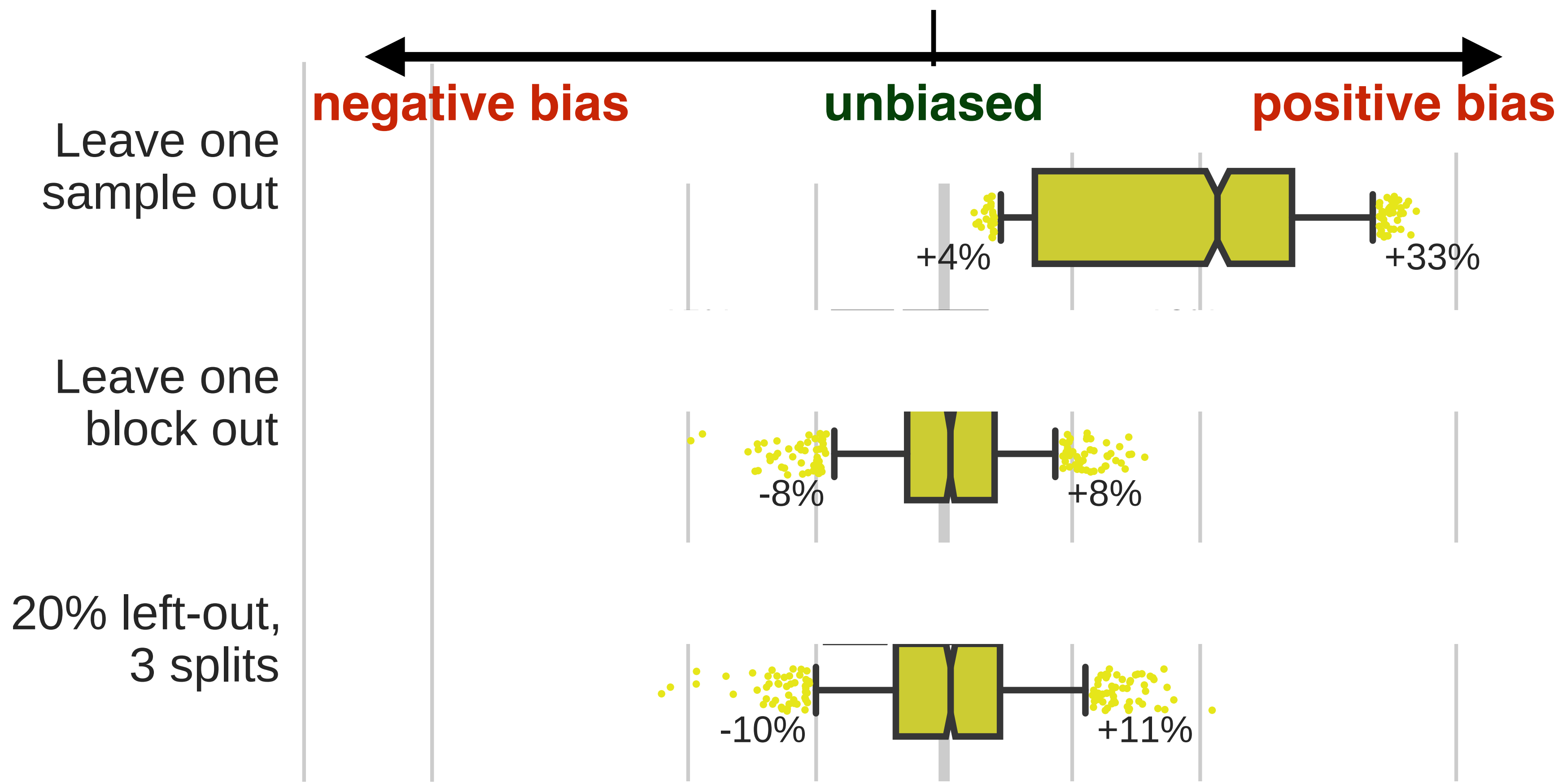
CV vs. Reporting: simulations



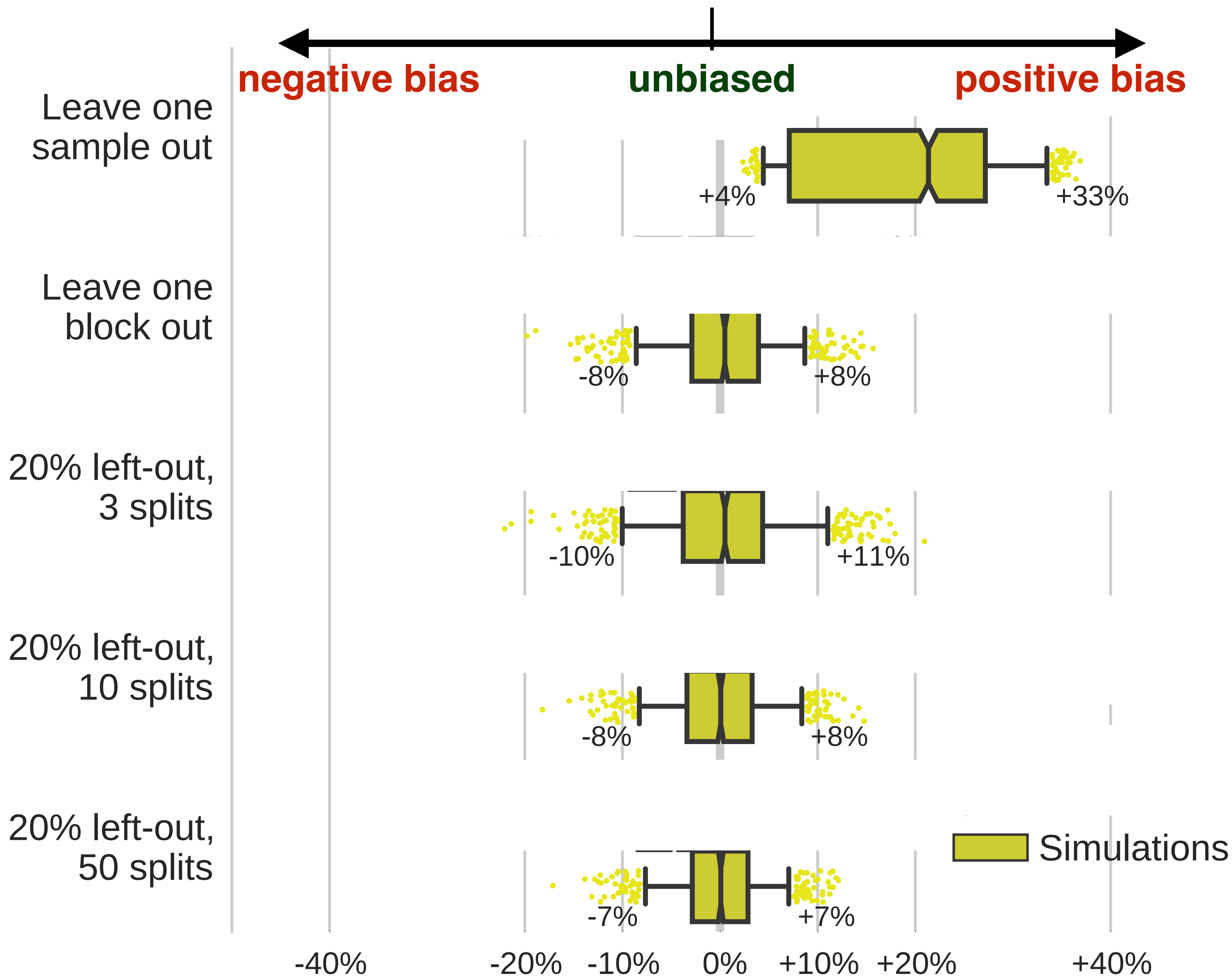
CV vs. Reporting: simulations



CV vs. Reporting: simulations



CV vs. Reporting: simulations



Meta-analysis of Kaggle classification competitions

Roelofs et al, (2019), "*A Meta-Analysis of Overfitting in Machine Learning*", NeurIPS.

Meta-analysis of Kaggle classification competitions

- Kaggle allows the reuse of public "test set" (or reporting set in our terminology)

Roelofs et al, (2019), "*A Meta-Analysis of Overfitting in Machine Learning*", NeurIPS.

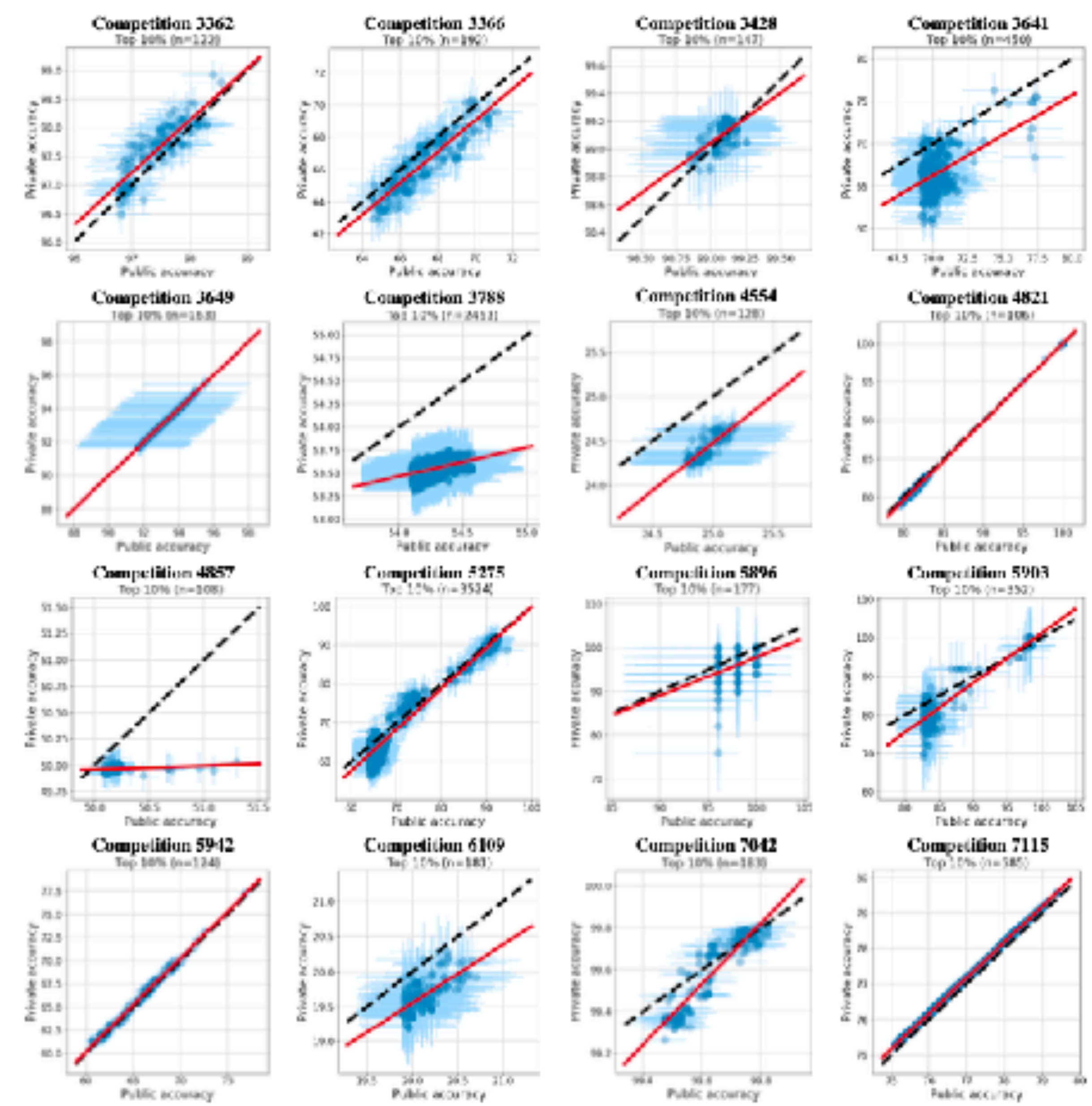
Meta-analysis of Kaggle classification competitions

- Kaggle allows the reuse of public "test set" (or reporting set in our terminology)
    - Roelofs et al studied relation between top 10% model's accuracy on public vs. private "*test sets*"

Roelofs et al, (2019), "*A Meta-Analysis of Overfitting in Machine Learning*", NeurIPS.

Meta-analysis of Kaggle classification competitions

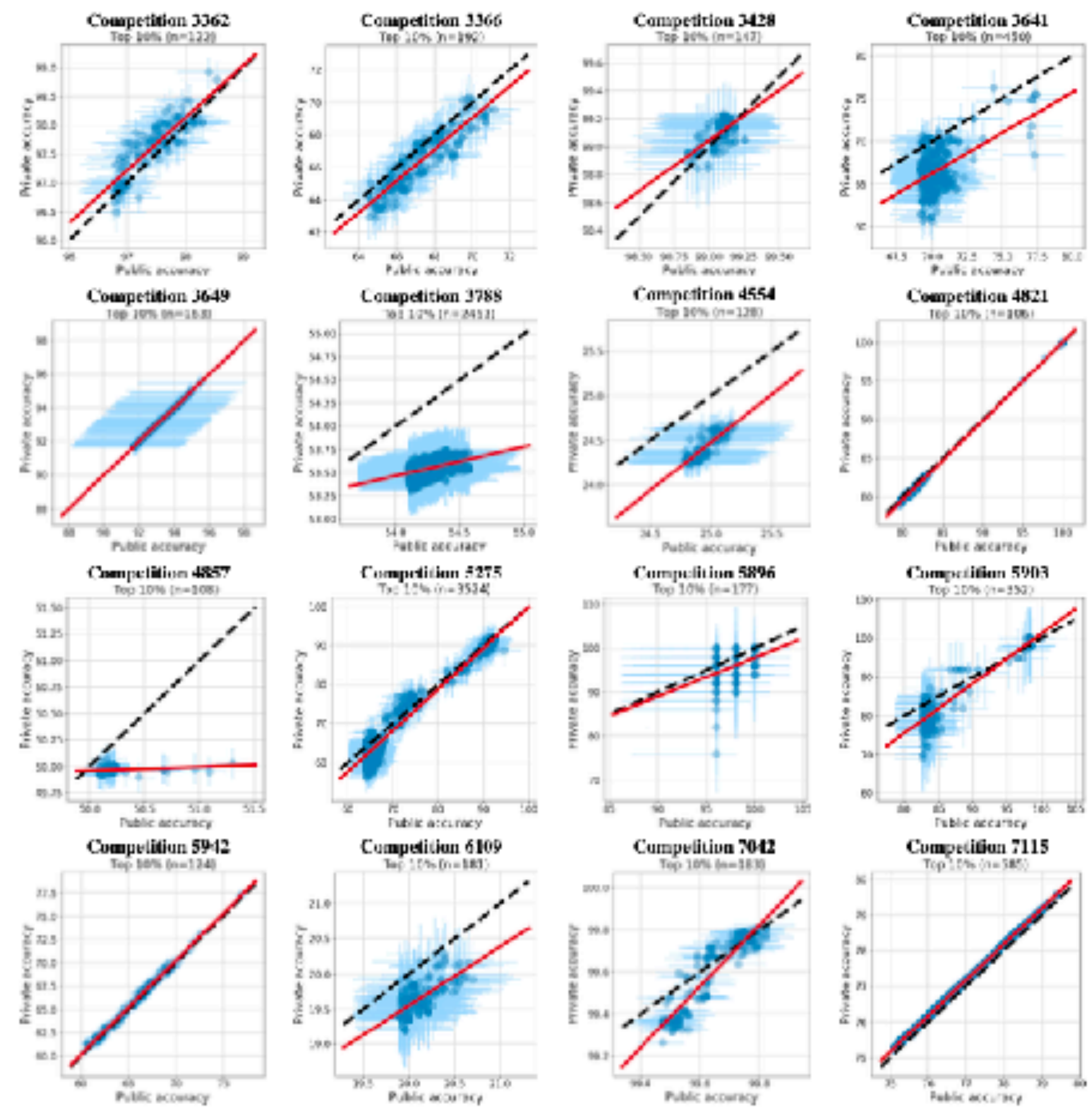
- Kaggle allows the reuse of public “test set” (or reporting set in our terminology)
    - Roelofs et al studied relation between top 10% model’s accuracy on public vs. private “*test sets*”



Roelofs et al, (2019), “*A Meta-Analysis of Overfitting in Machine Learning*”, NeurIPS.

Meta-analysis of Kaggle classification competitions

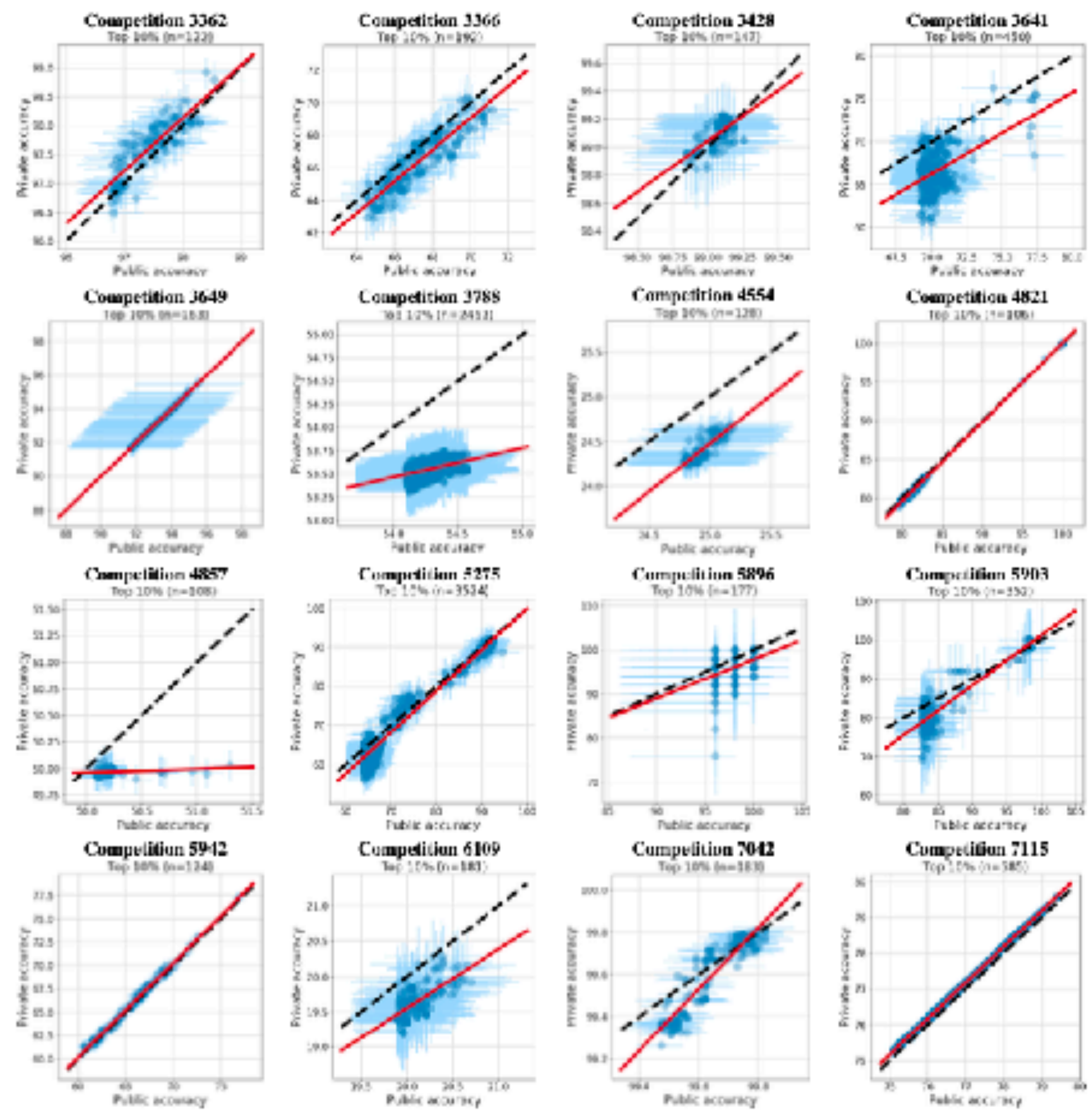
- Kaggle allows the reuse of public "test set" (or reporting set in our terminology)
  - Roelofs et al studied relation between top 10% model's accuracy on public vs. private "*test sets*"
- They observed test set reuse did not show drop in accuracy on the private test set, when



Roelofs et al, (2019), "*A Meta-Analysis of Overfitting in Machine Learning*", NeurIPS.

Meta-analysis of Kaggle classification competitions

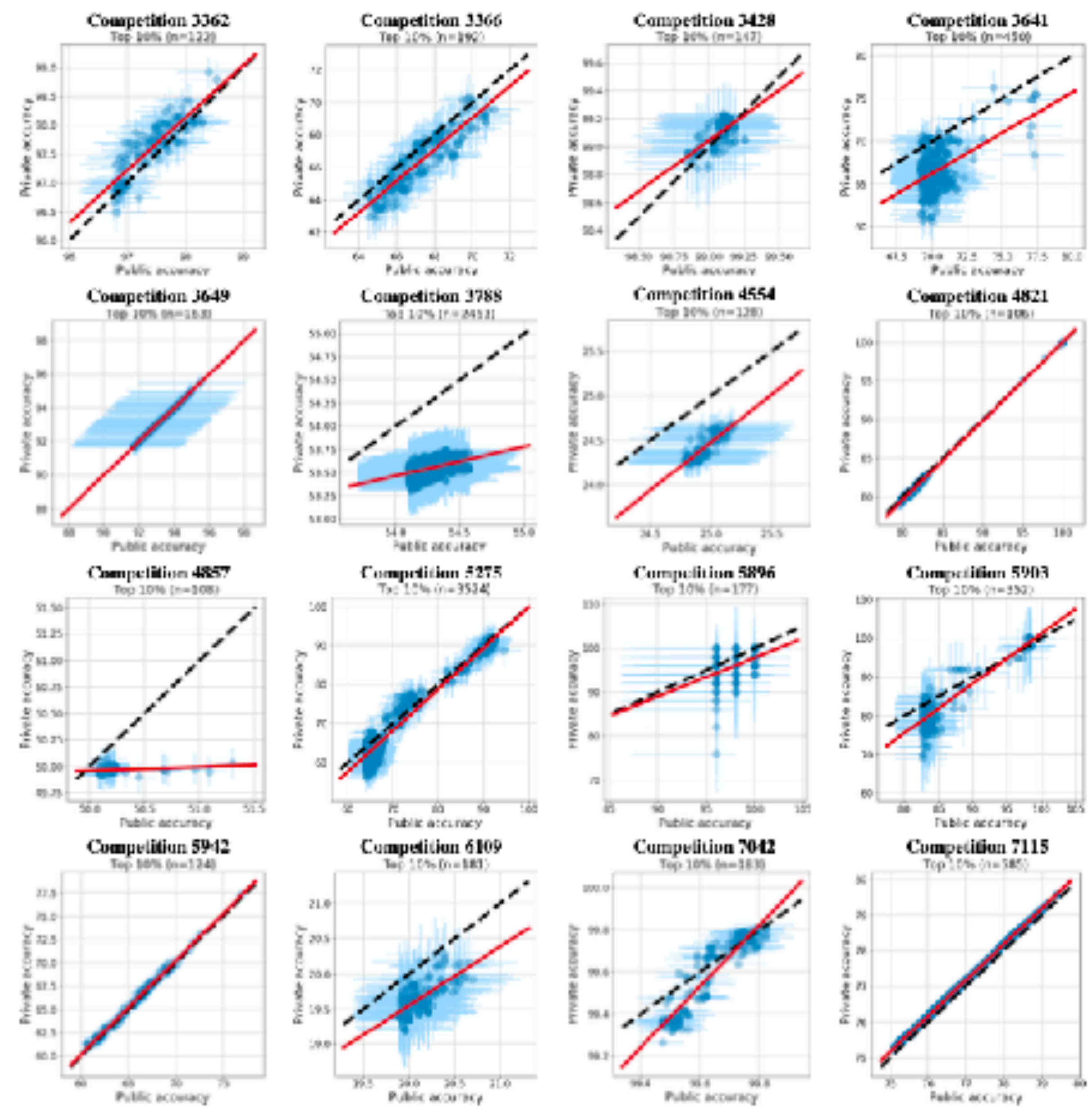
- Kaggle allows the reuse of public "test set" (or reporting set in our terminology)
  - Roelofs et al studied relation between top 10% model's accuracy on public vs. private "*test sets*"
- They observed test set reuse did not show drop in accuracy on the private test set, when
  - dataset splits are `iid` and



Roelofs et al, (2019), "*A Meta-Analysis of Overfitting in Machine Learning*", NeurIPS.

Meta-analysis of Kaggle classification competitions

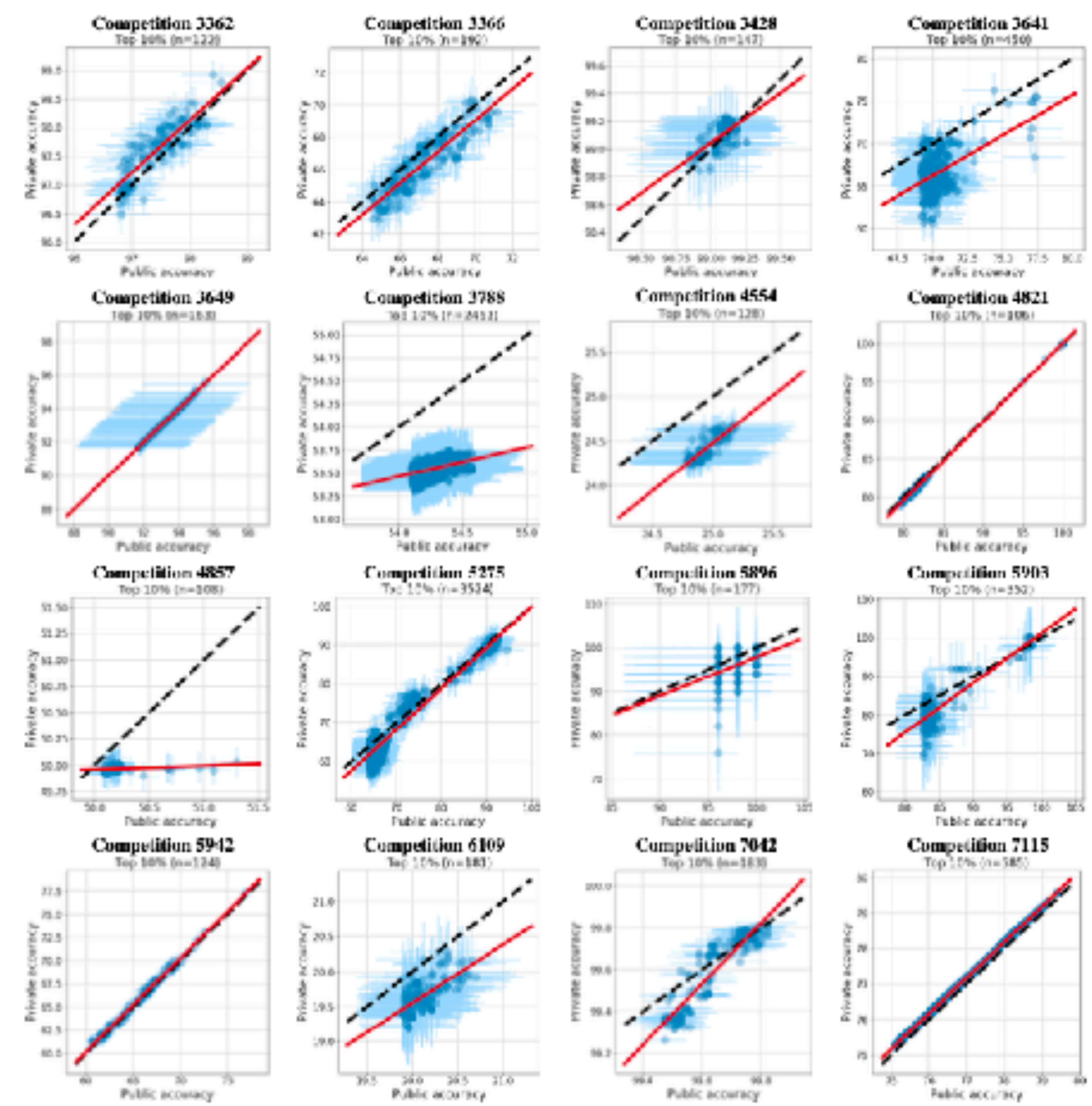
- Kaggle allows the reuse of public "test set" (or reporting set in our terminology)
    - Roelofs et al studied relation between top 10% model's accuracy on public vs. private "*test sets*"
- They observed test set reuse did not show drop in accuracy on the private test set, when
    - dataset splits are `iid` and
    - *test* sets are large: `N > 1K-10K`



Roelofs et al, (2019), "*A Meta-Analysis of Overfitting in Machine Learning*", NeurIPS.

Meta-analysis of Kaggle classification competitions

- Kaggle allows the reuse of public "test set" (or reporting set in our terminology)
    - Roelofs et al studied relation between top 10% model's accuracy on public vs. private "*test sets*"
- They observed test set reuse did not show drop in accuracy on the private test set, when
    - dataset splits are `iid` and
    - *test* sets are large: `N > 1K-10K`
- so, repeated holdout CV is a safe choice!



Roelofs et al, (2019), "*A Meta-Analysis of Overfitting in Machine Learning*", NeurIPS.

Confounds or covariates

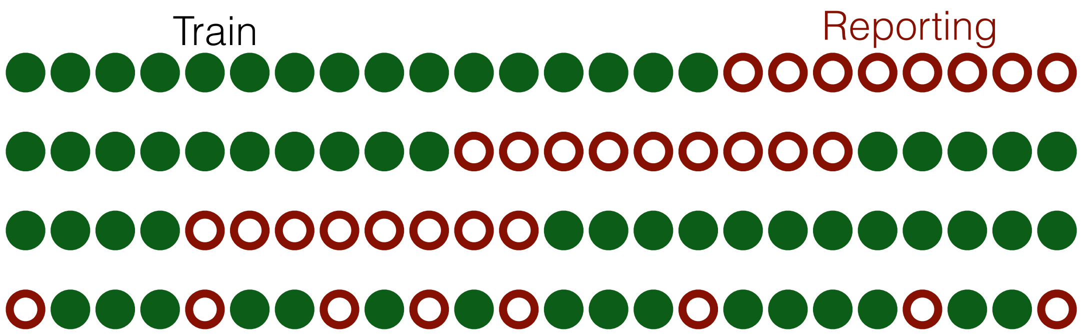
- Deconfounding (regressing out covariates) must be done
  - while respecting
    - the structure of nested CV: methods must be fitted and optimized only on the training set!
    - data types of confounds (numerical vs categorical etc)
  - be it regressing out, or multi-site harmonization,
    - or re- or subsampling the datasets

Commensurability across folds

- It's not enough to properly split each fold, and accurately evaluate classifier performance!

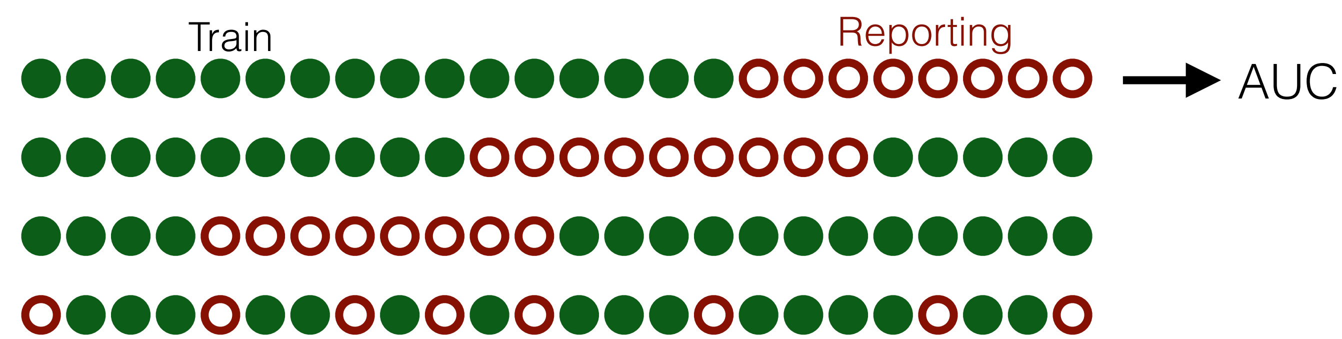
Commensurability across folds

- It's not enough to properly split each fold, and accurately evaluate classifier performance!



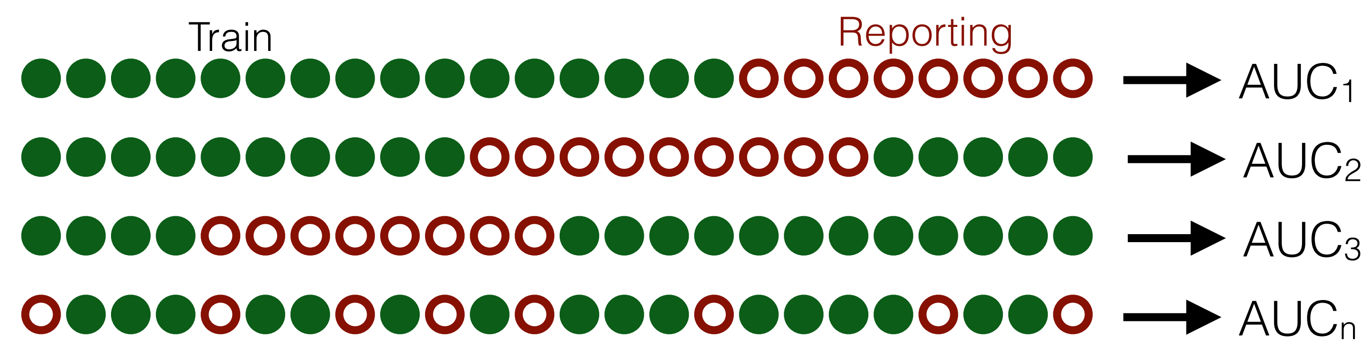
Commensurability across folds

- It's not enough to properly split each fold, and accurately evaluate classifier performance!



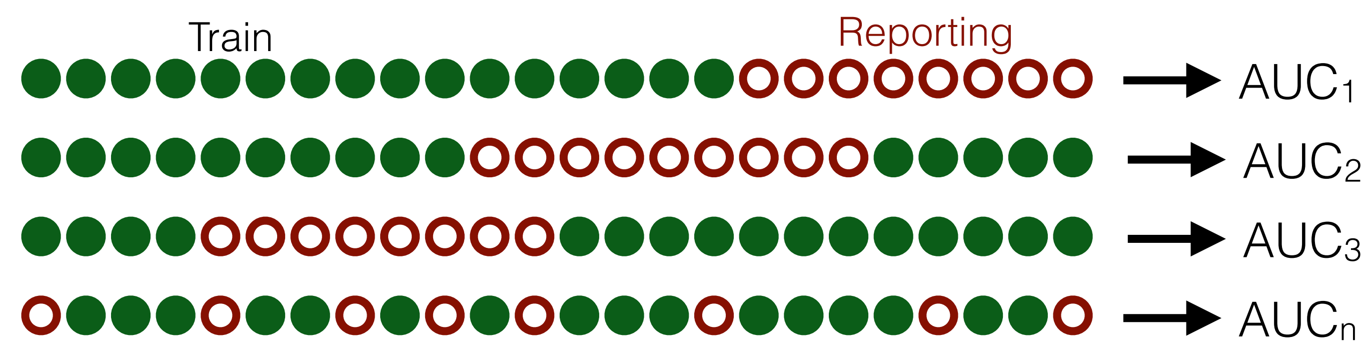
Commensurability across folds

- It's not enough to properly split each fold, and accurately evaluate classifier performance!



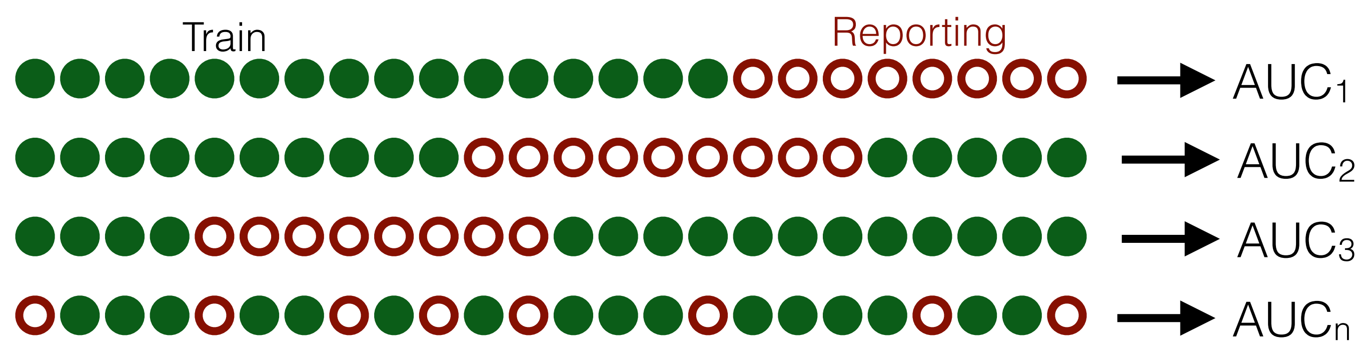
Commensurability across folds

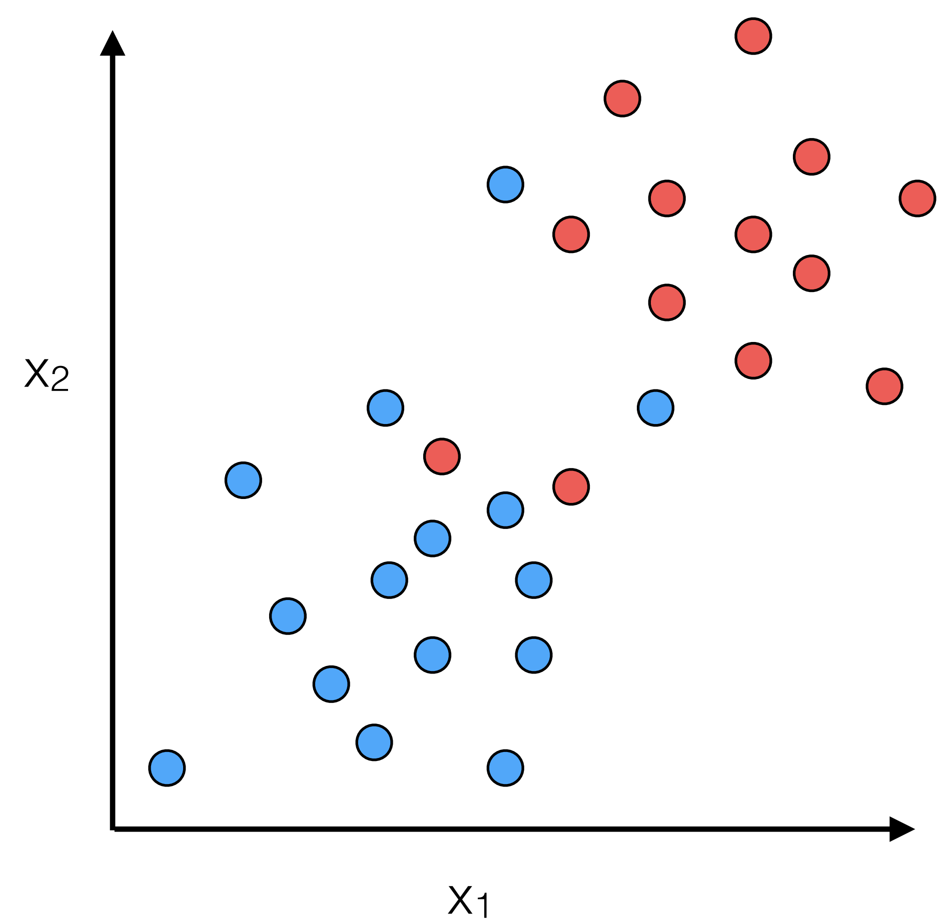
- It's not enough to properly split each fold, and accurately evaluate classifier performance!
- Not all measures across folds are *commensurate*!
    - e.g. decision scores from SVM (reference plane and zero are different!)
    - hence they can not be pooled across folds to construct an ROC!
    - Instead, make ROC per fold and compute AUC per fold, and then average AUC across folds!



Commensurability across folds

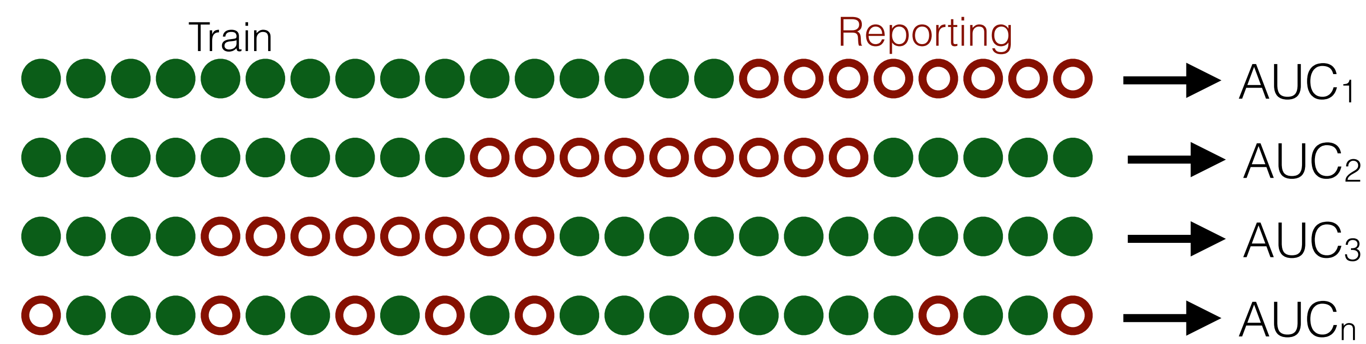
- It's not enough to properly split each fold, and accurately evaluate classifier performance!
- Not all measures across folds are *commensurate*!
  - e.g. decision scores from SVM (reference plane and zero are different!)
  - hence they can not be pooled across folds to construct an ROC!
  - Instead, make ROC per fold and compute AUC per fold, and then average AUC across folds!

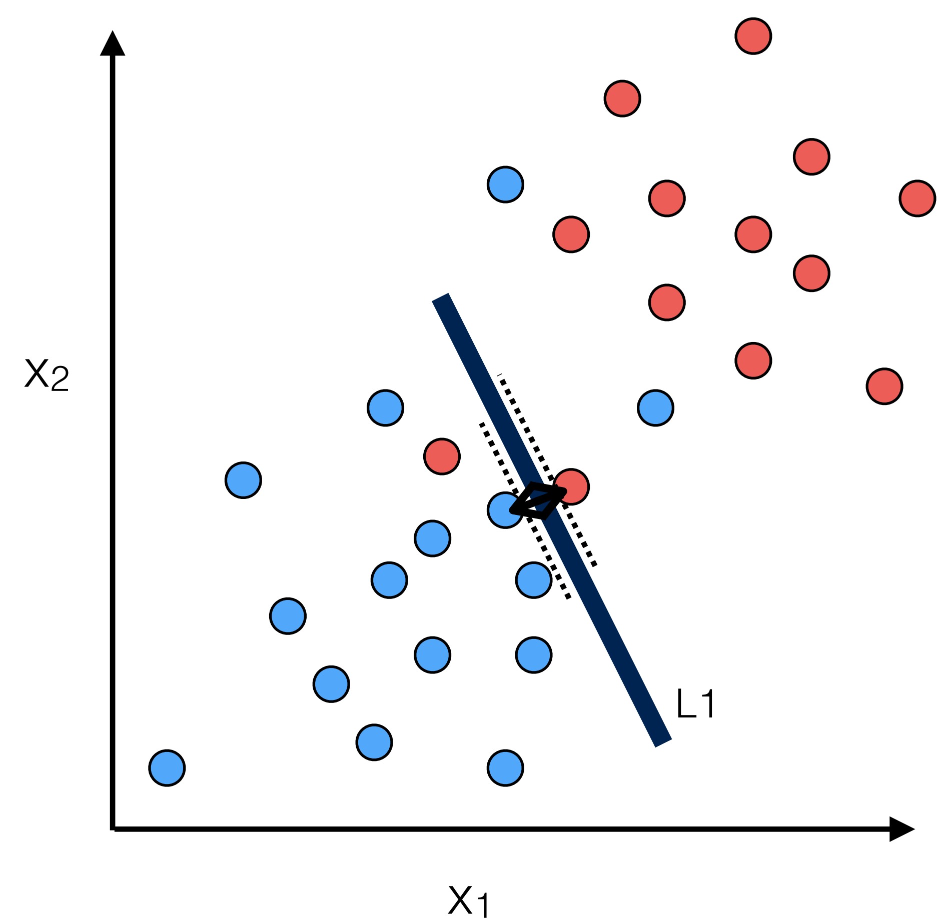




Commensurability across folds

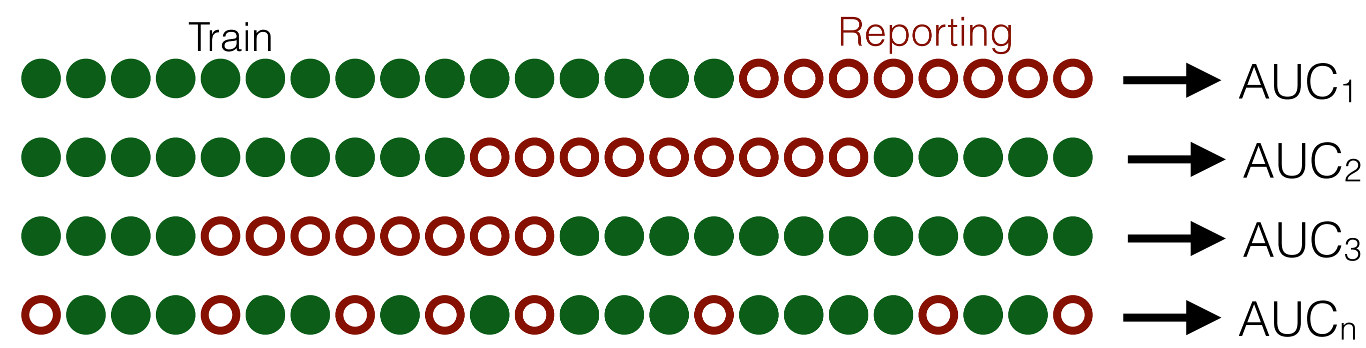
- It's not enough to properly split each fold, and accurately evaluate classifier performance!
- Not all measures across folds are *commensurate*!
  - e.g. decision scores from SVM (reference plane and zero are different!)
  - hence they can not be pooled across folds to construct an ROC!
  - Instead, make ROC per fold and compute AUC per fold, and then average AUC across folds!

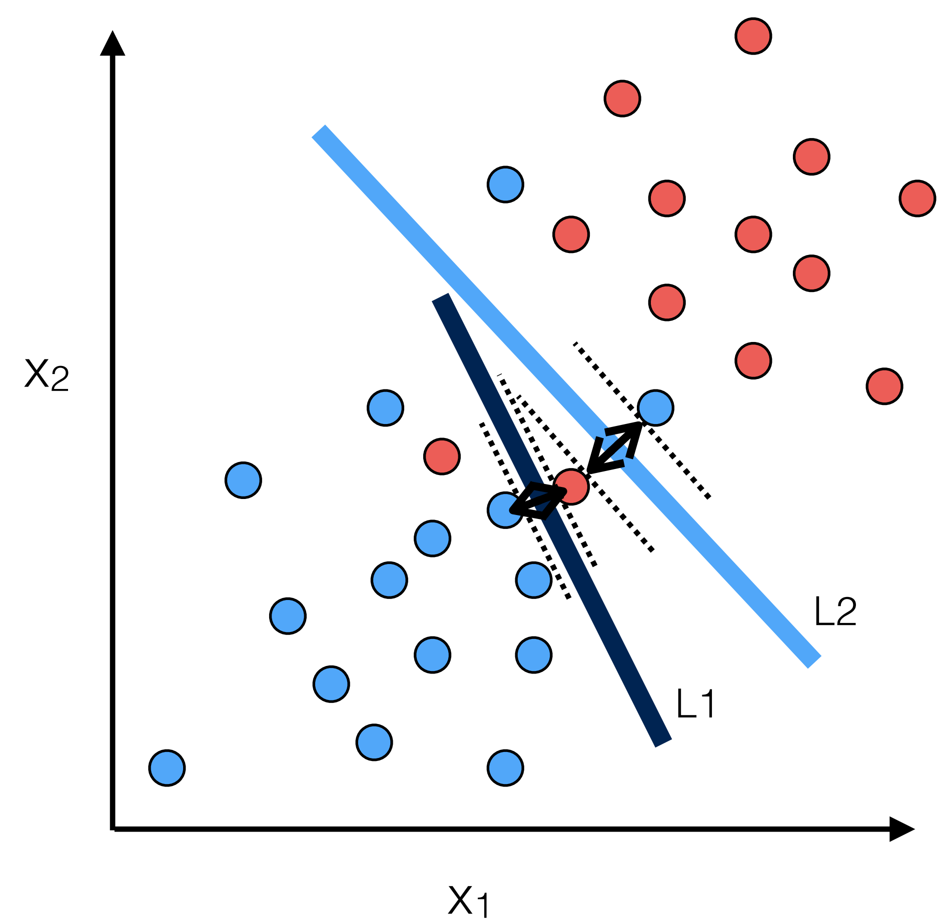




Commensurability across folds

- It's not enough to properly split each fold, and accurately evaluate classifier performance!
- Not all measures across folds are *commensurate*!
    - e.g. decision scores from SVM (reference plane and zero are different!)
    - hence they can not be pooled across folds to construct an ROC!
    - Instead, make ROC per fold and compute AUC per fold, and then average AUC across folds!





Performance Metrics

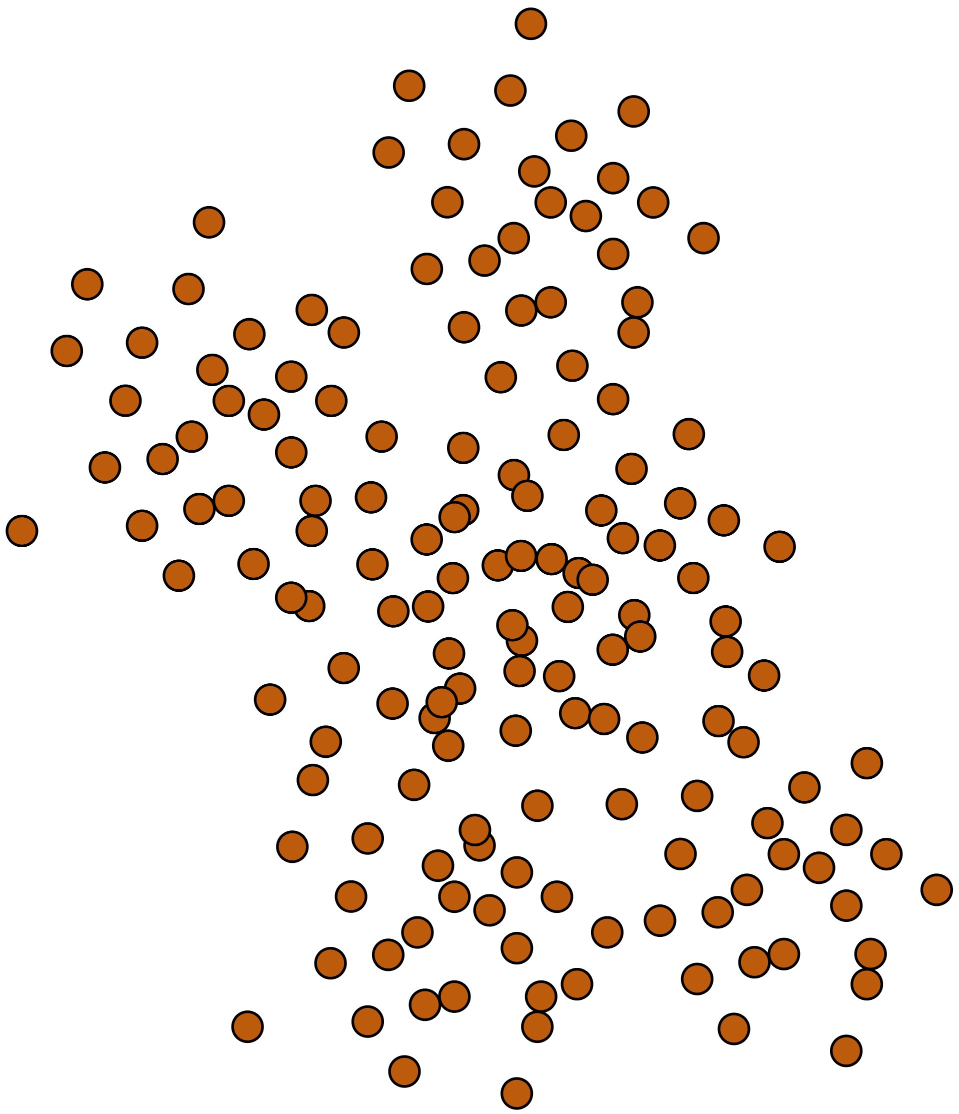
Metric	Commensurate across folds?	Advantages	Disadvantages
Accuracy / Error rate	Yes	Universally applicable; Multi-class;	Sensitive to class- and cost-imbalance
Area under ROC (AUC)	Only when ROC is computed within fold	Averages over all ratios of misclassification costs	Not easily extendable to multi-class problems
F1 score	Yes	Information retrieval	Does not take true negatives into account
Mean Squared Error (MSE)	Yes (within the same dataset/scales)	Intuitive	Not commensurable across different target scales!

Subtle Sources of Bias in CV

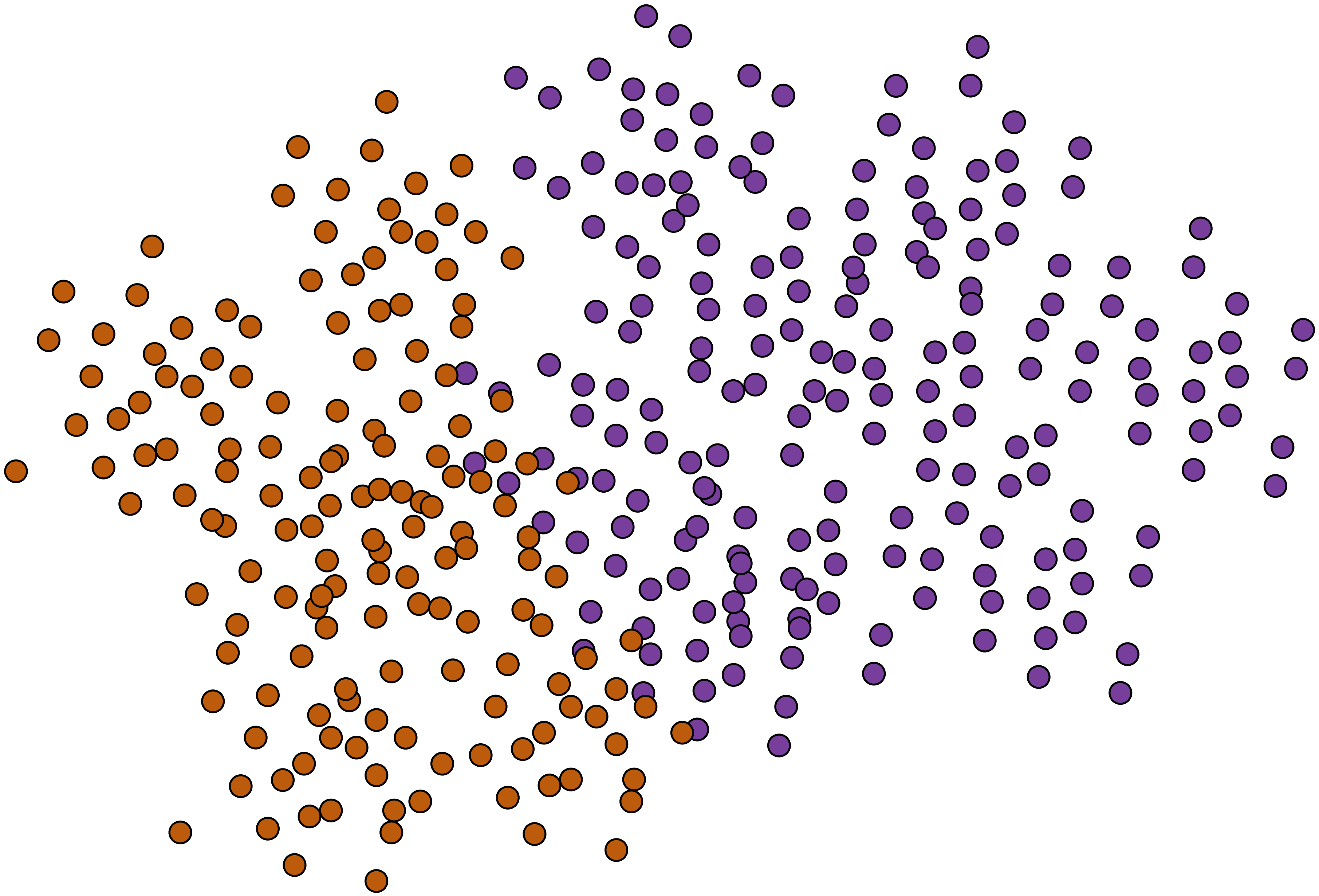
Type*	Approach	sexy name I made up	How to avoid it?
***k*-hacking**	Try many *k*'s in k-fold CV (or different training %) and report only the best	*k-hacking*	Pick *k*=10, repeat it *many* times (n>200 or as many as possible) and report the full distribution (not box plots)
metric-hacking	Try different performance metrics (accuracy, AUC, F1, error rate), and report the best	*m-hacking*	Choose the most appropriate and recognized metric for the problem e.g. AUC for binary classification etc
ROI-hacking	Assess many ROIs (or their features, or combinations), but report only the best	*r-hacking*	Adopt a whole-brain data-driven approach to discover best ROIs within an inner CV, then report their out-of-sample predictive accuracy
feature- or dataset-hacking	Try subsets of feature[s] or subsamples of dataset[s], but report only the best	*d-hacking*	Use and report on **everything:** all analyses on all datasets, try inter-dataset CV, run non-parametric statistical comparisons!

*exact incidence of these hacking approaches is unknown, but non-zero.

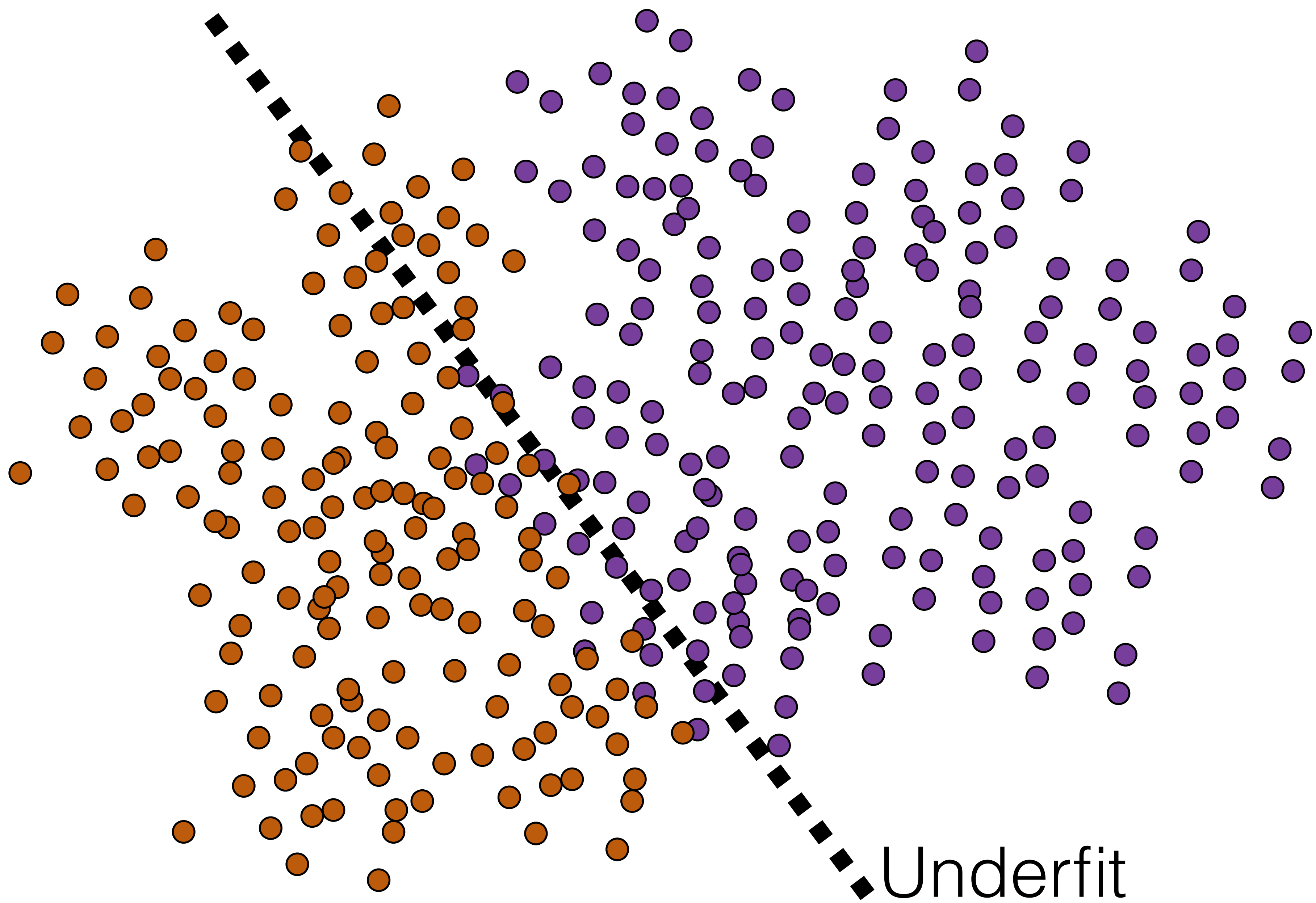
Overfitting



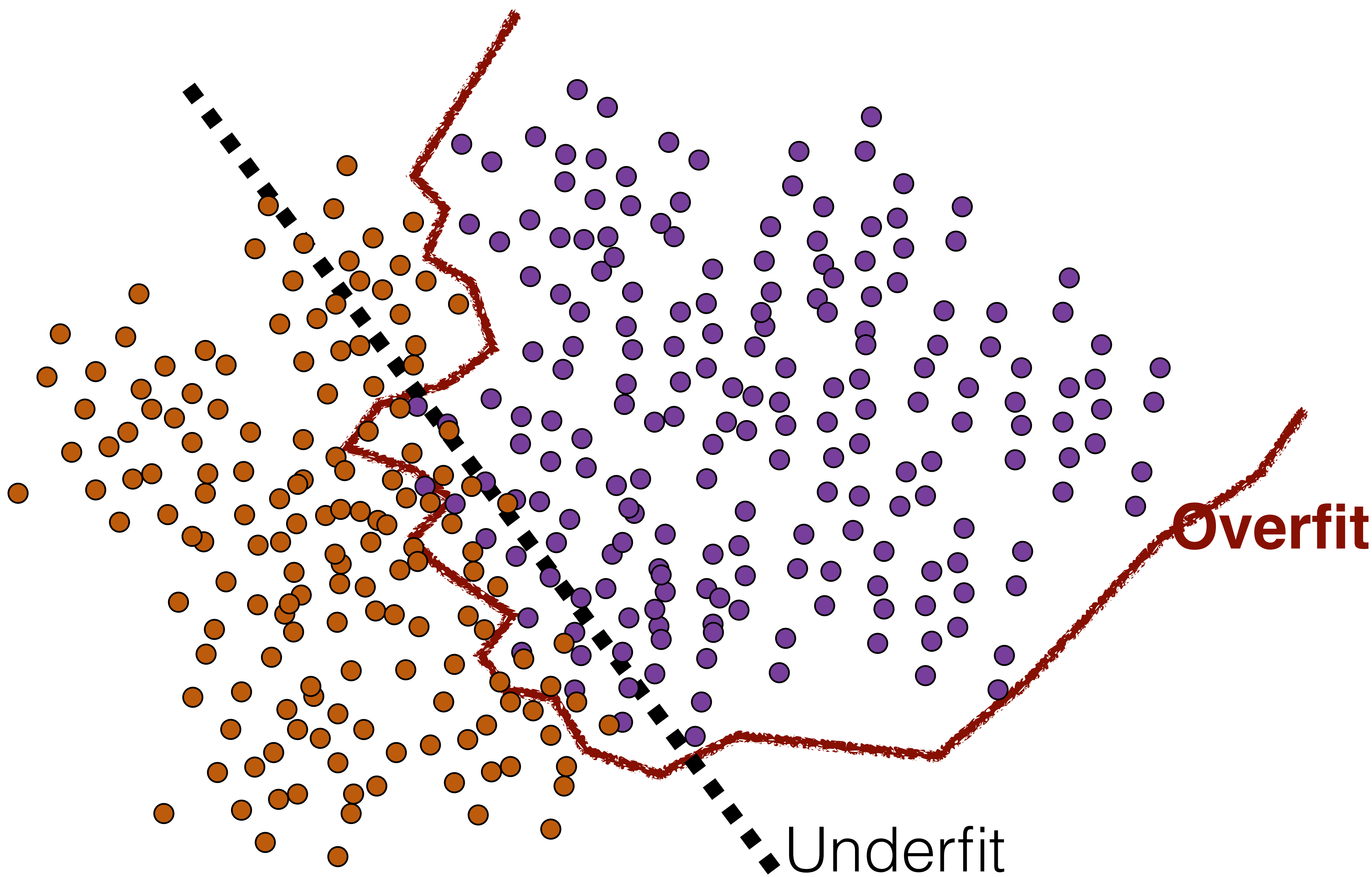
Overfitting



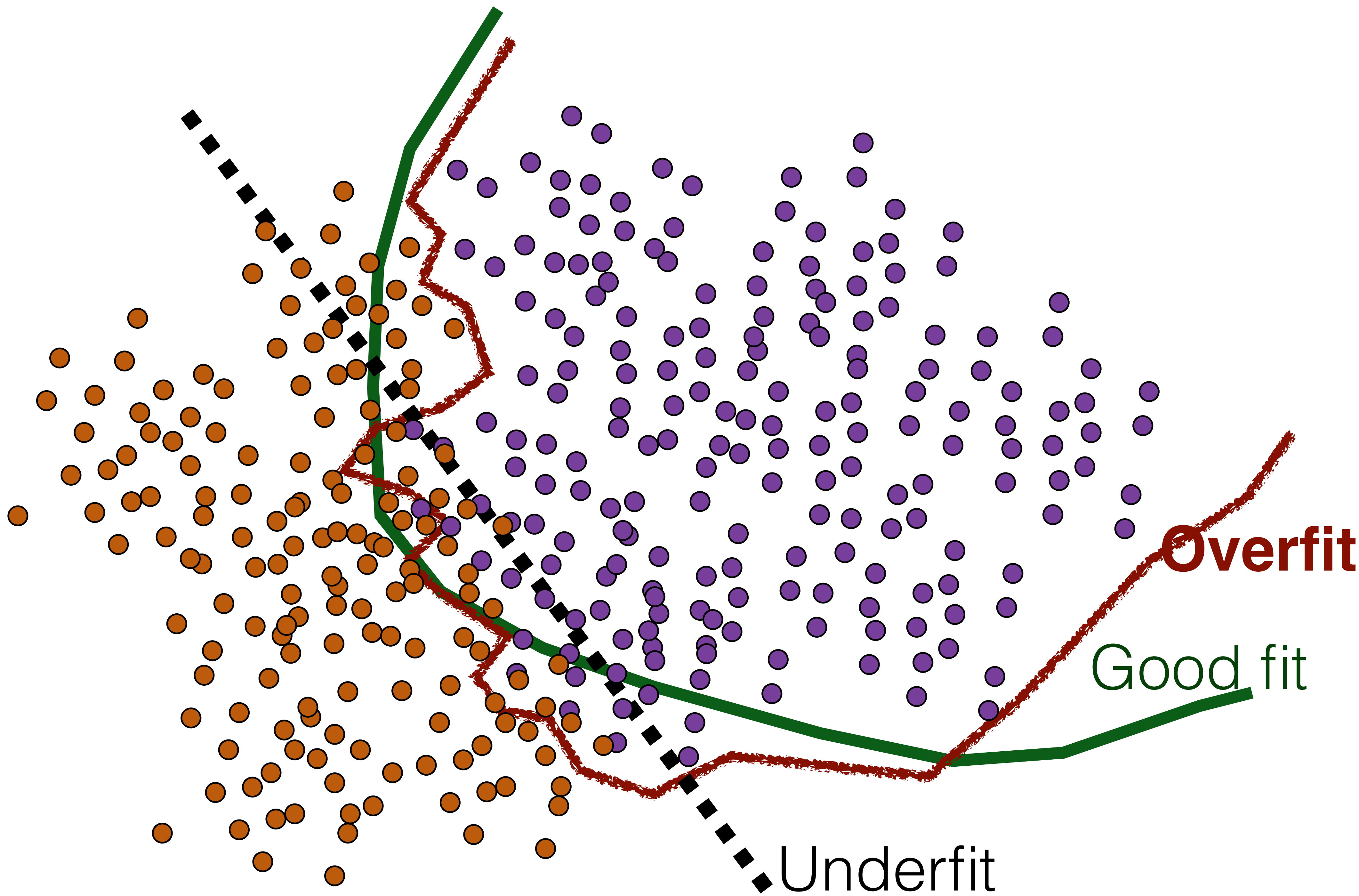
Overfitting



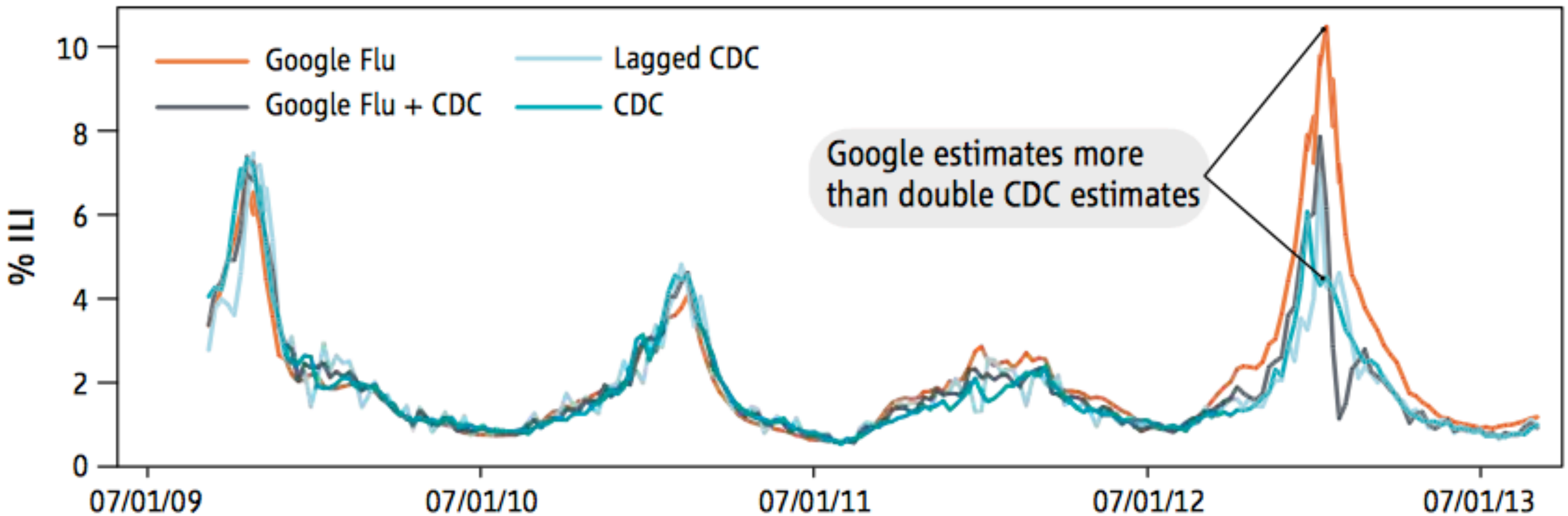
Overfitting



Overfitting

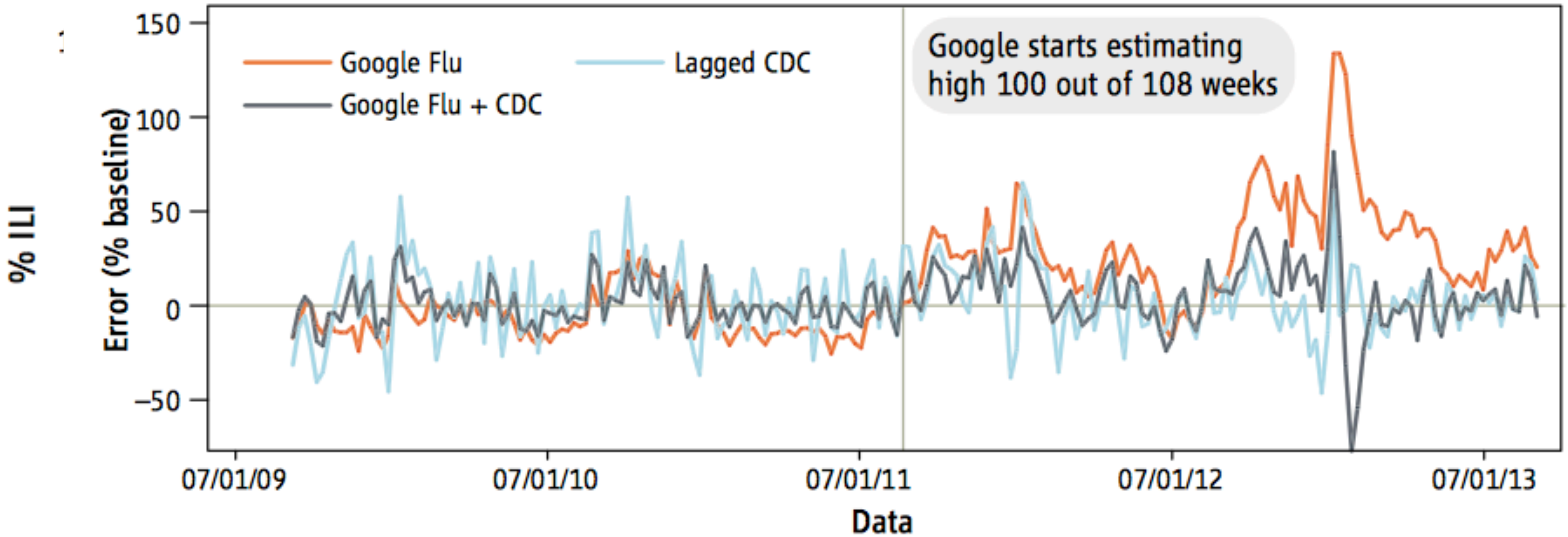


50 shades of overfitting



Reference: David Lazer, Ryan Kennedy, Gary King, Alessandro Vespignani. 2014. "The Parable of Google Flu: Traps in Big Data Analysis." Science, 14 March, 343: 1203-1205.

50 shades of overfitting



GFT overestimation. GFT overestimated the prevalence of flu in the 2012–2013 season and overshot the actual level in 2011–2012 by more than 50%. From 21 August 2011 to 1 September 2013, GFT reported overly high flu prevalence 100 out of 108 weeks. **(Top)** Estimates of doctor visits for ILI. "Lagged CDC" incorporates 52-week seasonality variables with lagged CDC data. "Google Flu + CDC" combines GFT, lagged CDC estimates, lagged error of GFT estimates, and 52-week seasonality variables. **(Bottom)** Error [as a percentage {[Non-CDC estmate)−(CDC estimate)]/(CDC) estimate)}. Both alternative models have much less error than GFT alone. Mean absolute error (MAE) during the out-of-sample period is 0.486 for GFT, 0.311 for lagged CDC, and 0.232 for combined GFT and CDC. All of these differences are statistically significant at $P < 0.05$. See SM.

Reference: David Lazer, Ryan Kennedy, Gary King, Alessandro Vespignani. 2014. "The Parable of Google Flu: Traps in Big Data Analysis." Science, 14 March, 343: 1203-1205.

“Clever forms of overfitting”

Name	Method	Explanation	Remedy
Traditional overfitting	Train a complex predictor on too-few examples.		1. Hold out pristine examples for testing. 2. Use a simpler predictor. 3. Get more training examples. 4. Integrate over many predictors. 5. Reject papers which do this.
Parameter tweak overfitting	Use a learning algorithm with many parameters. Choose the parameters based on the test set performance.	For example, choosing the features so as to optimize test set performance can achieve this.	Same as above.
Brittle measure	Use a measure of performance which is especially brittle to overfitting.	"entropy", "mutual information", and leave-one-out cross-validation are all surprisingly brittle. This is particularly severe when used in conjunction with another approach.	Prefer less brittle measures of performance.
Bad statistics	Misuse statistics to overstate confidences.	One common example is pretending that cross validation performance is drawn from an i.i.d. gaussian, then using standard confidence intervals. Cross validation errors are not independent. Another standard method is to make known-false assumptions about some system and then derive excessive confidence.	Don't do this. Reject papers which do this.
Choice of measure	Choose the best of Accuracy, error rate, (A)ROC, F1, percent improvement on the previous best, percent improvement of error rate, etc.. for your method. For bonus points, use ambiguous graphs.	This is fairly common and tempting.	Use canonical performance measures. For example, the performance measure directly motivated by the problem.
Incomplete Prediction	Instead of (say) making a multiclass prediction, make a set of binary predictions, then compute the optimal multiclass prediction.	Sometimes it's tempting to leave a gap filled in by a human when you don't otherwise succeed.	Reject papers which do this.
Human-loop overfitting.	Use a human as part of a learning algorithm and don't take into account overfitting by the entire human/computer interaction.	This is subtle and comes in many forms. One example is a human using a clustering algorithm (on training and test examples) to guide learning algorithm choice.	Make sure test examples are not available to the human.
Data set selection	Chose to report results on some subset of datasets where your algorithm performs well.	The reason why we test on natural datasets is because we believe there is some structure captured by the past problems that helps on future problems. Data set selection subverts this and is very difficult to detect.	Use comparisons on standard datasets. Select datasets without using the test set. Good Contest performance can't be faked this way.
Reprobleming	Alter the problem so that your performance improves.	For example, take a time series dataset and use cross validation. Or, ignore asymmetric false positive/false negative costs. This can be completely unintentional, for example when someone uses an ill-specified UCI dataset.	Discount papers which do this. Make sure problem specifications are clear.
Old datasets	Create an algorithm for the purpose of improving performance on old datasets.	After a dataset has been released, algorithms can be made to perform well on the dataset using a process of feedback design, indicating better performance than we might expect in the future. Some conferences have canonical datasets that have been used for a decade...	Prefer simplicity in algorithm design. Weight newer datasets higher in consideration. Making test examples not publicly available for datasets slows the feedback design process but does not eliminate it.
Overfitting by review	10 people submit a paper to a conference. The one with the best result is accepted.	This is a systemic problem which is very difficult to detect or eliminate. We want to prefer presentation of good results, but doing so can result in overfitting.	1. Be more pessimistic of confidence statements by papers at high rejection rate conferences. 2. Some people have advocated allowing the publishing of methods with poor performance. (I have doubts this would work.)

from http://hunch.net/?p=22

Limitations of CV

- Number of CV repetitions increases with

Limitations of CV

- Number of CV repetitions increases with
  - sample size: larger sample —> more repetitions

Limitations of CV

- Number of CV repetitions increases with
  - sample size: larger sample —> more repetitions
    - This can be an issue if the model training or evaluation is computationally expensive.

Limitations of CV

- Number of CV repetitions increases with
  - sample size: larger sample —> more repetitions
    - This can be an issue if the model training or evaluation is computationally expensive.
  - number of model parameters, exponentially

Limitations of CV

- Number of CV repetitions increases with
    - sample size: larger sample —> more repetitions
        - This can be an issue if the model training or evaluation is computationally expensive.
    - number of model parameters, exponentially
        - to choose the best combination!

Recommendations

1. Ensure the tuning and reporting sets are ***truly*** independent of the training set!
    - easy to commit mistakes in complicated analyses!

Recommendations

1. Ensure the tuning and reporting sets are ***truly*** independent of the training set!
    - easy to commit mistakes in complicated analyses!
2. Use repeated-holdout (10-50% for tuning and reporting sets)
    - respecting sample/dependency structure
    - ensuring independence between train & test sets

Recommendations

1. Ensure the tuning and reporting sets are ***truly*** independent of the training set!
    - easy to commit mistakes in complicated analyses!
2. Use repeated-holdout (10-50% for tuning and reporting sets)
    - respecting sample/dependency structure
    - ensuring independence between train & test sets
3. Handle confounds properly within nested-CV without double-dipping

Recommendations

1. Ensure the tuning and reporting sets are ***truly*** independent of the training set!
    - easy to commit mistakes in complicated analyses!
2. Use repeated-holdout (10-50% for tuning and reporting sets)
    - respecting sample/dependency structure
    - ensuring independence between train & test sets
3. Handle confounds properly within nested-CV without double-dipping
4. Choose your performance metric correctly!
    - Pool it across folds accurately.

Recommendations

1. Ensure the tuning and reporting sets are ***truly*** independent of the training set!
    - easy to commit mistakes in complicated analyses!
2. Use repeated-holdout (10-50% for tuning and reporting sets)
    - respecting sample/dependency structure
    - ensuring independence between train & test sets
3. Handle confounds properly within nested-CV without double-dipping
4. Choose your performance metric correctly!
    - Pool it across folds accurately.
5. Use largest reporting sets and number of repetitions when possible
    - Not possible with leave-one-sample-out.

Conclusions

Conclusions

- CV is necessary to estimate out-of-sample predictive performance
  - Results could vary considerably with a different CV scheme
  - CV results can have variance (>10%)

Conclusions

- CV is necessary to estimate out-of-sample predictive performance
  - Results could vary considerably with a different CV scheme
  - CV results can have variance (>10%)
- Document CV scheme in detail:
  - type of split
  - number of repetitions
  - Full distribution of estimates

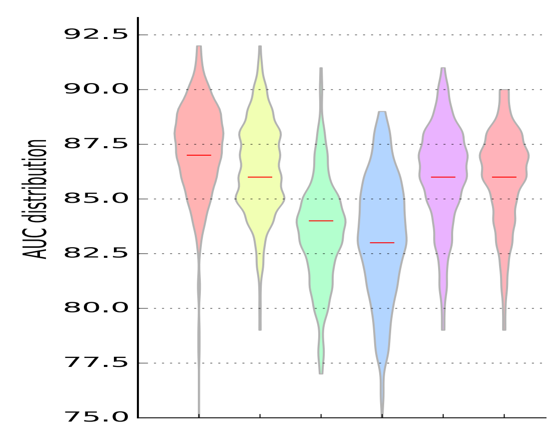
Conclusions

- CV is necessary to estimate out-of-sample predictive performance
  - Results could vary considerably with a different CV scheme
  - CV results can have variance (>10%)
- Document CV scheme in detail:
  - type of split
  - number of repetitions
  - Full distribution of estimates
- Proper splitting is not enough, proper pooling is needed too.

Conclusions

- CV is necessary to estimate out-of-sample predictive performance
  - Results could vary considerably with a different CV scheme
  - CV results can have variance (>10%)
- Document CV scheme in detail:
  - type of split
  - number of repetitions
  - Full distribution of estimates
- Proper splitting is not enough, proper pooling is needed too.

- **Bad** examples:
  - just mean: μ%
  - std. dev.: $\mu \pm \sigma$%
- **Good** examples:
  - Using 100 iterations of repeated holdout CV with 80% reserved for training+tuning, we obtain the following distribution of AUC.



References

- Arlot, S., & Celisse, A. (2010). *A survey of cross-validation procedures for model selection*. Statistics Surveys, 4, 40–79.
- Varoquaux, G., Raamana, P. R., Engemann, D. A., Hoyos-Idrobo, A., Schwartz, Y., & Thirion, B. (2016). *Assessing and tuning brain decoders: cross-validation, caveats, and guidelines*. NeuroImage. http://doi.org/10.1016/j.neuroimage.2016.10.038
- Forman, G. (2010). *Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement*. ACM SIGKDD Explorations Newsletter.
- Roelofs, R., Shankar, V., Recht, B., Fridovich-Keil, S., Hardt, M., Miller, J., & Schmidt, L. (2019). *A Meta-Analysis of Overfitting in Machine Learning*. In Advances in Neural Information Processing Systems (pp. 9175-9185)

neuropredict

CONFOUNDS

Follow @raamana_

neuropredict

CONFOUNDS

Follow @raamana_

Join me to help me with
`confounds` or `neuropredict`

during OHBM hackathon
as well as open science rooms

neuropredict

CONFOUNDS

Follow @raamana_

Join me to help me with
`confounds` or `neuropredict`

during OHBM hackathon
as well as open science rooms

`github.com/raamana/neuropredict`

`github.com/raamana/confounds`

Acknowledgements

Baycrest

ONTARIO BRAIN INSTITUTE

SFU

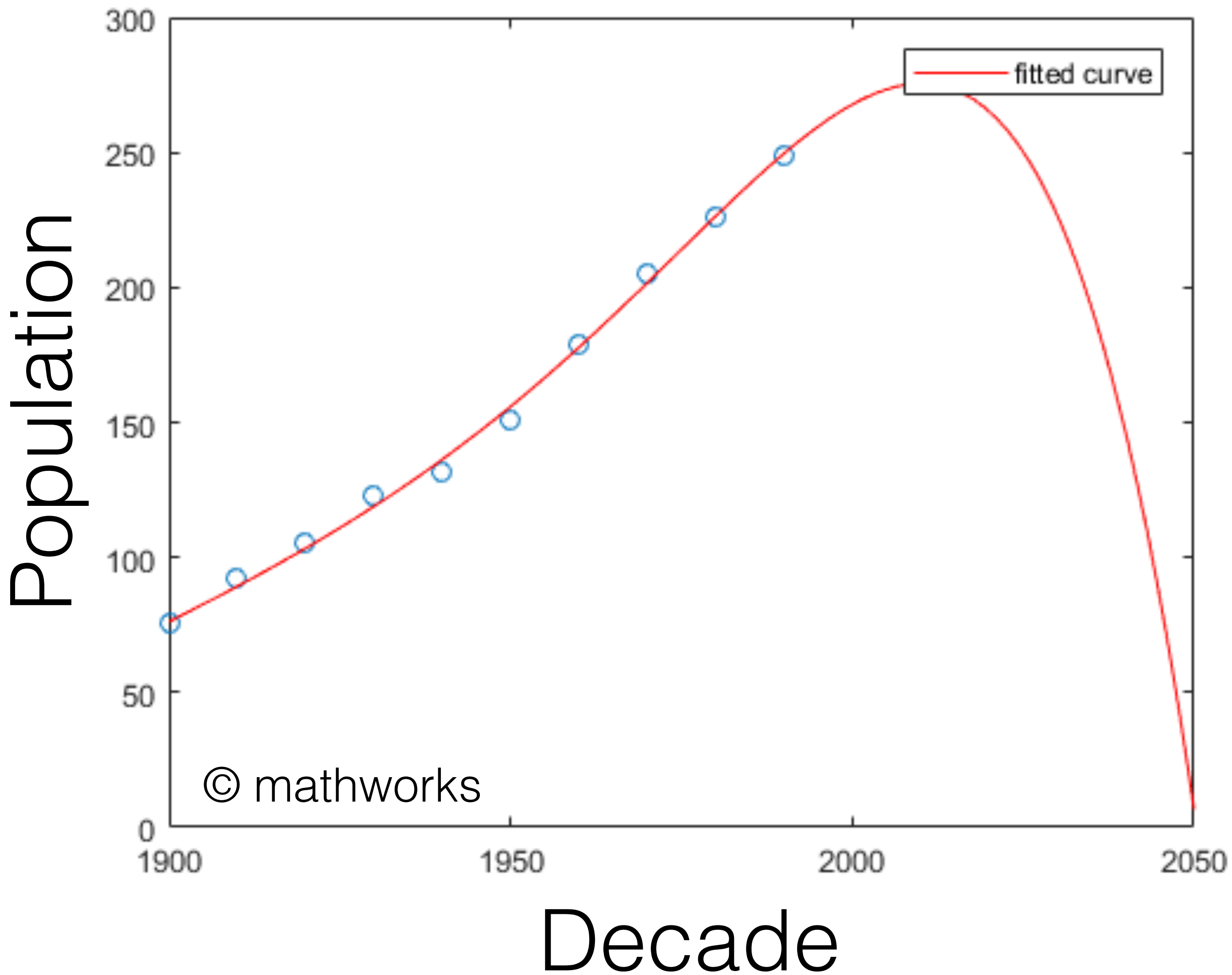
CONP PCNO

PARIETAL

Follow @raamana_

crossinvalidation.com

50 shades of overfitting



human annihilation?

Now, it's time to cross-validate!

Baycrest

Follow @raamana_

ONTARIO BRAIN INSTITUTE