

# **LOCO-MP: Built-in Uncertainty Quantification for LOCO Feature Importance**

---

Lili Zheng  
Department of Statistics, UIUC  
8/22/2025, EcoSta

Joint work with



Luqin Gan



Genevera I. Allen

# Table of contents

---

1. Motivation: Interpretable Machine Learning
2. Uncertainty Quantification for Feature Importance
  - Theoretical Guarantees
  - Simulations and Case Studies
3. Discussion and Conclusion

## **Motivation: Interpretable Machine Learning**

---

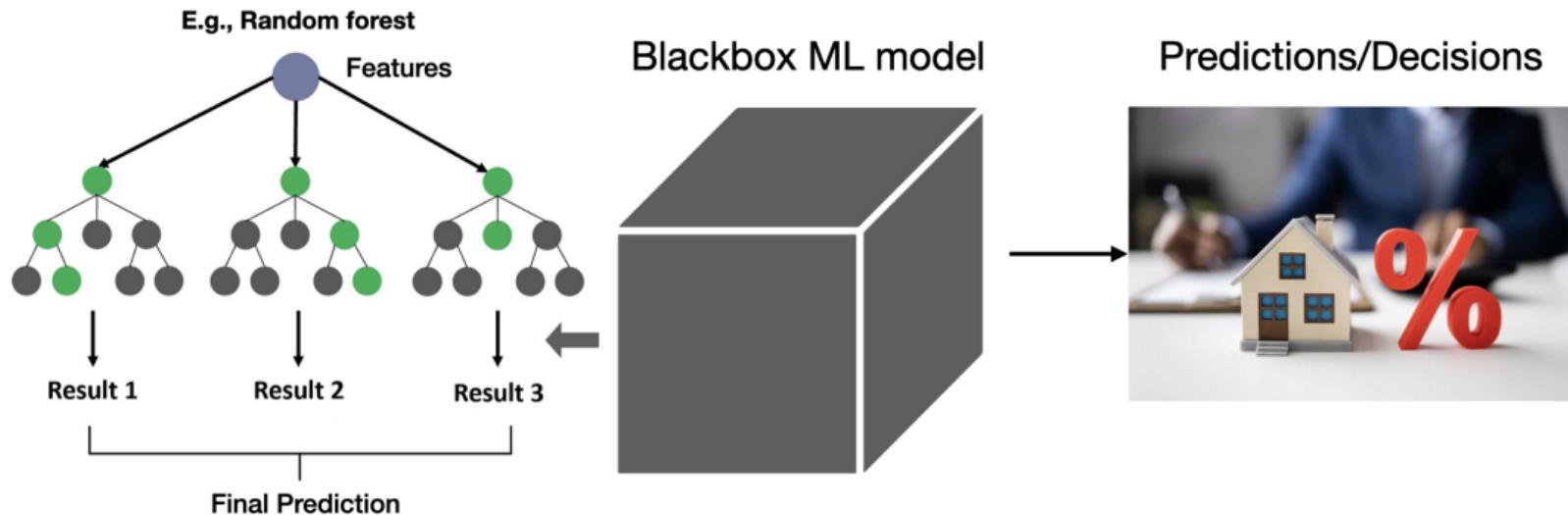
# Interpreting Black-box Machine Learning Models

Machine learning is widely applied in **high-stakes applications**:



Can we trust machine learning? Make it interpretable!

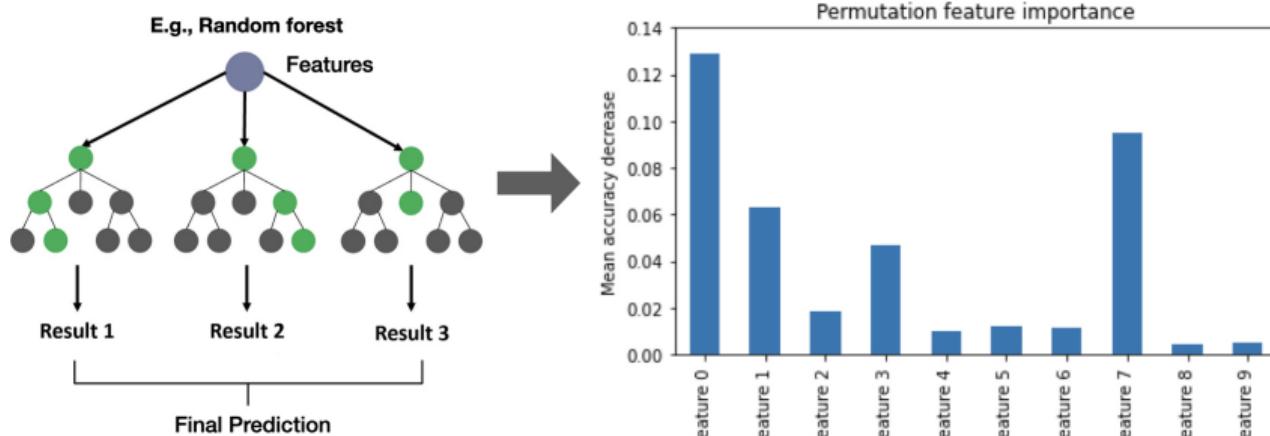
# Interpreting Black-box Machine Learning Models



What is this ML system's rationale of rejecting/approving mortgage applications?

# Feature Importance for Interpretable Machine Learning

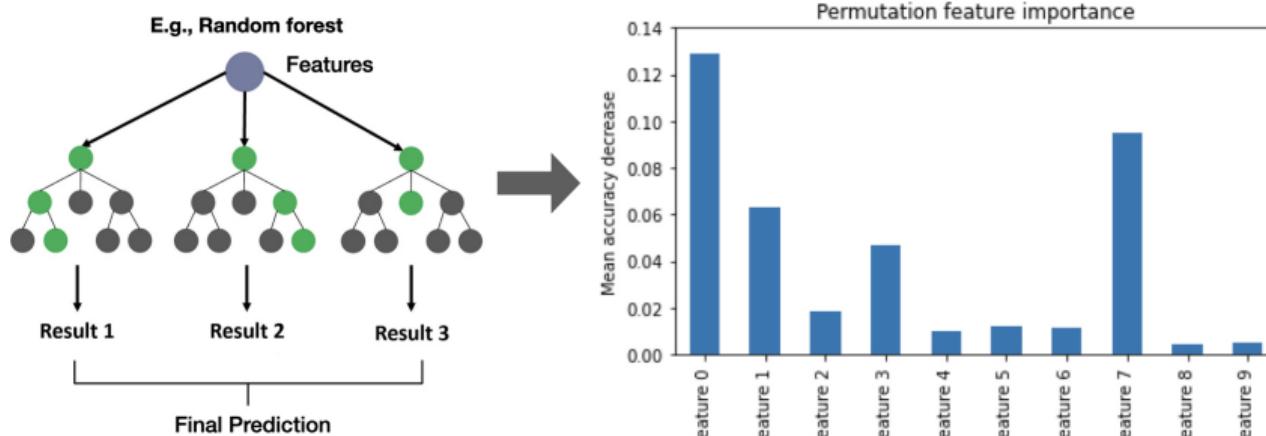
**Feature importance:** How does my model's prediction rely on each feature?



- Model-specific: defined for random forest, linear models, deep learning, etc.
- **Model-agnostic:** feature occlusion [Covert et al., 2021], permutation [König et al., 2021], Shapley values [Sundararajan and Najmi, 2020], etc.

# Feature Importance for Interpretable Machine Learning

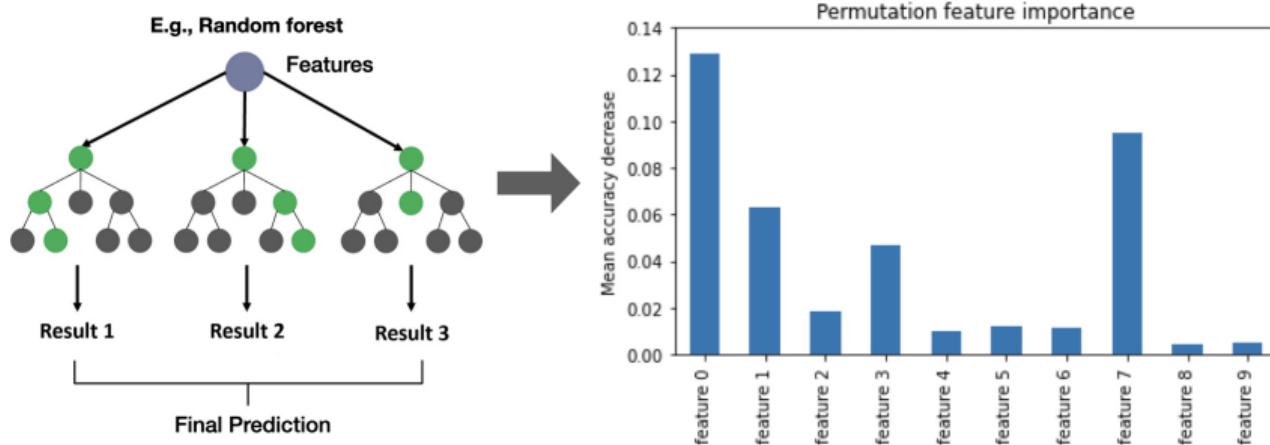
**Feature importance:** How does my model's prediction rely on each feature?



- We focus on **global** feature importance: explanation **for the whole population** instead of one instance.
- Answers “Which features often drives the decision for most applications?”

# Feature Importance for Interpretable Machine Learning

**Feature importance:** How does my model's prediction rely on each feature?



Many model-agnostic, global feature importance metrics exist.

Can we trust feature importance? Uncertainty quantification?

## Literature: Feature Importance Inference

---

- Inference for Lasso [Lee et al., 2016,  
Van de Geer et al., 2014]
- Conditional independence tests for  
random forest [Chi et al., 2022]
- Model-agnostic methods:  
Floodgate [Zhang and Janson, 2020],  
GCM [Shah and Peters, 2020], VIMP  
[Williamson et al., 2021]

## Literature: Feature Importance Inference

---

- Inference for Lasso [Lee et al., 2016, Van de Geer et al., 2014]
- Conditional independence tests for random forest [Chi et al., 2022]
- Model-agnostic methods:  
Floodgate [Zhang and Janson, 2020],  
GCM [Shah and Peters, 2020], VIMP  
[Williamson et al., 2021]

Wait a second...

- They all explain population data-generating model, not the ML model itself.
- ML models are only tools
- Impossible without strong assumptions about the data or model [Shah and Peters, 2020]!

## Literature: Feature Importance Inference

- Inference for Lasso [Lee et al., 2016, Van de Geer et al., 2014]
- Conditional independence tests for random forest [Chi et al., 2022]
- Model-agnostic methods:  
Floodgate [Zhang and Janson, 2020],  
GCM [Shah and Peters, 2020], VIMP  
[Williamson et al., 2021]

Wait a second...

- They all explain population data-generating model, not the ML model itself.
- ML models are only tools
- Impossible without strong assumptions about the data or model [Shah and Peters, 2020]!

We call this type **population feature importance**

# ML Feature Importance

---

The feature importance we want:

- Property of the **model**
- Which feature does my ML model rely on for decisions?
- Desired for **model diagnostics, auditing, and deployment**

# ML Feature Importance

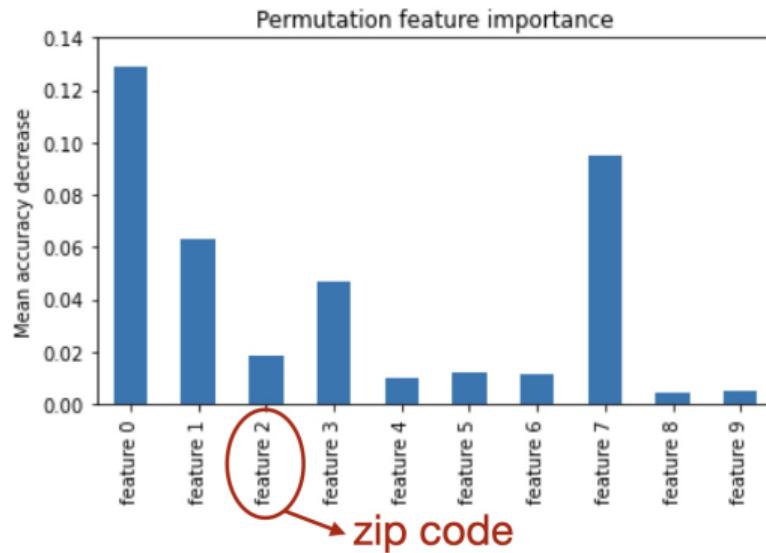
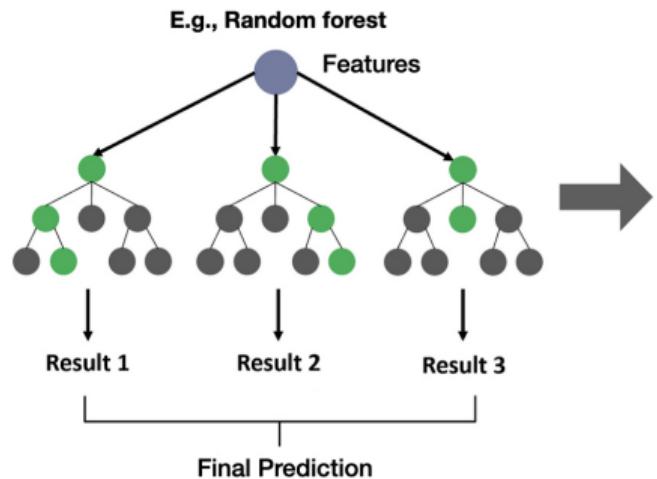
---

The feature importance we want:

- Property of the **model**
- Which feature does my ML model rely on for decisions?
- Desired for **model diagnostics, auditing, and deployment**

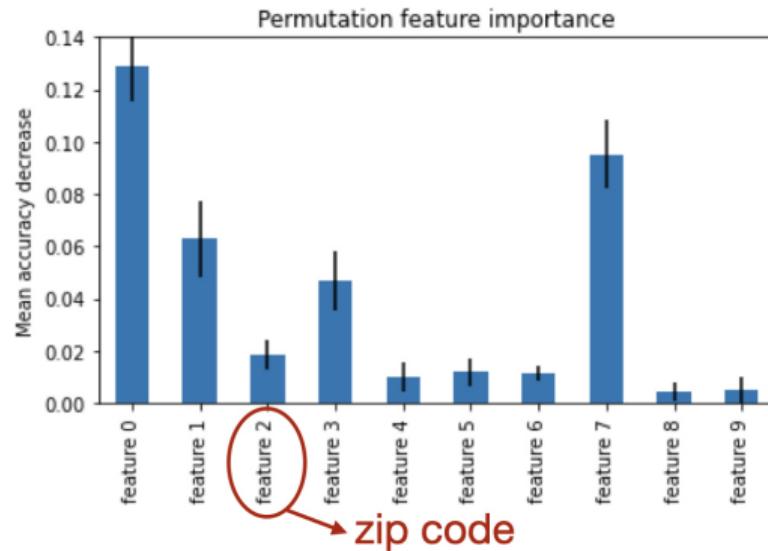
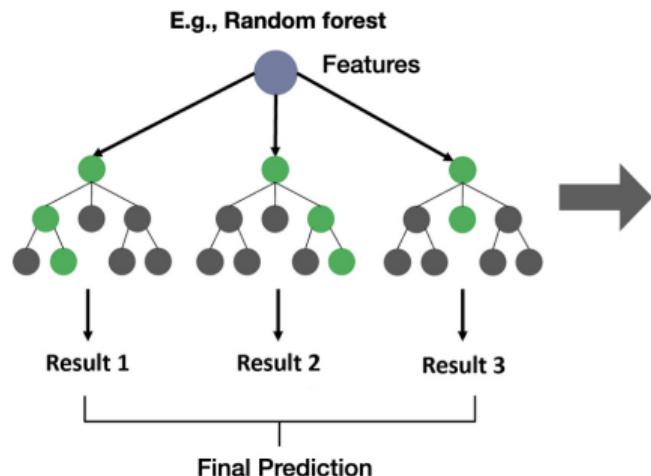
We call this type **ML feature importance**

# ML Feature Importance



- How much does mortgage decision rely on sensitive features?
- E.g., zip code is a proxy of race?
- Check ML feature importance

# ML Feature Importance



UQ for ML feature importance:

- has important societal consequences but is understudied!

## ML Feature Importance Inference

---

- Only a few works [Lei et al., 2018, Rinaldo et al., 2019, Watson and Wright, 2021]
- Some require knowing the distribution of features (model-X) [Watson and Wright, 2021]
- The most general, assumption-lean approach is the Leave-one-covariate-out (LOCO) inference [Lei et al., 2018, Rinaldo et al., 2019].
- Practical challenges: statistical & computational cost for inference is substantial.

# **Uncertainty Quantification for Feature Importance**

---

**Leave-One-Covariate-Out (LOCO) Inference**

[Lei et al., 2018, Rinaldo et al., 2019]:



### Leave-One-Covariate-Out (LOCO) Inference

[Lei et al., 2018, Rinaldo et al., 2019]:

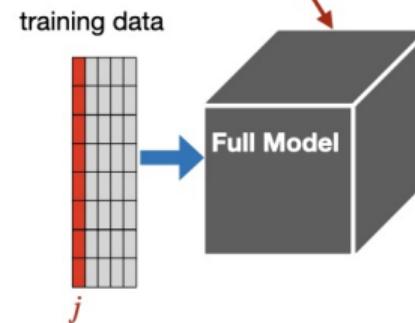


**Inference target:** Predictive power without feature  $j$  vs. with feature  $j$ .

## Prior Work: LOCO Inference

**Inference target:** Predictive power without feature  $j$  vs. with feature  $j$ .

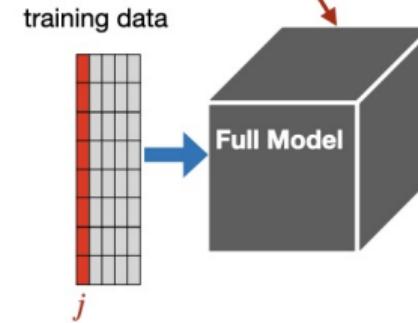
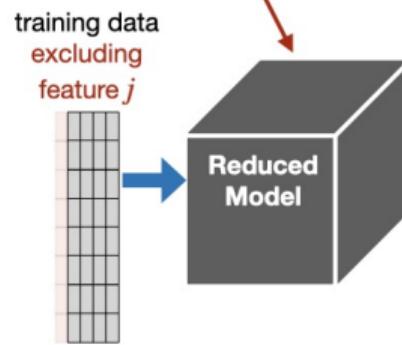
$$\Delta_j^*(\mathbf{X}, \mathbf{Y}) = \mathbb{E} [\text{Error} (Y^{\text{test}}, \mu_{-j}(X_{-j}^{\text{test}}; \mathbf{X}_{-j}, \mathbf{Y})) - \text{Error} (Y^{\text{test}}, \mu(X^{\text{test}}; \mathbf{X}, \mathbf{Y})) | \mathbf{X}, \mathbf{Y}]$$



## Prior Work: LOCO Inference

**Inference target:** Predictive power without feature  $j$  vs. with feature  $j$ .

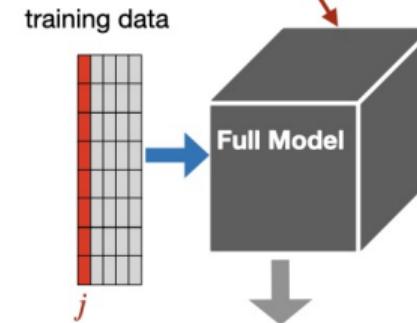
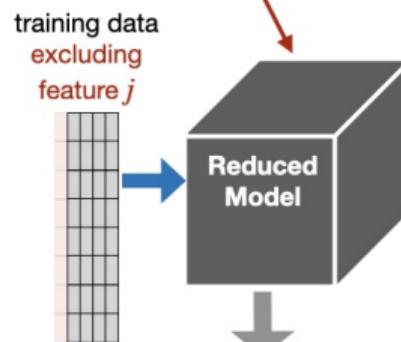
$$\Delta_j^*(\mathbf{X}, \mathbf{Y}) = \mathbb{E} [\text{Error} (Y^{\text{test}}, \mu_{-j}(X_{-j}^{\text{test}}; \mathbf{X}_{-j}, \mathbf{Y})) - \text{Error} (Y^{\text{test}}, \mu(X^{\text{test}}; \mathbf{X}, \mathbf{Y})) | \mathbf{X}, \mathbf{Y}]$$



## Prior Work: LOCO Inference

**Inference target:** Predictive power without feature  $j$  vs. with feature  $j$ .

$$\Delta_j^*(\mathbf{X}, \mathbf{Y}) = \mathbb{E} [\text{Error} (Y^{\text{test}}, \mu_{-j}(X_j^{\text{test}}; \mathbf{X}_{-j}, \mathbf{Y})) - \text{Error} (Y^{\text{test}}, \mu(X^{\text{test}}; \mathbf{X}, \mathbf{Y})) | \mathbf{X}, \mathbf{Y}]$$

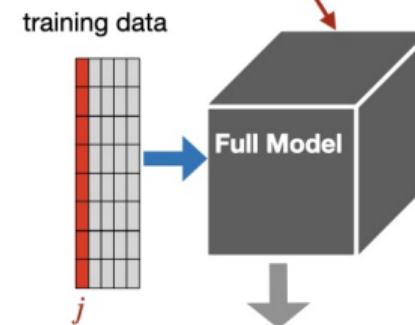
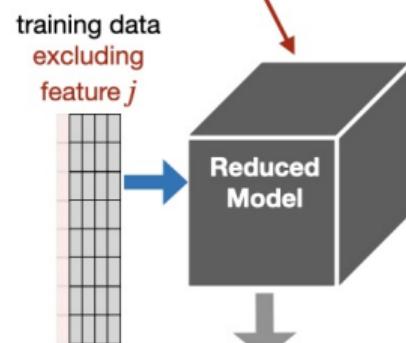


**Extra error due to feature occlusion:  $\Delta_j^*(X, Y)$**

## Prior Work: LOCO Inference

**Inference target:** Predictive power without feature  $j$  vs. with feature  $j$ .

$$\Delta_j^*(\mathbf{X}, \mathbf{Y}) = \mathbb{E} [\text{Error} (Y^{\text{test}}, \mu_{-j}(X_{-j}^{\text{test}}; \mathbf{X}_{-j}, \mathbf{Y})) - \text{Error} (Y^{\text{test}}, \mu(X^{\text{test}}; \mathbf{X}, \mathbf{Y})) | \mathbf{X}, \mathbf{Y}]$$



**Extra error due to feature occlusion:  $\Delta_j^*(X, Y)$**

How much does feature  $j$  **help or hurt** model  $\mu$ 's predictive performance?

### LOCO inference approach:

- Splits data; fits full and reduced models to training data
- Feature occlusion scores on test data  $\Rightarrow$  confidence intervals

## Prior Work: LOCO Inference

### LOCO inference approach:

- Splits data; fits full and reduced models to training data
- Feature occlusion scores on test data  $\Rightarrow$  confidence intervals

#### Advantages

- Model-agnostic (applicability).
- Statistically valid without assuming data distribution/model choice.

## LOCO inference approach:

- Splits data; fits full and reduced models to training data
- Feature occlusion scores on test data  $\Rightarrow$  confidence intervals

### Advantages

- Model-agnostic (applicability).
- Statistically valid without assuming data distribution/model choice.

### Challenges

- Data splitting loses data efficiency for both training and inference;
- Interpretation is not for the full model & depends on random data splitting
- Model refitting for each feature: prohibitive computation after model training

### LOCO inference approach:

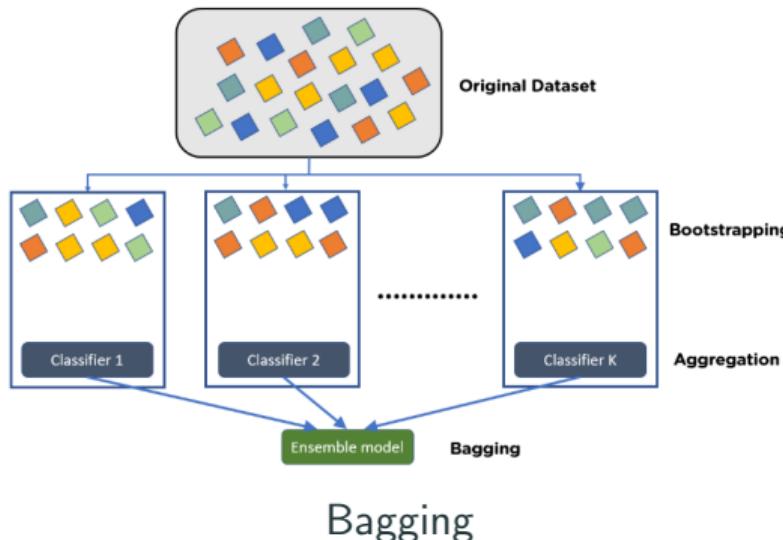
- Splits data; fits full and reduced models to training data
- Feature occlusion scores on test data  $\Rightarrow$  confidence intervals

#### Our Goal

Can we utilize the general LOCO framework to perform ML feature importance inference, while [avoiding data splitting and model refitting](#)?

# **Our Approach: LOCO Inference for an Ensemble Framework**

# LOCO Inference for Ensemble Learning



Picture source: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/bagging-in-machine-learning>

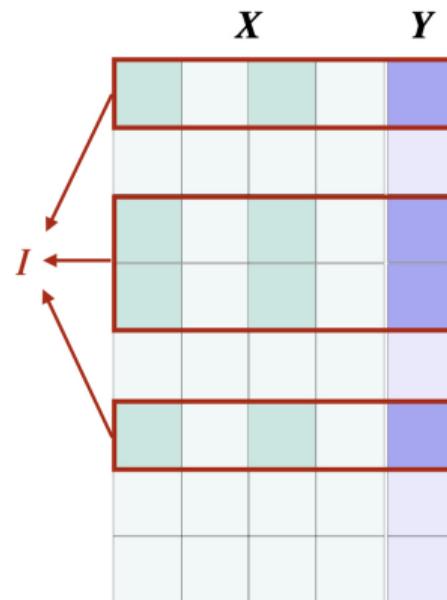
**Inspiration:** Jackknife+ After Bootstrap [Kim et al., 2020].

- Many ensemble methods are good predictors
- Conformal inference (Jackknife+) for bagging is **computationally free with no data-splitting!**

**Idea: Minipatch Ensembles.**

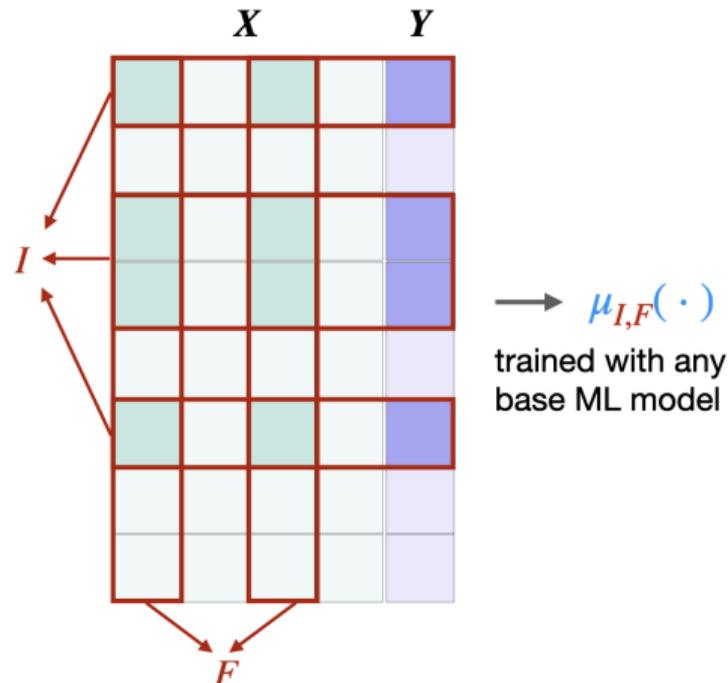
# Minipatch Ensemble Learning

**Minipatch ensembles:** like bagging, but double-subsampling for both observations and features [Yao and Allen, 2020].



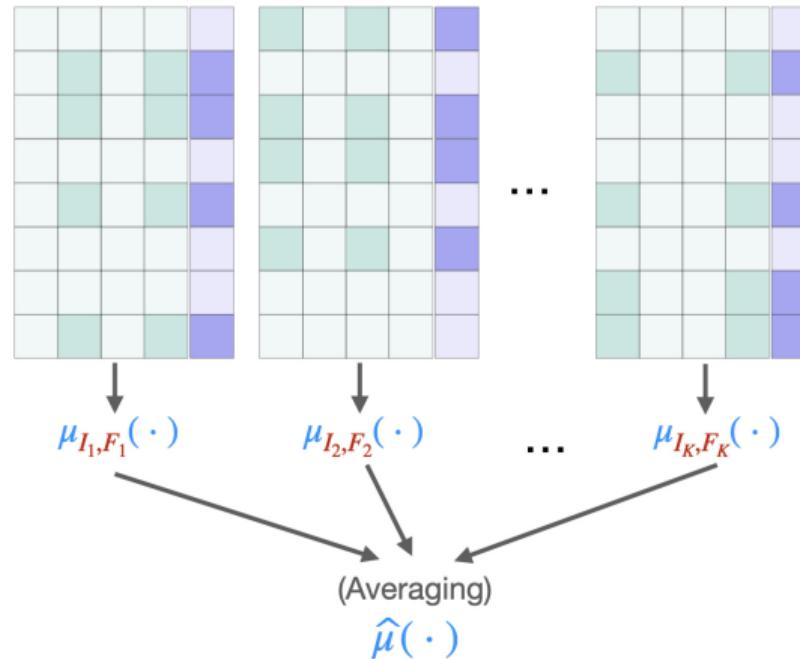
# Minipatch Ensemble Learning

**Minipatch ensembles:** like bagging, but double-subsampling for both observations and features [Yao and Allen, 2020].



# Minipatch Ensemble Learning

**Minipatch ensembles:** like bagging, but double-subsampling for both observations and features [Yao and Allen, 2020].

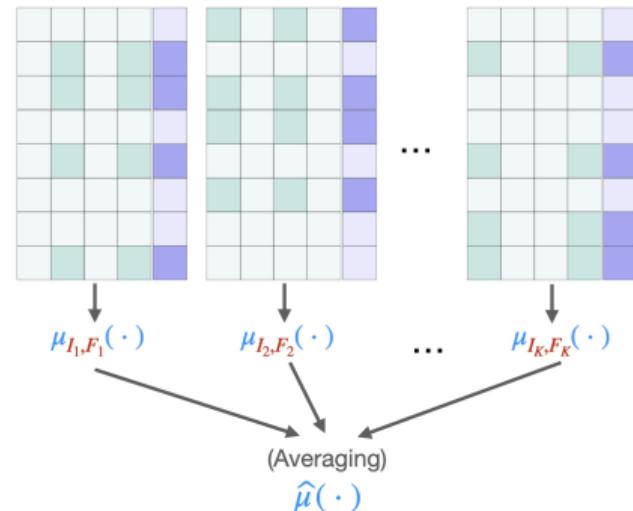


# Minipatch Ensemble Learning

**Inspiration:** Bagging; Random Forests [Louppe and Geurts, 2012]; Stochastic Optimization & Dropout.

## Advantages:

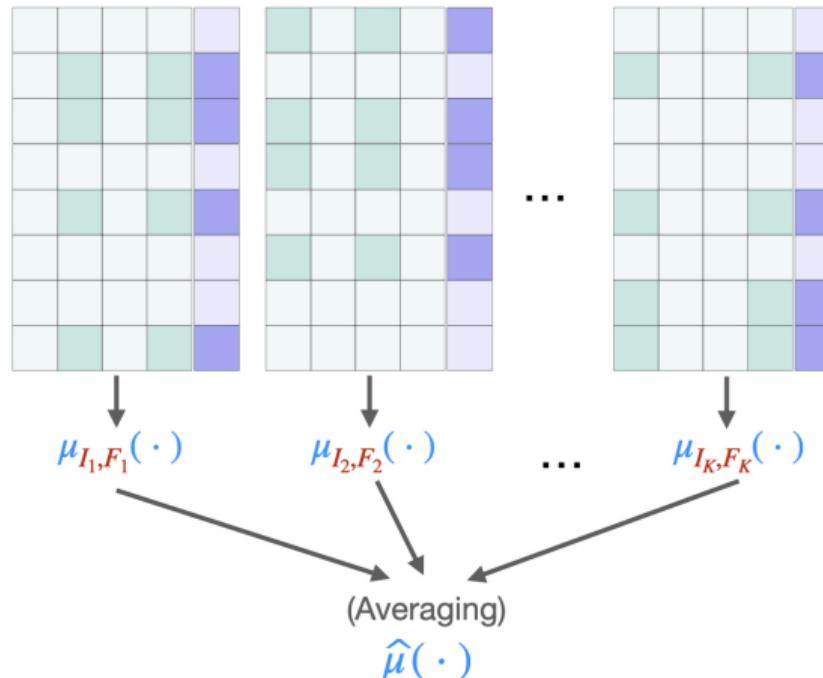
- Fast and easily parallelizable
- Ensemble diversity; **implicit regularization** [LeJeune et al., 2020, Yao et al., 2021]



**LOCO Inference for Minipatch Ensembles?**

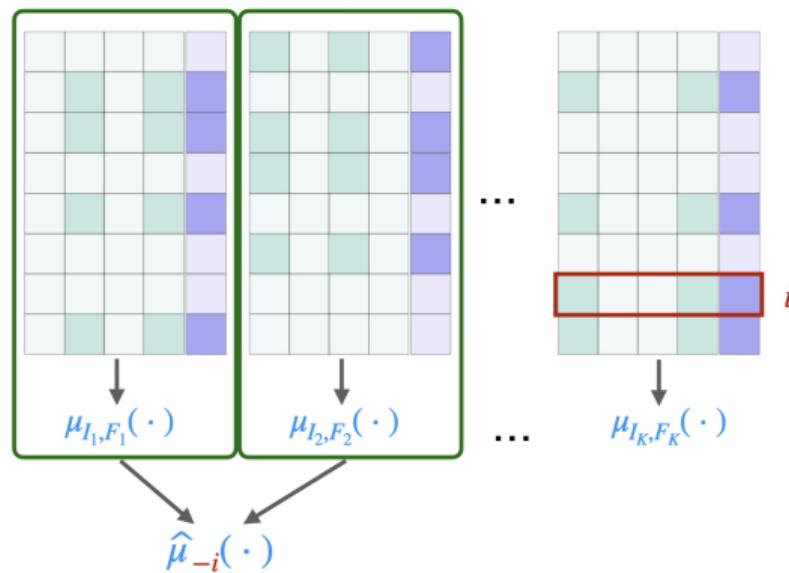
## Algorithm: LOCO for Minipatch

- Step 1. Fit minipatch learning predictor:  $\hat{\mu}$ .



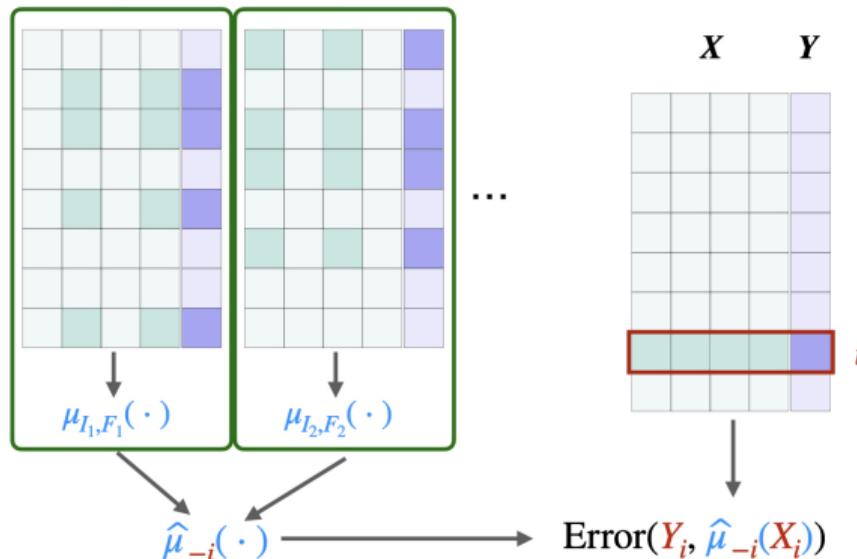
## Algorithm: LOCO for Minipatch

- **Step 2. LOO** (leave-one-observation-out) predictor:  $\hat{\mu}_{-i}(X_i)$ .
  - Ensemble minipatches **without observation  $i$** .
  - Compute test error on sample  $i$ .



## Algorithm: LOCO for Minipatch

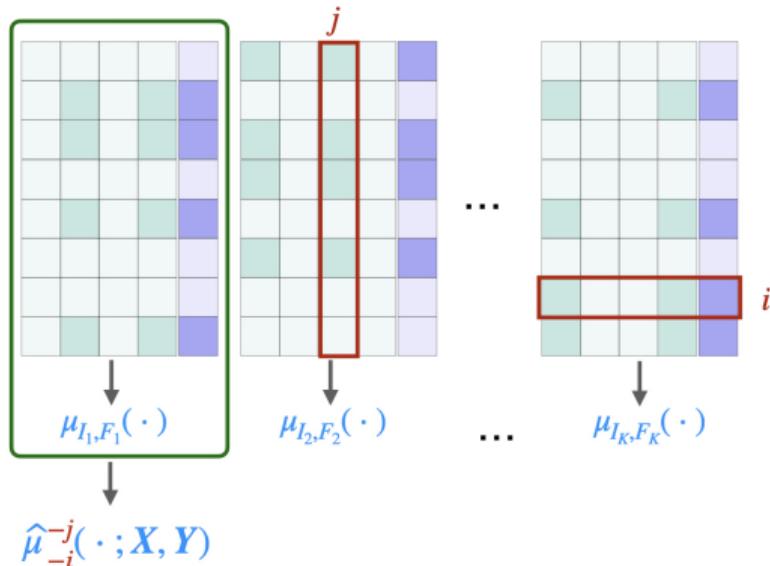
- Step 2. LOO (leave-one-observation-out) predictor:  $\hat{\mu}_{-i}(X_i)$ .
  - Ensemble minipatches **without observation  $i$** .
  - Compute test error on sample  $i$ .



No data-splitting!  
Simple model averaging;  
Free computationally!

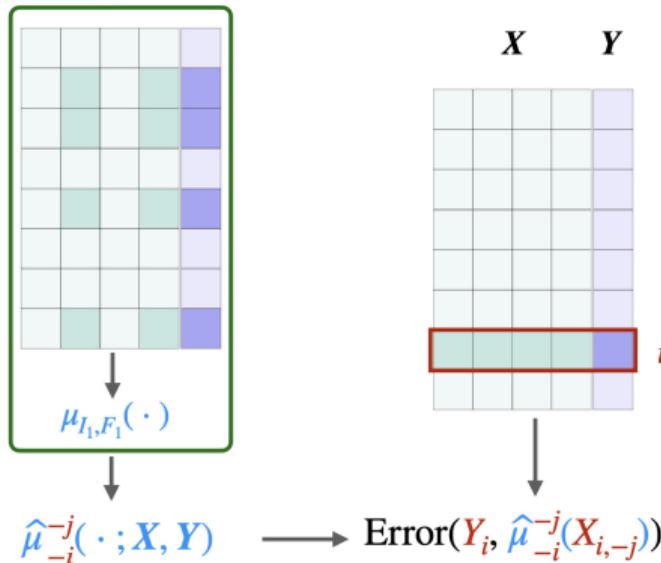
## Algorithm: LOCO for Minipatch

- **Step 3. LOCO-LOO** predictor:  $\hat{\mu}_{-i}^{-j}(X_i)$ .
  - Ensemble minipatches **without observation  $i$  and without feature  $j$ .**
  - Compute test error on sample  $i$ .



## Algorithm: LOCO for Minipatch

- **Step 3. LOCO-LOO** predictor:  $\hat{\mu}_{-i}^{-j}(X_i)$ .
  - Ensemble minipatches **without observation  $i$  and without feature  $j$ .**
  - Compute test error on sample  $i$ .



Simple model averaging;  
Free computationally!

## Algorithm: LOCO for Minipatch

---

- **Step 4.** Compute feature occlusion scores for observations  $1 \leq i \leq N$ :

$$\hat{\Delta}_j(X_i, Y_i) = \text{Error}(\textcolor{red}{Y_i}, \hat{\mu}_{-i}^{-j}(\textcolor{red}{X_i})) - \text{Error}(\textcolor{red}{Y_i}, \hat{\mu}_{-i}(\textcolor{red}{X_i})).$$

**Importance of feature  $j$  for predicting sample  $i$ .**

- **Step 5.** Construct asymptotically normal interval from  $\{\hat{\Delta}_j(X_i, Y_i)\}_{i=1}^N$ :

$$\hat{\mathbb{C}}_j = \left[ \bar{\Delta}_j - \frac{z_{\alpha/2} \hat{\sigma}_j}{\sqrt{N}}, \bar{\Delta}_j + \frac{z_{\alpha/2} \hat{\sigma}_j}{\sqrt{N}} \right],$$

$\bar{\Delta}_j$ : mean occlusion score,  $\hat{\sigma}_j$ : standard deviation.

## Algorithm: LOCO for Minipatch

---

### Full Algorithm

- **Step 1.** Fit minipatch learning predictor.
- **Step 2&3.** For each sample  $i$ , compute **LOO** and **LOCO-LOO** predictor by simple model averaging.
- **Step 4&5.** Construct the normal confidence interval.

## Algorithm: LOCO for Minipatch

### Full Algorithm

- **Step 1.** Fit minipatch learning predictor.
- **Step 2&3.** For each sample  $i$ , compute **LOO** and **LOCO-LOO** predictor by simple model averaging.
- **Step 4&5.** Construct the normal confidence interval.

#### Algorithmic advantages

- No **data-splitting**  $\Rightarrow$  all available data is used for **training** a good predictive model and for **powerful inference**.
- No **model-refitting**  $\Rightarrow$  once predictive model is trained, confidence intervals are **computationally free!**

## Algorithm: LOCO for Minipatch

### Algorithmic advantages

- No **data-splitting** ⇒ all available data is used for **training** a good predictive model and for **powerful inference**.
- No **model-refitting** ⇒ once predictive model is trained, confidence intervals are **computationally free!**

- An ensemble framework with **built-in** LOCO inference
- Once a minipatch ensemble is trained, **almost no extra cost** (data or computation) needed to get its own LOCO confidence intervals!
- As a comparison: *LOCO-Split require extra data set-aside for inference only, and model refitting for each feature of interest.*

# **Uncertainty Quantification for Feature Importance**

---

**Theoretical Guarantees**

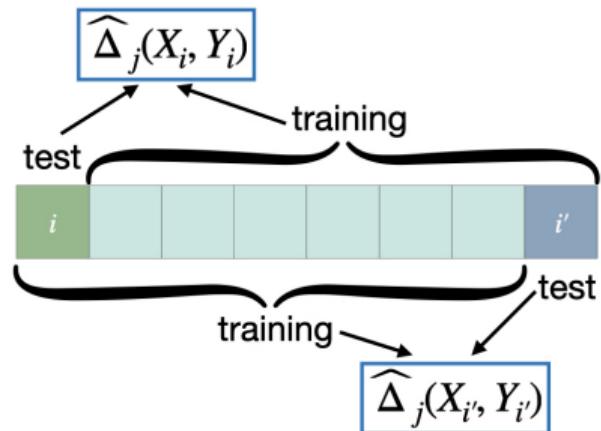
Does LOCO-MP confidence interval have valid coverage?

- Recall: our target  $\Delta_j^*(\mathbf{X}, \mathbf{Y})$  is a **function of the models**  $\mu_{-j}(\cdot)$ ,  $\mu(\cdot)$ , **trained by data**  $(\mathbf{X}, \mathbf{Y})$ .
- A selective inference problem. Common approaches: data-splitting; characterize statistic distribution conditioning on training (requires assumptions on data and model).
- Our approach: [leave-one-observation-out](#), independent training and testing for each  $\hat{\Delta}_j(X_i, Y_i)$ .

# Theoretical Guarantees

Does LOCO-MP confidence interval have valid coverage?

- However, dependency still exists amongst  $\{\widehat{\Delta}_j(X_i, Y_i)\}_{i=1}^N$ !
- $\widehat{\Delta}_j(X_i, Y_i)$  and  $\widehat{\Delta}_j(X_{i'}, Y_{i'})$  switches  $i$  and  $i'$  for training and testing; **share  $N - 2$  training samples.**



## Theoretical Guarantees

---

- A1.  $\text{Error}(Y, \hat{Y})$  is Lipschitz w.r.t. prediction  $\hat{Y}$ .
- A2. Average minipatch prediction is bounded (automatically hold for classification).
- A3. Small MP:  $n = o(\sqrt{N})$ ; If each base model is stable, this assumption can be further relaxed.
- A4. Large number of MPs:  $K \gg (N \log N) \vee (\frac{M}{m} \log M)$

## Theoretical Guarantees

- A1. Error( $Y, \hat{Y}$ ) is Lipschitz w.r.t. prediction  $\hat{Y}$ .
- A2. Average minipatch prediction is bounded (automatically hold for classification).
- A3. Small MP:  $n = o(\sqrt{N})$ ; If each base model is stable, this assumption can be further relaxed.
- A4. Large number of MPs:  $K \gg (N \log N) \vee (\frac{M}{m} \log M)$

### Theorem

Suppose samples  $(\mathbf{X}_i, Y_i)$  are i.i.d., and assumptions A1-A4 hold. Then

$$\lim_{N \rightarrow \infty} \mathbb{P}(\Delta_j^* \in \hat{\mathbb{C}}_j) = 1 - \alpha.$$

**Valid asymptotic coverage** under mild assumptions; applicable to any data distributions and base ML models.

## Theoretical Guarantees

---

- **Algorithmic stability:** prediction is stable against change in one training sample.
- Stability facilitates [statistical inference under dependency](#) [Bayle et al., 2020]!

## Theoretical Guarantees

---

- **Algorithmic stability:** prediction is stable against change in one training sample.
- Stability facilitates **statistical inference under dependency** [Bayle et al., 2020]!
- Minipatch ensembles are **stable with any base model and any data distribution!**
- Independent interest: stability also helps with conformal inference [Liang and Barber, 2023].

## Extension: Beyond Tiny Minipatches

---

For better predictiveness, MP size should be carefully chosen.

- Relax minipatch size assumption:  $n = o(\sqrt{N}) \rightarrow n \ll N/\log N$
- Data-driven tuning for minipatch size: train minipatch ensembles with a list of MP sizes; [minimize the LOO error \(easily computed\)](#).

## Extension: Beyond Tiny Minipatches

---

For better predictiveness, MP size should be carefully chosen.

- Relax minipatch size assumption:  $n = o(\sqrt{N}) \rightarrow n \ll N/\log N$
- Data-driven tuning for minipatch size: train minipatch ensembles with a list of MP sizes; [minimize the LOO error \(easily computed\)](#).

Inference still valid! As long as we [add a buffer to the confidence interval width](#):

$$\max \left\{ \frac{\hat{\sigma}_j}{\sqrt{N}}, c \sqrt{\widehat{\text{stb}}} \frac{n \log N}{N} \right\},$$

$\widehat{\text{stb}}$ : an estimate of the base model stability using trained minipatches.

# **Uncertainty Quantification for Feature Importance**

---

**Simulations and Case Studies**

## Simulation Framework

---

- Data generation process:
  - Sparse **linear** model
  - Sparse additive, **non-linear** model
- $M = 50, N = 100, 200, \dots, 2000.$

## Simulation Framework

---

- Data generation process:
  - Sparse **linear** model
  - Sparse additive, **non-linear** model
- $M = 50, N = 100, 200, \dots, 2000.$
- Base model: ridge regression, decision tree, kernel ridge regression.
- $m = 0.5M, n = N^{0.8}, K = 10,000.$

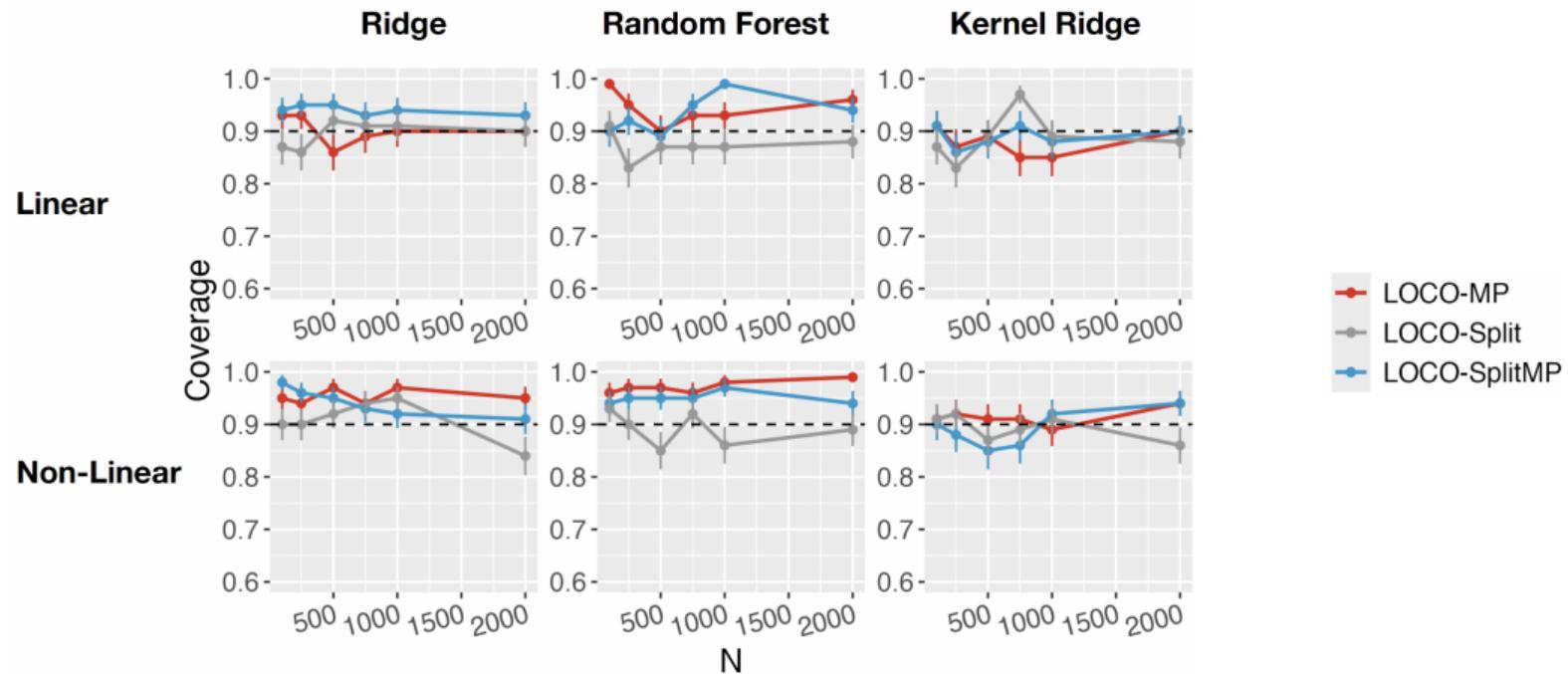
## Simulation Framework

---

- Data generation process:
  - Sparse linear model
  - Sparse additive, non-linear model
- $M = 50, N = 100, 200, \dots, 2000.$
- Base model: ridge regression, decision tree, kernel ridge regression.
- $m = 0.5M, n = N^{0.8}, K = 10,000.$
- Compare with the original LOCO method via data-splitting, where the trained models are
  - ridge, random forest, kernel ridge (**LOCO-Split**)
  - minipatch ensembles of these base models (**LOCO-SplitMP**)

# Simulations: Validation of Theory

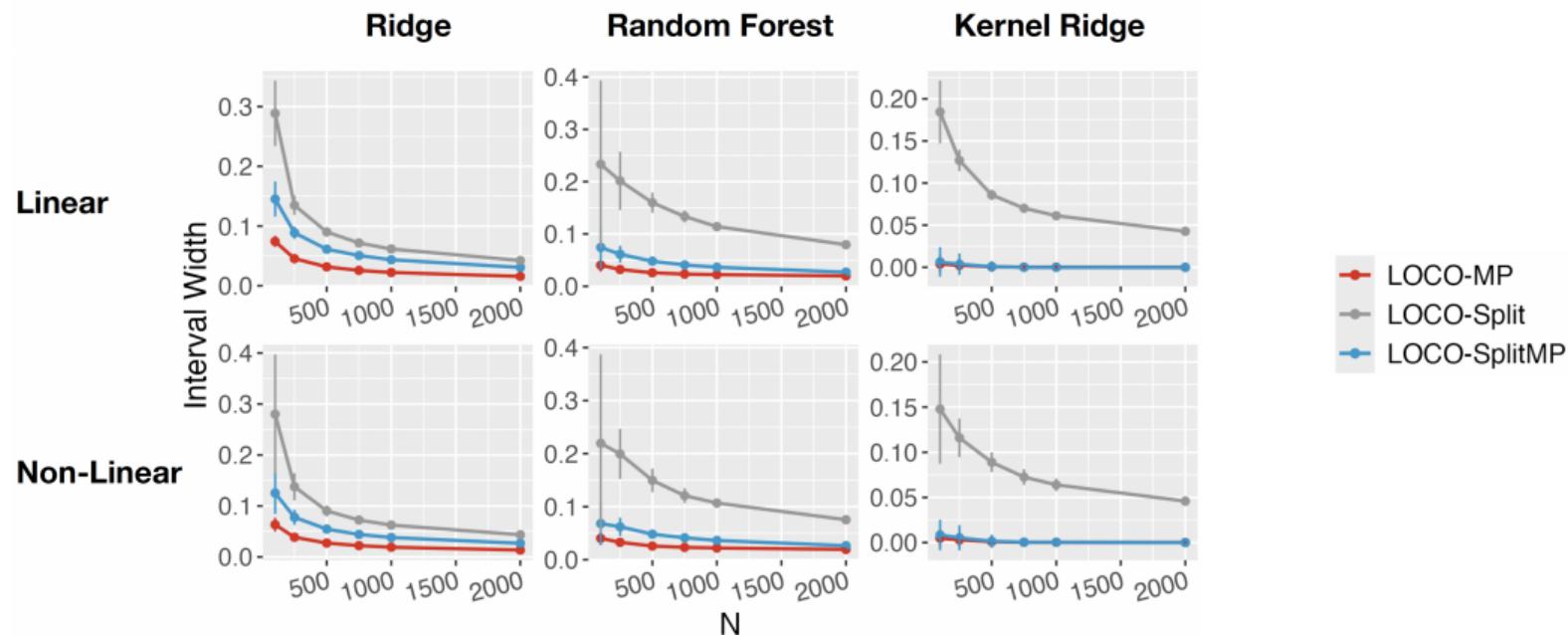
Regression, inference for a signal feature.



Asymptotically valid coverage.

# Simulations: Validation of Theory

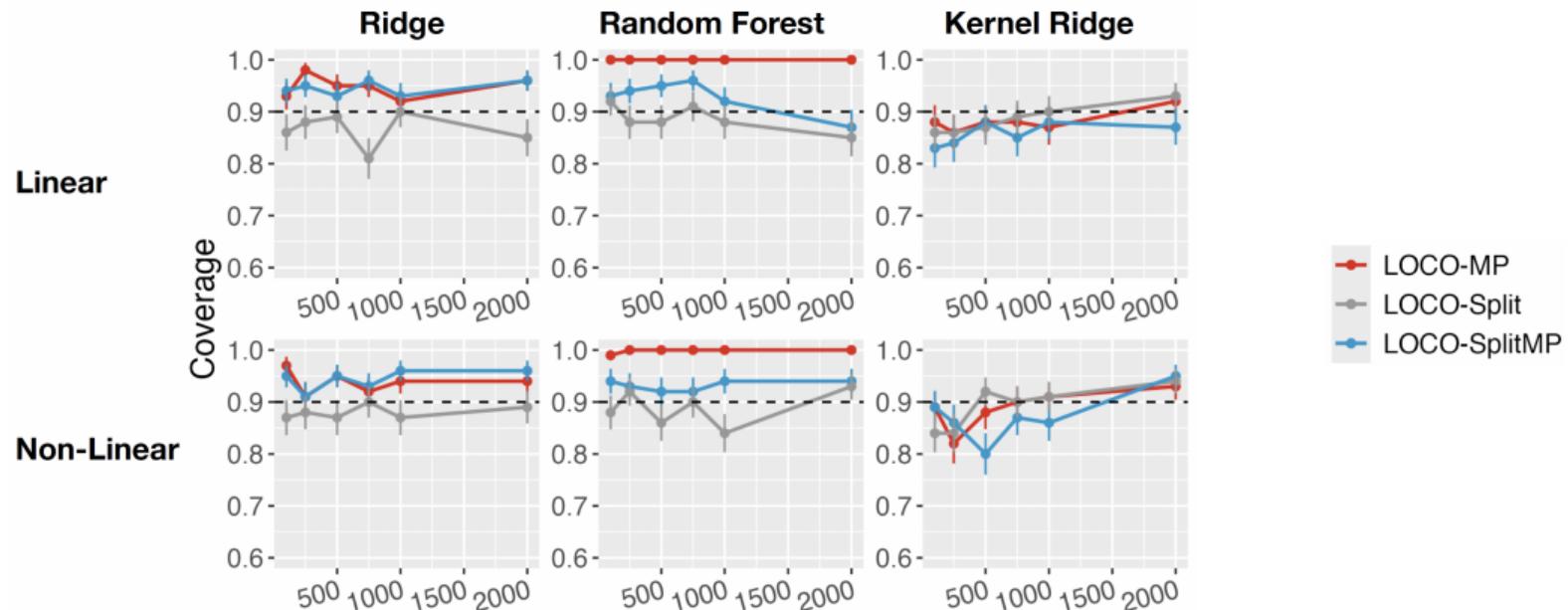
Regression, inference for a signal feature.



Width decreases as  $N$  increases; LOCO-MP has the smallest interval width.

# Simulations: Validation of Theory

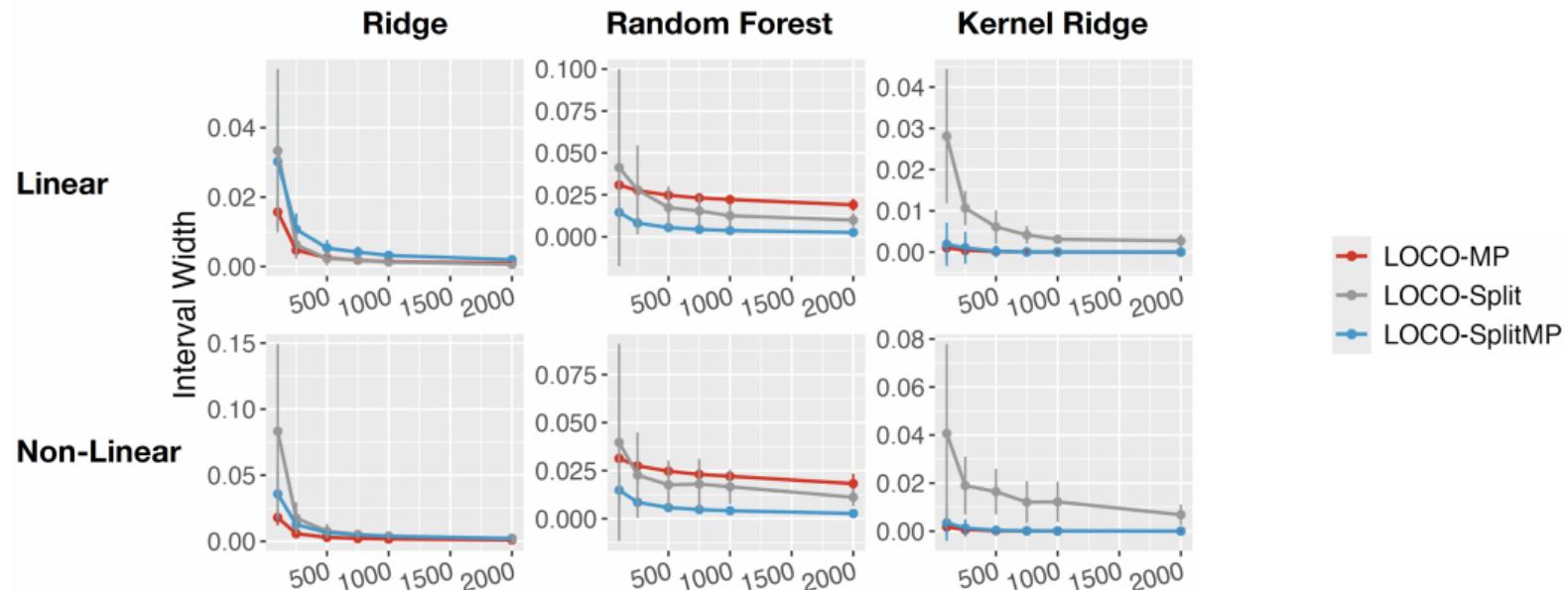
Regression, inference for a noise feature.



Slight over-coverage for random forest due to variance barrier.

# Simulations: Validation of Theory

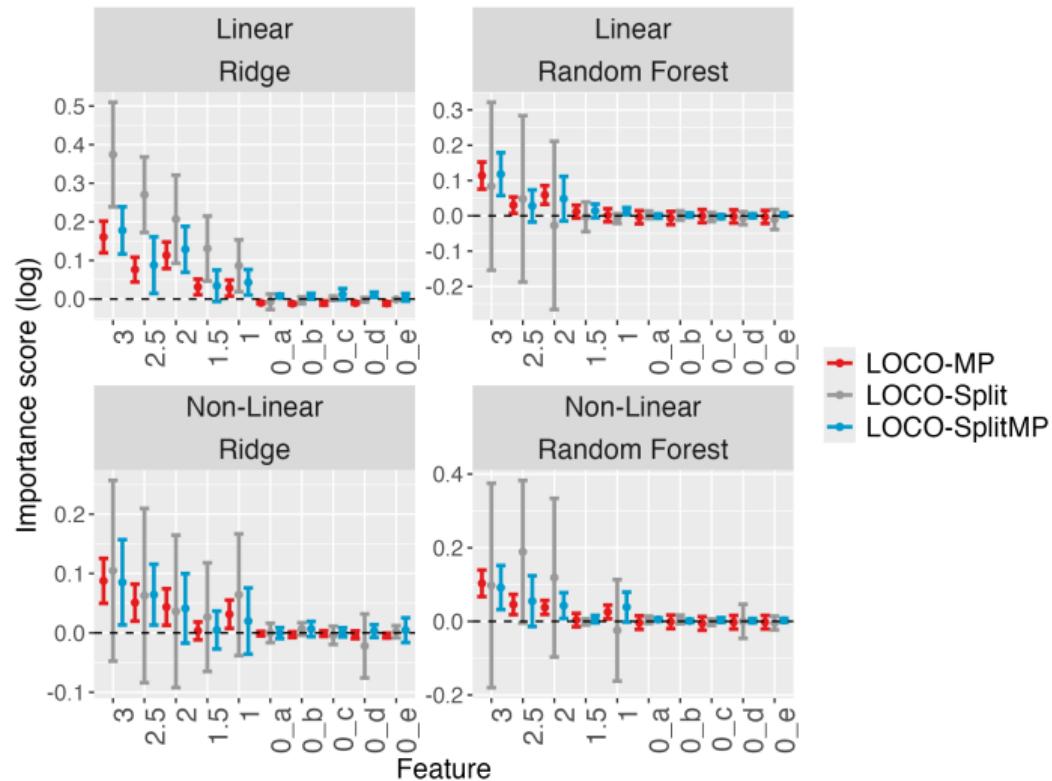
Regression, inference for a noise feature.



Slight over-coverage for random forest due to variance barrier.

# Simulation: Comparative Study

- $N = 200$
- Confidence intervals for features 1-10
- LOCO-Split fails to identify most signal features
- LOCO-MP has shorter intervals than LOCO-SplitMP

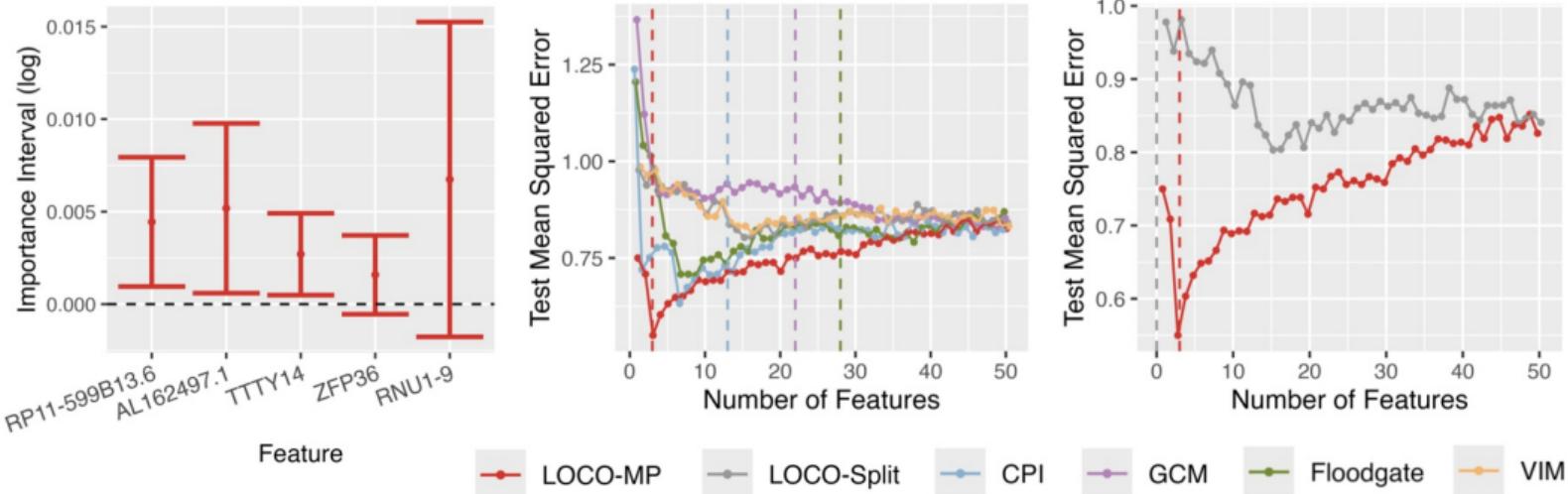


## Real Data Example

---

- Religious Orders Study Memory and Aging Project (ROSMAP) data [Bennett et al., 2018]
- Response: cognition scores of 507 patients
- Features: 86 biomarkers with variance  $\geq 0.5$
- Compare with LOCO-Split, CPI [Watson and Wright, 2021], GCM [Shah and Peters, 2020], Floodgate [Zhang and Janson, 2020], VIM [Williamson et al., 2021]

## Real Data Example



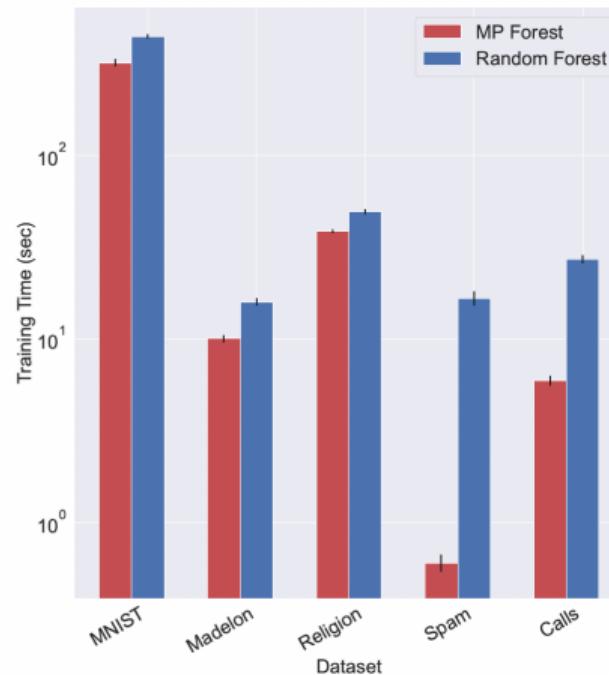
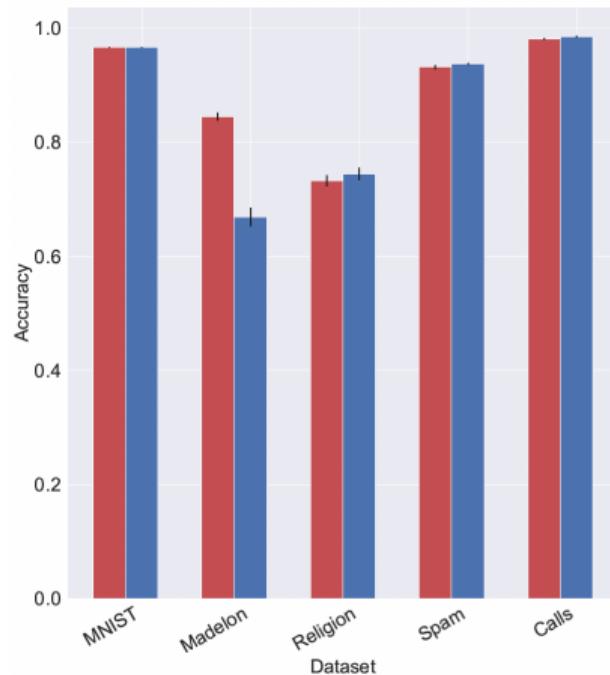
- LOCO-MP identify relevant biomarkers (supported by biomedical literature)
- Fit random forest with top  $k$  features and check prediction error on test set  
**(LOCO-MP selected the most informative features)**

## **Discussion and Conclusion**

---

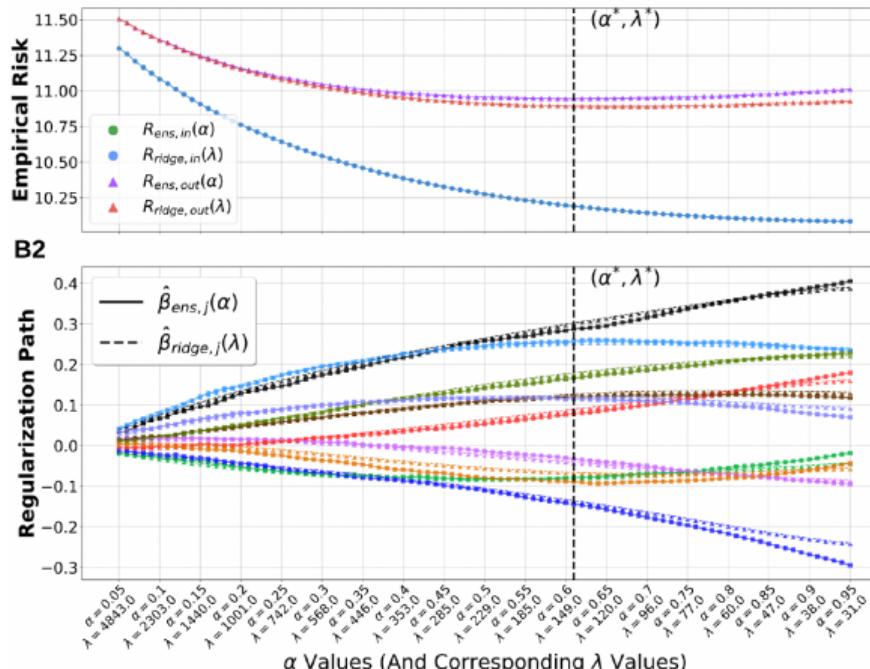
# What is the Minipatch Learning Predictor?

When base models are trees, Minipatch predictor is similar to random forest



# What is the Minipatch Learning Predictor?

When base models are linear regression, Minipatch predictor is equivalent to ridge regression [LeJeune et al., 2020, Yao et al., 2021]



## Minipatch Feature Importance vs. Population Feature Importance?

---

**Special Case: Linear Model.** For independent features,

- $\Delta_j^*$  concentrates around  $\tilde{\Delta}_j^*$ :  $\tilde{\Delta}_j^* \asymp 2\gamma \left( \beta_j^{*2} - \frac{\|\beta_{\setminus j}^*\|_2^2}{M-1} \right)$  (with  $\gamma = m/M$ ).
- Under assumptions on the minipatch size and number; valid coverage for  $\tilde{\Delta}_j^*$ .

## Minipatch Feature Importance vs. Population Feature Importance?

---

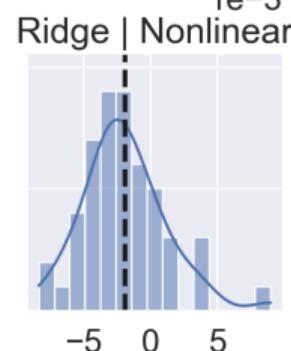
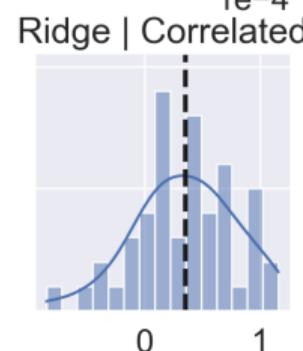
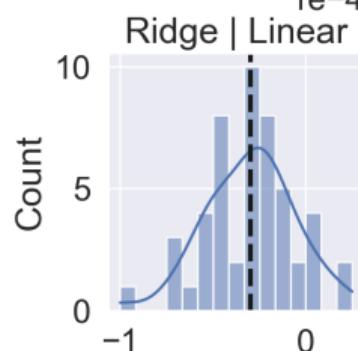
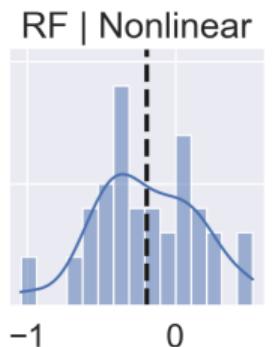
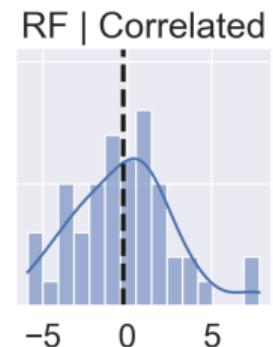
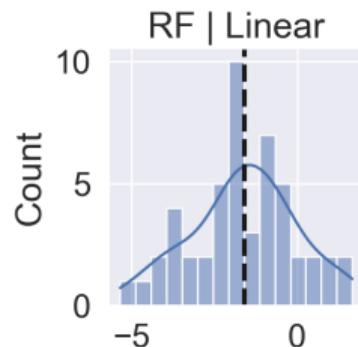
**Special Case: Linear Model.** For correlated features:

- When  $x_1$  and  $x_2$  have correlation  $\rho \rightarrow 1$ , we prove that  $\Delta_1^* \rightarrow \Delta_2^*$  and are a function of  $(\beta_1^* + \beta_2^*)^2$  for LOCO-MP.

As a comparison: original LOCO inference tends to miss correlated features.

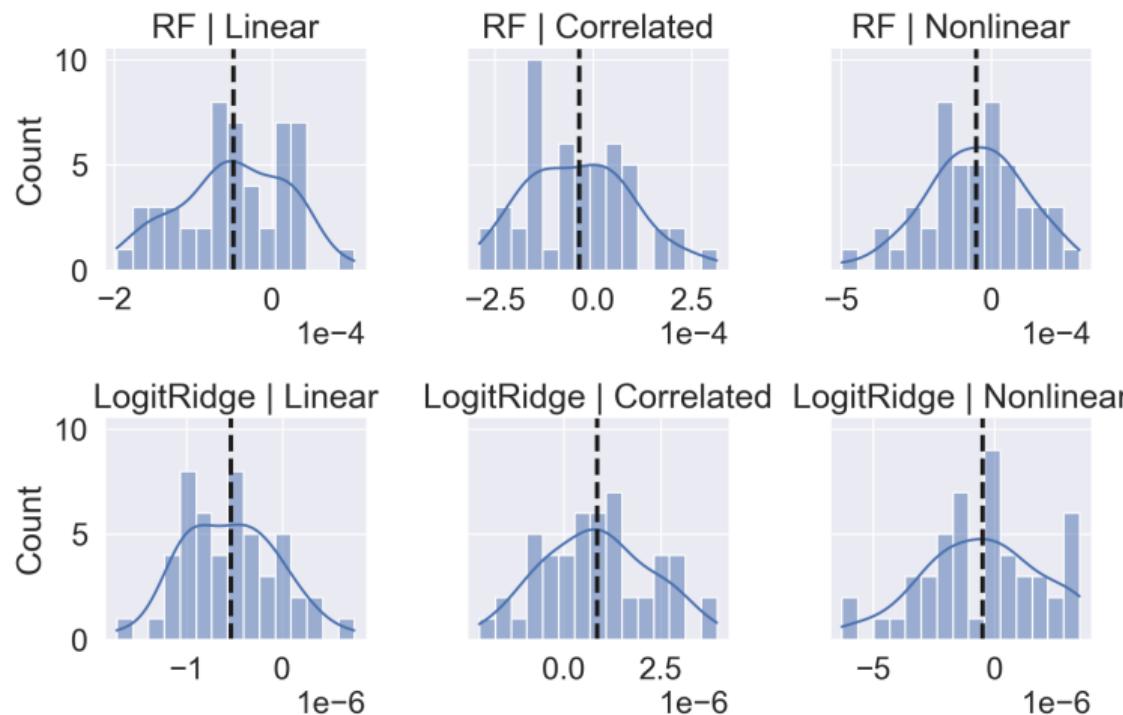
# Minipatch Feature Importance vs. Population Feature Importance?

Histograms of the inference target for a noise feature in the regression setting



# Minipatch Feature Importance vs. Population Feature Importance?

Histograms of the inference target for a noise feature in the classification setting



## Extension to Hypothesis Testing

---

Recall: our inference target  $\Delta_j^*(\mathbf{X}, \mathbf{Y})$  is a function of the data instead of a population quantity.

- $\mathcal{H}_0 : \Delta_j^*(\mathbf{X}, \mathbf{Y}) \leq 0$  is random event
- Directly inverting our confidence interval to hypothesis test guarantees:

$$\mathbb{P}(\text{We reject } \mathcal{H}_0 \text{ & } \mathcal{H}_0 \text{ is true}) \leq \alpha.$$

\* Different from the conditional Type I error control in selective inference literature.

## Conclusion

---

- Uncertainty quantification for ML feature importance for minipatch ensembles

## Conclusion

- **Uncertainty quantification for ML feature importance for minipatch ensembles**
  - Built-in framework: once trained, no extra cost for LOCO inference for all features!
  - Statistical efficiency: all data utilized for training & inference
  - Almost model-agnostic (within minipatch framework)
  - Assumption-light

# Conclusion

- **Uncertainty quantification for ML feature importance for minipatch ensembles**
  - Built-in framework: once trained, no extra cost for LOCO inference for all features!
  - Statistical efficiency: all data utilized for training & inference
  - Almost model-agnostic (within minipatch framework)
  - Assumption-light
- **Open questions**
  - Correct for multiplicity: FDR control?
  - Extension to high-dimensional setting: selective inference for top features?
  - Strengthen the minipatch predictor via adaptive sampling?
  - Beyond tabular data?

# Conclusion

- **Uncertainty quantification for ML feature importance for minipatch ensembles**
  - Built-in framework: once trained, no extra cost for LOCO inference for all features!
  - Statistical efficiency: all data utilized for training & inference
  - Almost model-agnostic (within minipatch framework)
  - Assumption-light
- **Open questions**
  - Correct for multiplicity: FDR control?
  - Extension to high-dimensional setting: selective inference for top features?
  - Strengthen the minipatch predictor via adaptive sampling?
  - Beyond tabular data?

## Conclusion

---

Paper: L. Gan\*, **L. Zheng\***, G. I. Allen (\*: equal contribution), “Model-Agnostic Confidence Intervals for Feature Importance: A Fast and Powerful Approach Using Minipatch Ensembles”, <https://arxiv.org/abs/2206.02088>. *Updated version coming soon!*

Thank you!

## References

---

- Pierre Bayle, Alexandre Bayle, Lucas Janson, and Lester Mackey. Cross-validation confidence intervals for test error. *Advances in Neural Information Processing Systems*, 33:16339–16350, 2020.
- David A Bennett, Aron S Buchman, Patricia A Boyle, Lisa L Barnes, Robert S Wilson, and Julie A Schneider. Religious orders study and rush memory and aging project. *Journal of Alzheimer's disease*, 64(s1):S161–S189, 2018.
- Chien-Ming Chi, Yingying Fan, and Jinchi Lv. Fact: High-dimensional random forests inference. *arXiv preprint arXiv:2207.01678*, 2022.
- Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.
- Byol Kim, Chen Xu, and Rina Foygel Barber. Predictive inference is free with the jackknife+-after-bootstrap. *arXiv preprint arXiv:2002.09025*, 2020.
- Gunnar König, Christoph Molnar, Bernd Bischl, and Moritz Grosse-Wentrup. Relative

feature importance. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9318–9325. IEEE, 2021.

Jason D Lee, Dennis L Sun, Yuekai Sun, and Jonathan E Taylor. Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927, 2016.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Daniel LeJeune, Hamid Javadi, and Richard Baraniuk. The implicit regularization of ordinary least squares ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 3525–3535. PMLR, 2020.

Ruiting Liang and Rina Foygel Barber. Algorithmic stability implies training-conditional coverage for distribution-free prediction methods. *arXiv preprint arXiv:2311.04295*, 2023.

Gilles Louppe and Pierre Geurts. Ensembles on random patches. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 346–361. Springer, 2012.

Alessandro Rinaldo, Larry Wasserman, and Max G'Sell. Bootstrapping and sample splitting for high-dimensional, assumption-lean inference. *The Annals of Statistics*, 47(6):3438–3469, 2019.

Rajen D Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3):1514–1538, 2020.

Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR, 2020.

Sara Van de Geer, Peter Bühlmann, Ya'acov Ritov, and Ruben Dezeure. On

asymptotically optimal confidence regions and tests for high-dimensional models.  
*The Annals of Statistics*, 42(3):1166–1202, 2014.

David S Watson and Marvin N Wright. Testing conditional independence in supervised learning algorithms. *Machine Learning*, 110(8):2107–2129, 2021.

Brian D Williamson, Peter B Gilbert, Noah R Simon, and Marco Carone. A general framework for inference on algorithm-agnostic variable importance. *Journal of the American Statistical Association*, (just-accepted):1–38, 2021.

Tianyi Yao and Genevera I Allen. Feature selection for huge data via minipatch learning. *arXiv preprint arXiv:2010.08529*, 2020.

Tianyi Yao, Daniel LeJeune, Hamid Javadi, Richard G Baraniuk, and Genevera I Allen. Minipatch learning as implicit ridge-like regularization. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 65–68. IEEE, 2021.

Lu Zhang and Lucas Janson. Floodgate: inference for model-free variable importance.  
*arXiv preprint arXiv:2007.01283*, 2020.