Increasing Profitability of Credit Scoring Models with Bias Correction Algorithms

Preprint

Slides

13.05.2022

AWS Interview

Presentation Outline

1. Background

- What is credit scoring?
- What are the business goals?

2. Problem Description

- Sampling bias illustration
- Bias impact on ML models

3. Approach

- Improving model evaluation
- Improving model training

4. Results

- Offline evaluation
- Business impact

Presentation Outline

1. Background

- What is credit scoring?
- What are the business goals?

2. Problem Description

- Sampling bias illustration
- Bias impact on ML models

3. Approach

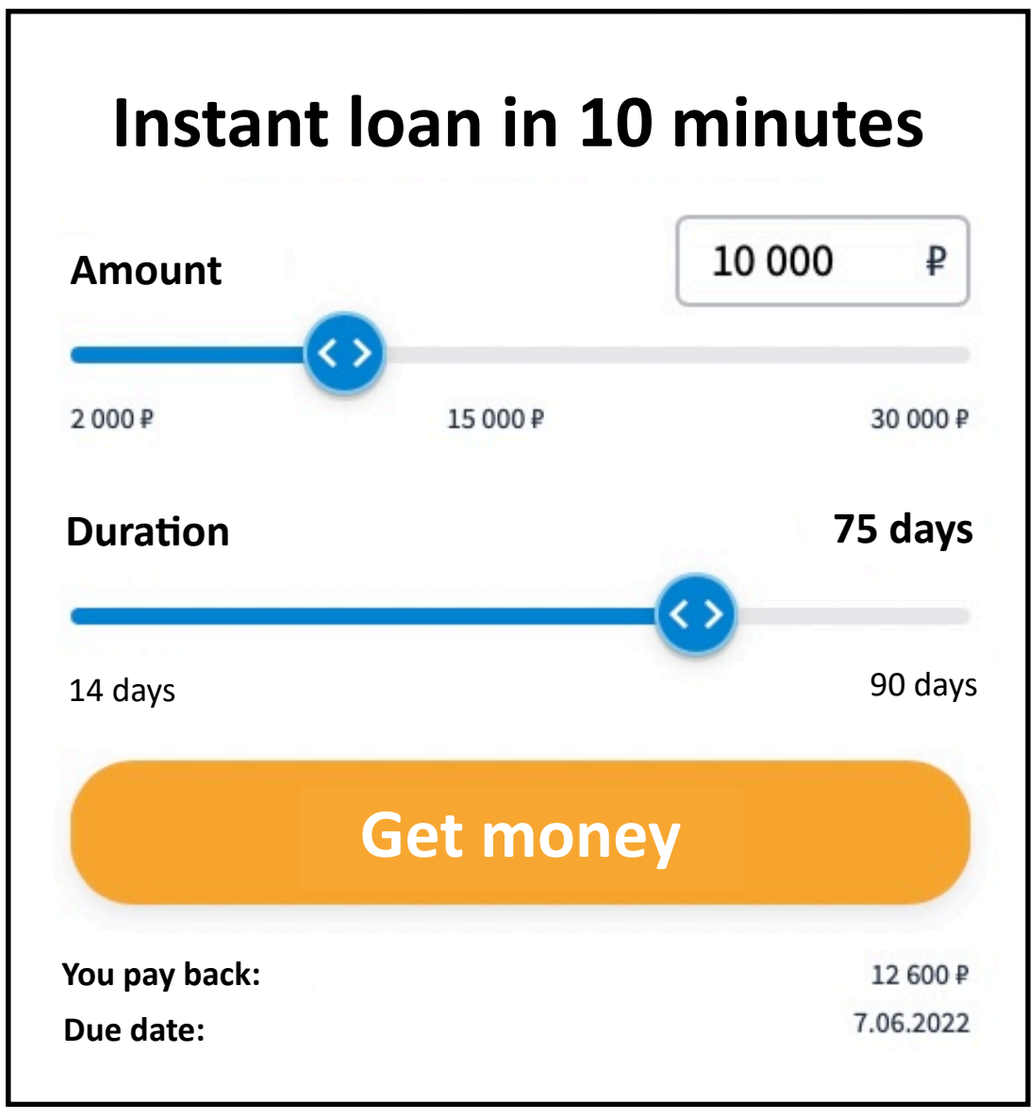
- Improving model evaluation
- Improving model training

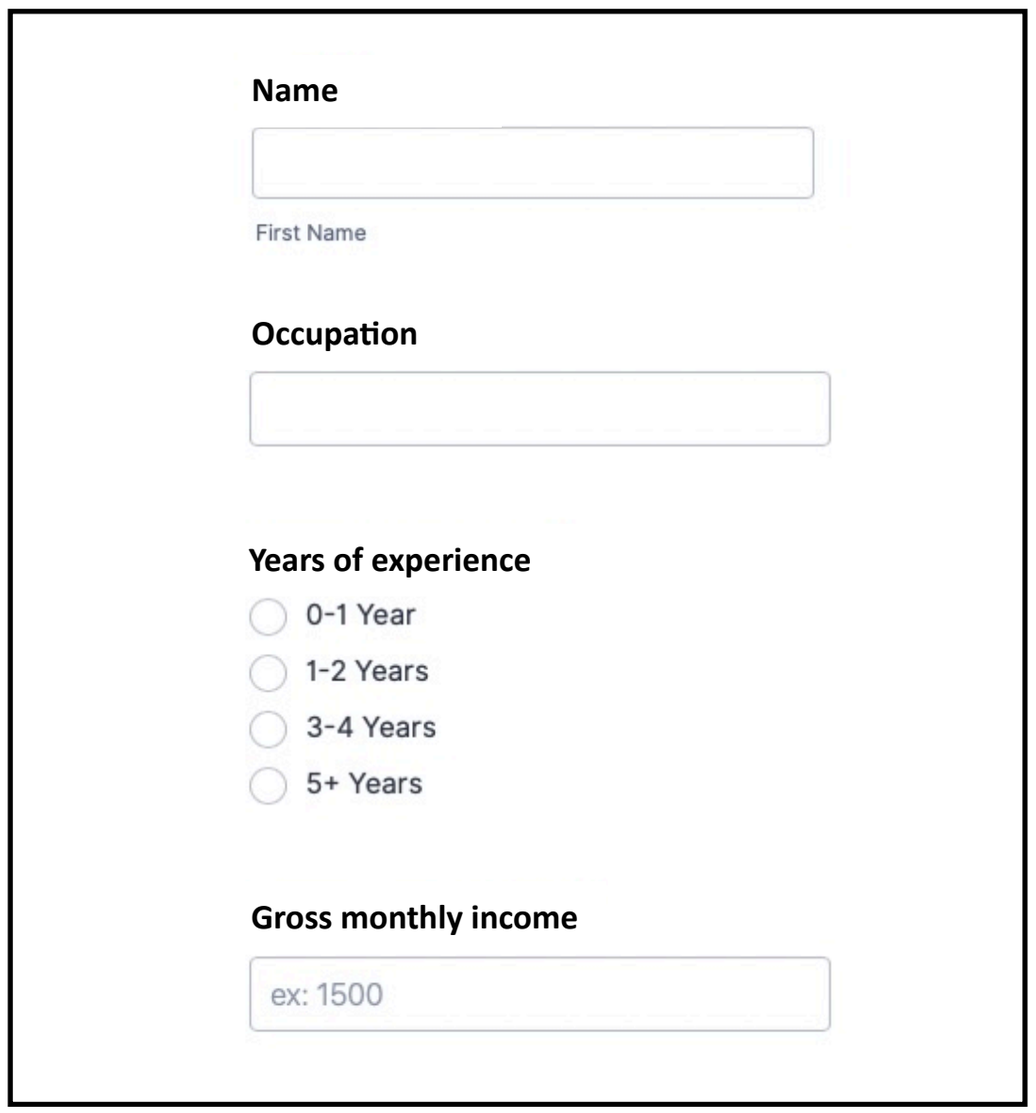
4. Results

- Offline evaluation
- Business impact

What is Credit Scoring?

Customer perspective:





What is Credit Scoring?

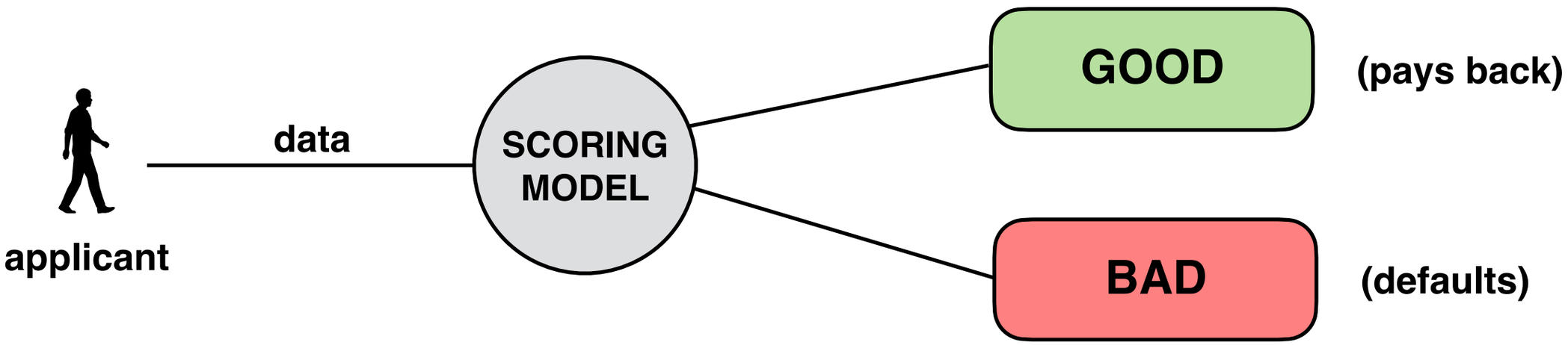
Customer perspective:

Image source: https://www.indusind.com/

What is Credit Scoring?

Business perspective:

- **classification task of distinguishing BAD and GOOD loans**
- **scorecard — model that predicts probability of default**
- **increasing reliance on Machine Learning** *(e.g., Wei et al. 2016)*
  - consumer credit in the US exceeds $4,325 billion[1]
  - FinTechs account for 49.4% of consumer loan market[2]



[1] The Federal Reserve: Statistical Release on Consumer Credit (2021)

[2] Experian: FinTech vs. Traditional FI Trends (2019)

Business Goals

Goal: **improving accuracy of credit scoring models**

Business Goals

Goal: **improving accuracy of credit scoring models**

Costs:

- **accepting BAD customer results in a high loss**
  - business: loss = amount that the client does not pay back
  - customer: long-term financial difficulties
- **rejecting GOOD customer results in a moderate loss**
  - business: loss = potential interest and fees earned from the client
  - customer: limited access to finance

Outcome \ Decision	Accept	Reject
GOOD	+ interest	- interest
BAD	- amount	0

Business Goals

Goal: **improving accuracy of credit scoring models**

Costs:

- **accepting BAD customer results in a high loss**
  - business: loss = amount that the client does not pay back
  - customer: long-term financial difficulties
- **rejecting GOOD customer results in a moderate loss**
  - business: loss = potential interest and fees earned from the client
  - customer: limited access to finance

Project goal:

- **maximize scorecard profitability**
  - minimize **BAD** rate among accepts

		Decision: Accept	Decision: Reject
Outcome	GOOD	+ interest	- interest
Outcome	BAD	- amount	0

Presentation Outline

1. Background

- What is credit scoring?
- What are the business goals?

2. Problem Description

- Sampling bias illustration
- Bias impact on ML models

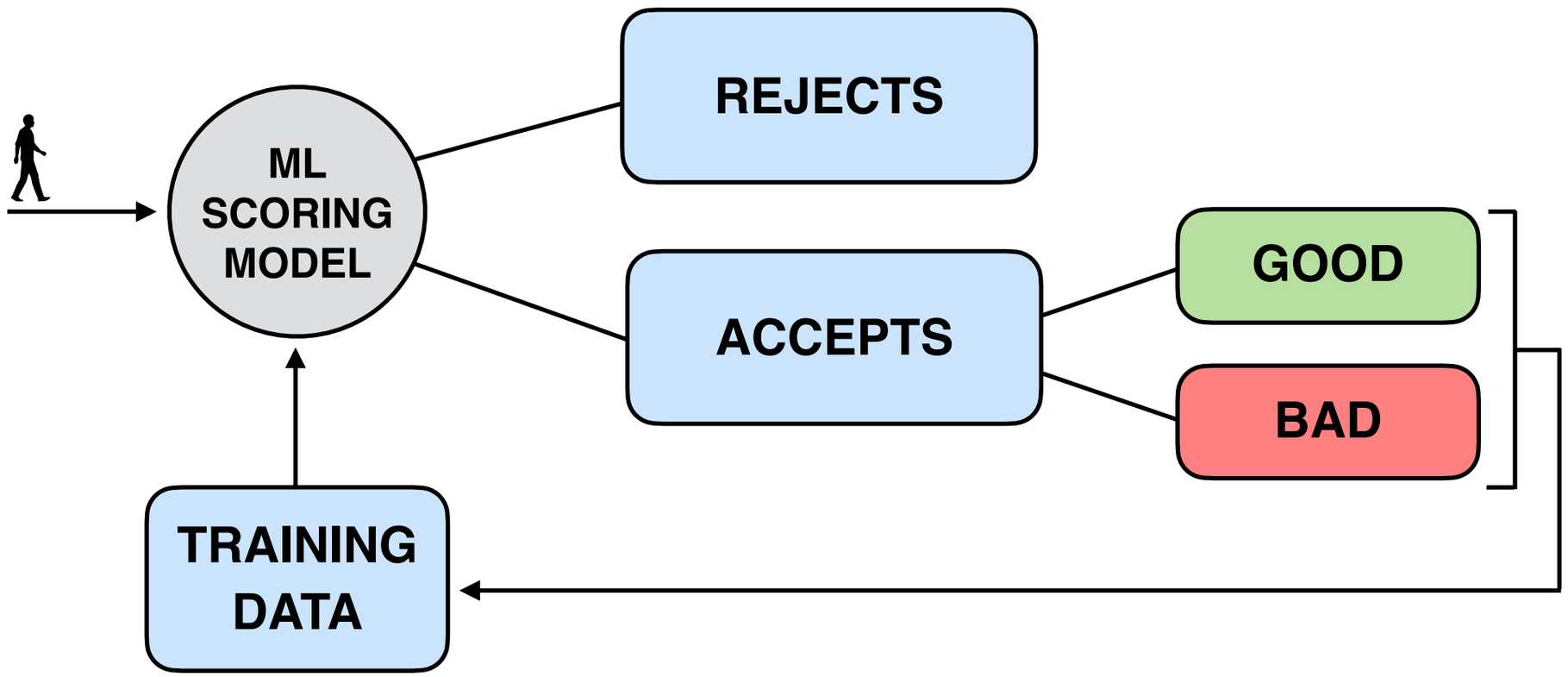
3. Approach

- Improving model evaluation
- Improving model training

4. Results

- Offline evaluation
- Business impact

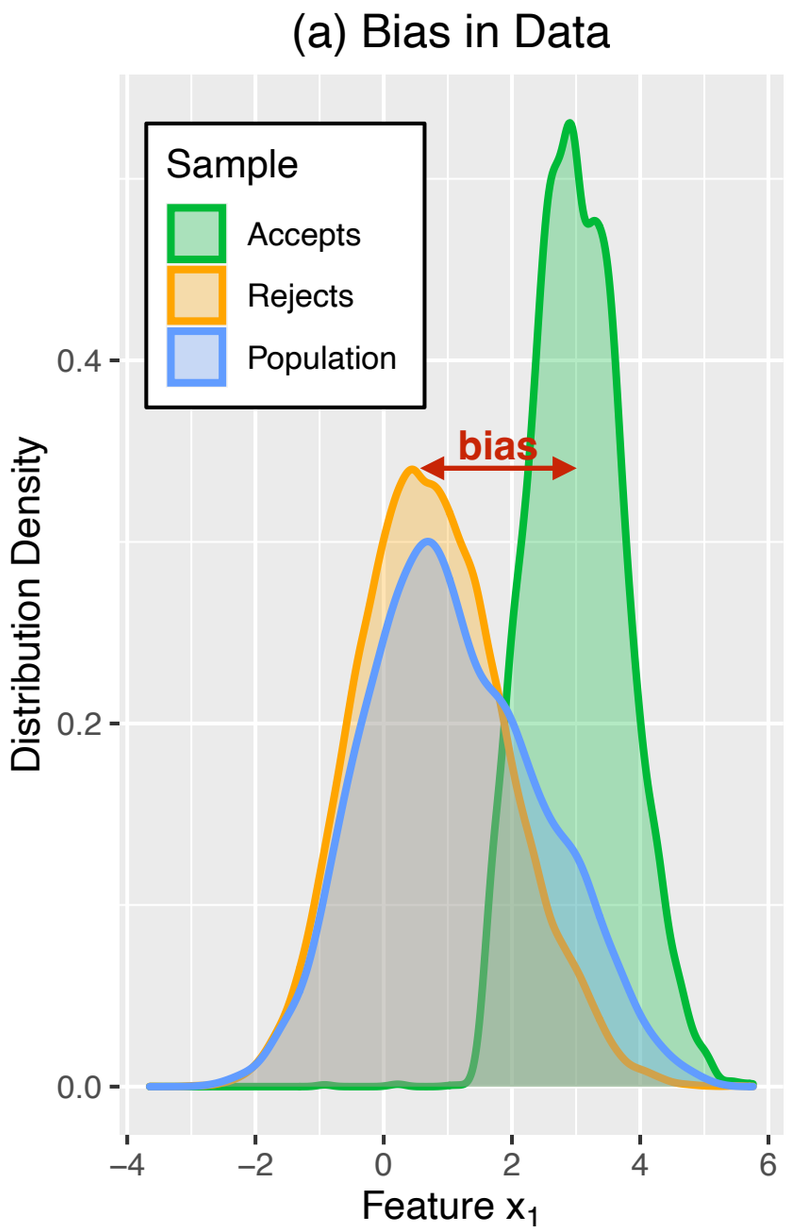
Loan Approval Process at Monedo



- **scoring model filters incoming loan applications**
  - ML model observes applicants' features and predicts **P(GOOD)**
  - top-ranked applicants are accepted and receive a loan
- **training a model requires data with known outcomes**
  - outcomes are only observed for previously **accepted clients**
  - labels of **rejects** are missing not at random *(Crook et al. 2004)*
  - historical data suffers from sampling bias

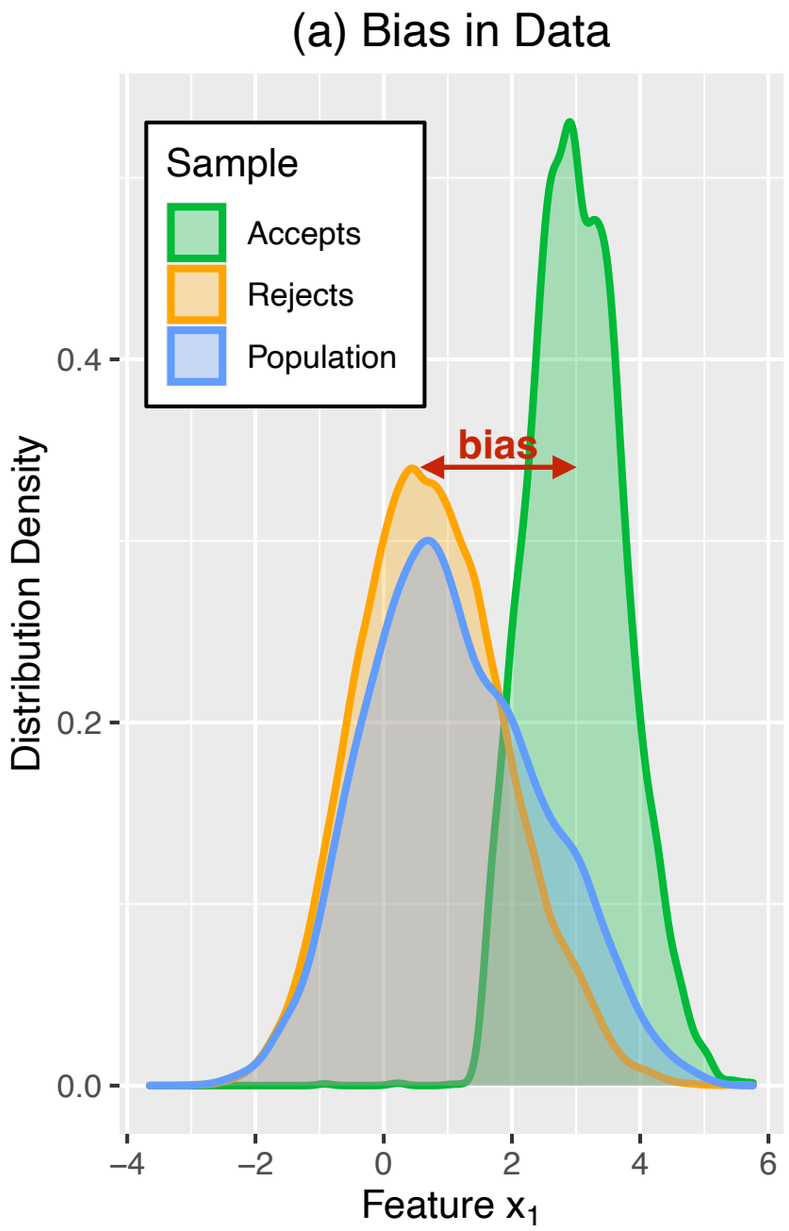
Sampling Bias Illustration

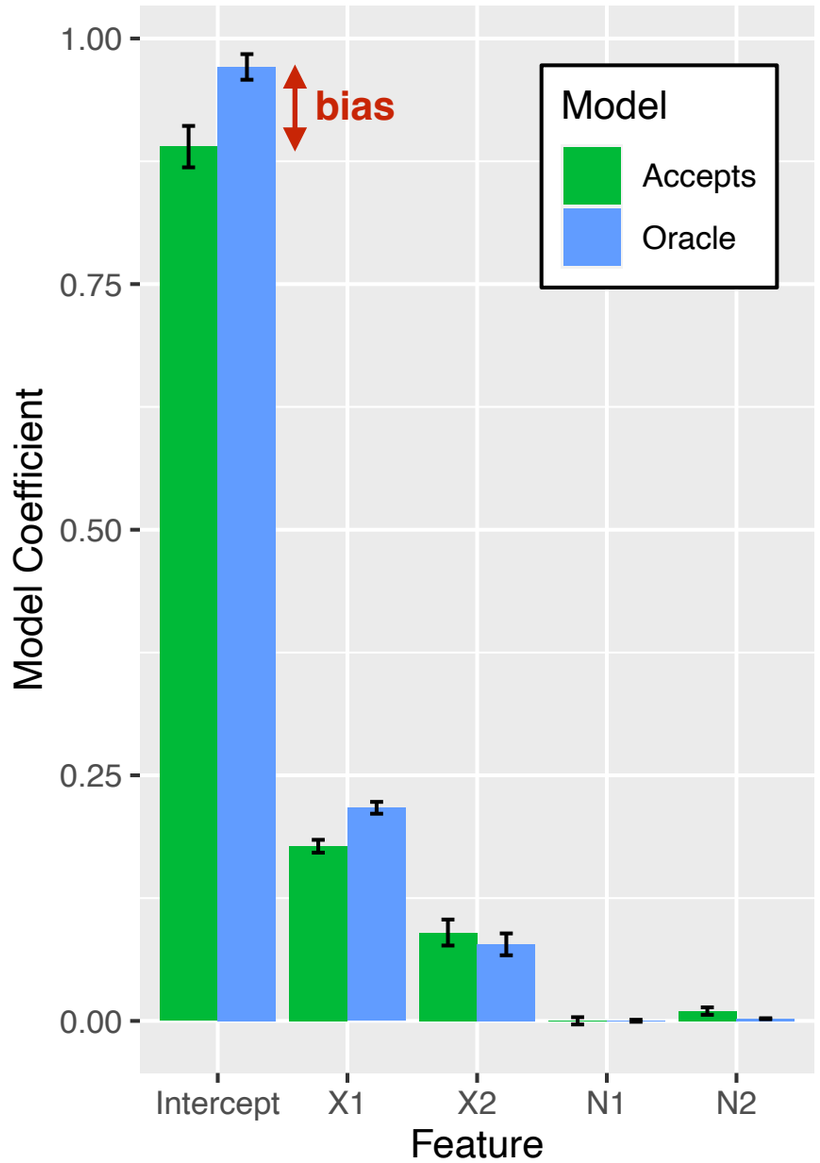
- **sampling bias** originates in the **training data**



Sampling Bias Illustration

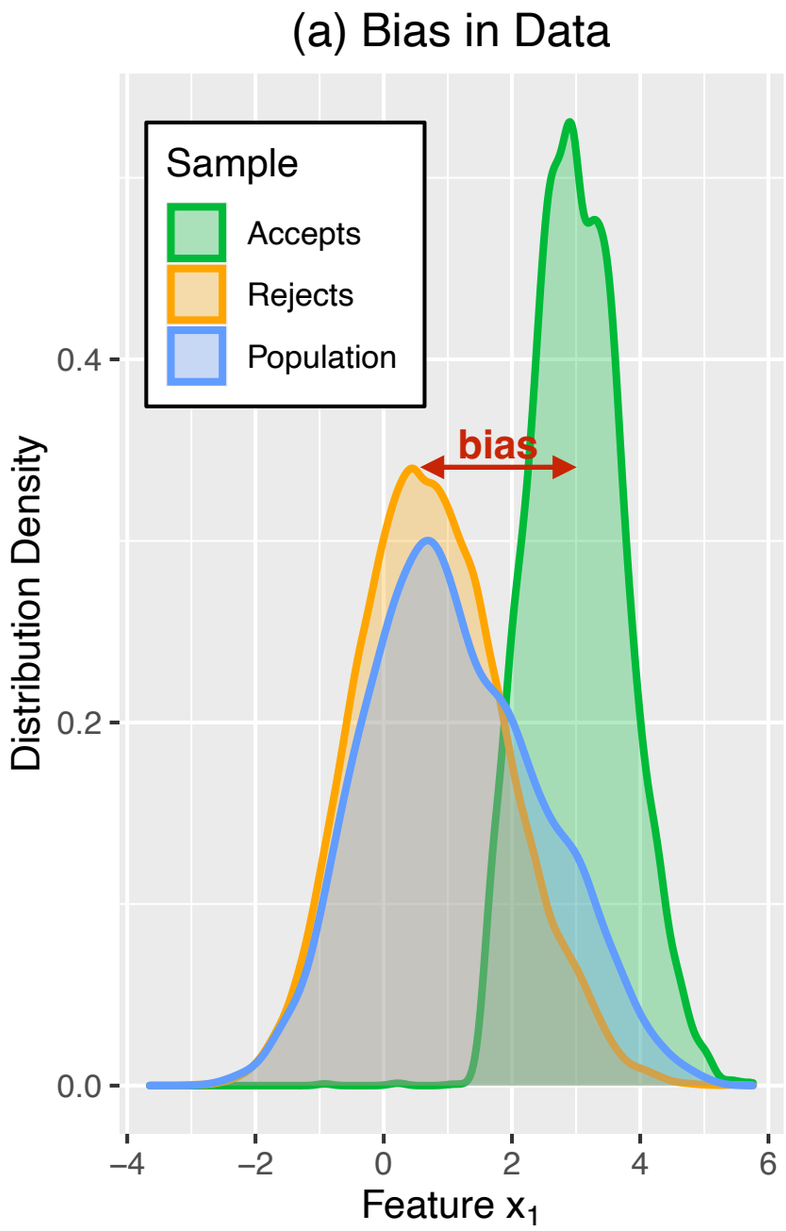
- **sampling bias** originates in the **training data**
- propagates to the **model parameters**

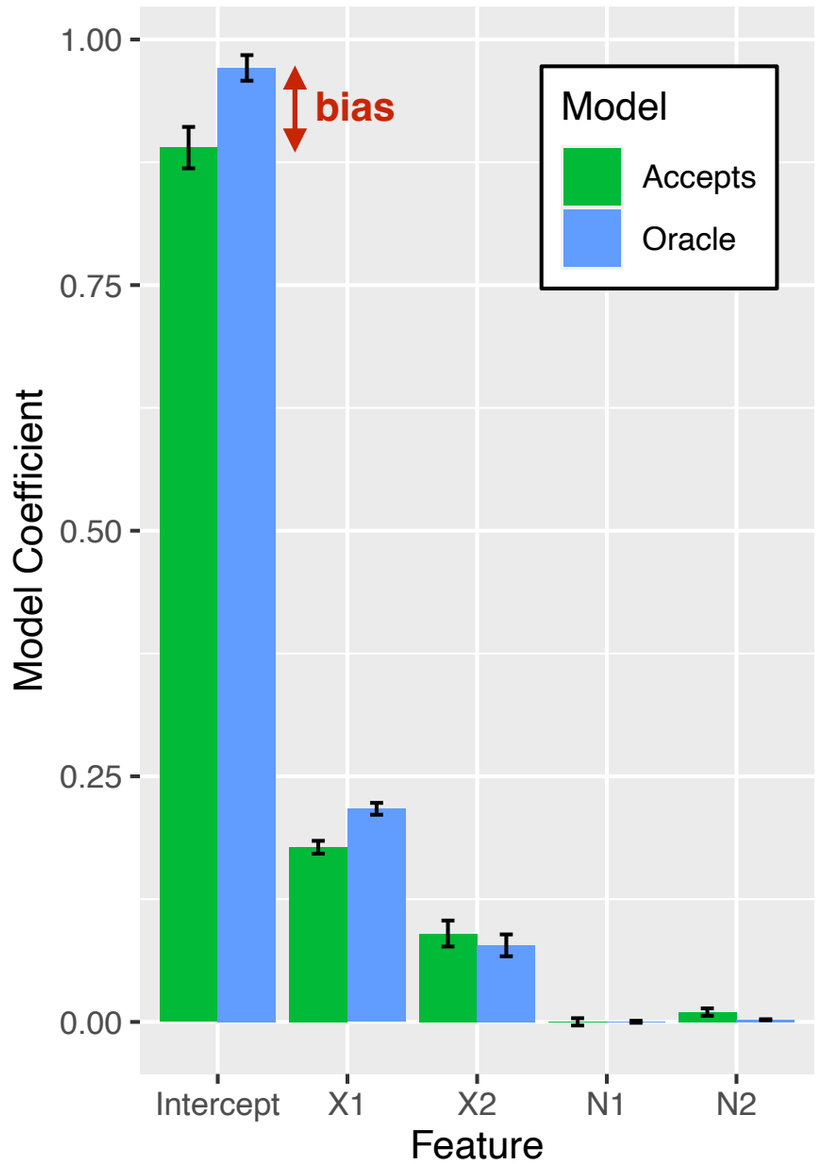


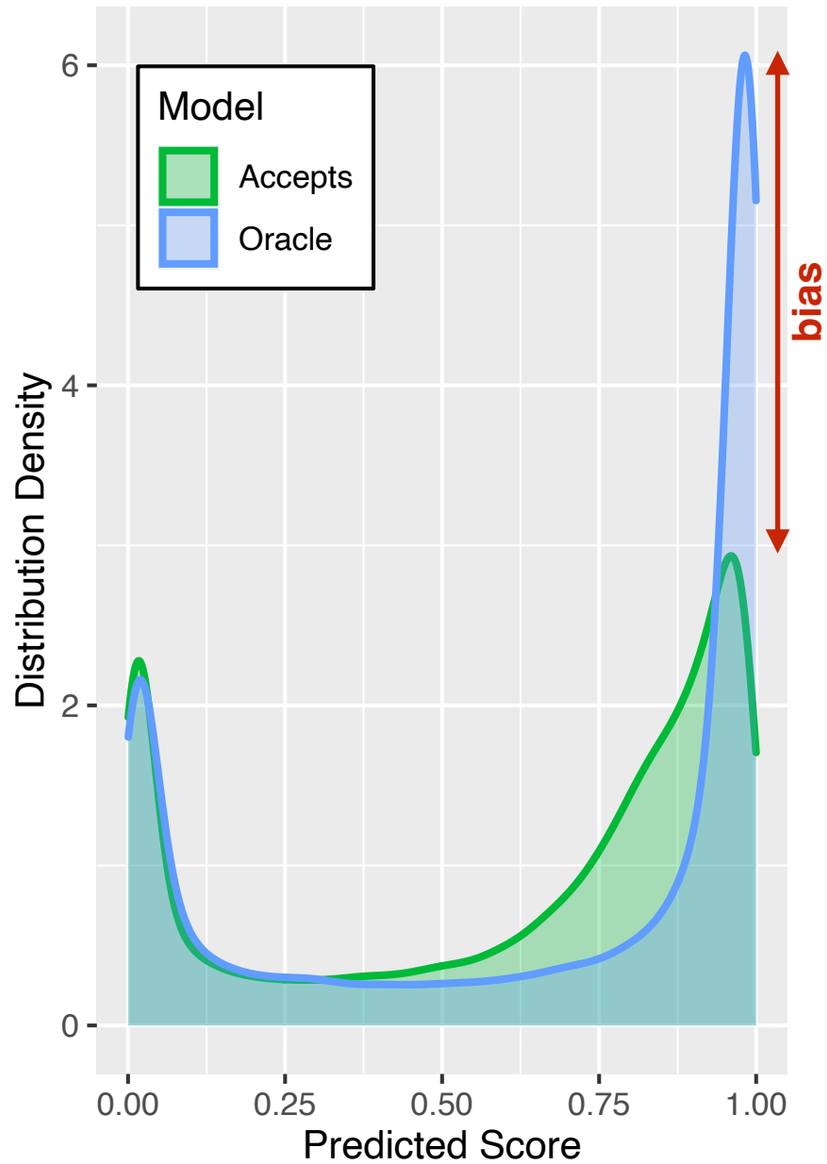


Sampling Bias Illustration

- **sampling bias** originates in the **training data**
- propagates to the **model parameters**
- and affects **model predictions**

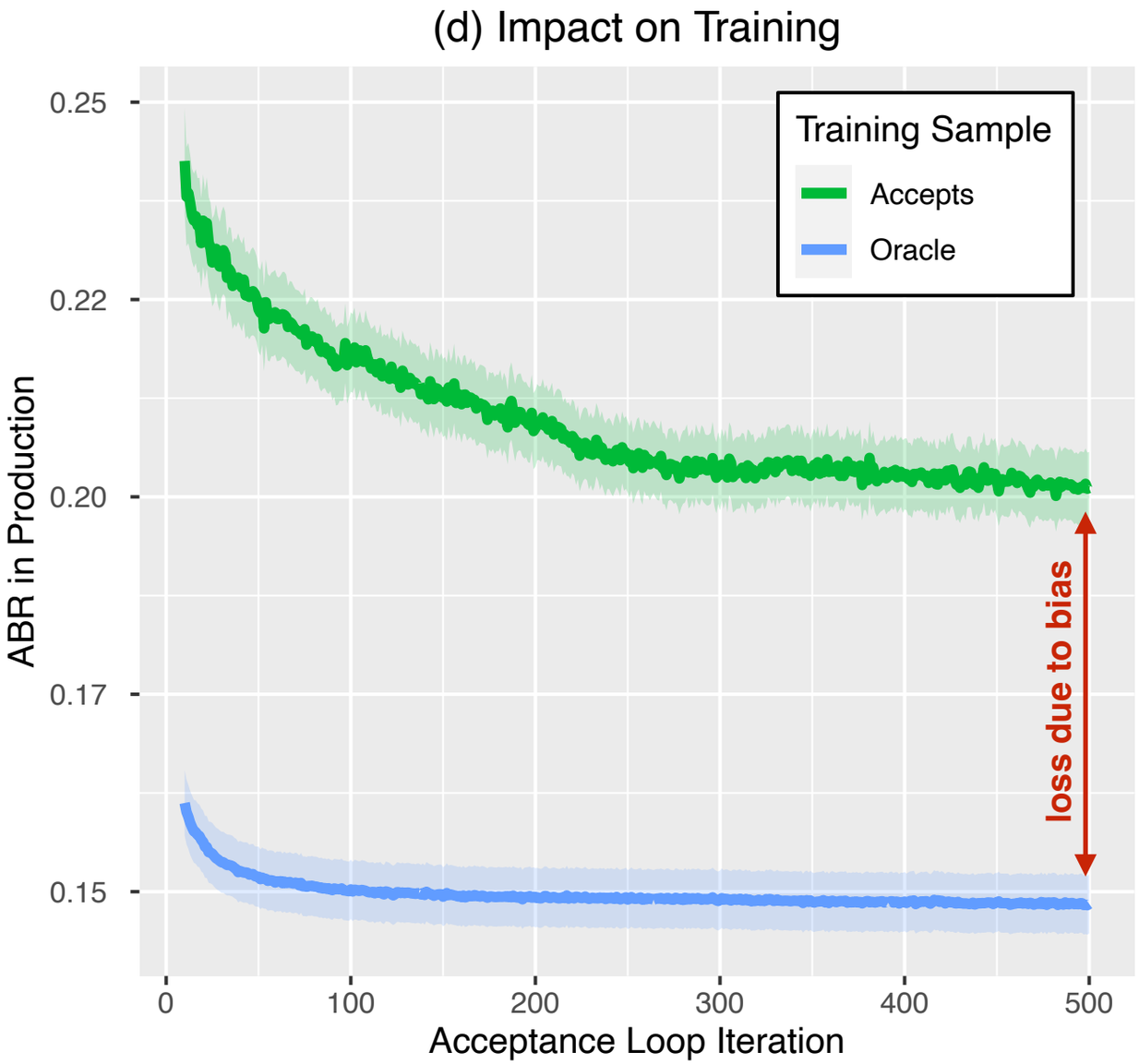






Sampling Bias Consequences

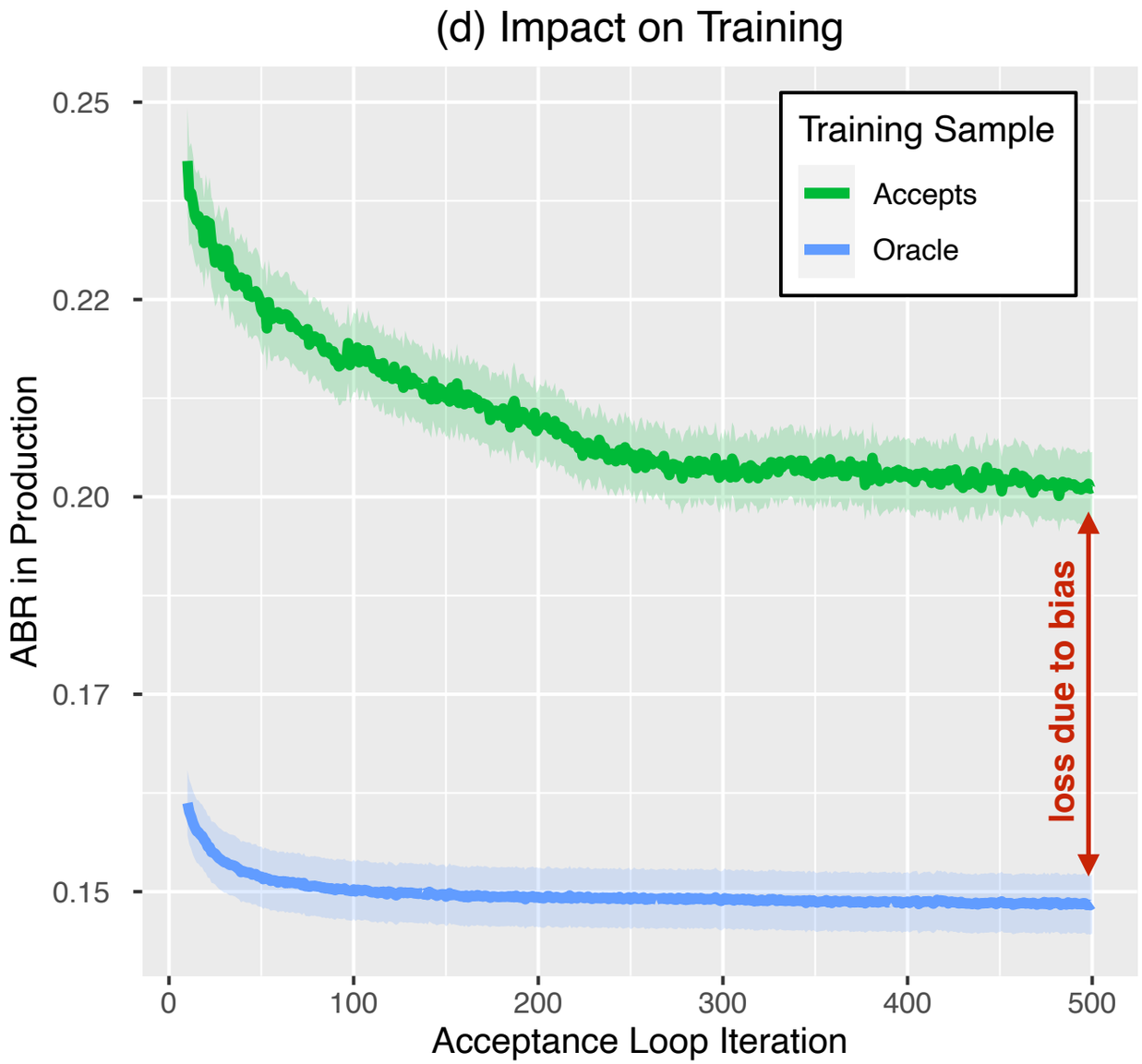
- **training a model** on a biased sample **decreases its production performance**



ABR = **BAD** rate when accepting top-30% applicants; lower is better

Sampling Bias Consequences

- **training a model** on a biased sample **decreases its production performance**



Decision

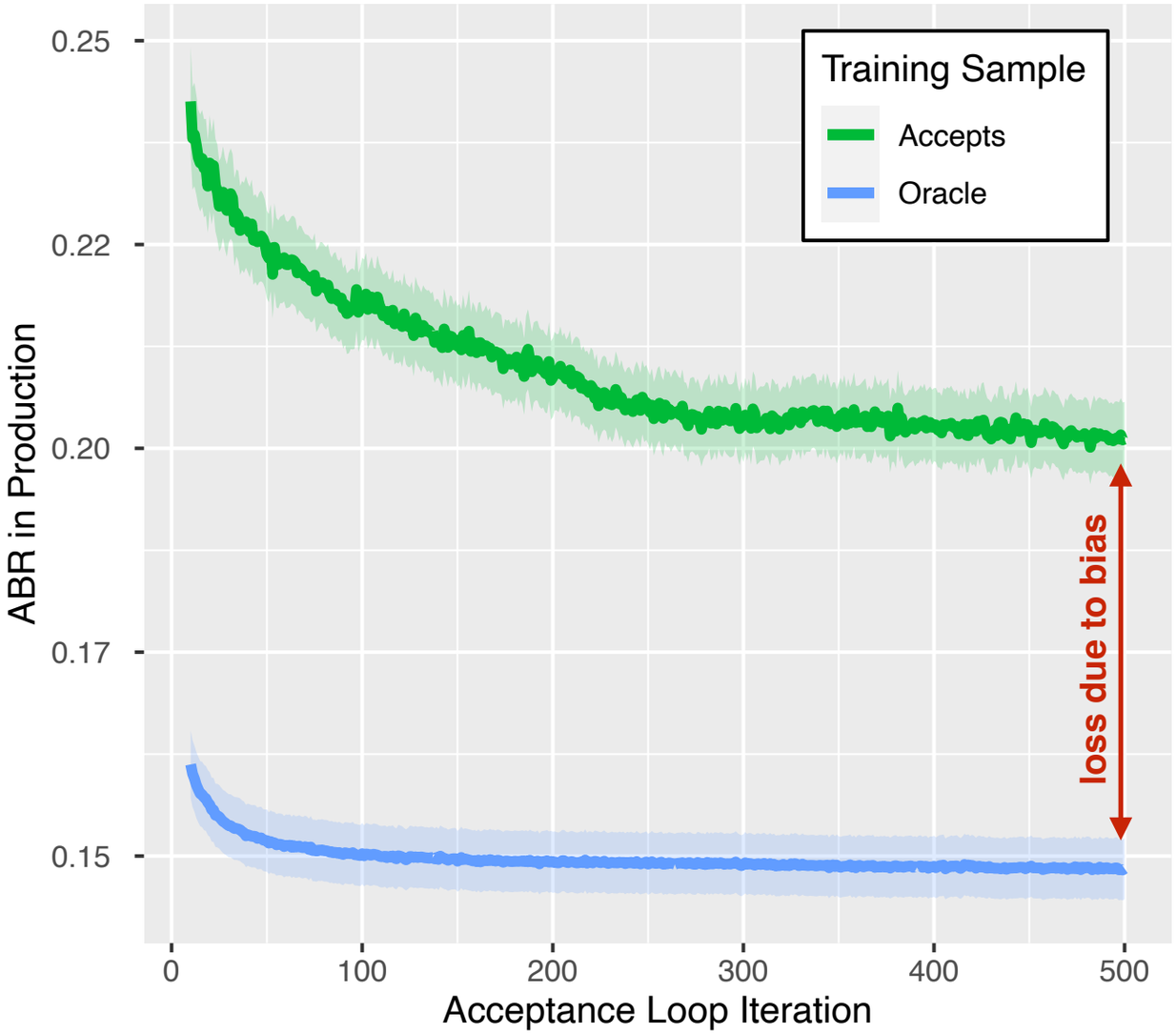
Outcome	Accept	Reject
GOOD	+ interest	- interest
BAD	- amount	0

ABR = **BAD** rate when accepting top-30% applicants; lower is better

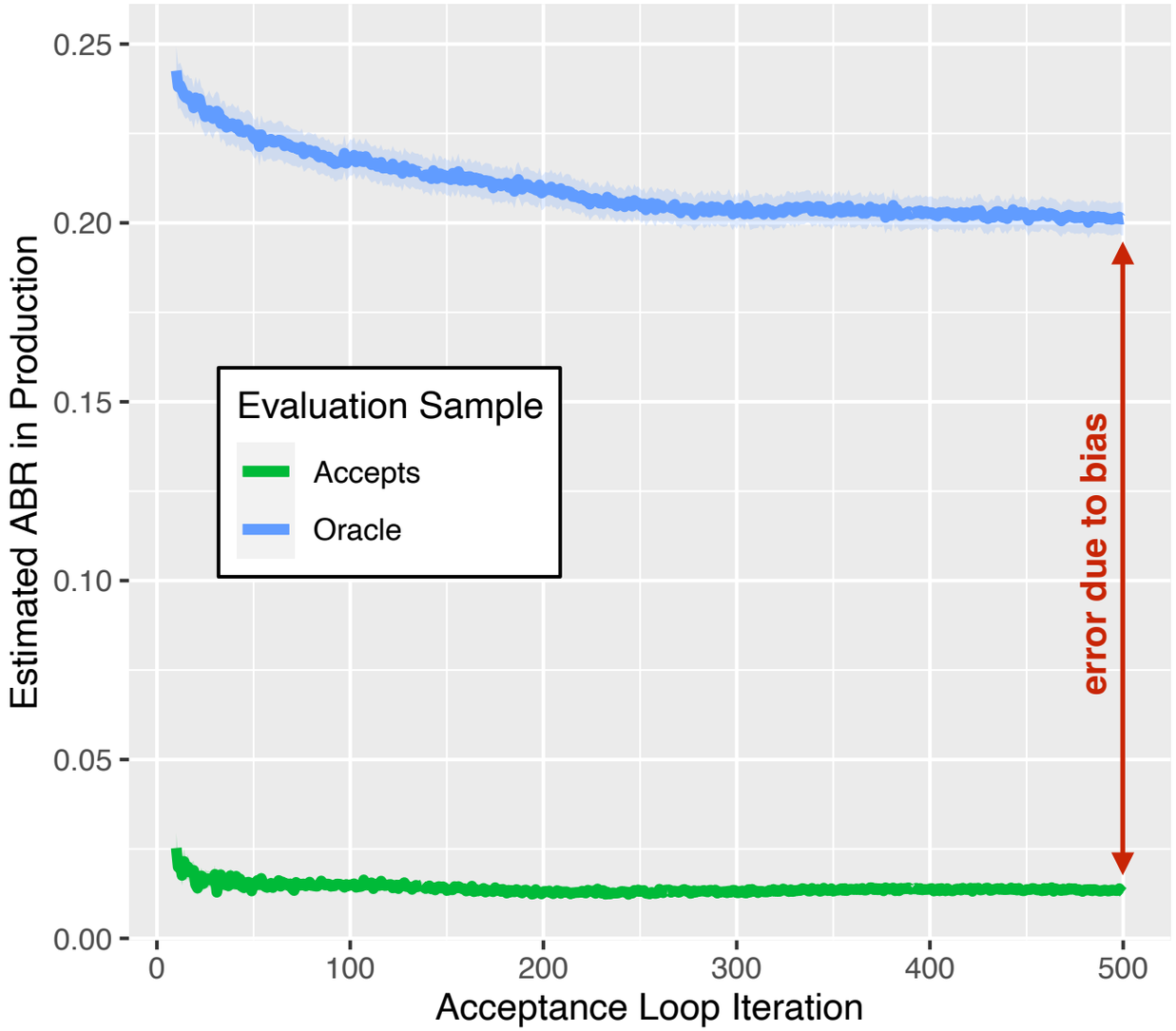
Sampling Bias Consequences

- **training a model** on a biased sample **decreases its production performance**
- **evaluating a model** on a biased sample provides a **misleading estimate**

(d) Impact on Training



(e) Impact on Evaluation

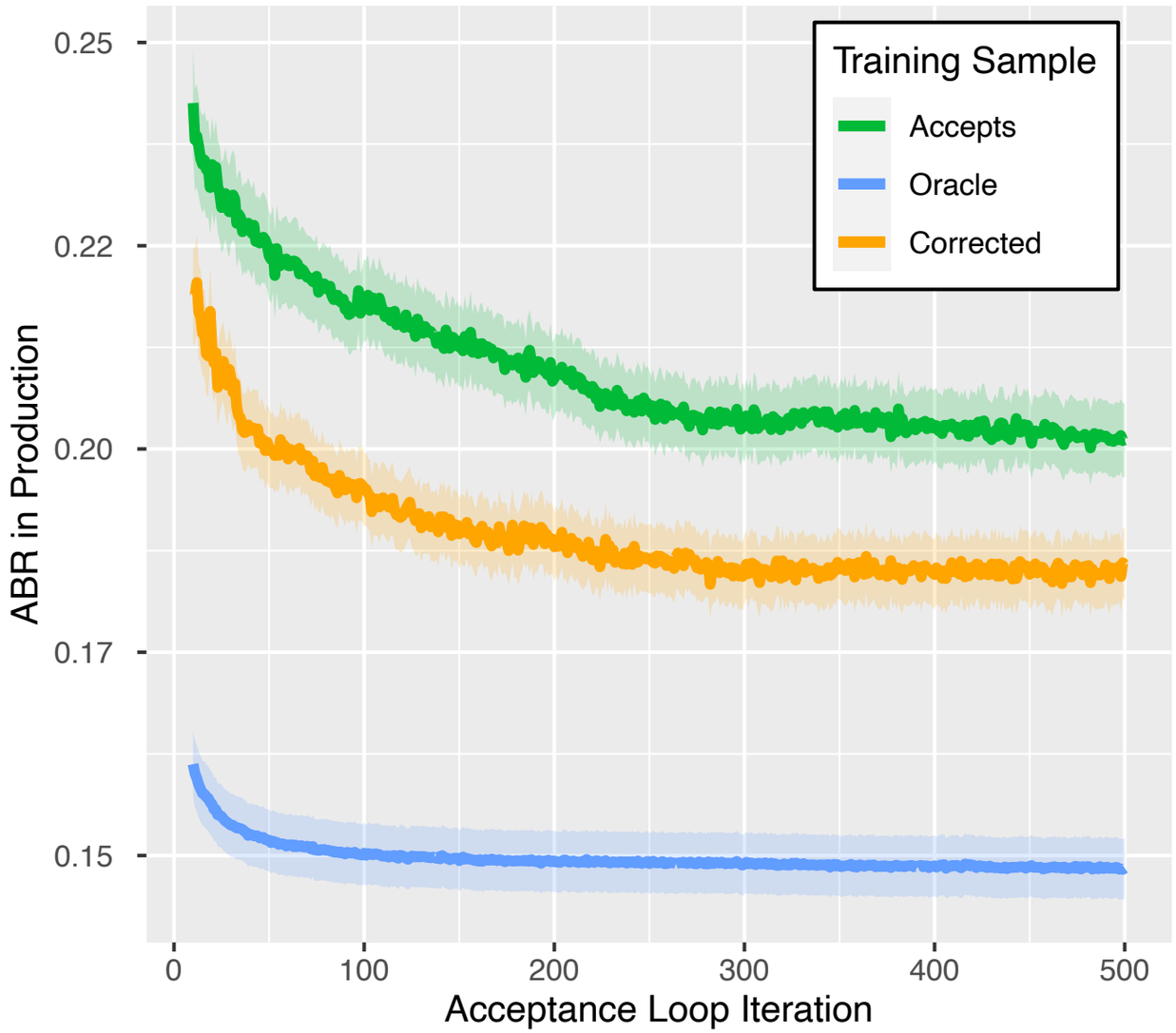


ABR = **BAD** rate when accepting top-30% applicants; lower is better

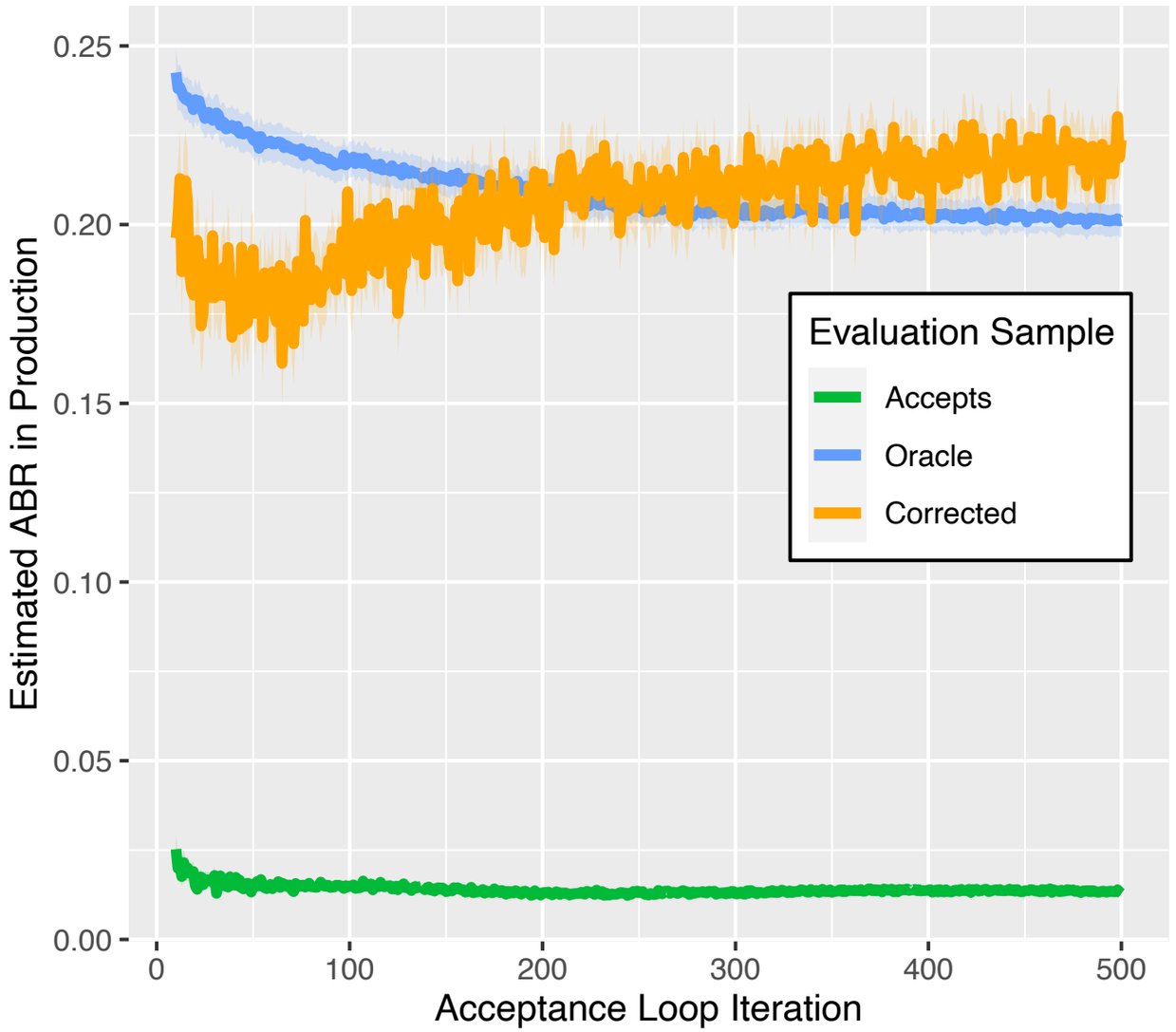
Potential Performance Gains

- **bias correction** can **improve the model performance in production**
- **bias correction** can provide a **better estimate of production performance**

(d) Impact on Training



(e) Impact on Evaluation



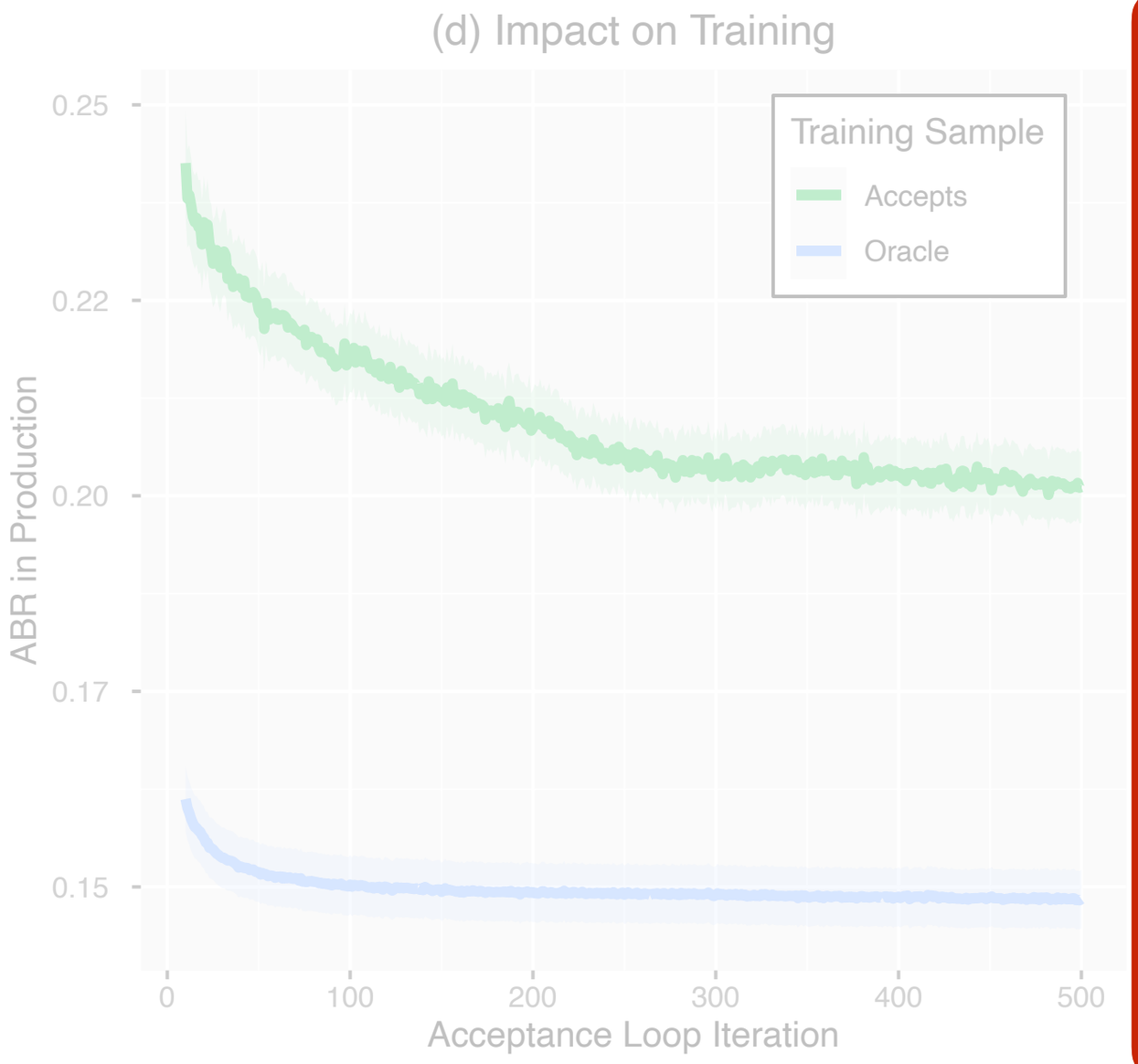
ABR = **BAD** rate when accepting top-30% applicants; lower is better

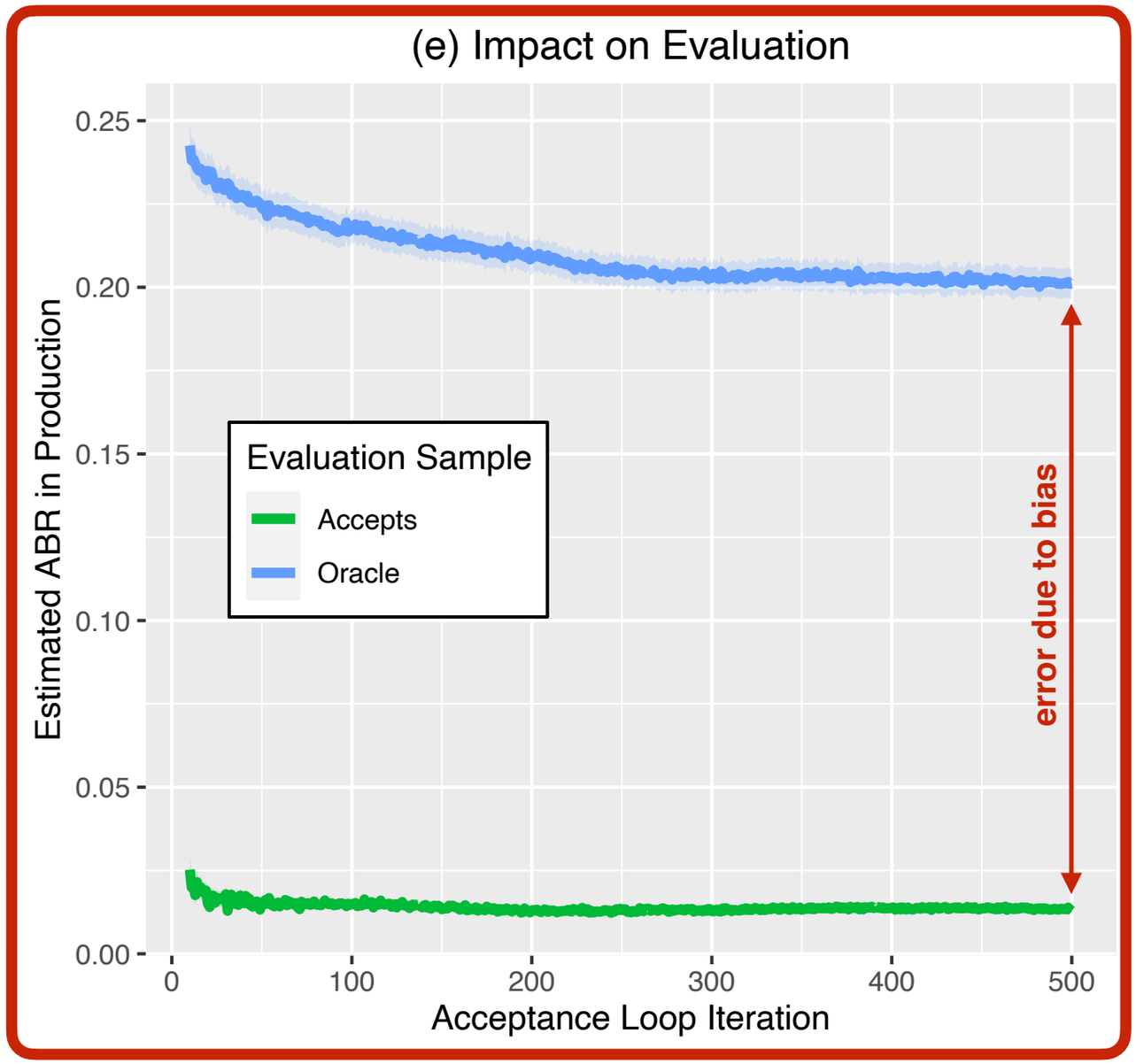
Presentation Outline

1. **Background**
   - What is credit scoring?
   - What are the business goals?
2. **Problem Description**
   - Sampling bias illustration
   - Bias impact on ML models
3. **Approach**
   - Improving model evaluation
   - Improving model training
4. **Results**
   - Offline evaluation
   - Business impact

Bias Impact on Evaluation

- **training a model** on a biased sample **decreases its production performance**
- **evaluating a model** on a biased sample provides a **misleading estimate**





ABR = **BAD** rate when accepting top-30% applicants; lower is better

Evaluation under Sampling Bias

How to improve evaluation?

Collect unbiased sample

- completely avoids sampling bias
- requires issuing loans to **random set of applicants** without scoring
- **issue:** very costly

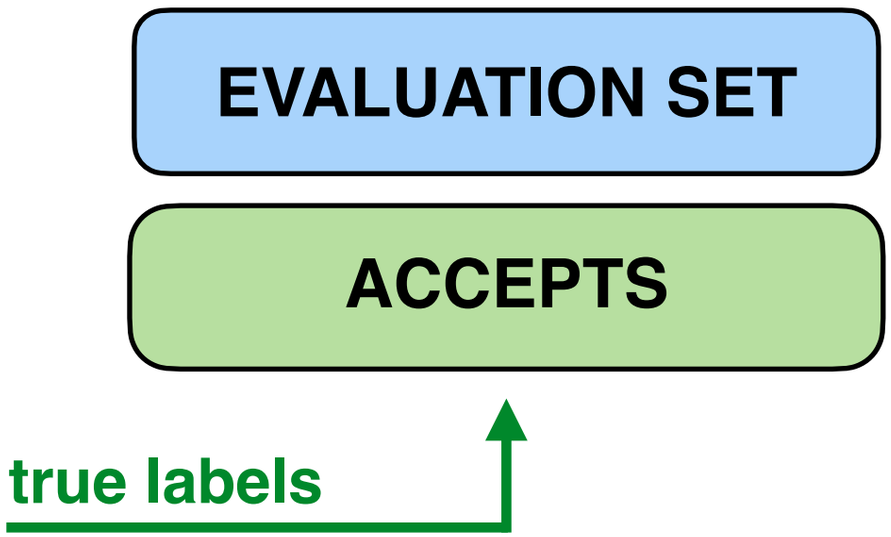
Adjust evaluation framework

- use bias correction methods to account for **distribution mismatch**
- **issue:** labels of **rejects** are unknown

Standard Practice: Evaluate on Accepts

Idea:

- evaluate metric ***M*** on evaluation set containing labeled **accepts**



State-of-the-Art: Reweighting

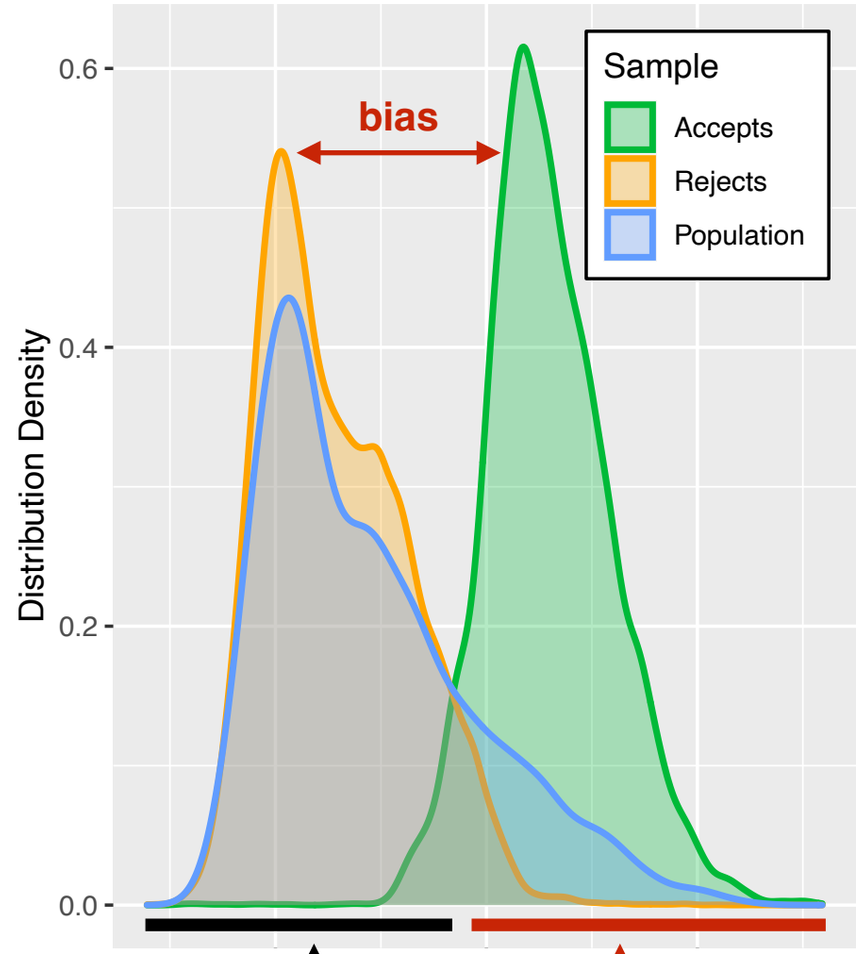
Idea:

- evaluate metric ***M*** on evaluation set containing labeled **accepts**
- reweigh the metric to focus on **representative cases**

EVALUATION SET

ACCEPTS

true labels



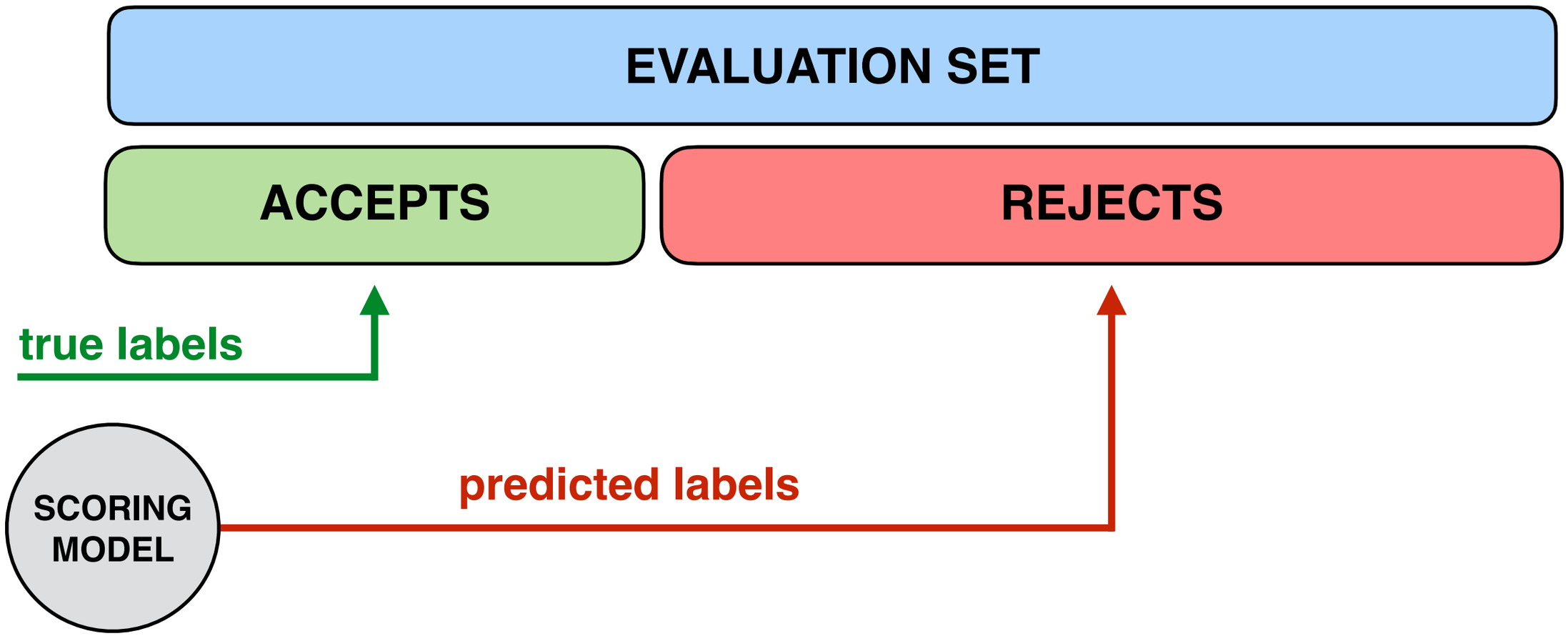
higher weight

lower weight

Bayesian Evaluation (BE)

Idea:

- evaluate metric ***M*** on evaluation set containing:
  - labeled **accepts**
  - pseudo-labeled **rejects**
- estimate prior **P(BAD)** for **rejects** using the **current scorecard *f(X)***



Bayesian Evaluation (BE)

Idea:

- evaluate metric ***M*** on evaluation set containing:
  - labeled **accepts**
  - pseudo-labeled **rejects**
- estimate prior **P(BAD)** for **rejects** using the **current scorecard *f(X)***

input : model $f(X)$, evaluation sample S consisting of labeled accepts $S^a = \{(\mathbf{X}^a, \mathbf{y}^a)\}$ and unlabeled rejects $\mathbf{X}^r$, prior $\mathbf{P}(\mathbf{y}^r|X^r)$, evaluation metric $M(f, S, \tau)$, meta-parameters j_{max}, ϵ

output: Bayesian evaluation metric $\mathrm{BM}(f, S, \tau)$

$j = 0$; $\Delta = \epsilon$; $E^c = \{\}$; // initialization

while $(j \leq j_{max})$ *and* $(\Delta \geq \epsilon)$ **do**

  $j = j + 1$

  $\mathbf{y}^r = \mathrm{binomial}(1, \mathbf{P}(\mathbf{y}^r|\mathbf{X}^r))$; // generate labels of rejects

  $S_j = \{(\mathbf{X}^a, \mathbf{y}^a)\} \cup \{(\mathbf{X}^r, \mathbf{y}^r)\}$; // construct evaluation sample

  $E^c_j = \sum_{i=1}^{j} M(f(X), S_i, \tau)/j$; // evaluate

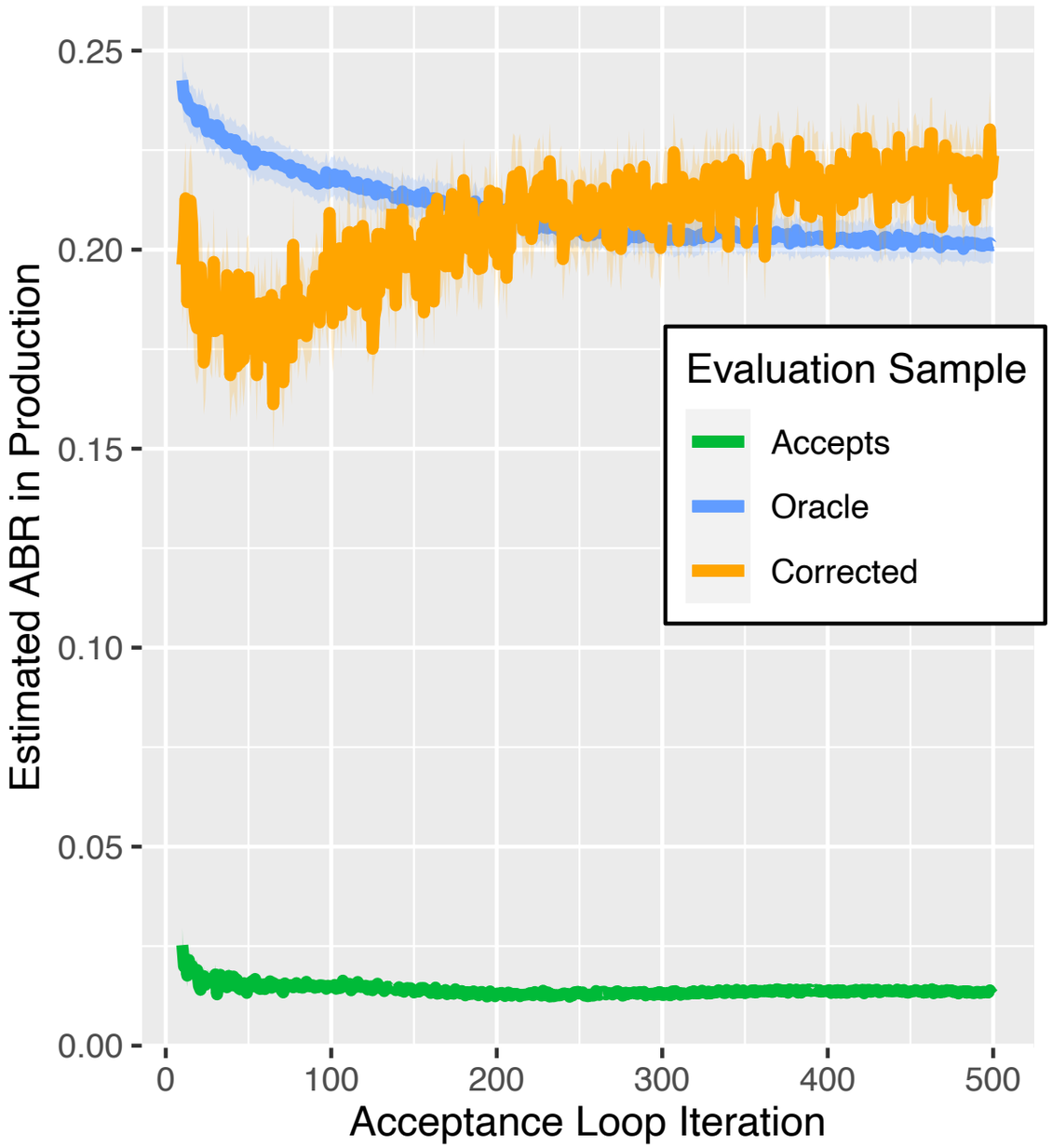
  $\Delta = E^c_j - E^c_{j-1}$; // check convergence

end

return $BM(f, S, \tau) = E^c_j$

BE: Simulation Results

Performance Dynamics



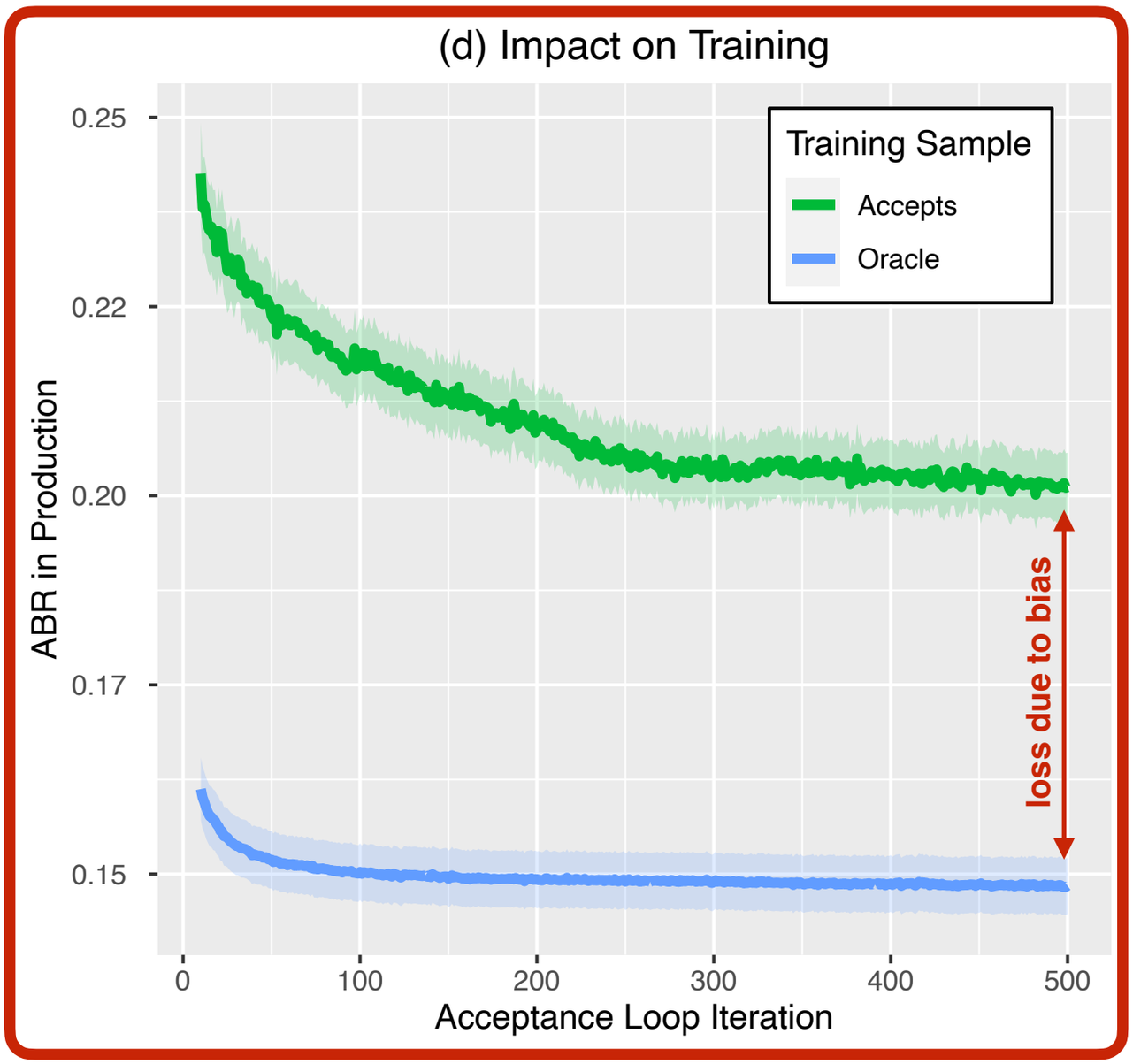
Aggregated Results

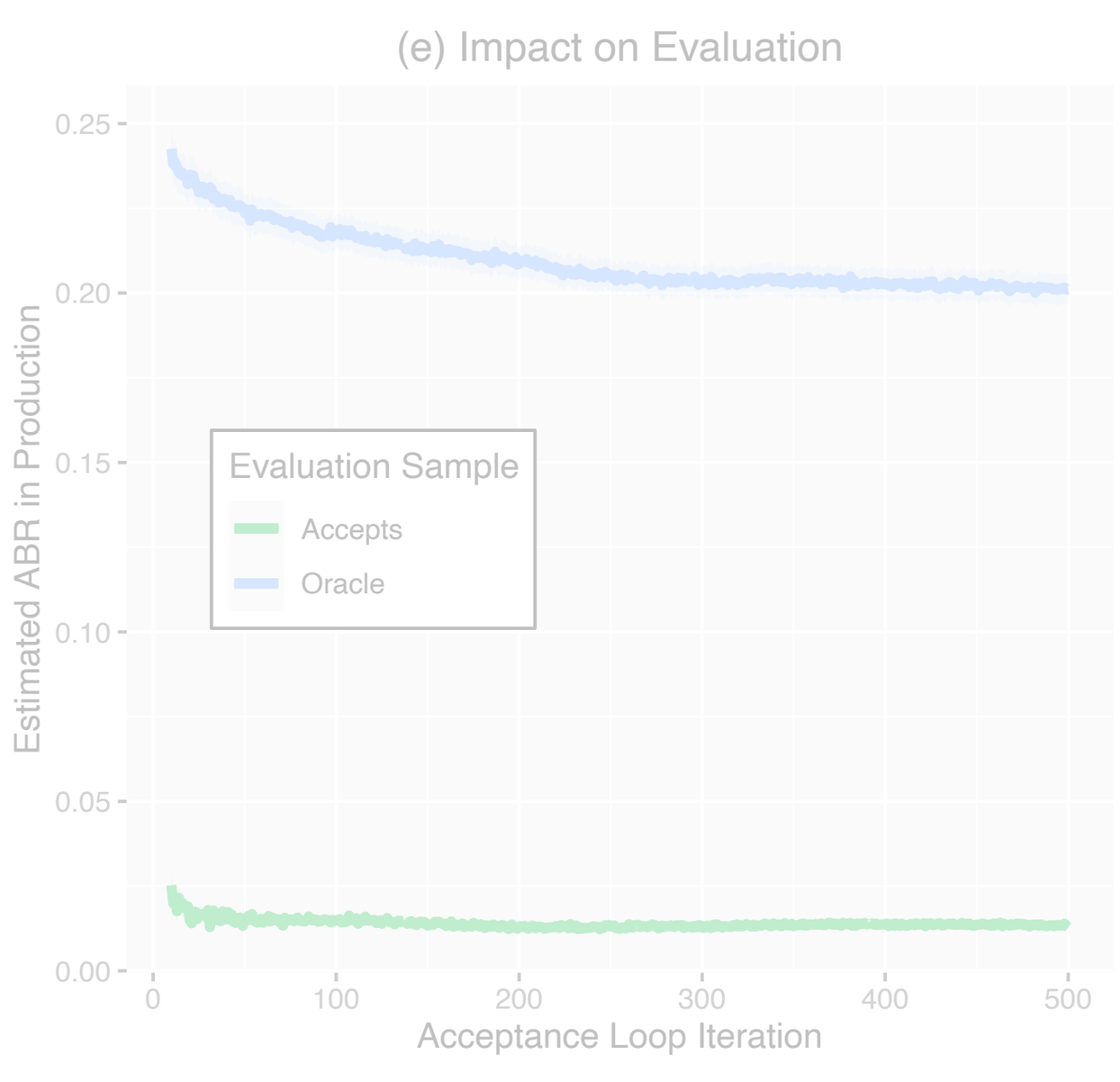
Metric	RMSE due to bias	Gains from BE
ABR	.2058	**55.83%**
BS	.0829	**36.55%**
AUC	.2072	**67.57%**
PAUC	.2699	**70.80%**

- BE improves **performance estimates**
- gains are **statistically significant** at 5%

Bias Impact on Training

- **training a model** on a biased sample **decreases its production performance**
- **evaluating a model** on a biased sample provides a **misleading estimate**





ABR = **BAD** rate when accepting top-30% applicants; lower is better

Training under Sampling Bias

How to improve training?

Collect unbiased sample

- completely avoids sampling bias
- **issue:** very costly

Data augmentation (label rejects)

- **predict labels** of **rejects**
- use combined data of **accepts** and **rejects** for model training
- **issue:** high risk of error propagation

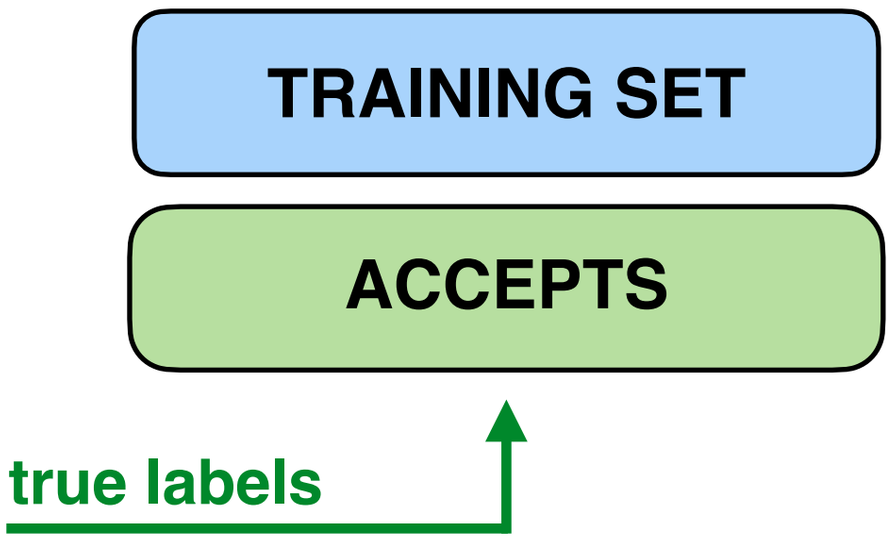
Extract information from rejects

- estimate **distribution mismatch** between **accepts** and **rejects**
- modify training procedure
- **issue:** hard in high-dimensional data

Standard Practice: Train on Accepts

Idea:

- train model ***f(x)*** on training set containing labeled **accepts**



State-of-the-Art: Reweighting

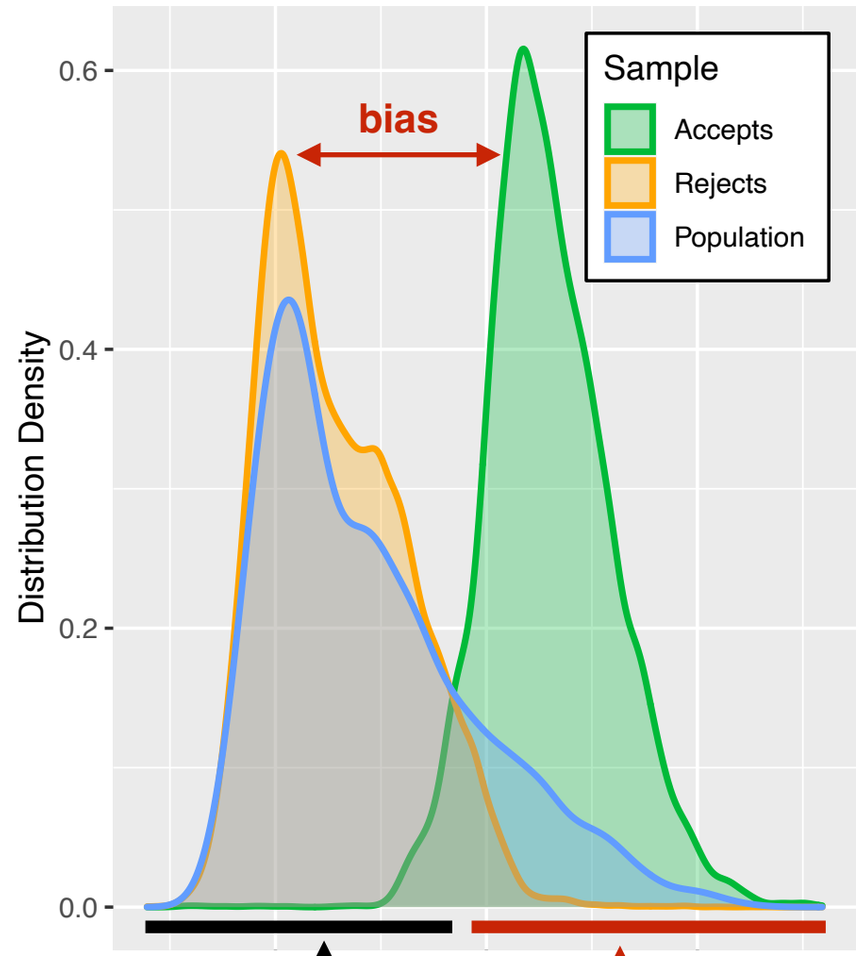
Idea:

- train model ***f(x)*** on training set containing labeled **accepts**
- reweigh model loss to focus on **representative cases**

TRAINING SET

ACCEPTS

true labels



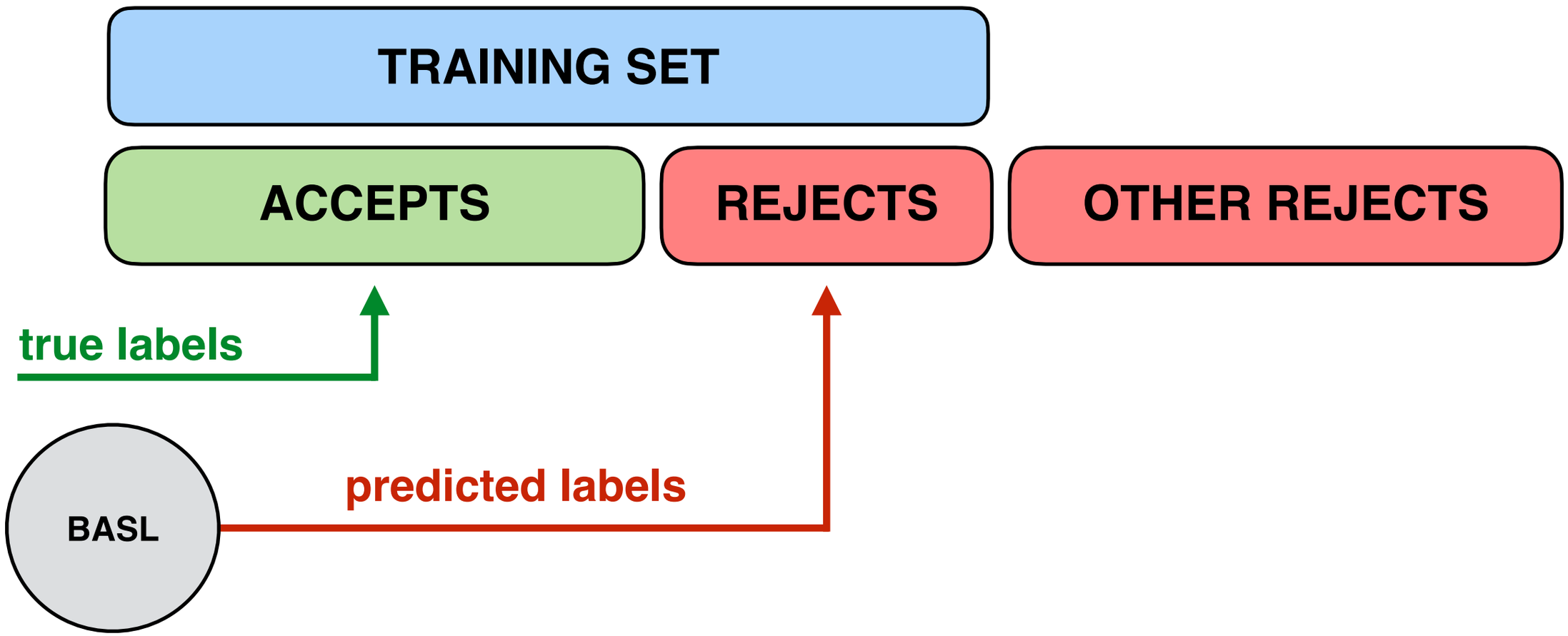
higher weight

lower weight

Bias-Aware Self-Learning (BASL)

Idea:

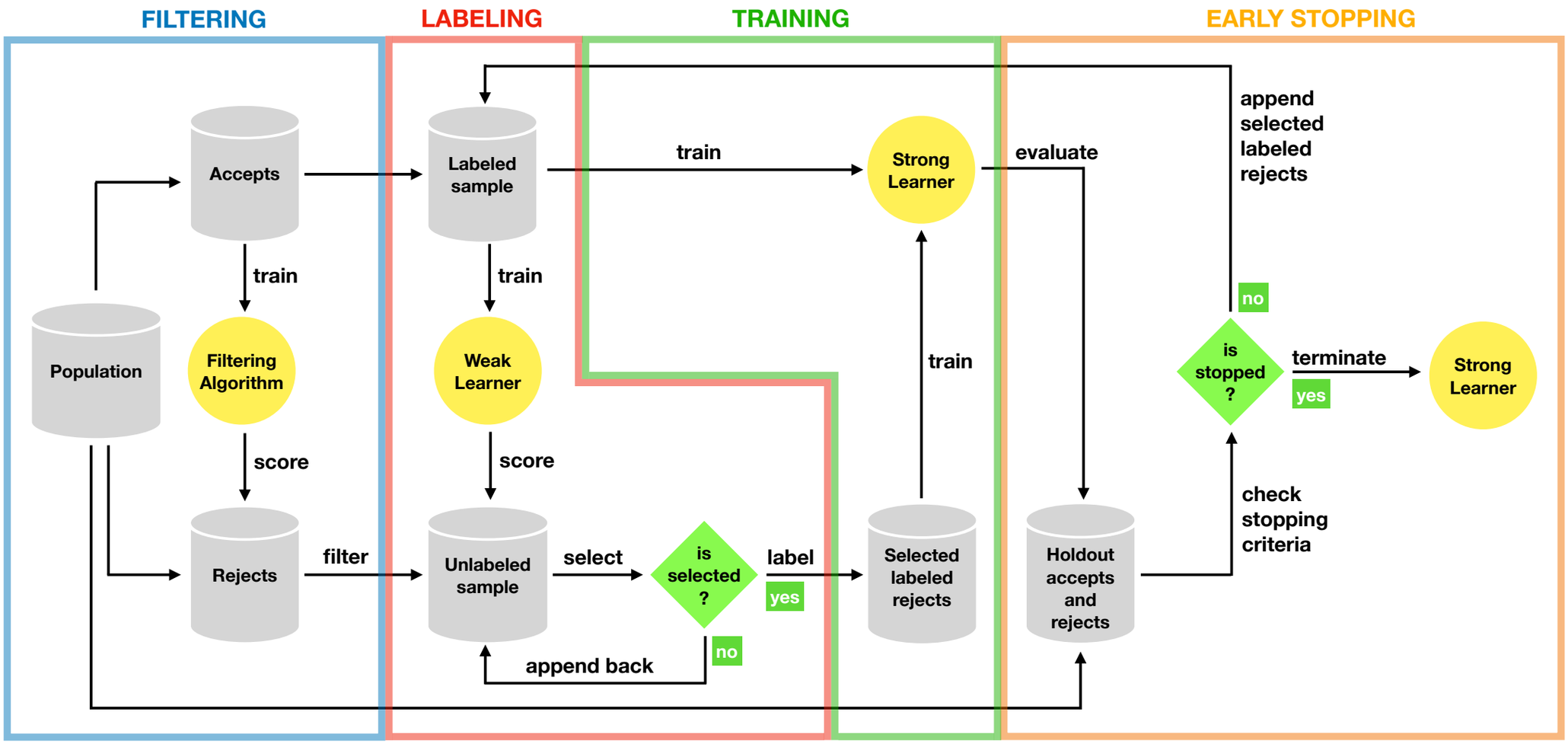
- train model ***f(x)*** on augmented training set containing:
  - labeled **accepts**
  - selected pseudo-labeled **rejects**
- use modified self-learning framework *(e.g., Triguero et al. 2013)*
  - implement techniques to reduce the risk of error propagation



Bias-Aware Self-Learning (BASL)

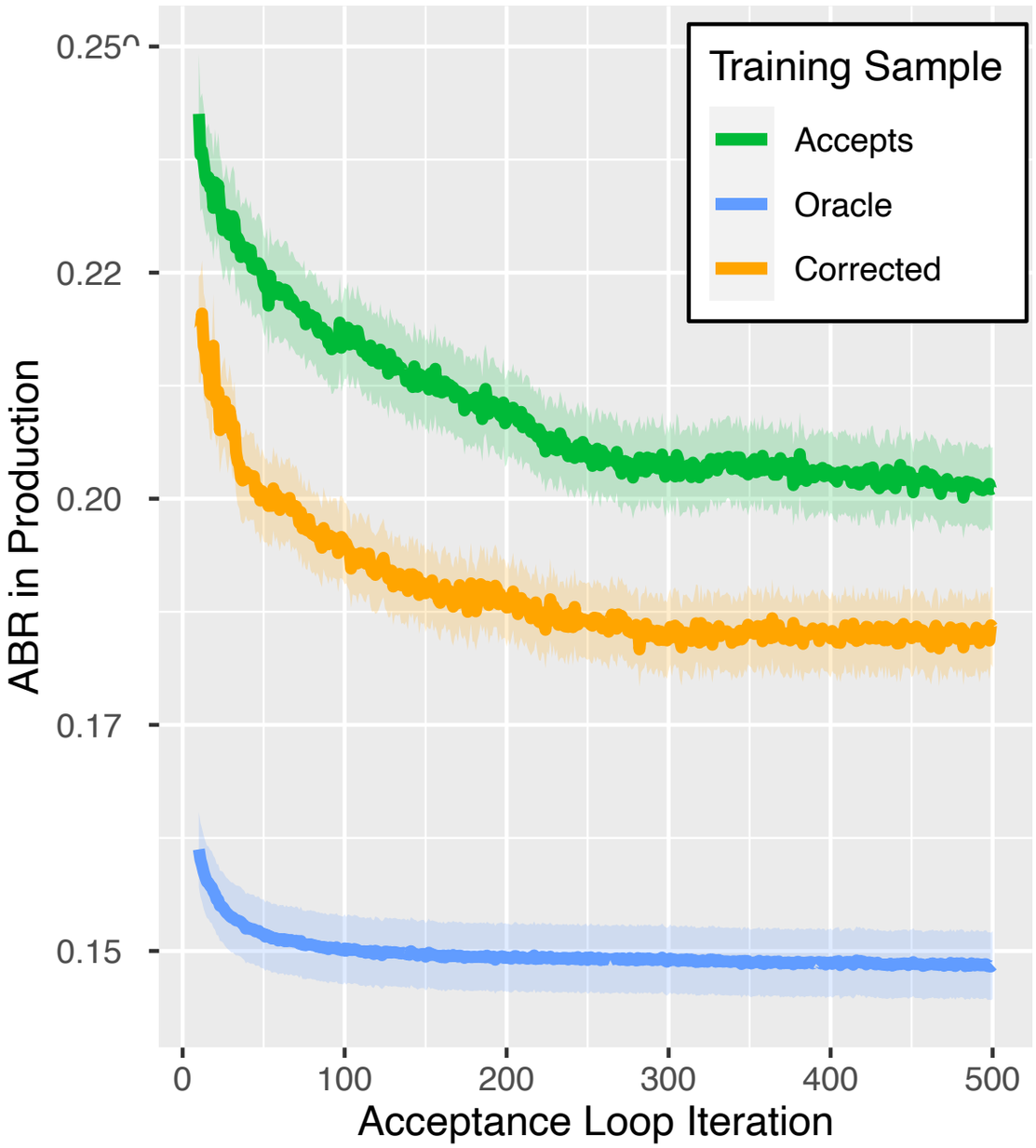
Idea:

- train model ***f(x)*** on augmented training set containing:
  - labeled **accepts**
  - selected pseudo-labeled **rejects**
- use modified self-learning framework *(e.g., Triguero et al. 2013)*
  - implement techniques to reduce the risk of error propagation



BASL: Simulation Results

Performance Dynamics



Aggregated Results

Metric	Loss due to bias	Gains from BASL
ABR	.0547	**36.86%**
BS	.0404	**45.28%**
AUC	.0589	**48.84%**
PAUC	.0488	**33.93%**

- BASL improves **model performance**
- gains are **statistically significant** at 5%

Presentation Outline

1. Background

- What is credit scoring?
- What are the business goals?

2. Problem Description

- Sampling bias illustration
- Bias impact on ML models

3. Approach

- Improving model evaluation
- Improving model training

4. Results

- Offline evaluation
- Business impact

Offline Evaluation: Experimental Setup

Data description:

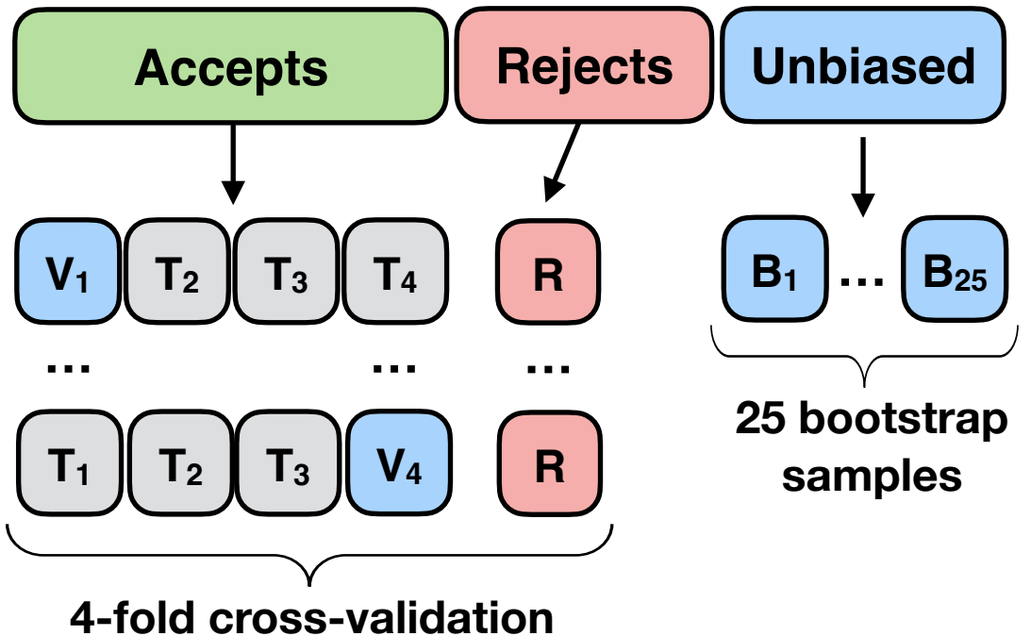
- consumer loans issued by Monedo in Spain in 2017 - 2019
- contains labeled accepts and unlabeled rejects
- includes unbiased sample: loans from randomized trial

Data summary:

	Accepts	Rejects	Unbiased
No. clients	39,579	18,047	1,967
No. features	2,410	2,410	2,410
BAD* rate	39 %	-	66 %

* missed payments for **3** consecutive months

Data organization:



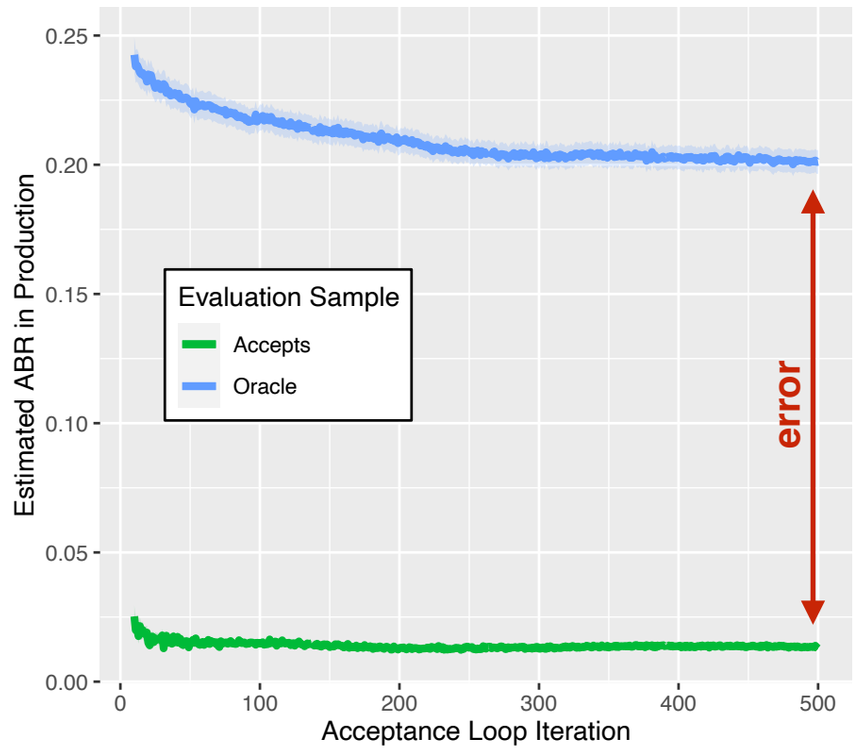
Experiment I: Improving Evaluation

Goal:

- **compare accuracy of evaluation methods**

Methodology:

- **build a scoring model and assess it on unbiased sample**
  - four evaluation metrics: ABR, BS, AUC, PAUC
- **evaluate the same model on historical data**
  - Bayesian evaluation
  - benchmarks
- **compute RMSE between the two estimates**



Experiment I: Results

Evaluation Method	ABR	BS	AUC	PAUC
Standard practice	**.0356**	**.0983**	**.1234**	**.0306**

- **ABR** = BAD rate at 30% acceptance
- **BS** = Brier Score
- **AUC** = area under the ROC curve
- **PAUC** = partial AUC at FNR in [0, 0.2]

Experiment II: Results

Evaluation Method	ABR	BS	AUC	PAUC
Standard practice	**.0356**	**.0983**	**.1234**	**.0306**
Doubly robust evaluation	.1167	-	-	.0506
Reweighting	.0315	.0826	.1277	.0348
Bayesian evaluation	**.0130**	**.0351**	**.0111**	**.0073**

- **ABR** = BAD rate at 30% acceptance
- **BS** = Brier Score
- **AUC** = area under the ROC curve
- **PAUC** = partial AUC at FNR in [0, 0.2]

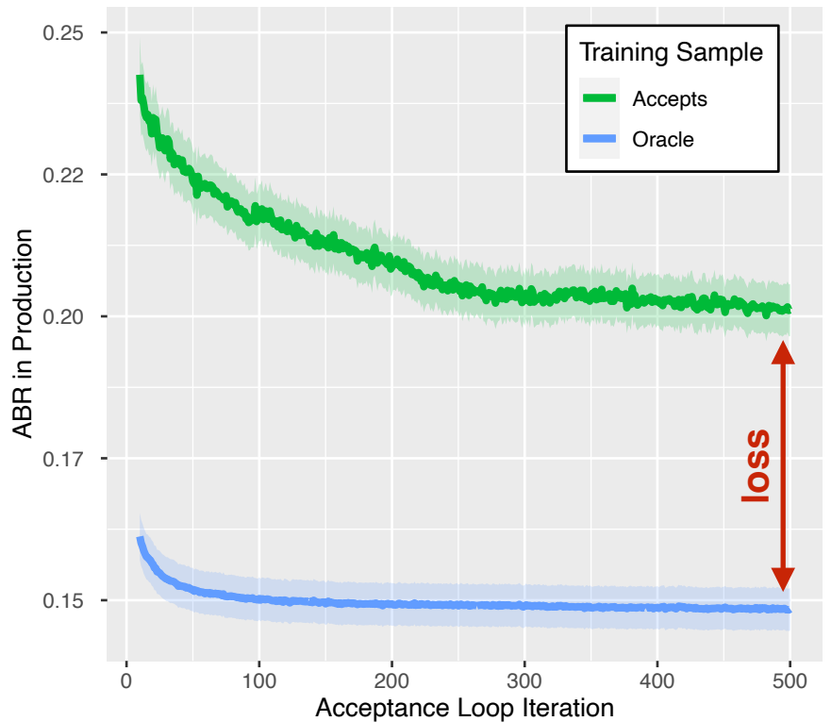
Experiment II: Improving Training

Goal:

- **compare performance of bias correction methods**

Methodology:

- **build a scoring model on accepts**
- **assess performance on unbiased sample**
  - four evaluation metrics: ABR, BS, AUC, PAUC
- **improve the model with bias correction methods**
  - BASL
  - benchmarks



Experiment II: Results

Training Method	ABR	BS	AUC	PAUC
Standard practice	**.2388**	**.1819**	**.7984**	**.6919**
Label all rejects as BAD	.3141	.2347	.6676	.6384
Bias-removing autoencoder	.3061	.2161	.7304	.6373
Heckman model	.3018	.2124	.7444	.6397
Bureau score based labels	.2514	.1860	.7978	.6783
Hard cutoff augmentation	.2458	.1830	.8033	.6790
Parceling	.2396	.1804	.8038	.6885
Reweighting	.2346	.1840	.8040	.6961
Bias-Aware Self-Learning	**.2211**	**.1761**	**.8166**	**.7075**

- **ABR** = BAD rate at 30% acceptance
- **BS** = Brier Score
- **AUC** = area under the ROC curve
- **PAUC** = partial AUC at FNR in [0, 0.2]

Business Impact: Setup

Parameters:

- **acceptance rate**
- **loan principal**
- **interest rate**

Two markets:

- **micro-loans**
- **installment loans**

	Micro loans	Installment loans
Acceptance rate α	[20%, 40%]	[10%, 20%]
Loan principal A	\$375 (SD = \$100)	\$17,100 (SD = \$1,000)
Total interest i	17.33% (SD = 1%)	10.36% (SD = 1%)

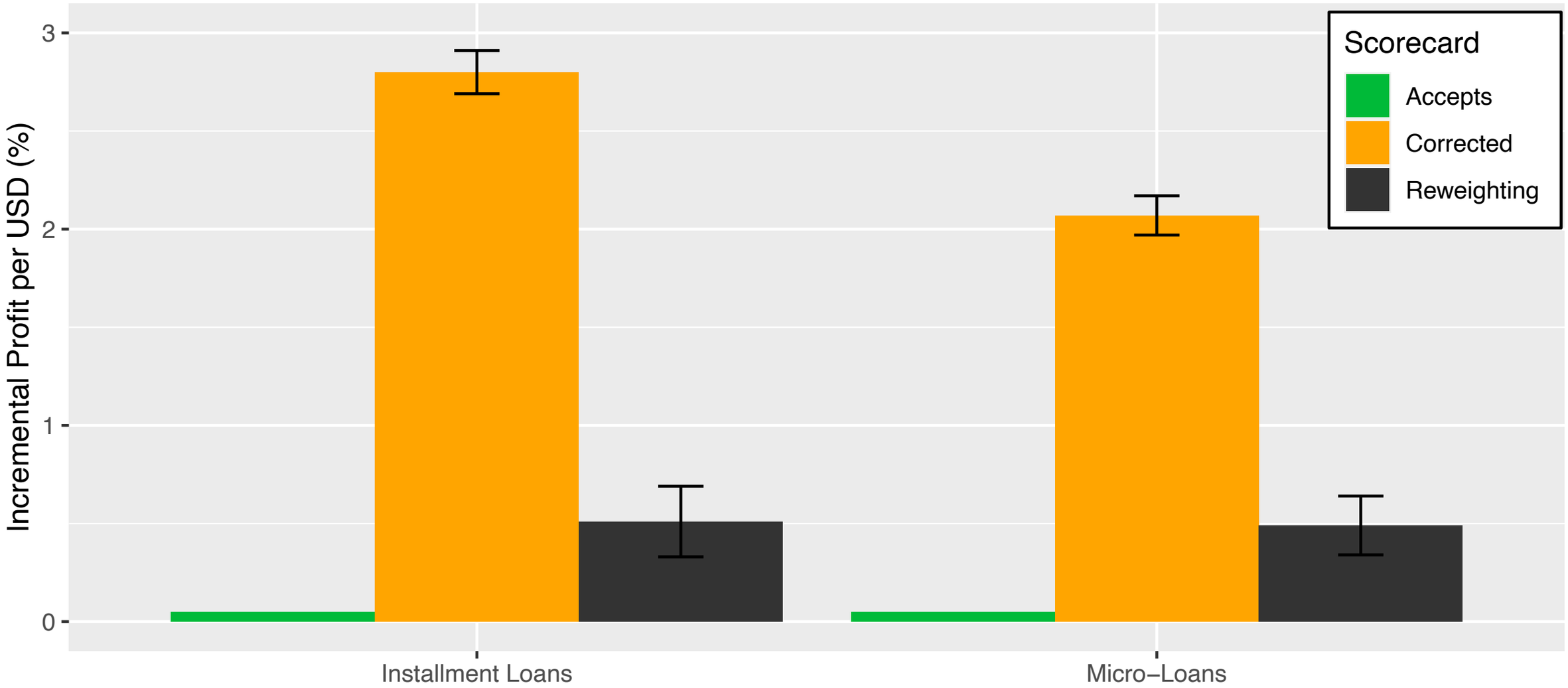
Calculations:

- **average profit per loan for each algorithm:**

$$\pi = \frac{1}{100}\sum_{j=1}^{100}\Big[\underbrace{(1-\mathrm{ABR}_j)\times A\times(1+i)}_{\textbf{GOOD clients}} - \underbrace{\mathrm{ABR}_j\times A\times(1+i)}_{\textbf{BAD clients}} - A\Big]$$

- **averaging over 100 values** (4-fold CV x 25 bootstrap samples)

Business Impact: Results

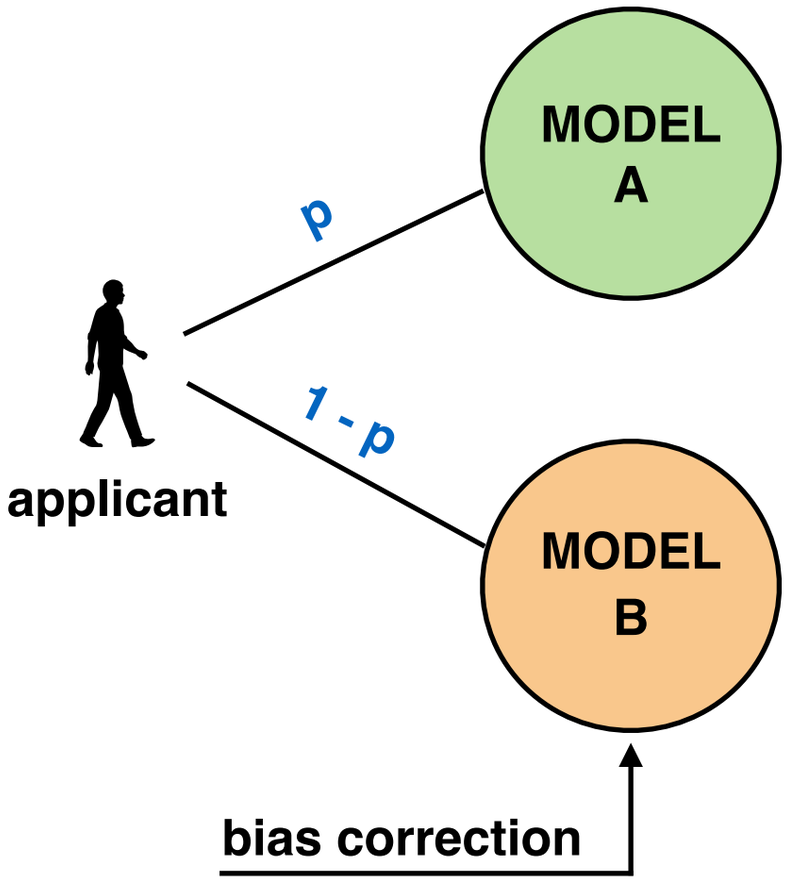


Incremental gains:

- installment loans: up to **$461.70** per loan
- micro-loans: up to **$7.78** per loan

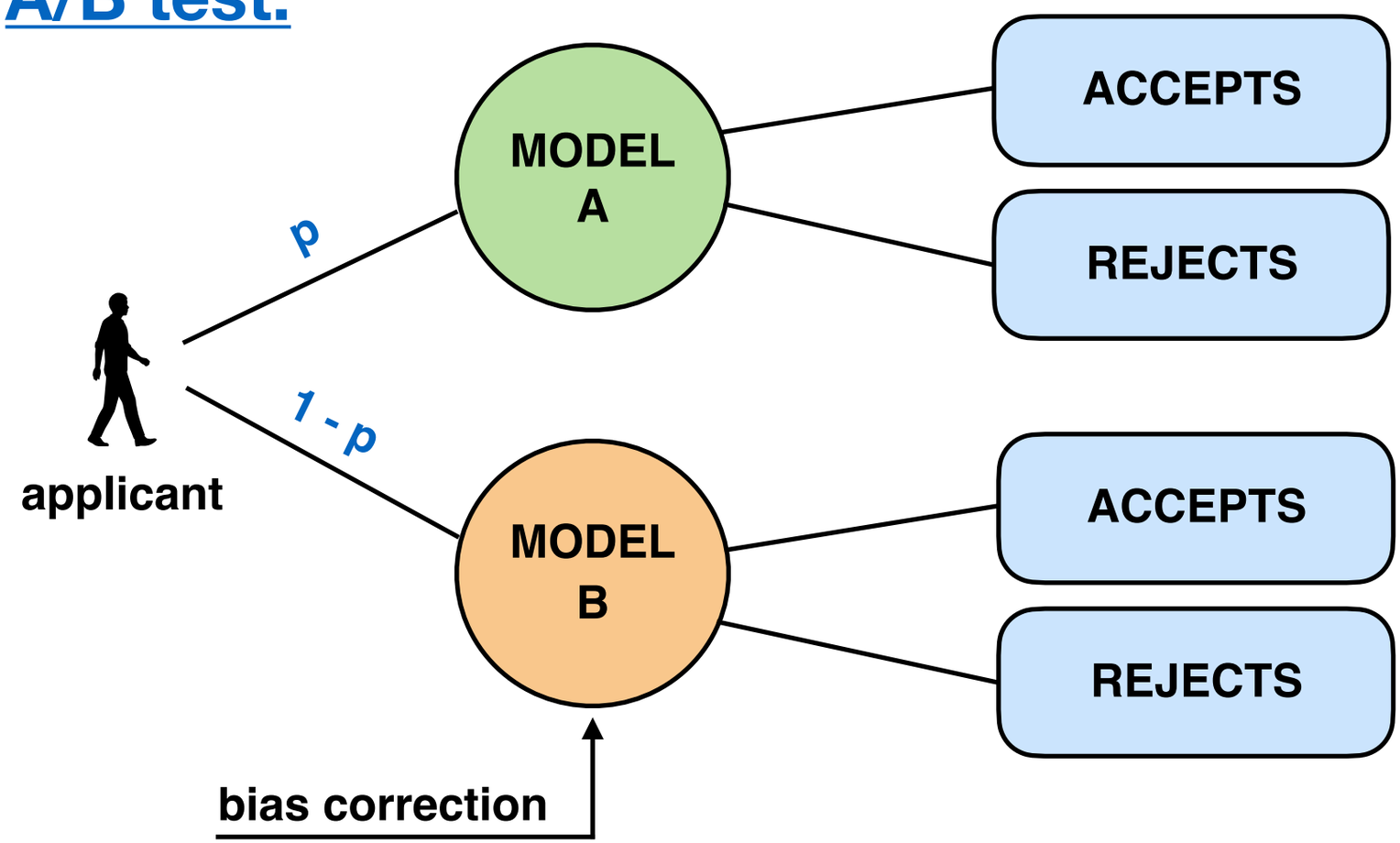
From Offline to Online

A/B test:



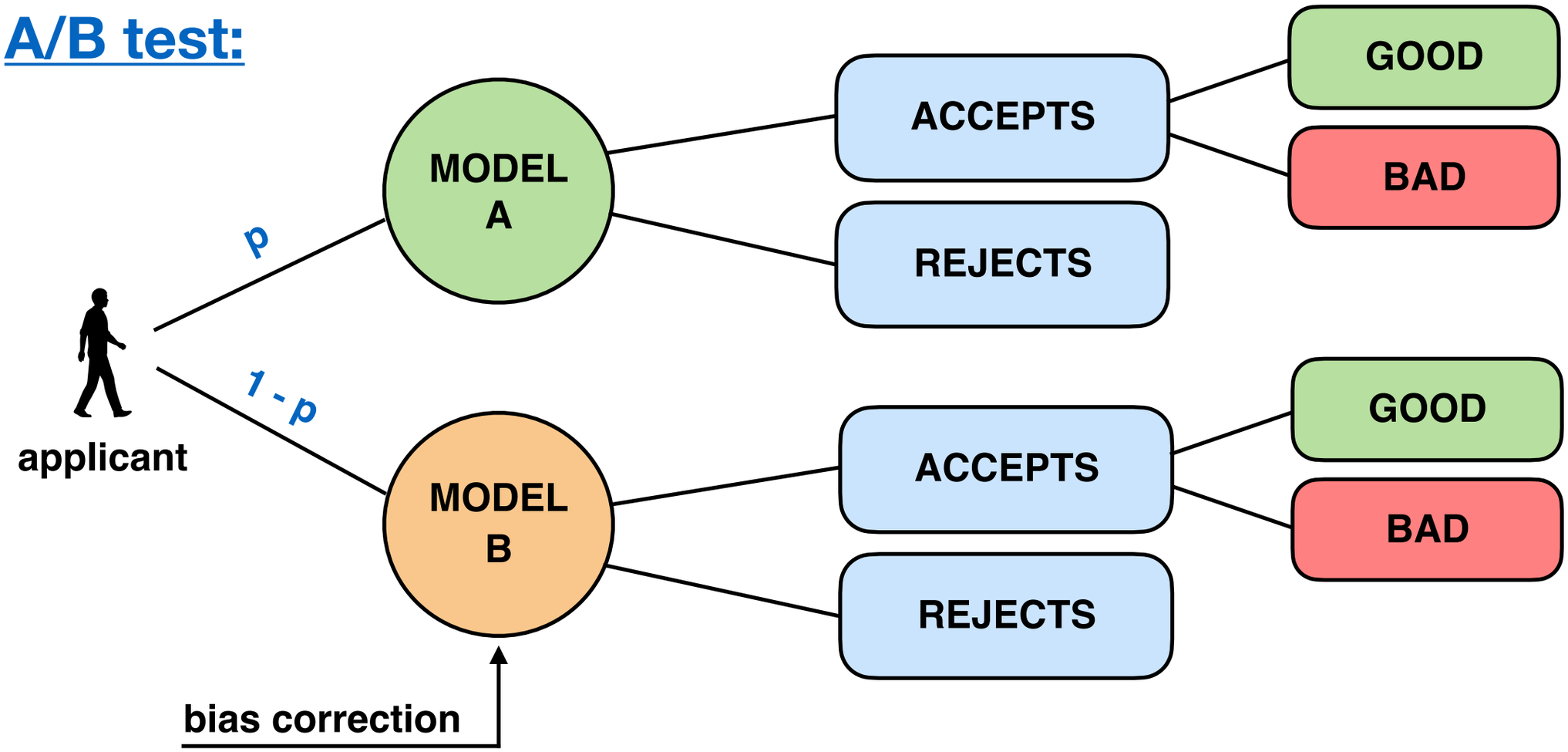
From Offline to Online

A/B test:



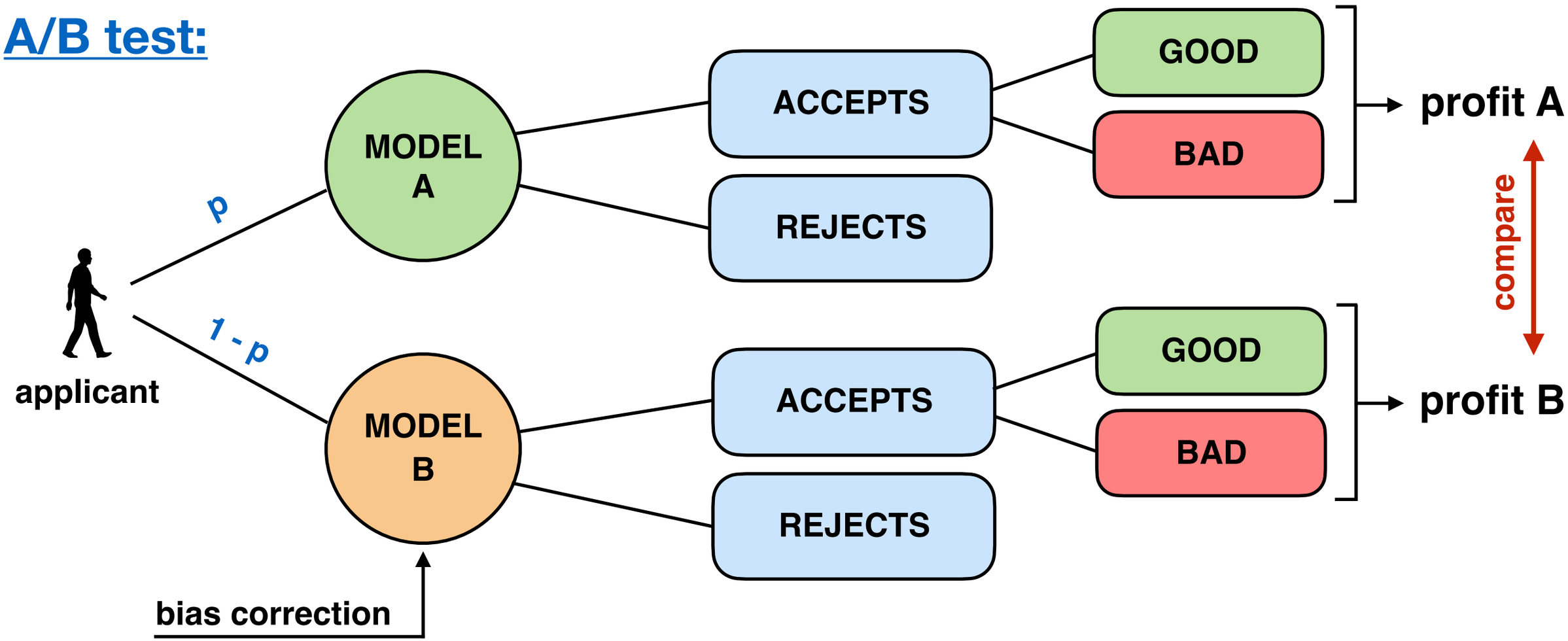
From Offline to Online

A/B test:



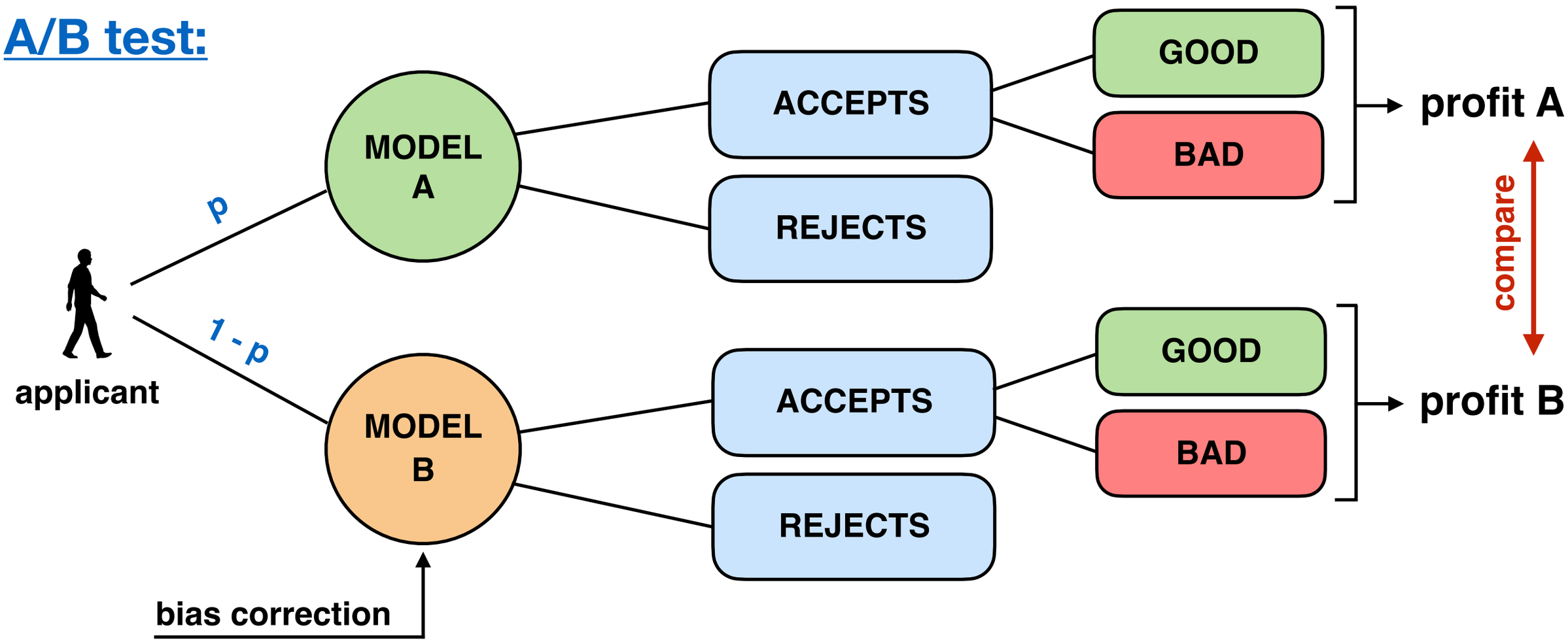
From Offline to Online

A/B test:



From Offline to Online

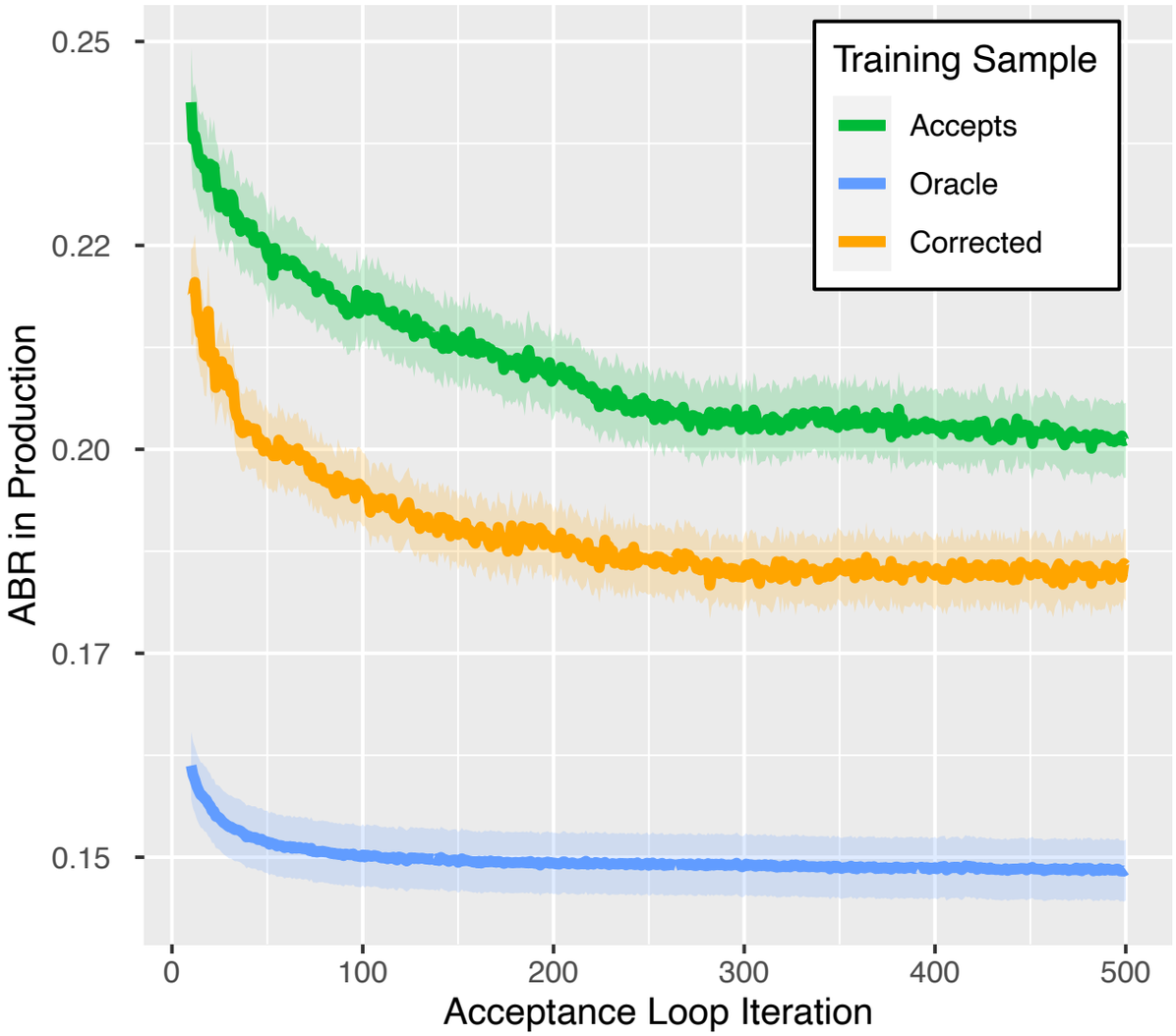
A/B test:

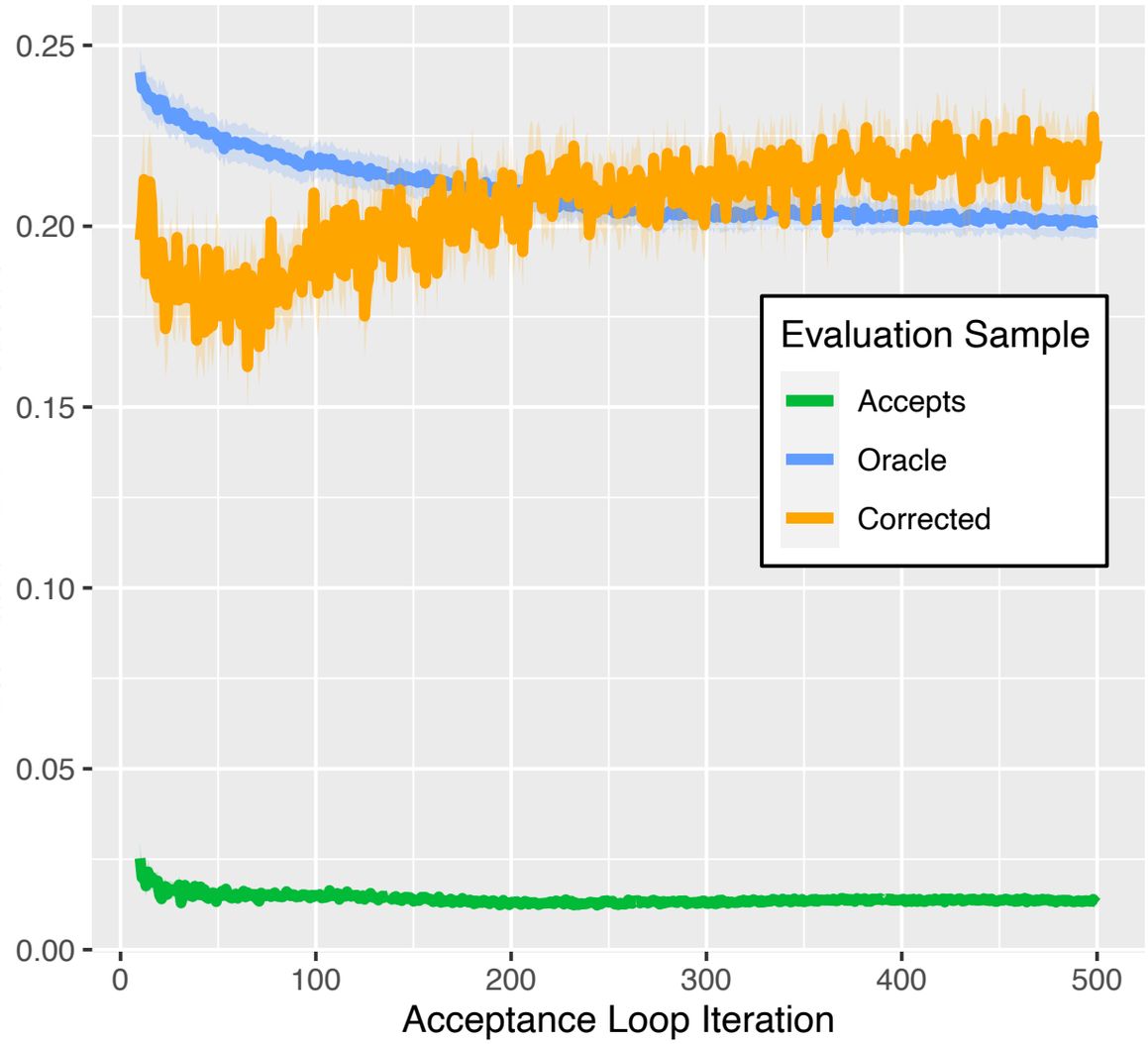


Challenges:

- long delay before observing the metrics
- regulations regarding data on rejected clients

Thank You for Your Attention!





Preprint:

Incremental gains:

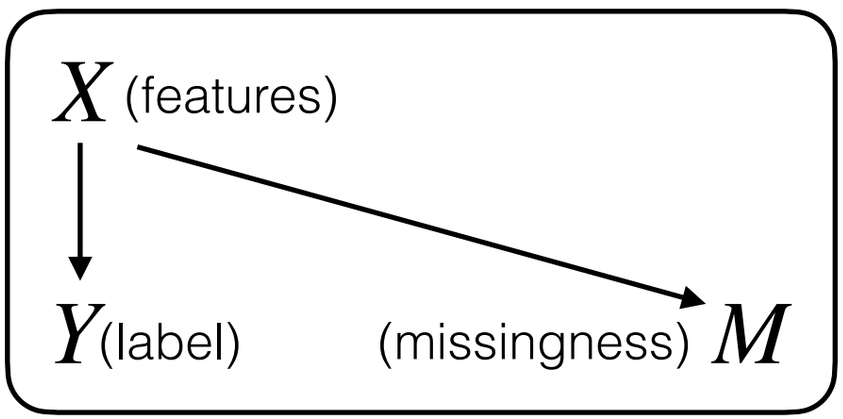
- installment loans: up to **$461.70** per loan
- micro-loans: up to **$7.78** per loan

Slides:

A1. RELATED WORK

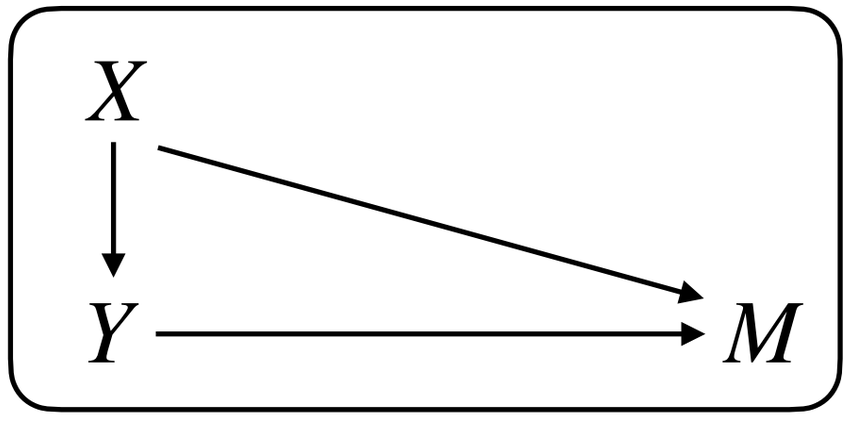
Sampling Bias as Missing Data Problem

Missing at random (MAR)



- **missingness depends on *X* but not on *Y***
  - e.g., fixed scorecard
- **affects training of some ML algorithms**
  - regression-type models (LR, NN) can extrapolate outside of the observed X space
  - tree-based models (RF, GB) cannot extrapolate *(Malistov & Trushin 2019)*
- **affects model evaluation**
  - performance estimates are misleading

Missing not at random (MNAR)



- **missingness depends on both *X* and *Y***
  - e.g., manual overwriting, omitted variables
- **affects model training**
- **affects model evaluation**

Missingness types by *(Little & Rubin 2019)*

Related Work in Different Streams

Missing data and sample selection

- **Assume labels are MNAR**
- **Sample selection methods** *(Langford et al. 2007)*
  - Heckman model *(Heckman 1976, 1979)*
  - Bivariate probit *(Meng & Schmidt 1985)*
- **Recent extensions**
  - Shadow variables *(Miao et al. 2019)*
  - Doubly robust evaluation *(Bias et al 2020)*

Train/test distribution discrepancy

- **Domain adaptation and covariate shift**
  - Representation change *(e.g., Caseiro et al. 2015)*
  - Bias-removing autoencoder *(Atan et al. 2018)*
  - Generative adversarial networks
- **Robust performance evaluation measures**
  - Modified AIC *(Shimodaira 2000)*
  - Generalization error *(Sugiyama et al. 2006)*
- **Transfer learning**

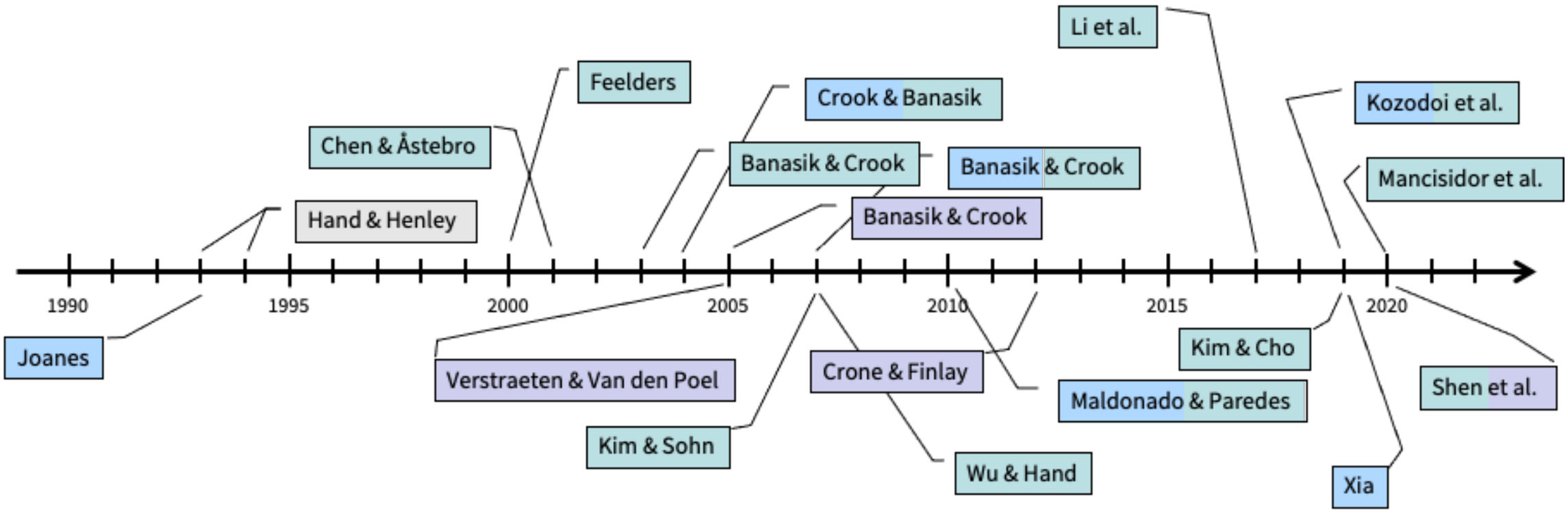
Policy evaluation and optimization

- **Contextual bandit setting** *(Langford et al. 2007)*
  - Maximizing reward across a set of actions
  - Rewards depend on the action
  - Rewards are partially observable
- **Policy evaluation and optimization**
  - Importance reweighing *(e.g., Zadrozny 2004)*
  - Doubly Robust estimator *(Dudik et al 2014)*
- **Dynamic treatment regimes in medical studies**

Training data augmentation

- **Semi-supervised learning**
  - Support vector machines *(e.g., Li et al. 2017)*
  - Self-learning *(e.g., Triguero et al. 2013)*
  - Co-training *(e.g., Chen et al. 2011)*
- **Reject inference in credit risk**
  - Hard cutoff augmentation *(Banasik et al. 2003)*
  - Parceling *(e.g., Siddiqi 2012)*
- **Active learning**

Related Work: Credit Scoring



Theory | Model-based | Labeling Rejects | Reweighting

Related Work: Credit Scoring

Reference	Implemented technique(s)	Training method(s)	Evaluation method(s)	Representative holdout sample	Profit gains	No. features
Joanes [49]	Reclassification	DA	–			3
Fogarty [33]	Multiple imputation	DA	–			10
Xia [103]	Outlier detection with isolation forest	DA	–			9
Liu et al. [66]	Ensembling classifiers and clusteres	MB	–			5, 23
Kang et al. [51]	Label spreading with oversampling	DA	–			22
Boyes et al. [16]	Heckman model variant (HM)	MB	–			42
Feelders [32]	Mixture modeling	MB	–			2
Chen et al. [21]	HM	MB	–	✓		24
Banasik et al. [9]	HM	MB	–	✓		30
Wu et al. [102]	HM	MB	–			2
Kim et al. [54]	HM	MB	–	✓		16
Chen et al. [22]	Bayesian model	MB	–		✓	40
Li et al. [59]	Semi-supervised SVM (S3VM)	MB	–			7
Marshall et al. [75]	HM	MB	–			18
Tian et al. [96]	Kernel-free fuzzy SVM	MB	–			7, 14
Xia et al. [104]	CPLE-LightGBM	MB	–			5, 17
Anderson [1]	Bayesian network	MB	–			7, 20
Kim et al. [53]	S3VM with label propagation	MB	–			17
Shen et al. [85]	Unsupervised transfer learning	MB	–			20
Banasik et al. [7]	Banded weights	RW	–	✓		30
Verstraeten et al. [98]	Resampling	RW	–	✓	✓	45
Bücker et al. [19]	Missing data based weights	RW	–			40
Crook et al. [26]	Banded weights, extrapolation	RW, DA	–	✓		30
Banasik et al. [8]	HM with banded weights	MB, RW	–	✓		30
Maldonado et al. [70]	Self-learning, S3VM	MB, DA	–			2, 20, 21
Anderson et al. [2]	HCA, Mixture modeling	DA, MB	–			12
Nguyen [78]	Parceling, HM, Banded weights	DA, MB, RW	–			9
Mancisidor et al. [72]	Bayesian model, self-learning, S3VM	DA, MB	–			7, 58
This paper	**BASL, Bayesian evaluation**	**DA**	**EF**	✓	✓	**2,410**

Abbreviations: DA = data augmentation, MB = model-based, RW = reweighting, EF = evaluation framework. "Representative holdout sample" indicates whether the study has access to a sample from the borrower's population for evaluation."Profit gains" indicates whether gains from bias correction are measured in terms of profit. "Heckman model variant" includes a linear Heckman model and a bivariate probit/logistic model with non-random sample selection.

Related Work: Bias Correction

Reference	Method	Type
Blitzer et al. [14]	Structural correspondence learning	RC
Daumé III [27]	Supervised feature augmentation	RC
Saenko et al. [83]	Supervised feature transformation	RC
Gopalan et al. [38]	Sampling geodesic flow	RC
Gong et al. [37]	Geodesic kernel flow	RC
Caseiro et al. [20]	Unsupervised feature transformation	RC
Saptal et al. [84]	Penalized feature selection	RC
Pan et al. [80]	Transfer component analysis	RC
Long et al. [68]	Transfer joint matching	RC
Sun et al. [95]	Correlation alignment	RC
Wang et al. [100]	Extreme dimension reduction	RC
Atan et al. [3]	Bias-removing autoencoder	RC
Heckman [42]	Heckman's model	MB
Meng et al. [77]	Heckman-style bivariate probit	MB
Lin et al. [60]	Modified SVM	MB
Daumé III et al. [28]	Maximum entropy genre adaptation	MB
Yang et al. [105]	Adapt-SVM	MB
Marlin et al. [73]	Multinomial mixture model	MB
Bickel et al. [13]	Kernel logistic regression	MB
Chen et al. [23]	Co-training for domain adaptation	MB, RC
Duan et al. [30]	Domain adaptation machine	MB
Long et al. [67]	Regularized least squares	MB
Liu et al. [64]	Robust bias-aware classifier	MB
Joachims et al. [48]	Modified ranking SVM	MB
Chen et al. [24]	Robust bias-aware regression	MB
Liu et al. [63]	Modified bias-aware classifier	MB
Kügelgen et al. [56]	Semi-generative model	MB

Reference	Method	Type
Heckman [42]	Heckman's model	MB
Meng et al. [77]	Heckman-style bivariate probit	MB
Lin et al. [60]	Modified SVM	MB
Daumé III et al. [28]	Maximum entropy genre adaptation	MB
Yang et al. [105]	Adapt-SVM	MB
Marlin et al. [73]	Multinomial mixture model	MB
Bickel et al. [13]	Kernel logistic regression	MB
Chen et al. [23]	Co-training for domain adaptation	MB, RC
Duan et al. [30]	Domain adaptation machine	MB
Long et al. [67]	Regularized least squares	MB
Liu et al. [64]	Robust bias-aware classifier	MB
Joachims et al. [48]	Modified ranking SVM	MB
Chen et al. [24]	Robust bias-aware regression	MB
Liu et al. [63]	Modified bias-aware classifier	MB
Kügelgen et al. [56]	Semi-generative model	MB
Rosenbaum et al. [81]	Model-based probabilities	RW
Shimodaira [86]	Distribution density ratios	RW
Zadrozny [106]	Selection probabilities are known	RW
Huang et al. [45]	Kernel mean matching	RW
Cortes et al. [25]	Cluster-based frequencies	RW
Sugiyama et al. [93]	Kullback-Leibler weights	RW
Kanamori et al. [50]	Least-squares importance fitting	RW
Loog [69]	Nearest-neighbor based weights	RW
Gong et al. [36]	Focusing on cases similar to test data	RW
Shimodaira [86]	Modified AIC	EM
Sugiyama et al. [94]	Subspace information criterion	EM
Sugiyama et al. [92]	Generalization error	EM
Sugiyama et al. [91]	Importance-weighted validation	EF
Bruzzone et al. [18]	Circular evaluation strategy	EF

How to Use Rejects?

Assigning labels:

- **Traditional:** hard cutoff augmentation, parcelling
- **Semi-supervised:** self-training, SVMs
- **Label noise:** CV-based
- **Evolutionary algorithms:** GA with labels as genes

Filtering rejects:

- **Instance selection:** genetic algorithms
- **Novelty detection:** isolation forest
- **Active learning:** error minimization, uncertainty sampling

Extracting information:

- **Autoencoders**

Bias-Removing Autoencoder

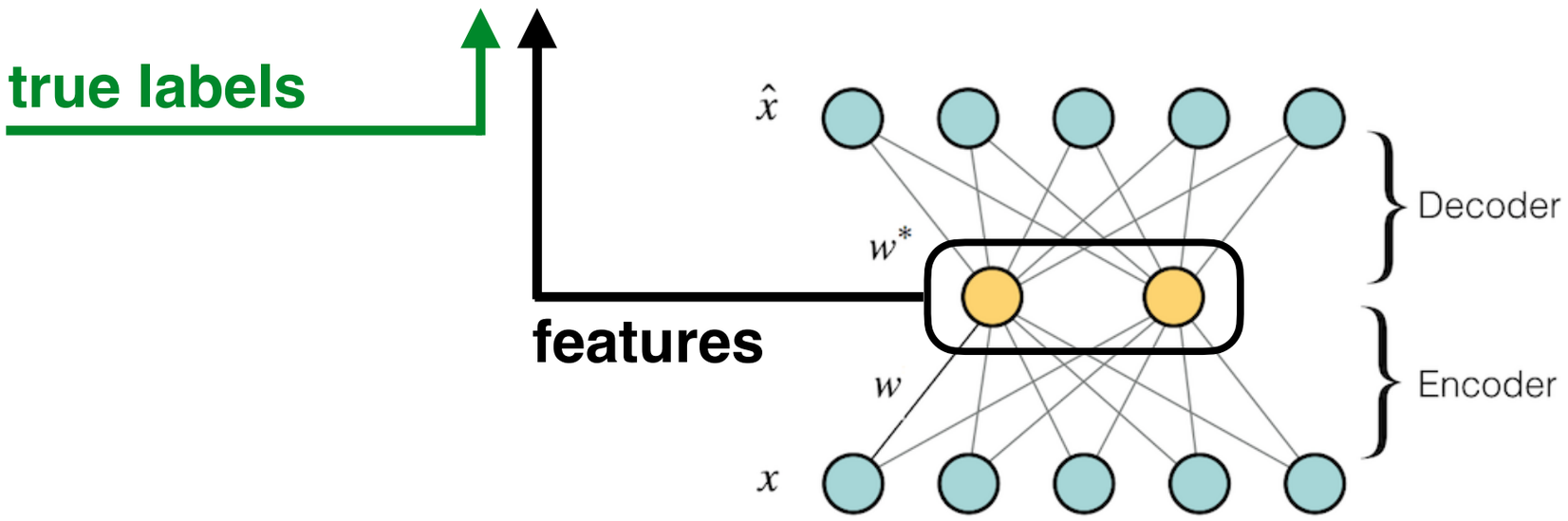
Idea:

- train Autoencoder on **accepts** + **rejects**
  - add distribution mismatch penalty to the loss function
- extract features from the bottleneck layer
- append new features to **accepts** and train a new model

TRAINING SET

ACCEPTS | **REJECTS**

true labels



Reject Inference Techniques

Label as BAD:

- Label all rejects as **BAD** risks
- Augment the known data with the labeled rejects
- Retrain the scoring model on the full data

Bureau score based inference:

- Use bureau scores to assign labels to **rejects**
- e.g., label **“A”** examples as **GOOD**, **“B”** and below - as **BAD**
- Augment the known data with the labeled rejects
- Retrain the scoring model on the full data
- Can use other informative attributes (e.g., previous loan delinquency)

Reject Inference Techniques

Hard cutoff augmentation:

- Train a scoring model based on **accepts**
- Predict default probabilities for **rejects** using this model
- Assign labels based on a certain threshold
- Augment the known data with the labeled rejects
- Retrain the scoring model on the full data

Parceling:

- Split **rejects** into groups based on the model score
- Assign labels within groups proportionally to the expected bad rate
- **BAD** ratio for rejects is usually assumed to be higher

Key References

■Banasik, J., Crook, J., & Thomas, L. C. (2003). Sample selection bias in credit scoring models. Journal of the Operational Research Society, 54(8), 822-832.

■Banasik, J., & Crook, J. N. (2005). Credit scoring, augmentation and lean models. *Journal of the Operational Research Society,* 56(9), 1072-1081.

■Banasik, J., & Crook, J. (2007). Reject inference, augmentation, and sample selection. European Journal of Operational Research, 183(3), 1582-1594.

■Bia, M., Huber, M., & Lafférs, L. (2020). Double machine learning for sample selection models. *ArXiv preprint*, arXiv:2012.00745v2.

■Chen T, Guestrin C. (2016). XGBoost: A Scalable Tree Boosting System. In B. Krishnapuram, M. Shah, A. J. Smola, C. Aggarwal, D. Shen & R. Rastogi (Eds.). *Proc. of the 22nd ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining (KDD'16)*, ACM, pp. 785-794.

■Chen, G. G., & Astebro, T. (2001). The economic value of reject inference in credit scoring. *Department of Management Science, University of Waterloo*.

■Crone, S. F., & Finlay, S. (2012). Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting,* 28(1), 224-238.

■Crook, J., & Banasik, J. (2004). Does reject inference really improve the performance of application scoring models? Journal of Banking & Finance, 28(4), 857-874.

■Dudik, M., Erhan, D., Langford, J., & Li, L. (2014). Doubly robust policy evaluation and optimization. Statistical Science, 29(4), 485-511.

■Feelders, A. (2000). Credit scoring and reject inference with mixture models. *Intelligent Systems in Accounting, Finance and Management Decision*, 9(1), 1-8.

■Hand, D. J., & Henley, W. E. (1993). Can reject inference ever work? IMA Journal of Management Mathematics, 5(1), 45-55.

■Huber, M. (2014). Treatment evaluation in the presence of sample selection. *Econometric Reviews*, 33(8), 869-905.

■Joanes, D. N. (1993). Reject inference applied to logistic regression for credit scoring. *IMA Journal of Management Mathematics,* 5(1), 35-43.

■Kim, Y., & Sohn, S. Y. (2007). Technology scoring model considering rejected applicants and effect of reject inference. *Journal of the Operational Research Society,* 58(10), 1341-1347.

■Kozodoi, N., Katsas, P., Lessmann, S., Moreira-Matias, L., & Papakonstantinou, K. (2019). *Shallow Self-Learning for Reject Inference in Credit Scoring*. Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD'2019), Springer.

Key References

■Langford, J., & Zhang, T. (2007). The Epoch-Greedy algorithm for contextual multi-armed bandits. Proc. of the 20th Intern. Conf. on Neural Information Processing Systems (NIPS'07), Vancouver, British Columbia, Canada.

■Li, Z., Tian, Y., Li, K., Zhou, F., & Yang, W. (2017). Reject inference in credit scoring using Semi-supervised Support Vector Machines. *Expert Systems with Applications*, 74, 105-114.

■Maldonado, S., & Paredes, G. (2010). *A Semi-supervised Approach for Reject Inference in Credit Scoring Using SVMs*. In P. Perner (Ed.). Advances in Data Mining. Applications and Theoretical Aspects. Proc. of the 10th Industrial Conf. on Data Mining, Springer, pp. 558-571.

■Mancisidor, R. A., Kampffmeyer, M., Aas, K., & Jenssen, R. (2020). Deep generative models for reject inference in credit scoring. *Knowledge-Based Systems*, 105758.

■Miao, W., & Tchetgen Tchetgen, E. J. (2016). On varieties of doubly robust estimators under missingness not at random with a shadow variable. *Biometrika*, 103(2), 475-482.

■Miao, W., Liu, L., Tchetgen, E. T., & Geng, Z. (2019). Identification, doubly robust estimation, and semiparametric efficiency theory of nonignorable missing data with a shadow variable. *ArXiv preprint*, arXiv:1509.02556v3.

■Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: Online Appendix. *European Journal of Operational Research,* 247, 124-136.

■Shen, F., Zhao, X., & Kou, G. (2020). Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory. *Decision Support Systems,* 137, 113366.

■Su, Y., Dimakopoulou, M., Krishnamurthy, A., & Dudik, M. (2020). Doubly robust off-policy evaluation with shrinkage. In D. Hal, III & S. Aarti (Eds.). *Proc. of the 37th Intern. Conf. on Machine Learning (ICML)*, PMLR, pp. 9167-9176.

■Verstraeten, G., & Poel, D. V. d. (2005). The impact of sample bias on consumer credit scoring performance and profitability. *Journal of the Operational Research Society,* 56, 981-992.

■Wu, I. D., & Hand, D. J. (2007). Handling selection bias when choosing actions in retail credit applications. *European Journal of Operational Research,* 183(3), 1560-1568.

■Xia, Y. (2019). A Novel Reject Inference Model Using Outlier Detection and Gradient Boosting Technique in Peer-to-Peer Lending. *IEEE Access,* 7, 92893-92907.

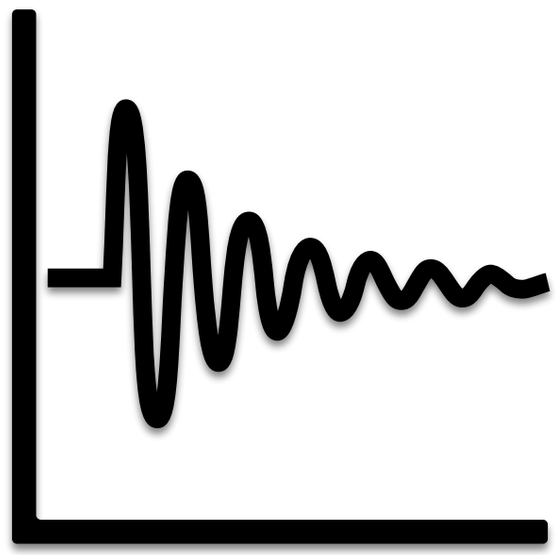
A2. CHALLENGES

Dataset Shift and Sampling Bias

- **distribution discrepancy is also affected by dataset shift**
  - complicates the correction of sampling bias between **accepts**/**rejects**
  - long delay between accepting an applicant and learning their label

- **covariate shift**
  - change in the feature distribution between train and test data
  - e.g., changes in the acceptance policy or marketing strategy

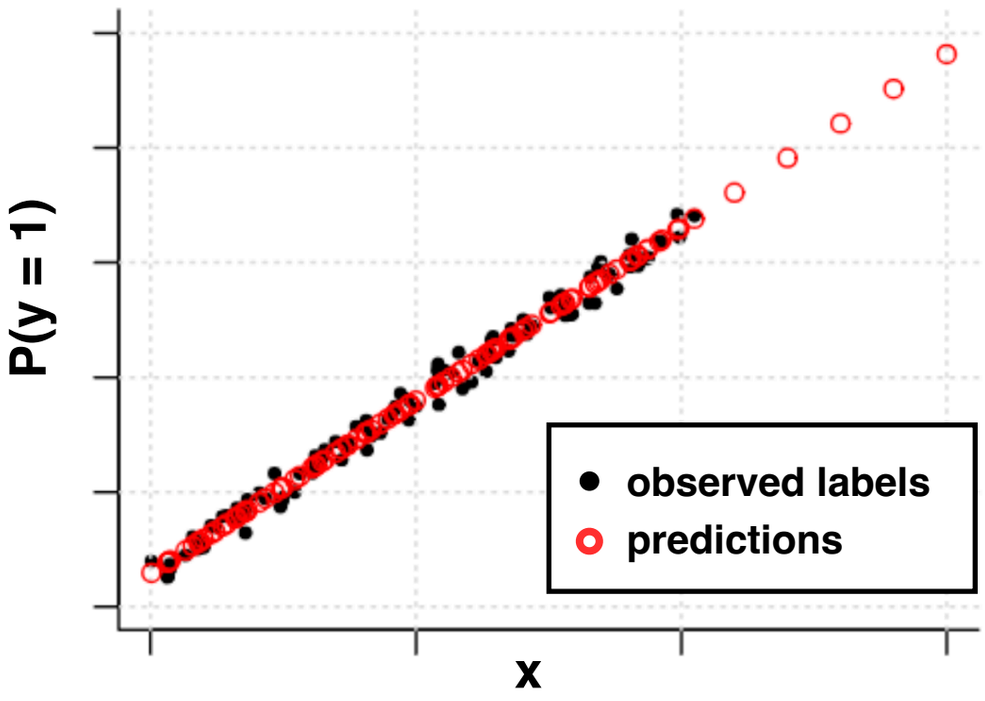
- **concept shift**
  - change in the functional feature-target relationship
  - e.g., changes in the business cycle



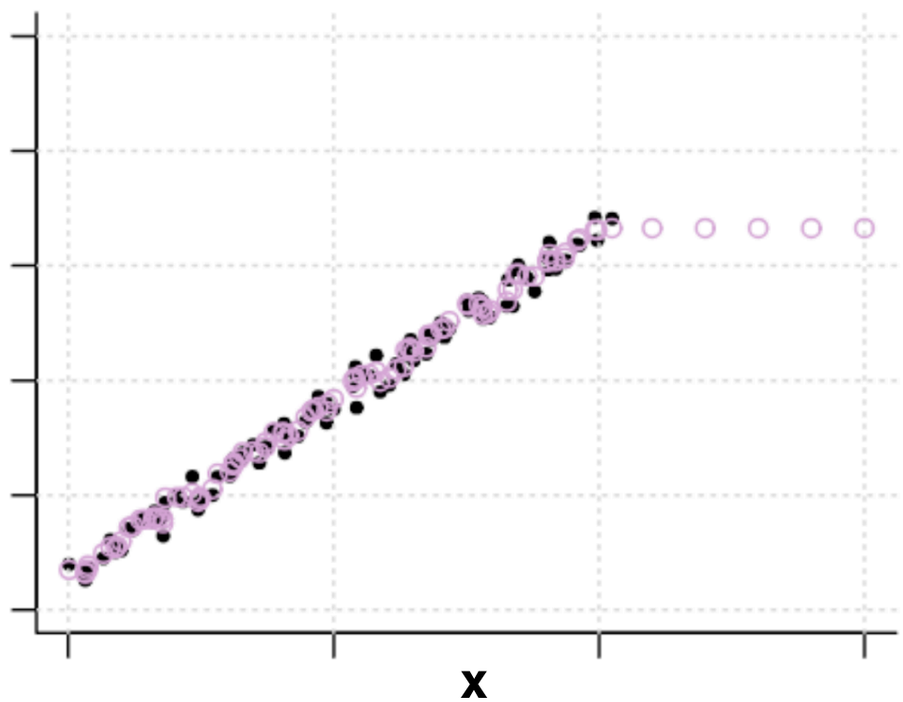
Sampling Bias in Different Environments

- **magnitude of sampling bias depends on many factors**
- **e.g., lower approval rates => stronger bias**
  - low acceptance increases difference between **accepts** and population
  - can make it too difficult for bias correction to work given a sparse sample
- **classifiers have different extrapolation ability and bias sensitivity**

Linear model
(e.g., Logistic Regression)

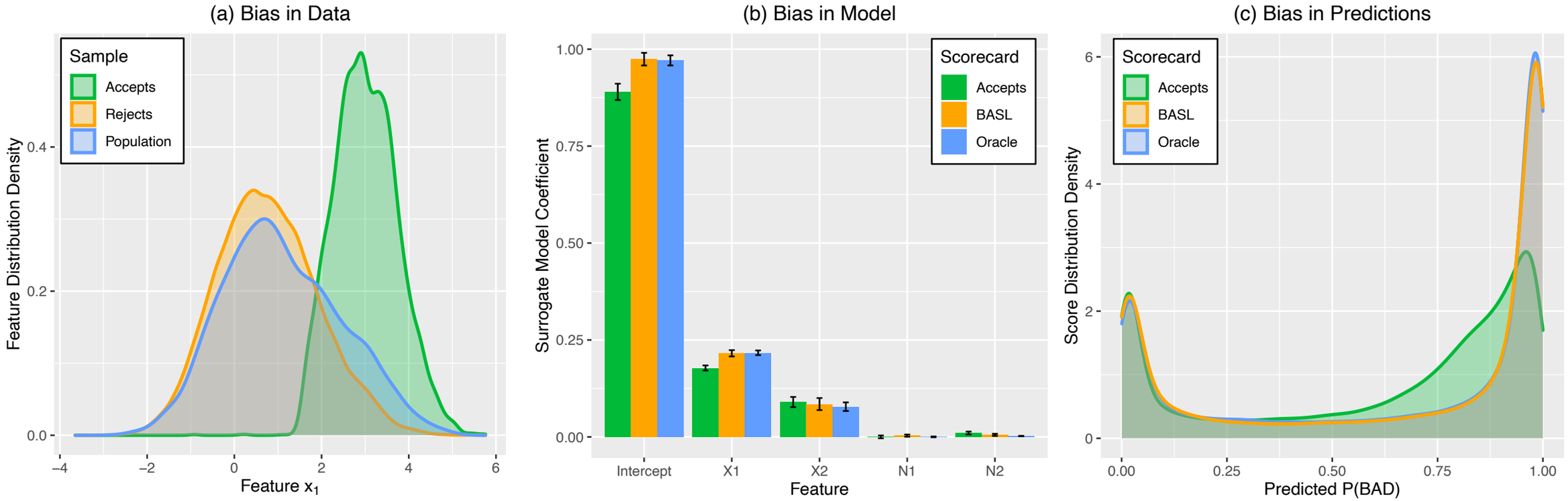


Tree-based model
(e.g., Gradient Boosting)



Some Further Challenges

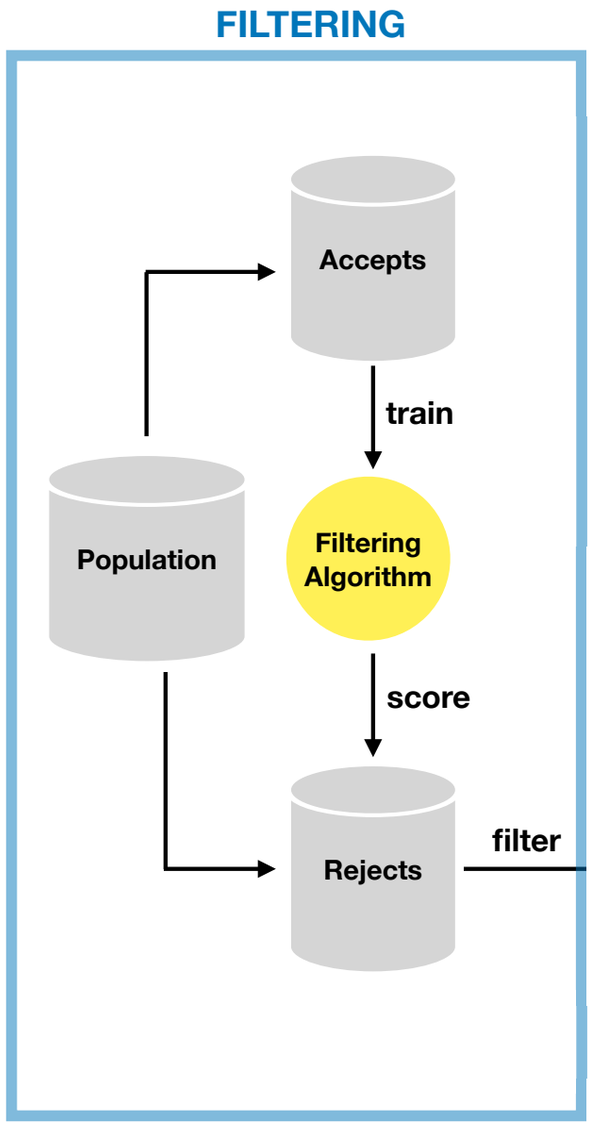
- **regulation-related challenges**
  - keeping data on **rejected applicants** might not be feasible
  - need to create synthetic samples similar to real **rejects**
- **bias illustration in ML models**
  - detecting bias in non-parametric models is **not straightforward**
  - need to illustrate bias through the lens of performance / model predictions



A3. TRAINING

BASL: Framework Stages [1/4]

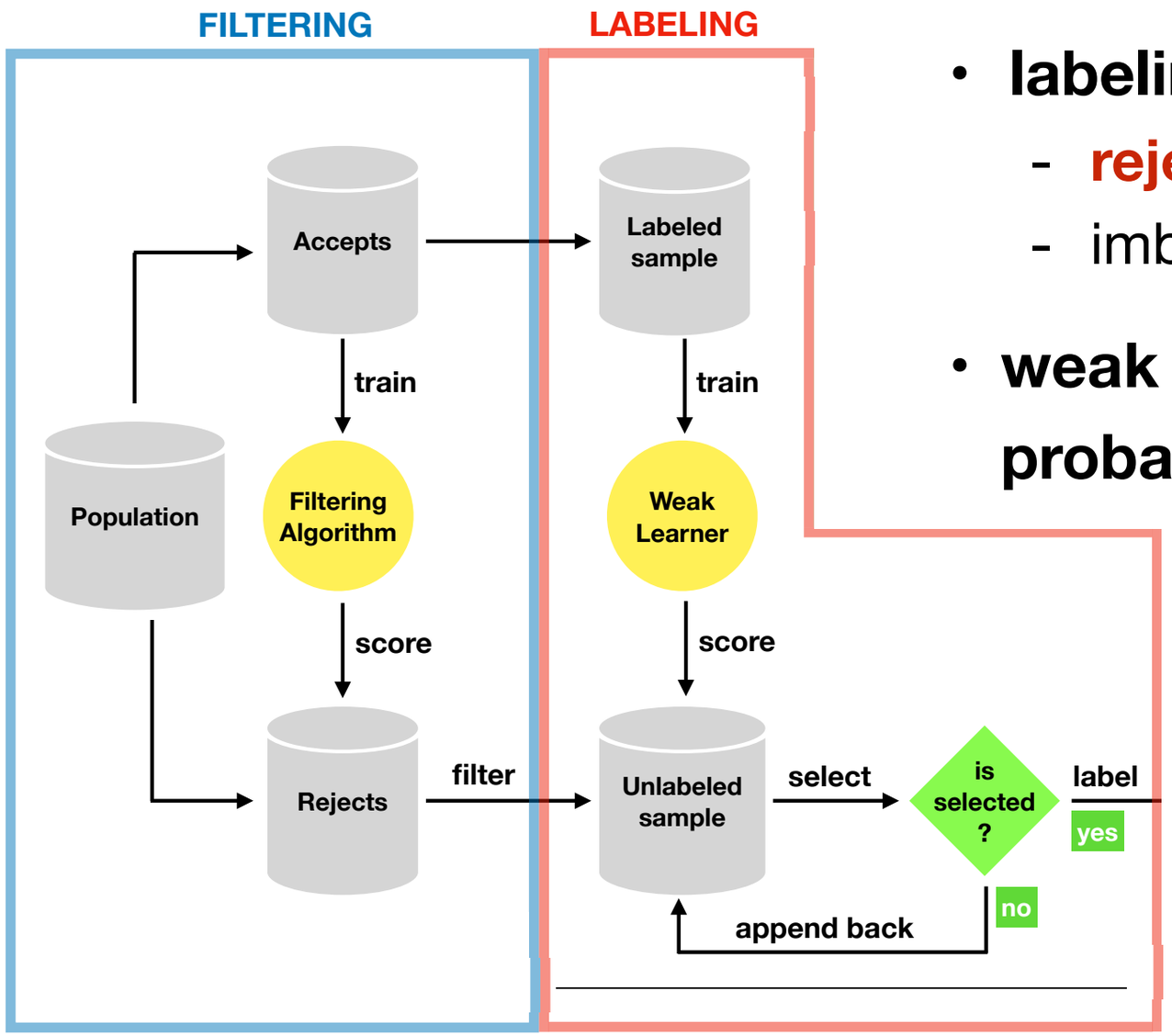
- **augmenting training data with selected rejects to reduce bias**
  - inspired by self-learning framework *(e.g., Triguero et al. 2013)*
  - model-agnostic nature
- **implement multiple techniques to reduce the risk of error propagation**



- **removing rejects whose distribution is most different from the observed accepts**
  - predictions for such cases may not be reliable
  - reduces risks of error propagation
- **estimating similarity using isolation forest**
  - novelty detection algorithm *(Liu et al. 2008)*
  - fitted on **accepts**

BASL: Framework Stages [2/4]

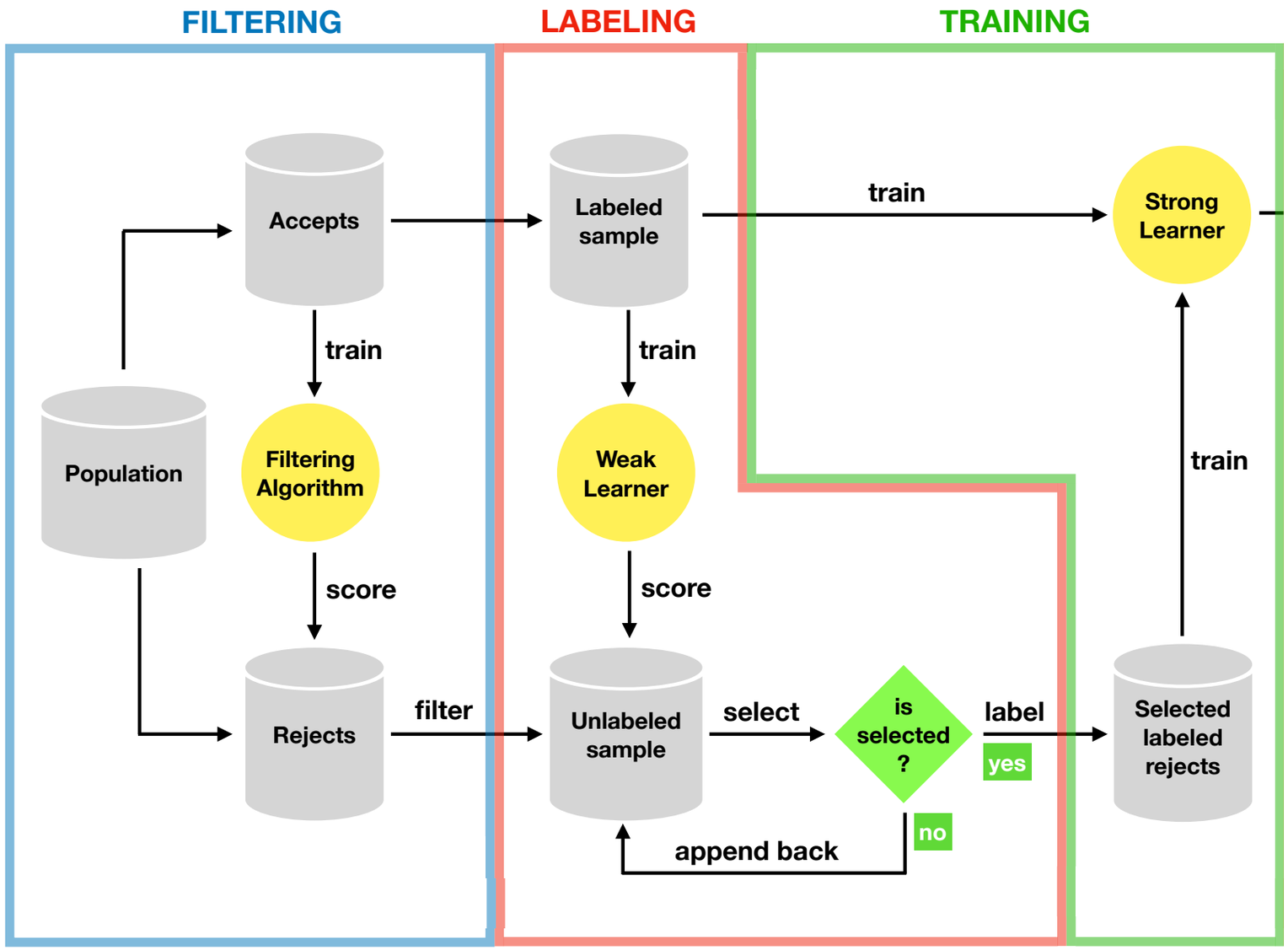
- **augmenting training data with selected rejects to reduce bias**
  - inspired by self-learning framework *(e.g., Triguero et al. 2013)*
  - model-agnostic nature
- **implement multiple techniques to reduce the risk of error propagation**



- **labeling and appending selected rejects**
  - **rejects** classified with high confidence
  - imbalance multiplier to add more **BAD** loans
- **weak learner produces well-calibrated probabilities** *(Niculescu-Mizil et al., 2005)*
  - L1-regularized logistic regression
  - trained on **accepts**
- **extrapolating outside observed *X***

BASL: Framework Stages [3/4]

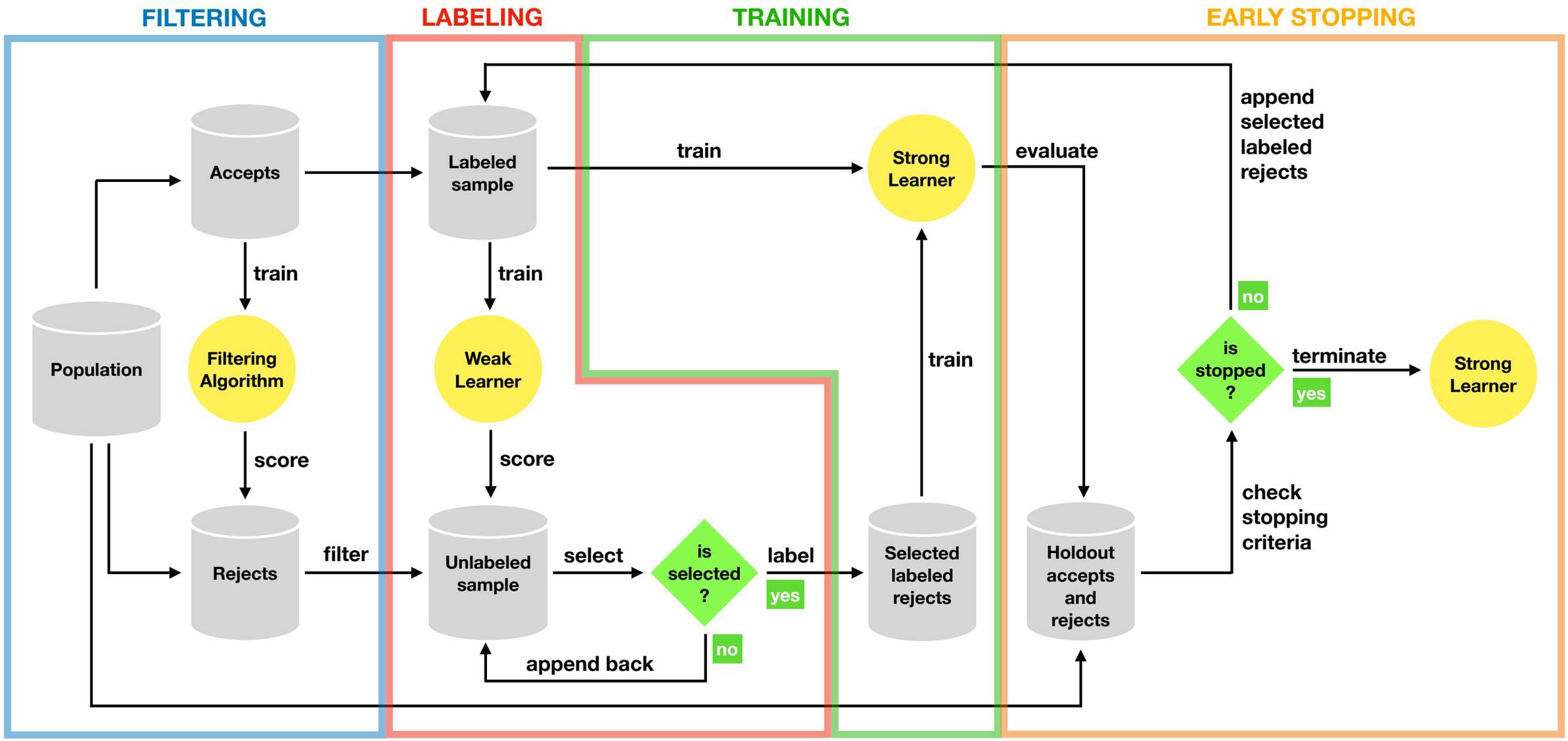
- **augmenting training data with selected rejects to reduce bias**
  - inspired by self-learning framework *(e.g., Triguero et al. 2013)*
  - model-agnostic nature
- **implement multiple techniques to reduce the risk of error propagation**



- **train on augmented data**
  - **accepts**
  - **labeled rejects**
- **use strong learner**
  - e.g., gradient boosting *(Chen et al., 2015)*

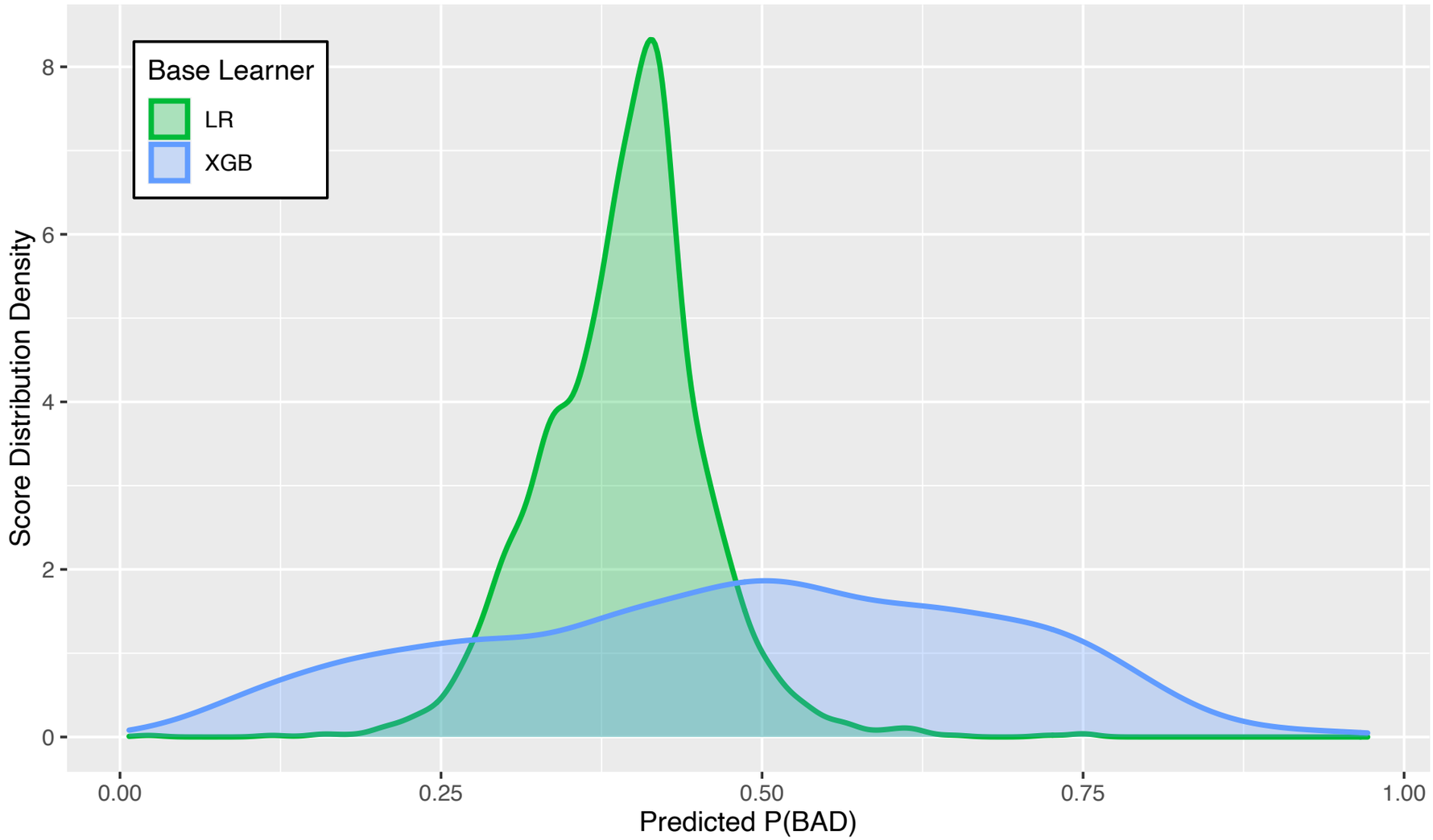
BASL: Framework Stages [4/4]

- **augmenting training data with selected rejects to reduce bias**
  - inspired by self-learning framework *(e.g., Triguero et al. 2013)*
  - model-agnostic nature
- **implement multiple techniques to reduce the risk of error propagation**



Comparing Prediction Density

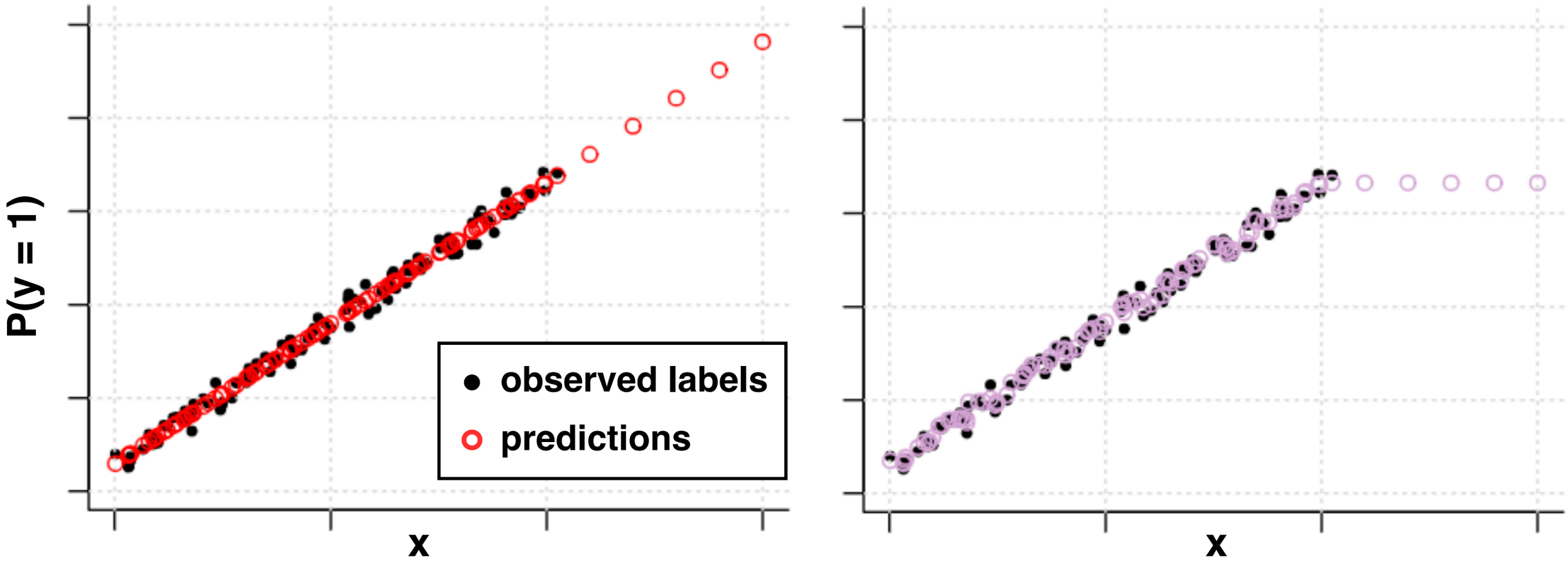
- **strong learners (e.g. XG) tend to produce overconfident scores on the extremes of the score distribution** *(Niculescu-Mizil et al., 2005)*
- **weak learners (e.g., LR) output well-calibrated probabilistic predictions**



Extrapolation: Regression vs Tree Methods

Regression-type model
(e.g., Logistic Regression)

Tree-based model
(e.g., Gradient Boosting)



- **logisitc regression** can extrapolate outside of the observed feature space
- **tree-based methods** can not extrapolate and produce constant predictions

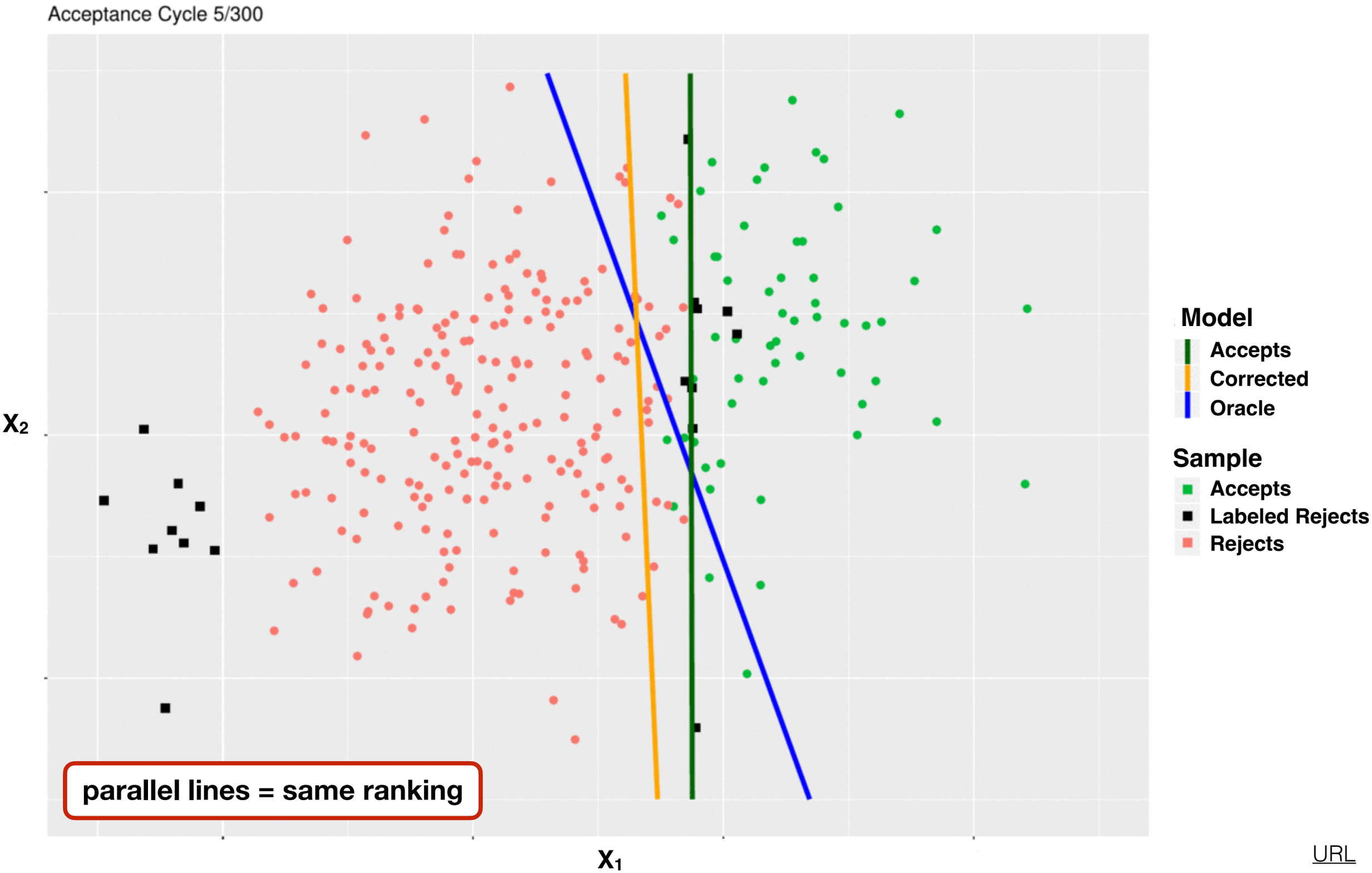
Source: Hengl, T. et al (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ*, *6*, e5518.

BASL: Ablation Study

Framework extension	AUC	BS	PAUC	ABR	Rank
Traditional self-learning	.8059 (.0010)	.1804 (.0004)	.6868 (.0011)	.2387 (.0020)	4.80
Filter rejects using isolation forest	.8054 (.0011)	.1790 (.0004)	.6981 (.0013)	.2312 (.0022)	3.93
Label rejects with a weak learner	.8134 (.0006)	.1774 (.0002)	.6992 (.0009)	.2294 (.0011)	3.60
Introduce the imbalance multiplier	.8133 (.0006)	.1796 (.0002)	.7026 (.0010)	.2238 (.0012)	3.48
Sampling rejects at each iteration	.8157 (.0006)	.1765 (.0002)	.7035 (.0010)	.2254 (.0013)	2.85
Performance-based early stopping	**.8166** (.0007)	**.1761** (.0003)	**.7075** (.0011)	**.2211** (.0012)	**2.34**

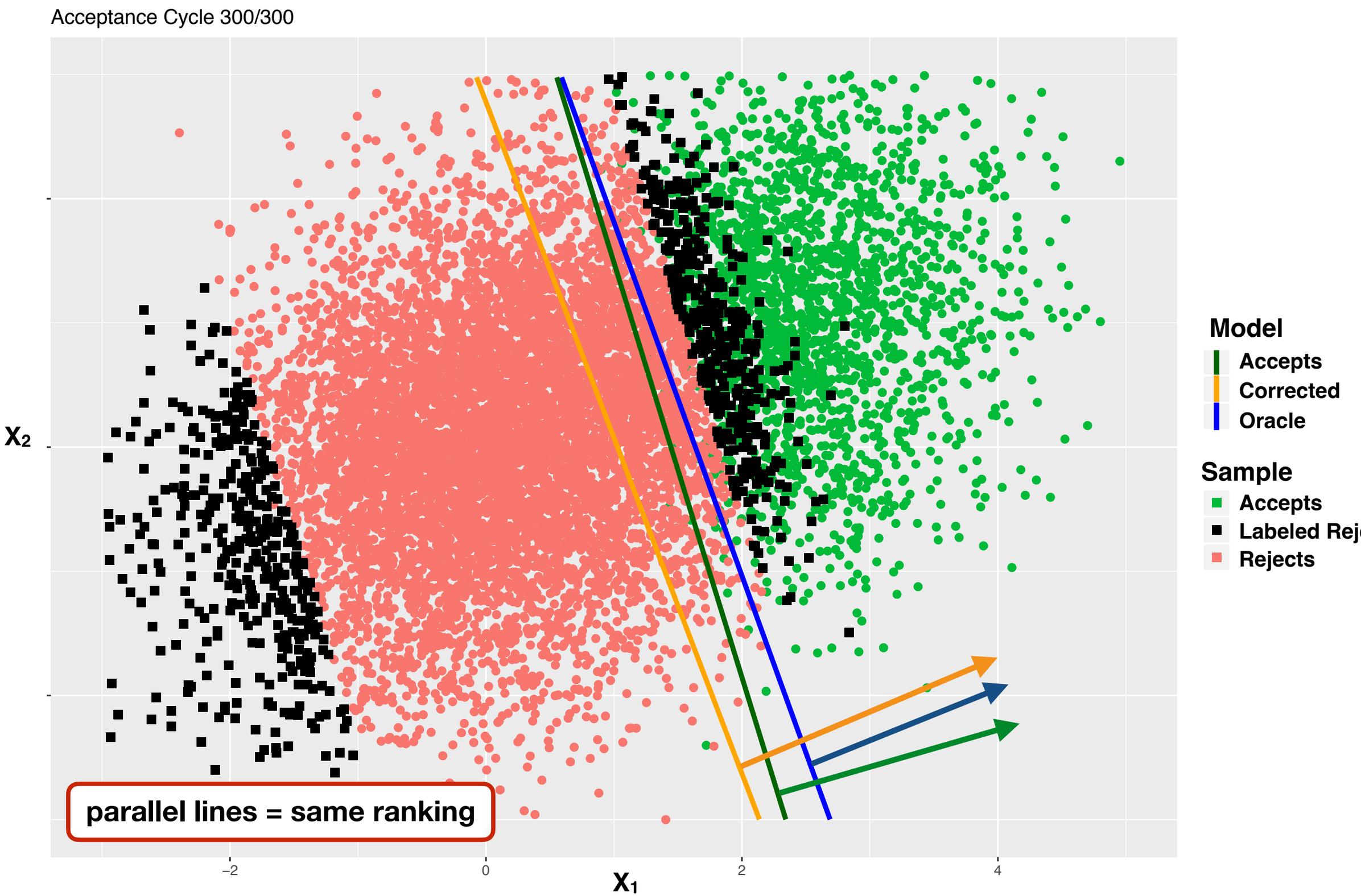
- **original self-learning algorithm as reference**
- **incrementally activate proposed extensions**
- **measure marginal performance of each extension**

BASL: Illustrative Example

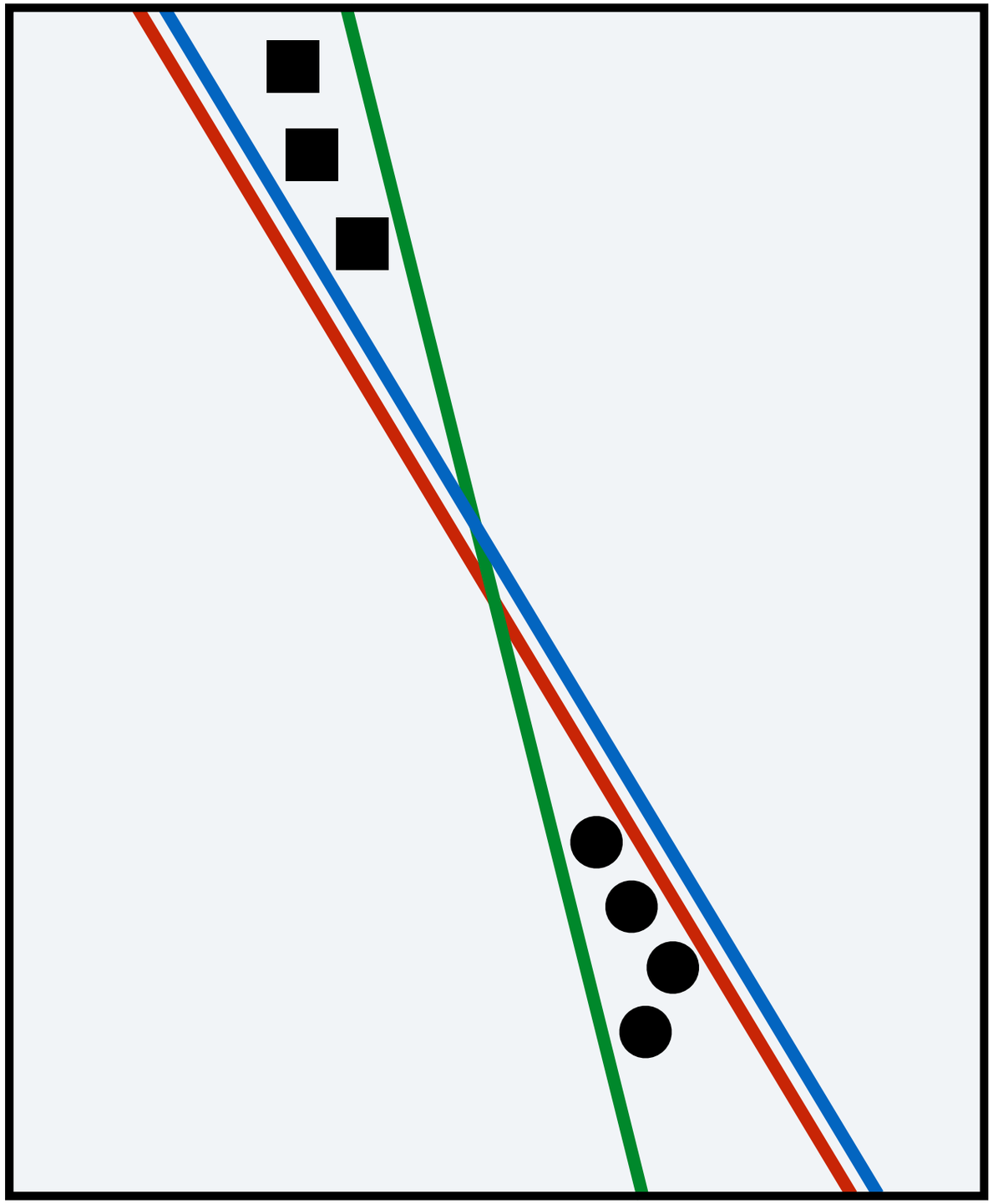


URL

BASL: Illustrative Example



Results on Synthetic Data: Example



- **Red model** ranks ■ higher than ●
- **Blue model** ranks ■ higher than ●
- **Green model** ranks ● higher than ■

BASL: Simulation Results

Missing at random (MAR)

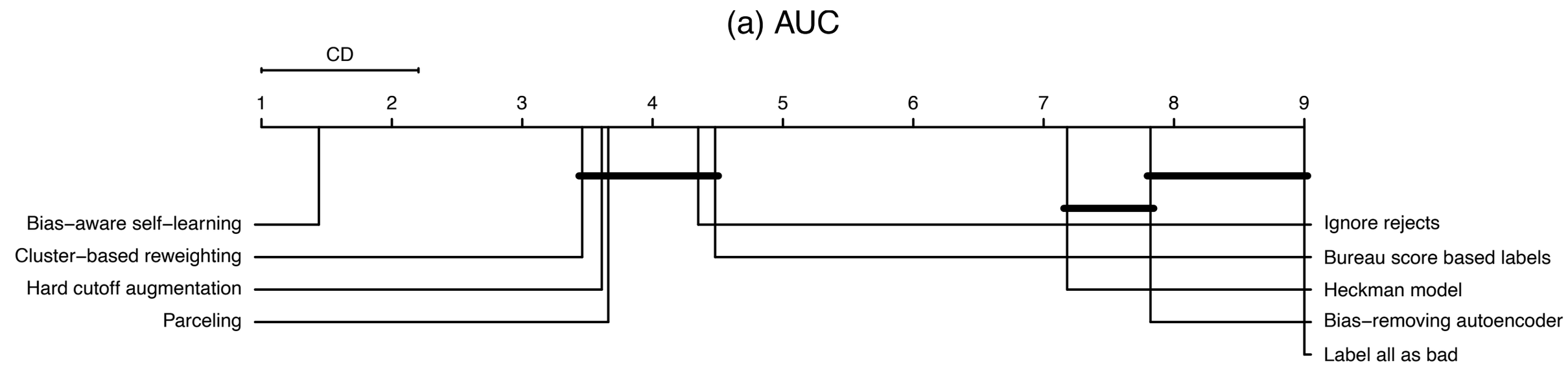
	Loss due to bias	Gain from BASL	Gain from Heckman
AUC	.0597	**39.83%**	11.54%
PAUC	.0528	**24.98%**	-39.81%
BRS	.0482	**40.47%**	-116.30%
ABR	.0589	**27.71%**	-40.48%
MMD	.5737	**7.93%**	-

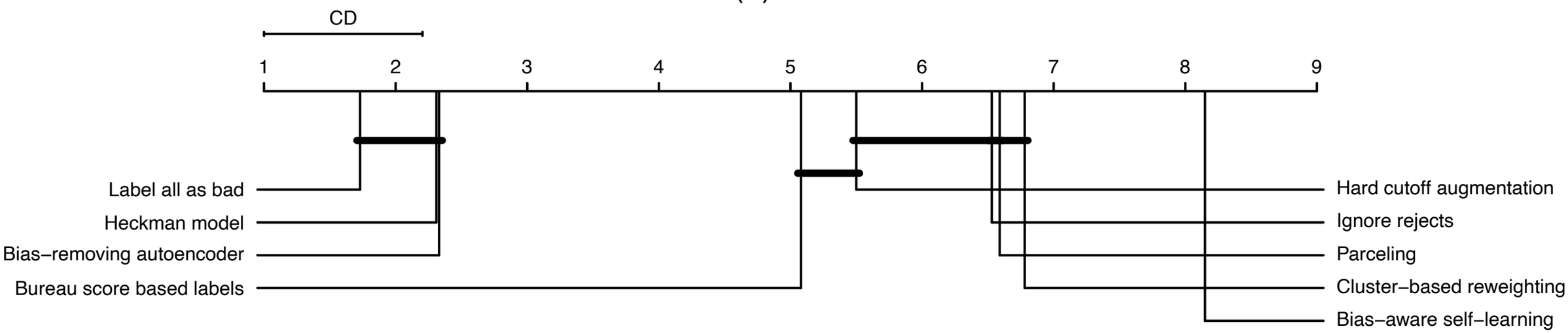
Missing not at random (MNAR)

	Loss due to bias	Gain from BASL	Gain from Heckman
AUC	.0589	**48.84%**	27.36%
PAUC	.0488	**33.93%**	-21.93%
BRS	.0404	**45.28%**	-129.17%
ABR	.0547	**36.86%**	-18.08%
MMD	.5746	**7.97%**	-

- BASL helps to improve **scorecard performance** in both settings
- data augmented with BASL demonstrates **lower sampling bias**
- all differences are **statistically significant** at 5% (Nemenyi rank test)

Significance Tests: Nemenyi Plots





A4. EVALUATION

Bayesian Evaluation Framework

Notation

$X \in \mathbb{R}^k$	denotes a loan applicant
$\mathbf{X} = (X_1, \ldots, X_n)^\top$	matrix of applicants' characteristics
$y \in \{0,1\}$	class label indicating repayment ($y = 0$) or default ($y = 1$)
$\mathbf{y}$	random vector of binary labels
$\mathbf{P}_{XY}, \mathbf{P}_X, \mathbf{P}_Y$	joint and marginal distributions of $\mathbf{X}$ and $\boldsymbol{y}$

Modeling task

- Given a set of IID credit applications $D = \{(\mathbf{X}, \mathbf{y})\}$ with $(\mathbf{X}, \mathbf{y}) \sim \mathbf{P}_{XY}$
- Lender's approval policy partitions D into $D^a = (\mathbf{X}^a, \mathbf{y}^a)$ and $D^r = (\mathbf{X}^r)$
- Infer function $f(\mathrm{X})$ approximating $\mathbf{P}(y = 1|X)$
  - Due to filtering, D^a has different empirical joint and marginal distributions compared to $\mathbf{P}_{XY}, \mathbf{P}_X, \mathbf{P}_Y$
  - Assuming MNAR, $\mathbf{P}(y = 1|X) \neq \mathbf{P}(y = 1|X^a)$ holds independent of the scoring model

input : model $f(X)$, evaluation sample S consisting of labeled accepts $S^a = \{(\mathbf{X}^a, \mathbf{y}^a)\}$ and unlabeled rejects $\mathbf{X}^r$, prior $\mathbf{P}(\mathbf{y}^r|X^r)$, evaluation metric $M(f, S, \tau)$, meta-parameters j_{max}, ϵ

output: Bayesian evaluation metric $\mathrm{BM}(f, S, \tau)$

$j = 0$; $\Delta = \epsilon$; $E^c = \{\}$; // initialization

while $(j \leq j_{max})$ *and* $(\Delta \geq \epsilon)$ **do**

  $j = j + 1$

  $\mathbf{y}^r = \mathrm{binomial}(1, \mathbf{P}(\mathbf{y}^r|\mathbf{X}^r))$; // generate labels of rejects

  $S_j = \{(\mathbf{X}^a, \mathbf{y}^a)\} \cup \{(\mathbf{X}^r, \mathbf{y}^r)\}$; // construct evaluation sample

  $E^c_j = \sum_{i=1}^{j} M(f(X), S_i, \tau)/j$; // evaluate

  $\Delta = E^c_j - E^c_{j-1}$; // check convergence

end

return $BM(f, S, \tau) = E^c_j$

Bayesian Evaluation: Simulation Results

Missing at random (MAR)

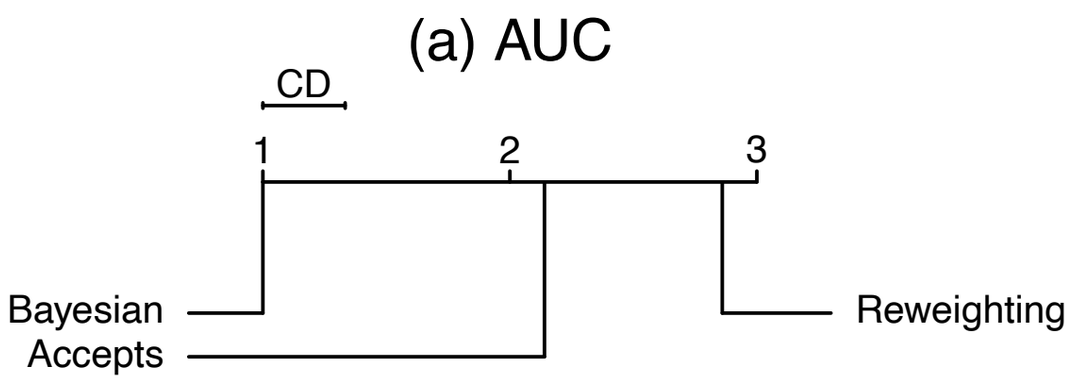
	Method	Bias	Variance	RMSE
AUC	Accepts	.2018	.0423	.2290
	IPW	.0762	**.0049**	.1113
	Bayesian	**.0093**	.0053	**.0740**
PAUC	Accepts	.2720	.0338	.2831
	IPW	.3120	.0222	.3525
	Bayesian	**.0113**	**.0076**	**.0835**
BRS	Accepts	.0797	**.0006**	.0907
	IPW	.0283	.0115	.0644
	Bayesian	**.0139**	.0026	**.0545**
ABR	Accepts	.1946	**.00003**	.1998
	IPW	.1915	.0003	.1973
	Bayesian	**.0166**	.0053	**.0956**

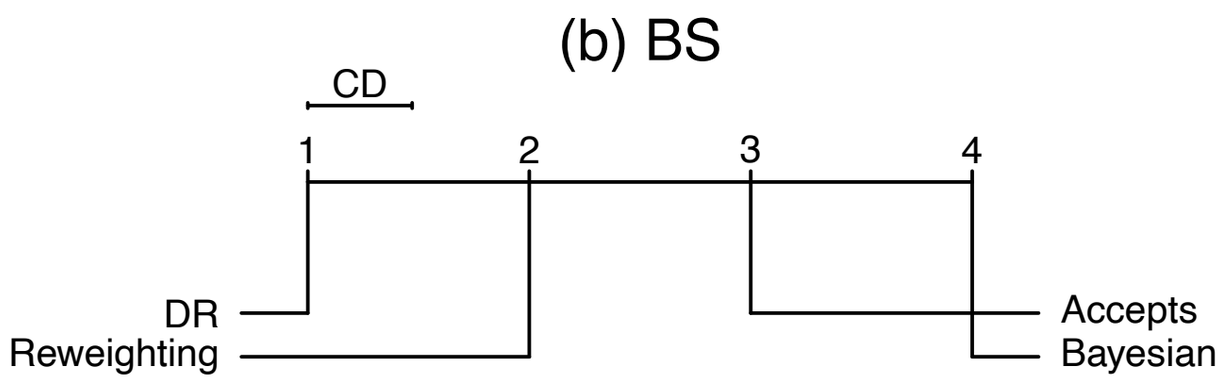
Missing not at random (MNAR)

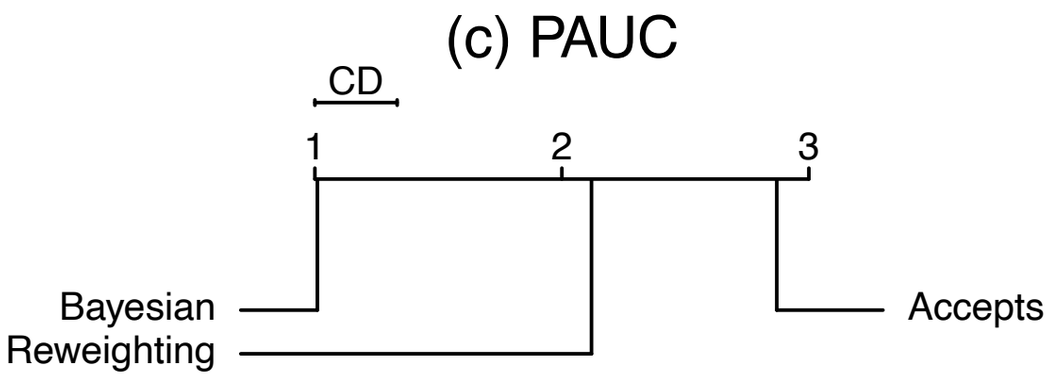
	Method	Bias	Variance	RMSE
AUC	Accepts	.1843	.0253	.2072
	IPW	.0707	.0047	.0988
	Bayesian	**.0068**	**.0048**	**.0672**
PAUC	Accepts	.2579	.0258	.2699
	IPW	.2969	.0145	.3346
	Bayesian	**.0117**	**.0063**	**.0788**
BRS	Accepts	.0754	**.0003**	.0829
	IPW	.0210	.0086	.0552
	Bayesian	**.0171**	.0026	**.0526**
ABR	Accepts	.2017	**.0001**	.2058
	IPW	.1988	.0003	.2034
	Bayesian	**.0065**	.0079	**.0909**

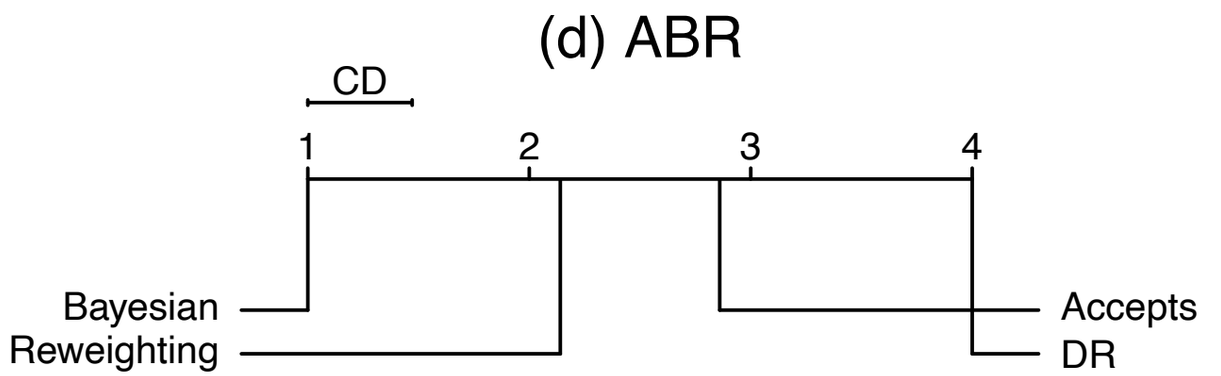
- Bayesian framework provides estimates with lower **bias** and **RMSE**
- all RMSE differences are **significant** at 5%

Significance Tests: Nemenyi Plots

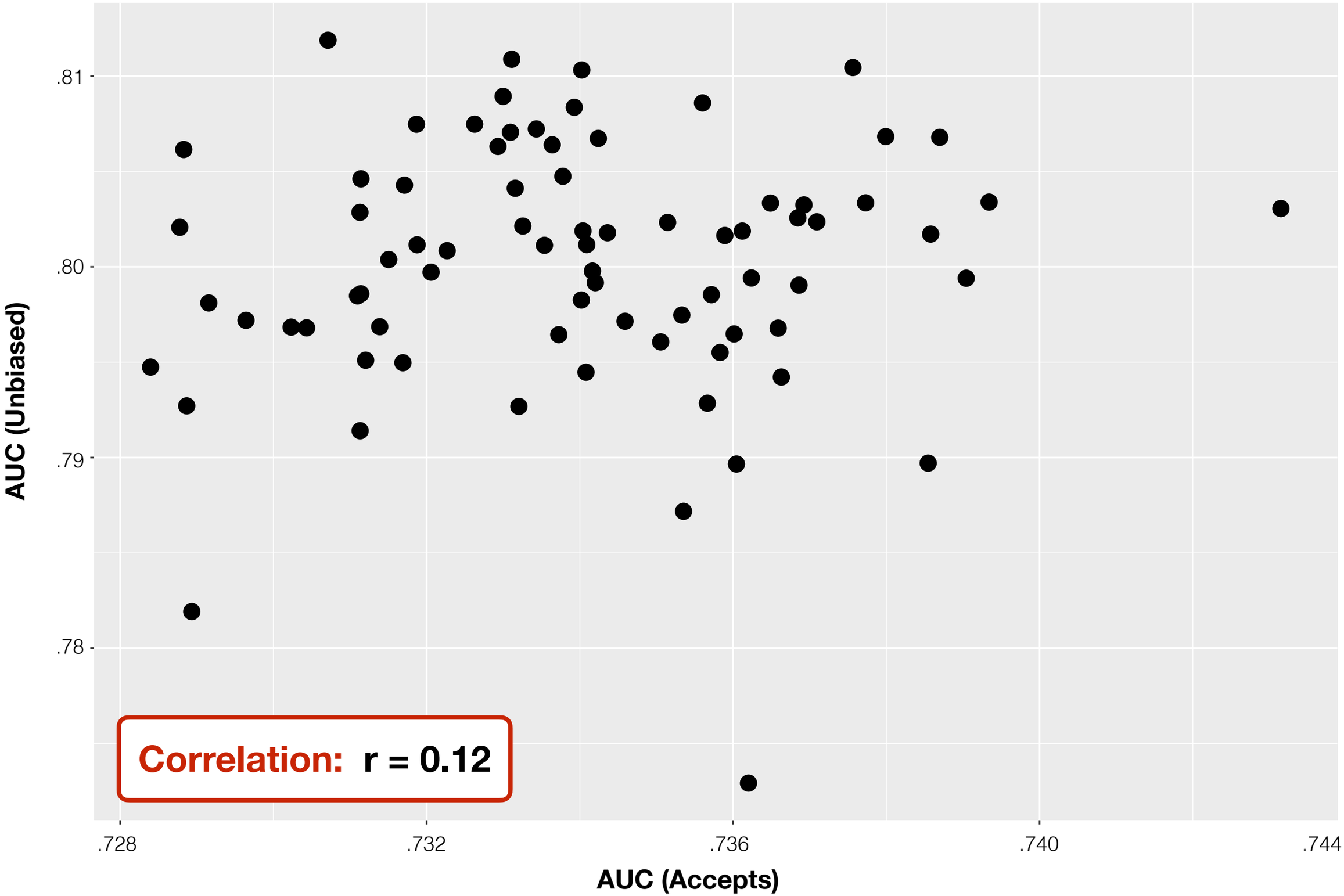








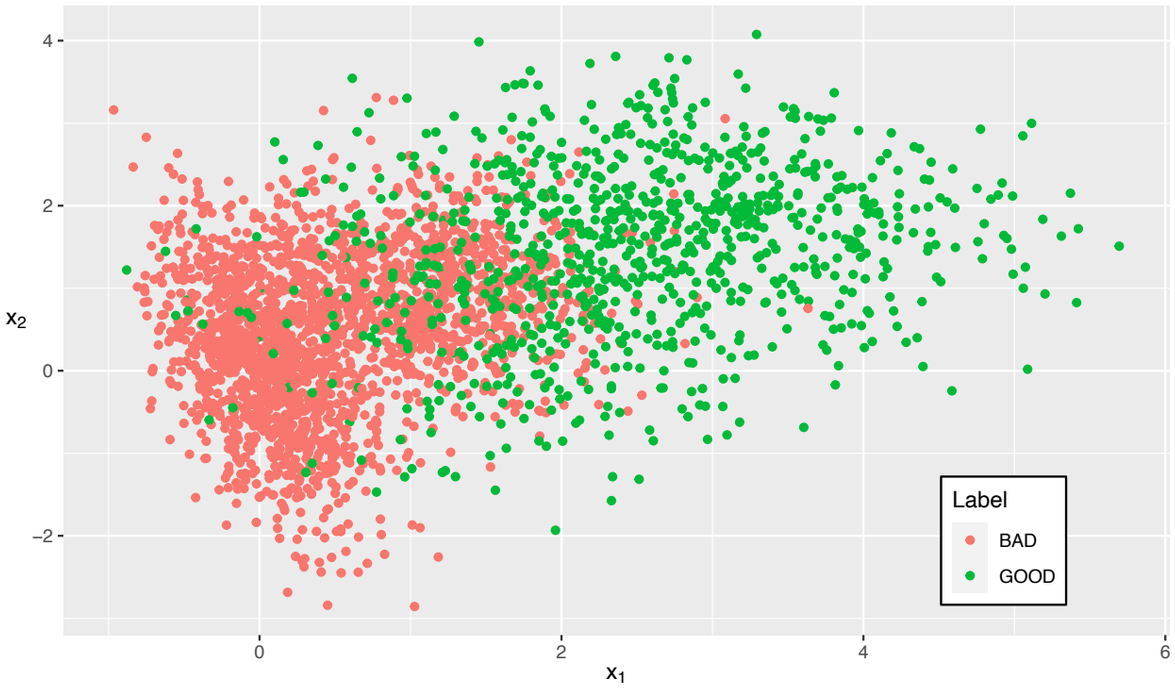
Evaluation Problem: Illustration

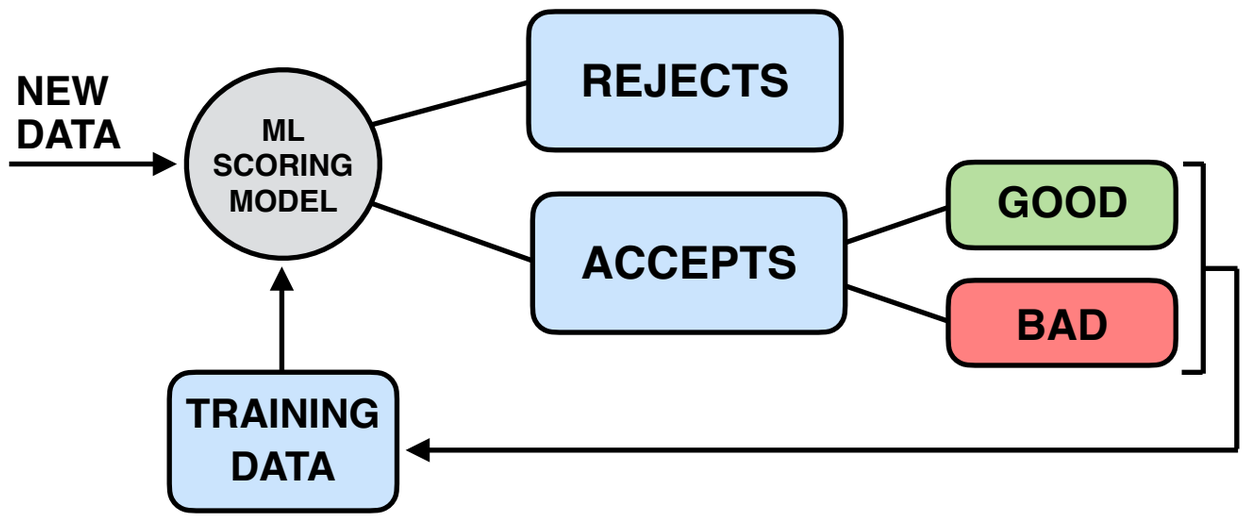


A5. SIMULATION

Simulation for Bias Illustration

- **sampling GOOD and BAD risks from Gaussian mixtures**
  - difference in class means and covariance
  - varying dimensionality, class imbalance, noise, etc.
- **simulating real-world acceptance loop**
  - generate a batch of new loan applications
  - use a scoring model to **accept** and **reject** applications
  - update the model after observing labels of **accepts**
  - repeat for 500 iterations and track performance on an unbiased sample





Data Generation & Acceptance Cycle

1. **Initial population**
   - Create a small (unbiased) sample of applicants from multivariate Gaussians
   - Use a simple rule to accept ***a%*** applicants (e.g. bureau score > ***b***)
   - Partition data into **A** (accepts) and **R** (rejects)

2. **Updating scorecard**
   - Train a scoring model **f(x)** on **A**

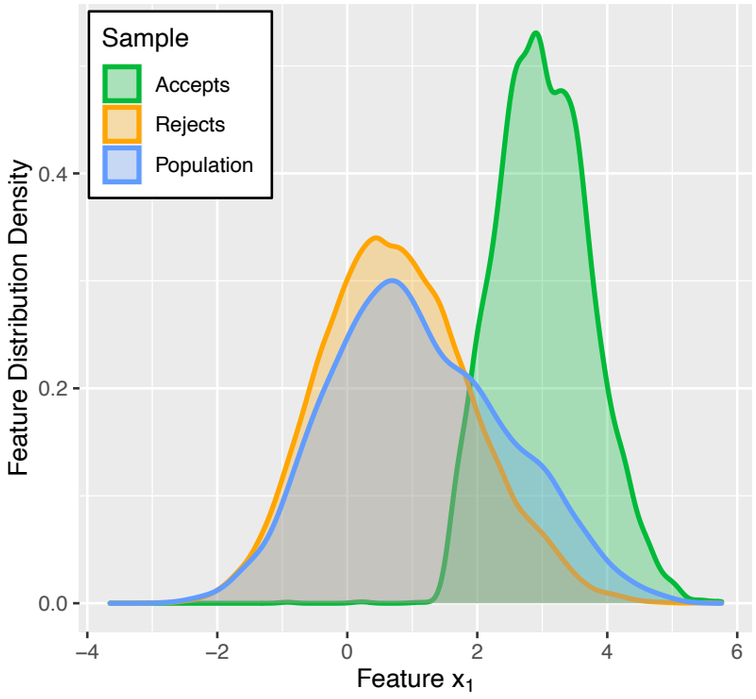
3. **New applicants**
   - Generate a new (unbiased) sample of applicants **N**
   - Score applicants in **N** using a scoring model **f(x)**
   - Use model scores to accept ***a%*** applicants from **N**
   - Partition **N** into A_N (new accepts) and R_N (new rejects)

4. **Merging data**
   - append A_N (with labels) to **A**; append R_N (no labels) to **R**

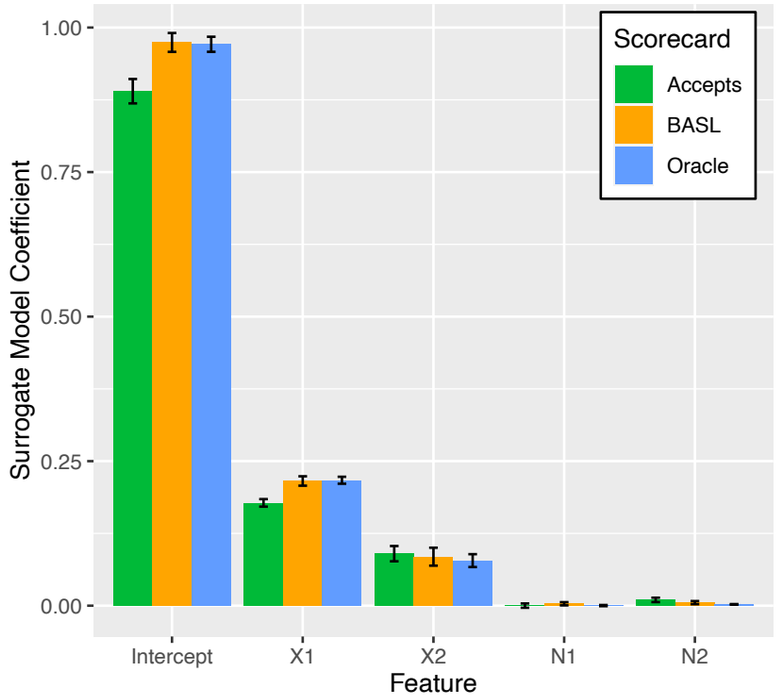
repeat for *g* iterations

Simulation Study Results

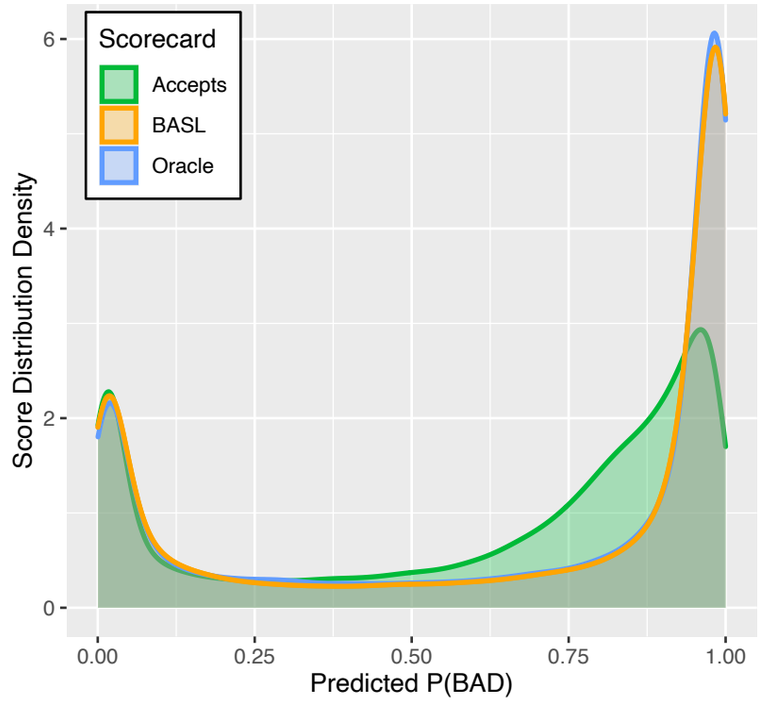
(a) Bias in Data



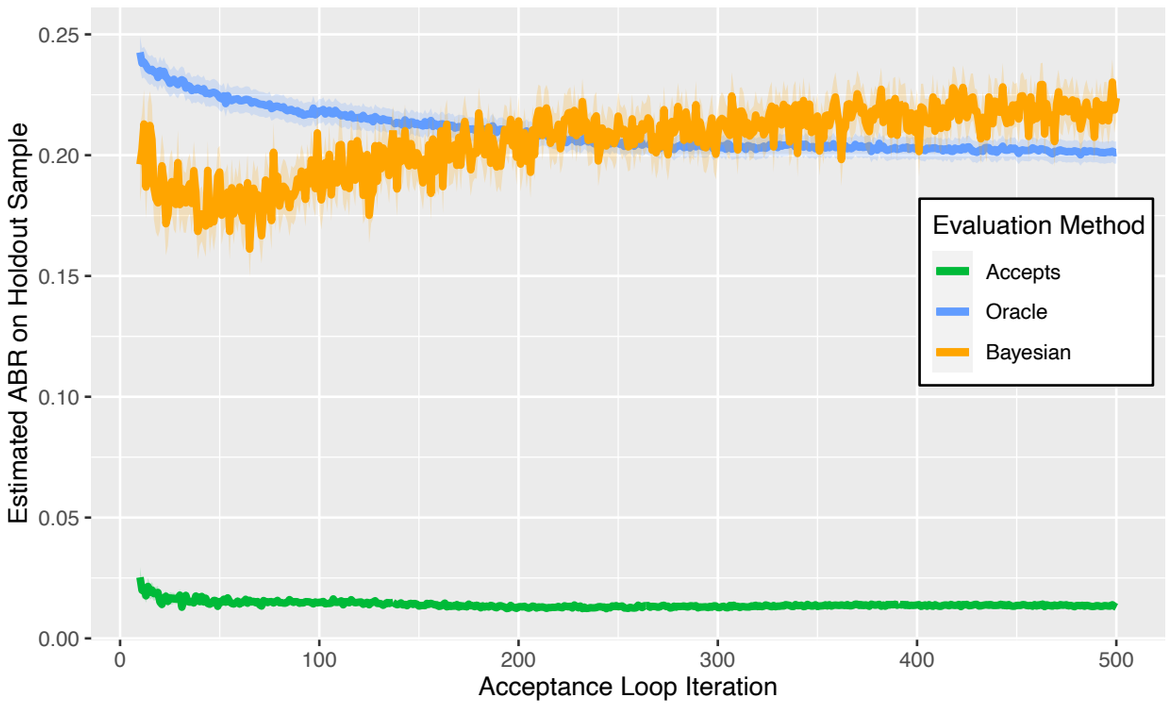
(b) Bias in Model



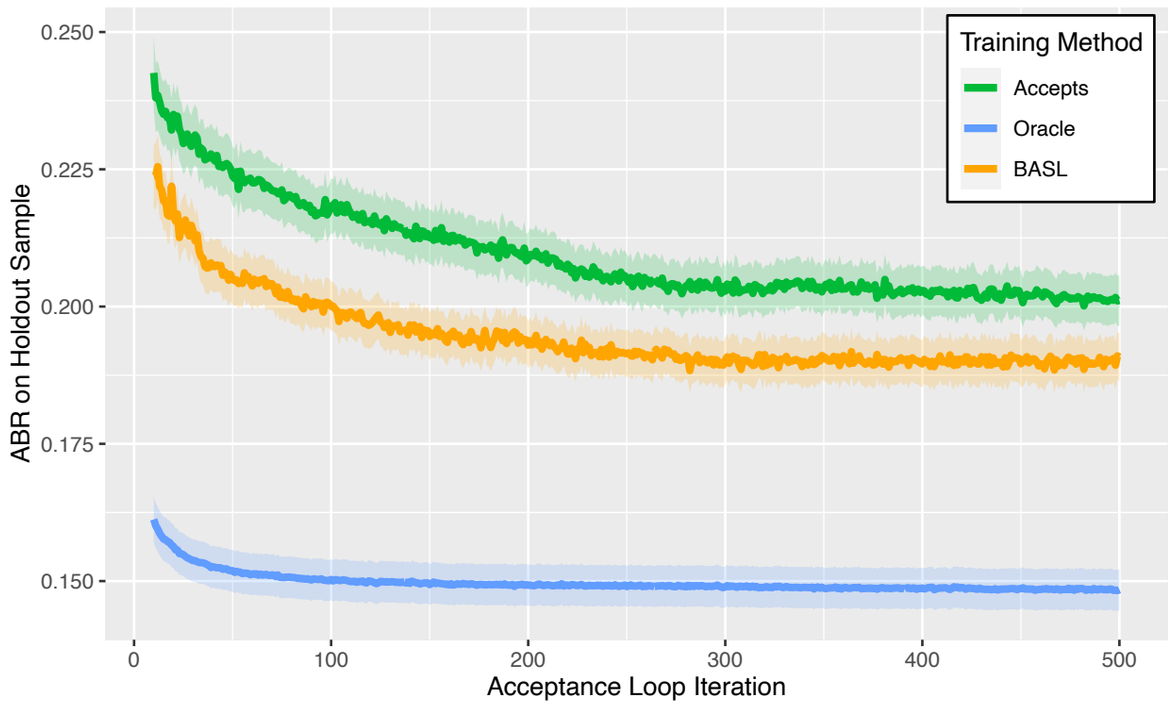
(c) Bias in Predictions



(d) Impact of Bias on Evaluation



(e) Impact of Bias on Training



Simulation Study Summary

Bayesian evaluation

- useful under both **MAR** and **MNAR**
  - consistent and statistically significant gains
- more helpful at **low acceptance**
  - error is present at any acceptance < 100%
  - highest gains at low acceptance
- performance depends on **two factors:**
  - **validation set distribution**
    - important to match population distribution
    - feasible when storing data on past **rejects**
  - **prior quality**
    - both accuracy and calibration of **P(BAD)** affect the performance
    - recommend using calibrated scores from the current scorecard

Bias-aware self-learning

- useful under both **MAR** and **MNAR**
  - consistent and statistically significant gains
- more helpful at **low acceptance**
  - loss is present if acceptance < 30%
  - highest gains at lower acceptance
- more helpful in **simpler tasks**
  - loss due to bias is consistently present irrespective of the complexity
  - easier to label rejects
- more helpful at **moderate imbalance**
  - highest gains at [2%, 5%] ratio of **BAD** applicants among **accepts**

Simulation Study: Missingness [1/2]

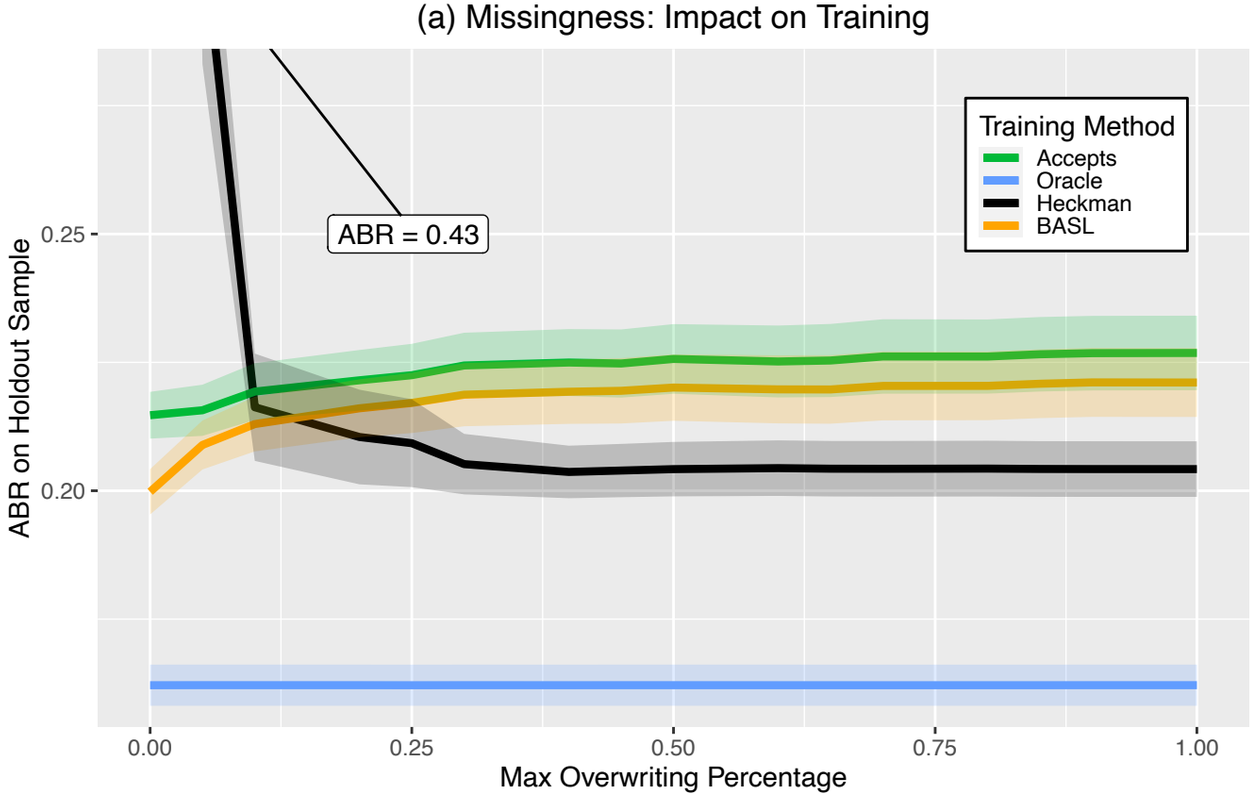
Missingness mechanism:

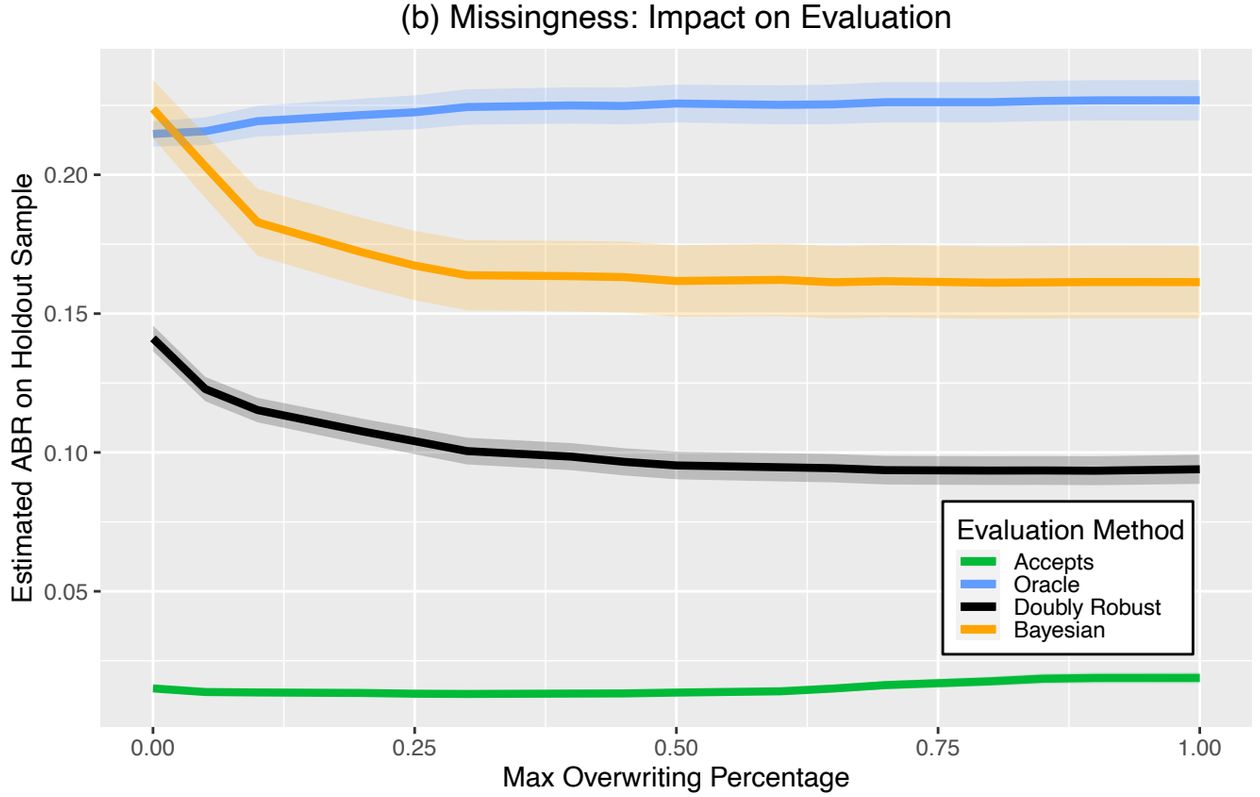
- **missing at random (MAR)**
  - generate data with two explanatory features
  - include both features in the scoring model
- **missing not at random (MNAR)**
  - generate data with three explanatory features
  - exclude one of the features from the scoring model
  - **omitted variable bias** leads to MNAR
    - variable that correlates with P(**BAD**) and P(**accepted**)
    - not included in both outcome and selection equations

Evaluation:

- **generating a holdout sample from population**
  - contains a representative sample of loan applications
  - allows to evaluate scoring model performance in real conditions

Simulation Study: Missingness [2/2]





- manual overwriting gears the data from **MAR** towards **MNAR**
  - introduces external feature in the selection equation
- **Bayesian evaluation** outperforms the best competitor in **both setups**
- **BASL** performs well when overwriting is **below 20%**
  - in credit scoring, such high values of overwriting are unlikely

BASL: Sensitivity Analysis

Goal:

analyzing how gains from **BASL** change under **different conditions**

Parameters:

- **past acceptance rate**
  - past acceptance policy determines the **magnitude of sampling bias**
  - sampling bias affects scorecard performance on unseen applicants
- **class imbalance**
  - class imbalance affects exposure to both client types in the **training data**
  - imbalance affects the magnitude of the loss due to sampling bias
- **class separation**
  - class separation affects the scorecard ability to distinguish **GOOD** and **BAD** risks
  - loss due to bias may depend on the classification task complexity

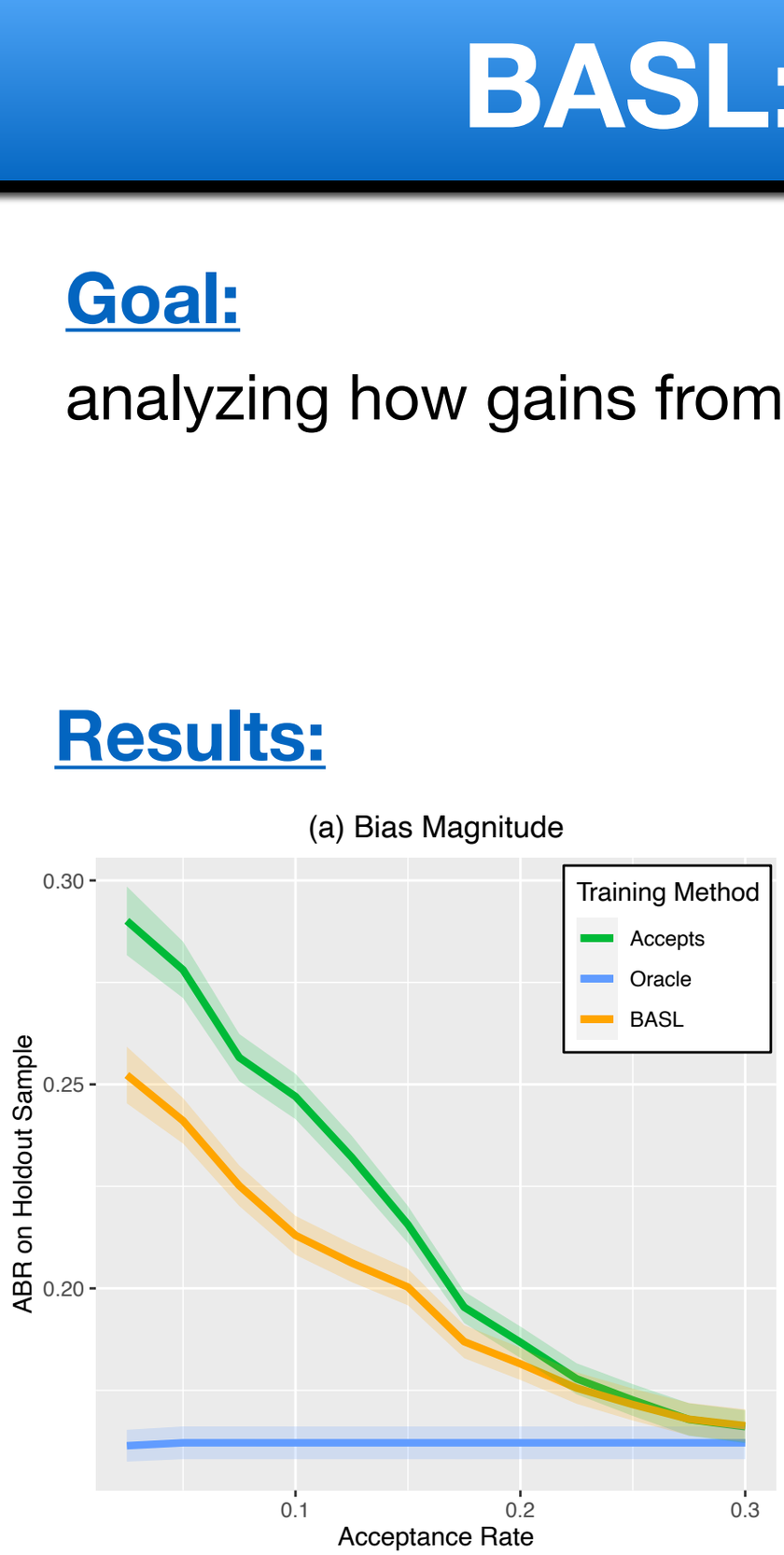
BASL: Sensitivity Analysis

Goal:

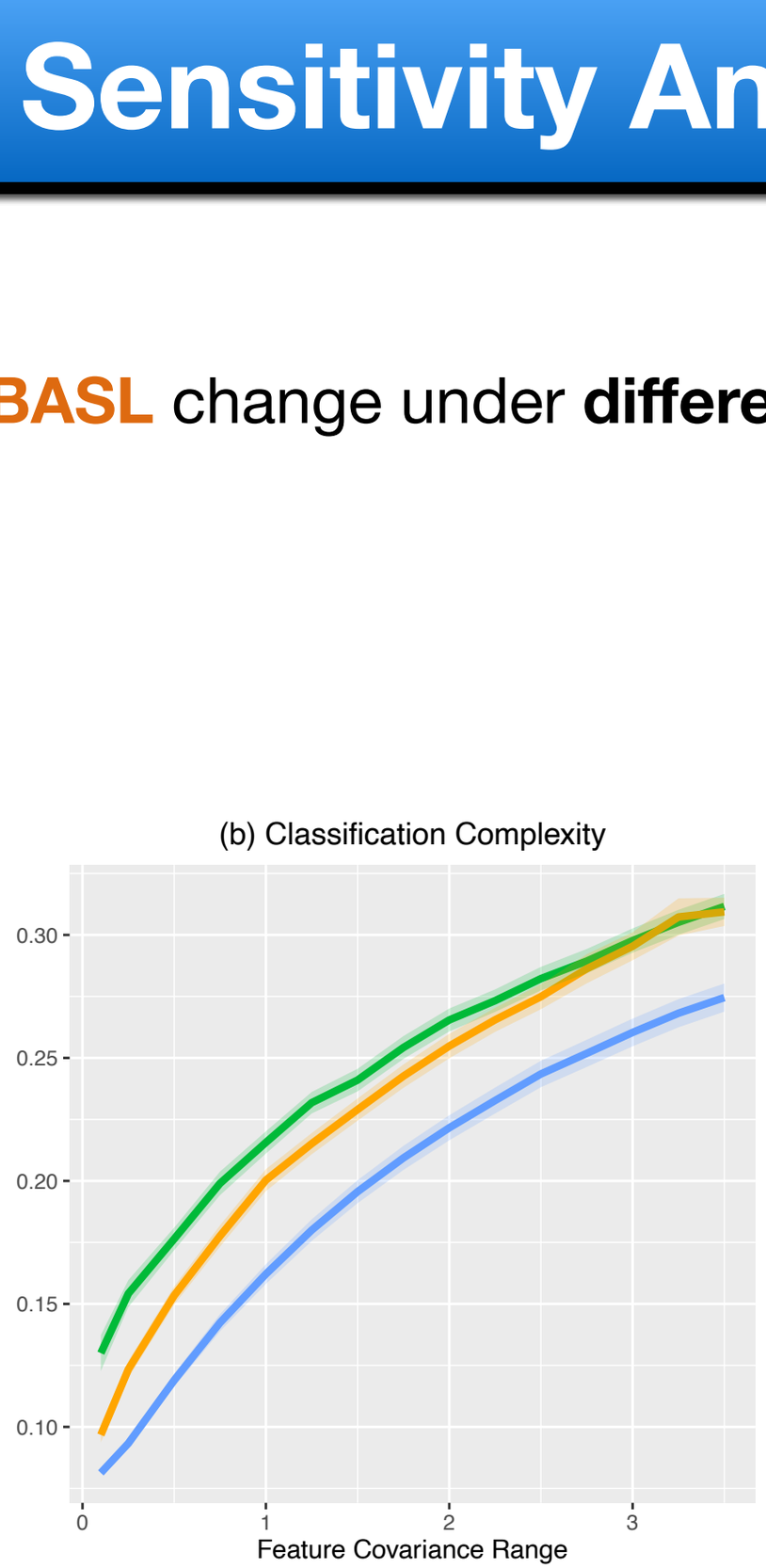
analyzing how gains from **BASL** change under **different conditions**

Results:

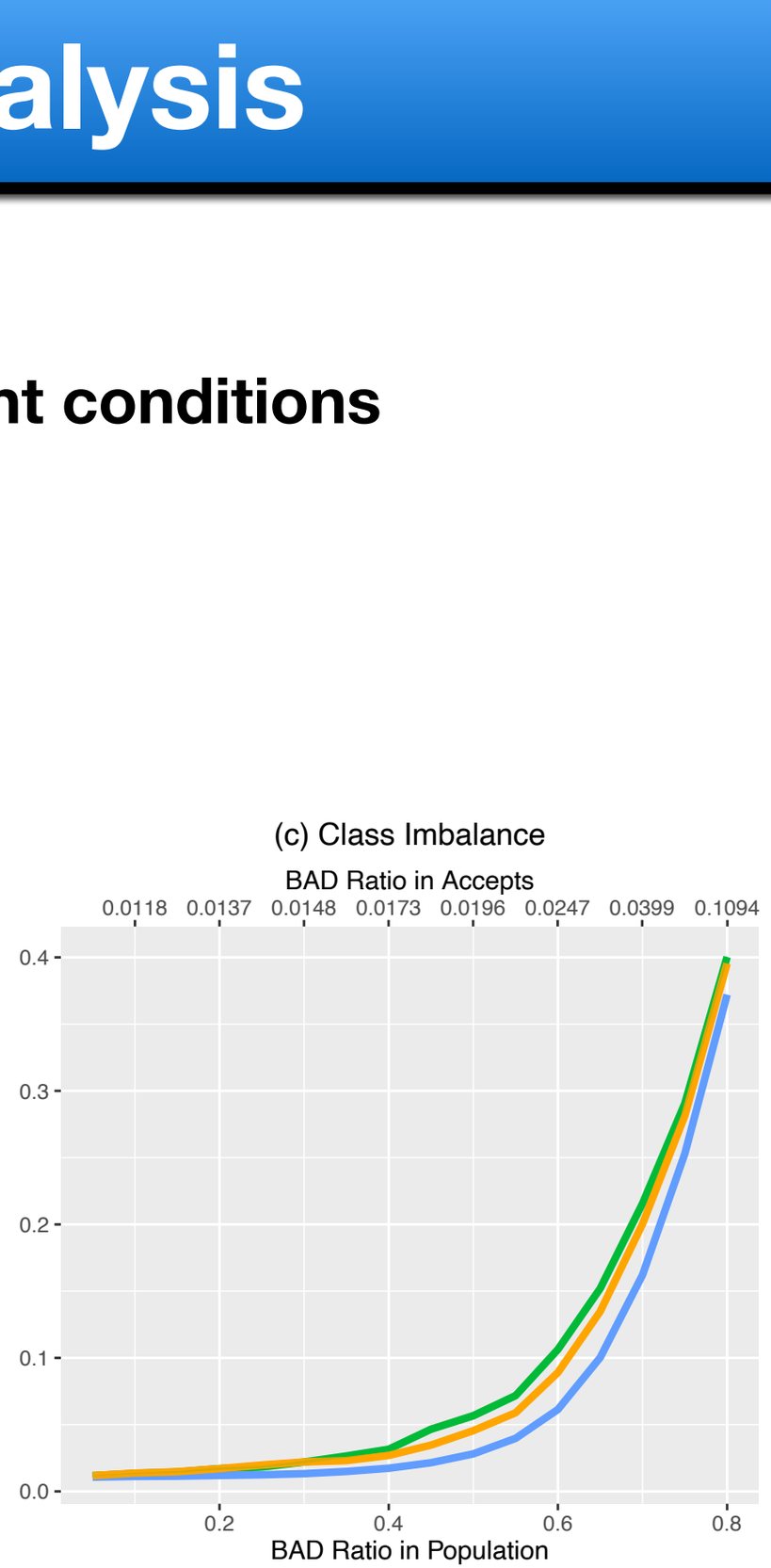
(a) Bias Magnitude



(b) Classification Complexity



(c) Class Imbalance



BASL Sensitivity: Acceptance [1/2]

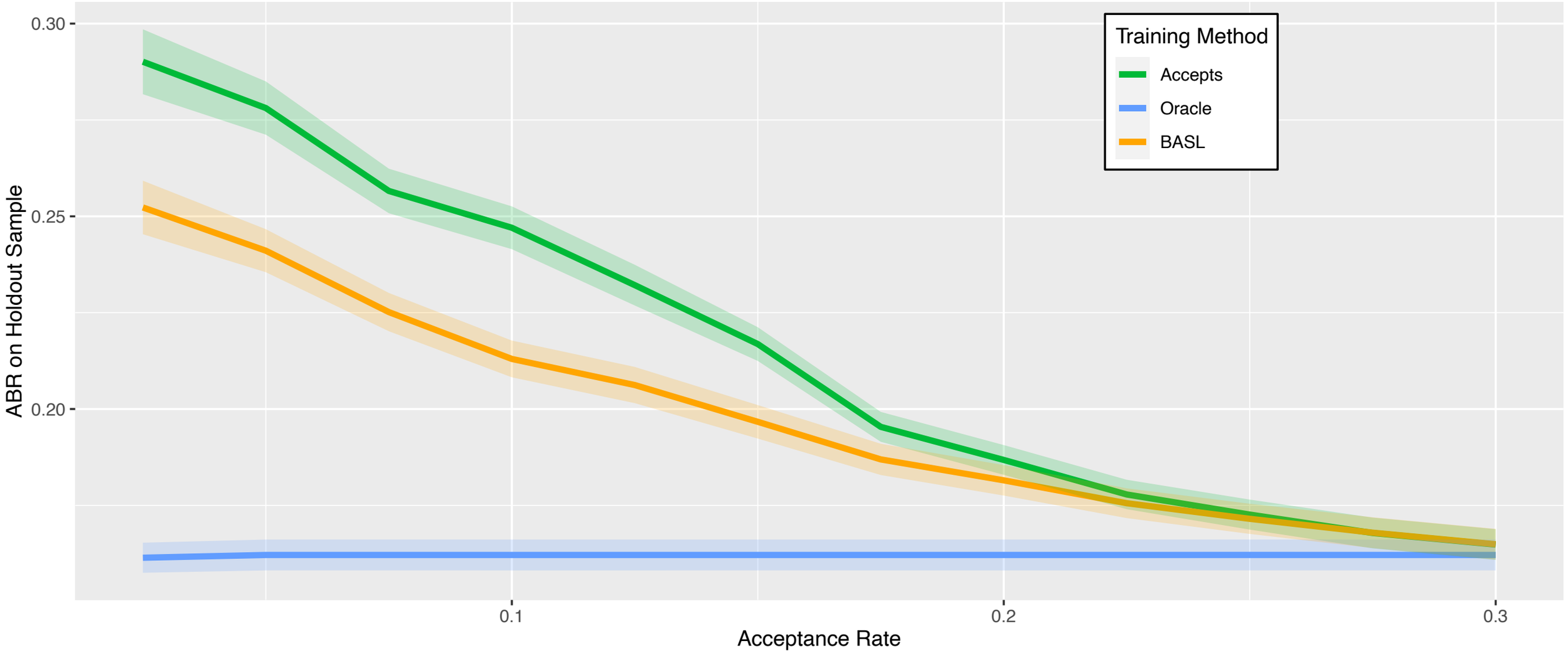
Setup:

- running acceptance loop with a grid of different **acceptance rates**
  - varying acceptance in **[1%, 2.5%, 5%, …, 30%]**
  - 100 simulation trials per each acceptance
- other acceptance loop parameters **are fixed**
  - same data generating process, acceptance policy, scorecard meta-parameters
  - only the split into **accepts**/**rejects** changes depending on acceptance
- evaluating performance of **BASL**
  - comparing to **accepts-based** scorecard and **oracle** scorecard
  - averaging performance over **100 trials**

Intuition:

- lower acceptance leads to a stronger sampling bias
- **BASL** should be more useful at lower acceptance

BASL Sensitivity: Acceptance [2/2]



Results:

- **lower acceptance** increases loss due to bias
  - correction is not required at acceptance above 25%
- **BASL** is more useful at lower acceptance
  - performance gains are highest at **very low acceptance**
  - more typical for prime market segments

BASL Sensitivity: Imbalance [1/2]

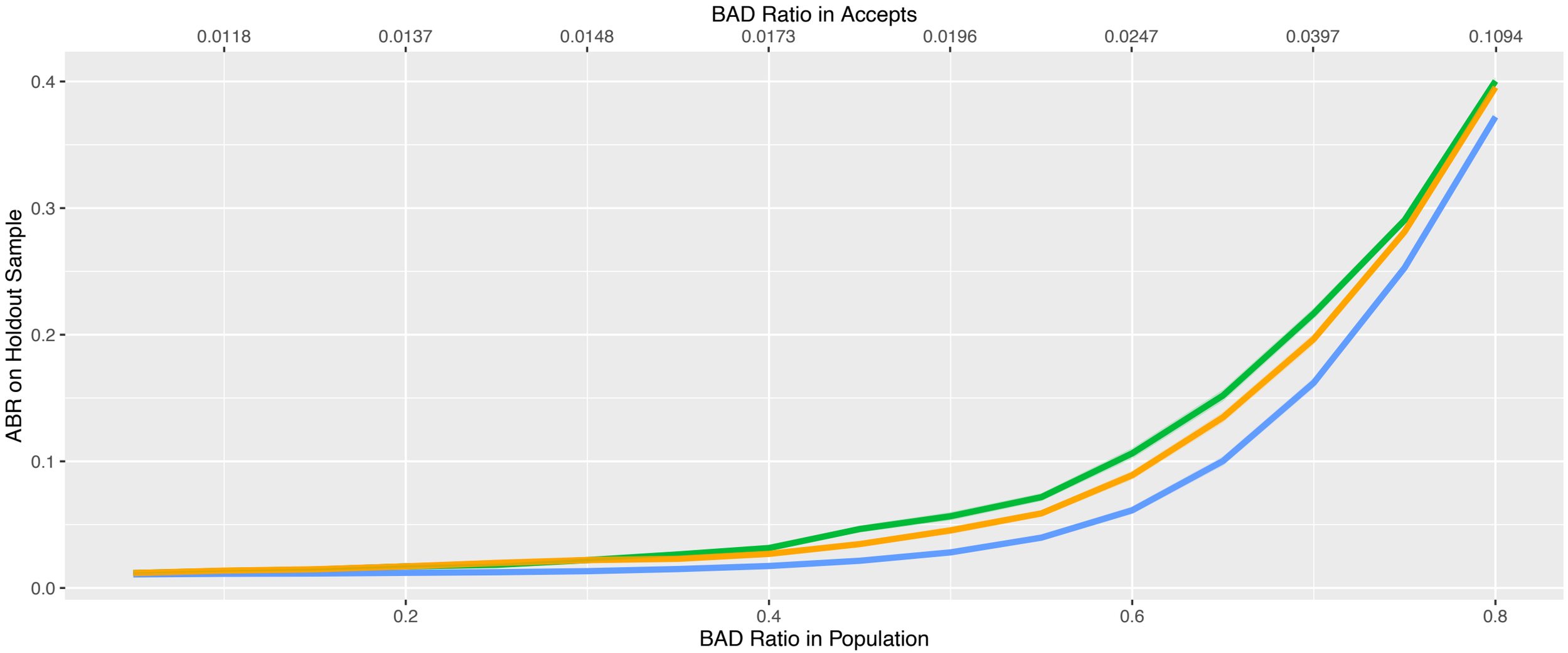
Setup:

- running acceptance loop with a grid of different **class imbalance ratios**
  - varying **BAD** rate in population in **[5%, 10%, 15%, …, 80%]**
  - 100 simulation trials per each **BAD** rate
- **imbalance in population** translates into **imbalance in accepts**
  - the latter depends on imbalance in population and the acceptance rate
  - manipulating **imbalance in population** as environment property
  - fixed **acceptance rate** to isolate the population class imbalance effect
- other acceptance loop parameters **are fixed**
- evaluating performance of **BASL**
  - comparing to **accepts-based** scorecard and **oracle** scorecard
  - averaging performance over **100 trials**

Intuition:

- severe imbalance limits the number of **BAD** cases in training data

BASL Sensitivity: Imbalance [2/2]



Results:

- even high **BAD** rate in population **creates imbalance** in **accepts**
- loss due to bias **shrinks** when class imbalance is **too severe**
  - specific to the ABR metric since it is based on the top least risky cases
- **BASL** is most helpful at **moderate imbalance** in **accepts** in [2%, 5%]

BASL Sensitivity: Separation [1/2]

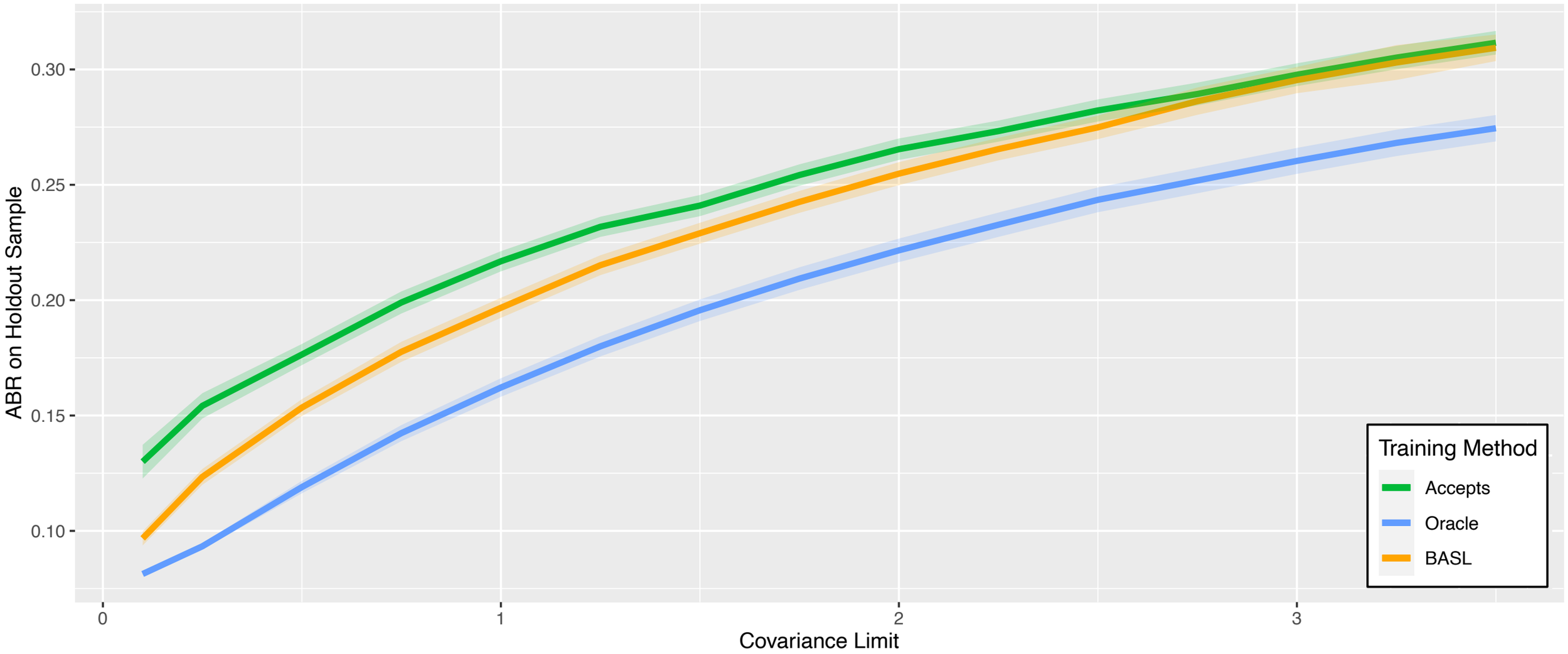
Setup:

- running acceptance loop with a grid of different **S**
  - **S** = **upper bound** on the randomly generated feature covariance values
  - varying **S** in **[0.25, 0.50, 0.75, …, 3.50]** to affect distribution
  - 100 simulation trials per each **S** value
- other acceptance loop parameters **are fixed**
- evaluating performance of **BASL**
  - comparing to **accepts-based** scorecard and **oracle** scorecard
  - averaging performance over **100 trials**

Intuition:

- higher sigma complicates the classification task
  - more intersection between the **GOOD** and **BAD** clusters
  - classes become more difficult to distinguish
- labeling rejects may be more difficult in complex tasks

BASL Sensitivity: Separation [2/2]



Results:

- loss due to bias is **consistently present**
  - magnitude does not depend on the classification task complexity
- **BASL** is more helpful for **simpler tasks**
  - easier to correctly label rejects by extrapolating patterns

Bayesian Evaluation: Sensitivity Analysis

Goals:

- analyze gains from **Bayesian framework** in **different conditions**
- **explain** the performance of **Bayesian framework**

Parameters:

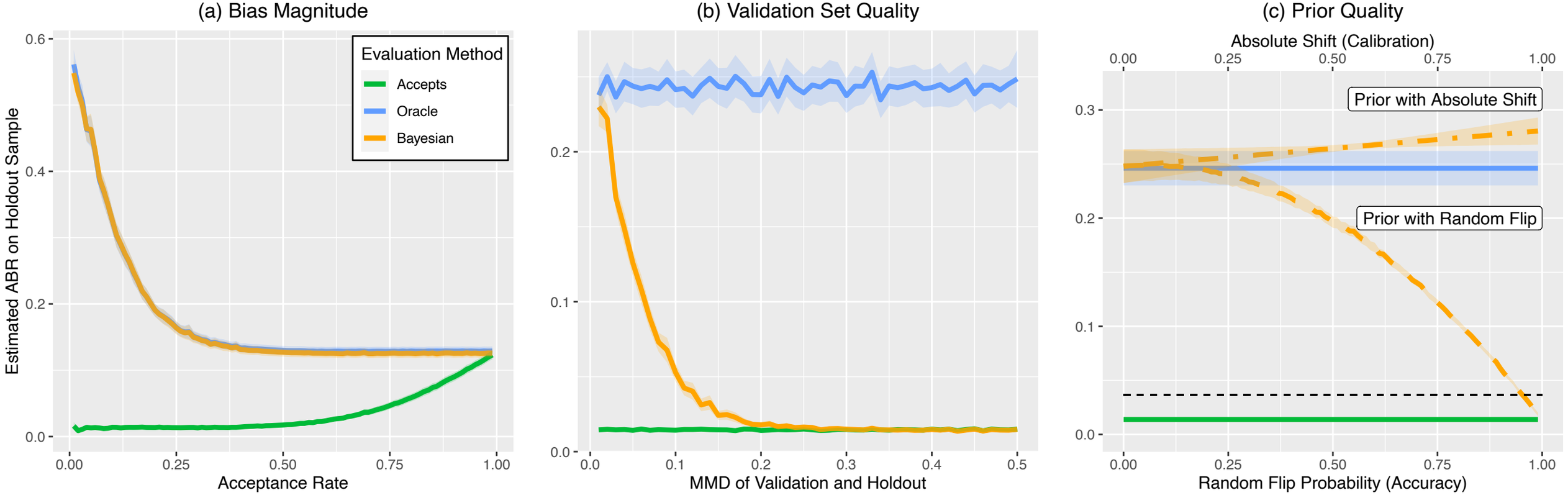
- **past acceptance**
  - acceptance policy determines the **magnitude of sampling bias**
  - sampling bias affects **evaluation error**
- **validation set quality**
  - Bayesian evaluation on validation set with **accepts** and **rejects**
  - **feature distribution** in validation set affects performance
- **prior P(BAD) quality**
  - **rejects** are pseudo-labeled using a prior **P(BAD)**
  - **accuracy** and **calibration** of prior affects performance

Bayesian Evaluation: Sensitivity Analysis

Goals:

- analyze gains from **Bayesian framework** in **different conditions**
- **explain** the performance of **Bayesian framework**

Results:



Bayesian Sensitivity: Acceptance [1/2]

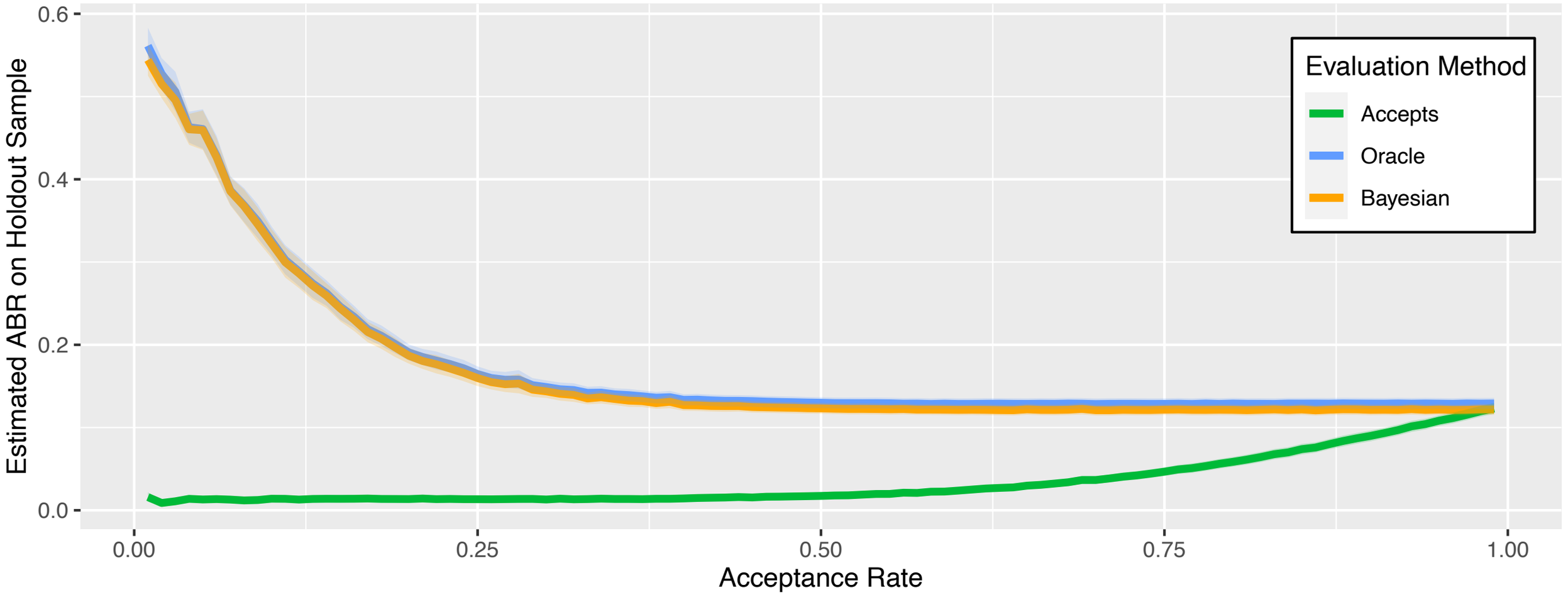
Setup:

- running acceptance loop with a grid of different **acceptance rates**
  - varying acceptance in **[1%, 2%, 3%, …, 100%]**
  - 100 simulation trials per each acceptance rate
- other acceptance loop parameters **are fixed**
- evaluating the **accepts-based** scorecard with different methods
  - **accepts-based** evaluation on a biased validation set
  - **oracle** evaluation on a sample from population
  - **Bayesian evaluation** on **accepts** and **rejects**
  - averaging performance over **100 trials**
- assuming **perfect prior** on **P(BAD)** for **rejects**
  - helps to isolate the acceptance effect
  - shows highest potential gains from the **Bayesian evaluation**

Intuition:

- lower acceptance leads to a stronger error due to sampling bias

Bayesian Sensitivity: Acceptance [2/2]



Results:

- evaluating on **accepts** is overoptimistic
  - ABR is consistently **underestimated**
  - evaluation error is present even at **high acceptance**
  - lower acceptance **increases evaluation error** due to sampling bias
- **Bayesian framework** helps at any acceptance
  - evaluation gains are higher at **low acceptance**

Bayesian Sensitivity: Validation Set [1/2]

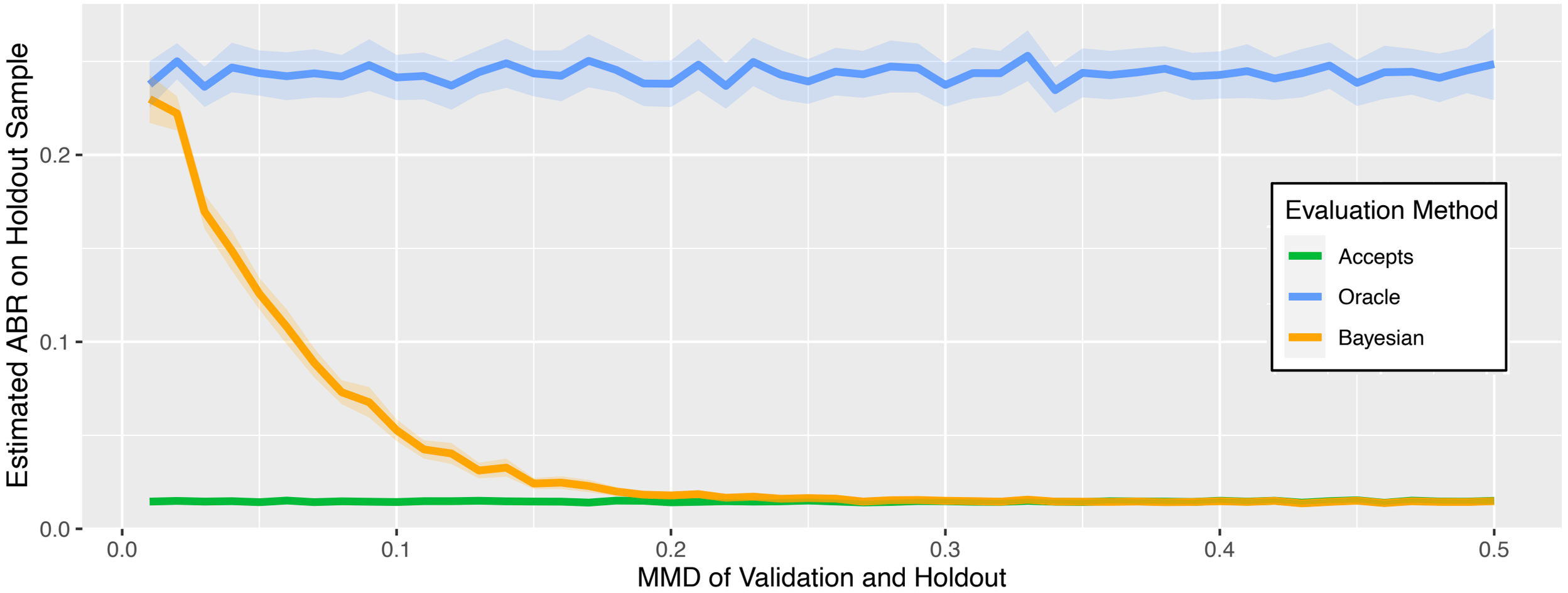
Setup:

- varying **accept**/**reject** ratio in validation set
  - the ration affects the feature distribution in the validation set
  - using **MMD** to measure **discrepancy** between validation and holdout
- other acceptance loop parameters **are fixed**
- evaluating the **accepts-based** scorecard with different methods
  - **accepts-based** evaluation on a biased validation set
  - **oracle** evaluation on a sample from population
  - **Bayesian evaluation** on **accepts** and **rejects**
  - averaging performance over **100 trials**
- assuming **perfect prior** on **P(BAD)** for **rejects**
  - helps to isolate the validation set quality effect

Intuition:

- better validation set should improve **Bayesian evaluation** performance

Bayesian Sensitivity: Validation Set [2/2]



Results:

- **matching population distribution** during evaluation is important
  - higher discrepancies lead to larger errors
- **Bayesian framework** shows good performance over **accepts** even under high distribution mismatch

Bayesian Sensitivity: Prior [1/2]

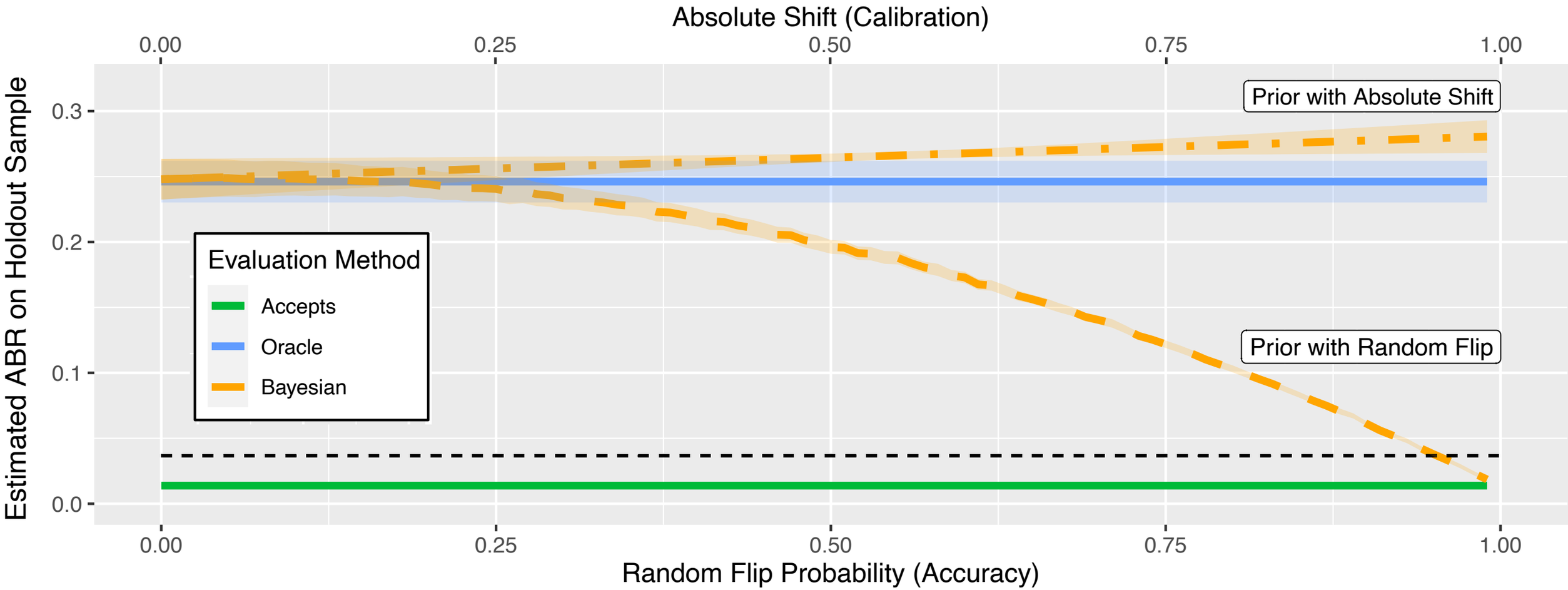
Setup:

- manipulating prior **P(BAD)** for **rejects** used by **Bayesian evaluation**
  - starting from the **perfect prior**
  - randomly flipping labels to affect **accuracy**
  - shifting labels to affect **calibration**
- other acceptance loop parameters **are fixed**
- evaluating the **accepts-based** scorecard with different methods
  - **accepts-based** evaluation on a biased validation set
  - **oracle** evaluation on a sample from population
  - **Bayesian evaluation** on **accepts** and **rejects**
  - averaging performance over **100 trials**

Intuition:

- better prior should improve **Bayesian evaluation** performance

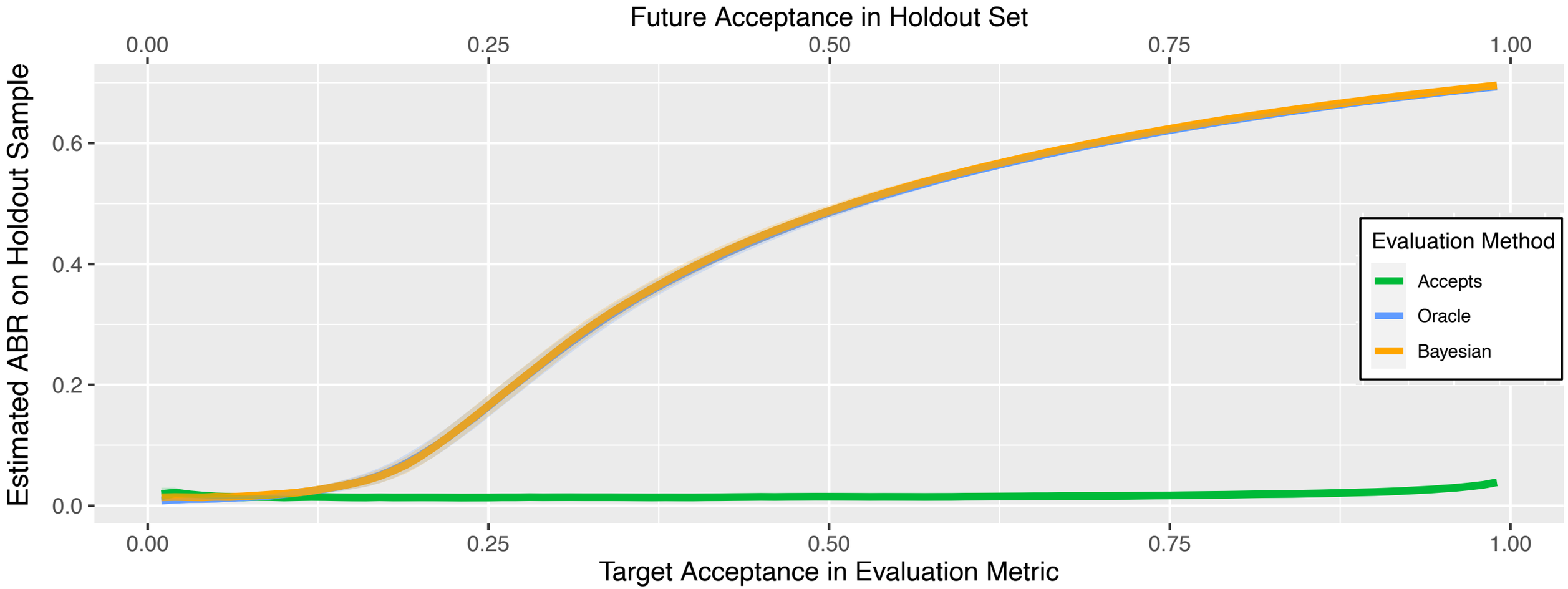
Bayesian Sensitivity: Prior [2/2]



Results:

- **prior quality** affects performance
  - both **accuracy** and **calibration**
  - **accuracy** is more important (for the ABR metric)
- **Bayesian framework** outperforms **accepts** even with a **poor prior**

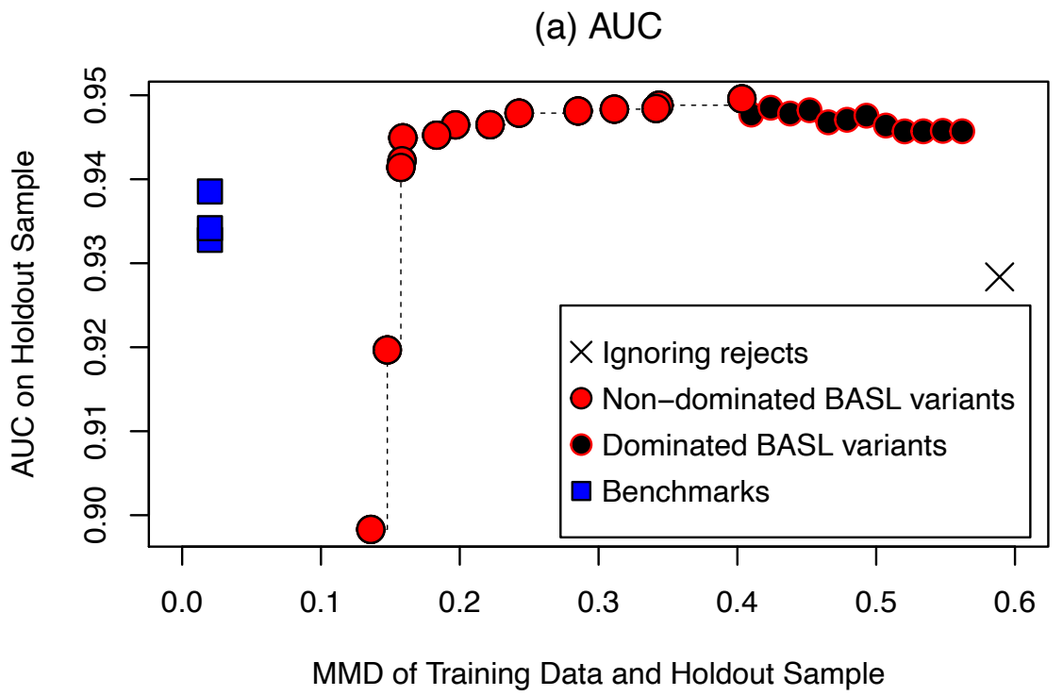
Bayesian Evaluation: Forward-Looking Tool

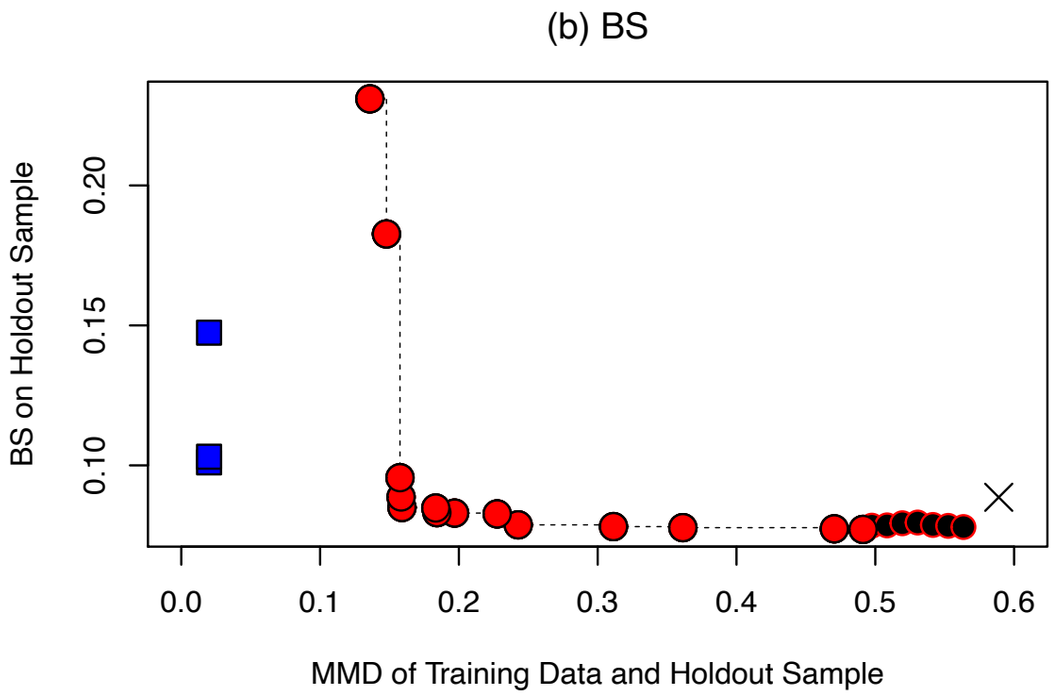


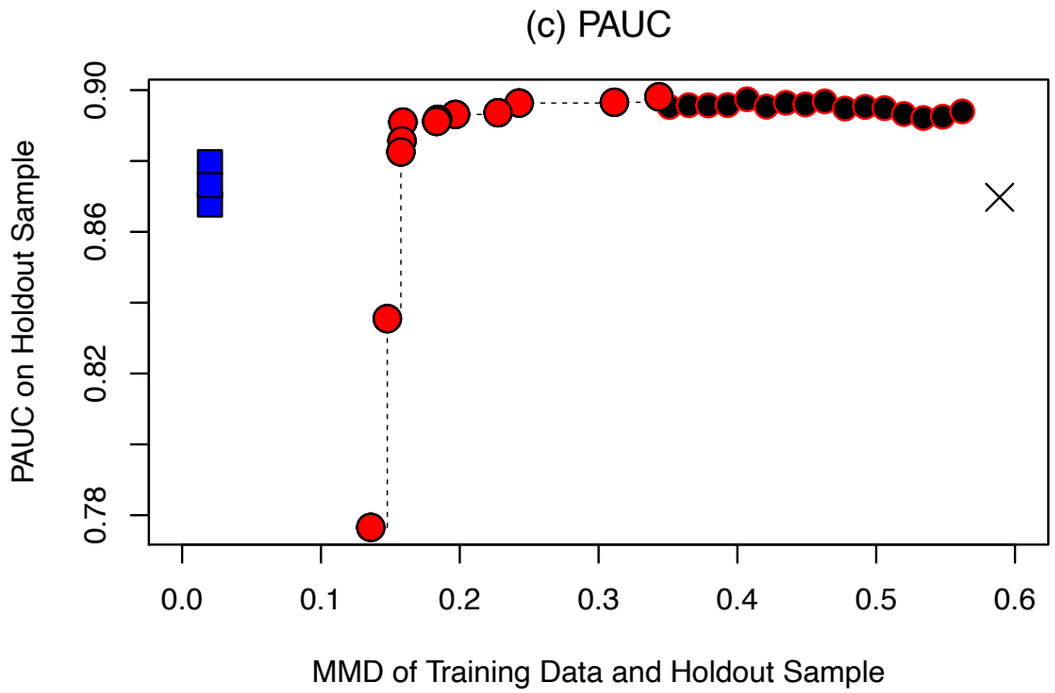
- varying **future acceptance** in [1%, 100%]
  - acceptance rate as a parameter of the ABR metric during evaluation
  - simulates **future changes** in the acceptance policy
- **Bayesian framework** provides good ABR predictions
  - evaluating on **accepts** is **misleading** at high target acceptance
  - **forward-looking tool** to predict future performance as a function of **acceptance**

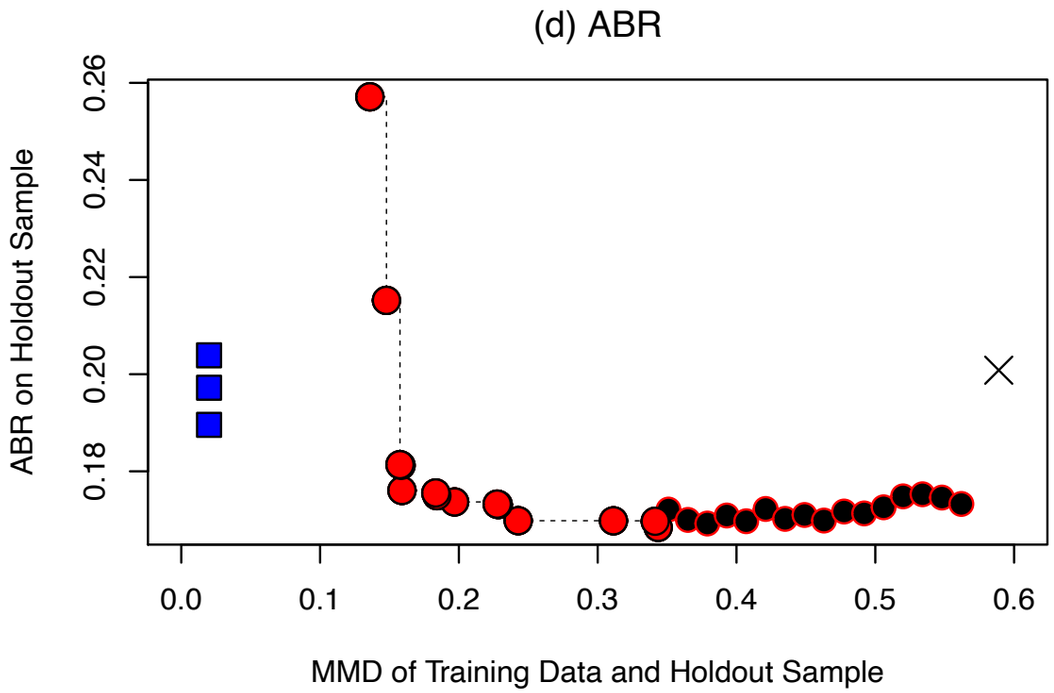
A6. FURTHER RESULTS

Data Augmentation: Pareto Frontiers









Heckman Model

Heckman model:

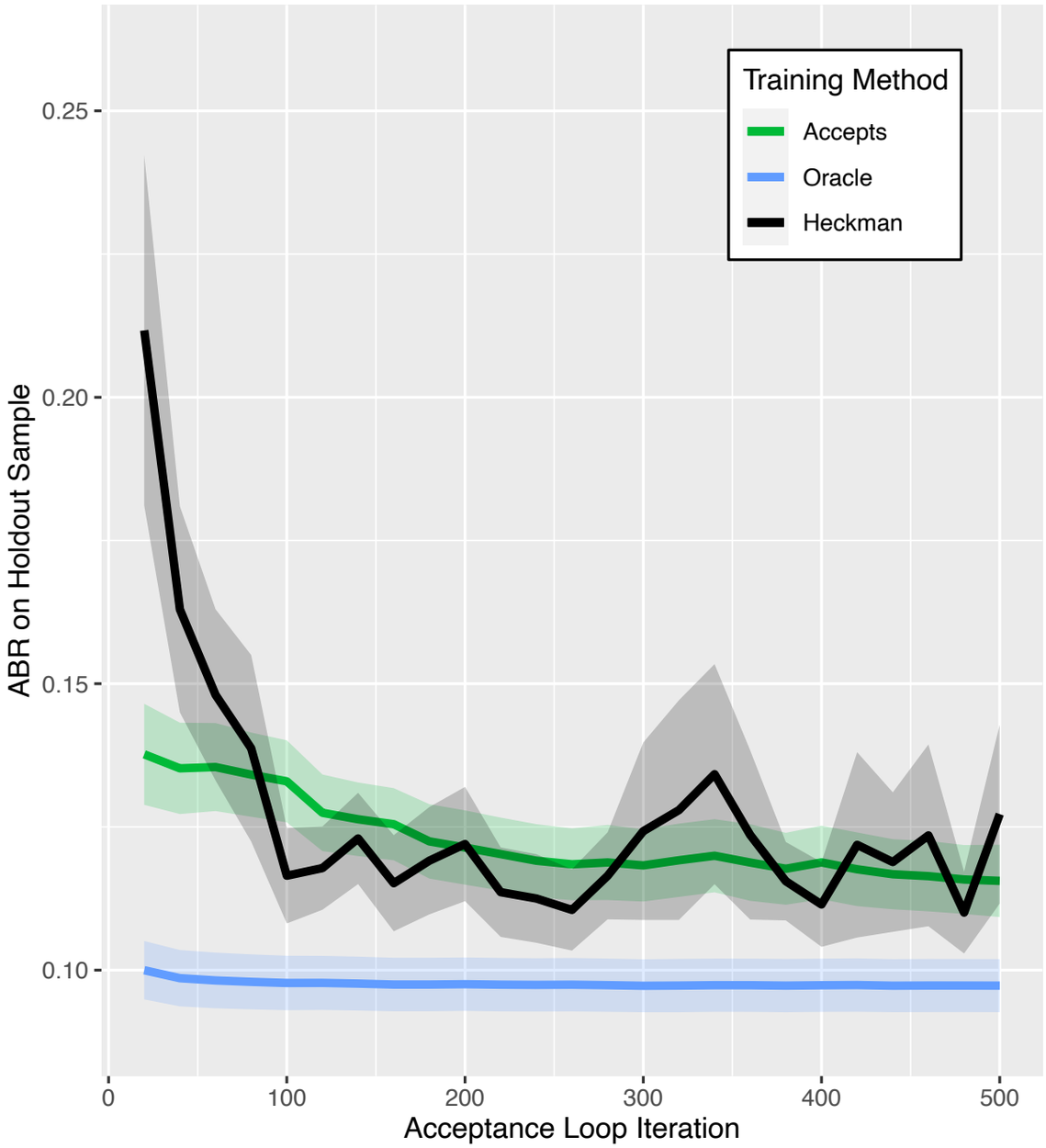
- **bivariate binary model with Heckman-type sample selection**
  - developed by (Meng & Shmidt, 1985)
  - applied to credit scoring (e.g, Banasik & Crook, 2007)
- **simultaneously estimates two equations:**
  - outcome equation: $\mathbf{P}(\text{BAD}) = X\beta + \varepsilon_1$
    - estimated on **accepts** only
  - selection equation: $\mathbf{P}(\text{accepted}) = X\gamma + \varepsilon_2$
    - estimated on both **accepts** and **rejects**
    - highly correlated with the outcome equation
    - uses the same set of applicant features
- **yields consistent estimates when** $\rho(\varepsilon_1, \varepsilon_2) \neq 0$

Model variants:

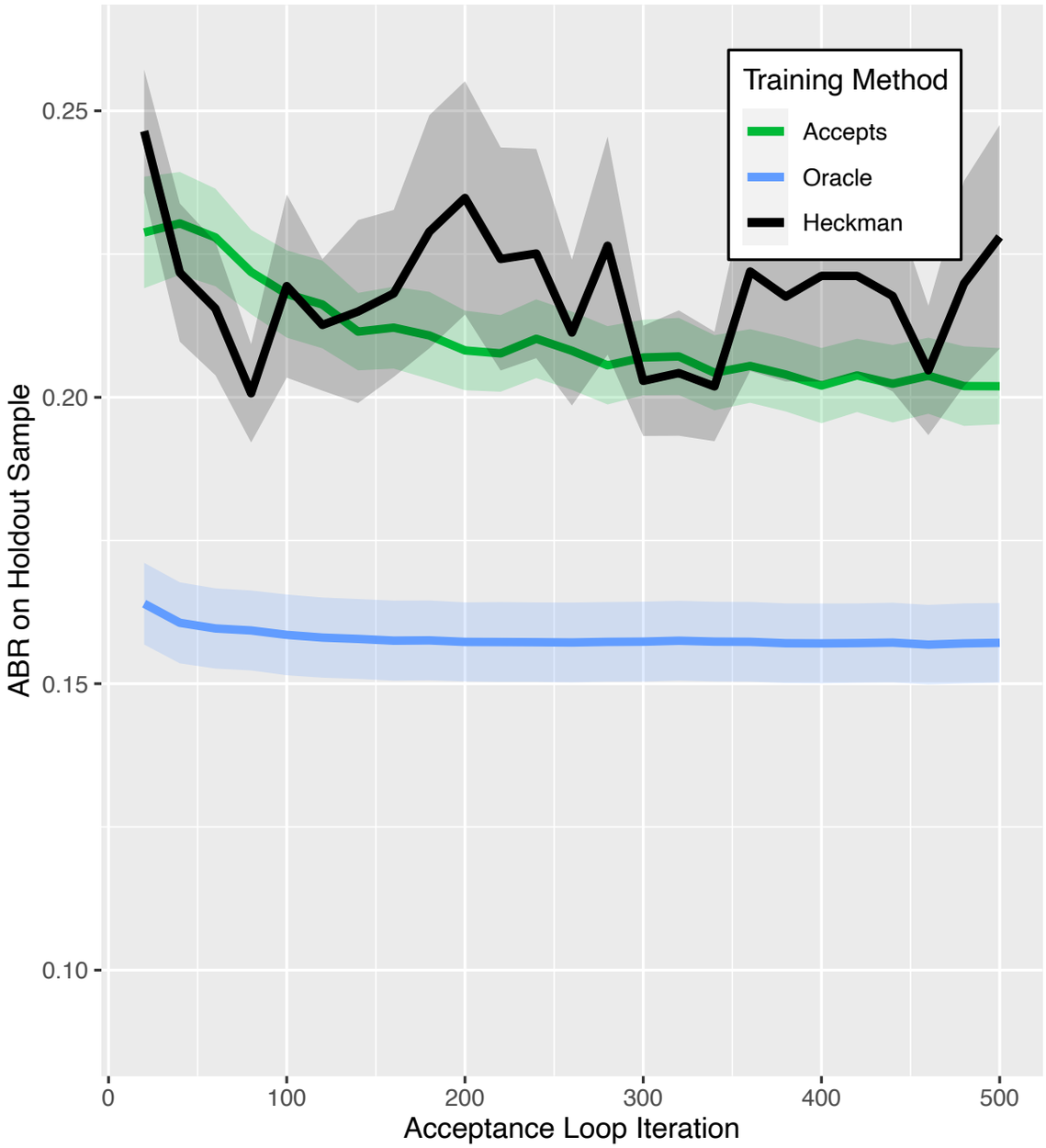
- **outcome & selection equation:** probit, logistic or linear (outcome only)
- **estimation:** maximum likelihood, two-stage method
- **nonlinearity:** with / without regression splines

Heckman on Synthetic Data

Data with
single Gaussians

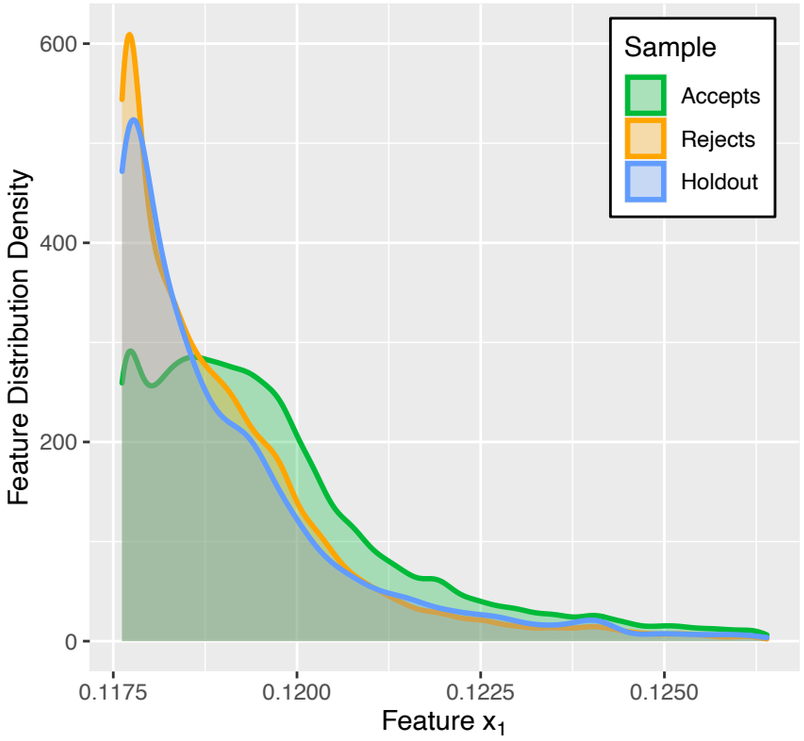


Data with
Gaussian mixtures

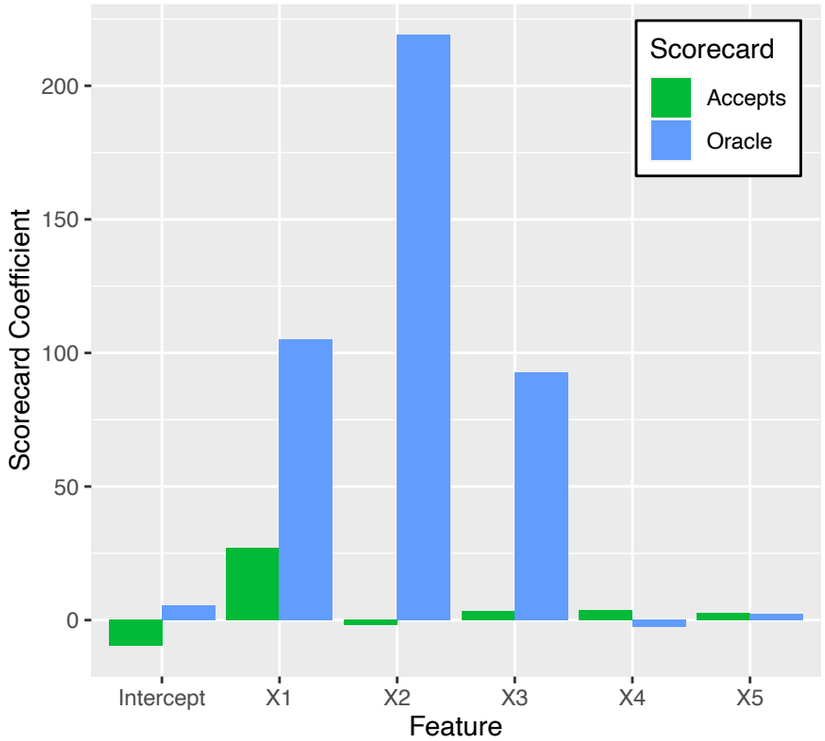


Sampling Bias on Real Data

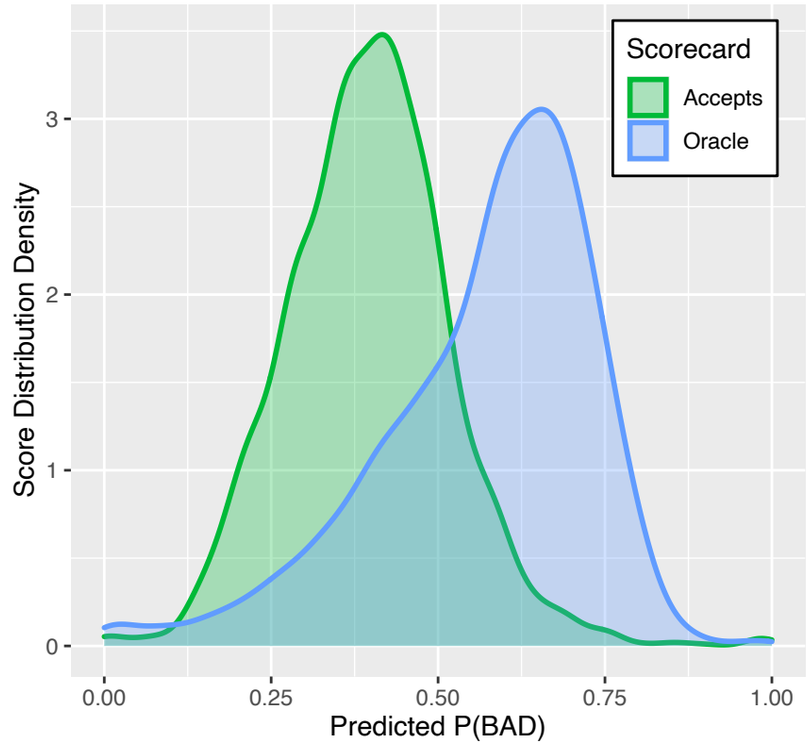
(a) Bias in Data



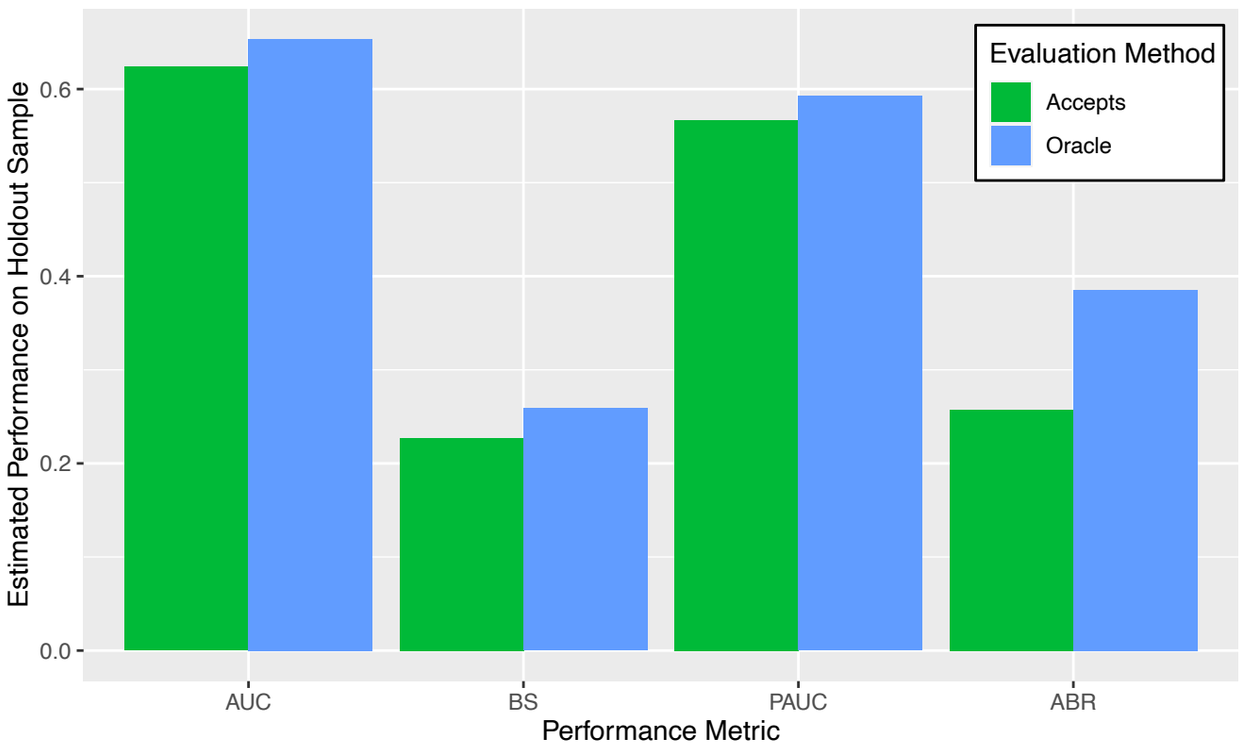
(b) Bias in Model



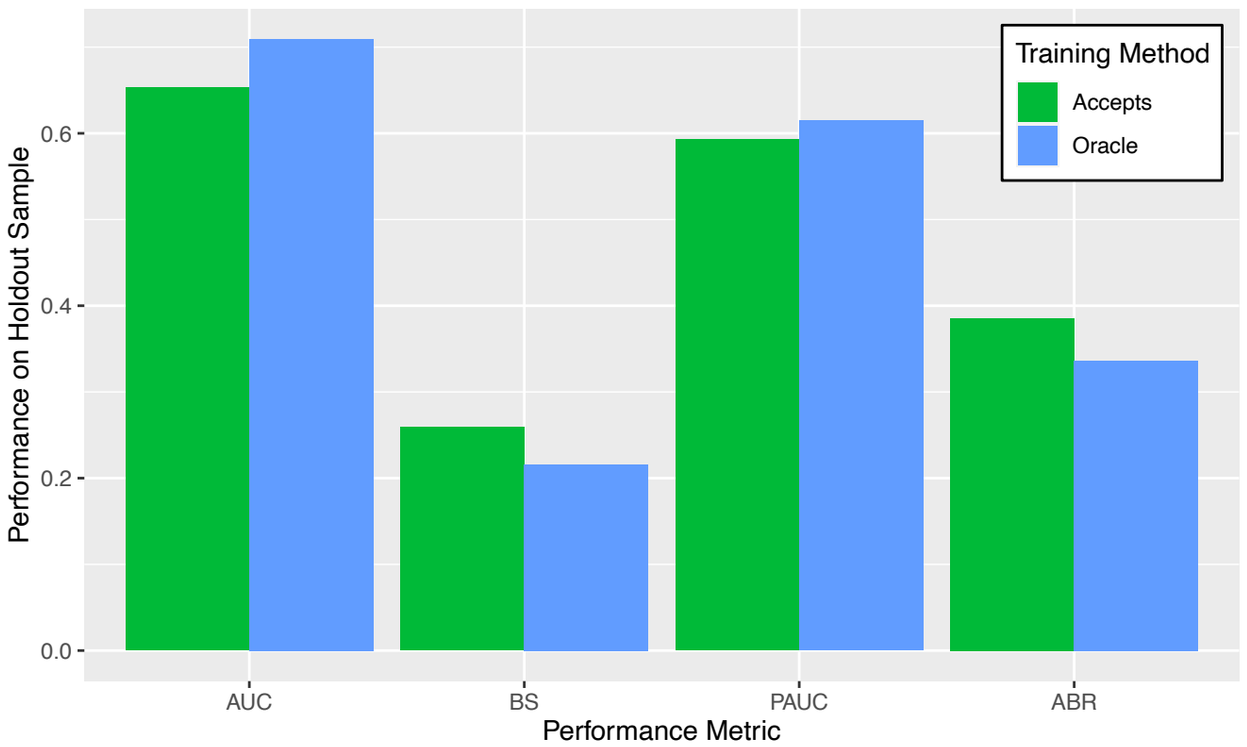
(c) Bias in Predictions



(d) Impact of Bias on Evaluation



(e) Impact of Bias on Training



Business Impact with Variable LGD

Parameters:

- **acceptance rate:** varied over reasonable range
- **principal and interest:** fixed for two markets
- **loss given default:** varied from 0 to 100%

Two markets:

- **micro-loans**
- **installment loans**

	Micro-loans	Installment loans
Acceptance rate α	[20%, 40%]	[10%, 20%]
Loan principal A	\$375 (SD = \$100)	\$17,100 (SD = \$1,000)
Total interest i	.1733 (SD = .01)	.1036 (SD = .01)
Loss given default L	[0, 1%, …, 100%]	

Calculations:

- **average profit per loan for each algorithm:**

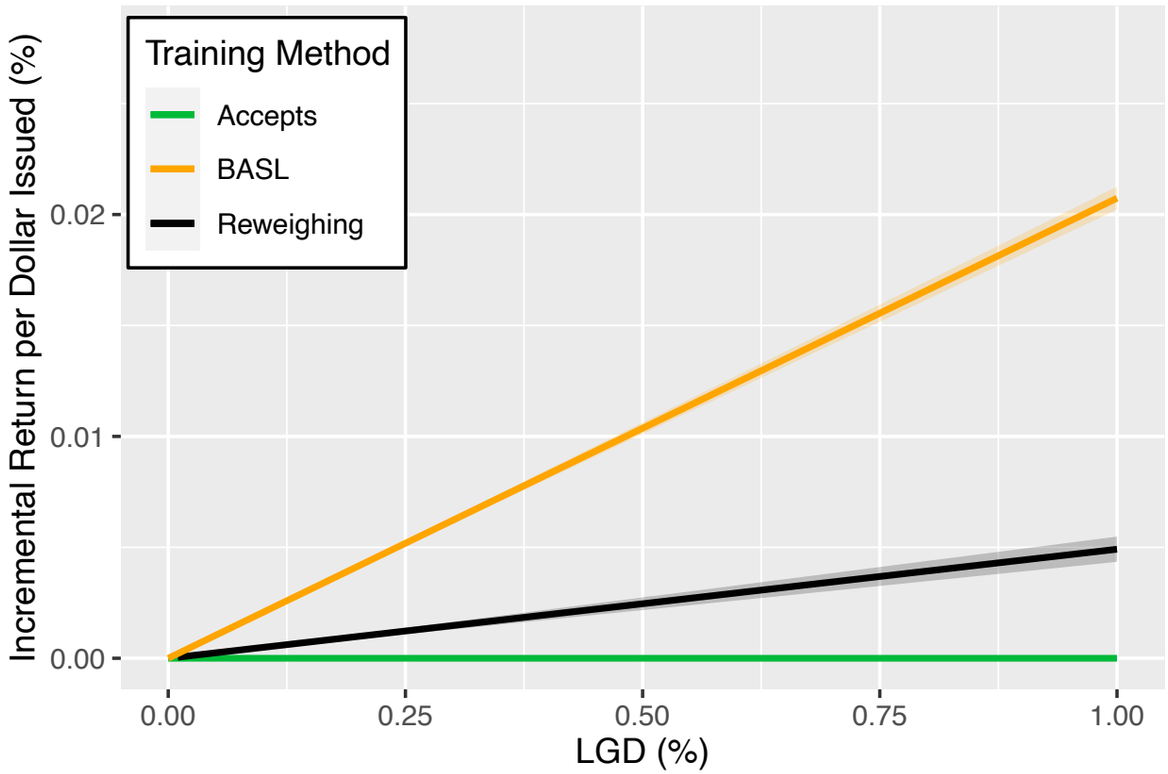
$$\pi = \frac{1}{100}\sum_{j=1}^{100}\Big[\underbrace{\mathrm{ABR}_j \times A \times (1+i) \times (1-L)}_{\text{BAD clients}} + \underbrace{(1-\mathrm{ABR}_j) \times A \times (1+i)}_{\text{GOOD clients}} - A\Big]$$

- **averaging over 100 ABR values** (4-fold CV x 25 bootstrap samples)

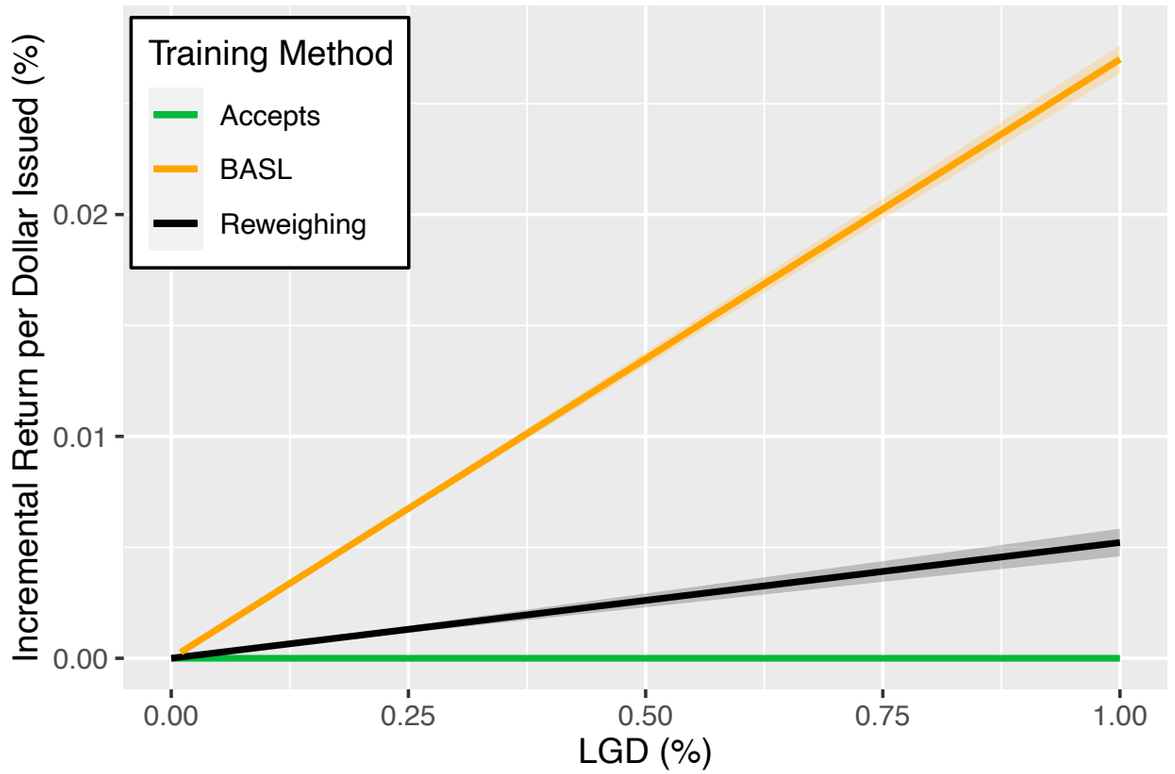
Business Impact with Variable LGD

- **comparing our framework and its strongest competitor**
- **measuring incremental profit per loan over ignoring rejects**

(a) Microloans



(b) Installment Loans

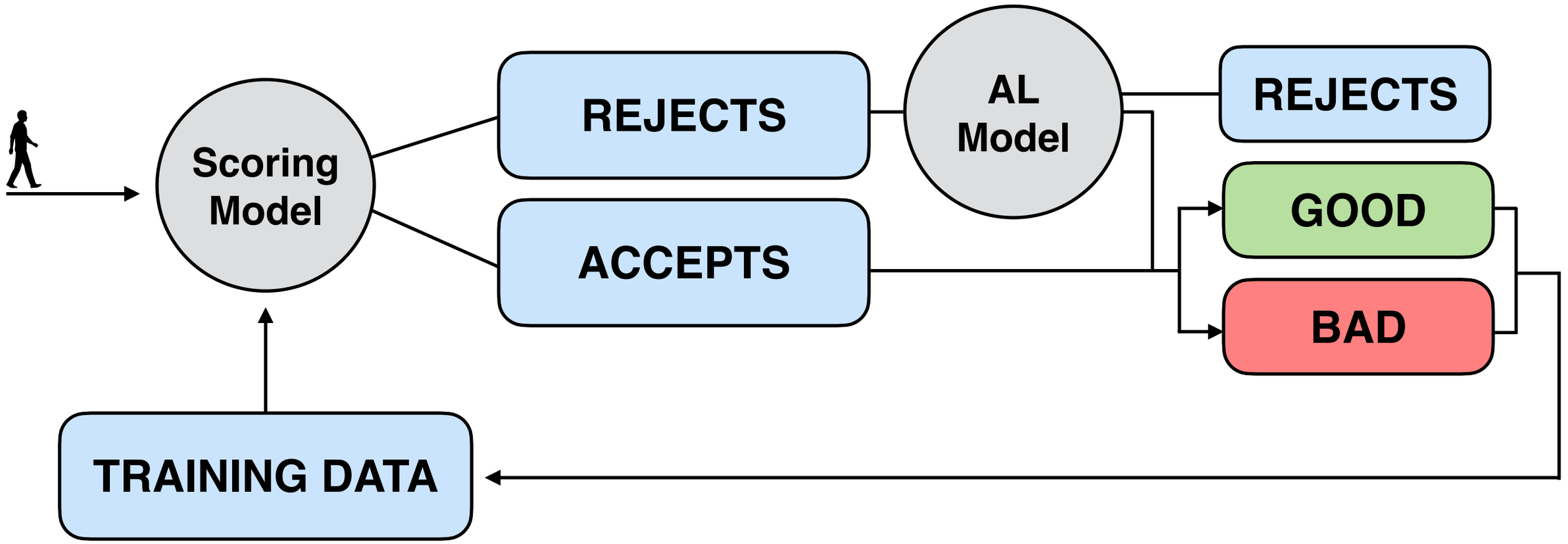


Incremental monetary gains:

- micro-loans: up to **$7.78** per loan
- installment loans: up to **$461.70** per loan

A7. ACTIVE LEARNING

Acceptance Loop with AL



- **scoring model filters incoming loan applications**
  - **ML model** observes features of incoming applicants
  - predicts whether an applicant will repay the loan
- **active learning selects additional cases rejected by a scorecard**
  - **AL model** observes features of rejects and scorecard predictions
  - predicts whether an applicant will be «useful»

Selected AL Techniques

Uncertainty sampling:

- selects observations that the ML model is **least confident about**
- e.g., cases with predicted **P(BAD)** close to 0.5

Query-by-committee (QBC):

- trains a set (committee) of ML models (e.g., on different training folds)
- selects observations where the committee **disagrees the most**
- e.g., cases with the highest Kullback-Leibler divergence over predictions

Optimized probabilistic active learning (OPAL):

- measures **«spatial usefulness»** of an unlabeled observation
- selects observations that maximize the expected reduction in (asymmetric) misclassification cost
- e.g., cases from high-density areas with potentially higher error costs

Experimental Setup

Acceptance loop:

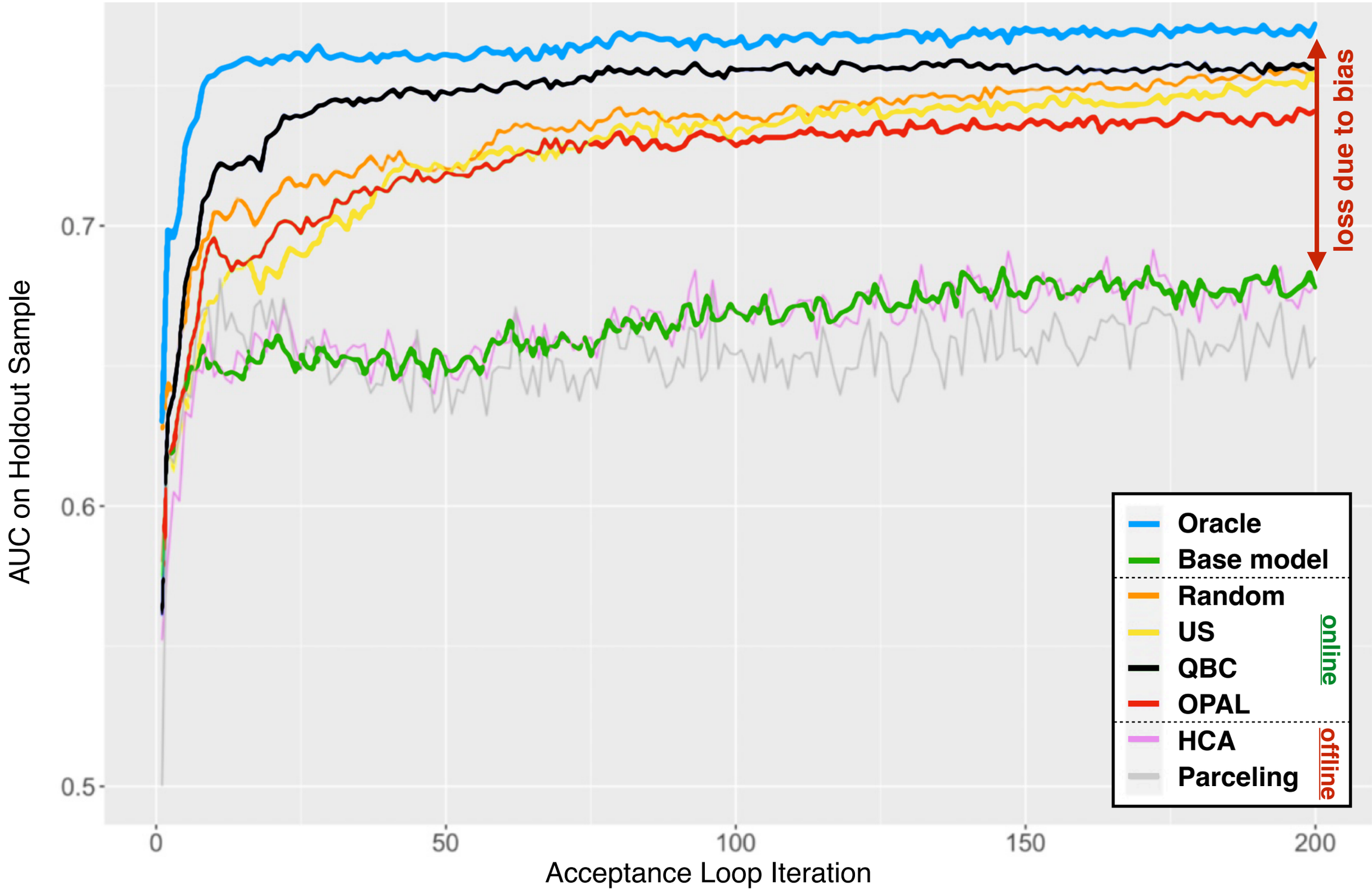
- draw / generate a batch of **new applications**
- **accept a subset** of loan applications
  - select 20% low-risk cases with **ML model**
  - select 10% «useful» cases with **AL model**
- **augment training data** with labeled accepts
- **retrain the scoring model** on new data
- **evaluate** performance on a holdout sample

repeat for 200 iterations

Performance evaluation:

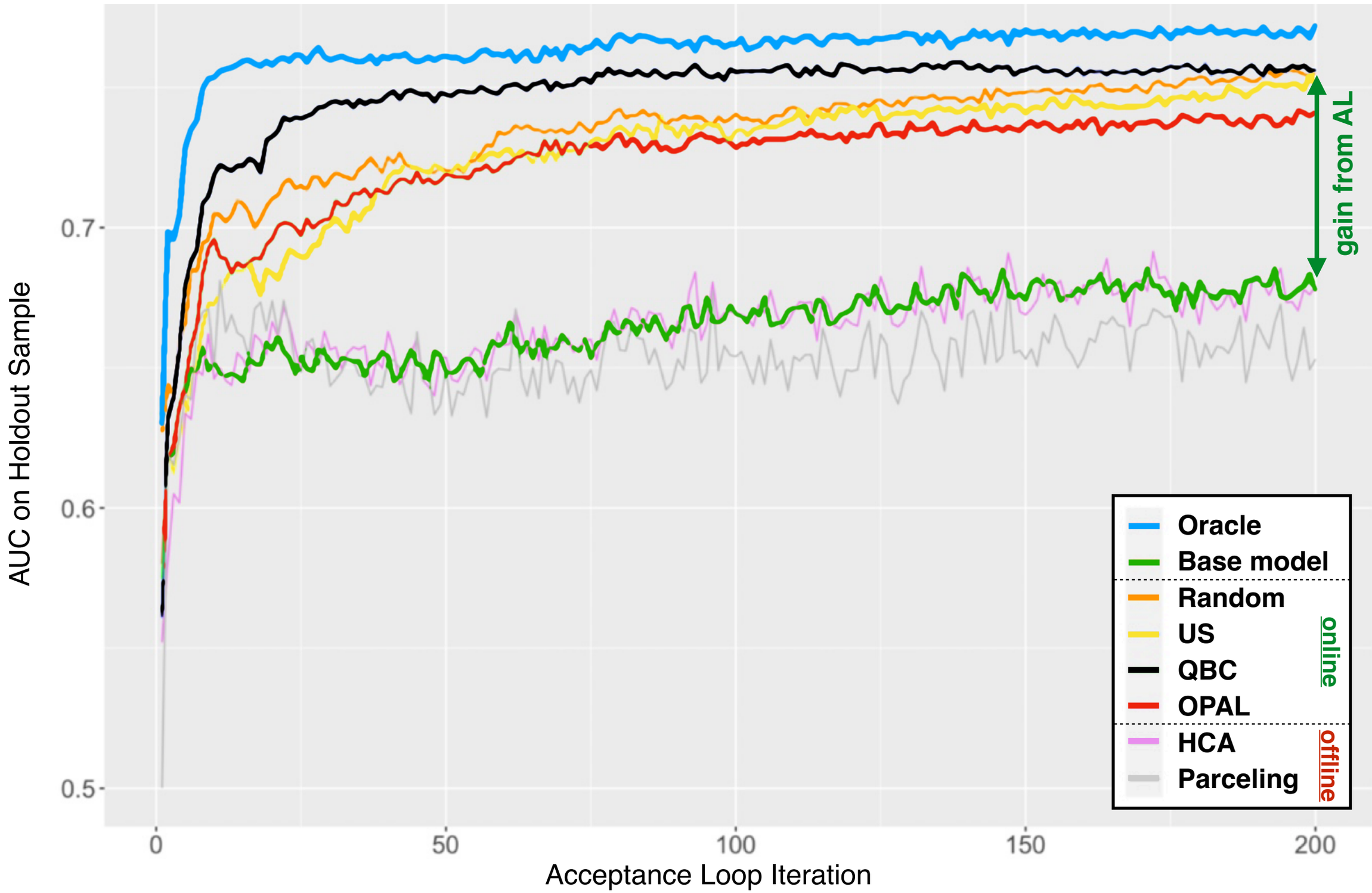
- **two cost / benefit components compared to base model:**
  - **model performance:** improved accuracy of the retrained **ML model**
  - **data augmentation:** accepting extra applicants with the **AL model**

Results: LendingClub [2/3]



AUC = area under the ROC curve; higher is better

Results: LendingClub [3/3]



AUC = area under the ROC curve; higher is better

Results: Overall Profit [2/2]

LendingClub Dataset

Method	Data profit	Model profit	Total profit
Random	.124	.000	.125
US	.132	**.001**	.133
QBC	**.154**	**.001**	**.155**
OPAL	.095	**.000**	.096
Oracle	1.271	.002	1.272

Synthetic Dataset

Method	Data profit	Model profit	Total profit
Random	-.098	.002	-.095
US	-.115	**.003**	-.112
QBC	**-.040**	**.003**	**-.036**
OPAL	-.167	**.003**	-.163
Oracle	-1.068	.005	-1.062

- **data profit** = profit from assigning loans to applicants selected with AL
- **model profit** = profit from model improvement after data augmentation
- values represent average profit per EUR issued

Summary

- **AL improves performance and profitability of credit scorecards**
  - positive gains in different performance metrics
  - query-by-committee demonstrates most potential
- **trade-off between labeling cost and model improvement**
  - labeling cost can outweigh the model improvement
  - percentage of labeled cases is an important meta-parameter
  - when to stop labeling?
- **further experiments needed to clarify the potential of AL**
  - strong impact of the data characteristics on costs & benefits
  - in which environments AL is useful?