HyperImpute:

Generalized Iterative Imputation with Automatic Model Selection

ICML2022

D. Jarrett*, B. Cebere*, T. Liu, A. Curth,
M. van der Schaar

van_der_Schaar
\ LAB
vanderschaar-lab.com

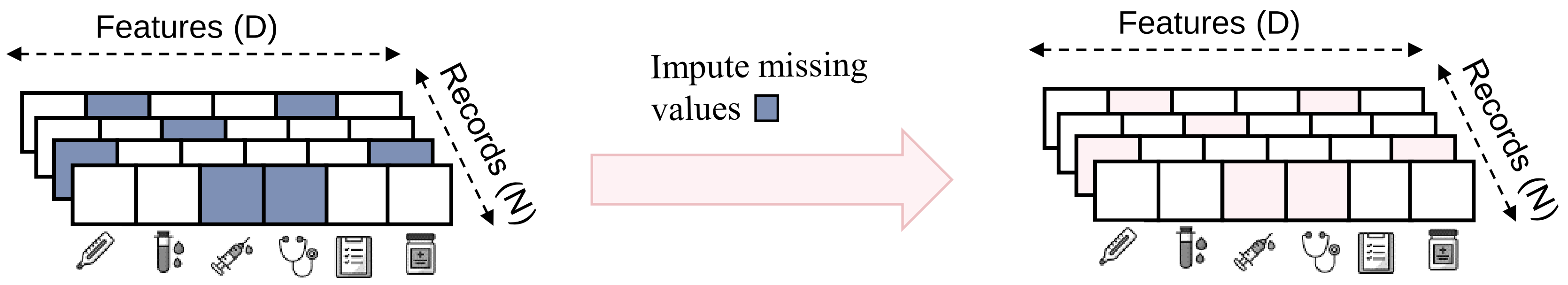
UNIVERSITY OF CAMBRIDGE

amc253@cam.ac.uk
@AliciaCurth
linkedin.com/in/alicia-curth

Setting: Missing data imputation

- **Motivation:** Missing data is a ubiquitous problem in real-life data collection
  - E.g. unrecorded patient characteristics, missing lab values
- **Goal:** Want to *know* likely values (regardless of downstream task)

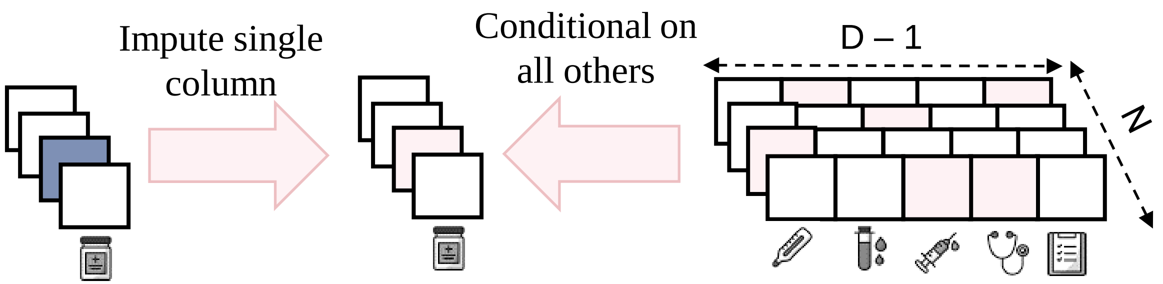
  → *Impute* values even when *no columns are complete*



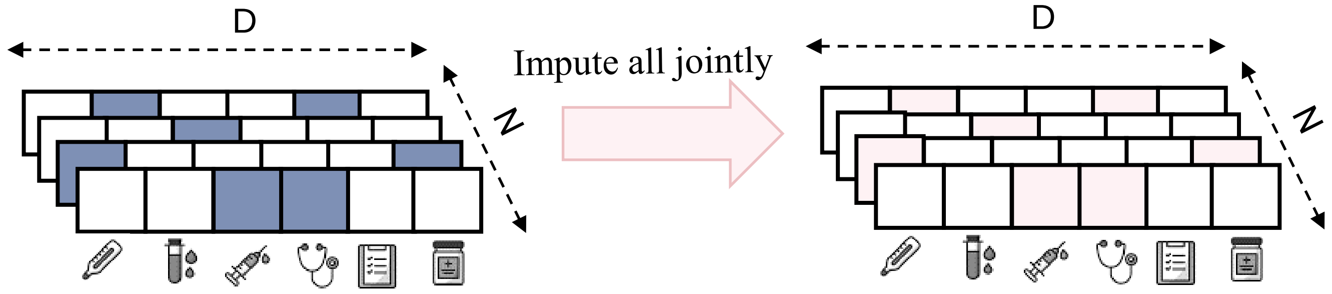
Related work: Much interest in imputation recently

Existing work relies on:

a) Iterative Imputation (*Imputation by chained equations – ICE*) with correctly pre-specified per-column (prediction) models *OR*

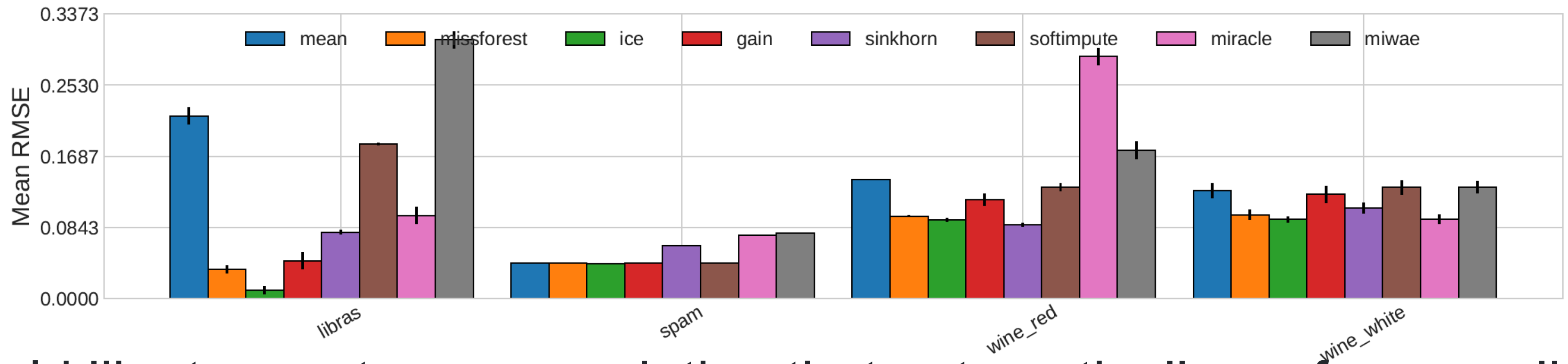


b) Deep Generative Models (*GAIN, MIWAE, MIRACLE*) as joint models for all features



Goal & desiderata for our new approach

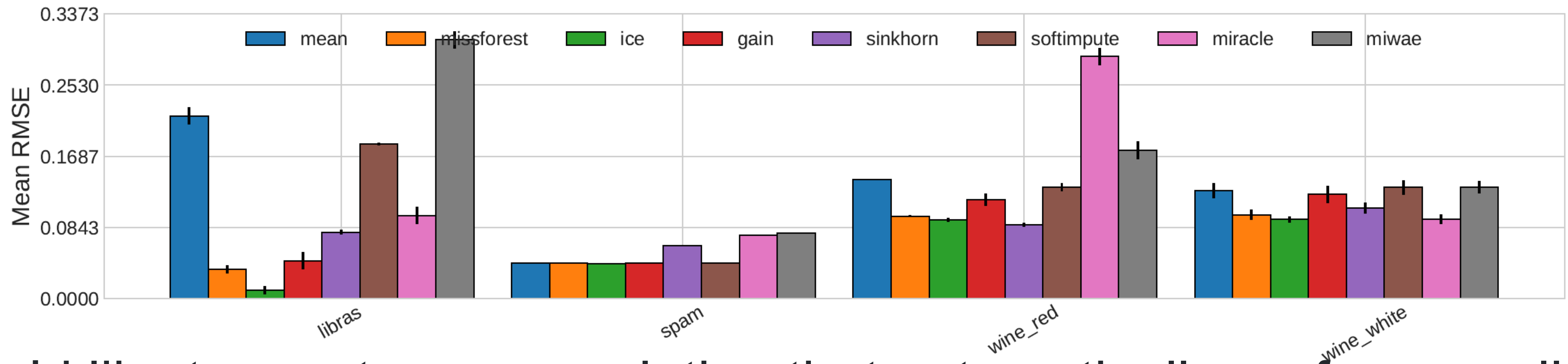
Different methods do well across different datasets and settings…



… so we'd like to create a new solution that automatically performs well in any scenario!

Goal & desiderata for our new approach

Different methods do well across different datasets and settings…



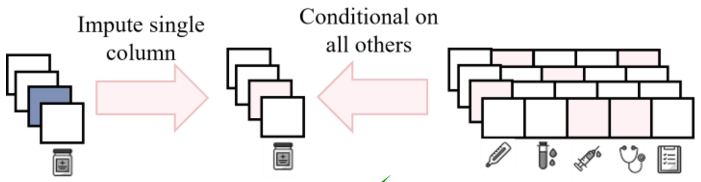
… so we'd like to create a new solution that automatically performs well in any scenario!

Three desiderata for our new solution:

1. *Weak assumptions:* Trainable without complete data, but not assume completely random missingness pattern
2. *Flexibility:* Combine flexibility of conditional specifications with capacity of deep approximators
3. *Easy Optimization:* Relieve burden of complete specification, and be easily & automatically optimized (tunable)

HyperImpute: A generalized iterative imputation approach

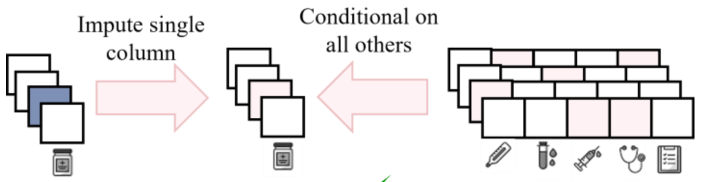
Why iterative imputation?



- *Weak assumptions*: Allows generic missing at random setting ✓
- *Flexibility:* Allows specifying different models for each column ✓
  → Easily incorporate different data-types and design-specifics, e.g. bounds
- *Easy Optimization:* column-wise perspective gives simple evaluation (prediction) criterion ✓

HyperImpute: A generalized iterative imputation approach

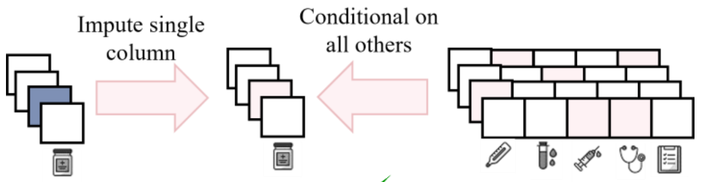
Why iterative imputation?



- *Weak assumptions*: Allows generic missing at random setting ✓
- *Flexibility:* Allows specifying different models for each column ✓
  → Easily incorporate different data-types and design-specifics, e.g. bounds
- *Easy Optimization:* column-wise perspective gives simple evaluation (prediction) criterion ✓ but search space is combinatorial in #columns ☹

HyperImpute: A generalized iterative imputation approach

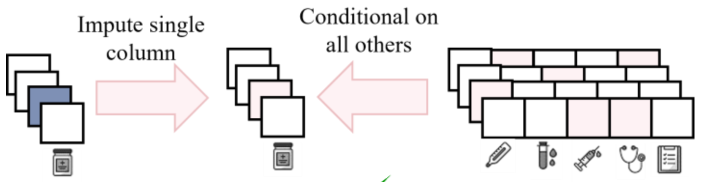
Why iterative imputation?



- *Weak assumptions*: Allows generic missing at random setting ✓
- *Flexibility:* Allows specifying different models for each column ✓
  → Easily incorporate different data-types and design-specifics, e.g. bounds
- *Easy Optimization:* column-wise perspective gives simple evaluation (prediction) criterion ✓ but search space is combinatorial in #columns ☹
  → Top-Down Search (across all combinations): intractable ☹

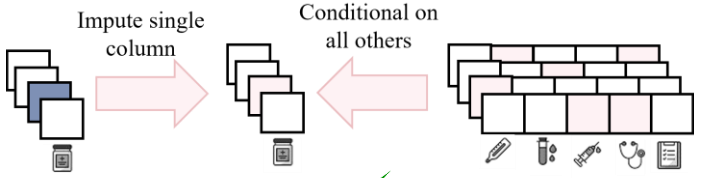
HyperImpute: A generalized iterative imputation approach

Why iterative imputation?



- *Weak assumptions*: Allows generic missing at random setting ✓
- *Flexibility:* Allows specifying different models for each column ✓
  → Easily incorporate different data-types and design-specifics, e.g. bounds
- *Easy Optimization:* column-wise perspective gives simple evaluation (prediction) criterion ✓ but search space is combinatorial in #columns ☹
  → Top-Down Search (across all combinations): intractable ☹
  → Concurrent Search (optimize columns in parallel): suboptimal ☹

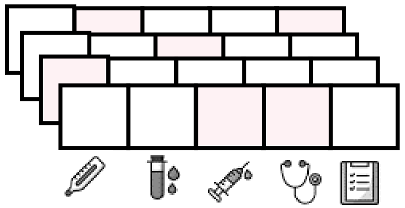
HyperImpute: A generalized iterative imputation approach



Why iterative imputation?

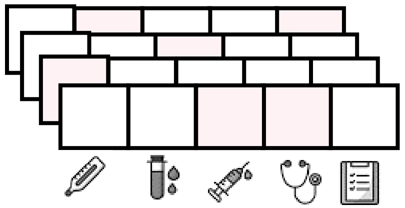
- *Weak assumptions*: Allows generic missing at random setting ✓
- *Flexibility:* Allows specifying different models for each column ✓
  → Easily incorporate different data-types and design-specifics, e.g. bounds
- *Easy Optimization:* column-wise perspective gives simple evaluation (prediction) criterion ✓ but search space is combinatorial in #columns ☹
  → Top-Down Search (across all combinations): intractable ☹
  → Concurrent Search (optimize columns in parallel): suboptimal ☹
  → HyperImpute: Iterative model search ☺ ✓

The HyperImpute Framework

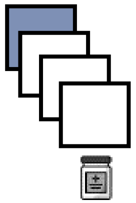


Given current
imputed dataset $\hat{\mathcal{D}}$

The HyperImpute Framework

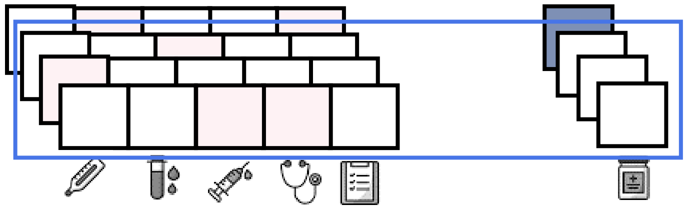


Given current imputed dataset $\hat{\mathcal{D}}$



1 Find best model for next column d

The HyperImpute Framework

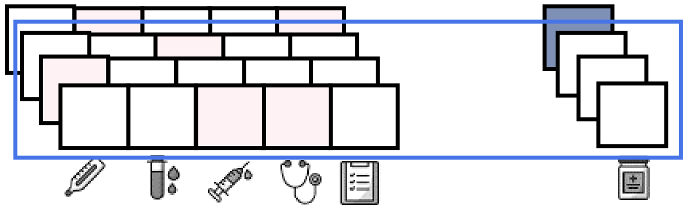


Using records $\mathcal{D}_d^{\text{obs}}$
with observed target

Given current
imputed dataset $\hat{\mathcal{D}}$

1 Find best model
for next column d

The HyperImpute Framework



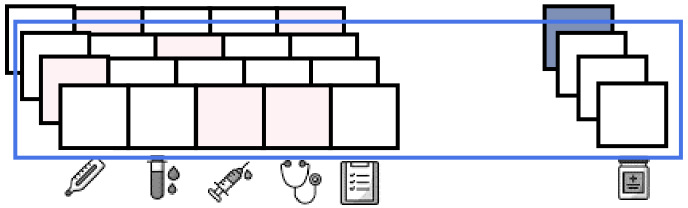
Using records $\mathcal{D}_d^{\text{obs}}$
with observed target

Given current
imputed dataset $\hat{\mathcal{D}}$

1 Find best model
for next column d

$$a_d \leftarrow \texttt{ModelSearch}(\mathcal{D}_d^{\text{obs}}, \hat{\mathcal{D}}_{-d}^{\text{obs}}, \mathcal{A})$$

The HyperImpute Framework



Using records $\mathcal{D}_d^{\text{obs}}$
with observed target

Given current
imputed dataset $\hat{\mathcal{D}}$

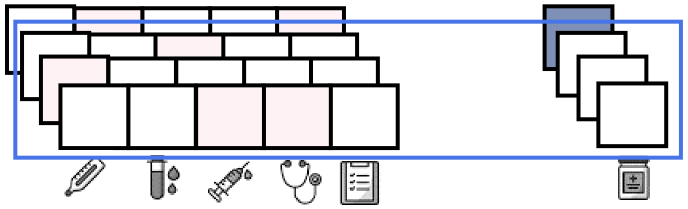
1 Find best model
for next column d

$$a_d \leftarrow \texttt{ModelSearch}(\mathcal{D}_d^{\text{obs}}, \hat{\mathcal{D}}_{-d}^{\text{obs}}, \mathcal{A})$$

Optimizer

- Naive Search
- Bayesian Opt.
- Hyperband Opt.
- ...

The HyperImpute Framework

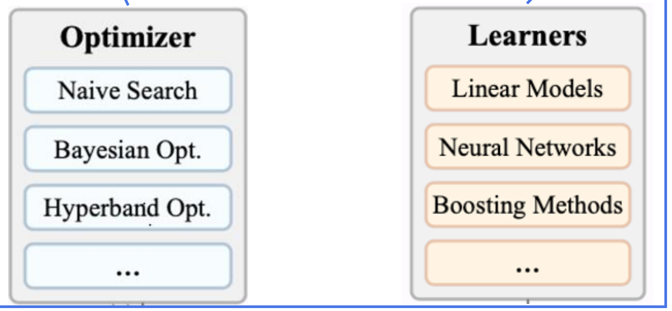


Using records $\mathcal{D}_d^{\text{obs}}$
with observed target

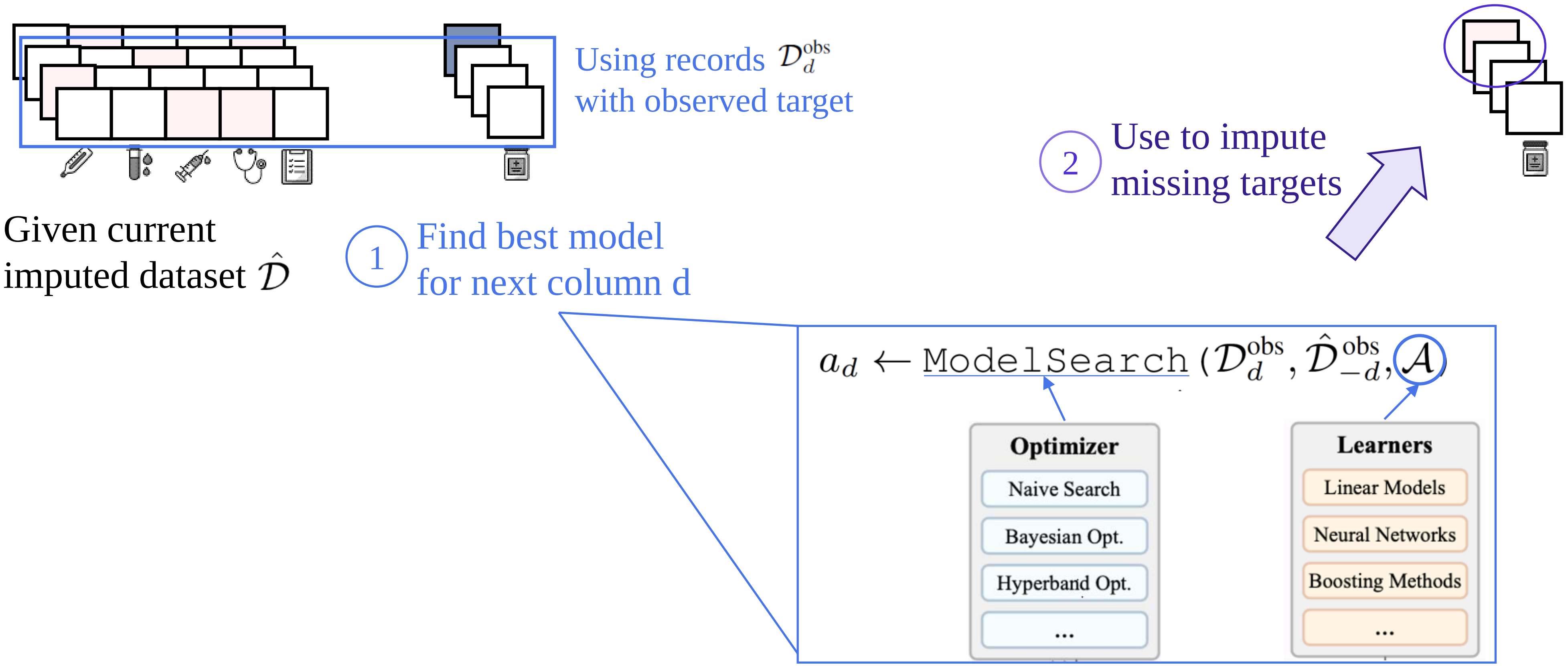
Given current
imputed dataset $\hat{\mathcal{D}}$

1 Find best model
for next column d

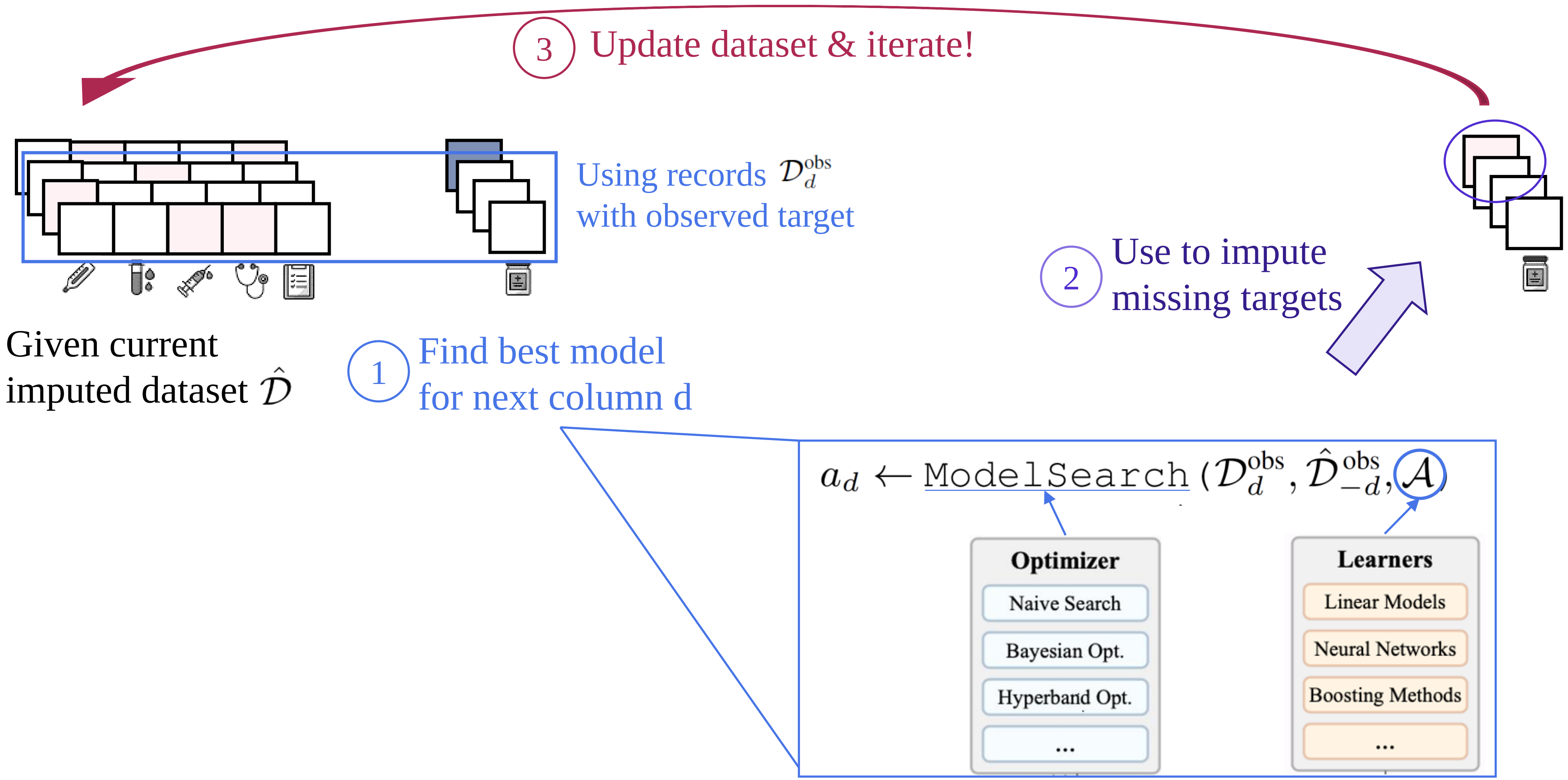
$$a_d \leftarrow \texttt{ModelSearch}(\mathcal{D}_d^{\text{obs}}, \hat{\mathcal{D}}_{-d}^{\text{obs}}, \mathcal{A})$$



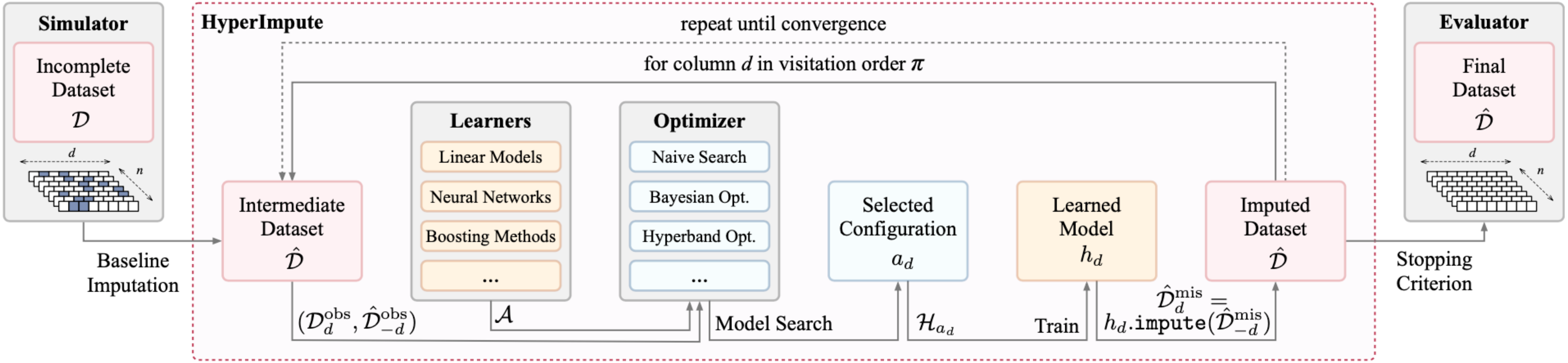
The HyperImpute Framework



The HyperImpute Framework



The HyperImpute package: A useful tool!

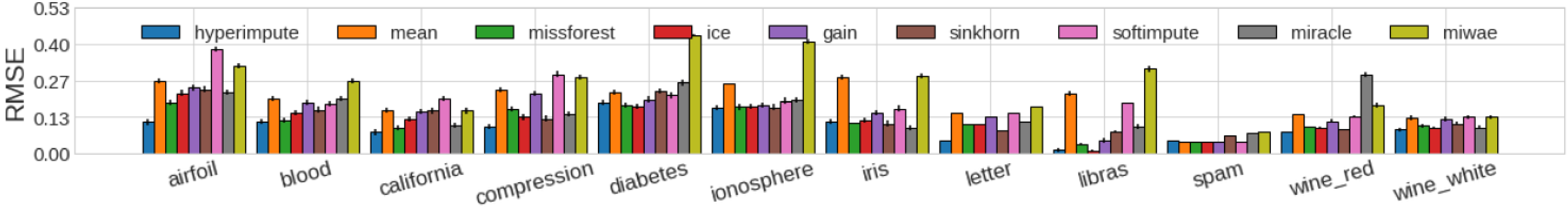


Implemented as an easy-to-use modular sklearn-style python package, incl. baselines and evaluation tools!

→ Available at https://github.com/vanderschaarlab/hyperimpute

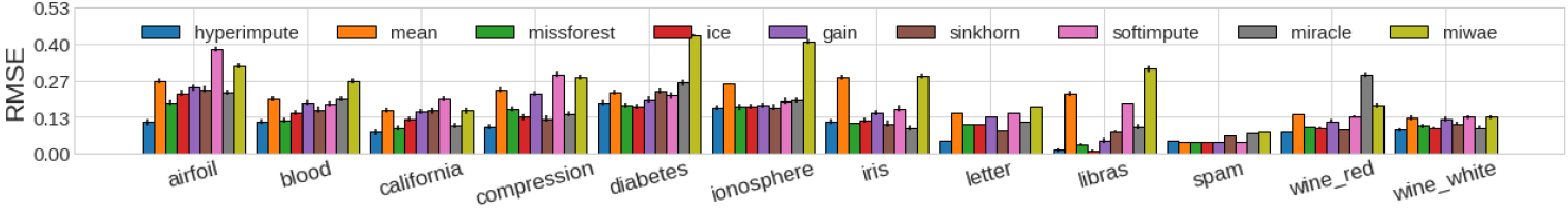
Empirical investigation

HyperImpute outperforms existing baselines across different datasets and missingness mechanisms..

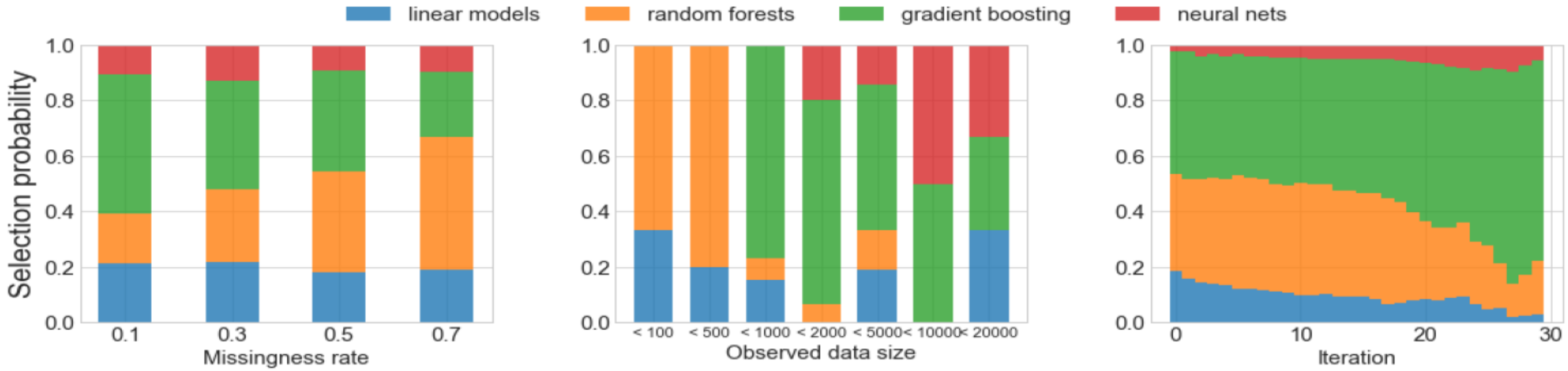


Empirical investigation

HyperImpute outperforms existing baselines across different datasets and missingness mechanisms..

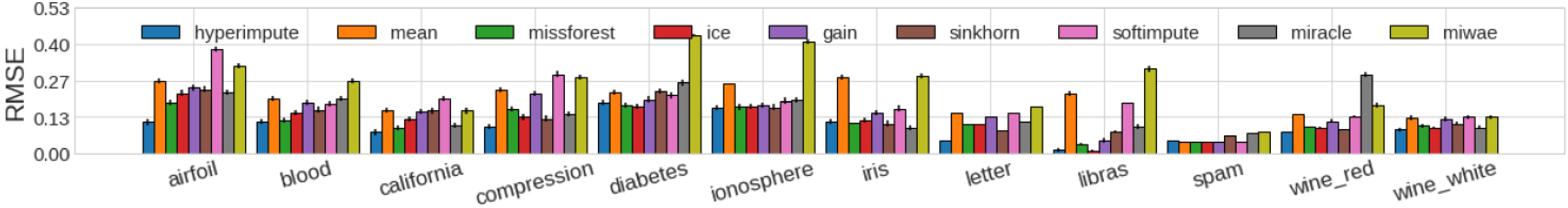


… and allows to provide interesting insights into selected model classes!

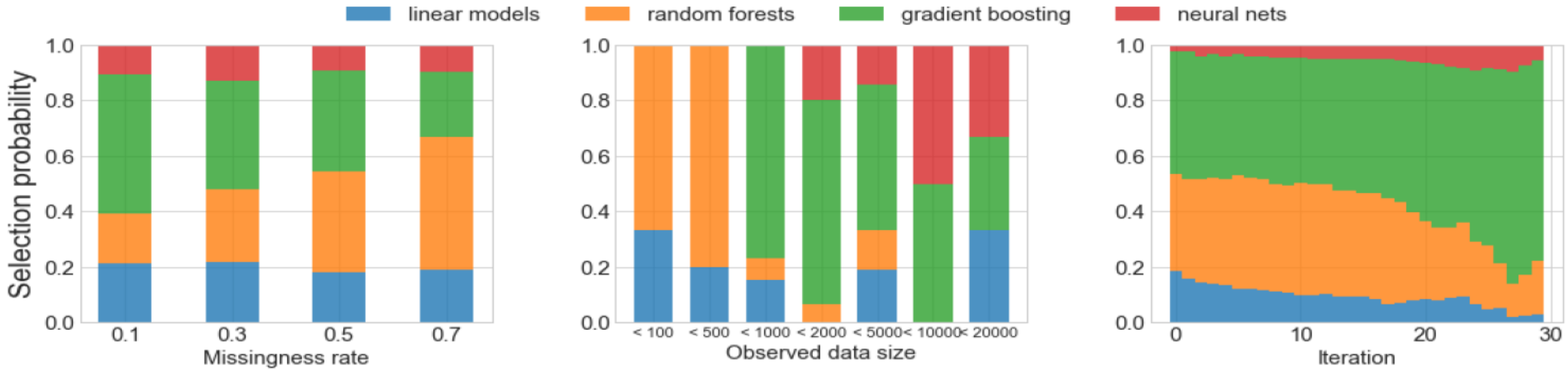


Empirical investigation

HyperImpute outperforms existing baselines across different datasets and missingness mechanisms..



… and allows to provide interesting insights into selected model classes!



Sources of gain, convergence and other scenario analyses covered in full paper!

Try it out!

Code: https://github.com/vanderschaarlab/hyperimpute
Paper: https://arxiv.org/abs/2206.07769

This work was supported by:

AstraZeneca

NSF

Alzheimer's Research UK | Make breakthroughs possible

Office of Naval Research Science & Technology