AutoML:
From Full Automation to A Human-Centric Approach

Prof. Marius Lindauer

Folien

Rise of Literacy

Photo by Anna Hunko on Unsplash

- Only priests were able to read and write
- People believed that they don't need to read and write
- They went to the holy buildings

- Today, everyone can read and write
- No one doubts the benefits of it

- **⇒ Democratization of literacy**

Inspired by Andrew Ng

Rise of AI Literacy?

Photo by Max Duzij on Unsplash

- Only highly educated people can program new AI applications
- Power lies only with the large IT companies

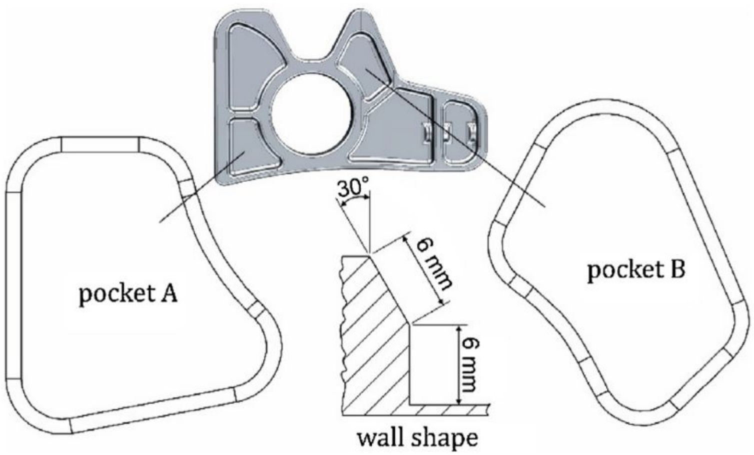
- In an age of limited resources, the need for efficient use gets more important

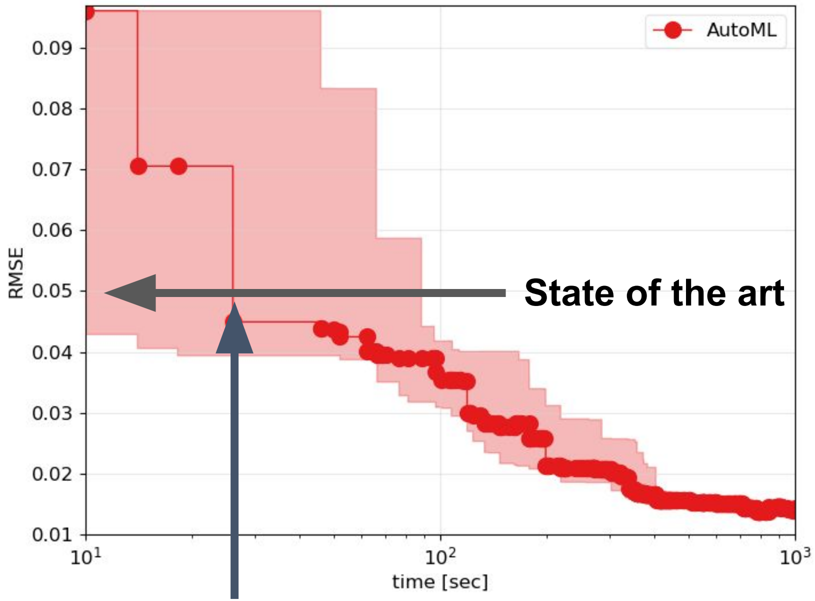
- **AutoML contributes to AI literacy!**

[See also my TEDx Talk]

Shape Error Prediction in Milling Process

[Denkena et al. SSRN'20]

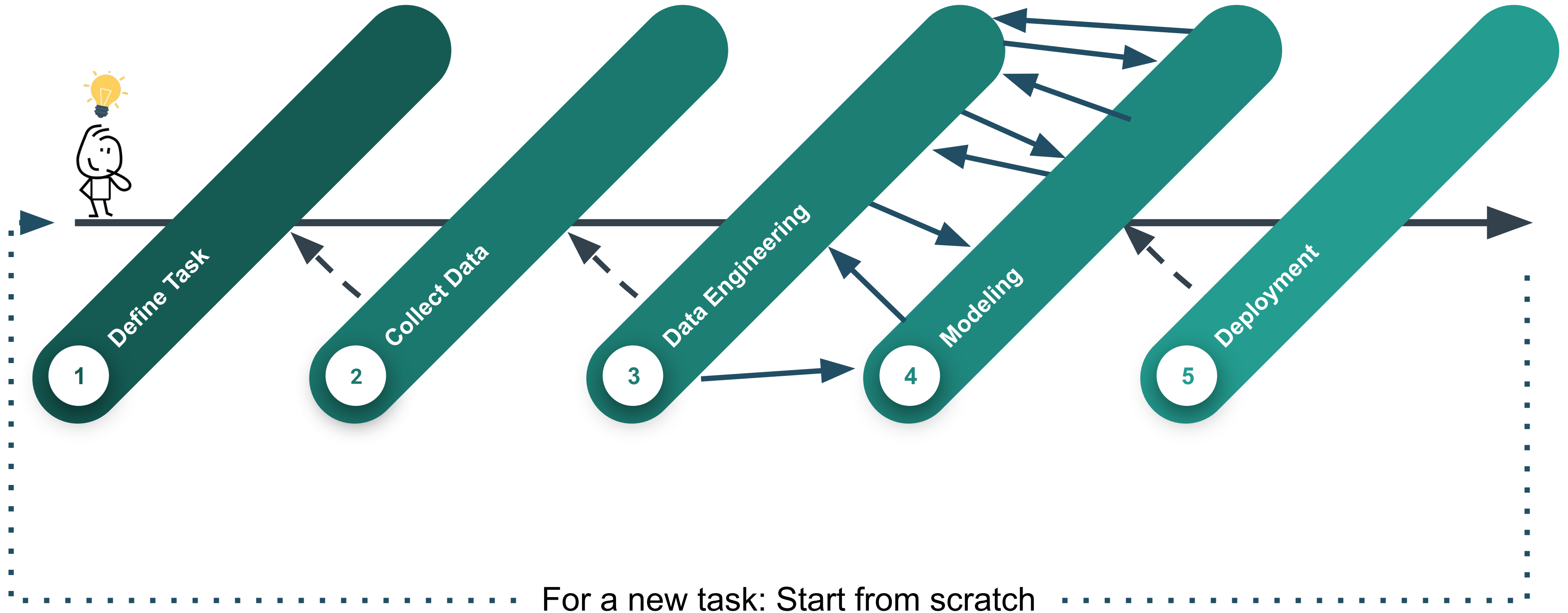




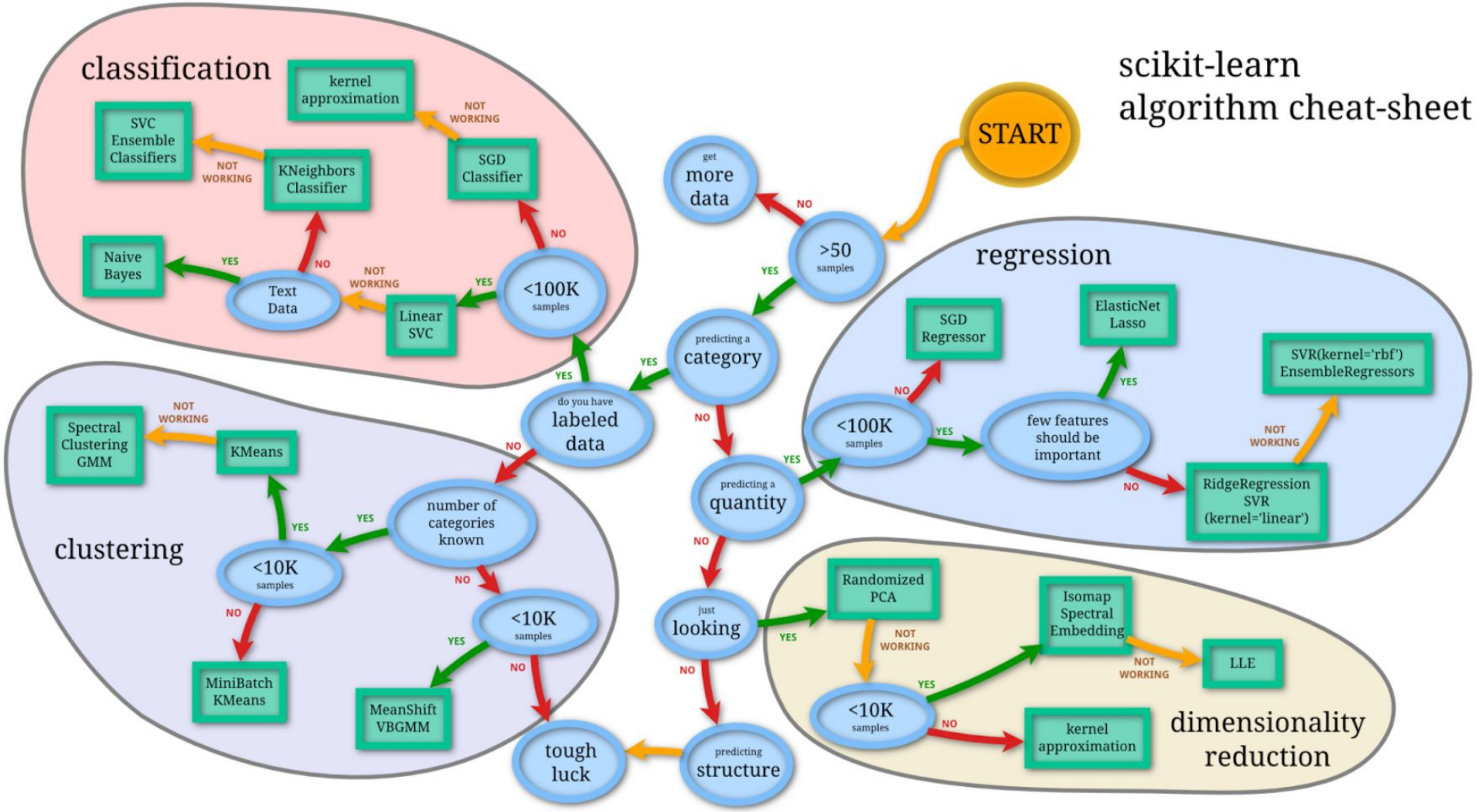
State of the art

Better than state of the art in less than 30sec!

Why does ML development take a lot of time?



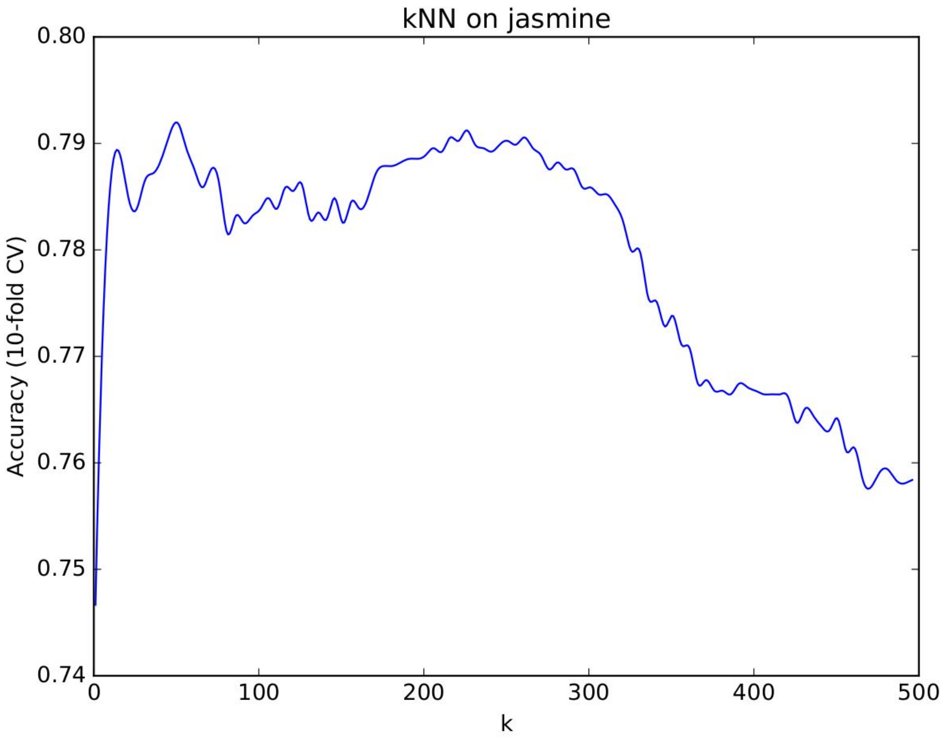
Design-Decisions?



source: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Toy Example: kNN

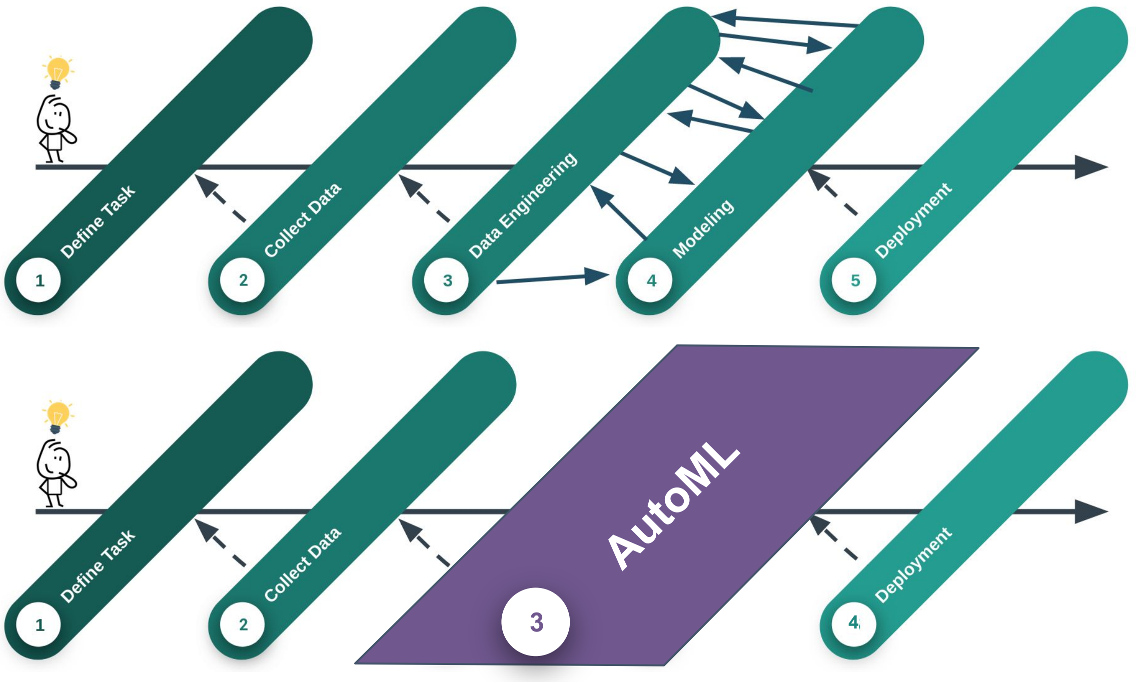
- *k*-nearest neighbors (kNN) is one of the **simplest ML algorithms**
- Size of neighbourhood (*k*) **is very important for its performance**
- The performance function depending on *k* is **quite complex** (not at all convex)



From ML Alchemy to Science

"You can teach an old dog new tricks" [Ruffinelli et al. 2020]
→ Hyperparameter optimization might not be the only required solution, but without it, it will also be hard.

ML vs AutoML



1
Define Task
2
Collect Data
3
Data Engineering
4
Modeling
5
Deployment
1
Define Task
2
Collect Data
3
AutoML
4
Deployment

Topics of AutoML

- **model selection** (e.g., Neural Architecture Search, ensembling)
- **configuration/tuning** (e.g., hyperparameter optimization via evolutionary algorithms, Bayesian optimization)
- **AutoML methodologies** (e.g., reinforcement learning, meta-learning, in-context learning, warmstarting, portfolios, multi-objective optimization, constrained optimization)
- **pipeline automation** (e.g., automated data wrangling, feature engineering, pipeline synthesis, and configuration)
- **automated procedures for diverse data** (e.g., tabular, relational, multimodal, etc.)
- **ensuring quality of results in AutoML** (e.g., fairness, interpretability, trustworthiness, sustainability, robustness, reproducibility)
- supporting **analysis and insight** from automated systems

Advantages

AutoML enables

- 🚀 More **efficient** research and development of ML applications
  - → AutoML has been shown to outperform humans on subproblems
- 🧮 More **systematic** research and development of ML applications
  - → no (human) bias or unsystematic evaluation
- 📋 More **reproducible** research
  - → since it is systematic!
- 🦾 **Broader use** of ML methods
  - → less required ML expert knowledge
  - → not only limited to computer scientists

Challenges

But, it is not that easy, because

🔁 Each dataset potentially requires **different optimal ML-designs**

→ Design decisions have to be made for each dataset again

⌛ Training of a single ML model can be **quite expensive**

→ We can not try many configurations

❓ Mathematical **relation** between design and performance is (often) **unknown**

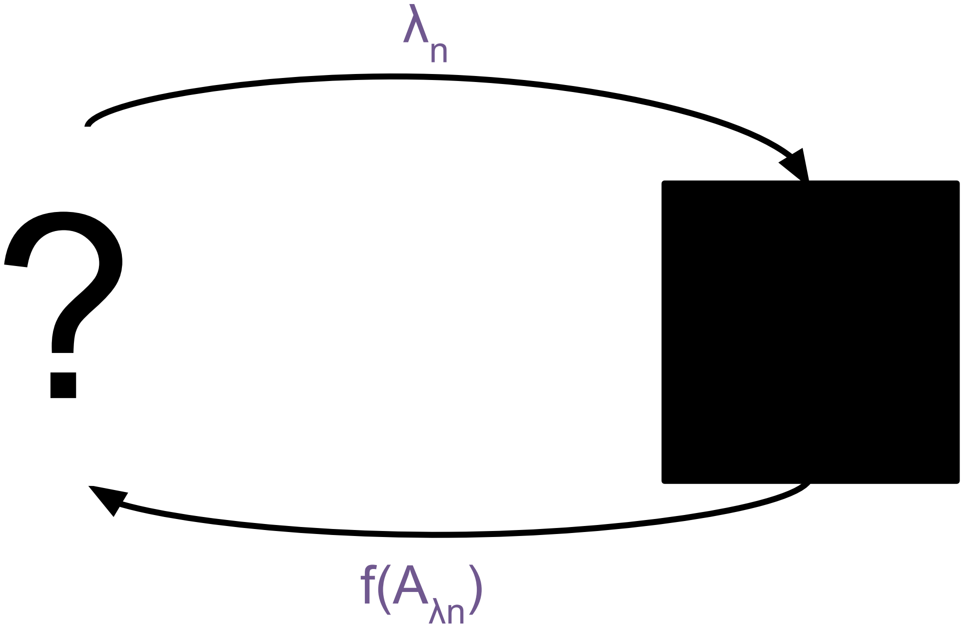
→ Gradient-based optimization not easily possible

🎛️ 🎚️ Optimization in **highly complex spaces**

→ including categorical, continuous and conditional dependencies

Some Basics on Hyperparamter Optimization

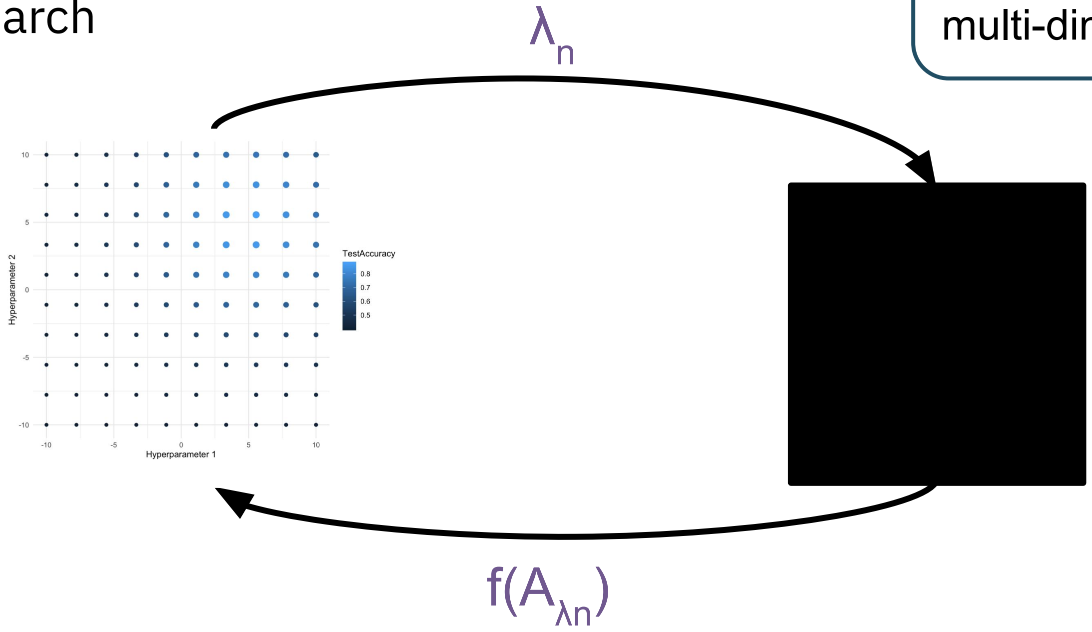
Black-Box Optimization Problem



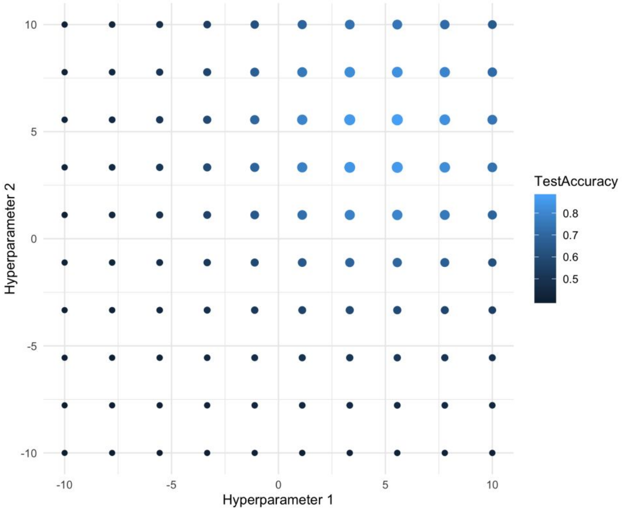
Option 1: Grid Search

a.k.a. exhaustive search;
a.k.a. advanced manual search

Popular technique: Evaluates all combinations on a pre-defined multi-dimensional grid



Option 1: Grid Search II



Advantages

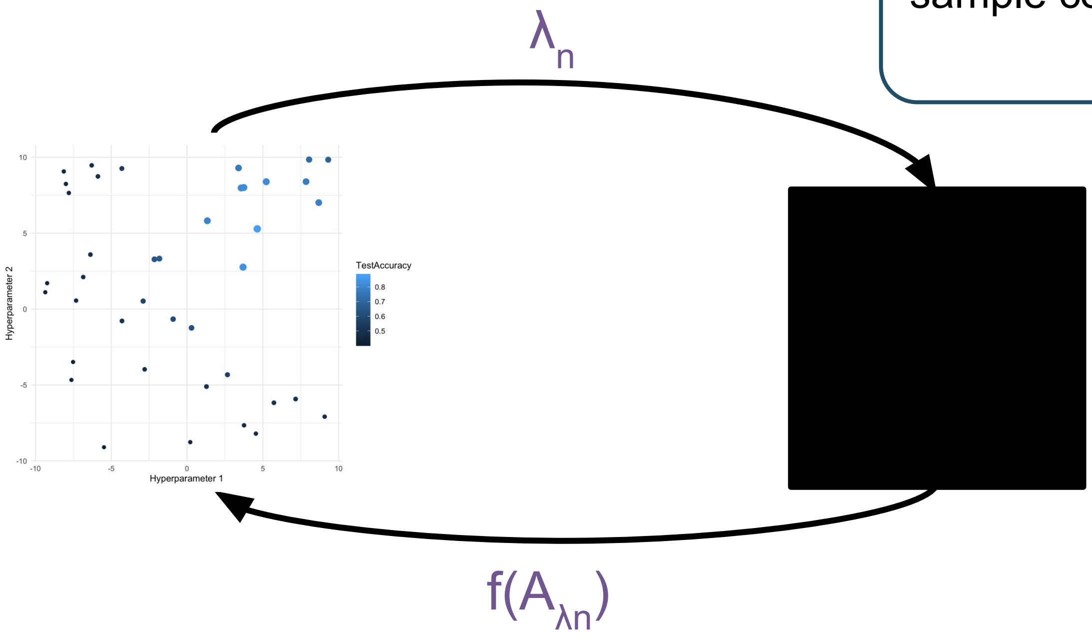
- Very easy to implement
- Very easy to parallelize
- Can handle all types of hyperparameters

Disadvantages

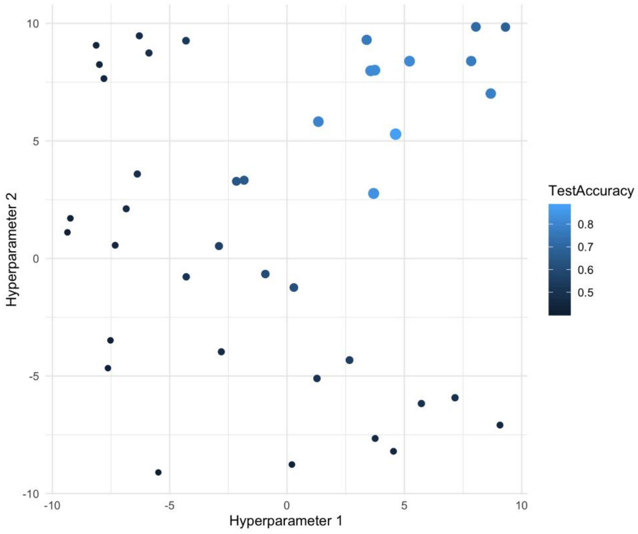
- Scales badly with #dimensions
- Inefficient: Searches irrelevant areas
- Requires to manual define discretization
- All grid points need to be evaluated

Option 2: Random Search

Variation of Grid Search: Uniformly sample configurations at random



Option 2: Random Search II



Advantages

- Very easy to implement
- Very easy to parallelize
- Can handle all types of hyperparameters
- No discretization required
- Anytime algorithm: Can be stopped and continued based on the available budget and performance goal.

Disadvantages

- Scales badly with #dimensions
- Inefficient: Searches irrelevant areas

Grid Search vs. Random Search

With a **budget** of T iterations:

Grid Search evaluates only $T^{\frac{1}{d}}$ unique values per dimension

Random Search evaluates (most likely) T different values per dimension

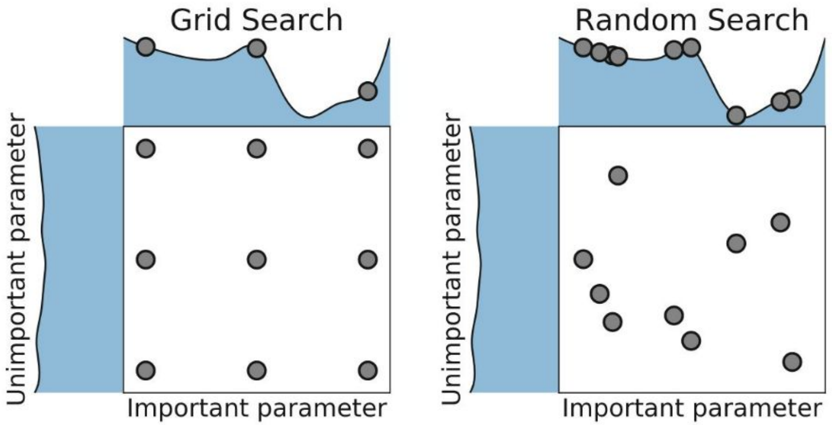
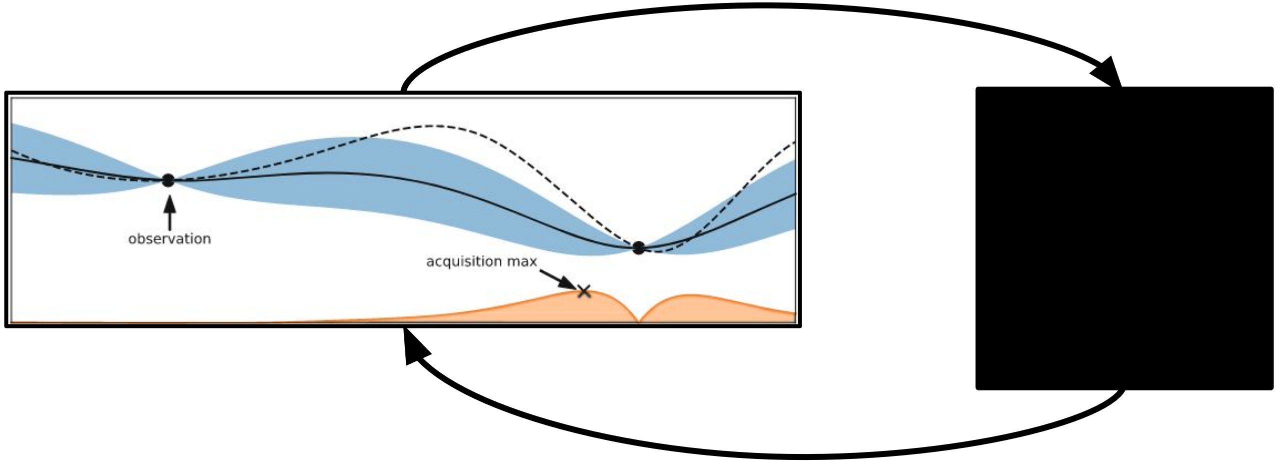


Image source: [Hutter et al. 2019]

→ Grid search can be disadvantageous if some hyperparameters have little of no impact on the performance [Bergstra et al. 2012]

Model-based Optimization

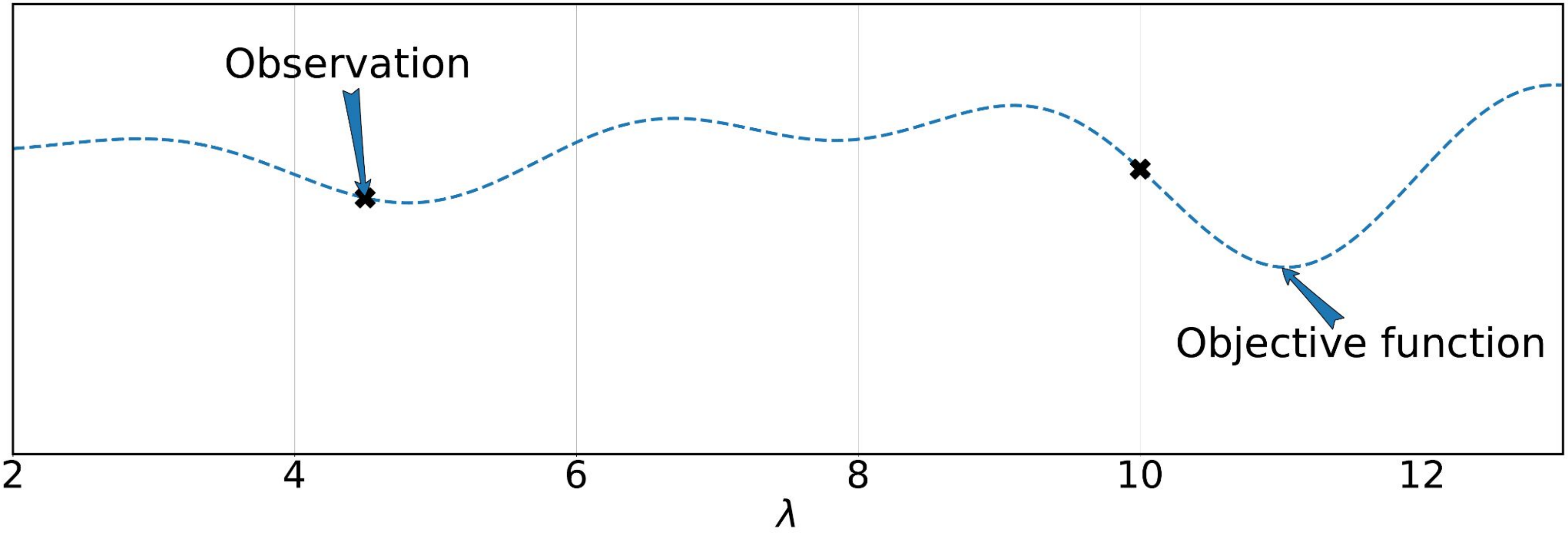
λ_n



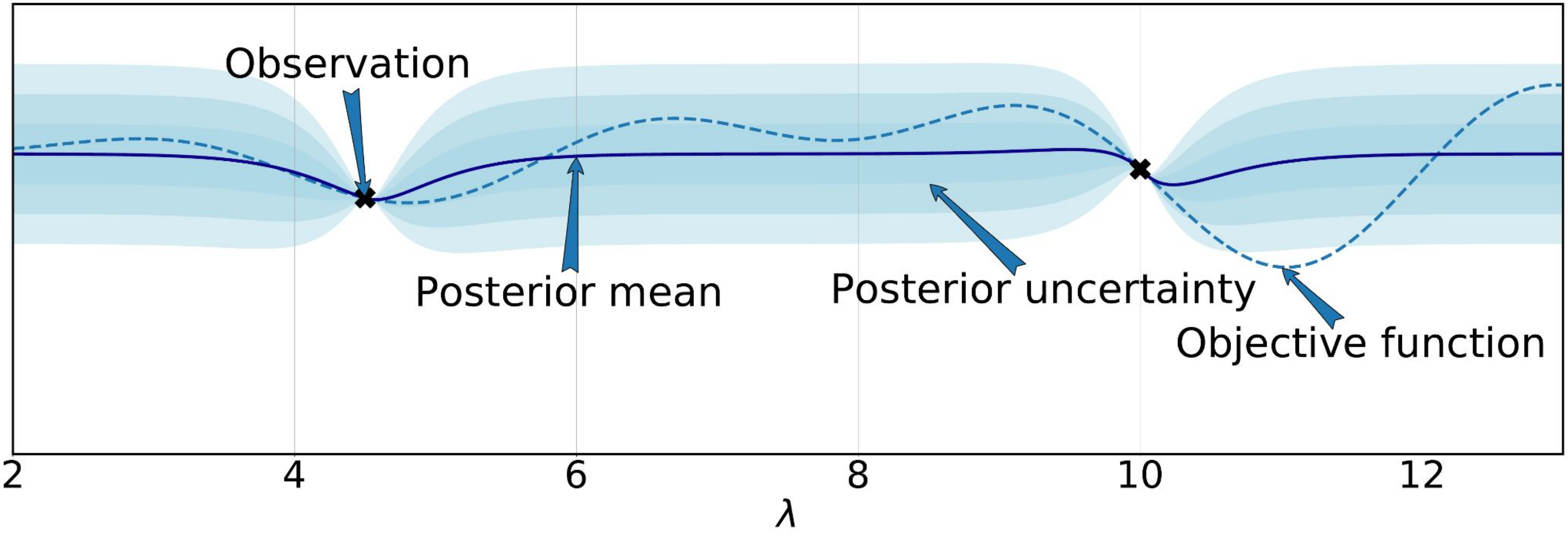
$f(A_{\lambda_n})$

Photo by Wilhelm Gunkel on Unsplash
Image by Feurer, Hutter: Hyperparameter Optimization
In: Automated Machine Learning

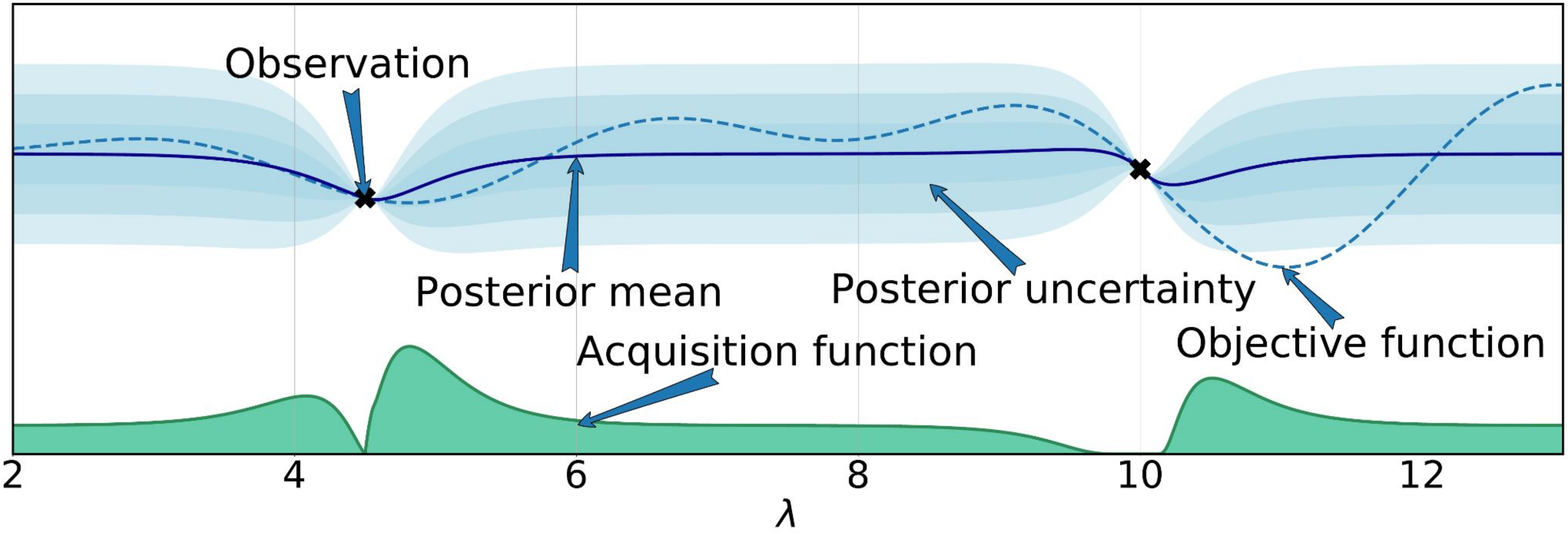
Bayesian Optimization in a Nutshell



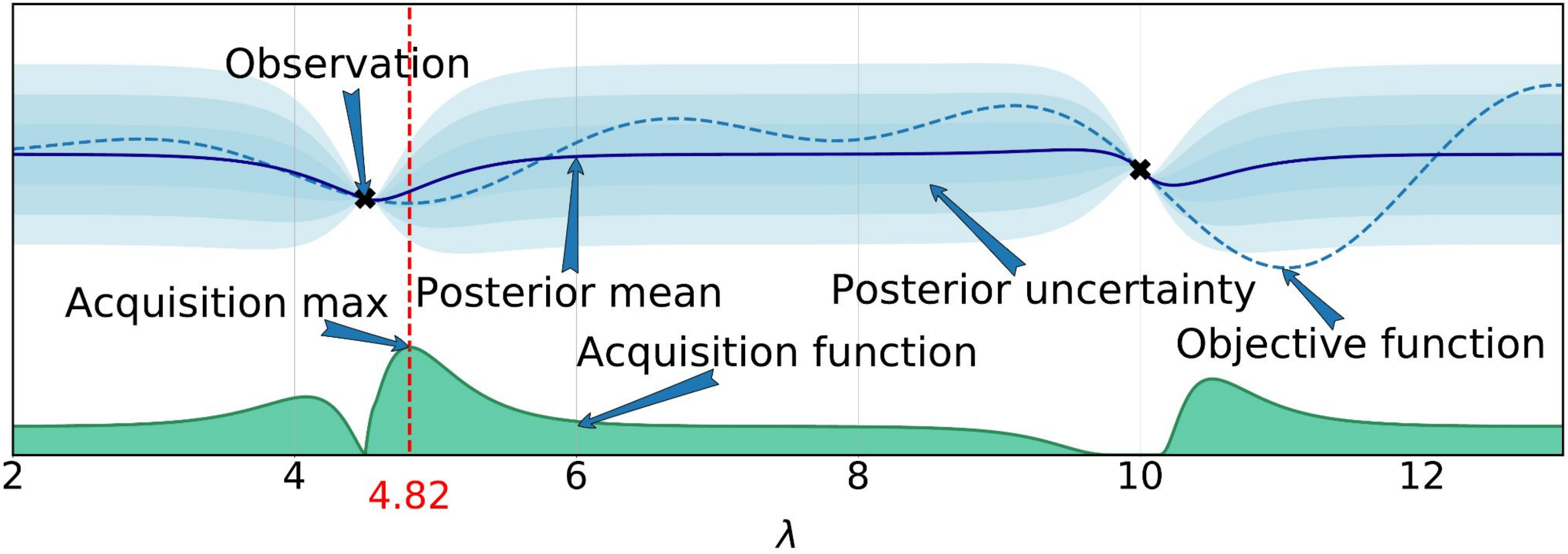
Bayesian Optimization in a Nutshell



Bayesian Optimization in a Nutshell



Bayesian Optimization in a Nutshell



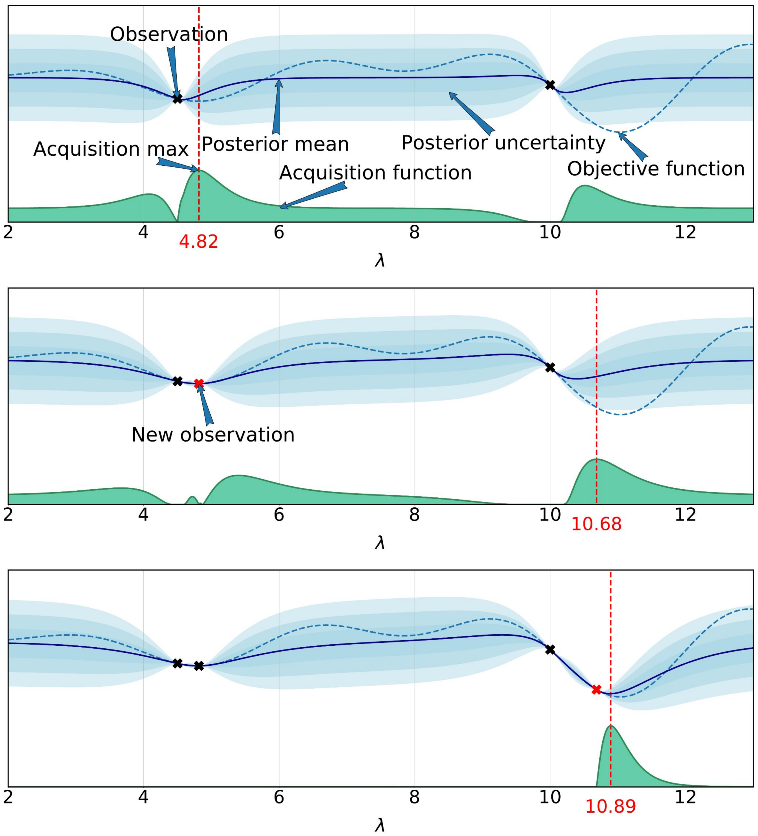
Bayesian Optimization in a Nutshell

General approach

- Fit a probabilistic model to the collected function samples $\langle \boldsymbol{\lambda}, c(\boldsymbol{\lambda}) \rangle$
- Use the model to guide optimization, trading off exploration *vs* exploitation

Popular approach in the statistics literature since Mockus et al. [1978]

- Efficient in #function evaluations
- Works when objective is nonconvex, noisy, has unknown derivatives, etc.
- Recent convergence results
  [Srinivas et al. 2009; Bull et al. 2011; de Freitas et al. 2012; Kawaguchi et al. 2015]



Bayesian Optimization: Pseudocode

BO loop

Require: Search space $\boldsymbol{\Lambda}$, cost function c, acquisition function u, predictive model $\hat{c}$, maximal number of function evaluations T

Result : Best configuration $\hat{\boldsymbol{\lambda}}$ (according to $\mathcal{D}$ or $\hat{c}$)

Initialize data $\mathcal{D}^{(0)}$ with initial observations

for $t = 1$ **to** T **do**

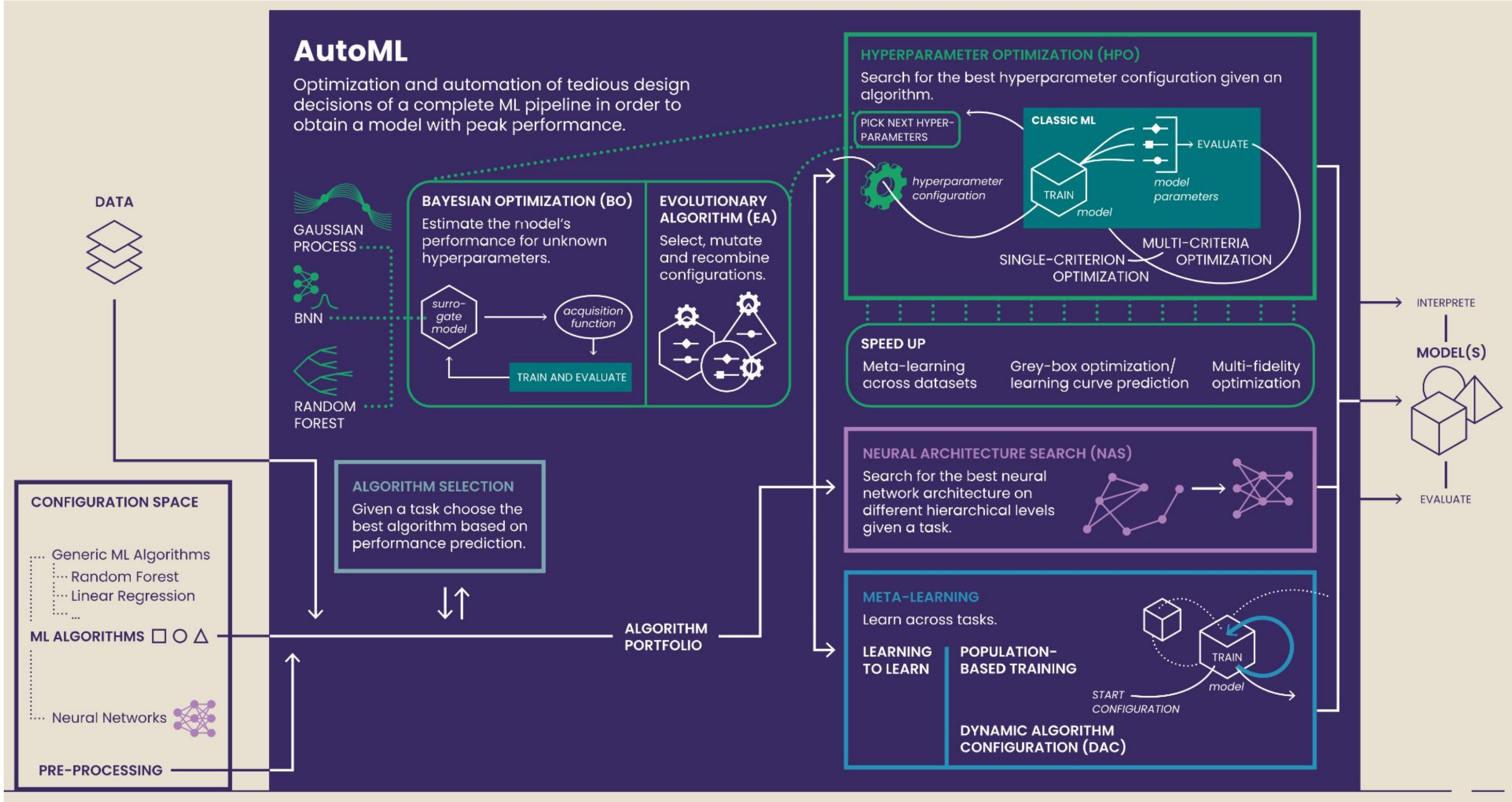
  Fit predictive model $\hat{c}^{(t)}$ on $\mathcal{D}^{(t-1)}$

  Select next query point: $\boldsymbol{\lambda}^{(t)} \in \arg\max_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} u(\boldsymbol{\lambda}; \mathcal{D}^{(t-1)}, \hat{c}^{(t)})$

  Query $c(\boldsymbol{\lambda}^{(t)})$

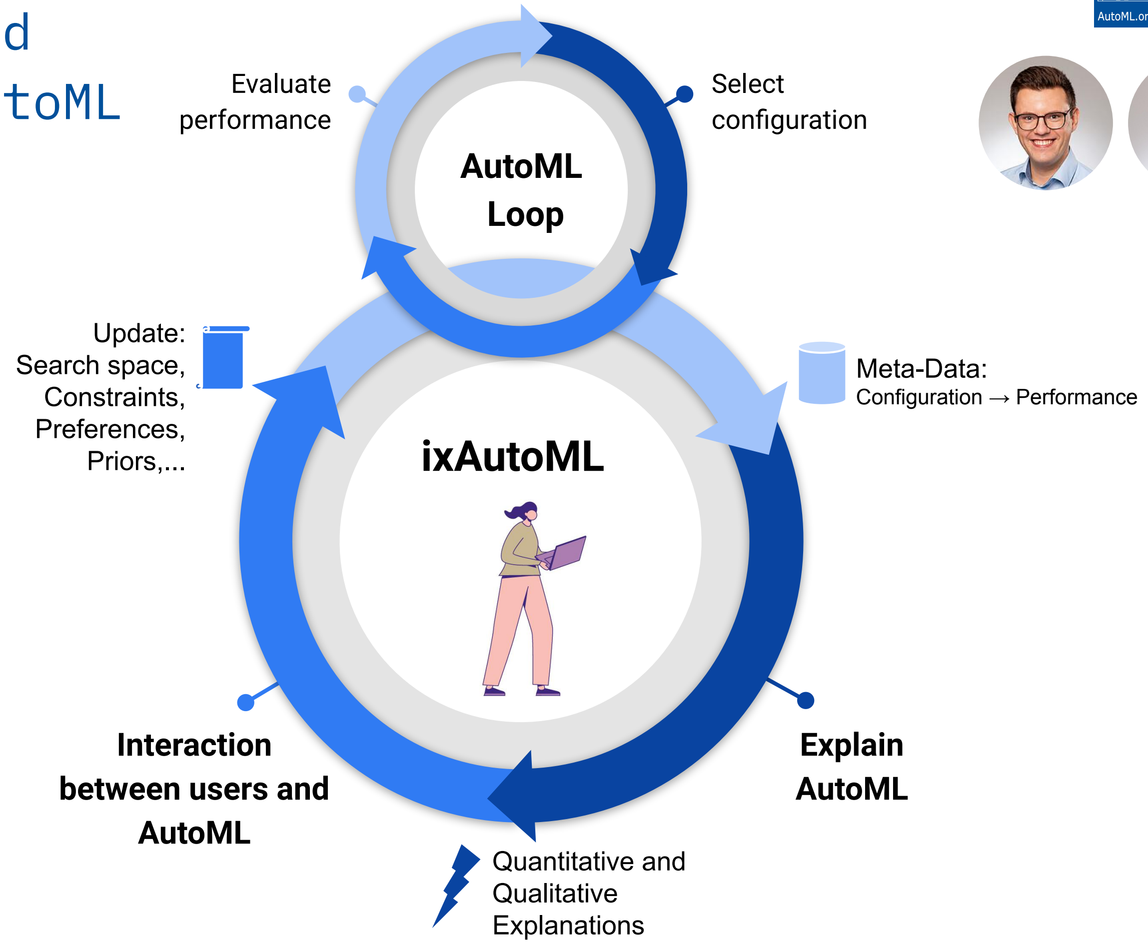
  Update data: $\mathcal{D}^{(t)} \leftarrow \mathcal{D}^{(t-1)} \cup \{\langle \boldsymbol{\lambda}^{(t)}, c(\boldsymbol{\lambda}^{(t)}) \rangle\}$

Is there more to AutoML?



Human-Centered AutoML

ixAutoML: interactive and explainable AutoML

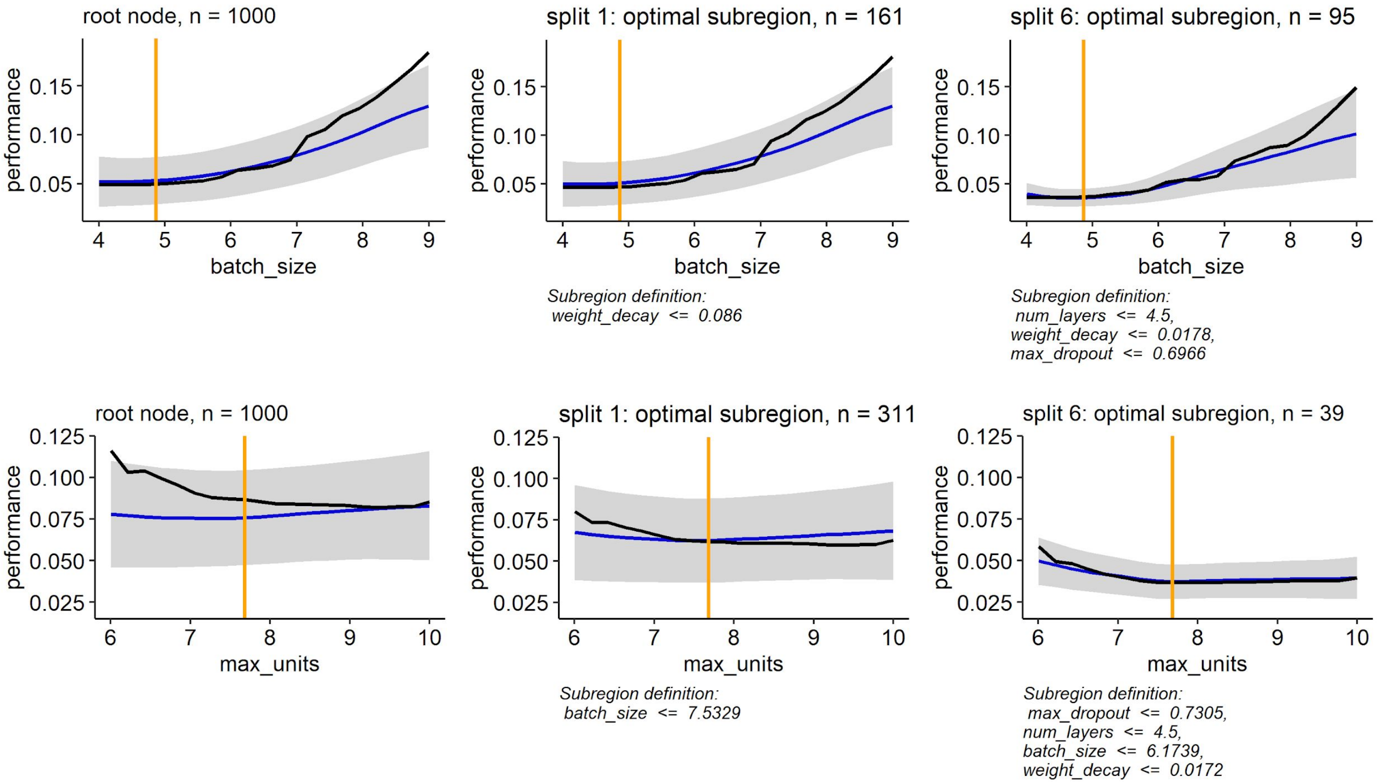


Explaining I: Partial Dependence Plots

Explaining Hyperparameter Effects via PDPs

[Moosbauer et al. NeurIPS’22]

Ground truth
PDP
incumbent



Partial Dependence Plots

[Moosbauer et al. NeurIPS'22]

For, a subset S of the hyperparameters, the partial dependence function is:

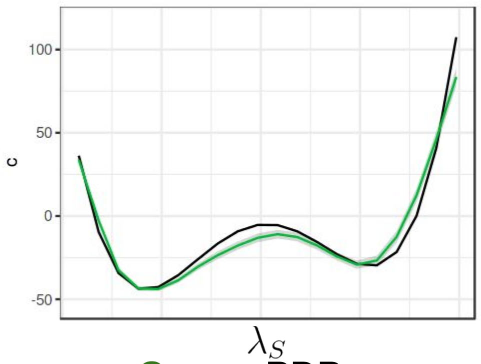
$$c_S(\lambda_S) := \mathbb{E}_{\lambda_C}\left[c(\lambda)\right] = \int_{\Lambda_C} c(\lambda_S, \lambda_C) d\mathbb{P}(\lambda_C)$$

and can be approximated by Monte-Carlo integration
on a surrogate model:

$$\hat{c}_S\left(\lambda_S\right) = \frac{1}{n}\sum_{i=1}^{n} \hat{m}\left(\lambda_S, \lambda_C^{(i)}\right)$$

where $\left(\lambda_C^{(i)}\right)_{i=1,\dots,n} \sim \mathbb{P}(\lambda_C)$ and λ_S for a set of grid points.

→ Average of ICE curves.



Green: PDP
Black: Ground truth

Partial Dependence Plots with Uncertainties

[Moosbauer et al. NeurIPS'22]

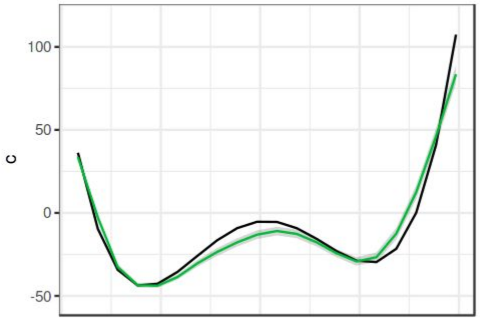
$$\hat{s}^2_S(\lambda_S) = \mathbb{V}_{\hat{c}}\left[\hat{c}_S\left(\lambda_S\right)\right] = \mathbb{V}_{\hat{c}}\left[\frac{1}{n}\sum_{i=1}^{n}\hat{c}\left(\lambda_S, \lambda_C^{(i)}\right)\right] = \frac{1}{n^2}1^\top \hat{K}\left(\lambda_S\right)\ 1.$$

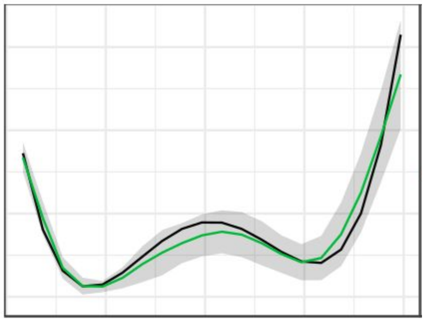
→ requires a kernel correctly specifying the covariance structure (e.g., GPs).

Approximation:

$$\hat{s}^2_S\left(\lambda_S\right) \approx \frac{1}{n}\sum_{i=1}^{n}\hat{K}\left(\lambda_S\right)_{i,i}$$

→ Model-agnostic (local) approximation

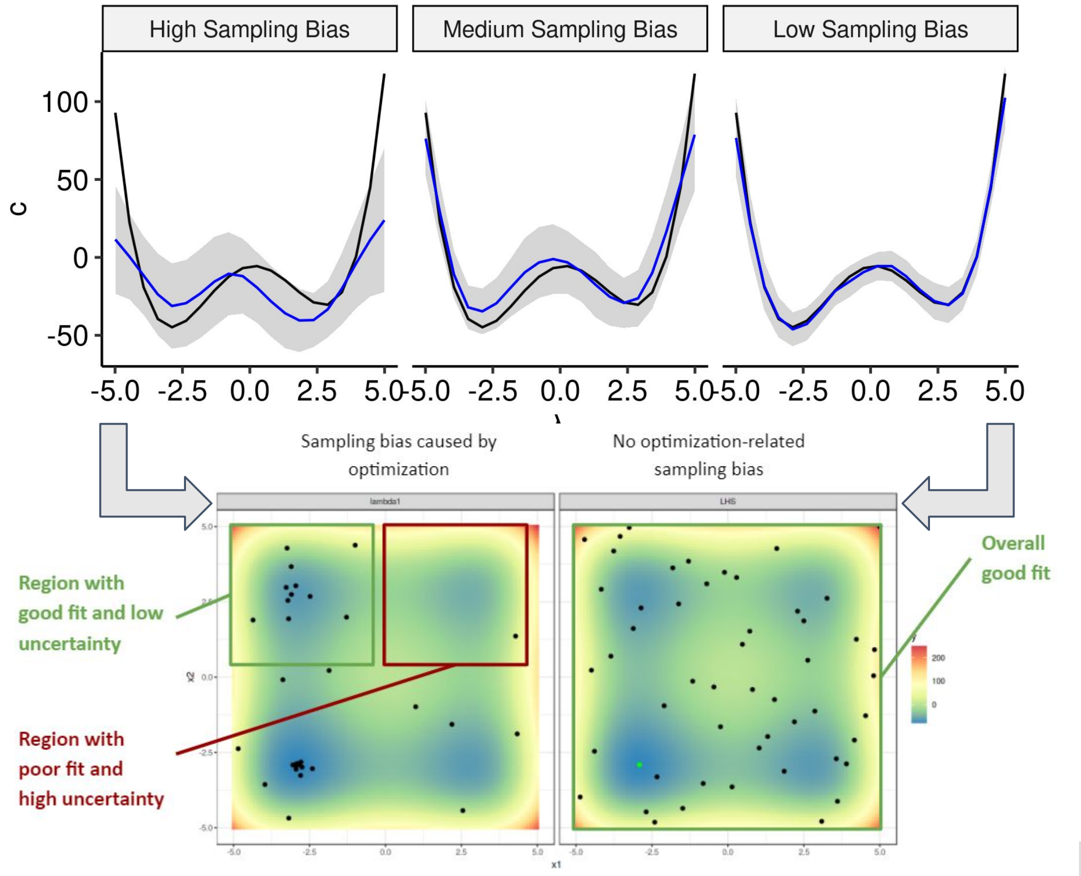




Ground truth
PDP
Uncertainty

Impact of Sampling Bias in Explaining AutoML

[Moosbauer et al. NeurIPS'22]



Explaining II: Symbolic Regression

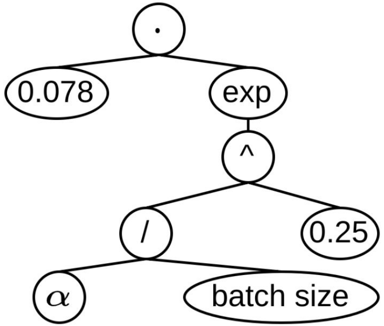
Symbolic Explanations for AutoML

[Segel et al. AutoML'23]

- Hyperparameter optimization (HPO) methods can find well-performing configurations efficiently
- Their **lack of transparency** can lead to missing trust of the users [Hasebrock et al. 2023]

Symbolic Explanations to the Rescue!

$$s(\alpha, \text{batch size}))$$
$$= 0.078 \cdot \exp\left((\alpha/\text{batch size})^{\frac{1}{4}}\right)$$

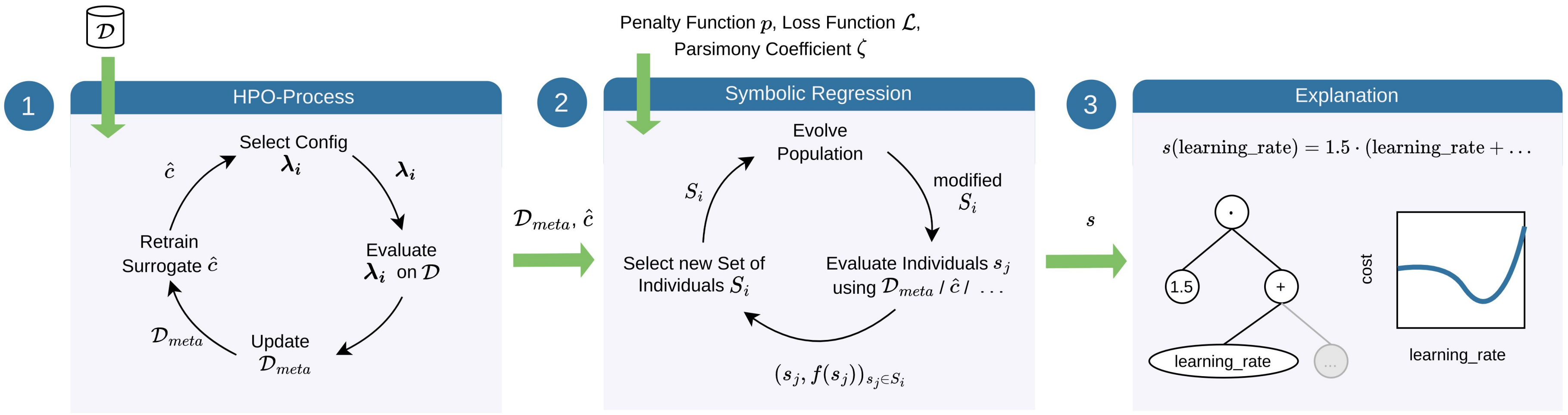


Symbolic Explanations for AutoML

[Segel et al. AutoML'23]

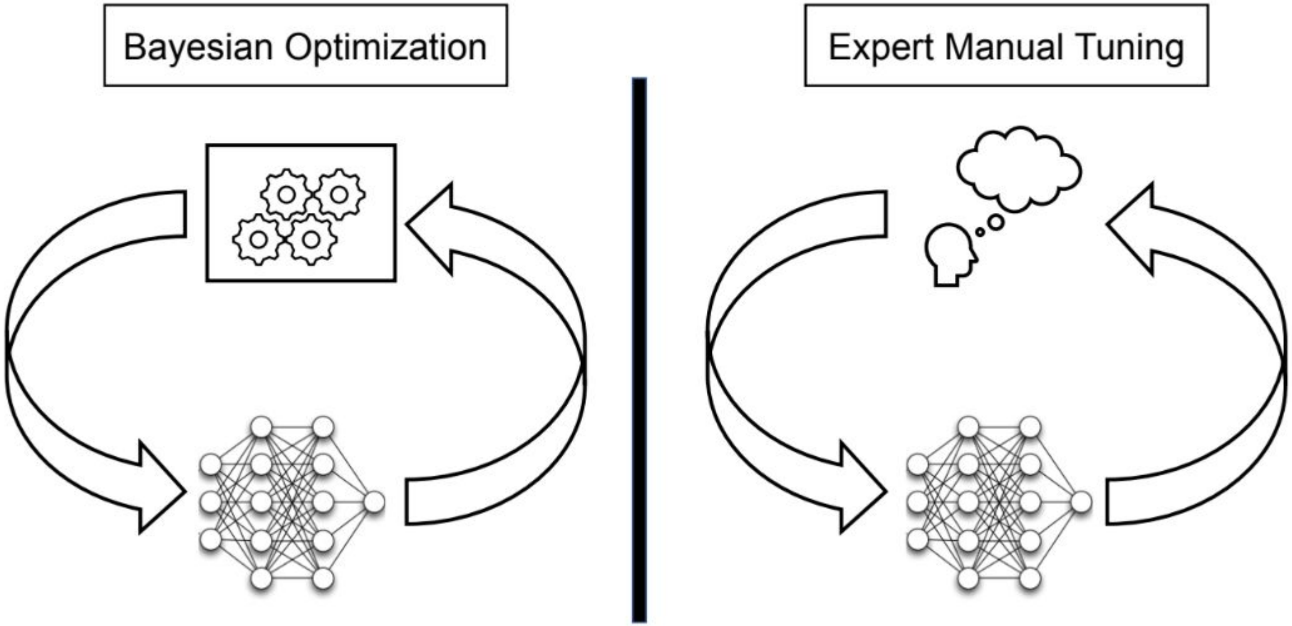
How to get more insights into hyperparameter effects?

- Employ **symbolic regression** to learn an **interpretable formula** that captures the relationship between hyperparameter configurations and model performance

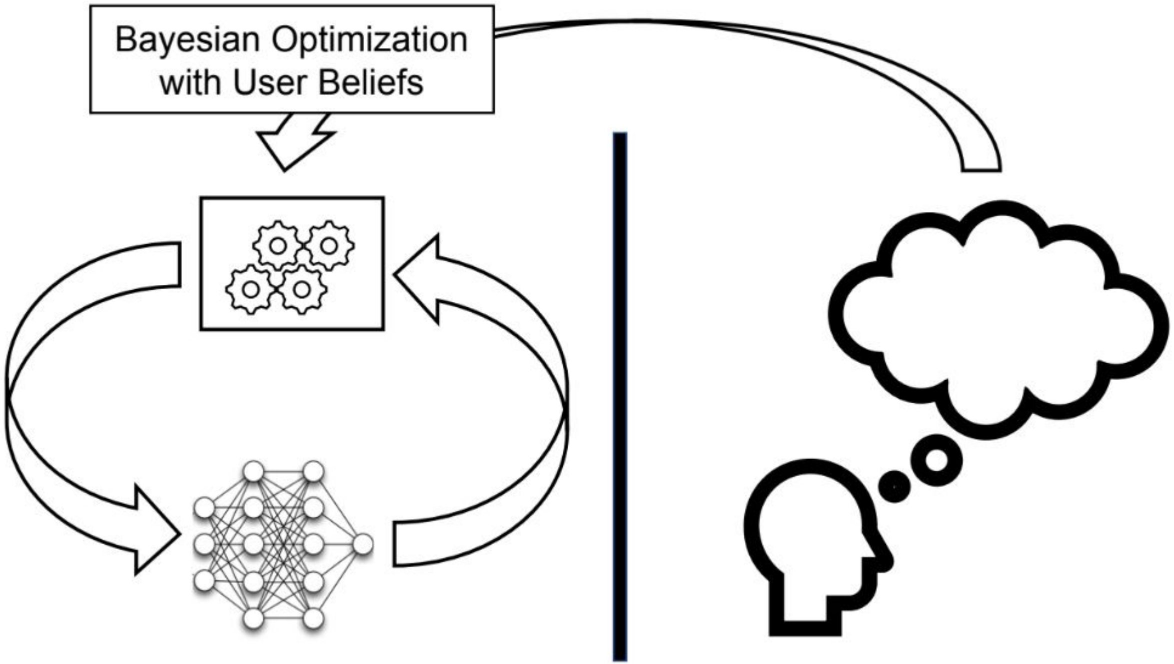


Interaction I: Expert-Priors

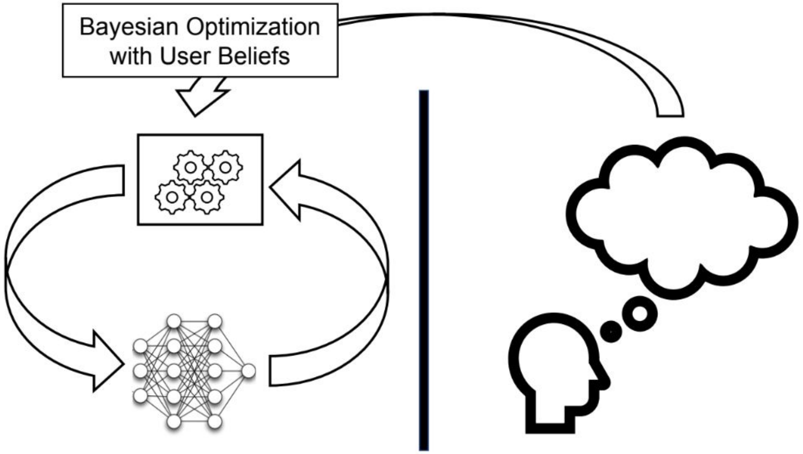
Bayesian Optimization vs Manual Tuning for HPO



Bayesian Optimization with Expert Knowledge



piBO [Hvarfner et al. ICLR'22]



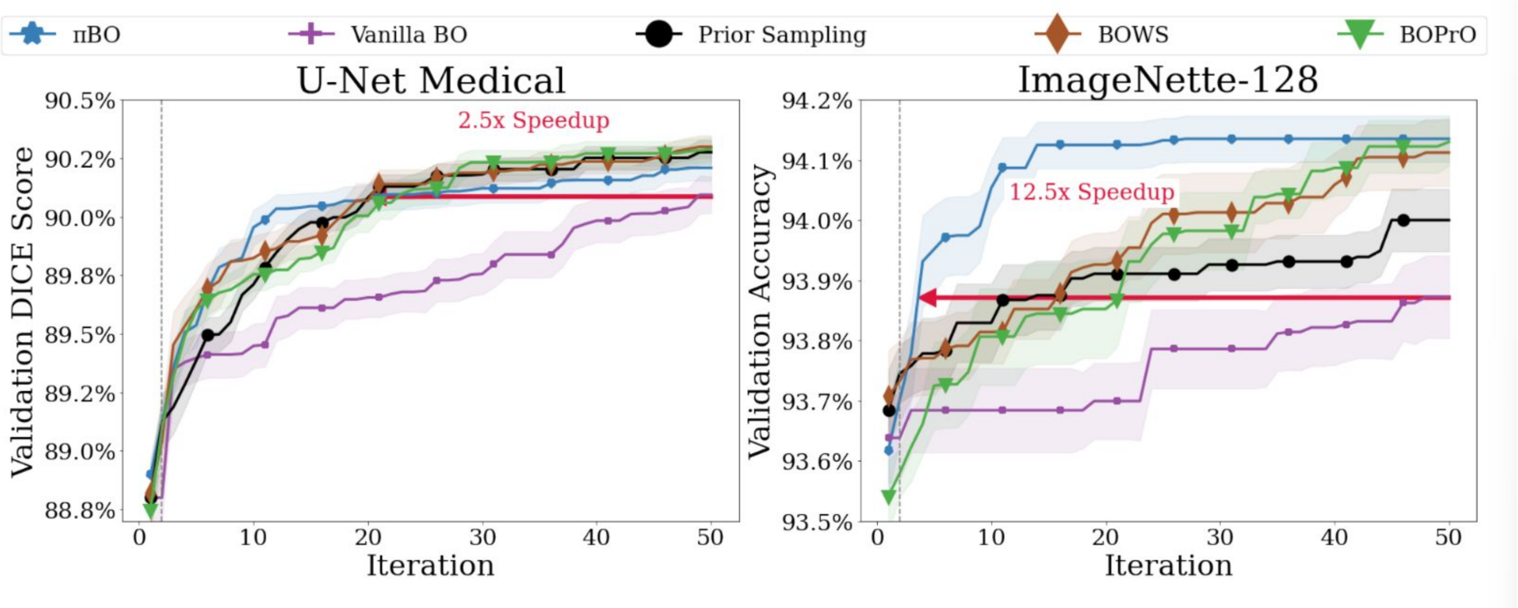
$$\boldsymbol{x}_n \in \underset{\boldsymbol{x} \in \mathcal{X}}{\arg\max}\, \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x})^{\beta/n}$$

Acquisition Function

User Prior

Speed of forgetting user prior

piBO [Hvarfner et al. ICLR'22]



- ➔ Uses expert knowledge to speed up Bayesian Optimization
- ➔ Robust also against wrong believes
- ➔ Substantially speeds up AutoML
- ➔ Follow up with PriorBand [Mallik et al. NeurIPS'23]

Interaction II: Preferences for Multi-Objective AutoML

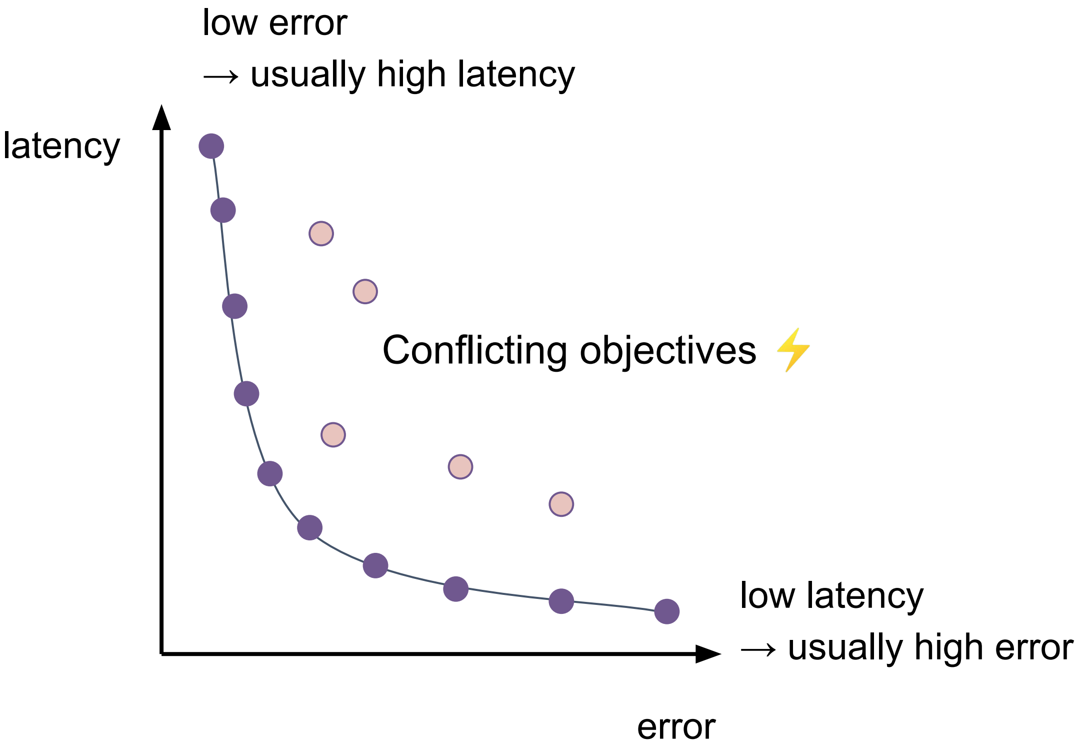
Multi-Objective AutoML

In practice, we often care about more than a single objective, e.g.

- error,
- inference time,
- unfairness,
- energy consumption,
- model complexity,
- and many more

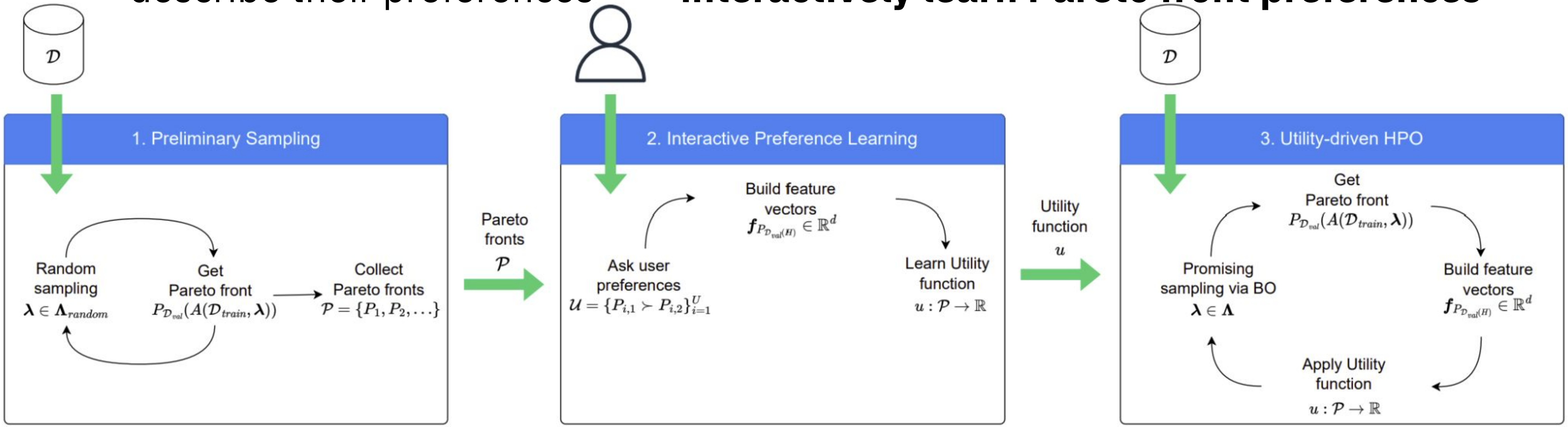
~~Goal: Find a Neural Network with high accuracy and low latency~~

Goal: Find the Pareto Set of Neural Networks that balance accuracy and latency.



Interactive HPO in Multi-Objective Problems via Preference Learning [Giovanelli et al. AAAI'24]

- Multi-objective (Auto)ML gets more and more important
  - e.g., hardware-aware NAS, fairness-aware AutoML or energy-efficient AutoML
- **Practical challenge**: Different multi-objective indicators lead to different approximated Pareto fronts and users cannot always mathematically describe their preferences ⇒ **interactively learn Pareto front preferences**



Evaluation of Preference-Learned Indicators

[Giovanelli et al. AAAI'24]



- Benchmark:
  - LCBench
  - Accuracy vs. Energy-Consumption
- Let's assume : User randomly chose a multi-objective (MO) indicator, but was actually hoping for the behavior of another MO indicator
- ⇒ learned preferences are better than randomly choosing a MO indicator

PB\IB	$HV \uparrow$	$SP \downarrow$	$MS \uparrow$	$R2 \downarrow$
$HV \uparrow$	0.76 (±0.17) \ **0.77 (±0.17)**	**0.76 (±0.17)** \ 0.52 (±0.24)	**0.76 (±0.17)** \ 0.52 (±0.21)	0.76 (±0.17) \ **0.77 (±0.16)**
$SP \downarrow$	**0.01 (±0.01)** \ 0.03 (±0.02)	**0.01 (±0.01)** \ **0.01 (±0.0)**	**0.01 (±0.01)** \ 0.04 (±0.03)	**0.01 (±0.01)** \ 0.04 (±0.02)
$MS \uparrow$	**0.61 (±0.09)** \ 0.19 (±0.08)	**0.61 (±0.09)** \ 0.19 (±0.12)	0.61 (±0.09) \ **0.65 (±0.06)**	**0.61 (±0.09)** \ 0.23 (±0.11)
$R2 \downarrow$	0.23 (±0.16) \ **0.22 (±0.16)**	**0.23 (±0.16)** \ 0.47 (±0.23)	**0.23 (±0.16)** \ 0.45 (±0.21)	0.23 (±0.16) \ **0.21 (±0.16)**

AutoML in Constrained Applications

AutoML in Heavily Constrained Applications

[Neutatz et al. VLDBJ’23]


Default AutoML Configuration

Validation Strategy:	Holdout 66/33
Ensembling:	yes
Incremental Training:	yes
Validation split reshuffle:	no
ML Hyperparameter space:	
SVM:	Yes
SVM_tol:	Yes
SVM_C:	Yes
Extra Trees:	Yes
KNN:	Yes
Multilayer Perceptron:	Yes
Any Feature Preprocessor:	Yes
302 hyperparameters	Yes

Adapt AutoML parameters to ML task and deactivate undesired ML hyperparameters

Dynamic AutoML Configuration

Validation Strategy:	Holdout 46/54
Ensembling:	no
Incremental Training:	yes
Validation split reshuffle:	yes
ML Hyperparameter space:	
SVM:	Yes
SVM_tol:	Yes
SVM_C:	No
Extra Trees:	Yes
KNN:	No
Multilayer Perceptron:	No
Any Feature Preprocessor:	No
302 hyperparameters	Yes/No

Searches for the optimal ML pipeline in the defined search space. A pipeline is defined by the selected ML hyperparameters.

ML Pipeline

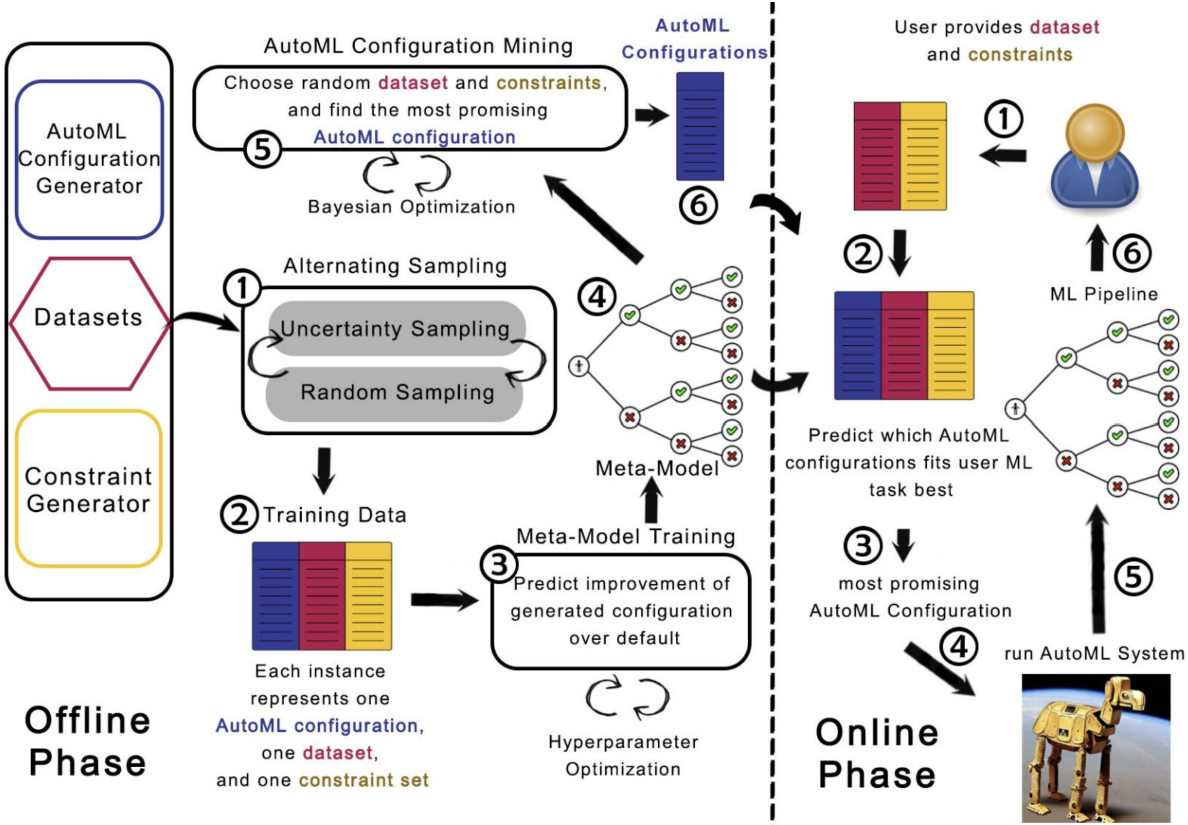
For SVM, the **model parameters** are the weights w:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b)\right)\right] + \lambda\|\mathbf{w}\|^2.$$

ML Hyperparameters:	
SVM:	Yes
SVM_tol:	1e-5
SVM_C:	1.0 (default)
Extra Trees:	No
KNN:	No
Multilayer Perceptron:	No
Any Feature Preprocessor:	No
302 hyperparameters	...

AutoML in Heavily Constrained Applications

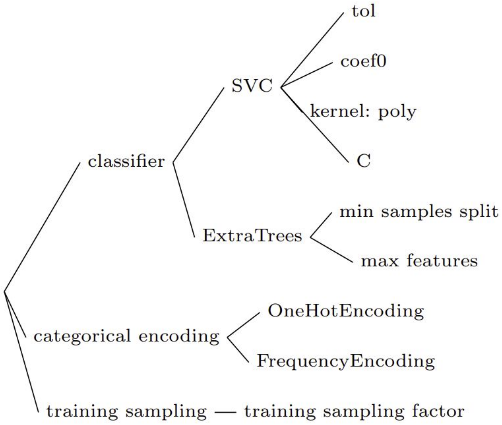
[Neutatz et al. VLDBJ'23]



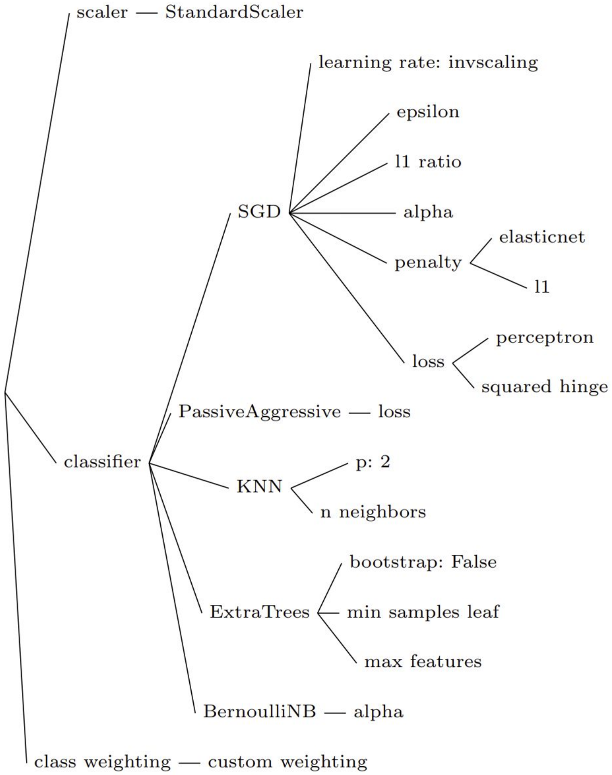
Possible application constraints:

- AutoML budget
- Inference time
- Memory consumption
- Energy consumption
- Fairness thresholds
- ...

Can it learn to select different configuration spaces? [Neutatz et al. 2023]



(a) Christine (1min search time)



(b) Robert (5min search time)

Take-Aways for Meta-Learning AutoML Conf.


- **Assumption**: If we invest more time into the development of AutoML packages (incl. meta-learning), we save a lot of compute resources for using it
- **Positive** take-away:
  Yes, we can meta-learn how to configure AutoML systems and achieve new state-of-the-art performance
- **Negative** take-away:
  We cannot easily do it for large AutoML budgets (beyond 10min) without enormous compute resources
- **Future challenge**: How to configure AutoML on expensive tasks;
  "Expensive" can mean:
  - very expensive ML models (e.g., LLMs)
  - very complex configuration spaces with thousands of ML trainings

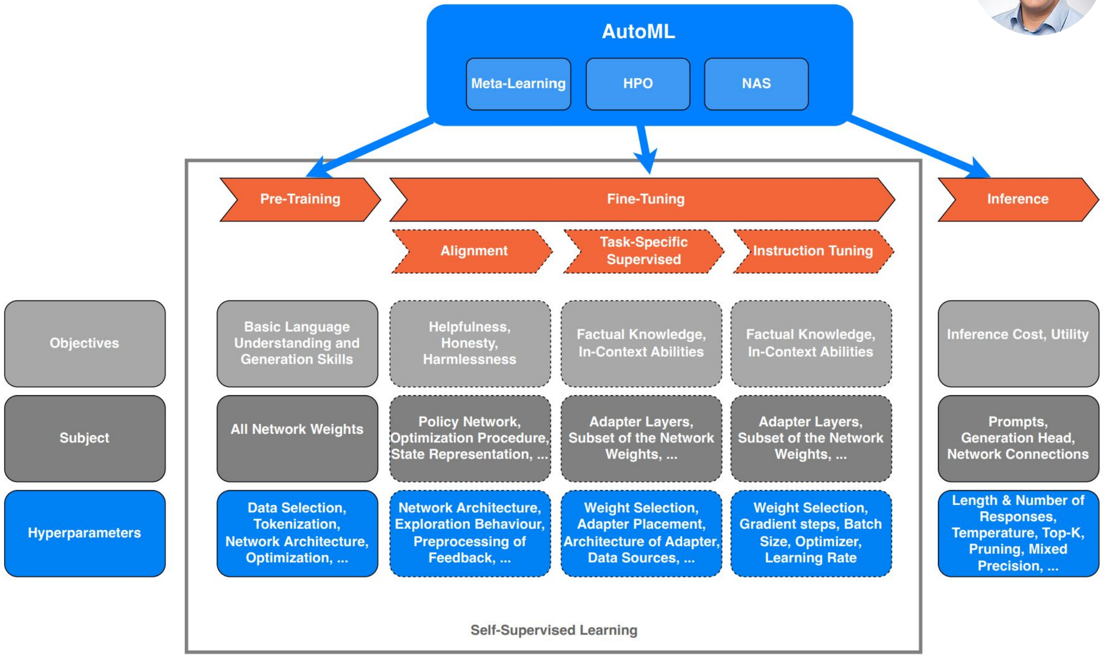
AutoML ⟺ LLMs

AutoML ➡ LLMs [Tornede et al. 2023]

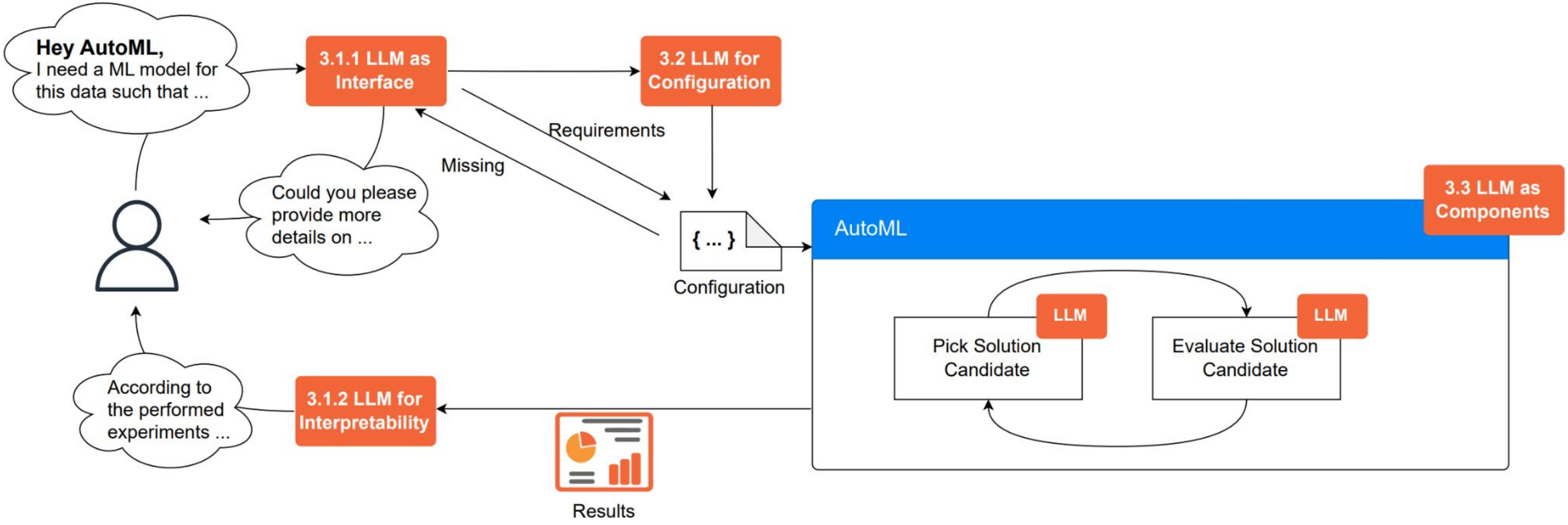
Leibniz Universität Hannover

Challenges

1. Cost of Pre-Training Base Models
2. Multitude of Different Stages
3. Multitude of Performance Indicators
4. Combination of Different Learning Paradigms
5. Determining Neural Architectures for LLMs



AutoML ⟵ LLMs [Tornede et al. 2023]



Green AutoML

Green AutoML



Energy-efficient AutoML

Data compression,
Zero-cost AutoML,
multi-fidelity,
intelligent stopping, …

Searching for Energy-Efficient Models

Model size constraint,
Energy-aware objective functions,
Energy efficient architectures,
Model compression, …

AutoML for Sustainability

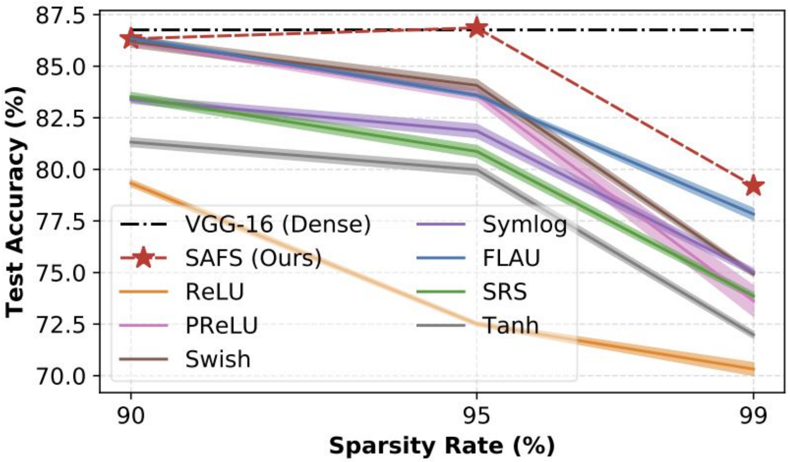
Plastic Litter Detection,
Green Assisted Driving,
Predictive Maintenance, …

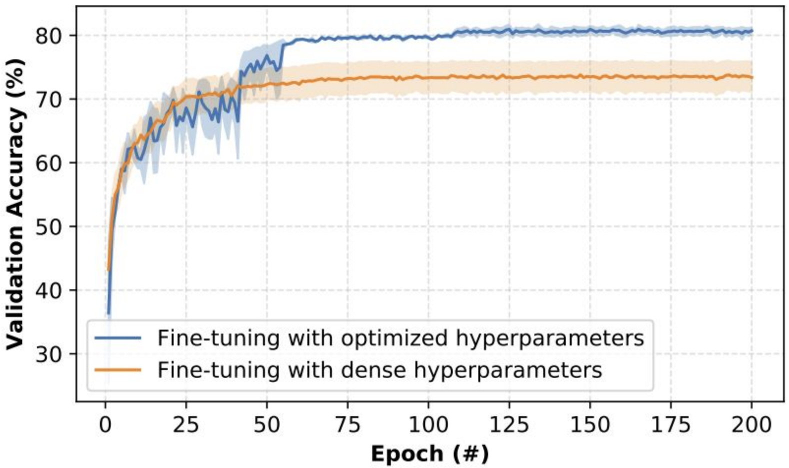
Create Attention

Tracking emissions,
awareness among students,
researchers, industry partners, …

Learning Activation Functions for Sparse Neural Networks [Loni et al. AutoML'23]

- Sparsifying networks can help to save a lot of compute power
- **Insights**:
  1. Using the same activation function class as for the dense network is suboptimal for pruning
  2. Hyperparameters have to adapted accordingly

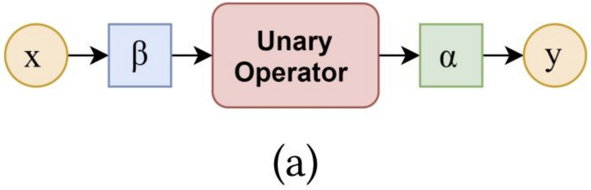




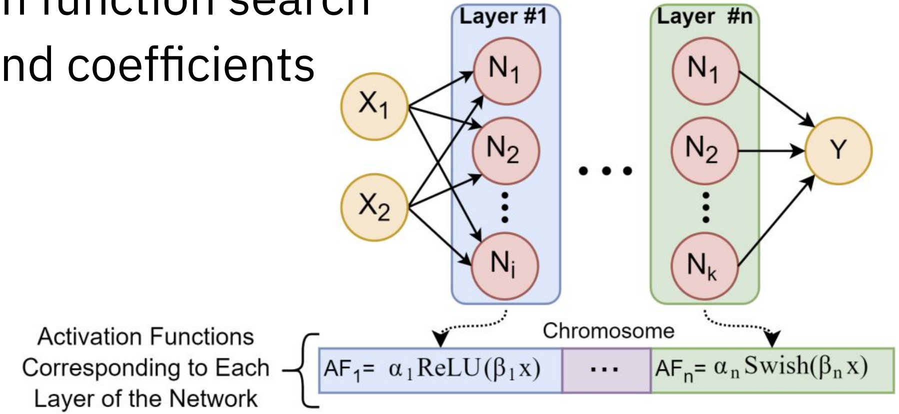
Learning Activation Functions for Sparse Neural Networks [Loni et al. AutoML'23]

Take-Aways:

- Search for activation functions for the pruning process
- Activation functions should even differ for different layers
  - Symlog and Acon in early layers
  - Swish in middle layers
- Stage 1: Use EA (LAHC) for activation function search
- Stage 2: Apply SGD-based HPO to find coefficients



(a)

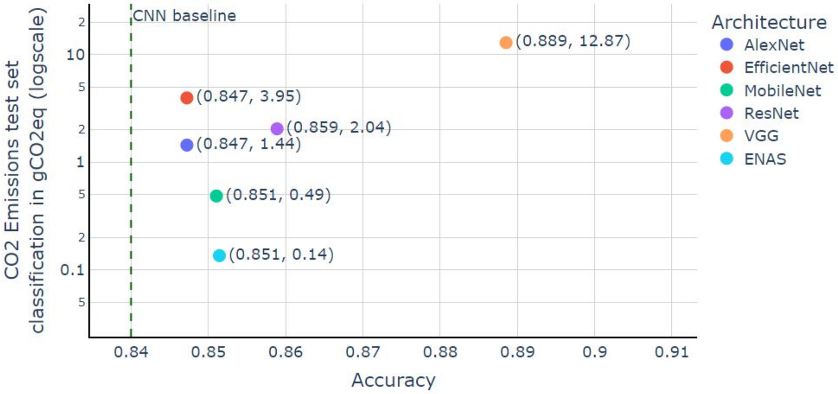


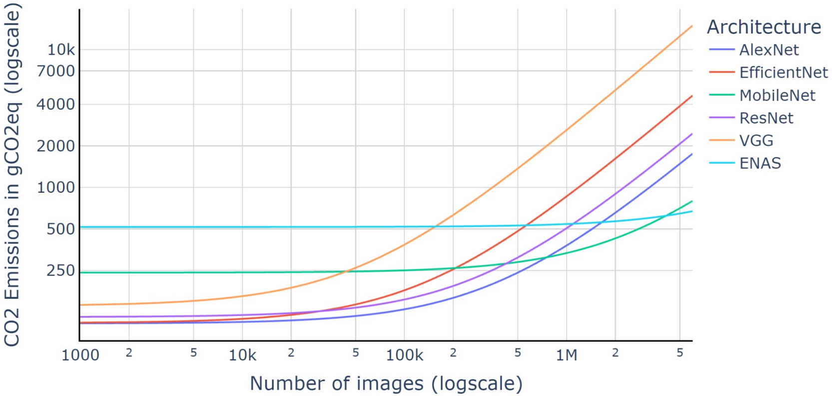
Green AutoML for Plastic Litter Detection

[Theodorakopoulos et al. CCAI’23]

Insights:

1. Architecture of DNNs with better accuracy
2. Architecture with lower CO_2 emissions
3. CO_2 emissions of AutoML training is compensated at inference





Packages from my Group

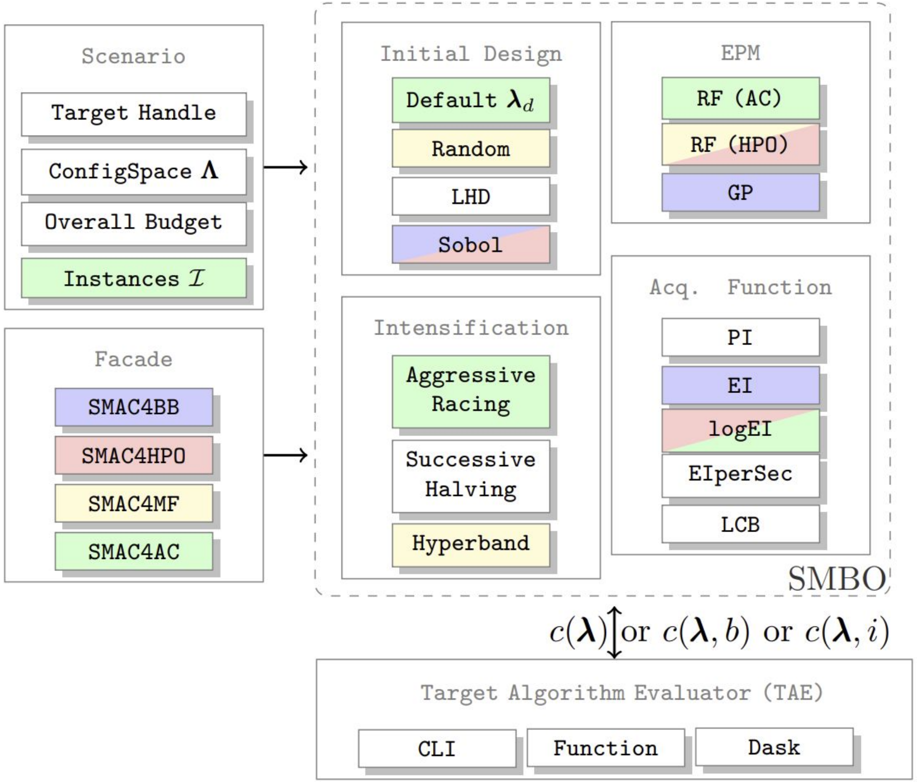
automl / SMAC3 Public

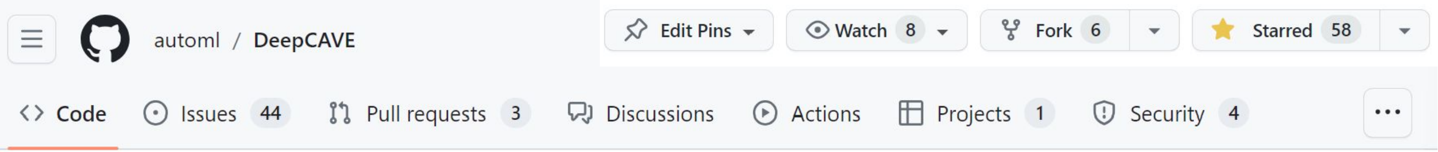
Notifications | Fork 206 | Star 920

Code | Issues 74 | Pull requests 10 | Actions | Projects 1 | Security | ...

> 67k downloads last month!

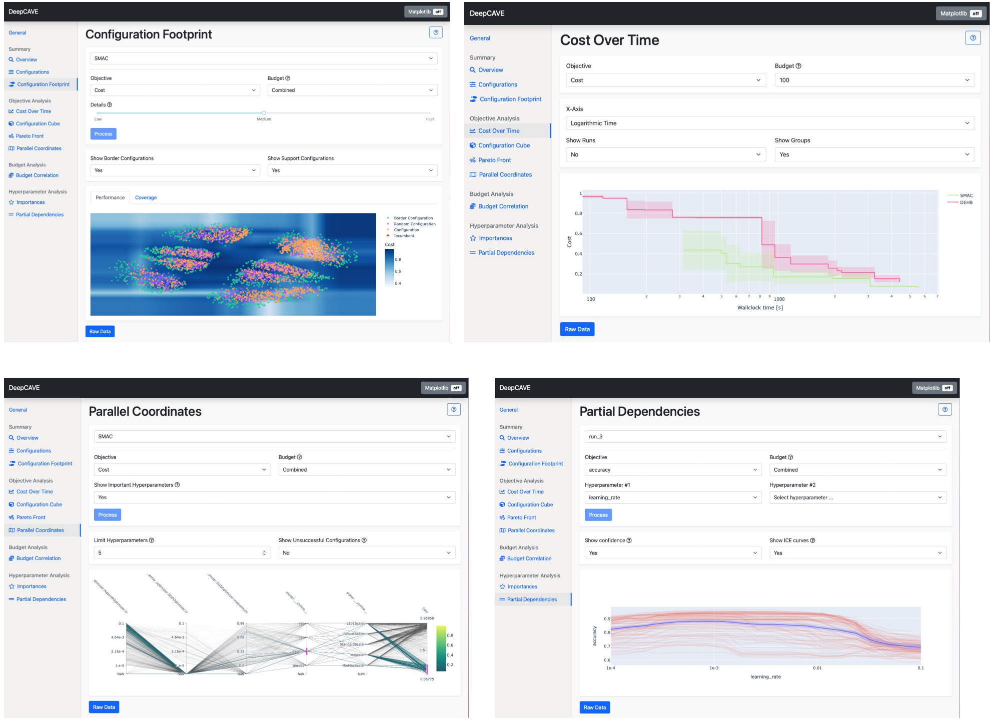
- Covers all kind of hyperparameter optimization use cases
- State-of-the-art techniques and performance
  - Bayesian Optimization
  - Multi-fidelity optimization
  - Multi-objective optimization
  - Algorithm configuration
- Highly configurable and modular
- Parallelizable





DeepCAVE [Sass et al. 2022]

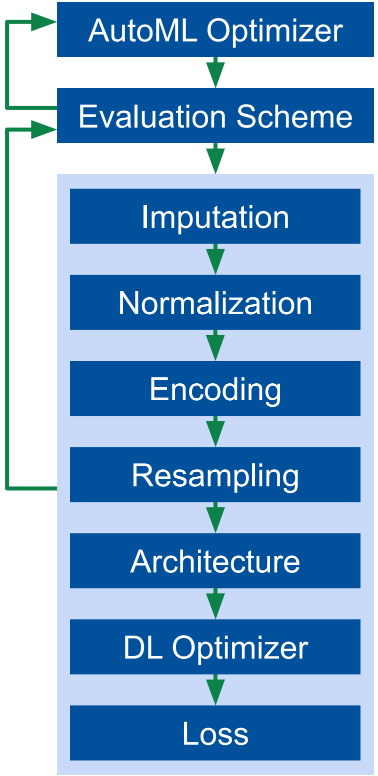
- **Interactive Dashboard** to self-analyze optimization runs/processes.
- **Analyzing while optimizing**
- **Exploration of multiple areas** like performance, hyperparameter and budget analysis.
- **Modularized plugin** structure with access to selected runs/groups to provide maximal flexibility.
- **Asynchronous execution** of expensive plugins and caching of their results.
- **API mode** gives you full access to the code



automl / Auto-PyTorch Public

Notifications | Fork 266 | Star 2.2k

Code | Issues 50 | Pull requests 20 | Actions | Projects 3 | Wiki | Security | ...



1. Automatic deep learning covering the entire DL pipeline
2. Joint hyperparameter and neural architecture search

➜ Auto-PyTorch Tabular
[Zimmer et al. IEEE TPAMI’21]

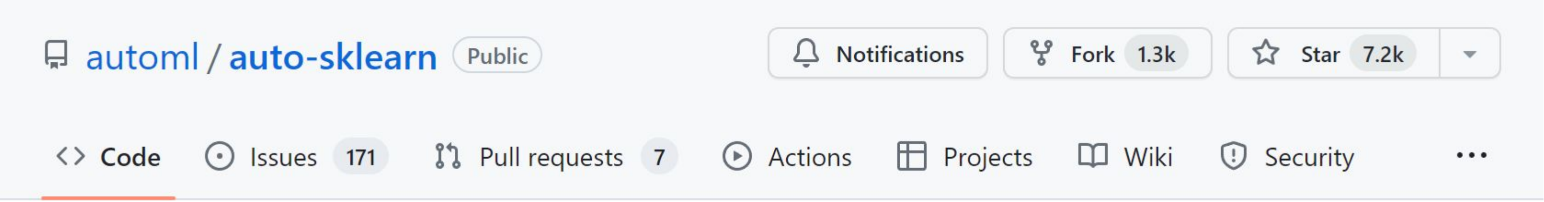
➜ State-of-the-art on tabular data with regularization cocktails
[Kadra et al. NeurIPS’21]

➜ Auto-PyTorch for Time Series Forecasting
[Deng et al. ECML’22]

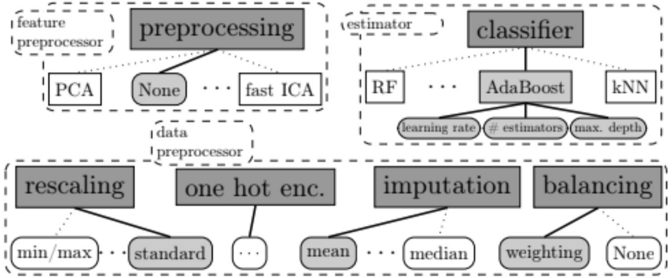
```
# initialise Auto-PyTorch api
api = TabularClassificationTask()

# Search for an ensemble of machine learning algorithms
api.search(
    X_train=X_train,
    y_train=y_train,
    X_test=X_test,
    y_test=y_test,
    optimize_metric='accuracy',
    total_walltime_limit=300,
    func_eval_time_limit_secs=50
)

# Calculate test accuracy
y_pred = api.predict(X_test)
```



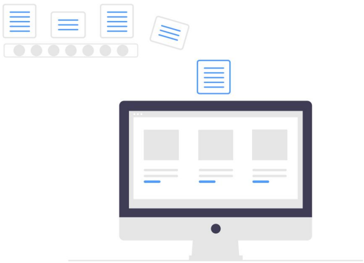
Takes care of finding well-performing ML-pipeline



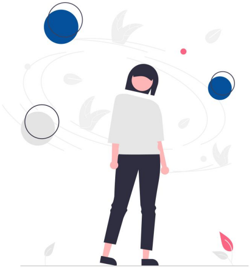
Easy-to-use

```
import autosklearn.classification as cls
automl = cls.AutoSklearnClassifier()
automl.fit(X_train, y_train)
y_hat = automl.predict(X_test)
```

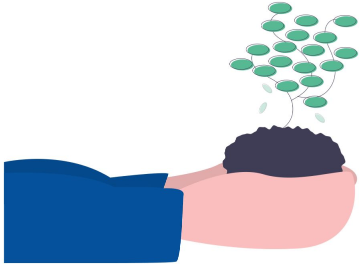
Our Research Foci



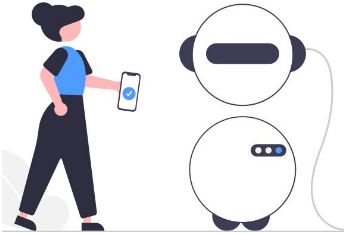
Core AutoML



Human-centered AutoML

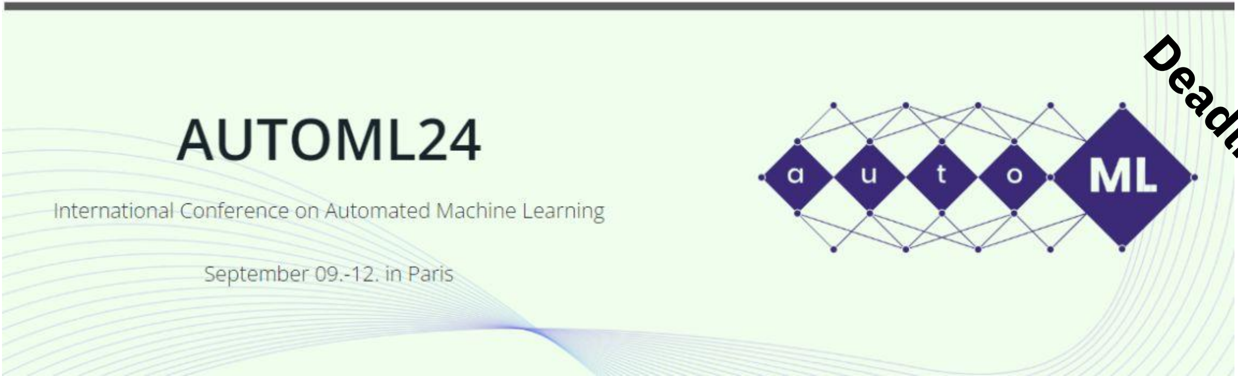


Green AutoML



AutoRL

AutoML Weeks 2024



Submission Deadline Feb 29th

Find Us

AutoML.org

automl-org

automl

@AutoML_org

LUH|AI

luh-ai

LUH-AI

@luh-ai

Funded by:

European Research Council
Established by the European Commission

DFG
Deutsche Forschungsgemeinschaft

L3S

DAAD
Deutscher Akademischer Austauschdienst
German Academic Exchange Service

Federal Ministry of Education and Research

Federal Ministry for Economic Affairs and Energy

Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection

Niedersachsen