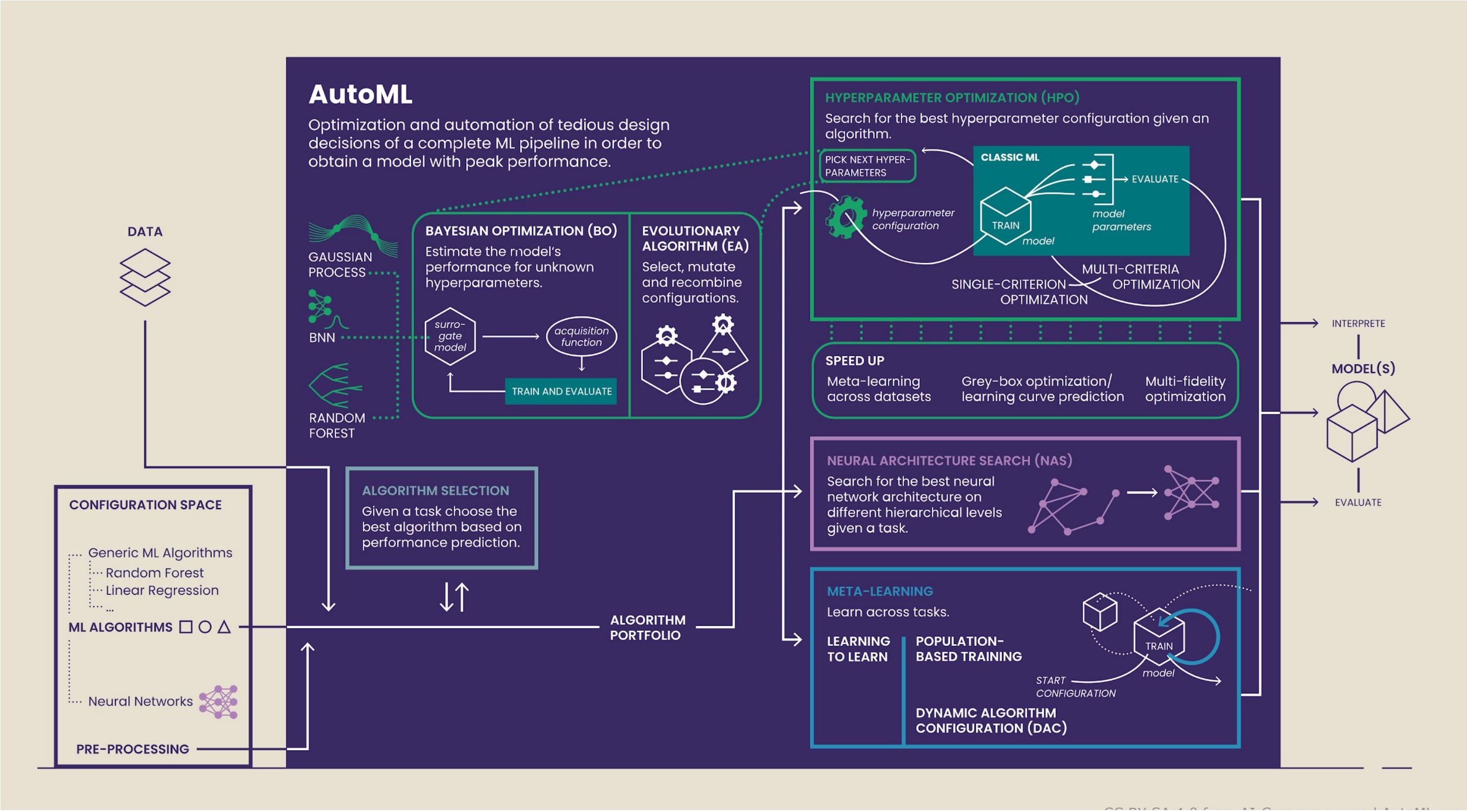
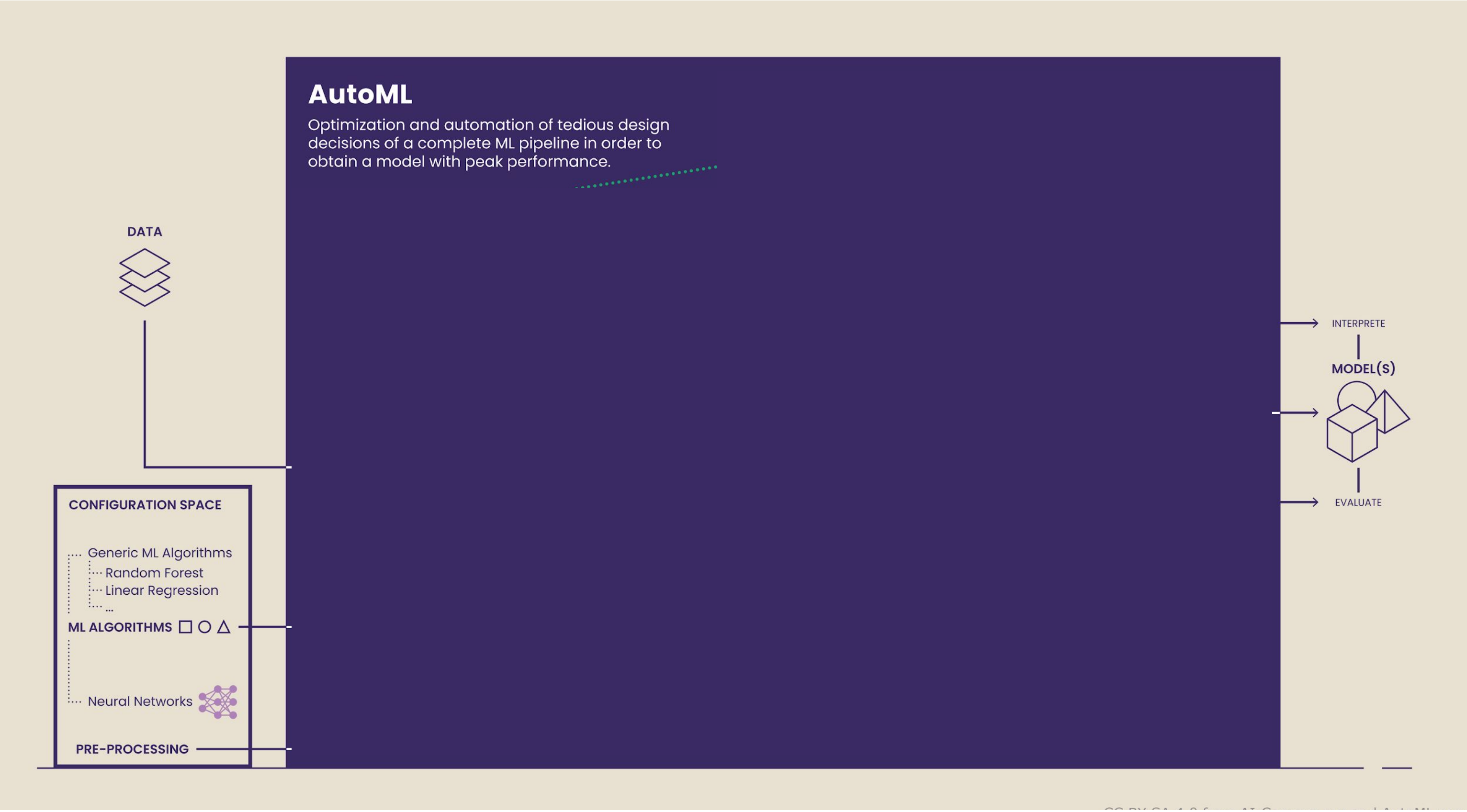
The AutoML Landscape

Marius Lindauer & Frank Hutter

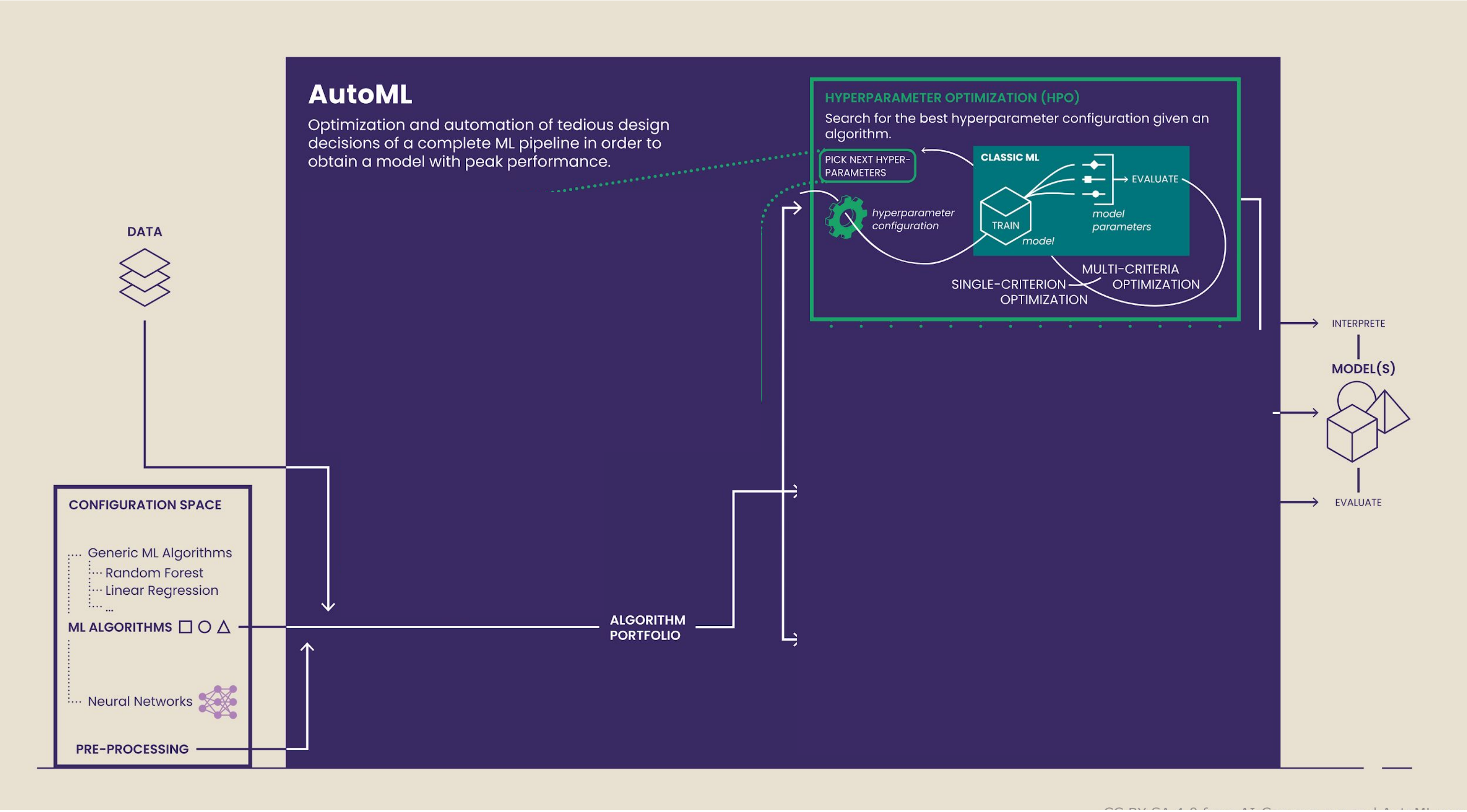
AutoML A-Z


DATA
CONFIGURATION SPACE
Generic ML Algorithms
Random Forest
Linear Regression
...
ML ALGORITHMS
Neural Networks
PRE-PROCESSING
AutoML
Optimization and automation of tedious design decisions of a complete ML pipeline in order to obtain a model with peak performance.
GAUSSIAN PROCESS
BNN
RANDOM FOREST
BAYESIAN OPTIMIZATION (BO)
Estimate the model's performance for unknown hyperparameters.
surro-gate model
acquisition function
TRAIN AND EVALUATE
EVOLUTIONARY ALGORITHM (EA)
Select, mutate and recombine configurations.
ALGORITHM SELECTION
Given a task choose the best algorithm based on performance prediction.
ALGORITHM PORTFOLIO
HYPERPARAMETER OPTIMIZATION (HPO)
Search for the best hyperparameter configuration given an algorithm.
PICK NEXT HYPER-PARAMETERS
CLASSIC ML
EVALUATE
hyperparameter configuration
TRAIN
model
model parameters
MULTI-CRITERIA OPTIMIZATION
SINGLE-CRITERION OPTIMIZATION
SPEED UP
Meta-learning across datasets
Grey-box optimization/ learning curve prediction
Multi-fidelity optimization
NEURAL ARCHITECTURE SEARCH (NAS)
Search for the best neural network architecture on different hierarchical levels given a task.
META-LEARNING
Learn across tasks.
LEARNING TO LEARN
POPULATION-BASED TRAINING
TRAIN
model
START CONFIGURATION
DYNAMIC ALGORITHM CONFIGURATION (DAC)
INTERPRETE
MODEL(S)
EVALUATE

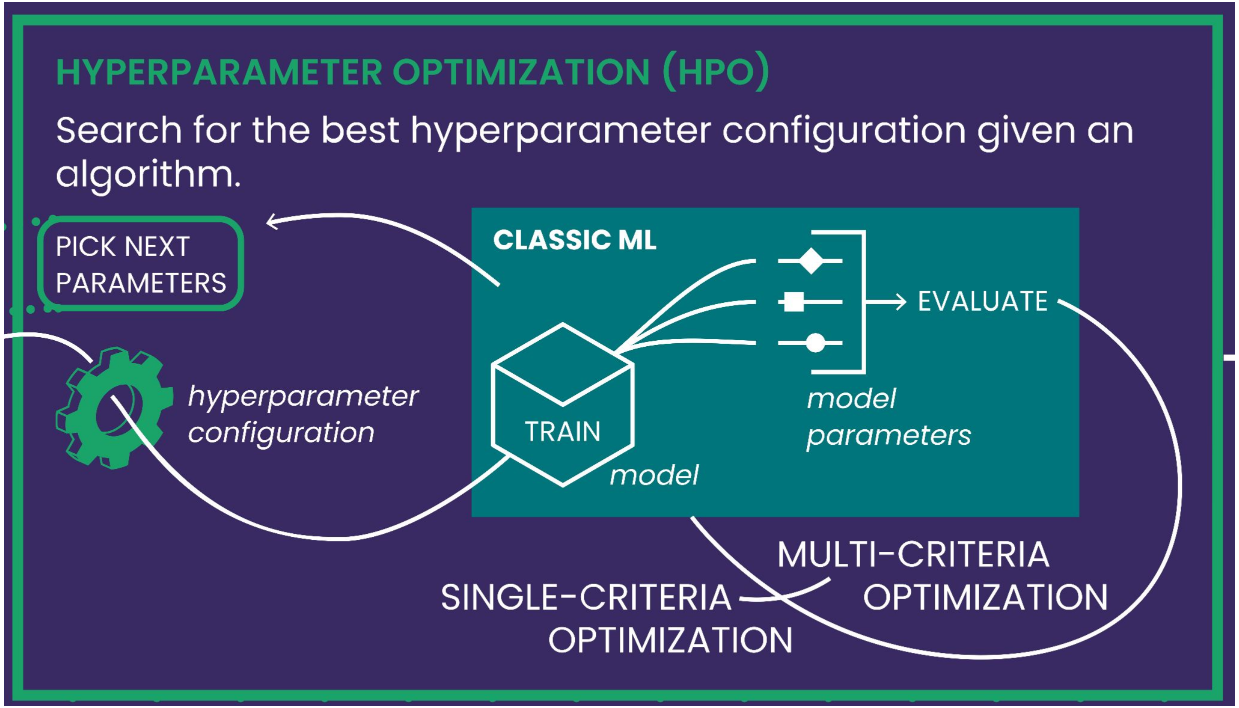
AutoML A-Z


AutoML
Optimization and automation of tedious design decisions of a complete ML pipeline in order to obtain a model with peak performance.
DATA
CONFIGURATION SPACE
Generic ML Algorithms
Random Forest
Linear Regression
...
ML ALGORITHMS
Neural Networks
PRE-PROCESSING
INTERPRETE
MODEL(S)
EVALUATE

Hyperparameter Optimization



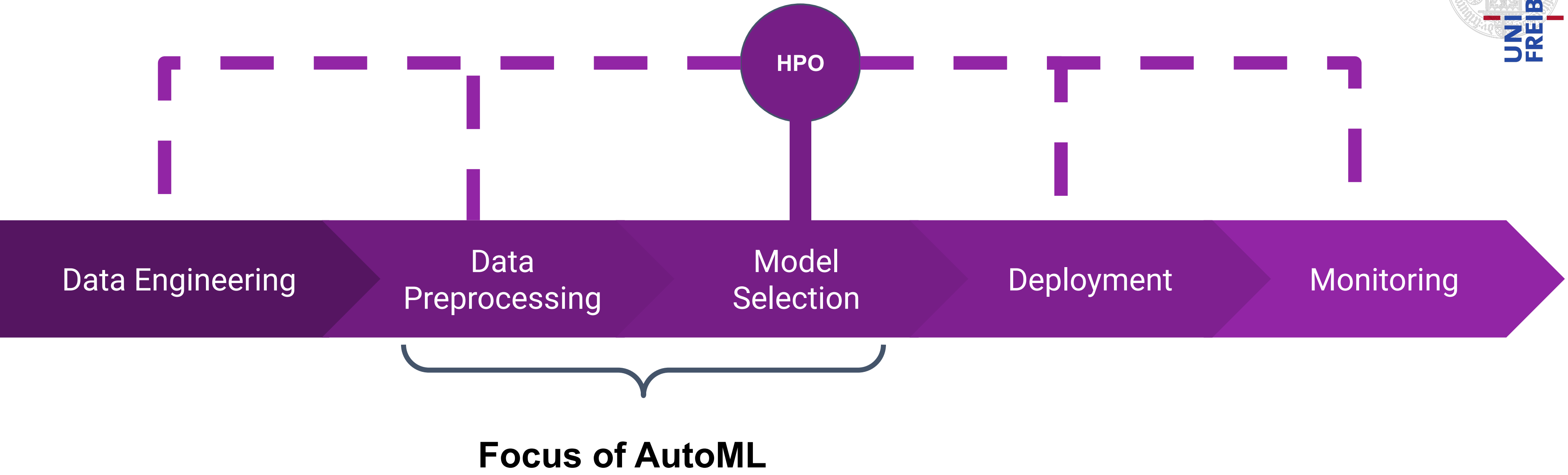
Hyperparameter Optimization



Optimize for

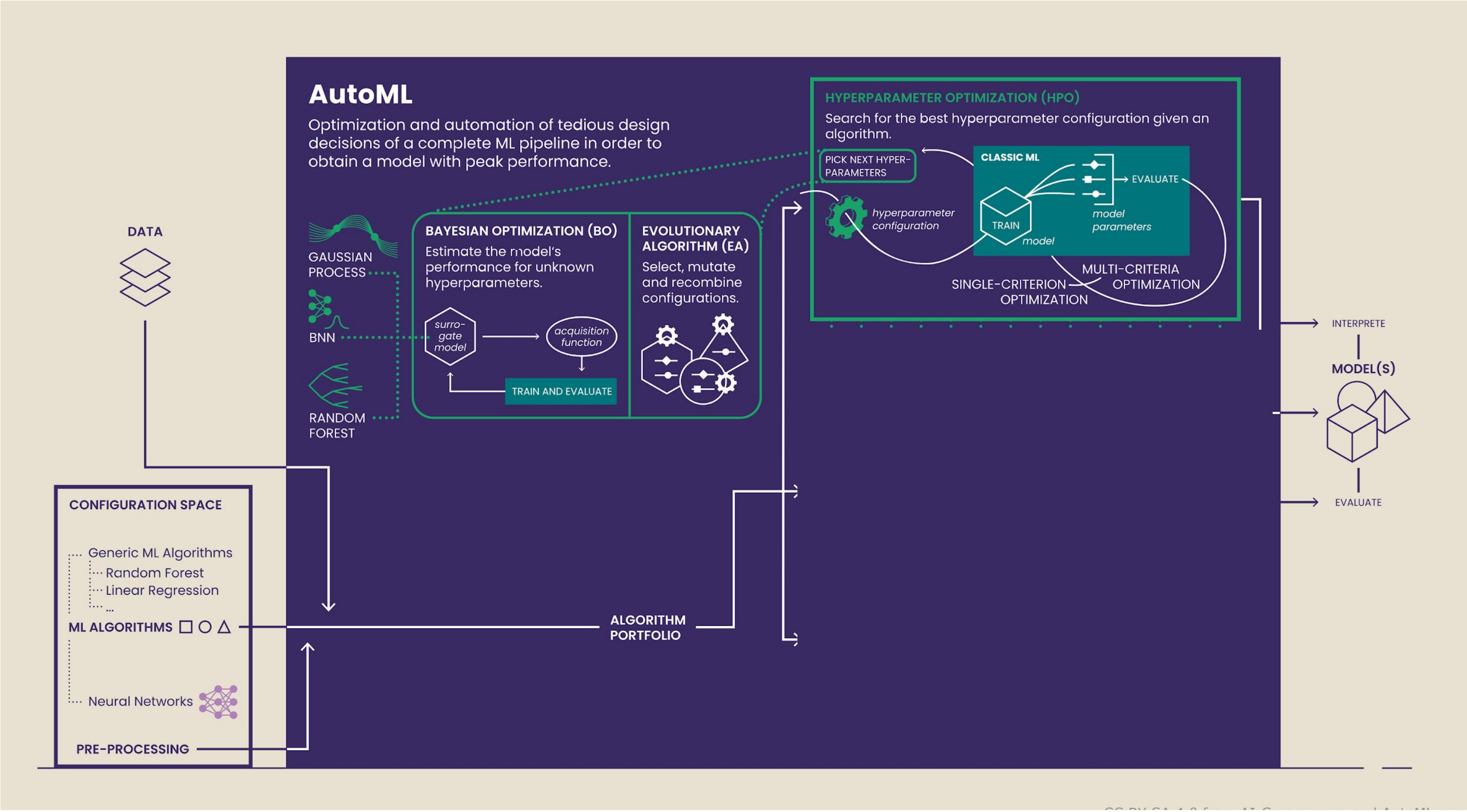
- Accuracy (& co)
- Memory consumption
- Energy consumption
- Inference time
- Training time
- Fairness
- Robustness
- Uncertainty quantification
- …

HPO → AutoML → AutoDS

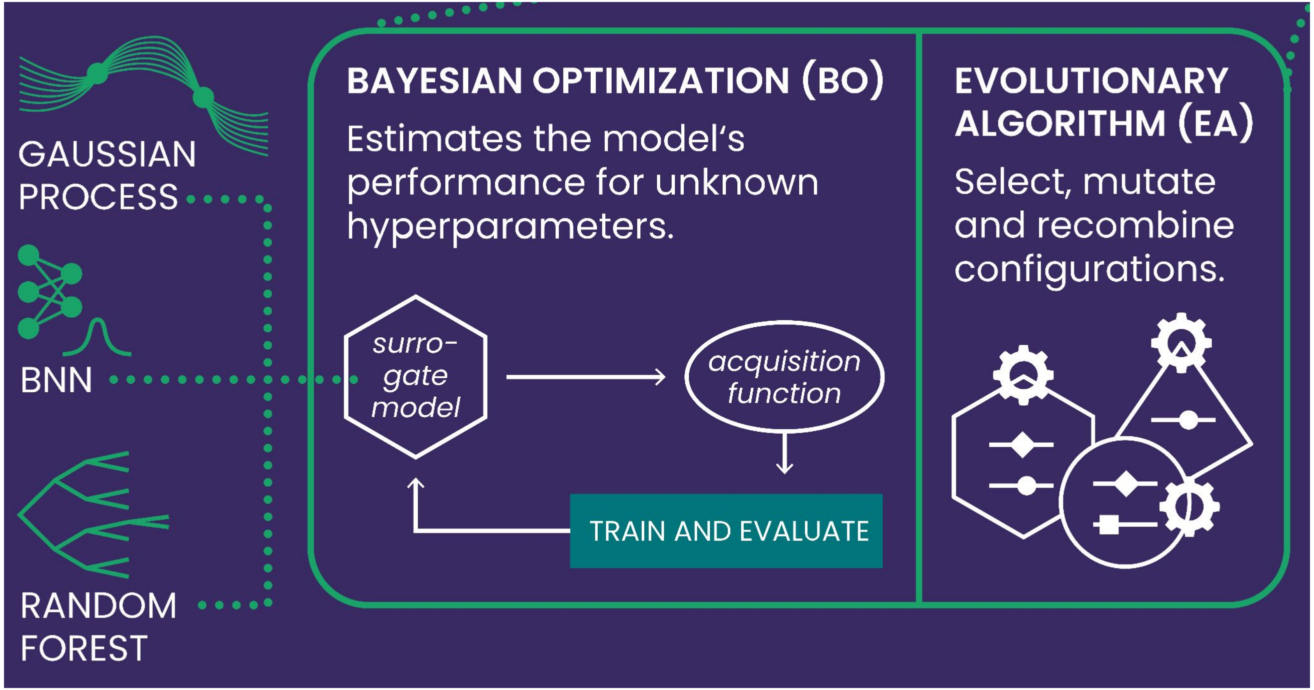


⇒ AutoDS
by Luc de Raedt

Optimizers for HPO



Optimizers for HPO

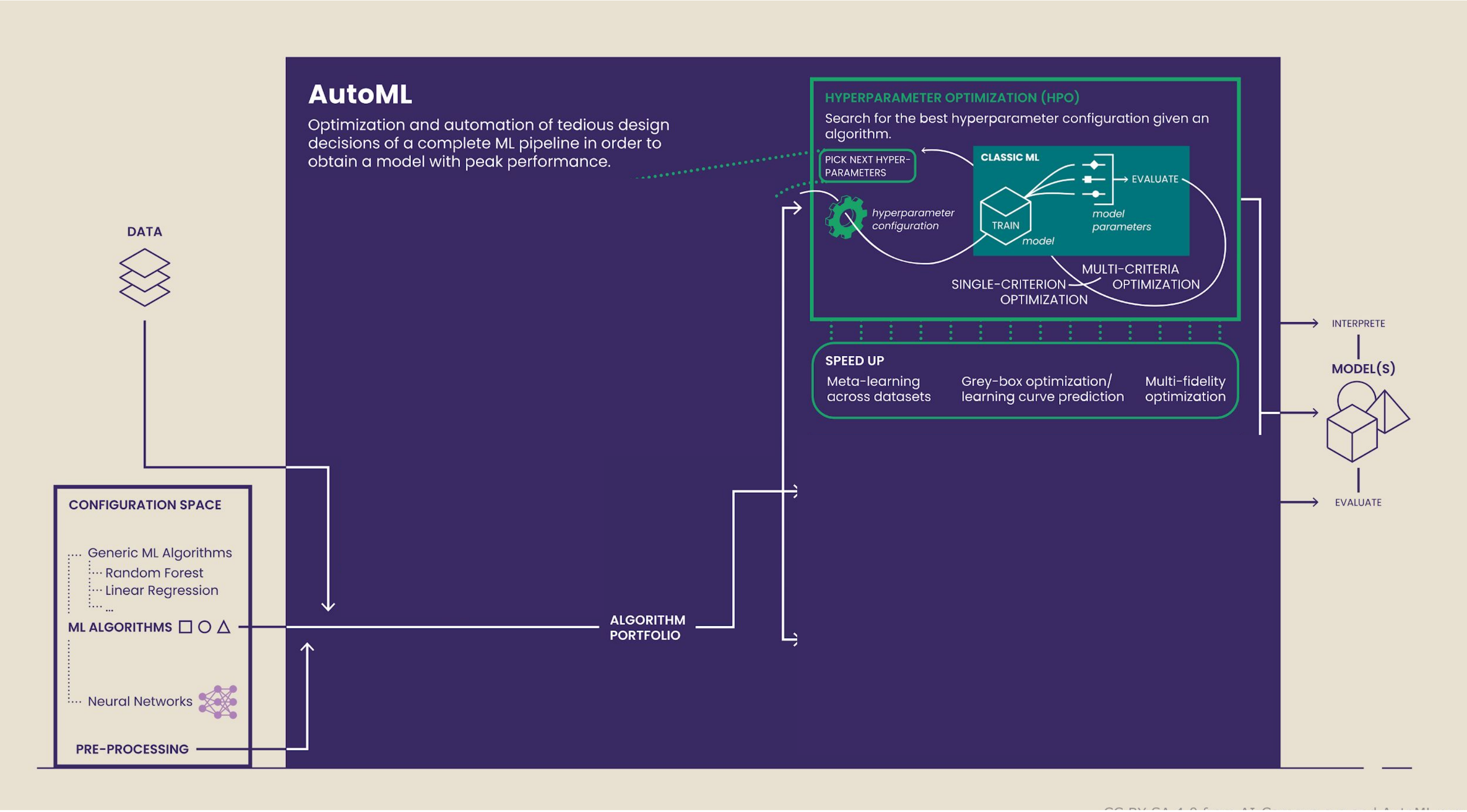


Further alternatives:

- Grid search
- Random search
- Reinforcement Learning
- Planning

⇒ H2O
by Erin LeDell

Speeding Up


AutoML
Optimization and automation of tedious design decisions of a complete ML pipeline in order to obtain a model with peak performance.
HYPERPARAMETER OPTIMIZATION (HPO)
Search for the best hyperparameter configuration given an algorithm.
PICK NEXT HYPER-PARAMETERS
CLASSIC ML
EVALUATE
hyperparameter configuration
TRAIN
model
model parameters
MULTI-CRITERIA OPTIMIZATION
SINGLE-CRITERION OPTIMIZATION
SPEED UP
Meta-learning across datasets
Grey-box optimization/ learning curve prediction
Multi-fidelity optimization
DATA
INTERPRETE
MODEL(S)
EVALUATE
CONFIGURATION SPACE
Generic ML Algorithms
Random Forest
Linear Regression
...
ML ALGORITHMS
Neural Networks
PRE-PROCESSING
ALGORITHM PORTFOLIO

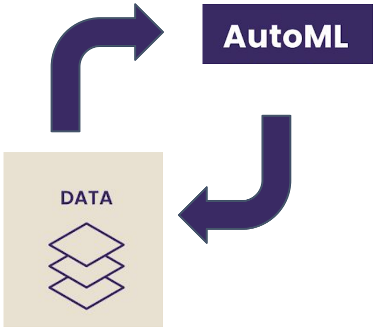
Speeding Up

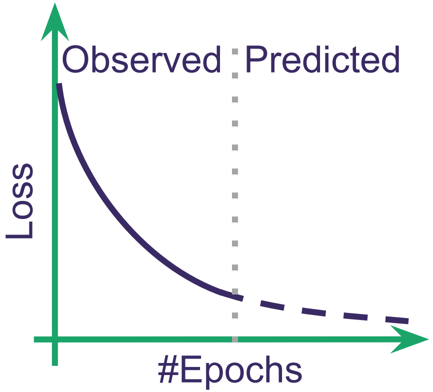
SPEED UP

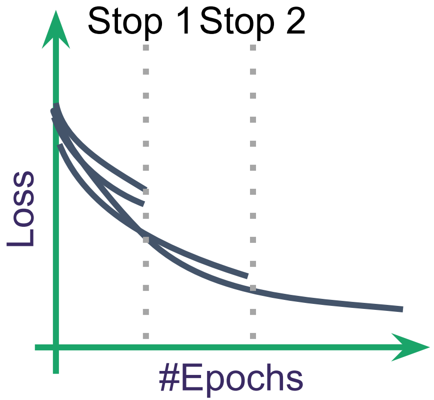
Meta-learning across datasets

Grey-box optimisation/ learning curve prediction

Multi-fidelity optimisation





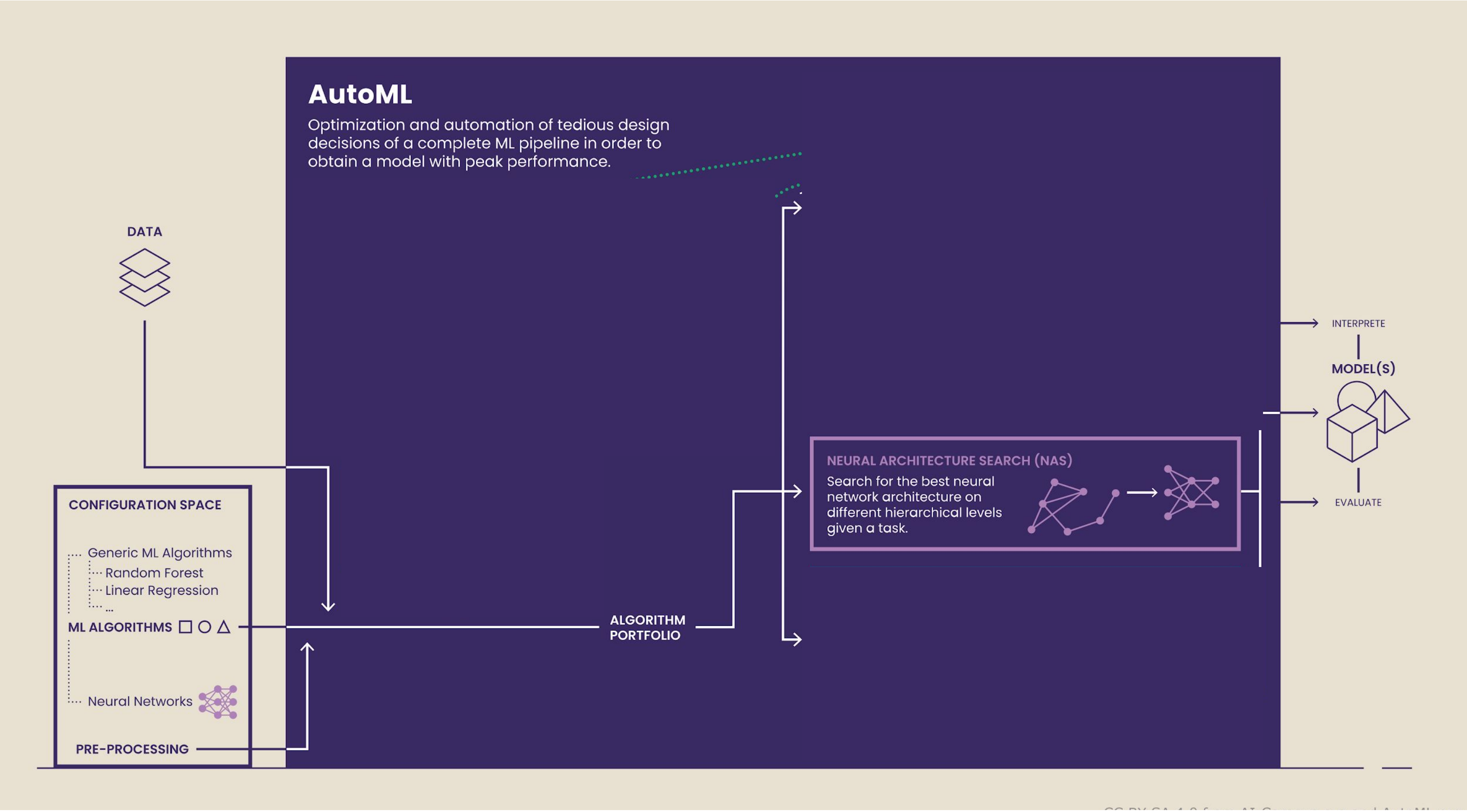


HPO Packages

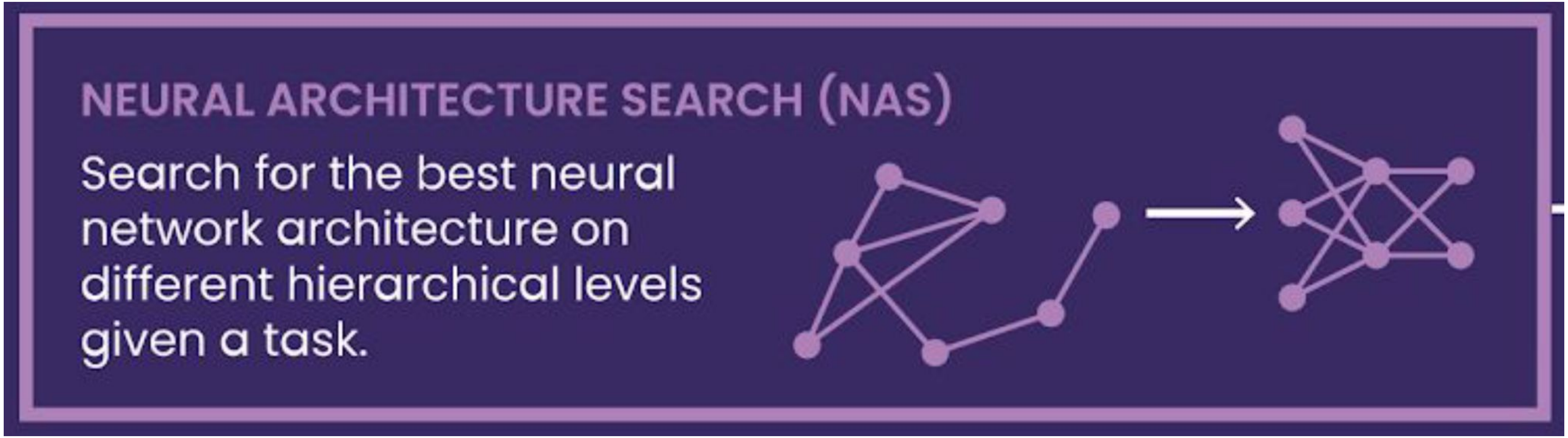
Package	Complex Hyperparameter Spaces	Multi-Objective	Multi-Fidelity	Instances	CLI	Parallelism
HyperMapper	✅	✅	❌	❌	❌	❌
Optuna	✅	✅	✅	❌	✅	✅
Hyperopt	✅	❌	❌	❌	✅	✅
BoTorch	❌	✅	✅	❌	❌	✅
OpenBox	✅	✅	❌	❌	❌	✅
HpBandSter	✅	❌	✅	❌	❌	✅
SMAC	✅	✅	✅	✅	✅	✅

last update of table in 2021

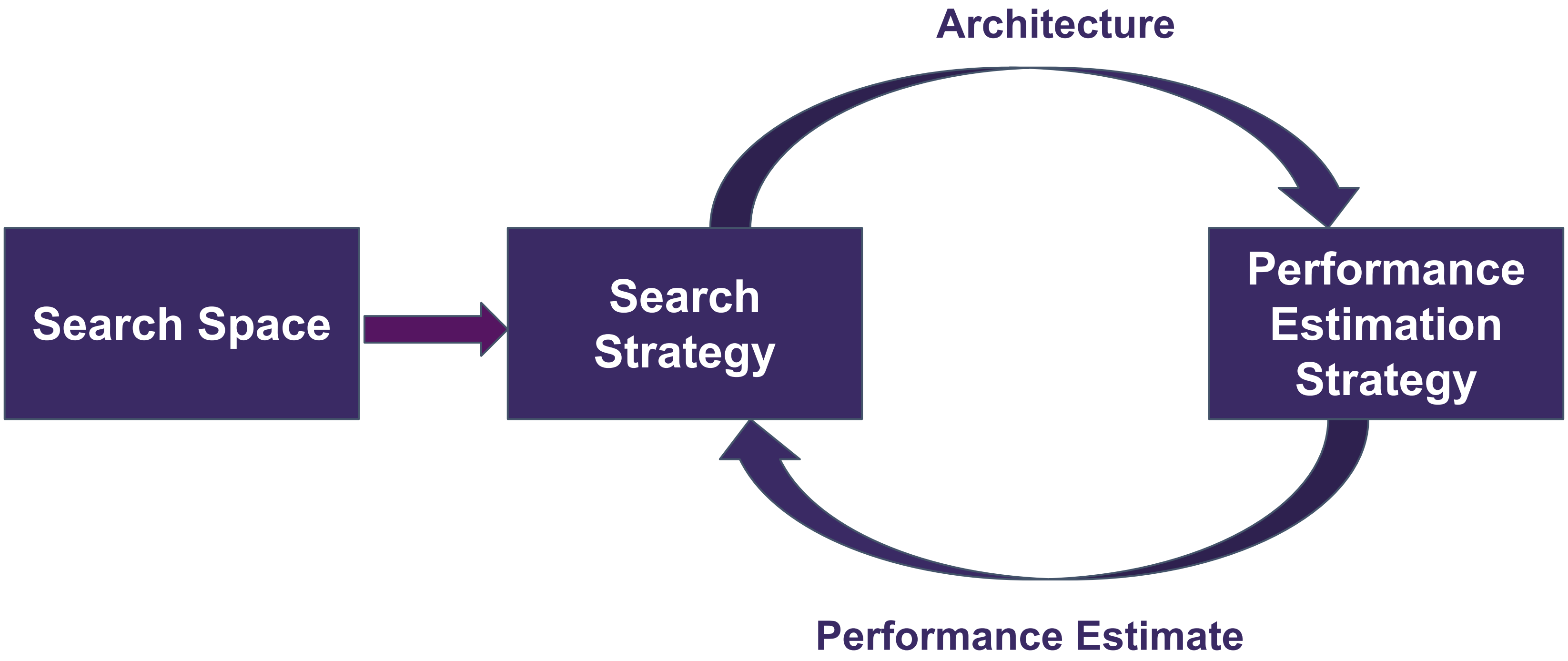
Neural Architecture Search (NAS)



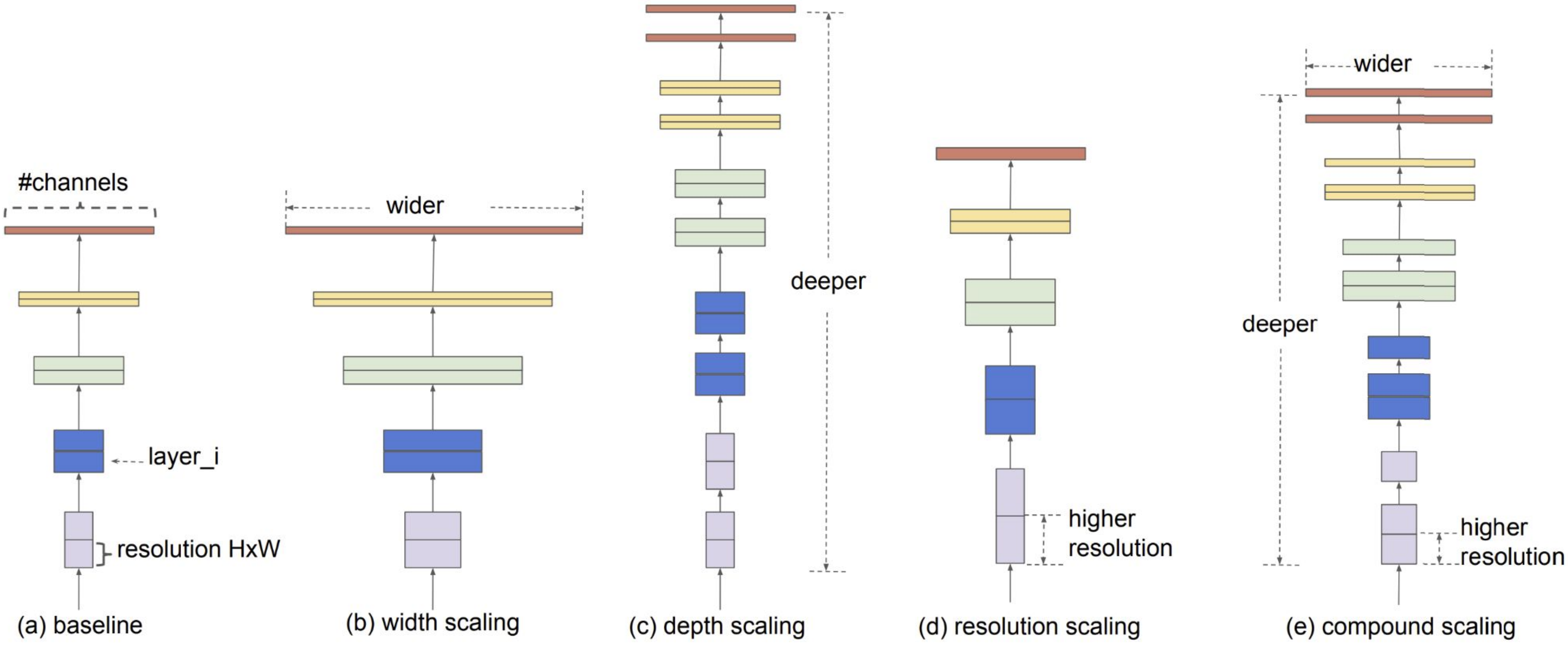
Neural Architecture Search (NAS)



The Components of NAS



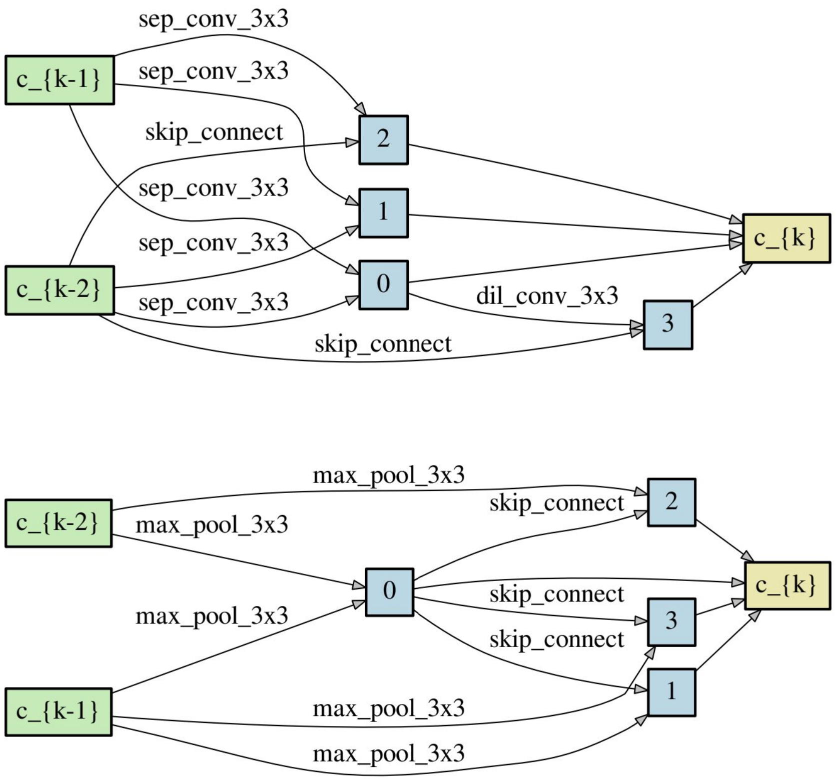
Search Space 1: Macro NAS



Source: [Tan & Le. 2019]

→ direct relationship to HPO: NAS as HPO

Search Space 2: Cell-based NAS



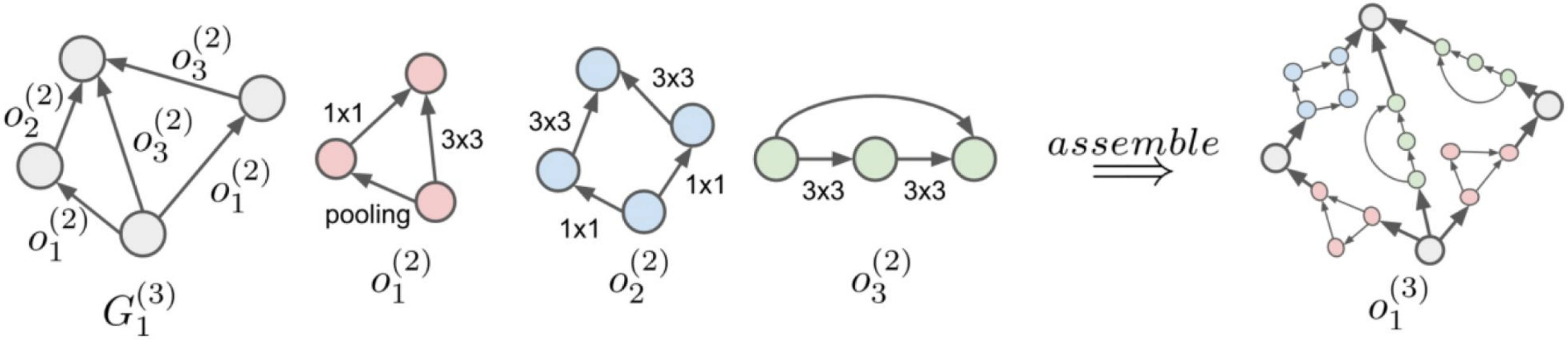
Source: [Liu et al. 2019]

Search Space 3: Hierarchical NAS

Search on multiple levels of the hierarchy

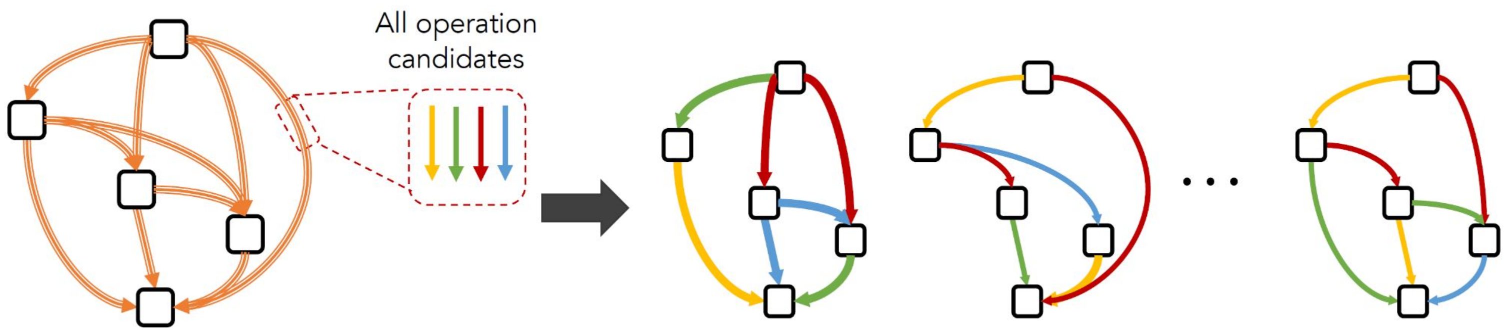
- **Lower levels:** create reusable building blocks
- **Higher levels:** combine building blocks

Like transformers are composed of lower-level building blocks (e.g., attention)



Source: [Liu et al, 2018]

Oneshot NAS: Weight Sharing Across Architectures



- For each choice between operations, the supernet includes all of them
- A linear number of weights shared by an exponential number of architectures
- Thus, updating the weights of one architecture simultaneously updates parts of the weights of exponentially many other architectures

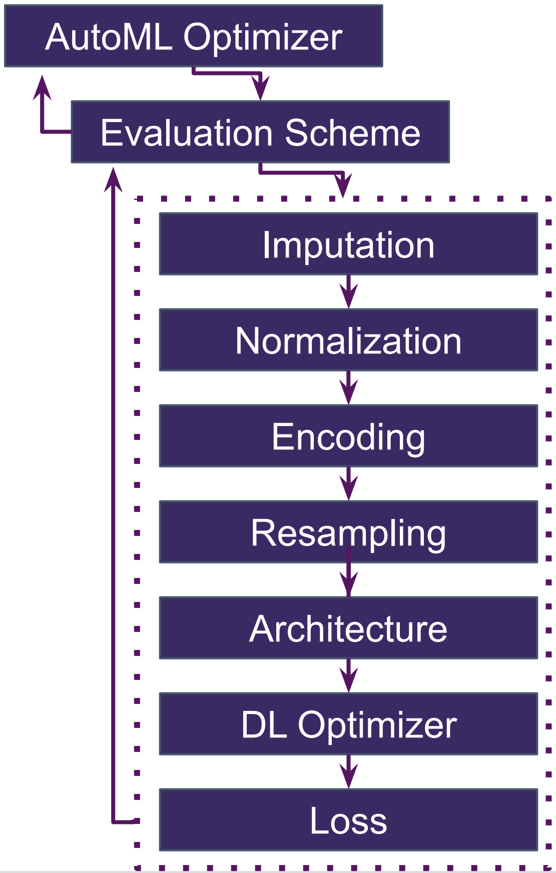
Zero-Cost Proxies for NAS

Very hot topic in NAS, but no consistent improvements over trivial baselines, such as #parameters or FLOPs

⇒ NAS
by Colin White

⇒ NASlib
by Rhea and Arjun

AutoDL: Joint NAS & HPO

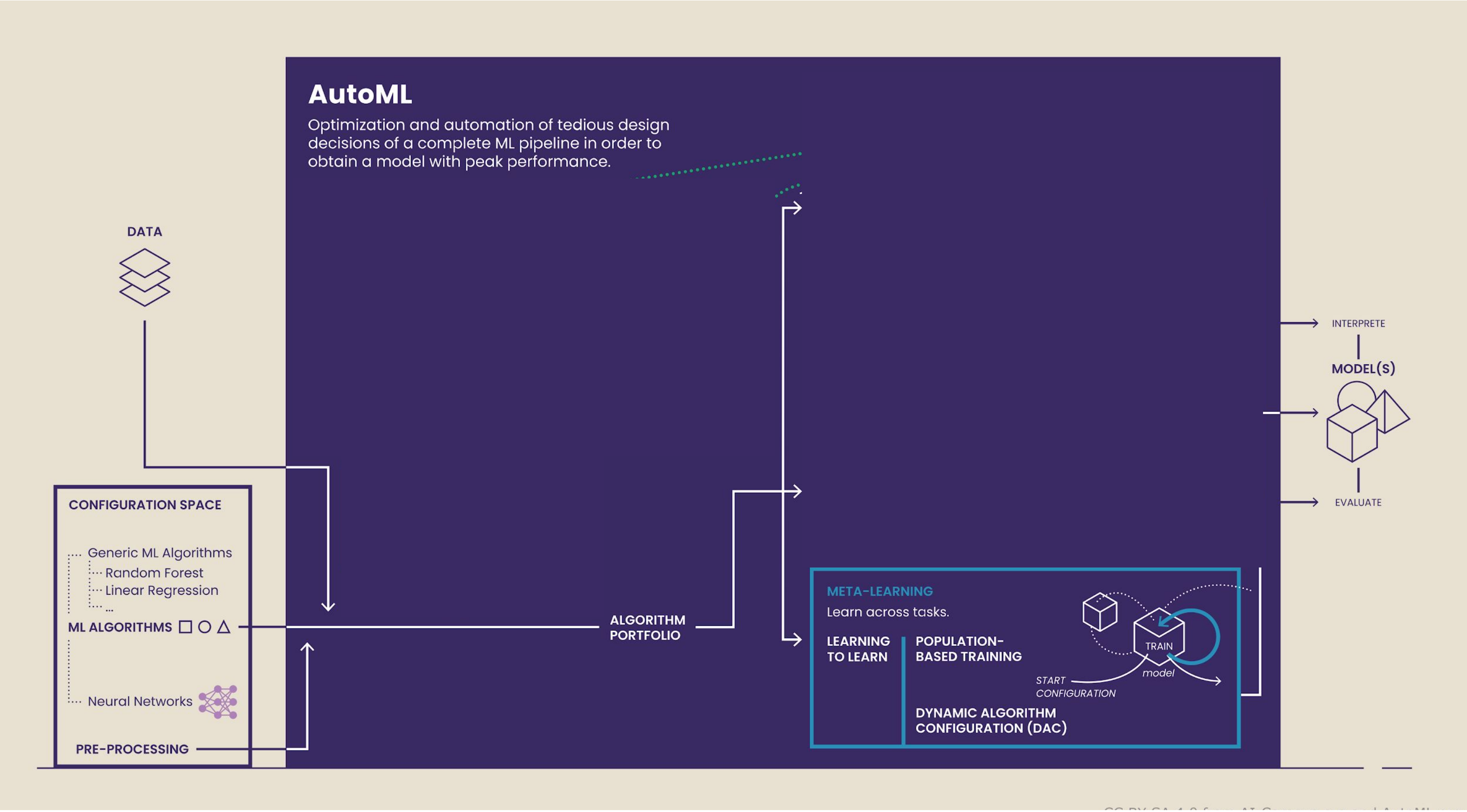


1. DL also includes complex pipelines
2. NAS & HPO need to go hand in hand

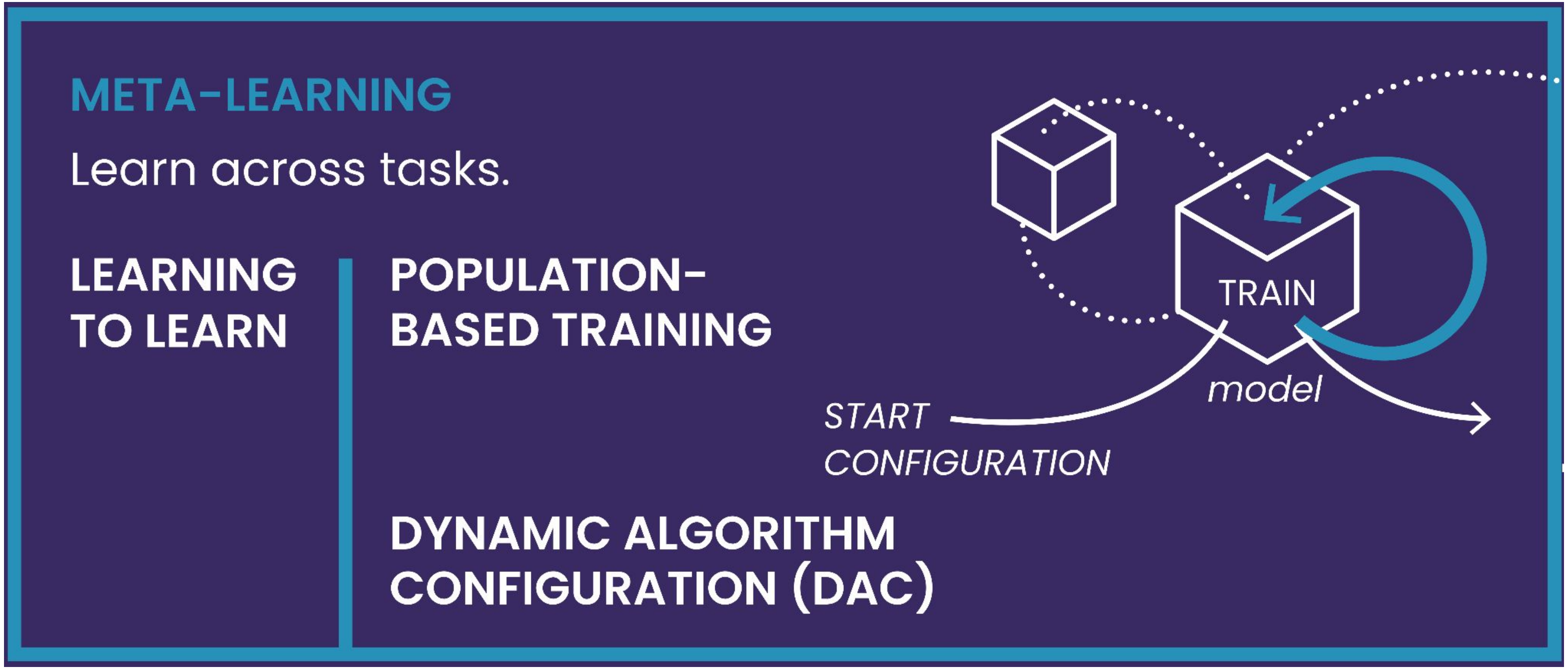
JAHS: Joint Architecture & Hyperparameter Search

Inspired by [Zimmer et al. 2021]

Meta-Learning



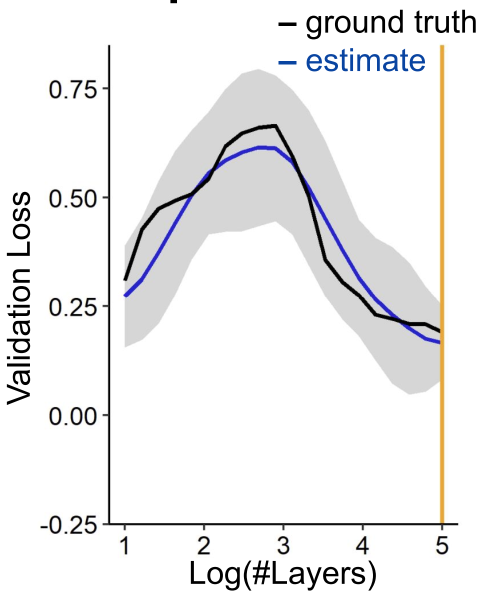
Meta-Learning



Learning about Learning Algorithms

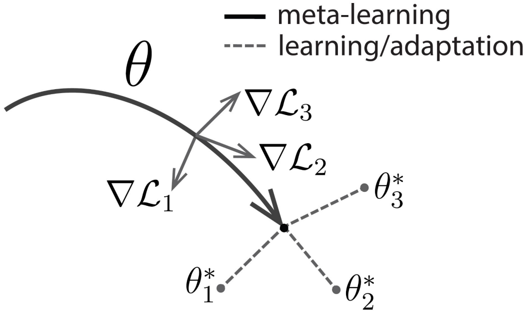
Performance prediction

Hyperparameter importance



Source: [Moosbauer et al. 2021]

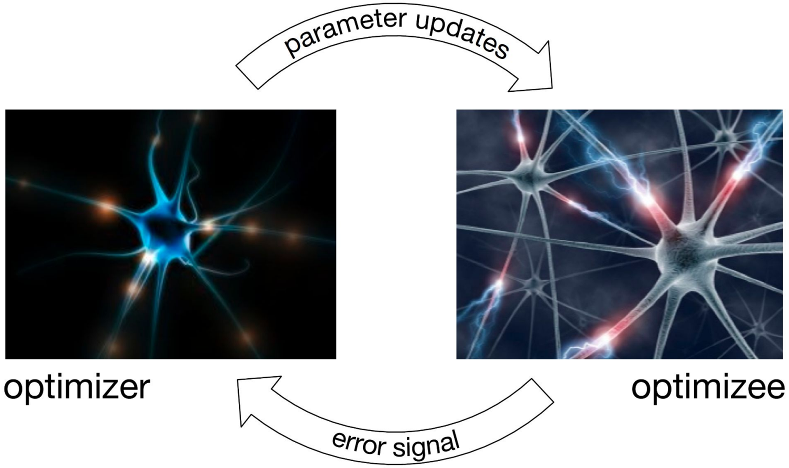
Learning NN weight initializations



Source: [Finn et al, 2017]

Learning to learn

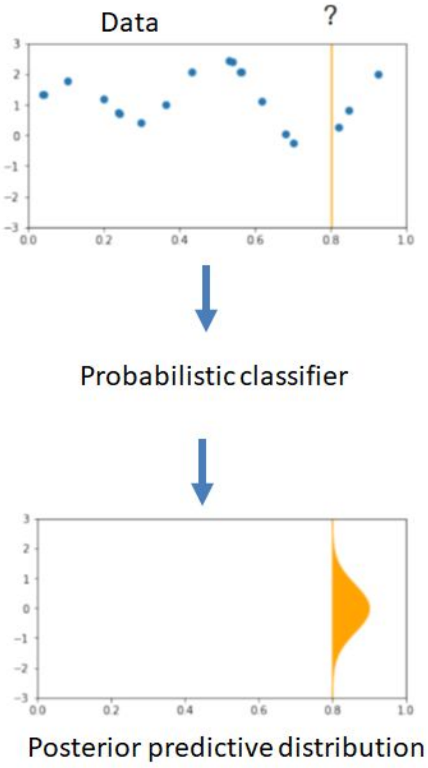
E.g., “Learning to learn by gradient descent by gradient descent“ [Chen et al. 2016]



Source: [Chen et al. 2016]

E.g., Alpha-Zero [Silver et al. 2017]

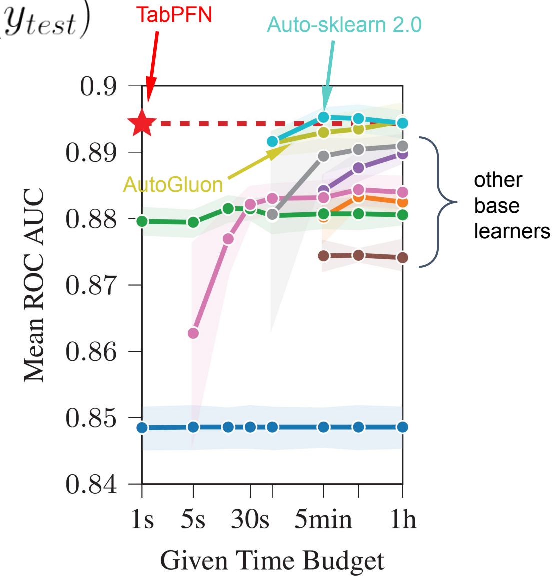
Learning to solve small tabular classification tasks



Learn a probabilistic classifier that accepts input $(X_{train}, \mathbf{y}_{train}, X_{test})$ and outputs $\mathrm{p}(y_{test})$

TabPFN [Müller et al, 2022]

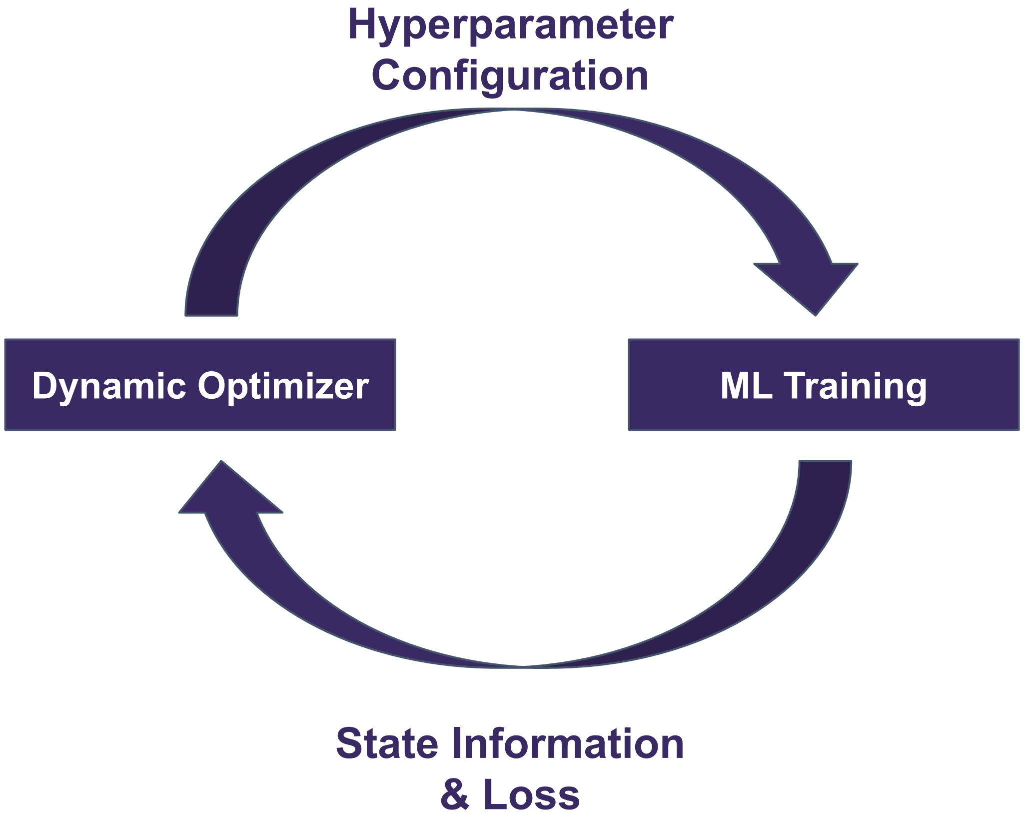
- Pretrain on millions of synthetic tabular datasets
- On a real dataset, simply apply a forward pass



SOTA predictions in < 1 second

Limitations: ≤ 1000 training data points, 100 features, 10 classes

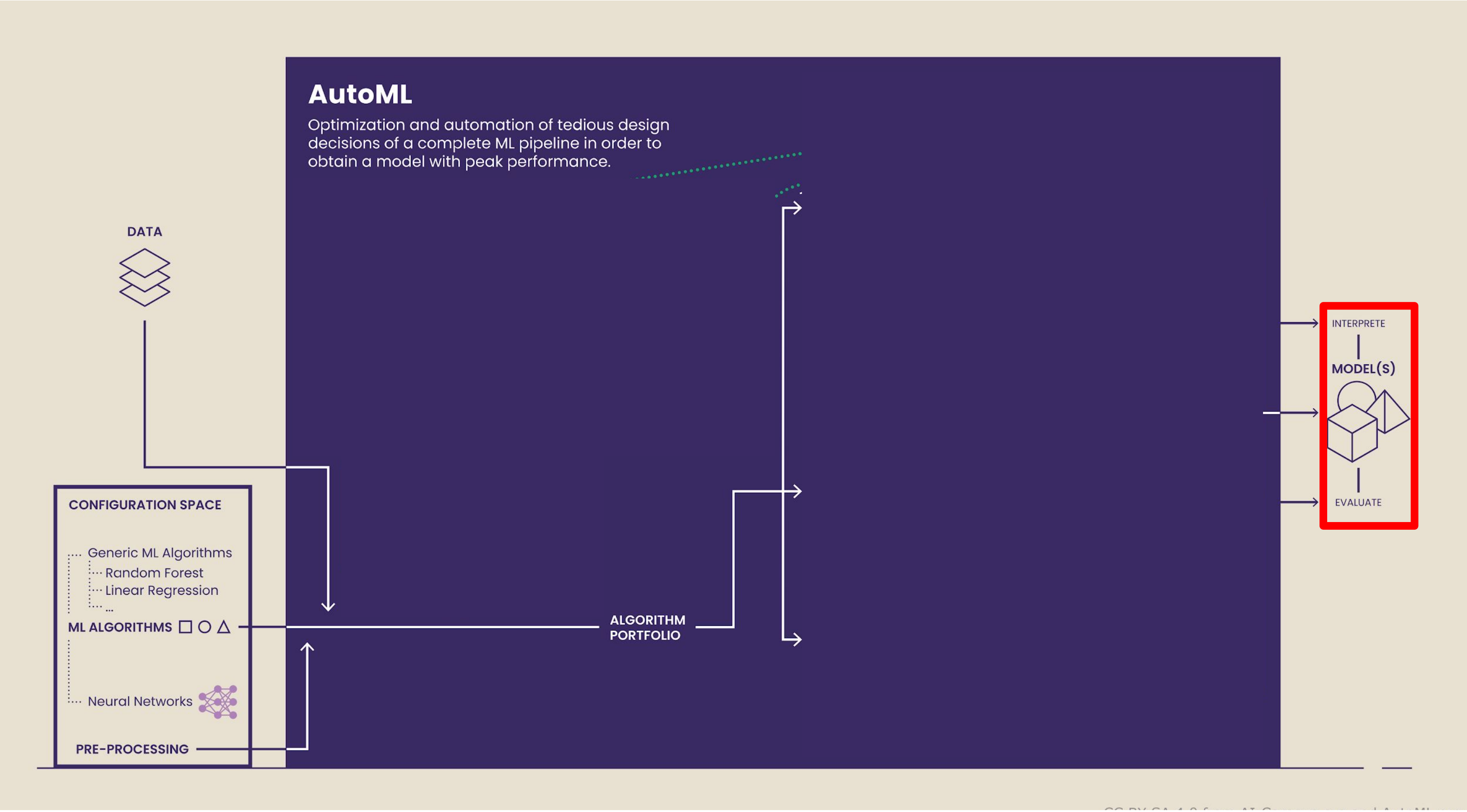
Dynamic AutoML



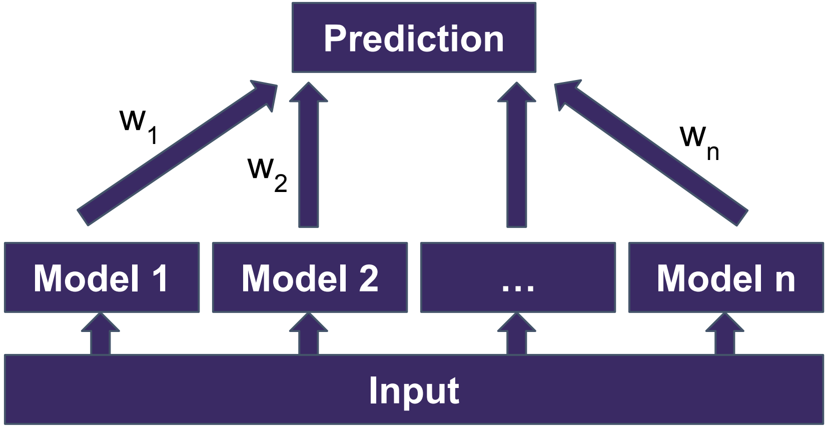
- Population-based Training [Jaderberg et al. 2017]
- Population-based Bandits [Parker-Holder et al. 2020]
- Dynamic Algorithm Configuration via RL [Biedenkapp et al. 2020, Adriaensen et al. 2022]

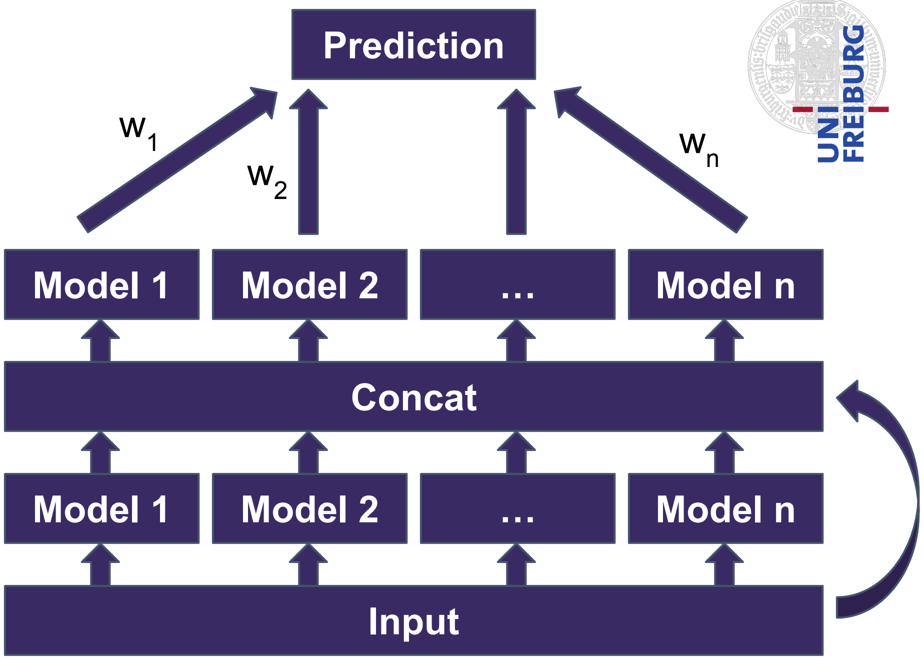
⇒ Dynamic Selection & Configuration by Carola Doerr

Final Step of AutoML



Ensembling vs Stacking

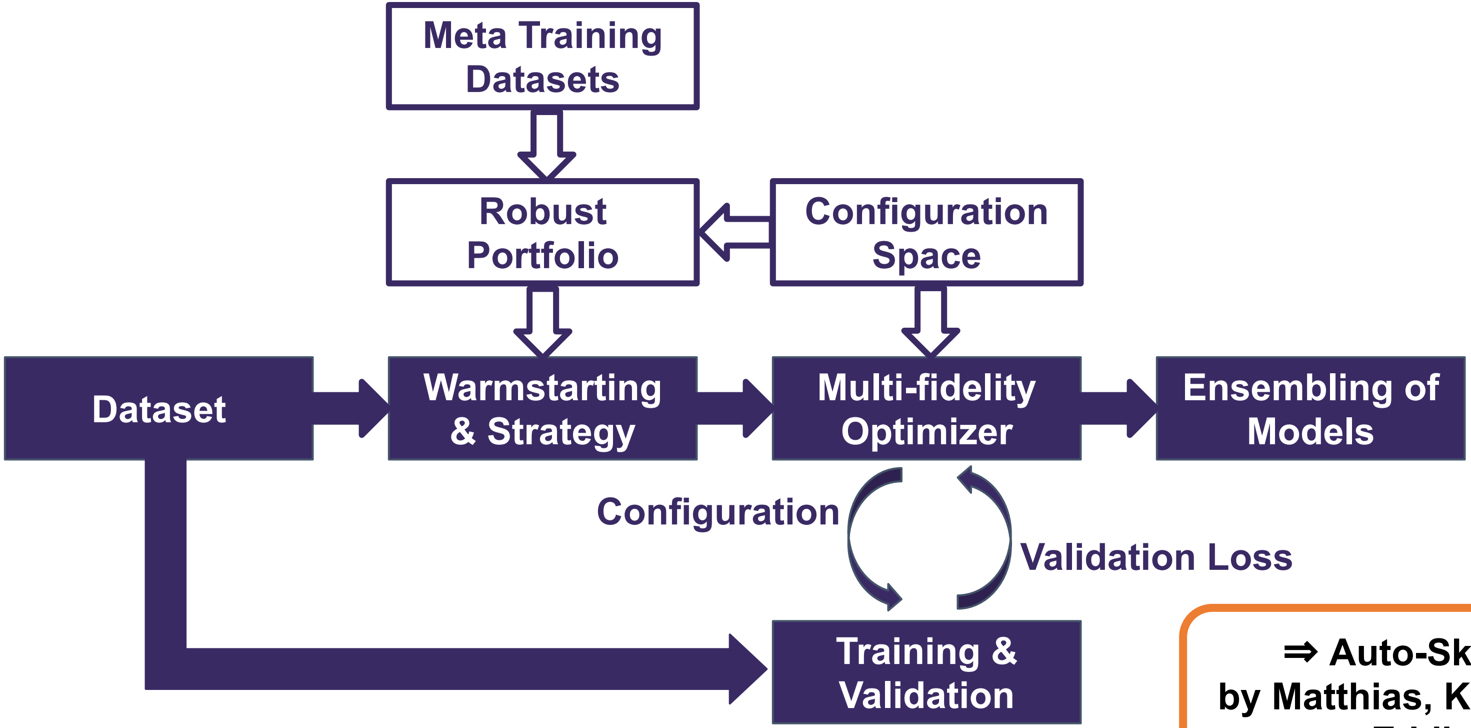




Source [Erickson et al. 2020]

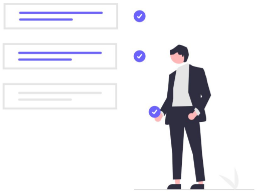
⇒ Auto-Gluon by Nick Erickson

Auto-Sklearn [Feurer et al. 2015, Feurer et al. 2022] & Auto-PyTorch [Zimmer et al. 2021]

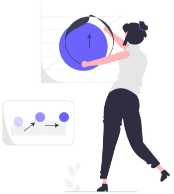


⇒ **Auto-Sklearn by Matthias, Katharina, Eddie**

AutoML-Ops



Determine your objectives, metrics and constraints



Design the configuration space

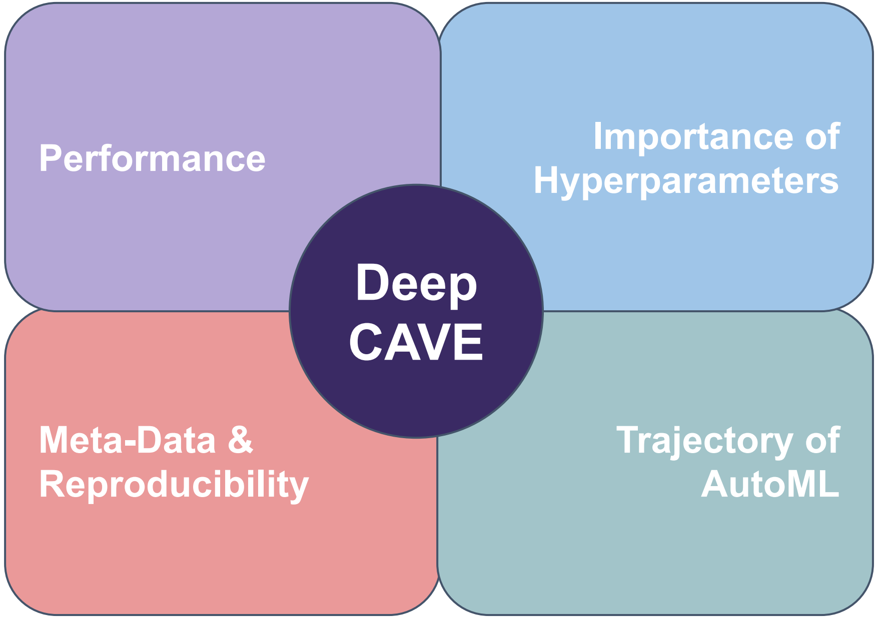
Choose your AutoML-Approach

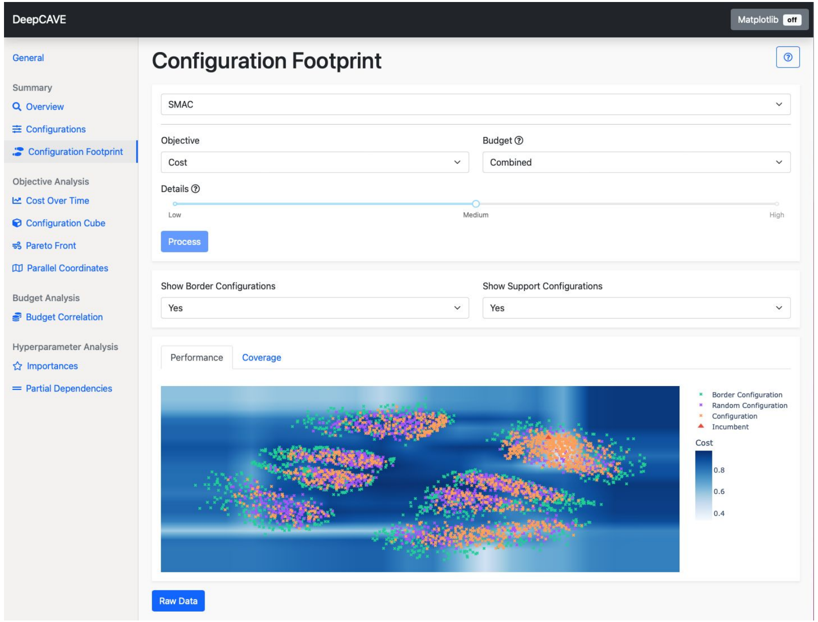
Determine Budgets

Running AutoML

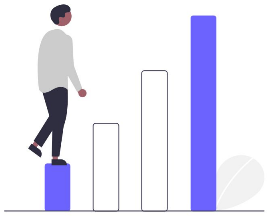
Monitor AutoML

Monitoring AutoML [Sass et al. 2022]

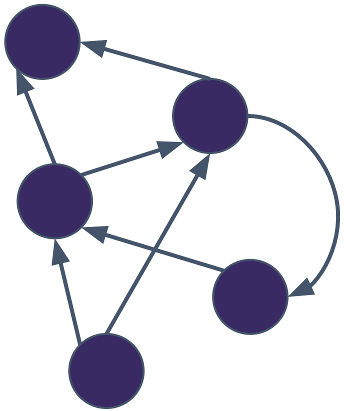




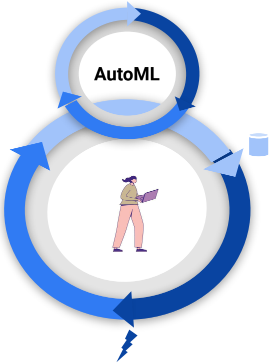
Selection of Open Challenges



Scaling up AutoML
for very large models

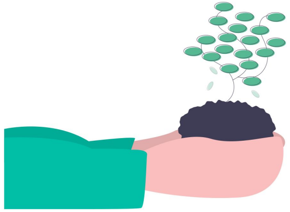


Finding substantially
novel architectures



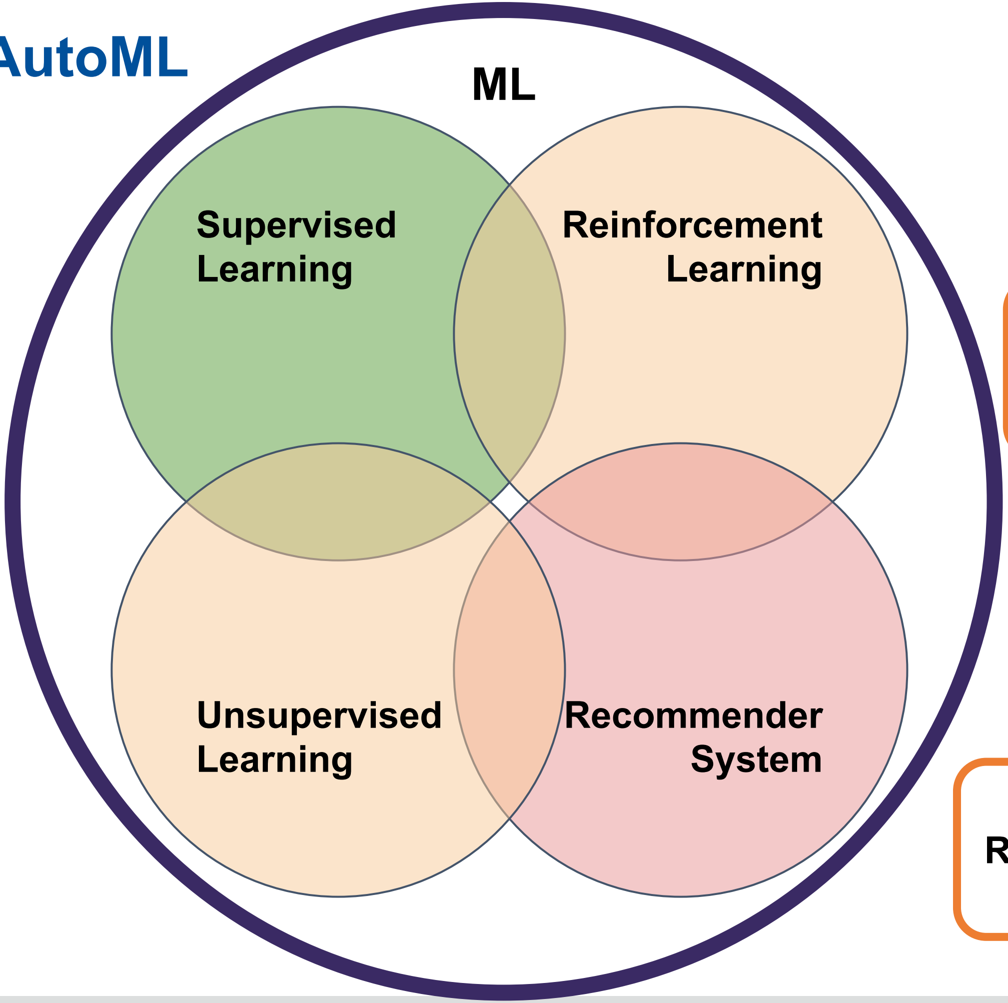
⇒ Expert Priors
by Luigi Nardi

Human-centered AutoML



Green AutoML

Maturity of AutoML



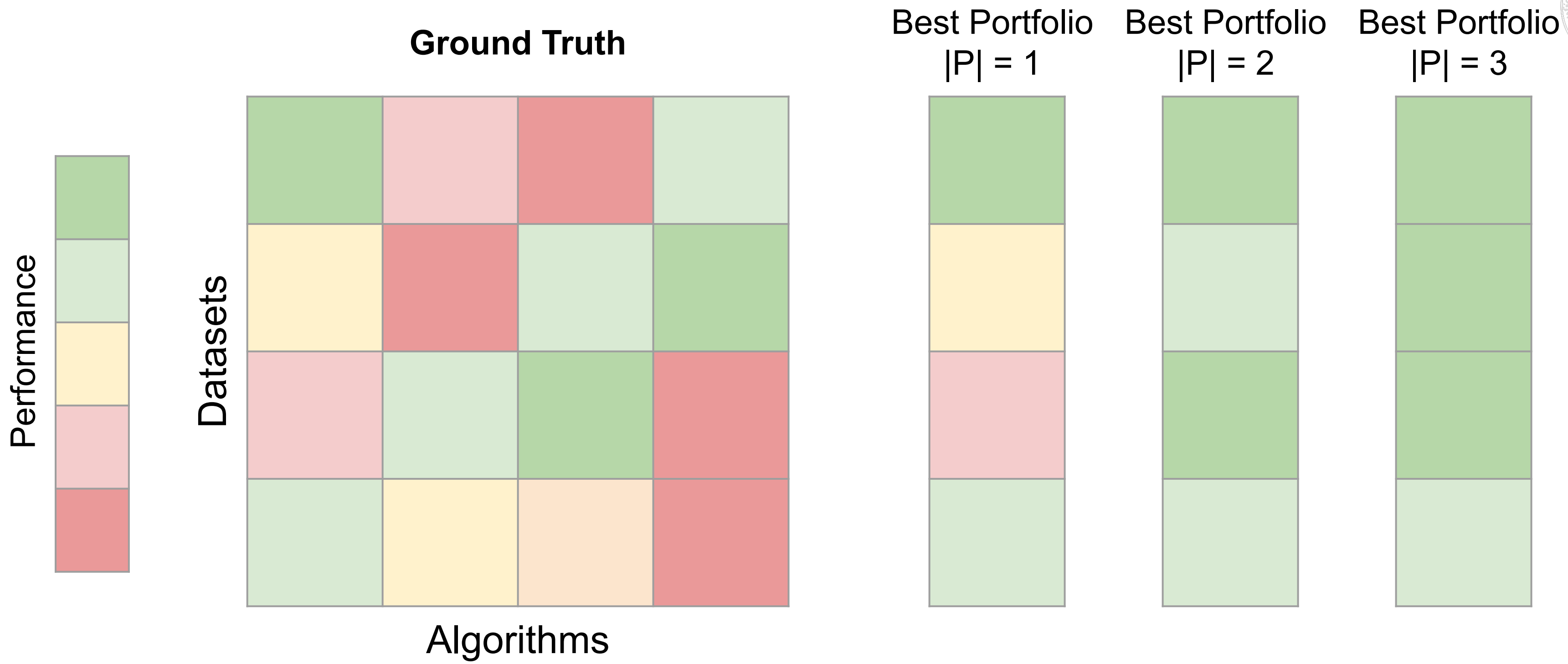
⇒ AutoRL
by Aleksandra Faust

⇒ AutoML &
Recommender Systems
by Joeran Beel

Have Fun with the AutoML Fall School 2022!

Backup slides

Portfolios for Warmstarting [Feurer et al. 2022]



Zero-Cost Proxies for NAS

ZC proxies are a particular type of performance predictor

- They aim to judge the performance of an architecture in a few seconds
- Often by a single forward pass on a mini-batch
- Thus, the term “zero-cost”

Examples

- Change of error when dropping network weights
- Dissimilarity of activation patterns for points in a batch

Very hot topic in NAS, but no consistent improvements over using number of parameters or FLOPS