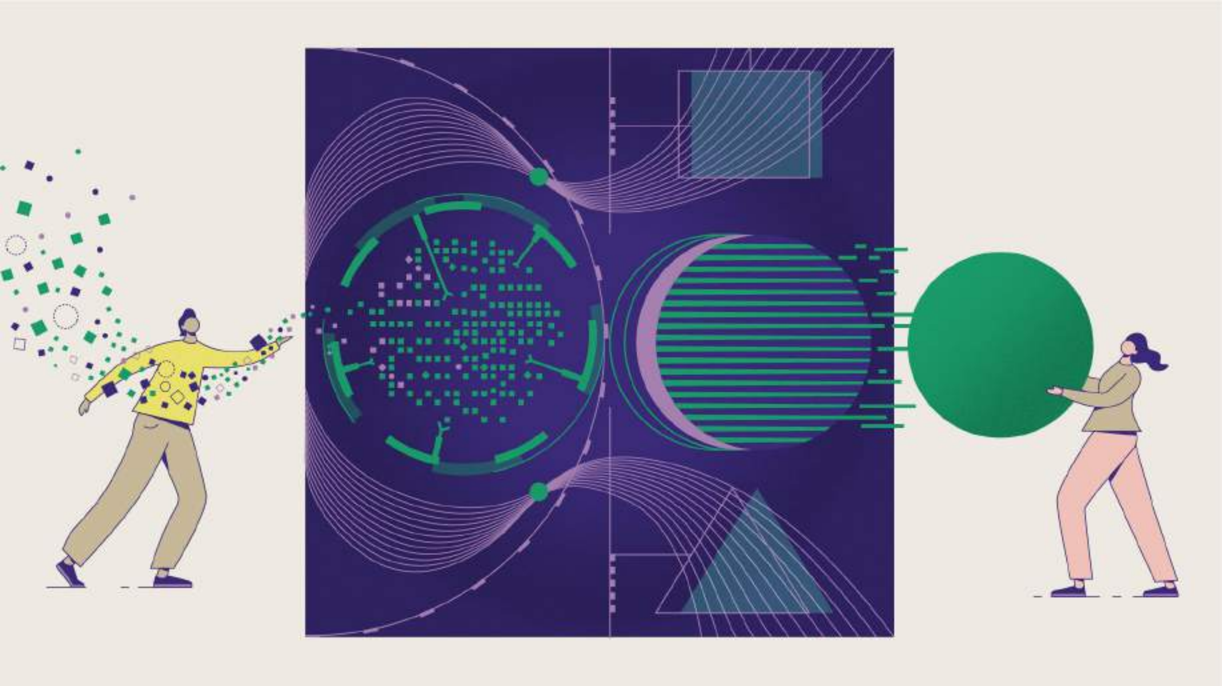


Efficient and Explainable AutoML

Marius Lindauer

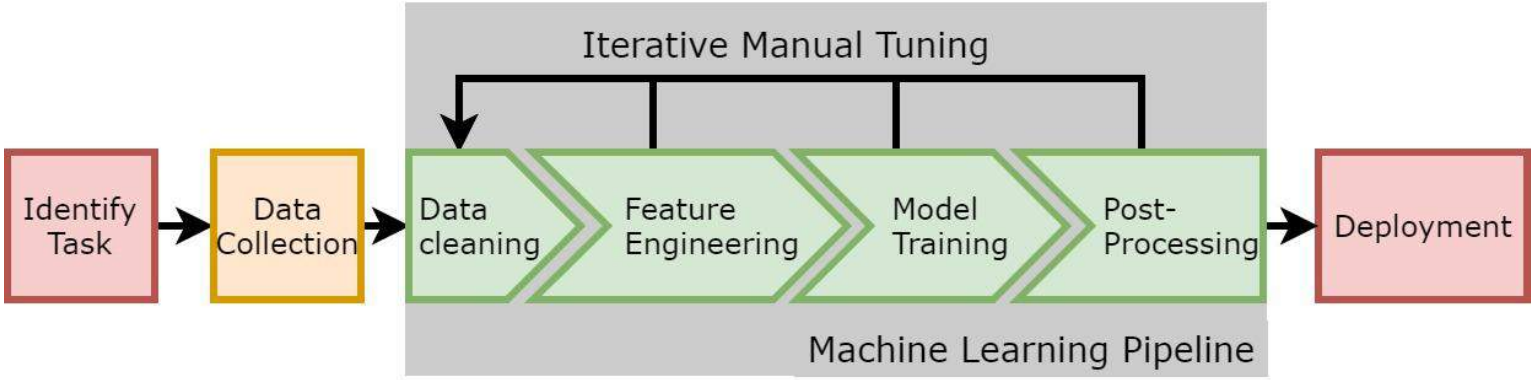
Leibniz University Hannover

Big Success of AI in Recent Years

Images courtesy of Shutterstock

From Manual to Automated Machine Learning

Automated Machine Learning Hannover



Mission Statement: Enabling users to efficiently apply ML!

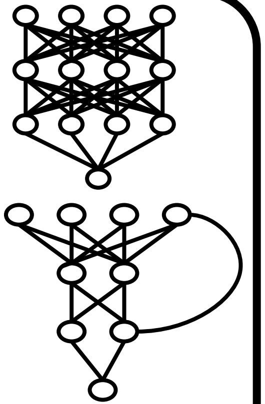
Design Decisions taken care by AutoML

Automated Machine Learning Hannover

AutoML.org

classifier	#λ
AdaBoost (AB)	4
Bernoulli naïve Bayes	2
decision tree (DT)	4
extreml. rand. trees	5
Gaussian naïve Bayes	-
gradient boosting (GB)	6
kNN	3
LDA	4
linear SVM	4
kernel SVM	7
multinomial naïve Bayes	2
passive aggressive	3
QDA	2
random forest (RF)	5
Linear Class. (SGD)	10

Algorithms



Architecture Design

preprocessor	#λ
extreml. rand. trees prepr.	5
fast ICA	4
feature agglomeration	4
kernel PCA	5
rand. kitchen sinks	2
linear SVM prepr.	3
no preprocessing	-
nystroem sampler	5
PCA	2
polynomial	3
random trees embed.	4
select percentile	2
select rates	3
one-hot encoding	2
imputation	1
balancing	1
rescaling	1

Pre-processing

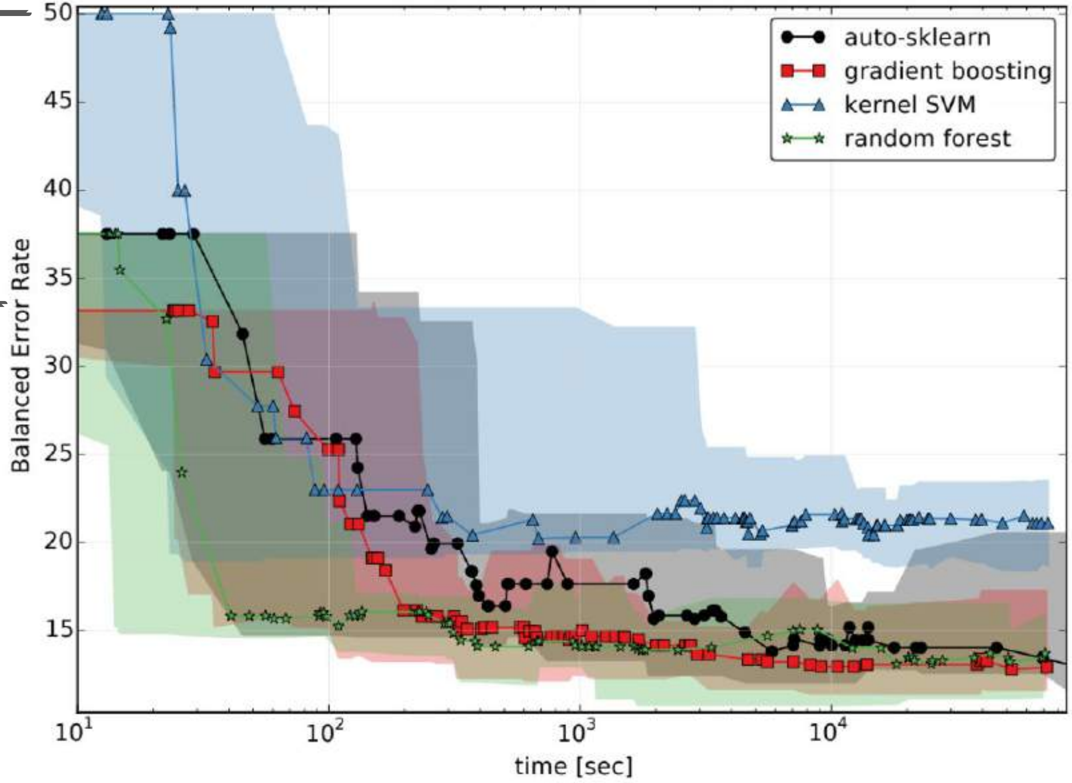
Hyper-parameters

...

Using AutoML Matters! (example on a specific dataset)

Automated Machine Learning Hannover

Choosing the correct algorithm
→ 17% improvement



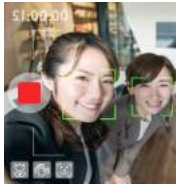
Optimized pipeline & hyperparameters
→ 20% - 29% improvement

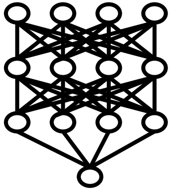
Exemplary Success Story:
Can DNNs outperform classical approaches on tabular data?

[Kadra et al. NeurIPS’21]

Previous Belief

1. Deep Neural Networks are especially well performing on high-dimensional data modalities (incl. images and text)

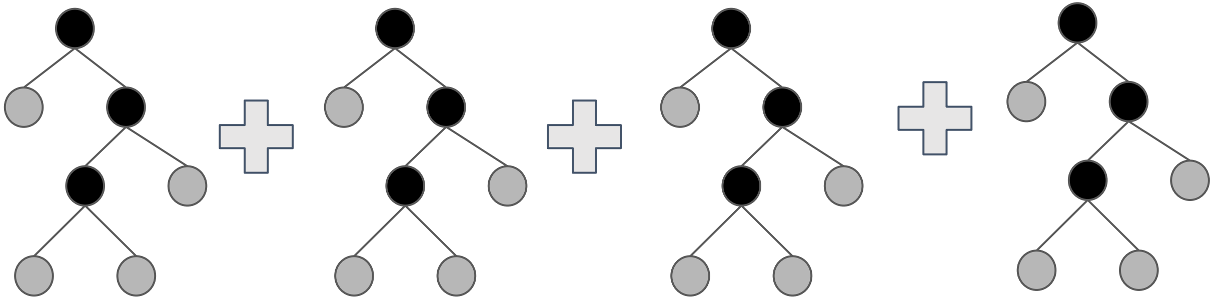




Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

2. Gradient-Boosted Decision Trees are state of the art on tabular data

	f_1	f_2	f_3	f_4	y
o_1	1	2	323	21	1
o_2	42	32	??	??	1
o_3	223	21	234	12	0
o_4	234	??	234	423	0
o_5	3	23	232	66	1



Idea I: Regularization against overfitting

Automated Machine Learning Hannover

AutoML.org

- DNNs are overparameterized and tabular datasets are often much smaller than image or text datasets
- → Sufficiently strong regularization could lead to better performance of DNNs?

- There are many regularization techniques for DNNs, incl.
  - Weight Decay [Krogh & Hertz 1991]
  - Dropout [Srivastava et al. 2014]
  - Batch Normalization [Ioffe & Szegedy 2015]
  - FGSM Adversarial Learning [Goodfellow et al. 2015]
  - Skip Connection [He et al. 2016]
  - Snapshot Ensembles [Loshchilov & Hutter 2017]
  - Shake-Shake [Gastaldi 2017]
  - Cut-Out [Devries & Taylor 2017]
  - Stochastic Weight Averaging [Izmailov et al. 2018]
  - Shake-Drop [Yamada et al. 2018]
  - Mix-Up [Zhang et al. 2018]
  - Lookahead Optimizer [Zhang et al. 2019]
  - Cut-Mix [Yun et al. 2019]

- Most of them developed for other data modalities (often computer vision)

Idea II: Use AutoML to find a Regularization Cocktail

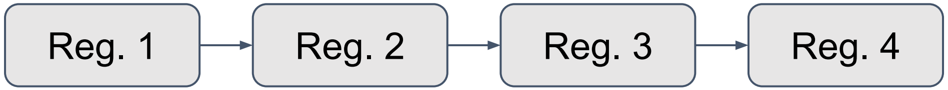
Automated Machine Learning Hannover
AutoML.org

Group	Regularizer	Hyperparameter	Type	Range	Conditionality
Implicit	BN	BN-active	Boolean	$\{\text{True}, \text{False}\}$	–
	SWA	SWA-active	Boolean	$\{\text{True}, \text{False}\}$	-
	LA	LA-active	Boolean	$\{\text{True}, \text{False}\}$	–
		Step size	Continuous	$[0.5, 0.8]$	LA-active
		Num. steps	Integer	$[5, 10]$	LA-active
W. Decay	WD	WD-active	Boolean	$\{\text{True}, \text{False}\}$	–
		Decay factor	Continuous	$[10^{-5}, 0.1]$	WD-active
Ensemble	DO	DO-active	Boolean	$\{\text{True}, \text{False}\}$	–
		Dropout shape	Nominal	{funnel, long funnel, diamond, hexagon, brick, triangle, stairs}	DO-active
		Drop rate	Continuous	$[0.0, 0.8]$	DO-active
	SE	SE-active	Boolean	$\{\text{True}, \text{False}\}$	-
Structural	SC	SC-active	Boolean	$\{\text{True}, \text{False}\}$	–
		MB choice	Nominal	{SS, SD, Standard}	SC-active
	SD	Max. probability	Continuous	$[0.0, 1.0]$	SC-active $\wedge$ MB choice = SD
	SS	-	-	-	SC-active $\wedge$ MB choice = SS
Augmentation	–	Augment	Nominal	{MU, CM, CO, AT, None}	–
	MU	Mix. magnitude	Continuous	$[0.0, 1.0]$	Augment = MU
	CM	Probability	Continuous	$[0.0, 1.0]$	Augment = CM
	CO	Probability	Continuous	$[0.0, 1.0]$	Augment = CO
		Patch ratio	Continuous	$[0.0, 1.0]$	Augment = CO
	AT	-	-	-	Augment = AT

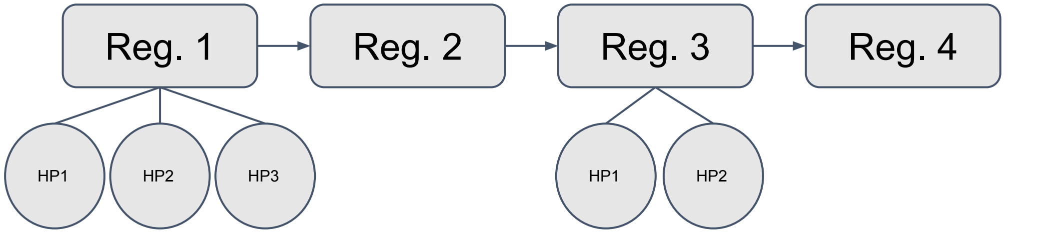
Challenges for AutoML

1. Training DNNs is fairly expensive

2. Pipeline of regularization techniques

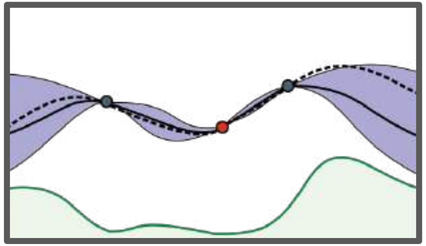


3. Each regularizer has its own hyperparameter → structured space



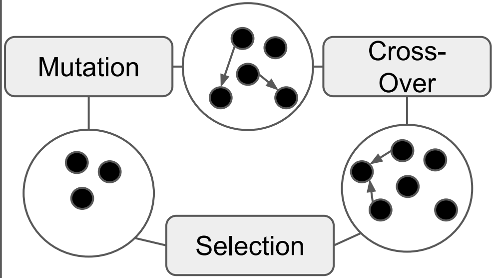
AutoML Techniques

Bayesian Optimization



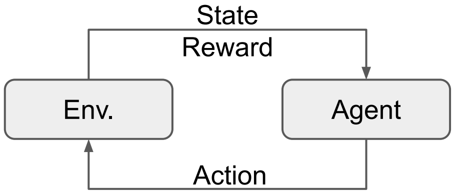
- Global optimization strategy
- Very sample efficient
- Very efficient for small/med. config. spaces

Evolutionary Algorithms



- Population based-approach
- Strong performance for longer budgets
- Easy to parallelize

Reinforcement Learning



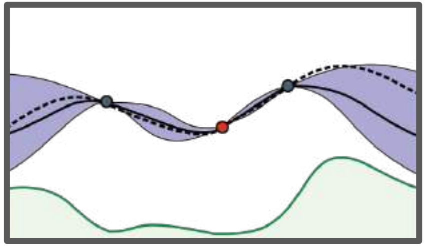
- Learning of a policy
- Can learn a generalizable policy
- Human-like approach

Planning

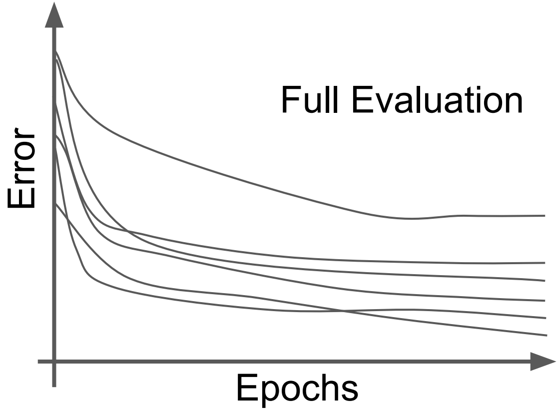
...

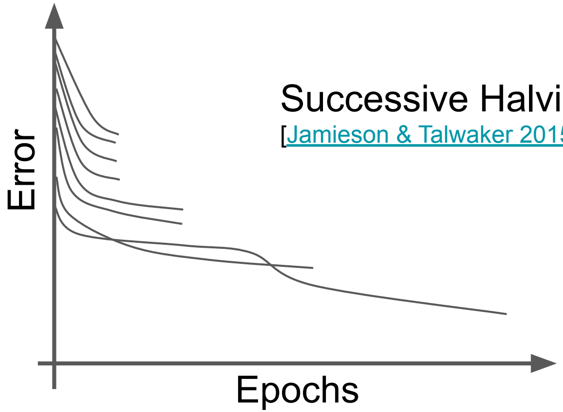
BOHB: Bayesian Optimization + Hyperband
[Falkner et al. 2018]

Bayesian Optimization



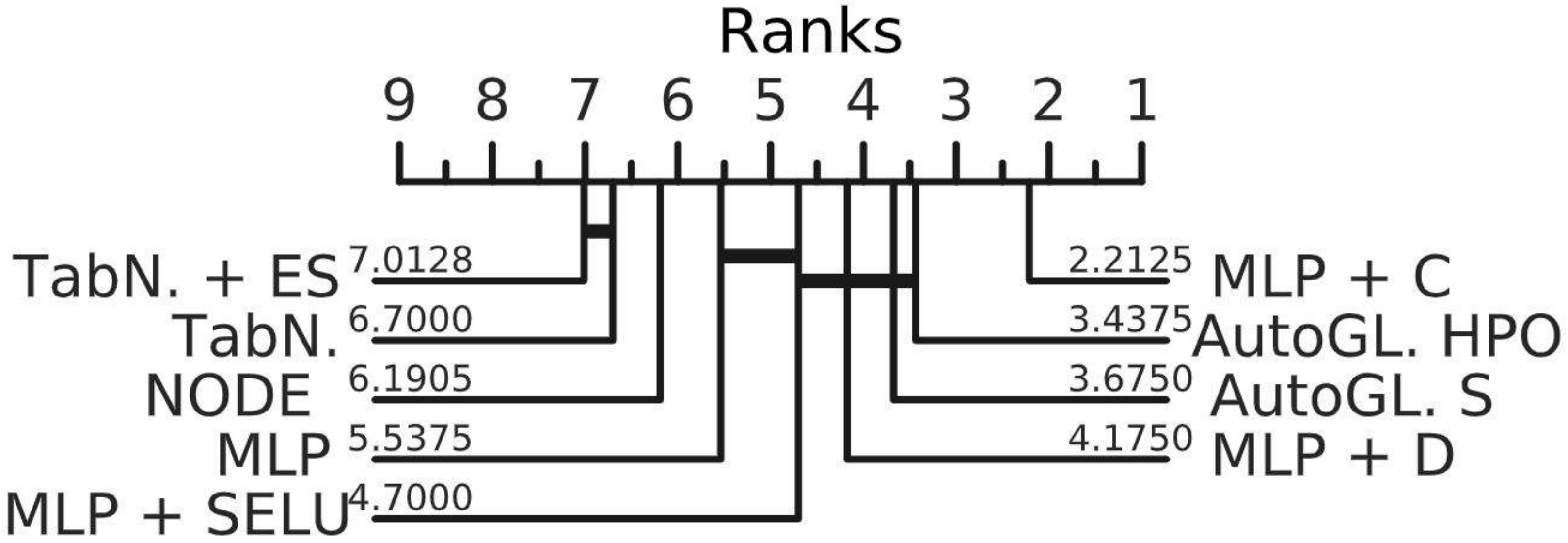
- Global optimization strategy
- Very sample efficient
- Very efficient for small/med. config. spaces



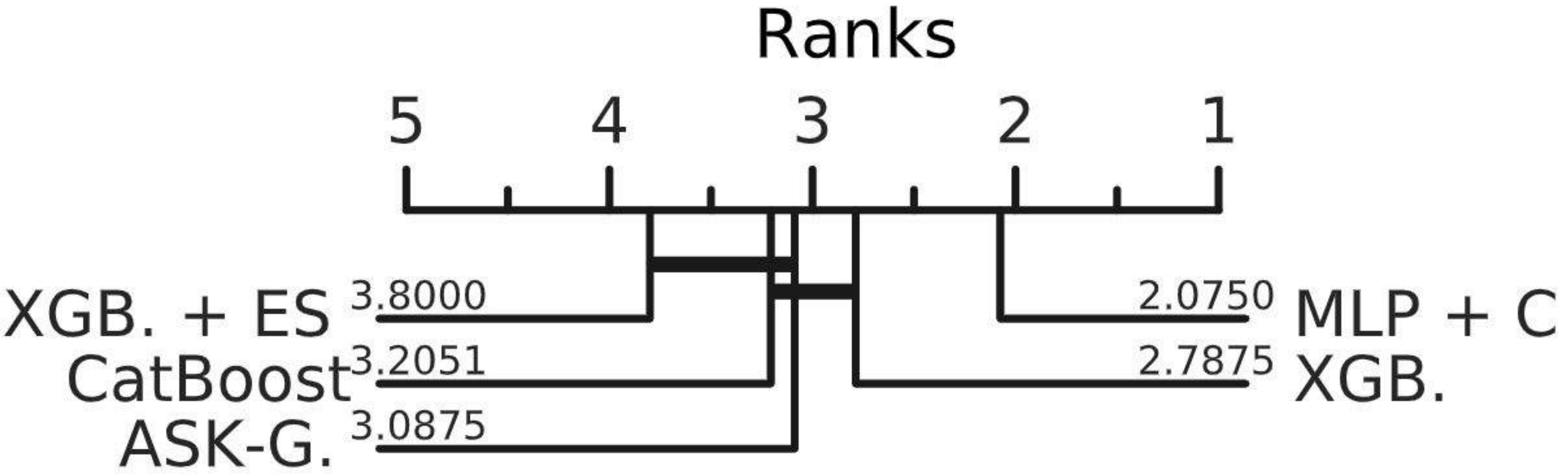


Successive Halving/Hyperband
[Jamieson & Talwaker 2015, Li et al. 2018]

Hypothesis 1: Regularization cocktails outperform state-of-the-art deep learning architectures on tabular datasets



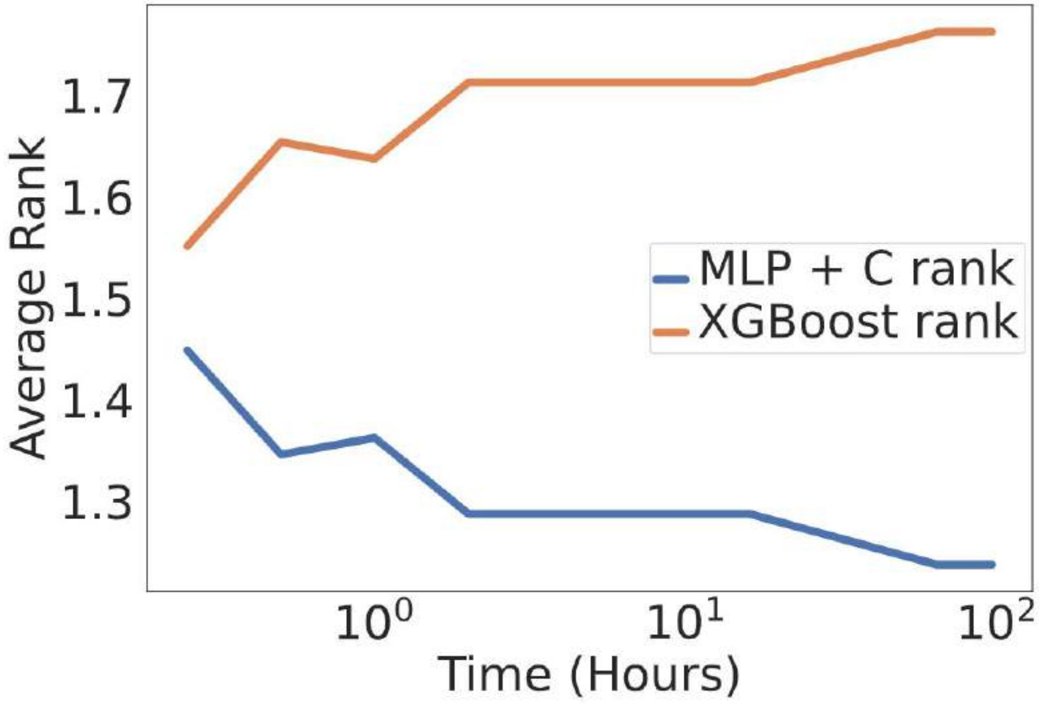
Hypothesis 2: Regularization cocktails outperform Gradient-Boosted Decision Trees (GBDTs)



Hypothesis 3:
Regularization cocktails are time-efficient and achieve strong anytime results.

Automated Machine Learning Hannover

AutoML.org



Limitations

1. **Only classification** and
   not regression, semi-supervised data or streaming data so far

2. We considered only somewhat **well-balanced datasets**

3. We used a fairly **simple DNN architecture and no general HPO**
   - simple multilayer perceptron (MLP)
   - fixed hyperparameters of the general training

4. There are **better AutoML frameworks** by now
   - e.g., we know that SMAC3 [Lindauer et al. 2021] performs often better than BOHB

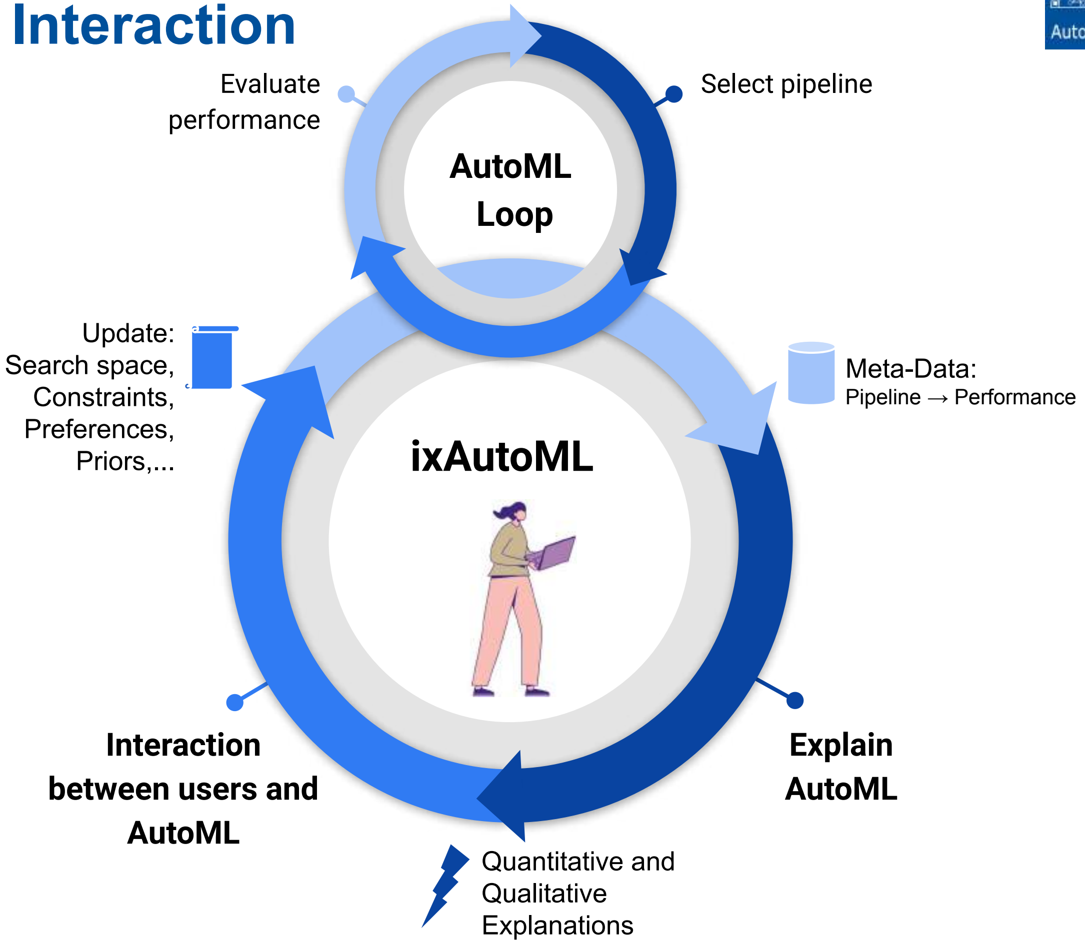
Humans and AutoML?

Towards Human-Centered AutoML

- Fully automated ML design can also receive pushback:
  - How to verify results (i.e., ML pipelines)?
  - How to bring in human expertise?
  - How to integrate into prototype-driven workflows?

- → Human-centered AutoML instead of fully automated ML

Explaining and Interaction



Can we explain what AutoML figured out?

[Moosbauer et al. NeurIPS'21]

Partial Dependence Plots [Friedman 2001]

Automated Machine Learning Hannover

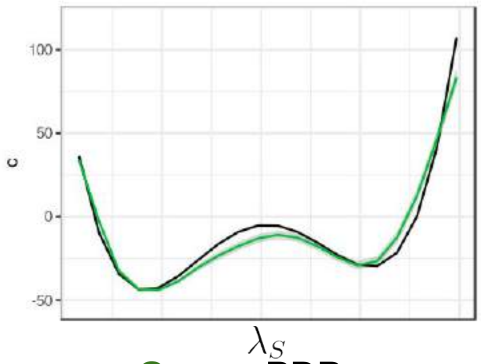
For, a subset S of the hyperparameters, the partial dependence function is:

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

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



Green: PDP
Black: Ground truth

→ Average of ICE curves.

Quantifying Uncertainties

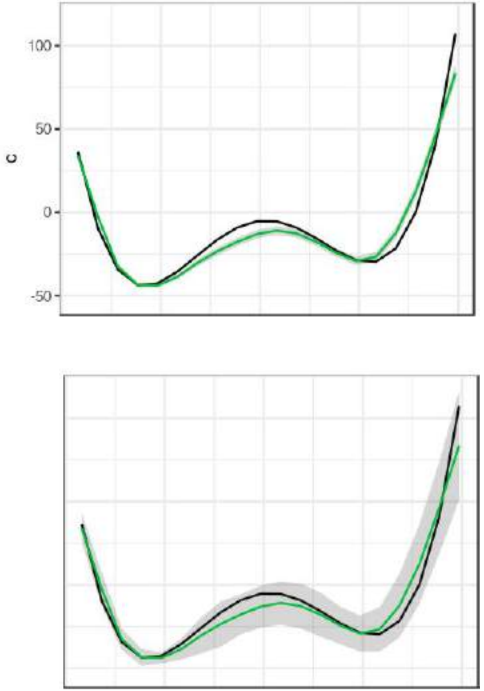
$$
\begin{aligned}
&\hat{s}_S^2(\lambda_S) \\
&= \mathbb{V}_{\hat{c}}\left[\hat{c}_S\left(\lambda_S\right)\right] \\
&= \mathbb{V}_{\hat{c}}\left[\frac{1}{n}\sum\nolimits_{i=1}^{n} \hat{c}\left(\lambda_S, \lambda_C^{(i)}\right)\right] \\
&= \frac{1}{n^2} 1^\top \hat{K}\left(\lambda_S\right)\ 1.
\end{aligned}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

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

Approximation:

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

→ Model-agnostic (local) approximation



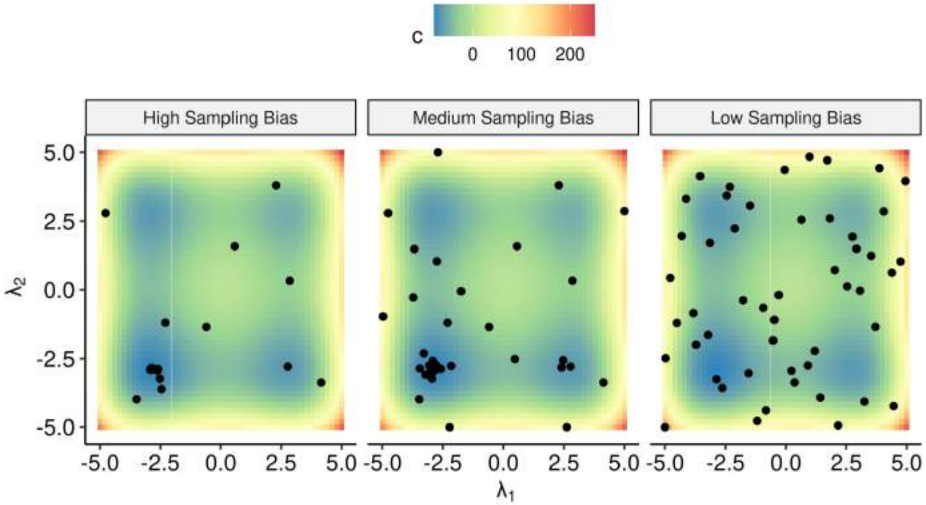
Ground truth
PDP
Uncertainty

Problem of Biased Sampling

- PDPs assume that the data is i.i.d.
- Obviously not the case for efficient AutoML tools with a focus on high-performance regions

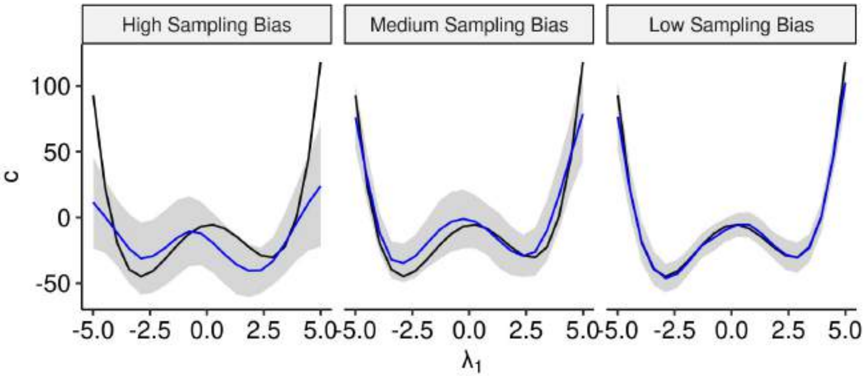
- Example:
  - BO with GPs and LCB
  - Different exploration rate for LCB to show different sampling bias

$$\mathrm{LCB}(\lambda) = \mu(\lambda) + \beta \cdot \sigma(\lambda)$$

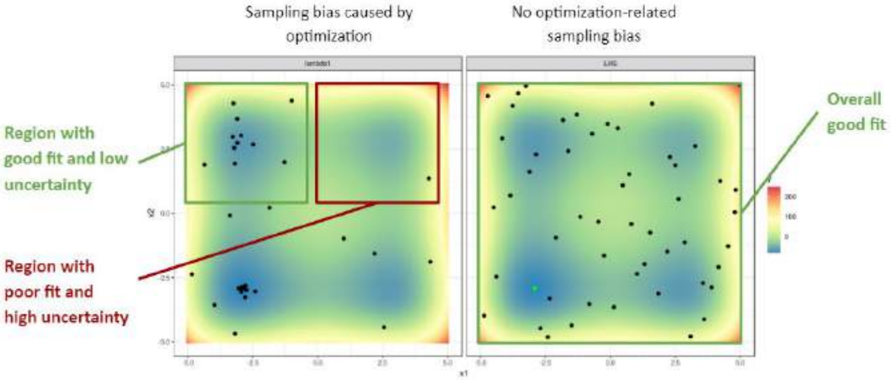


Impact of the Sampling Bias

- Simply using all observations from AutoML tools might lead to misleading PDPs
- Uncertainty estimates help to quantify the poor fits



→ of course, sampling bias is wanted and the solution cannot be to change the sampling behavior



Partitioning of Space

Partition space to obtain interpretable subspaces $\mathcal{N}'$.

Uncertainty variation across all ICE estimates:

$$L\left(\lambda_S, \mathcal{N}'\right) = \sum\nolimits_{i \in \mathcal{N}} \left(\hat{s}^2\left(\lambda_S, \lambda_C^{(i)}\right) - \hat{s}^2_{S|\mathcal{N}'}\left(\lambda_S\right)\right)^2$$

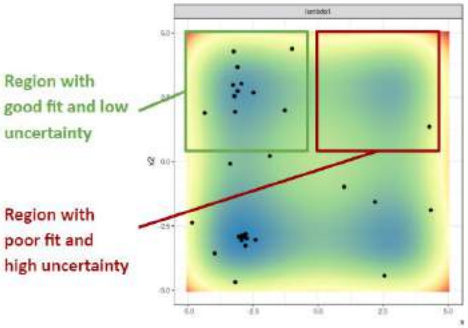
$$\hat{s}^2_{S|\mathcal{N}'}\left(\lambda_S\right) := \frac{1}{|\mathcal{N}'|} \sum_{i \in \mathcal{N}'} \hat{s}^2\left(\lambda_S, \lambda_C^{(i)}\right)$$

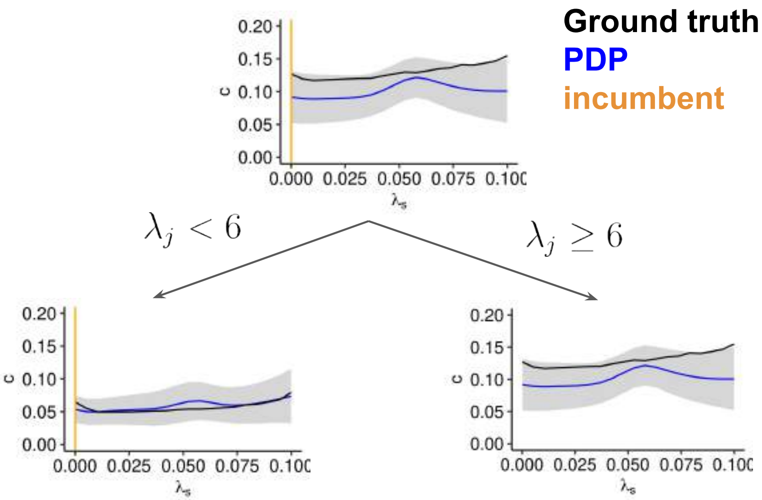
→ **Uncertainty structure of ICE curves should maximally agree**

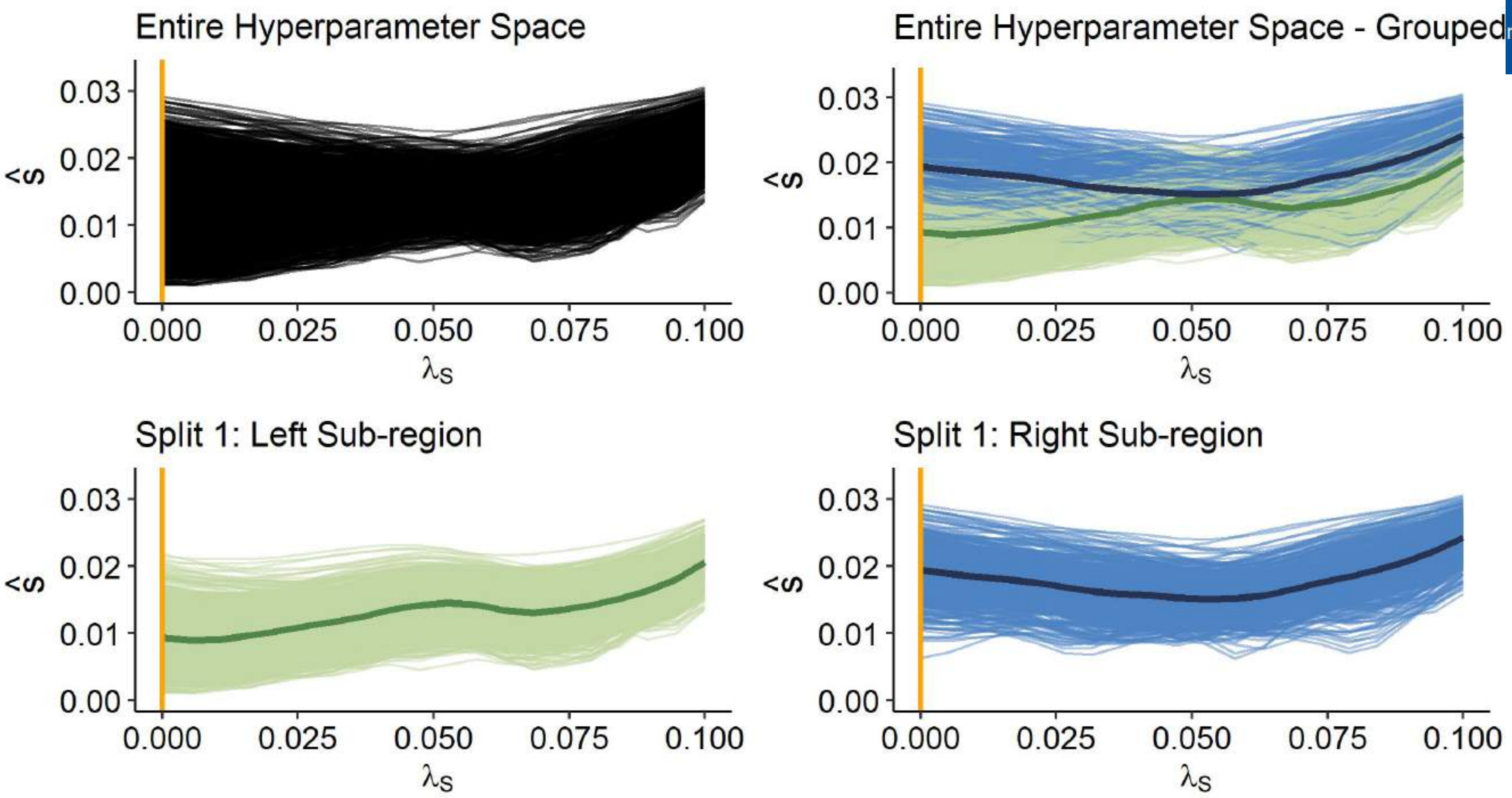
Split Loss = Aggregation over all grid points:

$$\mathcal{R}_{L2}(\mathcal{N}') = \sum\nolimits_{g=1}^{G} L(\lambda_S^{(g)}, \mathcal{N}')$$

Note (i): Partition only along the marginalized dimensions



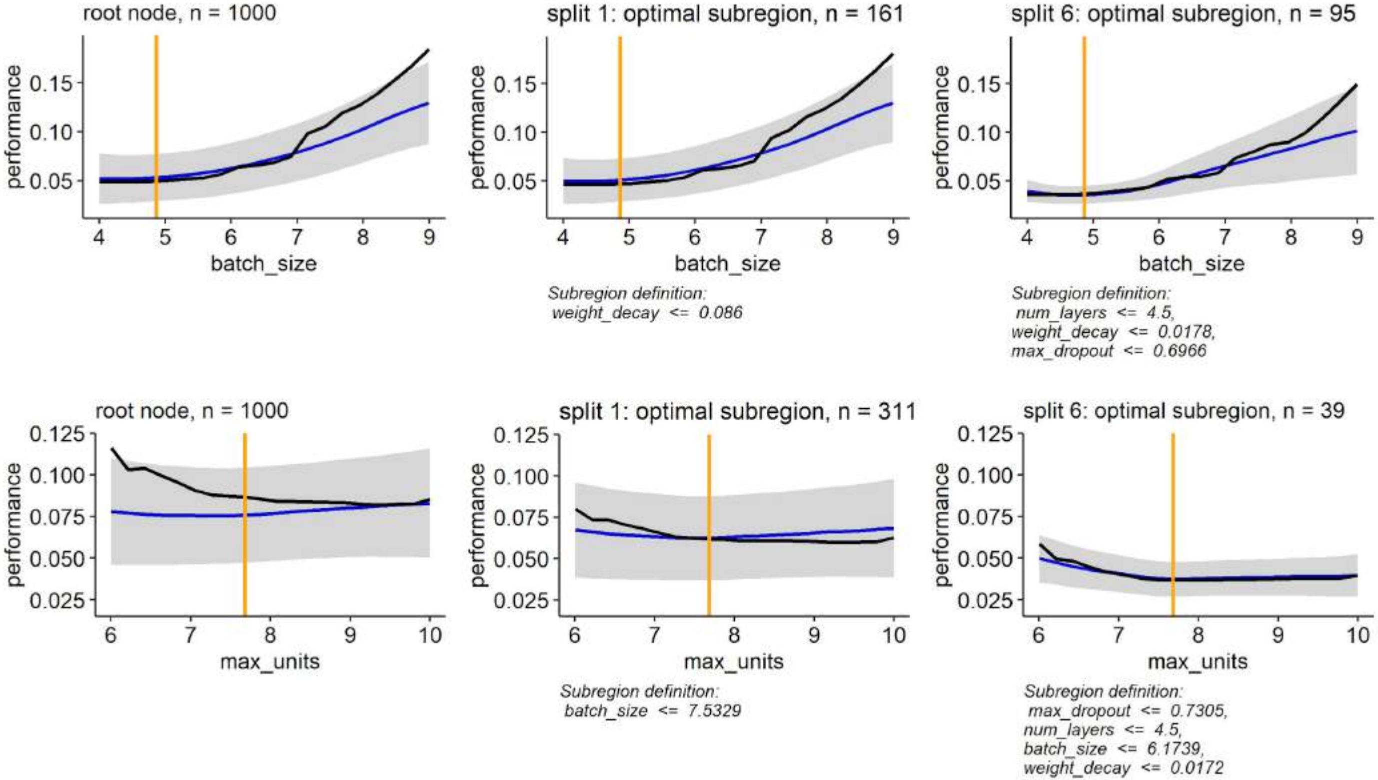



Entire Hyperparameter Space
Entire Hyperparameter Space - Grouped
Split 1: Left Sub-region
Split 1: Right Sub-region
0.03
0.02
0.01
0.00
0.000
0.025
0.050
0.075
0.100
λ_S
$\hat{s}$

Explaining AutoML via PDPs

Automated Machine Learning Hannover

Ground truth
PDP
incumbent



Can Developers also Bring in their Expertise?

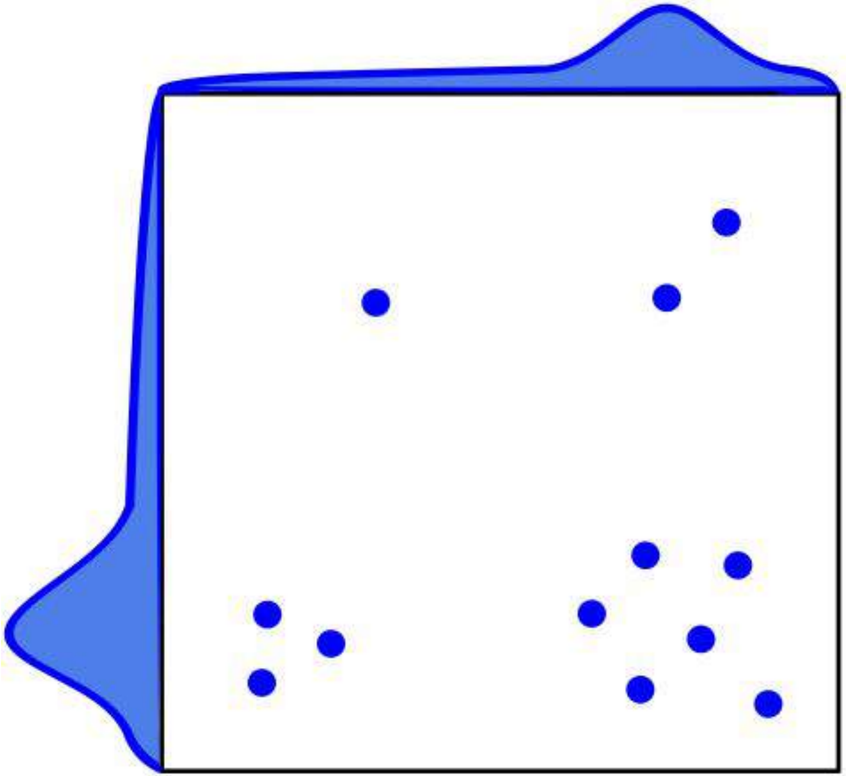
[Souza et al. ECML'21]



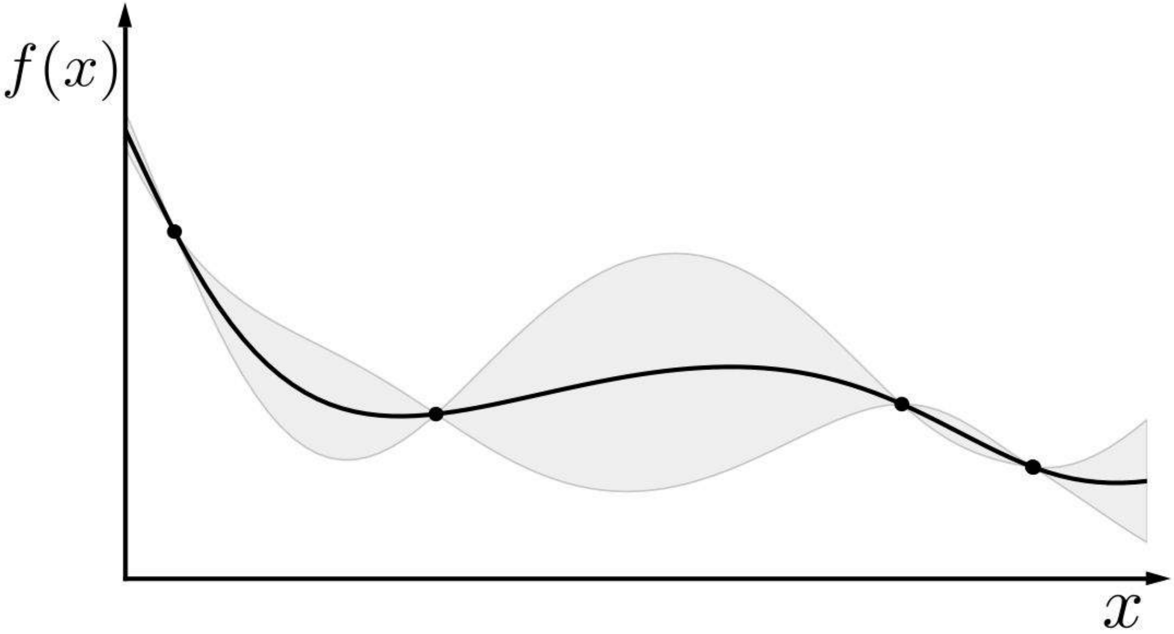

Integrating Human-Prior Knowledge

- ML practitioners often have an intuition for promising hyperparameter configurations
- → Sampling of configurations should focus in these regions

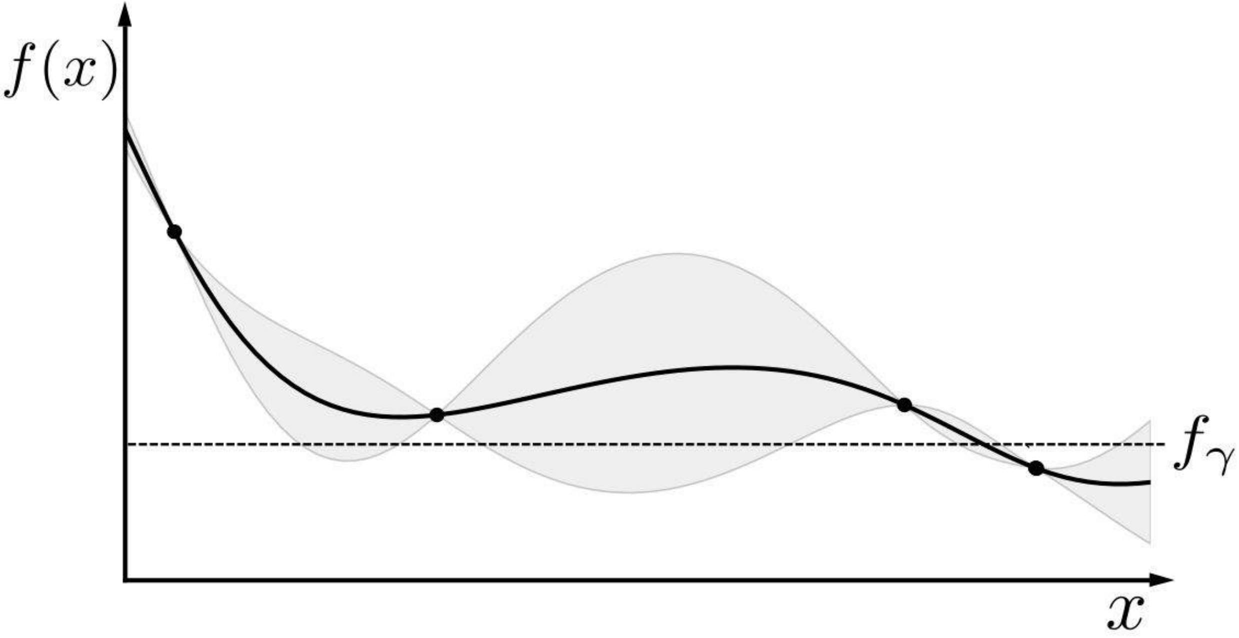
- However, practitioners can also be wrong with their intuition
- → Over time, we should trust the evaluated configurations and the surrogate more than the human expert



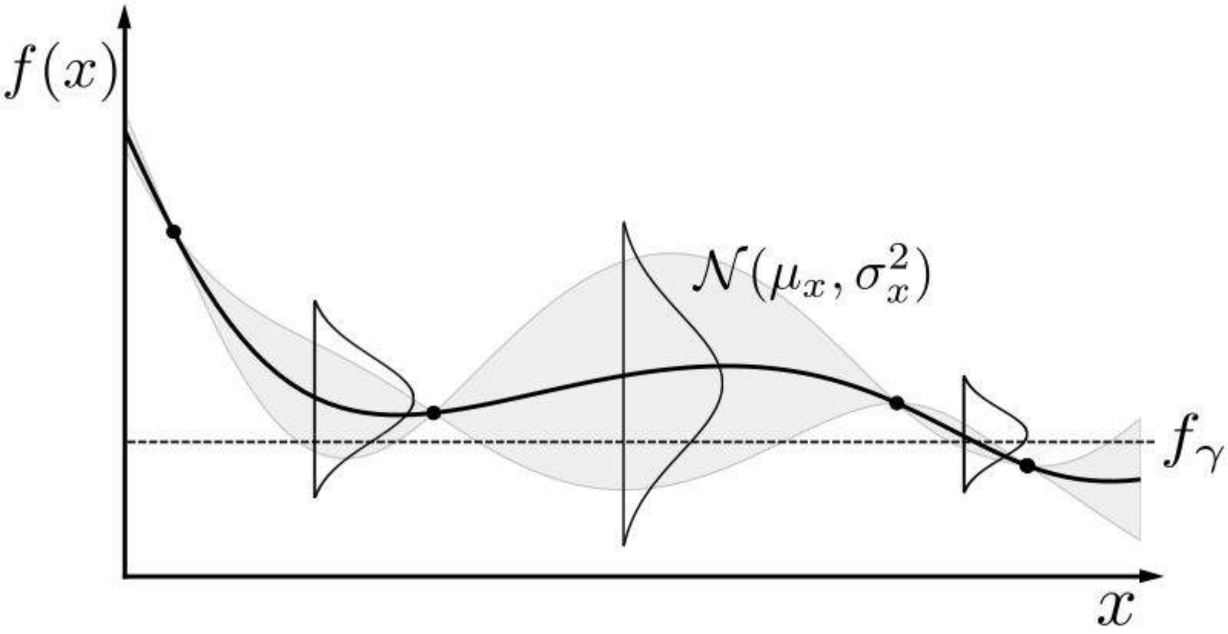
Human-Prior Knowledge -- BOPro



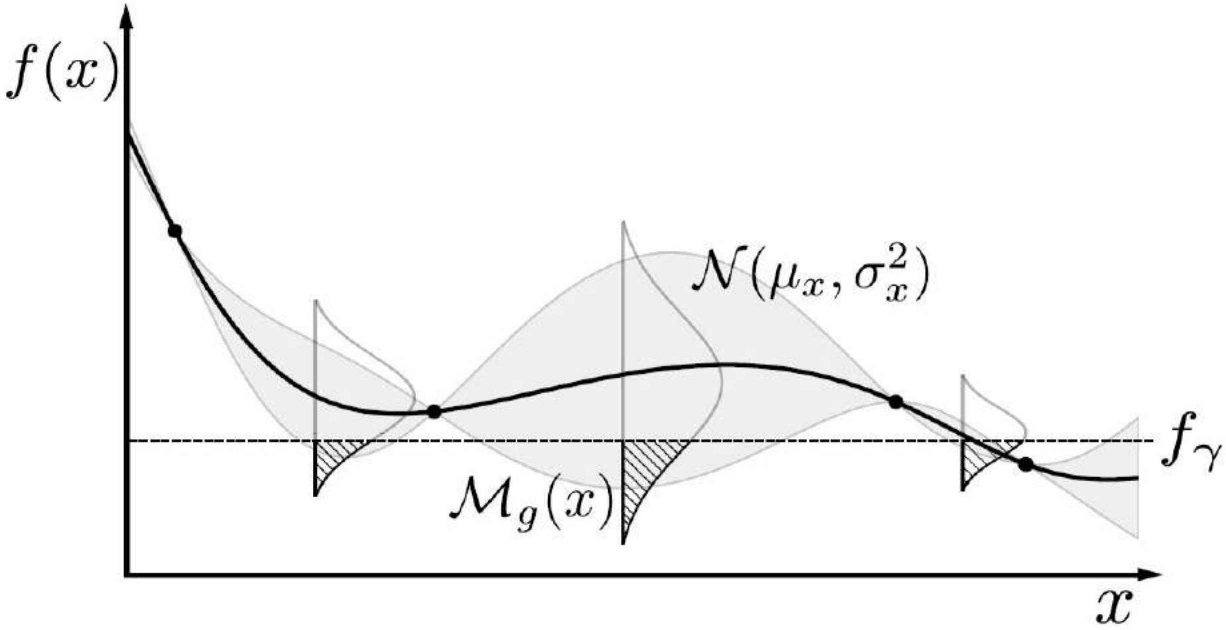
Human-Prior Knowledge -- BOPro



Human-Prior Knowledge -- BOPro



Human-Prior Knowledge -- BOPro



- Instead of modelling f(x), we model whether a configuration is "good" or "bad"
- Using the Gaussian distribution of a GP, we can determine the probability of being good

$$g(\boldsymbol{x}) \propto P_g(\boldsymbol{x}) \mathcal{M}_g(\boldsymbol{x})^{\frac{t}{\beta}}$$

Human-Prior Knowledge -- πBO [Hvarfner et al. 2021]

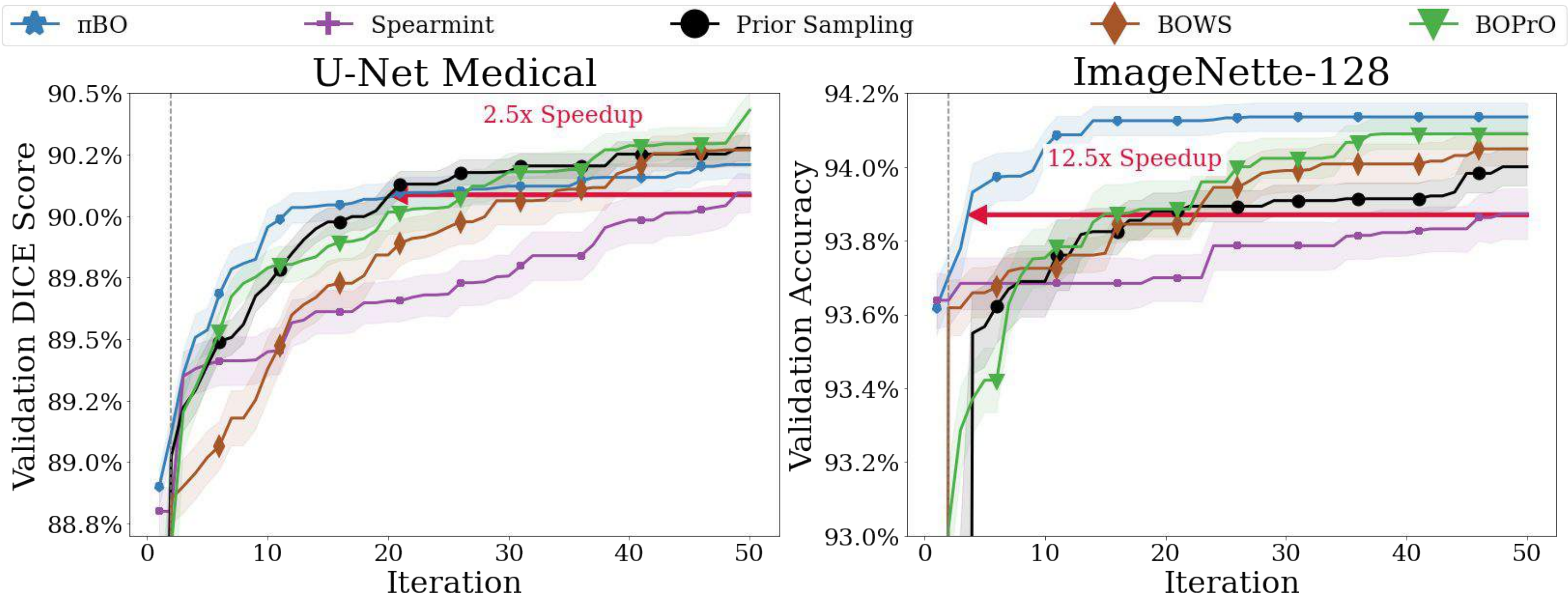
- BOPro has several assumptions on how to model the observations
- πBO is simpler: augment the the acquisition function of BO by a human prior preference

$$\alpha(x) = \hat{\mu}(x) + \kappa\hat{\sigma}(x)$$

$$\alpha_{\pi}(x) = \alpha(x)\pi(x)^{\frac{\beta}{t}}$$

- Advantages of πBO:
  - Can be combined with any acquisition function
  - Same convergence guarantees as with the original acquisition functions (e.g., EI)
  - Can again recover from misleading a-priori knowledge

Human-Prior Knowledge -- πBO [Hvarfner et al. 2021]

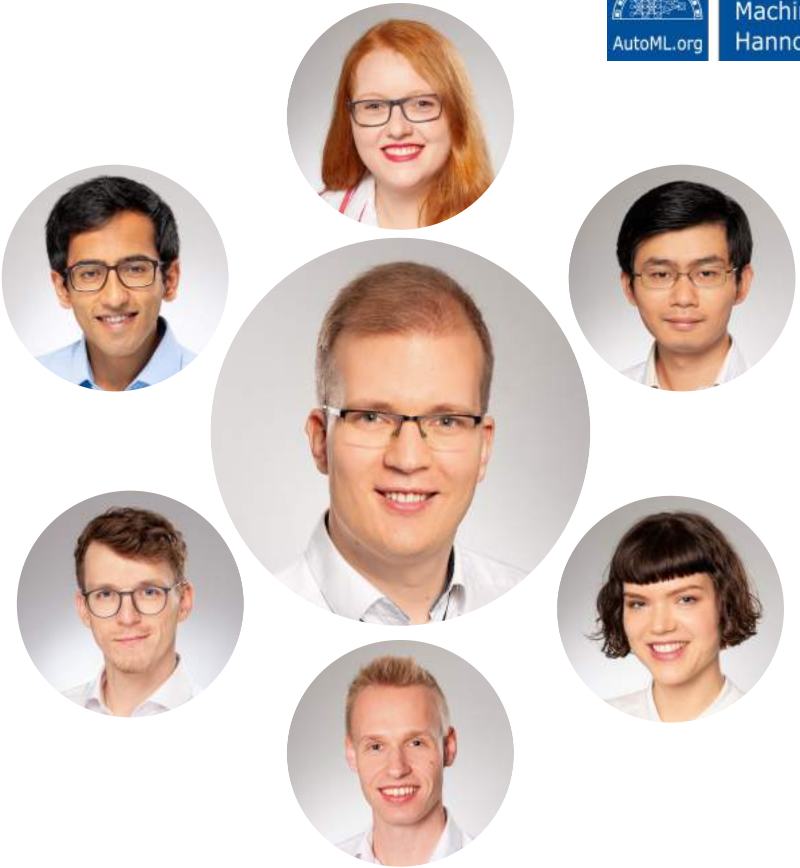


/AutoML_org/

/automl/

https://tinyurl.com/automlyt

Thank you!



Appendix

Cocktail Frequencies