Multiobjective optimization

Metaheuristics Summer School.

David Salinas, Freiburg University. July 2024.

Introduction

Introduction

- Optimization is often about optimising a single objective but what if we have many?

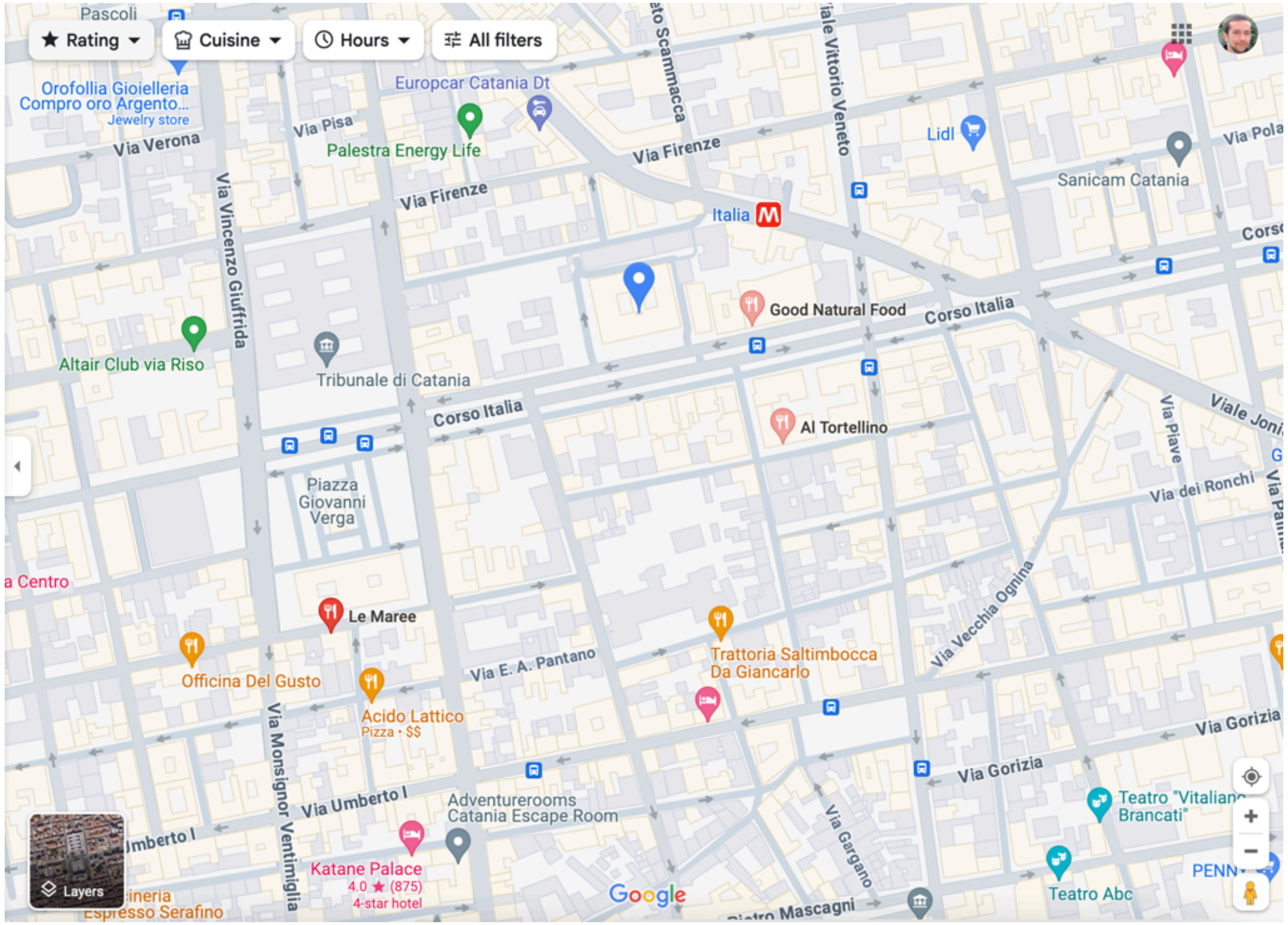
Introduction

- Optimization is often about optimising a single objective but what if we have many?

- Assume you want to pick a restaurant close to the Summer school…

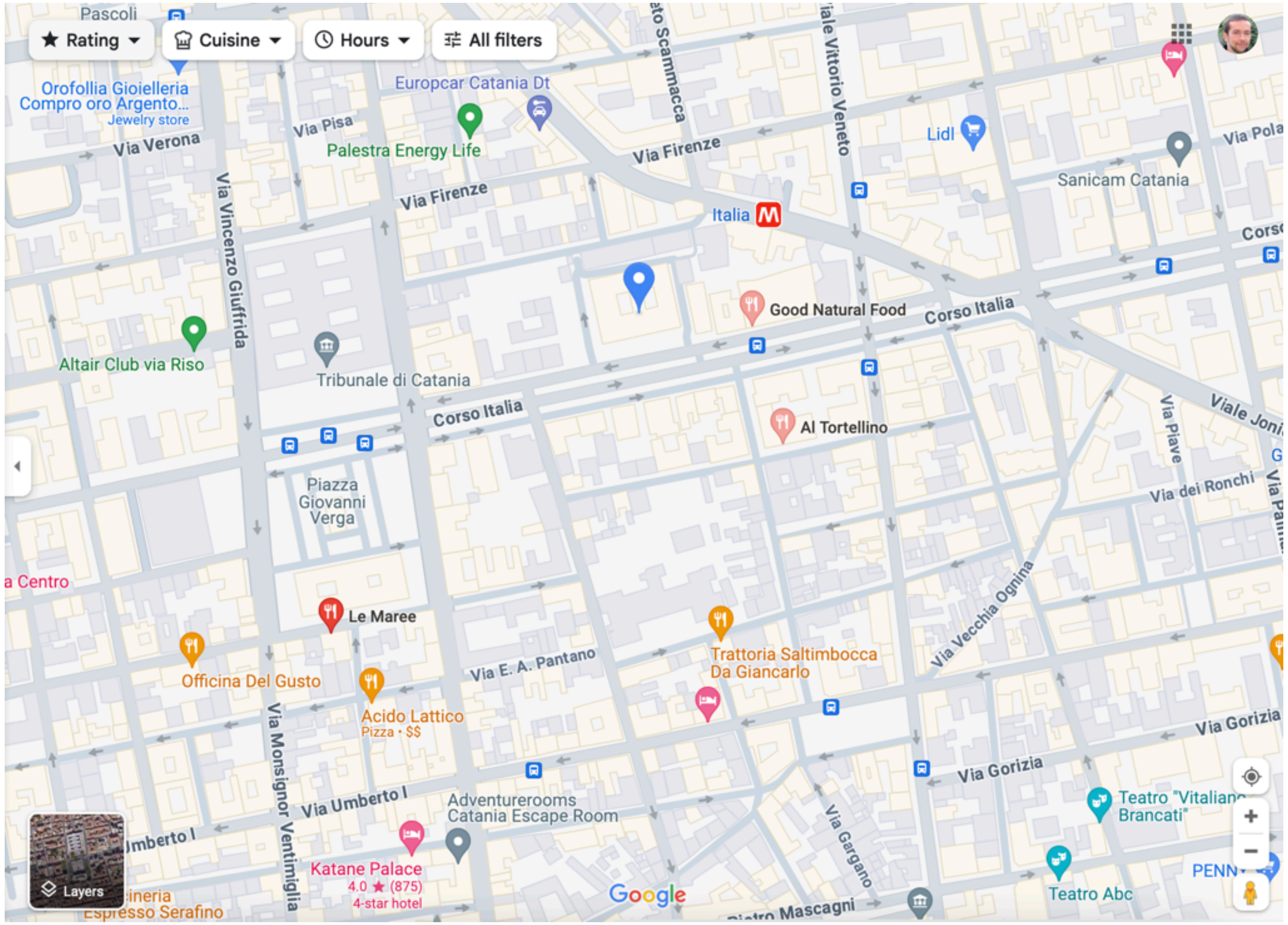
Introduction

- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…



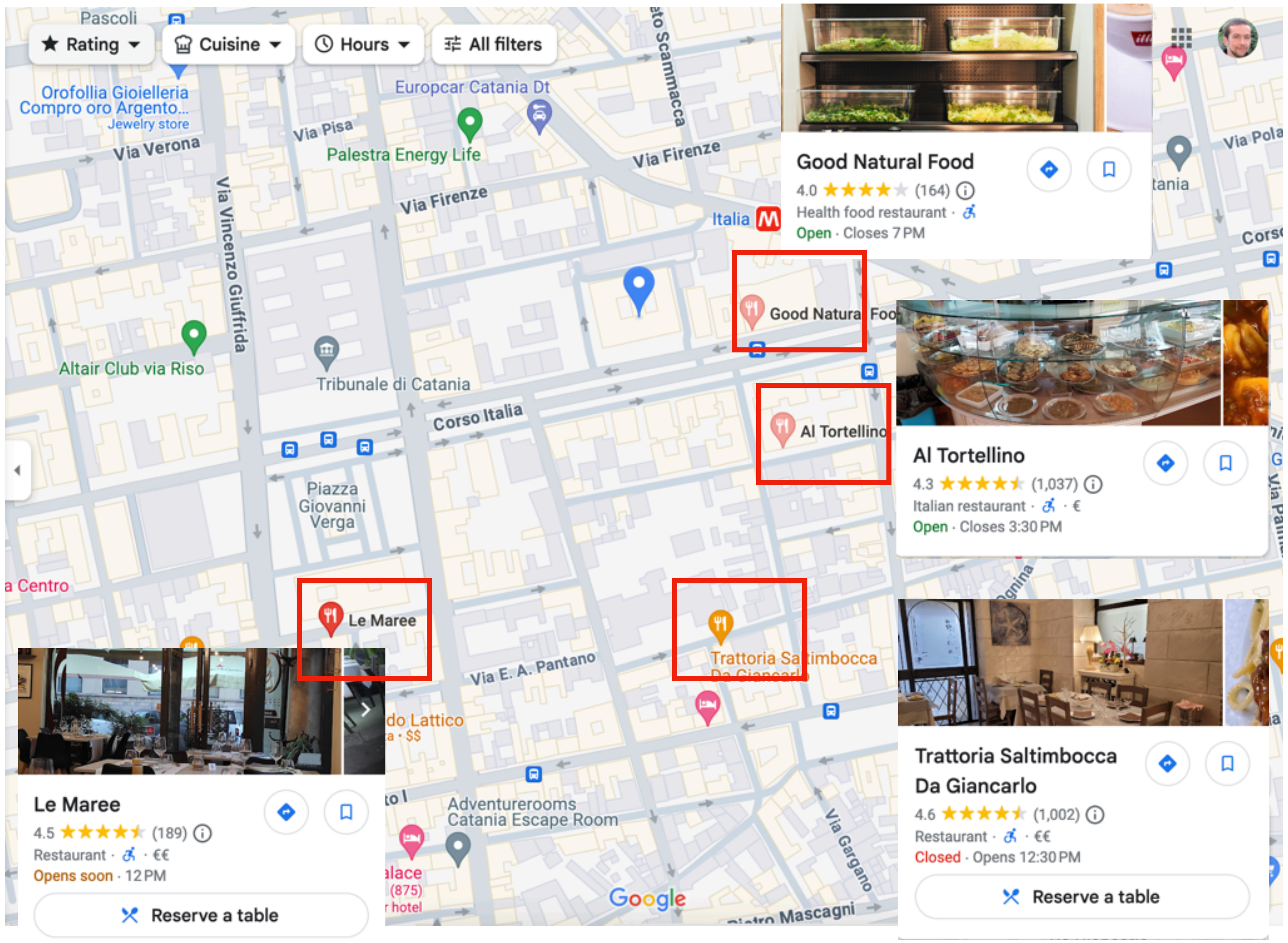
Introduction

- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…
- And you only care about rating **and** distance



Introduction

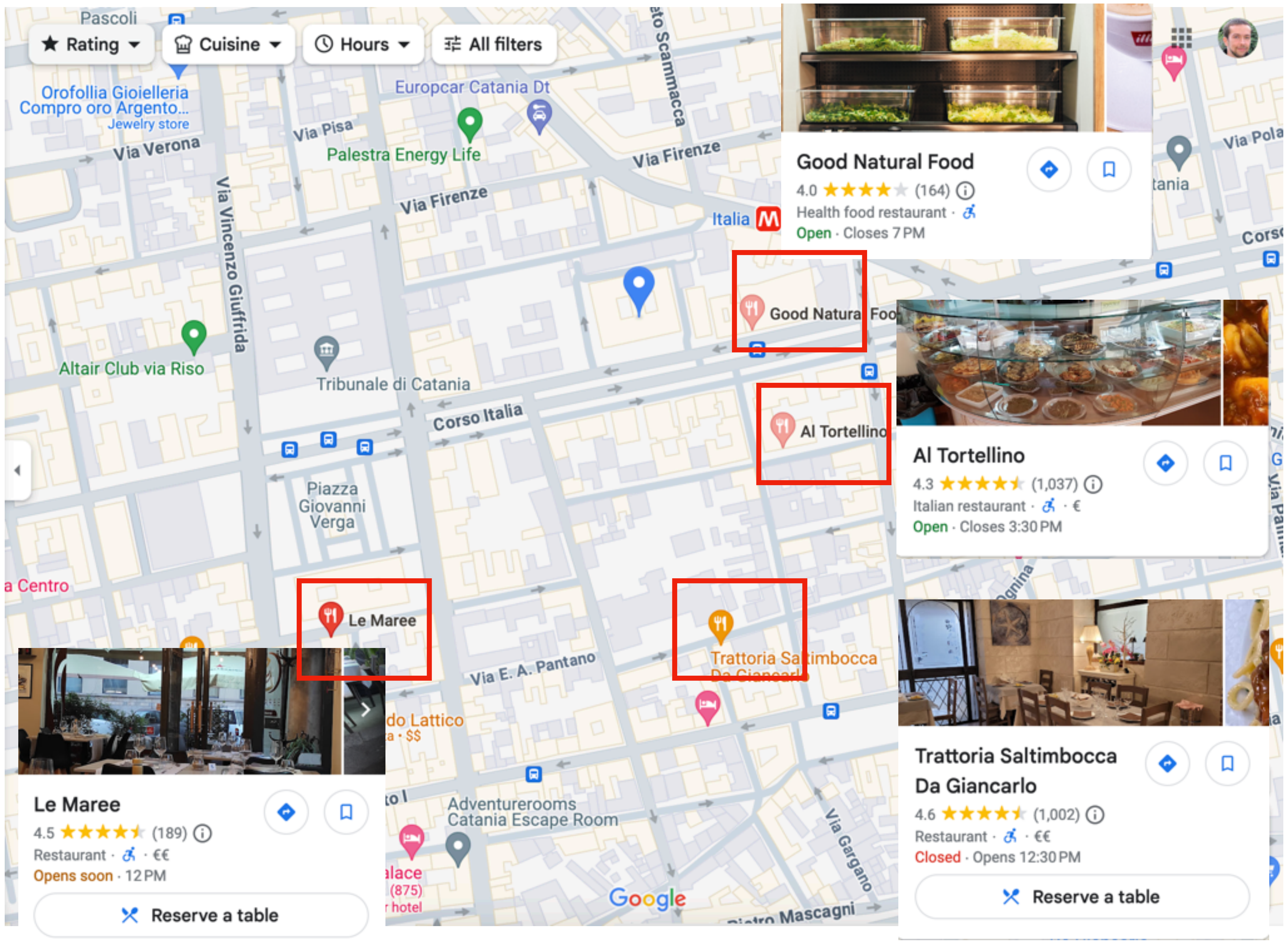
- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…
- And you only care about rating **and** distance



Introduction

- Optimization is often about optimising a single objective but what if we have many?
- Assume you want to pick a restaurant close to the Summer school…
- And you only care about rating **and** distance

🤔 How to select which restaurant to go?



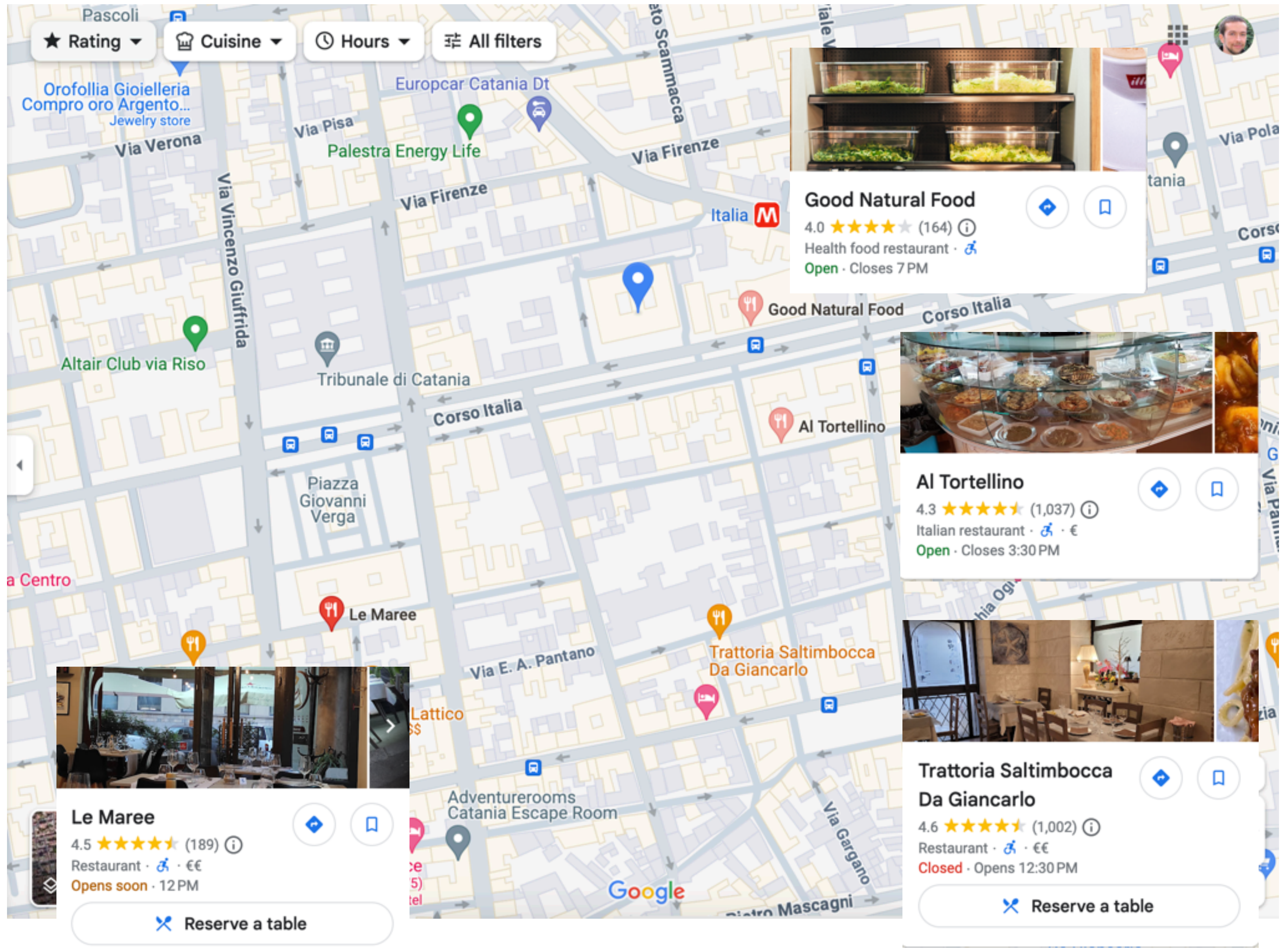
How to select which restaurant to go?

How to select which restaurant to go?

- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives

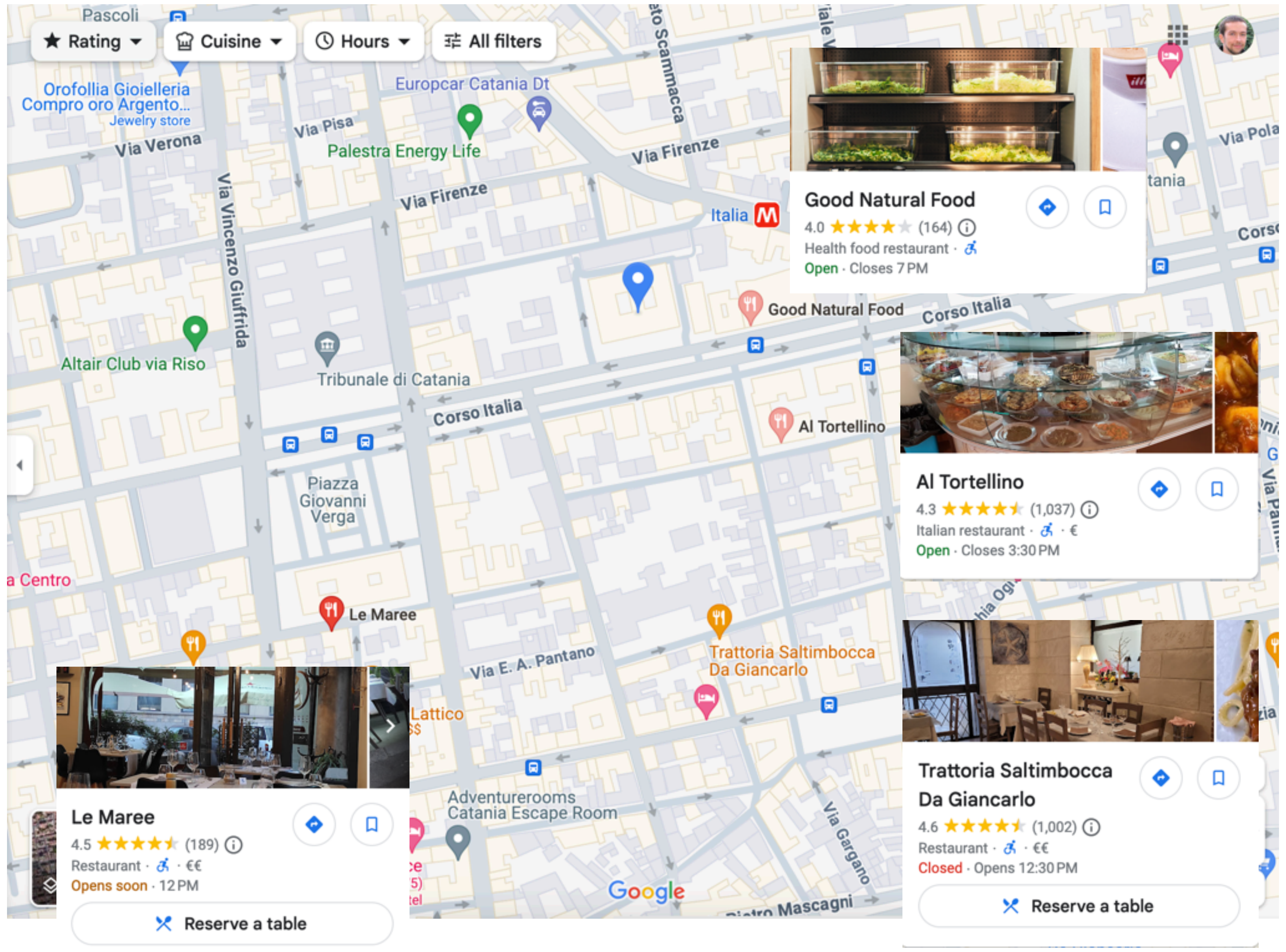
How to select which restaurant to go?

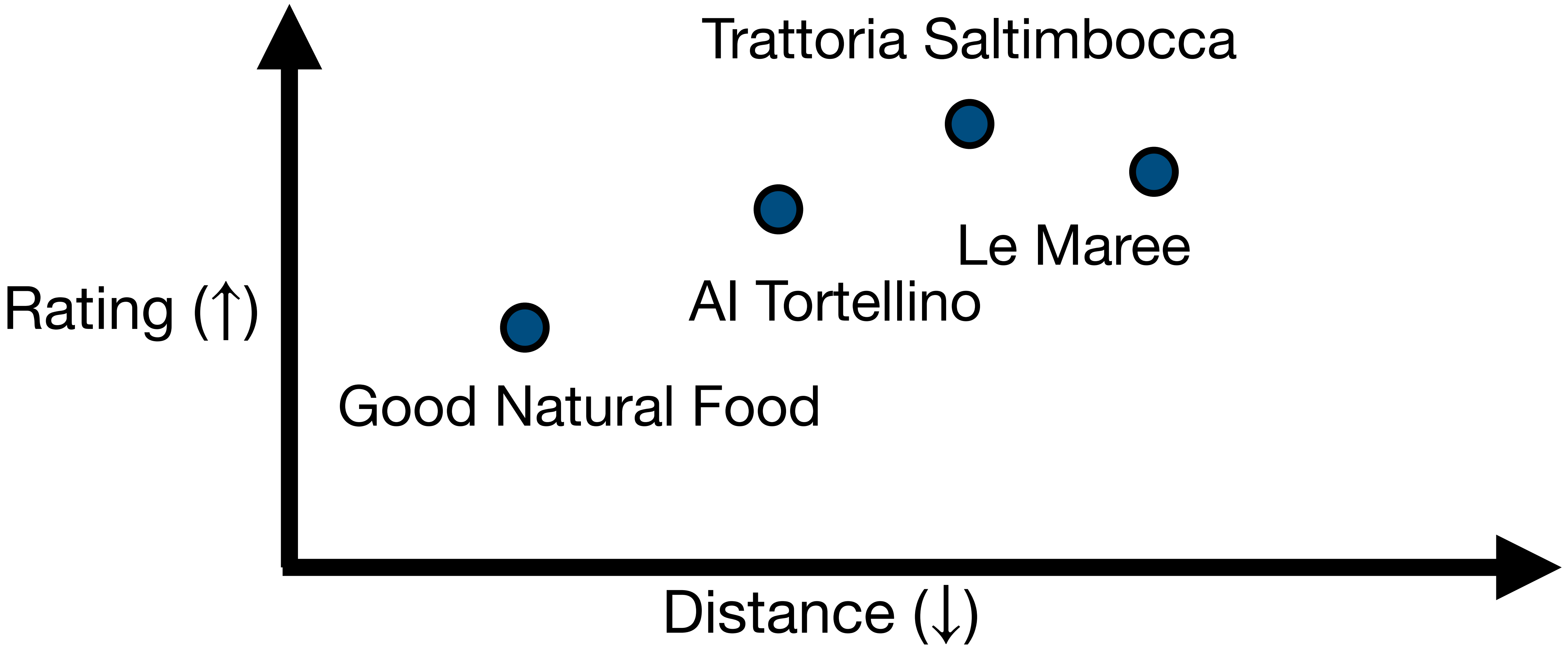
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives



How to select which restaurant to go?

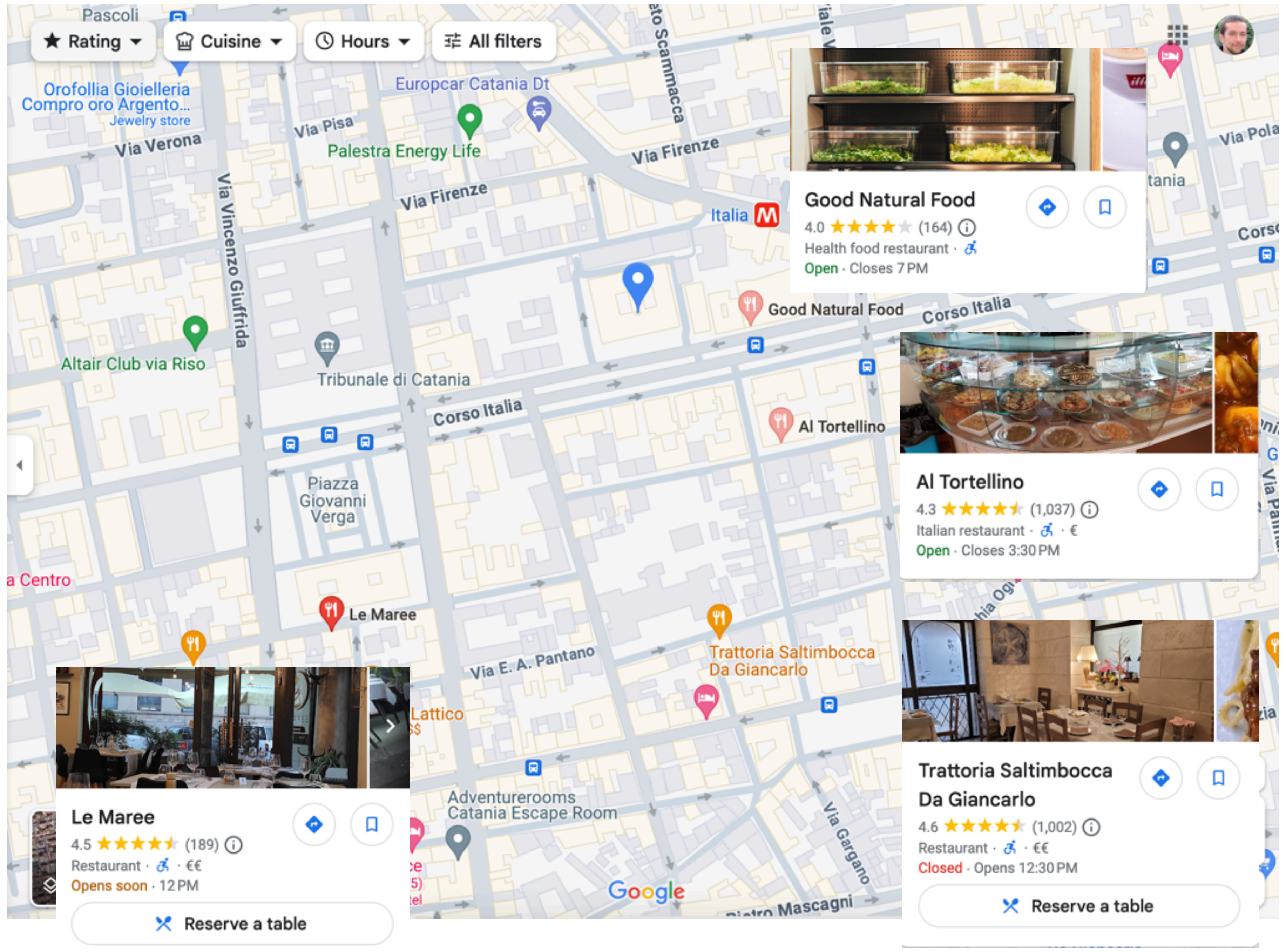
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives

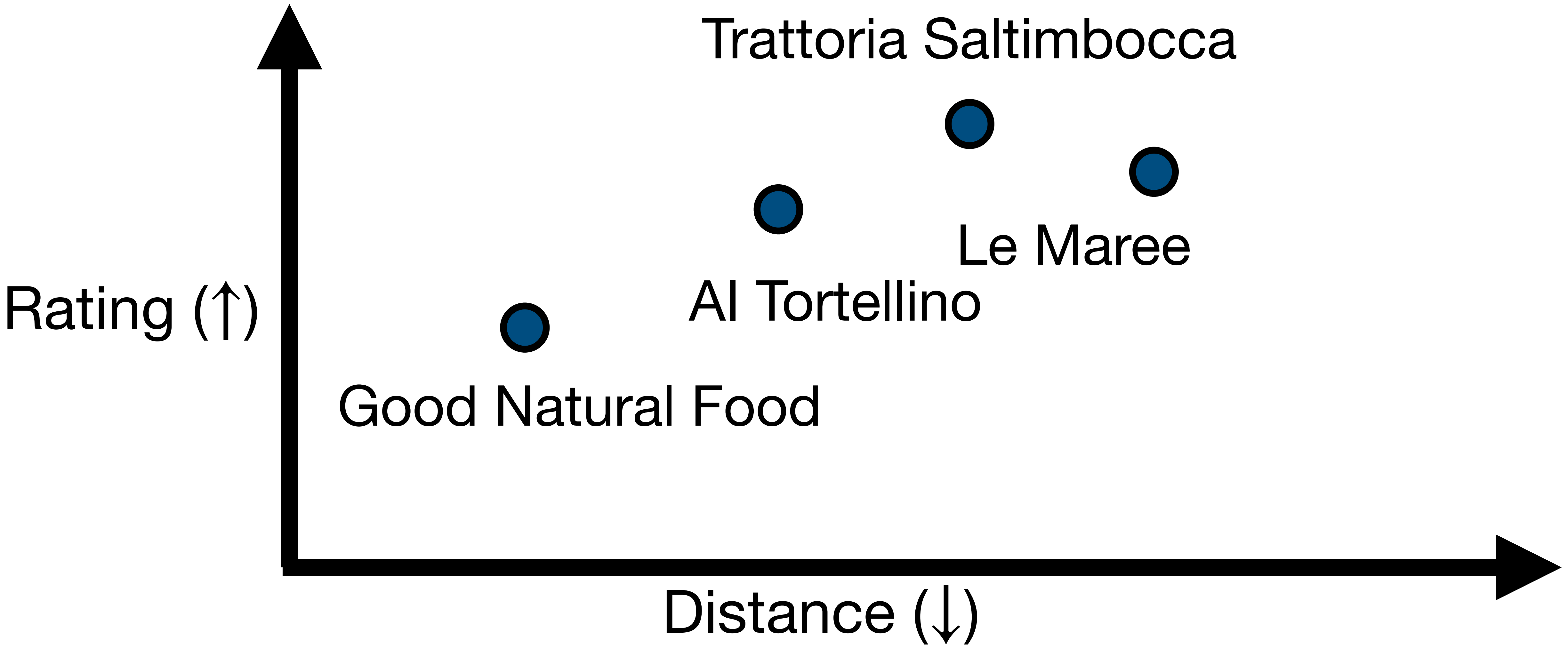




How to select which restaurant to go?

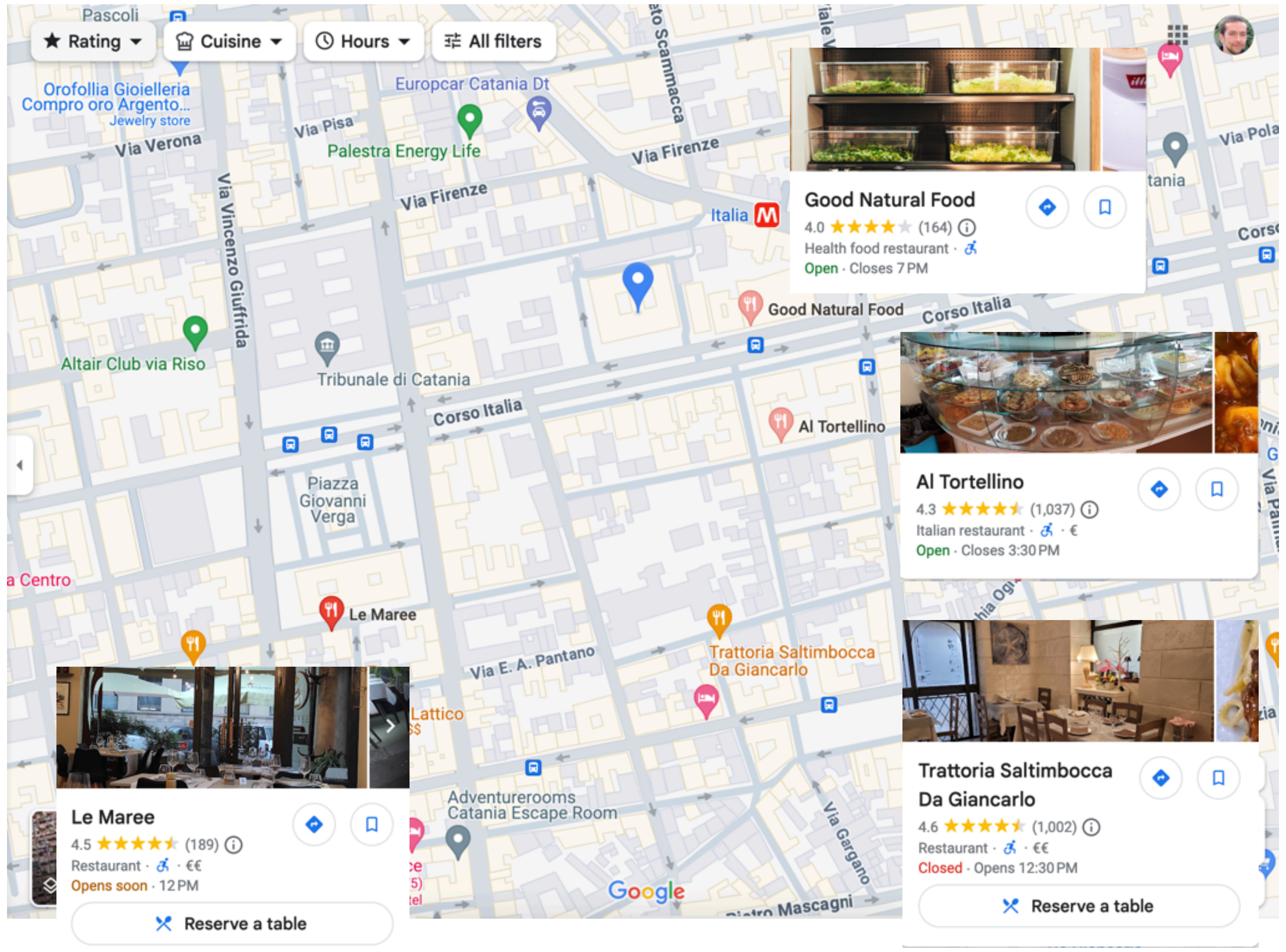
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!

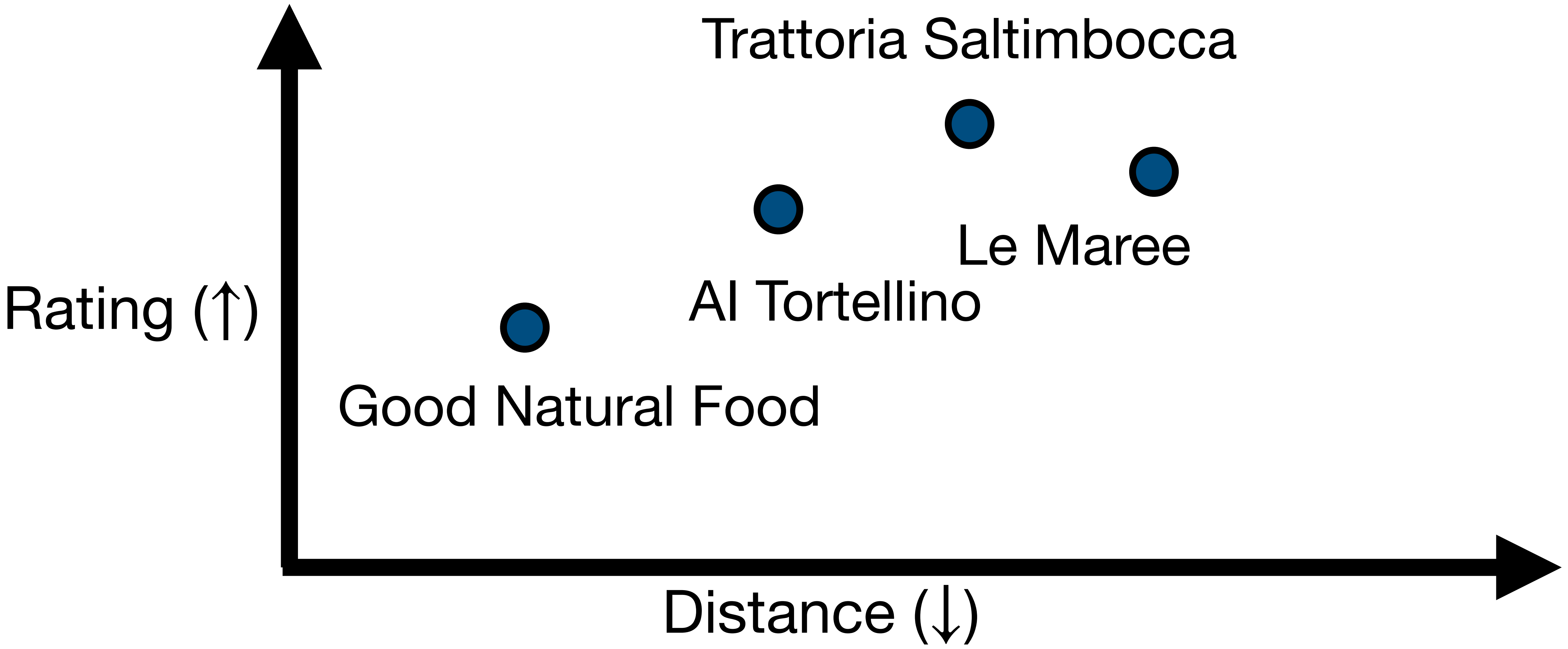




How to select which restaurant to go?

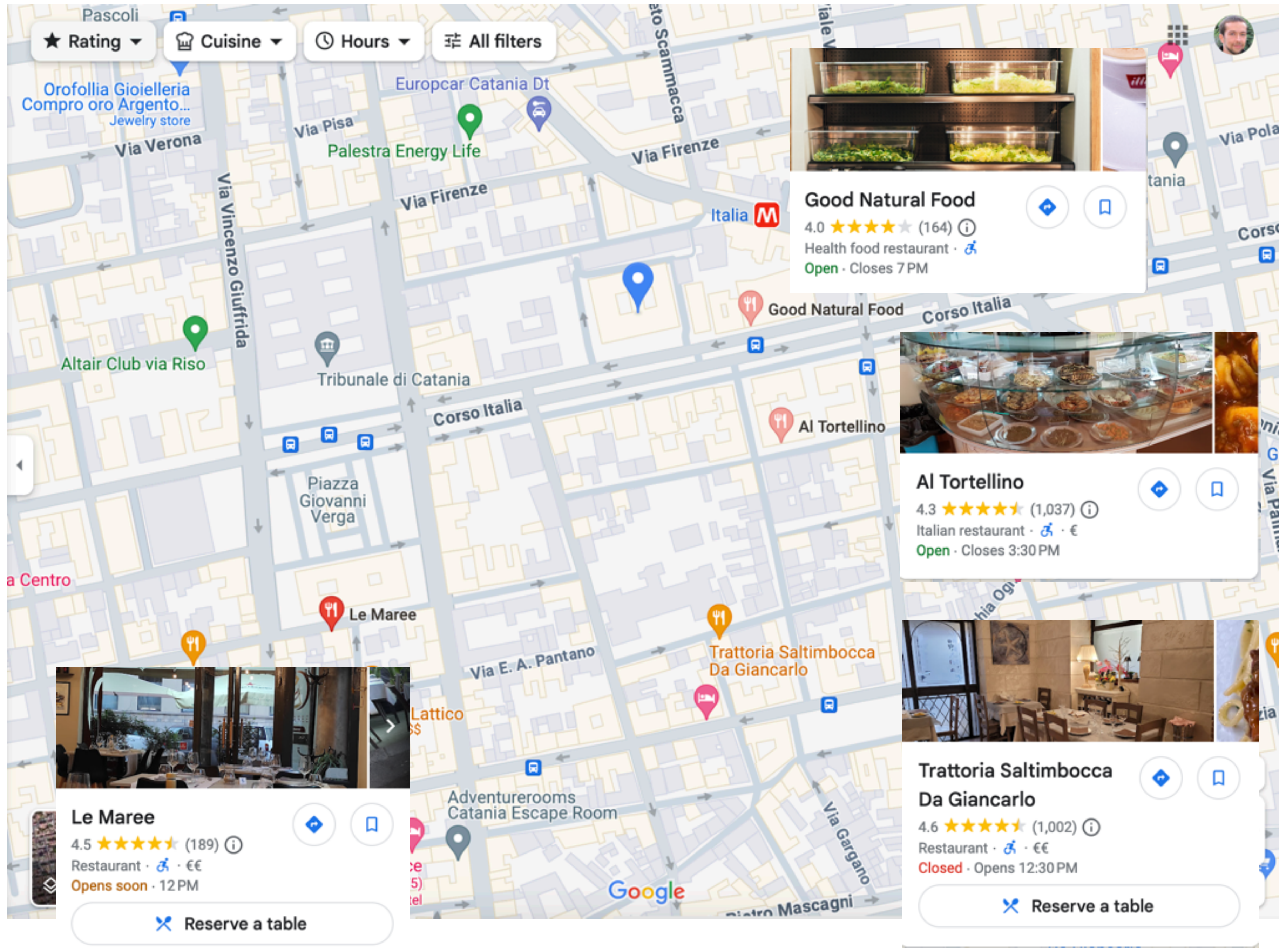
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?

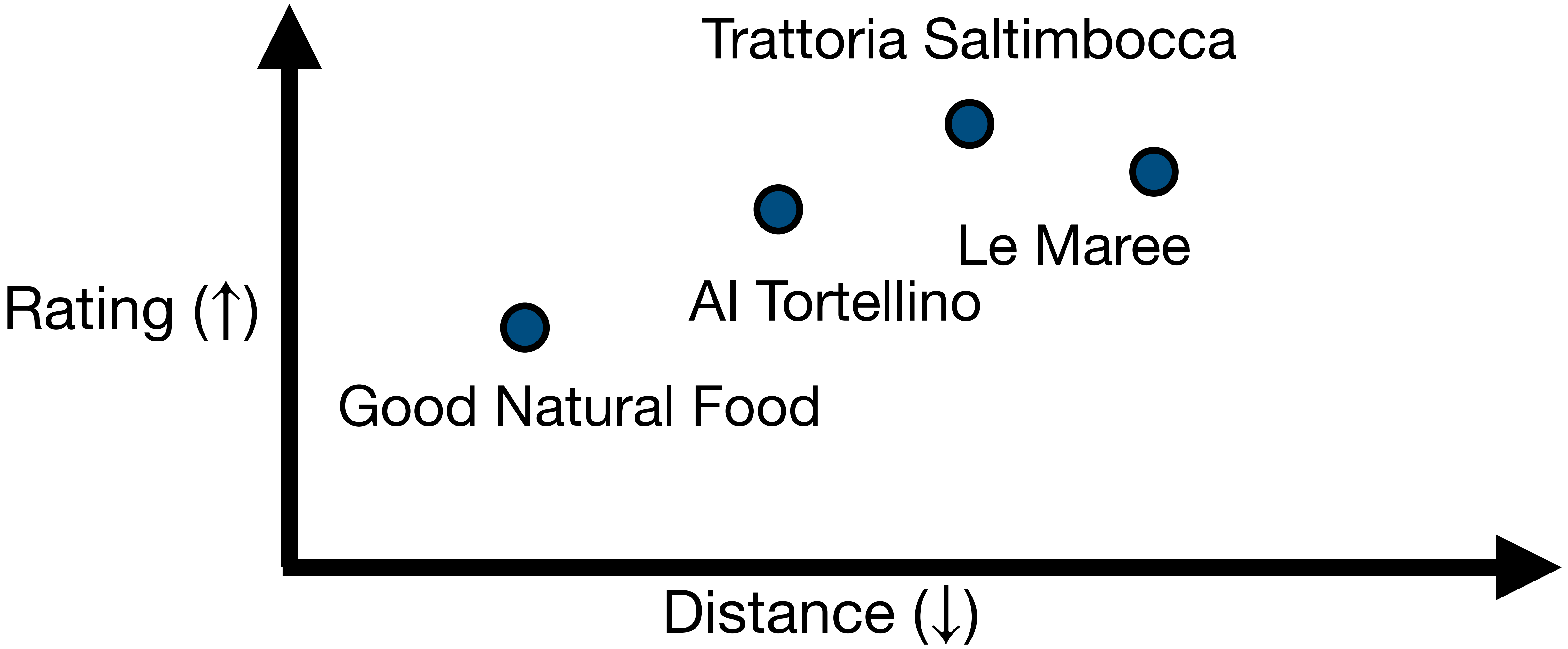




How to select which restaurant to go?

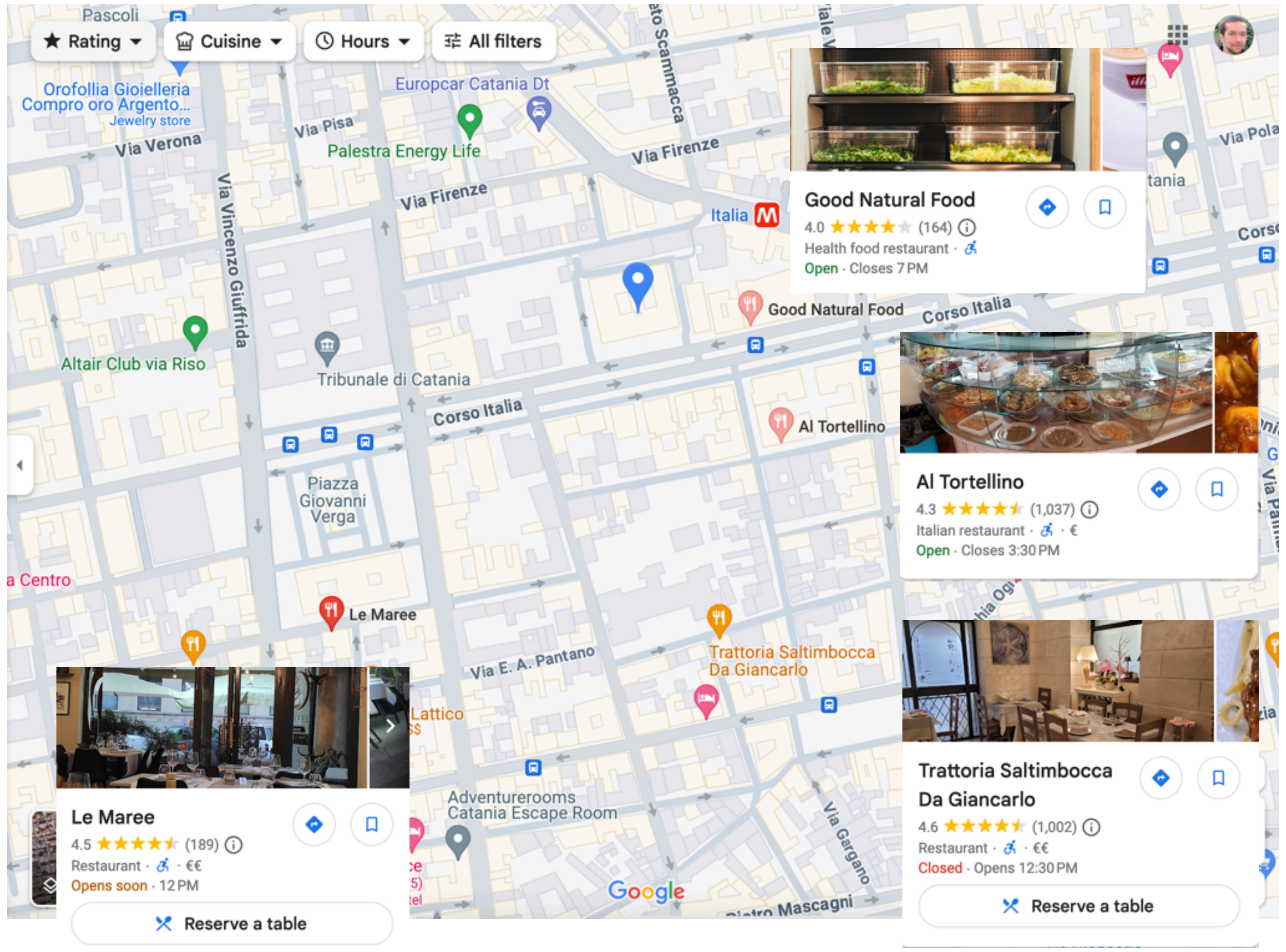
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal

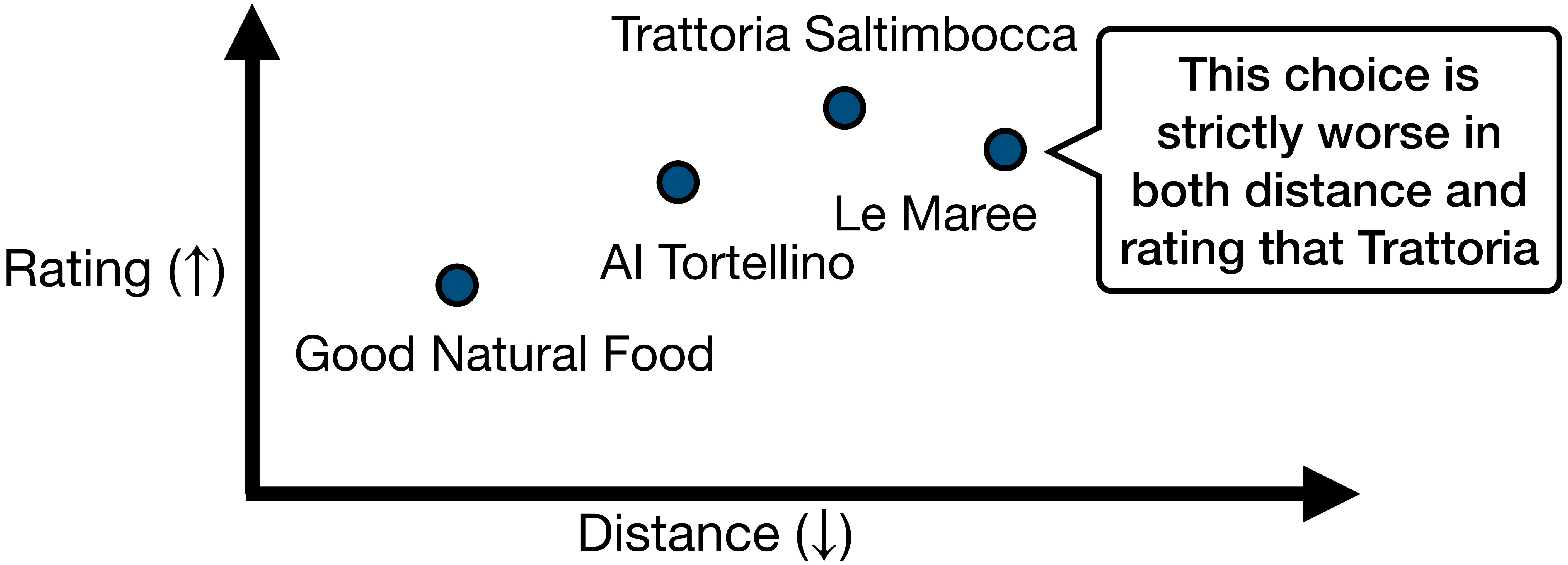




How to select which restaurant to go?

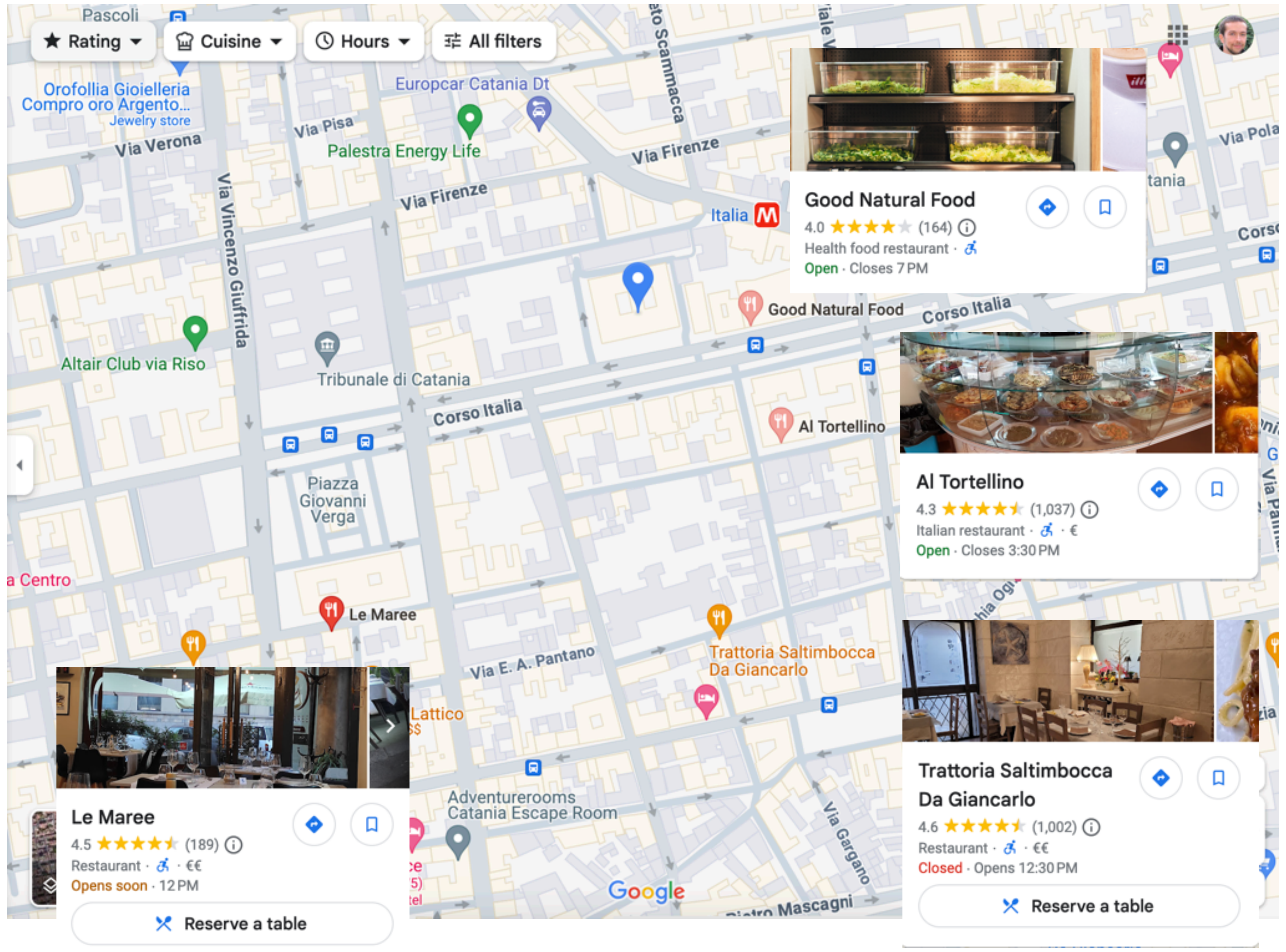
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal

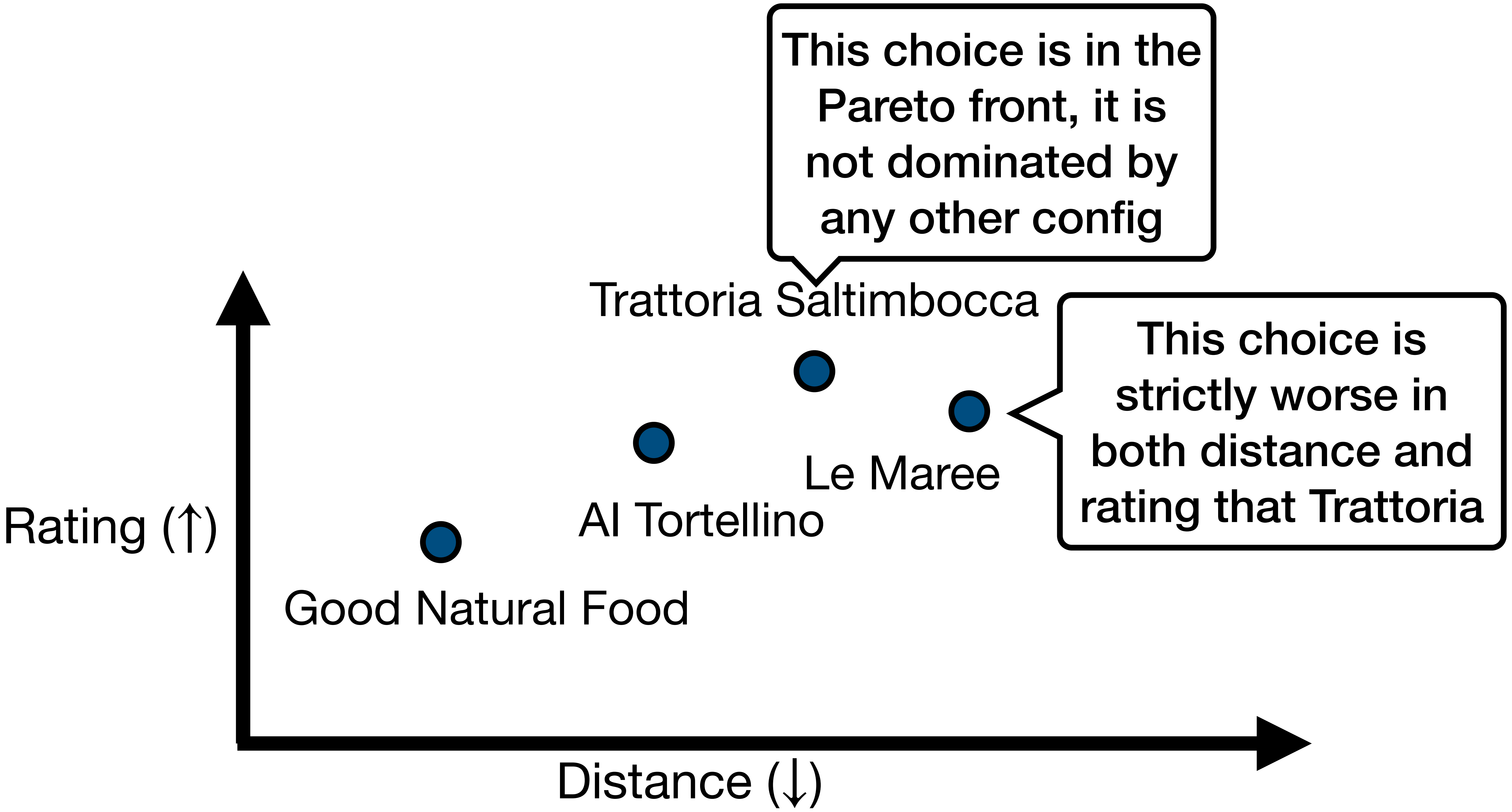




How to select which restaurant to go?

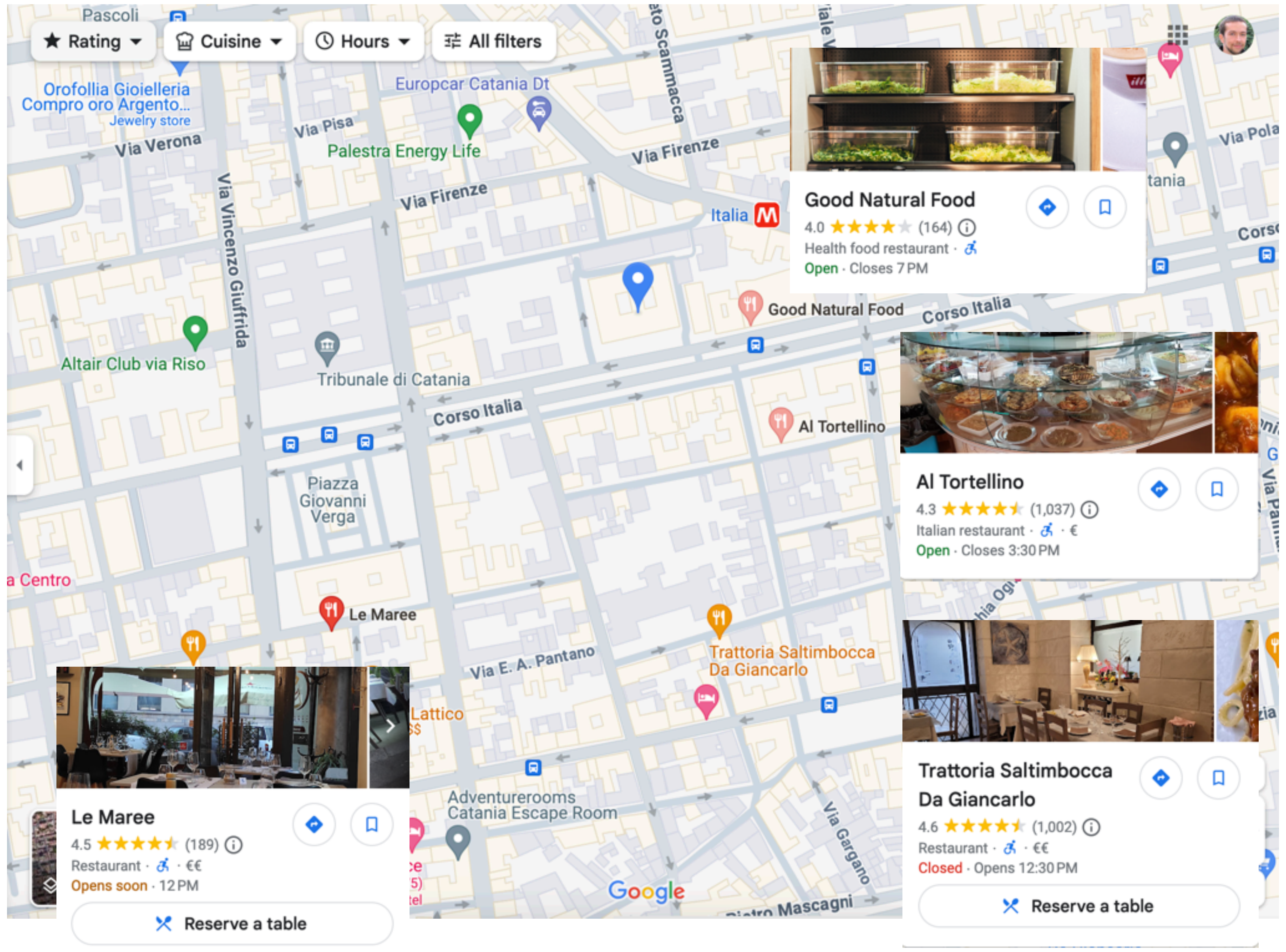
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal

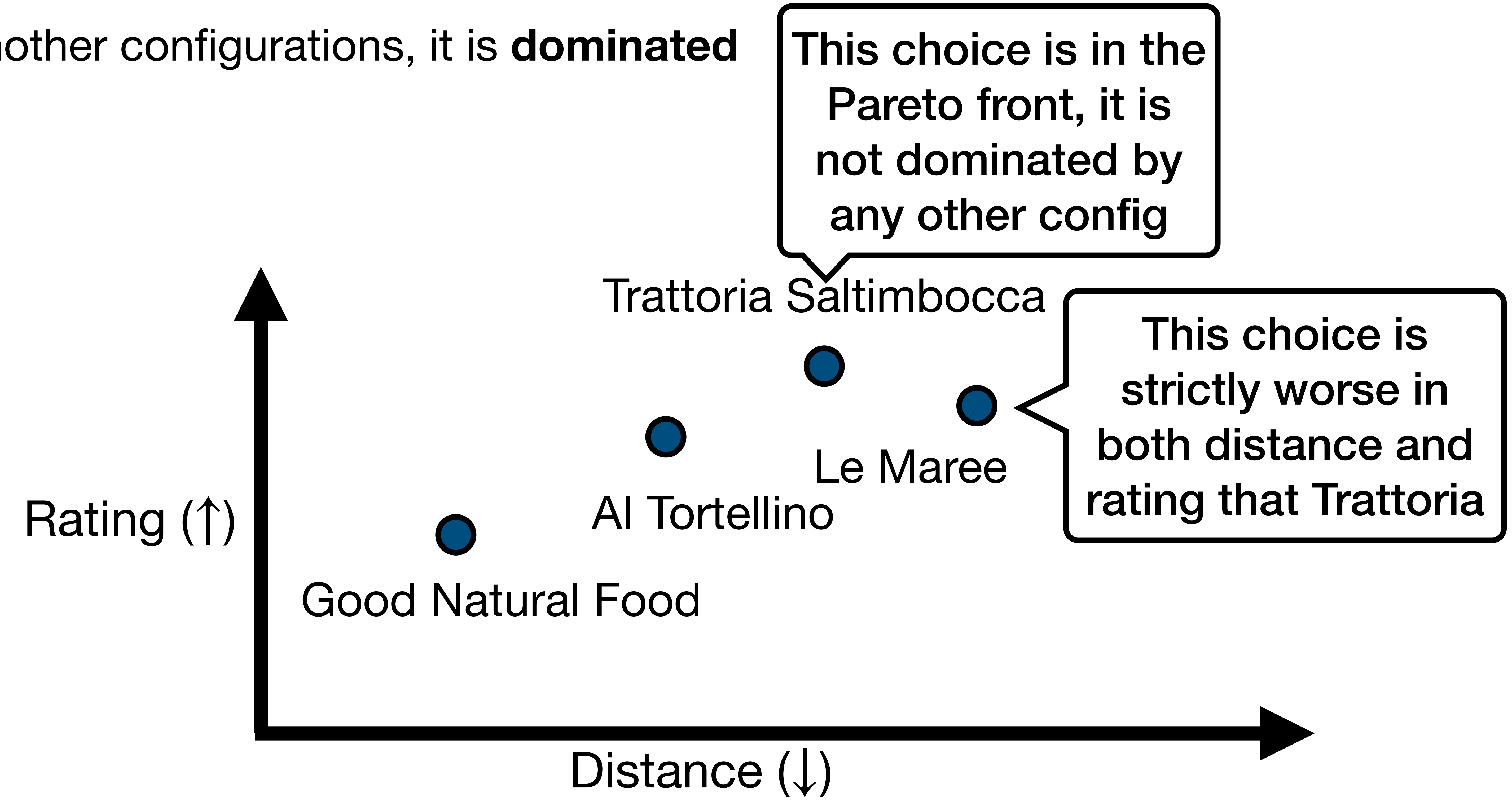




How to select which restaurant to go?

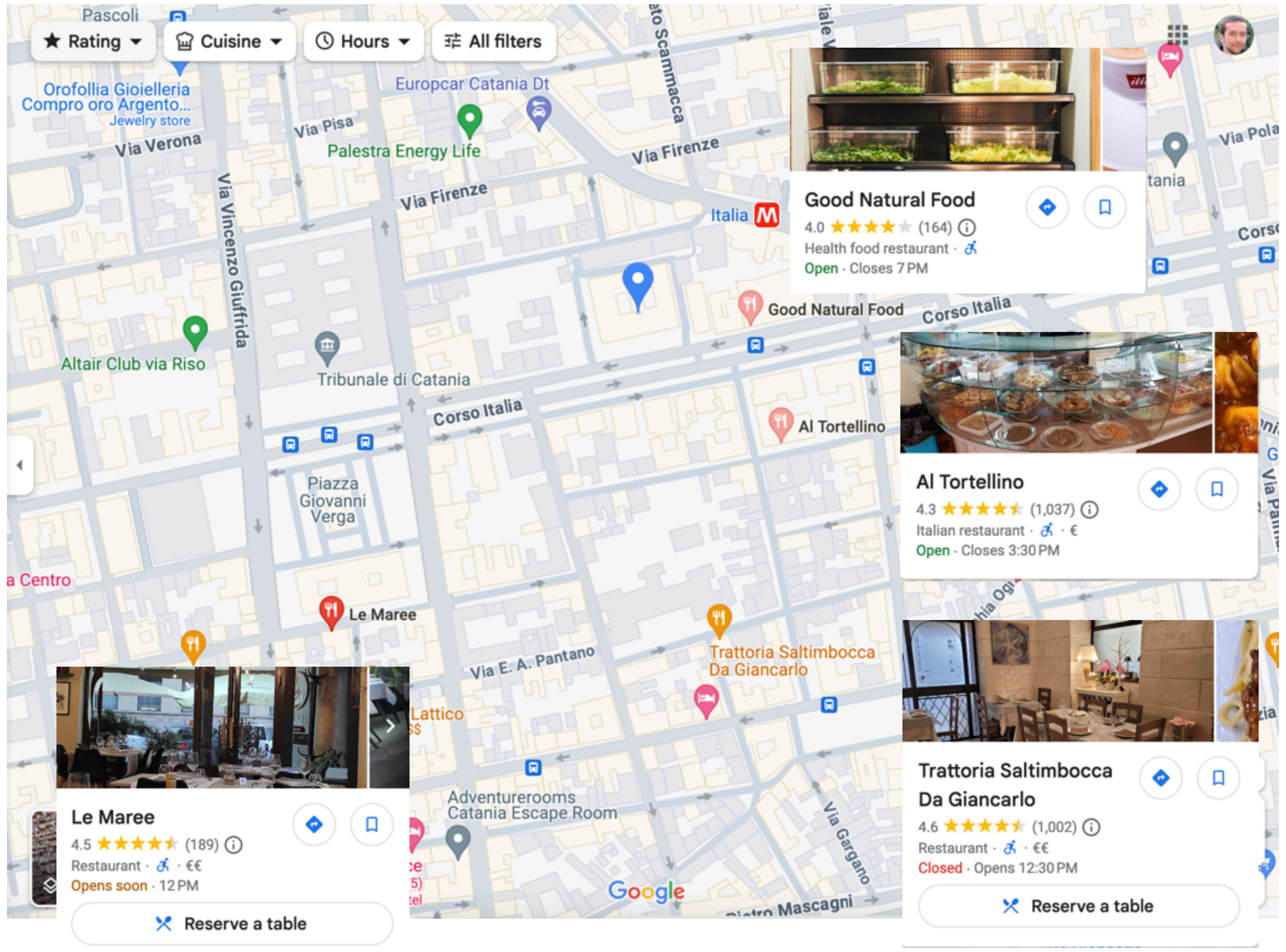
- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal
- If one configuration is worse for all metrics than another configurations, it is **dominated**

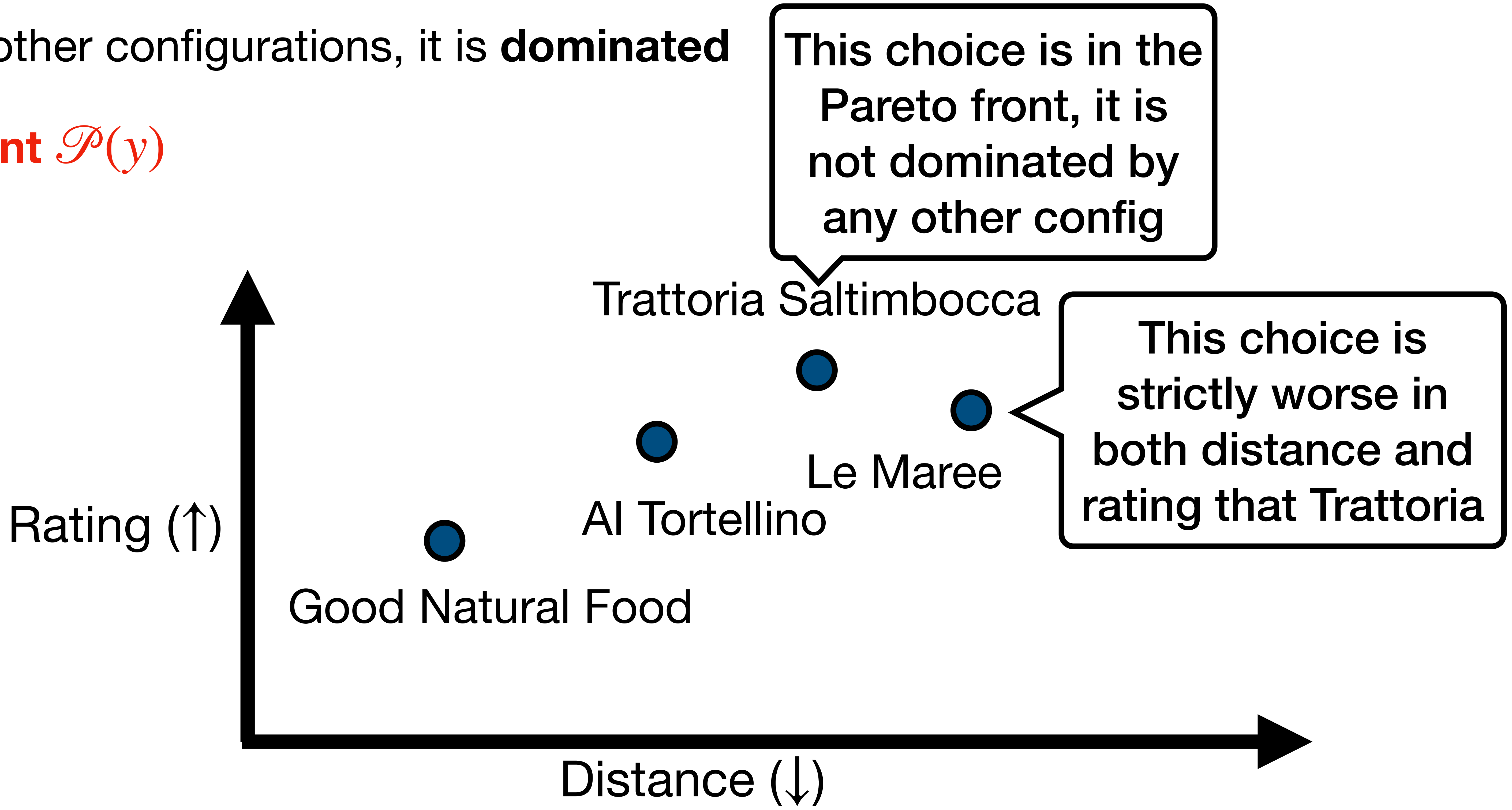




How to select which restaurant to go?

- Plot objective observations $y \in \mathbb{R}^{n \times d}$ where n is the number of configurations (restaurants) and d is the number of objectives
- No just a single best solution!
- 🤔 Which configuration to pick?
- Some choices are sub-optimal
- If one configuration is worse for all metrics than another configurations, it is **dominated**
- The set of non dominated options is the **Pareto front** $\mathscr{P}(y)$





Pareto front

Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized

Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.

Pareto front

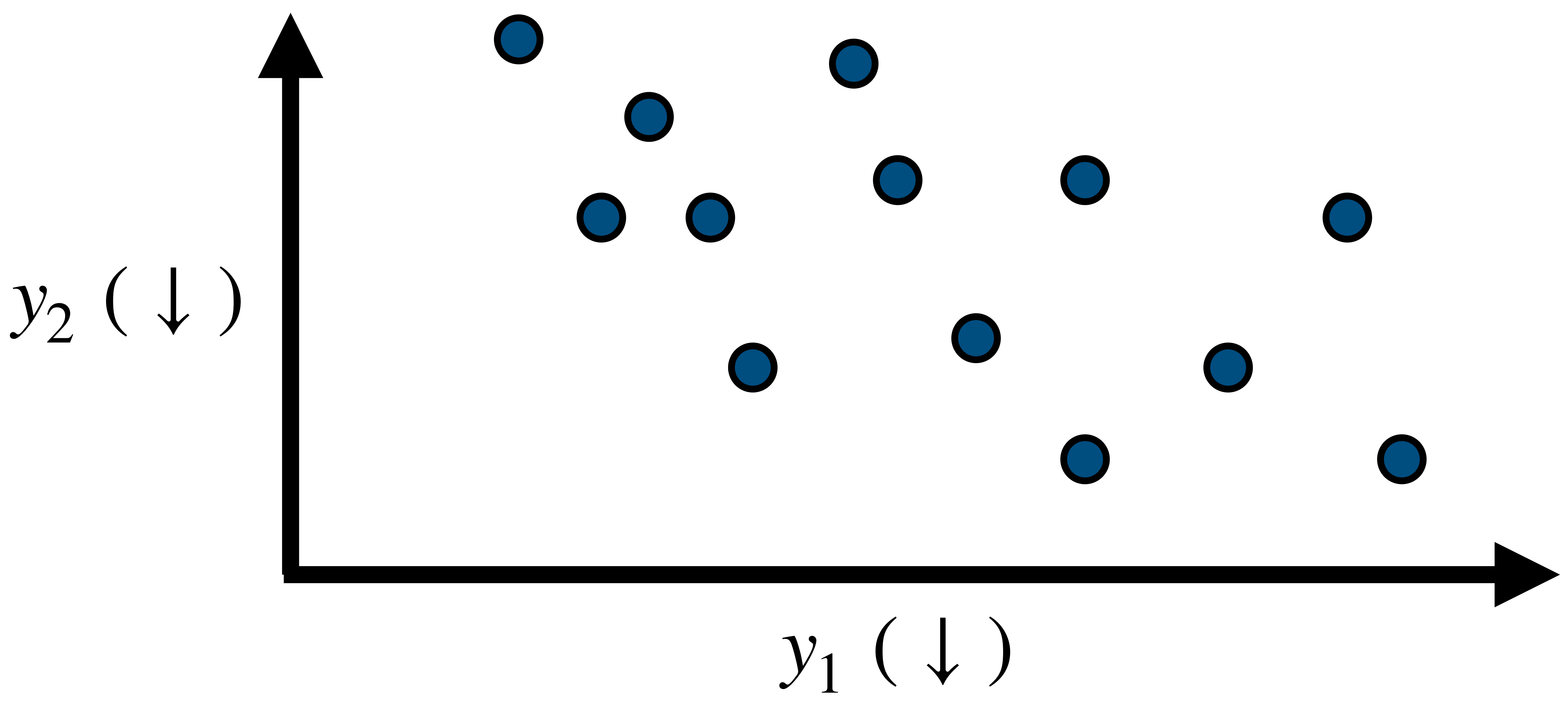
- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \,|\, \nexists y_j \prec y_i\}$

Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathscr{P}(y)$? 🤔

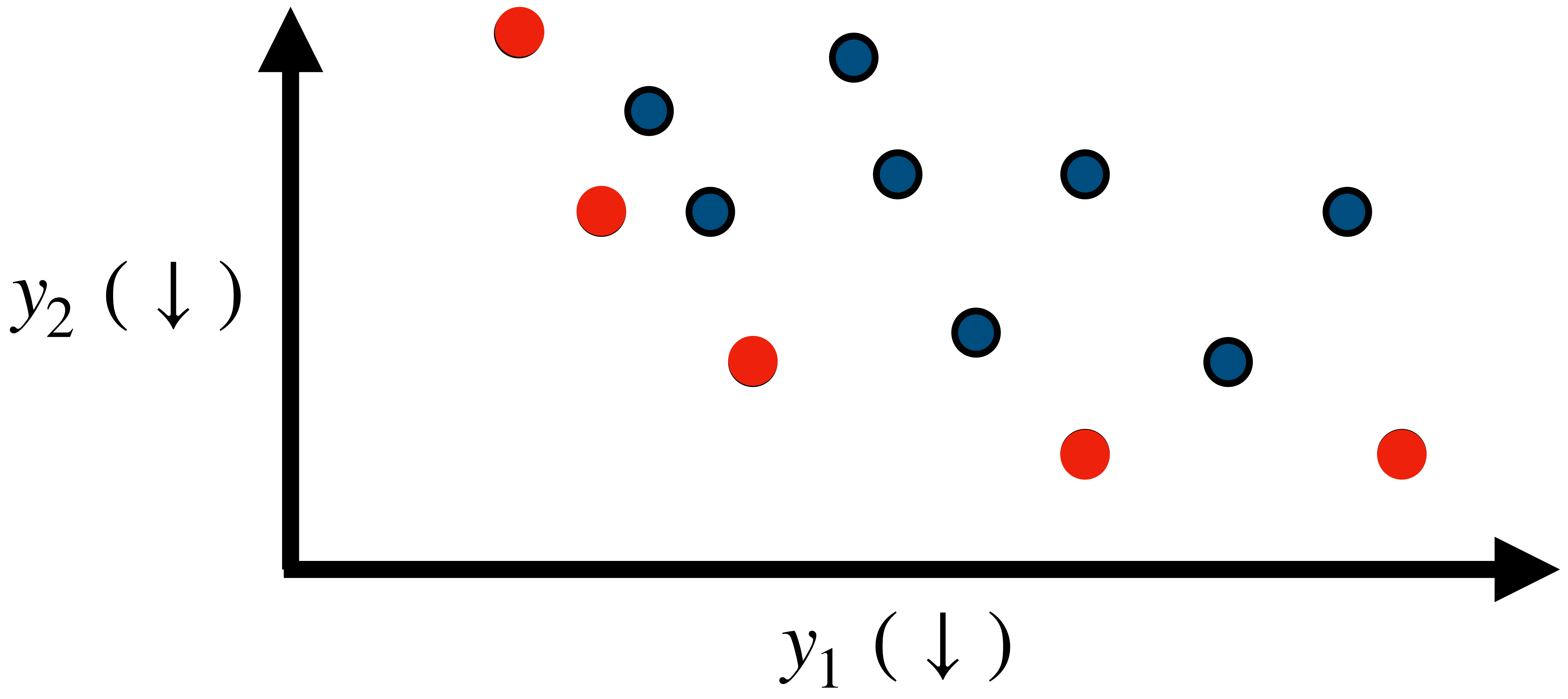
Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathscr{P}(y)$? 🤔



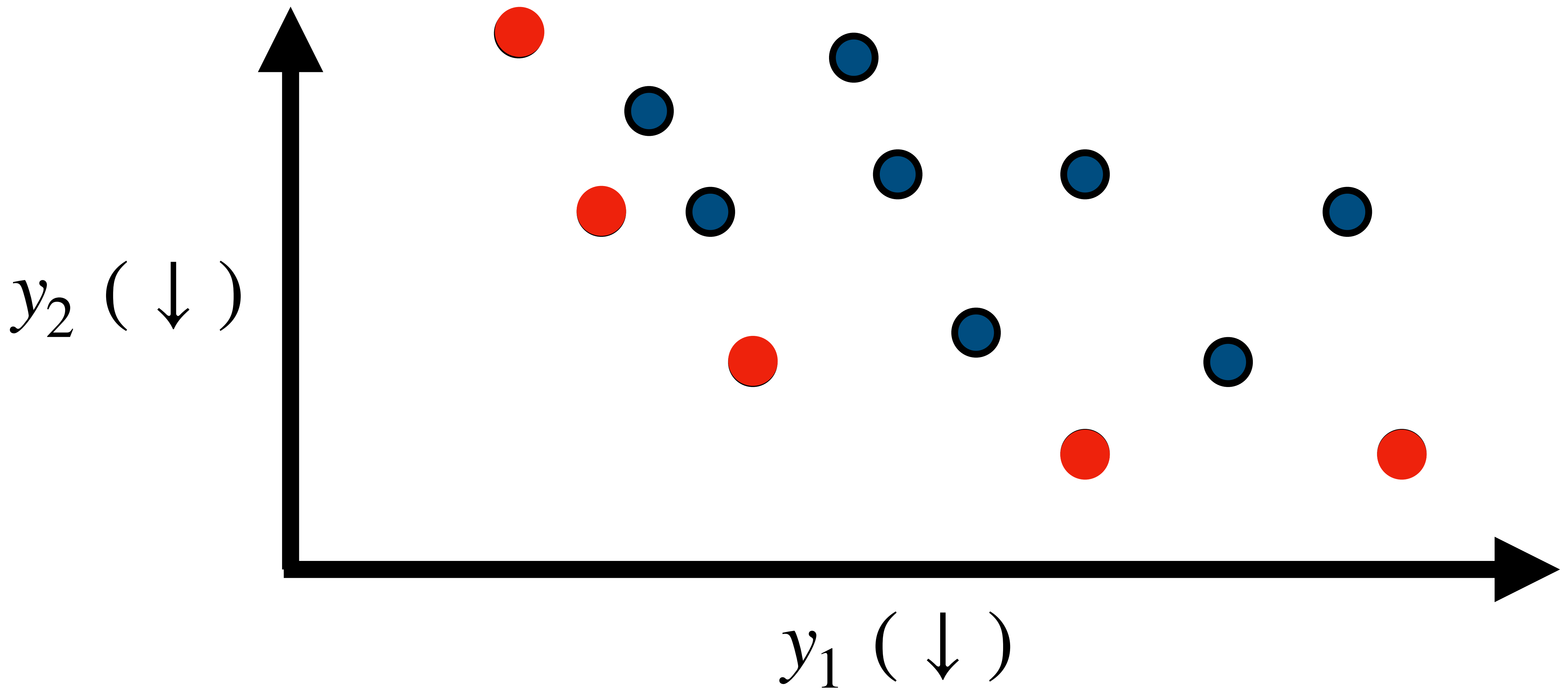
Pareto front

- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathcal{P}(y)$ is the set of non dominated options $\mathcal{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathcal{P}(y)$? 🤔



Pareto front

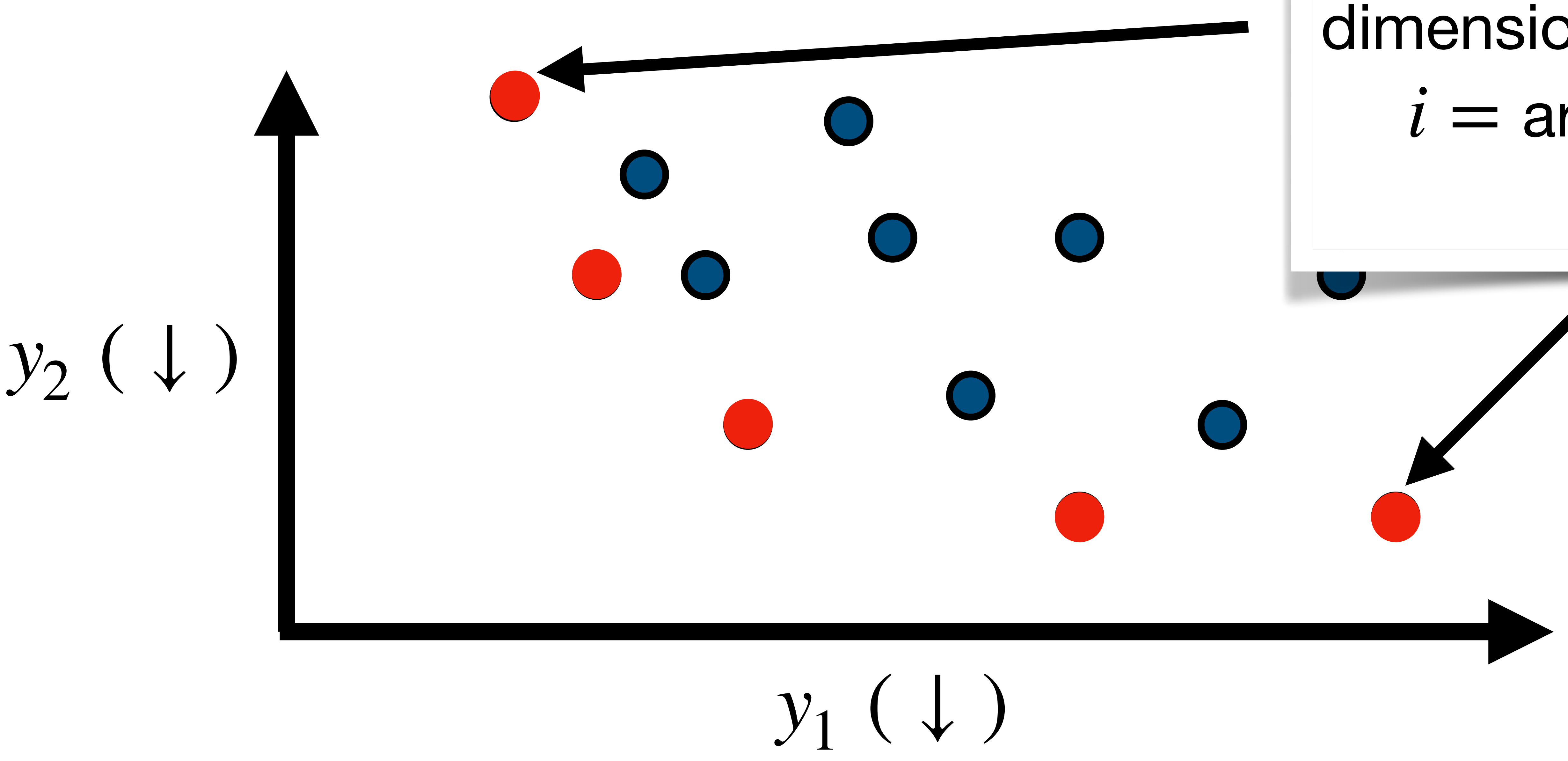
- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathscr{P}(y)$? 🤔
- 🤔 Given objectives $y \in \mathbb{R}^{n \times d}$, can you give a procedure that finds d configurations that are in the Pareto front $\mathscr{P}(y)$ in $\mathcal{O}(d)$?



Pareto front

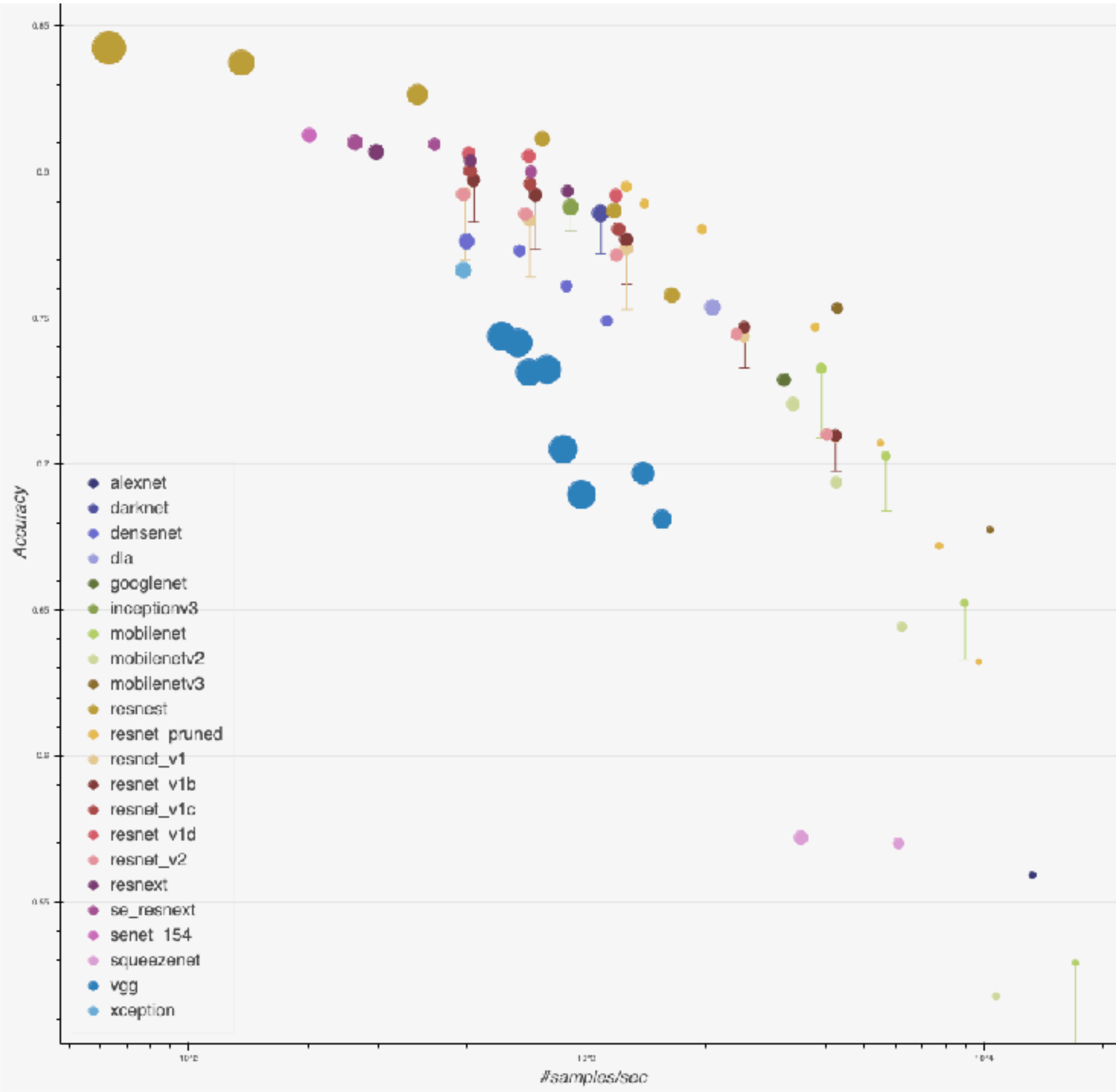
- Metrics observations $y \in \mathbb{R}^{n \times d}$, assume all metrics are to be minimized
- We say that y_i dominate y_j (which we denote $y_i \prec y_j$) iff $\forall k, y_{ik} \leq y_{jk}$ and $\exists k, y_{ik} < y_{jk}$.
- The Pareto front $\mathscr{P}(y)$ is the set of non dominated options $\mathscr{P}(y) = \{y_i \mid \nexists y_j \prec y_i\}$
- 🤔 Which configurations are in the Pareto front $\mathscr{P}(y)$? 🤔
- 🤔 Given objectives $y \in \mathbb{R}^{n \times d}$, can you give a procedure that finds d configurations that are in the Pareto front $\mathscr{P}(y)$ in $\mathcal{O}(d)$?

The configuration minimizing the j-th dimension is always on the Pareto front! $i = \text{argmin}_{i \in [n]} y_{ij}$ is always on the Pareto front



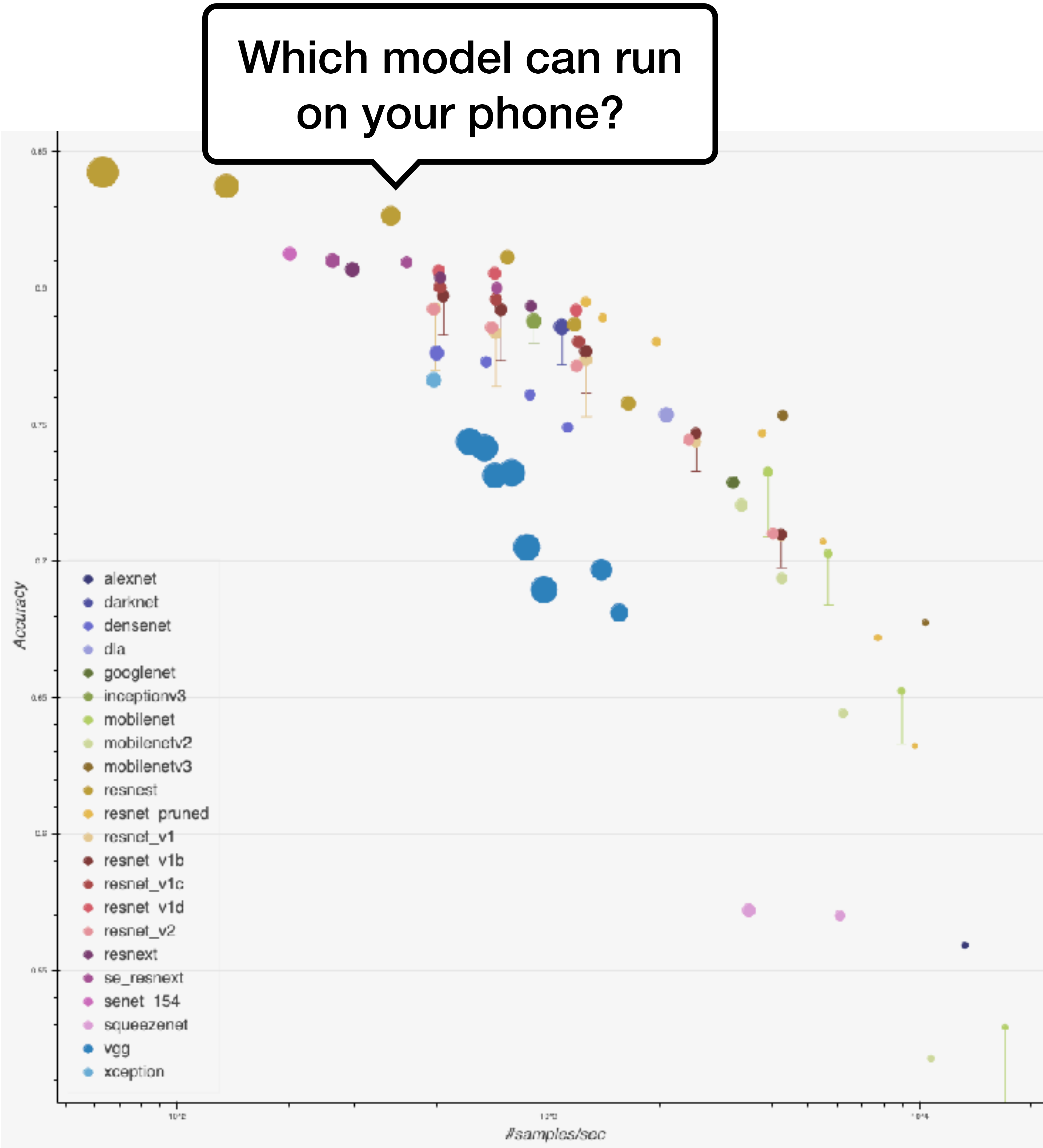
A common use case

A common use case



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

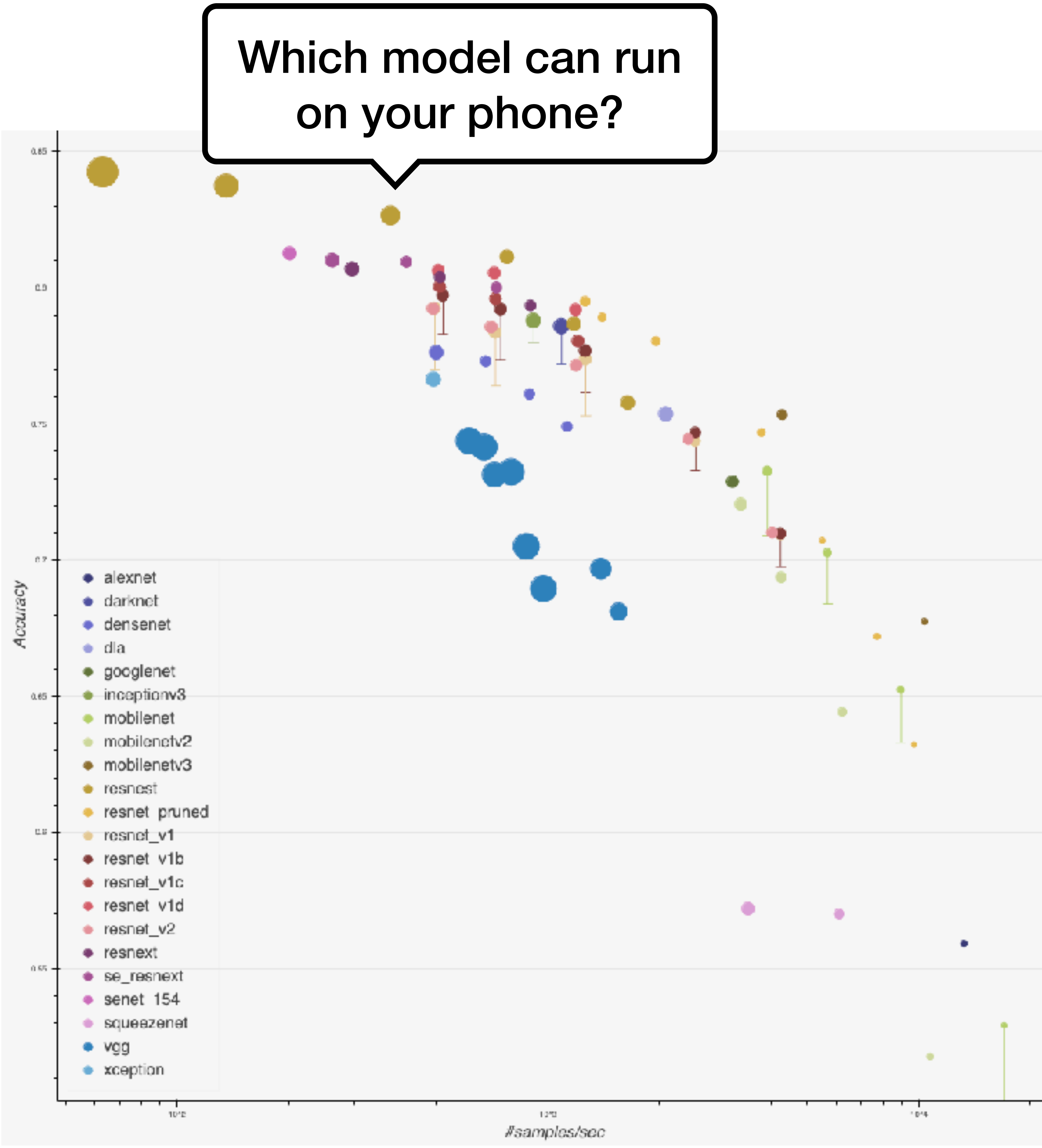
A common use case



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

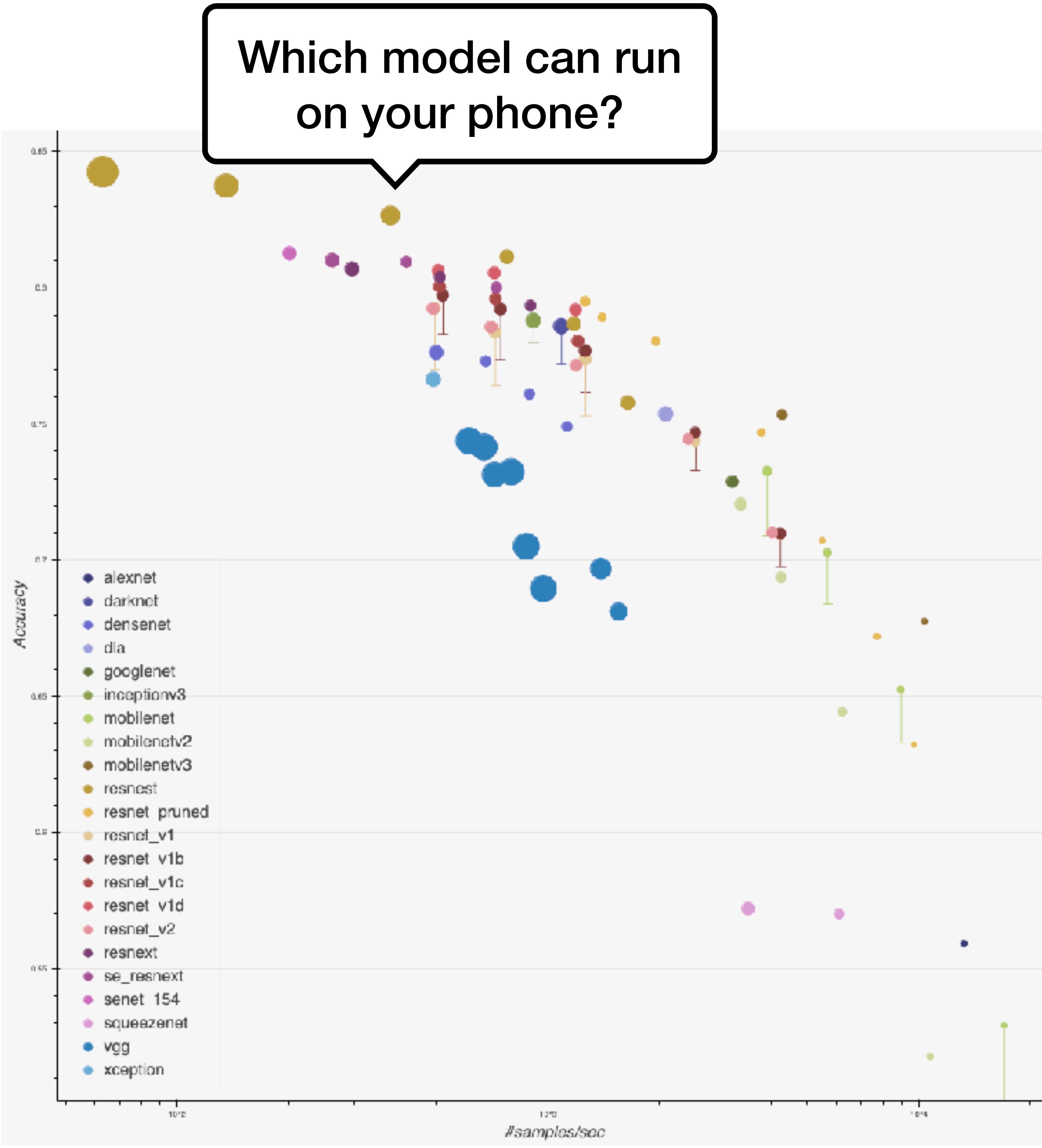
- Multiobjective optimization:



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

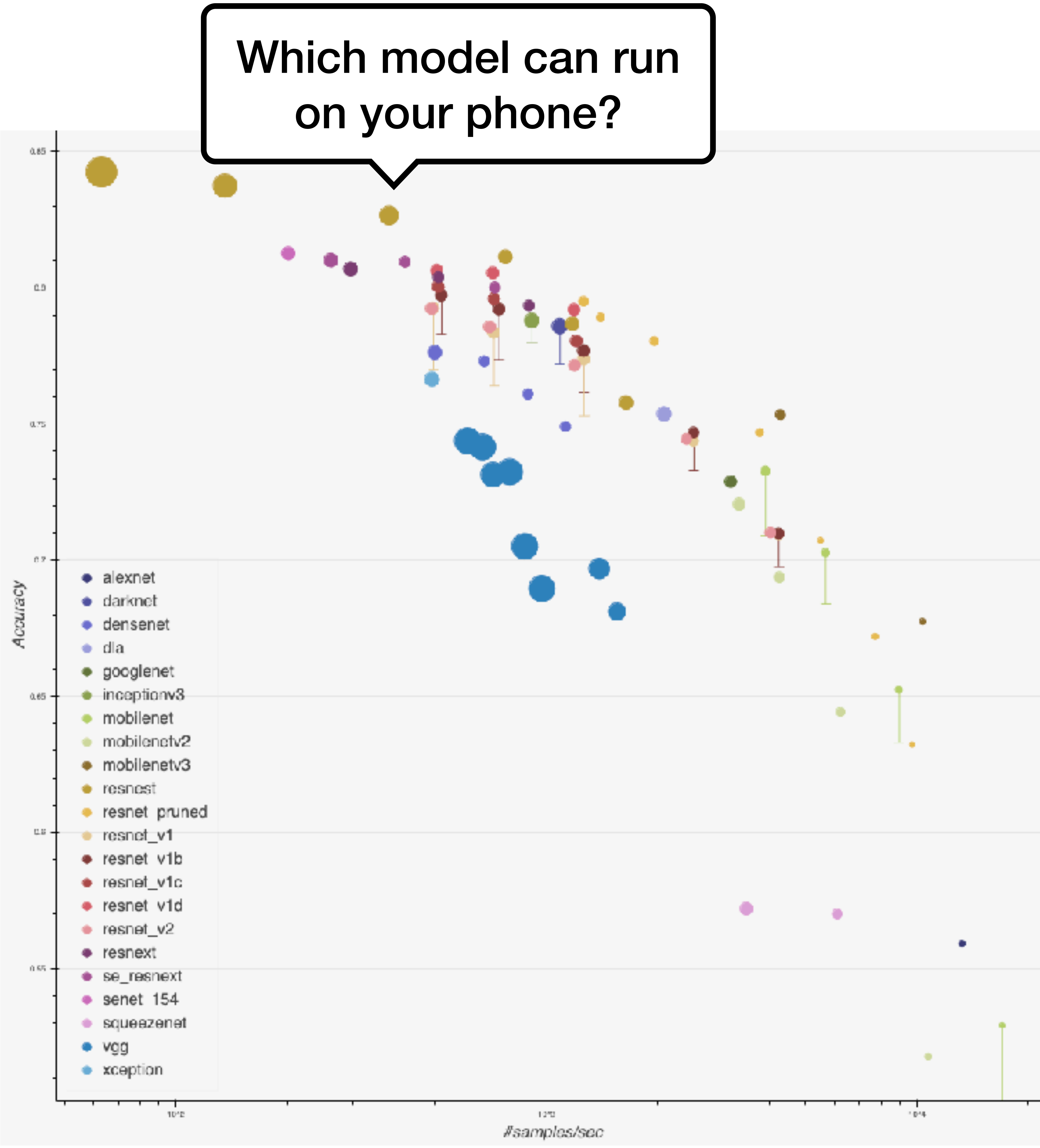
- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

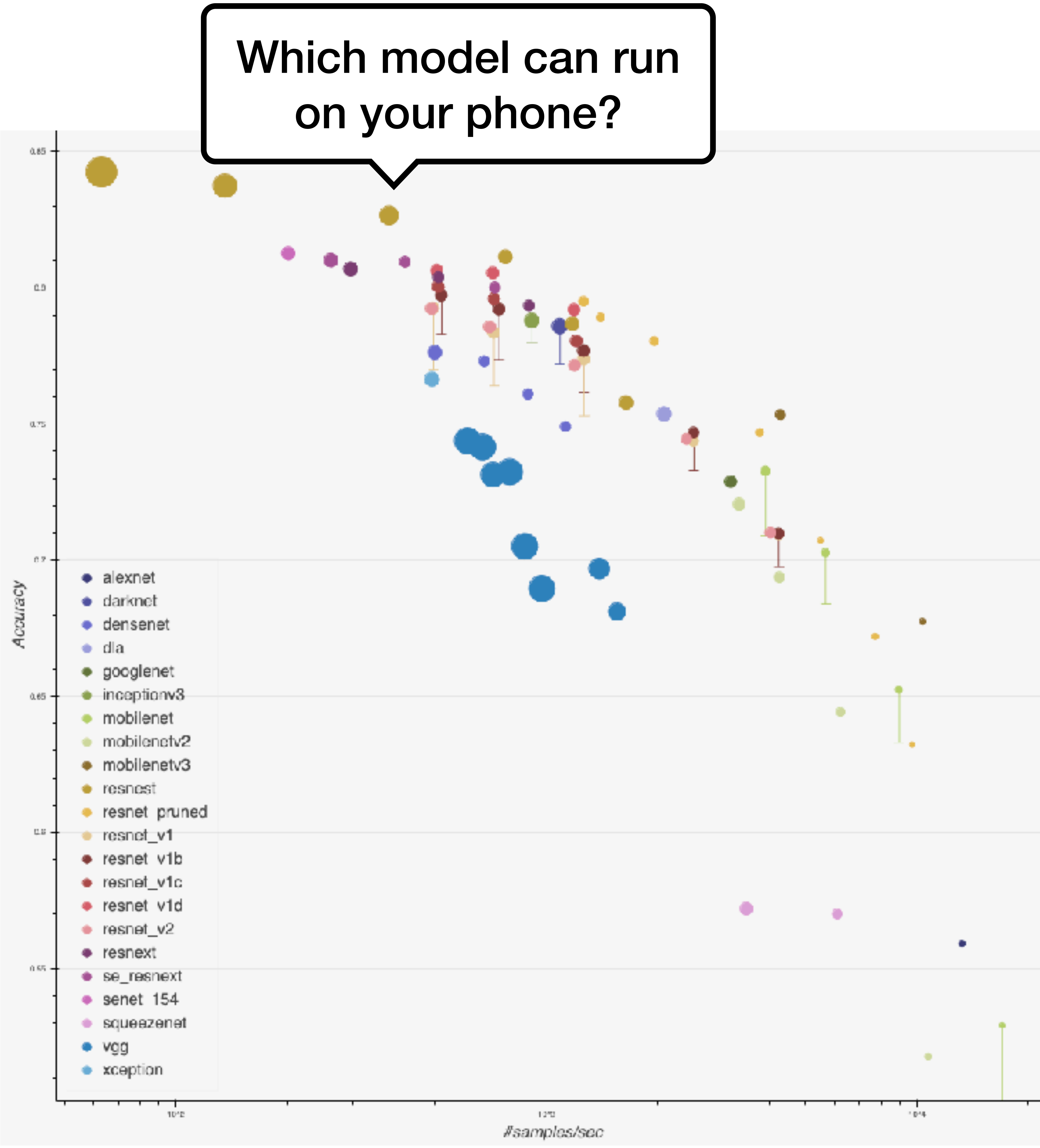
- Multiobjective optimization:
    - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
    - Use as few evaluation call as possible given that f is expensive!



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
  - Use as few evaluation call as possible given that f is expensive!
  - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)

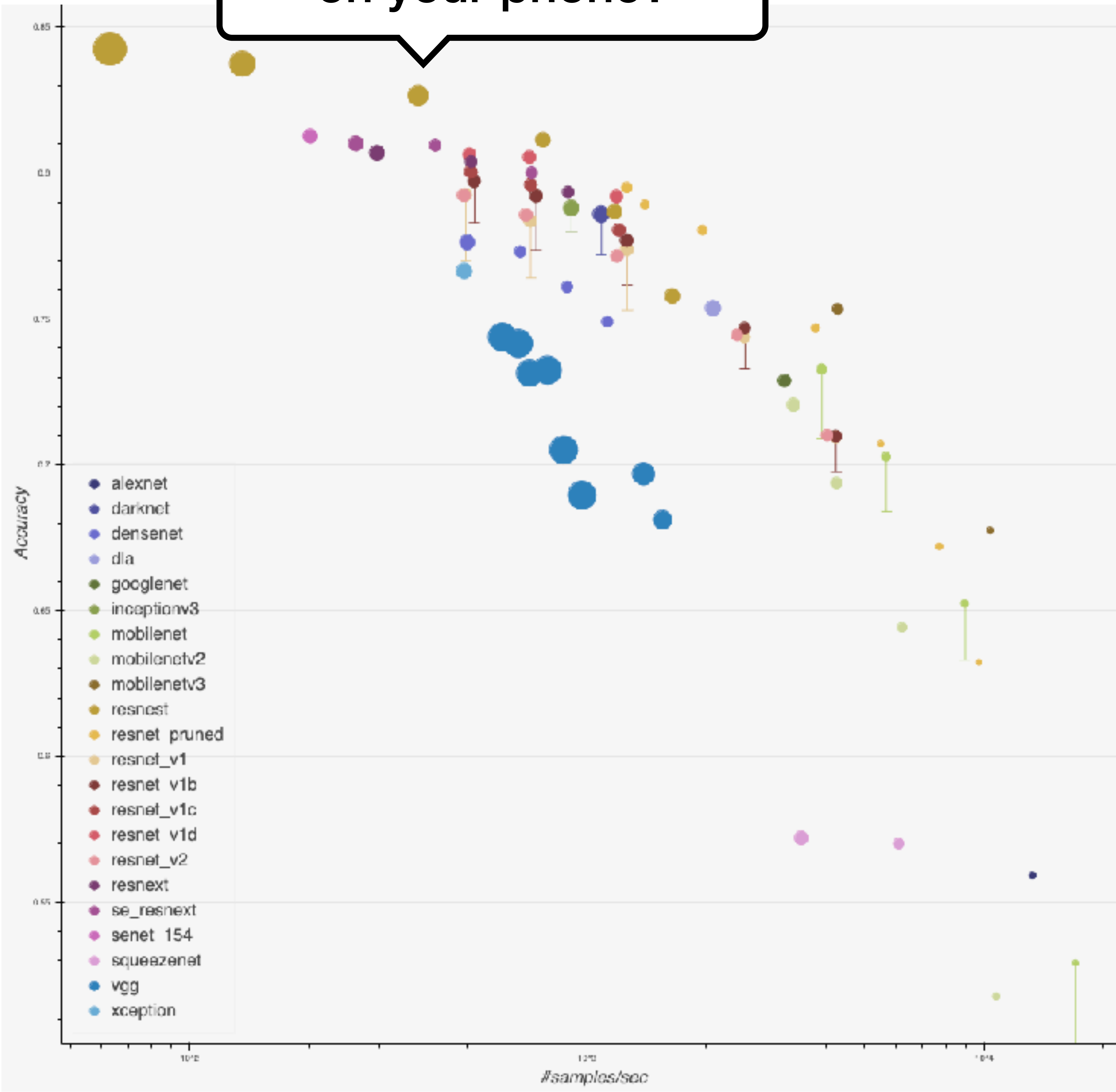


Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

- Multiobjective optimization:
    - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
    - Use as few evaluation call as possible given that f is expensive!
    - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
    - Useful when we dont know the constraint in advance!

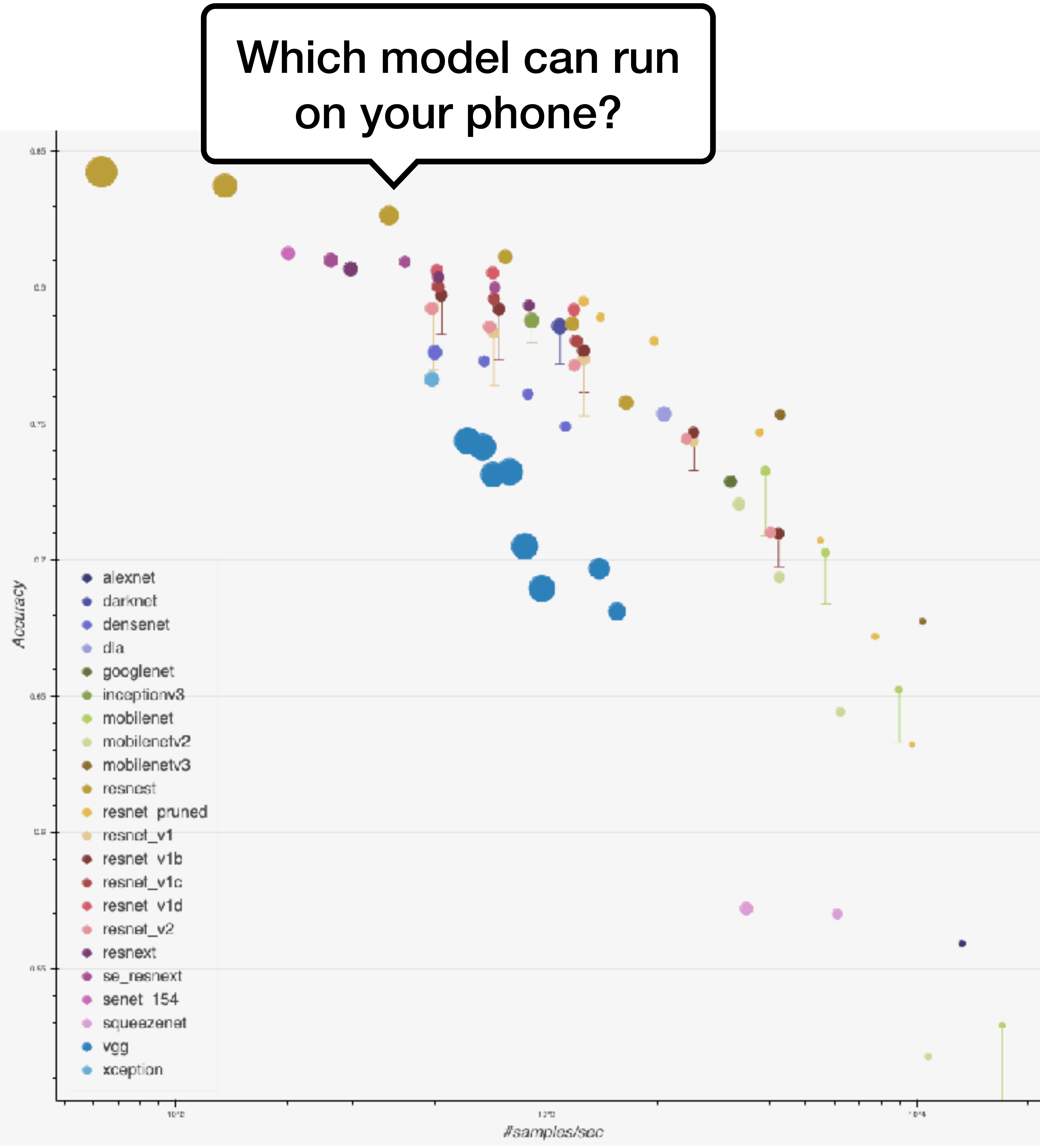
Which model can run on your phone?



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

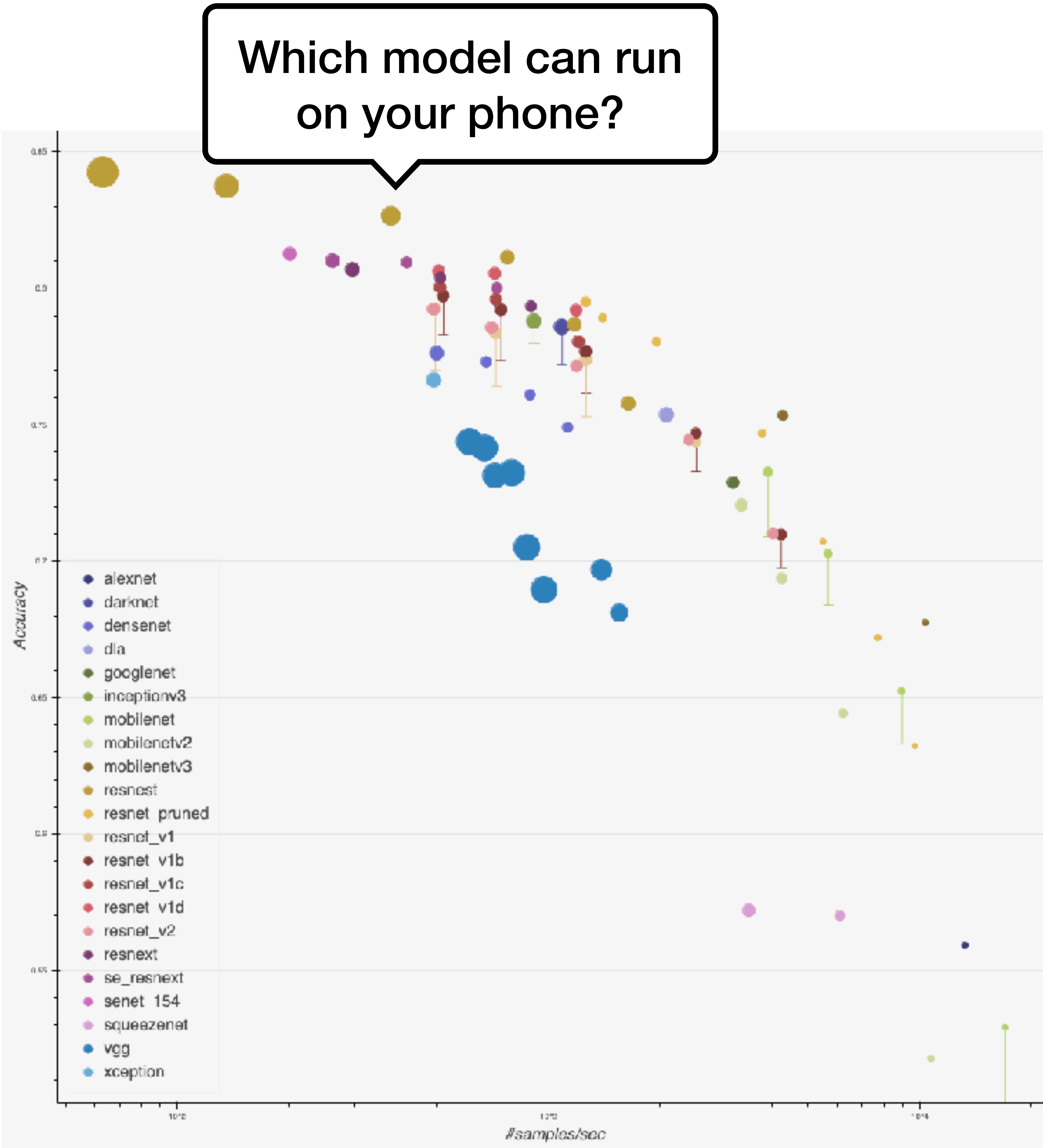
- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
  - Use as few evaluation call as possible given that f is expensive!
  - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
  - Useful when we dont know the constraint in advance!
- Recurring application example in ML:



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

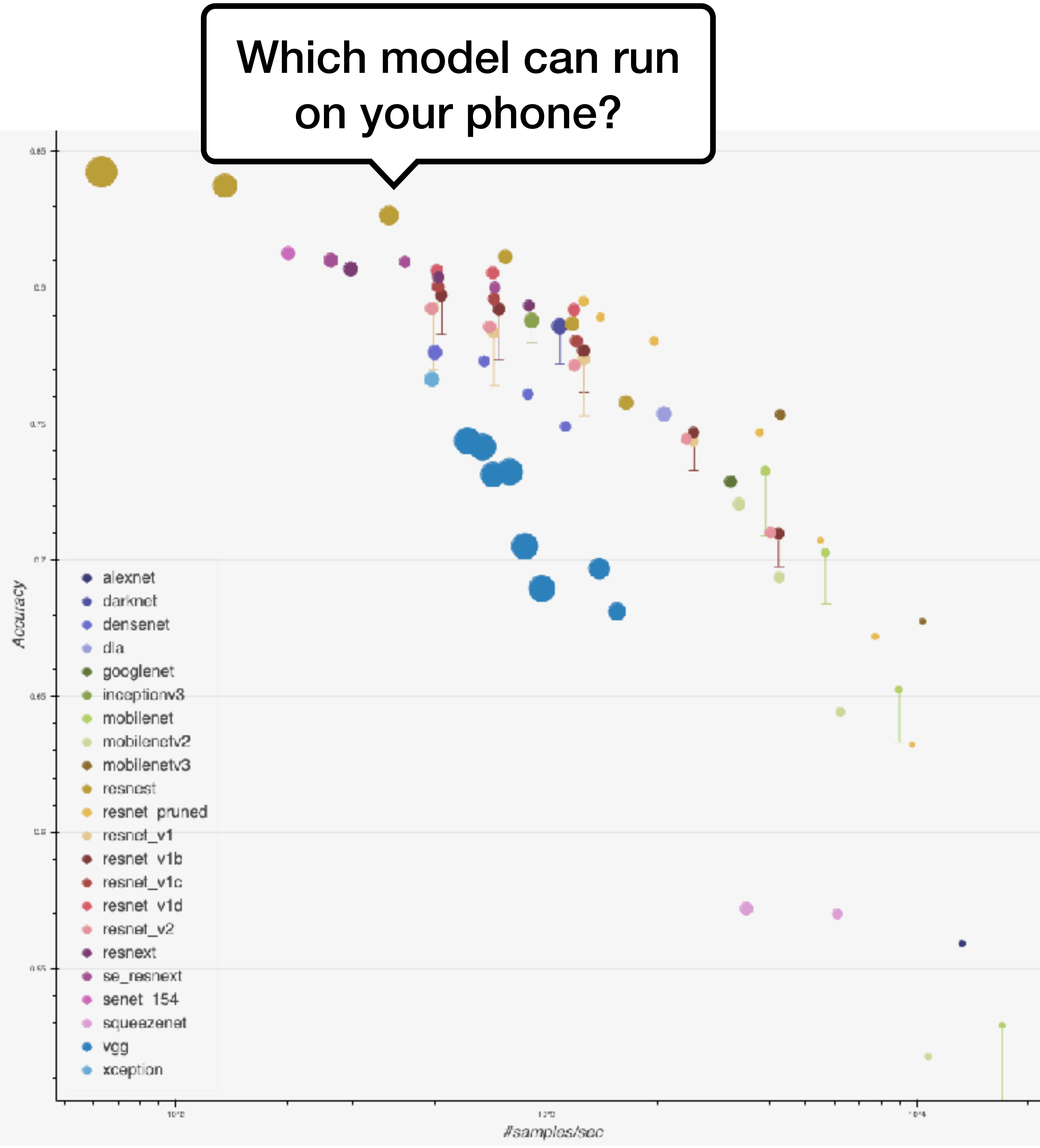
- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
  - Use as few evaluation call as possible given that f is expensive!
  - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
  - Useful when we dont know the constraint in advance!
- Recurring application example in ML:
  - LLM or CV model that trades accuracy to run on a phone with a reasonable latency



Computer vision models in GluonCV. Accuracy and throughput is displayed.

A common use case

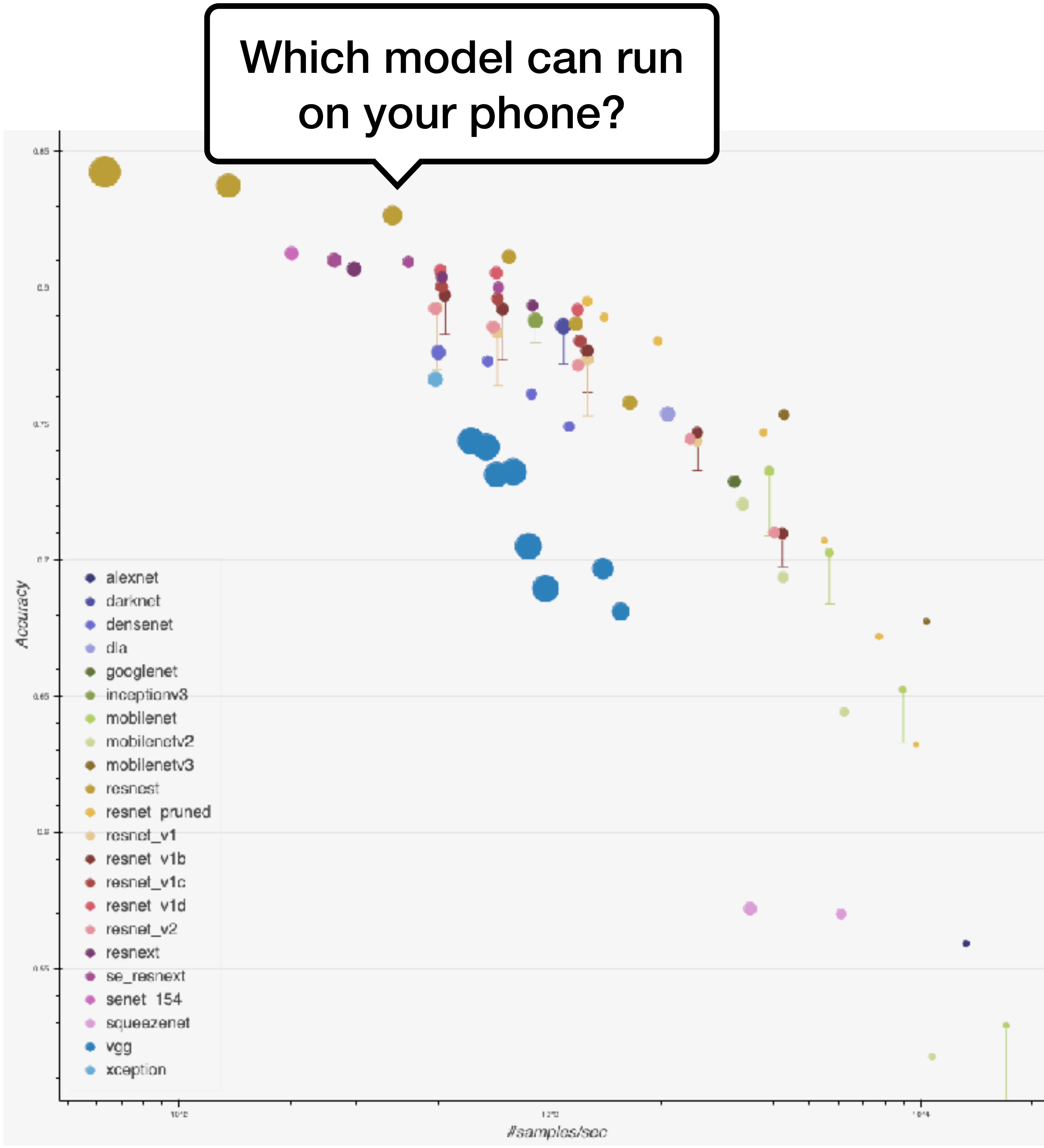
- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
  - Use as few evaluation call as possible given that f is expensive!
  - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
  - Useful when we dont know the constraint in advance!
- Recurring application example in ML:
  - LLM or CV model that trades accuracy to run on a phone with a reasonable latency
- Reasonable latency is hard to give in advance, it depends on the accuracy loss => a posteriori decision



Computer vision models in GluonCV. Accuracy and throughput is displayed.

A common use case

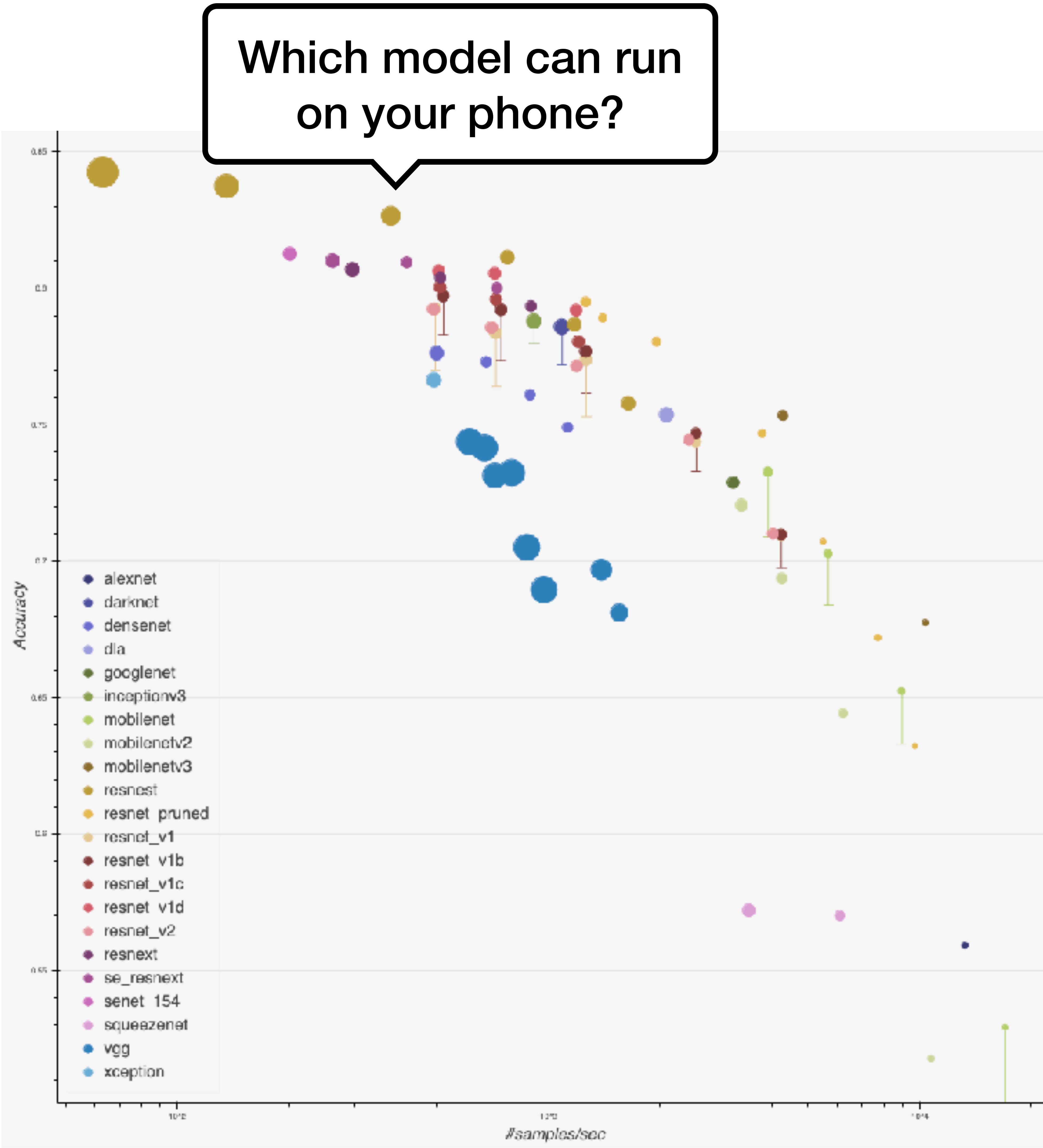
- Multiobjective optimization:
    - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
    - Use as few evaluation call as possible given that f is expensive!
    - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
    - Useful when we dont know the constraint in advance!
- Recurring application example in ML:
    - LLM or CV model that trades accuracy to run on a phone with a reasonable latency
- Reasonable latency is hard to give in advance, it depends on the accuracy loss => a posteriori decision
- Typical-metrics considered:



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

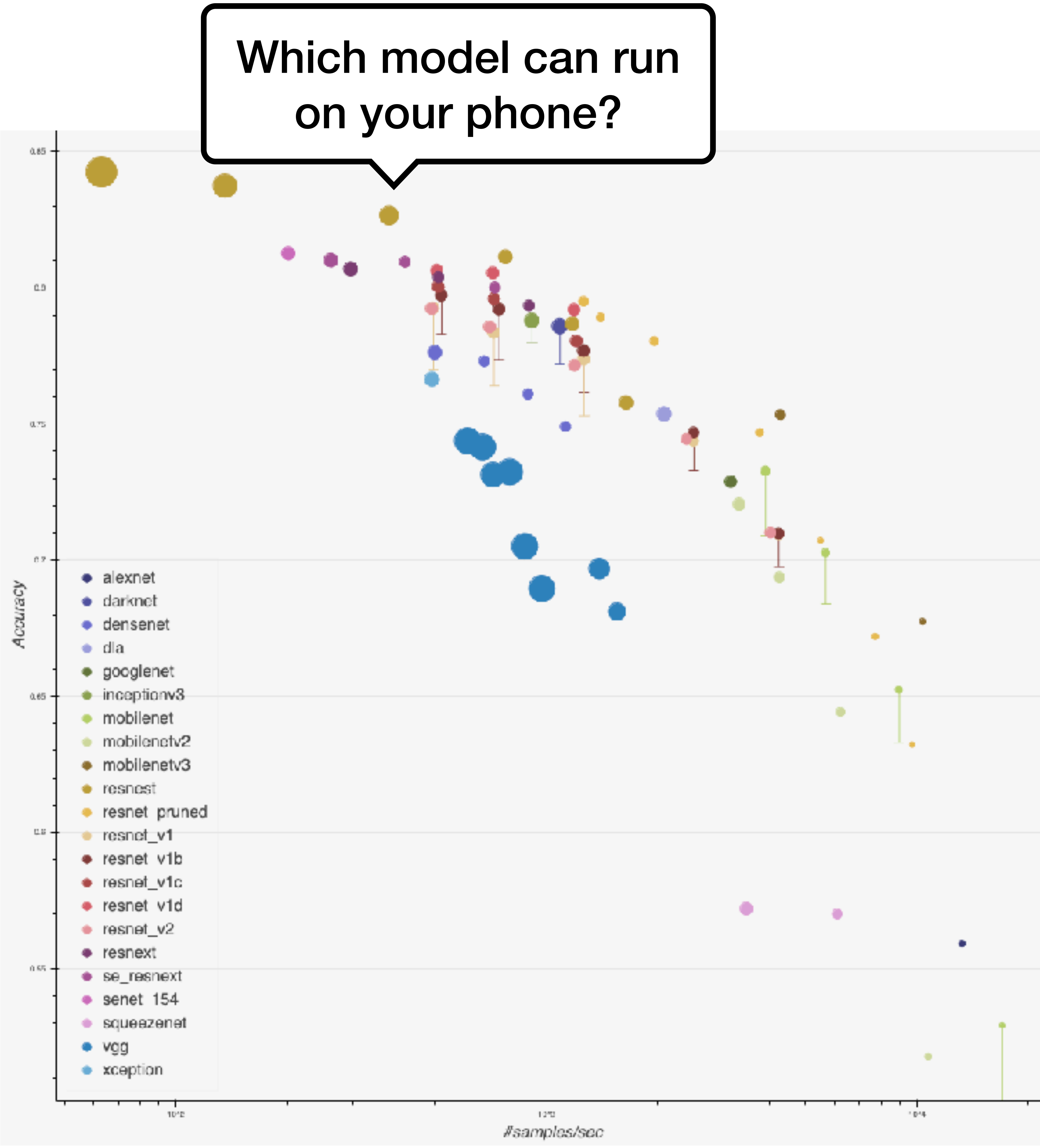
- Multiobjective optimization:
    - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
    - Use as few evaluation call as possible given that f is expensive!
    - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
    - Useful when we dont know the constraint in advance!
- Recurring application example in ML:
    - LLM or CV model that trades accuracy to run on a phone with a reasonable latency
- Reasonable latency is hard to give in advance, it depends on the accuracy loss => a posteriori decision
- Typical-metrics considered:
    - Accuracy



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

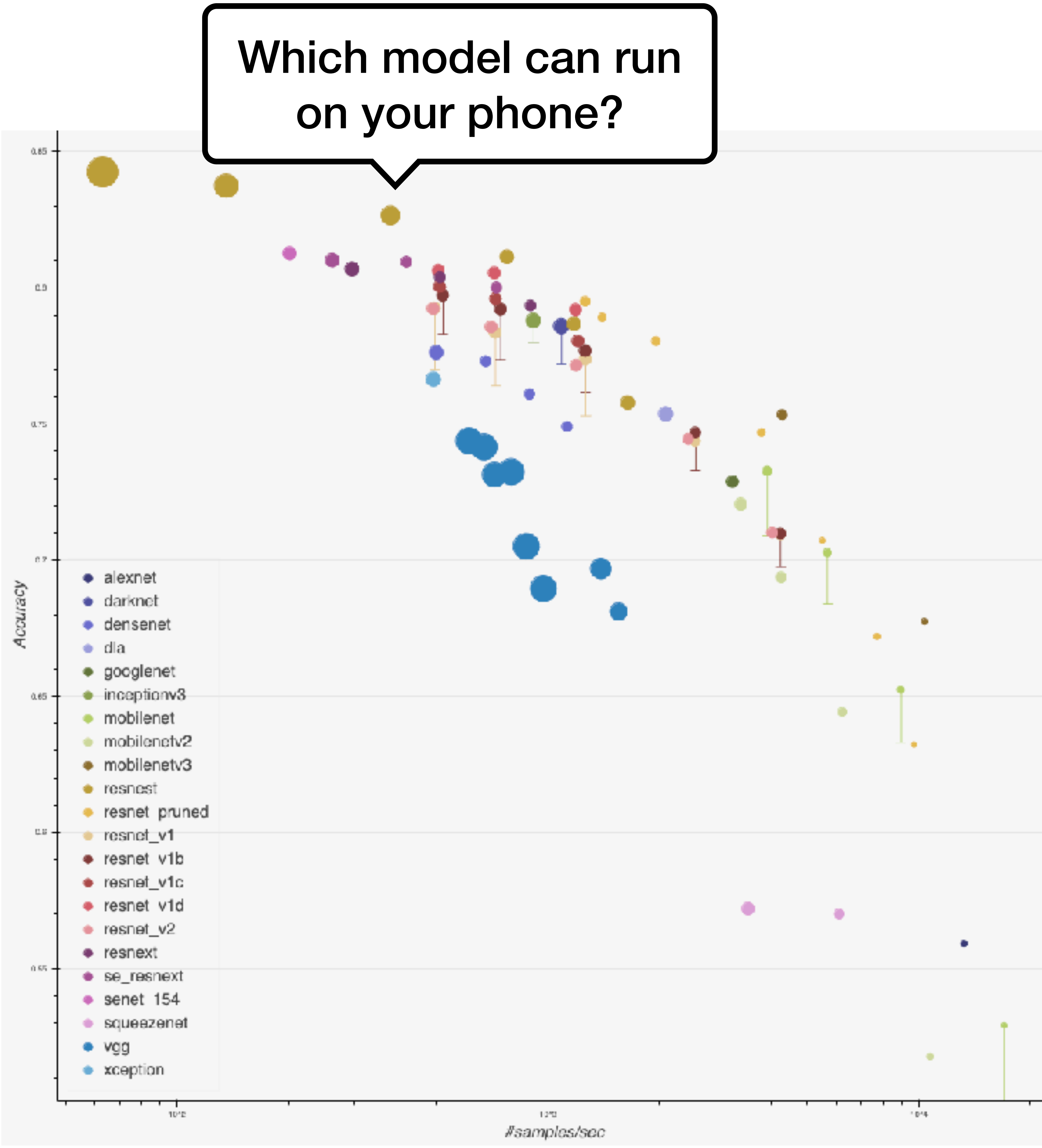
- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
  - Use as few evaluation call as possible given that f is expensive!
  - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
  - Useful when we dont know the constraint in advance!
- Recurring application example in ML:
  - LLM or CV model that trades accuracy to run on a phone with a reasonable latency
- Reasonable latency is hard to give in advance, it depends on the accuracy loss => a posteriori decision
- Typical-metrics considered:
  - Accuracy
  - Latency



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

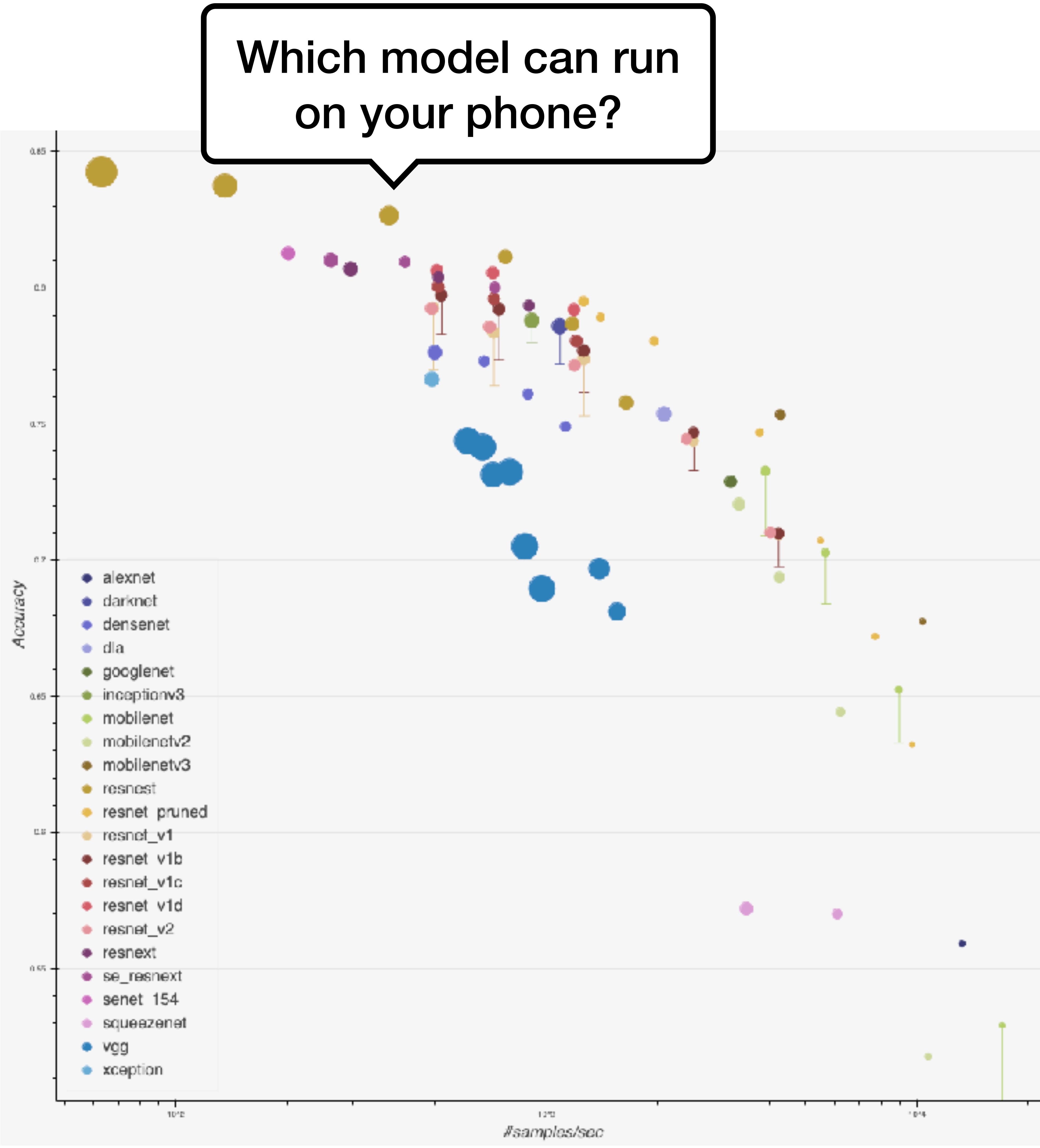
- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
  - Use as few evaluation call as possible given that f is expensive!
  - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
  - Useful when we dont know the constraint in advance!
- Recurring application example in ML:
  - LLM or CV model that trades accuracy to run on a phone with a reasonable latency
- Reasonable latency is hard to give in advance, it depends on the accuracy loss => a posteriori decision
- Typical-metrics considered:
  - Accuracy
  - Latency
  - Fairness



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

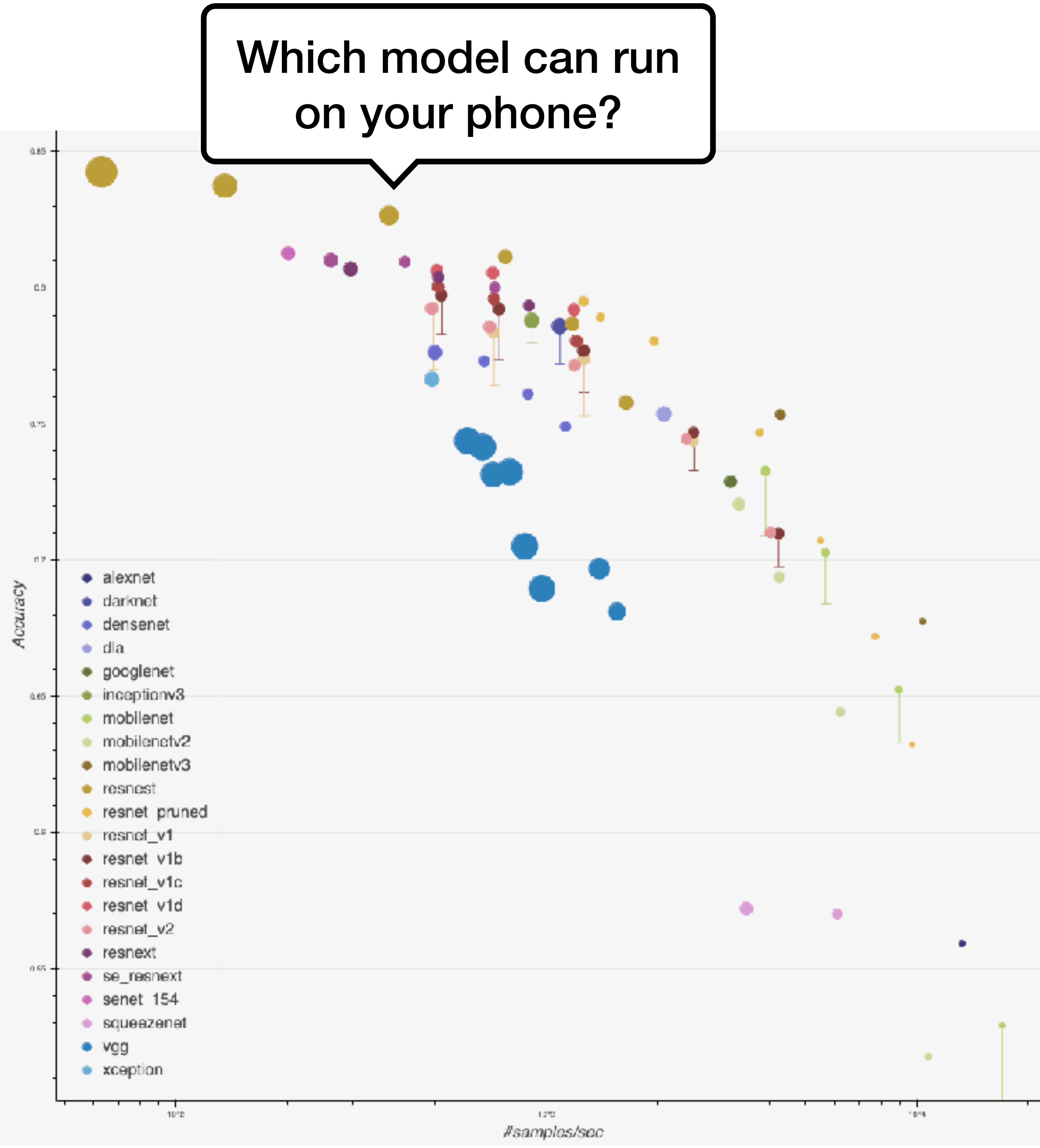
- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
  - Use as few evaluation call as possible given that f is expensive!
  - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
  - Useful when we dont know the constraint in advance!
- Recurring application example in ML:
  - LLM or CV model that trades accuracy to run on a phone with a reasonable latency
- Reasonable latency is hard to give in advance, it depends on the accuracy loss => a posteriori decision
- Typical-metrics considered:
  - Accuracy
  - Latency
  - Fairness
  - Memory consumption



Computer vision models in GluonCV.
Accuracy and throughput is displayed.

A common use case

- Multiobjective optimization:
  - Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front
  - Use as few evaluation call as possible given that f is expensive!
  - It allows to provide optimal choices for **a posteriori** decisions (in the least amount of queries)
  - Useful when we dont know the constraint in advance!
- Recurring application example in ML:
  - LLM or CV model that trades accuracy to run on a phone with a reasonable latency
- Reasonable latency is hard to give in advance, it depends on the accuracy loss => a posteriori decision
- Typical-metrics considered:
  - Accuracy
  - Latency
  - Fairness
  - Memory consumption
  - Energy consumption, …



Computer vision models in GluonCV. Accuracy and throughput is displayed.

Multiobjective landscape

Multiobjective landscape

- Our focus for this lecture is on blackbox multiobjective optimization
- Related areas:
  - Multiobjective RL: optimize an RL for multiple objective (eg a robot that minimize rewards and keep energy down)
  - Constrained optimization when constraint is known a priori (optimise while penalizing the constraint violation)
  - Multivariate analysis (Time series forecasting, causal analysis …)

Menù del giorno

Menù del giorno

- Problem formation and evaluations metrics ~10 min

Menù del giorno

- Problem formation and evaluations metrics ~10 min
- Optimization Methods ~30 min

Menù del giorno

- Problem formation and evaluations metrics ~10 min
- Optimization Methods ~30 min
- Theoretical foundations ~5 min

Menù del giorno

- Problem formation and evaluations metrics ~10 min
- Optimization Methods ~30 min
- Theoretical foundations ~5 min
- Applications ~15 min

Menù del giorno

- Problem formation and evaluations metrics ~10 min
- Optimization Methods ~30 min
- Theoretical foundations ~5 min
- Applications ~15 min
- Code and library ~5 min

Menù del giorno

- Problem formation and evaluations metrics ~10 min
- Optimization Methods ~30 min
- Theoretical foundations ~5 min
- Applications ~15 min
- Code and library ~5 min
- Conclusion

Problem formation and evaluations metrics

Hyperparameter optimization

Recap for single objective case

Hyperparameter optimization

Recap for single objective case

- Hyperparameter simple setting, find the best hyperparameter of $f(x) \in \mathbb{R}$

Hyperparameter optimization

Recap for single objective case

- Hyperparameter simple setting, find the best hyperparameter of $f(x) \in \mathbb{R}$
  - $x^* = \text{argmin}_{x \in \mathcal{X}} f(x)$

Hyperparameter optimization

Recap for single objective case

- Hyperparameter simple setting, find the best hyperparameter of $f(x) \in \mathbb{R}$
  - $x^* = \text{argmin}_{x \in \mathcal{X}} f(x)$
- $f(x)$ can be the accuracy obtained after training a neural network with hyperparameter x

Hyperparameter optimization

Recap for single objective case

- Hyperparameter simple setting, find the best hyperparameter of $f(x) \in \mathbb{R}$
  - $x^* = \text{argmin}_{x \in \mathcal{X}} f(x)$
- $f(x)$ can be the accuracy obtained after training a neural network with hyperparameter x
- $f(x)$ is assumed to be expensive, we want as little sample as possible

Hyperparameter optimization

Recap for single objective case

- Hyperparameter simple setting, find the best hyperparameter of $f(x) \in \mathbb{R}$
    - $x^* = \text{argmin}_{x \in \mathcal{X}} f(x)$
- $f(x)$ can be the accuracy obtained after training a neural network with hyperparameter x
- $f(x)$ is assumed to be expensive, we want as little sample as possible

An example of a search space $\mathcal{X}$

Hyperparameter	**Range**	**Scale**
Architecture	{ConvNext, ViT, EfficientNet}	Discrete
Dropout	$[0.0, 1.0]$	Uniform
Optimizer	{SGD, Adam, RMSProp}	Discrete
Learning Rate	$[10^{-5}, 10^{0}]$	Log

Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

Hyperparameter optimization

Recap for single objective case

- Hyperparameter simple setting, find the best hyperparameter of $f(x) \in \mathbb{R}$
  - $x^* = \text{argmin}_{x \in \mathcal{X}} f(x)$
- $f(x)$ can be the accuracy obtained after training a neural network with hyperparameter x
- $f(x)$ is assumed to be expensive, we want as little sample as possible

An example of a search space $\mathcal{X}$

Hyperparameter	Range	Scale
Architecture	{ConvNext, ViT, EfficientNet}	Discrete
Dropout	$[0.0, 1.0]$	Uniform
Optimizer	{SGD, Adam, RMSProp}	Discrete
Learning Rate	$[10^{-5}, 10^{0}]$	Log

Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

For instance, $x^* = [\text{ViT}, 0.2, \text{Adam}, 10^{-4}]$

Hyperparameter optimization

Recap for single objective case

- Hyperparameter simple setting, find the best hyperparameter of $f(x) \in \mathbb{R}$
  - $x^* = \text{argmin}_{x \in \mathcal{X}} f(x)$
- $f(x)$ can be the accuracy obtained after training a neural network with hyperparameter x
- $f(x)$ is assumed to be expensive, we want as little sample as possible

How to extend to multiple objectives?

An example of a search space $\mathcal{X}$

Hyperparameter	Range	Scale
Architecture	{ConvNext, ViT, EfficientNet}	Discrete
Dropout	$[0.0, 1.0]$	Uniform
Optimizer	{SGD, Adam, RMSProp}	Discrete
Learning Rate	$[10^{-5}, 10^{0}]$	Log

Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

For instance, $x^* = [\text{ViT}, 0.2, \text{Adam}, 10^{-4}]$

Problem formation and evaluations metrics

Problem formation and evaluations metrics

- Given a function $f : \mathbb{R}^m \to \mathbb{R}^d$ with m input dimensions and d objectives

Problem formation and evaluations metrics

- Given a function $f : \mathbb{R}^m \to \mathbb{R}^d$ with m input dimensions and d objectives
- Approximate the Pareto front in the least amount of iterations

Problem formation and evaluations metrics

- Given a function $f : \mathbb{R}^m \to \mathbb{R}^d$ with m input dimensions and d objectives
- Approximate the Pareto front in the least amount of iterations
- We need a notion of distance to quantify how far we are from the optimal solution to be able to formulate an optimization problem!

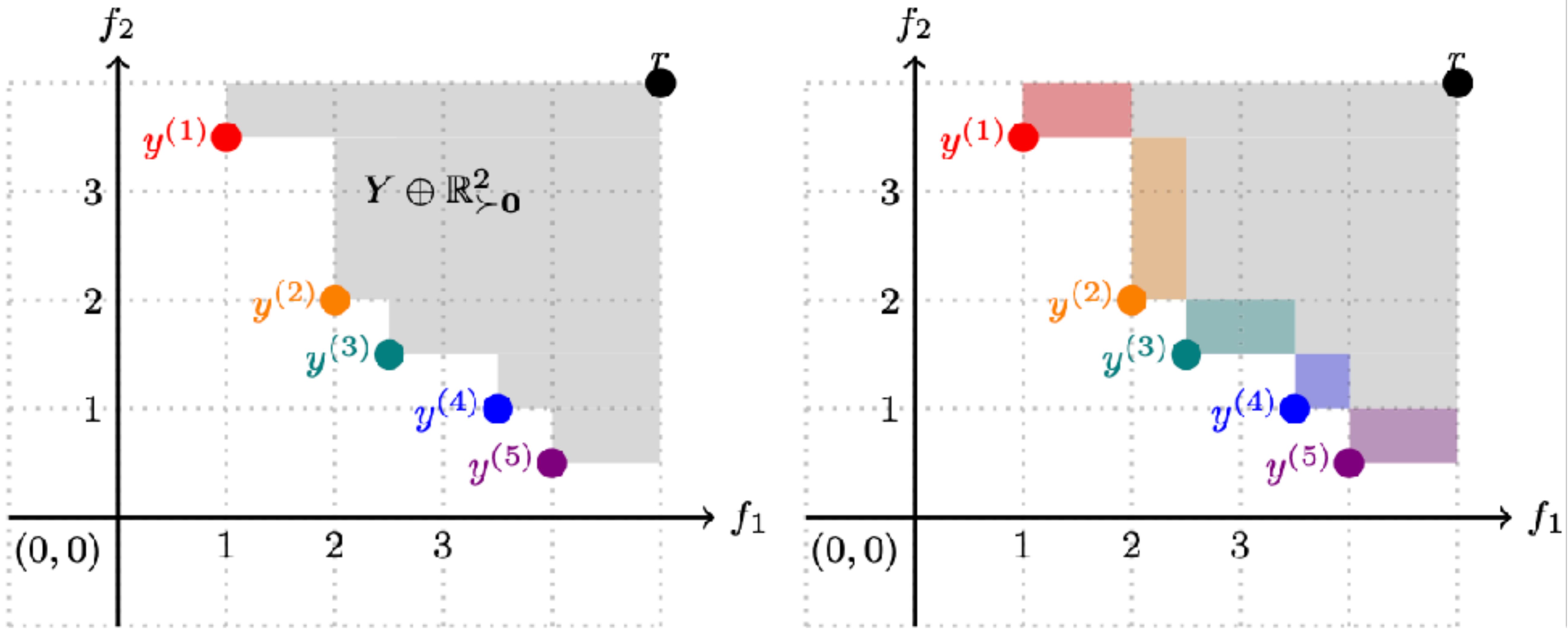
Problem formation and evaluations metrics

- Given a function $f : \mathbb{R}^m \rightarrow \mathbb{R}^d$ with m input dimensions and d objectives
- Approximate the Pareto front in the least amount of iterations
- We need a notion of distance to quantify how far we are from the optimal solution to be able to formulate an optimization problem!
- The hypervolume is the most common metric

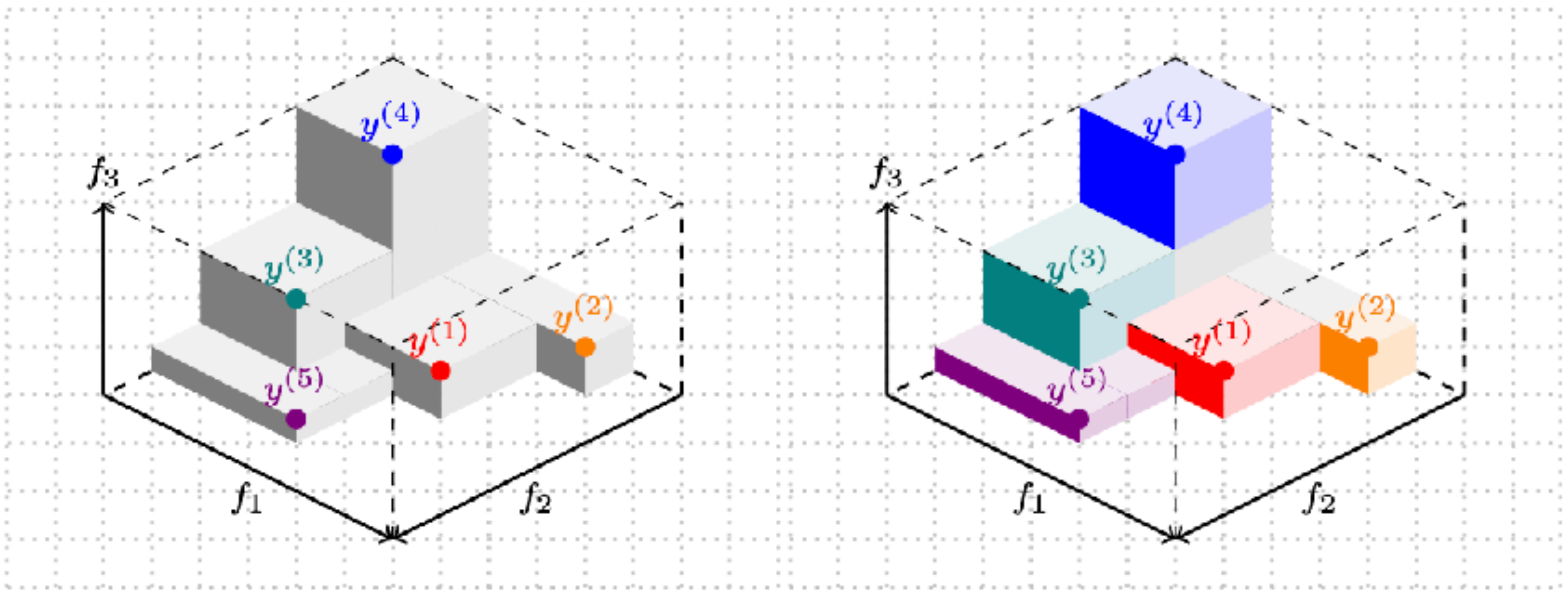
Hypervolume

Measuring multiobjective performance

A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]



2D



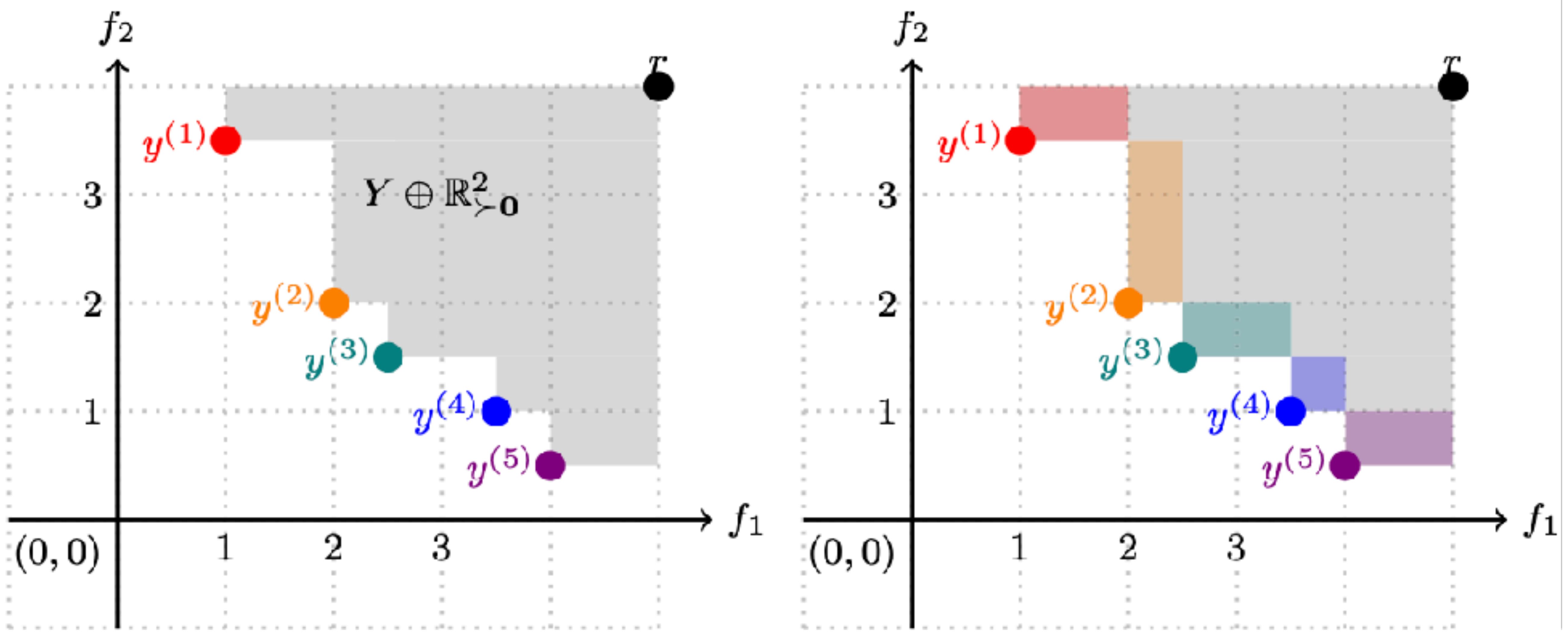
3D

Hypervolume

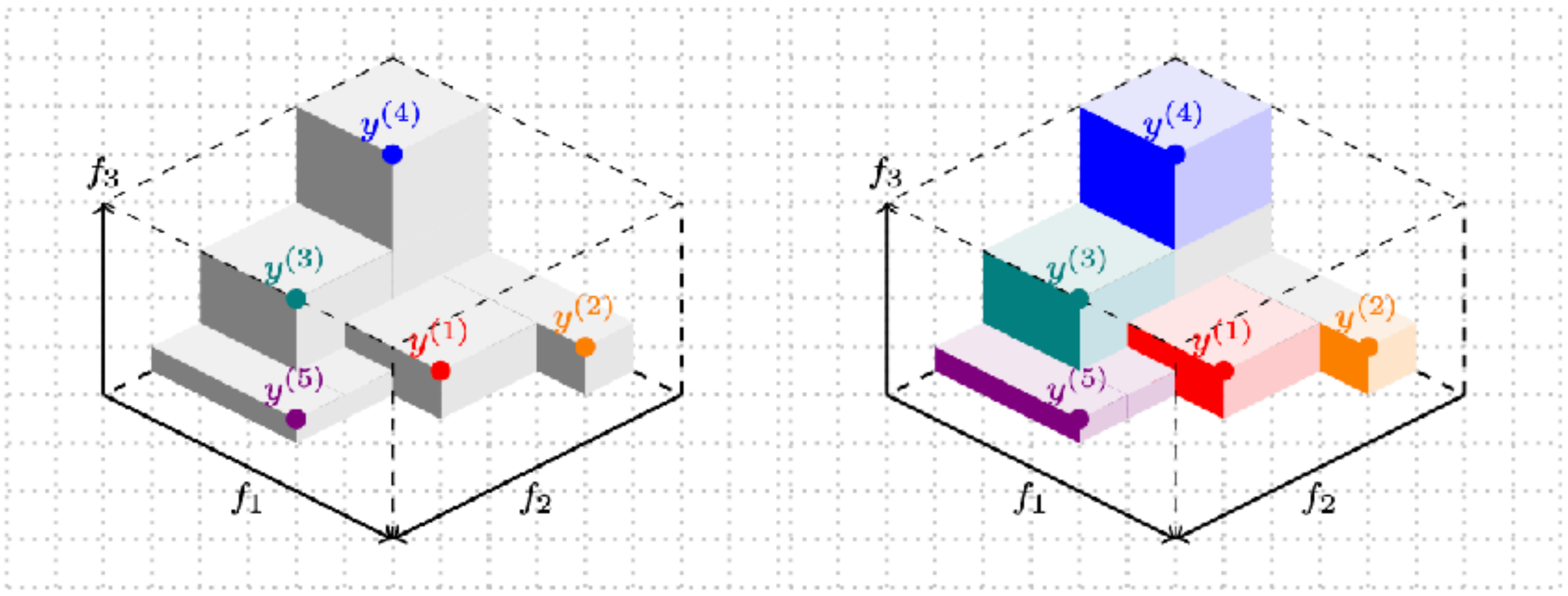
Measuring multiobjective performance

- Define a reference point $r \in \mathbb{R}^d$

A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]



2D



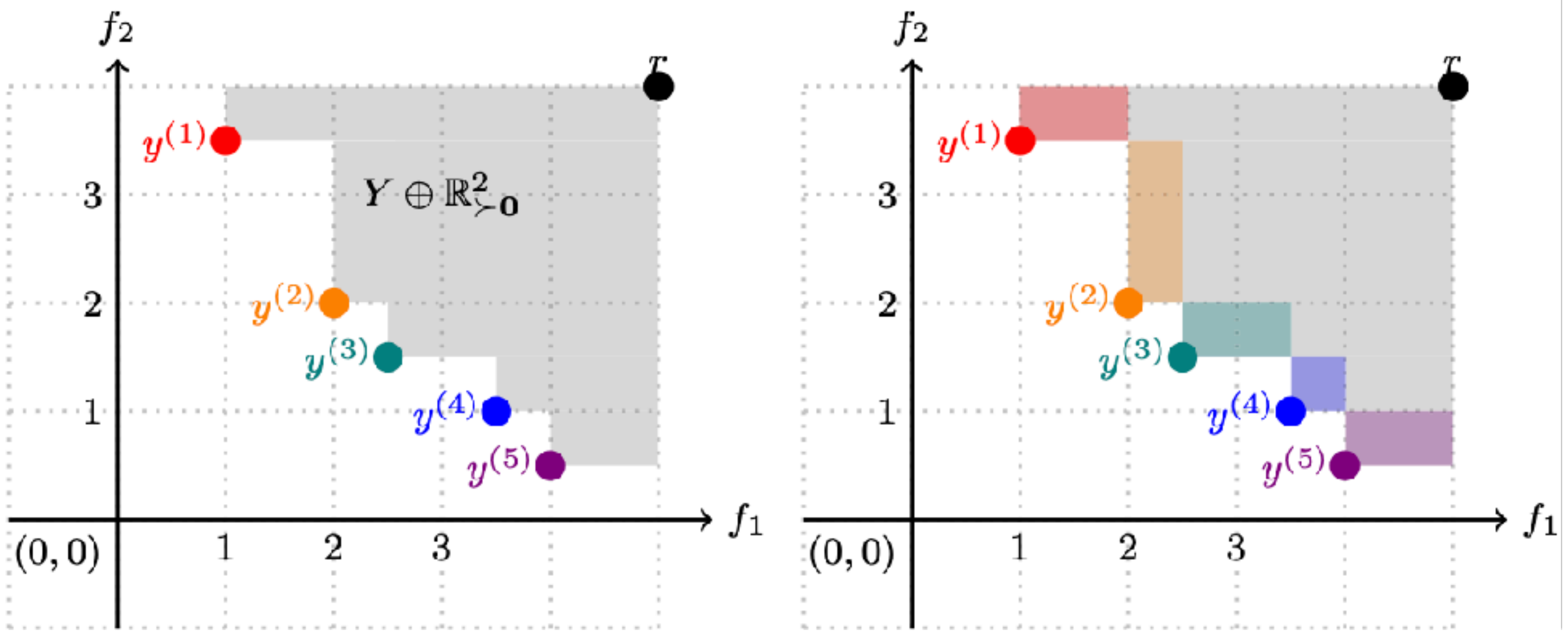
3D

Hypervolume

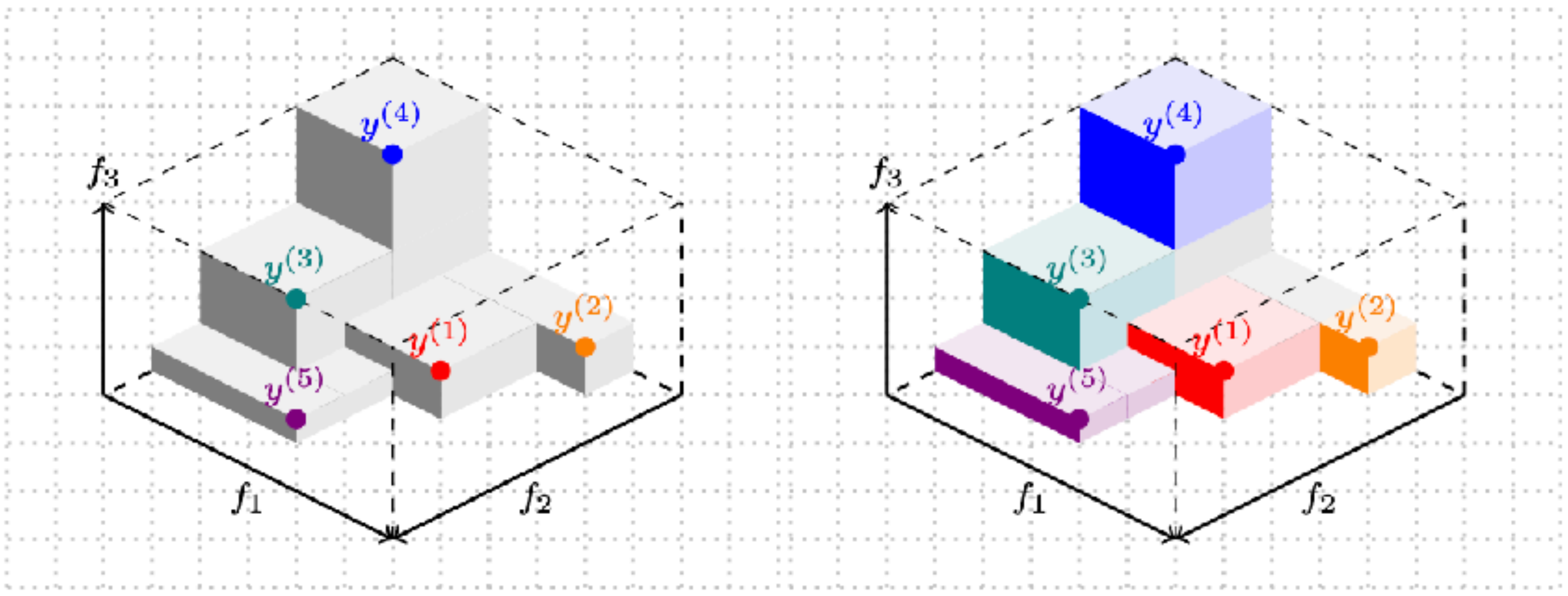
Measuring multiobjective performance

- Define a reference point $r \in \mathbb{R}^d$
- Hypervolume

A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]



2D



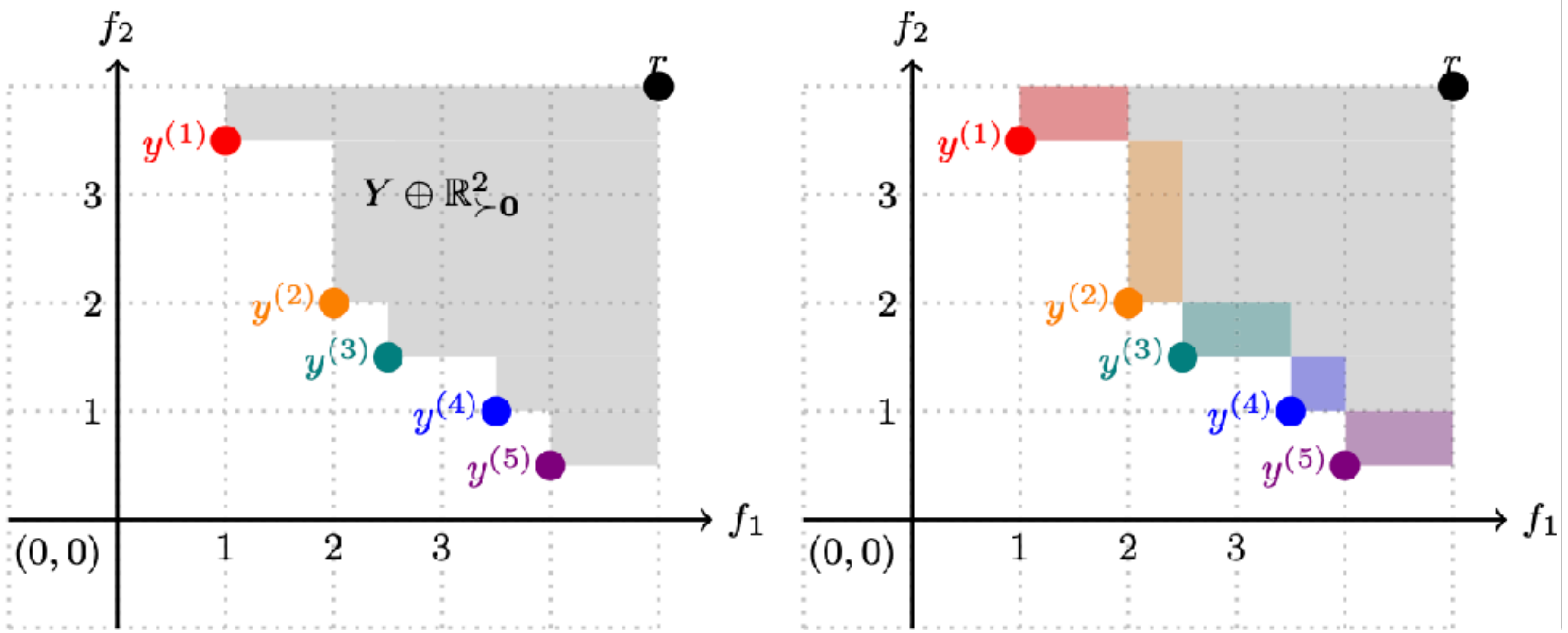
3D

Hypervolume

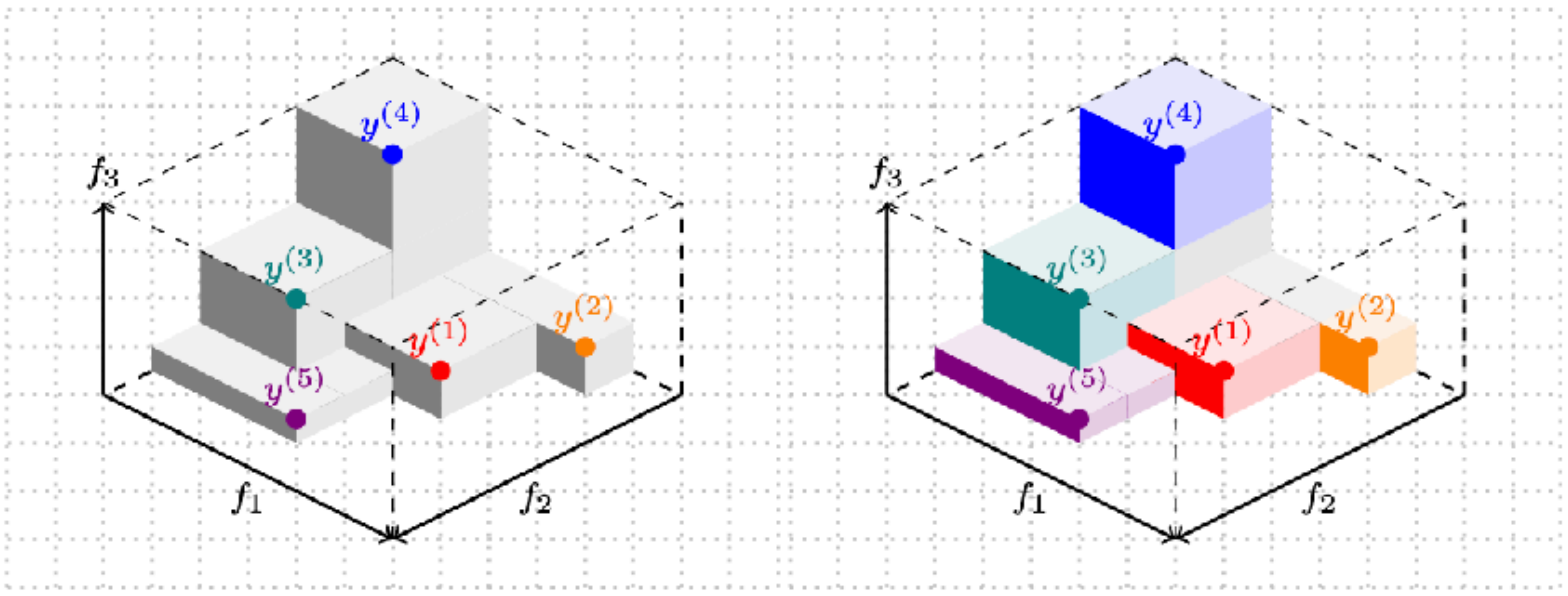
Measuring multiobjective performance

- Define a reference point $r \in \mathbb{R}^d$
- Hypervolume
  - $\mathrm{HV}_r(y)$ = “volume between r and the Pareto front $\mathscr{P}(y) \subset \mathbb{R}^d$”

A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]



2D



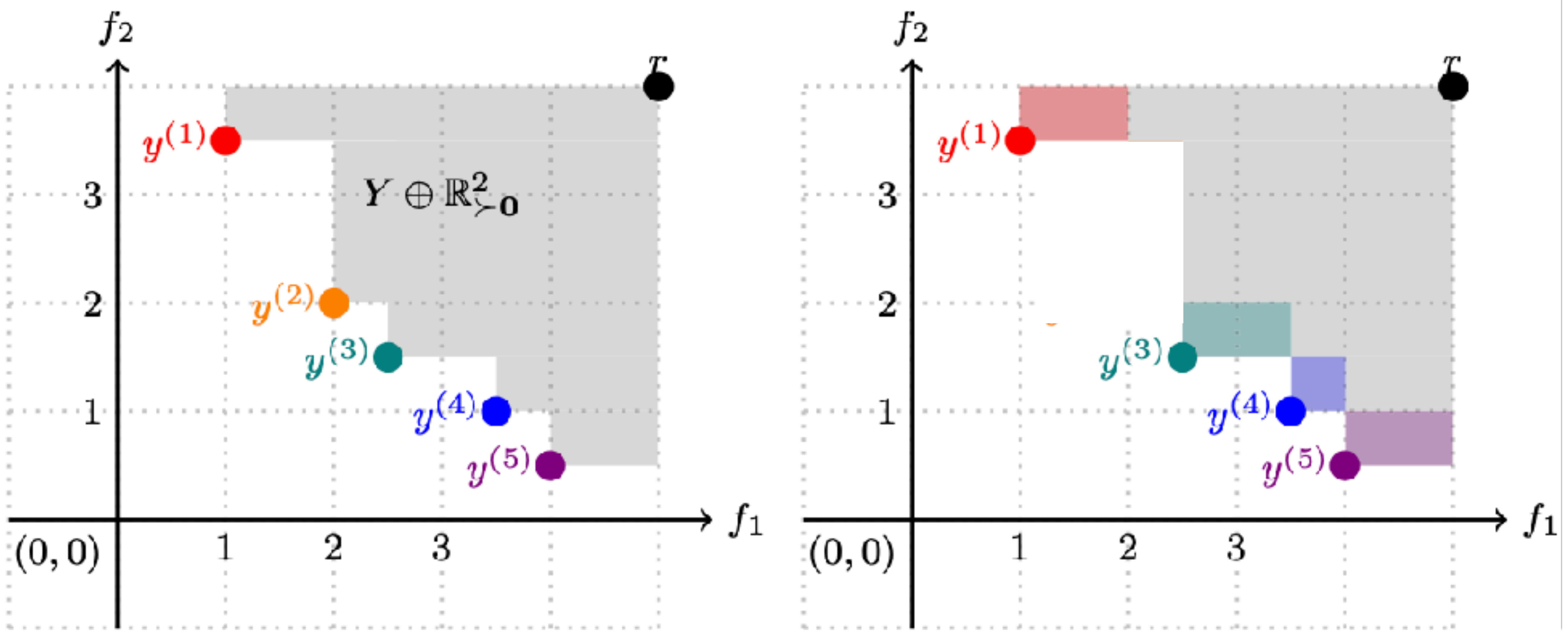
3D

Hypervolume

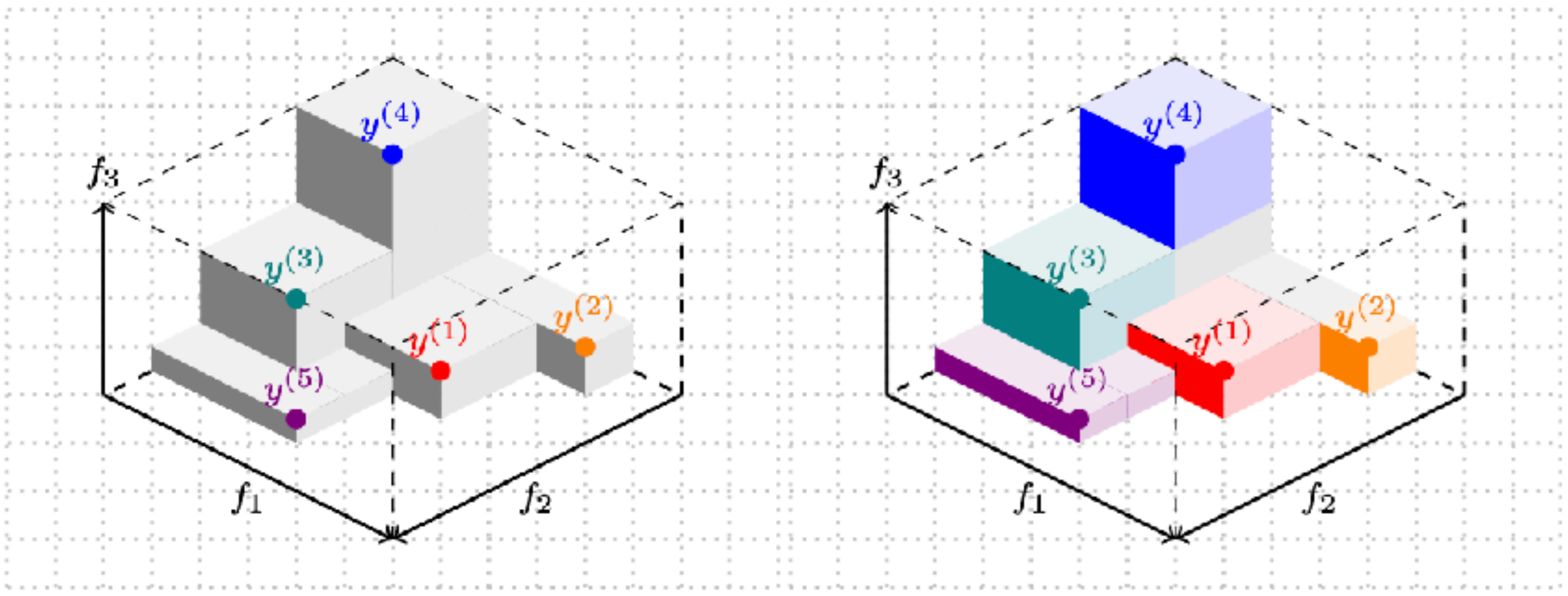
Measuring multiobjective performance

- Define a reference point $r \in \mathbb{R}^d$
- Hypervolume
    - $\mathrm{HV}_r(y)$ = “volume between r and the Pareto front $\mathcal{P}(y) \subset \mathbb{R}^d$”

A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]



2D



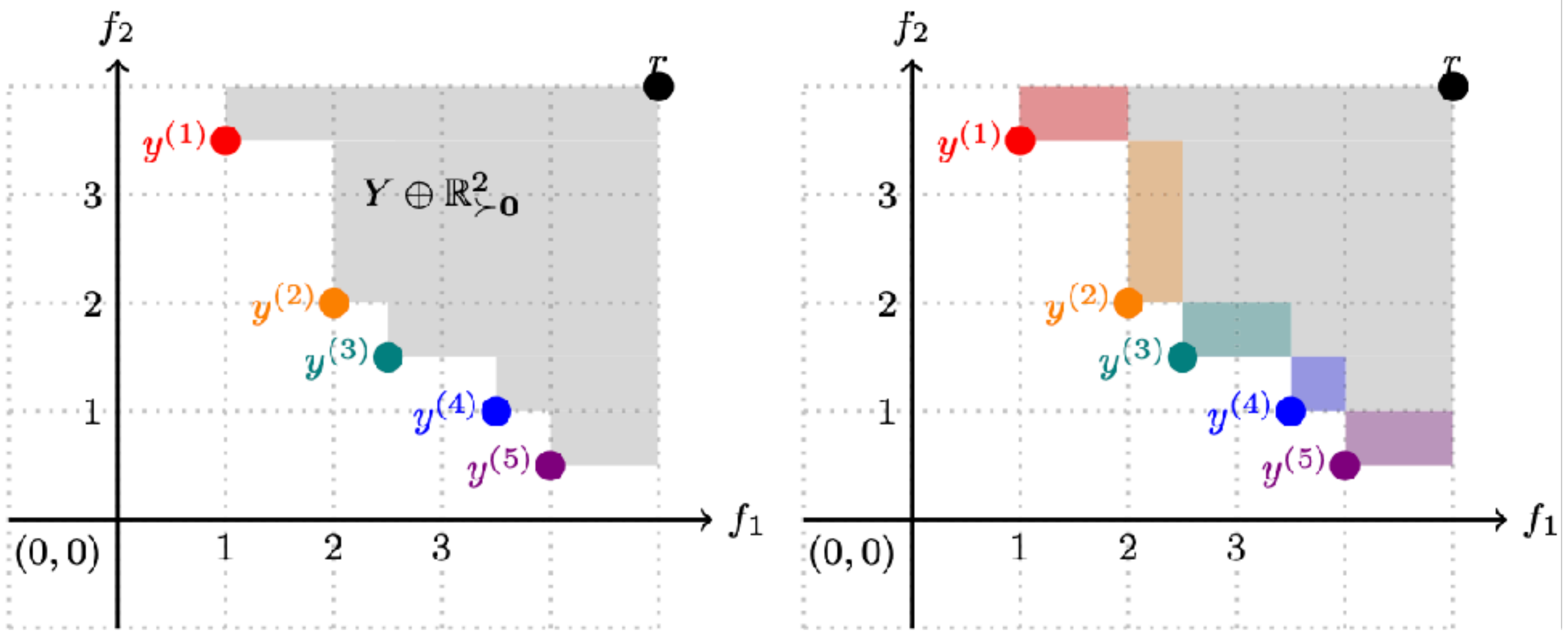
3D

Hypervolume

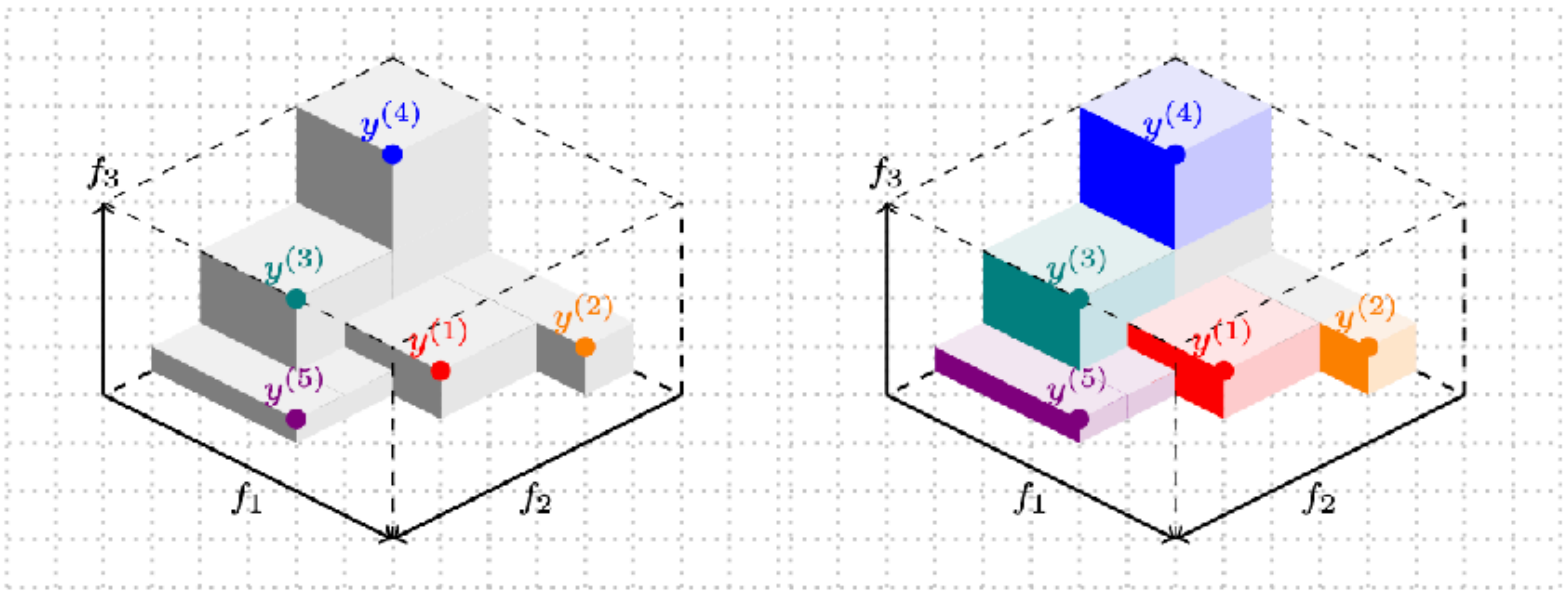
Measuring multiobjective performance

- Define a reference point $r \in \mathbb{R}^d$
- Hypervolume
  - $\mathrm{HV}_r(y)$ = "volume between r and the Pareto front $\mathscr{P}(y) \subset \mathbb{R}^d$"

A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]



2D



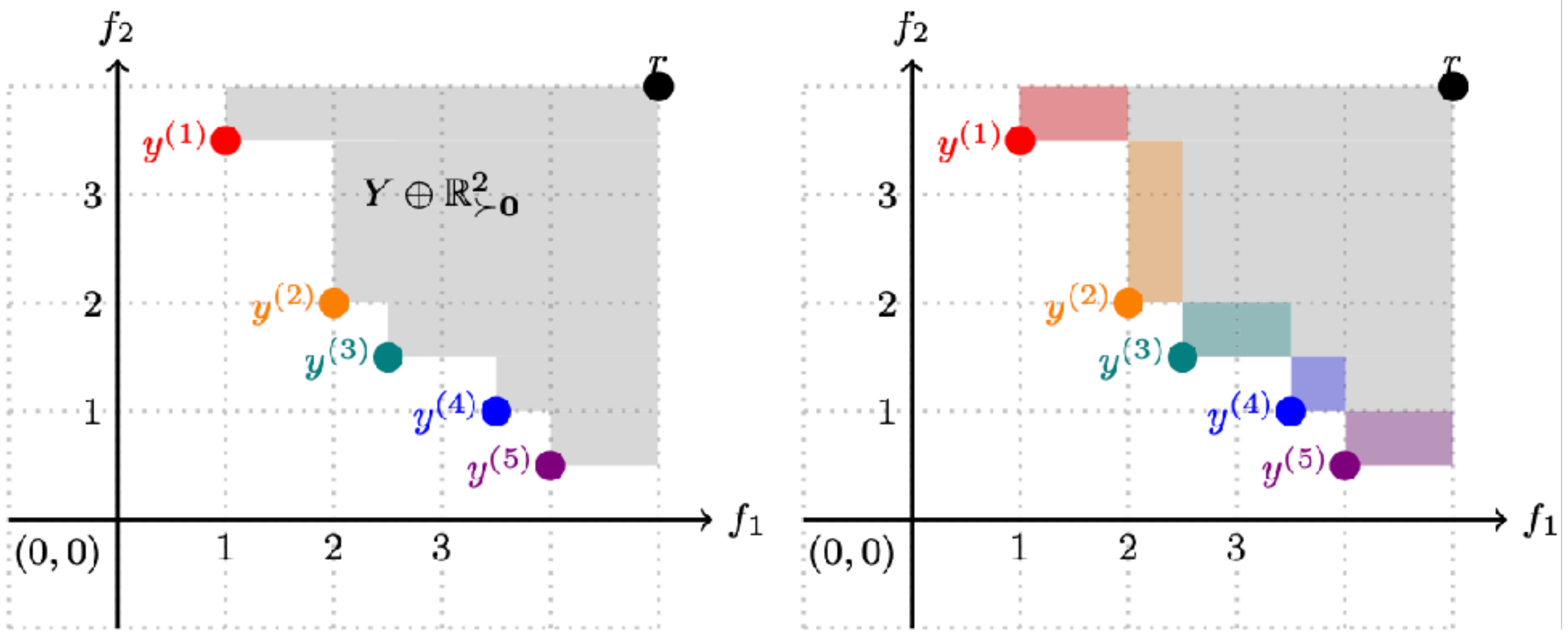
3D

Hypervolume

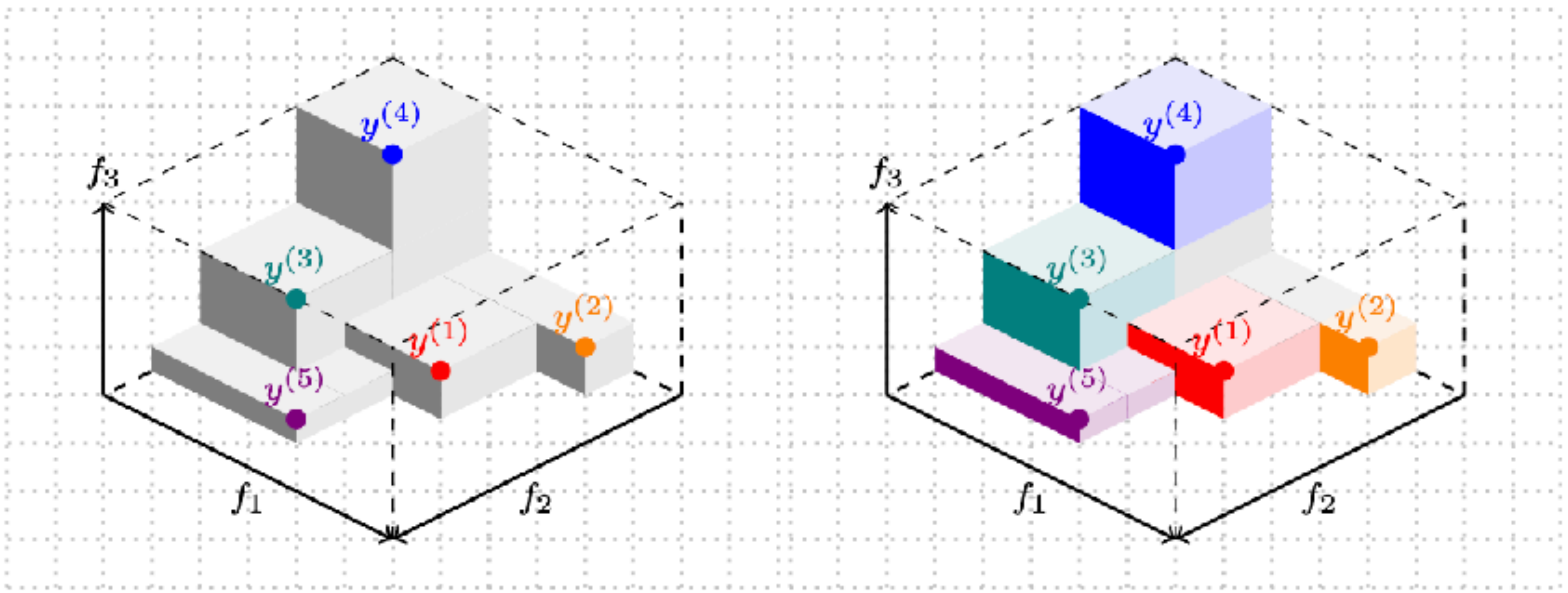
Measuring multiobjective performance

- Define a reference point $r \in \mathbb{R}^d$
- Hypervolume
  - $\text{HV}_r(y)$ = "volume between r and the Pareto front $\mathscr{P}(y) \subset \mathbb{R}^d$"
  - $\text{HV}_r(y) = \mathscr{V}(\{q \in \mathbb{R}^d \mid \exists p \in y, p \preceq q \text{ and } q \preceq r\})$

A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]



2D



3D

Multiobjective goal

Multiobjective goal

- Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front with as few samples as possible

Multiobjective goal

- Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front with as few samples as possible
- e.g. optimize $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

Multiobjective goal

- Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front with as few samples as possible
- e.g. optimize $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$
- The reference point is often chosen to be $1_d \in \mathbb{R}^d$ after having normalised the objectives in $[0,1]$

Multiobjective goal

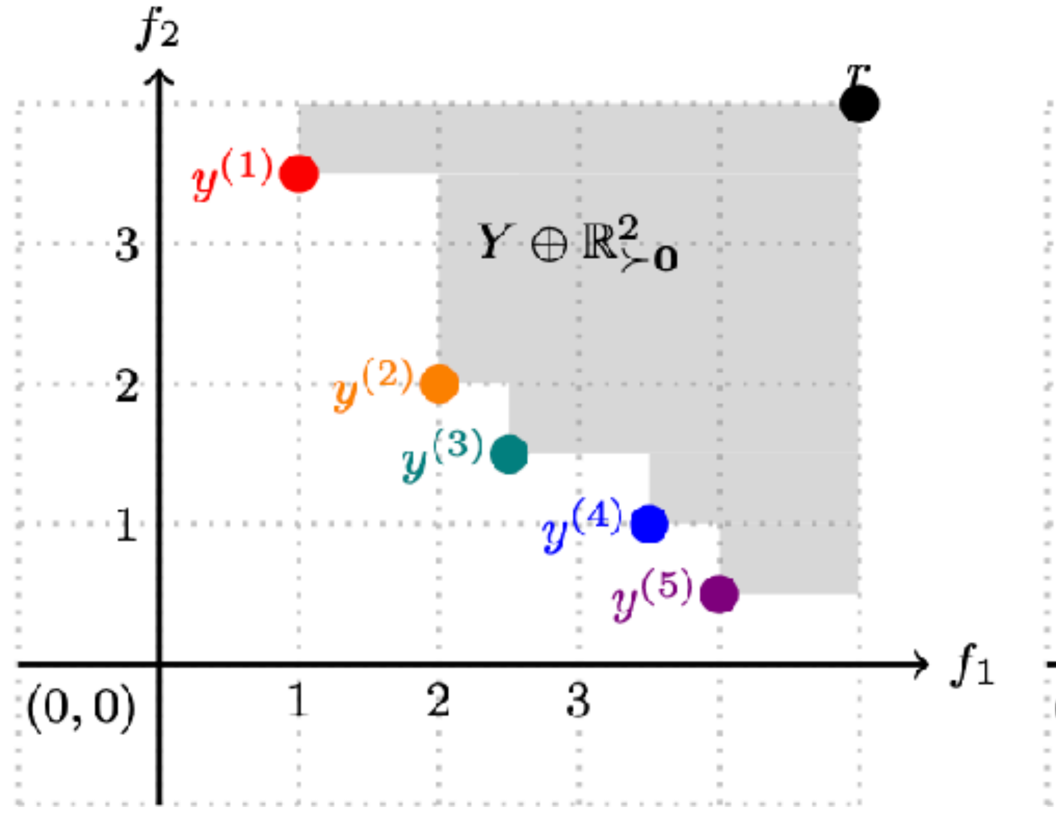
- Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front with as few samples as possible
- e.g. optimize $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$
- The reference point is often chosen to be $1_d \in \mathbb{R}^d$ after having normalised the objectives in $[0,1]$
- Sometimes we know the Pareto front (or we estimate after having run many optimizers), we can then instead say that we *minimize* the *hypervolume error*:

Multiobjective goal

- Given an expensive blackbox function $y = f(x) \in \mathbb{R}^d$ with d objectives, approximate the Pareto front with as few samples as possible
- e.g. optimize $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$
- The reference point is often chosen to be $1_d \in \mathbb{R}^d$ after having normalised the objectives in $[0,1]$
- Sometimes we know the Pareto front (or we estimate after having run many optimizers), we can then instead say that we *minimize* the *hypervolume error*:

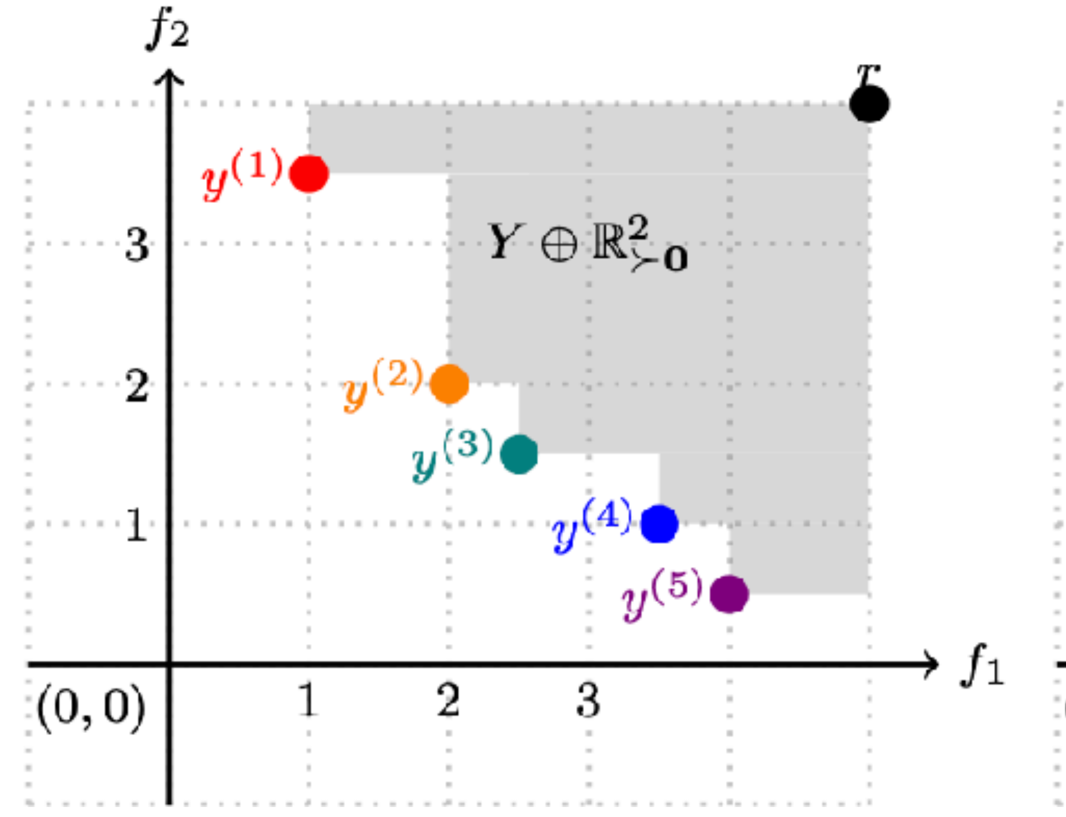
$$\text{HV}(\mathscr{P}(f)) - \text{HV}(\{f(x_1), \ldots, f(x_n)\})$$

Objective scaling transformation



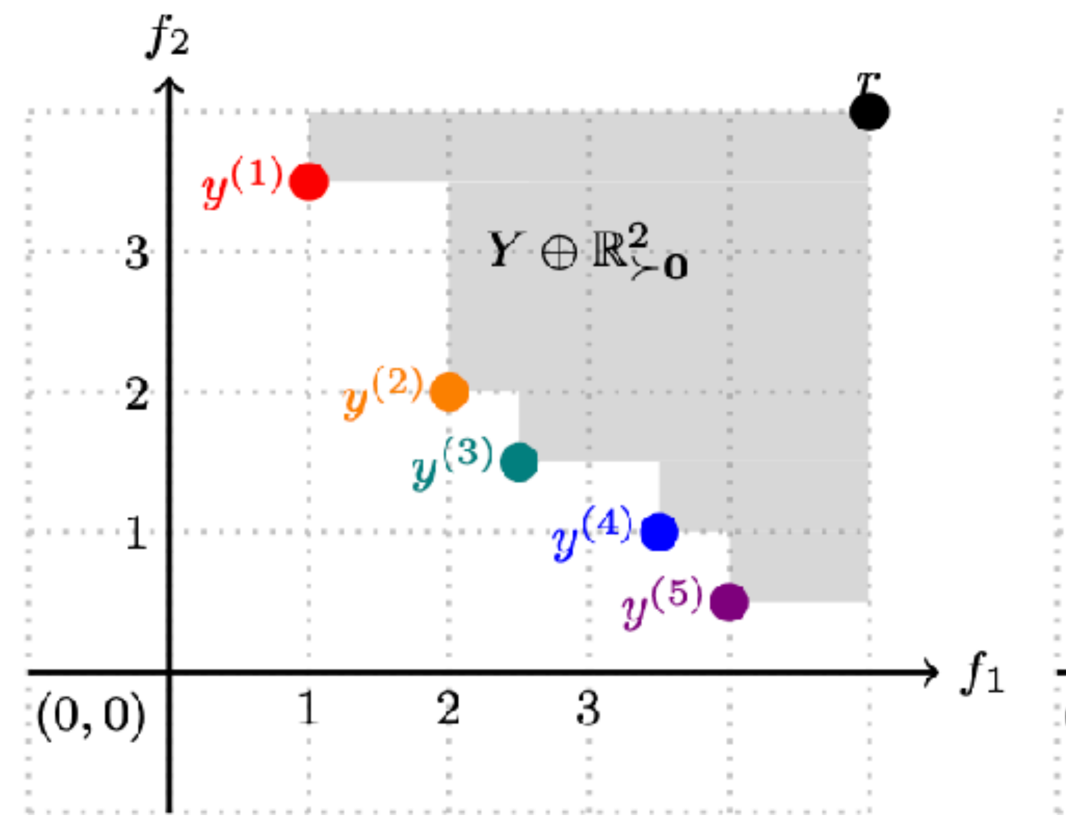
Objective scaling transformation

- We typically want objectives to be comparable and at a similar scale



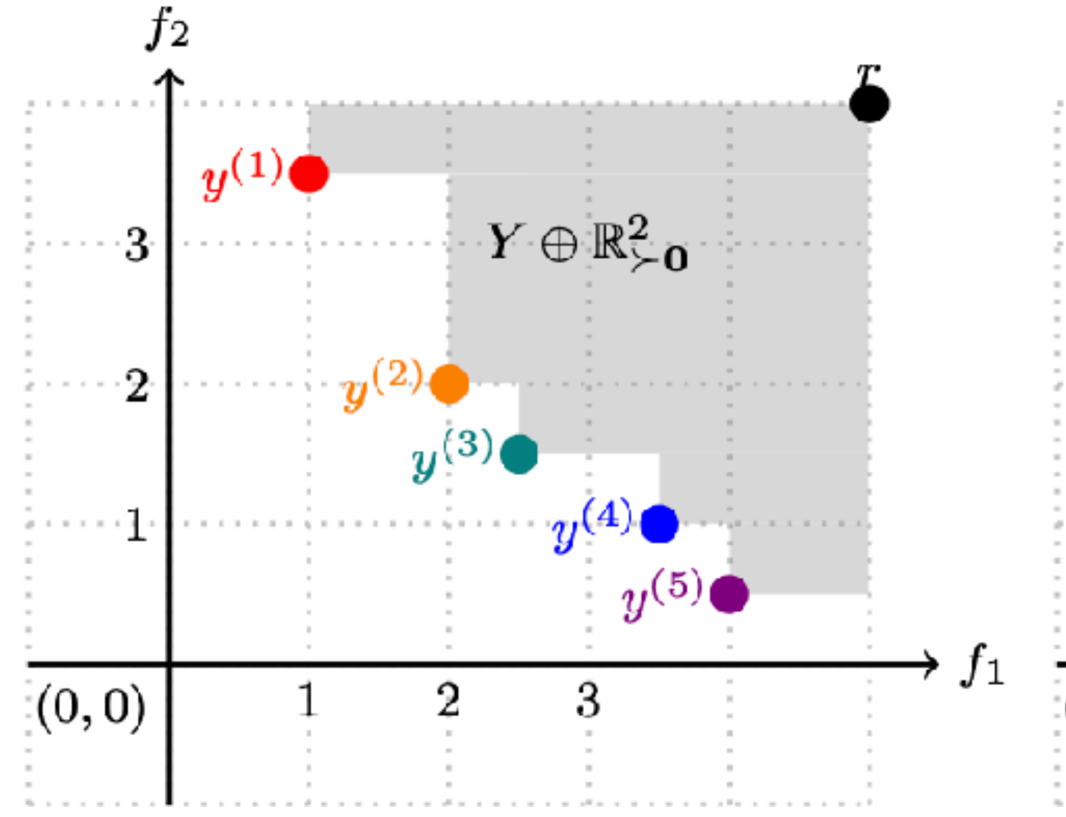
Objective scaling transformation

- We typically want objectives to be comparable and at a similar scale
- Several normalisation are often used for the objectives $y \in \mathbb{R}^{n \times d}$:



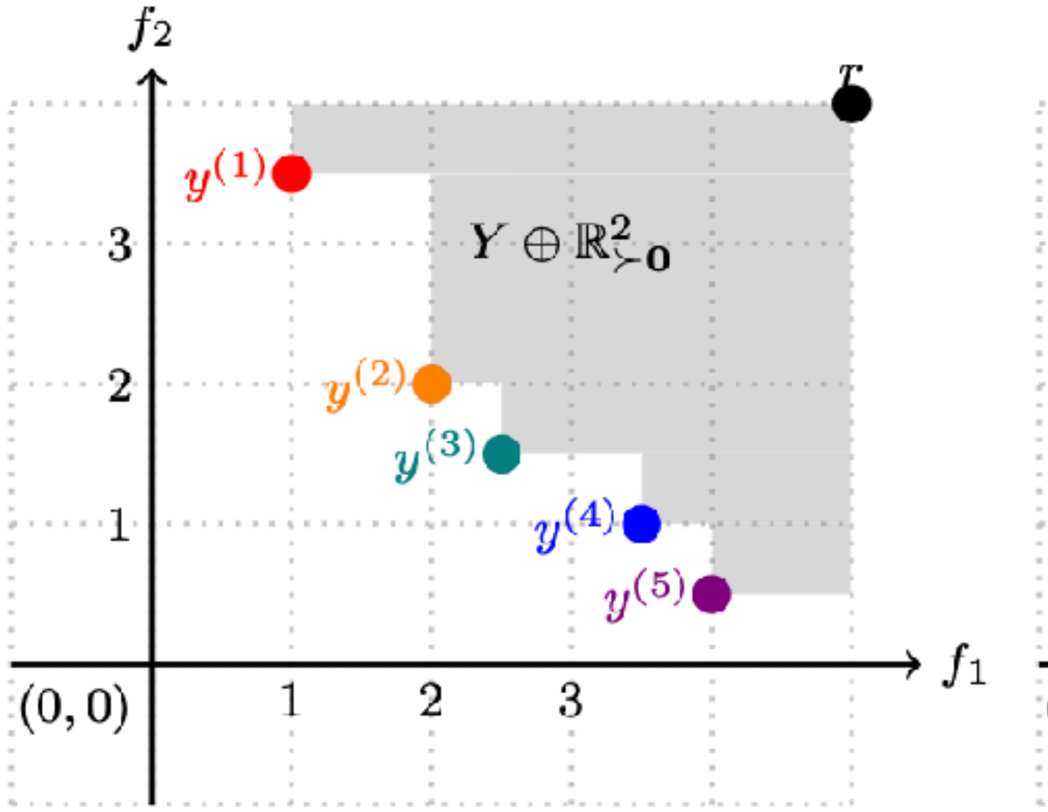
Objective scaling transformation

- We typically want objectives to be comparable and at a similar scale
- Several normalisation are often used for the objectives $y \in \mathbb{R}^{n \times d}$:
  - Min-max normalisation: $z = \dfrac{y - min(y)}{max(y) - min(y)} \in [0,1]$



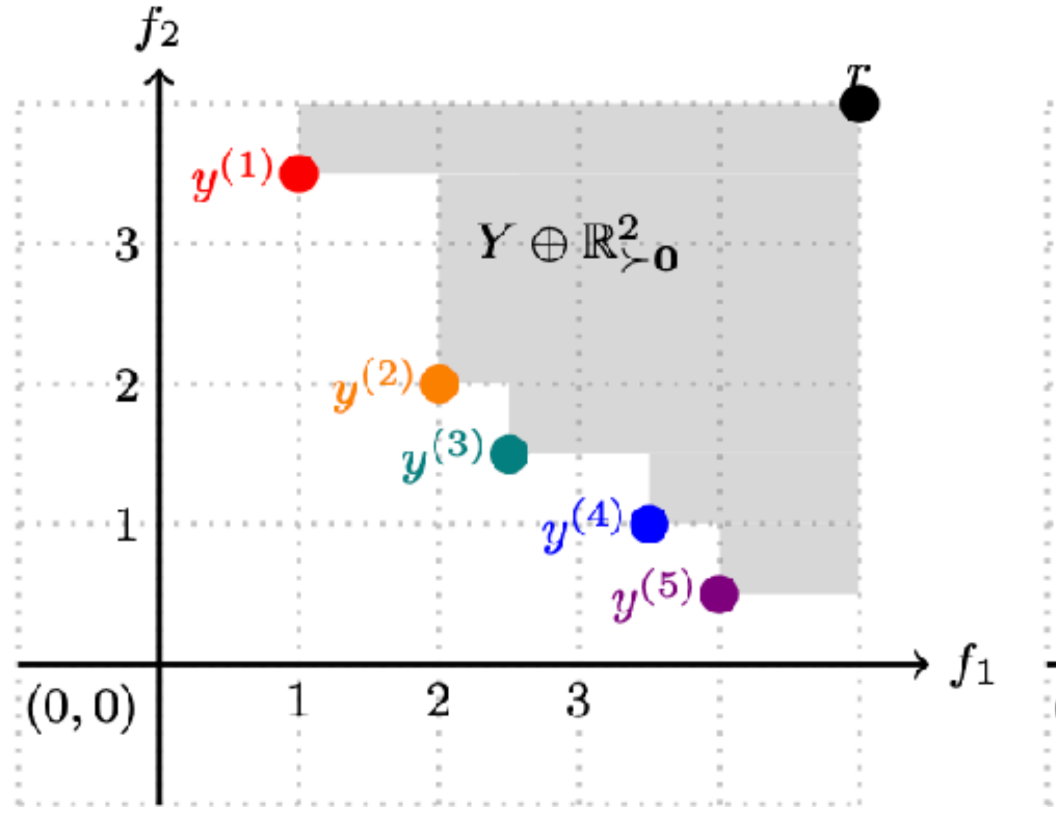
Objective scaling transformation

- We typically want objectives to be comparable and at a similar scale
- Several normalisation are often used for the objectives $y \in \mathbb{R}^{n \times d}$:
  - Min-max normalisation: $z = \dfrac{y - min(y)}{max(y) - min(y)} \in [0,1]$
  - CDF/rank/copula normalisation: $z = F_y(y)$ where F is the empirical CDF



Objective scaling transformation

- We typically want objectives to be comparable and at a similar scale
- Several normalisation are often used for the objectives $y \in \mathbb{R}^{n \times d}$:
  - Min-max normalisation: $z = \dfrac{y - min(y)}{max(y) - min(y)} \in [0,1]$
  - CDF/rank/copula normalisation: $z = \mathrm{F}_y(y)$ where F is the empirical CDF



This transformation makes the comparisons invariant through any monotonic change! Neat 👍 [Binois 2020]

Optimization Methods

Optimization Methods

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations
  - $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations
  - $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

🤔 **Which method could you think of?**

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations
  - $(x_1, \ldots, x_n) = \text{argmax}_{(x_1, \ldots, x_n) \in \mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

🤔 **Which method could you think of?**

- Main approaches:

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations
  - $(x_1, \ldots, x_n) = \text{argmax}_{(x_1, \ldots, x_n) \in \mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

🤔 **Which method could you think of?**

- Main approaches:
  - Scalarization: scalarize objectives and apply single objective method

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations
    - $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

🤔 **Which method could you think of?**

- Main approaches:
    - Scalarization: scalarize objectives and apply single objective method
    - Evolutionary Algorithms

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations
  - $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

🤔 **Which method could you think of?**

- Main approaches:
  - Scalarization: scalarize objectives and apply single objective method
  - Evolutionary Algorithms
  - Bayesian Optimization

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations
  - $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

🤔 **Which method could you think of?**

- Main approaches:
  - Scalarization: scalarize objectives and apply single objective method
  - Evolutionary Algorithms
  - Bayesian Optimization
  - Preference based

Optimization Methods

- Recall that we want to maximize the Hypervolume under a minimum amount of function evaluations
  - $(x_1, \ldots, x_n) = \text{argmax}_{(x_1,\ldots,x_n)\in\mathbb{R}^m} \text{HV}(\{f(x_1), \ldots, f(x_n)\})$

🤔 **Which method could you think of?**

- Main approaches:
  - Scalarization: scalarize objectives and apply single objective method
  - Evolutionary Algorithms
  - Bayesian Optimization
  - Preference based
  - Others: Population Based Training, Multi-fidelity

Optimization Methods

Scalarization

Optimization Methods

Scalarization

- Recall that $f(x) \in \mathbb{R}^d$ is our function to optimize

Optimization Methods

Scalarization

- Recall that $f(x) \in \mathbb{R}^d$ is our function to optimize
- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1\,\forall i$)

Optimization Methods

Scalarization

- Recall that $f(x) \in \mathbb{R}^d$ is our function to optimize
- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)
- 👍 We can optimize $g(x)$ with any single objective method!

Optimization Methods

Scalarization

- Recall that $f(x) \in \mathbb{R}^d$ is our function to optimize
- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)
- 👍 We can optimize $g(x)$ with any single objective method!
- 👎 Need to set the weights

Optimization Methods

Scalarization

- Recall that $f(x) \in \mathbb{R}^d$ is our function to optimize
- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)
- 👍 We can optimize $g(x)$ with any single objective method!
- 👎 Need to set the weights
- 👎 Impose some restrictions on the shape of the Pareto front

Optimization Methods

Scalarization

- Recall that $f(x) \in \mathbb{R}^d$ is our function to optimize
- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)
- 👍 We can optimize $g(x)$ with any single objective method!
- 👎 Need to set the weights
- 👎 Impose some restrictions on the shape of the Pareto front

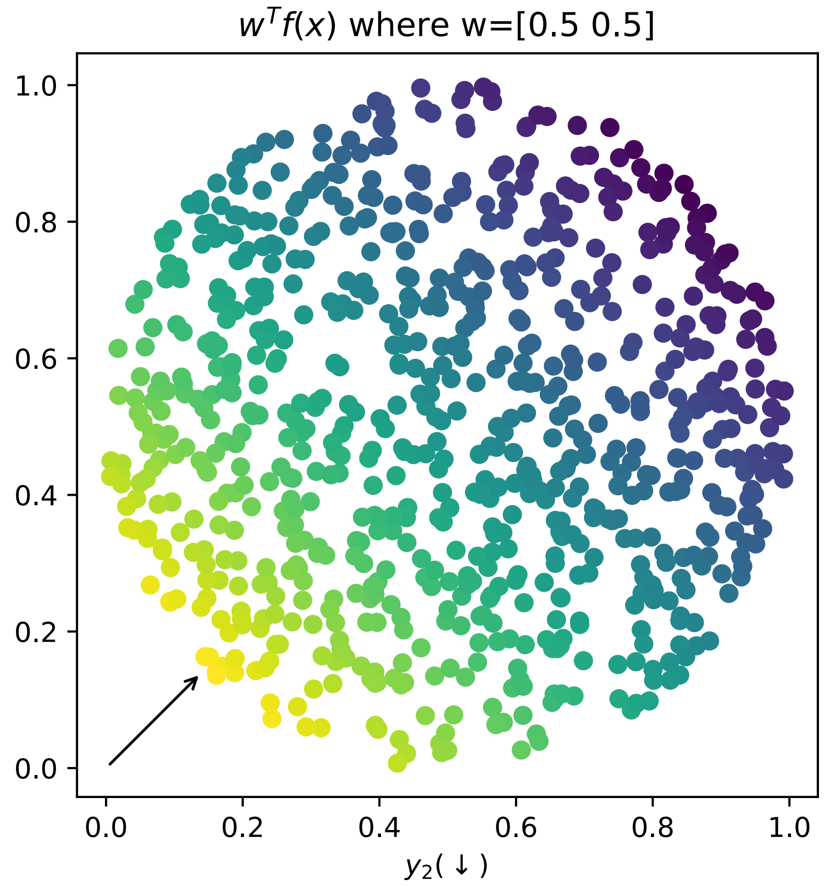
🤔 **Which one?**

Optimization Methods - scalarization

- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)

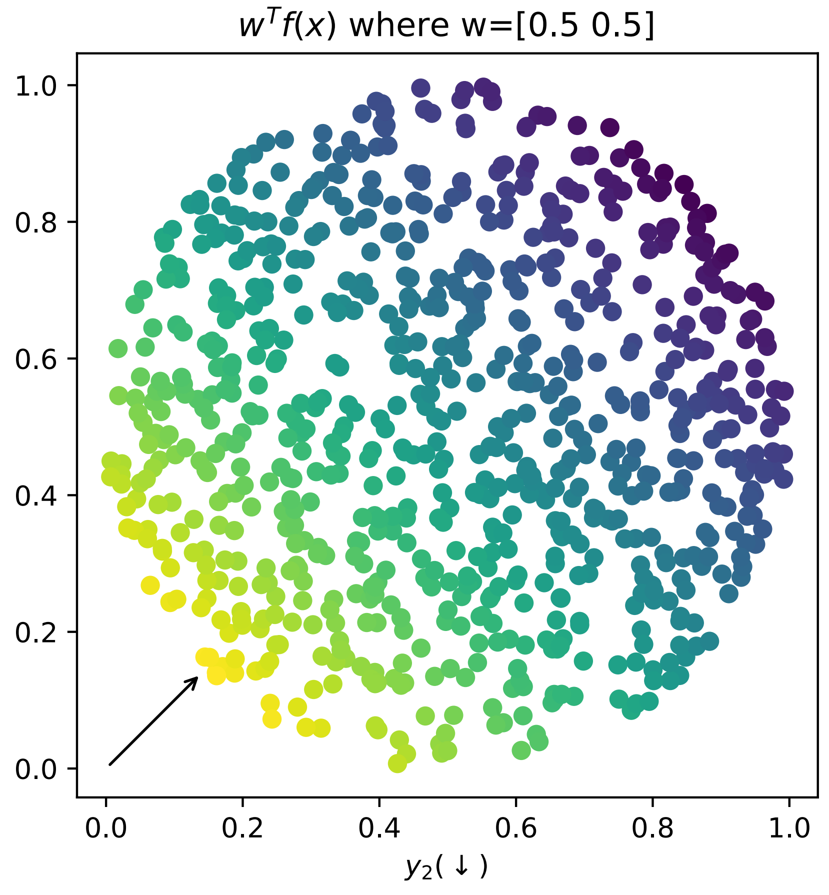
Optimization Methods - scalarization

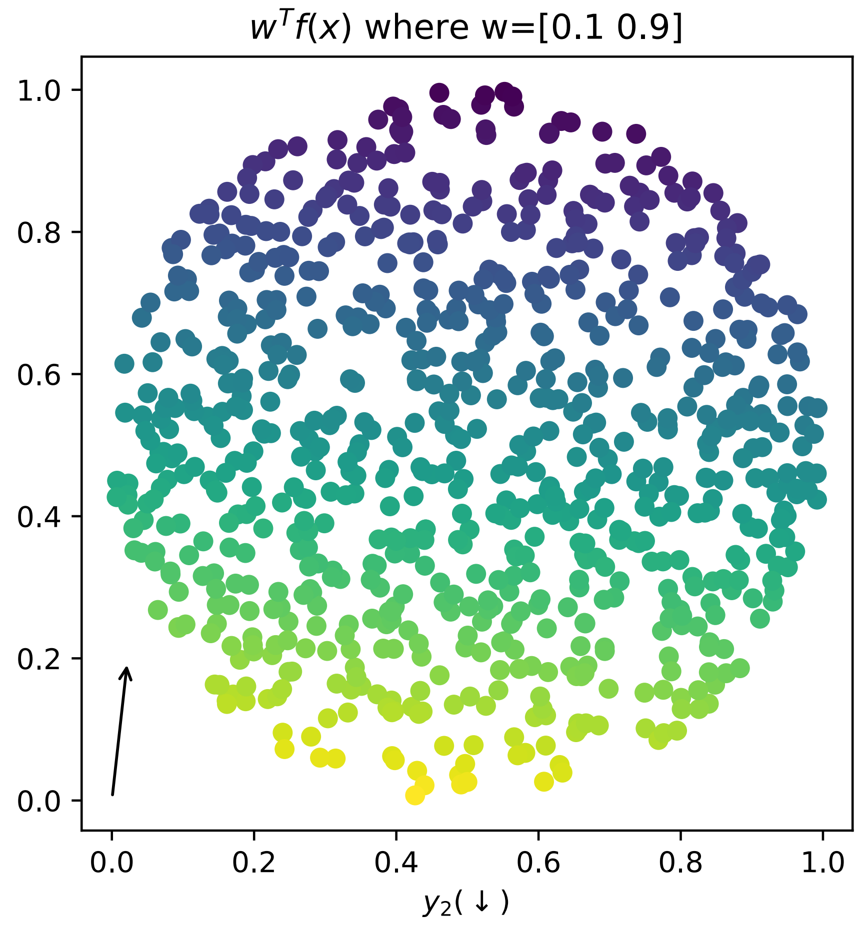
- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)



Optimization Methods - scalarization

- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)

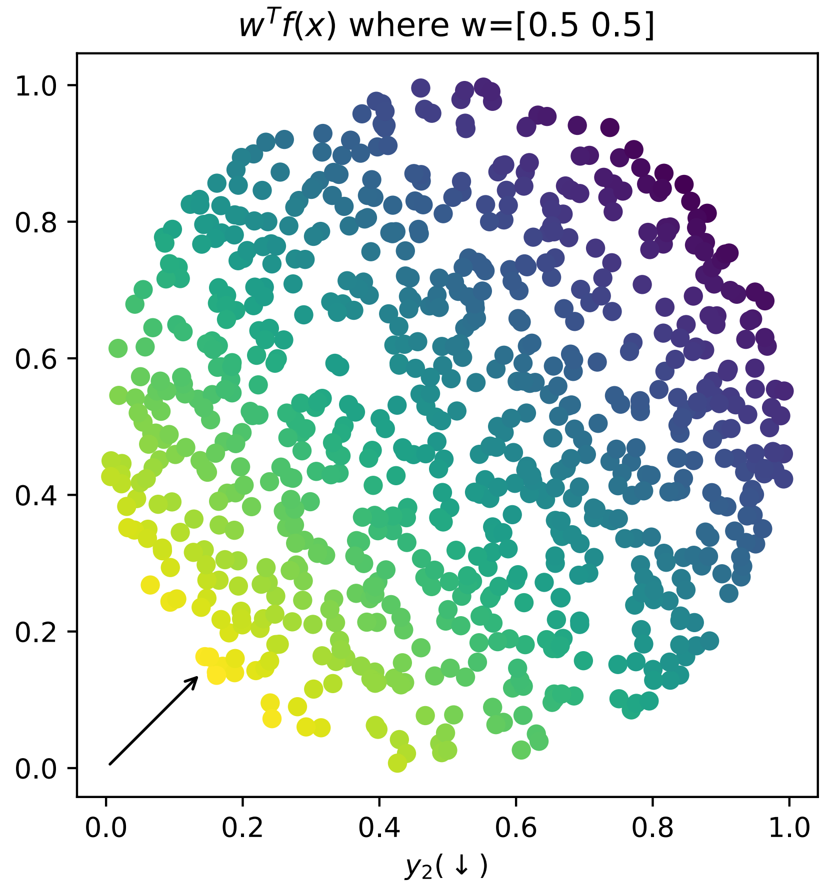


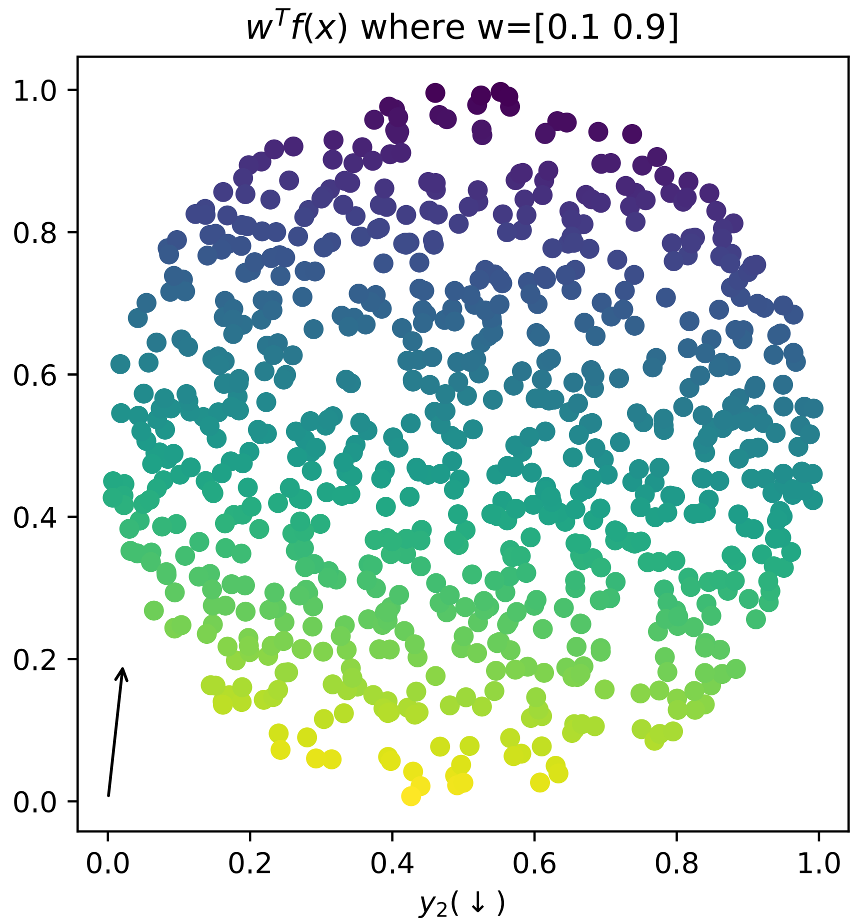


Optimization Methods - scalarization

- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)

We can vary the weights and obtain different part of the Pareto front

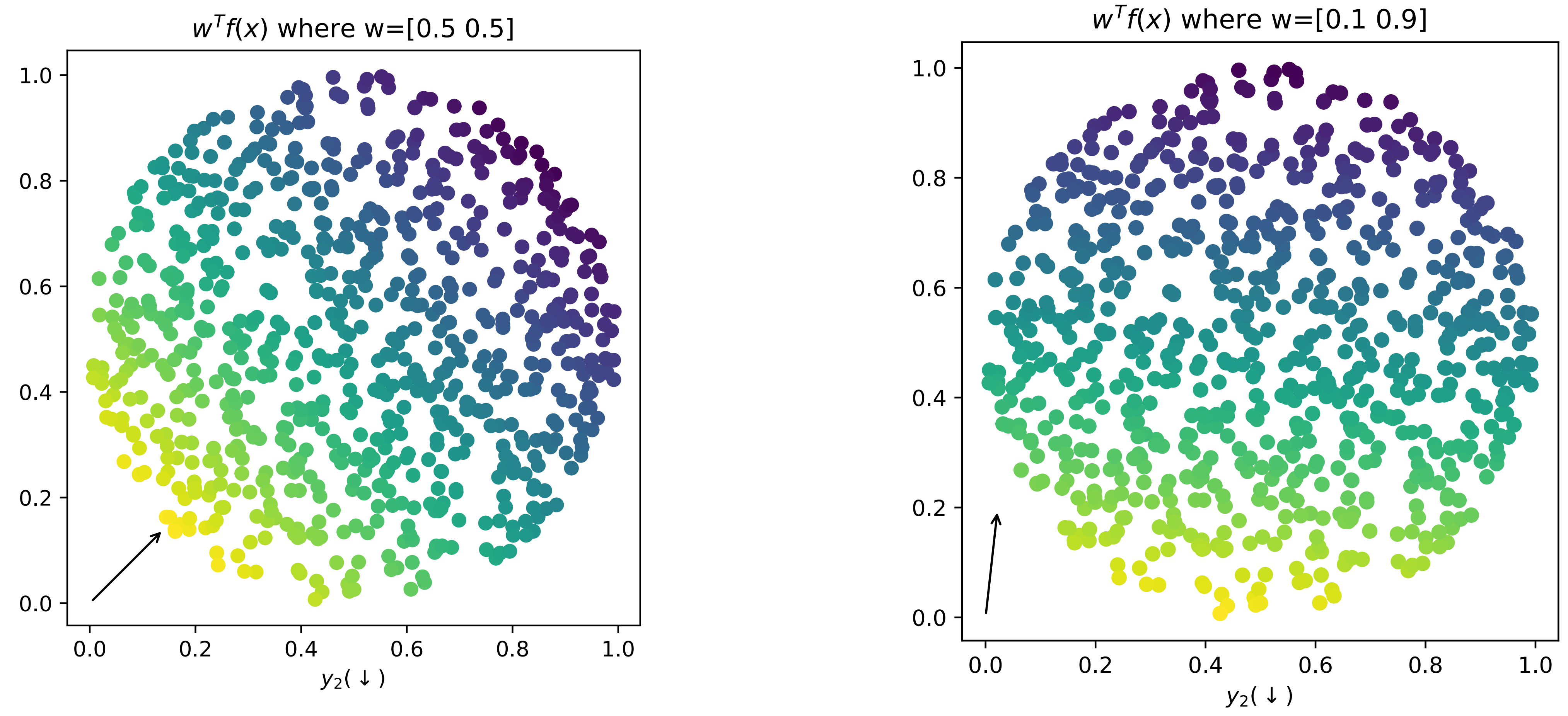




Optimization Methods - scalarization

- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)

We can vary the weights and obtain different part of the Pareto front

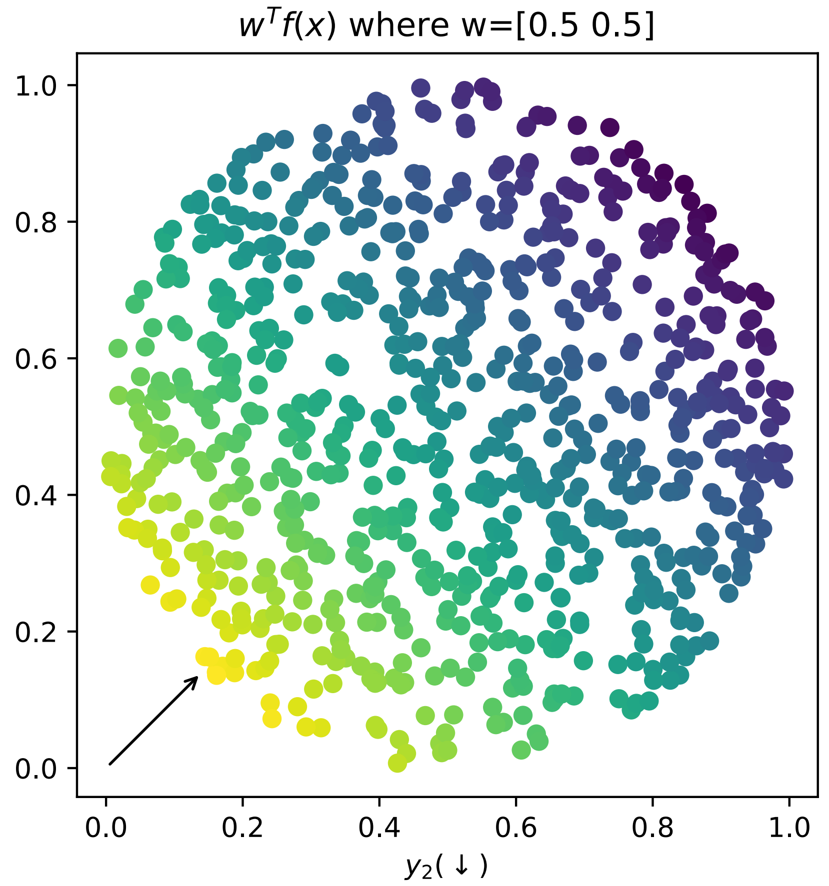


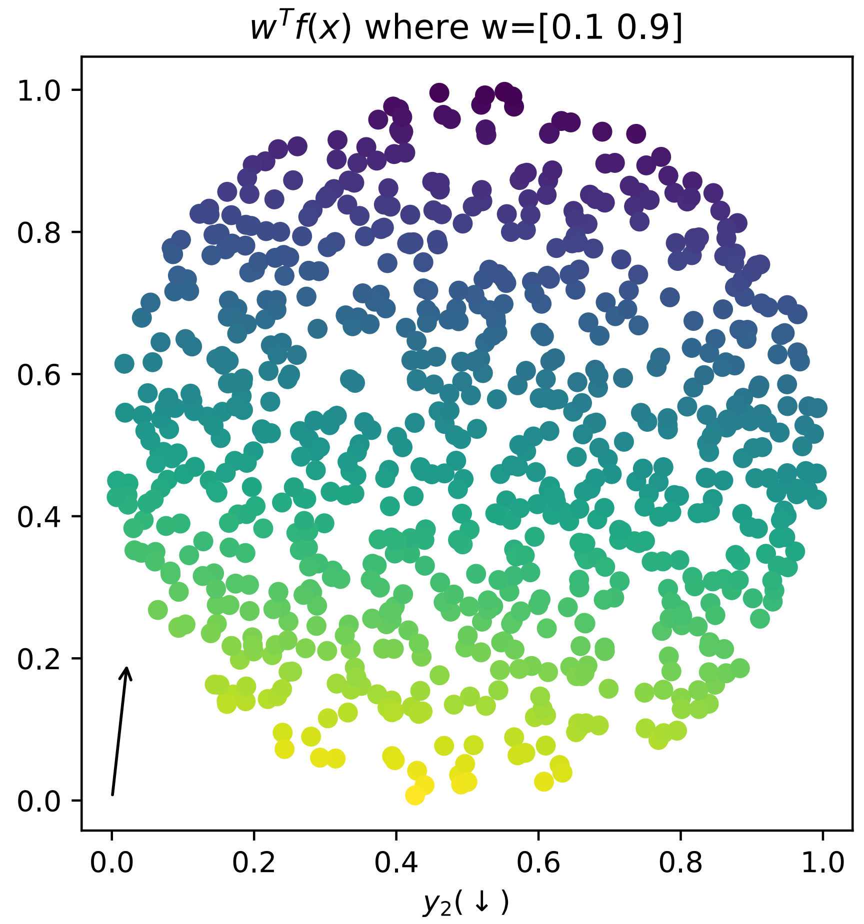
🤔 What is a limitation of this method?

Optimization Methods - scalarization

- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)

We can vary the weights and obtain different part of the Pareto front





🤔 What is a limitation of this method?

🙈 We can reach the whole Pareto front if its convex!

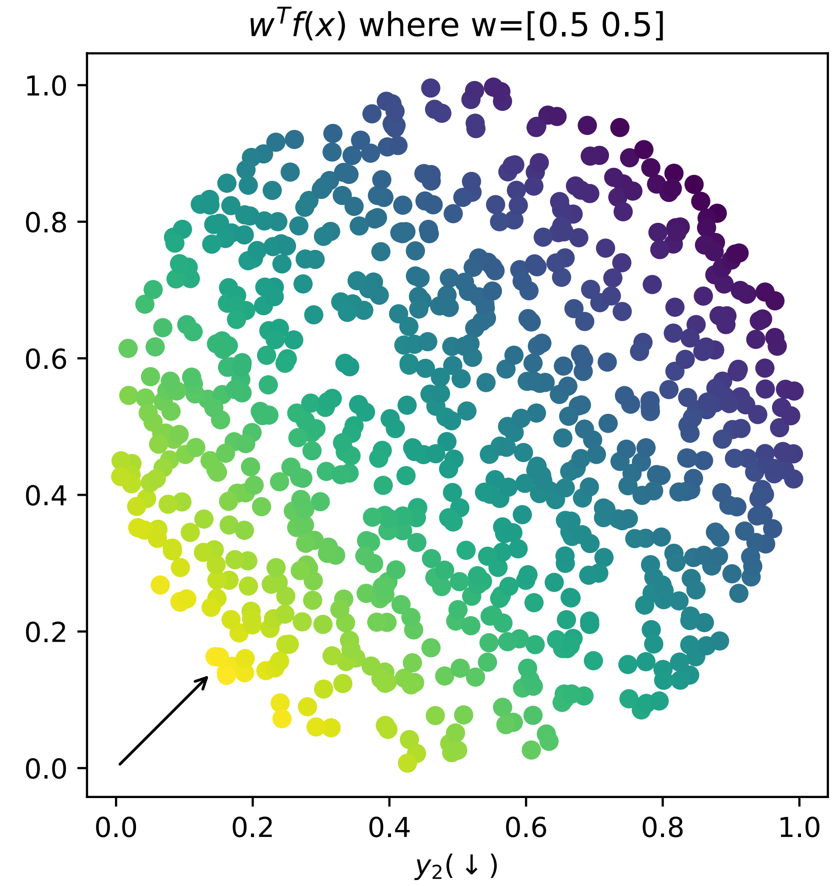
Optimization Methods - scalarization

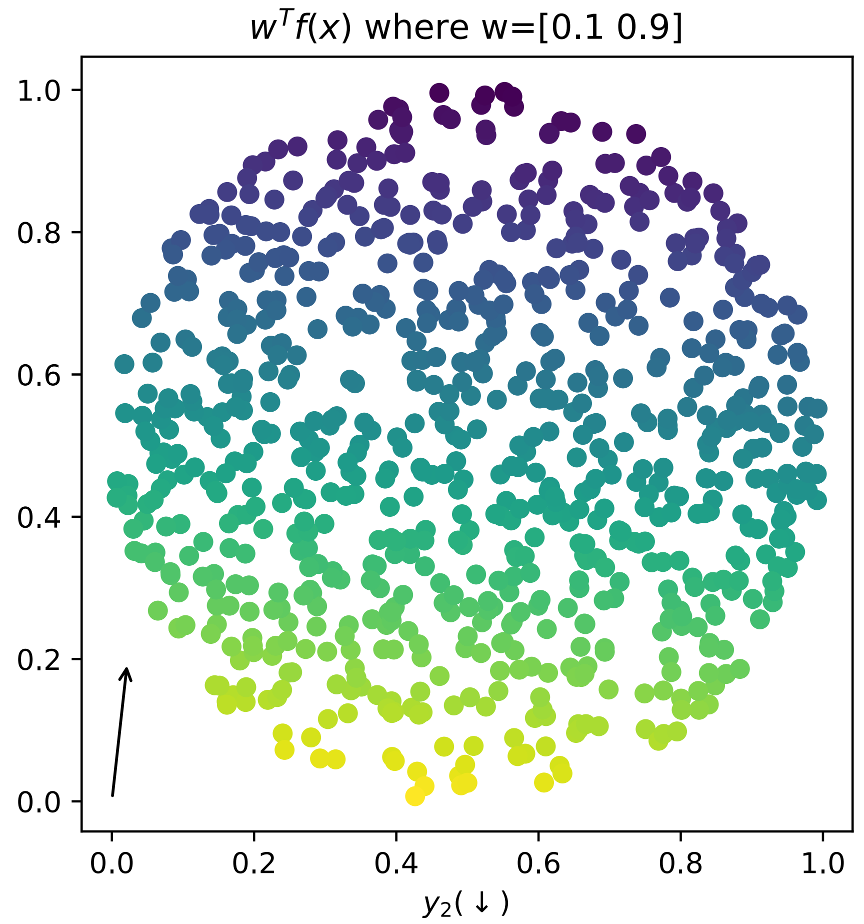
- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \,\forall i$)

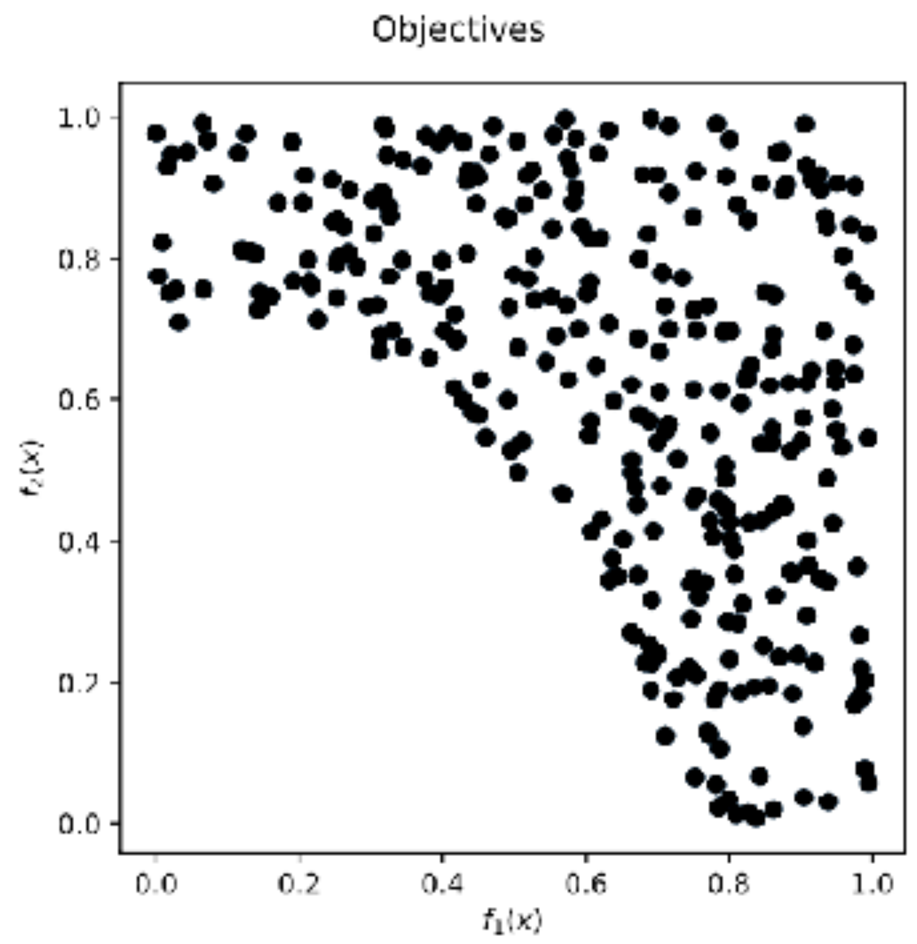
We can vary the weights and obtain different part of the Pareto front

🤔 What is a limitation of this method?

🙈 We can reach the whole Pareto front if its convex!







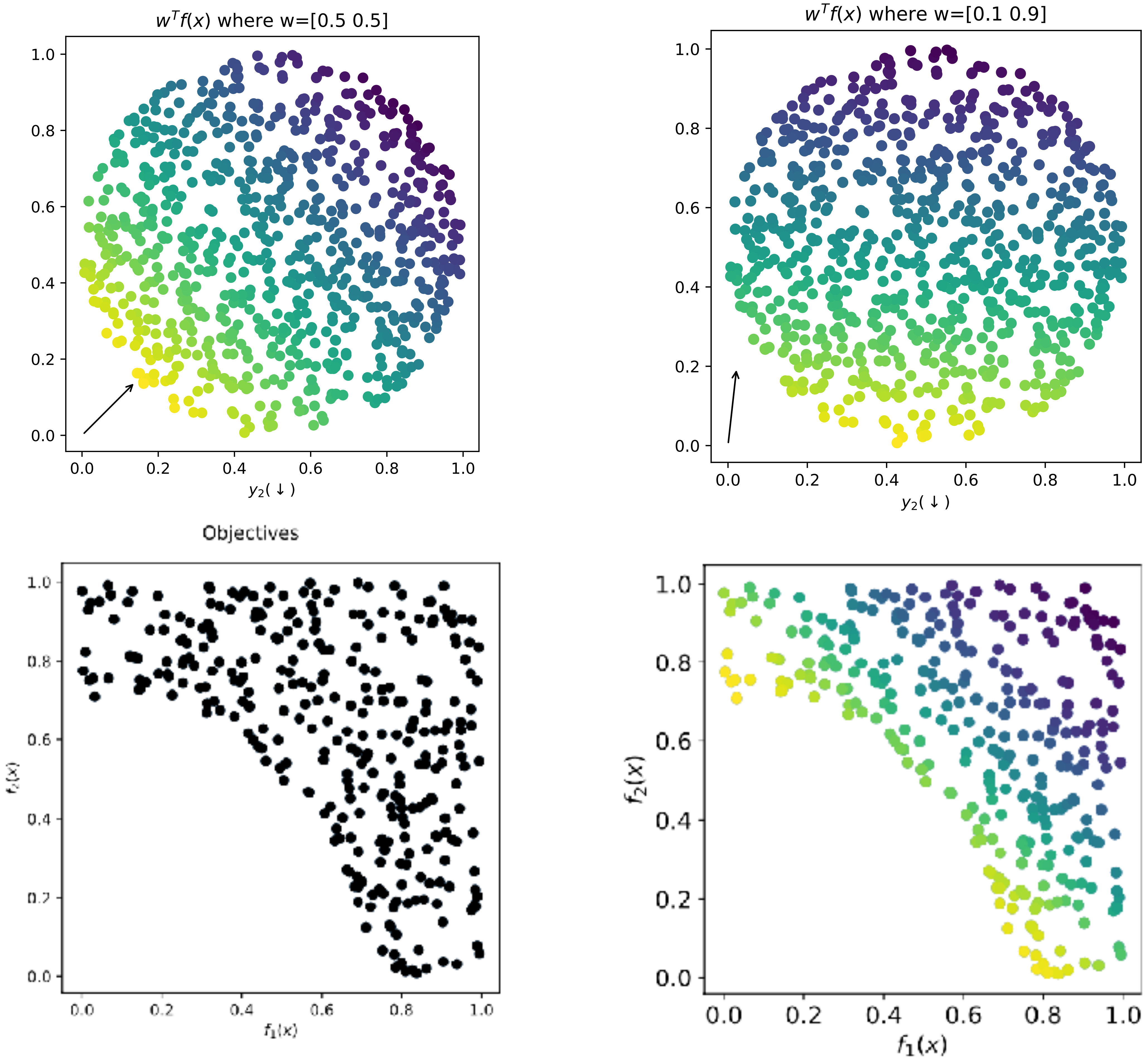
Optimization Methods - scalarization

- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)

We can vary the weights and obtain different part of the Pareto front

🤔 What is a limitation of this method?

🙈 We can reach the whole Pareto front if its convex!



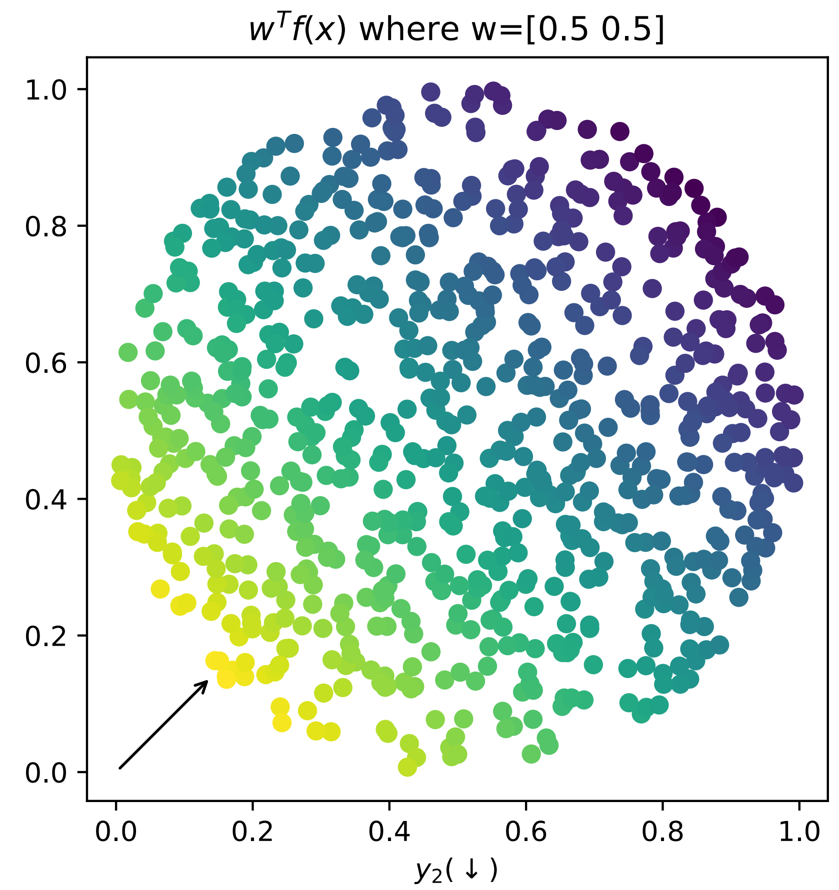
Optimization Methods - scalarization

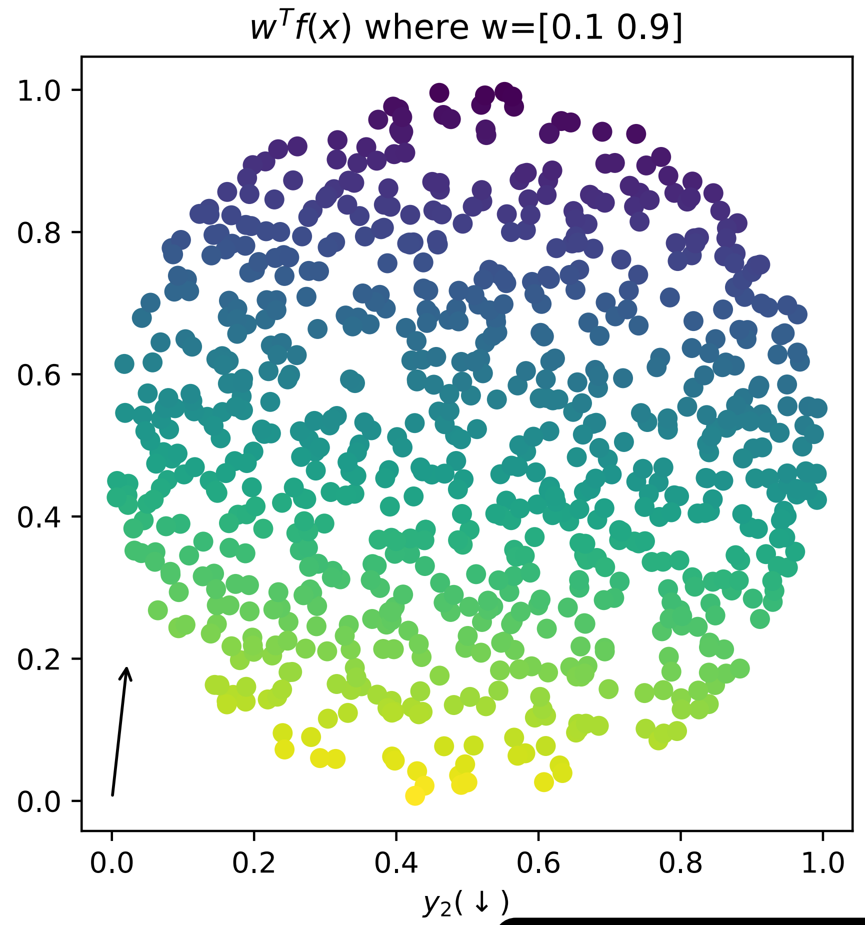
- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)

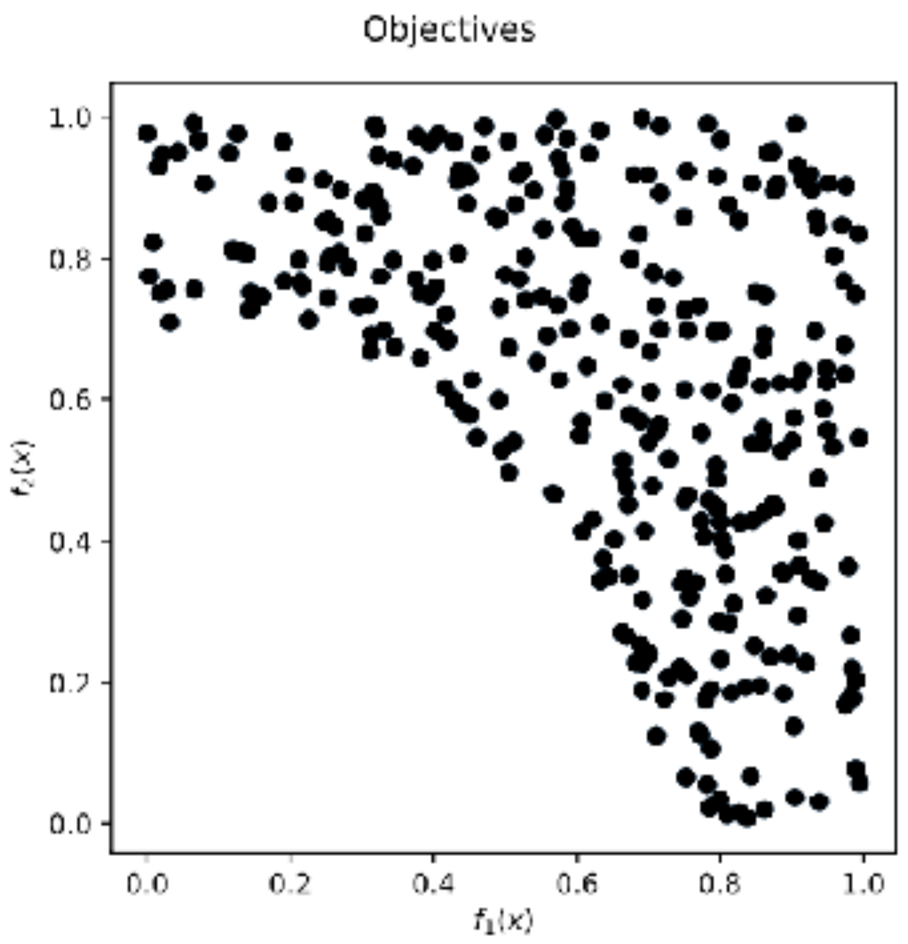
We can vary the weights and obtain different part of the Pareto front

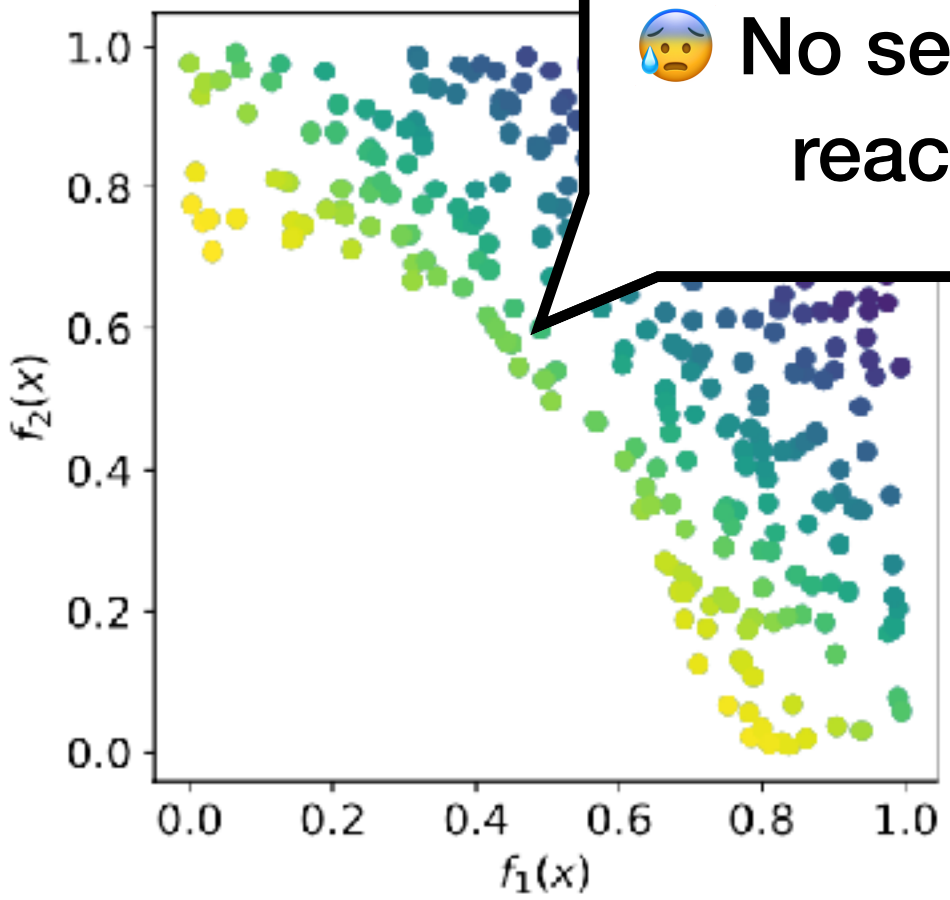
🤔 What is a limitation of this method?

🙈 We can reach the whole Pareto front if its convex!







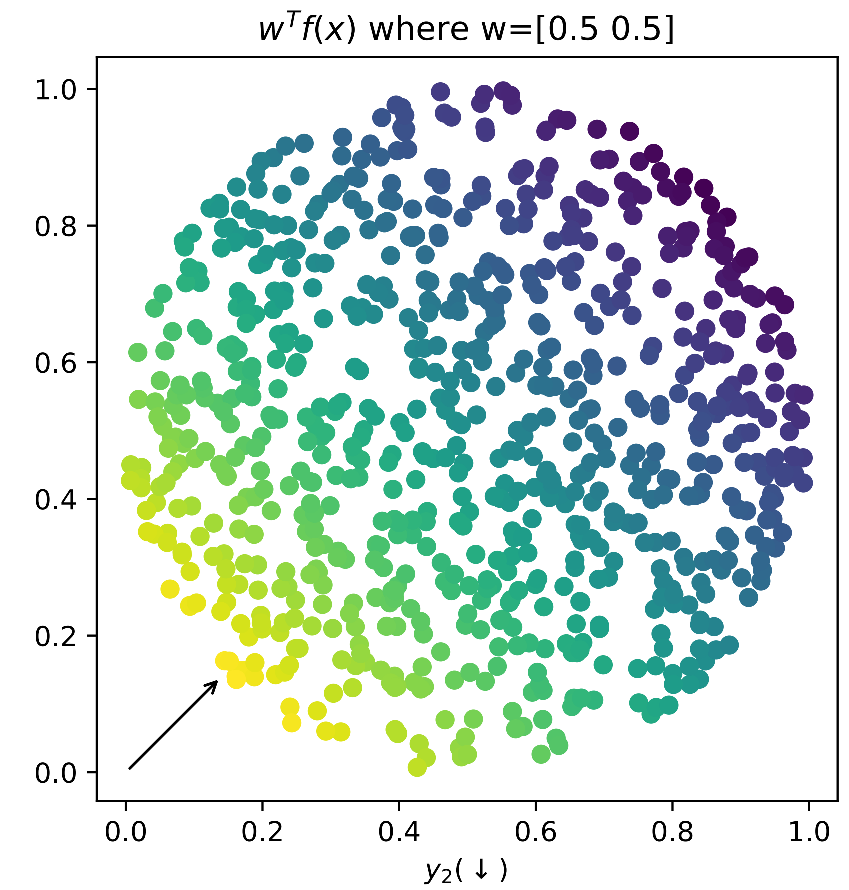


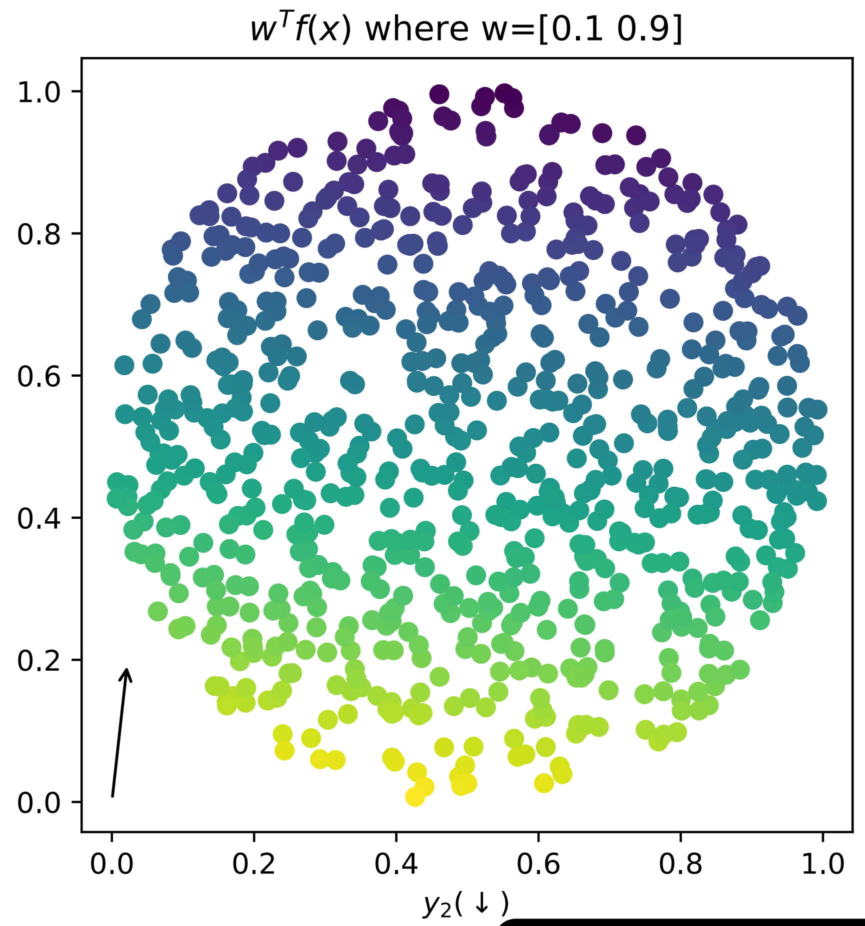
😰 No set of weights can reach this point!

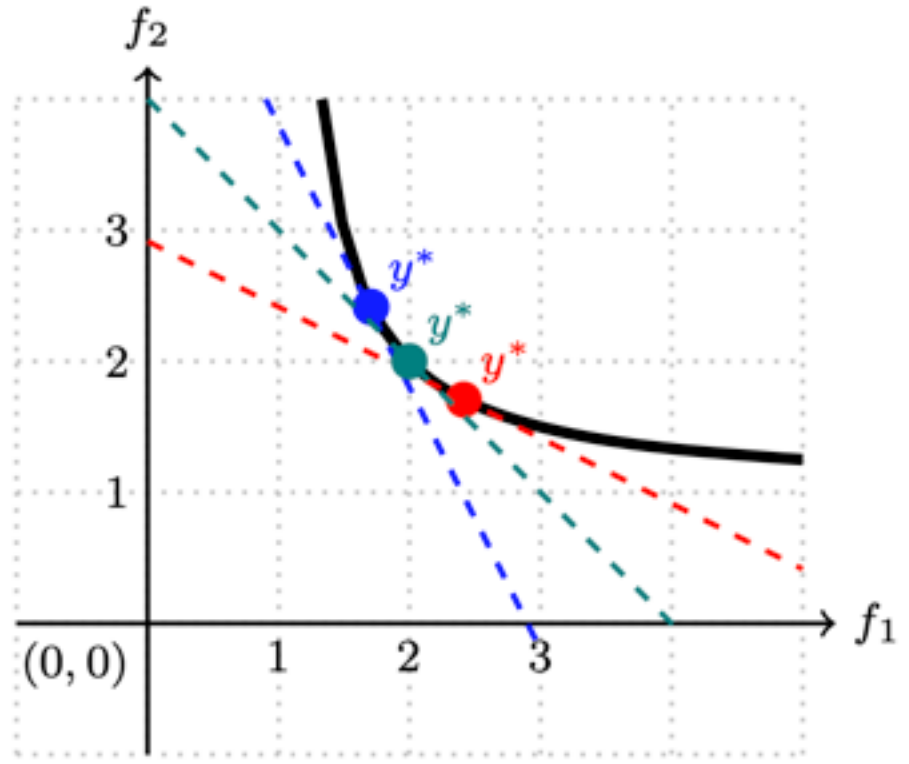
Optimization Methods - scalarization

- Simplest scalarization: $g(x) = \sum_{i=1}^{d} w_i f_i(x) = w^T f(x) \in \mathbb{R}$ where $w \in \mathbb{R}^d$ is a weight vector to be defined (for instance $w_i = 1 \, \forall i$)

We can vary the weights and obtain different part of the Pareto front

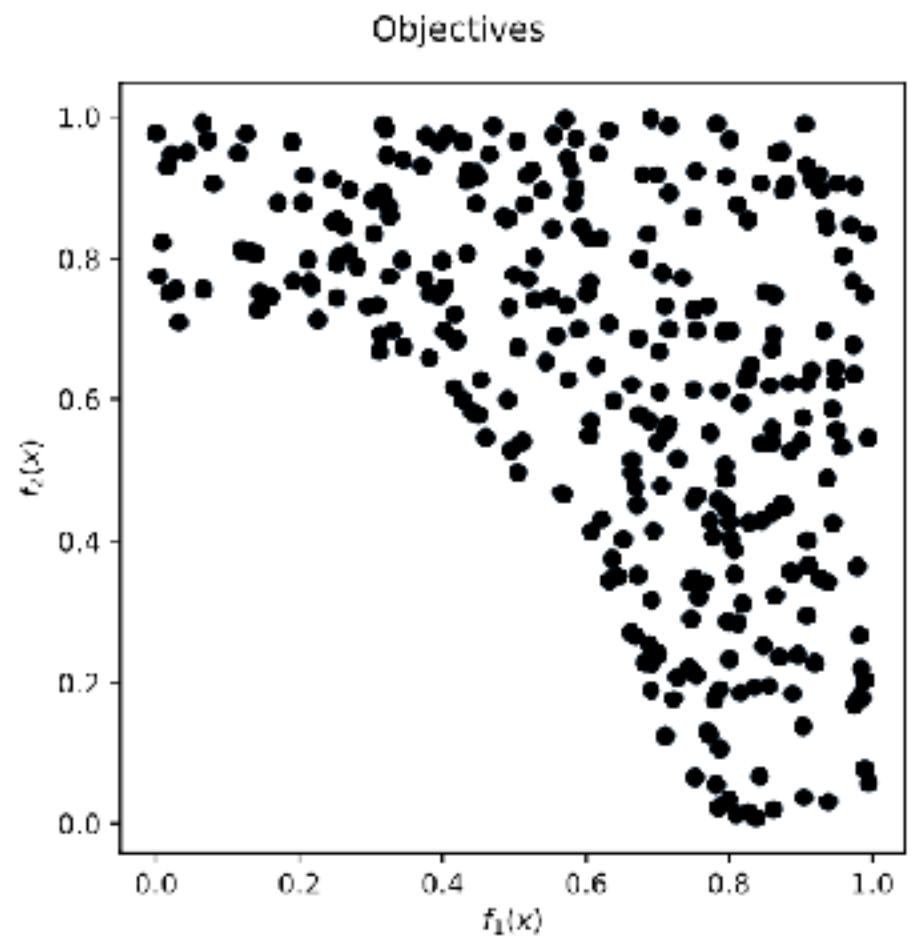


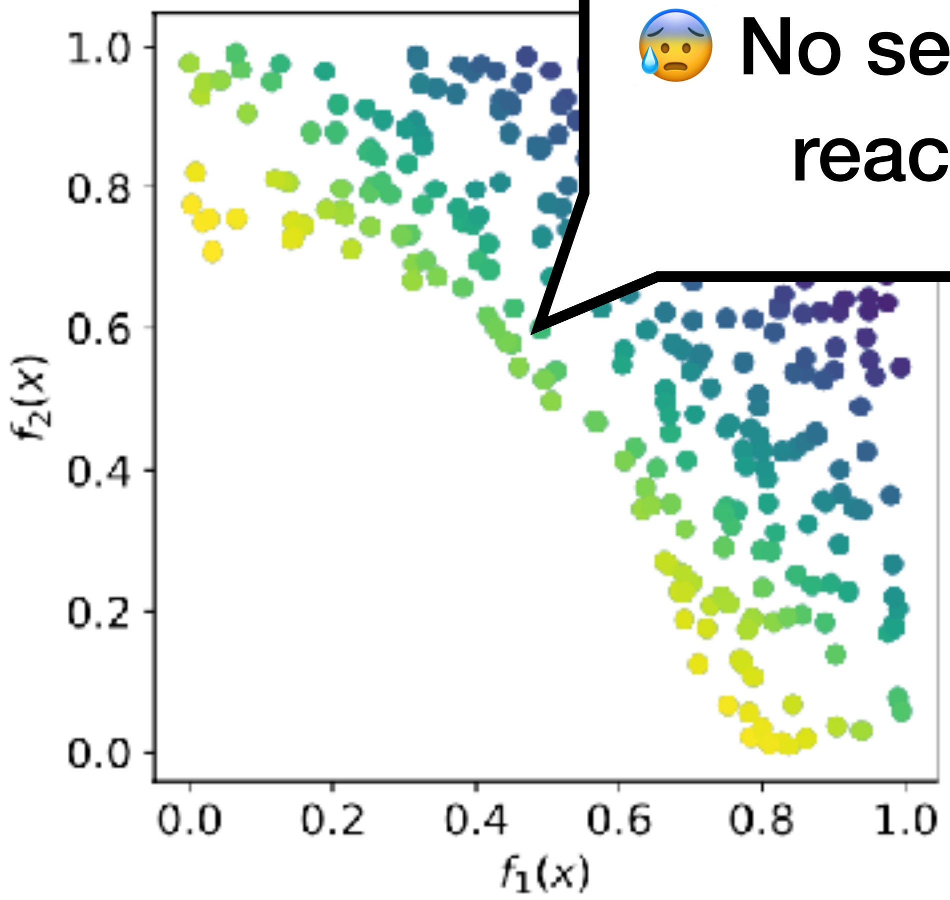




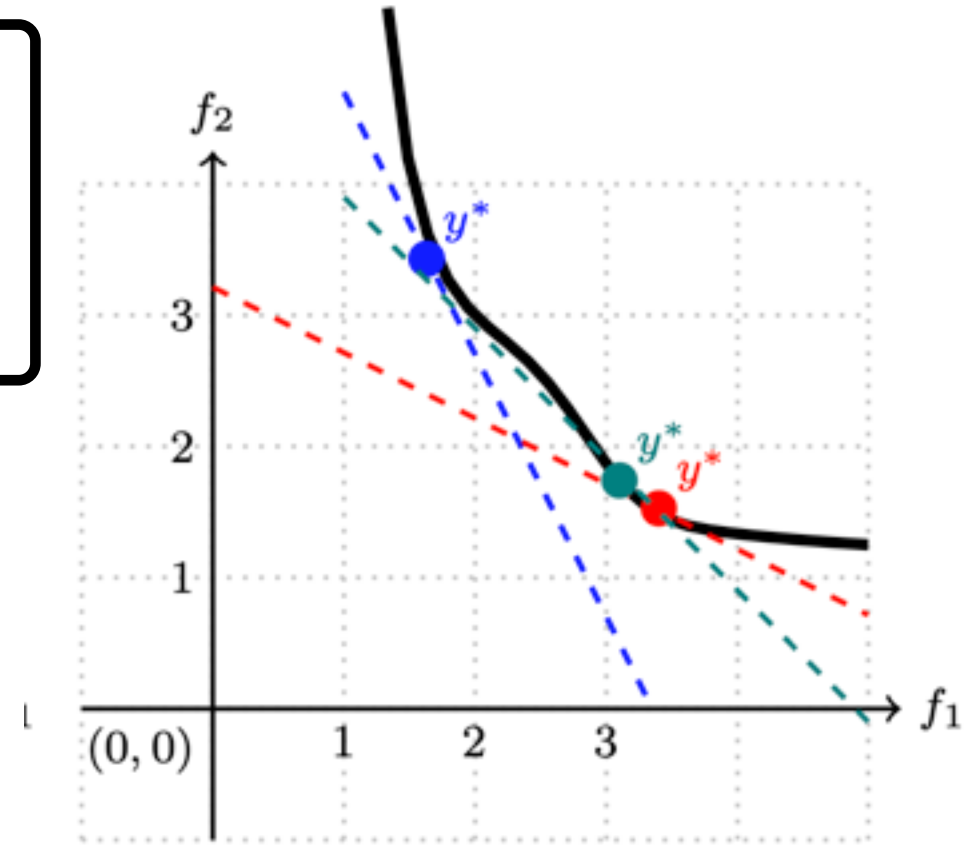
🤔 What is a limitation of this method?

🙈 We can reach the whole Pareto front if its convex!





😰 No set of weights can reach this point!



A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]

The shape of Pareto fronts

The shape of Pareto fronts

- In multiobjective optimization, one often optimize for an “accuracy” objective and another “model-size” objective (latency, number of parameters, …)

The shape of Pareto fronts

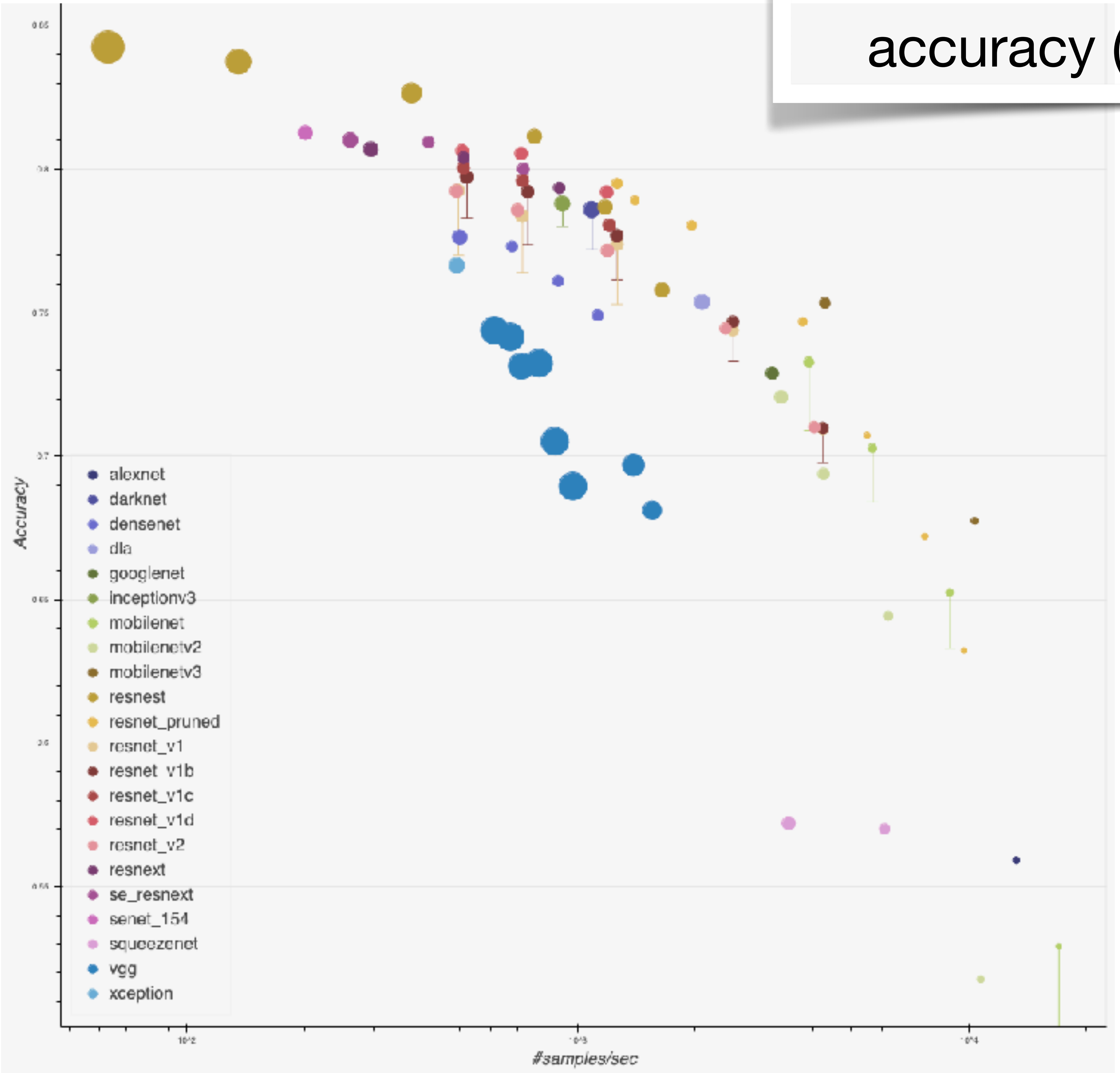
- In multiobjective optimization, one often optimize for an “accuracy” objective and another “model-size” objective (latency, number of parameters, …)
- In this case, the Pareto front is **convex**

The shape of Pareto fronts

- In multiobjective optimization, one often optimize for an “accuracy” objective and another “model-size” objective (latency, number of parameters, …)

- In this case, the Pareto front is **convex**

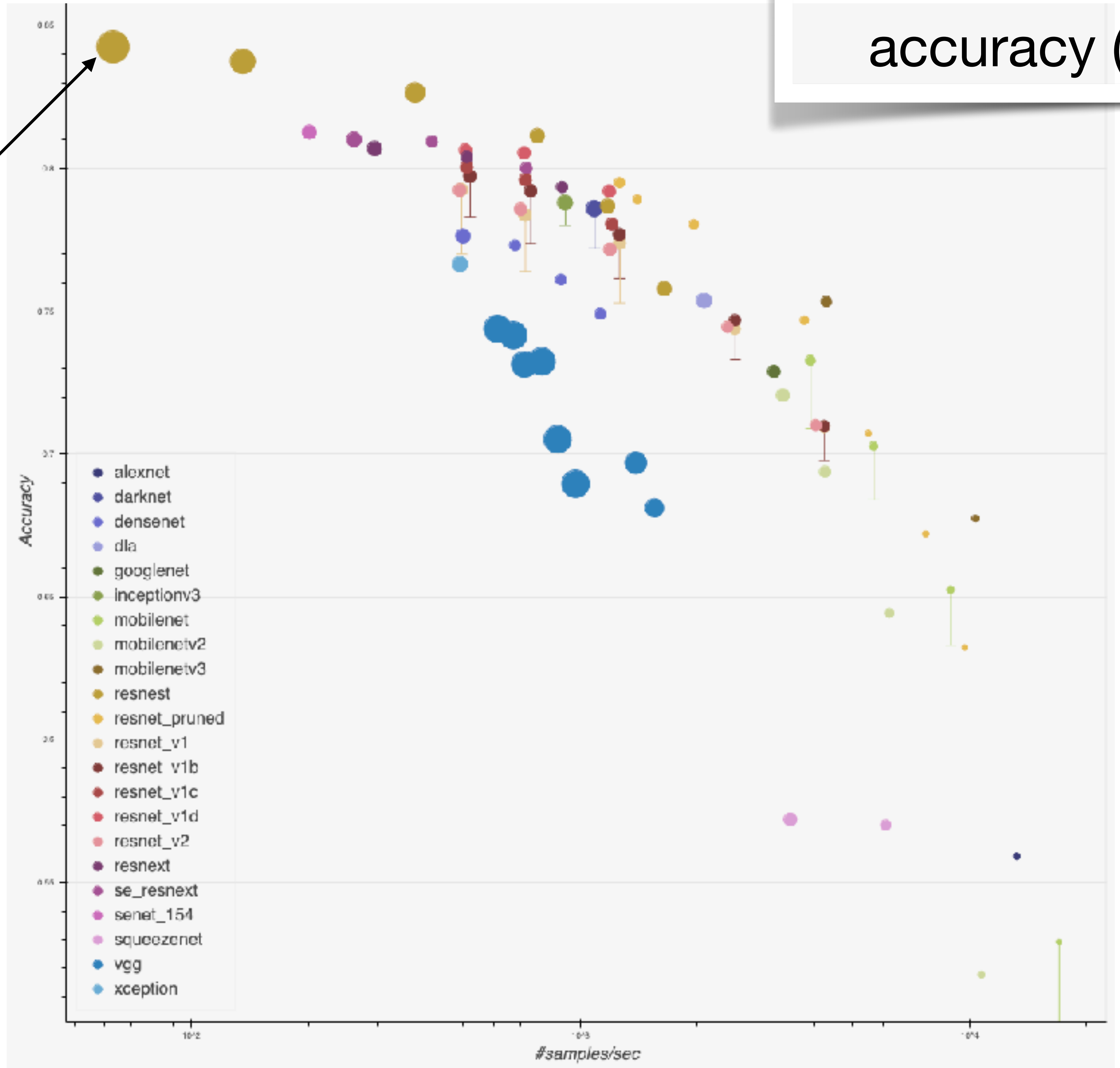
Computer vision models plotted by accuracy (↑) and throughput (↑)



The shape of Pareto fronts

- In multiobjective optimization, one often optimize for an “accuracy” objective and another “model-size” objective (latency, number of parameters, …)
- In this case, the Pareto front is **convex**

Computer vision models plotted by accuracy (↑) and throughput (↑)



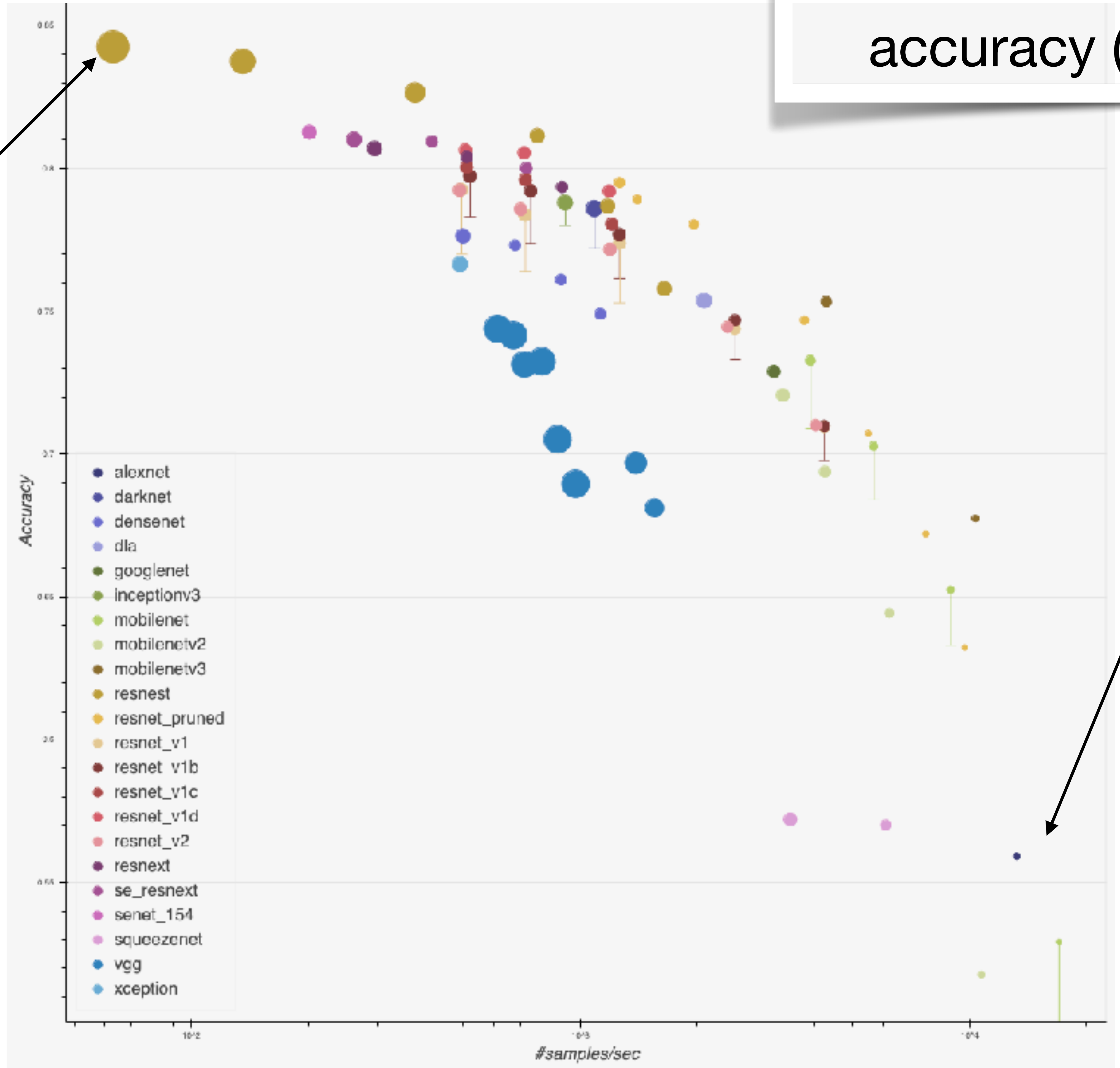
If we decrease our constrain on throughput starting from the maximum throughput, the accuracy drops slowly initially

The shape of Pareto fronts

- In multiobjective optimization, one often optimize for an “accuracy” objective and another “model-size” objective (latency, number of parameters, …)

- In this case, the Pareto front is **convex**

Computer vision models plotted by accuracy (↑) and throughput (↑)



If we decrease our constrain on throughput starting from the maximum throughput, the accuracy drops slowly initially

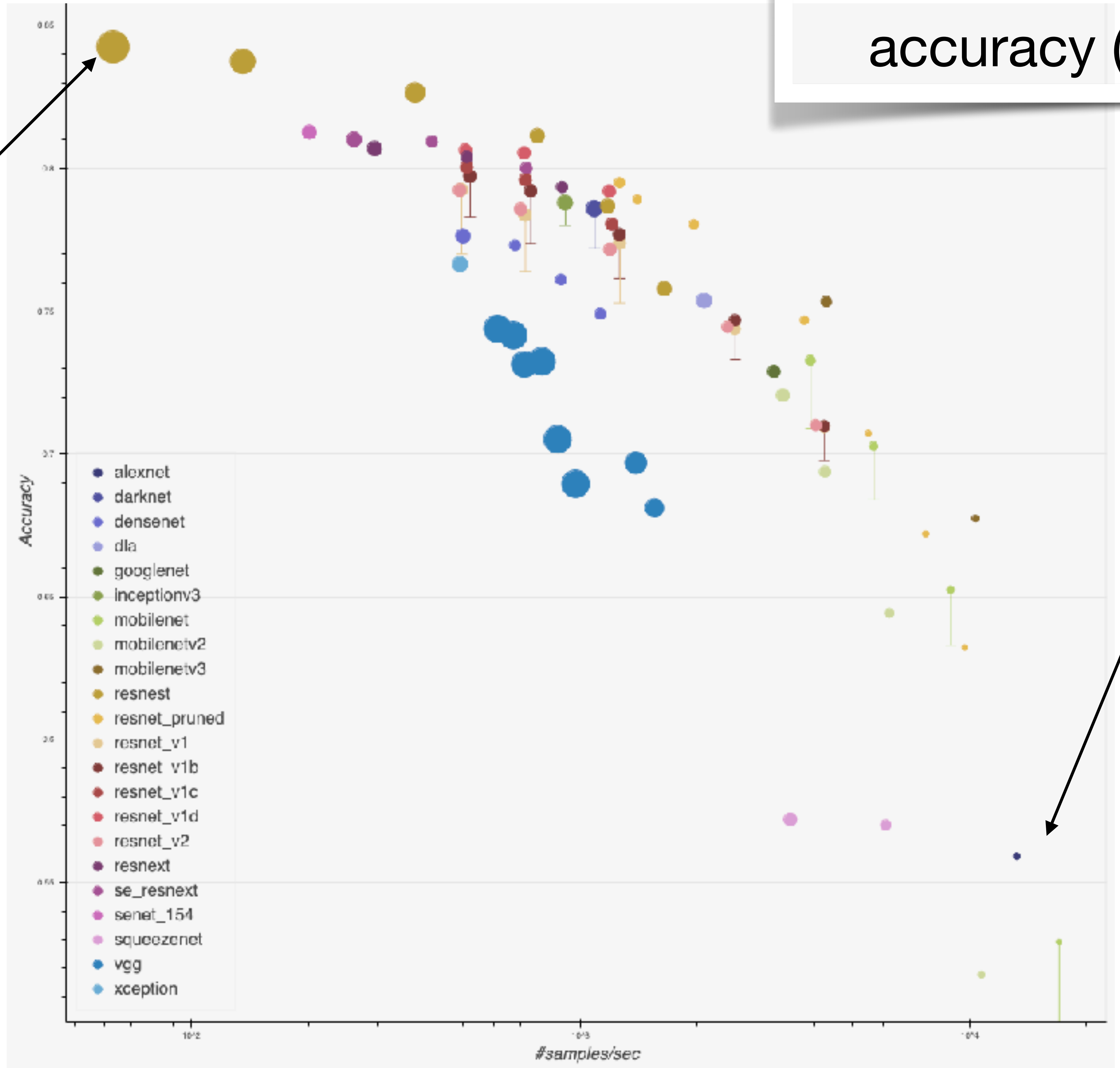
Conversely, if we release the constrain from the highest throughput, we get large gains

The shape of Pareto fronts

- In multiobjective optimization, one often optimize for an “accuracy” objective and another “model-size” objective (latency, number of parameters, …)
- In this case, the Pareto front is **convex**

Computer vision models plotted by accuracy (↑) and throughput (↑)

If we decrease our constrain on throughput starting from the maximum throughput, the accuracy drops slowly initially



Conversely, if we release the constrain from the highest throughput, we get large gains

🤔 But linear scalarization is still limiting, need to find weights, no general guarantee

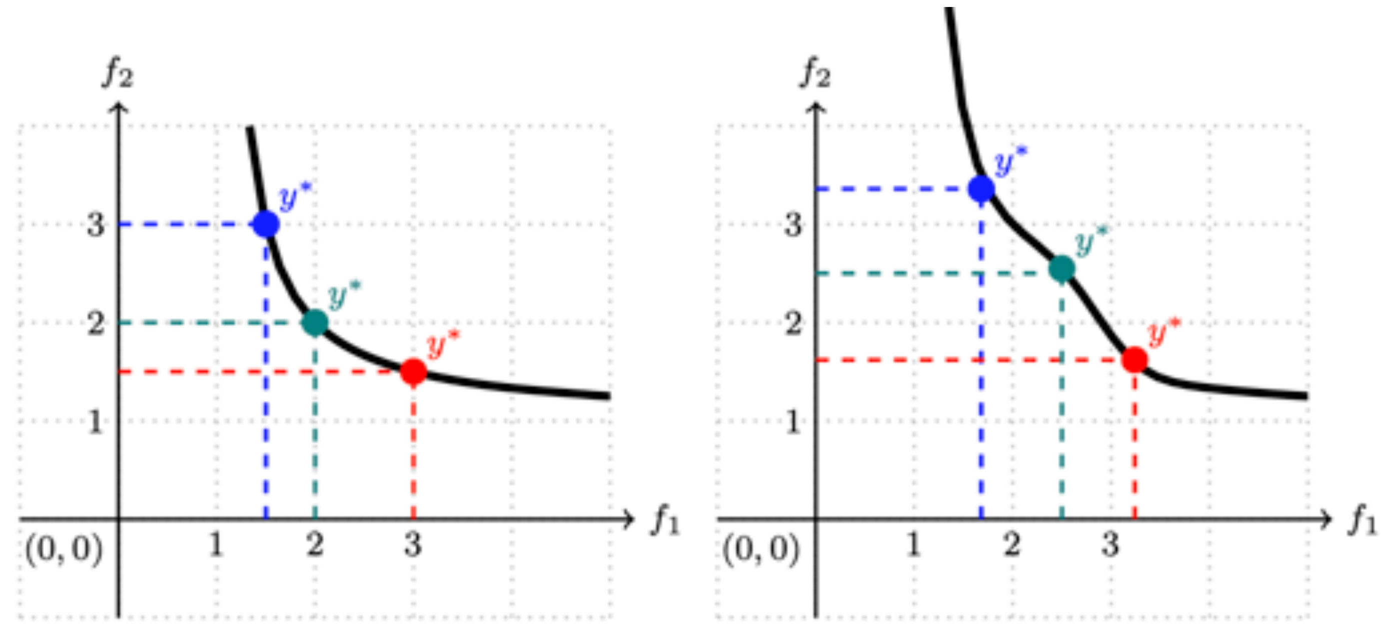
Some other scalarizations

Some other scalarizations

- Chebychev scalarization: $\max_{i=1..d} w_i \, | f_i(x) - z_i^* |$

Some other scalarizations

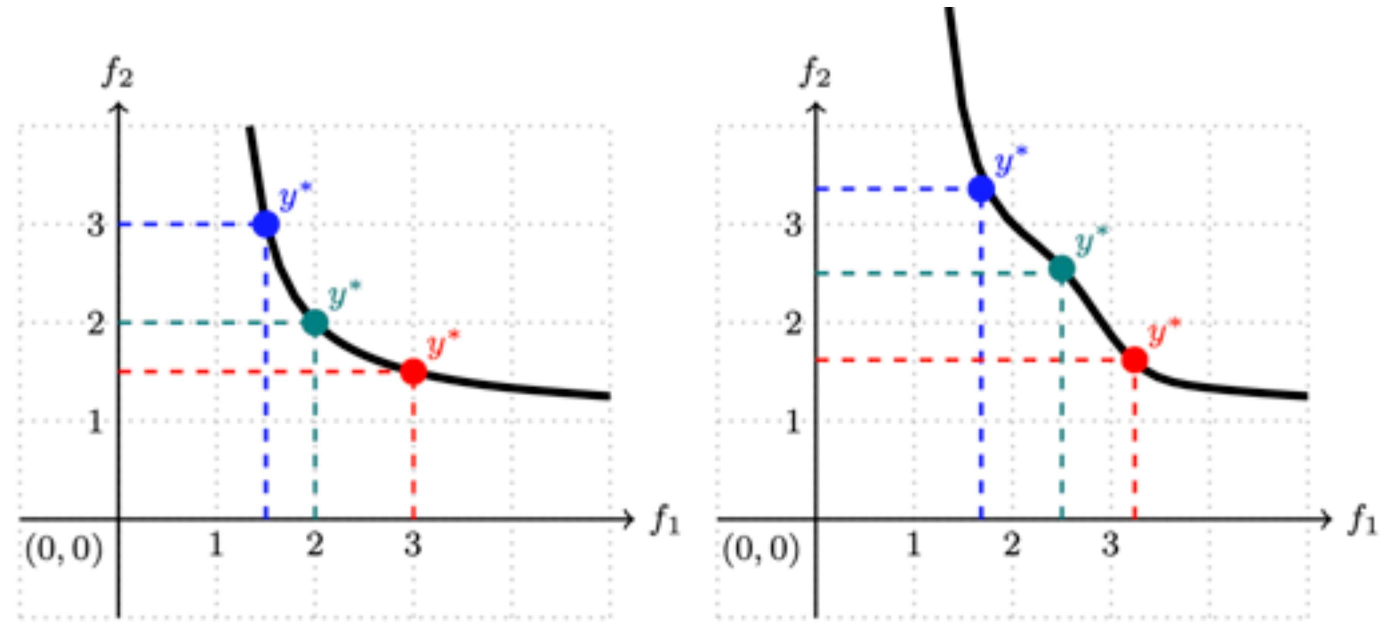
- Chebychev scalarization: $\max_{i=1..d} w_i \,|\, f_i(x) - z_i^* \,|$



Chebychev scalarization. A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]

Some other scalarizations

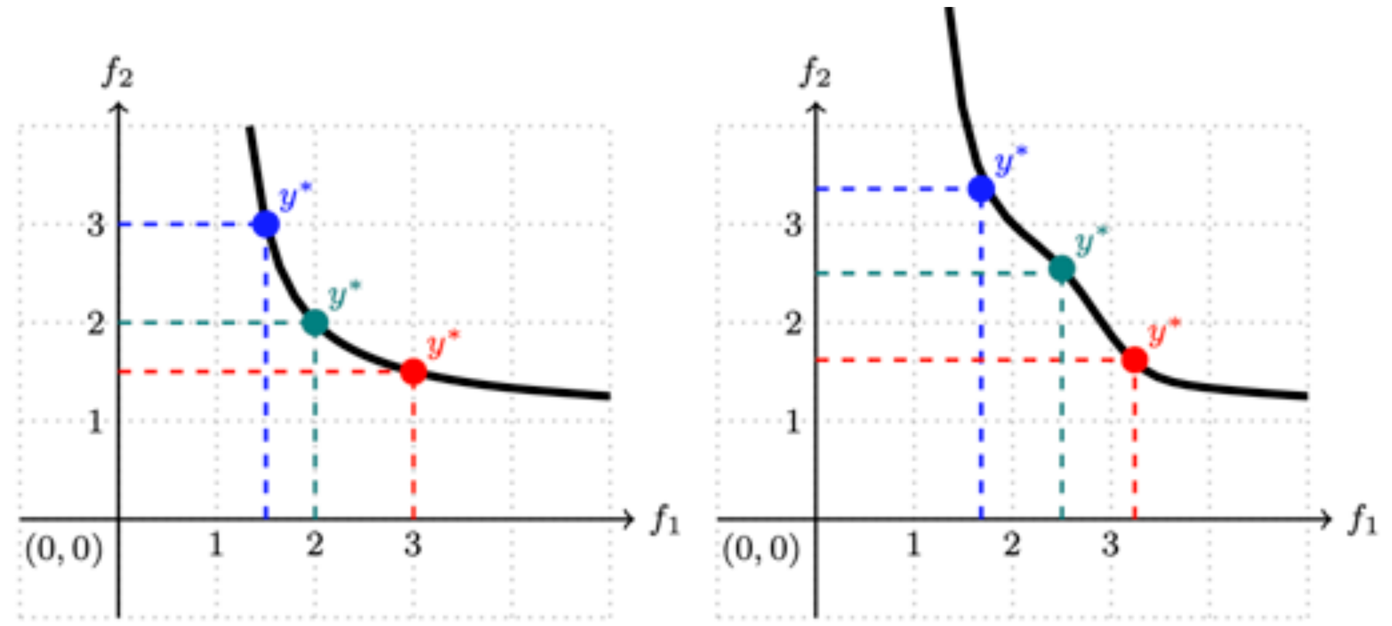
- Chebychev scalarization: $\max_{i=1..d} w_i | f_i(x) - z_i^* |$
  - 👍 Can model concave Pareto front



Chebychev scalarization. A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]

Some other scalarizations

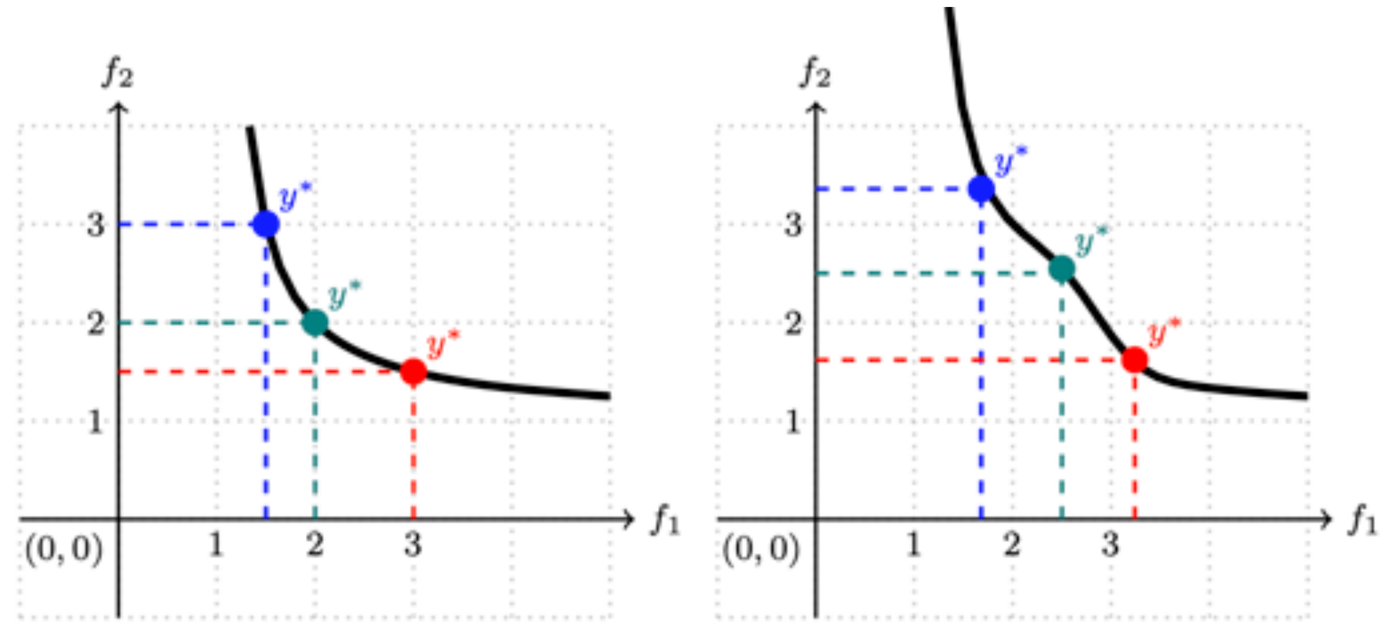
- Chebychev scalarization: $\max_{i=1..d} w_i \,|\, f_i(x) - z_i^* \,|$
    - 👍 Can model concave Pareto front
    - 👎 No regret bound



Chebychev scalarization. A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]

Some other scalarizations

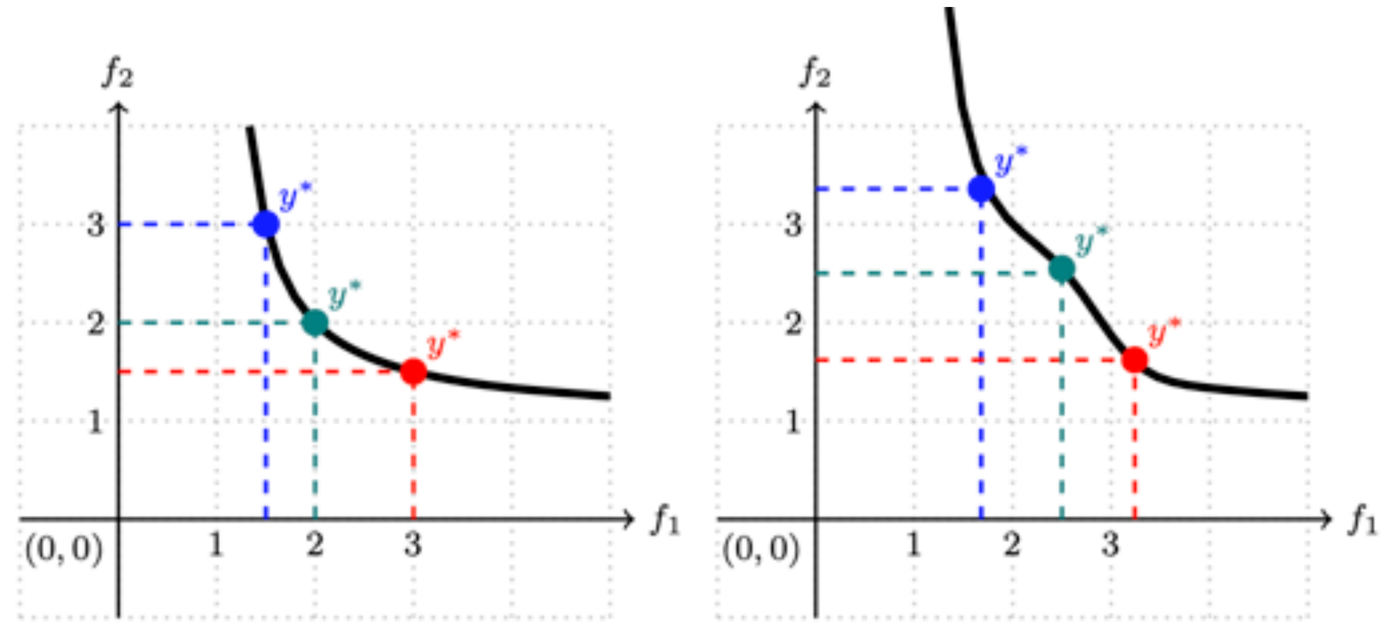
- Chebychev scalarization: $\max_{i=1..d} w_i \, | f_i(x) - z_i^* |$
  - 👍 Can model concave Pareto front
  - 👎 No regret bound
- Golovin scalarization: $\min_{i=1..d} \max(0, f_i(x)/w_i))^d$



Chebychev scalarization. A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]

Some other scalarizations

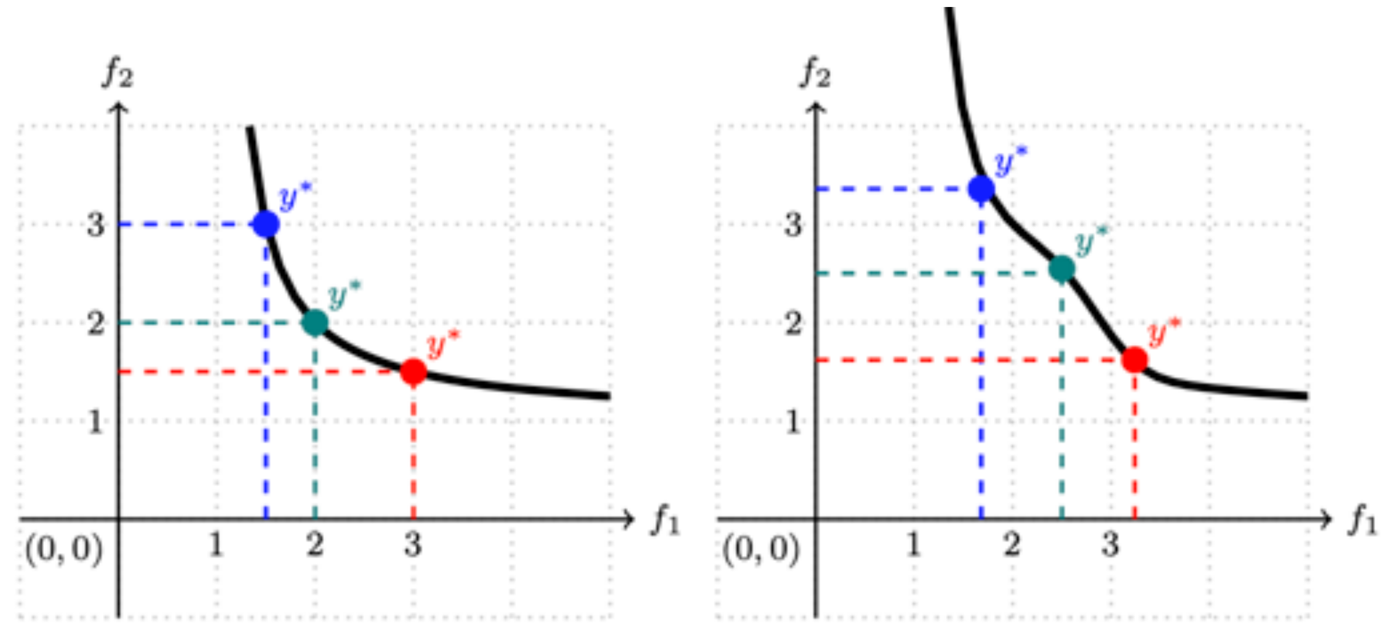
- Chebychev scalarization: $\max_{i=1..d} w_i |f_i(x) - z_i^*|$
  - 👍 Can model concave Pareto front
  - 👎 No regret bound
- Golovin scalarization: $\min_{i=1..d} \max(0, f_i(x)/w_i))^d$
  - 👍 Regret bound



Chebychev scalarization. A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]

Some other scalarizations

- Chebychev scalarization: $\max_{i=1..d} w_i \,|\, f_i(x) - z_i^* \,|$
  - 👍 Can model concave Pareto front
  - 👎 No regret bound
- Golovin scalarization: $\min_{i=1..d} \max(0, f_i(x)/w_i))^d$
  - 👍 Regret bound
  - 👎 Needs to be able to bound the objectives



Chebychev scalarization. A tutorial on multiobjective optimization: fundamentals and evolutionary methods [Emmerich 2018]

Evolutionary multiobjective optimization

Evolutionary multiobjective optimization

- Applies/extend Evolutionary Algorithms (EA)

Evolutionary multiobjective optimization

- Applies/extend Evolutionary Algorithms (EA)
- Three main approaches:

Evolutionary multiobjective optimization

- Applies/extend Evolutionary Algorithms (EA)
- Three main approaches:
  - Pareto based (NSGA-II): applies EA and uses **non-dominated sort** to sort candidates

Evolutionary multiobjective optimization

- Applies/extend Evolutionary Algorithms (EA)
- Three main approaches:
  - Pareto based (NSGA-II): applies EA and uses **non-dominated sort** to sort candidates
  - Indicator based (SMS-EMOA): optimises a set indicator (for instance hypervolume), for instance removes the solution with the smallest hypervolume contribution from the population

Evolutionary multiobjective optimization

- Applies/extend Evolutionary Algorithms (EA)
- Three main approaches:
  - Pareto based (NSGA-II): applies EA and uses **non-dominated sort** to sort candidates
  - Indicator based (SMS-EMOA): optimises a set indicator (for instance hypervolume), for instance removes the solution with the smallest hypervolume contribution from the population
  - Decomposition based (MOEA/D): decomposes into subregions of the Pareto front, for each applies different scalarization parameters

Non dominated sort

Non dominated sort

- Assume we want to minimize all objectives

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
    - Compute the Pareto front of y, break ties with an heuristic

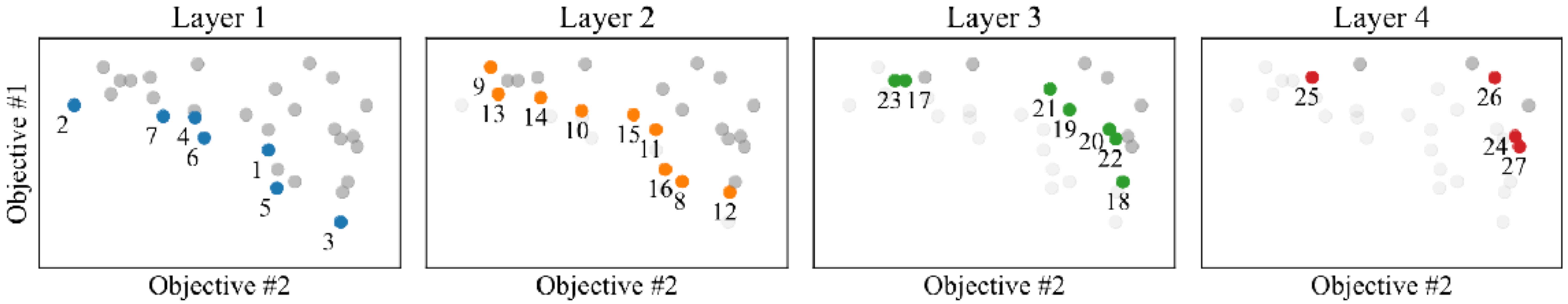


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic

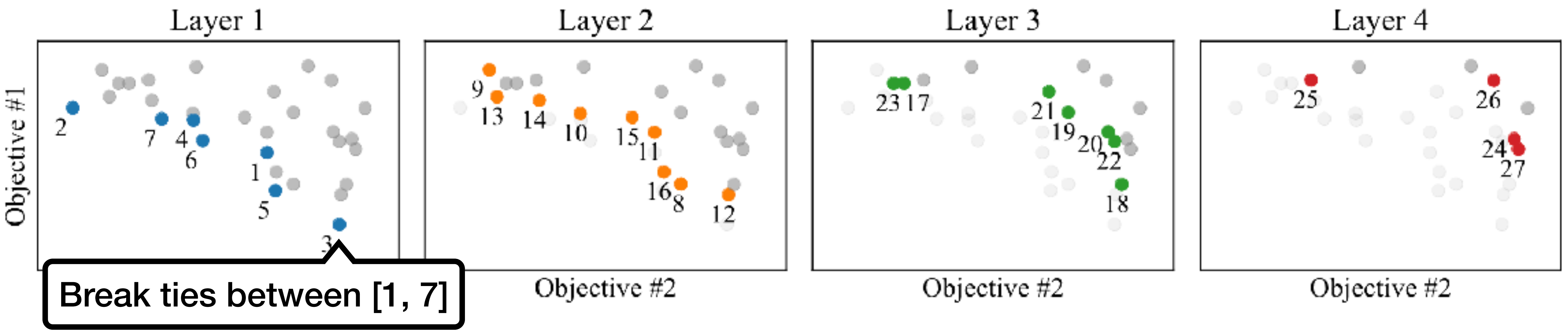


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic

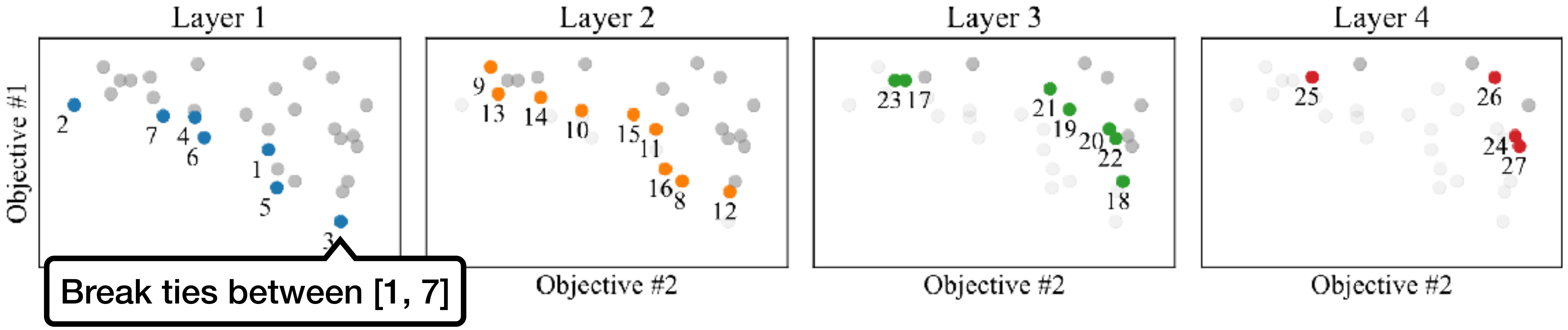


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic

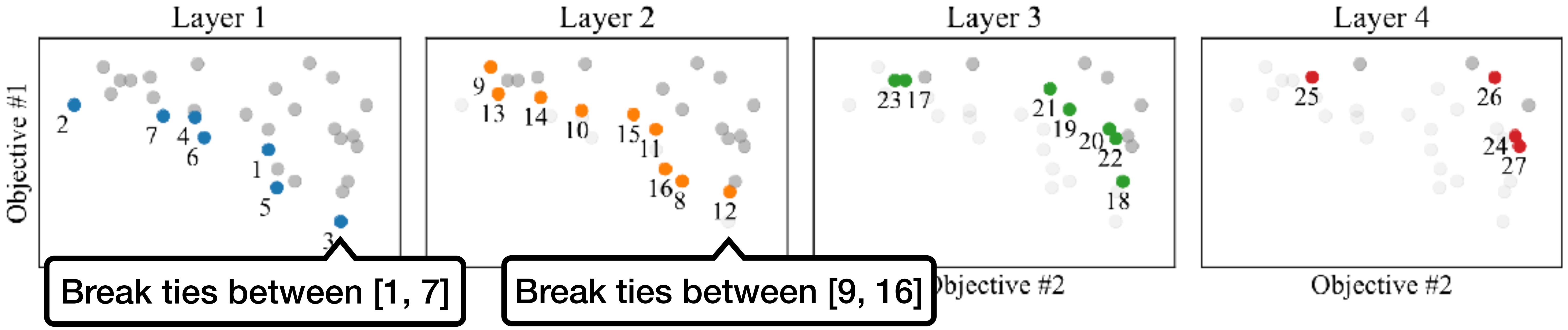


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:

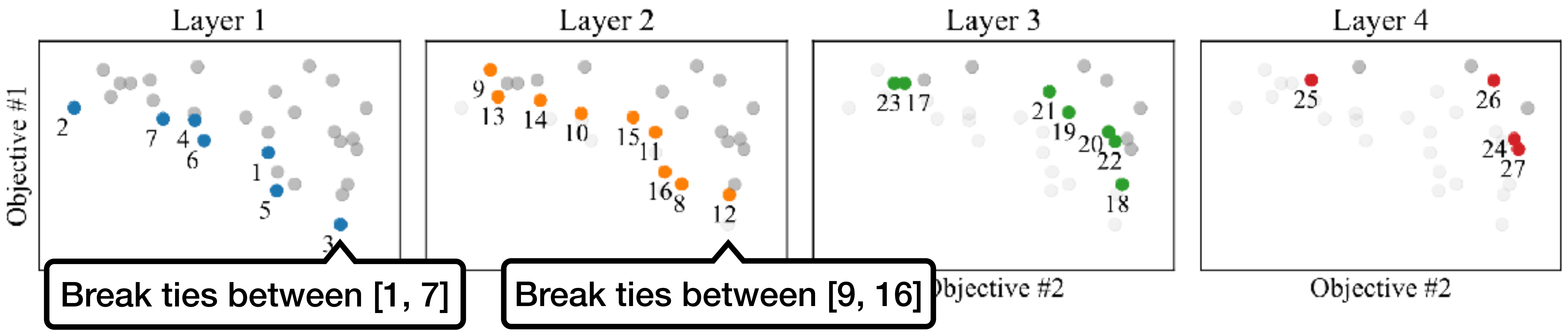


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathcal{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:
    - Crowding distance

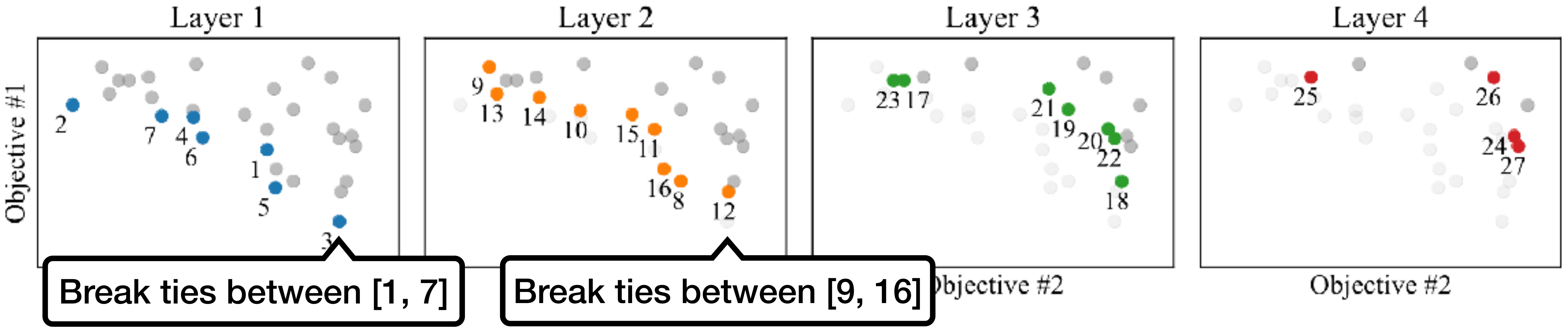


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathscr{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:
    - Crowding distance
    - Epsilon-net

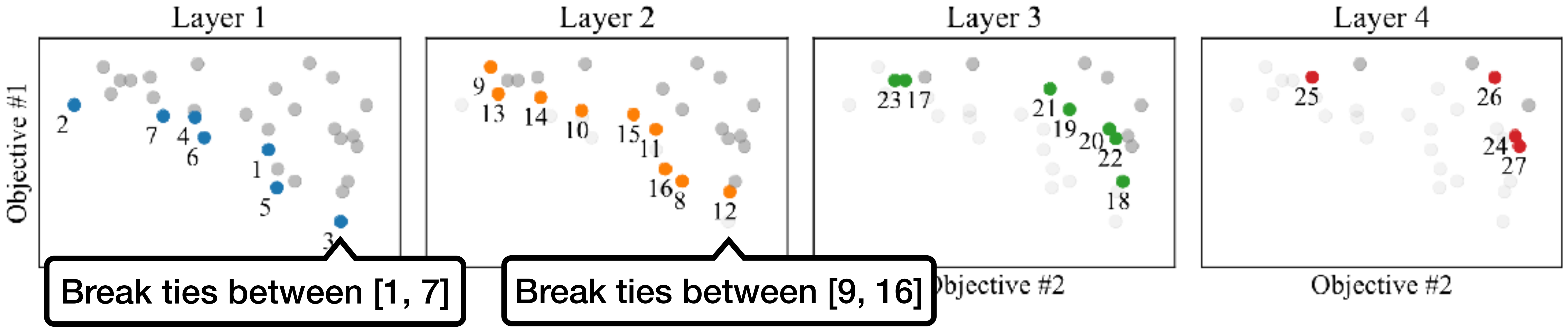


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

Non dominated sort

- Assume we want to minimize all objectives
- How to *rank* n observations $y \in \mathbb{R}^{n \times d}$ when we have $d > 1$ objectives?
- Non dominated sort (aka onion sort):
  - Compute the Pareto front of y, break ties with an heuristic
  - Compute the Pareto front of $y \backslash \mathcal{P}(y)$, break ties with an heuristic
  - Heuristic choices aims at selecting a subset with a good coverage:
    - Crowding distance
    - Epsilon-net
    - …

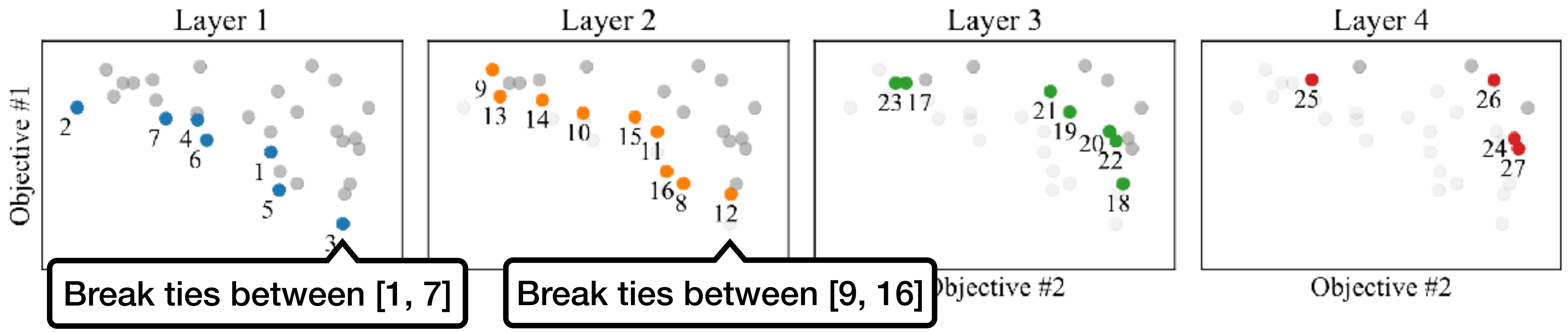
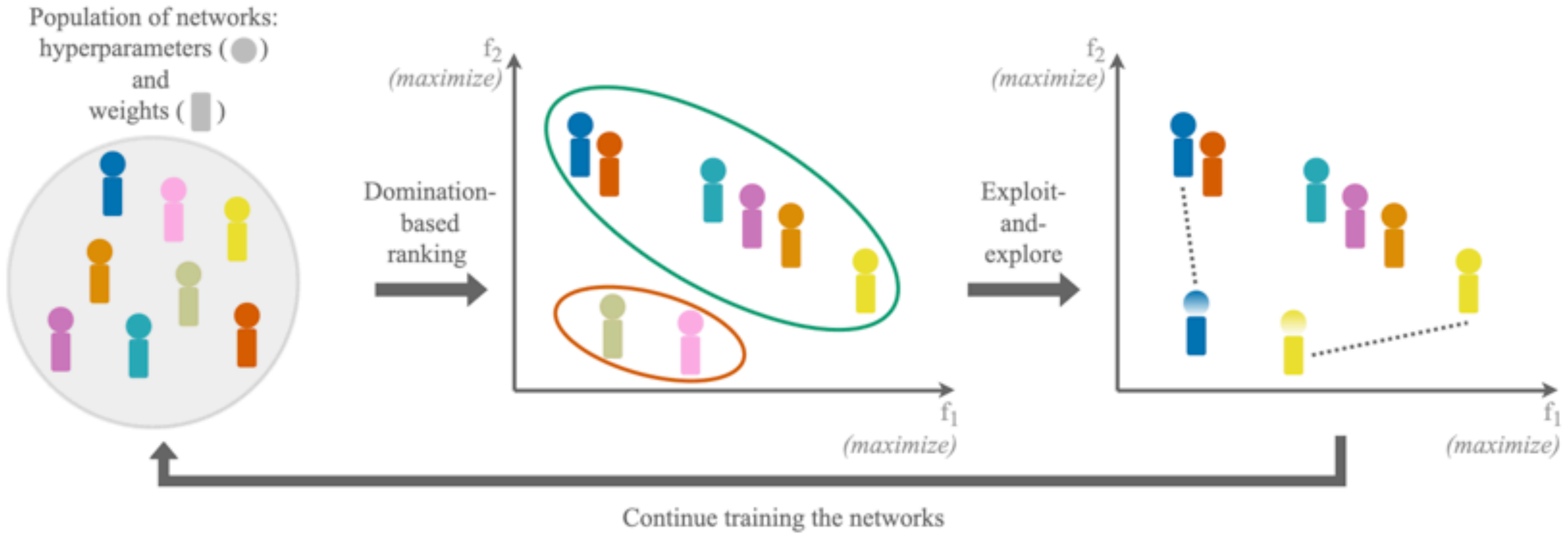


Figure 3: Illustration of non-dominated sorting. The layers show the partitioning of the data in Pareto fronts. The numbers depict the overall rank by computing the ϵ-net within each layer.

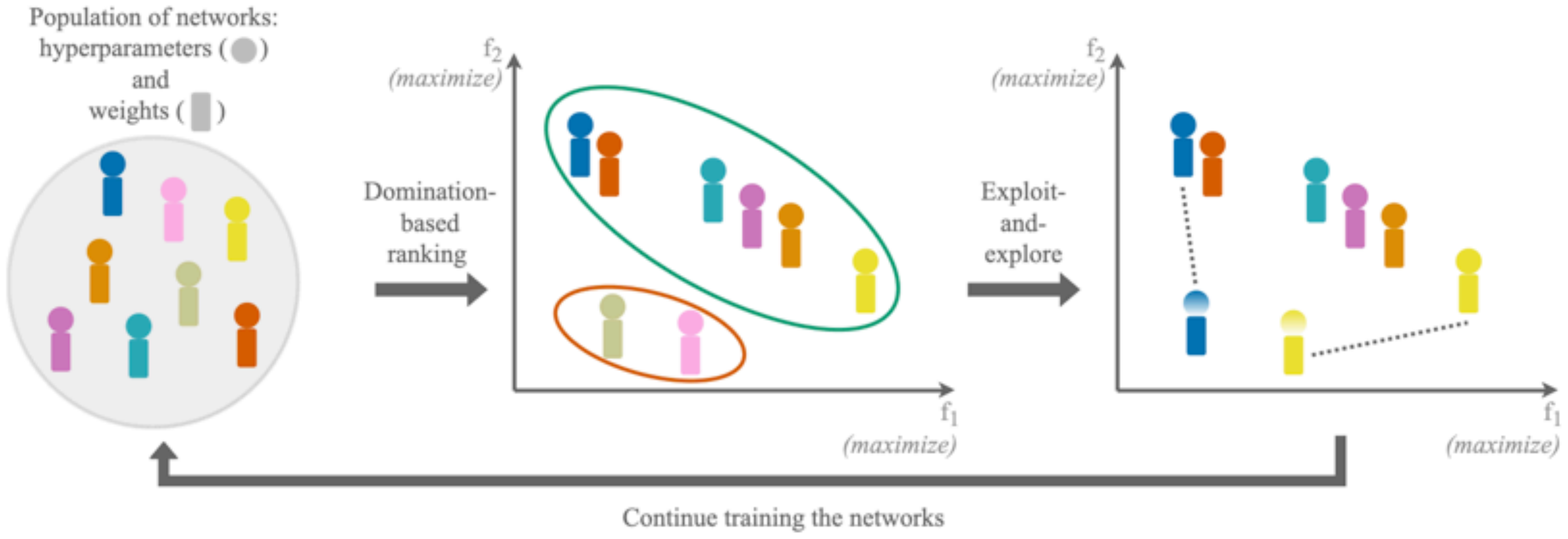
NSGA-II



Multi-Objective Population Based Training [Dushatskiy 2023]

NSGA-II

- Initialise population $X_n \subset \mathcal{X}^n$ of n configurations
- While not converged:
  - $X =$ mutate-and-combine(X_n) // gets many candidates, possibly more than n
  - $X =$ non-dominated-sort(X) // sort them in a multiobjective way
  - $X_n = X[: n]$ // keep top n candidates



Multi-Objective Population Based Training [Dushatskiy 2023]

Bayesian Optimization methods

GPareto [Binois 2019]

Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron

GPareto [Binois 2019]

Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*

GPareto [Binois 2019]

Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*
    - Some models the objectives jointly by modelling *correlations* between objectives [Wilson 2010, Alvarez 2011]

GPareto [Binois 2019]

Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*
    - Some models the objectives jointly by modelling *correlations* between objectives [Wilson 2010, Alvarez 2011]
- BO can be extended by considering multi-objective acquisition functions

GPareto [Binois 2019]

Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*
    - Some models the objectives jointly by modelling *correlations* between objectives [Wilson 2010, Alvarez 2011]
- BO can be extended by considering multi-objective acquisition functions
    - Expected hypervolume improvement (EHI) [Emmerich 2011]: pick configuration that have largest expected hypervolume gain

GPareto [Binois 2019]

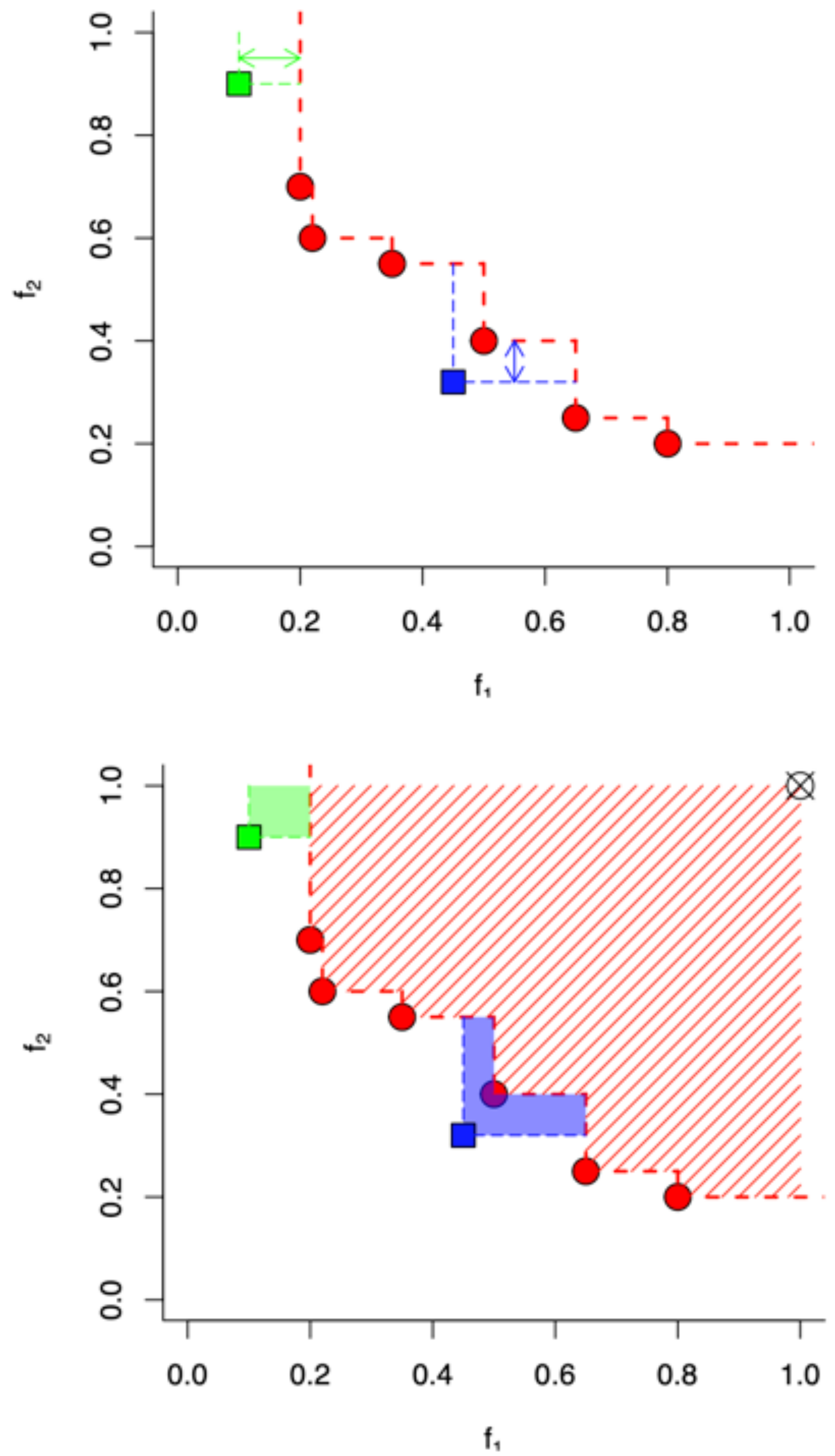
Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*
  - Some models the objectives jointly by modelling *correlations* between objectives [Wilson 2010, Alvarez 2011]
- BO can be extended by considering multi-objective acquisition functions
  - Expected hypervolume improvement (EHI) [Emmerich 2011]: pick configuration that have largest expected hypervolume gain
  - Expected maximin improvement (EMI) [Svenson 2016]: pick configuration that has largest expected maximin improvement

GPareto [Binois 2019]

Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*
  - Some models the objectives jointly by modelling *correlations* between objectives [Wilson 2010, Alvarez 2011]
- BO can be extended by considering multi-objective acquisition functions
  - Expected hypervolume improvement (EHI) [Emmerich 2011]: pick configuration that have largest expected hypervolume gain
  - Expected maximin improvement (EMI) [Svenson 2016]: pick configuration that has largest expected maximin improvement



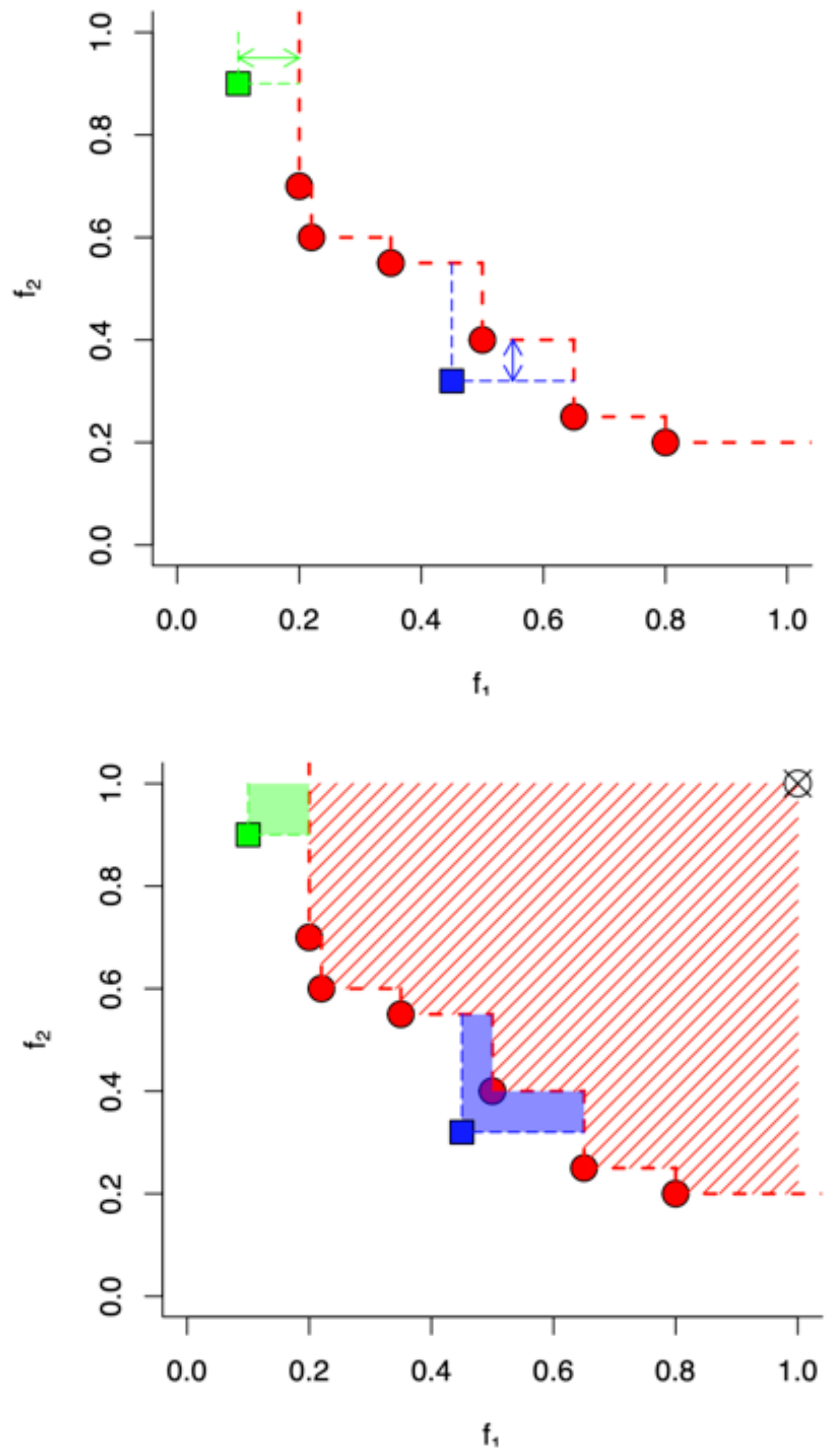
Margin and hypervolume improvement.

GPareto [Binois 2019]

GPareto [Binois 2019]

Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*
    - Some models the objectives jointly by modelling *correlations* between objectives [Wilson 2010, Alvarez 2011]
- BO can be extended by considering multi-objective acquisition functions
    - Expected hypervolume improvement (EHI) [Emmerich 2011]: pick configuration that have largest expected hypervolume gain
    - Expected maximin improvement (EMI) [Svenson 2016]: pick configuration that has largest expected maximin improvement
    - S-metric selection EGO (SMS) [Poinweiser 2008]: pick configuration that maximises hyper volume according to lower confidence bound



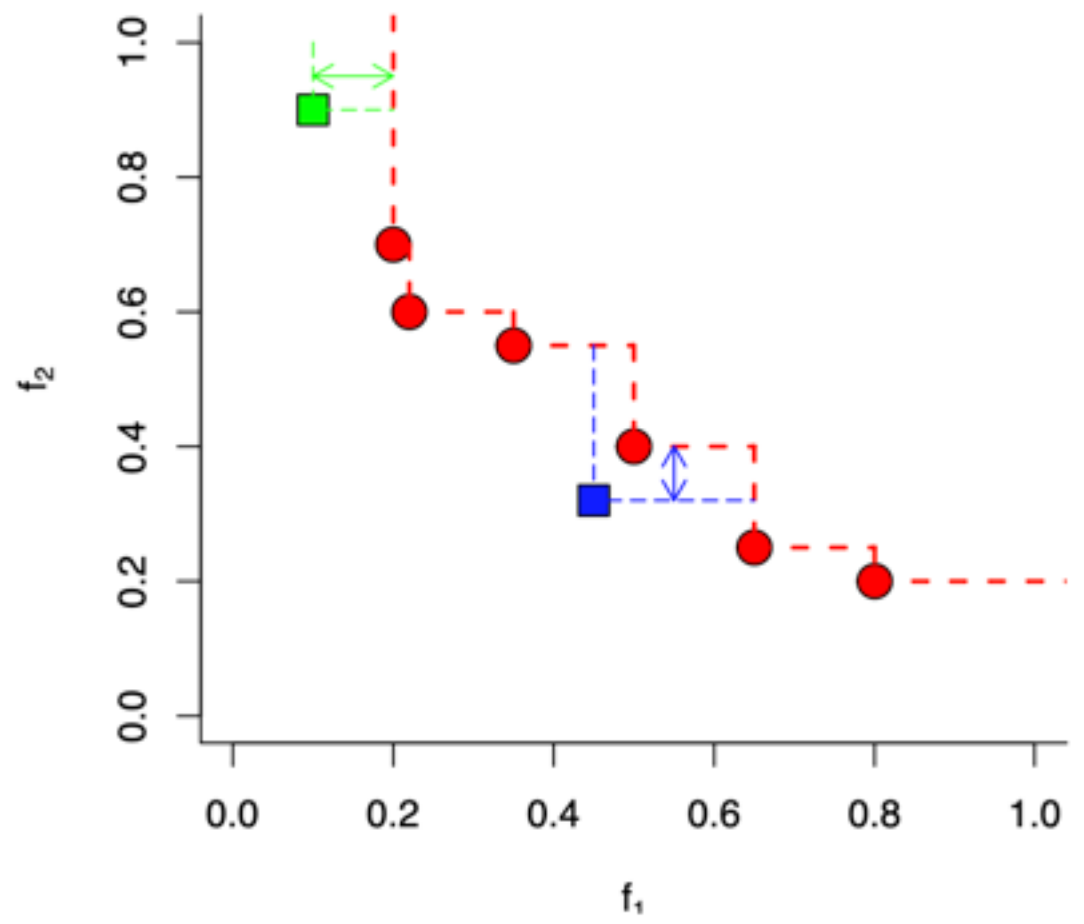
Margin and hypervolume improvement.

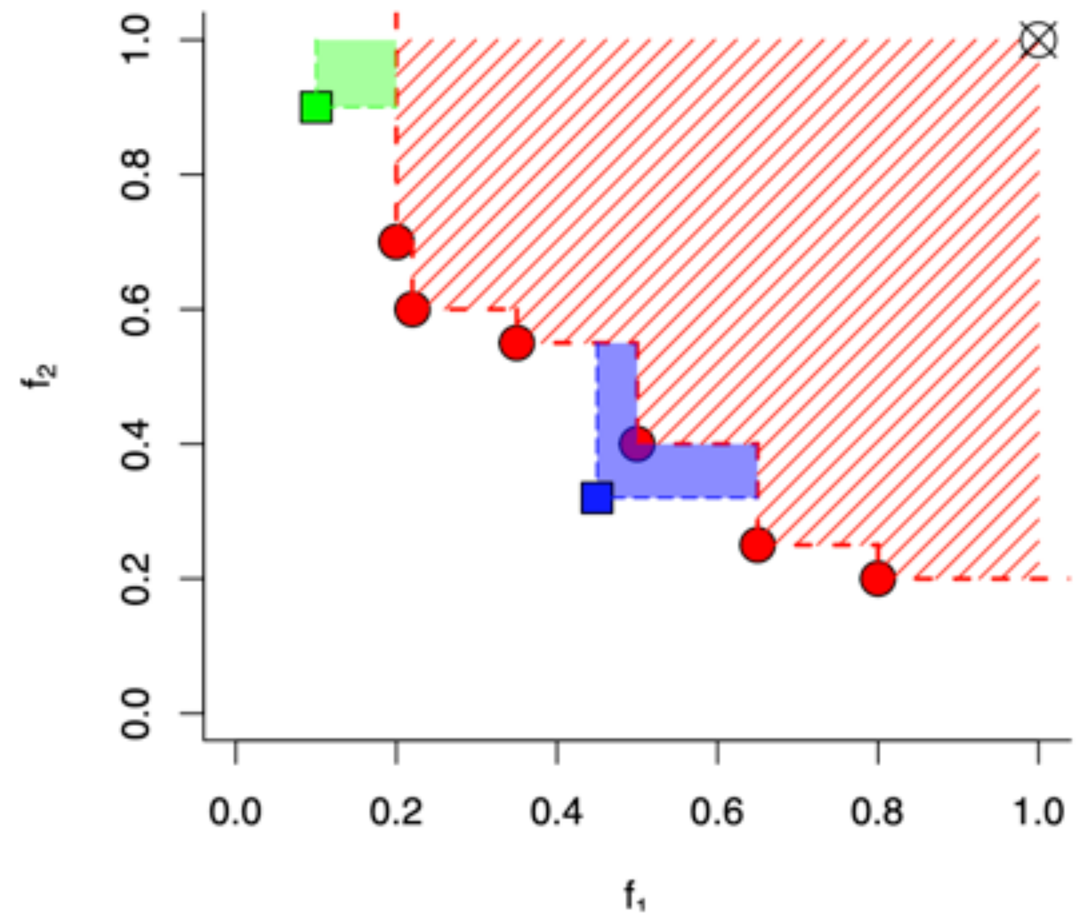
GPareto [Binois 2019]

GPareto [Binois 2019]

Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*
  - Some models the objectives jointly by modelling *correlations* between objectives [Wilson 2010, Alvarez 2011]
- BO can be extended by considering multi-objective acquisition functions
  - Expected hypervolume improvement (EHI) [Emmerich 2011]: pick configuration that have largest expected hypervolume gain
  - Expected maximin improvement (EMI) [Svenson 2016]: pick configuration that has largest expected maximin improvement
  - S-metric selection EGO (SMS) [Poinweiser 2008]: pick configuration that maximises hyper volume according to lower confidence bound
  - Stepwise uncertainty reduction (SUR) [Picheny 2015]: pick configuration that has largest proba of non-domination





Margin and hypervolume improvement.

GPareto [Binois 2019]

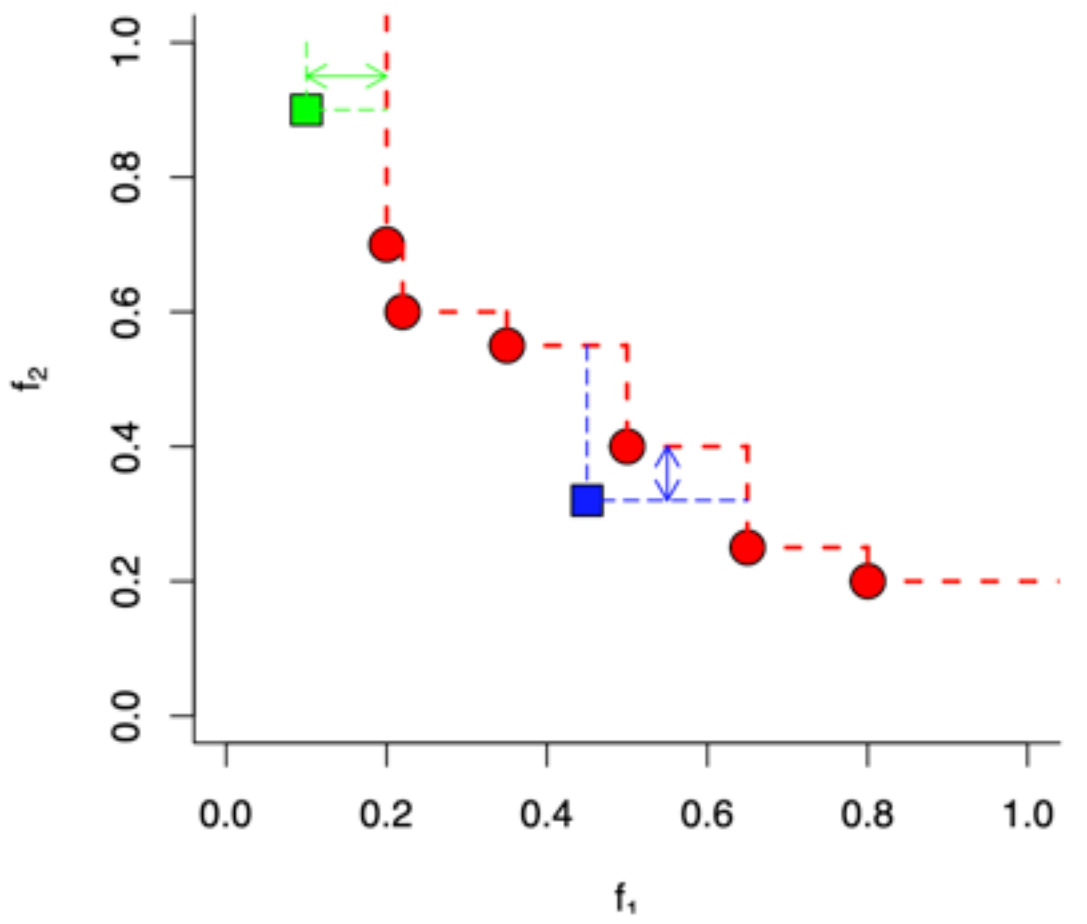
GPareto [Binois 2019]

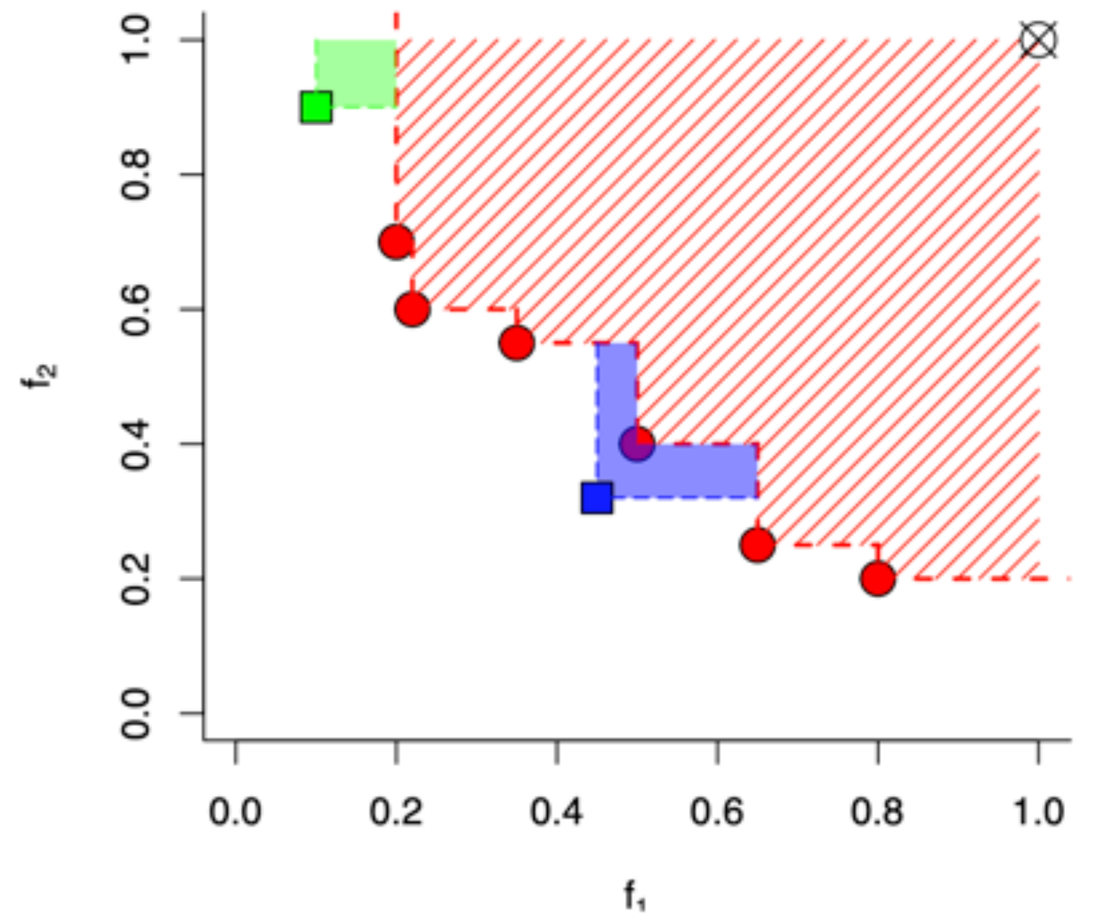
Bayesian Optimization methods

- Gaussian Process can be used as surrogate to estimate the function response $f_i(x)$ for any x (with uncertainty), see first lecture of Aaron
- Most multi-objective approaches fits GP on objectives *individually*
  - Some models the objectives jointly by modelling *correlations* between objectives [Wilson 2010, Alvarez 2011]
- BO can be extended by considering multi-objective acquisition functions
  - Expected hypervolume improvement (EHI) [Emmerich 2011]: pick configuration that have largest expected hypervolume gain
  - Expected maximin improvement (EMI) [Svenson 2016]: pick configuration that has largest expected maximin improvement
  - S-metric selection EGO (SMS) [Poinweiser 2008]: pick configuration that maximises hyper volume according to lower confidence bound
  - Stepwise uncertainty reduction (SUR) [Picheny 2015]: pick configuration that has largest proba of non-domination

Name	Indicator	Analytical	m	Cost	Scaling dependent
`crit_EHI`	Hypervolume	$m = 2$ only	Any	$+$ to $+++$	Yes
`crit_EMI`	Additive-ϵ	No	Any	$++$	Yes
`crit_SMS`	Hypervolume	Yes	Any	$+$	Yes
`crit_SUR`	Probability of non-domination	No	$m \leq 3$	$+++$	No

GPareto [Binois 2019]





Margin and hypervolume improvement.

GPareto [Binois 2019]

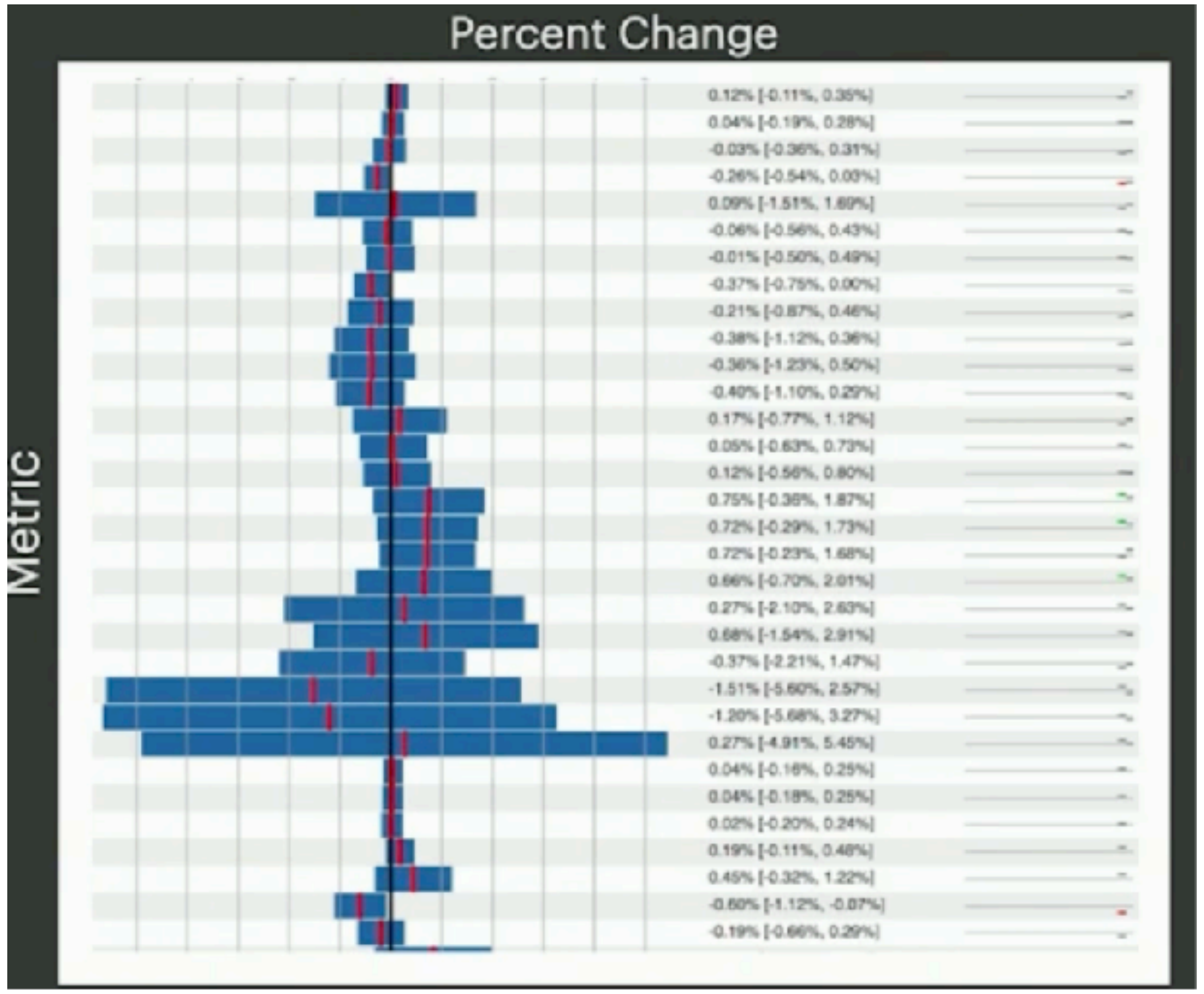
Preference-based methods

Preference-based methods

- Sometimes, we have many objectives and it is hard to know exactly how they should be prioritised

Preference-based methods

- Sometimes, we have many objectives and it is hard to know exactly how they should be prioritised

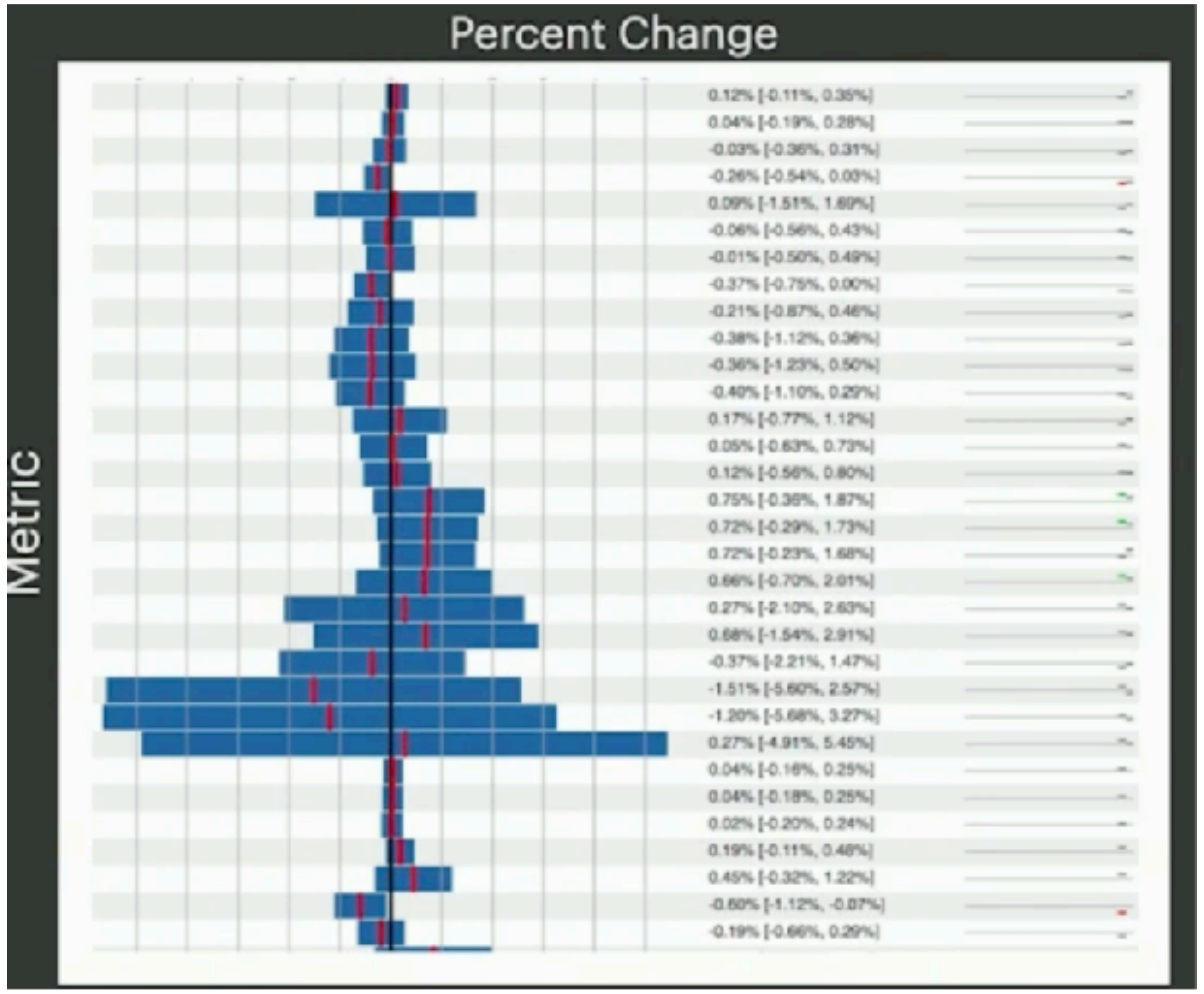


An AB test for many metrics on Facebook. *Beyond Loss Efficient Optimization of Living Machine Learning*. Bakshi Automl 2023

Preference-based methods

- Sometimes, we have many objectives and it is hard to know exactly how they should be prioritised

- Preference based methods ask users to provide feedback on which multiobjective solution is better

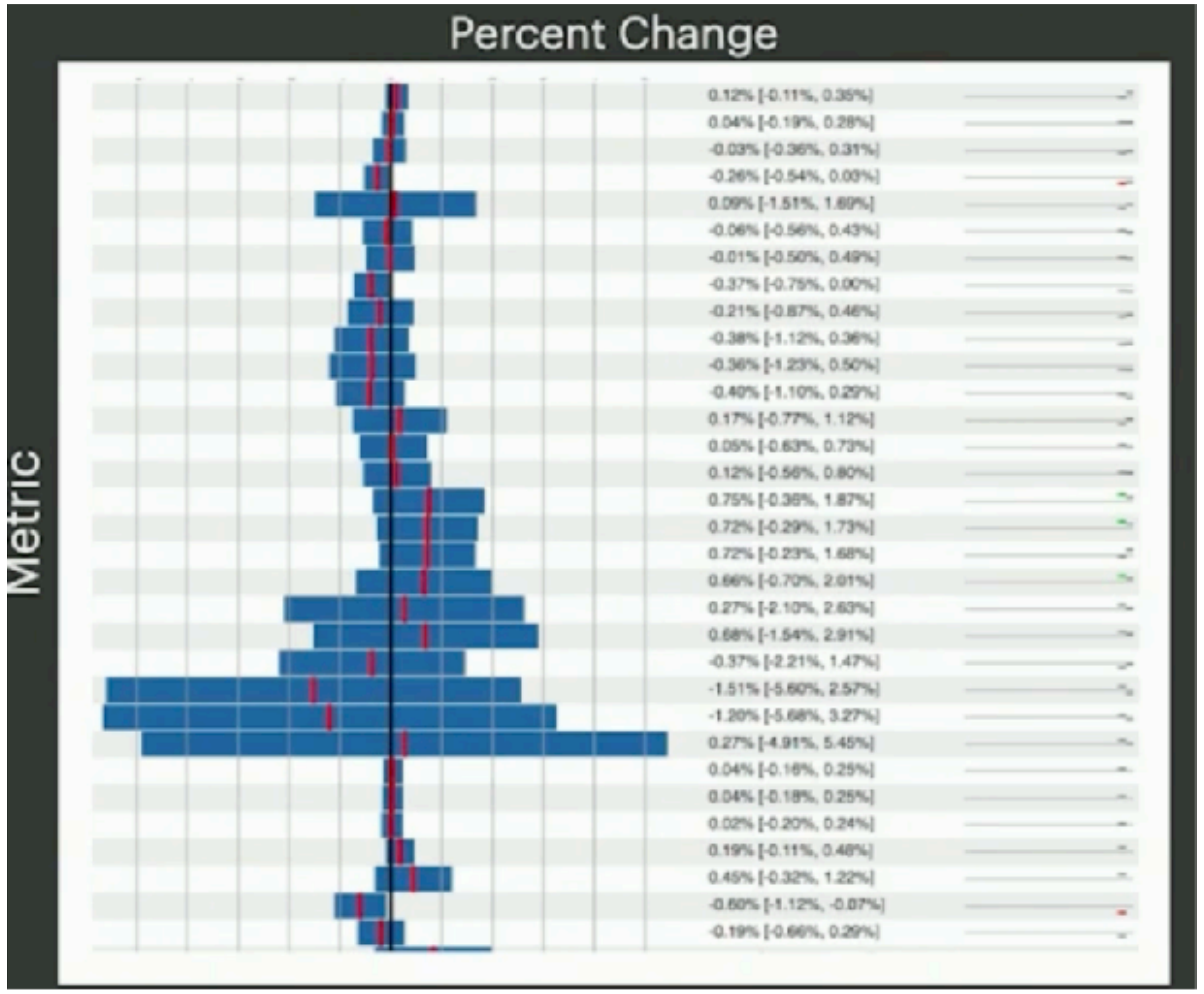


An AB test for many metrics on Facebook. *Beyond Loss Efficient Optimization of Living Machine Learning*. Bakshi Automl 2023

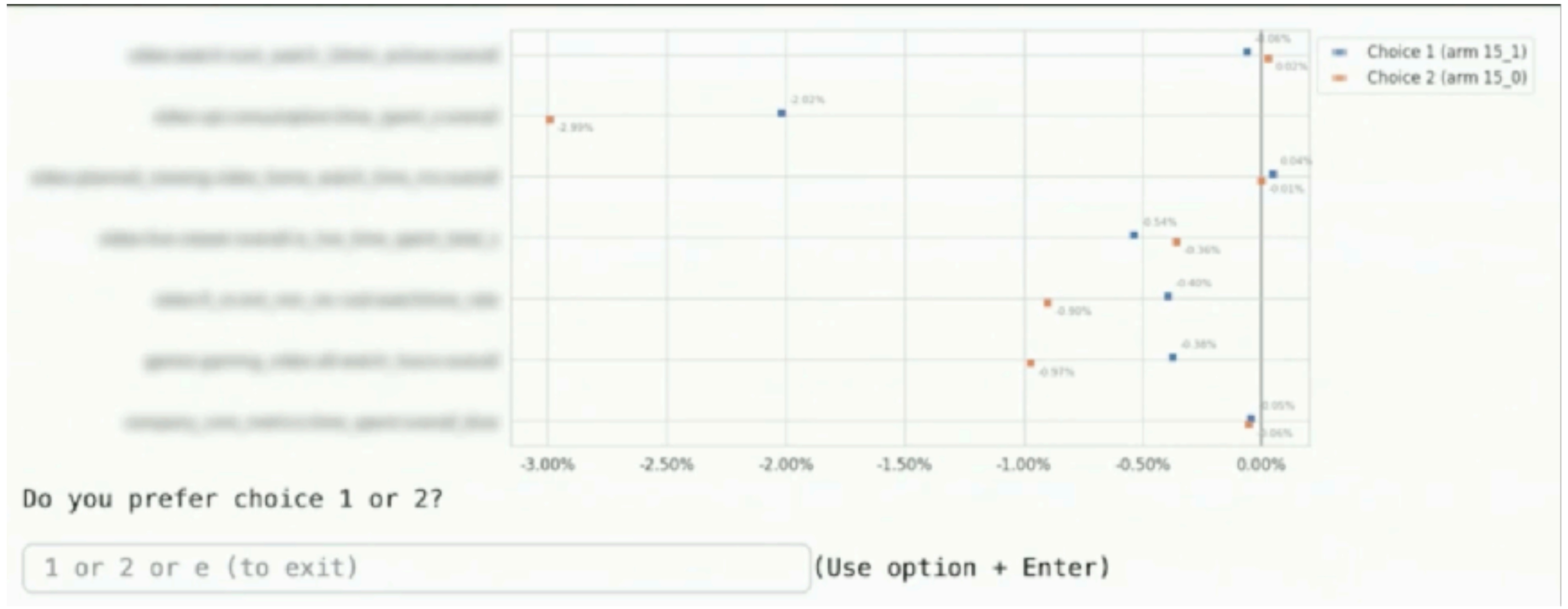
Preference-based methods

- Sometimes, we have many objectives and it is hard to know exactly how they should be prioritised

- Preference based methods ask users to provide feedback on which multiobjective solution is better

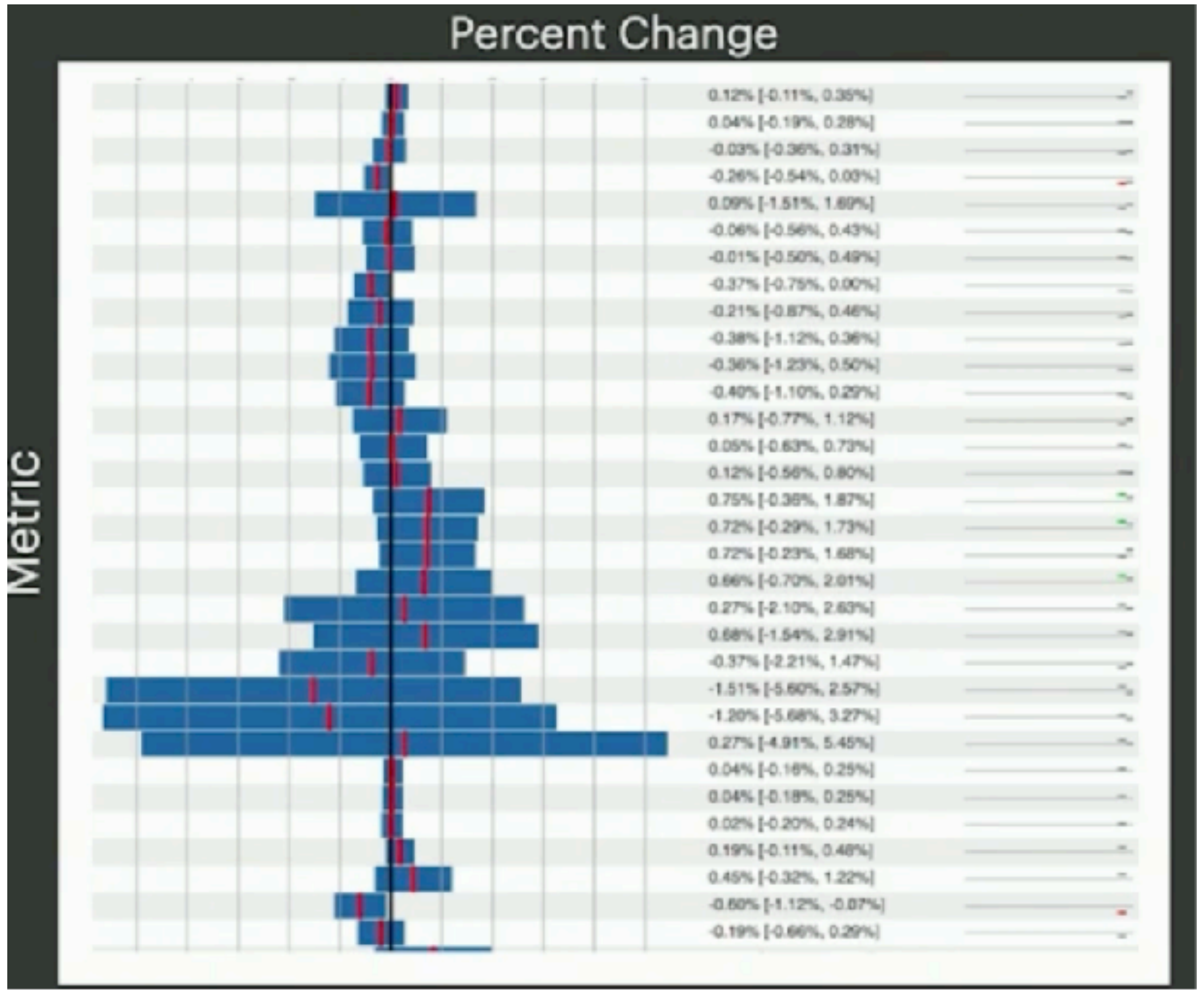


An AB test for many metrics on Facebook.
Beyond Loss Efficient Optimization of Living Machine Learning. Bakshi Automl 2023

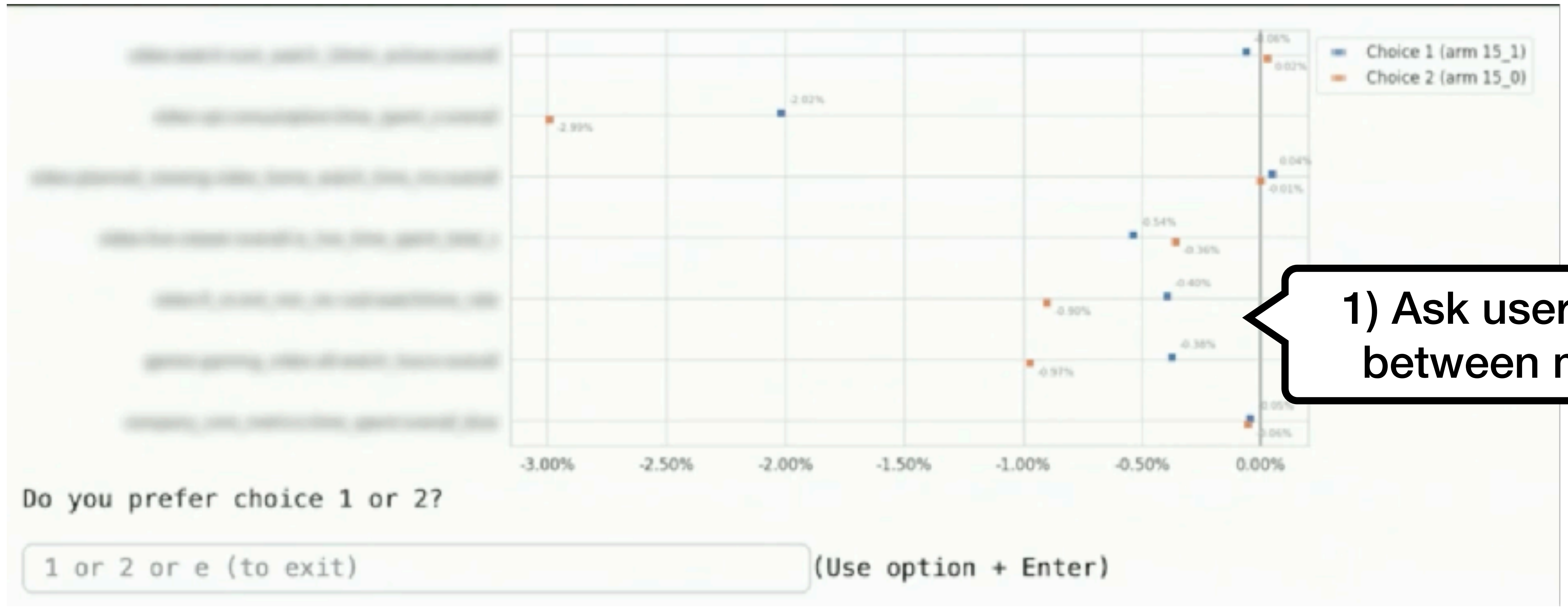


Preference-based methods

- Sometimes, we have many objectives and it is hard to know exactly how they should be prioritised
- Preference based methods ask users to provide feedback on which multiobjective solution is better



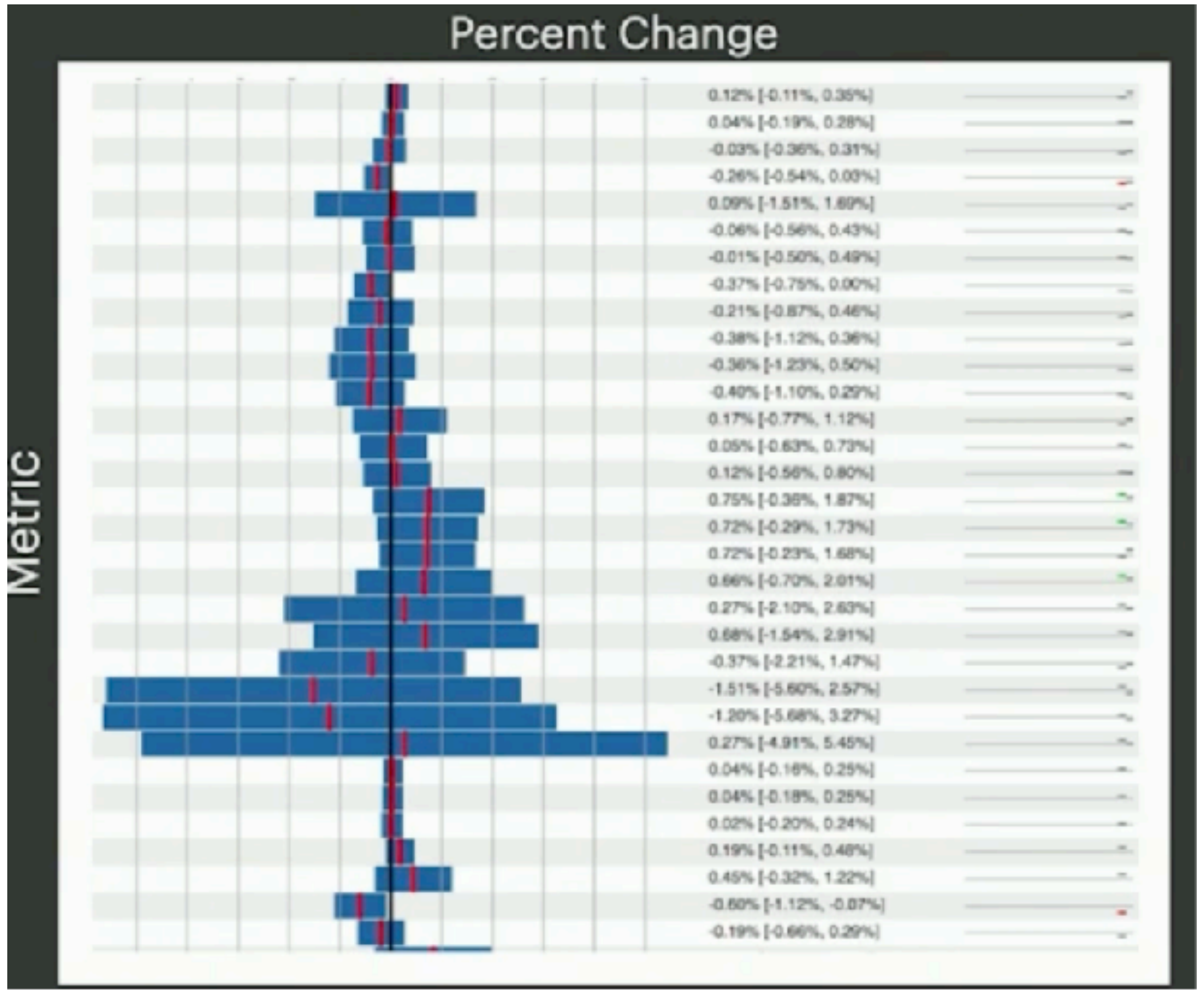
An AB test for many metrics on Facebook.
Beyond Loss Efficient Optimization of Living Machine Learning. Bakshi Automl 2023



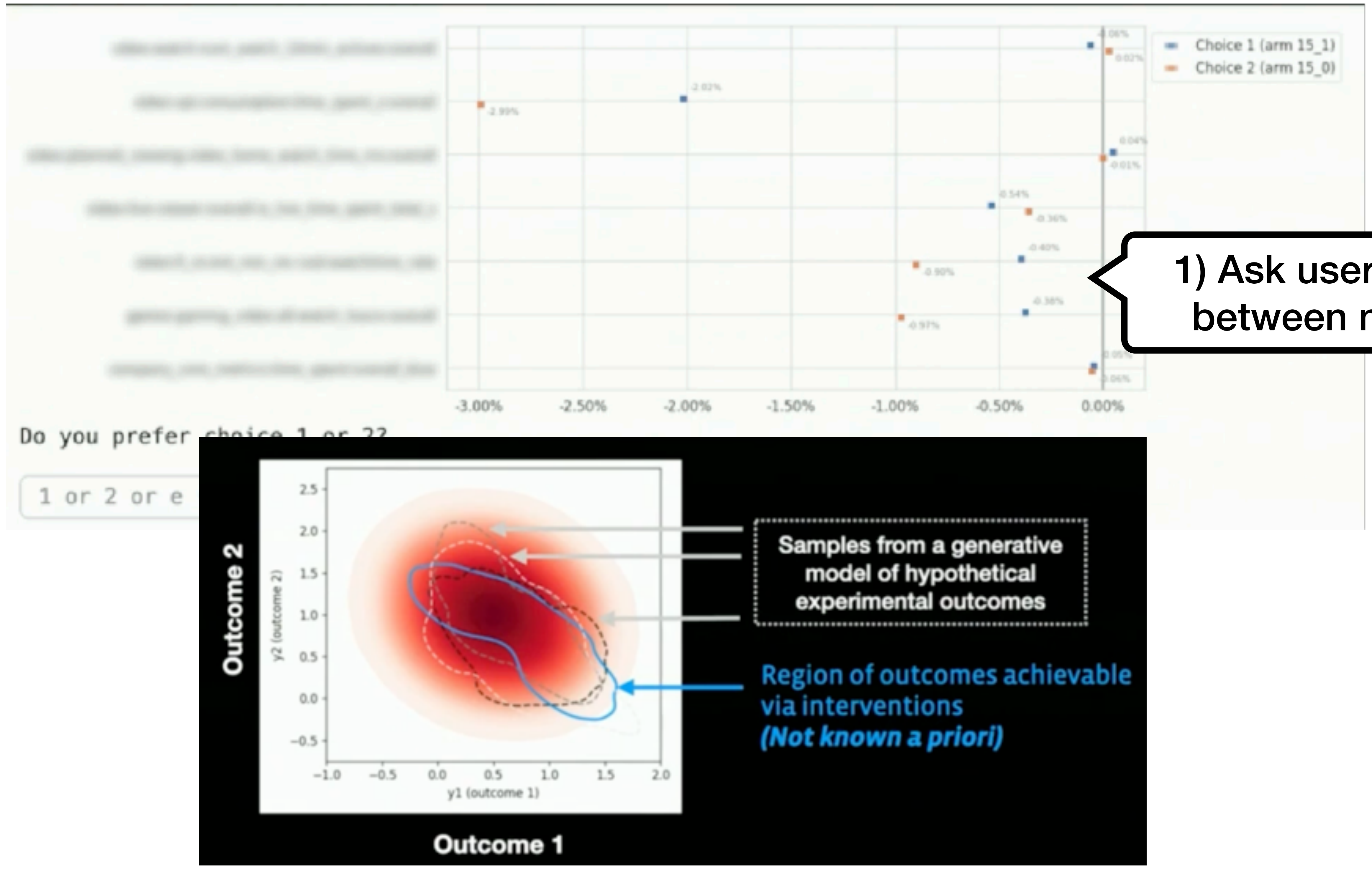
1) Ask user their preference between multiple choices

Preference-based methods

- Sometimes, we have many objectives and it is hard to know exactly how they should be prioritised
- Preference based methods ask users to provide feedback on which multiobjective solution is better



An AB test for many metrics on Facebook.
Beyond Loss Efficient Optimization of Living Machine Learning. Bakshi Automl 2023

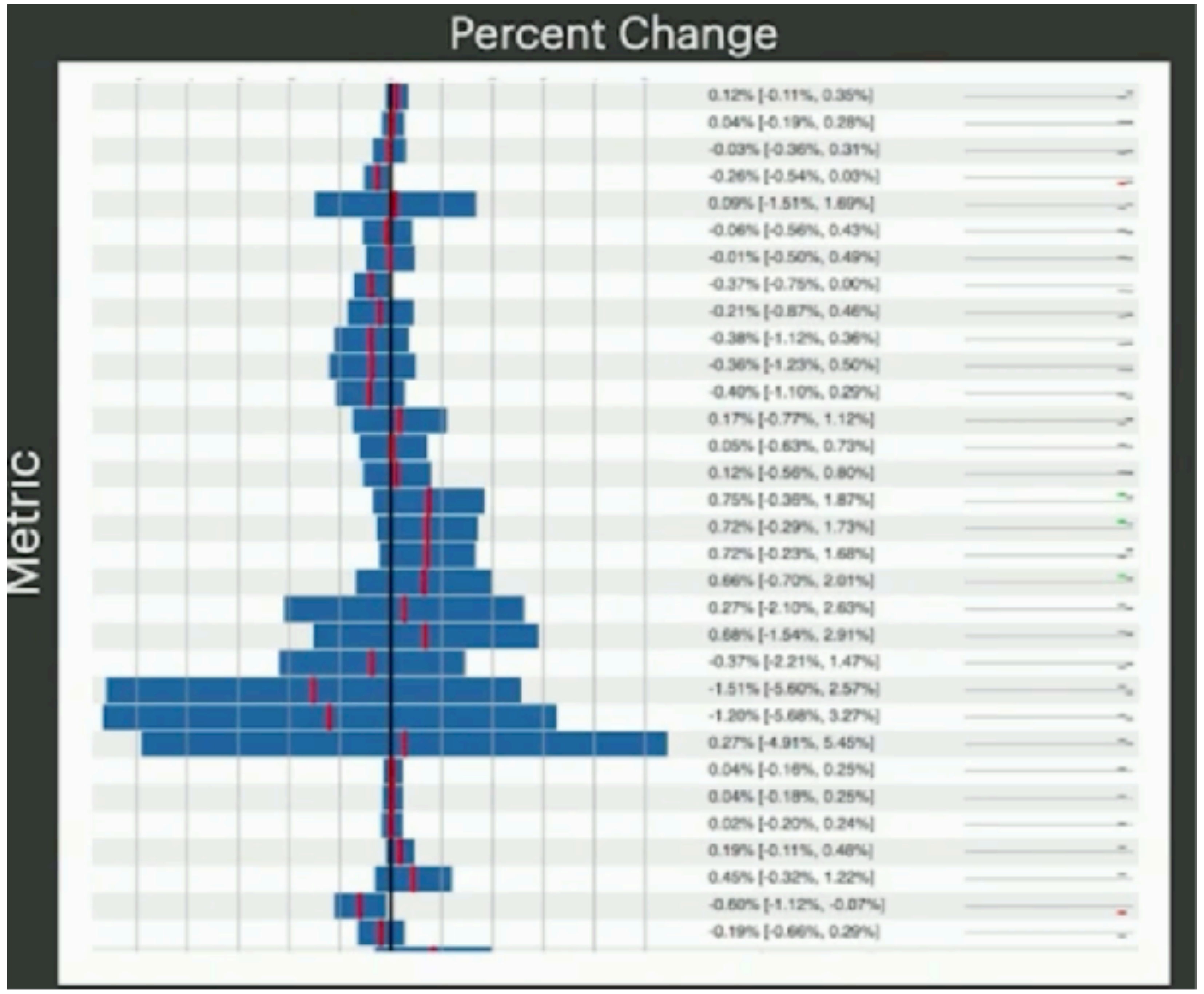


1) Ask user their preference between multiple choices

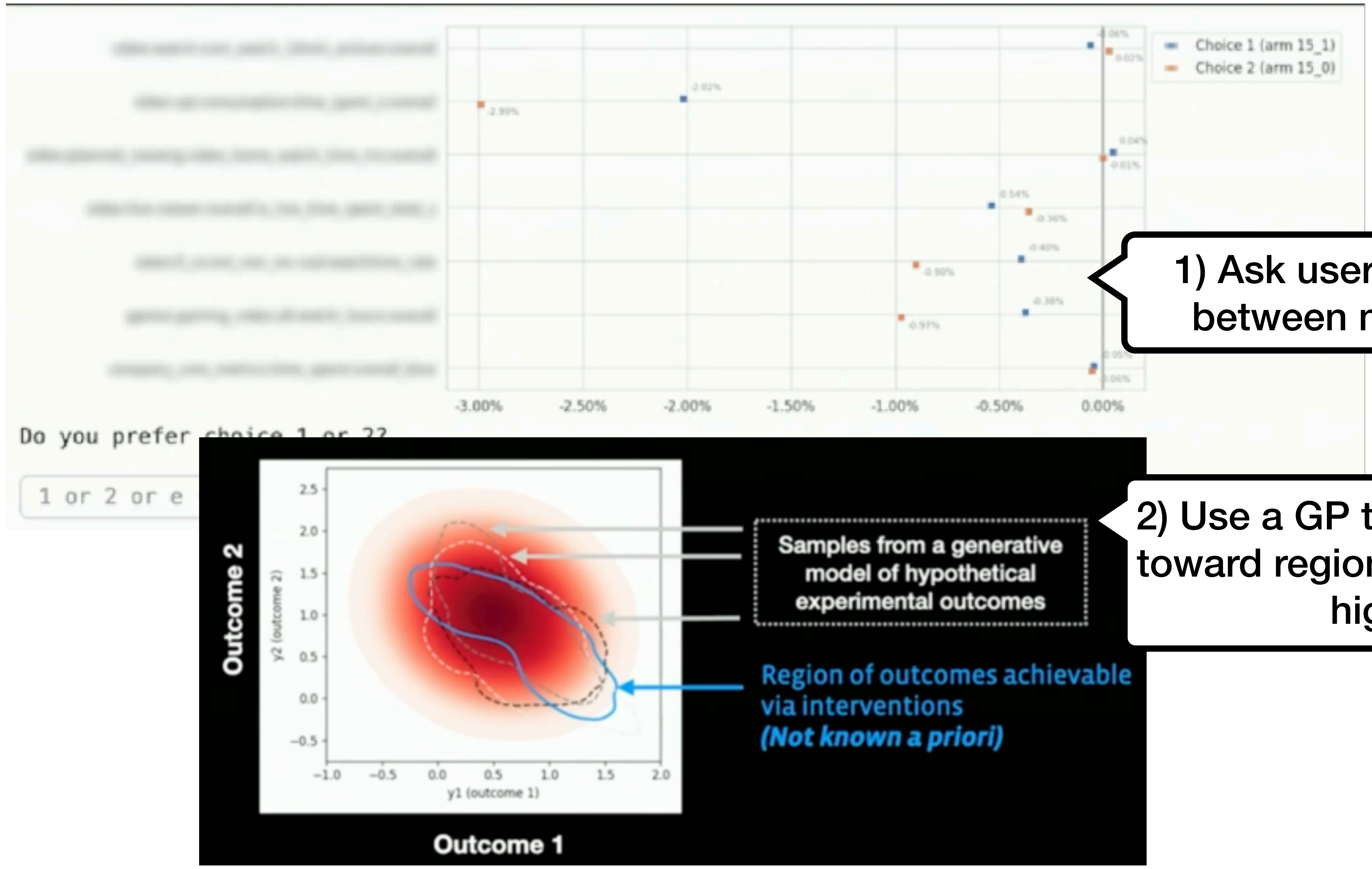
Preference-based methods

- Sometimes, we have many objectives and it is hard to know exactly how they should be prioritised

- Preference based methods ask users to provide feedback on which multiobjective solution is better



An AB test for many metrics on Facebook.
Beyond Loss Efficient Optimization of Living Machine Learning. Bakshi Automl 2023



1) Ask user their preference between multiple choices

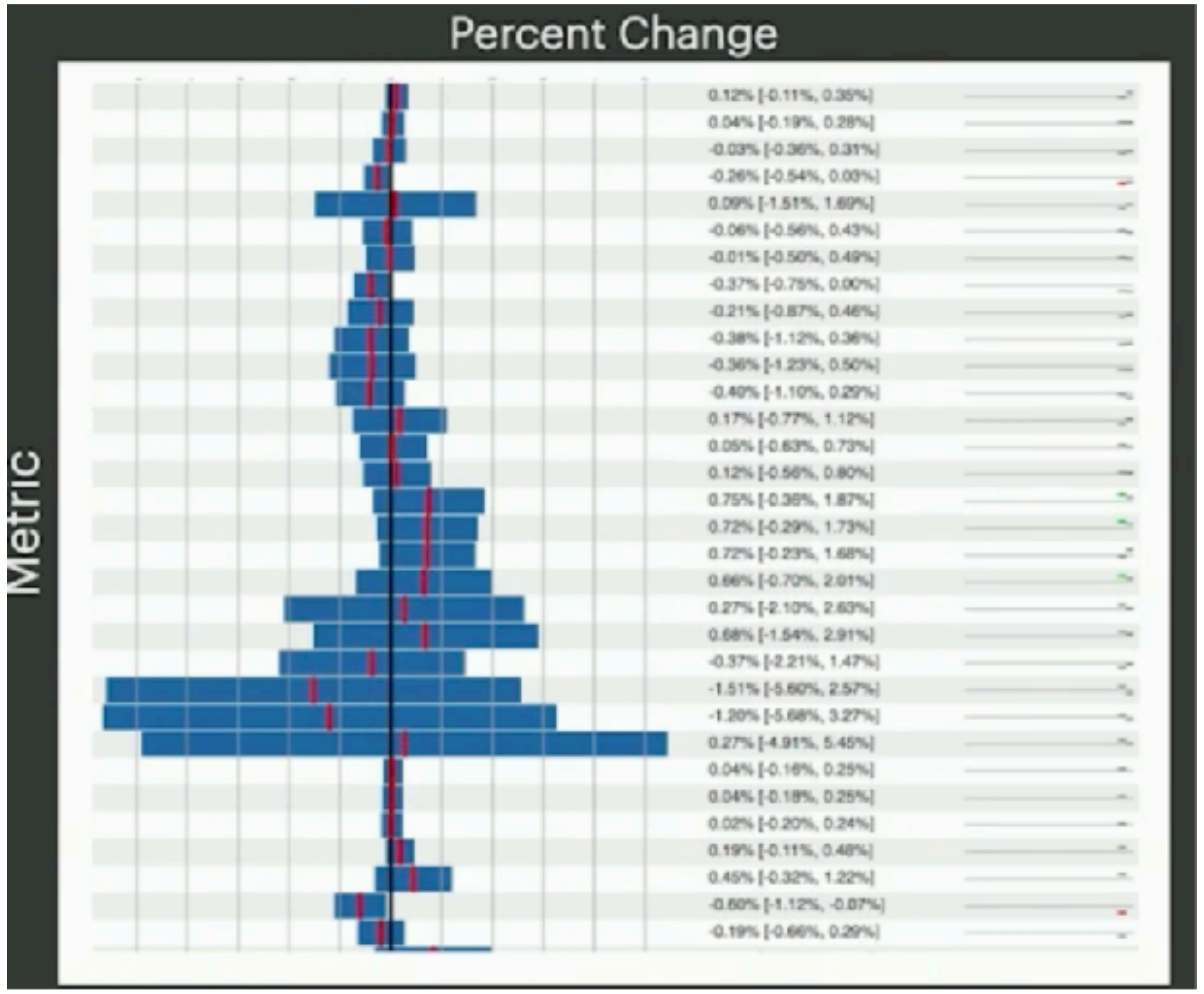
2) Use a GP to guide the search toward region of the space with high utility

Preference-based methods

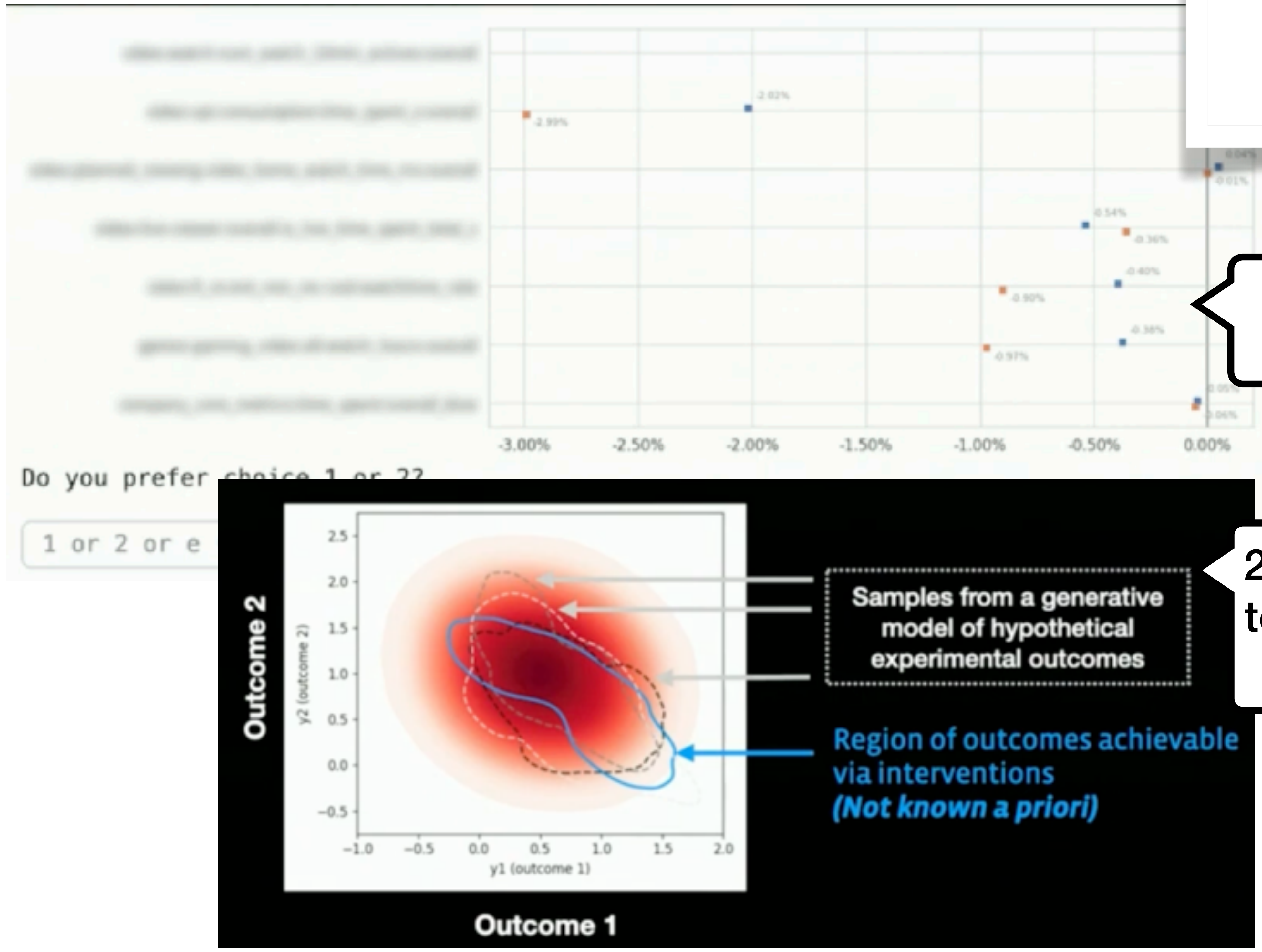
- Sometimes, we have many objectives and it is hard to know exactly how they should be prioritised

- Preference based methods ask users to provide feedback on which multiobjective solution is better

Preference-learning is also used in RLHF to fine tune LLMs so that we can talk with them!



An AB test for many metrics on Facebook. *Beyond Loss Efficient Optimization of Living Machine Learning*. Bakshi Automl 2023



1) Ask user their preference between multiple choices

2) Use a GP to guide the search toward region of the space with high utility

Multifidelity

Asynchronous Succesful halving (ASHA)

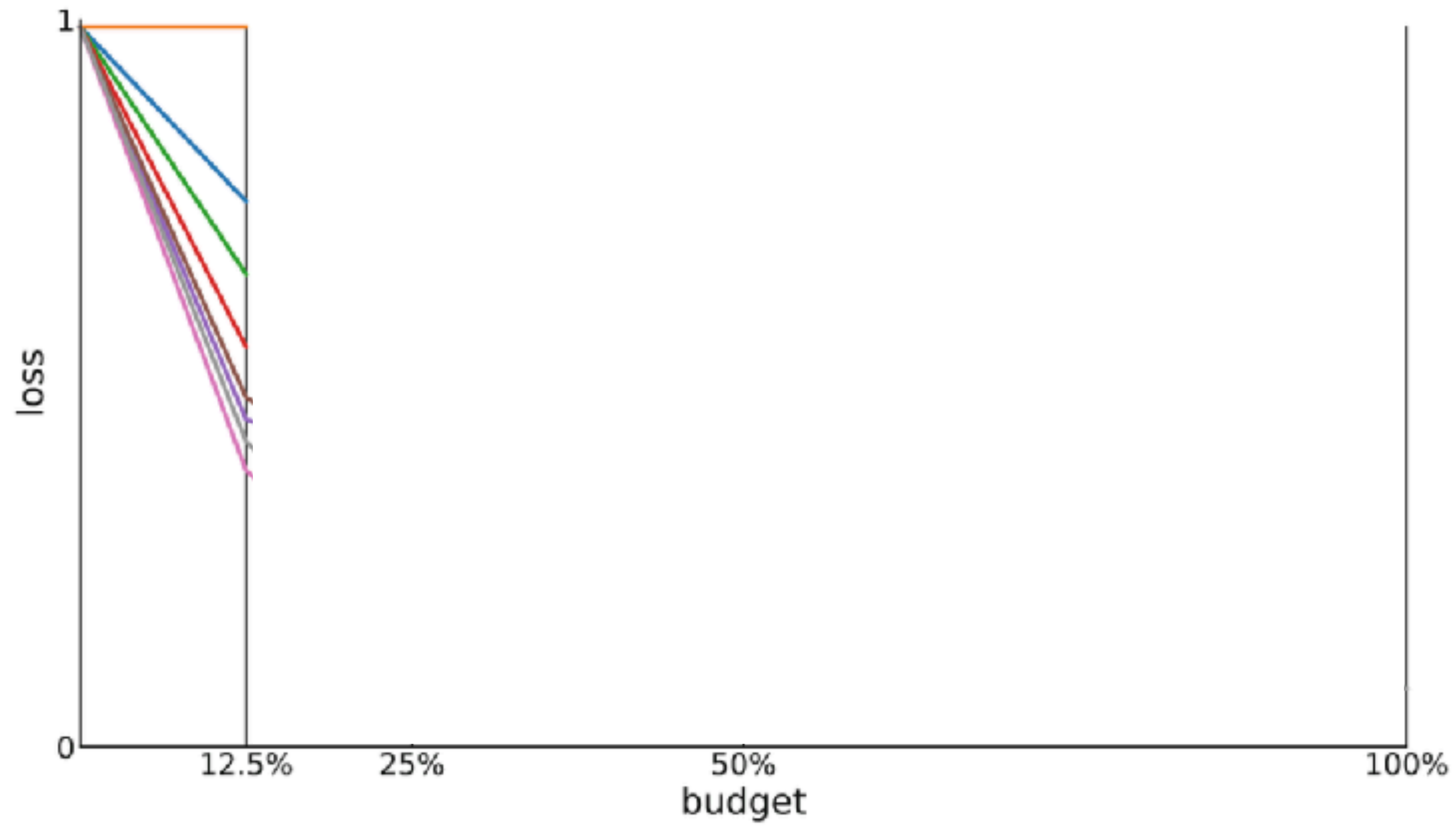


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?

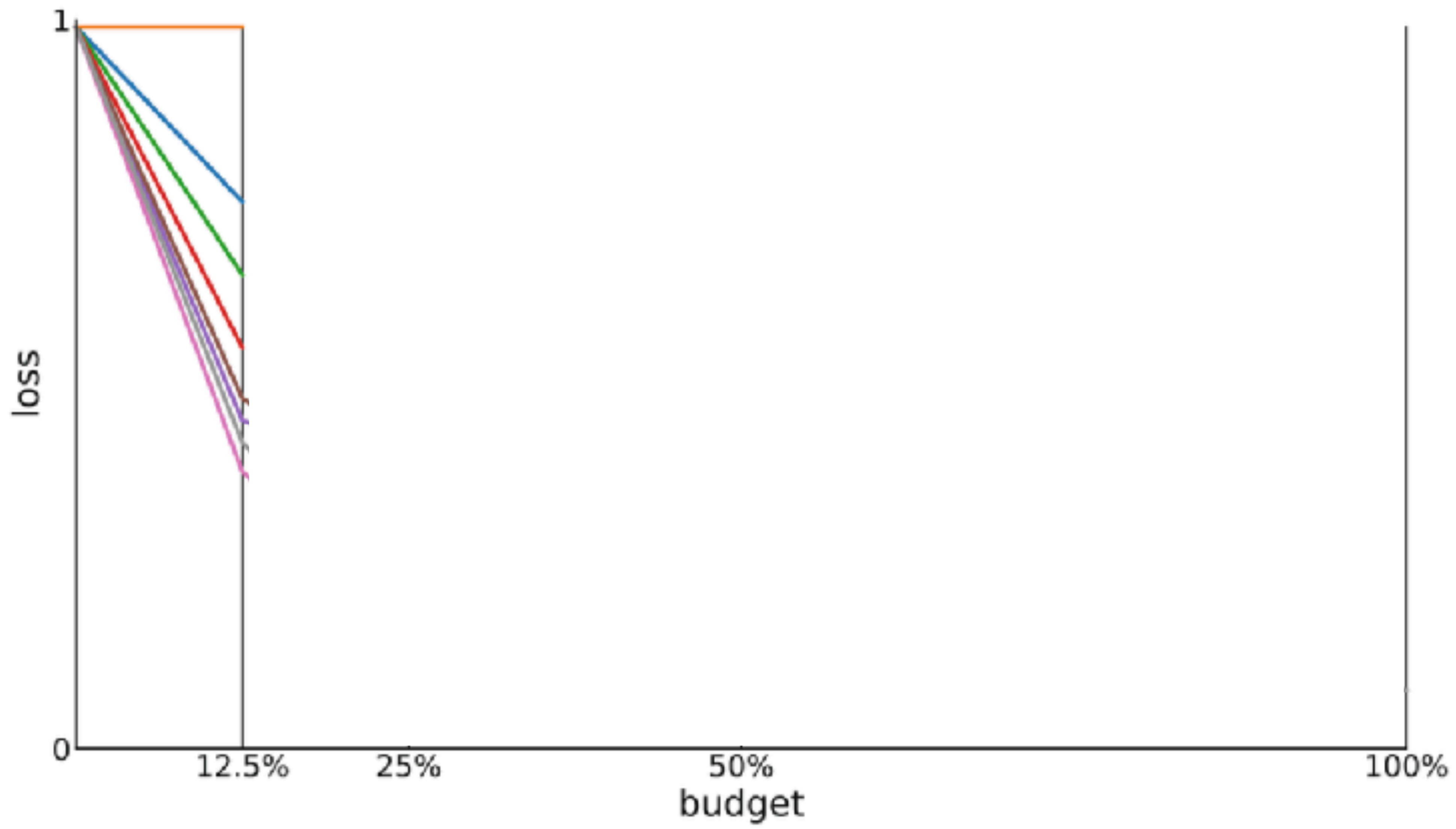


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

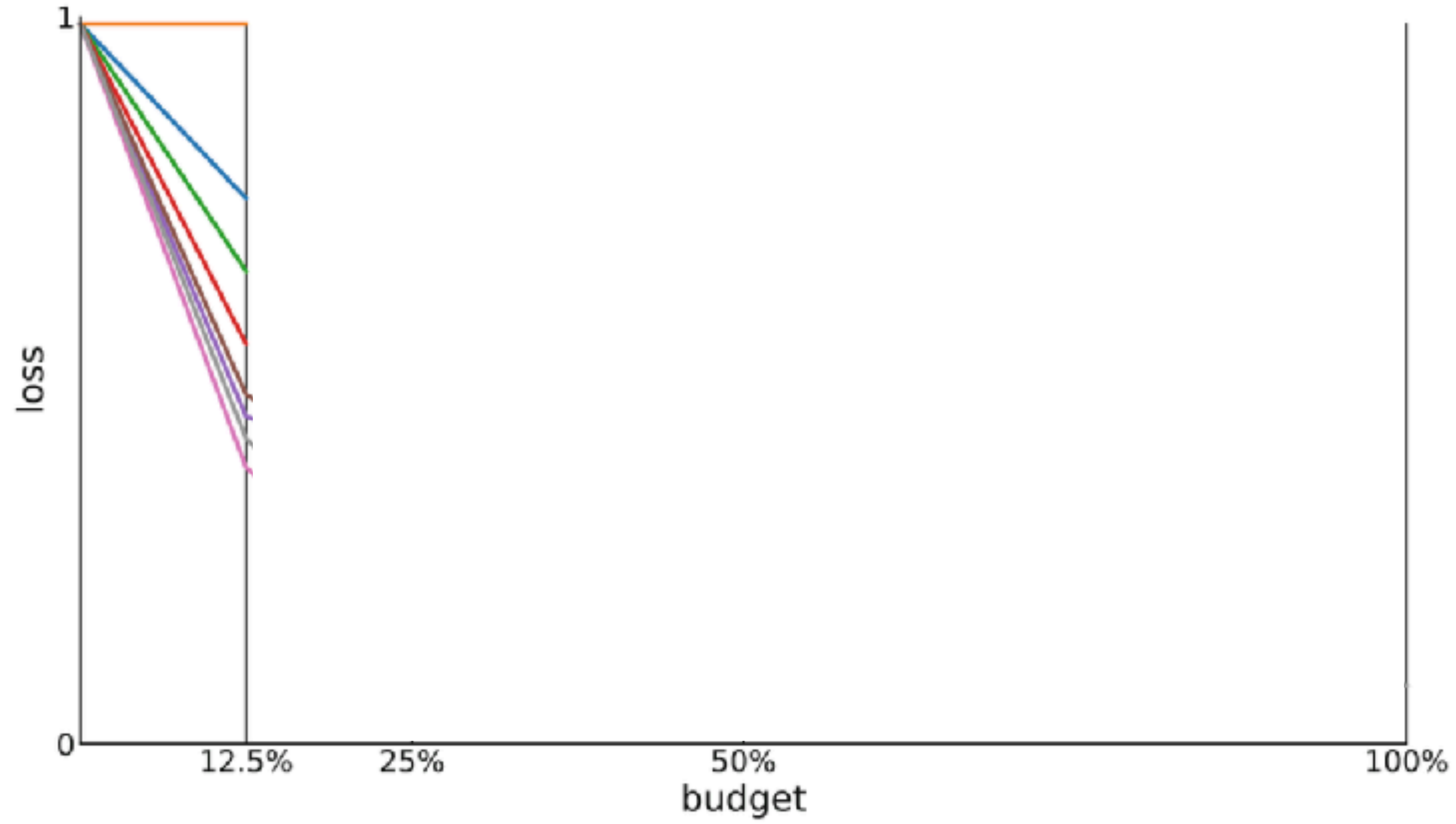


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

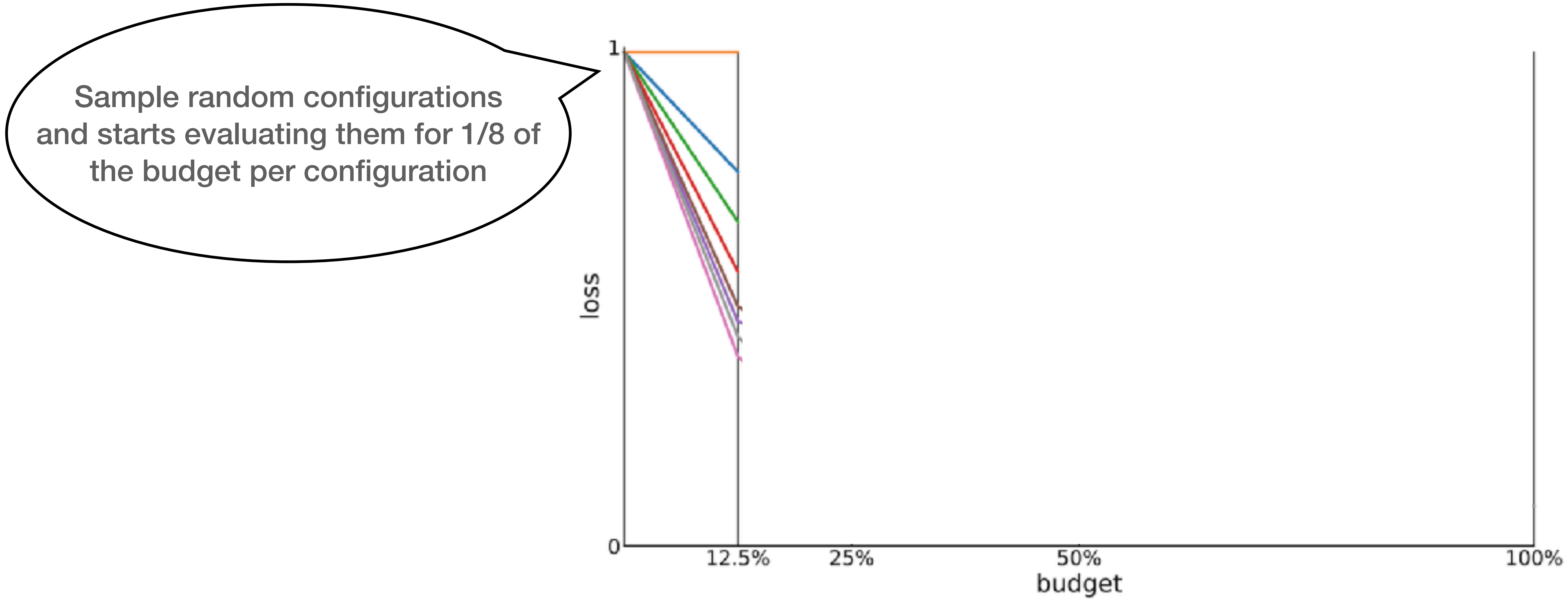


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

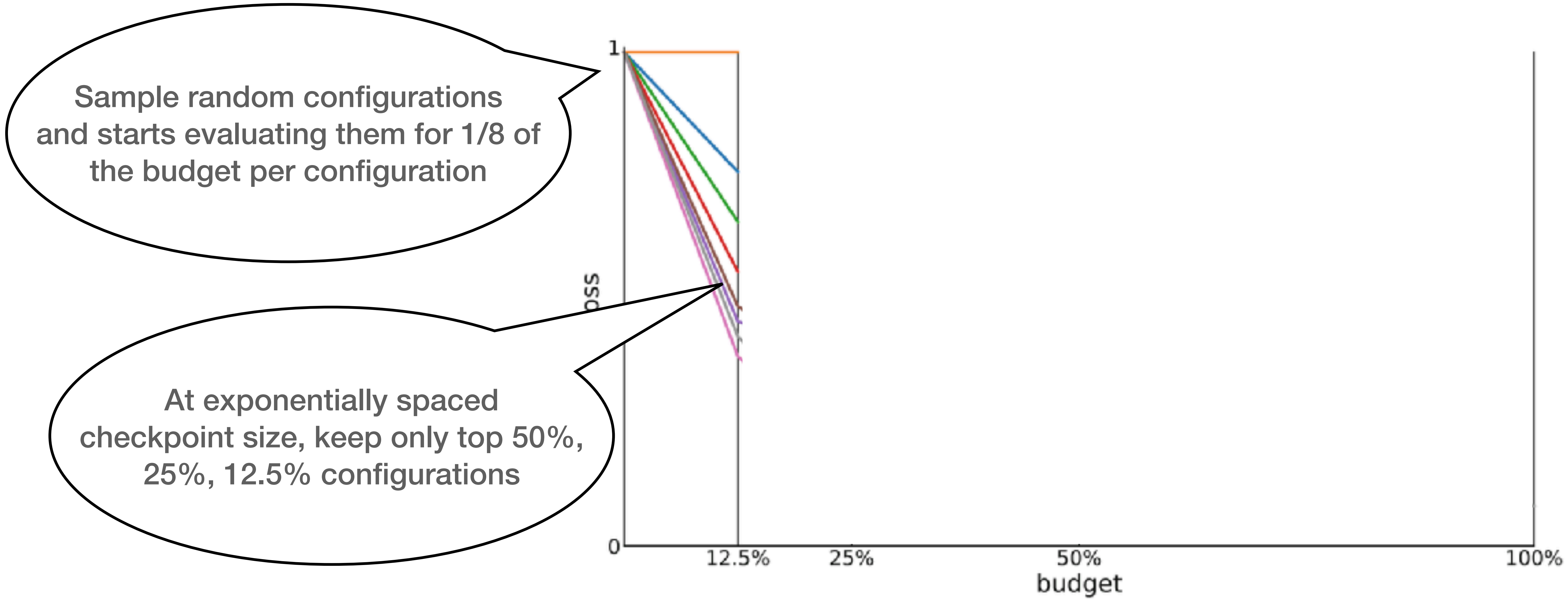


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

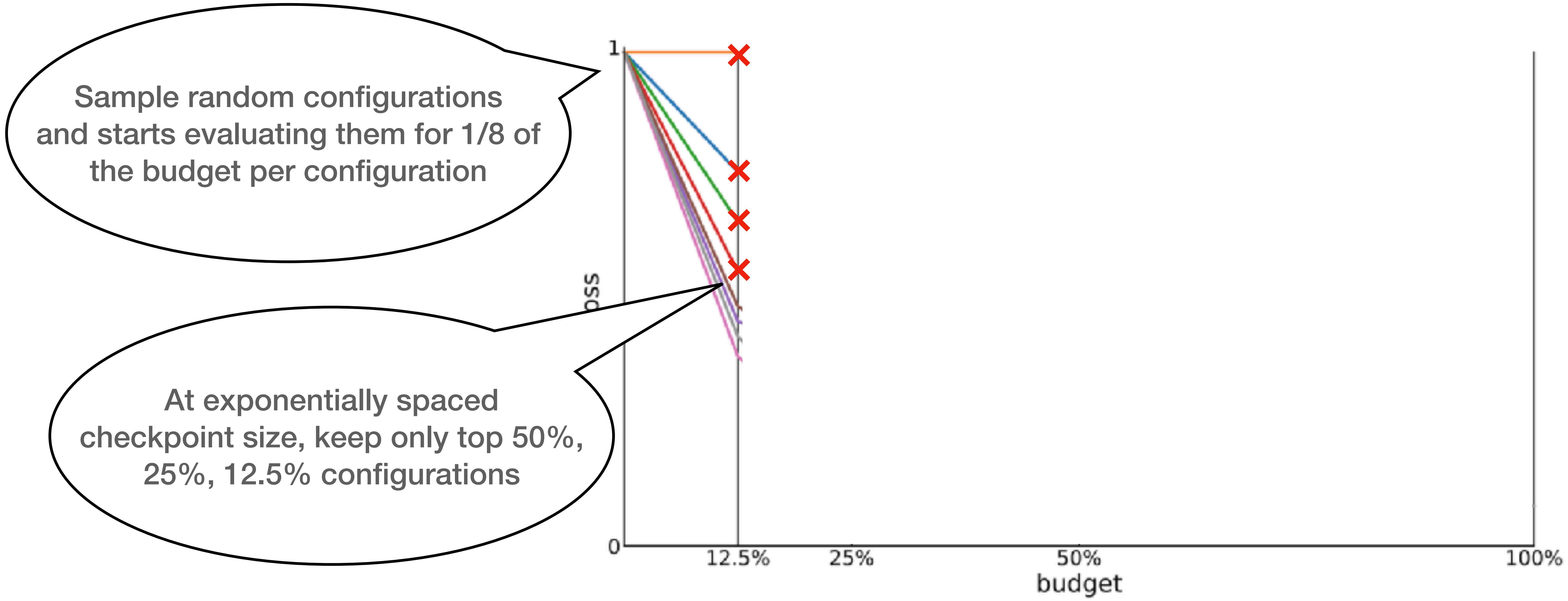


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

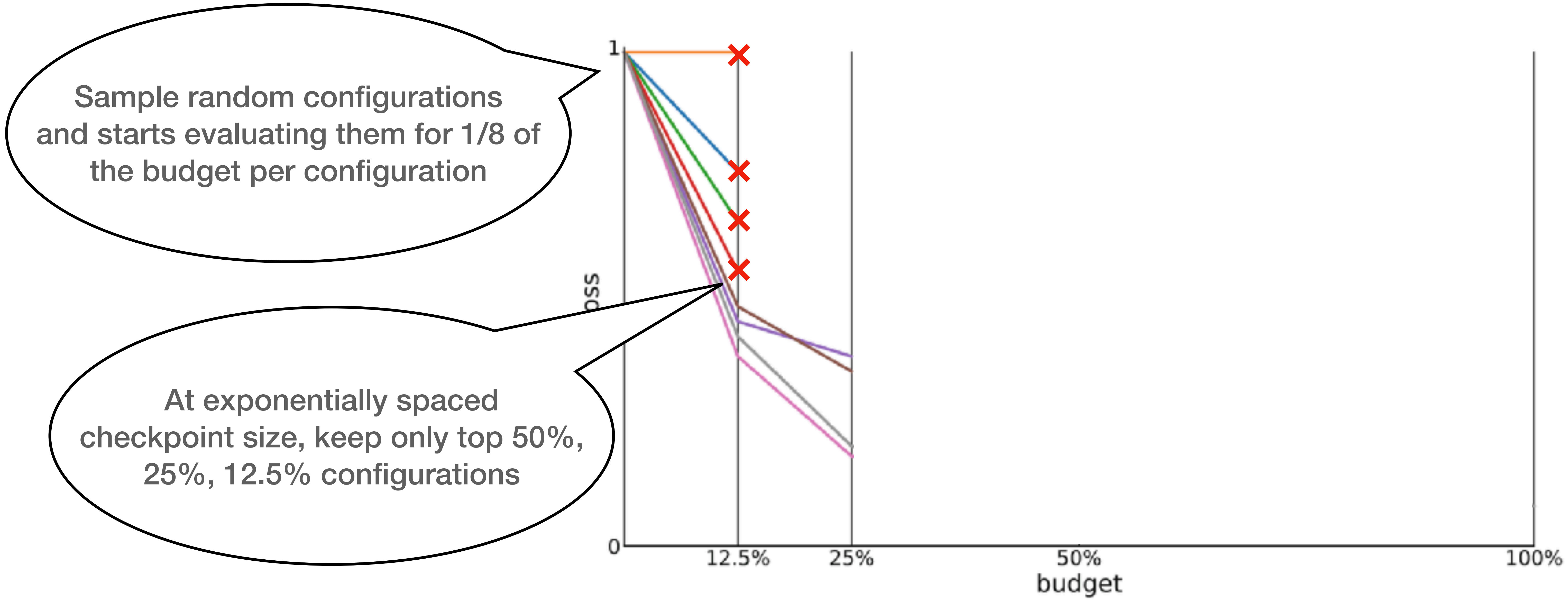


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

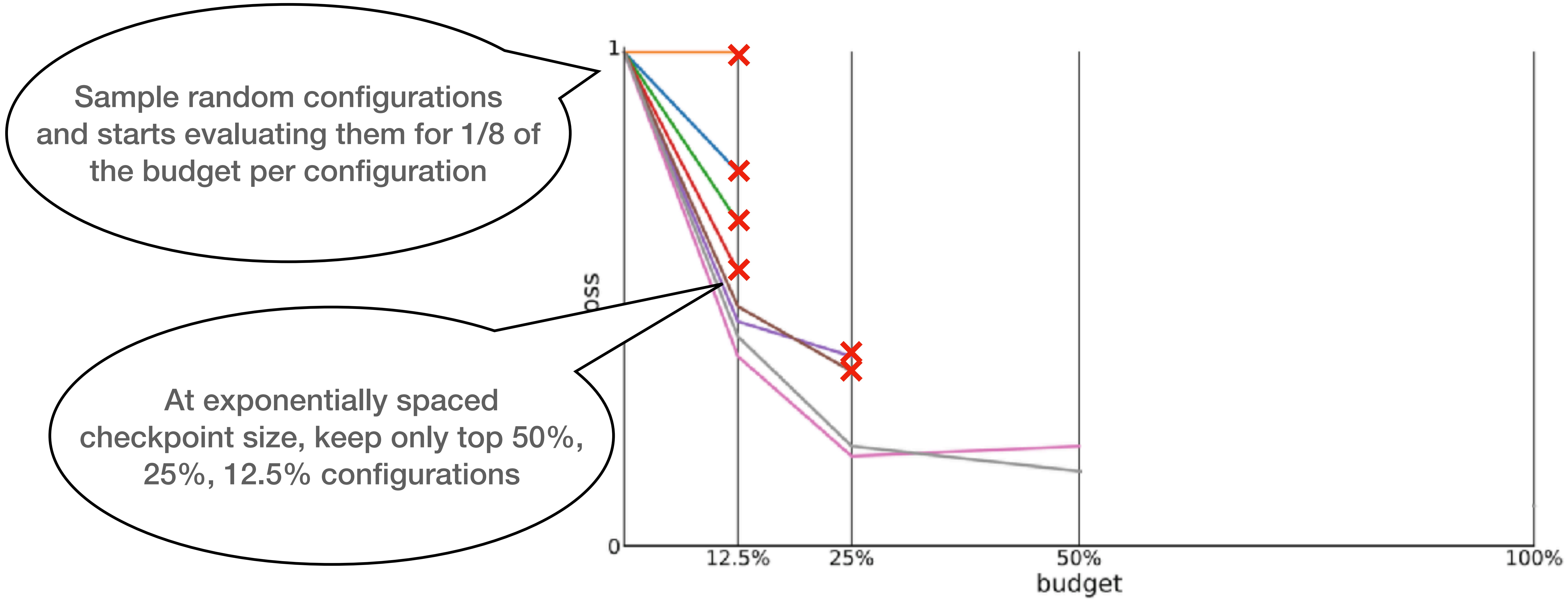


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

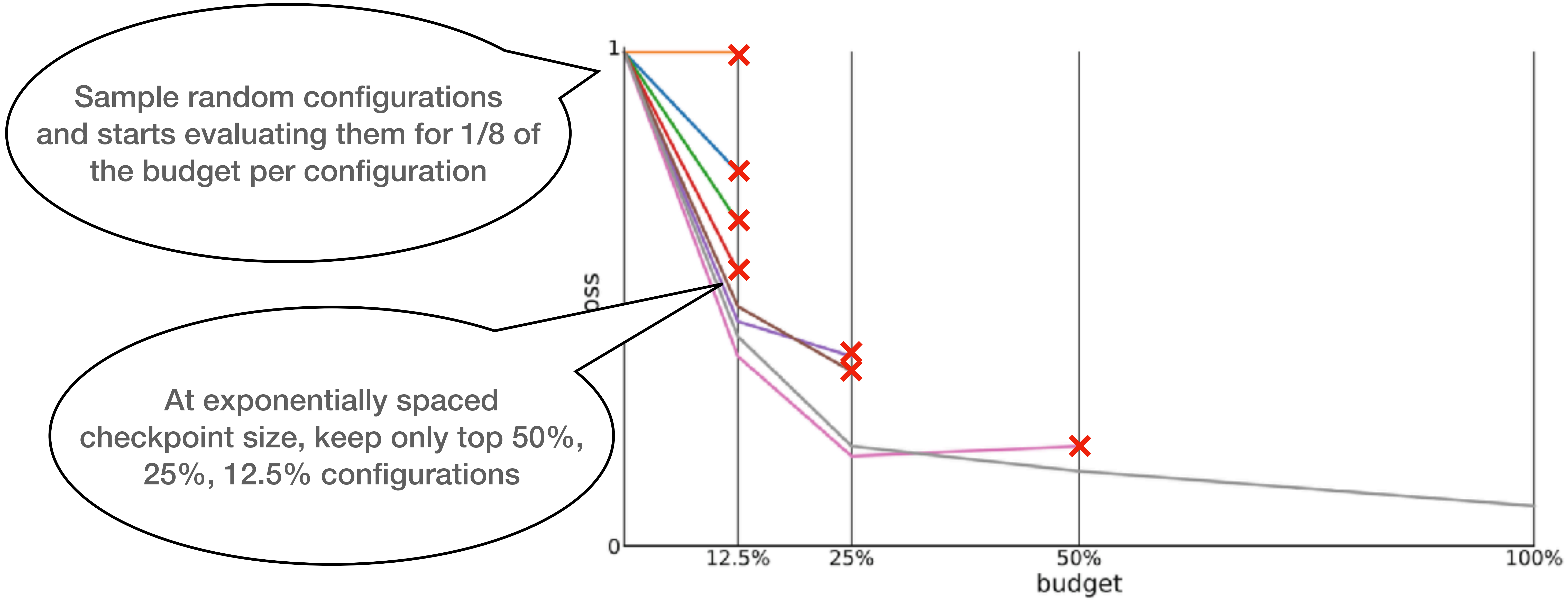


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

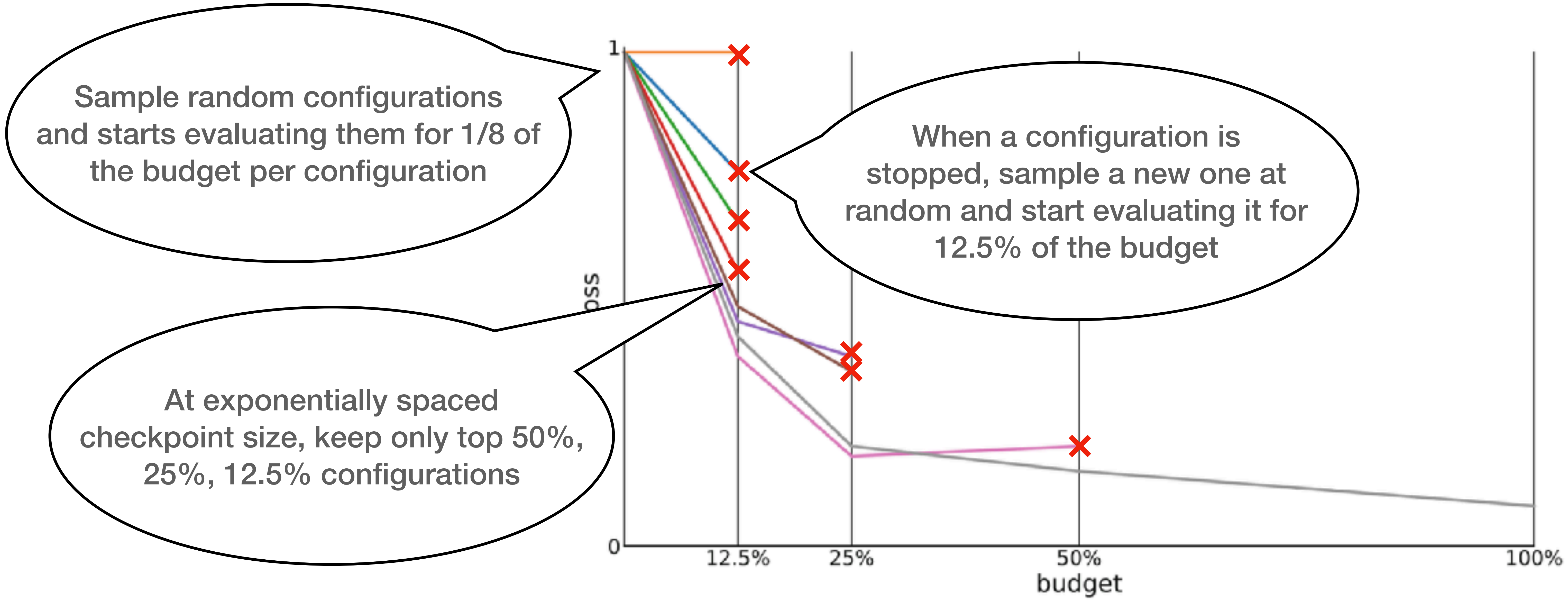


Image credit: Matthias Feurer.

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

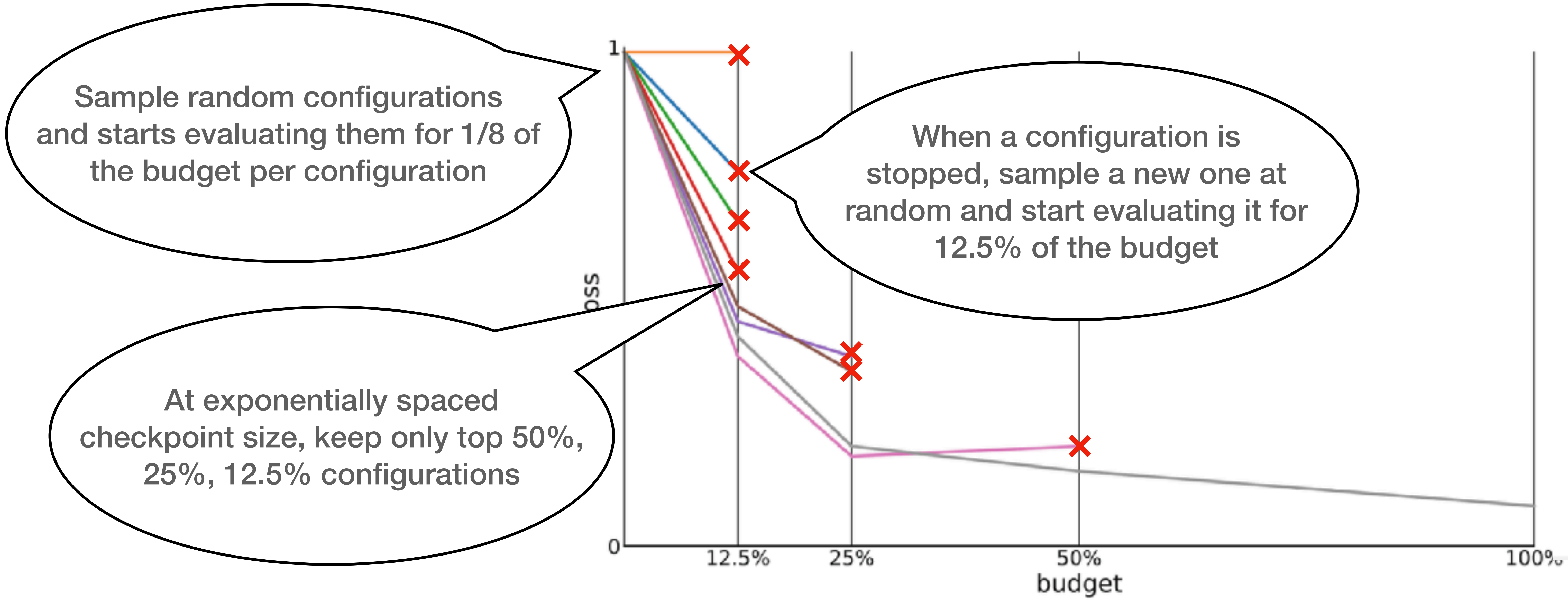
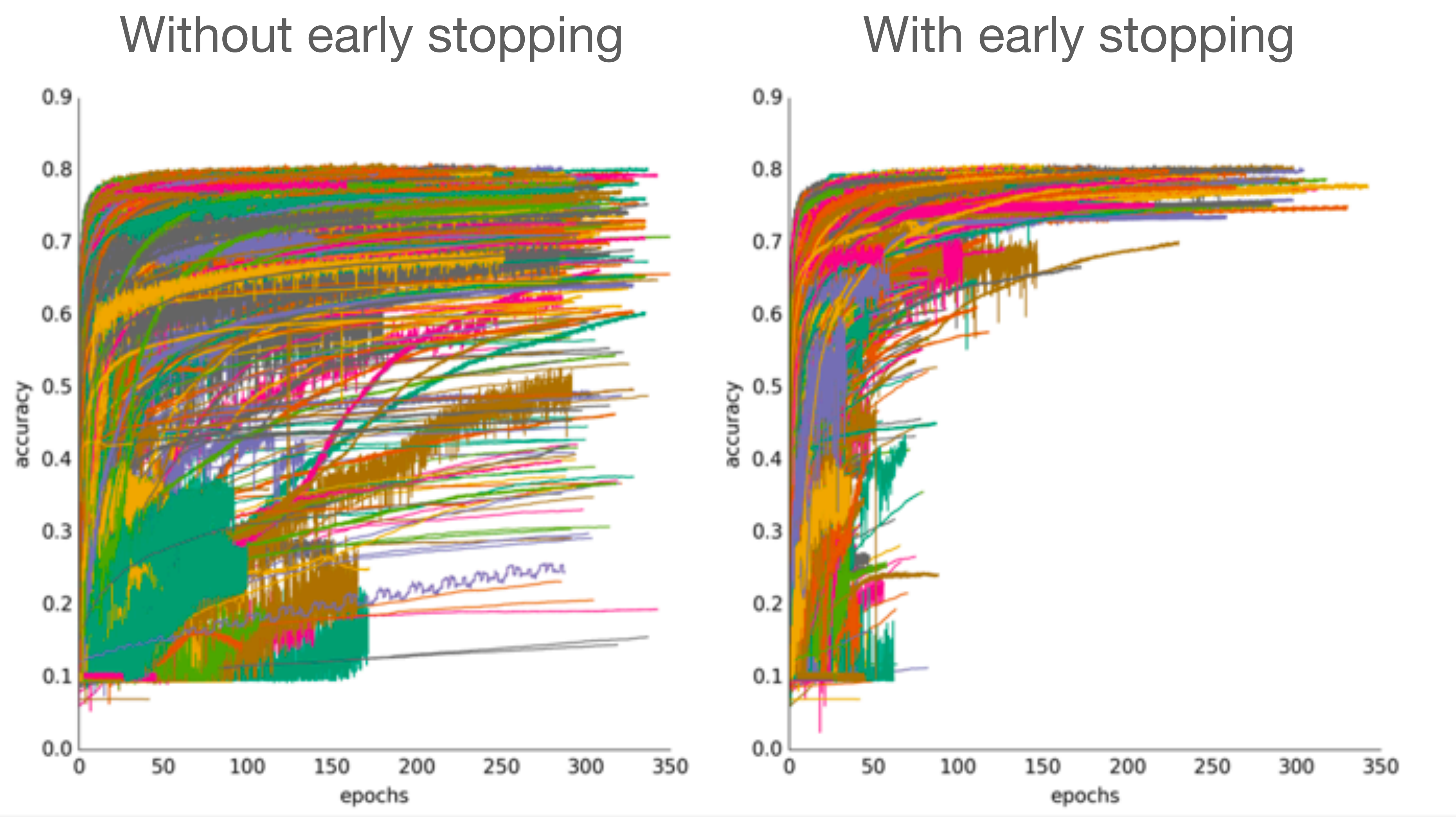


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

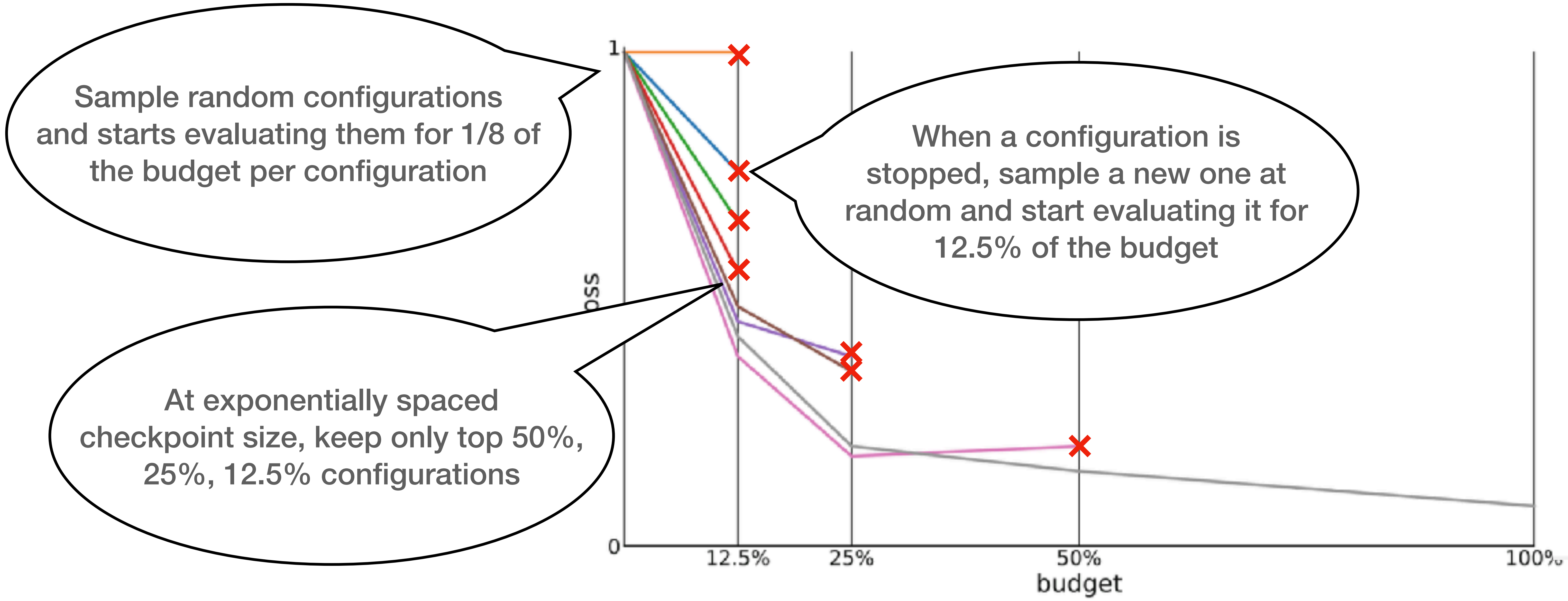
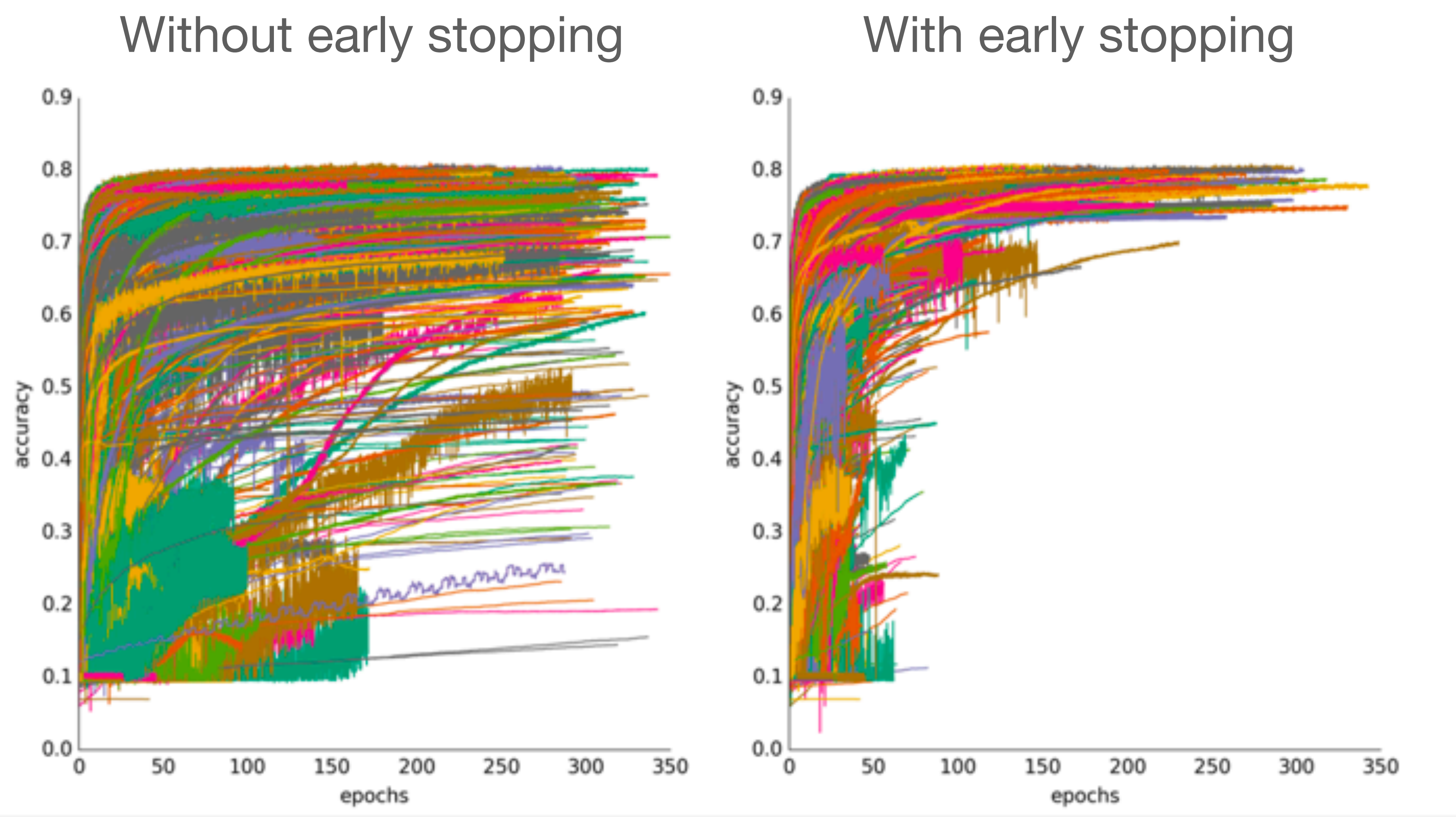


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

- 👍 **Great performance** in practice and one can use multiple workers

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

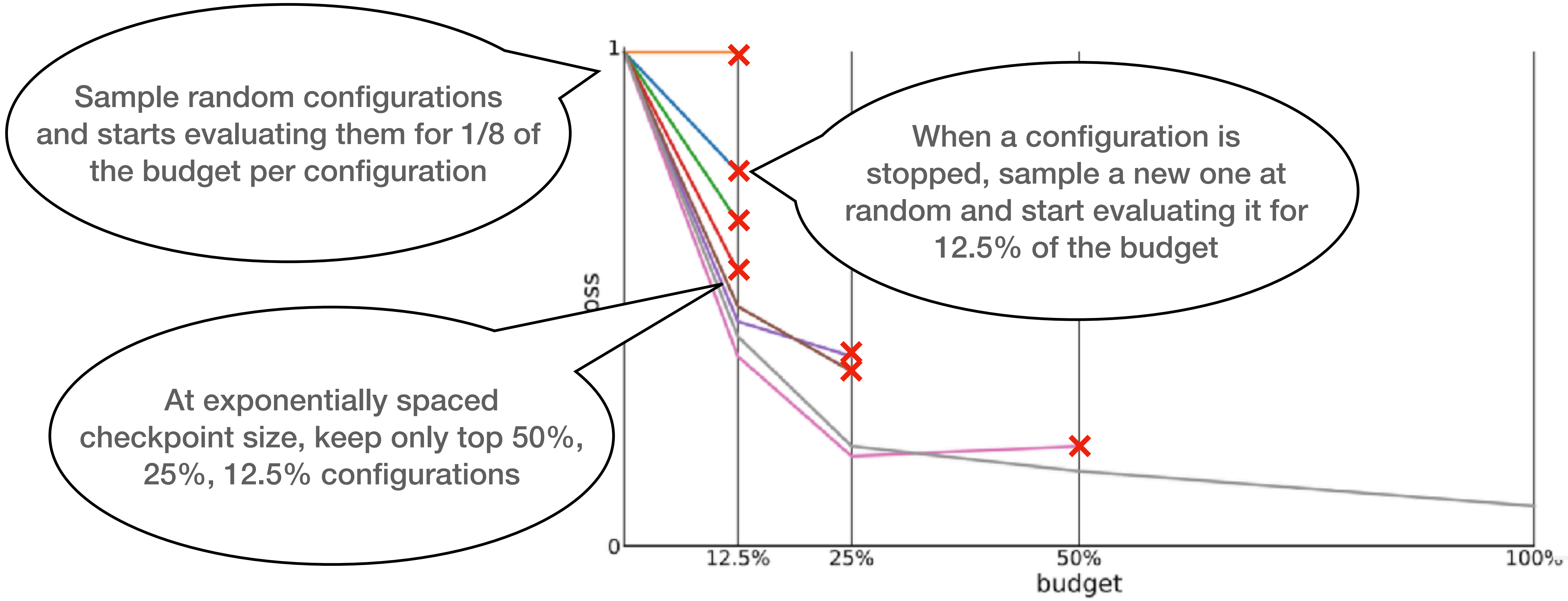
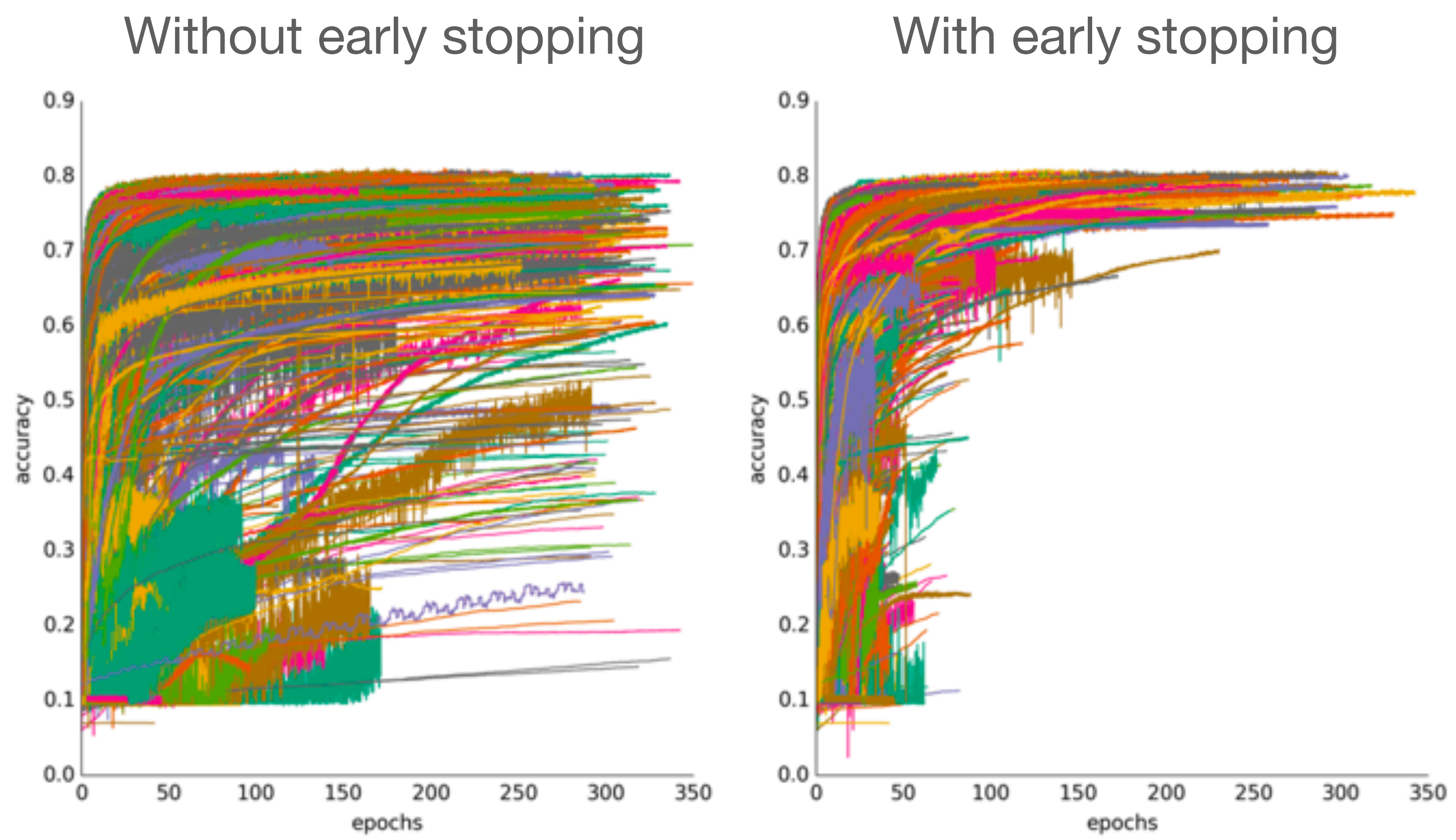


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

- 👍 **Great performance** in practice and one can use multiple workers
- 🤔 How can we extend the algorithm to handle multiple objectives?

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

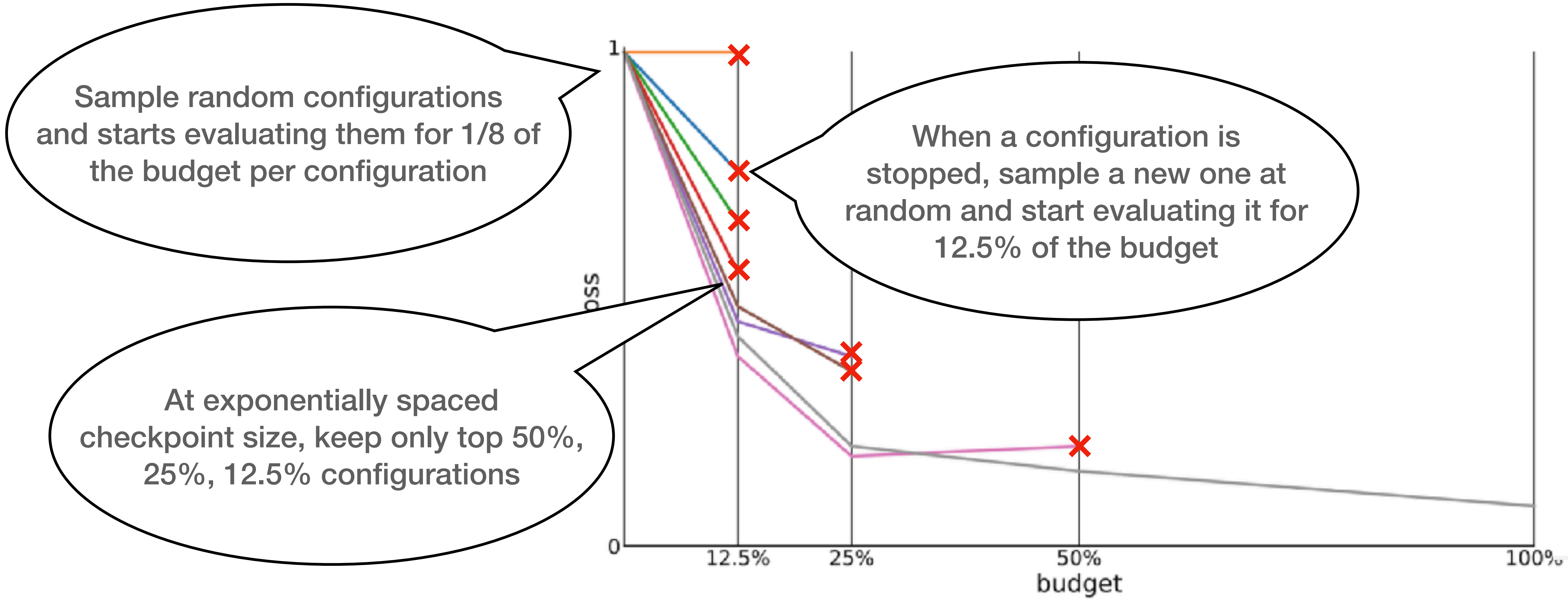
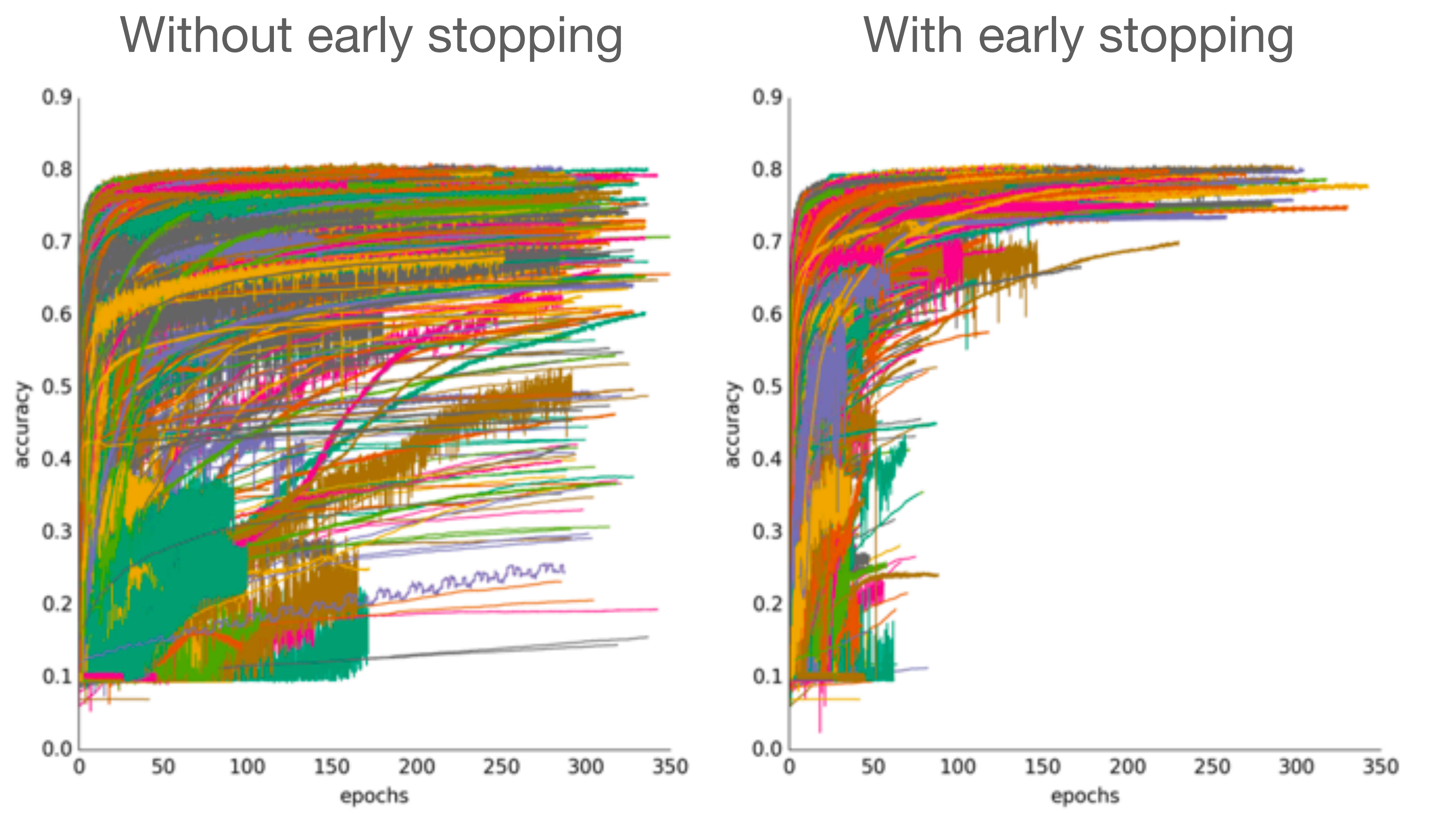


Image credit: Matthias Feurer.



Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

- 👍 **Great performance** in practice and one can use multiple workers
- 🤔 How can we extend the algorithm to handle multiple objectives?
- 🤔 We need to sort to discard the bottom half of configurations, how can we sort if we have multiple objectives?

Multifidelity

Asynchronous Succesful halving (ASHA)

- Typically, models are trained incrementally: can we stop bad configuration early?
- Early results (epochs) can be used to stop bad runs (early stopping)

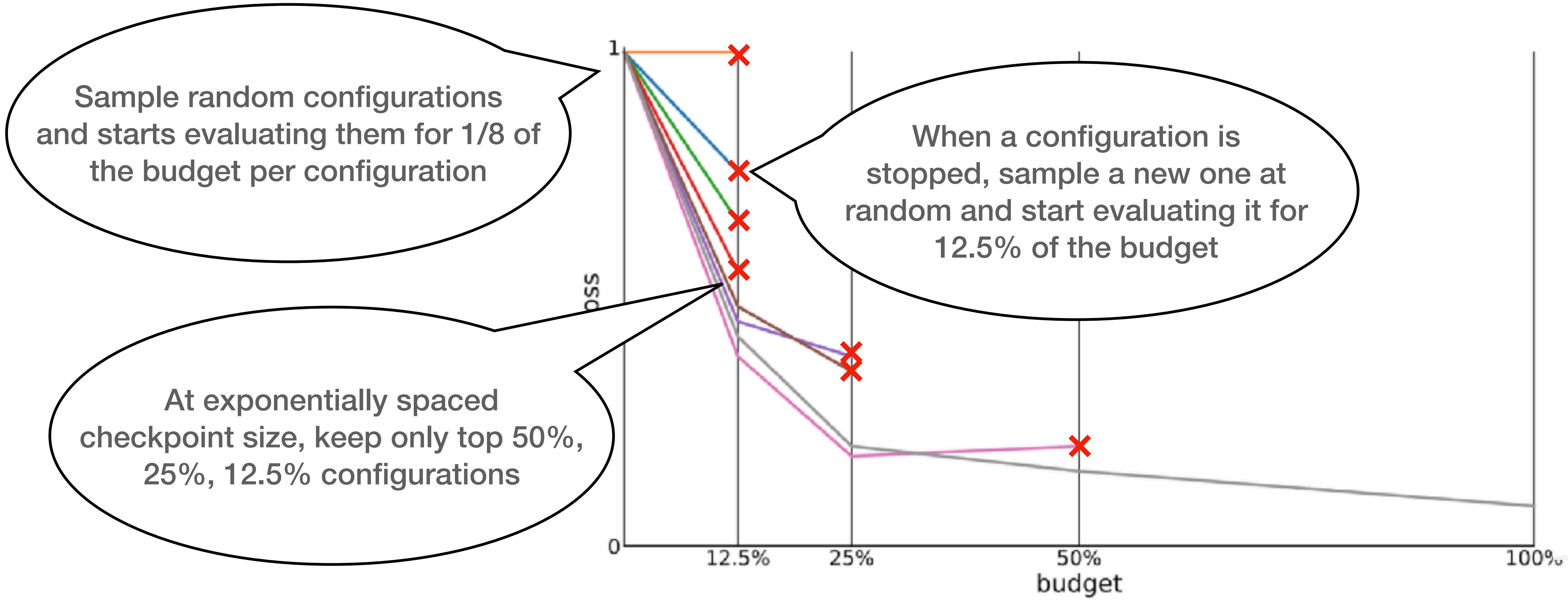
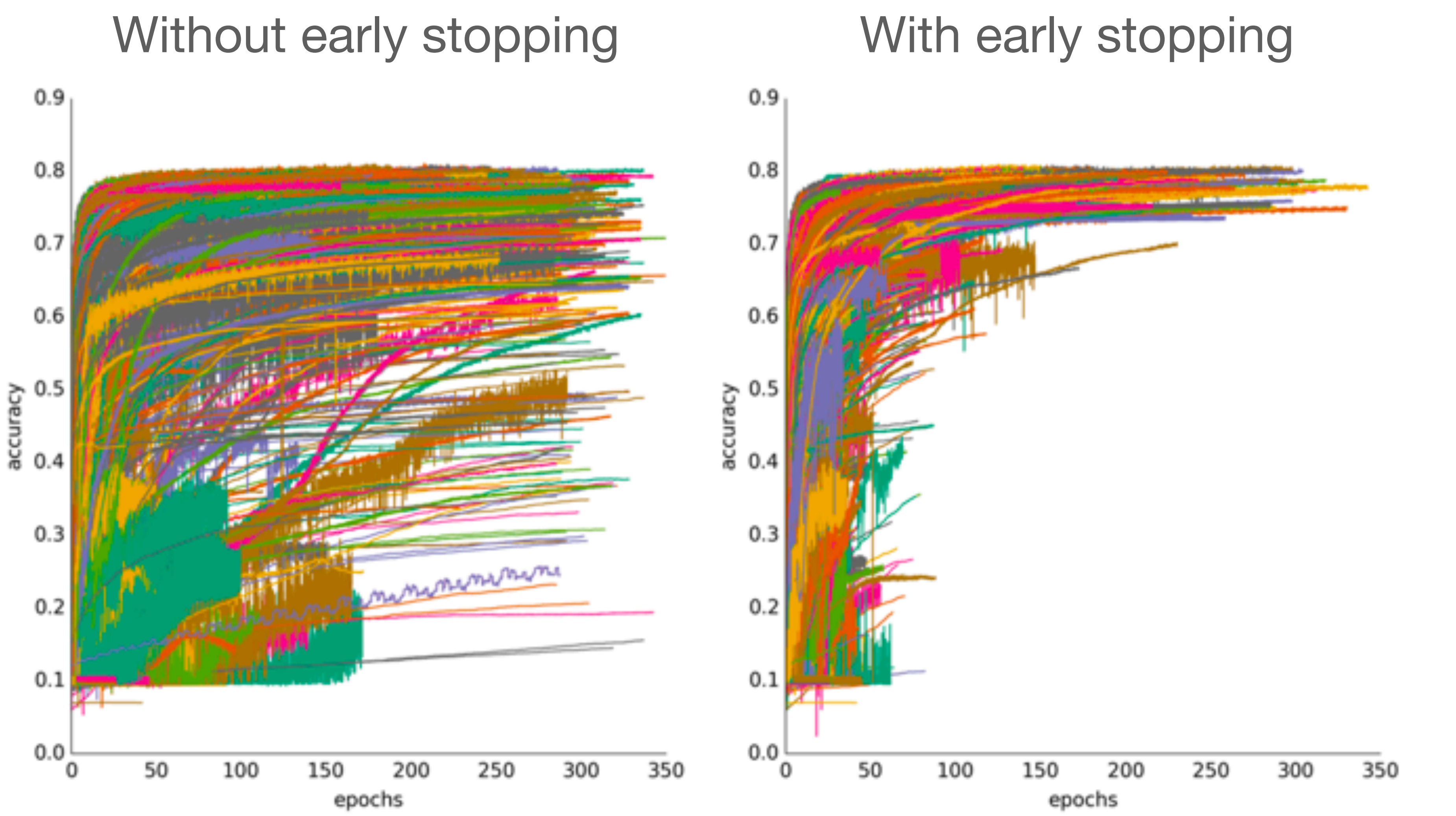


Image credit: Matthias Feurer.

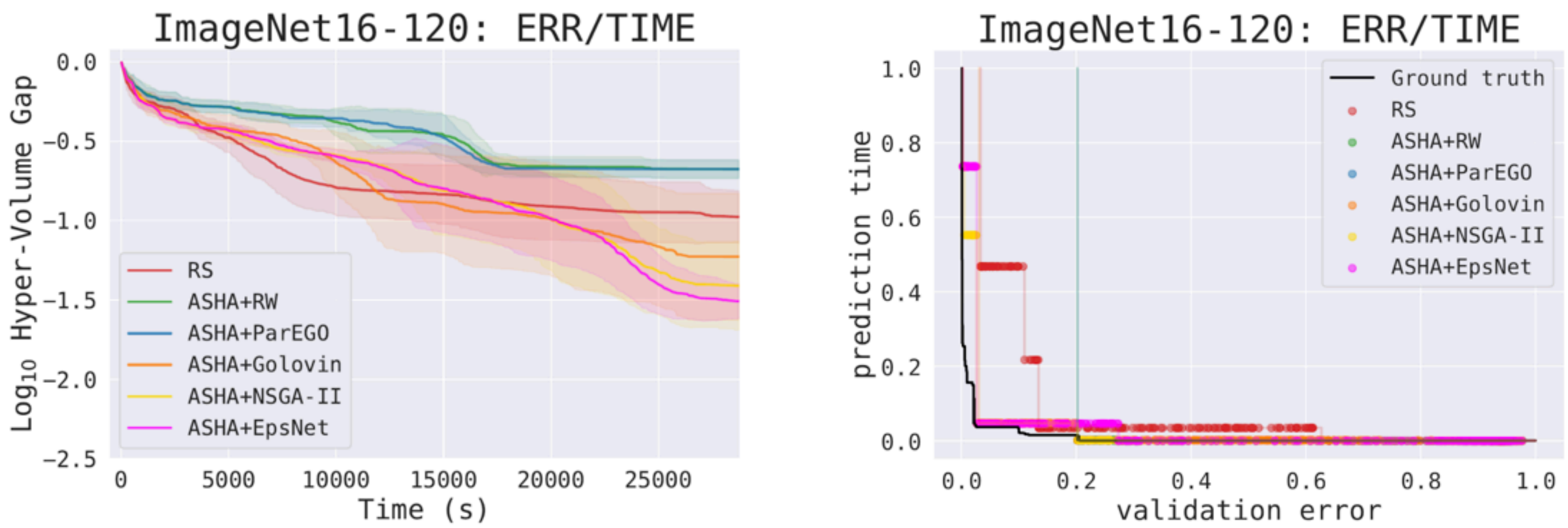


Wistuba and Grabocka. Meta-Learning for Hyperparameter Optimization 2023

- 👍 **Great performance** in practice and one can use multiple workers
- 🤔 How can we extend the algorithm to handle multiple objectives?
- 🤔 We need to sort to discard the bottom half of configurations, how can we sort if we have multiple objectives?

💡 Non-dominated sort allows to sort even when we have multiple objectives

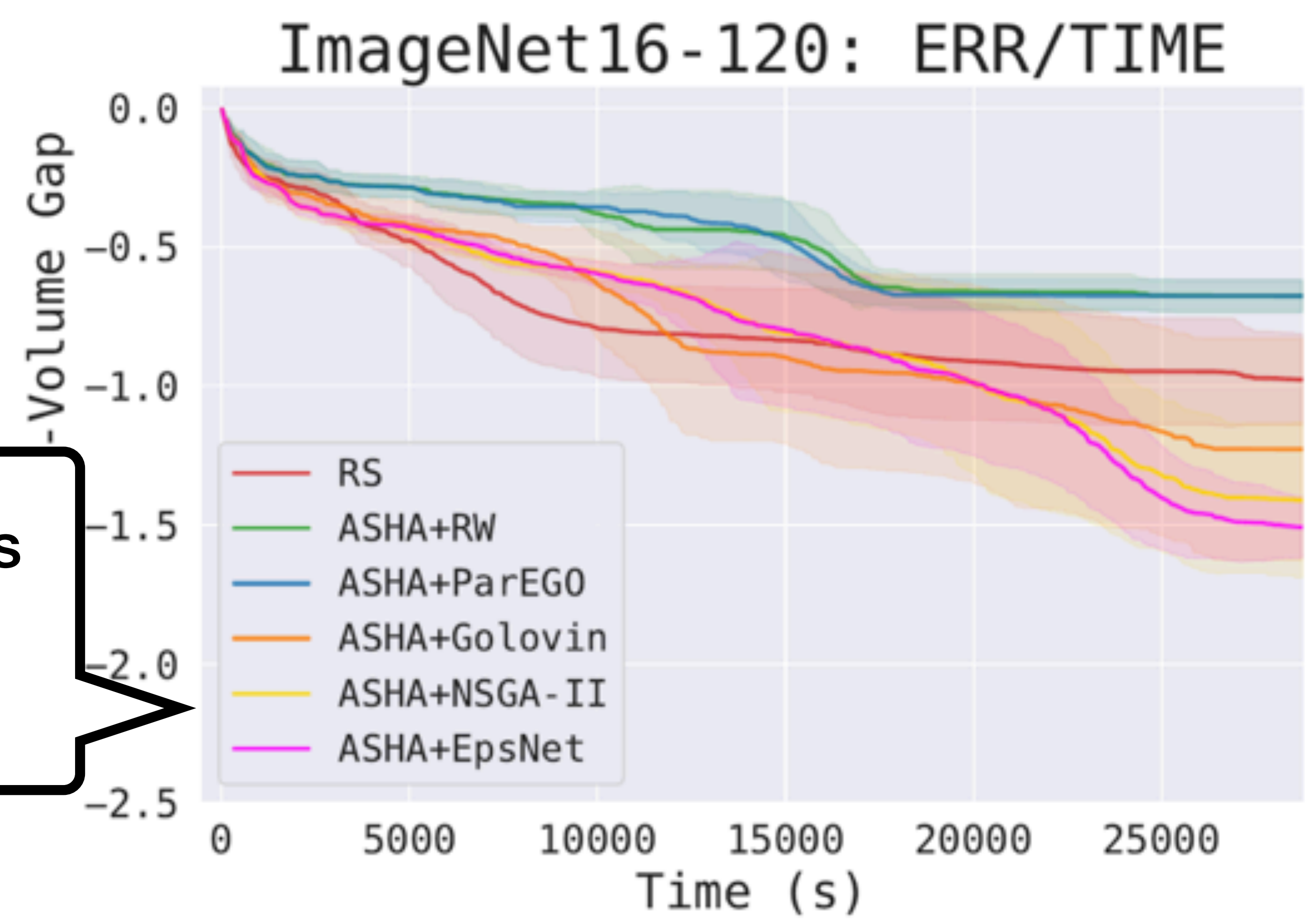
Extending Multifidelity to multi-objective

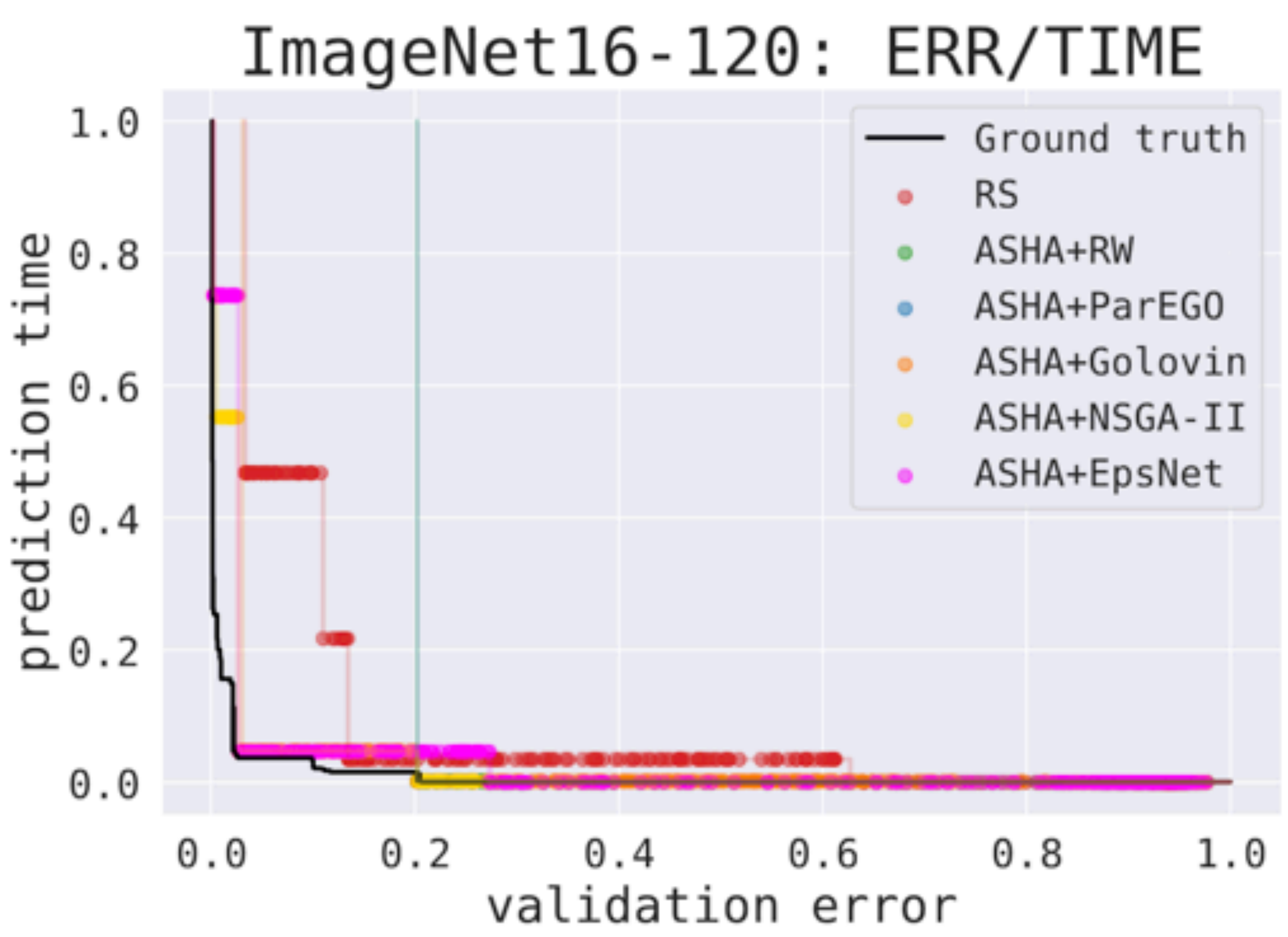


Multi-objective Asynchronous Successive Halving [Schmucker 2021]

Extending Multifidelity to multi-objective

Non-dominated sort, works much better than scalarization!

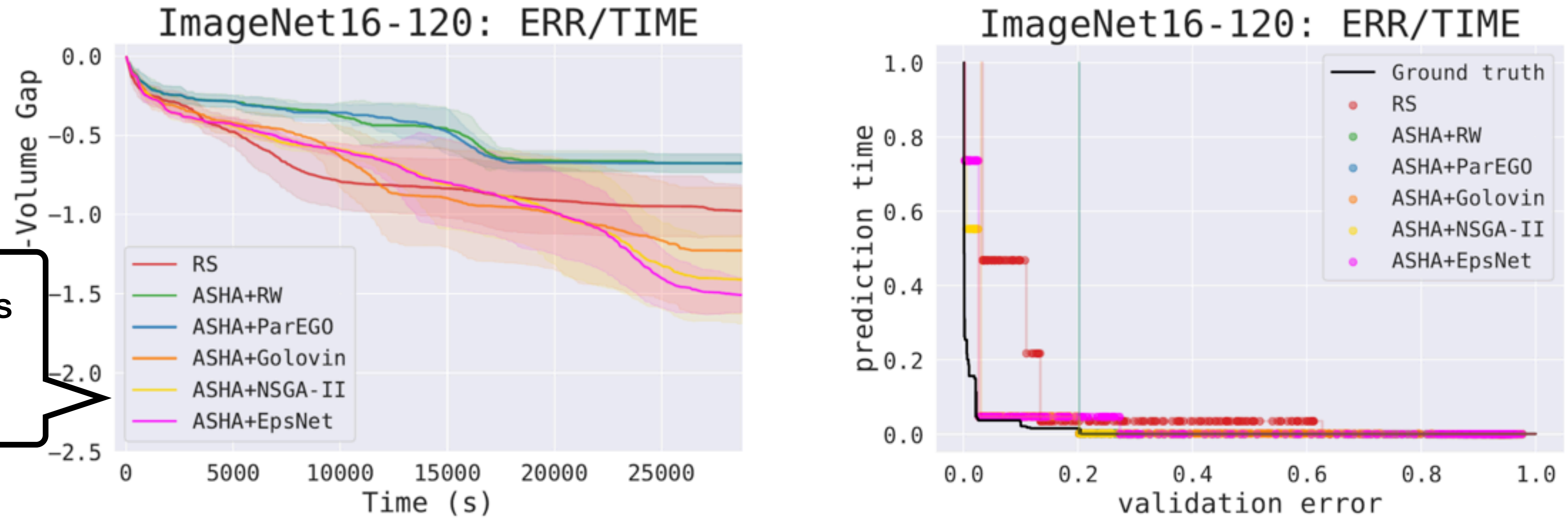




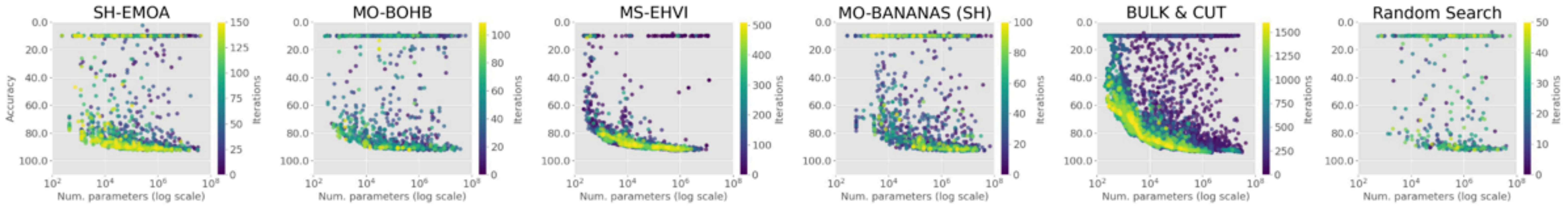
Multi-objective Asynchronous Successive Halving [Schmucker 2021]

Extending Multifidelity to multi-objective

Non-dominated sort, works much better than scalarization!



Multi-objective Asynchronous Successive Halving [Schmucker 2021]

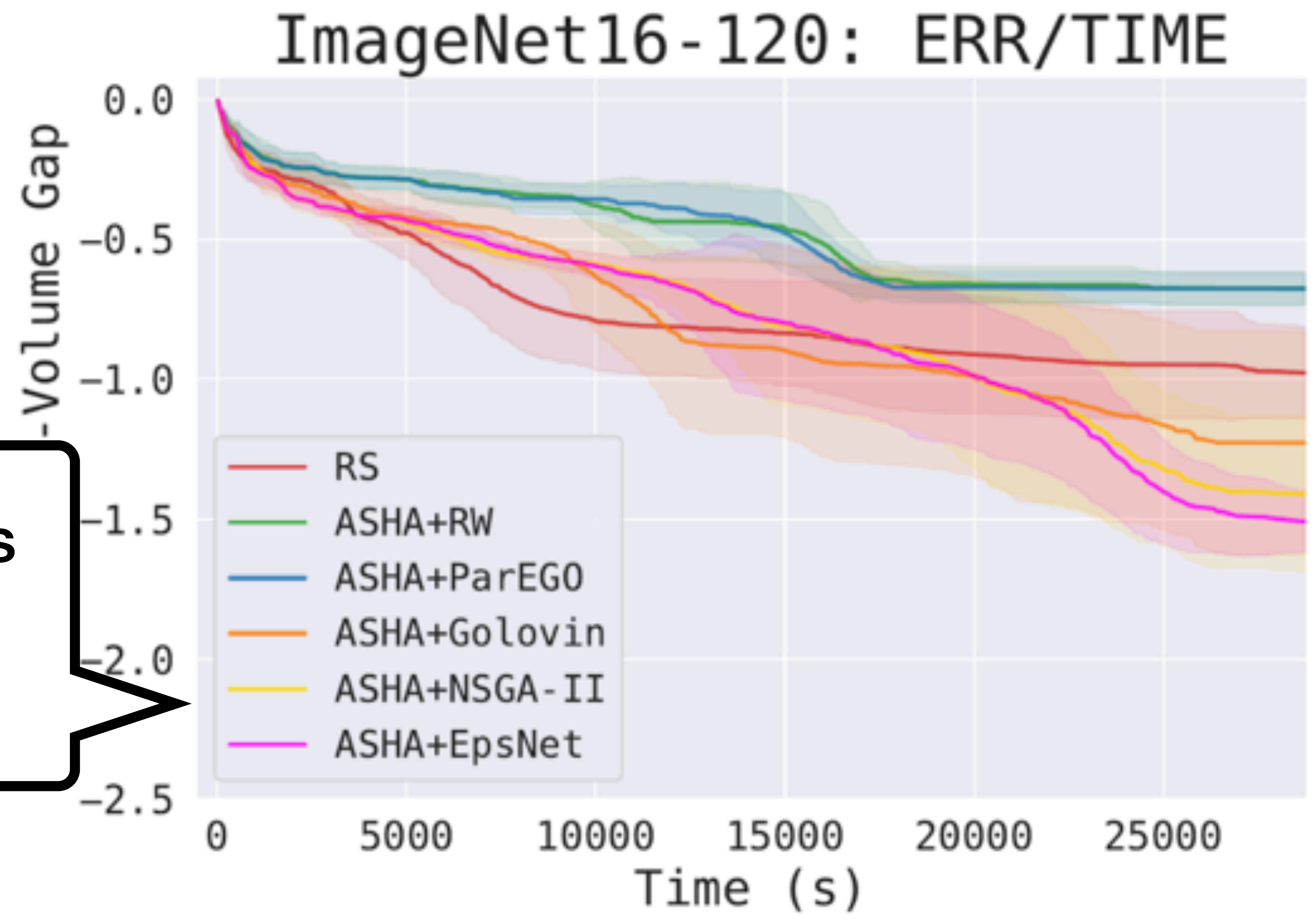


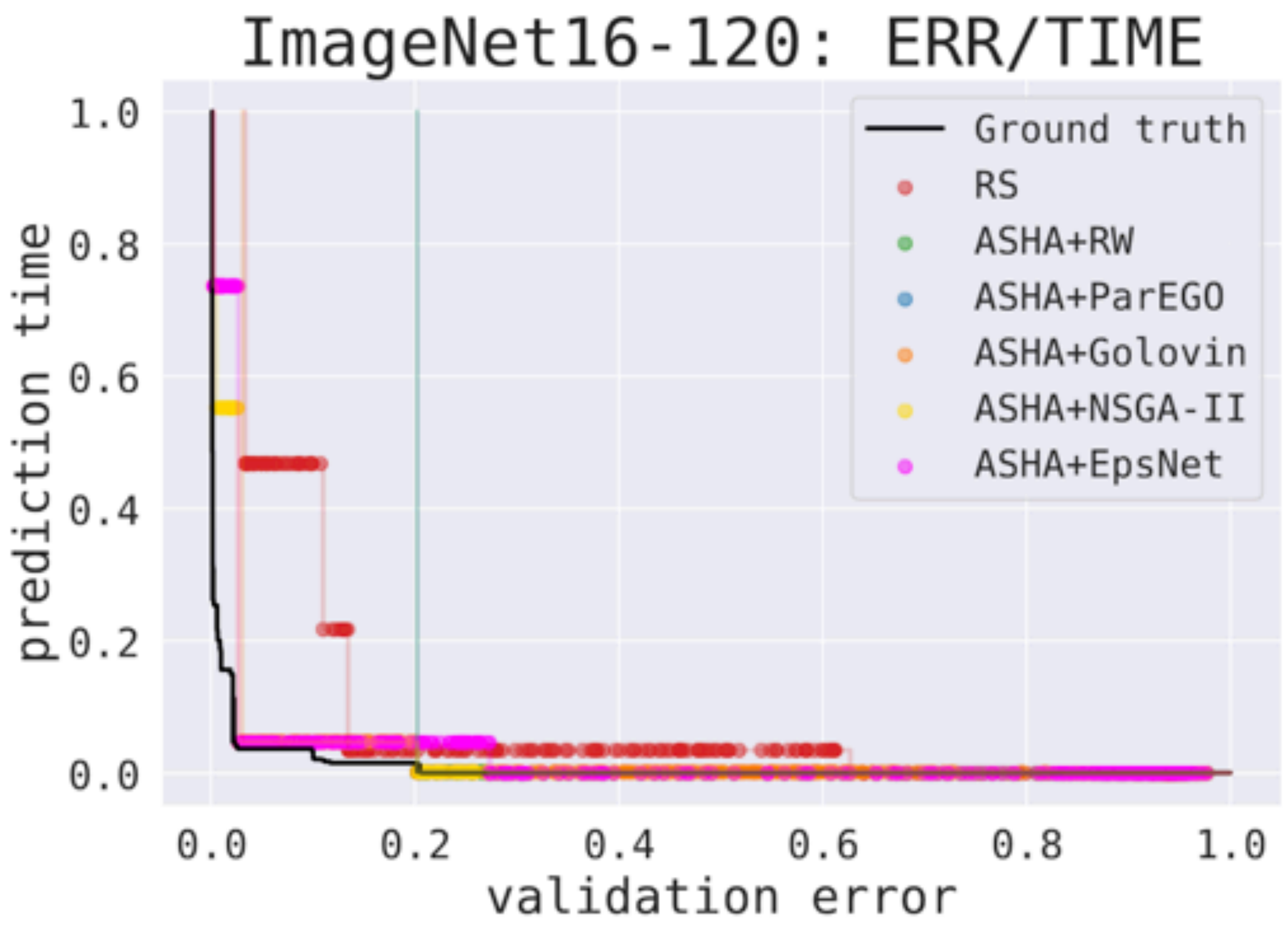
(b) Sampled configurations on Fashion-MNIST dataset.

Bag of Baselines for Multi-objective Joint Neural Architecture Search and Hyperparameter Optimization [Guererro Viu 2021]

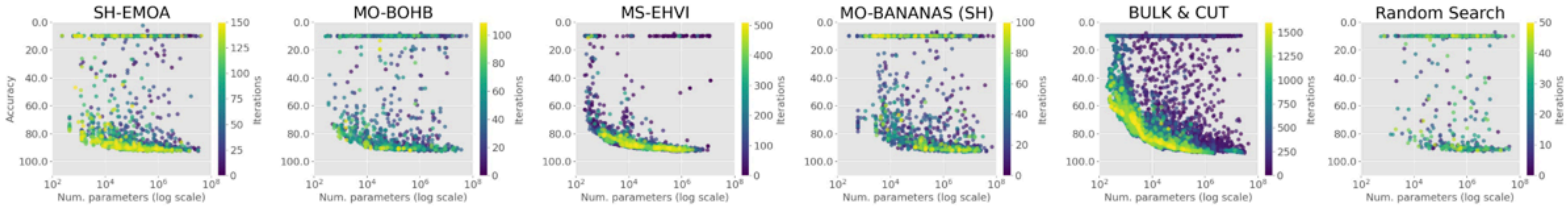
Extending Multifidelity to multi-objective

Non-dominated sort, works much better than scalarization!

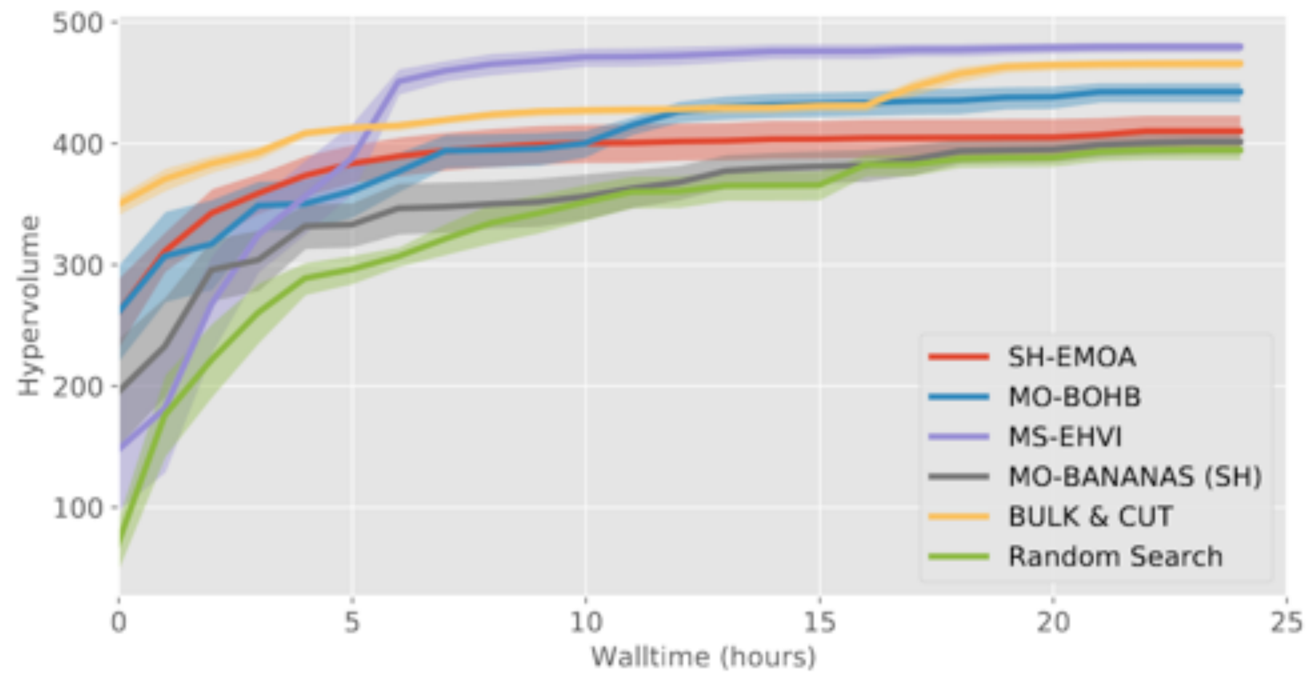




Multi-objective Asynchronous Successive Halving [Schmucker 2021]



(b) Sampled configurations on Fashion-MNIST dataset.

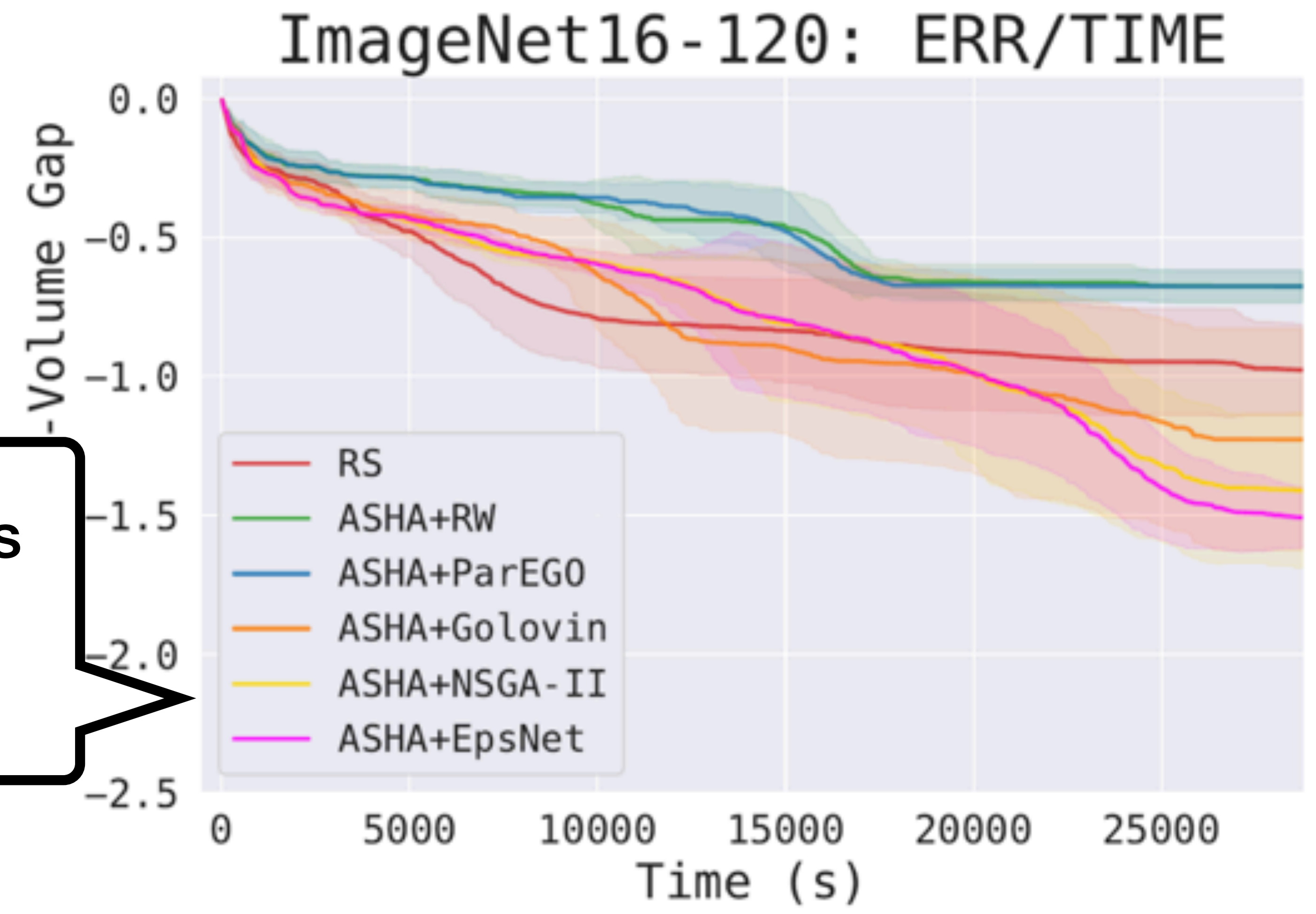


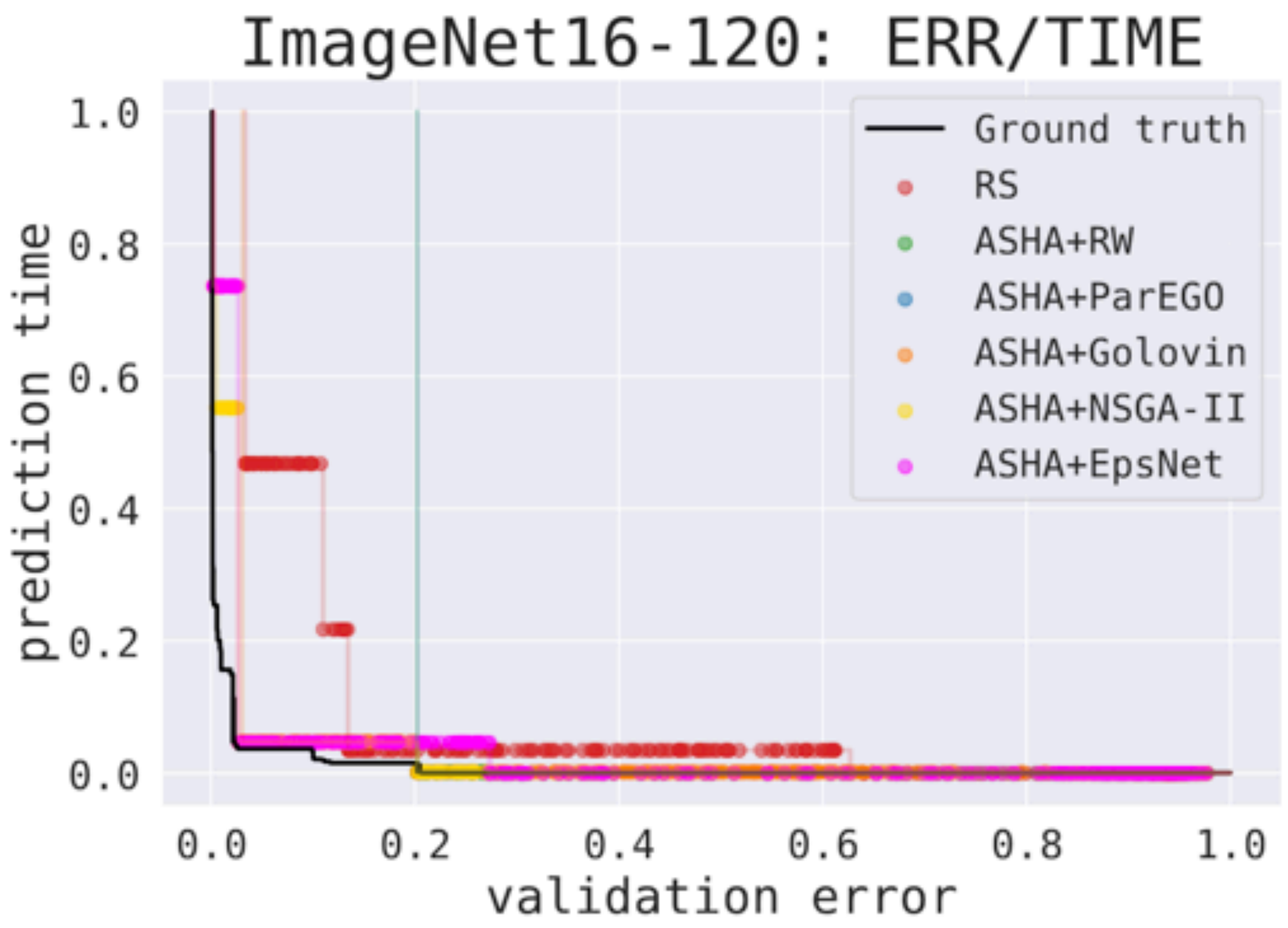
(d) Fashion-MNIST dataset on test split.

Bag of Baselines for Multi-objective Joint Neural Architecture Search and Hyperparameter Optimization [Guererro Viu 2021]

Extending Multifidelity to multi-objective

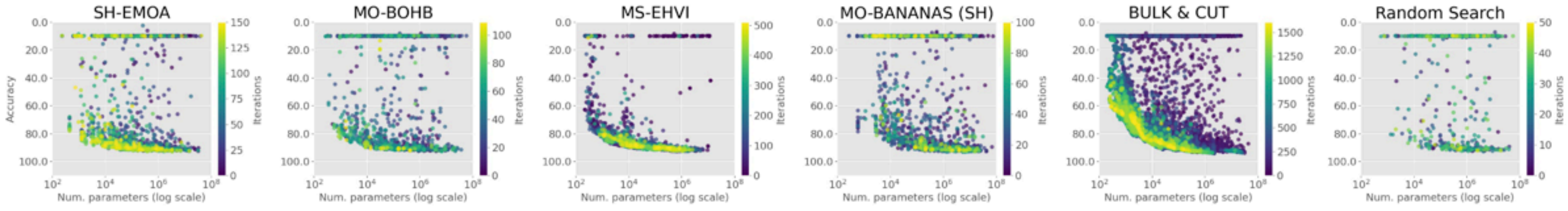
Non-dominated sort, works much better than scalarization!



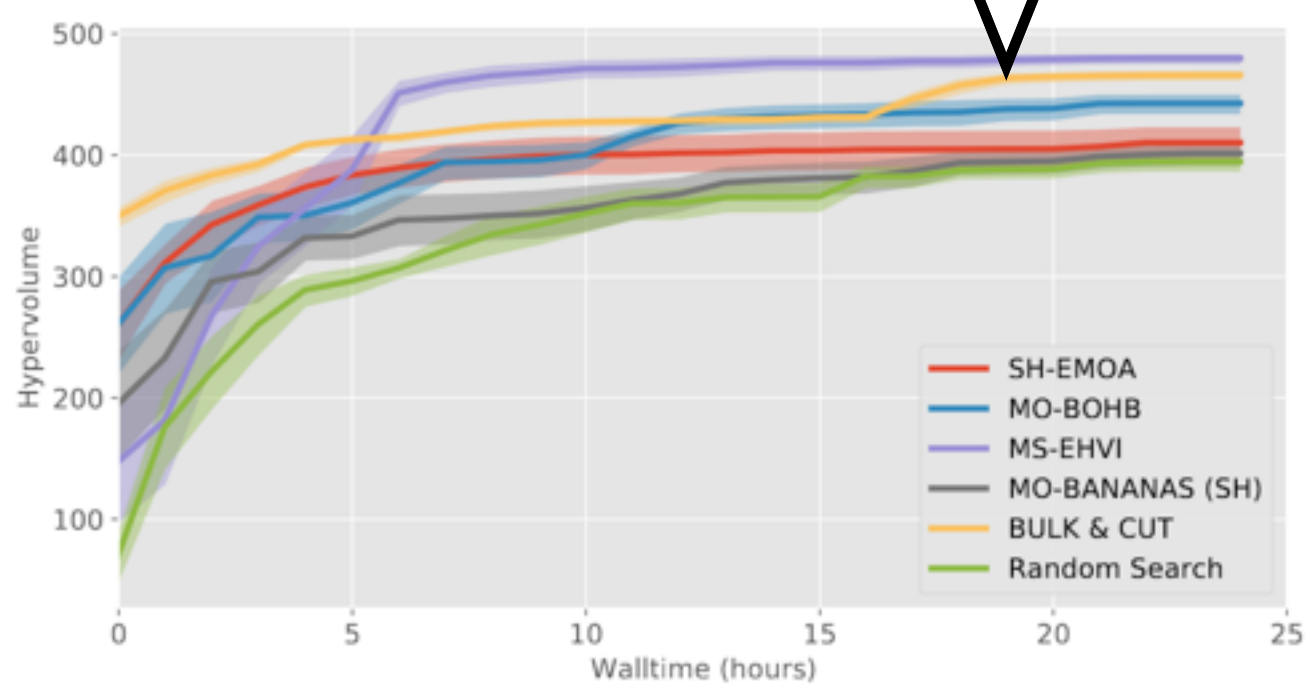


Multi-objective Asynchronous Successive Halving [Schmucker 2021]

Non-dominated sort, works well in general…



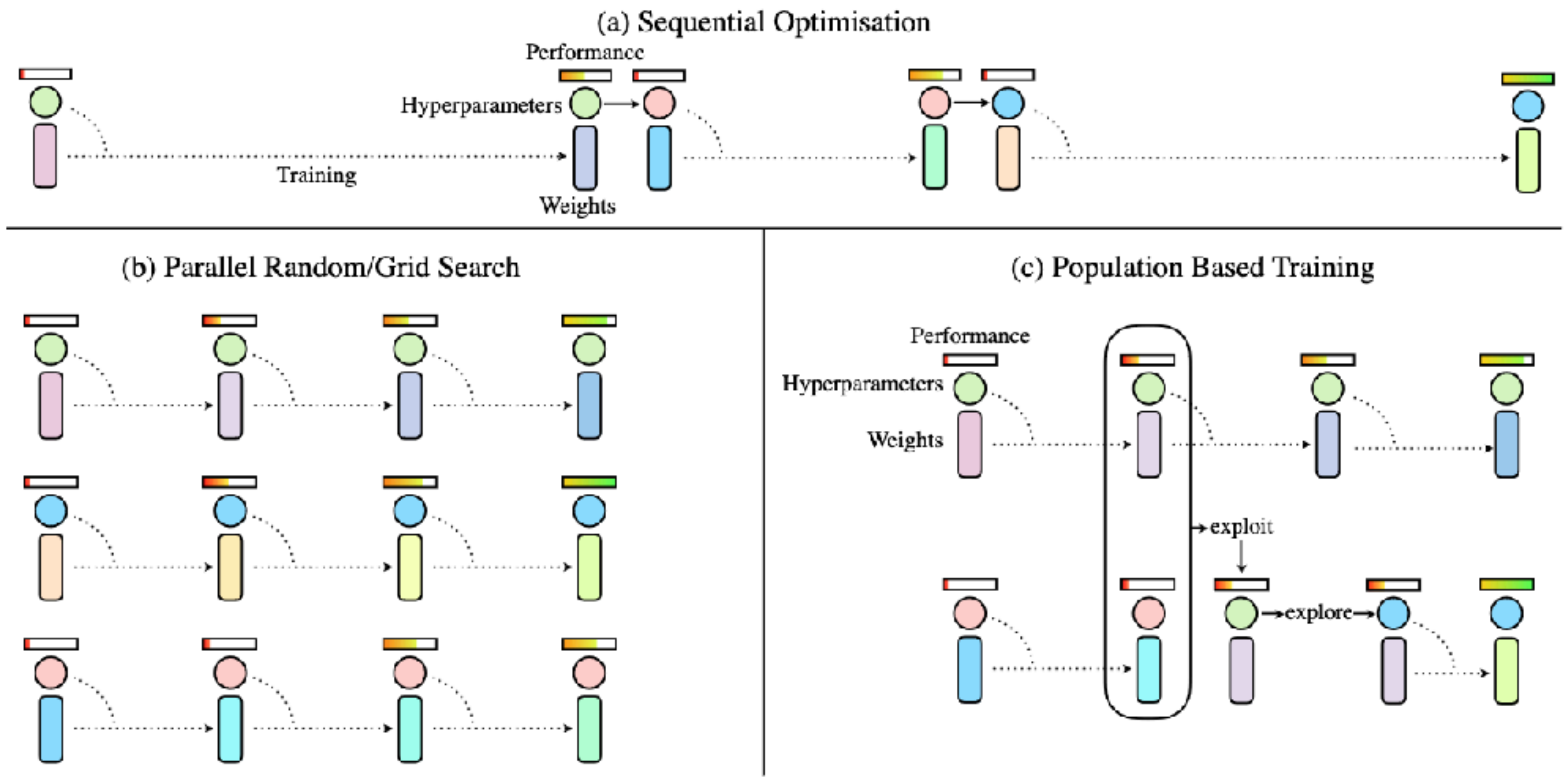
(b) Sampled configurations on Fashion-MNIST dataset.



(d) Fashion-MNIST dataset on test split.

Bag of Baselines for Multi-objective Joint Neural Architecture Search and Hyperparameter Optimization [Guererro Viu 2021]

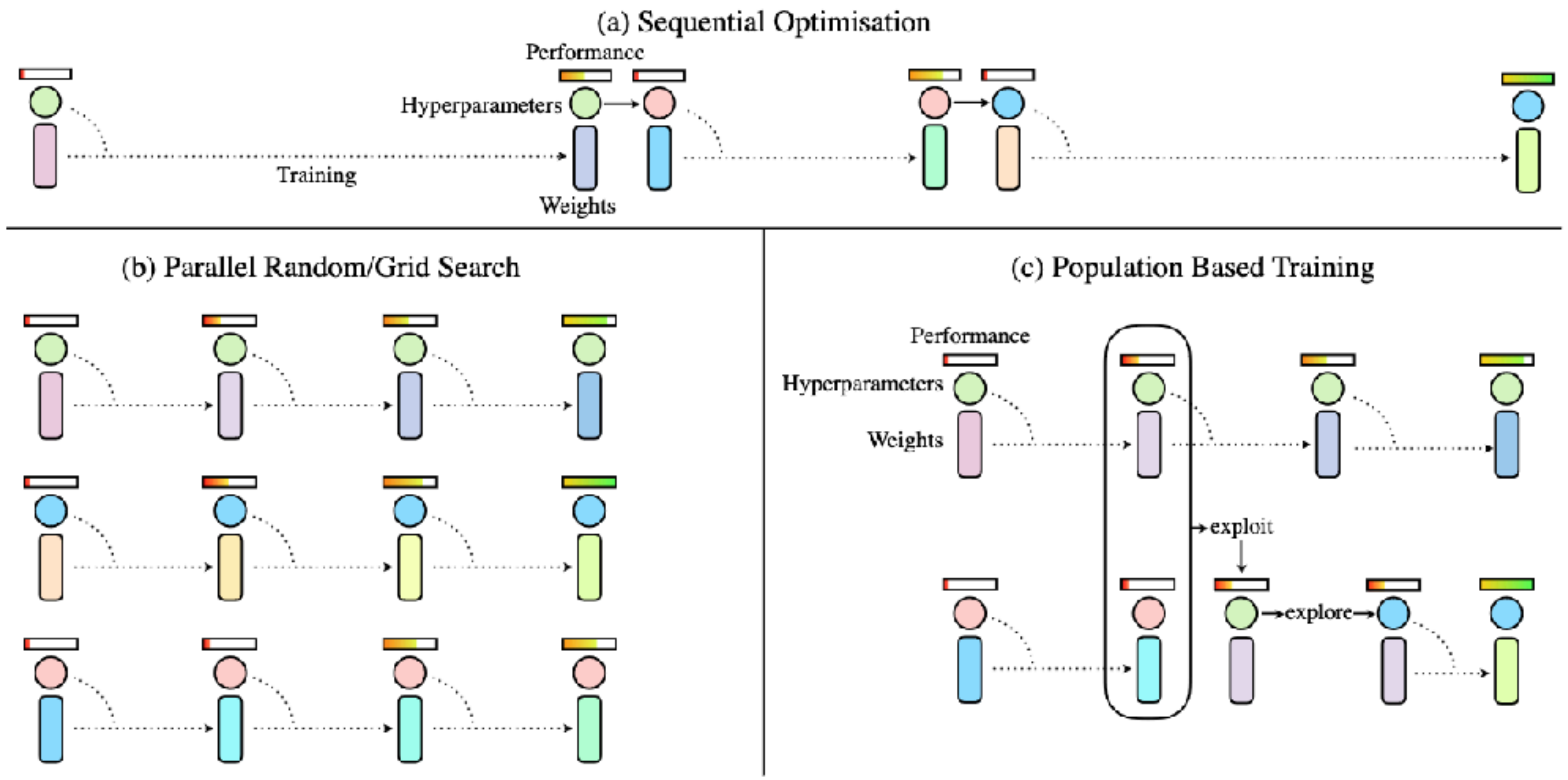
Population Based Training



https://deepmind.google/discover/blog/population-based-training-of-neural-networks/

Population Based Training

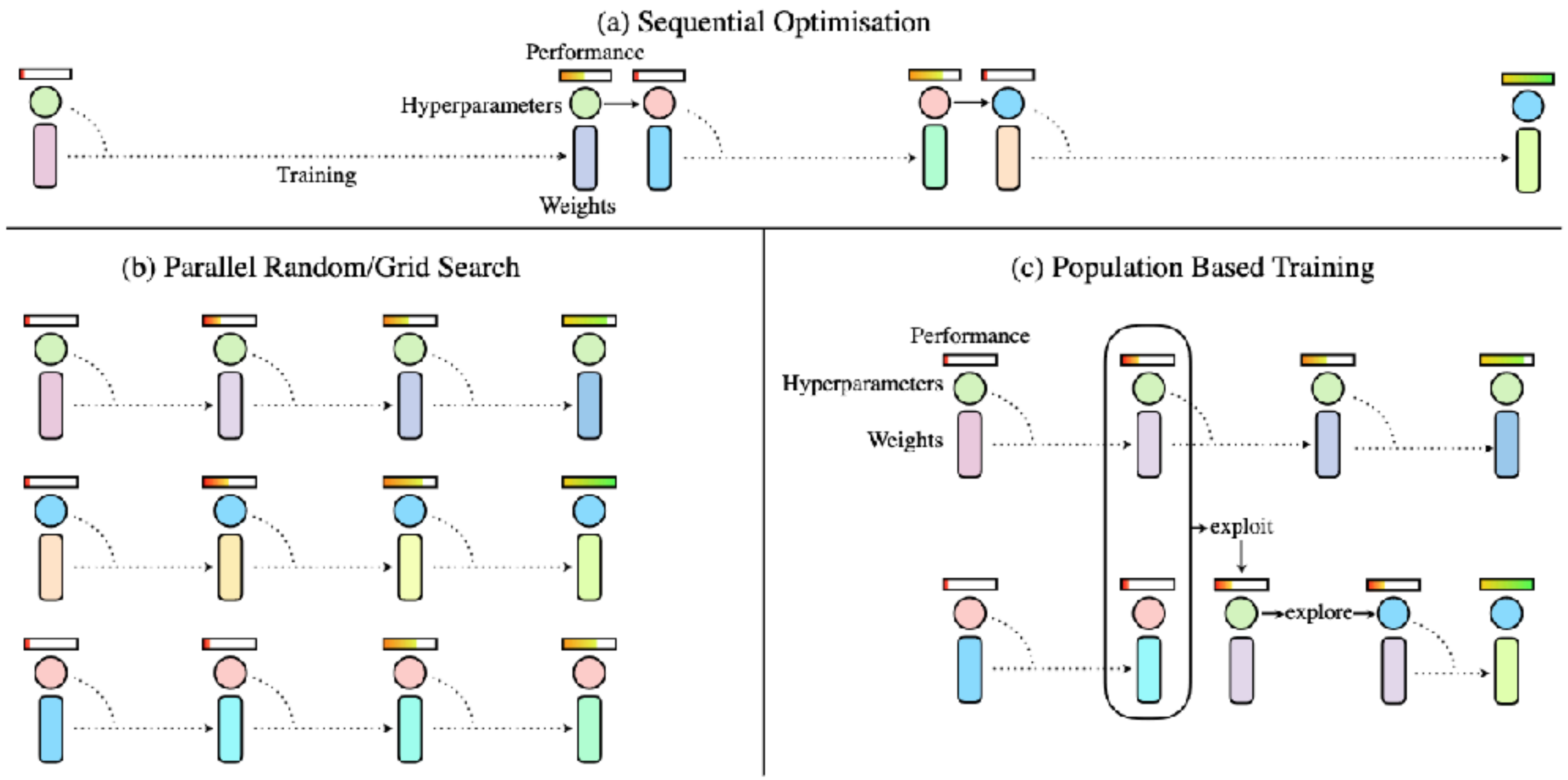
- Tune by having a population of candidates



https://deepmind.google/discover/blog/population-based-training-of-neural-networks/

Population Based Training

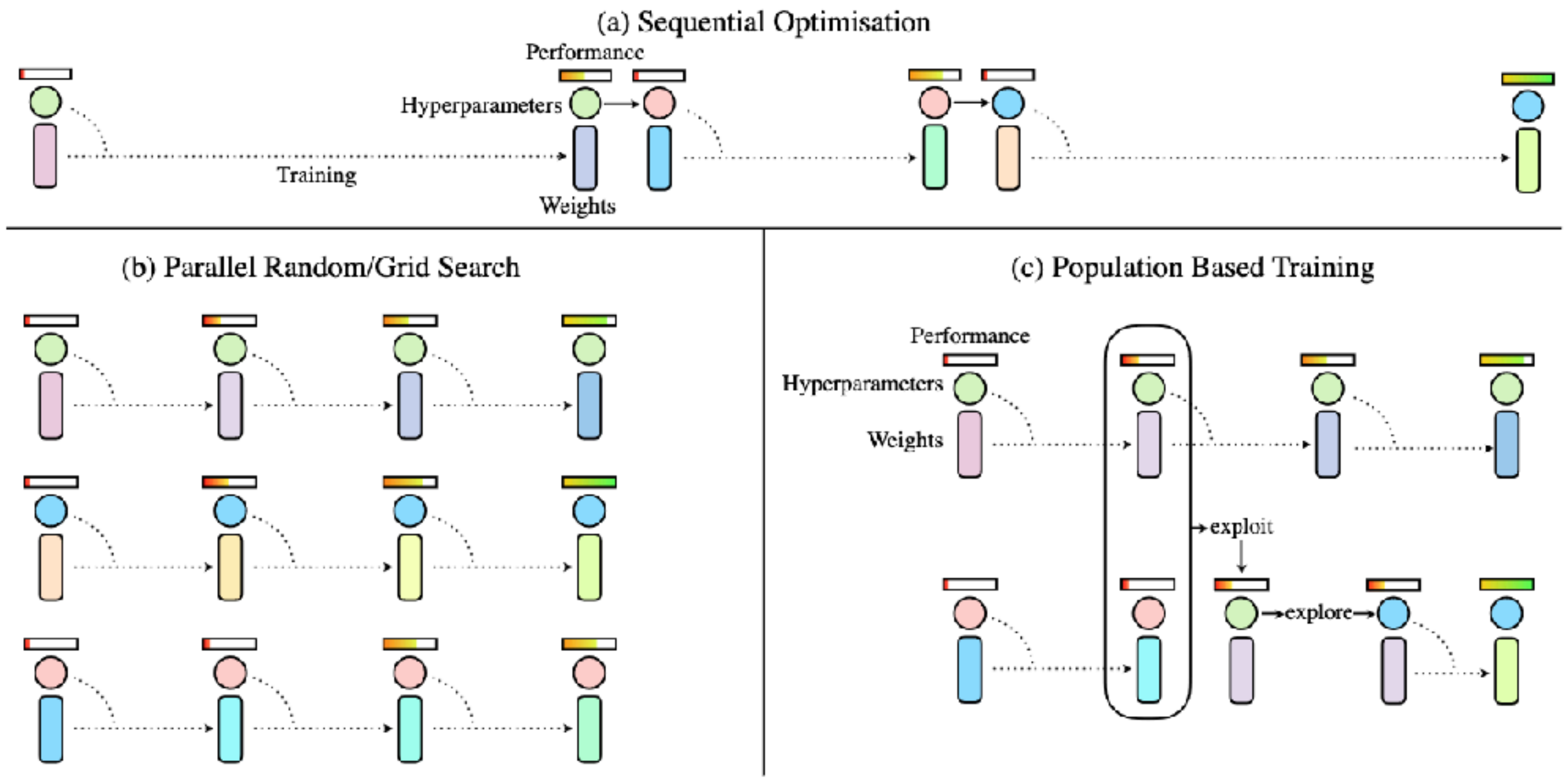
- Tune by having a population of candidates
- When performing mutation, reuse previous weights!



https://deepmind.google/discover/blog/population-based-training-of-neural-networks/

Population Based Training

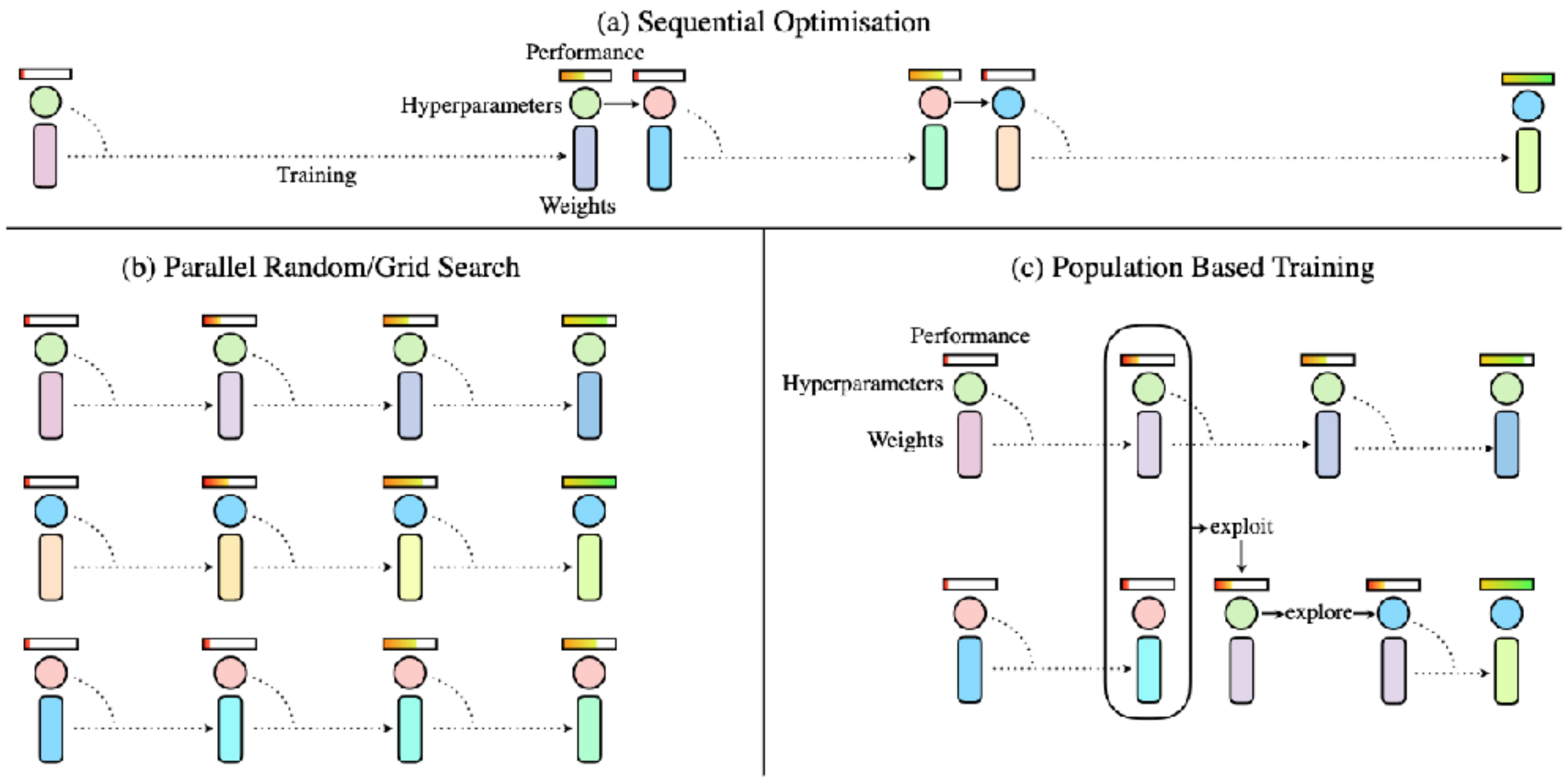
- Tune by having a population of candidates
- When performing mutation, reuse previous weights!
- Allows to learn learning rate schedule!

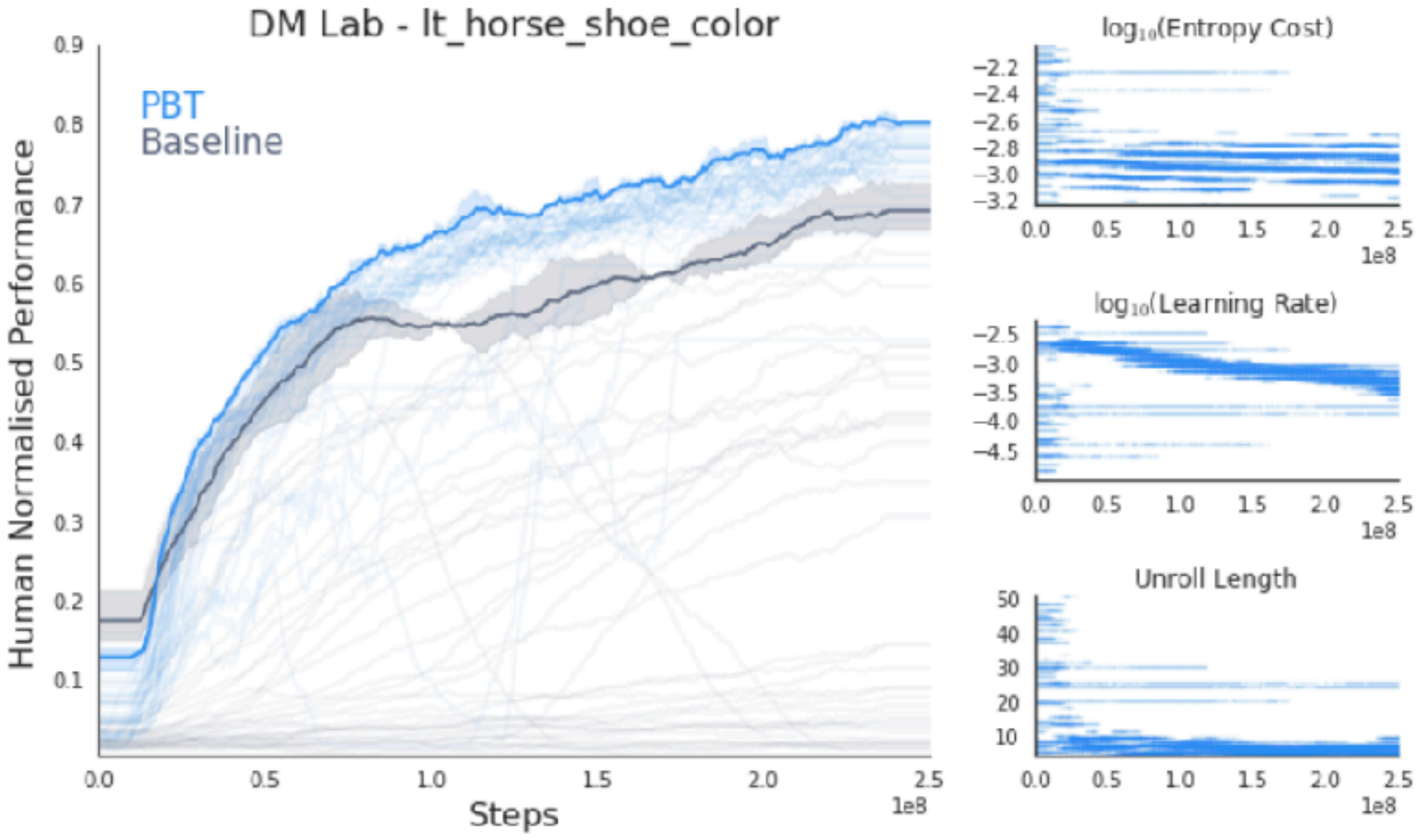


https://deepmind.google/discover/blog/population-based-training-of-neural-networks/

Population Based Training

- Tune by having a population of candidates
- When performing mutation, reuse previous weights!
- Allows to learn learning rate schedule!

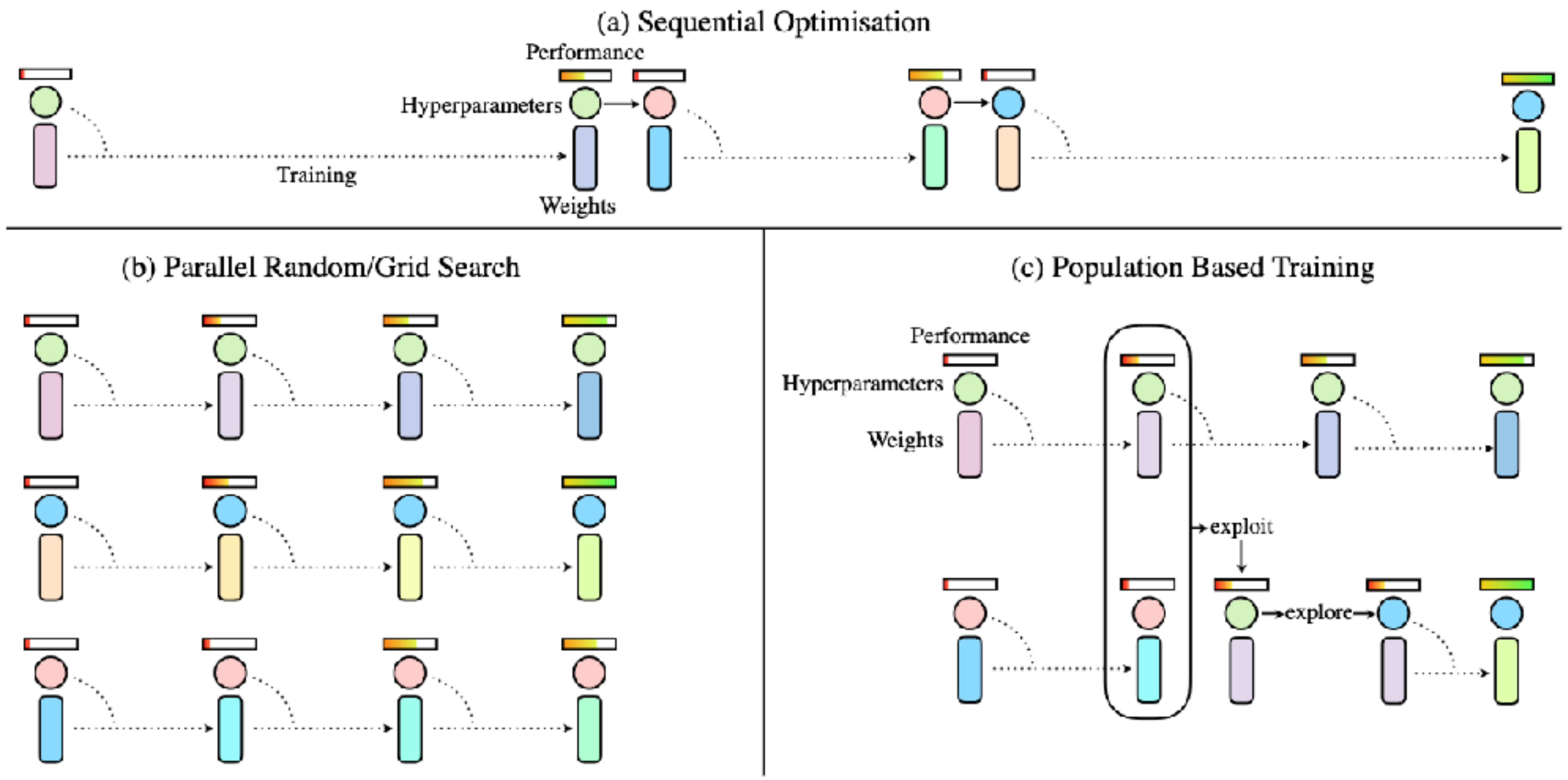


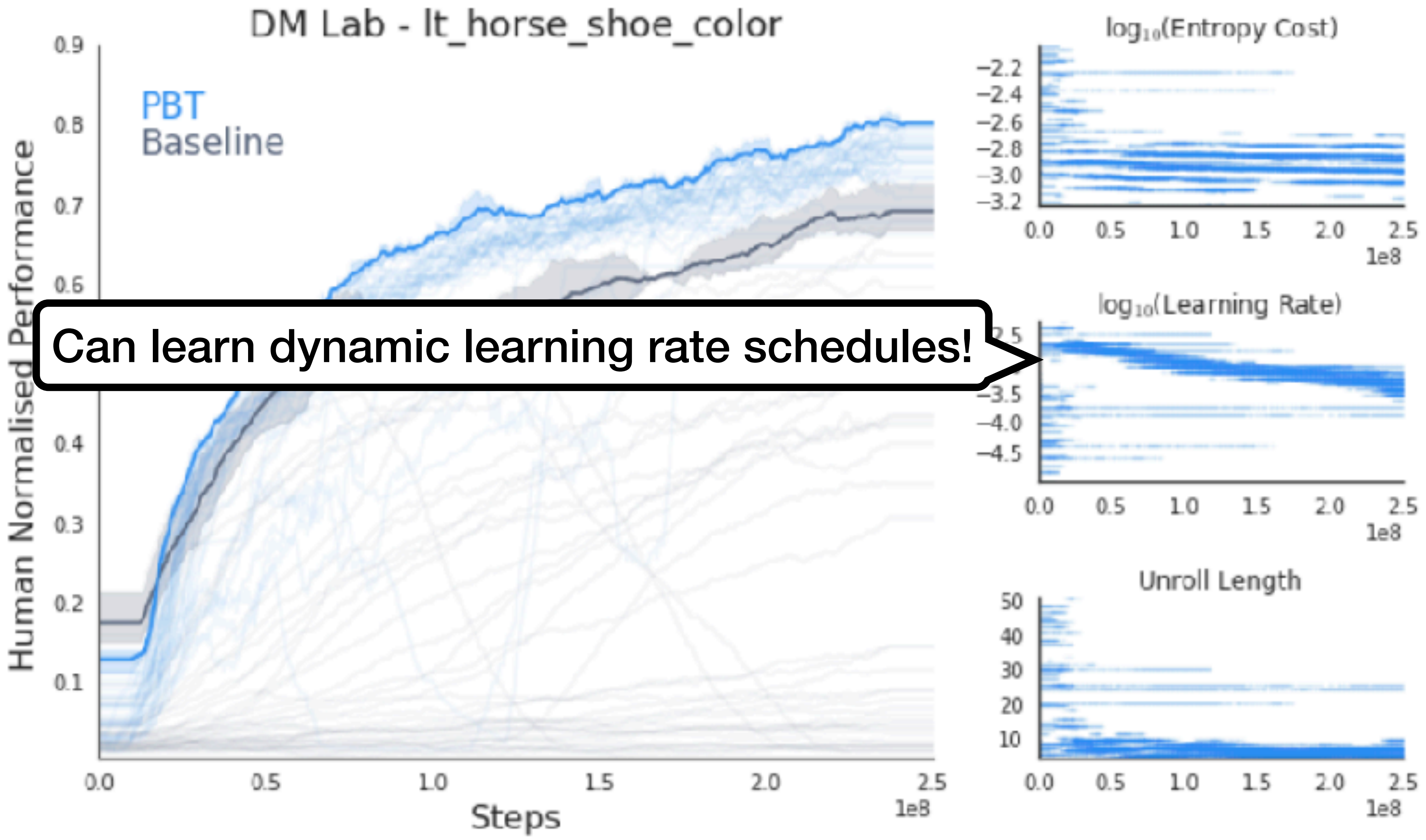


https://deepmind.google/discover/blog/population-based-training-of-neural-networks/

Population Based Training

- Tune by having a population of candidates
- When performing mutation, reuse previous weights!
- Allows to learn learning rate schedule!



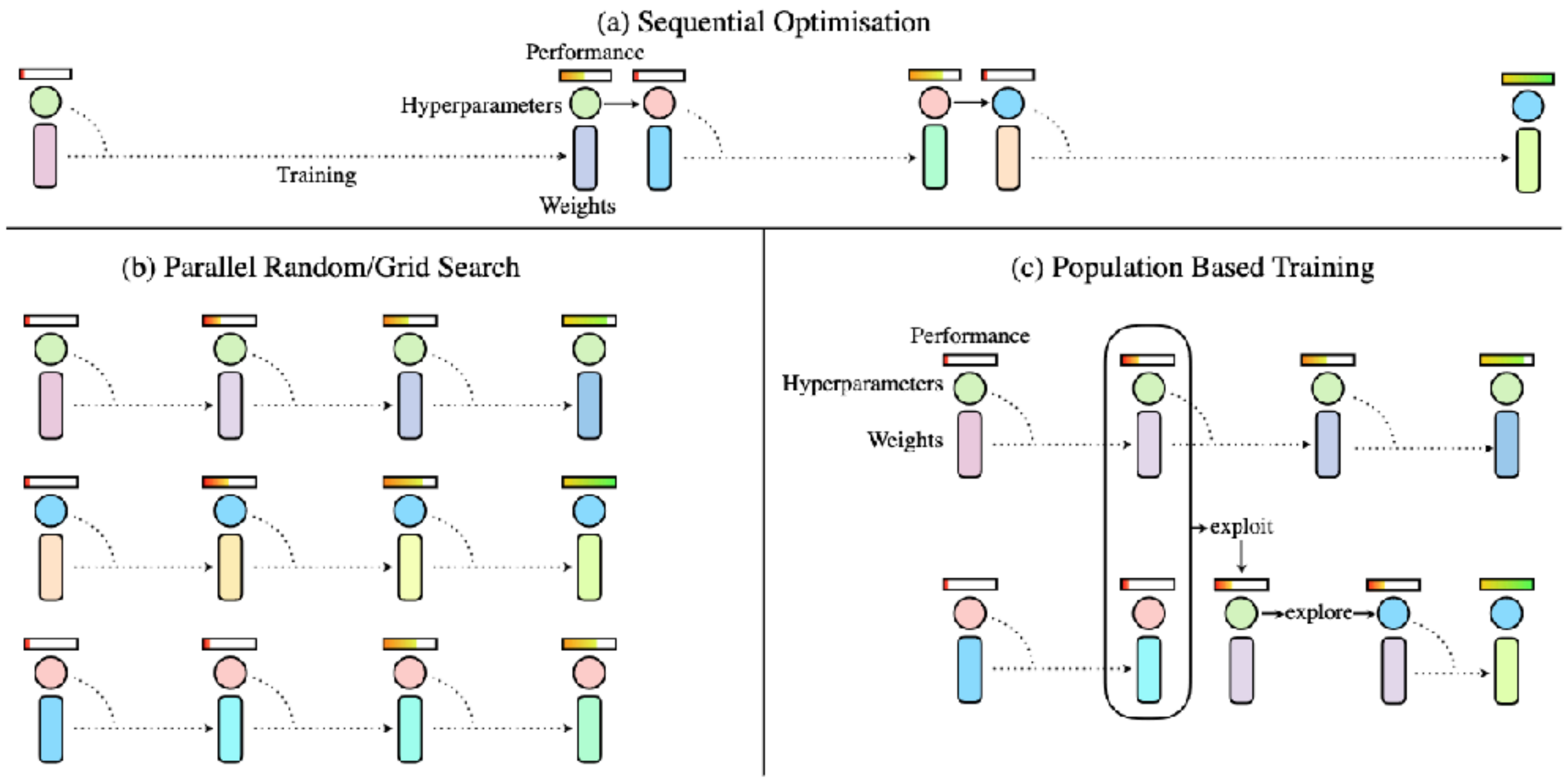


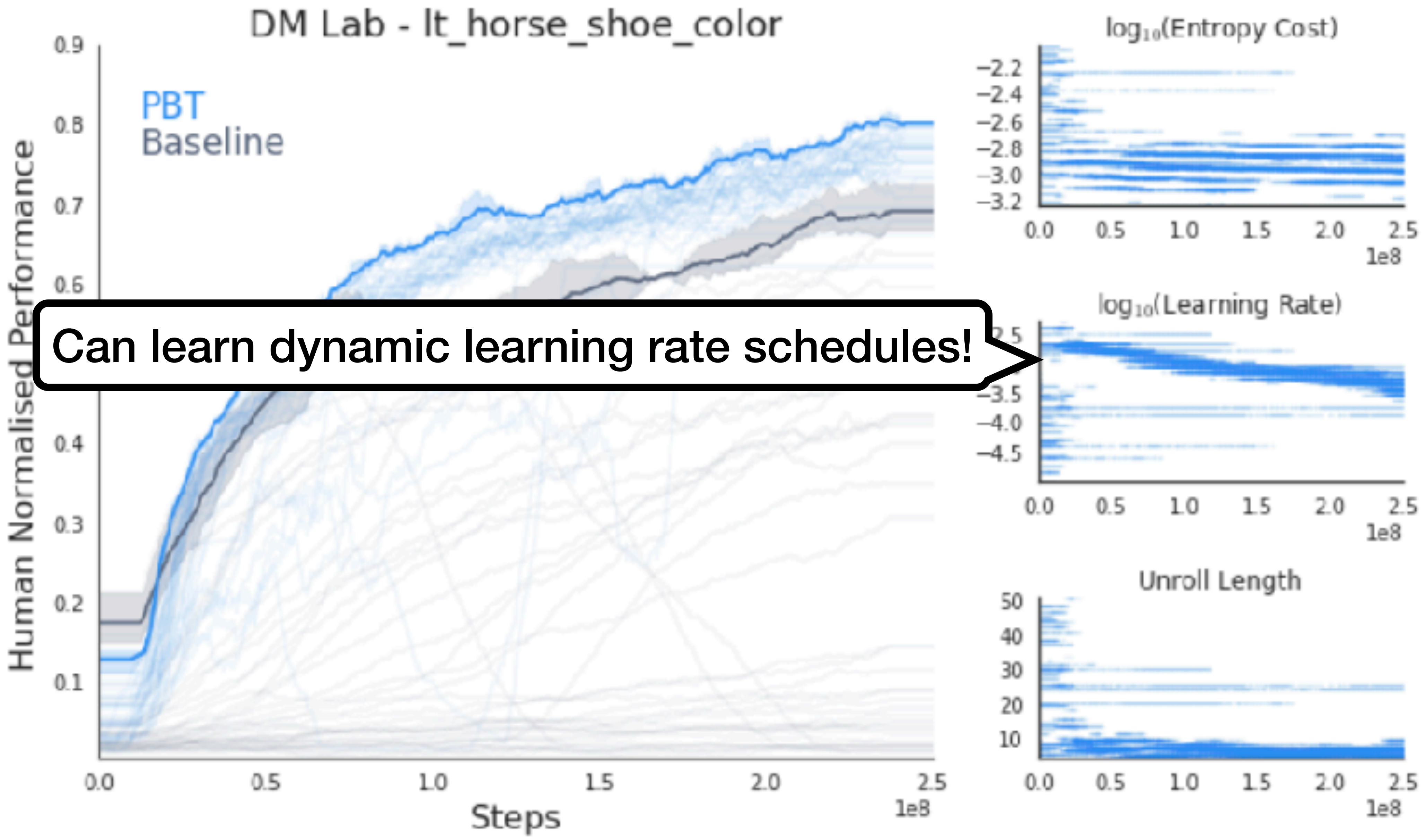
https://deepmind.google/discover/blog/population-based-training-of-neural-networks/

Population Based Training

- Tune by having a population of candidates
- When performing mutation, reuse previous weights!
- Allows to learn learning rate schedule!

🤔 How can we extend the algorithm to handle multiple objectives?





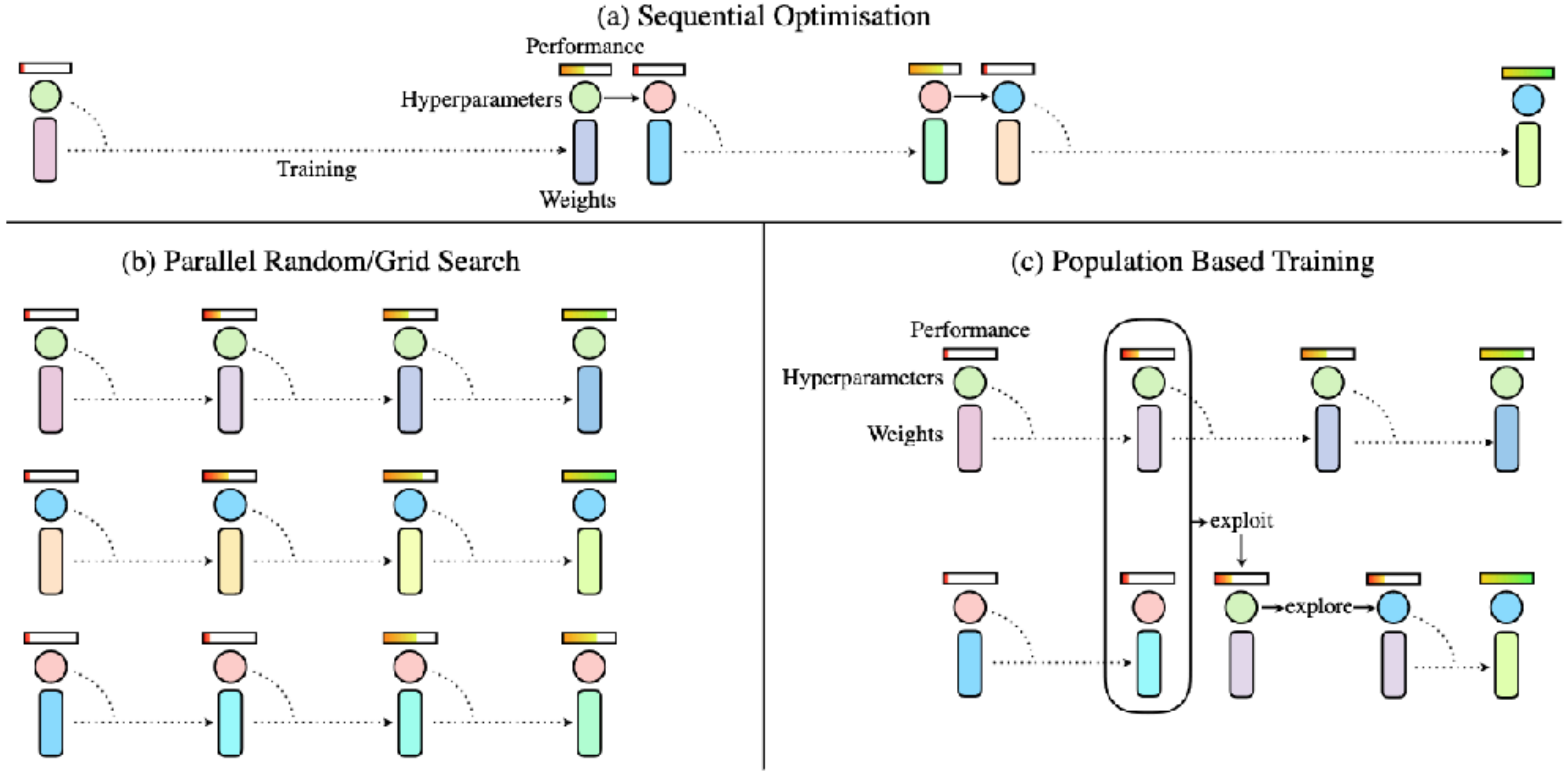
https://deepmind.google/discover/blog/population-based-training-of-neural-networks/

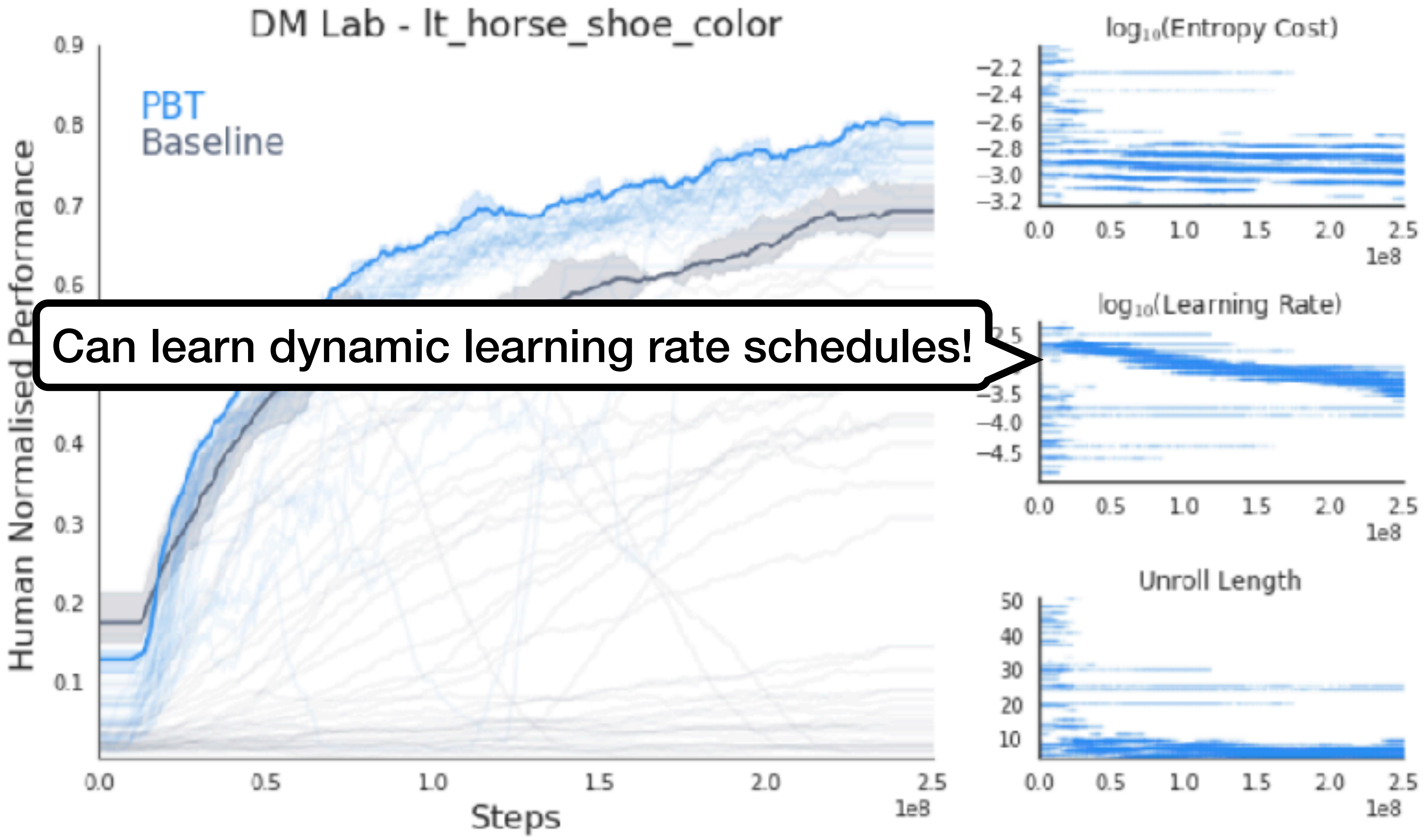
Population Based Training

- Tune by having a population of candidates
- When performing mutation, reuse previous weights!
- Allows to learn learning rate schedule!

🤔 How can we extend the algorithm to handle multiple objectives?

💡 Non-dominated sort allows to sort even when we have multiple objectives





https://deepmind.google/discover/blog/population-based-training-of-neural-networks/

Multiobjective Population Based Training

Multiobjective Population Based Training

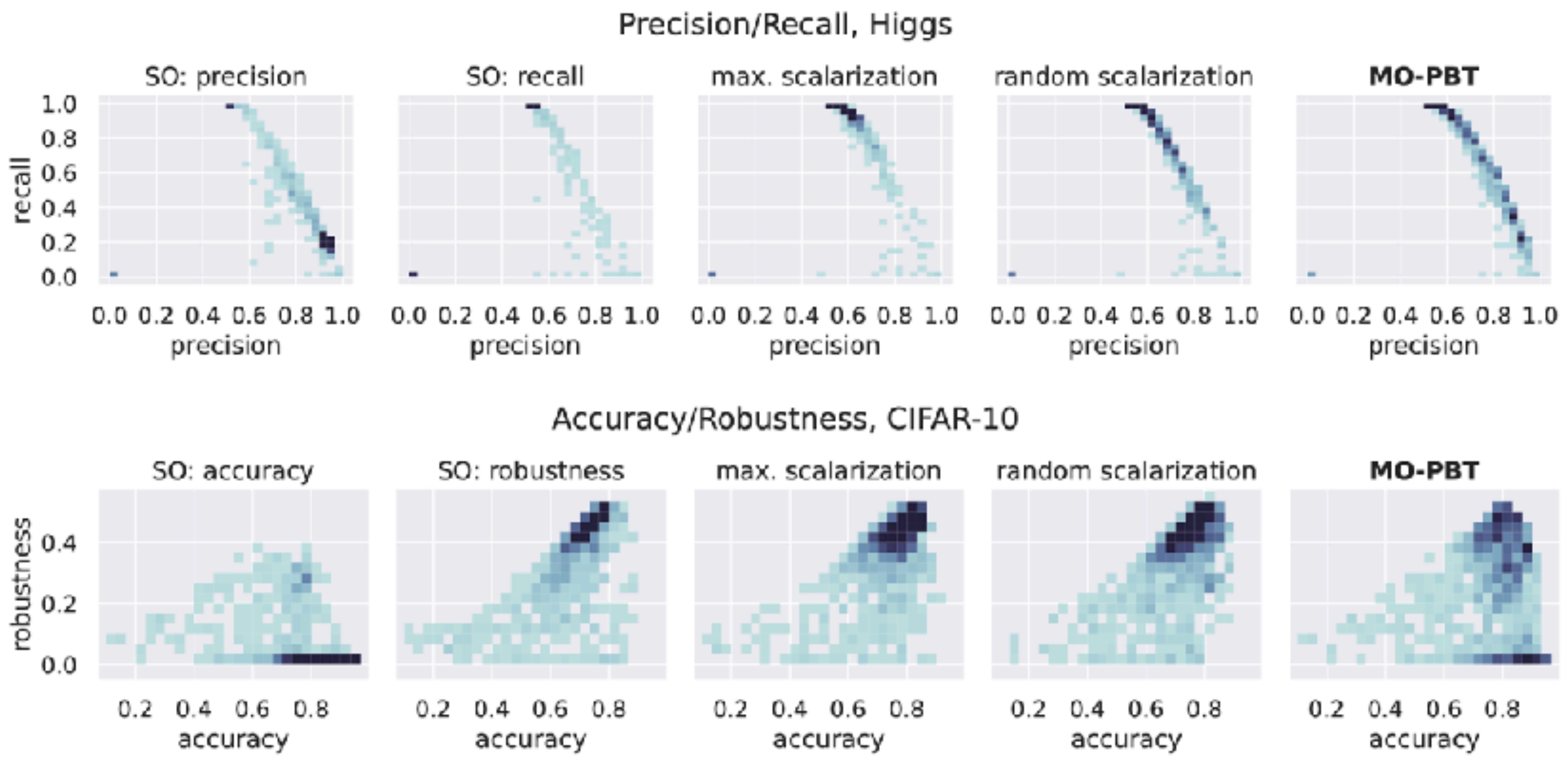
- Multiobjective Population Based Training: uses non-dominated sort with PBT

Multiobjective Population Based Training

- Multiobjective Population Based Training: uses non-dominated sort with PBT
- Extend PBT to multi objectives

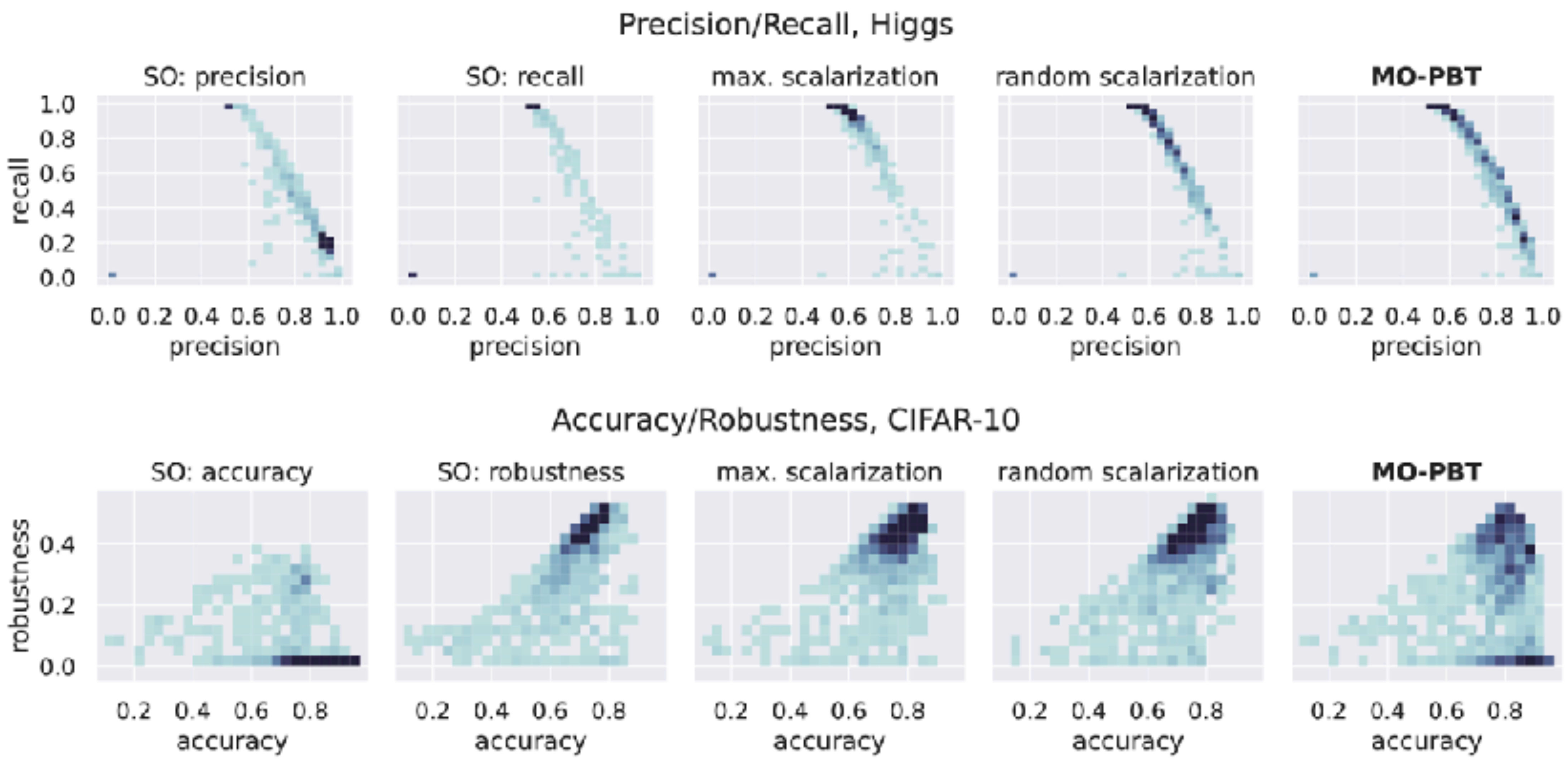
Multiobjective Population Based Training

- Multiobjective Population Based Training: uses non-dominated sort with PBT
- Extend PBT to multi objectives



Multiobjective Population Based Training

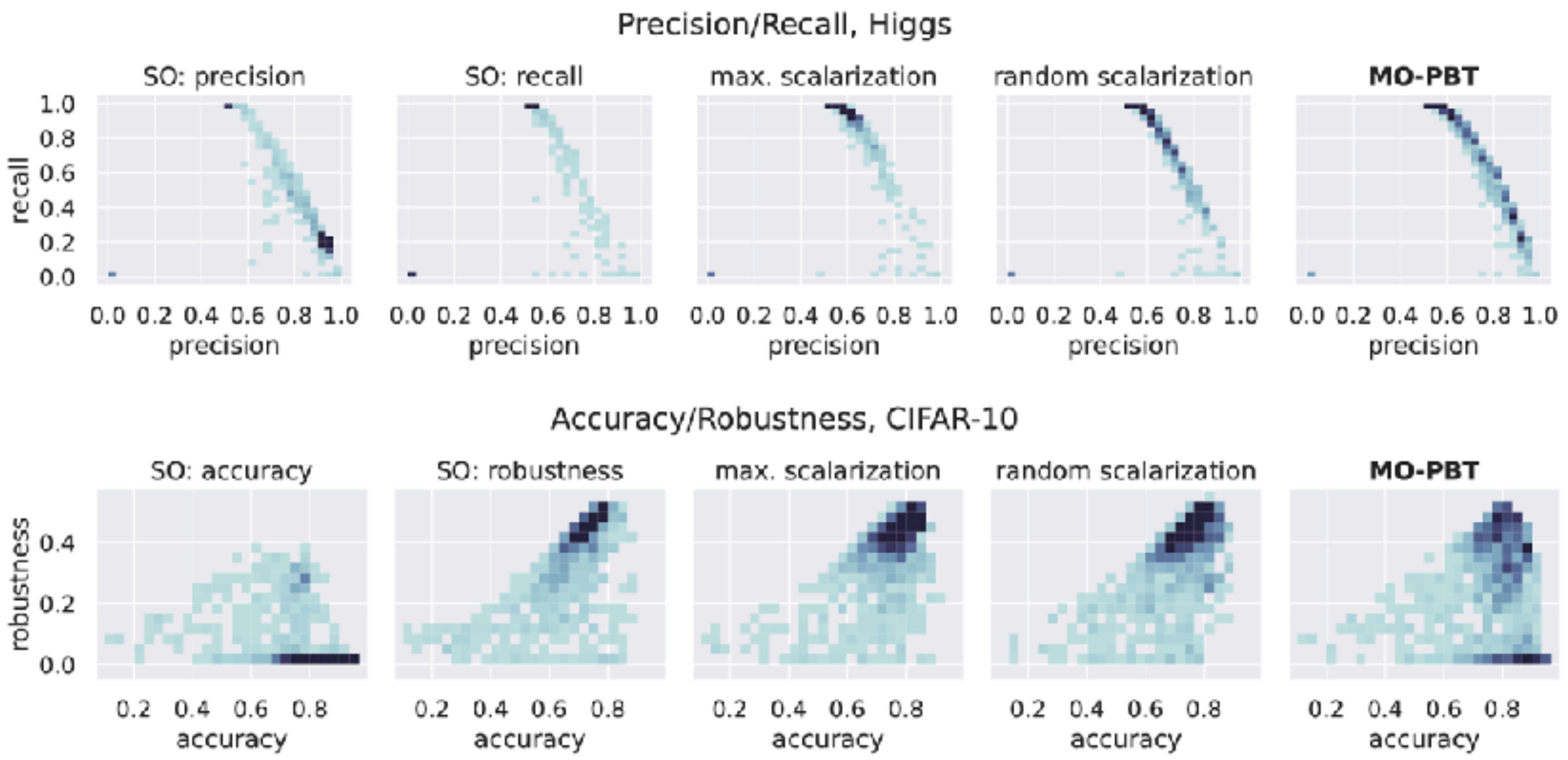
- Multiobjective Population Based Training: uses non-dominated sort with PBT
- Extend PBT to multi objectives



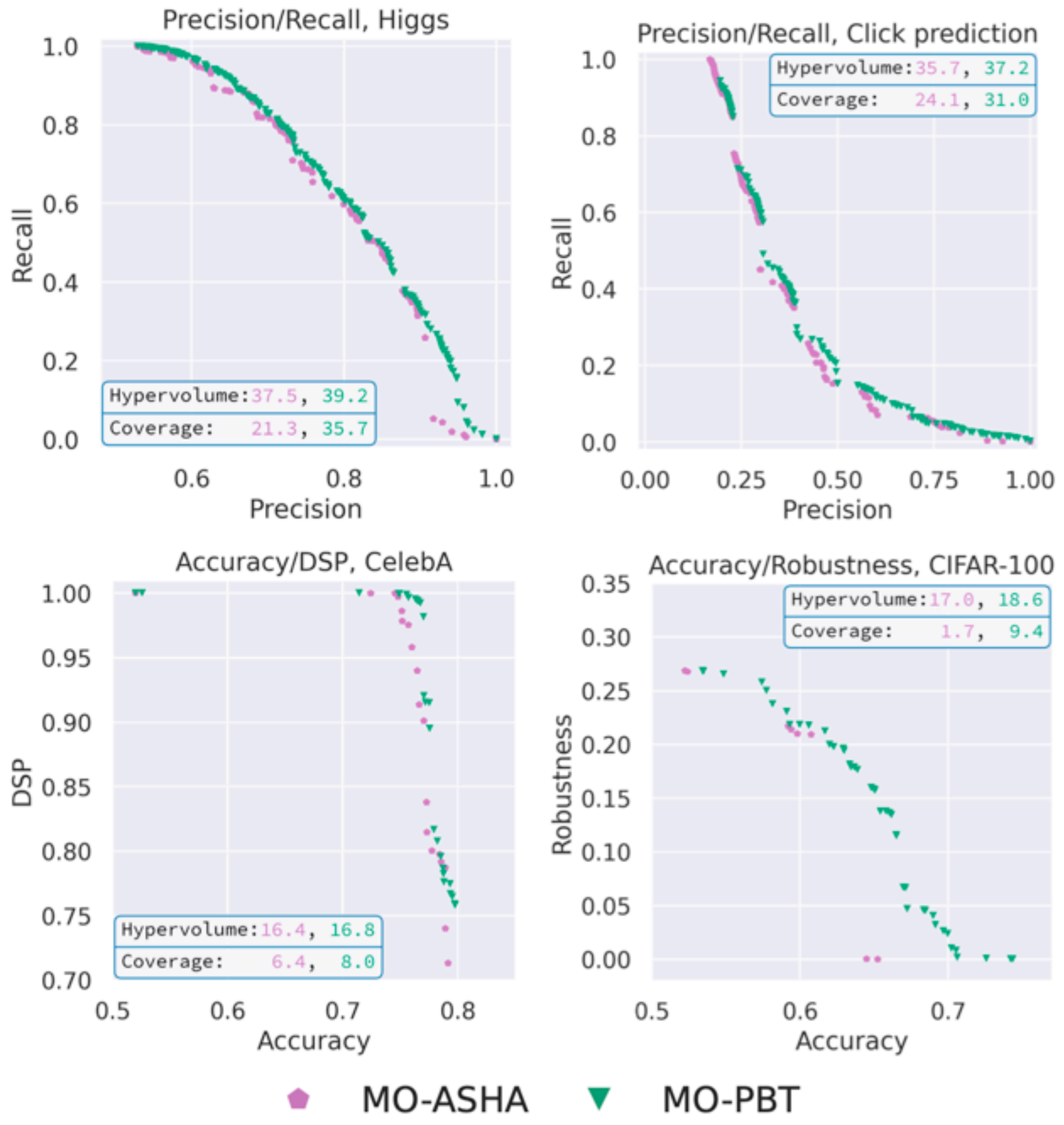
Good at distributing exploration on the Pareto Front

Multiobjective Population Based Training

- Multiobjective Population Based Training: uses non-dominated sort with PBT
- Extend PBT to multi objectives

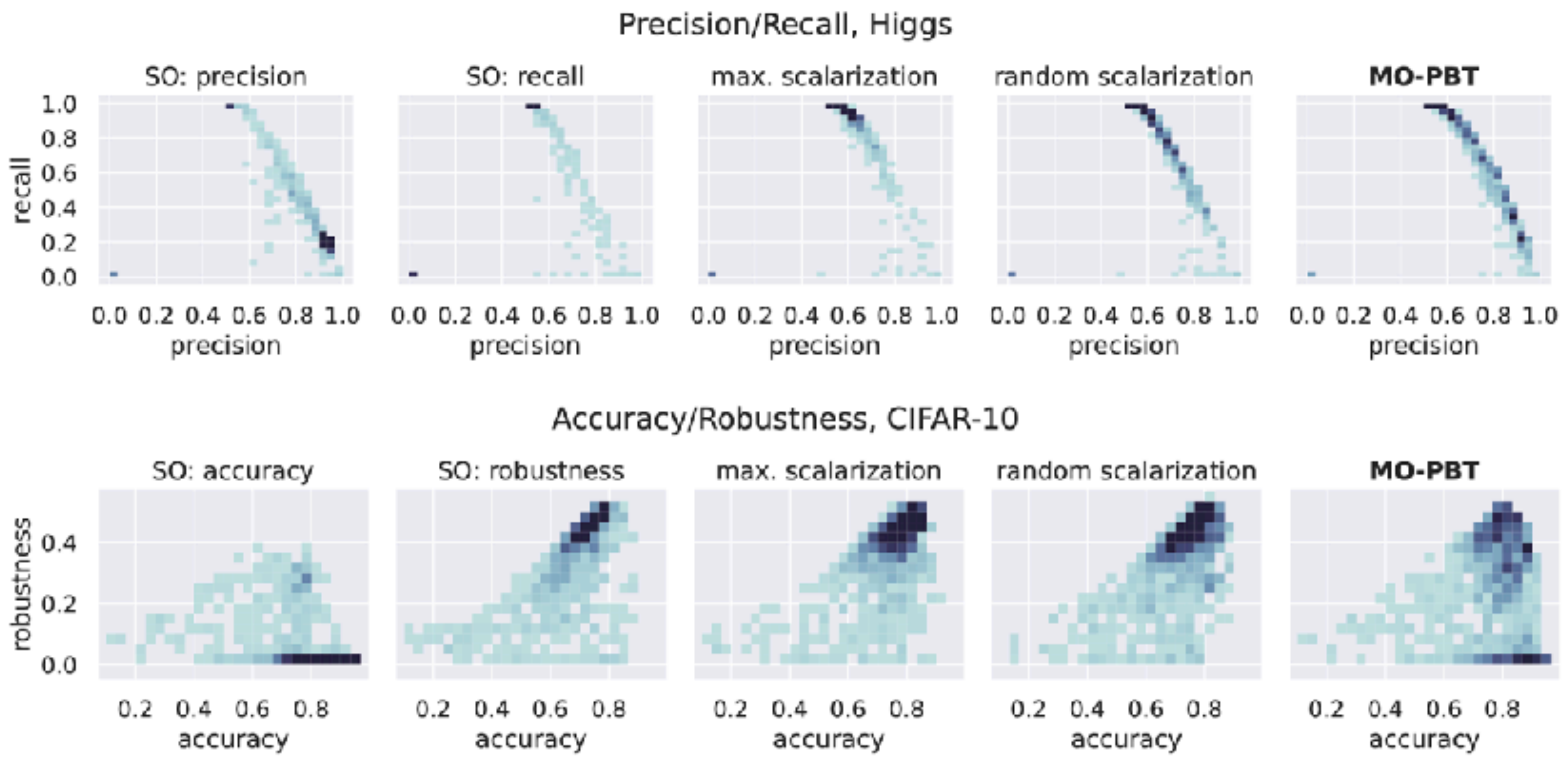


Good at distributing exploration on the Pareto Front

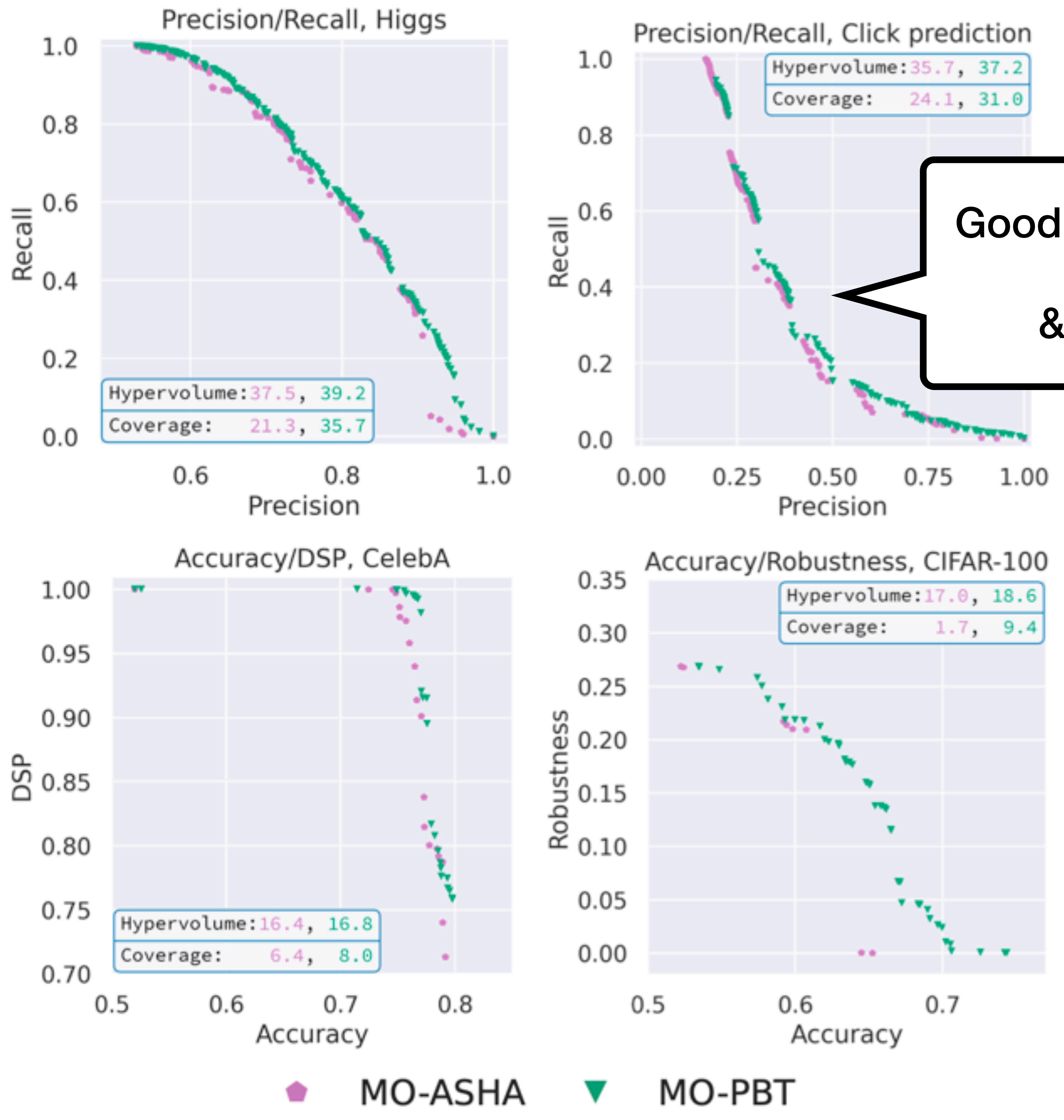


Multiobjective Population Based Training

- Multiobjective Population Based Training: uses non-dominated sort with PBT
- Extend PBT to multi objectives



Good at distributing exploration
on the Pareto Front



Good at distributing exploration
on the Pareto Front
& Better than MOASHA

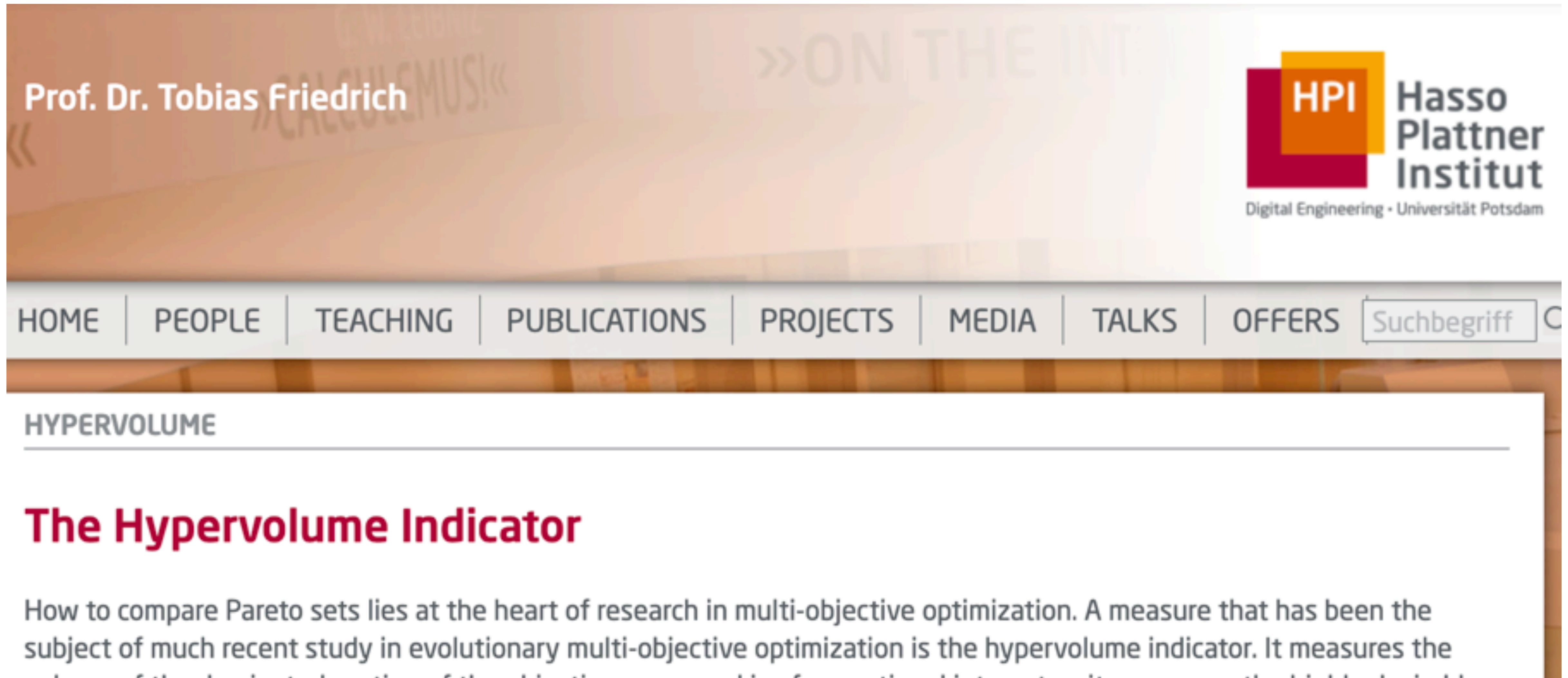
Some theoretical foundations

Some theoretical foundations

- Computational complexity of multiobjective quantities
- Regret bounds on scalarization methods
- Difficulties of high-dimensional multiobjective optimization
- Link with multivariate analysis and Copula

Theoretical foundations of multiobjective optimization

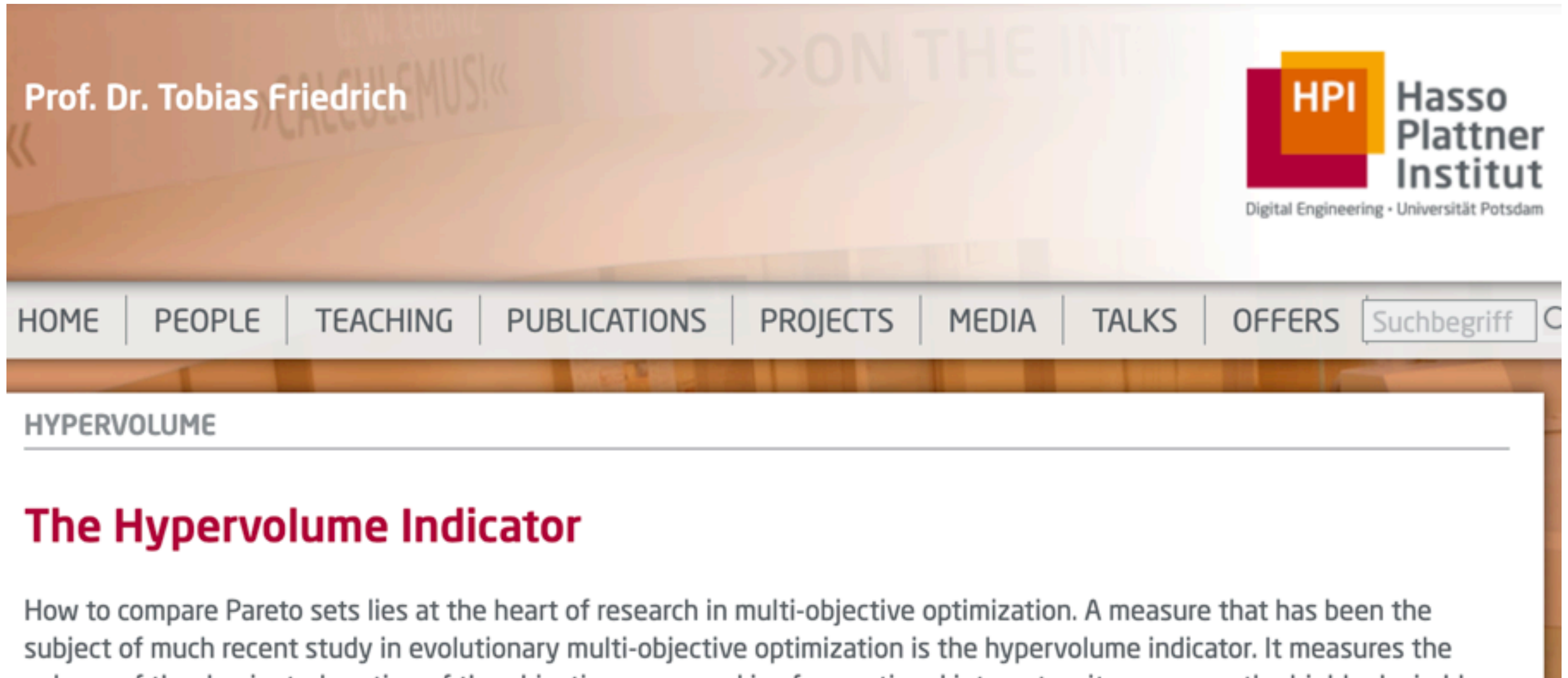
Computability results



Theoretical foundations of multiobjective optimization

Computability results

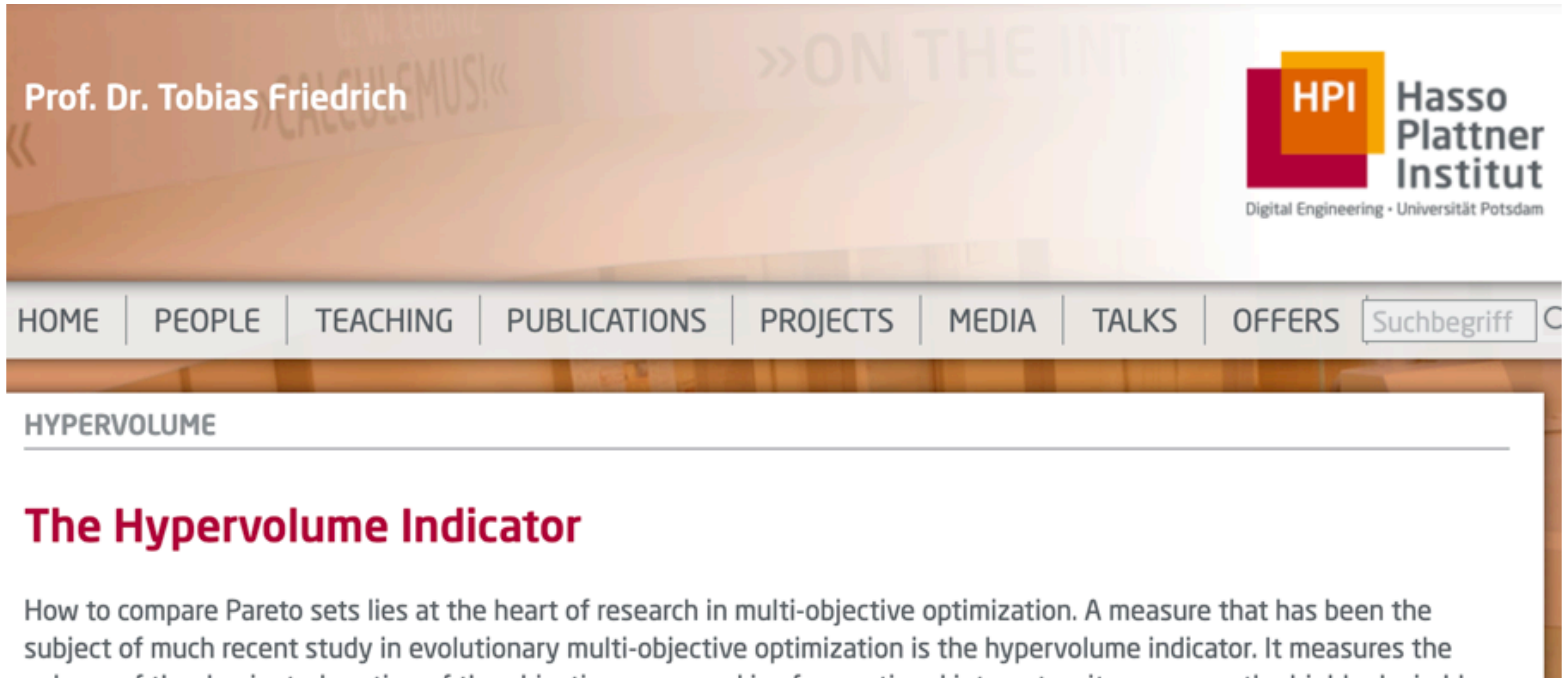
- If P $\neq$ NP then the hypervolume cannot be computed in polynomial time [Bringmann 2013]



Theoretical foundations of multiobjective optimization

Computability results

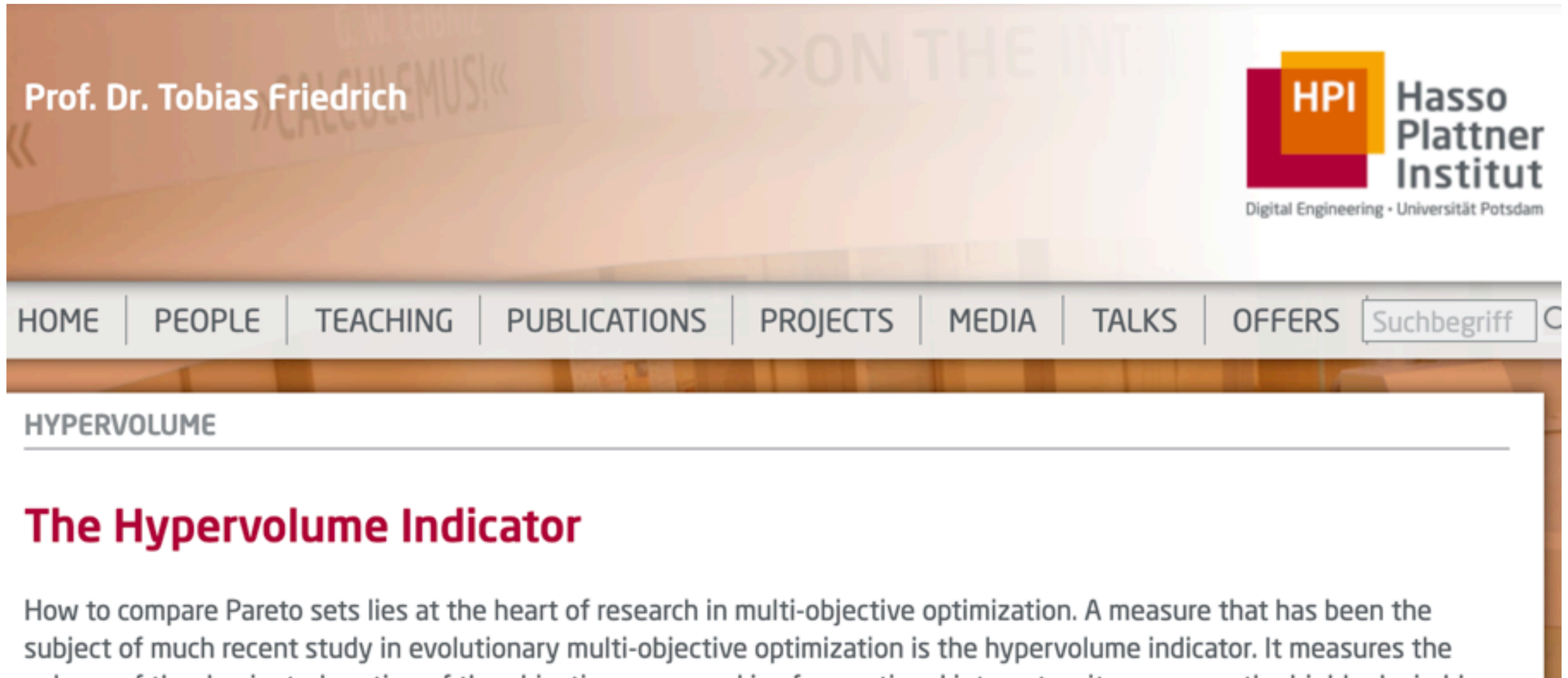
- If P $\neq$ NP then the hypervolume cannot be computed in polynomial time [Bringmann 2013]
- Assuming exponential hypothesis, the hypervolume of n points with d objectives can only be computed in $n^{\Omega(d)}$ [Bringmann 2013]



Theoretical foundations of multiobjective optimization

Computability results

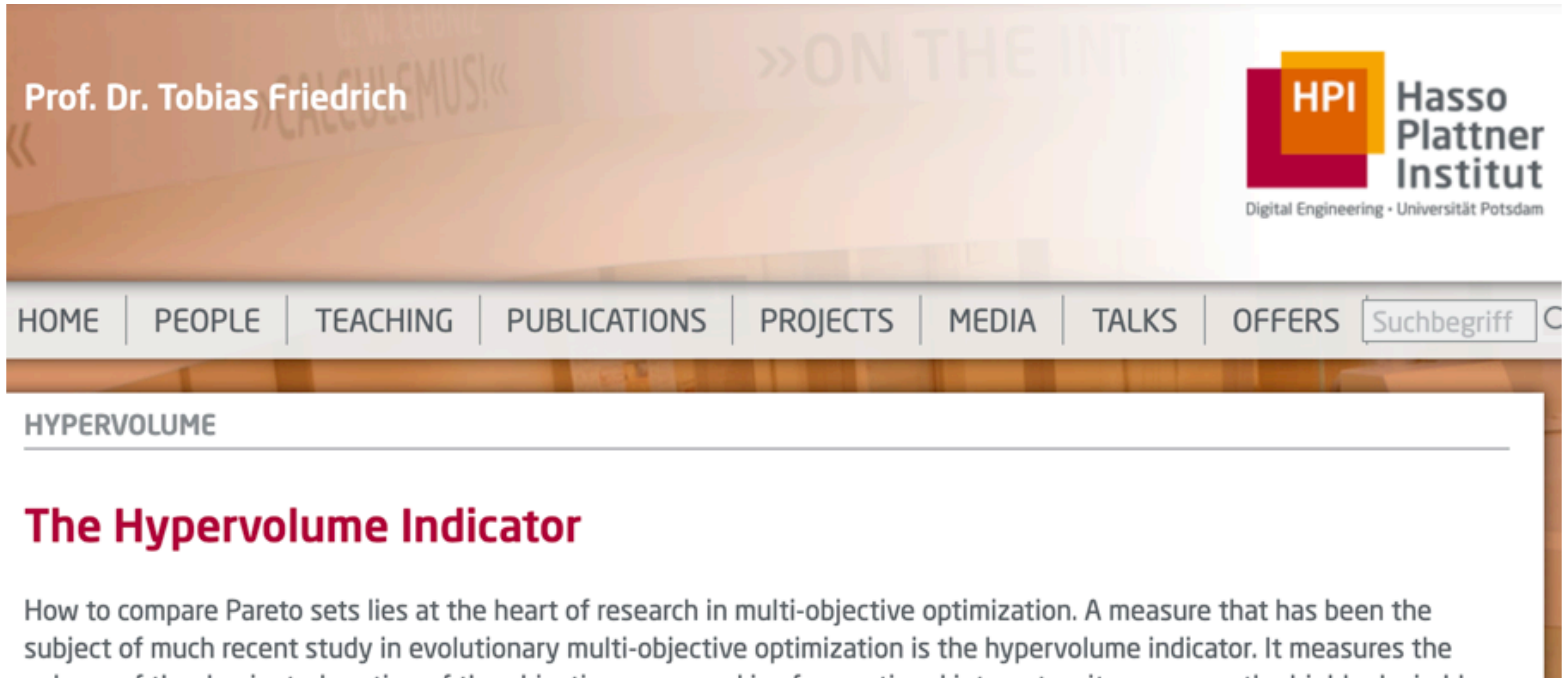
- If P $\neq$ NP then the hypervolume cannot be computed in polynomial time [Bringmann 2013]
- Assuming exponential hypothesis, the hypervolume of n points with d objectives can only be computed in $n^{\Omega(d)}$ [Bringmann 2013]
- However, approximations of the hypervolume can be computed in polynomial time [Bringmann 2010], implementations exists in *Pygmo*



Theoretical foundations of multiobjective optimization

Computability results

- If P $\neq$ NP then the hypervolume cannot be computed in polynomial time [Bringmann 2013]
- Assuming exponential hypothesis, the hypervolume of n points with d objectives can only be computed in $n^{\Omega(d)}$ [Bringmann 2013]
- However, approximations of the hypervolume can be computed in polynomial time [Bringmann 2010], implementations exists in *Pygmo*
- Digest of computational results: https://hpi.de/friedrich/research/the-hypervolume-indicator.html



Theoretical foundations of multiobjective optimization

Regret bounds

Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization [Golovin 2020]

Theoretical foundations of multiobjective optimization

Regret bounds

- Recall Golovin scalarization: $s_\lambda(y) = \min_{i=1..k} (\max(0, y_i/\lambda_i))^k)$

Theoretical foundations of multiobjective optimization

Regret bounds

- Recall Golovin scalarization: $s_\lambda(y) = \min_{i=1..k} (\max(0,\ y_i/\lambda_i))^k)$

Lemma 5 (Hypervolume as Scalarization). *Let* $Y = \{y_1, \ldots, y_m\}$ *be a set of* m *points in* $\mathbb{R}^k$*. Then, the hypervolume of* Y *with respect to a reference point* z *is given by:*

$$\mathcal{HV}_z(Y) = c_k \mathbb{E}_{\lambda \sim \mathcal{S}_+^{k-1}} \left[\max_{y \in Y} s_\lambda(y - z) \right]$$

where $s_\lambda(y) = \min_i (\max(0, y_i/\lambda_i))^k$ *and* $c_k = \frac{\pi^{k/2}}{2^k\ \Gamma(k/2+1)}$ *is a dimension-independent constant.*

Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization [Golovin 2020]

Theoretical foundations of multiobjective optimization

Regret bounds

- Recall Golovin scalarization: $s_\lambda(y) = \min_{i=1..k} (\max(0,\ y_i/\lambda_i))^k)$

Lemma 5 (Hypervolume as Scalarization). *Let $Y = \{y_1, ..., y_m\}$ be a set of m points in $\mathbb{R}^k$. Then, the hypervolume of Y with respect to a reference point z is given by:*

$$\mathcal{HV}_z(Y) = c_k \mathbb{E}_{\lambda \sim \mathcal{S}_+^{k-1}} \left[\max_{y \in Y} s_\lambda(y - z) \right]$$

where $s_\lambda(y) = \min_i (\max(0, y_i/\lambda_i))^k$ and $c_k = \frac{\pi^{k/2}}{2^k\ \Gamma(k/2+1)}$ is a dimension-independent constant.

This scalarization allows to approximate the hypervolume given many weights λ sampled on a sphere

Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization [Golovin 2020]

Theoretical foundations of multiobjective optimization

Regret bounds

- Recall Golovin scalarization: $s_\lambda(y) = \min_{i=1..k} (\max(0, y_i/\lambda_i))^k)$

Lemma 5 (Hypervolume as Scalarization). *Let $Y = \{y_1, ..., y_m\}$ be a set of m points in $\mathbb{R}^k$. Then, the hypervolume of Y with respect to a reference point z is given by:*

$$\mathcal{HV}_z(Y) = c_k \mathbb{E}_{\lambda \sim \mathcal{S}_+^{k-1}} \left[\max_{y \in Y} s_\lambda(y - z) \right]$$

where $s_\lambda(y) = \min_i (\max(0, y_i/\lambda_i))^k$ and $c_k = \frac{\pi^{k/2}}{2^k \, \Gamma(k/2+1)}$ is a dimension-independent constant.

This scalarization allows to approximate the hypervolume given many weights λ sampled on a sphere

Theorem 7 (Theorem 1 in [PKP18]). *Let each objective $f_i(x)$ for $x \in [0,1]^n$ follow a Gaussian distribution with marginal variances bounded by 1 and observation noises $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ are independent with $\sigma_i^2 \leq \sigma^2 \leq 1$. Let $\gamma_{T,k} \leq \gamma_T$, where $\gamma_{T,k}$ is the MIG for the k-th objective. Running Algorithm 1 with L-Lipschitz scalarizations on either UCB or TS acquisition function produces an expected cumulative regret after T steps that is bounded by:*

$$\mathbb{E}_[R_C(T)] = O(Lkn^{1/2}[\gamma_T T \ln(T)]^{1/2})$$

where the expectation is over choice of λ_t and $\mathcal{GP}$ measure.

Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization [Golovin 2020]

Theoretical foundations of multiobjective optimization

Regret bounds

- Recall Golovin scalarization: $s_\lambda(y) = \min_{i=1..k} (\max(0, y_i/\lambda_i))^k)$

Lemma 5 (Hypervolume as Scalarization). *Let $Y = \{y_1, ..., y_m\}$ be a set of m points in $\mathbb{R}^k$. Then, the hypervolume of Y with respect to a reference point z is given by:*

$$\mathcal{HV}_z(Y) = c_k \mathbb{E}_{\lambda \sim S_+^{k-1}} \left[\max_{y \in Y} s_\lambda(y - z) \right]$$

where $s_\lambda(y) = \min_i (\max(0, y_i/\lambda_i))^k$ and $c_k = \frac{\pi^{k/2}}{2^k \, \Gamma(k/2+1)}$ is a dimension-independent constant.

This scalarization allows to approximate the hypervolume given many weights λ sampled on a sphere

Theorem 7 (Theorem 1 in [PKP18]). *Let each objective $f_i(x)$ for $x \in [0,1]^n$ follow a Gaussian distribution with marginal variances bounded by 1 and observation noises $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ are independent with $\sigma_i^2 \leq \sigma^2 \leq 1$. Let $\gamma_{T,k} \leq \gamma_T$, where $\gamma_{T,k}$ is the MIG for the k-th objective. Running Algorithm 1 with L-Lipschitz scalarizations on either UCB or TS acquisition function produces an expected cumulative regret after T steps that is bounded by:*

$$\mathbb{E}_{[} R_C(T)] = O(Lkn^{1/2}[\gamma_T T \ln(T)]^{1/2})$$

where the expectation is over choice of λ_t and $\mathcal{GP}$ measure.

This allows to show that Bayesian Optimization average regret converges to zero

Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization [Golovin 2020]

Theoretical foundations of multiobjective optimization

Regret bounds

- Recall Golovin scalarization: $s_\lambda(y) = \min_{i=1..k} (\max(0,\ y_i/\lambda_i))^k)$

Lemma 5 (Hypervolume as Scalarization). *Let $Y = \{y_1, ..., y_m\}$ be a set of m points in $\mathbb{R}^k$. Then, the hypervolume of Y with respect to a reference point z is given by:*

$$\mathcal{HV}_z(Y) = c_k \mathbb{E}_{\lambda \sim \mathcal{S}_+^{k-1}} \left[\max_{y \in Y} s_\lambda(y - z) \right]$$

where $s_\lambda(y) = \min_i (\max(0, y_i/\lambda_i))^k$ and $c_k = \frac{\pi^{k/2}}{2^k\, \Gamma(k/2+1)}$ is a dimension-independent constant.

This scalarization allows to approximate the hypervolume given many weights λ sampled on a sphere

Theorem 7 (Theorem 1 in [PKP18]). *Let each objective $f_i(x)$ for $x \in [0,1]^n$ follow a Gaussian distribution with marginal variances bounded by 1 and observation noises $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ are independent with $\sigma_i^2 \leq \sigma^2 \leq 1$. Let $\gamma_{T,k} \leq \gamma_T$, where $\gamma_{T,k}$ is the MIG for the k-th objective. Running Algorithm 1 with L-Lipschitz scalarizations on either UCB or TS acquisition function produces an expected cumulative regret after T steps that is bounded by:*

$$\mathbb{E}_[R_C(T)] = O(Lkn^{1/2}[\gamma_T T \ln(T)]^{1/2})$$

where the expectation is over choice of λ_t and $\mathcal{GP}$ measure.

This allows to show that Bayesian Optimization average regret converges to zero

Theorem 2 (Convergence of Bayesian Optimization with Hypervolume Scalarization: Informal Restatement of Theorem 8). *The cumulative hypervolume regret for using random hypervolume scalarization with UCB or TS after T observations is upper bounded as*

$$\sum_{t=1}^{T} (\mathcal{HV}_z(Y^*) - \mathcal{HV}_z(Y_t)) \leq O(k^2 n^{1/2} [\gamma_T T \ln(T)]^{1/2})$$

Furthermore, $\mathcal{HV}_z(Y_T) \geq \mathcal{HV}_z(Y^*) - \epsilon_T$, where $\epsilon_T = O(k^2 n^{1/2}[\gamma_T \ln(T)/T]^{1/2})$.

Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization [Golovin 2020]

Theoretical foundations of multiobjective optimization

Regret bounds

- Recall Golovin scalarization: $s_\lambda(y) = \min_{i=1..k} (\max(0, y_i/\lambda_i))^k)$

Lemma 5 (Hypervolume as Scalarization). *Let $Y = \{y_1, ..., y_m\}$ be a set of m points in $\mathbb{R}^k$. Then, the hypervolume of Y with respect to a reference point z is given by:*

$$\mathcal{HV}_z(Y) = c_k \mathbb{E}_{\lambda \sim S_+^{k-1}} \left[\max_{y \in Y} s_\lambda(y - z) \right]$$

where $s_\lambda(y) = \min_i (\max(0, y_i/\lambda_i))^k$ and $c_k = \frac{\pi^{k/2}}{2^k \, \Gamma(k/2+1)}$ is a dimension-independent constant.

This scalarization allows to approximate the hypervolume given many weights λ sampled on a sphere

Theorem 7 (Theorem 1 in [PKP18]). *Let each objective $f_i(x)$ for $x \in [0,1]^n$ follow a Gaussian distribution with marginal variances bounded by 1 and observation noises $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ are independent with $\sigma_i^2 \le \sigma^2 \le 1$. Let $\gamma_{T,k} \le \gamma_T$, where $\gamma_{T,k}$ is the MIG for the k-th objective. Running Algorithm 1 with L-Lipschitz scalarizations on either UCB or TS acquisition function produces an expected cumulative regret after T steps that is bounded by:*

$$\mathbb{E}_[R_C(T)] = O(Lkn^{1/2}[\gamma_T T \ln(T)]^{1/2})$$

where the expectation is over choice of λ_t and $\mathcal{GP}$ measure.

This allows to show that Bayesian Optimization average regret converges to zero

Theorem 2 (Convergence of Bayesian Optimization with Hypervolume Scalarization: Informal Restatement of Theorem 8). *The cumulative hypervolume regret for using random hypervolume scalarization with UCB or TS after T observations is upper bounded as*

$$\sum_{t=1}^{T} (\mathcal{HV}_z(Y^*) - \mathcal{HV}_z(Y_t)) \le O(k^2 n^{1/2} [\gamma_T T \ln(T)]^{1/2})$$

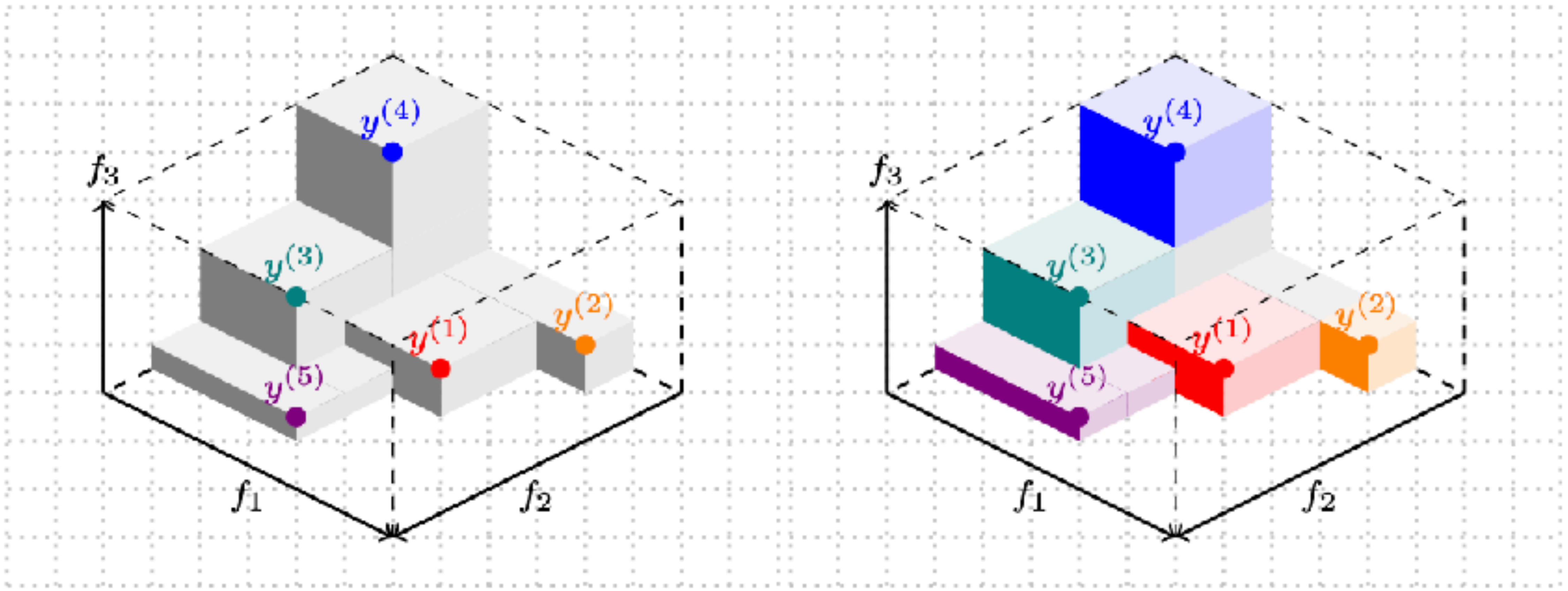
Furthermore, $\mathcal{HV}_z(Y_T) \ge \mathcal{HV}_z(Y^*) - \epsilon_T$, where $\epsilon_T = O(k^2 n^{1/2} [\gamma_T \ln(T)/T]^{1/2})$.

Informal statement, the average Hypervolume regret obtained with Bayesian Optimization goes to zero

Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization [Golovin 2020]

Theoretical foundations of multiobjective optimization

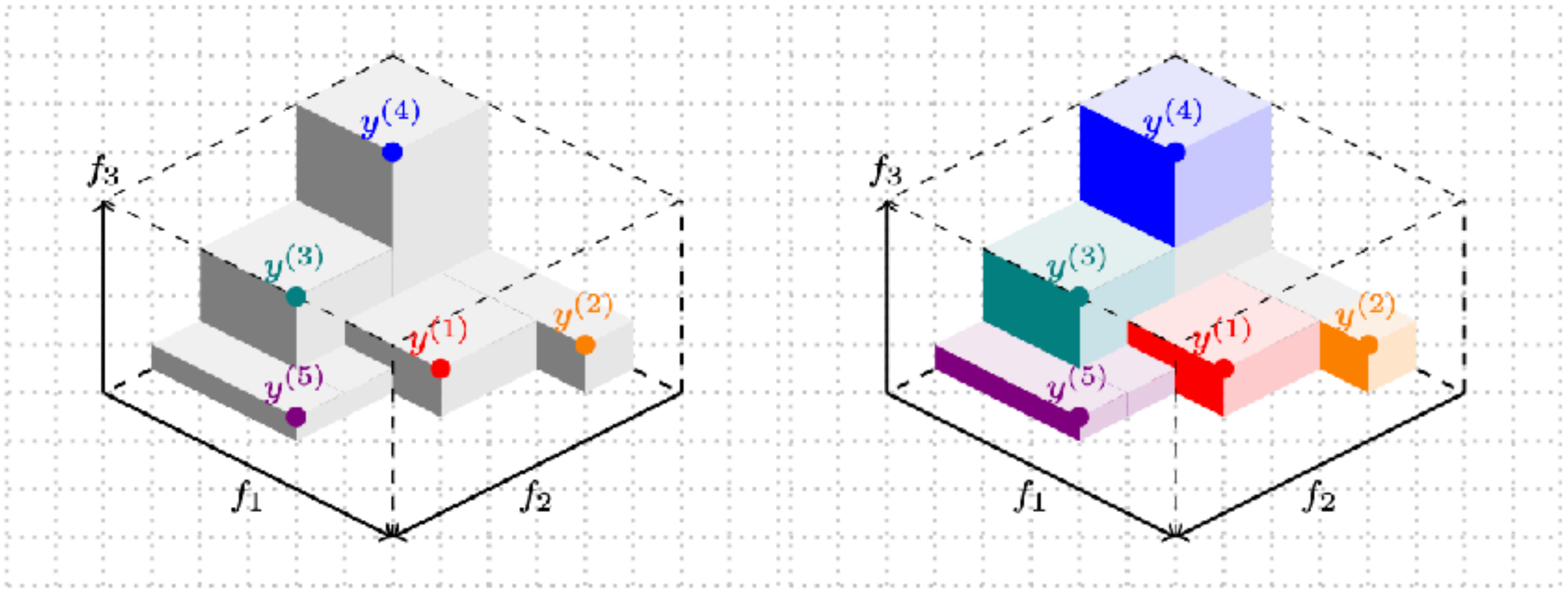
Difficulties of high-dimensional multiobjective optimization



Theoretical foundations of multiobjective optimization

Difficulties of high-dimensional multiobjective optimization

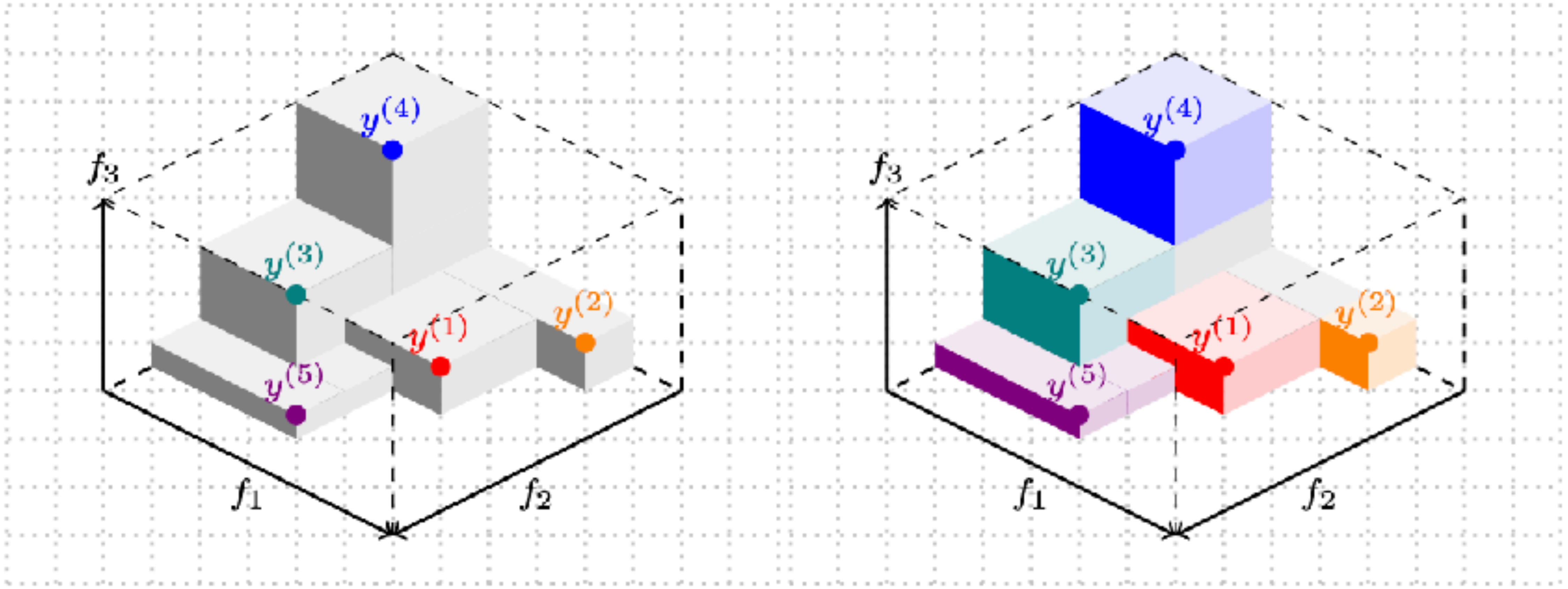
- What about optimizing many objectives?



Theoretical foundations of multiobjective optimization

Difficulties of high-dimensional multiobjective optimization

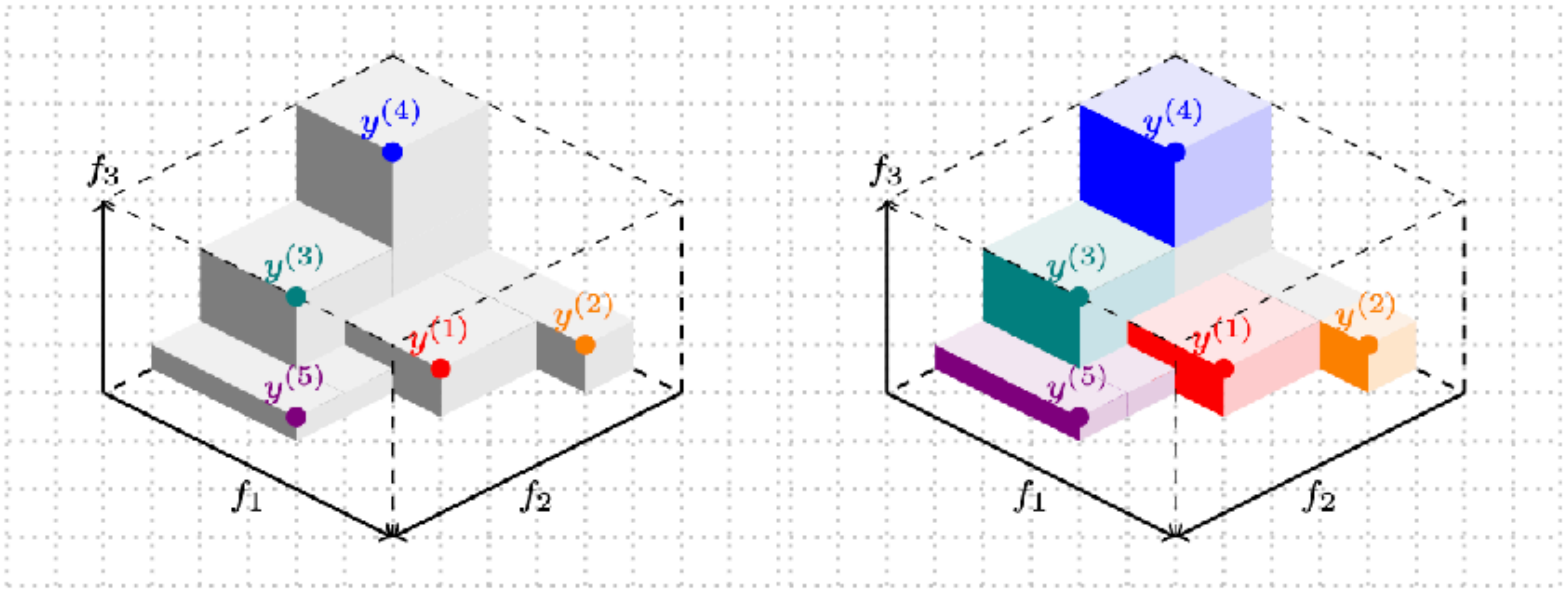
- What about optimizing many objectives?
    - Hard to visualise problems with more than 3 objectives



Theoretical foundations of multiobjective optimization

Difficulties of high-dimensional multiobjective optimization

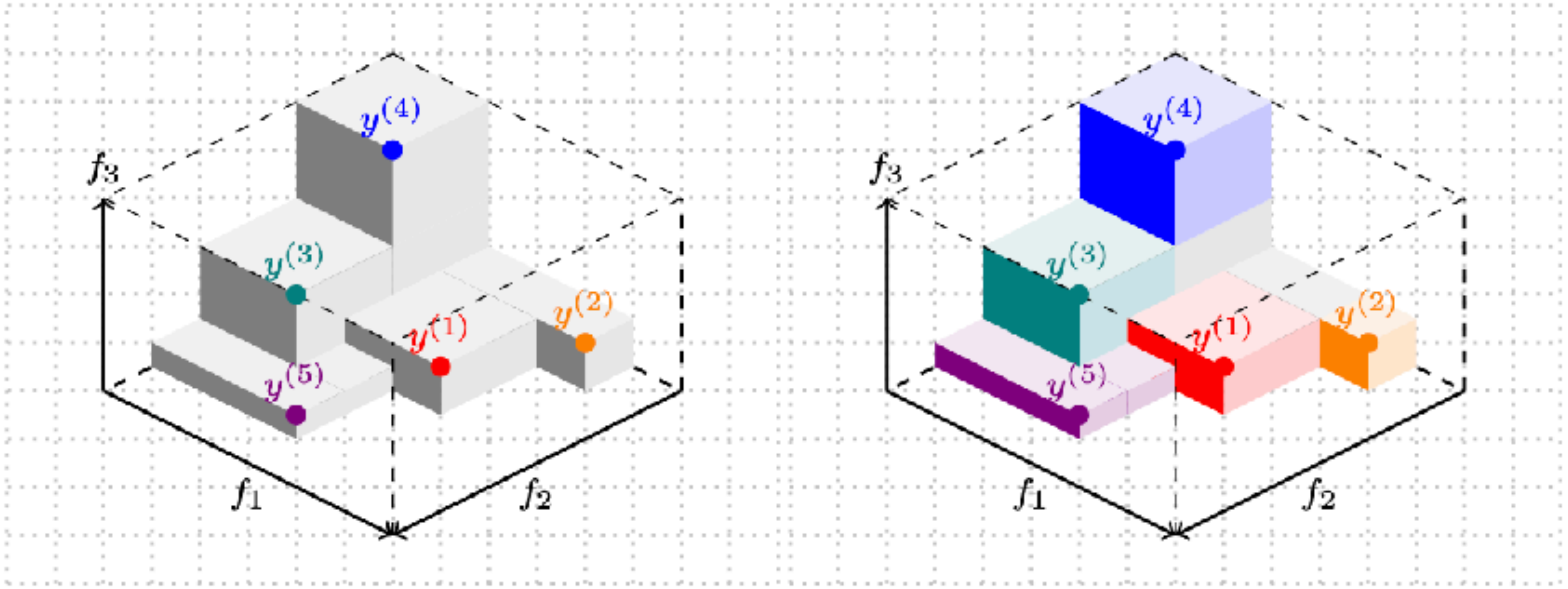
- What about optimizing many objectives?
  - Hard to visualise problems with more than 3 objectives
  - If P $\neq$ NP, the complexity of the hypervolume indicator computation grows super-polynomially with the number of objectives ($n^{\Omega(d)}$)



Theoretical foundations of multiobjective optimization

Difficulties of high-dimensional multiobjective optimization

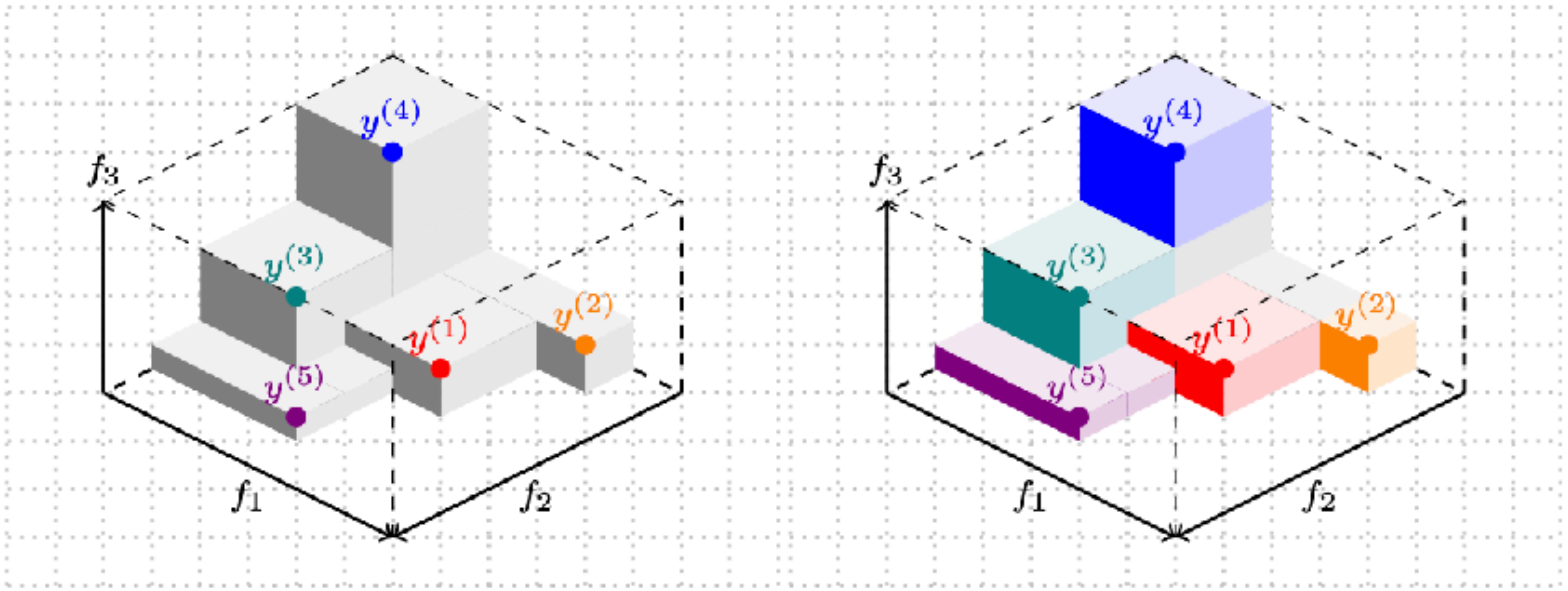
- What about optimizing many objectives?
  - Hard to visualise problems with more than 3 objectives
  - If P $\neq$ NP, the complexity of the hypervolume indicator computation grows super-polynomially with the number of objectives ($n^{\Omega(d)}$)
  - The ratio of non-dominated points increases rapidly with the number of objectives



Theoretical foundations of multiobjective optimization

Difficulties of high-dimensional multiobjective optimization

- What about optimizing many objectives?
  - Hard to visualise problems with more than 3 objectives
  - If P $\neq$ NP, the complexity of the hypervolume indicator computation grows super-polynomially with the number of objectives ($n^{\Omega(d)}$)
  - The ratio of non-dominated points increases rapidly with the number of objectives



For instance, the probability that a point is non-dominated in a uniformly distributed set of sample points grows exponentially fast towards 1 with the number of objectives. [Emmerich 2018]

Theoretical foundations of multiobjective optimization

Link with Copula theory

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

- Given a subset $G \subset \mathbb{R}^d$, the Pareto front $\mathscr{P}$ is defined by:

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

- Given a subset $G \subset \mathbb{R}^d$, the Pareto front $\mathscr{P}$ is defined by:
  - $\mathscr{P} = \{y \in G \mid \forall z \in G, y \preceq z\}$

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory
Mickaël Binois, Didier Rullière, Olivier Roustant

- Given a subset $G \subset \mathbb{R}^d$, the Pareto front $\mathscr{P}$ is defined by:
  - $\mathscr{P} = \{y \in G \mid \forall z \in G, y \preceq z\}$

Set of points in G that are not dominated

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory
Mickaël Binois, Didier Rullière, Olivier Roustant

- Given a subset $G \subset \mathbb{R}^d$, the Pareto front $\mathscr{P}$ is defined by:
  - $\mathscr{P} = \{y \in G \mid \forall z \in G, y \preceq z\}$ — Set of points in G that are not dominated
- Link with the multivariate cumulative distribution $F_Y(y) = \mathbb{P}[Y \preceq y]$

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory
Mickaël Binois, Didier Rullière, Olivier Roustant

- Given a subset $G \subset \mathbb{R}^d$, the Pareto front $\mathscr{P}$ is defined by:
  - $\mathscr{P} = \{y \in G \mid \forall z \in G, y \preceq z\}$ — Set of points in G that are not dominated
- Link with the multivariate cumulative distribution $F_Y(y) = \mathbb{P}[Y \preceq y]$
  - $y \in \mathscr{P} \Rightarrow F_Y(y) = 0$

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory
Mickaël Binois, Didier Rullière, Olivier Roustant

- Given a subset $G \subset \mathbb{R}^d$, the Pareto front $\mathscr{P}$ is defined by:
  - $\mathscr{P} = \{y \in G \mid \forall z \in G, y \preceq z\}$

    Set of points in G that are not dominated
- Link with the multivariate cumulative distribution $F_Y(y) = \mathbb{P}[Y \preceq y]$
  - $y \in \mathscr{P} \Rightarrow F_Y(y) = 0$

    The Pareto front belongs to the zero level set of the CDF F_Y

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

- Given a subset $G \subset \mathbb{R}^d$, the Pareto front $\mathscr{P}$ is defined by:
  - $\mathscr{P} = \{y \in G \mid \forall z \in G, y \preceq z\}$

Set of points in G that are not dominated

- Link with the multivariate cumulative distribution $F_Y(y) = \mathbb{P}[Y \preceq y]$
  - $y \in \mathscr{P} \Rightarrow F_Y(y) = 0$

The Pareto front belongs to the zero level set of the CDF F_Y

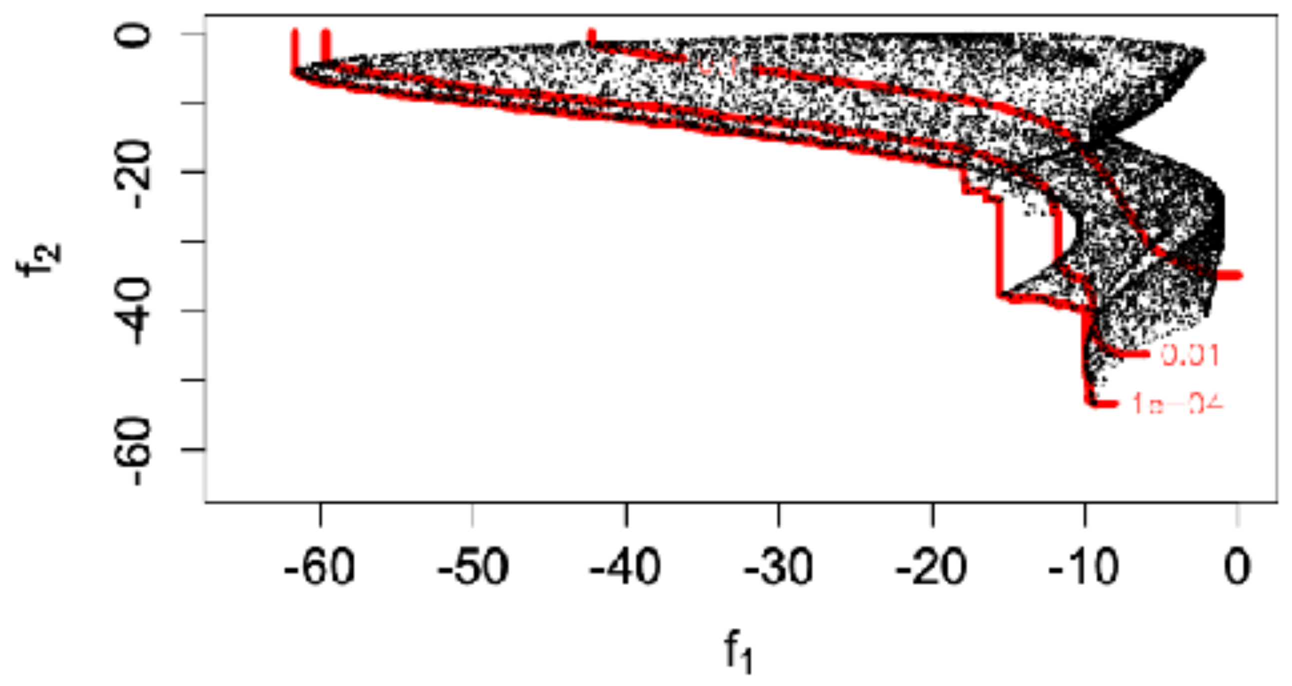


Figure 2: Level lines ∂L_α^F with $\alpha = 0.0001, 0.01, 0.1$ of the empirical cumulative distribution function of $\mathbf{f}(\mathbf{X})$ obtained with sampled points (in black), showing the link between the level line of level α and the Pareto front $\mathcal{P}$ (apart from the vertical and horizontal components), as α tends to zero.

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

- Given a subset $G \subset \mathbb{R}^d$, the Pareto front $\mathscr{P}$ is defined by:
  - $\mathscr{P} = \{y \in G \mid \forall z \in G, y \preceq z\}$

    Set of points in G that are not dominated
- Link with the multivariate cumulative distribution $F_Y(y) = \mathbb{P}[Y \preceq y]$
  - $y \in \mathscr{P} \Rightarrow F_Y(y) = 0$

The Pareto front belongs to the zero level set of the CDF F_Y

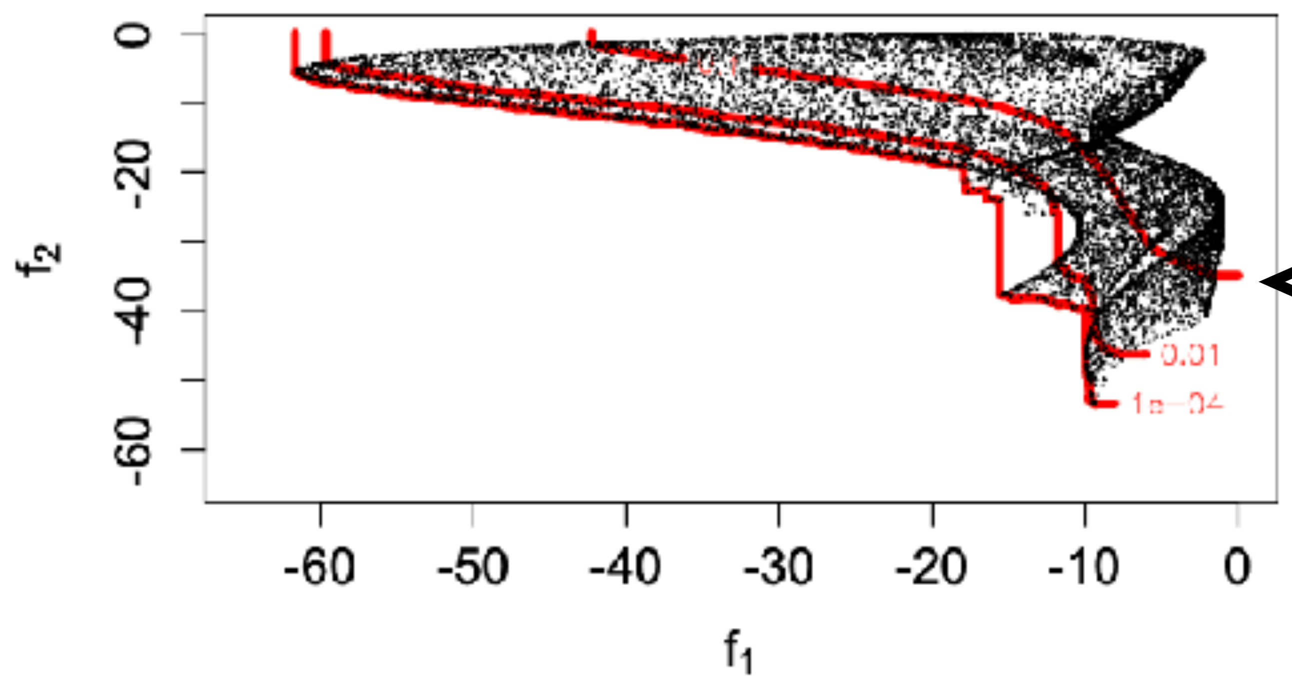


Figure 2: Level lines ∂L_α^F with $\alpha = 0.0001, 0.01, 0.1$ of the empirical cumulative distribution function of $\mathbf{f}(\mathbf{X})$ obtained with sampled points (in black), showing the link between the level line of level α and the Pareto front $\mathcal{P}$ (apart from the vertical and horizontal components), as α tends to zero.

The level set of F_Y, $\partial L_\alpha^F = \{y \in \mathbb{R}^d \mid F_Y(y) = \alpha\}$ converges to the Pareto front $\mathscr{P}$

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

- Copula are central in multivariate analysis

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem

For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:

$$F_Y(y_1, \ldots, y_d) = C(F_1(y_1), \ldots, F_d(y_d))$$

- Copula are central in multivariate analysis

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem
For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:
$$F_Y(y_1, \ldots, y_d) = C(F_1(y_1), \ldots, F_d(y_d))$$

$C : [0,1]^d \rightarrow [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

- Copula are central in multivariate analysis

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem

For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:

$$F_Y(y_1, \ldots, y_d) = C(F_1(y_1), \ldots, F_d(y_d))$$

$C : [0,1]^d \rightarrow [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

- Copula are central in multivariate analysis

- They allow to decouple the scaling effect on each variable F_i with the effect on the joint distribution 🥳

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem

For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:

$$F_Y(y_1, \ldots, y_d) = C(F_1(y_1), \ldots, F_d(y_d))$$

$C : [0,1]^d \to [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

- Copula are central in multivariate analysis
- They allow to decouple the scaling effect on each variable F_i with the effect on the joint distribution 🥳
- The α-level lines of C, i.e. $\{u \in [0,1]^d \mid C(u_1, \ldots, u_d) = \alpha\}$ are denoted ∂L_C^α

Theoretical foundations of multiobjective optimization

Link with Copula theory

Sklar theorem
For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:
$$F_Y(y_1, \ldots, y_d) = C(F_1(y_1), \ldots, F_d(y_d))$$

On the estimation of Pareto fronts from the point of view of copula theory
Mickaël Binois, Didier Rullière, Olivier Roustant

$C : [0,1]^d \to [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

- Copula are central in multivariate analysis
- They allow to decouple the scaling effect on each variable F_i with the effect on the joint distribution 🥳
- The α-level lines of C, i.e. $\{u \in [0,1]^d \mid C(u_1, \ldots, u_d) = \alpha\}$ are denoted ∂L_C^α
- The level lines of the Copula ∂L_C^α can be connected to the level lines of the multivariate CDF as follow:

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem
For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:
$F_Y(y_1, \ldots, y_d) = C(F_1(y_1), \ldots, F_d(y_d))$

$C : [0,1]^d \to [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

- Copula are central in multivariate analysis
- They allow to decouple the scaling effect on each variable F_i with the effect on the joint distribution 🥳
- The α-level lines of C, i.e. $\{u \in [0,1]^d \mid C(u_1, \ldots, u_d) = \alpha\}$ are denoted ∂L_C^α
- The level lines of the Copula ∂L_C^α can be connected to the level lines of the multivariate CDF as follow:
  - $\partial L_\alpha^F = \{(y_1, \ldots, y_d) = (F_1^{-1}(u_1), \ldots, F_m^{-1}(u_m)) \in \mathbb{R}^d, u \in \partial L_\alpha^C\}$

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem
For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:
$F_Y(y_1, \ldots, y_d) = C(F_1(y_1), \ldots, F_d(y_d))$

$C : [0,1]^d \to [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

- Copula are central in multivariate analysis
- They allow to decouple the scaling effect on each variable F_i with the effect on the joint distribution 🥳
- The α-level lines of C, i.e. $\{u \in [0,1]^d \mid C(u_1, \ldots, u_d) = \alpha\}$ are denoted ∂L_C^α
- The level lines of the Copula ∂L_C^α can be connected to the level lines of the multivariate CDF as follow:
  - $\partial L_\alpha^F = \{(y_1, \ldots, y_d) = (F_1^{-1}(u_1), \ldots, F_m^{-1}(u_m)) \in \mathbb{R}^d, u \in \partial L_\alpha^C\}$
- In addition to connect fundamental aspect of multivariate analysis, those connection allow to model the Pareto front by modelling Copulas

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem

For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:

$F_Y(y_1,$

$C : [0,1]^d \rightarrow [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

- Copula are
- They allow … h the effect on the joint distribution
- The α-level … denoted ∂L_C^α
- The level lin … ines of the multivariate CDF as follow:
- $\partial L_\alpha^F =$ … $L_\alpha^C\}$

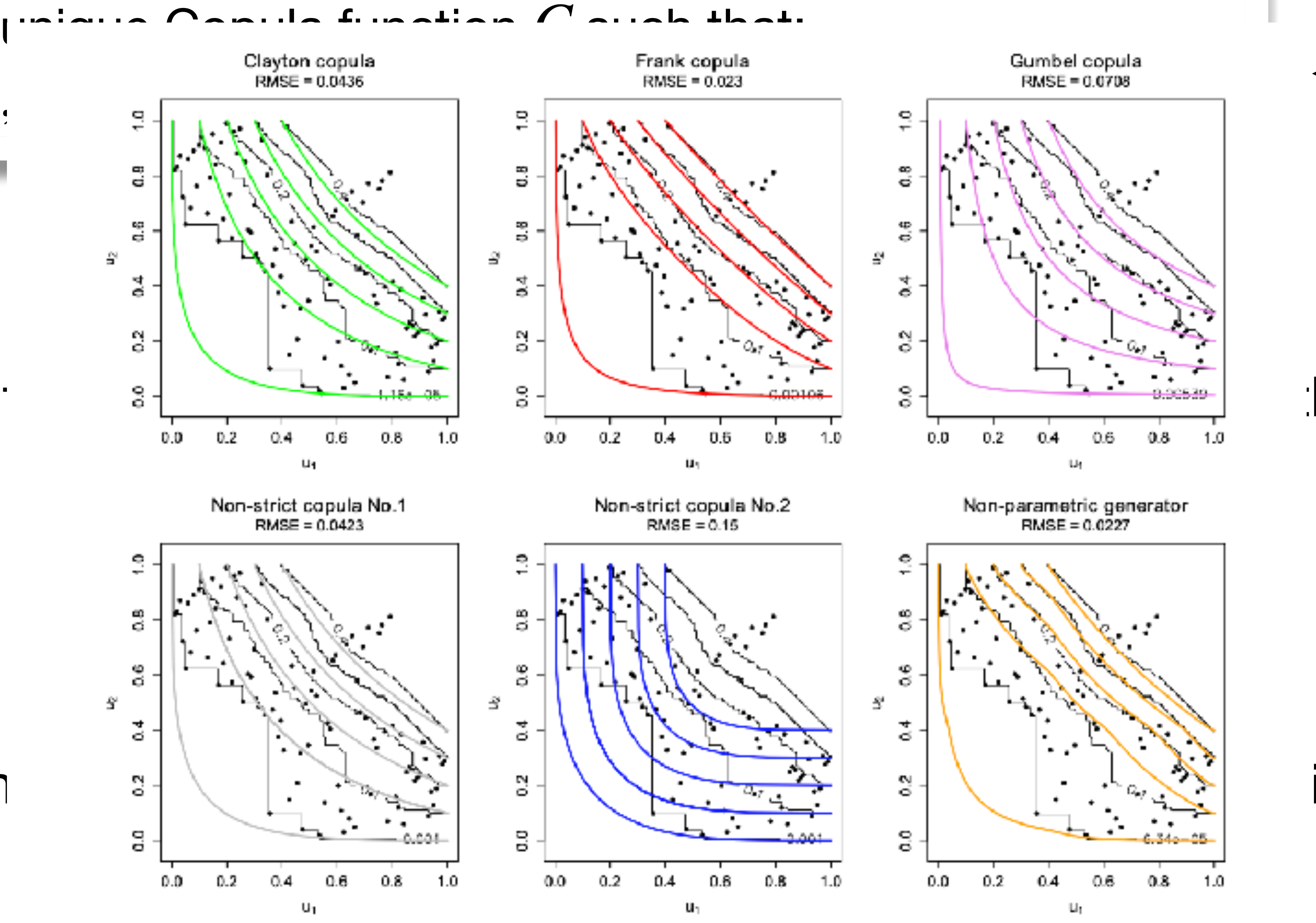


Figure 12: Levels lines $\partial L_\alpha^{C_\phi}$ of the different fitted Archimedean models based on the pseudo-data U^k, $k = 1, \ldots, n$, from test problem Poloni. The level lines correspond in each case to α^*, 0.1, 0.2, 0.3 and 0.4.

- In addition to connect fundamental aspect of multivariate analysis, those connection allow to model the Pareto front by modelling Copulas

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem

For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:

$F_Y(y_1,$

$C : [0,1]^d \rightarrow [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

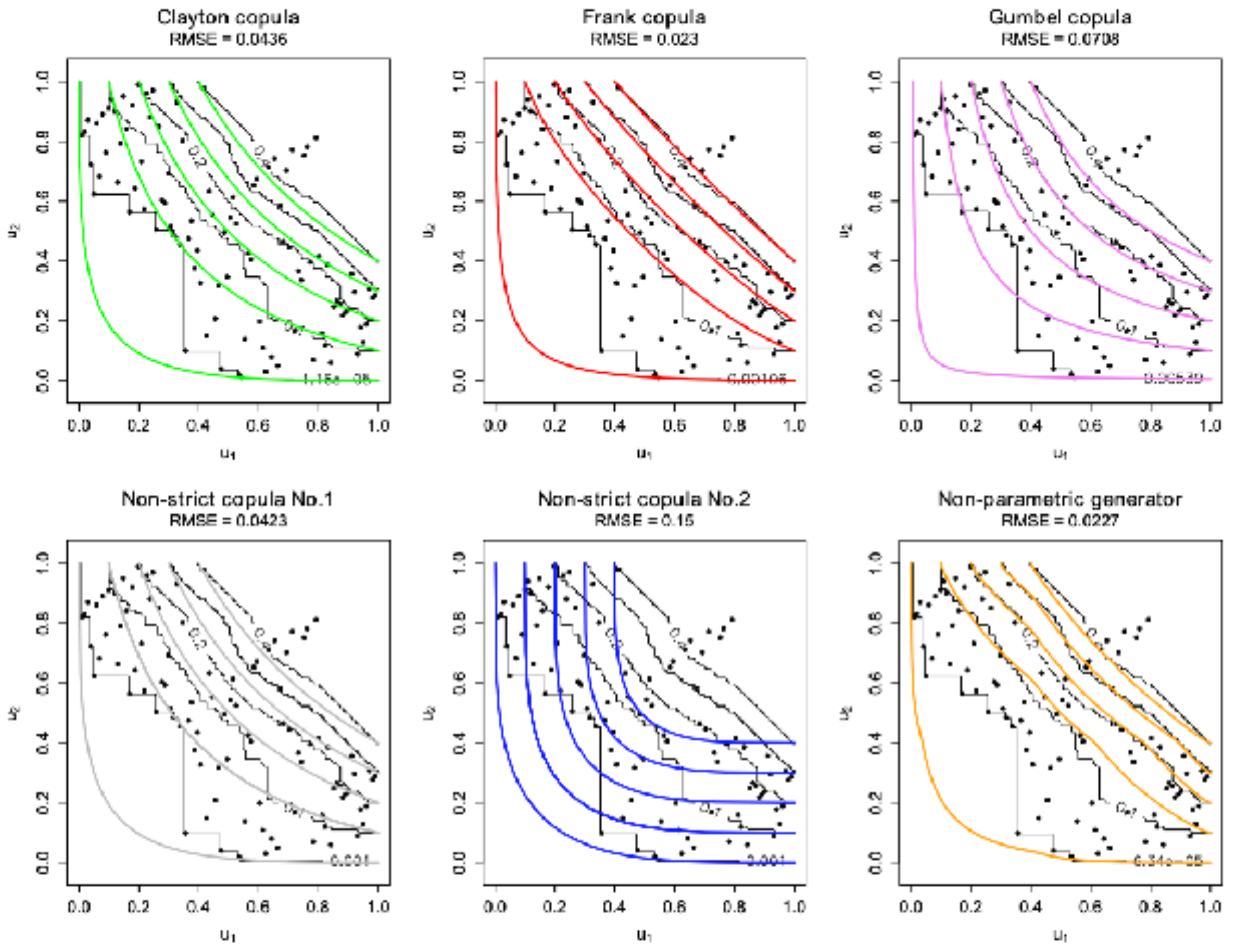


Figure 12: Levels lines $\partial L_\alpha^{C_\phi}$ of the different fitted Archimedean models based on the pseudo-data U^k, $k = 1, \ldots, n$, from test problem Poloni. The level lines correspond in each case to α^*, 0.1, 0.2, 0.3 and 0.4.

- Copula are
- They allow … h the effect on the joint distribution
- The α-level … denoted ∂L_C^α
- The level lin … ines of the multivariate CDF as follow:
- $\partial L_\alpha^F =$ … $L_\alpha^C\}$
- In addition to … , those connection allow to model the Pa…

One can model the Copula after having transformed each variables with the respective CDF, different modelling options are shown here…

Theoretical foundations of multiobjective optimization

Link with Copula theory

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

Sklar theorem

For any continuous multivariate distribution function F_Y, there exists a u…

$F_Y(y_1, …$

$C : [0,1]^d \rightarrow [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

- Copula are …
- They allow … …h the distribution
- The α-level … …den…
- The level lin… …ines as follow:
- $\partial L_\alpha^F =$ … $L_\alpha^C\}$
- In addition to … , those connection allow to model the Pa…

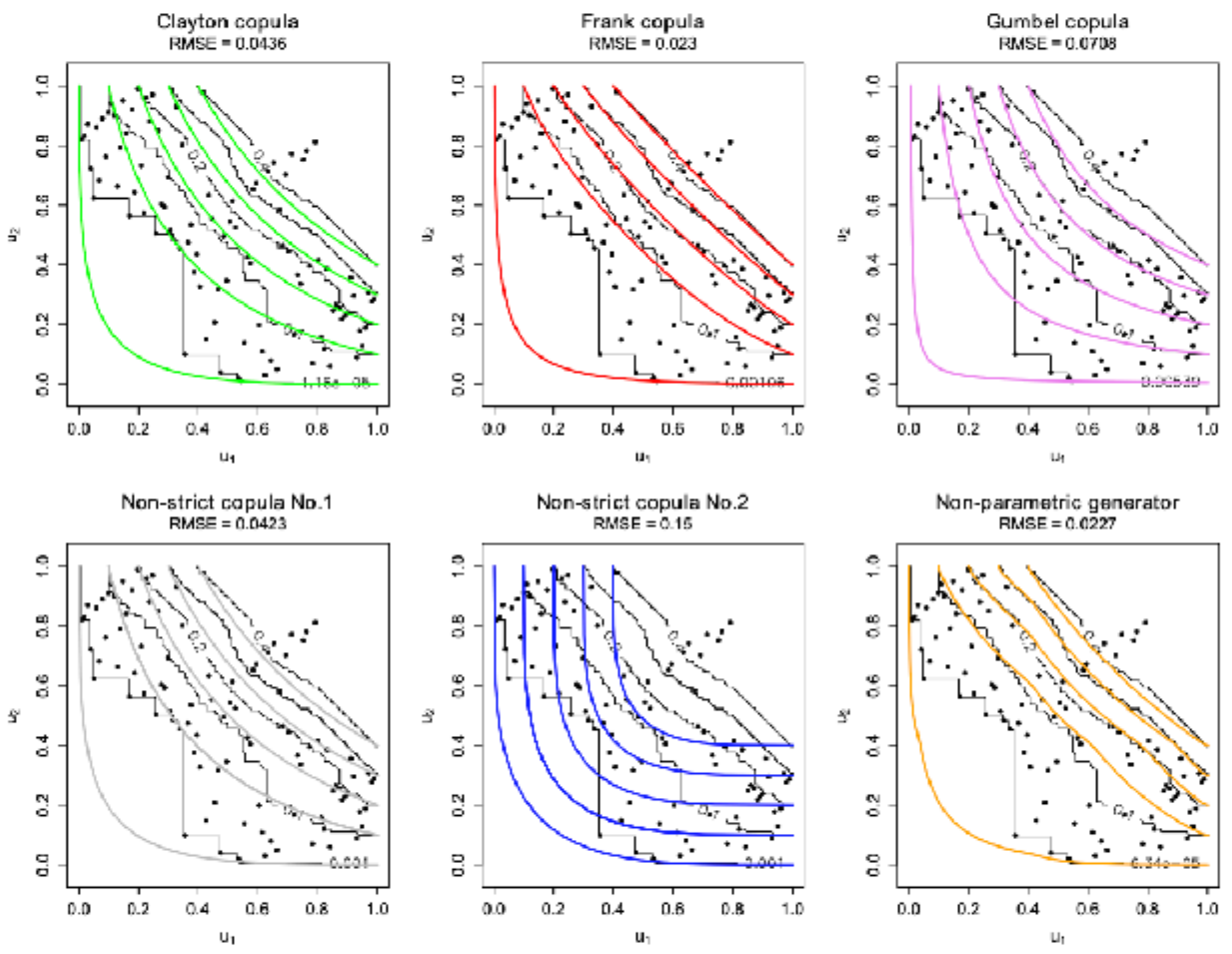


Figure 12: Levels lines $\partial L_\alpha^{C_\phi}$ of the different fitted Archimedean models based on the pseudo-data U^k, $k = 1, \ldots, n$, from test problem Poloni. The level lines correspond in each case to α^*, 0.1, 0.2, 0.3 and 0.4.

One can model the Copula after having transformed each variables with the respective CDF, different modelling options are shown here…

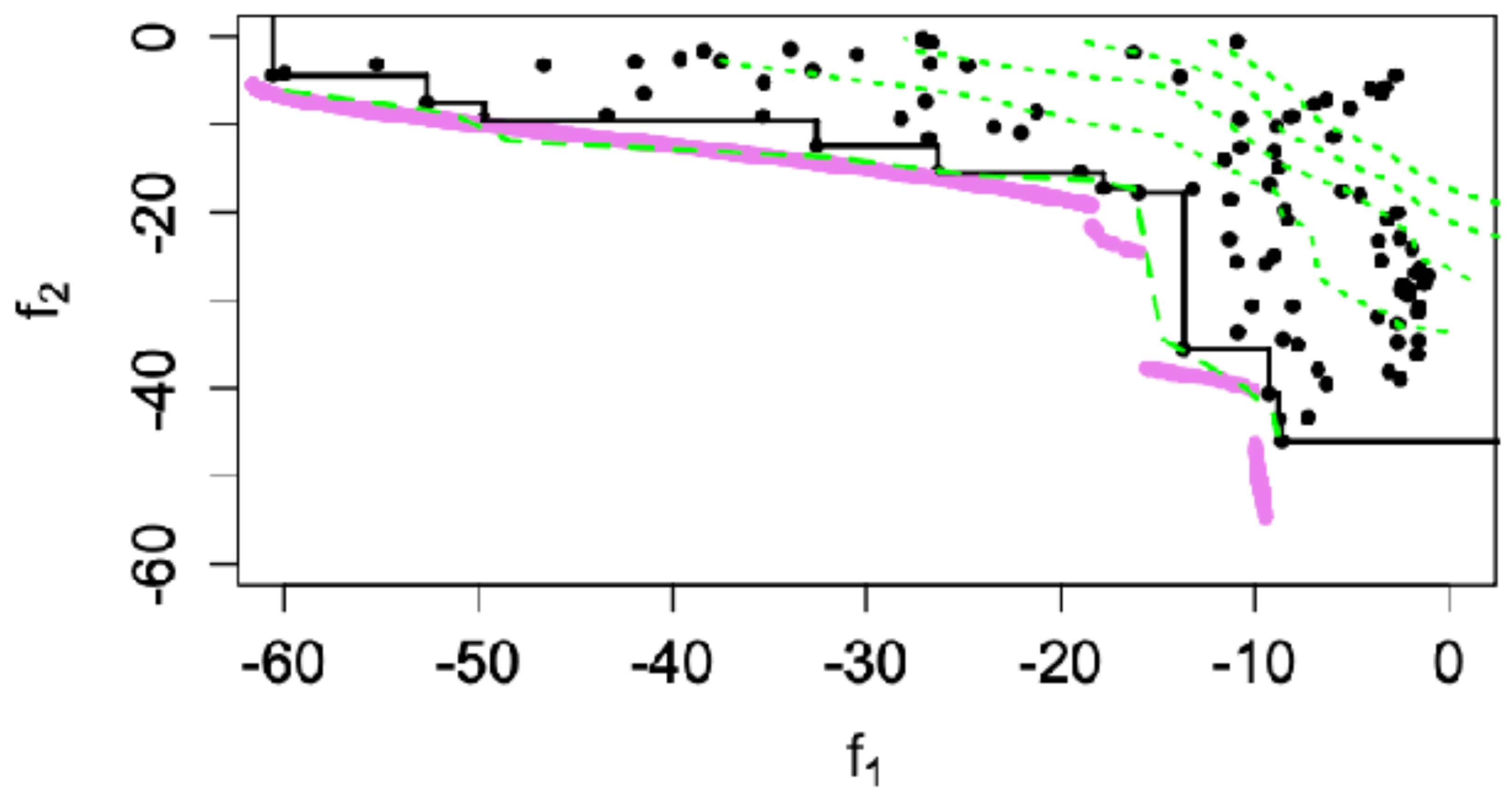


Figure 13: Estimated level line $\partial L_{\alpha^*}^F$ for the Poloni test problem (green dashed line), compared to the Pareto front approximation from the observation $\mathcal{P}_n$ (black line) and the true Pareto front $\mathcal{P}$ (violet solid line). Other level lines with levels 0.1, 0.2, 0.3 and 0.4 are also displayed with thinner lines.

Theoretical foundations of multiobjective optimization

Link with Copula theory

Sklar theorem

For any continuous multivariate distribution function F_Y, there exists a unique Copula function C such that:

$F_Y(y_1,$

- Copula are
- They allow ... h the distribution
- The α-level ... den
- The level lin ... ines as follow:
- $\partial L_\alpha^F = \dots L_\alpha^C\}$
- In addition to ..., those model the Pa

On the estimation of Pareto fronts from the point of view of copula theory

Mickaël Binois, Didier Rullière, Olivier Roustant

$C : [0,1]^d \rightarrow [0,1]$ is d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals

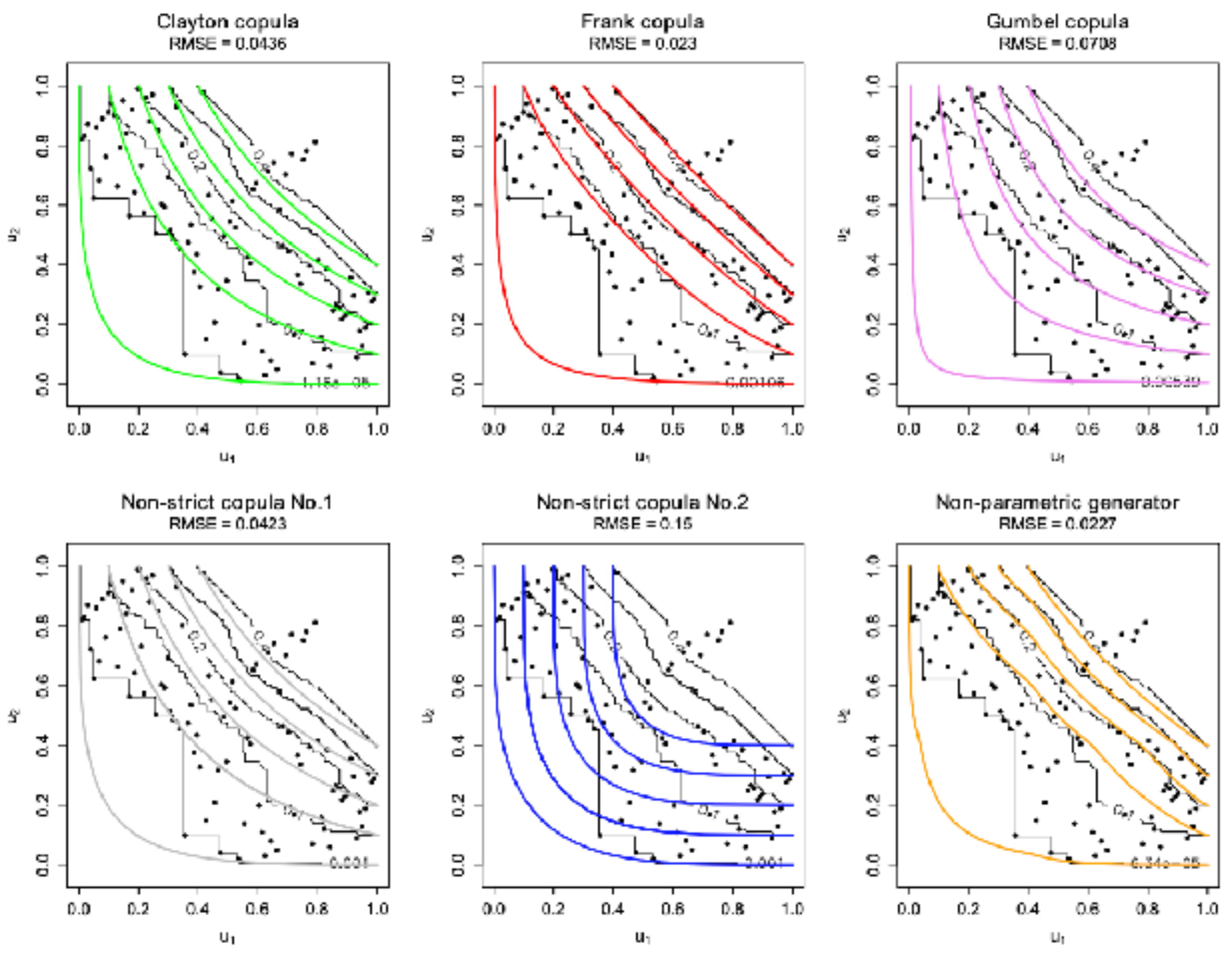


Figure 12: Levels lines $\partial L_\alpha^{C_\phi}$ of the different fitted Archimedean models based on the pseudo-data U^k, $k = 1, \dots, n$, from test problem Poloni. The level lines correspond in each case to α^*, 0.1, 0.2, 0.3 and 0.4.

One can model the Copula after having transformed each variables with the respective CDF, different modelling options are shown here…

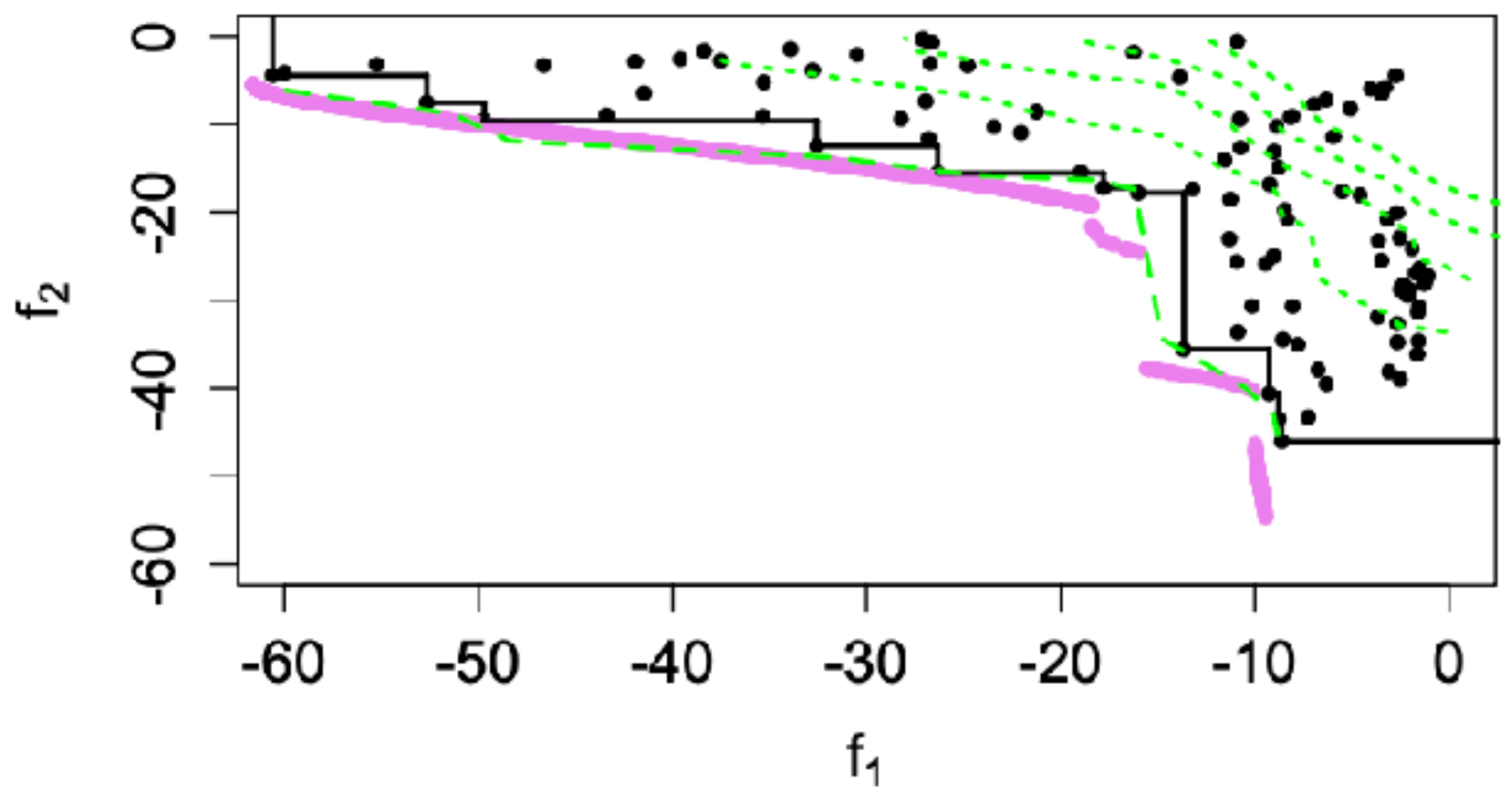


Figure 13: Estimated level line $\partial L_{\alpha^*}^F$ for the Poloni test problem (green dashed line), compared to the Pareto front approximation from the observation $\mathcal{P}_n$ (black line) and the true Pareto front $\mathcal{P}$ (violet solid line). Other level lines with levels 0.1, 0.2, 0.3 and 0.4 are also displayed with thinner lines.

… which allows to estimate the level set and the Pareto front

Applications & use cases

Applications

Hardware-aware NAS

Applications

Hardware-aware NAS

- Hardware-aware NAS looks at finding architecture with good latency/accuracy trade-offs…

Applications

Hardware-aware NAS

- Hardware-aware NAS looks at finding architecture with good latency/accuracy trade-offs…
- So that they can be deployed on device

Applications

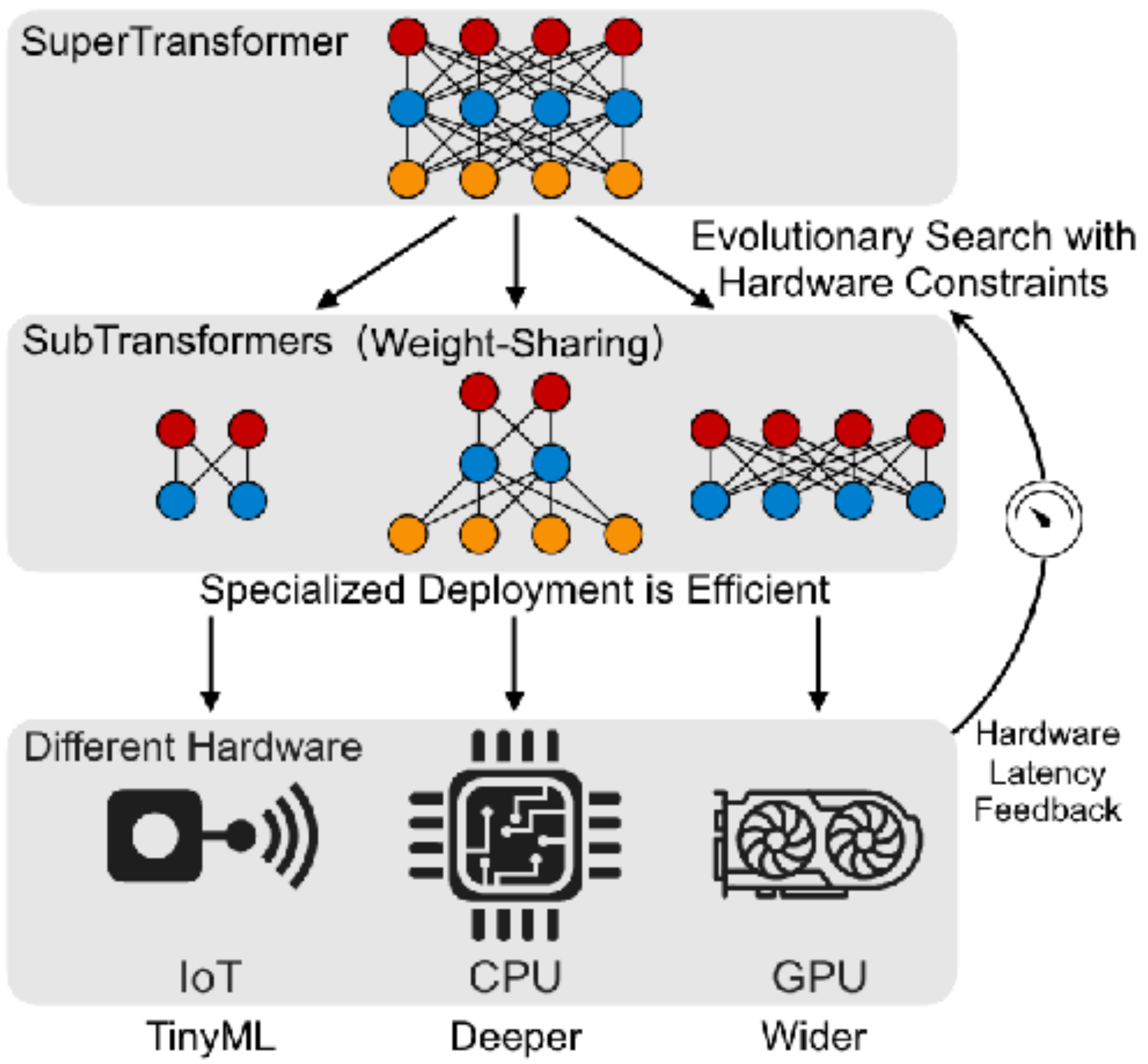
Hardware-aware NAS

- Hardware-aware NAS looks at finding architecture with good latency/accuracy trade-offs…
- So that they can be deployed on device
- Perform search on each device

Applications

Hardware-aware NAS

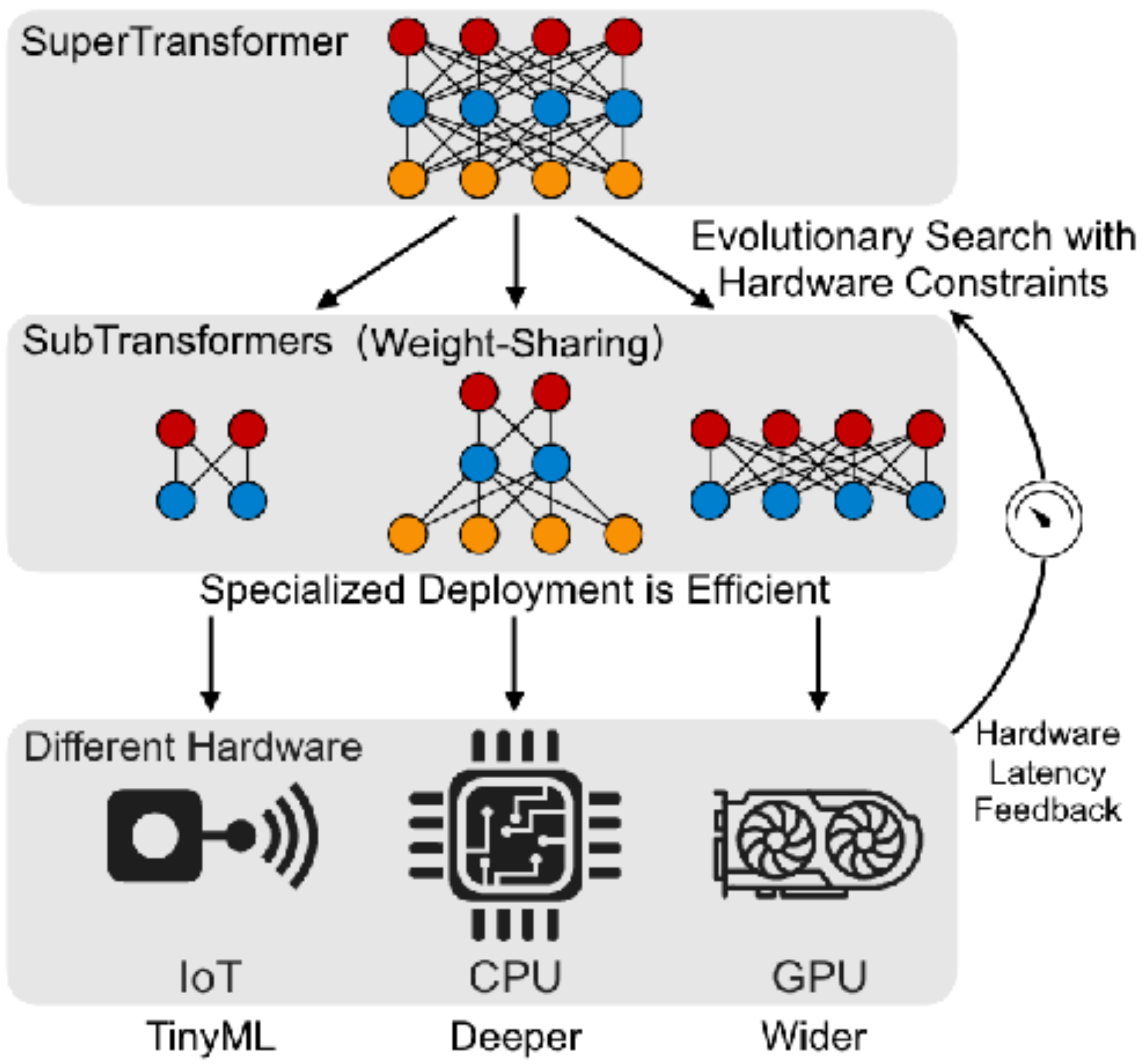
- Hardware-aware NAS looks at finding architecture with good latency/accuracy trade-offs…
- So that they can be deployed on device
- Perform search on each device

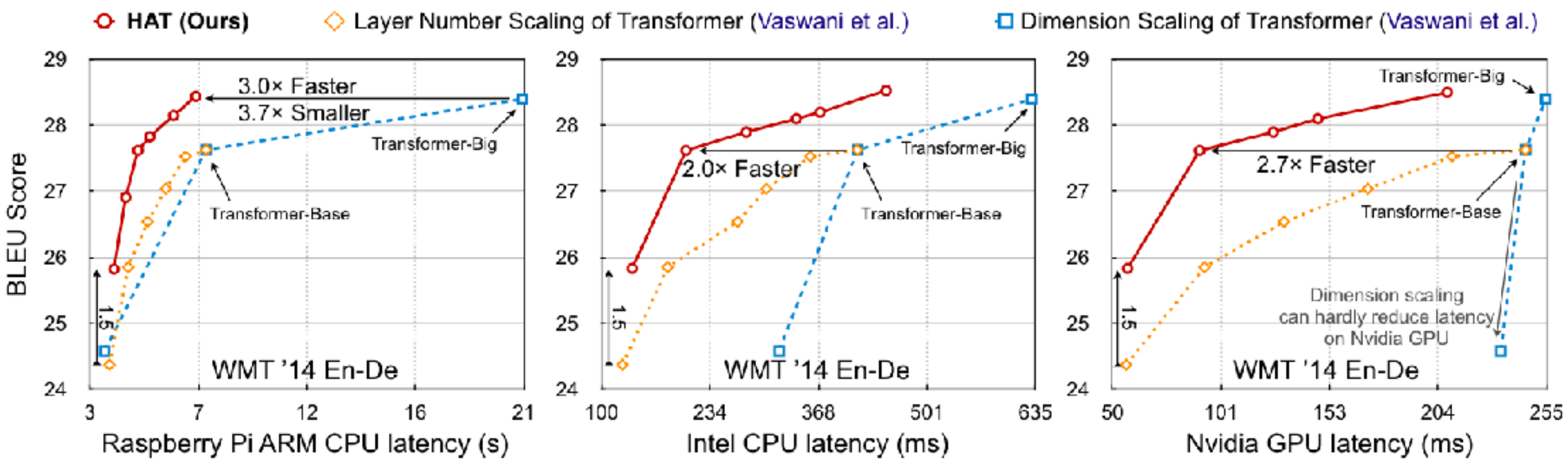


Applications

Hardware-aware NAS

- Hardware-aware NAS looks at finding architecture with good latency/accuracy trade-offs…
- So that they can be deployed on device
- Perform search on each device





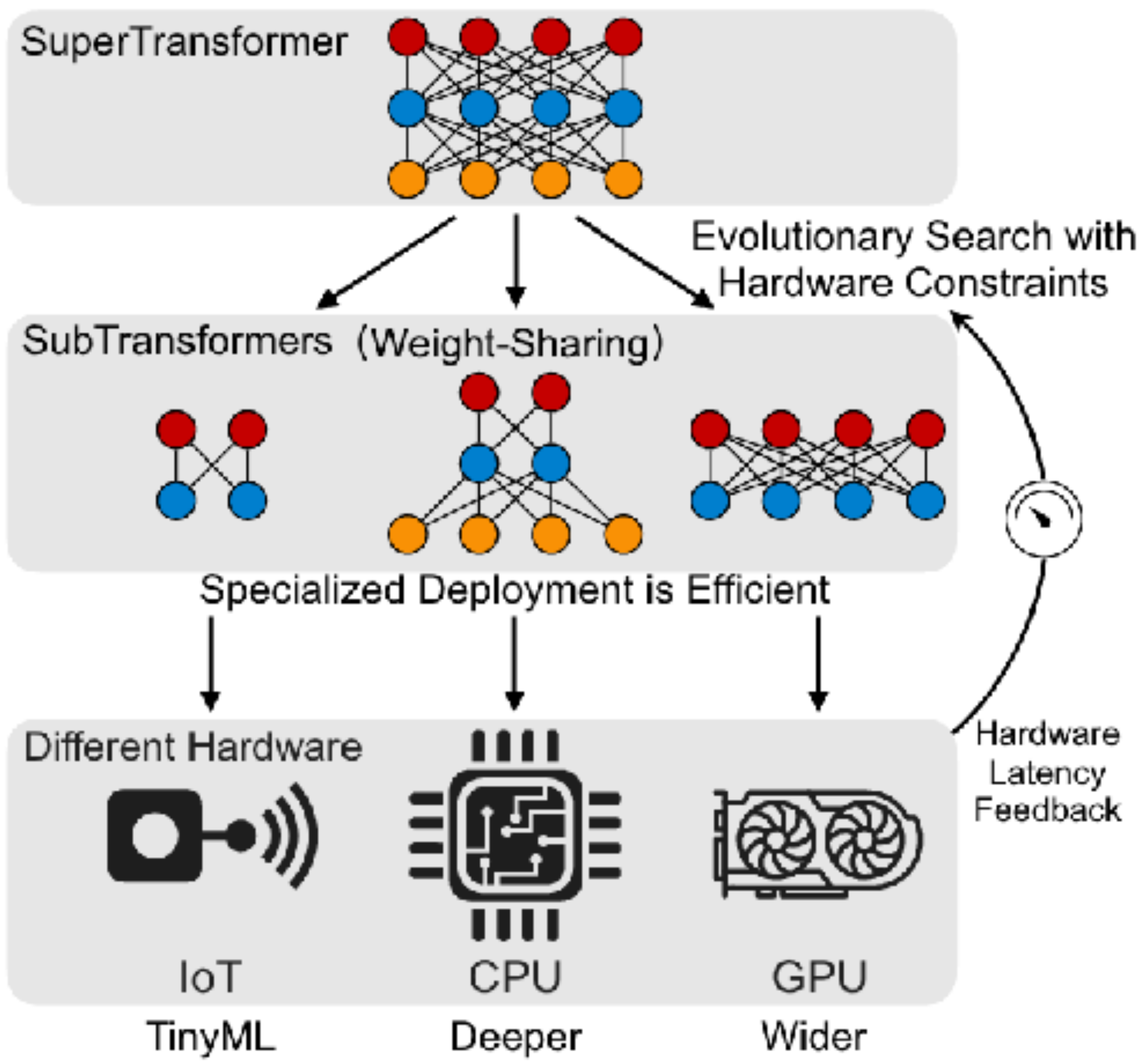
HAT: Hardware-Aware Transformers for Efficient Natural Language Processing [Wang 2020]

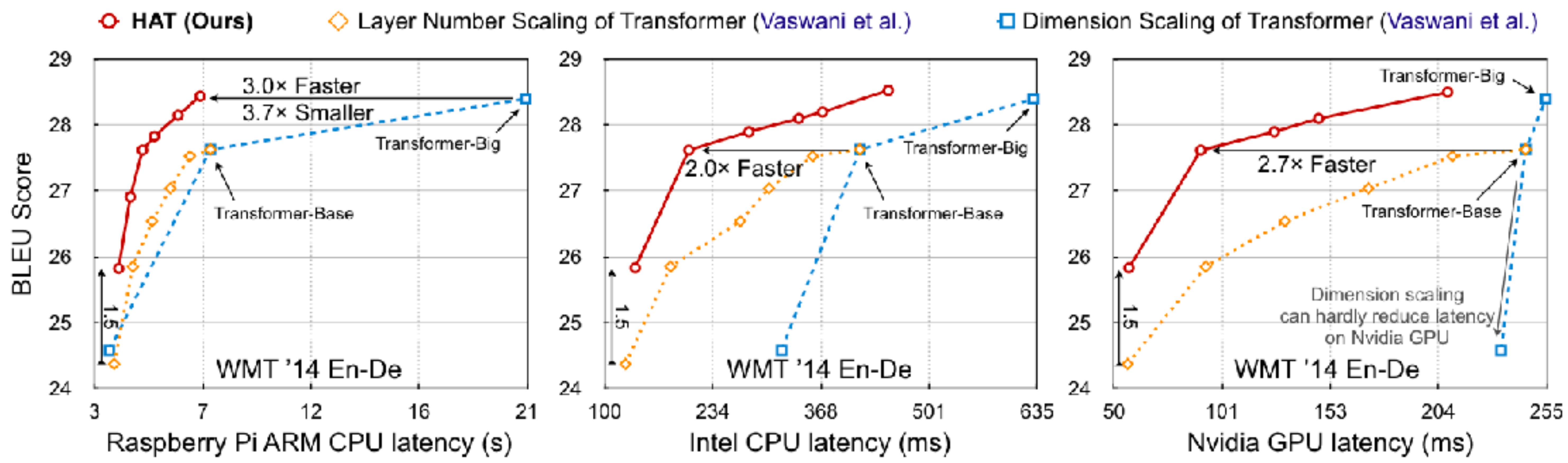
Applications

Hardware-aware NAS

- Hardware-aware NAS looks at finding architecture with good latency/accuracy trade-offs…
- So that they can be deployed on device
- Perform search on each device

🚀 Killer application of (multiobjective) NAS

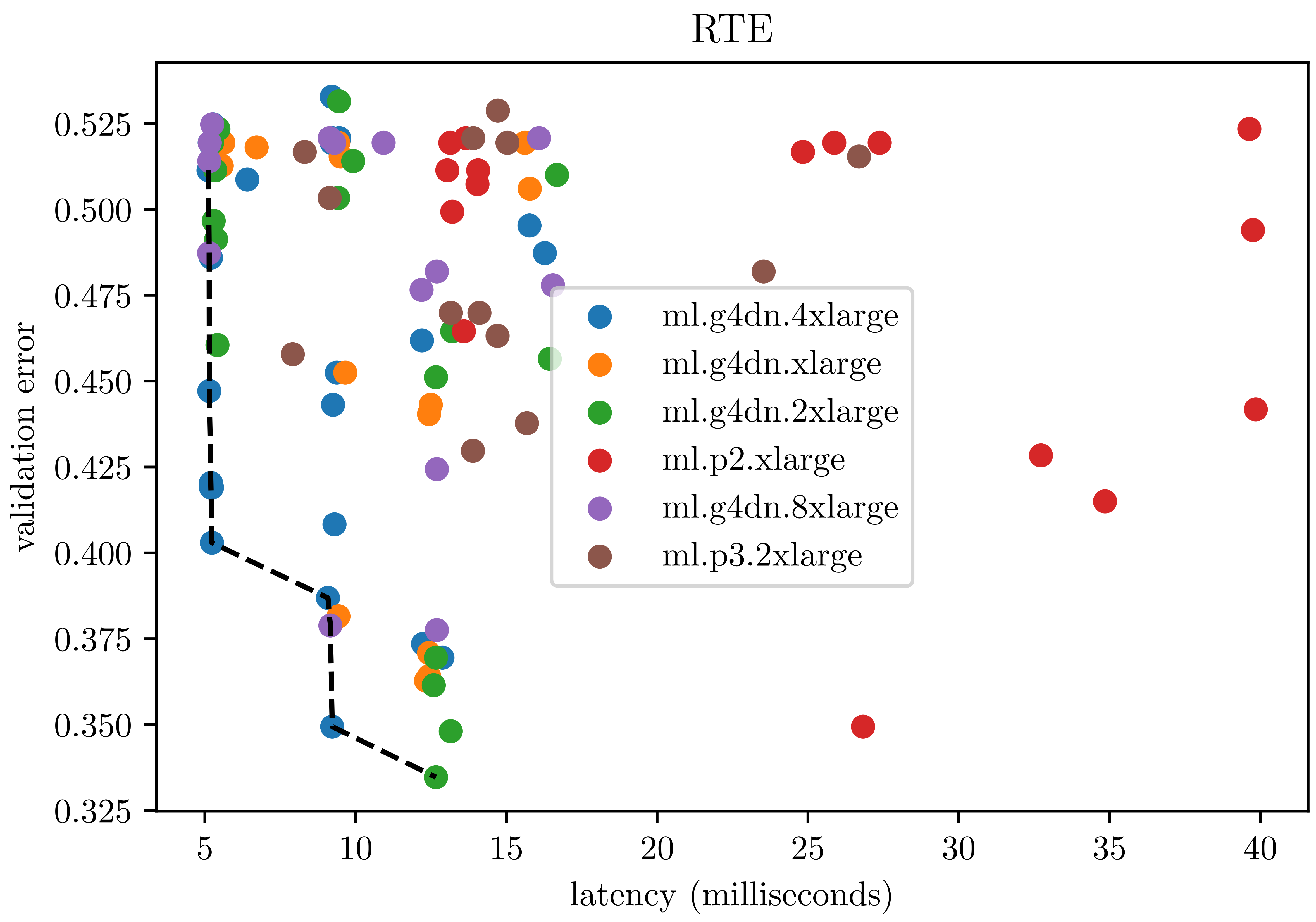




HAT: Hardware-Aware Transformers for Efficient Natural Language Processing [Wang 2020]

Applications

Tuning hyperparameter and hardware configurations

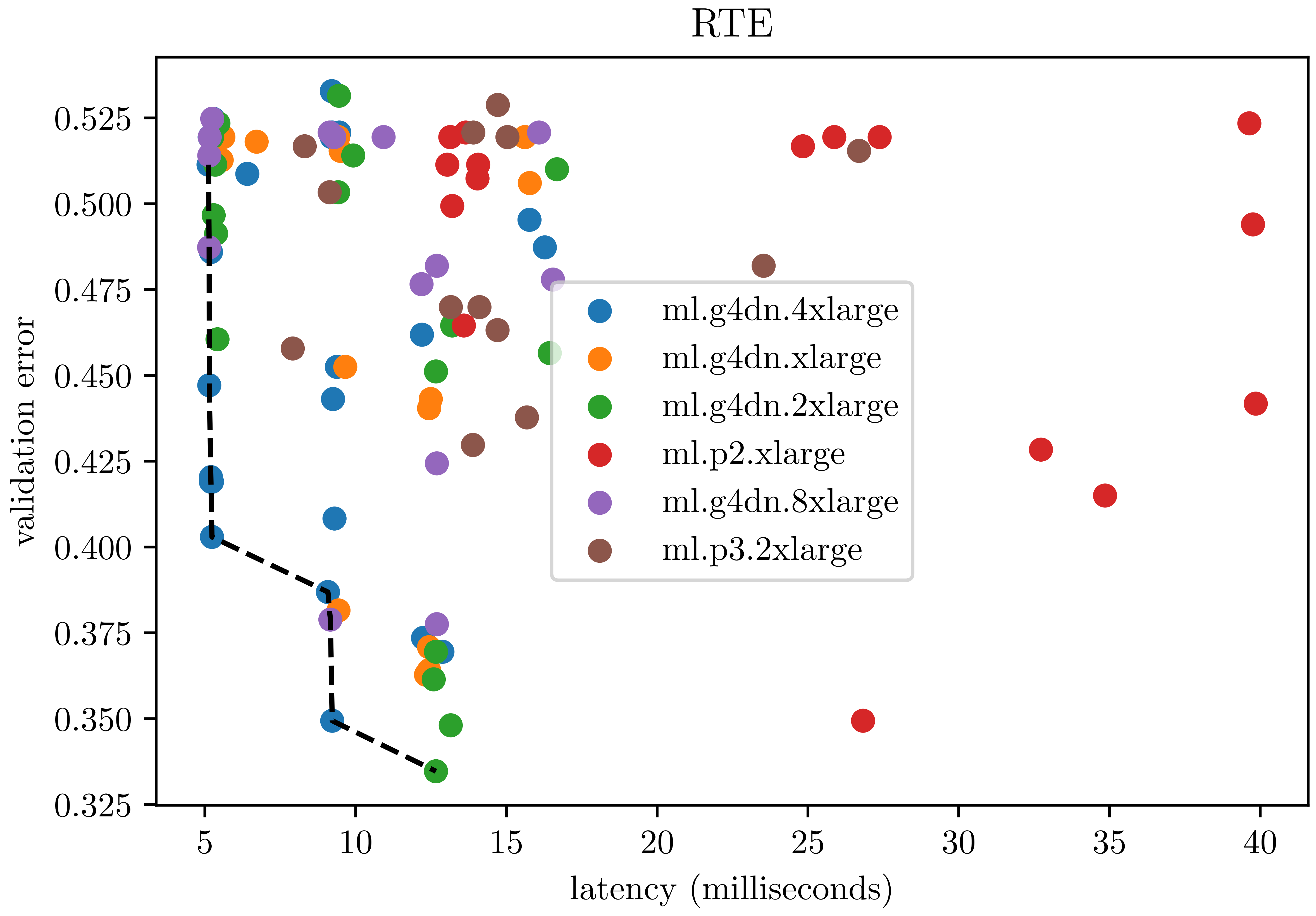


Tuning jointly machines and hyperparameters [Salinas et al 2022]

Applications

Tuning hyperparameter and hardware configurations

- Hyperparameter and machine types can be tuned at the same time!

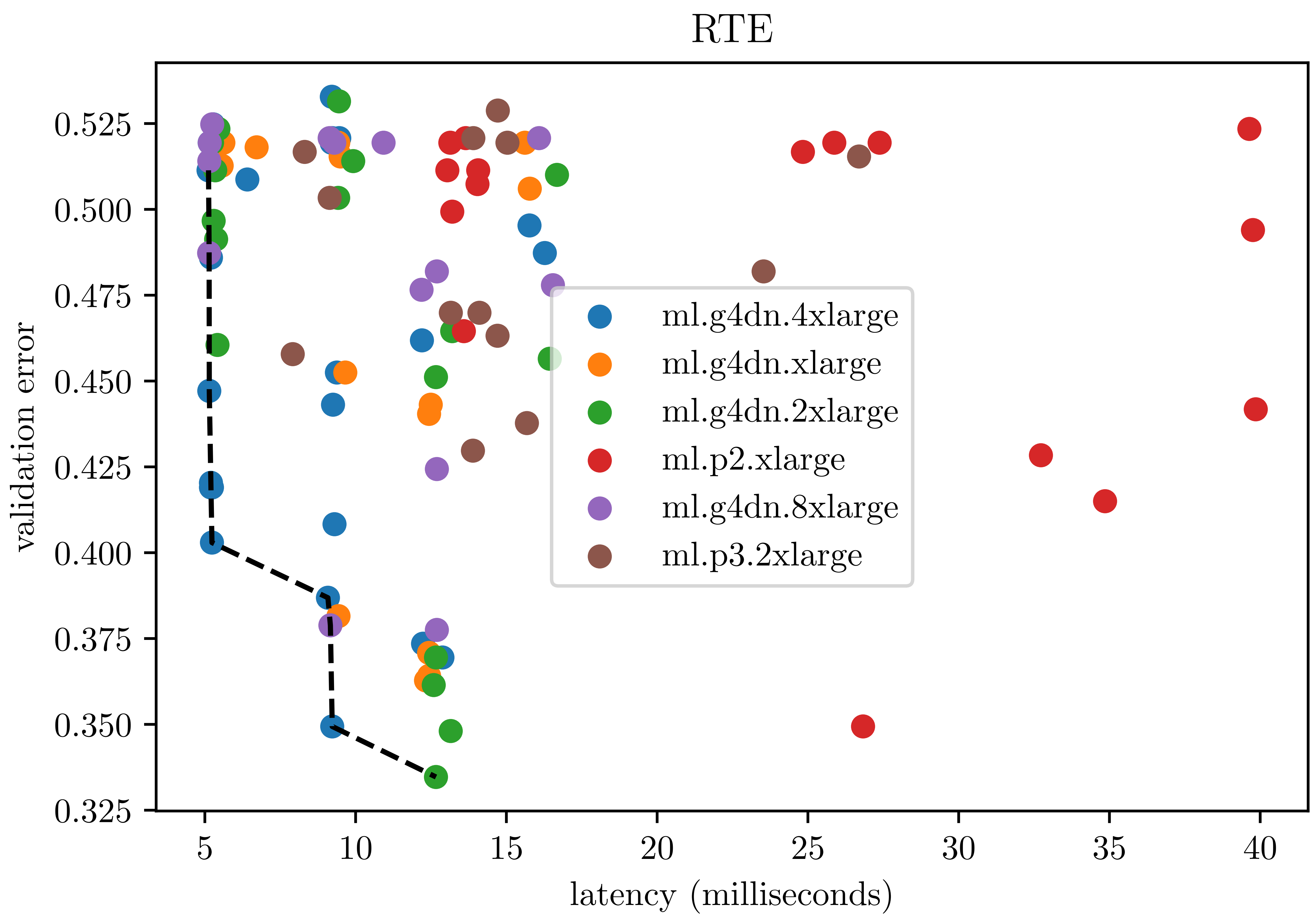


Tuning jointly machines and hyperparameters [Salinas et al 2022]

Applications

Tuning hyperparameter and hardware configurations

- Hyperparameter and machine types can be tuned at the same time!
- Add machine type (type/number of Cpu and Gpu => ~100 choices in AWS) to search space

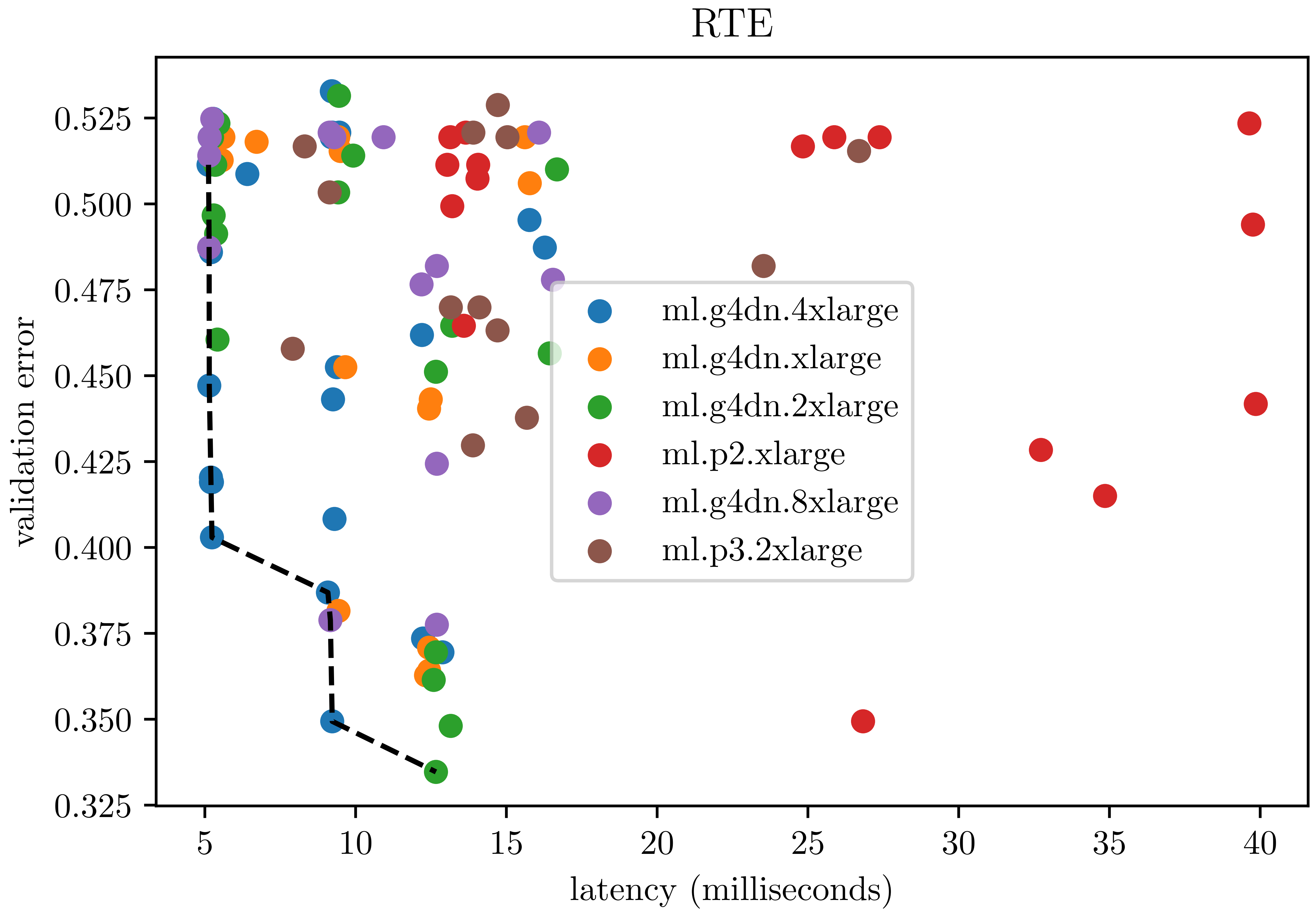


Tuning jointly machines and hyperparameters [Salinas et al 2022]

Applications

Tuning hyperparameter and hardware configurations

- Hyperparameter and machine types can be tuned at the same time!
- Add machine type (type/number of Cpu and Gpu => ~100 choices in AWS) to search space
- Machines have different speed and cost trade-offs

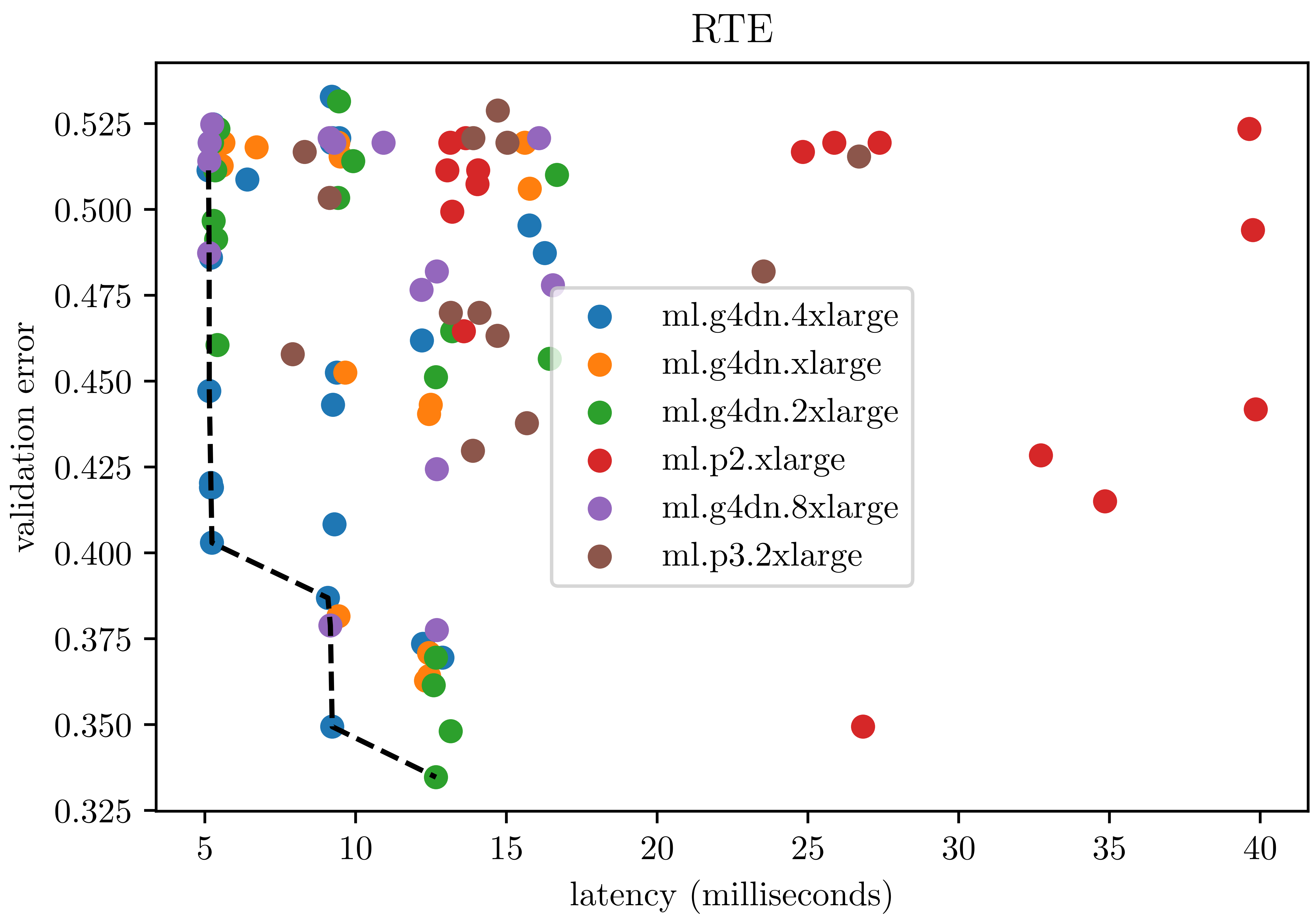


Tuning jointly machines and hyperparameters [Salinas et al 2022]

Applications

Tuning hyperparameter and hardware configurations

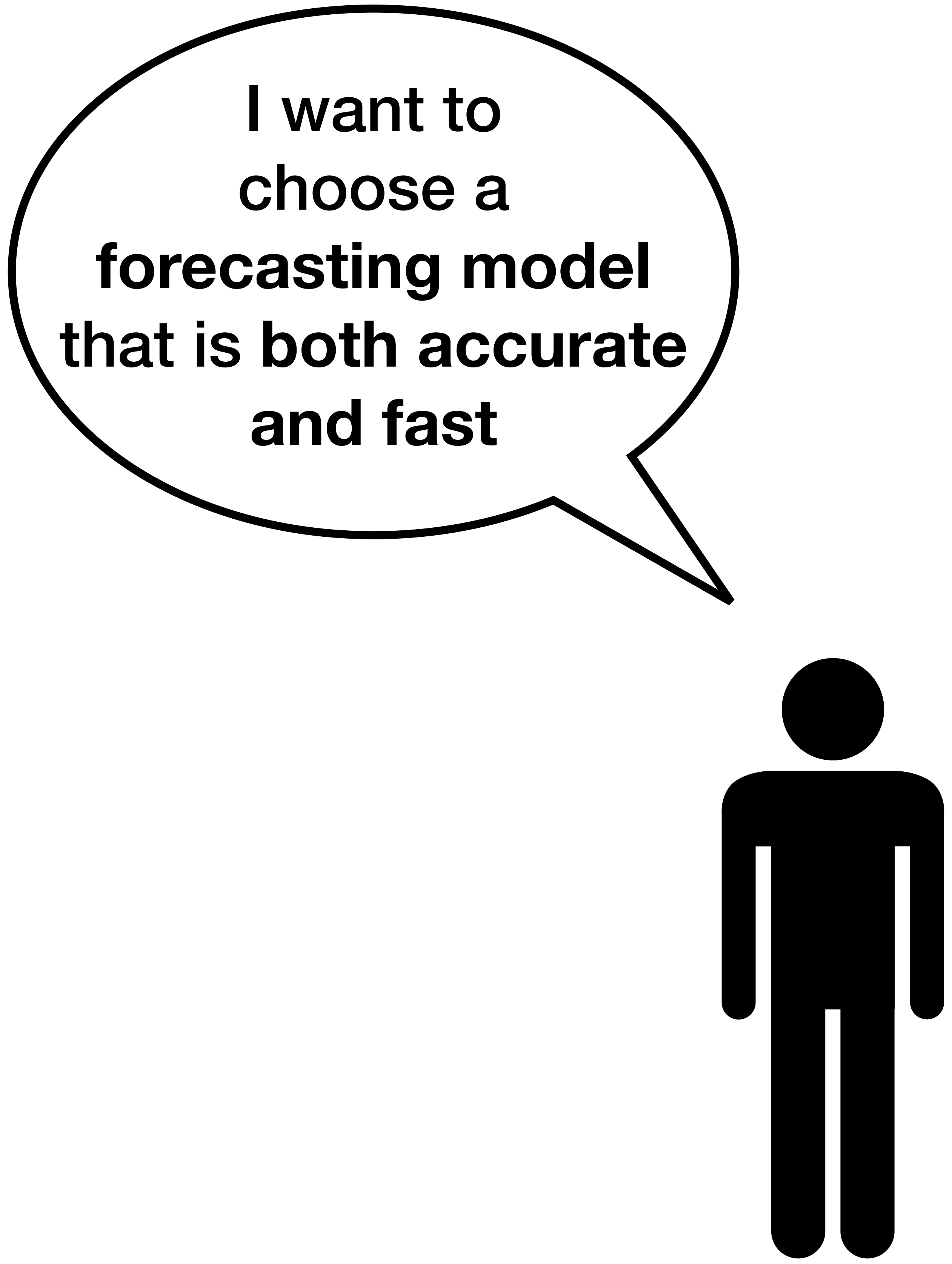
- Hyperparameter and machine types can be tuned at the same time!
- Add machine type (type/number of Cpu and Gpu => ~100 choices in AWS) to search space
- Machines have different speed and cost trade-offs
- Code example in Syne Tune [link]



Tuning jointly machines and hyperparameters [Salinas et al 2022]

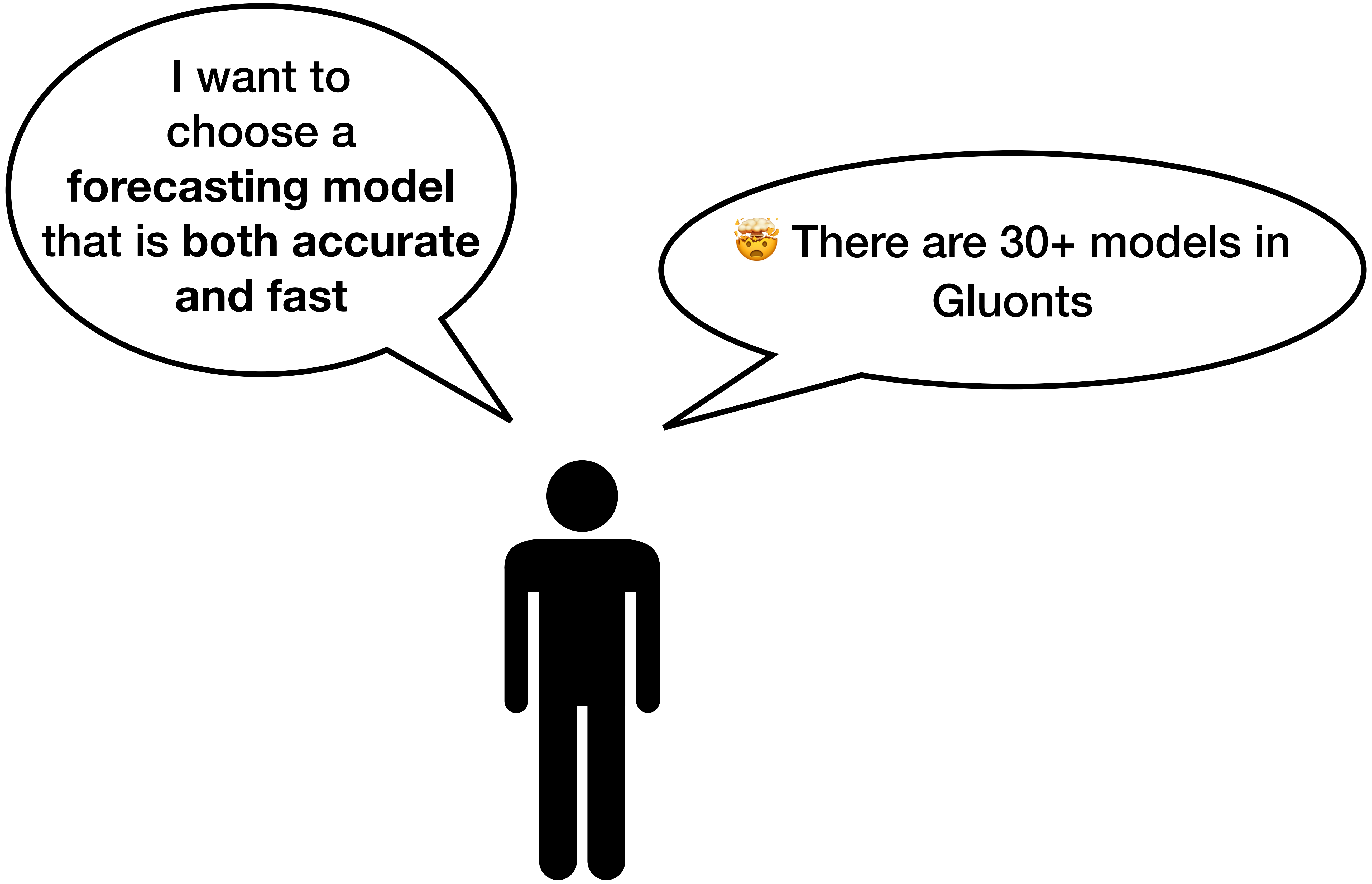
Applications

Multiobjective transfer learning

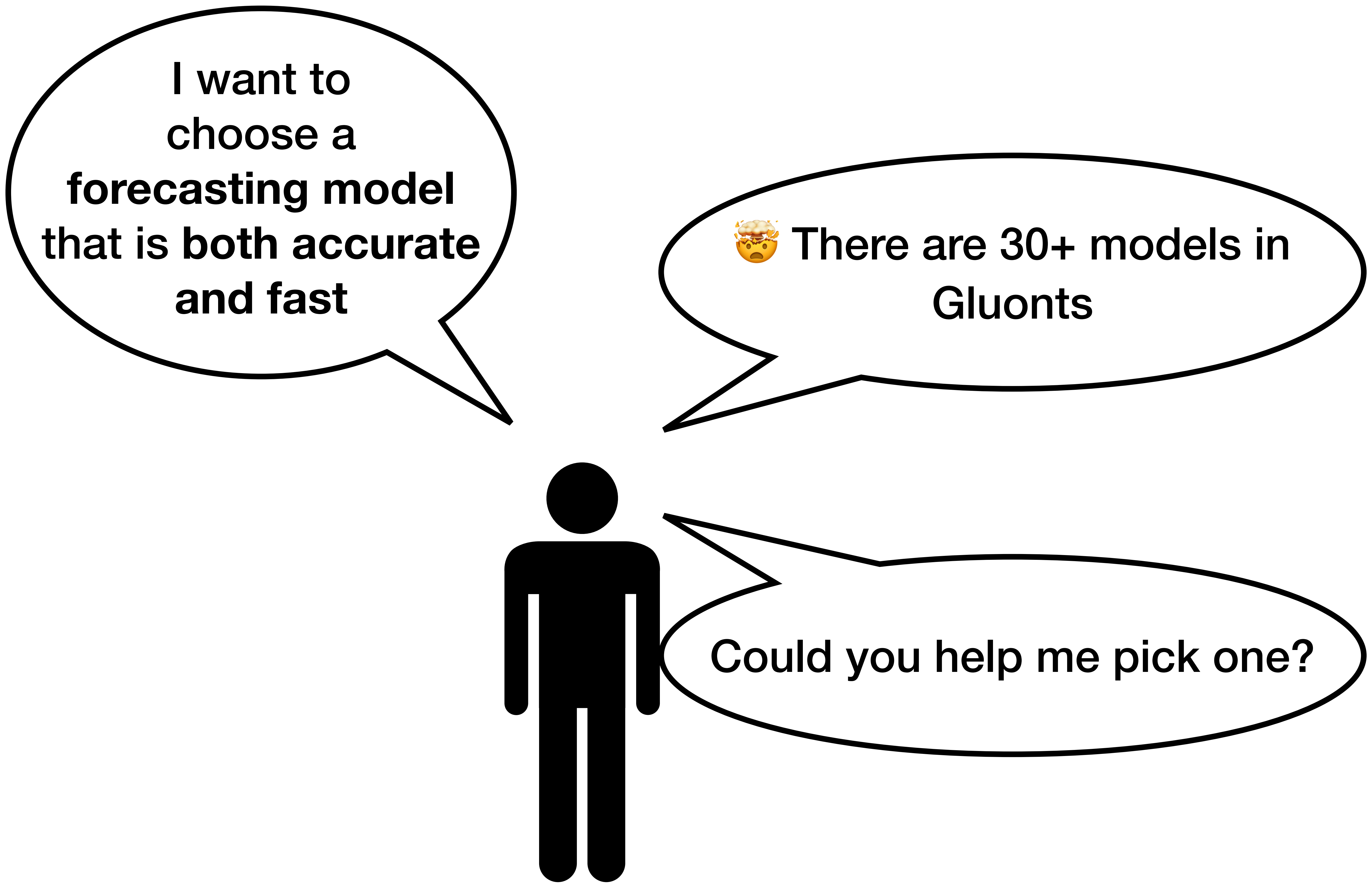


Applications

Multiobjective transfer learning



Applications
Multiobjective transfer learning



Multiobjective transfer learning

Multiobjective transfer learning

- $x \in \mathscr{X}$ space of models and hyperparameters

Multiobjective transfer learning

- $x \in \mathcal{X}$ space of models and hyperparameters
- $f(x) \in \mathbb{R}^d$ has d objectives (error, latency, fairness, …)·

Multiobjective transfer learning

- $x \in \mathcal{X}$ space of models and hyperparameters
- $f(x) \in \mathbb{R}^d$ has d objectives (error, latency, fairness, …).
- Evaluating all options in $\mathcal{X}$ is too expensive!

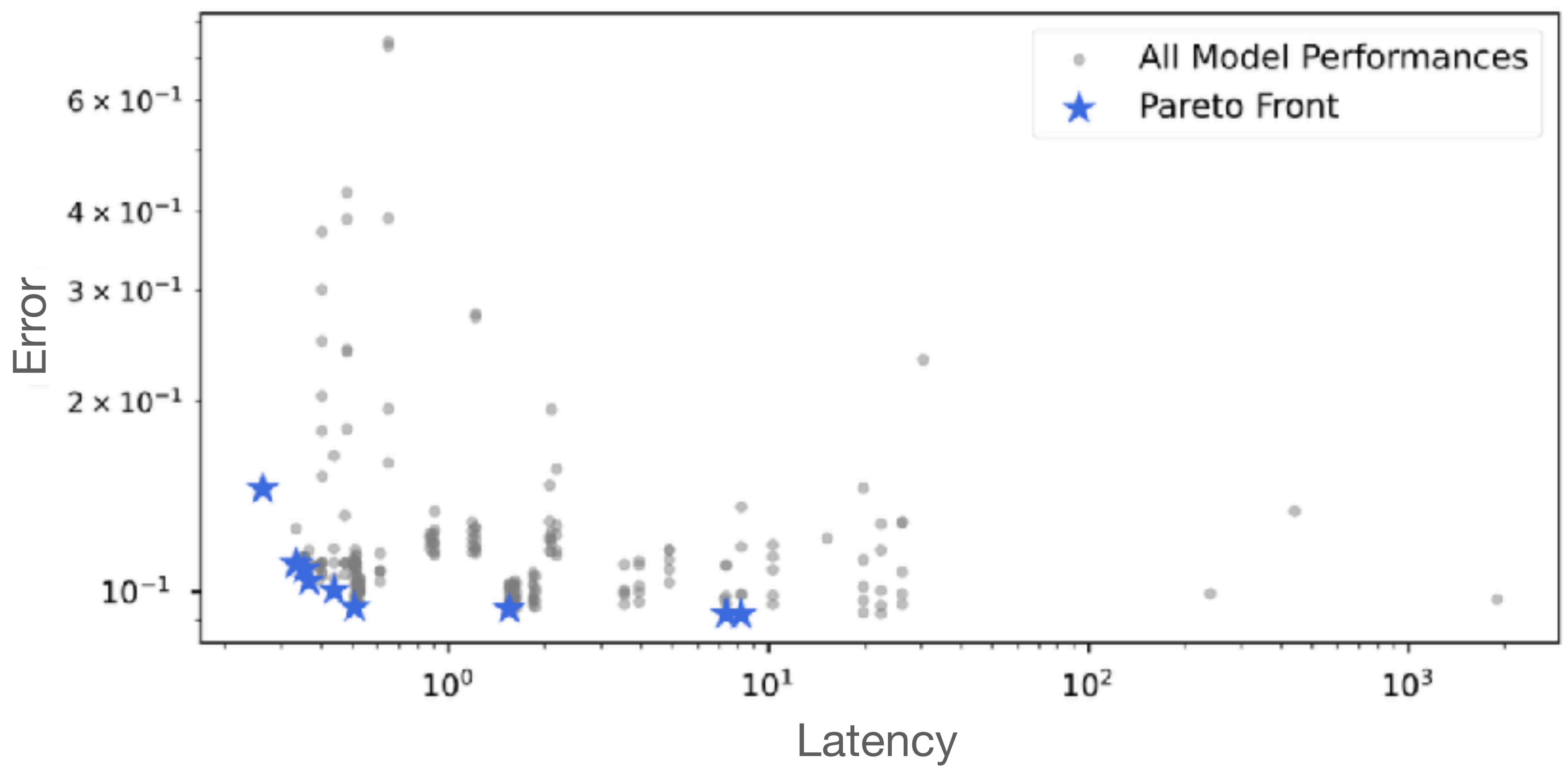
Multiobjective transfer learning

- $x \in \mathcal{X}$ space of models and hyperparameters
- $f(x) \in \mathbb{R}^d$ has d objectives (error, latency, fairness, …).
- Evaluating all options in $\mathcal{X}$ is too expensive!
- Given offline evaluations, can we approximate the Pareto Front in a zero-shot fashion?

Multiobjective transfer learning

- $x \in \mathscr{X}$ space of models and hyperparameters
- $f(x) \in \mathbb{R}^d$ has d objectives (error, latency, fairness, …).
- Evaluating all options in $\mathscr{X}$ is too expensive!
- Given offline evaluations, can we approximate the Pareto Front in a zero-shot fashion?

Error and latency for time series forecasting models



Multiobjective transfer learning

...given offline evaluations

Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…

Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…

task	model	learning-rate	#layers	error	latency
electricity	DeepAR	0.001	10	0.1	22.0
electricity	Transformer	0.001	10	0.08	2.5
electricity	Transformer	1.0	2	0.9	0.2
...					
traffic	DeepAR	0.1	2	0.9	0.2
traffic	Transformer	0.004	2.5	0.03	2.2

Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F

task	model	learning-rate	#layers	error	latency
electricity	DeepAR	0.001	10	0.1	22.0
electricity	Transformer	0.001	10	0.08	2.5
electricity	Transformer	1.0	2	0.9	0.2
...					
traffic	DeepAR	0.1	2	0.9	0.2
traffic	Transformer	0.004	2.5	0.03	2.2

Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F

task	model	learning-rate	#layers	z_{error}	z_{latency}
electricity	DeepAR	0.001	10	0.2	3.4
electricity	Transformer	0.001	10	-1.2	0.6
electricity	Transformer	1.0	2	1.5	-2.3
...					
traffic	DeepAR	0.1	2	2.2	-2.4
traffic	Transformer	0.004	2.5	-1.4	0.5

Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F
- 2. Fit d independent predictive models for each objectives $[z_1(x), \dots, z_d(x)]$

task	model	learning-rate	#layers	z_{error}	z_{latency}
electricity	DeepAR	0.001	10	0.2	3.4
electricity	Transformer	0.001	10	-1.2	0.6
electricity	Transformer	1.0	2	1.5	-2.3
...					
traffic	DeepAR	0.1	2	2.2	-2.4
traffic	Transformer	0.004	2.5	-1.4	0.5

Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F
- 2. Fit d independent predictive models for each objectives $[z_1(x), \ldots, z_d(x)]$
- 3. Predict objectives on the new task on all untrained models

task	model	learning-rate	#layers	z_{error}	z_{latency}
electricity	DeepAR	0.001	10	0.2	3.4
electricity	Transformer	0.001	10	-1.2	0.6
electricity	Transformer	1.0	2	1.5	-2.3
...					
traffic	DeepAR	0.1	2	2.2	-2.4
traffic	Transformer	0.004	2.5	-1.4	0.5

Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F
- 2. Fit d independent predictive models for each objectives $[z_1(x), \ldots, z_d(x)]$
- 3. Predict objectives on the new task on all untrained models

task	model	learning-rate	#layers	z_{error}	z_{latency}
solar	DeepAR	0.001	10	?	?
solar	Transformer	0.001	10	?	?
solar	Transformer	1.0	2	?	?
...					
solar	DeepAR	0.1	2	?	?
solar	Transformer	0.004	2.5	?	?

Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F
- 2. Fit d independent predictive models for each objectives $[z_1(x), \ldots, z_d(x)]$
- 3. Predict objectives on the new task on all untrained models

task	model	learning-rate	#layers	z_{error}	$z_{latency}$
solar	DeepAR	0.001	10	?	?
solar	Transformer	0.001	10	?	?
solar	Transformer	1.0	2	?	?
...					
solar	DeepAR	0.1	2	?	?
solar	Transformer	0.004	2.5	?	?

Predict objectives for each configurations

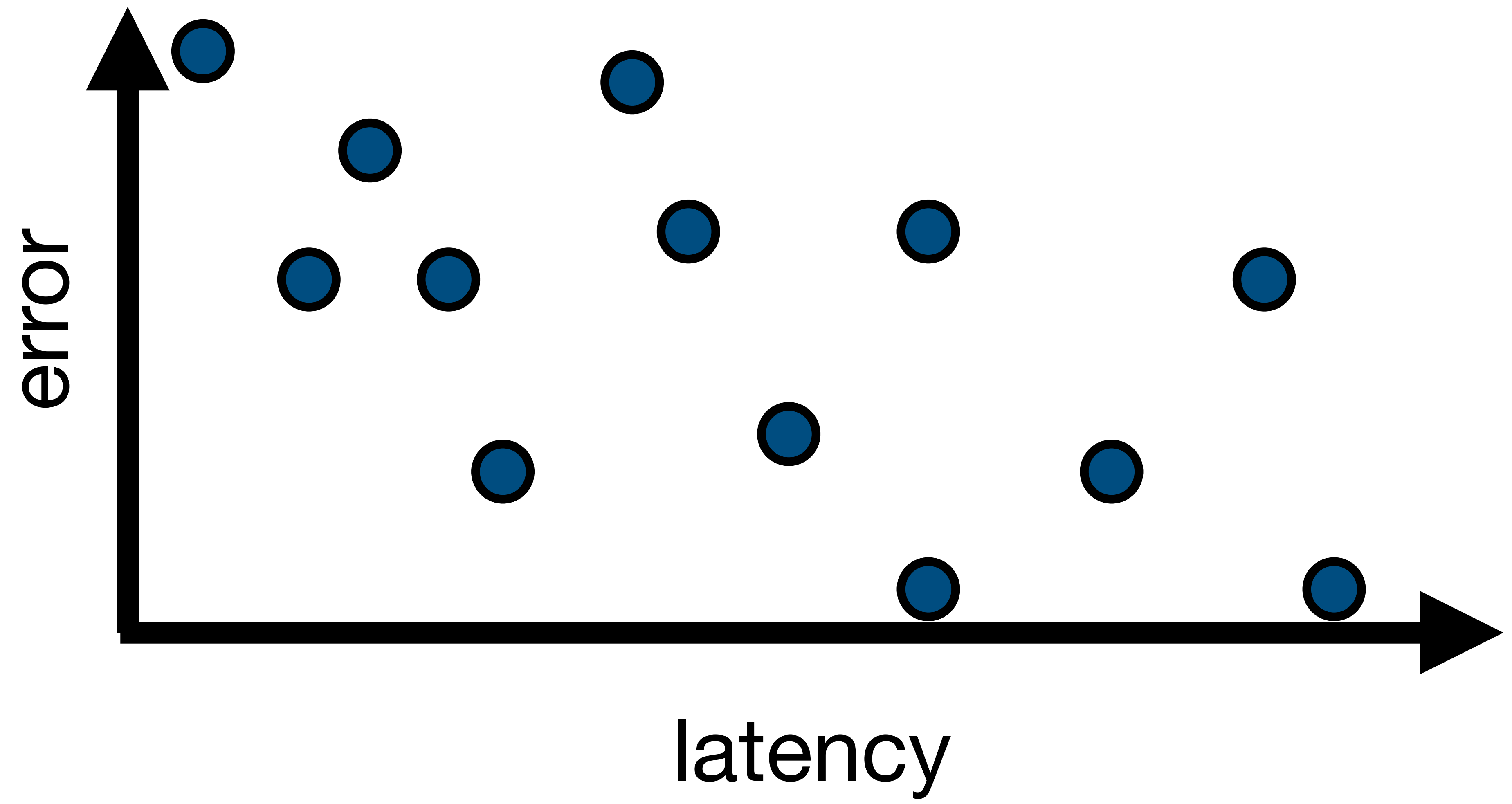
Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F
- 2. Fit d independent predictive models for each objectives $[z_1(x), \ldots, z_d(x)]$
- 3. Predict objectives on the new task on all untrained models

task	model	learning-rate	#layers	z_{error}	z_{latency}
solar	DeepAR	0.001	10	0.2	3.4
solar	Transformer	0.001	10	-1.2	0.6
solar	Transformer	1.0	2	1.5	-2.3
...					
solar	DeepAR	0.1	2	2.2	-2.4
solar	Transformer	0.004	2.5	-1.4	0.5

Predict objectives for each configurations →



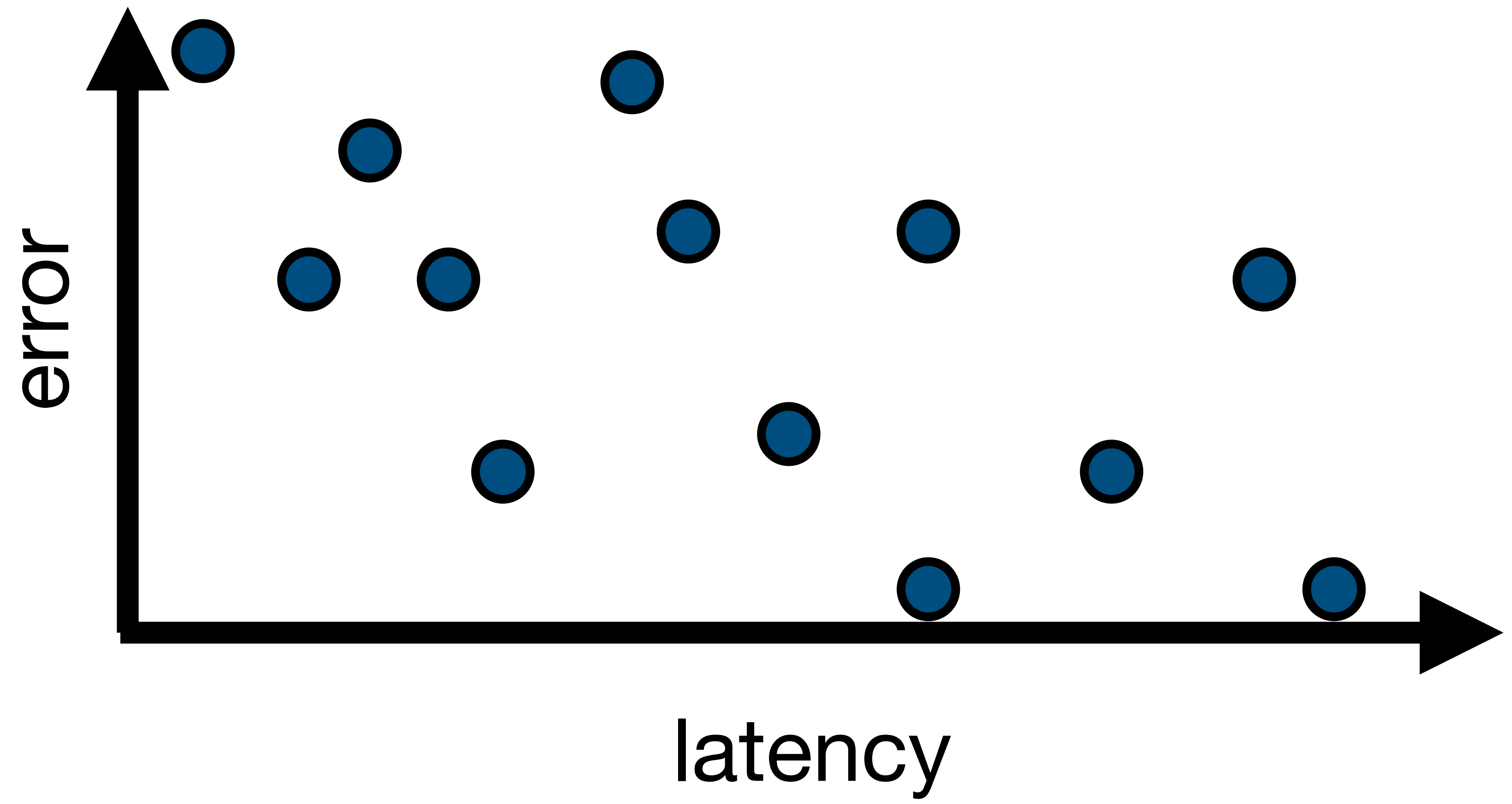
Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F
- 2. Fit d independent predictive models for each objectives $[z_1(x), ..., z_d(x)]$
- 3. Predict objectives on the new task on all untrained models
- 4. Return configuration on the Pareto front of predictions

task	model	learning-rate	#layers	z_{error}	$z_{latency}$
solar	DeepAR	0.001	10	0.2	3.4
solar	Transformer	0.001	10	-1.2	0.6
solar	Transformer	1.0	2	1.5	-2.3
...					
solar	DeepAR	0.1	2	2.2	-2.4
solar	Transformer	0.004	2.5	-1.4	0.5

Predict objectives for each configurations →



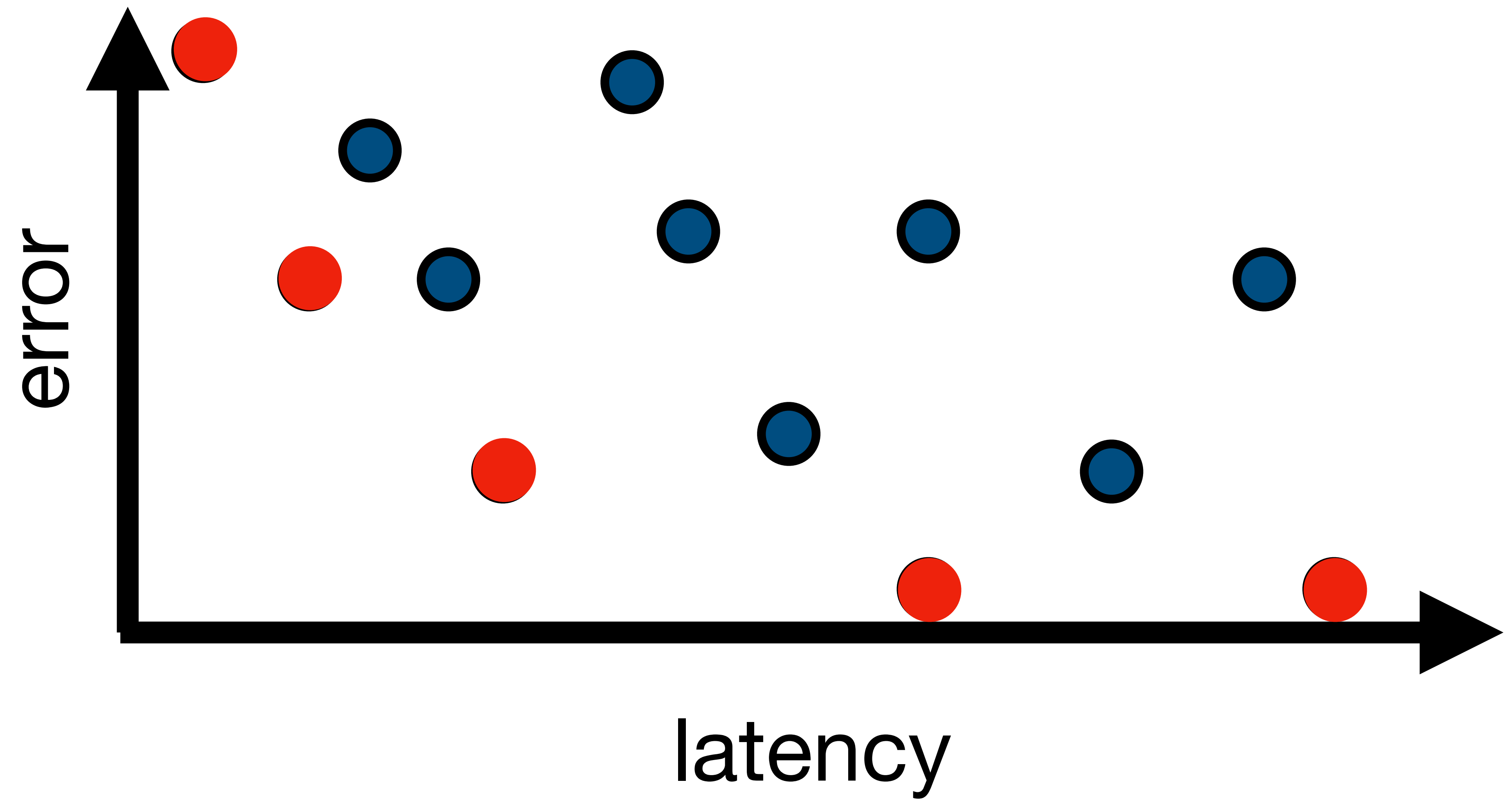
Multiobjective transfer learning

...given offline evaluations

- Assume we have access to offline evaluations…
- 1. Normalize each objective and task with $\psi = \Phi^{-1} \circ$ F
- 2. Fit d independent predictive models for each objectives $[z_1(x), \ldots, z_d(x)]$
- 3. Predict objectives on the new task on all untrained models
- 4. Return configuration on the Pareto front of predictions

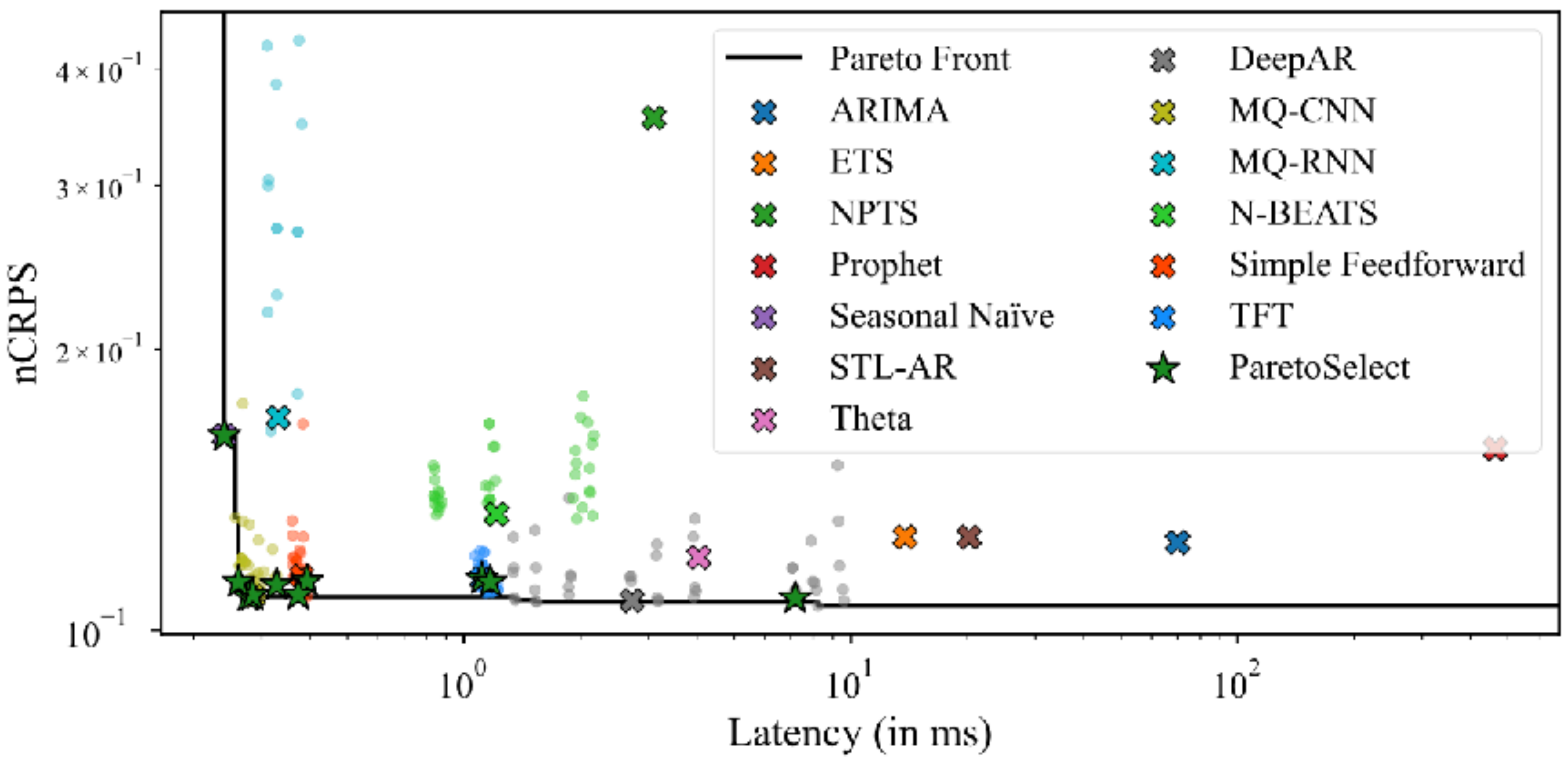
task	model	learning-rate	#layers	z_{error}	$z_{latency}$
solar	DeepAR	0.001	10	0.2	3.4
solar	Transformer	0.001	10	-1.2	0.6
solar	Transformer	1.0	2	1.5	-2.3
...					
solar	DeepAR	0.1	2	2.2	-2.4
solar	Transformer	0.004	2.5	-1.4	0.5

Predict objectives for each configurations →

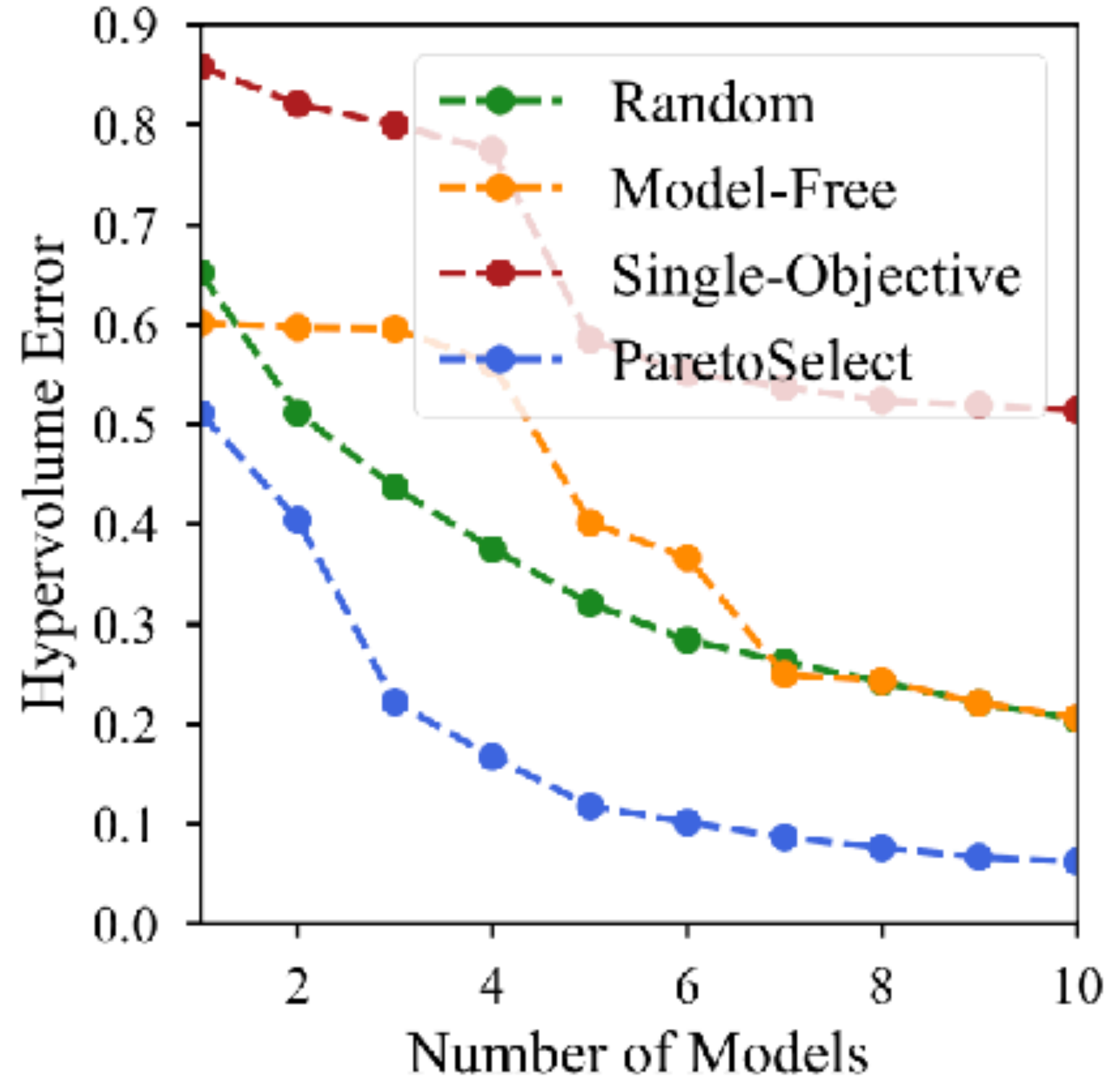


Multiobjective transfer learning

Zeroshot prediction of Pareto front through transfer learning



Example of one zero-shot selection in a fixed task

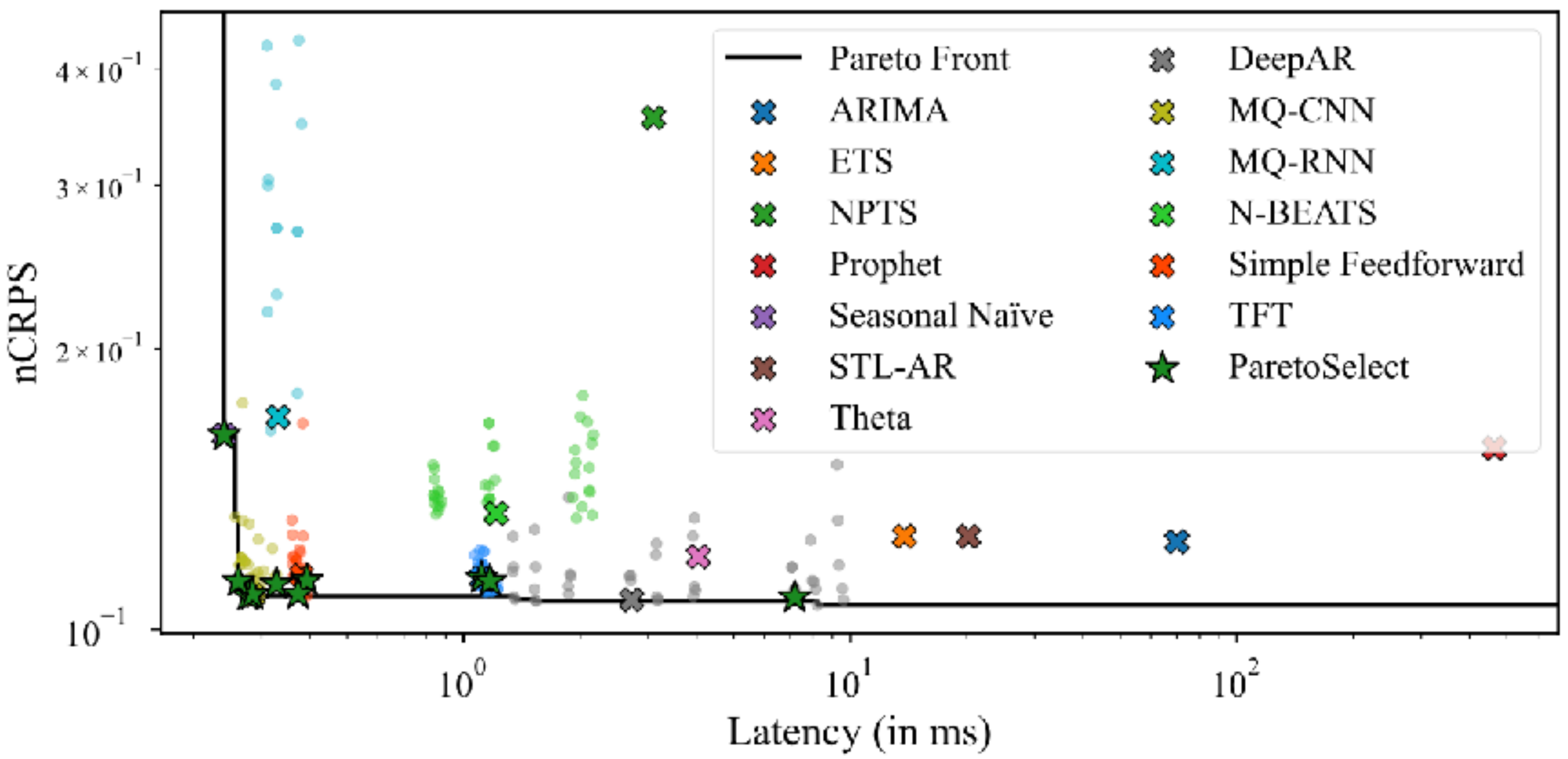


Average performance on all tasks

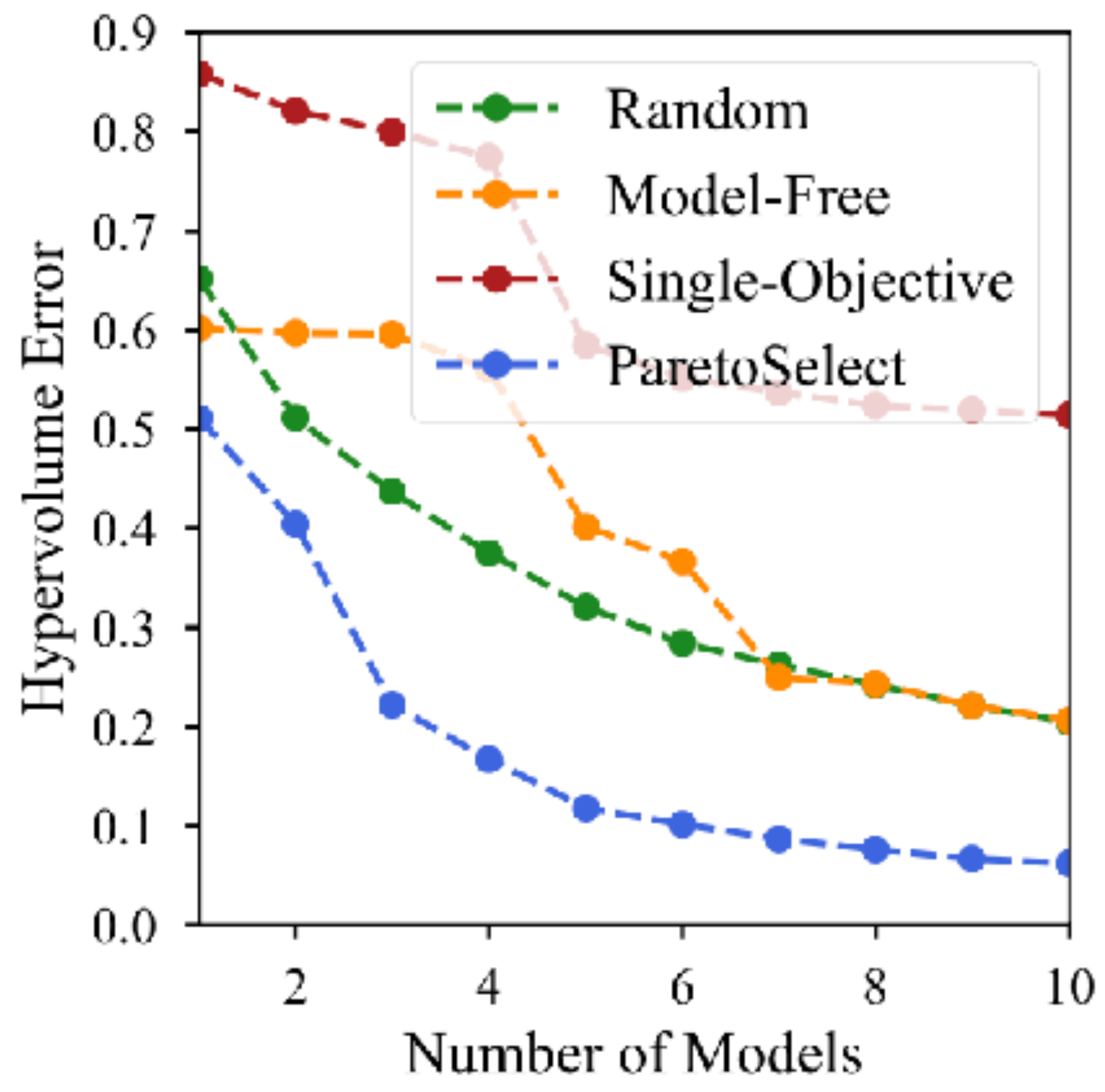
Multiobjective transfer learning

Zeroshot prediction of Pareto front through transfer learning

Reasonable approximation of Pareto front in zero shot fashion



Example of one zero-shot selection in a fixed task

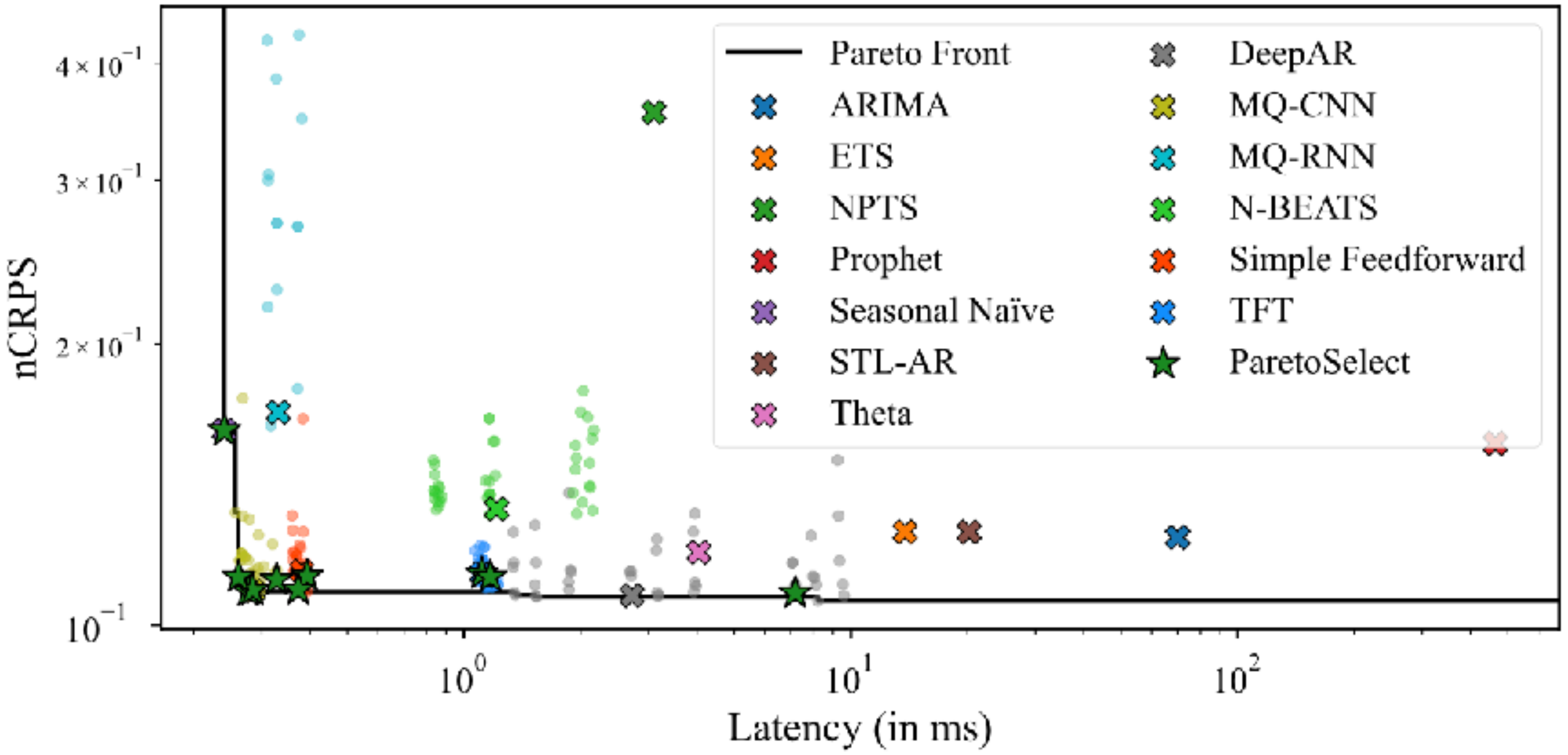


Average performance on all tasks

Multiobjective transfer learning

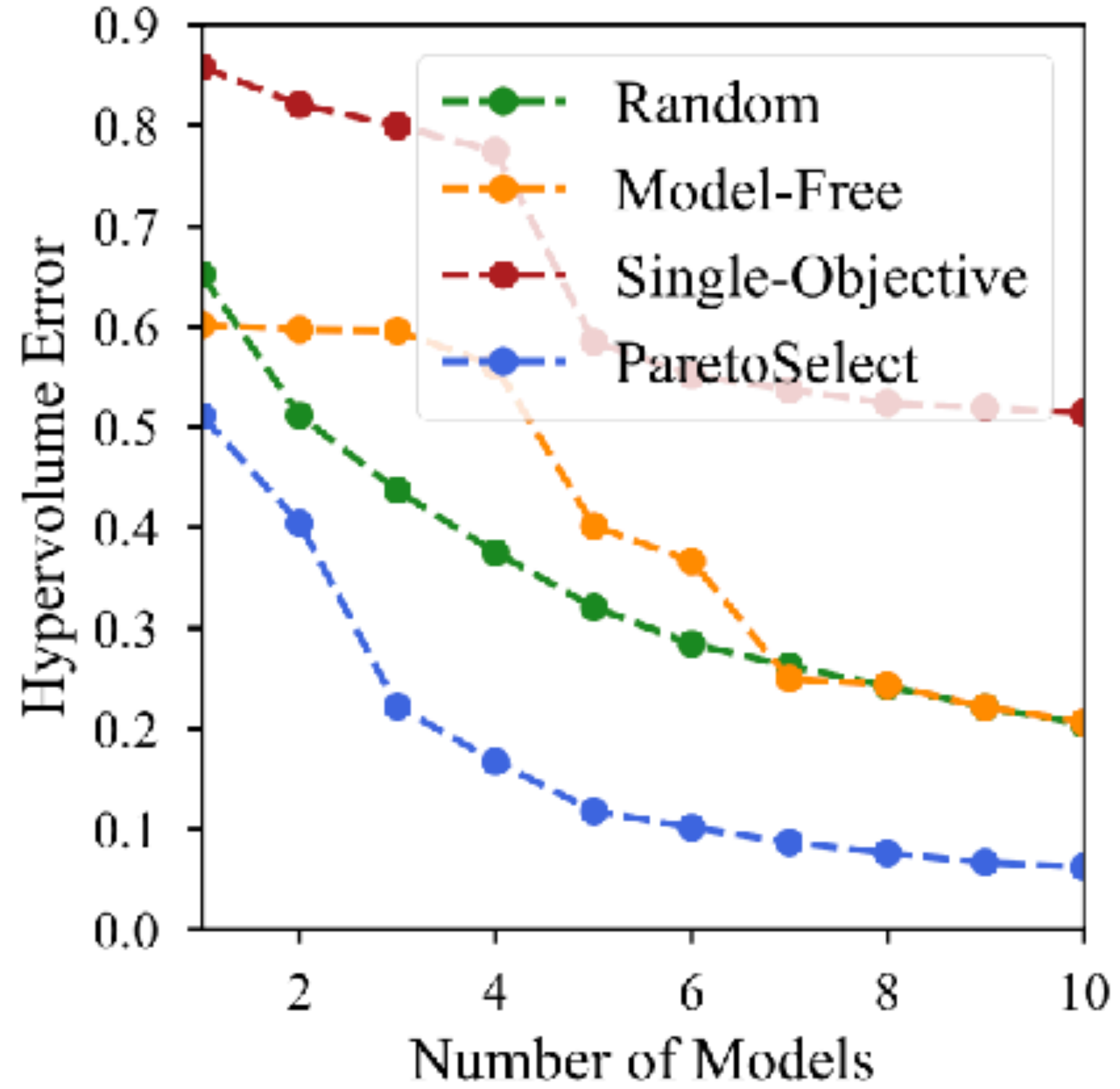
Zeroshot prediction of Pareto front through transfer learning

Reasonable approximation of Pareto front in zero shot fashion



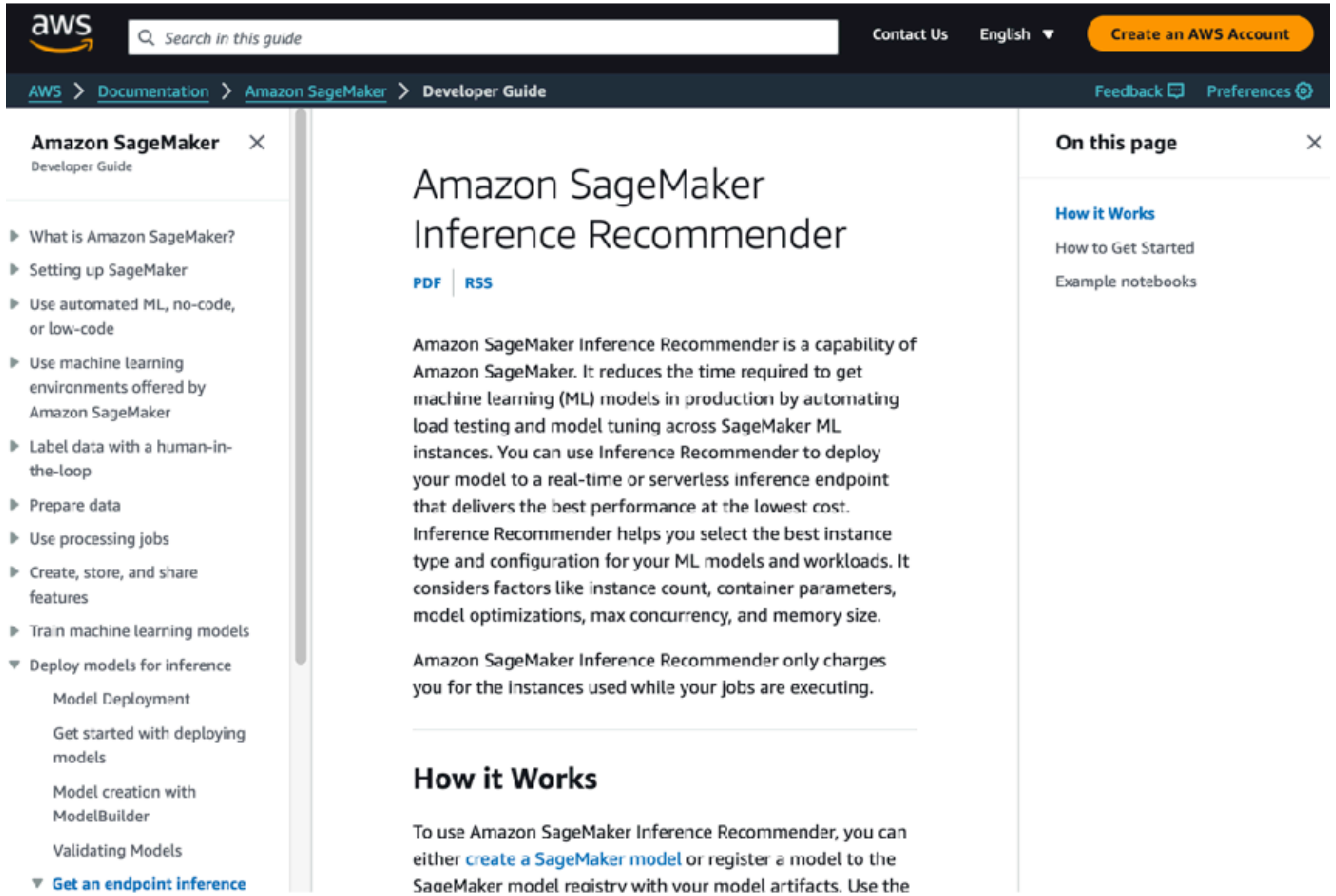
Example of one zero-shot selection in a fixed task

Much better hyper volume than baselines



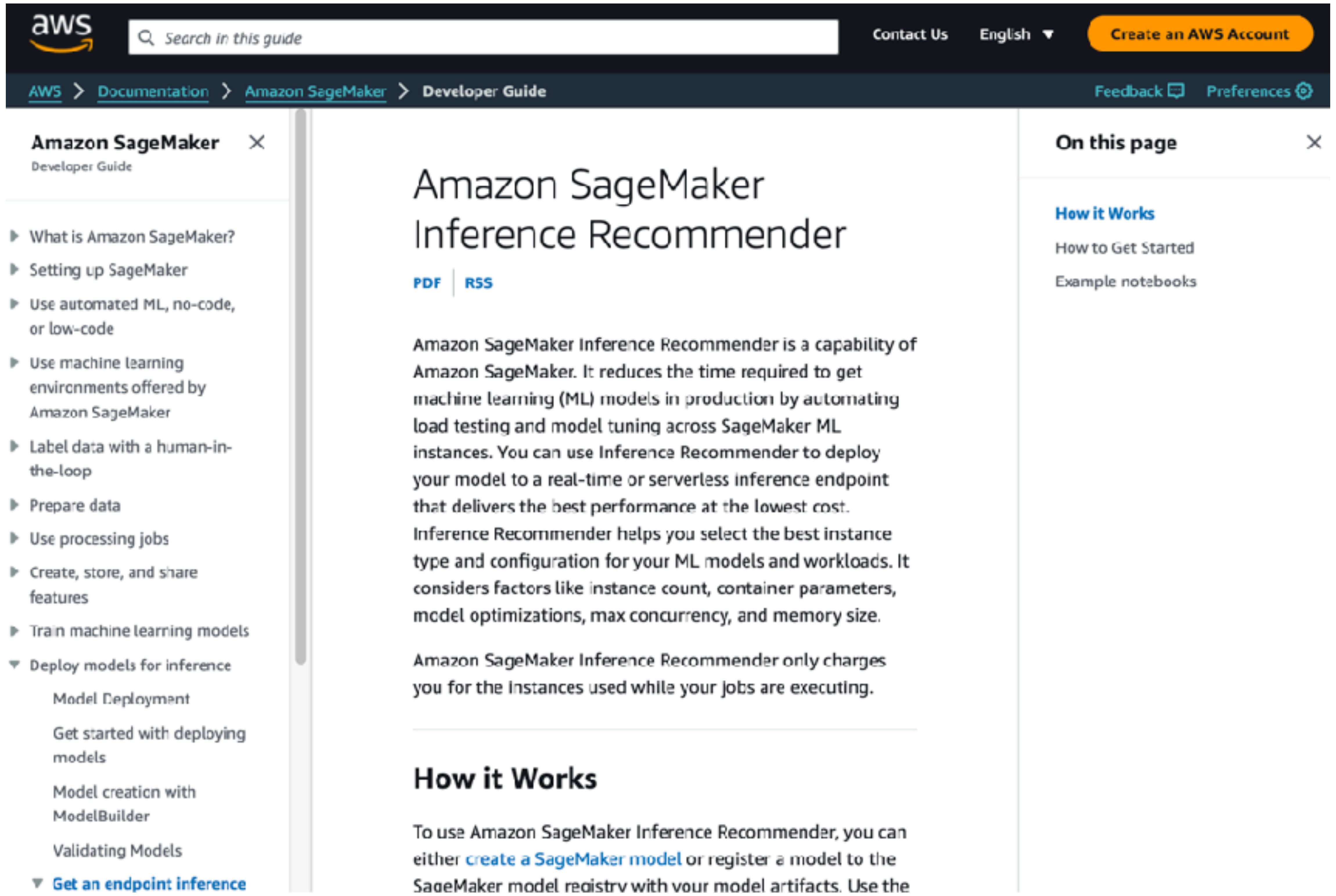
Average performance on all tasks

Applications: Instance Recommendation



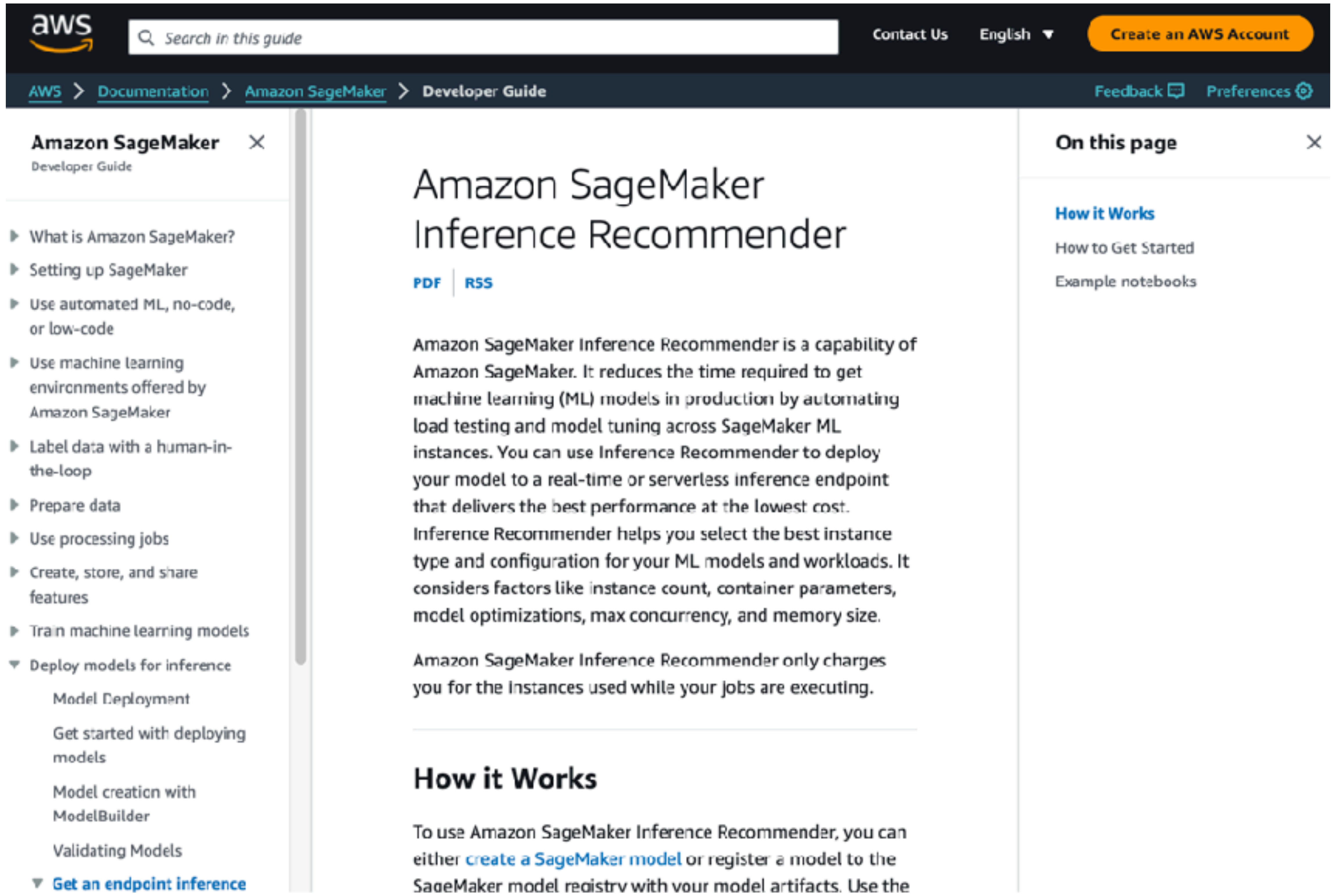
Applications: Instance Recommendation

- Instance recommendation for model deployment



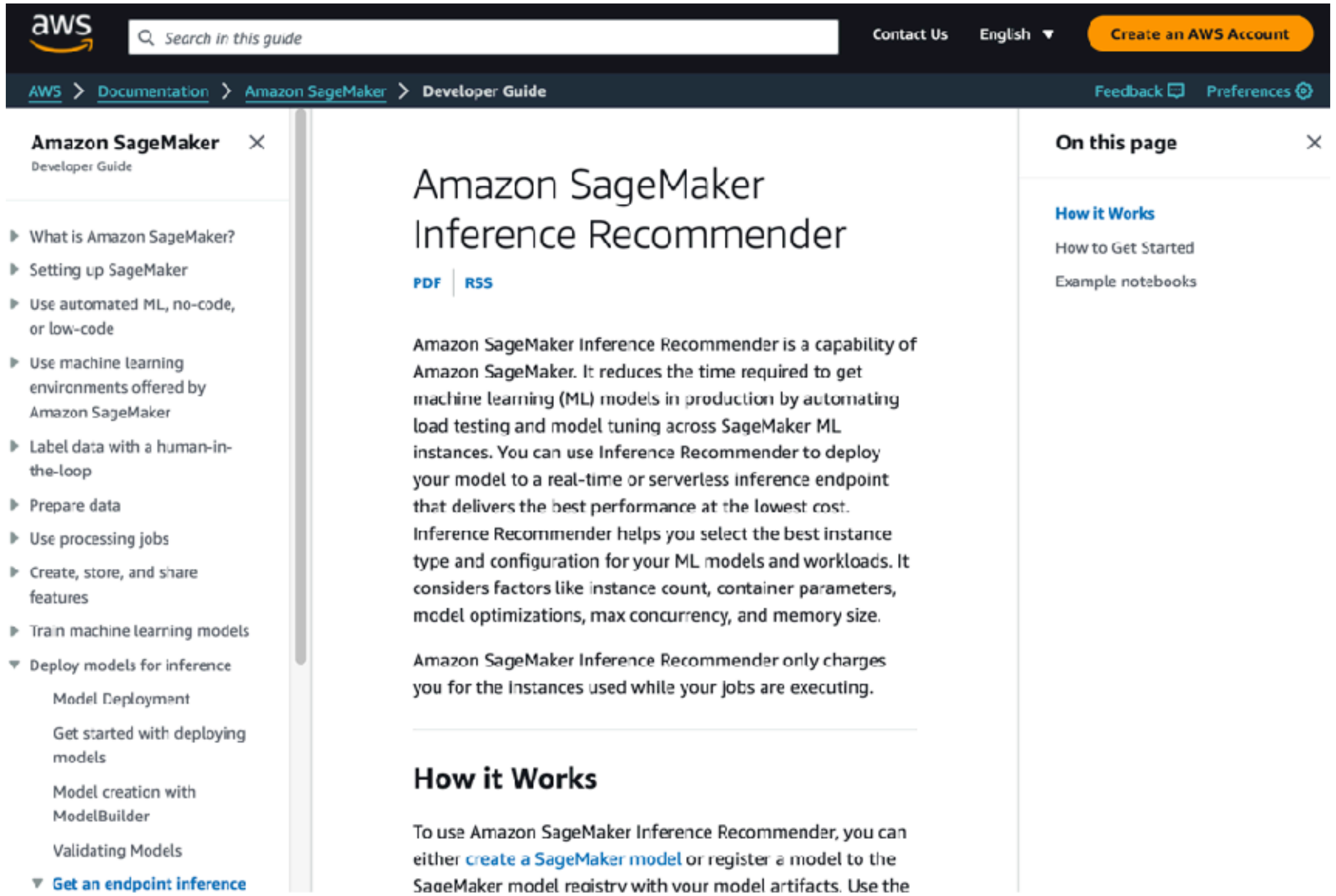
Applications: Instance Recommendation

- Instance recommendation for model deployment
- Recommend endpoint configuration (machine type, number of OMP thread, …) given a ML model



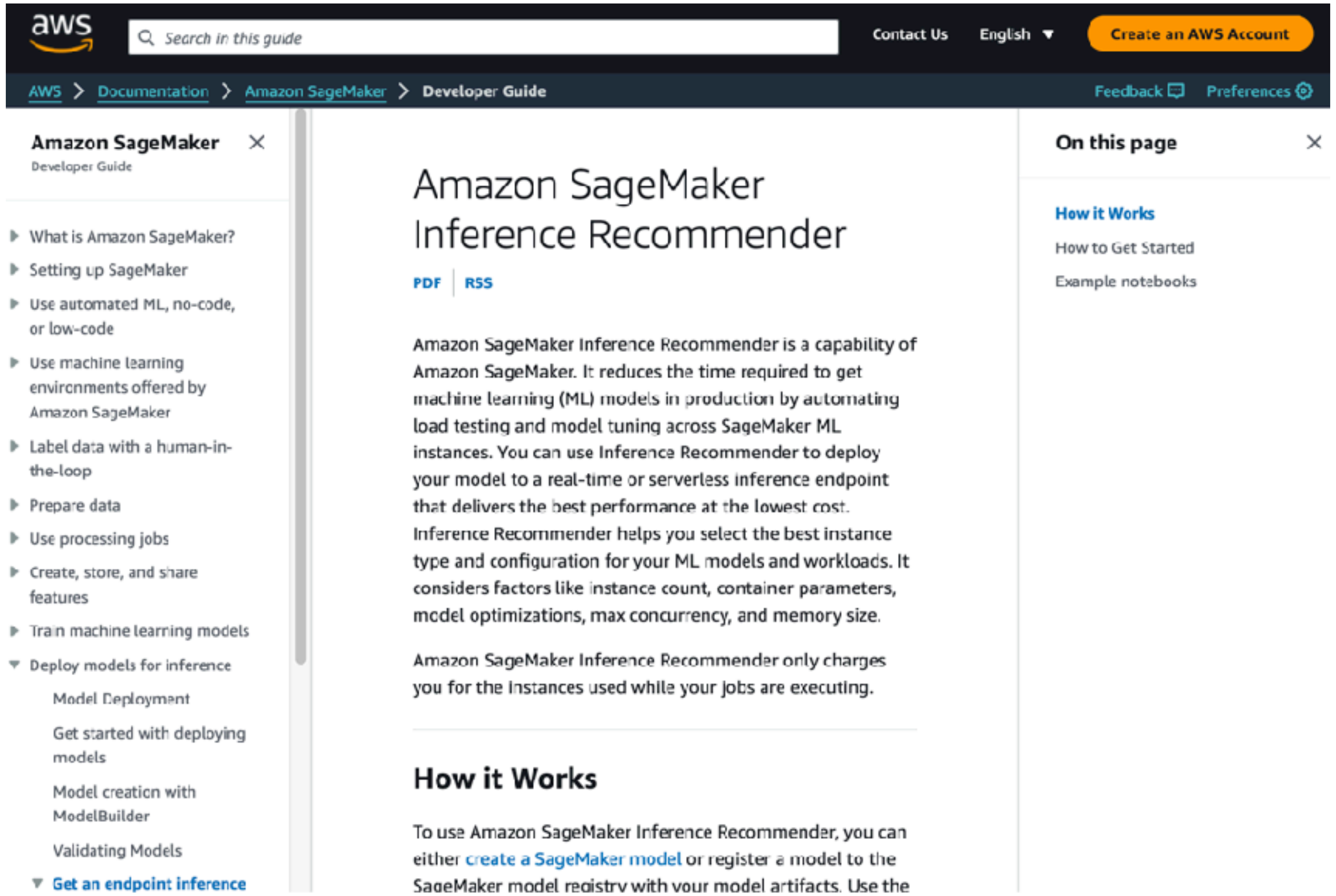
Applications: Instance Recommendation

- Instance recommendation for model deployment
- Recommend endpoint configuration (machine type, number of OMP thread, …) given a ML model
- Wants to optimise:



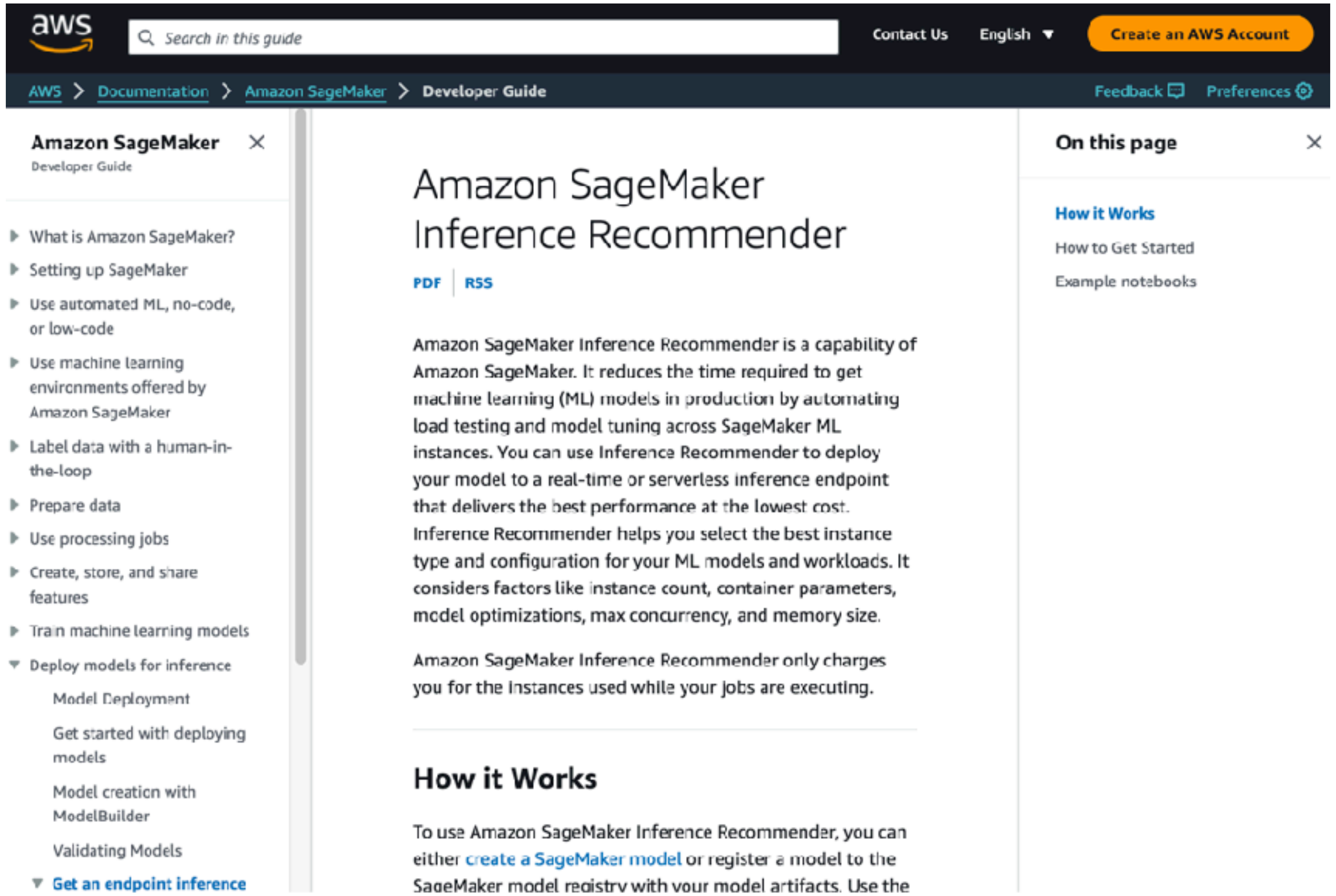
Applications: Instance Recommendation

- Instance recommendation for model deployment
- Recommend endpoint configuration (machine type, number of OMP thread, …) given a ML model
- Wants to optimise:
  - Latency



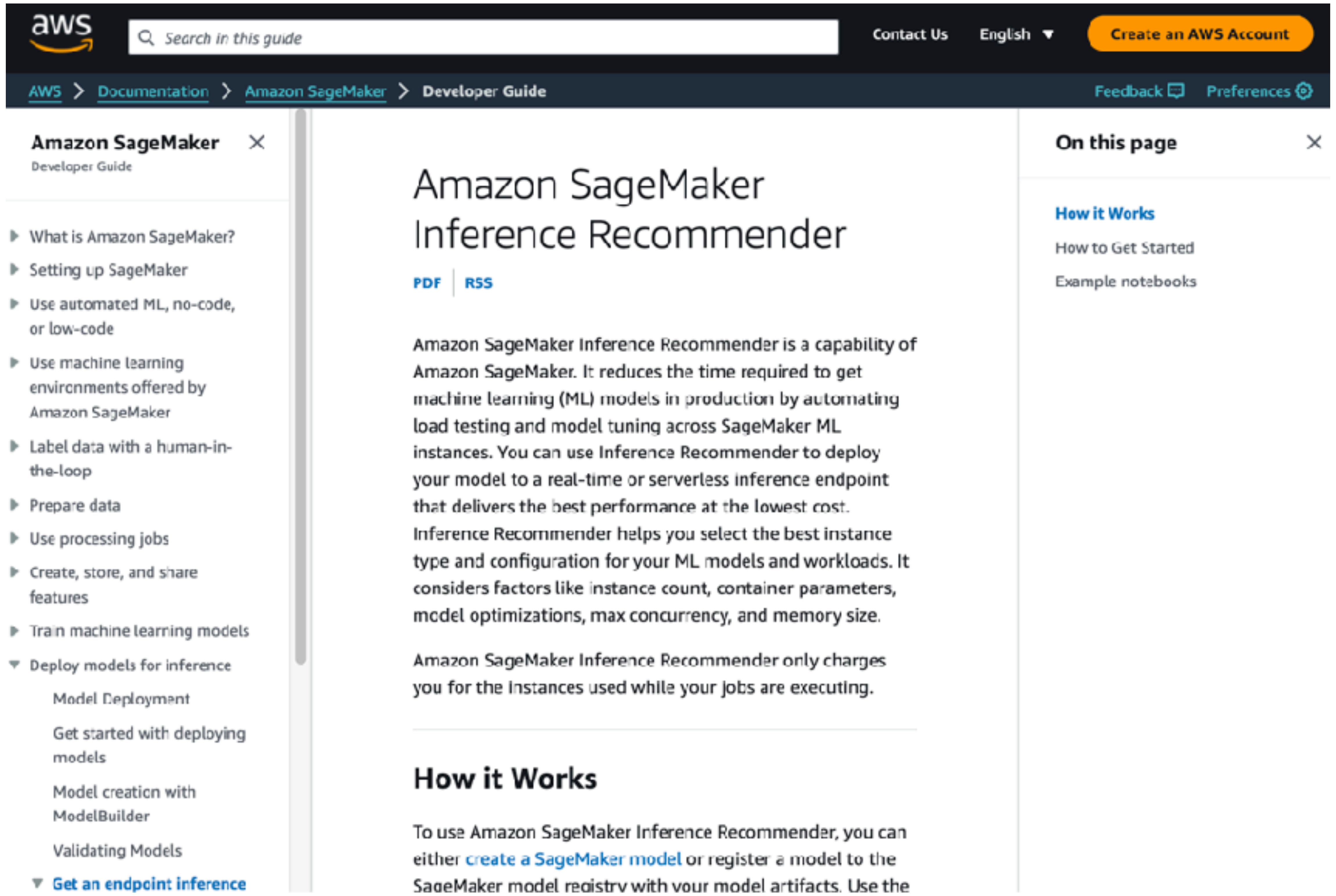
Applications: Instance Recommendation

- Instance recommendation for model deployment
- Recommend endpoint configuration (machine type, number of OMP thread, …) given a ML model
- Wants to optimise:
    - Latency
    - Throughput



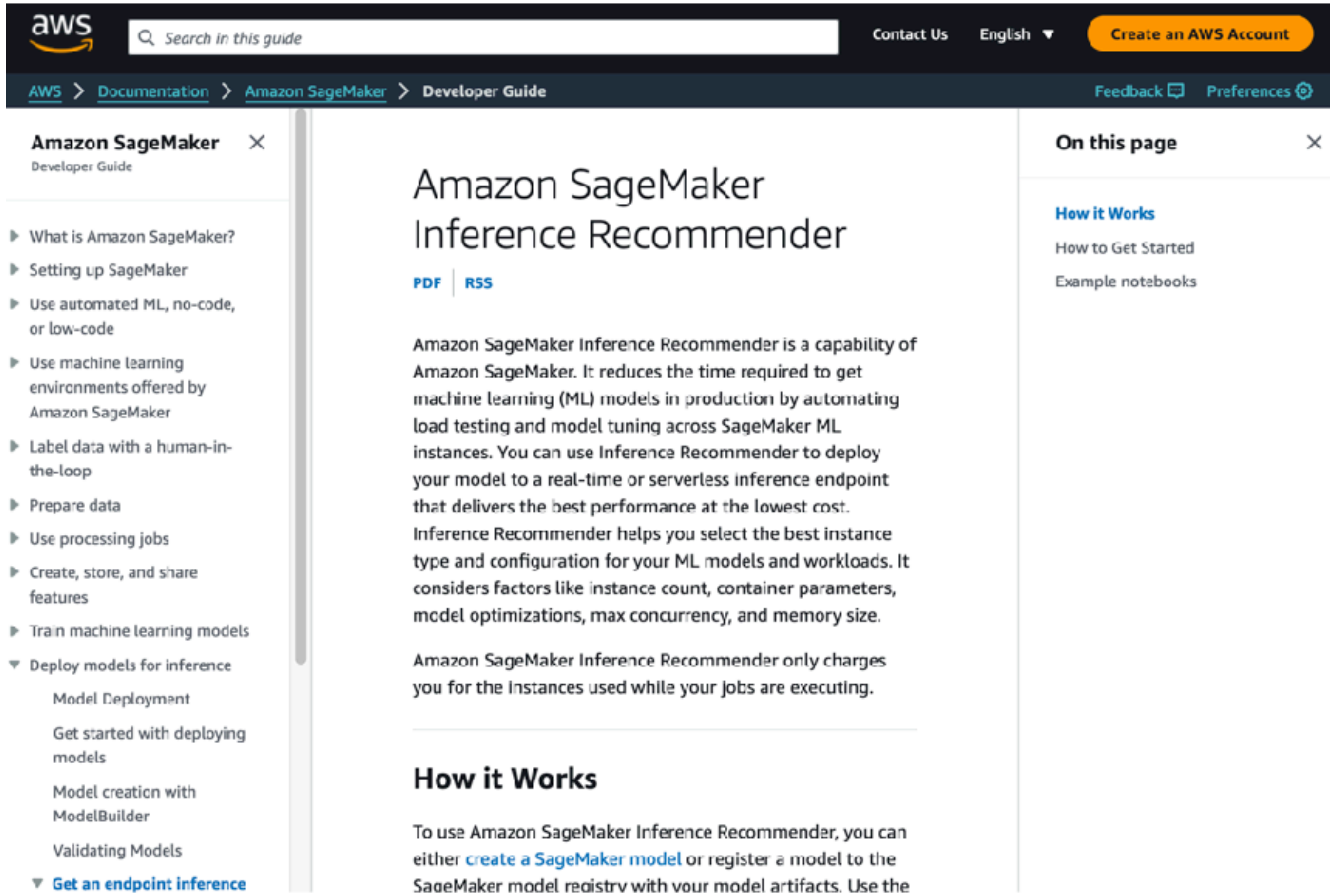
Applications: Instance Recommendation

- Instance recommendation for model deployment
- Recommend endpoint configuration (machine type, number of OMP thread, …) given a ML model
- Wants to optimise:
    - Latency
    - Throughput
    - Cost per hour

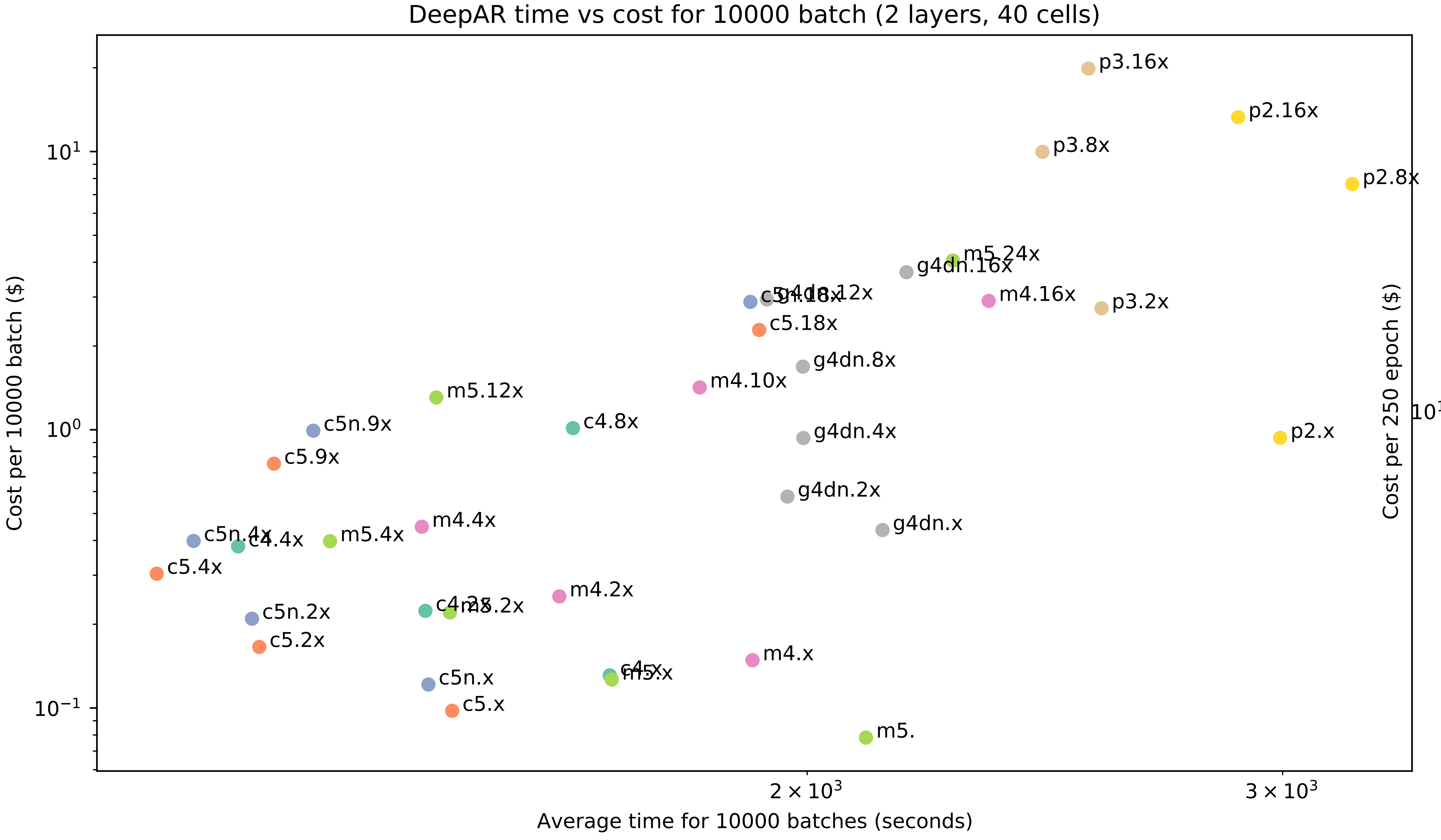


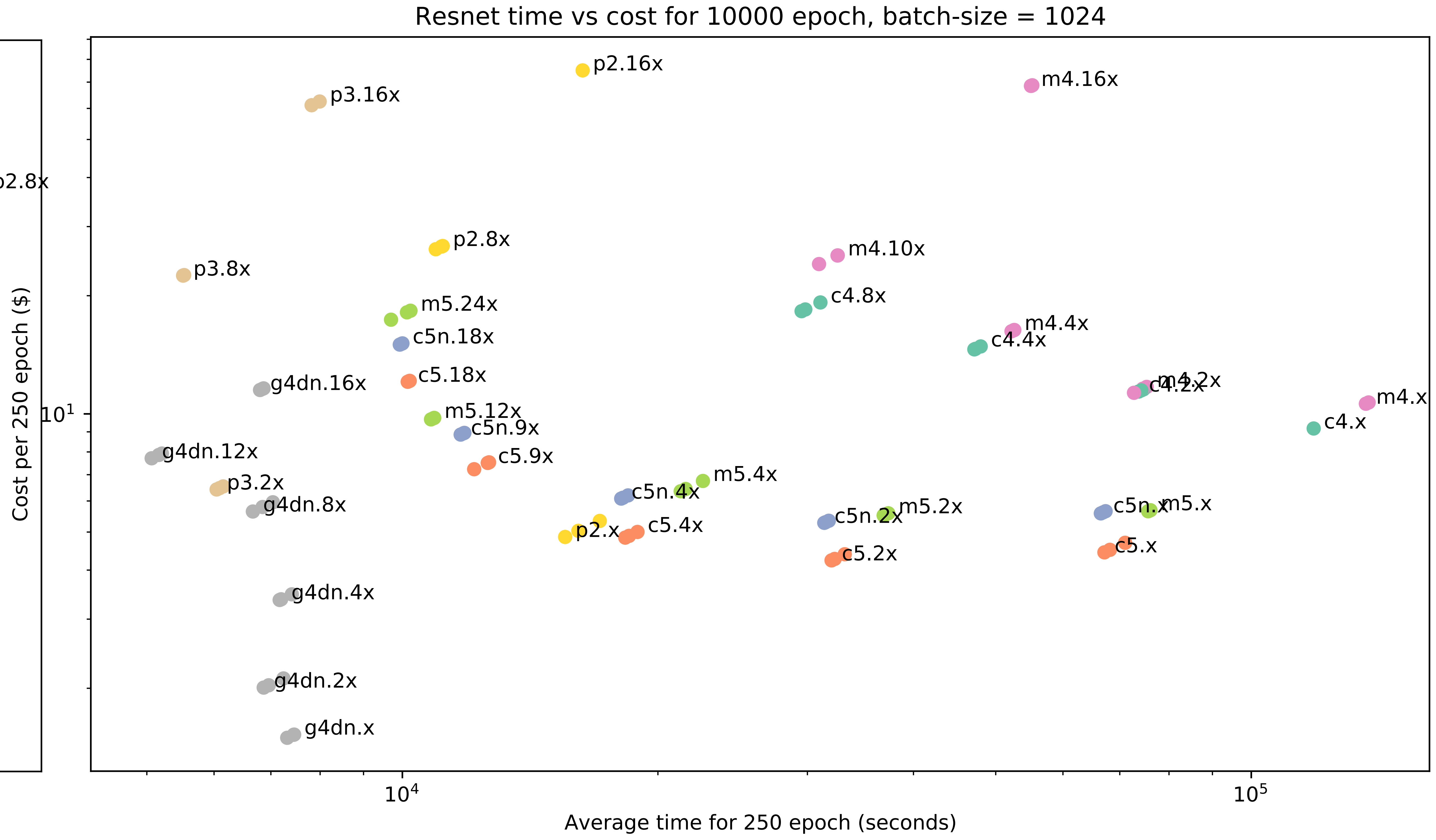
Applications: Instance Recommendation

- Instance recommendation for model deployment
- Recommend endpoint configuration (machine type, number of OMP thread, …) given a ML model
- Wants to optimise:
    - Latency
    - Throughput
    - Cost per hour
- Ideally, wants to get recommendation eg zeroshot recommendations



Applications
Machine type tuning

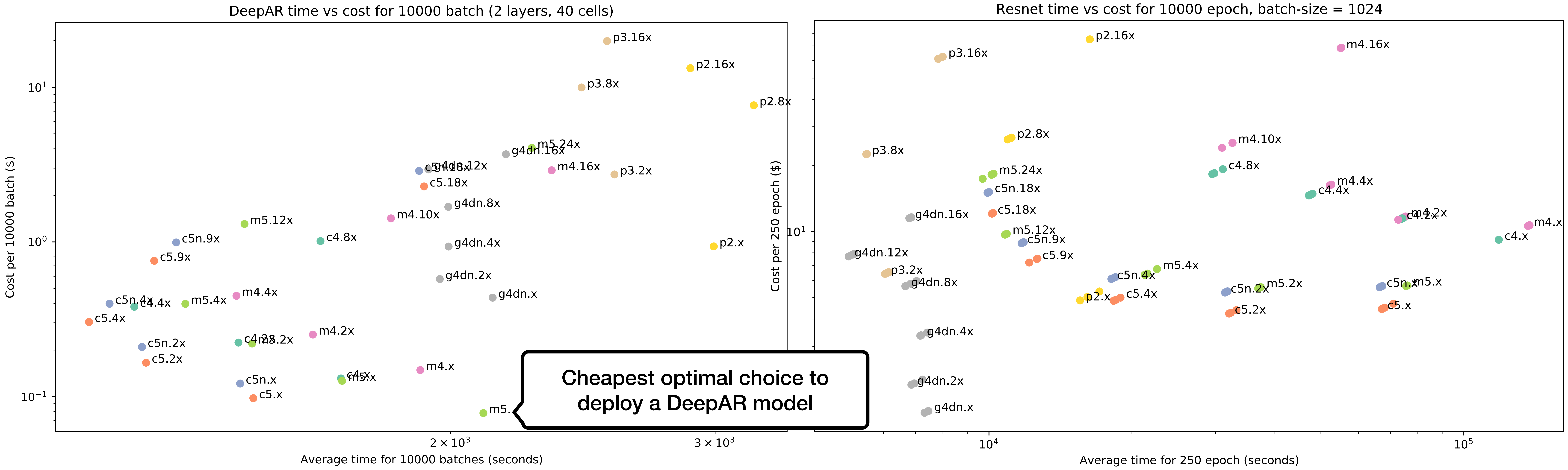




m5: 4 CPUs machine
c5.4x: 16 CPUs machine
g4dn.16x: 64 CPUs machine with one GPU (T4)
p3.2x: 8 CPU machines with V100 GPU

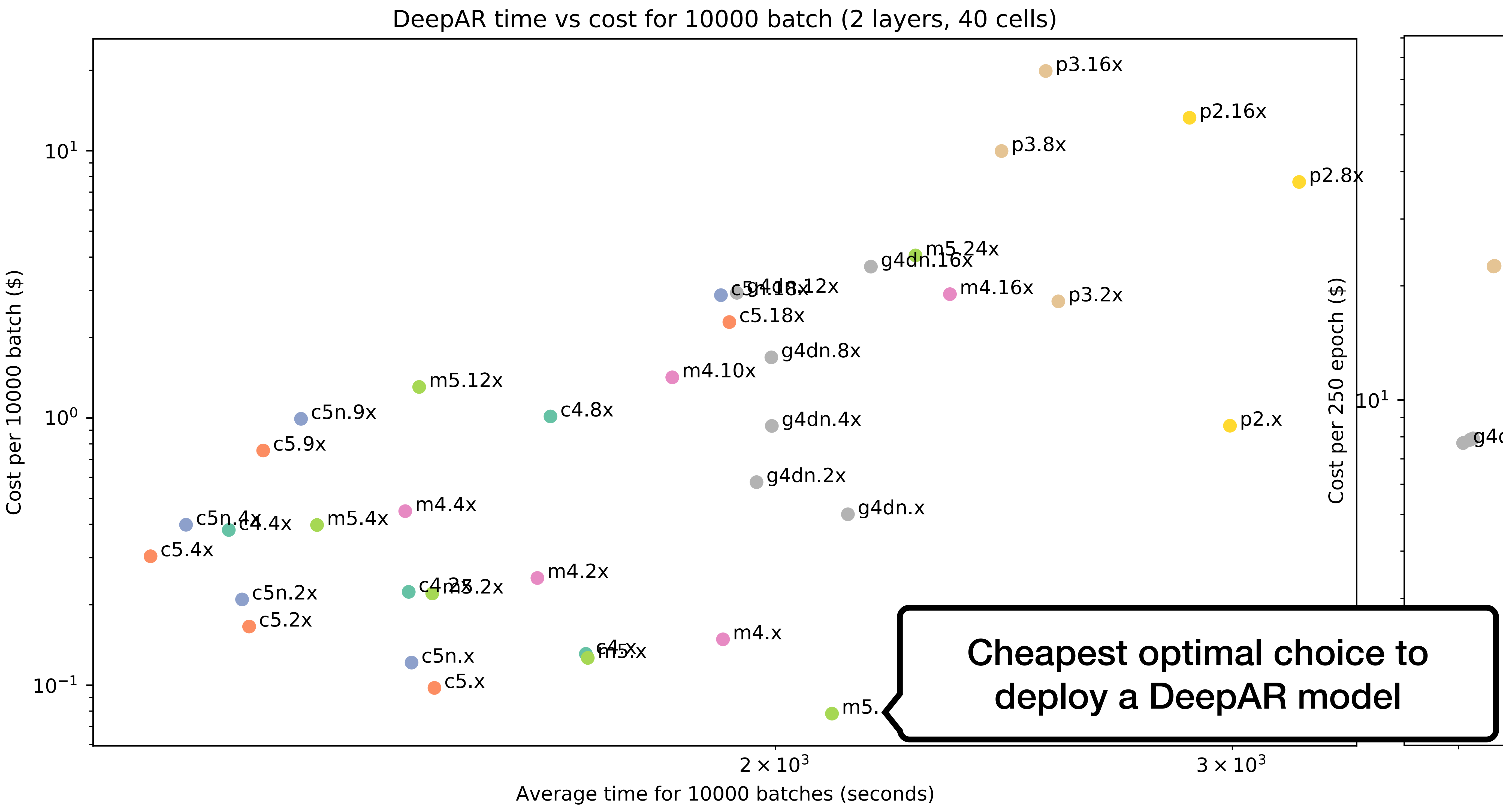
Applications

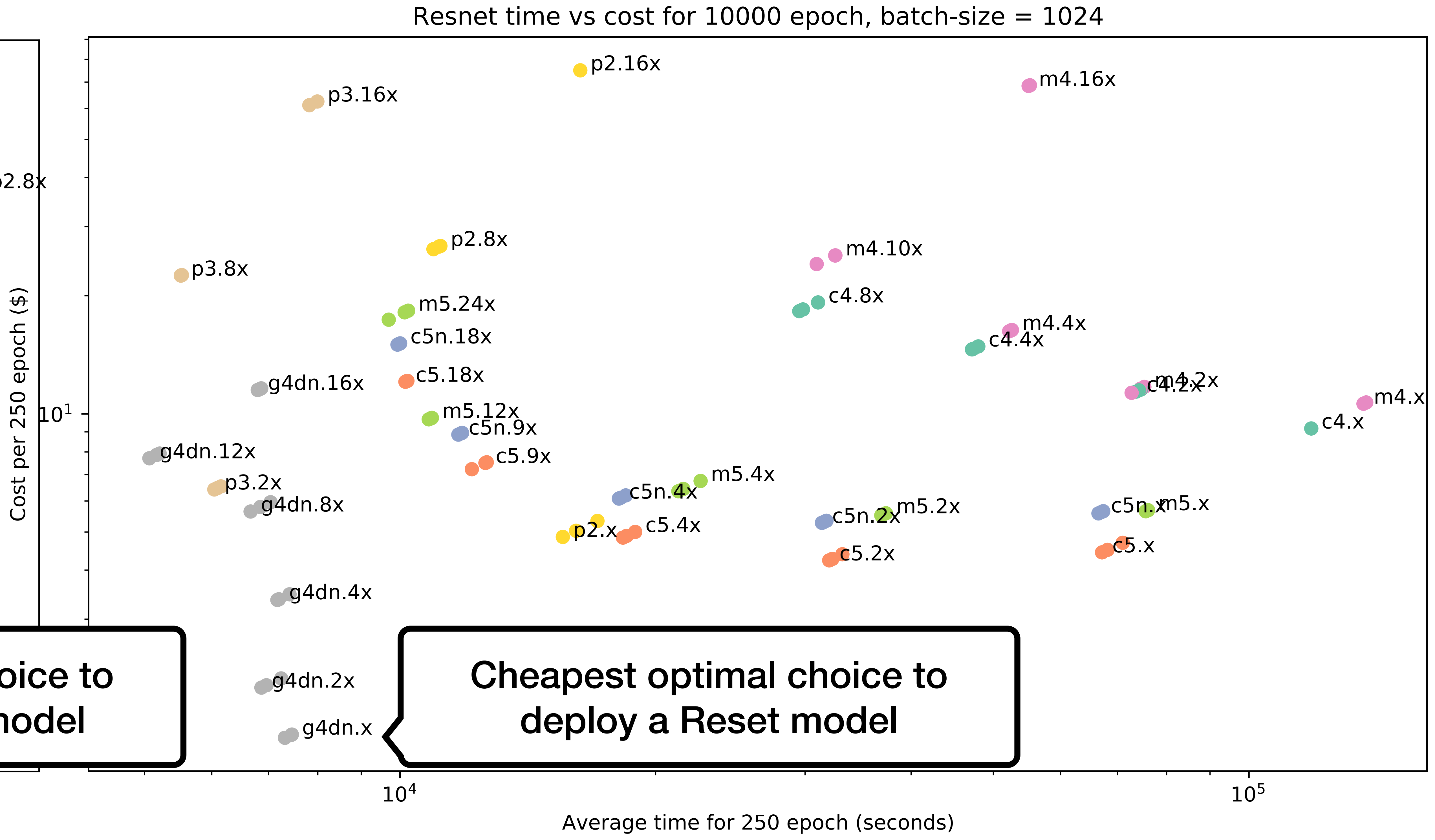
Machine type tuning



m5: 4 CPUs machine
c5.4x: 16 CPUs machine
g4dn.16x: 64 CPUs machine with one GPU (T4)
p3.2x: 8 CPU machines with V100 GPU

Applications
Machine type tuning

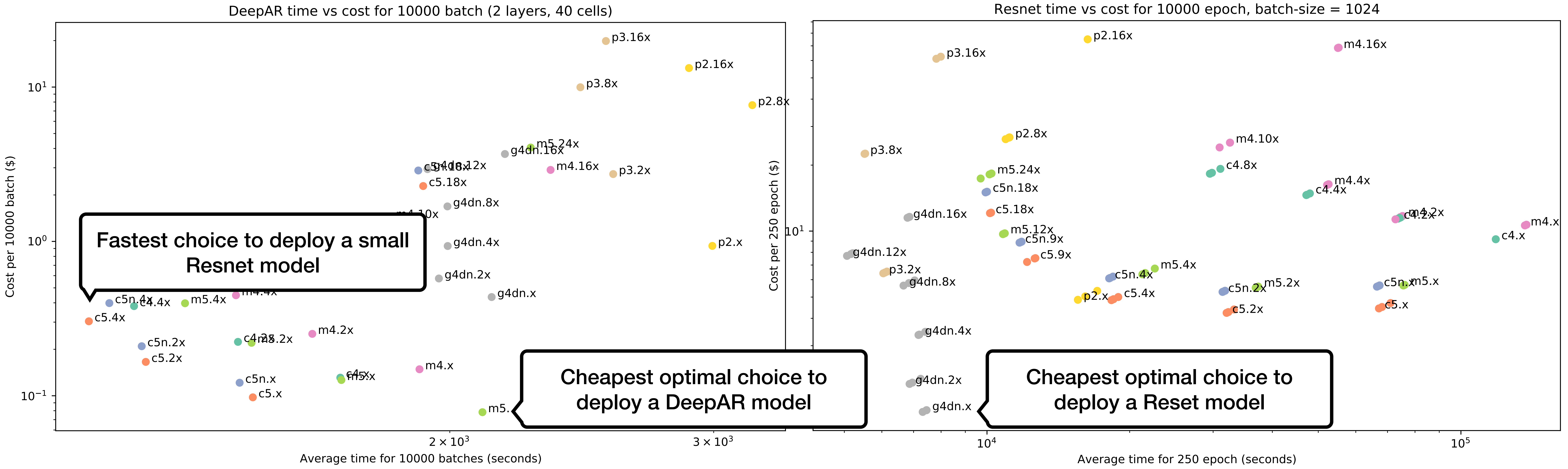




m5: 4 CPUs machine
c5.4x: 16 CPUs machine
g4dn.16x: 64 CPUs machine with one GPU (T4)
p3.2x: 8 CPU machines with V100 GPU

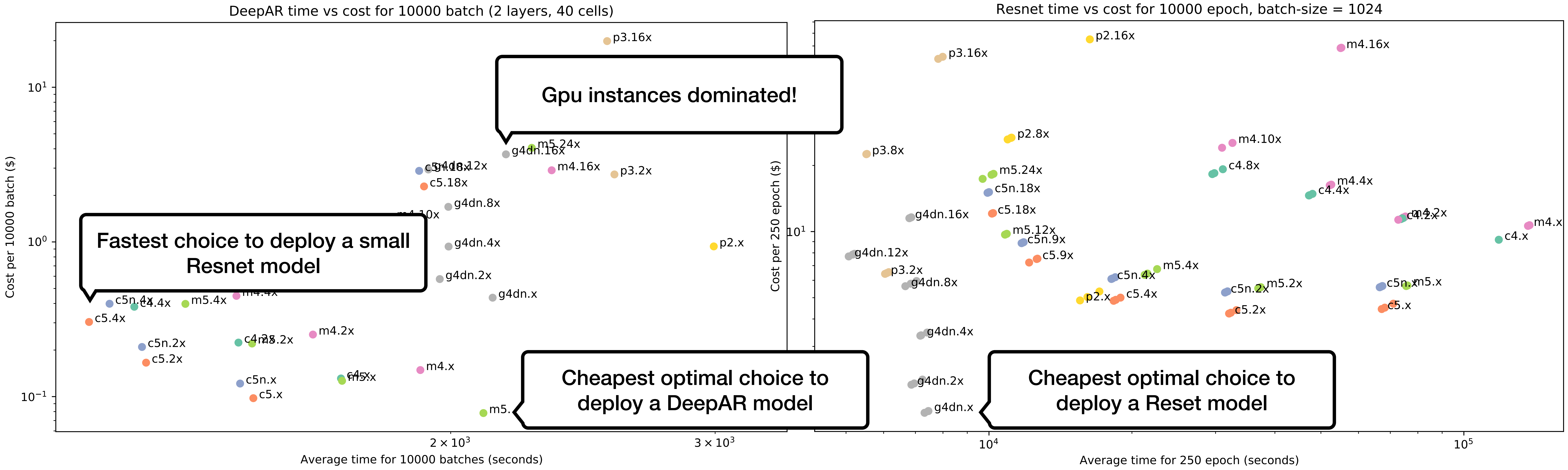
Applications

Machine type tuning



m5: 4 CPUs machine
c5.4x: 16 CPUs machine
g4dn.16x: 64 CPUs machine with one GPU (T4)
p3.2x: 8 CPU machines with V100 GPU

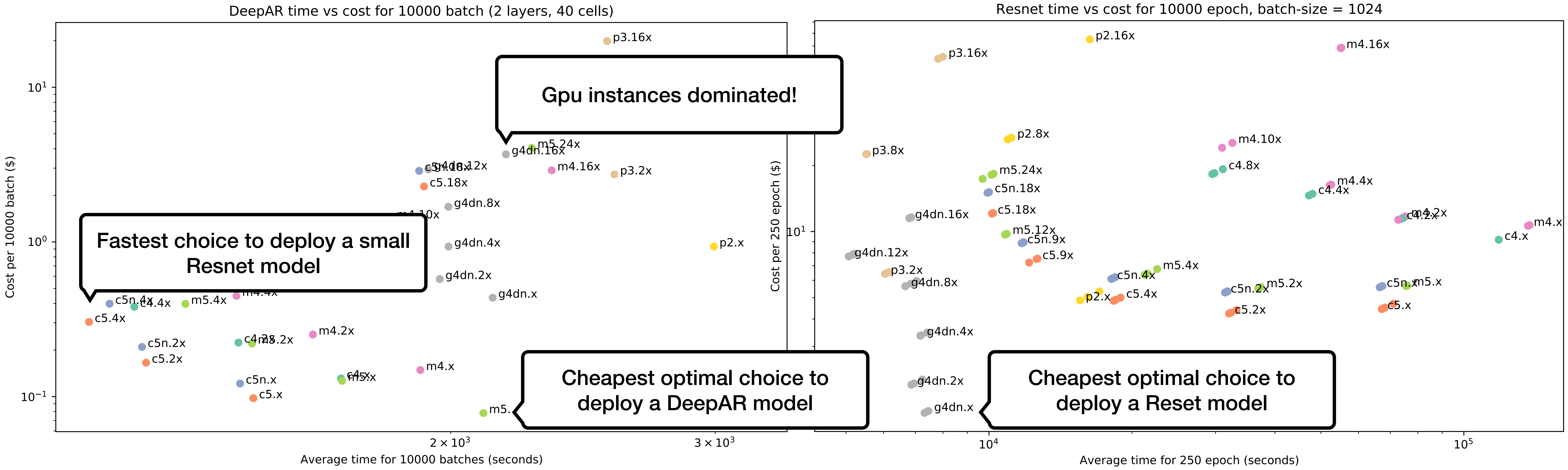
Applications
Machine type tuning



m5: 4 CPUs machine
c5.4x: 16 CPUs machine
g4dn.16x: 64 CPUs machine with one GPU (T4)
p3.2x: 8 CPU machines with V100 GPU

Applications

Machine type tuning

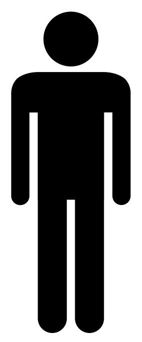


m5: 4 CPUs machine
c5.4x: 16 CPUs machine
g4dn.16x: 64 CPUs machine with one GPU (T4)
p3.2x: 8 CPU machines with V100 GPU

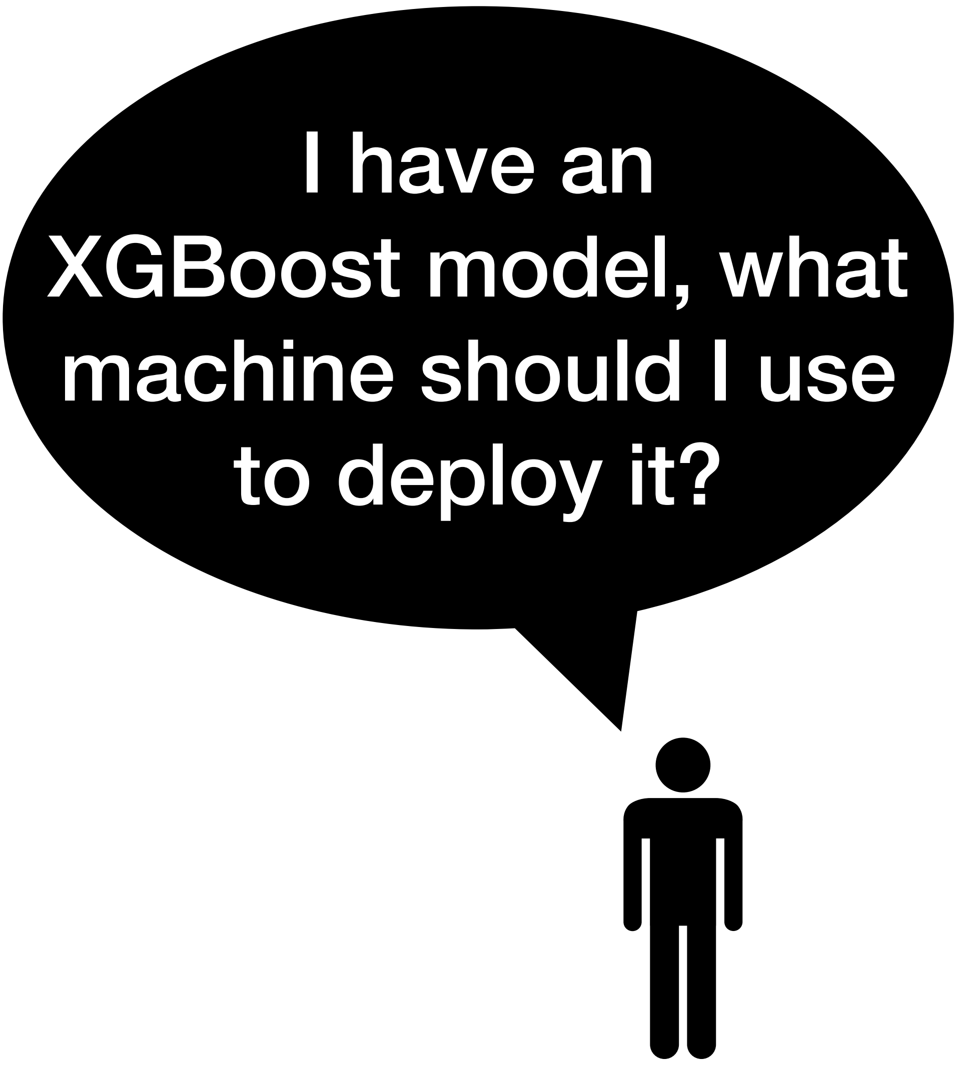
🤔 If we have some metadata on the model being used (reset, XGboost, …). Can we predict the Pareto front of hardware configurations?

Instance Recommendation

Instance Recommendation

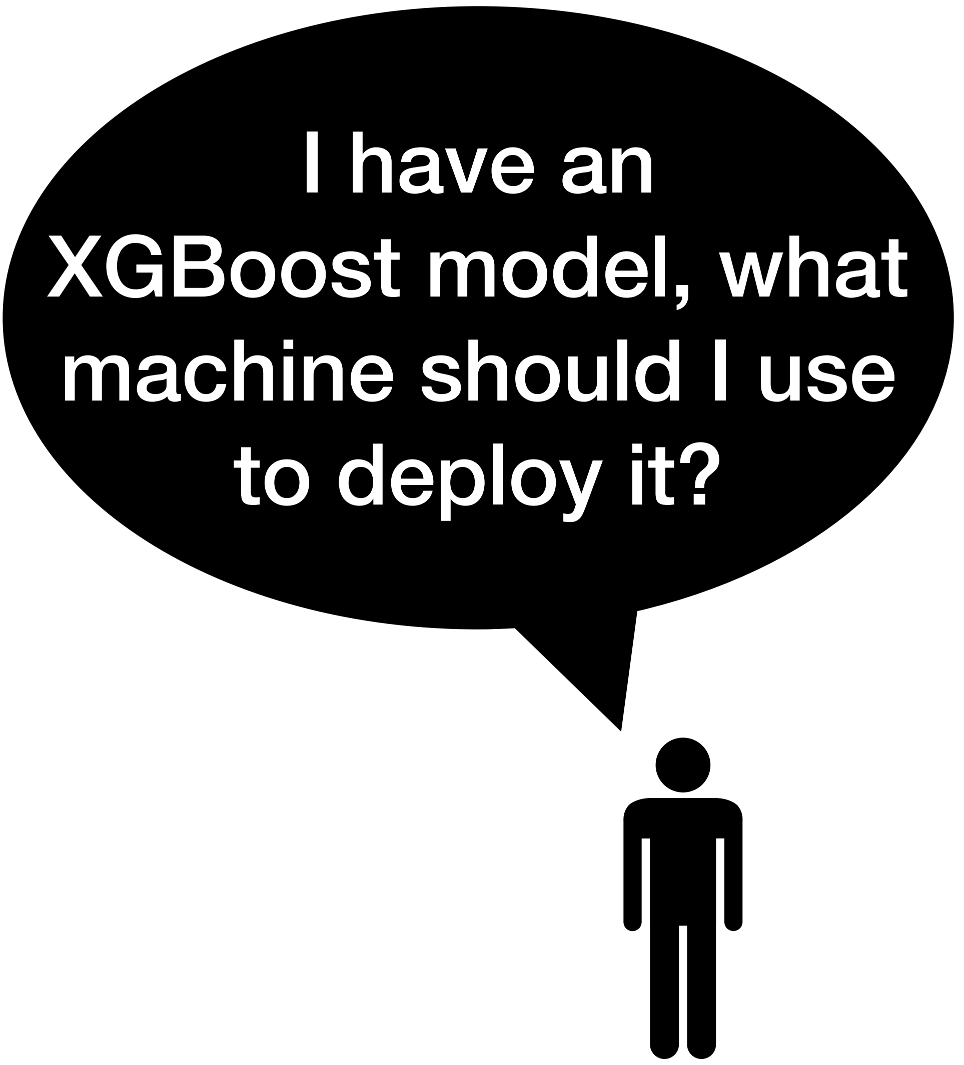


Instance Recommendation



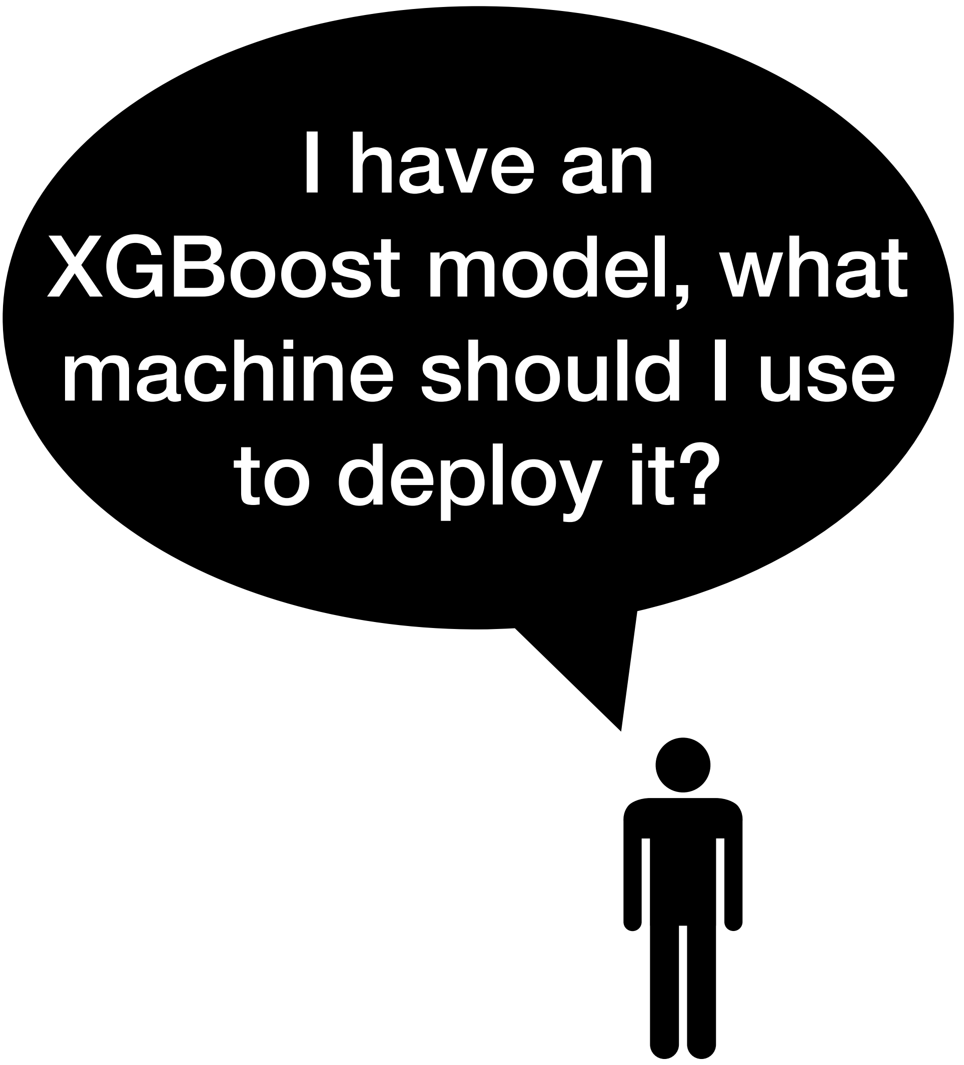
Instance Recommendation

- Sample many ML model and measure latency and cost on multiple machine



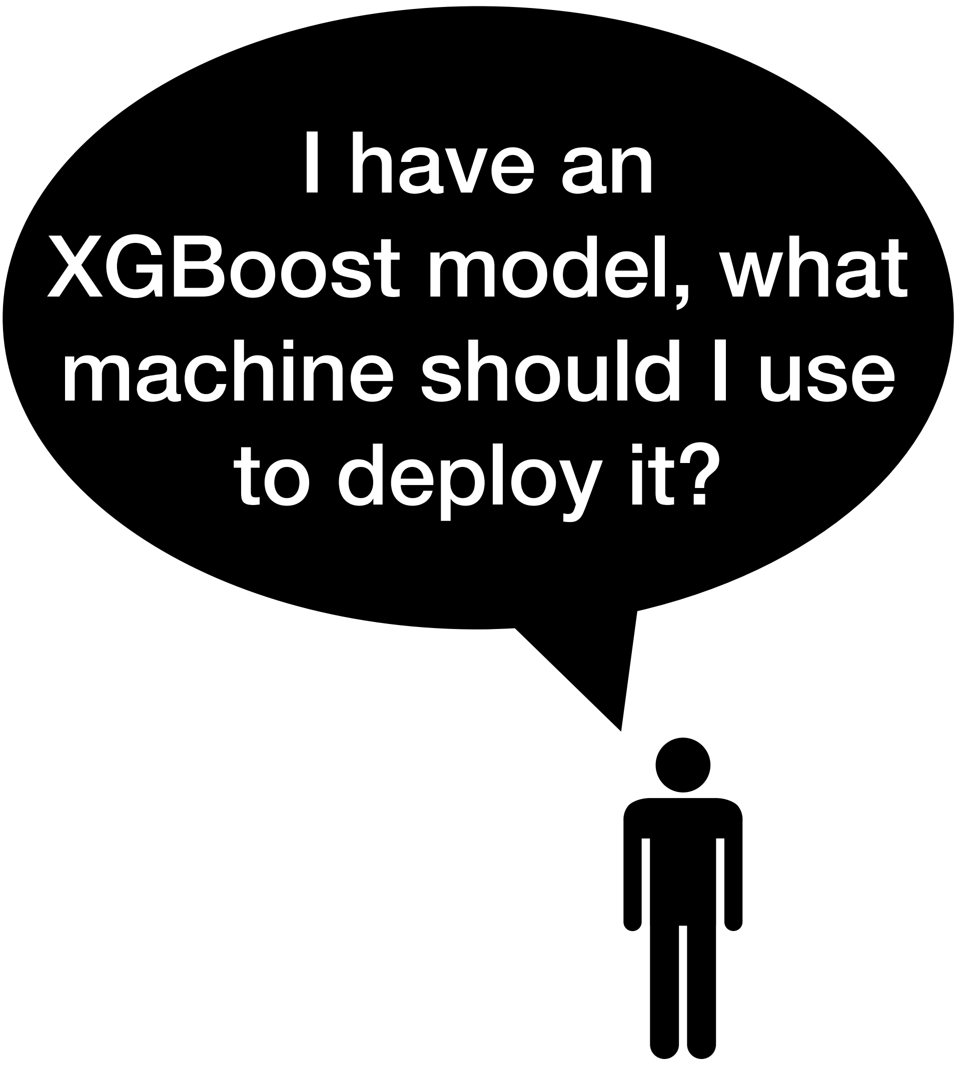
Instance Recommendation

- Sample many ML model and measure latency and cost on multiple machine
- Build a predictive model $\Phi_\theta(x, m) \in \mathbb{R}^2$ that predicts the latency and cost of the model on a machine m given metadata x



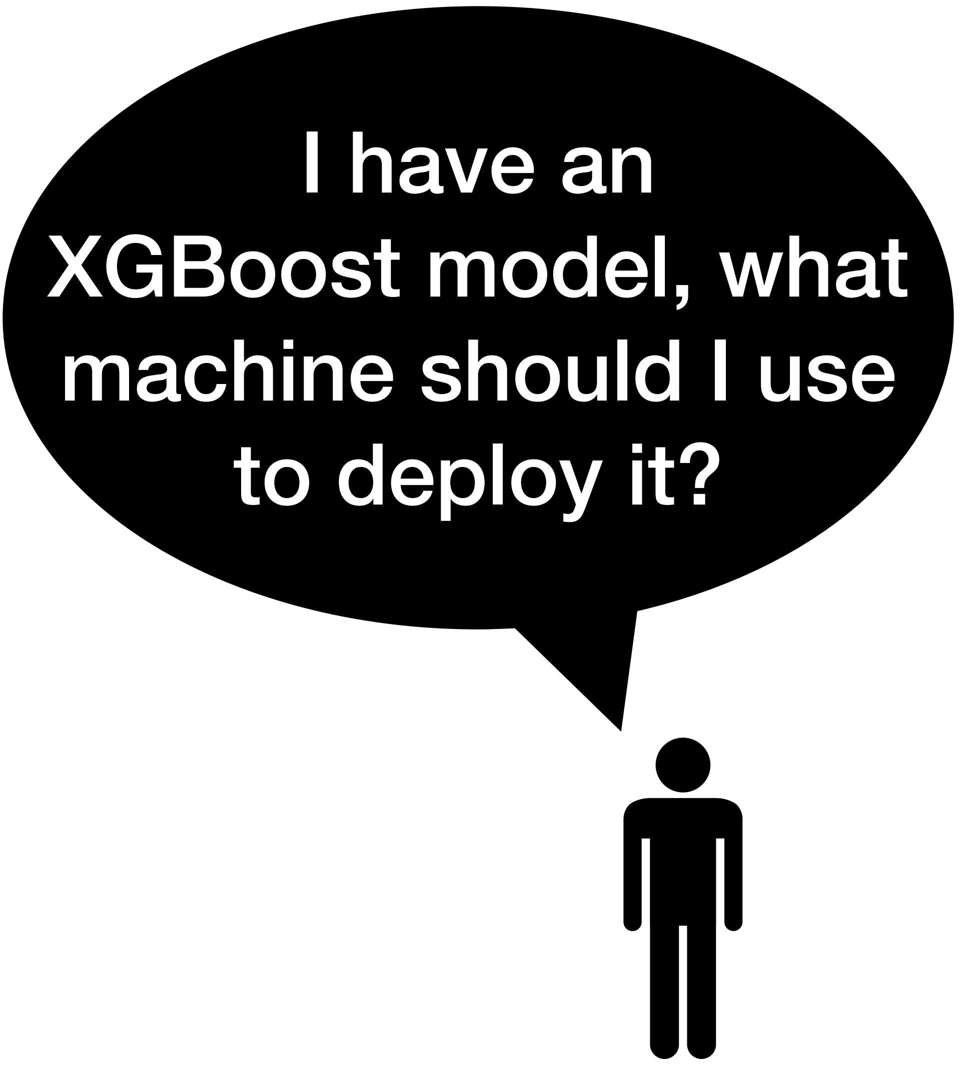
Instance Recommendation

- Sample many ML model and measure latency and cost on multiple machine
- Build a predictive model $\Phi_\theta(x, m) \in \mathbb{R}^2$ that predicts the latency and cost of the model on a machine m given metadata x
- Metadata contains features such as the framework type (Pytorch, XGBoost, …)



Instance Recommendation

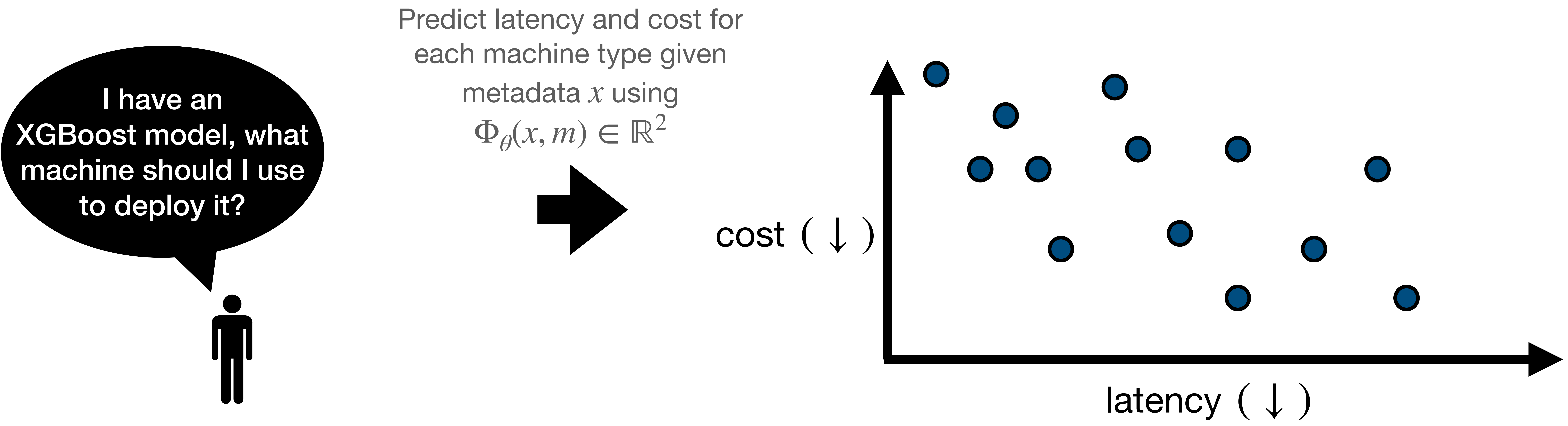
- Sample many ML model and measure latency and cost on multiple machine
- Build a predictive model $\Phi_\theta(x, m) \in \mathbb{R}^2$ that predicts the latency and cost of the model on a machine m given metadata x
- Metadata contains features such as the framework type (Pytorch, XGBoost, …)



Predict latency and cost for each machine type given metadata x using $\Phi_\theta(x, m) \in \mathbb{R}^2$

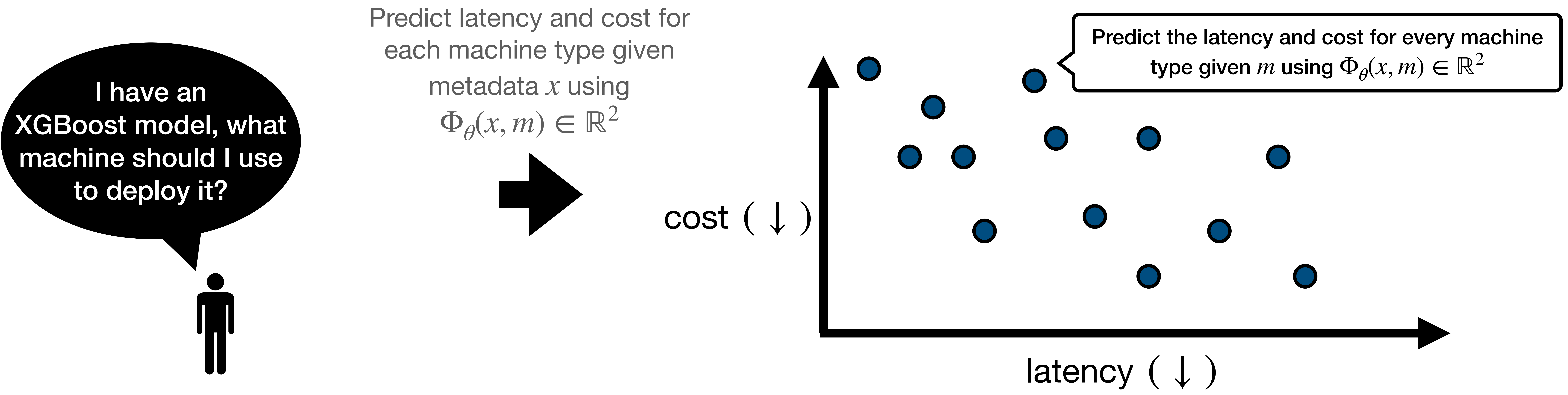
Instance Recommendation

- Sample many ML model and measure latency and cost on multiple machine
- Build a predictive model $\Phi_\theta(x, m) \in \mathbb{R}^2$ that predicts the latency and cost of the model on a machine m given metadata x
- Metadata contains features such as the framework type (Pytorch, XGBoost, …)



Instance Recommendation

- Sample many ML model and measure latency and cost on multiple machine
- Build a predictive model $\Phi_\theta(x, m) \in \mathbb{R}^2$ that predicts the latency and cost of the model on a machine m given metadata x
- Metadata contains features such as the framework type (Pytorch, XGBoost, …)



Instance Recommendation

- Sample many ML model and measure latency and cost on multiple machine
- Build a predictive model $\Phi_\theta(x, m) \in \mathbb{R}^2$ that predicts the latency and cost of the model on a machine m given metadata x
- Metadata contains features such as the framework type (Pytorch, XGBoost, …)

Instance Recommendation

- Sample many ML model and measure latency and cost on multiple machine
- Build a predictive model $\Phi_\theta(x, m) \in \mathbb{R}^2$ that predicts the latency and cost of the model on a machine m given metadata x
- Metadata contains features such as the framework type (Pytorch, XGBoost, …)

Instance Recommendation

- Sample many ML model and measure latency and cost on multiple machine
- Build a predictive model $\Phi_\theta(x, m) \in \mathbb{R}^2$ that predicts the latency and cost of the model on a machine m given metadata x
- Metadata contains features such as the framework type (Pytorch, XGBoost, …)

Applications

Tuning hyperparameter of LLM judges hardware configurations

Applications

Tuning hyperparameter of LLM judges hardware configurations

- Evaluating LLMs is expensive 💰

Applications

Tuning hyperparameter of LLM judges hardware configurations

- Evaluating LLMs is expensive 💰
- It costs ~3500$ to evaluate one model on Chatbot Arena with human annotations…

Applications

Tuning hyperparameter of LLM judges hardware configurations

- Evaluating LLMs is expensive 💰
- It costs ~3500$ to evaluate one model on Chatbot Arena with human annotations…
- To get an leaderboard of ELO ratings

Applications

Tuning hyperparameter of LLM judges hardware configurations

- Evaluating LLMs is expensive 💰
- It costs ~3500$ to evaluate one model on Chatbot Arena with human annotations…
- To get an leaderboard of ELO ratings
- 🤔 Can we get a cheaper approximation?

Applications

Tuning hyperparameter of LLM judges hardware configurations

- Evaluating LLMs is expensive 💰
- It costs ~3500$ to evaluate one model on Chatbot Arena with human annotations…
- To get an leaderboard of ELO ratings
- 🤔 Can we get a cheaper approximation?

Rank* (UB)	Model	Arena Score	95% CI	Votes	Organization	License
1	GPT-4o-2024-05-13	1287	+3/-3	56905	OpenAI	Proprietary
2	Claude 3.5 Sonnet	1272	+4/-4	24913	Anthropic	Proprietary
2	Gemini-Advanced-0514	1267	+3/-3	42981	Google	Proprietary
3	Gemini-1.5-Pro-API-0514	1262	+3/-3	49828	Google	Proprietary
4	Gemini-1.5-Pro-API-0409-Preview	1258	+3/-3	55567	Google	Proprietary
4	GPT-4-Turbo-2024-04-09	1257	+3/-4	72512	OpenAI	Proprietary
6	GPT-4-1106-preview	1251	+3/-3	86474	OpenAI	Proprietary
7	Claude 3 Opus	1248	+2/-2	143189	Anthropic	Proprietary
8	GPT-4-0125-preview	1246	+3/-2	79732	OpenAI	Proprietary

Applications

Tuning hyperparameter of LLM judges hardware configurations

- Evaluating LLMs is expensive 💰
- It costs ~3500$ to evaluate one model on Chatbot Arena with human annotations…
- To get an leaderboard of ELO ratings
- 🤔 Can we get a cheaper approximation?

Rank* (UB)	Model	Arena Score	95% CI	Votes	Organization	License
1	GPT-4o-2024-05-13	1287	+3/-3	56905	OpenAI	Proprietary
2	Claude 3.5 Sonnet	1272	+4/-4	24913	Anthropic	Proprietary
2	Gemini-Advanced-0514	1267	+3/-3	42981	Google	Proprietary
3	Gemini-1.5-Pro-API-0514	1262	+3/-3	49828	Google	Proprietary
4	Gemini-1.5-Pro-API-0409-Preview	1258	+3/-3	55567	Google	Proprietary
4	GPT-4-Turbo-2024-04-09	1257	+3/-4	72512	OpenAI	Proprietary
6	GPT-4-1106-preview	1251	+3/-3	86474	OpenAI	Proprietary
7	Claude 3 Opus	1248	+2/-2	143189	Anthropic	Proprietary
8	GPT-4-0125-preview	1246	+3/-2	79732	OpenAI	Proprietary

Work in progress

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge
    - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge
  - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
  - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge
  - **System prompt**: You’re an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
  - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
  - **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

  - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
  - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
  - **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings
  - **Model B**: To stream at 720p 30fps to Twitch while playing an online videogame, you'll need to consider the upload speed requirements…

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

  - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
  - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
  - **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings
  - **Model B**: To stream at 720p 30fps to Twitch while playing an online videogame, you'll need to consider the upload speed requirements…
  - **Best model** (must be A or B):

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

- **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
- **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
- **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings
- **Model B**: To stream at 720p 30fps to Twitch while playing an online videogame, you'll need to consider the upload speed requirements…
- **Best model** (must be A or B):

LLM call

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

  - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
  - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
  - **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings
  - **Model B**: To stream at 720p 30fps to Twitch while playing an online videogame, you'll need to consider the upload speed requirements…
  - **Best model** (must be A or B):

LLM call

"A"

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

  - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
  - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
  - **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings
  - **Model B**: To stream at 720p 30fps to Twitch while playing an online videogame, you'll need to consider the upload speed requirements…
  - **Best model** (must be A or B):

LLM call → "A"

- Use fix baseline for model A (typically GPT4)

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

  - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
  - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
  - **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings
  - **Model B**: To stream at 720p 30fps to Twitch while playing an online videogame, you'll need to consider the upload speed requirements…
  - **Best model** (must be A or B):

LLM call → "A"

- Use fix baseline for model A (typically GPT4)

- Report win rate for all methods

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

> - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
> - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
> - **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings
> - **Model B**: To stream at 720p 30fps to Twitch while playing an online videogame, you'll need to consider the upload speed requirements…
> - **Best model** (must be A or B):

LLM call

"A"

LLM judge has **many** hyperparameters!
- LLM model (llama3-70B, llama3-8B, GPT4)
- Prompt being used
- Judge LLM inference parameters (temperature & topk)
- Number of LLM samples
- Float precision (FP8, BF16, …)
- Number of instructions evaluated

- Use fix baseline for model A (typically GPT4)
- Report win rate for all methods

Applications

Tuning hyperparameter of LLM judges

- Cheaper alternative use LLM as a judge

> - **System prompt**: You're an LLM that evaluates the strength of other LLMs, please evaluate the two options provided carefully and answer which of model A or B is better.
> - **Instruction**: What is the minimum broadband upload speed to stream at 720p 30fps to twitch while playing an online videogame? Twitch recommends 3000kbps upload speed in my streaming software, but how much extra headroom do I need so the twitch stream and online gameplay are both smooth?
> - **Model A**: Streaming on Twitch at 720p 30fps with a bitrate of 3000kbps (kilobits per second) is within the recommended settings
> - **Model B**: To stream at 720p 30fps to Twitch while playing an online videogame, you'll need to consider the upload speed requirements…
> - **Best model** (must be A or B):

LLM call

"A"

- Use fix baseline for model A (typically GPT4)
- Report win rate for all methods

LLM judge has **many** hyperparameters!

- LLM model (llama3-70B, llama3-8B, GPT4)
- Prompt being used
- Judge LLM inference parameters (temperature & topk)
- Number of LLM samples
- Float precision (FP8, BF16, …)
- Number of instructions evaluated

… and **multiple** objectives

- Spearman correlation with ELO ratings
- Dollar cost to evaluate a model

Applications

Tuning hyperparameter of LLM judges

Applications

Tuning hyperparameter of LLM judges

Table 1. Separability and agreement per benchmark.

	Chatbot Arena (English-only)	MT-bench	AlpacaEval 2.0 LC (Length Controlled)	Arena-Hard-Auto-v0.1
Avg #prompts per model eval	10,000+	160	800	1,000
Agreement to Chatbot Arena with 95% CI	N/A	26.1%	81.2%	**89.1%**
Spearman Correlation	N/A	91.3%	90.8%	**94.1%**
Separability with 95% CI	85.8%	22.6%	83.2%	**87.4%**
Real-world	Yes	Mixed	Mixed	**Yes**
Freshness	Live	Static	Static	**Frequent Updates**
Eval cost per model	Very High	$10	$10	$25
Judge	Human	LLM	LLM	LLM

▸ *Results based on 20 top models from Chatbot Arena that are also presented on Alpaca Eval

Applications

Tuning hyperparameter of LLM judges

Table 1. Separability and agreement per benchmark.

	Chatbot Arena (English-only)	MT-bench	AlpacaEval 2.0 LC (Length Controlled)	Arena-Hard-Auto-v0.1
Avg #prompts per model eval	10,000+	160	800	1,000
Agreement to Chatbot Arena with 95% CI	N/A	26.1%	81.2%	**89.1%**
Spearman Correlation	N/A	91.3%	90.8%	**94.1%**
Separability with 95% CI	85.8%	22.6%	83.2%	**87.4%**
Real-world	Yes	Mixed	Mixed	**Yes**
Freshness	Live	Static	Static	**Frequent Updates**
Eval cost per model	Very High	$10	$10	$25
Judge				

▸ *Resu

🤩 **Ideal case for multi-objective optimization!**

Applications

Tuning hyperparameter of LLM judges

Table 1. Separability and agreement per benchmark.

	Chatbot Arena (English-only)	MT-bench	AlpacaEval 2.0 LC (Length Controlled)	Arena-Hard-Auto-v0.1
Avg #prompts per model eval	10,000+	160	800	1,000
Agreement to Chatbot Arena with 95% CI	N/A	26.1%	81.2%	**89.1%**
Spearman Correlation	N/A	91.3%	90.8%	**94.1%**
Separability with 95% CI	85.8%	22.6%	83.2%	**87.4%**
Real-world	Yes	Mixed	Mixed	**Yes**
Freshness	Live	Static	Static	**Frequent Updates**
Eval cost per model	Very High	$10	$10	$25
Judge				

▸ *Resu

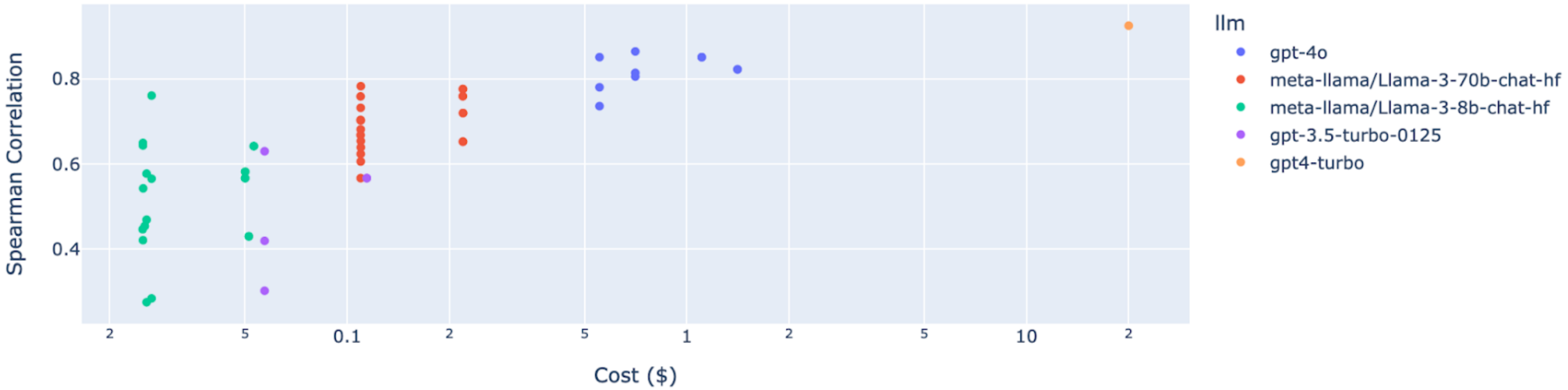
🤩 **Ideal case for multi-objective optimization!**

🤔 **Main difficulties to overcome**
Evaluating one judge configurations is too expensive (10$ x #models ~ 400$)

Two techniques: subselect most informative instructions and use multifidelity

Applications

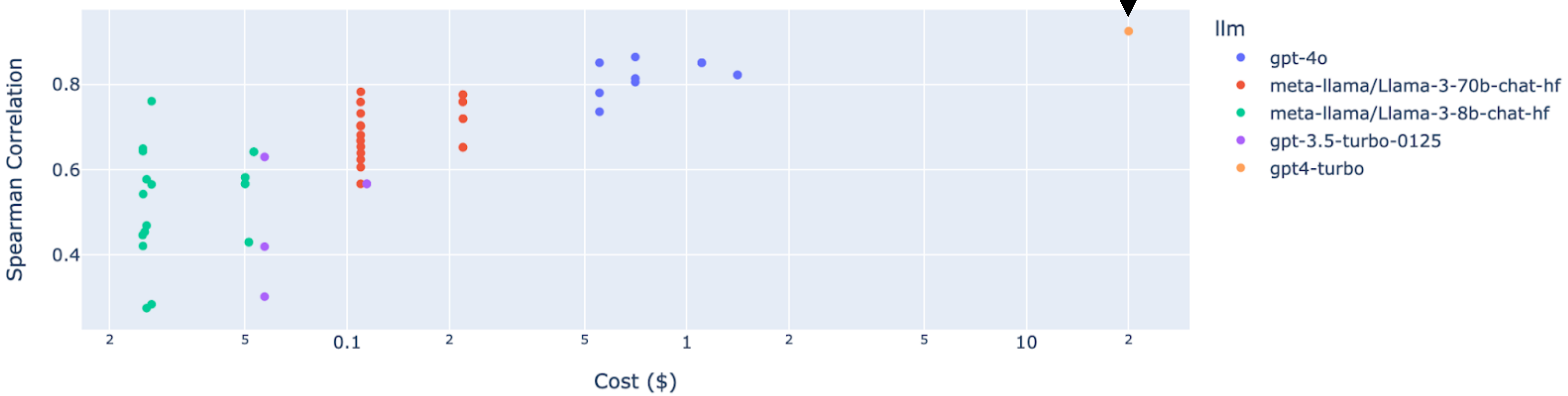
Tuning hyperparameter of LLM judges hardware configurations



Applications

Tuning hyperparameter of LLM judges hardware configurations

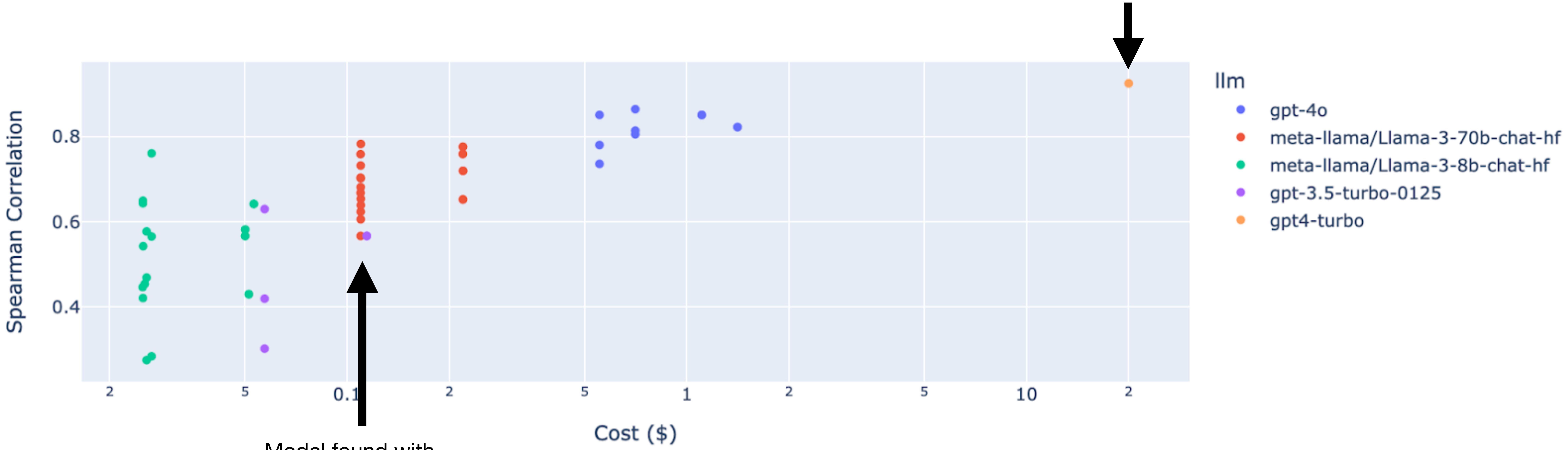
Baseline AlpacaEval hyperparameter, Spearman correlation with ELO ratings **0.93** and cost **20$**



Applications

Tuning hyperparameter of LLM judges hardware configurations

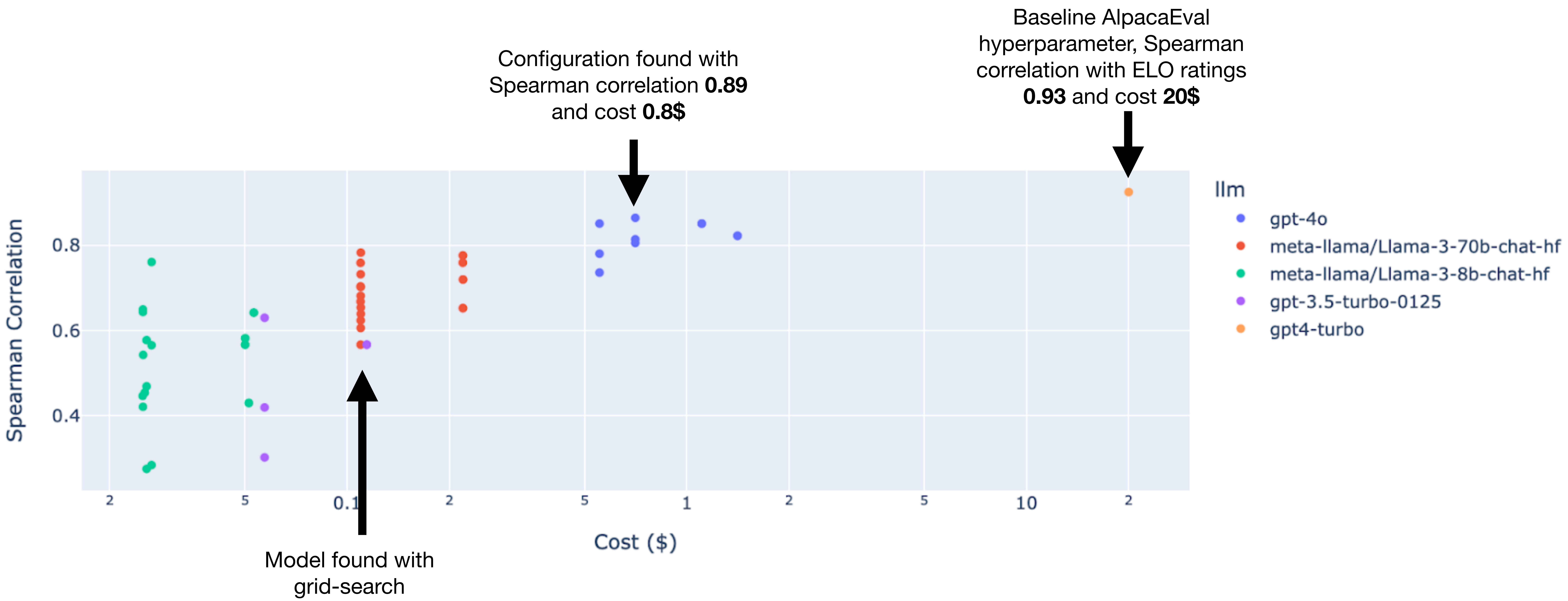
Baseline AlpacaEval hyperparameter, Spearman correlation with ELO ratings **0.93** and cost **20$**



Model found with grid-search

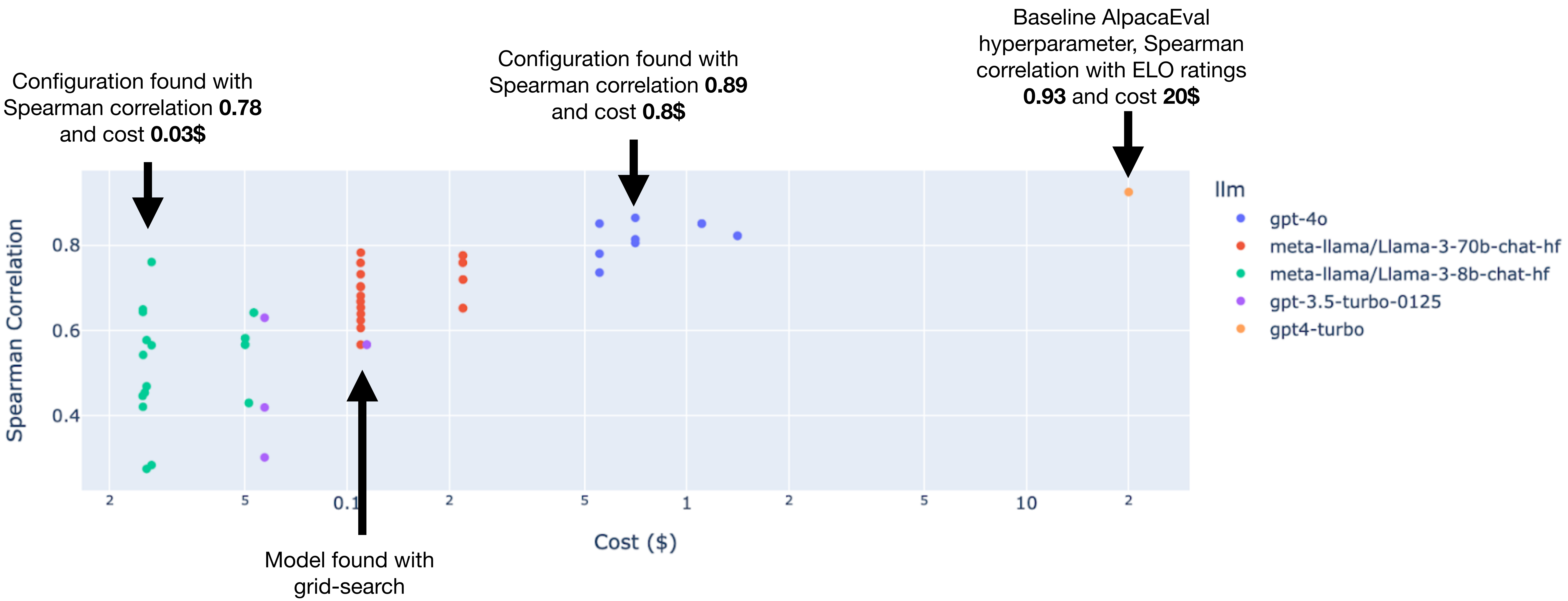
Applications

Tuning hyperparameter of LLM judges hardware configurations



Applications

Tuning hyperparameter of LLM judges hardware configurations



Code and libraries 👩🏽‍💻

Code

Some multiobjective libraries

Code

Some multiobjective libraries

README

GPareto

GPareto: Gaussian Processes for Pareto Front Estimation and Optimization

This R package provides tools for multi-objective optimization of expensive black-box functions along with estimation of Pareto fronts.

Installation

For the stable version: http://cran.r-project.org/package=GPareto

This the development version, contributions are welcomed.

Code

Some multiobjective libraries

README

GPareto

GPareto: Gaussian Processes for Pareto Front Estimation and Optimization

This R package provides tools for multi-objective
black-box functions along with estimation of Paret

Installation

For the stable version: http://cran.r-project.org/package=GPareto

This the development version, contributions are welcomed.

GPareto (R library)

- BO methods (EHI, EMI, SMS, SUR)

Code

Some multiobjective libraries

README

GPareto

GPareto: Gaussian Processes for Pareto Front Estimation and Optimization

This R package provides tools for multi-objective (…) black-box functions along with estimation of Paret(…)

Installation

For the stable version: http://cran.r-project.org/package=GPareto

This the development version, contributions are welcomed.

GPareto (R library)

- BO methods (EHI, EMI, SMS, SUR)

README Code of conduct MIT license

BoTorch

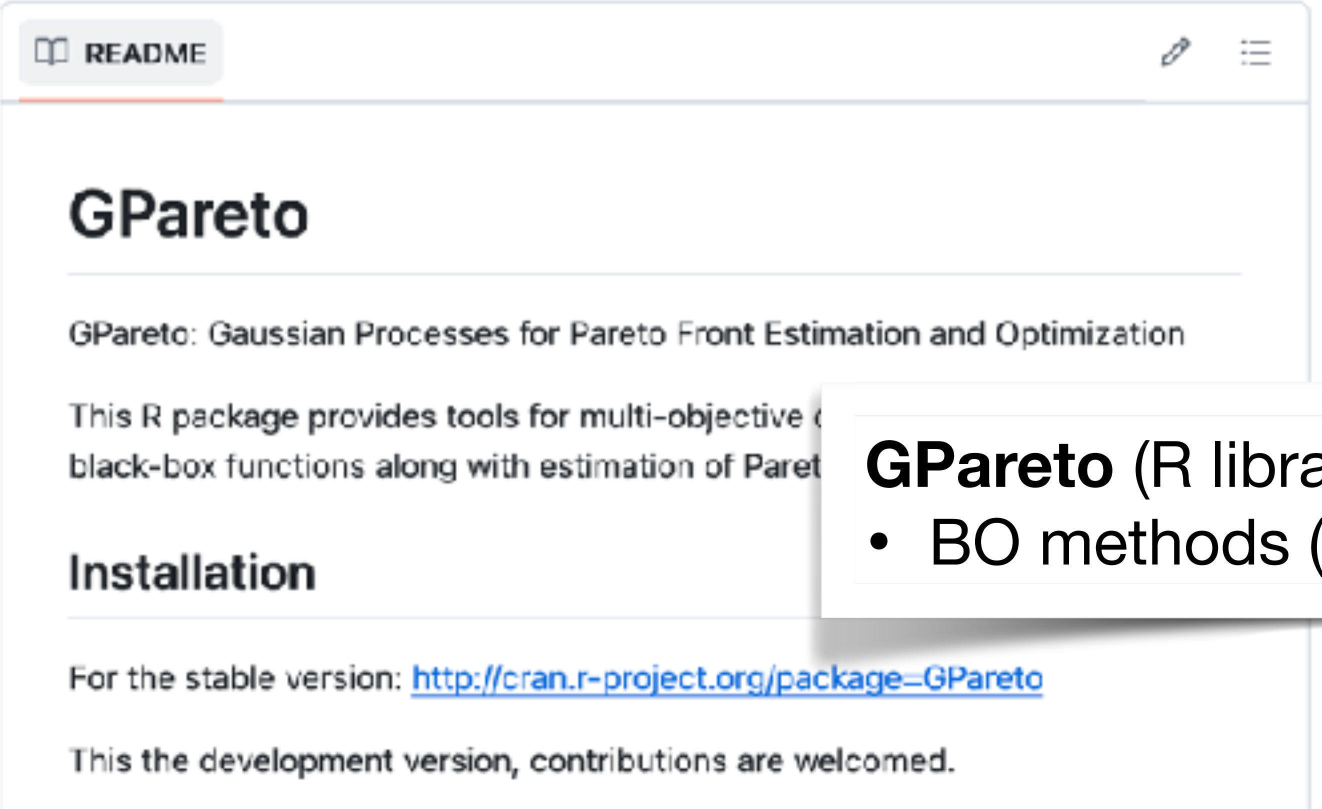
Support Ukraine | Lint passing | Test passing | Docs passing | Tutorials no status | coverage 100%

conda | pytorch v0.11.1 | pypi v0.11.1 | license MIT

BoTorch is a library for Bayesian Optimization built on PyTorch.

Code

Some multiobjective libraries



README

GPareto

GPareto: Gaussian Processes for Pareto Front Estimation and Optimization

This R package provides tools for multi-objective
black-box functions along with estimation of Paret

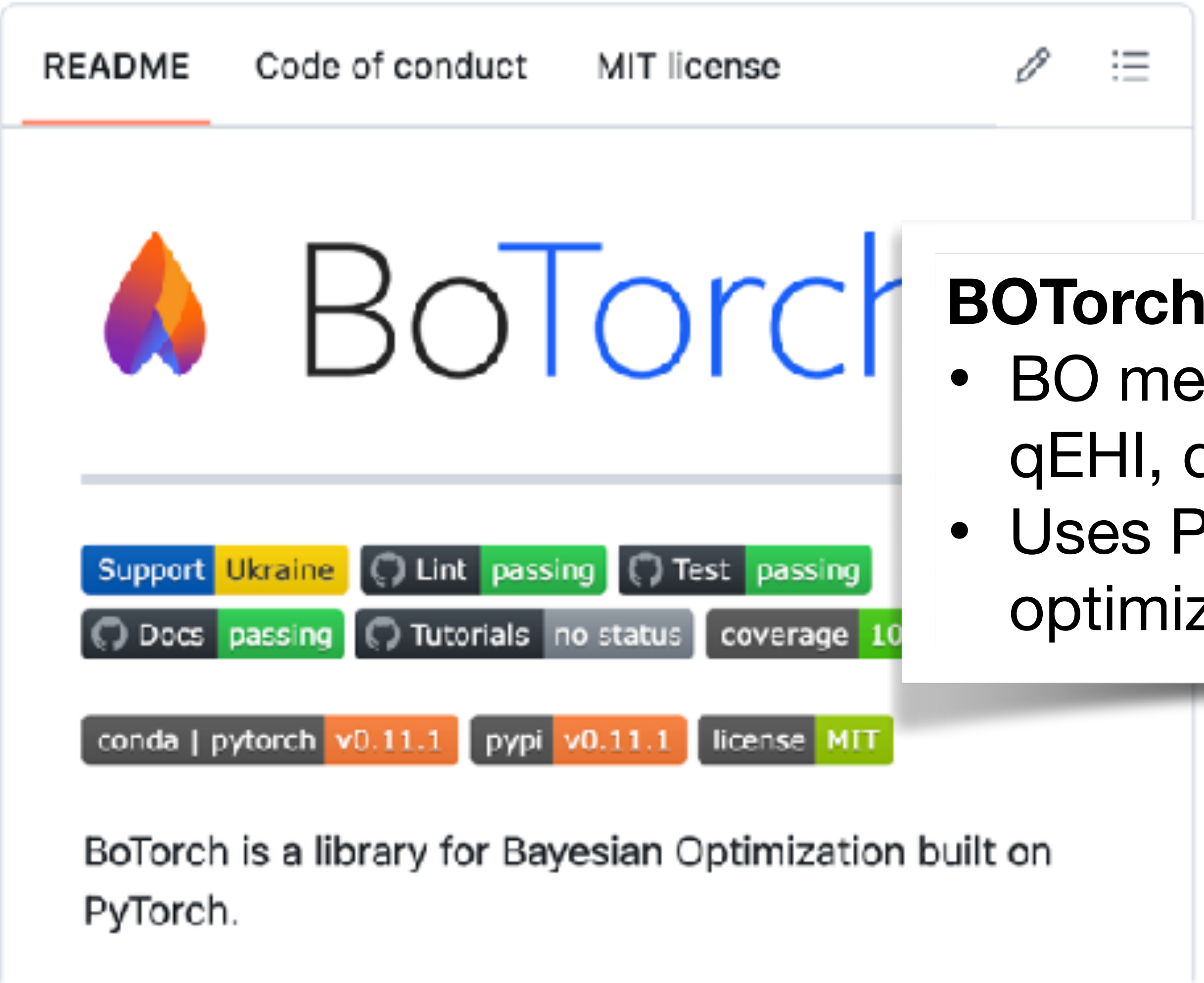
Installation

For the stable version: http://cran.r-project.org/package=GPareto

This the development version, contributions are welcomed.

GPareto (R library)

- BO methods (EHI, EMI, SMS, SUR)



README Code of conduct MIT license

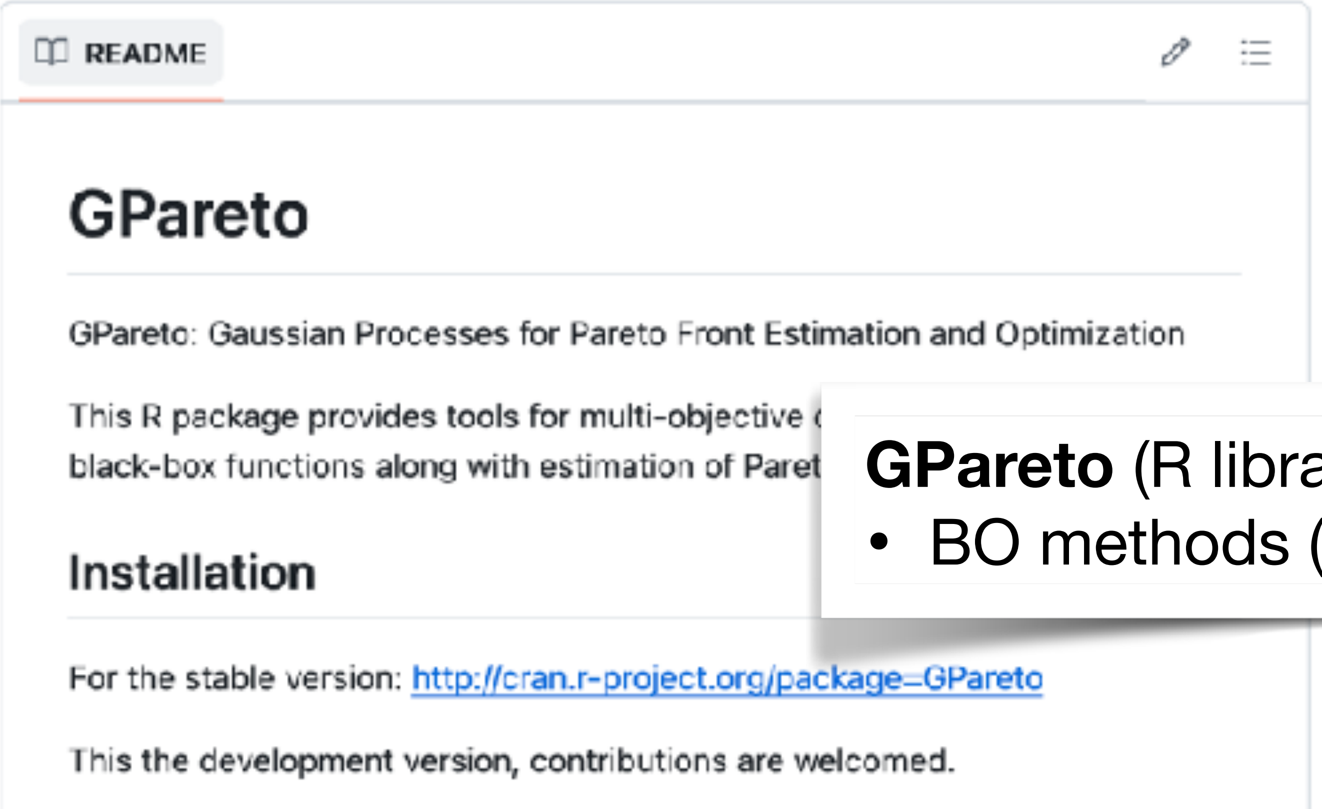
BoTorch is a library for Bayesian Optimization built on PyTorch.

BOTorch (Python)

- BO methods (EHI, qEHVI, qParEGO, qEHI, qNEHVI, …)
- Uses PyTorch to model the GP and optimize GP hyperparameters

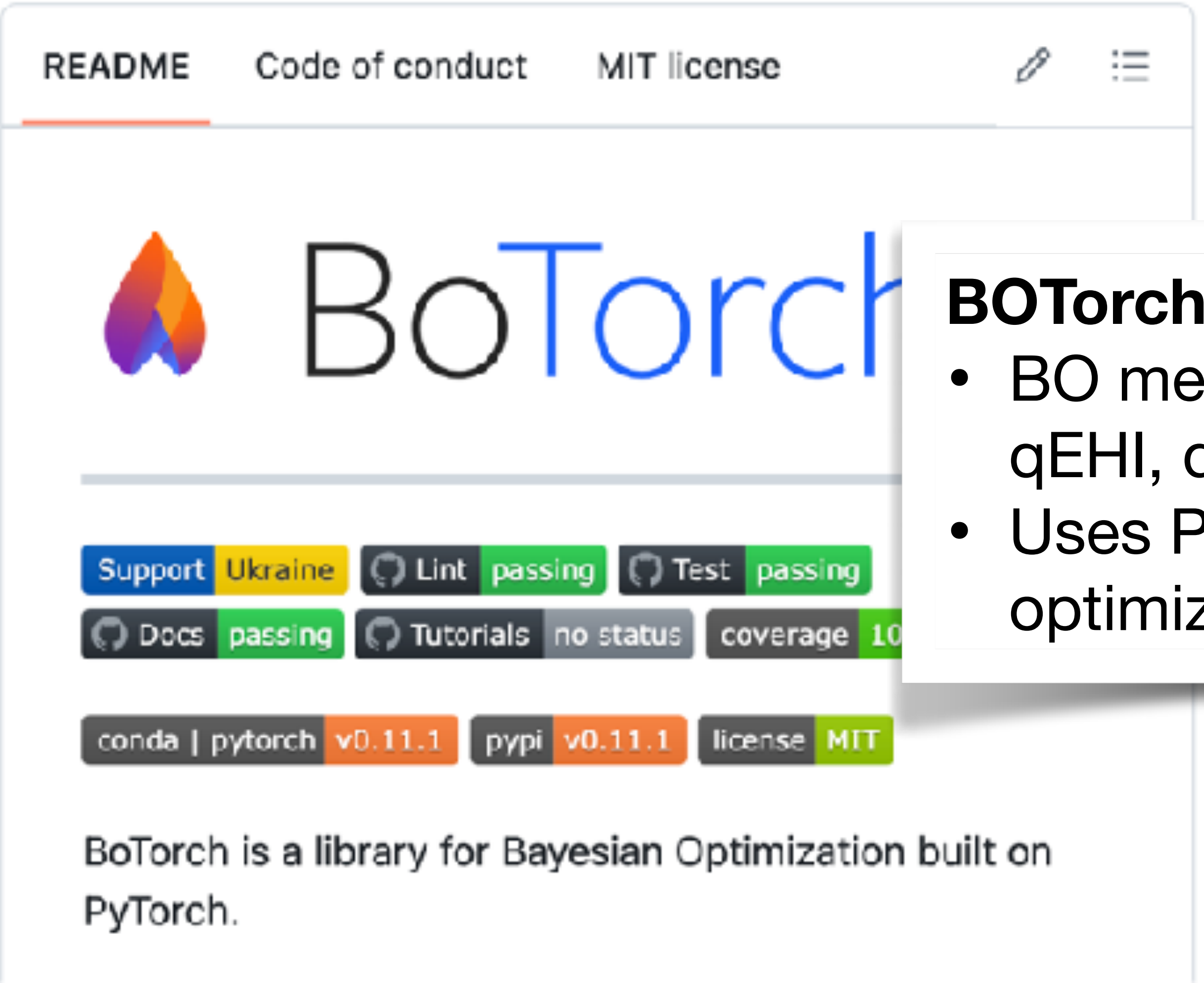
Code

Some multiobjective libraries



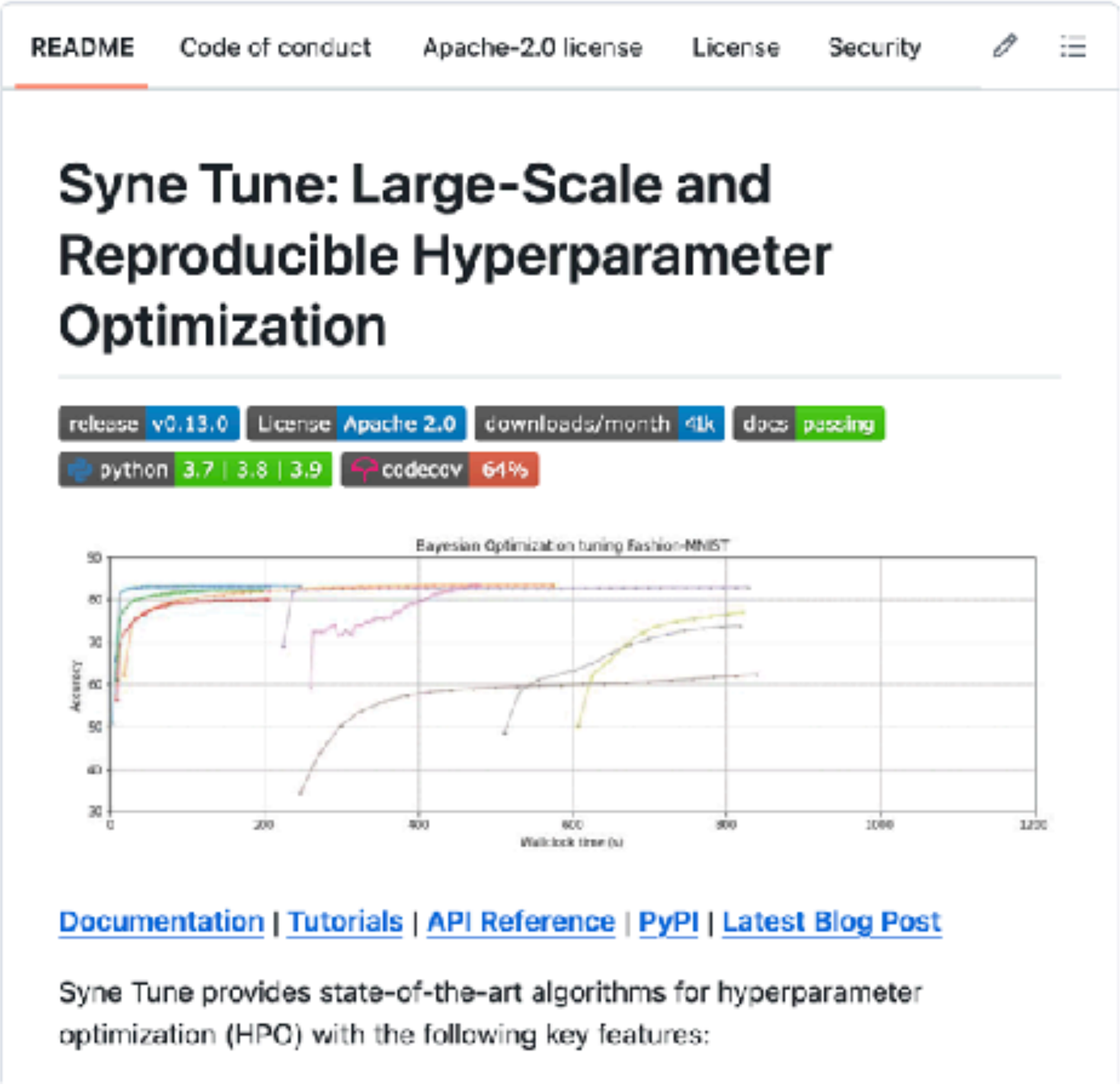
GPareto (R library)

- BO methods (EHI, EMI, SMS, SUR)



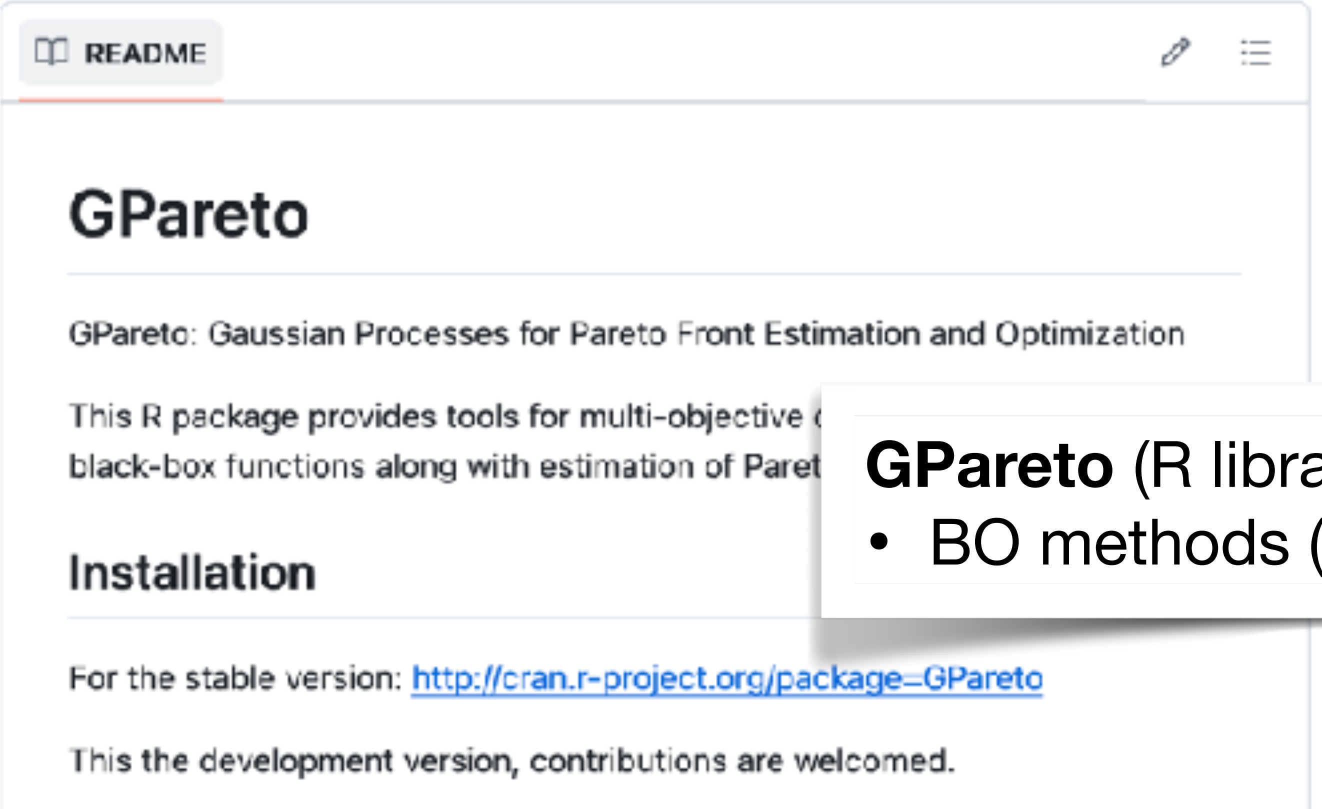
BOTorch (Python)

- BO methods (EHI, qEHVI, qParEGO, qEHI, qNEHVI, …)
- Uses PyTorch to model the GP and optimize GP hyperparameters

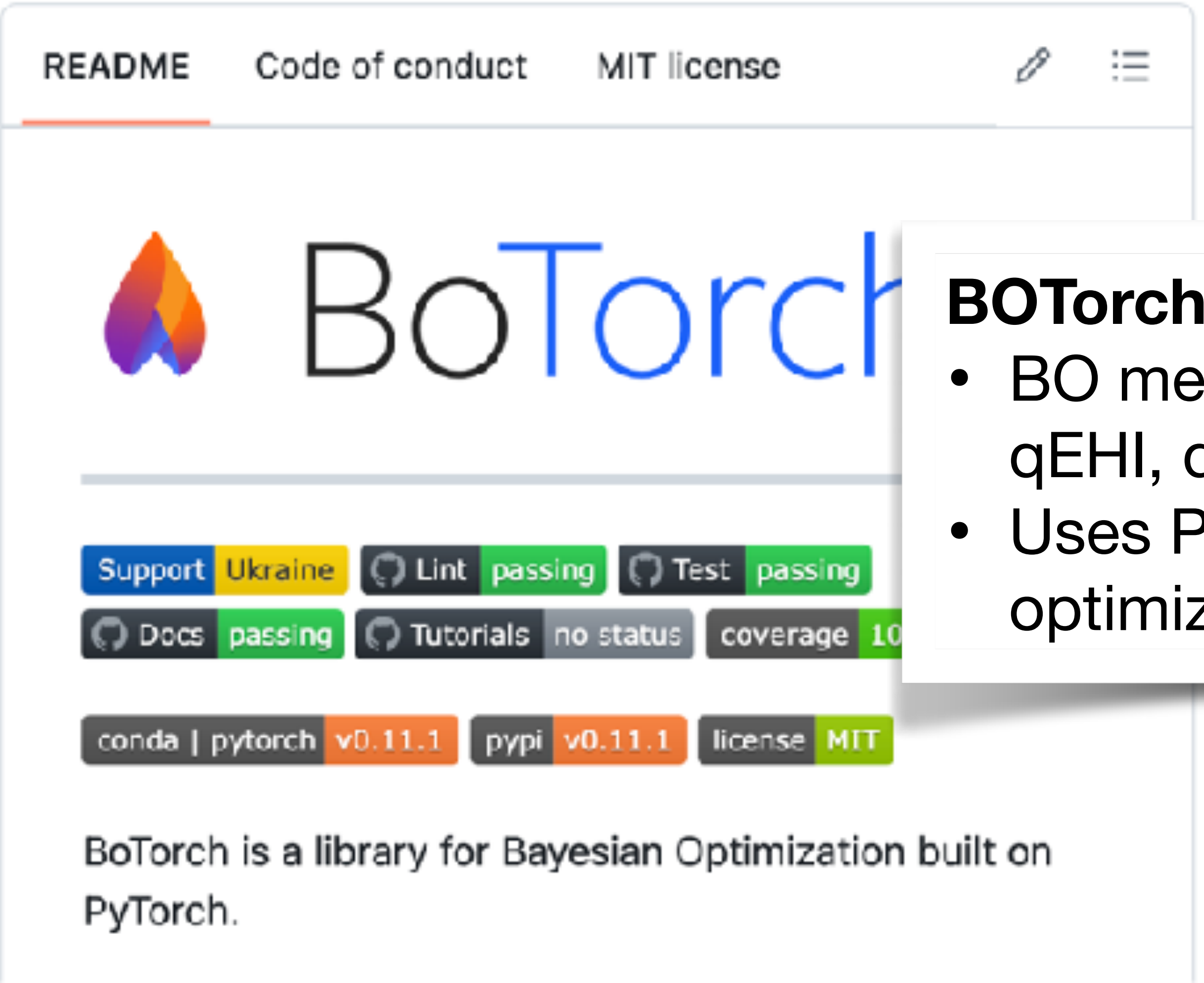


Code

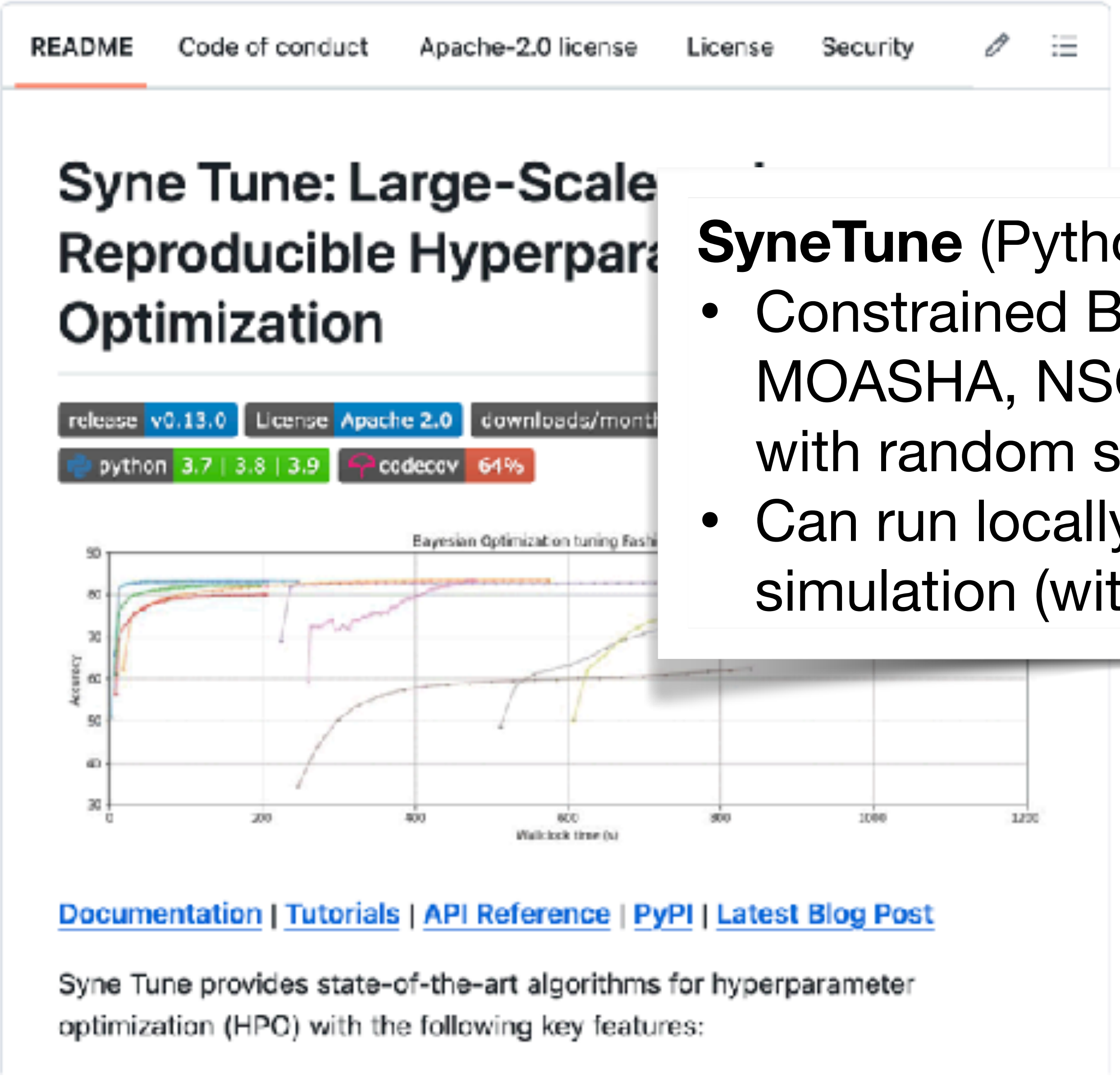
Some multiobjective libraries



GPareto (R library)
- BO methods (EHI, EMI, SMS, SUR)



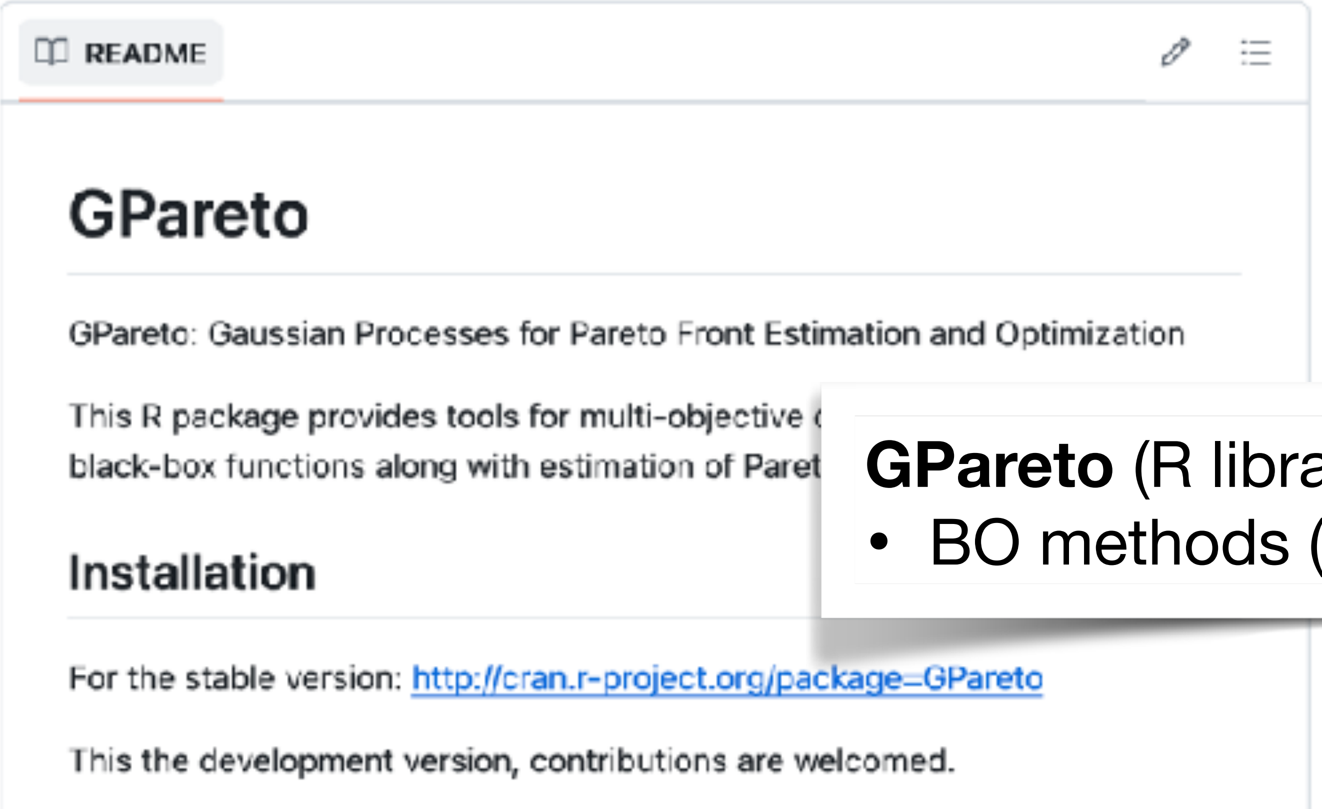
BOTorch (Python)
- BO methods (EHI, qEHVI, qParEGO, qEHI, qNEHVI, …)
- Uses PyTorch to model the GP and optimize GP hyperparameters



SyneTune (Python)
- Constrained Bayesian Optimization, MOASHA, NSGA-II, MSMOS, MSMOS with random scalarization
- Can run locally, on the cloud or with simulation (with precomputed results)

Code

Some multiobjective libraries


README

GPareto

GPareto: Gaussian Processes for Pareto Front Estimation and Optimization

This R package provides tools for multi-objective ... black-box functions along with estimation of Paret...

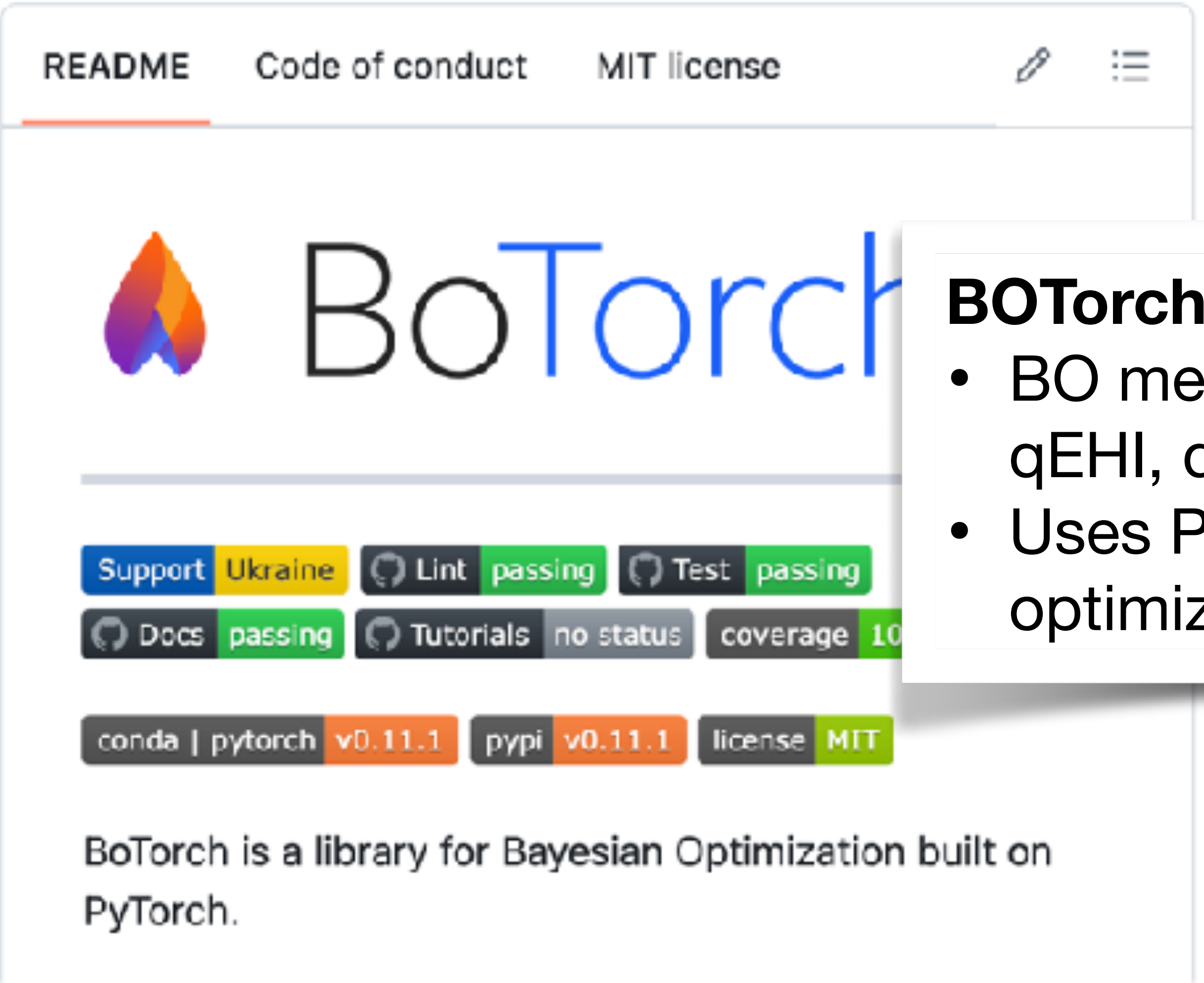
Installation

For the stable version: http://cran.r-project.org/package=GPareto

This the development version, contributions are welcomed.

GPareto (R library)

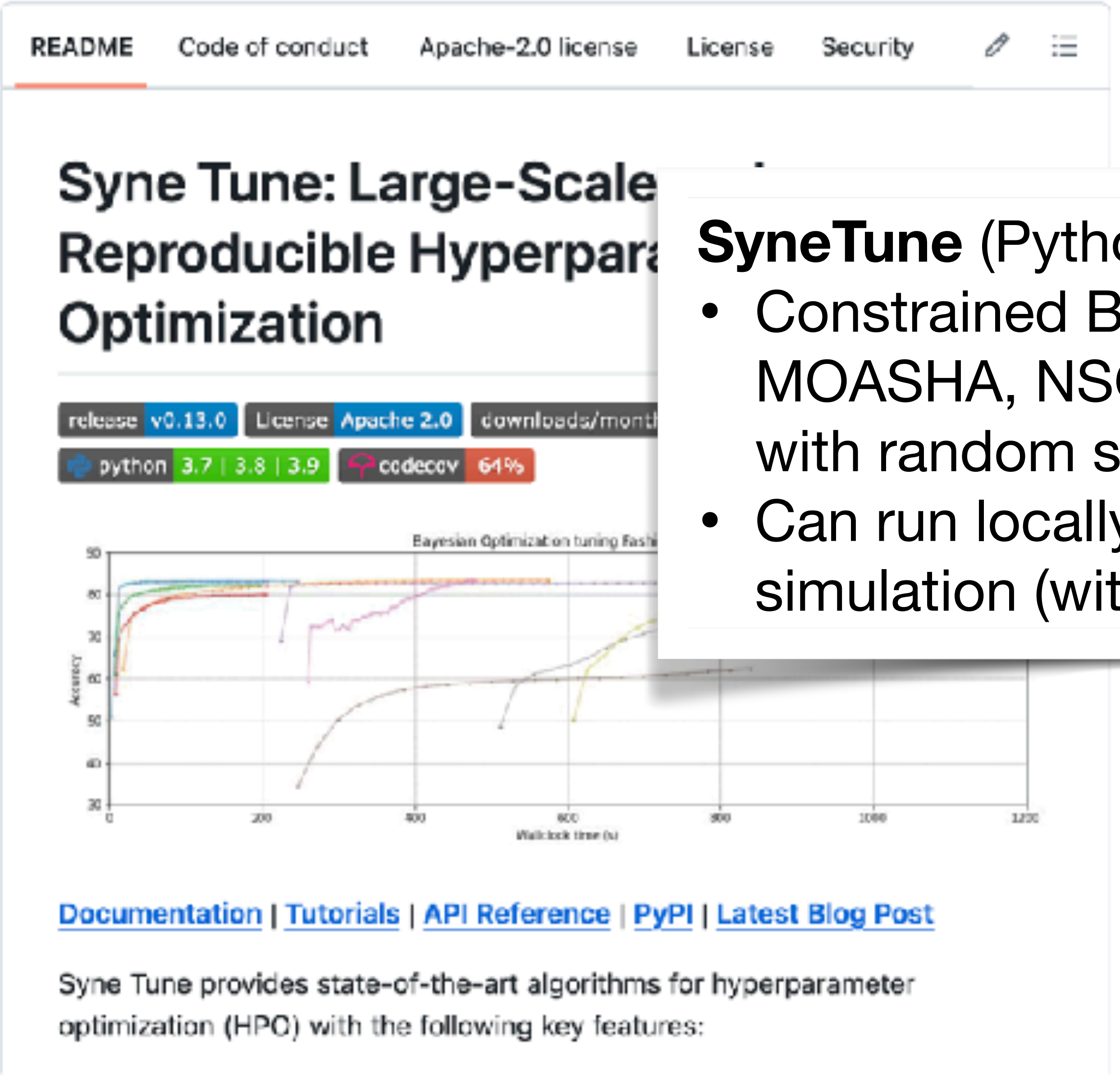
- BO methods (EHI, EMI, SMS, SUR)


README Code of conduct MIT license

BoTorch is a library for Bayesian Optimization built on PyTorch.

BOTorch (Python)

- BO methods (EHI, qEHVI, qParEGO, qEHI, qNEHVI, …)
- Uses PyTorch to model the GP and optimize GP hyperparameters


README Code of conduct Apache-2.0 license License Security

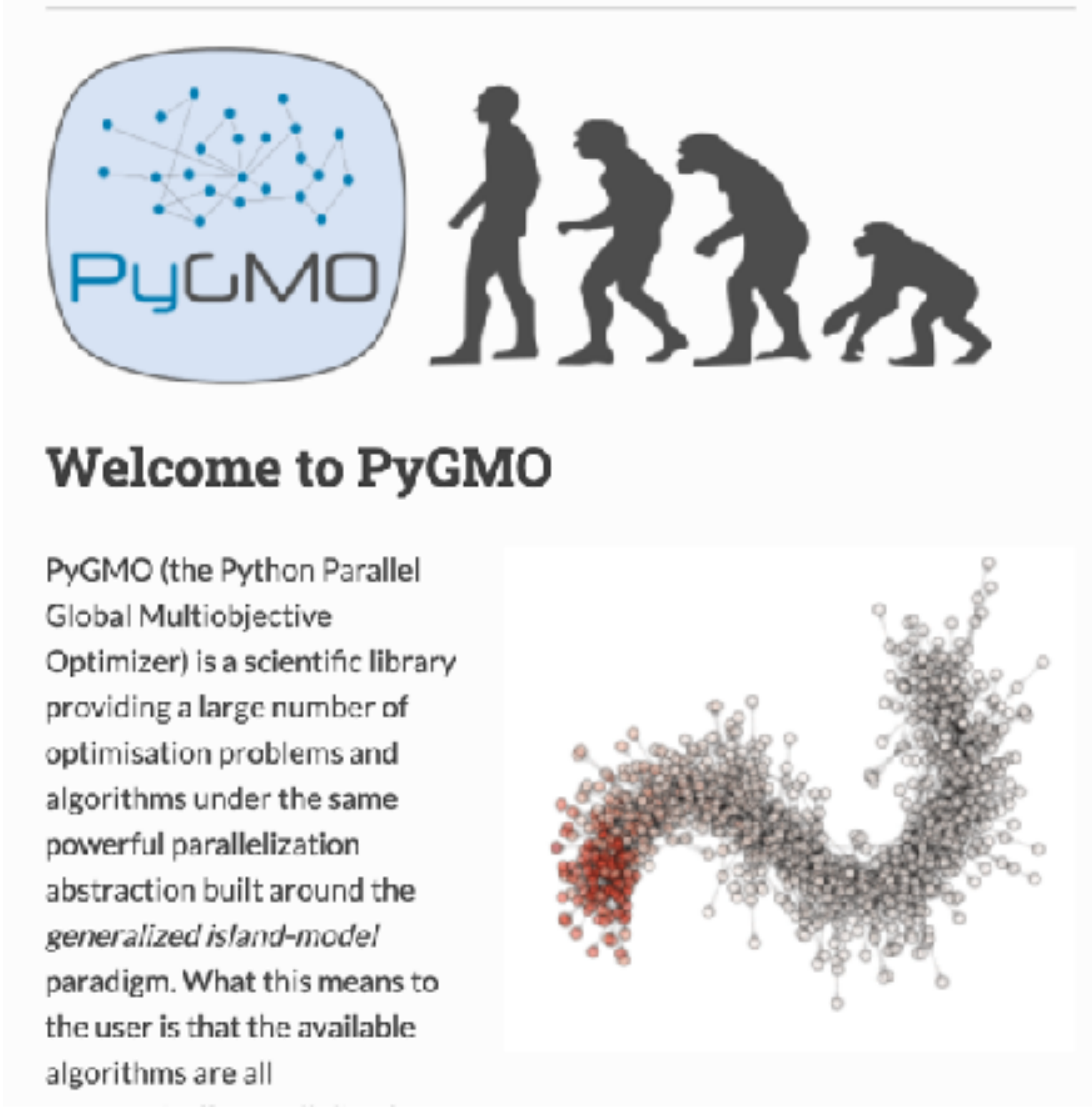
Syne Tune: Large-Scale ... Reproducible Hyperpara... Optimization

Documentation | Tutorials | API Reference | PyPI | Latest Blog Post

Syne Tune provides state-of-the-art algorithms for hyperparameter optimization (HPO) with the following key features:

SyneTune (Python)

- Constrained Bayesian Optimization, MOASHA, NSGA-II, MSMOS, MSMOS with random scalarization
- Can run locally, on the cloud or with simulation (with precomputed results)

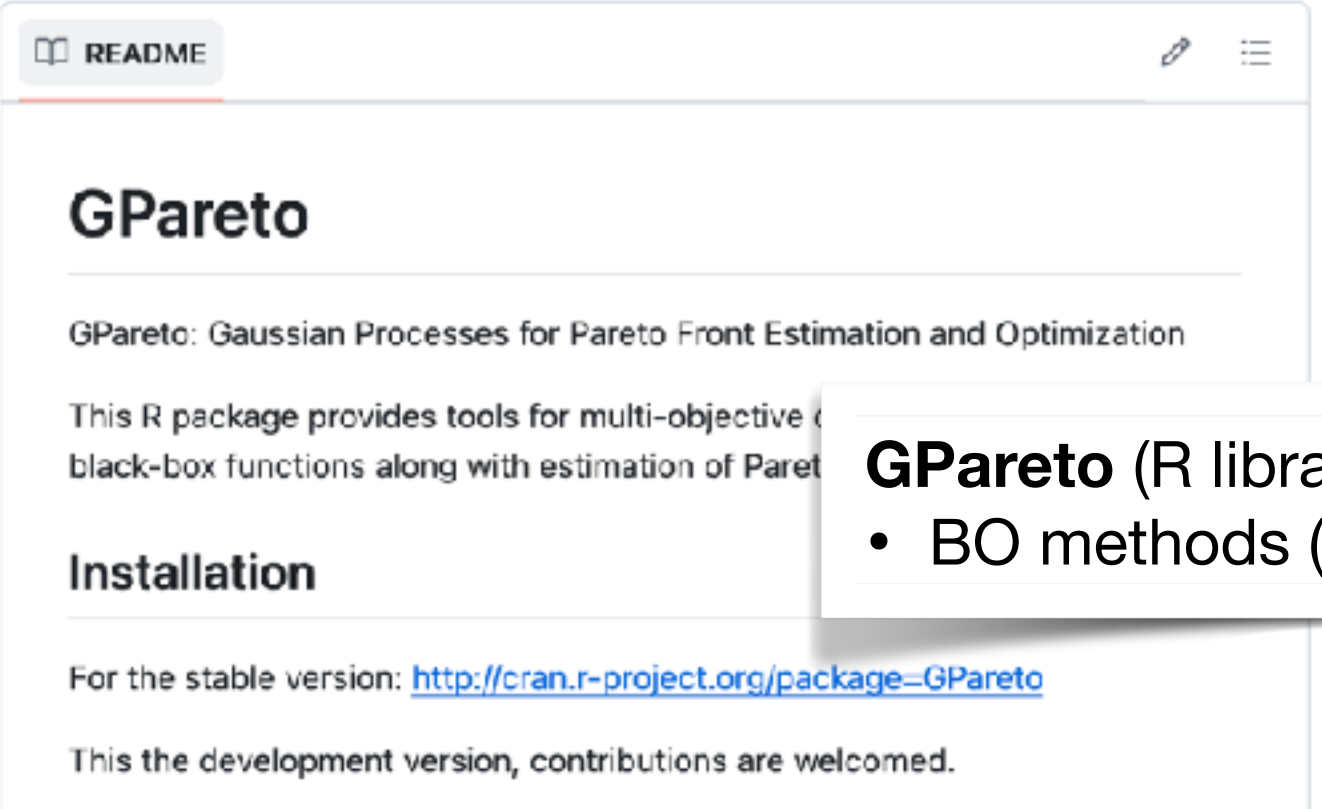


Welcome to PyGMO

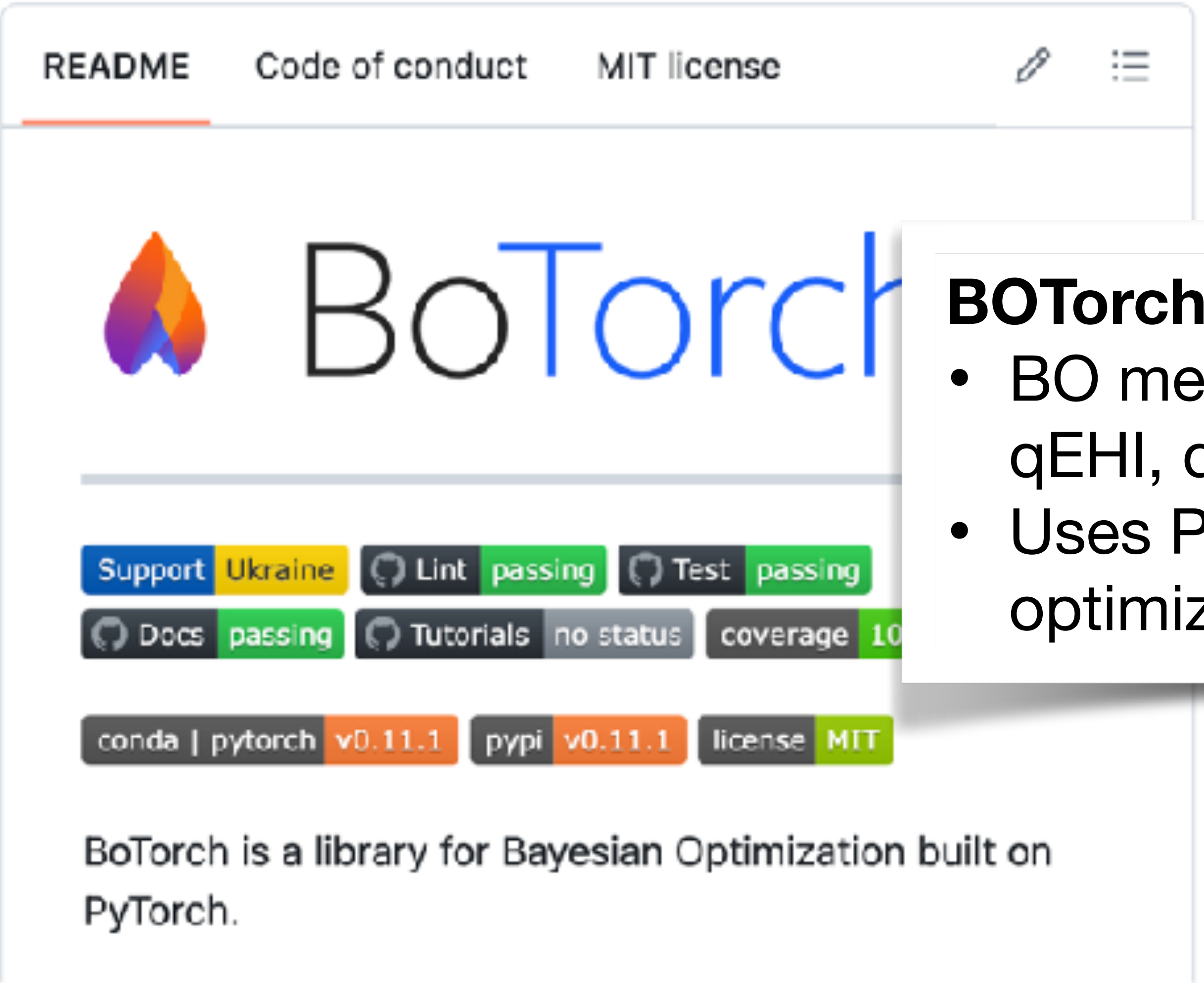
PyGMO (the Python Parallel Global Multiobjective Optimizer) is a scientific library providing a large number of optimisation problems and algorithms under the same powerful parallelization abstraction built around the *generalized island-model* paradigm. What this means to the user is that the available algorithms are all

Code

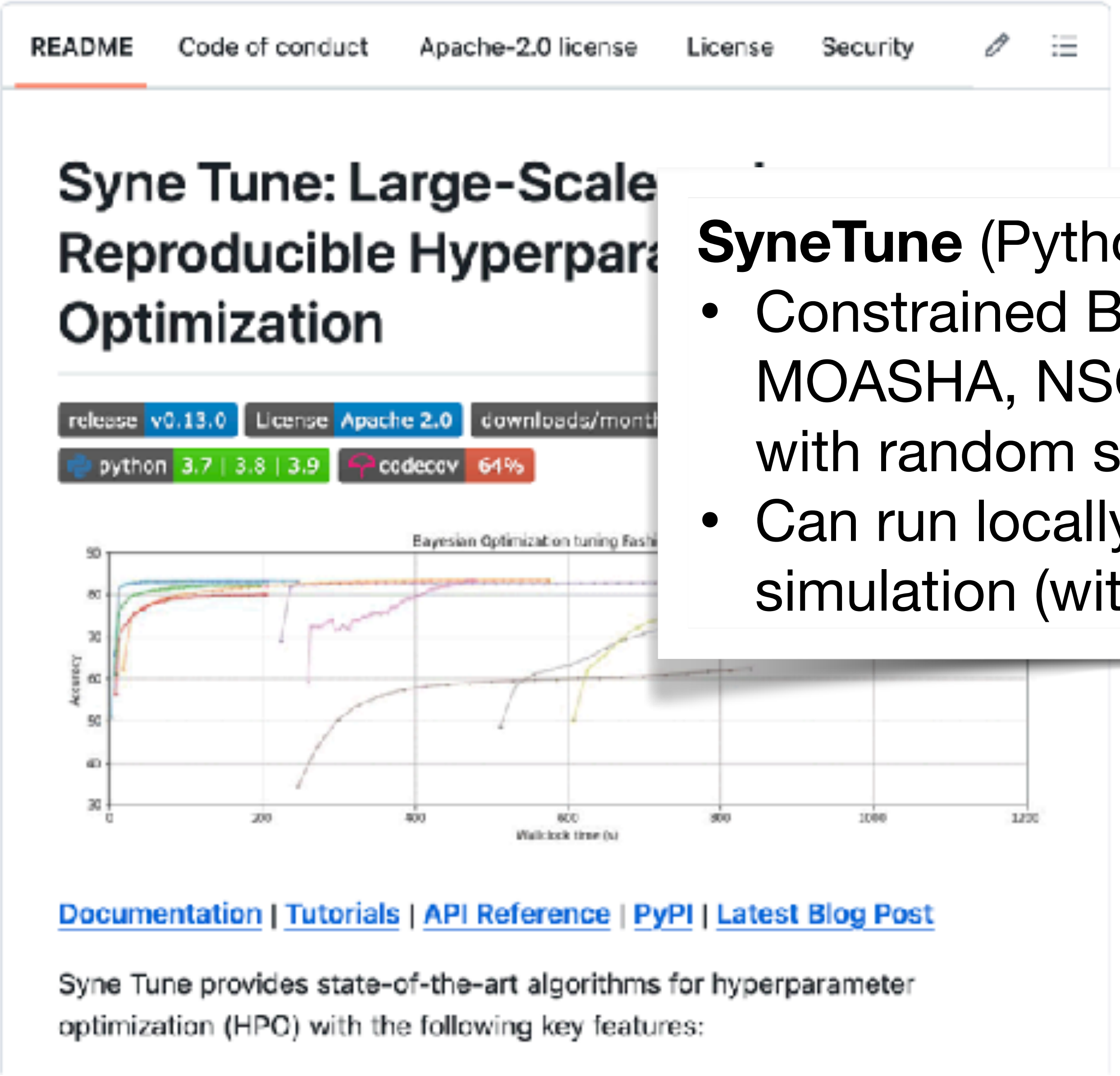
Some multiobjective libraries



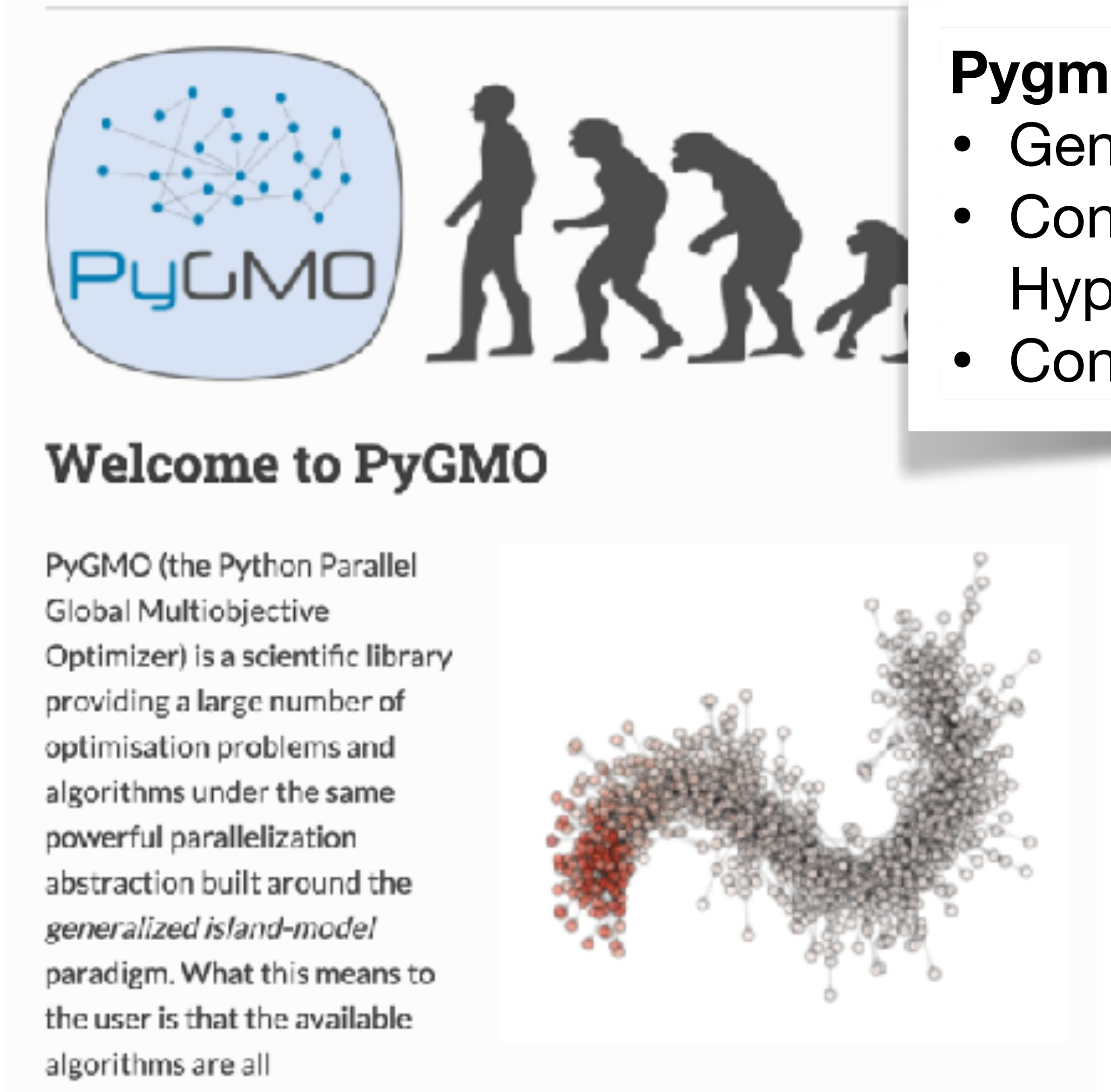
GPareto (R library)
- BO methods (EHI, EMI, SMS, SUR)



BOTorch (Python)
- BO methods (EHI, qEHVI, qParEGO, qEHI, qNEHVI, …)
- Uses PyTorch to model the GP and optimize GP hyperparameters



SyneTune (Python)
- Constrained Bayesian Optimization, MOASHA, NSGA-II, MSMOS, MSMOS with random scalarization
- Can run locally, on the cloud or with simulation (with precomputed results)



Pygmo (Python/c++)
- General python library
- Contains optimizers but also utilities to compute Hypervolume, Hypervolume contribution, …
- Contains approximation algorithms

Code

How to tune multiple objectives in Syne Tune

Step 1: report multiple objectives in a training script

```
if __name__ == "__main__":
    # plot_function()
    parser = ArgumentParser()
    parser.add_argument("--steps", type=int, required=True)
    parser.add_argument("--theta", type=float, required=True)
    parser.add_argument("--sleep_time", type=float, required=False, default=0.1)
    args, _ = parser.parse_known_args()

    assert 0 <= args.theta < np.pi / 2
    reporter = Reporter()
    for step in range(args.steps):
        y = f(t=step, theta=args.theta)
        reporter(step=step, **y)
```

https://github.com/syne-tune/syne-tune/blob/main/examples/launch_height_moasha.py

Code

How to tune multiple objectives in Syne Tune

Step 1: report multiple objectives in a training script

```
if __name__ == "__main__":
    # plot_function()
    parser = ArgumentParser()
    parser.add_argument("--steps", type=int, required=True)
    parser.add_argument("--theta", type=float, required=True)
    parser.add_argument("--sleep_time", type=float, required=False, default=0.1)
    args, _ = parser.parse_known_args()

    assert 0 <= args.theta < np.pi / 2
    reporter = Reporter()
    for step in range(args.steps):
        y = f(t=step, theta=args.theta)
        reporter(step=step, **y)
```

Step 2: call a multiobjective optimizer to tune the training script

```
entry_point = (
    Path(__file__).parent
    / "training_scripts"
    / "mo_artificial"
    / "mo_artificial.py"
)
mode = "min"

np.random.seed(0)
scheduler = MOASHA(
    max_t=max_steps,
    time_attr="step",
    mode=mode,
    metrics=["y1", "y2"],
    config_space=config_space,
)
trial_backend = LocalBackend(entry_point=str(entry_point))

stop_criterion = StoppingCriterion(max_wallclock_time=20)
tuner = Tuner(
    trial_backend=trial_backend,
    scheduler=scheduler,
    stop_criterion=stop_criterion,
    n_workers=n_workers,
    sleep_time=0.5,
)
tuner.run()
```

https://github.com/syne-tune/syne-tune/blob/main/examples/launch_height_moasha.py

Conclusion

Conclusion

- Multiobjective optimization allows to optimise multiple objectives at the same time

Conclusion

- Multiobjective optimization allows to optimise multiple objectives at the same time
- Generally not a single solution but a set of optimal solutions, the Pareto front

Conclusion

- Multiobjective optimization allows to optimise multiple objectives at the same time
- Generally not a single solution but a set of optimal solutions, the Pareto front
- Key methods:

Conclusion

- Multiobjective optimization allows to optimise multiple objectives at the same time
- Generally not a single solution but a set of optimal solutions, the Pareto front
- Key methods:
    - Scalarization

Conclusion

- Multiobjective optimization allows to optimise multiple objectives at the same time
- Generally not a single solution but a set of optimal solutions, the Pareto front
- Key methods:
  - Scalarization
  - Bayesian Optimization

Conclusion

- Multiobjective optimization allows to optimise multiple objectives at the same time
- Generally not a single solution but a set of optimal solutions, the Pareto front
- Key methods:
    - Scalarization
    - Bayesian Optimization
    - Evolution Algorithms

Conclusion

- Multiobjective optimization allows to optimise multiple objectives at the same time
- Generally not a single solution but a set of optimal solutions, the Pareto front
- Key methods:
  - Scalarization
  - Bayesian Optimization
  - Evolution Algorithms
- Many applications!

Conclusion

- Multiobjective optimization allows to optimise multiple objectives at the same time
- Generally not a single solution but a set of optimal solutions, the Pareto front
- Key methods:
    - Scalarization
    - Bayesian Optimization
    - Evolution Algorithms
- Many applications!
- Active area of research