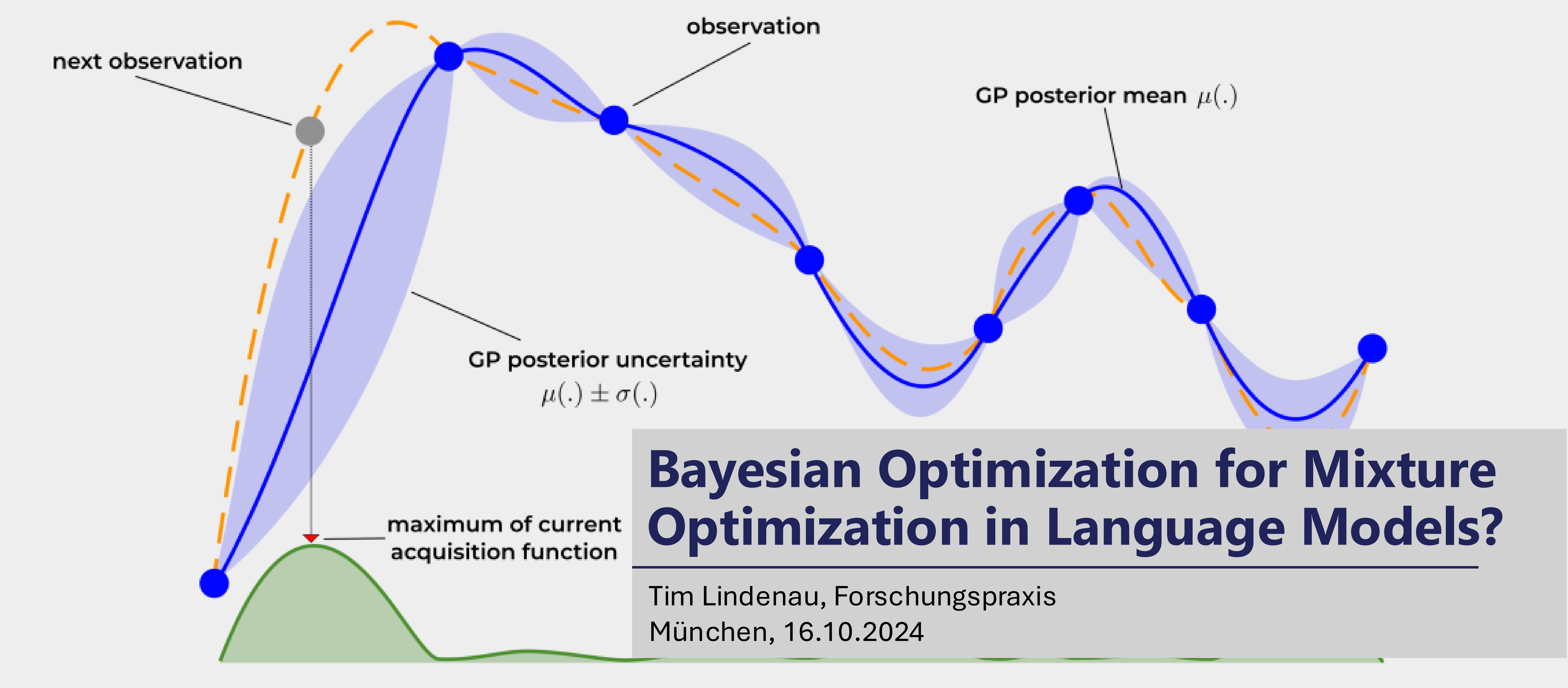


Bayesian Optimization for Mixture Optimization in Language Models?

Tim Lindenau, Forschungspraxis

München, 16.10.2024

A

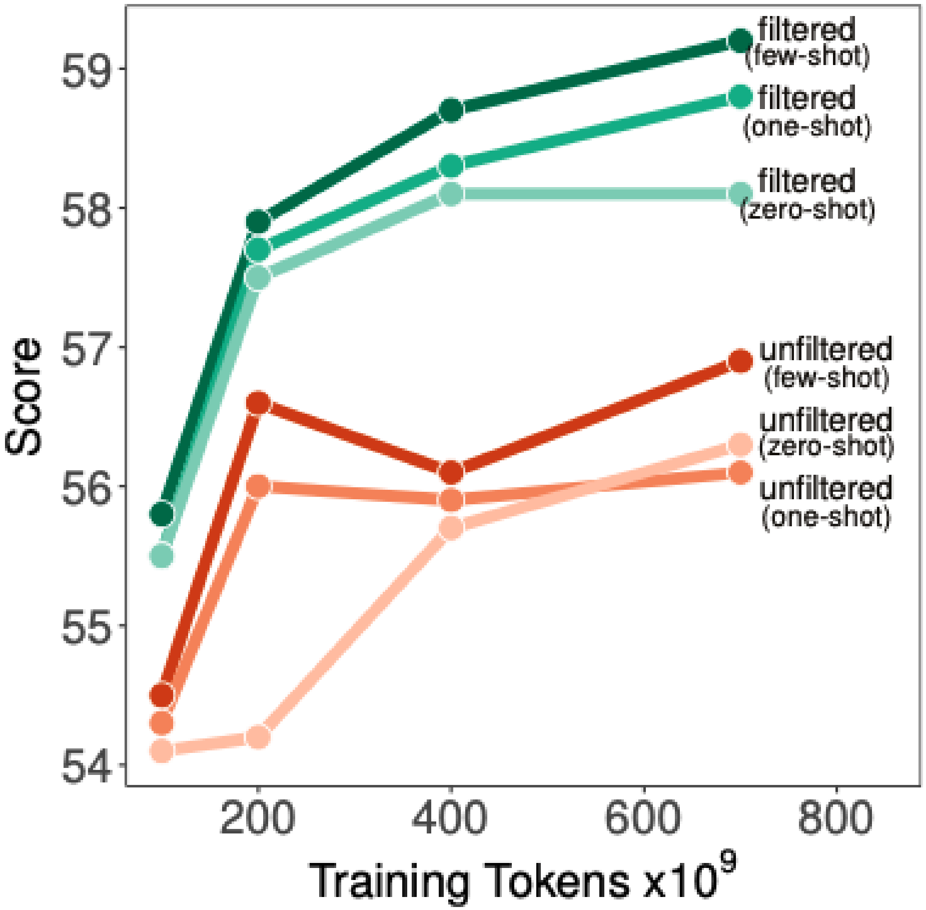
Motivation

Motivation: The choice of data is crucial for LMs

- Massive **scaling of training data** has been key factor towards the current **LLM progress**
- **Unequal utility** of data. Even **simple filtering** can already achieve **big performance improvements**
- **Existing data curation** methods unprincipled, mostly around **heuristics** and **human intuition** about data utility

➡ **Principled data-curation** required to unlock **faster training** and **better** model **performance**

Data filtering improves LM performance



Du et al 2021

Our Setting: Data curation by mixing different domains

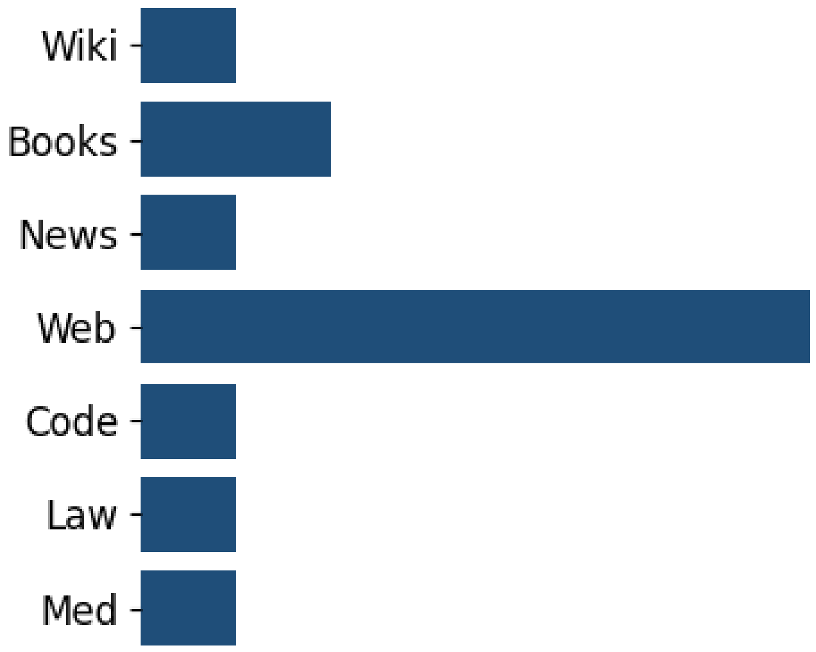
Problem Setup – Data Mixing:

- Training data from k **domains** (Wikipedia, Web, Code, etc.)
- For a **fixed budget**, how much of **each domain** to train on

Biggest Challenge – Runtime Overhead:

- **No** significant **compute overhead** in pre-training
- Mixture optimization on **small proxy model**, assuming scale invariance
- Focus on **sample efficiency** in training number of proxy models

Mixing proportions of different datasets for pre-training



Du et al 2021

Existing Mixture Optimization Approaches and their Pros & Cons

Name	Method	Pros & Cons
Online Mixing	• **Data Mixing while training** the production grade model • **Multi-Armed Bandit** to model which domain to sample **next token** • Domain **perplexity as reward function**	+ **Negligible run-time** overhead **without pre-training** proxy models - **Missing interpretability** into overall optimal mixture
DoReMi	• **Robust optimization** minimizing **worst case performance** • **Reference model** is pre-trained on some mixture • **Optimal mixture** to **minimize per-domain excess loss** compared to reference	+ **Only two proxy** models required + **Optimization criterion** promises good **generalizability** - Identified **mixture very sensitive** to model **architecture**
Mixing Laws	• Formulate **parametrized law** to model **performance** as **function of mixture weights** • **Law-fitting** based on **limited number of evaluations** • **Optimal mixture** predicted as **minimizer** of law	+ **Complexity** reduction of search space bounded too law - **Risk of modelling errors** in law, (example: BiMix not modelling any domain interactions)

Existing Mixture Optimization Approaches and their Pros & Cons

Name	Method	Pros & Cons
Online Mixing	• **Data Mixing while training** the production grade model • **Multi-Armed Bandit** to model which domain to sample **next token** • Domain **perplexity as reward function**	+ **Negligible run-time** overhead **without pre-training** proxy models - **Missing interpretability** into overall optimal mixture
DoReMi	• **Robust optimization** minimizing **worst case performance** • **Reference model** is pre-trained on some mixture • **Optimal mixture** to **minimize per-domain excess loss** compared to reference	+ **Only two proxy** models required + **Optimization criterion** promises good **generalizability** - Identified **mixture very sensitive** to model **architecture**
Mixing Laws	• Formulate **parametrized law** to model **performance** as **function of mixture weights** • **Law-fitting** based on **limited number of evaluations** • **Optimal mixture** predicted as **minimizer** of law	+ **Complexity** reduction of search space bounded too law - **Risk of modelling errors** in law, (example: BiMix not modelling any domain interactions)

Three key challenges to tackle for **novel mixture optimization approach**

1. **Compute Efficiency:** Minimal overhead unlocked by high sample efficiency
2. **Flexibility:** Reduced dependency on pre-defined functional shapes to prevent modelling errors
3. **Insight:** Identified mixture must provide insight into the true quality/utility of each domain after training

A

Method: Bayesian Optimization for Data Mixing

BO is a Sequential Optimization Process Towards Finding Function Optimums

- All **previous mixture weights** $\boldsymbol{x_i}$ and **performance value** $\boldsymbol{y_i}$ collected in dataset

$$\boldsymbol{D_t} = \{(\boldsymbol{y_i}, \boldsymbol{x_i})\}_{\boldsymbol{i=0}}^{\boldsymbol{t}}$$

- **Gaussian Process** provides **probability distribution** of performance at each sampling point

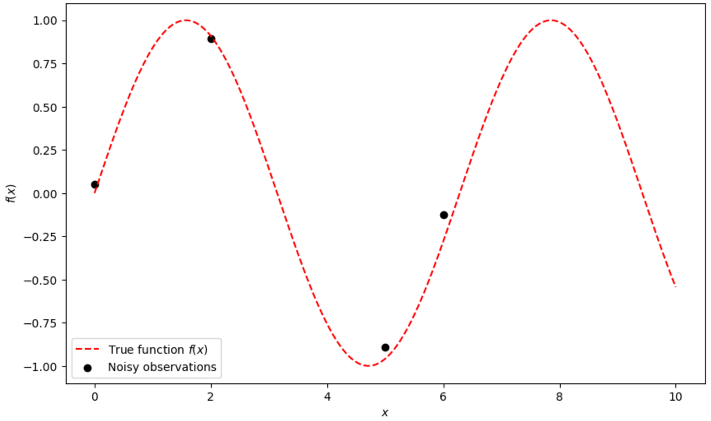
$$g(x, y \mid D) = \Pr(f(x) = y \mid D)$$

- **Acquisition function** measures utility of next-weight x_{t+1}
- **Tradeoff** between **exploitation** of previous well performing mixtures and **exploration** of unknown mixture regions
- **Maximizing acquisition** function gives **next mixture weight** in smart way

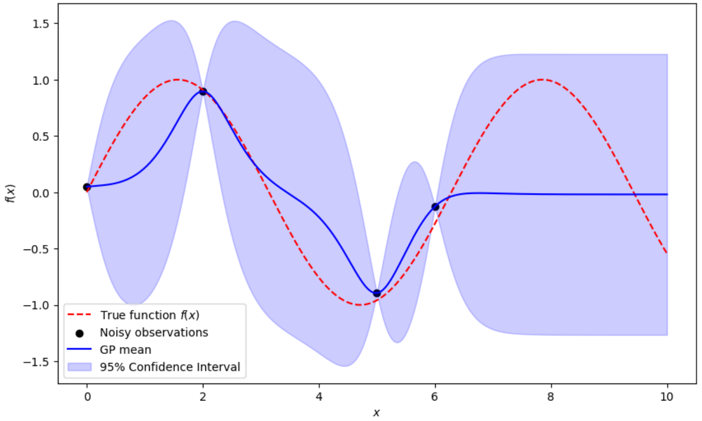
BO has potential for **high-sample efficiency,** without **systematic errors** introduced by **mixing laws**

BO Loop in Detail

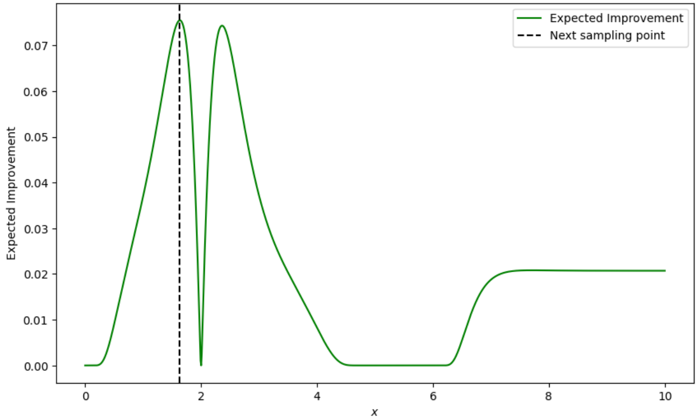
1 Existing function evaluations stored in dataset D_t



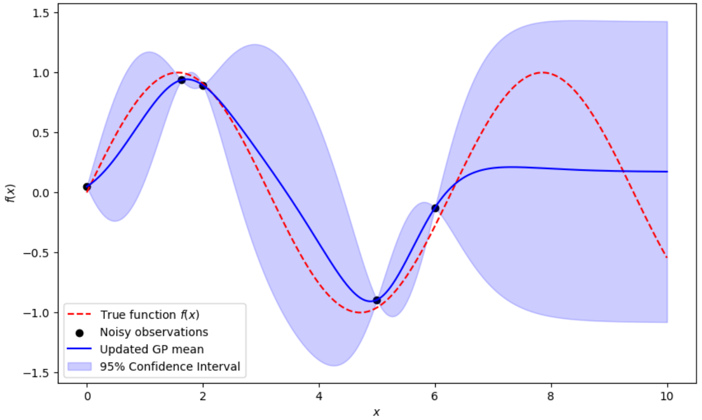
2 Gaussian Process builds model of the function of interest



3 Acquisition function measures utility of next sampling point

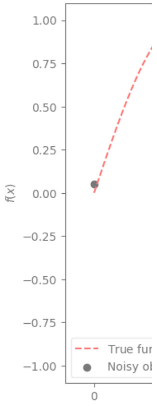


4 GP updated based on novel function evaluation

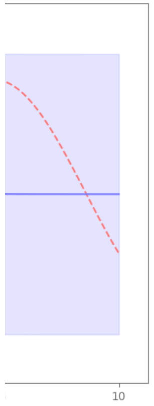


BO Loop in Detail

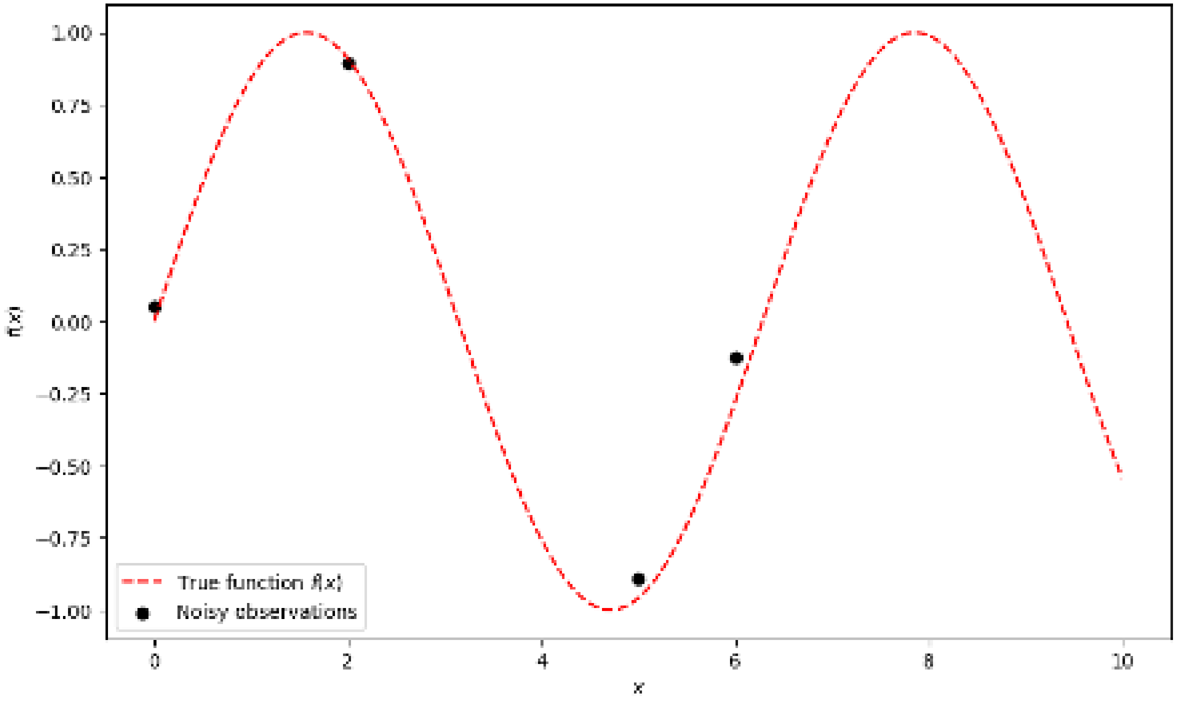
1 Existing function evaluations stored in dataset D_t



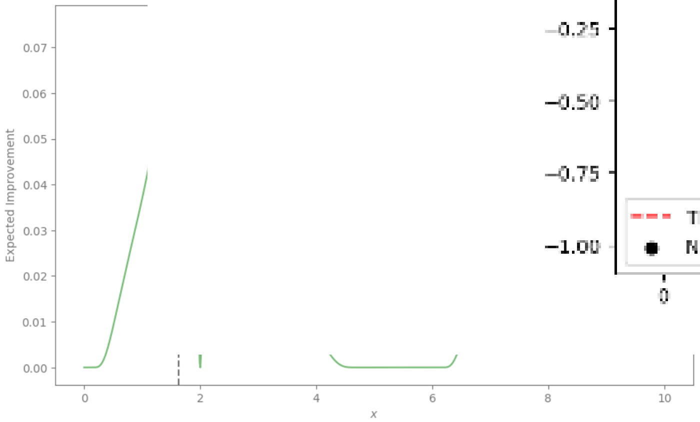
2 Gaussian Process builds model of the



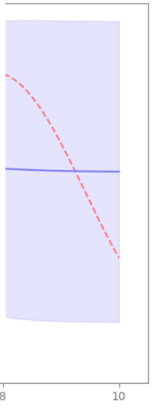
1 **Existing function evaluations stored in dataset D_t**



3 Acquisition sampling pc



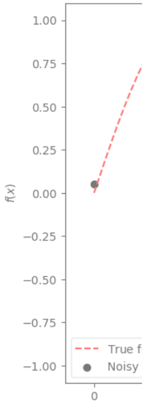
ıction

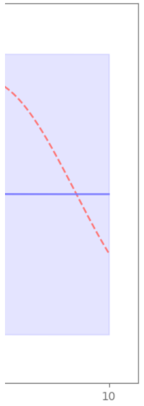


BO Loop in Detail

1 Existing function evaluations stored in dataset D_t

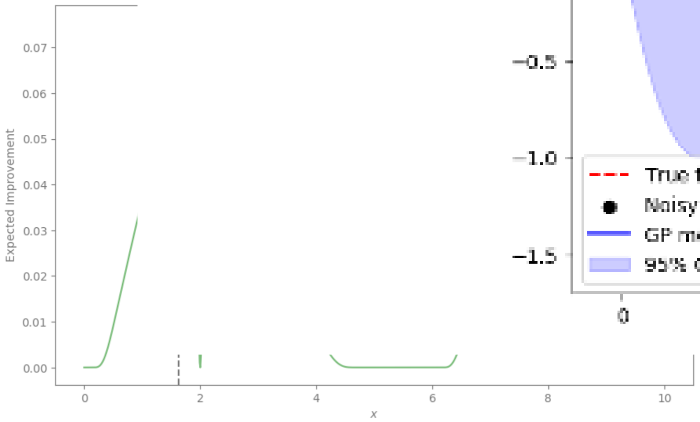
2 Gaussian Process builds model of the function if interest

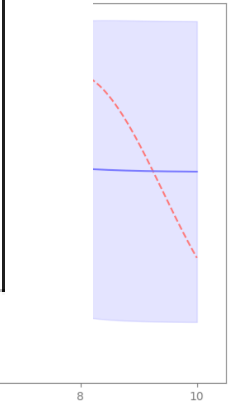




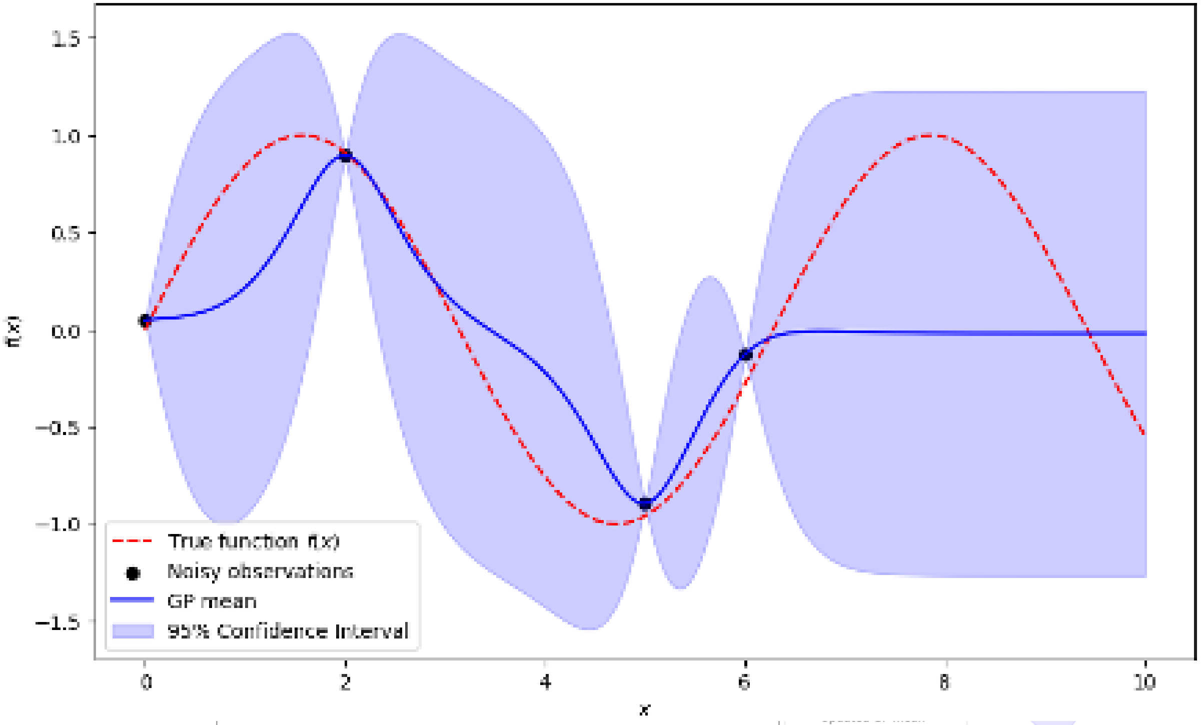
3 Acquisition sampling p

ction





2 Gaussian Process builds model of the function if interest

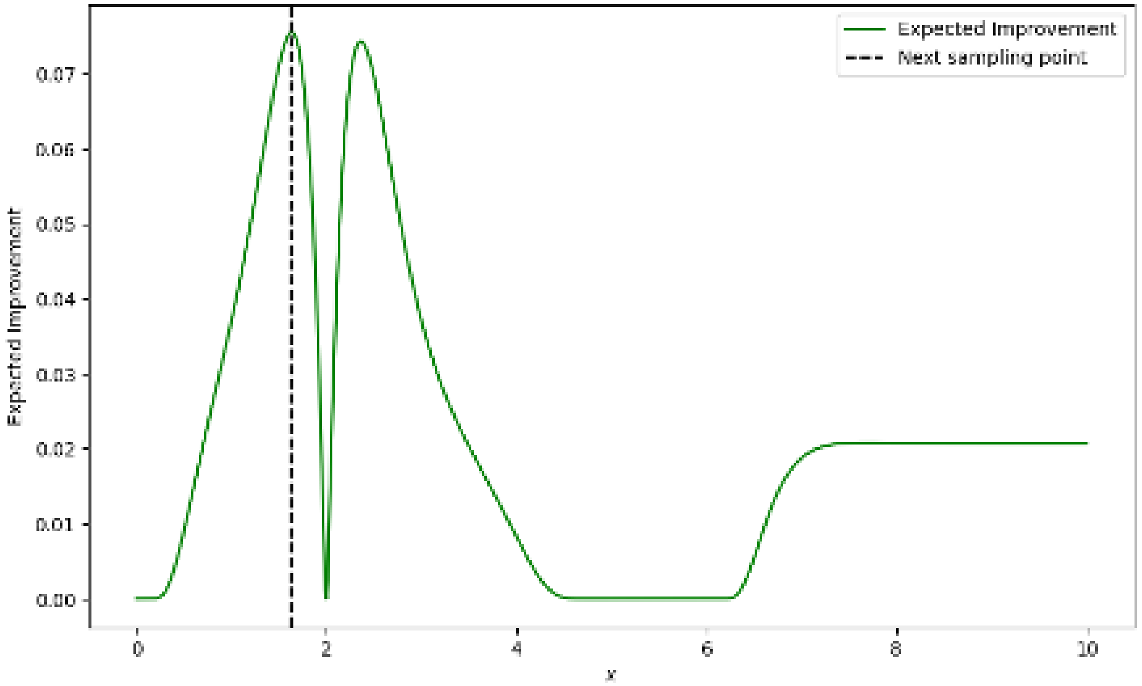


BO Loop in Detail

1. Existing function evaluations stored in dataset Γ

2. Gaussian Process builds model of the

3 Acquisition function measures utility of next sampling point

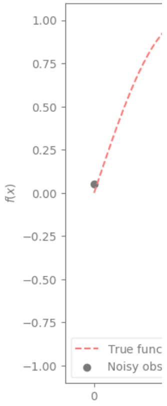


3. Acquisiti sampling

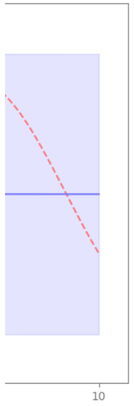
on

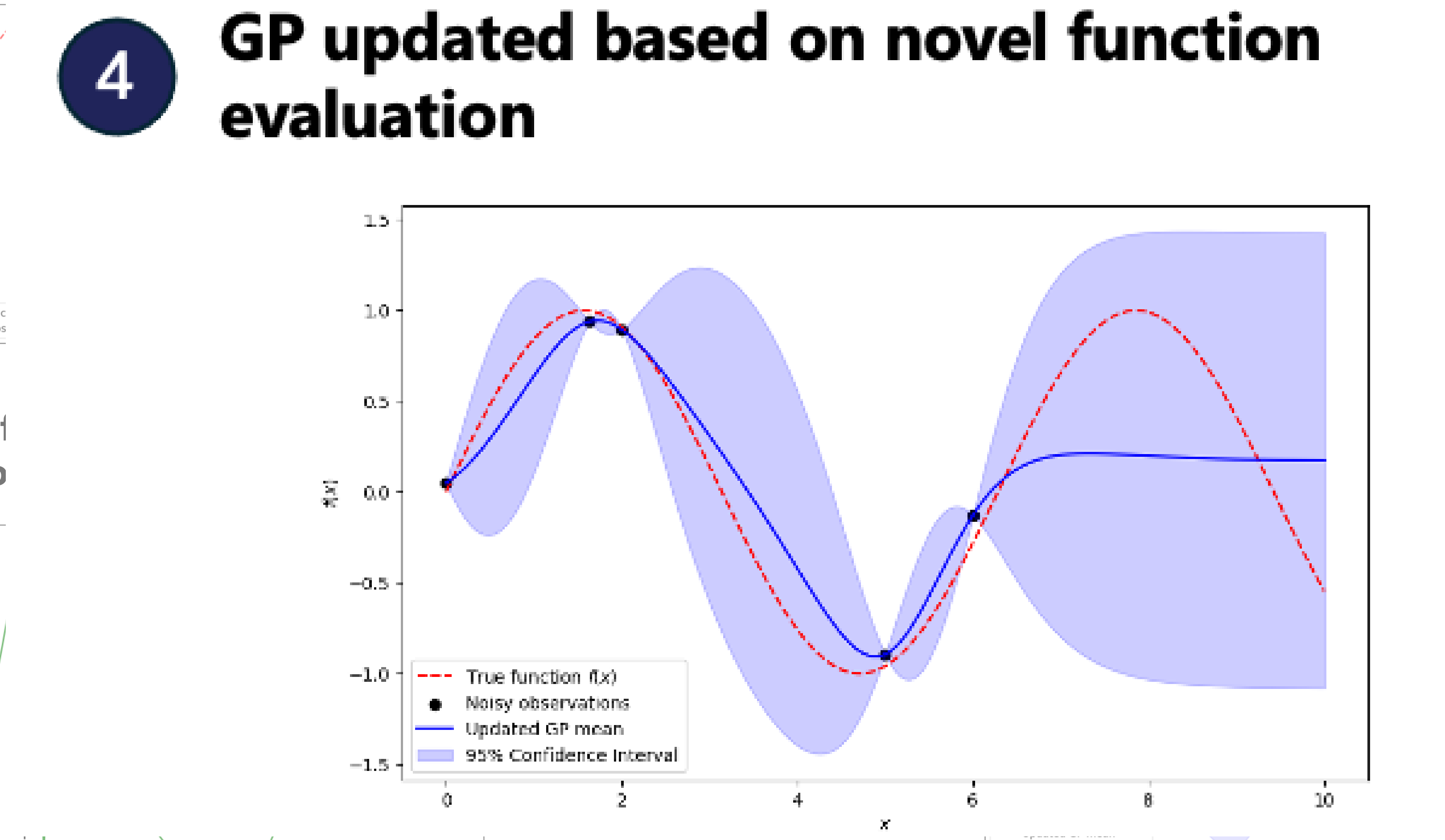
BO Loop in Detail

1 Existing function evaluations stored in dataset D_t



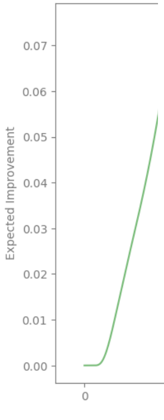
2 Gaussian Process builds model of the





3 Acquisition f
sampling po

ction



A

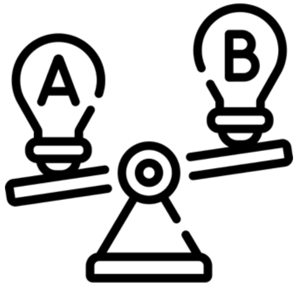
Experiments

Experimental Setup

Data Sources

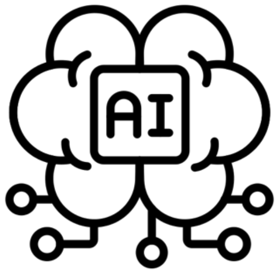
1. **Books:** Book Texts
2. **Common Crawl:** Web Data
3. **The Stack:** Code Data
4. **Pes2O:** Academic Papers
5. **Reddit:** Social Media

Baselines



1. **Sobol:** Random Search
2. **Uniform:** 20% uniform weighs
3. **LLaMa:** Weights reported
4. **BiMix:** SOTA Mixing Law
5. **MixLaws:** SOTA Mixing Law

Model & Metric

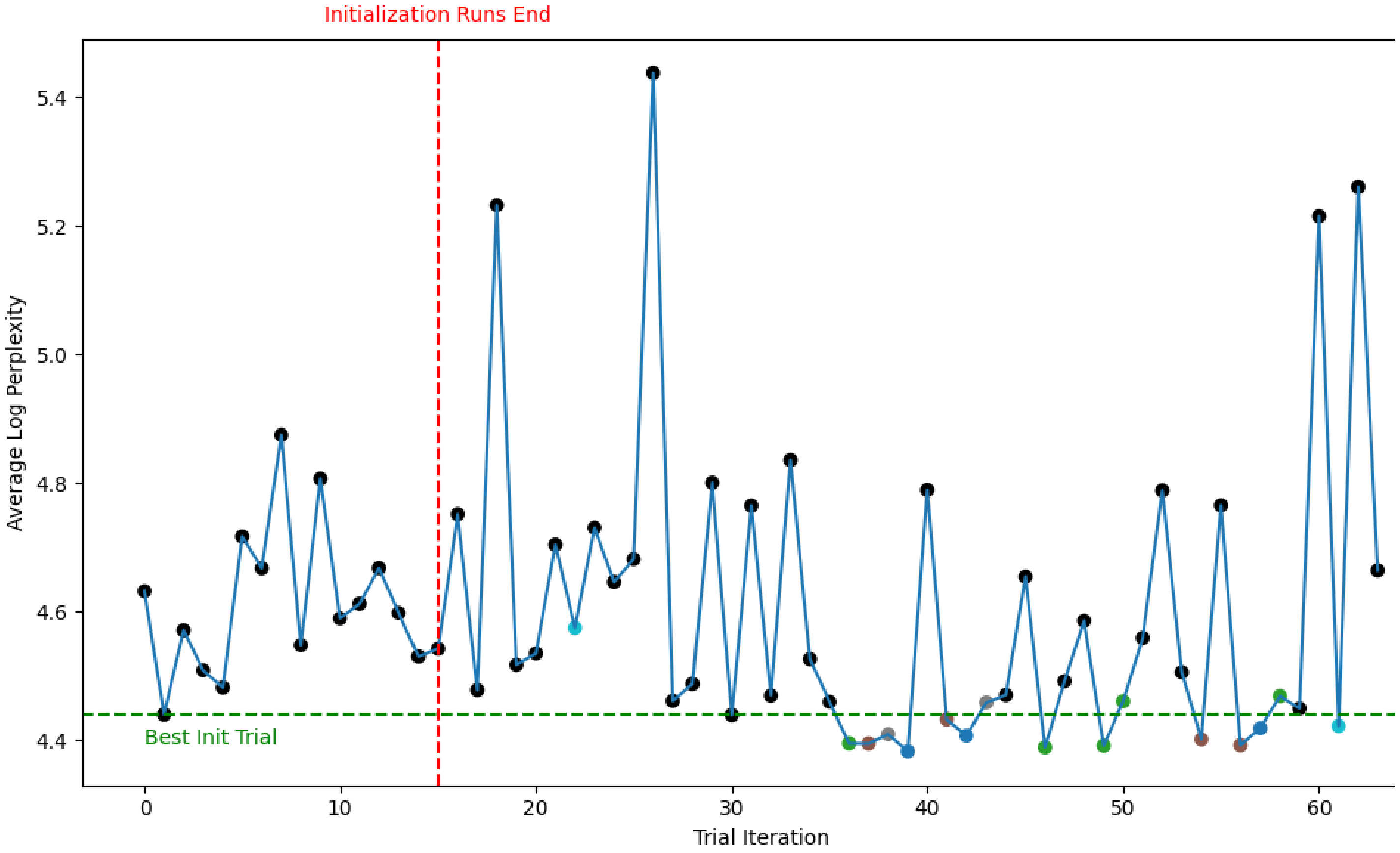


- **25M** parameters model size, trained **Chinchilla optimal**
- **64 trials** for each approach
- **Average perplexity** across all domains is **optimization criterion**[1]

[1]Each domain validated on held-out set, average compute across domains

Function Evaluations over time give insight into BO Process

BO performance over trial iterations[1]

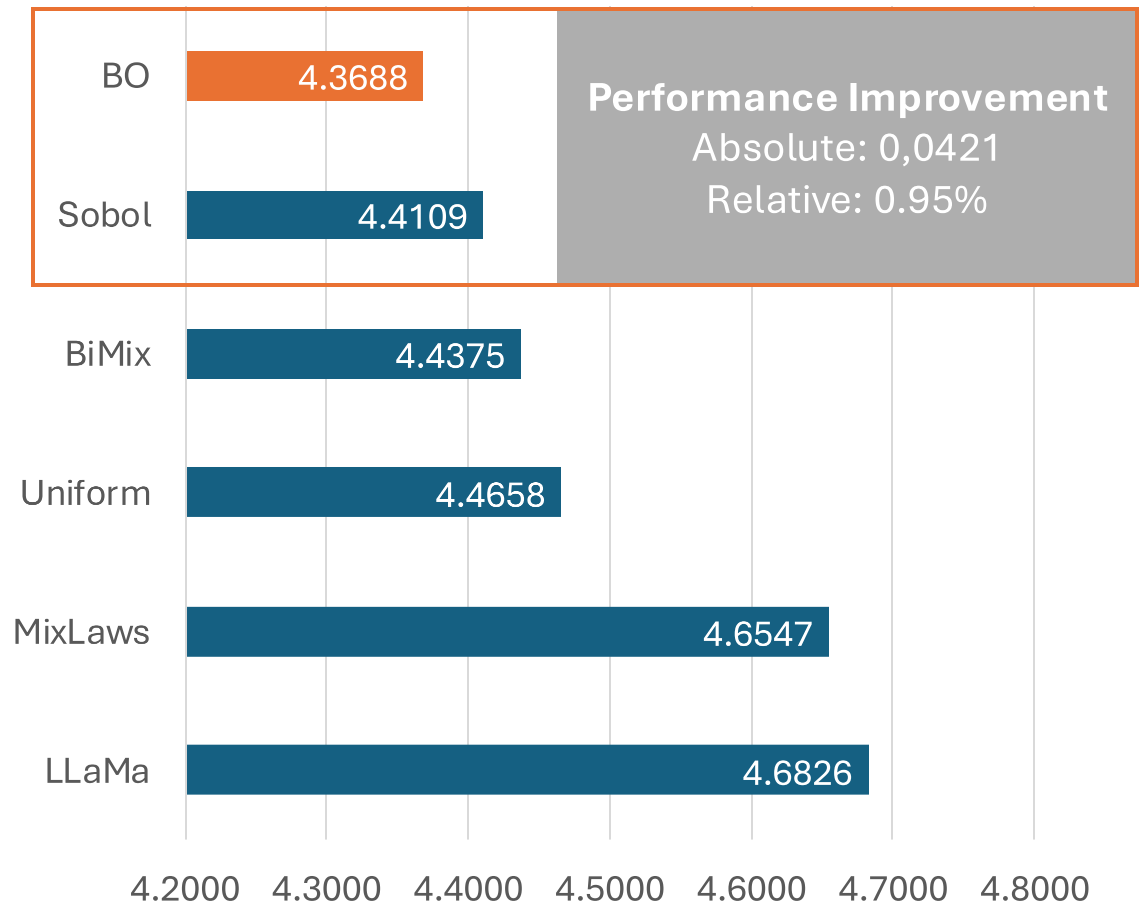


- BO achieves **significant improvement** over **initialization**
- **>35 trials required** before proposing good mixture
- **Exploitation:** many optimum share similar weight-cluster

[1]Colors visualize top 5 clusters of similar weights

BO is effective and outperforms all other baselines

Performance Comparison against benchmarks
[Values in avg. log perplexity]

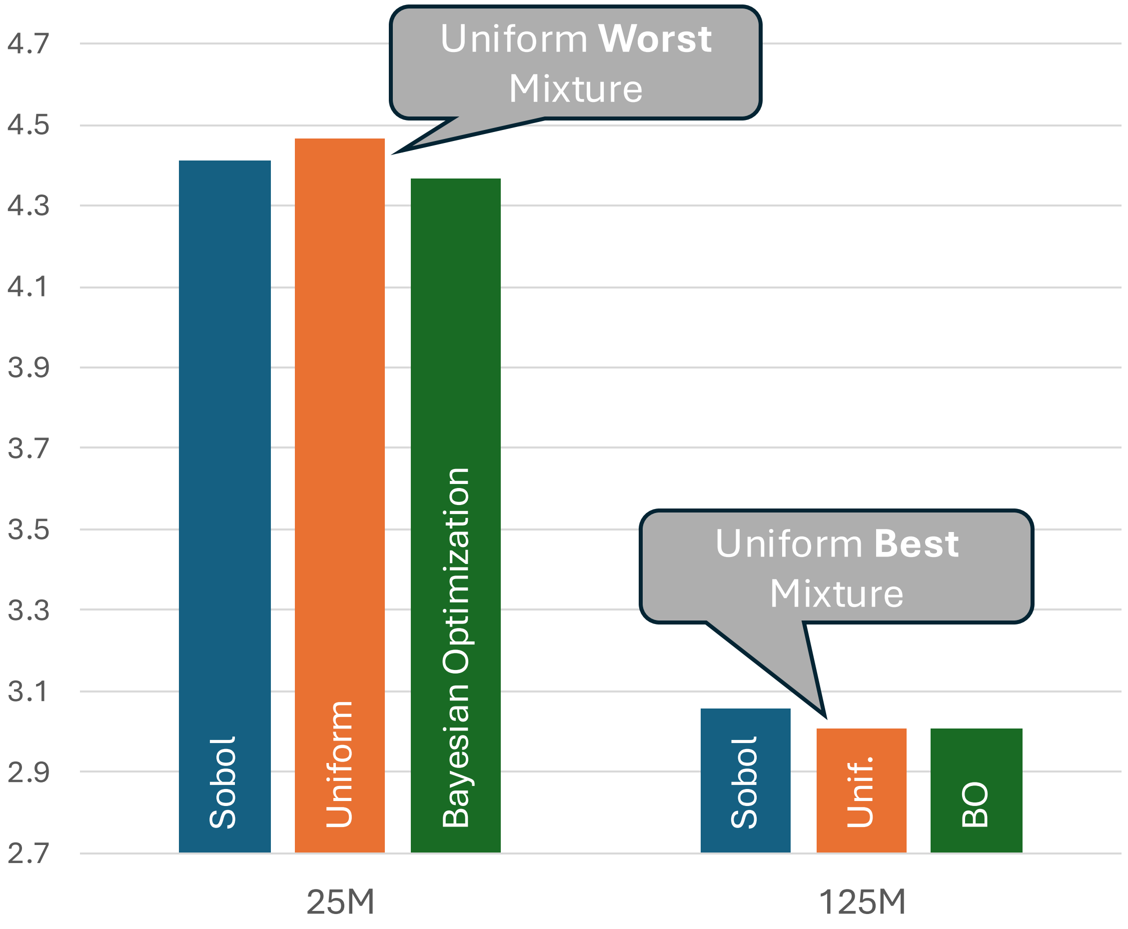


- **BO effective** and **identifies best-performing** mixture
- $\boldsymbol{\Delta 0.04}$ **improvement significant** and larger than experiment variance of $\sigma < 0.02$
- **No other** approach capable to **outperform random search**
- **MixLaws** with major **struggle** to model **true function shape** leading to bad performance
- **LLaMa weights worst**. High focus on CC could be disadvantage in large model[1]

[1]According to Kang et al.

Missing Scale Invariance Hinders Real World Application of BO

Scaling up same mixture from 25M model to 15M model [Values in avg. log perplexity]

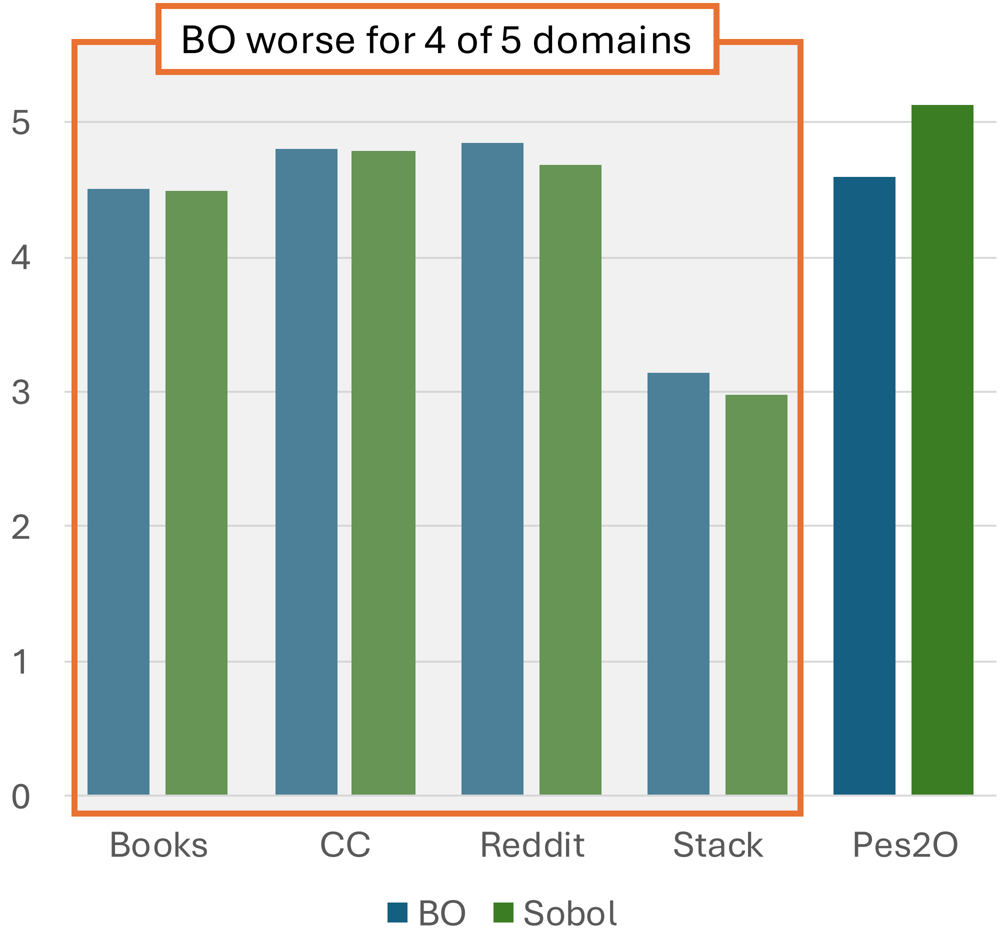


- **Assumption:** Optimal mixture identified at small proxy model translates to larger production model
- **Experiment:** Optimal 25M parameter mixture worse than non-optimal mixture when training 125M model
- When scaling up **uniform mixture** switches **from worst to best** performing mixture

➡ **Limited usability of our method.** More insight into scaling required to translate from proxy to production

Avg. Perplexity Criterion can Create Undesirable Optimization Incentives

Best BO mixture compared to best Sobol mixture per domain [Values in avg. log perplexity]



Goal of Data Selection:

- Create model that **performs well on training domains**
- Additional **generalization** to novel domains **desirable**

Limitations of Avg. Perplexity Criterion:

- Risk that model **trades of domains** to minimize overall loss
- Despite **better overall**, BO only outperforms Sobol on **one domain**

Conclusion

Conclusion: BO effective at identifying optimal mixtures at proxy model scale. Limitations currently prevent wide-scale adaptation

Result 1:

BO effective for optimizing mixtures of one architecture

- **BO only approach** capable of **outperforming random search**
- **Sample efficiency** expected from theory **validated** in practice
- **No magic:** Significant number of trials still required

Result 2:

Major limitations prevent adaptation of our (and other) methods in rea-world

- **Scale invariance** assumption **invalidated**. **Translating** insights from **proxy to production** model **difficult** in practice
- **Limitations** of **average performance criterion** identified. High **generalizability requires different** optimization **objective**
- **Not just our approach** suffers from limitations
  → **More research** required for **principle data mixtures**