Machine Learning-Based Estimation of Cumulants of Chiral Condensate via Multi-Ensemble Reweighting with Deborah.jl

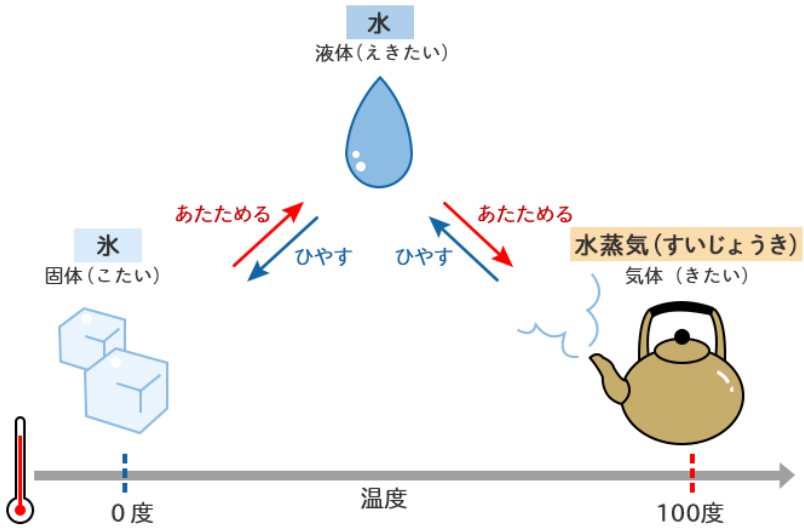
Benjamin J. Choi[1], Hiroshi Ohno[1], Akio Tomiya[2,3,4]

[1]Center for Computational Sciences, University of Tsukuba, Japan
[2]Division of Mathematical Sciences, Tokyo Woman's Christian University, Japan
[3]RIKEN Center for Computational Science, Kobe, Japan
[4]Department of Physics, Kyoto University, Japan

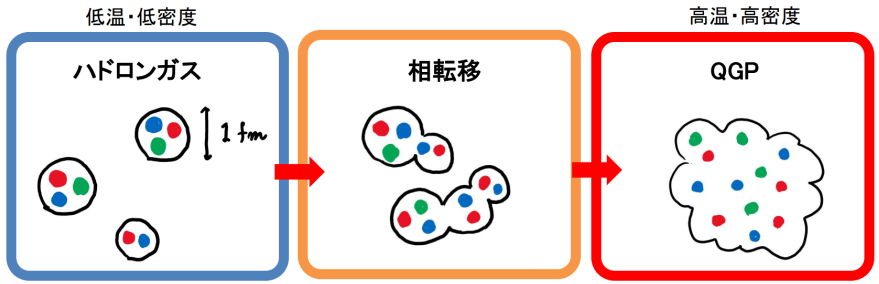
The 2nd "AI+HEP in East Asia" workshop @ KEK, Japan
21 January Reiwa 8th

筑波大学
計算科学研究センター
Center for Computational Sciences

Motivation of our work



(a) Okayama City Waterworks Bureau



(b) CCS, Univ. of Tsukuba

- Phase transition from hadron to QGP (quark-gluon plasma)
- We estimate the location of the critical endpoint (CEP) using the kurtosis intersection method, for example.
- Hence, we want to obtain cumulants of chiral order parameter, such as kurtosis, **precisely**.
  - ➡ This needs considerable computational cost.
- **Computational cost reduction using the machine learning!**

Introduction

- For Lattice QCD calculations
  - on observables such as cumulants of the chiral condensate,
  - the trace of operators ($\mathrm{Tr}\, O$) is often necessary.
    - *e.g.* $O = M^{-1}$ (inverse Dirac operator)
    - $\mathrm{Tr}\, O$ is obtained stochastically as $\mathrm{Tr}\, O \approx \frac{1}{h} \sum_{i=1}^{h} \eta_i^\dagger\, O\, \eta_i$, where η_i is random noise vector (e.g. Gaussian, Z_2, Z_4, $\cdots$).

- M (Dirac operator): a large sparse matrix on the lattice
  - ➡ Trace estimation by **linear CG solver** for stochastic sources
  - ➡ This needs considerable computational cost.

- We present our preliminary results on **cumulant estimation**
  - using the method proposed by Yoon *et al.*, PRD **100** 014504 (2019).

Cumulants of the chiral order parameter

- The cumulants are calculated with $\mathrm{Tr}\, M^{-n}$ $(n = 1, 2, 3, 4)$.
- For example, susceptibility (χ) is

$$\chi = \frac{\left\langle -N_{\mathrm{f}}\ \mathrm{Tr}\, M^{-2} + \left\{ N_{\mathrm{f}}\ \mathrm{Tr}\, M^{-1} \right\}^2 \right\rangle - \left\langle N_{\mathrm{f}}\ \mathrm{Tr}\, M^{-1} \right\rangle^2}{V}.$$

  - N_{f}: the number of quark flavor
  - V: the lattice volume

- In this way, we also calculate chiral condensate, skewness and kurtosis.

Machine Learning Estimation of Observables

Input

$$X = \{x_1, x_2, \cdots\}$$

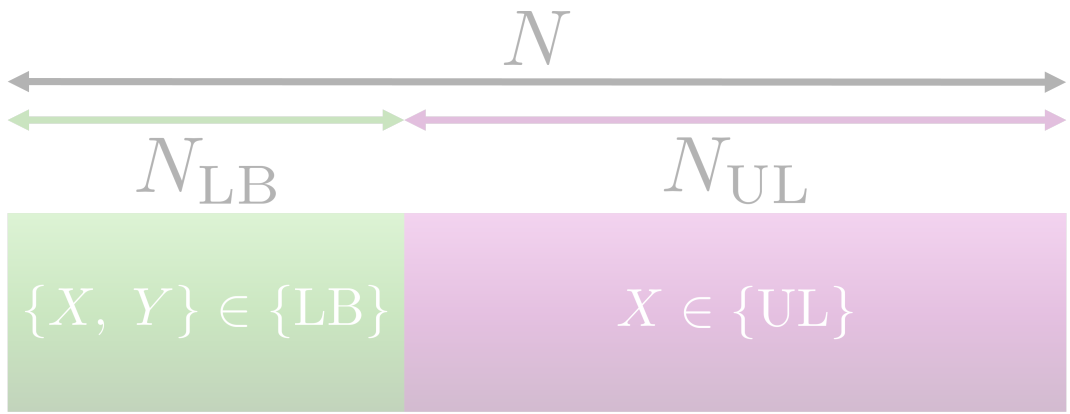
Machine

$f(X)$ ex) Gradient Boosting Decision Tree Algorithm

Output

$$Y^P \approx Y$$

- For ML estimation of $f(X) = Y^P \approx Y$,
  - # of data X: $N = N_{\mathrm{LB}} + N_{\mathrm{UL}}$
  - # of data Y: N_{LB}



1. Train f to get Y from X on labeled set.
2. Predict Y^P from X on **unlabeled** set:

$$f(X) = Y^P \approx Y\,.$$

- Do we use **all labeled set** for training?

Machine Learning Estimation of Observables

Input

$$X = \{x_1, x_2, \cdots\}$$

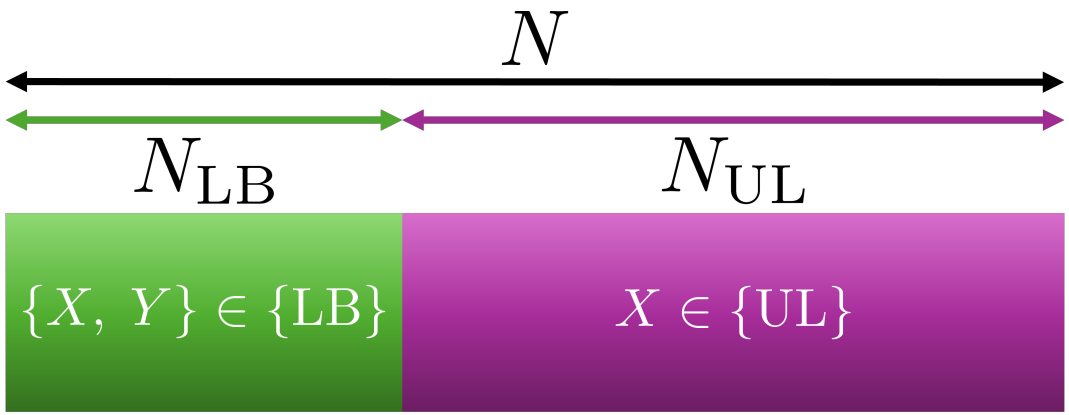
Machine

$f(X)$ ex) Gradient Boosting Decision Tree Algorithm

Output

$$Y^P \approx Y$$

- For ML estimation of $f(X) = Y^P \approx Y$,
  - # of data X: $N = N_{\mathrm{LB}} + N_{\mathrm{UL}}$
  - # of data Y: N_{LB}

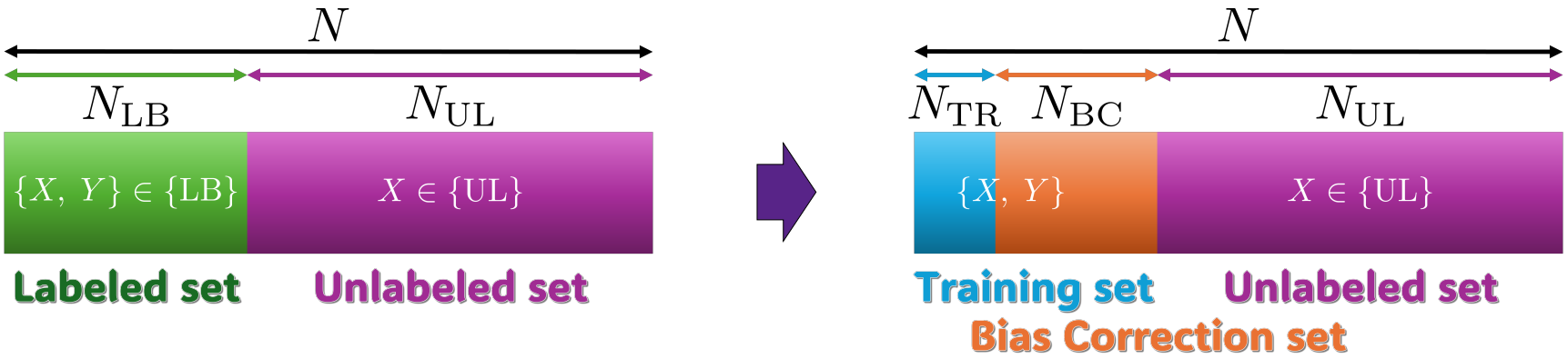


1. Train f to get Y from X on labeled set.
2. Predict Y^P from X on **unlabeled** set:

$$f(X) = Y^P \approx Y\,.$$

- Do we use **all labeled set** for training?

Bias correction in the ML estimation



- In general and in principle,

$$\bar{Y}_{(\rm UL)} = \frac{1}{N_{\rm UL}} \sum_{i=1}^{N_{\rm UL}} Y_i^P$$

is not exact due to the bias.

- Prediction bias in the ML

$$(\text{Bias}) = \langle Y \rangle - \langle Y^P \rangle$$

- We **split** labeled set into training and bias correction set.

1 Training with $X \in S_{\rm TR}^X$, $Y \in S_{\rm TR}^Y$

2 Predicting with

$$\frac{1}{N_{\rm UL}} \sum_{i=1}^{N_{\rm UL}} Y_i^P + \frac{1}{N_{\rm BC}} \sum_{j=1}^{N_{\rm BC}} (Y_j - Y_j^P)$$

(Yoon *et al.*, PRD **100** 014504)

Gradient boosting decision tree regression

- We use LightGBM (Microsoft) via `JuliaAI/MLJ.jl`.
  - boosting stage = 40 ➡ empirically determined with L2-Loss plot.
  - Depth of tree = 3, learning rate = 0.1, subsampling = 0.7
    ➡ Same with Yoon *et al.*, PRD **100** 014504 (2019)

- Statistical error estimation: Bootstrap resampling

- Check $\mathcal{P}1$ and $\mathcal{P}2$ as in Yoon *et al.*, PRD **100** 014504 (2019)

1 $\mathcal{P}1$: bias corrected ML prediction

$$\bar{Y}_{\mathcal{P}1} = \frac{1}{N_{\mathrm{UL}}} \sum_{i \in \{\mathrm{UL}\}} Y_i^{\mathrm{P}} + \frac{1}{N_{\mathrm{BC}}} \sum_{j \in \{\mathrm{BC}\}} \left(Y_j - Y_j^{\mathrm{P}} \right) \tag{1}$$

2 $\mathcal{P}2$: weighted average of $\mathcal{P}1$ and direct measured labeled set

$$\bar{Y}_{\mathcal{P}2} = \frac{N_{\mathrm{UL}}}{N} \bar{Y}_{\mathcal{P}1} + \frac{N_{\mathrm{LB}}}{N} \bar{Y}_{(\mathrm{LB})} \tag{2}$$

➡ To improve the statistical precision

Scanning along the ratio of labeled/training set

- N = # of total data set ($N = N_{\rm LB} + N_{\rm UL}$)
- $N_{\rm LB}$ = # of labeled set ($N_{\rm LB} = N_{\rm TR} + N_{\rm BC}$)
- $N_{\rm TR}$ = # of training set

1 Observe results for each $\mathcal{R}_{\rm LB} \equiv \dfrac{N_{\rm LB}}{N}$, *e.g.*:

$$\mathcal{R}_{\rm LB} = 5\%,\ 10\%,\ \ldots,\ 50\%\,.$$

2 Observe results for each $\mathcal{R}_{\rm TR} \equiv \dfrac{N_{\rm TR}}{N_{\rm LB}}$, *e.g.*:

$$\mathcal{R}_{\rm TR} = 0\%,\ 10\%,\ \ldots,\ 100\%\,.$$

- ➡ We perform scanning and observe results for each $(\mathcal{R}_{\rm LB}, \mathcal{R}_{\rm TR})$.

- We observe
  - $\mathcal{R}_{\rm TR} = 0$ % to check the labeled set itself,
  - $\mathcal{R}_{\rm TR} = 100$ % to check the result without the bias correction.

One of our methods to calculate the cumulants

- One of our approaches to calculate the machine-learning-derived cumulants is as follows.

1 We use $\mathrm{Tr}\, M^{-1}$ from the original dataset.

2 We perform ML estimation of $\mathrm{Tr}\, M^{-n}$ $(n = 2, 3, 4)$ from $\mathrm{Tr}\, M^{-1}$ as input.

3 Then we calculate the cumulants:
chiral condensate, susceptibility, skewness and kurtosis.

- ➡ We observe results for each $(\mathcal{R}_{\mathrm{LB}}, \mathcal{R}_{\mathrm{TR}})$.

K with ML results ($\mathcal{P}1$) at the κ_t: $1\% \leq \mathcal{R}_{\mathrm{LB}} \leq 25\%$

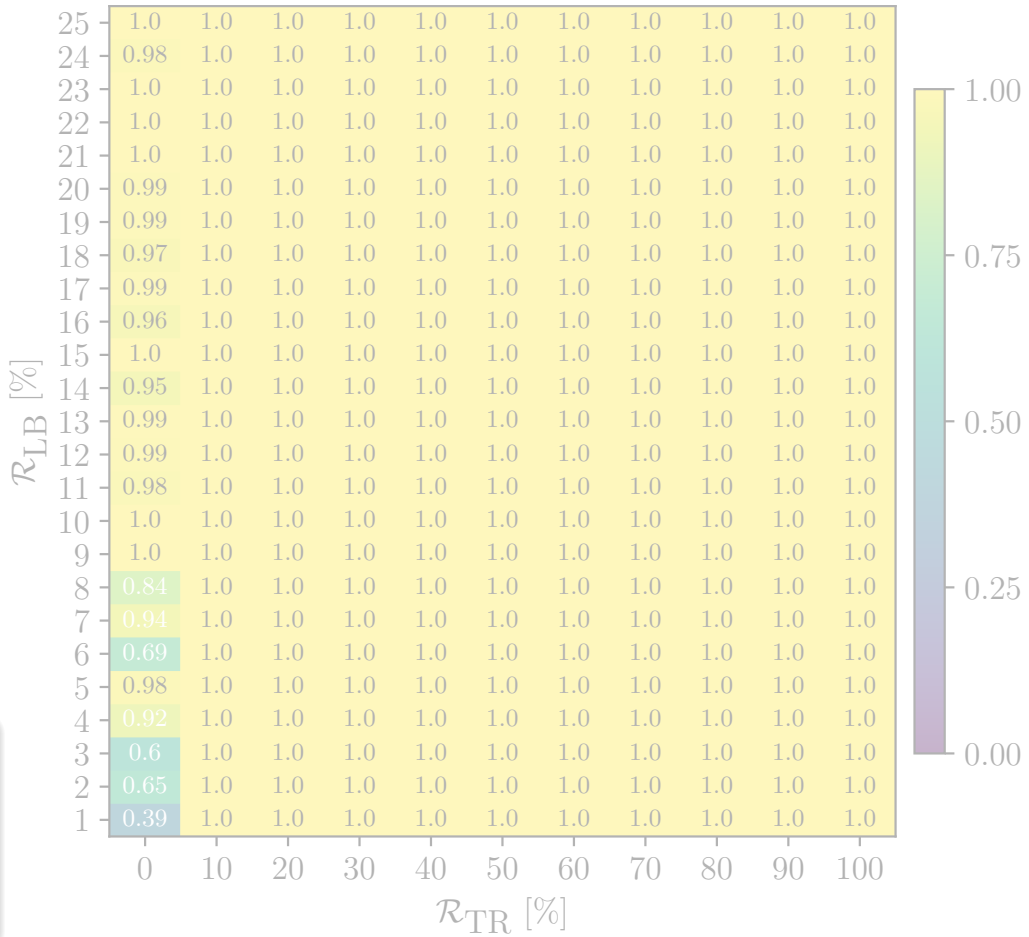
- Bhattacharyya coefficient

$$C_{\mathrm{B}}(x, r) = Z \exp\left[-\frac{x^2}{4(1+r^2)}\right]$$

$$\text{where} \quad Z = \sqrt{\frac{2r}{1+r^2}} \quad \text{and}$$

$$x = \frac{|\bar{Y}_{\mathrm{Orig}} - \bar{Y}_{\mathcal{P}1}|}{\sigma_{\mathrm{Orig}}}, \quad r = \frac{\sigma_{\mathcal{P}1}}{\sigma_{\mathrm{Orig}}}$$

- $C_{\mathrm{B}} \in [0, 1]$ (higher is better)
  - 1: best overlap
  - 0: no overlap



- Lattice setup: $12^3 \times 4$, $N_{\mathrm{f}} = 4$ Wilson clover quark, Iwasaki gluon with $\beta = 1.60$
- $\mathrm{Tr}\, M^{-1}$ baseline, $\mathrm{Tr}\, M^{-n}$ $(n = 2, 3, 4)$ trained with $\mathrm{Tr}\, M^{-1}$ by LightGBM.
- Data from "Ohno *et al.* PoS **LATTICE2018** (2018) 174"

K with ML results ($\mathcal{P}1$) at the κ_t: $1\% \leq \mathcal{R}_{\mathrm{LB}} \leq 25\%$

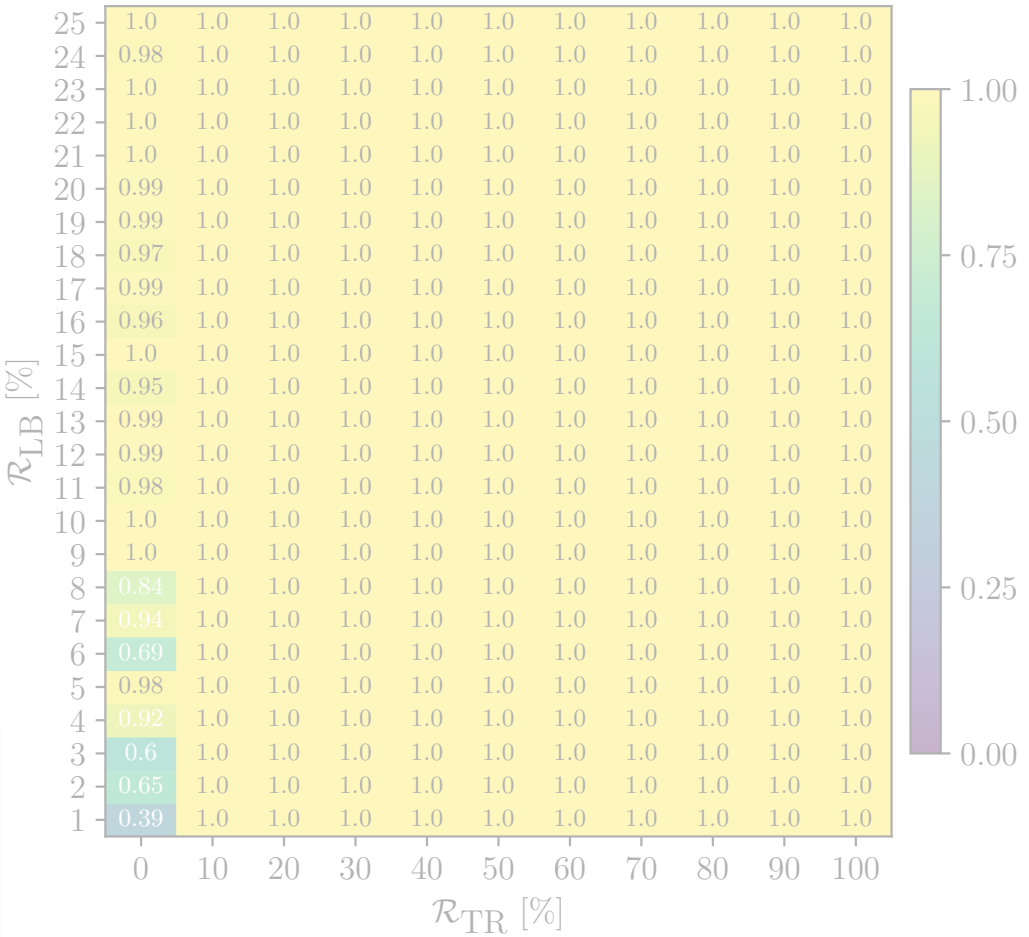
- Bhattacharyya coefficient

$$C_{\mathrm{B}}(x, r) = Z \exp\left[-\frac{x^2}{4(1+r^2)}\right]$$

$$\text{where} \quad Z = \sqrt{\frac{2r}{1+r^2}} \quad \text{and}$$

$$x = \frac{|\bar{Y}_{\mathrm{Orig}} - \bar{Y}_{\mathcal{P}1}|}{\sigma_{\mathrm{Orig}}}, \quad r = \frac{\sigma_{\mathcal{P}1}}{\sigma_{\mathrm{Orig}}}$$

- **$C_{\mathrm{B}} \in [0, 1]$ (higher is better)**
  - **1: best overlap**
  - **0: no overlap**



- Lattice setup: $12^3 \times 4$, $N_{\mathrm{f}} = 4$ Wilson clover quark, Iwasaki gluon with $\beta = 1.60$
- $\mathrm{Tr}\, M^{-1}$ baseline, $\mathrm{Tr}\, M^{-n}$ $(n = 2, 3, 4)$ trained with $\mathrm{Tr}\, M^{-1}$ by LightGBM.
- Data from "Ohno *et al.* PoS **LATTICE2018** (2018) 174"

K with ML results ($\mathcal{P}1$) at the κ_t: $1\% \leq \mathcal{R}_{\mathrm{LB}} \leq 25\%$

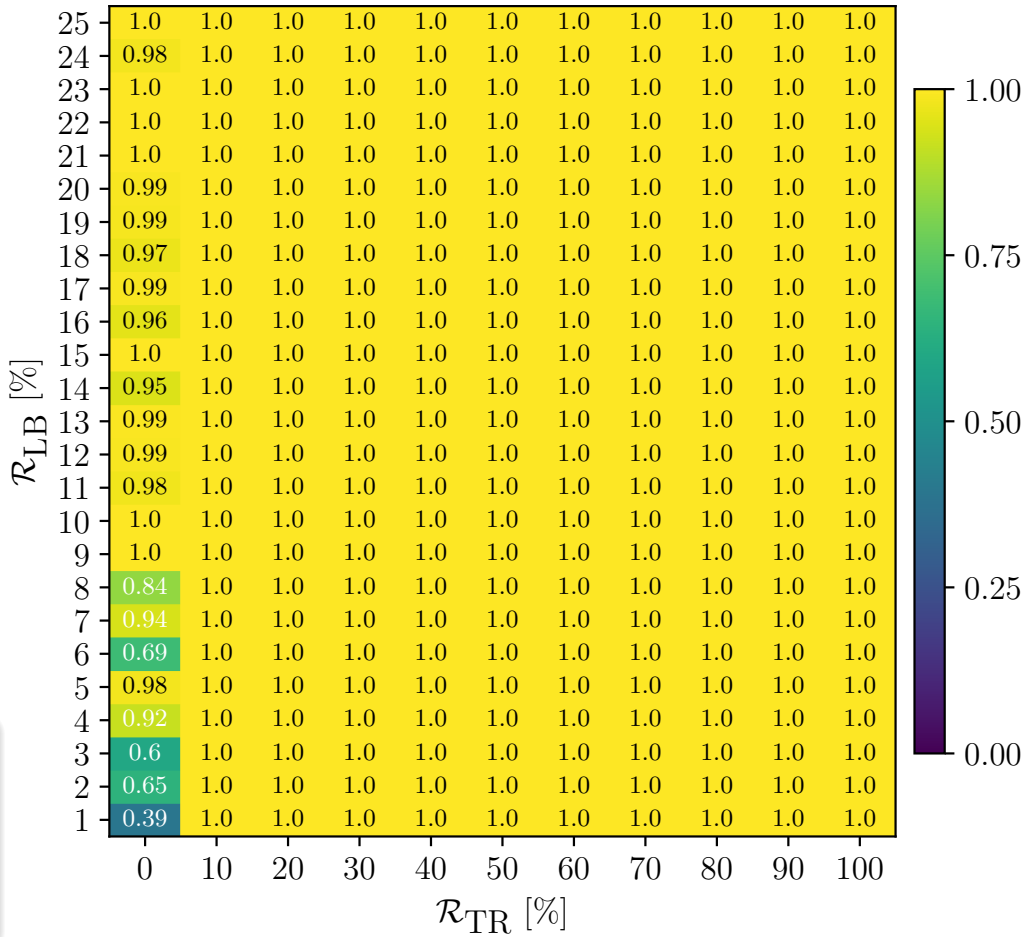
- Bhattacharyya coefficient

$$C_{\mathrm{B}}(x, r) = Z \exp\left[-\frac{x^2}{4(1+r^2)}\right]$$

$$\text{where} \quad Z = \sqrt{\frac{2r}{1+r^2}} \quad \text{and}$$

$$x = \frac{|\bar{Y}_{\mathrm{Orig}} - \bar{Y}_{\mathcal{P}1}|}{\sigma_{\mathrm{Orig}}}, \quad r = \frac{\sigma_{\mathcal{P}1}}{\sigma_{\mathrm{Orig}}}$$

- **$C_{\mathrm{B}} \in [0, 1]$ (higher is better)**
  - **1: best overlap**
  - **0: no overlap**



- Lattice setup: $12^3 \times 4$, $N_{\mathrm{f}} = 4$ Wilson clover quark, Iwasaki gluon with $\beta = 1.60$
- $\mathrm{Tr}\, M^{-1}$ baseline, $\mathrm{Tr}\, M^{-n}$ $(n = 2, 3, 4)$ trained with $\mathrm{Tr}\, M^{-1}$ by LightGBM.
- Data from "Ohno *et al.* PoS **LATTICE2018** (2018) 174"

K with ML results ($\mathcal{P}1$) at the κ_t: $1\% \leq \mathcal{R}_{\mathrm{LB}} \leq 25\%$

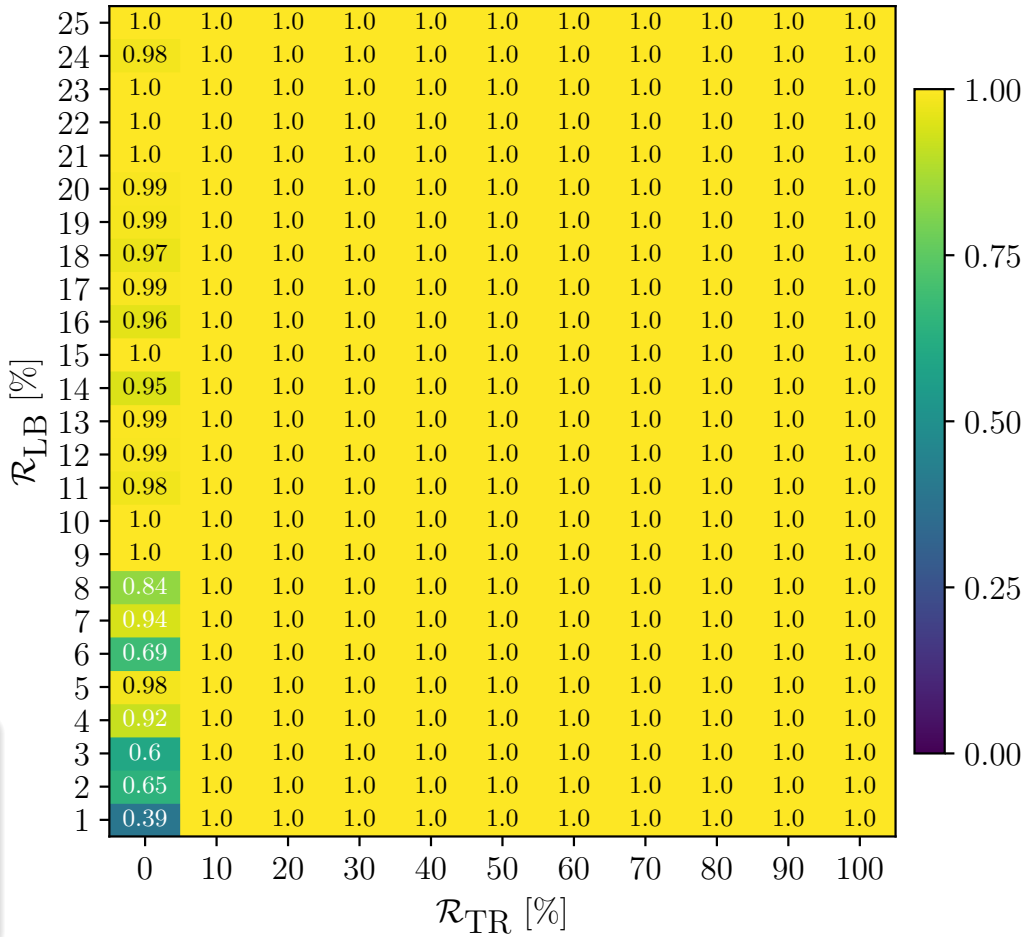
- Bhattacharyya coefficient

$$C_{\mathrm{B}}(x, r) = Z \exp\left[-\frac{x^2}{4(1+r^2)}\right]$$

$$\text{where} \quad Z = \sqrt{\frac{2r}{1+r^2}} \quad \text{and}$$

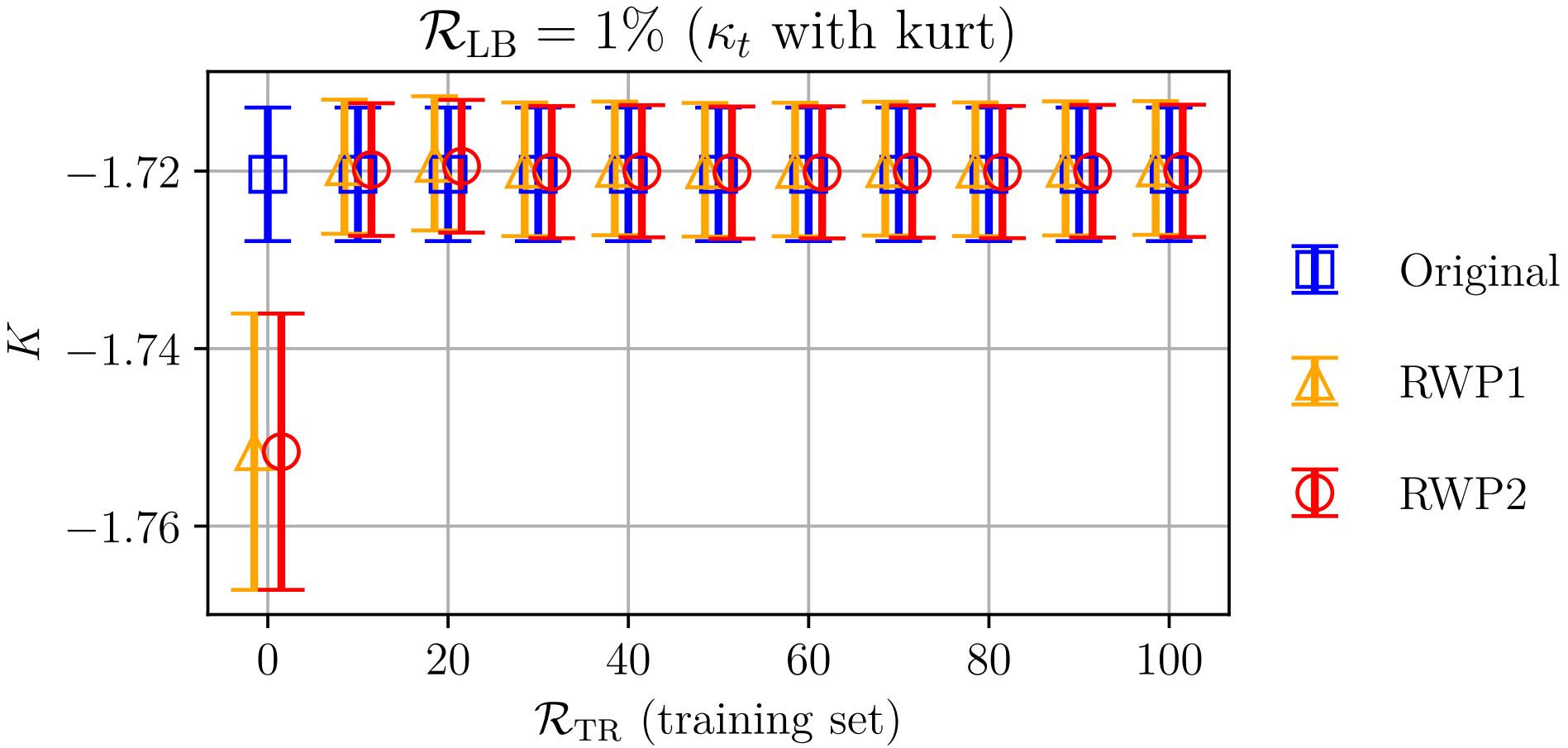
$$x = \frac{|\bar{Y}_{\mathrm{Orig}} - \bar{Y}_{\mathcal{P}1}|}{\sigma_{\mathrm{Orig}}}, \quad r = \frac{\sigma_{\mathcal{P}1}}{\sigma_{\mathrm{Orig}}}$$

- **$C_{\mathrm{B}} \in [0, 1]$ (higher is better)**
  - **1: best overlap**
  - **0: no overlap**



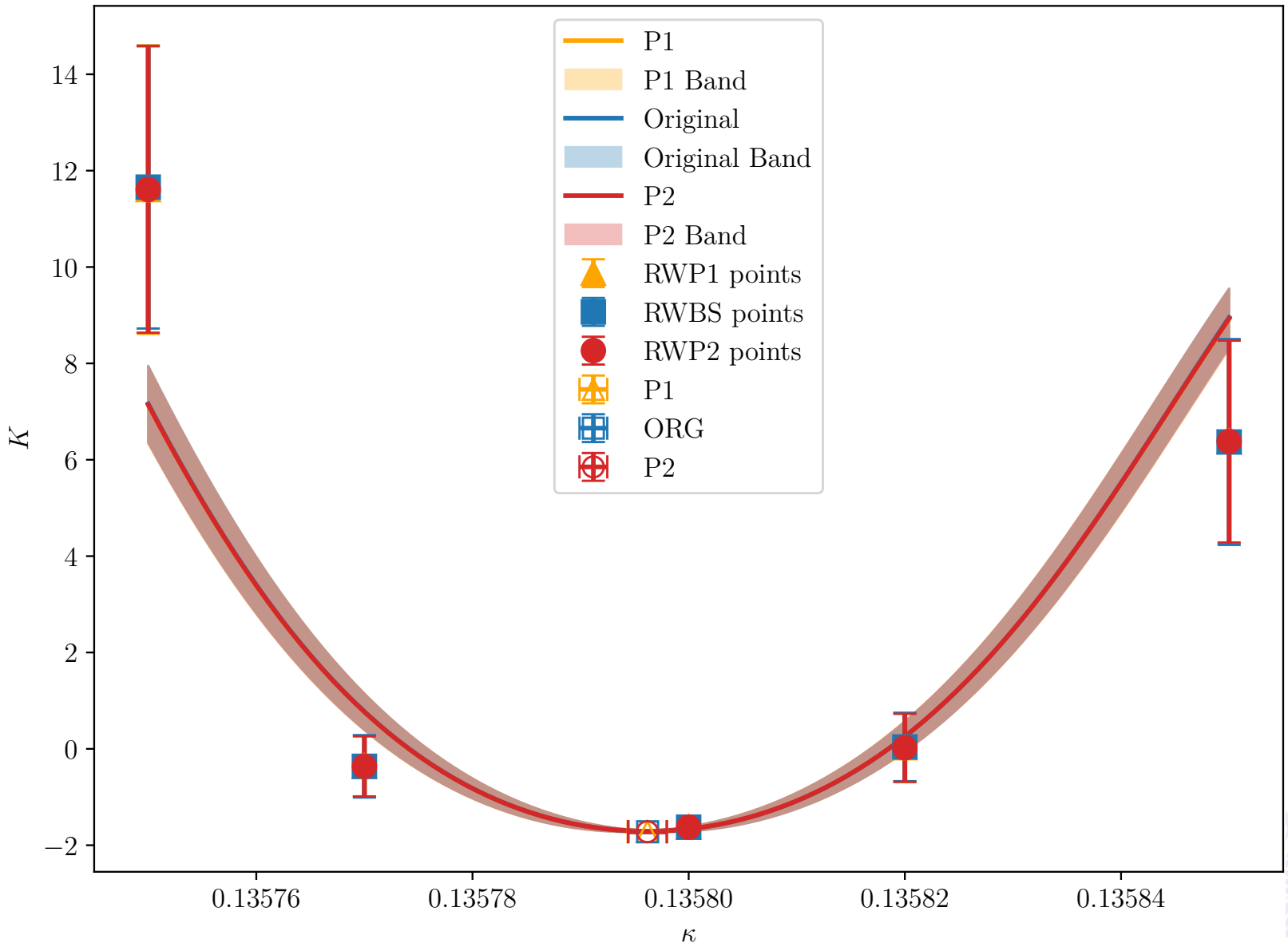
- 100 % original $\mathrm{Tr}\, M^{-1}$ ➡ drives Σ, χ, S, K
- $\mathrm{Tr}\, M^{-1}$ dominates all cumulant definitions!

$K(\kappa_t)$ with ML estimation at $\mathcal{R}_{\mathrm{LB}} = 1$ %



- At $\mathcal{R}_{\mathrm{TR}} = 0\%$, no ML used — only 1% $\mathcal{R}_{\mathrm{LB}}$ ➡ **very poor statistics!**
  - **blue square**: kurtosis with original dataset
  - **orange triangle**: kurtosis with ML estimation ($\mathcal{P}1$)
  - **red circle**: kurtosis with ML estimation ($\mathcal{P}2$)

Reweighting curve of K at $\{\mathcal{R}_{\mathrm{LB}}\,,\ \mathcal{R}_{\mathrm{TR}}\} = \{1\%\,,\ 10\%\}$



Summary (1)

- We calculated cumulants of the chiral order parameters
  - Using 100 % of original dataset of $\mathrm{Tr}\, M^{-1}$
  - and ML estimation results on $\mathrm{Tr}\, M^{-n}$ $(n = 2, 3, 4)$ with various partial ratios $(\mathcal{R}_{\mathrm{LB}}, \mathcal{R}_{\mathrm{TR}})$.
- In this setup, we found that the cumulants are calculated well even with $\mathcal{R}_{\mathrm{LB}} = 1$ %.
  - ➡ If this turns out to work well, then we can determine cumulants (quark condensate, susceptibility, skewness, kurtosis) with $\dfrac{100 + 1 + 1 + 1}{400} = \mathbf{25.75}\%$ of computing power for the case without ML estimation.
- Apply ML estimation even to $\mathrm{Tr}\, M^{-1}$!

Summary (2)

- We are currently exploring the possibility of reducing computing power even further by applying ML estimation also to $\mathrm{Tr}\, M^{-1}$.
  - This could potentially lower the requirement below **25.75 %** of the computing cost compared to the case without ML.

- Preliminary results are available from this setup.
  - These will be lightly introduced here.

- For the method to be viable, similar performance must hold across different lattice volumes and coupling constants. Testing of such cases is in progress.

- In this case, $\mathrm{Tr}\, M^{-n}$ $(n = 1, 2, 3, 4)$ are trained using Plaquette and Rectangle observables, and then combined to compute cumulants.

K with ML results ($\mathcal{P}1$) at the κ_t: $5\% \leq \mathcal{R}_{\mathrm{LB}} \leq 50\%$

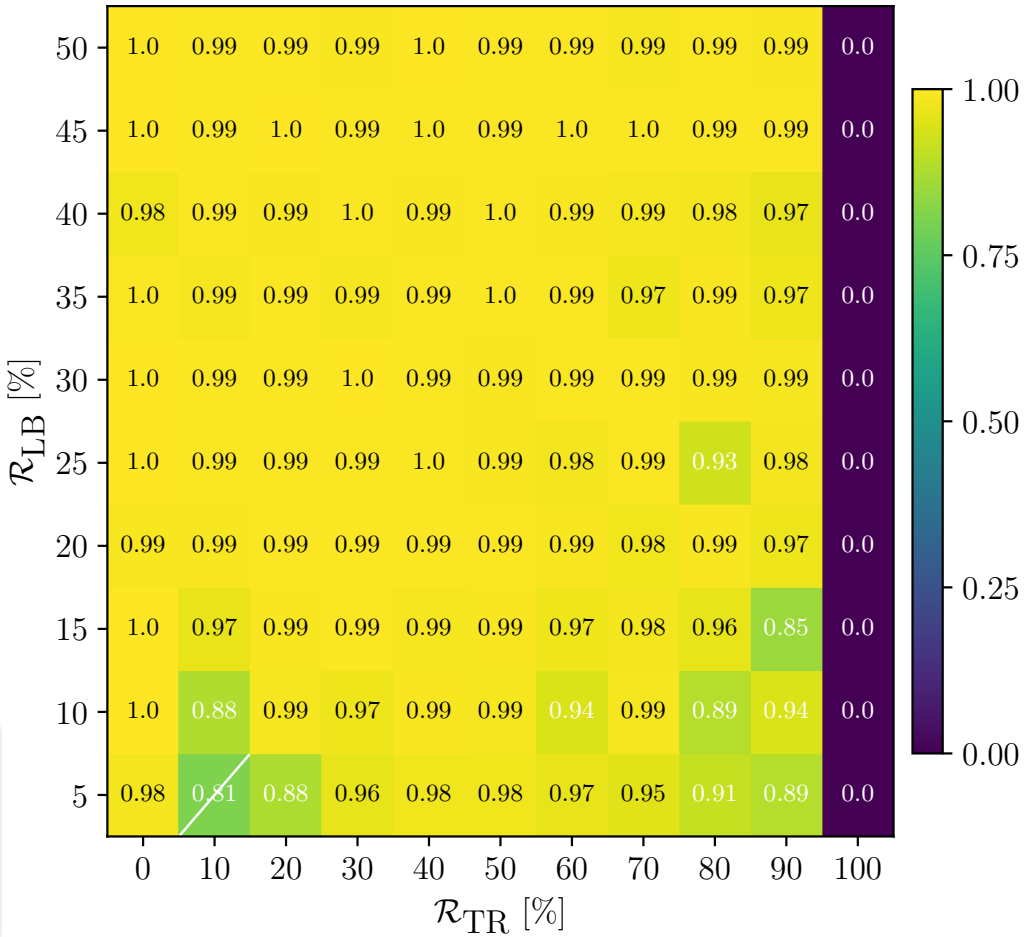
- Bhattacharyya coefficient **(note)**

$$C_{\mathrm{B}}(x, r) = Z \exp\left[-\frac{x^2}{4(1+r^2)}\right]$$

where $Z = \sqrt{\dfrac{2r}{1+r^2}}$ and

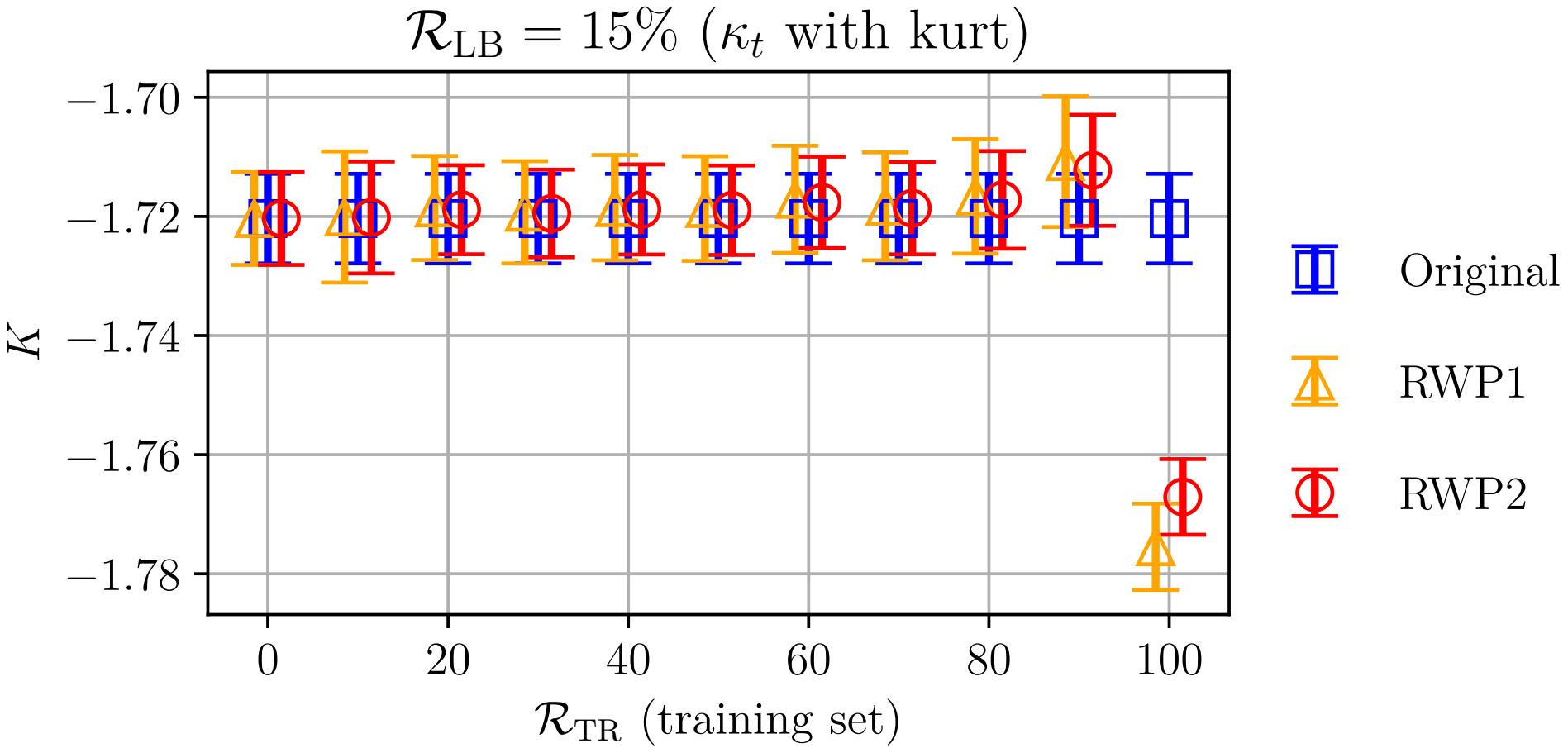
$$x = \frac{|\bar{Y}_{\mathrm{Orig}} - \bar{Y}_{\mathcal{P}1}|}{\sigma_{\mathrm{Orig}}}, \quad r = \frac{\sigma_{\mathcal{P}1}}{\sigma_{\mathrm{Orig}}}$$

- **$C_{\mathrm{B}} \in [0, 1]$ (higher is better)**
  - **1: best overlap**
  - **0: no overlap**



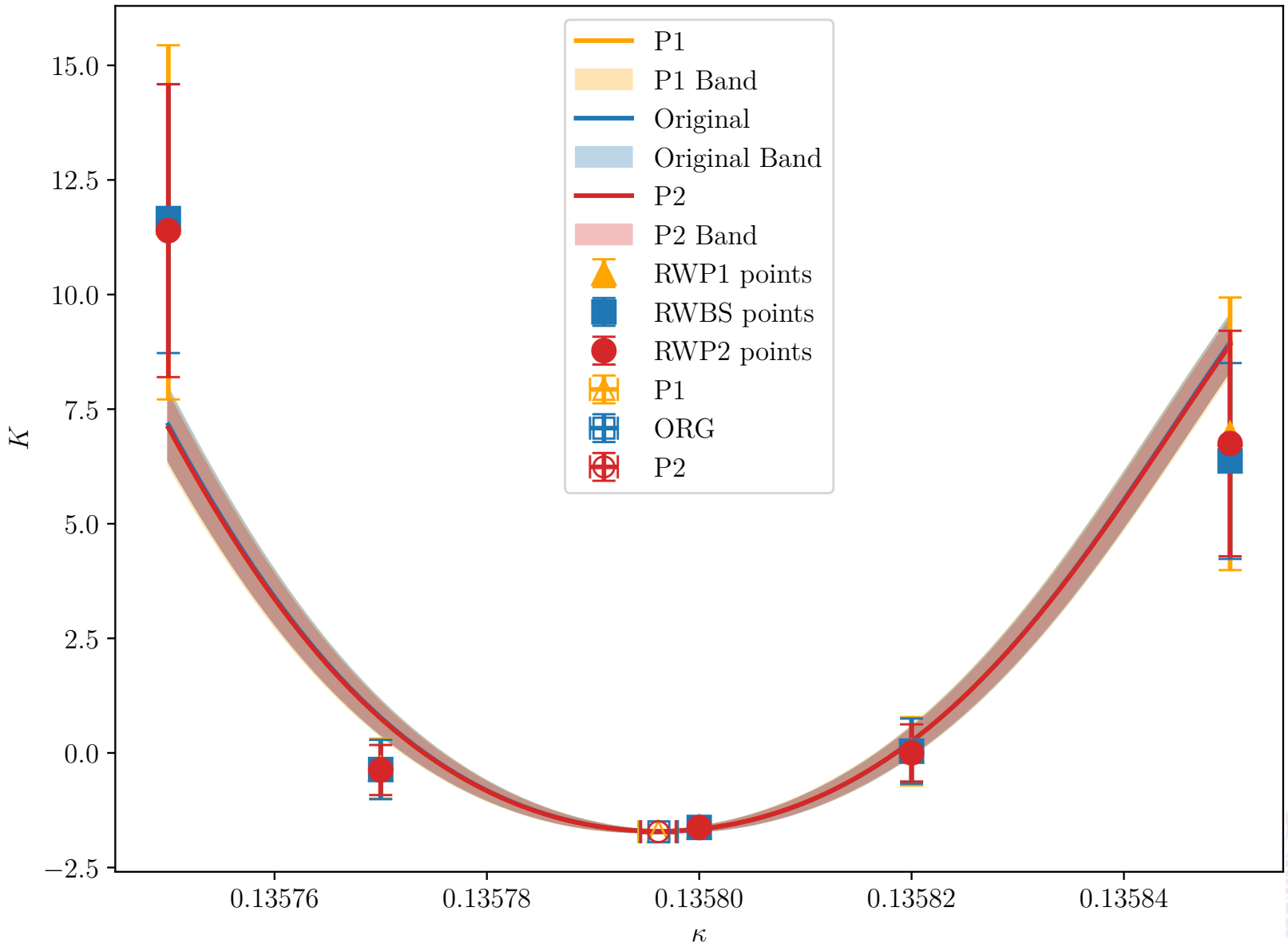
- Lattice setup: $12^3 \times 4$, $N_{\mathrm{f}} = 4$ Wilson clover quark, Iwasaki gluon with $\beta = 1.60$
- $\mathrm{Tr}\, M^{-n}$ $(n = 1, 2, 3, 4)$ trained with PLAQUETTE and RECTANGLE by LightGBM.
- Data from "Ohno *et al.* PoS **LATTICE2018** (2018) 174"

$K(\kappa_t)$ with ML estimation at $\mathcal{R}_{\rm LB} = 15\%$



- $\mathcal{R}_{\rm TR} = 100\%$: entire LB set used for training ➡ **no bias correction!**
  - **blue square**: kurtosis with original dataset
  - **orange triangle**: kurtosis with ML estimation ($\mathcal{P}1$)
  - **red circle**: kurtosis with ML estimation ($\mathcal{P}2$)

Reweighting curve of K at $\{\mathcal{R}_{\mathrm{LB}}\,,\ \mathcal{R}_{\mathrm{TR}}\} = \{15\%\,,\ 20\%\}$



Summary and To-Do List

- **Case 1:** Cumulant estimation using the original $\mathrm{Tr}\, M^{-1}$ data and training $\mathrm{Tr}\, M^{-n}$ $(n = 2, 3, 4)$ from it
  - Required computing power reduced to about **25.75%** of the full cost.

- **Case 2:** Training $\mathrm{Tr}\, M^{-n}$ $(n = 1, 2, 3, 4)$ simultaneously with Plaquette and Rectangle
  - In some preliminary datasets, the required computing power dropped below **25%**, and in certain cases reached as low as **∼20%**.
  - **Bias correction seems to play an important role for derived quantities such as cumulants.**

- Expand the analysis with other gauge ensembles.
  - Vary the lattice volume $V = N_{\mathrm{S}}^3 \times N_{\mathrm{T}}$ and the lattice spacing a.

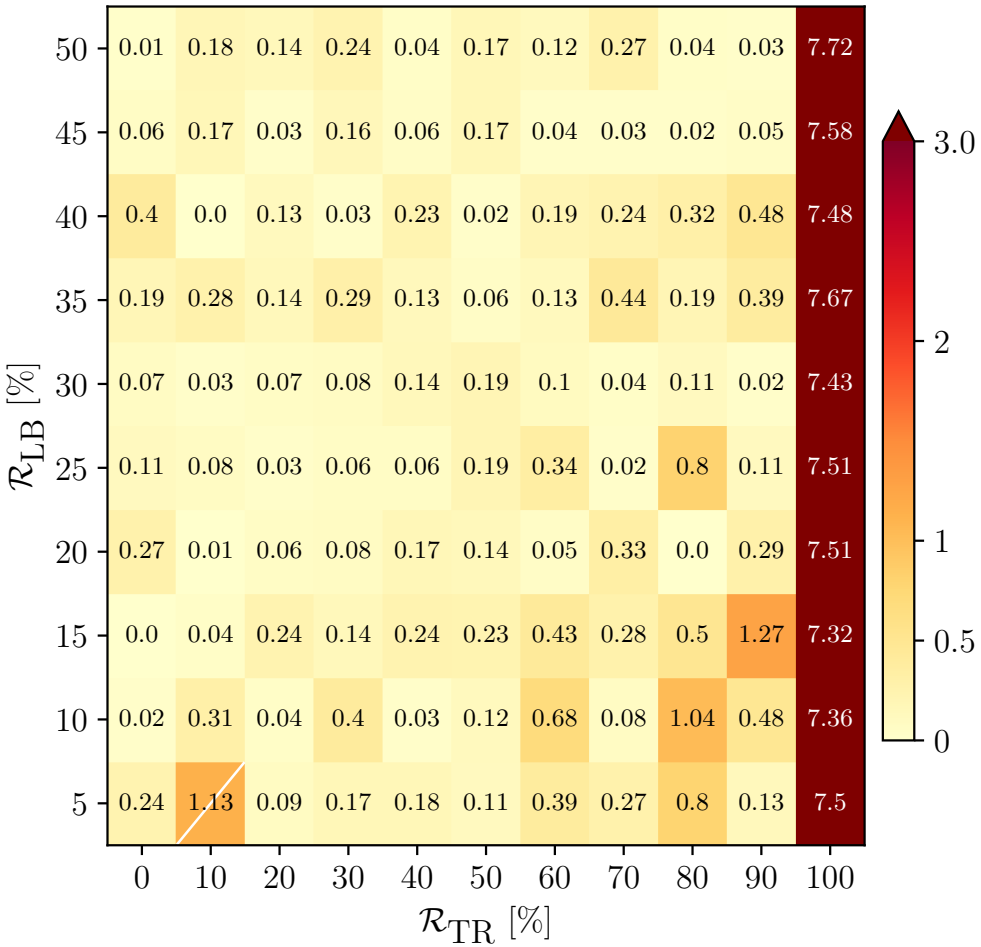
- The paper will be submitted soon, along with the public release of Deborah.jl. Please stay tuned!

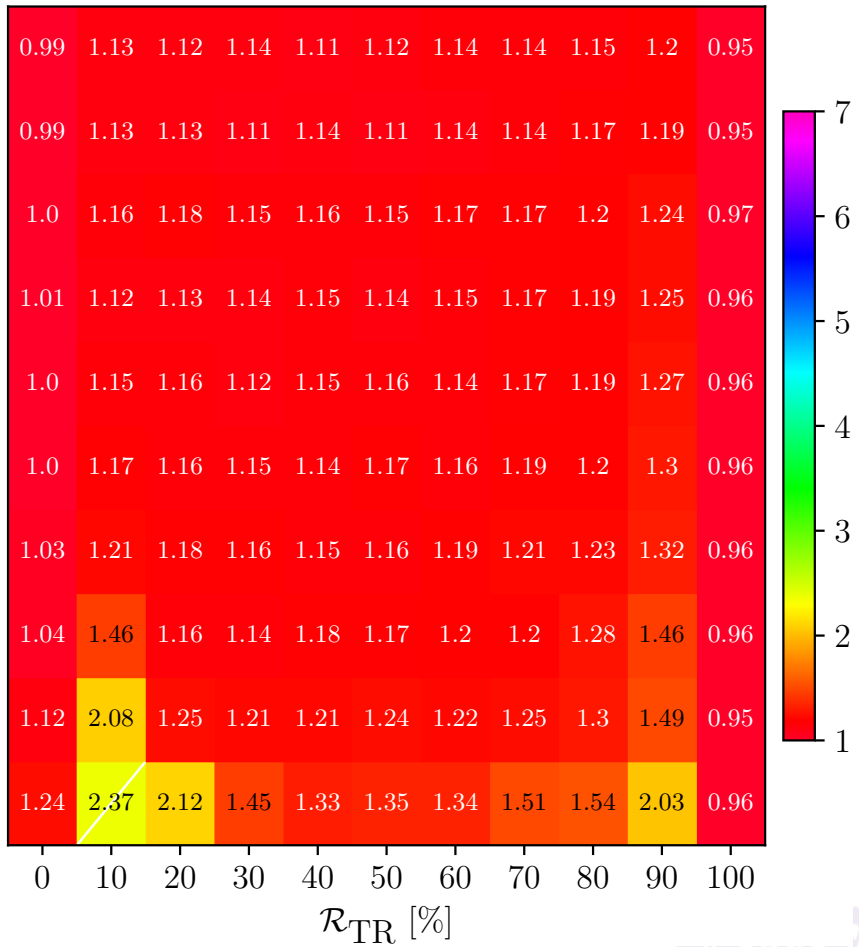
Thank you very much!

Backup slides

K with ML results ($\mathcal{P}1$) at the κ_t: $5\% \leq \mathcal{R}_{\mathrm{LB}} \leq 50\%$

- Left: x map of $C_{\mathrm{B}}(x, r)$; Right: r map of the same.





- When $x \equiv |\bar{Y}_{\mathrm{Orig}} - \bar{Y}_{\mathcal{P}1}|/\sigma_{\mathrm{Orig}} \approx 0$, the $\bar{Y}_{\mathcal{P}1}$ captures $\bar{Y}_{\mathrm{Orig}}$ **well**.
- When $r \equiv \sigma_{\mathcal{P}1}/\sigma_{\mathrm{Orig.}} \approx 1$, the $\sigma_{\mathcal{P}1}$ captures $\sigma_{\mathrm{Orig.}}$ **well**. **(note)**

K with ML results ($\mathcal{P}1$) at the κ_t: $1\% \leq \mathcal{R}_{\mathrm{LB}} \leq 25\%$

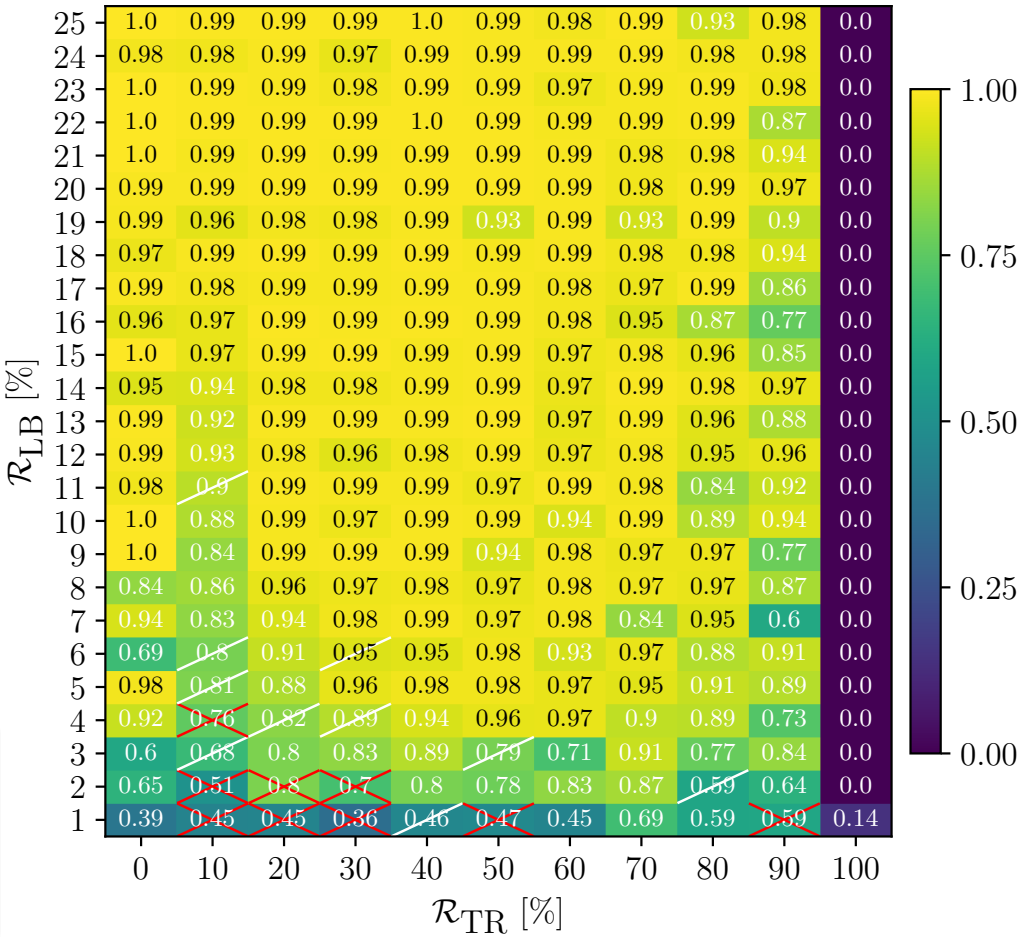
- Bhattacharyya coefficient

$$C_{\mathrm{B}}(x, r) = Z \exp\left[-\frac{x^2}{4(1+r^2)}\right]$$

$$\text{where} \quad Z = \sqrt{\frac{2r}{1+r^2}} \quad \text{and}$$

$$x = \frac{|\bar{Y}_{\mathrm{Orig}} - \bar{Y}_{\mathcal{P}1}|}{\sigma_{\mathrm{Orig}}}, \quad r = \frac{\sigma_{\mathcal{P}1}}{\sigma_{\mathrm{Orig}}}$$

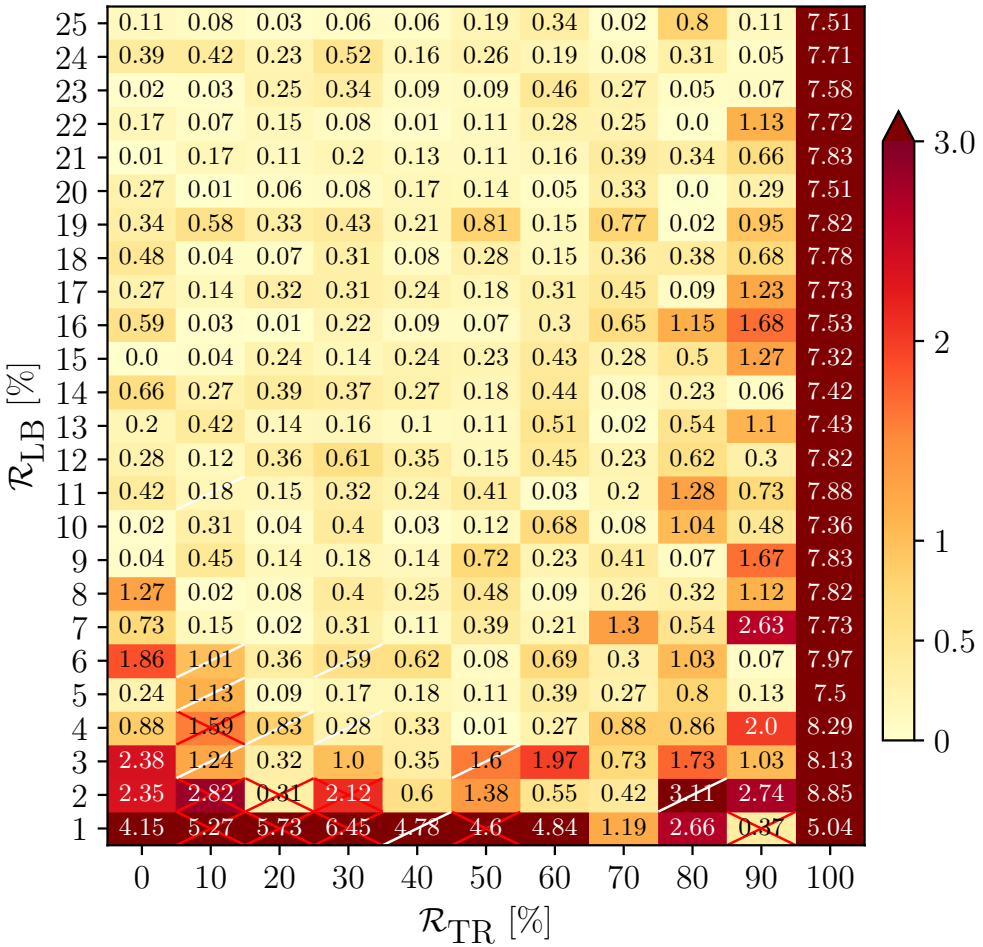
- **$C_{\mathrm{B}} \in [0, 1]$ (higher is better)**
  - **1: best overlap**
  - **0: no overlap**

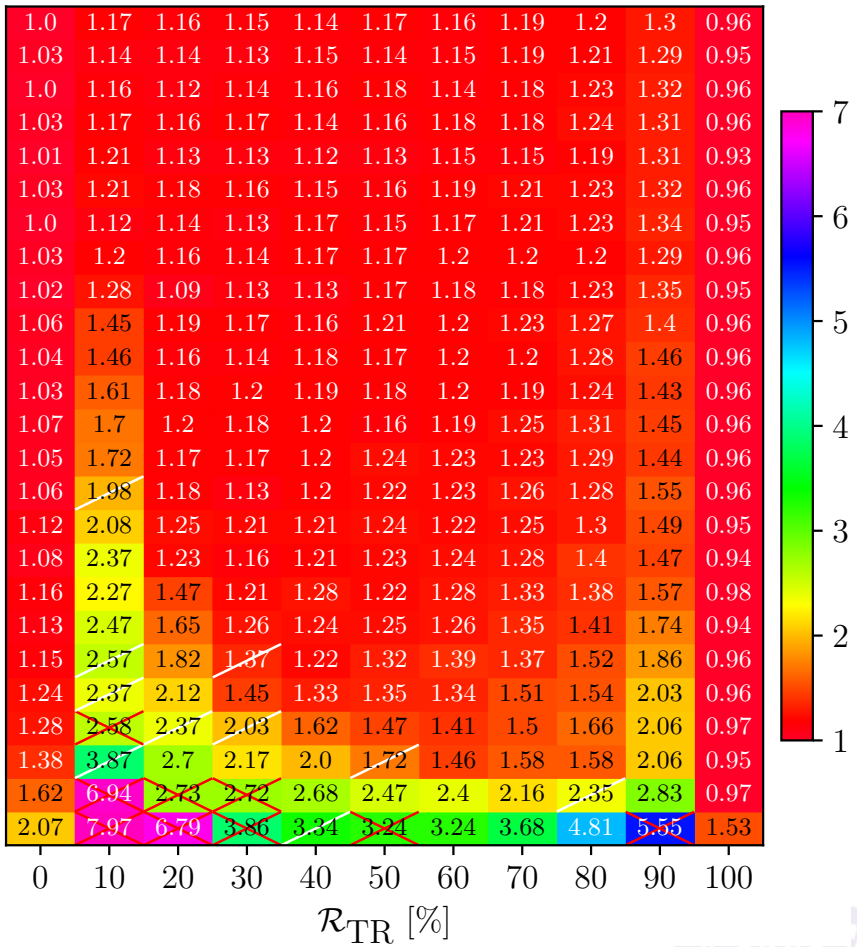


- Lattice setup: $12^3 \times 4$, $N_{\mathrm{f}} = 4$ Wilson clover quark, Iwasaki gluon with $\beta = 1.60$
- $\mathrm{Tr}\, M^{-n}$ $(n = 1, 2, 3, 4)$ trained with PLAQUETTE and RECTANGLE by LightGBM.
- Data from "Ohno *et al.* PoS **LATTICE2018** (2018) 174"

K with ML results ($\mathcal{P}1$) at the κ_t: $1\% \leq \mathcal{R}_{\mathrm{LB}} \leq 25\%$

- Left: x map of $C_{\mathrm{B}}(x, r)$; Right: r map of the same.

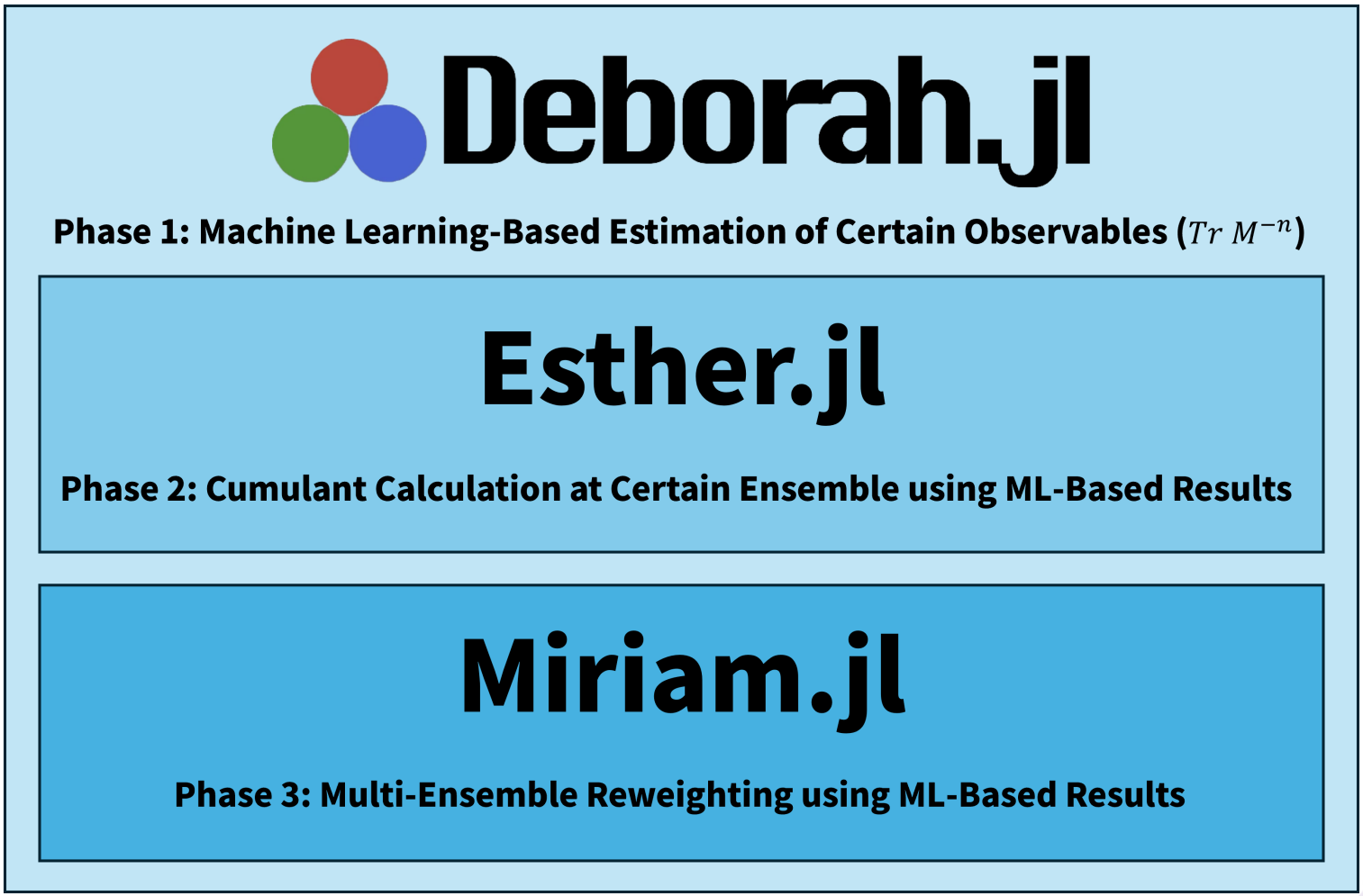




- When $x \equiv |\bar{Y}_{\mathrm{Orig}} - \bar{Y}_{\mathcal{P}1}|/\sigma_{\mathrm{Orig}} \approx 0$, the $\bar{Y}_{\mathcal{P}1}$ captures $\bar{Y}_{\mathrm{Orig}}$ **well**.
- When $r \equiv \sigma_{\mathcal{P}1}/\sigma_{\mathrm{Orig.}} \approx 1$, the $\sigma_{\mathcal{P}1}$ captures $\sigma_{\mathrm{Orig.}}$ **well**.

Introduction of Deborah.jl (1)

- We developed Deborah.jl for this work.



Introduction of Deborah.jl (2)

- Deborah.jl
  - `Deborah.jl` is an Estimation tool for Bias-cOrrected Regression Analysis with Heuristics.
  - Phase 1: Machine learning-based estimation of certain observables (In this work, $\mathrm{Tr}\, M^{-n}$, $n = 1, 2, 3, 4$)

- `Esther.jl`
  - `Esther.jl` is a Summary Tool for Higher-order cumulants through Estimation via Regression.
  - Phase 2: Cumulant calculation at certain ensemble using ML-based results

- `Miriam.jl`
  - MultI-Ensemble Reweighting & Interpolation Analysis with `Miriam.jl`
  - Phase 3: Multi-ensemble reweighting using ML-based results

What Deborah.jl can do (1)

- Flexible selection of features
  - Let X represent the input observable(s), and Y the output observable.
  - In machine learning terminology, X corresponds to the feature(s).
  - Deborah.jl supports flexible feature selection, allowing users **to choose the number of input features as desired**.
  - *ex)* You can do $\{X_1\,,\ X_2\,,\ X_3\,,\ X_4\,,\ X_5\,,\ X_6\,,\ X_7\} \to Y$ learning.

What Deborah.jl can do (2)

- Freely choose among various regression models

1. Baseline: Assume $X = Y = Y^P$, and perform only data splitting into subsets.
2. Random: Intentionally generate Gaussian noise as データラメ ($\mu = 0$, $\sigma \approx$ jackknife error of original Y)
3. Lasso and Ridge regression via `JuliaAI/MLJ.jl`
4. LightGBM(Microsoft) via `JuliaAI/MLJ.jl` or `PyCall.jl`
5. LightGBM(Microsoft) via `JuliaAI/MLJ.jl` with additional hyperparameter tuning (I call this the MiddleGBM mode.)
   - `num_iterations`: Total number of boosting rounds (controls overall model complexity) $\rightarrow 2 \leq n_{\text{iter}} \leq 100$
   - `learning_rate`: Shrinks the contribution of each tree (lower = slower but safer learning) $\rightarrow 0.1 \leq \alpha \leq 1.0$
   - `min_data_in_leaf`: Minimum number of samples in a leaf node (prevents overfitting) $\rightarrow 2 \leq n_{\text{data}} \leq 20$

What `Esther.jl` and `Miriam.jl` can do

- `Esther.jl` collects ML-estimated $\mathrm{Tr}\, M^{-n}$ $(n = 1, 2, 3, 4)$, and computes cumulants such as the quark condensate, susceptibility, skewness, and kurtosis for a single ensemble.

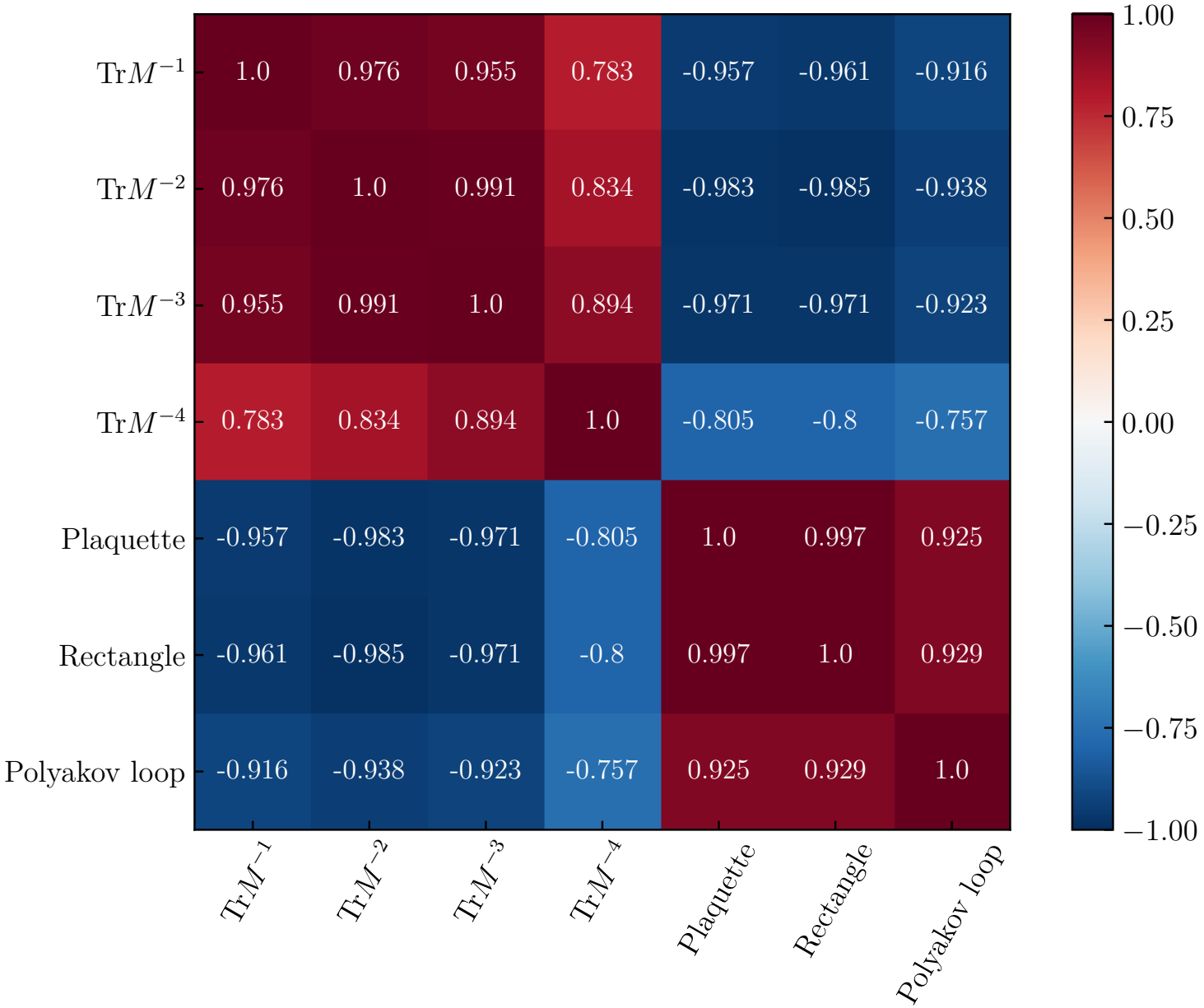
- `Miriam.jl` aggregates $\mathrm{Tr}\, M^{-n}$ $(n = 1, 2, 3, 4)$ data across multiple ensembles, enabling per-ensemble cumulant analysis and multi-ensemble reweighting by concatenating the results.

Measurement Information

ID	$N_S^3 \times N_T$	κ	N
`L12T4b1.60k13575`	$12^3 \times 4$	0.13575	20000
`L12T4b1.60k13577`	$12^3 \times 4$	0.13577	20000
`L12T4b1.60k13580`	$12^3 \times 4$	0.13580	20000
`L12T4b1.60k13582`	$12^3 \times 4$	0.13582	20000
`L12T4b1.60k13585`	$12^3 \times 4$	0.13585	20000

- Ohno *et al.* PoS **LATTICE2018** (2018) 174
- HW: `Oakforest-PACS` system (Boku *et al.* arXiv:1709.08785)
- SW: `BQCD` program (Haar *et al.* EPJWoC **175** 14011 (LAT2017))
- $N_f = 4$ Wilson clover quark action, Iwasaki gauge action

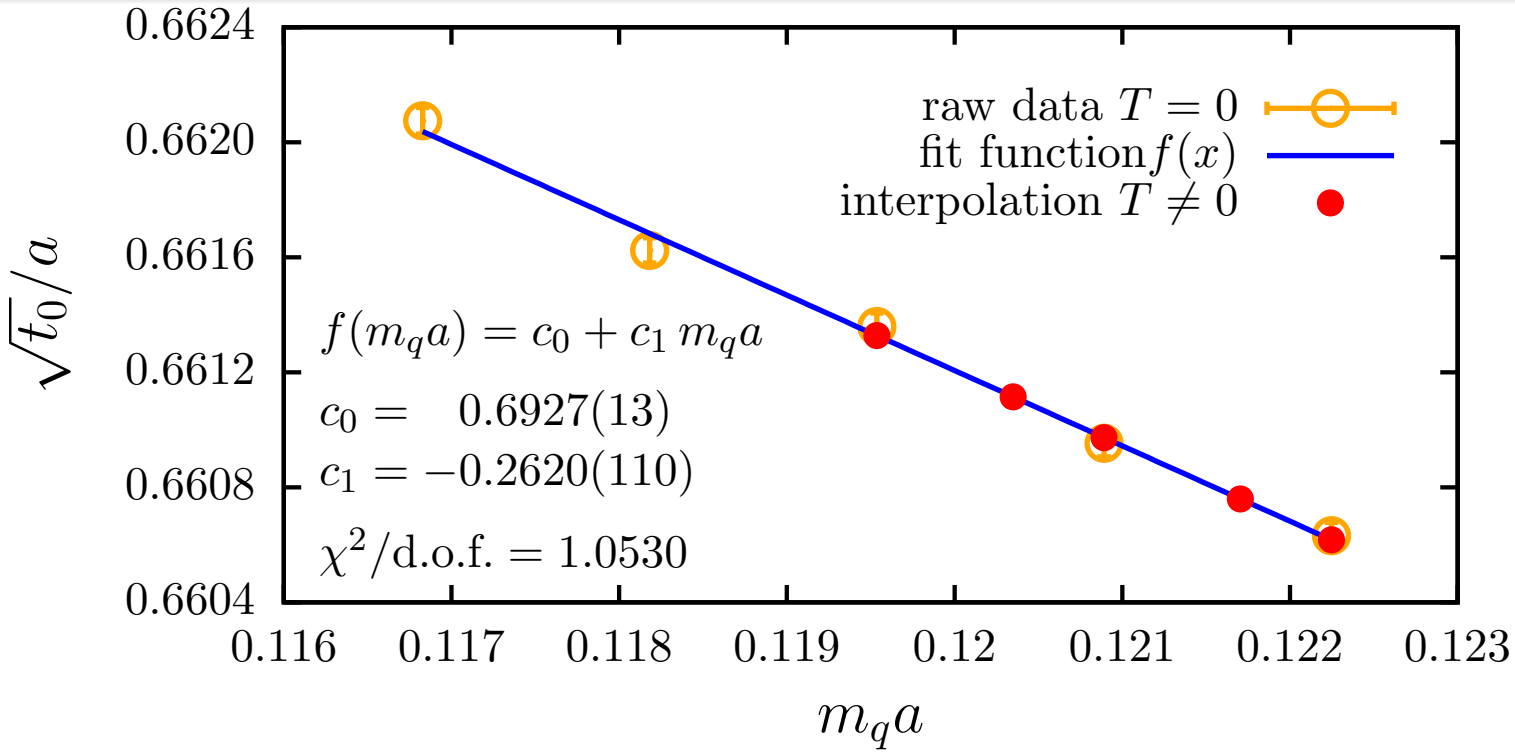
Correlation map between observables (L12T4b1.60k13575)



Why $N_f = 4$?

- In earlier $N_f = 3$ studies, the critical pion mass differed between staggered and Wilson quarks, leaving the CEP location uncertain.
- $N_f = 4$ still expects a first-order chiral transition, with possibly larger pion mass at the critical endpoint ➡ lower cost.
- Also avoids the rooted-determinant issue, enabling a clean comparison between staggered and Wilson types.

Fit result of Wilson flow scale parameter at zero T



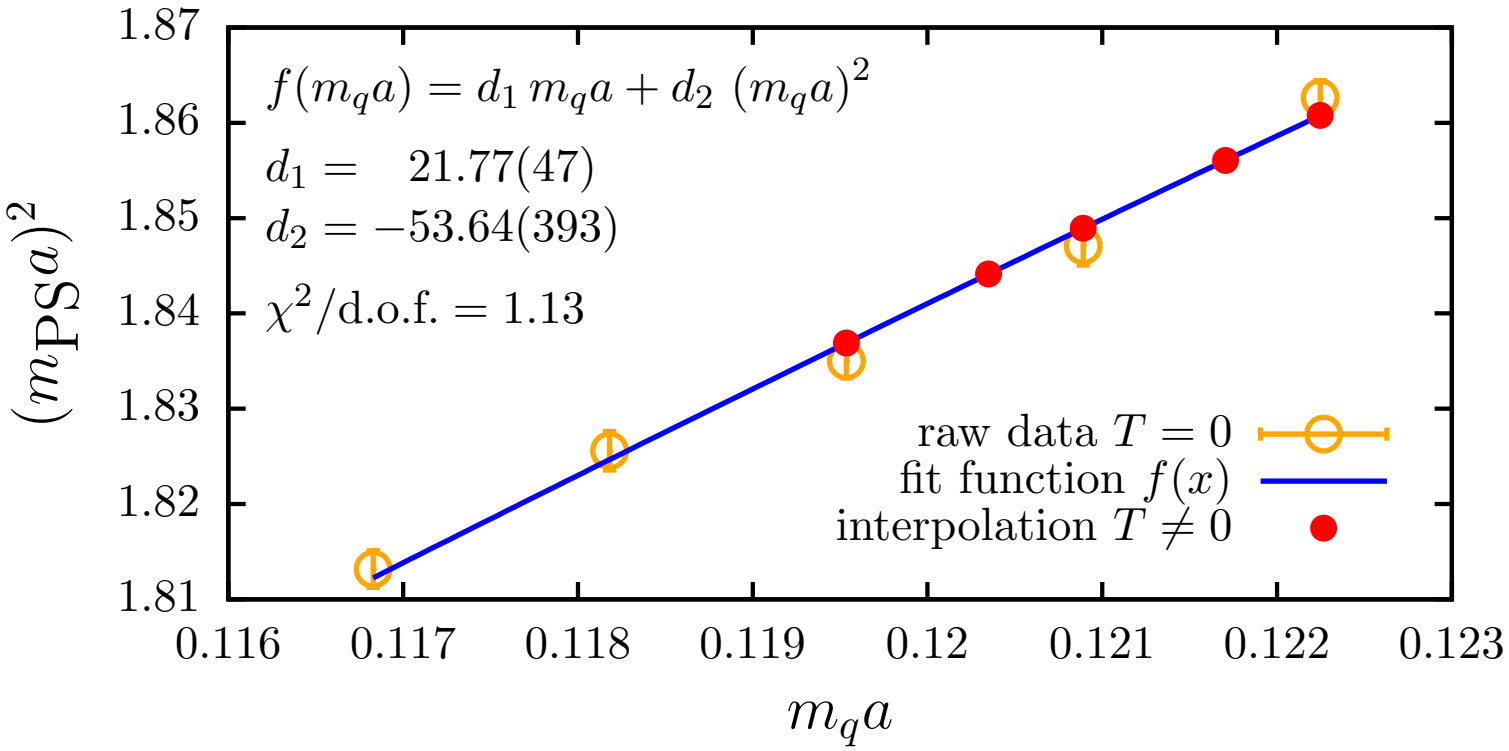
- $a \approx 0.22175$ fm $(\kappa = 0.13575)$
- $a \approx 0.22171$ fm $(\kappa = 0.13577)$
- $a \approx 0.22163$ fm $(\kappa = 0.13580)$
- $a \approx 0.22159$ fm $(\kappa = 0.13582)$
- $a \approx 0.22152$ fm $(\kappa = 0.13585)$

- $V = 16^3 \times 32$, $\beta = 1.60$
- Here, we use

$$\frac{1}{\sqrt{t_0}} = 1.347(30)\ \text{GeV}$$

BMW, JHEP **09** 010 (2012).

Fit result of pseudoscalar meson mass at zero T



- $m_{\mathrm{PS}} \approx 1.214$ GeV $(\kappa = 0.13575)$
- $m_{\mathrm{PS}} \approx 1.213$ GeV $(\kappa = 0.13577)$
- $m_{\mathrm{PS}} \approx 1.211$ GeV $(\kappa = 0.13580)$
- $m_{\mathrm{PS}} \approx 1.209$ GeV $(\kappa = 0.13582)$
- $m_{\mathrm{PS}} \approx 1.207$ GeV $(\kappa = 0.13585)$

- $V = 16^3 \times 32$, $\beta = 1.60$
- In summary, we have

$$m_{\mathrm{PS}} \approx 1.21 \text{ GeV}$$

in these 5 datasets.

Quark mass

- We obtain κ_c (kappa critical) from

$$\kappa_c(g_0^2) = 0.125 + 0.003681192\, g_0^2 + 0.000117\, (g_0^2)^2 + 0.000048\, (g_0^2)^3 - 0.000013\, (g_0^2)^4$$

where $g_0^2 = \dfrac{2N_c}{\beta}$ is the coupling constant.

- With $\beta = 1.60$, we have $\kappa_c \approx 0.14041$.

- We obtain quark mass $m_q a$ with

$$m_q a = \frac{1}{2}\left(\frac{1}{\kappa} - \frac{1}{\kappa_c}\right)$$

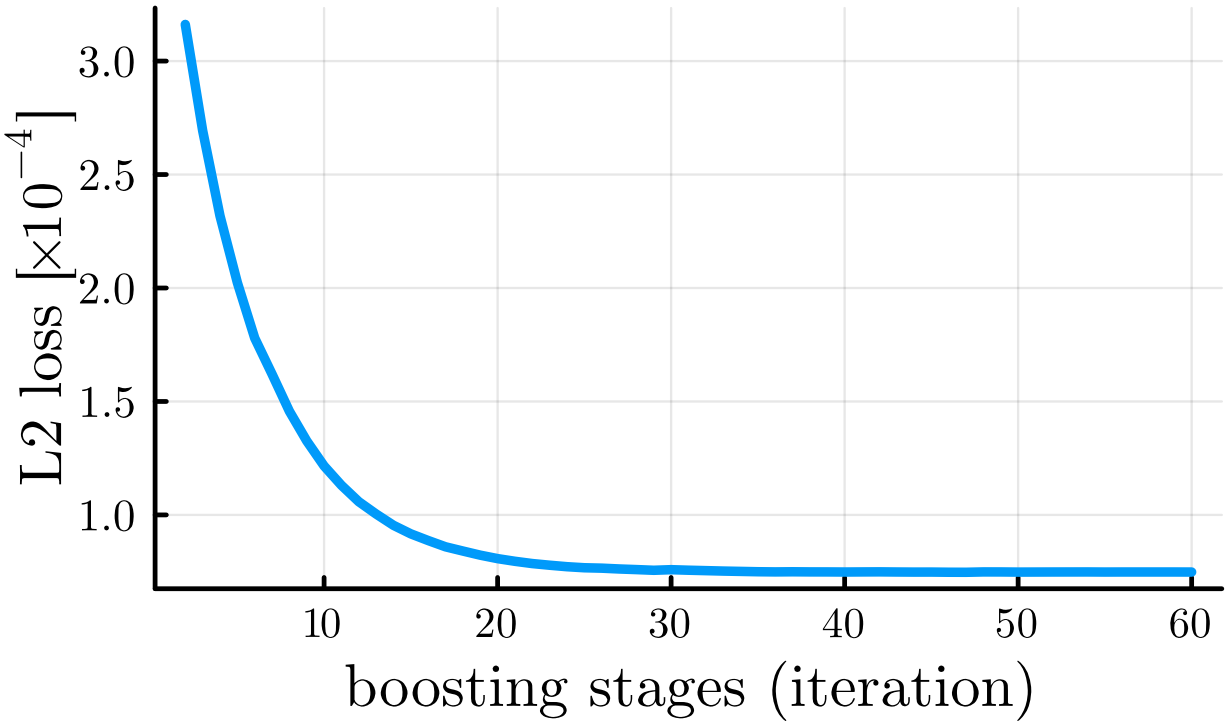
Example of computational gain

- To give one concrete example from meas. logs of the `BQCD` program,

κ	Total [h]	All CG [h]	1 Tr M^{-n} CG [h]
0.13575	65.91	59.12	3.94
0.13577	65.97	59.15	3.94
0.13580	65.51	58.66	3.91
0.13582	64.44	57.56	3.84
0.13585	63.35	56.58	3.77

  - $V = 16^3 \times 4$, $N_{\mathrm{conf}} = 5500$
  - For a single gauge conf., CG is called 150 times, where $N_{\mathrm{src}} = 10$.
  - For a *single* determination of Tr M^{-n}, it takes about 4 hours.
- For the GBDT regression on Deborah.jl (on a normal laptop),
  - Time for model training: $\approx$ 3 seconds
  - Time for $N_{\mathrm{BS}} = 1000$ bootstrap resampling: $\approx$ 2 seconds
  - ➡ negligible compared with CG time on supercomputer.

Optimal boosting stage

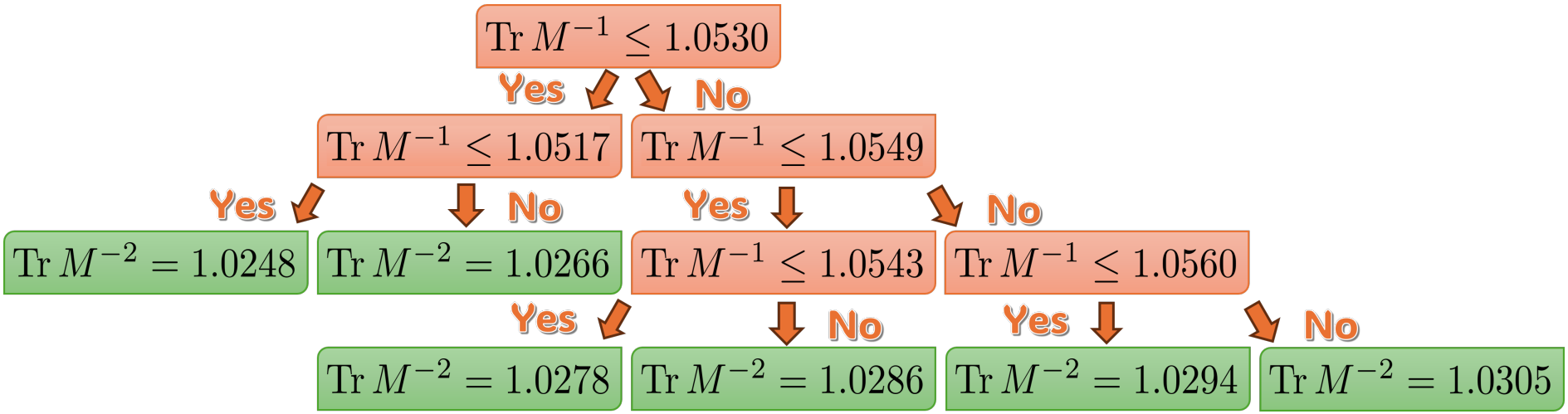


- x-axis: boosting stage (number of iteration)
- y-axis: L2-loss for machine learning $X \to Y$

$$(\text{L2 loss}) \equiv \sum_{i=1}^{N_{\text{TR}}} \left(Y_i - Y_i^P\right)^2$$

- optimal boosting stage $\approx$ **40**

Visualization of decision tree



- An example with $X = \mathrm{Tr}\, M^{-1}$, $Y = \mathrm{Tr}\, M^{-2}$ for $X \to Y$ learning
- Example for a boost stage.
- Depth of tree $= 3$
- Number of leaf (green cell, $Y = \mathrm{Tr}\, M^{-2}$) $= 6$