KAHLERT SCHOOL OF COMPUTING
THE UNIVERSITY OF UTAH

www.sci.utah.edu

Standard Gaussian Process is All You Need for High-Dimensional Bayesian Optimization

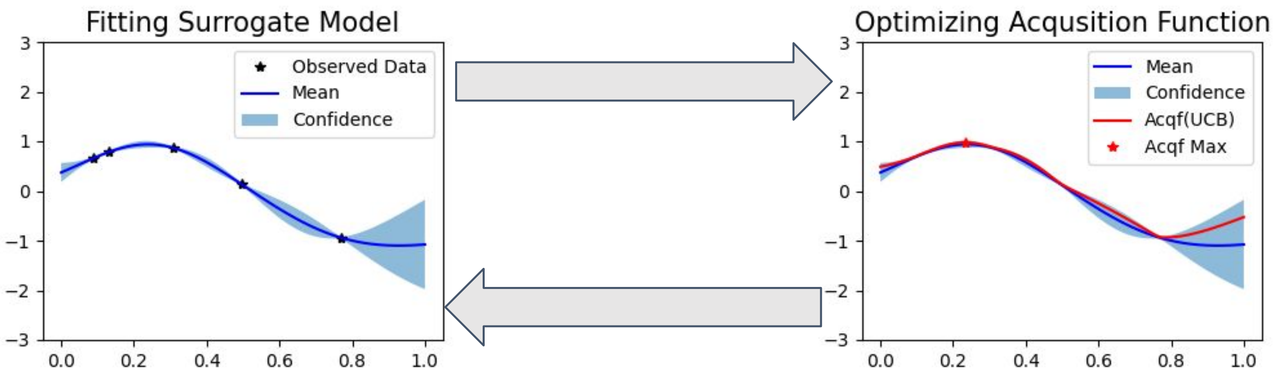
Zhitong Xu, Haitao Wang, Jeff M. Phillips, Shandian Zhe

Kahlert School of Computing, The University of Utah

Bayesian Optimization: blackbox functions

Bayesian optimization(BO): A powerful tool to optimize a **black box function**.

- Fitting a **surrogate model,** typically GP.
- Optimizing an **acquisition function**.



Challenge: High Dimensional

Common belief: BO with standard GP(a.k.a, Standard BO) is limited to low-dimensional problems(d≤20).

— [Frazier, 2018], [Nayebi et al., 2019], [Eriksson & Jankowiak, 2021], [Moriconi et al., 2020], [Letham et al., 2020], [Wang et al., 2016], [Li et al., 2016]

Current HDBO methods: Imposing strong structural assumption

- Structural Assumption in *Functional Space*
  - $f(x) = \sum_{j=1}^{M} f_j(x^j), f_j \sim \mathcal{GP}\left(m_j(x^j), \kappa_j(\cdot,\cdot)\right)$
  - Challenges: Learning the decomposition is hard in high-dimensional spaces.
- Structural Assumption in *Input Space*
  - $f(\mathbf{x}) \approx \mathbf{f_d}(\mathbf{Tx}), \forall \mathbf{x} \in \mathbb{R}^{\mathbf{D}}$
  - Challenges: Low dimensional embedding might not contain the **global optimum**.

Something is missing

1. How standard Bayesian Optimization perform in high-dimensional cases?
2. Why standard Bayesian optimization fail on high-dimensional setting?

Bayesian optimization with standard Gaussian process.

How Standard Bayesian Optimization perform in high-dimensional cases?

Lack sufficient **empirical evidence** that SBO really fails …

- GP(RBF) BO performance: **REMBO**[Wang et al., 2016], **BNNBO**[Li et al., 2024].
- GP(RBF) fitting: **SAASBO**[Eriksson & Jankowiak, 2021], **ALEBO**[Letham et al., 2020].
- **No empirical results** on BO performance with **ARD Matérn**.

Why Standard Bayesian optimization fail on high-dimensional setting?

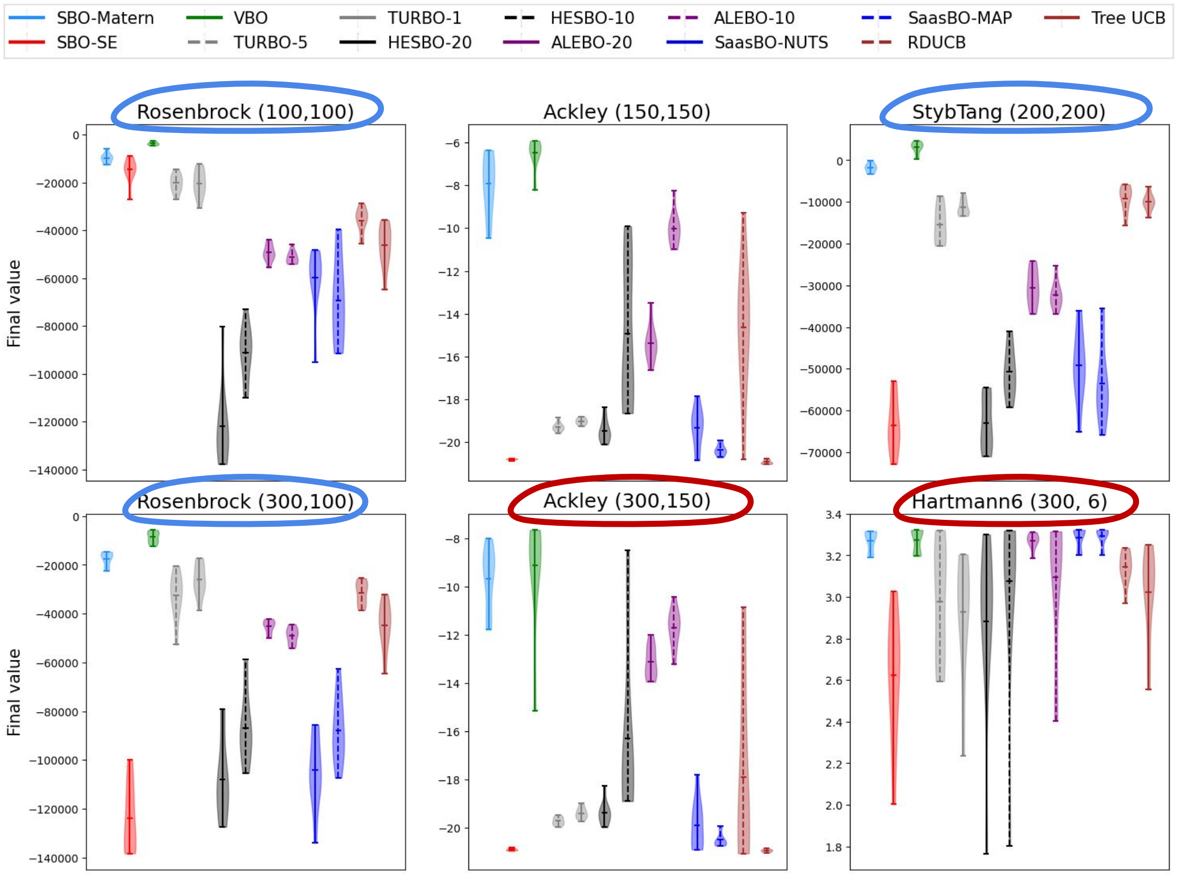
No specific analysis on what cause SBO fails on high-dimensional setting

…

Curse of dimensionality: [Binois et al., 2022]

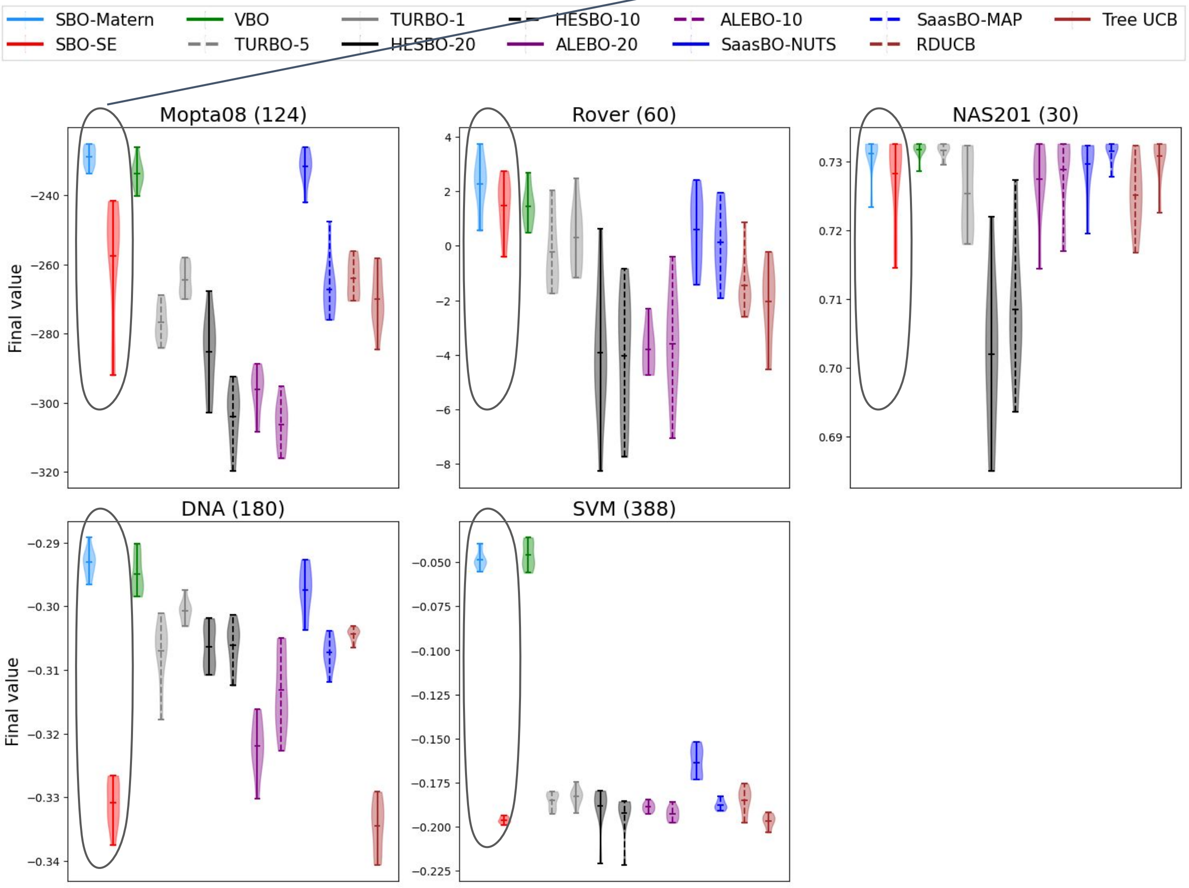
Standard BO Results(***Maximization***, higher the better)

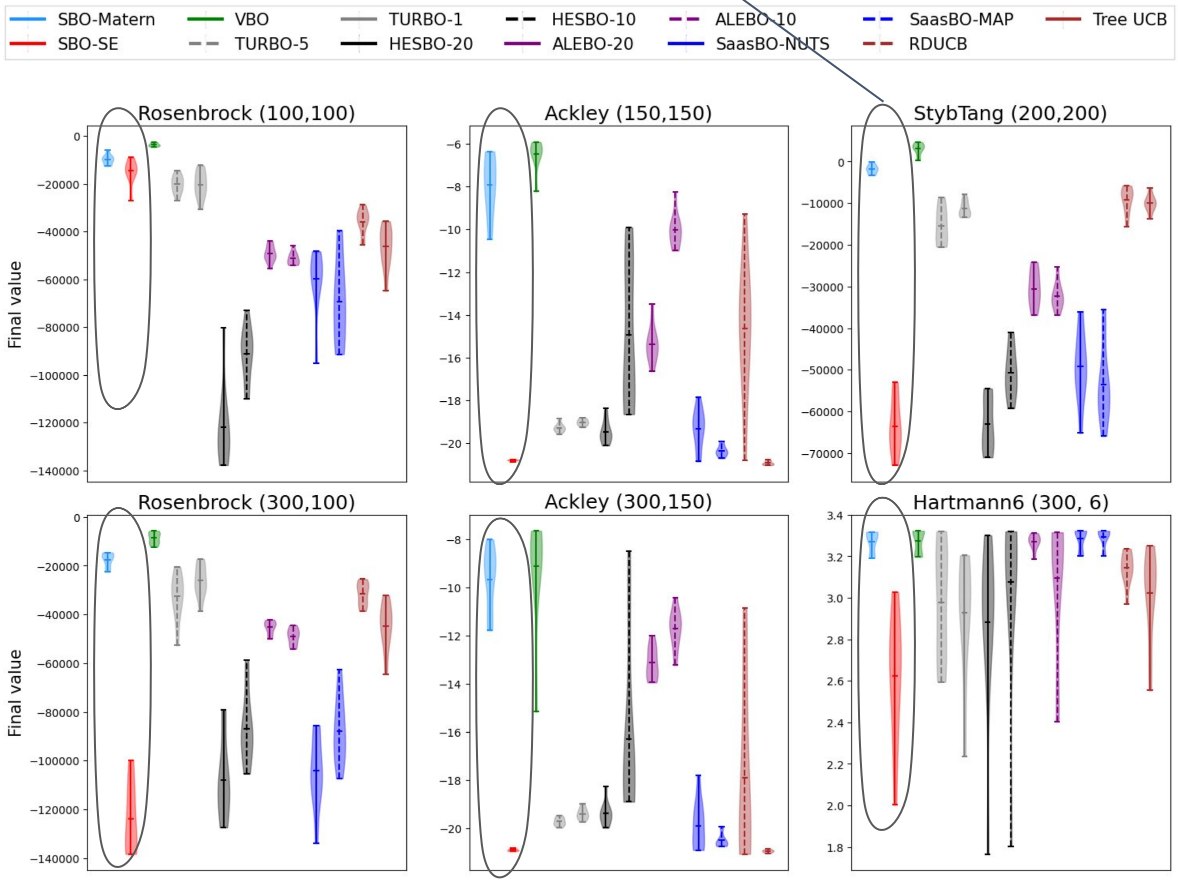
Blue: Contains additive structure.
Red: Exists low dimensional embedding.



Standard BO Results(***Maximization***, higher the better)

Standard BO performance(left: Matérn, right: SE)





Observation from Empirical results

- (SBO) Matern Kernel works among the on nearly all benchmarks.
- (SBO) Squared Exponential kernel **fail hard** on d>150.

Why?

We discovered: Gradient Vanishing in GP training, is the failure mode.

Lengthscale gradient in MLE training(or MAP with Uniform Prior) when fitting GP:

$$\frac{\partial \log p(\mathbf{y}|\mathbf{X})}{\partial \ell_k} = \frac{1}{2}\mathrm{tr}\left(\mathbf{A} \cdot \frac{\partial \mathbf{K}}{\partial \ell_{\mathbf{k}}}\right)$$

where $\mathbf{A} = \boldsymbol{\alpha}\boldsymbol{\alpha}^\top - (\mathbf{K} + \sigma^2\mathbf{I})^{-1}$ and $\boldsymbol{\alpha} = (\mathbf{K} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m})$.

With constant lengthscale initialization, $\ell_1 = \ldots = \ell_d = \ell_0$ at the beginning of fitting GP. (eg. GPyTorch initialize to Softplus(0.)=0.693)

Gradient term

Derivative term: $\frac{\partial \mathbf{K}}{\partial \ell_k}$

This term is data dependent.

- **SE Kernel :** $\kappa_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \mathbf{a} \exp(-\rho^2)$

$$\left[\frac{\partial \mathbf{K}}{\partial \ell_k}\right]_{ij} = \frac{\partial \kappa_{\text{SE}}(\mathbf{x_i}, \mathbf{x_j})}{\partial \ell_k} = \frac{2a}{\ell_k} \exp(-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{\ell_k^2}) \frac{(x_{ik} - x_{jk})^2}{\ell_k^2} \leq \frac{2a}{\ell_k} \cdot \frac{\rho^2}{e^{\rho^2}}$$

- **Matérn Kernel :** $\kappa_{\text{Matérn}}(\mathbf{x}, \mathbf{x}') = \mathbf{a} \left(\mathbf{1} + \sqrt{\mathbf{5}}\rho + \mathbf{5}\rho^2/\mathbf{3}\right) \exp\left(-\sqrt{\mathbf{5}}\rho\right)$

$$\frac{\partial \kappa_{\text{Matérn}}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \ell_k} = \frac{5}{3}(1 + \sqrt{5}\rho) \exp(-\sqrt{5}\rho) \frac{(x_{ik} - x_{jk})^2}{\ell_k^3} \leq \frac{5(1 + \sqrt{5})\rho}{3\ell_k^3} \cdot \frac{\rho^2}{e^{\sqrt{5}\rho}}$$

grows way slower compared to denominator in SE kernel,

Here, $\rho = \frac{\|\mathbf{x_i} - \mathbf{x_j}\|}{\ell_0}$ and i≠j.

Machine Epsilon Threshold $(\epsilon = 2^{-53})$

Proposition 4.1:

$$Given\ any\ \xi > 0, \frac{\rho^2}{e^{\rho^2}} < \xi\ when\ \rho > \tau_{SE} = \frac{1}{2} + \sqrt{\frac{1}{4} - \log \xi}$$

Proposition 4.3:

$$Given\ \forall \xi > 0, \frac{\rho^2}{e^{\sqrt{5}\rho}} < \xi\ when\ \rho > \tau_{Matérn} = \left(1 + \sqrt{1 + \log 1/5 - \log \xi}\right)^2 / \sqrt{5}$$

With machine epsilon $\epsilon = 2^{-53}$, the threshold for SE kernel is $\tau_{\mathrm{SE}} = 6.58$ and for Matérn kernel is $\tau_{\mathrm{Matérn}} = 21.98$.

Probabilistic Lower Bound (Constant initialization)

Lemma 4.2: Suppose the input domain is $[0, 1]^d$ and the input vectors are sampled independently from the uniform distribution, then for any constant threshold $\boldsymbol{\tau}$>0, when $d > 6\ell_0^2\tau^2$,

For SE Kernel, and Softplus(0.0) initialization:
$6\ell_0^2\tau^2$ = 124.7

$$p(\rho \geq \tau) > 1 - 2\exp\left(-\frac{(d - 6\ell_0^2\tau^2)^2}{18d}\right)$$

Probabilistic Lower Bound (Constant initialization)

Lower bound	0.95	0.99	0.995	0.999	0.9995	0.9999
SE (d)	172	205	219	250	264	294
Matérn (d)	980	1040	1064	1116	1137	1185

Probability of gradient vanishing, with $\ell_0 = 0.5$ and d.

Our fix to the failure mode

Basic Idea: Let's break the prerequisite in previous lower bound, $d > 6\ell_0^2\tau^2$.

Our proposed lengthscale initialization is:

$$\ell_0 = c\sqrt{d}, \quad c > 0.$$

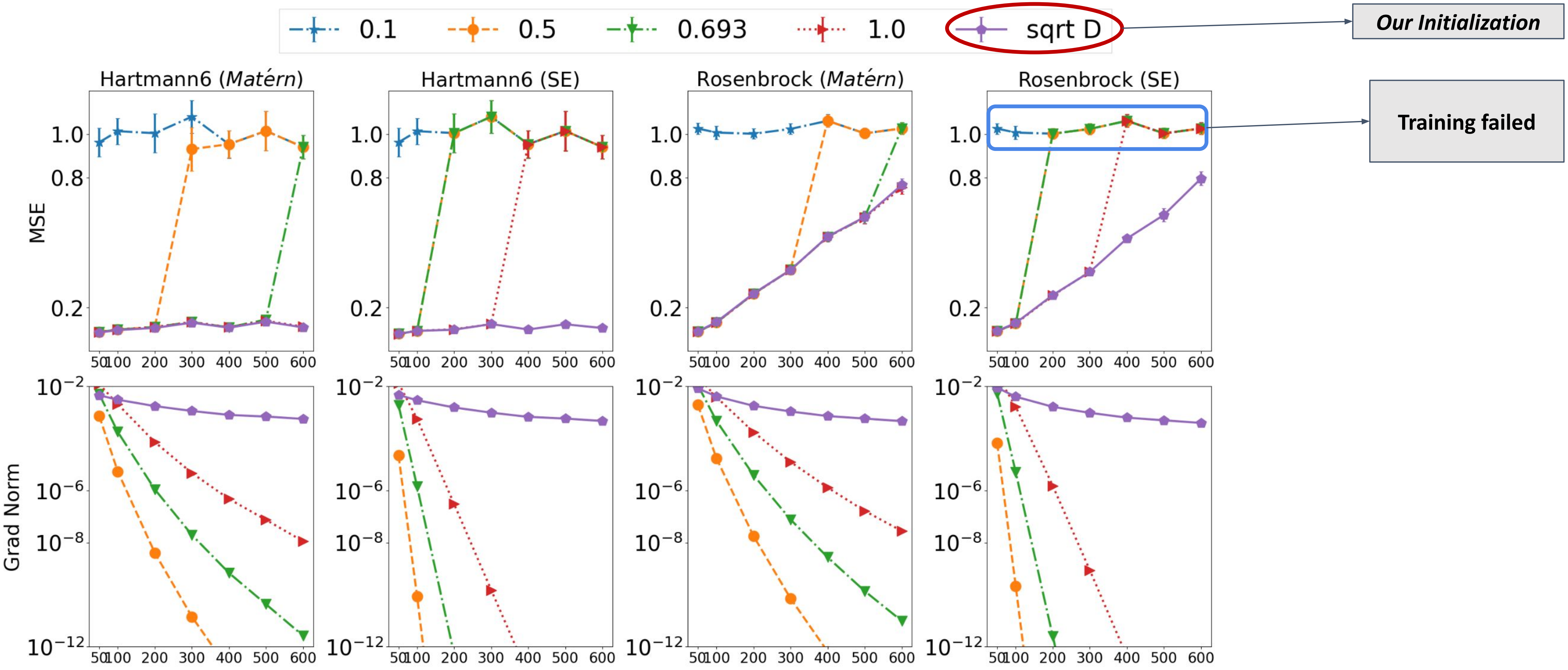
Guarantee (Our Initialization)

Lemma 5.1: Suppose the input domain is $[0,1]^d$ and each input vector is independently sampled from uniform distribution. Given any constant threshold $\boldsymbol{\tau}$>0, we set $\ell_0 = c\sqrt{d}$ such that $c > \frac{1}{\sqrt{6}\tau}$, then

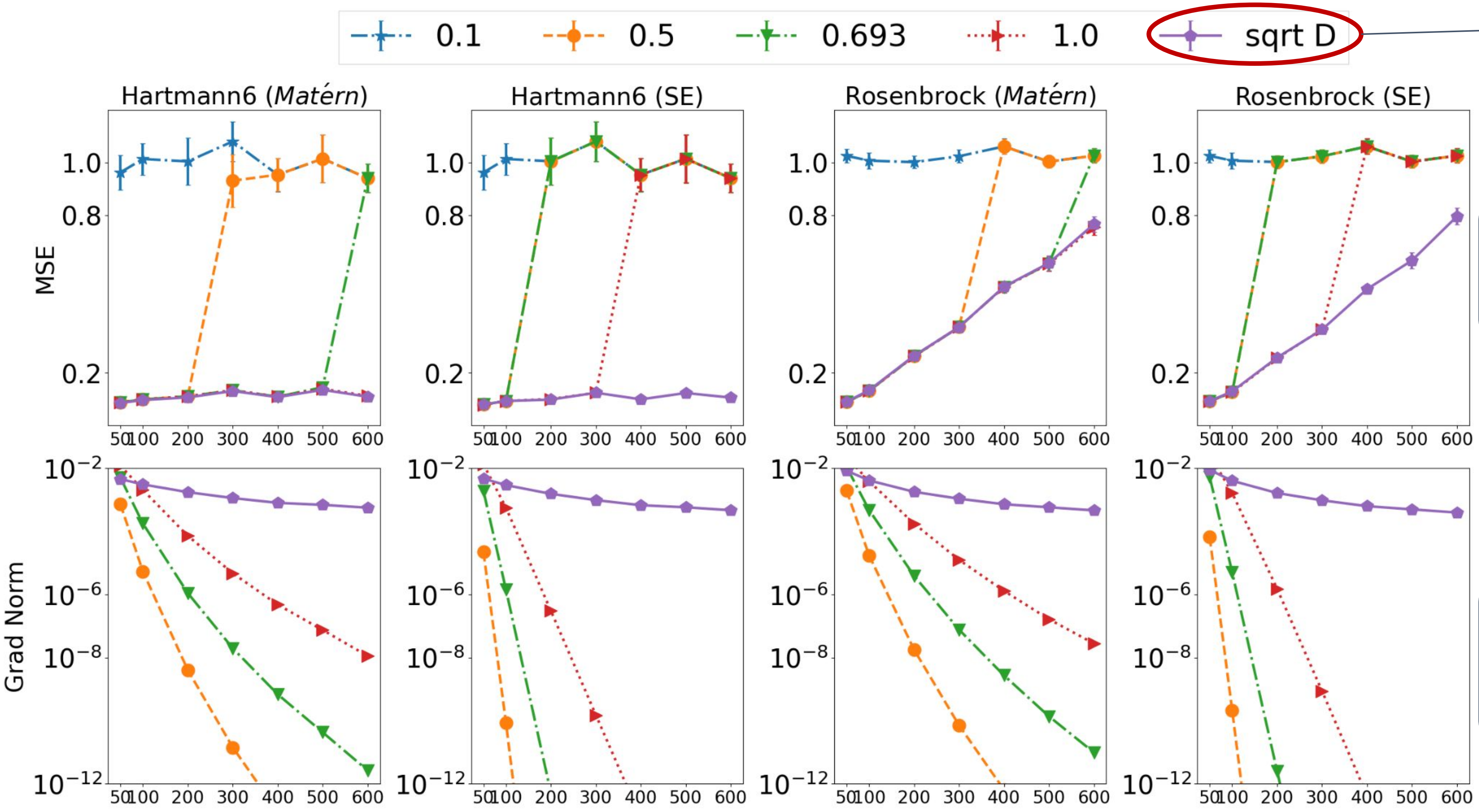
For SE kernel: c > 0.06.

$$p(\rho \geq \tau) \leq 2\exp\left(-2(c^2\tau^2 - \frac{1}{6})^2 d\right) \propto \exp(-\mathcal{O}(d)).$$

Numerical verification (GP)



Numerical verification (GP)

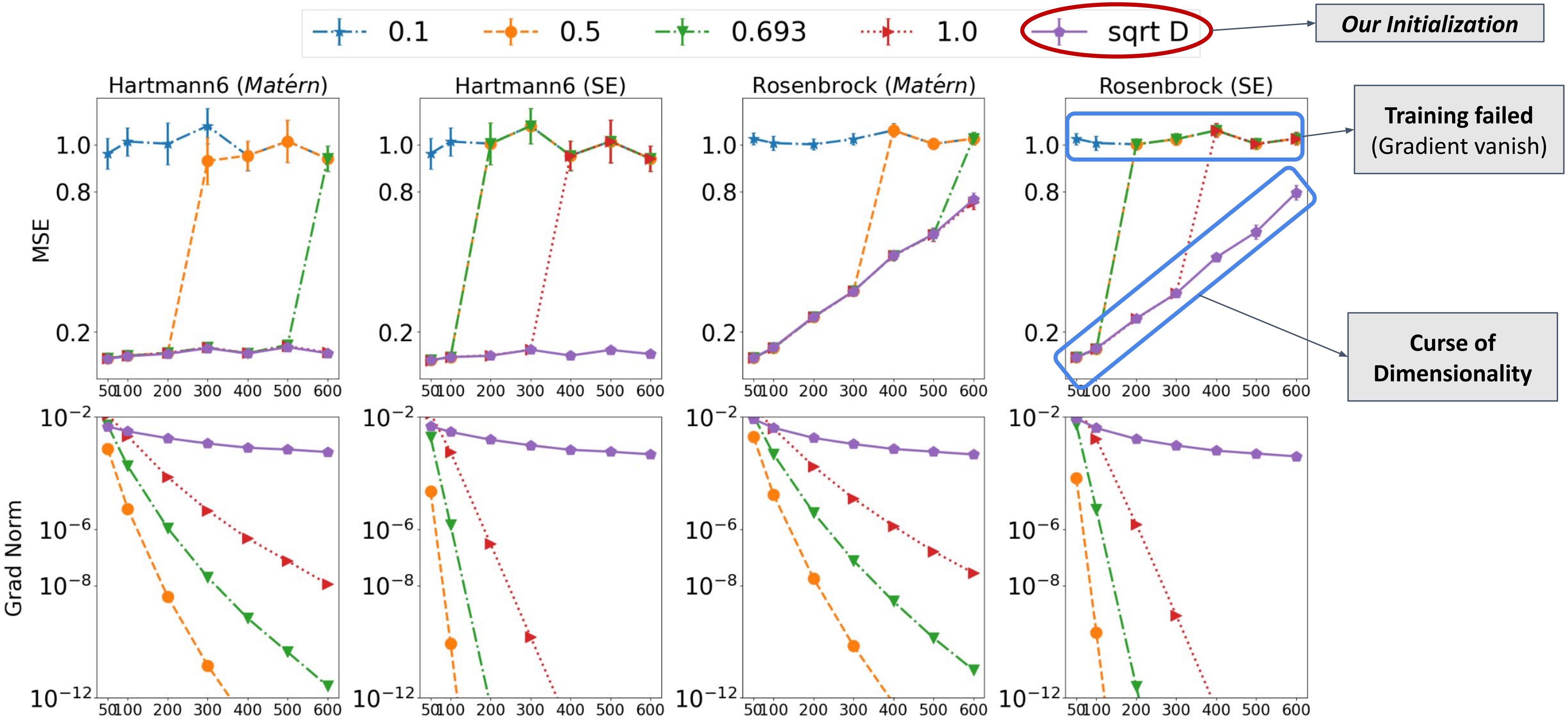


Our Initialization

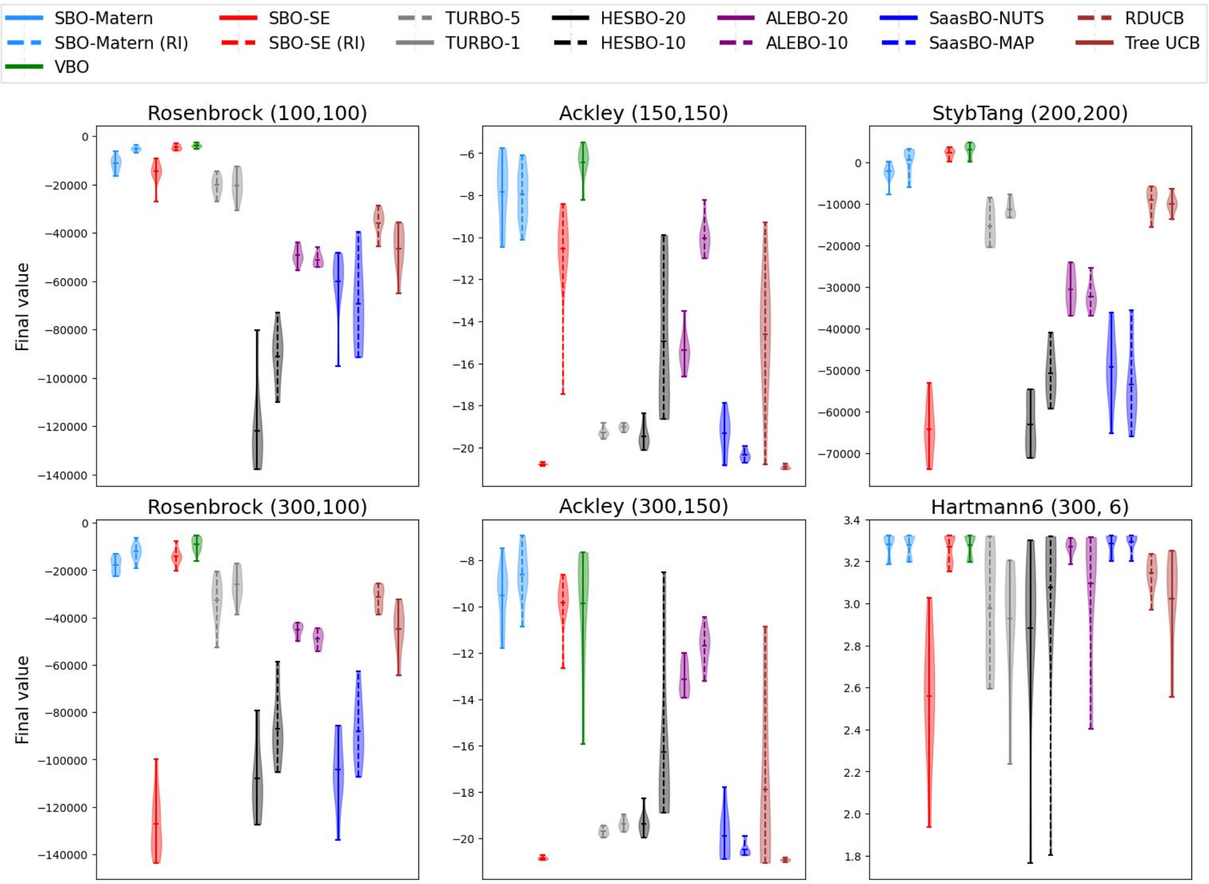
OBS: As d increase, MSE jumps to 1.0 for any constant initialization.

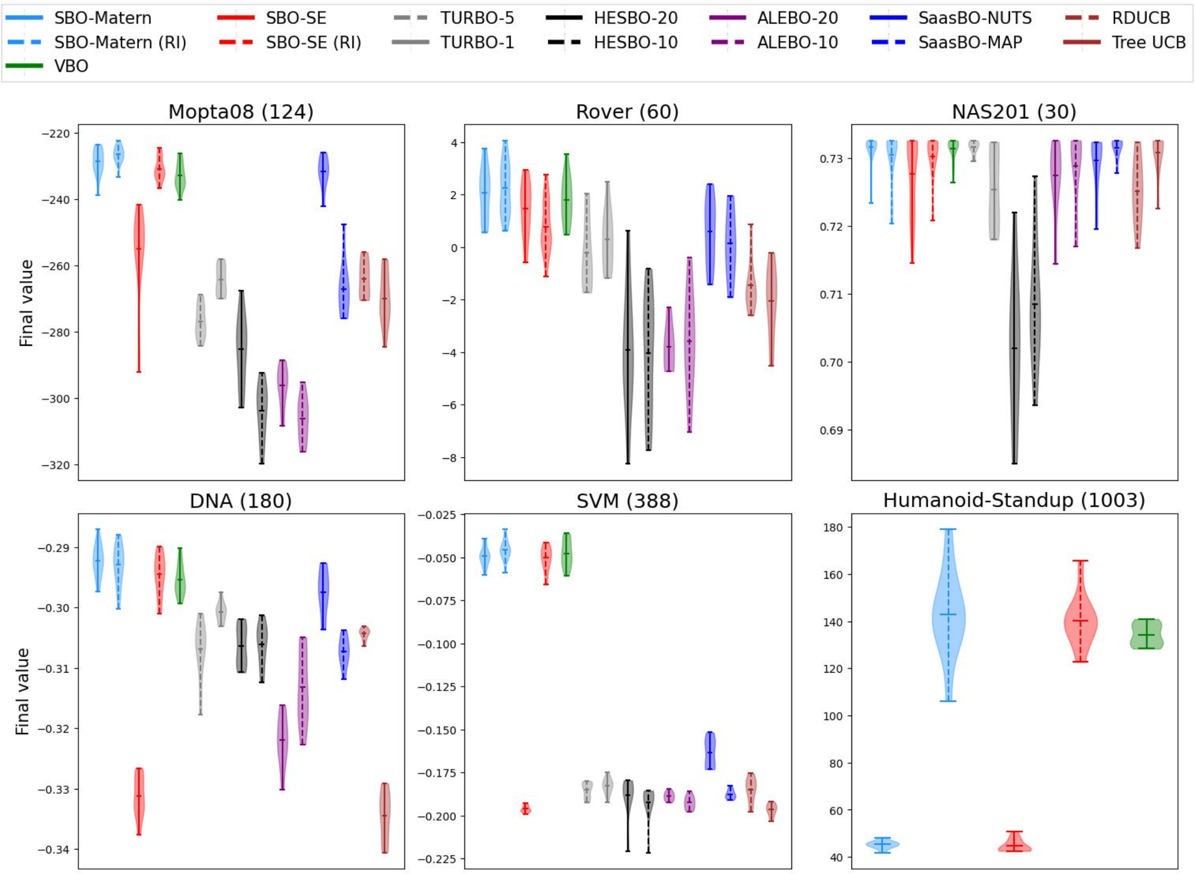
OBS: As d increase, gradient value decreases significantly, for any constant initialization.

Numerical verification(GP)



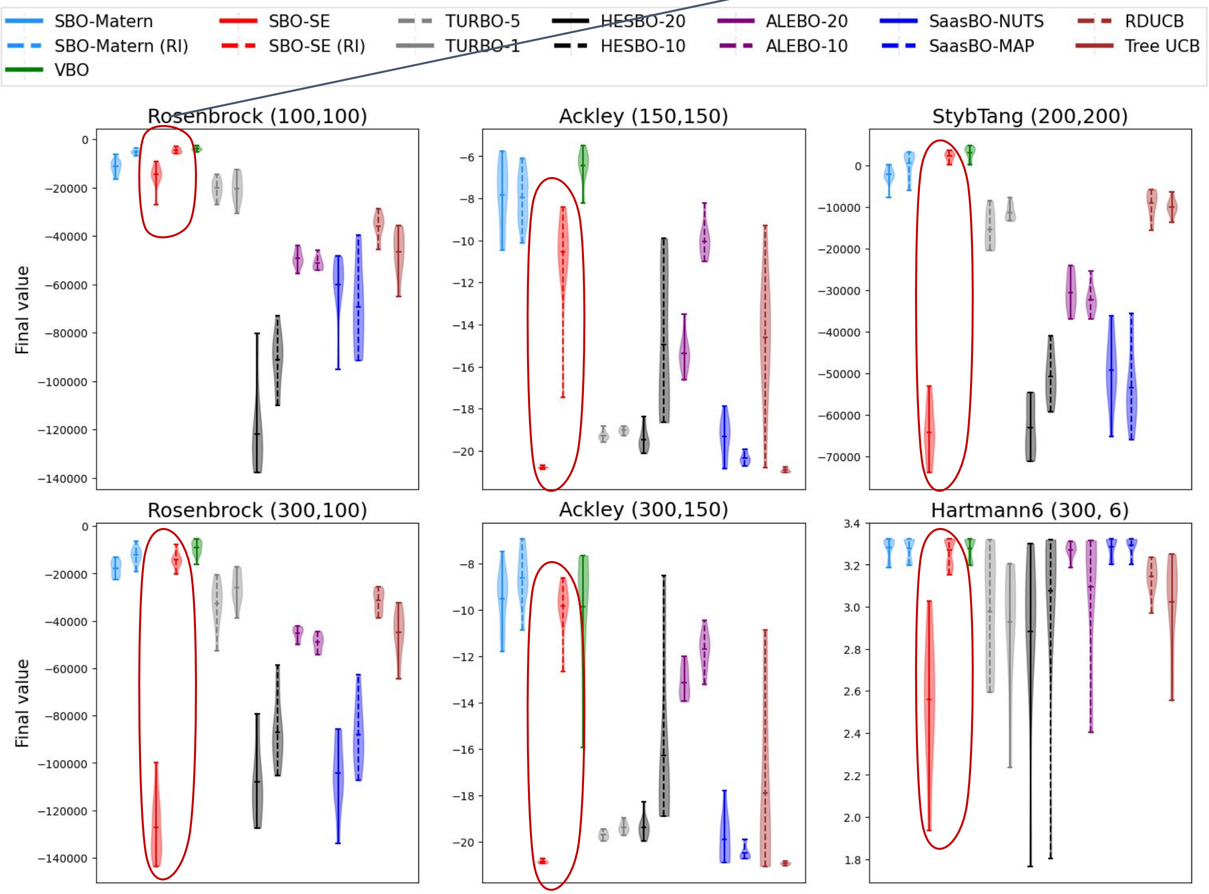
Standard BO performance with our Initialization (Violin)

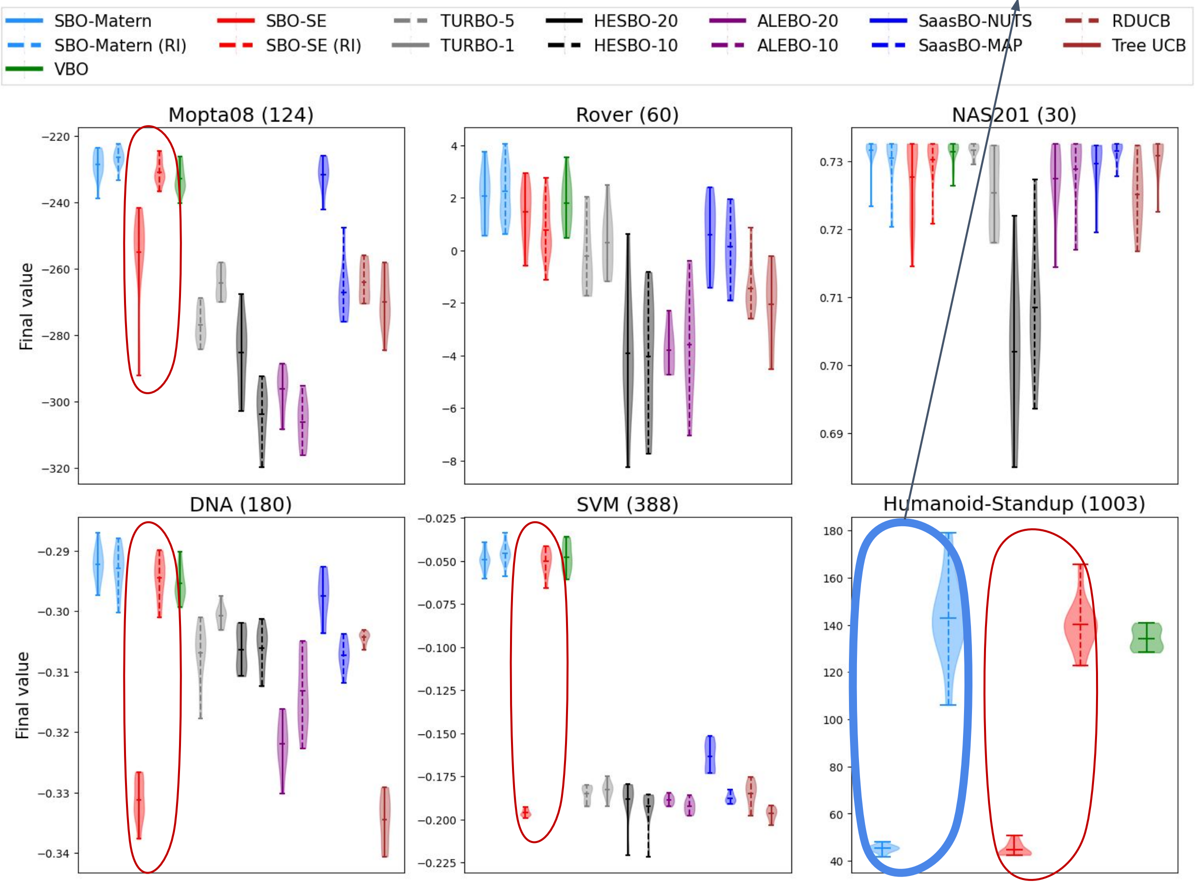




Standard BO performance with our initialization

Standard BO performance(**left**: ℓ_0 = 0.693, **right**: ℓ_0 = Sqrt(d)). **Blue**: Matérn, **Red**: SE.





Conclusion and open questions

- Draw attention to Standard GP based method.

- Other failure modes with Standard BO?
- Can we do better?

Q&A

Thank you, Welcome to our poster !