

# SACLA/SPring-8 におけるベイズ 最適化の効率化, 高度化の試み

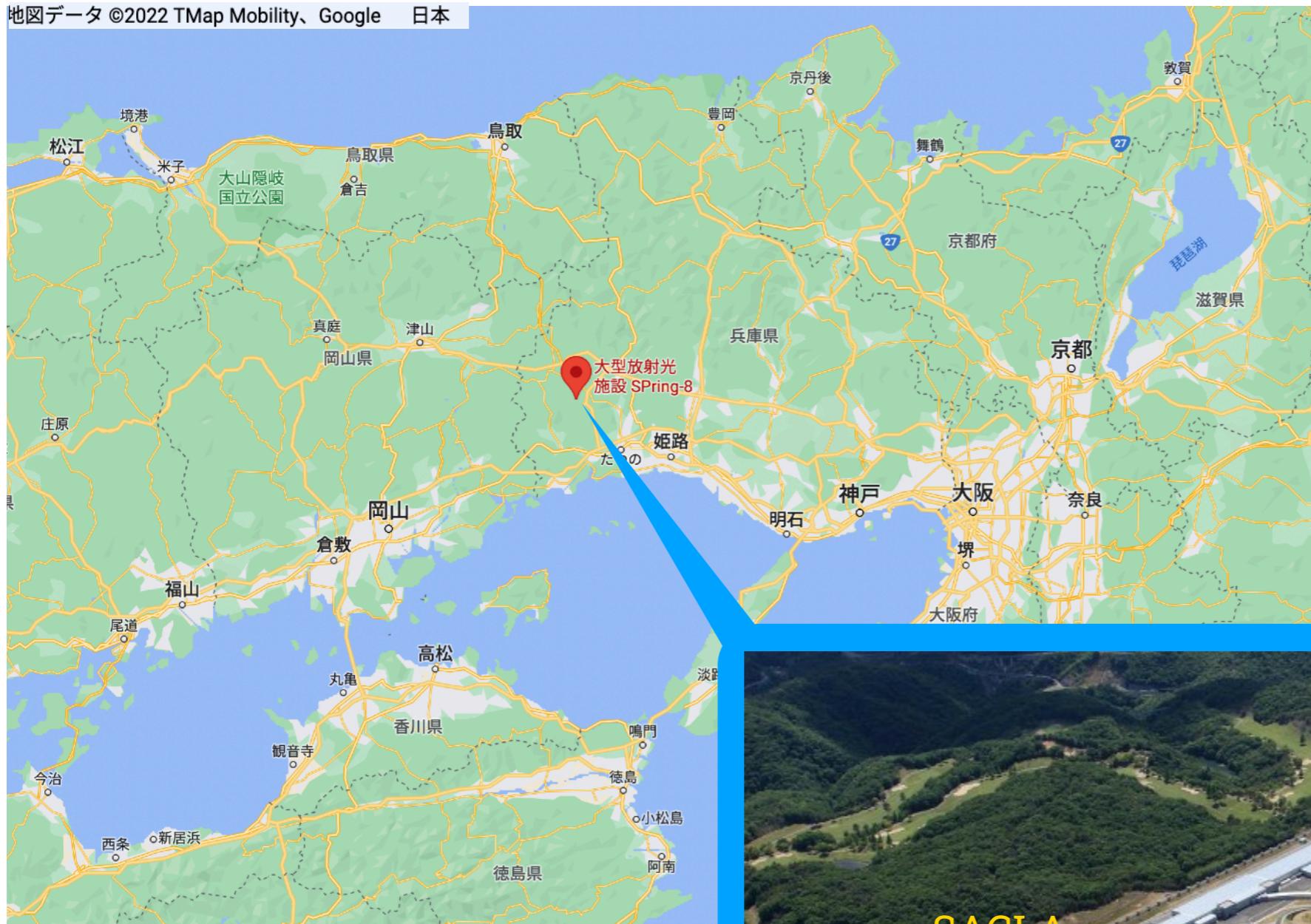
加速器・ビーム物理の機械学習ワークショップ2025@J-PARC, 2025/12/09

Eito Iwai (JASRI/RIKEN)

# SACLA/SPring-8

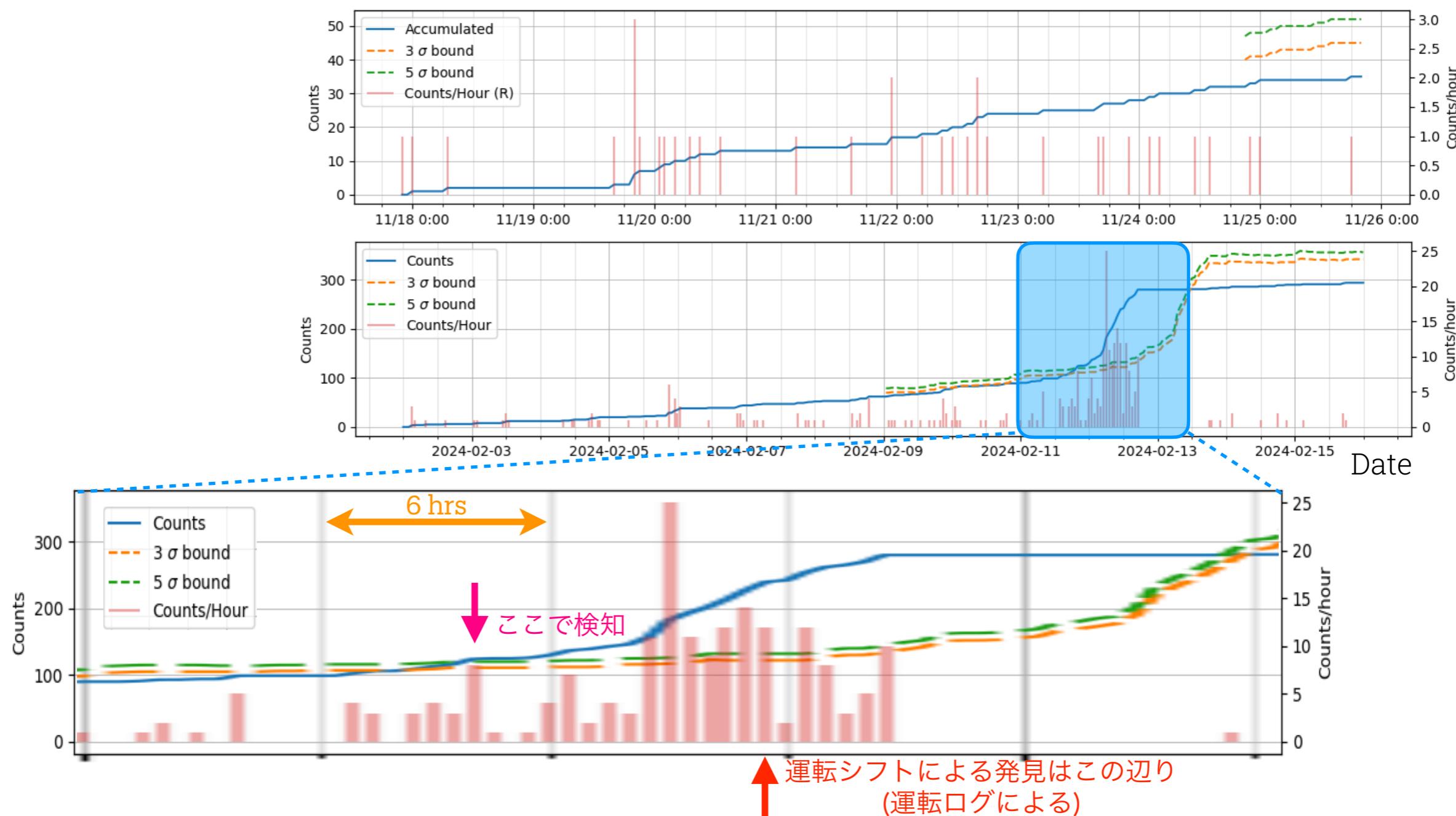
- Synchrotron radiation facility@Hyogo prefecture

地図データ ©2022 TMap Mobility、Google 日本



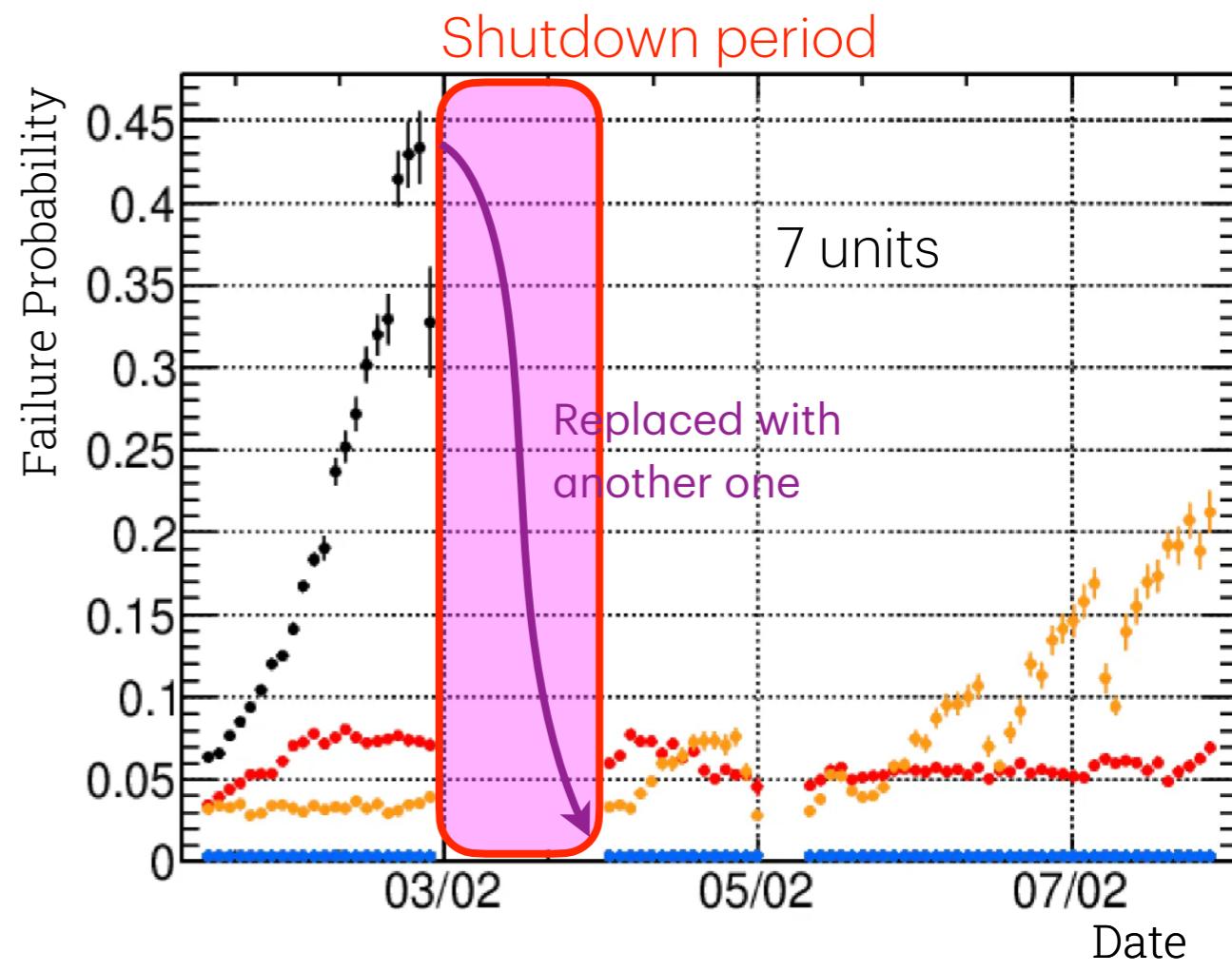
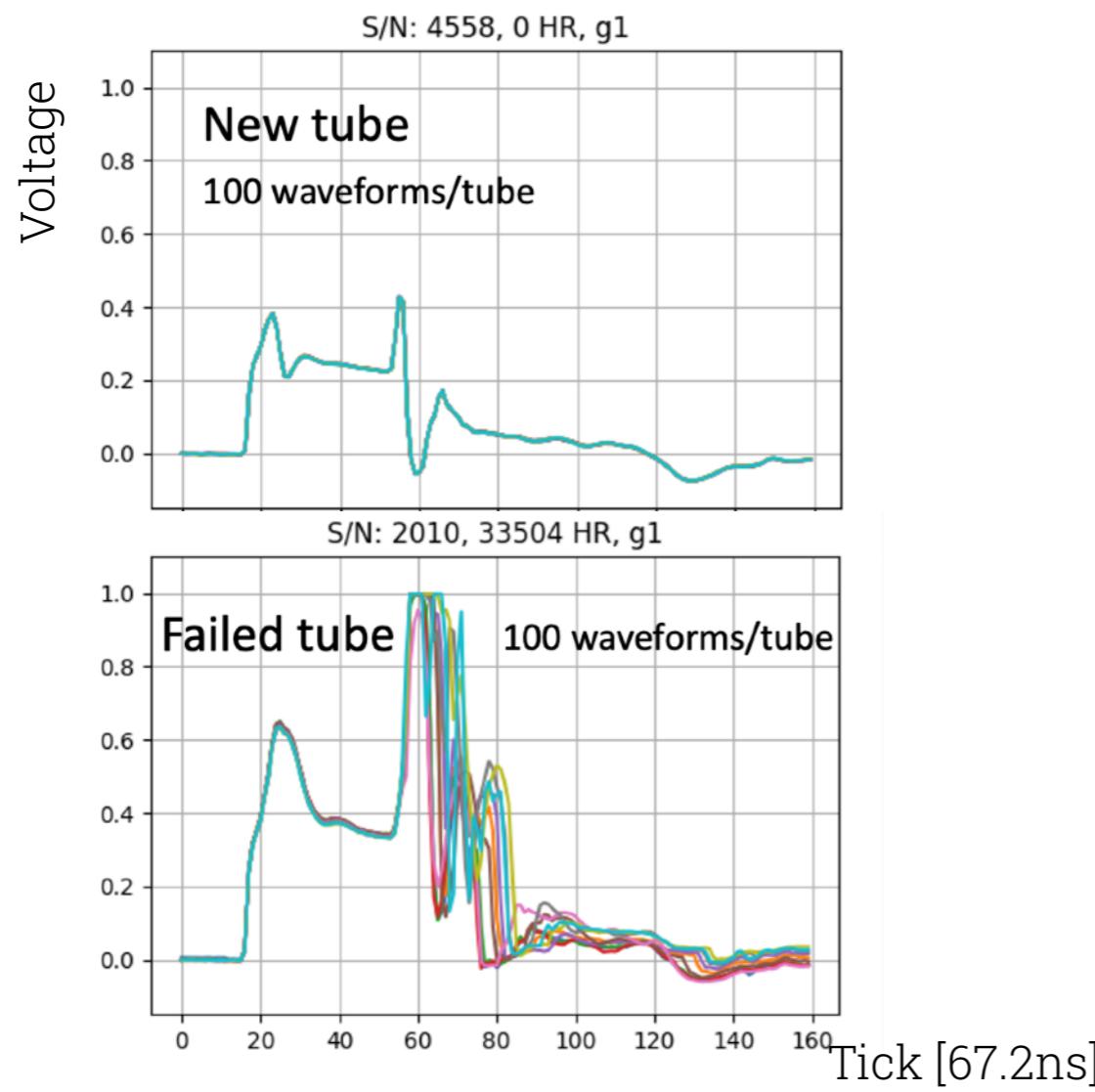
# Anomaly detection; thyratrons

- Thyraterons have two major failure modes
  - Rapidly increasing self-discharge rate
  - Other aging phenomena including surge over-voltage
- The first group can be monitored by using Bayesian inference
- The latter group can be identified with waveforms by using PCA based method [WIP]



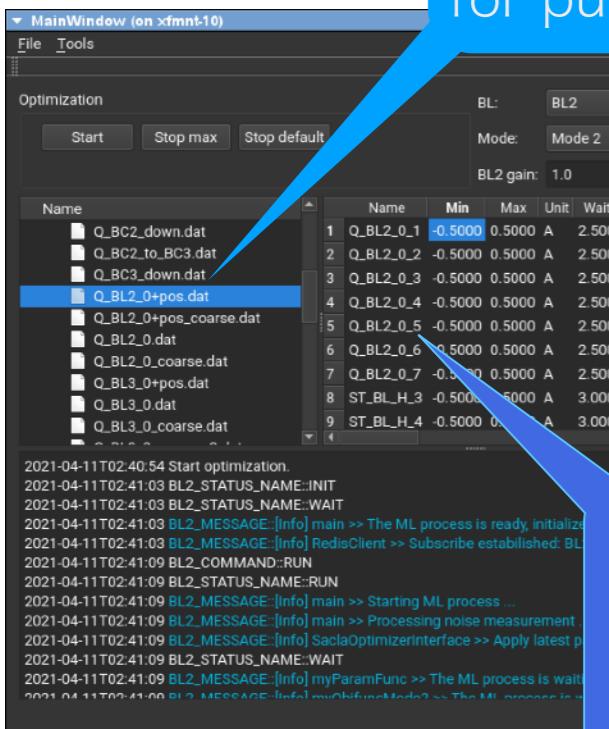
# Anomaly detection; thyratrons

- Thyratrons have two major failure modes
  - Rapidly increasing self-discharge rate
  - Other aging phenomena including surge over-voltage
- The first group can be monitored by using Bayesian inference
- The latter group can be identified with waveforms by using PCA based method [WiP]
  - Waveform evaluation based on grid waveforms taken at test-stand
  - Just started partial test operation on actual equipment.



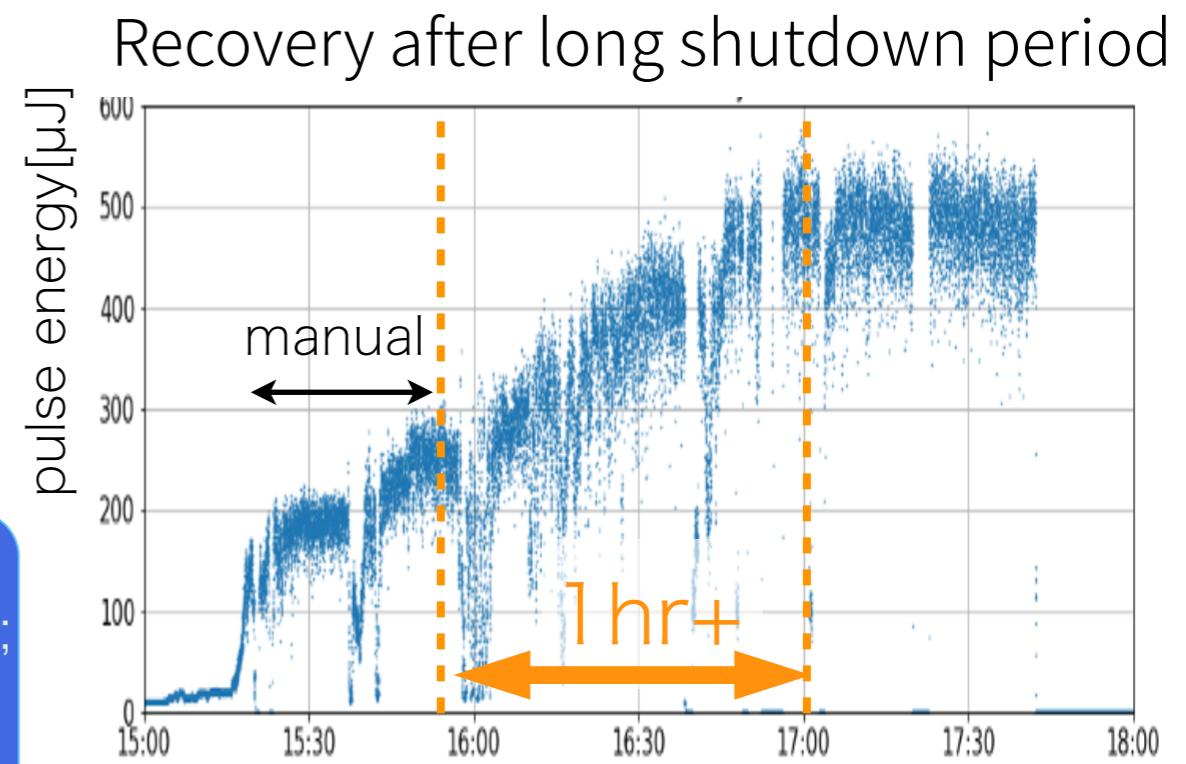
# Beam optimizer

- ▶ Introduced AI-driven automated tuning system; beam optimizer
  - Operation becomes more complicated even though the tuning time becomes constrained
  - Fully utilize the improved tunability and flexibility for efficient tuning and better XFEL performance



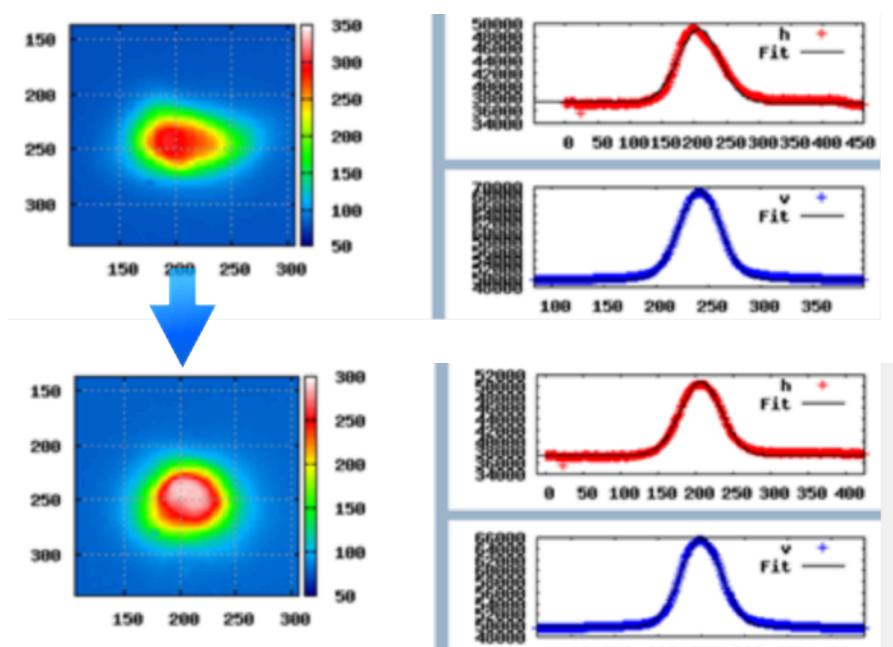
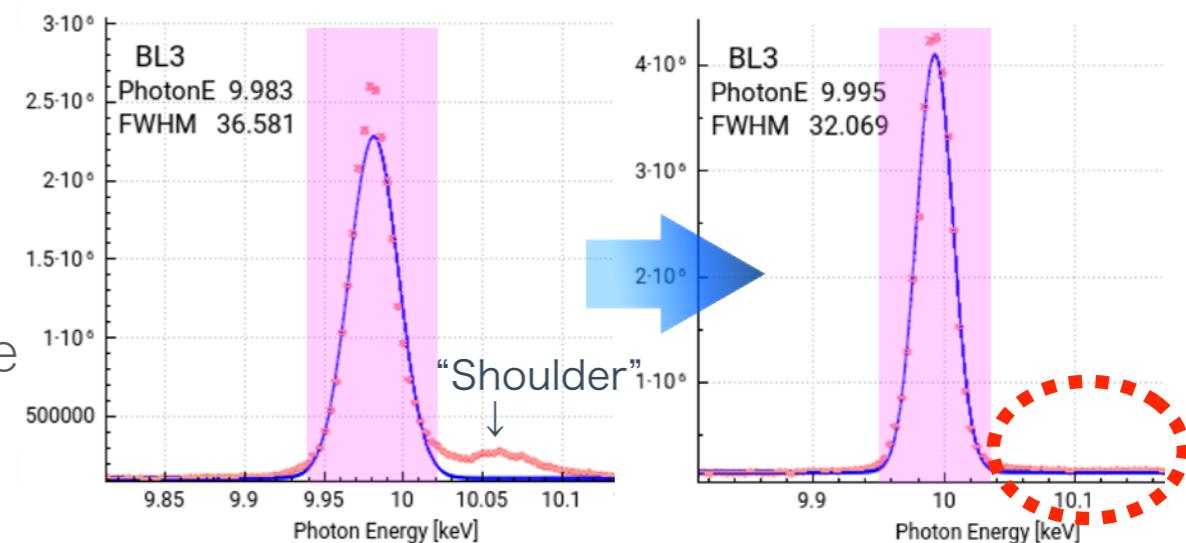
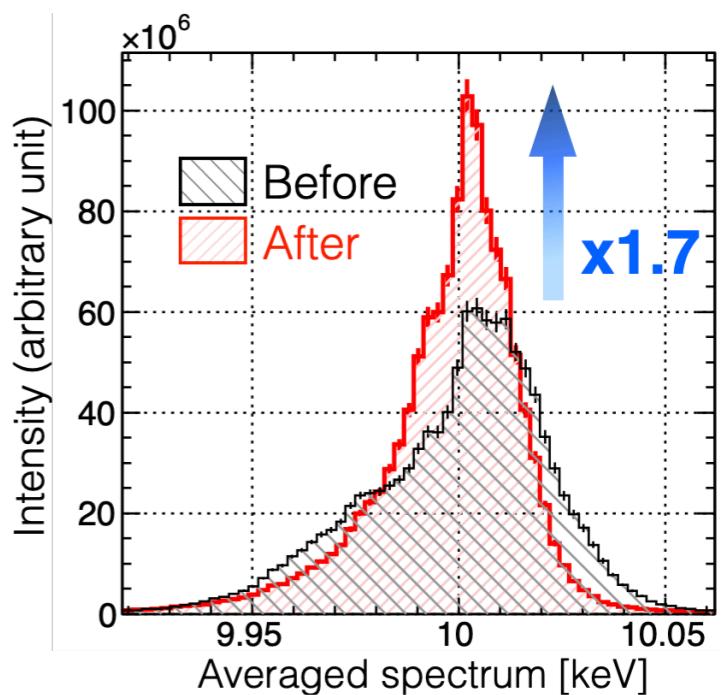
Templates of control parameters for purposes and sections

List of par-range, wait-time and control message for each device; RF phases, focusing magnets, ID injection trajectory, ID tapers



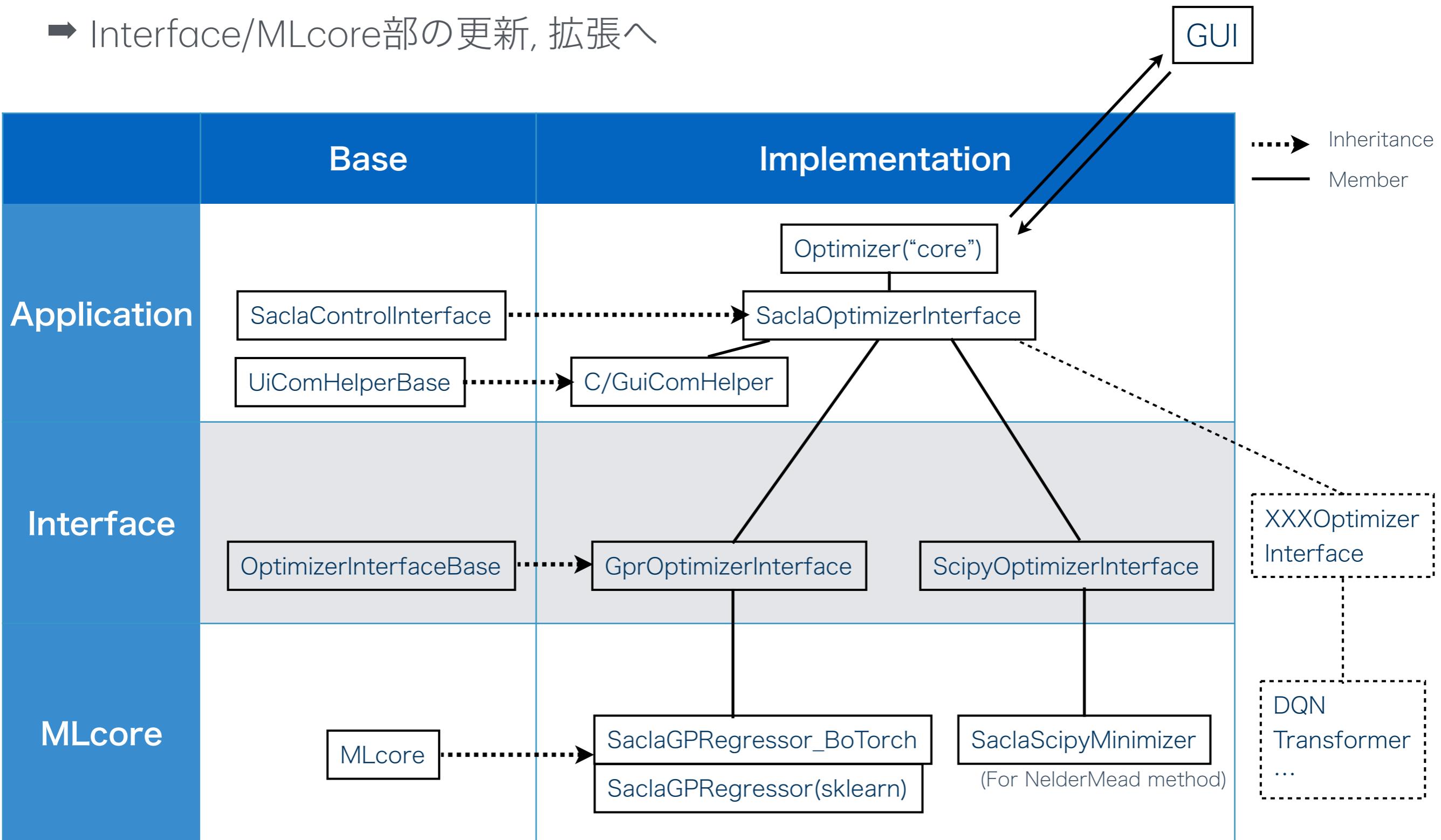
# Various optimization

- ▶ Spectral brightness
  - A new spectrometer with enough resolution to measure shot-by-shot spectral-width online installed permanently (with great help of BL group)
  - Optimized for spectral brightness of central wavelength
- ▶ Spectral shape
  - Request from an user: “Suppress peak height in side-band less than 3 % of one of central wavelength”
  - A “shoulder” often remains in spectrum even after the spectral brightness optimization
- ▶ XFEL spatial profile
  - Request from another user: “Round spatial profile with single ‘core’”
  - Double (or sometimes multiple-) “cores” often appear in spatial profile
- Tailor-made XFEL



# Optimizer 実装

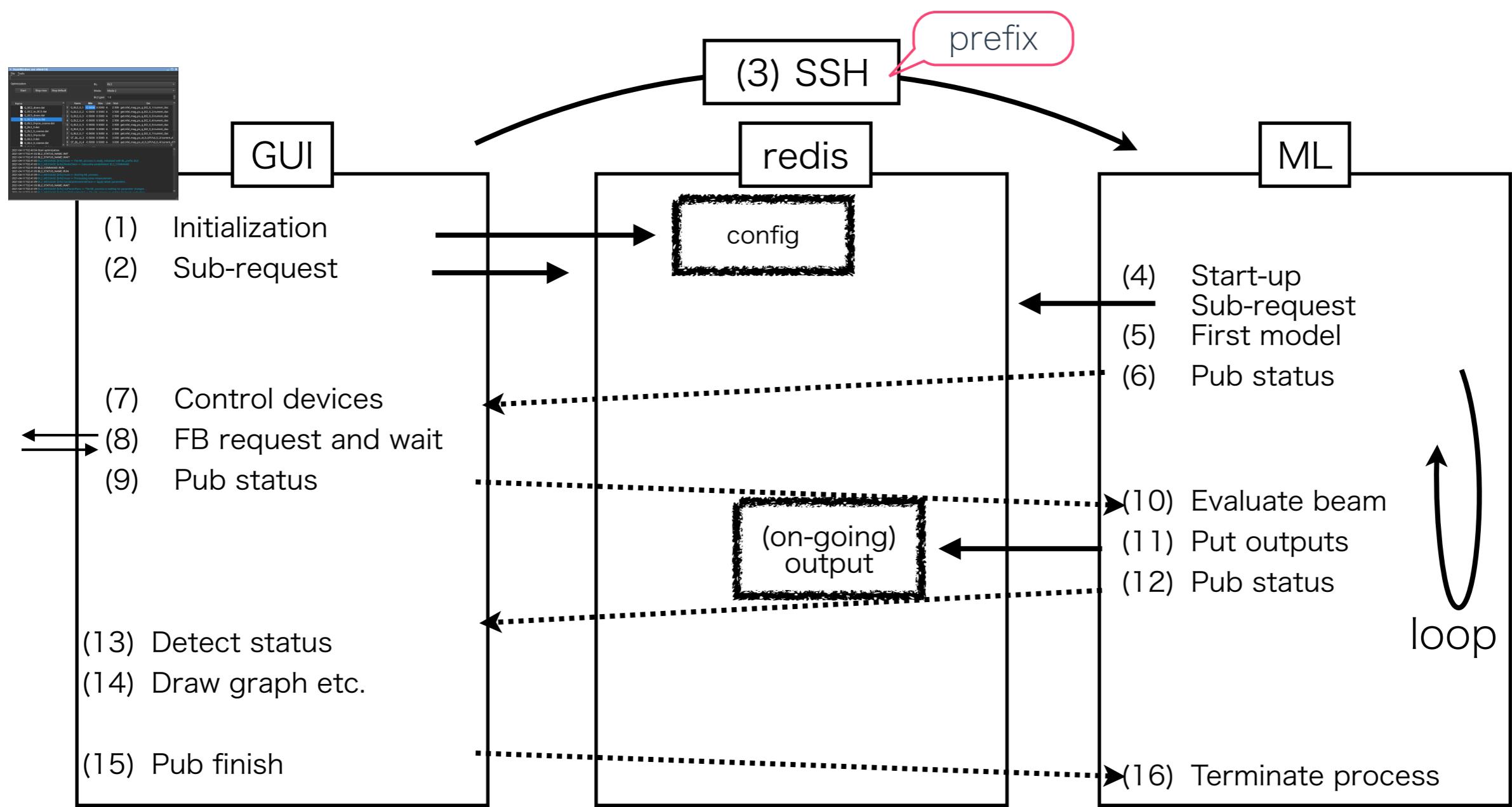
- ▶ 最初の実装以来、更新はもっぱら 'Application' 層 (応用部)が主
- ▶ Interface/MLcore部の更新、拡張へ



# Optimizer 実装

Translated into EPICS by Gemini 3.0

- ▶ redis (pub/sub, db) ~ Soft IOC(?)
  - In-memory DB ~ Process Database(?)
  - pub/sub ~ camonitor or event-driven CA (?)



# 探索範囲: 超立方体か超球か?

- ▶  $N$  が大きくなると、直感以上に超立方体の頂点までが遠くなり、超球/超立方体の体積比も急激に小さくなる
- ▶ パラメータレンジは運転員の感覚で設定されるが、感度がある(= 1つの軸だけで有意に性能が落ちうる)量になることが想定されるので、初期状態がますますであることを考えると  $N \rightarrow \infty$  の超立方体の頂点はあまりに遠い
- ▶ 超立方体を探索範囲とすると  $N \rightarrow \infty$  で、そんな遠い点ばかりであまりに非効率
- 初期サンプリングも、その後の探索範囲も超球ベースの方が効率が良さそう  
(探索範囲は constrained BO で制限?)

<b>N (Dimension)</b>	<b>V_Sphere / V_Cube</b>	<b>Center-to-Vertex</b>
<b>2</b>	0.7854	1.41
<b>3</b>	0.5236	1.73
...	...	...
<b>5</b>	0.1645	2.24
...	...	...
<b>10</b>	0.0025	3.16

# 超球/超立方体探索の pros/cons

- ▶ 超球: 初期状態からの相対変異で探索する加速器調整などに向いてる?
  - 探索空間が格段に (reasonable に) 狹い
  - 感度が高いパラメータのアームを稼ぐために、感度の低いパラメータを動かさないようにする抑止力が働く
  - 探索範囲の中に山頂が見えていなくても、登り傾向の尾根が見えていれば、何度か繰り返すことで頂上を目指せる
- ▶ 超立方体
  - パラメータ範囲が相対変位ではなく、絶対変位で与えられる境界条件のよな場合、ちゃんと端まで探さないと解が見つからないことも

# 初期サンプリングの効率化

○これまでパラメータ数Nについて、それぞれの軸方向に  $\pm 50\%$  と原点(初期パラメータ)の  $2N+1$  点をサンプリング → “正軸体頂点( $2N$ ) + 原点” の計  $2N+1$  個 (正方形( $N=2$ ), 正八面体( $N=3$ ), ...)

→ Nが15とかになると、モデル構築前にかなり時間が掛かる

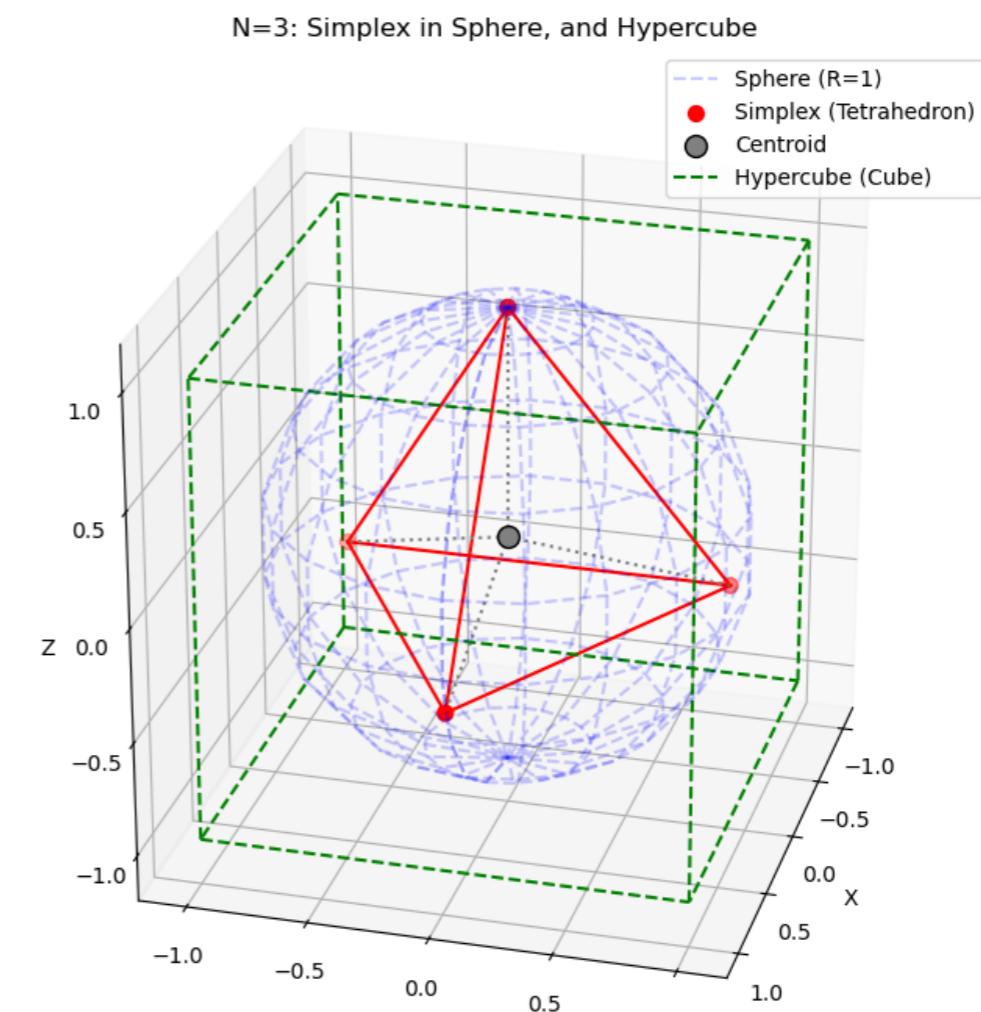
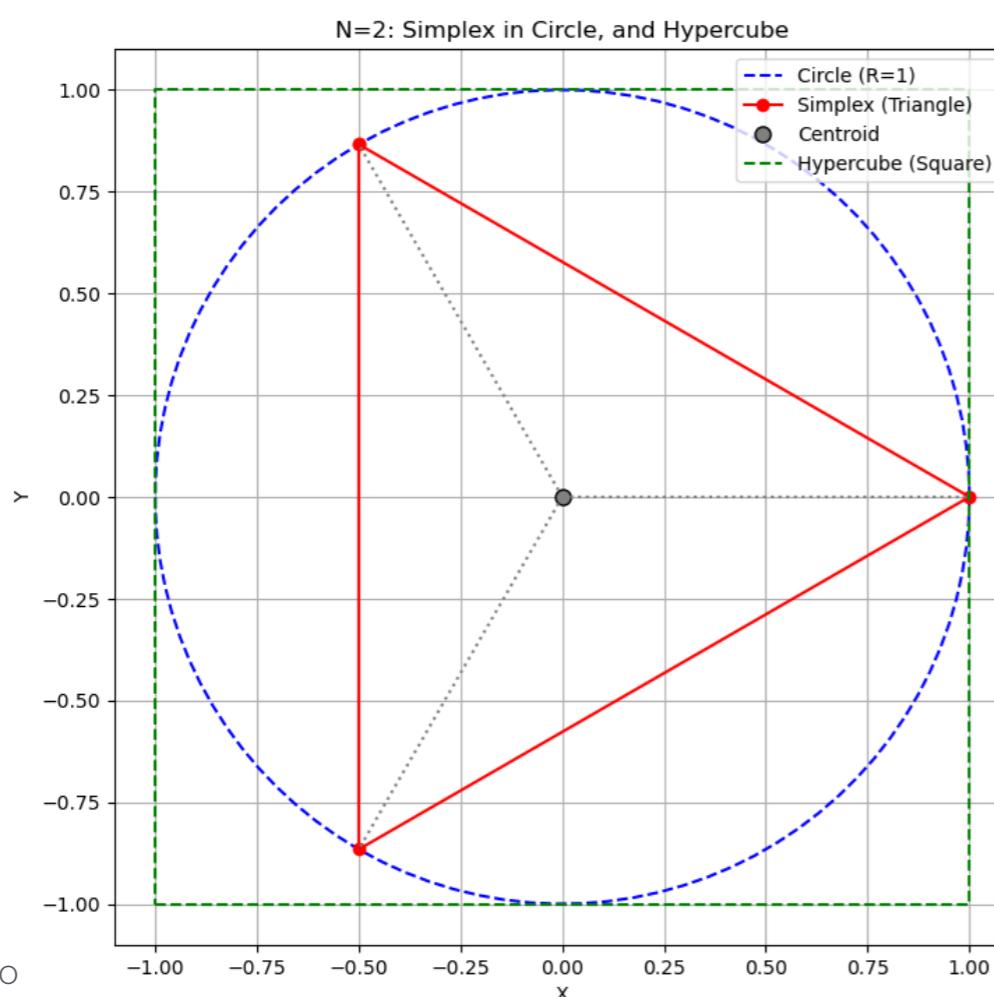
▶ “超球に内接する正単体”の  $N+1$  点と原点の計  $N+2$  にできないか？

単体: N次元空間を構成する  
最も基本的な図形

✓  $N \rightarrow \infty$  に連れて開始までサンプリング数 or 時間が約半分になる

✓ 初期データにクロスター(パラメータ間の相関)が多く含まれるようになる

- 初期状態でもある程度は出ているので(特にlinacでは)一つだけ動かして大きく伸びるという状況はレア？
- 1軸方向なら運転員の手動調整でもわかりやすい

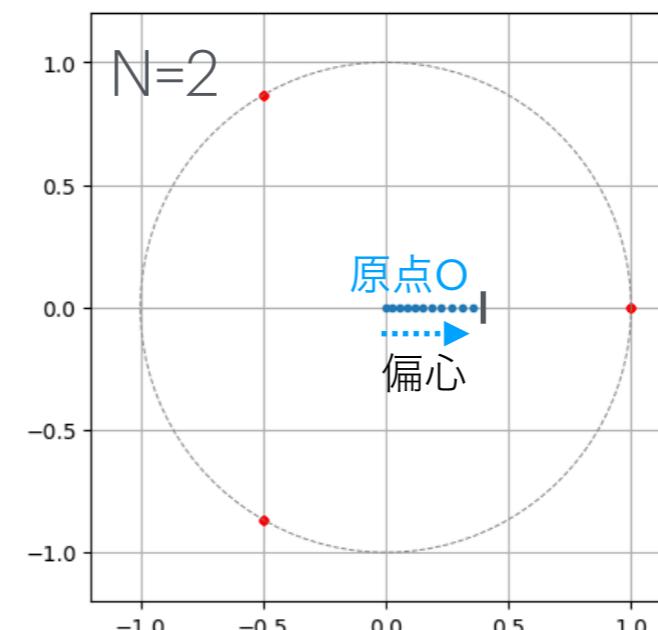
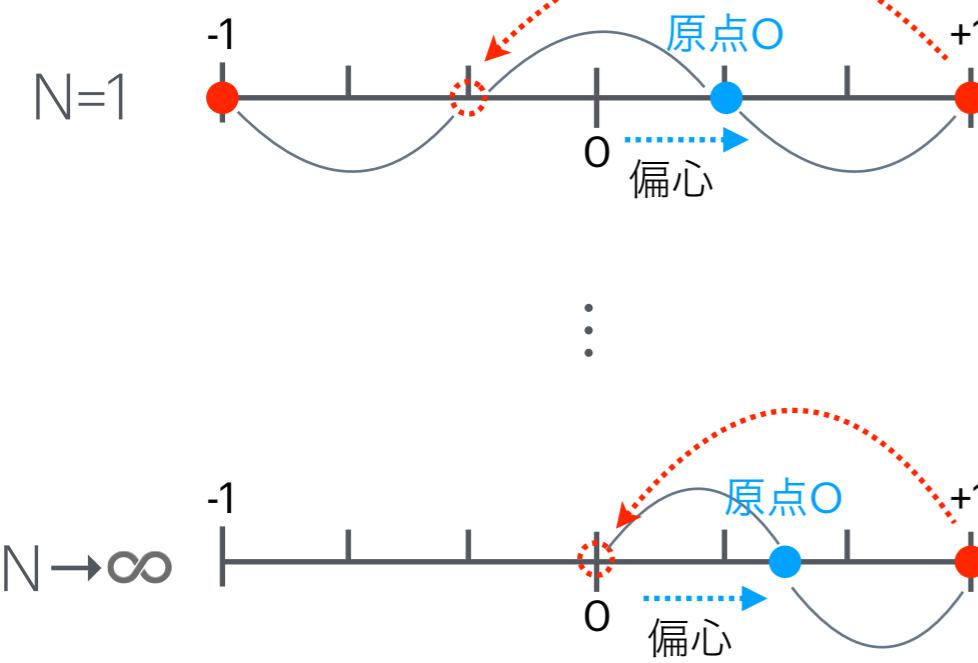


超球に内接する正単体の例: 正三角形( $N=2$ ), 正四面体( $N=3$ )

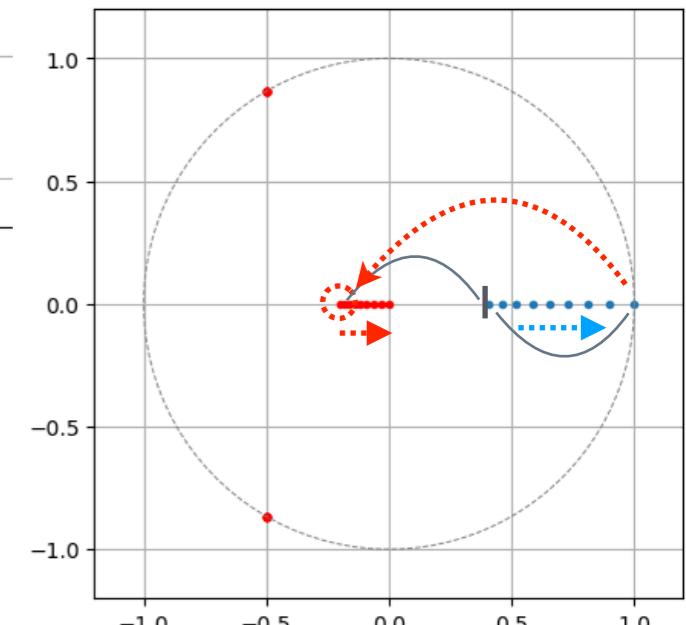
# パラメータ範囲が偏心している場合

- ▶ 原点  $(0,0,0,\dots)$  は調整前の初期状態を表すので、サンプリング点の一つに加えたいが、探索範囲の中心とは限らない
- ▶ (相対)パラメータレンジが  $[-1.0, +0.5]$  や  $[\pm 0, +2]$  のように ‘偏心’ している場合、原点+正単体に固定していると、原点がいざれかの頂点に近くなり ‘情報量’ が著しく落ちる可能性がある
  - 各軸上でサンプリングする元の実装では原点が range の  $\pm 1/3$  を跨ぐと原点を端点とするように変更していた
- ▶ 最近接距離が  $N$  で決まる閾値以下にならないように、 “原点+正単体” を “原点を一つの頂点とする準正単体 + それらの重心” に切り替える
  - 偏心ベクトル上に頂点の一つが来るように正単体全体を回転 (偏心量が non-zero の場合)  
(正単体には回転の自由度がある)
  - 一次元の場合の閾値は偏心量が  $1/3$ 、  $N \rightarrow \infty$  で重心が動かなくなるので閾値は  $1/2$

(閾値: 偏心の絶対値 =  $N/(2N+1)$ )



モード切替

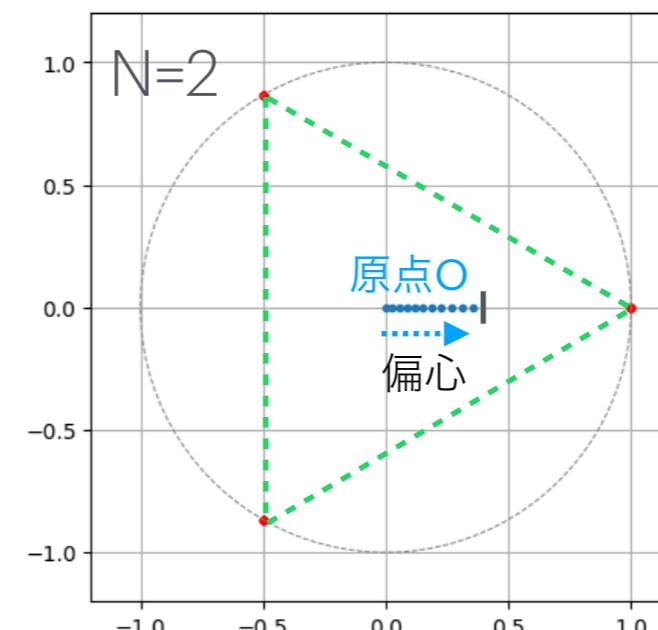
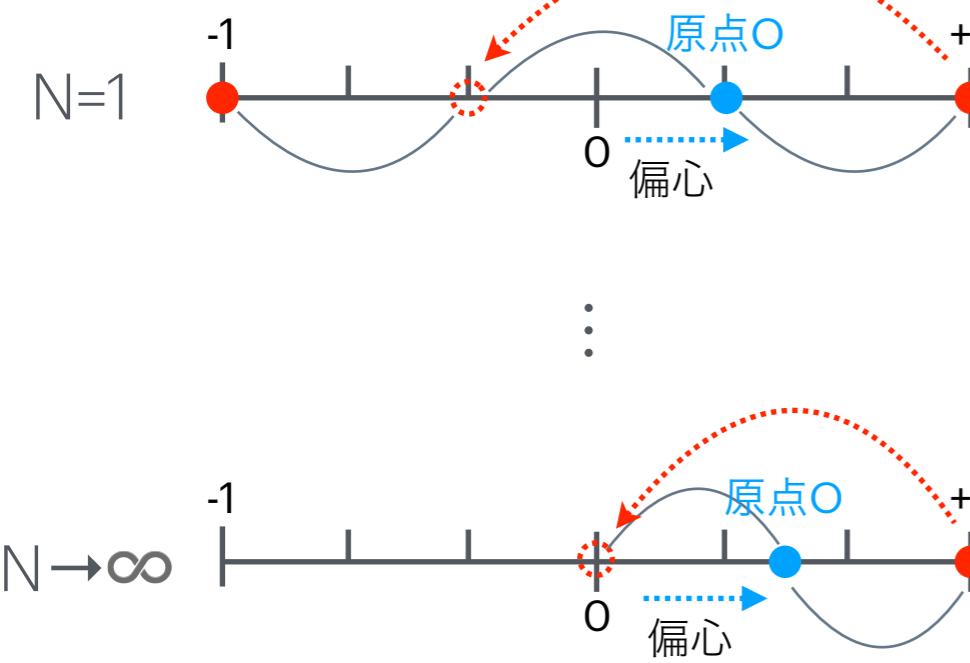


規格化された探索空間上の中点  $O(0, 0)$  と、実空間上の ‘原点’ は別なので注意

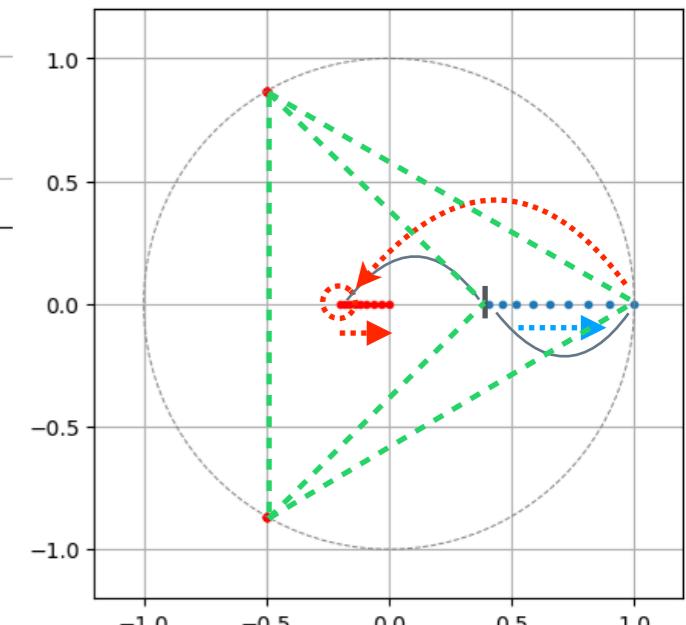
# パラメータ範囲が偏心している場合

- ▶ 原点  $(0,0,0,\dots)$  は調整前の初期状態を表すので、サンプリング点の一つに加えたいが、探索範囲の中心とは限らない
- ▶ (相対)パラメータレンジが  $[-1.0, +0.5]$  や  $[\pm 0, +2]$  のように ‘偏心’ している場合、原点+正単体に固定していると、原点がいざれかの頂点に近くなり ‘情報量’ が著しく落ちる可能性がある
  - 各軸上でサンプリングする元の実装では原点が range の  $\pm 1/3$  を跨ぐと原点を端点とするように変更していた
- ▶ 最近接距離が  $N$  で決まる閾値以下にならないように、 “原点+正単体” を “原点を一つの頂点とする準正単体 + それらの重心” に切り替える
  - 偏心ベクトル上に頂点の一つが来るように正単体全体を回転 (偏心量が non-zero の場合)  
(正単体には回転の自由度がある)
  - 一次元の場合の閾値は偏心量が  $1/3$ 、  $N \rightarrow \infty$  で重心が動かなくなるので閾値は  $1/2$

(閾値: 偏心の絶対値 =  $N/(2N+1)$ )



モード切替



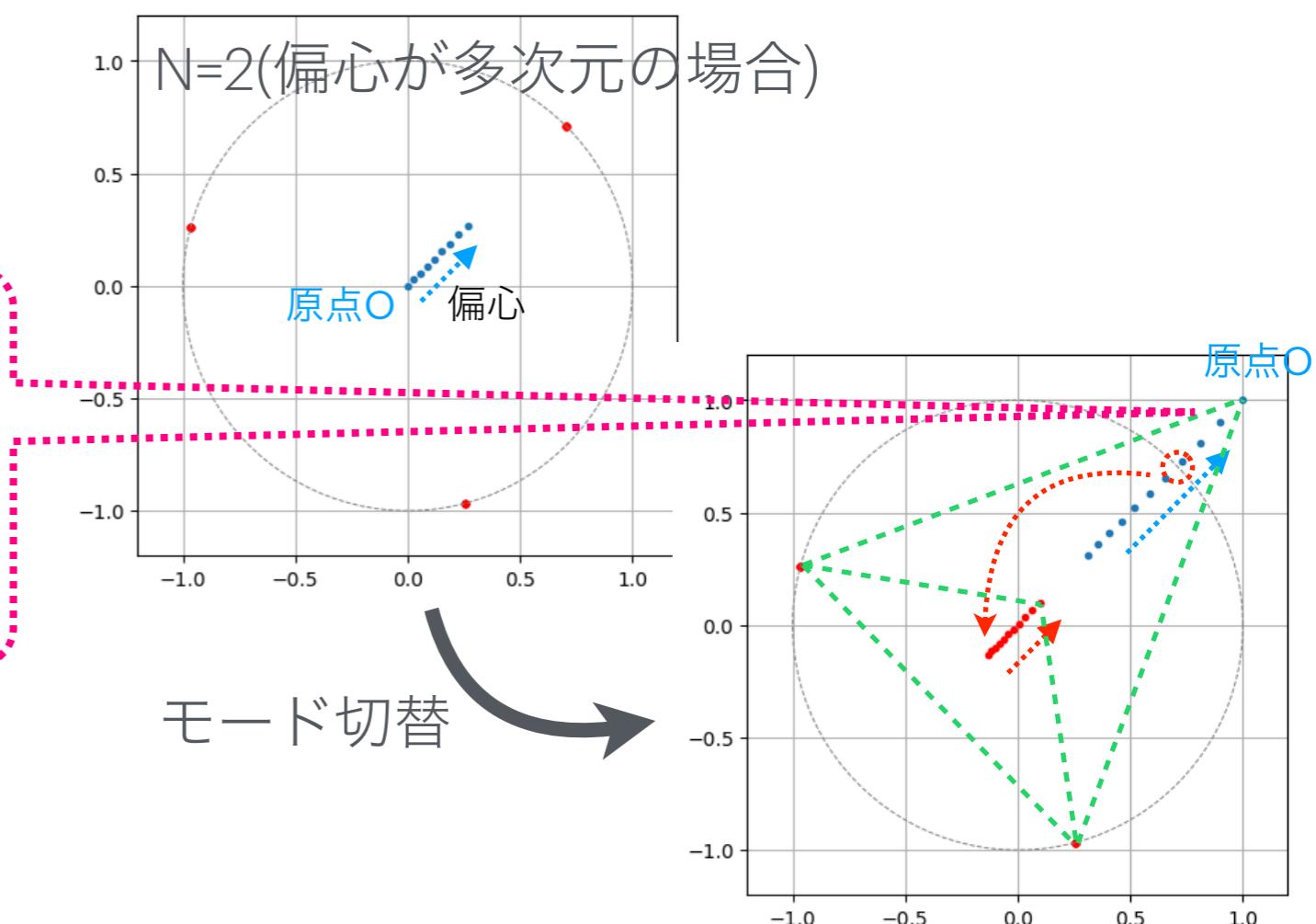
規格化された探索空間上の中点  $O(0, 0)$  と、実空間上の ‘原点’ は別なので注意

# パラメータ範囲が偏心している場合

- ▶ 原点  $(0,0,0,\dots)$  は調整前の初期状態を表すので、サンプリング点の一つに加えたいが、探索範囲の中心とは限らない
- ▶ (相対)パラメータレンジが  $[-1.0, +0.5]$  や  $[\pm 0, +2]$  のように ‘偏心’ している場合、原点+正単体に固定していると、原点がいざれかの頂点に近くなり ‘情報量’ が著しく落ちる可能性がある
  - 各軸上でサンプリングする元の実装では原点が range の  $\pm 1/3$  を跨ぐと原点を端点とするように変更していた
- ▶ 最近接距離が  $N$  で決まる閾値以下にならぬように、“原点+正単体” を “原点を一つの頂点とする準正単体 + それらの重心” に切り替える
  - 偏心ベクトル上に頂点の一つが来るように正単体全体を回転 (偏心量が non-zero の場合)  
(正単体には回転の自由度がある)
  - 一次元の場合の閾値は偏心量が  $1/3$ 、 $N \rightarrow \infty$  で重心が動かなくなるので閾値は  $1/2$

(閾値: 偏心の絶対値 =  $N/(2N+1)$ )

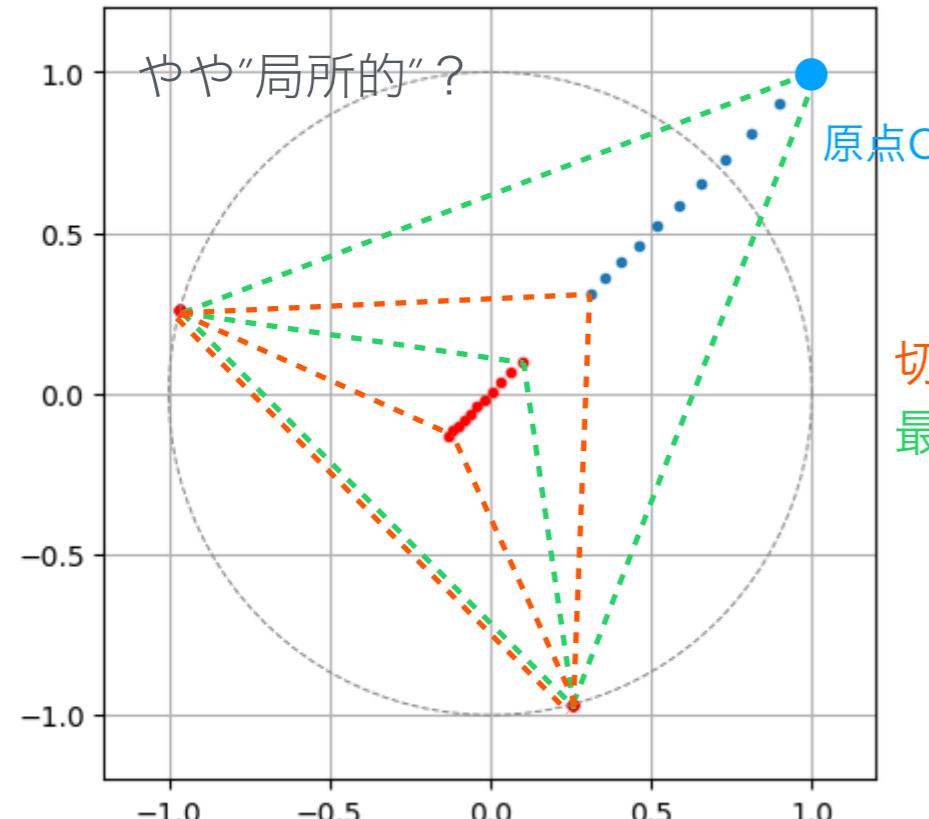
● 原点が超球外に突き出て、歪な単体になり、重心も原点方向に偏心しうるが、初期状態側に寄っているだけなので問題ないだろう



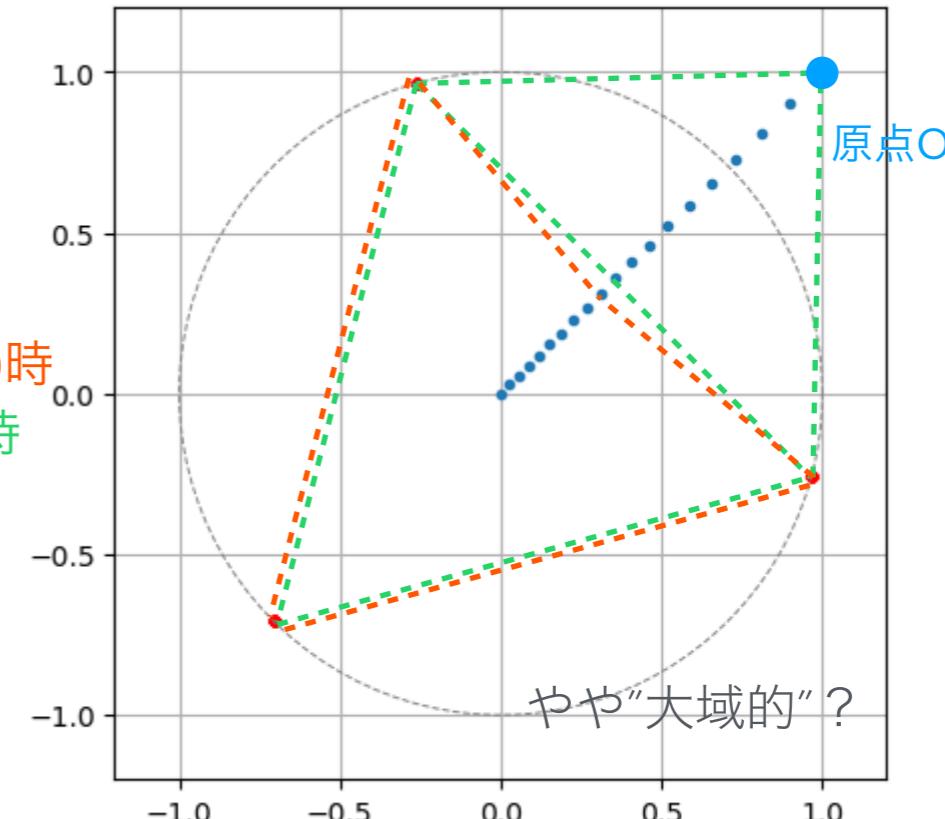
# パラメータ範囲が偏心している場合 (cont'd)

- 偏心していない場合: numpy の simplex を導出するメソッドが吐く  $V_0(1, 0, 0, \dots)$  を  $(1, 1, 1, 1, \dots)$  方向(最大混交)にアラインする
  - 偏心している場合: 偏心ベクトルと  $V_0(1, 0, 0, \dots)$  を反平行になるように回転する
    - 平行にアラインした時の最近接距離:  $(N+1)/(2N+1)$
    - 反平行にアラインした時の最近接距離:  $\sqrt{1-1/N^2}$ , モード切替不要
- $(N+1)/(2N+1) < \sqrt{1-1/N^2}$  ( $N>1$ ): '反平行' を選択すると単純な最近接距離は稼ぎやすい。  
しかし、'初期位置' から最も遠く離れた点を取ることになるので、初期状態周りの情報量は少ないかも？

$N=2$  最大偏心時: 平行戦略(左) と 反平行戦略(右)

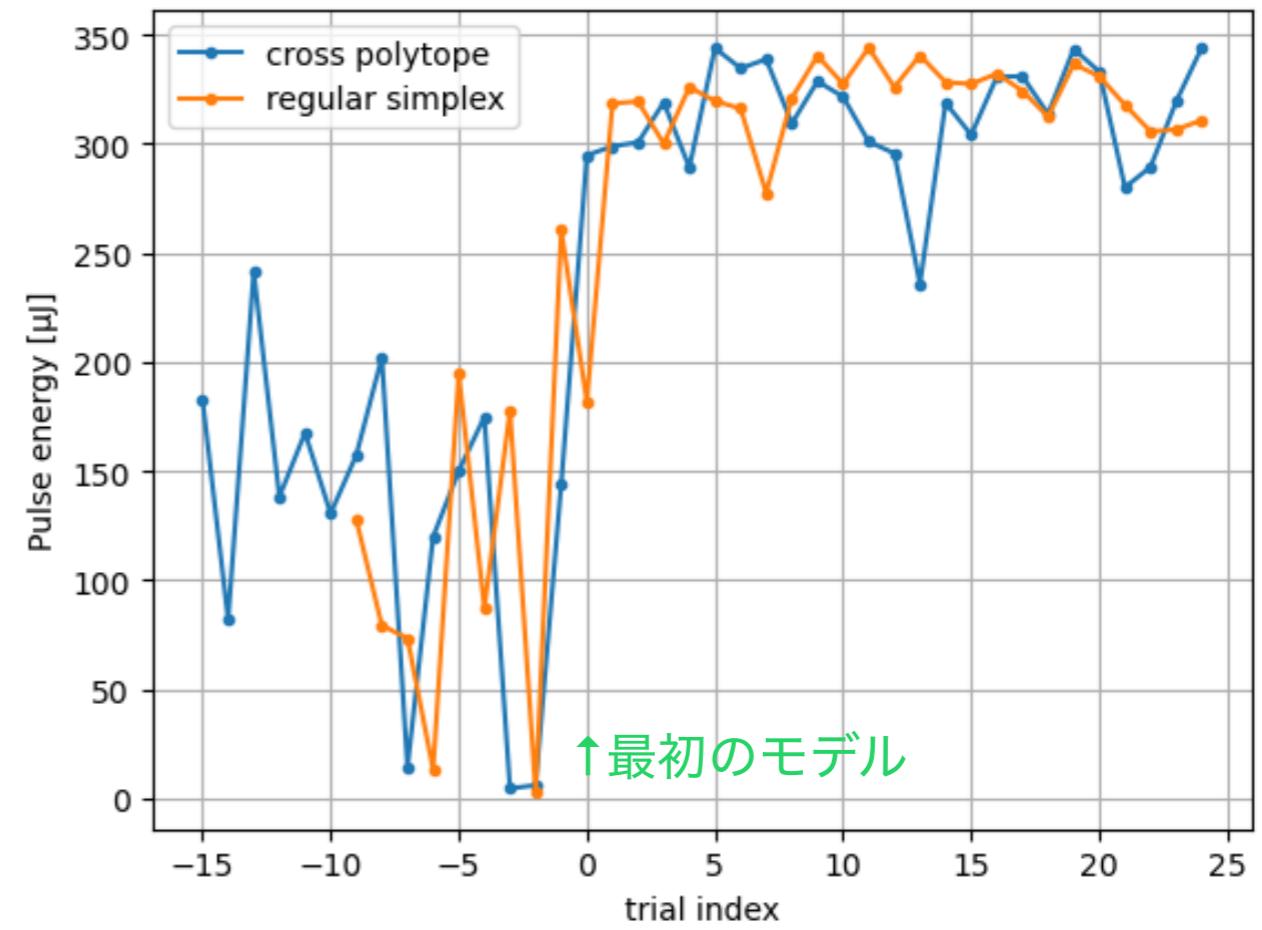
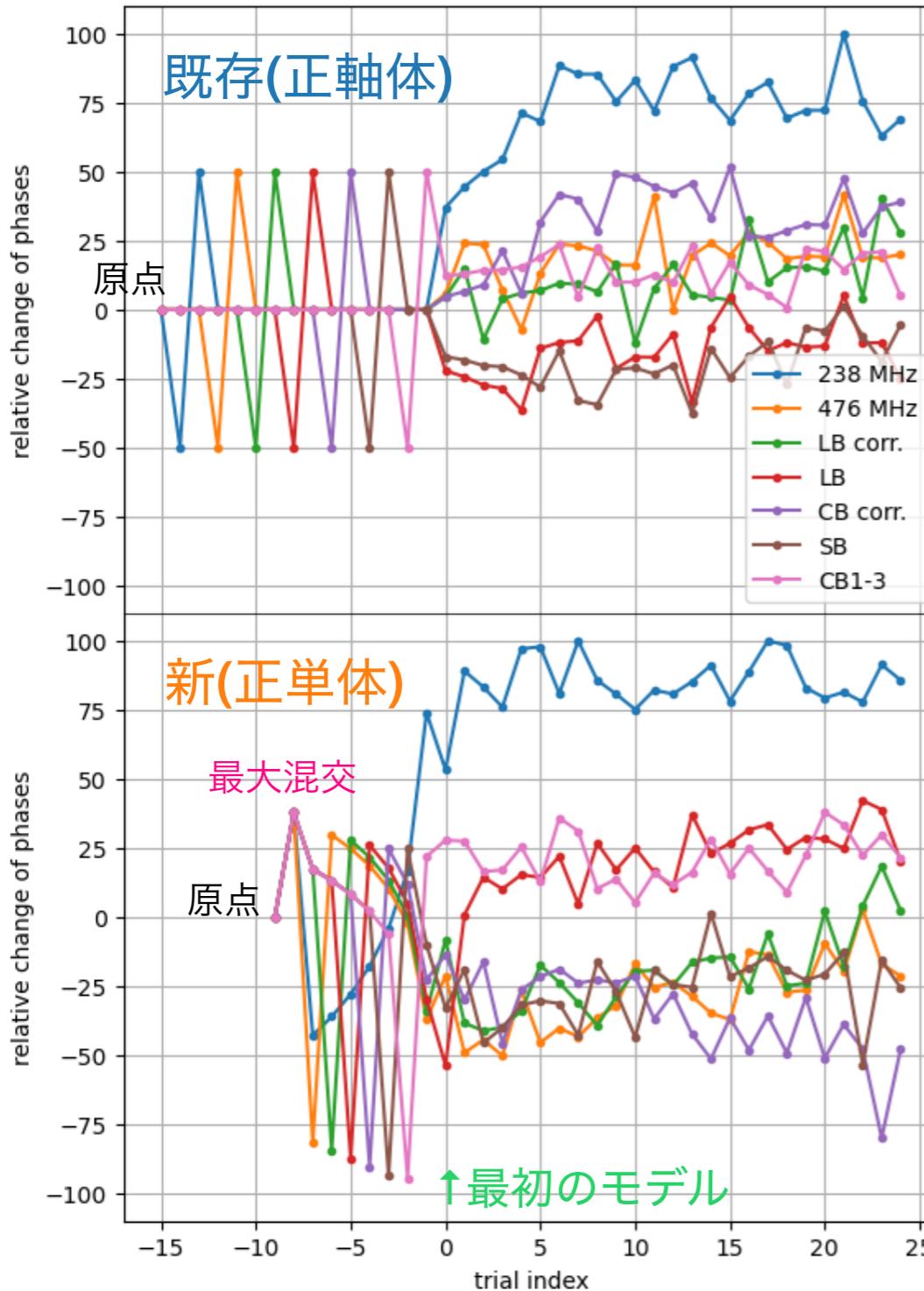


切替(平行)時  
最大偏心時



# 実機動作確認

- ▶ SACLÀ/XFEL のバンチ圧縮に関する 7 つの位相を適当にデチューン
- ▶ パルス強度を性能指標として最適化 (問題が簡単すぎたようだが...)



# Time-varying/Non-stationary BO

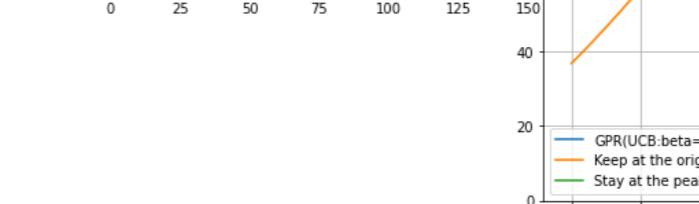
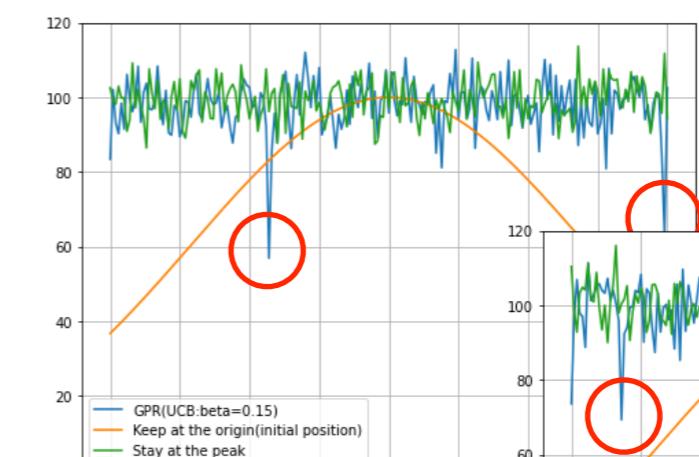
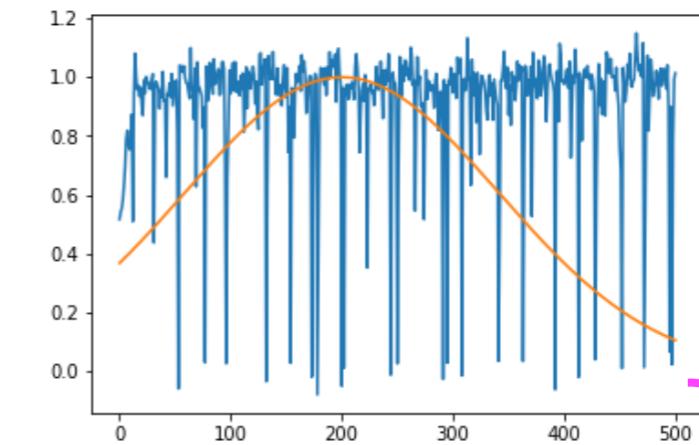
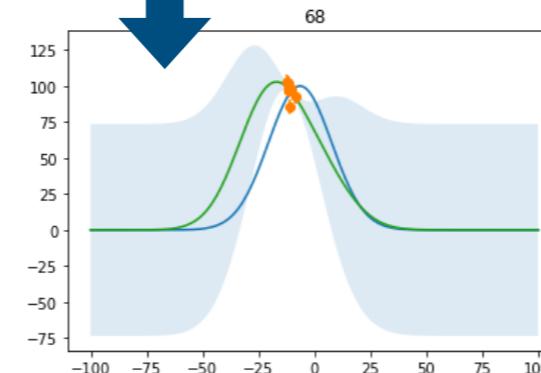
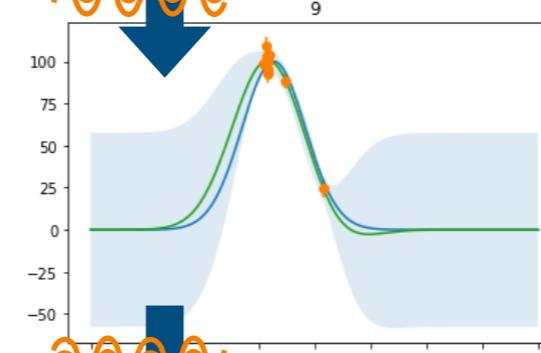
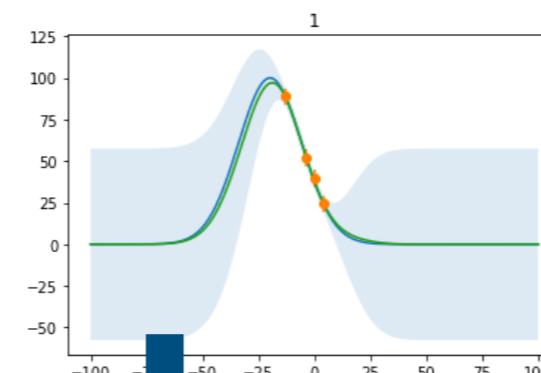
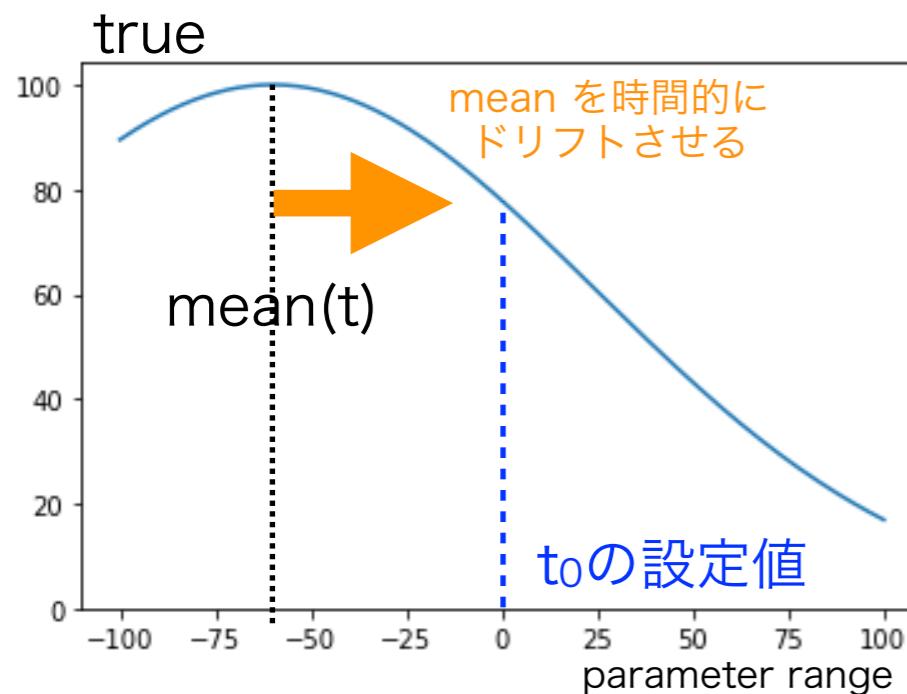
- ▶  $N$  パラメータ の既存の GP-BO に, 時間軸を追加して  $N+1$  化
- ▶ 時間軸には制御パラメータとは別の時間スケール(length scale制限)を課せるよう  
にする

## \* 実装

- GprOptimizerInterfaceを継承 (OptimizerInterface層)
- 時間情報を組み込む I/O のみ拡張  
調整パラメータ  $xN$  + 時間軸  $x1 \rightarrow$  MLcore層へ( $N+1$ )  $\rightarrow$  再分離
- SaclaGPRegressor\_BoTorch (MLcore層)的には  $N+1$  次元の普通の GP-BO
- 渡された timespan を基準に '今' を時間軸  $[0,1]$  の 0.9(可変) に固定  
(時間軸データはループ毎に  $\Delta t$  だけマイナス方向にシフトしてモデルを再生成)  
GP-BO で  $0.9 + \Delta t$  を評価
- 過去のデータは自然と影響が小さくなる

# Time-varying/Non-stationary BO

- ▶ 簡単な動作確認: 1次元 + 非相関の2パラメータ
    - 制御パラメータに対してガウス分布で応答するノブ
    - 中央値を時間に比例してシフトさせる(ドリフトを模擬)
    - これまで: UCB( $\beta < 1$ ), prior-mean=0 etc ではピーク外への制御を抑制しきれずもう一步

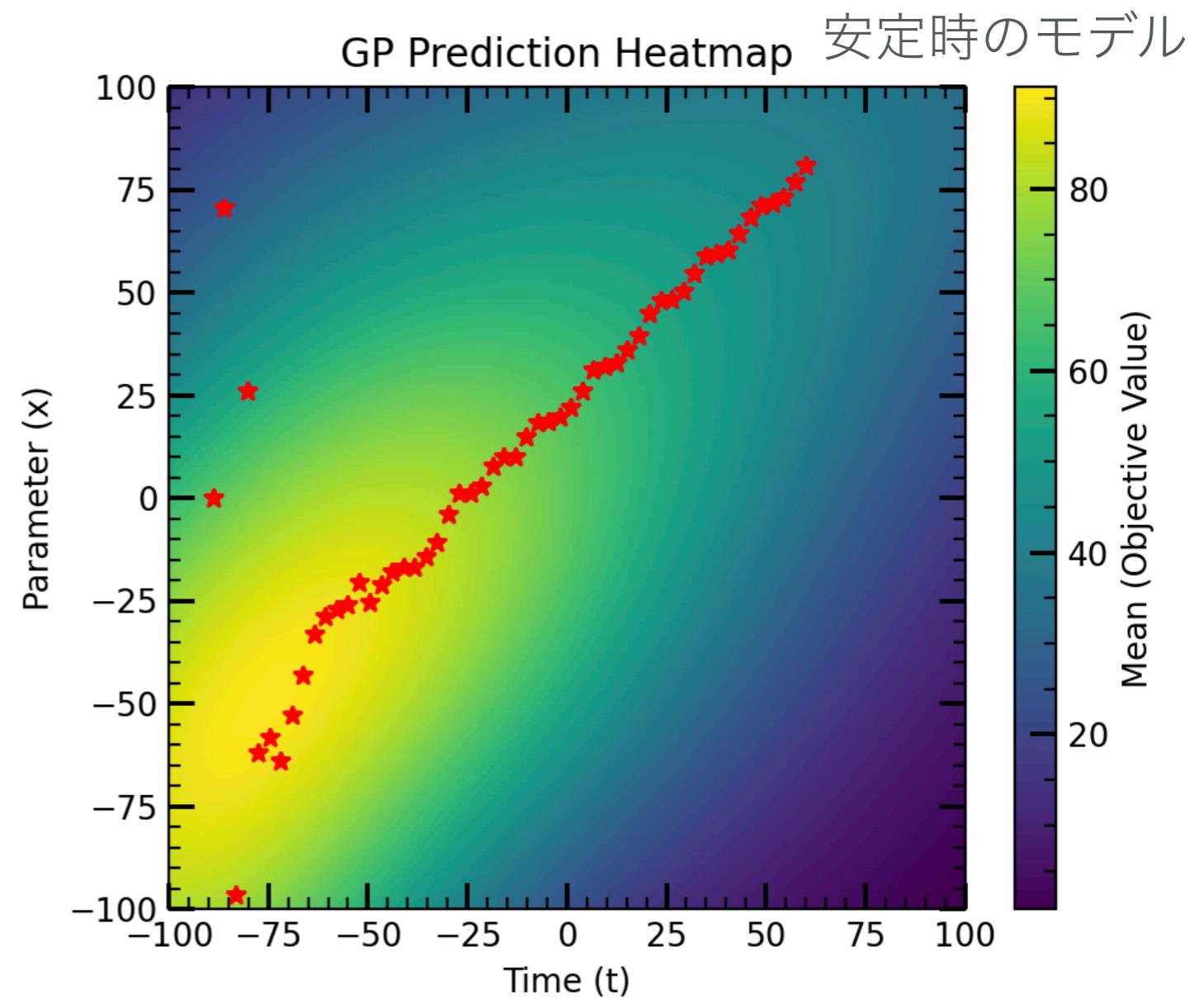
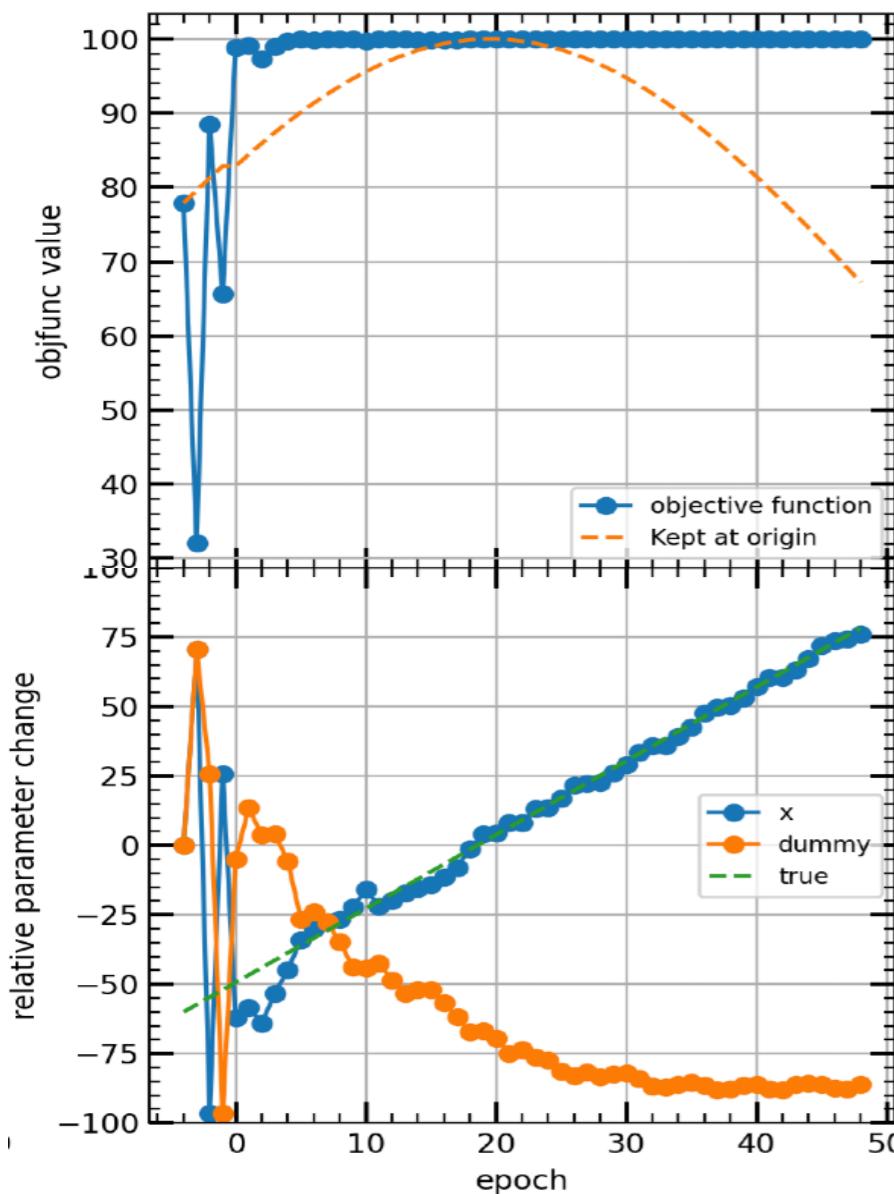


- EI
- Prior: AVG

- UCB( $\beta < 1$ )
- Prior: 0

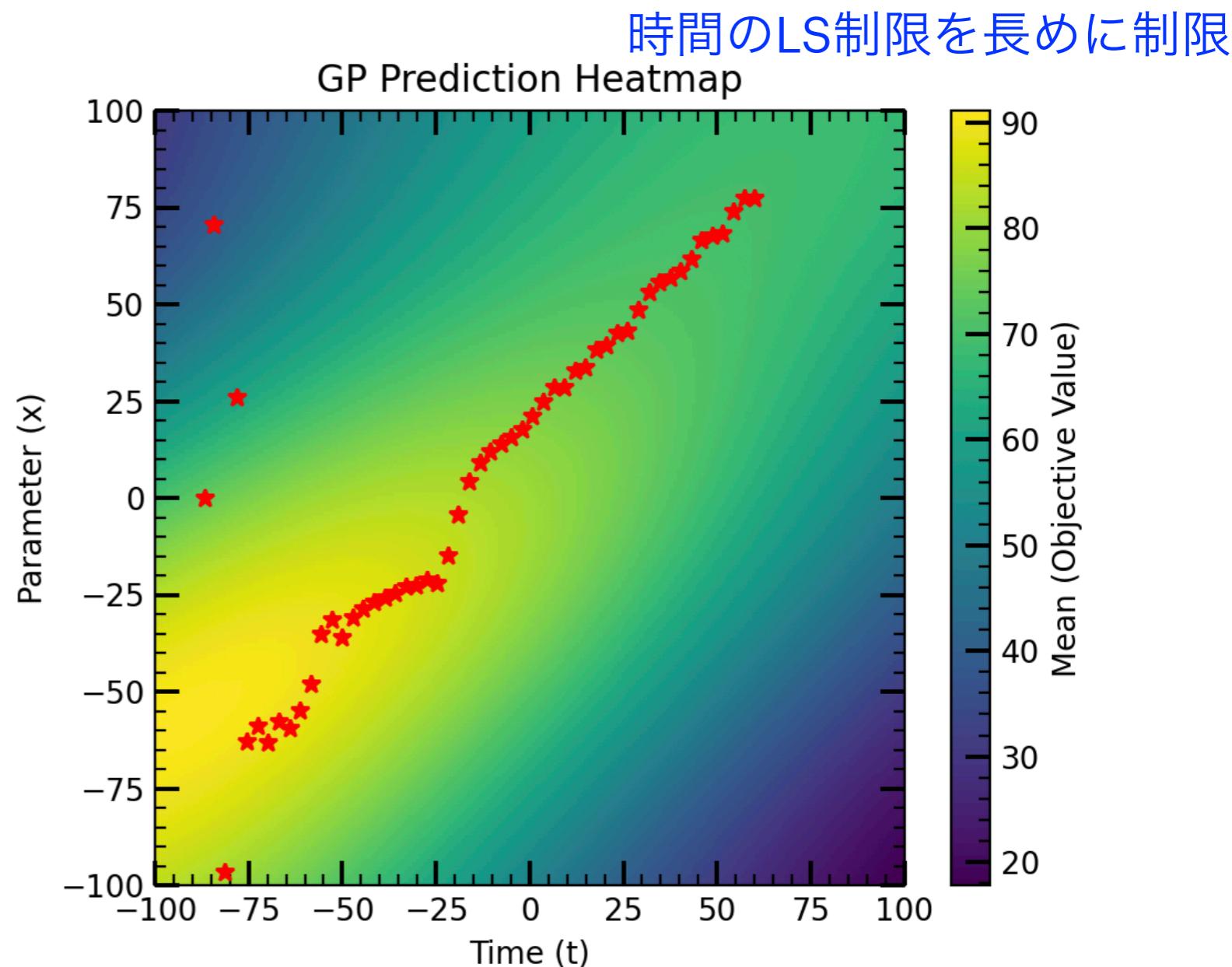
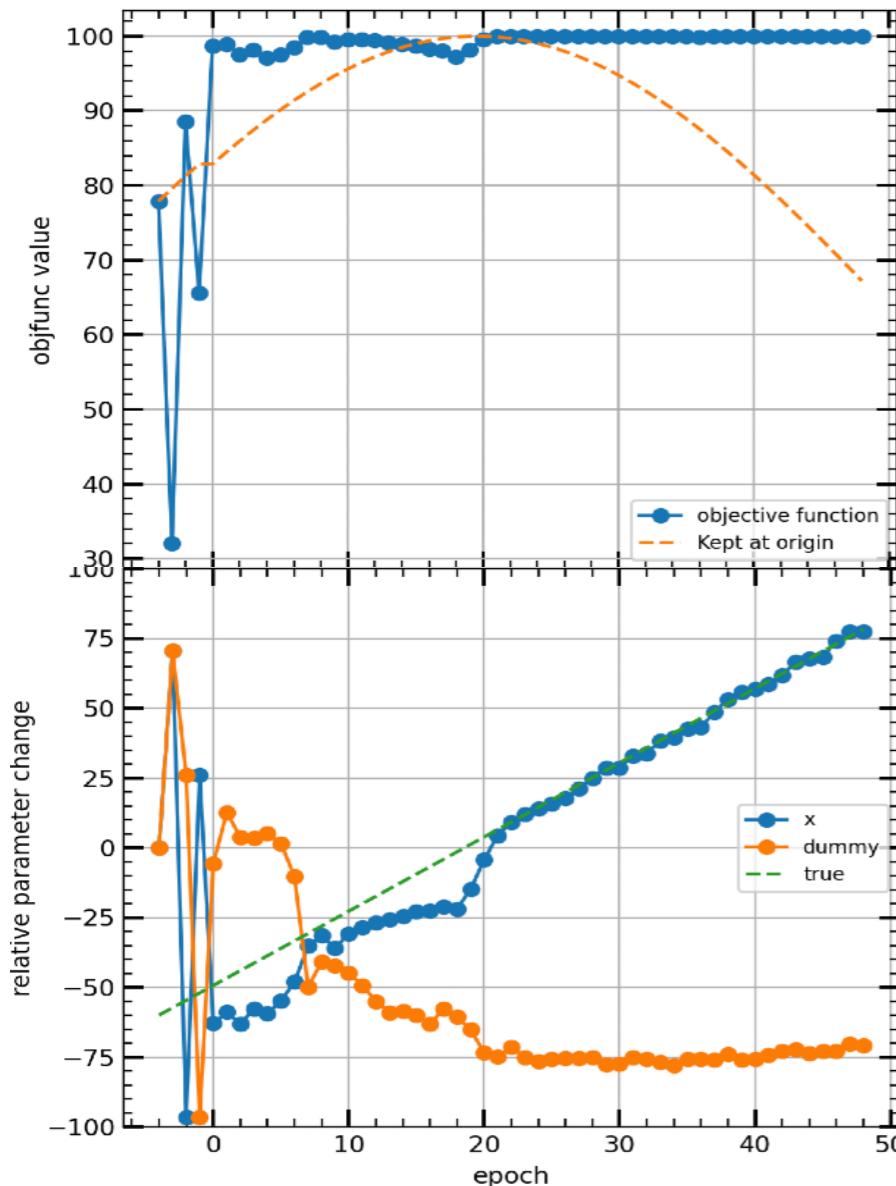
# Time-varying/Non-stationary BO

- ▶ 簡単な動作確認: 1次元 + 非相関の2パラメータ
  - 制御パラメータに対してガウス分布で応答するノブ
  - 中央値を時間に比例してシフトさせる(ドリフトを模擬)
- Time-varying BO 導入でモデルの‘停滯’を抑制できる?



# Time-varying/Non-stationary BO

- ▶ 簡単な動作確認: 1次元 + 非相関の2パラメータ
  - 制御パラメータに対してガウス分布で応答するノブ
  - 中央値を時間に比例してシフトさせる(ドリフトを模擬)
- Time-varying BO 導入でモデルの‘停滯’を抑制できる?



# Trust Region BO: TuRBO

## ▶ TuRBO-1: 局地探索

パラメータ数が増えるほど裾野に length scale が引っ張られて極値周辺のモデル化に制約が掛かる

→ 'Trust region' を設定する(領域を狭める)ことで極値近傍をよりデータに即してモデル化させる

## ▶ TuRBO-m: 複数の TuRBO-1 を用いたグローバル探索

- 元々のアイデアは Agent を複数 'ランダム' に配置して競わせる

→ 複数の 'シリーズ, 部位毎のセット' を並走して競わせる

= 「どのテンプレートから適用するか」という運転員依存だった部分を自動化

## \* 実装

- GprOptimizerInterfaceを継承(OptimizerInterface層)
- Trust region を定義して、領域内のデータのみをモデル化
- TuRBO-m: 複数の TuRBO-1 を競わせる

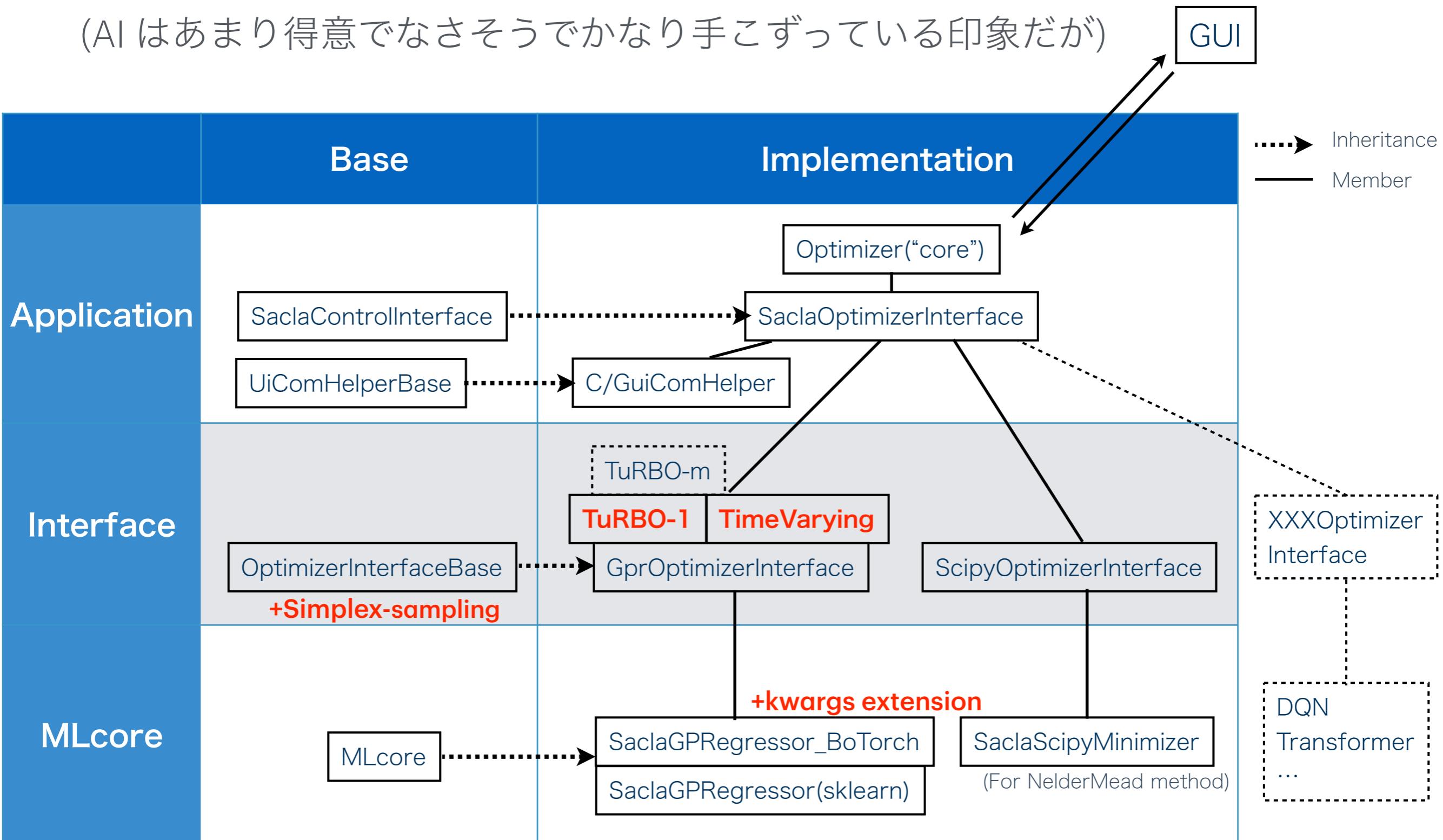
# 実装状況

	Status	強み	コメント
<b>ARB-kernel (BoTorch)</b>	<input checked="" type="checkbox"/> 実装 <input checked="" type="checkbox"/> 試験 <input checked="" type="checkbox"/> 導入	モデル自由度	追加というか実装抜け というかバグに近い sklearn版は元々対応
<b>Regular simplex による初期サンプリング</b>	<input checked="" type="checkbox"/> 実装 <input checked="" type="checkbox"/> 試験 <input checked="" type="checkbox"/> 導入	汎用効率化	
<b>Time-varying</b>	<input checked="" type="checkbox"/> 実装 <input checked="" type="checkbox"/> 試験(仮) <input type="checkbox"/> 導入	ドリフト等時間変化	
<b>TuRBO-1</b>	<input checked="" type="checkbox"/> 実装 <input checked="" type="checkbox"/> 試験(仮) <input type="checkbox"/> 導入	局地最適化	
<b>TuRBO-m</b>	N/A	グローバル最適化	'TuRBO-1' x N
<b>Constrained BO</b>	N/A	汎用	BoTorch の 'constraints' を活用予定 (MLcore層で吸収)

# 実装状況

▶ なるべく重複コードを抑制して差分で実装

(AI はあまり得意でなさそうでかなり手こづっている印象だが)



# 想定用途

	構成	狙い
ユーザー運転中の性能維持	Time-varying + Constrained + TuRBO-1 (UCB; $\beta < 1$ )	TuRBO-1: 局地最適化 Constrained: 悪影響を抑制 Time-varying: 時間変動
調整時のパラメータ最適化	Simplex-sampling + Time-varying + TuRBO-m (EI)	複数の‘パラメータテンプレート’を同時 Simplex: 多パラメータ Time-varying: 時間変動補償

# まとめ

- ▶ GP-BO の効率化, 高度化を進めている
  - ARB-kernel (BoTorch)
  - Regular simplex sampling
  - Time-varying BO
  - TuRBO (-1, -m)
  - Constrained BO
  - ...
- ➡ ユーザー利用中のピーク性能を維持, 高次元グローバル探索
- ▶ 最後に本研究は所属元で契約している Google Workspace の Gemini 2.5 Pro → Gemini 3 Pro (思考) さんに、何度も挫けそうにながらも協力してもらいました。
  - 今後の拡張性, 応用性まで考えた複雑な論理, スケーリングまで考慮した実装はまだ難しそう
  - 各種モジュールの API, 実装, 仕様などを知る取っ掛かりとしては便利かも

WS 後に追加

# WSでのコメント

- ▶ 時間が経って初期サンプリングがレンジ外になつたらどうなるか?
  - Timespan を短くして、初期サンプリングを意図的にレンジ外に(かつレンジ外のデータをクリップ; それ以外は本文中と同様)

