

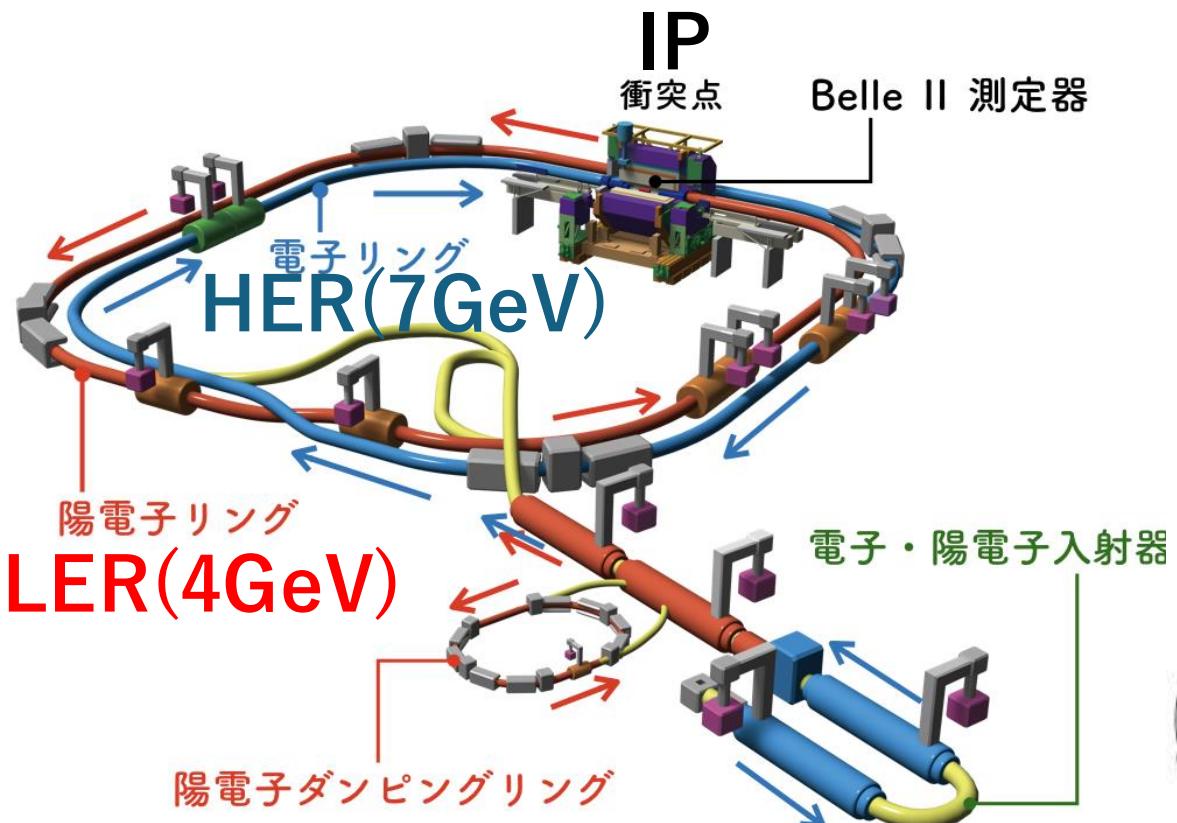
# Belle II 実験における IP knob tuningへの 機械学習の適用

瀧澤陸<sup>A</sup> 後田裕<sup>A B C</sup> 中山 浩幸<sup>B C</sup> 住澤 一高<sup>B C D</sup> 原 康二<sup>B C</sup>  
野尻美保子<sup>B E</sup> Ahmed Hammad<sup>B</sup>

東大理<sup>A</sup>, KEK素核研<sup>B</sup>, 総研大<sup>C</sup>, 奈良女大<sup>D</sup>, IPMU<sup>E</sup>

# Belle II 実験について

## Belle II 実験とSuperKEKB加速器



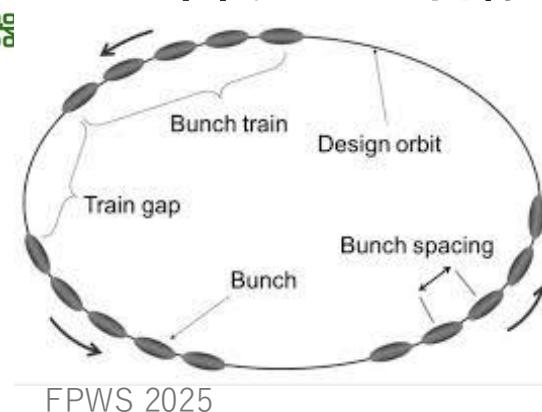
Belle II 実験は現在世界最高のLuminosityを誇る  
SuperKEKB加速器を用いて行われている実験

(瞬間)Luminosity  $L$

$$N = \sigma \int \mathbf{L} dt$$

散乱断面積  $\sigma$  が小さい崩壊の反応回数  $N$  を増やすためには  
Luminosity の向上が不可欠

2030年代までの目標Luminosity :  $6 \times 10^{35} [\text{cm}^{-2}\text{s}^{-1}]$



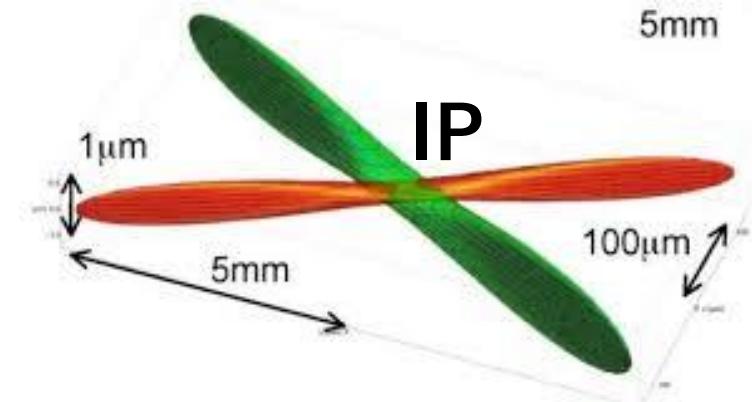
リングの中を電子と陽電子が  
それぞれバンチと呼ばれる塊を  
形成して周回している

# IP knob tuningとは

## Belle II 実験におけるIP knob tuning

Luminosityの簡単な表式 :  $L \propto \frac{N_b I_{b+} I_{b-}}{\Sigma_y \Sigma_z}$

( $\Sigma$  は衝突点で二つのビームの有効ビームサイズ :  $\Sigma_y^2 = \sigma_{y+}^2 + \sigma_{y-}^2$ )



Luminosityを上げるには衝突点での両方のビームサイズを下げる必要がある。  
→オペレーターが都度ビーム調整を行う。

**IP knob**…衝突点(IP:Interaction Point) でどのようなぶつけ方をするかの調整に用いられる  
ビーム力学のパラメータ。  
主要なものはHERとLERにそれぞれ6個( $R_1, R_2, R_3, R_4, \eta, \eta'$ )あり、合計で12個ある。

このIP knobを調整してバックグラウンドを抑えつつより大きなLuminosityを目指す作業を  
**IP knob tuning**と呼ぶ。

# 研究のモチベーション

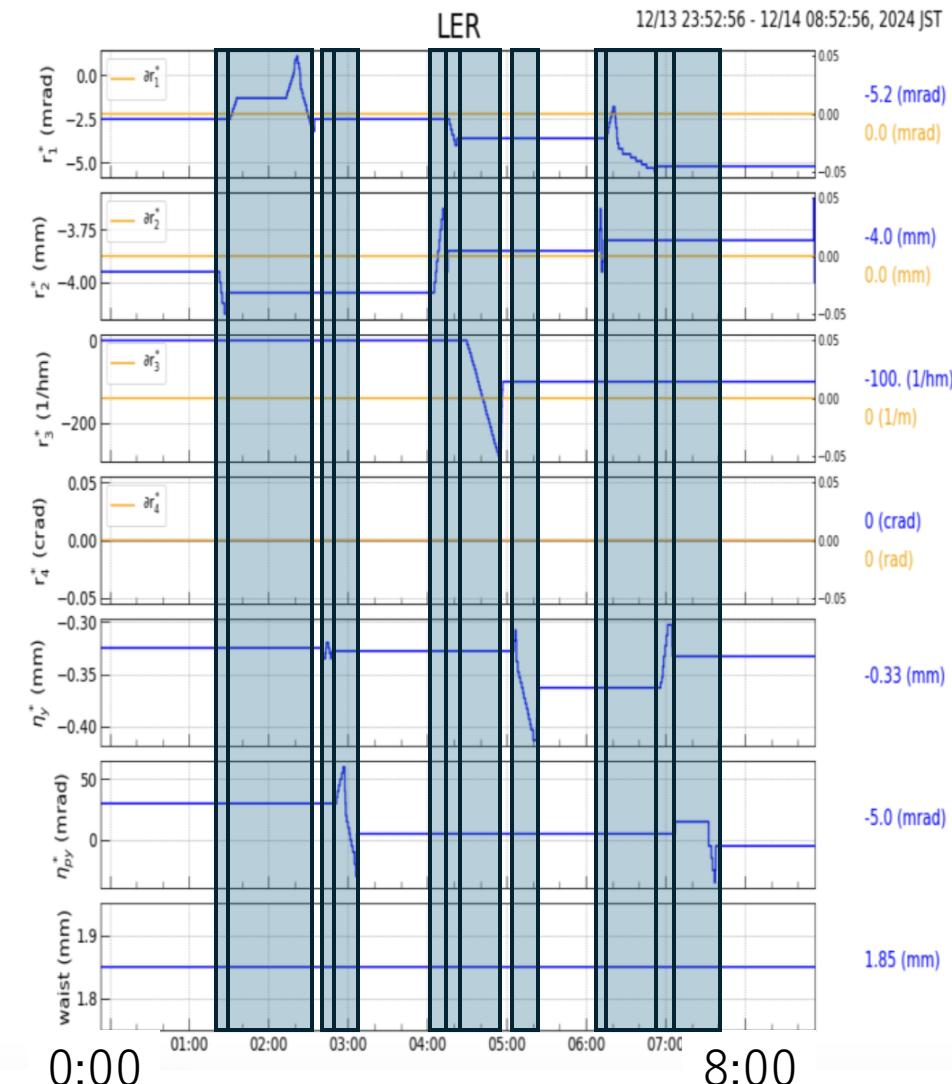
## 現在のIP knob tuning

(scan部分) オペレーターが手動で、  
一つずつIP knobを順次リニアスキャン

(set部分) オペレーターがBeam Backgroundが大き過ぎず  
Beam Sizeが小さくなり  
Luminosityが大きくなっているような  
IP knobの値を選んでセット

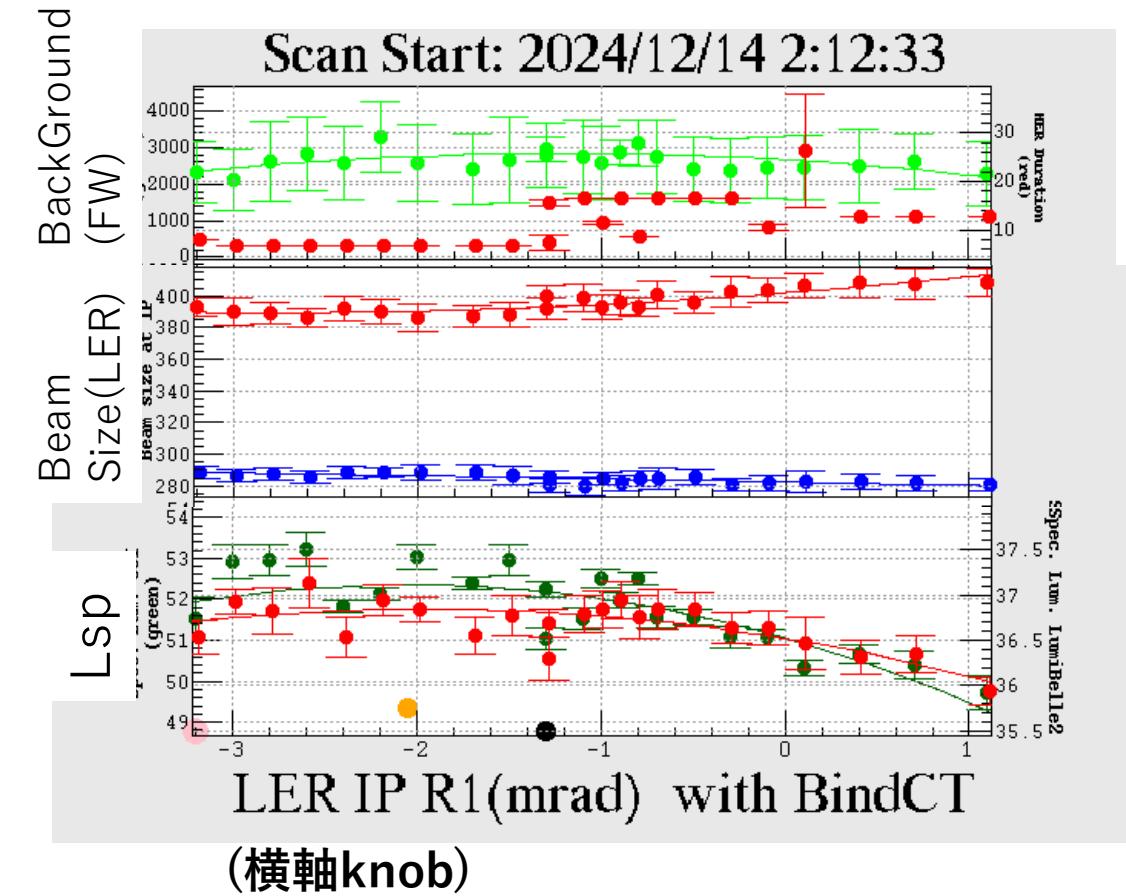
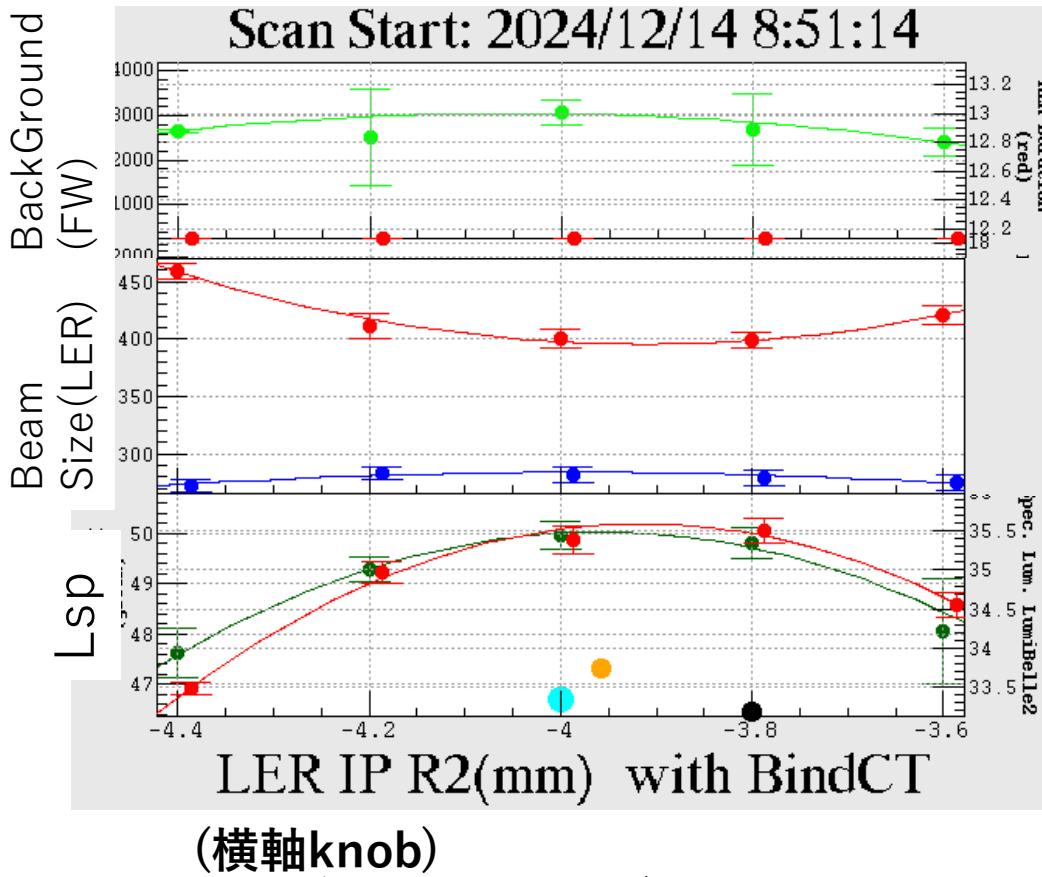
## 現在のIP knob tuningが抱える課題

- ・オペレーターによるtuningの個人差が存在する。
- ・オペレーターの拘束時間が大きい。  
(右図でも約半分ほど拘束されている)



# 研究のモチベーション

現在のIP knob tuningの例 オペレーターが見ている画面を切り取ったもの



# 研究のモチベーション

IP knob tuningに機械学習を適用して

- 現在のオペレーションよりも効率的なものにしたい
- 自動化を行いたい

## 機械学習の適用案

- (scan部分) ベイズ最適化などを採用した効率的なスキャンの実装
- (set部分) IP knobのscanを行った後どの値を選ぶかを決めるモデルの開発

# IP knob scan後にsetする値を提案するモデルの開発

## 本モデルを開発する目的

次のランで単純な1次元リニアスキャンは自動化予定  
(原さん、住澤さん)

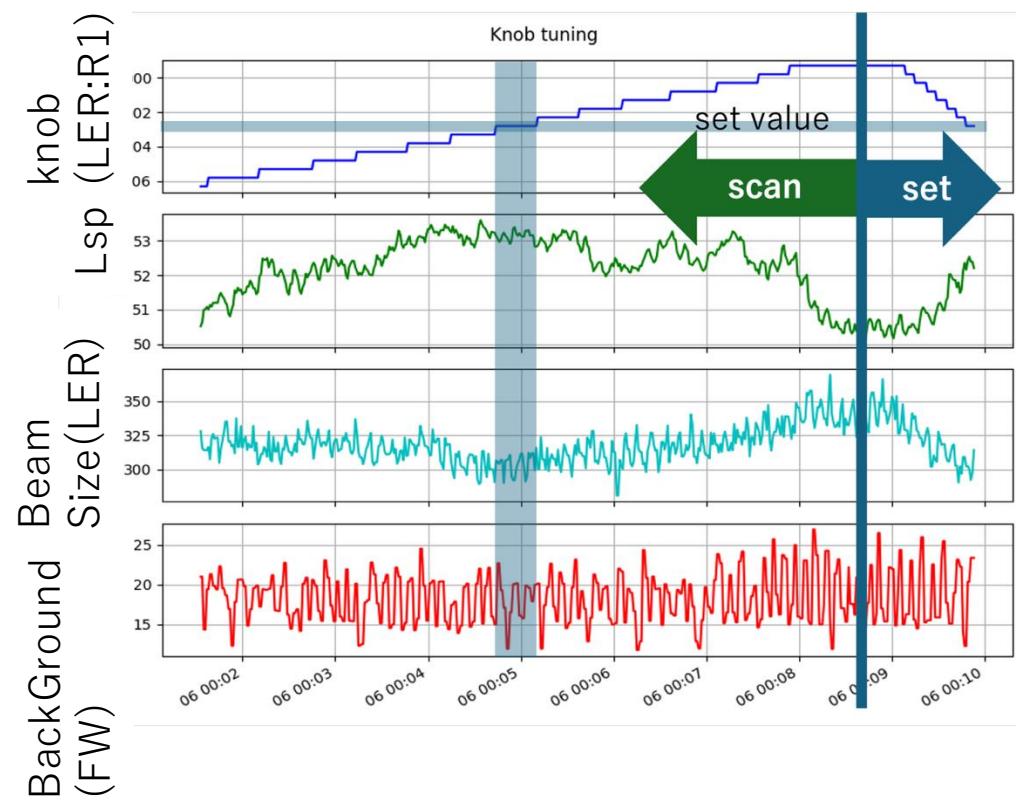
set部分をサポート出来ればさらにオペレーターの負担減少

## 機械学習を採用する理由

オペレーターはルミノシティやビームサイズ、  
バックグラウンドを総合的にみてsetする値を判断

しかし、明確化された選定基準がない

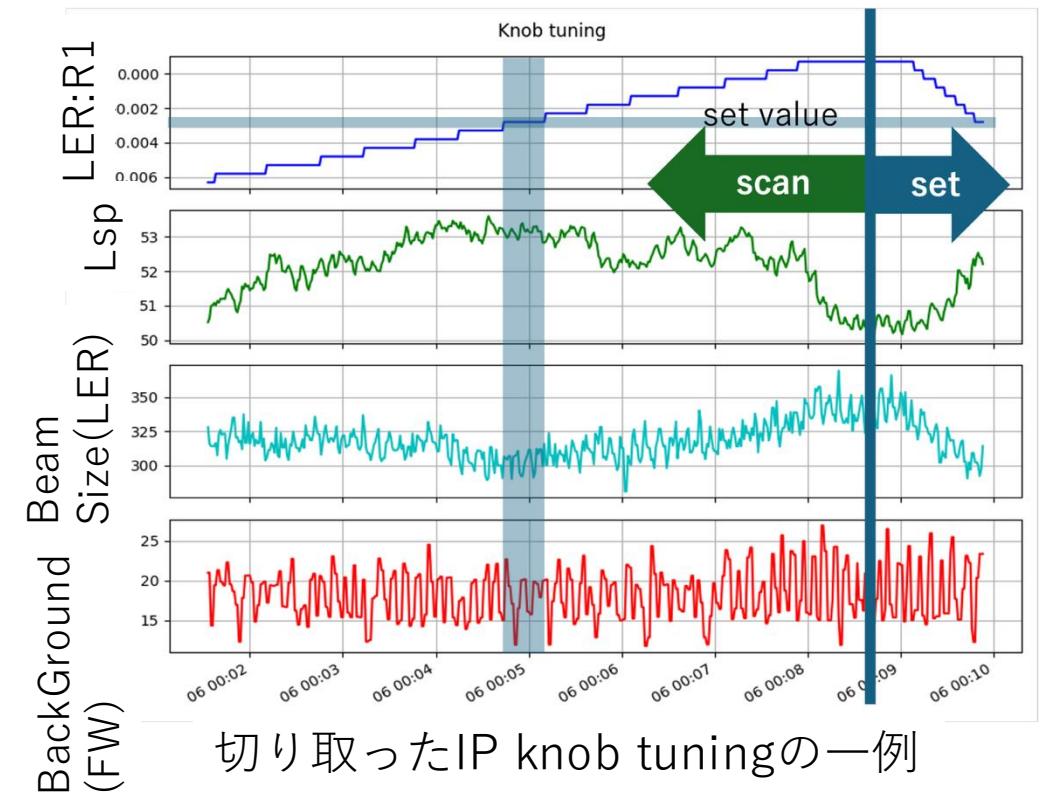
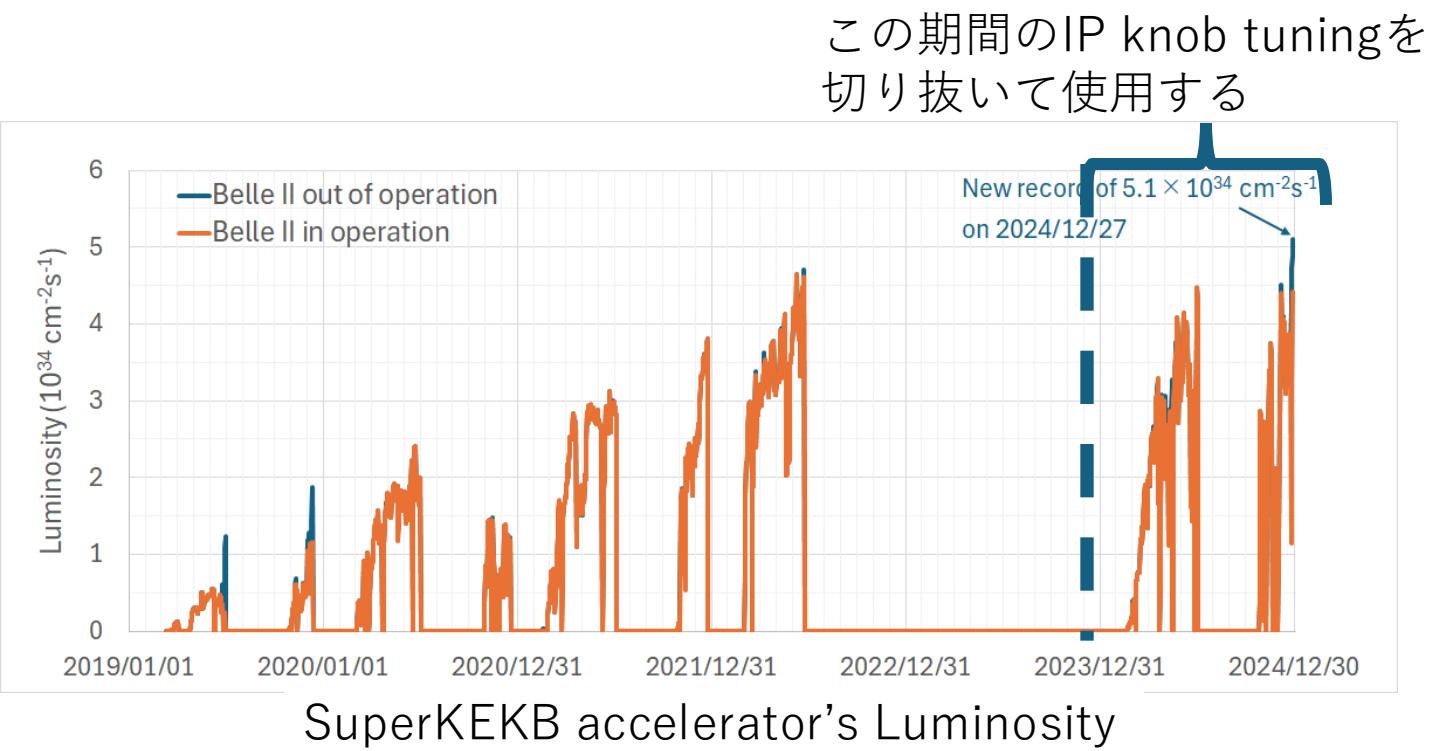
機械学習を用いて過去のオペレーターの判断を再現して  
新しいオペレーターでも安定したset値を実現したい



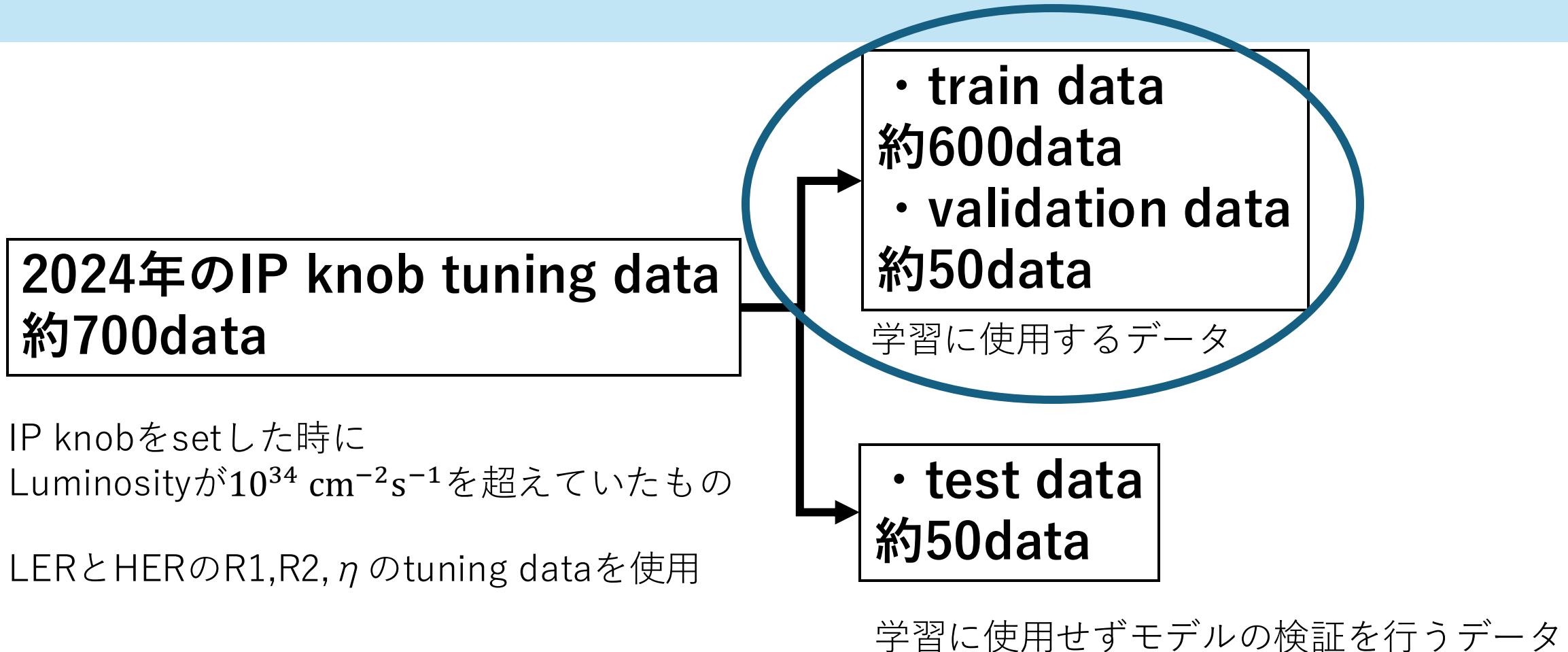
# 機械学習に使用するデータ

2024年のデータを使用

…加速器の状態がほぼ同じである期間で学習を行う  
(低ルミノシティ時のデータは除外)



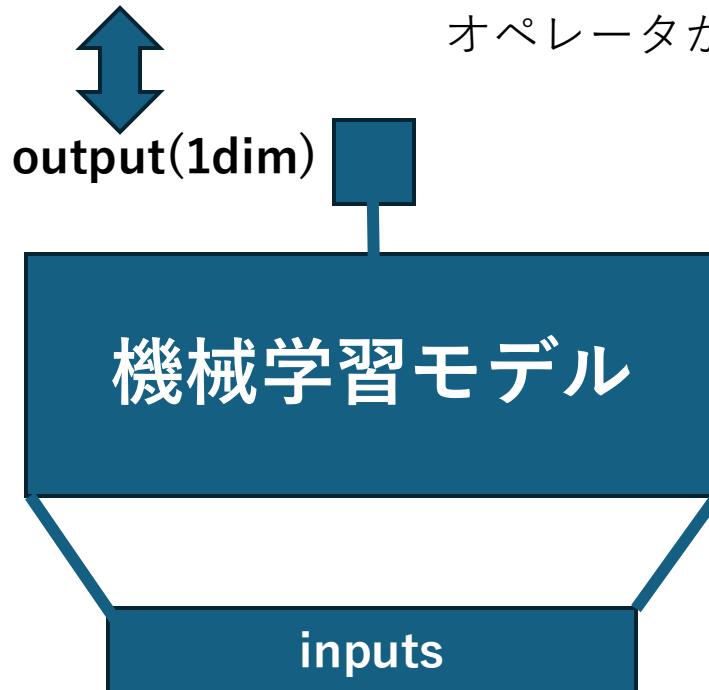
# 学習に使用するデータ



# 機械学習の準備

# 機械学習に用いるデータセット

**set value(正解値)** : 1回のスキャン後に  
オペレータが選んだ値



([6 × Scan's Time Length]dim)  
1回のスキャンで得られた各種データ  
(オペレータが判断に用いる全ての情報)

Luminosityとして**Specific Luminosity(L<sub>sp</sub>)**を使用

$$L_{sp} = \frac{L}{N_b I_b - I_{b+}}$$

→ビーム電流による単純なLuminosity増加効果を除外

## output(1dim)

**inputs**([6 × Scan's Time Length]dim) :

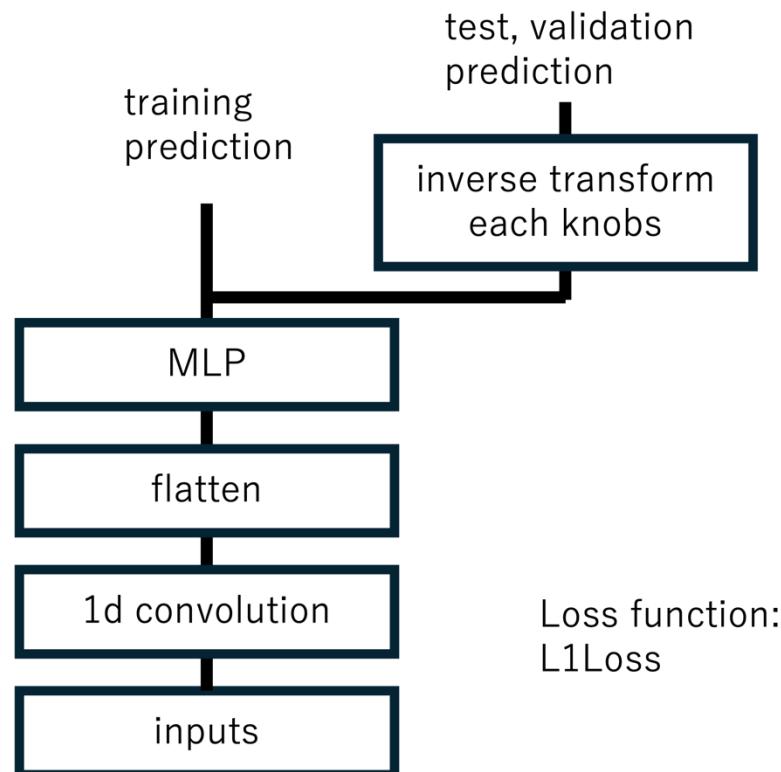
- Lsp[cm<sup>-2</sup>s<sup>-1</sup>mA<sup>-2</sup>(/10<sup>30</sup>)]
  - LER Beam Size[nm]
  - HER Beam Size[nm]
  - LER起因のBeam BackGround [Hz]
  - HER起因のBeam BackGround [Hz]
  - normalized knob parameter

**このoutputを実際のset valueに一致させることが目標**  
→set valueと最終的なモデルのoutputを比較して検証  
FPWS 2025 10

# 試したモデル(1)畳み込みニューラルネットワーク(CNN)

## 機械学習に用いるモデル

1,CNN(畳み込みニューラルネットワーク)



## モデルの特徴

- ・二層の畳み込み層を導入  
→ 加速器の応答の連続性を考慮  
→ IP knobに対する加速器の応答の揺らぎに頑健
- ・学習時に最小化させる損失関数としてL1Lossを使用  
 $(\sum |(\text{予測値}) - (\text{正解値})|)$   
→ 正解値からのずれに対して敏感

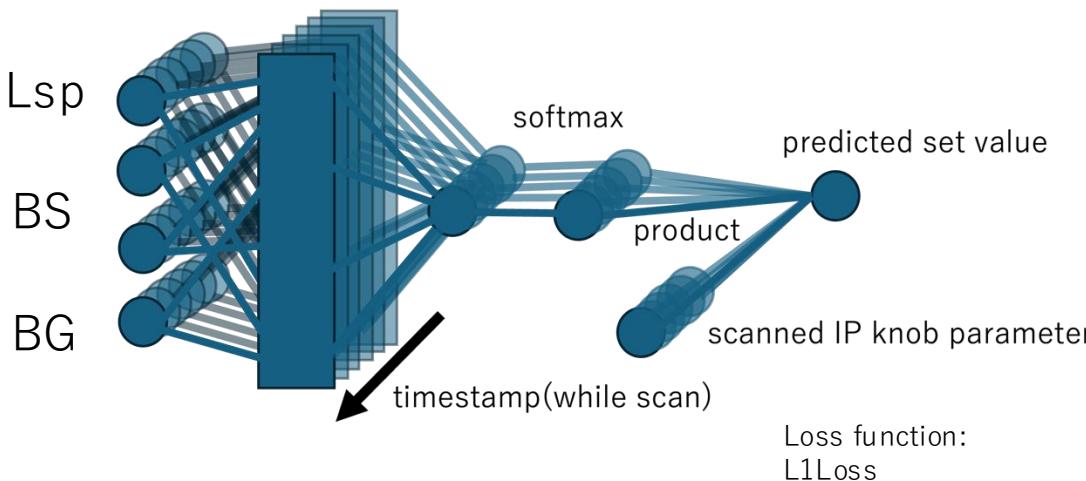
## このモデルを試した理由

- ・近いIP knobの値であれば加速器の応答も  
近い値であるという制約を付け加えられるから

# 試したモデル(2)加重和モデル(WS)

## 機械学習に用いるモデル

2, WS(加重和モデル)



Lsp : specific luminosity

BS : Beam Size

BG : BackGround

## モデルの特徴

- ・入力を変数とした評価関数を学習  
→オペレーターがどの変数を着目しているかを抽出
- ・評価関数を重みとする加重平均を出力  
→scan rangeの中から出力が選ばれるので安全
- ・学習時に最小化させる損失関数としてL1Lossを使用  
(Average(|(予測値) - (正解値)|))  
→正解値からのずれに対して敏感

## このモデルを試した理由

- ・scanに適用する機械学習の評価関数として使用可能

# 学習の評価に使用するデータ

2024年のIP knob tuning data  
約700data

IP knobをsetした時に  
Luminosityが $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ を超えていたもの  
LERとHERのR1,R2,  $\eta$  のtuning dataを使用

- train data  
約600data
- validation data  
約50data

学習に使用するデータ

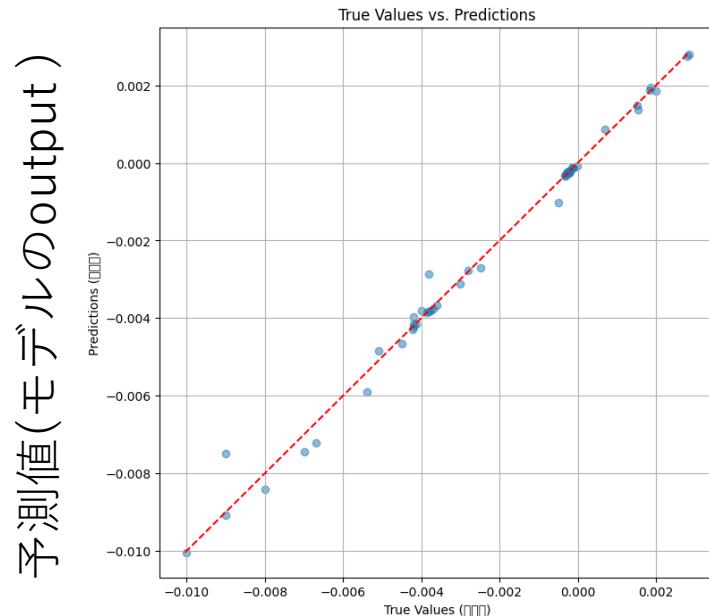
- test data  
約50data

学習に使用せずモデルの検証を行うデータ

# 機械学習の予測値と正解値の比較

Test data(=50 tuning data)を使用し、モデルを検証

1.CNN

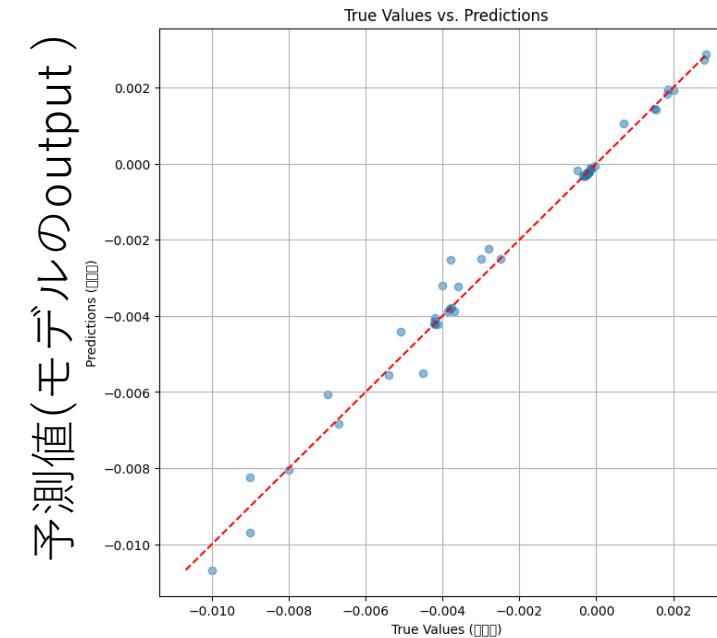


Test Loss: 0.16 (1stepの80%)

※現在のオペレーターの平均scan step width : 0.21

※Loss = Average(|(予測値) – (正解値)|)

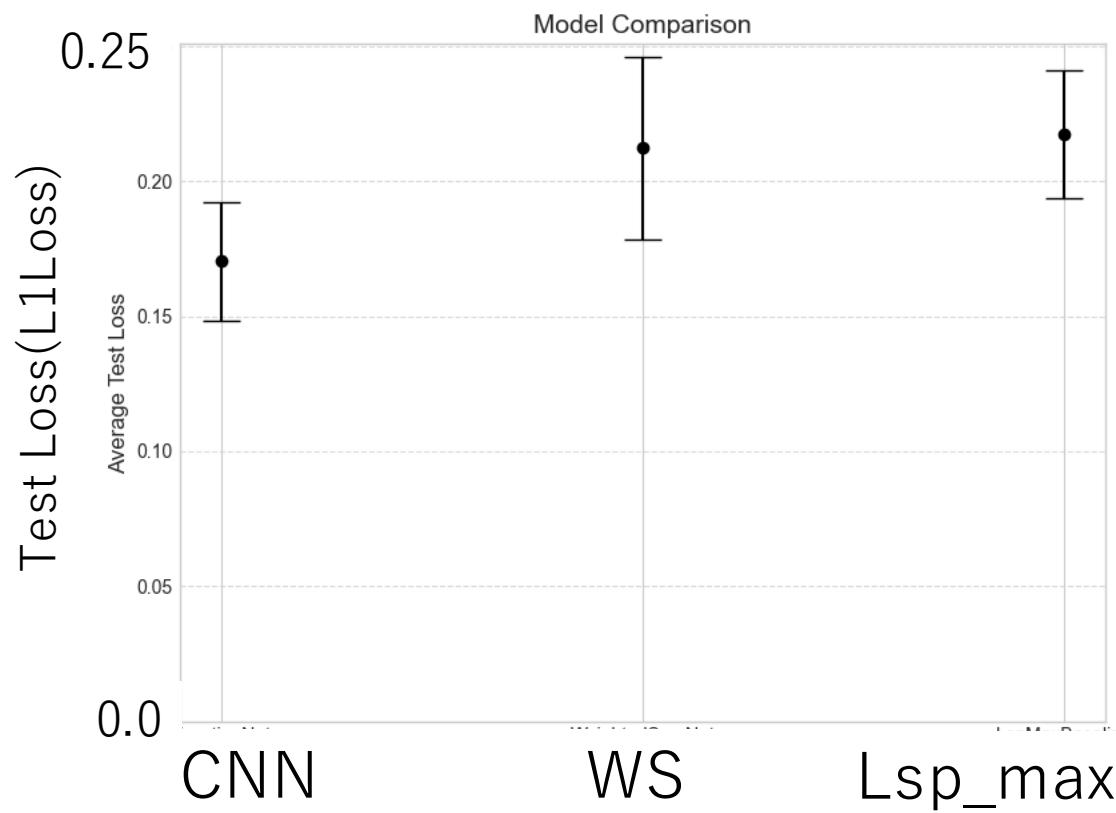
2. WS



Test Loss: 0.19 (1stepの90%)

# 機械学習の結果

## 各モデルのTest Lossの比較



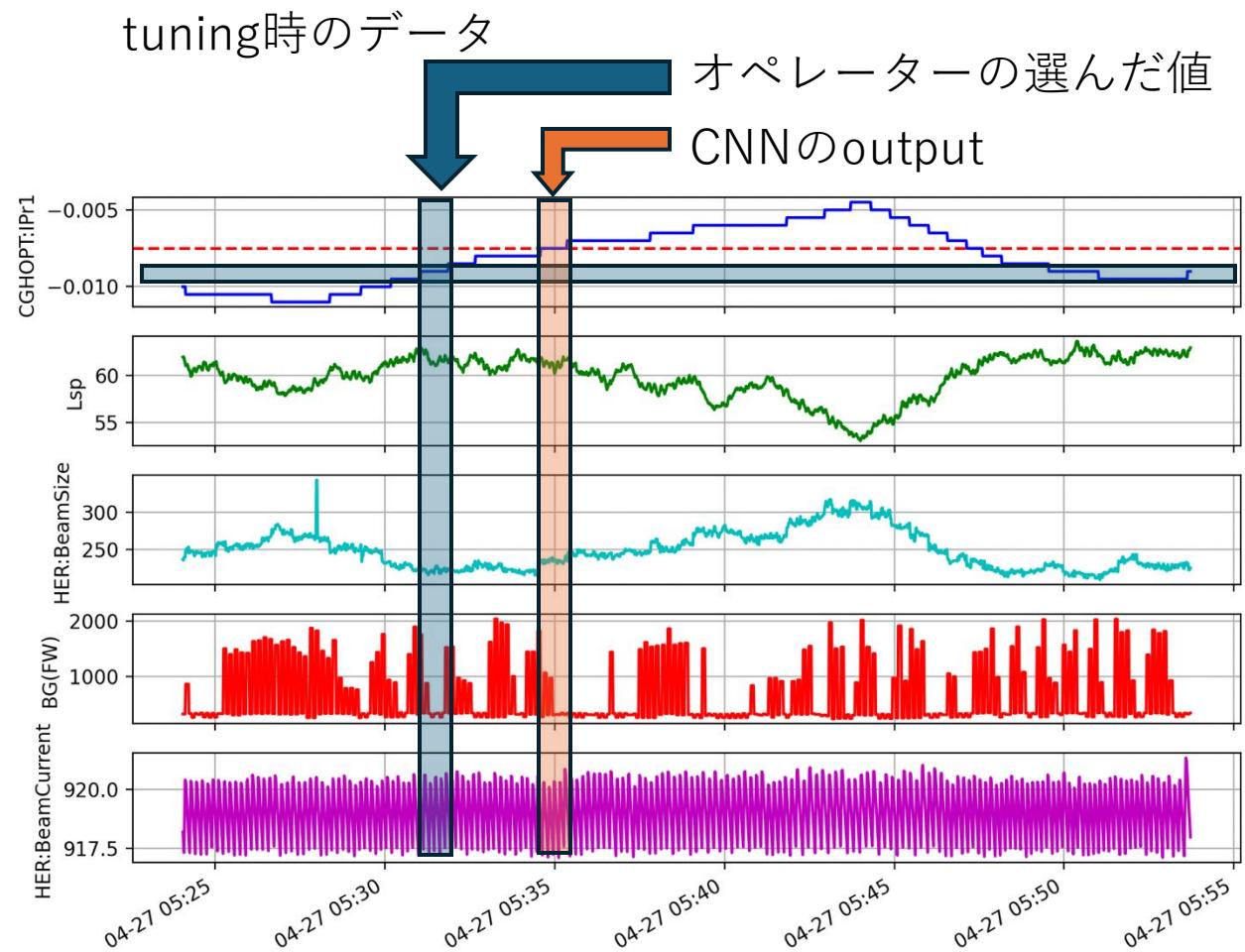
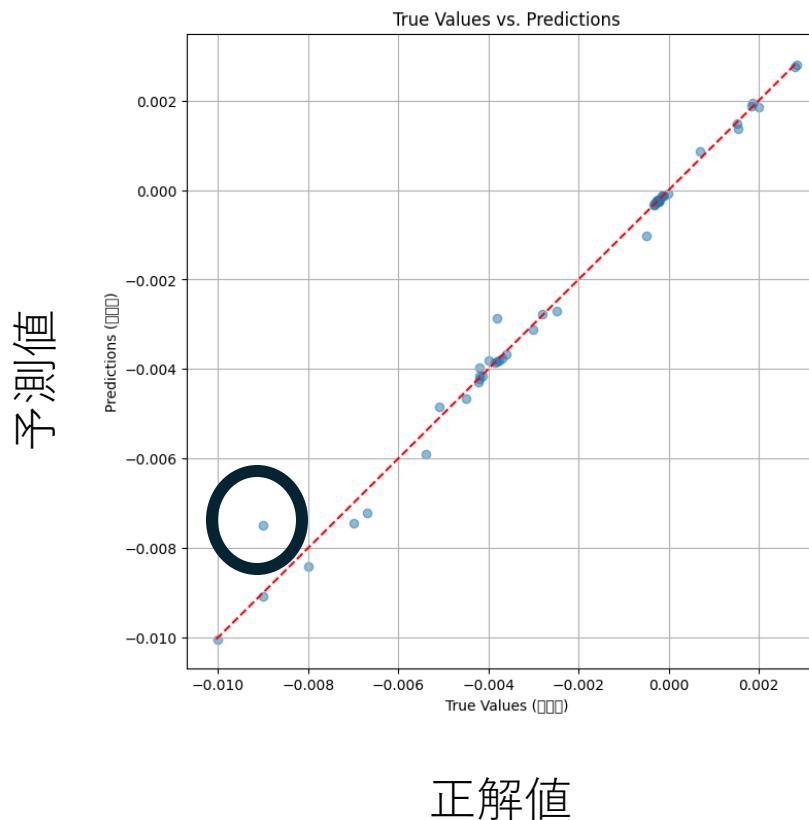
各モデルで20回検証データをシャッフルして学習を行い、それぞれの学習における Total Loss(Average(|(予測値) - (正解値)|))を計算し、分布をプロット

CNNとWSでは、  
**CNNの方がオペレーターの選択をより正確に再現できていると言える**

また、Lsp最大のところのknobを参照するよりも過去の選択を再現できている

# オペレーションとモデルのギャップ

## ミスマッチ(CNN)



モデルが惜しい値を選んでいることがわかる

# まとめと今後の展望

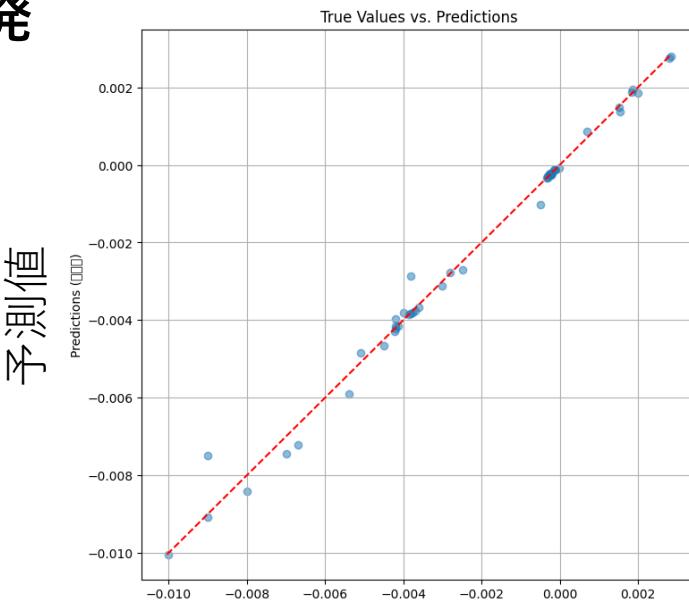
## まとめ

IP knob tuningに機械学習を導入する最初のステップとして  
**IP knobのscanを行った後どの値にセットするべきかを決めるモデルの開発**  
を行なった。

CNNを使ったモデルを用いることで良い精度で過去のオペレーター  
による最適値の選択を再現することができた。

## 今後の展望

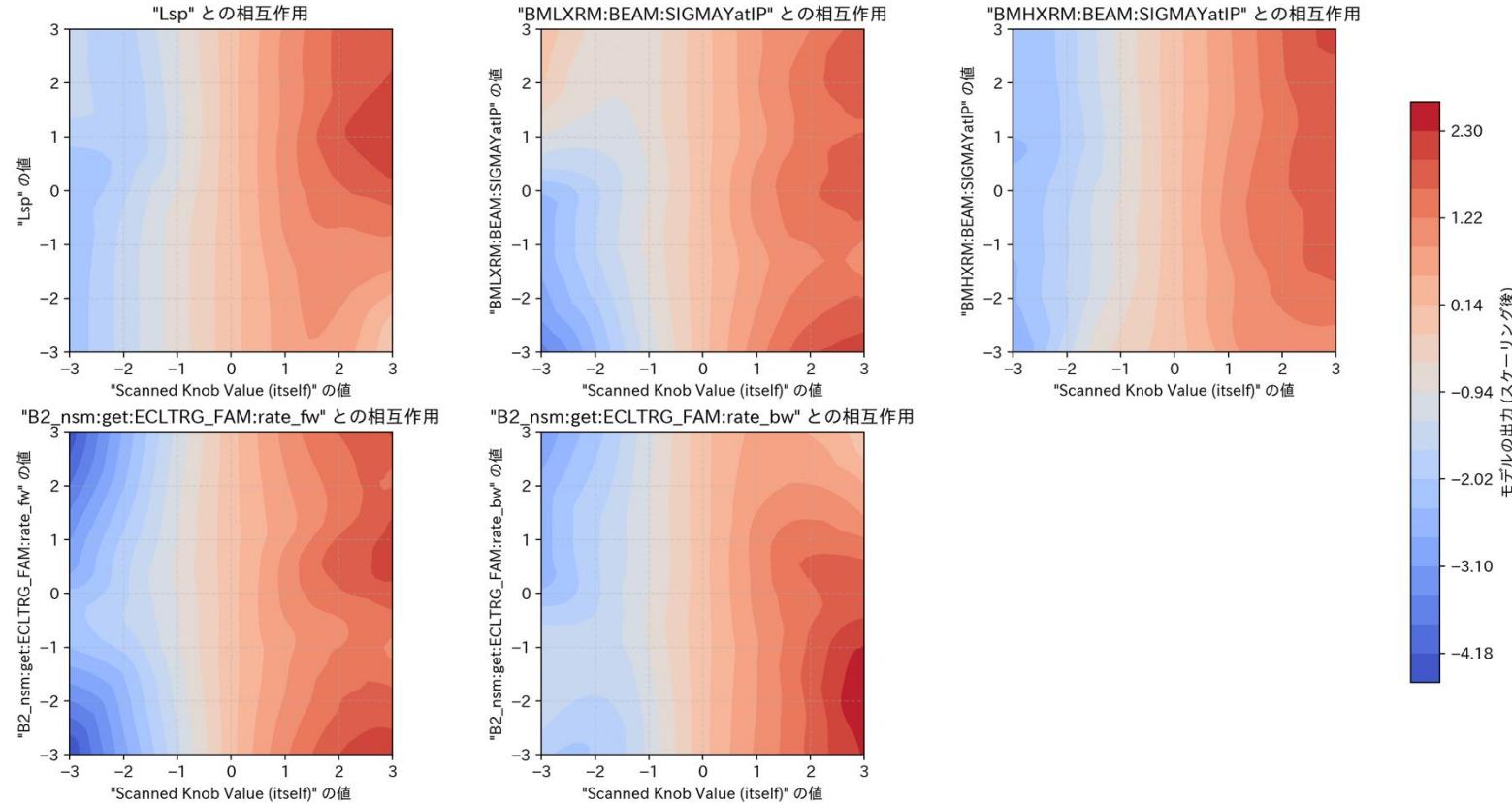
次はscan部分への機械学習の適用を進め更なる効率化を実現する。  
ベイズ最適化などを用いたscanで最適値に素早く辿り着くことを目指す。

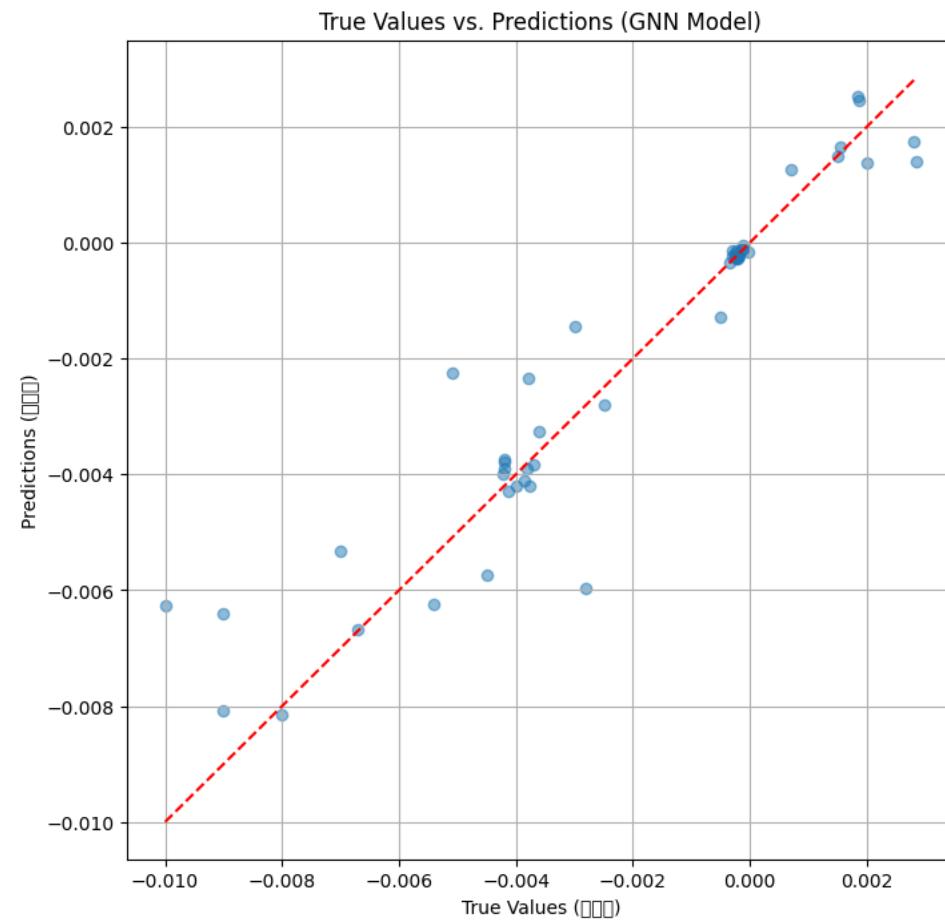


正解値

# バックアップ<sup>o</sup>

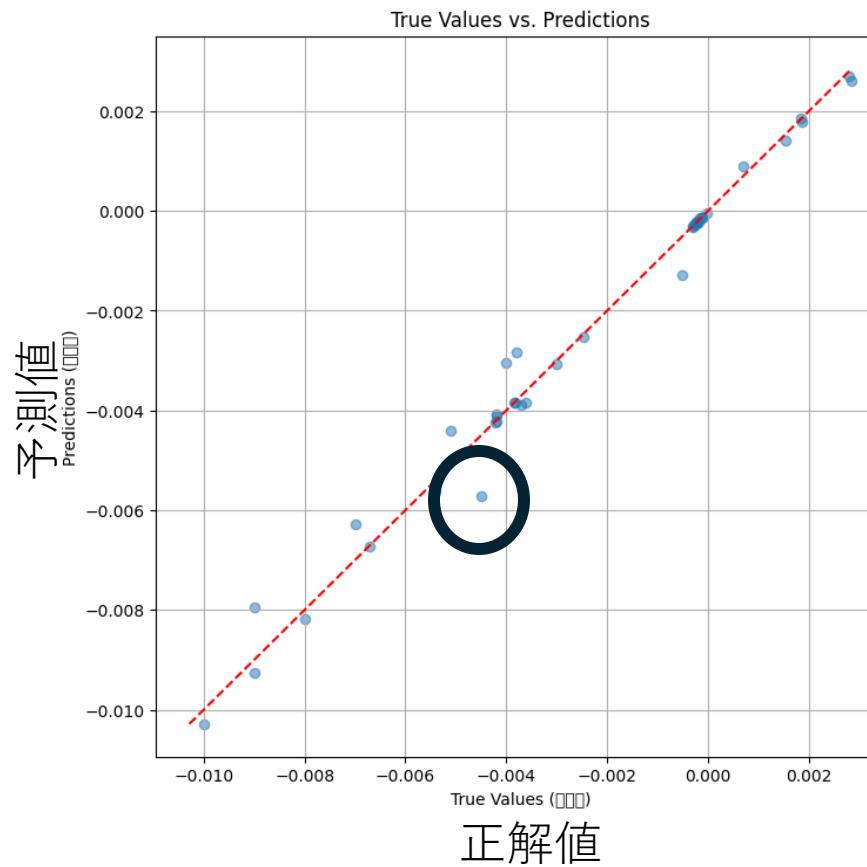
### Scanned Knob Valueと他特徴量の相互作用マップ





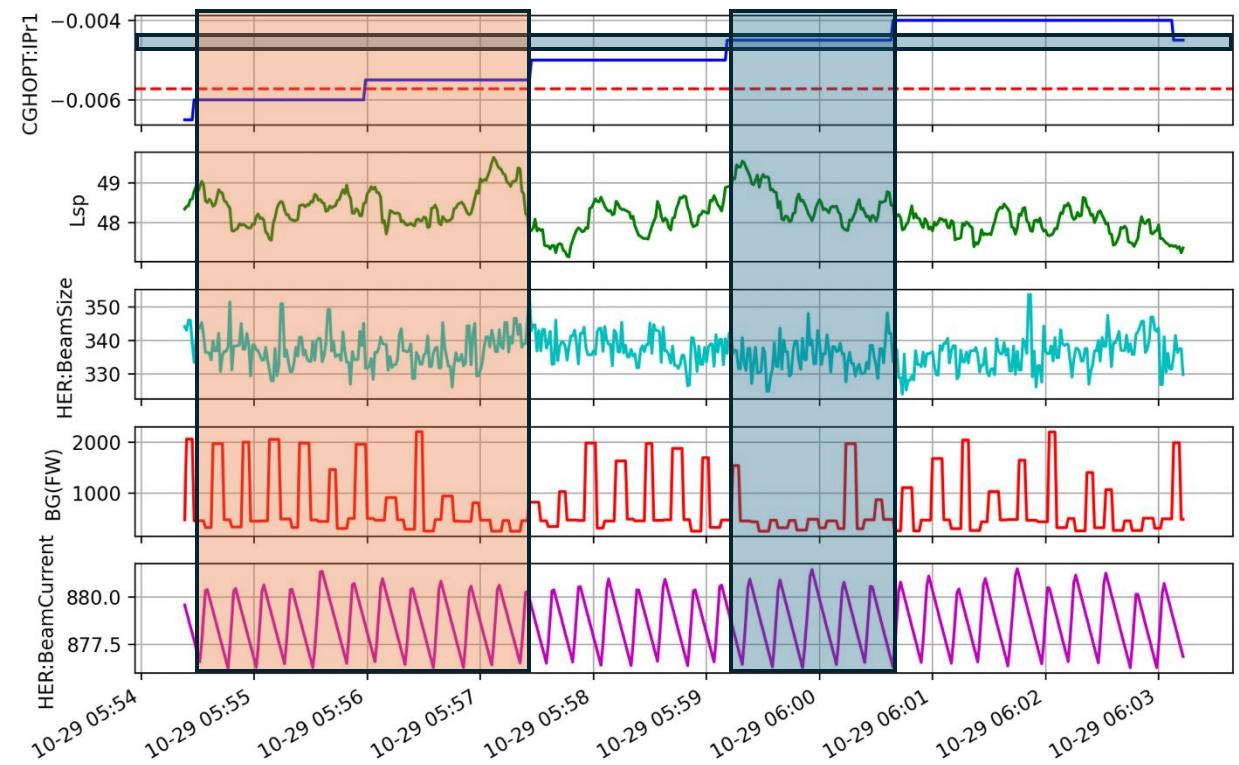
# オペレーションとモデルのギャップ

## ミスマッチその1(CNN)



tuning時のデータ

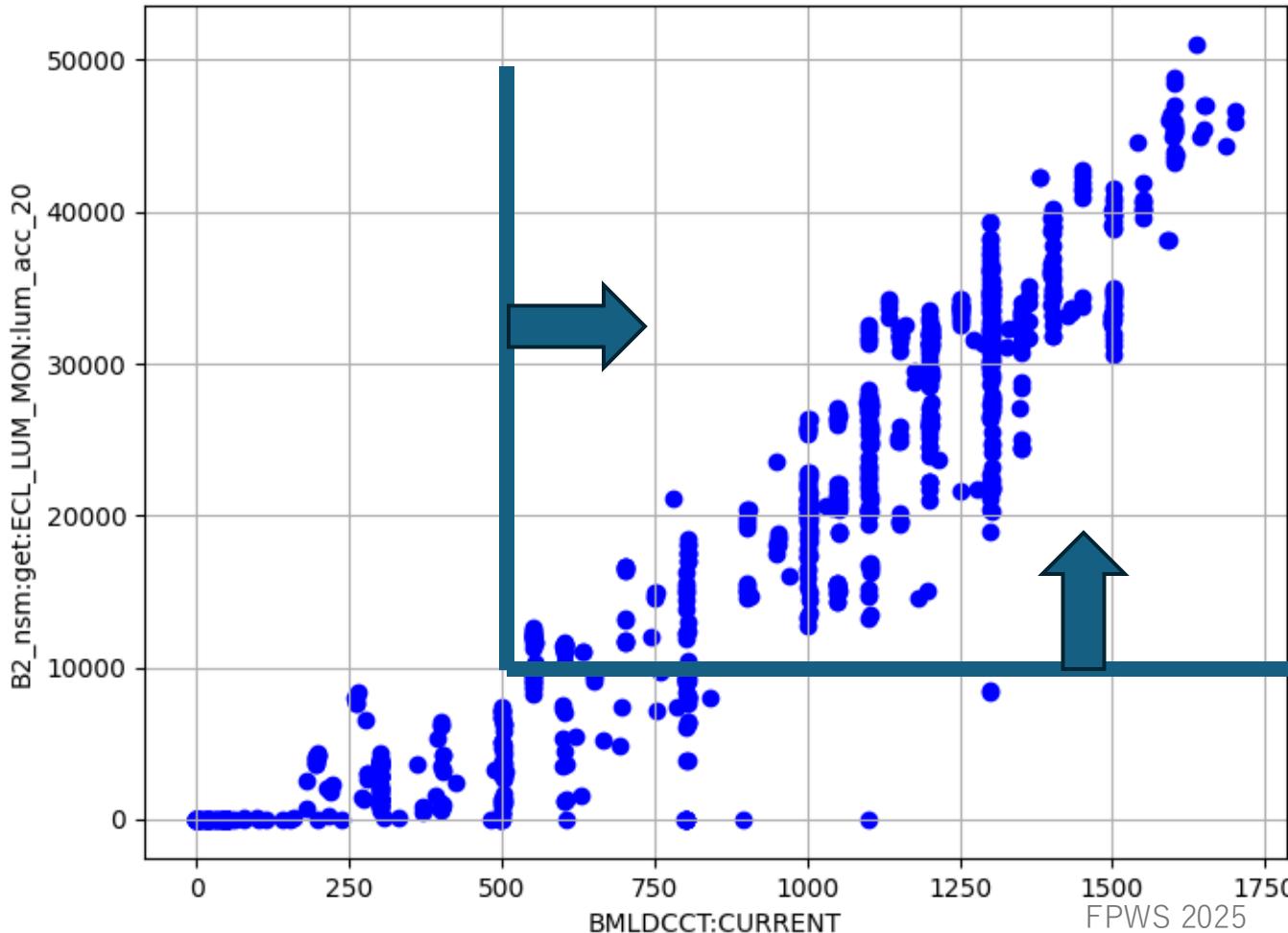
(青マーカーはオペレーターの選択、  
オレンジマーカーはモデルのoutputを表す)



逸脱した値を選んだわけではないことがわかるが、  
これだけでは判断が難しい

# 機械学習の手法

## 機械学習に用いるデータ



使用データに対するフィルター

(LER current)>500mA  
(Luminosity from ECL)> $10^{34} \text{ cm}^{-2}\text{s}^{-1}$

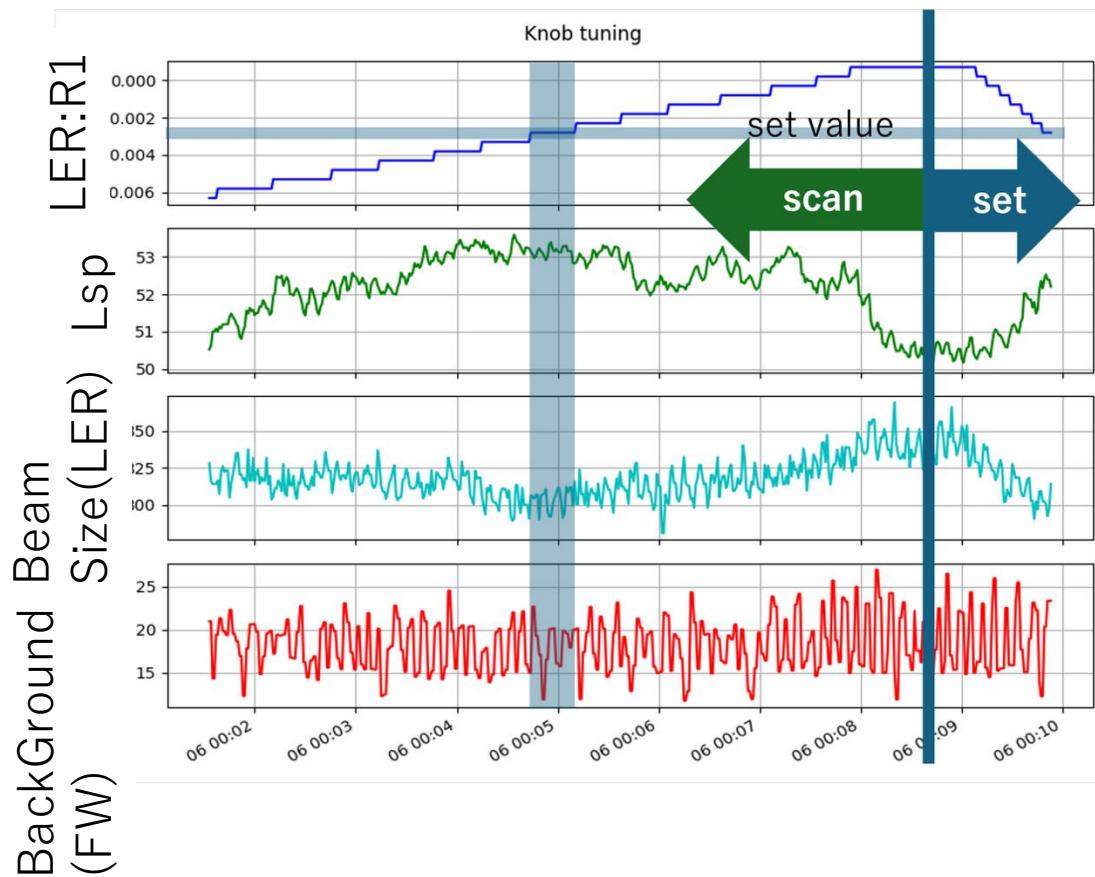
beam abortを含むなど目的の学習に悪影響を及ぼすものを除外

the number of training data : 667  
the number of validation data : 50  
the number of test data : 50

# 機械学習の準備

## 機械学習のinputの調整

例) LER R1のknob tuning



- 各scanの時間長が異なる  
→時間長方向を100segmentに分割し固定長へ変換
- 各scanの値域が異なる  
→それぞれのIP knobごとに規格化
- set valueがscanの最大値、最小値になっているものがある  
→scanが何らかの理由で打ち切られた可能性があるので除去
- ビームの調整でknobの値を0にするようなものがある  
→set valueが0であるものを除外

In the normal coordinate  
(no xy-coupling space)

$$\begin{pmatrix} X \\ P_x \\ Y \\ P_y \end{pmatrix} + \begin{pmatrix} \eta_x \\ \eta'_x \\ \eta_y \\ \eta'_y \end{pmatrix} \delta = \begin{pmatrix} \mu & 0 & -R_4 & R_2 \\ 0 & \mu & R_3 & -R_1 \\ R_1 & R_2 & \mu & 0 \\ R_3 & R_4 & 0 & \mu \end{pmatrix} \begin{pmatrix} x \\ p_x \\ y \\ p_y \end{pmatrix}$$

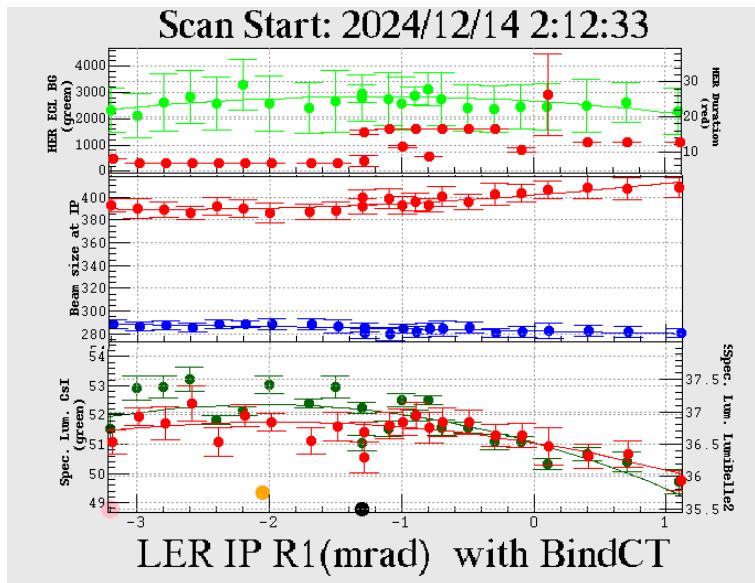
$$\mu^2 + (R_1 R_4 - R_2 R_3) = 1$$

$$\delta = \Delta p/p.$$

In the physical coordinate  
(real space)

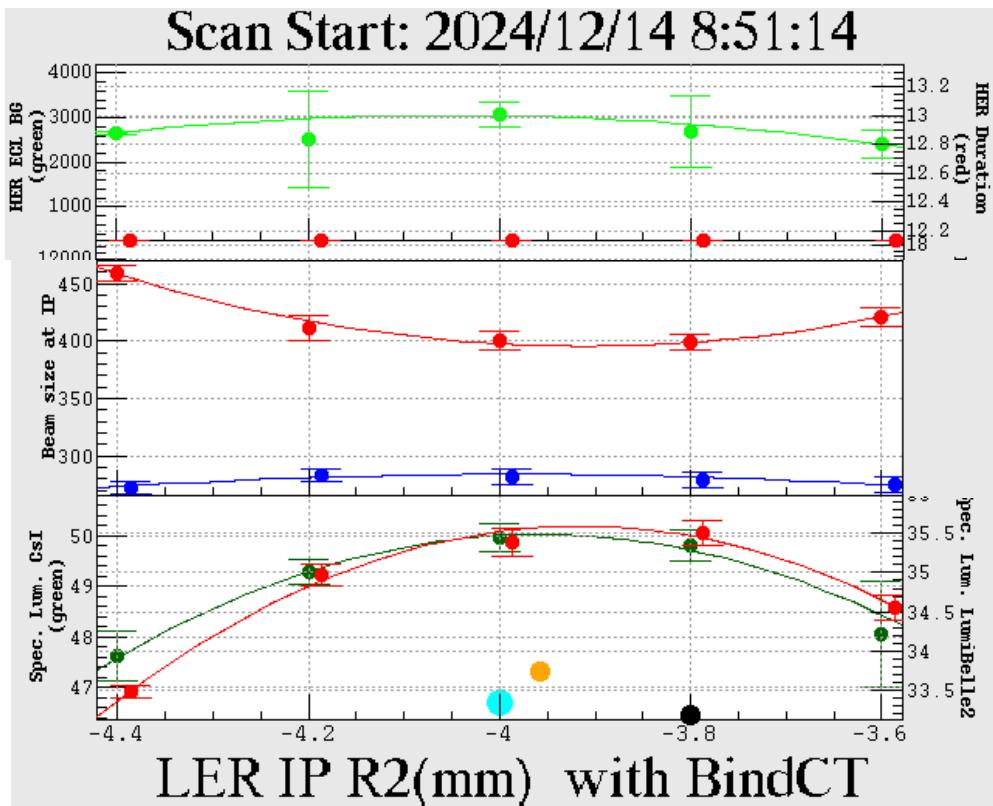
$$\begin{pmatrix} \eta_x \\ \eta'_x \\ \eta_y \\ \eta'_y \end{pmatrix} = \begin{pmatrix} \mu & 0 & -R_4 & R_2 \\ 0 & \mu & R_3 & -R_1 \\ R_1 & R_2 & \mu & 0 \\ R_3 & R_4 & 0 & \mu \end{pmatrix} \begin{pmatrix} \eta_x \\ \eta'_x \\ \eta_y \\ \eta'_y \end{pmatrix}$$

from “Single Knob Scan”(iida san)



FPWS 2025

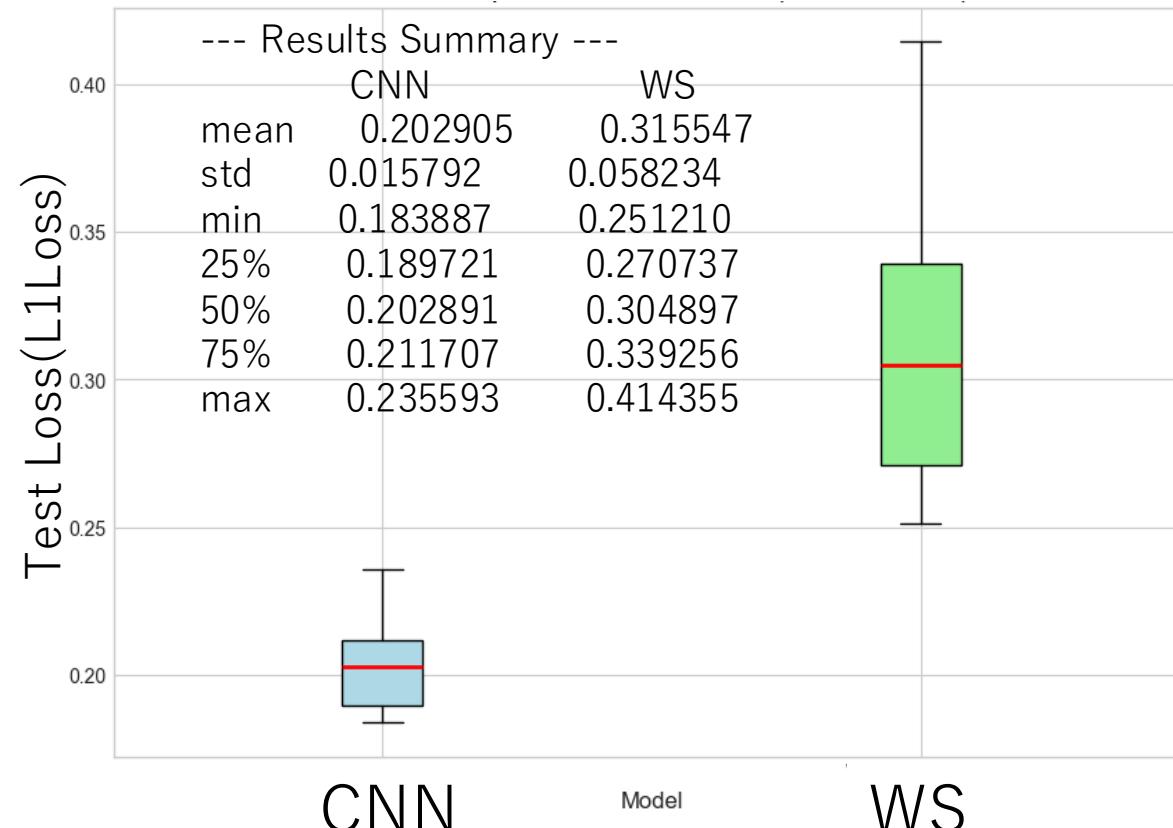
$$L = \frac{1}{2e\epsilon_e} \frac{\gamma_{\pm} I_{\pm}}{\beta_{y\pm}^*} \frac{2\sigma_{y\pm}^* \sigma_{x\pm}}{\Sigma_y \Sigma_x} \xi_{y\pm}^{ih} \frac{R_L}{R_{\xi y}^{\pm}}$$



26

# 機械学習の結果

## 各モデルのTest Lossの比較



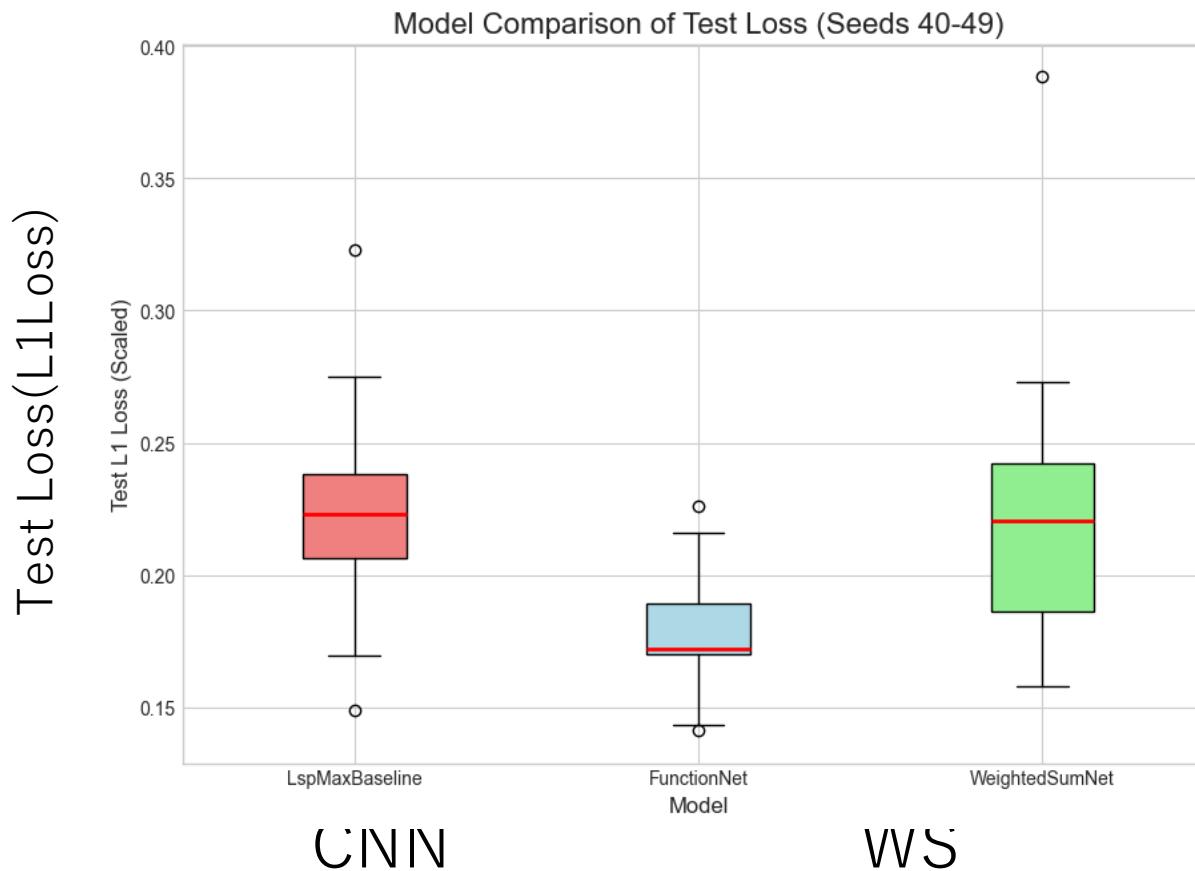
各モデルで異なる10個のシード値でTest Lossを計算  
その分布を箱ひげ図に表示

…シード値を変えると  
Train data, Validation data, Test data  
がシャッフルされる

CNNとWSでは有意な差が開いており、  
**単純な一次元scanであればCNNの方がオペレーターの選択をより正確に再現できていると言える**

# 機械学習の結果

## 各モデルのTest Lossの比較



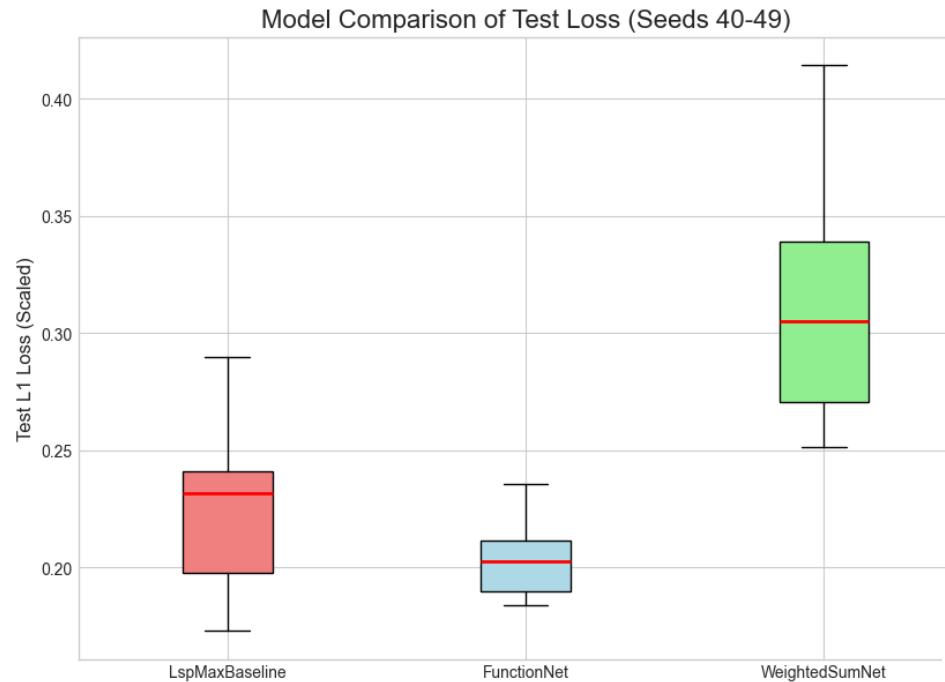
seq15

各モデルで異なる10個のシード値でTest Lossを計算  
その分布を箱ひげ図に表示

…シード値を変えると  
Train data, Validation data, Test data  
がシャッフルされる

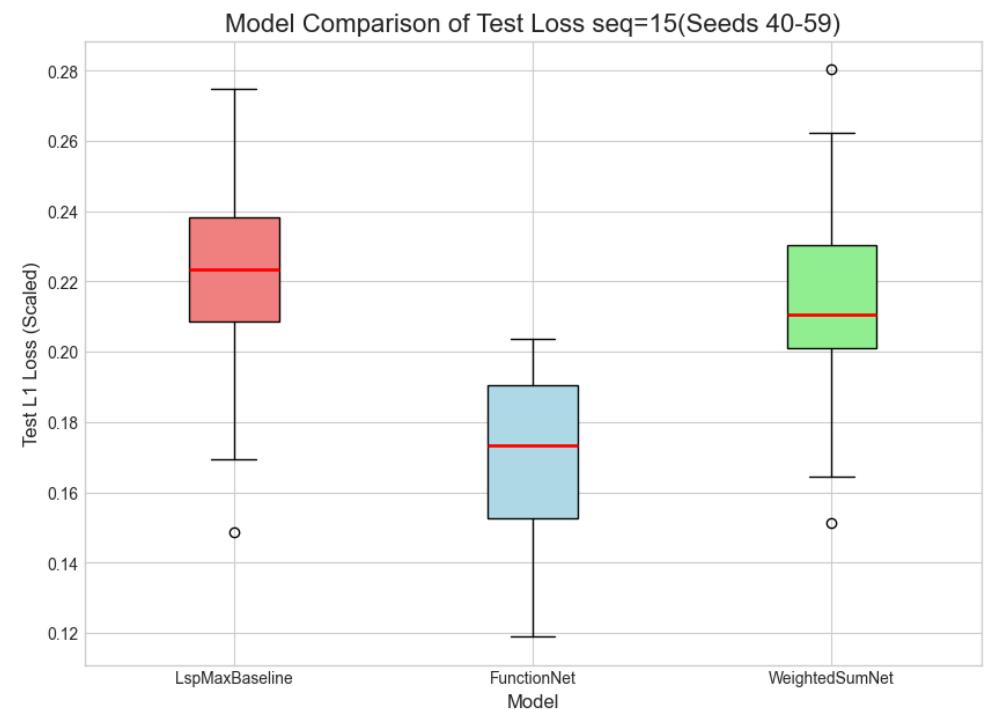
CNNとWSでは有意な差が開いており、  
**単純な一次元scanであればCNNの方がオペレーターの選択をより正確に再現できていると言える**

seq=100



CNN

seq=20

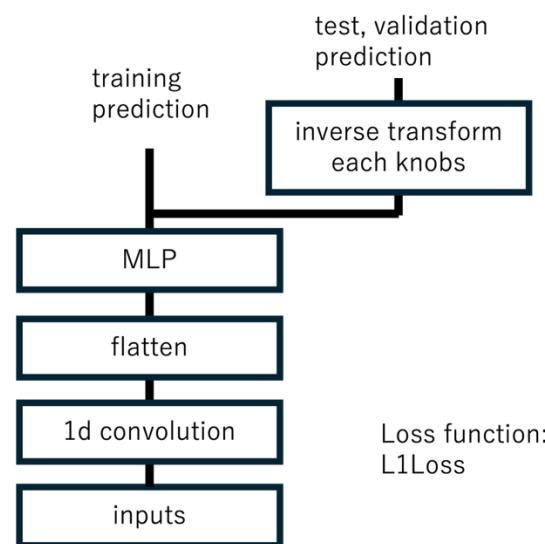


WS  
(加重和モデル)

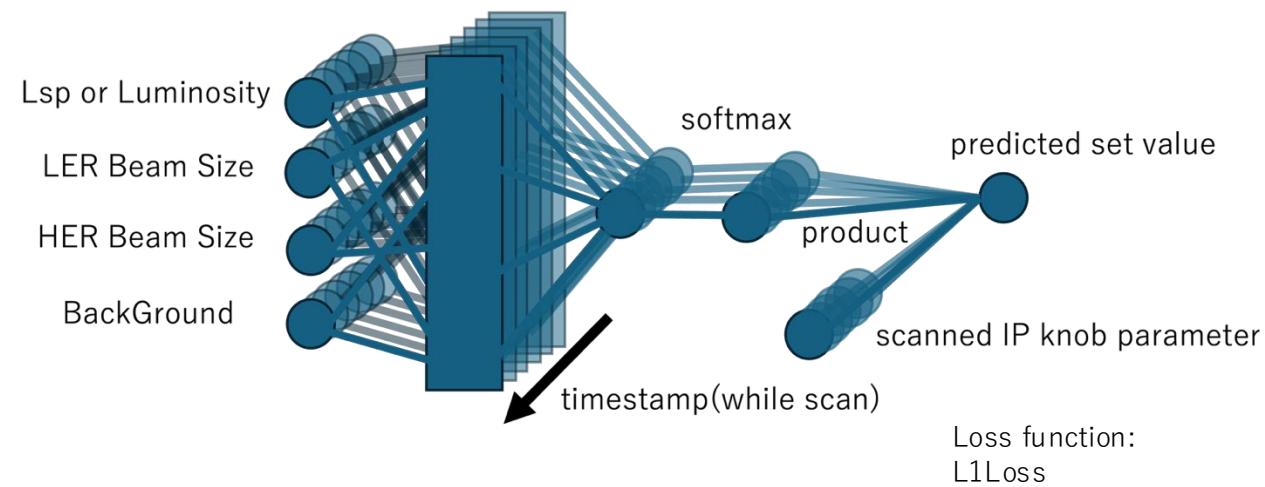
# 機械学習のモデル

## 機械学習に用いるモデル

### 1,CNN(畳み込みニューラルネットワーク)



### 2,加重平均モデル



メリット：時系列方向の連続性を取り込める

メリット：アウトプットが必ずscan rangeの中にいる

# 研究内容

## IP knobのscanを行った後どの値にセットするべきかを決めるモデルの開発

まずはIP knob tuningの半自動化を目指す

現在のオペレーションでは

scan部分…

あるIP knobをグリッドスキャンする



グリッドスキャンを自動化（住澤さん、原さん）

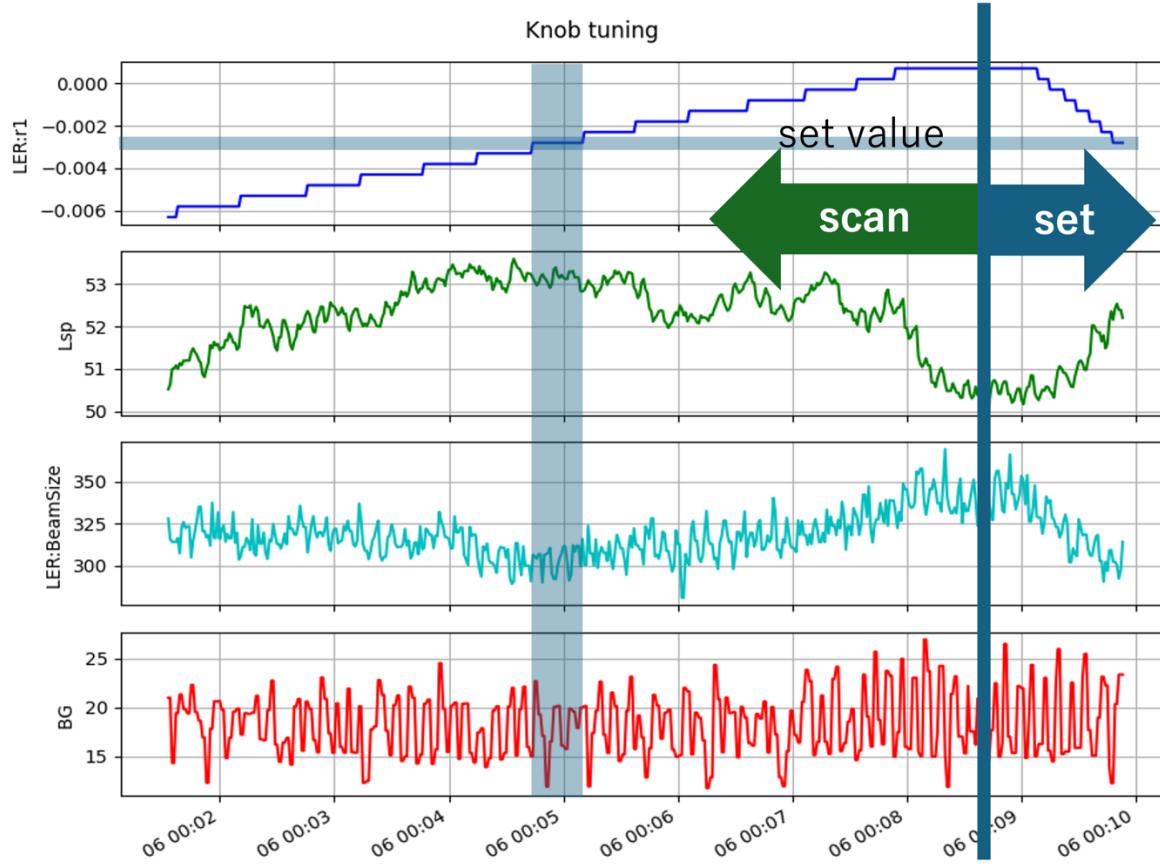
set部分…

オペレーターが良さそうな値でセットする

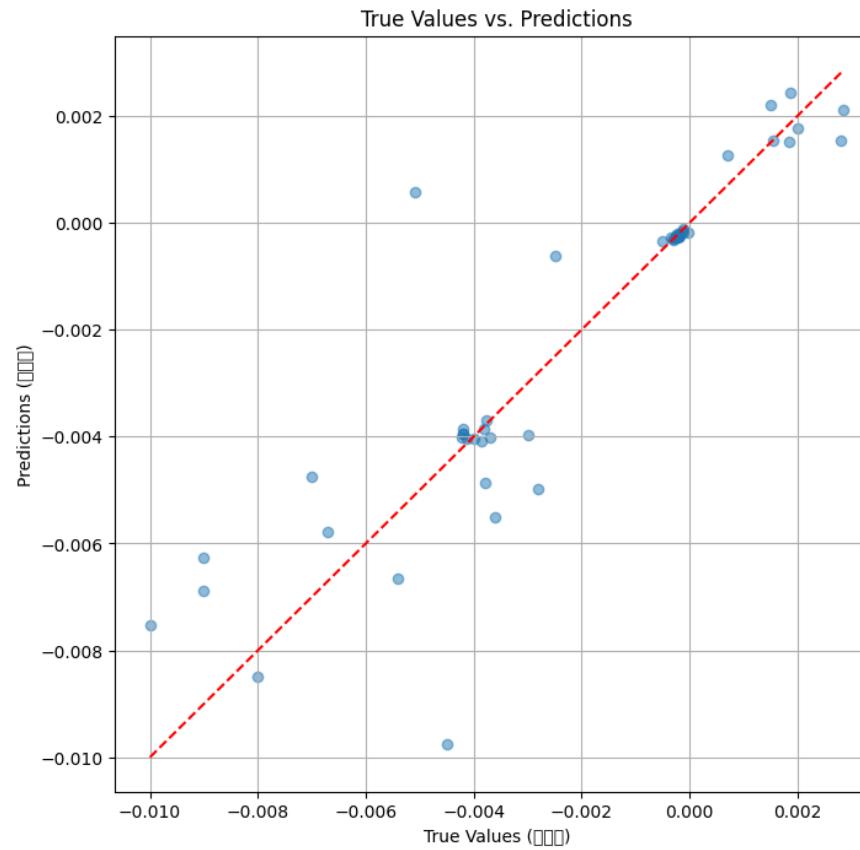
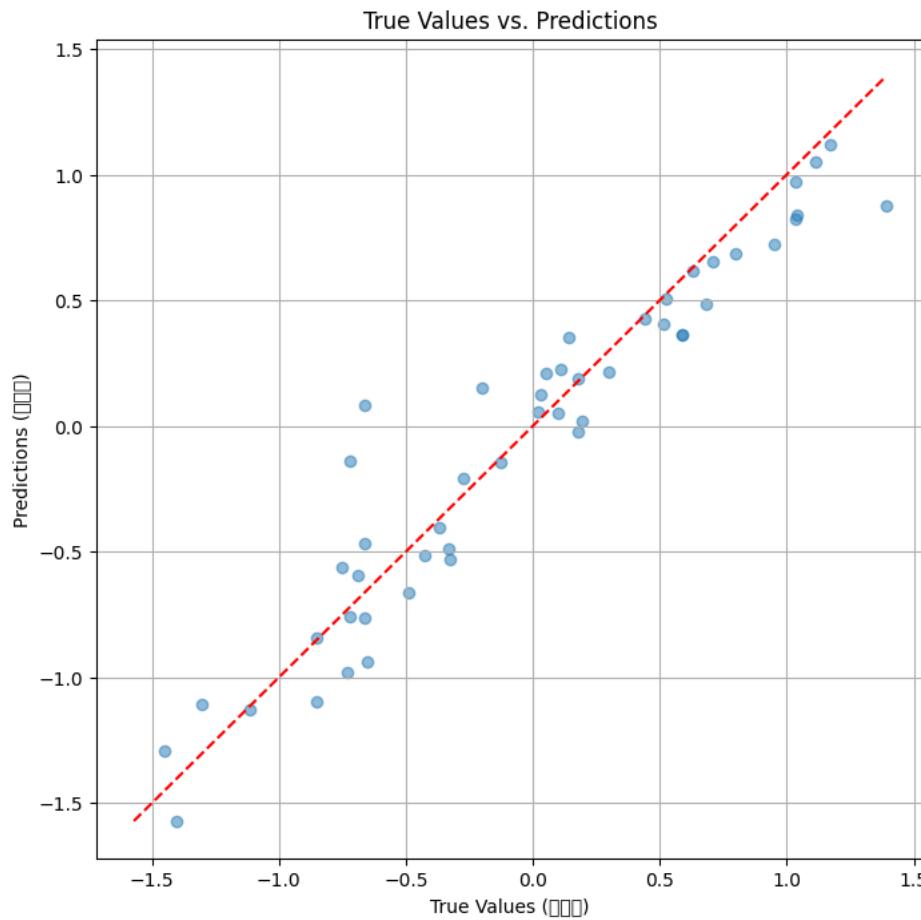


ここの人dependな部分を減らしたい

今までのIP knob tuningでのset部分を模倣するようなモデルを機械学習で開発したい

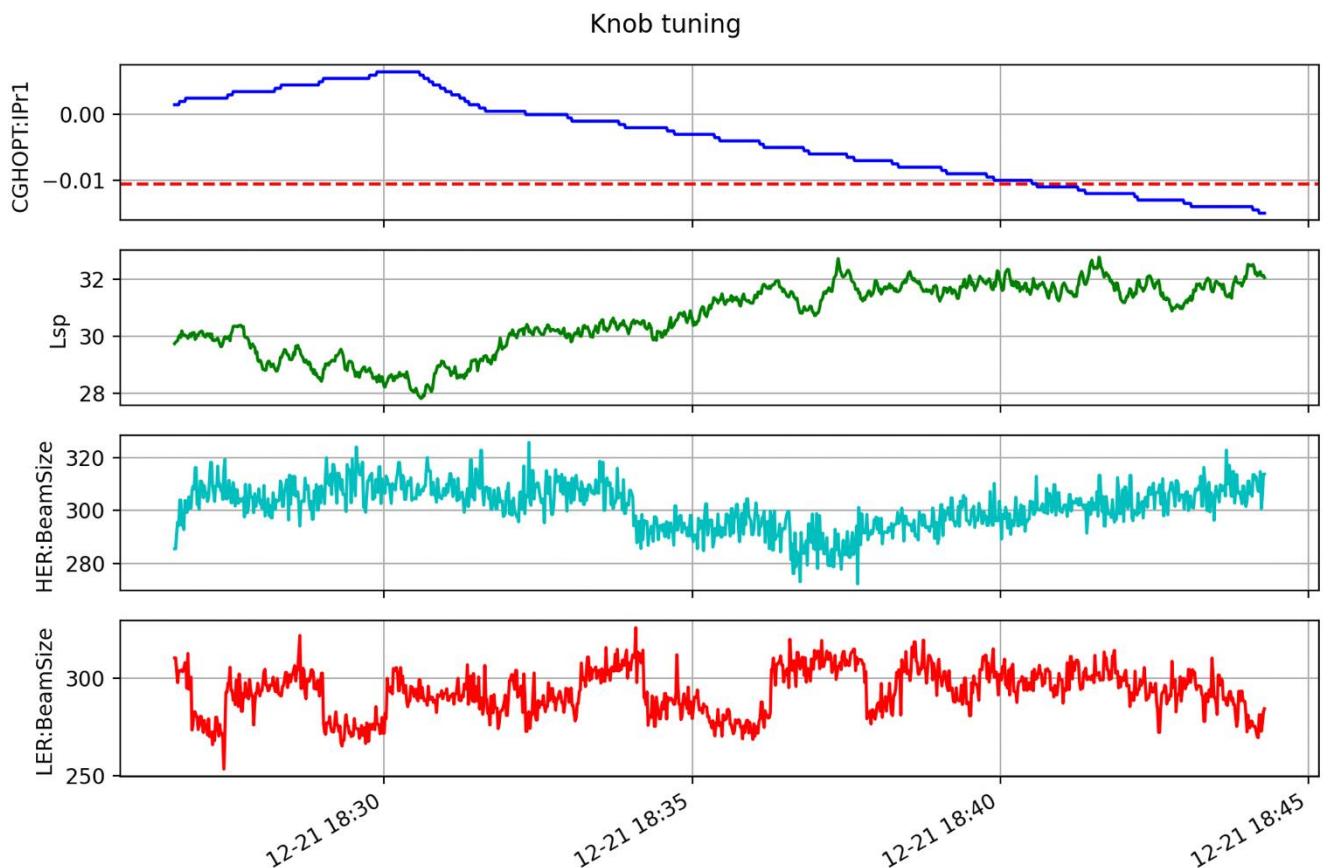
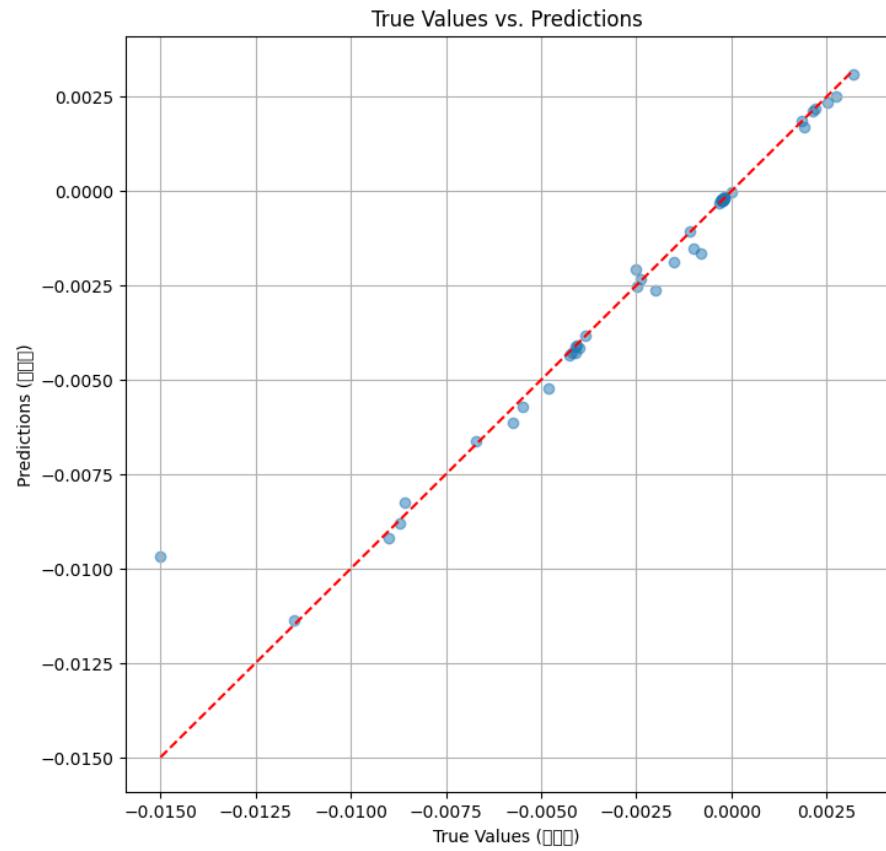


## scaled result



BSのみ見る

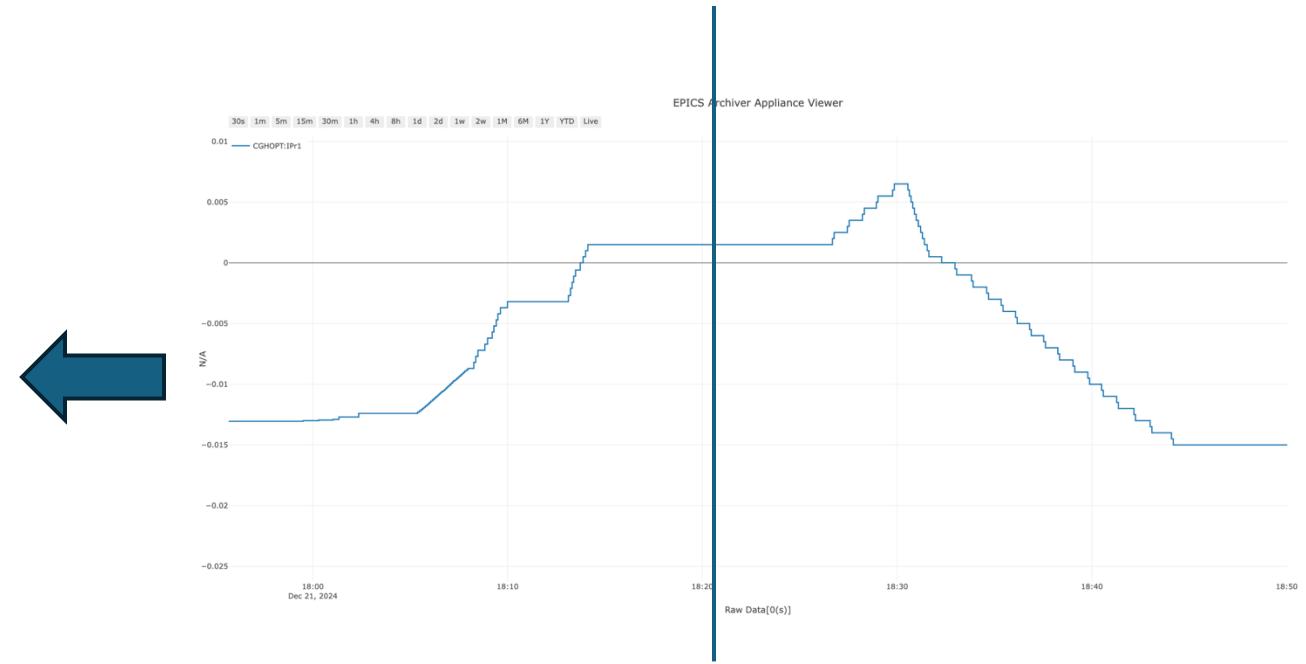
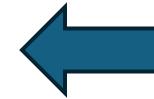
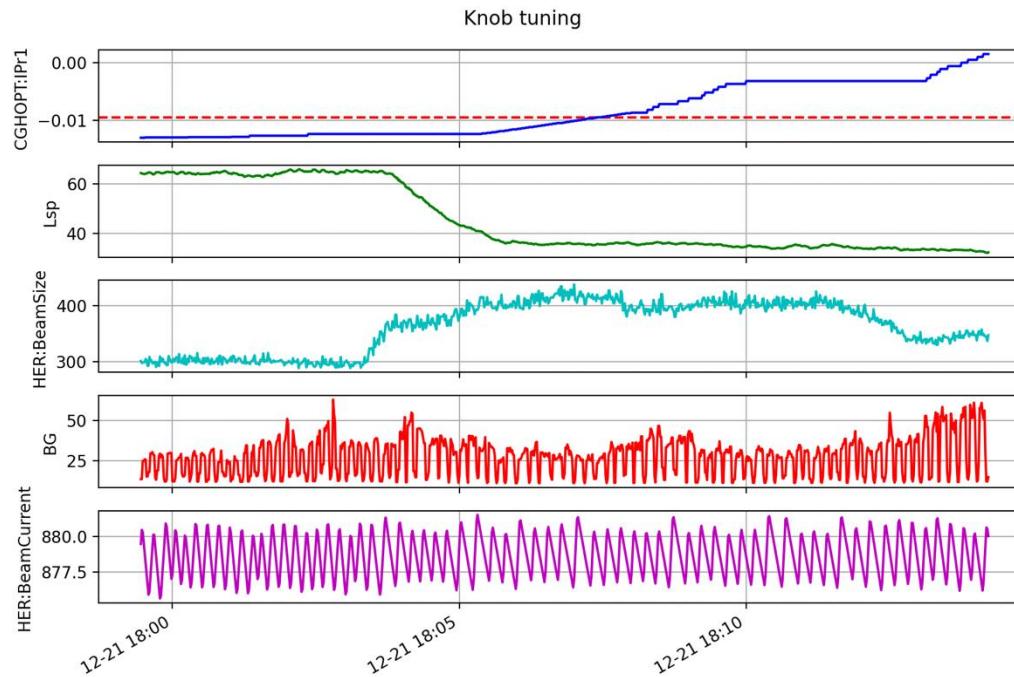
test loss 0.19...



# 手法

## 機械学習に用いるデータ

不自然なスキャンをしているものも除外(例：HER : IPR1)



## その他のフィルター

- 0setしたものを除外するなど

# 手法

## 機械学習に用いるデータセット

inputs(6dim): Lsp, Beam Size(LER,HER), BG(ECL), scanned knob parameter

- scanning timeはバラバラなので時間方向はsegment化(1batchあたり100x6の入力)
- IP knobごとにscan rangeが異なる  
→スケーラーをIP knobごとに別々で用意

メリット：

- 異なるscan rangeに柔軟に対応可能
- スケーラーもデータセット内で保持することでアウトプットからの復元も可能

デメリット：

- スキャン回数が少ないIP knobはtrain data, valid data, test dataに分割できない

