

Forecasting the **CLEAR** beam charge

How will the beam behave next, and can we see it coming?

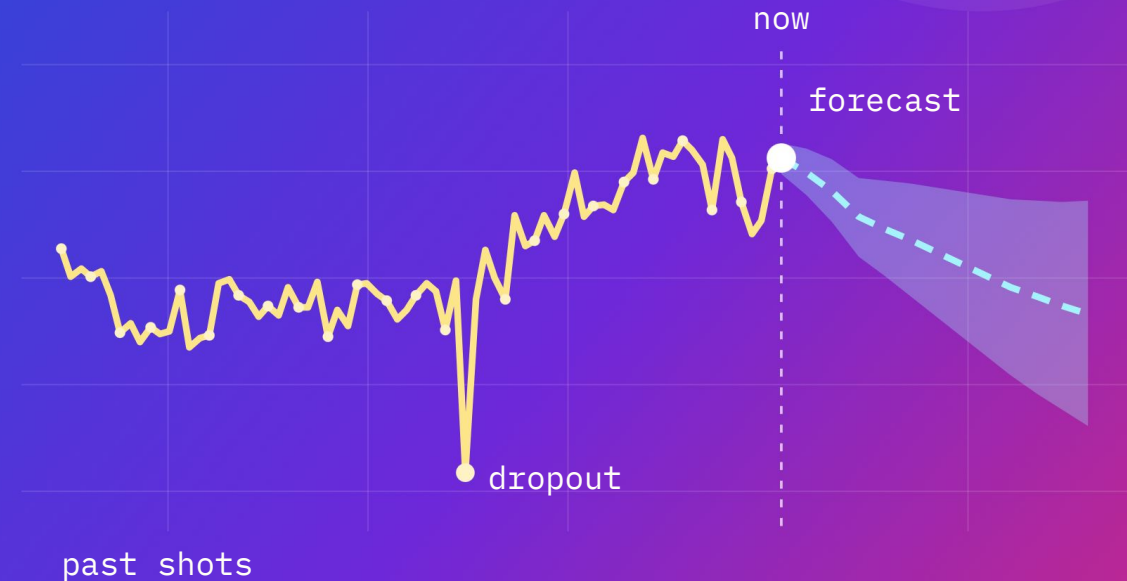
Antonio Gilardi (Team Leader & Case Holder)

CERN · CLEAR · antonio.gilardi@cern.ch

Jan Jug

Quentin Bruant

Alex Brynes



Three steps, one question

STEP 1 · EASY

Understand

raw run → clean table

Clean, synchronise, rank every signal.

STEP 2 · MEDIUM

Virtual diagnostic

$Q[t] = f(X[t])$

This shot's charge from this shot's signals.

STEP 3 · HARD

Forecast

$Q[t+1...t+H] = f(X[\leq t], Q[\leq t])$

The next shots' charge, from the past.

THEN, OPTIONAL

Close the loop with a controller $g(x)$ on laser power or RF phase. **Extremely promising results can earn real beam time on CLEAR.**

Outline

- First step:
 - Problem description
 - Data handling
 - Data cleaning
 - Main challenges
- Can we diagnose?
 - Problem description
 - Approach used
 - Model results
 - Main and open challenges
- Let's try to go further!
 - Problem description
 - Approach used
 - Model results
 - Main and open challenges
- Conclusion & Outlook
 - Where we are
 - What was done, what were the main challenges
 - Colorful solution
 - Highlights of the results
 - What we plan
 - Challenge identified
 - Future scheduling work

Let's try to understand

Process our raw material to extract the meaningful, quintessential piece of information

- In this part, we talk about cutting, stashing, synchronizing and correlating

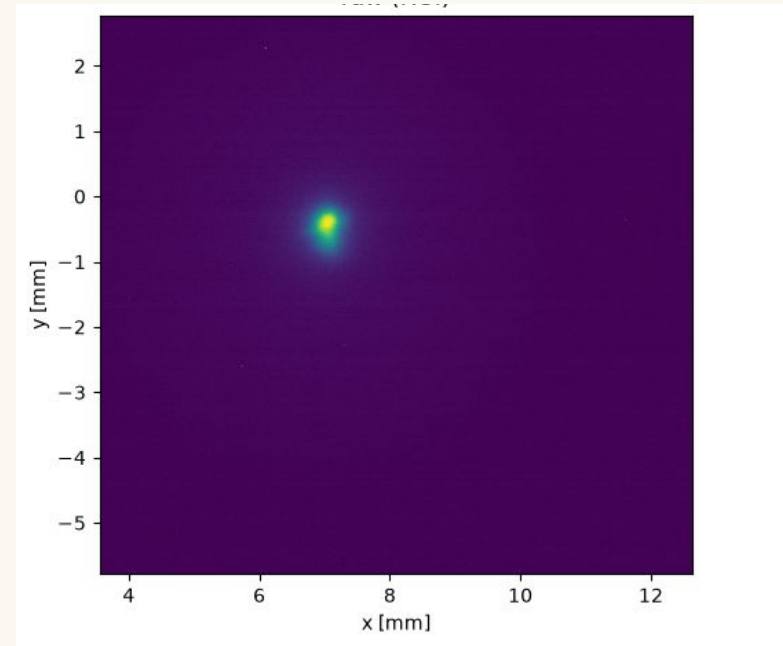
THE GOAL

1. **Clean** the raw datasets
2. **Understand** the weight of each variables in the output charge of CLEAR beam

The first step

- ~120 k shots, stored in 100 shot chunks
- Very heterogeneous datasets:
 - 1 series of image (BTV)
 - 72 series of scalars
 - 19 series of waveforms
- Unequal quality of the measurements

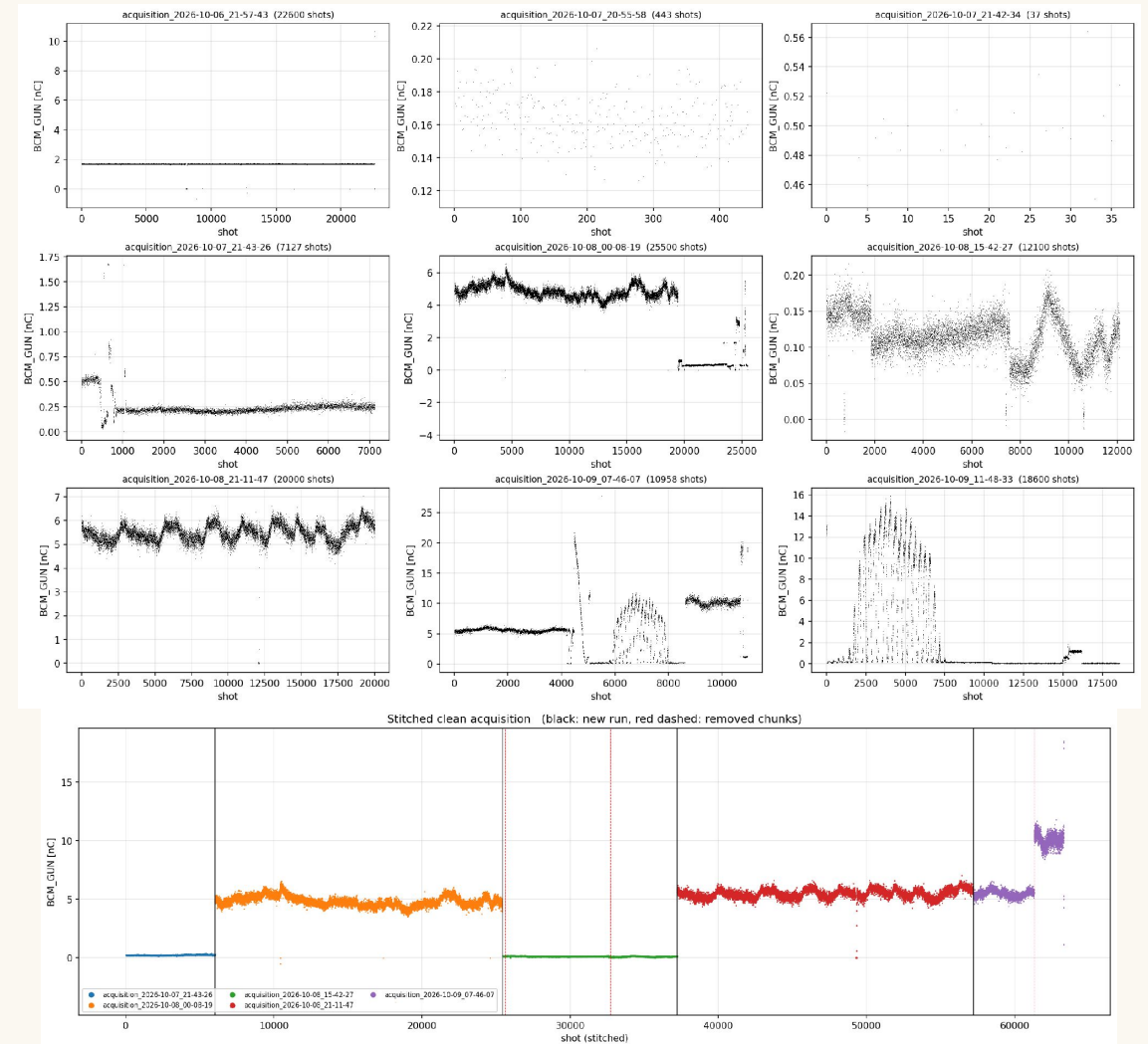
➔ **Need for a selection of representative situations to work on**



Run regimes, clean cuts and bad shots

- Every run on a different situation
- Plateaus at **0.1, 0.22, 4.8, 5.5 and 10 nC**, laser scans, and a bunch charge monitor (BCM) stuck at a 1.67 nC ceiling
- Clean cuts by inspection: 634 chunks, 63,327 shots, 8 contiguous segments in **3 regimes**:
 - Low charge (0.1 - 0.5 nC)
 - High charge (5 - 20 nC), relatively stable
 - High charge (5 - 20 nC), periodic drifts

All useful datasets stitched together



Synchronisation on the Beam pulse

Each variable has its own timestamp so we need to match them

➤ Lag scans: **lab laser vs charge**:

- at **lag = 0**: **R = 0.17**
- at **lag = +1**: **R = 0.83**
- camera peaks at lag 0

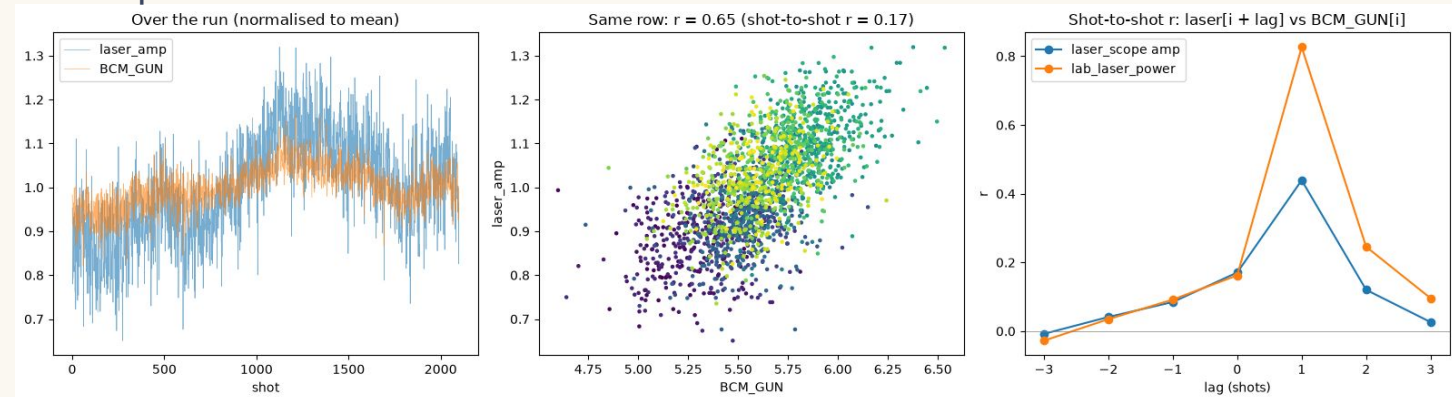
➤ Laser diode sensor repeat the previous pulse when read before 2.2 s after the trigger

➤ RF traces displays a +1 lag compared to their respective stamp; the camera stamp is one pulse late on 15 % of shots

ridge test R ² (common rows)	seg 3	seg 4	seg 5	seg 1	seg 7
row-shifted datasets	0.19	0.54	0.65	0.44	0.38
pulse-aligned datasets	0.25	0.41	0.89	0.38	0.76

Laser vs charge over a run, scatter, and the lag scan peaking at +1 row

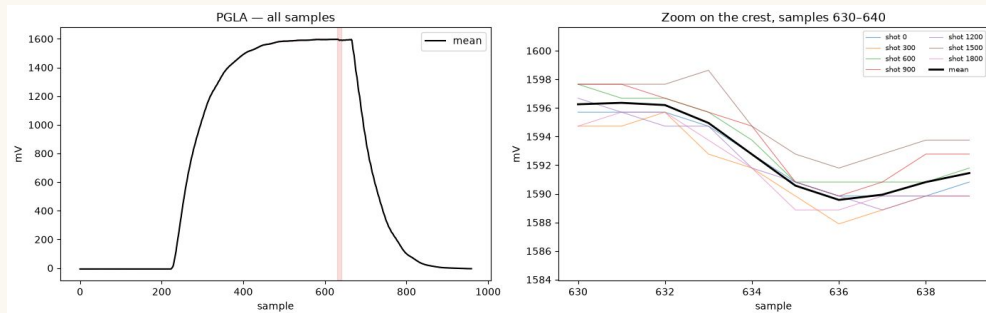
➔ Building of synchronized datasets : readings of unrecorded pulses become NaN instead of a neighbour's value



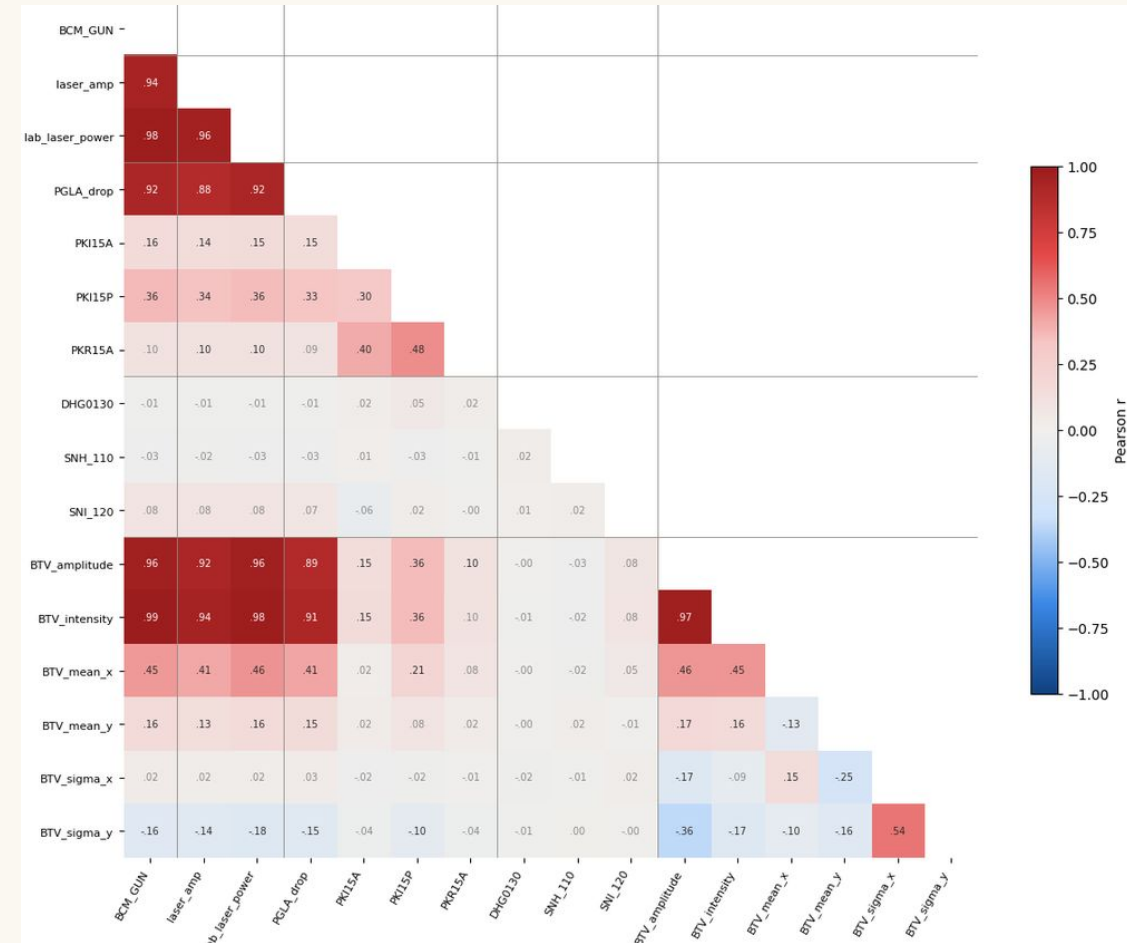
Independent variables?

Pearson Cross Correlation (R) enters the scene

- **Strongest:** lab laser power, camera intensity (**0.79**), PCA PC3, laser scope amplitude
- Klystron, magnets and climate: $|R| < 0.1$; ridge on all klystron samples has $CV R^2 \approx 0$
- RF beam-loading dip (~ 7 mV, samples 631→636) is linear in charge **0.3–10.4 nC**: a second charge monitor
- **Charge is very persistent:** "same as previous shot" $R^2 = 0.96$; shot-to-shot autocorrelation **0.67** at lag 1, **0** by lag 10



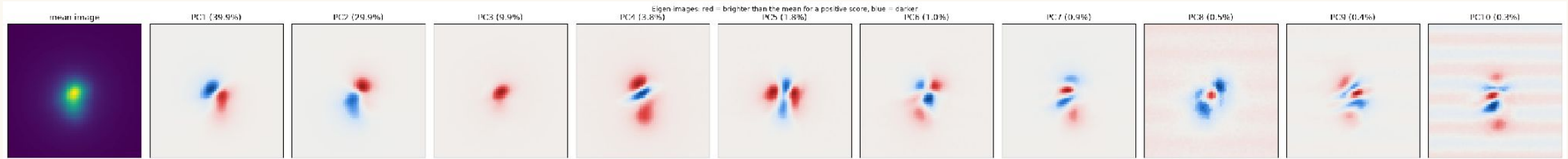
RF forward power. Left: full trace; Right: beam loading (peak-to-peak) could be used as proxy for bunch charge.



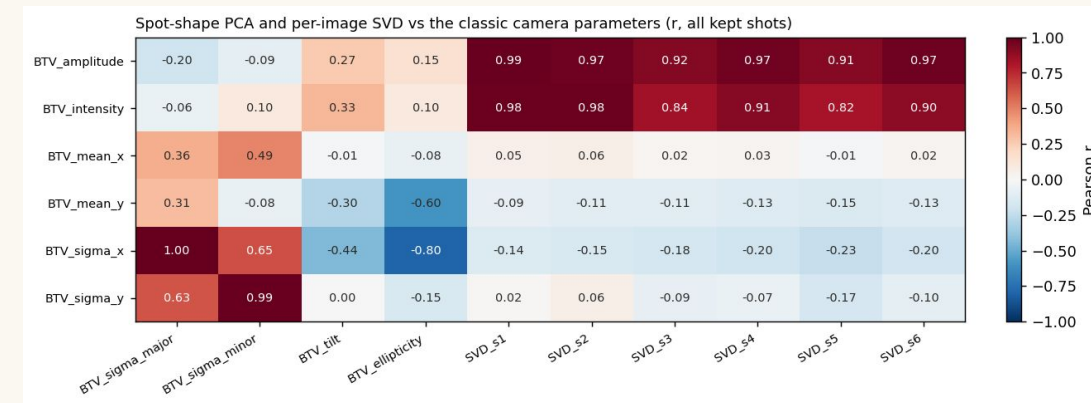
Pearson R between all variables in the dataset: BCM_GUN (leftmost column) is the target

Camera input: PCA, SVD, AE?

Beyond peak and integral: what information drive the output charge?



- Classic moments after ROI, background and median filter: intensity, centroid, sizes, tilt
- PCA over all shots: PC1 81.5 %, 6 PCs 97.5 % of variance explainability:
 - PC1-2 = centroid motion,
 - PC3-4 = brightness and size
- Per-image SVD:
 - s1 $\approx$ brightness ($r = 0.98$ with intensity)
 - no basis fitted across shots, so no leakage
- Greedy feature selection keeps 10 features:
 - lab laser, RF beam loading, SVD s1/s4-s6, camera intensity, Klystron power delivery, ...

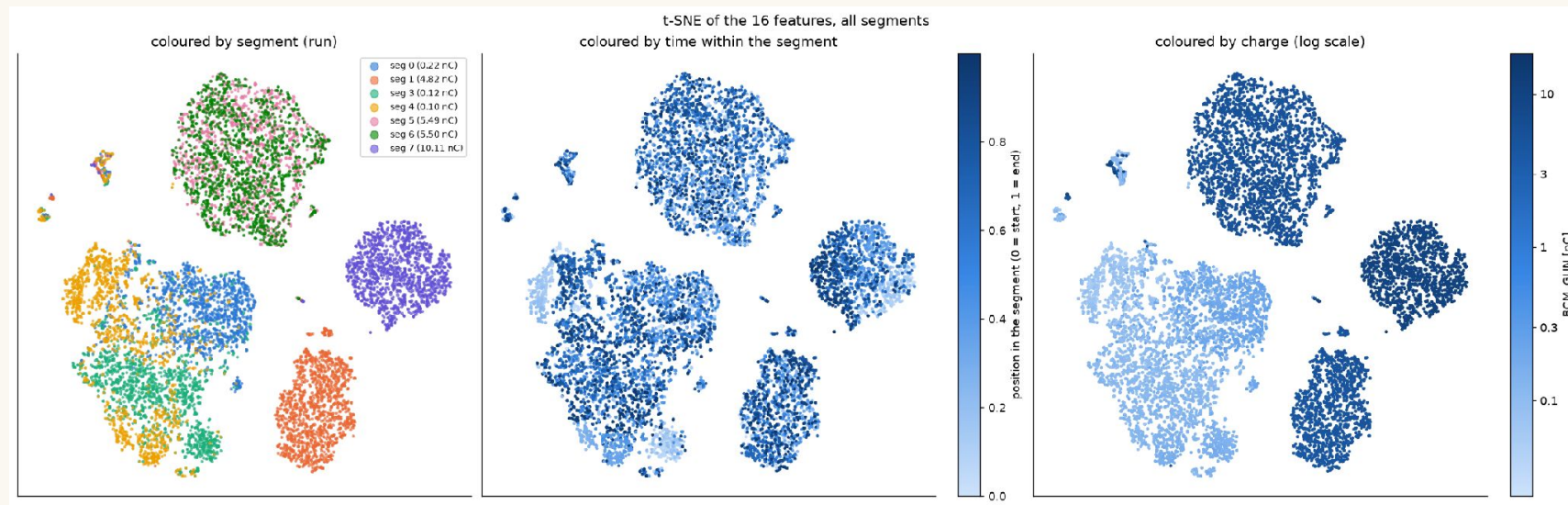


- CNN autoencoder (16-d latent) + LSTM on the latent sequence: prototype, encouraging results (see later)

Structure across runs: t-SNE of the 16 features

Shots cluster by run and charge level; the features themselves carry little memory.

10,500 shots, coloured by segment, by time and by charge.



- Segments 5 and 6 (both ~5.5 nC) overlap, segment 7 (10 nC) sits apart, the low-charge segments merge
- Predicting a feature from its previous shot: R^2 between -0.15 and 0.18,
 - ➔ The past adds little beyond the charge history

Can we diagnose?

Examine the performance of various models and try with and without BCM_GUN in the loop

➤ Models tried:

- linear regression
- random forest + xgboost
- MLP
- CNN: AE for images and temporal for history
- RNNs: LSTMs, GRUs

THE GOAL

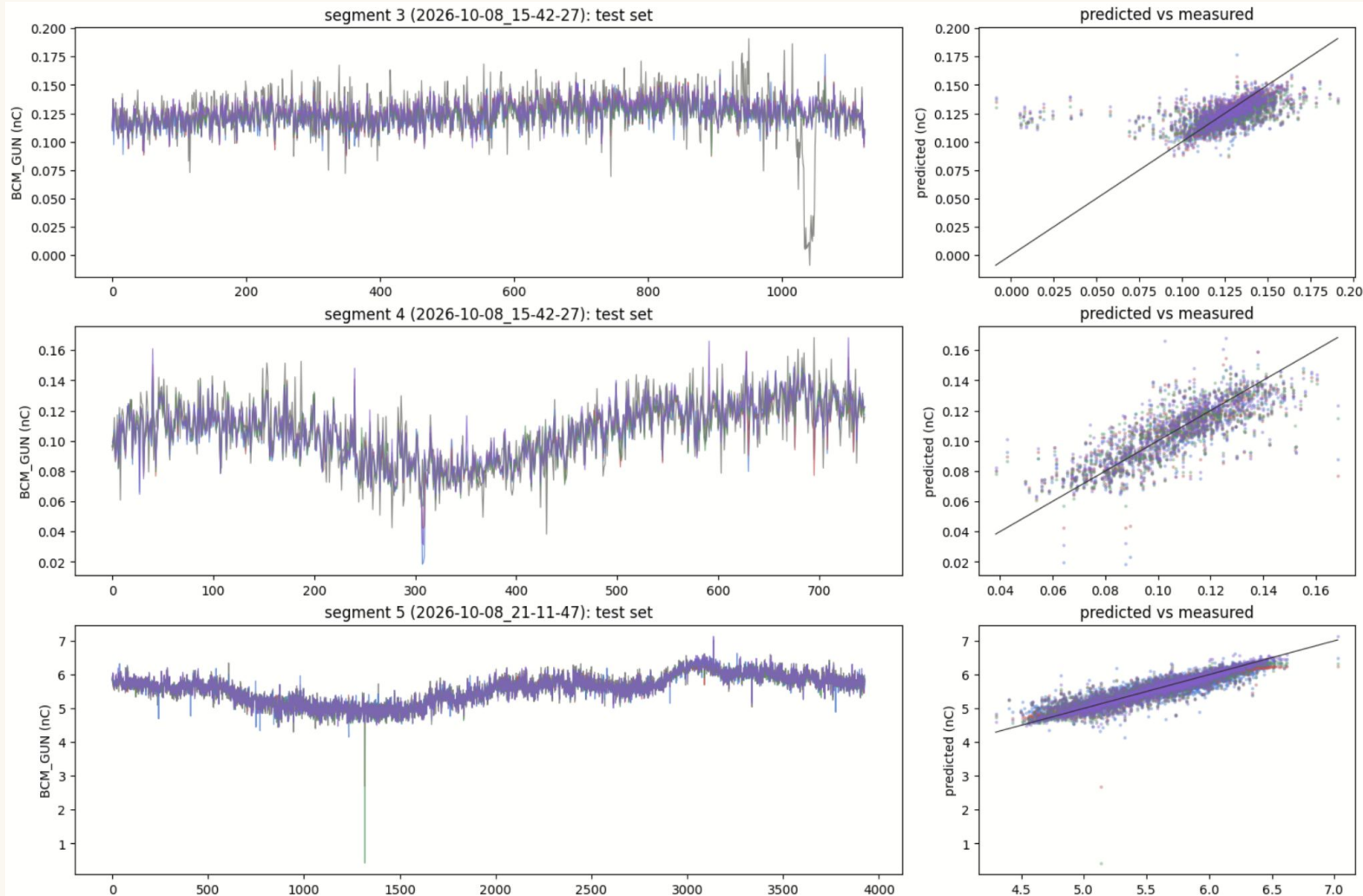
1. **Obtain good performance** in the virtual diagnostic fashion.
2. **Identify the right metric** and **try some basic model** (Linear Regression, Random Forest, XGBoost and MLP).

Vanilla models

How the different models behave on our data?

seg	regime	BCM mean pC	BCM std pC	linear	random forest	XGBoost	MLP (32-16)
0	low charge	251	23.3	13.4	12.0	16.8	11.2
3	low charge	126	20.6	11.0	10.1	10.7	9.9
4	low charge	107	21.2	9.5	9.3	9.2	9.6
5	high, repetitive	5583	427	143	96.1	90.7	85.7
1	high, non-repetitive	4862	376	166	134	185	141
6	high, non-repetitive	5571	191	47.8	63.8	57.2	50.4
7	high, non-repetitive	10106	1024	278	265	247	211

Vanilla models



- Visual inspection shows that the **models have learned to predict the BCM from the features**
- **Outliers** still **penalize** the models' **performances** a lot as seen on the correlation plot
 - The trend is well matched
- **Data might need better cleaning :**
 - Use of robust method such as **MedianAE**

Detrending

What happen when we introduce the trend followed by the BCM?

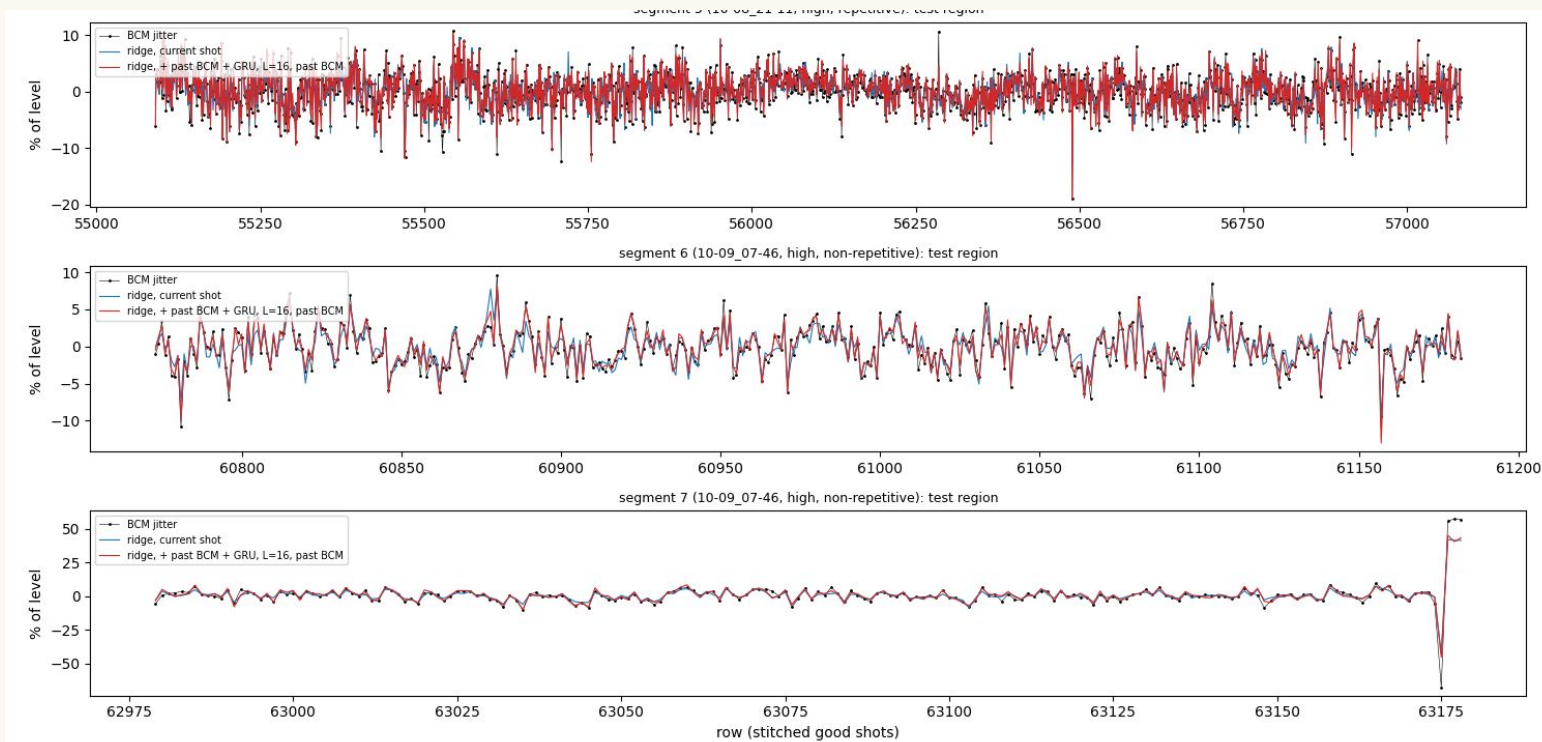
seg	regime	BCM mean pC	BCM std pC	linear	random forest	XGBoost	MLP (32-16)
0	low charge	251	23.3	10.8	9.9	10.0	9.7
3	low charge	126	20.6	10.8	10.4	10.4	10.4
4	low charge	107	21.2	9.5	8.7	8.8	8.9
5	high, repetitive	5583	427	131	90.4	87.6	86.6
1	high, non-repetitive	4862	376	130	96.5	97.3	92.7
6	high, non-repetitive	5571	191	29.1	43.0	33.6	29.1
7	high, non-repetitive	10106	1024	279	250	242	199

Sequence models

How LSTM applied on a CNN2D latentspace and a temporalCNN compare?

seg	regime	BCM mean pC	BCM std pC	temporal CNN (L = 50)	latent + features LSTM
0	low charge	251	23.3	13.1	10.2
3	low charge	126	20.6	9.6	9.7
4	low charge	107	21.2	8.9	8.7
5	high, repetitive	5583	427	84.2	81.7
1	high, non-repetitive	4862	376	119	205
6	high, non-repetitive	5571	191	63.5	155
7	high, non-repetitive	10106	1024	270	404

Slow level + shot-to-shot model



- Trailing level: baseline from 50 shot MA (zero jitter)
- **Ridge: predicts jitter from multiple instruments**
- **GRU: can capture nonlinear effects beyond Ridge**
- **Ridge+past BCM+GRU: two stages – ridge plus 16 jitter values, then 64-window GRU for the nonlinear correction**
 - (good for combining jumps with jitter)
- **+online refit: same as above, with weighted ridge based on recent errors**

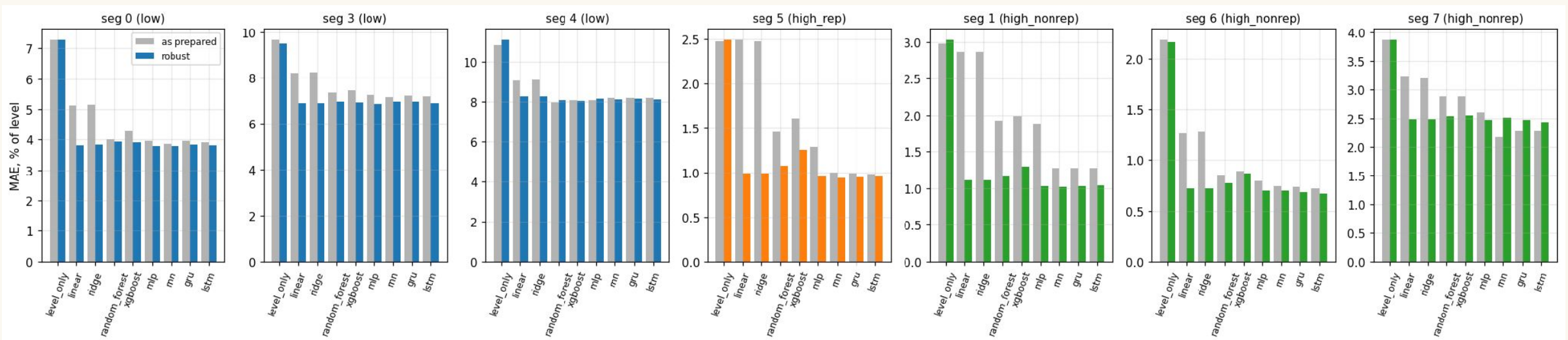
Slow level + shot-to-shot model

seg	regime	BCM mean pC	BCM std pC	Trailing level only	Ridge, current shot	GRU, window 16	Ridge + past BCM + GRU (64)	+ online refit
0	low charge	251	23.3	18.2	10.0	9.1	9.2	9.1
3	low charge	126	20.6	12.4	9.4	9.2	9.1	9.2
4	low charge	107	21.2	11.7	8.9	8.1	8.1	8.1
5	high, repetitive	5583	427	145.5	69.5	42.1	40.8	40.7
1	high, non- repetitive	4862	376	143.3	84.6	47.2	45.7	45.8
6	high, non-repetitive	5571	191	122.2	57.0	31.2	31.0	27.8
7	high, non-repetitive	10106	1024	428.1	174.4	226.3	147.0	133.2

Nine model families converge

Linear, ridge, random forest, XGBoost, MLP, RNN, GRU, LSTM, sparse GP and ensembles; MAE in % of the level.

- **Outlier-robust preparation brings all families within ~10 %:**
 - squared-loss models were chasing camera spikes (seg 1 ridge 2.86 → 1.12 %)
- **Pulse alignment helps the nonlinear models:**
 - XGBoost: **MAE = 1.26 → 0.95 %**
 - best LSTM: **MAE = 0.87 %**
 - **GRU / GP beat ridge by 11-22 % in high charge regime**



Forecasting

Repeat what has been done for virtual diagnostic in a more complex scenario

- **One step ahead, then up to 50 shots**
- **At least one sequence model: TFT, DeepAR, LSTM/GRU or a time-series foundation model**
- **Tune the context window and show its effect**

THE GOAL

1. **Identify the right metric and try some basic model**
2. **Beat** the rolling mean
3. **Deploy** in the machine the model

Next shot: the models from Tier 2

The last 5 BCMs + the current features and the L=50 temporal CNN

seg	regime	BCM mean pC	BCM std pC	mean of last 9	mean of last 50	linear	random forest	XGBoost	MLP (32-16)	temporal CNN (L = 50)
0	low charge	251	23.3	18.6	17.6	17.5	17.4	17.6	17.5	17.6
3	low charge	127	16.2	12.2	11.9	11.9	11.9	11.9	12.1	11.9
4	low charge	107	21.2	12.1	11.9	11.9	12.0	11.9	12.0	11.9
5	high, repetitive	5583	427.1	146.5	147.4	141.7	141.7	142.0	141.7	142.4
1	high, non-repetitive	4862	375.9	141.8	142.5	136.2	136.0	136.6	137.6	136.7
6	high, non-repetitive	5571	190.8	118.0	114.4	111.2	111.4	113.7	113.9	112.2
7	high, non-repetitive	10086	680.3	340.5	337.7	335.0	340.9	338.3	344.5	333.4

- **Mean of last 50 performs approximately the same as any model**, meaning that there is little extra information available
- **Temporal CNN basically performs as well as the rolling average**
- Note: no hyperparameter search was performed - these are vanilla models with default hyperparameter values

Next shot: beating the rolling mean of 9

Charge of shot $n+1$ from everything up to shot n ; last 10 % of each run.

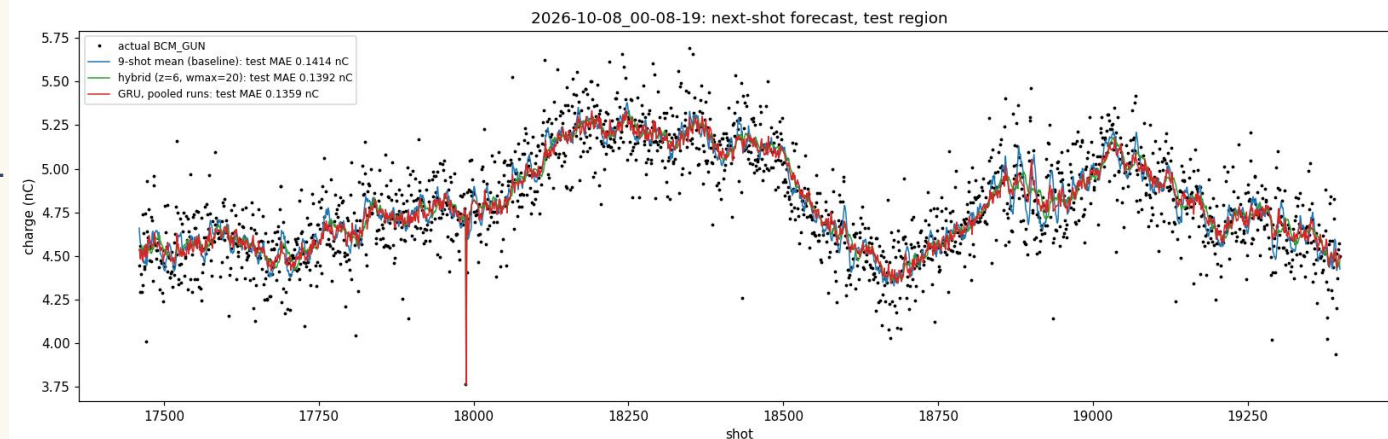
MAE vs 9-shot mean	Explanation	run 1 (10 nC)	run 2 (5.5)	run 3 (4.8)	run 4 (0.2)	run 5 (0.1)
9-shot mean (nC)	Used by operators – the target to beat	0.343	0.144	0.141	0.0195	0.0122
hybrid level	Same as above, but with a reset after a level change (deviation by 6x jitter, dropout, etc.)	−8.0 %	−1.9 %	−1.6 %	−1.7 %	−2.5 %
GRU, pooled runs	GRU trained on first 90% of all 5 runs, tested on last 10%	−9.2 %	−3.0 %	−3.9 %	−4.6 %	−3.3 %
GRU, other runs only	GRU trained on four runs and tested on the remaining one	−8.8 %	−2.3 %	−3.1 %	−4.8 %	−3.6 %

Block bootstrap*

Fake additional data by drawing from error of GRU and 9-shot mean

Since consecutive errors are related, we randomize them in blocks of 50 shots

- **GRU significantly better** on all 5 runs (block bootstrap*)
 - persistence is 25 % worse
- Run 1's gain is mostly its final level jump
 - (3.6 % on its stable shots)
- Context window 8–256 shots: $\Delta \leq 1.3$ pC
 - laser and other inputs add nothing



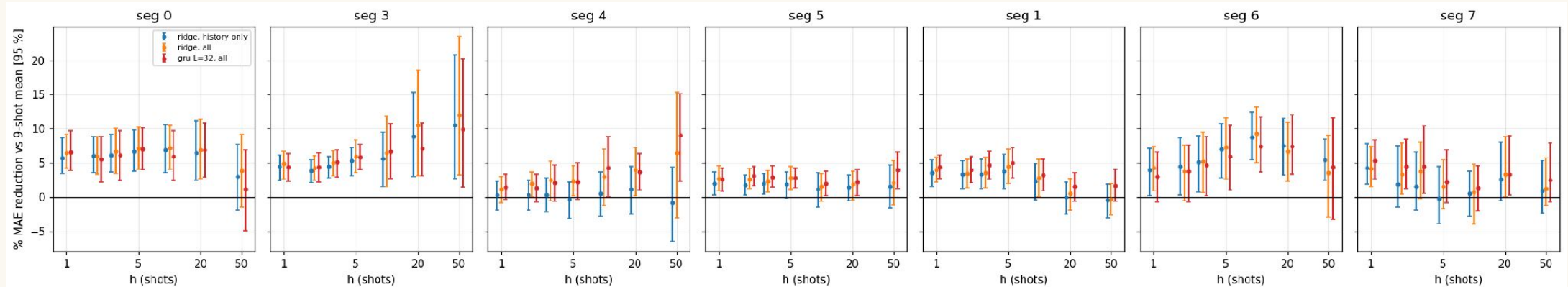
Ridge, past charge only (blue)

Ridge, past charge plus other diagnostics (orange)

GRU, past charge plus other diagnostics (red)

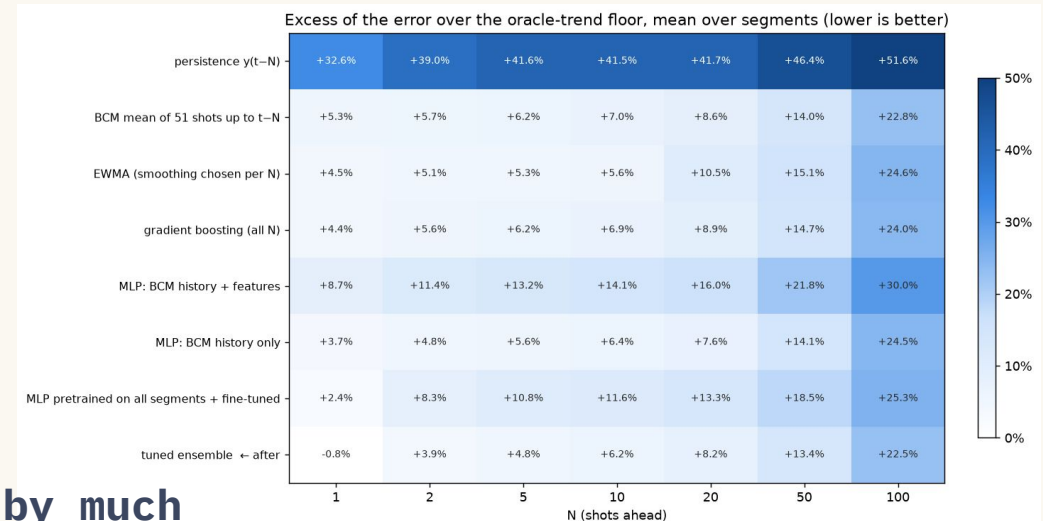
Further ahead, smaller gains

Multi-step forecasts vs the 9-shot mean & the oracle floor (slow trend + random noise from both past and future).



- **$h = 1-5$: 1.5–7 % better than the 9-shot mean,** significant on 5–7 of 7 segments; $h = 50$ mostly not
- **Ensemble at the floor up to $N \approx 2$, +6 % at $N = 10$, +22 % at $N = 100$**
- Instruments add $\sim 0.5-1$ point: **the next laser pulse is not predictable ($R^2 \leq 0.35$)**
- Simple and exponential moving average give **$\sim 5\%$ excess error.**

→ The heavier models help, but not by much



Summary: what worked and what did not

Worked

- Pulse-grid alignment of every reading
- Slow + fast split: trailing level + GRU, at the noise floor on 3 segments
- Beating the 9-shot mean on every run (-2 to -9 %)
- Measured BCM noise for honest intervals

Did not help

- Kalman / EMA / median levels, periodic terms
- Forecasting the instruments (two-step models)
- Attention, autoencoders, minimal hyperparameter optimization
- RF, climate and magnet inputs

Further work

Step 1:

- **Review the work** - a lot was done, **all of it in a Fast & Approximate manner**
- New information as we proceeded was not necessarily propagated properly back into the data preparation stage

Step 2:

- Better data alignment
- Redo the feature selection, perhaps with a full grid search

Step 3:

- Understand the model limitations and why they performed little better than the rolling means
- Select the most promising models and do a proper hyperparameter search

Step 4:

- ???

Step 5:

- Publication



THANK YOU

Let's predict the next shot

Antonio Gilardi

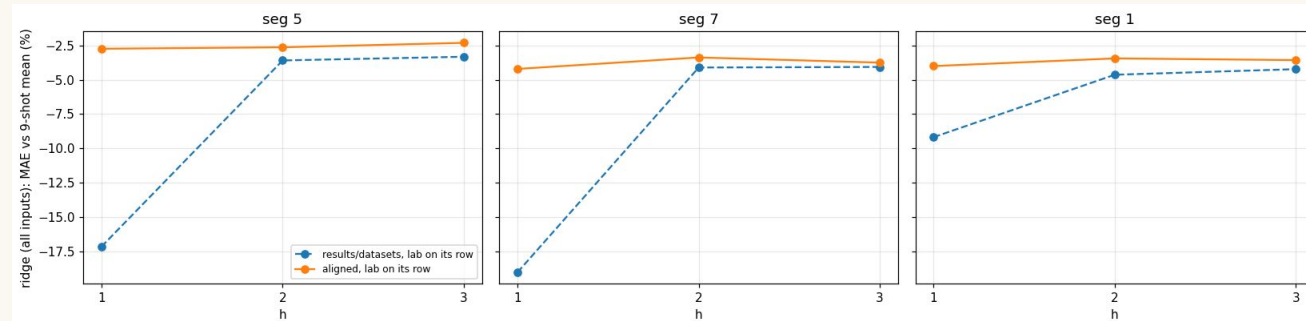
`antonio.gilardi@cern.ch`

Pitfall: a fake gain from misaligned data

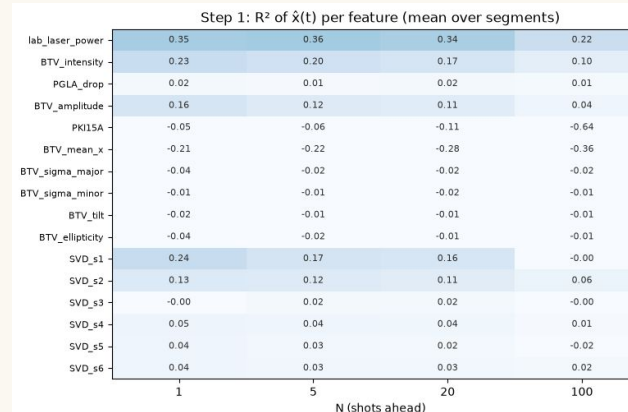
With the laser meter on the wrong row, the "forecast" sees the future.

- Row-shifted datasets: $h = 1$ gain of **-17 to -19 % (!!)** vs the 9-shot mean.
- **BUT:** the laser meter is often read before it has updated, so it still shows the previous pulse's value – a next-shot forecast can cheat.
- **Pulse-aligned:** -3 to -4 %, flat over horizon
- Two-step model (reusing the virtual diagnostic to feed into the forecast) is **no better** than the **51-shot mean**: present features cannot be forecast reliably.
- **Lesson:** check every variable's lag on the pulse grid before trusting a gain

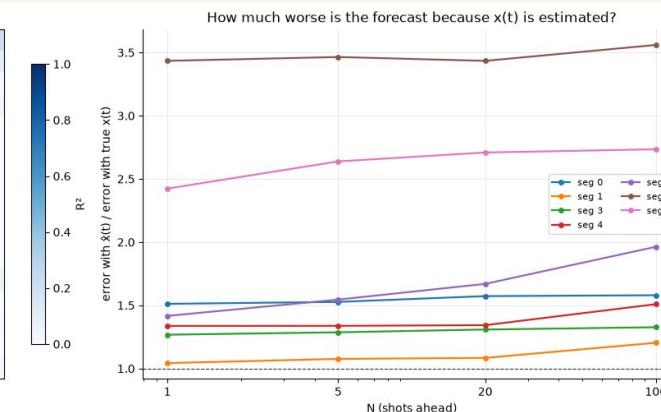
Ridge model using all inputs, predicting h shots ahead



R² at shot h for each feature



Forecasting for different segments

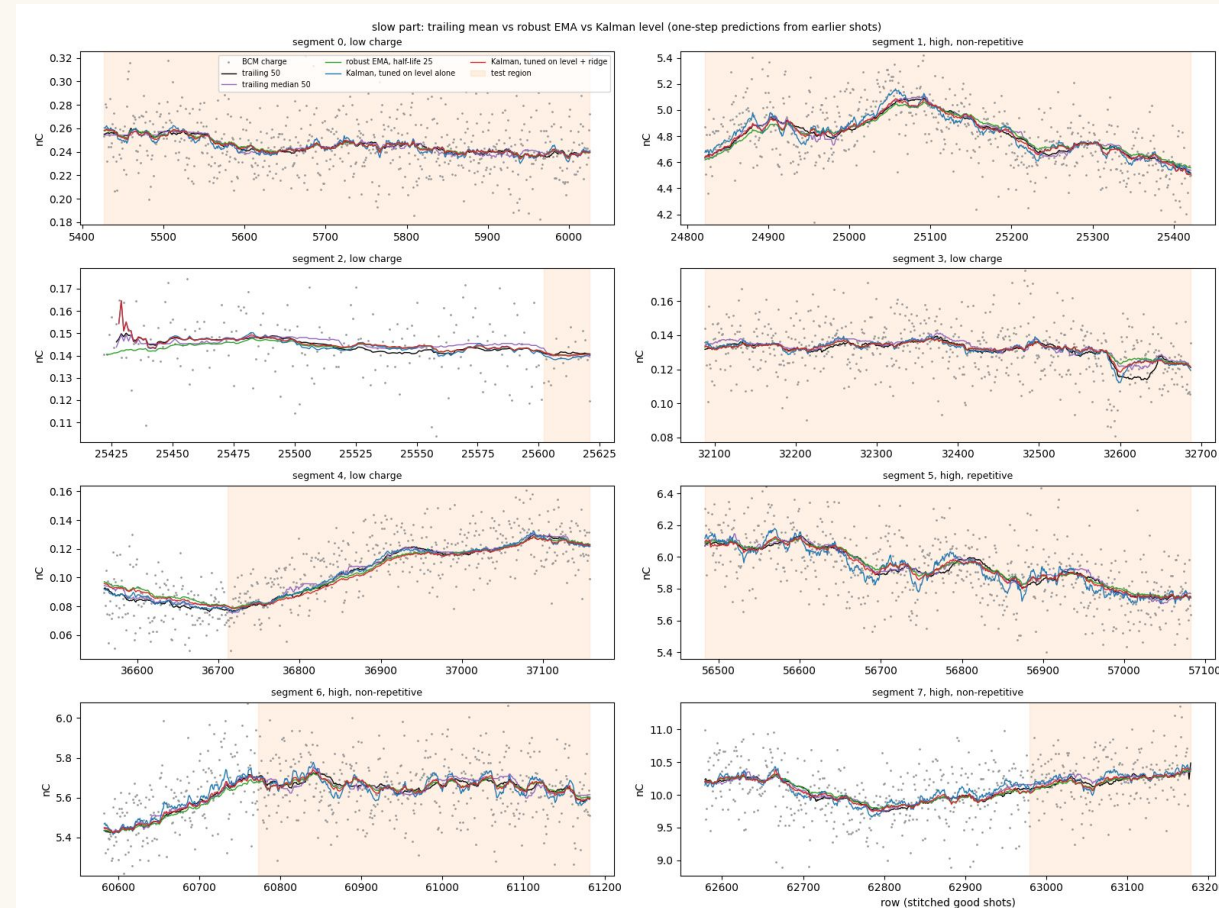


Filtering the slow level: simple wins

Trailing mean vs median vs robust EMA vs Kalman level filter, with and without a ridge on top.

MAE pC, + ridge	seg 3	seg 5	seg 1	seg 6
trailing mean 50	9.4	69.5	84.6	57.0
trailing median 50	9.2	71.2	84.6	59.0
robust EMA (half-life 25)	9.2	72.5	88.0	56.9
Kalman level	9.2	72.1	85.6	57.3

- Kalman tuned on the level alone is the best level (seg 5: 140.5 vs 145.5 pC) but **absorbs predictable jitter**
- Spectrum: no lines, residual white; only seg 3 has a slow drift the trailing mean misses
- **Online refit** after every shot: **-22 to -43 %** on the ridge, seg 6 31 → 28 pC on the best model



Noise floor and uncertainty

The BCM's own noise sets the floor;
prediction intervals checked on the test region.

- **BCM noise:** white, Gaussian core with heavy tails (0.2–1.5 % beyond 4σ); ~50 % of the jitter at low charge
- Rolling 90 % interval covers 87–92 % (± 5 –7 % at 5–10 nC)
- **GP** with the measured noise as its kernel noise: **no better MAE**, right intervals once missing camera frames get extra variance (86–95 %)
- **Minimax** (max-error) training **does not reduce the worst error**

