hep-th Prediction Bot

Luke Corcoran

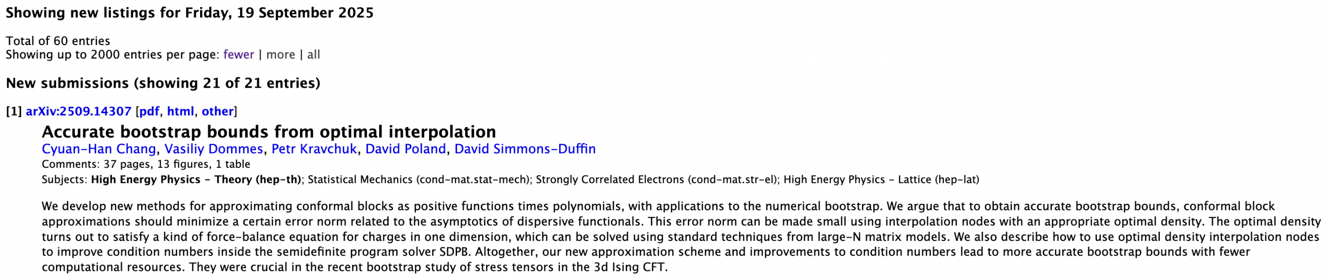
September 21, 2025

Abstract

hep-th Predictor is a Twitter bot which predicts the number of papers appearing on arXiv's high energy physics theory section every day. It is based on a LightGBM gradient-boosted tree model and is trained on 20 years of historical data. In this report, we give a short description of the background, features, and performance of the model.

1 Introduction

Every good theoretical physicist wakes up and opens https://arxiv.org/list/hep-th/new (figure 1) to see the latest papers in the field, look for new ideas, and check whether they have been scooped.



Showing new listings for Friday, 19 September 2025

Total of 60 entries
Showing up to 2000 entries per page: fewer | more | all

New submissions (showing 21 of 21 entries)

[1] arXiv:2509.14307 [pdf, html, other]

Accurate bootstrap bounds from optimal interpolation
Cyuan-Han Chang, Vasiliy Dommes, Petr Kravchuk, David Poland, David Simmons-Duffin
Comments: 37 pages, 13 figures, 1 table
Subjects: **High Energy Physics - Theory (hep-th)**; Statistical Mechanics (cond-mat.stat-mech); Strongly Correlated Electrons (cond-mat.str-el); High Energy Physics - Lattice (hep-lat)

We develop new methods for approximating conformal blocks as positive functions times polynomials, with applications to the numerical bootstrap. We argue that to obtain accurate bootstrap bounds, conformal block approximations should minimize a certain error norm related to the asymptotics of dispersive functionals. This error norm can be made small using interpolation nodes with an appropriate optimal density. The optimal density turns out to satisfy a kind of force-balance equation for charges in one dimension, which can be solved using standard techniques from large-N matrix models. We also describe how to use optimal density interpolation nodes to improve condition numbers inside the semidefinite program solver SDPB. Altogether, our new approximation scheme and improvements to condition numbers lead to more accurate bootstrap bounds with fewer computational resources. They were crucial in the recent bootstrap study of stress tensors in the 3d Ising CFT.

Figure 1: arXiv hep-th "new" page

The new page lists the current date and the number of new papers, followed by a list of these papers and their abstracts. It would be understandable to imagine that the number of papers appearing each day is fairly random, since there is apparently no reason why a physicist would rather submit their paper on one day over another. However, this is far from the case. Theoretical physicists are human after all[1], and various societal and cultural factors lead to interesting patterns in the paper counts.

For example, let's consider a histogram of the total number of papers appearing on each weekday since 2006 (figure 2). The first thing to notice is that there are no papers appearing on Saturday or Sunday: arXiv doesn't post on weekends. All papers submitted after the cutoff time (18:00 GMT) on Friday and before the cutoff time on Monday appear on the Tuesday list of papers. This is one reason for the large spike of papers announced on Tuesday. Another reason is that physicists typically want maximum visibility for their papers. Because of this they tend to avoid their papers appearing on Monday, since it is generally assumed that more physicists will be at work and checking arXiv from Tuesday - Thursday. This means that Wednesday is the best time for a paper to appear from a visibility point of view. The spike of papers on Friday with respect to Thursday is likely due to a rush to submit papers before the weekend.

[1] This fact is still disputed by some people.

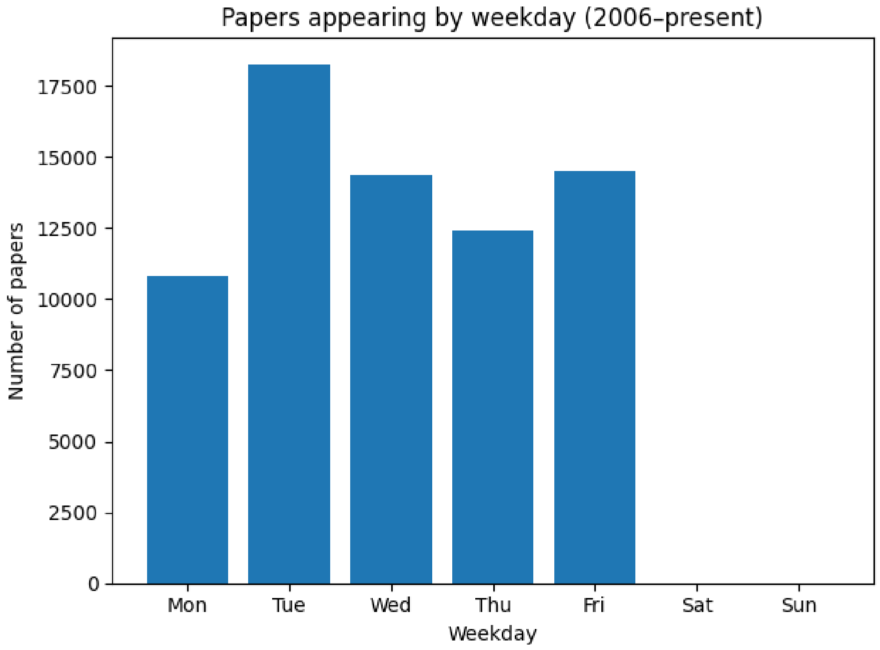


Figure 2: Number of papers appearing on each weekday since 2006

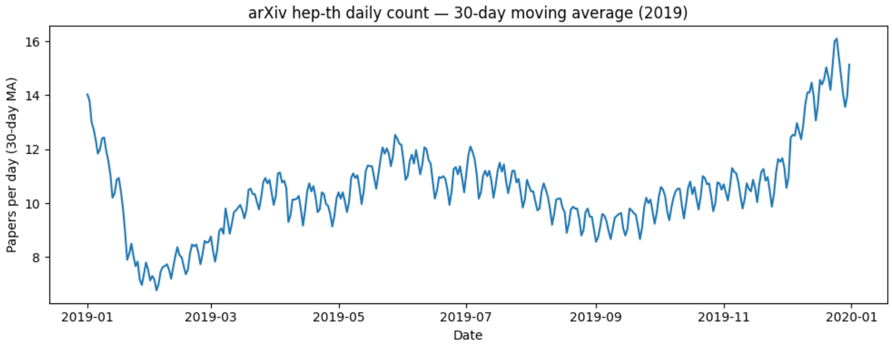


Figure 3: Number of papers appearing in 2019 (30 day rolling average)

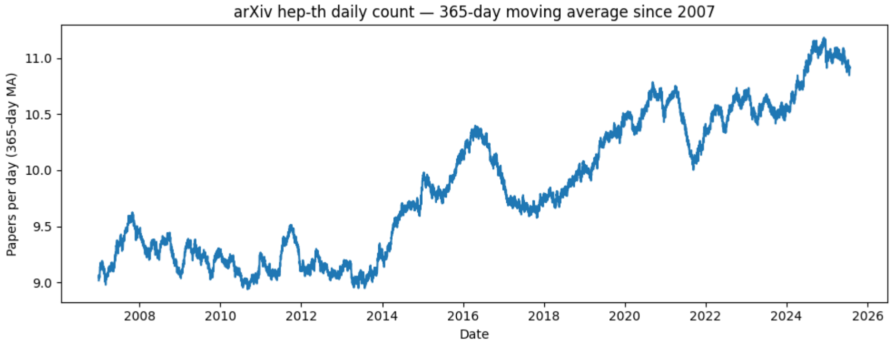


Figure 4: Number of papers appearing since 2006 (365 day rolling average)

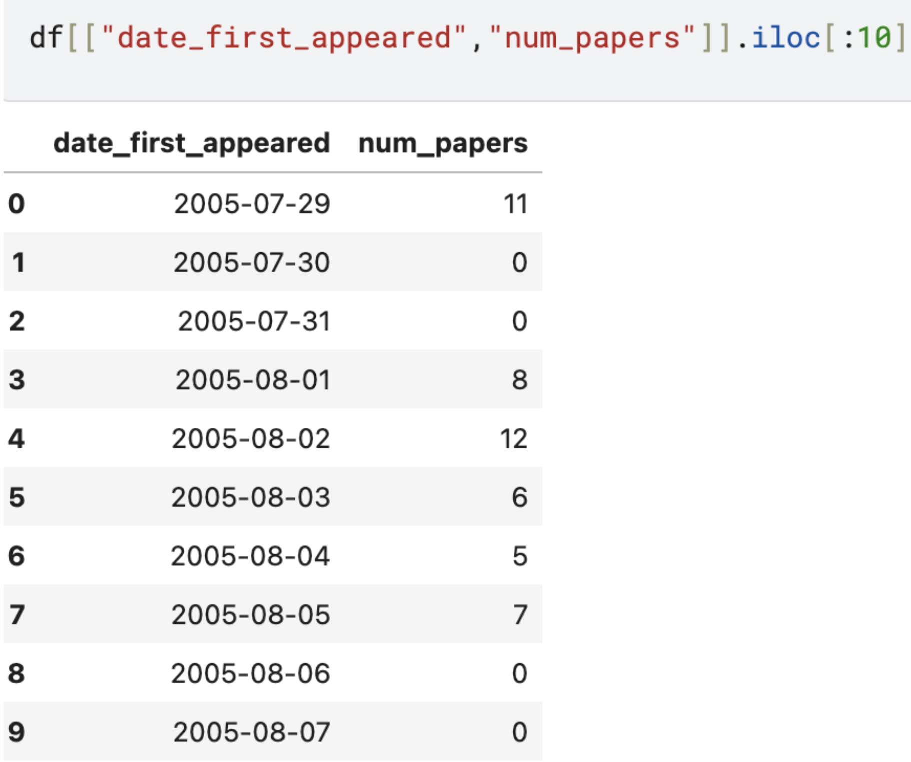
There are also strong seasonal features in the paper counts on a longer-term basis. In figure 3 we show a smoothed plot of the daily paper count in 2019. In this plot we see that the number of papers conforms strongly to the academic and holiday calendar. For instance, we see a very strong peak in the number of papers appearing in December. This is due to physicists rushing to get their papers out before the Christmas holidays. This peak is followed by a large dip in late December/early January before recovering as physicists come back to work. There is a smaller but noticeable peak around June which coincides with the end of the academic year, after which many physicists go on holidays again or have busy conference schedules.

As a final example, there is an overall trend of increase in papers as the years go by (figure 4). We see that despite some local variation (for example due to the COVID-19 pandemic), there is a clear increasing trend in the number of daily papers. This is potentially due to increased competition for grants, leading to a higher pressure to publish.

These examples show that the number of papers appearing on hep-th every day is not purely random, and is informed by a number of seasonal and trend features. There are several more relevant features; for example publications will surge slightly near an important conference. There is also a clear autocorrelation in the dataset which makes short-term lag features valuable. As such, we trained a machine learning model to forecast the number of papers appearing every day, based on a subset of relevant features.

2 Dataset, Features, and Model

Dataset. The original dataset consists of columns `date_first_appeared` and `num_papers`, and consists of data from **25-09-2006** to **24-07-2025**. We constructed this database using approximately 100,000 queries to the arXiv API. In figure 5 we display the first 10 entries.



```
df[["date_first_appeared","num_papers"]].iloc[:10]
```

	date_first_appeared	num_papers
0	2005-07-29	11
1	2005-07-30	0
2	2005-07-31	0
3	2005-08-01	8
4	2005-08-02	12
5	2005-08-03	6
6	2005-08-04	5
7	2005-08-05	7
8	2005-08-06	0
9	2005-08-07	0

Figure 5: dataframe - first 10 entries

Features. We wish to build a feature set to act as predictors for the target variable num_papers. Our feature vector combines calendar indicators and time–series summaries designed for short-horizon forecasting. Using inspiration from the discussion in the previous paragraph, and by testing various models on various feature sets, we settle on the following 14 features:

days_until_public_holiday and days_since_last_public_holiday encode proximity to public holidays.

day (day of month) and weekday (0=Mon,. . . ,6=Sun) capture regular calendar effects.

days_since_start provides a smooth long-run trend index.

mean_same_weekday_4w is the mean count for the same weekday over the previous four weeks (e.g., the last four Tuesdays), stabilizing weekday-specific variation.

days_since_last_conference and days_until_next_conference introduce exogenous signals from major-field conferences (we consider 3 conference series: Strings, Amplitudes, and IGST).

Rolling statistics roll_7, roll_30, roll_90, and roll_365 are trailing-window means that summarize short-, medium-, and long-run levels.

Finally, lag_1 and lag_7 are the previous day's and previous week's counts, respectively, capturing autocorrelation at daily and weekly frequencies.

Model. We use a gradient-boosted tree model because it can capture rich non-linear interactions between the target and features without heavy feature engineering. We choose LightGBM because it trains quickly on tabular data, handles missing values natively, and its leaf-wise tree growth tends to deliver state-of-the-art accuracy for small–to–medium datasets while remaining easy to interpret via feature importances/SHAP.

Tuning and validation. We selected hyperparameters via a grid search using mean absolute error (MAE) on a held-out validation set. The final LightGBM configuration was learning_rate=0.03, num_leaves=31, feature_fraction=0.9, bagging_fraction=0.8, bagging_freq=5, n_estimators=2000, random_state=42. Models were trained on data from **2006-09-25** to **2023-01-01** and validated on **2023-01-02** to **2025-07-24**. The best setting achieved a validation MAE of **2.59**.

Feature importances. Model-based importances from LightGBM (figure 6) highlight days_since_last_public_holiday as the strongest predictor, with days_until_public_holiday also near the top—consistent with surges in submissions immediately after holidays. Short-horizon level effects follow: the trailing means roll_7, roll_30, and roll_90 carry substantial weight, indicating that recent momentum matters more than any single lag. Systematic calendar structure is also important: weekday ranks highly (capturing the Tue–Thu bulge) and day (day of month) contributes nontrivially. A slow trend enters via days_since_start, while very long memory (roll_365) is less influential than short/medium windows. Pure lags lag_1 and lag_7 help but are not dominant, suggesting the rolling means summarize them more robustly. Conference timing features (days_since_last_conference, days_until_next_conference) and the same-weekday mean (mean_same_weekday_4w) add signal but sit in the lower tier relative to holiday proximity and recent averages.

Model accuracy. As mentioned, the chosen model leads to a validation set MAE of 2.59. To get a better sense of the accuracy of this model we compare it against more naive predictors. The first model we compare against is simply lag_7, i.e. we simply predict the number of papers that appeared on the same weekday one week previous. We also compare against mean_same_weekday_4w, we predicts the average of the last 4 same weekdays. For each of these models we plot the cumulative accuracy $|n_{\rm pred} - n_{\rm actual}|$ for non-weekend days. We see that the LightGBM model is meaningfully ahead of the other two models on the validation set, guessing correctly within 5 papers roughly 80% of the time.

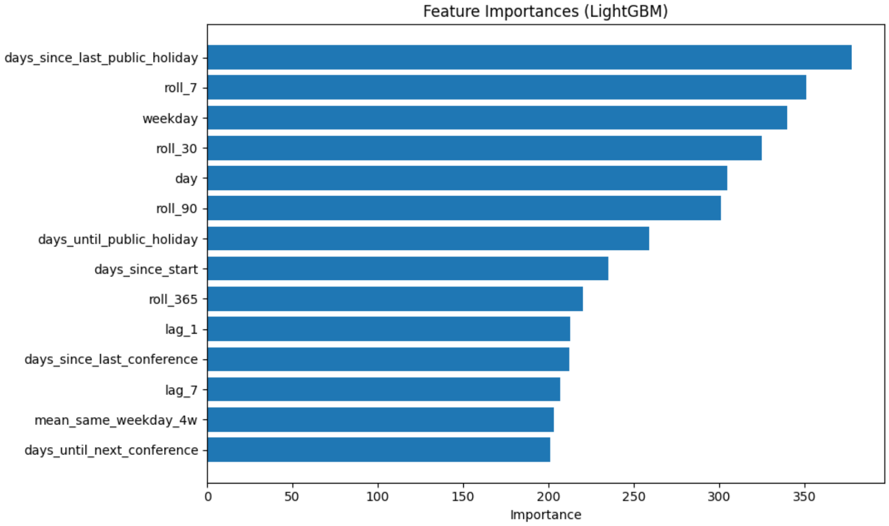


Figure 6: Feature importances

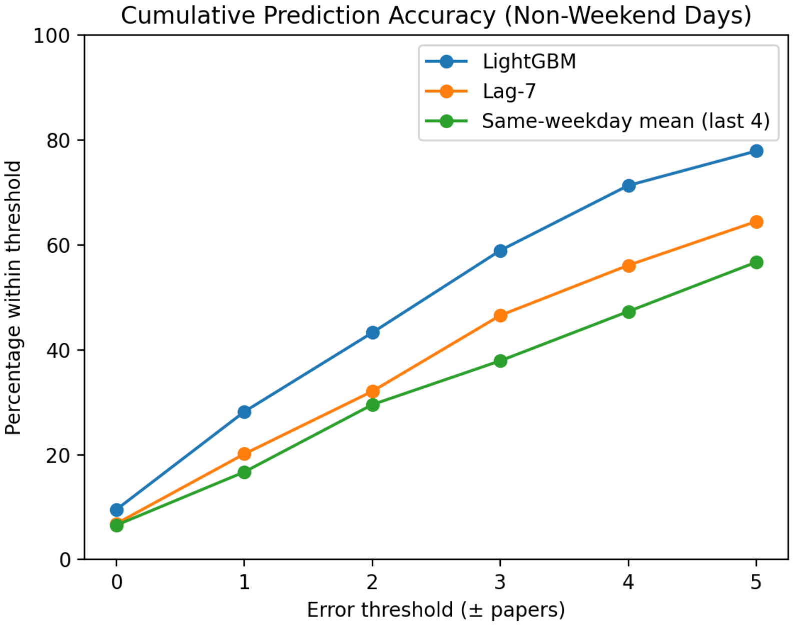


Figure 7: Comparison of LightGBM versus naive models for daily paper counts.