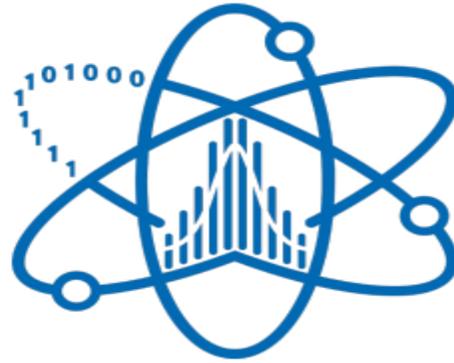




National Research  
**Tomsk  
State  
University**



**Лаборатория  
анализа данных  
физики высоких энергий**

Томского  
государственного  
университета

**Measurement of differential cross-sections of a single top quark  
produced in association with a  $W$  boson with ATLAS at  
 $\sqrt{s} = 13$  TeV**

# Progress Report

**Neda Firoz**

# Goal: separate $tW$ (top+anti-top) from $t\bar{t}$ in the 1j1b dilepton region

## •Inputs (9 variables used):

bdt\_centrality\_1l\_recalc\_NOSYS,  
bdt\_delta\_pT\_1l\_MET\_recalc\_NOSYS,  
S,  
bdt\_delta\_pT\_1lb\_MET\_recalc\_NOSYS,  
YS,  
bdt\_eta\_1lMetB\_recalc\_NOSYS,  
bdt\_m\_11b\_recalc\_NOSYS,  
bdt\_m\_12b\_recalc\_NOSYS,  
bdt\_pT\_1lMetB\_recalc\_NOSYS,  
bdt\_pT\_1lb\_recalc\_NOSYS,  
bdt\_sum\_ET\_recalc\_NOSYS.

•**Samples / tree:** all files' tree name is analysis

**Signal:**  $tW$  (top) +  $t\bar{W}$  (anti-top)

**Background:**  $t\bar{t}$  (non-all-had)

•**Event weights:** auto-resolved to  
 $\text{weight\_mc\_NOSYS} * \text{weight\_pileup\_NOSYS}$   
 $* \text{globalTriggerEffSF\_NOSYS}$ .

•**Bad values:** any variable  $\leq -990$  or non-finite is masked per event.

# Reference Article's Report on BDT

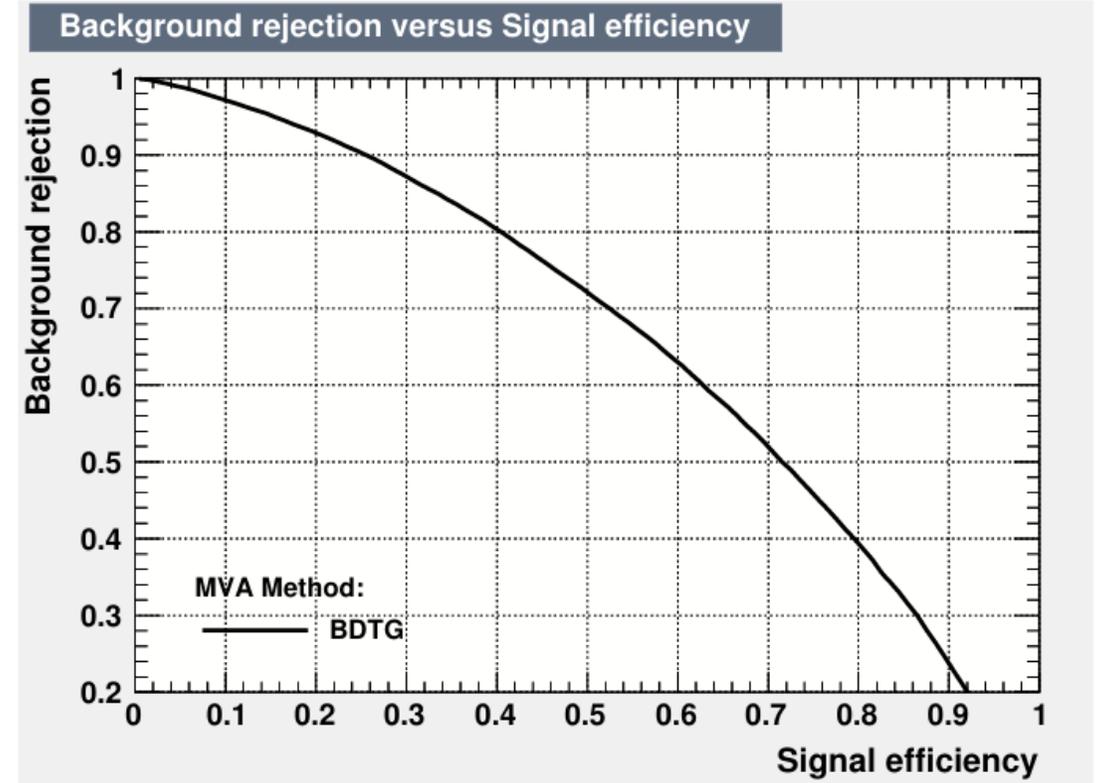
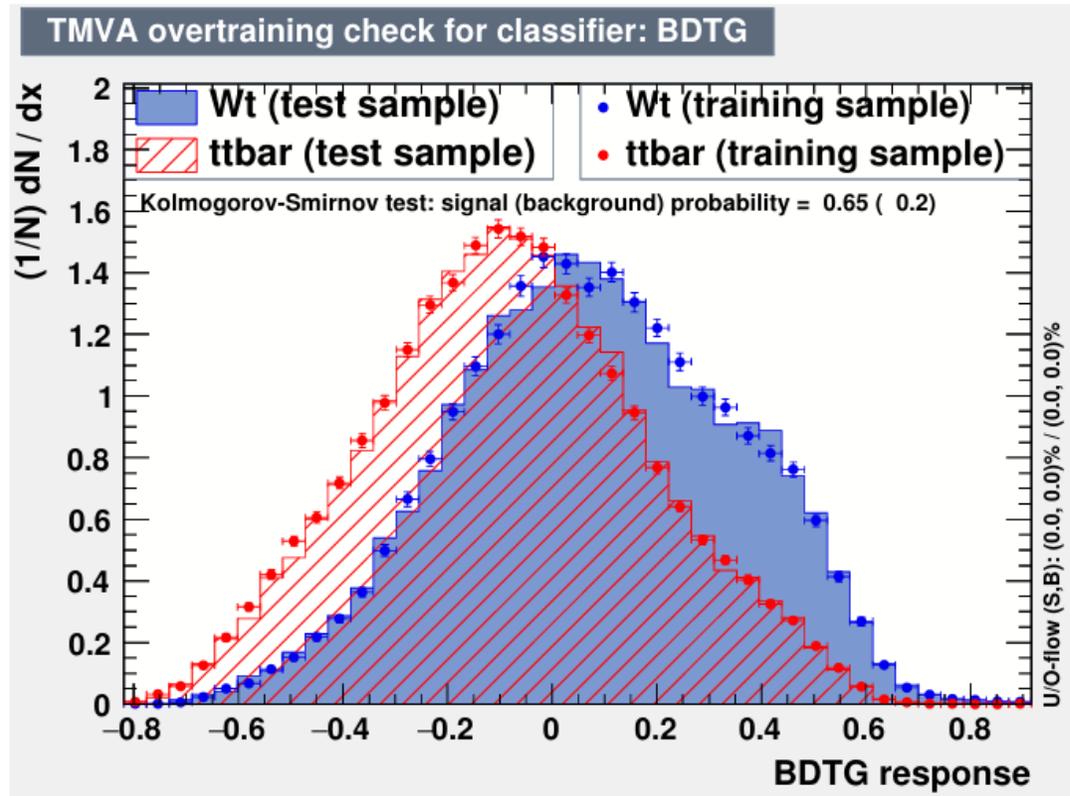


Figure 9: Comparison of test/training sample distributions and background rejection factor versus signal efficiency.

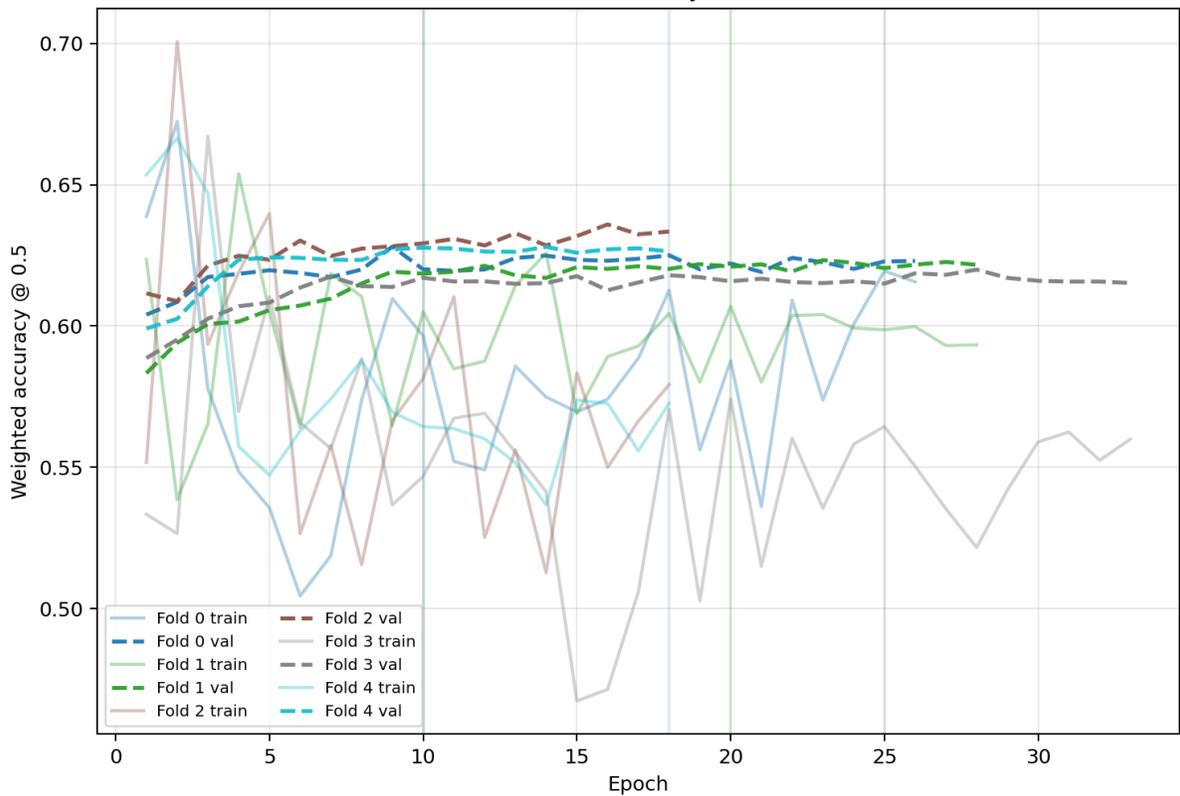
# LSTM

1. This script trains a binary classifier to distinguish:
  - signal events, labeled 1
  - background events, labeled 0
2. 9 variables were used as input features
3. The LSTM is being used as a recurrent classifier over an ordered sequence of tabular physics features. The features are not temporal measurements, but we impose a fixed order and let the recurrent architecture learn interactions across the feature sequence.
4. The total background weight is rescaled so that:  $\sum w_{background} \approx \sum w_{signal}$  (Class balancing step).
5. Nested cross-validation: an inner loop for hyperparameter optimization and an outer loop for unbiased performance estimation.
6. Each feature is standardized using training-set statistics only:  $x' = \frac{x - \mu_{train}}{\sigma_{train}}$ . Feature scaling is fitted on the training split only and then applied to validation data, preventing leakage.
7. The selected recurrent model, LSTM, reads the ordered feature vector as a short sequence. The code does not assume a fixed optimal epoch. Instead, it trains until validation AUC stops improving by at least min\_delta for patience consecutive epochs after a minimum training period. The model state from the best validation epoch is restored. Training uses weighted binary cross-entropy, AdamW, learning-rate reduction on plateau, and early stopping based on validation AUC. After training, the best model state is restored and weighted AUC was saved, weighted accuracy, ROC curves, score distributions, and signal-to-background ratio plots for each fold.

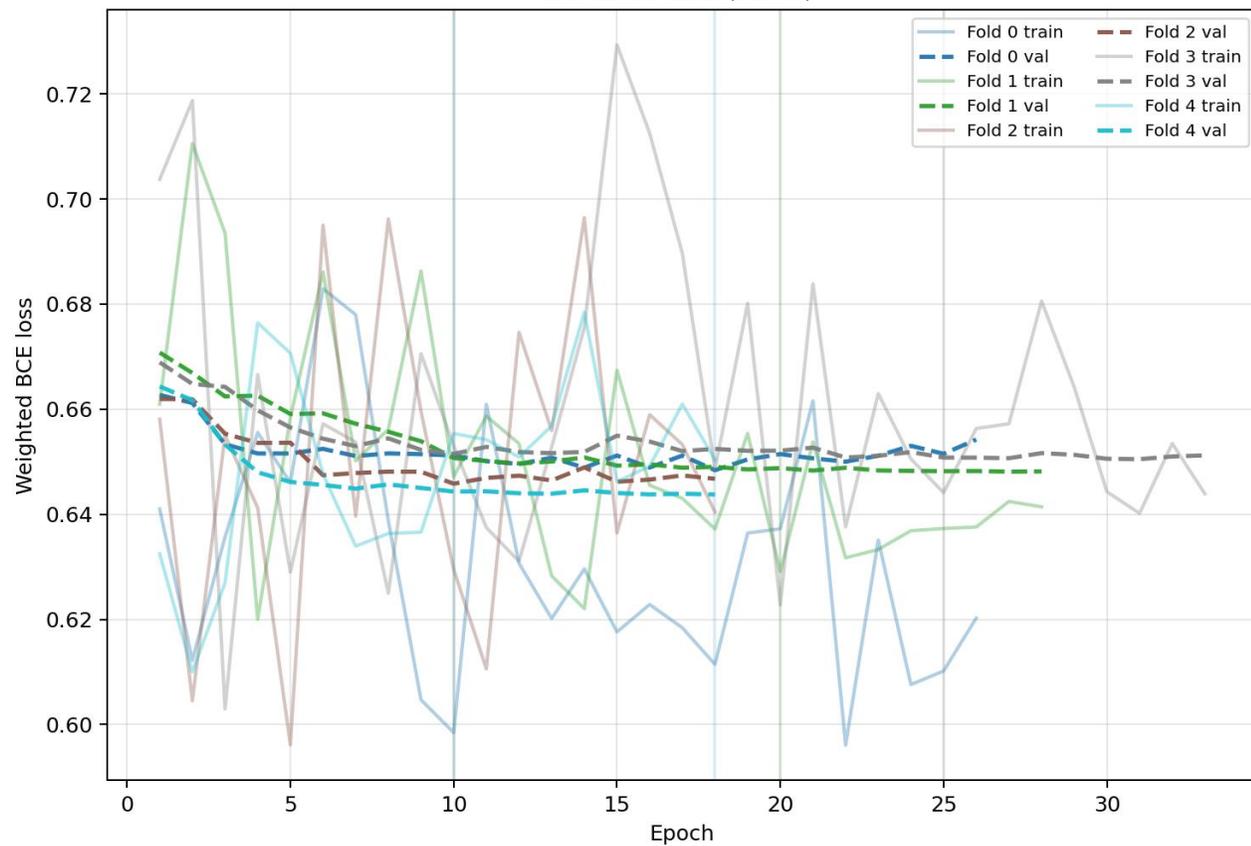
## Results from LSTM

| Fold | Model | Best_epoch | Train_auc | Val_auc  | Train_acc | Val_acc  | Train_loss | Val_loss | Runtime_sec |
|------|-------|------------|-----------|----------|-----------|----------|------------|----------|-------------|
| 0    | LSTM  | 18         | 0.687901  | 0.668634 | 0.633235  | 0.625021 | 0.636512   | 0.648389 | 3109.545    |
| 1    | LSTM  | 20         | 0.674345  | 0.666615 | 0.625498  | 0.620993 | 0.644828   | 0.64878  | 6157.698    |
| 2    | LSTM  | 10         | 0.675318  | 0.673298 | 0.628397  | 0.629227 | 0.643967   | 0.645816 | 4858.811    |
| 3    | LSTM  | 25         | 0.679743  | 0.664123 | 0.631556  | 0.615012 | 0.640638   | 0.650791 | 5837.697    |
| 4    | LSTM  | 10         | 0.674567  | 0.672803 | 0.626297  | 0.627709 | 0.644686   | 0.644331 | 7830.637    |

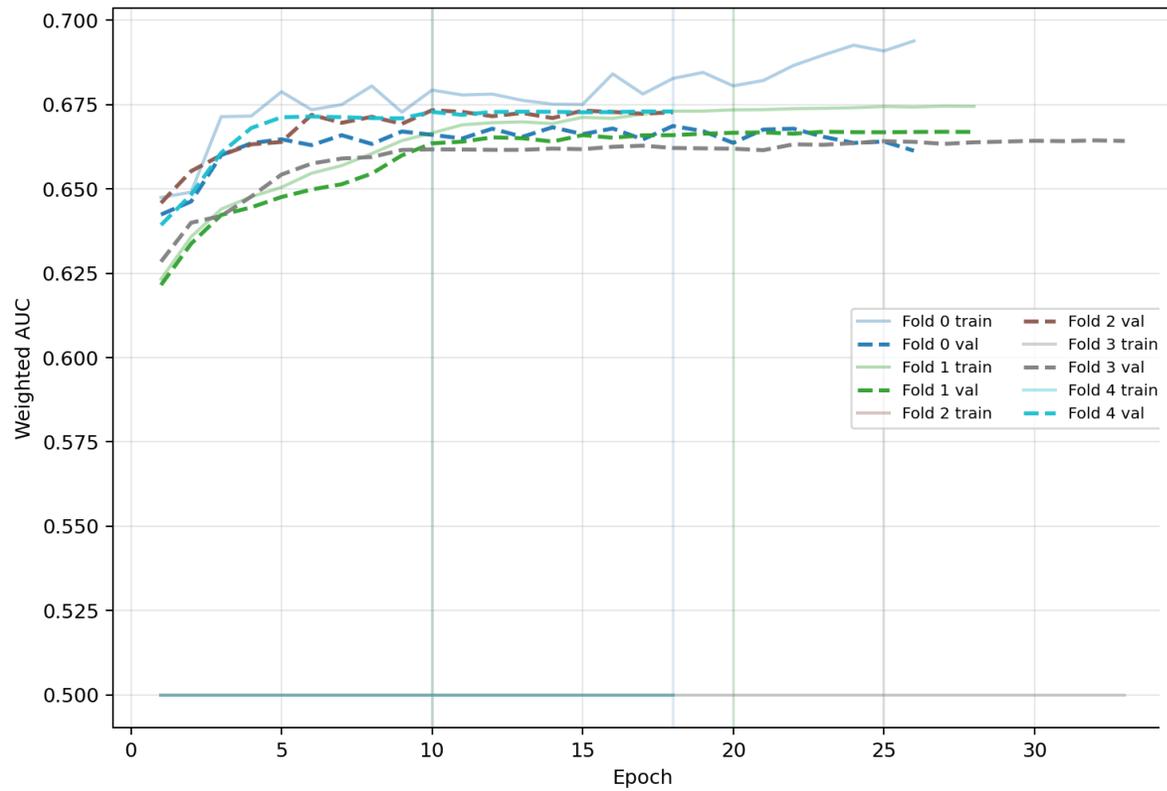
All folds: accuracy (LSTM)



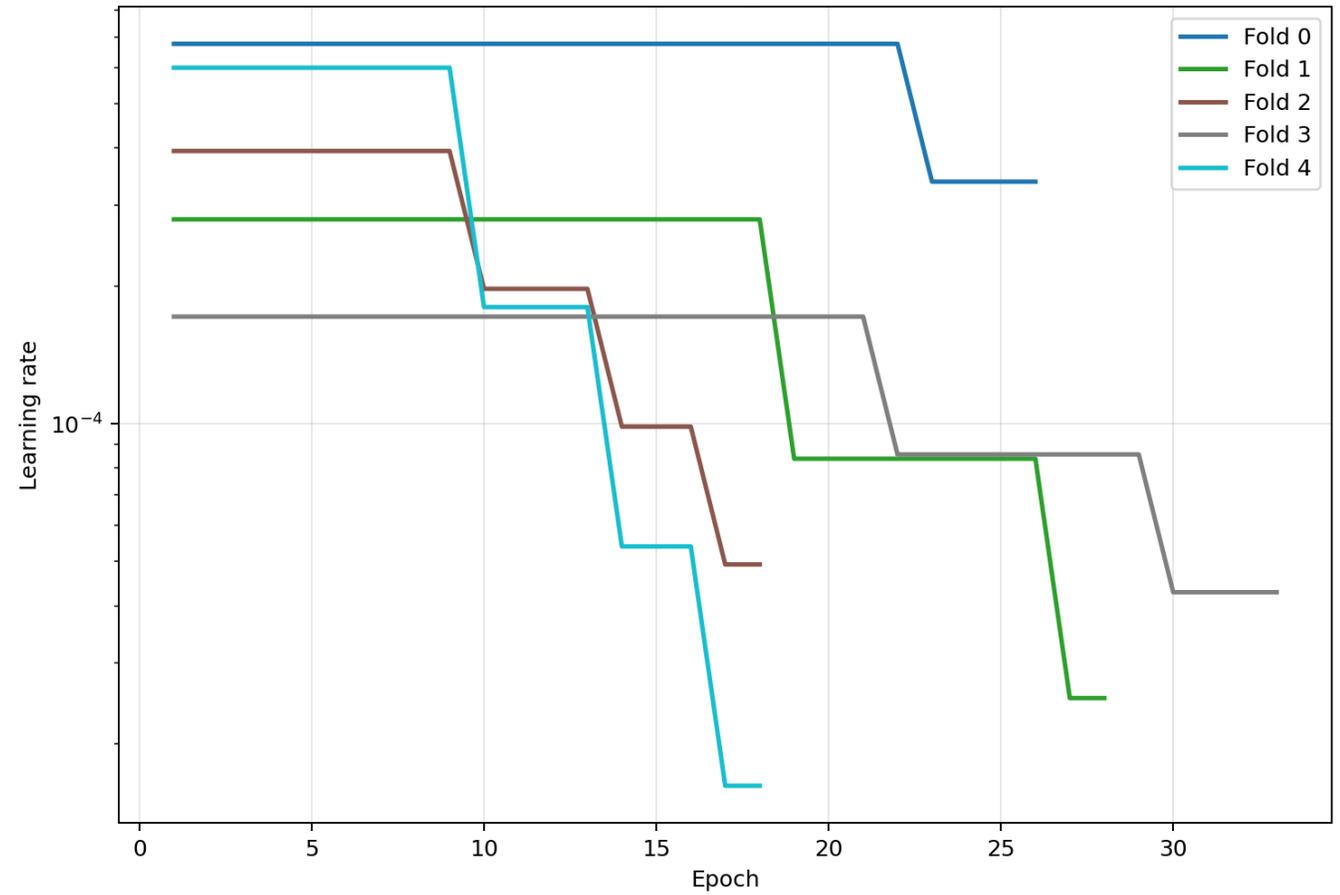
All folds: loss (LSTM)



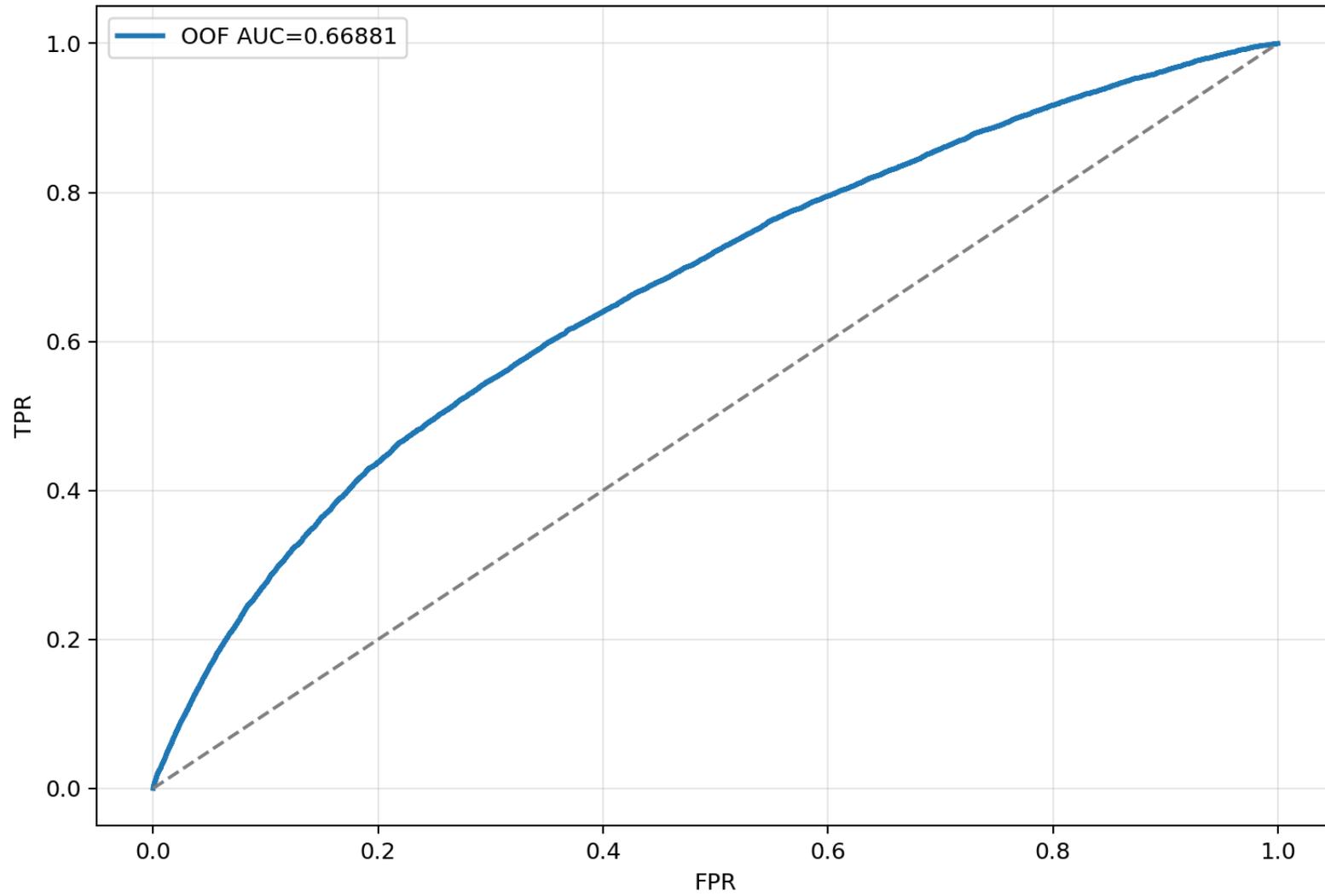
All folds: AUC (LSTM)



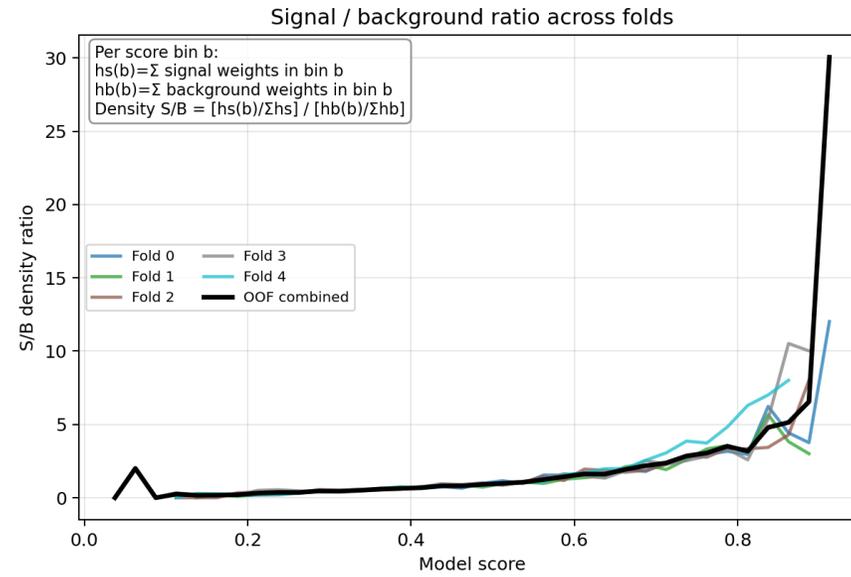
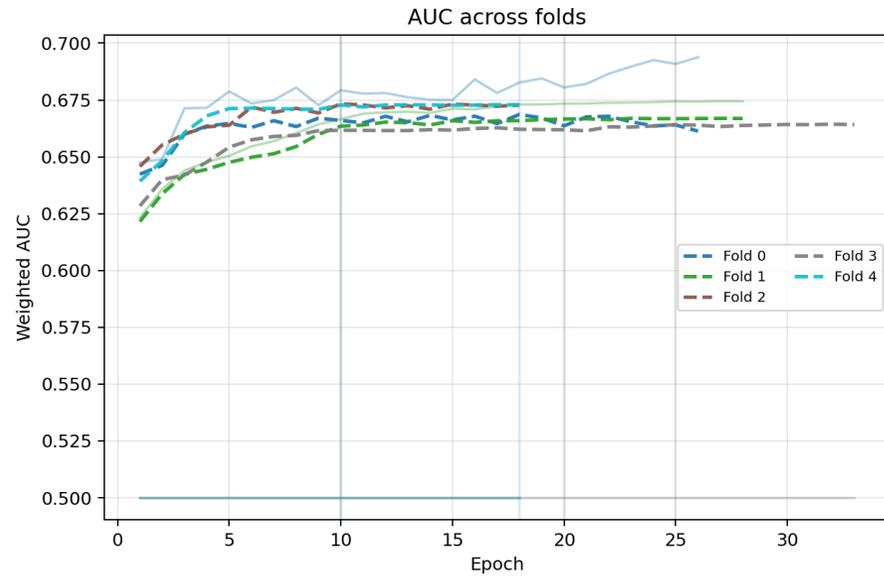
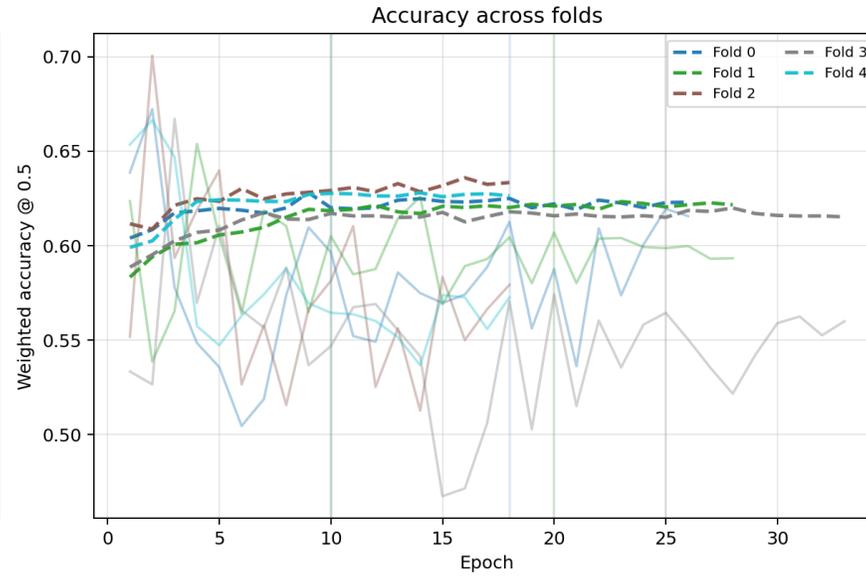
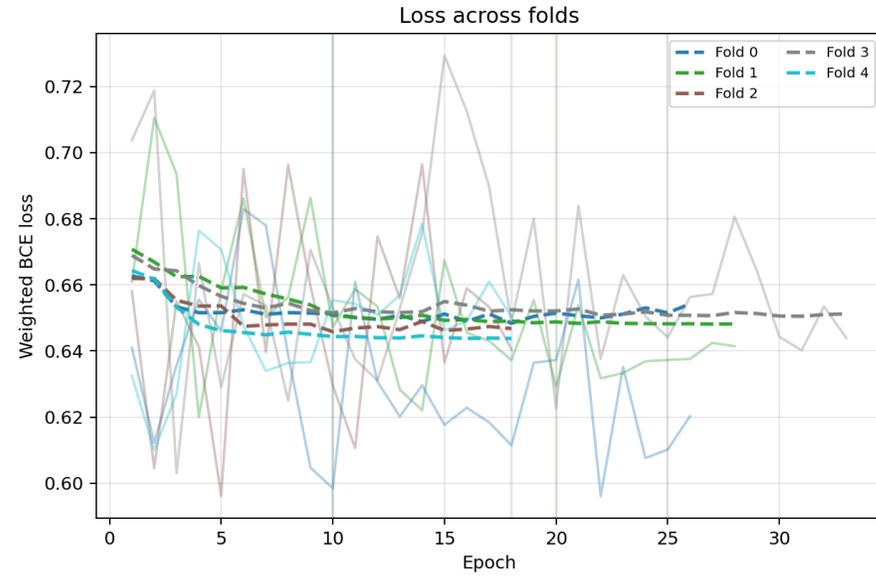
All folds: learning rate (LSTM)



Out-of-fold ROC (LSTM)



# Cross-validation dashboard (LSTM)



# Signal/background ratio plots

## Raw weighted S/B

For score bin  $b$ :

$$h_s(b) = \sum_{i \in \text{signal in bin } b} w_i$$

$$h_b(b) = \sum_{i \in \text{background in bin } b} w_i$$

Then:

$$\frac{S}{B_{\text{raw}}}(b) = \frac{h_s(b)}{h_b(b)}$$

This tells you the raw weighted signal-to-background ratio in each score region.

Reporting both raw weighted S/B and density-based S/B. The raw version reflects weighted yield purity, while the density ratio emphasizes shape separation.

## Density S/B

Computes normalized densities:

$$h_s^{\text{norm}}(b) = \frac{h_s(b)}{\sum_b h_s(b)}$$

$$h_b^{\text{norm}}(b) = \frac{h_b(b)}{\sum_b h_b(b)}$$

Then:

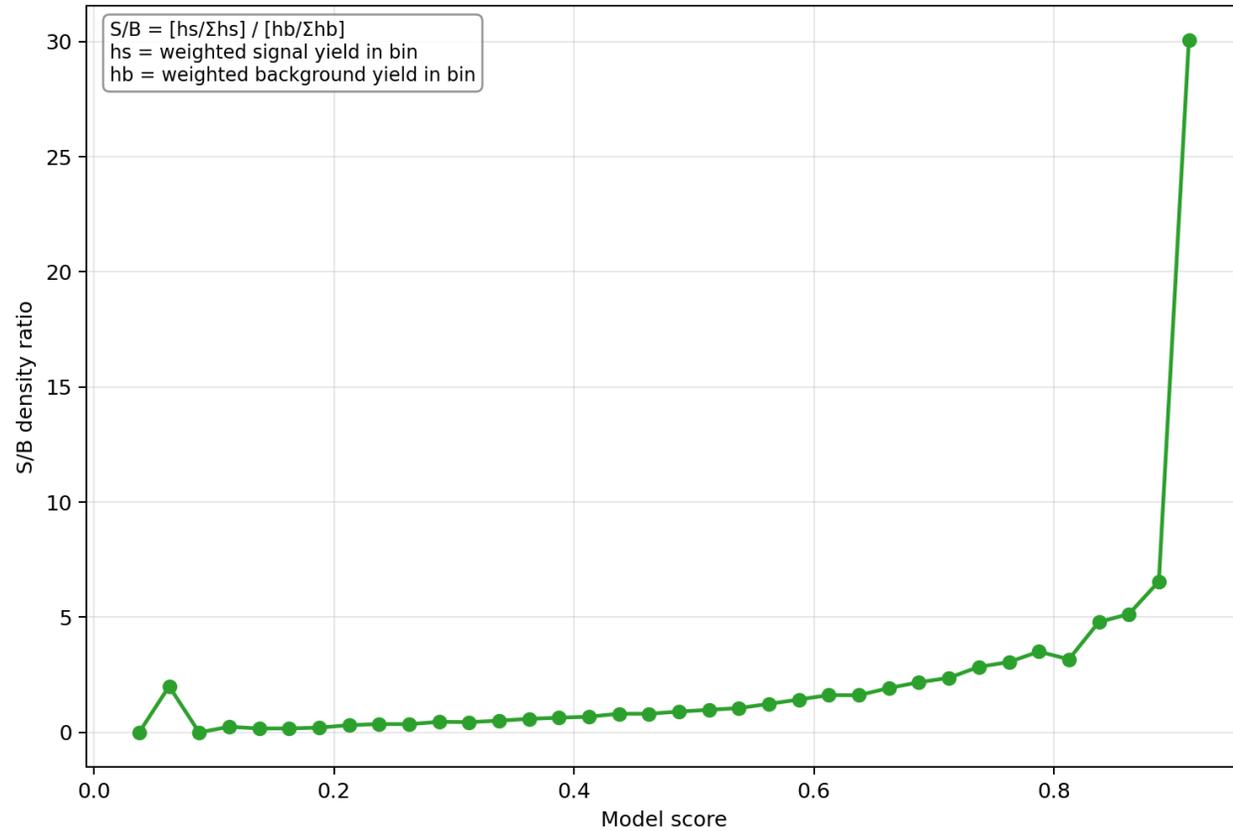
$$\frac{S}{B_{\text{density}}}(b) = \frac{h_s^{\text{norm}}(b)}{h_b^{\text{norm}}(b)}$$

## Why use density S/B

This compares **shape separation** rather than total normalization.

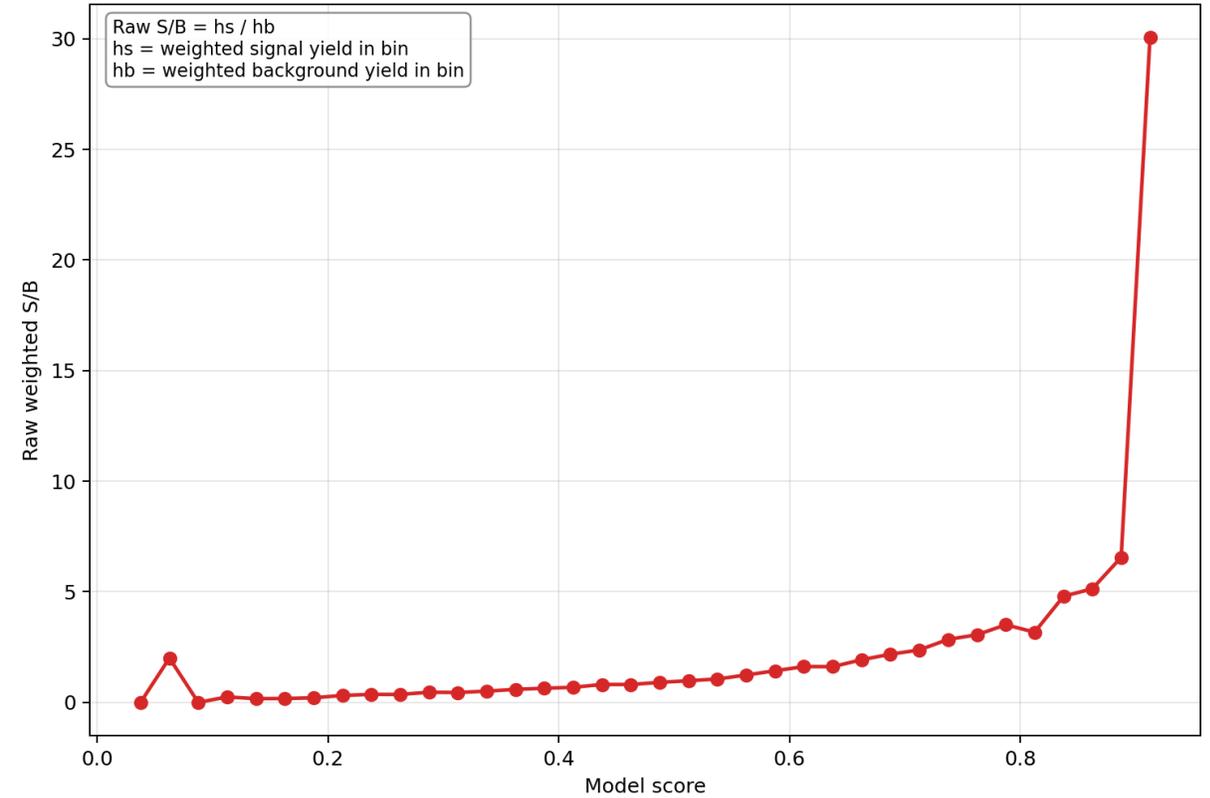
# OUT-OF-FOLD S/B CURVES

Out-of-fold S/B density ratio (LSTM)



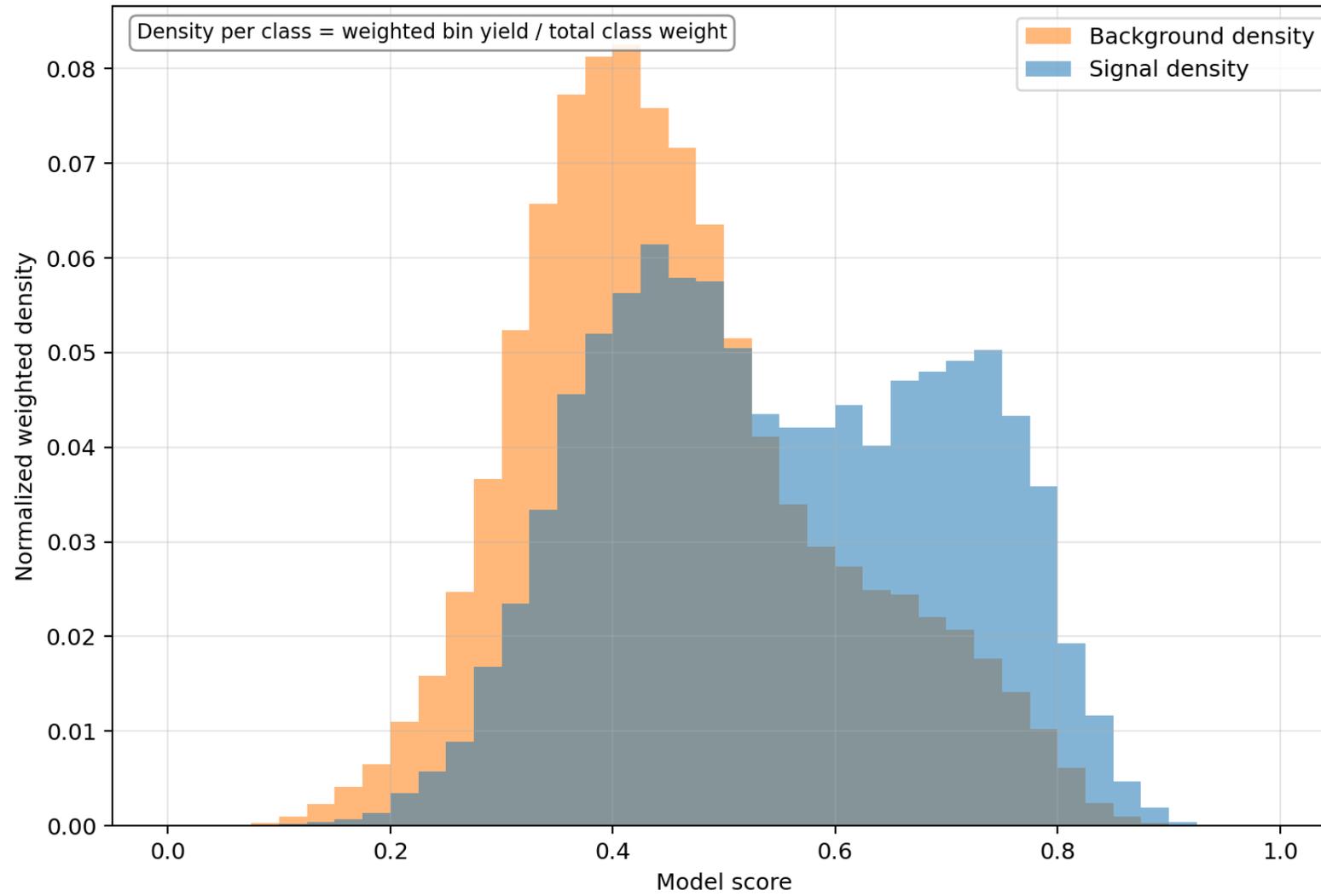
Density ratio emphasizes shape separation

Out-of-fold raw weighted S/B (LSTM)



Raw ratio reflects weighted purity per score bin

Out-of-fold score distribution (LSTM)



# Article Structure

## Literature Review Suggests:

**Concrete software platforms and specific network architectures documented in the supplied ATLAS and related papers, indicating where direct evidence exists and where the database suggests, not enough.**

|  |   |    |
|--|---|----|
| <b>Keras (Python)</b>                                  | Implementing DNNs for single-top and W/top classification in ATLAS studies<br>[explicitly cited implementations]                  | 2  |
| <b>TMVA (ROOT) BDTs</b>                                | BDT implementations used as baseline multivariate methods alongside DNNs in<br>ATLAS taggers                                      | 1  |
| <b>CNN on calorimeter images</b>                       | Whole-detector or calorimeter image classification for event/jet tasks (GPU/HPC<br>usage noted)                                   | 3  |
| <b>RNN / LSTM</b>                                      | Event topology or sequence modeling applied in ATLAS analyses (e.g., VBF studies)   | 4  |
| <b>Particle Transformer and<br/>ParticleNet</b>        | Transformer and graph-based architectures that outperform BDT baselines in<br>multi-object topologies (used in top/4-top studies) | 5  |
| <b>GN2 transformer</b>                                 | Transformer applied to jet flavour tagging at ATLAS, processing low-level tracking<br>inputs end-to-end                           | 6  |
| <b>Particle Dual Attention<br/>Transformer (P-DAT)</b> | Dual attention transformer for jet tagging capturing particle and channel interactions  | 11 |
| <b>MLP-Mixer with<br/>cross-attention</b>              | Efficient jet classification alternative to transformers, designed for computational<br>efficiency                                | 10 |
| <b>Mixture-of-Experts graph<br/>transformer</b>        | Architecture proposed for interpretable collision classification with expert<br>specialization                                    | 8  |

# Next Steps Planned

1. **Still Working on R caret package, Could not complete correct plots last week with sig/bkg ratio and identified some algorithms for comparison over Python and TMVA platforms.**
2. **Increasing number of variables for studying the effect of input features on algorithm's performance**
3. **For Plotting S/B ratio on Python, and R (transformers and LSTM)**
  - **Choosing a score cut using S/B (purity) vs efficiency trade-off;**
  - **Reporting expected yields after cut.**
4. **Continue to write the article**

# Article Structure

1. Introduction,
2. Physical Statement of the Problem
3. Data Description, Methods (Preprocessing and preparation of features, Set of input variables,
4. Description of machine learning algorithms
5. Quality assessment methodology
6. Results of comparison of algorithms,
7. Discussion, and
8. Conclusion

# Introduction Structure

- The introduction I've written contains 6 major sections:

## 1. Physics Motivation

- Importance of single-top  $tW$  production in Standard Model precision tests
- Challenges in separating  $tW$  signal from  $t\bar{t}$  background
- Relevance for higher energy physics searches

## 2. Machine Learning in ATLAS

### Comprehensive coverage of 7 algorithm families:

- Boosted Decision Trees (BDT) - Traditional baseline with TMVA
- Deep Neural Networks (DNN) - Keras/TensorFlow implementations
- Convolutional Neural Networks (CNN)
- Recurrent Networks (RNN/LSTM/GRU) - Sequence modeling
- Transformers - State-of-the-art with 30-50% performance gains
- Graph Neural Networks (GNN) - Particle-level relationships
- Hybrid Architectures - Dual-attention, mixture-of-experts

## 3. Software Platform Landscape

- TMVA: ROOT-integrated framework for BDTs and classical ML
- Python: Keras/TensorFlow, scikit-learn, PyTorch ecosystem
- R: Statistical computing with caret package (underexplored in ATLAS)

## 4. Research Gaps

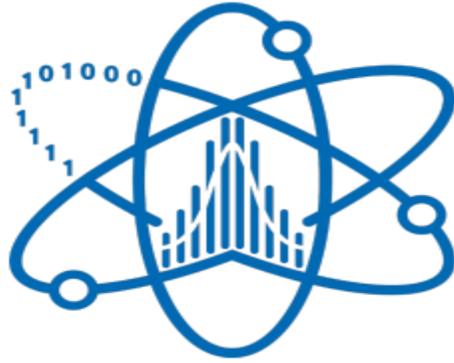
- Limited  $tW$ -specific performance benchmarks
- Lack of systematic platform comparisons
- Interpretability challenges
- Decorrelation from physics observables
- Computational accessibility

## 5. Research Objectives

- Systematic comparison across TMVA, Python, and R
- Evaluate classical and modern ML algorithms
- Establish  $tW$  vs  $t\bar{t}$  benchmarks
- Assess platform trade-offs
- Provide implementation guidance



National Research  
**Tomsk  
State  
University**



**Лаборатория  
анализа данных  
физики высоких энергий**

Томского  
государственного  
университета

**Thank you for your attention!!!**

