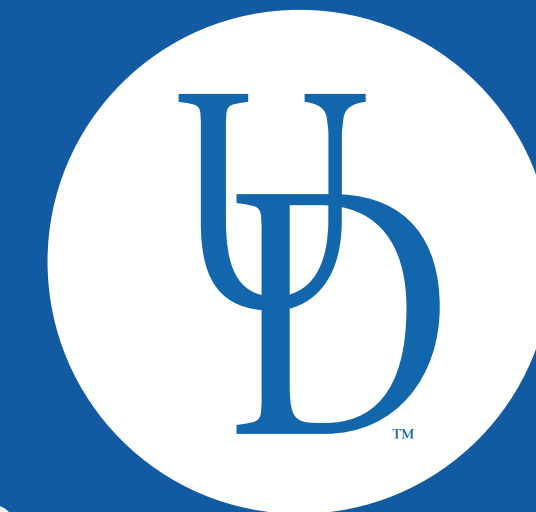




# Revealing the Power of Neural Networks to Capture Accurate Job Resource Usage from Unparsed Job Scripts and Application Inputs



Michael R. Wyatt II <sup>1,2</sup> Todd Gamblin (Advisor) <sup>1</sup> Adam Moody (Advisor) <sup>1</sup> Michela Taufer (Advisor) <sup>2</sup> | <sup>1</sup> Lawrence Livermore National Laboratory <sup>2</sup> University of Delaware

## Resource Prediction Challenges

- Next generation HPC schedulers rely on information about job resource usage to improve machine utilization [1]
- Analysis of job scripts via machine learning models can be used to improve knowledge of resource usage for HPC jobs [2,3]
  - Accurate Resource Usage Knowledge → Better Scheduling
- Traditional job script analysis (i.e., learning from parsed features of job scripts) represents each job with a limited number of features
  - Jobs look the same from the perspective of parsed features

```
#!/bin/bash
#PBS -l partition=cab
#PBS -l nodes=4
#PBS -l walltime=16:00:00
#PBS -q pbatch

cd $HOME/project_A
srun -n 64 ./GEOS -i Prop_bx_2a.xml
```

<cab, 4, 16, pbatch, GEOS>  
 ↓ F(x) = ?  
**Runtime: 12 hours**

```
#!/bin/bash
#PBS -l partition=cab
#PBS -l nodes=4
#PBS -l walltime=16:00:00
#PBS -q pbatch

cd $HOME/dev_new
srun -n 64 ./GEOS -i Prop_bx_1b.xml
```

<cab, 4, 16, pbatch, GEOS>  
 ↓ F(x) = ?  
**Runtime: 6 hours**

Traditional job script parsing cannot differentiate between these job scripts and provide accurate resource usage prediction

- Application input decks (i.e., input parameters) affect the amount of resources used by a job
- Small differences between two job scripts or input decks often create large differences in resource usage

## Revamping Job Script Roles

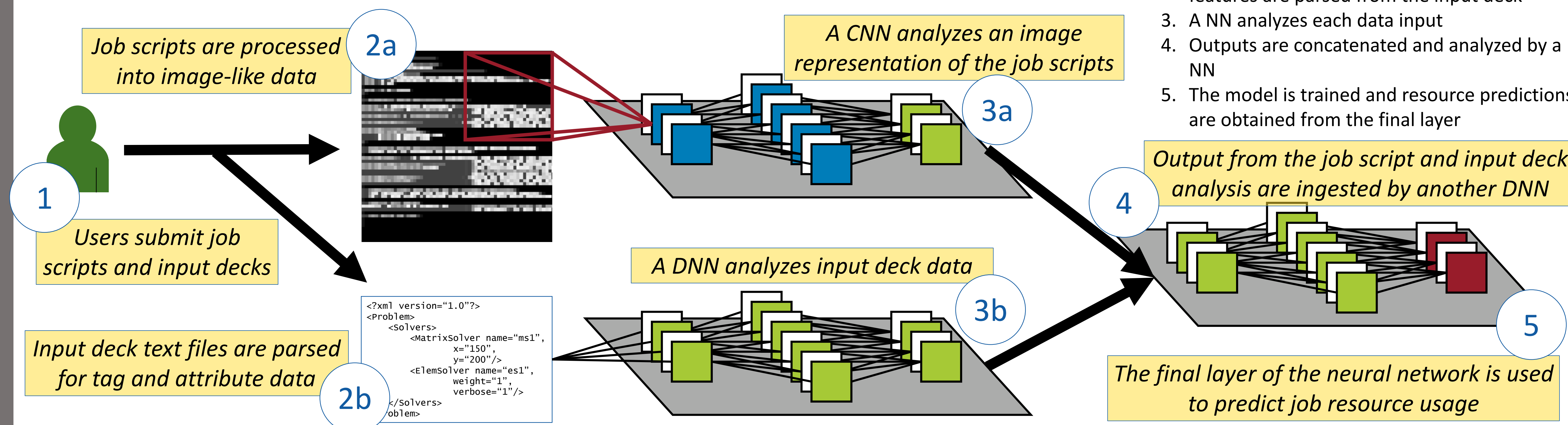
- We expand the representation of a job beyond parsed features:
  - Entire job script text
  - Application input decks
- A 2D Convolution Neural Network (CNN) analyzes entire job scripts
  - CNNs automatically detect and learn features in unparsed text
- A Deep Neural Network (DNN) analyzes application input deck features
  - Features are automatically parsed from the structured text

```
<Problem>
  <Solvers>
    <MatSolver name="ms1",
      x="150",
      y="200"/>
    <ElemSolver name="es1",
      weight="1",
      verbose="1",
      prof="3"/>
  </Solvers>
</Problem>
```

Application input decks are structured text data that are parsed for features (highlighted in purple)

## Learning from Job Script and Application Input Decks

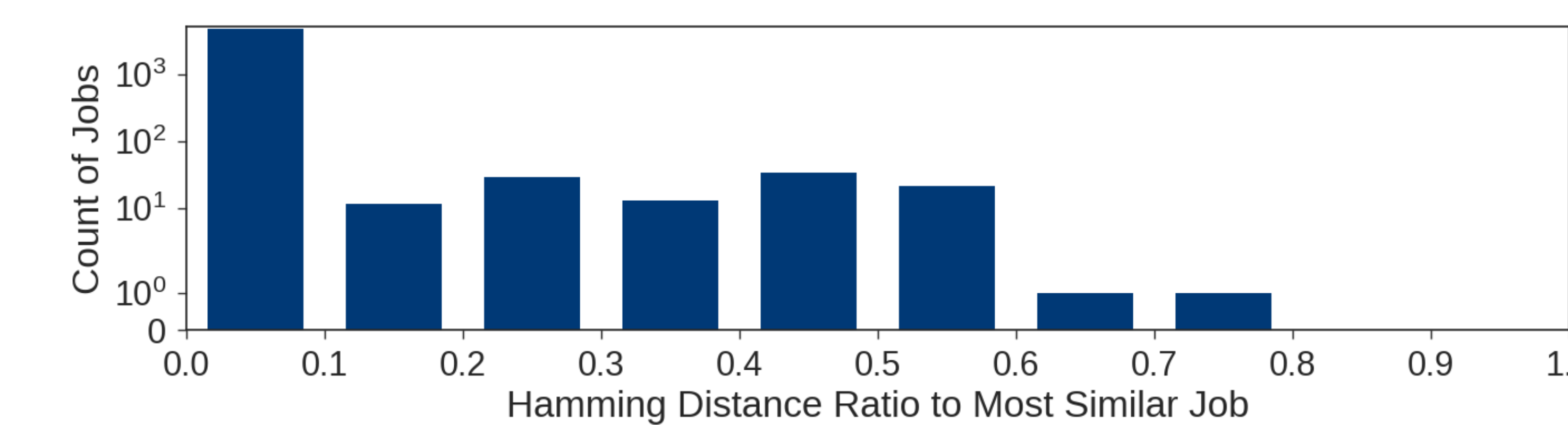
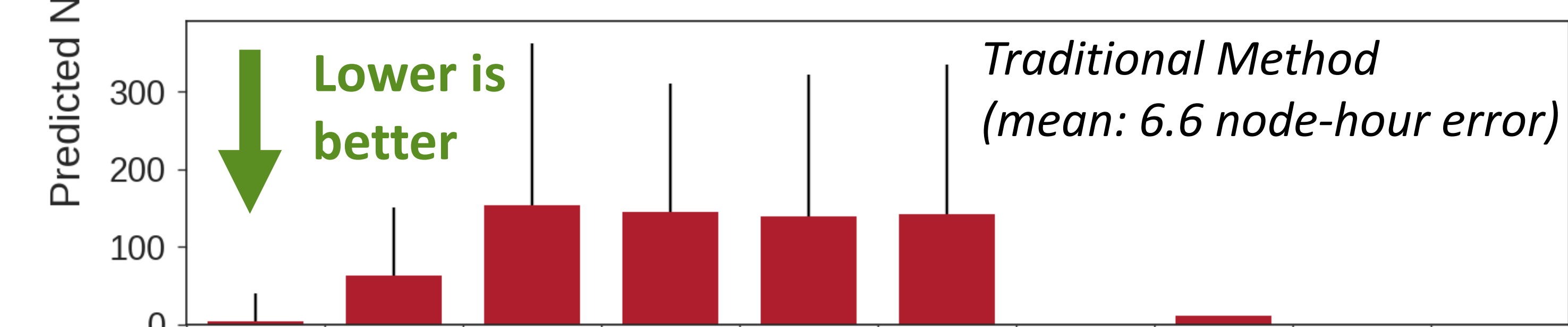
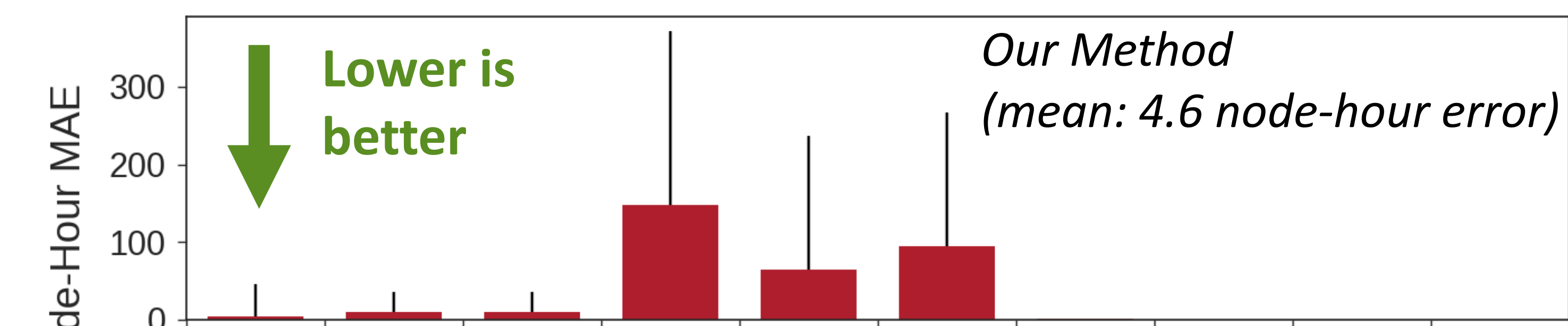
Can we leverage neural networks to predict resource use from just unparsed job scripts and application inputs?



- Users submit their job scripts and input decks
- Job scripts transform into image-like data and features are parsed from the input deck
- A NN analyzes each data input
- Outputs are concatenated and analyzed by a NN
- The model is trained and resource predictions are obtained from the final layer

## Observed Runtime Prediction Accuracy

- We validate our methods with data collected on over 5,000 jobs for the application GEOS executed on the Cab machine at LLNL
  - We collect job scripts, input decks, and execution information about each job
- Runtime (i.e., time between job start and finish) is predicted for each job in our dataset with 10-fold crossvalidation
- We compare our method against the traditional method for prediction accuracy
  - Parsed features from job scripts and **Random Forest Regressor**
- Prediction error is reported with **node-hour Mean Absolute Error (MAE)** and organized by hamming distance ratio to the most similar job script
 
$$\text{hammingDistanceRatio}(i, j) = \frac{\text{hammingDistance}(i, j)}{\max(\text{length}(i), \text{length}(j))}$$
  - Over **95%** of GEOS job scripts share more than **90%** of text with another GEOS job script
  - We decrease runtime prediction error from **6.6 node-hour MAE** with traditional methods to **4.6 node-hour MAE** with our neural network.
    - 420 node-day** reduction in error across all GEOS jobs



Jobs are organized by hamming distance ratio of the most similar job script. The bottom shows distribution of GEOS jobs. The top and middle show our methods reduce runtime prediction error over traditional methods

References:  
 [1] Herbein, Stephen, et al. "Scalable I/O-Aware Job Scheduling for Burst Buffer Enabled HPC Clusters." *Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing*. ACM, 2016.  
 [2] Smith, Warren, Valerie Taylor, and Ian Foster. "Using run-time predictions to estimate queue wait times and improve scheduler performance." *JSSPP*. 1999.  
 [3] McKenna, Ryan, et al. "Machine Learning Predictions of Runtime and IO Traffic on High-End Clusters." *Cluster Computing (CLUSTER)*, 2016 IEEE International Conference on. IEEE, 2016.