Introduction to Hepynet

a (D)NN assistant framework for HEP analysis

Zhe Yang (zheya@umich.edu)

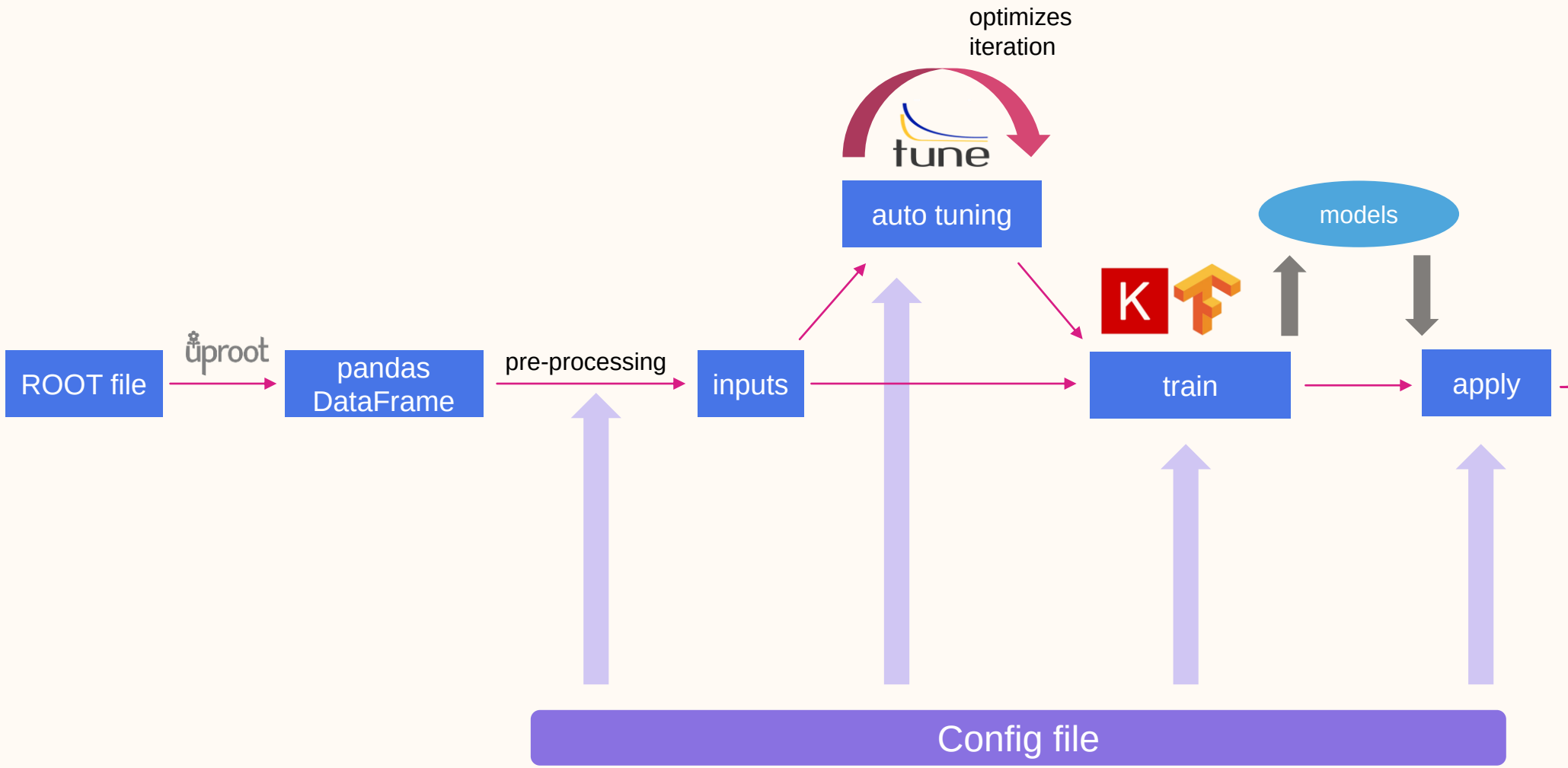
on behalf of the team

Idea of Hepynet

GitHub: Hepynet/hepynet

- What does Hepynet mean?
  - High energy physics, python-based, neural-network assistant framework
- What does Hepynet do?
  - major features
    - for HEP Analysis: performs typical MVA studies in physics analysis with NN
    - Config Driven: all tasks defined by simple config file(s)
    - Python Based: codes are written in pure Python
  - based on Keras with TensorFlow backend
- The goal of the Hepynet
  - quick set up for new studies/analyses
    - no need to write any code for NN as users
  - good records of NN study history
    - each config file defines a study
  - joint efforts of developers
    - the functionality developed by one person can be applied to another person's analysis easily

Workflow



- DNN scores
- ROC/AUC
- metrics history
- kinematics
- significance scan
- fitting inputs
- feature importance
- train/test compare
- correlations
- epoch checks
- …

Core Packages Used in Hepynet

- Data I/O
  - uproot: ROOT files ↔ pandas DataFrame
  - pandas: data processing for training & evaluation
- Training
  - Keras/TensorFlow: neural network framework widely used in industry
  - scikit-learn: assists training process
- Tuning
  - Ray-Tune: experiment execution and hyperparameter tuning at any scale
- Plot
  - matplotlib: python plotting

Config files

- Use YAML format to specify the details of a training
- 3 different type of jobs/configs
  - train/tune/apply
- Including 6+1 different sections
  - config: include other config files
  - job: config job type and job properties
  - input: specify input files and pre-processing procedures
  - train: define the model structure/strategy for the training
  - tune: schedule the auto hyperparameter tuning task
  - apply: book evaluation jobs to check model properties and dump output for fitting
  - run: contains some information needed during running
    - dynamically created during running

Config files Sections – config/job/input

- config
  - support recursive including, overwrite in order
- job
  - set up job type, save directory, random seed and other general job settings
- input
  - specify input source
  - pre-processing procedures

```yaml
config:
    include:
        - "share/zprime/input/input_hm.sec.yaml"
        - "share/zprime/train/train_hm.sec.yaml"

job:
    job_name: "default_train"
    job_type: "train"
    save_dir: "run/zprime/default_train/high_mass"
```

```yaml
input:
    input_path: "zprime/data_frames/21-0120-sys/high_mass/zprime_high_m_quadtype_2.feather"
    norm_array: true
    bkg_key: "all"
    sig_key: "all_norm"
    bkg_sumofweight: 100000
    sig_sumofweight: 100000

    bkg_list:
        - "bkg_qcd"
        - "bkg_ggZZ"
    sig_list:
        - "sig_Zp042"
        - "sig_Zp045"
        - "sig_Zp048"
        - "sig_Zp051"
        - "sig_Zp054"

    selected_features:
        - "m_truth"
        - "ptl1"
        - "ptl2"
        - "ptl3"
        - "ptl4"
        - "mz1_mz2"
        - "ptz1"
        - "ptz2"
        - "mzz"
        - "ptzz"

    validation_features:
        - "mz1"
        - "mz2"

    reset_feature: true
    reset_feature_name: "m_truth"
    rm_negative_weight_events: true
```

Config Files Sections - train

- train
  - choose model class
  - set up hyperparameters
  - set up monitored metrics
  - set up early-stopping strategy

```yaml
train:
    model_name: "zprime_pdnn_model"
    model_class: "Model_Sequential_Flat"
    layers: 2
    nodes: 256
    dropout_rate: 0
    momentum: 0.5
    nesterov: true
    test_rate: .2
    val_split: .25
    k_folds: 5
    learn_rate: 0.0391378
    learn_rate_decay: 0.00853803
    batch_size: 512
    epochs: 6
    train_metrics_weighted:
        - "accuracy"
        - "auc"

    # early stop setups
    use_early_stop: true
    early_stop_paras:
        monitor: "val_auc"
        min_delta: 0.0005
        patience: 3
        mode: "max"
        restore_best_weights: true

    # save model or not
    save_model: true

    verbose: 1
```

Config Files Sections - tune

- tune
  - set up algorithms for automatic hyperparameters tuning
    - multiple algorithm supported by Ray: Optuna, HyperOpt, BayesOpt…
  - set up search space
    - dimensions
    - range, distribution…

Note: the search progress can be resumed if the job stopped during the running accidently

```yaml
tune:
    tuner:
        scheduler_class: AsyncHyperBandScheduler
        scheduler:
            time_attr: training_iteration
            max_t: 50
            grace_period: 2
        algo_class: HEBOSearch
        algo:
            metric: min_limit
            mode: min
            max_concurrent: 16
        stopper_class: TrialPlateauStopper
        stopper:
            metric: min_limit_delta
            metric_threshold: 0
            mode: min
        run:
            metric: min_limit
            mode: min
            num_samples: 500
            resources_per_trial:
                cpu: 1
                gpu: 0
            log_to_file: true
    model_class: "Model_Sequential_Flat"
```

```yaml
model:
    layers:
        spacer: randint
        paras:
            lower: 2
            upper: 8
    nodes:
        spacer: choice
        paras:
            categories: [16, 32, 64, 128, 256]
    dropout_rate:
        spacer: quniform
        paras:
            lower: 0
            upper: 0.9
            q: 0.1
    momentum:
        spacer: quniform
        paras:
            lower: 0
            upper: 0.9
            q: 0.1
    nesterov: true
    test_rate: .2
    val_split: .25
    learn_rate:
        spacer: qloguniform
        paras:
            lower: 1.e-5
            upper: 1
            q: 5.e-6
    learn_rate_decay:
        spacer: qloguniform
        paras:
            lower: 1.e-5
            upper: 0.01
            q: 5.e-6
    batch_size:
        spacer: choice
        paras:
            categories: [32, 128, 256, 512, 1024]
```

Config Files Sections - apply

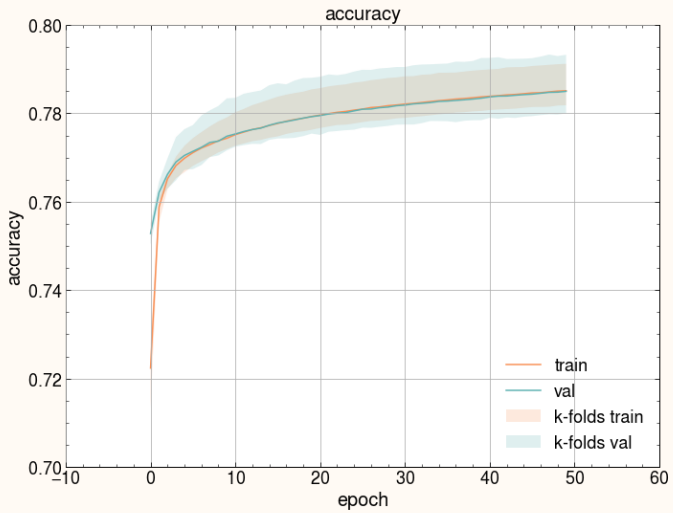
```yaml
apply:
    plot_atlas_label: True
    atlas_label:
        status: "wip"
        #simulation: "true"
        energy: "13 TeV"
        lumi: 139

    #check_model_epoch: true
    epoch_check_interval: 20

    book_history: true
    cfg_history:
        accuracy:
            plot_title: "accuracy history"
            save_format: "png"
        loss:
            plot_title: "loss history"
            save_format: "png"

    book_kine: true
    cfg_kine:
        bins: 40
        separate_bkg_sig: false

        m_truth:
            x_label: $m_{truth} [GeV]$
            bins: 80
            range_raw: [20, 110]

        mz1:
            x_label: $m_{Z_1} [GeV]$
            range_raw: [20, 110]
        mz2:
            x_label: $m_{Z_2} [GeV]$
            range_raw: [0, 100]
```

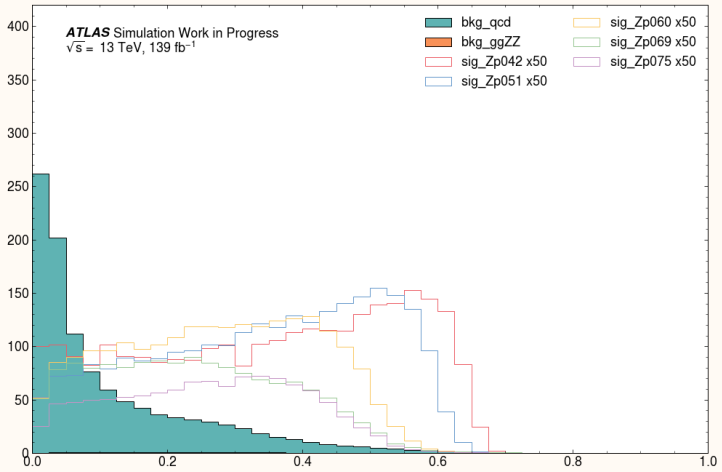
```yaml
book_fit_inputs: false
fit_df:
    branches:
        - "mz1"
        - "mz2"
    save_dir: "zprime/data_frames_fit/21-0120-sys/high_mass"

book_roc: true

book_train_test_compare: true
cfg_train_test_compare:
    plot_title: "train/test MVA scores compare"
    bins: 25
    range: [0, 1]
    density: true
    log: true
    save_format: "png"

book_mva_scores_data_mc: true
cfg_mva_scores_data_mc:
    sig_list:
        - "sig_Zp042"
        - "sig_Zp051"
        - "sig_Zp060"
        - "sig_Zp069"
        - "sig_Zp075"
    bkg_list:
        - "bkg_ggZZ"
        - "bkg_qcd"
    apply_data: false
    apply_data_range:
    plot_title: "MVA scores"
    bins: 40
    range: [0, 1]
    density: false
    sig_scale: 50
    log: true
    save_format: "png"
    use_root: false
```

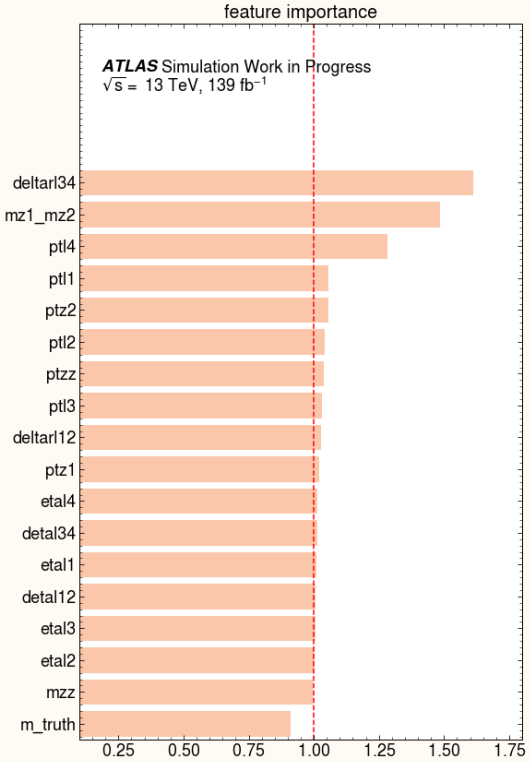
```yaml
book_significance_scan: true
cfg_significance_scan:
    significance_algo: "s_sqrt_sb_rel"

book_cut_kine_study: true
cfg_cut_kine_study:
    mz1:
        x_label: "$m_{Z1}$ (normalized value for training)"
        range_processed: [-3, 3]
    bins: 40
    range:
    histtype: "stepfilled"
    alpha: 0.3
    density: true
    save_format: "png"
    save_ratio_table: true
    dnn_cut_list:
        - 0.1
        - 0.2
        - 0.3
        - 0.4
        - 0.5

book_importance_study: true
cfg_importance_study:
    log: false
```

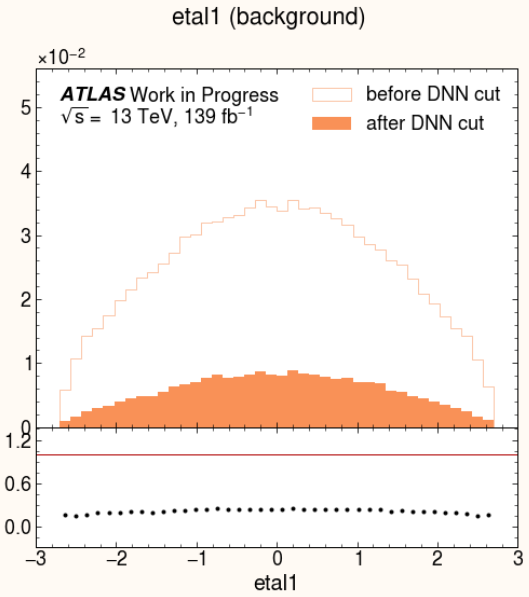
- config
  - book and set up needed studies

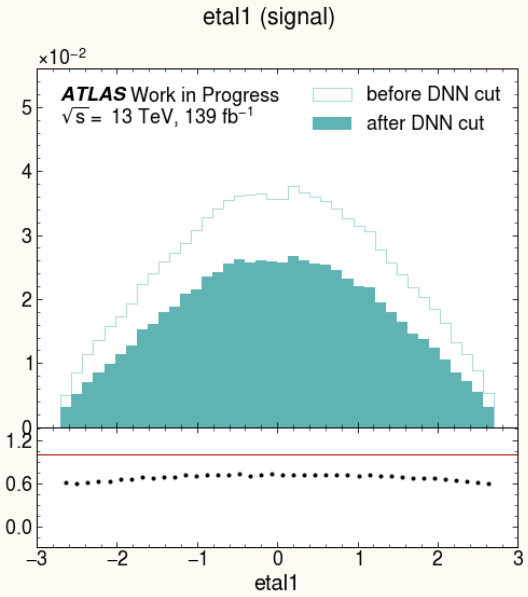
Part of Outputs Examples

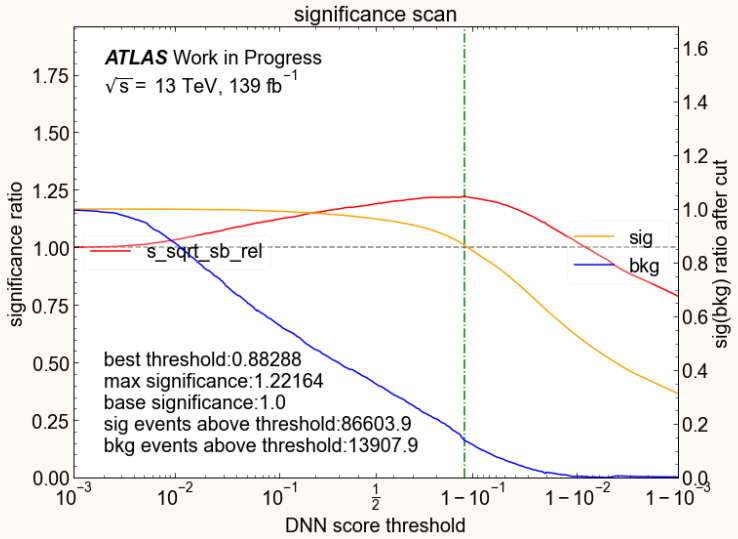


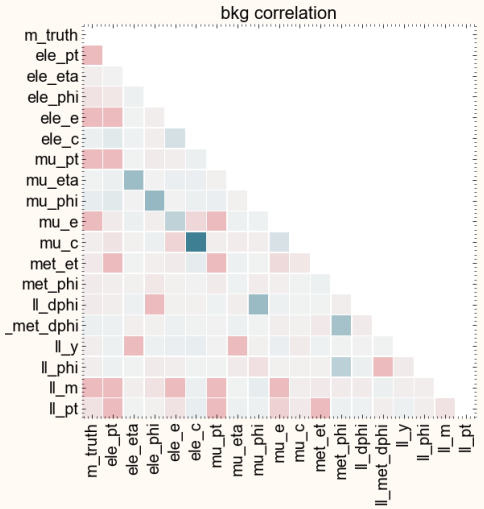












Summary

- Hepynet is being developed to help set up (D)NN studies in HEP analysis with simple configs
- The development is still at early stage
  - being improved based on usage at several ongoing analysis
- Suggestions and joint efforts for development as user/developer are really welcomed and appreciated

Backups

Recent Development Status

- Short-term update plan
  - [TODO] multi-class training support
    - already existed before
    - need to update for new data structure
  - [Done] Bayesian optimization functionality
    - already existed before
    - need to improve flexibility
    - need to rethink about how to set up surrogate model better
  - [WIP] better documentations
  - [WIP] bug-fix during usage in different analysis

- Long-term update directions
  - better plotting
  - support more DNN strategy (for example CWola)
  - convert model to the formats that can be used in analysis framework (ONNX)
  - customized NN code injection
  - performance optimization for large data (> 10 GB)
  - …

Usage

```
usage: hepynet [-h] [-d] [-r] [-t] [-v] [yaml_configs [yaml_configs ...]]

positional arguments:
  yaml_configs

optional arguments:
  -h, --help     show this help message and exit
  -d, --debug    run in debug mode
  -r, --resume   resume (tune) job
  -t, --time     display time
  -v, --verbose  verbose debug information
```

Running Outputs Example (train)

```
Using TensorFlow backend.
INFO GPU availability: 1 Physical GPUs, 1 Logical GPUs
INFO Configs in queue:
INFO .\share\lfv\master\default\emu\train_zprime.yaml
INFO ##########################################################################################
INFO Executing: .\share\lfv\master\default\emu\train_zprime.yaml
INFO Included config: share/lfv/input/input_emu_zprime.sec.yaml
INFO Included config: share/lfv/train/train_emu_zprime.sec.yaml

Config details >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[config]
    include :
        - share/lfv/input/input_emu_zprime.sec.yaml
        - share/lfv/train/train_emu_zprime.sec.yaml
    best_tune_overwrite : False
[job]
    job_name : train_zprime
    job_type : train
    save_dir : run/lfv/default/emu
    tune_job_name : TUNE_JOB_NAME_DEF
    train_job_name : TRAIN_JOB_NAME_DEF
    fix_rdm_seed : True
    rdm_seed : 1024
    date_str : None
    fix_date_str : False
[input]
    input_path : lfv/data_frames/v8/lfv_emu.feather
    reshape_input : True
    reweight_input : True
    negative_weight_process : to_zero
    feature_norm_alias :
    sig_sumofweight : 100000
    bkg_sumofweight : 100000
    data_sumofweight : 1000
```

```
INFO Setup work directory: run/lfv/default/emu/2021-07-06_train_zprime_v00
INFO Setting up model
INFO Processing inputs
INFO Loading processed input DataFrame...
INFO Loading raw input DataFrame...
INFO > Loading samples in list: ['sig_zp_500', 'sig_zp_700', 'sig_zp_1000', 'sig_zp_1500', 'sig_zp_2000', 'bkg_diboson', 'bkg_zll', 'bkg_top', 'bkg_wjets']
INFO > Cutting inputs according to expression: ll_m > 130
INFO > Successfully loaded raw input DataFrame...
INFO > Reshaping inputs
INFO Loading raw input DataFrame...
INFO > Loading samples in list: ['sig_zp_500', 'sig_zp_700', 'sig_zp_1000', 'sig_zp_1500', 'sig_zp_2000', 'bkg_diboson', 'bkg_zll', 'bkg_top', 'bkg_wjets']
INFO > Cutting inputs according to expression: ll_m > 130
INFO > Successfully loaded raw input DataFrame...
INFO > > Recalculated norm factors for feature ll_pt mean: 106.44026947021484, variance: 10268.0224609375
INFO > > Recalculated norm factors for feature mu_eta mean: -0.026888558641076088, variance: 1.9466654062271118
INFO > > Recalculated norm factors for feature ll_met_dphi mean: 0.9492524862289429, variance: 5.310612678527832
INFO > > Recalculated norm factors for feature mu_phi mean: -0.024281151592731476, variance: 3.362229585647583
INFO > > Recalculated norm factors for feature ele_c mean: 0.0037863103207200766, variance: 0.9999856352806091
INFO > > Recalculated norm factors for feature ll_phi mean: -0.05810349062085152, variance: 3.257352352142334
INFO > > Recalculated norm factors for feature m_truth mean: 765.1669921875, variance: 32040.111328125
INFO > > Recalculated norm factors for feature ele_pt mean: 248.3692169189453, variance: 14154.2841796875
INFO > > Recalculated norm factors for feature ele_e mean: 470.970703125, variance: 76534.7421875
INFO > > Recalculated norm factors for feature ele_eta mean: 0.00015733839245513082, variance: 1.704262137413025
INFO > > Recalculated norm factors for feature ll_dphi mean: 1.4168460369110107, variance: 6.866092681884766
INFO > > Recalculated norm factors for feature mu_e mean: 519.973876953125, variance: 97699.609375
INFO > > Recalculated norm factors for feature ll_m mean: 765.1669921875, variance: 32040.111328125
INFO > > Recalculated norm factors for feature met_phi mean: -0.0467032752931118, variance: 3.248337984085083
INFO > > Recalculated norm factors for feature mu_pt mean: 254.57571411132812, variance: 17091.560546875
INFO > > Recalculated norm factors for feature ll_y mean: -0.009456136263906956, variance: 0.5015740990638733
INFO > > Recalculated norm factors for feature met_et mean: 91965.109375, variance: 7951208960.0
INFO > > Recalculated norm factors for feature ele_phi mean: -0.014659207314252853, variance: 3.2278099060058594
INFO > > Recalculated norm factors for feature mu_c mean: -0.0037863103207200766, variance: 0.9999856352806091
INFO > Processing negative weights with method: to_zero
INFO > Reweighting inputs
INFO > Resetting physic para m_truth distribution for pDNN
INFO > Setting up y values
INFO > Tagging train / test
INFO > Successfully loaded processed input DataFrame...
INFO ------------------------------------
INFO Loading inputs
INFO Training start. Using model: lfv_pdnn_model
INFO Model info: Sequential model with flexible layers and nodes.
INFO Performing k-fold training 1/5
Model: "sequential_1"
```

```
INFO Training start. Using model: zprime_pdnn_model
INFO Model info: Sequential model with flexible layers and nodes.
INFO Performing k-fold training 5/5
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_13 (Dense)             (None, 256)               4864
_________________________________________________________________
dense_14 (Dense)             (None, 256)               65792
_________________________________________________________________
dense_15 (Dense)             (None, 1)                 257
=================================================================
Total params: 70,913
Trainable params: 70,913
Non-trainable params: 0
_________________________________________________________________
INFO > Training on 513087 events -> composition: {0: 341460, 1: 171627}
INFO > Validating on 128271 events -> composition: {0: 85364, 1: 42907}
Train on 513087 samples, validate on 128271 samples
Epoch 1/6
513087/513087 [==============================] - 3s 6us/step - loss: 0.4204 - accuracy: 0.9483 - auc: 0.7841 - val_loss: 0.4032 - val_accuracy: 0.9478 - val_auc: 0.8189
Epoch 2/6
513087/513087 [==============================] - 3s 5us/step - loss: 0.3992 - accuracy: 0.9483 - auc: 0.8259 - val_loss: 0.3956 - val_accuracy: 0.9478 - val_auc: 0.8305
Epoch 3/6
513087/513087 [==============================] - 3s 5us/step - loss: 0.3935 - accuracy: 0.9483 - auc: 0.8337 - val_loss: 0.3914 - val_accuracy: 0.9478 - val_auc: 0.8364
Epoch 4/6
513087/513087 [==============================] - 3s 5us/step - loss: 0.3899 - accuracy: 0.9483 - auc: 0.8385 - val_loss: 0.3884 - val_accuracy: 0.9478 - val_auc: 0.8403
Epoch 5/6
513087/513087 [==============================] - 3s 5us/step - loss: 0.3873 - accuracy: 0.9483 - auc: 0.8417 - val_loss: 0.3861 - val_accuracy: 0.9478 - val_auc: 0.8431
Epoch 6/6
513087/513087 [==============================] - 3s 5us/step - loss: 0.3852 - accuracy: 0.9483 - auc: 0.8443 - val_loss: 0.3843 - val_accuracy: 0.9479 - val_auc: 0.8453
INFO Training finished.
INFO Evaluate with test dataset:
160340/160340 [==============================] - 6s 37us/step
INFO > test - loss: 0.3821345170243535
INFO > test - accuracy: 0.9484157562255859
INFO > test - auc: 0.8458167314529419
INFO Time consumed: 00:02:04
INFO ##############################################################################
INFO Done!
INFO ##############################################################################
```