Machine learning in HEP

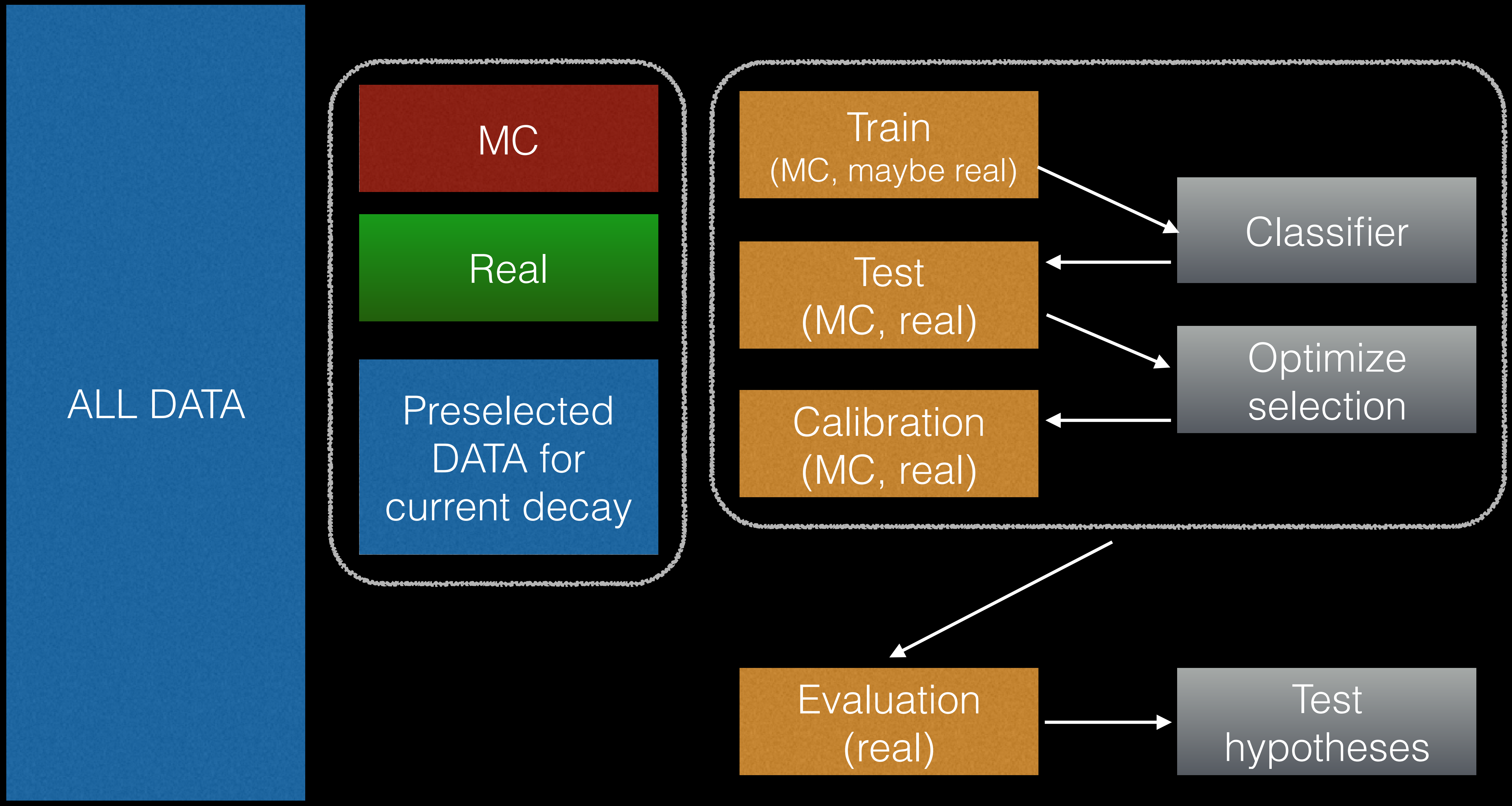
Likhomanenko Tatiana

Summer school on Machine Learning in High Energy Physics

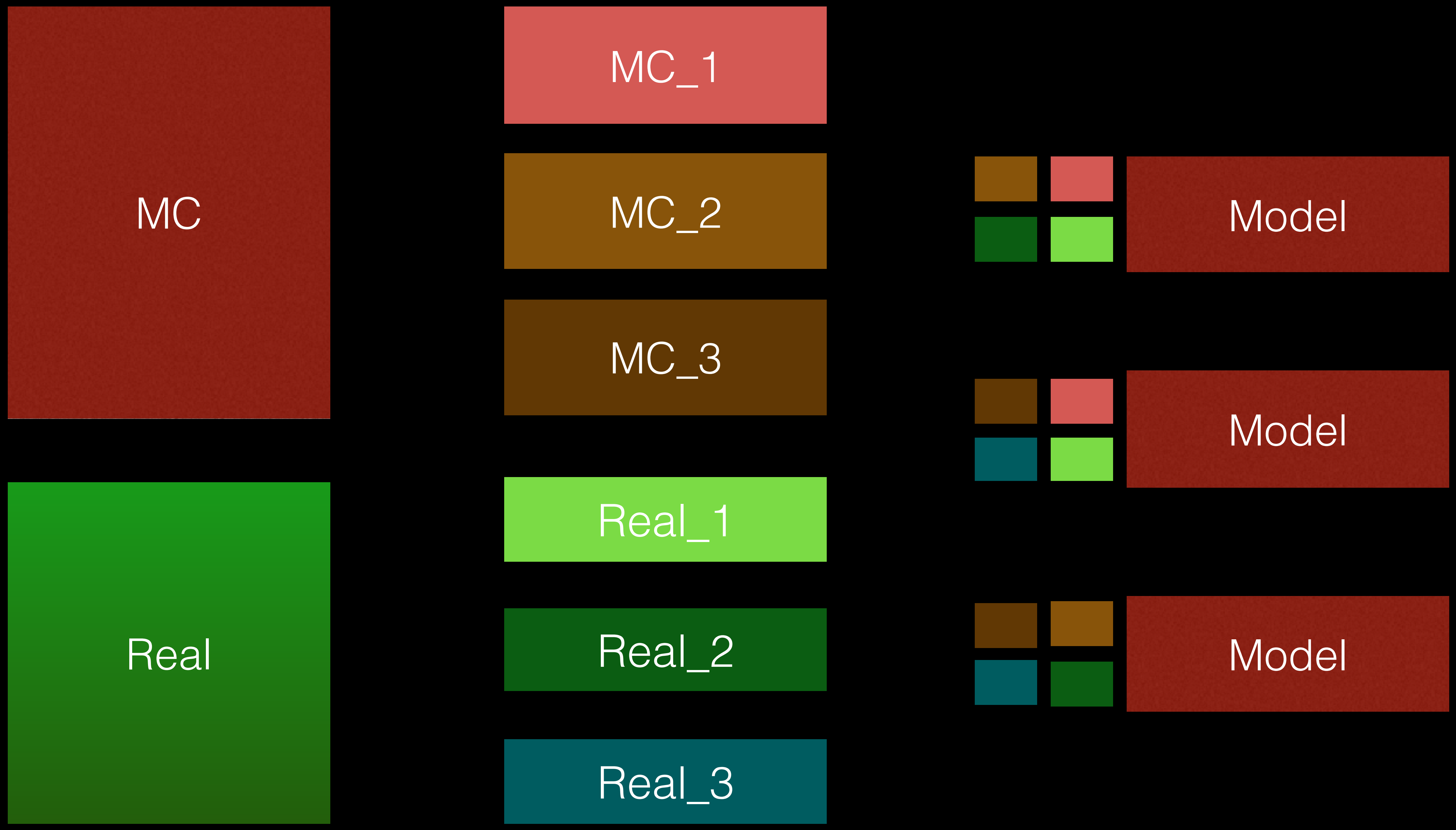
Meta Algorithms

- Ensembling (like AdaBoost over NeuralNet)
- Folding
- Stacking
- Hierarchical training
- and all combinations which you can imagine

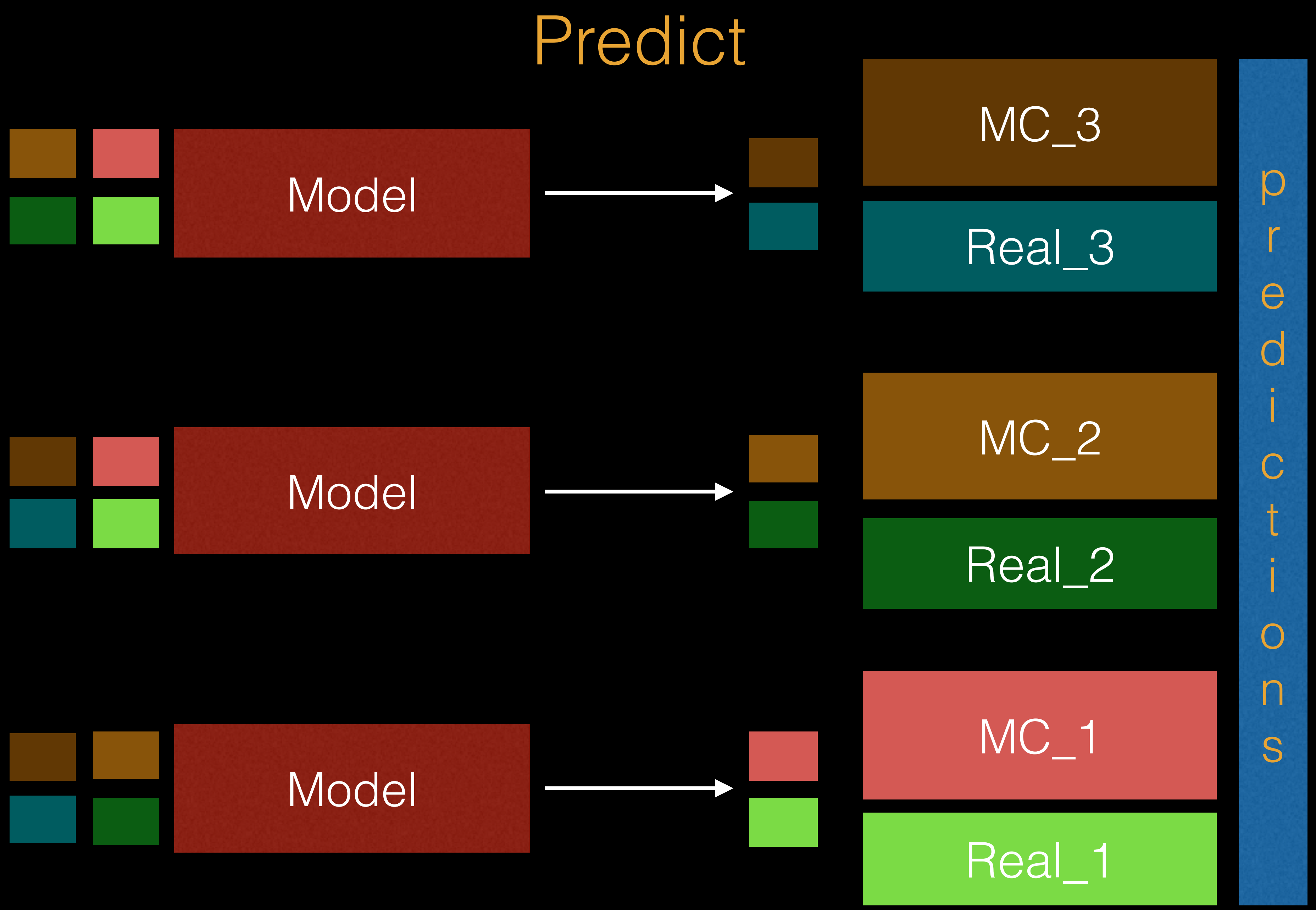
Data in analysis



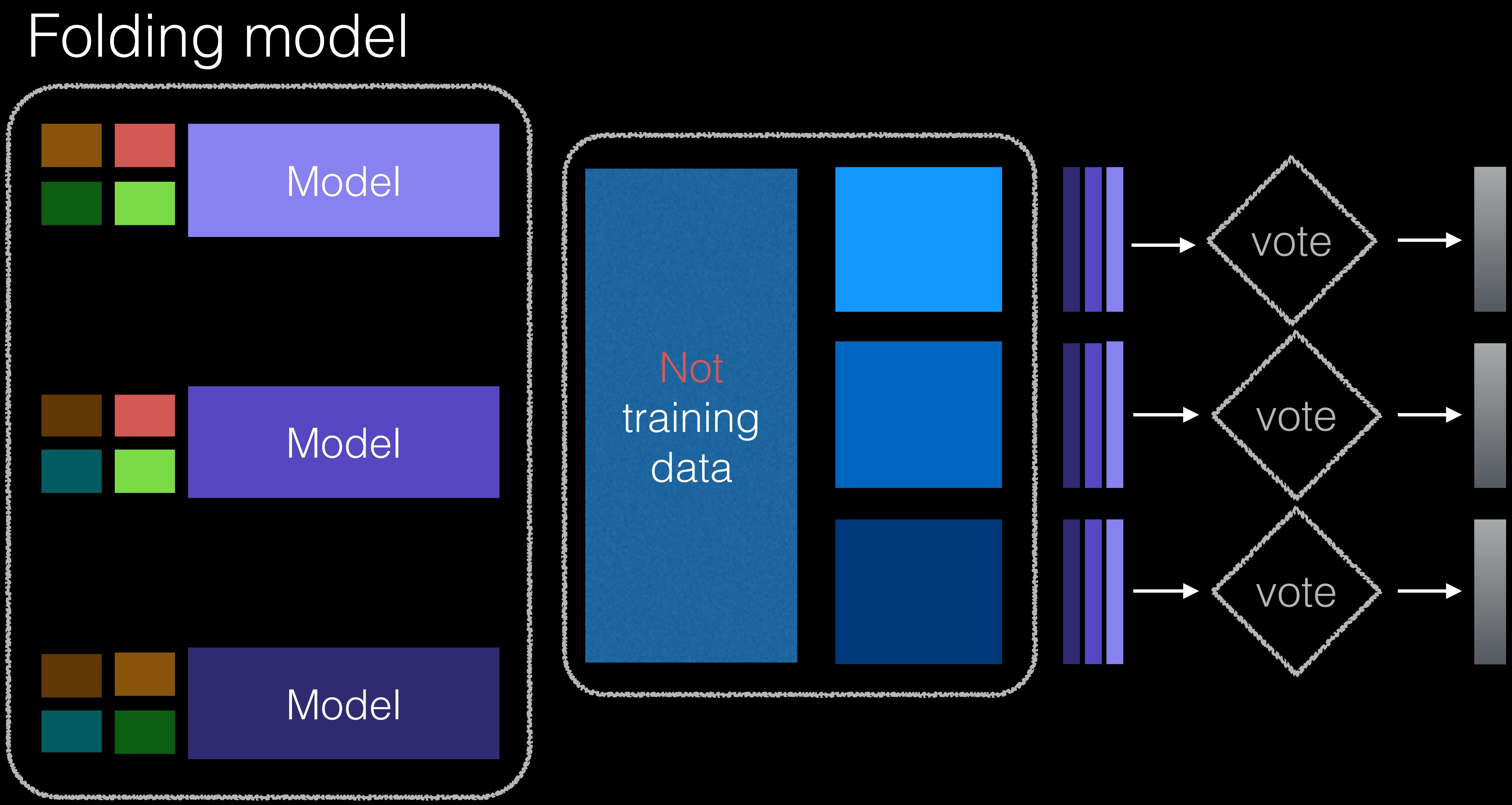
Meta algorithm: Folding Training



Meta algorithm: Folding
Predict training data



Meta algorithm: Folding
Predict another data



Meta algorithm: Folding
Vote functions

- Mean
- Max (depends on problem)
- the same principle as for training ~ take a random folding model
- linear combination
- some Regressor to correctly combine predictions (=> stacking over folding)

Often folding scheme is applied to the training sample (to mark real data on which folding was trained).

Meta algorithm: Folding

- Save real data (side bands) for analysis
- Data preselections
- Data transformations
- Feature extraction
- Hierarchical training: train folding => output of the classifier is a new feature => train another algorithm using new feature (can use all sample without removing a part of the data)
- …

REP

- unified classifiers wrapper for variety of implementations (sklearn interface)
  - TMVA
  - Sklearn
  - XGBoost
  - uBoost
  - Theanets
  - Pybrain
  - Neurolab
- parallel training of classifiers on cluster
- classification/regression reports with plots
- support for interactive plots
- grid-search algorithms with parallelized execution
- versioning of research using git
- pluggable quality metrics for classification
- meta-algorithm design (aka ‘rep-lego')
- http://yandex.github.io/rep/

Meta algorithm: Blending

Tau to three muons decay can appear from different sources:

$$\text{Promt } D_s^- \to \tau$$

$$\text{Promt } D^- \to \tau$$

$$\text{Non-promt } D_s^- \to \tau$$

$$\text{Non-promt } D^- \to \tau$$

$$X_b \to \tau$$

Can we use this information to improve our model?

Meta algorithm: Blending

- Will train each channel vs background
- Each model will define special tau source => like feature extraction (each model describe tau source)
- Use these predictions as additional features
- Check if this hierarchical training works!

Here is folding can be used to marked data by tau source models (without train1, train2, test split - only train, test split)

Meta algorithm: Random Forest

- Data with noise signal on low level data processing (trigger, tagging data)
- Monte Carlo data contains the whole event description: all tracks and SV
- Only 1 track or 1-2 SV are interesting for physics in each event.
- Event is tagged if at least 1 track / 1 SV is interesting
- Thus, training data contains many non-representative/noise signal events.
- Random Forest can be useful for this purposes (clean signal data) because of trees independency and stability to huge amount of noise!

ML problems for triggers, tagging, etc (whole event selection)

- The goal is to tag the whole event
- Event is tagged if at least one interesting track/SV exists
- Classifier is trained not on events (contains different tracks/SVs, which are not ordered), but on all tracks/SVs
- How to measure quality?

ML problems for triggers, tagging, etc (whole event selection)

- We need to assign some output for the whole event
- Often use the max of outputs of all tracks/SVs
- Now you can compute necessary metric
  - for triggers: fixed FPR (limited number of events to save)
  - for tagging: some physical parameter
- ROC curve on tracks/SVs doesn't show the really efficiency
- ROC curve for events is needed to compare classifiers!!!

Triggers and tagging predictions can be used in any decay analysis as input feature! They contain aggregation of low level information.