High confidence RTL power estimation with Machine learning

Ajay Singh Bist (ajay.b.singh@intel.com)
Mahesh Gupta Vutukuri (mahesh.g.vutukuri@intel.com)
Srinivas Rao Jammula (srinivas.r.jammula@intel.com)

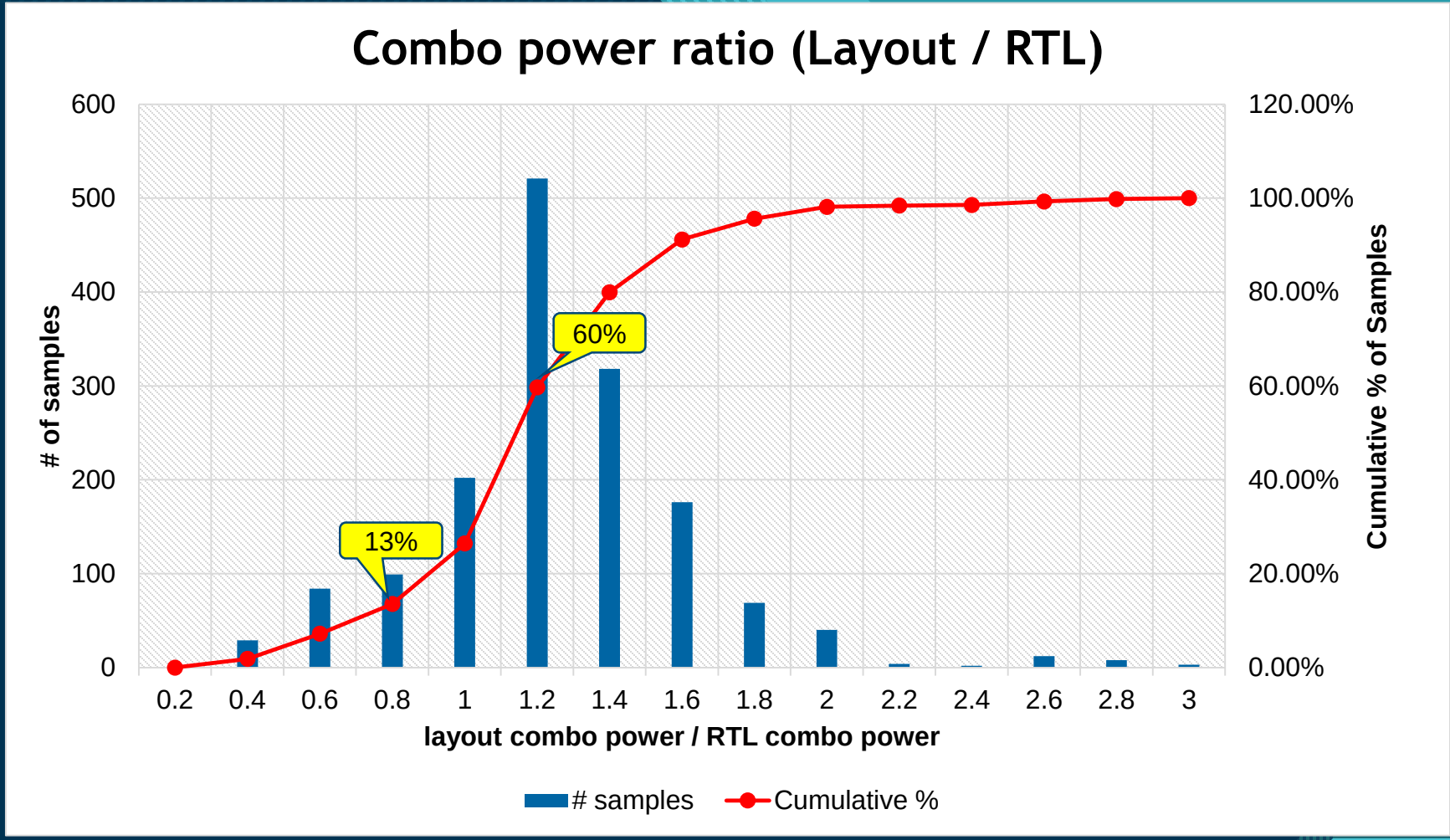
Intel Corporation

Motivation

- Major energy efficient implementation choices are made during RTL development stage
- Accurate RTL power estimation w.r.t layout based power estimation is absolutely critical for high energy efficiency
- Three major components of power – sequential, logic and clock
  - Clock power – not a significant portion in our IP of consideration
  - Sequential power – Improved correlation easily by Multi bit tuning
  - Logic power – 0.3x to 3x miscorrelation seen - Very challenging to correlate
- We explored Machine Learning (ML) Tree based algorithms and saw significant opportunity to improve the miscorrelation
- This will help us make correct implementation choices at RTL stage, to make our designs highly power optimized

Combo Power Correlation Problem

- ~13% samples have (Layout / RTL) combo power ratio less than 0.8
- ~40% samples have (Layout / RTL) combo power ratio higher than 1.2
- RTL power is smaller than Layout power for most of the samples
- Spread is on both sides. Can result in wrong assessment of RTL based power features.



RTL Power Estimation

RTL Power Estimation flow can be broken down into 3 main steps

Elaborate :

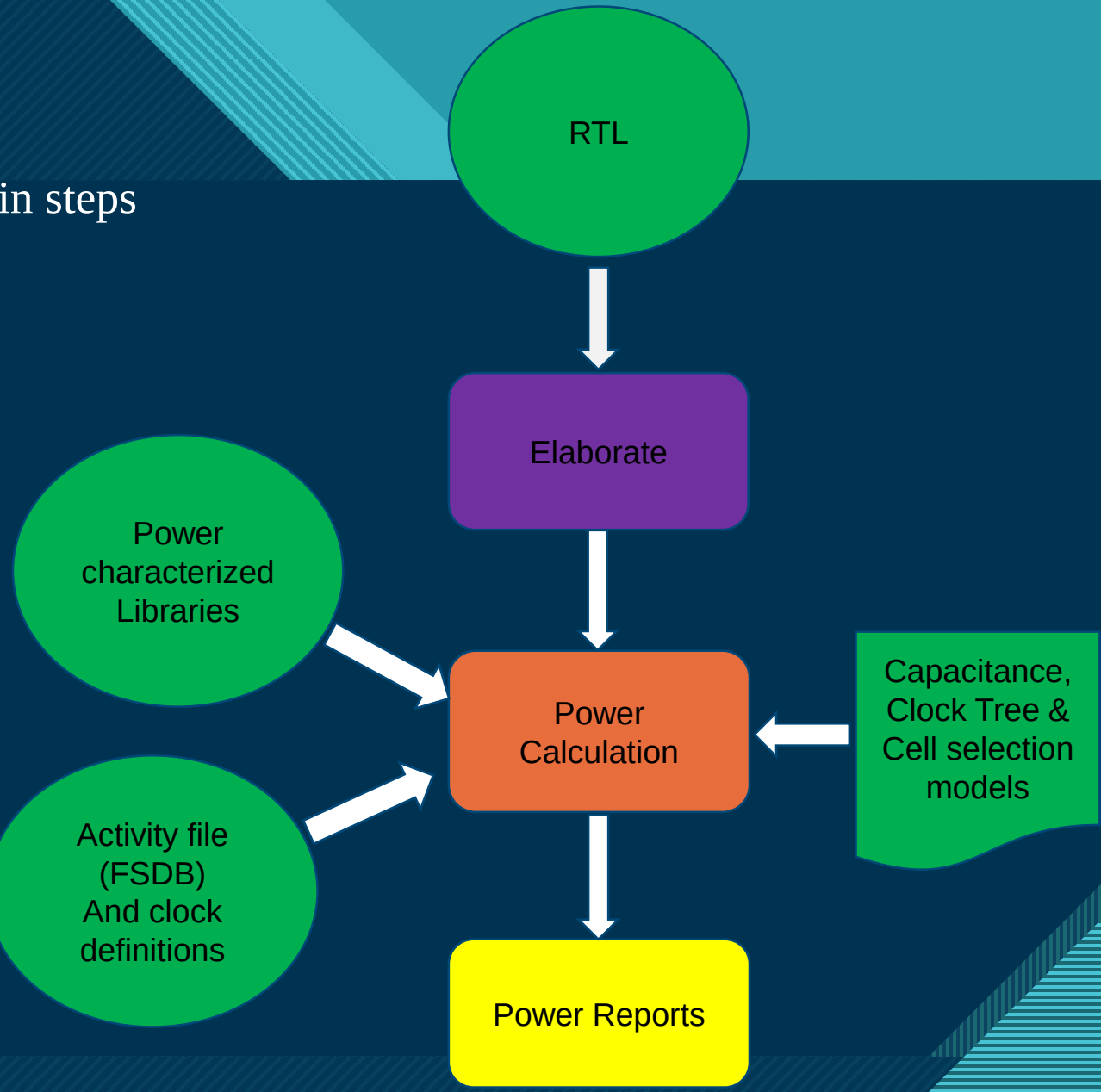
- RTL is read and compiled it into internal format

Power Calculation:

- Elaborated design is synthesized to gate level netlist
- Cell sizes, Clock tree structure and net capacitances are inferred from models derived from representative design
- Activity is annotated on the sequential from activity file and propagated to fanout logic
- Tool calculates power of design

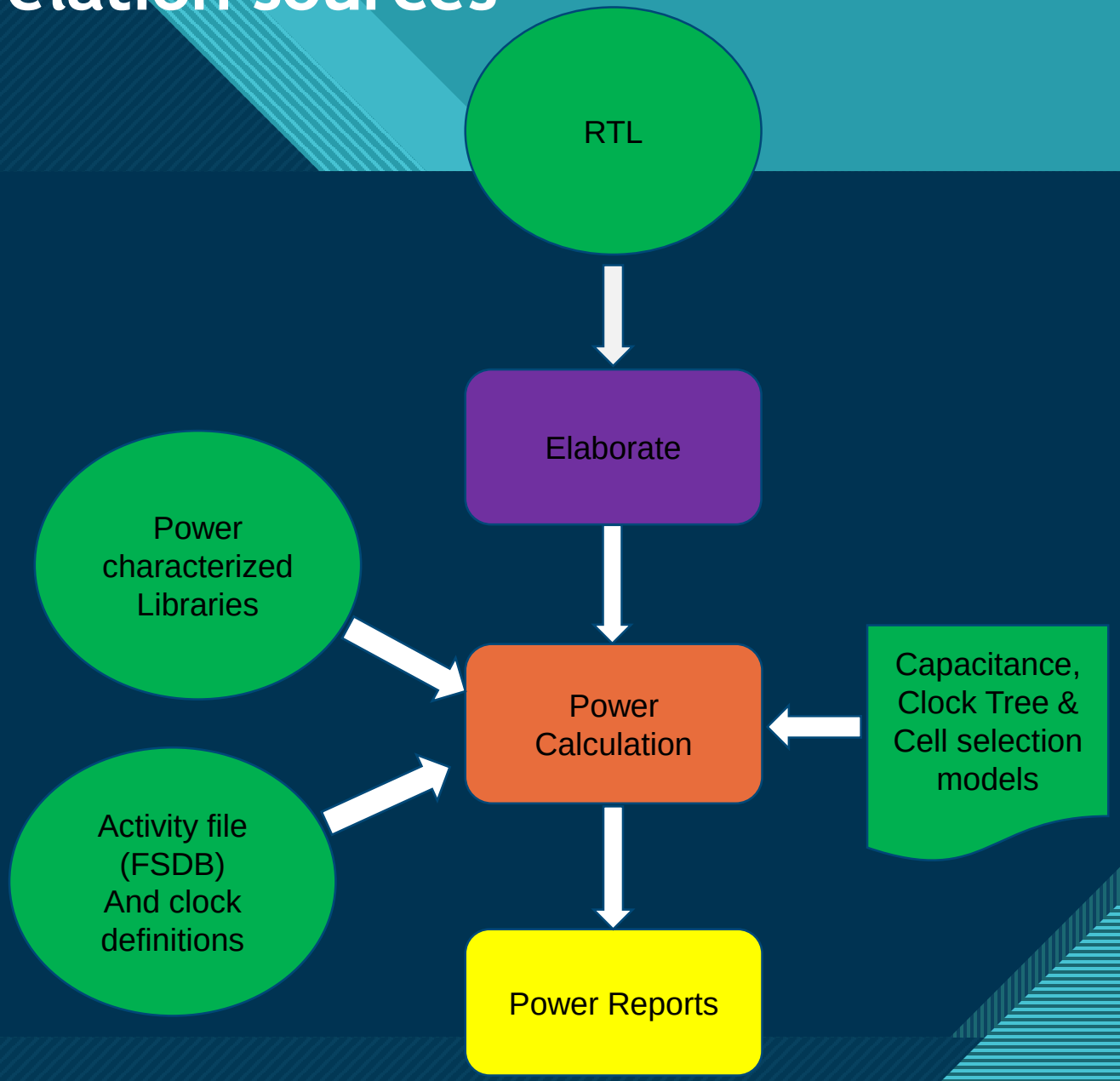
Power Reports:

- Power reports are generated for different design components at all design hierarchies

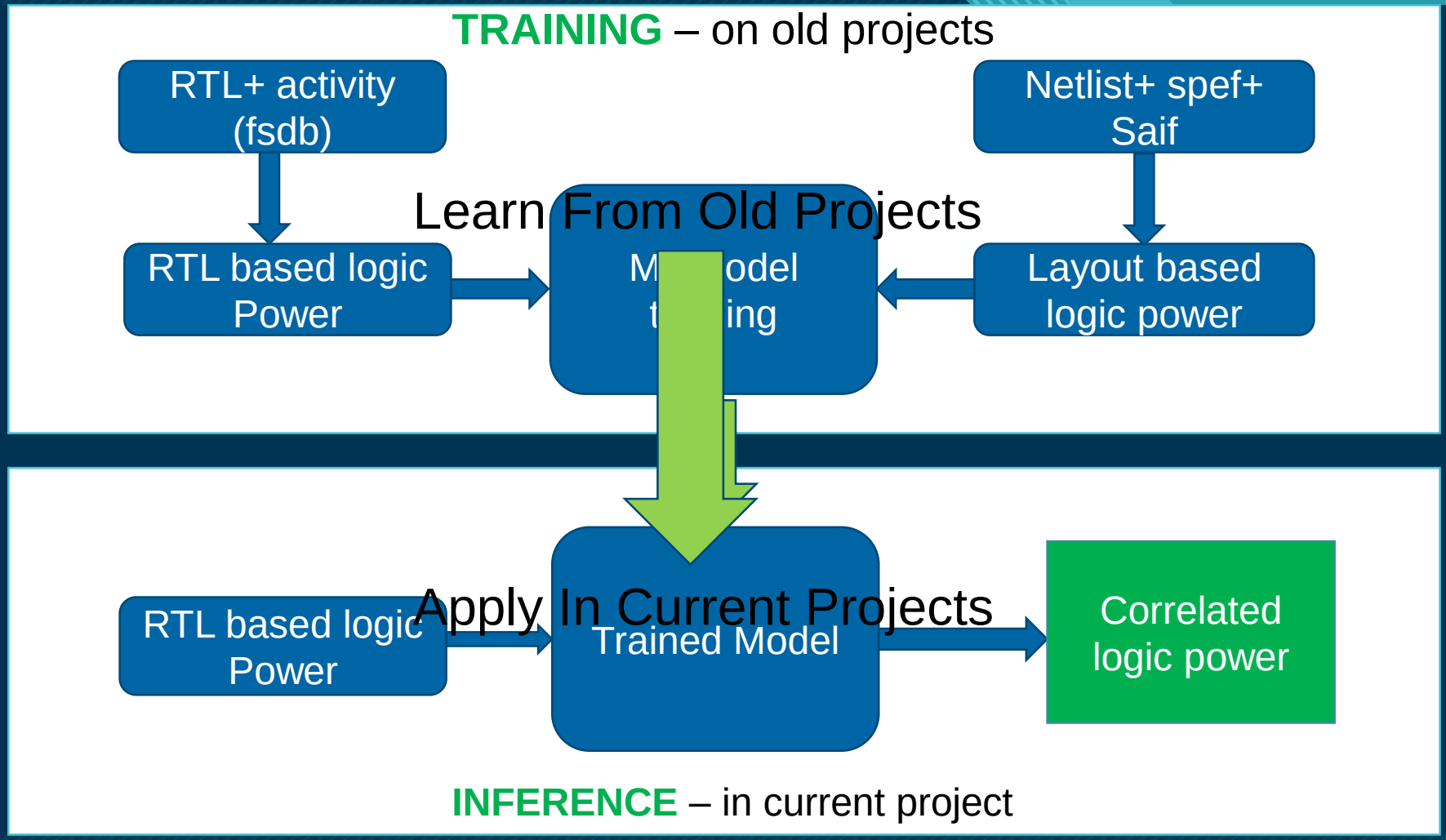


RTL Power estimation - Miscorrelation sources

- Different synthesis engine – logic implementation is different in RTL power estimation tool compared to standard physical implementation tool
- RTL power estimation tool does not use floorplan information
- The physical models (cell selection, clock tree and capacitance) used in the power calculation are lumped models



PROPOSED SOLUTION: ML to fix RTL vs Layout Power gap



ML Training : ML Model Input feature selection

Data Available in RTL Power estimation:

- Sequential power – power consumed by flops and latches
- Clock power – power consumed by clock tree
- combo power – component (multiplier, adders, inferred buffer, mux..) power breakup is available

Correlation of different component power with layout combo power:

- RTL combo components power and total logic power shows correlation with total Layout combo power

Input features used for ML model:

- Total RTL combo power
- Combo component power contribution % in total RTL combo power

ML Training: Data preparation for training

- ML model input (x) : total 19 features
  - ❖total_combo_power
  - ❖components like full_adder, mult, decoder, mux – as a % of total_combo_power
- ML model learnt output (y) : $\frac{(combo\ Power_{Layout\ based})}{(combo\ Power_{RTL\ based})}$, ranging from 0.2 ->3
- We need minimum of ~100x observations for each input feature to train ML algorithm, which is very challenging
  - ❖Multiple directed power tests
  - ❖multiple time windows – with varying levels of activity residencies
  - ❖Split the available data to train and test data sets

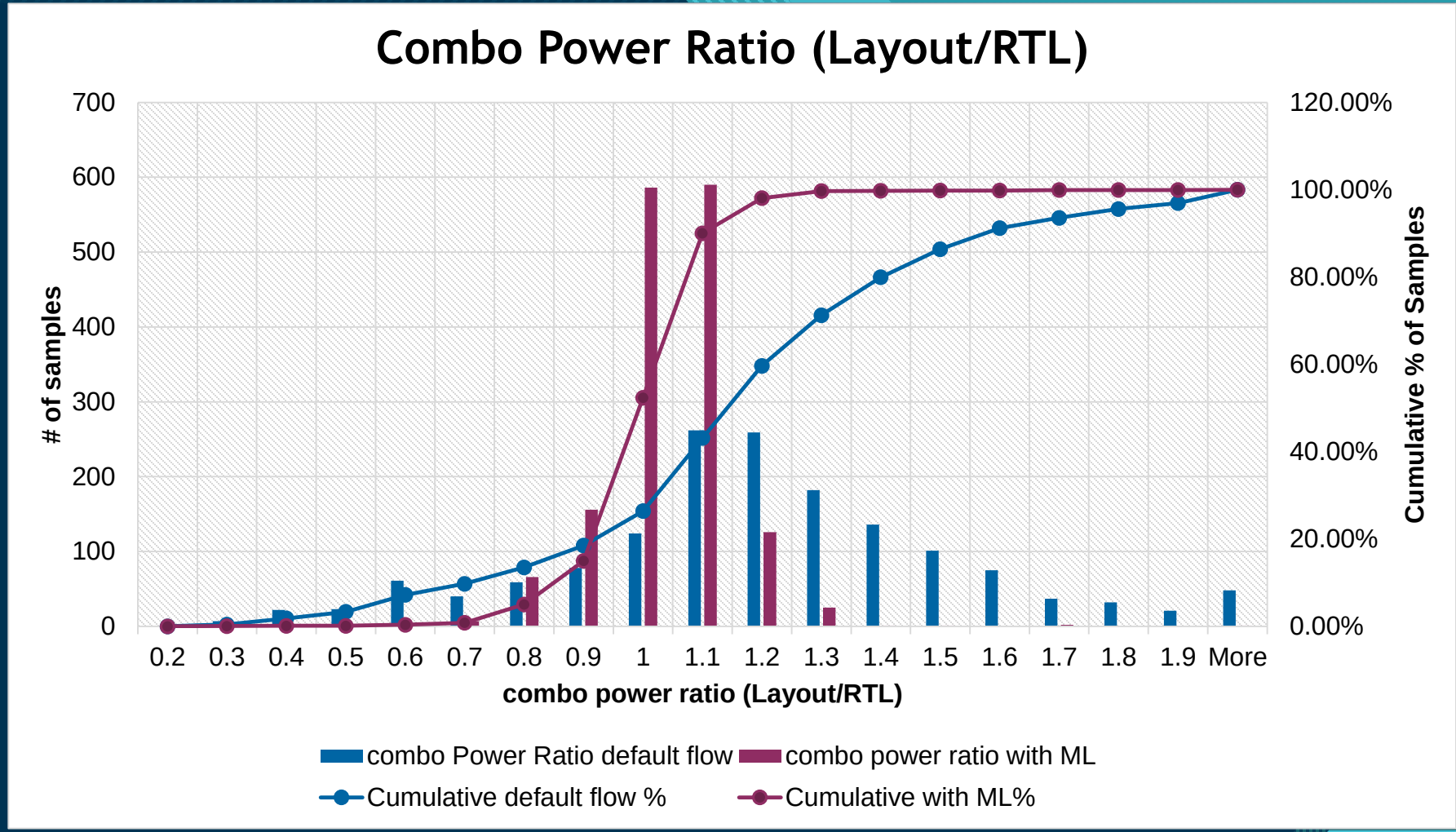
ML Training: Training & Cross-validation

- **ML Model Training & Cross Validation**:
  - ❖Non Parametric Tree based Ensemble class of regression algorithms* like Random Forest, (X/L)Gradient Boosting
  - ❖Grid search based group kfold Cross Validation for hyper parameter tuning
    - ❖(n_estimators, max_tree_depth, max_features.. etc)
    - ❖Best Model of CV is evaluated based on performance metrics (ex: MAE), before finalizing model for deployment
- Eventual Model Selection – Gradient Boosting Regression with tuned hyper parameters
- Train GBR Model with full training data set

*open source python based Scikit learn library is used for algorithms

Inference Results

- 93% samples within ±20% error with ML compared to only 47% in default flow
- Significant improvement in miscorrelation **HUGE WIN !!!**
- Enabled in Default Power estimation flow



Summary & Next steps

- **Summary**
  - Delivering energy efficient hardware accelerator is of paramount importance
  - Our design power optimization starts from RTL explorations continuing all the way to Netlist, layout, post Silicon optimizations
  - Major power opportunity exists in making correct RTL implementation choices, unfortunately huge miscorrelation existed to layout based power
  - We leveraged Machine Learning algorithms to improve miscorrelation significantly, enabling energy efficient RTL design
- **Next steps**
  - Extend the study to correlate power from logic synthesis to layout for retimed designs
  - Address miscorrelation issues on IPs where clock power is of significance