

Machine Learning for *Smartpixel* Detectors: Reproducing and Extending the ASIC Inference Pipeline

Corrinne Mills, Danush Shekar, [Sarah Waldych](#), *smartpixels* team

What are *Smartpixels*?

- *Smartpixels* enable real-time filtering of p_T^{high} particles from p_T^{low} particles before readout, using trained machine-learning models deployed on an ASIC
- High volumes of data in modern detectors require extreme readout times
- Most high energy physics searches are centered around heavy particles and rare decays which are produced from p_T^{high} particles
- p_T^{low} particles are abundant background

How do *Smartpixels* Work?

- Pixel measures charge deposited from particle
- Front end electronics amplify analog charge
- Analog charge digitized by a 2-bit ADC operating synchronously at 40 MHz (25 ns a cycle)
- Digitized pixel charges are summed across columns to make a y-profile

$$y_{profile} = (q_1, q_2, q_3, \dots, q_{13})$$

- $y_{profile}$ and global location of hit are inputs to filtering algorithm which filters tracks from particles with p_T^{low} from p_T^{high}

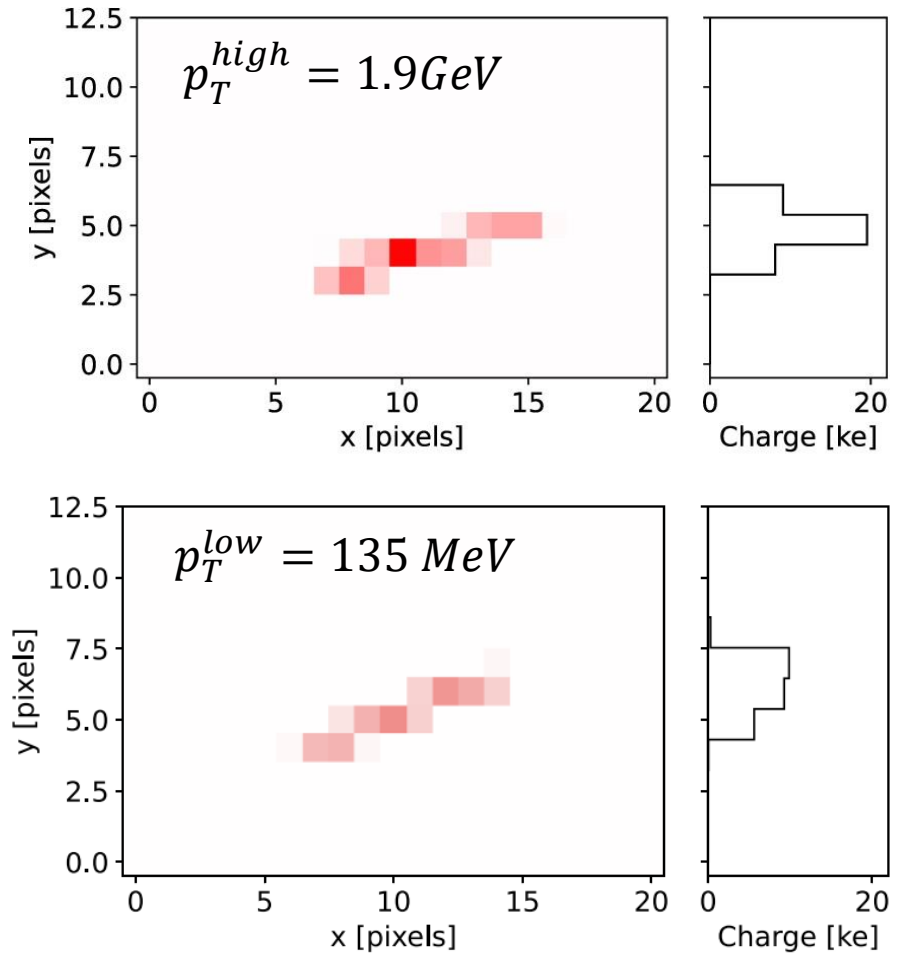
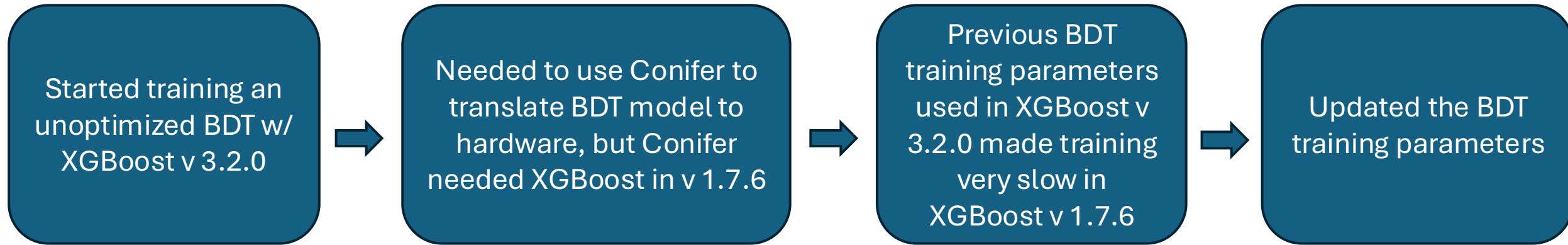


Figure: Visualization of summing columns to make $y_{profile}$

Recent Updates



Improved BDT training parameters



Previous training parameters

```
"objective": "multi:softprob",
"num_class": 3,
"max_depth": 5,
"eta": 0.05,
"subsample": 0.8,
"colsample_bytree": 0.8,
"min_child_weight": 1,
"lambda": 1.0,
"alpha": 0.0,
"seed": seed,
"eval_metric": ["mlogloss", "merror"]
```

```
X_bdt_train_old, X_bdt_val_old, y_bdt_train_old, y_bdt_val_old = train_test_split(
    X_train,
    y_train,
    test_size=0.2,
    random_state=seed,
    shuffle=True,
    stratify=y_train
)
```

```
dtrain_bdt_old = xgb.DMatrix(X_bdt_train_old, label=y_bdt_train_old)
dval_bdt_old = xgb.DMatrix(X_bdt_val_old, label=y_bdt_val_old)
dtest_bdt_old = xgb.DMatrix(X_test, label=y_test)
evals_result_bdt_old = {}
model_bdt_old = xgb.train(
    params_old,
    dtrain_bdt_old,
    num_boost_round=200,
    evals=[(dtrain_bdt_old, "train"), (dval_bdt_old, "val")],
    evals_result=evals_result_bdt_old,
    early_stopping_rounds=10, verbose_eval=0
)
```

Updated training parameters

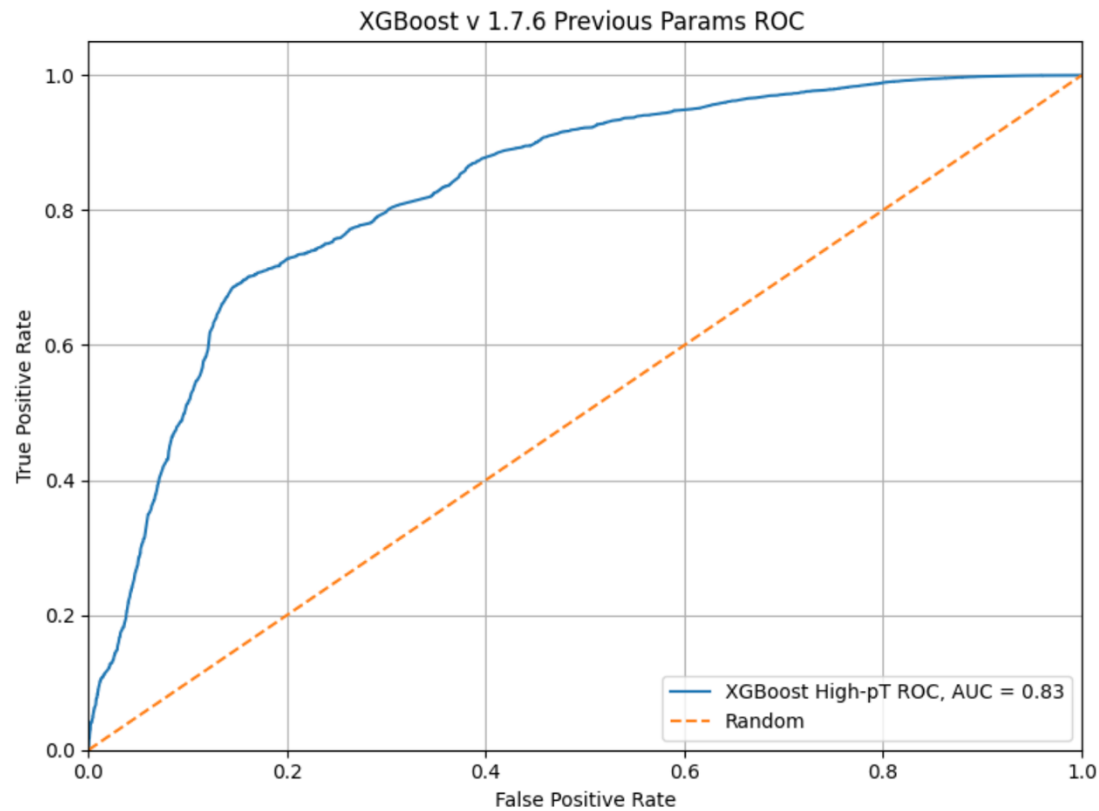
```
"objective": "multi:softprob",
"num_class": 3,
"max_depth": 5,
"eta": 1,
"subsample": 0.8,
"colsample_bytree": 0.8,
"min_child_weight": 1,
"lambda": 1.0,
"alpha": 0.0,
"seed": seed,
"eval_metric": ["mlogloss", "merror"],
"tree_method": "hist"
```

```
X_bdt_train, X_bdt_val, y_bdt_train, y_bdt_val = train_test_split(
    X_train,
    y_train,
    test_size=0.2,
    random_state=seed,
    shuffle=True,
    stratify=y_train
)
```

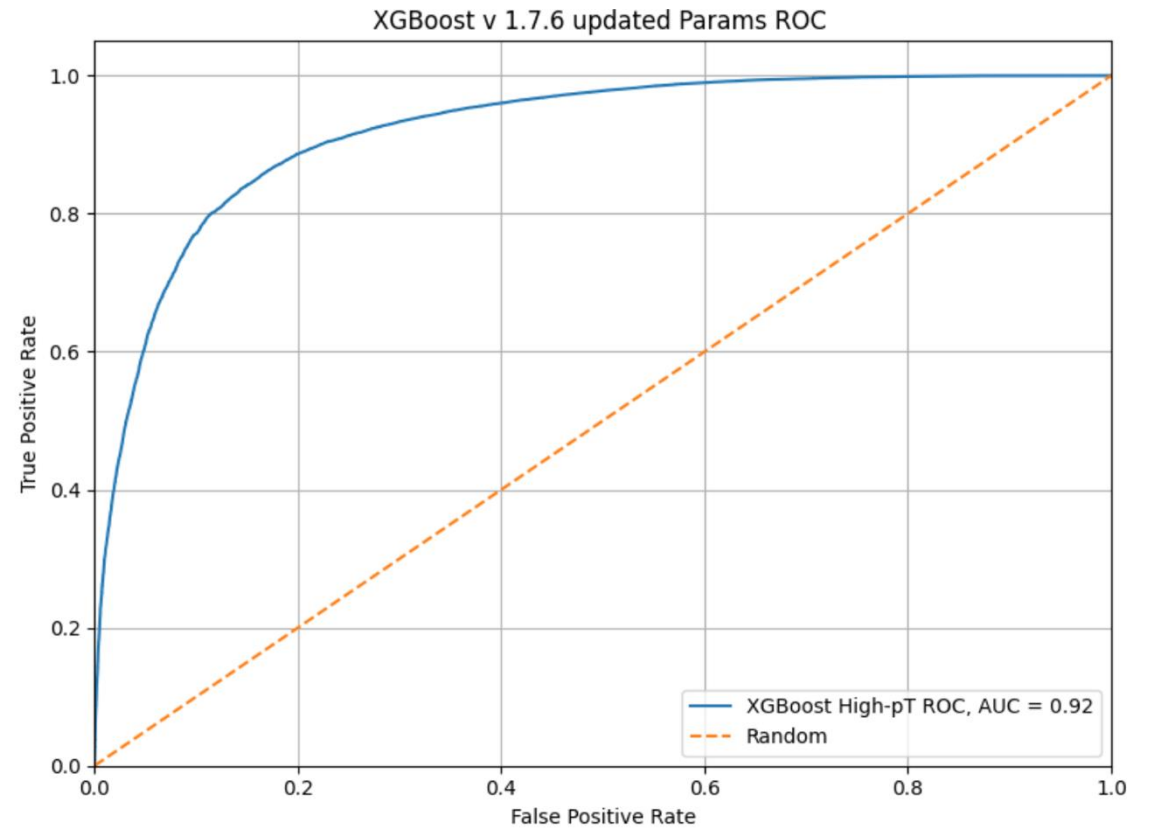
```
dtrain_bdt = xgb.DMatrix(X_bdt_train, label=y_bdt_train)
dval_bdt = xgb.DMatrix(X_bdt_val, label=y_bdt_val)
dtest_bdt = xgb.DMatrix(X_test, label=y_test)
evals_result_bdt = {}
model_bdt = xgb.train(
    params,
    dtrain_bdt,
    num_boost_round=25,
    evals=[(dtrain_bdt, "train"), (dval_bdt, "val")],
    evals_result=evals_result_bdt,
    early_stopping_rounds=10,
    verbose_eval=0
)
```

ROC between models XGBoost v 1.7.6

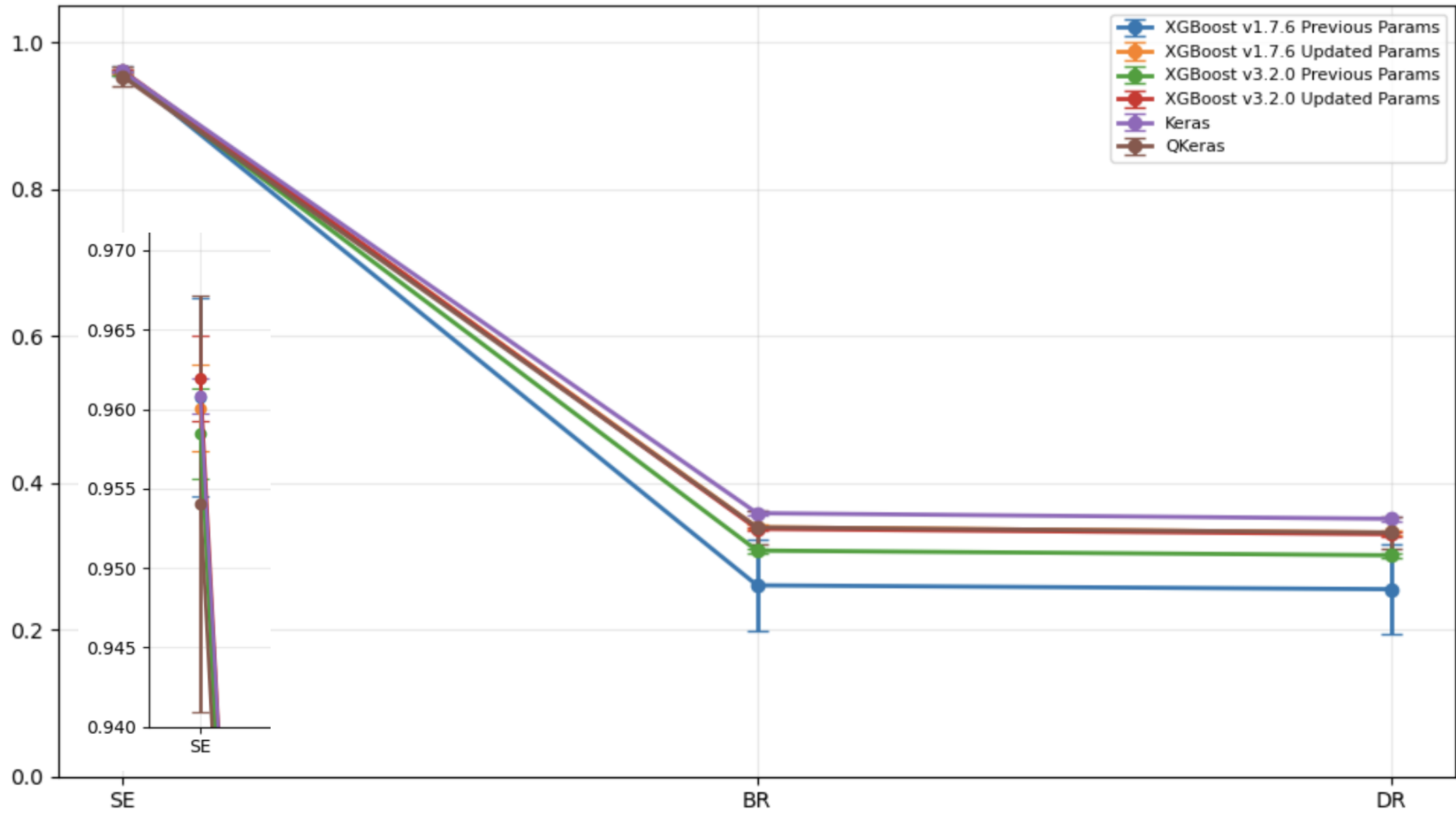
Previous training parameters

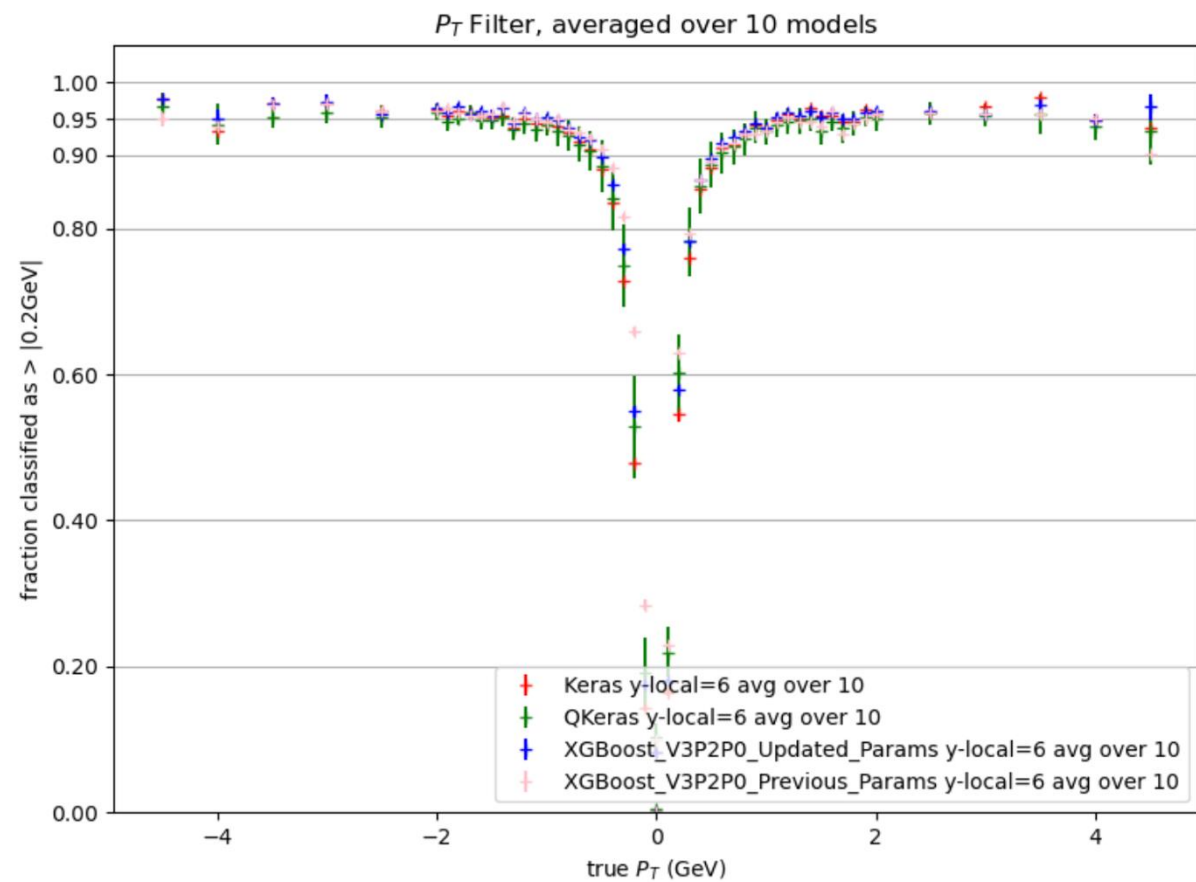
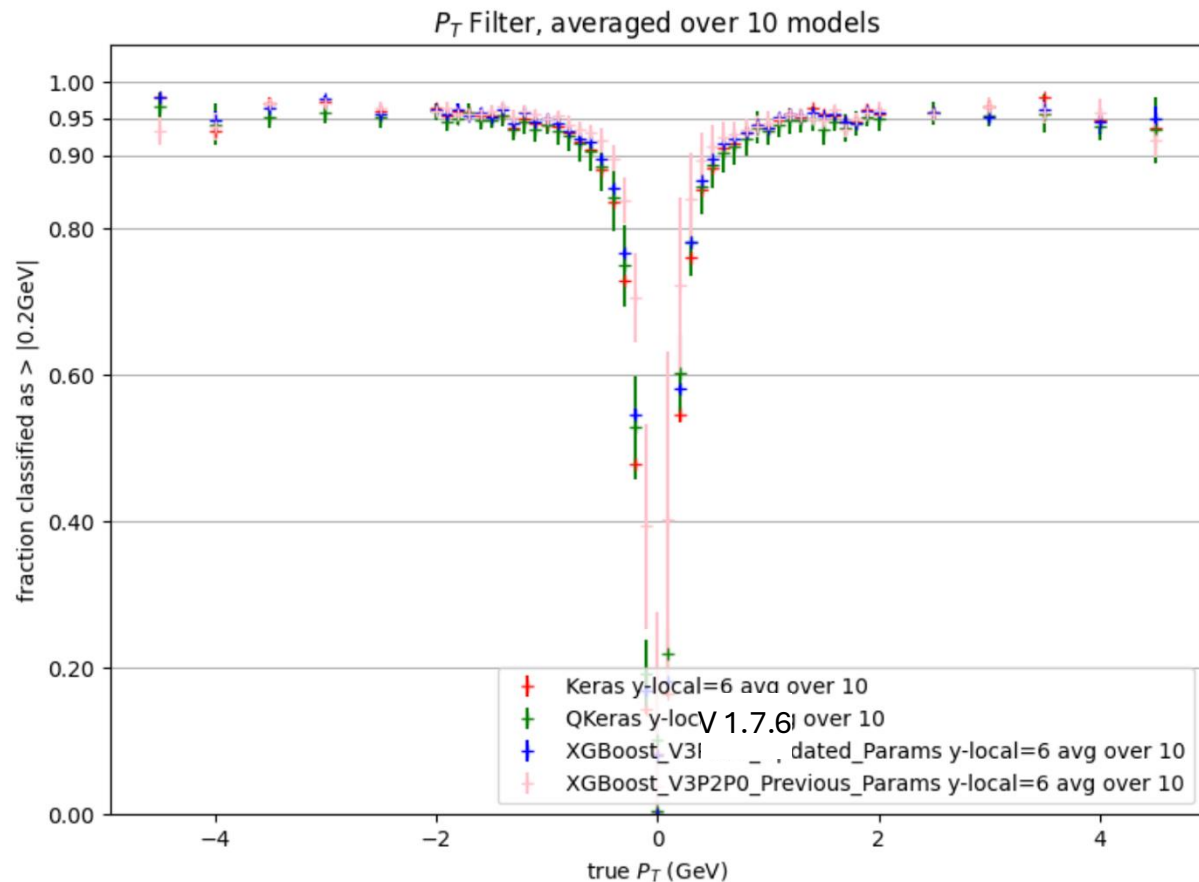


Updated training parameters



SE, BR, and DR Comparison





The previous training parameters in XGBoost v 1.7.6 causes large fluctuations compared to XGBoost v 3.2.0



Hardware translation of ML Model on XC7A100T FPGA

Qkeras NN d58w4a8 (no scaling)

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	32492	0	63400	51.25
LUT as Logic	32492	0	63400	51.25
LUT as Memory	0	0	19000	0.00
Slice Registers	8148	0	126800	6.43
Register as Flip Flop	8148	0	126800	6.43
Register as Latch	0	0	126800	0.00
F7 Muxes	0	0	31700	0.00
F8 Muxes	0	0	15850	0.00

Output according to hls4ml (Vivado v.2020.1)

XGBoost BDT

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	7579	0	0	63400	11.95
LUT as Logic	7499	0	0	63400	11.83
LUT as Memory	80	0	0	19000	0.42
LUT as Distributed RAM	0	0			
LUT as Shift Register	80	0			
Slice Registers	4309	0	0	126800	3.40
Register as Flip Flop	4309	0	0	126800	3.40
Register as Latch	0	0	0	126800	0.00
F7 Muxes	52	0	0	31700	0.16
F8 Muxes	4	0	0	15850	0.03

Output according to Conifer (Vivado v.2023.1)



Timing and Latency

Qkeras

+ Timing:

* Summary:

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	8.518 ns	1.25 ns

+ Latency:

* Summary:

Latency (cycles)		Latency (absolute)		Interval		Pipeline
min	max	min	max	min	max	Type
612	612	6.120 us	6.120 us	612	612	none

XGBoost BDT

+ Timing:

* Summary:

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	7.250 ns	2.70 ns

+ Latency:

* Summary:

Latency (cycles)		Latency (absolute)		Interval		Pipeline
min	max	min	max	min	max	Type
7	7	70.000 ns	70.000 ns	1	1	yes

Summary

- *Smartpixels* enable real-time ML-based filtering directly on the ASIC
- BDT trained from XGBoost has undergone initial hardware translation steps via Conifer
- BDT requires significantly less resources on the testing FPGA compared to the NN