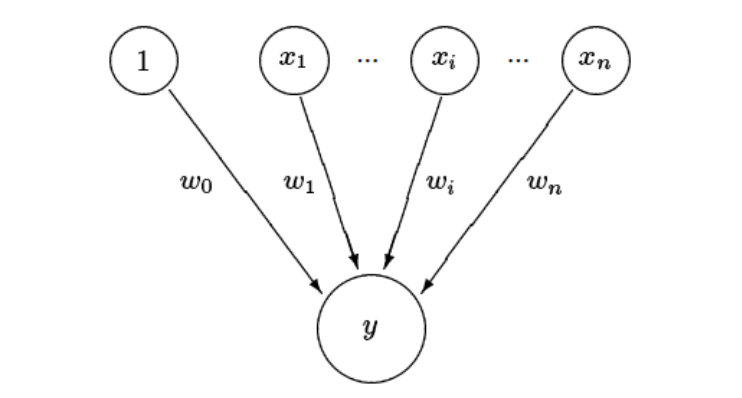
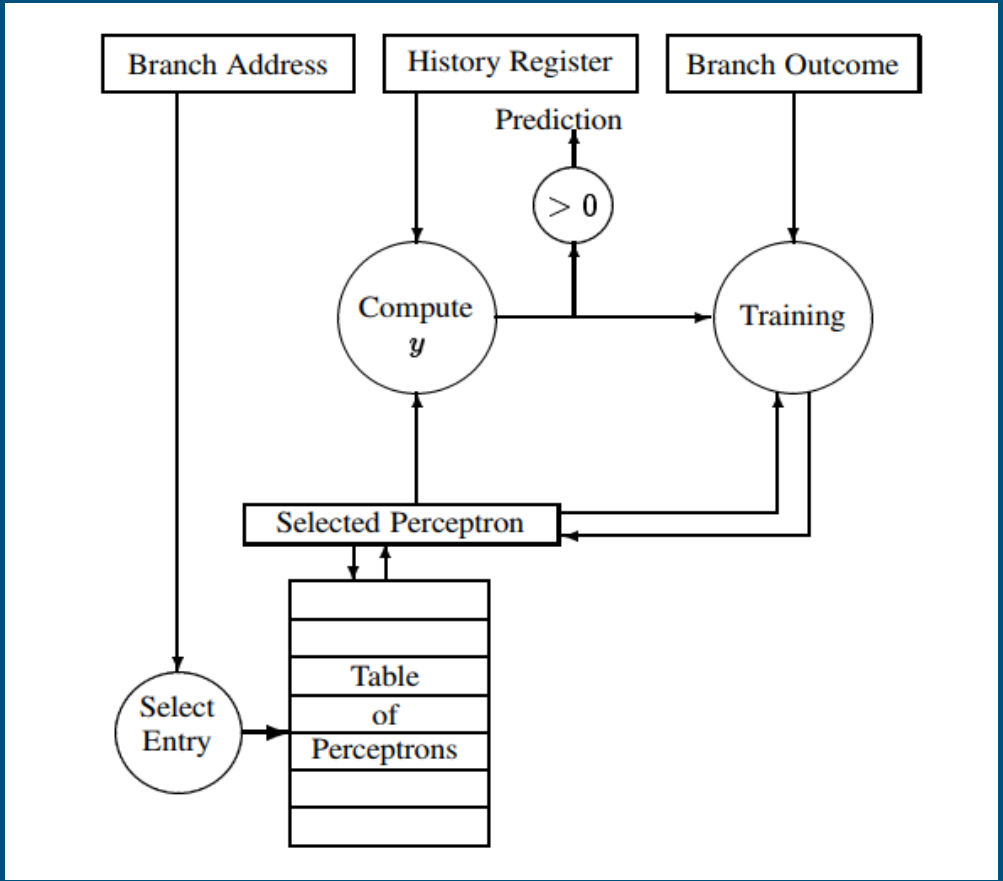
Neural Branch Prediction - gem5 simulation

Ravi Raju
N Vijaytarang
Ayush Gupta

Base neural predictor (global perceptron)





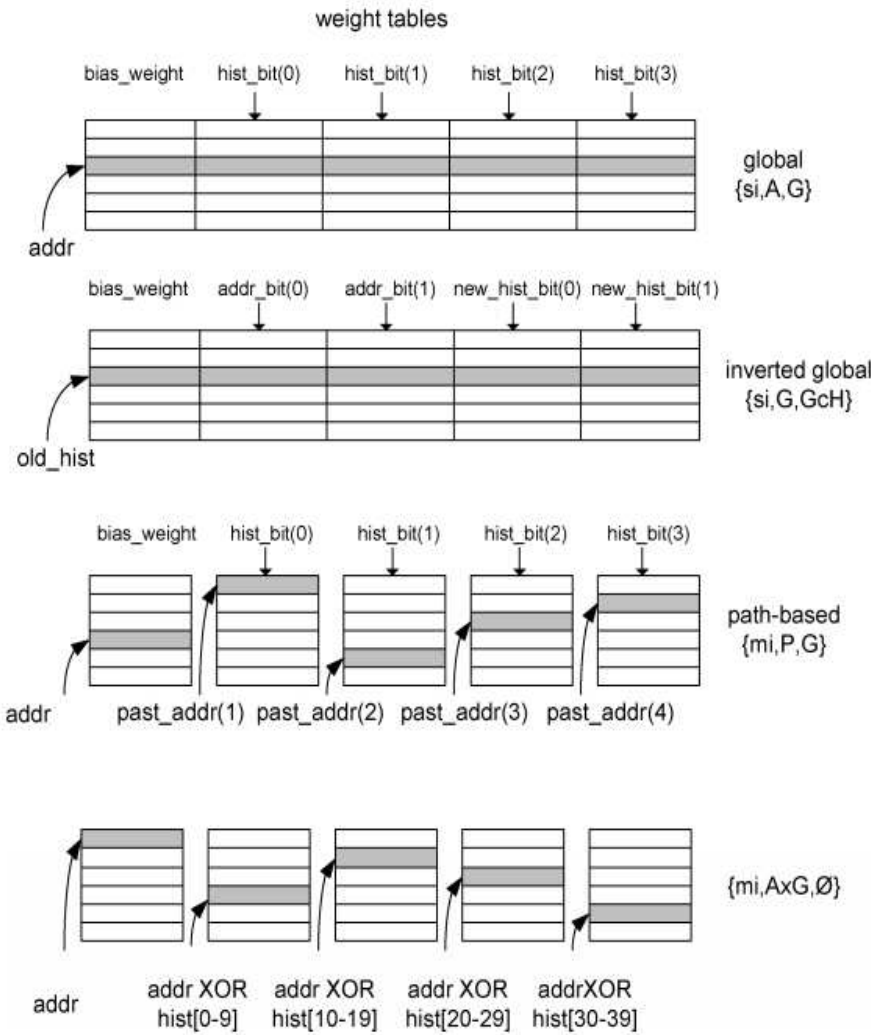
Hashed Perceptron

Multiple set of branches assigned to same weights

More than one set of information can be used to hash into the weight table

Different hashing function can be used such as : XOR, Concatenation etc.

Different indices can be assigned to different weights



Hashed Perceptron Advantages

The one-to-one correlation of weights to number of history bits in a perceptron is not necessary

Reduced number of adders

Allows use of longer history

Combining multiple ways to index weights in a single perceptron improves accuracy over using only a single way.

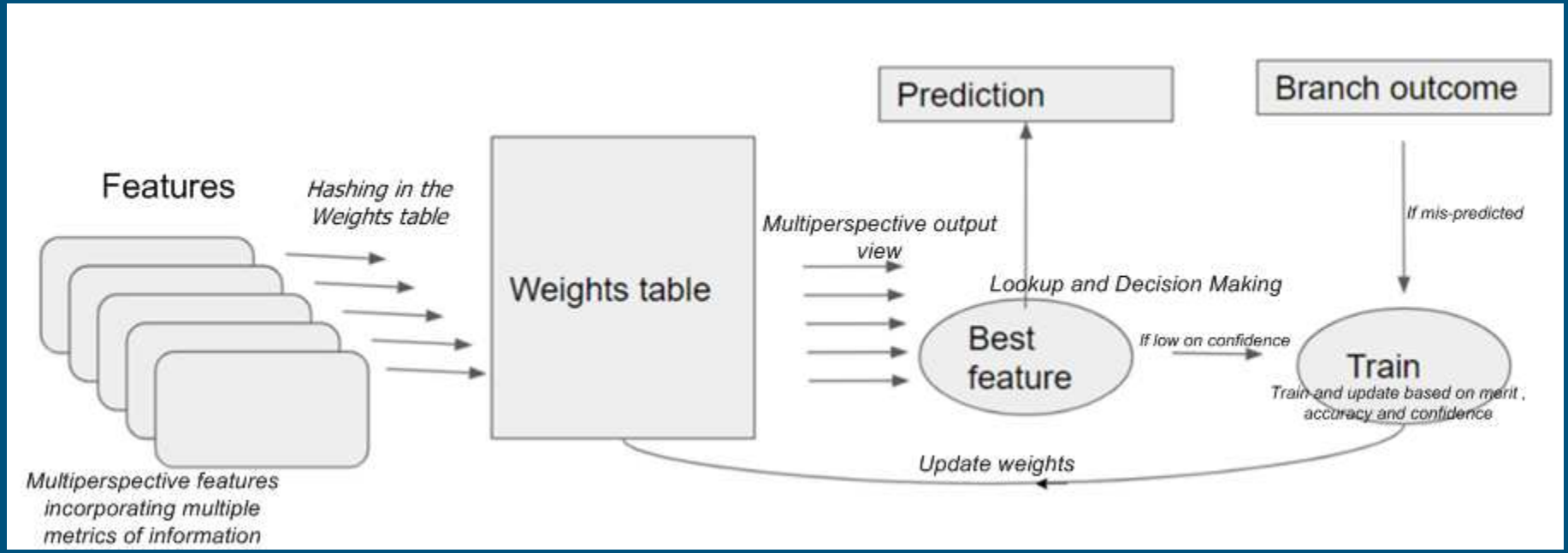
Multiperspective Predictor

Viewing branch history and path from multiple perspectives

A hashed perceptron predictor that uses not only hashed global path and pattern histories, but also variety of other kinds of features based on various organizations of branch histories and other metrics

All such weights for chosen through different features are used to make a prediction of taken or not taken

Multiperspective Predictor



Feature selection

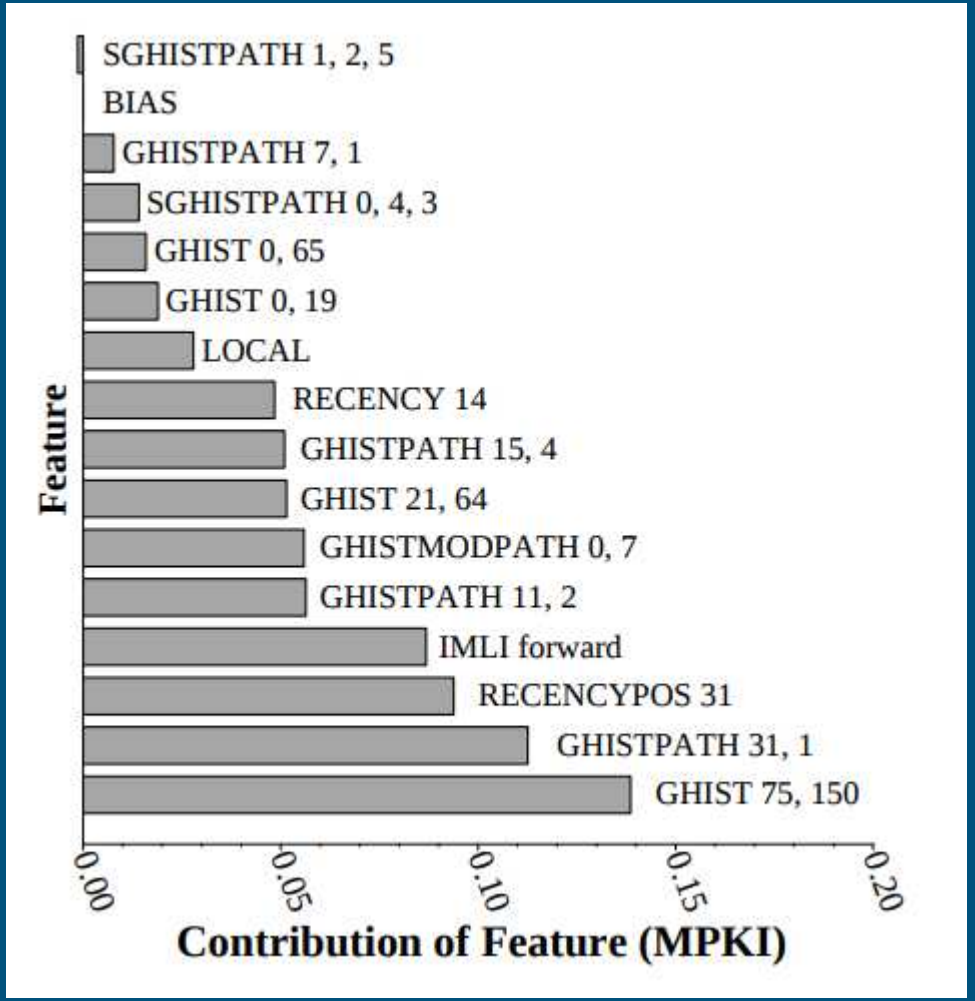
PC

PATH

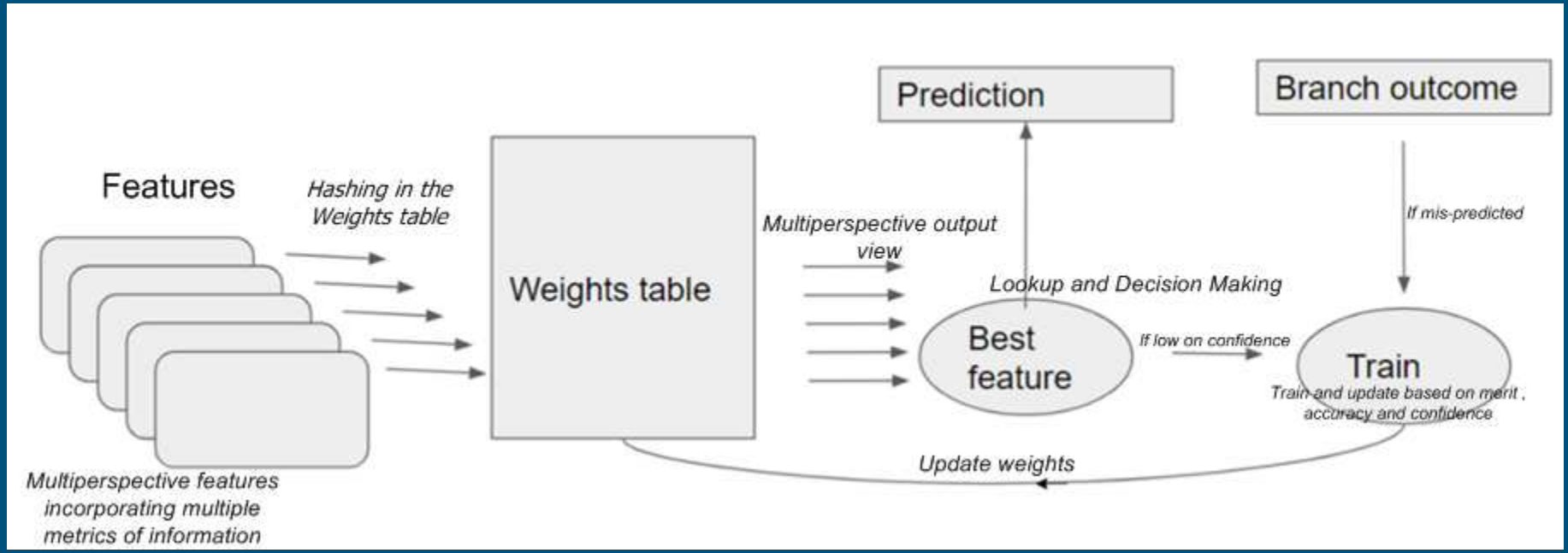
GHISTPATH

RECENCY

RECENCYPOS



Lookup & Update



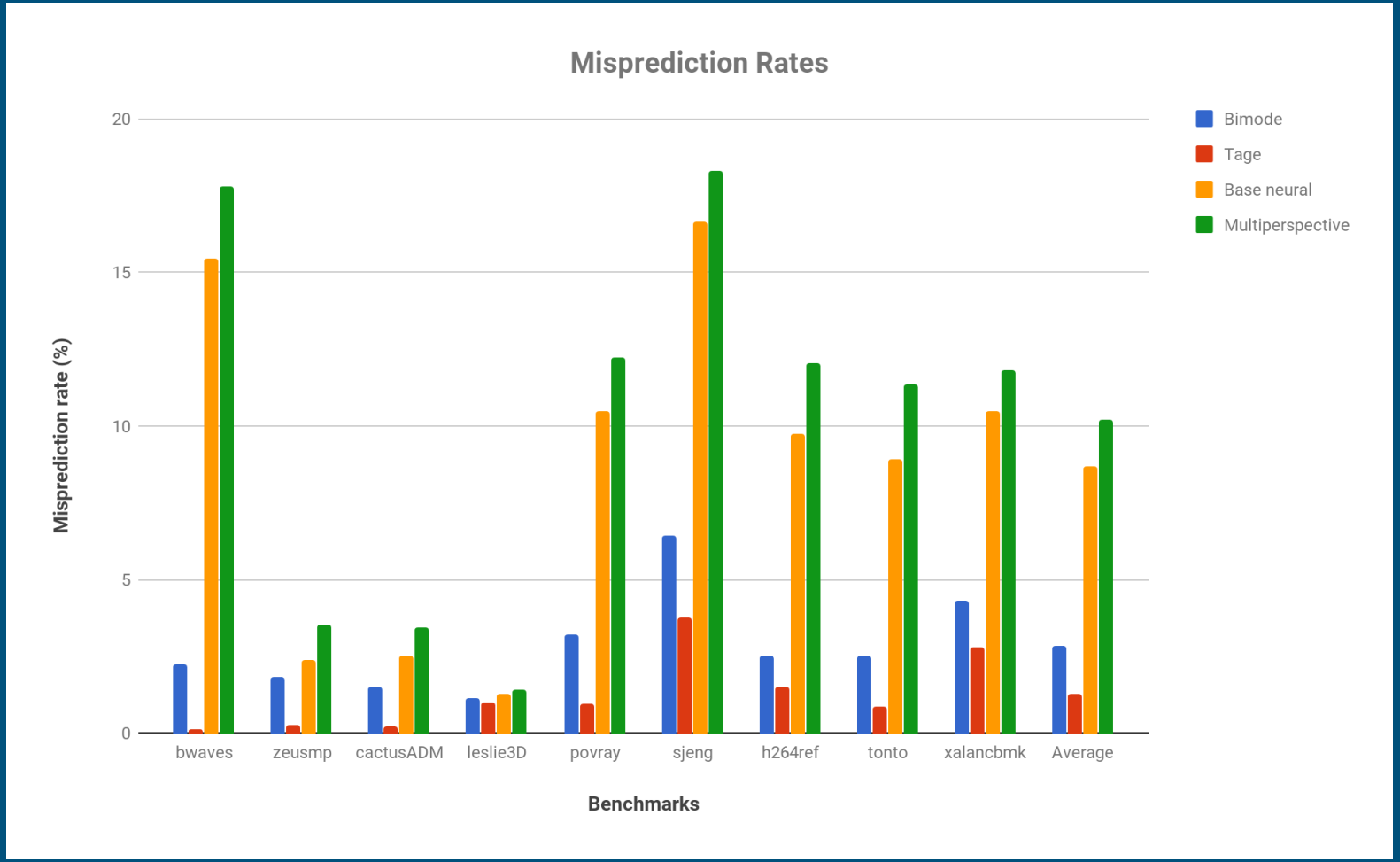
Experimental Setup

Ran gem5 ARM 2006 SPEC benchmarks

Fast-forward for 1 billion instructions with AtomicSimpleCPU

Swapped for O3-ARM core with updated branch predictor for another billion instructions

Results


Misprediction Rates
Misprediction rate (%)
20
15
10
5
0
bwaves
zeusmp
cactusADM
leslie3D
povray
sjeng
h264ref
tonto
xalancbmk
Average
Benchmarks
Bimode
Tage
Base neural
Multiperspective

Results

Lower number of features

Multiple hardware structures missing

Lookup function requires optimization

Future work/Conclusion

Add more features into multiperspective lookup

Better selection of best features to make final prediction

Training function could be better optimized

Adaptive theta thresholding could be implemented

Simpoints could be created to better represent the benchmark

Citations

David Tarjan and Kevin Skadron. Merging Path and Gshare Indexing in Perceptron Branch Prediction. CS 2004-38, University of Virginia, December 2004.

Daniel A. Jimenez. Multiperspective Perceptron Predictor. CBP2016. Texas A&M University. 2016

JIMENEZ , D. AND LIN, C. 2001. Dynamic branch prediction with perceptrons. In Proceedings of The Seventh International Symposium on High-Performance Computer Architecture. 197–206.

Patel, Yash. Neuropath. 2017. Github repository, https://github.com/yashpatel5400/neuropath/tree/master/predictor

Thank you and questions?