Temperature Prediction in Monolithic 3D Using Machine Learning Based Models

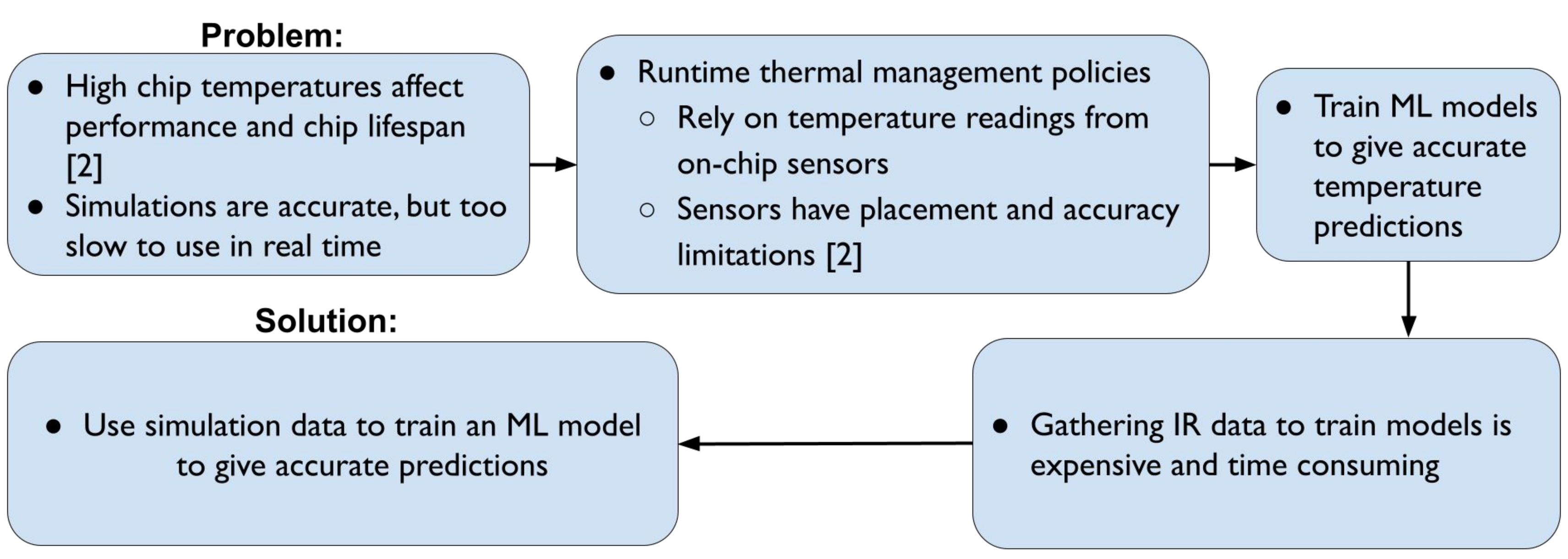
Alan Zhou[1,2], Amin Khodaverdian[2], Prof. Ayse K. Coskun[2]

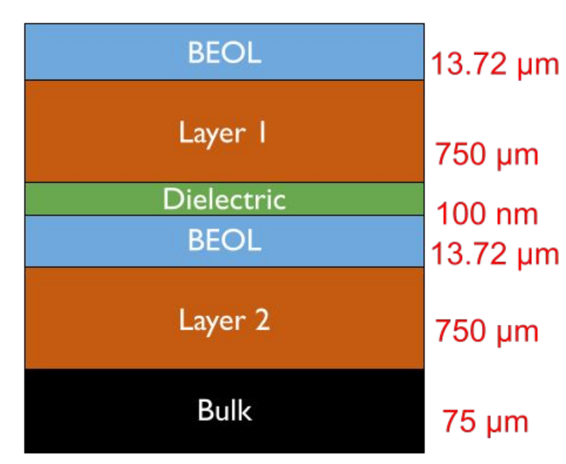
Episcopal Academy, 1785 Bishop White Drive, Newtown Square, PA 19073[1],
Department of Electrical & Computer Engineering, Boston University, 8 St. Mary's Street, Boston, MA 02215[2]

Introduction

Background:

- Monolithic 3D (M3D) integration is an emerging technology enabling the stacking of multiple transistor, or active, layers (also called tiers) within one Integrated Circuit (IC) [1]
- PACT is a compact thermal simulator developed by PEACLab that can generate accurate temperature data
- Previous work [2] developed a linear regression model to predict on-chip temperatures for the Intel i7 6950×Extreme Edition processor
- M3D systems face additional thermal issues due to various factors, making thermal management a critical issue [3]
- Runtime thermal management provide one way to manage high temperatures





M3D Diagram

Goal:

- Predict a 100x100 temperature node grid output from PACT for each active layer in a two-tiered M3D processor using a nonlinear regression model

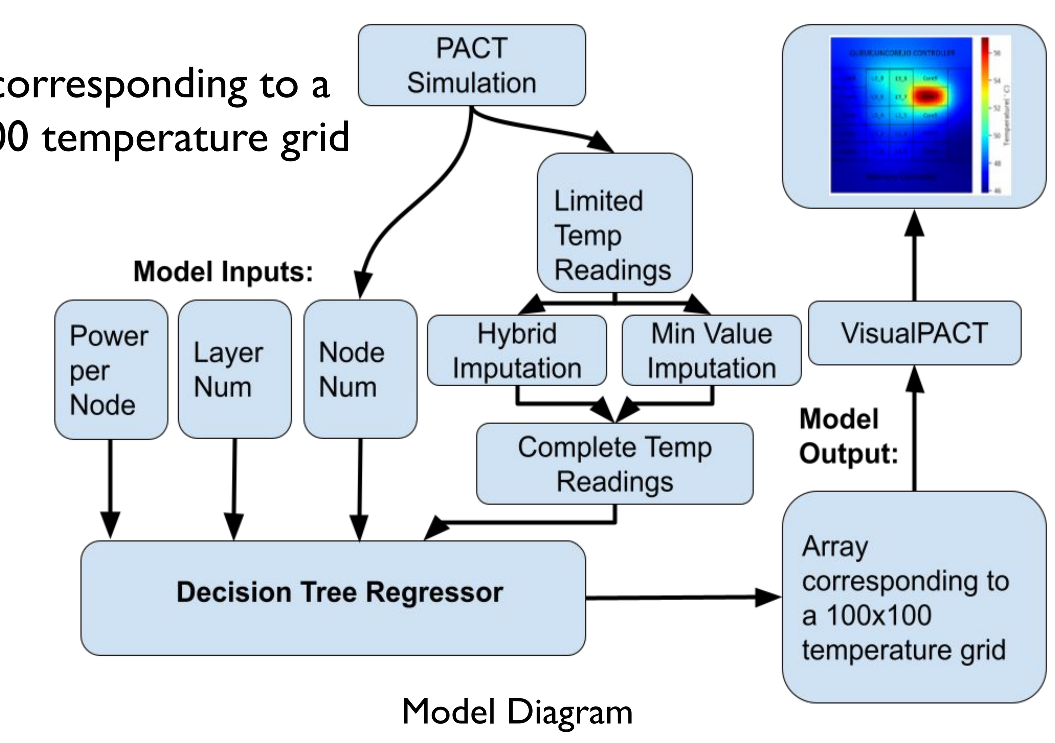
Methods

The Model:

- Decision Tree Regression model from Scikit Learn
  - Default hyperparameters
- Inputs:
  - Power Per Node
  - Node Location/Index
  - Temperature readings from sensors
  - Node Layer
- Model also tested without layer input
- Output:
  - Array corresponding to a 100x100 temperature grid

Dataset:

- 360 different workloads
- Each workload contains a 100x100 temperature node grid for each layer
- Train/test split of 0.75/0.25 and 0.65/0.35 for model with and without layer info, respectively



Model Diagram

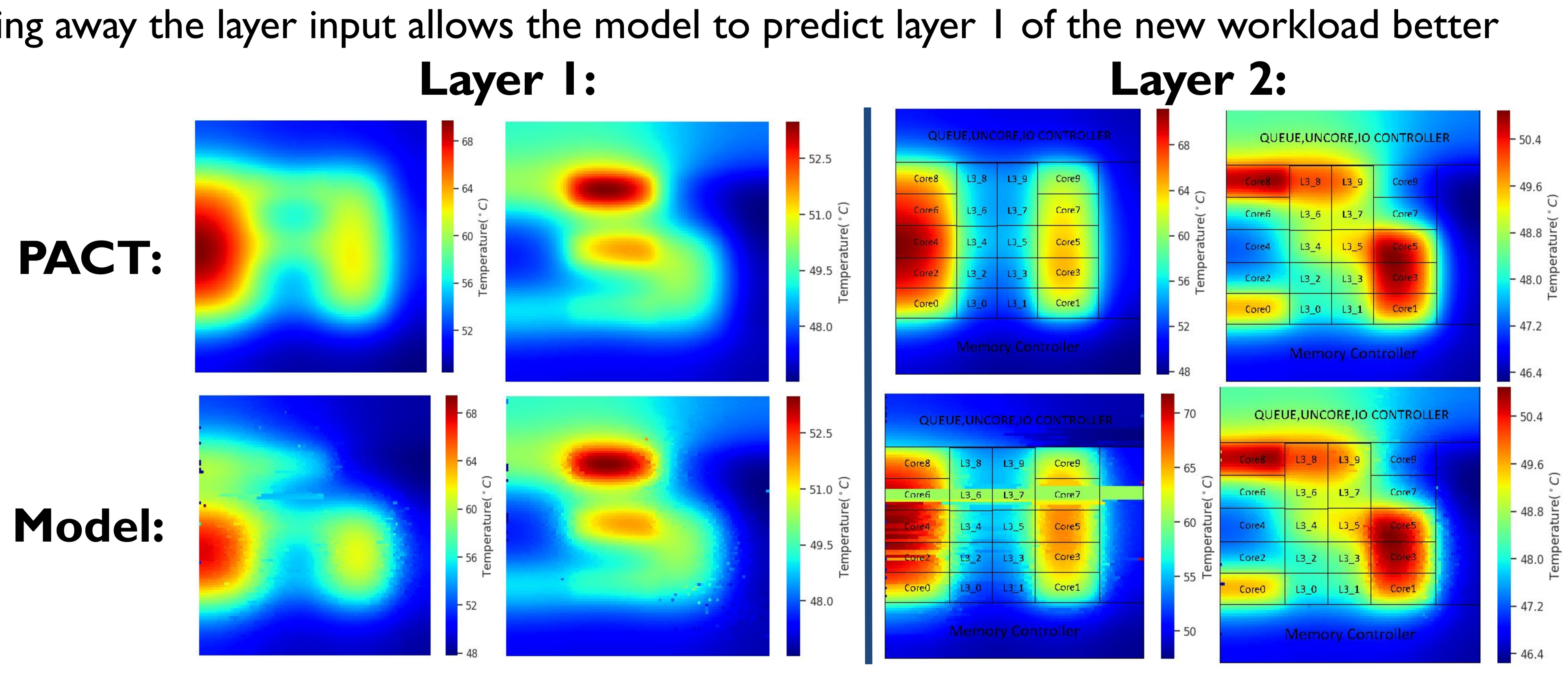
Results

Testing:

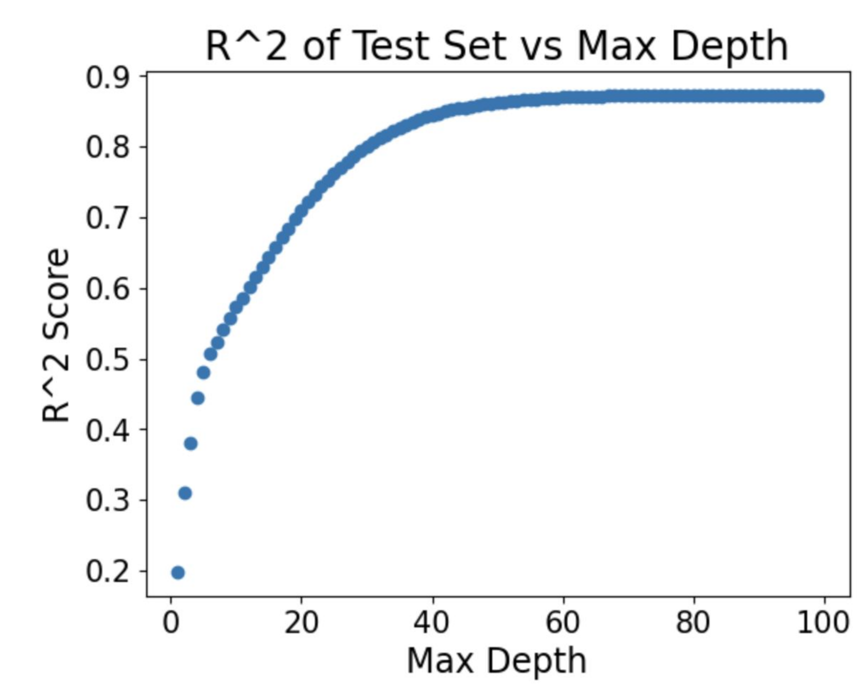
- Model was tested with two imputation methods and 10, 20, and 50 temperature readings given
- Model was also tested with taking away layer input, varying max depths, and a new workload
- Null temperature data filled using
  - Minimum temperature reading given
  - Hybrid approach
    - Bottom layer filled with Sklearn iterative imputer
    - Top layer filled with minimum value

Overall:

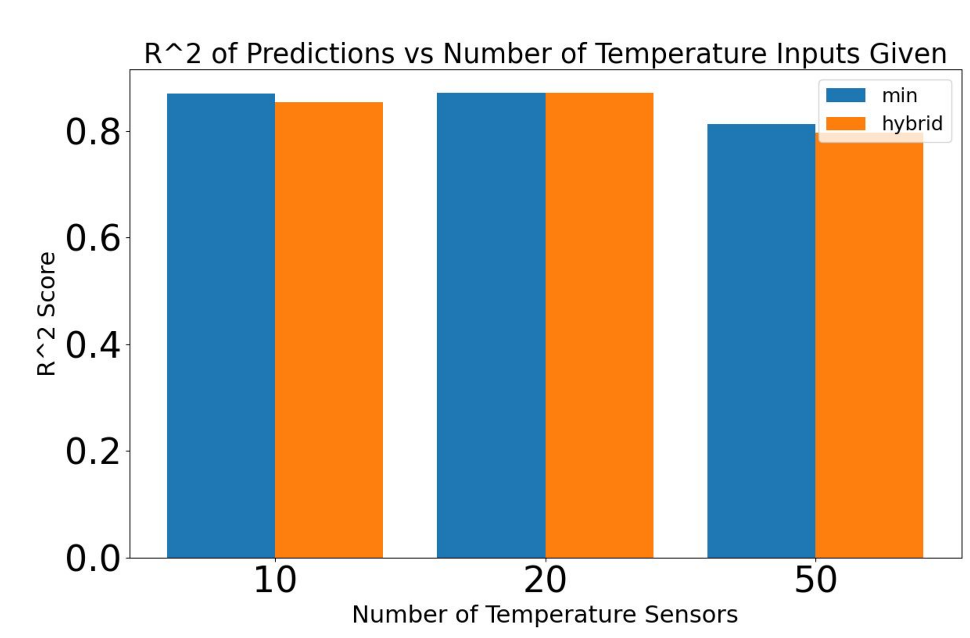
- Model is generally more accurate for layer 2
- Achieved a maximum R^2 score of 0.871, with both min and hybrid imputation, using 20 temperature readings
- Takes < 1 second to generate a prediction for one layer
- Taking away the layer input allows the model to predict layer 1 of the new workload better



Predictions from min, 10 temp readings on workload from dataset(right) and min, 10 readings, and no layer model on new workload(left)



Accuracy vs Decision Tree Max Depth for model given layer data



Accuracy vs Number of Sensor Readings for model given layer data

Conclusion

Summary:

- Decision Tree Regressor achieved good accuracy for predicting layer 2 temperatures, and decent accuracy for predicting layer 1 temperatures.
- Both data imputation methods had similar results, with a maximum overall R^2 score of 0.871
- Increasing the number of temperature inputs to 20 slightly increased accuracy, while 50 temperature readings and a new workload decreased accuracy
  - The decrease in accuracy could be a result of overfitting, limiting max depth and giving less input columns may have helped with overfitting

Future Work:

- Train the model using a larger dataset
- Tune the model hyperparameters, or try other regression models (e.g. Random Forest)
- Test the model with greater variation of temperature sensors and do multiple trials with varying sensor locations

References

[1] K. Dhananjay, P. Shukla, V. F. Pavlidis, A. Coskun and E. Salman, "Monolithic 3D Integrated Circuits: Recent Trends and Future Prospects," 2021. 1

[2] Knox, C, Yuan, Z, & Coskun, AK. "Machine Learning and Simulation Based Temperature Prediction on High-Performance Processors." 2022. 1

[3]. P. Shukla, A. K. Coskun, V. F. Pavlidis, and E. Salman, "An overview of thermal challenges and opportunities for monolithic 3D ICs," 2019. 2.

Acknowledgements

I would like to thank Professor Coskun for giving me opportunity to work on this project in the PEACLab. I would also like to thank Amin Khodaverdian for providing me mentorship and support throughout the project. Finally, I would like to thank Hudson Reynolds, Peter Zhao, Connor Casey, and the RISE program.