Volume 4 | Spring 2023

IEEE Student Magazine

Works from Students at the University of Cincinnati

IEEE
University of Cincinnati Branch

Contents

Foreword from Philip A. Wilsey

Philip A. Wilsey, *Professor*, Dept of Electrical and Computer Engineering

April 14, 2023

I have been a member of the IEEE since 1982 and have served as the faculty advisor to the UC student branch since 1998. The student branch of the IEEE at UC has always been very active and engaged. They hold regular meetings with professional speakers from both academia and industry. They maintain a close cooperation with the local professional chapter of the IEEE and routinely participate in regional IEEE activities. The launch of the UC EECS Student Magazine in 2019 has been another success story for the students. This magazine has been a great source of information regarding the many activities of UC students and provides exposure of student activities to numerous communities both internal and external to UC. As the faculty advisor to these outstanding students, I am pleased to be associated with the high-quality, professional students that matriculate at UC.

While the magazine is sponsored, managed, and published by the IEEE student branch, the group accepts contributions from many fields and disciplines. For example, this issue contains articles from both undergraduate and graduate students; students studying Electrical Engineering, Computer Engineering, Computer Science, Civil Engineering, Medical Sciences, Cybersecurity, and Professional Writing. The topics originate from co-op experiences, graduate and undergraduate research, senior project experiences, and student competitions.

I would like to thank the students for their hard work and dedication to this publication. I look forward to reading this and future issues so I can learn more about the experiences and activities of students at UC.

RevolutionUC 2023 Experience

Andy Au, *B.S., Cybersec.*

***Abstract*—I attended RevolutionUC in 2023, which was my first-ever hackathon. This paper describes my experience, project, and thoughts of the event. I created a project with my group and had a great time!**

I. Introduction

WHEN I attended RevolutionUC, I was a first-year Cybersecurity student with little to no experience with computer programming, networking, etc. Furthermore, this was my first Hackathon experience, so I thought that this was the perfect opportunity for learning and networking.

II. Experience

Upon my arrival, I did not have a team yet, but I was quickly able to find one. Incidentally, they were also first-year students in computer science. After some mild back and forth, we settled on creating “EtsUC,” an online thrift store for University of Cincinnati students to buy and/or sell between each other. It can be thought of as similar to the UC Bookstore, but with furniture. It took many hours for us to get started and agree on the overall layout of the website, but with a quickly diminishing twenty-four hours to create it, we decided to create an overly simplified proof-of-concept. This meant creating a log-in screen, a buy screen, sell screen, and the necessary code to link them all together. After spending most of the first half of the day relearning HTML, CSS, and other foundational programming languages, we all equally split up the work and worked for the entire 24 hours. Of course, it was not all work, as there were plenty of activities, food, and interesting people throughout the event, but we still worked right up to the last second. We did not win anything, but at least we did not get the “Most Useless” prize.

III. Project

Our project’s name was ”EtsUC” (a play on words with Etsy and UC). We wanted to create a project that allowed UC students to buy, sell, and/or rent their old goods, including but not limited to: textbooks, technology, clothing, and furniture. The prototype was just a proof-of-concept as we just included the log-in screen, a catalogue of the available goods, screens for buying and selling, a check-out screen, and code to tie all of them together. For the first half of the hackathon, our team was in-person, but after dinner, two of our members had to leave for the night; We had to asynchronously coordinate thereafter. I worked on the Buy and Sell screens successfully, but once the team split and we were nearing the deadline, I also helped to ”overhaul” the entire site, primarily to synchronize the different screens with the overall design.

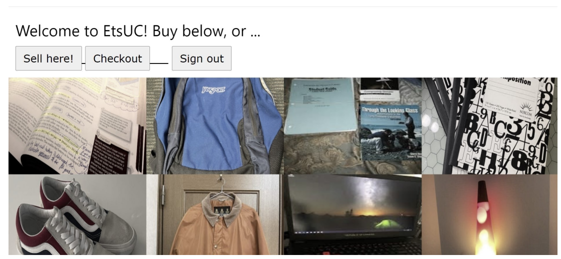

Fig. 1: EtsUC Interface Screenshot of the Main Screen

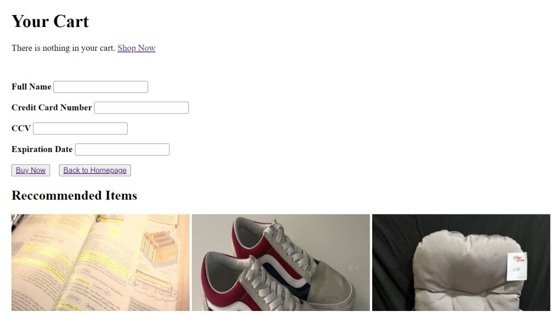

Fig. 2: EtsUC Interface Screenshot of the Checkout Screen

IV. Thoughts

I had fun throughout the entire event. It had been forever since the last time I went camping, and even though it was nowhere near as gritty as a Boy Scout camp-out, the comradery, fun, and valuable learning experiences were still there.

V. Conclusion

RevolutionUC was an amazing event and I sincerely believe everyone in a computer-based major should attend. The reasons for this include, but are not limited to, the networking opportunities, the opportunities to learn skills not taught in classes, and the possibility of taking hackathon ideas even further.

Parkinson's Together and Parkinson's Companion: A Perspective

Mallika Desai, *B.S./M.D., Med. Sci.*

***Abstract*—This paper is the story of Mallika Desai's journey as a student of the University of Cincinnati, volunteer at the Bethany Retirement Home, participant of RevolutionUC, and organizer for Parkinson's Together. Since her last semester of high school, Mallika has been immersed in the study of preventative medicine, research about Parkinson's Disease, and the spread of information about this detrimental disease.**

I. My Inspiration

When Ruth passed away, I felt like I had lost a family member even though she had been a stranger to me just five years earlier. I met Ruth at the end of seventh grade, when I began volunteering at a local retirement home. Ruth lit up every room that she walked into. She was the life of every bingo or game of bridge, had a puppy calendar she gushed about to anyone who would listen, and sang "Singing in the Rain" so much that it would ring in my ears while I waited for my school bus.

About a year into my time as a volunteer at Bethany Retirement Home, Ruth was diagnosed with Parkinson's disease. There had been hints of it earlier, but her tremors had worsened, and she was suffering from falls far more frequently. I watched her recede into her shell, not come out of her room, and become frustrated with everyone around her. She would panic and become upset due to something that initially seemed small to me, like not being able to get a book off a shelf. Sometimes she would simply stay in bed all day. Her children struggled to pay for the support their mother needed and did not know how to help. I felt as if I was losing the warm friend and confidante I knew, and I began to immerse myself in research on Parkinson's disease, medicine, and health policy. I reached out to a neurologist on my own to learn more about how I could help, but I continued to feel helpless. All I could do was sit with her and remind her that she was not alone.

Ruth passed away during the pandemic. Like many families who lost loved ones during the pandemic, I did not get to say goodbye. No volunteers or families were allowed in the facility, and I could not imagine how lonely Ruth must have been. By that point, I had decided that I wanted to become a neurologist. I needed to immerse myself in the study of Parkinson's Disease and become part of the effort for a cure to make sure people would not have to endure what Ruth went through. During my last semester of high school, with the chief of the movement disorders department at a nearby academic hospital as my mentor, I conducted my own research study of Parkinson's disease and the ethics of potential induced pluripotent stem cell therapies for Parkinson's.

II. Parkinson's Together

When I entered college, I carried my goal with me, starting a nonprofit organization that integrates more students like me into the care of Parkinson's patients. April 1st marked the first undergraduate student-led symposium for patients with Parkinson's disease, caregivers, medical students, undergraduate students coming from both medical and non-medical backgrounds, engineers, musicians, physicians, researchers, lawyers, and many other individuals coming together to uniquely implement their respective passions in service of patients - held by my team and I. Within the span of just under two years, we have applied to the IRS to become a nonprofit, organized a conference, created research studies from the ground up at the UC Gardner Neuroscience Institute, learned from the Cincinnati Ballet, participated in Parkinson's boxing and drumming classes, and, most importantly, made lifelong connections with patients at the Gardner Neuroscience Institute (UCGNI).

Fig. 1: Chair Yoga, hosted by Parkinson's Community Fitness at the Inaugural Parkinson's Together Symposium

III. RevolutionUC

Loving Ruth has done far more than fuel my life's work to become a neurologist who conducts research and creates innovative solutions to the problems that Parkinson's patients face. She also taught me the importance of extending myself to grow, which is how I found myself at the RevolutionUC Hackathon. I wanted to create a technology–driven tool that served as a way to encourage Parkinson's patients to take precautions and make lifestyle changes that have been shown to possibly slow progression of this disease. A common question I received in discussions with patients and retirement home residents with Parkinson's disease was how they could stop their condition from worsening. With Parkinson's disease being a neurodegenerative disorder with no cure, most patients experience a worsening of both physical and non-motor symptoms - tremors, rigidity, slowness and loss of controlled movement (bradykinesia, dyskinesia), increased on-and-off symptoms to medication, anxiety, depression, and many others effects. A concrete means of prevention, let alone a one-size-fits-all-method, is not currently possible. This is because Parkinson's disease is especially complex due to the fact that each patient's symptoms could vary drastically from another patient; one patient could have very mild tremors and still regularly go for runs while another may struggle to take a few steps. This is why my mentor, Dr. Espay, refers to Parkinson's disease as a syndrome, or range of conditions.

Fig. 2: Cardio Drumming, hosted by Parkinson's Community Fitness at the Inaugural Parkinson's Together Symposium

1) University of Cincinnati Gardner Neuroscience Institute
While shadowing at UCGNI, I found myself continually coming back to the issue of prevention. According to the Commonwealth Fund, the U.S. globally ranks prevention among the lowest in terms of the overall population living long, productive lives. Prevention has a similarly low ranking when it comes to providing equitable, accessible care to its most vulnerable patients. At the same time, the U.S. spends a far larger fraction of its GDP on healthcare than other countries. This seemingly paradoxical relationship between spending and outcomes can be traced back to the United States' prioritization of reactive healthcare over integrative, preventative care. Our emphasis on reactive care means that we have outstanding acute care technology to save people from emergencies such as heart attacks and strokes. Unfortunately, in my view, this has diverted investment away from developing technology on the preventative care side, which might save many patients from getting to the point of needing acute care to begin with. Recently, there has been a growing global movement to invest in preventative care [1], especially in response to research indicating that it can lower the prevalence of chronic illness and reduce healthcare costs [2].

Fig. 3: Mallika Desai, President of the Parkinson's Together Non-Profit, presenting at the Inaugural Parkinson's Together Symposium

While each Parkinson's patient is extremely unique, I found that every physician recommended exercise, a healthy diet, and social activities to help the patient's mental and physical condition, which was my inspiration for coming to RevolutionUC. If I could build a tool that encouraged residents like Ruth, who did not have family to observe her and remind her to do the preventative activities I tried to remind her of every time I came to volunteer, I could join the burgeoning global movement in technology and healthcare that emphasizes preventative measures in the field that I am passionate about. With my beginner-level understanding of biomedical informatics, I knew creating a tool like this was possible, and I found a wonderful team - Andy Muha (electrical engineering student), Ryan Gengler (computer science student), and Pavan Motagi (informational systems management graduate student), who worked through the night to create Parkinson's Companion. The application utilizes Alexa to converse with residents with Parkinson's disease in retirement homes to remind them not only to take their medications and finish their meals, but guide them through chair exercises, remind them to connect with their family members, and ask about how residents are feeling. The responses are recorded on a platform where family, nurses in understaffed retirement homes, and physicians are able to view them, and, in theory, allow them to support the resident 24/7. We envision that, with more development, we will be able to integrate tools such as enabling direct video calling to families for residents, tailoring voice recognition to better detect elderly voices with vocal tremors (a symptom of Parkinson's disease), and many more abilities that patients and families request. With the support of the innovative UC community and future funding from my nonprofit, we hope to add to the team and watch this project soar.

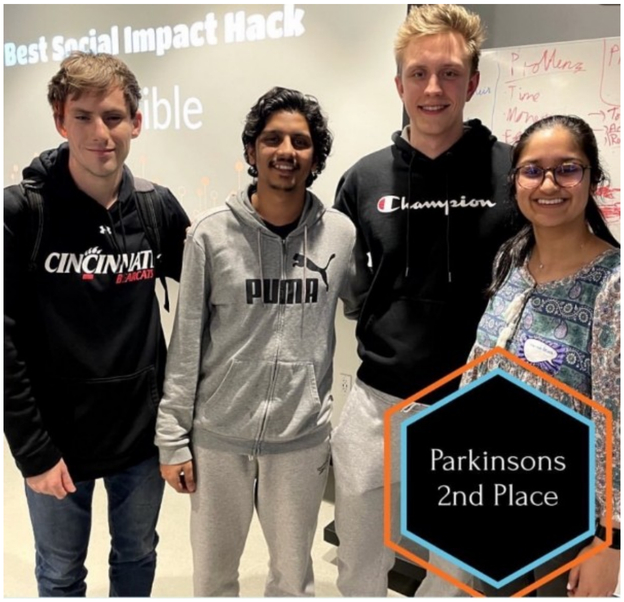

Fig. 4: Parkinson's Companion RevolutionUC Team

IV. Conclusion

My quest to become involved in helping patients with Parkinson's disease blossomed from an unexpected friendship that inspired me and gave me hope. I hope to carry on Ruth's light to many other students who have a passion for care and patients to ensure less patients struggle through the obstacles she faced.

References

[1] Sahar Arshad. *How to Invest in the Rapidly Growing Preventive Healthcare Industry*. June 2022. URL: https://www.nasdaq.com/articles/how-to-invest-in-the-rapidly-growing-preventive-healthcare-industry.

[2] *Health, United States, 2019*. 2021. URL: https://stacks.cdc.gov/view/cdc/100685.

MakeUC Hackathon 2022 Recap

Katy Hildebrant, *B.S., Elec. Eng.*

***Abstract*—MakeUC is a hackathon hosted by the Institute of Electonics and Electrical Engineers (IEEE) at the University of Cincinnati every October. It is a 24-hour event in which individuals or teams create software and/or hardware-centered projects, learn new skills from free workshops and resources, and network with sponsors and other participants. This past year's event took place online and in-person at the University of Cincinnati's main campus.**

I. Introduction

MAKEUC is the hackathon hosted by IEEE at the University of Cincinnati every October. It is a 24-hour event in which individuals or teams create projects centered around software and/or hardware, learn new skills, and network with each other and event sponsors. The 2022 competition took place on October 22nd and 23rd online and in-person at Rhodes Hall. The main activity was "hacking," but a sponsor expo and workshops were among the other fun activities available at the event.

II. Projects

All hackers were encouraged to submit a project individually or with a team of up to four people. A total of 56 projects were submitted. The top three winners were:

1) *CodeBook* by Md. Anzar Ahmad, Mohd Arshad
   Codebook is a website that aims to speed up the software development process for users of all skill levels. One main function is the ability for users to browse and upload code snippets with keywords and phrases. Another function is the "auto code" feature, which will generate code to do a specific task based on user input. Codebook can also accept code from the user and explain it in plain English as well as translate it from one programming language to another.
2) *Santellegrino* by Tom Jurien, Vlad Jurien
   Santellegrino aims to make clinical trials accessible to everyone by allowing them to participate from home. It utilizes an algorithm which only takes into account sound and video to calculate medical data such as heart rate, respiratory rate, nostril dilation, and facial thinning/ enlargement.
3) *Pollution Palooza* by Adam Webb, Grace Smith, Morgan Novean, Paul Weaver III
   Pollution Palooza is a game that takes players on a journey across the globe to learn about past environmental disasters. Each level increases in difficulty and represents a new recovery challenge for players to navigate.

III. Workshops

Workshops are one of the most enjoyable ways to expand your skills at a hackathon. Students worked alongside three of our sponsors to learn new tools and create a fun project:

1) *AWS 101* by Amazon Web Services
   An introduction to AWS cloud computing.
2) *Software Reverse Engineering 101* by HII
   An introduction to reverse engineering principles of both hardware and software with examples in Ghidra.
3) *Tracking Aircraft for Fun and Profit* by London Computer Systems
   An interactive demonstration on how to track nearby aircraft using public APIs.

IV. Organization

A huge thank you goes out to each and every student who volunteered as a MakeUC organizer in logistics, marketing, sponsorship, and technology. MakeUC would not have been impossible without all of you!

Fig. 1: Oluwaseun Abiodun Adekoya, Greg Muha (lead), Nishil Faldu, Natalia Lui, Trung Nguyen, Atharv Shete, Arnav Komaragiri (lead), Sam Burkhard, Katy Hildebrant (director), Jaran Chao, Elaine Mansour (lead), Cat Luong, Kristin Hildebrant, Jack Vo, John Whiting (lead)

V. Acknowledgements

IEEE at UC is also very grateful to the following:

- The MakeUC sponsors, for dedicating time, money, and other resources to supporting opportunities for students.
- The MakeUC judges and mentors, for providing valuable expertise and feedback to the participants.
- The MakeUC hackers, for showcasing their creativity with amazing, innovative projects.

VI. Learn More

If you would like to learn more about MakeUC, please visit our website at https://makeuc.io. If you would like to get involved with the event, or if you have any questions, please send us an email at info@makeuc.io.

My Experiences at the 2023 Transportation Research Board Annual Meeting

Wei Lin, *PhD. candidate, Civil Eng.*

***Abstract*—The Transportation Research Board (TRB) annual meeting is held in Washington, D.C., every January, which attracts lots of people (i.e., government, academia, industry) from the U.S. and around the world. The meeting covers all transportation modes such as public transportation, roadway transportation, air transportation, etc.**

I. Introduction

The Transportation Research Board's (TRB) annual meeting is the most comprehensive, professional, and popular conference meeting in the transportation field. Recently, the TRB's 102nd annual meeting was held January 8–12, 2023. The TRB aims to identify the best path in innovation and its mission contains three primary roles:

1) Research featured in more than 140 annual publications and 300 research projects.
2) Convene with 170+ standing technical committees, 8000+ active committee and panel members, etc.
3) Advise with more than 12 active consensus study committees [1][2].

There were more than 13,000 attendees at the meeting, perhaps the single largest gathering of transportation practitioners and researchers in the world [3]. There were more than 600 workshops, lecture sessions, committee sessions, and poster sessions held, and 200 exhibitions participating [4]. The exhibitions are shown in Figure 1.

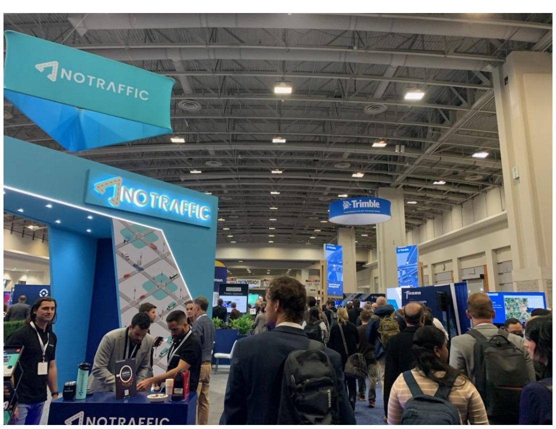

Fig. 1: Exhibitions at the 2023 TRB Annual Meeting.

Every year, the TRB's core technical activities are sponsored by many sources of financial sponsors which consist of federal, state, and local government agencies, professional societies, and organizations representing various industry groups [5]. The meeting is open to any participant, encouraging more people to get involved with TRB and advance transportation developments and their careers. The opportunities provided include volunteering, joining a TRB standing technical committee, participating in cooperative research program panels and projects, and attending TRB webinar programs [6].

II. Experience

Wei Lin is a fourth-year PhD student in the Transportation Engineering Program at the College of Engineering & Applied Science. Her research expertise lies in the area of intelligent transportation systems, data science, and traffic safety. Her research focuses on developing emerging connected vehicle and autonomous vehicle (CAV)-generated mobility data intelligence and analytics models to streamline cooperative traffic operation and safety and artificial intelligence applications with transportation automation, which are supported by the Internet of Things (IoT) technologies. This year is her third year to attend the TRB annual meeting; it was a wonderful experience for her. Notably, it is a great honor for her to receive the young scholar award as lead author for the paper "Modeling Intelligent CAV Data Fusion for Adaptive Signal Control" (TRBAM-23-01358) at the AED30 Information Systems and Technology Committee during the 2023 TRB annual meeting and present her research at the sub-committee meeting, as shown in Figure 2.

Fig. 2: Award session and Presentation at the 2023 TRB Annual Meeting.

Meanwhile, the paper was presented at the information systems and technologies poster sessions, as shown in Figure 3.

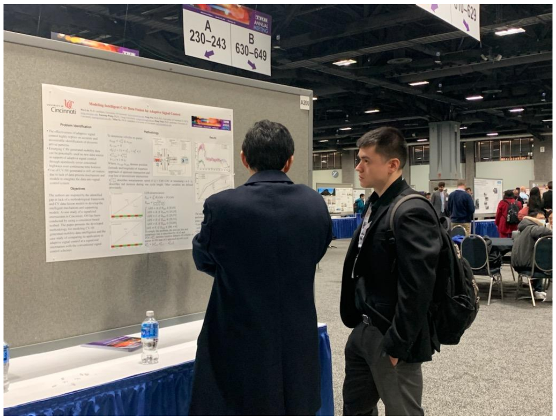

Fig. 3: Poster Presentation at the 2023 TRB Annual Meeting.

III. Giving Gratitude

Here, I express my greatest gratitude and sincere thanks to my advisor, Dr. Heng Wei. He is always very patient with me and open to discussion even when I did not quickly adapt to the pace of my PhD degree journey. He taught me to understand the nature of scientific and technological knowledge and perform high-quality research by offering me opportunities to get involved in a great number of research activities and teaching me philosophical logic for research and life. He is always guiding me down the right path.

References

[1] *About Us*. URL: https://www.nationalacademies.org/about.
[2] *About Transportation Research Board*. URL: https://www.nationalacademies.org/trb/about.
[3] *Stay Informed*.
[4] *Thank you for attending the 2023 TRB Annual Meeting*. URL: https://www.trb.org/AnnualMeeting/AnnualMeeting.aspx.
[5] *Transportation Research Board Core Program Sponsors*.
[6] *Get Involved with the Transportation Research Board*. URL: https://www.nationalacademies.org/trb/get-involved.

My Experience at the UC Innovation Challenge

Greg Muha, *B.S., Elec. Eng.*

***Abstract*—The Innovation Challenge is a semester-long experience where students are able to learn how to become entrepreneurs and to develop their ideas into products. There are weekly meetings where students at the University of Cincinnati (UC) are invited to collaborate. Free food is provided, and engineering students can even get paid if they satisfy the deliverables. These deliverables include recording a pitch deck and presenting at the final pitch competition. Regardless of your major, it is a great event to meet other motivated students who share a passion for innovation.**

I. Introduction

DURING the semester of Spring 2023, I participated in the Innovation Challenge hosted by the Engineering Tribunal. It was a fun and rewarding experience where I thought of an idea and worked with two other teammates to create a functional prototype. We met weekly to discuss our ideas and support each other through the innovation process. The first weeks of the Innovation Challenge consisted of brainstorming ideas. My idea was to crowdsource a chip design and draw inspiration from previously-existing routing games so that anyone could make integrated circuits in an easy and fun way.

II. Generating an Idea

Every week, our team met to develop our ideas and to learn to become better entrepreneurs. The first weeks consisted of ”learning the ropes of entrepreneurship” where we attended presentations and toured the 1819 facilities. One week, Andrew Jajack - a successful entrepreneur who started the first ever team from this region to be accepted into Y-Combinator - talked about his experience in Silicon Valley and the challenges of building a start-up. We learned a lot of valuable information, such as how to pick good partners and generate successful ideas.

I used my experience with Very-Large-Scale-Integration (VLSI) to think of products that could improve the design process. VLSI is the field of creating integrated circuits, which can have billions of individual transistors, all of which must be placed and routed so that the functional and timing requirements of a design are met. We underwent market research and I found that non-recurring engineering costs are becoming a dominant cost factor in integrated circuit production as they surpass billions of components [1]. The average silicon design cost for a system-on-chips (SoC) across all geometries was $4.8M in 2019 [2]. Commercial software for chip design is often expensive and difficult to use, requiring highly-experienced engineers. I believed that I could improve on the existing design software that is used to create these circuits. From my previous experience, I thought that chip design was something anyone could learn to do and I wanted to develop an application that could allow anyone to collaborate and contribute towards a chip design. If successfully implemented, chip design could be improved to become quick, accessible, and affordable.

I drew inspiration from popular mobile games where players connect dots on a screen. We figured that if people would play those types of games for fun, we would be able to profit off of their design as a chip. This would also be appealing to companies that do not have a team of experienced VLSI engineers, which requires a very large budget and timeline. We believed that this tool would be able to make custom silicon design more widely accessible to smaller companies. Traditionally, only large companies, like Apple, Intel, and NVIDIA, are able to make their own custom silicon design due to these high barriers of entry into the market.

Additionally, the main producers in VLSI design software are Synopsis and Cadence and they can charge upwards of tens-of-thousands of dollars per license for a piece of software. They also provide training at a very high cost, often requiring several hundreds-of-dollars just to train a single engineer on an expensive software. A typical chip-design flow requests several pieces of software. Our hopes for this project was to make at least a couple of steps of the physical design process very low-cost without requiring expensive software, labor, or time.

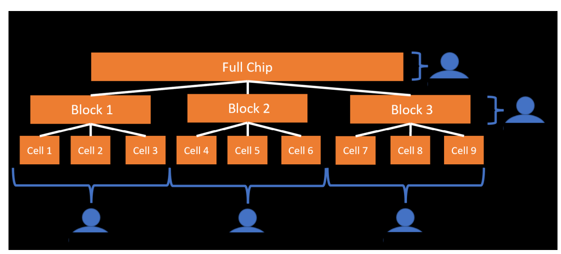

Fig. 1. Hierarchical Design Method

A. Tackling the Problem

I invited two partners to help me develop the idea - Andrew Muha, who has experience with back-end and front-end design, and Quan Le, who has website design experience. Our goal was to create a VLSI software tool that made chip design more accessible, cost effective, and fun. We aimed to build a JavaScript application that could be run in a web browser, importing parts of a design so that a user can place components and route the traces between them. The user application would communicate with a server that would provide parts of a design. One of the main challenges in collaborative chip design is that each component is interconnected with others, so graph partitioning is done on a design so that ”blocks” could be created that have the minimized number of connections between blocks. The tool would then combine the blocks so that another user could connect components together. This could be repeated in a hierarchical fashion until the entire chip was assembled (Fig. 1).

III. Prototyping

We utilized the skills of each team member to create a successful prototype. I used my VLSI experience from previous co-ops to make our software compatible with existing VLSI design flows. Andrew used his front-end experience to develop a user interface that was appealing and intuitive to use. Quan used his front-end experience to design an effective menu that allowed users to change the settings in the application.

We split development into a back-end and a front-end. I designed the back-end, which used C++ to parse a Verilog netlist. A netlist specifies the design of a circuit, which includes all of the components and the connections between them. A netlist is a type of hypergraph and can be partitioned into ”blocks” with a minimized number of connections using hypergraph partition techniques. I parsed the Verilog netlist into the format of an academic hypergraph, *hMETIS* [3]. The hMETIS tool then splits the netlist into the least connected blocks, which would be sent to users so that they could compile each block. That way, users could work on each one of their blocks as independently as possible, which makes combing blocks easier in the later stages.

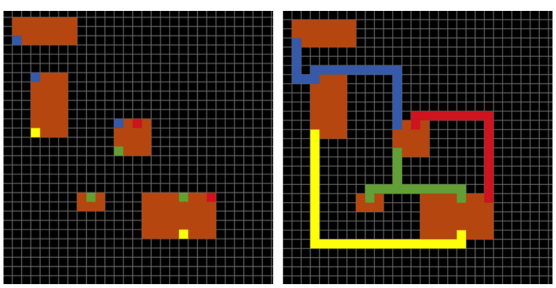

Fig. 2. User Interface for Connecting Blocks

On the front-end, Andrew used the HTML Canvas to create a GUI that enabled users to place blocks and route traces between them. The goal was for it to be as intuitive as possible. There were a variety of problems that needed to be solved. We wanted a user to be able to click on an existing trace and extend that trace by clicking on another empty section. As the user moved their mouse, the software would compute the shortest distance from the starting point, where they clicked, to where their mouse was currently located. This system allowed users to quickly place routes. Andrew also implemented a system that periodically checks each connection to ensure there was a pin. Cycle-checking is necessary so that the application will not run in an infinite loop.

IV. Final Pitch

The final pitch was in mid-April. We improved our initial pitch-deck to include pictures of our prototype and expanded our business model. We proposed that our application could be tested by any user. The biggest challenge was incorporating our 11-slide presentation into five minuets. The large prizes were awarded to the top three projects with additional prizes for category winners that include ”People’s choice” and ”Best Prototype.” I felt that my presentation was able to hold its own in front of the judges. Although my team did not win, I was happy to see an electrical engineering project win first place: a battery that can be recharged through kinetic motion.

V. Conclusion

Overall, this was a very fun and beneficial experience that I and many other participants were able to learn from. I will definitely be participating in the future and plan to generate more ideas, some of which might go on to be successful business ventures!

References

[1] D/D Advisors. *Let’s Build a Chip – With Math*. May 2021. URL: https://digitstodollars.com/2021/05/28/lets-build-a-chip-with-math/?utm_source=Benedict%5C%27s+Newsletter&utm_campaign=2f28777d01-Benedict%5C%27s+newsletter_COPY_01&utm_medium=email&utm_term=0_4999ca107f-2f28777d01-70794709.

[2] Semico Research. *Average Silicon Design Cost for SoCs Across all Geometries was $4.8M$ in 2019, says Semico Research*. URL: https://semico.com/content/average-silicon-design-cost-socs-across-all-geometries-was-48m-2019-says-semico-research.

[3] David A. Papa and Igor L. Markov. *Hypergraph Partitioning and Clustering*. University of Michigan, EECS Department. URL: https://web.eecs.umich.edu/~imarkov/pubs/book/part_survey.pdf.

2023 Senior Design

Jacob Rowland, *B.S., Elec. Eng. Tech.*

***Abstract*—My name is Jacob Rowland. I am a senior in the College of Electrical Engineering and Applied Sciences. I am majoring in the field of Electrical Engineering and Technology. This semester, I worked on my senior design project of utilizing a 2.4GHz Yagi Antenna to demonstrate how WI-FI Direct is a superior alternative to traditional WIFI in terms of signal strength and power consumption. My role in the team was antenna research and design.**

I. Background

I Have always been one of those people who was always tinkering with electronics and that hobby turned into a passion. I decided to attend the University of Cincinnati since it has a prestigious engineering program. Since my enrollment, I have had the opportunity to expand my engineering horizons and work with various companies such as American Municipal Power and Valco Melton.

II. Efficient Directional Wi-fi

My senior design project is to create a means to properly utilize directional WI-FI to cut down on the excess energy spent by broadcasting WI-FI throughout a given range.

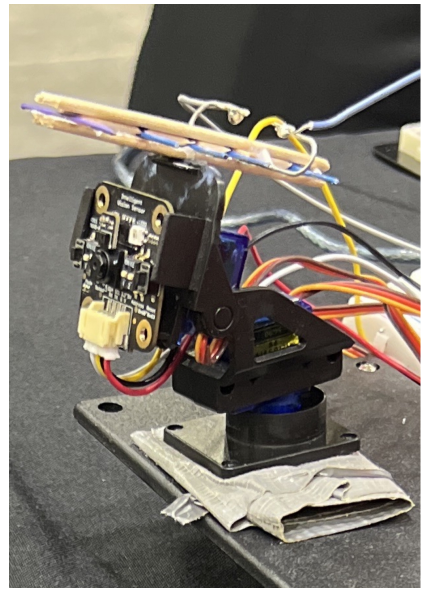

Fig. 1. Base station showing optical sensor and close up of Yagi Antenna

A. Design

For the project, I was assigned the task of researching antenna design and constructing one that is effective at a short to medium range for our testing purposes. This antenna is intended to be consistently adjusted through a motor that ensures the antenna is following the object meant to be receiving the WI-FI signal.

B. Prototyping and Testing

Although our final tests have yet to begin, prototyping for me began as early as deciding what materials were to be used in antenna construction. My goal in mind was to create an antenna that would be cost-effective as well as easy to replicate in the event of a technical failure. The model I used was a 2.4GHz Yagi-style antenna. This style of the antenna is easily recreated using popsicle sticks and paper clips, two items commonly found in a household. A WI-FI module is then soldered onto the antenna for testing purposes.

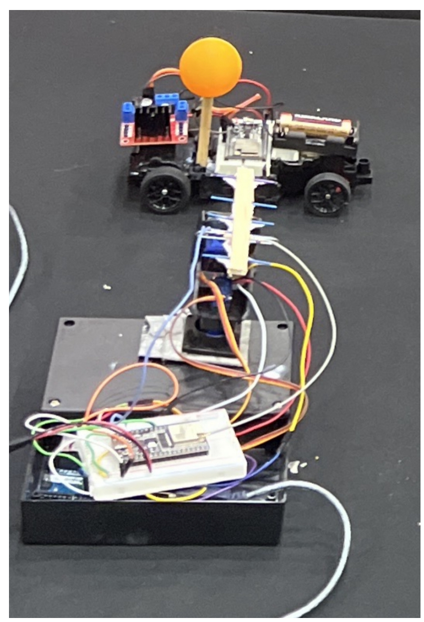

Fig. 2. Prototype being testing by sending signal to RC car

C. Improvements

The first improvement was optimizing the range of the Antenna. Our first model was much larger than intended and was estimated to have a range of 5 miles. While it could create fascinating results, it also created the issues of flexibility and accuracy when broadcasting to the RC car. The second improvement was the quality of the build. The first model had problems with fragility. During the final stages of construction, pieces began to fall off and, while it was salvageable, the malfunction could create issues during future data testing. To ensure future models were sturdier, a better frame was put in place and we conducted plenty of stability tests during construction to ensure all parts were appropriately attached.

III. Thoughts

Overall, this project has been a positive influence on me. Getting to work in a team environment has been a good learning experience. While our project is in the middle stages and won't be complete until roughly April, I look forward to seeing how our technology could be applied to modern WI-FI systems inside of the everyday home.

IV. Final Words

Thank you for reading what I had to say. If I had to give any final words of advice it would be to always contribute your part of the workload to a team. I have had times during this project where I fell ill and could not contribute as much as I wished, and I felt like I was being unfair to the others. A team can easily fall apart through the failures of one member.

The Effect of Artificial Intelligence on the Job Market

Quan Le, *B.S., Comp. Sci.*

***Abstract*—This paper provides an overview of the growing impact of Artificial Intelligence (AI) on various technology sectors and explores the positive and negative effects of AI in our lives. This paper argues that the continued advancement of AI will have a significant impact on the job market by creating new job opportunities, replacing certain roles, and requiring the development of new skill sets. By analyzing empirical data, this paper highlights the potential for AI to both replace and create unique jobs. Overall, this paper underscores the need for ongoing research and further evaluation of AI's impact on the job market and the development of effective strategies to address the resulting economic and social implications.**

I. Introduction

ARTIFICIAL INTELLIGENCE (AI) refers to the simulation of human intelligence in machines that are programmed to perform tasks that typically require human cognition such as learning, problem-solving, decision-making, and perception. Over the years, AI has rapidly developed to become an essential part of our work and our lives through healthcare and education. In healthcare, AI is being used for diagnosing diseases, predicting patient outcomes, and developing personalized treatment plans [1]. In education, AI is used for personalized learning, helping students in various degrees of learning through tools [2]. One of the most trending AI tools you may know of would be ChatGPT, a specialized AI tool which took the world by storm in November of 2022.

The increasing impact of AI is attributed to its ability to process vast amounts of data and to utilize that data to perform complex tasks within milliseconds much more efficiently and accurately than humans can. Bill Gates noted: ”The development of AI is as fundamental as the creation of the microprocessor, the personal computer, the Internet, and the mobile phone. **It will change the way people work, learn, travel, get health care, and communicate with each other**. Entire industries will reorient around it. Businesses will distinguish themselves by how well they use it” [2]. However, the widespread adoption of AI also raises concerns regarding employment.

II. The Impact of Artificial Intelligence

With the increasing capabilities of AI, there is growing concern about its effect on the **job market**. It has the potential to create new job opportunities and increase efficiency and productivity in the workplace, but at the same time, poses a threat to certain jobs that could become automated. Therefore, it is important to have a conversation on the positive and negative effects of AI on the job market.

A. The Positive Impact of AI

On the positive side, AI has the potential to create new job opportunities in fields such as data science, machine learning, and prompt creation. As AI technologies continue to advance, there will be a growing demand for professionals with the skills and knowledge to develop, implement, and manage these technologies. Furthermore, AI can increase efficiency and productivity in the workplace, freeing up time and resources that can be allocated to even greater means of production.

B. The Negative Impact of AI

On the negative side, AI poses a threat to certain jobs that could become automated such as routine manual labor, data entry, customer service roles, and programming. This could result in job displacement and unemployment for workers in these fields. This result is especially harmful towards those without the necessary skills in technology, do not have an educational background, or have not had exposure to technology, and they will struggle to find new employment opportunities.

III. A Symphony of Possibility: The Potential of Artificial Intelligence in the Job Market

In the online sphere, it is near impossible to not know about AI replacing the workforce and further changing our lives. This can be seen with ChatGPT and, as billionaire and entrepreneur Bill Gates wrote: ”You will be able to use natural language to have this agent help you with scheduling, communications, and e-commerce, and it will work across all your devices. Because of the cost of training the models and running the computations, creating a personal agent is not feasible yet, but thanks to the recent advances in AI, it is now a realistic goal. Some issues will need to be worked out: For example, can an insurance company ask your agent things about you without your permission? If so, how many people will choose not to use it?” [2].

AI has the potential to revolutionize the workplace by improving productivity in various ways. One of the key ways AI can improve workplace productivity is by automating routine and repetitive tasks. For example, in customer service, chatbots can be programmed to handle simple customer inquiries, allowing human employees the time to handle more complex issues. This not only increases efficiency, but also reduces the workload and stress on human employees.

In addition to automating routine tasks, AI can also assist employees in making better decisions. By analyzing large amounts of data and providing insights and predictions, AI can help employees make informed decisions quickly and accurately. For instance, in finance, AI-powered tools can analyze data to detect fraud and anomalies, enabling employees to take appropriate actions. With a reduced work time, the employee can focus on more high priority tasks and improve their productivity.

In the current AI era, the emergence of artificial intelligence technology has led to the rise of innovative startups and companies that are actively hiring people. Unlike the previous internet era, where the main focus was on software development and digital services, AI has opened up new opportunities for jobs in various sectors of employment. Many new companies based on AI technology are focused on developing cutting-edge AI tools and platforms that can be used across industries such as healthcare, finance, and education. These companies are not only creating new job roles for AI engineers and developers, but they are also employing individuals from other disciplines such as data scientists, business analysts, and marketing professionals. Companies that have been actively hiring people from across multiple disciplines include OpenAI, Speechmatics, Builder, and DeepMind. These companies are able to attract top talent from around the world and have created new job opportunities for individuals with diverse skill sets that would otherwise be excluded from the technological employment sector [3].

IV. The Symphony of Loss: How Artificial Intelligence is Hurting the Job Market

Automation was the leading cause of layoffs back when the main victims were physical workers. Now, with the skyrocketing growth of machine learning and AI, the target this time seems to be workers that have, are pursuing, and will be pursuing higher education.

One of the primary reasons that AI is causing job layoffs is due to its ability to automate repetitive and mundane tasks. Many industries, such as manufacturing and retail, have already implemented AI-powered automation, resulting in job losses for workers who previously performed these tasks. For example, self-checkout machines in retail stores are becoming much more common. Similarly, robots in factories and warehouses are being employed more frequently in manufacturing, leading to fewer positions for assembly line workers and this trend continues to grow [4].

Surprisingly, occupations that require highly specialized skills are also in the impact area of this invention. About 80 percent of mathematicians, accountants and auditors, news analysts, reporters, and programmers reported feeling the impact of AI [5]. This can take the form of occupation loss, changing to a different role, or requiring less time to finish mundane tasks.

Another way AI is causing job layoffs is through its ability to replace human decision-making in various fields. In industries such as finance and law, AI algorithms can perform analytical tasks and make decisions with increasing accuracy, leading to fewer job opportunities for professionals in these fields. Some financial firms have already begun to use AI to make investment decisions, leading to layoffs for traditional human stock traders. Similarly, some law firms are starting to use AI to analyze legal documents and make recommendations, which could lead to job losses for paralegals and junior lawyers.

V. Conclusion

Artificial Intelligence is becoming increasingly prevalent across various technology sectors and its impact on the job market cannot be overlooked. As AI continues to advance, it is expected to create new job opportunities as new companies are born through this need and require the development of new skill sets, prompting potential engineers to work harder than before. While this may lead to positive outcomes such as increased efficiency and productivity and excess time for design and creation, it also poses challenges such as job displacement and a need for re-training the workforce. Therefore, it is crucial to have this discussion as this technology not only impacts this generation but the future generation as well.

References

[1] JavaTpoint. *Artificial Intelligence in Healthcare*. 2023.

[2] Bill Gates. *The Age of AI Has Begun*. https://www.gatesnotes.com/The-Age-of-AI-Has-Begun. 2023.

[3] BBC News. *AI beats humans in eight-year-long StarCraft II gaming battle*. 2021. URL: https://www.bbc.com/news/technology-65102150 (visited on 04/21/2023).

[4] BBC News. "How AI is helping to predict and prevent suicides". In: *BBC News* (June 2019). URL: https://www.bbc.com/news/business-48760799.

[5] Luke Hurst. *OpenAI says 80% of workers could see their jobs impacted by AI*. Mar. 2023. URL: https://www.euronews.com/next/2023/03/23/openai-says-80-of-workers-could-see-their-jobs-impacted-by-ai-these-are-the-jobs-most-affer (visited on 04/21/2023).

History of LaTeX

Natalia Lui, *B.A., Rhet. and Prof. Writ.*

***Abstract*—LaTeX is an academic standard for journal submissions. This paper will explore the impact that LaTeX has had and will continue to have in the academic and scientific community.**

I. Introduction

ONE of the most important factors to success is longevity. Since the invention of the printing press in the 15th century, the mass production of knowledge is a convenience only humans have access to [1]. These printed pages hold formulas, concepts, laws, and words of wisdom from decades past. Computers are a recent invention compared to the commercialized paperbacks of famous writers, mathematicians, and philosophers, however, the uniformity of academic reports had much to be desired.

Since its release in 1984 by Leslie Lamport, LaTeX has become a free-to-use typesetting software for anyone with access to a computer and a passion to share their work. Access to an expansive online database of human knowledge and scientific discoveries was unheard of in the early 19th century. Such an evolution in technology has further improved this online landscape of connectivity and longevity.

II. What is LaTeX?

LaTeX is a document preparation system for high-quality typesetting that can be used for any form of publishing. It is most commonly used for medium to large technical or scientific documents. The difference between LaTeX and Microsoft Word or Google Docs is that LaTeX is not a word processor. It instead separates the content of the document with its design. Due to the nature of open-source software, there is a large and active community of users that create packages and distribution software to continually improve the accessibility, function, and dependability of LaTeX.

A. LaTeX vs TeX

As seen in their namesake, LaTeX and TeX are quite similar to one another. The difference being the use of macros, which shifts the focus of formatting your document to content creation [2]. TeX is a typesetting system written and designed by Donald Knuth, a computer scientist from Stanford University. Released in 1978 as an open-source project, TeX was a popular means of typesetting complex mathematical formulas within academia and continues its use in LaTeX, ConTeXt, and other macro packages [3]. The use of TeX created a uniform of mathematical spacing, hyphenation and justification, metafont, and macro language with specific commands for formatting details. [4]

Fun Fact
The correct pronunciation of LaTeX is "Lah-tech" or "Lay-tech" [5].

B. Features of LaTeX2e

Since its initial release, LaTeX has gone through many revisions and accumulated many more features. The following are some notable features found in LaTeX2e [5]:

- Typesetting journal articles, technical reports, books, and slide presentations.
- Control over large documents containing sectioning, cross-references, tables and figures.
- Typesetting of complex mathematical formulas.
- Advanced typesetting of mathematics with AMS-LaTeX.
- Automatic generation of bibliographies and indexes.
- Multi-lingual typesetting.
- Inclusion of artwork, and process or spot colour.
- Using PostScript or Metafont fonts.

C. Versions and Where to Download Them

LaTeX2e is the current version of LaTeX and can be used through downloading the software or using an online editor. Software is distributed through the Comprehensive TeX Archive Network (CTAN) servers and the TeX User Group (TUG) [6][7]. Online editors, such as Overleaf and Papeeria, are much more accessible and effective for team collaboration. These editors often require an account to save and access your work and provide priced packages for collaborative efforts and quality-of-life improvements.

III. Leslie Lamport

Lamport was born on February 7, 1941 and can be found at the Microsoft Research Center. Best known for his seminal work in distributed systems and the initial developer of LaTeX, Lamport was the winner of the 2013 Turing Award for imposing clear, well-defined coherence on the seemingly chaotic behavior of distributed computing systems, in which several autonomous computers communicate with each other by passing messages [8]. Lamport was a TeX user who needed macros to complete his work. He thought that, with a little more effort, he could make his macros accessible for others. More information about Leslie Lamport, and his account of how LaTeX came to be, can be found at his website [9].

IV. Staying Connected

In this digital world, staying connected is one of the most difficult endeavors to undertake. Fortunately, LaTeX is a large community with multiple communities to connect to:

1) The LaTeX2e Release Newsletter was last released on November 11th, 2022 and covered auto-detecting kex/value arguments and various smaller changes and improvements to the base, graphics, and tools bundles of LaTeX [10]. There are many more editions, so I highly recommend you to check them out yourself!
2) The LaTeX Project Team has a collection of articles, papers, and conference presentations that cover multiple topics on design, concepts, packages, the LaTeX license, and even the programming language itself [11].
3) If you ever find yourself unsure about a function in LaTeX or want to contribute to a discussion, check out https://tex.stackexchange.com, the Stack Exchange for LaTeX!
4) The TeX User Group is a membership-based not-for-profit organization, founded in 1980, for anyone who uses the TeX typesetting system created by Donald Knuth and/or is interested in typography and font design. Your own institution could have a community for LaTeX!

V. Conclusion

Regardless of your form of academia, writing is a necessary part of recording and sharing your work. Consider using LaTeX or TeX, and if you are an avid user, become part of the community! Remember, one of the most important factors to success is longevity, so publish what you love, share it with an online community, and be the spark that drives innovation.

References

[1] History.com Editors. *Printing Press*. May 2018. URL: https://www.history.com/topics/inventions/printing-press.

[2] Matt Harzewski. *The Student's Guide to LaTeX Markup: What It Is and Why You Want It*. July 2020. URL: https://medium.com/swlh/the-students-guide-to-latex-markup-what-it-is-and-why-you-want-it-651e723ce0c8.

[3] Alexia Gaudeul. *Do Open Source Developers Respond to Competition?: The (La)Tex Case Study*. Mar. 2007. URL: http://dx.doi.org/10.2139/ssrn.908946.

[4] *TeX*. URL: https://en.wikipedia.org/wiki/TeX#cite_note-5.

[5] *About LaTeX*. URL: https://www.latex-project.org/about/.

[6] *LaTeX – A TeX macro package that defines LaTeX*. URL: https://www.ctan.org/pkg/latex.

[7] *Free TeX implementations*. URL: https://www.tug.org/interest.html#free.

[8] *About: Leslie Lamport*. URL: https://dbpedia.org/page/Leslie_Lamport.

[9] *My Writings*. URL: http://lamport.azurewebsites.net/pubs/pubs.html.

[10] *LaTeX2e Release Newsletters*. URL: https://www.latex-project.org/news/latex2e-news/.

[11] *Publications by the LaTeX Project Team*. URL: https://www.latex-project.org/publications/.

Haskell Vs. Python, A Functional Dive

Ishaan Santhis, *M.S., Comp. Sci.*

I. Introduction

HASKELL and Python are functional programming languages with unique features that distinguish them from one another. Developed in 1987, Haskell is a purely functional language that uses lambda calculus, which means that identical inputs to a function always produce the same results. Haskell is used to create server-side applications, spam filters, and compilers, and is characterized by its declarative code and lazy evaluation. In contrast to traditional programming languages, Haskell employs expressions rather than statements and uses a garbage collector to dispose of data that is not susceptible to change.

Python is a dynamically-typed programming language that was first released in 1991. It is a multi-paradigm language, meaning it supports functional programming paradigms. Python is known for its simplicity, readability, and ease of use. Python supports functional programming concepts like higher-order functions, lambda expressions, and list comprehensions.

II. KEY DIFFERENCES

One of the key differences between Haskell and Python is their approach to type-checking. Haskell's strong static typing system detects type errors at compile-time while Python uses dynamic typing which can only catch type errors at run-time.

Another significant difference is their approach to concurrency and parallelism. Haskell supports both concurrency and parallelism, allowing Haskell programs to be written more parallelizable than Python counterparts. Python has been traditionally more challenging to parallelize due to its use of the Global Interpreter Lock (GIL). However, recent versions of Python have introduced the asyncio library, which improves parallelism.

Regarding performance, Haskell is generally faster than Python, especially for scientific and numerical computing tasks. Haskell's static typing system and lazy evaluation enable it to optimize code more effectively than Python. Python, however, has a vast ecosystem of third-party libraries and frameworks, which makes it a popular choice for data analysis and machine learning tasks.

Python is generally considered to be more beginner-friendly than Haskell regarding ease of use and learning curve. Python has simpler syntax and a more forgiving type system, which makes it easier for new programmers to get started. Haskell has a steeper learning curve, especially for programmers unfamiliar with functional programming concepts. However, once programmers become proficient in Haskell, they often find it to be an elegant and powerful language.

III. Key Points About Haskell

Haskell is a purely functional programming language that has been widely used in a variety of contexts. Some of the best examples of Haskell's benefits in functional programming include:

1) Concise and expressive code. Haskell's strong type system, lazy evaluation, and powerful abstractions allow developers to write code that is both elegant and expressive.
2) Concurrency and parallelism. Haskell's functional nature makes it particularly well-suited for concurrent and parallel programming. The language provides strong support for asynchronous programming and has powerful abstractions for parallelism, such as the ”par” and ”pseq” operators.
3) Type safety. Haskell's strong type system helps catch errors at compile time, reducing the likelihood of run-time errors and making code more reliable.
4) Performance. Haskell's lazy evaluation and the sophisticated run-time system can lead to significant performance gains, particularly for compute-bound applications.
5) Research and experimentation. Haskell's purity and strong type system make it a popular choice for researchers and academics exploring new ideas in functional programming.

A few places where Haskell has Excelled since its origins:

1) GHC (Glasgow Haskell Compiler) is a high-performance, open-source compiler for Haskell that has been widely adopted and is used in many production systems.
2) Facebook's Haxl library is a Haskell library for efficient, concurrent data access that was developed at Facebook and is used extensively in their backend systems.
3) Pandoc is a Haskell library for converting between different markup formats, such as Markdown, HTML, and LaTeX. It is widely used in the academic community and by publishers.
4) QuickCheck is a Haskell library for property-based testing that has been widely adopted in the software industry.
5) XMonad is a tiling window manager for Linux that is written in Haskell and has a strong following in the Linux community.

IV. Key Points About Python

Python is a multi-paradigm programming language that can be used for functional programming. Some of the best examples of Python's usefulness in functional programming include:

1) Rapid prototyping. Python is a high-level language with a concise syntax that allows for quick and easy experimentation. This makes it ideal for prototyping functional

solutions before implementing them in a lower-level language.
2) Easy to learn. Python's simple syntax and dynamic typing make it easy to learn and use, even for beginners. This makes it a great language for introducing functional programming concepts to new programmers.
3) Large standard library. Python comes with a large standard library that includes many functional programming constructs, such as map, filter, and reduce. This allows developers to write functional code without needing to import additional libraries.
4) Functional libraries. Python has several functional programming libraries, such as functools and itertools, that provide additional tools for working with functions and data.
5) Data analysis and scientific computing. Python is widely used in data analysis and scientific computing, where functional programming techniques are often used to process large datasets.

Similarly, A few places where Python Excelled ever since it originated:

1) NumPy is a Python library for scientific computing that provides support for large, multi- dimensional arrays and matrices. It is widely used in data analysis and scientific computing.
5) Flask is a lightweight web framework for Python that supports functional programming concepts, such as decorators, which allow developers to write modular and reusable code.

Even though Haskell and Python have their differences, they are both programming languages that support functional programming.

V. Differences Regarding the Functional Program Aspects of Haskell and Python

1) Type system. Haskell has a strong, static type system that enforces type safety and reduces the likelihood of run-time errors. Python, on the other hand, has a dynamic type system that allows for more flexibility and ease of use but can lead to more run-time errors.
2) Immutability. In Haskell, data structures are immutable by default, meaning that once a value is created, it cannot be modified. This makes it easier to reason about code and eliminates many common sources of bugs. Python supports immutability, but it is not enforced by default.
2) Pandas is a Python library for data analysis that provides support for data manipulation, analysis, and visualization. It includes functional programming constructs, such as map and filter, that make it easier to work with data.
3) PySpark is a Python library for distributed computing that provides support for functional programming constructs, such as map, filter, and reduce, to process large datasets.
4) Django is a Python web framework that supports functional programming concepts, such as function-based views, which allow developers to write web applications using functional programming techniques.
3) Lazy evaluation. Haskell is a lazily evaluated language, meaning that expressions are not evaluated until they are needed. This can lead to more efficient code and better performance for certain types of applications. Python is not lazily evaluated, but it does support lazy evaluation through libraries like itertools.
4) Recursion. Haskell is optimized for recursion, which is a key feature of functional programming. Python supports recursion, but it is not optimized for it and can lead to performance issues for deeply recursive functions.
5) Functional constructs. Haskell has built-in support for functional programming constructs like monads, functors, and higher-order functions. While Python also supports these constructs, they are not built into the language and require additional libraries to use.

Overall, Haskell is often seen as a purely functional language that is better suited for large-scale applications that require strong type safety and high performance. Python, on the other hand, is often seen as a more flexible language that is better suited for rapid prototyping, data analysis, and scientific computing.

VI. Conclusion

To conclude, Haskell and Python have their strengths and weaknesses as functional languages. Haskell is known for its strong static typing system, lazy evaluation, and emphasis on higher-order functions, while Python is known for its simplicity, readability, and vast ecosystem of libraries and frameworks. Haskell may be faster and more reliable for specific tasks, while Python may be easier to learn and more flexible for others. Ultimately, the choice between these two languages depends on the project's requirements and the programmer's preference.

Improving Optimization Algorithms in Maximum Likelihood Fits

Abhigyan Acherjee, *B.S., Comp. Sci.*

***Abstract*—Optimization and maximum likelihood fits are an important component of particle physics and machine learning, offering key insights. Minuit is one such algorithm, included in the MINUIT software package, used to find the minimum value of a multi parameter fit function and analyze the shape of the function around the minimum. While Minuit works very well for simple functions with a small number of parameters, for complex functions with many parameters and/or constraints, it does not always converge well, or it converges very slowly. This paper investigates methods of improving the algorithm to conduct faster fits through a variety of empirical methods and in the future non-gradient descent methods. To that end, we have developed a test suite for examining the behavior of Minuit through the test suite.**

I. Introduction

A Large class of problems in many different fields of research can be reduced to the problem of finding the smallest value taken on by a function of one or more variable parameters [1]. Examples come from fields such as industrial processing (minimization of production costs) and machine learning-such as cost functions and chi squares. This problem is referred to by different names for different problems. Depending on the field, it may be called minimization, maximizing likelihood, or maximizing efficiency. Here for the purposes of minimizing ambiguity, we will refer to the problem at hand as function minimization.

II. Background

Minuit was conceived as a tool to find the minimum value of a multi parameter fit function and analyze the shape of the function around the minimum [2]. In our project, we were more focused on finding the parameters of the cost function that produced the minimum. The main purpose is foreseen for statistical analysis, working on chi square or log-likelihood functions, to compute the best-fit parameter values and uncertainties, including correlations between parameters. It is especially suited for handling difficult problems, including those which may require guidance to find the correct solution [3].

III. Methodology

Minuit uses a form of gradient descent to compute the minima. For gradient descent, the algorithm starts at an arbitrary point and computes the derivative at that point, and uses the tangent line to evaluate the steepness of the slope. The slope will inform updates to the parameters using weights and bias. The slope at the starting point will be steeper, but as new parameters are generated, the steepness reduces until it reaches the lowest point on the curve known as the point of convergence [4]. Gradient descent needs two things: the learning rate and a cost function. The learning rate is the size of steps taken to reach the minima. A higher learning rate means larger steps, but risks overshooting the minima while a smaller learning rate might cause significant inefficiencies. The cost function, referred to as FCN here, measures the difference between the actual y and the predicted y at the current position. It continuously iterates, moving along the direction of steepest descent [5]. Minuit has been in use since the 70's as a tool of choice for optimization of fit functions. As a result, a set of assumptions have developed about its behavior. Before we delve into that, we would like to define the use of the following keywords:

- Experiment: A singular convergence-this includes the list of iterations it takes for Minuit to find the converged values for the parameters in question. Iteration: A single calculation of the FCN during a convergence. Depending on the cost function and the size of the dataset, the number of iterations vary. Eg: for a simple linear fit, a convergence involves 33 iterations.
- Iteration: A single calculation of the FCN during a convergence. Depending on the cost function and the size of the dataset, the number of iterations vary. Eg: for a simple linear fit, a convergence involves 33 iterations.
- Generation: A set of FCN calculations using parameters that are close to each other, used to calculate derivatives numerically.

The terms are explained more widely in the figures and in the appendix.

To investigate the behavior of Minuit, we have developed a test suite of code which is used to compute a set of variables which can be used for a. verifying a set of behaviors we suspect Minuit exhibits and b. testing a set of hypotheses regarding how we intend to optimize it. The former includes the following assumptions:

(a) Increasing the number of parameters increases the number of generations.
(b) Increasing the number of datapoints increases the number of generations.
(c) Fit parameters fairly close to the final values could be achieved with a smaller subset of the data.
(d) Using fit parameters obtained by running Minuit on a

subset of the data, as starting values for a fit on the entire data, we can achieve a significant reduction in runtime.

The test suite is being used to test the aforementioned hypotheses on a variety of “simpler fits” ranging from 2-6 parameter functions (i.e quadratic to quintic cost functions). We intend use the results we get from the behavior of Minuit on these fits on functions that are more commonly encountered while dealing with problems in particle physics. To that end, we have devised a set of variables that we utilize in the test suite-these include multiple measures of timing the fits, looking at the variation in the FCN calculations over different iterations, and testing on an 8 parameter function to somewhat reflect the ”complexity” of fits usually encountered by Minuit. We are also utilizing “pull values” for sanity checks as they are a commonly used statistical concept used to measure the significance of a signal. It is given by:

$$Pull\ Value = (V_{observed} - V_{expected})/\sigma \tag{1}$$

where sigma is the associated uncertainty and V is the value of a fit parameter. As a general rule, pull values centered around 0 with an average rms value of 1 indicate that the parameter values are not skewed.

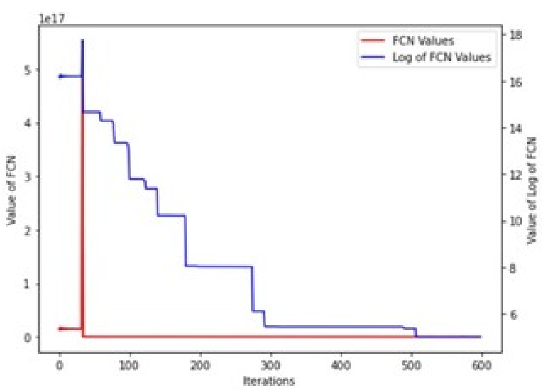

Fig. 1: Log of FCN (in red) and FCN (in blue) versus number of iterations for an 8-parameter function fit.

IV. Results and Discussion

The utility of the testing suite yielded a set of results which were significant for both verifying our hypotheses and some results which could be useful in terms of reducing the time taken for an experiment. They are as follows: i. For linear, quadratic, and cubic fits, time to converge on an average does increase by a factor of n on increasing the number of data points by a factor of n. ii. As the complexity of the fit model increases, the precision of each parameter fitted decreases. iii. The pull distributions for functions with 2-5 parameter fits are centered around 0 with an average rms width of 1, indicating events are unbiased and error estimates are accurate. iv. Using parameter fit values from a subset of the data as the starting values for fitting the entire data on an 8-parameter function, we were able to reduce the runtime of the fit by 38%.

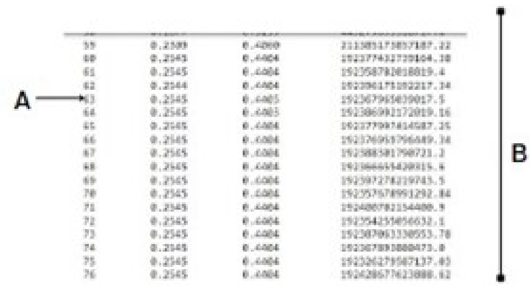

Fig. 2: Here, A represents a single iteration, while B represents a set of iterations which form the entire fit.

V. Conclusion

Given the results that we have so far, the next steps would involve taking our code suite and running the fits on dedicated GPUs without overheads from background processes. This would give us reproducibility and also give us more accurate time measurements. We also intend to the strategy of performing fits on increasingly smaller datasets and using those as starting values for a larger dataset.

VI. Acknowledgements

I would like to thank Dr. Michael Sokoloff and Dr. John Gallagher for their help, advise and constant encouragement through the duration of this project. This would not have been possible without their help.

References

[1] F. (n.d.). James. *Minuit: Function Minimization and error analysis - CERN.root.cern.ce.* 1998. URL: https://cds.cern.ch/record/2296388/files/minuit.pdf.

[2] J. F. (n.d.) Goldstein A. A. Price. “An effective algorithm for minimization -numerische mathematik”. In: (1966). URL: https://link.springer.com/article/10.1007/BF02162162.

[3] (n.d.) *iminuit user guide. - iminuit 2.21.3 compiled with ROOT-v6-25-02-2017-gd0b406db5e documentation.* ROOT Project. URL: https://iminuit.readthedocs.io/en/stable/index.html.

[4] CERN. “Determining the statistical significance of experimental results”. In: (1981). URL: https://cds.cern.ch/record/1050342?ln=en.

[5] (n.d.) “What is gradient descent?” In: (). URL: https://www.ibm.com/topics/gradient-descent.

ACSes: Ant Colony Semantic Search of Vector Encoded Documents

Arnav Chandra Komaragiri, *B.S., Comp. Sci.,* Robbie Schad, *B.S./M.S., Comp. Sci.*

***Abstract*—In this manuscript, we present ACSeS, a modification of Ant-Colony Optimization applied to vectors to allow for self-organization and search of BERT embedding vectors. ACSeS applies the original mechanisms of Ant-Colony optimization to vectors, simulating ants moving through a toroidal lattice network depositing their pheromone vectors. By this mechanism, we observe ants forming fractal spanning trees across the network for specific topics, along with shared pathways and dedicated "hot and cold zones" emerging from ant interactions. All code can be found at https://github.com/arnavkomaragiri/acses/tree/acses**

I. Introduction

SEMANTIC analysis of text has been a prominent interest of research in recent years, especially with the emergence of Transformer Networks and Large Language Models (LLMs) in Vaswani et al. [1]. With the advent of Transformer Networks, deep semantic analysis of text has become possible on a much wider scale, with work like Touvron et al. [2] offering the ability to run LLMs even on laptops. With this, works like Touvron et al. [2] provide the potential to apply these models to document embedding and vector analysis, encoding the meaning of a document into a N-dimensional vector. However, analysis of these vectors has conventionally only been done in a centralized fashion, ignoring other approaches from complex adaptive systems. In this manuscript we propose ACSeS, an application of Ant-Colony Optimization [3] on publicly accessible news datasets (Misra et al.) [4][5], using Siamese BERT Networks from Reimers et al. [6] to encode documents into vector representations. The goal of this approach is for self-organized semantic search; we hope to introduce a decentralized algorithm for document search and storage across a lattice network, one that is resistant to noise, and is capable of self-organizing different kinds of information. We use Fernandes et al. [7] as an initial framework, however we make a number of modifications to this algorithm to respect context-specific goals and properties of the target data distribution.

Our goal by applying this algorithm to document embeddings is to demonstrate the efficacy of applying complex systems-based approaches to modern data distributions through applications of modern NLP advancements. The use of complex systems in the organization of this data offers massive potential, including applications like self-organizing search systems, semantic analysis of topic connectivity in news articles, and a framework with which to organize vector-encoded information in an active learning context. We hope this manuscript begins to deliver on these goals, and may serve as a foundation for future application of complex systems to modern NLP data.

II. Prior Work

A. KohonAnts Ant Clustering and Pattern Classification

KohonAnts was a publication from Fernandes et al. [7] initially applying Ant-Colony Optimization (ACO) [3] to vector encodings, demonstrating the potential of ACO-based methods for self-organization and classification of vectors. The algorithm was initially applied on rudimentary tabular data, but could theoretically be applied to arbitrary vector encodings with some modifications. Much like the original Ant-Colony Optimization algorithm [3], the Fernandes et al. [7] approach consisted of the following steps:

- Ant Decision Making
  - Ants decide where to go based on current pheromone landscape
- Pheromone Deposition
  - Ants deposit their own specific pheromone on their environment

1) Ant Decision Making: The Ant Decision Making process in Fernandes et al. [7] essentially allowed each ant to examine its surrounding neighbors and identify which neighbor to traverse to. With that, let's define some common notation. We use $\vec{a}_k$ to denote the ant's respective vector, and n_k to denote the ant's current node. We also denote $N_{k,r}$ to be the neighborhood reachable within r steps around n_k. With this, the pheromone vector at node n_i is defined as $\vec{v}(n_i)$. We also define the following functions for all ants, the centroid pheromone rule, the ant weighting rule, and the edge pheromone rule:

$$CTR(n_i) = \frac{1}{|N_{i,r}|} \sum_{n \in N_{i,r}} [\vec{v}(n)]$$

$$W(\sigma) = (1 + \frac{\delta}{1 + \sigma\delta})^{\beta}$$

$$\sigma(\vec{v}, n_j) = \sqrt{(\vec{v} - CTR(n_j))^T(\vec{v} - CTR(n_j))}$$

These rules were defined as follows:

- First, an ant randomly decides if it will be greedy with probability q
- If an ant is being greedy, the neighbor to move to is decided via the following:

$$n_k = \arg\max_{n \in N_{k,r}} [W(\sigma(\vec{v}(n_k), n))d(\Delta n)]$$

Note that $d(\Delta n)$ denotes a penalty for changing directions.

- If an ant is not greedy, the neighbor to move to is decided via a weighted roulette wheel, with the probability of candidate neighbor n_c being chosen defined as follows:

$$p(n_c|n_k) = \frac{W(\sigma(\vec{v}(n_k), n_c)d(\Delta n_c)}{\sum_{n \in N_{k,r}}[W(\sigma(\vec{v}(n_k), n)d(\Delta n)]}$$

2) Pheromone Deposition: The Pheromone deposition process deposits some scalar multiple of the ant's vector onto the given node, essentially making the node's pheromone a combination of all the ants that have passed through it. The pheromone deposition rule operates by the following equation:

$$\vec{v}(n_k) = \vec{v}(n_k) + R * [\vec{a_k} - \vec{v}(n_k)]$$

where the reinforce R is defined as:

$$R = \alpha * (1 - \bar{D}(\vec{a_k}, n_k))$$

and:

$$\bar{D}(\vec{v}, n_i) = \frac{\sigma(\vec{v}, n_i)}{|\vec{v}|}$$

The function of the reinforce R here is especially critical, as this prevents ants from overwriting each others' trails. Should $R \neq 0$ when there is no match between the current ant and its node, then ants can potentially overwrite each other and lead to destructive pheromone updates.

Finally, a pheromone evaporation stage is applied at the end of every step to decay the network back to the initial configuration, defined as follows:

$$\vec{v}(n_k) = \vec{v}(n_k) - (1 - \rho)\vec{v}^{(0)}(n_k)$$

, where $\vec{v}^{(0)}(n_k)$ is the initial pheromone for a node.

B. Sentence-BERT Document Embedding Vectors

SentenceBERT embeddings were proposed by Reimers et al. [6] to extract vector representations of arbitrary length documents via Transformer networks. As such, these transformers are fine-tuned in a Siamese network setup, where two different documents are reduced to vector representations then classified using a softmax classifier. The resulting fine-tuned transformer networks are then used to extract vector representations from documents which can be compared via inverse-cosine distance. We do not further explore the nature of this setup as this was not significantly modified in the design of ACSeS; the SentenceBERT embeddings were largely just used for feature extraction with little work on specifics of the feature extraction process.

III. ACSeS Algorithm Definition

With the definitions of both the KohonAnts [7] and SentenceBERT [6] setups, we can move on to defining the ACSeS algorithm. Like KohonAnts [7], ACSeS has two stages:

The ant decision making process in ACSeS is fairly similar to the Fernandes et al. [7] setup, however we make a number of modifications to the algorithm. For instance, instead of using the vector of the current node in all computations for ant decision making, we instead use the ant vector. Additionally, if an ant is being greedy but no other neighbor is as good as the current node, the ant stops moving. This is to provide ants with a stop condition, something Fernandes et al. [7] does not explicitly do. Finally, we switch the KohonAnts [7] algorithm from using Euclidean distance to compute σ to Inverse Cosine distance, since this metric more closely aligns with the SentenceBERT setup [6]. These changes yield the new updated ant decision making process: First we define our distance function:

$$\sigma(\vec{v}, n_j) = 1 - \frac{\vec{v}^T CTR(n_j)}{||\vec{v}|| * ||CTR(n_j)||}$$

Then we define our algorithm:

- First, an ant randomly decides if it will be greedy with probability q
- If an ant is being greedy, a candidate neighbor is decided via:

  $$n_k = \arg \max_{n \in N_{k,r}} [W(\sigma(\vec{a_k}, n))d(\Delta n)]$$

  - If $W(\sigma(\vec{a_k}, n_c)) < W(\sigma(\vec{a_k}, n_k))$, stop the ant at n_k
  - Else, move to candidate neighbor n_c

  Note that $d(\Delta n)$ denotes a penalty for changing directions.
- If an ant is not greedy, the neighbor to move to is decided via a weighted roulette wheel, with the probability of candidate neighbor n_c being chosen defined as follows:

  $$p(n_c|n_k) = \frac{W(\sigma(\vec{a_k}, n_c)d(\Delta n_c)}{\sum_{n \in N_{k,r}}[W(\sigma(\vec{a_k}, n)d(\Delta n)]}$$

The pheromone deposition process is also slightly different, being defined as follows:

$$\vec{v}(n_i) = \vec{v}(n_i) + \theta(n_k, n_i) * R^u * [\vec{a_k} - \vec{v}(n_i)], n_i \in N_{k,r}, u \approx 3$$

where the reinforce R is defined as:

$$R = \alpha * (1 - \bar{D}(\vec{a_k}, n_c))$$

and:

$$\bar{D}(\vec{v}, n_i) = \frac{\sigma(\vec{v}, n_i)}{2}$$

. We make a core change in the pheromone deposition process by applying a power function to the reinforce prior to scaling the pheromone step by it. This has the effect of ensuring that not every ant will impact the pheromone; only significant matches of ants will deposit pheromones enough to significantly change the pheromone vector. Note that we also introduce a neighborhood function $\theta(n_k, n_c)$ as a neighborhood function similar to a standard Self-Organizing Feature Map [8], defined as follows:

$$\theta(n_k, n_i) = b^{-s(n_k, n_i)}, b \approx 4$$

Note that $s(n_k, n_c)$ is some distance function between nodes n_k and n_i and n_i is any node in the immediate neighborhood of the updated node, or $N_{k,r}$.

Finally, we use an updated pheromone evaporation rule designed to properly evaporate the pheromone vectors to their initial configuration, defined as follows:

$$\vec{v}(n_k) = \rho\vec{v}(n_k) - (1 - \rho)\vec{v}^{(0)}(n_k)$$

, where $\vec{v}^{(0)}(n_k)$ is the initial pheromone for a node.

IV. Experiments

A. Setup

To test the algorithm, we applied the ACSeS algorithm to the Huffington Post News Caption Dataset [4]. We first apply SentenceBERT with the pretrained all-mpnet-base-v2 to a combination of the article headline and short description separated by '–'. We then run this string through the SentenceBERT embedding to get our document embeddings. Next we balance the dataset by categories, ensuring an equal number of documents from each category are present. Finally, we randomly select an even number of documents from each category such that the number of documents sums to the number of initial ants on the network.

To initialize the setup, we randomly assign one ant an embedding vector from our set of precomputed embedding vectors, then randomly place an ant at a random node in the network. We then do this for every ant we are running, thus ensuring every embedding vector in our dataset is sent through the network at least once. Finally, we run the ants for N steps, randomly shuffling the order of ants to process at each step and updating each ant in the new randomized order. When ants stop, we either initialize them with their same topic vector at a new location if they did not make sufficient steps (25 by default) or we give them a new topic vector and re-initialize the ant at a new position. This results in documents being sent through the network multiple times, simulated repeated queries for specific content or ideas.

Additionally, we "erode" the network every 50 timesteps by deleting any documents at a given node if their embedding vectors don't properly match the pheromone vector. This is done to ensure that documents are not rendered "out-of-date" by decaying pheromones, ensuring document storage remains versatile. Finally, we stop ants when they receive two consecutive stop signals, confirming that the ant is truly in a good position and allowing for some persistence in ant traversals. Once ants are stopped, we deposit the ant's document at the node until the document is eroded away.

For more specifics, in the following results we run the system with 5000 ants for 200 timesteps. We use $\beta = 32, \delta = 0.2$ for ant weighting, $u = 3$ for reinforce scaling, and we use an evaporation factor $\rho = 0.995$

B. Results

When applying this algorithm, we get a variety of results, including the following:

1) Ant Age Analysis: When analyzing both the ant age histogram (Fig 1) and the ant age log-log plot, we notice that the ages of ants remain relatively bounded, avoiding many of the pitfalls that emerge with fat-tailed distributions. Specifically, we note that these plots imply that the ant age follows a geometric distribution, which means that the average ant age exists and can be bounded. This potentially offers bounding ant ages, ensuring that search results are achievable within a reasonable number of steps. From the existing results, we observe that ants mostly only make ¡100 steps, indicating that the ants only explore, at most, 10% of the network. However, we observe that ants are still able to find meaningful matches in the network, indicating that the pathways built are either primarily local or very efficient in finding documents.

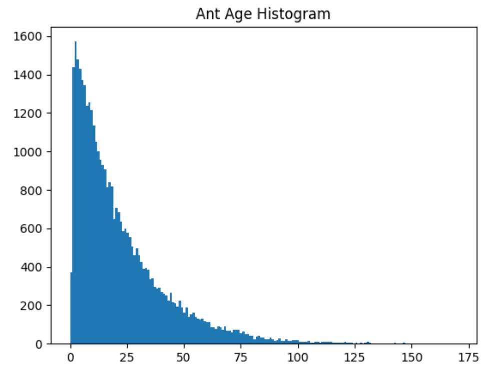

Fig. 1. Ant Age Histogram

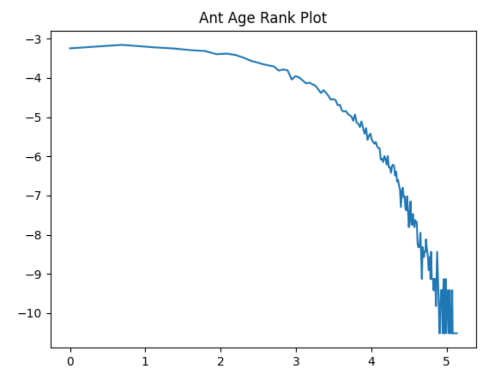

Fig. 2. Ant Age Log-Log Frequency Plot

2) Network Pheromone Maps: As a secondary visualizer, we generate the pheromone match between a document vector for some given prompt ("us political news" in this case) and every node in the network. This is done to reveal any pathways that the ants form to guide other ants from any node in the network to the documents. Specifically, we compare the network's pheromone match with the prompt "us political news" and the document "It Just Got Easier For Detroit Students To Pay For College – Some rare good news for kids in the struggling school district" [9].

The primary observation we see in these pheromone maps are the types of connections that ants forge. We see with the document's activity map on the left, Fig 4, that documents are quickly clustered into a handful of extremely good matches, with other regions having low matches simply designed to serve as pathways. However, for the more general prompt of "us political news," the activity map is much more active, with fractal trails connecting "cold zones" on the network to "hot zones" where documents are stored. These trails are highly connected since the system prefers to build webs of connectivity instead of trees. This is likely due to maintain stability on the network, so that if a bad ant step is made ants can still be guided back to their relevant document and in turn

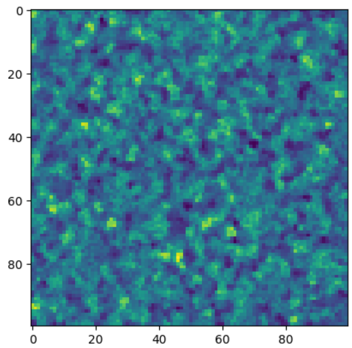

Fig. 3. US Political News Pheromone Match Heatmap

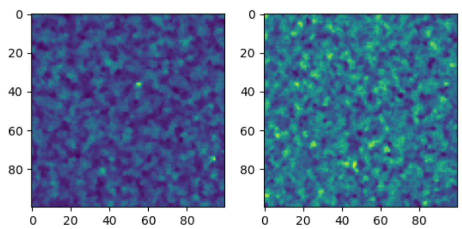

Fig. 4. Detroit Students vs US Politics Pheromone Comparisons

lay down new trails. Some trails are shared between the two topics but, ultimately, the trails don't completely match as the ants represent two different topics. Further work is needed to analyze the impact of the various hyperparameters on these pheromone trails.

3) Test Search: Finally, we test the performance of this algorithm as an actual search engine. We randomly initialize an ant in the network with the document "us political news" and the appropriate embedding vector extracted from SentenceBERT [6].

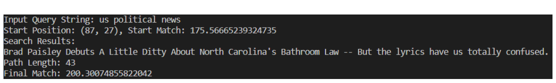

Fig. 5. Search Results for "US Political News"

We see here that while the search query shown in Fig 5 returns a limited number of results, the match of this algorithm is relatively good. The returned document mentions topics about transgender rights in the US [10], which is something of great political importance in current times. This is also numerically reflected in the pheromone matches, with the initial pheromone match being 175.56 and the final pheromone match being 200.37. Finally, we see that the ant was able to find this result in only 43 steps, only having to traverse .43% of the network to find a satisfactory match.

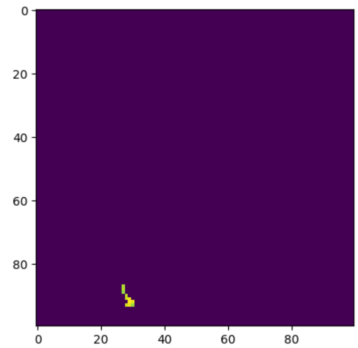

Fig. 6. Ant Path Colored by Pheromone Match

These patterns are also reflected in the ant's path. As seen in Fig 6, the ant takes a relatively fast path to a zone where a good match can be found. Following this, the ant appears to "jitter" around until it finds the appropriate document, likely a consequence of not every node in the network possessing a document. However, these initial results are fascinating in that they illustrate that the network's built structure is effective for document retrieval. Further analysis is needed on the distribution of final pheromone matches and search relevance, but the initial results are promising.

V. Conclusions and Next Steps

In conclusion, we see that ACO-based methods [3][7] are a viable approach to analyze SentenceBERT [4] topic vectors, despite the high dimensionality and complexity of these topic vectors.

However, we emphasize that this algorithm is still very much a work in progress; ants are still very naive about the paths they choose and the evolution of this network over time has yet to be analyzed. Future work may focus on the impact of the ant stop conditions, neighborhood pheromone deposition rules, and more intelligent analysis of candidate neighbors based on potential matches beyond the immediately observable candidate neighborhood. Ultimately though, we hope that this approach demonstrates the efficacy of complex-systems inspired approaches for self-organization in complex data distributions, and can be a new approach for decentralized self-organization of language and documents.

References

[1] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[2] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL].

[3] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization". In: *IEEE Computational Intelligence Magazine* 1.4 (2006), pp. 28–39. DOI: 10.1109/MCI.2006.329691.

[4] Rishabh Misra. "News Category Dataset". In: *arXiv preprint arXiv:2209.11429* (2022).

[5] Rishabh Misra and Jigyasa Grover. *Sculpting Data for ML: The first act of Machine Learning*. Jan. 2021. ISBN: 9798585463570.

[6] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL].

[7] C. Fernandes et al. *KohonAnts: A Self-Organizing Ant Algorithm for Clustering and Pattern Classification*. 2008. arXiv: 0803.2695 [cs.NE].

[8] T. Kohonen. "The self-organizing map". In: *Proceedings of the IEEE* 78.9 (1990), pp. 1464–1480. DOI: 10.1109/5.58325.

[9] Rebecca Klein. *It just got easier for Detroit students to pay for college*. Mar. 2016. URL: https://www.huffpost.com/entry/detroit-promise-zone-college-tuition_n_56f447cee4b014d3fe22a1a7.

[10] Noah Michelson. *Brad Paisley debuts a little ditty about North Carolina's bathroom law*. May 2016. URL: https://www.huffpost.com/entry/brad-paisley-transgender-bathrooms-north-carolina_n_5745ba1ee4b03ede4413659d.

The Effects of Noisy Gates on the Quantum Teleportation Protocol and Grover's Algorithm

Alexander Shiveley, *M.S., Comp. Sci.*

***Abstract*—In this work, we study the effects of the presence of noise in the sequence of gates used both in the quantum teleportation protocol and Grover's algorithm. The reliability of both algorithms is studied by computing the fidelity factor which tests the amount of overlap between the final single state or multiple states, as calculated with and without noisy gates.**

I. Introduction

QUANTUM teleportation is a protocol for transferring information of a quantum bit from one location to another. Rather than physically transferring the qubit itself, teleportation takes advantage of an entangled quantum bit pair, a Bell state, to transfer information from the initial quantum bit to the entangled quantum bit of the receiver. The quantum teleportation protocol was explored in 1993 and then experimentally realized in 1997 [1][2][3]. Grover's algorithm is a quantum algorithm to search databases by applying a sequence of quantum gates repeatedly based on the size of the database. The advantage of Grover's algorithm is only $O(\sqrt{N})$ steps are required while for a classical algorithm, $O(N)$ steps are required to search the same database [4][5].

In this work, we study the effects of the presence of noise in the sequence of gates used both in the quantum teleportation protocol and Grover search algorithm. Noisy gates are introduced through modifications of the unitary operators representing their action on a single or a collection of qubits. The effects of noisy gates are simulated in a MATLAB environment and as a function of parameters to allow tuning the strength of the noise independently in each of the gates. We study the reliability of the quantum teleportation algorithm for the four different initial Bell states of two qubits. We illustrate the variation in the location of the final qubit on the Bloch sphere as the number of noise increases in the individual gates. For Grover's algorithm, we study the effects of noise in the individual Hadamard gates and phase gates used to implement imperfect phase inversions in the Grover iteration operator. We study the reliability of the Grover search algorithm as a function of the size of the database up to a maximum of 13 qubits. The reliability of the quantum teleportation protocol and search algorithm is studied by computing the fidelity factor which tests the amount of overlap between the final single state or multiple states, as calculated with and without noise gates [6],

$$Fidelity = |\langle\psi_{noisy}|\psi_{noiseless}\rangle|^2 \quad (1)$$

The distribution of the final states $|\psi_{noisy}\rangle$ are plotted onto a Bloch sphere as a function of the amount of noise. We also make histograms of the fidelity factor for a large number 10^4 of applications of both algorithms. Average values of the fidelity factor are then calculated. The decrease in the average fidelity factor from an ideal value of 1 is assessed as a function of the amount of noise.

II. Results and Discussion

For the quantum teleportation algorithm, each instance of the Pauli X, Pauli Z, Hadamard, and CNOT gates is generated with randomly modified parameters using the rand function provided by the MATLAB library. For each application of the algorithm, the gates of the circuit as shown in Figure 1 will be different and may affect the final state.

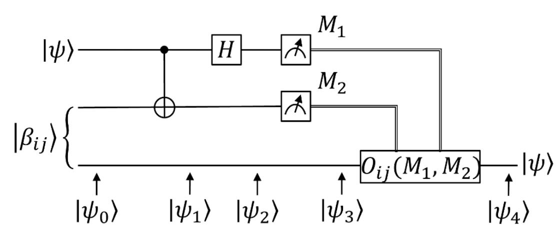

Fig. 1. Teleportation Circuit Diagram using any initial Bell State; $O(i, j)$ is the sequence of gates which must be applied to teleport the qubit $|\psi\rangle$ for an arbitrary Bell state $|\beta_{ij}\rangle$ defined by Table 1. Based on the circuit from Nielsen and Cheung [7].

TABLE I
Operators based on the Bell state applied to $|\psi_3\rangle$ in the circuit defined by Figure 1. The power of each gate is an expression of logical operators.

Bell state $|\beta_{ij}\rangle$	$O(i, j)$ operator
$|\beta_{00}\rangle$	$Z^{M_1}X^{M_2}$
$|\beta_{01}\rangle$	$Z^{M_1}X^{\overline{M_2}}$
$|\beta_{10}\rangle$	$Z^{M_1 \wedge M_2}X^{M_2}Z^{\overline{M_1} \vee M_2}$
$|\beta_{11}\rangle$	$X^{M_2}Z^{\overline{M_1} \vee M_2}XZ^{M_1 \wedge M_2}$

We then determine the average fidelity for each of the final states over the 4 measurement cases (M_1, M_2) over a range of noise values. Increasing the amount of noise has shown that average fidelity decreases in some cases or remains constant. We show that the decrease in average fidelity as a function of noise strength depends on the initial state, the Bell state used, and four different classical measurements needed to generate the teleported state.

For Grover's algorithm, noise is introduced in the Hadamard gates and phase gates U_0 and U_{query} where the diagonals of U_0 are 1 at the starting index and -1 elsewhere while the diagonals of U_{query} are -1 at the target index and 1 elsewhere.

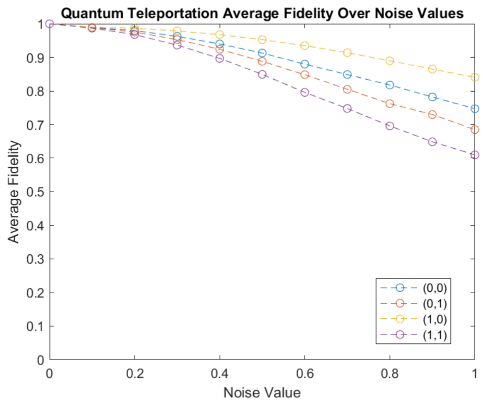

Fig. 2. Teleportation average fidelity values over noise values from 0.1 to 1.0 of $|\psi\rangle = e^{i\chi}\begin{bmatrix}\cos(\theta/2)\\ e^{i\phi}\sin(\theta/2)\end{bmatrix}$ where $\theta = 75°, \phi = 130°, \chi = 0$ and using $|\beta_{10}\rangle$ as the initial Bell state. Each series of data is for a measurement case (M_1, M_2).

$$U_0 = \begin{bmatrix}1 & \dots & 0\\ \vdots & \ddots & \vdots\\ 0 & \dots & -1\end{bmatrix} \qquad U_{query} = \begin{bmatrix}-1 & \dots & 0\\ \vdots & \ddots & \vdots\\ 0 & \dots & 1\end{bmatrix} \tag{2}$$

Using the rand function in the MATLAB library, noise is introduced by replacing -1 in U_0 and U_{query} with $e^{-i\pi(1+\epsilon)}$ where ϵ is the noise parameter. We then determine the probability of the target state after the expected number of Grover iterations of reaching the maximum probability and the fidelity factor of the final states.

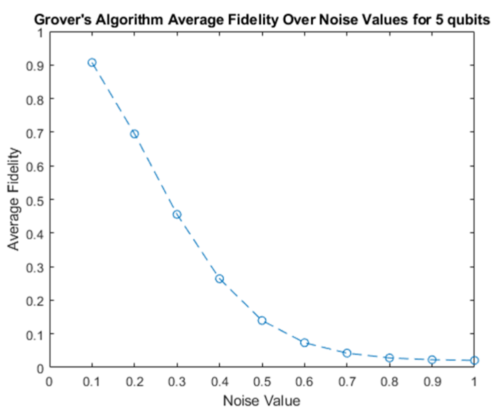

Fig. 3. Grover's algorithm average fidelity for 5 qubits from the initial state $|00000\rangle$ to the target state $|11111\rangle$

III. Conclusion

Noise in quantum computing affects the reliability of quantum algorithms. The significance of noisy gates on the quantum teleportation and Grover algorithms are explored. Preliminary results show the effects of noise on quantum teleportation depend on the quantum bit being transferred and the Bell state used. Preliminary results show the effects of noise on Grover's algorithm lower the maximum probability significantly of a successful search as the number of qubits increases.

References

[1] Charles H. Bennett et al. "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels". In: *Physical Review Letters* 70 (13 Mar. 1993), pp. 1895–1899. DOI: 10.1103/PhysRevLett.70.1895. URL: https://www.ncbi.nlm.nih.gov/pubmed/10053414.

[2] D. Boschi et al. "Experimental Realization of Teleporting an Unknown Pure Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels". In: *Physical review letters* 80.6 (Feb. 1998), pp. 1121–1125. DOI: 10.1103/PhysRevLett.80.1121.

[3] Dik Bouwmeester et al. "Experimental quantum teleportation". In: *Nature (London)* 390.6660 (Dec. 1997), pp. 575–579. DOI: 10.1038/37539. URL: http://dx.doi.org/10.1038/37539.

[4] Lov K. Grover. "A fast quantum mechanical algorithm for database search". In: (May 1996). URL: https://arxiv.org/abs/quant-ph/9605043.

[5] Lov K. Grover. "From Schrödinger's equation to the quantum search algorithm". In: *American journal of physics* 69.7 (July 2001), pp. 769–777. DOI: 10.1119/1.1359518. URL: http://dx.doi.org/10.1119/1.1359518.

[6] Hong-Yang Lin. "A Generalization of Quaternions and Their Applications". In: (June 2021).

[7] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. 10th anniversary ed. Cambridge [u.a.]: Cambridge Univ. Press, 2010. URL: http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&local_base=BVB01&doc_number=021169410&sequence=000002&line_number=0001&func_code=DB_RECORDS&service_type=MEDIA.

Systolic Arrays

Srimanth Tenneti, *M.S., Elec. Eng.*

***Abstract*—This article talks about the importance of Systolic arrays and their applications in modern-day Machine Learning. It also explains a sample matrix multiplication implementation using Verilog HDL.**

I. Introduction

SYSTOLIC Arrays are hardware elements that can perform traditional computing tasks like convolution, matrix multiplication, etc in a much more efficient manner. They intend to replace pipelined structures with processing elements that can be designed to target a specific operation.

To understand the working principle of the systolic arrays we would have to consider an example. So, for this article, we would consider a 2x2 matrix multiplier as the target operation.

II. Architecture

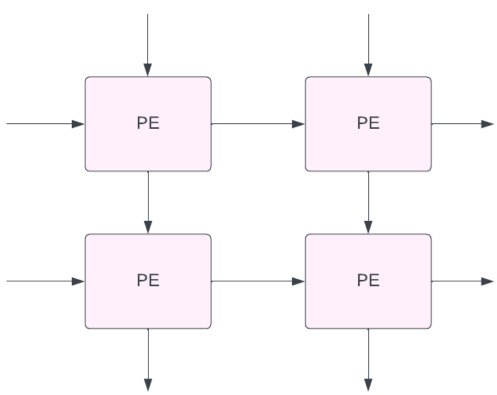

Fig. 1. 2x2 Systolic Array Architecture

In a systolic array, the processing elements are connected to one another as depicted in Fig. 1. To design a processing element (PE) for the matrix multiplication task we first start by analysing the operation. We notice that the operation can be split into 2 major subtasks:

1) Multiplication of the current bits
2) Accumulation of the previous bits

Now as we see that multiplication and addition are the key elements, we would need a multiplier and an adder designed for the data width supported by the systolic array. Each processing element would have to multiply the data it gets from the horizontal and vertical directions and then accumulate the result in its internal accumulator to achieve the two subtasks discussed above. So, from our analysis, we can conclude that the processing element would have a multiplier and an accumulator. The next major step would be to decide the data widths of these elements. Assuming we have 8-bit data to operate on the multiplier would output 16 bits and then the accumulator should be able to process 16 bits.

III. Design

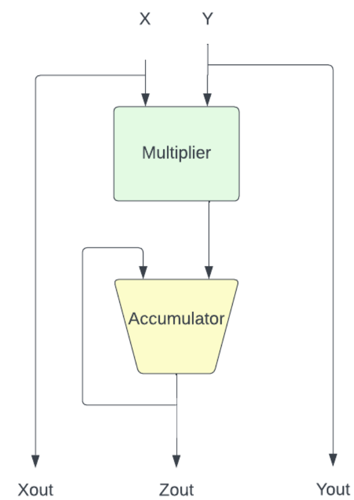

Fig. 2. Processing Element Design for Matrix Multiplication

The above figure is the processing element we designed for the matrix multiplication problem.

```
module PE #(parameter W = 8)(
  input clk,
  input rst,
  input [W - 1 : 0] X,
  input [W - 1 : 0] Y,
  output [2 * W : 0] Out,
  output [W - 1 : 0] Xout,
  output [W - 1 : 0] Yout
);

  reg [2 * W : 0] O;

  always @ (posedge clk or negedge rst)
    begin
      if (~rst)
        begin
          O <= 0;
        end
      else
        begin
          O <= O + X * Y;
        end
```

```
    end

  assign Out = O;
  assign Xout = X;
  assign Yout = Y;

endmodule
```

The code snippet above implements the proposed design in Fig. 2. To test this design, we come up with a simple testbench.

```
module PE_TEST #(parameter W = 8)();

  reg clk;
  reg rst;
  reg [W-1:0] X;
  reg [W-1:0] Y;
  wire [2*W:0] Out;
  wire [W-1:0] Xout;
  wire [W-1:0] Yout;

  initial
    begin
      X = 0;
      Y = 0;
      clk = 0;
      rst = 0;
      #10;
      rst = 1;
      forever #2 clk = ~clk;
    end

  PE pe0 (.*);

  initial
    begin
      #10; X = 8'd2; Y = 8'd2;
      #10; X = 8'd4; Y = 8'd8;
     #20; $finish();
    end
endmodule
```

The result of the simulation should look something like Fig. 3 below.

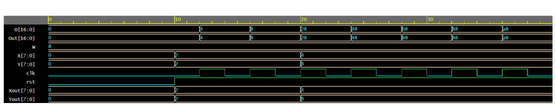

Fig. 3. PE Simulation

Now we connect the PEs in the configuration discussed in Fig. 1 to implement the 2x2 matrix multiplication array.

```
module matarray #(parameter W = 8)(

  input clk ,
  input rst ,

  input [W-1:0] a0, a1, b0, b1,
  output [2*W:0] o1, o2, o3, o4
);

  wire [W-1:0] a01, a23, b02, b13;

  PE pe0 (clk, rst, a0, b0, o1, a01, b02);
  PE pe1 (clk, rst, a01, b1, o2, , b13);
  PE pe2 (clk, rst, a1, b02, o3, a23, );
  PE pe3 (clk, rst, a23, b13, o4, , );

endmodule
```

The following code snippet implements a testbench for the above design.

```
module mattest #(parameter W = 8)();

  reg clk;
  reg rst;
  reg [W-1:0] a0,a1,b0,b1;
  wire [2*W:0] o1,o2,o3,o4;

  initial
    begin
      clk = 0;
      rst = 0;
      #10;
      rst = 1;
    end

  initial
    begin
      forever #2 clk = ~clk;
    end

  matarray ma0 (.*);

  initial
    begin
      #4 a0 = 1; b0 = 1 ; a1 = 0; b1 = 0;
      #6 a0 = 1; b0 = 1 ; a1 = 1; b1 = 1;
      #6 a0 = 0; b0 = 0 ; a1 = 1; b1 = 1;
      #4 a0 = 0; b0 = 0 ; a1 = 0; b1 = 0;
      #100; $finish;
    end

endmodule
```

The above test should generate a simulation result like the Fig. 4 below.

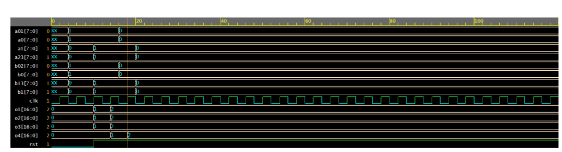

Fig. 4. Matrix Multiplication Simulation

IV. Applications

Systolic Array-based hardware elements can be found in high-performance computing systems that aim to target AI/ML, Image Processing, Data Science, and Digital Signal Processing applications.

The best example of the Systolic Array would be the Tensor Processing Unit developed by Google (Fig. 5). The TPU is used by Google Cloud and other Google services that aim to target AI/ML applications [1].

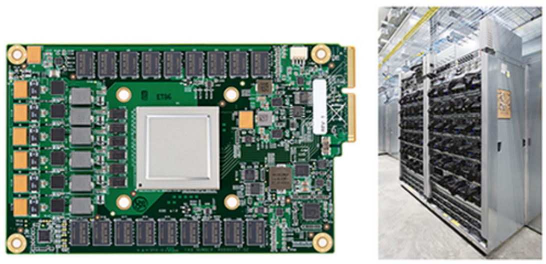

Fig. 5. Google's Tensor Processing Unit - Courtesy of Google - https://cloud.google.com/blog/products/ai-machine-learning/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

Advantages:

1) Highly regular and modular designs
2) Highly synchronized design
3) Fast and Cost Efficient

Disadvantages:

1) Memory Bottlenecks
2) Lack of proper interconnect framework
3) Synchronization issues

References

[1] *An in-depth look at Google's first Tensor Processing Unit (TPU)*. https://cloud.google.com/blog/products/ai-machine-learning/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu. Accessed: 2023-04-1.

NOTES

NOTES

NOTES