My decision to pursue graduate studies is a natural outcome of my experiences gained as a master's student at the **Indian Institute of Science (IISc)** and more recently as a research staff in the **Systems Group** at **Microsoft Research India**. I am interested in **distributed systems** and **data-intensive cloud computing** — in particular, the system-side problems associated with **learning and deploying machine learning models at scale**. My research endeavors have resulted in publications at well-known conferences such as **OSDI [1]** and **ICDE [3]**. Through graduate studies at Stanford, I wish to take the first step towards my long term goal of a career in research.

In the recent past, there has been significant interest in Graph Neural Networks (GNNs)—neural networks that operate on graph structured data. Contemporary systems for **distributed GNN training** have been heavily influenced by, often building upon, existing systems for DNN training and graph processing systems. While in-principle this might look as the most natural path forward, retrofitting algorithms and leveraging tradeoffs designed in context of DNN training and/or graph processing systems lowers compute resource utilization due to communication stalls and more importantly lead to scalability bottlenecks. Based on this, I worked with **Anand Iyer** to build the **P^3** [**1**] system, which enables efficient distributed training of GNNs on large input graphs. P^3, published in **OSDI 2021**, effectively eliminates communication stalls by independently partitioning the graph along feature and structural dimension; unlike just the structural dimension, as done by most partitioners (e.g., METIS) designed for graph processing systems. Such independent partitioning scheme combined with intra-layer model parallelism and pipelining enables new push-pull based distributed training strategy that can outperform previous data parallel methods, achieving low communication and partitioning overhead, and high resource utilization. We are currently integrating pipelined push-pull parallelism in Microsoft's DeepGraph GNN training engine to train state-of-the-art models with thousands of GPUs.

More recently, I have been looking into **model serving systems**. Large-scale pre-trained language models such as BERT have brought significant improvements to NLP applications but at the cost of heavy computational burden. A growing body of work aims to alleviate this by exploiting the difference in the classification difficulty among samples and early-exiting at different stages of the network. Although early-exit deep neural networks (EE-DNN) have demonstrated promising accuracy–latency trade-offs, we observe that the coarse-grained batching – an essential technique to increase throughput, becomes suboptimal in effectively ensuring high hardware utilization as samples dynamically exit at different stages of the network. Thus, with **Anand Iyer**, I have proposed **SURGEON [2]**, a system which takes EE-DNN model, throughput and latency requirements as inputs, and produces a model partition and service assignment across heterogeneous resources that satisfies SLA constraints using as few GPUs as possible. Key idea in SURGEON is to consolidate two or more batches at partition boundary, thereby creating a larger batch which improves hardware efficiency. SURGEON uses dynamic programming to adaptively identify when/how EE-DNN layers are to be partitioned and which/how-many devices are required to service each partition. I am currently evaluating the benefits of SURGEON for different EE-DNN model architectures and service-level objectives.

Before MSR, I completed my master's (by research) in Computer and Data Systems at the **Indian Institute of Science (IISc), Bangalore** advised by **Yogesh Simmhan**. At IISc, my thesis focused on system-side optimizations for **distributed temporal**[1] **graph analytics**. Despite their growing availability, existing abstractions and frameworks do not scale well due to redundant computing and/or messaging across time-points. We address this gap through **GRAPHITE [3]**, which uses time-interval as the data-parallel unit of computation. GRAPHITE relies on our novel time-warp operator, which automatically partitions a vertex's temporal state, and temporally aligns and groups messages to these states. This eases the temporal reasoning required by the user logic, and avoids redundant execution of user logic and messaging within an interval to provide key performance benefits. This work was published in **ICDE 2020** and is currently used at IISc, IIT-Jodhpur, and IIIT-Hyderabad for contact tracing and analytics over temporal graphs[2].

[1] Graphs whose structure and attributes evolve over time

[2] https://covid19.iisc.ac.in/gocoronago-contact-tracing-app-and-network-analytics/

WAVE [4], an extension of GRAPHITE, incorporates **dependency-driven incremental processing** by tracking dependencies to capture how intermediate values get computed, and then uses this information to incrementally propagate the impact of change across intermediate values. WAVE was presented at the **2nd ACM Student Research Competition (Graduate Category)** co-located with SOSP 2019. Out of the 15 participating submissions, WAVE was one of the 5 finalists and subsequently awarded the Bronze Medal.

I realize the need for a strong theoretical foundation to pursue advanced research. In this direction, I have always striven for academic excellence - I stood top of the class during both bachelor's and master's studies. My time at IISc offered me opportunities to assist with two graduate courses, and participate in Artifact Evaluation Committee (AEC)[3] and Shadow Program Committee[4] at several conferences. These experiences have taught me how to organize and articulate ideas more effectively, participating in faux-PC discussion showed me how I can better interpret the subtext of reviews and comments before/after rebuttal, and seeing the difference between the submitted papers I reviewed and the accepted papers presented at EuroSys 2021 gave me insight into what the authors may have learned from reviewer feedback. The time I spent in the industry prior to and after attending graduate school helped me earn valuable soft skills - time management, team work and co-operation, which I believe are vital for survival in tough academic settings.

I believe my experience with data-intensive systems at Microsoft Research has offered me a unique perspective into practical problems experienced by developers and cloud operators in deploying, operating, and monitoring computer systems at scale, and makes me uniquely qualified to tackle the big picture questions in this area. At Stanford, I would like to continue to work on efficient **systems infrastructure and tooling for emerging data-intensive workloads**. As machine learning models become larger and are increasingly used in safety and performance critical applications, demand for compute grows, and hardware accelerators evolve rapidly, I see this as an exciting area to work on from a number of angles such as performance, resiliency, resource-efficiency, and/or affordability. Stanford's demonstrated leadership in data-intensive systems research, its exemplary faculty, as well as unique inter-disciplinary atmosphere make it the ideal environment for me to conduct my graduate study.. **Prof. Matei Zaharia**, **Prof. Christos Kozyrakis**, and **Prof. Jure Leskovec** are faculty I would especially like to work with. Over the years, I have been influenced by several of their research works on new approaches and programming models for cloud-computing, frameworks for efficient deep learning training, and real-time model serving (e.g., Spark, Snorkel, Shinjuku, PipeDream, ROC, INFaaS, GNNAutoScale, GraphSAGE), and some of the insights I have been able to draw from these works have become a natural part of my master's thesis and research papers. An opportunity to work with them, is extremely appealing to me as a budding systems researcher.

In summary, I believe I bring with me research experience, industry-sharpened programming and soft skills, and above all, an insatiable desire to learn and excel. I look forward to the next milestone in my life – a PhD in Computer Science from Stanford.

[1] Swapnil Gandhi, Anand Padmanabha Iyer, "P3: Distributed Deep Graph Learning at Scale", In proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation (**OSDI 2021**), July 2021. https://swapnilgandhi.com/papers/p3-osdi21.pdf

[2] Swapnil Gandhi, Anand Padmanabha Iyer, "SURGEON: Fast & Efficient DNN Inference Using Practical Early-Exit Networks" (**On-going Project**)

[3] Swapnil Gandhi, Yogesh Simmhan, "An Interval-centric Model for Distributed Computing over Temporal Graphs", In proceedings of the 36th IEEE International Conference on Data Engineering (**ICDE 2020**), Dallas, Texas, April 2020. https://swapnilgandhi.com/papers/icm-icde20.pdf

[4] Swapnil Gandhi, "Wave: A Substrate for Distributed Incremental Graph Processing on Commodity Clusters", 2nd ACM Student Research Competition (**SRC**) at 27th Symposium on Operating Systems Principles (**SOSP 2019**), Ontario, Canada, Oct 2019. https://swapnilgandhi.com/papers/wave-sosp19.pdf

[3] AEC SOSP 2019, OSDI 2020 and ASPLOS 2020
[4] Shadow PC EuroSys 2021 and 2022