My Ph.D. work has focused on examining the workflow and workload of deep learning model inference oriented at mobile devices. Specifically, this research has resulted in 7 peer-reviewed publications, 4 of which are first-author. These works have been largely at cloud- and edge-computing conferences (ACSOS'21 [3], IC2E'20 [4], HotEdge'18 [5], DIDL'19 [2]), as well as performance modeling conferences (ICPE'21 [6], PERFORMANCE'20 [1]), as well as Machine Learning specific conferences (DIDL'19 [2]). My future research aims are aimed at first applying my research to specific fields, such as intelligent embedded sensors, and then expanding my research by exploring new ways to improve and apply deep learning through federated learning and model personalization.

Background and motivation. Mobile deep inference is the use of deep learning models on mobile devices and is an essential step for leveraging deep learning models in real-world environments, such as in augmented reality applications, neural machine translation, and virtual personal assistants. Enabling end-users to use these models on their mobile devices allows for widespread adoption of advanced deep learning models, and allows for more in-depth exploration of the possibilities of using these models. However, mobile deep learning has to contend with constrained local resources, variable network connectivity, and an ever-increasing range of models being used within applications. Therefore, the core research question of my research is how do we provide access to state-of-the-art models to end-users in a *fast* and *predictable* way, and how do we do so with a reasonable amount of resources.

Ph.D. Research

My research answers this question in several ways. By quantifying the constraints and explicitly making decisions based on performance models it was possible to reduce response and bound response latency while opportunistically increasing accuracy (ICPE'21 [6], IC2E'20 [4]). Through a consideration of workload characters, it was possible to identify inefficient allocations of resources and introduce a mechanism for trading off between memory usage and latency (ACSOS'21 [3]). Finally, in my thesis, I aim to schedule multiple deep learning tasks concurrently, while being aware of constraints and data usage requirements. The rest of this statement will discuss the three main thrusts of my thesis as well as future research directions.

Per-request Optimization. Using deep learning models consists of more than just model execution, especially when considering mobile devices, so to improve the performance of deep learning inference for mobile devices I broke the execution into its component steps and optimized the most important of these. First, I approach how to improve response latency and accuracy of individual inference requests. In *PieSlicer* (ICPE'21 [6]), through characterization and modeling of the preparation of input data, I demonstrate it is possible to adapt the execution to reduce the response latency. In *MDInference* (IC2E'20 [4]), I introduce an approach that considers a set of similar models to allow for adapting the execution to match a time budget, thus allowing serving systems to meet specific response latencies, as well as improving accuracy whenever possible. Together, these allow me to reduce response latency and use this reduction to improve the accuracy of inferences for resource-constrained mobile devices.

Model-level Caching. Leveraging cloud-based resources allows much more powerful deep learning models to be executed, but results in many multitudes more requests for many more unique models, due to the number of mobile devices and the diversity of their needs. In *CremeBrulee* (ACSOS'21 [3]), I showed that the need to load these models into memory quickly outpaces the actual computation. To address this, I introduced *model-level caching*, whereby deep learning models are treated as cacheable objects. This allows models to be moved smoothly in and out of memory as discrete units that are easy to reason about. By considering not only their characteristics but also the characteristics of their workload, it becomes straightforward to make good eviction decisions. Even a simple eviction-based model caching approach allows us to dramatically reduce memory requirements for hosting, and thus ensure better resource allocation and reduce hosting costs for deep learning models.

Workload-level scheduling. To further reduce the load on cloud-based resources, my current research aims to leverage both on-device and in-cloud resources for complex inference tasks. Many workloads, such as augmented reality, require the execution of complex combinations of models formed into directed acyclic graphs. Submitting all of these tasks to the cloud would not only incur high network latency, but also high serving costs. Instead, I aim to schedule the execution of individual tasks across both on-device and cloud-based resources that reduce latency by avoiding unnecessary data transfer and avoid resource contention between individual tasks and all executing DAGs. This will reduce latency and resource usage.

Low-level execution optimization. I was also 2nd author of two publications (DIDL'19 [2], PERFORMANCE'20 [1]) which focused on low-level concurrent execution of deep learning models on GPUs. This work introduced a new model for considering concurrent executions on GPUs that was shown to be more accurate than previous models, allowing for finer-grained insights into how to execute two deep learning models concurrently. This is essential for scaling deep learning inference serving since previous works have often shown that sharing resources can lead to dramatic decreases in performance due to resource contention. In these works I provided application-level insights, helping to bridge from low-level operations to high-level implications.

Future Directions

Sensor systems are currently facing a host of new challenges due to the sheer amount of data being collected and the move towards making sensors even more intelligent through deep learning. These two challenges are paired in that machine learning is increasingly being used to do on-device filtering so only the most relevant data is submitted to centralized servers. However, sensors are extremely constrained in terms of energy, computation, memory, and bandwidth, driving a need to only execute selectively. As opposed to my previous work where every inference had to generate a response (ICPE'21 [6], IC2E'20 [4]), this problem instead needs to be approached by considering which inferences are achievable, which are meaningful, and which should be filtered out.

Environmental Adaption. The wide deployment of sensors also leads to challenges with the accuracy of their on-device models as well, since a wider deployment leads to more

variety of environments experienced. This is particularly important in situations where environments are largely similar but have many micro-environments, such as you would find in an agricultural application. The core challenge would be using the generalized insights of each sensor to build more accurate models, but allow individual models to specialize in their environments. This specialization of models largely is at odds with the generalization of models, where overfitting to any particular class is considered undesirable behavior. The goal of this research would be to enable federated learning that allows not only for the per-device specialization but also to improve the shared base model. This line of research will build both on my ongoing research on on-device inference (HotEdge'18 [5]), but also on my mentorship experience that focused on federated learning techniques and applications.

Hardware Adaptation. A further challenge of sensors is hardware diversity. To make widely available and used sensors they must be cheap and easy to repair, which means being able to support a wide range of replacement hardware, both as sensors and, in the case of embodied agents, for mobility and interaction. Adapting using this new hardware can result in inaccurate control mechanisms and a loss of accuracy in results and potentially in control. The core challenge is that adapting to new hardware needs to be done quickly, and maybe in novel combinations, especially when using low-cost hardware. The goal of this research would be to enable such changes to be made by transferring existing models for sensing and control, and as well as personalizing them to the hardware combinations present. This also would build well on my previous work on managing on-device resources (ICPE'21 [6]) to run multiple models, as well as determining the most essential models to keep on the device, which can leverage model-level caching techniques (ACSOS'21 [3]).

In summary, my research focus is to identify issues surrounding mobile systems, identify the impact of specific constraints and techniques to improve performance and resource utilization. This work can have many impacts, both socially by improving user experience, and by allowing for other resource-constrained devices to utilize deep learning models effectively.

References

[1] Guin Gilman, **Ogden, Samuel S**, Tian Guo, and Robert J Walls. "Demystifying the Placement Policies of the GPU Thread Block Scheduler for Concurrent Kernels". In: *38th International Symposium on Computer Performance, Modeling, Measurements and Evaluation* (2020).

[2] Guin R Gilman, **Ogden, Samuel S**, Robert J Walls, and Tian Guo. "Challenges and Opportunities of DNN Model Execution Caching". In: *Proceedings of the Workshop on Distributed Infrastructures for Deep Learning*. 2019.

[3] **Ogden, Samuel S**, Guin R Gilman, Robert J Walls, and Tian Guo. "Many Models at the Edge: Scaling Deep Inference via Model-Level Caching". In: *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. ACSOS '21. Institute of Electrical and Electronics Engineers, Sept. 29, 2021.

[4] **Ogden, Samuel S** and T Guo. "MDINFERENCE: Balancing Inference Accuracy and Latency for Mobile Applications". In: *2020 IEEE International Conference on Cloud Engineering (IC2E)*. Los Alamitos, CA, USA: IEEE Computer Society, Apr. 2020.

[5] **Ogden, Samuel S** and Tian Guo. "MODI: Mobile Deep Inference Made Efficient by Edge Computing". In: *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*. Boston, MA: USENIX Association, 2018.

[6] **Ogden, Samuel S**, Xiangnan Kong, and Tian Guo. "PieSlicer: Dynamically Improving Response Time for Cloud-Based CNN Inference". In: *Proceedings of the ACM/SPEC International Conference on Performance Engineering*. 2021, pp. 249–256.