Research Bio

Ehsan Yousefzadeh-Asl-Miandoab

ehsanyusefzadehasl@gmail.com

How I see Research

I would like to start with how I see research. I see research as an iterative process of asking questions, finding answers, and solving the problems that emerge along the way.

We start with an initial question and investigate what others have already done. As our understanding develops, we refine the question: we may narrow it down, broaden it, or reformulate it entirely as we gain a better view of the research landscape. From there, research becomes a continuous loop of questioning, refinement, experimentation, and problem solving.

This is also where I believe strong technical foundations become particularly valuable. A deep understanding of computer systems, including hardware architecture and the interaction between hardware and software, together with software engineering, data structures and algorithms, databases and SQL, and data analysis and machine learning, provides the foundation for investigating research questions systematically and turning ideas into working systems and reproducible experiments.

Several characteristics keep a researcher moving through this loop: curiosity, dedication, and the humility to question one's own reasoning and findings while remaining open to the perspectives of others.

From GPU Architecture to AI Systems

I formally started doing research during my master's studies, working on GPU microarchitecture and using architectural simulators to evaluate new ideas. At one point, I considered leaving the topic for an area where I felt more confident that I could generate ideas and implement and test them more quickly. Instead, this period became an important turning point. It pushed me to strengthen my understanding of parallel computer architecture and programming.

I also learned an important practical lesson: to work efficiently in this type of research, the evaluation process needs to be automated as much as possible. Initially, I was intimidated

by the prospect of navigating a large simulator codebase, understanding where changes had to be made, and building the infrastructure needed to evaluate my ideas. This was also before tools such as today's LLM-based programming assistants were available. Through persistence, however, I developed a set of technical skills that later became highly useful during my PhD and postdoctoral research.

I completed my master's studies with two publications. The first grew directly from my master's thesis and investigated how to better utilize GPUs' on-chip shared-memory resources:

"**OSM: Off-Chip Shared Memory for GPUs**," IEEE Transactions on Parallel and Distributed Systems, 2022.

The second resulted from a collaboration with a PhD student in our lab and investigated how resources left underutilized because of non-uniform workload behavior could be exploited for other purposes rather than remaining idle:

"**NURA: A Framework for Supporting Non-Uniform Resource Accesses in GPUs**," Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2022.

These projects established a theme that has continued throughout my research: **understanding where computing resources are underutilized and designing mechanisms that make better use of them**.

Resource-Efficient Deep Learning Systems

I had always planned to pursue doctoral studies. As AI and deep learning rapidly developed, I became particularly interested in the intersection of AI systems and hardware. I therefore took the opportunity to work with Pinar Tözün on improving GPU efficiency for deep learning training.

This transition required significant learning. I came from hardware and computer architecture, while my supervisor came from database systems. At the beginning, I had little background in machine learning, deep learning, or data science. I therefore took courses and studied books and research papers before deciding where I could make a meaningful contribution.

After this initial exploration, we focused on an important observation: GPU utilization during deep learning training was frequently reported as low, suggesting substantial resource inefficiency.

Understanding GPU Utilization

My first question was whether the metrics and tools commonly used to discuss GPU utilization were actually sufficient.

What exactly does GPU utilization tell us? Which metrics should we monitor? Which tools should we use? When is monitoring sufficient, and when is detailed profiling necessary?

Answering these questions resulted in my first PhD publication:

"**Profiling and Monitoring Deep Learning Training Tasks**," EuroMLSys @ EuroSys 2023.

This study provided the measurement foundation for the research that followed. Rather than immediately designing a resource-management mechanism, I first wanted to understand what we could reliably observe about GPU workloads and which measurements could realistically be incorporated into an online resource manager.

From Measurement to GPU Sharing

The next question was how to translate these observations into a practical mechanism for improving GPU utilization.

We chose workload collocation: allowing multiple deep learning training tasks to share a GPU when sufficient resources are available. Although GPU sharing provides only a limited abstraction for controlling interference, it offered an important practical advantage.

Our design philosophy was to build a resource manager that could operate without expensive offline profiling and could realistically be deployed in environments such as universities or small and medium-sized organizations. In such environments, workloads arrive independently from different users and research groups, rather than forming a single large workload around which the entire infrastructure can be optimized.

We therefore investigated the effects of collocating deep learning training tasks, studying throughput, slowdown, energy consumption, and different degrees of collocation. These experiments resulted in:

"**An Analysis of Collocating Deep Learning Training Tasks**," EuroMLSys @ EuroSys 2024.

This work revealed two important challenges.

The first was performance interference. Collocating workloads can slow individual tasks down, meaning that a resource manager must determine whether the overall throughput or efficiency gain justifies that slowdown.

The second was more fundamental: GPU memory capacity. If a resource manager decides to place another training task on a GPU without sufficient available memory, the newly arriving task can fail with an out-of-memory (OOM) error. Because such failures are introduced by the resource manager's own collocation decisions, handling them becomes part of the system's responsibility.

Can We Predict GPU Memory Requirements?

My initial idea was to prevent OOM failures entirely. If we could estimate the GPU memory required by a training task before assigning it to a GPU—or before collocating it with another task—we could reject unsafe placements beforehand.

Profiling was an obvious option, but it conflicted with our design philosophy. Profiling either requires additional idle hardware or introduces delay before scheduling decisions can be made. Moreover, determining how long a workload needs to be profiled before its resource behavior is adequately represented is itself difficult and workload-dependent.

This led to another research question:

Can we estimate the GPU memory requirements of deep learning training tasks without profiling them first?

Investigating this question became a demanding iterative process of constructing datasets, analyzing workload behavior, training estimation models, examining failures, and repeatedly refining the approach. Also, reaching out to machine learning and data specialists for getting intuition. Importantly, the work also taught me where estimation works and where its limitations become fundamental.

The study resulted in:

"**Estimating GPU Memory and GPU Utilization for Deep Learning Training Tasks: Opportunities and Limitations**," EuroMLSys @ EuroSys 2026.

Putting the Pieces Together: A Runtime Resource Manager

After studying measurement and monitoring, collocation in isolation, GPU-memory estimation, and its limitations, I started bringing these pieces together into a runtime resource-management system.

A central design choice was to actively monitor GPUs at runtime using lightweight system-level metrics to capture their current load and resource availability, without requiring integration between the machine-learning framework and the scheduler. This keeps the system practical and broadly applicable across different training workloads and frameworks, while still giving the resource manager enough information to make informed placement and collocation decisions.

The resulting system incorporates task collocation directly into the task-to-GPU mapping process. Rather than assuming that memory requirements can always be predicted correctly, it treats OOM failures as conditions that the runtime itself must be able to handle. GPU-memory estimators can inform scheduling decisions, while continuous runtime monitoring provides up-to-date information about the actual state of the GPUs.

The system supports different collocation policies and led to an important finding: for deep learning training workloads, accurate GPU-memory estimation alone is not the defining factor determining whether resource utilization and system efficiency improve. Lightweight

runtime monitoring, adaptive placement decisions, and mechanisms for handling failures and interference are equally important.

This work has undergone major revisions and is currently submitted to SoCC 2026. We initially called the system CARMA, but during the later stages of the project I renamed it AEGIS, which better reflects its role as a runtime mechanism that protects the system while enabling more aggressive GPU sharing.

Together, these projects form the main research trajectory of my PhD and postdoctoral work:

measurement → **understanding inefficiency** → **GPU sharing** → **resource estimation** → **lightweight runtime monitoring and resource management**.

Collaborative Research

Although resource-efficient AI systems have been the main trajectory of my research, I have also pursued several collaborative projects. These collaborations have been particularly valuable because they allowed me to apply a systems and resource-efficiency perspective to problems originating in other research areas.

During a research visit to the University of Basel, I discussed privacy-preserving deep learning with a PhD student working in the area. I suggested investigating not only the privacy and model-quality properties of different methods but also their CPU and GPU resource costs. This collaboration resulted in:

“**DP-Morph: Improving the Privacy-Utility-Performance Trade-off for Differentially Private OCT Segmentation**,” AISec @ CCS 2025.

Another collaboration emerged while I was developing the GPU resource estimators. To overcome some of the challenges I was facing, I collaborated with Reza (a PhD student) whose background was closer to deep learning. During our discussions, I also became involved in his work on reducing the human cost of medical-data annotation through

automated methods. I contributed by analyzing the computational costs of the proposed method and comparing them with the cost of employing professionals for manual annotation. This resulted in:

“**PiMPiC: An Overlap-Aware Contrastive Learning Framework for 3D Patch-Based Medical Image Segmentation**,” DEMI @ MICCA 2025.

During my postdoctoral research, I also shared an office with Alex (a PhD student) investigating metadata-based techniques for efficient database access. His work compared different metadata statistics and their effectiveness across different datasets and queries. I became interested in the project because of its connection to a question that appears repeatedly in my own research: what information is worth collecting, and when does the additional information justify its cost?

Our collaboration resulted in:

“**Benchmarking Column Statistics for Analytical Query Pruning**,” DBTest @ SIGMOD 2026.

Finally, as a member of the Resource-Aware Data Systems (RAD) group during my PhD and postdoctoral studies, I contributed to additional collaborative projects. For the RADT framework, I contributed to the development of its listeners and metrics; this work was published at DEEM @ SIGMOD 2023. I also contributed to a study of LLM systems through discussions about experimental methodology, interpretation of figures, and how visualizations can reveal—or sometimes obscure—the behavior of the systems being evaluated. This work was published at EuroMLSys @ EuroSys 2026.

These collaborations strengthened another aspect of how I approach research: I enjoy bringing a systems, measurement, and resource-efficiency perspective to problems originating in neighboring areas.