Modern intelligent systems are becoming increasingly monolithic, powered by gigantic foundation models trained on trillions of tokens of web data. To democratize AI systems, it is imperative to ensure that they are not limited to running on multi-accelerator clusters but also on commodity devices like mobile phones seamlessly. Additionally, foundation models exhibit a performance disparity between frequently encountered *head* tasks in the training data and less common *tail* tasks, necessitating their adaptation through efficient retrieval of relevant contextual data. Furthermore, echoing human learning principles, not all tasks are equally challenging or require the entirety of the vast web data. My research methodology centers on translating these concepts into practical solutions for real-world implementation, ensuring that these intelligent systems can be scaled reliably and responsibly to serve all users equitably.

With the goal of **efficient, elastic and contextual intelligence**, I focus on building fundamental machine learning (ML) building blocks that encompass: (1) elastic representations and models for accurate, adaptive and efficient deployment and (2) mechanisms to make contextual data efficiently accessible to models for equitable adaptation.

Towards the elastic modeling, along with traditional efficiency techniques [2, 3, 21, 22], I introduced the "Matryoshka" way of packing information in a dense vector – the atomic building block of every ML model. This enabled elastic multi-granular matryoshka embeddings for datapoints [5] as well as elastic universal matryoshka neural network models [10] at web-scale. Concretely, matryoshka representation learning (MRL) [5] is now a default design choice for **universal embedding models at Google** that power image search, photos, and multimodal ads directly **impacting over a Billion users** every day. At the same time, Matryoshka Transformer (MatFormer) [10] enables model *virtualization* for adaptive computation across discriminative and generative tasks and is a **next generation architecture** for web-scale foundation models.

Towards efficient access to contextual data, I revisited the fundamental problem of dense retrieval [4, 14, 20] that powers all of modern-day search [18, 28]. I developed approximate nearest neighbor search (ANNS) methods that leverage the elastic embeddings for flexible search [20] and end-to-end differentiable search solutions that are more data-driven [14]. Furthermore, I also fundamentally rethought traditional search by learning compact binary codes for data points that double as accurate representations and efficient web-scale indices [4]. This enabled us to accurately index **1 Billion images with only 8 GB memory** and can power offline web-scale search on a smartphone. Inspired by priming [24] from cognitive psychology, I believe that foundation models can adapt [16, 31] efficiently to the *tail* tasks in the presence of appropriate retrieved data from the vast web and more broadly the whole world.

The overarching theme of my research is to improve the **building blocks of ML systems to do more for the same resource usage with simple and scalable techniques**. Due to the fundamental nature, the techniques I build, for modeling and retrieving data, work together seamlessly and can help build truly elastic and adaptive web-scale intelligent systems to serve the users dynamically and equitably based on task, context, and resource constraints. Finally, each of these research directions stands on its own merit solving high-impact fundamental problems like large-scale search and efficient deployment that have potential applications across various fields that extend beyond computer science.

Past Research

Efficient and elastic ML models. Even back in 2017, when I started working on ML research, the state-of-the-art (SOTA) deep learning (DL) models were computationally very expensive. For example, it was not until late 2017 that wakeword detection of "Hey Siri" happened on-device [25]. I realized the importance of having efficient ML models that can be deployed in extremely resource-constrained settings to make intelligent systems real-time and ubiquitous.

To enable real-time intelligence on the edge, I developed FastGRNN [2, 1] that powered "Hey Cortana" wakeword detection with a 1 KB model at *Microsoft*. FastGRNN was focused on efficiency stemming from simple algorithmic design and a combination of now ubiquitous compression techniques. We further extended these ideas to build an on-device radar-based poacher detection system in remote wildlife reserves [21] and reduce the RAM usage, to less than 256 KB, of deep CNN models for object detection by using RNNPool [22] to perform accurate downsampling. Furthermore, I introduced the concept of learnable sparsity [3] which was the first end-to-end differentiable method that achieved SOTA "Accuracy vs FLOPs vs Model size". While extremely powerful, these efficiency techniques often demand extra training for specific resource constraints. They are also hard to apply dynamically, even on the cloud, based on the server load and other requirements. This necessitates the development of **elastic** entities that can seamlessly adapt to evolving downstream requirements without incurring any additional costs.

To address this fundamental problem, I proposed "Matryoshka" structure in dense vector representations to order the information from left to right based on importance in a nested fashion. Matryoshka representation learning (MRL) [5] helps neural networks output dense vectors that are inherently multi-granular by jointly optimizing the same learning task at a select few embedding granularities. MRL helps obtain accurate low-dimensional representations of desired quality and cost/size by taking the appropriate number of leftmost coordinates. This helps **elastically** cater to downstream tasks of varying requirements like retrieval, classification, etc., in the transfer learning paradigm. MRL is simple, scalable, and agnostic to representation learning setups, modalities, and models which made it a default choice in universal embedding models at Google that rely on a single model to perform a variety of tasks but are restricted by the most latency-sensitive task like web search. Today, matryoshka representations serve over **a Billion users daily across Google products** and have been widely explored in the industry.

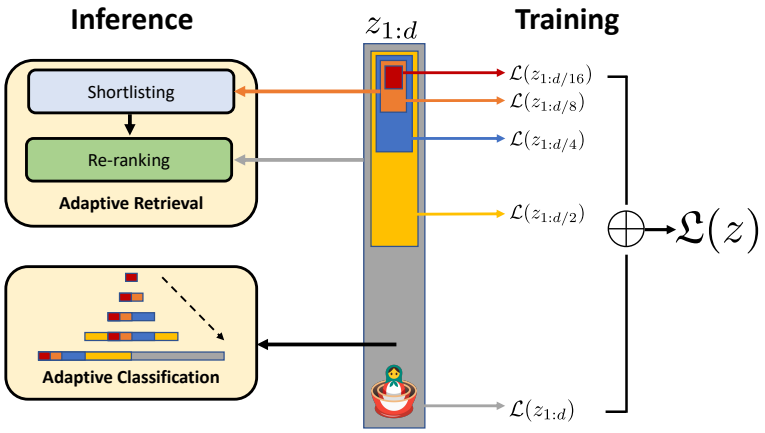

Figure 1: Matryoshka style ordered and nested information packing in a dense vector representation.

Incorporating similar ideas into the weights of neural networks, not just the final embeddings out of them would help train one universal model that can be elastically deployed across setups and tasks at no additional cost. We developed MatFormer [10] which brought the matryoshka structure to all of the Transformer [27] architecture. MatFormer enables extractions of 100s of smaller accurate models for a **wide range of static deployment constraints and also supports**

dynamic conditional inference on-the-fly based on task hardness [23] and resource constraints. Additionally, MatFormer provides smaller submodels that are inherently consistent with the universal model due to the preservation of metric-space structure. This allows for significant speed-ups in inference time optimization of generative language models [15] as well as enabling adaptive query encoders for large-scale retrieval for the first time. Similar to MRL, MatFormer is domain and setup agnostic while scaling, to internet-scale, as reliably as the default Transformer. Overall, MatFormer is a next-generation architecture that elicits **elasticity** and **virtualization** within foundation models that form the basis of modern-day web-scale intelligent systems.

Efficient data retrieval. Humans perceive things in a relative sense by comparing them to preexisting memories [6] – web-scale machine learning is no different. Modern web search consists of data points encoded as embeddings using a neural network, followed by building an efficient ANNS index to effectively search over 100 billion instances for a given query. While remarkable, the fundamentals behind this have not been revisited in a long time. This made me work on improving and rethinking the fundamental building blocks of semantic search.

We focused on flexibility within ANNS building blocks [20] by leveraging the multi-granular and elastic MRL representations for web-scale data. This helped design $2-10\times$ efficient ANNS indices for web data without compromising accuracy. Now, matryoshka representations could be used at web-scale for on-the-fly adaptive and equitable search without the need to rebuild indices across granularities and is being actively explored at Google scale. Despite their success, representations and ANNS indices do not inform each other in a data-driven fashion leading to sub-optimality. To this end, we showed that jointly training representations and the a differentiable tree-based ANNS index [14] improves accuracy and load-balancing while reducing latency compared to existing modular systems. However, this solution still has human intervention, through the inductive bias on the ANNS structure, which I wish to completely get rid of.

As an alternative, I rethought the entire pipeline as a representation learning problem, through the lens of compression and scalable instance classification, where each data point is assigned a learned low-dimensional binary code [4]. These binary codes have the required semantic information for downstream tasks, while also acting as a native hash-based index for all the data points. This works at scale resulting in an accurate encoding of 1 Billion images with just 8 bytes per image which also serves as an extremely efficient web-scale index for search on-demand. Rethinking search to be end-to-end differentiable and free of scaffolds can result in large amounts of data being available for offline search based on the context during deployment.

Finally, most datasets used for large-scale training are not fair in their distribution across various axes. To alleviate some of this problem, we also curated and audited web-scale datasets for underserved tasks like 3D modeling [9] and multi-lingual NLP [13]. Without online adaptation of foundation models, either through priming [31] or retrieval augmentation [16], using relevant retrieved data based on the context, it is nearly impossible for equitable serving of intelligent systems. My research on efficient data retrieval tackles the algorithmic aspects of web-scale search to get relevant information effectively by allocating compute elastically based on the need.

Future Research Agenda

I envision a future where the AI systems cater to every user accurately, reliably and equitably based on their needs in real-time. To realize this goal, I shall leverage my experience and expertise in elastic modelling and efficient data retrieval to pursue the following directions.

End-to-end elastic search. Bringing together everything I have developed towards elastic modelling, representations and ANNS will result in a truly elastic end-to-end learned search system that maintains accuracy for *head* tasks at fraction of the cost while being able to spend lot more resources to cater the rare *tail* queries to not leave any data or user behind. Currently, I am co-leading a team at Google to build a prototype at scale for real-world use cases.

Indexing the world. Rethinking search through end-to-end representation learning and compression opens up a new and on-the-fly way to index the entire world, not just the web. Imagine a robot that is moving around and perceiving its surroundings, at the moment it can rarely remember everything it saw, heard or felt. Enabling efficient representation that doubles as index of the perceived states would help any embodied or intelligent agent make a more informed decision like what a human or even a crow would. This also helps in improving privacy by enabling on-device indexing and search without compromising on accuracy for smart devices [11]. Beyond perception, this representation learning paradigm assists in any setup that requires accurate and fast search on all the candidates like in drug discovery [8] or protein structure generation [26]. I wish to expand on potential applications of efficient large-scale search and storage across natural sciences for a more grounded use of generative foundation models.

Contextual foundation models. Hallucinations are the Achilles' heel of modern generative models, especially for tail tasks. While post-hoc retrieval augmentation [16, 19] can fix some of the issues and make the generations more grounded and diverse, I look forward to building contextual foundation models that are explicitly designed and optimized to leverage retrieval of relevant contextual data and external memory banks as core components in their inference [7].

Continually learning intelligent systems. While human learning is never ending, the machine equivalent, continual learning has hit a road block owing to the issues in evaluation [29]. Loosely drawing parallels to human brain or to a great extent emulating the modern-day computer architecture, elastic models can act as hierarchical information packing and learning entities. I would like to revisit continual learning, through the lens of elastic and contextual modelling, in real-world to capture trends across temporal scales while discovering new things along the way [30] to eventually build a world model along side fast local models across time-scales [12].

In sum, my research focuses on designing fundamental ML algorithms with strong empirical performance and real-world deployability geared towards enabling **efficient, elastic and contextual intelligence** that can bring the systems ever so close to the efficiency of the human brain [17].

[1] **A, Kusupati**, D. Dennis, C. Gupta, A. Kumar, H. V. Simhadri, and S. Patil. The EdgeML Library: An ML library for machine learning on the Edge, 2017. URL https://github.com/Microsoft/EdgeML.

[2] **A. Kusupati**, M. Singh, K. Bhatia, A. Kumar, P. Jain, and M. Varma. FastGRNN: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network. *Advances in neural information processing systems*, 2018.

[3] **A, Kusupati**, V. Ramanujan, R. Somani, M. Wortsman, P. Jain, S. Kakade, and A. Farhadi. Soft threshold weight reparameterization for learnable sparsity. In *International Conference on Machine Learning*, 2020.

[4] **A. Kusupati**, M. Wallingford, V. Ramanujan, R. Somani, J. S. Park, K. Pillutla, P. Jain, S. Kakade, and A. Farhadi. LLC: Accurate, multi-purpose learnt low-dimensional binary codes. *Advances in neural information processing systems*, 2021.

[5] **A. Kusupati***, G. Bhatt*, A. Rege*, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain, and A. Farhadi. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 2022.

[6] M. Bar. The proactive brain: memory for predictions. *Philosophical Transactions of the Royal Society*, 2009.

[7] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2022.

[8] S. Dara, S. Dhamercherla, S. S. Jadav, C. M. Babu, and M. J. Ahsan. Machine learning in drug discovery: a review. *Artificial Intelligence Review*, 55(3):1947–1999, 2022.

[9] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, **A. Kusupati**, A. Fan, et al. Objaverse-XL: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems, Datasets and Benchmarks*, 2023.

[10] Devvrit*, S. Kudugunta*, **A, Kusupati***, T. Dettmers, K. Chen, I. Dhillon, Y. Tsvetkov, H. Hajishirzi, S. Kakade, A. Farhadi, and P. Jain. MatFormer: Nested transformer for elastic inference. *arXiv:2310.07707*, 2023.

[11] Humane. Humane ai pin. *Humane blog*, 2023. URL https://hu.ma.ne/aipin.

[12] D. Kahneman. *Thinking, fast and slow*. macmillan, 2011.

[13] S. Kudugunta, I. Caswell, B. Zhang, X. Garcia, C. A. Choquette-Choo, K. Lee, D. Xin, **A. Kusupati**, R. Stella, A. Bapna, et al. MADLAD-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems, Datasets and Benchmarks*, 2023.

[14] R. Kumar, A. Mittal, N. Gupta, **A. Kusupati**, I. Dhillon, and P. Jain. EHI: End-to-end learning of hierarchical index for efficient dense retrieval. *arXiv:2310.08891*, 2023.

[15] Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, 2023.

[16] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 2020.

[17] H. Moravec. When will computer hardware match the human brain. *Journal of evolution and technology*, 1998.

[18] P. Nayak. Understanding searches better than ever before. *Google AI Blog*, 2019. URL https://blog.google/products/search/search-language-understanding-bert/.

[19] A. Rege and **A. Kusupati**. FReAD: Faithful retrieval augmented diffusion models. *Work in progress*, 2023.

[20] A. Rege*, **A. Kusupati***, A. Fan, Q. Cao, S. Kakade, P. Jain, and A. Farhadi. AdANNS: A framework for adaptive semantic search. *Advances in Neural Information Processing Systems*, 2023.

[21] D. Roy, S. Srivastava, **A. Kusupati**, P. Jain, M. Varma, and A. Arora. One size does not fit all: Multi-scale, cascaded rnns for radar classification. In *Proceedings of the ACM International Conference on Systems for Energy-efficient Buildings, Cities, and Transportation*, 2019.

[22] O. Saha, **A. Kusupati**, H. V. Simhadri, M. Varma, and P. Jain. RNNPool: Efficient non-linear pooling for ram constrained inference. *Advances in Neural Information Processing Systems*, 2020.

[23] M. Salehi, S. Mehta, **A. Kusupati**, A. Farhadi, and H. Hajishirzi. SHARCS: Efficient transformers through routing with dynamic width sub-networks. *Findings of Empirical Methods in Natural Language Processing*, 2023.

[24] D. L. Schacter and R. L. Buckner. Priming and the brain. *Neuron*, 20(2):185–195, 1998.

[25] Siri Team. Hey siri: An on-device dnn-powered voice trigger for apple's personal assistant, 2017. URL https://machinelearning.apple.com/2017/10/01/hey-siri.html.

[26] M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, C. L. Gilchrist, J. Söding, and M. Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02, 2022.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

[28] C. Waldburger. As search needs evolve, microsoft makes ai tools for better search available to researchers and developers. *Microsoft AI Blog*, 2019. URL https://blogs.microsoft.com/ai/bing-vector-search/.

[29] M. Wallingford, **A. Kusupati**, K. Alizadeh-Vahid, A. Walsman, A. Kembhavi, and A. Farhadi. FLUID: A unified evaluation framework for flexible sequential data. *Transactions on Machine Learning Research*, 2023.

[30] M. Wallingford, **A. Kusupati**, A. Fang, V. Ramanujan, A. Kembhavi, R. Mottaghi, and A. Farhadi. Neural radiance field codebooks. *International Conference on Learning Representations*, 2023.

[31] M. Wallingford, V. Ramanujan, A. Fang, **A. Kusupati**, R. Mottaghi, A. Kembhavi, L. Schmidt, and A. Farhadi. Neural priming for sample-efficient adaptation. *Advances in Neural Information Processing Systems*, 2023.