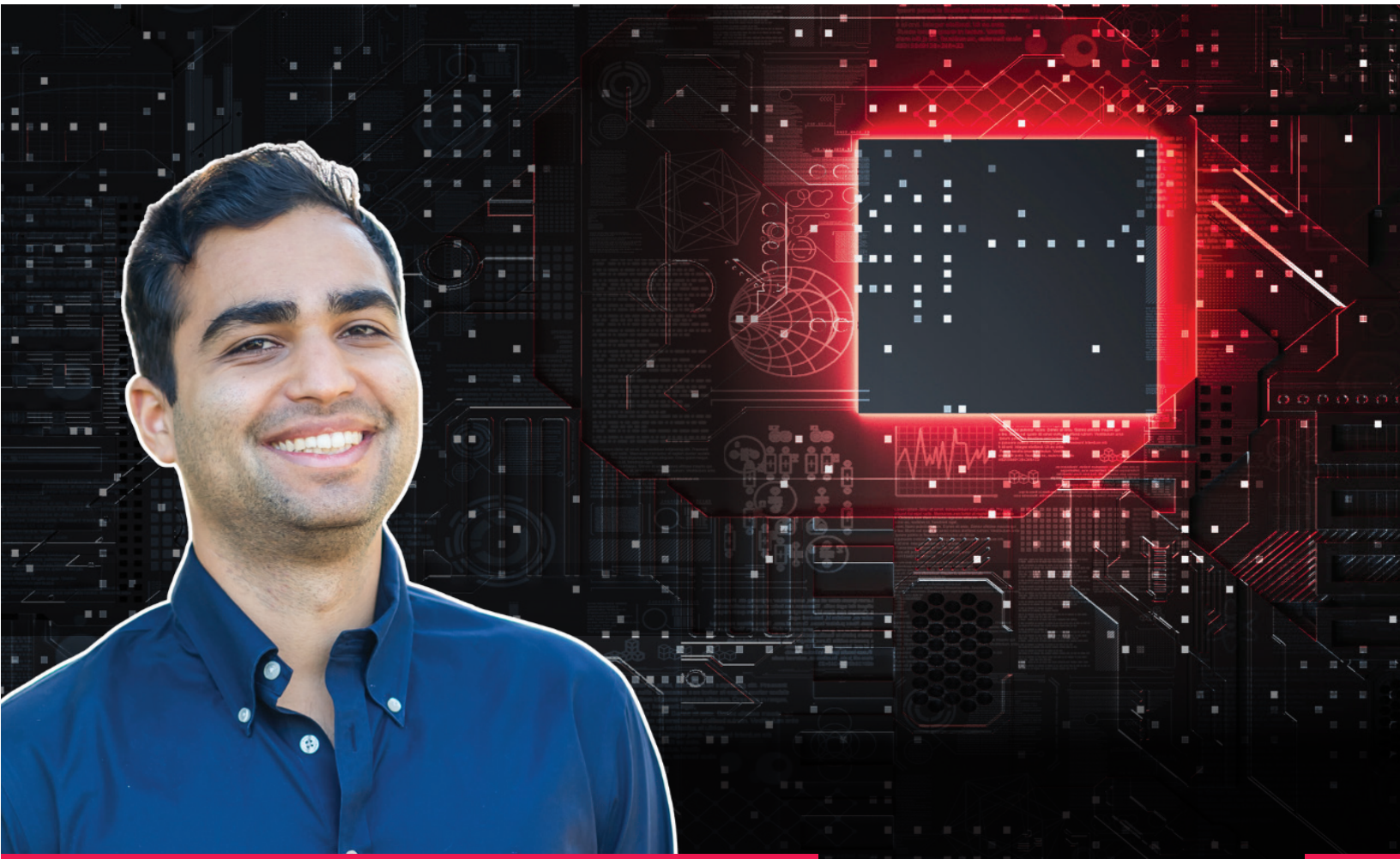
MARKTECHPOST

Small Language Models

Partners: Real AI DSC EUROPE

Featured: Predibase activeloop llmware



Exclusive Talk

Devvret Rishi

CEO and Cofounder of Predibase

NOVEMBER 24

MARKTECHPOST

Marktechpost Media Inc. is an AI Research and Dev News Platform

* Monthly Traffic on Marktechpost.com: 2 Million+
* AI Community: 500k+ Members
* Newsletter

EDITORIAL TEAM

Asif Razzaq

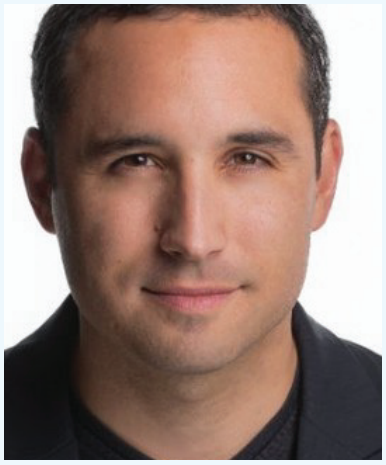
CEO and Editor of Marktechpost

Leading the charge in AI media innovation, **Asif Razzaq is a seasoned media professional and visionary entrepreneur with over eight years of experience in the Artificial Intelligence (AI) and Machine Learning (ML) Business industry.** As the CEO and Editor of Marktechpost Media Inc., a California-based AI digital publishing platform, Asif has dedicated his career to making AI and ML accessible to a broad audience through compelling and insightful media content.Under his stewardship, Marktechpost has evolved into a premier destination for AI enthusiasts, professionals, and researchers, boasting approximately 1.5 million monthly visitors.

Asif's expertise lies in curating and producing high-quality articles, podcasts, and newsletters that translate complex AI and ML concepts into engaging and easily understandable formats. His commitment to excellence ensures that the platform delivers technically sound information while remaining accessible to readers with varying levels of expertise.

Asif's passion for leveraging AI for social good is evident in his media initiatives. He leads a dynamic team of writers, editors, and marketers who share his vision of demystifying AI and ML technologies. By providing in-depth coverage of machine learning and deep learning news, Marktechpost under Asif's leadership has become synonymous with quality and reliability in AI media.

Through his work, Asif continues to influence the discourse in the AI community, fostering greater understanding and appreciation of AI's potential. His role as a media professional is not just about reporting on AI advancements but also about inspiring others to explore the limitless possibilities that AI offers for the future.



Jean-Marc Mommessin

Co-Founder Marktechpost

Jean-Marc is a seasoned AI executive with a track record of driving growth for AI-powered solutions. He founded a computer vision company in 2006 and has since become a recognized leader in the field, frequently speaking at AI conferences. He holds an MBA from Stanford and a Master's in Engineering from Arts et Métiers. Jean-Marc has filed patents on agentic AI and has played key leadership roles, including leading sales at Exodigo.ai and other high-growth startups. He is currently based in Palo Alto, CA.

Tarry Singh

Co-Founder Marktechpost

Tarry Singh is the Board Director & CEO of DK AI Holding, as well as a co-founder and AI researcher for Real AI, an enterprise AI startup, and Earths-can.io, an energy-focused AI startup. These ventures are part of NVIDIA's Inception Program, highlighting top AI startups worldwide. With 30 years of expertise, Tarry has advised global executives, government leaders, and country states on building data-driven organizations from the ground up. A frequent speaker at global AI leadership summits, he leads high-impact workshops with a team specializing in Generative AI, NLP, Computer Vision, Robotics, and other AI fields.

Tarry also co-supervises PhD projects and serves as a visiting professor and advisory board member in Machine Learning at universities in Italy and the Netherlands. Known as a AI and Data Science top voice on LinkedIn since 2018, he has co-developed Europe's first human-centric AI MSc program and contributed to Coursera's leading Deep Learning specialization.

Aleksandar Linc-Djordjevic

Managing Director of the DSC Franchise

Aleksandar Linc-Djordjevic is the founder and Managing Director of the DSC Franchise, one of the largest independent tech conference franchises focused on Data & AI. With over 15 years of experience in profit and non-profit event management, Aleksandar is dedicated to leveraging data for positive global impact, empowering professionals and organizations with the latest innovations and best practices. As of July 31, 2024, he also serves as a member of the National AI Council of Serbia. Aleksandar is a certified trainer and mentor, having delivered over 500 hours of training and holding certifications in organizational development, communication, project management, and fundraising.

He has actively contributed to national strategies for youth and artificial intelligence in Serbia and partnered with institutions to advance data science and AI. Outside of work, Aleksandar enjoys writing poetry and following the NBA.

Table of Contents

Small Language Models,Quid?

LLMs have made significant strides powering amazing applications. However, their substantial computational and memory requirements often make them impractical for deployment in resource-constrained environments like on premise, mobile devices, embedded systems, or real-time applications. This limitation has led to the development of Small Language Models (SLMs), which aim to retain much of the performance of their larger counterparts while significantly reducing their size and computational demands.

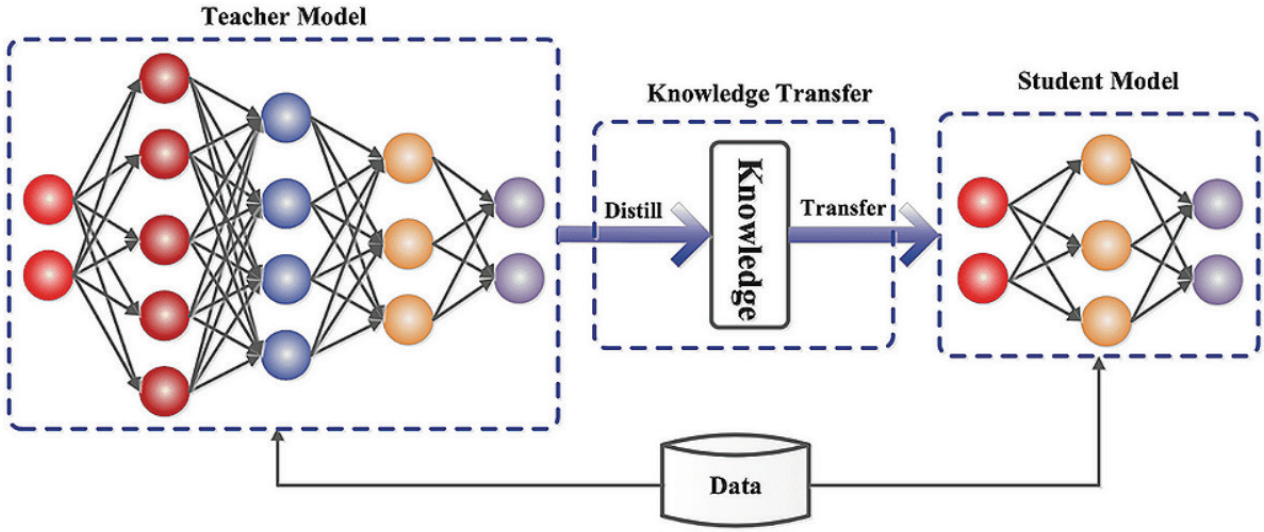
How are SLMs possible? The answer lies in a combination of model compression techniques, efficient neural network architectures, and task-specific optimizations.

Model Distillation:

Transferring Knowledge Efficiently

Model distillation, also known as knowledge distillation, is a pivotal technique for building efficient machine learning models. Introduced by Geoffrey Hinton in his seminal paper "Distilling the Knowledge in a Neural Network" (2015), the process involves transferring knowledge from a large, complex model (the teacher) to a smaller, more deployable one (the student). The goal is to preserve the performance of the original model while significantly reducing memory usage and computational demands.

The core idea is that the student model learns to mimic the predictions and behavior of the teacher model. Instead of training solely on hard labels (the correct outputs), the student model is trained on the "soft" outputs or logits of the teacher model. These soft targets contain rich information about the probabilities of various possible outputs, allowing the student model to capture nuanced relationships between inputs and outputs that would be lost with traditional supervised learning methods.



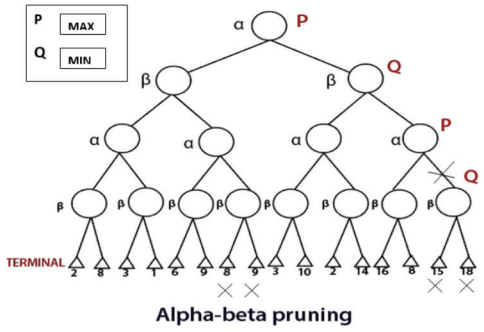
Parameter Efficiency:

Pruning and Quantization

Achieving parameter efficiency is crucial for reducing the size and computational complexity of language models. Techniques such as pruning, quantization, and low-rank matrix factorization enable models to eliminate unnecessary parameters or compress weights while retaining critical information.

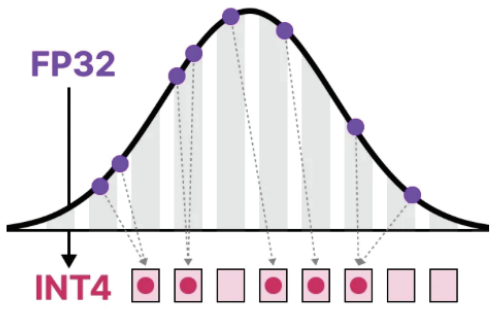
Pruning

Pruning involves removing less important neurons or connections within a neural network. By analyzing the weights and activations, the model identifies and eliminates components that contribute minimally to the output. This process reduces the overall number of parameters and can be performed iteratively to fine-tune the model's size and performance.



Quantization

Quantization reduces the precision of the model weights and activations. For example, converting 32-bit floating-point numbers to 8-bit integers can dramatically lower memory usage and computational cost. Quantization may introduce noise, but advanced techniques can mitigate performance degradation.



Transfer Learning:
Leveraging Pre-trained Models

Transfer learning plays a vital role in making SLMs possible by allowing smaller models to fine-tune on specific tasks using relatively small amounts of data. Instead of training from scratch, SLMs leverage large-scale pre-training on vast datasets, which enables them to perform well on specialized tasks with minimal fine-tuning.

During training, SLMs can learn from the soft outputs of a larger model rather than hard labels. This approach enables the smaller model to capture intricate patterns and relationships within the data that might be missed with traditional supervised learning methods.

Efficient Architectures: Scaling Down Transformers

Transformers can be scaled down: we can develop small language models that maintain essential capabilities while operating more efficiently.

Applications of Small Language Models

The advancements in SLMs have opened up numerous applications across various industries and devices:

On-Device AI
SLMs enable AI functionalities directly on devices like smartphones. Companies like Apple incorporate SLMs for features such as on-device speech recognition, predictive text input, and personal assistants.

Privacy: Data remains on the device, enhancing user privacy.

Reduced Latency: Eliminates the need for constant server communication, resulting in faster responses.

Offline Functionality: Allows features to work without internet connectivity.

Enterprise and Network Applications
In sectors like banking and finance, SLMs can be deployed within secure networks to process sensitive data without transmitting it externally.

Security Compliance: Meets regulatory requirements for data handling.

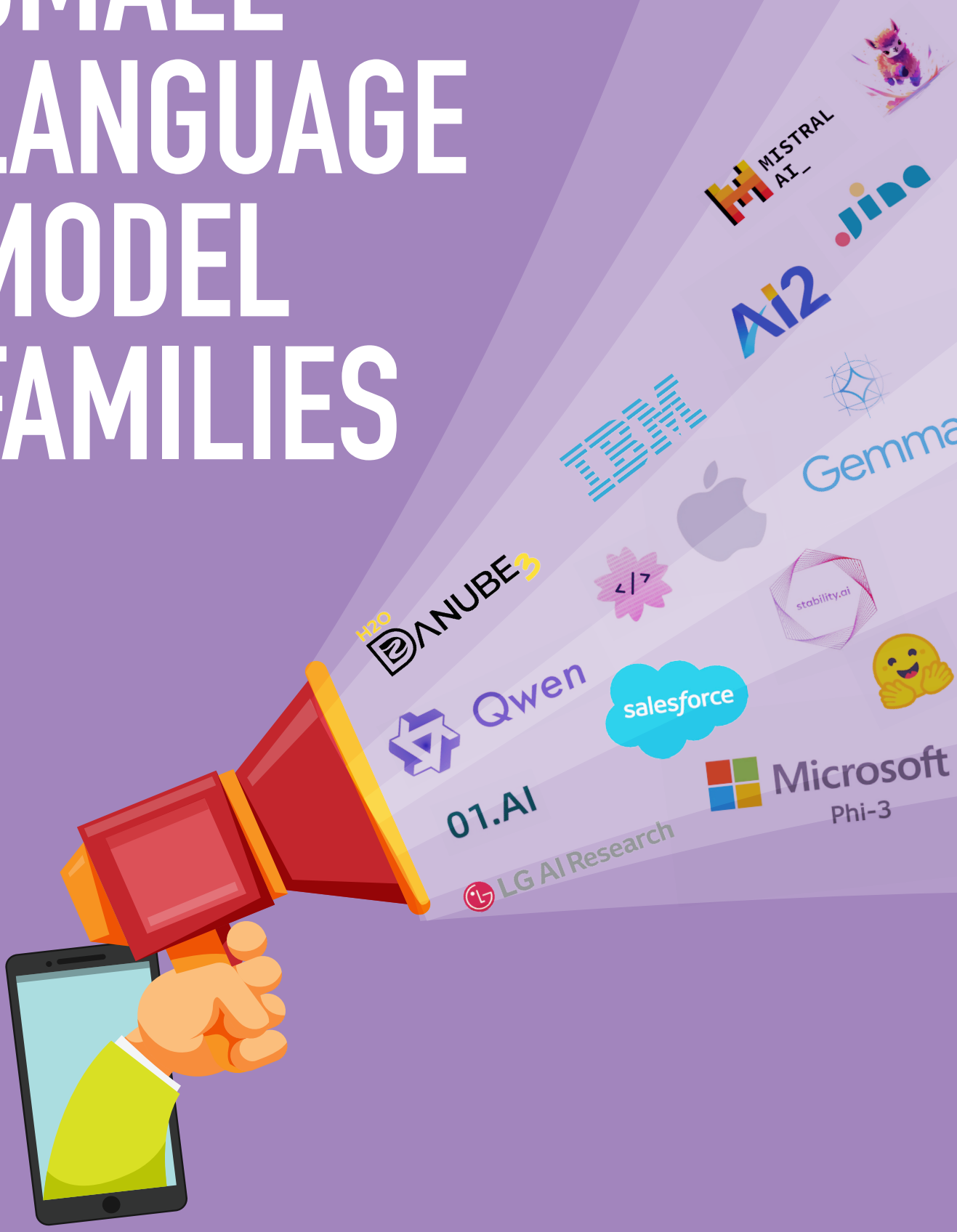
Custom Solutions: Tailored models for specific enterprise needs.

Efficiency: Handles large volumes of data in real-time with reduced infrastructure costs.

Robotics and IoT
SLMs empower robots and Internet of Things (IoT) devices with natural language understanding capabilities.

Edge Computing: Processes data locally on devices with limited computational power.
Real-Time Interaction: Enables immediate responses and interactions.
Energy Efficiency: Reduces power consumption, crucial for battery-operated devices.

SMALL LANGUAGE MODEL FAMILIES



Small Language Model Families

Google Gemma

Gemma is a family of lightweight, state-of-the-art open models from Google, built from the same research and technology used to create the Gemini models. They are text-to-text, decoder-only large language models, available in English, with open weights for both pre-trained variants and instruction-tuned variants. Gemma models are well-suited for a variety of text generation tasks, including question answering, summarization, and reasoning. Their relatively small size makes it possible to deploy them in environments with limited resources such as a laptop, desktop, or your own cloud infrastructure, democratizing access to state-of-the-art AI models and helping foster innovation for everyone.

H2O Danube

H2O-Danube is a family of small 1.8B parameter language models, including H2O-Danube-1.8B, trained on 1T tokens, and its enhanced successor H2O-Danube2-1.8B, trained on an additional 2T tokens. These models excel in performance across a wide array of benchmarks, with H2O-Danube2-1.8B currently holding the top position on the Open LLM Leaderboard for models below the 2B parameter mark. Drawing on principles from LLama 2 and Mistral, H2O-Danube models integrate advanced techniques for pre-training large language models, further enhanced by supervised fine-tuning and direct preference optimization in chat models. All models are made openly available under the Apache 2.0 license, expanding access to powerful LLMs and fostering innovation.

IBM PowerLM 3B

IBM's dense and mixture of experts (MoE) language models are trained using an innovative power learning rate scheduler, designed to address the complexity of tuning hyperparameters for large-scale models. Determining the ideal learning rate for models with billions or trillions of parameters is notoriously difficult, given the interplay between batch size, model size, number of training tokens, and other factors. Recent studies suggest using small proxy models and datasets for hyperparameter optimization, with findings transposed to larger models. While zero-shot transferability of size-related hyperparameters (like depth and width) is well-explored, transferring from small to large datasets remains under-studied. This work delves into the relationship between learning rate, batch size, and training tokens for the newly introduced WSD scheduler, conducting extensive small-scale tests to drive optimization for large models.

01.AI

Yi-Coder is a series of open-source large language models (LLMs) designed specifically for code generation and comprehension, offering state-of-the-art performance with models under 10 billion parameters. Available in two sizes—1.5B and 9B parameters—Yi-Coder provides both base and chat versions, optimized for efficient inference and flexible training. Yi-Coder-9B, the larger model, is built on the foundation of Yi-9B with the addition of 2.4 trillion high-quality tokens, carefully curated from GitHub repositories and code-related data filtered from CommonCrawl.

Key features of Yi-Coder include Extensive Pretraining: Trained on 2.4 trillion tokens across 52 major programming languages, ensuring broad and deep coding knowledge, Long-Context Modeling: A maximum context window of 128K tokens enables project-level code understanding and generation, allowing for more comprehensive outputs and Top-Tier Performance: Despite its smaller size, Yi-Coder-9B surpasses other models in the sub-10B range, including CodeQwen1.5 7B and CodeGeex4 9B, and even competes with much larger models like DeepSeek-Coder 33B.

AMD

AMD-135M model has performance comparable to popular models in the market, we benchmarked with several open-source models of similar size. The result demonstrated the state-of-the-art performance of AMD-135M that exceeds Llama-68M and LLama-160M models on tasks like Hellaswag, SciQ, and ARC-Easy. Also, the performance is on par with GPT2-124M and OPT-125M for the given tasks under Hellaswag, WinoGrande, SciQ, MMLU and ARC-Easy

Microsoft

Microsoft Phi-3

The Phi-3 model collection represents the latest innovation in Microsoft's family of Small Language Models (SLMs), engineered for exceptional performance and cost-efficiency. Designed to surpass models of similar and even larger sizes, Phi-3 excels across multiple benchmarks in language, reasoning, coding, and math. The release of Phi-3 models provides Azure customers with a broader selection of high-quality models, delivering practical and versatile solutions for building generative AI applications.

Since its launch in April 2024, feedback from customers and the community has helped shape the evolution of the Phi-3 models. Today, we are excited to introduce Phi-3.5-mini, Phi-3.5-vision, and a new member of the family, Phi-3.5-MoE, a Mixture-of-Experts (MoE) model.

Phi-3.5-mini: Enhances multi-lingual capabilities with a 128K context window, enabling improved language support across diverse tasks.

Phi-3.5-vision: Elevates multi-frame image understanding and reasoning, significantly boosting performance on single-image benchmarks.

Phi-3.5-MoE: Featuring 16 experts and 6.6B active parameters, this model delivers high performance, reduced latency, and multi-lingual support while implementing robust safety measures. Phi-3.5-MoE outperforms much larger models, further solidifying the efficacy of the Phi series.

Qwen

Qwen2.5 is the latest addition to the Qwen family of open-source language models, designed to deliver cutting-edge performance across a wide range of tasks. Built from the same technology as its predecessor, Qwen2, these dense, decoder-only models are available in multiple sizes and are ideal for text generation, coding, and mathematical reasoning. With open weights provided for both base and specialized variants, Qwen2.5 models are accessible to developers for custom training and deployment.

Qwen2.5 comes in a variety of sizes—ranging from 0.5B to 72B parameters—and is joined by specialized models for coding (Qwen2.5-Coder) and mathematics (Qwen2.5-Math), making them highly versatile for a wide array of applications. The relatively small models, such as the 1.5B and 7B variants, are particularly well-suited for environments with limited resources, while the larger models, like the 72B, provide top-tier performance for complex tasks.

The open-weight models, except for the 3B and 72B variants, are licensed under Apache 2.0, further democratizing access to powerful LLMs. In addition to these releases, Qwen2.5 also includes Qwen-Plus and Qwen-Turbo APIs through Model Studio, and features performance-enhanced versions like Qwen2-VL-72B, expanding the Qwen family's capabilities for developers everywhere.

TinyLlama

TinyLlama is an open-source project designed to pretrain a compact 1.1B parameter model on 3 trillion tokens. By leveraging optimized training techniques, the project aims to complete this ambitious pretraining within just 90 days, using a cluster of 16 A100-40G GPUs. Training began on September 1, 2023, marking a significant step forward in developing lightweight, high-performance language models.

TinyLlama adopts the exact same architecture and tokenizer as Llama 2, making it fully compatible with numerous open-source projects built on the Llama platform. With only 1.1B parameters, TinyLlama offers a highly efficient solution for applications that require minimal computational and memory resources, while still delivering robust performance across a range of tasks.

Deepseek

Janus is a novel autoregressive framework that unifies multimodal understanding and generation. It addresses the limitations of previous approaches by decoupling visual encoding into separate pathways, while still utilizing a single, unified transformer architecture for processing. The decoupling not only alleviates the conflict between the visual encoder's roles in understanding and generation, but also enhances the framework's flexibility. Janus surpasses previous unified model and matches or exceeds the performance of task-specific models. The simplicity, high flexibility, and effectiveness of Janus make it a strong candidate for next-generation unified multimodal models. They released Janus-1.3B.

Stability AI

StableLM-3B-4E1T is a 3 billion parameter decoder-only language model pre-trained on 1 trillion tokens of diverse English and code datasets for 4 epochs. The model is pre-trained under the multi-epoch regime to study the impact of repeated tokens on downstream performance.

Allen Ai

OLMoE-1B-7B is a Mixture-of-Experts LLM with 1B active and 7B total parameters released in September 2024 (0924). It yields state-of-the-art performance among models with a similar cost (1B) and is competitive with much larger models like Llama2-13B. OLMoE is 100% open-source

Apple OpenELM

OpenELM, a family of Open Efficient Language Models from Apple Machine learning research lab. OpenELM uses a layer-wise scaling strategy to efficiently allocate parameters within each layer of the transformer model, leading to enhanced accuracy. OpenELM models were pretrained using the CoreNet library. Apple release both pretrained and instruction tuned models with 270M, 450M, 1.1B and 3B parameters. The pre-training dataset contains RefinedWeb, deduplicated PILE, a subset of RedPajama, and a subset of Dolma v1.6, totaling approximately 1.8 trillion tokens.

Mistral

Mistral offers Les Ministraux. These models set a new frontier in knowledge, commonsense, reasoning, function-calling, and efficiency in the sub-10B category, and can be used or tuned to a variety of uses, from orchestrating agentic workflows to creating specialist task workers. Both models support up to 128k context length (currently 32k on vLLM) and Ministral 8B has a special interleaved sliding-window attention pattern for faster and memory-efficient inference.

SmolLM

SmolLM, a series of compact yet powerful language models, has been developed by Hugging Face. Available in three sizes (135M, 360M, and 1.7B parameters), these models are trained on Cosmo-Corpus, a carefully curated dataset. This dataset includes Cosmopedia v2 (synthetic textbooks and stories generated by Mixtral), Python-Edu (educational Python samples), and FineWeb-Edu (deduplicated educational web content). SmolLM models have demonstrated impressive performance in common sense reasoning and world knowledge tasks when compared to other models of similar size.

Meta

Meta is pushing and leading the open source movement. They just released Llama 3.2. Llama 3.2 includes multilingual text-only models (1B, 3B) and text-image models (11B, 90B), with quantized versions of 1B and 3B offering on average up to 56% smaller size and 2-3x speedup.

These models can fine-tuned and distilled fully.

BigCode

BigCode is an open scientific collaboration dedicated to the ethical development and application of large language models for code. Supported by ServiceNow and Hugging Face, BigCode fosters open governance within the machine learning and open source communities.

StarCoder2-3B is a 3 billion parameter language model trained on a diverse dataset of 17 programming languages sourced from The Stack v2, excluding opt-out requests. Employing Grouped Query Attention and a sliding window mechanism, StarCoder2-3B boasts a context window of 16,384 tokens. Trained on over 3 trillion tokens using the Fill-in-the-Middle objective, this model demonstrates exceptional performance in code generation and understanding tasks.

Jina

Jina Reader-LM is a series of models that convert HTML content to Markdown content, which is useful for content conversion tasks. The model is trained on a curated collection of HTML content and its corresponding Markdown content.

Salesforce

Salesforce's Open-Source SLMs: A Range of Options
Salesforce has introduced a suite of open-source Sequence-to-Sequence Language Models (SLMs) designed to meet various needs.

Tiny (xLAM-1B): This "Tiny Giant" model, with its 1 billion parameters, is perfect for on-device applications where larger models are impractical. Its compact size makes it ideal for creating powerful AI assistants that can run locally on smartphones or other devices with limited computing resources.

Small (xLAM-7B): Designed for quick academic exploration, the 7B model offers a balance between performance and resource consumption. It's suitable for planning and reasoning tasks in agentic applications where a lightweight environment is preferred.

Medium (xLAM-8x7B): This 8x7B mixture-of-experts model is tailored for industrial applications that require a balance of latency, resource consumption, and performance.

LG AI Research

LG AI Research provides a family of Models called EXAONE. EXAONE stands for EXpert AI for EveryONE, a vision that LG is committed to realizing. EXAONEPath is a patch-level pathology pretrained model with 86 million parameters. The model was pretrained on 285,153,903 patches extracted from a total of 34,795 WSIs. EXAONEPath demonstrates superior performance considering the number of WSIs used and the model's parameter count. This model is open sourced.

Exclusive Interview

Devvret Rishi

CEO and Cofounder

Predibase

What inspired you to found Predibase, and what gap in the market did you aim to address?

We started Predibase in 2021 with the mission to democratize deep learning. At that time, we saw that leading tech companies like Google, Apple, and Uber—where my co-founders and I previously worked—were leveraging neural network models, especially large pre-trained ones, to build better systems for tasks like recommendation engines and working with unstructured data such as text and images. However, most companies were still relying on outdated methods like linear regression or tree-based models. Our goal was to democratize access to these advanced neural networks.

We built Predibase on top of an open-source project my co-founder Pierre had started while at Uber. Initially, we believed the way to democratize deep learning would be through platforms like ours, but we were surprised by how quickly the field evolved. What really changed the game was the emergence of models with massive parameter counts, like transformers. When scaled up by 100x or 1000x, these models gained emergent generative properties. Suddenly, engineers could interact with them simply by prompting, without any initial training.

Our platform initially focused on fine-tuning models like BERT in 2021-2022, which were considered large at the time. But as generative AI evolved, we saw that engineers needed more than just pre-trained models—they needed a way to customize them efficiently. This reinforced our original vision. While we initially focused on democratizing deep learning through fine-tuning, we realized that the need for customization platforms like Predibase had only grown stronger.

Your results seem almost magical; how do you do it?

The core of our success comes from recognizing that machine learning has fundamentally changed. Five years ago, the way you trained models was by throwing a lot of data at them, training from scratch, and waiting hours or days for the process to converge. While training and fine-tuning aren't going away, there has been a fundamental shift in how models are trained. The biggest trend driving this shift is the technical innovation behind Low-Rank Adaptation (LoRA). LoRA introduced the idea that you can modify only a small fraction of a model's parameters—typically less than 1%—and still achieve the same level of performance as if you had fine-tuned all 7 billion parameters. This approach allows the model to behave and perform at a high level while being much more efficient.

Many customers assume that training or fine-tuning models will take days and cost tens of thousands of dollars. In contrast, with Predibase, we can fine-tune most models in 30 minutes to an hour for as little as $5-$50. This efficiency empowers teams to experiment more freely and reduces the barriers to building custom models.

So I think the magic in our results is really threefold:
The first key insight we had was recognizing that the way models are trained would change significantly. We fully committed to parameter-efficient fine-tuning, enabling users to achieve high-quality results much faster and with a much smaller computational footprint.

The second step was integrating parameter-efficient training with parameter-efficient serving. We used LoRA-based training and LoRA-optimized serving through our open-source framework, LoRAX. LoRAX allows a single deployment to support multiple fine-tuned models, which means you can achieve excellent results by having many specialized fine-tunes—perhaps one per customer—without significantly increasing serving costs.

The final ingredient behind our success is a lot of hard work and benchmarking. We've fine-tuned hundreds of billions of tokens on our platform and tens of thousands of models ourselves. This hands-on experience has given us deep insights into which parameter combinations work best for different use cases. When a customer uploads a dataset and selects a model, we have prior knowledge of how to train that model most effectively—what LoRA rank to use, how large the model should be, and how long to train it. It all comes down to being empirical, and our extensive research, including the Predibase Fine-Tuning Leaderboard, has been baked into the platform to make this process seamless for users.

Where/when does your solution deliver the best results?

Our platform delivers the best results for specialized tasks. As one of our customers put it, "Generalized intelligence might be great, but we don't need our point-of-sale assistant to recite French poetry."

We've seen this in our Fine-Tuning Leaderboard as well, which shows that fine-tuned models excel at handling specific, focused tasks. LoRA-based fine-tuning and serving are especially effective in these scenarios,

enabling organizations to achieve high-quality results tailored to their needs. This approach ensures they get the precision they require without the unnecessary overhead of larger, general-purpose models.

How does your solution help address the huge cost of running LLMs?

We've built over 50 optimizations into our fine-tuning stack, incorporating the latest findings from the research community. These optimizations allow you to fine-tune models with minimal resources while still achieving high-quality results. As a result, fine-tuning can typically be completed in minutes or hours–not days–for just $5 to $50, a fraction of what traditional methods would cost.

On the inference side–where a typical organization allocates most of their sped–we tackle costs with GPU autoscaling, so you only pay for the compute you use. Turbo LoRA ensures models are optimized for fast inference with low latency, and our LoRAX framework allows multiple fine-tuned models to run from a single GPU. This means you can efficiently serve fine-tuned models from fewer GPUs, helping keep your infrastructure costs low while supporting high-volume real-time workloads.

Large enterprises are very concerned about data security and IP, how do you address this?

We get it—data security and IP protection are top priorities, especially for enterprises handling sensitive information. That's why we offer the ability to deploy Predibase in your Virtual Private Cloud or in our cloud. This ensures that data stays under your control, with all the security policies you need, including SOC II Type II compliance. Whether you're in finance, healthcare, or any other regulated industry, you can fine-tune and deploy models with the confidence that your data and IP are safe.

How easy/complicated is it to use Predibase?

You can get started with Predibase in as few as ~10 lines of code. Whether you're an engineer or a data scientist, our platform abstracts away the complexities of fine-tuning and deploying models. You can get started through our web interface or SDK, upload your dataset, select a model, and kick off training in no time. We've built Predibase to make fine-tuning as simple as possible, so teams can focus on outcomes instead of wrestling with infrastructure.

Inference speed is key in many use cases, how does Predibase help with that aspect?

Predibase boosts inference speed with Turbo LoRA, which increases throughput by up to 4x, and FP8 quantization, which cuts the memory footprint in half for faster processing. On top of that, the LoRAX framework lets multiple fine-tuned models run on a single GPU, reducing costs and improving efficiency. With GPU autoscaling, the platform adjusts resources in real-time based on demand, ensuring fast responses during traffic spikes without overpaying for idle infrastructure. This combination guarantees fast, cost-effective model serving, whether for production workloads or high-volume AI applications.

How fast is the payback on the fine-tuning initial cost?

The payback on fine-tuning with Predibase is incredibly fast because LoRA fine-tuning is remarkably cheap compared to full fine-tuning. Many people still assume that fine-tuning is expensive, imagining the high costs of full model retraining—but with LoRA, fine-tuning typically costs only $5 to $50 for a job, making it a low-risk, high-return investment. With Predibase, enterprises can fine-tune efficiently without running dozens of expensive, time-consuming experiments. This enables rapid deployment of specialized, high-performing models.

How are you different from other fine tuning providers?

Predibase stands out with a comprehensive fine-tuning platform that just works—no out-of-memory errors while training or unexpected drops in throughput while serving. We've built 50+ optimizations directly into our stack to ensure smooth, high-performance fine-tuning. Combined with LoRAX–which lets you efficiently serve hundreds of fine-tuned adapters on a single GPU–our Turbo LoRA, FP8 quantization, and GPU autoscaling make our model serving infrastructure industry-leading, delivering faster responses at lower costs.

We've seen too many teams get bogged down managing infrastructure, building data pipelines, and debugging fragmented open-source tools—leaving less time to actually build and productionize AI. That's why we provide an end-to-end platform backed by a dedicated team of ML engineers to help you every step of the way. Whether you prefer the flexibility of SaaS in our cloud or full control with VPC deployments in yours, Predibase frees you from the operational burden, so you can focus on delivering impactful AI solutions.

What are some of the companies that you're working with and what problem are they solving with SLMs?

Checkr leverages Predibase to improve the accuracy and efficiency of background checks. They process millions of checks monthly, but 2% of the data in one part of the background check workflow—often messy and unstructured—needed human review. With Predibase, Checkr fine-tuned a small language model, achieving 90%+ accuracy, outperforming GPT-4, and reducing inference costs by 5x. This enabled them to replace manual review with real-time automated decisions, meeting tight latency SLAs and improving customer experience.

Convirza, on the other hand, processes over a million phone calls per month to extract actionable insights that help coach call agents. Previously, managing infrastructure for their AI models was complex and often too much of a burden for their small AI team. With Predibase's LoRAX multi-adapter serving, they're able to consolidate 60 adapters into a single deployment, reducing overhead and allowing them to iterate on new models much faster. This efficiency lets them focus on building AI solutions, not infrastructure, unlocking new capabilities for their customers, like creating bespoke call performance indicators on the fly.

Both companies highlight how small language models fine-tuned on Predibase outperform larger models while cutting costs, improving response times, and streamlining operations.

How do you see the industry evolving?

There are two big wars happening in generative AI infrastructure. The first is the competition between small, fine-tuned language models and large, general-purpose models. The second is the battle between open-source and commercial solutions.

The question that comes up a lot is: will the future be about small, task-specific, fine-tuned models, or large, general-purpose ones? I'm convinced it's going to be more and more about small, fine-tuned models and we've already seen this shift starting. In 2023, the market's focus was all about making models as big as possible, which worked well for quick prototyping. But as companies move into production, the focus shifts to cost, quality, and latency.

A lot of studies have pointed out that the economics of Gen AI haven't always added up—too much spend, too little benefit. You can't justify spending billions on infrastructure to solve relatively simple automation tasks. That's where smaller, task-specific models come in. As teams graduate from prototyping into production, these models will grow in importance.

And if you look at organizations using Gen AI seriously at scale, almost all of them follow this path as they mature. It's the same reason OpenAI felt the need to roll out something like GPT-4o-mini. I think this trend will continue, and it's a good thing for the industry because it forces costs to align with ROI.

Talking about the second trend, my view is that the entire pie for both open-source and commercial models will grow very quickly, but the relative share of open-source is going to grow much faster than the commercial side. Based on an A16Z Generative AI survey from 2023, people were looking to spend a lot on LLMs, especially in the enterprise segment. But in 2023–the year of prototyping, as many people say–80 to 90% of the usage was estimated as closed source. However, two-thirds of AI leaders have expressed plans to increase their open-source usage, targeting a 50/50 split.

Historically, most machine learning has been built on open-source architectures, so this shift aligns with the broader trajectory of the industry.

What problems are left unsolved and where do you see the greatest opportunity?

I think the biggest unsolved problem—and one I find really exciting—is how to create a flywheel where models get better as they're used. What I mean is introducing a real active learning process for LLMs. Right now, what I hear from organizations is that when they move to production, they can often get a model to 70% accuracy with prompt engineering alone. But as they try to push further, they only see marginal improvements—maybe going from 70% to 71%.

What they really want is a way to reach 80% or 90% accuracy, and they hope that by deploying the model, they can collect enough data to keep improving it. But that workflow isn't solved yet. The way many companies handle it now is by releasing a model at 70%, collecting production data, manually reviewing it, and then fine-tuning the model based on annotated datasets. But this approach just doesn't scale—there's no way to manually review enough data, especially as LLMs handle millions of queries in production.

The real opportunity, in my opinion, lies in building a system where models can improve automatically over time. For example, if a model launches with 70% accuracy in a new domain, you need a way to leverage production data to fine-tune it iteratively. I think the key will be applying some of the breakthroughs we're already seeing—like using LLMs as judges or generating synthetic data—to create that flywheel. With such a system, a model could launch at 50-70% accuracy, collect data from real use, and improve on its own.

This idea was partially realized in recommender systems, but it hasn't yet been achieved with generative AI at scale. That's where I think the industry is headed, and it's where I see the most exciting potential for growth.

AI's Big Lie:
Why Smaller Models are Smarter (and Cheaper)

Introduction

Small Language Models (SLMs) are changing the game for enterprises, offering faster and cheaper alternatives to massive Large Language Models (LLMs) and custom-built models. However, the real magic lies in LoRA-based fine-tuning, where adjusting less than 1% of a model's parameters can unlock nearly 99% of the performance of a fully customized model.

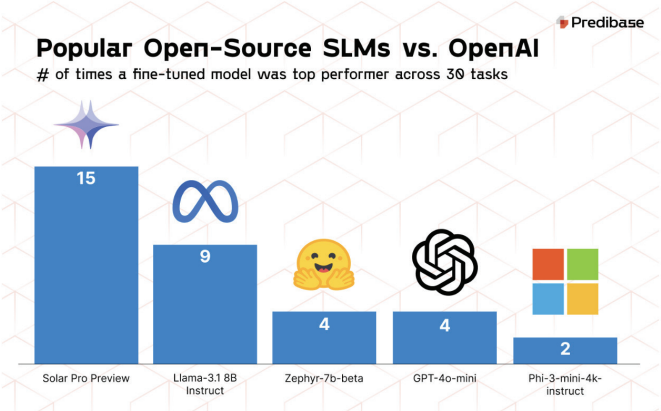
While fine-tuning is easy to get started with, it's harder to master. Many enterprises dive in expecting quick results, only to realize how challenging high-quality fine-tuning and production deployment can be. While anyone can fine-tune an open-source model, few have the expertise and infrastructure to take it all the way to production-ready AI.

The trend is clear: smaller, fine-tuned models will drive the next wave of enterprise AI, delivering performance tailored to specific business needs at a fraction of the cost of larger models.

Anyone Can Fine-Tune. Not Everyone Can Bring Fine-Tuning to Production.

While the idea of fine-tuning is appealing, getting an open-source model to production requires overcoming significant challenges. Many enterprises underestimate the effort involved—OSS models don't work out of the box, and building the infrastructure for fine-tuning and serving these models is no easy task. Open-source models offer great flexibility, but they need fine-tuning to deliver real-world performance. To unlock their potential, enterprises must bring their own data to adapt the model to their domain. On top of that, fine-tuning infrastructure is critical—without the right tools and resources, even promising OSS models can fall short of expectations.

Fine-tuning bridges the gap between general-purpose models and high-performing, domain-specific solutions. When done well, fine-tuned models outperform larger, pre-trained alternatives.

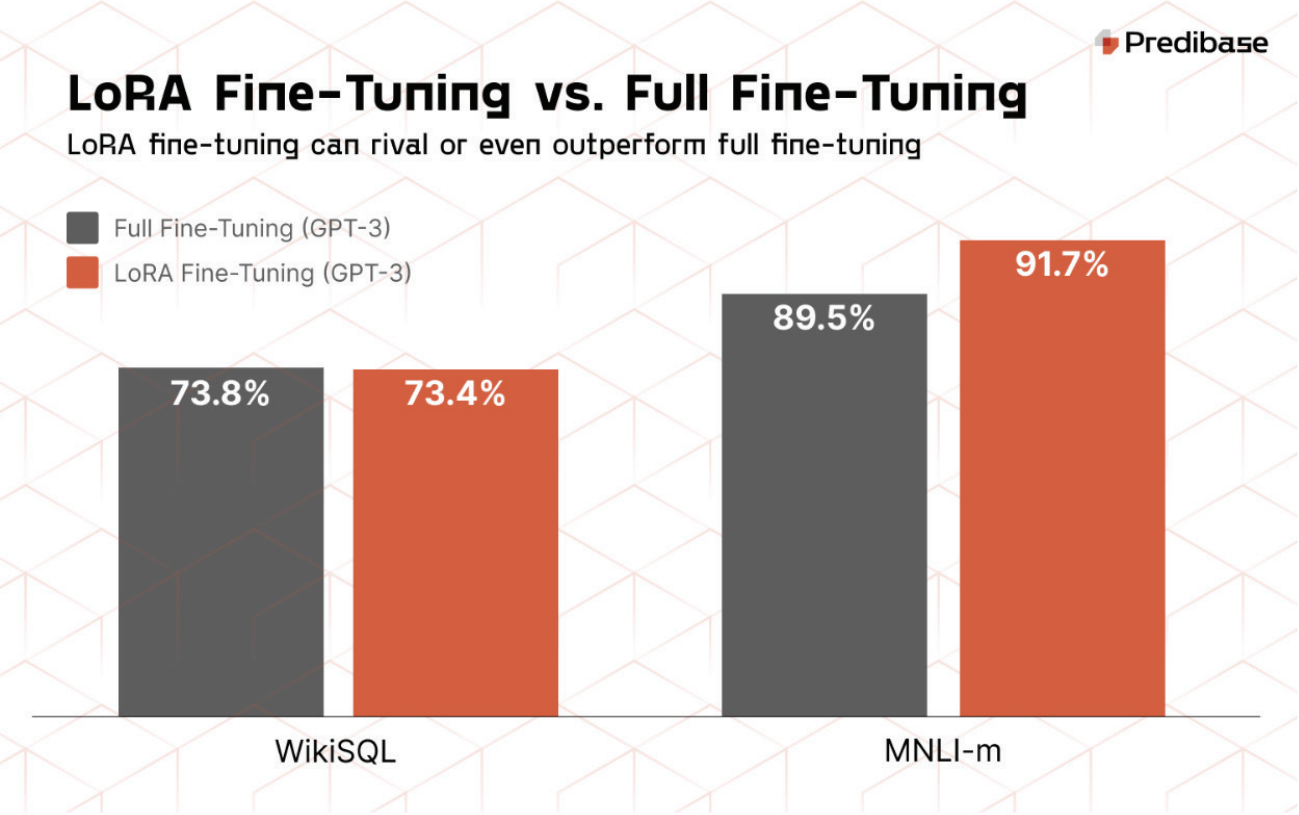


Example: *Predibase's Fine-Tuning Leaderboard demonstrates how open-source models, when properly fine-tuned, outperform GPT-4 and fine-tuned GPT-4o-mini on specialized tasks*

LoRA Fine-Tuning Leads the Way

LoRA-based fine-tuning is transforming the way enterprises customize language models. By adjusting fewer than 1% of a model's parameters, LoRA fine-tuning delivers nearly the same performance as a fully trained custom model, but with a fraction of the effort, time, and cost. This technique has become the industry's go-to method, as it allows companies to efficiently tailor models to specific use cases without requiring massive compute resources or long training cycles.

Traditional fine-tuning of large models can be time-consuming, expensive, and resource-heavy, often requiring access to multiple GPUs for days or weeks. LoRA overcomes these barriers by freezing most of the model's parameters and only training lightweight adapters, making it faster and cheaper to update models for specific domains. This means that organizations no longer need to choose between performance and cost-efficiency—they can have both.

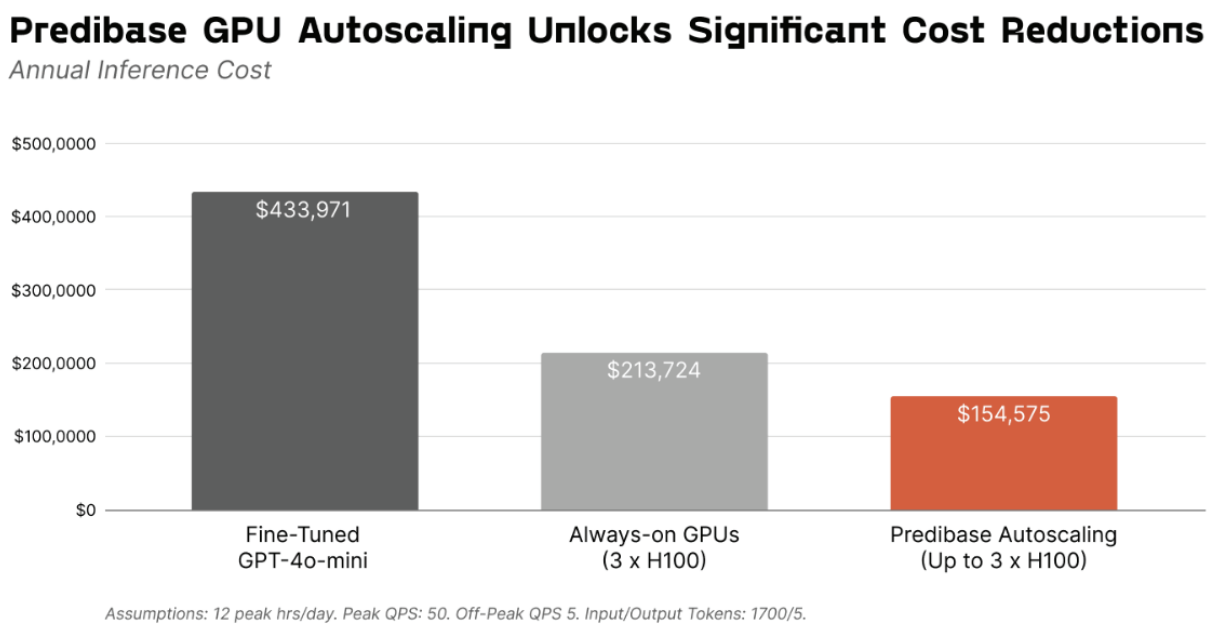


Infrastructure is Critical

Fine-tuning alone won't unlock the full potential of small language models—reliable and scalable infrastructure is essential to bring fine-tuned models into production and maintain their performance at scale. A well-built inference stack ensures models can operate efficiently, scale seamlessly, and meet enterprise requirements without sacrificing speed, security, or cost-efficiency. Here are a few key features to consider:

Multi-Model Serving: The ability to run multiple fine-tuned models on a single GPU is crucial for cost savings and efficient scaling. LoRA eXchange (LoRAX) is our framework that maximizes GPU utilization by eliminating the need for dedicated hardware for each model, allowing organizations to deploy and serve domain-specific models without adding complexity. Multi-model serving helps streamline infrastructure, making it easier to handle multiple use cases simultaneously.

GPU Autoscaling: An effective inference stack must include GPU autoscaling to ensure that resources match demand. During peak usage, autoscaling provides the necessary GPU capacity to maintain performance, while during off-peak hours, it scales down to zero, minimizing infrastructure costs. This flexibility ensures that companies only pay for the resources they use, preventing overprovisioning while guaranteeing optimal performance.



Fast Throughput: For inference workloads, speed matters. Optimizations like Speculative Decoding and FP8 quantization can 4x throughput compared to base models, ensuring fast, cost-efficient performance. These optimizations are essential for applications where real-time or low-latency responses are required, such as customer service bots, fraud detection systems, or document automation tools.

Enterprise Readiness: To meet the needs of modern businesses, an inference stack must be enterprise-ready, ensuring scalability, security, and flexibility. This includes multi-cloud support to avoid vendor lock-in, VPC deployments for enhanced privacy, and compliance with SOC II standards to ensure security and trust. Additionally, multi-region high availability is essential to keep models accessible across geographic regions, minimizing downtime, improving fault tolerance, and ensuring low-latency responses by serving users from the closest data centers. With these capabilities, businesses can confidently scale AI deployments, ensuring continuous operations even during disruptions.

Building and maintaining this type of infrastructure can be complex and resource-intensive, requiring expertise in orchestration, GPU management, and model optimization. *Predibase's Inference Engine* was built with these challenges in mind, offering a next-generation stack designed for enterprises deploying fine-tuned models at scale.

How Checkr and Convirza Cut Costs and Boost Performance with Fine-Tuned Models

Real-world use cases demonstrate how enterprises can unlock the full potential of fine-tuned SLMs with the right combination of infrastructure and customization. Both Checkr and Convirza serve as prime examples of how adopting smaller, fine-tuned models—as opposed to large pre-trained LLMs—can transform AI initiatives.

These companies faced common challenges: needing faster, more accurate models while keeping infrastructure costs manageable. By leveraging Predibase's platform, they overcame the inherent limitations of open-source models and deployed fine-tuned, domain-specific solutions that outperformed more resource-heavy models.

Checkr: Leveraged Predibase's platform to fine-tune small language models for background checks, achieving higher accuracy, lower latency, and a 5X cost reduction compared to traditional LLM approaches. *Learn more.*

Convirza: Deployed nearly 60 fine-tuned adapters on a single base model deployment that scales from zero to 10 GPUs based on demand, significantly reducing infrastructure costs.

Predibase's Competitive Edge

Predibase stands out by simplifying the entire fine-tuning and deployment process, offering tools designed to combine ease of use with enterprise-scale performance. Whether you're a team with deep AI expertise or one just starting out, the platform's intuitive interface and SDK make it easy to fine-tune models without requiring extensive in-house knowledge.

In addition to its accessibility, Predibase provides flexibility through SaaS and VPC options, giving companies the ability to tailor deployments to meet their specific security, compliance, and customization needs. This dual delivery model ensures that organizations can choose the right infrastructure setup to align with their operational requirements.

Finally, the platform enables the use of small language models fine-tuned and optimized for specific business needs. By focusing on deploying models that are neither too large nor too complex, Predibase helps companies maximize performance and efficiency without unnecessary infrastructure costs or complexity.

Conclusion

Smaller, fine-tuned models are the future of enterprise AI, offering a compelling combination of performance, cost-efficiency, and scalability. However, success requires more than just fine-tuning—it demands reliable infrastructure and expertise to take models from experimentation to production.

Explore Predibase's platform and *Fine-Tuning Leaderboard* to unlock the potential of customized, high-performance SLMs and stay ahead in the evolving world of AI.

Devvret Rishi

CEO and Cofounder

BIO

Devvret is the **CEO and Cofounder of Predibase.** Prior he was an ML product leader at Google working across products like Firebase, Google Research and the Google Assistant as well as Vertex AI. While there, Dev was also the first product lead for Kaggle – a data science and machine learning community with over 8 million users worldwide. Dev's academic background is in computer science and statistics, and he holds a masters in computer science from Harvard University focused on ML.

Predibase

Fine-Tune and Deploy the Future

Unlock the Power of Specialized AI with Predibase

Build Smarter. Deploy Faster. Scale Effortlessly.

Harness the power of small language models (SLMs) without the infrastructure headache. Predibase makes it easy to fine-tune and deploy specialized models that perform at scale.

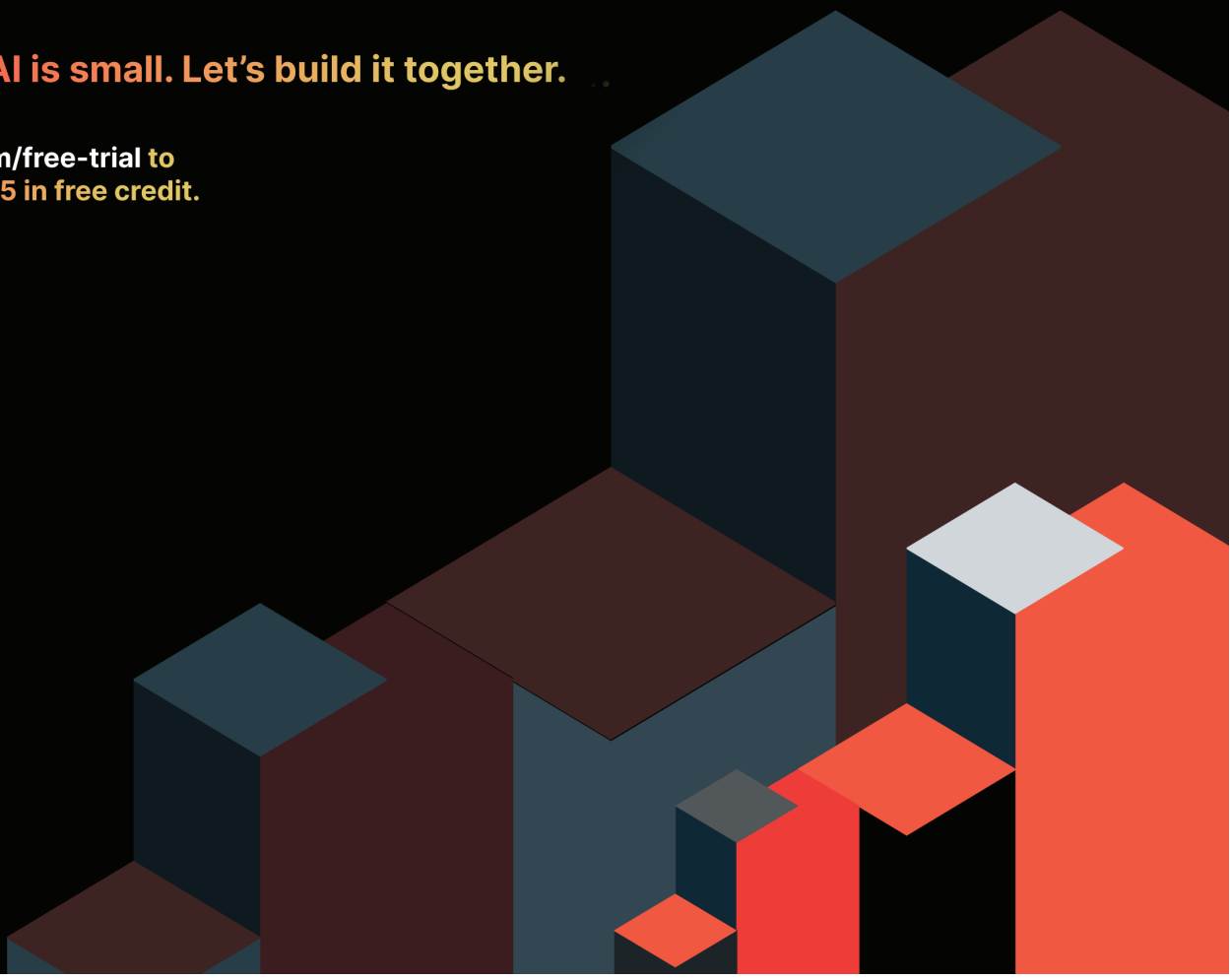
Multi-Model Serving: Deploy multiple fine-tuned models on a single GPU with LoRAX for seamless, cost-effective switching.

Autoscaling GPUs: Scale resources automatically to ensure high throughput and low latency without the hassle or overhead.

Turbo LoRA + FP8: Achieve 4x faster throughput with optimized performance, lower memory use, and reduced costs for high-volume AI workloads.

The future of AI is small. Let's build it together.

Visit predibase.com/free-trial to get started with $25 in free credit.



Top Small Language Models (SLMs)

Qwen

Qwen2 1.5B Instruct
Qwen2 2B Instruct
Qwen1.5 1.8B
Qwen1.5 1.8B Chat
Qwen2 Math 1.5B Instruct
Qwen2 Math 1.5B
Qwen2.5

01-ai

Yi Coder 1.5B Chat

Microsoft

Phi 3.5 Mini Instruct
Phi 3 Mini 128K Instruct
Phi 2
Phi 1 5

Google

Gemini Nano
Gemma 2
Gemma 2 2B It
Gemma 2 2B
Gemma 1.2B It
Recurrentgemma 2B It
Recurrentgemma 2B
CodeGemma 2B
Shieldgemma 2B

TinyLlama

TinyLlama
TinyLlama 1.1B Chat V1.0
TinyLlama 1.1B Intermediate Step 1431K 3T
TinyLlama 1.1B Intermediate Step 1195K Token 2

Numind

NuExtract

IBM

PowerLM 3B

Princeton-nlp

Sheared LLaMA 1.3B

Predibase

Lots

h2o

H2O-Danube2-1.8B
H2o Danube2 1.8B Chat
H2o Danube2 1.8B Chat

Sarvamai

Sarvam 2B V0.5

Anakin87

Phi 3.5 Mini ITA

Allenai

OLMoE 1B 7B 0924
OLMo 1B Hf
OLMoE 1B 7B 0924 Instruct

jpacifico

Chocolatine 3B Instruct DPO Revise
Chocolatine 3B Instruct DPO V1.2
Chocolatine 3B Instruct DPO V1.0

Alibaba-NLP

Gte Qwen2 1.5B Instruct

Ba2han

Llama Phi 3 DoRA

Top Small Language Models (SLMs)

Apple

OpenELM
OpenELM 3B Instruct

Jina Ai

Reader-LM
Reader Lm 1.5B

Mistral

Mistral small
Mistral NeMo instruct

Salesforce

xLAM-1B
XLAM 1B Fc R

Stabilityai

Stablelm 3B 4e1t
Stablelm Zephyr 3B

Bigcode

Starcoder2 3B

Cognitivecomputation

Dolphin 2.9.4 Gemma2 2B

dunzhang

Stella En 1.5B V5

HuggingFace

SmolLM 1.7B
SmolLM 1.7B Instruct

katuni4ka

Tiny Random Codegen2
Tiny Random Snowflake
Tiny Random Dbrx

Meta

Llama-3.1-8B
Llama-3.2-3B
Llama-3.2-1B

VAGOsolutions

SauerkrautLM Gemma 2 2B It

Syed-Hasan-8503

Phi 3 Mini 4K Instruct Cpo Simpo

Rasyosef

Phi 2 Instruct V0.1

Artples

L MChat Small

DeepMount00

Qwen2 1.5B Ita

Mikayel Harutyunyan

Head of Marketing & Growth, Activeloop

BIO

Mikayel Harutyunyan is passionate about marketing, human behavior research, and AI. His journey includes leading developer marketing at Activeloop, launching YouTube Music in Central Europe, and contributing research to combat the impact of fake news. Mikayel's work has been published in journals like Nature Human Behavior and Affective Science, earning him recognition as one of Czechia's top social science researchers under 33, with over 300 citations. He has also launched GenAI360, a popular series of Generative AI courses, along with an eponymous weekly AI newsletter with over 45,000 subscribers.

How has your company DNA been shaped by the academic journey of Davit, the founder and CEO of Activeloop? What motivated him to focus on building AI infrastructure?

Davit's journey into AI infrastructure began during his time as a ML researcher at Princeton Neuroscience Lab, where he faced firsthand the complexities of managing datasets for training machine learning models. He often found himself spending an excessive amount of time not only on model training but also dealing with the logistics of data handling—retrieving, cleaning, and transforming data. This became a significant bottleneck to innovation (and was extremely costly to the lab he was working at), leading him to question how to make data handling as seamless as possible, allowing researchers and developers to focus on building and improving AI models.

This realization sparked the creation of Activeloop. Davit wanted to build the future of AI data—a powerful vision aimed at empowering everyone with the tools to access, visualize, and manage data effortlessly, thereby accelerating AI innovation. This is how Deep Lake was born—addressing the need for scalable and efficient data infrastructure for machine learning, inspired by his experiences as a researcher needing better ways to handle data. Activeloop's goal has always been to create infrastructure that not only makes data management easier but also provides new ways to interact with and derive insights from data, ultimately bridging the gap between raw data and AI-driven insights. This vision continues to guide Activeloop's mission to this day.

Deep Lake, your company's flagship product, focuses on fast & accurate search on object storage. In addition, you enable fast data streaming to GPUs to train & fine-tune models at low cost. What is special about Deep Lake architecture that enables this, particularly in comparison to in-memory databases?

Deep Lake: The Database for AI

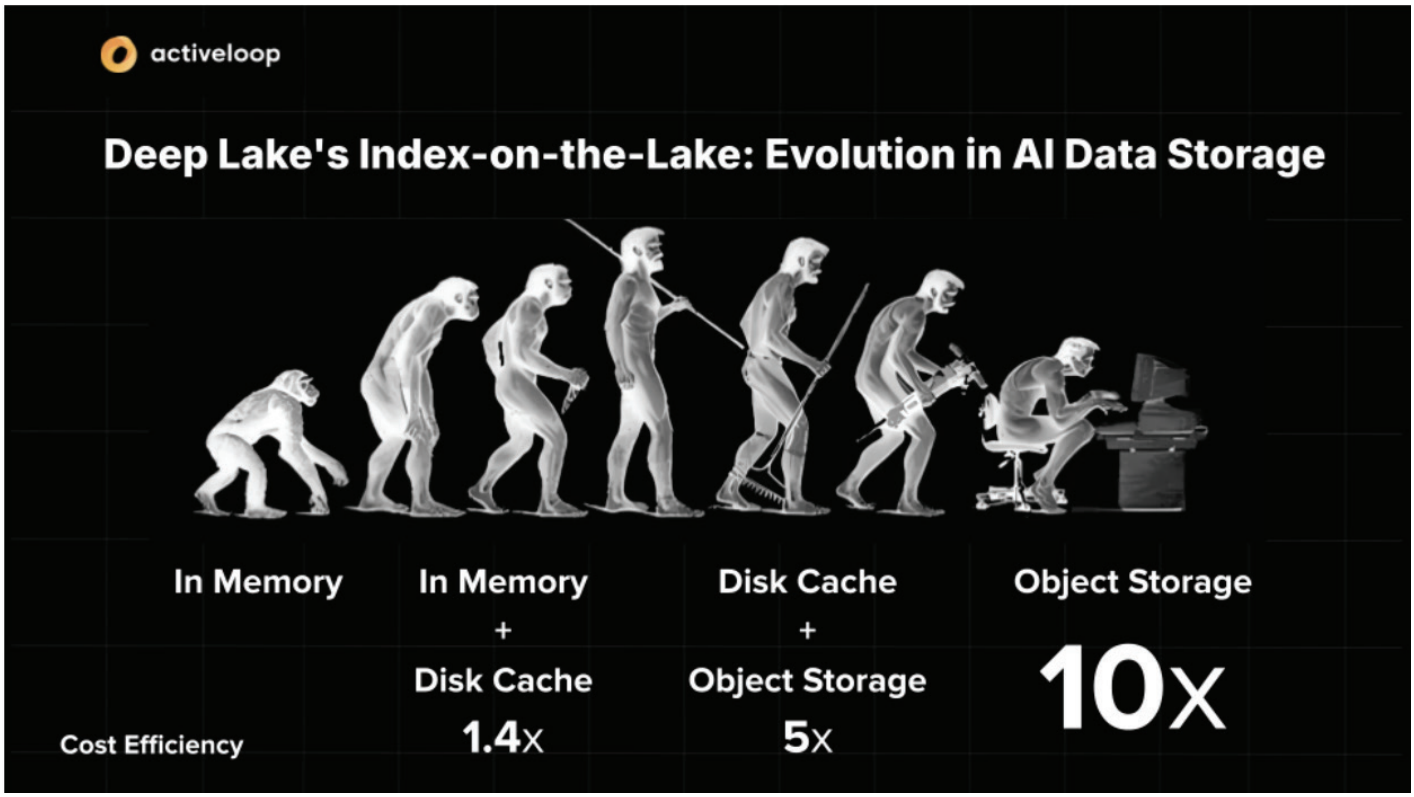
Deep Lake is the database for AI, built specifically for the unique needs of AI training, particularly with large and multi-modal datasets. According to McKinsey's 2023 report on the state of AI, generative AI adoption is growing rapidly, with 55% of companies now using AI in at least one business function, highlighting the need for robust and scalable data infrastructure like Deep Lake. Unlike traditional in-memory databases, which require entire datasets to be loaded into memory for processing, Deep Lake offers an index-on-the-lake approach. This means data is indexed directly within the object storage, allowing it to be accessed and processed on demand without the need to load everything into RAM. This is a completely new approach—previously, we also pioneered a less efficient method of streaming data from object storage to compute and materializing queries of unstructured data on-the-fly. With the release of Deep Lake 4.0, the index has also been offloaded to the lake, making this approach 10x more cost-efficient, particularly for large-scale AI projects.

Multi-Modal Data Handling with Tensorial Format

A core part of Deep Lake's architecture that makes it truly multi-modal is its use of the tensorial format. This format is specifically designed to handle multi-dimensional data, which is essential for representing the wide variety of data types that AI models need to process—such as images, text, audio, and more. By storing data in tensor format, Deep Lake allows seamless integration of these diverse data types into a single pipeline, enabling complex, multi-modal workflows that are necessary for cutting-edge AI applications. The tensorial format also supports efficient data streaming to GPUs, optimizing the training process for models that need to work with various data modalities simultaneously. This stands in stark contrast to vector databases or legacy data storage solutions, which force users to frankenstein together various tools to mimic workflows that Deep Lake handles natively.

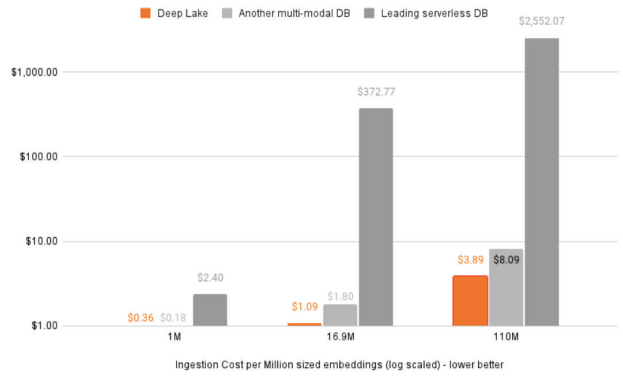
Overcoming Limitations of In-Memory Databases

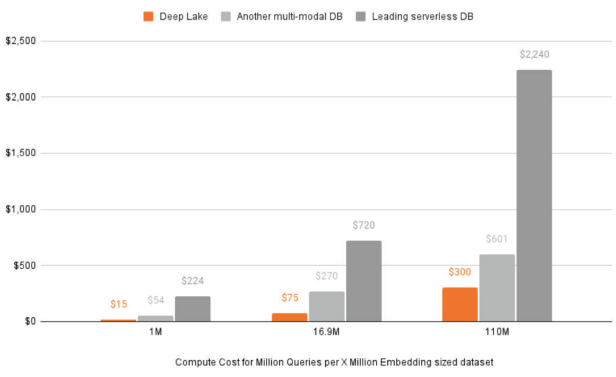
In-memory databases are often limited by the available memory, which restricts the size of datasets they can handle and makes them very costly to operate. Deep Lake overcomes this limitation by fetching data in real-time from object storage, enabling the training of models on massive datasets without being restricted by memory capacity. The index-on-the-lake strategy also provides a more flexible and cost-efficient data retrieval process, optimizing both latency and cost. This technology also enables joining datasets across clouds and manipulating them as if they were saved in one location.



Scalability and Efficiency for LLMs with Deep Lake

The rapid adoption of AI across various business functions has made scalability and efficiency more crucial than ever. Deep Lake makes it possible to scale efficiently without being bound by memory constraints, significantly improving the speed at which models can be trained and queried—addressing the growing demands of modern AI systems. In the context of LLMs, where datasets are massive, the ability to perform both training and querying computations closer to where the data resides (i.e., data locality) helps in reducing latency and cost. Essentially, Deep Lake transforms the way data connects to AI, making both the training and querying processes much faster, cost-effective, and scalable. By avoiding the need to load entire datasets into memory, as well as offloading the index to the object storage, Deep Lake reduces the risks of bottlenecks and allows researchers to experiment more freely with different model architectures, which is crucial for innovation in AI. We see customers like Matterport using Deep Lake to rapidly iterate on model architectures and datasets—by swapping just two lines of code. We also see customers using Deep Lake to search through the entirety of YouTube to predict the most successful content, at a fraction of the cost (20-40x cheaper).





There's a growing interest in smaller language models due to their efficiency. How do you see language models evolving in the next few years?

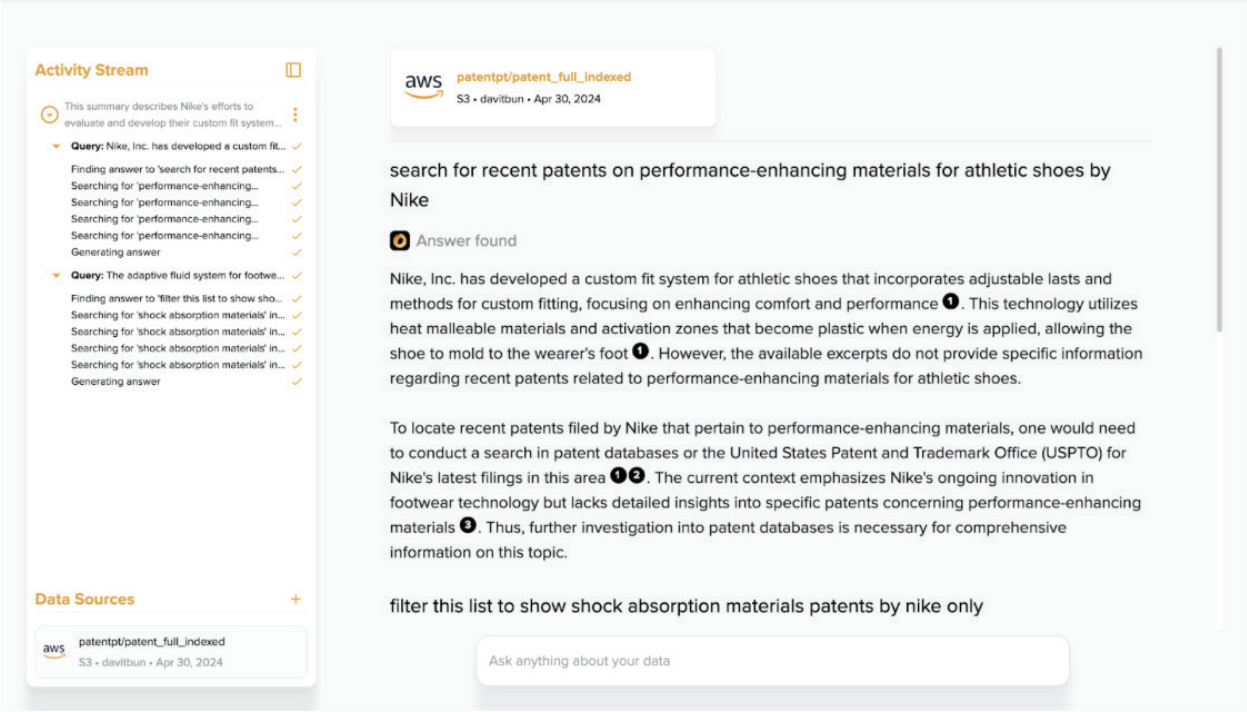
I believe smaller language models will become increasingly powerful and widespread, thanks to innovations in techniques like parameter-efficient fine-tuning, distillation, and the use of high-quality datasets. As highlighted in recent research, model robustness can be maintained even after pruning deeper layers, enabling smaller and more efficient models without sacrificing performance. For instance, a Meta/MIT team found that up to 50% of a model's layers could be pruned with negligible performance drop, while NVIDIA's research demonstrated that pruning layers, neurons, and attention heads, followed by efficient fine-tuning, could achieve significant reductions while maintaining effectiveness. Moreover, MINITRON models derived from Nemotron-4 15B achieved similar or superior performance to larger models like Mistral 7B and Llama-3 8B using up to 40x fewer training tokens.

The concept of distillation is also becoming more popular, as seen with Google's distillation of Gemini models, and other companies' similar efforts. Andrej Karpathy and others have argued that current large model sizes might be a reflection of inefficient training, and distillation helps address this inefficiency. For example, Llama 3.1 40B is being used for distillation, which helps improve smaller models' performance. These efforts are pushing the boundaries of what smaller language models can achieve with fewer resources, making them more practical for a variety of applications.

Interesting! What role does Activeloop play in supporting SLM development?

Activeloop plays a crucial role here by providing the data infrastructure that is nimble enough to handle the nuances of smaller language models. Our goal is to enable developers to easily access and manage high-quality datasets, optimize model training, and even facilitate deployment on edge devices where smaller models are key. Deep Lake's ability to support efficient data streaming, indexing, and retrieval means that small models can be trained and iterated upon quickly, which is crucial for their evolution. The flexibility of our platform allows researchers to experiment with various training approaches without worrying about data-related bottlenecks, which is especially important when dealing with smaller models that require careful optimization.

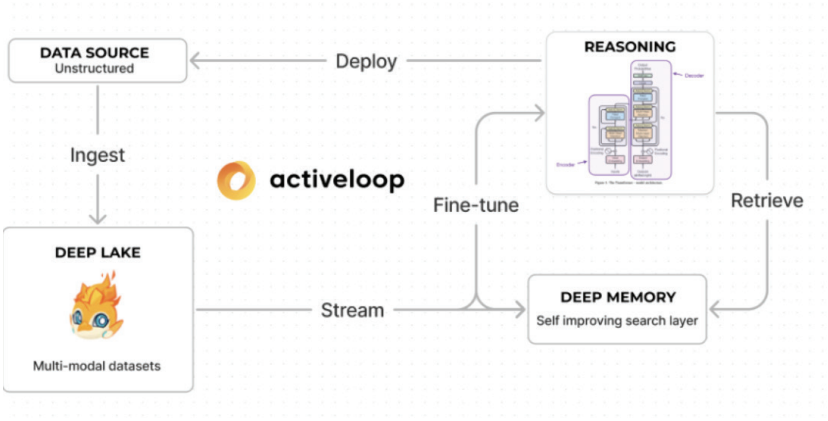
One example of our innovation is PatentPT, a solution specifically designed for patent search and generation. PatentPT leverages our unique data infrastructure to streamline the process of retrieving and analyzing patent information, making it more efficient and cost-effective. By integrating AI into patent-related workflows, PatentPT is a domain-specific SLM that helps enterprises generate and search for patents more accurately and quickly, reducing costs and improving productivity across the board.

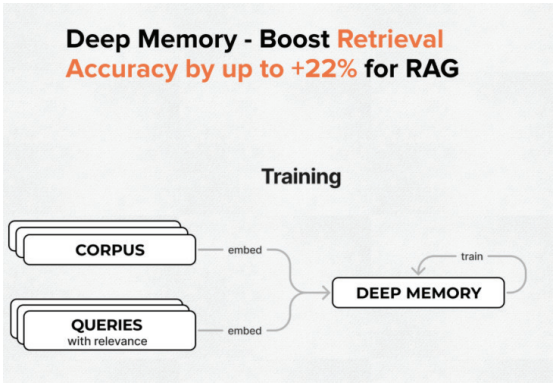


Mikayel, as a leader in AI, what is the top bottleneck that you're seeing in enterprise knowledge management, considering accuracy and cost, assuming security has been resolved?

Aside from security, the biggest bottleneck we see in enterprise knowledge management is the difficulty in achieving high accuracy at a reasonable cost when retrieving relevant information. In some cases, achieving high accuracy is crucial for ensuring that the right information is retrieved effectively. Accuracy directly impacts the reliability and usefulness of the insights generated by AI systems. Without a high level of accuracy, the risk of misinformation or incorrect decision-making increases, which can be especially costly and damaging in enterprise environments - and even cost lives (think in sensitive industries such as drug development).

With the introduction of Deep Memory, a feature of Deep Lake, our approach optimizes Retrieval-Augmented Generation (RAG) applications by increasing their accuracy by up to 22.5% on average without a significant increase in computational cost. Deep Memory acts like an internal Google search, learning which answers are likely to be most useful to the user, and bridges the gap between the embedding space of questions and indexed documents to serve the most relevant answers, without a need for a reranker or fine-tuned embeddings.





LLMs unlocked a new chapter in AI usage in enterprises. How should enterprises adapt their data infrastructure?

The rise of LLMs has opened a new chapter in AI for enterprises, fundamentally changing how they can derive value from data. To truly harness the potential of LLMs, enterprises need to rethink their data infrastructure. Traditional data management systems are not optimized for the type of streaming and multi-modal data required for effective AI training and deployment. Deep Lake, with its streaming-first architecture and native multi-modal support, provides enterprises with the agility needed to handle large-scale datasets directly from object storage. This minimizes latency, reduces costs, and maximizes scalability, enabling enterprises to leverage LLMs in a practical, efficient manner.

Enterprises must adapt by embracing data lakes that are designed with AI in mind—data lakes that allow data to be processed as needed, rather than being moved and transformed repeatedly. This kind of infrastructure is essential for enterprises aiming to deploy LLMs at scale, where the ability to perform computations closer to data and optimize workflows is key. By using Deep Lake, organizations can create a seamless flow of data from storage to model training, ensuring that they can take full advantage of the capabilities of LLMs while maintaining cost efficiency and performance.

How do you see the future of AI model deployment, particularly in the context of healthcare and life sciences?

Healthcare and life sciences are fields where the value of AI lies in unlocking scientific knowledge and accelerating discovery. By deploying AI models, these industries can transform the way research is conducted and how knowledge is applied in clinical settings. We jointly innovate with our customers - largest Fortune 500 MedTech companies, Healthcare AI leaders like Bayer or life sciences leaders like Flagship Pioneering. They have fundamentally rethought their approach to data infrasturcture, to derive more insights from the data they have, at scale.

One of the major challenges in healthcare and life

sciences is the sheer volume and the multi-modality of data generated—from patient records, MRIs and clinical trial data to 40M+ scientific papers on PubMed. Deep Lake's infrastructure is designed to handle such vast amounts of multi-modal data and retrieve it efficiently for AI. We leverage AI models for ingestion and and extraction of multi-modal relationships between models, and enabling researchers to focus on deriving insights rather than dealing with data logistics.

With Deep Lake, we're seeing teams build high-accuracy enterprise search across public research and private data - while staying compliant with relevant regulations. These internal search enginges then enable them to do things that would otherwise take large teams of analysts several months to complete.

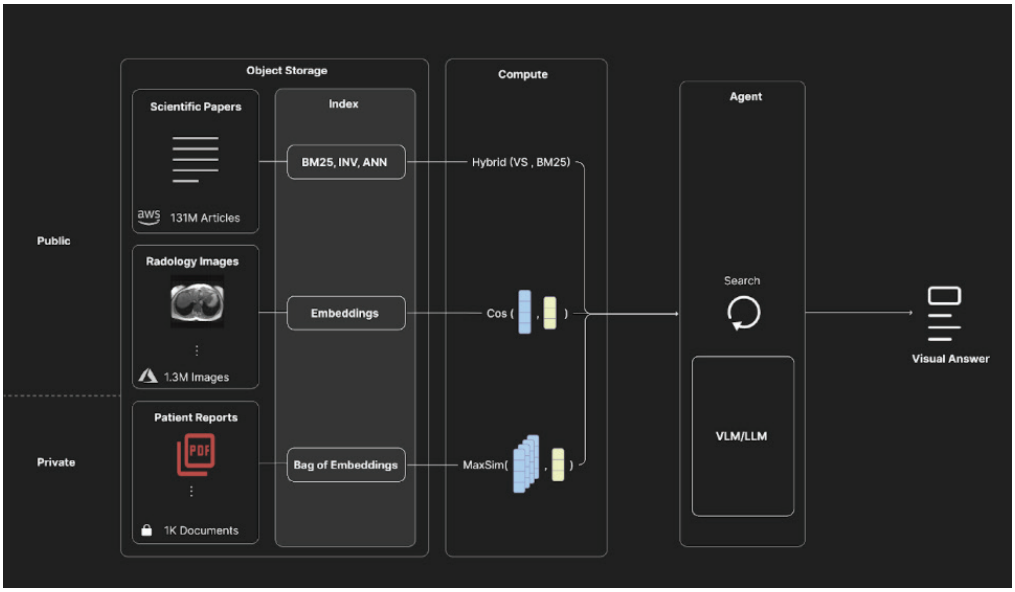
Finally, looking ahead, what are some emerging trends or challenges in AI development that you are most excited about tackling through Activeloop's offerings?

One of the biggest trends I'm excited about is the convergence of multi-modal learning and the increasing need for AI models that can seamlessly understand and integrate different types of data—text, images, video, and even sensor data. With Deep Lake, Activeloop has positioned itself at the heart of this challenge by offering a unified data infrastructure that can handle multi-modal datasets in an AI-native, tensorial format, thus accelerating the development of these advanced AI systems. The ability to work with multi-modal data will be crucial for the next generation of AI applications, which need to understand the world in a more holistic way.

Another area that excites me is unlocking knowledge at scale. With the exponential growth of data, the challenge lies not just in connecting it to AI, but in deriving meaningful insights efficiently and solving the 'last mile problem' - whether it's getting AI to do what you want, or getting it do it highly accurately, at a reasonable cost. We see that in areas like customer support, achieving 95% accuracy is not enough - it does still matter to that 1 in 20 cases aren't resolved correctly. In the 'least' worst case scenario - you get 260 nuggets in your McDonalds drive-thru order or convince a chatbot to sell you a car for $0. In actual worst case scenario - it can cost millions in wasted R&D and human lives. With solutions like Deep Lake, companies can build accurate knowledge retrieval engines with AI. This means that organizations can transform raw data into actionable insights that drive innovation and return on investment across industries.

Finally - it's end-to-end neural search. At our conference in October, RetrieveX, we've unveiled end-to-end neural search, which refers to a search mechanism where the entire process, from multi-modal data ingestion, to query formulation to result retrieval, is driven by neural networks. Unlike traditional search systems that rely on manual feature engineering or keyword matching, as well as clunky OCR pipelines that leave a lot of insights 'on the table', end-to-end neural search leverages deep learning models to fully comprehend the data, the question user is trying to answer with it, plan and execute a complex query that will address everything.

It's time databases powering AI became powered by AI themselves. Over the years, databases have evolved—from simple storage systems to sophisticated solutions that support large-scale, real-time data processing. The next step in this evolution is AI-powered databases that adapt and learn, making them not only storage and retrieval systems but intelligent data foundations. Databases must evolve to be powered and continually improved by AI—and I'm excited to see Activeloop leading the charge in this.



LLMWare.ai: Enabling Decentralized AI on the Edge

BIO

Darren Oberst is the Co-Founder and CTO of LLMWare, an innovative AI framework using small specialized models to revolutionize the landscape of AI application development for on-device and private deployment for financial services and other regulated industries.

Prior to LLMWare, Darren worked in senior roles in large tech companies for over 20 years. Darren founded and grew HCL Software, and served in senior leadership roles at IBM in Software, Services and the Financial Services vertical.

Darren is currently focused on building a pioneering enterprise AI application platform which includes retrieval augmented generation with AI agents using fine-tuned small specialized models that can be deployed locally and privately.

Darren graduated from UC Berkeley with Highest Honors in Physics and Philosophy and Harvard Law School with Honors.

The Next Wave of AI: Decentralized, Secure, and Cost-Effective Deployment

Recent advancements in small language models (SLMs), combined with new AI-enabled hardware capabilities in PCs (AI PCs), have converged to make AI deployment at the edge a practical reality, unlocking breakthrough opportunities to deploy high-quality generative AI cost-effectively and privately. This creates the opportunity to reimagine the prevailing paradigm in generative AI today, which is a centralized "frontier" large language model (LLM) that operates on a fleet of expensive GPUs to a new era of generative AI which is decentralized, flexible, and lowers the center of gravity of AI deployments to mainstream commodity servers and directly on endpoint users' machines. This paradigm shift radically reduces run-rate inferencing costs, accelerates the ability to fine-tune models and workflows, and ensures data privacy and wide-spread, secure end-user enterprise access.

This next era of decentralized generative AI is enabled primarily by two major trends that have accelerated over the last 12 months that look poised to continue over the next year:

1- A paradigm of "90% as effective" at 1% of the cost, combined with top-off fine-tuning to bridge the gap of the last 10%, is emerging as a common practical strategy.

SLMs have experienced a much more rapid improvement curve over the last 18 months than frontier models, with SLMs being able to incorporate most of the novel capabilities of larger LLMs through a variety of improved instruct datasets, teacher-student paradigms, model pruning, quantization and leveraging similar architectural and training technique improvements used in larger models. In most tasks, SLMs can perform in substantially similar ways to larger models, with the added benefit of both training and inferencing costs that are often 10-100X cheaper while being able to also rapidly (and cost effectively) fine-tuned to adapt to a specific purpose.

2- Inferencing will become a common everyday computing task.

Edge-ready GPUs and NPUs are coming and will be widely available over the next 12 months. The AI PC is coming to enterprises in 2025, and features integrated GPU, NPU and other "AI accelerators", as developed by chipsets from Intel, AMD and Qualcomm. The AI PC will unlock new possibilities for routinely deploying SLMs in the range of 1-10 billion parameters, and for some uses, models as large as 15-20 billion parameters can be quantized and run effectively on a Windows laptop. By pulling model inferencing to the edge, it will also create a significant commodity server 'offload' opportunity. Departments and users will be able to rely on small inference servers to run batch background inferencing processes safely and securely, working in a seamless 'hybrid' mode with edge devices, much like the early days of PCs on every desktop fueled the evolution of 'client-server' pulling processing away from centralized mainframes into clusters of processing around both the edge and localized servers.

Benefits of Decentralized AI

High-quality smaller models, deployed on the edge, will be transformational in removing the most common roadblocks for generative AI pilots to move into production:

Data Privacy – the entire data pipeline can live within the enterprise security zone, greatly reducing concerns of data leakage, and for highly-sensitive processes, entire generative AI workflows can be executed in an "air-gapped" environment;

Security and Control – breaking open the "black box" of generative AI – all of the code, model parameters and metadata will be accessible to enterprise security, safety and control teams to monitor, evaluate and continuously improve. In addition, smaller models generally present a smaller attack surface for security breaches, and are less vulnerable to malicious attacks;

Adaptation – flexible, hybrid, decentralized deployments can be adapted more easily, both the data pipeline and workflow, as well as fine-tuning (and changing) the underlying models; and

Sustainability – projects can be delivered with predict-

able, lower costs, much like any other software or technology project.

But .... It's not that easy!

As these trendlines emerge for enterprise decentralized deployment, however, there are a number of new problems to be solved. Namely, it is not that simple to put together an integrated generative AI software stack that can be easily deployed at scale, accessible to end users, customizable for developers and manageable for enterprise IT.

The generative AI software stack that has emerged over the last two years through a series of open-source projects is often optimized for complex server environments, API-based models, and requires installing dozens of discrete dependencies and integrations among open-source libraries. Common elements such as python environments, pip installs, bash scripts, CLI interfaces, and Docker containers which are the glue that tie together most open-source deployments (combined with a healthy willingness to "tinker" around the edges) are simply not consistent with deployment in most enterprise edge environments.

Introducing LLMWare's Model HQ: A Comprehensive Platform for AI Deployment

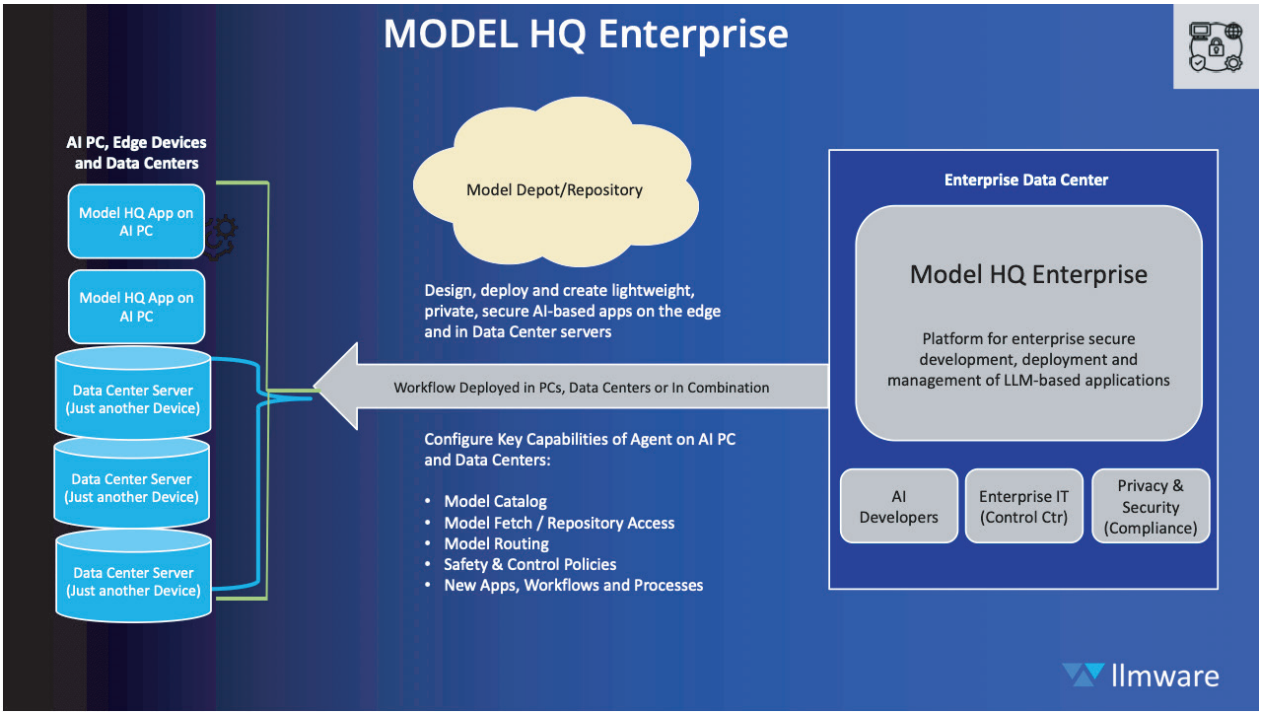
Model HQ is the first-in-kind comprehensive platform for AI deployment in Private Cloud, Data Centers and User PCs, and is designed to manage the entire lifecycle of lightweight, private LLM-based applications safely and efficiently. It gives enterprises full control over deploying AI workflows directly on any deployment model, including user PCs, by offering the easiest and most automated way to leverage the best AI framework and model for their hardware. Specifically, Model HQ is designed to run out-of-the-box for Intel-based AI PCs with Intel Core Ultra Processors (Series 1 and 2), providing no code, point and click solutions for information retrieval such as Retrieval Augmented Generation, chatbot, natural language SQL queries and table-extraction on device.

AI developers and IT teams can quickly deploy a variety of pre-built AI workflows or use the Model HQ platform for easy, low to no-code development, utilizing over 100 models from LLMWare's Model Depot, including a large collection of the most popular and leading-edge small language models in the OpenVINO and ONNX formats optimized for Intel Core Ultra Processors.

With Model HQ, enterprise AI developers can seamlessly update and deploy AI workflows to users while benefitting from integrated safety and security features. These include safeguards that detect compromised models and checks for prompt injections, toxicity, bias, and hallucinations. Additionally, the platform features a centralized Compliance Station that offers on-demand safety and data configuration settings, AI Explainability Tracing, Data Privacy Guard with PII filtering, and a comprehensive Audit Log with automated reporting.

The counter-part receiver of Model HQ is the Client Agent for Users. Client Agent is an executable software package that has been carefully designed to provide a compelling end-user experience with "point and click" simplicity. The Client Agent delivers a powerful set of "packaged" underlying capabilities that can be configured, controlled and extended by enterprise IT and AI development teams. The Client Agent also provides the on-device "run time" platform that can execute a wide range of AI-based processes that allows developers to 'roll out' new applications and workflows which can then be installed to the Client Agent.

In addition to workflow, Model HQ is designed to enable enterprise IT engineers to manage a fleet of "distributed" inference servers that now exist among end-users as well as the ability to rapidly roll-out new policies and procedures across all of their AI processes. The Client Agent can be managed remotely and dynamically through configuration updates, including model catalog, model routing, model access, a wide range of inference preview and postview filters, and capturing of an inference history that can be analyzed and reviewed for costing, compliance, auditing, performance, and ongoing improvements.



Auto AI-Optimization: Streamlining AI Framework Deployment

A key feature of the Model HQ platform is its Auto AI-Optimization technology, which automatically identifies and deploys the optimal AI framework for the user's specific hardware environment. Model HQ simplifies the use of AI frameworks by abstracting the complexities involved in implementing them, allowing users to leverage the most suitable and optimized model type for their tasks. For instance, on laptops equipped with Intel Core Ultra Processors with integrated GPUs (iGPUs), the Client Agent automatically detects and utilizes the iGPU to enhance AI workflows with an optimized model when available. This automated approach not only optimizes performance but also reduces the complexity and time required for users to achieve the best AI model execution on their devices, enhancing productivity and enabling seamless integration of AI capabilities across diverse hardware environments.

By supporting the most popular AI frameworks - PyTorch, GGUF, ONNX, and OpenVINO – Model HQ enables seamless execution of most models on compatible hardware without requiring additional platform configurations. Recognizing the significant performance variations between model technologies on different platforms, the ability to "mix and match" inferencing technologies without the complexities of dependency management or rigid workflow integrations is crucial. This approach ensures superior execution speeds and resource efficiency.

Private, Self-Managed Model Catalog

Further enhancing its value, Model HQ supports the creation of a fully private, self-hosted, and self-managed model catalog within the secure perimeter of the enterprise. This catalog encompasses every stage of the model lifecycle, including secure storage, ensuring that all model-related activities remain within the organization's private security domain.

This capability empowers developers to download models from open-source repositories, fine-tune them, and then securely store them in a private repository. By doing so, enterprises can completely sever ties with open-source repositories and external cloud training services, enabling models to be fully managed and utilized within the enterprise's own secure environment. This not only enhances security but also ensures complete control over the AI model lifecycle, from development to deployment.

Model HQ offers a wide range of built-in interfaces that simplify the integration of LLM-based applications into user interfaces and other software with low to no code. It includes a complete set of REST APIs and allows users to easily switch between local execution and API endpoints. This flexibility also enables exposing AI functionalities on an AI PC as REST endpoints, making it straightforward to deploy and test local applications. This setup supports hybrid solutions where some parts utilize the local GPU

capabilities of AI PCs, while others leverage models and services distributed across other AI PCs, private datacenters, or even public clouds. Additionally, Model HQ includes integrated UI development tools, enabling rapid creation and deployment of UI-based applications without extensive coding.

Compliance Station: Ensuring AI Safety, Privacy, and Accountability

Model HQ's Compliance Station is a comprehensive feature designed to address the critical requirements of data privacy, safety, and compliance in enterprise AI deployments. This integrated suite provides robust tools and configurations to ensure that AI workflows operate within stringent regulatory and organizational standards.

Model HQ also provides built-in filters for key PII indicators (e.g., Social Security numbers, emails, credit card numbers), as well as classifiers for detecting prompt injections, toxic content and bias. Multiple output and storage options are available to facilitate analysis and review of all captured metadata, supporting comprehensive compliance and performance analytics for compliance and audit records. By incorporating advanced compliance and safety features, Model HQ empowers enterprises to deploy AI with confidence, ensuring that all AI-driven processes are secure, transparent, and compliant with both legal and organizational standards.

Conclusion

By integrating edge device capability with platforms like Model HQ, enterprises can further streamline AI implementation, benefitting from a comprehensive end-to-end solution that supports the full lifecycle of AI workflow—from development and deployment to monitoring and compliance. The combination of high-performance hardware, optimized AI frameworks, and robust management tools provides businesses with the agility and control needed to innovate and scale AI-driven processes across diverse environments.

In conclusion, the recent advancements in SLMs and AI PCs are not just incremental improvements but a transformative leap that enables a decentralized, efficient, and secure AI ecosystem. This new paradigm empowers enterprises to unlock the full potential of generative AI, driving productivity gains and innovation directly from its private cloud, data center or the user's PC, thereby paving the way for the next generation of AI-powered business solutions.

ABOUT LLMWare.ai

LLMWare.ai provides an end-to-end solution for running, deploying and creating AI-based applications using Small Language Models for the enterprise. Selected by Github as a leading open-source technology shaping the future of AI in 2024, LLMWare is a pioneer in deploying and fine-tuning Small Language Models particularly for use in highly-regulated or data sensitive industries and a leader in cutting-edge AI app deployment platforms. For additional information, including product, blogs and latest research reports, please visit llmware.ai

"LLMWare Announces Collaboration with Intel for Deploying Gen AI on AI PCs," October 15, 2024, https://finance.yahoo.com/news/llmware-announces-collaboration-intel-deploying-130000313.html

Small Language Model

llmware

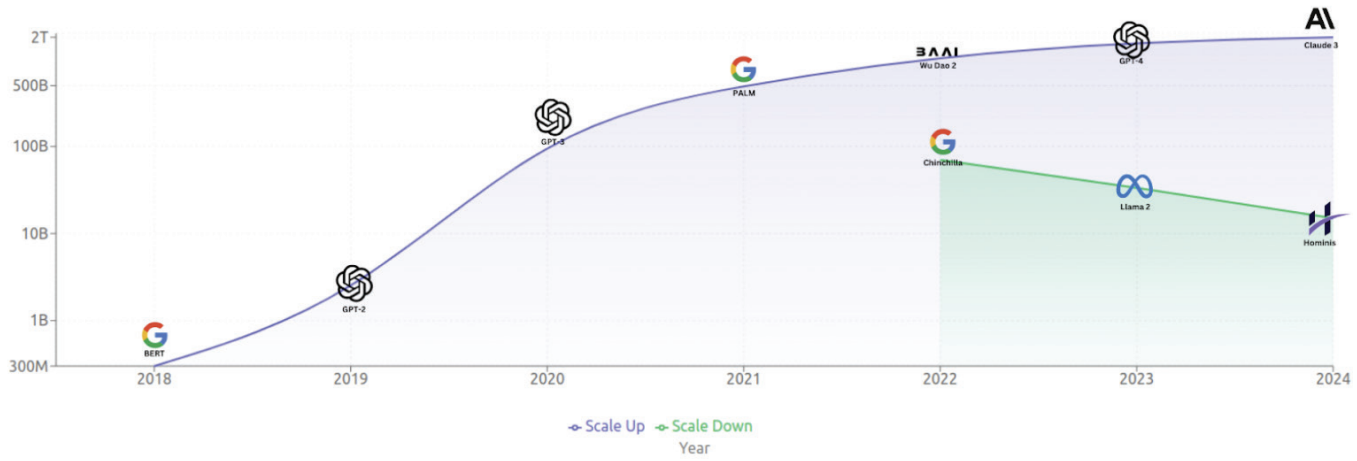
Decentralized AI at your Fingertips

SMALL LANGUAGE MODELS MADE EASY

Hominis:

The Power of Lean AI Models without Compromising Performance

The trajectory of artificial intelligence (AI) development has been largely driven by the notion that "bigger is better." Since the introduction of Transformers in 2017, and the launch of large language models (LLMs) like BERT in 2018, the trend has been to scale models to ever-larger sizes—often reaching trillions of parameters—in pursuit of improved accuracy and capability. However, recent advancements are challenging this paradigm, emphasizing that performance and efficiency need not come at the cost of massive scale. In 2022, the Chinchilla paper marked a pivotal shift, highlighting the importance of optimizing models not just by increasing their size, but through better data quality and more efficient training. This recalibration continued with models like Llama, and today, it is exemplified by Small Language Models (SLMs) such as Hominis.



Hominis, designed for enterprise AI, leverages these principles of computational efficiency and sustainability to deliver high-performance solutions that meet real-world needs—without the significant computational overhead associated with larger counterparts.

A Data-Driven Approach to Model Development

The development of Hominis was guided by a detailed scientific process, beginning with a large-scale analysis of commercial usage patterns for language models. One of the key insights from this research was that most enterprises deploy models under 3 billion parameters, largely due to hardware and resource constraints. However, based on our evaluations, a 3-billion-parameter model often lacks the capacity to handle more complex, domain-specific tasks without significant trade-offs in performance when trained from scratch. This led us to the strategic decision to develop a 15-billion-parameter model, which could later in turn be distilled down.

One of the key factors that influenced our decision to scale to 15 billion parameters was the advancement of quantization techniques, particularly Q4 quantization. This method reduces the precision of the model to 4-bit integers, leading to a 4x reduction in memory and computational demands while preserving nearly all of the performance of the full bf16 (16-bit) model during inference. As a result, models from the Hominis family can be loaded and run on consumer-grade GPUs with more than 4.5 GB of RAM. Depending on the context and the length of the required generation, this allows Hominis to perform inference on standard consumer hardware without the need for any specialized setup.

1 Patwardhan, N.; Marrone, S.; Sansone, C. Transformers in the Real World: A Survey on NLP Applications. Information 2023, 14, 242. https://doi.org/10.3390/info14040242

Ablation Studies and Architectural Choices

A crucial aspect of the development process for Hominis involved extensive ablation studies to evaluate and refine our transformer architecture. We systematically tested various attention mechanisms, focusing on balancing performance, accuracy, and computational efficiency. After comparing several transformer variants—including Reformer, Informer, Longformer, and synthetic attention mechanisms—we found that matrix approximation methods did not offer sufficient performance gains in real-world tasks. Instead, Flash Attention, which leverages hardware-aware acceleration techniques, proved to be far more effective. It maximized the utilization of modern hardware, achieving efficient, high-speed attention calculations without compromising accuracy. As a result, Flash Attention was integrated into the core Hominis architecture.

Additionally, we chose to adopt a decoder-only architecture for Hominis due to its simplicity and ease of training, particularly when handling text as the primary modality. Text, with its variable structure and high semantic complexity, does not naturally lend itself to fixed-length encoding when trained in an autoregressive manner, making a decoder-only approach more suitable for capturing the richness of the input. This architecture choice allows Hominis to maintain flexibility in handling variable-length sequences across a wide range of tasks.

However, for other modalities such as images or videos, which typically have fixed maximum dimensions, we plan to retrofit an encoder on top of the Hominis model for more efficient processing. These modalities are more structured, and their fixed dimensions can benefit from an encoder-decoder architecture to compress and manage data effectively. This will allow Hominis to extend its versatility beyond text, adapting to various domains while maintaining its core efficiency.

Further experiments also led to important insights regarding layer types. Mixture of Experts (MoE) layers were explored for their potential to enhance the model's adaptability. However, they significantly slowed down training compared to traditional linear layers. Despite this, MoE layers have demonstrated strong potential in specialized, fine-tuned applications. Consequently, we opted to use linear layers during the base training of Hominis to optimize training speed and computational efficiency. MoE layers can then be applied post-training during fine-tuning to support domain-specific tasks, allowing Hominis to maintain flexibility and adaptability for enterprise applications without incurring high upfront training costs.

Novel Training Methodology

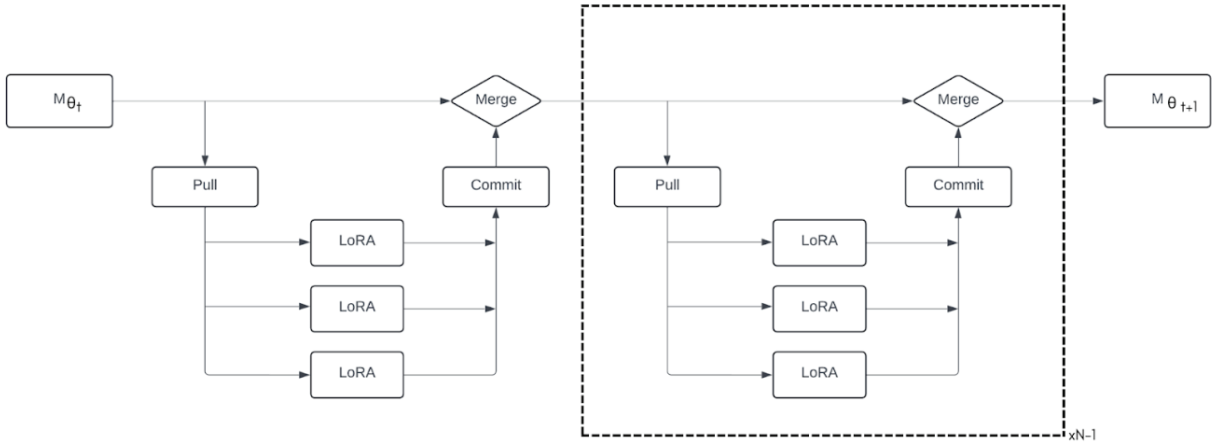
In our training of the Hominis models, we adopted a variant of LoRA the explorer approach to reduce computational costs and enhance efficiency that draws inspiration from Low-Rank Adaptation (LoRA), a technique traditionally used in fine-tuning. Our approach, extended the application of LoRA principles to the training phase itself, making it a key component in optimizing performance and resource use during model pre-training.

Specifically, we deployed a per-GPU LoRA matrix that received updates exclusively from data partitioned to that specific GPU. Periodically, these individual LoRA matrices were merged together and then added to a primary weight matrix, shared across all GPUs, enabling global synchronization every few minibatches. This design helped reduce the memory and computational burden, preserving model performance while minimizing the training costs.

By integrating this LoRA-based strategy into the core training process, we achieved a more resource-efficient training paradigm. This advancement allows Hominis to maintain competitive accuracy and scalability when compared to significantly larger models, all while keeping both computational expenses and energy consumption at lower levels.

To further enhance the efficiency of the training process, we employed MuTransfer, a method designed to minimize the need for costly hyperparameter tuning across different tasks and datasets. This technique allowed us to transfer learned hyperparameters from one training scenario to another, significantly reducing the trial-and-error process that typically consumes substantial computational resources.

2Huh, Minyoung, et al. "Training neural networks from scratch with parallel low-rank adapters." arXiv preprint arXiv:2402.16828 (2024).



Benchmarking Hominis: Real-World Performance

We conducted benchmarking against LLaMA2, recognizing that while both Hominis and LLaMA2 are foundational models, future iterations such as LLaMA3 incorporate not only raw pretraining but also extensive instruction tuning. These comparisons provide a clearer understanding of the Hominis models' raw performance, offering valuable insights for future fine-tuning and adaptation to specialized tasks.

Our evaluation focused on three iterations of Hominis Core 15B across a diverse set of tasks, including code generation, mathematical reasoning, and world knowledge. In the domain of code generation, Hominis Core v1504 outperformed LLaMA2 on the MBPP dataset with a score of 36.5, compared to LLaMA2's 30.6. The Hominis model also demonstrated superior mathematical reasoning, achieving a score of 13.5 on the MATH dataset, significantly higher than LLaMA2's 3.9. In tasks assessing world knowledge, such as TriviaQA and NaturalQuestions, Hominis Core 13B's performance closely aligned with LLaMA2.

While these results highlight the raw capabilities of the Core model, we are currently benchmarking the Hominis Instruct model, designed specifically for enhanced instruction-following. Early results from user preference testing, particularly in comparison to closed-source models like GPT-4 and Claude, show promising improvements. The Instruct model demonstrates greater accuracy in task-specific queries and higher adaptability to

nuanced instructions, which could prove particularly beneficial in enterprise settings such as automation and customer service.

Energy efficiency has been a key consideration in our benchmarking process. Hominis consumed 23,961.6 kWh during training, resulting in a carbon footprint of just 11.38 metric tons of CO2, making it one of the most eco-friendly models in its class. Additionally, the model's ability to operate on consumer-grade GPUs enhances its accessibility for businesses without large-scale data center capabilities. This combination of energy efficiency and scalability positions Hominis as a practical solution for organizations seeking sustainable AI deployment.

Benchmark Category	Dataset	LLaMA2 13B Score	Hominis Core 13B v0504	Hominis Core 13B v1004	Hominis Core 13B v1504
Code Evaluation	MBPP	30.6	20.8	33.0	36.5
	HumanEval	18.3	12.4	20.8	20.2
Math Evaluation	GSM8K	28.7	20.4	28.9	28.7
	MATH	3.9	4.5	11.2	13.5
World Knowledge	NaturalQuestions (0-shot)	16.1	16.1	16.3	16.4
	TriviaQA (0-shot)	73.1	65.8	67.2	73.7

Looking Ahead: The Evolution of Hominis

The future of Hominis focuses on expanding accessibility and optimizing efficiency across a variety of use cases. Building upon the Core and Instruct 15-billion-parameter model, our ongoing efforts include the development of distilled variants, such as 7-billion and 1-billion-parameter versions. These smaller models are designed to bring advanced AI capabilities to enterprises with more limited computational resources, without sacrificing the high levels of performance and efficiency established by the larger model.

Our research and development efforts also extend to regional and culturally specific models, such as Hominis Italia and Hominis Dutch, which are being developed to meet the unique linguistic and cultural needs of local markets. These models will allow enterprises to deploy AI that understands and responds to the specific contexts of their target regions, enhancing the relevance and usability of the technology.

Additionally, we are continually refining the Hominis CI platform, which offers seamless fine-tuning and deployment of models, making customization more intuitive and less resource-intensive. This platform is designed to allow enterprises to easily adapt Hominis to their specific needs, ensuring that AI solutions can be both powerful and cost-effective without requiring extensive technical expertise or infrastructure.

Conclusion

The development of Hominis embodies a rigorous, scientifically-driven approach to creating AI models that balance power, efficiency, and accessibility. Through careful architectural choices, innovative training methodologies, and the adoption of quantization and advanced optimization techniques, Hominis demonstrates that smaller models can deliver exceptional value in real-world enterprise applications without the burdensome costs of larger systems.

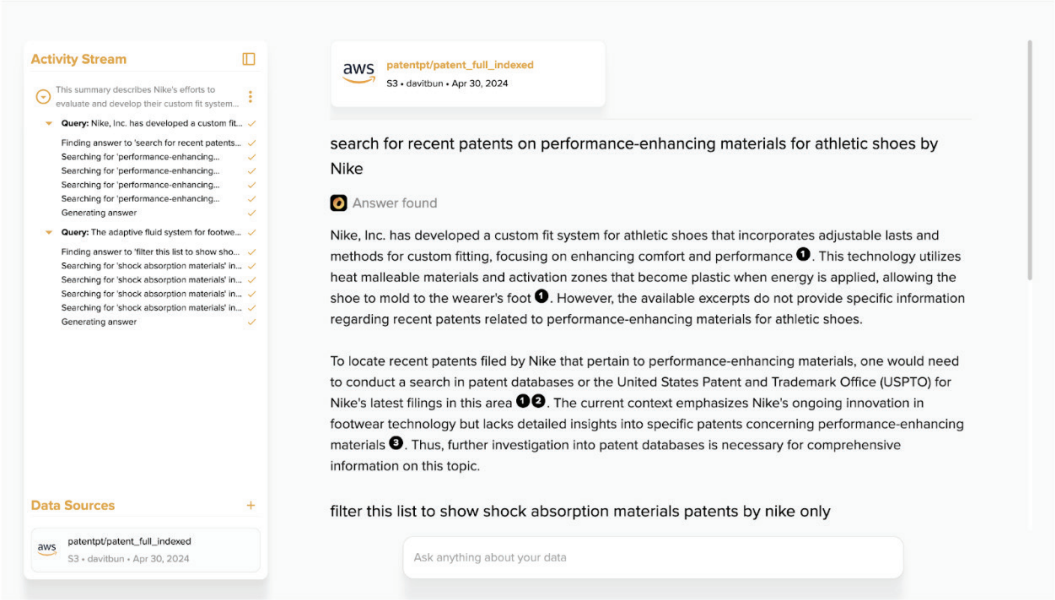
Hominis is currently under closed beta.

Narendra Patwardhan

Narendra Patwardhan is the CTO of deepkapha AI labs and a Ph.D. student at the University of Naples. With a master's from Michigan Technological University, he previously served as Head of AI at various startups, leading several DARPA projects. Narendra's expertise lies in optimizing neural network-based algorithms for edge devices, bridging sophisticated AI with resource-constrained environments. His work combines academic research with practical industry applications, driving innovation in AI and edge computing.

How to Use RAG & Small Language Models for a Legal AI Search Engine for Patents

As enterprises look to leverage the power of LLMs on their data, they need to enable AI search over vast, multi-modal information sources, both internal and external, spanning public and private data. Building a robust AI search engine is challenging, especially considering the cost of creating an internal Google-level system. Instead, a more feasible, future-proof approach is to build an AI search system powered by a data retrieval engine, complemented by custom fine-tuned small language models (SLMs) and a multi-agent system to efficiently manage search and retrieval tasks. This is what it looks like:

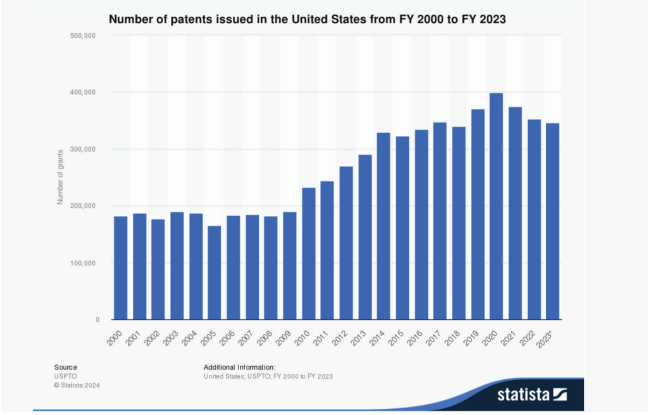


For enterprises, making search both accurate and affordable is essential. Leveraging object storage allows organizations to reduce costs while maintaining scalability and efficiency, making AI-powered search more accessible for enterprises dealing with vast datasets.

In this article, we showcase 'PatentPT,' a legal AI use case for defensible patent generation and efficient AI search, powered by Deep Lake and a custom-built Small Language Model (SLM). But first, why is patent search and creation a problem?

Why is Patent Search and Creation a Problem?

The United States Patent and Trademark Office (USPTO) website, which was last updated in the early 2000s, was once a technological marvel but has since become extremely slow and outdated. Given the complexity of the patent landscape, where in the US alone 594,340 patents were filed in 2023 (Statista) and around 80 million patents exist in total, navigating this vast corpus effectively requires advanced tools. At least 95% of these patents receive a non-final rejection (BigPatent-Data/USPTO), and it takes 2-4 weeks on average for an experienced attorney to finalize a single patent draft.



AI Search Engine for Patent Search and Patent Generation

To streamline the cumbersome process of searching for related patents and creating a unique, defensible one, we developed 'PatentPT'—an advanced SLM-based Retrieval Augmented Generation (RAG) tool for patent search and generation—alongside Deep Lake by Activeloop, a multi-modal database for AI that enables fast and accurate AI search on object storage.

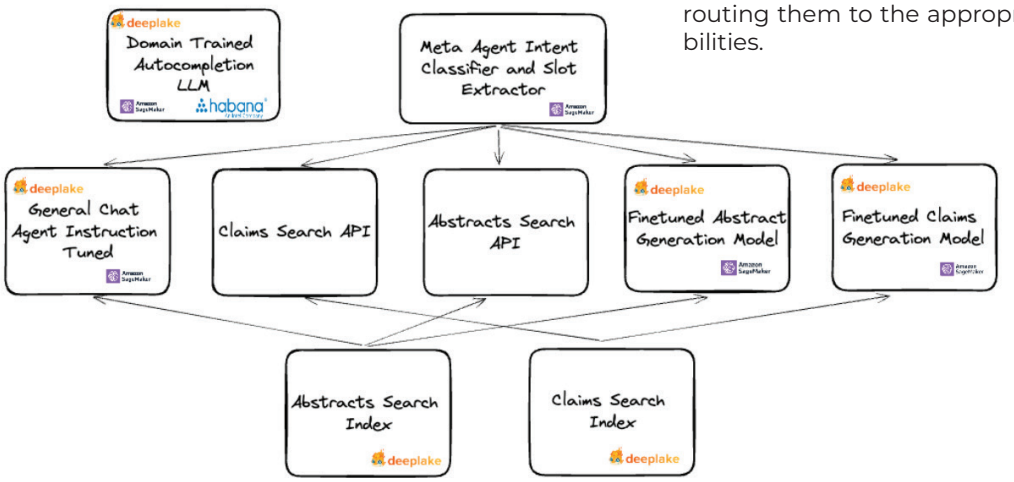
Introducing PatentPT: Architecture Blueprint for AI Patent Search and Generation

PatentPT is a custom-built patent search and retrieval application designed to address the shortcomings of legacy systems like the USPTO, which have rigid search capabilities and outdated technologies. With a custom SLM-enabled approach, PatentPT delivers a seamless experience for patent question answering, text generation, and retrieval.

PatentPT Key Features:

- Autocomplete
- Patent Search on Abstracts and Claims
- Abstract and Claim Generation
- General Question-Answering Chat

PatentPT's user-friendly interface integrates all these features, ensuring a smooth and efficient patent research process. A meta-agent manages user queries, routing them to the appropriate LLM models and capabilities.



Technical Journey Behind PatentPT

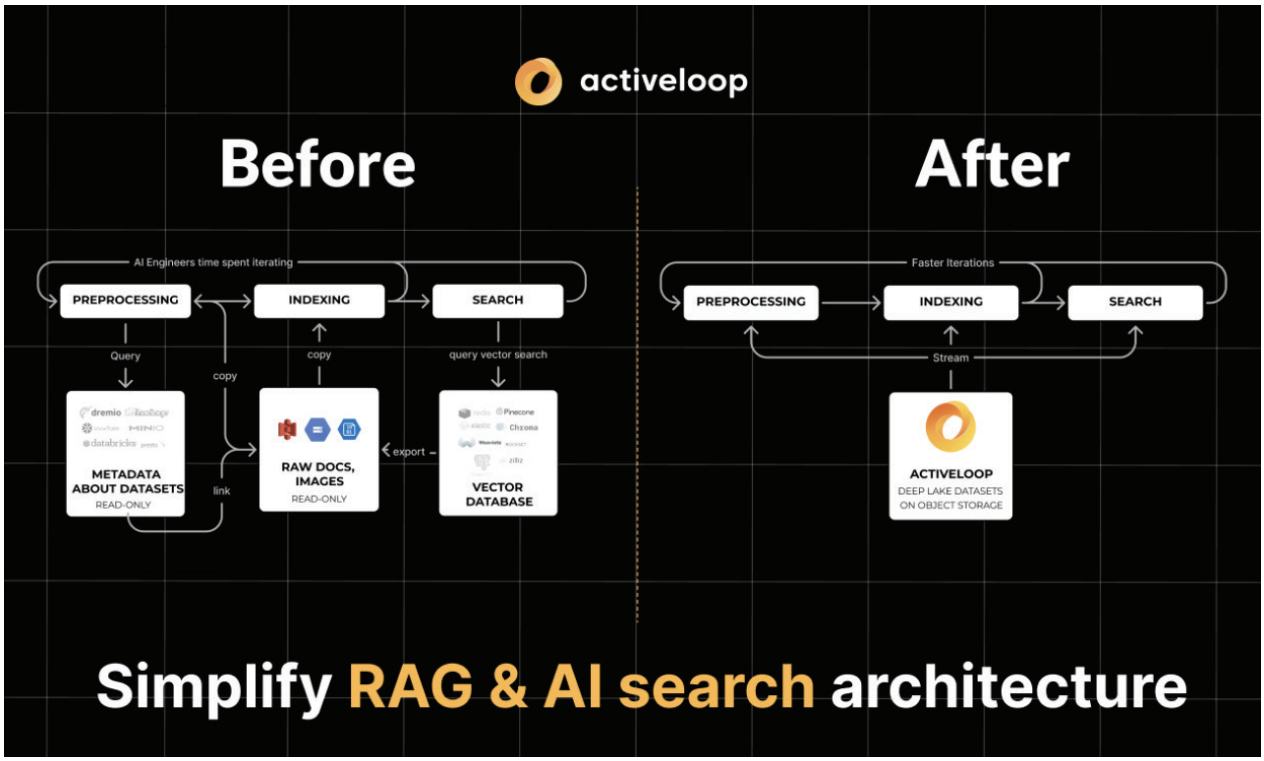
The technical backbone of PatentPT involved creating an ensemble of fine-tuned LLMs and search indices to maximize accuracy and performance. We began by domain training a base LLM on the entire corpus of USPTO patents, comprising 8 million patents and over 40 billion words. This domain-specific pre-training provided the foundation for various APIs, such as autocompletion and custom patent search features.

Domain Training the Small Language Model for PatentPT

The data preparation stage also involved leveraging Activeloop Deep Lake's efficient dataloader. Unlike conventional dataloaders, Deep Lake's streaming dataloader ensures that data is transferred seamlessly from remote storage to compute devices during the model training process, eliminating bottlenecks caused by disk I/O operations. Deep Lake's dataloader integrates with deep learning frameworks like PyTorch and TensorFlow, allowing more efficient training.

Benchmark comparisons show that Deep Lake's dataloader outperforms others, such as Petastorm and PyTorch's default dataloader, in iteration speed and training time. This efficiency was particularly crucial for handling the large patent corpus during both domain training and fine-tuning phases.

The data preparation stage also involved leveraging Activeloop Deep Lake's efficient dataloader. Unlike

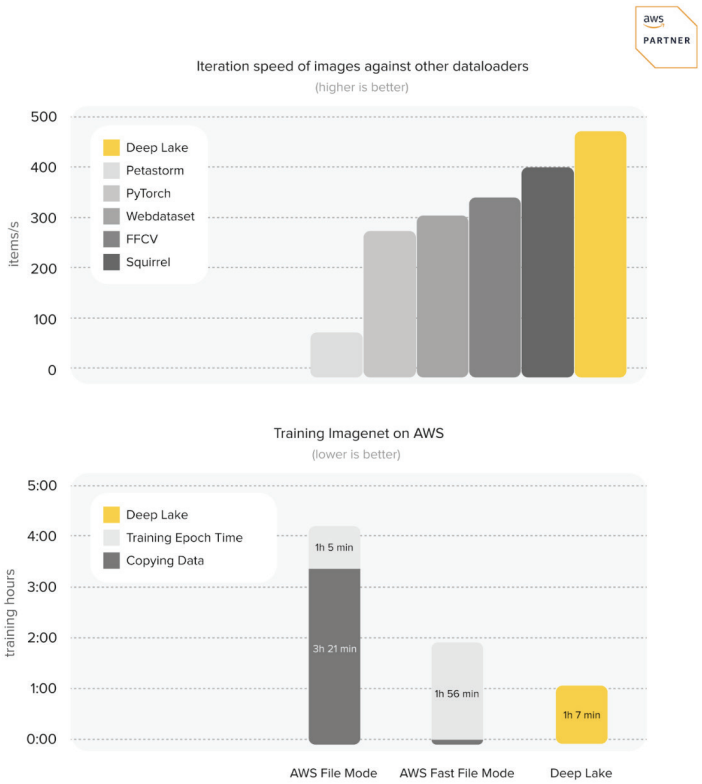


conventional dataloaders, Deep Lake's streaming dataloader ensures that data is transferred seamlessly from remote storage to compute devices during the model training process, eliminating bottlenecks caused by disk I/O operations. Deep Lake's dataloader integrates with deep learning frameworks like PyTorch and TensorFlow, allowing more efficient training.

Benchmark comparisons show that Deep Lake's dataloader outperforms others, such as Petastorm and PyTorch's default dataloader, in iteration speed and training time. This efficiency was particularly crucial for handling the large patent corpus during both domain training and fine-tuning phases.

Using Deep Lake's dataloader reduced training time for the Small Language Model, allowing more focus on optimization and fine-tuning. Training tasks that previously took over five hours were completed in just over an hour, significantly improving productivity and reducing costs.

To train the base patent SLM, we used Intel's Habana Gaudi HPUs for their efficiency in handling transformer models. The process involved tokenizing the entire dataset (which took 18 hours) and training for 24 days to achieve optimal validation loss convergence. The domain-trained SLM served as the basis for the autocompletion API and was further fine-tuned for other tasks such as abstract and claim generation.



Fine-tuning Small Language Models for Patent Generation

We fine-tuned additional LLMs for generating abstracts and claims by using specific datasets derived from patent descriptions. Leveraging the Deep Lake dataloader and Hugging Face PEFT techniques, we used LoRA weights for targeted fine-tuning without altering the core model weights. This was possible thanks to Deep Lake's native integration with TorchTune by PyTorch.

Setting Up Search Indices with Deep Lake

To support search capabilities, we created search indices based on patent abstracts and claims using a custom featurizer derived from our domain-trained LLM. Deep Lake offers embedding, lexical, and inverted indices to optimize retrieval performance and improve efficiency.

We used the database for AI for indexing, which offered enhanced retrieval accuracy by up to 22.5% on average, and is up to 10x more cost-efficient than in-memory databases thanks to its index-on-the-lake technology, which enables efficient querying directly from object storage.

RAG and SLMs for Cost-Efficient Patent AI Search with Deep Lake 4.0

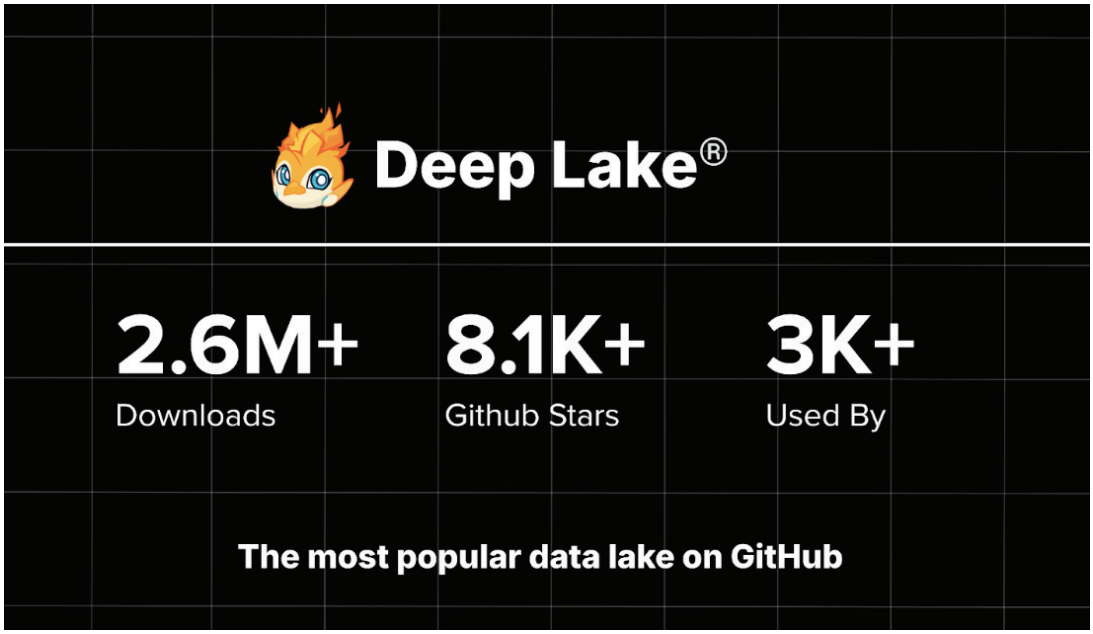
Deep Lake 4.0's index-on-the-lake technology enables scalable AI retrieval by storing indices directly on object storage like Amazon S3, making it a cost-effective solution for massive datasets without in-memory storage costs.

Key Benefits of Index-on-the-Lake Technology by Deep Lake

Cost Efficiency: By offloading indices to object storage, Deep Lake 4.0 reduces the need for expensive compute resources, resulting in up to 10x cost savings compared to traditional databases. It also delivers sub-second query times, making it suitable for real-time applications.

Scalability: Storing the index on object storage allows Deep Lake to scale effortlessly with growing data volumes. There is no need to worry about running out of memory or upgrading hardware to accommodate larger datasets. Deep Lake 4.0 ensures that your retrieval system can grow alongside your data without compromising on performance.

Multi-Modal Capabilities: Deep Lake 4.0 supports multiple data modalities, including embeddings, text, images, and 3D data, making it ideal for use cases that require rich contextual understanding. By leveraging multi-modal data, Deep Lake enhances the accuracy of search results and enables more sophisticated AI applications.



How to Build Your Own AI Search Engine with a Custom Small Language Model

Creating your own AI-powered search engine may seem like a daunting task, but recent advancements in LLMs and databases for AI like Deep Lake have made it more accessible than ever. By using a custom small language model, you can build a highly efficient AI search engine tailored to your specific needs. Below, we outline the steps required to build such a system, inspired by the development of PatentPT and Deep Lake 4.0.

Step 1: Define Your Use Case and Data Requirements

The first step in building an AI search engine is to define the domain and scope of your search engine. Whether you're working on patents, legal documents, or scientific research, identifying the types of
data and the specific queries users will perform is crucial. Collect a high-quality dataset that captures the entire spectrum of information relevant to your use case.

Step 2: Preprocess and Tokenize Your Dataset

Once you have your dataset, preprocessing is essential. Convert your documents into a machine-readable format, clean any inconsistencies, and structure the data into meaningful fields. Use a tokenizer, such as those provided by Hugging Face, to prepare your dataset for training. Tokenization is a computationally intensive step, but it sets the foundation for the model's training process.

Step 3: Domain Training Your Small Language Model

For your AI search engine, it's important to start with domain training a small language model. You can use open-source models like GPT-2 or DistilBERT as a base and fine-tune them on your domain-specific dataset. This helps the model learn the terminology and nuances of your specific field. Training can be done on specialized hardware, such as GPUs or HPUs, using libraries like Optimum for efficient model training.

Step 4: Fine-tune for Specific Search and Generation Tasks

Once you have a domain-trained model, the next step is to fine-tune it for specific tasks like search, autocomplete, and content generation. Use PEFT (Parameter-Efficient Fine-Tuning) techniques to optimize your model without requiring vast computational resources. Fine-tune separate models for different tasks, such as answering questions, generating summaries, or completing text.

Step 5: Create Custom Featurizers for Search Indexing

To enhance search accuracy, extract custom features from your domain-trained model. Use the last hidden layer representations to create feature vectors that are more accurate for semantic search compared to traditional embeddings. These custom featurizers can be used to index your dataset, making retrieval more efficient and context-aware.

Step 6: Set Up a Database for Indexing

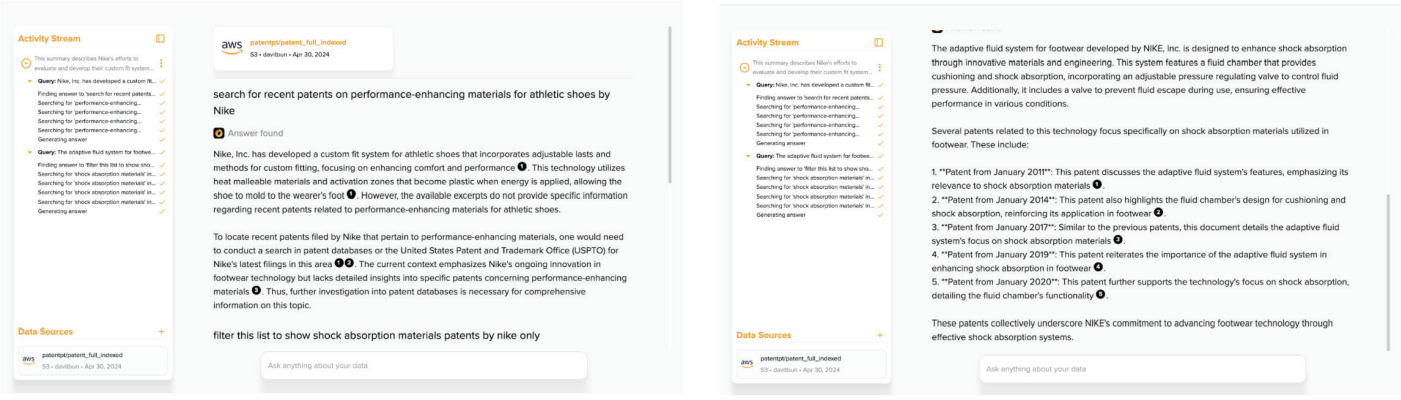
To store and retrieve documents efficiently, set up a database for AI. Activeloop Deep Lake is a database for AI that allows for cost-efficient and highly accurate retrieval for LLMs, providing cost-efficient storage and sub-second query times. Deep Lake provides various types of indices, including embedding, lexical, and inverted indices, which cater to different retrieval needs and optimize search accuracy and efficiency.

Step 7: Deploy Your Search Engine APIs

Develop APIs to interact with your AI search engine. These APIs will handle user queries, call the appropriate model or search index, and return results. You can use frameworks like Flask or FastAPI to create and deploy these endpoints. Deploy your language models on cloud platforms like AWS or Azure to ensure scalability and availability.

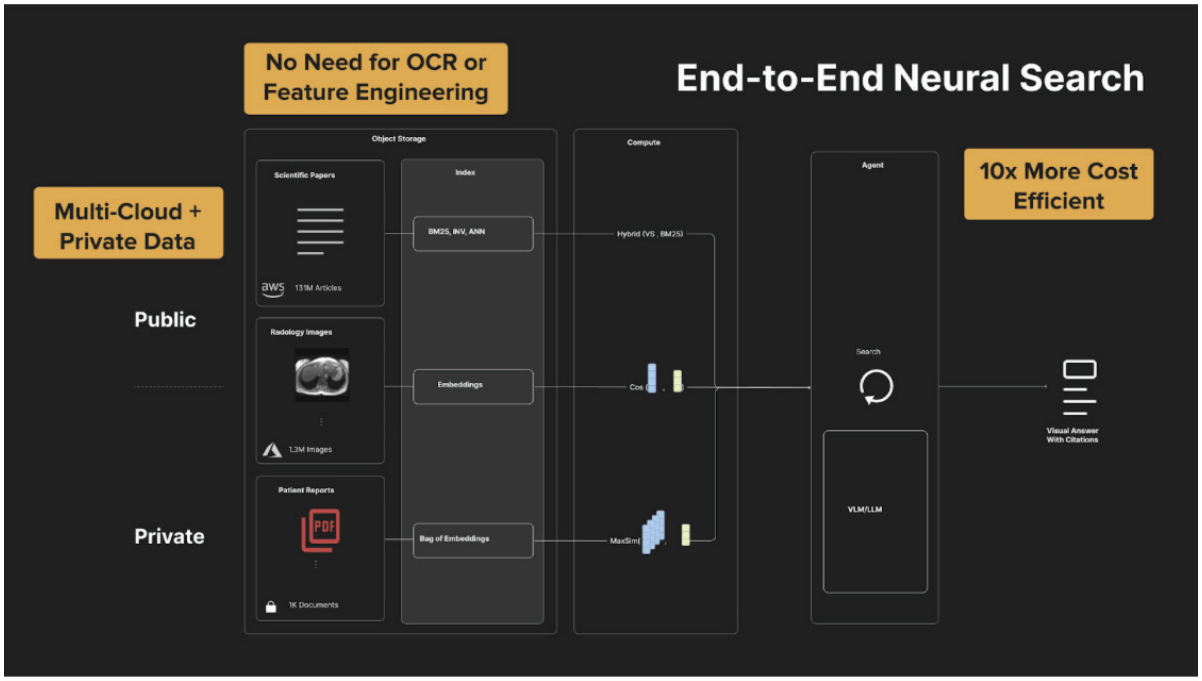
Step 8: Optimize for Real-Time Inference

Real-time inference is crucial for a responsive search engine. Use cloud-based deployment services, such as Amazon SageMaker, to host your fine-tuned models and APIs. Consider using containerized environments for easy scaling. With a small language model, you can achieve low-latency responses suitable for a real-time search experience.



Deep Lake 4.0: Revolutionizing Multi-Modal AI Retrieval

The technical evolution of PatentPT naturally led us to explore innovative approaches in AI retrieval. Traditional AI data retrieval systems face three significant challenges: limited modalities, lack of accuracy, and high costs at scale. Deep Lake 4.0 addresses these challenges by introducing true multi-modality, enhancing accuracy, and reducing costs using its unique index-on-the-lake technology.



Challenges in AI Retrieval

Deep Lake has been a game-changer in overcoming challenges like:

Lack of True Multi-Modality: Activeloop's collaborations with Matterport and Bayer revealed that using raw data enriched with metadata across different modalities—such as text, images, and 3D scans—delivers superior results compared to conventional methods that only store vectorized data.

Inaccuracy at Scale: In domains like healthcare and legal, achieving accurate retrieval at scale is paramount. Deep Lake's multi-modal indexing and filtering capabilities provide the precision needed for these critical sectors.

High Costs of Building Advanced RAG Systems: By utilizing Deep Lake 4.0, organizations can build cost-effective Retrieval-Augmented Generation (RAG) systems without the billion-dollar budgets typically required for Google-level search experiences.

Limited Memory and Scalability: Deep Lake's architecture, designed around object storage, eliminates the need for costly in-memory databases, making it more suitable for growing datasets without compromising on speed or performance.

Redefining AI Retrieval with Activeloop Deep Lake 4.0

The newest version of Deep Lake, Deep Lake 4.0 offers:

Fast and Accurate Retrieval: Combining multiple indexing techniques—such as embedding, lexical, and inverted indexes—Deep Lake ensures rapid search responses even for large datasets.

Cost Efficiency: The separation of compute from storage and the offloading of indexes to object storage results in up to 10x cost savings compared to traditional AI retrieval systems.

Enhanced Multi-Modal Capabilities: The inherent support for n-dimensional arrays and seamless integration into Visual Language Models (VLMs) makes Deep Lake 4.0 ready for next-generation AI search.

The Synergy Between PatentPT and Deep Lake 4.0

The integration of Deep Lake into PatentPT's architecture has elevated the patent retrieval experience, combining advanced LLMs with sub-second search capabilities. This synergy allows PatentPT to deliver precise and nuanced responses for patent searches and claims generation without the cost and performance trade-offs associated with traditional databases.

Deep Lake's index-on-the-lake technology enables PatentPT ensemble of models, trained and fine-tuned with Deep Lake dataloader to search and reason through millions of patent records in real-time, providing instant and accurate responses to user queries. By leveraging Deep Lake's multi-modal indexing, PatentPT can go beyond conventional keyword-based searches to understand the full context of patent documents, thereby enhancing both the accuracy and relevance of results.

Conclusion

The journey of building PatentPT, powered by Activeloop Deep Lake underscores the transformative potential of fine-tuned LLMs and cutting-edge data retrieval technologies. By tackling specific domain challenges in enterprise knowledge management with custom solutions like PatentPT, we can push the boundaries of what is possible with LLMs, unlocking more value with multi-modal knowledge that already sits in data lakes of organizations.

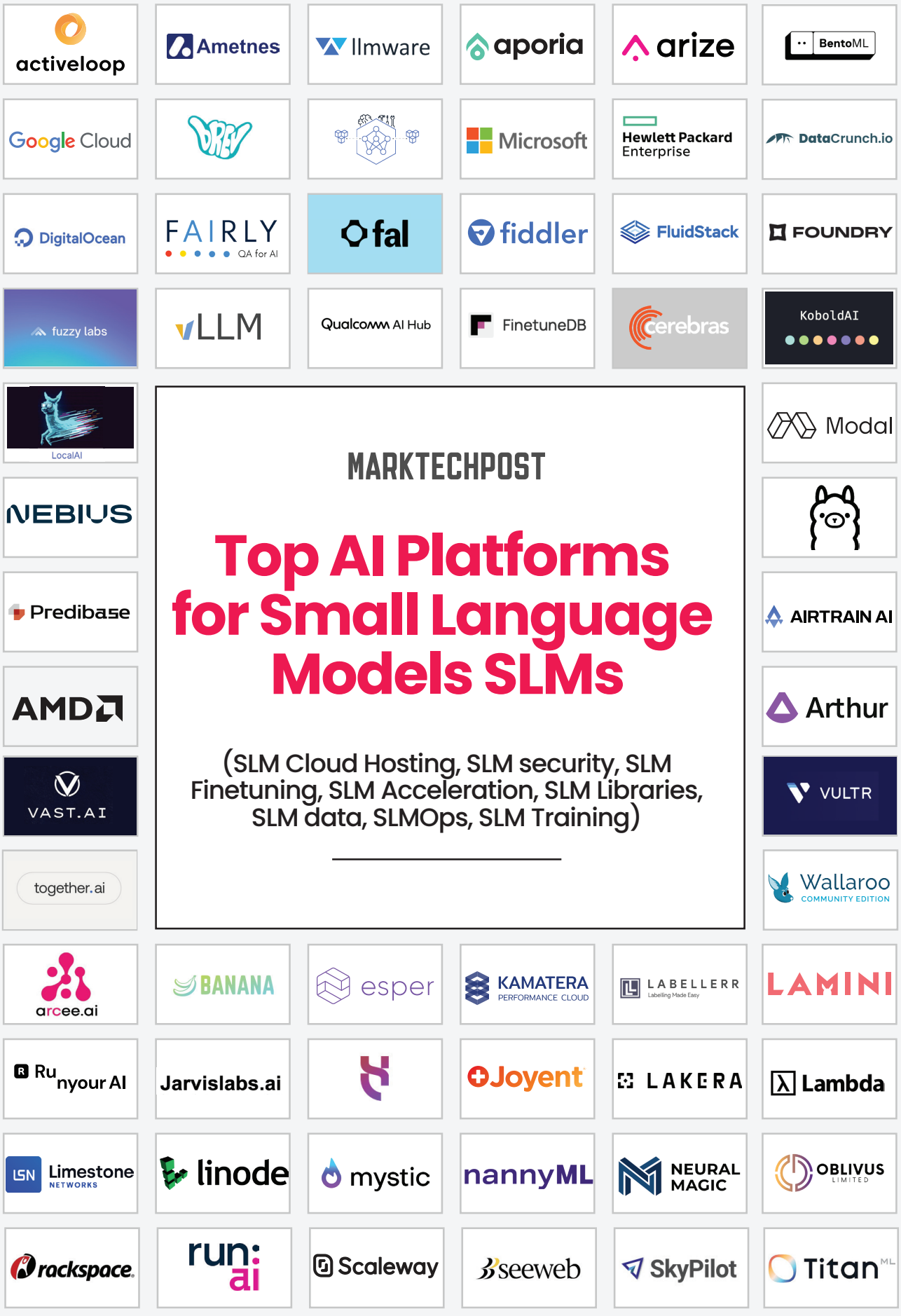
Meanwhile, Activeloop Deep Lake enables cost-efficient and highly accurate AI retrieval by focusing on multi-modality, advanced retrieval techniques, and index-on-the-lake. The combination of these technologies represents a significant leap forward in delivering specialized, high-performance AI solutions.

If you're looking to experience next-generation retrieval and AI capabilities, consider exploring PatentPT for patent-related queries or diving into Activeloop Deep Lake for multi-modal AI search on object storage.

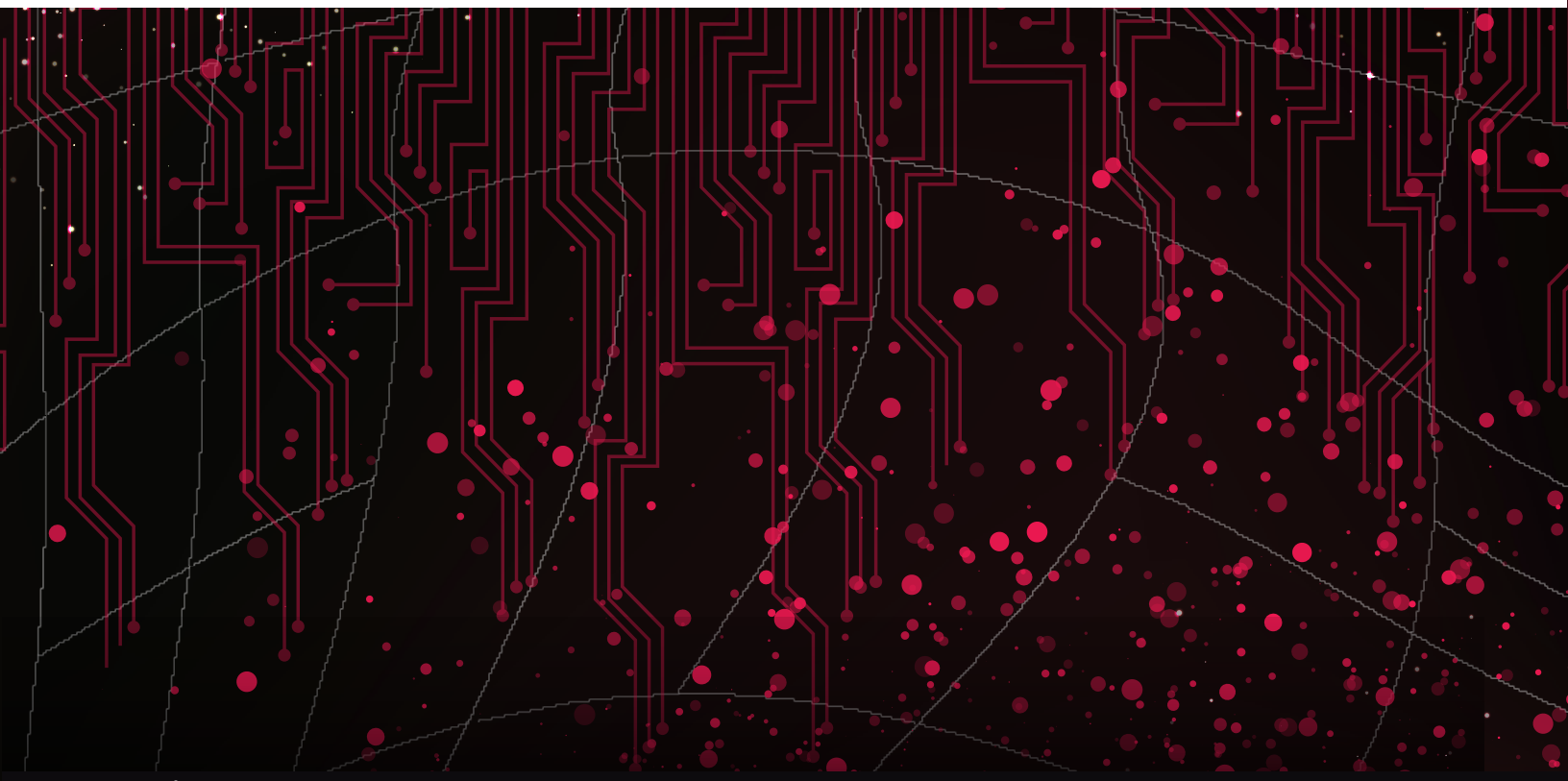
Learn more at activeloop.ai.

Mikayel Harutyunyan
Head of Marketing & Growth, Activeloop

Mikayel Harutyunyan is passionate about marketing, human behavior research, and AI. His journey includes leading developer marketing at Activeloop, launching YouTube Music in Central Europe, and contributing research to combat the impact of fake news. Mikayel's work has been published in journals like Nature Human Behavior and Affective Science, earning him recognition as one of Czechia's top social science researchers under 33, with over 300 citations. He has also launched GenAI360, a popular series of Generative AI courses, along with an eponymous weekly AI newsletter with over 45,000 subscribers.


activeloop
Ametnes
llmware
aporia
arize
BentoML
Google Cloud
Microsoft
Hewlett Packard Enterprise
DataCrunch.io
DigitalOcean
FAIRLY QA for AI
fal
fiddler
FluidStack
FOUNDRY
fuzzy labs
vLLM
Qualcomm AI Hub
FinetuneDB
cerebras
KoboldAI
LocalAI
Modal
MARKTECHPOST
Top AI Platforms for Small Language Models SLMs
(SLM Cloud Hosting, SLM security, SLM Finetuning, SLM Acceleration, SLM Libraries, SLM data, SLMOps, SLM Training)
NEBIUS
Predibase
AIRTRAIN AI
AMD
Arthur
VAST.AI
VULTR
together.ai
Wallaroo COMMUNITY EDITION
arcee.ai
BANANA
esper
KAMATERA PERFORMANCE CLOUD
LABELLERR
LAMINI
Runyour AI
Jarvislabs.ai
Joyent
LAKERA
Lambda
Limestone NETWORKS
linode
mystic
nannyML
NEURAL MAGIC
OBLIVUS LIMITED
rackspace
run:ai
Scaleway
seeweb
SkyPilot
Titan ML

MARKTECHPOST



Design Team: Hema Dhawan, Sanija Jain

Editorial Team: Nikhil, Asjad, Aswin, Divyesh, Pragati, Adeeba, Afeerah, Sajjad, and Nazmi *(IIT KGP)*
Sana Hasan *(IIT Madras)* ,Tanya Malhotra, Aabis Islam

Marktechpost Media, Inc.
California, USA

asif@marktechpost.com

www.marktechpost.com