JUL 11, 2023

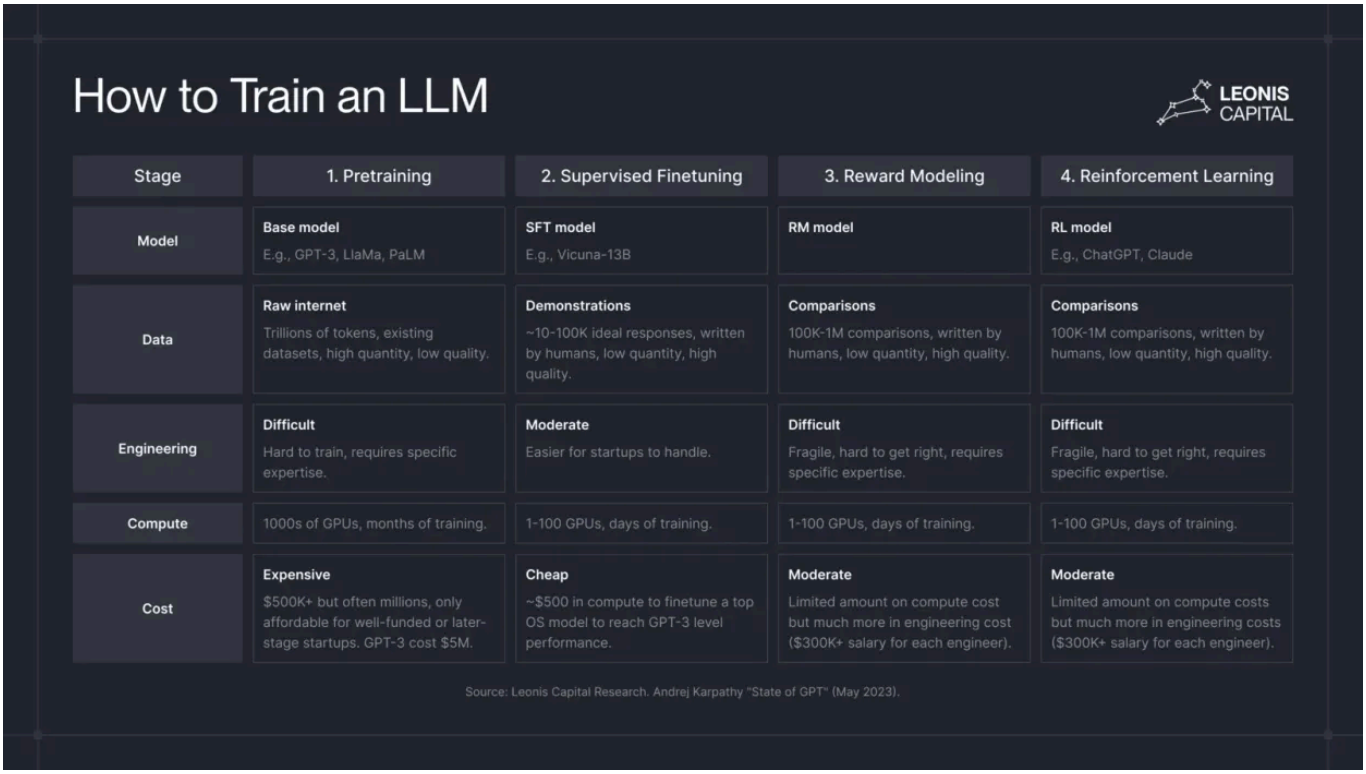
Demystifying the LLM Tech Stack (Part I: The Infrastructure Layer)

How LLM models work.



By Jenny Xiao, Jay Zhao, and Val Gui

Leonis Capital · leoniscap.com/research/demystifying-the-llm-tech-stack-(part-i-the-infrastructure-layer)/



[Full webinar deck available here.]

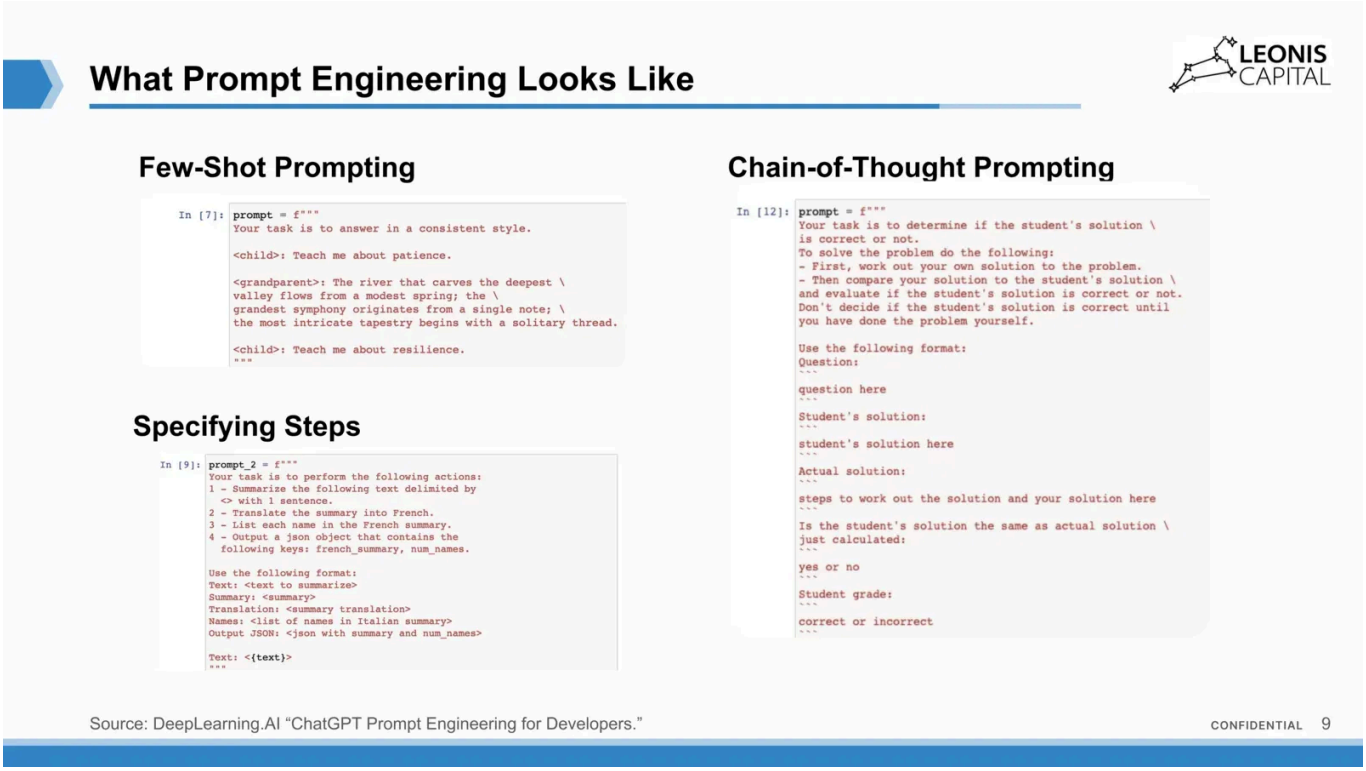
At Leonis Capital, we're passionate about understanding the rapidly evolving landscape of AI, particularly LLMs. As investors in this space, we believe it's crucial to have a nuanced technical understanding of the infrastructure powering AI applications. That's why we recently hosted a series of webinars diving deep into the LLM tech stack, exploring the key decisions AI startups face when building their products. In this post, we'll share some of the key insights from the first session which focuses on the infrastructure layer.

The Three Paths: Prompt Engineering, Fine-Tuning, and Training from Scratch

We break down the three main approaches to using LLMs: 1) using an out-of-the-box model with prompt engineering, 2) fine-tuning an existing open-source or closed-source model, and 3) training a custom model from scratch. Each approach has its own set of tradeoffs in terms of performance, cost, development speed, and potential for building a long-term moat.

Prompt Engineering: The Quick and Easy Route

Prompt engineering involves using an existing model like GPT-4 and crafting specific prompts to get it to perform desired tasks. This is by far the most common approach for early-stage startups, and for good reason. It offers several advantages, including rapid implementation without the need for specialized machine learning expertise. By harnessing state-of-the-art models like GPT-4, startups can capitalize on advanced AI capabilities without the overhead of infrastructure setup or extensive training requirements.



However, prompt engineering does present certain drawbacks that startups must consider as they scale. As usage increases, so do API costs, potentially becoming a significant expense. Moreover, startups using this method have limited control over the underlying model's behavior,

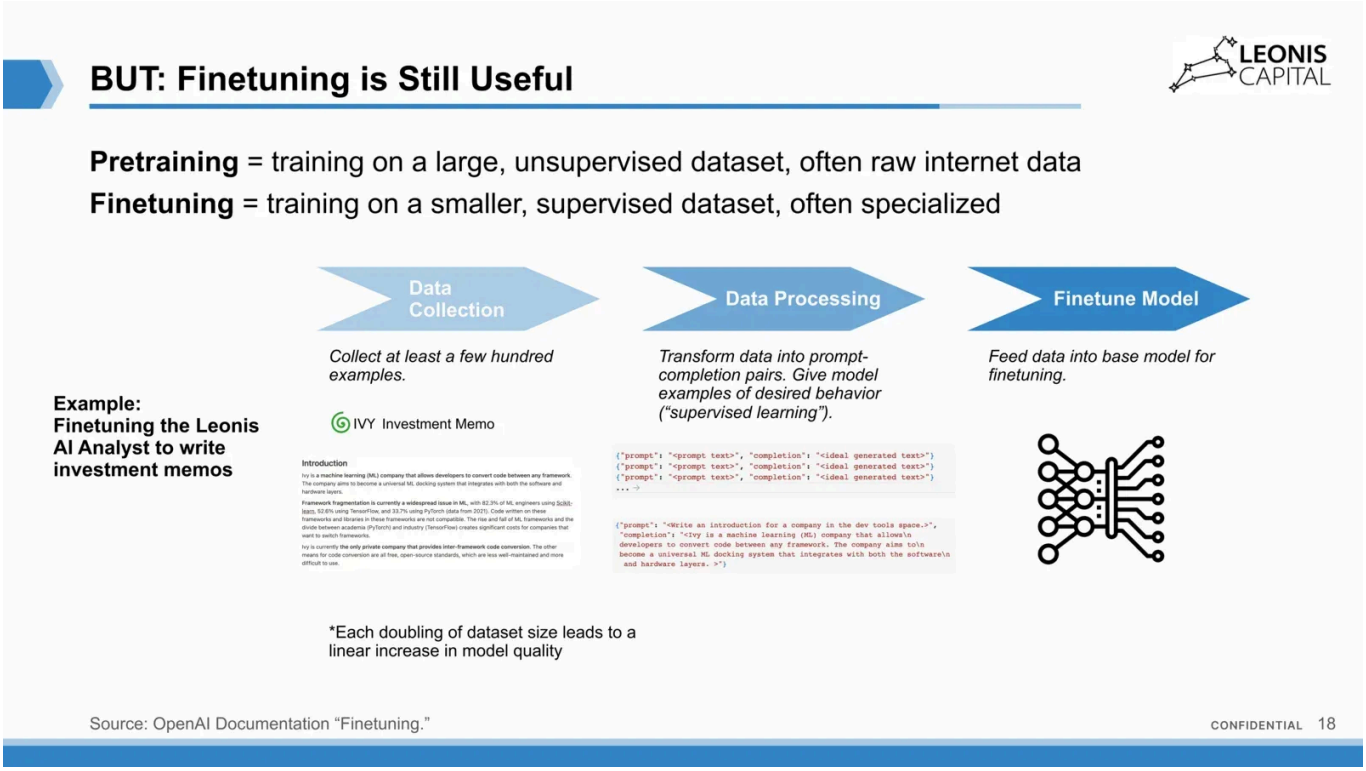
which may pose challenges in tailoring outputs to specific needs. There are also concerns regarding data privacy when relying on third-party APIs, which could impact user trust and regulatory compliance. From a technical standpoint, prompt engineering may be perceived as less defensible compared to more customized approaches.

For many use cases, prompt engineering is an excellent way to validate ideas and reach initial product-market fit. However, it's important to have a plan for how to evolve beyond this approach as a company grows.

Fine-Tuning: The Middle Ground

Fine-tuning involves taking an existing pre-trained model and further training it on a specific dataset for a particular use case. This can be done with both open-source models like LLaMA or closed-source models via APIs.

Fine-tuning offers several advantages. It can significantly enhance performance compared to prompt engineering alone, leveraging the efficiency of compute resources and tokens. This approach allows startups to customize models to suit particular domains, fostering more nuanced outputs tailored to specific business needs. Moreover, fine-tuning often proves more cost-effective as operations scale, offering potential savings compared to relying solely on third-party APIs.



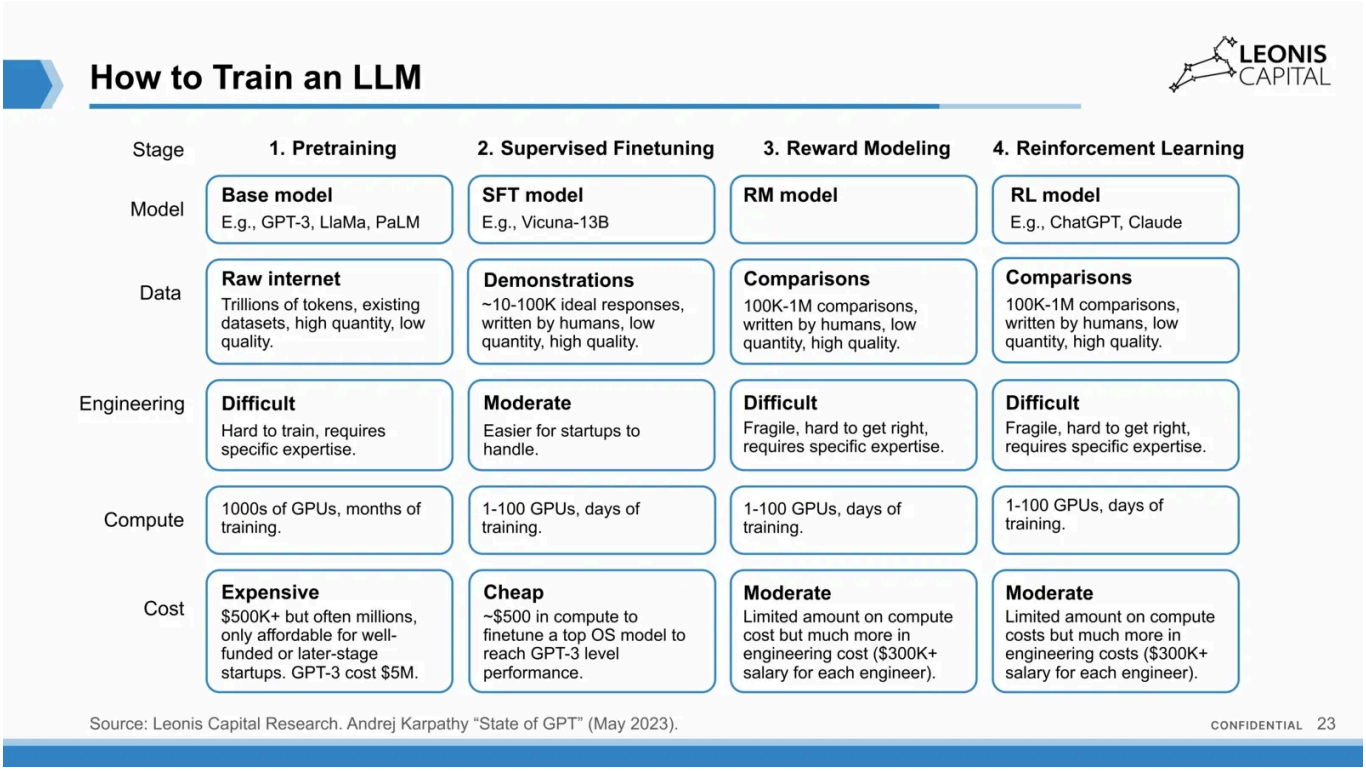
However, implementing fine-tuning requires a higher degree of technical expertise and infrastructure readiness compared to prompt engineering. Despite its advantages, fine-tuned models may still lag behind top proprietary models in performance, particularly in achieving cutting-edge results. Startups also face challenges in injecting deep domain knowledge beyond superficial adaptations, limiting the extent of customization. Furthermore, navigating licenses for commercial use, especially with open-source models, poses potential legal complexities.

Fine-tuning represents a solid middle ground for many AI startups. It allows for some customization and potential cost savings while still leveraging the power of foundation models trained on massive datasets.

Training from Scratch: The Full Control Option

For companies with significant resources and very specific needs, training a custom LLM from scratch is an option. This involves building and training a model architecture on a large corpus of data.

The benefits of training from scratch are profound. It grants complete autonomy over both the model's architecture and the data used for training, facilitating unparalleled customization and control. This approach also enables companies to embed deep domain expertise directly into the model, potentially establishing a robust technical advantage. Additionally, by eschewing reliance on third-party APIs or models, organizations can mitigate risks associated with external dependencies.



However, embarking on this path entails considerable challenges. The costs involved, primarily in computing power, can reach millions of dollars, making it financially prohibitive for most startups, especially in their early stages. Moreover, the endeavor demands top-tier machine learning talent capable of navigating complex model development and training processes. Despite these investments, there remains a significant risk of the custom model falling short of performance benchmarks set by established proprietary models. Furthermore, the lengthy development cycles inherent in training from scratch can delay time-to-market and amplify operational uncertainties.

While training from scratch offers the most control and potential for differentiation, it's simply not feasible for most startups, especially in the early stages. The costs and talent requirements are prohibitive, and the risk of underperforming existing models is high.

Which Path Should Startups Take?

Based on our research and experience investing in AI companies, we generally recommend a phased approach for startups:

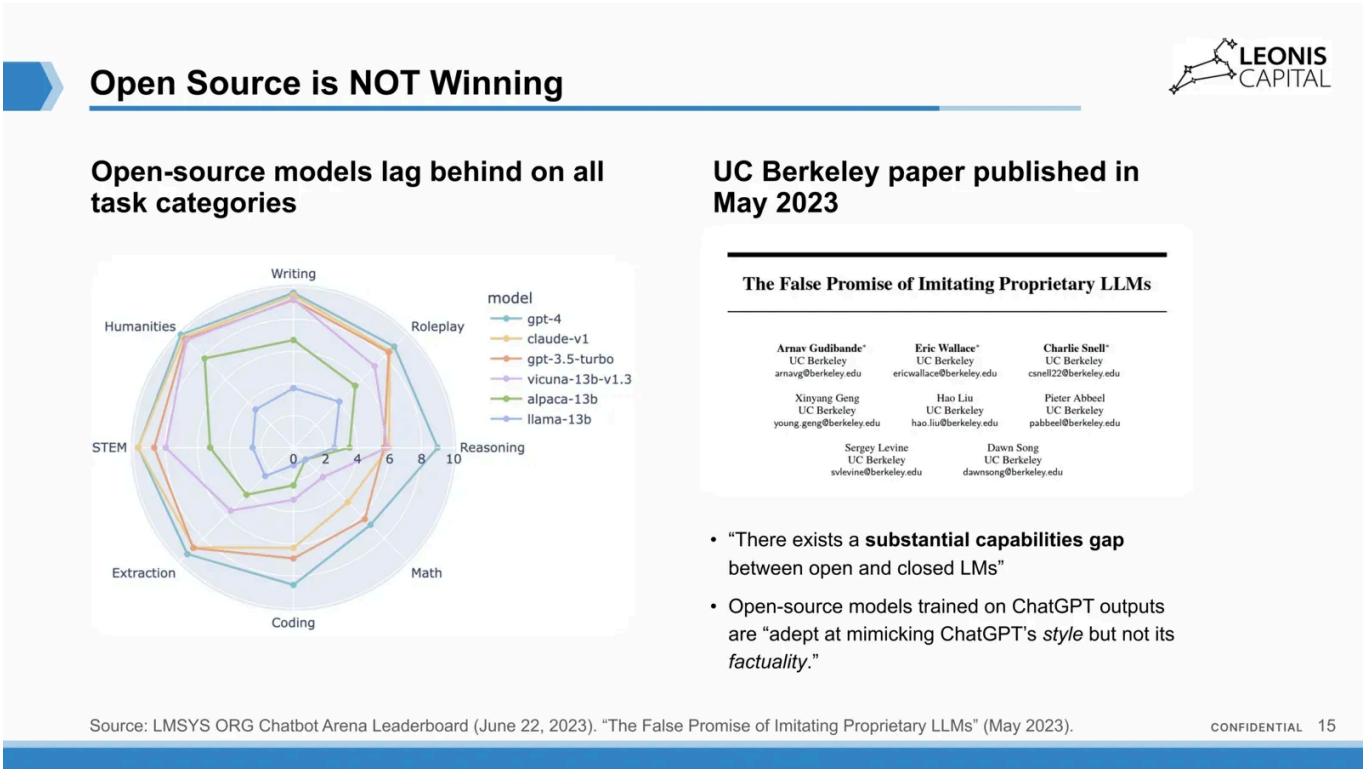
1. **Prototyping & PMF:** Start with prompt engineering using top proprietary models like GPT-4. This allows for rapid iteration and validation of ideas.

2. **Optimizing Costs:** As usage scales, look to optimize prompts, implement caching strategies, and potentially fine-tune models to reduce API costs.
3. **Building a Moat:** Once product-market fit is established and resources allow, consider training custom models to create true technical differentiation.

This approach allows startups to move quickly in the early stages while laying the groundwork for building deeper technical advantages over time.

The Open Source Question

There's been a lot of excitement recently about the potential for open-source models to challenge the dominance of proprietary offerings from companies like OpenAI and Anthropic. While open-source models have made impressive strides, our research suggests the reality is more nuanced.



Open-source models still lag significantly behind their proprietary counterparts in terms of performance across various tasks. The top proprietary models consistently outperform open-source alternatives in benchmarks and real-world applications. Moreover, commercial adoption of open-source models is hindered by restrictive licensing terms that limit their use in commercial settings. This constraint poses challenges for startups and enterprises looking to integrate these

models into their products and services. Achieving state-of-the-art results with open-source models demands substantial compute resources and specialized expertise. This requirement can be a barrier for many organizations, particularly smaller startups with limited resources.

That said, open-source models do offer advantages in terms of cost, privacy, and customizability. For many applications, they may be "good enough" while providing more control and potential for differentiation.

What We Look For in AI Startups

As investors evaluating AI startups, understanding these infrastructure decisions is crucial. Here are some key things we look for:

- **Thoughtful model selection:** Does the team have a clear rationale for their choice of base model and approach (prompt engineering vs. fine-tuning vs. training)?
- **Data strategy**: How does the company plan to acquire the data needed to fine-tune or train custom models in the future?
- **Regulatory awareness:** Is the team considering potential data privacy regulations and other compliance issues in their infrastructure choices?
- **Long-term vision:** Do they have a roadmap for how their AI infrastructure will evolve as the company grows?
- **Realistic expectations:** Are they overpromising on the capabilities of fine-tuned or custom models, especially in the early stages?

We're cautious of companies raising large rounds solely to train custom models before proving product-market fit. While building proprietary models can be a powerful moat, it's rarely the right first step for an early-stage startup.

The Future of LLM Infrastructure

Looking ahead, several key trends are poised to shape the future of LLM infrastructure. Firstly, **specialized models** tailored to specific domains are expected to proliferate, promising enhanced performance and efficiency for niche use cases. Secondly, **advancements in hardware architectures and AI-specific chips** will likely transform the economics of deploying large-scale models, potentially reducing costs and improving computational efficiency. Additionally, **hybrid approaches** integrating proprietary APIs, fine-tuned models, and custom components are anticipated to emerge as the preferred strategy, enabling organizations to optimize performance while managing costs effectively. Moreover, the fine-tuning process is set to evolve with **new techniques** enhancing its ability to inject domain-specific knowledge and enhance overall model performance. Lastly, **increasing regulatory scrutiny** around AI safety and data privacy will

shape how companies navigate the use of third-party models and APIs, underscoring the importance of compliance and ethical considerations in AI development. These developments highlight a dynamic landscape where strategic decisions in AI infrastructure will play a crucial role in future technological advancements and market competitiveness.

The world of LLM infrastructure is complex and rapidly evolving. For AI startups, choosing the right approach is a crucial decision that impacts development speed, costs, performance, and long-term defensibility. At Leonis Capital, we believe that understanding these technical nuances is essential for making informed investment decisions in the AI space. By combining deep technical knowledge with traditional startup evaluation criteria, we aim to identify and support the most promising companies building the future of artificial intelligence.

Stay tuned for our upcoming posts diving deeper into LLM developer tools and strategies for building lasting moats in AI applications!