Problem

The greatest bottleneck to AI development is not data, compute, or energy – it's the way we build it. The current building process requires engineers to interface with code, developer platforms, and other mediums that are highly idiosyncratic, documentation heavy, and frustrating to work with. This consequently takes up most of the developer's limited time and mental bandwidth, instead of what's actually important: the architecture and intelligence of their models. Specifically, they must tirelessly forage for information on libraries, frameworks, syntax, etc.[1] – set up configurations for their infrastructure, environment, and data pipelines[2] – write the actual code for the system and enter a painfully slow loop of debugging until it eventually works[3]. The problem is exponentially magnified when expanded to teams, organizations, and the developer community. Current building mediums facilitate knowledge siloing at every level, locking up communication and collaboration. Additionally, education is forced to focus on all the technical nuances of how to work with these mediums rather than the foundational concepts and mathematical theory underlying AI systems. All of this needless complexity is crippling the progress of millions of developers working on critical problems – ultimately turning what should be an AI golden age into a period of sparse advancements made by a handful of big players.

Product

GraphSpace takes the next step in how AI is built. On the platform, models, pipelines and infrastructure are represented in the same way they are explained and conceptualized: visually, where information flows from one component to another.

GraphSpace makes it possible to work at any level of a system with total clarity, from low-level ML operations to high-level model orchestration. For example, if a user is working on an agentic AI system, they can zoom into one of its agents (a GPT), then into an attention head, into a neural network, and finally into a neuron – learning or editing at any level as needed. This allows team members to collaborate within the same environment, regardless of what level of the system each is focused on.

Systems can be built directly by the user or generated by prompting a GPT. Real-time updates are streamed to the user during model training and inference. A visual version control system makes it easy to keep track of updates. And because models are expressed as graphs, and not code, they can be stored in GPU memory, ready to run. This means no more waiting for frameworks to startup or models to compile and load into memory[4]. All of this makes it possible to have immediate results, zero friction, and a seamless flow from concept to creation.

About Me

One of my earliest memories is taking apart my parents coffee maker, plugging it into an outlet, and watching it explode causing all the breakers in our apartment to flip. I was a kid with a compulsive need to deconstruct and discover. Eventually I graduated from taking things apart to building them – resulting in many projects over the years, a hovercraft, rocket motor thrust stand, gyroscope rocket, DIY drone, hybrid rocket engine, electric skateboard. When I got my hands on a computer I dived into designing more complex and ambitious projects but struggled to afford building them. Both my parents were teachers who lived paycheck to paycheck so I needed my own source of income. I got my parents to make a brokerage account for me and after discovering a tutorial for automating trading with the brokerage's python API my programming journey began. This was over ten years ago, and since then I used my programming skills to analyze investment data and grow my account 30x, to create two mobile apps, and to build the prototype for GraphSpace.

I attended the University of Washington where I studied economics and computer science. While there I also took finance and engineering classes and I joined the university's rocket propulsion and satellite clubs. I believe that exploring different fields allows for the cross pollination of ideas, giving me a robust and diverse

---

[1] 66% of AI/ML developers spend more than 10 hours per month on configuration and setup according to CIVO. Microsoft estimates 50% of developer time is spent information foraging.

[2] Google estimates 5% of time is spent on actual AI/ML engineering – where the other 95% is spent on creating configurations and data pipelines, and setting up infrastructure.

[3] Microsoft estimates developers spend 11% of their time debugging.

[4] Startup time typically ranges from 10 to 60 seconds due to library loading, device discovery and setup. Compiling and loading can take 3 to 30 seconds, depending on how complex the model is. This can be very frustrating, especially when debugging and having to restart many times to find the cause of a problem.

knowledge base to draw from. Putting that into practice, and observing the struggles of fellow students trying to build and understand AI, the idea for GraphSpace emerged. I've been bootstrapping the company ever since graduating in 2023. And because of my successful investment portfolio, I am able to work on it as my full time job. I've also cut personal costs down to the theoretical minimum, so everything can go to the company.

Since starting GraphSpace, I've learned more than in any other period of my life. My approach to problem solving evolved drastically and many eye-opening lessons were drilled into my mind. Despite the many hard-hitting mistakes, my drive to see GraphSpace come to fruition is stronger than ever. And after two years of solo development, I'm now exclusively focused on taking GraphSpace from prototype to product.

Competitive Landscape

There have been many attempts to simplify AI development. Companies including Microsoft, Google, IBM, and DataRobot all have visual drag-and-drop interfaces for composing high-level AI pipelines - where nodes that represent things like ML models, python scripts, or data sources are dragged onto a canvas and connected together. Startups like Visual Blocks, PerceptiLabs, and Cerbrec focus on the lower-level side and allow users to create individual ML models within a visual drag-and-drop interface, acting as a wrapper around legacy code frameworks like TensorFlow. However, the biggest complaint from developers is that all these platforms end up forcing them into a one-size-fits-all approach that fails to address the nuanced requirements of different domains. Also, none of these platforms go beyond connecting surface-level components together: they don't display things like the actual tensors going from operation to operation or model updates during training or data transformations. They force users into a fixed plane of abstraction rather than allowing users to travel freely through the abstraction hierarchy, resulting in oversimplifying and blocking off parts of the system which could be central to the user's problem space. These platforms are codeless for the sake of being codeless and ultimately don't add value beyond what developers already use.

Market Opportunity

The artificial intelligence market presents an extraordinary potential for growth, expanding from $300 billion (2024) to $3 trillion by 2032. With the AI/ML development platform segment projected to surge from $5 billion to $75 billion over the same period. Also, the $250 billion edTech market is growing at an average rate of 15% per year, with Computer Science being the one of the largest contributing education sectors. There are also around 10 million developers worldwide still using legacy AI frameworks. GraphSpace is strategically positioned to capitalize on this convergence by targeting all three segments: companies, educational institutions, and the broader developer community. GraphSpace will directly address the needs underpinning the entire AI industry.

What will development look like in ten years? Platforms are gradually becoming more aligned with how we think and process information.Throughput is highest when information flowing from computer to person is visual. Users will prompt the computer to get it to build something and it will show what it has built. Then users will edit by manipulating the visual medium and prompting it further. In addition to GraphSpace, this workflow is becoming increasingly popular especially with emerging development platforms (such as Webflow and Zapier). This will eventually become the norm, as platform idiosyncrasies and heavy documentation fade from popular usage. In addition to this paradigm shift, there are significant economic and social tailwinds:

- There is a widespread and growing curiosity about AI/ML reflected in the surge of highly popular online courses, attracting learners from around the world[5].
- Explosion in enterprise AI adoption[6]
- AI talent shortage[7].
- The existential risk from AI is a result of the lack of players. The high barrier to entry is manufactured by a cartel of AI companies, leveraging an illusion of complexity to protect their position. This creates a world where the person who knows the most jargon beats the person with the best ideas. Lowering the entry barrier will help democratize AI – directly combating the topheaviness of the industry and mitigating existential risk.

[5] Coursera had 3 million AI/ML course enrollments in 2024. CodeAcademy's AI courses had enrollment up 55% in 2024.
[6] 78% of companies have adopted AI technologies as of 2025.
[7] Reuters research estimates there is a 50% AI talent gap in the US.

Business Roadmap

GraphSpace will have a SaaS subscription model, launching with pricing for institutions (year 1), and then adding pro and free tiers (year 2). First targeting educational institutions for the initial market segment. This is ideal for multiple reasons:

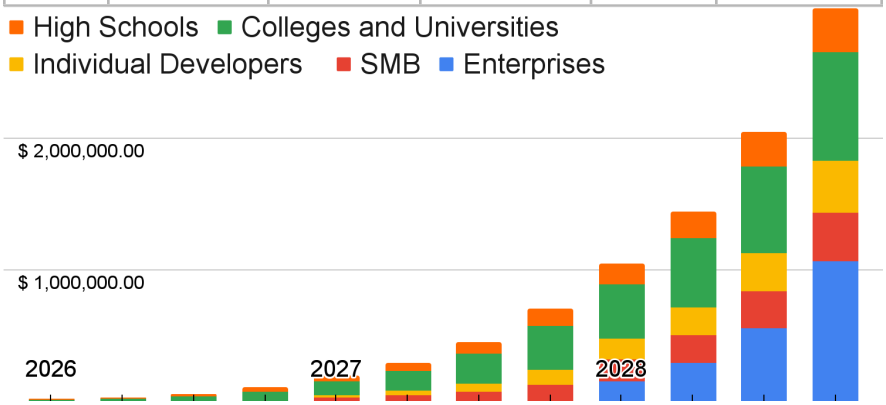
- Clear value proposition: educational institutions have an immediate need for the platform to help teach students how to build AI through a common sense, easy to understand interface.
- Controlled environment: the academic setting provides a contained, low-risk environment to find and fix problems.
- Rapid feedback: compared to other markets education is ideally situated for asking questions, providing feedback and iterating new technology.

This education-first strategy will allow for rapid iteration of the platform before scaling to the broader market. It will also mean students will be familiar with the platform once they enter the job market, possibly putting pressure on companies to drop legacy frameworks. Next, SMB, startups, and individual developers will be targeted, further refining the platform with a focus on data pipelines, infrastructure, copilot[8], autocomplete[9], and public repos[10]. Finally, the product will be released to large enterprises, where the focus will be on mechanistic interpretability tools[11], converting legacy code[12], integrations[13], and creating custom processing architectures[14].

Financial Projections

GraphSpace will follow a staggered market entry strategy. New segments will be targeted at the start of each year, over the course of three years. Pricing will be refined once costs and demand curves become more concrete. CAC and churn numbers are based on segment averages. And the initial quarter acquisition and growth estimates are based on what is likely possible given a moderate to aggressive marketing strategy.

	Year 1 (2026)			**Year 2 (2027)**			**Year 3 (2028)**			
Added Segment	Colleges and Universities			High Schools			Small/Medium Businesses		Individual Developers	Enterprises
Pricing [15]	$20 (/student)			$20 (/student)			$500 + $50/seat		$20	$10k + $1k/seat
CAC	$300			$200			$150		$30	$5,000
Initial[16]	10			10			30		1000	5
Growth [17]	2x	1.5x	1.3x	2x	1.5x	1.3x	2x	1.5x	2x 1.5x	2x
Churn	3%			3%			12%		10%	4%



[8] At this point, there should be enough user repositories and therefore enough data to train an AI model that outputs other AI models from an input prompt. Allowing the user to ask it to build a new model or modify an existing one.

[9] Also because of the user model repositories, an autocomplete feature can be created. This would predict what the user is building and actively suggest what they need to do next.

[10] Public repositories will have significant network effects and foster engagement from the wider developer community.

[11] Understanding how models work under the hood, and what cause/effect relationships led to their output. This is important for compliance, research into internal dynamics, and finding out what needs to be done to improve the model. GraphSpace will provide tools for quickly visualizing, analyzing, and modifying model feature circuits, and superpositions.

[12] Converting TensorFlow or Pytorch models to models within GraphSpace.

[13] Most importantly HuggingFace for model parameters. This would allow users to use model parameters from HuggingFaces's vast collection immediately in their GraphSpace models.

[14] An AI model's architecture can directly inform hardware design. We're already seeing custom chips being made for LLMs, and as the AI marketplace continues to grow, demand for more custom architectures will grow as well. GraphSpace will provide an easy way for users to create custom, model-derived hardware.

[15] Cost per month. The chart assumes 5 seats for SMB and enterprises.

[16] The number projected to be acquired in the first quarter that the segment is targeted.

[17] Estimated quarterly growth for each year from the initial year to year three.