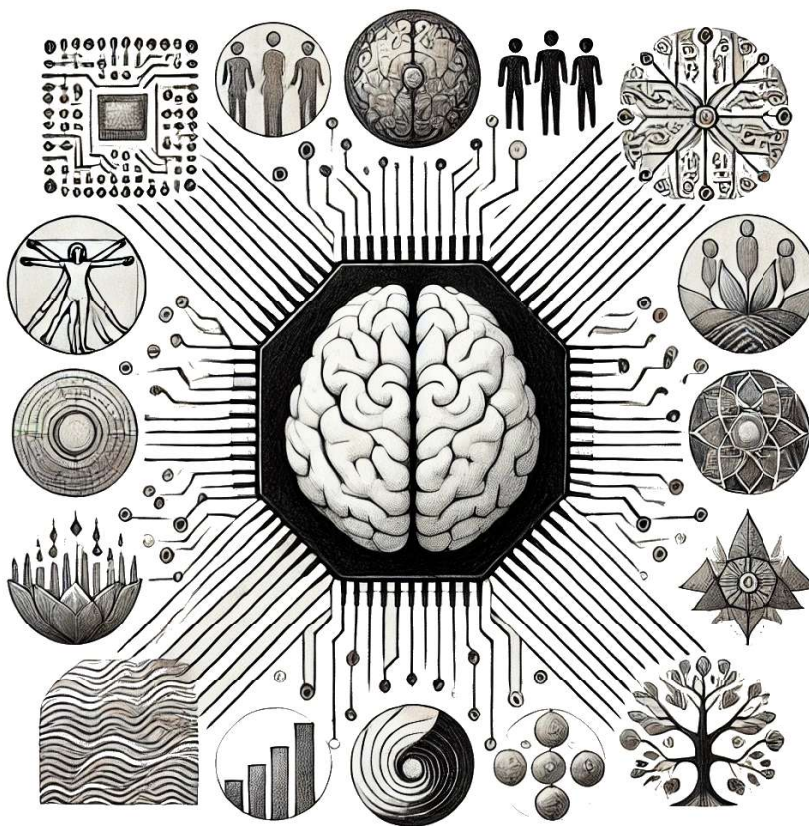


Vision Paper (Preview Edition):

Bringing Bio-Inspired Computing to Life

A Blueprint for a Paradigm-Shifting Technology and More Exponential
Improvements in Computing (Beyond Moore's Law)



Dr. Pouya Baniasadi

Cambridge, England, UK

©2025 Cambridge Brain Ltd

First published: 22 Mar 2025, Last updated: 25 Mar 2025

Version: 0.92.r (partially redacted)

Copyright and Intellectual Property Notice

This material is is protected by copyright under U.K. Copyright laws and is the property of the author and the company (*Cambridge Brain Ltd* registered in England and Wales with company number 13892180). Unless you obtain a written permission, you may not copy, reproduce, distribute, publish, display, perform, modify, create derivative works, transmit, or in any way exploit any such content, nor may you distribute any part of this content over any network, including a local area network, sell or offer it for sale, or use such content to construct any kind of software product or database. Its contents, as a whole or in part, are not to be used as training data for a language model or any other form of software or computer hardware design. For permission to use the content please contact the author.

Permission to access the material included in this book is subject to the reader's recognition of author's intellectual property rights to any original material related to a process or design, described or outlined in the book. Furthermore, the reader acknowledges that the use of the material is intended solely for non-commercial use. Any other use (for example, any design or software product related to the original material) requires written permission from the author or a license agreement. If the reader is not in agreement with this recognition, please abort accessing the material beyond beyond this point and return it to the publisher.

The author is open to any questions or requests for permissions. To discuss these matters, please contact the author at: books@cambridgebrain.co.uk and please ensure that the subject line contains the book title.

Reader's Guide

This book outlines a vision for a technology that does not yet fully exist. Specifically, it describes a new form of computational stack, with architecture that more closely resembles biological computation than current architectures such as the von Neumann architecture. The document draws inspiration from historical vision documents of profound influence, such as Francis Bacon's *Novum Organum* and *New Atlantis* in the 17th century, which led to systematic investments in science and technology. It is also inspired by mid-20th-century vision papers that anticipated personal computers years before they became a reality. They proved necessary in laying out a path for innovators to pursue seemingly impossible ideas. Some of those visions eventually altered the course of history and shape our world today—remarkable achievements considering the technologies they described did not exist at the time.

Realizing a vision is far more complex than developing new technologies alone. Rather than taking a purely scientific or technical approach, the aim here is to present a clear picture of the opportunity space and to offer an overview of all major aspects involved in bringing this new technology to life. Success depends on communities of innovators who understand what they are creating, why it matters, which technical directions are worth exploring, and how to sustain a long-term strategy.

This book aims to be relatively accessible to a general audience, while providing a comprehensive overview of the technical and strategic thinking behind this initiative, presented in clear, detailed language. It is divided into five parts:

Part I - An Overview of Bio-inspired Computing provides a concise introduction to our approach by discussing the broader context of bio-inspired computing. This includes emerging technologies and opportunities in this area to improve computer design and architecture across different levels of computing technology. For a higher-level overview of the computing market, refer to Appendix A, which is followed by an analysis of barriers to entry for emerging technologies in Appendix B.

Part II - Our Blueprint for Next-Generation Computing (Bio-Arch™) outlines the high-level technical details of our innovative approach, emphasizing what sets it apart from alternatives in bio-inspired computing. This section also lays out the theoretical assumptions underpinning the approach in plain language, facilitating critical engagement with technical experts.

Part III - Path to Value Creation for Bio-Arch™ Computing explores the product opportunities enabled by the emerging technology. This includes short-to-medium term opportunities validated up to Technology Readiness Level 3, as well as potential applications that require feasibility studies or significant exploratory innovations.

Part IV - Executing the Vision examines case studies of successful emerging technologies, highlighting lessons learned that have shaped our execution strategy. This section culminates in a long-term plan for realizing the vision, starting from modest beginnings.

Navigate to the parts most aligned with your interests

This document is structured for multiple audiences, and not every section is equally relevant to all readers. I recommend using the **table of contents** to locate the parts most aligned with your interests. Below are specific reading suggestions for different types of readers:

Technically-focused readers: For those critically evaluating our technical approach, the most important section is the discussion of proofs of concept in Section 13.2. This section establishes the technical narrative and it is the evidence for far-reaching potential of this emerging technology. The technical merit of this section is critical in the viability of the emerging technology.

Scientifically-curious readers: To explore the broader opportunity of bio-inspired computing, Chapter 3 is essential. It maps the potential of this field and provides the necessary context for understanding Part III, which focuses on value creation. Relevant additional context can be found in Appendix B.

Product-focused readers: Readers interested in commercial applications should focus on Part III, which outlines opportunities for creating user value. Background information in Chapter 3 and Appendix B will provide helpful context.

Business and execution strategy readers: For insights into the strategic execution of this initiative, Part IV is key. This section includes historical lessons and outlines the high-level stages necessary for maximizing the chances of realizing this vision.

General audience: For an overview of the big picture, Part I is recommended in its entirety. It provides a clear introduction to the emerging technology and the opportunity space driving this initiative.

Notes on the Preview Edition

This version of the book is still in the preview stage and some content may be incomplete, insufficiently reviewed and references may be missing. This version may contain unintentional errors and inaccuracies. We encourage the reader to contact the author to provide feedback and suggestions by visiting: <https://cambridgebrain.co.uk/books>

Alternatively, you can contact us via email at books@cambridgebrain.co.uk. For emails, please ensure that the subject line contains the book title.

Acknowledgements

The motivation to write this book, as well as its overall structure, has been profoundly shaped by the communities I've been fortunate to be part of.

I am deeply grateful to *Entrepreneur First* for running the *Polaris Fellowship*, which created a unique environment for exploring early-stage ideas. I especially thank *Arnaud Shenck* for introducing me to historical works documenting the conditions and forces at the origins of scientific and technological movements. Reading Francis Bacon's writings from the dawn of the scientific revolution, alongside early visionary proposals of computing that anticipated personal computers and mobile phones long before they existed, inspired me to turn scattered notes into a coherent manuscript — the foundation of this book.

My gratitude also extends to my fellow Polaris participants, and in particular *Dr. David Jordan*, for their encouragement and for the many thought-provoking discussions that elevated my thinking on the topics covered here.

I owe a special thanks to the *Cambridge Accelerate* program, run by the University of Cambridge, for the chance to learn from a wide range of industry experts. I am especially grateful to *Michael Grange*, Program Manager at ARM and my lead mentor, whose detailed critiques and feedback helped shape much of the non-technical content in this book. I also thank *Tony Kypreos* for his encouragements to write this book to articulate a deep technology perspective, and to follow that path.

Finally, I want to express my heartfelt appreciation to *Luther Phillips*, Program Manager of Cambridge Accelerate, and *Jack Davis*, the program's Administrator. Their efforts have created an extraordinary space for founders at the earliest and most vulnerable stages of their journey. Luther's unwavering dedication to nurturing the next generation of entrepreneurs — especially at those critical junctures where ideas are still forming — has made a world of difference to me and to many others forging similar paths.

Contents

Copyright Notice and Terms of Use	i
Reader's Guide	ii
Acknowledgements	v
Introduction	1
Nature points to an extraordinary path	2
Distinction with other bio-inspired computing paradigms	3
How to assess early signs of promise	5
The case for a long-term focus to maximize value	7
Part I - An Overview of Bio-inspired Computing	11
1 Aspirational Vision and Impact	11
1.1 Beyond the Bicycle for the Mind	11
Nature's Computers Point to a Path beyond Moore's Law	12
1.2 Supercomputing without Supercomputers	13
1.3 The Need for a Paradigm Shift	14
1.4 The Moonshot Scenario	16
1.5 Aspirational Challenges	18
The End of Moore's Law	18
Scaling Laws of Capabilities	20
Sustainability of Trillion-Dollar Supercomputing Infrastructure	20
Interpretability	21
Inequality and Technological Capabilities	22
Edge Computing	23
1.6 Unique Technical Presuppositions of this Proposal (Bio-Arch™)	23
1.7 Evidence of Feasibility	25
1.8 An Execution Roadmap from a Humble Beginning	26
The importance of strong proofs of concepts	26

	The importance of tangible value creation	27
2	Overview of Current Bio-inspired Computing Approaches	29
2.1	Early Computing	30
2.2	Modern Machine Learning	32
	Artificial Neural Networks and Hopfield Networks	33
	Recurrent Neural Networks and Deep Learning	34
	Reinforcement Learning	35
	Transformers and Attention Mechanisms	36
	Sparse Computation and Mixture of Experts	37
	Existing Gaps with Human Learning	38
2.3	Neuromorphic Hardware	38
	Analog and Mixed-Signal Neuromorphics	39
	Digital Spiking Neuromorphics	40
	Memristor and Compute-in-Memory Neuromorphics	41
	Photonic and Emerging Neuromorphic Approaches	42
	Limitations and Challenges	43
2.4	Artificial General Intelligence (AGI)	44
	Controversies and Methodological Debates	45
	Universally Accepted Requirements	45
	Efficiency: A Central Criterion	45
	Multidisciplinary Perspectives and Historical Lessons	46
2.5	Other Bio-inspired Computing Approaches	46
	DNA Molecular Computing	47
	Immune-Inspired Computing	47
	Organoid Intelligence	48
	Evolutionary Algorithms	48
	Swarm Intelligence	49
3	The Landscape of Bio-inspired Computing Architectures	53
3.1	Computational Abstraction Levels	53
	Introduction to Abstraction Layers	53
	What are Abstraction Layers in Computing	54
	The Computing Abstraction Stack	55
3.2	Bio-inspired Physical Layer (Hardware)	56
	Physical Layer Research Questions:	58
3.3	Bio-inspired Microarchitecture	58

	Replicating Computing Structures	59
	Replicating Emergent Properties of Structures	59
	Microarchitecture Research Questions:	60
3.4	Bio-inspired Instruction Set Architecture (ISA)	60
	ISA Research Questions:	61
3.5	Compiler Design for Bio-inspired ISA	61
	Compiler Design Research Questions:	62
3.6	Bio-inspired Acceleration through Focusing on Problem Classes	62
	Acceleration Research Questions:	63
3.7	Bio-inspired Learning	64
	Research Questions	64
3.8	Concluding Remarks	64

Part II - Bio-Arch™: Our Blueprint for Next-Generation Computers 69

4	An Alternative Methodology for Bio-inspired Computing	69
4.1	Nature's Discovery of Information Theory	69
	Rediscovery by Humans	70
4.2	Epistemology of Bio-inspired Approaches	71
5	Technical Vision (Simplified)	75
5.1	Design Overview of the Bio-Arch™ Computing Architecture	75
	Core Technical Challenge	75
	Technical Differences of Bio-Arch™ Architecture with Predominant Paradigms	76
	Mathematical Design	78
5.2	Expressivity of a General Problem Solver	78
	General Problem-Solving Ability	78
	Selecting the Right Level of Expressivity	79
	A Computer for the NP Complexity Class	79
5.3	Basic Calculations: The Symmetry Processor	80
	Nature of Basic Calculations in Natural Systems	80
	Symmetries as Core Operations	80
5.4	Instruction Set Architecture (ISA) for the Symmetry Machine	81
	Symmetry-Based ISA	81
	Projection Operators for NP-Complete Problems	81
5.5	Compiler for General Problem-Solving	82

5.6	Learning from Data Using Weak Convergence	82
	Integrating Learning into the New Paradigm	82
	Efficiency in Learning	82
5.7	Conclusion	83
6	Proofs of Concept for Bio-Arch™ Computing	85
6.1	What constitutes a good proof of concept?	85
	Early PoC Indicators of Disruptive Computing Technologies	87
6.2	PoC 1: Computational Search - Pushing the Boundaries of Computational Ca- pability	87
6.3	PoC 2: Transformer (Compiler) - Generalization Across Problem Domains . .	89
6.4	PoC 3: Acceleration and Integration with Hardware	91
6.5	Planning: Envisioning a Path Forward	92
7	The Body of Work behind the Proofs of Concepts (Redacted)	93
7.1	Proof of Concept in Network Analysis and Computational Search	93
7.2	Proof of concept for a simulated ISA	93
7.3	Foundations of a Learning Paradigm	93
8	Existing Foundational Work (Redacted)	95
8.1	Structural Theory for Effective Heuristics	95
8.2	A Mathematical Explanation for Brain Efficiency	95
8.3	Optimization by Leveraging Group Theory	96
8.4	A Model of Natural-Computing	96
8.5	Symmetry Operations using Inhibitory and Excitatory Neurons	96
Part III - Path to Value Creation for Bio-Arch™ Computing		101
9	Product Opportunities using Existing Proofs of Concepts (in 2025)	101
9.1	Commercial Mathematical Software	101
	The Operations Research Software Space	101
	Pattern Recognition Software Space	102
	Significance of the Proof-of-Concept	104
9.2	Close-to-hardware Integration	105
	About the Broader Opportunity Space	106
	Hardware Accelerations for Targeted Applications	107
10	Product Opportunities using Existing Formulations	109

10.1	Computational Booster for AI models	109
10.2	Booster for Drug discovery AI models	111
	Further Reading	113
10.3	General-purpose Bio-Arch™ Edge architecture	114
	Specific Examples in RISC Architecture Adaptations	114
11	Product Opportunities requiring Significant Innovation	117
11.1	Analogue-Digital Bio-Arch™ Computing Architecture	117
11.2	The Crown Jewel: Rapid AI learning	119
Part IV - Executing the Vision		123
12	Execution Challenges for a New Paradigm	123
12.1	Forming Institutes for Bio-inspired Computing	123
	Objectives	124
	Framework for Execution Strategy	125
12.2	Adoption by Talent	125
	Talent Case Study: Deep Learning after AlexNet	126
	Talent Case Study: Quantum Computing after Shor's Algorithm	128
	Strategy Principle 1.1: Pre-commercial Proofs of Concept	129
	Strategy Principle 1.2: Technical Incentives	129
	Strategy Principle 1.3: Prioritizing Talent Acquisition	130
	Strategy Principle 1.4: Going for the Moonshot	131
12.3	Third-Party Developer Ecosystem Adoption	132
	Ecosystem Case Study: NVIDIA's CUDA	133
	Ecosystem Case Study: ARM vs. Intel	134
	Strategy Principle 2.1: Establishing a Network of Constructive Partnerships	135
	Strategy Principle 2.2: Integration with Current Paradigms	135
	Strategy Principle 2.3: Developing a Long-Term Stack	136
12.4	Commercialization and Access to Capital	136
	Commercialization Case Study: ARM Licensing Model	137
	Commercialization Case Study: NVIDIA's Focus on Graphics Cards	138
	Pre-commercial Case Study: Neuromorphic Computing	138
	Strategy Principle 3.1: Narrowing Competition through Dominating Niche Applications and Focused Expansion	140
	Strategy Principle 3.2: Prioritizing the Creation of Key Commercial Assets	140
	Strategy Principle 3.3: Commercial Demonstration	141

Strategy Principle 3.4: Endurance for Building a Platform Technology	142
Strategy Principle 3.5: Conservative Capital Structure	142
13 An Execution Strategy	145
13.1 Becoming a Viable Emerging Technology	145
Milestones for an Emerging Bio-inspired Computing Technology Validation	146
13.2 Phase 1: Conceptual Proof of Concepts	147
Defining Success in Phase 1	148
13.3 Phase 2: Outperforming One Best-in-Class Computing Method	149
Defining Success in Phase 2	150
13.4 Phase 3: Solving Previously Unsolved Problems	151
Defining Success in Phase 3	152
13.5 Phase 4: Zero to One (Reaching the Deep-Tech Escape Velocity)	153
Defining Success in Phase 4	154
13.6 Phase 5+: Long-Term Focus on Building the Stack	155
Architectures for Rapid AI Learning	156
Edge Architecture	157
13.7 Contingency Plan: Value Creation without Meeting All Milestones	158
 Appendix	 163
A Structure of the Computing Market	163
A.1 Outline and Recommendations to the Reader	163
A.2 Computing Market Segmented using the Abstraction Layers	163
Physical Layer (Hardware)	164
Microarchitecture	165
Instruction Set Architecture (ISA)	167
Operating System Layer	168
Middleware Layer	170
Application Software Layer	172
A.3 A Deeper Dive into Instruction Set Architecture	174
Historical Development of ISA	174
Specialized ISAs for AI and Machine Learning	176
Business Models and Protection Strategies for New ISAs	178
Summary of ISA	178
Additional material: ISA examples of RISC-V and ARM	179

B	Barriers to Entry for Emerging Computing Technology	183
B.1	Physical Layer (Hardware)	183
B.2	Microarchitecture	184
B.3	Instruction Set Architecture (ISA)	186
B.4	Middleware Layer	187
B.5	Summary of Barriers to Entry	188

Introduction

“Any sufficiently advanced technology is indistinguishable from magic” – Arthur C. Clarke

This document serves as a vision document, and a comprehensive manifesto, outlining the **vision for a new bio-inspired computing** paradigm. It explores two related opportunities: the broader domain of bio-inspired computing—a vibrant field of research and development encompassing everything from neuromorphic hardware to bio-inspired software—and the specific approach we are developing, which we refer to as **Bio-Arch™ Computing**. Beyond presenting an aspirational vision for a disruptive technology, it provides a detailed summary of the conceptual framework and a tactical roadmap for execution, from a humble project lead by a few early innovators, aspiring to become a new movement in an emerging technology.

One of the key aims of the document is to offer potential supporters and partners, whether technical or non-technical, an integrated and holistic understanding of both the **theoretical underpinnings** and the **practical strategy** required to bring this paradigm to fruition. By consolidating these views into a single document, I aim to inform, gauge interest and curiosity, foster collaboration, seek feedback, and track progress against defined milestones and towards the realization of a long-term, complex, high-impact vision.

Nature points to an extraordinary path

Observations of biological computation form the foundation of this initiative. The human brain is an extraordinarily capable supercomputer that operates on the energy of a single light bulb. Moreover, it can learn from remarkably few samples, a feat unmatched by any current model of learning. Nearly all superhuman achievements in artificial learning rely on vastly larger datasets, and comparisons do not control for energy and data. This disparity suggests that the physical limits of computational power and efficiency far exceed our current systems, hinting at an extraordinary opportunity to create bio-inspired computers capable of currently unattainable tasks with relatively modest resources.

This initiative assumes that if nature can achieve such remarkable capabilities, then reverse engineering them is possible. Imagine, for the sake of the argument, the impact of computers

that are millions of times more capable than current computers given the same amount of resources (power, material, data, and time).

At their limit, such advancements would enable:

- ◆ **Supercomputing without supercomputers:** Transform edge computing by enabling local supercomputing processing without reliance on third-party supercomputing in data centres.
- ◆ **Altering Economics of Computing:** Dominate alternative computation methods due to dramatically lower resource needs (power, material, and data) within areas of application. If costs become orders of magnitude lower than the competition, this alters the economics of the market within those sectors.
- ◆ **New Applications and Markets:** Enable entirely new forms of inference with limited data and solving currently unsolvable computational problems, leading to completely new markets for new applications.

Although a specific approach is guaranteed to achieve every desired outcome, nature itself proves that these possibilities are real and that the laws of physics favor our efforts. The extraordinary potential is already demonstrated by nature, making this pursuit not only intriguing but also essential for unlocking a new era of computational innovation.

Realizing this technology demands a significant alignment of effort. A sustained multidisciplinary research and development effort, a credible plan to bring its benefits to the market through useful products, and the commitment of exceptional talent to a long-term vision. Turning this vision into reality requires an organized initiative that not only executes on the necessary technical innovation, but also addresses how to sustain long-term progress within today's economic realities and markets. In other words, how can this technology create value for its users at its stages of development? This document, therefore, serves not just as a description of an exciting technology, but as a blueprint for making it a reality.

Distinction with other bio-inspired computing paradigms

Rooted in Algebraic Structures

The role of algebraic structures that computers are optimized for is widely understood (but often not discussed in terms of algebraic structures). For instance, the difference between a GPU and a CPU can be seen as each architecture being tailored for different algebraic operations; matrix computations for GPUs, and sequential computations for CPU. In essence, these hardware distinctions reflect variations in the specific algebraic operations each system can perform.

Another example is the attention mechanism in transformers, which introduces a dynamic,

context-sensitive algebraic capacity, replacing rigid, static transformations with flexible, adaptable ones. This innovation can be viewed as enabling a novel algebraic capability in computational learning systems, one that almost certainly exists in the brain.

The key insight and motivation is that **computational architectures exhibit algebraic properties** that determine their power, expressivity, efficiency, and effectiveness. The driving question behind this initiative is to identify algebraic structures the brain leverages for general computation and learning. Investigating this leads to architectures that mimic biology's underlying algebraic constructs, rather than its physical configurations. In this sense, the work is deeply mathematical, yet it also demands knowledge of both computing and biological computation. In summary, since the effectiveness and performance of computational design are related to algebraic structures, benefiting from bio-inspired structures can help with computational design.

Emphasis on Filtering irrelevance

I hypothesize that the brain's unparalleled efficiency is fundamentally tied to its **ability to filter irrelevance** in ways that are not yet fully acknowledged or utilized in computer architecture (software and hardware). My approach focuses on **recreating the mathematical conditions** necessary to replicate this phenomenon in computer architecture, and I refer to this approach as **Bio-Arch™ Computing**. Through this lens, I have developed **proof-of-concepts** on computational problems widely regarded as being **at the very edge of possible computational capabilities**, benchmarks that are regarded to be some of the hardest computational problems known to computer science. We believe that these early demonstrations are early signs that we are on the right path to create significant value in future applications.

The **key difference** from other bio-inspired computing paradigms is our focus on recreating the mathematical constraints that enable biological computing to be efficient, specifically their ability to filter irrelevance. For instance, neuromorphic computing, an other bio-inspired approach, emphasizes mimicking the brain's physical architectures. In contrast, we concentrate on the underlying mathematical properties behind efficiency, properties that can be achieved using entirely different hardware platforms. Similarly, in bio-inspired software, while deep learning relies on neuron-like architectures to enable learning, our approach aims to recreate the **mathematical properties of an efficient learning system**, independent of neuron-like designs or specific material forms.

To better illustrate this point with an analogy, consider the physical form of a modern computer, which is heavily based on logic implemented by *transistors* (a small electrical device). It is notable that a transistor - *the material structure* - is the physical component used to implement a computer's logical operations, logical operations - *the mathematical struc-*

ture - can be implemented using other materials. For example, early computers used vacuum tubes or punch cards. While the choice of material is important and some materials may be more suitable than others, its relevance lies in its ability to implement the mathematics of computer logic. Similarly, **our approach is rooted in deep mathematical insights into what makes a computational system effective despite constrained resources**, rather than relying on the specific physical forms in which these principles manifest. Our methodology is based on a mathematical framework inspired by nature's implementation of learning and computation, drawing far more from **cognitive neuroscience** than from cellular neuroscience and material structures of neurons.

How to assess early signs of promise

At the inception of any transformative technology, there is a transition period during which the technology's true value is not immediately apparent. For instance, developing innovative computer hardware can take significant time from design to product, and in non-computing fields, novel medications require significant testing before delivering value in healthcare. Therefore, it is natural to rely on early signs and proofs-of-concept to assess the potential of these deep technologies.

Our current proofs-of-concept showcase our approach's effectiveness in solving notoriously challenging classes of **computational search problems** and **pattern recognition**. We believe these early successes have potentially far-reaching implications for future commercial software applications and computer architecture. Given the technical complexity of our proofs-of-concept and their nontrivial link to value creation, this vision paper presents our rationale in plain language, without excessive technical detail. The specifics of these proofs of concepts are discussed in Chapter 6, but I lay out the basic considerations below in the context of the development of our technology.

What Makes a Strong yet Early Proof-of-Concept?

Imagine stepping back to the early days of transformative computer architectures that changed the world, before their commercial applications were not yet obvious to everyone. How could we identify their potential early? For example, what mindset did you adopt to identify the true potential in deep learning, NVIDIA's CUDA, and Arm's CPU architecture? Which criteria proved most crucial? I argue that in all cases, these were two types of early indicators of tremendous potential:

1. **Extraordinary Computational Results** (even within a narrow domains): Achieve unmatched performance on challenging computational problems, even if initially con-

fined to a narrow or noncommercial scope.

2. **Generalizability:** compelling evidence that the same or similar methods can be applied across a wide range of applications in the future, even if they are not applicable right away.

For example, deep learning, NVIDIA's CUDA, and RISC architecture (Arm) each demonstrated significantly superior performance in specific, often initially noncommercial, areas. But equally important, there was compelling evidence that their potential could be extended to other domains. For instance, the 2014 breakthrough of deep learning in image recognition was remarkable not only for its benchmark performance but also for the clear path it provided to generalize these benefits. In contrast, IBM's Deep Blue, which defeated the world chess champion in 1998, failed to yield broader benefits because its methodology was not generalizable.

In our case, we are committed not only to achieving extraordinary computational results, orders of magnitude better than alternative methods, but also to ensuring that our approach is generalizable, thereby delivering tangible value. Additional specifics are included in Chapter 6.

The case for a long-term focus to maximize value

Major innovations in computer architecture have always required a sustained, long-term research and development effort before reaching their value creation stage. For example, NVIDIA's Cuda that powers most AI software operated at significant net losses for over a decade and was initially mostly used in academic settings. Similarly, it took a few years for deep learning from achieving remarkable results on academic benchmarks to find its way into commercial applications.

Sustained effort is structurally essential in computer architecture because it requires compatibility with many layers of the technology stack; from hardware physics and microarchitecture to basic instruction sets, before useful software can be built on top. While incremental improvements within existing stacks yield short-term gains, major innovations that depart from established technology stacks require time to develop the ecosystem necessary to maximize their benefits. This means that the realization of the full value of a technology requires a long-term development effort.