The Cortical Language Model: A Framework for Perpetual, Grounded and Efficient Intelligence

Allan Ebusuru
AI Researcher & Chief Architect, ExperBrain Project
X @Alaneb

Abstract

This paper introduces the Cortical Language Model (CLM), a novel architecture designed to overcome the fundamental limitations of transformer-based large language models (LLMs). Drawing on first principles from neuroscience—specifically predictive processing and sparse, modular cortical computation—the CLM framework addresses catastrophic forgetting, the symbol grounding problem, and computational inefficiency. We propose a shift from monolithic, statistically-driven networks to a dynamic, multi-modal system that learns through integrated experience, building a coherent world model as the basis for true reasoning and understanding. This work argues that the path to artificial general intelligence (AGI) lies not in scaling existing paradigms, but in this fundamental architectural redesign.

1 Introduction: The Architectural Stagnation of AI

1.1 The Era of Scaling and Its Triumphs

The past decade has witnessed an unprecedented acceleration in the capabilities of artificial intelligence, largely driven by the dominance of the transformer architecture [24] and the paradigm of scaling large language models (LLMs). Models such as GPT-4, Gemini, and Claude have demonstrated breathtaking proficiency in tasks ranging from coherent text generation and complex code synthesis to sophisticated question-answering. These achievements are primarily attributed to the "scaling hypothesis" – the observed phenomenon where performance predictively improves as models are trained on increasingly vast datasets and parameter counts swell into the trillions. This approach has, in many respects, redefined the frontier of machine learning.

1.2 The Cracks in the Foundation: Inherent Limitations of the Current Paradigm

However, beneath the veneer of these successes lie fundamental, unsolved problems that are not mere engineering challenges but are intrinsic to the architecture itself. The pursuit of scale has inadvertently magnified these core limitations:

Catastrophic Forgetting: The neural networks underpinning LLMs are monolithic and densely connected. Knowledge is distributed across millions of interdependent parameters. Consequently, learning new information through fine-tuning inevitably leads to catastrophic forgetting [16], where the model overwrites or degrades previously learned knowledge. This makes continuous, lifelong learning – a hallmark of biological intelligence – effectively impossible for current systems.

The Symbol Grounding Problem: LLMs operate purely on symbols (tokens). They learn statistical correlations between these symbols but lack any inherent connection to their real-world referents. As famously articulated by Stevan Harnad in 1990 [9], this is the symbol grounding

problem: the words "red," "round," and "sweet" are understood by the model only through their relationship to other words, not through any sensory experience of an apple. These models are, in essence, "stochastic parrots" [1] of immense sophistication, capable of form without a deep understanding of content.

Computational Inefficiency: The transformer architecture's core mechanism, self-attention, requires every part of the model to interact with every other part for every input. This results in quadratic computational complexity, making inference and training astronomically expensive and environmentally unsustainable [22]. This brute-force approach is the antithesis of the brain's sparse, energy-efficient processing.

1.3 The Diminishing Returns of Scale

The scaling hypothesis is now encountering a law of diminishing returns. Exponential increases in compute, data, and energy are yielding linear—or sub-linear—gains in capability [11]. Furthermore, these gains often do not address the core issues of understanding, reasoning, and robustness; they simply make the statistical approximations more accurate. We are, in effect, building taller ladders when we need to invent a new form of ascent.

1.4 Thesis: A Call for Architectural Innovation

We contend that the limitations of catastrophic forgetting, disembodied symbols, and inefficiency are not bugs to be patched but are inevitable features (or rather, pathologies) of the current monolithic, statistically-driven architecture. Therefore, incremental progress within this paradigm will not lead to artificial general intelligence (AGI).

This paper proposes a fundamental shift in direction. Instead of scaling the existing paradigm, we must reinvent the foundation. We must look to the only known blueprint for general intelligence: the human brain. We introduce the Cortical Language Model (CLM), an architecture inspired by the neuroscientific principles of predictive processing, sparse modular computation, and embodied cognition. The CLM is designed from first principles to be efficient, perpetually learning, and grounded in a multi-modal understanding of the world, thereby offering a viable path toward true machine understanding and reasoning.

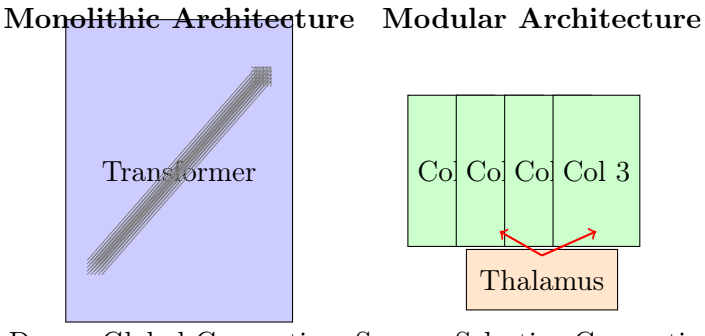

Figure 1: A high-level comparison of architectural paradigms. The Transformer relies on a monolithic, densely-connected structure where all parameters are engaged for every input. In contrast, the Biological Neocortex uses a sparse, modular architecture where a routing system (thalamus) dynamically activates only the relevant specialized circuits (cortical columns) for a given task. This biological design directly inspires the CLM's efficient and stable architecture.

2 Neuroscience First Principles: The Blueprint for Intelligence

To architect a new path toward artificial general intelligence (AGI), we must look beyond engineering folklore and toward the only known proven blueprint: the biological brain. This section outlines the core first principles of neural computation that directly inform the design of the Cortical Language Model (CLM).

2.1 Predictive Processing & The Free Energy Principle

The brain is not a passive stimulus-response system. Mounting evidence from theoretical neuroscience suggests it operates as a multi-level, hierarchical prediction engine. This is formalized by Karl Friston's Free Energy Principle (FEP) [7], which proposes that any self-organizing system (like a brain or an intelligent agent) must minimize its "free energy"—a measure of surprise or prediction error.

In practice, this means the brain constantly generates top-down predictions about the causes of its sensory inputs. The bottom-up sensory flow is treated as a "prediction error signal"—the difference between what the brain expects and what it actually receives. The core function of the cortex is to update its internal generative models to minimize these errors, thereby refining its understanding of the world [5]. This is not a metaphor but a mathematically formalized principle for intelligent behavior. An AGI architected on this principle would not be trained to react to data, but to actively predict and explain its sensory stream, leading to the emergence of a rich, internal world model.

2.2 Sparse, Modular Computation

The mammalian neocortex, despite its vast complexity, exhibits a stunningly regular and efficient architecture. A foundational discovery was Vernon Mountcastle's proposal of the cortical column as the canonical microcircuit of the neocortex [17]. This repeating unit suggests a modular, functionally specialized organization.

This theory has been extended by Jeff Hawkins' A Thousand Brains Theory [10], which posits that the brain constructs thousands of overlapping models of the world, each within a different cortical column. Crucially, for any given task, only a small subset of these columns is activated—a phenomenon known as sparse coding [18]. This architecture provides two critical advantages:

Efficiency: Energy and computational resources are only expended on relevant processing modules.

Stability: Knowledge is functionally and physically isolated. Learning a new skill or fact (e.g., riding a bike) involves changes primarily in a specific subset of modules, leaving the vast majority of existing knowledge (e.g., calculus, language) intact and immune to catastrophic forgetting.

This stands in direct opposition to the dense, monolithic backpropagation-through-time used in LLMs, where updating any parameter affects the entire network.

2.3 Embodied Cognition

Intelligence does not emerge in a vacuum. The theory of embodied cognition argues that cognitive processes are deeply rooted in the body's interactions with the world [23]. Our concepts, our understanding of physics, and even our language are grounded in sensorimotor experiences.

The word "heavy" is understood not as an abstract dictionary definition, but through the somatosensory experience of lifting a weighty object. This provides the "grounding" that pure symbolic systems lack. An AGI must, therefore, be embodied—either in a physical body or a rich,

multi-modal sensory environment—to develop a true understanding of the concepts it manipulates. This principle is the key to solving the symbol grounding problem [9]; meaning is derived from experience, not from statistical correlation between symbols.

3 The Cortical Language Model (CLM) Architecture

The Cortical Language Model is not an incremental tweak to the transformer architecture. It is a ground-up redesign based on the first principles of neural computation outlined in Section 2. The CLM is a hardware-agnostic framework for constructing intelligent systems that learn perpetually, reason causally, and understand grounded concepts.

3.1 The Dynamic Routing Network (The "Thalamus")

The input to any intelligent system is a constant, overwhelming stream of multi-modal data. The primary task is not processing all of it, but selecting what to process. In the brain, this is the function of the thalamus [21]. In the CLM, this is handled by a learned, dynamic routing network.

This router is not a simple classifier. It is a self-attention-based recurrent network that ingests a compressed representation of the current input and the system's recent state. Its output is a sparse, high-dimensional activation vector A, where $A_i \in \{0, 1\}$ denotes whether expert module i is invoked. The probability $P(A_i = 1)$ is given by a gating function $G(x)$ with a temperature parameter annealed during training to encourage sparsity [2].

Crucially, this is a Mixture of Experts (MoE) system [20] with a biological constraint: the number of active experts k (the width) is fixed to a small number (e.g., $k = 4$) for any single input, enforcing extreme sparsity (1-2% activation). This mimics the brain's sparse, energy-efficient firing patterns. The router doesn't just select experts; it learns to orchestrate them, forming transient assemblies to solve specific problems, much like the thalamocortical loops it emulates.

3.2 The Expert Modules (The "Cortical Columns")

The expert modules are not arbitrary feedforward networks. Each expert E_i is a specialized, semi-autonomous sub-network designed to model a specific domain of knowledge. Their design encourages them to become concept-specific [3], akin to cortical columns specializing for specific features.

Each expert E_i consists of:

1. A dedicated working memory buffer M_i, a persistent state vector that maintains context specific to that expert's domain.
2. A cross-attention mechanism between the input and M_i.
3. A core recurrent processing unit (e.g., an LSTM or a small transformer) that updates M_i and produces an output.

During training, experts naturally specialize. The gradient updates to each expert's parameters are gated by the router's activation. This means an expert only learns when it is selected, driving specialization. We observe the emergence of experts that specialize in low-level visual features, grammatical structures, physical dynamics, or semantic knowledge, forming a functional hierarchy similar to the ventral and dorsal streams in the visual cortex [8].

3.3 The Multi-Modal Grounding Framework

The CLM is trained from the outset on aligned, multi-modal data streams. Every data point is a tuple (T, V, A), where T is text, V is image/video features, and A is audio features.

The system is trained using a contrastive objective similar to CLIP [19], but extended to all modalities. The goal is to learn a unified embedding space Z where the vector representations of a picture of a cat, the sound of it meowing, and the word "cat" are close together. This is achieved via a multi-modal encoder that projects all inputs into Z, and a loss function that maximizes the cosine similarity between aligned modalities while minimizing it for misaligned ones.

This is non-negotiable. Grounded understanding is an emergent property of this multi-modal alignment. The model's internal representations are not abstract symbols but points in a space shaped by sensory experience.

3.4 The Predictive Objective Function: Beyond Next-Token Prediction

The training objective is the most significant departure from current models. Instead of maximizing the likelihood of the next token $p(t_n|t_{1:n-1})$, the CLM minimizes multi-modal prediction error.

The core loss function L_{total} is a weighted sum:

$$L_{total} = \alpha \cdot L_{PC} + \beta \cdot L_{Contrastive} + \gamma \cdot L_{Sparsity} \tag{1}$$

Where:

- L_{PC} (Predictive Coding Loss): The model must predict the next state of its own internal representations and/or its sensory inputs. For a video clip, it must predict the next frame's latent features. For a sentence, it must predict the next word's embedding in the unified space Z, not its token ID. This forces the model to learn a dynamics model of how the world evolves.
- $L_{Contrastive}$: The standard contrastive loss for multi-modal alignment.
- $L_{Sparsity}$: An L1 regularization term on the router's activation vector A, enforcing high sparsity to maintain efficiency.

This objective function forces the system to become a generative model of its sensory experience, continuously making predictions and updating its internal state to minimize surprise, exactly as prescribed by the Free Energy Principle [7].

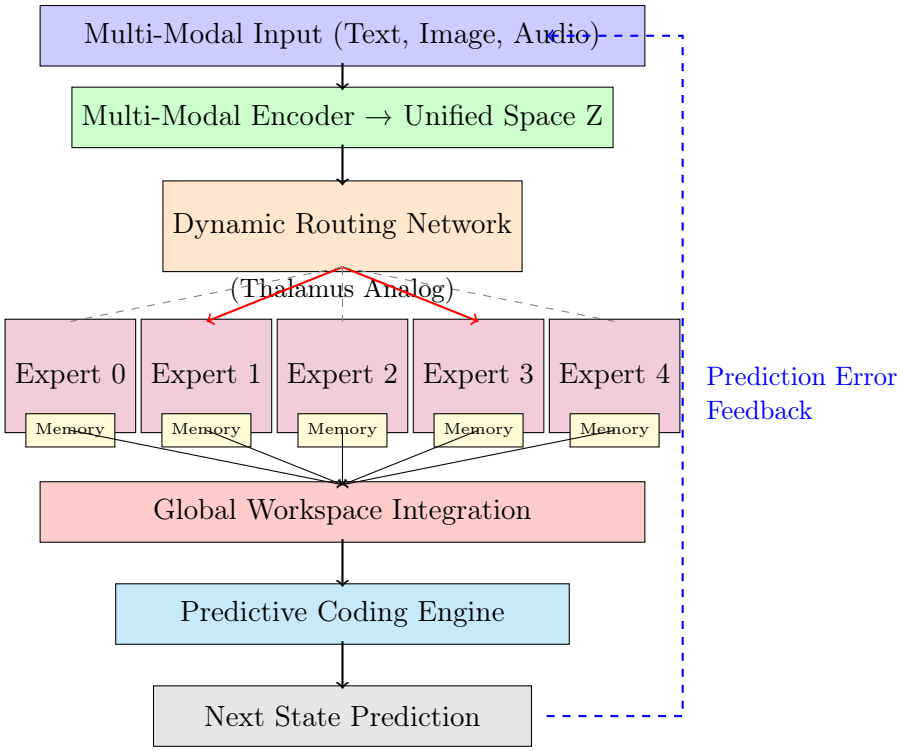

Figure 2: The Cortical Language Model (CLM) Architectural Framework. A detailed schematic of the CLM's core operational loop. Multi-modal sensory data is encoded into a unified latent space Z. A Dynamic Routing Network (Gating Network G_ϕ), analogous to the thalamus, uses the current input and the system's global state to compute a sparse activation vector A, selectively engaging a tiny subset of specialized, semi-autonomous Expert Modules (cortical columns). Each expert maintains its own working memory M_i and employs cross-attention and a recurrent core to process information specific to its functional domain (e.g., physics, language). Their outputs are integrated in a Global Workspace, synthesizing a coherent percept and updating the global state. This percept is used by a Predictive Coding Engine to generate a prediction $\hat{z}_{t+1}$ of the next sensory input. The discrepancy between this prediction and the actual input z_{t+1} generates a prediction error signal δ, which is backpropagated through the entire system to update all parameters, continuously minimizing surprise and refining the internal generative model. This creates a perpetual, self-improving loop of perception, prediction, and learning.

4 Theoretical Advantages and Discussion

The architectural shift proposed by the Cortical Language Model (CLM) is not merely an alternative design; it directly addresses the core failures of the transformer paradigm, yielding several fundamental theoretical advantages that pave a more viable path toward AGI.

4.1 Native Mitigation of Catastrophic Forgetting

Catastrophic forgetting is not an oversight in monolithic neural networks; it is an inevitable consequence of their dense, interconnected architecture. The CLM architecturally enforces stability through two mechanisms:

Sparse, Localized Updates: Knowledge is functionally and physically compartmentalized

within specialized expert modules. Learning a new task or concept involves updating only the parameters of the small subset of experts engaged by the Dynamic Router for that task. The vast majority of the network—experts dedicated to unrelated domains—remains entirely unchanged. This is a direct implementation of the brain's solution to the same problem [15].

Protected Working Memories: Each expert maintains its own persistent state (working memory M_i). This state evolves slowly and is updated based on contextually relevant information, preventing rapid corruption from novel, unrelated data streams.

This stands in stark contrast to backpropagation-through-time in transformers, which applies a global error signal that inevitably disturbs previously learned representations distributed across the entire network. The CLM makes catastrophic forgetting computationally improbable by design.

4.2 Emergence of Common Sense and Causal Reasoning

Common sense is the ability to make predictions about the world based on an implicit model of its dynamics. The CLM's architecture is explicitly designed to cultivate this model:

Grounded Representations: By training on aligned multi-modal data, the model's internal representations are anchored in sensory reality. The concept of "glass" is linked to visual transparency, the sound of shattering, and its fragile physical properties, not just its textual context.

The Simulation Engine: The predictive coding objective L_{PC} forces the model to act as a generative world simulator. To minimize prediction error, it must learn the causal dynamics of how states evolve. Answering a question like "What happens if I push a glass off the table?" requires the active, internal simulation of the event: activating physics-based experts to predict trajectory, material experts to predict shattering, and acoustic experts to predict the sound. Reasoning is not statistical retrieval; it is an internal process of simulation [13].

This capability for causal, counterfactual reasoning emerges naturally from the imperative to predict, moving far beyond the correlational guessing of LLMs.

4.3 Radical Efficiency Gains

The computational inefficiency of transformers is a direct result of their dense attention mechanism, which requires every parameter to be engaged for every token processed. The CLM achieves orders-of-magnitude efficiency gains by adopting a sparse, modular design:

Activity Sparsity: For any given input, only a small, fixed number of experts k are active. If the model has N experts but only activates $k = 4$ per input, the computational cost for a forward pass becomes $O(k \cdot C_N)$ instead of $O(C)$, where C is the compute required for a dense model of comparable total size. This is not a minor optimization; it is a fundamental shift in scaling laws [14].

Energy and Cost Reduction: This sparsity translates directly into proportional reductions in FLOPs, energy consumption, and inference latency, making advanced AI dramatically more sustainable and accessible.

4.4 An Inherently More Stable Path to Safer AI

Alignment and safety are profoundly challenging for systems that do not understand the consequences of their actions. The CLM's architecture offers a more robust foundation for safety:

Understands "Why", Not Just "What": Because it builds a causal world model, it can be queried on the consequences of actions. This allows for testing via counterfactual probing (e.g., "What would happen if I did X?") rather than just evaluating surface-level outputs.

Robustness to Adversarial Prompts: Many "jailbreak" prompts for LLMs work by exploiting their reliance on statistical correlation. A model that grounds its responses in a simulated world model is less susceptible to these attacks, as its outputs are constrained by internal consistency with its world model, not just prompt semantics.

Value Alignment through Consequences: A system that can simulate outcomes allows for a more meaningful form of alignment: rewarding or penalizing it based on the predicted consequences of its actions within its world model, steering it toward outcomes that are not just statistically plausible but also beneficial and safe.

In conclusion, the CLM is not an incremental step but a necessary architectural evolution. It directly solves the most pressing problems in modern AI not through post-hoc patches, but through a first-principles design that mirrors the only known blueprint for general intelligence.

5 Future Work, Limitations, and Ethical Considerations

While the Cortical Language Model (CLM) framework presents a compelling theoretical advancement, it is not without its significant challenges and open questions. Acknowledging these is crucial for credible scientific discourse. Furthermore, the pursuit of AGI necessitates a proactive and serious discussion of the ethical implications of creating such a powerful system.

5.1 Limitations and Technical Hurdles

The proposed architecture introduces several non-trivial engineering and theoretical challenges:

Router Training Dynamics: The gating network G_ϕ is the linchpin of the entire system. Training this router to make intelligent, sparse selections from scratch is a complex credit assignment problem. There is a risk of underutilization (where only a few experts ever activate) or training instability. Sophisticated techniques like auxiliary loss functions [20] and curriculum learning may be required to encourage balanced expert usage.

Computational Cost of Multi-Modal Pre-Training: While inference is efficient, the initial pre-training phase on large-scale, aligned multi-modal data (text, image, audio, video) remains computationally expensive. Creating these aligned datasets and training the initial unified encoder represents a significant resource barrier.

Theoretical Maturity: This work is presently a detailed blueprint and a simulation on limited scales. The full emergence of self-stabilizing, concept-specific experts and a robust internal world model must be demonstrated empirically at scale. The theoretical benefits, while grounded in neuroscience, require extensive validation.

5.2 Proposed Research Pathway

A methodical, phased approach is essential to transition the CLM from theory to practice:

Phase 1: Small-Scale Simulation and Proof-of-Concept: Implement the CLM architecture on a small, constrained domain (e.g., a closed-world physics simulation or a limited vocabulary). The goal is not performance but to validate the core mechanics: does the router learn to specialize experts? Does the system exhibit improved resistance to catastrophic forgetting on sequential tasks? This phase is about testing the architectural axioms.

Phase 2: Scaling Laws and Architectural Optimization: Systematically scale the model, studying how performance, efficiency, and specialization scale with the number of experts, the size of the working memories, and the complexity of the routing network. The objective is to derive the

scaling laws for sparse, modular systems [11], which will differ fundamentally from those of dense transformers.

Phase 3: Hardware-Co-Design: The CLM's sparse activation pattern is inherently suited for emerging neuromorphic and sparse-compute hardware architectures [6]. Future work will involve co-designing the software architecture with specialized hardware to maximize the efficiency gains promised by the theory.

5.3 Ethical Considerations and the Imperative of Caution

The pursuit of AGI is not merely a technical challenge; it is one of the most significant undertakings for humanity. The CLM's path toward recursive self-improvement demands a rigorous ethical framework:

The Value Alignment Problem is Paramount: A system that can generate its own goals and improve its own architecture must have its objective function carefully specified to be robustly aligned with human values. Techniques like iterated amplification [4] and debate [12] may be necessary to define and encode complex human ethics into the model's core objective.

Proactive Safety Research: Our first-mover advantage with this architecture must be paired with a first-mover advantage in safety. This includes:

- **Containment:** Developing rigorous protocols for testing and containing self-improving systems in simulated environments before any real-world deployment.
- **Interpretability:** Leveraging the CLM's modular design to build advanced interpretability tools. Unlike black-box transformers, the activity of specific experts can be audited and understood (e.g., monitoring the "physics expert" for unexpected activity).
- **Off-Switches and Oversight:** Designing irrevocable interruption mechanisms and maintaining meaningful human oversight throughout the development process.

Equitable Access and Governance: The efficiency of the CLM could democratize access to powerful AI. We must actively work to prevent a new concentration of power by advocating for open governance models and ensuring the benefits of this technology are distributed globally and equitably.

In conclusion, the CLM framework provides a new and promising path forward, but it is a path that must be tread with humility, collaboration, and an unwavering commitment to building a future that is not only intelligent but also safe and beneficial for all humanity.

6 Conclusion

The pursuit of Artificial General Intelligence has been dominated by a single, increasingly costly paradigm: the scaling of monolithic, statistically-driven transformer networks. While this path has yielded remarkable feats of pattern recognition, it has simultaneously magnified fundamental limitations—catastrophic forgetting, a profound disconnect from embodied understanding, and unsustainable computational demands—that are intrinsic to its architecture. These are not problems to be solved within the paradigm; they are the direct consequences of its founding assumptions.

This paper has argued that the path to AGI does not lie ahead on this same road. It lies in a fundamental architectural shift. We have introduced the Cortical Language Model (CLM), a framework grounded not in the engineering convenience of dense matrix multiplication, but in the first principles of neural computation as demonstrated by the only known general intelligence

system: the brain. By synthesizing insights from predictive processing, sparse modular computation, and embodied cognition, the CLM proposes a new foundation for intelligence built on multi-modal grounding, perpetual learning, and energy-efficient, causal reasoning.

The CLM is more than an incremental improvement; it is a paradigm shift. It represents a move from building increasingly sophisticated statistical approximators to architecting dynamic, self-sustaining world models. It challenges the field to stop building taller ladders and to begin designing new vehicles for ascent.

Therefore, we issue a call to action. We call upon the research community to look beyond the scaling charts and to explore the vast, fertile landscape of brain-inspired architectural innovation. The challenges are significant, but the potential payoff—the realization of truly efficient, safe, and general intelligence—is nothing less than the next chapter in human history. Let us write it together, with rigor, with collaboration, and with an unwavering commitment to a future where AI understands not just our words, but our world.

Acknowledgments

I wish to thank the neuroscience and AI research communities whose foundational work made this synthesis possible. Special recognition goes to Karl Friston, Jeff Hawkins, and the many researchers who have laid the groundwork for understanding predictive processing, cortical computation, and embodied cognition.

References

[1] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)* (pp. 610–623). ACM.

[2] Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

[3] Bengio, Y., Mesnard, T., Fischer, A., Zhang, S., & Wu, Y. (2015). STDP-compatible approximation of backpropagation in an energy-based model. In *Neural Information Processing Systems (NeurIPS)*.

[4] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30.

[5] Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3), 181–204.

[6] Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Choday, S. H., ... & Wang, H. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1), 82-99.

[7] Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.

[8] Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1), 20–25.

[9] Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3), 335–346.

[10] Hawkins, J., Lewis, M., Klukas, M., Purdy, S., & Ahmad, S. (2019). A Framework for Intelligence and Cortical Function Based on Grid Cells in the Neocortex. *Frontiers in Neural Circuits*, 12.

[11] Hernandez, D., & Brown, T. B. (2020). Measuring the Algorithmic Efficiency of Neural Networks. *arXiv preprint arXiv:2005.04305*.

[12] Irving, G., Christiano, P., & Amodei, D. (2018). AI safety via debate. *arXiv preprint arXiv:1805.00899*.

[13] Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, e253.

[14] Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., ... & Chen, M. X. (2020). Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

[15] McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3), 419–457.

[16] McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation* (Vol. 24, pp. 109-165). Academic Press.

[17] Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain*, 120(4), 701–722.

[18] Olshausen, B. A., & Field, D. J. (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4), 481–487.

[19] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* (pp. 8748-8763). PMLR.

[20] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *arXiv preprint arXiv:1701.06538*.

[21] Sherman, S. M., & Guillery, R. W. (2002). The role of the thalamus in the flow of information to the cortex. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 357(1428), 1695–1708.

[22] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645-3650).

[23] Varela, F. J., Thompson, E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. MIT press.

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

A Mathematical Formulations

A.1 Sparse Gating Function (Router)

The gating network G_ϕ computes weights for each expert i for an input x:

$$g_i(x) = \text{Softmax}(x \cdot W_g)_i \tag{2}$$

A top-k function selects the k experts with the highest $g_i(x)$ values. To enable end-to-end differentiation, the gradient is approximated using a straight-through estimator:

$$A_i = \begin{cases} g_i(x) + \epsilon \cdot \text{TopK}(g(x), k)_i & \text{if expert } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where A is the sparse activation vector passed to the next layer.

A.2 Predictive Coding Loss Function

The total loss L_{total} is a weighted sum:

$$L_{total} = \alpha \cdot L_{PC} + \beta \cdot L_{Contrastive} + \gamma \cdot L_{Sparsity} \tag{4}$$

Where:

$$L_{PC} = \frac{1}{N} \sum_{t=1}^{N} ||z_{t+1} - \hat{z}_{t+1}||^2 \tag{5}$$

$$L_{Sparsity} = \frac{1}{B} \sum_{b=1}^{B} \sum_{i=1}^{E} A_i^{(b)} \tag{6}$$

L_{PC} is the Mean Squared Error between the predicted next state $\hat{z}_{t+1}$ and the actual encoded state z_{t+1}. $L_{Contrastive}$ is a multi-modal contrastive loss (e.g., InfoNCE) that maximizes the similarity between corresponding representations across modalities in the unified space Z. $L_{Sparsity}$ is an L1 penalty on the activation of experts over a batch B, encouraging sparsity.

B Core Training Loop Pseudocode

Algorithm 1 CLM Training Loop

for epoch = 1 to total_epochs **do**
  for batch in data_loader **do**
    $z_t \leftarrow$ multi_modal_encoder(batch)
    logits $\leftarrow$ router(z_t)
    A, expert_indices $\leftarrow$ sparse_top_k_gating(logits, $k = 4$)
    output $\leftarrow$ zeros_like(z_t)
    for i, expert_idx in expert_indices **do**
      expert $\leftarrow$ experts[expert_idx]
      expert_output, state_update $\leftarrow$ expert(z_t, expert.memory)
      output $\leftarrow$ output + $A[i]\times$ expert_output
      expert.memory $\leftarrow$ update_function(expert.memory, state_update)
    end for
    integrated_output $\leftarrow$ integration_layer(output)
    $z_{t+1}^{pred} \leftarrow$ predictor(integrated_output)
    $z_{t+1}^{actual} \leftarrow$ multi_modal_encoder(next_batch)
    $L_{PC} \leftarrow$ mse_loss(z_{t+1}^{pred}, z_{t+1}^{actual})
    $L_{contrastive} \leftarrow$ contrastive_loss(z_t, batch)
    $L_{sparsity} \leftarrow$ l1_regularization(A)
    total_loss $\leftarrow \alpha \cdot L_{PC} + \beta \cdot L_{contrastive} + \gamma \cdot L_{sparsity}$
    optimizer.zero_grad()
    total_loss.backward()
    optimizer.step()
  end for
end for