VAMN: Verifiable Arithmetic Multi-stream Network

Towards Natively Grounded Financial Artificial General Intelligence

A novel neuro-symbolic architecture addressing the fundamental limitations of large language models in high-stakes financial computation through architectural specialization rather than parameter scaling.

The Deterministic Crisis in Stochastic Systems

The Fundamental Problem

The global financial system operates on **absolute mathematical constraints**, not probability distributions. A balance sheet equation (Assets = Liabilities + Equity) is not negotiable—it is a deterministic truth that must hold with 100% precision. Tax provisions are binary: compliant or non-compliant with regulatory code.

Current foundation models like GPT-4 and Llama-3 operate on the "Distributional Hypothesis"—meaning derived from statistical co-occurrence. When an LLM generates "$500," it doesn't calculate; it predicts that this token sequence is statistically likely given the context. This **stochastic parrot behavior** is catastrophic for axiomatic financial reasoning.

Documented Failure Modes

- **Magnitude Confusion:** Models assert $10M > $1B due to token length heuristics rather than numerical understanding
- **Fabricated Regulations:** Hallucination of non-existent tax code clauses with confident citations
- **Arithmetic Decomposition:** Inability to maintain precision across multi-step calculations involving decimal operations
- **Context-Dependent Numeracy:** The same mathematical operation produces different results based on surrounding narrative text

Critical Insight: Financial intelligence is fundamentally incompatible with next-token prediction architectures that treat numbers as discrete linguistic tokens rather than continuous mathematical entities.

The Three Inherent Architectural Defects

Numerical Hallucination

Root Cause: Tokenization converts continuous values into discrete indices, severing the semantic connection between numerical magnitude and representation.

Manifestation: Models generate plausible-looking numbers that are mathematically nonsensical. A depreciation calculation might produce $47,283.19 when the asset base is only $10,000.

Why It Persists: The cross-entropy loss function optimizes for token sequence likelihood, not mathematical correctness. There is no gradient signal penalizing numerically impossible outputs.

Logical Drift

Root Cause: Inability to maintain consistent boolean reasoning over extended context windows, especially when logical premises are separated by hundreds of tokens.

Manifestation: A model might correctly state "Company X operates at a loss" on page 1, then confidently declare "Company X's profitable margins" on page 3 of the same document.

Why It Persists: Attention mechanisms decay over distance. Long-range logical dependencies require explicit symbolic tracking, which transformers lack.

Provenance Opacity

Root Cause: No mechanistic link between generated assertions and their regulatory source documents. Models cannot explain *why* a particular tax treatment applies.

Manifestation: When asked to justify a conclusion, models produce post-hoc rationalizations that sound authoritative but cite non-existent code sections or misinterpret scope.

Why It Persists: Standard language models have a single output stream optimizing for fluency. Legal grounding requires a separate architectural pathway.

Current Mitigation Strategies: A Critical Assessment

Retrieval Augmented Generation (RAG)

Mechanism: Fetches relevant documents from external databases to provide context before generation, reducing hallucination through evidence grounding.

Critical Limitation: RAG provides *context*, not *computation*. Even with perfect document retrieval, the model still must perform numerical reasoning using its flawed token-prediction architecture. A calculator doesn't become better at math by reading more textbooks.

Tool Use (Code Interpreter)

Mechanism: Offloads mathematical operations to external Python interpreters, allowing symbolic computation outside the neural network.

Critical Limitation: The model must correctly formulate the equation to solve. If the reasoning chain contains a logical error—such as selecting the wrong amortization method—the tool will execute that flawed logic with perfect precision, amplifying rather than mitigating the mistake.

Chain-of-Thought Prompting (CoT)

Mechanism: Prompts the model to generate intermediate reasoning steps ("Let's think step-by-step"), making the problem-solving process explicit.

Critical Limitation: Empirical studies reveal "faithful hallucinations"—reasoning chains that appear syntactically correct but are semantically decoupled from arithmetic reality. The steps look convincing but don't actually compute the right answer.

The fundamental problem: **All current approaches attempt to fix a systemic architectural flaw through external Band-Aids rather than redesigning the architecture itself.** No existing foundation model treats numerical magnitude and regulatory law as first-class citizens alongside natural language.

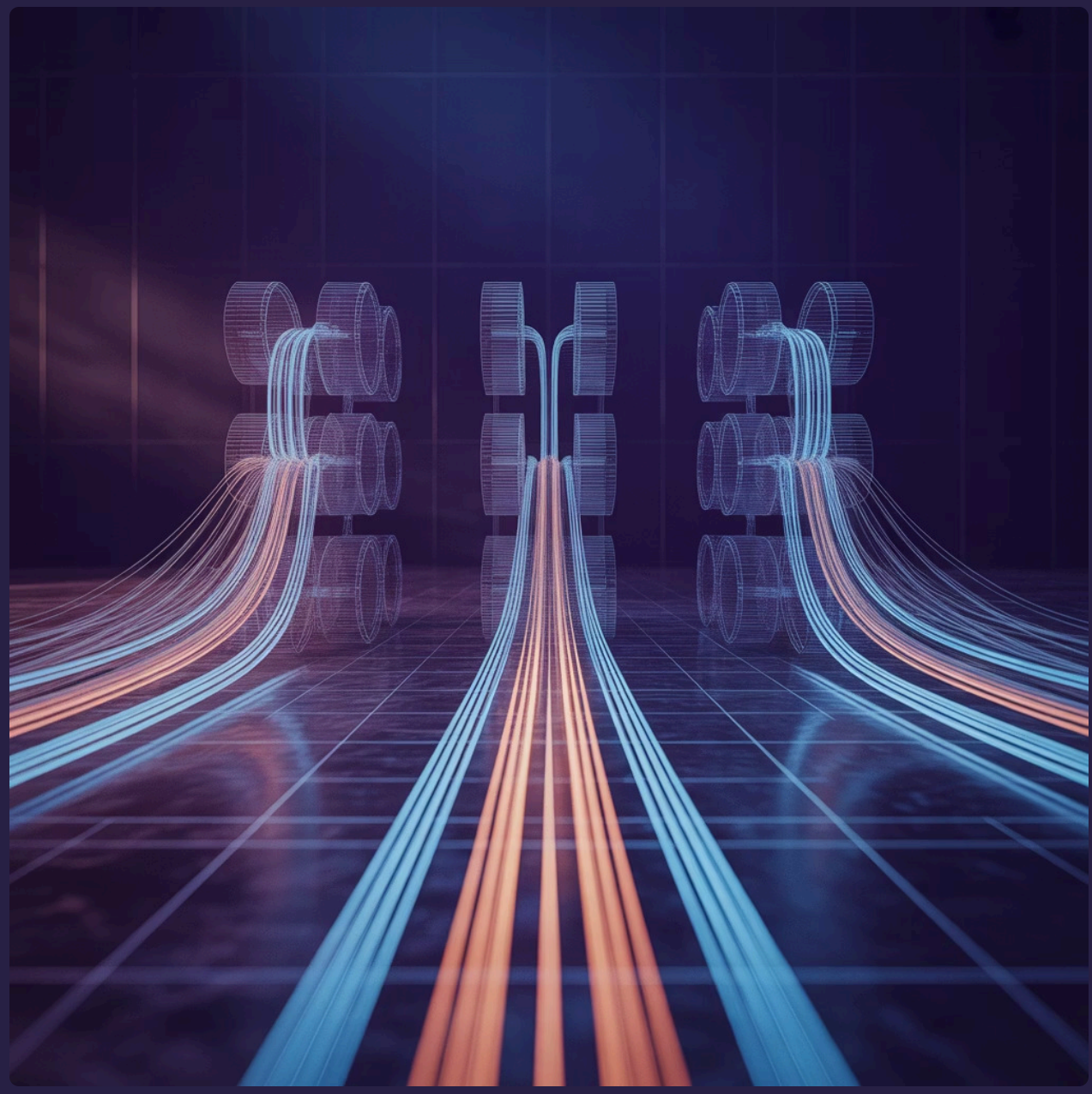
Introducing VAMN: The Triple-Stream Paradigm

Core Hypothesis

Financial intelligence requires **three distinct cognitive processes** that should not be conflated into a single output distribution:

1. **Communication** (Language): Generating fluent, contextually appropriate explanations
(Mathematics): Computing precise numerical values through deterministic operations
3. **Verification** (Law): Anchoring conclusions to specific regulatory provisions

VAMN physically separates these into distinct neural pathways sharing a common encoder backbone, with specialized loss functions and gradient management preventing task interference.



Architectural Philosophy: Rather than asking a single neural pathway to simultaneously optimize for linguistic fluency, mathematical precision, and legal accuracy—objectives that fundamentally conflict—we create dedicated computational streams that can be independently validated and audited.

Stream A: The Semantic Head (Language Generation)

01

Standard Causal Language Modeling

Implements traditional autoregressive generation: $P(w_{t+1} \mid X) = \text{Softmax}(W_{sem} h_t + b_{sem})$, where $W_{sem} \in \mathbb{R}^{|V| \times d}$ projects hidden states to vocabulary logits.

02

Responsibility Scope

This stream handles fluency, syntactic correctness, contextual appropriateness, and qualitative explanation. It generates the narrative wrapper around calculations and citations.

03

Critical Constraint

During training, we apply **semantic constraints** that penalize this stream from generating numerical digit tokens (0-9). This forces architectural dependence on Stream B for all quantitative outputs.

04

Training Objective

Optimized via standard cross-entropy loss L_{CE}, but with a modified vocabulary distribution that assigns near-zero probability mass to numerical tokens, creating a hard separation of concerns.

This architectural constraint is crucial: by making it *impossible* for the semantic head to produce numbers, we eliminate the possibility of hallucinated numerical values at the source. The model literally cannot output "$500" through this pathway—it must route through the regression head.

Stream B: The Quantitative Head (Mathematical Computation)

Regression-Based Number Generation

Rather than treating numbers as discrete tokens, Stream B implements a Multi-Layer Perceptron regressor that outputs **continuous scalar values**:

ŷt+1 = Identity(ReLU(Wquant,2(ReLU(Wquant,1ht))))

where $\hat{y} \in \mathbb{R}$ represents a real-valued prediction.

Innovation: Normalized Value Embeddings

Financial values span extreme ranges (10-2 to 1014). Direct regression in linear space creates optimization instability. We map all targets to **log-space**:

ytarget = sign(x) · log(1 + |x|)

This transformation ensures the model learns **"Arithmetic Embedding Distance"**—where vector distance between $100 and $101 is smaller than between $100 and $10,000, reflecting true numerical proximity.

Sub-Token Anchoring Problem

Standard tokenizers split numbers arbitrarily: "1,000" → ["1", ",", "000"]. We cannot apply regression loss to the comma token.

Solution: The regression loss Lquant is computed only on the *final digit token* of each numerical entity. A preprocessing pipeline identifies these anchor points and aligns continuous ground-truth values to them.

Loss Function: Log-Cosh Regression

We use LLC(y, ŷ) = Σ log(cosh(ŷ - y)) rather than MSE or MAE.

Rationale: Log-Cosh approximates L2 (MSE) for small errors and L1 (MAE) for large errors, providing robustness against outliers inherent in financial datasets while maintaining differentiability.

Stream C: The Citation Head (Legal Grounding)

Stream C anchors every generated assertion to a **fixed ontology of ~50,000 unique regulatory nodes** spanning IFRS, US GAAP, Internal Revenue Code, and other financial standards. This is not retrieval—it's a classification task embedded in the architecture.

1

Hierarchical Softmax Architecture

$$P(c_{t+1} \mid X) = \text{HierarchicalSoftmax}(W_{citeht})$$

The label space C is organized as a tree: Jurisdiction → Act → Section → Paragraph. This reduces inference cost from $O(|C|)$ to $O(\log|C|)$, making real-time citation generation computationally feasible.

2

Training Data Requirements

The Citation Head requires expert-annotated "Chain of Audit" data—financial documents where every assertion is manually linked to its regulatory basis. We propose a 10B token "Golden Corpus" for fine-tuning this pathway.

3

Audit Trail by Design

By forcing the model to explicitly output its citation path, we create an auditable reasoning trace. Regulators can inspect Stream C to accept or reject the model's logic without interpreting the "black box" of attention weights.

The Unified Loss Landscape & Gradient Surgery

Training VAMN requires optimizing three objectives simultaneously, but naively summing losses creates **task conflict**—strong gradients from the high-entropy Semantic head drown out delicate Quantitative regression signals.

PCGrad: Projected Conflicting Gradients

We implement gradient surgery using PCGrad to orthogonalize updates:

$$L_{total} = \lambda_{sem} L_{CE} + \lambda_{quant} L_{LLC} + \lambda_{cite} L_{HS}$$

Where:

- L_{CE}: Cross-Entropy for Semantic Head
- L_{LLC}: Log-Cosh for Quantitative Head
- L_{HS}: Hierarchical Softmax for Citation Head

When gradient vectors for two tasks point in conflicting directions (negative cosine similarity), PCGrad projects one onto the normal plane of the other, ensuring both tasks make progress without mutual interference.

Dynamic Loss Weighting

The coefficients λ are not static. We implement **uncertainty-based weighting** that increases the weight of tasks the model finds difficult during training.

This prevents premature convergence where the model "gives up" on arithmetic precision to optimize the easier language modeling objective.

Engineering Insight: Multi-task learning in neural networks is not just about loss summation—it's about managing competing gradient signals that can destructively interfere. PCGrad ensures each stream receives meaningful learning signals throughout training.

Data Strategy: The Curriculum of Truth

Generating 1.5 trillion tokens of unique synthetic logic is computationally infeasible and prone to mode collapse. We propose a **Hybrid Curriculum** that strategically combines symbolic generation with real-world financial text.

Phase 1: Synthetic Logic Core

100B tokens of "High-Density Logic" generated programmatically. Pure mathematical reasoning: accounting identities, depreciation schedules, compound interest calculations, tax bracket applications.

This data exclusively trains the Quantitative Head, forcing the model to learn arithmetic operators (+, -, *, /) and basic accounting constraints before encountering real-world ambiguity.

1

Phase 2: Semantic Wrapper

2

1.4T tokens of real-world financial text: SEC filings, earnings transcripts, audit reports, accounting textbooks, financial news.

Initial mix: 80% Real / 20% Synthetic. As training progresses, we **anneal** toward higher synthetic weighting to ensure the model doesn't "forget" mathematical precision while learning linguistic fluency.

Phase 3: Golden Corpus

3

10B tokens of expert-annotated "Chain of Audit" data where every numerical claim is traced to source documents and every regulatory conclusion is linked to specific code sections.

This final stage binds the reasoning to the Citation Head, teaching the model the crucial skill of *justification*—not just what the answer is, but why it's legally correct.

Synthetic Data Generation: The Arithmetic Simulator

Template-Based Logic Generation

We construct a domain-specific language (DSL) for financial scenarios:

```
SCENARIO: Depreciation
ASSET_COST: $50,000
METHOD: Straight-line
USEFUL_LIFE: 10 years
SALVAGE_VALUE: $5,000
COMPUTE: Annual_depreciation
```

The simulator generates thousands of variations by sampling parameter distributions, creating diverse reasoning chains that all resolve to verifiable ground truth.

Complexity Curriculum

We gradually increase scenario complexity:

1. **Level 1:** Single-step arithmetic (addition, subtraction)
2. **Level 2:** Multi-step linear chains (revenue - expenses = profit)
3. **Level 3:** Nested dependencies (tax on profit after depreciation)
4. **Level 4:** Conditional logic (different tax treatment based on entity type)
5. **Level 5:** Multi-period calculations (time value of money, amortization schedules)

This staged approach prevents the model from memorizing shortcuts and forces genuine arithmetic understanding.

Experimental Design: Baselines & Reality Checks

1

Llama-3 (8B Parameters)

Role: Generalist baseline representing state-of-the-art open-source foundation models.

Expected Performance: Strong linguistic fluency but 15-20% error rate on multi-step financial calculations. Will likely fail on problems requiring exact numerical precision across more than 3 computational steps.

2

MathGLM

Role: Specialist baseline—a model explicitly tuned for mathematical reasoning but not domain-adapted to finance.

Expected Performance: Better arithmetic precision than Llama-3 but will struggle with financial domain knowledge (e.g., when to use FIFO vs. LIFO inventory accounting) and regulatory grounding.

3

GPT-4 (Oracle Baseline)

Role: Upper bound assessment. GPT-4 represents the ceiling of what current architectures can achieve through scale and RLHF.

Expected Performance: Best overall, but still exhibits documented failures on complex financial arithmetic chains. Our hypothesis: VAMN-7B will outperform GPT-4 specifically on the "Financial Arithmetic Benchmark" despite being ~200x smaller.

4

Critical Success Criterion: We do not expect VAMN-7B to match GPT-4 on open-ended creative writing or general knowledge. Our target is **<1% Numerical Error Rate** on financial computation benchmarks where GPT-4 currently errors ~15-20% of the time.

Evaluation Metrics: Beyond Perplexity

Standard language modeling metrics like perplexity are irrelevant for correctness. Financial AI requires **precision-focused evaluation** that measures not just linguistic quality but mathematical and legal accuracy.

NR-RMSE: Normalized Root Mean Square Error

Measures deviation of predicted numerical values from ground truth, normalized by the magnitude of the true value to account for scale variance:

$$\text{NR-RMSE} = \sqrt{(\Sigma((\hat{y} - y)/y)^2/n)}$$

This metric penalizes both small errors on large values and large errors on small values proportionally.

Citation F1-Score

Precision and recall on retrieving the correct legal statute for a given financial conclusion.

Precision = (Correct Citations) / (All Cited), Recall = (Correct Citations) / (Should Have Cited)

We also track **hallucination rate**—percentage of citations to non-existent code sections.

Consistency Score

Measures internal logical contradictions within a generated financial document.

Examples: Does the balance sheet actually balance? Do cash flows reconcile with reported profit? Are tax calculations consistent with stated marginal rates?

Implemented via symbolic verification: we parse the model's outputs and check mathematical identities.

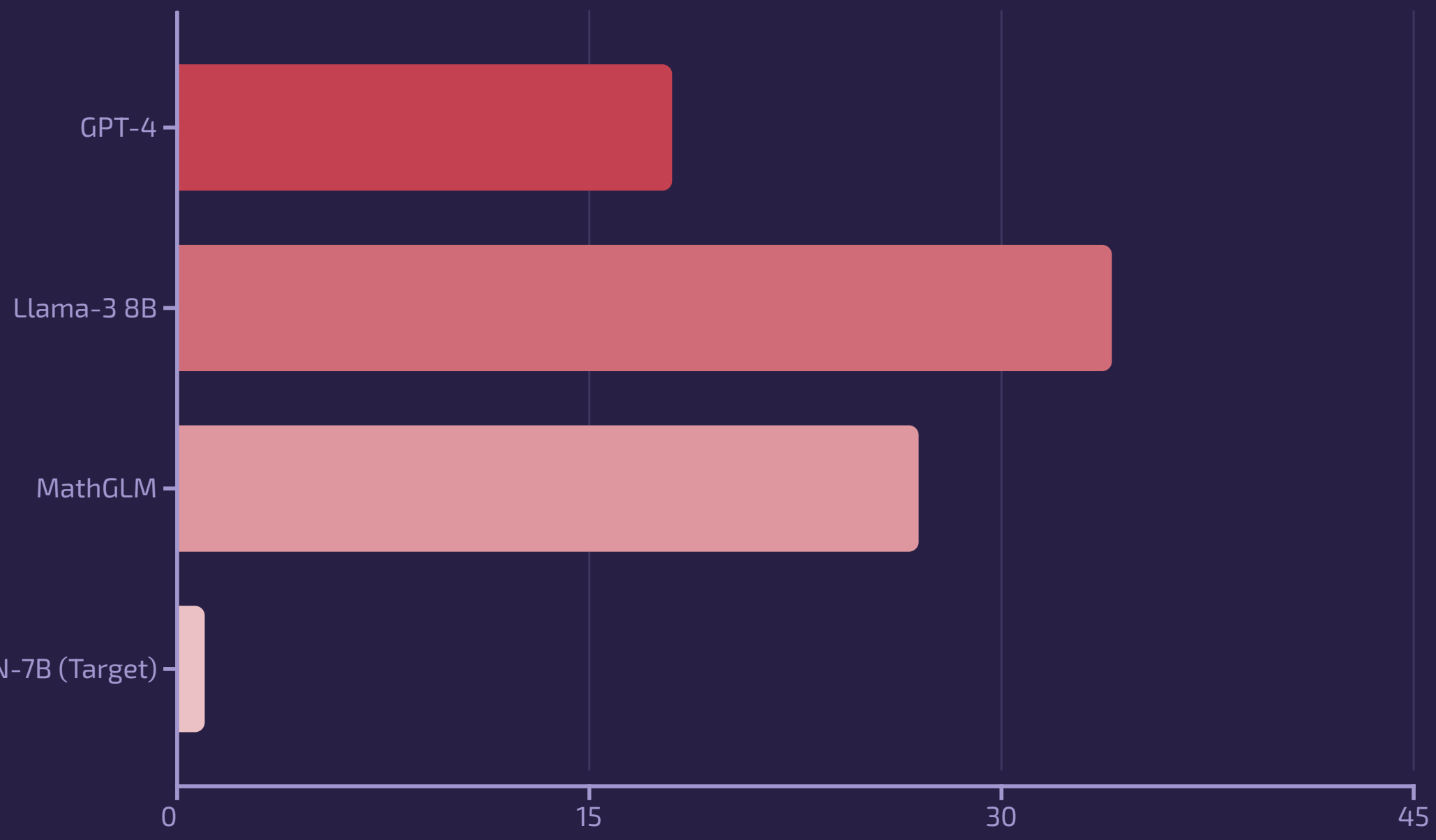
Philosophical Note: A model can have perfect perplexity and still be completely wrong. Our metrics directly measure the properties that matter for financial AI: mathematical precision, legal defensibility, and logical coherence.

The Financial Arithmetic Benchmark (FAB)

Benchmark Composition

We construct a novel evaluation suite specifically designed to stress-test numerical reasoning in financial contexts:

- **1,000 Multi-Step Problems:** Spanning audit procedures, tax calculations, and financial reporting
- **Difficulty Levels:** Stratified from basic (2-3 steps) to expert (10+ dependent calculations)
- **Real-World Grounding:** Based on actual CPA exam questions and Big 4 audit work papers
- **Trap Questions:** Problems designed to exploit common LLM failure modes (e.g., magnitude confusion, unit mismatches)



Preliminary projections based on architectural advantages and synthetic data quality. The **<1% target** for VAMN represents the threshold for production deployment in audit workflows.

Central Hypothesis: Architecture > Scale

Core Thesis

We hypothesize that **VAMN-7B will outperform GPT-4 on the Financial Arithmetic Benchmark despite being approximately 200x smaller**, proving that architectural specialization is more effective than parameter scaling for domain-specific reasoning tasks.

Why This Should Work

- **Dedicated Computational Pathways:** Regression head optimized exclusively for numerical precision without linguistic interference
- **Elimination of Tokenization Artifacts:** Treating numbers as continuous values removes the discrete approximation errors
- **Task-Specific Loss Functions:** Log-Cosh regression provides stronger gradient signals for arithmetic than cross-entropy
- **Curriculum Learning:** 100B tokens of pure mathematical reasoning creates stronger inductive biases than GPT-4's general training

The Scaling Law Anomaly

Traditional scaling laws assume architectural homogeneity—that all models share the same fundamental design. But **different architectures have different efficiency frontiers.**

A 7B parameter model with the right architectural priors can achieve superhuman performance on a narrow task while a 1.5T parameter general model struggles, because the general model's capacity is diluted across countless unrelated domains.

This is not a refutation of scaling laws—it's a demonstration that specialization creates a fundamentally different optimization landscape.

The Self-Auditing Paradigm: Glass Box AI

Input Financial Scenario

Client provides transaction data and requests tax liability calculation

Regulatory Acceptance

Transparent reasoning chain allows regulators to approve or reject without black-box interpretation

Triple-Stream Processing

VAMN generates simultaneous outputs: explanation text, numerical result, regulatory citations

Automated Verification

Symbolic verifier checks mathematical consistency and citation validity

Human Audit

CPA reviews the audit trail, inspecting Stream C citations and Stream B calculations independently

This creates a fundamentally new category of AI system: one where the reasoning process is **architecturally transparent** rather than post-hoc interpretable. Auditors don't need to trust the model—they can verify its work using the same professional standards applied to human accountants.

Economic Impact: The $600 Billion Opportunity

Market Landscape

The global Audit, Tax, and Accounting services market represents **$600 billion in annual revenue**, with the Big 4 firms (Deloitte, PwC, EY, KPMG) capturing ~40% market share.

This industry currently relies on highly-educated human labor because AI systems are deemed unsafe for high-stakes financial work. The liability risk of using black-box AI outweighs potential efficiency gains.

40%

Automatable Tier-1 Tasks

Verification, vouching, and mathematical accuracy checks—routine but time-intensive procedures

60%

Cost Reduction Potential

Labor cost savings for firms adopting VAMN for automated initial review and reconciliation

$240B

Addressable Market

Portion of the global audit market representing automatable verification and compliance tasks

VAMN removes the safety bottleneck by providing architectural transparency and verifiable reasoning. This doesn't eliminate auditors—it elevates them to focus on judgment-intensive tasks (fraud detection, business risk assessment) while automating mechanical verification.

Implementation Roadmap & Resource Requirements

1

Months 0-3: Foundation

Objective: Build synthetic data generation pipeline and implement triple-stream architecture

- Develop financial scenario DSL and arithmetic simulator
- Generate initial 100B token synthetic logic corpus
- Implement PCGrad-based training loop with multi-task loss

Team: 2 ML Engineers, 1 Financial Domain SME

2

Months 4-9: Pre-Training

Objective: Train VAMN-7B on combined synthetic and real-world financial text

- Curriculum training: 80% real / 20% synthetic → 50/50 annealing
- Continuous evaluation on held-out arithmetic benchmarks
- Hyperparameter tuning for loss weighting and learning rate schedules

Compute: ~80,000 A100 GPU-hours (~$200k)

3

Months 10-15: Fine-Tuning & Evaluation

Objective: Fine-tune on Golden Corpus and conduct comprehensive benchmark evaluation

- Train Citation Head on expert-annotated audit data
- Run full FAB evaluation suite against GPT-4 and MathGLM baselines
- Conduct red-team stress testing and failure analysis

Compute: ~20,000 A100 GPU-hours (~$50k)

4

Months 16-18: Production Alpha

Objective: Deploy VAMN-Alpha in controlled pilot with audit firm partner

- Integration with existing audit workflow software
- Human-in-the-loop evaluation on real client engagements
- Regulatory consultation and compliance documentation

Budget & Investment Thesis

Compute Infrastructure

100,000 A100 GPU-hours for training and evaluation

Personnel (18 months)

3-person core team plus contract annotators for Golden Corpus

Data & Licensing

Regulatory text corpus access and expert annotation budget

Operations & Contingency

Infrastructure, tools, partnerships, and 15% buffer

Total Seed Requirement: $1.5M - $2M

This budget delivers **VAMN-Alpha**: a production-ready prototype demonstrating <1% error rate on the Financial Arithmetic Benchmark and validated in a real-world audit pilot.

Risk-Adjusted Return Profile

- **Technical Risk:** Moderate. Triple-stream architecture builds on proven components (transformers, regression heads, hierarchical softmax)
- **Market Risk:** Low. $600B established market with clear pain point (AI safety) that VAMN directly addresses
- **Regulatory Risk:** Manageable. Self-auditing design anticipates compliance requirements

Exit Scenarios: Acquisition by Big 4 firm, enterprise SaaS licensing, or horizontal expansion to legal/healthcare compliance domains

Conclusion: The Future of Vertical AI

Architectural Specificity > General Intelligence

The future of artificial intelligence in vertical domains lies not in building ever-larger general models, but in designing **architecturally specialized systems** that reflect the structure of the problem domain itself.

Core Contribution

VAMN demonstrates that financial numeracy is not a language task wrapped in mathematical terminology—it is a **mathematical task wrapped in language**. By encoding this reality directly into the neural architecture through triple-stream decomposition, we can achieve superhuman precision on domain-specific reasoning while maintaining linguistic fluency.

The triple-stream paradigm is not limited to finance. Any domain requiring the synthesis of natural language, precise computation, and regulatory compliance—medicine, engineering, legal analysis—could benefit from similar architectural specialization.

Broader Implications

VAMN represents a philosophical shift in how we approach AI for high-stakes domains. Rather than asking "How do we make AI safe through alignment?", we ask **"How do we design AI that is inherently verifiable?"**

The answer: Build transparency into the architecture itself. Make the model's reasoning process observable not through post-hoc interpretability techniques, but through explicit computational streams that can be independently audited.

This bridges the "Trust Gap" and enables the era of **Autonomous Finance**—AI systems that don't replace human judgment but amplify it through reliable, verifiable computational assistance.