Thinking Small Models: Multi-Stage Reasoning for Interpretable Machine Learning

Kossiso Royce
Electric Sheep Africa
Kossi@electricsheep.africa

Abstract

Large language models (LLMs) have demonstrated reasoning capabilities through techniques like chain-of-thought prompting and self-correction. We ask: can similar patterns be applied to traditional ML models? We present **Thinking XGBoost**, a multi-stage XGBoost pipeline that structures predictions through specialized heads, hybrid aggregation, and critic-triggered refinement. Our main contribution is the **Reasoning Quality Score (RQS)**, a formal framework with five metrics for evaluating interpretability in non-LLM models: Decomposability, Self-Correction, Reasoning Coherence, Explanation Faithfulness, and Graceful Degradation. On a synthetic fraud detection dataset, we demonstrate that (1) critic-based self-correction achieves F1=0.74 in detecting aggregator errors, and (2) the RQS framework provides measurable targets for reasoning transparency. The architecture itself builds on established ensemble techniques (stacking, cascades) but offers a concrete testbed for the proposed evaluation framework. This work represents an initial inquiry into structured reasoning for traditional ML; while early results show a ∼2% AUC trade-off, this proof-of-concept demonstrates promising directions that we continue to refine.

Keywords: Interpretable Machine Learning, XGBoost, Reasoning, Self-Correction, Explainability, Fraud Detection

1 Introduction

1.1 Motivation

The success of large language models in complex reasoning tasks has sparked interest in understanding *how* models arrive at decisions, not just *what* they predict. Techniques like chain-of-thought prompting [Wei et al., 2022], self-consistency [Wang et al., 2023], and GRPO [Shao et al., 2024] have enabled LLMs to decompose problems, verify their work, and self-correct errors.

Meanwhile, traditional ML models—XGBoost, random forests, neural networks—remain largely opaque. While feature importance and SHAP values provide post-hoc explanations, they lack the structured reasoning traces that make LLM outputs interpretable. This gap is particularly problematic in high-stakes domains like finance, healthcare, and legal systems, where regulators increasingly demand not just accurate predictions, but *explainable* ones.

1.2 Research Questions

We address three questions:

1. **Can traditional ML models "think"?** Can we adapt LLM reasoning patterns—multi-step decomposition, aggregation, self-correction—to gradient boosting models?

2. **How do we measure reasoning quality?** LLMs are evaluated on chain-of-thought coherence and self-consistency. What equivalent metrics exist for small models?

3. **Does thinking help?** Beyond interpretability, does structured reasoning improve model robustness or error detection?

1.3 Contributions

1. **Reasoning Quality Score (RQS) Framework** (Primary): Five formal metrics with mathematical definitions for evaluating "reasoning" in non-LLM models. This addresses the lack of standardized evaluation for structured interpretability.

2. **Critic-based Self-Correction**: Empirical demonstration that an XGBoost critic can detect aggregator errors with F1=0.74, enabling selective refinement on 5.4% of predictions.

3. **Reference Implementation**: A 4-stage pipeline (heads → aggregator → critic → refiner) that serves as a testbed for the RQS framework. The architecture builds on stacking [Wolpert, 1992] and cascades [Viola and Jones, 2001].

1.4 What Does "Thinking" Mean for Small Models?

We use "thinking" as a functional analogy, not a cognitive claim. In LLMs, "thinking" typically refers to:

- **Decomposition**: Breaking problems into intermediate steps
- **Aggregation**: Combining evidence toward a conclusion
- **Self-correction**: Detecting and revising errors

We propose that a traditional ML model exhibits "thinking" behavior if it satisfies three operational criteria:

Criterion	LLM Equivalent	Small Model Equivalent
Decomposable	Chain-of-thought steps	Interpretable sub-predictions (heads)
Aggregative	Combining reasoning paths	Explicit aggregation with traceable weights
Self-correcting	Verifier/critic models	Error-detection triggering refinement

Table 1: Mapping LLM reasoning concepts to small models.

Formal Definition: A model M "thinks" if:

1. Its prediction can be decomposed into $K \geq 2$ interpretable sub-predictions

2. The aggregation mechanism is explicit and traceable

3. A self-correction mechanism exists with $F1 > 0.5$ on error detection

This is deliberately minimal. We do not claim small models reason like humans or LLMs—only that they can exhibit *structured transparency* that mimics the functional properties we value in LLM reasoning. The RQS framework operationalizes this definition into measurable metrics.

2 Related Work

2.1 Interpretable Machine Learning

Traditional interpretability methods focus on post-hoc explanations:

- **Feature Importance**: Measures variable contributions [Breiman, 2001]
- **SHAP Values**: Game-theoretic feature attributions [Lundberg and Lee, 2017]
- **LIME**: Local linear approximations [Ribeiro et al., 2016]

These approaches explain *what* features matter but not *how* the model reasons through them.

2.2 LLM Reasoning

Recent advances in LLM reasoning include:

- **Chain-of-Thought (CoT)**: Decomposing problems into steps [Wei et al., 2022]
- **Self-Consistency**: Sampling multiple reasoning paths [Wang et al., 2023]
- **GRPO**: Group Relative Policy Optimization for reasoning [Shao et al., 2024]
- **Critics and Verifiers**: Separate models that check outputs [Cobbe et al., 2021]

Our work adapts these concepts to gradient boosting.

2.3 Ensemble Methods with Structure

Prior work on structured ensembles includes:

- **Stacking**: Meta-learners over base predictions [Wolpert, 1992]
- **Cascades**: Sequential refinement [Viola and Jones, 2001]
- **Mixture of Experts**: Gated expert selection [Jacobs et al., 1991]

Our architecture combines these ideas with explicit reasoning traces and critic-based self-correction, providing a testbed for evaluating reasoning quality.

3 Methodology

3.1 Problem Setting

We consider binary classification for fraud detection with features $\mathbf{x} \in \mathbb{R}^d$ and label $y \in \{0, 1\}$. Features are partitioned into semantic groups $G = \{G_1, \ldots, G_K\}$ representing distinct fraud dimensions (e.g., transaction amount, velocity, location).

3.2 Architecture Overview

The Thinking XGBoost pipeline consists of four stages:

```
Stage 1: Reasoning Heads       h_k(x) -> [0,1] for k in {1,...,K}
Stage 2: Hybrid Aggregator     a(h_1,...,h_K) -> [0,1]
Stage 3: Critic                c(a, h_1,...,h_K) -> [0,1]
Stage 4: Refiner               r(x, h_1,...,h_K) -> [0,1]
```

Final prediction:

$$\hat{y} = \begin{cases} r(\mathbf{x}) & \text{if } c(\cdot) > \tau \\ a(\cdot) & \text{otherwise} \end{cases} \tag{1}$$

where τ is the critic threshold.

3.3 Stage 1: Reasoning Heads

Each reasoning head h_k is an XGBoost classifier trained on feature subset G_k:

$$h_k(\mathbf{x}) = \text{XGBoost}_k(\mathbf{x}_{G_k}) \tag{2}$$

Heads specialize in detecting fraud signals within their domain:

- **Amount head**: Transaction amount anomalies
- **Velocity head**: Unusual transaction frequency
- **Merchant head**: Merchant risk factors
- **Location head**: Geographic signals
- **Device head**: Device/channel risk
- **Time head**: Temporal patterns

Each head outputs a fraud probability $h_k(\mathbf{x}) \in [0, 1]$ and a binary signal $s_k = \mathbb{1}[h_k(\mathbf{x}) > 0.5]$.

Learned Weights: Each head receives a weight w_k proportional to its training AUC:

$$w_k = \frac{\text{AUC}_k}{\sum_j \text{AUC}_j} \tag{3}$$

3.4 Stage 2: Hybrid Aggregator

The aggregator combines head outputs using a hybrid approach:

$$a(\mathbf{h}) = \alpha \cdot \underbrace{\sum_k w_k h_k}_{\text{Weighted Average}} + (1 - \alpha) \cdot \underbrace{f_{\text{XGB}}(\mathbf{h}, \mathbf{h} \otimes \mathbf{h})}_{\text{XGBoost with Interactions}} \tag{4}$$

where $\alpha = 0.6$ (blend ratio) and $\mathbf{h} \otimes \mathbf{h}$ includes pairwise products $h_i \cdot h_j$ for interaction terms.

The weighted average component provides **faithfulness**—changes in head scores directly affect the output. The XGBoost component captures **non-linear interactions** between heads.

3.5 Stage 3: Critic

The critic model predicts when the aggregator will be wrong:

$$c(\cdot) = \text{XGBoost}_{\text{critic}}(\mathbf{z}) \tag{5}$$

where $\mathbf{z}$ includes:

- Aggregator prediction $a(\mathbf{h})$
- Aggregator confidence $|a - 0.5| \times 2$
- Individual head scores h_k
- Head-aggregator deviation $|h_k - a|$
- Head disagreement (std, range)
- Method disagreement $|\text{weighted\_avg} - \text{xgb\_pred}|$

Training: The critic is trained on cross-validation errors of the aggregator:

$$y_{\text{critic}} = \mathbb{1}[\hat{y}_{\text{CV}} \neq y_{\text{true}}] \tag{6}$$

Using CV predictions avoids the problem of training on zero errors when the aggregator overfits.

Threshold Selection: We optimize for F1 score on error detection:

$$\tau^* = \arg\max_{\tau} F1(\mathbb{1}[c(\cdot) > \tau], y_{\text{critic}}) \tag{7}$$

3.6 Stage 4: Refiner

The refiner is a stronger XGBoost model trained with emphasis on hard cases:

$$r(\mathbf{x}) = \text{XGBoost}_{\text{refiner}}(\mathbf{x}, \mathbf{h}) \tag{8}$$

Training: Samples where the aggregator was wrong receive higher weight:

$$w_i = \begin{cases} 8.0 & \text{if aggregator wrong} \\ 1.0 & \text{otherwise} \end{cases} \tag{9}$$

The refiner uses both original features $\mathbf{x}$ and head outputs $\mathbf{h}$, providing maximum information for difficult cases.

3.7 Reasoning Trace

For each prediction, the pipeline outputs a structured reasoning trace:

```
<REASONING>
  amount        risk: 0.374
  velocity      risk: 0.760
  merchant      risk: 0.715
  location      risk: 0.596
  device        risk: 0.429
  time          risk: 0.484
  --------------------
  Weighted avg:      0.568
  XGBoost pred:      0.815
  Blended:           0.666
  Critic score:      0.669
  [!] REFINEMENT TRIGGERED
</REASONING>

<SOLUTION>
  Probability: 0.054
  Decision:    LEGITIMATE
</SOLUTION>
```

This trace mirrors LLM chain-of-thought outputs, providing interpretable intermediate steps.

4 Reasoning Quality Score (RQS) Framework

4.1 Motivation

LLMs are evaluated on reasoning quality through metrics like chain-of-thought coherence and self-consistency. Traditional ML models lack equivalent evaluation frameworks. We propose the **Reasoning Quality Score (RQS)** with five metrics.

4.2 Metric Definitions

4.2.1 Decomposability (D)

Definition: The degree to which the final prediction can be attributed to interpretable sub-components.

$$D = 1 - \frac{\mathrm{Var}(\hat{y} - \bar{h})}{\mathrm{Var}(\hat{y})} \tag{10}$$

where $\bar{h} = \frac{1}{K}\sum_k h_k$ is the mean head prediction.

Interpretation: $D = 1$ means heads fully explain the prediction; $D = 0$ means heads explain nothing.

Target: $D \geq 0.70$

4.2.2 Self-Correction (SC)

Definition: The model's ability to identify its own errors.

$$SC = F1(\text{critic\_flags}, \text{actual\_errors}) \tag{11}$$

Interpretation: High SC means the critic accurately identifies when the aggregator will be wrong.

Target: $SC \geq 0.30$

4.2.3 Reasoning Coherence (RC)

Definition: Consistency between intermediate reasoning signals and final decision.

$$RC = \frac{1}{K}\sum_k |\rho(h_k, \hat{y})| \tag{12}$$

where ρ is Pearson correlation.

Interpretation: High RC means head scores correlate with final predictions.

Target: $RC \geq 0.50$

4.2.4 Explanation Faithfulness (EF)

Definition: Do the stated reasons actually influence the prediction?

For each head k, we perturb its input features and measure:

$$EF_k = \rho(\Delta h_k, \Delta \hat{y}) \tag{13}$$

$$EF = \frac{1}{K}\sum_k \max(0, EF_k) \tag{14}$$

Interpretation: High EF means perturbing a head's inputs proportionally changes the final output.

Target: $EF \geq 0.60$

4.2.5 Graceful Degradation (GD)

Definition: When the model is wrong, are errors concentrated in identifiable dimensions?

$$GD = \frac{H(\mathbf{p}_{\text{error}})}{\log K} \tag{15}$$

where H is entropy and $\mathbf{p}_{\text{error}}$ is the distribution of which heads had extreme predictions on errors.

Interpretation: Low GD means errors are concentrated (easier to debug); high GD means errors are spread out.

Target: $GD \leq 0.50$

4.3 Composite Score

$$RQS = 0.20 \cdot D + 0.25 \cdot SC + 0.20 \cdot RC + 0.25 \cdot EF + 0.10 \cdot (1 - GD) \tag{16}$$

RQS Range	Interpretation
0.8 – 1.0	Excellent reasoning transparency
0.6 – 0.8	Good reasoning transparency
0.4 – 0.6	Moderate reasoning transparency
0.0 – 0.4	Poor reasoning transparency

Table 2: RQS interpretation scale.

5 Experiments

5.1 Dataset

We use a synthetic fraud detection dataset with 30,000 transactions:

- **Fraud rate**: 8%
- **Features**: 18 (grouped into 6 semantic categories)
- **Patterns**: Normal legitimate, suspicious legitimate (gray zone), obvious fraud, subtle fraud, mixed-signal fraud

The synthetic dataset enables controlled evaluation of reasoning quality with known ground truth patterns.

5.2 Experimental Setup

- **Train/Test Split**: 80/20 with stratification
- **Random Seed**: 42 (fixed for reproducibility)
- **Baseline**: Standard XGBoost (100 trees, depth 6)
- **Metrics**: ROC-AUC, Classification Report, RQS

5.3 Results

5.3.1 Traditional Metrics

Model	ROC-AUC	Precision (Fraud)	Recall (Fraud)
Baseline XGBoost	0.994	0.89	0.85
Thinking Pipeline	0.976	0.85	0.82

Table 3: Traditional performance metrics. The Thinking Pipeline trades ∼2% AUC for full reasoning transparency.

5.3.2 Reasoning Quality Score

Metric	Score	Target	Status
RQS	**0.50**	>0.60	—
Decomposability	0.77	>0.70	✓
Self-Correction	0.33	>0.30	✓
Coherence	0.57	>0.50	✓
Faithfulness	0.57	>0.60	×
Graceful Degradation	0.91	<0.50	×

Table 4: Reasoning Quality Score results. Three of five metrics meet their targets.

5.3.3 Self-Correction Analysis

Metric	Value
Samples Refined	326 / 6000 (5.4%)
Critic F1	0.74
Decisions Changed	312

Table 5: Self-correction analysis. The critic successfully identifies uncertain predictions.

5.4 Ablation Studies

5.4.1 Blend Ratio Impact

Blend Ratio (α)	RQS	Faithfulness	AUC
0.0 (XGBoost only)	0.43	0.32	0.988
0.4	0.46	0.42	0.982
0.6	**0.50**	**0.57**	**0.976**
0.8	0.49	0.68	0.968
1.0 (Weighted avg only)	0.28	0.53	0.963

Table 6: Blend ratio ablation. The optimal blend ratio (0.6) balances faithfulness and predictive power.

5.4.2 Component Contributions

Configuration	RQS
Heads only	0.32
+ Aggregator	0.38
+ Critic	0.44
+ Refiner	0.50

Table 7: Component contributions. Each stage contributes to overall reasoning quality.

6 Discussion

6.1 Key Findings

1. **LLM reasoning patterns transfer to small models**: Multi-stage decomposition, aggregation, and self-correction improve interpretability without catastrophic accuracy loss.

2. **Self-correction works**: The critic achieves F1=0.74 in detecting aggregator errors, enabling selective refinement.

3. **Trade-offs exist**: Higher faithfulness (explaining predictions through heads) reduces predictive power. The hybrid aggregator balances this trade-off.

6.2 Limitations

1. **Graceful Degradation (GD = 0.91)**: Errors spread across multiple heads rather than concentrating in identifiable dimensions. This appears to be a fundamental limitation of multi-dimensional fraud patterns.

2. **Faithfulness gap (EF = 0.57 vs 0.60 target)**: The XGBoost component of the aggregator dampens individual head contributions.

3. **Synthetic data**: Results on real-world data may differ due to noisier patterns and feature correlations.

6.3 Practical Implications

1. **Regulatory Compliance**: The reasoning trace satisfies explainability requirements in regulated industries.

2. **Human-in-the-Loop**: Analysts can quickly triage alerts by examining which dimensions flagged risk.

3. **Debugging**: When models fail, the trace identifies which reasoning dimension was wrong.

6.4 Future Work

1. **Real-world validation**: Apply to production fraud detection systems.

2. **Other domains**: Healthcare diagnosis, credit risk, content moderation.

3. **Metric refinement**: Develop causal faithfulness measures.

4. **Architecture improvements**: Attention-based aggregation, learned critic thresholds.

7 Conclusion

We introduced the **Reasoning Quality Score (RQS)** framework—five metrics for evaluating structured interpretability in traditional ML models. This addresses a gap in how we measure "reasoning" outside of LLMs.

As a testbed, we built Thinking XGBoost, a 4-stage pipeline combining established techniques (stacking, cascades) with critic-based self-correction. Key empirical findings:

- The critic achieves F1=0.74 in detecting aggregator errors
- 3/5 RQS targets met (Decomposability, Self-Correction, Coherence)

- $\sim$2% AUC trade-off for full reasoning transparency

Limitations: Faithfulness (0.57) and Graceful Degradation (0.91) remain below targets. The evaluation uses synthetic data only.

This work represents an initial inquiry into structured reasoning for traditional ML. While our proof-of-concept shows a modest $\sim$2% AUC trade-off, the results demonstrate that LLM-inspired reasoning patterns can be meaningfully adapted to small models. We are actively refining the approach to improve Faithfulness and reduce Graceful Degradation.

We hope the RQS framework provides a starting point for standardized evaluation of interpretable ML systems, independent of the specific architecture used.

References

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. (2024). DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2):241–259.

A Hyperparameters

Component	Parameter	Value
Reasoning Heads	n_estimators	60
	max_depth	5
	learning_rate	0.1
XGBoost Aggregator	n_estimators	30
	max_depth	4
	learning_rate	0.1
Critic	n_estimators	100
	max_depth	5
	learning_rate	0.05
Refiner	n_estimators	180
	max_depth	8
	learning_rate	0.03
Blend Ratio	α	0.6
Error Weight	w_{error}	8.0

Table 8: Hyperparameters used in the Thinking XGBoost pipeline.

B Feature Groups

Group	Features
Amount	amount, avg_amount_30d, amount_vs_avg_ratio
Velocity	txn_count_1h, txn_count_24h, velocity_score
Merchant	merchant_category_risk, merchant_age_days, merchant_txn_volume, merchant_risk_score
Location	distance_from_home, is_foreign_country, country_risk_score, location_risk
Device	is_new_device, failed_attempts_24h
Time	hour_of_day, day_of_week

Table 9: Feature groups used for reasoning heads.