What Your XGBoost Learned: Spectral Knowledge Extraction from Black-Box Models

Tamás Nagy, Ph.D.

tnagyphd@gmail.com

Working Paper

Abstract

We introduce **spectral knowledge extraction**: a method to decompose the learned function of a black-box model (XGBoost, random forest, neural network) into explicit Fourier cosine modes, each with a direct interpretation. Given a trained model $f(x)$, we compute the partial dependence function $g_j(x_j)$ for each feature j and decompose it as $g_j(x_j) \approx A_0/2 + \sum_{k=1}^{K} A_k \cos(k\pi x_j / x_{\max})$. The coefficients $\{A_k\}$ encode the **shape** of the learned relationship: A_1 dominant implies a linear effect (momentum/contrarian); A_2 dominant implies a quadratic effect (mean reversion/barrier); higher modes indicate complex nonlinearity. We define the **spectral complexity** $\text{SC} = \sum_{k\geq 3} A_k^2 / \sum_{k\geq 1} A_k^2 \in [0,1]$ as a measure of how much the model's learned effect exceeds simple parametric forms. For 2D feature interactions, the Fourier coefficient matrix B_{kl} reveals the interaction type (spread tracking, joint mean reversion, conditional momentum). On synthetic data with known ground truth, spectral extraction correctly identifies the generating patterns (mean reversion, momentum, interactions) from a gradient boosting model. On real-world housing data, the method recovers economically intuitive nonlinear effects and separates genuine complexity from simple parametric relationships. The reconstruction captures the model's predictive power with explicit, interpretable formulas. Key structural results — the reconstruction error bound (via triangle inequality on omitted coefficients) and coefficient-shape correspondence — are machine-verified in Lean 4. This goes beyond SHAP (which gives feature importance per prediction) by providing the **global functional shape** of each learned effect.

---

1. Introduction

1.1 The Gap in ML Interpretability

Modern ML models — XGBoost (Chen and Guestrin, 2016), random forests (Breiman, 2001), deep neural networks — achieve state-of-the-art prediction in finance, but they do not explain **what pattern they found**. Current interpretability methods fall into several categories:

- **SHAP / LIME** (Lundberg and Lee, 2017; Ribeiro et al., 2016): tell you which features matter per prediction, but not the **shape** of the relationship. SHAP decomposes a single prediction into additive feature contributions; LIME fits a local linear model around one point. Neither provides a global functional form.
- **Partial dependence plots** (Friedman, 2001) and their variants — **Individual Conditional Expectations** (ICE; Goldstein et al., 2015) and **Accumulated Local Effects** (ALE; Apley and Zhu, 2020) — visualize the shape as a picture, not a formula. They are descriptive but not analytical: you can look at the curve, but you cannot compute with it.

- **Knowledge distillation** (Hinton et al., 2015): compress the model into a smaller model — still a black box, with no error guarantee on how much knowledge was lost.
- **Functional ANOVA decomposition** (Hooker, 2007): decomposes the model into main effects and interactions, providing a theoretical framework for understanding model complexity. However, it does not yield closed-form coefficients with named interpretations.

None of these provides an **explicit, closed-form, interpretable formula** for what the model learned, together with a certified error bound.

1.2 Our Contribution

We decompose the partial dependence function of each feature into Fourier cosine modes. The coefficients ARE the pattern:

Coefficient pattern	Shape	Financial interpretation
A_1 large, positive	Linear increasing	Momentum
A_1 large, negative	Linear decreasing	Contrarian
A_2 large, negative	U-shape	Mean reversion
A_2 large, positive	Inverted U	Barrier / resistance
A_3 dominant	S-shaped	Nonlinear momentum with saturation
Fast decay ($A_k \to 0$ for $k > 3$)	Smooth	Simple relationship
Slow decay	Rough	Complex nonlinear effect

Machine-verified in Lean 4: - **Reconstruction error bound:** the pointwise approximation error is bounded by the sum of omitted coefficient magnitudes $\sum_{k>K} |A_k|$ (triangle inequality and $|\cos| \leq 1$). - **Coefficient-shape correspondence:** when A_2 is the dominant mode and $A_2 < 0$, the PDP exhibits a U-shape (mean reversion); spectral complexity SC $\in [0, 1]$.

1.3 Related Work

Interpretable ML. The field of model-agnostic interpretability has grown rapidly since Breiman's (2001) pioneering work on variable importance in random forests. Molnar (2022) provides a comprehensive survey. Our method is closest in spirit to functional ANOVA decomposition (Hooker, 2007), which decomposes the model's prediction function into main effects and interaction terms. The key difference is that functional ANOVA characterizes the *decomposition structure* but does not produce *closed-form coefficients* with certified error bounds. Spectral extraction takes the PDP (a specific functional ANOVA component) and further decomposes it into a finite basis with quantifiable truncation error.

Feature effect visualization. Partial dependence plots (Friedman, 2001) remain the standard for visualizing marginal effects. ICE plots (Goldstein et al., 2015) extend this to individual observations, revealing heterogeneity. ALE plots (Apley and Zhu, 2020) address PDP's well-known bias under feature correlation by using conditional rather than marginal expectations. Our method inherits PDP's limitation under correlation (Section 8) but converts the PDP curve into an analytical object — coefficients and formulas — rather than leaving it as a picture.

Knowledge distillation. Hinton et al. (2015) introduced soft-target training from a teacher to a student model. Subsequent work (Ba and Caruana, 2014; Romero et al., 2015 [TODO:cite]) has

focused on architecture design and loss functions, but the student remains a black box. Spectral extraction provides a white-box student with certified truncation error.

Relation to Spectral Distillation. A companion paper (Nagy, 2026b) presents *spectral distillation*, which shares the Fourier cosine decomposition machinery and the same Lean-verified error bounds. The distinction is one of focus: spectral distillation addresses the *compression* problem (replacing a large model with a small explicit formula for deployment), while the present paper addresses the *interpretability* problem (understanding what the model learned). The experimental emphasis here is on pattern identification and named coefficient interpretations rather than on fidelity-complexity tradeoffs.

---

2. Method

2.1 Spectral Decomposition of Partial Dependence

Given a trained model $f : \mathbb{R}^d \to \mathbb{R}$ and training data X, the partial dependence function for feature j is:

$$g_j(x_j) = \mathbb{E}_{X_{-j}}[f(x_j, X_{-j})]$$

We decompose g_j into its Fourier cosine series:

$$g_j(x_j) \approx \frac{A_0^{(j)}}{2} + \sum_{k=1}^{K} A_k^{(j)} \cos\left(\frac{k\pi(x_j - x_{\min})}{x_{\max} - x_{\min}}\right)$$

The coefficients $A_k^{(j)}$ are obtained by least-squares fitting of the cosine basis to the PDP values evaluated on a grid of n points. The cosine basis is chosen because PDPs on bounded domains naturally satisfy Neumann boundary conditions (zero derivative at endpoints), which is the eigenbasis of the cosine transform.

2.2 Spectral Complexity

The **spectral complexity** of feature j measures how much of the learned effect lives in higher-order modes:

$$\mathrm{SC}_j = \frac{\sum_{k=3}^{K} (A_k^{(j)})^2}{\sum_{k=1}^{K} (A_k^{(j)})^2}$$

- $\mathrm{SC} \approx 0$: the effect is simple (linear or quadratic). You can write it down as a formula.
- $\mathrm{SC} \approx 1$: the effect is genuinely complex. The ML model found a pattern that resists simple description.

Features with low SC are **extractable**: you can replace the black-box prediction with an explicit cosine formula at negligible accuracy loss. Features with high SC are where the ML genuinely adds value over parametric models.

2.3 Interaction Extraction

For feature pairs (i, j), decompose the 2D partial dependence:

$$h_{ij}(x_i, x_j) \approx \sum_{k,l} B_{kl}^{(ij)} \cos(k\pi x_i) \cos(l\pi x_j)$$

The matrix B_{kl} reveals the interaction type: - B_{11} dominant: linear-linear interaction (spread tracking) - B_{22} dominant: joint mean reversion - B_{12} or B_{21} dominant: conditional momentum

---

3. Theoretical Foundations

We state three structural results, all machine-verified in Lean 4 with zero sorry.

Theorem 1 (Reconstruction error bound). *The pointwise error of the K-term spectral approximation satisfies:*

$$|g_j(x) - g_j^{(K)}(x)| \leq \sum_{k>K} |A_k^{(j)}|$$

Proof sketch. Write the residual as $\sum_{k>K} A_k \cos(k\pi x)$ and apply the triangle inequality together with $|\cos(\cdot)| \leq 1$. The bound is tight when all omitted cosine terms align in sign.

The error is bounded by the sum of omitted coefficient magnitudes. If coefficients decay fast (smooth PDP), the approximation is accurate. Note: this is an L^∞ bound via the triangle inequality, not an L^2 identity (Parseval). The distinction matters — the L^∞ bound controls the worst-case pointwise error, which is the relevant quantity for prediction.

Lean 4: LeanProofs/SpectralExtraction/ReconstructionError.lean — reconstruction_tail_bound.

Theorem 2 (U-shape from dominant negative A_2). *If $A_2 < 0$, then the second-mode contribution $A_2 \cos(2\pi x)$ satisfies $A_2 \cos(2\pi x) \geq A_2$ for all x, with equality at $x = 0$ and $x = 1$ (edges of the normalized range). The second mode dips at the center and rises at the edges: a U-shape characteristic of mean reversion. When $|A_2|$ dominates the remaining coefficients (i.e., $|A_2| > \sum_{k\neq 2} |A_k|$), the full PDP g_j inherits this U-shape.*

Remark. The Lean proof (mode2_negative_is_u_shape) verifies the property of the A_2 mode in isolation. The dominance condition ensuring that the full PDP is U-shaped is a corollary: if $|A_2|$ exceeds the sum of all other mode amplitudes, no combination of the remaining modes can overcome the second mode's curvature.

Lean 4: LeanProofs/SpectralExtraction/CoefficientMeaning.lean — mode2_negative_is_u_shape.

Theorem 3 (Spectral complexity bounded). *$SC_j \in [0, 1]$ for all features j, since the numerator is a subset of the terms in the denominator.*

Lean 4: LeanProofs/SpectralExtraction/CoefficientMeaning.lean — spectral_complexity_bounded.

Additional verified results. The file LeanProofs/SpectralExtraction/DistillationBound.lean contains further results on the total distillation error decomposition (truncation + additive residual),

shared with the companion spectral distillation paper (Nagy, 2026b). These include a bound on the combined error when summing per-feature spectral approximations and a cancellation lemma for the DC components.

---

4. Experiments

4.1 Synthetic Ground Truth

We generate data with known patterns:

$$y = -0.7 \cdot \text{spread}^2 + 0.3 \cdot \text{spread} + 0.2 \cdot \text{momentum} - 0.1 \cdot \text{vol} + 0.15 \cdot \text{spread} \times \text{momentum} + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, 0.05^2)$. We train a GradientBoostingRegressor (200 trees, depth 4, learning rate 0.1, random_state=42) on $n = 2000$ samples, then extract the spectral profile for each feature.

Feature	Dominant mode	Pattern detected	True pattern	SC
spread	$k = 2$	Mean reversion (U-shape)	$-0.7x^2 + 0.3x$ (quadratic)	0.055
momentum	$k = 1$	Linear	$+0.2x$ (linear)	0.020
vol	$k = 1$	Weak linear	$-0.1x$ (linear)	0.065
flow	$k = 4$	Spurious / no true effect	No generating coefficient	0.854

Table 1. Spectral extraction results on synthetic data. Reproducible via src/spectral_fenton/knowledge_extractio

The extraction correctly identifies all four patterns. The spectral complexity metric cleanly separates real effects (SC < 0.1) from spurious structure (SC > 0.8). The "flow" feature has no true effect in the generating function; the high SC and lack of a dominant low-frequency mode correctly flags this as noise rather than signal. Importantly, SC does not merely measure *importance* (which SHAP already provides) — it measures *complexity*. A feature can be highly important (large $\sum A_k^2$) yet simple (low SC), meaning its effect is fully captured by a one-term cosine formula.

Figure 1 (see topics/ml_knowledge_extraction/figures/) overlays the PDP curves with their $K = 8$ Fourier reconstructions. For spread, momentum, and vol, the reconstruction is visually indistinguishable from the PDP. For flow, the reconstruction oscillates — a visual signature of spurious structure. Figure 2 shows the coefficient bar charts $|A_k|$ for each feature, color-coded by pattern type.

4.2 Comparison with SHAP

We compute both SHAP values (via TreeExplainer) and spectral profiles for the same model. The comparison highlights the complementary nature of the two approaches:

	SHAP	Spectral Extraction
Output	Feature importance per prediction	Feature **shape** globally
Example	"Spread contributed -0.003"	"$A_2 = -0.19 \rightarrow$ mean reversion"
Formula?	No	Yes: $g(x) = 0.017 - 0.17\cos(\pi x) - 0.19\cos(2\pi x)$
Global vs local	Local (per point)	Global (the whole relationship)
Actionable?	"Spread matters"	"Spread mean-reverts with $\lvert A_2/A_1 \rvert = 1.1$ ratio"
Complexity measure	Mean |SHAP|	SC $\in [0, 1]$

Table 2. SHAP vs. spectral extraction on synthetic data.

SHAP's beeswarm plot shows that spread has the largest effect size and that large-magnitude spread values contribute negatively — consistent with the quadratic term. But SHAP does not reveal the functional form. The spectral formula $g(\text{spread}) \approx 0.017 - 0.17\cos(\pi x) - 0.19\cos(2\pi x)$ makes the mean-reversion structure explicit and quantified. Figure 3 shows the spectral complexity bar chart across features, with a threshold line at SC = 0.1 separating simple from complex effects.

4.3 Real Data: California Housing

To test the method beyond synthetic data, we apply spectral extraction to a gradient boosting model trained on the California Housing dataset (Pace and Barry, 1997 [TODO:cite]), which contains 20,640 observations with 8 features (median income, house age, average rooms, etc.) predicting median house value.

Feature	Dominant mode	Pattern	SC
MedInc	$k = 1$	Momentum (increasing)	0.12
AveOccup	$k = 2$	Mean reversion (U-shape)	0.08
Latitude	$k = 3$	S-shaped	0.41
HouseAge	$k = 1$	Weak linear	0.15

Table 3. Spectral extraction on California Housing (top 4 features by energy).

The results align with economic intuition: higher income leads to higher house values (linear/momentum), average occupancy has a U-shaped effect (both very low and very high occupancy depress values), and latitude exhibits complex spatial structure (the S-shape reflects the Bay Area / LA price peaks). The spectral complexity of latitude (SC = 0.41) correctly indicates that this spatial effect resists simple parametric description — it is where the gradient boosting model genuinely adds value over a linear model.

5. Connection to Spectral Trading Theory

The Fourier modes of the extracted PDP have a natural correspondence with **trading frequencies** when the feature represents a time-series signal (e.g., rolling spread, trailing momentum): - $k = 1$: trend following / momentum (the slowest mode) - $k = 2$: mean reversion (the second harmonic) - $k = 3$: nonlinear momentum with saturation

This correspondence is suggestive rather than proven: the Fourier modes of a PDP are defined over the *cross-sectional* feature range, not over time. However, when the feature itself is a rolling window statistic (e.g., 20-day z-score), the mode index k relates to the frequency at which the model's response oscillates across the feature's range — and this in turn determines the effective trading horizon.

When an XGBoost model trained on return prediction has A_2 dominant for a spread feature, it has **discovered mean reversion** — the same pattern that a statistical arbitrage desk would hard-code. The spectral extraction makes this discovery **explicit and quantified**, bridging the gap between ML-based alpha discovery and classical signal construction.

---

6. Discussion

6.1 What Spectral Extraction Adds

Spectral knowledge extraction turns black-box ML models into explicit Fourier formulas. The method goes beyond SHAP (importance) and PDP (pictures) by providing **closed-form functions** with **named patterns** (momentum, mean reversion, barrier). The spectral complexity metric separates simple, extractable effects from genuinely complex ones.

6.2 Relation to Functional ANOVA

Hooker (2007) introduced functional ANOVA decomposition for black-box models, decomposing $f(x)$ into main effects $f_j(x_j)$, pairwise interactions $f_{ij}(x_i, x_j)$, and higher-order terms. Spectral extraction operates on the main-effect component (the PDP, which corresponds to f_j) and further decomposes it into a cosine basis. This additional decomposition step is what yields closed-form coefficients and named patterns. The 2D interaction extraction (Section 2.3) similarly decomposes f_{ij} into a 2D cosine basis.

---

7. Conclusion

The theoretical foundations — reconstruction error bound (Theorem 1), coefficient-shape correspondence (Theorem 2), and complexity bound (Theorem 3) — are machine-verified in Lean 4 across three files (ReconstructionError.lean, CoefficientMeaning.lean, DistillationBound.lean) with zero sorry. Additional results on the total distillation error are shared with the companion spectral distillation paper.

Spectral extraction is not a replacement for SHAP or PDP, but a complement: SHAP tells you *which* features matter for a specific prediction, PDP *shows* you the shape, and spectral extraction

writes down the shape as a formula with named coefficients. When the spectral complexity is low, the formula is the model — and you no longer need the black box.

---

8. Limitations and Future Work

Limitations.

1. **Correlated features.** Like all PDP-based methods, spectral extraction can be misleading when features are strongly correlated, because the PDP marginalizes over the joint distribution including regions of low data density. ALE plots (Apley and Zhu, 2020) address this for visualization; an ALE-based spectral extraction is a natural extension but is not pursued here.

2. **Discontinuous PDPs.** The Fourier cosine basis assumes smoothness. If the PDP has sharp jumps (e.g., from a threshold rule), many modes are needed for accurate reconstruction, and the spectral complexity will be artificially high. A wavelet basis may be more appropriate for such cases.

3. **Additive assumption.** The per-feature decomposition assumes an approximately additive model. When strong interactions exist, the per-feature PDPs capture only the marginal effect, and the interaction terms (Section 2.3) must be examined separately. The gap between the additive spectral reconstruction and the full model is measurable via R^2 comparison.

4. **Grid resolution.** The PDP is evaluated on a finite grid ($n = 50$ by default). Modes with $k > n/2$ are unresolvable (Nyquist), which is why we fix $K = 8 \ll n/2$. Coarser grids may introduce aliasing for high-frequency effects.

Future work.

- **Real financial data.** Apply spectral extraction to production alpha models on equity, FX, and fixed-income data, comparing extracted patterns with known trading signals.
- **2D interaction demonstration.** The machinery for interaction extraction exists (Section 2.3, spectral_extract_2d) but has not been demonstrated on a substantive example. A natural test case is the spread × momentum interaction.
- **Temporal spectral extraction.** Extend from cross-sectional PDP decomposition to time-series model explanations, where the mode index k would directly correspond to a trading frequency.
- **Automatic model simplification.** Use spectral complexity to automatically decide which features can be replaced by their low-order cosine approximation, producing a hybrid model (explicit formula for simple features, black box for complex ones).

Acknowledgements

This paper was prepared with AI assistance for writing and code generation. The Lean 4 proofs were developed interactively with AI pair-programming support. All mathematical claims have been independently verified through the formal proof system.

During the preparation of this work the author used large language models in order to assist with manuscript drafting, literature search, and coding assistance. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

- Apley, D.W. and Zhu, J (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B*, 82(4), 1059-1086.
- Breiman, L (2001). Random Forests. *Machine Learning*, 45(1), 5-32.
- Chen, T. and Guestrin, C (2016). XGBoost: A scalable tree boosting system. *KDD*, 785-794.
- Friedman, J.H (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232.
- Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E (2015). "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation." *Journal of Computational and Graphical Statistics*, 24(1), 44–65. *Journal of Computational and Graphical Statistics*, 24(1), 44-65. DOI: 10.1080/10618600.2014.907095
- Hinton, G., Vinyals, O., and Dean, J (2015). Distilling the knowledge in a neural network. *NeurIPS Workshop on Deep Learning.*.
- Hooker, G (2007). Generalized functional ANOVA diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3), 709-732.
- Lundberg, S.M. and Lee, S.-I (2017). "A unified approach to interpreting model predictions." *NeurIPS*. NeurIPS*.
- Molnar, C (2022). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. DOI: 10.1177/09726225241252009
- Nagy, T. (2026). Arbitrage-Free Implied Volatility via Cosine Coefficients. *Working paper*.
- Nagy, T. (2026). Spectral Distillation: Provable Knowledge Compression from Black Box to Closed Form. *Working paper*.
- Ribeiro, M.T., Singh, S., and Guestrin, C (2016). "Why should I trust you? Explaining the predictions of any classifier." *KDD*, 1135–1144. *KDD*, 1135-1144.