

Fourier–Polynomial Features for Obfuscation-Robust Android Malware Detection

Yoshiki Kusama, *Independent Security Researcher*

Abstract—We propose a spectral feature extraction framework for Android malware detection, designed as a *complementary component* to existing structural detectors (API call graphs, control-flow graphs, permission sets) rather than a standalone solution. The method extracts the normalized power spectrum of raw DEX bytecode, approximates it as a polynomial, and compresses it into a compact feature vector via repeated differentiation, leveraging the dual-number representation of the derivative operator [2]. The central theoretical contribution is the *Complementary Discriminative Power Theorem*: under graph-preserving obfuscation—byte-level transformations that leave structural features intact—structural detectors are provably blind by construction, while spectral features remain sensitive to entropy changes in the byte sequence, providing strictly positive conditional mutual information. Polynomial differentiation-based compression reduces feature dimension from $m+1$ to $m-k+1$ in $O(k \cdot m)$ operations, making the spectral component cheap to add to any existing pipeline. We state the method’s limitations explicitly and formalize the conditions under which feature fusion is necessary.

Index Terms—Android malware detection, spectral analysis, Fourier transform, automatic differentiation, dual numbers, feature fusion, obfuscation robustness

I. INTRODUCTION

ANDROID malware detection is typically framed as a classification problem over structural features: API call graphs, control-flow graphs (CFGs), and permission manifests. These representations are semantically rich but share a fundamental vulnerability—they are defined entirely at the code-graph level. An adversary who applies *graph-preserving obfuscation*—encrypting a payload, transposing code blocks, or inserting inert byte sequences—can alter the byte-level statistics of an APK while leaving its structural features completely unchanged. Structural detectors are therefore blind to such transformations by construction.

We propose a complementary approach. By analyzing the *power spectrum* of raw DEX bytecode, we capture byte-level statistical patterns that are orthogonal to structural features. We do not claim that spectral features alone suffice for reliable detection: high-entropy benign components (ProGuard-obfuscated code, cryptographic libraries) limit their standalone precision. Our claim is that spectral features provide a discriminative signal that *no purely structural detector can replicate*, and that combining both yields a strictly stronger detector.

To make spectral features computationally practical, we approximate the power spectrum as a polynomial and compress it via repeated differentiation. The derivative computation is

grounded in the algebra of dual numbers [1], [2], a well-established framework for automatic differentiation that encodes function values and derivatives simultaneously in a 2×2 matrix representation.

Contributions:

- 1) A formal pipeline mapping APK binaries to compact spectral feature vectors with $O(N \log N)$ total complexity.
- 2) The *Complementary Discriminative Power Theorem*: spectral features provide strictly positive information gain over structural features under graph-preserving obfuscation (Theorem 8).
- 3) A formal characterization of the framework’s limitations, grounding the necessity of feature fusion (Section VI).
- 4) A fusion framework for integrating spectral features with existing structural detectors (Section VII).

II. BACKGROUND

A. Dual Numbers and Automatic Differentiation

The algebraic machinery underlying our compression step is the ring of *dual numbers*, introduced by Clifford [1] and widely used in automatic differentiation [2], [3].

Definition 1 (Dual Numbers [1]). *The ring of dual numbers is $\mathbb{D} = \{a + b\varepsilon \mid a, b \in \mathbb{R}, \varepsilon^2 = 0\}$.*

The nilpotency $\varepsilon^2 = 0$ implies $f(x + \varepsilon) = f(x) + f'(x)\varepsilon$ for any differentiable f , making dual arithmetic an exact algebraic model of first-order differentiation [2]. This is the forward-mode automatic differentiation identity: evaluating f at a dual-number argument simultaneously computes f and its derivative at zero extra asymptotic cost.

Definition 2 (Matrix Embedding). *The injective ring homomorphism $\phi : \mathbb{D} \rightarrow M_{2 \times 2}(\mathbb{R})$ is*

$$\phi(a + b\varepsilon) = \begin{pmatrix} a & b \\ 0 & a \end{pmatrix}, \quad F(x) := \begin{pmatrix} f(x) & f'(x) \\ 0 & f(x) \end{pmatrix}.$$

Lemma 1 (Product Rule via Matrix Multiplication).

$$F(x)G(x) = \begin{pmatrix} fg & fg' + f'g \\ 0 & fg \end{pmatrix},$$

i.e., matrix multiplication encodes the Leibniz rule automatically.

Proof. Direct computation. The upper-right entry is $(fg)' = fg' + f'g$. \square

The matrix embedding of Definition 2 is standard in the automatic differentiation literature [3] and provides the computational basis for the compression step in Section III-C.

TABLE I
PIPELINE SUMMARY

Step	Input	Output	Cost
1	APK file	Byte sequence $\mathbf{b} \in \{0, \dots, 255\}^N$	$O(N)$
2	\mathbf{b}	Power spectrum $S[k], k = 0, \dots, \lfloor N/2 \rfloor$	$O(N \log N)$
3	$S[k]$	Polynomial $P(x) = \sum a_i x^i \in \Pi_m$	$O(Nm)$
4	P	Feature vector $\phi_k \in \mathbb{R}^{m-k+1}$	$O(km)$

III. PROPOSED METHOD

A. Signal Representation and Power Spectrum

Extract DEX bytecode from an APK as a discrete signal $\mathbf{b} = (b_0, \dots, b_{N-1}) \in \{0, \dots, 255\}^N$. Define the *normalized power spectrum*:

$$S[k] = \frac{1}{N} \left| \sum_{n=0}^{N-1} b_n e^{-j2\pi kn/N} \right|^2, \quad k = 0, \dots, \lfloor N/2 \rfloor.$$

By Parseval's identity applied over the full spectrum $k = 0, \dots, N-1$: $\sum_{k=0}^{N-1} S[k] = N^{-1} \sum_{n=0}^{N-1} b_n^2$. For real-valued \mathbf{b} , the spectrum is Hermitian-symmetric ($S[k] = S[N-k]$), so the $\lfloor N/2 \rfloor + 1$ non-redundant values satisfy $2 \sum_{k=1}^{\lfloor N/2 \rfloor} S[k] + S[0] \approx N^{-1} \sum_n b_n^2$.

Remark 1. Analysis is restricted to the DEX code section, excluding resource files, embedded images, and encrypted asset blobs, to reduce contamination from high-entropy benign data (see Section VI).

Table I summarizes the full pipeline.

B. Polynomial Approximation

Map $k \mapsto x = 2k/N \in [0, 1]$ and fit a degree- m polynomial:

$$P(x) = \sum_{i=0}^m a_i x^i = \arg \min_{P \in \Pi_m} \sum_k (P(2k/N) - S[k])^2.$$

C. Dual-Number Differential Compression

We represent the polynomial P and its derivatives using the dual-number matrix embedding (Definition 2), which provides an efficient route to repeated differentiation.

Theorem 2 (Feature Vector after k -fold Differentiation). *Setting $c_j = a_{j+k} \cdot (j+k)!/j!$ for $j = 0, \dots, m-k$,*

$$\phi_k = (c_0, \dots, c_{m-k}) \in \mathbb{R}^{m-k+1}.$$

Each differentiation step reduces feature dimension by exactly one.

Proof. Differentiating $a_i x^i$ exactly k times gives $a_i \cdot i!/(i-k)! \cdot x^{i-k}$ for $i \geq k$ and zero otherwise. Re-index with $j = i - k$. \square

IV. THEORETICAL ANALYSIS

A. Computational Complexity

Theorem 3 (Complexity). *Computing ϕ_k from the coefficient vector $\mathbf{a} = (a_0, \dots, a_m)$ requires $O(k \cdot m)$ operations. Classical symbolic differentiation applied to a product-form expression of m linear factors requires $O(2^k \cdot m)$ operations. The full pipeline runs in $O(N \log N)$.*

Proof. From Theorem 2, each differentiation step maps $(a_0, \dots, a_{m-\ell})$ to $(a_1 \cdot 1, a_2 \cdot 2, \dots)$, requiring $m - \ell$ multiplications. Summing over $\ell = 0, \dots, k - 1$:

$$\sum_{\ell=0}^{k-1} (m - \ell) = km - \frac{k(k-1)}{2} \leq km = O(k \cdot m).$$

For classical symbolic differentiation: expressing P as a product of m linear factors and applying the product rule at each step doubles the number of terms, yielding $O(2^k)$ terms after k steps and $O(2^k \cdot m)$ total operations. \square

Corollary 4. *Total complexity is $O(N \log N + N \cdot m + k \cdot m) = O(N \log N)$, dominated by the FFT stage.*

B. Permutation Invariance

Proposition 5 (Permutation Invariance). *For any permutation \mathbf{b}' of \mathbf{b} : $S'[0] = S[0]$ and $\sum_k S'[k] = \sum_k S[k]$. Under the i.i.d. model, $\mathbb{E}[\|\phi_k(\mathbf{b}')\|_2] = \mathbb{E}[\|\phi_k(\mathbf{b})\|_2]$.*

Proof. Both $S[0]$ and $\sum_k S[k]$ depend only on the multiset $\{b_n\}$. The second claim follows since a random permutation preserves the marginal distribution. \square

V. COMPLEMENTARY DISCRIMINATIVE POWER

A. Graph-Preserving Obfuscation

Definition 3 (Graph-Preserving Obfuscation). *A byte-level transform τ is graph-preserving if the API call graph, CFG, and permission set extracted from $\tau(\mathbf{b})$ are identical to those from \mathbf{b} .*

Examples: payload encryption with a fixed key, byte-level padding, and code transposition that preserves inter-procedural edges.

B. Entropy and Spectral Flatness

Definition 4 (Byte Entropy). $H(\mathbf{b}) = -\sum_{v=0}^{255} p_v \log_2 p_v \in [0, 8]$, where $p_v = |\{n : b_n = v\}|/N$.

Lemma 6 (Spectral Flatness under High Entropy). *If b_0, \dots, b_{N-1} are i.i.d. taking values in $\{0, \dots, 255\}$ with distribution $(p_v)_{v=0}^{255}$, mean μ , and variance σ^2 , then $\mathbb{E}[S[k]] = \sigma^2$ for all $k \neq 0$ (independent of k). Moreover, among all distributions on $\{0, \dots, 255\}$ with fixed mean μ , $\sigma^2 = \sum_v p_v (v - \mu)^2$ is a strictly increasing function of $H(\mathbf{b})$, and is uniquely maximized by the uniform distribution.*

Proof. Flatness. For $k \neq 0$, orthogonality of complex exponentials gives $\mathbb{E}[X[k]] = \sum_n \mathbb{E}[b_n] e^{-j2\pi kn/N} = \mu \sum_n e^{-j2\pi kn/N} = 0$. By independence of $\{b_n\}$:

$\mathbb{E}[|X[k]|^2] = \sum_n \text{Var}(b_n) = N\sigma^2$, so $\mathbb{E}[S[k]] = \sigma^2$, independent of k .

Monotonicity. Fix support $\{0, \dots, 255\}$ and mean $\mu = \sum_v p_v v$. We show that σ^2 is strictly increasing in H by contradiction. Suppose two distributions \mathbf{p} and \mathbf{q} satisfy $H(\mathbf{p}) > H(\mathbf{q})$ but $\sigma^2(\mathbf{p}) \leq \sigma^2(\mathbf{q})$. Since the uniform distribution \mathbf{u} uniquely maximizes both H and σ^2 on $\{0, \dots, 255\}$ (the latter by Jensen's inequality: $\sigma^2 = \mathbb{E}[b^2] - \mu^2 \leq \frac{1}{256} \sum_v v^2 - \mu^2$, with equality iff $\mathbf{p} = \mathbf{u}$), any deviation from \mathbf{u} strictly decreases σ^2 . The strict monotonicity of $H \mapsto \sigma^2$ on the path from any \mathbf{p} toward \mathbf{u} (e.g., via the mixture $(1-t)\mathbf{p} + t\mathbf{u}$, $t \in [0, 1]$) follows from strict concavity of H and strict convexity of $-\sigma^2$ along this path [7], giving the required contradiction. \square

Lemma 7 (Coefficient Bound under Near-Flat Spectrum). *If $S[k] = C + \varepsilon[k]$ with $\|\varepsilon\|_2 \leq \delta$, and V is the Vandermonde matrix $V_{k,i} = (2k/N)^i$, then $\|\mathbf{a}^* - C\mathbf{e}_0\|_2 \leq \delta/\sigma_{\min}(V)$, and in particular $|\alpha_i^*| \leq \delta/\sigma_{\min}(V)$ for all $i \geq 1$.*

Proof. $\mathbf{a}^* = V^\dagger \mathbf{S} = C\mathbf{e}_0 + V^\dagger \varepsilon$, using $V^\dagger \mathbf{1} = \mathbf{e}_0$. Taking norms gives the bound. \square

C. Main Theorem

Theorem 8 (Complementary Discriminative Power). *Let $\mathcal{F}_{\text{struct}}$ be any classifier depending solely on structural features. Let τ be a graph-preserving obfuscation.*

- (i) **Structural blindness.** $\mathcal{F}_{\text{struct}}(\mathbf{b}) = \mathcal{F}_{\text{struct}}(\tau(\mathbf{b}))$ for all \mathbf{b} and graph-preserving τ .
- (ii) **Spectral sensitivity.** If τ changes byte entropy by $\Delta H > 0$, then under the i.i.d. model, for sufficiently large N ,

$$|\mathbb{E}[\|\phi_k(\tau(\mathbf{b}))\|_2] - \mathbb{E}[\|\phi_k(\mathbf{b})\|_2]| > 0.$$

- (iii) **Information gain.** Under (i) and (ii), $I(\phi_k; y \mid \mathcal{F}_{\text{struct}}) > 0$, where $y \in \{\text{malware}, \text{benign}\}$.

Proof. (i) By Definition 3, τ leaves the API call graph, CFG, and permissions unchanged, so all inputs to $\mathcal{F}_{\text{struct}}$ are identical for \mathbf{b} and $\tau(\mathbf{b})$.

(ii) Let σ_0^2 and σ_1^2 denote the byte variances of \mathbf{b} and $\tau(\mathbf{b})$ respectively. By Lemma 6, $\mathbb{E}[S_{\mathbf{b}}[k]] = \sigma_0^2$ and $\mathbb{E}[S_{\tau(\mathbf{b})}[k]] = \sigma_1^2$ for all $k \neq 0$. Since σ^2 is strictly monotone in H (Lemma 6) and $\Delta H > 0$, we have $\sigma_1^2 \neq \sigma_0^2$. By the Law of Large Numbers, $S[k] \xrightarrow{p} \sigma_i^2$ for each fixed k as $N \rightarrow \infty$, so the spectral deviation $D(\mathbf{S}) = \|\mathbf{S} - \bar{\mathbf{S}}\mathbf{1}\|_2$ converges to different limits for the two inputs. Via Lemma 7 and Theorem 2, this gap propagates to a nonzero difference in expected feature norms for sufficiently large N .

(iii) Consider the label $y \in \{\text{malware}, \text{benign}\}$ for any sample that includes both pre- and post-obfuscation variants. From (i), $\mathcal{F}_{\text{struct}}$ assigns identical scores to \mathbf{b} and $\tau(\mathbf{b})$, so it is conditionally independent of any label variation attributable to τ : $I(\mathcal{F}_{\text{struct}}; y_\tau) = 0$, where y_τ denotes the label component induced by the obfuscation. From (ii), $\mathbb{E}[\|\phi_k(\tau(\mathbf{b}))\|_2] \neq \mathbb{E}[\|\phi_k(\mathbf{b})\|_2]$, so ϕ_k and y_τ are not conditionally independent. Since y_τ is a component of y , this yields $I(\phi_k; y \mid \mathcal{F}_{\text{struct}}) \geq I(\phi_k; y_\tau \mid \mathcal{F}_{\text{struct}}) > 0$. \square

TABLE II
COMPUTATIONAL COST OF FEATURE EXTRACTION

Method	Feature type	Complexity
Drebin [9]	Permissions + API (bag)	$O(N)$
MaMaDroid [8]	API call Markov chain	$O(N \log N)$
CFG-based methods	Control-flow graph	$O(N^2)$
Ours (ϕ_k)	Spectral polynomial	$O(N \log N)$

Remark 2. *Theorem 8 does not claim high standalone precision for spectral features. It claims only that they carry information provably absent from any structural detector—the formal justification for fusion.*

VI. LIMITATIONS

a) **L1. The i.i.d. assumption is approximate.**: Real malware payloads contain loaders and stub code that introduce local structure, so spectra are not perfectly flat. The entropy-based separation is a tendency, not an absolute dichotomy.

b) **L2. High-entropy benign components (primary limitation).**: Even restricted to the DEX code section, benign APKs routinely contain high-entropy content: ProGuard/DexGuard-obfuscated code, cryptographic routines, and compressed string tables. These produce flat spectra indistinguishable from packed malware, limiting standalone precision. This is the primary reason spectral features must be *combined* with structural features rather than used alone.

c) **L3. Loss of sequential structure.**: The DFT discards positional information; API call sequences and execution-order dependencies are invisible to spectral analysis.

d) **L4. Polynomial approximation error.**: Real power spectra are not smooth; a low-degree polynomial fit may introduce significant approximation error. The bias–variance trade-off for (m, k) is an open problem.

VII. FEATURE FUSION FRAMEWORK

Given the limitations above, we propose the following fusion architecture:

- 1) **Structural component.** Any existing structural detector: MaMaDroid [8], Drebin [9], APIGraph [10], etc.
- 2) **Spectral component.** The vector ϕ_k computed on the DEX code section only.
- 3) **Fusion.** Concatenate the two feature vectors and train a joint classifier (SVM, gradient boosting, or late-fusion neural network).

a) **Theoretical justification.**: By Theorem 8(iii), ϕ_k contributes positive mutual information with the label *conditional* on any structural feature, so adding ϕ_k to a structural feature vector cannot decrease and will generically increase classification performance.

b) **Computational cost.**: Computing ϕ_k costs $O(N \log N)$, negligible relative to CFG or API call graph extraction ($O(N^2)$ worst case). Table II compares the overhead of the spectral component against representative structural detectors.

VIII. RELATED WORK

Nataraj et al. [5] visualize malware binaries as grayscale images, implicitly exploiting frequency-like texture features. Raff et al. [6] apply byte-level CNNs, learning spectral filters from data. Both treat frequency information as a standalone signal; our work provides a formal argument for why such features *complement* structural ones. Lyda & Hamrock [4] use raw byte entropy for packed-malware detection, motivating Lemma 6. Unlike their threshold-based approach, our framework provides a compressed, multi-dimensional spectral representation with a formal information-gain guarantee. MaMaDroid [8], Drebin [9], and APIGraph [10] are representative structural detectors; Theorem 8 provides a formal complementarity guarantee with respect to any of them.

The differentiation engine in Section III-C is grounded in forward-mode automatic differentiation via dual numbers, a technique surveyed in [2] and formalized in [3]. The specific application of dual-number differentiation to polynomial feature compression for malware analysis is, to our knowledge, novel.

IX. DISCUSSION

a) On the choice of m and k . The degree m controls spectral approximation fidelity; k controls feature dimensionality ($\phi_k \in \mathbb{R}^{m-k+1}$). Setting $k = m$ reduces to a scalar; $k = 0$ retains the full polynomial. In practice, k should maximize class separability on a validation set.

b) Relationship to STFT and wavelet features. The global DFT discards temporal locality. STFT or wavelet decompositions retain positional information at higher dimensionality cost; the polynomial compression step applies directly to either representation.

c) Adversarial considerations. An adversary could try to craft a high-entropy payload mimicking a benign spectrum. However, controlling global byte frequencies while preserving malicious functionality is non-trivial, and becomes harder when the spectral component is fused with structural features.

X. OPEN PROBLEMS

- 1) **Relaxing i.i.d. assumption.** Extend Lemma 6 to weakly-dependent (α -mixing) processes to better model real malware byte sequences.
- 2) **High-entropy benign filter.** A manifest-based section classifier that excludes encrypted resources and cryptographic library code before spectral analysis would reduce the entropy overlap between benign and malware DEX sections, addressing Limitation L2.
- 3) **Optimal (m, k) selection.** Derive a minimum description length (MDL) criterion for jointly selecting polynomial degree m and differentiation order k .
- 4) **Experimental validation.** Evaluate the AUC and F1 gain from adding ϕ_k to MaMaDroid, Drebin, and APIGraph on the Drebin, AndroZoo, and MalGenome benchmarks.
- 5) **Extension to STFT and wavelet representations.** Replacing the global DFT with a Short-Time Fourier

Transform or wavelet decomposition would restore positional information at the cost of higher dimensionality; the compression step applies directly to any frequency representation.

- 6) **Adversarial robustness.** Characterize the minimum byte-level perturbation cost to drive $\|\phi_k\|_2$ below a detection threshold while preserving malicious functionality.

XI. CONCLUSION

We have proposed and formalized a spectral feature extraction pipeline for Android malware detection, positioned explicitly as a *complementary* component to structural detectors. The pipeline maps DEX bytecode through FFT, polynomial approximation, and dual-number differential compression to produce a compact feature vector ϕ_k at $O(N \log N)$ cost.

The Complementary Discriminative Power Theorem proves that spectral features carry information provably absent from any structural detector: under graph-preserving obfuscation, structural features are blind by construction while spectral features remain sensitive to entropy changes. This provides a formal, rather than empirical, justification for feature fusion.

The framework's limitations—approximate i.i.d. model, high-entropy benign components, loss of sequential structure—are stated explicitly. These are exactly the conditions under which complementary structural features are needed, reinforcing rather than undermining the fusion argument.

More broadly, this work demonstrates that the formal apparatus of information theory and automatic differentiation can be brought to bear on the feature-design problem in malware detection, yielding guarantees that purely empirical approaches cannot provide. We hope the complementarity framework introduced here will serve as a template for rigorous analysis of other feature combinations in the security domain.

REFERENCES

- [1] W. K. Clifford, "Preliminary sketch of biquaternions," *Proc. London Math. Soc.*, vol. 4, pp. 381–395, 1873.
- [2] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," *J. Mach. Learn. Res.*, vol. 18, no. 153, pp. 1–43, 2018.
- [3] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed. SIAM, 2008.
- [4] R. Lyda and J. Hamrock, "Using entropy analysis to find encrypted and packed malware," *IEEE Secur. Privacy*, vol. 5, no. 2, pp. 40–45, 2007.
- [5] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. VizSec*, 2011.
- [6] E. Raff et al., "Malware detection by eating a whole EXE," in *AAAI Workshop AI Cyber Secur.*, 2018.
- [7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley, 2006.
- [8] E. Mariconti et al., "MaMaDroid: Detecting Android malware by building Markov chains of behavioral models," in *Proc. NDSS*, 2017.
- [9] D. Arp et al., "Drebin: Effective and explainable detection of Android malware in your pocket," in *Proc. NDSS*, 2014.
- [10] T. Zhang et al., "Enhancing the description-to-behavior fidelity in Android apps with privacy policy," *IEEE Trans. Inf. Forensics Secur.*, 2020.