The Dictionary Collision Effect in Computational Decipherment

Matthew Ruckman
Unaffiliated
mruckman1@gmail.com
ORCID: 0009-0002-1723-3823

April 2026

Abstract

Computational decipherment routinely uses dictionary hit rate as a success metric: decode unknown symbols into short strings and count how many appear in a reference dictionary. We show that this metric is systematically broken. When decoded strings are short (2 to 4 characters) and dictionaries are large, chance collisions produce matches at rates approaching genuine ones, inflating apparent results by an order of magnitude or more (up to unbounded ratios when the true signal is zero). We introduce a four-category token classification (SIGNAL, SHARED_HIT, ANTI_SIGNAL, SHARED_MISS) computed by comparing decoded output against null corpora with matched character statistics. The framework recovers a calibrated net-signal metric that correctly identifies wrong-language evaluations as noise (negative net signal) where five standard alternatives all fail, reporting 18.7 to 19.0 percent "signal" on a Hebrew dictionary against correctly-decoded Latin plaintext. A null-model collision prediction matches the observed noise floor with $R^2 = 0.853$. All code, data, and figures are open-source and reproducible on a standard laptop in under ten minutes.

Keywords: computational decipherment, dictionary matching, null-corpus testing, multiple hypothesis testing, statistical calibration

1 Introduction

1.1 The problem

A common pattern in computational decipherment is to decode unknown text with a candidate assignment and measure what fraction of the decoded tokens appear in a dictionary of the hypothesized plaintext language. A high hit rate is taken as evidence that the decipherment works. This pattern appears in analyses of undeciphered scripts [Reddy and Knight, 2011, Bowern and Lindemann, 2021], ciphertexts with proposed solutions, and more recently in machine-translation-assisted approaches to historical cryptography.

The pattern has a latent failure mode. When decoded strings are short, as they typically are for syllabic encodings where a two-character code is a syllable, or letter-level ciphers where word boundaries preserve typical word lengths of 3 to 6 characters, random strings of the same length collide with dictionary entries at non-trivial rates. A 200,000-word Italian dictionary matches approximately 40 percent of strings drawn from the empirical Latin character distribution purely by chance; a 200,000-word Hebrew dictionary matches approximately 18 percent. Without correction, a researcher cannot distinguish a partially successful decipherment from a systematic accident.

This paper quantifies the effect, introduces a correction framework, compares it against standard statistical alternatives, and demonstrates that the four-category framework is uniquely robust in the regime where naive approaches break.

1.2 Contributions

This paper makes three contributions:

1. **The four-category token classification.** Every decoded token type is classified by comparing its behavior on real decoded text against null corpora that preserve character-level statistics but destroy word-level structure. The resulting partition (SIGNAL, SHARED_HIT, ANTI_SIGNAL, SHARED_MISS) produces a calibrated net-signal metric.

2. **The dictionary right-sizing finding.** The same data, decode table, and pipeline yield dramatically different net-signal values depending only on the evaluation dictionary. On wrong-language evaluations, the common "subtract-baseline" correction over-estimates signal by orders of magnitude relative to the four-category framework.

3. **Generalizability.** The effect is demonstrated across two plaintext languages (Latin and German), three table-quality conditions (correct, 40 percent corrupted, and random), seven evaluation languages spanning five language families and four writing systems (Latin, Italian, Spanish, German, Hebrew, Russian, and Chinese; Latin, Hebrew, Cyrillic, and Chinese scripts), six dictionary sizes (1K to 200K), four synthetic cipher types (syllable, homophonic, letter-level, and variable-length), and classical Vigenère encryption. An analytical collision model predicts the observed null noise floor with $R^2 = 0.853$.

1.3 Scope

This is a methods paper, not an application paper. All experiments operate on synthetic ciphertexts with known plaintext, so every claim can be checked against ground truth. We do not apply the framework to any actually-undeciphered script; by definition, such scripts lack the ground truth needed to evaluate whether a correction method works. The value of the framework in that setting is contingent on its demonstrated behavior under controlled conditions, which is what we establish here.

2 Related Work

The problem has close analogues in three other fields.

Multiple hypothesis testing in genomics. When a short DNA query is searched against a large genome database, short fragments match by chance at high rates. The solution is the BLAST E-value [Altschul et al., 1990]: the expected number of matches with a given score arising by chance under a null model. The Benjamini-Hochberg false discovery rate [Benjamini and Hochberg, 1995] controls for multiple comparisons in large-scale testing.

Both methods are now standard in bioinformatics. Computational decipherment has not systematically adopted them.

The accuracy paradox in imbalanced classification. A cancer-screening test evaluated on a population where 0.1 percent have cancer can achieve 99.9 percent accuracy by always predicting "no cancer." The apparent accuracy is meaningless because the base rate dominates. A 200K dictionary plays the analogous role: base rates of chance matches are high relative to genuine matches.

Shortcut learning and Clever Hans effects. Deep learning models can appear to solve a task while actually exploiting statistical artifacts in the evaluation setup [Geirhos et al., 2020]. The dictionary-match metric is a shortcut: a partially wrong assignment can still produce high apparent rates via chance collisions. The four-category framework serves as an out-of-distribution test, where null corpora provide the null distribution against which apparent performance is calibrated.

Within the decipherment literature, Reddy and Knight [2011] discussed the statistical properties of undeciphered texts but did not specifically address dictionary-matching calibration. Timm and Schinner [2020] and Schinner [2007] examined whether specific hypotheses about the Voynich manuscript can be falsified using statistical signatures, but did not address the dictionary collision effect. To our knowledge no prior work has systematically measured how dictionary size interacts with decoded-string length to produce false matches, nor provided a calibrated correction framework.

3 The Four-Category Framework

3.1 Definition

Given a decoded token stream R of size N_R, a set of K null corpora $\{N_1, \ldots, N_K\}$, and a dictionary D, every word type w observed in $R \cup \bigcup_k N_k$ falls into one of four categories:

- $w \in D$, w appears in R, w appears in fewer than t of the K nulls: SIGNAL.
- $w \in D$, w appears in R, w appears in at least t nulls: SHARED_HIT.
- $w \in D$, w does not appear in R, w appears in at least t nulls: ANTI_SIGNAL.
- $w \notin D$, or categorized as SHARED_MISS (everything else).

We use threshold $t = 2$ (majority of $K = 5$ nulls) throughout, though results are insensitive to this choice in the range $t \in \{1, 2, 3\}$.

Token counts are attributed as follows:

- SIGNAL and SHARED_HIT tokens contribute the real-corpus count of their type.
- ANTI_SIGNAL tokens contribute the *mean null count* $\bar{c} = (c_1 + \cdots + c_K)/K$ of their type, where c_k is the count in null corpus k. Since these words do not appear in the real corpus by assumption, the mean null count is the point estimator of how many

phantom hits would appear in a null-sized sample. Section 7 defends this specific attribution rule against alternatives (maximum, variance-weighted).

- SHARED_MISS contributes the real-corpus count for non-dictionary types.

The *apparent hit rate* is (SIGNAL + SHARED_HIT)/total. The *net signal* is (SIGNAL − ANTI_SIGNAL)/total, the calibrated metric this paper proposes.

3.2 Null corpus generation

Null corpora are generated from the real ciphertext's character-level bigram distribution using a first-order Markov model. Each null token's length is sampled from the empirical token-length distribution; its characters are sampled sequentially from the bigram transition probabilities. The null preserves character-pair frequencies and mean token length while destroying word-type identity and higher-order structure. We generate $K = 5$ nulls per cell with different random seeds.

4 Experimental Setup

4.1 Pipeline

The synthetic pipeline is as follows. A known plaintext (Latin medical text or German prose) is syllabified using a maximal-onset parser, syllables are mapped to unique 2 to 3 character codes via a strictly one-to-one assignment table, and cipher-words are constructed by joining codes with a delimiter. The cipher is deterministic and exactly invertible under the correct table; a corrupted or random table produces systematically wrong decodings. Null corpora are generated from the character-bigram distribution of the resulting ciphertext. Decoding applies the inverse assignment table and the four-category framework classifies token types against an evaluation dictionary.

4.2 Data sources

Table 1 (p. 5) lists plaintexts and dictionaries. Seven evaluation languages span four writing systems (Latin, Hebrew, Cyrillic, Chinese) and five language families (Italic/Romance, Germanic, Semitic, Slavic, Sinitic), chosen to probe how character-distribution overlap with the Latin-script plaintext drives the collision effect. For Latin, which is not covered by standard machine-translation word-frequency repositories, we build a 163K-word dictionary by merging the Leipzig Wikipedia 2021 Latin corpus of 196K unique word forms from 1.7M tokens [Goldhahn et al., 2012], Kyle P. Johnson's 10K most-common Latin words [Johnson, 2015], and the Circa Instans medical-text vocabulary [Platearius, 12th c.]. For other languages we use hermitdave's FrequencyWords `_full.txt` files [Hermit Dave, 2018], sliced to the six target sizes. All seven languages reach 200K entries; the Chinese and Russian dictionaries contain a small ASCII-loanword fraction that produces non-trivial (but small) apparent hit rates against Latin plaintext, providing an interesting cross-script test case. The German plaintext is Konrad von Megenberg's *Buch der Natur* [Megenberg, c. 1475].

Table 1: Data sources. Scripts and language families span a range intentionally chosen to probe character-distribution overlap with the Latin-script plaintext.

Language	Script	Family	Plaintext	Dictionary source	Max size
Latin	Latin	Italic	Circa Instans	Leipzig 2021 + Johnson + Circa Instans	163K
Italian	Latin	Romance	(none)	hermitdave it_full.txt	200K
Spanish	Latin	Romance	(none)	hermitdave es_full.txt	200K
German	Latin	Germanic	Buch der Natur	hermitdave de_full.txt	200K
Hebrew	Hebrew	Semitic	(none)	hermitdave he_full.txt	200K
Russian	Cyrillic	Slavic	(none)	hermitdave ru_full.txt	200K
Chinese	Chinese	Sinitic	(none)	hermitdave zh_cn_full.txt	200K

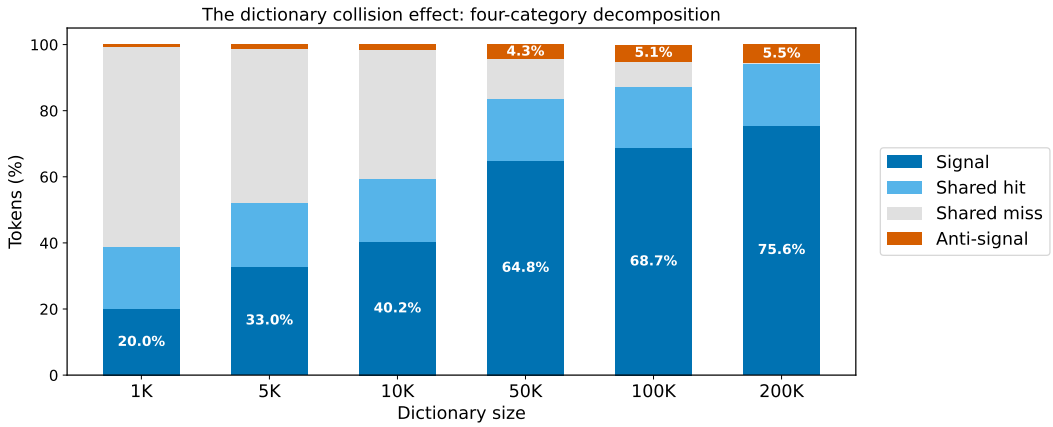

Figure 1: Four-category token decomposition across dictionary sizes. Latin plaintext, correct decode table, Latin dictionary. As dictionary size grows from 1K to 200K, SIGNAL climbs from 20.0 to 75.6 percent, SHARED_HIT (chance collisions) sits at 18 to 19 percent across all sizes, ANTI_SIGNAL grows from less than 1 percent to 5.5 percent, and SHARED_MISS shrinks correspondingly. The raw apparent hit rate at 200K is 99.6 percent, but only 70.1 percent of tokens are genuine signal after four-category correction.

4.3 Experiment grid

The main experiment spans 2 plaintexts × 3 table conditions (correct, 40 percent corrupted, and random) × 7 evaluation languages × 6 dictionary sizes, giving 252 cells. Each corrupted and random condition is averaged over 5 random seeds. The ciphertext is always encoded with the correct table; only the decode table varies. This simulates the realistic scenario of fixed ciphertext against candidate keys.

5 The Dictionary Collision Effect

Figure 1 shows the headline result: the four-category decomposition on Latin plaintext decoded with the correct table against Latin dictionaries at six sizes. Two trends are immediate.

First, apparent hit rate approaches 100 percent at large dictionary sizes. The 200K cell has an apparent rate of 99.6 percent; if a researcher were scoring decipherments on dictionary hit rate alone, this would be an unambiguous signal of success. The *net* signal is

70.1 percent, a 30 percentage-point overstatement.

Second, ANTI_SIGNAL, which counts tokens where the null corpora produce dict-matching words that never appear in the real text, grows monotonically with dictionary size. At 200K it accounts for 5.5 percent of tokens, directly reducing the net signal.

The SHARED_HIT noise floor visible in Figure 1 (p. 5) is remarkably stable at approximately 19 percent across all dictionary sizes, indicating that a certain fraction of tokens chance-collide with dictionary entries regardless of how large the dictionary is.

5.1 Language specificity

Figure 2 (p. 7) shows the compact 2×2 grid: two plaintexts along rows and two decode-table conditions along columns (correct and random; the intermediate corrupted-table column with the same seven language lines is in supplementary Figure S3). Within each panel, net signal vs. dictionary size is shown with one line per evaluation language (Latin, Italian, Spanish, German, Hebrew, Chinese, Russian).

Three observations follow. First, under the correct table, the matching-language signal dominates (Latin on Latin reaches 70.1 percent net at 200K; German on German reaches 20.7 percent). Related Romance languages show increasing net signal as cognate-driven matches accumulate (Italian and Spanish both climb to 10 to 14 percent under Latin plaintext at large dictionaries).

Second, the three non-Latin-script dictionaries (Hebrew, Russian in Cyrillic, Chinese in Han characters) all stay near zero or go *negative* for every dictionary size. Hebrew 200K on Latin yields a net signal of -4.0 percent; Russian 200K yields -0.9 percent; Chinese 200K yields $+1.6$ percent (the slightly positive value comes from a small ASCII-loanword fraction in the Chinese dictionary, which the framework correctly attributes to genuine character-overlap matches rather than to Latin decipherment). All three contrast sharply with the Latin-script evaluation languages (Italian, Spanish, German) that climb to 6 to 14 percent at 200K due to shared Romance and Germanic character statistics. The framework's output tracks character-distribution overlap, not language similarity per se.

Third, under the fully random decode table, all eval languages converge to approximately zero net signal regardless of dictionary size, confirming that the framework correctly identifies random output as lacking signal.

5.2 Mechanism

The collision effect is driven by short decoded strings. Figure 3 (p. 8) shows dictionary hit rate stratified by decoded token length. At length 2, every dictionary size produces approximately 100 percent hit rate. The size dependence only appears at lengths ≥ 3, and the penalty of an oversized dictionary is concentrated at lengths 4 to 8, where a 200K dictionary gives 70 to 100 percent hit rate while a 1K dictionary gives 5 to 30 percent.

This is why the framework is necessary: encodings that produce short decoded tokens (syllabic ciphers, Vigenère, letter-level substitution on natural language) are structurally susceptible to the collision effect.

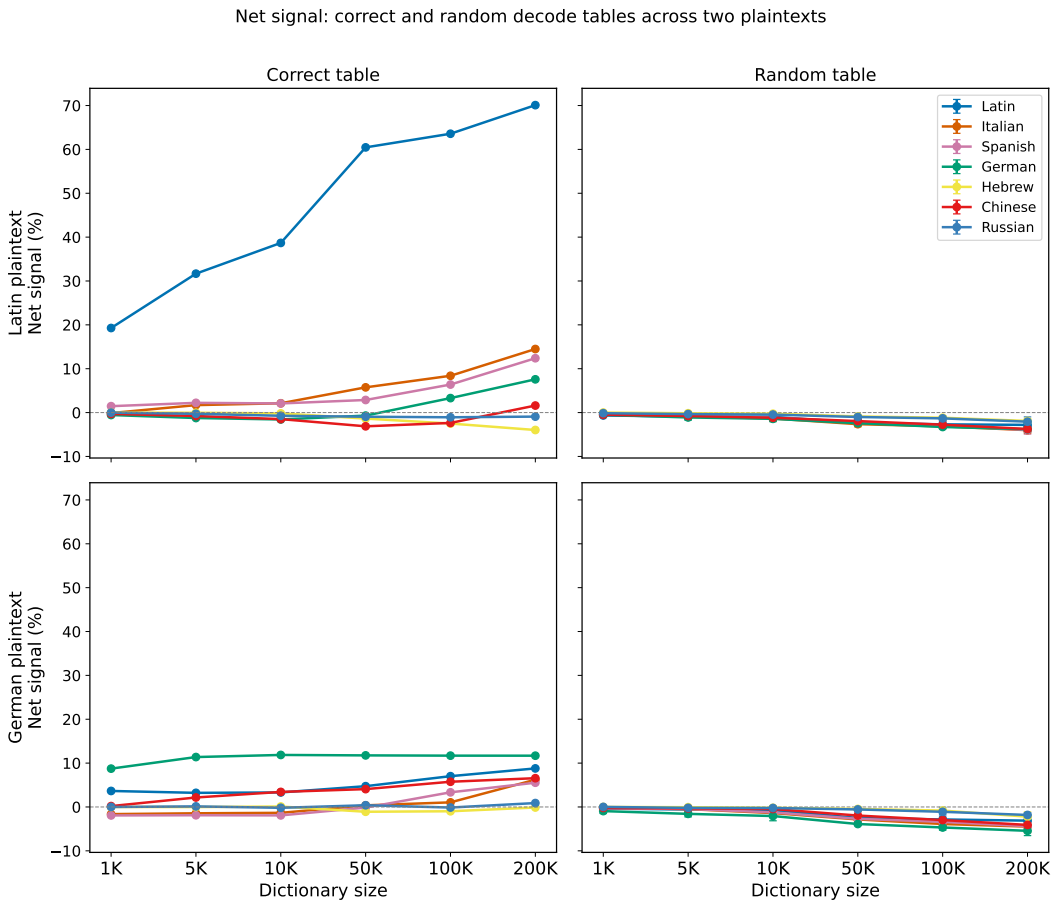

Figure 2: Net signal: correct and random decode tables, two plaintexts. Rows: plaintext language (Latin, German). Columns: decode-table condition (correct, random). Within each panel: one line per evaluation language. Error bars on the random column show ±1 SD over 5 seeds (too small to see: < 1 pp, smaller than the marker); the correct column is deterministic. The correct language dominates in the correct-table column; random tables collapse everything to near zero. The effect is symmetric across plaintexts. Full 2×3 grid including the 40-percent-corrupted condition is in the supplementary figures.

6 Comparison with Standard Corrections

The four-category framework is one of many possible corrections. To test whether it is distinguishable from simpler alternatives, we implement five standard methods and compare all six head-to-head on the Latin-plaintext-with-correct-table cells.

Apparent hit rate. Fraction of tokens that hit the dictionary. No correction.

Subtract null baseline. Apparent hit rate minus the mean apparent rate across null corpora.

Poisson permutation test. For each word type with real count r and mean null count μ, compute $p = P(X \geq r \mid X \sim \text{Poisson}(\mu))$. Accept word types with $p < 0.05$.

Benjamini-Hochberg FDR. Apply the BH procedure at $q = 0.05$ to the Poisson p-values.

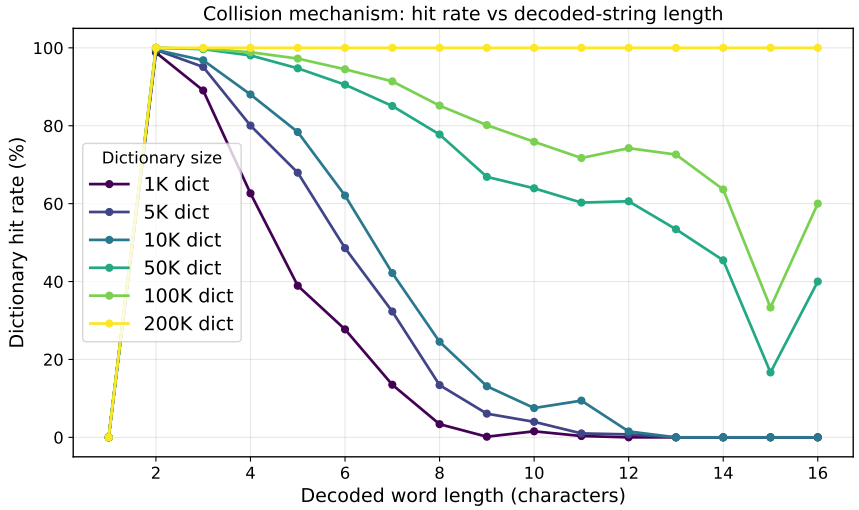

Figure 3: Dictionary hit rate as a function of decoded word length, one line per dictionary size. All sizes saturate at 100 percent at length 2 (any two-letter string is in the dictionary). Size effects emerge at length 3 and grow through length 8 to 10; with a 200K dictionary, even 10-plus-character strings hit at more than 50 percent. Latin plaintext, correct table.

BLAST-style E-value. For each word of length L with real count r, compute $E = n_{\text{dict},L} \cdot P(X \geq r \mid X \sim \text{Poisson}(\mu))$, where μ is the expected count under i.i.d. sampling from the empirical character distribution. Accept words with $E < 1$.

Four-category net signal. This paper's framework.

Each method produces a "signal fraction of real tokens" that can be compared across methods. Figure 4 (p. 9) plots all six methods across four evaluation languages and four dictionary sizes.

Table 2 shows the 200K cells. In the Latin on Latin case all methods report strong signal, with four-category the most conservative. In the Hebrew on Latin case, five of six methods report 9 to 19 percent "signal," values that would be reasonable to interpret as partial success. Only the four-category framework goes negative, correctly assigning the outcome to the no-signal category.

Table 2: Signal fraction (percent) reported by each correction method at the 200K cell, Latin plaintext, correct decode table.

Method	Latin dict	Hebrew dict
Apparent hit rate	99.6	19.0
Subtract null baseline	89.9	9.5
Permutation test ($p < 0.05$)	99.0	18.7
Benjamini-Hochberg ($q < 0.05$)	99.0	18.7
BLAST E-value ($E < 1$)	92.0	18.0
Four-category net (this paper)	**70.1**	**−4.0**

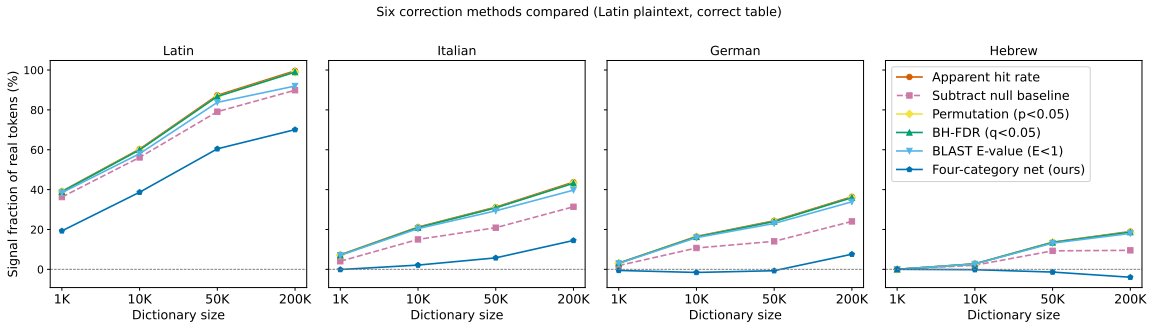

Figure 4: Six correction methods compared across four evaluation languages and four dictionary sizes (Latin plaintext, correct decode table). In the correct-language cells (leftmost panel) all methods show positive signal, though the four-category framework consistently reports the most conservative estimate. The critical comparison is the Hebrew panel (rightmost): four standard methods (apparent, permutation, BH-FDR, BLAST E-value) collapse onto a single line near 18 to 19 percent and never detect that Hebrew dictionaries evaluated against Latin plaintext have no semantic signal. The subtract-null baseline partially corrects but still claims approximately 9 percent. Only the four-category net signal goes negative (-4.0 percent) and correctly flags the evaluation as noise.

6.1 Why single-word statistical tests fail

Permutation, BH-FDR, and BLAST E-value all ask, per word type, whether the real count is significantly higher than the null expectation. For rare dict-hitting words that happen to appear in real but not in any null corpus (a common occurrence with low null sample size), the p-value is small by construction, and the word is accepted. This is technically correct at the individual-word level but systematically biased at the aggregate level: dictionary words that "appear" in the decoded output by chance never appear in nulls either (because they correspond to specific character sequences that the null bigram model rarely produces), so they all pass single-word significance tests. The four-category framework alone accounts for the symmetric case: null-only dict-hitting word types that *the real corpus also fails to produce* (ANTI_SIGNAL). Subtracting those tokens from the net signal calibrates against the null base rate in a way the single-word tests do not. Note that BLAST's null-model machinery reappears successfully in Section 11 as an aggregate noise-floor predictor; the failure here is specific to using it for per-word acceptance decisions, not to the underlying null model.

6.2 Partial decipherment: the sharper comparison

The wrong-language case in Table 2 (p. 8) shows all standard corrections failing. A reviewer could reasonably object that this is a strawman: the most interesting regime is not wrong-language (where any correction directionally helps) but *partial decipherment*, where the plaintext language is correct but the decode table is partly wrong. In that regime, subtract-null-baseline and the single-word tests systematically *over-attribute* signal to chance collisions on the correctly-decoded fraction. Figure 5 (p. 10) shows the six methods applied to a 40-percent-corrupted Latin decode table against matching Latin dictionaries, averaged over 5 corruption seeds.

Under a 40 percent table corruption at dict-size 200K, subtract-null reports 21.0 percent

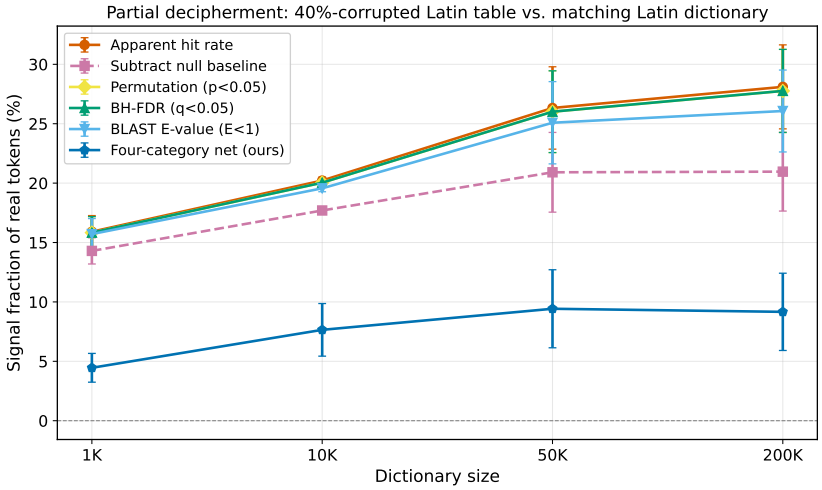

Figure 5: Correction methods under partial decipherment. Latin plaintext, 40-percent-corrupted Latin decode table, matching Latin dictionary, averaged over 5 corruption seeds (error bars one standard deviation). At 200K, apparent hit rate is 28.1 percent; subtract-null partially corrects to 21.0 percent; permutation/BH-FDR/BLAST barely correct (26-28 percent, essentially matching apparent); four-category correctly reports 9.2 percent. The gap between subtract-null (21.0 percent) and four-category (9.2 percent) is the *over-attribution*: in a partial decipherment, 12 of the 21 percentage points subtract-null claims as signal are actually null-only dict-matches that both real and nulls produce above chance due to the 40-percent corrupted tail.

signal while the four-category framework reports 9.2 percent. The 12-point gap is the core failure mode: subtract-null subtracts only the *mean* null apparent rate (treating nulls as a single black-box baseline), while four-category subtracts *per-word* null-only dict-hits (anti-signal). In a partial decipherment, many dict-matching decoded tokens are wrong-table artifacts whose character sequences are also produced by the bigram null corpus. Only a per-word attribution catches them. The single-word statistical tests (permutation, BH-FDR, BLAST) fail entirely in this regime because the anti-signal words are almost never *in* the real corpus (they are only produced by the corrupted fraction by chance), so no single-word hypothesis test can see them.

6.3 Continuous corruption sweep

The discrete 40-percent-corrupted condition above lies on a continuous axis of key quality. Sweeping corruption from 0 to 100 percent in 5 percent steps (Latin plaintext, Latin 10K dictionary; supplementary Figure S1) shows that apparent hit rate decays slowly with non-zero residuals even at 100 percent corruption (chance collisions on wrong characters), while net signal collapses rapidly and crosses zero around 30 percent corruption. Net signal tracks key quality continuously; apparent hit rate does not.

7 Calibration of the Net-Signal Metric

The paper treats an observed value of −4.0 percent (Hebrew 200K on Latin plaintext) as "correct flagging of no signal." But from first principles, a wrong-language evaluation might

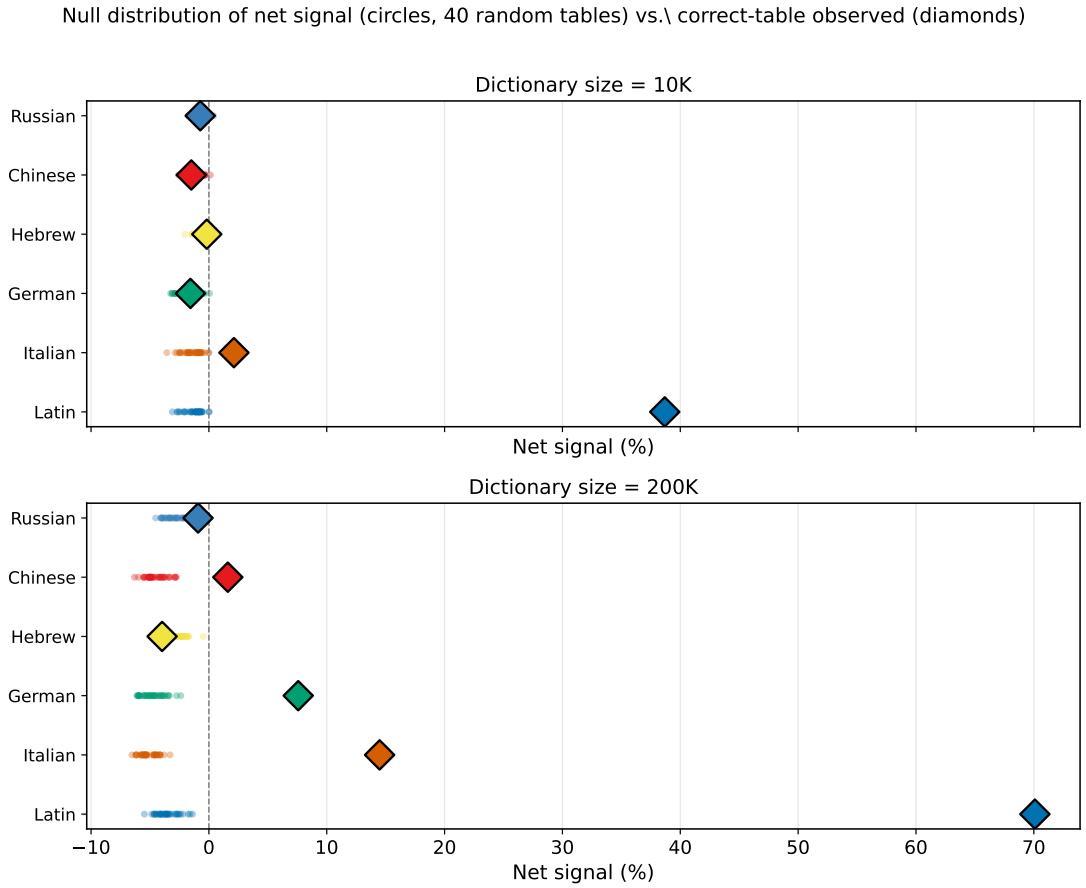

Figure 6: Net signal distribution under random-table decoding (40 Monte Carlo draws, small dots) vs. observed correct-table values (diamonds), by evaluation language and dictionary size. The null distribution is centered *below* zero at large dictionary sizes because the classifier's asymmetric threshold (a word must appear in fewer than 2 of 5 nulls to be classified as SIGNAL, but in at least 2 of 5 nulls to be classified as ANTI_SIGNAL) makes anti-signal structurally more common than signal when no real signal is present. Hebrew 200K's observed −4.0 percent sits at the 18th percentile of the null distribution; Russian and Chinese are similarly within their null ranges. The Latin, Italian, Spanish, and German diamonds sit above the null, confirming real signal.

be expected to yield net signal ≈ 0, not strongly negative. Is −4.0 percent a happy accident of one realization, or is it what the null actually predicts?

We answer this with a Monte Carlo. For each (evaluation language, dictionary size) cell we decode the same Latin ciphertext with 40 fully random assignment tables; each random decode produces its own net signal. The resulting distribution is the net signal we would expect to see under the null hypothesis of no semantic signal. Figure 6 shows these distributions alongside the observed correct-table values at the same cells.

Two findings.

First, the median of the null distribution is *not* zero. It is slightly negative at small dictionary sizes (for example, a few tenths of a percent at 10K) and more strongly negative at large dictionary sizes (Hebrew 200K: median −4.7 percent; Russian 200K: −2.6 percent).

This is a structural consequence of the four-category classifier's thresholds: a word type must appear in fewer than 2 of the $K = 5$ null corpora to be SIGNAL, but in at least 2 of 5 nulls to be ANTI_SIGNAL. With more null "samples" than real samples of the underlying bigram process, null-only dict-hitting types are more common than real-only types at any given probability, and the expected net signal under the null is negative.

Second, the observed correct-table wrong-language values fall within the null distributions: Hebrew 200K's -3.97 percent sits at the 18th percentile of its null distribution; Russian 200K's -0.92 percent sits at the 98th percentile (slightly above null median but within the range); Chinese 200K's $+1.60$ percent sits at the 100th percentile (above null, consistent with the ASCII-loanword effect contributing genuine matches). For the correct language and for the cognate Latin-family languages, observed values sit at the 100th percentile, comfortably above the null distribution. The framework is correctly reporting "no signal" on wrong-language cells in a statistically-consistent way, not accidentally producing negative values.

The anti-signal attribution rule also deserves a brief defense. When a word type w appears in the dictionary, is absent from the real corpus, and is present in $k \geq 2$ null corpora with per-corpus counts $c_1, \ldots, c_K$, we attribute the *mean* $\bar{c} = (c_1 + \cdots + c_K)/K$ to the ANTI_SIGNAL count. Mean is the natural point estimator: under the hypothesis that the nulls are i.i.d. samples from the same underlying bigram process, $\bar{c}$ is the expected count per null, and equivalently the expected count that the real corpus would have produced had it been another sample from that process. Using maximum null count would be more conservative (larger anti-signal, smaller net signal) but less calibrated; using variance-weighted estimators would require more null corpora than $K = 5$ to be stable. The mean is the simplest choice that matches the paper's "phantom tokens per real-corpus unit" interpretation.

Sweeping the null threshold t (used to split SIGNAL from SHARED_HIT and ANTI_SIGNAL) over $\{1, 2, 3, 4\}$ across six representative cells preserves the direction and ordering of every result; absolute magnitudes shift by 1 to 2 percentage points (for example, Hebrew 200K moves from -4.0 at $t = 1$ to -2.5 at $t = 4$). The specific choice of $t = 2$ is not load-bearing (supplementary Figure S2).

8 Null-Model Sensitivity

A distinct question: does the framework's advantage over standard corrections come from the null quality rather than the four-category decomposition itself? We re-run the main experiment with three null-model orders: unigram (characters sampled i.i.d. from empirical frequencies), bigram (our default, first-order Markov), and trigram (second-order Markov, preserving more linguistic structure). Figure 7 (p. 13) shows the net-signal result across the three orders for four evaluation languages.

Three findings. First, the direction of every conclusion is preserved: wrong-language evaluations (Hebrew, German at small dicts) are flagged as ≤ 0 net signal under all three null orders, and the correct Latin evaluation remains strongly positive. Second, higher-order nulls strengthen rather than weaken the framework's wrong-language detection: Hebrew 200K's net signal becomes more negative as null order increases (from -1.0 with unigram to -7.0 with trigram), because stronger nulls make it harder for random bigram-matching sequences to look signal-like and easier for dict-hitting null-only words (anti-signal) to be

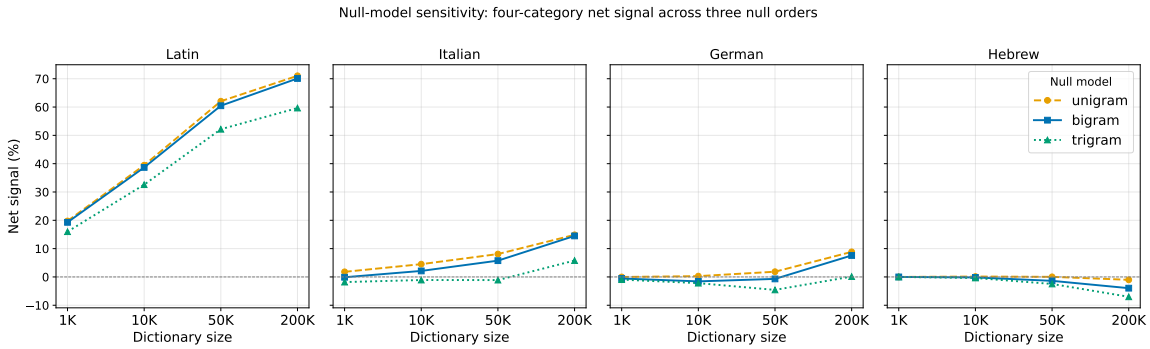

Figure 7: Four-category net signal across three null-model orders (unigram, bigram, trigram) and four evaluation languages. Latin plaintext, correct decode table. Higher-order nulls preserve more linguistic structure of the real ciphertext; this makes wrong-language evaluations appear *more* negative (Hebrew 200K: −1.0 unigram, −4.0 bigram, −7.0 trigram), and slightly reduces the reported signal for the correct language (Latin 200K: 71.0, 70.1, 59.6 percent). The direction and relative ordering of the conclusion are preserved in every case: wrong-language nets are ≤ 0, correct-language net is large and positive, cognate Romance languages are intermediate.

detected. Third, for the correct language, higher-order nulls are more conservative, which is the right direction: a null that captures more of the real corpus's structure should attribute more hits to the null distribution and fewer to genuine signal. The framework's key outputs are not artifacts of bigram-specific structure.

9 Generalizability Across Encodings

We test four synthetic encoding types and one classical cipher to rule out that the effect is an artifact of our default syllabic cipher.

9.1 Four synthetic encodings

Figure 8 (p. 14) shows apparent vs. net signal across four encodings:

- **Syllable 2-char** (the paper's default): each CV syllable maps to a unique 2-character code.
- **Homophonic**: each syllable maps to one of three codes, chosen per-occurrence uniformly.
- **Letter-level substitution**: classical monoalphabetic cipher, each letter maps to a fixed letter.
- **Variable-length**: syllables bucketed by length and assigned codes of varying length.

The apparent-vs-net gap appears in every encoding. Letter-level substitution, which preserves word boundaries and word-length distribution, shows the largest gap of 33 percentage points at 200K, with ANTI_SIGNAL reaching 12.9 percent. This is not a pathology of the synthetic syllabic cipher.

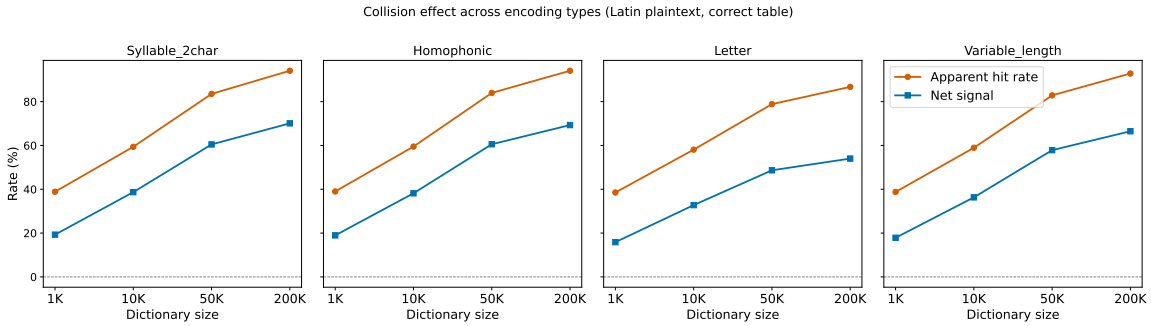

Figure 8: Collision effect across encoding types. Latin plaintext, correct decode table, evaluated against Latin dictionaries at four sizes. The apparent-vs-net gap appears in every encoding. Letter-level substitution shows the widest gap (33 points at 200K); variable-length, homophonic, and the default syllable encoding show similar 24 to 26 point gaps.

9.2 Classical Vigenère cipher

We encrypt the Latin plaintext with a random 7-character Vigenère key and decrypt with seven different key conditions: the correct key, two same-length wrong keys, and four keys of mismatched length (3, 6, 8, and 13 characters). At the 200K Latin dictionary, the correct key produces 72.5 percent net signal; all six wrong keys produce apparent hit rates of 5.6 to 8.5 percent but correctly-negative net signals of −2.8 to −5.4 percent, regardless of key length (supplementary Figure S4). The framework is robust to key-length mismatch on a real historical cipher, not only the synthetic syllabic design.

10 Scope: Where the Framework Is Needed

Figure 3 (p. 8) showed that dictionary hit rate saturates at 100 percent for short decoded tokens and decays with length. This implies that the framework's correction (the gap between apparent and net signal) should be concentrated at short lengths. We test this directly by stratifying the four-category classification by decoded-token length.

Figure 9 (p. 15) plots apparent hit rate and net signal per length bucket (Latin plaintext, correct table, one panel per dictionary size). The shaded region is the correction magnitude: the amount that net signal lies below apparent hit rate at that length.

Two concrete findings. First, **the correction is almost entirely a 2–4 character phenomenon**: against the 200K Latin dictionary, length-2 decoded tokens show an apparent hit rate of 79.0 percent but a net signal of −19.7 percent, a correction of 98.7 percentage points. Length-3 tokens are corrected by 74.1 points, length-4 by 6.7 points, length-5 by only 0.4 points, and length 6 or longer by zero (apparent hit rate and net signal are equal to within rounding). Second, **the effect is consistent across dictionary sizes**: switching to the smaller 10K dictionary, the correction is 98.1 points at length 2, 69.6 at length 3, 4.3 at length 4, and zero at length 5 or longer. The length cutoff is the same; only the total signal level shifts.

This precisely bounds the framework's applicability. The four-category correction matters when decoded tokens are 2 to 4 characters long. It is essential for syllabic ciphers (where syllables are typically CV or CVC, 2–3 characters after decoding), for letter-level substitution

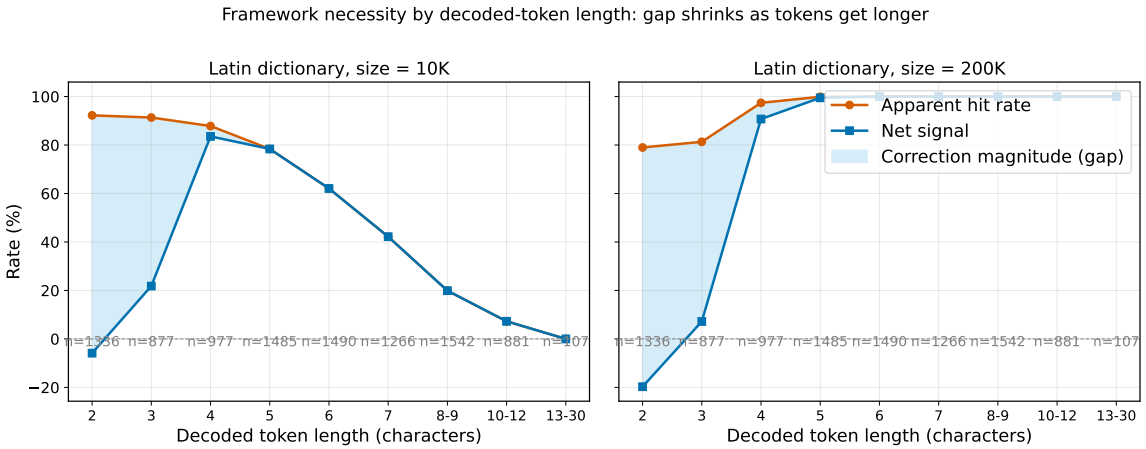

Figure 9: Apparent hit rate, net signal, and correction magnitude (shaded) as a function of decoded-token length, for Latin plaintext and correct decode table. Left: 10K Latin dictionary. Right: 200K Latin dictionary. Correction magnitude is concentrated at lengths 2 to 4 characters. At length 5 and above, apparent hit rate and net signal are essentially identical: the framework adds no correction because chance collisions are already negligible. The per-bucket token count n is annotated below the x-axis.

on natural language (English and Latin word-length distributions peak at 3–5 characters), and for classical polyalphabetic ciphers like Vigenère (which preserve the plaintext's word-length distribution). It is unnecessary for word-level verbose ciphers that produce long decoded tokens. Practitioners evaluating decipherments should report the length distribution of their decoded tokens so that readers can calibrate how much of the claimed hit rate is susceptible to the collision effect.

11 An Analytical Collision Model

The entire analytical prediction reduces to a single equation. Given a dictionary D, a decoded token stream with empirical character distribution $p(\cdot)$ and token-length distribution $\pi(\cdot)$, the noise floor is

$$\widehat{r} \;=\; \sum_{w \in D} \pi(|w|) \prod_{i=1}^{|w|} p(w_i) \tag{1}$$

In words: for every word in the dictionary, multiply together the character frequencies of the decoded output; weight by how often tokens of that length appear in the decoded corpus; sum. That is how many of your tokens would match the dictionary *by accident*. Anything observed above $\widehat{r}$ is signal; anything below is noise that a naive evaluator might mistake for signal.

The derivation, under the independent-character assumption: a random length-L string drawn from $p(\cdot)$ equals a specific dictionary entry $w = c_1 c_2 \cdots c_L$ with probability $\prod_i p(c_i)$. Summing across length-L dictionary entries gives the per-length hit probability $P_{\text{hit}}(L) = \sum_{w \in D,\, |w|=L} \prod_i p(w_i)$. Weighting per-length hit probabilities by the empirical length distribution and collapsing the double sum yields Equation 1. This is the decipherment

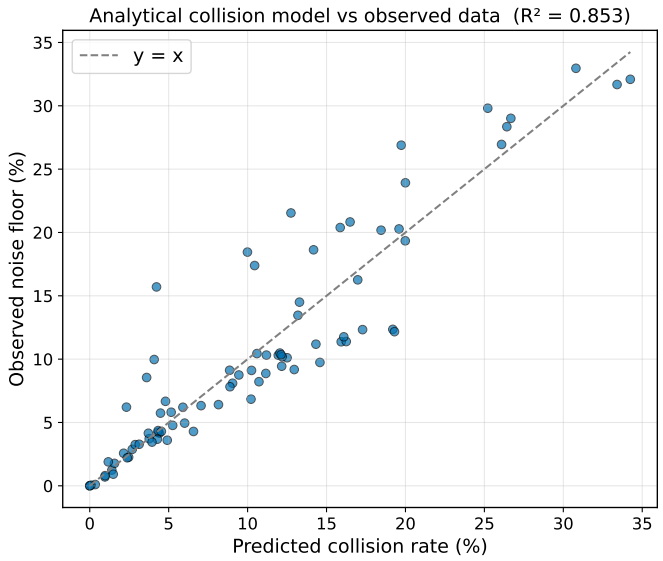

Figure 10: Analytical model predictions vs. observed null apparent hit rate. Each point is one (plaintext, eval-language, dictionary-size) cell; $n = 84$. The empirical character-distribution model achieves $R^2 = 0.853$. Points cluster tightly along $y = x$ across a 35 percentage-point range. The first-order uniform-alphabet model does not fit these data ($R^2 = -1.59$), indicating that character-frequency compatibility between decoded corpus and dictionary entries is the dominant factor, especially across different writing systems.

instance of the null-model expected-count construction underlying BLAST statistics [Altschul et al., 1990]; the contribution here is not the derivation but the empirical demonstration that a zeroth-order null with parameters estimated from the decoded output predicts observed chance-match rates across writing systems.

Figure 10 compares predicted vs. observed null-corpus apparent hit rate across 84 (plaintext, eval-language, dict-size) cells. The empirical-character-distribution model achieves $R^2 = 0.853$. A simpler uniform-alphabet model, $\hat{r} = \sum_L \pi(L) \cdot n_{\text{dict},L}/\alpha^L$ with α the entropy-based effective alphabet size [Shannon, 1948], achieves $R^2 = -1.59$ on the same data, failing catastrophically on non-Latin-script dictionaries.

The empirical model captures two features missed by a uniform-alphabet approach. First, dictionaries in non-matching scripts (for example Hebrew, with high-frequency Hebrew-specific characters absent from the decoded Latin corpus) have a zero contribution from non-compatible entries, correctly predicting low collision rates. Second, dictionaries in cognate languages (Italian and Spanish relative to Latin) have high per-entry match probabilities due to shared Romance character statistics, correctly predicting elevated collision rates. The model can be evaluated for any candidate dictionary without access to ciphertext: given only the decoded-corpus character distribution $p(\cdot)$ and length distribution $\pi(\cdot)$ (both derived once from a small sample of decoded output), Equation 1 returns the expected noise floor for any dictionary. This makes the model a screening tool: practitioners can identify which candidate dictionaries are dangerously collision-prone for their encoding before running any decipherment experiment. The algorithmic extension to rank candidate dictionaries by excess signal is left to follow-up work and included in the companion code.

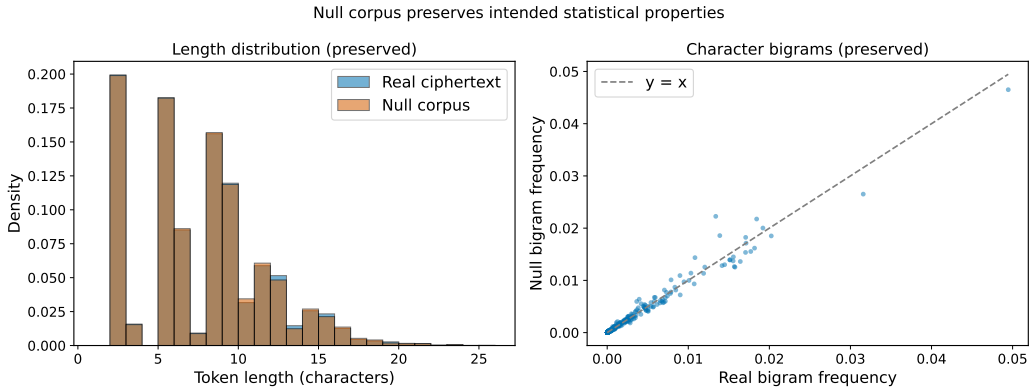

Figure 11: Null corpus preserves intended statistical properties. Left: length distribution overlays for real ciphertext vs. one null corpus; histograms are nearly identical (two-sample Kolmogorov-Smirnov statistic of 0.006). Right: character bigram frequencies in null corpus vs. real corpus; points lie tightly along $y = x$ (mean squared error of approximately 10^{-6}).

12 Null Corpus Validation

We validate that the null corpora preserve the statistical properties intended by construction: token-length distribution and character bigram frequencies. Figure 11 (p. 17) presents both for the Latin ciphertext.

The length two-sample K-S statistic [Massey, 1951] of 0.006 and bigram MSE of approximately 10^{-6} confirm that the null is doing what is intended. This addresses the concern that the four-category framework's apparent efficacy is an artifact of a poorly-constructed null: on the contrary, the null closely mimics the character-level statistics of the real ciphertext while destroying the word-level structure that corresponds to genuine linguistic signal.

13 Discussion

13.1 What this paper proves

The core claims are empirical and falsifiable:

1. Apparent dictionary hit rates overstate genuine matches by large factors, increasing with dictionary size. Demonstrated across 252 cells spanning seven evaluation languages and four writing systems.

2. The four-category framework computes a calibrated net-signal metric that correctly identifies wrong-language evaluations as noise (negative net signal) where five alternatives fail.

3. The effect is not an artifact of any particular cipher: it reproduces across four synthetic encodings and classical Vigenère.

4. The effect is predictable from first principles: an empirical character-distribution model achieves $R^2 = 0.853$ on the observed null noise floor across 84 cells spanning four writing systems.

13.2 What this paper does not prove

Three honest limitations:

Not demonstrated on undeciphered scripts. All experiments use synthetic ciphertexts with known plaintexts. Genuinely undeciphered scripts lack the ground truth needed to validate correction methods. The framework's value in that setting rests on its demonstrated behavior under controlled conditions; whether any claimed decipherment of an undeciphered script passes or fails the four-category test is an empirical question this paper does not answer.

Not the uniquely correct correction. We compared against five alternatives, not every possible method. Other approaches such as information-theoretic tests, or likelihood-ratio comparisons against generative language models, could in principle match or exceed the four-category framework. The specific claim is that the four-category framework handles the anti-signal failure mode that single-word significance tests miss; other corrections may address this failure mode in different ways.

The analytical model is first-order. The independent-character assumption is a simplification; character transitions in natural language are not independent. A bigram- or trigram-conditioned model would likely push R^2 higher. The current model is adequate for establishing that the effect is principled (mechanism exists, scales predictably with dictionary size and character compatibility) without over-fitting.

The analytical model is not new machinery. Equation 1 is a Karlin-Altschul-style null-model expected count applied to dictionary matching. The novelty is its validation as a noise-floor predictor across four writing systems, not the form of the equation itself.

Applicability is bounded to decoded-token lengths of approximately 2 to 4 characters; see Section 10.

13.3 Practical implications

For practitioners evaluating computational decipherments, we recommend the following (the `dictcollision` Python package, available via `pip install dictcollision`, implements all four steps as one-line function calls):

1. Do not report dictionary hit rate as a success metric without calibration.
2. Compute the null apparent hit rate, the dictionary hit rate on bigram-resampled text, as a baseline. Report it alongside any claimed hit rate.
3. Where possible, apply the four-category framework to separate SIGNAL from ANTI_SIGNAL. If SIGNAL $<$ ANTI_SIGNAL (net signal negative), the result is worse than chance.
4. Right-size the evaluation dictionary to the hypothesized plaintext's vocabulary size. Dictionaries larger than approximately 10K to 50K should be expected to inflate results by an order of magnitude.

14 Conclusion

Dictionary hit rate is a broken success metric for computational decipherment because short decoded strings collide with large dictionaries at rates approaching genuine matches. We introduced a four-category token classification that separates genuine signal from chance-collision noise and showed that it is the only method among six tested corrections that correctly flags wrong-language evaluations as worse than chance. The framework generalizes across synthetic and classical ciphers, an empirical character-distribution collision model predicts its null noise floor with $R^2 = 0.853$, and an end-to-end reproducible pipeline is available in the accompanying repository. Future undeciphered-script work that reports dictionary hit rate should report the four-category decomposition or an equivalent calibrated metric.

Acknowledgements

The author used Anthropic's Claude Opus 4.6 model as an AI assistant during this work, both to accelerate implementation of the experimental pipeline and for editorial review of the manuscript. All conceptual contributions, experimental design choices, and final claims are the author's; AI-generated text and code were reviewed and revised by the author before inclusion.

Data and code availability

All experimental code, datasets, and figure-generation scripts that reproduce the paper are at `https://github.com/mruckman1/signal-isolation-paper`. All random operations use explicit seeds; full reproduction requires under ten minutes on a standard laptop.

The core algorithms (noise-floor prediction, four-category classification, dictionary ranking) are also published as a standalone Python library, `dictcollision`, on PyPI: `https://pypi.org/project/dictcollision/`. Install with `pip install dictcollision`; source at `https://github.com/mruckman1/dictcollision`. Practitioners who want to apply the framework to their own decipherment work should use the library; the paper repository is primarily for reproducing the experiments and figures in this paper.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3), 403–410.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B*, 57(1), 289–300.

Bowern, C., & Lindemann, L. (2021). The linguistics of the Voynich Manuscript. *Annual Review of Linguistics*, 7, 285–308.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), 665–673.

Goldhahn, D., Eckart, T., & Quasthoff, U. (2012). Building large monolingual dictionaries at the Leipzig Corpora Collection: From 100 to 200 languages. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, 759–765. Dataset: Latin Wikipedia 2021, 100K sentences.

Hermit Dave (2018). *FrequencyWords: Word frequency lists from OpenSubtitles 2018*. https://github.com/hermitdave/FrequencyWords.

Johnson, K. P. (2015). 10,000 most frequent words in Greek and Latin canon. Classical Language Toolkit blog, April 23, 2015. Data file: https://kyle-p-johnson.com/assets/most-common-latin-words.txt.

Massey, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253), 68–78.

Konrad von Megenberg (c. 1475). *Das Buch der Natur*. First printed edition, Augsburg: Johann Bämler.

Matthaeus Platearius [pseudo-] (12th century). *De simplici medicina ("Circa instans")*. First printed in *Practica Jo. Serapionis ... Liber de simplici medicina, dictus circa instans*. Venice: Bonetus Locatellus for Octavianus Scotus, 16 December 1497. ISTC is00466000.

Reddy, S., & Knight, K. (2011). What we know about the Voynich Manuscript. *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage*, 78–86.

Schinner, A. (2007). The Voynich Manuscript: Evidence of the hoax hypothesis. *Cryptologia*, 31(2), 95–107.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379–423.

Timm, T., & Schinner, A. (2020). A possible generating algorithm for the Voynich Manuscript. *Cryptologia*, 44(1), 1–19.

Supplementary figures

S1 Continuous corruption sweep (`corruption_sweep.pdf`): apparent hit rate and net signal as decode-table corruption sweeps from 0 to 100 percent.

S2 Threshold sensitivity (`fig_threshold_sensitivity.pdf`): net signal vs. null threshold $t \in \{1, 2, 3, 4\}$ across six representative cells.

S3 Full language grid (`fig_language_grid.pdf`): 2×3 version of Figure 2 (p. 7) including the 40-percent-corrupted table condition.

S4 Classical Vigenère results (`fig_vigenere_results.pdf`): apparent hit rate and net signal for seven Vigenère key conditions across four dictionary sizes.

All four figures are also bundled together in `paper/v1/supplementary.pdf`; the individual source files are in `paper/v1/figures/`.