Arithmetic Coding, Quotients, and Prime Powers: The $E \to \mathbb{N} \to Q$ Bridge

Claude and MJC

February 2026

Abstract

We show that arithmetic coding, the quotient $Q = e/|E|$, and the prime power encoding $E \to \mathbb{N}$ are three views of the same construction. In arithmetic coding, we narrow a subinterval of $[0, 1)$. In the $E \to \mathbb{N}$ map, we encode the total event as an integer e with maximum $|E|$, and the quotient $Q = e/|E|$ is the position within $[0, 1)$ that arithmetic coding selects. Scaling by probability under the model recovers the code length. With a factor 2π, the quotient becomes an angle $\theta = 2\pi e/|E|$ on the unit circle, connecting compression to phase. Via the prime power embedding, each event space maps to a unique prime, and the fundamental theorem of arithmetic guarantees that the factorization of e reads back the full event state. We develop this connection, show that prime-power structure introduces multiplicative number theory into compression, and identify the quotient as the bridge between the additive world of bits and the multiplicative world of events.

This is v2 of the UM arithmetic coding paper. v1 established the basic construction; this version develops the $E \to \mathbb{N} \to Q$ connection at MJC's direction.

1 Three Views of the Same Thing

1.1 Arithmetic Coding: The Interval View

Arithmetic coding encodes a sequence $b_1, \ldots, b_n$ by maintaining an interval $[L, H) \subseteq [0, 1)$. At each step, the model predicts $P(b \mid \text{history})$, the interval is subdivided proportionally, and the subinterval for the actual byte b_t becomes the new interval. The final interval $[L, H)$ has width $\prod_t P(b_t \mid b_1^{t-1})$, and any point in this interval (encoded in $\lceil -\log_2(H - L) \rceil + 1$ bits) suffices to reconstruct the sequence.

The key object is the *position within the interval*: where the actual outcome falls relative to the model's prediction.

1.2 The Quotient View: $Q = e/|E|$

In the UM framework, the total event e at position t is an integer encoding everything that happened: which byte was input, what the hidden state is, what the output prediction was. The event space E has a finite number of possible states $|E|$, so $e \in \{0, 1, \ldots, |E| - 1\}$.

The *quotient* is:

$$Q_t = \frac{e_t}{|E|} \in [0, 1).$$

This is the position of the actual event within the space of all possible events. Under a uniform model, Q_t is the fraction of the event space "below" the actual outcome—exactly the quantity that arithmetic coding encodes.

Under a non-uniform model with probabilities $P(e)$, we define the *probability-weighted quotient*:

$$Q_t^P = F(e_t) + \tfrac{1}{2}P(e_t)$$

where $F(e) = \sum_{e' < e} P(e')$ is the CDF. This is the midpoint of the interval assigned to event e_t by arithmetic coding.

Proposition 1 (Quotient = Arithmetic Code Position)**.** *The probability-weighted quotient Q_t^P is the center of the arithmetic coding interval for event e_t. The code length for position t is $-\log_2 P(e_t) = -\log_2 |Q_t^P$ interval width$|$. Under the true data distribution, Q_t^P is approximately uniform on $[0, 1)$—this is the "luck" of position t.*

The connection is immediate: arithmetic coding *is* the quotient, computed incrementally, with probability weighting.

1.3 The Angle View: $\theta = 2\pi Q$

MJC prefers to include a factor 2π and interpret the quotient as an angle:

$$\theta_t = 2\pi \cdot \frac{e_t}{|E|} \in [0, 2\pi).$$

This maps each event to a point on the unit circle S^1. The arithmetic coding interval becomes an arc. The code length for position t is $-\log_2(\text{arc length}/2\pi)$.

The angle view has several advantages:

- **Periodicity.** The circle wraps around: $\theta = 0$ and $\theta = 2\pi$ are the same point. This is natural for cyclic event spaces (e.g., hours of day, characters in a periodic pattern).
- **Phase.** Each position's event is a phase. The sequence of phases $\theta_1, \theta_2, \ldots$ traces a path on S^1. Compression corresponds to phase predictability: when the model knows where the phase will land, the arc is narrow and few bits are needed.
- **Fourier.** The phase representation connects naturally to Fourier analysis. A pattern that repeats every k positions has phase $\theta_t \approx 2\pi t/k \pmod{2\pi}$, detectable as a peak in the DFT of the θ sequence.
- **Composition.** When two event spaces combine, their angles add (modulo 2π). The combined phase $\theta = \theta_A + \theta_B$ is the angle of the product event.

2 $E \to \mathbb{N}$: The Prime Power Encoding

2.1 Construction

Each event space E_i gets a unique prime p_i. The exponent encodes which event within E_i is currently true. If there are k event spaces with current values $v_1, v_2, \ldots, v_k$, the total event is:

$$e = \prod_{i=1}^{k} p_i^{v_i} \in \mathbb{N}.$$

This is a bijection by the fundamental theorem of arithmetic: the prime factorization of e uniquely recovers all event values $(v_1, \ldots, v_k)$.

Definition 2 (Prime Power Event Encoding)**.** *Given k event spaces $E_1, \ldots, E_k$ with value ranges $\{0, \ldots, n_i - 1\}$ and assigned primes $p_1 < p_2 < \cdots < p_k$, the* prime power encoding *is:*

$$\phi : E_1 \times \cdots \times E_k \to \mathbb{N}, \qquad \phi(v_1, \ldots, v_k) = \prod_{i=1}^{k} p_i^{v_i}.$$

The maximum is $|E| = \prod_{i=1}^{k} p_i^{n_i - 1}$. The quotient is $Q = \phi(v_1, \ldots, v_k)/|E|$.

2.2 Example: Two Event Spaces

Consider input character (256 values, prime $p_1 = 2$) and position (1024 values, prime $p_2 = 3$). At position t with input byte b:

$$S_t = 2^b \cdot 3^t.$$

The maximum is $|E| = 2^{255} \cdot 3^{1023}$. The quotient:

$$Q_t = \frac{2^{b_t} \cdot 3^t}{2^{255} \cdot 3^{1023}} = 2^{b_t - 255} \cdot 3^{t-1023}.$$

For our 1024-byte experiment, the full dataset encodes as:

$$P(E) = \sum_{t=1}^{1024} S_t = \sum_{t=1}^{1024} 2^{b_t} \cdot 3^t,$$

a 497-digit integer. Each term's prime factorization reads back (b_t, t)—the byte and its position.

2.3 The Dataset as a Single Integer

The sum $P(E) = \sum_t S_t$ encodes the full dataset as one integer. This is not merely notational: the arithmetic structure of $P(E)$ reflects the statistical structure of the data.

Proposition 3 (Factoring $P(E)$ Recovers the Data)**.** *Since each S_t has a unique power of p_2 (the time prime), the terms are distinguishable in $P(E)$. Specifically:*

1. *$P(E) \bmod p_2^{t+1}$ isolates the contribution of positions $\leq t$.*
2. *The coefficient of p_2^t in the p_2-adic expansion of $P(E)$ recovers S_t/p_2^t, which factors into the non-time event values at position t.*

3 Quotient and Compression

3.1 The Uniform Quotient

Under a uniform model ($P(e) = 1/|E|$ for all e), the code length is $\log_2 |E|$ bits per position—no compression. The quotient $Q = e/|E|$ is uniform on $[0, 1)$, and the angle $\theta = 2\pi Q$ is uniform on the circle.

3.2 The Model-Weighted Quotient

A model assigns non-uniform probabilities. The model-weighted quotient uses the CDF instead of the raw index:

$$Q^P = F(e) + \tfrac{1}{2}P(e)$$

The code length is $-\log_2 P(e)$ bits. High-probability events get wide intervals (few bits); low-probability events get narrow intervals (many bits).

The total code length is:

$$L = \sum_{t=1}^{n} -\log_2 P(e_t) = n \cdot H_{\text{cross}}$$

where H_{cross} is the cross-entropy. This is independent of the prime assignment, the ordering of events, or whether we use the angle view—all that matters is the probability the model assigns to each actual event.

3.3 What the Quotient Adds

If the code length is the same regardless, why bother with Q?

Because the quotient reveals *structure* that the code length hides. Two examples:

1. **Pattern detection.** At positions where the model is confident and correct, Q^P clusters near 0.5 (the center of a wide interval). At positions of genuine surprise, Q^P is near 0 or 1 (the edges of a narrow interval). The *distribution* of Q^P values diagnoses model quality—flat = calibrated, peaked = overconfident, U-shaped = underconfident.

2. **Factored quotients.** Since $e = \prod p_i^{v_i}$, the quotient decomposes:

$$Q = \frac{e}{|E|} = \prod_{i=1}^{k} \frac{p_i^{v_i}}{p_i^{n_i - 1}} = \prod_{i=1}^{k} p_i^{v_i - (n_i - 1)}.$$

   Each factor $p_i^{v_i-(n_i-1)}$ is the per-event-space contribution to the quotient. Taking logs:

$$\log Q = \sum_{i=1}^{k} (v_i - (n_i - 1)) \log p_i.$$

   The additive decomposition of $\log Q$ over event spaces parallels the additive decomposition of code length over model components. The prime assignment determines the "exchange rate" between event spaces.

4 Number Theory of Compression

4.1 Multiplicative Structure

The prime power encoding introduces multiplicative number theory into compression. Key correspondences:

Number Theory	Compression
Prime p_i	Event space E_i
Exponent v_i	Event value (which event is true)
Product $\prod p_i^{v_i}$	Joint event (full state)
$\gcd(e_s, e_t)$	Shared state between positions s, t
$\mathrm{lcm}(e_s, e_t)$	Union of events at s and t
Factorization	Event space decomposition
Coprimality	Independence of event spaces
Divisibility $e_s \mid e_t$	s's state is a sub-state of t's

Proposition 4 (GCD Detects Shared State)**.** $\gcd(S_s, S_t) = \prod_i p_i^{\min(v_i^{(s)}, v_i^{(t)})}$. *This equals 1 (the positions share no event values) if and only if all event spaces have different values at s and t. When event space E_j has the same value at both positions ($v_j^{(s)} = v_j^{(t)}$), $p_j^{v_j}$ divides the GCD.*

This gives a purely arithmetic characterization of patterns. A UM pattern "when offset $-d$ has value a in E_j, predict value b in E_{out}" fires at position t iff $p_j^a \mid S_{t-d}$ and $p_{\text{out}}^b \mid S_t$. Pattern discovery becomes a factorization problem.

4.2 The Sum and the Product

Two natural aggregates over the dataset:

The **sum** $P(E) = \sum_t S_t$ encodes the dataset as a single integer. It preserves individual terms (recoverable via the time prime). The sum is the object from our earlier work.

The **product** $\Pi(E) = \prod_t S_t = \prod_t \prod_i p_i^{v_i^{(t)}} = \prod_i p_i^{\sum_t v_i^{(t)}}$. The exponent of each prime in $\Pi(E)$ is the *sum* of that event space's values across all positions. For the input character ES (prime 2), this sum counts the total "weight" of each character—a frequency statistic.

Observation 5 (Product Encodes Marginal Statistics)**.** $\Pi(E) = \prod_i p_i^{V_i}$ *where* $V_i = \sum_t v_i^{(t)}$ *is the total value of event space* i*. The exponents are the sufficient statistics for the marginal distribution of each event space. No cross-position structure is preserved.*

The quotient $\Pi(E)/P(E)$ is therefore a measure of how much structure is lost when going from the full joint to the marginals—a purely arithmetic quantity that characterizes the "interestingness" of the dataset.

4.3 The Möbius Function and Inclusion-Exclusion

The Möbius function $\mu(n)$ performs inclusion-exclusion over divisors. In our setting, μ inverts the divisor sum, which corresponds to pattern overlap correction.

If $f(e)$ counts how many positions match event e (i.e., $f(e) = |\{t : e \mid S_t\}|$), then the Möbius inversion gives the "exclusive" count—positions matching exactly the events in e and no finer pattern:

$$g(e) = \sum_{d|e} \mu(e/d)\, f(d).$$

This is the number-theoretic version of the UM's pattern counting with overlap correction. The discount operation in Kneser–Ney smoothing (identified as approximate GCD removal in the KN-quotient paper) has a cousin in Möbius inversion: both subtract over-counted shared structure.

4.4 Euler's Totient and Event Space Size

Euler's totient $\varphi(n)$ counts integers coprime to n below n. For $n = \prod p_i^{a_i}$, $\varphi(n) = n \prod(1 - 1/p_i)$.

In our encoding, $\varphi(|E|)$ counts the number of "maximally informative" events—those coprime to the boundary of the event space. The ratio $\varphi(|E|)/|E| = \prod(1 - 1/p_i)$ decreases as more event spaces are added (each new prime p_i reduces it), reflecting the increasing rarity of states that are maximally informative about all event spaces simultaneously.

For k binary event spaces (p_i = the i-th prime, $n_i = 2$):

$$|E| = \prod_{i=1}^{k} p_i, \qquad \frac{\varphi(|E|)}{|E|} = \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right).$$

By Mertens' theorem, this is $\sim e^{-\gamma}/\ln p_k$ for large k, where $\gamma \approx 0.577$ is the Euler–Mascheroni constant. The density of maximally-informative events decays logarithmically with the number of event spaces—a fundamental sparsity result.

5 The Phase Circle and Compression

5.1 Events as Phases

With the angle map $\theta_t = 2\pi e_t/|E|$, each position is a point on the unit circle. The dataset is a collection of n points on S^1.

Under a uniform model, these points are uniformly distributed and incompressible. Under a good model, the points cluster into predictable arcs, and the code encodes which arc each point falls in.

5.2 The Phase Walk

The sequence $\theta_1, \theta_2, \ldots, \theta_n$ traces a "phase walk" on the circle. The angular velocity

$$\omega_t = \theta_t - \theta_{t-1} \pmod{2\pi}$$

measures how much the event state changed between consecutive positions. When the UM is tracking a stable pattern (e.g., mid-word in a known word), ω_t is small and predictable. At surprise points (word boundaries, topic changes, rare characters), ω_t is large and unpredictable.

The total path length $\sum_t |\omega_t|$ is a measure of the dataset's "event-space complexity"—how much the state wanders. The model's job is to predict not just where the phase is, but where it will go next.

5.3 Factored Phases

Since $e = \prod p_i^{v_i}$ and $|E| = \prod p_i^{n_i - 1}$:

$$\theta = 2\pi \cdot \frac{e}{|E|} = 2\pi \cdot \prod_i p_i^{v_i - (n_i - 1)}.$$

Taking the logarithm and using the angle's additive structure in log-space:

$$\frac{\theta}{2\pi} = \exp\left(\sum_i (v_i - (n_i - 1)) \ln p_i\right).$$

Each event space contributes independently to the log-quotient, with weight $\ln p_i$. The assignment of primes to event spaces determines these weights. Small primes (2, 3, 5) give the most weight per unit exponent; large primes contribute less per unit.

This suggests a *natural prime assignment*: the most important event space (the one whose value carries the most predictive information) should get the smallest prime. For byte-level prediction, this is the output character space, which should get prime 2.

6 Connecting All Three Views

6.1 The Bridge Theorem

Theorem 6 (AC–Quotient–Prime Bridge)**.** *Let $\mathcal{M}$ be a UM with k event spaces, and let $b_1, \ldots, b_n$ be a byte sequence. Then:*

1. *The arithmetic coding interval after processing b_1^t is $[L_t, H_t) \subseteq [0, 1)$.*
2. *The model-weighted quotient $Q_t^P = F(b_t) + \frac{1}{2}P(b_t)$ lies in $[L_t, H_t)$ (it is the midpoint of the subinterval for b_t).*
3. *The prime-encoded event $e_t = \prod_i p_i^{v_i^{(t)}}$ determines Q_t^P via the model's CDF applied to the event whose output component is b_t.*
4. *The code length is $\ell = \sum_t -\log_2 P(b_t) = \sum_t -\log_2(H_t - L_t) + O(1)$ bits.*
5. *The angle $\theta_t = 2\pi Q_t^P$ gives the position on the unit circle, with arc length $2\pi P(b_t)$.*

All five quantities—interval, quotient, prime encoding, code length, and angle—are deterministic functions of each other given the model.

Proof. (1)–(4) are standard properties of arithmetic coding. For (5), $\theta_t = 2\pi Q_t^P$ by definition, and the arc length $2\pi(H_t - L_t) = 2\pi P(b_t)$ is the probability scaled to the circle's circumference. □

6.2 What Each View Reveals

View	Primary Object	Best For
Arithmetic coding	Interval $[L, H)$	Implementation, bit-exact coding
Quotient Q	Position in $[0, 1)$	Luck, calibration, comparison
Angle θ	Phase on S^1	Periodicity, dynamics, Fourier
Prime encoding e	Integer in $\mathbb{N}$	Factorization, patterns, GCD
Code length ℓ	Bits	Evaluation, Hutter Prize score

The views are equivalent but not interchangeable: each emphasizes different structure. The prime encoding reveals factored event-space structure that is invisible in the interval view. The angle view reveals periodicity invisible in the quotient. The code length is the practical bottom line.

7 Implications

7.1 Prime Assignment as Model Architecture

The choice of which event space gets which prime is an architectural decision. Analogies:

- Assigning the output ES to prime 2 is like making prediction the "most significant bit" of the state.
- Assigning the time ES to a large prime is like treating position as a minor annotation (appropriate for online models where position is implicit).
- Two isomorphic models with different prime assignments produce different $P(E)$ integers but the same code length. The prime assignment is a *gauge choice*.

7.2 The Compression–Factoring Duality

Compression = finding structure = factoring the event space into components that predict each other. The prime power encoding makes this literal:

- A compressible dataset has a "smooth" $P(E)$ (many small prime factors, regular patterns in the exponents).
- An incompressible dataset has a "rough" $P(E)$ (large prime factors, irregular exponents).
- Adding a new event space (a new prime) helps compression iff it captures structure not already encoded by existing primes.

7.3 Toward a Hutter Prize Submission

Our best UM (v16: KN-6 + sparse + match, 1.588 bpc) can be wrapped in arithmetic coding to produce a compressor scoring ∼189 MB. The $E \to \mathbb{N} \to Q$ framework does not change the score—it provides a *theory* of why the score is what it is.

Specifically: the 0.70 bpc gap to the fx2-cmix record (0.886 bpc) corresponds to event spaces that our model has not yet discovered. Each new event space, correctly combined, narrows the gap. The prime encoding provides a language for describing exactly which event spaces are missing and how they would factor the dataset's structure.

8 Conclusion

Arithmetic coding, the quotient $Q = e/|E|$, and the prime power encoding are three facets of one construction:

1. **AC** encodes a sequence by narrowing intervals according to model probabilities.
2. Q measures the luck of each event as a fraction of the event space, or equivalently as an angle $\theta = 2\pi Q$ on the unit circle.
3. $E \to \mathbb{N}$ maps the factored event state to a single integer via prime powers, making the fundamental theorem of arithmetic the readout mechanism.

The bridge is the quotient: $Q = e/|E|$ maps the integer to $[0, 1)$, which is the interval that arithmetic coding selects. Probability weighting converts the uniform quotient to the model-weighted quotient, and the code length falls out as $-\log_2 P(e)$.

The prime encoding contributes more than notation. It brings multiplicative number theory—GCD for shared state, Möbius inversion for pattern overlap, Euler's totient for sparsity—into the toolkit for analyzing and constructing UMs. The angle interpretation adds phase dynamics and Fourier analysis.

Together, these tools provide a mathematical framework where compression, prediction, event spaces, prime factorization, and circular geometry are all views of one underlying object: the structure of the data as seen through the model.

References

[1] Claude and MJC, "Arithmetic Coding via Universal Models: Any UM is a Compressor," 2026. (v1 of this paper.)

[2] Claude and MJC, "Event Arithmetic: E onto $\mathbb{N}$," 2026.

[3] Claude and MJC, "Kneser–Ney as Quotient," 2026.

[4] Claude, "Exploring the Prime Encoding: Eight Experiments on 1024 Bytes," 2026.

[5] Claude and MJC, "Match Models, Sparse Contexts, and the Combination Problem," 2026.

[6] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Comm. ACM*, 1987.

[7] F. Mertens, "Ein Beitrag zur analytischen Zahlentheorie," *J. Reine Angew. Math.*, 1874.