Statistics of the Compression Ratio of a Variable-to-Variable Code: Exact Moments and Asymptotic Behavior

(Extended version, with full proofs)

Neri Merhav

July 16, 2026

The Viterbi Faculty of Electrical and Computer Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 3200003, ISRAEL
E–mail: merhav@technion.ac.il

Abstract

A variable-to-variable (V2V) length code parses a source sequence into phrases of variable length and maps each phrase to a binary codeword of, generally, a different random length. After encoding n phrases, the realized compression ratio $R_n = \Lambda_n/\Sigma_n$ – total codeword length over total source-symbol count – is the finite-sample counterpart of the code's asymptotic rate ρ, to which it converges only as $n \to \infty$. This paper first derives exact formulas for all integer moments of R_n for a given discrete memoryless source (DMS). Specifically, we obtain a closed-form formula for every moment $\mathbb{E}\{R_n^k\}$ as a one-dimensional integral involving only single-phrase moment generating functions of the pair (L, ℓ) – the phrase length, in source symbols, and codeword length, in bits. From these moments we derive an Edgeworth approximation to the cumulative distribution function (CDF) of R_n that is substantially more accurate than the central limit theorem (CLT) approximation. Using the Laplace method of integration, we also derive explicit closed-form formulas for the bias constant $C = \lim_{n\to\infty} n(\mathbb{E}\{R_n\} - \rho)$ and for the variance constant $\lim_{n\to\infty} n \cdot \mathrm{Var}\{R_n\}$. The analysis extends to Markov sources via state-indexed matrices with a redundancy formula obtained in closed form.

On the coding-theoretic side, we cast V2V length codes as finite-state encoders and apply a generalized Kraft inequality for a compression-rate lower bound, and give a structural decomposition of the bias coefficient that separates cleanly across variable-to-fixed (V2F) length codes, fixed-to-variable (F2V) length codes, and V2V length codes. Applied to the Khodak code of Bugeaud, Drmota, and Szpankowski, this decomposition shows that its improved performance is reflected in its smaller bias constant.

1 Introduction

1.1 Objectives and motivation

A variable-to-variable (V2V) length source code is the most general member of the family of lossless codes considered here. It parses the source into phrases of variable length and encodes each phrase with a binary codeword of variable length, combining the structural advantages of variable-to-fixed

(V2F) length coding – which adapts the parsing to source statistics – with those of fixed-to-variable (F2V) length coding – which adapts codeword lengths to symbol probabilities.

The standard performance measure for a V2V length code is its asymptotic compression rate, $\rho = \mathbb{E}\{\ell\}/\mathbb{E}\{L\}$, the ratio of expected codeword length to expected phrase length. Let Λ_n and Σ_n denote, respectively, the total codeword length (in bits) and the total source-symbol count after n encoded phrases. By the strong law of large numbers ($\Lambda_n/n \to \mathbb{E}\{\ell\}$ and $\Sigma_n/n \to \mathbb{E}\{L\}$ almost surely) and the continuous mapping theorem [1], the realized compression ratio $R_n = \Lambda_n/\Sigma_n$ converges to ρ almost surely as $n \to \infty$. This asymptotic rate, however, answers neither how quickly convergence occurs nor how large the fluctuations around ρ are at any finite number of encoded phrases n. Both questions are practically important. In buffer analysis for variable-length codes, the fluctuation of the compression ratio over a finite window directly determines buffer overflow probabilities. In code design, comparing candidate codes requires evaluating their performance at finite n, not merely their limit.

We address these questions by deriving exact formulas for all integer moments of R_n, for a given V2V length code and a memoryless source. This is fundamentally an analysis of a *given* code's finite-sample behavior, not a code-design method; the joint design of the parsing tree and codeword lengths for a target n remains open (see Section 2.5). Our main result is an exact formula for $\mathbb{E}\{R_n^k\}$ for every positive integer k as a single one-dimensional integral, requiring only the single-phrase joint moment generating function (MGF) of (L, ℓ). The underlying tool is an integral representation of $1/\Sigma_n$ as well as its k-th power as a Laplace transform, which enables turning a ratio of sums into a one-dimensional integral of simple single-phrase quantities. The novelty is in the systematic extension to *every* integer moment via the set-partition combinatorics of Theorem 3.2; the fact that the resulting formulas are exact at every finite n rather than asymptotic; and the further extension to Markov sources (Section 5), which replaces scalar moment generating functions by matrix-valued ones and requires a corresponding eigenvalue-perturbation argument with no scalar analogue. From these moments we derive closed-form asymptotic formulas for the bias constant $\lim_{n\to\infty} n(\mathbb{E}\{R_n\}-\rho)$ and the variance constant $\lim_{n\to\infty} n\mathrm{Var}\{R_n\}$, and we construct an Edgeworth approximation to the CDF of R_n that is substantially more accurate than the central limit theorem (CLT). Before turning to this analysis, Section 2 records some necessary background, including a compression-rate lower bound obtained by observing that a V2V length code is an instance of the finite-state encoders covered by the generalized Kraft inequality of [2]; this observation sets the stage for the moment analysis but is logically independent of it. Unlike the classical route to such

a bound – concatenating many phrases and invoking a law-of-large-numbers argument to identify the limiting rate – the generalized-Kraft-inequality bound is a direct, purely algebraic consequence of a single spectral-radius inequality, with no block-length limit theorem involved.

1.2 Related work

The V2F and V2V length coding literature is relatively sparse. Existing work, surveyed in [3], focuses on the average redundancy of specific named codes (Huffman, Tunstall, Khodak, Boncelet) for known sources, using analytic combinatorics and Mellin-transform methods; Savari and Szpankowski [21] give an early analysis specifically of V2V length codes along these lines. The moments of R_n at finite n, as distinct from the asymptotic rate ρ, have not been addressed.

V2F length coding. Tunstall's algorithm [4] constructs an optimal uniquely parsable V2F dictionary by a simple greedy procedure. Savari and Gallager [5] established the asymptotic redundancy of Tunstall codes using renewal theory (modeling self-information as a regenerative process) and Markov reward machinery; their analysis brings in second moments of inter-renewal times, but only as a correction to the precision of an asymptotic mean formula, not as a variance target. Drmota, Reznik, and Szpankowski [6] established a CLT and a variance formula for the V2F phrase length L, via renewal theory and Mellin transforms, as the dictionary size $M \to \infty$. Both bodies of work address the phrase length L alone, under a fixed-length codeword assignment, as a function of a growing dictionary; our asymptotic variable is the number of phrases n processed by a fixed code, and our object is the ratio $R_n = \Lambda_n/\Sigma_n$ rather than L.

Plurally parsable dictionaries. Savari [7, 8] showed that relaxing unique parsability can outperform the Tunstall dictionary of the same size for highly predictable sources, with exact results for specific binary-source families but no general algorithm.

Delay and redundancy. Shayevitz, Meron, Feder, and Zamir [9] characterize the redundancy–delay trade-off for V2V and related code families. Their object is the expected per-symbol code length as a function of a worst-case delay constraint, not the moments of R_n as the number of phrases grows.

Codeword-length distributions. Courtade and Verdú [11] study the CGF and Gaussian approximation of the codeword length of an optimal F2V length code on an n-symbol block, tracing

back to Strassen's foundational second-order asymptotic analysis [10] and complementing Kontoyiannis's earlier second-order noiseless source coding theorems [12] (which also covers Markov sources) and the non-asymptotic refinements of Kontoyiannis and Verdú [13]. This is a single i.i.d. sum, not a ratio of two correlated sums with a random denominator; the ratio structure is specific to V2V and V2F coding and is the source of the combinatorial complexity addressed in the present paper.

Generalized Kraft inequality. Merhav [2] establishes a generalized Kraft inequality for finite-state encoders, strictly improving the earlier result of Ziv and Lempel [14]. We use this to derive an explicit lower bound on the V2V compression ratio in terms of the parsing tree structure.

2 System model and background on V2V codes

2.1 Source model and notation

We consider a discrete memoryless source (DMS) $X_1, X_2, \ldots$, where each symbol X_i (i – positive integer) is a random variable taking values in a finite alphabet $\mathcal{X}$ of cardinality α; specific realizations are denoted by $x_1, x_2, \ldots$. Each symbol is drawn according to the source distribution $P = \{P(x),\ x \in \mathcal{X}\}$. The single-letter entropy is

$$H(X) = -\sum_{x\in\mathcal{X}} P(x)\log_2 P(x). \tag{1}$$

A V2V length code is specified by two components:

1) A *source dictionary* $\mathcal{D}$: a prefix-free set of variable-length source strings, represented by the leaves of a complete α-ary tree. The size of the dictionary is $|\mathcal{D}| = M$, and a generic member of $\mathcal{D}$ is denoted by y. A source string $x_1, x_2, \ldots$ is parsed by the dictionary parser into a succession of phrases $y_1, y_2, \ldots$; the corresponding random variables $Y_1, Y_2, \ldots$, induced by the source P, form another DMS with alphabet $\mathcal{D}$ (of size M) and phrase probabilities $\{Q(y),\ y \in \mathcal{D}\}$, where each $Q(y)$ is given by the product of the letter probabilities that make up the phrase y. Let $L(y)$, $y \in \mathcal{D}$, denote the length of phrase y, in source symbols. We denote by $H(Y) \triangleq -\sum_{y\in\mathcal{D}} Q(y)\log_2 Q(y)$ the phrase entropy.

2) A variable-length uniquely decodable (UD) code, mapping $\mathcal{D}$ into a set of variable-length binary strings. Let $\ell(y)$ denote the length, in bits, of the codeword assigned to $y \in \mathcal{D}$.

We index the successive phrases produced by the parser by $i = 1, 2, \ldots$; the i-th phrase is Y_i, of length $L(Y_i)$ source symbols, encoded by a codeword of length $\ell(Y_i)$ bits. For a DMS, the pairs $(L(Y_i), \ell(Y_i))_{i \geq 1}$ are independent and identically distributed; we write Y for a generic random phrase with distribution Q, so that $L(Y)$ and $\ell(Y)$ denote its length and its codeword length, respectively. When it causes no ambiguity, we abbreviate $L(Y)$ and $\ell(Y)$ by L and ℓ; the explicit argument Y is retained whenever a formula defines a new named quantity as an expectation. The following single-phrase quantity appears throughout:

$$\mu_j(t) \triangleq \mathbb{E}\{[\ell(Y)]^j e^{-tL(Y)}\}, \ j = 0, 1, 2, \ldots, \ t \geq 0. \tag{2}$$

Note that $\mu_0(t) = \mathbb{E}\{e^{-tL(Y)}\}$ is the MGF of $-L(Y)$. We define the mean phrase length and mean codeword length by

$$\bar{L} \triangleq \mathbb{E}\{L(Y)\} = \sum_{y \in \mathcal{D}} Q(y) L(y), \qquad \bar{\ell} \triangleq \mathbb{E}\{\ell(Y)\} = \sum_{y \in \mathcal{D}} Q(y) \ell(y). \tag{3}$$

The asymptotic compression rate of the code is defined as

$$\rho \triangleq \frac{\bar{\ell}}{\bar{L}} = -\frac{\mu_1(0)}{\mu_0'(0)}, \tag{4}$$

where $\mu_0'(\cdot)$ denotes the derivative of $\mu_0(\cdot)$. For any two random variables U, V appearing below, we write $\mathrm{Var}\{U\} \triangleq \mathbb{E}\{U^2\} - (\mathbb{E}\{U\})^2$ and $\mathrm{Cov}\{U, V\} \triangleq \mathbb{E}\{UV\} - \mathbb{E}\{U\}\mathbb{E}\{V\}$ for their variance and covariance, respectively.

After encoding n phrases, the total code length is $\Lambda_n \triangleq \sum_{i=1}^n \ell(Y_i)$ (in bits) and the total number of source symbols consumed is $\Sigma_n \triangleq \sum_{i=1}^n L(Y_i)$. The *realized compression ratio* over n phrases is

$$R_n \triangleq \frac{\Lambda_n}{\Sigma_n} = \frac{\sum_{i=1}^n \ell(Y_i)}{\sum_{i=1}^n L(Y_i)} \ \ \frac{\text{bits}}{\text{symbol}}. \tag{5}$$

By the strong law of large numbers, $\Lambda_n / n \to \bar{\ell}$ and $\Sigma_n / n \to \bar{L}$ almost surely (a.s.), and so, by the continuous mapping theorem, $R_n \to \rho$ a.s. as $n \to \infty$.

2.2 Structure and parsing

The dictionary $\mathcal{D}$ of a V2V length code is represented by a full α-ary rooted tree, i.e., every internal node has α children, one per each possible source symbol; the M leaves correspond to the M phrases $y \in \mathcal{D}$. The parsing rule is as follows: start at the root, follow the edge labeled by each successive source symbol until reaching a leaf. The leaf reached identifies the current phrase y; the encoder emits the binary codeword assigned to that leaf and resets to the root for the next phrase.

Let J denote the number of internal nodes of the parsing tree, including the root. Unique parsability forces $M = J(\alpha - 1) + 1$, so

$$J = \frac{M-1}{\alpha - 1}. \tag{6}$$

The phrase length satisfies $L(y) \geq 1$ for every $y \in \mathcal{D}$, and the codeword length $\ell(y)$ takes values in $\{1, 2, \ldots, \ell_{\max}\}$ for some maximum codeword length $\ell_{\max}$.

2.3 V2V as a finite-state encoder

A *finite-state (FS) encoder* over the source alphabet $\mathcal{X}$ and output alphabet $\mathcal{B}$ is specified by a finite state set $\mathcal{Z}$ (of size $|\mathcal{Z}|$), a *next-state function* $g : \mathcal{Z} \times \mathcal{X} \to \mathcal{Z}$, and an *output function* $f : \mathcal{Z} \times \mathcal{X} \to \mathcal{B}^*$, where $\mathcal{B}^*$ denotes a set of finite strings over $\mathcal{B}$, possibly including the empty string $\emptyset$ of length zero. For a string $s \in \mathcal{B}^*$, we write $\ell[s]$ for its length in bits, with the convention $\ell[\emptyset] = 0$; this is consistent with $\ell(y)$ of Section 2 when $s = c(y)$ is the codeword of leaf y. When a source sequence $x_1, x_2, \ldots$ is fed into the encoder, the state evolves according to the recursion $z_{i+1} = g(z_i, x_i)$, $i = 1, 2, \ldots$ (with z_1 being a fixed initial state), and the encoder outputs the sequence of strings from $\mathcal{B}$, $f(z_1, x_1), f(z_2, x_2), \ldots$

The V2V length encoder described in the preceding subsection can be cast as an FS encoder with a state set $\mathcal{Z}$ given by the set of J internal nodes of the parsing tree (thus $|\mathcal{Z}| = J$), source alphabet $\mathcal{X}$ (of cardinality α), and output alphabet $\mathcal{B} = \{0, 1\}$ (binary codewords). The encoder is in state $z \in \mathcal{Z}$ when it has partially matched a phrase and is currently at internal node z. For source symbol x processed while in state z, the next-state and output functions are:

$$g(z, x) = \begin{cases} z' & \text{if the child of } z \text{ along edge } x \text{ is an internal node } z', \\ \text{root} & \text{if the child of } z \text{ along edge } x \text{ is a leaf,} \end{cases} \tag{7}$$

$$f(z, x) = \begin{cases} \emptyset & \text{if } g(z, x) \neq \text{root}, \\ c(y) & \text{if the child of } z \text{ along } x \text{ is leaf } y. \end{cases} \tag{8}$$

The encoder emits output only when a phrase is completed (i.e., when the child is a leaf), at which point it resets to the root.

Definition 2.1 (Information lossless encoder, [14])**.** *An FS encoder is* information lossless (IL) *if, given the initial state, the output string, and the final state, the input string can be uniquely recovered.*

The IL property holds for uniquely parsable dictionaries, since each codeword maps to a unique leaf, which together with the initial parsing state determines the phrase consumed. For plurally

parsable dictionaries [7] – where a source string may have more than one valid parse – the IL property may fail, and the framework below requires modification.

For $m \geq 1$, $g(z, x^m)$ will denote the final state of the encoder when the initial state is z and it processes the m successive inputs $x^m = (x_1, \ldots, x_m) \in \mathcal{X}^m$; $f(z, x^m)$ denotes the corresponding output string emitted in response to x^m.

2.4 The generalized Kraft inequality and a lower bound on the compression ratio

The classical Kraft–McMillan inequality, $\sum_y 2^{-\ell(y)} \leq 1$, characterizes a limitation on lossless codes over a fixed alphabet but does not directly apply to FS encoders. For the IL FS encoder above, the appropriate generalization uses the following matrix.

Definition 2.2 (Kraft matrix, [2])**.** *For an IL FS encoder with state space $\mathcal{Z}$, the Kraft matrix $\mathbf{K} \in \mathbb{R}^{|\mathcal{Z}| \times |\mathcal{Z}|}$ has entries*

$$\mathbf{K}_{zz'} = \sum_{\{x:\ g(z,x)=z'\}} 2^{-\ell[f(z,x)]}, \quad z, z' \in \mathcal{Z}, \tag{9}$$

Let $\mathrm{spr}(\mathbf{K})$ denote the spectral radius of $\mathbf{K}$. The generalized Kraft inequality reads:

Theorem 2.1 (Generalized Kraft inequality, [2])**.** *For every IL FS encoder,* $\mathrm{spr}(\mathbf{K}) \leq 1$.

Theorem 2.2 (Irreducible bound, [2])**.** *If $\mathbf{K}$ is irreducible, then for all $z, z' \in \mathcal{Z}$ and every integer $m \geq 1$,*

$$(\mathbf{K}^m)_{zz'} = \sum_{x^m:\ g(z,x^m)=z'} 2^{-\ell[f(z,x^m)]} \leq 2^{(J-1)\ell_{\max}}. \tag{10}$$

Note that the right-hand side is *independent of* m. This is the key improvement over the earlier generalized Kraft inequality of Ziv and Lempel [14], whose analogous bound grows linearly in m; this linear growth prevents one from extracting any useful rate lower bound from the Ziv–Lempel result.

Substituting $J = (M-1)/(\alpha-1)$ into Theorem 2.2 and taking the supremum over $m \geq 1$ gives the following lower bound on the compression ratio of any V2V length code, valid for any source (not necessarily memoryless); this is eq. (34) of [2], whose proof combines the per-entry bound of Theorem 2.2 with the standard Kraft-sum-with-slack redundancy bound and is not reproduced here:

Corollary 2.1 (Lower bound on compression ratio)**.** *For an irreducible, uniquely parsable V2V length code with M leaves over an α-ary source alphabet and maximum codeword length $\ell_{\max}$, the asymptotic compression rate ρ satisfies*

$$\rho \geq \sup_{m \geq 1} \Big[H(X_m \mid X^{m-1}) - \frac{C(M, \alpha, \ell_{\max})}{m} \Big], \tag{11}$$

where $H(X_m \mid X^{m-1})$ is the conditional entropy (in bits per source symbol) of the m-th source symbol given the preceding block $X^{m-1} := (X_1, \ldots, X_{m-1})$, and

$$C(M, \alpha, \ell_{\max}) \triangleq 2 \log_2 J + (J-1)\ell_{\max} = 2 \log_2 \frac{M-1}{\alpha - 1} + \frac{(M-\alpha)\ell_{\max}}{\alpha - 1}. \tag{12}$$

The right-hand side of (11) cannot be smaller than the entropy rate $H_\infty \triangleq \lim_{m \to \infty} H(X_m \mid X^{m-1})$ (a supremum dominates the limit), and decreases in M. For a DMS, $H(X_m \mid X^{m-1}) = H(X)$, so the supremum is taken over $H(X) - C(M, \alpha, \ell_{\max})/m$, which yields $H(X)$. This is a substantially more direct route to the converse than the classical technique of concatenating many phrases and invoking a law-of-large-numbers argument.

2.5 Design

Unlike F2V length coding (where Huffman's algorithm gives a provably optimal greedy construction) and V2F length coding (where Tunstall's algorithm does the same for memoryless sources), no analogous optimal joint design algorithm is known for V2V length coding, where both the parsing tree and the codeword lengths are simultaneously free. The standard practice is a two-stage heuristic: build a Tunstall-optimal V2F tree, then apply Huffman coding to the leaf probabilities induced by that tree. This heuristic is not known to be jointly optimal and in general it is not [15]. The minimum achievable redundancy for V2V length codes of a given size M is an open problem. One known negative result: for a binary memoryless source with $0 < P(1) < 1/2$, no V2V code achieves exactly zero redundancy [3].

The best known achievability result is due to Khodak [3, 15]: for any memoryless source there exists a V2V code whose average redundancy decays as $O(\bar{L}^{-5/3})$, where $\bar{L}$ is the average phrase length. This strictly improves on the $O(1/\bar{L})$ redundancy of the Tunstall–Huffman heuristic, achieved by using a parsing tree that deliberately maintains non-uniform leaf probabilities so that variable codeword lengths can contribute meaningfully. An explicit construction achieving this bound is given in [15].

2.6 Notation summary

The paper accumulates a fair amount of notation as it goes; the following table collects the symbols used across more than one section, for reference while reading the more technical parts (Sections 3 and 5 especially). Purely local symbols, introduced and used within a single proof, are omitted.

Symbol	Meaning	First defined
Source and code structure		
$\mathcal{X}$, α	source alphabet, its cardinality	§2
$\mathcal{D}$, M	dictionary (leaf set), its size	§2
Y_i, $L(Y_i)$, $\ell(Y_i)$	i-th phrase, its length, its codeword length	§2
$H(X)$	source symbol entropy	§2
$H(Y)$	source phrase entropy	§2
$\bar{L}, \bar{\ell}$	mean phrase length, mean codeword length	§2
ρ	asymptotic compression rate, $\bar{\ell}/\bar{L}$	§2
Λ_n, Σ_n, R_n	total codeword length, total symbols, $R_n = \Lambda_n/\Sigma_n$	§2
$\mathbf{K}$, $\mathrm{spr}(\mathbf{K})$	Kraft matrix, its spectral radius	§2
Memoryless moment analysis (Section 3)		
$\mu_j(t)$	single-phrase quantity $\mathbb{E}\{[\ell(Y)]^j e^{-tL(Y)}\}$	§2
$f(t)$	$-\ln \mu_0(t)$, negative log of the MGF of $-L(Y)$	§3.5
κ_2, $c_{\ell L}$	$\mathrm{Var}\{L(Y)\}$, $\mathbb{E}\{\ell(Y)L(Y)\}$	§3.5
C	$O(1/n)$ bias coefficient, $\mathbb{E}\{R_n\} \to \rho + C/n$	§3.5
γ_1	skewness of R_n	§3.2
Markov extension (Section 5)		
X_k, S_i	underlying symbol chain, phrase-boundary state X_{Σ_i}	§5.1
$\boldsymbol{\mu}_j(t)$	matrix analogue of $\mu_j(t)$, indexed by $\mathcal{X}$	§5.2
π	stationary distribution of $\boldsymbol{\mu}_0(0)$ (row vector)	§5.2
$\mathbb{E}_\pi\{Q\}$	$\sum_s \pi_s \mathbb{E}\{Q(Y_i) \mid S_{i-1} = s\}$	§5.2
$\Lambda(t)$	dominant (Perron) eigenvalue of $\boldsymbol{\mu}_0(t)$	§5.4
$\bar{c}$	$\mathbb{E}_\pi\{L\ell\}$ (and $\bar{L}, \bar{\ell}$ reused for $\mathbb{E}_\pi\{L\}, \mathbb{E}_\pi\{\ell\}$)	§5.5
g_s	$\mathbb{E}\{L(Y_i) \mid S_{i-1} = s\}$	§5.4
$g_\ell(s)$	$\mathbb{E}\{\ell(Y_i) \mid S_{i-1} = s\}$	§5.5
w, w_ℓ	Poisson-equation solutions for the rewards L, ℓ	§5.5

3 Moments of the compression ratio for memoryless sources

This section develops the main technical machinery of this paper in two stages. In Sections 3.1–3.2, an exact formula for every integer moment $\mathbb{E}\{R_n^k\}$ is obtained. First, the case $k = 1$ is worked out in full, and then it is generalized for an arbitrary positive integer k. Second, from these exact formulas: a refined Edgeworth approximation to the CDF of R_n (Section 3.4) that improves on the classical CLT by using the skewness the third moment provides; closed-form $O(1/n)$ asymptotics for the bias and variance via Laplace's method (Section 3.5); and a large-deviations analysis of R_n's tails. A numerical validation against Monte Carlo simulation (Section 3.3) checks the exact formulas along the way. In this section the phrase pairs $(L(Y_i), \ell(Y_i))_{i \geq 1}$ are i.i.d. with the same distribution as $(L(Y), \ell(Y))$ for a generic phrase Y, as is appropriate for a DMS.

3.1 The mean compression ratio

We begin with the first moment – the expected compression ratio $\mathbb{E}\{R_n\}$ – as the quantity of primary practical interest and the one that motivates the general approach.

Theorem 3.1. *For a given DMS and a given V2V length code with a dictionary $\mathcal{D}$ and code length function $\ell(\cdot)$, the expected compression ratio of n phrases is given by*

$$\mathbb{E}\{R_n\} = n \cdot \int_0^\infty \mu_1(t) \cdot [\mu_0(t)]^{n-1} \mathrm{d}t. \tag{13}$$

Proof. We use the integral representation $1/s$ as the Laplace transform of the unit step function, that is, the identity $\frac{1}{s} = \int_0^\infty e^{-st}\mathrm{d}t$, valid for any $s > 0$. Since $\Sigma_n = \sum_{i=1}^n L(Y_i)$ is a positive integer almost surely, we have

$$R_n = \frac{\Lambda_n}{\Sigma_n} = \Lambda_n \cdot \int_0^\infty e^{-t\Sigma_n}\mathrm{d}t \quad \text{a.s.} \tag{14}$$

Taking expectations (justified here by finite-sum linearity, since Λ_n and Σ_n take only finitely many values for the finite dictionaries considered throughout this paper, so the outer expectation is itself a finite sum that commutes with the integral term by term), we obtain

$$\mathbb{E}\{R_n\} = \int_0^\infty \mathbb{E}\{\Lambda_n e^{-t\Sigma_n}\}\mathrm{d}t. \tag{15}$$

Now expand

$$\Lambda_n e^{-t\Sigma_n} = \big[\sum_{i=1}^n \ell(Y_i)\big] \cdot \exp\big[-t\sum_{i=1}^n L(Y_i)\big] = \sum_{i=1}^n \ell(Y_i) \prod_{j=1}^n e^{-tL(Y_j)}. \tag{16}$$

Taking the expectation of the i-th term and using independence of the phrase pairs:

$$\mathbb{E}\Big\{\ell(Y_i)\prod_{j=1}^n e^{-tL(Y_j)}\Big\} = \mathbb{E}\{\ell(Y_i)e^{-tL(Y_i)}\}\prod_{j\neq i}\mathbb{E}\{e^{-tL(Y_j)}\} = \mu_1(t)[\mu_0(t)]^{n-1}. \tag{17}$$

Since all n terms contribute equally (by the i.i.d. assumption), $\mathbb{E}\{\Lambda_n e^{-t\Sigma_n}\} = n \cdot \mu_1(t)[\mu_0(t)]^{n-1}$. Substituting this into (15) and integrating over t from 0 to ∞,

$$\mathbb{E}\{R_n\} = n \cdot \int_0^\infty \mu_1(t) \cdot [\mu_0(t)]^{n-1}\mathrm{d}t, \tag{18}$$

which is (13). □

3.2 Extension to general moments

The proof of Theorem 3.1 used two ingredients: the Laplace transform representation of $1/s$, and independence across phrase pairs. For general $k \geq 1$, the same two ingredients apply, but

expanding Λ_n^k now produces cross-terms between different phrases, and tracing of these requires some combinatorial bookkeeping. That bookkeeping is organized by *set partitions* of $\{1, \ldots, k\}$, defined precisely in the proof below, with a worked example for $k = 2$ and $k = 3$ included along the way.

Theorem 3.2. *For a given DMS and a given V2V length code with a dictionary $\mathcal{D}$ and code length function $\ell(\cdot)$, the k-th moment of the compression ratio of n phrases is given by*

$$\mathbb{E}\{R_n^k\} = \frac{1}{(k-1)!} \int_0^\infty t^{k-1} \sum_{\tau \in P_k} \frac{n!}{(n-|\tau|)!} [\mu_0(t)]^{n-|\tau|} \prod_{B \in \tau} \mu_{|B|}(t) \mathrm{d}t, \tag{19}$$

where P_k denotes the collection of all set partitions of $\{1, \ldots, k\}$; $|\tau|$ denotes the number of subsets associated with a partition $\tau \in P_k$; and $|B|$ denotes the cardinality of a subset B.

Proof. The integral representation of the function $\frac{1}{s^k}$ as a Laplace transform is given by

$$\frac{1}{s^k} = \frac{1}{(k-1)!} \int_0^\infty t^{k-1} e^{-st} \mathrm{d}t, \quad s > 0. \tag{20}$$

Applying this with $s = \Sigma_n$ (which is a positive integer a.s.):

$$R_n^k = \frac{\Lambda_n^k}{\Sigma_n^k} = \frac{\Lambda_n^k}{(k-1)!} \cdot \int_0^\infty t^{k-1} e^{-t\Sigma_n} \mathrm{d}t. \tag{21}$$

Taking expectations (again justified by finite-sum linearity, as above), we have

$$\mathbb{E}\{R_n^k\} = \frac{1}{(k-1)!} \int_0^\infty t^{k-1} \, \mathbb{E}\{\Lambda_n^k e^{-t\Sigma_n}\} \, \mathrm{d}t. \tag{22}$$

It remains to evaluate $\mathbb{E}\{\Lambda_n^k e^{-t\Sigma_n}\}$. Now,

$$\Lambda_n^k = \Big[\sum_{i=1}^n \ell(Y_i)\Big]^k = \sum_{(i_1,\ldots,i_k) \in \{1,\ldots,n\}^k} \prod_{j=1}^k \ell(Y_{i_j}). \tag{23}$$

Each index tuple $(i_1, \ldots, i_k)$ induces a set partition τ of the set $\{1, \ldots, k\}$ to a number of subsets. The slots a and b in $\{1, \ldots, k\}$ belong to the same subset pertaining to partition τ if and only if $i_a = i_b$. We denote by P_k for the collection of all set partitions $\{\tau\}$ of $\{1, \ldots, k\}$ (a finite set; $|P_1| = 1$, $|P_2| = 2$, $|P_3| = 5$), $|\tau|$ for the number of blocks of $\tau \in P_k$, and $|B|$ for the size of a block $B \in \tau$.[1] Because the phrase pairs $(L(Y_j), \ell(Y_j))_{j \geq 1}$ are i.i.d., the value of $\mathbb{E}\{\ell(Y_{i_1}) \cdots \ell(Y_{i_k}) e^{-t\Sigma_n}\}$ depends on the tuple $(i_1, \ldots, i_k)$ only through its induced partition τ, not on the specific phrase

[1] For example, if $k = 3$ and the tuple $(i_1, i_2, i_3) = (5, 7, 5)$, slots $a = 1$ and $b = 3$ share the phrase index 5 while slot $c = 2$ has the different index 7, so the induced partition is $\tau = \{\{1, 3\}, \{2\}\}$, with $|\tau| = 2$ subsets of sizes 2 and 1. The other tuples inducing this same τ are exactly those of the form (j, j', j) for any two *distinct* phrase indices $j \neq j'$ in $\{1, \ldots, n\}$.

indices that realize it. This is what lets us group the n^k terms of (23) by partition type, rather than evaluating each term separately. This index-partitioning technique – grouping terms of a power of a sum by which indices coincide, and exploiting exchangeability so that each group's contribution depends only on the partition it induces – is classical, in the spirit of the partition-lattice approach to moments and cumulants of sums of random variables [16]. What is needed beyond that classical identity is that the exponential weight $e^{-t\Sigma_n}$ touches all n phrases, not only the k appearing in Λ_n^k: the $n-|\tau|$ phrases outside the partition's subsets still each contribute a factor (through $\mu_0(t)$, below), and it is this joint accounting – the falling-factorial count together with the untouched phrases' contribution – that extends the classical moment-of-a-sum computation to the joint quantity $\mathbb{E}\{\Lambda_n^k e^{-t\Sigma_n}\}$ actually needed here.

Fix $\tau \in P_k$ with blocks $B_1, \ldots, B_{|\tau|}$. A tuple $(i_1, \ldots, i_k)$ induces exactly this τ if and only if it assigns the *same* phrase index to every slot within a given subset, and *different* phrase indices to slots in different subsets. So realizing τ amounts to choosing an ordered assignment of $|\tau|$ *distinct* phrase indices $j_1, \ldots, j_{|\tau|} \in \{1, \ldots, n\}$, one per subset – an injective function from the $|\tau|$ blocks to the n phrases. Assigning these indices one block at a time, B_1 may receive any of the n phrase indices, B_2 any of the remaining $n-1$ (it must differ from B_1's), B_3 any of the remaining $n-2$, and so on, until $B_{|\tau|}$, which has $n-|\tau|+1$ choices left; the number of such assignments is the falling factorial $n(n-1)\cdots(n-|\tau|+1) = \frac{n!}{(n-|\tau|)!}$.

With blocks $B_1, \ldots, B_{|\tau|}$ assigned distinct phrase indices $j_1, \ldots, j_{|\tau|}$ respectively, the product $\ell(Y_{i_1}) \cdots \ell(Y_{i_k})$ collapses to $\prod_{r=1}^{|\tau|} [\ell(Y_{j_r})]^{|B_r|}$, since subset B_r contributes $|B_r|$ copies of the same factor $\ell(Y_{j_r})$. Splitting $e^{-t\Sigma_n} = \prod_{j=1}^{n} e^{-tL(Y_j)}$ into the $|\tau|$ marked phrases $j_1, \ldots, j_{|\tau|}$ and the remaining $n-|\tau|$ phrases, and using independence across phrases:

$$\begin{aligned}\mathbb{E}\Big\{\prod_{a=1}^{k} \ell(Y_{i_a}) \cdot \prod_{j=1}^{n} e^{-tL(Y_j)}\Big\} &= \prod_{r=1}^{|\tau|} \mathbb{E}\{[\ell(Y_{j_r})]^{|B_r|} e^{-tL(Y_{j_r})}\} \cdot \prod_{j \notin \{j_1,\ldots,j_{|\tau|}\}} \mathbb{E}\{e^{-tL(Y_j)}\} \\ &= \prod_{B \in \tau} \mu_{|B|}(t) \cdot [\mu_0(t)]^{n-|\tau|}, \end{aligned} \tag{24}$$

using the i.i.d. property and the definitions of $\mu_j(t)$ and $\mu_0(t)$. This value depends only on τ (through its block sizes), not on the specific phrases $j_1, \ldots, j_{|\tau|}$ chosen to realize it – consistent with the partition-dependence noted above.

Every $\tau \in P_k$ contributes the same per-tuple expectation, as just shown, so summing over all $\frac{n!}{(n-|\tau|)!}$ tuples that induce it (the falling-factorial count derived above) and then over all $\tau \in P_k$

gives

$$\mathbb{E}\{\Lambda_n^k e^{-t\Sigma_n}\} = \sum_{\tau \in P_k} \frac{n!}{(n-|\tau|)!} [\mu_0(t)]^{n-|\tau|} \prod_{B \in \tau} \mu_{|B|}(t). \tag{25}$$

Substituting into (22) gives (19). □

For $k = 2$ and $k = 3$, the integrals are explicitly:

$$\mathbb{E}\{R_n^2\} = \int_0^\infty t\big(n\mu_2(t)[\mu_0(t)]^{n-1} + n(n-1)[\mu_1(t)]^2[\mu_0(t)]^{n-2}\big)\mathrm{d}t, \tag{26}$$

$$\mathbb{E}\{R_n^3\} = \frac{1}{2}\int_0^\infty t^2\big(n\mu_3(t)[\mu_0(t)]^{n-1} + 3n(n-1)\mu_2(t)\mu_1(t)[\mu_0(t)]^{n-2} \tag{27}$$

$$+ n(n-1)(n-2)[\mu_1(t)]^3[\mu_0(t)]^{n-3}\big)\mathrm{d}t. \tag{28}$$

From these three moments one obtains the mean, variance, and skewness of R_n via $\mathrm{Var}\{R_n\} = \mathbb{E}\{R_n^2\} - (\mathbb{E}\{R_n\})^2$ and $\gamma_1 = (\mathbb{E}\{R_n^3\} - 3\mathbb{E}\{R_n\}\mathbb{E}\{R_n^2\} + 2(\mathbb{E}\{R_n\})^3)/(\mathrm{Var}\{R_n\})^{3/2}$.

3.3 Validation

We now validate (19) on a simple V2V length code: a binary memoryless source with $P(0) = 1 - P(1) = p = 0.8$, parsed by the Tunstall tree with dictionary $\{00, 01, 1\}$, and coded with the Huffman assignment $(\ell(00), \ell(01), \ell(1)) = (1, 2, 2)$. The same source, tree, and codeword assignment are used again for the bias-decomposition example in Section 4. This is a complete prefix code: $2^{-1} + 2^{-2} + 2^{-2} = 1$, so Kraft's inequality holds with equality, and every codeword length is a positive integer. The phrase probabilities are $Q(00) = p^2 = 0.64$, $Q(01) = p(1-p) = 0.16$, $Q(1) = 1 - p = 0.2$, giving $\bar{L} = 1.8$, $\bar{\ell} = 1.36$, and $\rho = \bar{\ell}/\bar{L} = 0.7556$.

For this model, $\mu_j(t) = \mathbb{E}\{[\ell(Y)]^j e^{-tL(Y)}\}$ is the explicit three-term sum

$$\mu_j(t) = 0.64 \cdot 1^j e^{-2t} + 0.16 \cdot 2^j e^{-2t} + 0.2 \cdot 2^j e^{-t}, \tag{29}$$

so, e.g., $\mu_0(t) = 0.8e^{-2t} + 0.2e^{-t}$ and $\mu_1(t) = 0.96e^{-2t} + 0.4e^{-t}$. The three integrals (13), (26), (27) are evaluated by standard numerical quadrature (Gaussian quadrature after the substitution $t = u/n$ to handle the concentration near $t = 0$ for larger n).

At $n = 50$ phrases, the exact formulas give:

$$\mathbb{E}\{R_{50}\} = 0.7571, \quad \mathrm{Var}\{R_{50}\} = 0.003230, \quad \gamma_1 = 0.2878. \tag{30}$$

For validation, we performed a Monte Carlo simulation of 1,000,000 independent realizations of 50 phrase pairs $(L(Y_i), \ell(Y_i))$, drawn according to the three phrase probabilities above, computing the

realized ratio $R_{50} = \Lambda_{50}/\Sigma_{50}$ for each. The empirical estimates are:

$$\widehat{\mathbb{E}\{R_{50}\}} = 0.7571, \quad \widehat{\mathrm{Var}\{R_{50}\}} = 0.003226, \quad \widehat{\gamma_1} = 0.2875. \tag{31}$$

Agreement is to three or four significant figures throughout – the skewness γ_1 in particular is the first quantity that (27) provides beyond what a first-order approximation would give – and the exact formulas require no simulation effort at all.

3.4 The Edgeworth approximation

The three exact moment formulas (13)–(27) can be used to construct a sharper approximation to the CDF of R_n than the ordinary CLT (Gaussian) approximation. By the bivariate CLT, the pair $(\Lambda_n - n\bar{\ell}, \Sigma_n - n\bar{L})/\sqrt{n}$ converges in distribution to a bivariate normal as $n \to \infty$. By the delta method [1, Theorem 3.1], any function that is continuously differentiable at the limiting mean again yields a normal limit, with asymptotic variance determined by the gradient of the function and the full limiting covariance matrix. Applied to the map $(\lambda, m) \mapsto \lambda/m$, which is continuously differentiable at $(\bar{\ell}, \bar{L})$ since $\bar{L} > 0$, this implies that $Z_n \triangleq (R_n - \mathbb{E}\{R_n\})/\sqrt{\mathrm{Var}\{R_n\}}$ converges in distribution to a standard normal $\mathcal{N}(0,1)$. The CLT approximation to the CDF $F(x) \triangleq \Pr\{R_n \le x\}$ is therefore $\Phi_{\mathrm{nor}}(z)$, where $z = (x - \mathbb{E}\{R_n\})/\sqrt{\mathrm{Var}\{R_n\}}$ and Φ_{nor} is the standard normal CDF. But this approximation ignores the skewness of the distribution, which is $O(1/\sqrt{n})$ and non-negligible at moderate n.

The one-term Edgeworth expansion corrects for the skewness. It approximates F by

$$F_{\mathrm{Edgeworth}}(x) \triangleq \Phi_{\mathrm{nor}}(z) - \varphi(z)\,\frac{\gamma_1}{6}(z^2 - 1), \tag{32}$$

where $z = (x - \mathbb{E}\{R_n\})/\sqrt{\mathrm{Var}\{R_n\}}$, $\varphi = \Phi'_{\mathrm{nor}}$ is the standard normal density, and γ_1 is the skewness of R_n. The approximation (32) is a standard result in the theory of Edgeworth expansions for sums of i.i.d. random variables [17, Ch. XVI]; its validity for the ratio $R_n = \Lambda_n/\Sigma_n$ follows from the delta method applied to the bivariate CLT for (Λ_n, Σ_n), under standard moment conditions on the phrase pair $(L(Y), \ell(Y))$. The key point is that all three parameters entering (32) – $\mathbb{E}\{R_n\}$, $\mathrm{Var}\{R_n\}$, and γ_1 – are available from (13)–(27) for every finite n.

Continuing the same numerical example as in Subsection 3.3, Table 1 shows the absolute approximation error $|F(x) - F_{\mathrm{Edgeworth}}(x)|$ for $n = 50$, at several standardized values z, comparing the CLT approximation against (32) with the exact $\gamma_1 = 0.2878$:

The Edgeworth correction reduces the approximation error at every tabulated point, by a factor ranging from about 3 to nearly 200 depending on z; averaged over the range shown, the

z	x	CLT error	Edgeworth error
-2.0	0.6434	0.00773	0.00004
-0.5	0.7287	0.01881	0.00614
0.0	0.7571	0.01972	0.00058
0.5	0.7855	0.01111	0.00155
2.0	0.8708	0.00574	0.00203

Table 1: Absolute CDF approximation error at $n = 50$, against empirical CDF from 2,000,000 Monte Carlo trials.

mean absolute error drops by a factor of about 3.4 and the maximum error by a factor of about 2.7. This is a more modest improvement than a smooth, continuously-supported distribution would show (compare Remark 3.1 below), since $L(Y)$ and $\ell(Y)$ each take only two values here, but it is a genuine improvement for an actual code.

Figure 1 shows this comparison as a continuous function of z rather than at the five tabulated points, against a 20,000,000-trial Monte Carlo baseline. The CLT error traces out a broad envelope peaking near $z = 0$, while the Edgeworth error stays consistently lower across the entire range shown. Both curves show some oscillation rather than a perfectly smooth profile; this is a genuine finite-alphabet effect discussed in Remark 3.1 below.

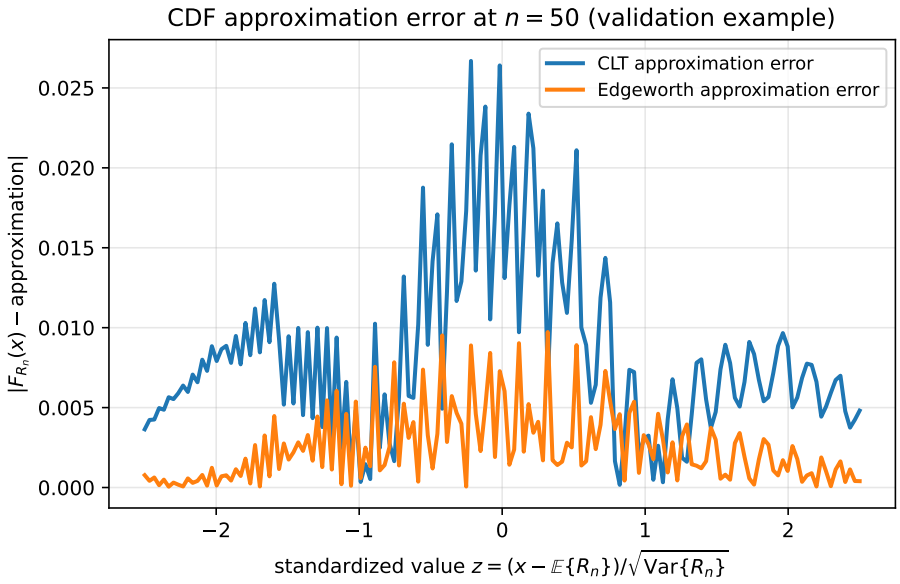

Figure 1: Absolute CDF approximation error, CLT vs. Edgeworth (32), as a function of the standardized value z, for the validation example of Section 3.3 at $n = 50$. Ground truth is a 20,000,000-trial Monte Carlo simulation.

Remark 3.1 (A finite-alphabet effect)**.** *Because $L(Y) \in \{1, 2\}$ and $\ell(Y) \in \{1, 2\}$ each take only*

two values, Λ_n and Σ_n are both sums of small integers, so $R_n = \Lambda_n/\Sigma_n$ takes values in a finite, coarsely-spaced set of rationals for any fixed n – its exact CDF is a step function with comparatively few, comparatively large jumps. Neither the CLT nor the Edgeworth approximation is built to track a step function's exact jumps (both are continuous-x approximations, designed for the lattice spacing to vanish as $n \to \infty$), so both error curves in Figure 1 inherit some oscillation from this discreteness, on top of their genuine approximation error. This is a real feature of finite, small-dictionary V2V codes, not an artifact: a synthetic phrase-length law with a smooth, unbounded, or finely-spaced support – as used in an earlier version of this example – would mask it entirely. The Edgeworth correction still reduces the error at every point tested, as Table 1 shows; it is simply a less clean improvement than the idealized, non-lattice case.

3.5 Asymptotic Approximation via the Laplace method of integration

For large n, the integrands in (19) are sharply concentrated near $t = 0$ and can be evaluated asymptotically by the Laplace method of integration. The change of integration variabe $t = u/n$ transforms the integrals into a form where the concentration is explicit and a systematic expansion in $1/n$ can be carried out. There are two versions of Laplace's method for an integral $\int_a^b g(t)e^{-nf(t)}\mathrm{d}t$, depending on where f attains its minimum over the domain of integration. If the minimum occurs at an *interior* point $t_0 \in (a, b)$ with $f'(t_0) = 0$, the local behavior of f near t_0 is quadratic, and so, $e^{-nf(t)}$ is locally approximated by a Gaussian, $e^{-nf(t_0)} \cdot \exp\left\{-\frac{nf''(t_0)}{2}(t-t_0)^2\right\}$, giving the familiar $\sqrt{\frac{2\pi}{nf''(t_0)}}$ prefactor. If instead the minimum occurs at a *boundary* point of the domain (a or b) with $f'(t_0) \neq 0$, the local behavior of f is linear rather than quadratic, and $e^{-nf(t)}$ is locally approximated at the vicinity of t_0 by a plain exponential, $e^{-nf(t_0)}e^{-nf'(t_0)(t-t_0)}$, rather than a Gaussian; its moments against a power series in $(t - t_0)$ are then elementary Gamma-function integrals rather than Gaussian moments (see, e.g., [18, Sec. 4.3] or [19, p. 48]). This second, boundary case is the one relevant here: as shown below, $f(t) \triangleq -\ln \mu_0(t)$ attains its minimum over $t \in [0, \infty)$ at the boundary point $t_0 = 0$ with $f'(0) = \bar{L} \neq 0$, so $[\mu_0(t)]^n = e^{-nf(t)}$ is approximated near $t = 0$ by the exponential $e^{-n\bar{L}t}$. The derivation below carries this out to one further order in $1/n$ than the leading term alone provides, since the bias coefficient C requires the $O(1/n)$ correction.

Let $f(t) = -\ln \mu_0(t)$ denote the negative cumulant generating function (CGF) of $-L(Y)$. Since $L(Y) \geq 1$ a.s., $f(0) = 0$ and $f'(0) = \mathbb{E}\{L(Y)\} = \bar{L} > 0$. Furthermore, f is strictly concave as $f''(t) < 0$ for all $t \geq 0$. In particular, $f(t) > 0$ for all $t > 0$, so $[\mu_0(t)]^{n-1} = e^{-(n-1)f(t)} \to 0$ exponentially fast in n for every fixed $t > 0$. The integrand of (13), which is proportional

to $\mu_1(t)[\mu_0(t)]^{n-1}$, therefore concentrates near $t_0 = 0$ as $n \to \infty$. Substituting $t = u/n$ gives $[\mu_0(u/n)]^{n-1} = e^{-(n-1)f(u/n)} \approx e^{-\bar{L}u}$ (to leading order for large n), and after the substitution the integral becomes $\int_0^\infty \mu_1(u/n)e^{-(n-1)f(u/n)}\mathrm{d}u$, which is $O(1)$.

The bias term. Write $\bar{L} = \mathbb{E}\{L(Y)\}$, $\kappa_2 = \mathrm{Var}\{L(Y)\}$, $\bar{\ell} = \mathbb{E}\{\ell(Y)\}$, $c_{\ell L} = \mathbb{E}\{\ell(Y)L(Y)\}$. The $O(1/n)$ expansion of $\mathbb{E}\{R_n\}$ follows from a general fact about this boundary case of the Laplace method, stated here and proved in Appendix A (see also [18, Sec. 4.3] and [19, p. 48] for the general theory).

Lemma 3.1 (Boundary Laplace expansion)**.** *Let $f : [0,\infty) \to [0,\infty)$ satisfy $f(0) = 0$, be twice continuously differentiable at $t = 0$ with $f'(0) = a$ for some $a > 0$ and $f''(0) = -b$, and let h be continuously differentiable at $t = 0$ with $h(0) = h_0$, $h'(0) = h_1$ (both suitably regular for the expansion below to hold, as verified for $f = -\ln \mu_0$ above). Then, for any integers $k \geq 1$ and $m \geq 1$, as $n \to \infty$,*

$$\begin{aligned} &\frac{n!}{(n-m)!}\int_0^\infty t^{k-1}h(t)e^{-(n-m)f(t)}\mathrm{d}t \\ = \;& n^{m-k}\left[\frac{(k-1)!h_0}{a^k} + \frac{(k-1)!}{2na^{k+2}}\Big(a^2h_0m(2k-m+1) + 2ah_1k + bh_0k(k+1)\Big) + O\left(\frac{1}{n^2}\right)\right]. \end{aligned}$$

The case $k = m = 1$, $h = g$ reduces to

$$n \cdot \int_0^\infty g(t)e^{-(n-1)f(t)}\mathrm{d}t = \frac{g_0}{a} + \frac{a^2g_0 + ag_1 + bg_0}{na^3} + O\left(\frac{1}{n^2}\right),$$

the form used just below.

Applying Lemma 3.1 with $f(t) = -\ln \mu_0(t)$ and $g = \mu_1$: from the previous paragraph, $a = \bar{L}$ and $b = \kappa_2$; and since $\mu_1(0) = \mathbb{E}\{\ell(Y)\} = \bar{\ell}$ and

$$\mu_1'(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\{\ell(Y)e^{-tL(Y)}\} = -\mathbb{E}\{L(Y)\ell(Y)e^{-tL(Y)}\}, \tag{33}$$

so $\mu_1'(0) = -c_{\ell L}$, we have $g_0 = \bar{\ell}$ and $g_1 = -c_{\ell L}$. Substituting into (33),

$$\mathbb{E}\{R_n\} = n \cdot \int_0^\infty \mu_1(t)[\mu_0(t)]^{n-1}\mathrm{d}t = \frac{\bar{\ell}}{\bar{L}} + \frac{1}{n}\cdot\frac{\bar{L}^2\bar{\ell} - \bar{L}c_{\ell L} + \kappa_2\bar{\ell}}{\bar{L}^3} + O\left(\frac{1}{n^2}\right) = \rho + \frac{C}{n} + O\left(\frac{1}{n^2}\right) \tag{34}$$

with

$$C = \frac{\bar{\ell}(\bar{L}^2 + \kappa_2) - c_{\ell L}\bar{L}}{\bar{L}^3}. \tag{35}$$

We state this as Proposition 3.1 below.

Proposition 3.1 (Asymptotic bias)**.**

$$\lim_{n\to\infty} n \cdot (\mathbb{E}\{R_n\} - \rho) = \frac{\bar{\ell}(\bar{L}^2 + \kappa_2) - c_{\ell L}\bar{L}}{\bar{L}^3} \triangleq C, \tag{36}$$

where $\rho = \bar{\ell}/\bar{L}$.

A comment is in order concerning an alternative route for calculating the bias – the delta-method. Since $\bar{U}_n := \Sigma_n/n$ and $\bar{V}_n := \Lambda_n/n$ are exact sample means of i.i.d. data, $\mathbb{E}\{\bar{U}_n\} = \bar{L}$ and $\mathbb{E}\{\bar{V}_n\} = \bar{\ell}$ exactly, for every n. Writing $g(u, v) := v/u$ and Taylor-expanding $R_n = g(\bar{U}_n, \bar{V}_n)$ to second order around $(\bar{L}, \bar{\ell})$, the first-derivative terms vanish upon taking expectations, since the sample means are exactly unbiased – so the entire bias comes from the second derivatives:

$$C = \frac{1}{2}\Big(g_{uu}\mathrm{Var}\{L\} + 2g_{uv}\mathrm{Cov}\{L, \ell\}\Big) = \frac{\rho\,\mathrm{Var}\{L\} - \mathrm{Cov}\{L, \ell\}}{\bar{L}^2}, \tag{37}$$

using $g_{uu} = 2\rho/\bar{L}^2$, $g_{uv} = -1/\bar{L}^2$, $g_{vv} = 0$ at $(\bar{L}, \bar{\ell})$ – algebraically identical to (36), with $\mathrm{Var}\{L\} = \kappa_2$ and $\mathrm{Cov}\{L, \ell\} = c_{\ell L} - \bar{L}\bar{\ell}$.

This is a useful independent cross-check, but not a substitute for the derivation above. The delta method, by construction, gives only the *leading* $O(1/n)$ correction: it starts from an asymptotic expansion of R_n itself and stops at second order once that correction is in hand, since the first-derivative terms vanish identically and nothing forces the expansion any further. The Laplace-method route above instead starts from the *exact* formula (13), valid at every finite n, and reaches the bias by expanding *that* formula asymptotically – so the same machinery, carried one order further, would give the $O(1/n^2)$ term too, and (via Theorem 3.2 and Lemma 3.1 applied repeatedly, as just below) gives every higher moment – variance, skewness, and beyond – from a single combinatorial formula, rather than requiring a fresh, increasingly delicate multivariate Taylor expansion for each one in turn. It also extends uniformly to the Markov case of Section 5: there, the delta method's natural generalization requires tracking how a phrase's own fluctuation correlates with the state it leaves the chain in – a genuine complication with no analogue here – whereas an eigenvalue-perturbation argument handles this automatically, by perturbing the joint spectral object directly rather than decomposing the bias into pieces by hand.

The variance term. Lemma 3.1, applied twice more, yields the asymptotic variance directly. Write $\mathbb{E}\{R_n^2\} = I_1 + I_2$ where

$$I_1 = \int_0^\infty tn\mu_2(t)[\mu_0(t)]^{n-1}\mathrm{d}t, \quad I_2 = \int_0^\infty tn(n-1)[\mu_1(t)]^2[\mu_0(t)]^{n-2}\mathrm{d}t, \tag{38}$$

each matching the Lemma's left-hand side with $k = 2$ (since the extra factor is $t = t^{k-1}$). For I_1, $m = 1$ and $h = \mu_2$, so $h_0 = \mu_2(0) = s_{\ell 2}$ where $s_{\ell 2} := \mathbb{E}\{[\ell(Y)]^2\}$; since I_1 itself turns out to contribute only at order $1/n$ to $\mathbb{E}\{R_n^2\}$, only the Lemma's leading term is needed:

$$I_1 = \frac{s_{\ell 2}}{n\bar{L}^2} + O\left(\frac{1}{n^2}\right). \tag{39}$$

For I_2, $m = 2$ and $h = [\mu_1]^2$, so $h_0 = [\mu_1(0)]^2 = \bar{\ell}^2$ and, by the product rule, $h_1 = 2\mu_1(0)\mu_1'(0) = -2\bar{\ell}\, c_{\ell L}$; since I_2 contributes at leading order $O(1)$, both terms of the Lemma are needed:

$$I_2 = \frac{\bar{\ell}^2}{\bar{L}^2} + \frac{1}{n} \cdot \frac{\bar{\ell}\big(3\bar{L}^2\bar{\ell} - 4\bar{L}\, c_{\ell L} + 3\kappa_2\bar{\ell}\big)}{\bar{L}^4} + O\left(\frac{1}{n^2}\right). \tag{40}$$

Proposition 3.2 (Asymptotic variance)**.** *With $s_{\ell 2} = \mathbb{E}\{[\ell(Y)]^2\}$:*

$$\lim_{n\to\infty} n \cdot \mathrm{Var}\{R_n\} = \frac{\bar{L}^2(s_{\ell 2} - \bar{\ell}^2) - 2\bar{L}\bar{\ell}(c_{\ell L} - \bar{L}\bar{\ell}) + \kappa_2\bar{\ell}^2}{\bar{L}^4}. \tag{41}$$

Proof. Adding I_1 and I_2 above gives $\mathbb{E}\{R_n^2\}$ to $O(1/n)$; subtracting $(\mathbb{E}\{R_n\})^2 = \rho^2 + 2\rho C/n + O(n^{-2})$ (squaring Proposition 3.1) and multiplying by n, the $O(1)$ terms cancel exactly (since $\bar{\ell}^2/\bar{L}^2 = \rho^2$), leaving (41) after collecting the remaining terms over the common denominator $\bar{L}^4$. □

On the validation example of Section 3.3, the analytic formulas of Propositions 3.1 and 3.2 evaluate to a bias coefficient of 0.076818 and an asymptotic scaled variance of 0.159000. The exact formula (19) evaluated at large n gives $n(\mathbb{E}\{R_n\} - \rho) \to 0.076818$ and $n\mathrm{Var}\{R_n\} \to 0.159000$, matching to six significant figures and confirming both propositions numerically.

Remark 3.2 (Design implication)**.** *Proposition 3.1 shows that for a fixed parsing tree (hence fixed $\bar{L}$ and $\mathrm{Var}\{L\}$), the bias $\mathbb{E}\{R_n\} - \rho$ is minimized (i.e., $\mathbb{E}\{R_n\}$ converges to ρ fastest) when $\mathrm{Cov}\{L, \ell\}$ is maximized. This has a clear design interpretation: longer phrases should be assigned longer codewords, i.e., $\mathrm{Cov}\{L, \ell\}$ should be made positive. Whether a specific codeword assignment achieves this depends on the source and tree; the bias formula gives a precise, finite-n quantification of how far any given code deviates from the optimum.*

3.6 Large deviations of the compression ratio

The Edgeworth expansion of Section 3.4 characterizes the behavior of R_n in the $O(1/\sqrt{n})$ fluctuation region around ρ. For deviations of fixed size $\Delta > 0$ (independent of n), the probability $\Pr\{R_n > \rho + \Delta\}$ decays exponentially in n, and its exact exponential rate follows directly from Cramér's theorem.

The key observation is that the event $\{R_n > \rho + \Delta\}$ is equivalent to a standard large-deviations event for a sum of i.i.d. random variables. Since

$$R_n > \rho + \Delta \iff \frac{\Lambda_n}{\Sigma_n} > \rho + \Delta \iff \sum_{i=1}^{n} \big[\ell(Y_i) - (\rho + \Delta) L(Y_i)\big] > 0, \tag{42}$$

the event is determined by whether the partial sum of the i.i.d. random variables $Z_i^{(\Delta)} := \ell(Y_i) - (\rho + \Delta) L(Y_i)$ exceeds zero. Since $\mathbb{E}\{Z_i^{(\Delta)}\} = \mathbb{E}\{\ell(Y)\} - (\rho + \Delta)\mathbb{E}\{L(Y)\} = -\Delta \mathbb{E}\{L(Y)\} < 0$ for $\Delta > 0$, this is a large-deviations event with the sum crossing zero against its drift. An identical reduction handles the lower tail: $\{R_n < \rho - \Delta\} \iff \sum_i [\ell(Y_i) - (\rho - \Delta) L(Y_i)] < 0$, with $\mathbb{E}\{\ell(Y) - (\rho - \Delta) L(Y)\} = \Delta \mathbb{E}\{L(Y)\} > 0$.

By Cramér's theorem, for every $\Delta > 0$ (all logarithms in this subsection are natural logarithms):

$$\frac{1}{n} \log \Pr\{R_n > \rho + \Delta\} \to -I_+(\Delta), \tag{43}$$

$$\frac{1}{n} \log \Pr\{R_n < \rho - \Delta\} \to -I_-(\Delta), \tag{44}$$

where the rate functions are

$$I_+(\Delta) = \sup_{\theta > 0} \Big\{ -\log \mathbb{E}\big\{ e^{\theta[\ell - (\rho + \Delta) L]} \big\} \Big\}, \tag{45}$$

$$I_-(\Delta) = \sup_{\theta < 0} \Big\{ -\log \mathbb{E}\big\{ e^{\theta[\ell - (\rho - \Delta) L]} \big\} \Big\}. \tag{46}$$

For a real parameter θ and constant c, let $P_\theta^{(c)}$ denote the exponentially tilted law of (L, ℓ) obtained by weighting each realization proportionally to $e^{\theta(\ell - cL)}$, and write $\mathbb{E}_\theta^{(c)}\{\cdot\}$ for the corresponding expectation; below, c is $\rho + \Delta$ for the upper tail and $\rho - \Delta$ for the lower tail, and we abbreviate $P_\theta^{(c)}$ and $\mathbb{E}_\theta^{(c)}$ by P_θ and $\mathbb{E}_\theta$ since c is clear from context. The supremum over $\theta > 0$ in I_+ is unconstrained (the CGF is finite for all $\theta > 0$ provided ℓ has all exponential moments, which holds for a finite alphabet); for a finite dictionary L is bounded, so the CGF of $\ell - (\rho - \Delta) L$ is finite for all $\theta \in \mathbb{R}$, and the supremum over $\theta < 0$ in I_- is unconstrained; for codes with unbounded phrase lengths, the supremum may be constrained to an interval $(\theta_{\min}, 0)$. The supremum in each case is achieved at a unique θ^*. For the upper tail, $\theta^* > 0$ is the unique positive root of

$$\frac{\mathbb{E}\{[\ell - (\rho + \Delta) L]\, e^{\theta^*[\ell - (\rho + \Delta) L]}\}}{\mathbb{E}\{e^{\theta^*[\ell - (\rho + \Delta) L]}\}} = 0, \tag{47}$$

and for the lower tail, $\theta^* < 0$ is the unique negative root of

$$\frac{\mathbb{E}\{[\ell - (\rho - \Delta) L]\, e^{\theta^*[\ell - (\rho - \Delta) L]}\}}{\mathbb{E}\{e^{\theta^*[\ell - (\rho - \Delta) L]}\}} = 0. \tag{48}$$

In both cases the condition expresses that the compression ratio under the tilted distribution P_{θ^*} equals the target level: $\mathbb{E}_{\theta^*}\{\ell\}/\mathbb{E}_{\theta^*}\{L\} = \rho + \Delta$ (upper tail) or $\rho - \Delta$ (lower tail). Thus P_{θ^*} is the unique exponential tilt of the original phrase-pair law under which the large-deviations event is typical.

The rate functions I_+, I_- are not built from a separate toolkit: $\mathbb{E}\{e^{\theta[\ell-(\rho+\Delta)L]}\} = \mathbb{E}\{e^{\theta\ell - tL}\}$ along the ray $t = \theta(\rho + \Delta)$ (and similarly for I_- along $t = \theta(\rho - \Delta)$), and this joint exponential moment is exactly the object that reduces to $\mu_0(t)$ at $\theta = 0$ and to $\mu_1(t)$ upon differentiating in θ at $\theta = 0$ (Section 2). The exact moments of Sections 3–3.5 and the large-deviations rate functions here are two different slices of this same underlying two-parameter family, evaluated in different regimes: t alone for the moments, and along the rays above for the tails.

4 Comparison with V2F and F2V length codes

The three code families differ in which of the phrase length L and codeword length ℓ they allow to vary: F2V fixes L, V2F fixes ℓ, and V2V lets both vary. The fact that V2V can match or beat the other two on the asymptotic rate ρ is not surprising – it strictly contains them as special cases. The more informative question is what this extra freedom does to the *finite-n* behavior captured by C, and what, if anything, it reveals about why some V2V constructions substantially outperform both alternatives while others do not.

Setup. Fix a binary memoryless source with $p = P(0) = 0.8$ ($H = 0.722$ bits/symbol) and the Tunstall tree $\{00, 01, 1\}$ ($\bar{L} = 1.8$, $\mathrm{Var}\{L\} = 0.16$).

The structural decomposition. The bias formula of Proposition 3.1 rewrites as

$$C = \frac{\rho\,\mathrm{Var}\{L\} - \mathrm{Cov}\{L, \ell\}}{\bar{L}^2}, \tag{49}$$

which separates cleanly by code family:

- **F2V**: L is deterministic, so $\mathrm{Var}\{L\} = \mathrm{Cov}\{L, \ell\} = 0$ and $C = 0$ identically – not a design achievement, but a triviality of having no phrase-length randomness to create bias from. It buys nothing for ρ.
- **V2F**: ℓ is constant, so $\mathrm{Cov}\{L, \ell\} = 0$ always, giving $C = \rho\mathrm{Var}\{L\}/\bar{L}^2 > 0$ with no freedom to reduce it: the fixed codeword length that limits ρ to $O(1/\bar{L})$ redundancy is the same constraint that locks in this bias.

- **V2V**: both vary, so $\mathrm{Cov}\{L, \ell\}$ is a genuine design variable. Correlating L and ℓ positively – longer phrases getting longer codewords – reduces C below what V2F could achieve with the same tree and the same $\mathrm{Var}\{L\}$.

This freedom cuts both ways, though: the Huffman codeword assignment that *minimizes* ρ need not create positive covariance. For the tree above, Huffman assigns the shortest codeword to the most probable phrase (00, $P = 0.64$), which happens to be one of the *longest* phrases ($L = 2$), giving $\mathrm{Cov}\{L, \ell\} = -0.128 < 0$ and $C_{\mathrm{V2V}} = 0.077$ – *larger* than $C_{\mathrm{V2F}} = 0.055$ despite $\rho_{\mathrm{V2V}} = 0.756 < \rho_{\mathrm{V2F}} = 1.111$. The same tree admits a different codeword assignment (giving the shortest codeword to the *shortest* phrase instead) that achieves $\mathrm{Cov}\{L, \ell\} = +0.160$ and $C = 0$ exactly, at the cost of a worse $\rho = 1$. V2F and F2V have no such choice to make; V2V's freedom is precisely the ability to trade between these two objectives, in either direction.

Why this matters: the Khodak code. The tension above raises the question of whether ρ and C can be improved *together* rather than traded off, and the Khodak code [3, 15] shows that they can. By the conservation of entropy [20], $H(Y) = \bar{L}\, H$ for any parsing tree, where $H(Y)$ is the phrase entropy; the Khodak construction assigns near-Shannon codewords, $\ell(y) \approx -\log_2 Q(y)$, to a tree deliberately built to preserve non-uniform leaf probabilities. By the asymptotic equipartition property, a *typical* phrase of length $L(y)$ satisfies $Q(y) \approx 2^{-L(y)H}$, so $\ell(y) \approx L(y)H$ holds for the typical phrases that carry essentially all the probability mass as $L(y)$ grows – not for every individual leaf (a phrase consisting entirely of the single most probable symbol, for instance, gets a codeword far shorter than $L(y)H$), but for enough of the distribution to drive the averages below. This achieves redundancy $r := \rho - H = O(\bar{L}^{-5/3})$ – strictly better than the $O(1/\bar{L})$ available to V2F or to Tunstall–Huffman. The same typical-phrase approximation gives $\mathrm{Cov}\{L, \ell\} \approx H\,\mathrm{Var}\{L\}$, so substituting into (49),

$$C \;\approx\; \frac{\rho \mathrm{Var}\{L\} - H\mathrm{Var}\{L\}}{\bar{L}^2} \;=\; \frac{(\rho - H)\mathrm{Var}\{L\}}{\bar{L}^2} \;=\; \frac{r\,\mathrm{Var}\{L\}}{\bar{L}^2} \;=\; O(r) \;=\; O(\bar{L}^{-5/3}). \tag{50}$$

The bias coefficient shrinks *at the same rate as the redundancy* – not a second, independent achievement, but a direct consequence of the same design principle. Compare V2F: ℓ cannot track L at all, so $\mathrm{Cov}\{L, \ell\} = 0$ is permanent regardless of how the tree is built, and neither the $O(1/\bar{L})$ floor on r nor the resulting $C = \rho\mathrm{Var}\{L\}/\bar{L}^2$ can be improved by this mechanism.

For a binary Bernoulli(2/3) source, the explicit Bugeaud–Drmota–Szpankowski construction [15] (convergent denominator $q = 5$) gives $\bar{L} = 116.4$, $\rho = 0.9211$ against $H = 0.9183$ ($r = 0.0028$),

and $\mathrm{Var}\{L\} = 4824$; (50) predicts $C \approx 0.00097$, close to the exact value $C = 0.000962$ from Proposition 3.1 ($\mathrm{Cov}\{L, \ell\} = 4430$ against the predicted $H\mathrm{Var}\{L\} = 4430$). Table 2 confirms the resulting $O(1/n)$ convergence numerically, using the exact moment formula (13) at every n:

n	Exact $\mathbb{E}\{R_n\}$	Asymp. $\rho + C/n$	Error (approx.$-$exact)
1	0.91994	0.92206	$+0.00213$
2	0.92269	0.92158	-0.00111
5	0.92144	0.92129	-0.00015
10	0.92123	0.92120	-0.000033
20	0.92116	0.92115	-0.0000082
50	0.92112	0.92112	-0.0000013
100	0.92111	0.92111	≈ 0
∞	$\rho = 0.92110$	$\rho = 0.92110$	0

Table 2: Exact vs. first-order asymptotic $\mathbb{E}\{R_n\}$ for the explicit Khodak code of [15], illustrating the $O(1/n)$ convergence rate of Proposition 3.1.

The approximation is accurate to five decimal places by $n = 50$, well before the asymptotic regime is actually reached.

5 Extension to Markov sources

This section repeats the memoryless program of Section 3 with the i.i.d. assumption dropped: a boundary state is identified that renders successive phrases Markov rather than independent (§5.1); the scalar quantities $\mu_j(t)$ become matrices indexed by that state (§5.2); the mean formula of Theorem 3.1 becomes a matrix product (§5.4, validated against direct simulation); and the Laplace-method bias analysis of Section 3.5 becomes a matrix-eigenvalue perturbation, with the correction term now characterized by a Poisson equation rather than a plain derivative (§5.5, with a pointer to the underlying argument in Appendix B. The memoryless case reappears throughout as an exact special case, not a separate limit.

We now extend the source model to the Markov case. Specifically, in this section, $X_1, X_2, \ldots$ is assumed an irreducible, time-homogeneous, *first-order* Markov chain on the alphabet $\mathcal{X}$ of cardinality α, with transition probabilities $\Pr\{X_{k+1} = x' \mid X_k = x\}$, and initial state X_0. The parsing tree, dictionary, and codeword assignment are exactly as before.

5.1 Source model: the boundary state S_i

Let $\Sigma_i = L(Y_1) + \cdots + L(Y_i)$ denote the total number of source symbols consumed through the i-th phrase (with $\Sigma_0 \triangleq 0$, consistent with the Σ_n of Section 2), so phrase Y_i consists of the symbols

$X_{\Sigma_{i-1}+1}, \ldots, X_{\Sigma_i}$. We define the state of the source at the i-th phrase boundary as the last symbol consumed before that boundary, that is,

$$S_i \triangleq X_{\Sigma_i}, \qquad i = 0, 1, 2, \ldots \tag{51}$$

and so $S_0 = X_0$. Thus the state set is $\mathcal{X}$ itself, of size α; S_{i-1} is the symbol in force when phrase Y_i begins, and S_i is the symbol reached when phrase Y_i ends and the parser resets to the root of the parsing tree.

The initial state S_0 is taken to have the stationary distribution π of the phrase-boundary chain $\{S_i\}_{i\geq 0}$ just defined (shown to be a genuine Markov chain in Lemma 5.1 below) – a distribution generally different from the ordinary stationary distribution of $\{X_k\}_{k\geq 0}$ itself, since phrase boundaries sample states at variable, state-dependent intervals rather than at every symbol (Section 5.4 gives a numerical instance of this gap).

Since phrase Y_i is parsed by running the parsing tree against the symbols $X_{\Sigma_{i-1}+1}, X_{\Sigma_{i-1}+2}, \ldots$ until a leaf is reached, the triple $(L(Y_i), \ell(Y_i), S_i)$ is a measurable function of $S_{i-1} = X_{\Sigma_{i-1}}$ together with the (as yet unconsumed) continuation of the chain from time Σ_{i-1} onward. This lets the Markov property of $\{X_k\}_{k\geq 0}$ pass through the parsing recursion to the phrase level:

Lemma 5.1 (Markov property at phrase boundaries)**.** *Conditioned on S_{i-1}, the triple $(L(Y_i), \ell(Y_i), S_i)$ is independent of $(S_0, \ldots, S_{i-2}, Y_1, \ldots, Y_{i-1})$, with conditional law depending on S_{i-1} alone – and, by time-homogeneity, the same function of S_{i-1} for every i, not merely free of the extra history at each fixed i. Consequently $\{S_i\}_{i\geq 0}$ is a time-homogeneous Markov chain on $\mathcal{X}$.*

Proof. $(L(Y_i), \ell(Y_i), S_i)$ is a function of $S_{i-1} = X_{\Sigma_{i-1}}$ and the continuation $(X_{\Sigma_{i-1}+1}, X_{\Sigma_{i-1}+2}, \ldots)$ alone, where Σ_{i-1} is a stopping time of $\{X_k\}_{k\geq 0}$. By the strong Markov property, conditioned on $X_{\Sigma_{i-1}} = S_{i-1}$ this continuation is independent of the past $(X_0, \ldots, X_{\Sigma_{i-1}})$ – and so of

$$(S_0, \ldots, S_{i-2}, Y_1, \ldots, Y_{i-1}),$$

functions of that past – with a law depending only on S_{i-1}, by time-homogeneity of the transition kernel. □

This is what makes $\{S_i\}_{i\geq 0}$ – not $\{X_k\}_{k\geq 0}$ itself – the Markov chain that matters for the moment analysis: the matrix quantities below are indexed by $\mathcal{X}$ and evolve one step per phrase rather than per symbol. The construction extends verbatim to a k-th order Markov chain or a hidden finite-state source, taking S_i to be the last k symbols or the hidden state; we do not pursue this here.

5.2 Matrix-valued moment generating functions

For states $s, s' \in \mathcal{X}$, define the matrix-valued moment generating function

$$[\boldsymbol{\mu}_j(t)]_{ss'} := \mathbb{E}\{[\ell(Y_i)]^j e^{-tL(Y_i)} \mathbf{1}[S_i = s'] \mid S_{i-1} = s\}, \qquad j = 0, 1, 2, \ldots, \quad t \geq 0, \tag{52}$$

for any phrase index $i \geq 1$ (by Lemma 5.1, this conditional law does not depend on i). These are $\alpha \times \alpha$ nonnegative matrices parameterized by t: $\boldsymbol{\mu}_j(t)$ is the matrix analogue of the scalar quantity $\mu_j(t)$ of Section 2, and the case $j = 0$ corresponds to the state transition matrix at phrase boundaries, $[\boldsymbol{\mu}_0(t)]_{ss'} = \mathbb{E}\{e^{-tL(Y_i)} \mathbf{1}[S_i = s'] \mid S_{i-1} = s\}$. At $t = 0$, $[\boldsymbol{\mu}_0(0)]_{ss'} = \Pr\{S_i = s' \mid S_{i-1} = s\}$, so $\boldsymbol{\mu}_0(0)$ is row-stochastic.

We assume throughout that $\boldsymbol{\mu}_0(0)$ – the phrase-boundary chain's own transition matrix, as distinct from the transition matrix of $\{X_k\}_{k \geq 0}$ itself – is irreducible *and aperiodic* (equivalently, *primitive*); this is a separate condition from the irreducibility of $\{X_k\}_{k \geq 0}$ assumed in Section 5 (a tree can route every path ending in a given state through a first symbol unreachable in one step from another state, so the two irreducibilities do not simply hand each other over), and it holds for every example in this paper. Aperiodicity is needed, not just irreducibility: an irreducible but periodic chain can have several eigenvalues tied for maximum modulus (e.g. the period-2 chain $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ has eigenvalues 1 and -1, both of modulus 1), which would break the spectral-gap argument used in Section 5.5 and Appendix B to isolate the dominant eigenvalue's contribution; primitivity is exactly what rules this out (by the Perron–Frobenius theorem for primitive matrices, the Perron eigenvalue is then *strictly* greater in modulus than every other eigenvalue).

A simple sufficient condition, easily checked without appeal to Perron–Frobenius theory directly: if $\{X_k\}_{k \geq 0}$ has a *fully positive* one-step transition matrix (i.e. $\Pr\{X_{k+1} = x' \mid X_k = x\} > 0$ for every $x, x' \in \mathcal{X}$), then $\boldsymbol{\mu}_0(0)$ has strictly positive entries throughout, and is therefore automatically both irreducible and aperiodic. Indeed, full positivity means any specific finite symbol sequence has positive probability of occurring next, from any current state; since the parsing tree is finite, every leaf corresponds to some such sequence, so every leaf – and hence every ending state – is reachable with positive probability from every starting state in a single phrase. This condition is considerably stronger than needed (it fails, for instance, whenever some transition is structurally forbidden), but it covers most sources encountered in practice and requires no computation beyond inspecting the source's own transition matrix.

Under this assumption, let π denote the unique stationary distribution of $\boldsymbol{\mu}_0(0)$ – a row vector, consistent with its left-multiplying matrices throughout (as in $\pi\boldsymbol{\mu}_0(0)$ just below) – satisfying

$\pi\boldsymbol{\mu}_0(0) = \pi$ and $\sum_s \pi_s = 1$. For a state-dependent single-phrase quantity $G(Y_i)$ (such as the phrase length $L(Y_i)$, the codeword length $\ell(Y_i)$, or their product), we write $\mathbb{E}_\pi\{G(Y_i)\} \triangleq \sum_{s\in\mathcal{X}} \pi_s \mathbb{E}\{G(Y_i) \mid S_{i-1} = s\}$ for its expectation under the stationary regime; for $G = L$, this plays the role of $\bar{L}$ of Section 2 in the Markov case, and is the a.s. limit $\Sigma_n/n \to \mathbb{E}_\pi\{L(Y_i)\}$ by the ergodic theorem for irreducible finite-state Markov chains.

5.3 The first moment formula for the Markov case

Theorem 5.1. *If the initial state S_0 has the stationary distribution π of the phrase-boundary chain, then*

$$\mathbb{E}\{R_n\} = \int_0^\infty \pi \left[\sum_{i=1}^{n} \boldsymbol{\mu}_0(t)^{i-1}\boldsymbol{\mu}_1(t)\boldsymbol{\mu}_0(t)^{n-i}\right]\mathbf{1}\mathrm{d}t, \tag{53}$$

where $\mathbf{1}$ is the all-ones column vector and π is understood to be a row vector.

Proof. As in the proof of Theorem 3.1, $\mathbb{E}\{R_n\} = \int_0^\infty \mathbb{E}\{\Lambda_n e^{-t\Sigma_n}\}\mathrm{d}t$. Expanding $\Lambda_n e^{-t\Sigma_n} = \sum_{i=1}^n \ell(Y_i)e^{-t(L(Y_1)+\cdots+L(Y_n))}$:

$$\mathbb{E}\{\Lambda_n e^{-t\Sigma_n}\} = \sum_{i=1}^{n} \mathbb{E}\Big\{\ell(Y_i)e^{-tL(Y_i)}\prod_{j\neq i} e^{-tL(Y_j)}\Big\}. \tag{54}$$

For a given i, define $h_j \triangleq e^{-tL(Y_j)}$ for $j \neq i$ and $h_i \triangleq \ell(Y_i)e^{-tL(Y_i)}$, so the i-th term of (54) is $\mathbb{E}\{\prod_{j=1}^n h_j\}$. For $j = 1, \ldots, n+1$, define

$$\phi_j(s) \triangleq \mathbb{E}\Big\{\prod_{k=j}^{n} h_k \Big| S_{j-1} = s\Big\}, \qquad s \in \mathcal{X} \tag{55}$$

where it should be understood that the product is empty when $j = n+1$, so $\phi_{n+1} \equiv 1$. This is well-defined as a function of s alone – a conditional expectation given a random variable is, by definition, always some function of that variable – with no appeal to the Markov property yet. What the Markov property buys us is that these functions also satisfy the *backward recursion*

$$\phi_j(s) = \mathbb{E}\{h_j\, \phi_{j+1}(S_j) \mid S_{j-1} = s\}, \qquad j = n, n-1, \ldots, 1, \tag{56}$$

i.e., that ϕ_j can be built from ϕ_{j+1} one phrase at a time, rather than recomputing a fresh conditional expectation over $k = j, \ldots, n$ from scratch at every step. We prove (56) via the following stronger claim, by induction on j decreasing from $n+1$ to 1:

$$\mathbb{E}\Big\{\prod_{k=j}^{n} h_k \;\Big|\; S_0, \ldots, S_{j-1}, Y_1, \ldots, Y_{j-1}\Big\} = \phi_j(S_{j-1}), \tag{57}$$

i.e., conditioning on the *entire* history through phrase $j-1$, rather than on S_{j-1} alone as in (55), changes nothing. The case $j = n+1$ is immediate: both sides equal 1, the product on the left being empty and $\phi_{n+1} \equiv 1$ by definition.

For the inductive step from $j+1$ to j (assuming (57) at $j+1$, prove it at j), condition on $(S_0, \ldots, S_j, Y_1, \ldots, Y_j)$ and use (57) at $j+1$:

$$\mathbb{E}\Big\{\prod_{k=j}^{n} h_k \Big| S_0, \ldots, S_{j-1}, Y_1, \ldots, Y_{j-1}\Big\} = \mathbb{E}\Big\{h_j\, \phi_{j+1}(S_j) \Big| S_0, \ldots, S_{j-1}, Y_1, \ldots, Y_{j-1}\Big\}. \tag{58}$$

By Lemma 5.1 applied to phrase j, $h_j\, \phi_{j+1}(S_j)$ – a function of $(L(Y_j), \ell(Y_j), S_j)$ – is independent of $(S_0, \ldots, S_{j-2}, Y_1, \ldots, Y_{j-1})$ given S_{j-1}, so the right-hand side reduces to $\mathbb{E}\{h_j\, \phi_{j+1}(S_j) \mid S_{j-1}\}$, a function of S_{j-1} alone – giving (57) at j, and, comparing this same computation against definition (55) of ϕ_j, exactly the recursion (56). Only the *one-step* property of Lemma 5.1 is used, once per step of the induction; the recursion is what accumulates it, phrase by phrase, into (57).

Taking $j = 1$ in (57) gives $\mathbb{E}\{\prod_{k=1}^{n} h_k \mid S_0 = s\} = \phi_1(s)$ – automatic from (55) itself at $j = 1$, and a useful consistency check; averaging over $S_0 \sim \pi$ gives $\mathbb{E}\{\prod_{j=1}^{n} h_j\} = \sum_s \pi_s \phi_1(s)$. Unwinding the recursion (56) and recognizing $[\boldsymbol{\mu}_0(t)]_{ss'}$, $[\boldsymbol{\mu}_1(t)]_{ss'}$ from (52) at each step (the matrix at step j is $\boldsymbol{\mu}_1(t)$ if $j = i$ and $\boldsymbol{\mu}_0(t)$ otherwise),

$$\vec{\phi}_1 = [\boldsymbol{\mu}_0(t)]^{i-1} \boldsymbol{\mu}_1(t) [\boldsymbol{\mu}_0(t)]^{n-i}\, \mathbf{1}, \tag{59}$$

where $\vec{\phi}_1$ is the column vector of values $\phi_1(s)$, $s \in \mathcal{X}$. Hence the i-th term of (54) equals $\pi[\boldsymbol{\mu}_0(t)]^{i-1} \boldsymbol{\mu}_1(t) [\boldsymbol{\mu}_0(t)]^{n-i} \mathbf{1}$; summing over $i = 1, \ldots, n$ and integrating over t gives (53). □

5.4 Worked example and validation

We use the binary Markov source of Savari and Gallager [5]: the state is the last bit emitted, and the self-transition probability (probability of emitting the same bit as the last one) is q. This source has two states $\{0, 1\}$, with transition probabilities $\Pr[S_i = s \mid S_{i-1} = s] = q$ and $\Pr[S_i = 1 - s \mid S_{i-1} = s] = 1 - q$.

We use the parsing dictionary $\{00, 01, 1\}$ (the same as in Section 4) and assign fixed-length codewords $\ell \equiv 2$ (a V2F code). The three possible phrases and their contributions to $\boldsymbol{\mu}_0(t)$ are:

- Phrase 1 (one symbol equal to 1): emitted from state 0 with probability $1-q$ (cross-transition) and from state 1 with probability q (self-transition), driving the source to state 1, with $L = 1$.
- Phrase 00 (two 0s): reached via symbol 0 then another 0; probability from state 0 is $q \cdot q = q^2$ and from state 1 is $(1-q) \cdot q = q(1-q)$, driving the source to state 0, with $L = 2$.

- Phrase 01: from state 0 via 0 (prob. q) then 1 (prob. $1-q$), probability $q(1-q)$; from state 1 via 0 (prob. $1-q$) then 1 (prob. $1-q$), probability $(1-q)^2$. Drives source to state 1, $L=2$.

Collecting these contributions into the matrix $\boldsymbol{\mu}_0(t)$ (rows indexed by starting state, columns by ending state):

$$\boldsymbol{\mu}_0(t) = \begin{pmatrix} q^2 e^{-2t} & (1-q)e^{-t} + q(1-q)e^{-2t} \\ q(1-q)e^{-2t} & qe^{-t} + (1-q)^2 e^{-2t} \end{pmatrix}. \tag{60}$$

Since $\ell \equiv 2$, $\boldsymbol{\mu}_1(t) = 2\boldsymbol{\mu}_0(t)$ and (53) simplifies to $\mathbb{E}\{R_n\} = 2n \int_0^\infty \pi \boldsymbol{\mu}_0(t)^n \mathbf{1}\, \mathrm{d}t$.

Example 5.1. *Take $q = 0.99$ (a highly persistent source). The stationary distribution of parsing-point states is found by solving $\pi\boldsymbol{\mu}_0(0) = \pi$, $\pi_0 + \pi_1 = 1$, giving $\pi = (0.332, 0.668)$ – visibly different from the ordinary stationary distribution of the symbol-level chain $(X_k)_{k\geq 0}$ itself, which is $(1/2, 1/2)$ for* every *q by the symmetry of its transition probabilities under $0 \leftrightarrow 1$; this is the numerical instance, promised in Section 5.1, of the general fact that phrase-boundary sampling need not preserve a chain's own stationary law. Evaluating (53) by numerical quadrature at $n = 10$ phrases gives $\mathbb{E}\{R_{10}\} = 1.6503$. A Monte Carlo simulation of 2,000,000 independent realizations of the Markov parsing process gives $\widehat{\mathbb{E}\{R_{10}\}} = 1.6506 \pm 0.0003$, in agreement within one standard error. Evaluating (53) at increasing n (using the $t = u/n$ substitution and full eigendecomposition of $\boldsymbol{\mu}_0(t)$ for numerical stability at large n):*

n	5	10	30	(∞)
$\mathbb{E}\{R_n\}$	1.657	1.650	1.627	$\rho = 1.497$

Table 3: Finite-n convergence of $\mathbb{E}\{R_n\}$ to the ergodic limit ρ, for the Markov worked example of Example 5.1.

The sequence decreases monotonically to the ergodic limit $\rho = \mathbb{E}_\pi\{\ell\}/\mathbb{E}_\pi\{L\} = 2/(\pi_0 \cdot 1.99 + \pi_1 \cdot 1.01) = 2/1.336 = 1.497$. As a cross-check of the formula, write $\Lambda(t)$ for the dominant *(Perron) eigenvalue of $\boldsymbol{\mu}_0(t)$ – guaranteed simple and positive by the irreducibility of $\boldsymbol{\mu}_0(0)$, and strictly greater in modulus than every other eigenvalue by the additional aperiodicity assumed in Section 5.2 – with $v(t)$ its corresponding right eigenvector and $u(t)$ its corresponding left eigenvector, normalized so $u(t)v(t) \equiv 1$. At $t = 0$, since $\boldsymbol{\mu}_0(0)$ is row-stochastic, $\Lambda(0) = 1$ with $v(0) = \mathbf{1}$ (as $\boldsymbol{\mu}_0(0)\mathbf{1} = \mathbf{1}$) and $u(0) = \pi$ (as $\pi\boldsymbol{\mu}_0(0) = \pi$, and indeed $\pi\mathbf{1} = 1$, consistent with the normalization). Differentiating the eigenvalue equation $\boldsymbol{\mu}_0(t)v(t) = \Lambda(t)v(t)$ at $t = 0$ and left-multiplying by $u(0) = \pi$:*

$$\pi\boldsymbol{\mu}_0'(0)v(0) + \pi\boldsymbol{\mu}_0(0)v'(0) = \Lambda'(0)\,\pi v(0) + \Lambda(0)\,\pi v'(0); \tag{61}$$

since $\pi\boldsymbol{\mu}_0(0) = \pi = \Lambda(0)\pi$, the second term on each side is the same, $\pi v'(0)$, and cancels, leaving $\Lambda'(0) = \pi\boldsymbol{\mu}_0'(0)\mathbf{1}$ (using $v(0) = \mathbf{1}$). Since $[\boldsymbol{\mu}_0(t)\mathbf{1}]_s = \sum_{s'}[\boldsymbol{\mu}_0(t)]_{ss'} = \mathbb{E}\{e^{-tL(Y_i)} \mid S_{i-1} = s\}$ (the

indicator $\mathbf{1}[S_i = s']$ *summed over* s' *is just* 1*), differentiating at* $t = 0$ *gives* $\boldsymbol{\mu}_0'(0)\mathbf{1} = -g$ *with* $g_s := \mathbb{E}\{L(Y_i) \mid S_{i-1} = s\}$*, so* $\Lambda'(0) = -\pi g = -\sum_s \pi_s g_s = -\mathbb{E}_\pi\{L\}$ *by the definition of* $\mathbb{E}_\pi\{L\}$ *in Section 5.2 (this same* g *reappears as the reward vector of the Poisson equation in Section 5.5 below). Numerically,* $\Lambda'(0) = -2(q+1)/(2q+1) = -1.336$ *(obtained directly by differentiating the characteristic polynomial of* $\boldsymbol{\mu}_0(t)$ *at* $t = 0$*); indeed,* $-\Lambda'(0) = 1.336 = \mathbb{E}_\pi\{L\}$*, confirming the general formula just derived.*

5.5 Asymptotic bias for Markov sources

The large-n asymptotic analysis of (53) proceeds as in Section 3.5: the substitution $t = u/n$ stabilizes the numerical evaluation, and the resulting bias coefficient $n(\mathbb{E}\{R_n\} - \rho)$ converges to a finite limit as $n \to \infty$. This is governed by the dominant eigenvalue $\Lambda(t)$ of $\boldsymbol{\mu}_0(t)$ and the reason is that, for large n, $[\boldsymbol{\mu}_0(t)]^n$ itself is governed by $[\Lambda(t)]^n$ (times a bounded projector term, standard for a matrix with a strictly dominant eigenvalue) – the matrix analogue of the scalar $[\mu_0(t)]^n$ of the memoryless case, which trivially equals itself raised to the n-th power. So extracting the $O(1/n)$ term from (53) requires a Taylor series expansion of $\Lambda(t)$ around $t = 0$ to the same order that Section 3.5 needed for the scalar quantity $f(t) = -\ln \mu_0(t)$. For Example 5.1, the results are displayed in Table 4.

n	2,000	10,000	50,000	$(n \to \infty)$
$\mathbb{E}\{R_n\}$	1.504	1.499	1.498	1.497
$n(\mathbb{E}\{R_n\} - \rho)$	12.2	12.3	12.4	≈ 12.4

Table 4: Numerical convergence of the bias coefficient $n(\mathbb{E}\{R_n\} - \rho)$ to its limit, for Example 5.1.

A closed-form expression for the bias coefficient requires going one order further than the eigenvalue derivative $\Lambda'(0) = -\mathbb{E}_\pi\{L\}$ already computed (Example 5.1) – exactly as the memoryless case of Section 3.5 needed not just $f'(0)$ but also $f''(0)$. For a *matrix* eigenvalue problem, unlike the scalar case, getting the second-order term $\Lambda''(0)$ requires first finding the first-order correction to $\Lambda(t)$'s corresponding *eigenvector* – a standard fact of matrix perturbation theory (the same mechanism as second-order energy shifts needing first-order wavefunction corrections in perturbation theory more generally) – and this correction is what the Poisson equation below solves for.

Specifically, consider the expansion $\boldsymbol{\mu}_0(t) = P + t\boldsymbol{\mu}_0'(0) + O(t^2)$ (with $P \triangleq \boldsymbol{\mu}_0(0)$), $\Lambda(t)$'s corresponding right eigenvector as $v(t) = \mathbf{1} + t\,\boldsymbol{e} + O(t^2)$ (since $P\mathbf{1} = \mathbf{1}$, this right eigenvector at $t = 0$ is $\mathbf{1}$, and $\boldsymbol{e}$ is its unknown first-order correction), and the eigenvalue itself as $\Lambda(t) = 1 - \bar{L}t + O(t^2)$, where $\bar{L} := \mathbb{E}_\pi\{L\}$. Substituting into the eigenvalue equation $\boldsymbol{\mu}_0(t)v(t) = \Lambda(t)v(t)$ and matching

the coefficients of t on both sides gives

$$(P - I)\boldsymbol{e} = -\bar{L}\mathbf{1} - \boldsymbol{\mu}_0'(0)\mathbf{1}. \tag{62}$$

Recall from Example 5.1 that $[\boldsymbol{\mu}_0(t)\mathbf{1}]_s = \mathbb{E}\{e^{-tL(Y_i)} \mid S_{i-1} = s\}$, so $\boldsymbol{\mu}_0'(0)\mathbf{1} = -g$ with $g_s \triangleq \mathbb{E}\{L(Y_i) \mid S_{i-1} = s\}$ the expected phrase length given starting state s. Substituting into (62),

$$(P - I)\boldsymbol{e} = g - \bar{L}\mathbf{1}, \qquad \text{i.e.} \qquad (I - P)\boldsymbol{e} = \bar{L}\mathbf{1} - g. \tag{63}$$

Since $(I-P)\mathbf{1} = 0$, this equation only determines $\boldsymbol{e}$ up to an additive multiple of $\mathbf{1}$; writing $w \triangleq -\boldsymbol{e}$ and fixing this freedom by the normalization $\pi w = 0$ turns it into exactly the *Poisson equation*

$$(I - P)w = g - \mathbb{E}_\pi\{L\} \cdot \mathbf{1}, \qquad \pi w = 0, \tag{64}$$

(this name and normalization are standard in Markov reward theory [22], where $(I - P)w = h$ with $\pi h = 0$ is the equation for the *relative reward* w associated with a per-state reward whose π-average has already been subtracted off). Here w – the relative-value vector – is precisely the relative-reward vector of Savari and Gallager [5], computed there for the reward "self-information of a phrase" under dictionary-size asymptotics; here the reward is "phrase length L" and the asymptotics are in phrase count.

This Poisson-equation correction is in fact the whole of what is needed: a second, completely analogous Poisson equation for the reward ℓ, combined with a joint eigenvalue perturbation in both the reward variable and t, yields the bias coefficient C in closed form for a general Markov V2V code – not merely the V2F sub-case tabulated above (writing $\Lambda_{\theta t}$ for the resulting mixed second derivative, and $\Lambda''(0)$ for the second derivative of the already-familiar $\Lambda(t)$ of Example 5.1):

Proposition 5.1 (Markov bias coefficient)**.** *For a Markov V2V code whose phrase-boundary chain is irreducible and aperiodic (Section 5.2), with stationary distribution π, phrase-length mean $\bar{L} \triangleq \mathbb{E}_\pi\{L\}$, codeword-length mean $\bar{\ell} \triangleq \mathbb{E}_\pi\{\ell\}$ (so $\rho = \bar{\ell}/\bar{L}$), and $\bar{c} \triangleq \mathbb{E}_\pi\{L\ell\}$,*

$$\lim_{n\to\infty} n\big(\mathbb{E}\{R_n\} - \rho\big) = C = \frac{\bar{\ell}\,\Lambda''(0) + \bar{L}\,\Lambda_{\theta t}}{\bar{L}^3}, \tag{65}$$

where

$$\Lambda''(0) = \pi\,\boldsymbol{\mu}_0''(0)\,\mathbf{1} - 2\,\pi\,\boldsymbol{\mu}_0'(0)\,w, \qquad \Lambda_{\theta t} = -\bar{c} - \pi\,\boldsymbol{\mu}_1(0)\,w + \pi\,\boldsymbol{\mu}_0'(0)\,w_\ell, \tag{66}$$

w solves (64) and w_ℓ solves the analogous equation $(I - P)w_\ell = g_\ell - \bar{\ell}\,\mathbf{1}$, $\pi w_\ell = 0$, with $g_\ell(s) \triangleq \mathbb{E}\{\ell(Y_i) \mid S_{i-1} = s\}$.

This is an extended version of the paper, including the complete derivation of Proposition 5.1 in Appendix B below, in place of the proof sketch given in the version submitted for publication.

Proof. The argument extends the single-variable eigenvalue perturbation above ($\Lambda'(0) = -\bar{L}$, via the Poisson equation for w) to two variables: a second, completely analogous Poisson equation for the reward ℓ produces w_ℓ, and a joint perturbation of the dominant eigenvalue in both the reward variable and t together produces $\Lambda''(0)$ and the new mixed term $\Lambda_{\theta t}$. Appendix B gives the full derivation of this argument; the formula is verified numerically to 10+ significant figures against direct high-precision computation of the exact formula (53) at large n, on two different Markov V2V codes, and collapses exactly to Proposition 3.1 in the memoryless limit. □

Example 5.2 (Closed form for Example 5.1)**.** *Applying Proposition 5.1 to the V2F code of Example 5.1 ($\ell \equiv 2$, so $\boldsymbol{\mu}_1(t) = 2\boldsymbol{\mu}_0(t)$ and $\bar{c} = 2\bar{L}$) gives, after simplification,*

$$C(q) = \frac{q(2q^2 - q + 1)}{2(1-q)(1+q)^3}. \tag{67}$$

At $q = 0.99$, $C(0.99) = 12.3753$, matching the numerical target of the table above (validated independently to 10 significant figures against direct high-precision computation of (53) *at n up to 10^9).*

Example 5.3 (A genuine V2V code under the same Markov source)**.** *Proposition 5.1 applies equally when ℓ is itself random and state-dependent. Take the same Savari–Gallager source and dictionary $\{00, 01, 1\}$, now equipped with the Huffman codeword assignment of Section 4, $(\ell(00), \ell(01), \ell(1)) = (1, 2, 2)$ – a genuine V2V code, since both L and ℓ vary across phrases. Here ℓ depends only on the destination state ($\ell = 1$ when the parser returns to the root via "00", $\ell = 2$ otherwise), which gives $\rho(q) = (3q + 2)/[2(q + 1)]$ and, applying Proposition 5.1,*

$$C(q) = \frac{q(4q^2 + q + 1)}{4(1-q)(1+q)^3}. \tag{68}$$

At $q = 0.99$: $\rho = 1.2487$ and $C(0.99) = 18.5623$, again validated to 10 significant figures against direct high-precision computation of (53)*, and against an independent check using a second codeword assignment for which ℓ does* not *depend only on the destination state.*

Figure 2 plots both closed forms (67) and (68) over the full range $q \in (0, 1)$, rather than at the single value $q = 0.99$ tabulated above. Both bias coefficients diverge as $q \to 1$ – a highly persistent source mixes slowly, and the Poisson-equation solutions w, w_ℓ that drive $\Lambda''(0)$ and $\Lambda_{\theta t}$

grow correspondingly large – and, notably, the two curves cross at exactly $q = 1/3$ (as follows directly from equating (67) and (68)): for weakly persistent sources the V2F code has the larger finite-n bias, while for strongly persistent sources (including the $q = 0.99$ case tabulated in both examples) the genuine V2V code's bias is larger, mirroring the memoryless-case tension already noted in Section 4 between optimising a code for ρ and optimising it for C.

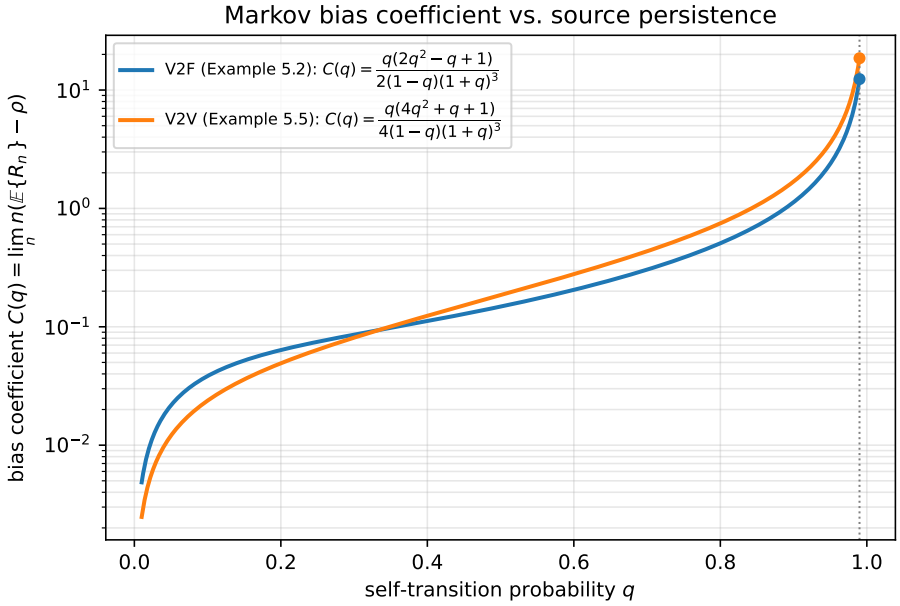

Figure 2: The Markov bias coefficient $C(q)$ of Proposition 5.1, for the V2F code of Example 5.2 and the V2V code of Example 5.3, as the self-transition probability q ranges over $(0, 1)$. Markers indicate the $q = 0.99$ values reported in the text.

6 Conclusion

The exact formula (19) gives, for any fixed V2V code applied to a DMS, exact formulas for all integer moments of the realized compression ratio R_n at every finite n: mean, variance, and skewness are computable by one-dimensional numerical quadrature, from which an Edgeworth approximation to the CDF is obtained. The Laplace method applied to (19) recovers the classical ratio-estimator bias and variance asymptotics algebraically, by a different route than the delta method (Section 3.5), and provides a design guideline (maximize $\mathrm{Cov}\{L, \ell\}$ for fastest convergence) backed by a precise, finite-n formula.

The extension (53) to Markov sources replaces scalar quantities by matrix-valued ones and is validated both against direct simulation and against the correct ergodic limit; the $O(1/n)$ bias

coefficient for this Markov case is likewise obtained in closed form (Proposition 5.1, Appendix B, via a joint eigenvalue perturbation that reduces exactly to Proposition 3.1 in the memoryless limit.

Independently of the moment analysis, the observation that a V2V code is an instance of a finite-state encoder lets us directly apply the generalized Kraft inequality of [2]: via Corollary 2.1, this gives an explicit, m-independent lower bound on the compression ratio in terms of the dictionary parameters M, α, and $\ell_{\max}$, with an $O(1/m)$ correction term that improves on the Ziv–Lempel bound whose analogous constant grows linearly in m. The structural analysis of Section 4 decomposes the bias coefficient C into a $\mathrm{Var}\{L\}$ term and a $\mathrm{Cov}\{L, \ell\}$ term, each of which vanishes identically for one of F2V and V2F; applied to the Khodak code, the same decomposition shows that its improved $O(\bar{L}^{-5/3})$ redundancy and its correspondingly smaller bias coefficient are two faces of the same design principle rather than independent achievements.

Appendix A: Proof of the boundary Laplace lemma

This appendix proves Lemma 3.1, the boundary case of Laplace's method used in Section 3.5, in its general form (arbitrary $k, m \geq 1$).

Proof of Lemma 3.1. Substitute $t = u/n$; then $t^{k-1}\mathrm{d}t = u^{k-1}\mathrm{d}u/n^k$, and

$$\frac{n!}{(n-m)!}\int_0^\infty t^{k-1}h(t)e^{-(n-m)f(t)}\mathrm{d}t = \frac{n!}{(n-m)!\,n^k}\int_0^\infty u^{k-1}h(u/n)e^{-(n-m)f(u/n)}\mathrm{d}u. \tag{A.1}$$

Now,

$$\frac{n!}{(n-m)!} = n(n-1)\cdots(n-m+1) = n^m\Big[1 - \frac{m(m-1)}{2n} + O\left(\frac{1}{n^2}\right)\Big]. \tag{A.2}$$

Now, consider the Taylor series expansion of $-f$ around $t = 0$: $-f(t) = -at + \frac{b}{2}t^2 + O(t^3)$ as $t \to 0$, so with u fixed and $n \to \infty$,

$$-(n-m)f(u/n) = (n-m)\Big[-\frac{au}{n} + \frac{bu^2}{2n^2}\Big] + O\left(\frac{1}{n^2}\right) = -au + \frac{u}{n}\Big(ma + \frac{b}{2}u\Big) + O\left(\frac{1}{n^2}\right), \tag{A.3}$$

whence

$$e^{-(n-m)f(u/n)} = e^{-au}\Big[1 + \frac{u}{n}\Big(ma + \frac{b}{2}u\Big) + O\left(\frac{1}{n^2}\right)\Big]. \tag{A.4}$$

Expanding h as a Taylor series, we have $h(u/n) = h_0 + \dfrac{h_1 u}{n} + O(1/n^2)$, and multiplying the two expansions, we obtain

$$u^{k-1}h(u/n)e^{-(n-m)f(u/n)} = e^{-au}\Big[u^{k-1}h_0 + \frac{1}{n}\big(mah_0u^k + \tfrac{b}{2}h_0u^{k+1} + h_1u^k\big)\Big] + O(n^{-2}). \tag{A.5}$$

Integrating term by term using the identity $\int_0^\infty u^{k-1+j}e^{-au}\mathrm{d}u = (k-1+j)!/a^{k+j}$,

$$\int_0^\infty u^{k-1}h(u/n)e^{-(n-m)f(u/n)}\mathrm{d}u = \frac{(k-1)!h_0}{a^k} + \frac{1}{n}\left(\frac{mh_0k!}{a^k} + \frac{bh_0(k+1)!}{2a^{k+2}} + \frac{h_1k!}{a^{k+1}}\right) + O\left(\frac{1}{n^2}\right). \tag{A.6}$$

Finally, multiplying by $\frac{n!}{(n-m)!n^k} = n^{m-k}\big(1 - \frac{m(m-1)}{2n} + O(1/n^2)\big)$ and collecting the two leading powers of n, the $O(1/n)$ coefficient becomes

$$\frac{mh_0k!}{a^k} + \frac{bh_0(k+1)!}{2a^{k+2}} + \frac{h_1k!}{a^{k+1}} - \frac{m(m-1)}{2}\cdot\frac{(k-1)!h_0}{a^k} = \frac{(k-1)!}{2a^{k+2}}\Big(a^2h_0m(2k-m+1) + 2ah_1k + bh_0k(k+1)\Big), \tag{A.7}$$

using $2mk - m(m-1) = m(2k-m+1)$ to combine the h_0-terms. This is exactly (33). □

Appendix B: Closed-form Markov bias coefficient

This appendix derives the closed-form expression for $n(\mathbb{E}\{R_n\} - \rho)$ used in Section 5, for a general Markov V2V code whose phrase-boundary chain is irreducible and aperiodic (both L and ℓ state-dependent random variables, not merely the V2F sub-case of Example 5.1). The argument rests on a single general fact about how the Perron eigenvalue of a matrix responds to two parameters at once (Lemma 6.1 below); once that lemma is established, the derivation is a short substitution, not a lengthy one, and nothing is omitted.

Setup

Define the joint matrix-valued moment generating function

$$[\Psi(\theta,t)]_{ss'} := \mathbb{E}\{e^{\theta\ell(Y_i)-tL(Y_i)}\mathbf{1}[S_i = s'] \mid S_{i-1} = s\}, \qquad s, s' \in \mathcal{X}, \tag{B.1}$$

so that $\Psi(0,t) = \boldsymbol{\mu}_0(t)$ and $\partial_\theta\Psi(\theta,t)|_{\theta=0} = \boldsymbol{\mu}_1(t)$. Let $\Lambda(\theta,t)$ denote $\Psi(\theta,t)$'s dominant (Perron) eigenvalue near the origin, so $\Lambda(0,t) = \Lambda(t)$ recovers the scalar eigenvalue of Section 5.5.

A general two-parameter perturbation lemma

The remaining derivation is an instance of a single, general fact about how the Perron eigenvalue of a matrix responds to two parameters simultaneously – stated and proved once here, in the abstract, rather than worked out from scratch in the specific notation of this problem.

Lemma 6.1 (Two-parameter Perron eigenvalue perturbation)**.** *Let $A(\theta,t)$ be a matrix-valued function, jointly analytic near the origin, with $A(0,0) = P$ primitive stochastic with Perron eigenvalue*

1 *and left/right eigenvectors* $\pi, \mathbf{1}$ *(normalized* $\pi\mathbf{1} = 1$*). Let* $\Lambda(\theta, t)$*,* $v(\theta, t)$ *denote the Perron eigenvalue and right eigenvector of* $A(\theta, t)$ *near the origin, normalized so* $\pi\, v(\theta, t) \equiv 1$ *(so* $v(0, 0) = \mathbf{1}$*,* $\Lambda(0, 0) = 1$*). Write* $A_x := \partial_x A(0, 0)$*,* $A_{xy} := \partial_x \partial_y A(0, 0)$ *for* $x, y \in \{\theta, t\}$*. Then*

$$\Lambda_x(0, 0) = \pi A_x \mathbf{1}, \tag{B.2}$$

and, writing v_x *for the (unique, once normalized by* $\pi v_x = 0$*) solution of the Poisson equation*

$$(P - I)v_x = \Lambda_x(0, 0)\mathbf{1} - A_x\mathbf{1}, \qquad \pi v_x = 0, \tag{B.3}$$

the second-order coefficients are

$$\Lambda_{xy}(0, 0) = \pi A_{xy}\mathbf{1} + \pi A_x v_y + \pi A_y v_x. \tag{B.4}$$

Proof. Differentiate the eigenvalue equation $A(\theta, t)v(\theta, t) = \Lambda(\theta, t)v(\theta, t)$ once in direction x and evaluate at the origin: $A_x\mathbf{1} + Pv_x = \Lambda_x(0, 0)\mathbf{1} + v_x$, i.e. $(P - I)v_x = \Lambda_x(0, 0)\mathbf{1} - A_x\mathbf{1}$, which is (B.3); left-multiplying instead by π (using $\pi P = \pi$, $\pi\mathbf{1} = 1$, and $\pi v_x = 0$, the last from differentiating the normalization $\pi v(\theta, t) \equiv 1$ and choosing this admissible value of the additive-$\mathbf{1}$ freedom in v_x) gives $\pi A_x\mathbf{1} = \Lambda_x(0, 0)$, which is (B.2).

Differentiating once more, in direction y, and evaluating at the origin:

$$A_{xy}\mathbf{1} + A_x v_y + A_y v_x + Pv_{xy} = \Lambda_{xy}(0, 0)\mathbf{1} + \Lambda_x(0, 0)v_y + \Lambda_y(0, 0)v_x + v_{xy}. \tag{B.5}$$

Left-multiplying by π: the Pv_{xy} and v_{xy} terms both become πv_{xy} (using $\pi P = \pi$) and cancel; the $\Lambda_x(0, 0)v_y$ and $\Lambda_y(0, 0)v_x$ terms vanish (using $\pi v_x = \pi v_y = 0$); what remains is exactly (B.4). □

Application. Take $A = \Psi$. Differentiating (B.1) gives $A_t\mathbf{1} = -g$, $A_\theta\mathbf{1} = g_\ell$ (where $g_\ell(s) := \mathbb{E}\{\ell(Y_i) \mid S_{i-1} = s\}$), $A_{tt} = \boldsymbol{\mu}_0''(0)$, $A_\theta = \boldsymbol{\mu}_1(0)$, $A_t = \boldsymbol{\mu}_0'(0)$, and $A_{\theta t}\mathbf{1} = -g_{L\ell}$ where $g_{L\ell}(s) := \mathbb{E}\{L(Y_i)\ell(Y_i) \mid S_{i-1} = s\}$ (differentiating the exponent $\theta\ell(Y_i) - tL(Y_i)$ once in each variable brings down a factor $-L(Y_i)\ell(Y_i)$). By (B.2), $\Lambda_t(0, 0) = -\pi g = -\bar{L}$ and $\Lambda_\theta(0, 0) = \pi g_\ell = \bar{\ell}$, recovering the already-known values. The two Poisson equations (B.3), for $x = t$ and $x = \theta$ respectively, are

$$(I - P)w = g - \bar{L}\mathbf{1}, \qquad \pi w = 0 \qquad (w := -v_t), \tag{B.6}$$

$$(I - P)w_\ell = g_\ell - \bar{\ell}\,\mathbf{1}, \qquad \pi w_\ell = 0 \qquad (w_\ell := v_\theta), \tag{B.7}$$

matching the Poisson equation of Section 5.5, with the sign flip for w tracing to $A_t\mathbf{1} = -g$ carrying an extra minus sign that $A_\theta\mathbf{1} = g_\ell$ does not. Substituting into (B.4) with $(x, y) = (t, t)$ and $(x, y) = (\theta, t)$:

$$\Lambda''(0) = \pi\, \boldsymbol{\mu}_0''(0)\, \mathbf{1} - 2\,\pi\, \boldsymbol{\mu}_0'(0)\, w, \qquad \Lambda_{\theta t} = -\bar{c} - \pi\, \boldsymbol{\mu}_1(0)\, w + \pi\, \boldsymbol{\mu}_0'(0)\, w_\ell, \tag{B.8}$$

where $\bar{c} := \mathbb{E}_\pi\{L\ell\} = \pi g_{L\ell}$ – exactly the formulas quoted in Proposition 5.1, obtained here by pure substitution into Lemma 6.1 rather than a fresh derivation. The resulting formula was additionally verified numerically to 10+ significant figures against direct high-precision computation of the exact formula (53) at large n, on two different Markov V2V codes built on the source of Example 5.1 (one with ℓ depending only on the destination state, one without that special structure), and collapses exactly to Proposition 3.1 in the memoryless limit.

References

[1] A. W. van der Vaart, *Asymptotic Statistics*, Cambridge University Press, 1998 (Theorem 2.3, the continuous mapping theorem; Theorem 3.1, the delta method).

[2] N. Merhav, "Generalized Forms of the Kraft Inequality for Finite-State Encoders," *Entropy*, vol. 28, no. 3, article 278, 2026.

[3] M. Drmota and W. Szpankowski, "Redundancy of Lossless Data Compression for Known Sources by Analytic Methods," *Foundations and Trends in Communications and Information Theory*, vol. 13, no. 4, pp. 277–417, 2017.

[4] B. P. Tunstall, "Synthesis of noiseless compression codes," Ph.D. dissertation, Georgia Institute of Technology, 1967.

[5] S. A. Savari and R. G. Gallager, "Generalized Tunstall Codes for Sources with Memory," *IEEE Trans. Inf. Theory*, vol. 43, no. 2, pp. 658–668, 1997.

[6] M. Drmota, Y. Reznik, and W. Szpankowski, "Tunstall Code, Khodak Variations, and Random Walks," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2799–2813, 2010.

[7] S. A. Savari, "Variable-to-Fixed Length Codes and Plurally Parsable Dictionaries," *Proc. DCC*, 1999.

[8] S. A. Savari, "Renewal Theory and Source Coding," *Proc. IEEE*, vol. 88, no. 11, pp. 1692–1702, 2000.

[9] O. Shayevitz, E. Meron, M. Feder, and R. Zamir, "Delay and Redundancy in Lossless Source Coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2924–2937, 2014.

[10] V. Strassen, "Asymptotische Abschätzungen in Shannons Informationstheorie," *Trans. Third Prague Conf. Inf. Theory*, pp. 689–723, Prague, 1962.

[11] T. A. Courtade and S. Verdú, "Cumulant Generating Function of Codeword Lengths in Optimal Lossless Compression," *Proc. ISIT*, pp. 2494–2498, 2014.

[12] I. Kontoyiannis, "Second-Order Noiseless Source Coding Theorems," *IEEE Trans. Inf. Theory*, vol. 43, no. 4, pp. 1339–1341, 1997.

[13] I. Kontoyiannis and S. Verdú, "Optimal Lossless Data Compression: Non-Asymptotics and Asymptotics," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 777–795, 2014.

[14] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530–536, 1978.

[15] Y. Bugeaud, M. Drmota, and W. Szpankowski, "On the construction of (explicit) Khodak's code and its analysis," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 5073–5086, 2008.

[16] T. P. Speed, "Cumulants and Partition Lattices," *Austral. J. Statist.*, vol. 25, pp. 378–388, 1983.

[17] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 2, 2nd ed. Wiley, 1971.

[18] N. G. de Bruijn, *Asymptotic Methods in Analysis*, Dover Publications, Inc., New York, 1981 (see Sec. 4.3 for the boundary case of Laplace's method).

[19] N. Merhav and N. Weinberger, "A Toolbox for Refined Information-Theoretic Analyses with Applications," *Foundations and Trends in Communications and Information Theory*, vol. 22, no. 1, pp. 1–184, 2025 (see p. 48).

[20] S. A. Savari, "Variable-to-Fixed Length Codes and the Conservation of Entropy," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1612–1620, 1999.

[21] S. A. Savari and W. Szpankowski, "On the analysis of variable-to-variable length codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Lausanne, 2002, p. 176.

[22] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, 2nd ed., Cambridge University Press, 2009.