SENECA: SMALL-SAMPLE DISCRETE ENTROPY ESTIMATION VIA SELF-CONSISTENT MISSING MASS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete entropy estimation is a classic information theory problem, wherein the average information content of a discrete random variable is estimated from samples alone. Naive approaches, such as the plugin method, fail to account for the probability mass associated with members of the random variable's support that are unobserved in a given sample, known as the "missing mass." The resulting systemic underestimation is particularly problematic when data is time-consuming or costly to gather. We propose SENECA, an entropy estimation scheme based on a novel "self-consistent" missing mass calculation. Extensive numerical experiments indicate that our approach outperforms many state-of-the-art alternatives overall in the small-sample setting. We then apply SENECA to two practical use cases, namely biodiversity estimation and the detection of incorrect large language model responses, where our method is competitive with domain-specific approaches. Our work advances SENECA as an effective drop-in replacement for small-sample entropy estimation, with broad utility across several domains.

1 INTRODUCTION

How surprising is a random variable? Can the expected amount of surprise of a random variable, known as information entropy, be reliably estimated from samples alone? Such are fundamental questions in information theory and uncertainty quantification, with consequences in machine learning and other data-driven disciplines. A classic application is population ecology, where the entropy of species prevalence has served as an index of biodiversity for several decades (Sherwin & Prat i Fornells, 2019). Entropy estimators have also been employed to calculate information gain for decision trees (Nowozin, 2012) and mutual information for gene association network formulation (Hausser & Strimmer, 2009). Extensive applications appear in natural language processing (NLP), as well, where samples may correspond to words of a vocabulary (Grignetti, 1964; Arora et al., 2022), meaning-clusters produced by a word sense induction system (Li et al., 2014), or class labels of text produced by a generative model (Kuhn et al., 2023). An example of the latter is *semantic entropy*, a popular method for predictive uncertainty quantification in large language models (LLMs), often used for hallucination detection in question-answering (QA) tasks (Farquhar et al., 2024).

As we discuss in Section 2, discrete entropy estimators are biased (Paninski, 2003), and many have high error when sample size is small. In practical scenarios, samples may be difficult to obtain, so methods well-suited for small sample sizes are crucial (Chao & Jost, 2012). In population ecology, for instance, sample collection can be labor-intensive, involving manual collection and species identification (Janzen, 1973a). In natural language processing applications, semantic entropy and other variability-based LLM uncertainty quantification (UQ) methods typically rely on only 10 or so samples (McCabe et al., 2026), for reasons that include the computational complexity of repeated text generation and response classification (Kuhn et al., 2023; Farquhar et al., 2024). In this small-sample setting, McCabe et al. (2026) demonstrated that the plugin method employed by Farquhar et al. (2024) in calculating "black-box" semantic entropy underestimates "true" semantic entropy.

Our contributions. To address the above limitations, this work advances the following:

1. We propose **SENECA** (small-sample discrete entropy estimation via self-consistent missing mass), an entropy estimator based on shrinkage and fixed-point iteration (Section 3).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2. We assess SENECA for entropy estimation, demonstrating **improved accuracy over existing estimators in the small-sample setting** in numerical experiments on 72 unique distributions (Section 4). Our method performs especially favorably in the under-sampled regime, where observation of all members of the support is impossible.
3. We demonstrate broad interdisciplinary applications by extending our evaluation to **biodiversity estimation** and **LLM uncertainty quantification** (Section 5). Toward the former, **SENECA consistently outranks other methods** on 58 real species-prevalence datasets. Regarding the latter, SENECA attains the **second-highest aggregated AUROC** among several competitive baselines, **outranking other entropy-based methods**.

2 BACKGROUND

Entropy estimation. Let $A^n = \{\{a_1, a_2, \ldots, a_n\}\}$ be a multiset of n independent and identically distributed (IID) samples drawn with replacement from a population of items. A classification function $h : \mathcal{A} \to \mathcal{X}$ maps items to labels, inducing the multiset of label occurrences $X^n = \{\{x_1, x_2, \ldots, x_n\}\} = \{\{h(a)|a \in A^n\}\}$. Without knowing the multinomial label-prevalence distribution $P = \langle p_u | u \in \mathcal{X}\rangle$ or support $|\mathcal{X}|$, our goal is to estimate the entropy (Shannon, 1948):

$$\mathbb{H}(P) = -\sum_{u\in\mathcal{X}} p_u \log p_u.$$

Plugin entropy. Let the label-counting function $N_{u,h}(A^n) = \sum_{a\in A^n} \mathbb{1}_{h(a)=u}$ denote the number of times items belonging to the label u appear in A^n (for brevity, we will use $N_{u,h}(A^n)$ and $N_u(X^n)$ interchangeably). The so-called "plugin" entropy estimator relies on maximum-likelihood prevalence estimates $\hat{p}_u^{ML} = \frac{N_u(X^n)}{n}$ (i.e., the empirical frequencies of observed labels):

$$\hat{\mathbb{H}}^{plugin}(X^n) = -\sum_{u\in X^n_{\neq}} \hat{p}_u^{ML} \log \hat{p}_u^{ML}, \tag{1}$$

where $X^n_{\neq}$ denotes the set of unique items of X^n. The plugin method is consistent (Antos & Kontoyiannis, 2001), but negatively biased (Basharin, 1959), failing to account for unobserved labels (Figure 1A). In the *under-sampled regime*, where $|\mathcal{X}| > n$, unobserved labels are guaranteed.

Missing mass estimation. The total probability mass associated with unobserved labels, known as the *missing mass*, corresponds with the probability that a subsequent observation a_{n+1} will belong to a new, unobserved label (i.e., $h(a_{n+1}) \in U$, where $U = \mathcal{X} \setminus X^n_{\neq}$):

$$M(X^n) = \sum_{u\in\mathcal{X}} p_u \mathbb{1}_{N_u(n)=0}.$$

In our notation, $M(X^n)$ is the *particular* missing mass associated with X^n, and M is a random variable. The classic Good-Turing missing mass estimator is given by $\hat{M}^{GT}(X^n) = \frac{\Phi_1(X^n)}{n}$, where $\Phi_i(X^n) = \sum_{u\in\mathcal{X}} \mathbb{1}_{N_u(X^n)=i}$ (Good, 1953). The multiset $\{\{\Phi_i | i \in \mathbb{Z}_{[0,n]}\}\}$ denotes the frequency of frequencies (FoFs) or "fingerprint" of the sample (Valiant & Valiant, 2017; Wu & Yang, 2019). $\hat{M}^{GT}$ may be viewed as a leave-one-out method (McAllester & Schapire, 2001) that is unbiased with respect to $M(X^{n-1})$ (Juang & Lo, 2002), but not $M(X^n)$ (McAllester & Schapire, 2000; Antos & Kontoyiannis, 2001). Indeed, $\hat{M}^{GT}$ has bias at most $\frac{1}{n}$ (McAllester & Schapire, 2000), but its worst-case risk mean-squared error is at least $\frac{0.608}{n} + o(\frac{1}{n})$ (Rajaraman et al., 2017).

Coverage-adjustment. The complement of the missing mass is known as the sample coverage (Roswell et al., 2021). The coverage-adjusted entropy estimator of Chao & Shen (2003) scales observed labels prevalence by Good-Turing-estimated sample coverage (i.e., $\hat{p}_u^{CS} = (1-\hat{M}^{GT})\hat{p}_u^{ML}$) and applies a Horvitz–Thompson scheme (Horvitz & Thompson, 1952) to estimate the total entropy:

$$\hat{\mathbb{H}}^{CS} = -\sum_{u\in X^n_{\neq}} \frac{\hat{p}_u^{CS} \log \hat{p}_u^{CS}}{1-(1-\hat{p}_u^{CS})^n}. \tag{2}$$

By way of $\hat{M}^{GT}$, $\hat{\mathbb{H}}^{CS}$ uses only one FoF: the number of "singletons" Φ_1. While $\hat{\mathbb{H}}^{CS}$ can excel for small sample sizes (Pinchas et al., 2024), its dependence on the singleton rate Φ_1 can be problematic; if $\Phi_1(X^n) = n$, for example, the summand of Equation 2 is $\frac{0\log 0}{0}$, and $\hat{p}_u^{CS} = \hat{p}_u^{ML}$ when

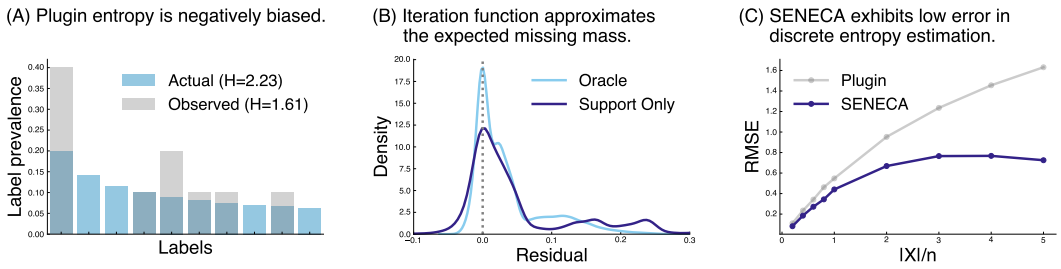

Figure 1: **Visual overview. (A) The plugin method can underestimate entropy when elements of the support go unobserved.** Shown: a Zipf distribution with shape parameter 0.5 and support size of 10 (cyan). A sample of size 10 is drawn, and the empirical frequencies of observed labels are depicted (translucent grey overlay). Labels without grey overlay are unobserved; their cumulative probability mass is the "missing mass." **(B) When the support size and missing mass are known, Eq. 4 approximates the *expected* missing mass.** Shown: kernel density plots of residuals of two methods of estimating $\mathbb{E}_{X^n\sim P}[M]$ - $\mu(M(X^n), |U|, X^n)$ (Oracle) and the fixed-point missing mass calculated by SENECA with *known* support size (Support Only) - under the simulation settings used in our numerical experiments (i.e., $n = 10$, 72 unique distributions, each over 1000 trials; see Figure 2). The grey dotted line highlights a residual of zero. **(C) SENECA uses fixed-point iteration and a support size *estimate* to achieve low error.** Shown: simulation results for the Zipf-0.5 distribution only (for symmetry with panel A) over 9 support sizes, each over 1000 trials.

$\Phi_1(X^n) = 0$. Subsequent work by Chao et al. (2013) employs two FoFs: Φ_1 and Φ_2. Such methods successfully reduce the bias associated with entropy estimation, but no unbiased estimator exists in general (Paninski, 2003), and low-bias methods may exhibit high variance (Hastie et al., 2009).

3 Entropy estimation via "self-consistent" missing mass

Herein, we motivate and introduce SENECA. We begin by decomposing the expected missing mass of a multinomial label-prevalence distribution P over support $\mathcal{X}$ for samples of size n into "observed" and "unobserved" components, for a realized X^n:

$$\mathbb{E}_{X^n\sim P}[M] = \sum_{u\in\mathcal{X}} p_u\Big(1-p_u\Big)^n = \underbrace{\Big(\sum_{u\in X^n_{\neq}} p_u\Big(1-p_u\Big)^n\Big)}_{\text{"observed" component}} + \underbrace{\Big(\sum_{u\in U} p_u\Big(1-p_u\Big)^n\Big)}_{\text{"unobserved" component}}. \tag{3}$$

We are left to estimate the label-prevalences within the observed and unobserved components of Eq. 3. For the former, we notice that $\hat{p}^{ML}_u$ is conditionally positively biased on non-zero missing mass. For this reason, we employ *isotropic shrinkage* of the maximum likelihood prevalence estimator, scaling empirical frequencies of all observed labels by a single scalar (i.e., $\hat{p}^{IS}_u = \alpha\hat{p}^{ML}_u \forall u \in X^n_{\neq}$ for some $\alpha \in (0,1)$). If, hypothetically, $M(X^n)$ is known, then $\alpha = 1 - M(X^n)$ preserves the missing mass, since $M(X^n) = 1-\sum_{u\in X^n_{\neq}}\hat{p}^{IS}_u = 1-\alpha\sum_{u\in X^n_{\neq}}\hat{p}^{ML}_u = 1-\alpha$.[1] Under *natural estimation*, all labels appearing the same number of times in X^n are assigned the same prevalence estimate (Orlitsky & Suresh, 2015), an assumption made by the overwhelming majority of distribution estimators (Balasundaram & Thangaraj, 2025). Consequently, unobserved labels are assigned prevalence $\frac{M(X^n)}{|U|}$ if $|U|$ is known and at least one. With the above in mind, we construct the following function of $m \in \mathbb{R}_{[0,1]}$ and $\upsilon \in \mathbb{R}_{\geq 0}$ and sample labels X^n, based on Eq. 3:

$$\mu(m,\upsilon,X^n) = \underbrace{(1-m)\sum_{u\in X^n_{\neq}}\hat{p}^{ML}_u\Big(1-(1-m)\hat{p}^{ML}_u\Big)^n}_{\text{"observed" component}} + \underbrace{m\Big(1-\frac{m}{\max(\upsilon,1)}\Big)^n}_{\text{"unobserved" component}}, \tag{4}$$

[1] Exchanging $\hat{M}^{GT}$ for $M(X^n)$ leads to the observed species prevalence estimates of Chao & Shen (2003).

where the $\max(\upsilon, 1)$ ensures that the unobserved component is defined. The sub-Gaussianity of missing mass (Skorski, 2023; Li & Tian, 2024) implies large deviations are rare: realizations of M are unlikely to substantially stray from expectation. It stands to reason that $\mu(M(X^n), |U|, X^n) \approx \mathbb{E}_{X^n \sim P}[M]$, which we observe in practice; the cyan curve in Figure 1B ("Oracle") depicts concentration near zero of residuals between $\mathbb{E}_{X^n \sim P}[M]$ and $\mu(M(X^n), |U|, X^n)$ over many settings.

Of course, the true values of $M(X^n)$ and $|U|$ are generally unknown. Instead, given an estimate of the number of unobserved labels υ, we propose a *self-consistent* missing mass estimate:

$$\hat{M}^{SC}(X^n) \triangleq m^* : \mu(m^*, \upsilon, X^n) = m^*. \tag{5}$$

The fixed point solution of Eq. 5 is guaranteed to exist (Lemma 1, Appendix A). The indigo curve in Figure 1B ("Support Only") illustrates that using the *true* number of unobserved labels and performing fixed-point iteration on Eq. 5 induces similarly small, albeit slightly larger, residuals as the oracle setting. In the absence of the true number unobserved labels, we propose the following recipe: employ an *estimate* of the support size $|\widehat{\mathcal{X}}|$, set $\upsilon = \max(|\widehat{\mathcal{X}}| - |X^n_{\neq}|, 1)$ and adjust empirical label-prevalences by $(1 - \hat{M}^{SC}(X^n))$ in a Chao-Shen-like entropy estimator:

$$\text{SENECA}(X^n) = -\left(1 - \hat{M}^{SC}(X^n)\right) \sum_{u \in X^n_{\neq}} \frac{\hat{p}^{ML}_u \log\left((1 - \hat{M}^{SC}(X^n))\hat{p}^{ML}_u\right)}{1 - \left(1 - \left((1 - \hat{M}^{SC}(X^n))\hat{p}^{ML}_u\right)\right)^n}. \tag{6}$$

Algorithm 1 (Appendix B.1) summarizes our approach. Except where stated otherwise, our implementation calculates $|\widehat{\mathcal{X}}|$ via a regularized weighted Chebyshev estimator suitable for small-sample settings (Chien & Milenkovic, 2019; Rana et al., 2022) We provide further details in Appendix B.2, including an ablation confirming that our headline results are not particularly sensitive to our choice of support size estimator. Intuitively, minor differences in the estimate of support size have little effect on SENECA's accuracy, since the "unobserved" component of Eq. 4 is small. Although our priority is entropy estimation, we also evaluate the proposed missing mass estimator in its own right (Appendix F), finding that it outperforms several alternatives in our setting of interest.

SENECA aims to address the small-sample setting, which is the focus of the empirical results to follow. For completeness, we also provide a consistency result characterizing SENECA's large-sample behavior, with supporting lemmas postponed to Appendix A:

Theorem 1. *Let $X^n \sim P$ be a multiset of n IID samples drawn from a multinomial distribution P over finite support $\mathcal{X}$, with true entropy $H(P)$. Then, as $n \to \infty$, $\text{SENECA}(X^n) \xrightarrow{p} H(P)$.*

Proof. Lemmas 2 and 3 establish that $\hat{M}^{SC}_n(X^n) \in [0, 1)$ and $\hat{M}^{SC} \to 0$ as $n \to \infty$, respecitvely. By Lemma 4, any Horvitz-Thompson-style entropy estimator relying on a missing mass estimator with the two aforementioned properties is weakly consistent for distributions with finite entropy variance. Since distributions over finite support have finite entropy variance, the result follows. □

4 Assessing entropy estimators via simulation

Our first batch of experiments assesses SENECA as a small-sample entropy estimator, alongside an array of prior methods found in information theory, ecological statistics, and physics literature.

4.1 Methods

Numerical experiments using 72 unique distributions. We validate the performance of our entropy estimator across 9 support sizes for each of 8 families of prevalence distribution on a simulated item-capture task. For each distribution, we construct a synthetic population of 100 items divided into a variable number of mutually exclusive labels (i.e., $|\mathcal{X}| = 2, 4, 6, 8, 10, 20, 30, 40, 50$ in Figure 2 and Table 1, $|\mathcal{X}| = 4, 8, 12, 16, 20, 40, 60, 80, 100$ in Table 2). Overall label-prevalences (i.e., $P = \{\{p_u | u \in \mathcal{X}\}\}$ follow distributions found in prior works. In particular, the first six are used repeatedly in related literature (Orlitsky et al., 2016; Hao & Li, 2020; Lee & Böhme, 2025) and the

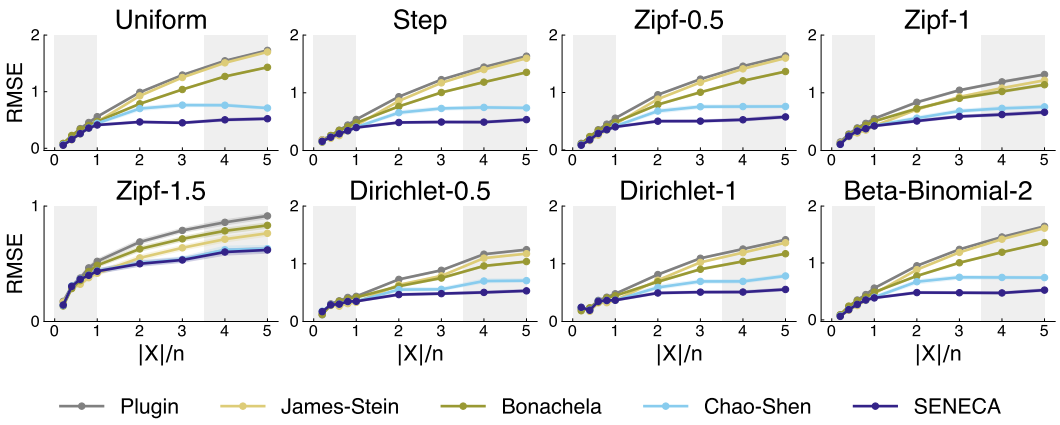

Figure 2: **Entropy estimation in small-sample item capture.** For each of 8 families of label-prevalence distribution, each item's capture probability is determined uniformly over its label's prevalence probability mass. The number of labels $|\mathcal{X}|$ varies from 2 to 50 across trials. For each trial, $n = 10$ items are captured, and the entropy of the label-prevalence distribution is estimated. Results are averaged over 1000 trials per label count. Colored regions about each line plot depict 95% bias-corrected and accelerated confidence intervals, based on 1000 bootstrap replications. In each plot, the leftmost grey region corresponds to the well-sampled regime (i.e., $|\mathcal{X}| = \frac{n}{5}, \frac{2n}{5}, \frac{3n}{5}, \frac{4n}{5}, n$), where items to the right correspond to the under-sampled regime (i.e., $|\mathcal{X}| = 2n, 3n, 4n, 5n$). The rightmost grey region corresponds to the support-risky regime, where no consistent estimator for support size exists. Full results with all 8 estimators may be found in Figure 6 (Appendix C).

following two are additional inclusions from other works (Orlitsky & Suresh, 2015; Painsky, 2023): a uniform distribution (i.e., $p_u = \frac{1}{|\mathcal{X}|}$); a so-called "step" distribution, where half the labels have $p_u = \frac{1}{2|\mathcal{X}|}$ and half have $p_u = \frac{3}{2|\mathcal{X}|}$; three Zipf distributions (i.e., $p_u \propto u^{-\alpha}$), with shape parameters $\alpha = 0.5, 1,$ and 1.5; two distributions generated from Dirichlet priors (Dirichlet-0.5 and Dirichlet-1); and a beta-binomial distribution whose parameters are both 2 (i.e., $p_u = \binom{|\mathcal{X}|}{u} \frac{B(u+\alpha, |\mathcal{X}|-u+\beta)}{B(\alpha,\beta)}$), where B denotes the beta function. Item-level capture probabilities are uniform within each class.

Estimators and evaluation metrics. We compare against six alternative entropy estimators: the plugin estimator (Eq. 1), the Grassberger estimator (Grassberger, 1988; 2003), the James-Stein estimator (Hausser & Strimmer, 2009), the Bonachela estimator (Bonachela et al., 2008), the Chao-Shen estimator (Eq. 2) (Chao & Shen, 2003), the two-FoF Chao-Wang-Jost estimator (Chao et al., 2013), and the linear programming-based estimator of Valiant & Valiant (2017). The James-Stein and Bonachela estimators are explicitly specialized for small sample sizes, and the Chao-Shen and Chao-Wang-Jost methods are acclaimed for fast convergence to the ground truth (La Torre et al., 2025). Our primary metric for evaluating estimators is the root-mean-squared error (RMSE), which we later decompose into empirical estimates of bias and variance (Appendix C). Because RMSE's sampling distribution can be biased, naive percentile bootstrap intervals may deviate from observed RMSEs. For this reason, we calculate uncertainty about these evaluation metrics via Bias-Corrected and Accelerated (BCa) bootstrap confidence intervals (Efron, 1987). BCa is not necessarily suitable for our aggregation of aggregate RMSEs across regimes, so we perform 1000 bootstrap replications and calculate pivot confidence intervals, instead. In Table 1, we report the larger of the two sides of each interval, making the corresponding confidence level slightly more conservative than 95%.[2]

4.2 Results

SENECA achieves consistently low error across label distributions and sampling regimes. For visual clarity, Figure 2 shows entropy estimation results for a subset of the considered baselines:

[2]Due to the large number of trials, we anticipate these confidence intervals to be quite small (Faber, 1999).

Table 1: **Regime-averaged error for entropy estimation in small-sample item capture.** Using the data from Figure 2, we calculate the root-mean-squared error (RMSE ↓) for each estimator and average results within the well-sampled (Well, i.e., $|\mathcal{X}| \leq n$) and under-sampled (Under, i.e., $|\mathcal{X}| > n$) regimes (Reg.). The estimators are Plugin (ML), Grassberger (G), James-Stein (JS), Bonachela (B), Chao-Shen (CS), Chao-Wang-Jost (CWJ), Valiant-Valiant (VV), and SENECA (highlighted in blue). The distributions are Uniform, Step, Zipf-0.5, Zipf-1, Zipf-1.5, Dirichlet-0.5 (Diri-0.5), Dirichlet-1 (Diri-1), and Beta-Binomial-2 (BB-2). The lowest and second-lowest values in each row, prior to rounding, are bold and underlined, respectively. Bootstrapped confidence intervals (grey text) are slightly more conservative than 95%, as described in Section 4.1. In parentheses, we show the relative difference between SENECA's averaged RMSE and the lowest value in each row.

Distr.	Reg.	ML	G	JS	B	CS	CWJ	VV	SENECA
Uniform	Well	0.33 ±.00	0.25 ±.00	0.26 ±.00	0.30 ±.00	0.26 ±.01	**0.25** ±.01	0.36 ±.01	0.25 ±.01 (0.7%)
	Under	1.39 ±.00	0.66 ±.01	1.35 ±.00	1.13 ±.01	0.73 ±.01	0.56 ±.01	0.62 ±.01	**0.49** ±.01 (0.0%)
Step	Well	0.35 ±.01	0.29 ±.01	0.28 ±.01	0.33 ±.01	0.29 ±.01	0.29 ±.01	0.38 ±.01	**0.28** ±.01 (0.0%)
	Under	1.31 ±.01	0.64 ±.01	1.26 ±.00	1.08 ±.01	0.72 ±.02	0.58 ±.01	0.64 ±.01	**0.50** ±.01 (0.0%)
Zipf-0.5	Well	0.34 ±.01	0.26 ±.01	0.26 ±.01	0.31 ±.01	0.26 ±.01	0.26 ±.01	0.36 ±.01	**0.26** ±.01 (0.0%)
	Under	1.32 ±.01	0.66 ±.01	1.27 ±.00	1.09 ±.01	0.74 ±.02	0.59 ±.01	0.66 ±.01	**0.53** ±.01 (0.0%)
Zipf-1	Well	0.37 ±.01	0.32 ±.01	**0.29** ±.01	0.34 ±.01	0.30 ±.01	0.32 ±.01	0.38 ±.01	0.30 ±.01 (3.1%)
	Under	1.10 ±.01	0.62 ±.01	0.98 ±.01	0.95 ±.01	0.68 ±.01	0.63 ±.01	0.69 ±.01	**0.60** ±.01 (0.0%)
Zipf-1.5	Well	0.37 ±.01	0.35 ±.01	**0.32** ±.01	0.34 ±.01	0.33 ±.01	0.35 ±.01	0.37 ±.01	0.33 ±.01 (4.7%)
	Under	0.81 ±.01	0.56 ±.01	0.67 ±.01	0.74 ±.01	0.57 ±.01	0.60 ±.01	0.66 ±.01	**0.56** ±.01 (0.0%)
Diri-0.5	Well	0.33 ±.01	0.30 ±.01	**0.26** ±.01	0.31 ±.01	0.29 ±.01	0.29 ±.01	0.35 ±.01	0.29 ±.01 (10.8%)
	Under	1.01 ±.01	0.52 ±.01	0.92 ±.01	0.84 ±.01	0.63 ±.02	0.55 ±.01	0.62 ±.01	**0.50** ±.01 (0.0%)
Diri-1	Well	0.34 ±.01	0.31 ±.01	**0.29** ±.01	0.32 ±.01	0.31 ±.01	0.31 ±.01	0.37 ±.01	0.30 ±.01 (6.4%)
	Under	1.15 ±.01	0.58 ±.01	1.07 ±.01	0.95 ±.01	0.69 ±.02	0.59 ±.01	0.65 ±.01	**0.52** ±.01 (0.0%)
BB-2	Well	0.33 ±.01	0.26 ±.00	0.26 ±.01	0.31 ±.01	0.25 ±.01	**0.25** ±.01	0.36 ±.01	0.25 ±.01 (0.3%)
	Under	1.32 ±.00	0.64 ±.01	1.27 ±.00	1.08 ±.01	0.73 ±.01	0.57 ±.01	0.63 ±.01	**0.49** ±.01 (0.0%)

the naive plugin estimator, the small-sample specialist James-Stein and Bonachela estimators, and the classic Chao-Shen estimator. Full results with all considered estimators are shown in Figure 6 (Appendix C). In both figures, SENECA achieves consistently low error in the under-sampled regime. We further examine these results in Table 1, averaging errors across domain sizes within the well-sampled ($|\mathcal{X}| \leq n$) and under-sampled ($|\mathcal{X}| > n$) regimes. The James-Stein estimator, which excels in the well-sampled settings, is less reliable in the under-sampled regime. SENECA performs well in both regimes, attaining the lowest average error in all under-sampled settings and lowest or second-lowest error in all but one well-sampled settings. Practically, the regime is rarely known in advance, underscoring the importance of balanced performance across regimes.

SENECA excels even where support estimation is risky. A result of Wu & Yang (2019) indicates that consistent estimation of $|\mathcal{X}|$ is only possible with at least $\frac{|\mathcal{X}|}{\log |\mathcal{X}|}$ samples. We therefore denote the *support-risky* regime as that for which $|\mathcal{X}| > \gamma$, where γ is the largest integer for which the sample size $n > \frac{\gamma}{\log \gamma}$ (e.g., $\gamma = 35$ for $n = 10$). We illustrate this support-risky regime in the rightmost grey regions of each subplot in Figure 2. Despite reliance on a support size estimate, our method maintains its top-end performance, with low RMSE well into the support-risky regime.

SENECA balances bias and variance to achieve low error. The well-known bias-variance tradeoff (Hastie et al., 2009) is no less applicable to entropy estimation: minimizing error warrants management of both components. An empirical bias-variance decomposition of the results in Figure 6 is provided in Figure 7 (Appendix C). In general, the estimators with the lowest bias in the under-sampled regime tend to exhibit the highest variance (e.g., Chao-Shen), and vice-versa (e.g., Plugin) in the under-sampled regime. On the other hand, SENECA maintains comparably low or moderate bias and variance across settings, resulting in consistently low RMSE overall.

5 APPLICATIONS

We extend our comparison of entropy estimators to biodiversity quantification, where the entropy of species-prevalence distributions is a standard metric (Sherwin & Prat i Fornells, 2019). Here, "items" in our problem statement correspond to organisms, and "labels" are their species. We then conclude our experiments by applying SENECA to incorrect LLM response detection. Variability-based uncertainty quantification techniques operate under the assumption that LLMs may be more uncertain about a query when they express greater semantic variability under repeated sampling.

5.1 METHODS

5.1.1 BIODIVERSITY ESTIMATION

Datasets. Janzen (1973b) meticulously reports insect species frequencies observed in sweep samples across 25 tropical sites under various times of day (i.e., day, night) and conditions (i.e., wet, dry); the dataset is described in further detail in Janzen (1973a). We treat each of the 58 collections of true bugs (order Hemiptera) as a complete population, similar to Chao & Shen (2003). The corresponding species-prevalence distributions vary substantially in population size ($|A| \leq 1407$) and support size ($|\mathcal{X}| \leq 67$). The distributions' coefficients of variation range from 0 to approximately 3.5, indicating that the considered populations consist of both heavy-tailed and light-tailed distributions. We draw samples of various sizes ($n = 10, 20, 30, 40, 50$) and estimate the entropy of the complete species distributions accordingly, employing the following methods.

Estimators and evaluation metrics. We use the same entropy estimators itemized in Section 4.1, comparing the entropy estimated from samples to the entropy of the "complete" population. To select an estimator that performs best overall, we seek reliability across numerous distributions and sample sizes, where RMSE magnitudes vary substantially. Therefore, we treat the RMSE-based ranking of estimators for each collection and sample size as a "ballot" and report the Borda count, which allocates points to candidates (estimators) according to the number of other candidates they outrank in each ballot. Consequently, the estimator with the highest Borda count consistently achieves lower RMSE than others, across the populations and sample sizes considered. The notable source of uncertainty in this aggregated count arises from the underlying species prevalence collections, so we calculate 95% pivot confidence intervals by bootstrapping over collections.

5.2 LLM PREDICTIVE UNCERTAINTY QUANTIFICATION

Models. We select five LLMs, ranging from 3.8 to 32 billion total parameters: the 3.8B-parameter Phi-4-mini-instruct (Abouelenin et al., 2025), Mistral-7B-Instruct-v0.3, the 14B-parameter Phi-4 (Abdin et al., 2024), Granite-3.0-8B-Instruct (Granite Team, 2024), and the 32B-parameter Granite-4.0-H-Small (IBM Research, 2025). The first four are dense models, and the latter is a "hybrid" mixture-of-experts model with 9B active parameters. Additional details are found in Appendix B.3.

Datasets. Our experiments use six permissively licensed datasets concerning biomedical subjects, mathematics word problems, common misconceptions, and general question-answering, namely BioASQ Task 11b (Krithara et al., 2023), SQuAD 2.0 (Rajpurkar et al., 2016), SVAMP (Patel et al., 2021), HotpotQA (Yang et al., 2018), TruthfulQA (Lin et al., 2022), and SimpleQA-Verified (Haas et al., 2025), using the first 1000 exemplars from each dataset. TruthfulQA has fewer than 1000 total items, so we use the full dataset, with the exception of the several prompts for which the reference answer is listed as "I have no comment." Following Farquhar et al. (2024), we make the QA task more challenging by omitting context passages in those datasets that offer them.

Baselines. We consider several black and grey-box methods (i.e., not using and using response log-probabilities, respectively) for LLM UQ. The black-box methods are discrete semantic entropy (DSE, (Farquhar et al., 2024)), U_{EigV} (Lin et al., 2024), kernel language entropy (KLE, (Nikitin et al., 2024)), and SENECA. The grey-box methods are predictive entropy (PE, (Kadavath et al., 2022)), semantic entropy (SE, (Kuhn et al., 2023)), G-NLL (Aichberger et al., 2026), and surprise (which is similar to PE, (Ismayilzada et al., 2025)).

The baselines in this section differ from those used previously, because they are domain-specific. U_{EigV} is a support size estimator, and PE, DSE, SE, KLE, and the SENECA methods are entropy estimators of various forms. DSE is equivalent to the established Plugin method, carried out over semantic equivalence classes. SE is similar to DSE, but class prevalences are based on sequence log-probabilities, rather than empirical frequencies. Baselines are detailed further in Appendix B.6.

Semantic clustering. SE, DSE, and SENECA calculate entropy over semantic equivalence classes, which are obtained via semantic clustering of responses. For all three, we follow the bidirectional entailment classification method of Kuhn et al. (2023) (Appendix B.4). KLE and U_{EigV} do not explicitly perform semantic clustering, but they do rely on natural language inference (NLI)-based semantic graphs. For consistency, we use the same NLI model for all five above methods.

LLM-as-judge. To assess LLM correctness in response to a query, we draw a low-temperature ($\tau = 0.1$) "best guess" sample, as seen in prior works (Farquhar et al., 2024; Nikitin et al., 2024; Nguyen et al., 2025), and prompt an LLM judge to gauge its consistency with a reference answer (Lin et al., 2024); further details are provided in Appendix B.5. In Appendix E, we also consider a binary (i.e., yes/no) LLM judge of correctness, with qualitatively indistinguishable results.

Metrics. We measure success in classifying incorrect responses by the area under the receiver operating characteristic curve (AUROC), as seen in prior works (Kuhn et al., 2023; Lin et al., 2024; Farquhar et al., 2024; Aichberger et al., 2026). Empirical AUROC is a parameter estimate - in this case, that of the probability that an UQ method assigns a higher uncertainty to a randomly-selected incorrect response than a random correct one. As such, we represent the uncertainty of each AUROC point estimate by 95% confidence intervals (CIs), calculated using the DeLong method (DeLong et al., 1988; Sun & Xu, 2014). Because each model-dataset pair represents a distinct setting, we aggregate the AUROC estimates and CIs for methods (i.e., for Figure 4) using DerSimonian-Laird random effects meta-analysis (DerSimonian & Laird, 1986) with Hartung-Knapp-Sidik-Jonkman error adjustment (Hartung & Knapp, 2001; Sidik & Jonkman, 2002; IntHout et al., 2014).

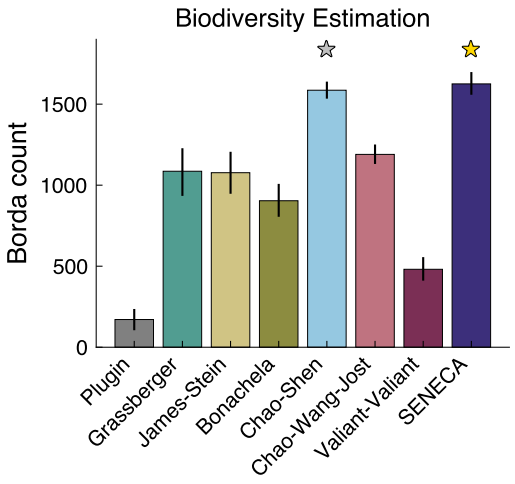

Figure 3: **Summary of entropy methods for biodiversity estimation across** 58 **tropical insect collections.** Treating each collection of true bugs reported by Janzen (1973b) as a complete population, we draw samples ($n = 10, 20, 30, 40, 50$) and estimate the entropy of the population's species-prevalence distribution. RMSE results (Figure 9) are averaged over 1000 trials per population and sample size. Results are summarized by Borda count. Error bars reflect 95% pivot confidence intervals, obtained via 1000 bootstrap replicates over the collections. Gold and silver stars indicate the methods with highest and second-highest aggregated parameter estimates, respectively.

5.3 Results

SENECA consistently outranks other methods for biodiversity quantification across 58 **ecological collections.** We apply the methods described herein to estimate tropical true bug biodiversity, in terms of the entropy their corresponding species-prevalence distributions. We summarize the results in Figure 3, where our method most consistently achieves low RMSE, as judged by Borda count. The next highest-scoring estimators are Chao-Shen and Chao-Wang-Jost, consistent with the under-sampling results of our simulation study, since many of these species-prevalence distributions are heavy-tailed. Chao-Shen and SENECA both take a Horvitz-Thompson-style approach, but SENECA's self-consistent missing mass estimator leads to a Borda count several dozen larger than the former's. Full results for each population are shown in Figure 9 (Appendix C).

On the LLM uncertainty quantification task, SENECA outranks the other entropy-based methods (namely, SE, DSE, KLE, and PE), as measured by aggregated AUROC point estimate (Figure 4). Full results are shown in Figure 10. SE and DSE are canonical notions of the aleatoric component of semantic uncertainty (Park & Cho, 2026); in this setting, SENECA approximates the same estimand. Because SENECA's aggregated AUROC is higher than those of SE and DSE, we conclude that aleatoric uncertainty estimation in LLMs is enhanced by SENECA's alternative entropy estimation strategy. Total predictive uncertainty also includes an epistemic term, which recent works have elicited via methods that include model weight perturbation (Liu et al., 2026), diverse sampling (Park & Cho, 2026), or semantic similarity of responses from an assortment of other models (Hamidieh et al., 2026). Generally, these approaches are computationally involved or call for white-box model access. Nonetheless, SENECA is complementary to such techniques by improving the estimation of the aleatoric term.

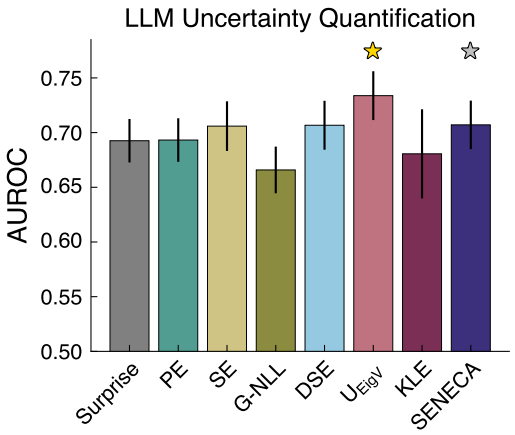

Figure 4: **Summary of LLM UQ results using consistency-based LLM judge.** AUROC values and parameter uncertainties are aggregated across models and datasets, as described in Section 5.2, using the consistency-based LLM-as-judge (Appendix B.5). Gold and silver stars indicate the methods with highest and second-highest aggregated parameter estimates, respectively.

SENECA is competitive with purpose-built LLM hallucination detection methods, such as U_{EigV}, KLE, and G-NLL. On the incorrectness detection task, SENECA attains the second-highest aggregated AUROC point estimate (Figure 4), outranked only by the purpose-built support size estimator U_{EigV}. We note that U_{EigV} is a semantic support size estimator strictly viable for natural language generation (NLG), which incorporates semantic information that is not provided to SENECA. On the other hand, SENECA is a broadly-applicable discrete entropy estimator.

6 Discussion

6.1 Conclusion

We present SENECA, an entropy estimation scheme that excels in the small-sample setting, especially in the presence of under-sampling. In numerical experiments, we demonstrate consistently low RMSE across 72 unique label-prevalence distributions, compared to seven baselines (Section 4.2). We extend this evaluation to 58 ecological collections transcribed from Janzen (1973b), where SENECA also excels. In application to the classification of incorrect LLM responses, the point estimate for SENECA's overall AUROC is second-highest among several purpose-built baselines (Section 5.3). The underlying missing mass estimator may be of independent interest, as well (Appendix F); it estimates the probability that a subsequently sampled item deserves a label that has not yet been observed. We conclude that SENECA may be utile to many diverse technical disciplines.

6.2 Limitations and future work

In this work, our priority is the small-sample setting, where the effects of under-sampling and estimator variance are especially pertinent. Although the method proposed herein exhibits low error in our experiments, it is not unbiased. Indeed, unbiased discrete entropy estimators do not exist in general (Paninski, 2003), and estimators classically may trade off bias and variance (Hastie et al., 2009). As a consequence, there are scenarios for which our estimator may not be best-suited. For instance, in the unusual event that the regime is known to be well-sampled before estimation, our Table 1 indicates that the James-Stein estimator may be preferable. Overall performance may also differ in the large-sample setting, since several entropy estimators are consistent (e.g., Plugin (Antos & Kontoyiannis, 2001; Arora et al., 2022) and Chao-Shen (Vu et al., 2007)).

6.3 REPRODUCIBILITY STATEMENT

Experimental settings, datasets, and baselines are outlined in Sections 4.1 and 5.1, with additional implementation details in Appendix B. Computational resources for LLM experiments are described in Appendix B.3. Use of commercial serverless endpoints with multiple workers make exact compute consumption difficult to track precisely, but we estimate that the LLM experiments took between 30 and 35 GPU-hours. Semantic clustering was performed on a CPU (24 GB RAM), running continuously for roughly 3 days. All remaining experiments and calculations occurred on the same CPU, requiring a few hours of total execution time. The two Phi models are released under the MIT license, and Mistral, Granite, and DeBERTa models under Apache 2.0. The BioASQ dataset is licensed under CC BY 2.5, SQuAD 2.0 and HotpotQA under CC BY SA 4.0, SVAMP and SimpleQA-Verified under MIT, and TruthfulQA under Apache 2.0. Section 5.1.1 describes usage of species-prevalence records, similar to (Chao & Shen, 2003; Chao et al., 2013). We do not re-release these records, but they are visible in their corresponding publication (Janzen, 1973b). We will release a code repository upon publication.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 Technical Report. *arXiv preprint arXiv:2412.08905*, 2024.

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-Mini Technical Report: Compact yet Powerful Multimodal Language Models via Mixture-of-LoRAs. *arXiv preprint arXiv:2503.01743*, 2025.

Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. Rethinking Uncertainty Estimation in LLMs: A Principled Single-Sequence Measure. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wdhruVcRx1.

András Antos and Ioannis Kontoyiannis. Convergence properties of functional estimates for discrete distributions. *Random Structures & Algorithms*, 19(3-4):163–193, 2001.

Aryaman Arora, Clara Meister, and Ryan Cotterell. Estimating the Entropy of Linguistic Distributions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 175–195, 2022.

Haricharan Balasundaram and Andrew Thangaraj. Estimating Error in Natural Distribution Estimation. In *2025 61st Allerton Conference on Communication, Control, and Computing Proceedings*. Allerton Conference on Communication, Control, and Computing, 2025.

Georgij P Basharin. On a Statistical Estimate for the Entropy of a Sequence of Independent Random Variables. *Theory of Probability & Its Applications*, 4(3):333–336, 1959.

Juan A Bonachela, Haye Hinrichsen, and Miguel A Munoz. Entropy estimates of small data sets. *Journal of Physics A: Mathematical and Theoretical*, 41(20):202001, 2008.

Richard L Burden and J Douglas Faires. Accelerating Convergence. In *Numerical Analysis*, pp. 89. Brooks/Cole, Cengage Learning, 9th edition, 2011. ISBN 978-0538733519.

Carlson Moses Büth, Kishor Acharya, and Massimiliano Zanin. infomeasure: a comprehensive Python package for information theory measures and estimators. *Scientific Reports*, 15(1):29323, 2025.

Anne Chao. Nonparametric Estimation of the Number of Classes in a Population. *Scandinavian Journal of Statistics*, pp. 265–270, 1984.

Anne Chao. Species estimation and applications. In Samuel Kotz, Campbell B Read, Brani Vidakovic, and Norman L. Johnson (eds.), *Encyclopedia of Statistical Sciences*, volume 12, pp. 7907–7916. Wiley, New York, 2 edition, 2005.

Anne Chao and Lou Jost. Diversity measures. *Encyclopedia of Theoretical Ecology*, pp. 203–207, 2012.

Anne Chao and Tsung-Jen Shen. Nonparametric estimation of Shannon's index of diversity when there are unseen species in sample. *Environmental and Ecological Statistics*, 10:429–443, 2003.

Anne Chao, Yun Tao Wang, and Lou Jost. Entropy and the species accumulation curve: a novel entropy estimator via discovery rates of new species. *Methods in Ecology and Evolution*, 4(11): 1091–1100, 2013.

I Chien and Olgica Milenkovic. Regularized Weighted Chebyshev Approximations for Support Estimation. *arXiv preprint arXiv:1901.07506*, 2019.

Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach. *Biometrics*, pp. 837–845, 1988.

Yongcui Deng, Alexander K Umbach, and Josh D Neufeld. Nonparametric richness estimators Chao1 and ACE must not be used with amplicon sequence variant data. *The ISME Journal*, 18 (1):wrae106, 2024.

Rebecca DerSimonian and Nan Laird. Meta-analysis in clinical trials. *Controlled Clinical Trials*, 7 (3):177–188, 1986.

Bradley Efron. Beiter Bootstrap Confidence Intervals. *Journal of the American statistical Association*, 82(397):171–185, 1987.

Johanna Elena Schmitz and Sven Rahmann. A comprehensive review and evaluation of species richness estimation. *Briefings in Bioinformatics*, 26(2):bbaf158, 2025.

Nicolaas (Klaas) M Faber. Estimating the uncertainty in estimates of root mean square error of prediction: application to determining the size of an adequate test set in multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 49(1):79–89, 1999.

Sebastian Farquhar, Joel Kossen, Lukas Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0.

Irving J Good. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, 40(3-4):237–264, 1953.

IBM Granite Team. Granite 3.0 Language Models, October 2024. URL https://github.com/ibm-granite/granite-3.0-language-models/.

Peter Grassberger. Finite sample corrections to entropy and dimension estimates. *Physics Letters A*, 128(6-7):369–373, 1988.

Peter Grassberger. Entropy Estimates from Insufficient Samplings. *arXiv preprint physics/0307138*, 2003.

Mario C Grignetti. A Note on the Entropy of Words in Printed English. *Information and Control*, 7 (3):304–306, 1964.

Lukas Haas, Gal Yona, Giovanni D'Antonio, Sasha Goldshtein, and Dipanjan Das. SimpleQA Verified: A Reliable Factuality Benchmark to Measure Parametric Knowledge. *arXiv preprint arXiv:2509.07968*, 2025.

Kimia Hamidieh, Veronika Thost, Walter Gerych, Mikhail Yurochkin, and Marzyeh Ghassemi. Complementing Self-Consistency with Cross-Model Disagreement for Uncertainty Quantification. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=lOoRJo8xWy.

Yi Hao and Ping Li. Optimal Prediction of the Number of Unseen Species with Multiplicity. *Advances in Neural Information Processing Systems*, 33:8553–8564, 2020.

Joachim Hartung and Guido Knapp. On tests of the overall treatment effect in meta-analysis with normally distributed responses. *Statistics in Medicine*, 20(12):1771–1782, 2001.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Model Selection and the Bias-Variance Tradeoff. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, pp. 37–38. Springer, 2009. ISBN 978-0387848570.

Jean Hausser and Korbinian Strimmer. Entropy Inference and the James-Stein Estimator, with Application to Nonlinear Gene Association Networks. *Journal of Machine Learning Research*, 10 (7), 2009.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations*, 2021.

Daniel G Horvitz and Donovan J Thompson. A Generalization of Sampling Without Replacement from a Finite Universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

IBM Research. Granite 4.0 language models. https://github.com/ibm-granite/granite-4.0-language-models, 2025.

Joanna IntHout, John Ioannidis, and George F Borm. The Hartung-Knapp-Sidik-Jonkman method for random effects meta-analysis is straightforward and considerably outperforms the standard DerSimonian-Laird method. *BMC Medical Research Methodology*, 14(1):25, 2014.

Mete Ismayilzada, Antonio Laverghetta Jr., Simone A. Luchini, Reet Patel, Antoine Bosselut, Lonneke Van Der Plas, and Roger E. Beaty. Creative Preference Optimization. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 9580–9609, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.509. URL https://aclanthology.org/2025.findings-emnlp.509/.

Daniel H Janzen. Sweep Samples of Tropical Foliage Insects: Description of Study Sites, With Data on Species Abundances and Size Distributions. *Ecology*, 54(3):659–686, 1973a.

Daniel H Janzen. Sweep Samples of Tropical Foliage Insects: Effects of Seasons, Vegetation Types, Elevation, Time of Day, and Insularity. *Ecology*, 54(3):687–708, 1973b.

Biing-Hwang Juang and SH Lo. On the bias of the Turing-Good estimate of probabilities. *IEEE Transactions on Signal Processing*, 42(2):496–498, 2002.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language Models (Mostly) Know What They Know. *arXiv preprint arXiv:2207.05221*, 2022.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. BioASQ-QA: A manually curated corpus for Biomedical Question Answering. *Scientific Data*, 10(1):170, 2023.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Ilaria Pia La Torre, David A Kelly, Hector D Menendez, and David Clark. To BEE or Not to BEE: Estimating more than Entropy with Biased Entropy Estimators. *arXiv preprint arXiv:2501.11395*, 2025.

Seongmin Lee and Marcel Böhme. How Much Is Unseen Depends Chiefly on Information About the Seen. In *Proceedings of the 13th International Conference on Learning Representations*, May 2025.

Linlin Li, Ivan Titov, and Caroline Sporleder. Improved Estimation of Entropy for Evaluation of Word Sense Induction. *Computational Linguistics*, 40(3):671–685, September 2014. doi: 10.1162/COLI_a_00196. URL https://aclanthology.org/J14-3007/.

Yanpeng Li and Boping Tian. On the sub-Gaussianity of the missing mass. *Statistics & Probability Letters*, 206:109991, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with Confidence: Uncertainty Quantification for Black-box Large Language Models. *Transactions on Machine Learning Research*, 2024, 2024. ISSN 2835-8856.

Litian Liu, Reza Pourreza, Sunny Panchal, Apratim Bhattacharyya, Yubing Jian, Yao Qin, and Roland Memisevic. Enhancing Hallucination Detection through Noise Injection. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=WnM3sluiVn.

David McAllester and Robert E Schapire. Learning theory and language modeling. In *Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

David A McAllester and Robert E Schapire. On the Convergence Rate of Good-Turing Estimators. In *Conference on Learning Theory*, pp. 1–6, 2000.

Lucas Hurley McCabe, Rimon Melamed, Thomas Hartvigsen, and H Howie Huang. Estimating Semantic Alphabet Size for LLM Uncertainty Quantification. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=uYK6GPVg1O.

Ilya Nemenman, Fariel Shafee, and William Bialek. Entropy and Inference, Revisited. *Advances in Neural Information Processing Systems*, 14, 2001.

Dang Nguyen, Ali Payani, and Baharan Mirzasoleiman. Beyond Semantic Entropy: Boosting LLM Uncertainty Quantification with Pairwise Semantic Similarity. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4530–4540, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel Language Entropy: Fine-grained Uncertainty Quantification for LLMs from Semantic Similarities. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8901–8929, 2024.

Sebastian Nowozin. Improved Information Gain Estimates for Decision Tree Induction. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 571–578, 2012.

Alon Orlitsky and Ananda Theertha Suresh. Competitive Distribution Estimation: Why is Good-Turing Good. *Advances in Neural Information Processing Systems*, 28, 2015.

Alon Orlitsky, Ananda Theertha Suresh, and Yihong Wu. Optimal prediction of the number of unseen species. *Proceedings of the National Academy of Sciences*, 113(47):13283–13288, 2016.

Amichai Painsky. Generalized Good-Turing Improves Missing Mass Estimation. *Journal of the American Statistical Association*, 118(543):1890–1899, 2023.

Liam Paninski. Estimation of Entropy and Mutual Information. *Neural Computation*, 15(6):1191–1253, 2003.

Ji Won Park and Kyunghyun Cho. Efficient semantic uncertainty quantification in language models via diversity-steered sampling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=IiEtQPGVyV.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP Models really able to Solve Simple Math Word Problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168.

Assaf Pinchas, Irad Ben-Gal, and Amichai Painsky. A Comparative Analysis of Discrete Entropy Estimators for Large-Alphabet Problems. *Entropy*, 26(5):369, 2024.

Nikhilesh Rajaraman, Andrew Thangaraj, and Ananda Theertha Suresh. Minimax Risk for Missing Mass Estimation. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 3025–3029. IEEE, 2017.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Vishal Rana, Eli Chien, Jianhao Peng, and Olgica Milenkovic. Small-Sample Estimation of the Mutational Support and Distribution of SARS-CoV-2. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(1):668–682, 2022.

Michael Roswell, Jonathan Dushoff, and Rachael Winfree. A conceptual guide to measuring species diversity. *Oikos*, 130(3):321–338, 2021.

Claude Elwood Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

William B Sherwin and Narcis Prat i Fornells. The Introduction of Entropy and Information Methods to Ecology by Ramon Margalef. *Entropy*, 21(8):794, 2019.

Kurex Sidik and Jeffrey N Jonkman. A simple confidence interval for meta-analysis. *Statistics in Medicine*, 21(21):3153–3159, 2002.

Maciej Skorski. Mean-squared accuracy of good-turing estimator. In *IEEE International Symposium on Information Theory*, pp. 2846–2851. IEEE, 2021.

Maciej Skorski. On Sub-Gaussian Concentration of Missing Mass. *Theory of Probability & Its Applications*, 68(2):324–329, 2023.

Xu Sun and Weichao Xu. Fast Implementation of DeLong's Algorithm for Comparing the Areas Under Correlated Receiver Operating Characteristic Curves. *IEEE Signal Processing Letters*, 21 (11):1389–1393, 2014.

Gregory Valiant and Paul Valiant. Estimating the Unseen: Improved Estimators for Entropy and Other Properties. *Journal of the ACM (JACM)*, 64(6):1–41, 2017.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Vincent Q Vu, Bin Yu, and Robert E Kass. Coverage-adjusted entropy estimation. *Statistics in Medicine*, 26(21):4039–4060, 2007.

Yihong Wu and Pengkun Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *The Annals of Statistics*, 47(2):857–883, 2019.

Abraham J Wyner and Dean Foster. On the lower limits of entropy estimation. *IEEE Transactions on Information Theory*, 2003.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259.

A SUPPORTING LEMMAS

Lemma 1. *Given υ and X^n, Eq. 5 has a fixed point.*

Proof. For brevity, we will write $\mu(m, \upsilon, X^n)$ as $\mu(m)$. $\mu(m)$ is the sum of positive polynomials, which are continuous everywhere, so $\mu(m)$ is continuous on $[0, 1]$. If $m \in [0, 1]$, then $\mu(m) \leq (1-m)\left(\sum_{u \in X^n_{\neq}} \hat{p}_u^{ML}\right) + m = 1$. Each term of $\mu(m)$ is positive, so $m \in [0, 1]$ implies $\mu(m) \in [0, 1]$, establishing self-mapping over $[0, 1]$. Therefore, a fixed point exists, by the 1D Brouwer Fixed Point Theorem (i.e., Intermediate Value Theorem). □

Lemma 2. *Let $X^n \sim P$ be a multiset of n IID samples drawn from a multinomial distribution P. Then $\hat{M}^{SC}(X^n) \in [0, 1)$.*

Proof. By Lemma 1, μ is self-mapping over $[0, 1]$. Since $\mu(1) = \left(1 - \frac{1}{\max(\upsilon, 1)}\right)^n \in [0, 1)$, the fixed point cannot be 1, and the result follows. □

Lemma 3. *Let $X^n \sim P$ be a multiset of n IID samples drawn from a multinomial distribution P. Then $\hat{M}^{SC}(X^n) \to 0$ as $n \to \infty$.*

Proof. The summand of the first term of $\mu(m, \upsilon, X^n)$ is maximized by $(1-m)\hat{p}_u^{ML} = \frac{1}{n+1}$ (see Lemma 1 of Skorski (2021)), so:

$$\begin{aligned}(1-m)\sum_{u \in X^n_{\neq}} \hat{p}_u^{ML}\left((1-(1-m)\hat{p}_u^{ML}\right)^n &\leq \sum_{u \in X^n_{\neq}} \frac{1}{n+1}\left(1 - \frac{1}{n+1}\right)^n \\ &\leq |\mathcal{X}|\frac{1}{n+1}\left(1 - \frac{1}{n+1}\right)^n \\ &= O\left(\frac{1}{n}\right).\end{aligned}$$

The second term is similar:

$$\begin{aligned}m\left(1 - \frac{m}{\max(\upsilon, 1)}\right)^n &= \max(\upsilon, 1)\frac{m}{\max(\upsilon, 1)}\left(1 - \frac{m}{\max(\upsilon, 1)}\right)^n \\ &\leq \max(\upsilon, 1)\frac{1}{n+1}\left(1 - \frac{1}{n+1}\right)^n \\ &\leq |\mathcal{X}|\frac{1}{n+1}\left(1 - \frac{1}{n+1}\right)^n \\ &= O\left(\frac{1}{n}\right).\end{aligned}$$

The result follows from selecting an arbitrarily small $m^* \in (0, 1]$ for which there is an $n^* : n > n^*$ implies $\mu(m) < m^*$ for all $m \in [0, 1]$ and observing that, consequently, fixed-point iteration on $\mu(m)$ cannot exceed m^* for such $n > n^*, m \in [0, 1]$. □

Lemma 4. *(Extension of Vu et al. (2007).) Let $X^n \sim P$ be a multiset of n IID samples drawn from a multinomial distribution P with finite entropy variance.*[3] *Let $\hat{M}$ be any missing mass estimator satisfying $\hat{M} \in [0, 1)$ and $\hat{M} \xrightarrow{p} 0$ as $n \to \infty$. Let $\hat{H}^{HT}_{\hat{M}}$ denote its corresponding "Horvitz-Thompson-style" coverage-adjusted entropy estimator, i.e.:*

$$\hat{H}^{HT}_{\hat{M}}(X^n) = -\sum_{u \in X^n_{\neq}} \frac{\hat{p}_u^{HT} \log \hat{p}_u^{HT}}{1 - (1 - \hat{p}_u^{HT})^n},$$

where $\hat{p}_u^{HT} = (1 - \hat{M}(X^n))\hat{p}_u^{ML}$. Then $\hat{H}^{HT}_{\hat{M}} \xrightarrow{p} 0$ as $n \to \infty$.

[3] i.e., $\sum_{u \in \mathcal{X}} p_u (\log p_u)^2$ (Wyner & Foster, 2003; Vu et al., 2007), which is finite for fixed distributions over finite support.

Algorithm 1 SENECA

Require: Set of label-occurrences X^n, initial support size estimate $\widehat{|\mathcal{X}|}$.
Ensure: Entropy estimate $\hat{\mathbb{H}}^{SC}$, missing mass estimate $\hat{M}^{SC}$

```
P^ML ← {N_u(X^n)/n | u ∈ X^n_≠}        ▷ Empirical prevalences p̂^ML_u for observed labels
v ← max(|X̂| − |X^n_≠|, 1)               ▷ Estimate excess support size
m_0 ← 1/2                                ▷ Initial guess
Perform fixed-point iteration on μ (Eq. 4) to obtain M̂^SC
if M̂^SC ∈ (−5 × 10^−10, 0) then
    M̂^SC ← 0                            ▷ Floating-point correction (Appendix B.1)
end if
Estimate entropy via Eq. 6               ▷ Horvitz–Thompson estimator using M̂^SC
```

Proof. The following inequality, valid for missing mass estimators satisfying $\hat{M} \in [0, 1)$ and $\hat{M} \xrightarrow{p} 0$ over distributions with finite entropy variance, arises from Vu et al. (2007) (the proof of the authors' Theorem 1):

$$|\hat{H}^{HT}_{\hat{M}} - \hat{H}^{plugin}| \leq A_n + B_n + C_n + D_n \text{ , where}$$

$$A_n = \frac{\hat{M}}{1 - (1 - \frac{1-\hat{M}}{n})^n} \hat{H}^{plugin}$$

$$B_n = \frac{(1 - \frac{1-\hat{M}}{n})^n}{1 - (1 - \frac{1-\hat{M}}{n})^n} \left(\frac{(\log n)^2}{\sqrt{n}} + \log n \sum_{u: p_u \leq \frac{1}{\sqrt{n}}} \hat{p}^{ML}_u \right)$$

$$C_n = \frac{(1 - \frac{\log n}{n})^n}{1 - (1 - \frac{\log n}{n})^n} \hat{H}^{plugin} \text{ , and}$$

$$D_n = \frac{|\log(1 - \hat{M})|}{1 - (1 - \frac{1-\hat{M}}{n})^n},$$

with B_n being $O_P(\frac{1}{\log n})$ and C_n being $O_P(\frac{1}{n})$ (Vu et al., 2007). Reviewing the remaining terms:

$$A_n \leq \frac{\hat{M}}{1 - \exp(\hat{M} - 1)} \hat{H}^{plugin} \xrightarrow{p} 0 \text{ , and}$$

$$D_n \leq \frac{|\log(1 - \hat{M})|}{1 - \exp(\hat{M} - 1)} \xrightarrow{p} 0, \tag{7}$$

Therefore, $|\hat{H}^{HT}_{\hat{M}} - \hat{H}^{plugin}| \xrightarrow{p} 0$ and $\hat{H}^{HT}_{\hat{M}} \xrightarrow{p} \hat{H}^{plugin}$. Consequently, $\hat{H}^{HT}_{\hat{M}} \xrightarrow{p} \hat{H}^{plugin}$ and $\hat{H}^{HT}_{\hat{M}}$ converges in probability to the true entropy. □

B Additional implementation details

B.1 Entropy estimation

Algorithm 1 specifies the procedure for calculating SENECA. Our method involves fixed-point iteration, for which we employ Steffensen's Method (Burden & Faires, 2011), via the SciPy library (Virtanen et al., 2020). We begin the iteration with an initial value of 0.5. The true fixed point is guaranteed to lie in $[0, 1)$ (Lemma 2), but we observe occasional instances in which the optimizer terminates at an exceedingly small negative value in $(-5 \times 10^{-10}, 0)$, which we attribute to floating-point error; our implementation clips such values to 0.

Most other entropy estimators - i.e., Grassberger, James-Stein, Bonachela, Chao-Shen, Chao-Wang-Jost, and NSB (see Appendix D) - are implemented via the *infomeasure* library (Büth et al., 2025). We note that the library's implementation of the Chao-Shen estimator applies an adjusted Good-

Turing estimator $\hat{M}^{GT} = \frac{\Phi(X^n)-1}{n}$ when $\Phi(X^n) = n$, to avoid summands of $\frac{0 \log 0}{0}$. The Valiant-Valiant entropy estimator is a linear programming-based method, implemented by translating the authors' MATLAB code to Python.

B.2 Support size estimation

SENECA uses an estimate of the support size to calculate the "unobserved" component of Eq. 4. Except where stated otherwise, we employ the regularized weighted Chebyshev (RWC) estimator of Chien & Milenkovic (2019); Rana et al. (2022), which is intended for small sample sizes. In particular, we employ their RWC-S variant, which exhibits the best worst-case performance in the source material (Chien & Milenkovic, 2019). In this section, we will briefly describe this and three other options and illustrate that our results are generally not sensitive to our choice of support size estimator.

The straightforward-to-calculate Chao1 support estimator employs two FoFs (Chao, 1984):

$$\widehat{|\mathcal{X}|}_{Chao1,1} = |\mathcal{X}^n_{\neq}| + \frac{\Phi_1(X^n)^2}{2\Phi_2(X^n)}. \tag{8}$$

In population ecology, where support size is often known as "species richness," Chao1 tends to underestimate its quantity of interest, though it performs well for power law distributions (Elena Schmitz & Rahmann, 2025). A variation that avoids division by zero in the absence of "doubletons" is given by:

$$\widehat{|\mathcal{X}|}_{Chao1,2} = |\mathcal{X}^n_{\neq}| + \frac{\Phi_1(X^n)\Big(\Phi_1(X^n) - 1\Big)}{2\Big(\Phi_2(X^n) + 1\Big)}, \tag{9}$$

a "bias-corrected" version (Chao, 2005) that is "now widely preferred" to Eq. 8 (Deng et al., 2024). Several more recent methods, such as the aforementioned RWC (Chien & Milenkovic, 2019; Rana et al., 2022), that of Wu & Yang (2019) (Wu-Yang), and that of Valiant & Valiant (2017) (Valiant-Valiant), rely on more involved techniques at the expense of computational complexity. We discuss pertinent hyperparameters for the latter three below, but the remaining implementation details are beyond the scope of this work, and we refer interested readers to their corresponding publications.

The Wu-Yang, RWC, and Valiant-Valiant support size estimators require a regularization parameter p_{min}, corresponding to the assumption that all labels have true probability mass at least $p_{min} = \min\{p_u \forall u \in \mathcal{X}\}$. Since p_{min} is generally unknown, we apply a heuristic inspired by that of Valiant & Valiant (2017) and Thm. 1 of Wu & Yang (2019):

$$\hat{p}_{min} = \begin{cases} \left(\frac{\arg\min_{i:\Phi_i(X^n)>0} \Phi_i(X^n)}{n}\right)^2, & \text{if } \Phi_1(X^n) = 0 \\ \frac{1}{\arg\max_{\gamma \in \mathbb{Z}} (n > \frac{\gamma}{\log \gamma})}, & \text{otherwise.} \end{cases}$$

The heuristic for the zero-singletons cases is drawn directly from the source code of Valiant & Valiant (2017). The other heuristic arises from the result of Wu & Yang (2019) that consistent estimation of $|\mathcal{X}|$ is only possible with at least $\frac{|\mathcal{X}|}{\log |\mathcal{X}|}$ samples; since a uniform distribution over $|\mathcal{X}|$ maximizes the minimum non-zero mass, we select the largest integer value of γ such that $n > \frac{\gamma}{\log \gamma}$. We employ the above heuristic for the RWC and Wu-Yang estimators, which do not have their own heuristic for the regularization parameter. The Valiant-Valiant estimator is implemented with its own original heuristic. The Wu-Yang method requires two additional hyperparameters: γ, c_0, and c_1, for which we employ the default settings from the source material's experiments (i.e., $c_0 = 0.45$, $c_1 = 0.5$) (Wu & Yang, 2019). The RWC support estimator admits additional hyperparameters c_0, dictating the polynomial degree, and nC, dictating the number of grid points. We apply the default of $c_0 = 0.558$ from the paper and $nC = 1000$, as recommended for $n \leq 1000$ in the corresponding source code (Chien & Milenkovic, 2019; Rana et al., 2022). Finally, we clip the lower bound of all support size estimates used in SENECA to ensure that the estimated number of unobserved labels is at least one.

In Figure 5, we consider the bias-corrected Chao1, Wu-Yang, RWC, and Valiant-Valiant methods as drop-in support size estimators in SENECA, alongside plugin entropy for reference. We find that

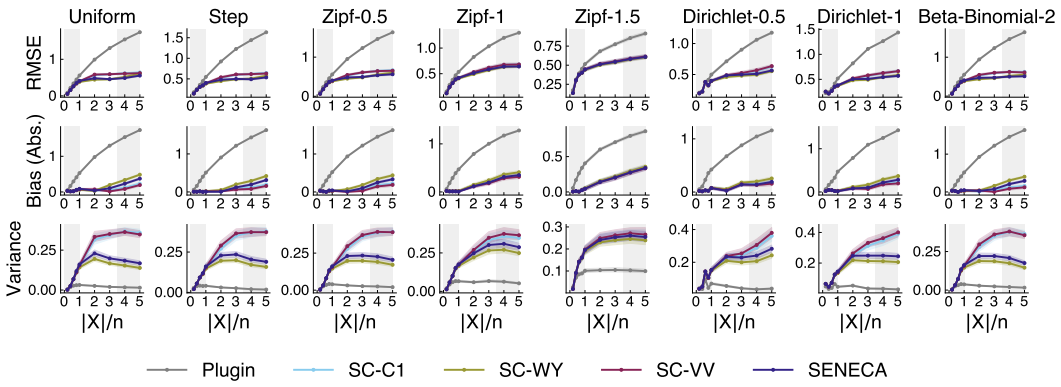

Figure 5: **Support size estimator ablation for SENECA implementations.** The entropy estimators shown are the plugin estimator (Plugin), SENECA with bias-corrected Chao1 support estimation (SC-C1), SENECA with Wu-Yang support estimation (SC-WY), SENECA with Valiant-Valiant support estimation (SC-VV), and SENECA with RWC support estimation, as used in the remainder of this work (SENECA). Each item's capture probability is determined uniformly over its label's prevalence probability mass. For each of 8 families of label-prevalence distribution, we generate 100 items per label. The number of labels $|\mathcal{X}|$ varies from 2 to 50 across trials. For each trial, $n = 10$ item-capture samples are drawn, and the missing mass of the label-prevalence distribution is estimated according to each method. Results are averaged over 1000 trials per label count. Colored regions about each line plot depict 95% bias-corrected and accelerated confidence intervals, based on 1000 bootstrap replications. In each plot, leftmost grey region corresponds to the well-sampled regime (i.e., $|\mathcal{X}| = \frac{n}{5}, \frac{2n}{5}, \frac{3n}{5}, \frac{4n}{5}, n$), where items to the right correspond to the under-sampled regime (i.e., $|\mathcal{X}| = 2n, 3n, 4n, 5n$). The rightmost grey region corresponds to the support-risky regime, where no consistent estimator for support size exists.

alternative choices of support estimator lead to roughly interchangeable results to those presented in the main body of this work. Minor differences are apparent in the bias-variance decomposition. For instance, the variants using Chao1 and Valiant-Valiant tend to exhibit slightly lower bias and higher variance in the under-sampled regime, and vice-versa for the Wu-Yang and RWC variants.

B.3 Text Generation Details

For each model-dataset pair, we sample $n = 10$ responses of up to 100 tokens each, at a temperature of $\tau = 1.0$ with nucleus sampling $p = 0.9$, following Farquhar et al. (2024). Following Kuhn et al. (2023), log-probabilities for these high-temperature sequences are length-normalized (i.e., by number of tokens). To support the G-NLL method, we also sample a response with temperature $\tau = 0$ and nucleus sampling $p = 1.0$ (Aichberger et al., 2026); log-probabilities for this greedy sequence are *not* length-normalized, which the authors contend works best. LLM sampling was performed vLLM (Kwon et al., 2023) serverless endpoints hosted on Runpod, with KV cache reuse where possible. All LLMs were hosted on NVIDIA RTX 4090 GPUs (24 GB VRAM), with the exception of Phi-4, which was hosted on NVIDIA A40 instances (48 GB VRAM), and Granite-4.0-H-Small, which ran on RTX PRO 6000s (96 GB VRAM). Following Farquhar et al. (2024); Nikitin et al. (2024), we pre-pend all queries with the following to induce sentence-length responses:

Template for Pre-pending Prompts

Answer the following question in a single brief but complete sentence:

Two LLMs required custom prompt templates to induce well-formed responses. For Phi-4, we used the following, based on recommendation in its corresponding HuggingFace listing:[4]

[4] https://huggingface.co/microsoft/phi-4

Prompt Template for Phi-4

<|im_start|>user
{prompt}<|im_end|>
<|im_start|>assistant

We used the following for Granite-4.0-H-Small, based on recommendation in its HuggingFace listing:[5]

Prompt Template for Granite-4.0-H-Small

<|start_of_role|>user<|end_of_role|>
{prompt}<|end_of_text|>
<|start_of_role|>assistant<|end_of_role|>

B.4 SEMANTIC CLUSTERING

To assign "semantic equivalence class" labels to LLM responses, we employ the bidirectional entailment clustering (BEC) procedure of Kuhn et al. (2023) with "strict entailment", where responses are greedily assigned to the same class if they are deemed semantically equivalent, and instantiate new classes otherwise. To assess semantic equivalence, we employ a DeBERTa-based NLI model, with the prompt pre-pended to each response; in particular, we use a fine-tuned version of *DeBERTaV3* (He et al., 2021).[6] Before prompting the NLI model, we remove the pre-pended portion of the query (i.e., "Answer the following question in a single brief but complete sentence:") to limit extraneous context.

B.5 LLM-AS-JUDGE

To assess the correctness of LLM responses, we compare reference answers to a single "best guess" response sampled at $\tau = 0.1$, following prior work (Farquhar et al., 2024; Nikitin et al., 2024; Nguyen et al., 2025). McCabe et al. (2026) uses the same LLM-as-judge prompt as Lin et al. (2024), but adds an in-context example and XML-style tags to improve rating parsing. We add an additional instruction to limit verbose responses:

Prompt for Assessing Correctness of LLM Generations

Rate the level of consistency between the answer to the question and the reference answer, from 0 to 100. Output the float rating inside <rating>< /rating> tags. Only provide the rating, without any additional commentary.
Here is an example output:
Question: What is the capital of France? Reference: Paris is the capital of France. Answer: The capital of France is Paris. <rating>100< /rating>
Now rate the following:
Question: {question}
Reference: {groundtruth}
Answer: {pred}

If multiple reference examples are available, we retain the highest such rating among them. Like Lin et al. (2024), we treat a response as correct if its ultimate rating exceeds 70/100. Our judge model is the 32B-parameter Granite-4.0-H-Small, generating at most 100 tokens using a temperature of $\tau = 0.1$. Like in the semantic clustering, we remove the pre-pended portion of the query before prompting the judge. In Appendix E, we consider an alternative LLM judge also found in related work, with qualitatively interchangeable results.

[5] https://huggingface.co/ibm-granite/granite-4.0-h-small
[6] https://huggingface.co/cross-encoder/nli-deberta-v3-base

B.6 LLM UNCERTAINTY QUANTIFICATION METHODS

In our notation, variability-based uncertainty quantification techniques obtain a set A^n of n LLM responses to a query q. The semantic clustering algorithm (e.g., BEC, Appendix B.4) serves as the classification function h that produces the set of label occurrences X^n and corresponding set of unique observed labels $X^n_{\neq}$. Discrete semantic entropy (DSE) simply reports the plugin entropy estimate over $X^n_{\neq}$ (Farquhar et al., 2024). The "white-box" semantic entropy (SE) replaces the empirical label frequencies of plugin entropy with probability estimates based on sequence log-probabilities (Kuhn et al., 2023):[7]

$$SE(q) = -\sum_{x \in X^n_{\neq}} \hat{p}(x|q) \log \hat{p}(x|q), \text{ where}$$

$$\hat{p}(x|q) = \frac{\sum_a \mathbb{1}_{h(a)=x} p(a|q)}{\sum_a p(a|q)}.$$

Kernel language entropy (KLE) does not explicitly use semantic clusters. Instead, it constructs an NLI-based semantic graph of responses, applies a kernel to the graph's Laplacian, and reports the von Neumann entropy of the resultant matrix. Letting $f_{ij} = 1, 0$, or 0.5 if the NLI model gauges entailment, contradiction, or neutral, respectively, for response pair a_i, a_j, the semantic graph has $n \times n$ weighted adjacency matrix whose entries are given by $w_{ij} = f_{ij} + f_{ji}$. Our implementation uses the heat kernel $K_t = exp(-tL)$, which the authors report favorably, with "reasonable" default hyperparameter $t = 0.3$ (Nikitin et al., 2024).

Like KLE, U_{EigV} also relies on an NLI-based semantic graph, though its particular construction differs. Using the NLI model, the bidirectional entailment probability e_{ij} is calculated for each response pair a_i, a_j. The matrix of weights $w_{ij} = \frac{e_{ij}+e_{ji}}{2}$ serves as the weighted adjacency matrix of a semantic graph. The metric is then calculated based on the graph Laplacian's spectrum $\Lambda(L)$:

$$U_{EigV} = \sum_{\lambda \in \Lambda(L)} \max(0, 1 - \lambda),$$

which is interpreted as an estimate of the support size $|\mathcal{X}|$ (Lin et al., 2024). For consistency, we use the same NLI model for all relevant methods (i.e., those relying on semantic clustering or semantic graphs).

Predictive entropy (PE) and surprise are straightforward functions of the log-probabilities of sampled sequences. We calculate the former by the average of negative length-normalized sequence log-probabilities (Kadavath et al., 2022) and the latter by average exponentiated (base 2) negative length-normalized sequence log-probabilities (Ismayilzada et al., 2025). G-NLL calls for one response to be drawn using greedy decoding, reporting the corresponding raw (i.e., not length-normalized) negative sequence log-probability (Aichberger et al., 2026).

C ADDITIONAL ENTROPY ESTIMATION RESULTS

We expand Figure 2 to display all considered entropy estimators in Figure 6, where SENECA continues to achieve low RMSE in the under-sampled regime. In Table 2, we repeat the analysis of Table 1, using a twice-large sample size ($n = 20$). Results are qualitatively similar to the $n = 10$ setting, especially in the under-sampled regime, where SENECA achieves the lowest or second-lowest averaged error in all cases but one. When SENECA is not the lowest-error estimator (e.g., in the well-sampled regime), its error is within a few percentage points of the lowest. With this larger sample size, the James-Stein estimator is no longer top-performing in the well-sampled regime, and the Chao-Shen estimator improves relatively.

[7] We have detailed DSE and SE in reverse-chronological order of publication for descriptive convenience.

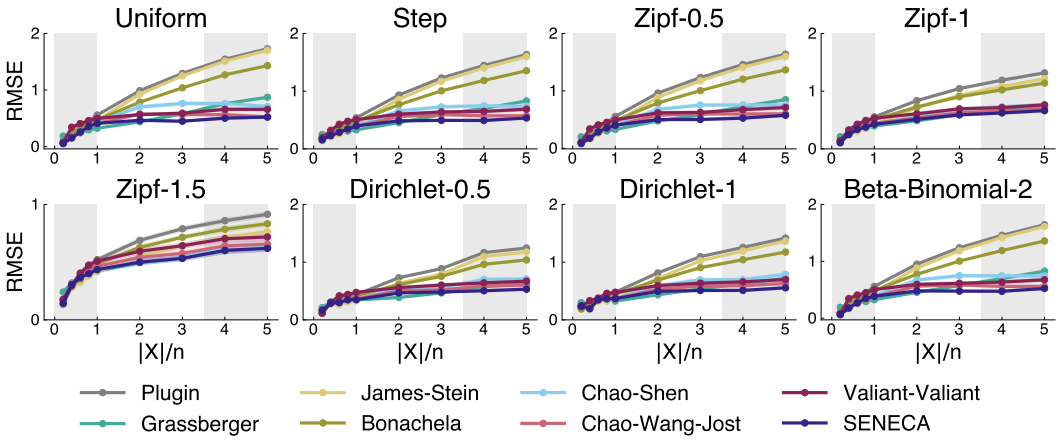

Figure 6: **Entropy estimation in small-sample item capture (all estimators).** For each of 8 families of label-prevalence distribution, each item's capture probability is determined uniformly over its label's prevalence probability mass. The number of labels $|\mathcal{X}|$ varies from 2 to 50 across trials. For each trial, $n = 10$ items are captured, and the entropy of the label-prevalence distribution is estimated according to each of seven methods. Results are averaged over 1000 trials per label count. Colored regions about each line plot depict 95% bias-corrected and accelerated confidence intervals, based on 1000 bootstrap replications. In each plot, the leftmost grey region corresponds to the well-sampled regime (i.e., $|\mathcal{X}| = \frac{n}{5}, \frac{2n}{5}, \frac{3n}{5}, \frac{4n}{5}, n$), where items to the right correspond to the under-sampled regime (i.e., $|\mathcal{X}| = 2n, 3n, 4n, 5n$). The rightmost grey region corresponds to the support-risky regime, where no consistent estimator for support size exists. Empirical estimator variances are shown in the corresponding Figure 7.

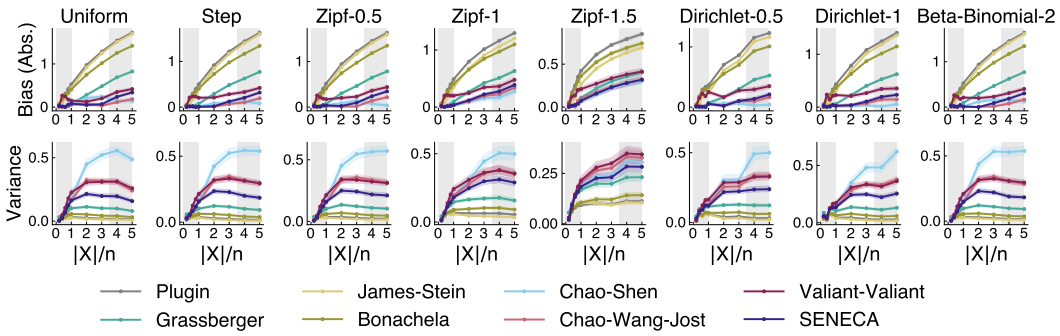

Figure 7: **Empirical bias-variance decomposition of entropy estimation in small-sample item capture.** For each of 8 families of label-prevalence distribution, we generate 100 items per label. Each item's capture probability is determined uniformly over its label's prevalence probability mass. The number of labels $|\mathcal{X}|$ varies from 2 to 50 across trials. For each trial, $n = 10$ item-capture samples are drawn, and the entropy of the label-prevalence distribution is estimated according to each of seven methods. Results are averaged over 1000 trials per label count. Colored regions about each line plot depict 95% bias-corrected and accelerated confidence intervals, based on 1000 bootstrap replications. In each plot, leftmost grey region corresponds to the well-sampled regime (i.e., $|\mathcal{X}| = \frac{n}{5}, \frac{2n}{5}, \frac{3n}{5}, \frac{4n}{5}, n$), where items to the right correspond to the under-sampled regime (i.e., $|\mathcal{X}| = 2n, 3n, 4n, 5n$). The rightmost grey region corresponds to the support-risky regime, where no consistent estimator for support size exists.

Table 2: **Regime-averaged error for entropy estimation in small-sample item capture.** We repeat the analysis of Table 1 using $n = 20$ instead of $n = 10$. We calculate the MSE for each estimator normalized by that of the worst estimator and average results within the well-sampled (Well, i.e., $|\mathcal{X}| \leq n$) and under-sampled (Under, i.e., $|\mathcal{X}| > n$) regimes (Reg.). The estimators are Plugin (ML), Grassberger (G), James-Stein (JS), Bonachela (B), Chao-Shen (CS), Chao-Wang-Jost (CWJ), Valiant-Valiant (VV), and SENECA (highlighted in blue). The distributions are Uniform, Step, Zipf-0.5, Zipf-1, Zipf-1.5, Dirichlet-0.5 (Diri-0.5), Dirichlet-1 (Diri-1), and Beta-Binomial-2 (BB-2). The lowest and second-lowest values in each row, prior to rounding, are bold and underlined, respectively. Bootstrapped confidence intervals (grey text) are slightly more conservative than 95%, as described in Section 4.1. In parentheses, we show the relative difference between SENECA's averaged RMSE and the lowest value in each row.

Distr.	Reg.	ML	G	JS	B	CS	CWJ	VV	SENECA
Uniform	Well	0.31 ±.00	0.17 ±.00	0.22 ±.00	**0.16** ±.00	0.17 ±.00	0.17 ±.00	0.29 ±.00	0.18 ±.00 (7.2%)
	Under	1.40 ±.00	0.65 ±.01	1.35 ±.00	0.65 ±.01	0.65 ±.02	0.58 ±.01	0.56 ±.01	**0.43** ±.01 (0.0%)
Step	Well	0.32 ±.00	0.20 ±.00	0.23 ±.00	0.20 ±.00	**0.20** ±.00	0.20 ±.00	0.29 ±.00	0.20 ±.00 (2.4%)
	Under	1.32 ±.00	0.61 ±.01	1.26 ±.00	0.62 ±.01	0.59 ±.02	0.54 ±.01	0.55 ±.01	**0.43** ±.01 (0.0%)
Zipf-0.5	Well	0.31 ±.00	0.19 ±.00	0.21 ±.00	0.19 ±.00	**0.18** ±.00	0.19 ±.00	0.29 ±.00	0.19 ±.00 (3.0%)
	Under	1.31 ±.00	0.62 ±.01	1.25 ±.00	0.63 ±.01	0.59 ±.02	0.55 ±.01	0.55 ±.01	**0.46** ±.01 (0.0%)
Zipf-1	Well	0.33 ±.00	0.24 ±.00	0.23 ±.00	0.24 ±.00	**0.22** ±.00	0.25 ±.00	0.30 ±.00	0.22 ±.00 (0.1%)
	Under	1.05 ±.01	0.57 ±.01	0.88 ±.01	0.61 ±.01	0.51 ±.01	**0.50** ±.01	0.54 ±.01	0.52 ±.01 (2.8%)
Zipf-1.5	Well	0.33 ±.00	0.28 ±.00	0.27 ±.00	0.27 ±.00	0.26 ±.00	0.29 ±.00	0.30 ±.00	**0.26** ±.00 (0.0%)
	Under	0.70 ±.01	0.47 ±.01	0.55 ±.01	0.50 ±.01	**0.43** ±.01	0.47 ±.01	0.51 ±.01	0.43 ±.01 (0.7%)
Diri-0.5	Well	0.28 ±.00	0.22 ±.00	**0.22** ±.00	0.23 ±.00	0.22 ±.00	0.23 ±.00	0.28 ±.00	0.22 ±.00 (3.4%)
	Under	0.98 ±.01	0.47 ±.01	0.86 ±.00	0.49 ±.01	0.44 ±.01	0.46 ±.01	0.48 ±.01	**0.43** ±.01 (0.0%)
Diri-1	Well	0.29 ±.00	0.21 ±.00	0.20 ±.00	0.21 ±.00	**0.20** ±.00	0.21 ±.00	0.28 ±.00	0.20 ±.00 (1.1%)
	Under	1.13 ±.00	0.53 ±.01	1.05 ±.00	0.53 ±.01	0.51 ±.01	0.50 ±.01	0.52 ±.01	**0.44** ±.01 (0.0%)
BB-2	Well	0.31 ±.00	0.18 ±.00	0.21 ±.00	0.18 ±.00	**0.17** ±.00	0.18 ±.00	0.29 ±.00	0.18 ±.00 (3.5%)
	Under	1.32 ±.00	0.61 ±.01	1.27 ±.00	0.61 ±.01	0.61 ±.02	0.56 ±.01	0.55 ±.01	**0.44** ±.01 (0.0%)

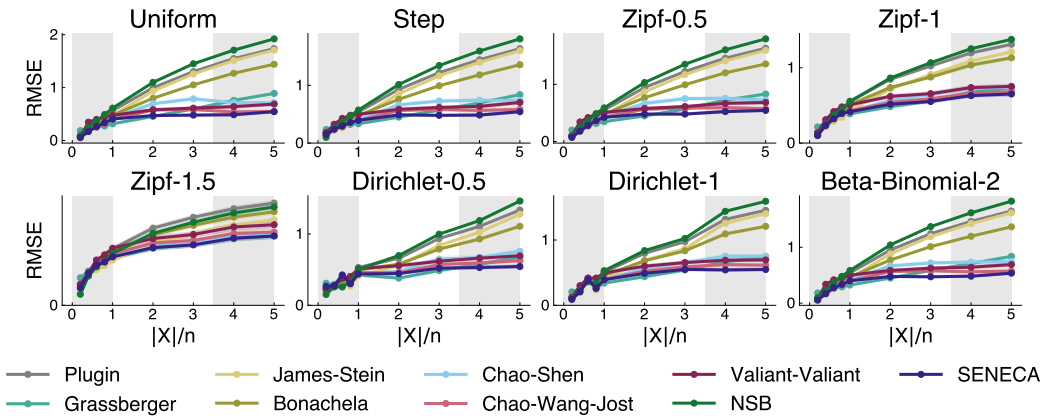

Figure 8: **Entropy estimation in small-sample item capture (all estimators, plus the NSB estimator).** For each of 8 families of label-prevalence distribution, each item's capture probability is determined uniformly over its label's prevalence probability mass. The number of labels $|\mathcal{X}|$ varies from 2 to 50 across trials. For each trial, $n = 10$ items are captured, and the entropy of the label-prevalence distribution is estimated according to each of seven methods. Results are averaged over 1000 trials per label count. Colored regions about each line plot depict 95% bias-corrected and accelerated confidence intervals, based on 1000 bootstrap replications. Instances for which the NSB estimator returned null values are ignored. In each plot, the leftmost grey region corresponds to the well-sampled regime (i.e., $|\mathcal{X}| = \frac{n}{5}, \frac{2n}{5}, \frac{3n}{5}, \frac{4n}{5}, n$), where items to the right correspond to the under-sampled regime (i.e., $|\mathcal{X}| = 2n, 3n, 4n, 5n$). The rightmost grey region corresponds to the support-risky regime, where no consistent estimator for support size exists.

D Comparison to the NSB estimator

In this section, we briefly consider the Nemenman-Shafee-Bialek (NSB) estimator, a Bayesian approach introduced in Nemenman et al. (2001). We benchmark against this estimator here, and not in prior sections, because NSB periodically returns null results (i.e., when $\Phi_1(X^n) = n$). Therefore, we discard said null instances (~6% of trials in the simulation study), but this makes the confidence intervals for NSB not directly comparable to the others'. Figure 8 adds the NSB estimator to Figure 6. There are two settings - the well-sampled instances of Zipf-1.5 and Dirichlet-0.5 - for which NSB appears quite strong. On the other hand, NSB does not perform reliably in the under-sampled settings herein.

E Additional LLM UQ results

In the main body of this work and in Figure 10, our LLM UQ experiments gauge the correctness of an LLM response using a consistency-based judge, described in Appendix B.5. Other related works employ a binary LLM judge that simply rates responses as correct or incorrect. We consider this approach, as well, using the following prompt:

Binary LLM-as-Judge

We are assessing the quality of answers to the follwoing question: {question}
The expected answer is: {groundtruth}
The proposed answer is: {pred}
Within the context of the question, does the proposed answer mean the same as the expected answer? Respond only with yes or no inside <rating></rating> tags. Only provide the rating, without any additional commentary.

The above prompt is drawn near-identically from Farquhar et al. (2024), but we request rating tags for easy parsing, like in the consistency-based prompt. A summary figure of the results, similar to Figure 4, is shown in Figure 11, with full results in Figure 11. We observe no qualitative difference arising from the difference in judge strategies.

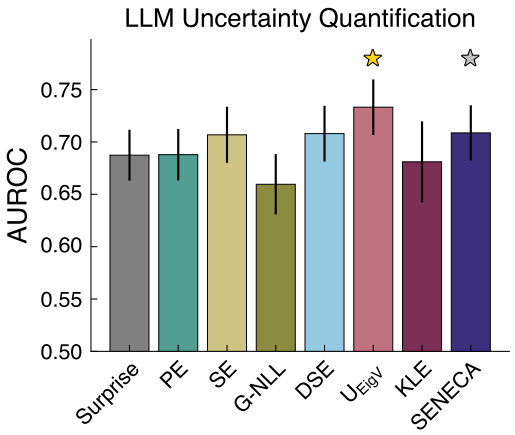

Figure 11: **Summary of LLM UQ results using binary LLM judge of correctness.** Using the binary LLM-as-judge prompt described in Appendix E, "best guess" responses are labeled as correct/incorrect and UQ methods are assessed for their ability to detect incorrect responses. AUROC values and parameter uncertainties are aggregated across models and datasets, as described in Section 5.2, to produce the above per-method summary. Gold and silver stars indicate the methods with highest and second-highest aggregated parameter estimates, respectively.

F THE MISSING MASS ESTIMATOR

Although our primary goal is estimating entropy from small sample sizes, our method for doing so involves a new "self-consistent" missing mass estimator. In this section, we evaluate $\hat{M}^{SC}$ in isolation. For baseline comparison, we consider the following alternatives, illustrating a range from high bias/low variance to low bias/high variance estimators: the classic Good-Turing method (Good, 1953); the so-called "minimal bias" estimator of Lee & Böhme (2025) (Lee-Böhme); and the closed-form, two-FoF estimator of Painsky (2023), which is arises from minimax risk minimization (Painsky). We also include a missing mass estimator based on the support size estimator of Wu & Yang (2019) (Wu-Yang), which is also described in Section B.2. Wu & Yang (2019) convert the Good-Turing missing mass into a support size estimator via $\widehat{|\mathcal{X}|}^{GT} = \frac{|X_1^n|}{1-\hat{M}^{GT}}$. By similar reasoning, Lee & Böhme (2025) construct a missing mass estimator from the Wu-Yang support size via $\hat{M}^{WY} = 1 - \frac{|X_1^n|}{\widehat{|\mathcal{X}|}^{WY}}$; we use the same, bounding values to the $[0, 1]$ interval.

In Figure 13, we conduct a similar analysis to that of Figures 2 and 7, displaying the RMSE, bias, and variance of missing mass estimators instead of those of entropy estimators. In general, our method exhibits comparable (albeit higher) variance to that of Painsky's two-FoF method, but does so with lower bias in the under-sampled regime. True to its name, the Lee-Böhme "minimal bias" estimator exhibits the lowest bias overall. Similar to Tables 1 and 2, Table 3 averages RMSE values from the data for Figure 13 within the well-sampled and under-sampled regimes. SENECA especially excels in the latter, with the lowest or second-lowest regime-averaged RMSE in $7/8$ under-sampled settings. This is particularly apparent among distributions that are fatter tailed (i.e., Uniform, Step, Zipf-0.5, Beta-Binomial-2), where the differences between the smallest and next-smallest errors are large. On the other hand, SENECA exhibits the second-lowest regime-averaged RMSE in $5/8$ well-sampled settings, where Painsky's method is generally preferable. In those settings, however, SENECA's aggregated RMSE is typically within a few percentage points of the lowest, underscoring balanced performance across regimes.

G AI USE STATEMENT

LLMs were employed for coding, statistical question-answering, and LaTeX formatting in the preparation of this work. The core ideas and written content were human-generated. Mathematical content was human-drafted and revised based on LLM feedback, and existing literature were discussed conversationally with LLMs during the preparation process.

Table 3: **Regime-averaged error for missing mass estimation in small-sample item capture.** Using the data from Figure 13 ($n = 10$), we calculate the root-mean-squared error (RMSE) for each estimator and average results within the well-sampled ($|\mathcal{X}| \leq n$) and under-sampled ($|\mathcal{X}| > n$) regimes. The lowest and second-lowest values in each row are bold and underlined, respectively. The estimators are Good-Turing (GT), Lee-Böhme (LB), Wu-Yang (WY), Painsky (P), and the underlying missing mass estimator used in SENECA (highlighted in blue). In parentheses, we show the amount by which SENECA's averaged RMSE is from the lowest value in each row.

Distribution	Regime	GT	LB	WY	P	SENECA
Uniform	$\|S\| \leq n$	0.177 ±.004	0.185 ±.008	**0.165** ±.004	0.167 ±.003	0.174 ±.003 (5.4%)
	$\|S\| > n$	0.204 ±.004	0.218 ±.004	0.289 ±.002	0.222 ±.003	**0.164** ±.003 (0.0%)
Step	$\|S\| \leq n$	0.172 ±.004	0.217 ±.008	0.187 ±.003	**0.160** ±.003	0.168 ±.003 (4.9%)
	$\|S\| > n$	0.215 ±.004	0.228 ±.004	0.257 ±.003	0.204 ±.004	**0.157** ±.004 (0.0%)
Zipf-0.5	$\|S\| \leq n$	0.173 ±.004	0.184 ±.009	0.169 ±.004	**0.163** ±.003	0.171 ±.004 (4.9%)
	$\|S\| > n$	0.214 ±.004	0.225 ±.005	0.254 ±.003	0.207 ±.003	**0.155** ±.004 (0.0%)
Zipf-1	$\|S\| \leq n$	0.161 ±.004	0.197 ±.010	0.174 ±.003	**0.148** ±.003	0.159 ±.003 (7.0%)
	$\|S\| > n$	0.219 ±.005	0.225 ±.005	**0.151** ±.004	0.183 ±.004	0.162 ±.004 (6.8%)
Zipf-1.5	$\|S\| \leq n$	0.148 ±.003	0.203 ±.012	0.190 ±.003	**0.131** ±.003	0.142 ±.003 (8.7%)
	$\|S\| > n$	0.192 ±.004	0.196 ±.005	**0.129** ±.003	0.159 ±.004	0.174 ±.003 (35.4%)
Dirichlet-0.5	$\|S\| \leq n$	0.144 ±.003	0.239 ±.013	0.191 ±.003	**0.134** ±.003	0.139 ±.003 (3.8%)
	$\|S\| > n$	0.238 ±.005	0.244 ±.005	**0.164** ±.004	0.173 ±.004	0.172 ±.004 (5.3%)
Dirichlet-1	$\|S\| \leq n$	0.153 ±.003	0.261 ±.007	0.194 ±.003	**0.144** ±.003	0.150 ±.003 (4.3%)
	$\|S\| > n$	0.226 ±.004	0.236 ±.005	0.205 ±.003	0.190 ±.004	**0.158** ±.004 (0.0%)
Beta-Binomial-2	$\|S\| \leq n$	0.174 ±.005	0.186 ±.011	**0.165** ±.004	0.165 ±.004	0.172 ±.004 (4.4%)
	$\|S\| > n$	0.212 ±.005	0.224 ±.005	0.265 ±.003	0.206 ±.003	**0.156** ±.004 (0.0%)

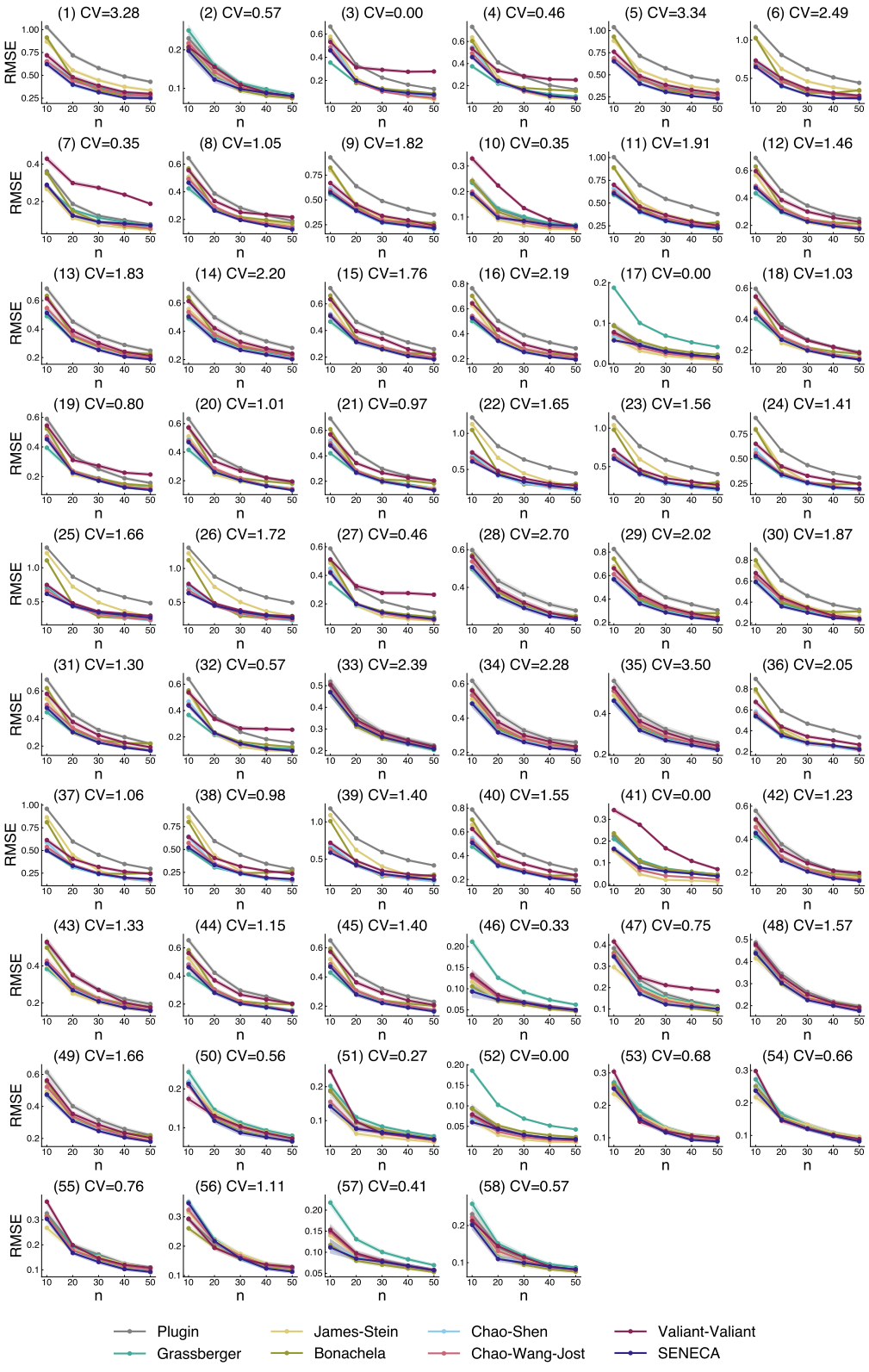

Figure 9: **Entropy estimation in** 58 **real species-prevalence distributions.** Collections of true bugs from Janzen (1973b) are treated as full populations, similar to Chao & Shen (2003), for each of which we compare 8 entropy estimators. Values are averaged over 1000 trials per sample size and population. Color regions about each line plot depict 95% bias-corrected and accelerated CIs, based on 1000 bootstrap replications. CI lower bounds are clipped to ensure non-negativity.

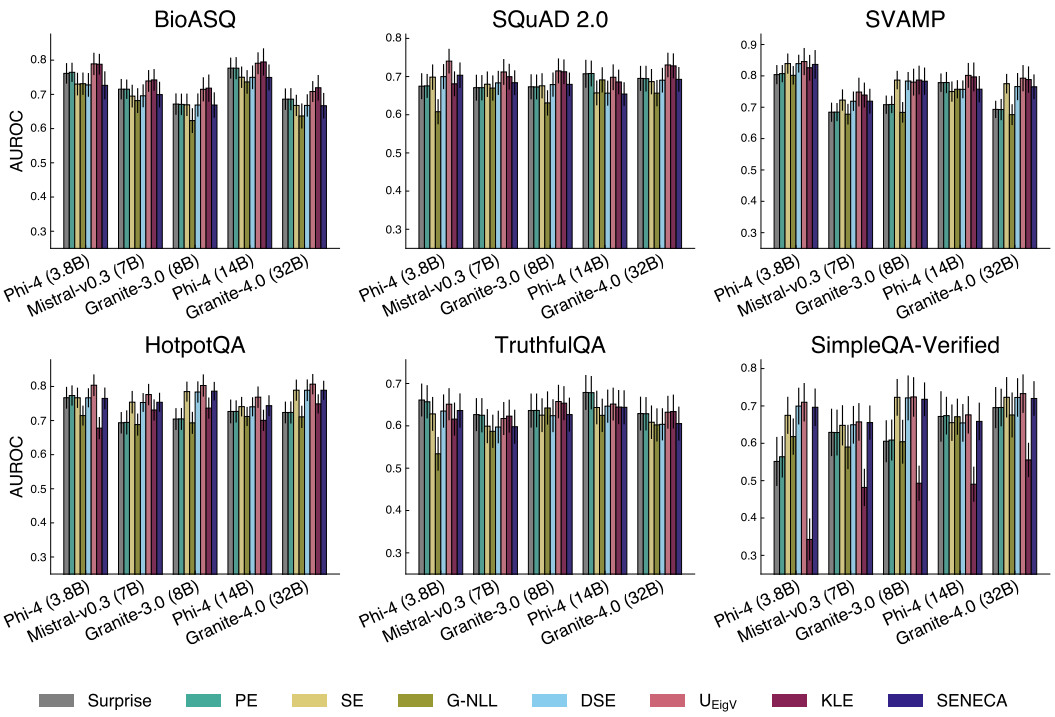

Figure 10: **LLM uncertainty quantification on six question-answering datasets (consistency judge).** 8 methods are employed to detect incorrect LLM responses, as determined by the consistency-based judge described in Section B.5. Methods are evaluated by area under the receiver operating characteristic curve (AUROC). 95% CIs about AUROC estimates are calculated via the DeLong method (DeLong et al., 1988; Sun & Xu, 2014). Results are aggregated and summarized in Figure 4.

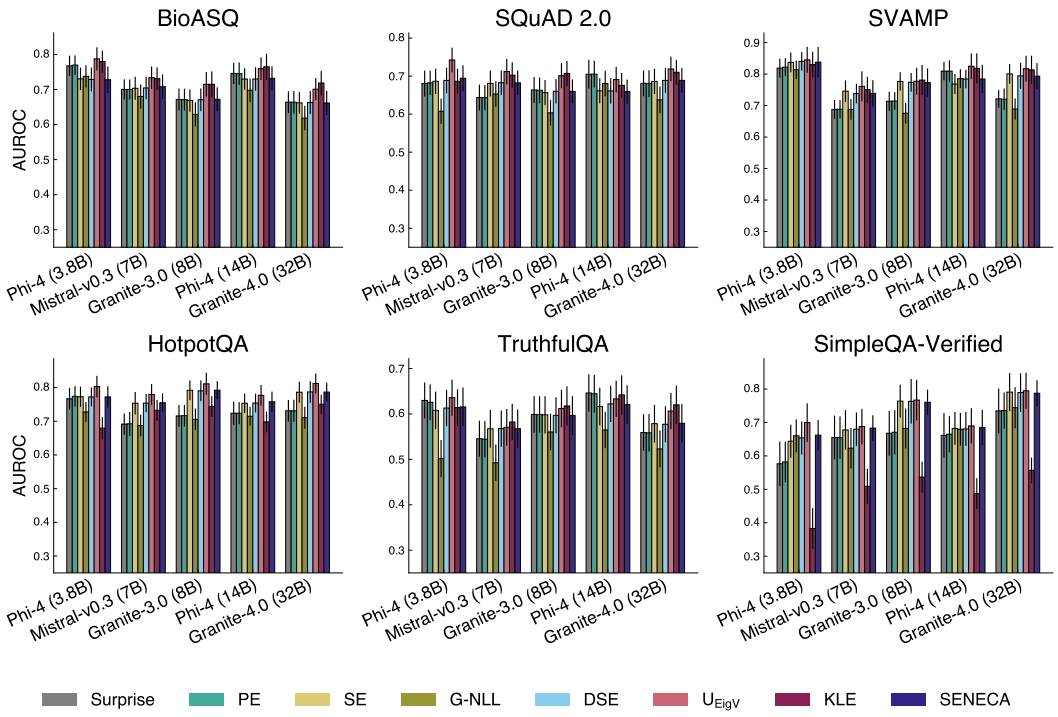

Figure 12: **LLM uncertainty quantification on six question-answering datasets (binary judge).** 8 methods are employed to detect incorrect LLM responses, as determined by the binary judge described in Section E. Methods are evaluated by area under the receiver operating characteristic curve (AUROC). 95% CIs about AUROC estimates are calculated via the DeLong method (DeLong et al., 1988; Sun & Xu, 2014). Results are aggregated and summarized in Figure 11.

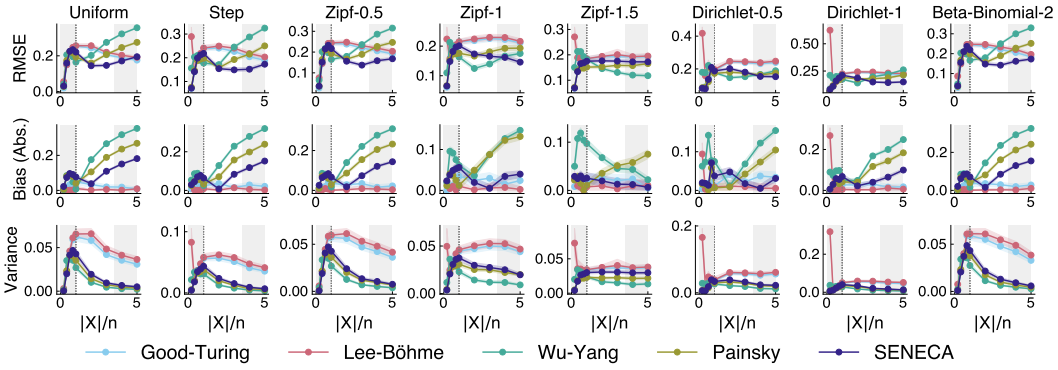

Figure 13: **Empirical RMSE, bias, and variance of missing mass estimation in small-sample item capture.** For each of 8 families of label-prevalence distribution, we generate 100 items per label. Each item's capture probability is determined uniformly over its label's prevalence probability mass. The number of labels $|\mathcal{X}|$ varies from 2 to 50 across trials. For each trial, $n = 10$ item-capture samples are drawn, and the missing mass of the label-prevalence distribution is estimated according to each of four methods. Results are averaged over 1000 trials per label count. Colored regions about each line plot depict 95% bias-corrected and accelerated confidence intervals, based on 1000 bootstrap replications. In each plot, leftmost grey region corresponds to the well-sampled regime (i.e., $|\mathcal{X}| = \frac{n}{5}, \frac{2n}{5}, \frac{3n}{5}, \frac{4n}{5}, n$), where items to the right correspond to the under-sampled regime (i.e., $|\mathcal{X}| = 2n, 3n, 4n, 5n$). The rightmost grey region corresponds to the support-risky regime, where no consistent estimator for support size exists.